



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**INSTRUMENTACIÓN DE UN SOCKET PROTÉSICO
A NIVEL TRANSFEMORAL**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECATRÓNICO

P R E S E N T A:

JORGE EDUARDO ESPARZA RAMOS



**DIRECTORA DE TESIS:
M.I. HANNA LESLYE GARCÍA GUERRA
CIUDAD UNIVERSITARIA, CD. MX, 2016**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A la Universidad Nacional Autónoma de México, por darme la oportunidad de formarme como profesional de la Ingeniería Mecatrónica.

A mi asesora, M.I. Hanna Leslie García Guerra por el apoyo, el tiempo que me dedicó, su paciencia y todos los consejos y enseñanzas que me dio para poder completar este trabajo.

A todos mis maestros de la carrera de los que aprendí no solo los aspectos teóricos y prácticos de la Ingeniería, sino también de la vida.

Agradezco al proyecto PAPIIT por el apoyo para la realización de este trabajo. Esta tesis se realizó en el marco del proyecto PAPIIT IT101915 "Mecatrónica en los sistemas biomédicos".

Dedicatoria

Este trabajo se lo dedico a mi mamá, quien me ha dado la fuerza para conseguir todos mis objetivos, el pilar de mi vida, y la persona que más admiro. Te agradezco todos los consejos, exigencias, y el impulso que siempre has tenido para mí, te quiero mucho.

A mi Lili, gracias por alentarme en todo momento, por creer en mí y por estar conmigo en las buenas y en las malas, eres el amor de mi vida.

A mi familia de la que siempre recibí mucho apoyo, en especial a mi tía Lupita y a mi tía Celia, gracias por estar en los momentos que siempre las necesité.

A mis amigos Chucho, Brian, Bere y Nancy gracias por los momentos tan agradables que hemos pasado juntos.

ÍNDICE

PLANTEAMIENTO DEL PROBLEMA.....	9
HIPÓTESIS	10
OBJETIVOS.....	10
JUSTIFICACIÓN.....	11
ALCANCES	11
CAPÍTULO 1. ANTECEDENTES	12
1.1 Discapacidad y amputación.....	12
1.1.1 Discapacidad	12
1.1.2 Estadísticas en México de la discapacidad	12
1.1.3 Amputación.....	13
1.2 Prótesis	14
1.2.1 Miembro inferior.....	14
1.2.2 Prótesis para miembro inferior	14
1.3 Clasificación de las prótesis para extremidades	15
1.3.1 Prótesis por su funcionalidad y apariencia	15
1.3.2 Prótesis por el lugar de la amputación.....	15
1.3.3 Prótesis transfemoral	16
1.4 Partes de una prótesis transfemoral	16
1.4.1 Socket.....	17
1.4.2 Rodilla artificial.....	17
1.4.3 Extensión tibial.....	19
1.4.4 Pie artificial.....	19
1.5 Tecnología actual	19
1.5.1 Materiales	20
1.5.2 Manufactura	21
1.5.3 Neumática e hidráulica	22
1.5.4 Control	23
1.5.5 Electromiografía.....	24
1.5.6 Mecanomiografía	24
1.6 Empresas dedicadas al desarrollo de prótesis de extremidades	24
1.6.1 C-Leg – Ottobock®	25
1.6.2 Genium – Ottobock®	26
1.6.3 Aqualine – Ottobock®	27
1.6.4 Endolite®	27

1.6.5 Total Knee – Össur®	28
1.6.6 Rheo Knee – Össur®	29
1.6.7 Power Knee – Össur®	29
1.6.8 Symbionic Leg – Össur®	30
1.6.9 Plié Knee – Freedom Innovations®	30
1.7 Patentes	31
1.7.1 Socket cojín neumático con relleno poroso	31
1.7.2 Socket protésico ajustable	32
1.7.3 Socket protésico con depósito de vacío autónomo	32
1.7.4 Sistema de ajuste dinámico de geometría variable para usar con un dispositivo corporal	33
 CAPÍTULO 2. DISEÑO CONCEPTUAL DE LA INSTRUMENTACIÓN	 36
2.1 Planteamiento de Requerimientos	39
2.1.1 Requerimientos	41
2.2 Presión, humedad y temperatura	41
2.2.1 Presión	42
2.2.2 Humedad y temperatura	42
2.3 Selección de elementos para el sistema electrónico	43
2.3.1 Microcontrolador	44
2.3.2 Módulo de Bluetooth	45
2.3.3 Sensor de temperatura y humedad	46
2.3.4 Sensor de presión	48
2.3.5 Servomotor	49
2.3.6 Suministro de energía	50
2.3.7 Componentes adicionales	51
2.4 Selección de elementos para el sistema neumático	52
2.4.1 Sujetadores	52
2.4.2 Conductos para el flujo de aire	53
2.4.3 Válvulas	53
2.4.4 Bomba neumática	54
2.4.4.1 Clasificación de las bombas	54
2.4.4.2 Presión máxima que debe suministrar la bomba a los sujetadores	56
2.4.4.3 Dimensiones de la bomba	56
2.4.4.4 Nivel de sonido	56
2.4.4.5 Características eléctricas	56
2.5 Selección de elementos para el Sistema de programación	57
2.5.1 Android	57
2.5.2 Interfaz de usuario para el socket protésico	59
2.6 Control del sistema de autoajuste del socket	60
2.6.1 Función de transferencia	60

2.6.2 Sistema de control del socket	60
2.6.3 Planta del sistema de autoajuste del socket	60
2.6.3.1 Relación del área de entrada y salida de aire con la posición angular del servomotor	61
2.6.3.2 Relación del caudal con el área de entrada y salida de aire	62
2.6.3.3 Relación entre la presión en los sujetadores y el caudal del aire	62
2.6.3.4 Función de transferencia del sistema	63
2.6.4 Controlador del sistema de autoajuste del socket	64
2.6.5 Control planteado para el sistema de autoajuste del socket	65
CAPÍTULO 3. DISEÑO DE CONFIGURACIÓN	67
3.1 Sistema electrónico	67
3.1.1 Microcontrolador	67
3.1.2 Módulo de Bluetooth	69
3.1.3 Sensor de temperatura y humedad	70
3.1.3.1 Sensor DHT22	72
3.1.4 Sensor de presión	74
3.1.5 Botón de inflado/desinflado	75
3.1.6 Servomotor	75
3.1.7 Circuito para la instrumentación	78
3.1.8 Batería para alimentar al circuito	78
3.2 Sistema neumático	79
3.2.1 Sujetadores	79
3.2.2 Servoválvula y conductos para el flujo de aire	80
3.2.3 Bomba neumática	84
3.3 Control del sistema de autoajuste	84
3.3.1 Ajuste en la planta del sistema de autoajuste	85
3.3.2 Ajuste en el controlador del sistema de autoajuste	86
3.3.3 Discretización del control del sistema de autoajuste	94
3.4 CAD del socket protésico con la instrumentación	97
3.5 Programación de la interfaz de usuario	101
3.5.1 Disposición del diseño de la interfaz	101
CAPÍTULO 4. RESULTADOS Y CONCLUSIONES	109
4.1 Pruebas y resultados	110
4.2 Conclusiones	113
CAPÍTULO 5. RECOMENDACIONES PARA EL TRABAJO A FUTURO Y EL DISEÑO DE DETALLE	116

APÉNDICE	117
A1. Microcontrolador PIC16F887	118
A2. Módulo de Bluetooth RN-42	124
A3. Sensor DHT22	131
A4. Sensor MPX5050DP	137
A5. Servomotor SG90.....	145
A6. Esquema del circuito del sistema	148
A7. Código de programación del microcontrolador.....	150
A8. Programación de la aplicación desarrollada en App Inventor	160
REFERENCIAS.....	167

Índice de Figuras

Capítulo 1. ANTECEDENTES

Figura 1.1.	Partes de una prótesis de miembro inferior a nivel transfemoral considerando el socket y la extensión femoral como partes individuales.	16
Figura 1.2.	Partes de una prótesis de miembro inferior a nivel transfemoral considerando el socket y la extensión femoral como una sola parte.	17
Figura 1.3.	Rodilla de fricción constante con un eje.	18
Figura 1.4.	Sistema de bloqueo de rodilla accionado por el peso (dos ejes).	18
Figura 1.5.	Rodilla policéntrica.	18
Figura 1.6.	Ejemplos de extensión tibial (endoesquelética y exoesquelética).	19
Figura 1.7.	Prótesis Exo impresa en 3D.	22
Figura 1.8.	Prótesis con sistema neumático APSS. a) cámara de aire, b) tarjeta de circuito de control, c) bomba, d) válvula, e) batería, f) sistema de operación, g) prótesis transtibial ensamblada [38].	23
Figura 1.9.	Funcionamiento del sistema APSS.	24
Figura 1.10.	Prótesis C-Leg.	26
Figura 1.11.	Prótesis Genium.	26
Figura 1.12.	Prótesis Aqualine.	27
Figura 1.13.	Pie élan de Endolite®.	27
Figura 1.14.	Rodilla prostética policéntrica KX06® de Endolite®.	28
Figura 1.15.	Prótesis de rodilla Total Knee® 2100 - Össur®.	28
Figura 1.16.	Prótesis de rodilla Rheo Knee® 2100 - Össur®.	29
Figura 1.17.	Prótesis de rodilla Power Knee® - Össur®.	29
Figura 1.18.	Prótesis transfemoral Symbionic Leg® - Össur®.	30
Figura 1.19.	Prótesis transfemoral The Pilé® - Freedom Innovations®.	30
Figura 1.20.	Socket con cojín neumático y relleno poroso.	31
Figura 1.21.	Socket protésico ajustable.	32
Figura 1.22.	Socket protésico con depósito de vacío.	33
Figura 1.23.	Socket con sistema dinámico de geometría.	33

Capítulo 2. DISEÑO CONCEPTUAL

Figura 2.1.	Diagrama de tiempo para el envío de datos [59].	48
Figura 2.2.	Sensor DHT11 [59].	48
Figura 2.3.	Ejemplo de PWM para un servomotor.	50
Figura 2.4.	Sujetadores lateral y medial, anterior, posterior y superior [44].	53
Figura 2.5.	Válvula de tres vías [23].	54
Figura 2.6.	Logotipo de Android Studio.	58
Figura 2.7.	Logotipo de App Inventor.	58
Figura 2.8.	Logotipo de Eclipse.	58

Figura 2.9.	Modelo de un sistema de control retroalimentado.	60
Figura 2.10.	Posiciones angulares de la válvula en las que existe intersección de áreas [23].	61
Figura 2.11.	Esquema del control planteado para el socket [23].	65
Figura 2.12.	Esquema del control con función de transferencia total de la planta del socket [23].	66

Capítulo 3. DISEÑO DE CONFIGURACIÓN

Figura 3.1.	Microcontrolador PIC16F887 utilizado para la instrumentación.	69
Figura 3.2.	Tarjeta para el módulo RN42 de Sparkfun [74].	70
Figura 3.3.	Sensor de temperatura y humedad DHT22.	72
Figura 3.4.	Sensor de presión MPX5050DP.	74
Figura 3.5.	Servomotor SG90.	76
Figura 3.6.	Características del PWM para el servomotor SG90.	76
Figura 3.7.	Circuito Impreso para la instrumentación del socket.	78
Figura 3.8.	Zonas de apertura de la válvula de tres vías.	80
Figura 3.9.	Posición de Bloqueo de la servoválvula.	81
Figura 3.10.	Posición de mínima entrada de aire hacia los sujetadores.	82
Figura 3.11.	Posición de máxima entrada de aire hacia los sujetadores.	82
Figura 3.12.	Posición de mínima salida de aire desde los sujetadores.	83
Figura 3.13.	Posición de máxima salida de aire desde los sujetadores.	83
Figura 3.14.	Bomba neumática.	84
Figura 3.15.	Utilización de la herramienta rltool de Matlab.	91
Figura 3.16.	LGR mediante la herramienta rltool de Matlab.	92
Figura 3.17.	Polos de lazo abierto mostrados en el LGR mediante la herramienta rltool.	93
Figura 3.18.	Diagrama de bloques del control del socket mediante Simulink.	93
Figura 3.19.	Comportamiento de la presión con el control PI.	93
Figura 3.20.	Discretización de una señal en tiempo continuo.	94
Figura 3.21.	Mapa de flujo reducido de la rutina del control PI en el microcontrolador.	96
Figura 3.22.	Vista de sección superior del socket con la ubicación de la servoválvula, el sensor de presión y la batería en la base del socket.	98
Figura 3.23.	Vista de sección superior del socket con la ubicación de la bomba y el circuito.	98
Figura 3.24.	Aumento de 25 mm de largo en la base del socket.	99
Figura 3.25.	Distancia mínima entre la malla y el elemento más cercano en la parte inferior.	99
Figura 3.26.	Vista de sección que muestra el acomodo de los elementos de cada sistema en la parte interna de la base del socket.	100
Figura 3.27.	Diseño final del socket.	100
Figura 3.28.	Dos de las posibles primeras visualizaciones de la interfaz de usuario una vez instalada la aplicación en el dispositivo móvil.	101

Figura 3.29.	Indicación de la aplicación para conectar al dispositivo móvil con el socket.	102
Figura 3.30.	Ajustes del Bluetooth del dispositivo móvil.	103
Figura 3.31.	Emparejamiento entre el socket y el dispositivo móvil.	103
Figura 3.32.	Aviso de ayuda de la primera visualización de la interfaz.	104
Figura 3.33.	Conexión entre el socket y el dispositivo móvil.	104
Figura 3.34.	Visualizaciones donde se muestran los botones de presión, temperatura y humedad, información y salir. Además se muestra el aviso de información al presionar el botón con el mismo nombre.	105
Figura 3.35.	Visualización inicial de la actividad para el monitoreo y control de la presión.	106
Figura 3.36.	Diferentes ajustes de presión y la respuesta del sistema de control, mostrados en la gráfica.	106
Figura 3.37.	Almacenaje de la gráfica del control y monitoreo de la presión en los sujetadores.	107
Figura 3.38.	Visualización de la actividad para el monitoreo de los datos de la temperatura y la humedad en el socket.	107
Figura 3.39.	Almacenaje de los datos de temperatura y humedad.	108
Figura 3.40.	Actividad principal luego de desconectar al socket y al dispositivo móvil.	108
 Capítulo 4. RESULTADOS Y CONCLUSIONES		
Figura 4.1.	Modelo para las pruebas de la instrumentación en el socket.	109
Figura 4.2.	Modelo con simulación de un muñón mediante una bolsa de aire, listo para realizar las pruebas.	110
Figura 4.3.	Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms). La línea azul representa la señal de referencia de 0 a 70 mmHg. La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.	111
Figura 4.4.	Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms). La línea azul representa la señal de referencia con un cambio de 70 a 90 mmHg. La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.	112
Figura 4.5.	Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms). La línea azul representa la señal de referencia de 70 mmHg. La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.	112
Figura 4.6.	Gráfica de temperatura (°C) y humedad (%HR) contra el tiempo (cada punto equivale a 2 minutos). La línea azul representa los valores de humedad en el socket cada dos minutos en la última hora de utilización de su utilización. La línea naranja indica los datos de temperatura en el socket en el mismo intervalo de tiempo que los obtenidos de la humedad.	113

Índice de Tablas

Capítulo 2. DISEÑO CONCEPTUAL

Tabla 2.1.	Clases de Bluetooth.	46
------------	----------------------	----

Capítulo 3. DISEÑO DE CONFIGURACIÓN

Tabla 3.1.	Características del microcontrolador PIC16F887.	69
Tabla 3.2.	Características de distintos sensores de temperatura y humedad.	71
Tabla 3.3.	Componentes eléctricos y electrónicos de la instrumentación.	78

Resumen

El presente proyecto es un nuevo paso en el diseño del socket autoajutable para prótesis a nivel transfemoral propuesto en las tesis "Diseño de un socket ajustable para prótesis de miembro inferior" [2], "Diseño de un socket autoajutable para prótesis de miembro inferior" [44] y "Diseño de un sistema de autoajuste para un socket protésico a nivel transfemoral" [23]. Un socket es la parte de una prótesis con la que el usuario hace contacto directo con ésta. El sistema de autoajuste permite que el socket se ajuste automáticamente al miembro residual de un usuario con amputación transfemoral, esto lo realiza mediante bolsas de aire (sujetadores) controladas por un sistema electrónico y neumático. El sistema de autoajuste tiene un control PI para el ajuste de la presión en los sujetadores del socket, el cual es implementado en un microcontrolador, el sistema neumático es impulsado por una bomba de aire y controlado por válvulas. En este nuevo trabajo, se realizó la instrumentación del socket protésico del sistema mencionado para poder detectar parámetros como temperatura, humedad y presión en los sujetadores. Para analizar estos parámetros y controlar el nivel de presión en los sujetadores, se diseñó una interfaz de usuario mediante una aplicación en Android. Estos datos son recabados por la aplicación, a través de Bluetooth. En trabajos anteriores, el control PI de la presión en los sujetadores presentaba un tiempo de asentamiento mayor a un minuto, el cual fue mejorado significativamente en siete segundos. Por otro lado, se seleccionó una bomba con un nivel de sonido menor al que presentaba la utilizada en un diseño anterior del socket, con el fin de beneficiar al usuario. Finalmente, se puede asegurar que la instrumentación y la mejora en los sistemas de control y neumático le permiten ahora al usuario contar con un socket más seguro y confiable.

Esta tesis se realizó en el marco del proyecto PAPIIT IT101915 "Mecatrónica en los sistemas biomédicos".

Planteamiento del problema

Es necesario comprobar el funcionamiento del socket autoajustable para amputación a nivel transfemoral, con el fin de saber si le es útil al usuario de prótesis. Para ello, se requiere conocer las variables físicas de la persona como la temperatura y la humedad en el muñón, además de la presión en los sujetadores. Estas variables pueden ser analizadas por un especialista en salud para recomendar al usuario tomar ciertas medidas de uso del socket. Esto podría prevenir la presencia de anomalías en el muñón del usuario de la prótesis, tales como abscesos, isquemia, procesos infecciosos, gangrena, etc. Por otro lado, estas variables le pueden servir a un ingeniero para realizar ajustes al diseño del socket y la prótesis. Un problema existente es que actualmente no se cuenta con una interfaz que permita conocer los valores de estas propiedades físicas. Así que, dentro de la instrumentación del socket, se realizará una interfaz en una aplicación que pueda generar una mejor realimentación para el usuario de su prótesis. Otro problema que presenta el socket es el tiempo que tarda el control del sistema de autoajuste en alcanzar su presión en los sujetadores, ante un cambio en el volumen del muñón, algo que se buscará mejorar en este trabajo. Y finalmente, el nivel de sonido actual del sistema neumático puede llegar a ser molesto para el usuario de la prótesis, por lo que se tendrá en cuenta reducir dicho nivel.

Hipótesis

La instrumentación del socket protésico a nivel transfemoral permitirá hacer más eficiente la interacción con el sistema de autoajuste y permitirá obtener las variables necesarias al realizar las pruebas físicas con el usuario de la prótesis. Esto mejorará el diseño del socket para hacerlo más confiable y seguro.

Objetivos

General

- Instrumentar, mejorar el control y el sistema neumático de un socket protésico autoajustable a nivel transfemoral.

Particulares

1. Medir temperatura, humedad y presión en el socket.
2. Transferir los datos de temperatura, humedad y presión hacia un dispositivo móvil con sistema operativo Android.
3. Desarrollar una aplicación en Android para monitorear y almacenar datos de temperatura, humedad y presión, así como para controlar el sistema de autoajuste del socket.
4. Mejorar el control del sistema de autoajuste.
5. Realizar mejoras al sistema neumático del sistema de autoajuste del socket.

Justificación

El desarrollo de este proyecto tiene como justificación el hecho de mejorar el sistema de autoajuste diseñado en la tesis [2]. La instrumentación permitirá tener datos de temperatura, humedad y presión en los sujetadores del socket para poder realizar pruebas físicas con el usuario, utilizando el socket. Esto permitirá al usuario de prótesis para amputación a nivel transfemoral, tener un socket más práctico, cómodo, confiable y seguro.

Alcances

El presente proyecto es un paso más en el diseño del socket autoajutable para prótesis a nivel transfemoral propuesto en las tesis [2], [23] y [44].

En estos trabajos se diseñó un socket que lograra solucionar los problemas presentes en el muñón de un usuario con amputación a nivel transfemoral, tales como el cambio en el volumen, dolor, lesiones, alteraciones en la piel, etc.

Ahora, en este trabajo, se busca instrumentar al socket protésico del sistema mencionado permitiendo detectar parámetros como presión, temperatura y humedad, para posteriormente enviar esta información hacia un dispositivo móvil. Además se desarrollará una aplicación en el SO Android capaz de controlar el sistema de autoajuste, monitorear y almacenar los datos recibidos de presión, temperatura y humedad.

El sistema se llevará hasta la etapa de diseño de configuración.

Capítulo 1. ANTECEDENTES

1.1 Discapacidad y amputación

1.1.1 Discapacidad

La Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud (CIF), desarrollada por la Organización Mundial de la Salud, define a la discapacidad como un término que engloba las deficiencias, limitaciones en la actividad, o las restricciones en la participación [24].

De acuerdo con la OMS, más de mil millones de personas en el mundo sufren alguna discapacidad de cualquier tipo. Esto representa cerca del 15% de la población total [25].

1.1.2 Estadísticas en México de la discapacidad

En México, hasta el año 2012 se contabilizaron 6.1 millones de habitantes que vivían con algún tipo de discapacidad (ver, escuchar, comunicarse, caminar, poner atención, aprender, etc.), lo que representa el 6.6% de la población nacional. Según el Instituto Nacional de Estadística y Geografía, dentro de este porcentaje más de la mitad de las personas son adultos mayores (51.4%). Posteriormente se hallan los adultos de entre 30 y 59 años con un 33.7%. Luego, los jóvenes se encuentran con el 7.6% de 15 a 29 años de edad. Al final están los niños de cero a catorce años con el 7.3% de la población con alguna discapacidad en México. Esto concluye que la mayor parte de las personas con algún tipo de discapacidad tienen más de 29 años (85% del total de esta población) [26].

De acuerdo a lo reportado por el INEGI, en el tema de género, son más las mujeres con discapacidad que los hombres con un 52.3%, del total de las personas con discapacidad. Sin embargo, esto depende del grupo de género, pues en los niños y jóvenes, es más alto el porcentaje en el sexo masculino [26].

El INEGI, a través de la Encuesta Nacional de Ingresos y Gastos de los Hogares del 2012 [26], reportó también que entre menos ingresos cuente un hogar, mayor será la presencia de personas con discapacidad. Además, se encontró en esta encuesta que los hogares en los que no hay personas con discapacidad tienen mayores ingresos que en los que sí hay. Las causas de esto se sugiere pueden ser que por lo general, alguien debe tener la responsabilidad del cuidado y la atención de la persona discapacitada, por la estructura familiar, el número de habitantes, o el tipo de hogar en que se viva.

Por otro lado, el tipo de discapacidad más recurrente en México es la de caminar con un 57.5%, seguida de la de ver, oír, hablar o comunicarse, mental, atender el cuidado personal, y por último la de poner atención. Además, se debe señalar, que las discapacidades para caminar, ver y oír son más frecuentes en adultos y adultos mayores que en niños y jóvenes. De acuerdo al INEGI, las personas que presentan al menos dificultad al caminar son en su mayoría adultos y adultos mayores, con 49% y 71.9% del total de la población con discapacidad [26].

La enfermedad es la principal causa de discapacidad en México con un 38.5%, le siguen la edad avanzada con el 31%, el nacimiento con el 15% y al final los accidentes con el 12%. En el caso de las personas que viven con alguna discapacidad para caminar, las principales causas encontradas son tener alguna enfermedad (42.3%), tener edad avanzada (30.3%) y haber sufrido algún accidente (20%) [26].

1.1.3 Amputación

Una amputación es la pérdida de una extremidad debido a un trauma, a una cirugía o a una enfermedad. En el caso de las amputaciones traumáticas, puede haber surgido de un accidente automovilístico, guerra, de accidentes con herramientas o maquinaria, etc. [29]. Mientras que otro tipo de amputaciones surgen de complicaciones con alguna enfermedad, siendo la más frecuente la diabetes.

De acuerdo a datos del Instituto Mexicano del Seguro Social, el 70% de las amputaciones realizadas ahí, son debido a complicaciones en pacientes diabéticos, que no fueron controlados adecuadamente o no recibieron una atención médica oportuna [27]. La razón por la cual una persona con diabetes es altamente propensa a sufrir una amputación es debido a los cambios fisiológicos que sufre el paciente. Una de las causas es la disminución en la irrigación sanguínea que genera una caída en la sensibilidad al dolor y, al momento de tener alguna lesión, el paciente no se da cuenta de la gravedad de un daño que puede generar la amputación parcial o total de la extremidad [28].

La pérdida de una extremidad inferior por arriba del nivel de la rodilla se le conoce como amputación transfemoral, y es uno de los casos más comunes en amputaciones. Por ejemplo, de acuerdo a la Secretaría de Salud, en México en el año 2013, hubo 75 mil casos de amputaciones, de los cuales más de la cuarta parte fueron arriba de la rodilla [75].

Las personas que sufren algún tipo de amputación deben de readaptarse a su vida, para ello en el caso de las amputaciones transfemorales, es necesario el uso de una prótesis. El usuario de una prótesis debe de comenzar su proceso de adaptación a ésta entendiendo que el muñón es una nueva parte de su cuerpo. Se debe trabajar también en la recuperación de los músculos perdidos así como en la detección de aquellos que aún le son útiles al usuario de la prótesis.

El hecho de adquirir una prótesis por parte de la persona que sufrió una amputación de una extremidad no es únicamente el sustituir a ésta última. El usuario debe ser atendido psicológicamente debido al trauma emocional ocasionado por la amputación. Es incluso recomendable que especialistas tanatológicos traten al usuario debido al duelo que existe por la pérdida de una parte de su cuerpo.

Además de los tratamientos psicológicos que el usuario de una prótesis debe llevar, también se debe tener un entrenamiento para el uso de la prótesis. Dicho entrenamiento puede durar de seis meses hasta un año, con el objetivo primordial de hacer que el usuario

recupere en gran medida sus capacidades motoras. Cuando una persona pierde una extremidad inferior y posteriormente comienza a utilizar una prótesis, existe un cambio en la simetría del cuerpo, la cual se debe de tratar mediante las terapias y el entrenamiento con la nueva prótesis. Otros objetivos importantes de una prótesis para extremidad inferior es lograr una bipedestación eficiente, lograr un movimiento natural de la marcha y la reincorporación del usuario a sus actividades normales [28].

1.2 Prótesis

Prótesis tiene su origen en el griego antiguo “prothesis” que significa puesta o aplicación [1]. Es un aparato usado en medicina para reemplazar alguna parte del cuerpo amputada debido a algún trauma o alguna enfermedad.

Los aparatos protésicos son reemplazos artificiales de alguna parte del cuerpo dañada o perdida, como los ojos, los dientes, las piernas, los brazos, las manos, etc.

Pueden haber múltiples razones por las que una persona llegue a necesitar una prótesis, incluyendo causas como accidente o defectos congénitos. Una extremidad artificial es un tipo de prótesis la cual reemplaza a una extremidad que ha sido perdida, como un brazo o una pierna [4].

Para la selección de los materiales, así como el establecimiento del tamaño y la forma utilizados en una prótesis de miembro inferior a nivel transfemoral, se toman en cuenta diversos aspectos. Entre ellos se encuentra el peso de la prótesis, el cual debe ser similar al miembro amputado [8], y el tamaño, que depende del material a ser utilizado y la forma del miembro residual del usuario.

1.2.1 Miembro inferior

El miembro inferior es la parte del aparato locomotor que se divide en cuatro segmentos principales: pelvis o cadera, muslo, pierna y pie [2].

1.2.2 Prótesis para miembro inferior

Con base en trabajos previos realizados [2], se puede establecer que la prótesis para miembro inferior ofrece principalmente las siguientes funciones:

- Capacidad de apoyo estático
- Flexión de la prótesis en la fase de oscilación
- Suplir la función de las articulaciones inexistentes después de la amputación
- Capacidad de control de movimiento o interacción entre el usuario y la prótesis
- Capacidad de amortiguación ante impactos
- Conseguir una marcha normal

1.3 Clasificación de las prótesis para extremidades

El tipo de prótesis a utilizar dependerá de diversos factores a tomar en consideración. Las prótesis pueden ser seleccionadas o diseñadas de acuerdo a las necesidades del usuario. El usuario puede necesitar una prótesis para diferentes tareas, dependiendo de la edad que tenga, algunas tareas específicas que desee desarrollar o incluso para la vida cotidiana.

Las prótesis para extremidades amputadas se pueden clasificar en dos categorías, por su funcionalidad y apariencia o por el lugar de la amputación.

1.3.1 Prótesis por su funcionalidad y apariencia

De acuerdo a las necesidades del usuario de prótesis relacionadas al uso y funcionabilidad así como a la estética de ésta, los tipos de prótesis existentes para extremidades amputadas son tres:

- Prótesis cosméticas o estéticas, las cuales son ligeras y generalmente baratas, teniendo como desventaja un grado de movimiento muy limitado y únicamente pueden tomar objetos muy ligeros, en este caso el usuario le deja a sus extremidades restantes hacer todas las funciones principales [3]; este tipo de prótesis, se le conoce también como pasiva, ya que no necesita de algún elemento que le proporcione energía adicional para realizar cierta acción [11].
- Prótesis impulsadas por el cuerpo, aquellas que necesitan de los músculos de la zona cercana a la prótesis para poder trabajar, permiten tener más grados de libertad y sentir al usuario la fuerza pero solo pueden controlar un movimiento a la vez y causan fatiga rápidamente [3].
- Prótesis mioeléctricas, que utilizan la acción de las señales eléctricas en los músculos residuales de la extremidad amputada; este tipo de prótesis amplifica dichas señales para mover motores eléctricos; sin embargo, tienden a ser prótesis más pesadas que las de tipo convencional y por lo general son más caras [3]; tanto en este tipo de prótesis como en las que son impulsadas por el cuerpo, se requiere del empleo de una fuerza externa (un motor o algún otro actuador) o interna para que la prótesis realice la acción, debido a ello, se les conoce como prótesis activas [11].

1.3.2 Prótesis por el lugar de la amputación

Con base en la ubicación de la amputación, las prótesis para extremidades amputadas se pueden dividir en dos grupos: para miembro superior y para miembro inferior:

- Prótesis de miembro superior (incluyendo hombros, brazos y manos), este tipo incluye la prótesis transradial (miembro artificial que reemplaza una mano perdida debajo del codo) y la prótesis transhumeral (aquellas que reemplazan una mano perdida por arriba del codo).
- Prótesis de miembro inferior (incluyendo caderas, piernas y pies), que se subdivide en prótesis transtibial (aquel miembro artificial que reemplaza una pierna perdida

por debajo de la rodilla) y prótesis transfemoral (aquel miembro artificial que reemplaza una pierna perdida por arriba de la rodilla) [4].

1.3.3 Prótesis transfemoral

Dado que el presente trabajo se enfoca en las prótesis transfemorales, se presenta a continuación la definición de dicho tipo prótesis. Se entiende como prótesis para amputación transfemoral aquel dispositivo externo capaz de reemplazar la ausencia del miembro inferior, en donde la rodilla ha sido removida pero aún se conserva parte del fémur [2], [7].

1.4 Partes de una prótesis transfemoral

Los principales elementos de una prótesis para miembro inferior a nivel transfemoral [8] se muestran en la Figura 1.1 y son los siguientes: el socket, la extensión femoral, la rodilla artificial, la extensión tibial y el pie. Sin embargo, el socket puede llegar a hacer en determinado momento la función de la extensión femoral, como se muestra en la Figura 1.2 [10], en donde el socket es conectado directamente a la rodilla artificial.

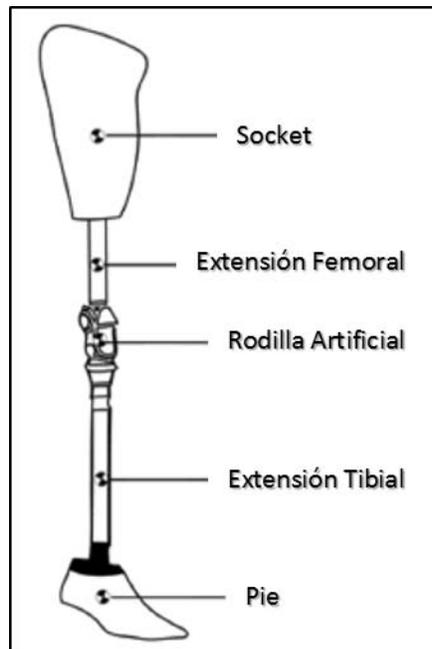


Figura 1.1. Partes de una prótesis de miembro inferior a nivel transfemoral considerando el socket y la extensión femoral como partes individuales.

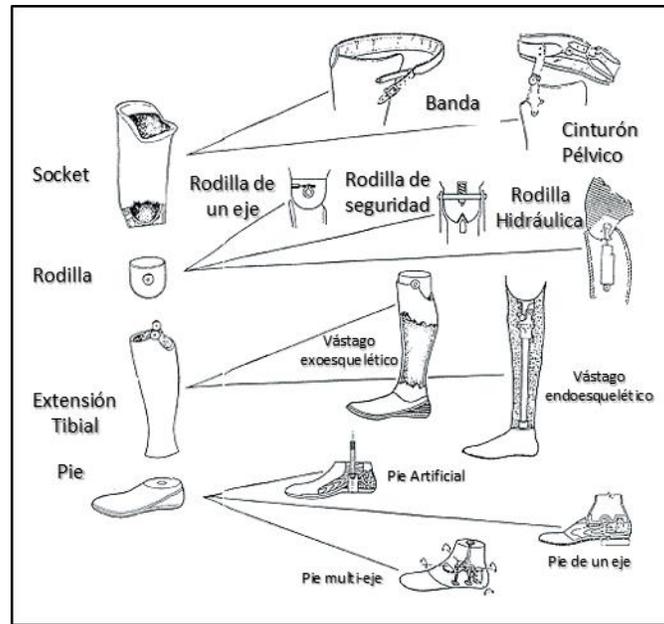


Figura 1.2. Partes de una prótesis de miembro inferior a nivel transfemoral considerando el socket y la extensión femoral como una sola parte.

1.4.1 Socket

El socket es la parte de la prótesis en la que el usuario hace contacto directo (mediante la suspensión dentro del socket) con la prótesis. Por medio del socket, se transfiere el peso del usuario desde la amputación hasta el piso a través del resto de la prótesis [10].

El socket debe tener un diseño tal que sea cómodo para el usuario y le sea funcional. Se debe evitar que el usuario presente problemas de circulación causados por el socket, teniendo en consideración que éste debe estar bien sujeto al muñón.

1.4.2 Rodilla artificial

Existen múltiples formas de conectar el socket con la extensión tibial, para poder simular una rodilla artificial. La apariencia debe ser normal al caminar, y la forma más simple de lograr esto es utilizando un perno que conecte el socket con la extensión tibial. Cabe señalar que el eje del perno se debe colocar de tal forma que no atraviese la trayectoria del peso del cuerpo hacia el suelo, para evitar una deformación mientras el usuario se encuentra de pie. La principal desventaja de tener un solo eje es que la apariencia únicamente sería normal a una velocidad constante al caminar y el usuario tiene que evitar superficies irregulares para evitar tropiezos [10]. En la Figura 1.3 se muestra un ejemplo de lo explicado previamente [10].

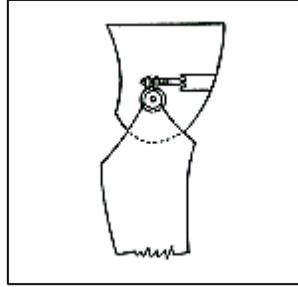


Figura 1.3. Rodilla de fricción constante con un eje.

Otro sistema utilizado en prótesis transfemorales [10] para simular la articulación entre el fémur y la tibia es el bloqueo de rodilla accionado por el peso; el cual consiste en dos pernos, uno pivotea en el otro cuando el usuario se encuentra de pie y la fuerza del peso del cuerpo se acopla a un freno que evita que la rodilla artificial se pandee. En la Figura 1.4 se muestra un ejemplo de este tipo de rodilla artificial [15].

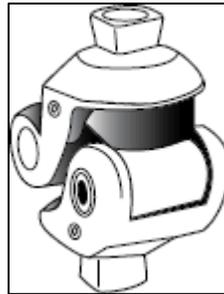


Figura 1.4. Sistema de bloqueo de rodilla accionado por el peso (dos ejes).

Las rodillas policéntricas, son otro tipo de sistema utilizado en la implementación de rodillas artificiales. En este tipo de rodilla artificial el centro de rotación es móvil con el fin de dar una mayor estabilidad al talón [10]. El mecanismo que utiliza este tipo de rodilla, es de cuatro barras y se basa en un centro móvil de rotación que se bloquea dependiendo el grado de flexión de la rodilla [14]. En la Figura 1.5 se da un ejemplo gráfico del mecanismo de este tipo de rodillas [15].

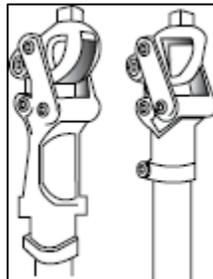


Figura 1.5. Rodilla policéntrica.

Algunos otros sistemas de rodilla artificial utilizan la ayuda de la hidráulica y la neumática para poder funcionar. Para ello, en el caso más simple, un pistón se conecta a un pivote en la parte del socket o en la extensión femoral cerca del perno de la rodilla, y un cilindro es

conectado a un pivote en la extensión tibial en la prótesis [10]. Con esto se logra una especie de amortiguamiento que ayuda a regular la velocidad al momento de caminar.

1.4.3 Extensión tibial

La principal tarea de la extensión tibial, o el vástago, es la de transferir las cargas verticales que se producen por el peso del usuario hacia el piso. Existen dos tipos, los de exoesqueleto y los de endoesqueleto. En los primeros, las fuerzas se distribuyen a través de las paredes externas de la extensión tibial. En los de tipo endoesqueleto, las fuerzas se distribuyen en la estructura central que por lo general se encuentra recubierta por una carcasa con forma de pierna, solo para dar estética a la prótesis [10]. En la Figura 1.6 se muestran estos dos tipos de extensiones tibiales [10].

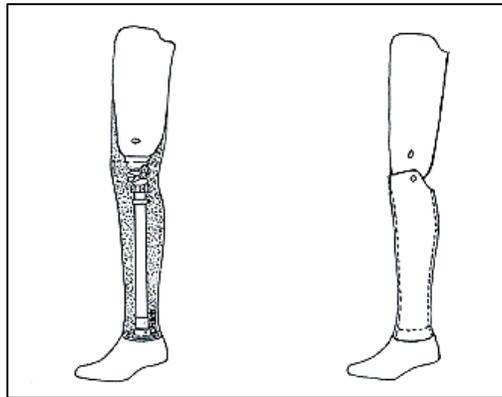


Figura 1.6. Ejemplos de extensión tibial (endoesquelética y exoesquelética).

1.4.4 Pie artificial

El sistema final, que hace contacto directo con el piso es el del pie artificial. Este sistema puede tomar en consideración también al tobillo, pero eso depende del tipo de diseño y precisión que se desee implementar, ya que cada uno de ellos presenta ventajas y desventajas particulares. Aquellos que consideran al tobillo se les conocen como pies articulados, y los que no presentan uniones móviles entre la extensión tibial (vástago) y el pie, se les conoce como no articulados [10].

Los pies articulados presentan mecanismos parecidos a los implementados en las rodillas artificiales. Los pies artificiales de una sola articulación funcionan con un perno (un solo eje) y dos amortiguadores que permiten mayor o menor movimiento, según se necesite. Los de varios ejes ayudan al usuario a utilizar su prótesis sobre superficies irregulares, ya que proveen de movimiento en diferentes direcciones. Por otro lado, los pies no articulados pueden ser rígidos o flexibles, con caucho en la base para un posible amortiguamiento. Algunos de los pies no articulados presentan dedos moldeados para mayor estética [10].

1.5 Tecnología actual

En años recientes se ha observado un avance significativo en las prótesis para extremidades, esto debido a que la tecnología actual se ha desarrollado a gran velocidad.

1.5.1 Materiales

Para la correcta selección del material que la prótesis requiere, se toman en cuenta múltiples consideraciones de diseño. Entre las cuales se encuentran el costo del material, su proceso de manufactura a llevarse a cabo, las cargas a las que el material estará siendo sujeto, el ambiente en el que se usará, la compatibilidad con el usuario y los aspectos estéticos [16].

Las primeras prótesis de las que se tiene registro utilizaban madera para su construcción, principalmente de sauce. Estaban constituidas por un socket que recibía al muñón y debajo se encontraba la extensión tibial junto con el pie protésico. Además, eran forradas con cuero pintado [39].

El uso de nuevos plásticos y otros materiales como fibra de carbono, han permitido a las prótesis de extremidades ser más ligeras y resistentes que sus antecesoras [4]. Sus ventajas son evidentes al reducir el peso de las prótesis y permitirle al usuario usar menos energía de la que éste necesitaba para hacer a su prótesis funcionar.

Los sockets protésicos generalmente son fabricados con fibra de carbono, que presenta características ligeras y alta resistencia comparada con otros materiales. Este material también permite el uso de *liners*, que son la interface entre el socket y el muñón. Los *liners* pueden estar hechos de silicona, copolímeros, poliuretano, y algunos otros de aceite mineral. Ayudan a la adherencia con el socket y reducen las lesiones óseas al servir como amortiguadores [34]; sin embargo, presentan problemas de transmisión de calor y humedad [36].

Por otro lado se encuentran los sockets de resina de poliéster, los cuales a diferencia de los hechos con fibra de carbono, son más pesados y menos flexibles; sin embargo, son más económicos. Por su estructura rígida, pueden también utilizar *liners* [35].

Otros materiales utilizados en las prótesis para extremidades actuales, son la silicona o el PVC (polivinilo clorado). Estos materiales han hecho posible mejorar la estética de las prótesis y lograr construirlas con una apariencia más cercana a las extremidades reales [4]. Estos materiales se usan por lo general en prótesis estéticas (pasivas), las cuales son fabricadas en serie. Al hacer la comparación entre el empleo de estos dos materiales, se reporta [12] que las prótesis estéticas de silicona tienden a ser más reales que las de PVC; llegan a ser más resistentes a manchas, son más flexibles, resisten mayores temperaturas, y son menos propensas a causar reacciones en el cuerpo del usuario. También se ha observado [12] que, debido a la fragilidad de las prótesis estéticas de PVC, éstas llegan a durar un promedio de seis años; sin embargo, las prótesis de silicona tienden a ser más caras.

El aluminio, es otro material que está presente en la mayoría de las prótesis. Algunos diseños han considerado el uso de diversas aleaciones como la 6061-T6, ya que resulta más práctico en la manufactura que otras aleaciones [16].

1.5.2 Manufactura

A lo largo de la historia, la construcción de las prótesis ha evolucionado significativamente. En México, por ejemplo, uno de los pioneros fue José Miguel Muñoz González quien se dedicó a principios del siglo XIX a desarrollar prótesis de piernas que fuesen más económicas y más prácticas que aquellas que se traían de España. En 1838 construyó una prótesis de caucho y madera para la pierna izquierda que había perdido en una batalla Santa Anna [30]. Sin embargo, con el paso del tiempo, los materiales y la construcción de las prótesis han avanzado. Actualmente, la construcción de los sockets protésicos para amputaciones transfemorales se puede realizar de manera artesanal o por medio de manufactura asistida por computadora (CAM).

Artesanalmente, el socket para la prótesis es hecho a la medida del muñón del usuario, y generalmente mediante un molde de yeso a mano. Posteriormente mediante el material adecuado como fibra de carbono se genera el socket para ser utilizado por el usuario de la prótesis.

Por otro lado, los sockets desarrollados por el método de CAD CAM, permiten que el proceso de fabricación sea más sencillo y más rápido en comparación al proceso artesanal. Existen muchas formas de realizar el Diseño Asistido por Computadora para posteriormente manufacturar el socket. Sistemas Ortopédicos del Noroeste es una empresa que realiza este tipo de manufactura, mediante la ayuda de la tecnología OSSUR-CAD PROSTHESIS CAM. El primer paso es realizar la medición del perímetro del muñón a diferentes alturas, posteriormente se realizan fotografías digitales en las cuatro caras del muñón y, finalmente, los datos procesados para tener el diseño de CAD. Este proceso genera un código para ser utilizado en una máquina CNC fresadora que genera el molde del socket [31].

El desarrollo de la tecnología de impresión en 3D ha permitido que la manufactura en diversos campos de la ingeniería se vean beneficiados. En el caso de la construcción de prótesis, se ha logrado construir prototipos en un tiempo menor comparado con las técnicas previamente expuestas en este trabajo. Además de la rapidez en la manufactura, se ha mejorado en la precisión y el diseño que con otras tecnologías no era posible realizar por la complejidad que presentaba. Otra ventaja que presentan las prótesis impresas en 3D es que pueden llegar a ser más económicas pues pueden ser diseñadas con la ausencia mecanismos complejos que implican un trabajo especializado aumentando el costo de fabricación.

El diseñador industrial William Root ha utilizado la tecnología de impresión 3D para fabricar prótesis de pierna para amputaciones a nivel transfemoral. Comienza escaneando el muñón del usuario de la prótesis, mediante una tecnología llama FitSocket, desarrollada en el laboratorio de Biomecatrónica del MIT (Massachusetts Institute of Technology), la cual crea un modelo virtual de la extremidad residual. Además, el escáner puede detectar propiedades biomecánicas en el tejido de la pierna, lo que permite realizar un diseño de socket más acorde a cada usuario y un mejor ajuste. Posteriormente se diseña la prótesis

para imprimirla en forma de una malla de titanio (como se muestra en la figura siguiente) mediante patrones que la hacen muy ligera [37].



Figura 1.7. Prótesis Exo impresa en 3D.

1.5.3 Neumática e hidráulica

Muchas de las prótesis de miembro inferior a nivel transfemoral utilizan dispositivos hidráulicos (con aceite de silicón) y neumáticos para su funcionamiento. Estos se encuentran ubicados generalmente en la rodilla protésica. En dichas prótesis, se controla el ritmo del caminar del usuario mediante pistones, y aquellos que resultan ser más eficientes son los de sistemas hidráulicos, ya que entregan mayor fuerza al momento de su implementación con respecto de aquellos que se manejan con neumática. La principal ventaja de este tipo de prótesis radica en que permiten al usuario realizar una marcha más normal en cuanto a su movimiento; sin embargo, llegan a ser más caros y poco ligeros [16], [17].

Otro uso de la neumática en sockets de amputación de extremidades inferiores se ve reflejado en el desarrollo de un sistema de suspensión para prótesis transtibial [38] conocido por sus siglas en inglés como APSS (Air Pneumatic Suspension System), creado por ingenieros de la Universidad de Malaya en Malasia. Este sistema utiliza una cámara de aire que realiza la tarea de suspensión dentro del socket protésico al mantener siempre la misma presión hacia el muñón. No se especifica el algoritmo de control; sin embargo, mediante un microcontrolador, un sensor de presión de estado sólido, una bomba y válvulas reguladoras de presión (como se muestra en la siguiente figura) se logra que sea fácil el ponerse y quitarse la prótesis. Por otro lado se adapta a los cambios diarios del volumen del muñón y mejora el ajuste.



Figura 1.8. Prótesis con sistema neumático APSS. a) cámara de aire, b) tarjeta de circuito de control, c) bomba, d) válvula, e) batería, f) sistema de operación, g) prótesis transtibial ensamblada [38].

1.5.4 Control

En la actualidad ya existen múltiples empresas que diseñan, manufacturan y distribuyen prótesis inteligentes con tecnología avanzada. Estas prótesis se integran de microprocesadores y sensores que detectan el movimiento de la prótesis, para poder controlar actuadores de todo tipo, desde hidráulicos y neumáticos hasta eléctricos. Por ejemplo, dependiendo del movimiento y de la velocidad detectada por los sensores en la prótesis, se ajusta el fluido en el cilindro neumático o hidráulico [16], [17].

Para poder controlar una prótesis, por lo general se suelen utilizar mecanismos como interruptores o comandos pre-programados, y en otros casos se emplean las mismas señales biológicas que genera el cuerpo humano (mediante electromiografía, mecanomiografía, encefalografía, etc.) [11]. En el caso de la implementación del control en algunas prótesis de extremidades inferiores, se tiende a utilizar preferentemente, microprocesadores. En estos, son programados determinados algoritmos o se implanta inteligencia artificial para tener una respuesta apropiada para las diversas situaciones del entorno en el que se use la prótesis [5]. Generalmente se tiende a utilizar control con lazo cerrado para tener una mejor respuesta de las prótesis que son automatizadas. En un control en lazo abierto, en el que el sistema funciona independientemente de las

condiciones del usuario o la amputación, pueden existir perturbaciones que afecten al sistema de la prótesis, por ello en las tecnologías actuales se prefiere evitar este último tipo de control.

En el sistema APSS, previamente mencionado, se desarrolló un control de un socket para una prótesis transtibial en el que se establece una presión fija para la cámara de aire, que debe permanecer constante a pesar de los cambios de volumen en el muñón del usuario. En el siguiente diagrama se muestra el funcionamiento general del sistema sin especificar a detalle el tipo de control que se aplicó [38].

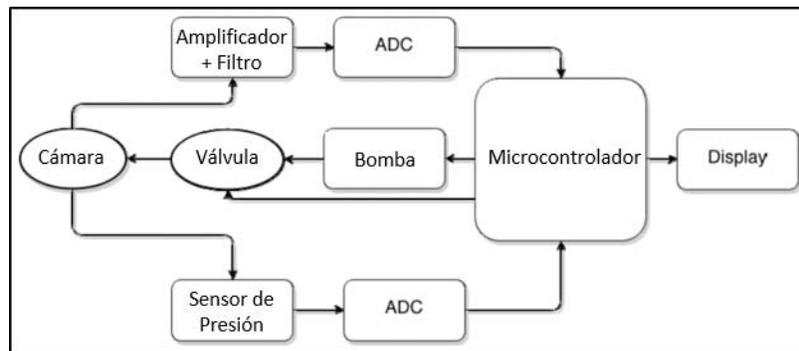


Figura 1.9. Funcionamiento del sistema APSS.

1.5.5 Electromiografía

Otro tipo de tecnología utilizada en las prótesis para amputación de una extremidad es la electromiografía como se muestra en algunos trabajos [5], [6], [9]. Se trata de una herramienta para el monitoreo de la actividad muscular que puede ser no invasiva (detectando la señal desde la piel) o invasiva (usando cables finos insertados en el músculo) [5]. En esta técnica se mide el voltaje generado durante la contracción de los músculos (en este caso de los músculos del miembro residual), lo cual entrega información acerca de su fisiología. El método consiste en rectificar la señal obtenida y posteriormente hacerla pasar por un filtro pasa-bajas. Según las investigaciones [9], la frecuencia de corte para el filtro no se ha definido del todo, y se han usado diferentes valores para diferentes actividades.

1.5.6 Mecanomiografía

Además de la electromiografía existe otra técnica de control utilizada en las prótesis para amputación de una extremidad, y ésta es la mecanomiografía. Se trata de una técnica en la que las vibraciones de la frecuencia en resonancia producidas al contraer los músculos, son grabadas como sonido. Con esta técnica es posible controlar una prótesis. Esta grabación es utilizada como la estimación de la fuerza necesaria para mover la prótesis [5].

1.6 Empresas dedicadas al desarrollo de prótesis de extremidades

Actualmente existen más de 100 diseños diferentes a nivel mundial en el mercado [15]. Algunas de las empresas más importantes, encargadas del diseño, la fabricación, y la

distribución de las prótesis de miembro inferior a nivel transfemoral se enlistan a continuación:

- Ottobock®
- Össur®
- Endolite®
- Intelligent Plus®
- Seattle®
- Power Knee®
- Proteor®
- Freedom Innovations®
- Medi®

Siendo unas de las más representativas a nivel mundial, en el desarrollo de prótesis, las empresas Otto Bock® y Össur®, las cuales inclusive han llegado a realizar acuerdos comerciales de manera simultánea [76].

Algunos ejemplos de prótesis para miembro inferior disponibles en el mercado y sus características, se describen a continuación:

1.6.1 C-Leg – Ottobock®

La compañía Ottobock®, fundada en Berlín, Alemania en 1919; tiene entre sus productos de prótesis de pierna, a la más representativa del mercado, y una de las de mayor tecnología implementada. Se trata de una prótesis que cuenta con un sistema hidráulico con el que mediante un pistón pivoteado a la extensión tibial y un cilindro pivoteado a la base del socket protésico, ofrece al usuario una mayor aproximación a la marcha natural.

El sistema hidráulico está acoplado a un sistema electrónico que contiene un microprocesador, el cual controla el movimiento de la articulación de la rodilla. Dicho sistema evalúa cada 0.02 segundos, mediante sensores, múltiples parámetros, entre ellos la carga, la longitud y la frecuencia de los pasos para permitirle a la rodilla artificial moverse de maneras particulares para diferentes contextos. Utiliza una batería Li-Ion con una capacidad de 40 a 45 horas. También incluye un control remoto que el usuario manipula para realizar algunos ajustes al sistema hidráulico. Puede aguantar un peso de hasta 125 kg, para usuarios de cualquier edad y una velocidad máxima de 5 km/h. Puede bajar escaleras, ya que esta prótesis reacciona y controla el arco de giro, frenando paulatinamente. Entre sus desventajas se encuentran los límites mecánicos, ya que no cuenta con inteligencia artificial o no es fácil de adaptarse a situaciones imprevistas. Otra desventaja es su precio inicial y el costo de mantenimiento los cuales son muy elevados. Para poder utilizar la prótesis es necesario acudir con un protesista certificado con el sistema C-Leg, y en México es posible contactarlos en Nuevo León, Ciudad de México, y Jalisco únicamente [16], [18].



Figura 1.10. Prótesis C-Leg.

1.6.2 Genium – Ottobock®

Además de la prótesis C-Leg, existe actualmente otra prótesis controlada mediante microprocesador que también es construida por Ottobock, pero que en rendimiento supera a la C-Leg. Genium tiene una tecnología llamada “Marcha Fisiológica Optimizada” que permite al usuario de la prótesis mejorar su movilidad en su vida cotidiana. Al momento de tocar el suelo en la marcha, la rodilla artificial se encuentra ligeramente flexionada, permitiendo que el pie haga contacto completo con el piso más rápidamente lo que permite que se use menos fuerza. Además, comparada con la C-Leg, los usuarios han reportado que es más confiable para subir y bajar escaleras, ya que se sienten más seguros [32]. Puede soportar hasta 150 kg y su peso es de 1320 g. El material principal del que está constituido es fibra de carbono, haciendo que la prótesis sea altamente resistente y ligera. Además cuenta con una batería Li-Ion que es capaz de durar hasta cinco días. Otra innovación es su carga inductiva, en la que el cargador de la prótesis es fijado magnéticamente a esta mientras se está cargando [33].



Figura 1.11. Prótesis Genium.

1.6.3 Aqualine – Ottobock®

Otra de las prótesis de Ottobock es la Aqualine (figura 1.12), la cual es capaz de ser usada en el agua, a diferencia de la C-Leg. Cuenta con una planta antideslizante para poder utilizarla en una alberca o en la regadera. Además cuenta con una rodilla artificial con un solo eje de rotación, un sistema hidráulico y un mecanismo que bloquea la rodilla cuando el usuario lo requiere [32].



Figura 1.12. Prótesis Aqualine.

1.6.4 Endolite®

Endolite pertenece a la compañía británica Blatchford Inc, y es una empresa dedicada a la manufactura de miembros artificiales y componentes protésicos. Produce pies artificiales, rodillas, tobillos y adaptadores, sockets y accesorios extras para las prótesis de pierna.

Entre sus principales productos se encuentran el pie élan® que contiene un microprocesador que controla la velocidad de la caminata mediante sensores que monitorean el ambiente y mediante un algoritmo se cambian las características del pie. Puede subir o bajar rampas y escaleras. Utiliza un tobillo hidráulico. En la figura 1.13 se muestra una imagen de este producto [20]. Soporta 124.7 kg, pesa 1.2 kg.



Figura 1.13. Pie élan de Endolite®.

Otro de sus productos es la rodilla policéntrica KX06 que además está acoplada a un pistón hidráulico. Permite una flexión de 160°, es resistente al agua, se puede utilizar para hacer bicicleta y puede llegar a soportar hasta 149.6 kg. En la figura 1.14, se muestra la imagen de la prótesis [20].



Figura 1.14. Rodilla protésica policéntrica KX06® de Endolite®.

1.6.5 Total Knee - Össur®

La compañía islandesa Össur®, fundada en 1971; proporciona tecnología en los campos de prótesis, órtesis y terapia. Los dispositivos que desarrolla son pies artificiales, rodillas, piernas, etc.

Uno de principales productos de esta compañía es la Total Knee®, una rodilla mecánica policéntrica con un control de oscilación hidráulico de 3 fases. Esta prótesis puede llegar a soportar hasta 125 kg, su peso es de 900 g y puede flexionarse hasta 160°. A diferencia de otras prótesis, esta le permite al usuario poder flexionar la rodilla al levantarse. En la figura 1.15 se observa dicha prótesis [21].



Figura 1.15. Prótesis de rodilla Total Knee® 2100 - Össur®.

1.6.6 Rheo Knee – Össur®

Otro producto de la empresa Össur es la rodilla protésica Rheo Knee, la cual contiene un microprocesador que permite adaptar al usuario y al ambiente con un buen balance y estabilidad. Contiene 5 sensores que detectan la marcha, incluyendo un giroscopio para lograr una buena respuesta en la marcha. Esta prótesis, hecha de aluminio, soporta un peso máximo de 136 kg, pesa 1.61 kg y puede llegar a flexionarse hasta 120°. Para su sistema eléctrico utiliza una batería Li-Ion de 1880 mAh que puede durar de 49 hasta 72 horas. En la figura 1.16 se muestra dicha prótesis de rodilla [21].



Figura 1.16. Prótesis de rodilla Rheo Knee® 2100 - Össur®.

1.6.7 Power Knee – Össur®

Esta prótesis de rodilla (mostrada en la figura 1.17) para amputación transfemoral, presenta una articulación accionada por un motor eléctrico. Permite al usuario amputado acelerar su rehabilitación, en comparación a otras prótesis. Mediante un software llamado PowerLogic Workbench, el protesista y los terapeutas tienen una retroalimentación durante el entrenamiento. Esta prótesis soporta un peso máximo de 165 kg, pesa 2.7 kg más la batería que pesa 0.49 kg. Puede llegar a flexionarse hasta 120° y está hecha de aluminio [21].



Figura 1.17. Prótesis de rodilla Power Knee® - Össur®.

1.6.8 Symbiotic Leg – Össur®

Esta prótesis de Össur® es biónica, la cual tiene una rodilla artificial con su microprocesador, una extensión tibial endoesquelética y un pie artificial con una articulación para el tobillo también con un microprocesador, para una flexión proactiva. Otra característica de esta pierna artificial, es que integra inteligencia artificial, la cual le permite a la prótesis aprender la manera en la que el usuario camina (su estilo) y a realizar algunos cambios de acuerdo a la velocidad y a la superficie sobre la que camina. Esta prótesis soporta un peso máximo de 125 kg, pesa 3 kg y puede llegar a flexionarse hasta 120°. Su batería puede llegar a durar de 16 a 24 horas continuas, dependiendo de la actividad [21].



Figura 1.18. Prótesis transfemoral Symbiotic Leg® - Össur®.

1.6.9 Plié Knee – Freedom Innovations®

La empresa estadounidense Freedom Innovations, se encarga de desarrollar prótesis para miembro inferior. Entre los productos desarrollados por esta empresa, como pies artificiales y rodillas protésicas, se encuentra The Plié® (figura 1.19). Esta prótesis se trata de una rodilla controlada por microprocesador con un sistema hidráulico, la cual es considerada la de mejor respuesta entre las prótesis controladas por microprocesador. Puede ser utilizada en actividades acuáticas (resistente al agua), es estable en rampas, etc. Esta prótesis se puede flexionar hasta 125°, soporta un máximo de 125 kg en actividades moderadas, pesa 1.235 kg y su batería puede llegar a durar más de 24 horas [22].



Figura 1.19. Prótesis transfemoral The Plié® - Freedom Innovations®.

1.7 Patentes

Además de las prótesis existentes en el mercado como las mencionadas previamente, también se puede encontrar proyectos de patentes publicadas. Es importante registrar la patente de una invención para proteger el trabajo realizado, así como para poder explotar económicamente con seguridad un proyecto tan importante como lo es una prótesis.

Existen múltiples patentes registradas alrededor del mundo, y a continuación se expondrán algunas de ellas relacionadas con los sockets protésicos a nivel transfemoral.

1.7.1 Socket cojín neumático con relleno poroso

Uno de los proyectos encontrados es una patente que presenta un socket con cojín neumático y un relleno poroso de 1967 [40]. Su objetivo es reducir los daños producidos en el muñón de los usuarios de prótesis, como llagas, ulceraciones e infecciones, causadas principalmente por los esfuerzos que existen en el área de contacto entre el socket y el miembro residual. Está compuesto por una resina porosa que sirve como amortiguador. Esta resina se encuentra dentro de una cámara de aire que sirve también como contacto directo con el muñón de la persona. No especifica exactamente cómo inflar la cámara de aire que se ubicará en el socket; sin embargo, sugiere el uso de una bomba de aire que podría ser llevada por el usuario en uno de sus bolsillos, en una bolsa de mano o en la parte hueca de la prótesis. En la siguiente figura se muestra un diagrama de cómo está constituida la cámara, y un corte del socket, ya con la cámara de aire en él.

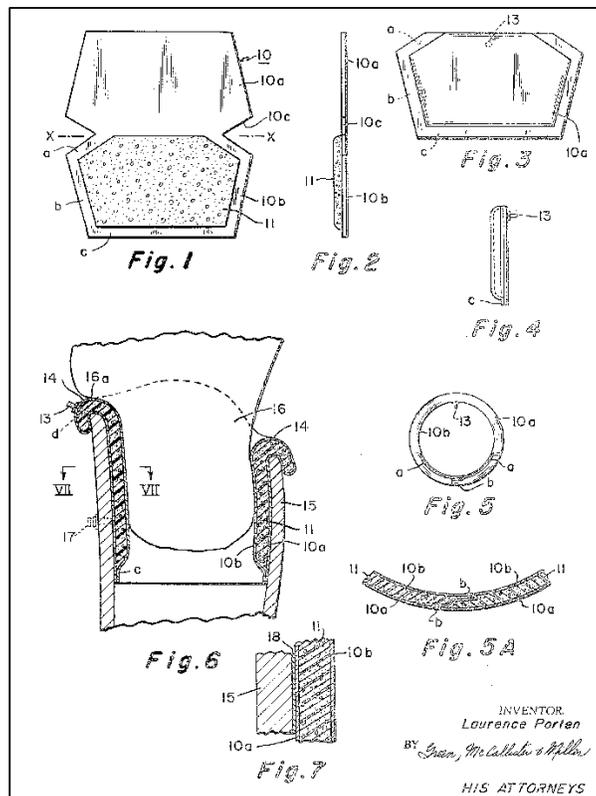


Figura 1.20. Socket con cojín neumático y relleno poroso.

1.7.2 Socket protésico ajustable

En la patente US 6991657 B1 del año 2006 se tiene un socket protésico para amputación transfemoral, que pretendía mejorar los sockets existentes. Propone un socket con alta estabilidad y confort así como un buen control de la movilidad y una transmisión del peso óptima [41]. Está compuesto por tres partes principales. Cada una de ellas se constituye de cuatro secciones, debido a que se trata de un socket cuadrilátero (pared anterior, pared posterior, pared medial y pared lateral). La primera parte es una copa, siendo la pieza más externa del socket. Otra de las piezas de este socket es la pared lateral, la cual es envuelta por la copa y, de igual forma que ésta última, incluye una extensión para sujetarse con mayor firmeza al muñón. La última parte del socket es una cubierta que hace contacto directo con el muñón, y es envuelta por la pared lateral. Se recomienda que tanto la copa como la pared lateral y la cubierta que hace contacto con el muñón sean construidas con un material polímero llamado poliolefina. Así que el socket ajustable se ve ensamblado con esas tres partes mediante aparatos de montaje como el velcro. En la siguiente imagen se muestra el esquema del socket ensamblado en una vista de explosión.

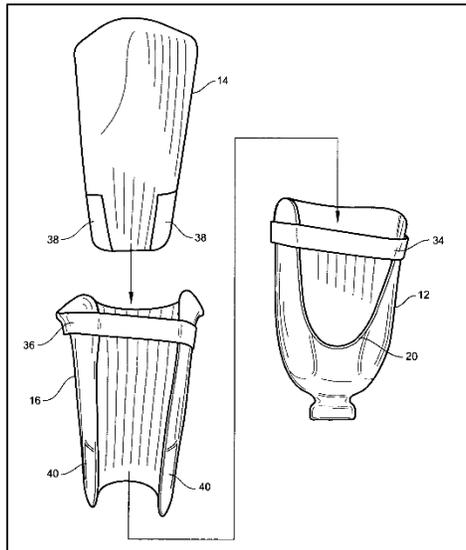


Figura 1.21. Socket protésico ajustable.

1.7.3 Socket protésico con depósito de vacío autónomo

La patente US 7427297 B2 del año 2008 presenta un diseño que permite al muñón estar firmemente sujeto al socket protésico para amputaciones transfemorales [42]. Para lograrlo, utiliza un depósito de vacío en forma de copa, ubicado entre dos paredes herméticas que evitan el paso de aire. Estas paredes son un socket interior rígido y un socket exterior de soporte de carga rígido, entre los cuales se halla el depósito de vacío. El material sugerido para las paredes de este socket es la fibra de carbono, pues presenta propiedades adecuadas de resistencia para implementar el vacío y, tiene muchas aplicaciones protésicas incluyendo a los socket protésicos.

Para lograr el vacío en el depósito del socket, se conecta una bomba al puerto de evacuación, que succiona el aire, hasta lograr una presión debajo de la atmosférica. Después de haber conseguido el vacío, la bomba es removida y el depósito se mantiene en ese estado gracias a una válvula check, que evita la salida de aire al ambiente. Esto permite que el miembro residual con su liner, quede sujeto firmemente al socket. En la siguiente imagen se observa la constitución de dicho socket [42].

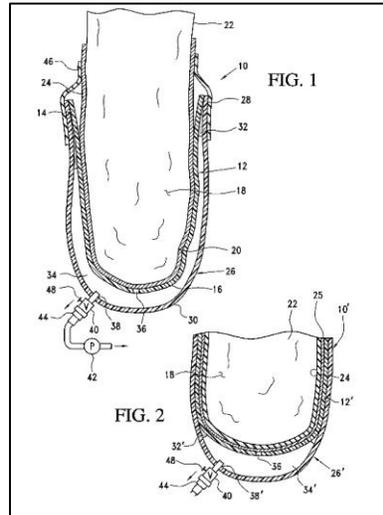


Figura 1.22. Socket protésico con depósito de vacío.

1.7.4 Sistema de ajuste dinámico de geometría variable para usar con un dispositivo corporal

En esta patente (US 6585774 B2) del 2003, el socket para amputación transtibial o transfemoral usa un líquido no compresible para asegurar el ajuste con el miembro residual, ante los cambios en su geometría. Utiliza las cámaras donde se concentra el líquido como bombas y una unidad de control que regula los cambios en el volumen al caminar. El líquido que utiliza es agua con anticongelante preferentemente [43].

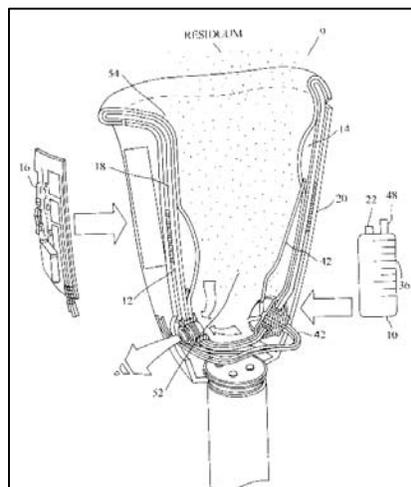


Figura 1.23. Socket con sistema dinámico de geometría.

La información expuesta en este primer capítulo permite llevar a cabo un análisis de los aspectos fundamentales para el diseño de una prótesis para amputación de miembro inferior a nivel transfemoral. Se partió de la definición de la discapacidad provisto por la OMS para tener una perspectiva general del tema a ser tratado.

Se concluye que la amputación a nivel transfemoral es el tipo de amputación más recurrente en México y que sus tres causas más comunes son la enfermedad, la edad avanzada y los accidentes. Por otro lado se hizo énfasis en el proceso que debe llevar una persona que se le ha realizado una amputación. Este comprende la correcta obtención de la prótesis, la atención psicológica y el entrenamiento para la adaptación con la prótesis.

A partir de las consideraciones anteriores se debió definir lo que es una prótesis para amputación de miembro inferior a nivel transfemoral, sus funciones principales y, el tipo de prótesis dependiendo de la clasificación. Además, para poder entender el proceso de diseño se vio necesario dar a conocer las partes y funciones de una prótesis, siendo la más importante el socket protésico.

En la tecnología actual se observó que los materiales más comunes en una prótesis son la fibra de carbono, el aluminio, el titanio y el PVC para la estructura. Además, para los demás componentes se utiliza la silicona, la resina de poliéster, los copolímeros, el poliuretano, el aceite mineral, etc. En cuanto a la manufactura se encontraron tres formas más utilizadas para el desarrollo de sockets. La artesanal es una de ellas, mediante el uso de yeso. El otro tipo de manufactura es mediante CAD CAM y CNC. Y la forma de manufactura más reciente encontrada fue la de impresión 3D. El control de las prótesis se encontró que es llevado a cabo por microprocesadores, sensores y actuadores. Las tecnologías más utilizadas en las prótesis para el control son las de electromiografía, mecanomiografía y encefalografía.

Las empresas más importantes en el desarrollo de prótesis a nivel mundial encontradas fueron Ottobock y Össur. Y entre las prótesis más avanzadas se encuentran la C-Leg, la Genium, Aqualine, Power Knee y Pilé Knee, entre otras.

También se investigaron las patentes de sockets con relación a la prótesis a ser instrumentada en este trabajo de tesis. Entre ellas se encontró una con un socket neumático de resina porosa para amortiguamiento. Otra tenía un socket ajustable de tres partes de poliolefina. Otra patente era de un socket con un depósito de vacío dentro de dos sockets rígidos de fibra de carbono y con una válvula check. La última patente descrita era acerca de una socket dinámico hidráulico para el ajuste de los cambios de geometría del muñón.

Esta información será de gran ayuda en los próximos capítulos para el desarrollo de la instrumentación del socket protésico a nivel transfemoral. La tendencia indica que el socket protésico debe contar con materiales ligeros como la fibra de carbono, el titanio, o algún material que presente una manufactura sencilla y de bajo costo.

Además, en concordancia con las patentes encontradas, el socket autoajustable cuenta con sistema de control que será modificado de acuerdo con las necesidades de la instrumentación.

Otra tendencia importante, además de los materiales, es la tecnología implementada en cuanto al control electrónico. Actualmente, las prótesis de pierna más avanzadas cuentan con microprocesadores que le permiten a la prótesis realizar tareas específicas para mejorar la marcha, la bipedestación, etc. En el presente trabajo se podrá contar con la instrumentación del socket, la cual permitirá al usuario tener una interacción más estrecha con su prótesis.

Capítulo 2. DISEÑO CONCEPTUAL DE LA INSTRUMENTACIÓN

El socket protésico es una parte fundamental de una prótesis de pierna. Éste representa, en la mayoría de los casos, el primer contacto físico que un usuario tiene con la prótesis, por lo que es importante contar con un diseño que cumpla con las necesidades primordiales.

El diseño del socket a instrumentar ha evolucionado gracias a los diversos trabajos previos que se han realizado.

En la tesis para maestría “Diseño de un socket autoajustable para prótesis de miembro inferior” [44], se realiza la primera propuesta de un sistema de autoajuste para el socket protésico. Ahí se plantean las partes principales que debe de llevar el socket para lograr que éste sea autoajustable. Se hace énfasis en la utilización de un sensor que detecte la presión en los sujetadores inflables. Éstos últimos son la parte fundamental para el sistema de ajuste, pues permiten eliminar los problemas en la adaptación al tamaño del socket por parte del usuario al existir variación en la masa muscular del muñón.

Además del sensor, en dicha tesis también se propone el uso de válvulas que regulen la entrada y salida del aire hacia y desde los sujetadores. Para el accionamiento del flujo del aire se considera utilizar una bomba tanto manual como automática [44]. El disponer de dos bombas en el socket, incrementa el tamaño y el peso en el socket. Con la intención de contar con un socket con un sistema autoajustable útil para el usuario, se analizará la idea de descartar la bomba manual para hacer al socket totalmente automático. El diseño y la instrumentación a realizarse en el presente trabajo, contará con una interfaz simple de manejar para controlar la presión del socket. Por otro lado, se buscará que el control del sistema neumático a ser diseñado, presente la característica de tener una respuesta rápida.

Posteriormente se propone el manejo de un microcontrolador que active y regule las señales del sensor de presión, además de controlar el funcionamiento de las válvulas. Y finalmente, se recomienda utilizar una batería recargable, ligera y que le sea útil al usuario cuando éste se encuentre realizando sus actividades diarias [44].

En la tesis profesional “Diseño de un sistema de autoajuste para un socket protésico a nivel transfemoral” [23] se realiza un avance significativo en el que se hace la primera implementación del socket autoajustable propuesto en trabajos previos [44]. Este trabajo divide en tres partes al sistema de autoajuste: ajuste automático, suministro de aire y ajuste manual. Se logra construir un modelo funcional del socket con el sistema autoajustable, sin llegar al diseño de detalle. Además, desarrolla un algoritmo para el control automático de la presión en los sujetadores. Este algoritmo consta de un control proporcional e integral, que posteriormente es programado en un microcontrolador ATMEGA328P. Este diseño utiliza servoválvulas para la entrada y salida de aire de los sujetadores, una bomba neumática para el sistema de autoajuste y, una válvula para poder hacer uso del sistema manual de entrada de aire. Dentro de las principales desventajas que se presentan en este

sistema, están el tamaño y el número de componentes utilizados, así como el peso excesivo que el socket podría llegar a presentar en un diseño de detalle.

Para probar el funcionamiento óptimo de la prótesis, además de contar con un prototipo que cuente con todas las partes de ésta (incluyendo al socket), fue necesario proponer una metodología a seguir, lo que se realizó en el trabajo de tesis “Metodología de pruebas estáticas y dinámicas para un socket protésico de miembro inferior a nivel transfemoral” [46]. Dicho trabajo realizó un análisis de elemento finito de los componentes del socket ajustable, diseñado en trabajos previos [2] y [44] con el fin de seleccionar los materiales adecuados en cada uno de ellos. Además, se elaboró los formatos de consentimiento informado, para avalar que la persona que se prestaría a realizar los análisis dinámicos del socket, presentase condiciones adecuadas en su muñón, conociese los riesgos, y estuviese de acuerdo en participar en dichas pruebas. Y finalmente propuso una evaluación dinámica para la prótesis completa. Sin embargo, en el análisis de elemento finito, para la elección de los materiales, no se tomó en cuenta el sistema autoajustable, que cuenta con más elementos que los existentes en el socket ajustable.

Para el desarrollo de la interfaz, en el trabajo “Diseño e implementación de una aplicación móvil para monitoreo de un socket autoajustable” [45] se generó una aplicación en Android para teléfonos inteligentes. Además, se implementó un circuito electrónico para medir la fuerza ejercida por el muñón sobre el socket (sensor FSR 406) y para medir temperatura y humedad (sensor DHT11) dentro del socket. Los datos de la temperatura, la humedad y la fuerza ejercida, son enviados a una tarjeta Arduino UNO, y posteriormente son almacenados en una memoria microSD. Posteriormente, en la aplicación, se logra observar los datos en una gráfica. Los datos son enviados mediante un módulo de Bluetooth HC-06, desde la tarjeta Arduino UNO hasta el dispositivo móvil. La principal desventaja de este diseño, es que no se considera realmente el sistema autoajustable. Por lo que en el presente trabajo se realizará la manipulación y monitoreo de dicho sistema, además de tomar en cuenta los datos de temperatura y humedad.

La instrumentación del socket en este trabajo pretende, además de unificar los trabajos realizados previamente, contar con un proyecto más sólido y concreto que permita acercarse a un prototipo real del socket. Esto generaría un estímulo y un incentivo para poder continuar trabajando paralelamente con los diseños complementarios de las partes de la prótesis. Y así comenzar a realizar las pruebas en usuarios y trabajar en un diseño final.

La instrumentación del socket protésico se dividió en los siguientes sistemas:

- **Sistema electrónico.** Este sistema está constituido por los elementos físicos necesarios para controlar y monitorear la instrumentación del socket. Dentro de dichos elementos se encuentra el microcontrolador, el módulo de Bluetooth para la transferencia de datos, el sensor de temperatura y humedad, el sensor de presión, el botón de inflado, y el servomotor para controlar la servoválvula del sistema neumático.

- **Sistema de programación.** Este sistema involucra el desarrollo de la interfaz en Android, desde la selección de la manera en la que se programará, hasta la manera en la que funcionará de la aplicación.
- **Sistema neumático.** Este sistema está integrado por los elementos que intervienen en la entrada y salida de aire hacia y desde los sujetadores del socket. Por lo tanto, consta de la servoválvula que permite la entrada, la salida, o el bloqueo del flujo de aire. También incluye la bomba neumática que impulsa el aire, las válvulas de emergencia, los conductos por los que pasa el aire y, los sujetadores que funcionan como capacitores neumáticos.

Por otro lado, la instrumentación del socket protésico presenta las siguientes funciones:

- Permite conocer el comportamiento de las variables físicas. Esto significa poder obtener los datos de la temperatura y la humedad en el socket en un intervalo de tiempo definido, almacenarlos y analizarlos a través del uso de gráficas u otros medios.
- Monitorear el funcionamiento del control de presión. En el sistema de autoajuste del socket es necesario conocer el rendimiento real del control de la presión. Para saber si realmente se están cumpliendo los objetivos de control, se puede realizar un monitoreo de los cambios en los valores de la presión y cómo es que el sistema llega al valor de referencia. Es por ello que, a través de una interfaz, se mostrará dicho comportamiento con el fin de comprobar el correcto funcionamiento del control de presión en socket protésico.
- Manipular la referencia en el control de presión. Además de monitorear el comportamiento del control de presión en el sistema de autoajuste del socket ante los cambios de presión, se plantea en este trabajo la posibilidad de actualizar la referencia de la presión existente en los sujetadores del socket protésico, a través de una interfaz.

Además de las funciones de la instrumentación del socket, planteadas previamente, se tomarán en consideración aquellas propuestas en trabajos previos referentes a este socket protésico [23]. Tales consideraciones son: lograr un ajuste seguro y cómodo al muñón del usuario del socket, dar seguridad en la bipedestación y en la marcha, crear una interacción sencilla entre el usuario y la prótesis y, ajustar automáticamente al muñón del usuario ante los cambios geométricos y de volumen presentes.

Dentro de los principales beneficios que implica instrumentar al socket protésico, se encuentran los siguientes:

- Poder controlar mediante un dispositivo móvil, el funcionamiento del socket.

- Capacidad para conocer los cambios en las variables físicas del socket como la temperatura, la humedad y la presión.
- Simplicidad en la interfaz de usuario para permitir un uso fácil de ésta.
- Generar herramientas para tener una certeza de que el control de presión en el socket se encuentra funcionando adecuadamente.
- Además de la instrumentación, mediante la selección adecuada de los elementos en los sistemas electrónico y neumático, se logrará tener un socket compacto e integral, reduciendo el peso y el costo de la prótesis.
- Tener un socket que cuente con los recursos necesarios para ser integrado en una prótesis de pierna a nivel transfemoral.
- Contar con un socket con un sistema de autoajuste confiable para el usuario, de respuesta rápida ante los cambios geométricos y de volumen en el muñón.

2.1 Planteamiento de Requerimientos

Debido a que no se han desarrollado los sistemas para contar con una prótesis de pierna a nivel transfemoral completamente diseñada, que incluya la integración del socket (del presente trabajo), la rodilla protésica, la extensión tibial y el pie protésico, no es posible comprobar el funcionamiento de dicha prótesis (al estar siendo utilizada por una persona). Sin embargo, sí es posible desarrollar los medios que permitan saber si el socket puede ser útil a un usuario de prótesis transfemoral. Es por ello que lo que se pretende en este trabajo es completar el diseño del socket autoajutable, para posteriormente integrarlo en un ensamble final de la prótesis, desde una perspectiva modular. Con el diseño de la prótesis final, sería posible llevar a cabo las pruebas necesarias con un usuario, haciendo uso de las herramientas que se proponen en el presente trabajo con la instrumentación del socket.

Una de las etapas de la instrumentación del socket protésico, es permitirle al usuario observar cómo han estado cambiando la temperatura y la humedad dentro del socket. Estos datos le son útiles a los médicos especialistas como los traumatólogos y los dermatólogos. El contar con una herramienta para conocer estos datos, les ayuda a predecir diagnósticos probables, y tratamientos para posibles problemas. En caso de detectar irregularidades, el especialista decidirá si es necesario dar al usuario de la prótesis un tratamiento especial o recomendarle llevar un uso más adecuado con la prótesis. En el caso del diseñador, el poder conocer los datos de temperatura y humedad en el ambiente interno del socket, le permitirá tomar la decisión de ajustar el diseño del socket con el fin de mejorar el comportamiento de dichas variables. Posiblemente se realicen cambios en el material que hace contacto directo con el muñón, o se mejore la ventilación existente en el socket. Estas decisiones le competen únicamente al diseñador, con base al análisis del comportamiento de la temperatura y la humedad, así como a las recomendaciones del usuario y del médico especialista.

De acuerdo con médicos del Instituto Mexicano del Seguro Social, tanto en los músculos como en los huesos se maneja una temperatura corporal normal de 36 °C. Además, un incremento en la temperatura puede significar la presencia de algún tipo de tumoración o

alguna otra anomalía. Lo mismo ocurre al bajar la temperatura, lo que podría indicar una mala circulación en el muñón. La humedad va de la mano con la temperatura, para que el músculo tenga un funcionamiento adecuado. En las amputaciones de personas diabéticas, a diferencia de las amputaciones debidas a traumas, es más peligrosa la existencia de una variación en la temperatura y la humedad, pues puede existir la posibilidad de que se gangrene la extremidad residual.

Los mismos médicos especialistas del IMSS aseguran que en el aumento en la temperatura (y la humedad) en el muñón propicia un ambiente en el que las bacterias se pueden generar y producir abscesos de pus. La temperatura normal que debe estar presente en el cuerpo humano es de 36 a 37 °C, al superar este rango, se le conoce como febrícula, hasta 37.9 °C, y arriba de 38 °C ya es fiebre. Si el muñón del usuario de la prótesis presenta un proceso de hipertermia, se puede pensar en la existencia de un proceso infeccioso. Por otro lado, cuando la temperatura baja de los 36 °C (hipotermia), es probable que se trate de una deficiencia en la circulación, particularmente una insuficiencia arterial (isquemia). Esto es debido a que las arterias llevan al tejido sangre oxigenada, y al caer la temperatura se puede suponer que la arteria se ha tapado. También se puede observar en este caso que el tejido cambia a un color morado o azul y el pulso se encuentra disminuido o no se siente. La presencia de isquemia puede conducir a una necrosis o muerte del tejido. Esta situación sería contraproducente para una persona con amputación ya que podría conllevar a no ser candidata a ser usuario de una prótesis. En algunos casos, puede existir un proceso de revascularización, en el que se crean nuevos vasos sanguíneos, y así evitar la necrosis.

En cuanto a la humedad, los médicos del IMSS explican que un exceso en la humedad puede propiciar la generación de hongos. En contraste, la baja humedad puede producir lesiones y erosiones en los tejidos, lo que puede condicionar la existencia de heridas, con las cuales una persona no puede utilizar una prótesis.

Otra etapa en la instrumentación del socket protésico es medir y manipular la presión de aire en los sujetadores del socket, por medio de un dispositivo móvil. La ventaja de utilizar un dispositivo móvil como interfaz es que en México existe una gran cantidad de éstos, y su crecimiento se mantiene a la alza. En el segundo trimestre de 2015 se logró contabilizar 62.5 millones de dispositivos en México, lo que representó un crecimiento de 41.4% respecto al mismo periodo del año anterior, según cifras de la consultora The Competitive Intelligence Unit (CIU) [48]. La forma de medir la presión del aire en los sujetadores ya se contempló en los trabajos previamente realizados [23], por lo que en este trabajo se analizará la manera en la que se pretende disponer al sensor de presión, y su nueva programación en el microcontrolador propuesto.

Finalmente, se realizará la integración de los sistemas, por lo que se generará un diseño asistido por computadora del socket protésico que incluya al circuito electrónico, la batería, los sensores, la servoválvula, etc. Esto tomando en consideración las especificaciones sugeridas en trabajos previos [46], donde se plantea el espacio interno mínimo requerido

para que la malla elástica del socket se pueda deformar y lograr contener al muñón del usuario.

2.1.1 Requerimientos

Unificación de sistemas. Lograr medir temperatura, humedad y presión en el socket e implementar el control proporcional integral, todo en un mismo dispositivo.

Manejo sencillo. La instrumentación del socket protésico permitirá al usuario de la prótesis controlar la referencia del autoajuste por medio de una interfaz sencilla y concreta, que además contará con las herramientas necesarias para el monitoreo de las variables físicas en el socket.

Mantenimiento de usuario sencillo. Se buscará que el socket sea sencillo de manipular, en caso de que el usuario requiera dar limpieza al socket, éste no tendrá problemas para acceder a cada parte.

Mantenimiento interno accesible. Además la ubicación de las partes que involucran a la instrumentación, estarán ubicadas de tal forma que su mantenimiento será fácil. Si se necesitase un cambio de pieza o revisar los elementos electrónicos, como el módulo de Bluetooth, la batería, o el sensor de presión, estos estarán dispuestos de modo que el usuario o el experto en prótesis puedan acceder a ellos rápida y cómodamente.

Visualización de datos de variables simple. El usuario, gracias al desarrollo de herramientas en la aplicación de un dispositivo móvil, podrá observar cómo han cambiado la temperatura, humedad y presión en el socket protésico.

Comprobación de funcionamiento del control. Mediante la interfaz a diseñarse en un dispositivo móvil, se contará con una herramienta poderosa para corroborar que el control de presión se está llevando a cabo correctamente. Por medio de gráficas se podrá saber que los requerimientos del sistema de control se están cumpliendo.

Compactar al socket. El presente trabajo, dentro de la instrumentación del socket protésico, se busca contar con un sistema que tenga los componentes adecuados y mínimos necesarios para que funcione de manera eficiente. Es por ello que se tendrá especial atención a reducir el tamaño de los elementos físicos que componen al socket. Esto además, conllevará a tener un socket más ligero, reduciendo así la masa del sistema.

2.2 Presión, humedad y temperatura

La instrumentación del socket protésico involucra la medición de 3 variables físicas importantes, como lo son la presión, la humedad y la temperatura. El conocer estos datos, sirve como herramienta para saber si el diseño del socket es el adecuado para ser utilizado en un muñón de una persona con amputación transfemoral.

2.2.1 Presión

La presión es la primera variable a considerar en el diseño conceptual. El socket protésico autoajutable es un sistema neumático, es decir, utiliza la energía del aire para lograr retener al muñón del usuario y ajustarse a los cambios volumétricos y geométricos de éste.

Es posible definir a la presión como una fuerza normal que ejerce un fluido por unidad de área [49]. En el Sistema Internacional de Unidades, es una unidad derivada, cuyo nombre es el pascal, y al ser una fuerza por unidad de área, se define de la siguiente manera:

$$1 Pa = 1 N/m^2$$

Además del pascal, existe otra unidad de uso común para denotar a la presión llamada milímetro de mercurio. Esta unidad fue definida a partir de lo que se conoce como atmósfera estándar, la cual es una presión ejercida por una columna de mercurio de 760 mm de altura a 0 °C bajo una gravedad de 9.807 m/s² [49]. La conversión de pascales a milímetros de mercurio es la siguiente:

$$1 mm Hg = 133.322 Pa$$

Existen diferentes términos para referirse a la presión, dependiendo al tipo del que se esté refiriendo. La presión absoluta es una magnitud física que depende de la posición. Es la suma de la presión manométrica y la presión atmosférica local. La presión atmosférica depende de la posición, a mayor altitud, existe menor presión atmosférica y viceversa. Así que se tiene la siguiente expresión para conocer la presión absoluta [49]:

$$P_{abs} = P_{atm} + P_{man}$$

La presión en el socket protésico ha sido discutida ampliamente en trabajos realizados previamente. Lo que se pretende medir y controlar exactamente, es la presión de aire existente dentro de los sujetadores del socket. Queda fuera del alcance de este trabajo medir la presión sanguínea del muñón del usuario de la prótesis.

2.2.2 Humedad y temperatura

Otra propiedad que se pretende monitorear es la humedad existente en el socket. El aire es una mezcla de gases, constituido en su mayoría por nitrógeno y oxígeno. Adicionalmente, el aire puede contener ciertas moléculas de agua pero en pequeño porcentaje, generalmente menos del tres por ciento. Debido a esto, se puede afirmar que el aire es una mezcla de aire seco con un poco de vapor de agua. A este vapor de agua en el aire se le conoce como humedad. Así que, la presión atmosférica es la suma de la presión de aire seco con la presión de vapor [50]:

$$P_{atm} = P_a + P_{vapor}$$

La humedad relativa es una relación entre la cantidad real existente en el aire de vapor de agua (a una temperatura determinada), con respecto a la máxima cantidad que el aire puede tener en esa temperatura.

Cuando el aire alcanza la máxima cantidad de vapor de agua (humedad) que es capaz de contener, a este punto se le conoce como aire saturado. Si se le agrega más humedad, el vapor de agua se condensará. Esta cantidad que el aire puede contener depende de la temperatura. Mientras más temperatura tenga el aire, más habilidad tendrá para contener humedad. La comodidad dependerá entonces, de la cantidad de humedad que tenga el aire m_v , en relación a la máxima cantidad de vapor de agua que el aire pueda contener m_g . A esta relación se le conoce como humedad relativa [51]:

$$\phi = \frac{m_v}{m_g}$$

La humedad relativa puede variar de cero (aire seco) a uno (aire saturado) o porcentualmente, de cero a 100 % [51].

Debido a que la humedad en el socket no se puede controlar de manera automática, el objetivo de monitorear este dato es permitirle al diseñador de la prótesis realizar los ajustes necesarios para que el usuario tenga confort al usarla. Cuando ya se tenga la posibilidad de realizar las pruebas dinámicas y estáticas propuestas previamente [46], se podrá ajustar el diseño del socket para lograr tener un rango de valores de humedad cercano a la comodidad del usuario. En algunos textos [50] se plantea que la humedad relativa adecuada para el confort térmico del ser humano oscila entre el 40 y el 60 por ciento.

El último dato a monitorear es la temperatura en el socket. Este dato es importante ya que puede prevenir problemas en el muñón a causa de alguna anomalía en el diseño del socket o debido a cuestiones físicas del mismo usuario. Si el usuario ha presentado niveles de temperatura fuera de lo normal, el encargado de su salud determinará las recomendaciones pertinentes para el uso de la prótesis. Además, contar con este dato le permitirá de igual manera, al diseñador, poder saber si es necesario algún ajuste o modificación en el diseño del socket, o algún cambio en el material que entra en contacto directamente con el muñón. La temperatura es un dato que está ligado directamente con la humedad relativa del socket. Esto se debe a que, como se mencionó anteriormente, el aire es capaz de contener mayor humedad si la temperatura aumenta.

2.3 Selección de elementos para el sistema electrónico

El sistema electrónico es la base de la instrumentación del socket protésico a nivel transfemoral. La correcta selección de los componentes de este sistema, permitirá tener un diseño de configuración eficiente y práctico.

Las secciones fundamentales de este sistema son las siguientes:

- Microcontrolador
- Módulo de Bluetooth
- Sensores de temperatura, humedad y presión
- Servomotor
- Suministro de energía
- Componentes adicionales

2.3.1 Microcontrolador

Este dispositivo es el cerebro de la instrumentación del socket. Un microcontrolador es una computadora en un chip único hecha especialmente para aplicaciones de control embebido. Tiene módulos integrados que sirven de herramientas para poder realizar las tareas programadas en él. Un microcontrolador puede tener convertidores A/D, convertidores D/A, puertos de entrada-salida, etc. Internamente también pueden llegar a contar con manejo de interrupciones y timers para mejorar la precisión de los procesos a controlar [52].

Los microcontroladores pueden ser programados en diversos lenguajes, tanto de bajo nivel, como de alto nivel. El programarlos en lenguaje ensamblador permite desarrollar programas más eficientes, eliminando código basura en comparación de programarlos con un lenguaje de alto nivel. Sin embargo, a pesar de lo eficiente que un programa termina siendo en cuanto a espacio y detección de errores, programar en lenguaje ensamblador puede ser una tarea muy lenta. Otra desventaja es que éste lenguaje será específico para cada tipo de microcontrolador. Por esto es que una manera más sencilla y rápida de programar un microcontrolador es hacerlo con un lenguaje de alto nivel, como C, C++, o BASIC. Estos lenguajes permiten reducir tiempo de programación, y en caso de utilizar un IDE (Ambiente de Desarrollo Integrado en español) adecuado se pueden detectar los errores oportunamente.

En la tesis “Diseño de un sistema de autoajuste para un socket protésico a nivel transfemoral” [23], se utilizó un microcontrolador ATMEGA328P desarrollado por el fabricante estadounidense Atmel, con las siguientes características:

- Velocidad de reloj: 16 MHz
- Alimentación: 5 V
- 14 entradas digitales
- 6 salidas de PWM (8 bits)
- 6 entradas analógicas (10 bits)
- EEPROM de 1 KB
- Memoria Flash de 32 KB
- SRAM de 2 KB
- Encapsulado DIP de 28 pines

Para este trabajo, se seleccionará un microcontrolador que cuente con las entradas y salidas necesarias para los sensores, y los actuadores. Además se buscará que se cuente con convertidor analógico-digital, módulo PWM (modulación de ancho de pulso), módulo USART (transmisor receptor asíncrono, síncrono universal), posibilidad de interrupciones, timers, memoria EEPROM, y una velocidad del CPU adecuada.

2.3.2 Módulo de Bluetooth

Para la transmisión y recepción de datos entre el socket y el dispositivo móvil, es necesario definir cuál será el módulo de comunicación inalámbrica adecuado.

Debido a que actualmente la gran mayoría de los dispositivos móviles actuales (smartphones, tablets, etc.) cuentan con un módulo de Bluetooth, en el presente trabajo se utilizará esta tecnología para la transmisión de datos.

Un protocolo de comunicación es un conjunto de reglas y normas dentro de una transmisión de datos entre dos entidades, ya sea a través de software o hardware. Dichas reglas definen la sintaxis y la sincronización para que dichas entidades puedan comunicarse [53].

Bluetooth es un protocolo de comunicación inalámbrico que permite la transmisión de datos como voz, música, video, y otra información entre los dispositivos conectados. Fue creado por la empresa sueca Ericsson en 1994 con el fin de sustituir los cables del estándar RS-232. El nombre del protocolo proviene del rey danés Harald Blåtand que traducido al inglés es Harold Bluetooth [54].

Algunas características importantes del Bluetooth son que es de bajo costo y de bajo consumo [55] a diferencia de otras tecnologías inalámbricas como Wi-Fi o ZigBee. Bluetooth trabaja en la transmisión de datos a cortas distancias (hasta 100 metros) dentro de una PAN (red de área personal) [54].

Se trata de un protocolo de comunicación inalámbrico que opera en la banda ISM (banda Industrial, Científica y Médica de radiofrecuencia electromagnética que no necesita licencia para ser utilizada) de 2.4 hasta 2.485 GHz. Utiliza el espectro ensanchado por salto de frecuencia (FHSS) que ayuda a reducir el ruido principalmente y su comunicación es de tipo full-duplex [55].

Al proceso de conectar dos dispositivos por medio de Bluetooth, se le conoce como emparejamiento. A las redes entre dispositivos conectados a través de este protocolo (hasta ocho dispositivos) se le conoce como piconets, y cada dispositivo puede además, pertenecer a varias piconets [55].

Su rango para su utilización depende la clase y de la manera en la que los encargados de la manufactura hayan decidido implementar el Bluetooth. Por lo que existen las siguientes clases dependiendo de su potencia y su alcance [55], [56]:

Clase	Potencia máxima mW	Alcance aproximado
Clase 1	100	100 metros
Clase 2	2.5	10 metros
Clase 3	1	1 metro

Tabla 2.1. Clases de Bluetooth.

En trabajo “Diseño e implementación de una aplicación móvil para monitoreo de un socket autoajustable” [45], se utilizó el módulo de Bluetooth HC-06 de origen chino. Se comunica mediante un puerto serial, y necesita ser alimentado con 3.3 V. Es de clase 2 y de acuerdo a su hoja de especificaciones tiene un alcance de solo 10 metros.

En el trabajo para la instrumentación del socket, se planteará un módulo que cuente con los recursos suficientes para la transmisión y recepción de datos de manera eficiente y, que sea de bajo consumo energético.

2.3.3 Sensor de temperatura y humedad

Para monitorear los datos de temperatura y humedad es necesario seleccionar el sensor adecuado. Dicho sensor debe contar con las características necesarias para poder implementarlo en la instrumentación del socket.

Un sensor es un dispositivo que puede detectar magnitudes físicas del entorno y transformarlas en algún tipo de señal de salida.

Parámetros de un sensor [57], [58]

Rango/Campo de medida. Son el valor máximo y mínimo de la magnitud física que el sensor puede medir. Los valores de entrada fuera de este rango son ininteligibles para el sensor y pueden causarle un daño irreversible. Se expresa con sus valores extremos.

Error absoluto. Es la diferencia entre el valor leído por el sensor y el valor verdadero de la magnitud física.

$$E_{absoluto} = v_{leido} - v_{verdadero}$$

Error relativo. Este error es la relación entre el error absoluto y el valor verdadero de la magnitud física.

$$E_{relativo} = \frac{E_{absoluto}}{v_{verdadero}}$$

Exactitud. Es una cualidad del sensor para dar medidas más próximas al valor verdadero de la magnitud física. Para poder evaluar la exactitud de un sensor es necesario compararlo con una medida estándar, o se debe comparar con algún otro sensor del que se conozca su exactitud. Se puede expresar de la siguiente forma:

$$Exacitud = \frac{E_{absoluto}}{Rango}$$

Precisión. Se lo conoce a la cualidad del sensor que implica concordancia con lecturas sucesivas, bajo un mismo estímulo de entrada. Es diferente a la exactitud pues no se refiere a la cercanía del valor verdadero de la magnitud física. Se relaciona directamente con la varianza de un conjunto de mediciones hechas por el sensor.

Sensibilidad. Se refiere a la tasa de variación de la salida del sensor con respecto a la medida de entrada de la magnitud física.

Histéresis. Es la variación en los valores de salida de un sensor cuando se realiza una medición de forma ascendente, difiriendo con los valores de una medición descendente. Es una diferencia entre dos valores de salida ante la misma entrada, dependiendo de cuál haya sido la trayectoria del sensor.

Resolución. Esta característica es la variación mínima de entrada al sensor que se necesita hacer para que se genere una respuesta detectable a la salida.

Tiempo de respuesta. Es el tiempo necesario para que el sensor entregue una respuesta estable ante un estímulo de entrada.

El sensor de temperatura y humedad utilizado previamente en el diseño del socket protésico fue el DHT11. Su voltaje de alimentación es de 5 V. Este sensor incluye un componente resistivo para medir la humedad y un componente NTC (termistor que disminuye su resistencia cuando la temperatura aumenta) para medir temperatura. Estos dos componentes están conectados a un microcontrolador interno que es el encargado de mandar las señales digitales de la temperatura y la humedad relativa.

La comunicación del sensor para la transferencia de datos se da mediante comunicación serial a través de un solo hilo y de manera bidireccional. El proceso de comunicación se lleva a cabo en tres etapas. La primera es la señal de inicio, en la que el sensor le avisa al microcontrolador externo que está listo para enviar los datos de temperatura y humedad. Al recibir la respuesta del microcontrolador, el sensor comienza el envío de datos por medio de cinco bytes, bit a bit. Los bits son enviados comenzando por el bit más significativo (MSB). Cada byte representa una parte de los datos de temperatura y humedad. El primer byte enviado representa la parte entera del dato de humedad, el segundo es su parte decimal, el tercer byte es la parte entera de la temperatura, el cuarto es la parte decimal de la temperatura y, el último byte representa los últimos 8 bits de la suma de los cuatro anteriores bytes.

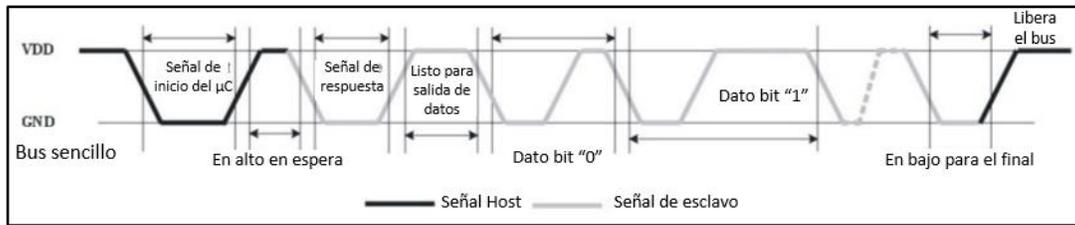


Figura 2.1. Diagrama de tiempo para el envío de datos [59].

Así, el microcontrolador puede realizar una rutina para la recepción y el procesamiento de datos, acorde con las señales que reciba desde el sensor.

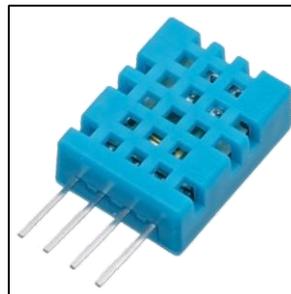


Figura 2.2. Sensor DHT11 [59].

Este sensor presenta las siguientes características, para la medición de humedad:

- Rango: 20-90%RH
- Precisión. $\pm 5\%$ RH
- Resolución: $\pm 1\%$ RH
- Tiempo de respuesta: 6 s

Y para la temperatura, presenta las siguientes:

- Rango: 0-50 °C
- Precisión. $\pm 2\%$ °C
- Resolución: $\pm 1\%$ °C
- Tiempo de respuesta: 10 s

En el presente trabajo se buscará contar con un sensor que se acople a las necesidades de la instrumentación del socket. Se analizará el tipo de comunicación que el sensor demande para la transmisión de datos y las características o parámetros del sensor principalmente.

2.3.4 Sensor de presión

Otra magnitud física a medir en la instrumentación del socket protésico es la presión. Esta variable es posiblemente la más importante debido a su relación directa con el sistema de autoajuste del socket. El seleccionar el sensor adecuado permitirá tener una mejor

exactitud de la presión existente en los sujetadores, lo que permitirá realizar los ajustes necesarios por parte de sistema de controla para llegar a la referencia deseada de presión.

En un diseño realizado previamente del socket protésico [23], se seleccionó un sensor MPX5050DP, el cual es un sensor de salida analógica. Se trata de un sensor piezoresistivo que entrega una salida de voltaje analógica. Mide la diferencia de presiones manométricas entre las dos entradas de aire, y presenta las siguientes características:

- Voltaje de alimentación: 5 V
- Rango: 0 - 50 kPa
- Precisión. $\pm 2.5\%$ kPa
- Tiempo de respuesta: 1 ms
- Sensibilidad: 90 mV/kPa
- Offset de presión mínima: 0.2 V

Se debe seleccionar para la correcta instrumentación, un sensor de presión adecuado a los requerimientos del sistema. Este debe de ser capaz de medir la presión dentro de los sujetadores del socket con buena precisión. Además, el rango de presión que pueda medir el sensor, debe de cubrir al menos los valores que los sujetadores pueden llegar a tener.

2.3.5 Servomotor

Para que el sistema de autoajuste funcione se necesita seleccionar un servomotor que pueda controlar a las servoválvulas, previamente propuestas en trabajos relacionados al diseño de este socket [23]. El servomotor gira a la válvula para permitir la salida o entrada de determinada cantidad de aire. El área de entrada de aire a través de la válvula determina el flujo que pasa por ésta. Si los sujetadores necesitan más presión, el servomotor girará la válvula ciertos grados para dejar que entre aire. La cantidad de grados que se gira la válvula depende de la variación de presión existente. Si la presión sufre un cambio considerable, entonces el servomotor posicionará la válvula de tal manera que el flujo sea suficiente para aumentar o disminuir la presión de aire.

Un servomotor es un dispositivo con un motor (puede ser de corriente directa o corriente alterna), un sistema reductor de velocidad que multiplica su torque, y un circuito interno que controla su posición angular. El servomotor es capaz de ubicarse en una posición angular específica y permanecer ahí de manera estable [60]. Generalmente el recorrido de un servomotor es de 180°; sin embargo, puede ser modificado para alcanzar los 360°.

Los servomotores son controlados mediante modulación de ancho de pulso (PWM). Se debe mandar una señal al servomotor de PWM con una frecuencia de ancho de pulso establecida. La duración del pulso define la posición angular que tomará [61].

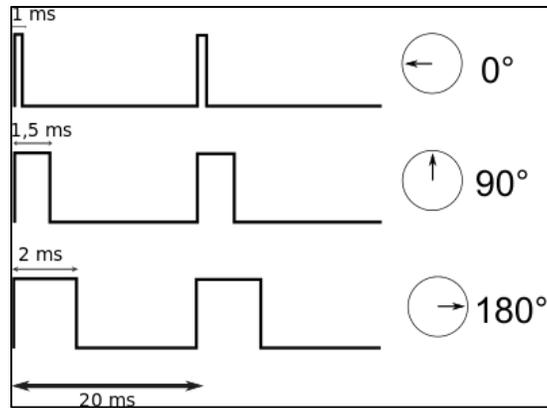


Figura 2.3. Ejemplo de PWM para un servomotor.

El servomotor utilizado en un diseño previo del socket [23], es un POWER HD 3001 HB, y presenta las siguientes características:

- Dimensiones: 60 x 30 x 70 mm
- Par: 4.4 Kg.cm (alimentado con 6 V)
- Velocidad: 60° casa 0.12 s (alimentado con 6 V)

Para el presente trabajo, se buscará contar con un servomotor que pueda ser conectado a las válvulas de aire. Además, se tendrá en consideración el espacio disponible en el socket para la colocación del servo. Por otro lado, se analizará su resolución para tener un control más exacto en el sistema de autoajuste.

2.3.6 Suministro de energía

La alimentación de energía para el sistema electrónico será proporcionada por una batería. El objetivo principal de la batería será permitirle al usuario usar su prótesis en sus actividades diarias, sin la necesidad de tener que cargar la energía constantemente.

La batería del circuito debe ser recargable, de bajo peso y tamaño, con una buena capacidad de suministro de corriente, y que dure lo necesario para el uso de la prótesis. El nivel de energía requerido dependerá de las necesidades de cada dispositivo del circuito electrónico. Estos dispositivos son el microcontrolador y el hardware que se conecta a este. Los sensores de temperatura, humedad y presión, el módulo de Bluetooth, el servomotor, y los demás circuitos integrados, son ese hardware adicional.

La velocidad de operación del reloj del sistema de un microcontrolador depende directamente del nivel de voltaje que a éste se le esté proporcionando. Es por ello que el suministro de energía es dependiente de las necesidades de operación, en este caso, del microcontrolador. En consecuencia, se deberán de seleccionar los componentes que se acerquen al nivel de voltaje utilizado en el microcontrolador. Esto para evitar pérdidas de energía mediante el uso de múltiples reguladores de voltaje.

La capacidad de una batería se mide en ampere-hora (Ah) o miliampere-hora (mAh), y se define como la cantidad de corriente que puede suministrar la batería en una hora [62]. Por ejemplo, si una batería tiene una capacidad de 6 Ah, significa que en una hora, es capaz de suministrar 6 A constantes, hasta que se descarga. Este dato es importante pues una vez que se conozca el consumo total del circuito para la instrumentación del socket, se podrá definir la capacidad necesaria de la batería.

La constitución química de la batería también es un factor a considerar. Una vez habiendo habido seleccionado la capacidad necesaria de la batería, el tipo de batería que se escoja incidirá en el peso y el tamaño de ésta. Por lo que si se escoge una batería de alta densidad o de grandes dimensiones, el diseño del socket tendría que incluir este problema. A diferencia de las baterías de Zinc-Carbón o Ni-Cd, las que permiten guardar mayor energía en un espacio reducido son las de litio [62].

Las baterías de litio-Ion (Li-ion) son celdas recargables que utilizan reacciones en las que se intercalan iones de litio entre los electrodos. Estas presentan alta capacidad de energía. Su carga nominal oscila entre los 3.6 y 3.7 V. Por otro lado, también existen las baterías de polímero de litio (LiPo), las cuales usan un polímero o un electrolito de gel. A diferencia de las baterías Li-ion, estas tienen la ventaja de poder darles la forma que se desee, tienen una densidad de carga mayor y su descarga es un poco más eficiente. Su voltaje nominal es de 3.7 V.

En el diseño del sistema de autoajuste del socket desarrollado previamente [23], se seleccionó una batería tipo LiPo con una capacidad de 6 Ah. Para aumentar su capacidad y alcanzar un voltaje mayor a los 5 V (requerido por el circuito diseñado en ese trabajo), se colocaron dos en serie y posteriormente otras dos en paralelo. Con esto se consiguió un voltaje de 7.4 V y una capacidad de 12 Ah. Esto cumplió los requerimientos planteados para cubrir las actividades cercanas a 14 horas, con un consumo máximo del sistema de 860 mA.

La elección de la batería para la instrumentación del socket protésico se basará, como se mencionó previamente, en el consumo energético del circuito y en el tiempo necesario para que el usuario pueda desarrollar sus actividades diarias sin tener que recargar al socket continuamente.

2.3.7 Componentes adicionales

Además de los elementos antes mencionados, se debe considerar aquellos componentes adicionales que servirán para hacer funcionar al sistema electrónico. Por ejemplo, el microcontrolador necesita de un cristal para regular la velocidad del mismo. Y para la parte de la alimentación, es importante seleccionar la batería correspondiente y sus reguladores adecuados. La selección de esta última dependerá del consumo final estimado de corriente que el circuito demande.

Por otro lado, en caso de así requerirlo, el usuario podrá contar con un botón de inflado de los sujetadores, acoplado al socket protésico. Este botón le permitirá al usuario alcanzar un nivel de referencia de presión predefinido, sin la necesidad de utilizar la interfaz en el dispositivo móvil.

2.4 Selección de elementos para el sistema neumático

El sistema neumático involucra los componentes que ayudarán al sistema de autoajuste a proporcionar la energía del aire. La función de este sistema es permitir el flujo de aire a través de los conductos que conectan a cada componente.

Las principales partes de este sistema son las siguientes:

- Sujetadores
- Conductos para el flujo de aire
- Válvulas
- Bomba neumática

Para poder contar con un socket protésico compacto y ligero, es necesario seleccionar los elementos del sistema neumático que sean de peso y tamaño reducido. Además, se tendrá en consideración el trabajo previo del diseño del socket protésico [23], en el que se utilizaron estos elementos del sistema neumático. Se buscará reducir el peso y el tamaño de cada componente, además de mejorar otras características, principalmente en la bomba neumática.

2.4.1 Sujetadores

Los sujetadores son el elemento de ajuste del socket protésico. Estos sujetadores fueron diseñados en el trabajo “Diseño de un socket autoajutable para prótesis de miembro inferior” [44]. Son cinco y, dependiendo de su ubicación en el socket, llevan su nombre:

- Sujetador lateral
- Sujetador medial
- Sujetador anterior
- Sujetador posterior
- Sujetador superior

El diseño de cada uno de los sujetadores se realizó con base en la posición de los músculos que conforman el muñón de un usuario. Con esto se permite tener mayor comodidad al momento de utilizar el socket en las actividades diarias [44].

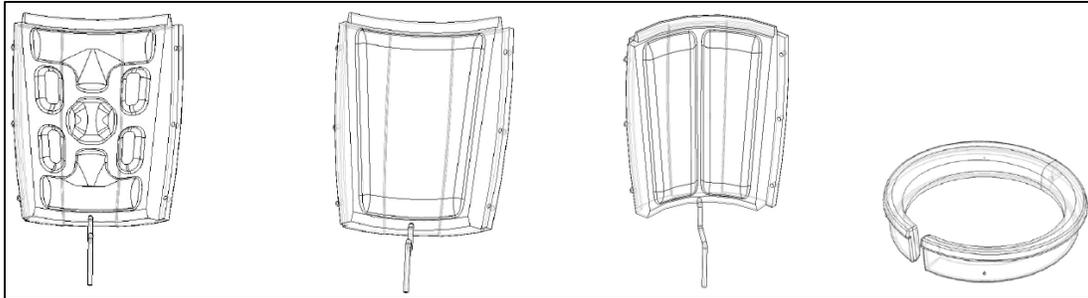


Figura 2.4. Sujetadores lateral y medial, anterior, posterior y superior [44].

Los sujetadores inflables fueron diseñados con la finalidad de poder ajustarse a los cambios en el volumen y la geometría del usuario. Permiten una sujeción segura y cómoda. Además, son los elementos que reciben, transmiten y amortiguan la fuerza ejercida por el peso del usuario. Y finalmente, tienen la función de mantener al muñón dentro del socket.

En el presente trabajo de la instrumentación del socket, únicamente se valorará un cambio en la cantidad de sujetadores a monitorear y controlar su presión directamente. En la tesis del diseño del sistema de autoajuste [23], se controlaban en pares los sujetadores, lo que obligó a utilizar dos servoválvulas y dos sensores de presión. Esto se analizará para ver la posibilidad de reducir el número de válvulas y de los sensores.

2.4.2 Conductos para el flujo de aire

Los conductos por donde pasa el aire son mangueras flexibles de plástico. Su función es conducir el flujo de aire entre los componentes del sistema neumático. En este diseño para la instrumentación del socket, se buscará que la trayectoria de los conductos sea la más corta para reducir el peso. Además, se tomará en cuenta la medida del diámetro de los conductos de entrada/salida de aire de los componentes que integran el sistema neumático. Esto para poder seleccionar aquellos que puedan ser conectados de manera adecuada, evitando la presencia de fugas.

2.4.3 Válvulas

Éstas son las encargadas de permitir la entrada o la salida del aire hacia o desde los sujetadores inflables. En este sistema neumático, están las válvulas para el inflado o desinflado manual, y las utilizadas junto con el servomotor, como servoválvulas.

Las válvulas que se utilizaron en el trabajo del diseño del sistema de autoajuste del socket [23], fueron llaves de tres vías (o tres conductos) de uso médico. El funcionamiento de las válvulas, para el control de la presión del socket, es mediante la variación del área por la que pasa el aire. Esta área es la intersección entre el orificio del mango de la válvula y el orificio de la base de la válvula.

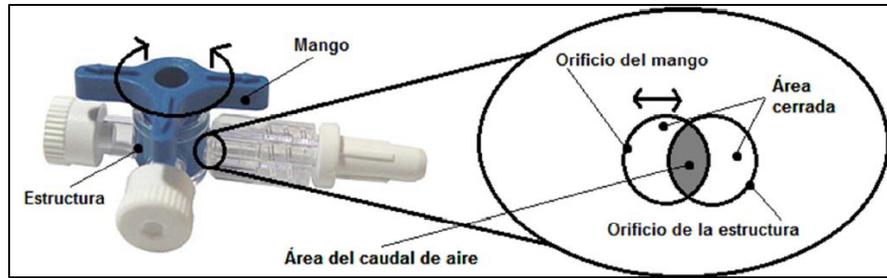


Figura 2.5. Válvula de tres vías [23].

Las servoválvulas [23] son la unión entre una de las llaves de tres vías y el servo-motor. Dependiendo de la posición angular de la salida del servomotor acoplado al mango (manteniendo fija a la base de la válvula), es como el área de paso del flujo cambiará.

Debido a que este trabajo se enfocará principalmente a la instrumentación del socket protésico, únicamente se analizará la forma en la que estas válvulas son acopladas al nuevo servomotor.

2.4.4 Bomba neumática

2.4.4.1 Clasificación de las bombas

Las bombas se pueden dividir en dos tipos básicos, de desplazamiento positivo o dinámicas [70].

Las de desplazamiento positivo funcionan mediante el cambio del volumen interno, obligando al fluido a moverse gracias a un contorno móvil. El fluido entra por un orificio y es succionado. Luego de esto, es expulsado a través de otro orificio de descarga [70]. Este tipo de bombas se puede clasificar de la siguiente manera:

- Alternativas
 - Émbolo/Pistón
 - Diafragma
- Rotativas
 - Rotor simple
 - Rotor múltiple

Las bombas dinámicas le dan una cantidad de movimiento al fluido a través de álabes o paletas giratorias. A diferencia de las de desplazamiento positivo, el volumen en éstas es abierto. Y la presión es generada al cambiar la alta velocidad del fluido al salir de un difusor [70]. Se clasifican en:

- Rotativas
 - Centrífugas o de flujo de salida radial
 - Flujo axial
 - Flujo mixto

- Diseños Especiales
 - Bomba de chorro o eyector
 - Bombas electromagnéticas para metales líquidos
 - Actuadores: (martinetes hidráulicos o neumáticos)

Por lo general, las bombas utilizadas para suministrar presiones no mayores a los 300 mmHg, son de desplazamiento positivo, alternativas y de diafragma. Éstas funcionan mediante un motor de corriente directa con un eje excéntrico. Este eje sirve para subir o bajar el diafragma de la bomba, con el fin de cambiar el volumen de esta e impulsar al aire. La bomba es de desplazamiento positivo pues el aire se mueve debido a una caída en el volumen interno de la bomba. Y el diafragma realiza un movimiento lineal y alternativo.

La bomba utilizada en un diseño previo del sistema de autoajuste del socket protésico [23], es de diafragma y tomó como consideración las recomendaciones hechas en otro diseño previo [2], y tuvo las siguientes características:

- Voltaje de alimentación: 6 V
- Presión de bombeo (a 6 V): 300 mm Hg
- Dimensiones: 65 x 35 x 20 mm
- Nivel de sonido (30 cm de distancia): 60 dB
- Peso: 56 g

Con un motor FK-280SA-18165 "Mabuchi motors" de las siguientes características:

- Rango de voltaje de operación: 10 a 15 V
- Consumo de corriente sin carga: 0.075 A
- Velocidad sin carga: 12300 RPM
- Velocidad nominal a máxima eficiencia: 10710 RPM
- Par (a 0.5 A y 3.81 W): 34.7 g-cm (3.4 mN-m)
- Diámetro del eje: 0.079 pulgadas

Para la selección de la bomba neumática en la instrumentación del socket, se deben de tomar en cuenta los siguientes parámetros:

- Rango de presiones que pueda suministrar a los sujetadores
- Dimensiones
- Material
- Nivel de sonido
- Características eléctricas

2.4.4.2 Presión máxima que debe suministrar la bomba a los sujetadores

De acuerdo a la investigación llevada a cabo en previos trabajos del diseño del socket [23], el rango de presiones recomendado para tener un eficiente sistema de autoajuste es de entre 70 y 90 mmHg. Según lo reportado, esto es debido a que superando estos niveles, el usuario comienza a sentir molestias físicas en el muñón.

Con base en esto, se buscará que la bomba cumpla con la entrega mínima de presión de 100 mmHg o mayor.

2.4.4.3 Dimensiones de la bomba

Para definir adecuadamente las dimensiones de la bomba de aire, se debe de tomar en cuenta lo siguiente:

- El tamaño de la bomba debe ser igual o menor al de la bomba utilizada previamente [23].
- El espacio máximo dentro del socket para el sistema de autoajuste (volumen) debe ser el necesario para poder colocar a la bomba.

2.4.4.4 Nivel de sonido

El sistema de autoajuste necesita de la activación de la bomba para funcionar. En el momento en que el autoajuste se realiza, éste lo hace constantemente, debido a los cambios en el volumen del muñón, que generan variaciones en la presión de los sujetadores. Para ajustar esos cambios, la bomba se activa y comienza su función de inflado en los sujetadores. El funcionamiento de la bomba hace que se genere ruido, lo que puede llegar a ser molesto para el usuario de la prótesis, es por ello que al seleccionar la bomba se buscará que el nivel de sonido producido sea el más bajo posible. La bomba seleccionada en trabajos previos del diseño del socket [23] no especifica algún valor de nivel de sonido nominal; sin embargo, se realizó una medición a 30 cm de la bomba en operación, obteniendo un valor de 60 dB. Debido a su tamaño y al tipo de bomba (de diafragma) resulta ser bastante ruidosa, por lo que este aspecto se intentará reducir al seleccionar la nueva bomba.

2.4.4.5 Características eléctricas

Debido a que el microcontrolador a utilizar probablemente trabajará a 5 V, se puede plantear que la bomba tenga una alimentación similar a dicho voltaje. Será preferible que no sobrepase los 6 V ya que habría que utilizar una fuente de energía mayor, propiciando un gasto energético muy alto a través de los reguladores de voltaje al alcanzar los 5 V en el microcontrolador y los demás componentes del socket. Además de esto, se buscará que no sobrepase la corriente de la bomba utilizada previamente [23] pues también generaría un mayor consumo de energía.

2.5 Selección de elementos para el Sistema de programación

El sistema de programación es una de las etapas de la instrumentación del socket que involucra el desarrollo de una interfaz para un dispositivo móvil en el sistema operativo Android.

Una de las razones para diseñar la interfaz de usuario en un dispositivo móvil, es que actualmente gran parte de la población mundial cuenta con uno de ellos. Por lo que, al usuario del socket protésico se le hará práctico poder interactuar con éste a través de una aplicación. Como se ha mencionado previamente, el uso de dispositivos móviles en México ha venido a la alza. De acuerdo a cifras de eMarketer, una compañía investigadora de mercado, hasta finales de 2014 se sabía que una cuarta parte de la población mexicana contaba con un Smartphone. Pero además, se espera que para el 2018, esta cantidad aumente hasta cerca de la mitad de la población. De este total, los jóvenes de entre 18 y 34 años son los que mayor cantidad de Smartphones poseen [64].

Actualmente en México, Android es el sistema operativo móvil más utilizado. Según reportes de comScore (empresa dedicada a la investigación de marketing), en la edición 2015 de Futuro Digital en México, Android está presente en el 82.5% de los dispositivos móviles, por encima de iOS y Windows con el 10.3% y el 4.2% respectivamente [63].

2.5.1 Android

Android es un sistema operativo móvil, de código abierto desarrollado por Google, basado en el kernel de Linux. Actualmente sus aplicaciones son desarrolladas para ser utilizadas en smartphones, tablets, wearables, televisores, automóviles e incluso en los lentes de Google.

Una aplicación en Android es escrita en el lenguaje de programación Java (lenguaje orientado a objetos), y se necesita de las herramientas necesarias para compilar el código de la aplicación. Para poder desarrollar una aplicación en Android, se utiliza el SDK (Software Development Kit). Este incluye al Android SDK Manager, encargado de administrar los paquetes de Android, como las plataformas instaladas y las imágenes del sistema. También contiene al AVD Manager, que permite crear dispositivos virtuales de Android para correrlos en un emulador. Para poder utilizar los dispositivos virtuales de Android se utiliza la herramienta del emulador, para detectar errores y probar las aplicaciones desarrolladas en un ambiente real de Android. Y además, el SDK contiene al Dalvik Debug Monitor Server, el cual sirve para depurar la aplicación diseñada [65].

El SDK permite compilar el código desarrollado, en conjunto con todos los archivos necesarios para la aplicación y, genera un archivo APK. Este archivo es un paquete de Android (extensión .apk) que le sirve a los dispositivos con sistema operativo Android, para instalar la aplicación [69].

Para poder desarrollar una aplicación en Android, además de SDK, es recomendable utilizar algún IDE (Ambiente de Desarrollo Integrado en español) u otra herramienta de desarrollo.

Esto permitirá crear aplicaciones más eficientemente al integrar las herramientas de Android como un editor de textos, un emulador, un compilador y un depurador de errores en una sola interfaz.

Android Studio es el IDE oficial para desarrollar aplicaciones. Fue desarrollado por Google, y basado en IntelliJ. Permite tener una vista previa de la aplicación que se está desarrollando, en tiempo real [66]. Las aplicaciones, como se mencionó previamente, son programadas en Java, por lo que este IDE, permite además, utilizar todas las herramientas que se requieran del SDK de Android.



Figura 2.6. Logotipo de Android Studio.

App Inventor es un ambiente de desarrollo en la red para crear aplicaciones en Android. Está basado en la librería de Java de bloques del MIT. Utiliza programación mediante bloques, en lugar de generar un código de programación. Sin embargo, da la posibilidad de implementar características útiles para un desarrollador. Permite el acceso al GPS, al acelerómetro, a la información de la orientación, los servicios de telefonía como envío de textos y llamadas, almacenaje en la memoria, servicios Web, el uso del Bluetooth, etc. Para programar en App Inventor solo se deben utilizar los bloques adecuados y editar sus propiedades [67].



Figura 2.7. Logotipo de App Inventor.

Eclipse fue, hasta finales de 2014, el IDE oficial para desarrollar aplicaciones en Android. Utilizando Java como lenguaje de programación, este IDE permite desarrollar proyectos de Android. Para poder tener un ambiente de desarrollo para crear aplicaciones, es necesario instalarle un plugin de Google, denominado ADT (Herramientas de desarrollo de Android en español). Este le permite a Eclipse crear interfaces de usuario, y exportar archivos en extensión .apk para poder distribuir las aplicaciones de Android [68].



Figura 2.8. Logotipo de Eclipse.

Otras herramientas para el desarrollo de aplicaciones en Android son las siguientes:

- HyperNext Android Creator
- Corona SDK
- Delphi
- Processing
- Visual Studio 2015
- Kivy
- Basic4android
- Lazarus

2.5.2 Interfaz de usuario para el socket protésico

En el trabajo de tesis profesional “Diseño e implementación de una aplicación móvil para monitoreo de un socket autoajustable” [45], se desarrolló una aplicación móvil para Android, mediante App Inventor. Esta aplicación permite lo siguiente:

- Conectarse a través del módulo Bluetooth con el microcontrolador (ATMEGA328P) del socket protésico.
- Escribir el nombre de usuario.
- Guardar datos de temperatura y humedad en una memoria microSD conectada al microcontrolador.
- Generar un archivo con los datos obtenidos desde la memoria microSD.
- Mostrar la fuerza aplicada en cada sujetador, hora y fecha actual, la temperatura y la humedad relativa existentes en ese momento.
- Graficar los datos de la temperatura.

En la instrumentación del socket protésico se contará con la posibilidad de tener una interfaz para que el usuario pueda interactuar con el socket. Esta interfaz será capaz de realizar las siguientes tareas:

- Conectarse inalámbricamente con el socket a través del módulo de Bluetooth.
- Mostrar datos de temperatura y humedad gráficamente.
- Mostrar datos de presión del aire en los sujetadores del socket.
- Permitir el ajuste de la presión en los sujetadores del socket.
- Mostrar el comportamiento del control del sistema de autoajuste.
- Comprobar que el control sistema de autoajuste está funcionando correctamente.
- Permitir la inflar o desinflar los sujetadores.
- Dar recomendaciones del uso adecuado de la prótesis.

2.6 Control del sistema de autoajuste del socket

El sistema de autoajuste del socket protésico, planteado previamente [23], [44], es el encargado de ajustar, amoldar y adoptar el socket a los cambios en el volumen del muñón del usuario, tanto manualmente como automáticamente. Este sistema es regulado por un controlador PI (proporcional e integral).

2.6.1 Función de transferencia

En la teoría de control, se define a una función de transferencia como la relación de la entrada con respecto a la salida en el dominio de Laplace. De acuerdo a Ogata [72], una función de transferencia es una relación entre la transformada de Laplace de la salida de un sistema y la transformada de Laplace de la entrada, suponiendo que las condiciones iniciales son nulas. Esto es cierto para un sistema de ecuaciones diferenciales lineales e invariantes en el tiempo.

2.6.2 Sistema de control del socket

El control del sistema de autoajuste resulta ser de retroalimentación. Este tipo de control funciona mediante la comparación constante entre la salida del sistema y su entrada. Esta diferencia es el error existente, el cual se busca reducir mediante la acción de control en la planta. El control se aplica a la planta, siempre y cuando exista el error. La siguiente figura muestra el sistema de control del socket.

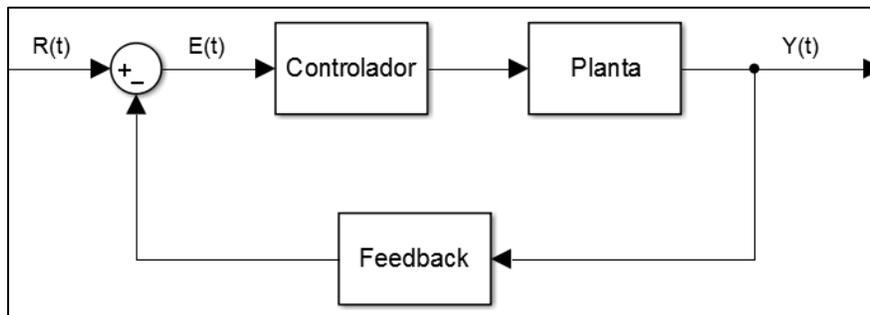


Figura 2.9. Modelo de un sistema de control retroalimentado.

En la figura anterior, se tiene lo siguiente:

$R(t)$ = Referencia del punto de ajuste deseado (presión en mmHg)

$E(t)$ = Error (diferencia de presiones)

$Y(t)$ = Respuesta/Salida medida (presión en mmHg)

2.6.3 Planta del sistema de autoajuste del socket

La planta del sistema de autoajuste del socket, ya ha sido planteada previamente [23]. En esta, se consideran a los elementos de ajuste, como aquellos a ser controlados. La propiedad que se controla es la presión interna, por lo tanto, es la salida $Y(t)$ del sistema. Para poder definir la función de transferencia de la planta, se consideró a los elementos de ajuste como capacitores neumáticos. Una vez habiendo seleccionado la salida a medir (la

presión), en el sistema de control, lo siguiente fue definir a la entrada de la planta. La presión del aire en los sujetadores varía dependiendo del flujo que entre en ellos; sin embargo, es la servoválvula la encargada de regular el caudal de aire. Entonces, se planteó una función de transferencia entre la presión de los sujetadores (como salida) y el flujo o caudal del aire. Además, de acuerdo al tamaño del área de intersección en la válvula, se tiene un flujo mayor o menor hacia los sujetadores. Por lo que, la siguiente función de transferencia definida fue la del caudal (salida) con respecto al área (entrada). Posteriormente, para poder ajustar el área de intersección, el servomotor gira el mango de la válvula hasta alcanzar cierta posición angular. Esta última, resulta ser la entrada de la planta del sistema de autoajuste. Por lo que se definió también la relación entre el área de intersección entre la servoválvula y la posición angular. El conjunto de estas tres funciones, da como resultado la función de transferencia total de la planta.

2.6.3.1 Relación del área de entrada y salida de aire con la posición angular del servomotor

La servoválvula, varía el área por donde fluye el aire hacia o desde los sujetadores, dependiendo de la intersección de los orificios del mango de la llave tres vías y de su estructura. En trabajos previos [23], se logró observar que dependiendo de la posición angular del mango, el área variaba. Además, el área cambiaba de su máximo valor hasta el mínimo en un rango de 54°. Cuando se superaba este rango, ya no había intersección entre los orificios de la válvula, los cuales tienen un diámetro de 2 mm. El área aumentaba desde cero, cuando comenzaba la intersección, hasta su máximo, cuando se avanzaban 27°. Después de alcanzar este valor, y al aumentar la posición angular hasta 54°, se disminuía el área hasta cero nuevamente (ver figura siguiente). En ese mismo trabajo se concluyó que el comportamiento entre el área A (m^2) y la posición angular θ ($^\circ$) estaba muy cercano a ser lineal. Por lo que se definió la siguiente función de transferencia constante:

$$H_{A\theta}(s) = \frac{A}{\theta} = 1 \times 10^{-7}$$



Figura 2.10. Posiciones angulares de la válvula en las que existe intersección de áreas [23].

2.6.3.2 Relación del caudal con el área de entrada y salida de aire

Para poder definir la relación entre el flujo de aire y el área de intersección en la válvula por donde pasa éste, se realizó un análisis de flujo másico a través de un orificio, en el trabajo previamente mencionado del sistema de autoajuste del socket [23]. La fórmula de flujo másico del sistema neumático del socket protésico, considerando un flujo sónico con $P_2=0.528*P_1$ (P_1 y P_2 son la presión absoluta previa y posterior al orificio de entrada de aire respectivamente), definida por K. Ogata [72], es la siguiente [23]:

$$q = 0.0404 * c * A * \frac{P_1}{\sqrt{T}}$$

En la que:

$$q = \text{flujo másico} \frac{kg}{seg}$$

$$c = 1 \text{ Coeficiente de descarga}$$

$$T = 289 \text{ K Temperatura de entrada del sistema}$$

Por lo tanto:

$$q = 286.74 * A \frac{kg}{seg}$$

Así que, la función de transferencia obtenida, entre el caudal q (kg/seg) y el área A (m^2), se constituyó de la siguiente manera [23]:

$$H_{qA}(s) = 286.74 = \frac{q(s)}{A(s)}$$

2.6.3.3 Relación entre la presión en los sujetadores y el caudal del aire

Al modelarse los sujetadores como capacitores neumáticos, se necesitó definir, precisamente, su capacitancia neumática. Este es la relación entre la variación de la masa con respecto a la variación de la presión [72]:

$$C = \frac{\Delta masa}{\Delta presión} \frac{kg}{N/m^2} \text{ o } \frac{Kg * m^2}{N}$$

$$C = \frac{dm}{dP} = V \frac{d\rho}{dP}$$

Finalmente, se dedujo que la capacitancia neumática puede definirse (asumiendo que el aire es un gas ideal), como sigue [23]:

$$C = \frac{V}{n * R_{aire} * T}$$

En la que:

$V =$ Volumen de los sujetadores

$n = 1$ (proceso isotérmico)

$$R_{aire} = 287 \frac{N * m}{kg * K}$$

$T = 310.5 K$ Temperatura del sistema y que es la del cuerpo humano a $37.5^{\circ}C$

El volumen de los sujetadores se estableció como $V = 1.936 \times 10^{-3} m^3$. Así que, la capacitancia neumática resultó [23]:

$$C = \frac{1.936 \times 10^{-3}}{(1) * (287) * (310.5)} \frac{kg}{N/m^2}$$

$$C = 21.725 \times 10^{-9} \frac{kg}{N/m^2}$$

Una vez habiendo obtenido la capacitancia de los sujetadores, se procedió a modelar al sistema neumático. La función de transferencia resultante ente la presión de los sujetadores y el flujo másico de aire, se muestra a continuación.

$$H_{pq}(s) = \frac{p(s)}{q(s)} = \frac{1/C}{s}$$

$$H_{pq}(s) = \frac{4.603 \times 10^7}{s}$$

En donde:

$p(s) =$ Presión manométrica en el sujetador en Pa

$q(s) =$ Flujo de entrada/salida de aire en kg/seg

$H_{pq}(s) =$ Función de transferencia entre la presión y el flujo másico de aire

$$C = 21.725 \times 10^{-9} \frac{kg}{N/m^2} \text{ Capacitancia neumática del sujetador}$$

2.6.3.4 Función de transferencia del sistema

Una vez habiendo obtenido las relaciones anteriores, se propuso la función de transferencia de la planta del sistema, de la siguiente manera:

$$H(s) = H_{pq}(s)H_{qA}(s)H_{A\theta}(s)$$

$$H(s) = \left(\frac{4.603 \times 10^7}{s} \right) (286.74)(1 \times 10^{-7})$$

$$H(s) = \frac{1319.86}{s}$$

2.6.4 Controlador del sistema de autoajuste del socket

En el sistema de autoajuste del socket, se implementa un controlador de tipo PI (proporcional integral), presentando la siguiente función de transferencia:

$$G(s) = k_p \left(1 + \frac{1}{T_i s} \right)$$

En donde k_p es la ganancia proporcional del controlador y T_i es el tiempo de integración.

Como se observa en la anterior función de transferencia, tiene un cero en $s = -1/T_i$ y un polo en $s = 0$. Si se diseña un control PI con los valores adecuados de K y T_i , se puede lograr una respuesta transitoria a una entrada escalón, con un sobrepaso pequeño o inexistente. Además, el controlador PI es un filtro pasa-bajas, lo que permite atenuar los componentes con altas frecuencias de la señal; esto genera que el controlador sea más lento que uno con un control derivativo [71]. Sin embargo, en el sistema de autoajuste del socket, este controlador es suficiente, pues el sistema neumático es un sistema lento, a diferencia de los sistemas eléctricos. Esto se debe a que los ajustes en la presión pueden llegar a tardar algunos segundos, en contraste con los sistemas electrónicos que pueden reaccionar en milisegundos ante los cambios presentes en un sistema de control.

El controlador PI implementado en previos trabajos, fue diseñado para cumplir con la estabilidad, regulación (sin error en estado permanente), desempeño (respuesta rápida) y que pudiese ser regulado a diferentes valores de referencia [23]. Los valores obtenidos, después de un análisis, fueron los siguientes:

$$k_p = 1.71$$

$$k_i = 6.84$$

Estos valores representan las constantes proporcional y de integración, en el control PI. La constante k_i se relaciona con la función de transferencia $G(s)$, de la siguiente manera:

$$k_i = \frac{k_p}{T_i}$$

En el siguiente capítulo se presentará una nueva propuesta del control PI para mejorar el rendimiento del sistema.

2.6.5 Control planteado para el sistema de autoajuste del socket

Con estas consideraciones realizadas, se implementó digitalmente el controlador diseñado en el microcontrolador ATMEGA328P y se probó en el sistema de autoajuste creado en dicho trabajo. En la siguiente figura se muestra el esquema del control implementado en socket [23]. Sin embargo, se debe señalar que no se contó con alguna herramienta que pudiese comprobar el funcionamiento real del control automático. Es por esta razón, que en el presente trabajo se desarrollará una aplicación en la que se pueda observar el comportamiento del control, para saber si se están cumpliendo los requerimientos de diseño del controlador.

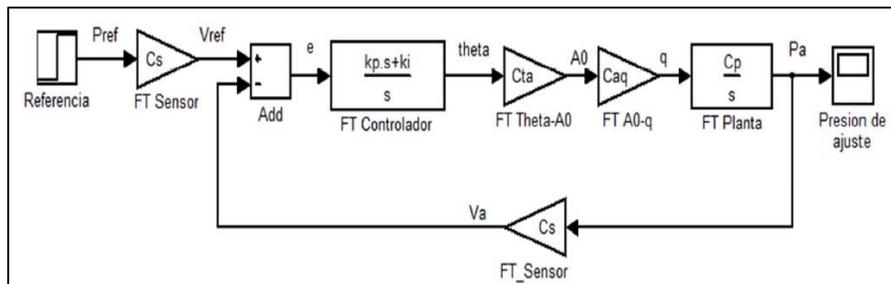


Figura 2.11. Esquema del control planteado para el socket [23].

En el esquema mostrado arriba, los valores de cada variable y su significado se detallan a continuación:

- *Pref*: Presión de referencia en Pa
- *Vref*: Voltaje de referencia
- *Va*: Voltaje de ajuste
- *e*: Diferencia entre *Vref* y *Va*
- *kp*: Constante proporcional del controlador
- *ki*: Constante integral del controlador
- *theta*: Salida del controlador (posición angular del servomotor) en grados
- *A0*: Área de paso del flujo de aire en m^2
- *q*: Flujo de aire en $\frac{kg}{s}$
- *Pa*: Presión de ajuste en Pa
- *Cta*: Constante de la función de transferencia entre el área de paso del flujo de aire y la posición angular del servomotor, $Cta = 1 \times 10^{-7}$
- *Ccq*: Constante de la función de transferencia entre el flujo de aire y el área de paso, $Ccq = 286.74$
- *Cp*: Constante de la función de transferencia entre la presión de ajuste y el flujo de aire, $Cp = 4.603 \times 10^7$
- *Cs* = Constante del sensor de presión 0.0121

La función de transferencia de la planta se simplificó posteriormente, unificando C_{ta} , C_{aq} y C_p , obteniendo una nueva constante para la función de transferencia total [23]:

$$C_{tp} = C_{ta} * C_{aq} * C_p$$

En la siguiente figura se muestra el esquema del control, con a función de transferencia total de la planta:

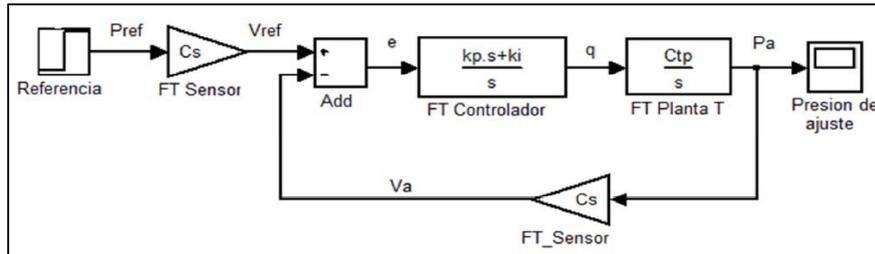


Figura 2.12. Esquema del control con función de transferencia total de la planta del socket [23].

En este capítulo se conceptualizaron los elementos del socket autoajustable para el correcto entendimiento y desarrollo de la instrumentación. Además, se analizaron los trabajos previamente realizados que se relacionan con el socket protésico autoajustable de este trabajo. Esto permite sentar una base para poder realizar una instrumentación enfocada en las necesidades del diseño de este socket.

Por otro lado, se plantearon los requerimientos de la instrumentación del socket. Estos requerimientos se verán reflejados en cada uno de los sistemas y etapas de este trabajo y evitar que se diseñe a la deriva.

Para cada sistema se presentaron los conceptos importantes para partir de una base en el posterior diseño de configuración. Se analizaron además, los elementos de cada uno de los sistemas previamente utilizados en el socket y su posible modificación para la instrumentación.

Finalmente se presentó el control de presión existente en el socket, lo que servirá en el diseño de configuración para poder mejorarlo en su tiempo de respuesta y comprobarlo mediante la interfaz.

Capítulo 3. DISEÑO DE CONFIGURACIÓN

En este capítulo, se presenta el desarrollo de la instrumentación del socket protésico a nivel transfemoral. Se detalla la selección de los componentes en los sistemas electrónico, de programación y neumático. En cada sistema se justifica la razón de la utilización de dichos elementos, con base en los requerimientos señalados en el capítulo anterior.

También se describe el diseño del circuito electrónico, que es dependiente del espacio en el socket en el que se ubicará. Así mismo, se muestra el desarrollo del CAD del socket protésico autoajustable conteniendo la instrumentación propuesta. Esto con el fin de incentivar la manufactura posterior del socket y su utilización a futuro en pruebas estáticas y dinámicas.

Además, se diseña un algoritmo del funcionamiento general de la instrumentación del socket. Este algoritmo involucra el comportamiento del sistema electrónico y neumático, en conjunto con el de la aplicación en Android para un dispositivo móvil.

Finalmente, se explica el funcionamiento general del socket, siendo utilizado con la aplicación en un dispositivo móvil.

3.1 Sistema electrónico

La primera parte a desarrollar es el sistema electrónico. Los componentes electrónicos se seleccionaron de acuerdo con las necesidades de la instrumentación del socket, mencionadas previamente.

3.1.1 Microcontrolador

Para poder controlar al sistema de autoajuste e implementar la instrumentación, se debió seleccionar el microcontrolador que cumpliera con los requerimientos necesarios.

La primera característica que debió cumplir el microcontrolador, es que contase con los pines de entradas y salidas suficientes para controlar al socket. Los pines de entrada son utilizados para recibir información externa, ya sea desde los sensores o desde algún módulo de comunicación. Y los pines de las salidas son los encargados de mandar señales hacia los actuadores, con el fin de realizar una tarea específica; sin embargo, también pueden utilizarse para comunicarse con algún módulo de comunicación. En el caso del socket protésico, las entradas y salidas son las siguientes:

Entrada

- Sensor de Presión
- Sensor de Temperatura y Humedad
- Botón de inflado/desinflado

Salida

- Servomotor
- Bomba neumática

Entrada/Salida

- Módulo Bluetooth

Debido a que algunos de los componentes electrónicos del socket trabajan con señales analógicas, el microcontrolador debe de contar con al menos un convertidor analógico digital.

En el funcionamiento de la servoválvula, es necesario utilizar un servomotor, que se encarga del posicionamiento del mango en la válvula, para regular la entrada y salida de aire. Como ya se mencionó previamente, un servomotor necesita recibir pulsos de voltaje a determinada frecuencia para poder definir su posición. Así que, el microcontrolador debe de contar con al menos un módulo de PWM.

Para la transmisión y recepción de datos entre el socket y el dispositivo móvil, se utiliza un módulo de Bluetooth, el cual funciona mediante comunicación serial. Es por eso que el microcontrolador seleccionado tiene que tener un módulo USART o de algún otro tipo como SPP o HCI.

El algoritmo a implementarse involucra algunas funciones que deben de estar trabajando al mismo tiempo. Así que, se hace énfasis en utilizar un microcontrolador que permita el uso de múltiples interrupciones. Esto evitará la necesidad de colocar diferentes dispositivos para realizar cada una de las tareas.

Debido a que en el algoritmo del funcionamiento del socket, se deben de realizar determinadas tareas en un tiempo definido, es conveniente también, que el microcontrolador cuente con múltiples timers.

Para guardar los datos obtenidos de los sensores, es necesario que se cuente con una memoria interna, como la EEPROM.

El microcontrolador seleccionado fue el PIC16F887, de la compañía Microchip. De acuerdo con su hoja de datos, presenta una arquitectura RISC de 35 instrucciones. Su frecuencia de operación máxima es de 20 MHz. Si se trabajase con una velocidad menor, también se debe ajustar su voltaje de alimentación, el cual puede variar desde 2 hasta 5.5 V. Se escogió el empaque de 40 pines debido a la facilidad de su manejo y tamaño, de los cuales, 35 son de entrada/salida. Cuenta con una memoria de programa de tipo Flash con 8192 palabras. Su memoria de datos (RAM) es de 368 bytes. Además, cuenta con una memoria EEPROM de 255 bytes que puede soportar escritura hasta 1,000,000 de veces. Presenta 14 canales de convertidor analógico digital. Para poder utilizar un módulo PWM por Hardware, cuenta con ECCP. También, este microcontrolador, presenta la característica de tener un módulo

EUSART, que es la versión mejorada del módulo USART. Y finalmente, permite utilizar timers de manera síncrona o asíncrona.

Dispositivo	Max Velocidad CPU	Pines	Memoria de programa	Memoria de Datos		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	Timers 8/16-bit
			Flash 8/16-bit (palabras)	SRAM (bytes)	EEPROM (bytes)					
PIC16F887	20 MHz	40	8192	368	256	35	14	1/1	1	2/1

Tabla 3.1. Características del microcontrolador PIC16F887.

Las características de este microcontrolador, permitirán que los elementos planteados en los requerimientos puedan ser implementados para la instrumentación.

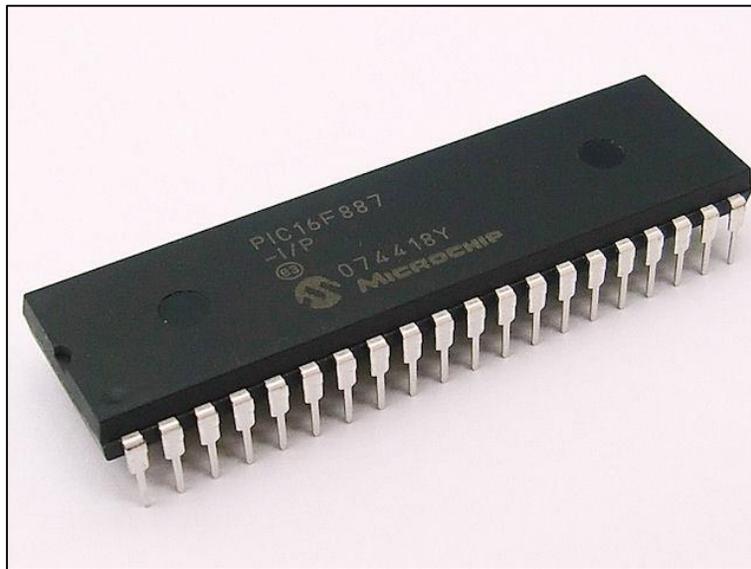


Figura 3.1. Microcontrolador PIC16F887 utilizado para la instrumentación.

3.1.2 Módulo de Bluetooth

Para lograr la comunicación entre el microcontrolador y el dispositivo móvil, se seleccionó el módulo de Bluetooth RN42 (mostrado en la siguiente figura) de baja potencia de la compañía Roving Networks. A diferencia del módulo utilizado previamente [45], el HC-06, este módulo está debidamente certificado. Cuenta con la certificación FCC, la cual garantiza que el dispositivo cumple los estándares de interferencia electromagnética establecidos por la Comisión Federal de Comunicaciones (FCC, por sus siglas en inglés) de Estados Unidos. Por otro lado, cuenta con la certificación CE (Conforme para Europa), que corrobora que el dispositivo cumple con las normas de seguridad, salud y ambientales para equipos electrónicos, médicos y de telecomunicaciones [73]. También está certificado con Bluetooth SIG, lo que garantiza que el dispositivo cumple con los acuerdos de licencia de Bluetooth. Esto da la seguridad de contar con un módulo que tiene bajas probabilidades de fallar, en comparación con el HC-06 de origen chino.

Para la implementación del circuito de este trabajo, se utilizó la tarjeta SparkFun Bluetooth Mate Silver [74] desarrollada por la empresa Sparkfun. Esta tarjeta tiene al módulo RN42, y presenta la ventaja de poder alimentarse de 3.3 a 6 V, y permite ser soldada más fácilmente al circuito del socket.

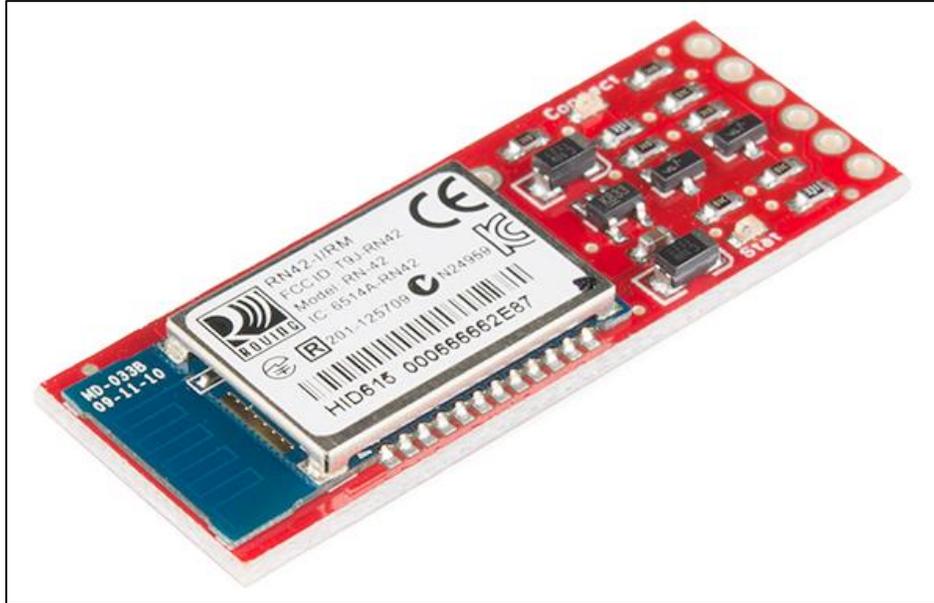


Figura 3.2. Tarjeta para el módulo RN42 de Sparkfun [74].

Con base en la hoja de datos, presente en los anexos de este trabajo, se presentan las siguientes características del módulo:

- Alimentación de 5 V (circuito adaptado)
- Consumo promedio de 25 mA
- Comunicación serial para comunicarse con el microcontrolador, mediante el módulo UART y radiofrecuencia en el aire
- Es de clase 2
- Puede comunicarse a una distancia de hasta 20 metros
- Tiene velocidades de tasa de baudios desde 1200 bps hasta 921 Kbps

Debido a la velocidad seleccionada en el microcontrolador, este módulo se trabajará a una tasa de baudios de 9600 bps, la cual es la configurada por defecto.

3.1.3 Sensor de temperatura y humedad

El componente para medir la temperatura y la humedad del socket protésico debe de lograr transmitir los datos con una precisión y exactitud alta. Además, es necesario que la comunicación sea sencilla para la transmisión hacia el microcontrolador.

A continuación se muestra una tabla de las características de seis sensores que pueden medir temperatura y humedad. También se engloba el sensor implementado previamente en el trabajo del diseño de una aplicación para el socket [45].

Sensor	Características		
	Humedad	Temperatura	Comunicación
DTH11	Rango: 20-90%RH Resolución: $\pm 1\%$ RH Precisión: $\pm 5\%$ RH T. de respuesta: 6 s	Rango: 0-50 °C Resolución: $\pm 1^\circ\text{C}$ Precisión: $\pm 2^\circ\text{C}$ T. de respuesta: 10 s	Serial Un hilo
DHT22	Rango: 0-100%RH Resolución: $\pm 0.1\%$ RH Precisión: $\pm 2\%$ RH T. de respuesta: 2 s	Rango: -40 a 125 °C Resolución: $\pm 0.1^\circ\text{C}$ Precisión: $\pm 0.2^\circ\text{C}$ T. de respuesta: 2 s	Serial Un hilo
PH1125	Rango: 10-95%RH Precisión: $\pm 2\%$ RH Resolución: 0.4 %RH T. de respuesta: 10 s	Rango: -30°C a +80°C Resolución: 0.2 °C Precisión: $\pm 0.75^\circ\text{C}$	Analógica (fórmula)
HHT02D	Rango: 0-100%RH Resolución: $\pm 0.03\%$ RH Precisión: $\pm 4.5\%$ RH T. de respuesta: 4 s	Rango: -40 a 123.8 °C Resolución: $\pm 0.01^\circ\text{C}$ Precisión: $\pm 0.5^\circ\text{C}$ T. de respuesta: 5-30 s	Serial Dos hilos
SHT15	Rango: 0-100%RH Resolución: $\pm 0.05\%$ RH Precisión: $\pm 2\%$ RH T. de respuesta: 8 s	Rango: -40 a 123.8 °C Resolución: $\pm 0.01^\circ\text{C}$ Precisión: $\pm 0.3^\circ\text{C}$ T. de respuesta: 5-30 s	Serial Dos hilos
HTU21D-F	Rango: 0-100 %RH Resolución: $\pm 0.04\%$ RH Precisión: $\pm 2\%$ RH T. de respuesta: 5-10 s	Rango: -40 a 125 °C Resolución: $\pm 0.01^\circ\text{C}$ Precisión: $\pm 0.3^\circ\text{C}$ T. de respuesta: 10 s	Serial I ² C

Tabla 3.2. Características de distintos sensores de temperatura y humedad.

Para elegir el sensor, se debió analizar cada una de sus características. El sensor seleccionado fue el DHT22 supera, en todos los aspectos, a su antecesor utilizado en este socket, el DHT11. El rango de presión y temperatura es más amplio, su resolución es mejor en temperatura y humedad y, resulta ser más preciso. El tiempo de respuesta de este sensor es más rápido, pues únicamente se debe esperar dos segundos para tener una nueva muestra de temperatura y humedad. En cuanto al tamaño, resulta ser muy similar al DHT11, por lo que no existe un aumento significativo que repercuta en el CAD del socket.

El sensor PH1125 tiene la misma precisión que el DHT22, y al ser analógico, su programación puede resultar más sencilla; sin embargo, tiene un tiempo de respuesta mayor y el costo puede llegar a elevarse, dependiendo del lugar de adquisición, hasta siete veces.

Otro de los sensores propuestos fue el HHT02D. Este sensor, tiene un rango muy similar al DHT22, tanto en temperatura y humedad y, una mejor resolución ($\pm 0.03\%$ RH y ± 0.01 °C). Sin embargo, es menos preciso en la humedad relativa y, el tiempo de respuesta en la temperatura puede llegar a alcanzar hasta 30 segundos. Si se llegase a necesitar el dato de temperatura, no se podría tomar tan rápidamente como con el DHT22 (tiempo de respuesta de dos segundos).

El sensor SHT15 es muy similar al HHT02 en las características del rango y la resolución. La desventaja que presenta es que aumenta el tiempo de respuesta, no solo en la medición de humedad (hasta 30 s), sino que también lo hace en la de temperatura (ocho segundos).

El último sensor analizado fue el HTU21D-F, que a diferencia de los anteriores, se comunica mediante el protocolo I²C. Es muy similar en precisión al DHT22 y tiene mejor resolución. Sin embargo, el tiempo de respuesta es hasta cinco veces superior y su precio es ligeramente mayor.

3.1.3.1 Sensor DHT22

El DHT22 (mostrado en la siguiente figura) es un sensor digital de temperatura y humedad de bajo costo. Su sensor interno de humedad es de tipo capacitivo y para medir la temperatura utiliza un termistor. Estos elementos son conectados a un microcontrolador interno de ocho bits, mediante el cual se envían los datos digitales hacia un microcontrolador externo.

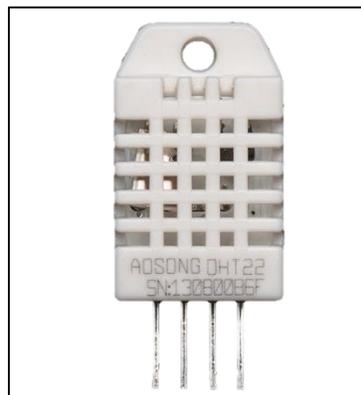


Figura 3.3. Sensor de temperatura y humedad DHT22.

Este sensor presenta las siguientes características:

- Voltaje de alimentación: 3.3 a 5.5 V
- Consumo al medir: 1.5 mA
- Rango de humedad: 0-100 %RH
- Resolución de humedad: ± 0.1 % RH
- Precisión de humedad: ± 2 %RH

- Rango de temperatura: -40 a 125 °C
- Resolución de temperatura: ± 0.1 °C
- Precisión de temperatura: ± 0.2 °C
- Tiempo de respuesta: 2 s

Para la comunicación de los datos digitales con el microcontrolador, el sensor utiliza un bus de un solo hilo. El proceso de comunicación debió de ser implementado en el PIC16F887 para poder recibir los datos desde el sensor.

Los datos de temperatura y humedad son enviados en 40 bits hacia el microcontrolador PIC16F887. Estos bits están divididos en cinco bytes. Los primeros dos bytes enviados corresponden al valor de la humedad relativa multiplicado por diez. Los segundos dos bytes son diez veces el valor de la temperatura. El último byte sirve para verificar que los datos fueron enviados correctamente, y corresponde a los ocho bits menos significativos de la suma de los cuatro anteriores bytes. Los bits son enviados hacia el microcontrolador de manera secuencial comenzando por el bit menos significativo.

El proceso de envío de datos de temperatura y humedad, basado en la hoja de datos del sensor (anexa al final de esta tesis), es el siguiente:

1. El DHT22 espera la señal de inicio del microcontrolador PIC16F887, y cuando la recibe, el sensor le manda una señal de respuesta para indicar que está listo para mandar los datos.
2. El sensor comienza a mandar los 40 bits hacia el microcontrolador. Cada bit comienza con una señal en bajo de 50 μ s. Posteriormente cambia a alto y dependiendo de cuánto tiempo dure así, se tratará de un uno o un cero. Si dura entre 26 y 28 μ s, es un cero, pero si dura 70 μ s el bit es un uno.

El pseudocódigo implementado en el microcontrolador es el siguiente:

1. Configuración del microcontrolador. Se configura el puerto de entrada para el sensor de temperatura y humedad. También se configuran los timers necesarios para implementar la rutina de comunicación con el sensor. Y finalmente se configuran las interrupciones de los timers.
2. Declarar arreglos y variables para guardar datos de temperatura y humedad.
3. Mandar señal de inicio del microcontrolador hacia el sensor DHT22.
4. Revisar la respuesta del sensor hacia el microcontrolador.
5. Si hubo respuesta continuar a 6; de lo contrario, regresar a 3.
6. Leer y guardar bytes enviados por el sensor DHT22.
7. Revisar si hubo error al recibir los datos.
8. Si hubo error regresar a 3; de lo contrario, continuar.
9. Convertir los bytes recibidos a datos de temperatura y humedad.
10. Guardar los datos en un arreglo.

11. Si se han almacenado suficientes datos en el arreglo, continuar a 12; de lo contrario, regresar 3.
12. Guardar datos de temperatura y humedad en la EEPROM.
13. Esperar hasta el siguiente tiempo de lectura del sensor y regresar al punto 3.

Más adelante, en los anexos de esta tesis, se describe el código en lenguaje C programado en el microcontrolador PIC16F887, a partir del pseudocódigo anterior.

3.1.4 Sensor de presión

El otro sensor del sistema de autoajuste del socket es el sensor de presión. Este elemento es uno de los más importantes pues mide la variable que será controlada en el sistema. El sensor de presión seleccionado previamente [44], [23] cumple con los requerimientos para ser utilizado en el sistema de autoajuste. Esto se debe a que la presión de referencia en los sujetadores, seguirá estando en el mismo rango de valores de 70 a 90 mmHg, el cual es cubierto por el sensor de presión. Además, en este trabajo se utilizará una bomba neumática de reducidas dimensiones, como las utilizadas en aplicaciones médicas (para baumanómetros principalmente). Estas bombas, por lo general, no superan los 40 kPa de presión manométrica, y el sensor puede llegar a medir hasta 50 kPa.



Figura 3.4. Sensor de presión MPX5050DP.

Las características de este sensor (mostrado en la figura superior) ya han sido descritas en el capítulo anterior. La señal que entrega al microcontrolador es un voltaje analógico, y es convertida a una señal digital mediante el convertidor analógico digital. La función de transferencia de este sensor entre el voltaje de salida V y la presión manométrica P en milímetros de mercurio, es la siguiente [23]:

$$V = 0.0121P + 0.1992$$

La presión entregada por el sensor es utilizada en el control del sistema de autoajuste para la obtención del error entre la salida y la referencia del sistema. Por esta razón, es necesario definir a la presión (en milímetros de mercurio) en función del valor digital del convertidor

ADC. El voltaje recibido en el convertidor se relaciona con el valor digital de la siguiente manera:

$$V = \frac{5adc}{MaxADC}$$

En donde:

adc = Valor digital de la señal analógica

MaxADC = Valor digital máximo del ADC, igual a $2^n - 1$

n = Número de bits para dividir la señal analógica

La presión se puede despejar de la función de transferencia entre el voltaje de salida y la presión manométrica, definida previamente.

$$P = \left(\frac{1}{0.0121} \right) (V - 0.2)$$

Si se sustituye el voltaje:

$$P = \left(\frac{1}{0.0121} \right) \left(\frac{5adc}{MaxADC} - 0.2 \right)$$

Y si se simplifica la función:

$$P = \left(\frac{1}{0.0121MaxADC} \right) (5adc - 0.2MaxADC)$$

Con esta función se puede manipular el valor de la presión (en milímetros de mercurio), en el controlador del sistema, para poder calcular el error. El valor de *MaxADC* es constante una vez que se configuró la resolución del convertidor analógico digital.

3.1.5 Botón de inflado/desinflado

La última entrada al microcontrolador es el botón de inflado y desinflado. El tipo de botón utilizado es normalmente abierto. Este elemento le permitirá al usuario inflar los sujetadores hasta un nivel de presión de referencia, en caso de que los sujetadores no hayan sido inflados aún. Si los sujetadores se encuentran con una presión dentro del rango de valores permitidos o superior a este, el botón servirá para liberar el aire en los sujetadores hasta alcanzar la presión del medio (atmosférica). Este botón se encontrará ubicado en la parte externa del socket con el fin de que sea accesible para el usuario.

3.1.6 Servomotor

El sistema de autoajuste del socket lleva acabo su función de control gracias a la posición angular del mango en la servoválvula, lo que permite la regulación del flujo de aire hacia o desde los sujetadores. El servomotor seleccionado, para el presente trabajo de la

instrumentación del socket, es diferente al utilizado en el anterior trabajo del diseño de autoajuste [23]. Se trata del micro servomotor SG90 con las siguientes características (obtenidas de la hoja de datos anexada en la parte final de este escrito):

- Peso: 9 g
- Dimensiones: 22.2 x 11.8 x 31 mm
- Par: 1.8 Kg.cm (alimentado con 5 V)
- Velocidad: 60° cada 0.1 s (alimentado con 5 V)



Figura 3.5. Servomotor SG90.

Una de las características principales consideradas para la selección de este servomotor fue que fuese de un tamaño reducido para poder colocarlo correctamente en el socket. En comparación al servomotor utilizado previamente, el SG90 es mucho más pequeño en volumen, como se puede observar en las dimensiones descritas previamente de ambos servomotores.

La posición angular de este servomotor tiene un rango de 180°, y depende de la frecuencia del ancho de pulso que se le ajuste. La duración del ciclo de trabajo para la posición angular inicial (0°) debe de ser de aproximadamente 1 ms. Y para alcanzar su posición final de 180°, el ciclo de trabajo debe durar aproximadamente 2 ms. El periodo en el que debe de trabajar el servomotor es de 50 Hz (20 ms).

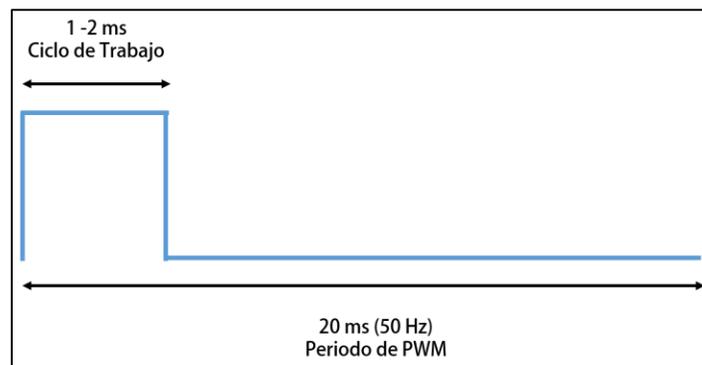


Figura 3.6. Características del PWM para el servomotor SG90.

La entrada de la función de transferencia del sistema de control es la posición angular de la servoválvula, que es también la del servomotor. Por lo tanto, para definir un ángulo de entrada, se debe de definir al mismo tiempo el ciclo de trabajo del PWM. El PWM por hardware del microcontrolador PIC16F887 utiliza al timer 2 para poder ser implementado. El periodo T_{PWM} del PWM puede ser calculado con la siguiente fórmula, que involucra al registro PR2 del timer 2 y a la prescala programable (divisor de frecuencia asociada a los pulsos del reloj del microcontrolador):

$$T_{PWM} = (PR2 + 1)(PRE)(1 \text{ c. m.})$$

El periodo $PR2$ es un número de 8 bits de cero a 255. La prescala PRE puede ser de 1, 4 o 16. Y el ciclo máquina $c. m.$ depende de la frecuencia del oscilador F_{osc} calculándose con la siguiente fórmula:

$$1 \text{ c. m.} = \frac{4}{F_{osc}}$$

Sin embargo, para lograr que el PWM tenga un periodo aproximado de 20 ms, sería necesario reducir la velocidad del reloj del microcontrolador hasta 500 kHz, con 156 de valor de PR2 y 16 de prescala. Esta reducción en la velocidad implica no poder realizar eficientemente la transmisión de datos por comunicación serial a través del módulo Bluetooth, pues no se podría configurar correctamente la tasa de baudios a 9600.

Debido a estos problemas, se decidió diseñar un PWM por software, mediante la interrupción de un timer cada 20 ms, y cambiando a uno el pin conectado al servomotor durante el ciclo de trabajo obtenido de una variable. El valor del ciclo de trabajo es definido mediante el algoritmo del control PI, descrito posteriormente en este capítulo.

El pseudocódigo implementado en el microcontrolador, para el control del servomotor, es el siguiente:

1. Configuración del microcontrolador. Se configura el puerto de salida para el servomotor. Luego se configura al timer para mandar una interrupción cada 20 ms. Y finalmente se configura la interrupción del timer.
2. Declarar la variable para el ciclo de trabajo del PWM.
3. Esperar hasta que se presente la interrupción del timer.
4. Al presentarse la interrupción del timer, reconfigurarlo para volver a interrumpirse cada 20 ms.
5. Cambiar a uno la salida del servomotor, y esperar el tiempo definido por la variable del ciclo de trabajo (establecido mediante el algoritmo del control PI).
6. Cambiar a cero la salida del servomotor.
7. Volver al punto 3.

Al final de este trabajo, en los anexos, se describe el código en lenguaje C programado en el microcontrolador PIC16F887, basado en el pseudocódigo de arriba.

3.1.7 Circuito para la instrumentación

El circuito desarrollado para la instrumentación del socket, es la unificación de los elementos del sistema electrónico, para que funcionen en conjunto. El programa utilizado para el diseño del circuito impreso fue EAGLE, el diseño de configuración del circuito se muestra en la figura que aparece a continuación. Los elementos del sistema electrónico fueron acomodados en una tarjeta en forma de medio círculo, con un diámetro de 67 mm, suficiente para ser ubicada en el interior de la base del socket.

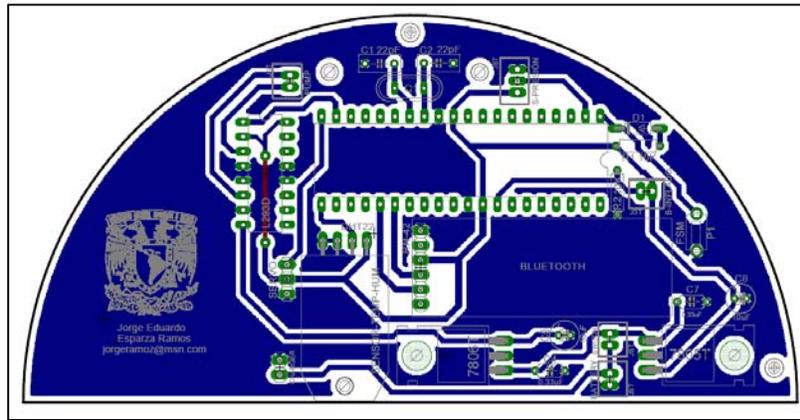


Figura 3.7. Circuito Impreso para la instrumentación del socket.

3.1.8 Batería para alimentar al circuito

La batería que se utilizará en el socket es de polímero de Litio, y debe poder suministrar la energía suficiente para que el usuario de la prótesis no tenga que recargarla en al menos 14 horas de uso continuo aproximadamente, similar a la diseñada en el trabajo previo del sistema de autoajuste [23]. Para seleccionar la batería, se analizó la cantidad máxima de energía consumida por los componentes eléctricos y electrónicos del sistema, como se muestra en la siguiente tabla.

Componente	Consumo máximo (mA)
Regulador 7805T	7
Regulador 7806T	6
Módulo de Bluetooth RN-42	40
Bomba neumática	300
Servomotor SG90	220
Sensor presión MPX5050	10
Sensor DHT22	1.5
Microcontrolador PIC16f887	95
Circuito restante	40
TOTAL	719.5

Tabla 3.3. Componentes eléctricos y electrónicos de la instrumentación.

Este consumo total del circuito del socket refleja el consumo máximo que podría presentar al considerar los valores máximos de los principales componentes. A diferencia del diseño previo [23], en este análisis solo se consideró un servomotor, debido a que únicamente se utilizará una servoválvula para el sistema de autoajuste del socket. Además, tanto la bomba neumática como el servomotor no están trabajando constantemente, solamente se activan cuando existe una variación de la presión, lo que genera una reducción en el consumo real del circuito. Se utilizará el mismo arreglo del diseño del sistema de autoajuste anterior [23], mediante cuatro baterías de 3.7 V de 6 Ah, dos en serie para tener 7.4 V y éstas en paralelo, para poder alcanzar 12 Ah. Esto permitiría utilizar el socket durante 17 horas aproximadamente.

3.2 Sistema neumático

La segunda parte del diseño de configuración corresponde al sistema neumático, el cual lo compone los sujetadores, las válvulas, los conductos para el flujo de aire y la bomba neumática. En la selección de cada uno de los elementos se consideraron las necesidades del socket, definidas previamente en este mismo trabajo.

3.2.1 Sujetadores

La modificación en el número, tamaño y geometría de los sujetadores no fue abordada en este trabajo. Se seguirá respetando el contar con cinco sujetadores: el superior, el posterior, el anterior, y los últimos dos de igual geometría, el lateral y medial. El tamaño y geometría no sufrieron modificaciones debido a que no afectan considerablemente al diseño de la instrumentación del socket.

La única modificación que sufrieron los sujetadores, es en la forma de monitorear y regular la presión en ellos. Con el fin de reducir el peso agregado al socket y evitar aumentar espacio en este debido a la cantidad de servoválvulas y sensores existentes, se decidió no controlar en pares a los sujetadores. En este trabajo se propone controlar a los sujetadores en conjunto, haciendo que cada uno de ellos esté conectados para el flujo de aire. Mediante esta configuración, la presión será la misma en todos los sujetadores por lo que únicamente se necesitará un solo sensor de presión y una sola servoválvula. La posible desventaja que puede presentar esta conexión entre los sujetadores es la distribución del aire dentro de éstos, la cual no sería igualmente uniforme como lo es al trabajar en pares. Sin embargo, el consumo de energía se reduce significativamente, pues el servomotor de cada una de las servoválvulas es responsable de un alto porcentaje de dicho consumo. Esto permite alargar la vida de las baterías del socket. Además, el nivel de sonido provocado por los servomotores también se reduce. Por otro lado, el peso del socket también es reducido al eliminar la masa de un servomotor, una válvula de tres vías, un sensor de presión, los conductos necesarios para la conexión de estos elementos con los sujetadores y los cables para el sensor.

En esta nueva configuración propuesta, el sensor de presión está conectado al sujetador lateral, mientras que, para la regulación del flujo de aire, la servoválvula está conectada al

sujetador medial. Posteriormente, en este capítulo, se muestra el CAD propuesto del socket, en el que se muestra la conexión de los sujetadores con el sensor de presión MPX5050DP y la servoválvula, además del acomodo de los demás componentes.

3.2.2 Servoválvula y conductos para el flujo de aire

La válvula de tres vías es la misma que se implementó en el diseño de autoajuste anterior del socket [23], debido a que cumple eficientemente con la función de regular el flujo de aire y su tamaño es adecuado para el espacio interno disponible en el socket. La instrumentación del socket no se enfoca en modificar o diseñar una válvula nueva, se deja a consideración de trabajos posteriores la selección o el diseño de una válvula con mejores características que la actual.

La servoválvula del presente trabajo únicamente se ve modificada en el uso de un nuevo servomotor (SG90). El ensamble entre este último y la válvula de tres vías se logra al introducir el engrane externo del servomotor en el hueco del mango de la válvula.

Existen tres estados principales de la servoválvula para poder implementar el control de la presión, los cuales son el bloqueo de aire en cualquier dirección, la entrada de aire hacia los sujetadores y, la salida de aire desde los sujetadores. Para poder definir la posición angular de la servoválvula que permitiese concordar con los estados mencionados, se decidió comprobar experimentalmente el ángulo de apertura de cada una de las vías en la válvula. En el trabajo previo realizado [23] se sugirió que el ángulo máximo óptimo era de 27°; sin embargo, mediante diferentes valores de posicionamiento del servomotor, se comprobó que el ángulo en el que se comienza a lograr el flujo de aire es a los 25°. Es por ello que en este trabajo se utilizó esa posición angular como referencia para la programación del control. En la siguiente figura se muestra la válvula de tres vías con las zonas en las que cada una de las vías permite el flujo de aire, o lo limita.

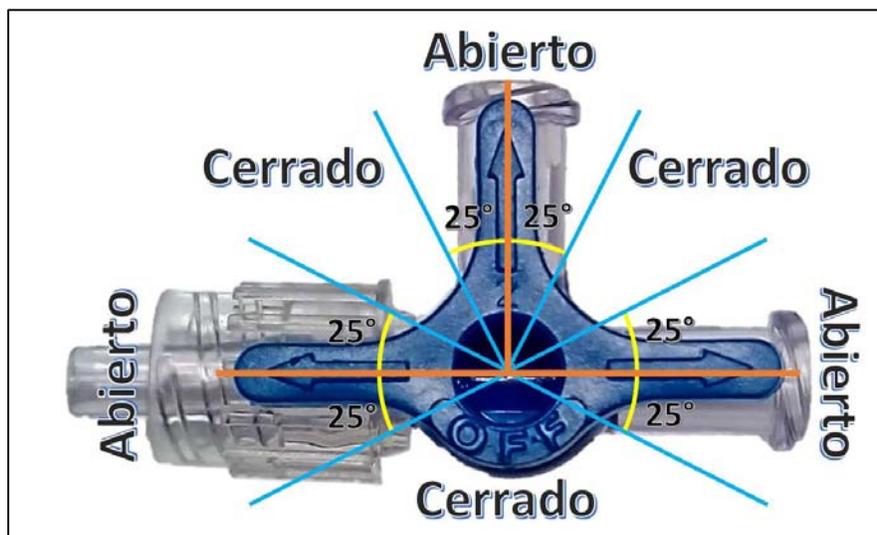


Figura 3.8. Zonas de apertura de la válvula de tres vías.

Como ya se mencionó previamente en este capítulo, la posición angular de la servoválvula se regula mediante el ciclo de trabajo del servomotor. Mediante la experimentación con el SG90, se pudo definir los valores para los que éste pueda tener una posición entre cero y 180°. El valor del ciclo de trabajo del PWM para que la posición del servomotor fuese de 0° resultó ser de 575 μ s, mientras que para alcanzar la posición de 180°, el ciclo de trabajo debe de durar 2375 μ s. Estos datos no son los mismos que se especifican en la hoja de datos del servomotor; sin embargo, son muy cercanos a ellos, y logran alcanzar la posición deseada con una alta precisión.

Los tres estados definidos de la servoválvula se logran posicionando al mango de la válvula de diferente manera. La primera posición es la de bloqueo, que se logra al cerrar la comunicación entre la entrada de aire por parte de la bomba y el conducto hacia los sujetadores, como se muestra en la siguiente figura. La posición angular del servomotor para poder bloquear el flujo de aire hacia o desde los sujetadores, se obtiene al fijar al ciclo de trabajo del PWM con 1475 μ s. En esta posición, la bomba de aire permanece desactivada.

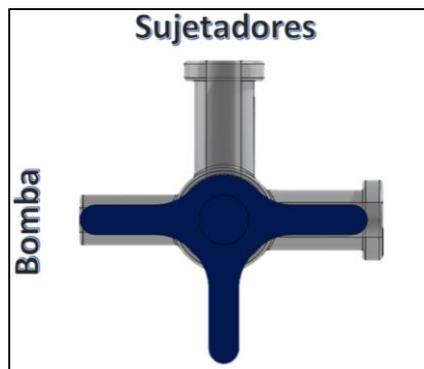


Figura 3.9. Posición de Bloqueo de la servoválvula.

La segunda posición de la servoválvula es la entrada de aire hacia los sujetadores. Esta puede tomar diversos valores en el rango de 25° desde el momento en el que el área del mango de la válvula se interseca con el área de la base de la válvula. El flujo de aire hacia los sujetadores comienza después de haber girado 65° al mango (mediante el servomotor) en dirección de las manecillas del reloj, a partir de la posición de bloqueo, como se muestra en la siguiente figura. Con esta posición se logra el flujo de aire mínimo hacia los sujetadores. El servomotor puede alcanzar esta posición, fijando la duración de su ciclo de trabajo en 2125 μ s.

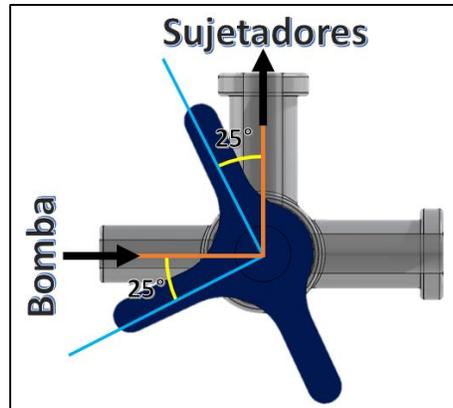


Figura 3.10. Posición de mínima entrada de aire hacia los sujetadores.

El flujo máximo de entrada de aire hacia los sujetadores se logra cuando se gira el mango los restantes 25° (figura siguiente) en dirección de las manecillas del reloj. En esa trayectoria el flujo de aire aumentará linealmente como se demostró en el diseño del sistema de autoajuste realizado previamente [23]. Esta posición se obtiene al fijar el ciclo de trabajo del PWM para el servomotor, con 2375 μ s. Por lo tanto, el flujo de entrada de aire hacia los sujetadores puede regularse con el ciclo de trabajo desde 2125 hasta 2375 μ s, con lo que se obtiene un rango de 250 μ s. Cabe señalar que durante la entrada de aire, la bomba se mantiene activada.

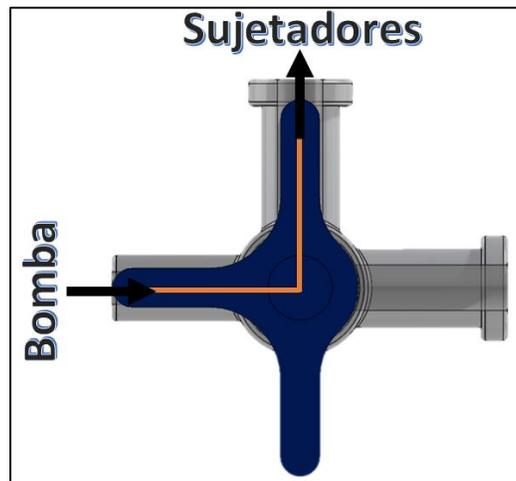


Figura 3.11. Posición de máxima entrada de aire hacia los sujetadores.

La tercera posición de la servoválvula es la salida de aire desde los sujetadores. El flujo mínimo de salida de aire se logra al girar 65° el mango en sentido contrario de las manecillas del reloj, a partir de la posición de bloqueo, como se observa en la figura siguiente. El valor del ciclo de trabajo del PWM del el servomotor para lograr esta posición es de 825 μ s. En esta posición no se activa la bomba neumática.

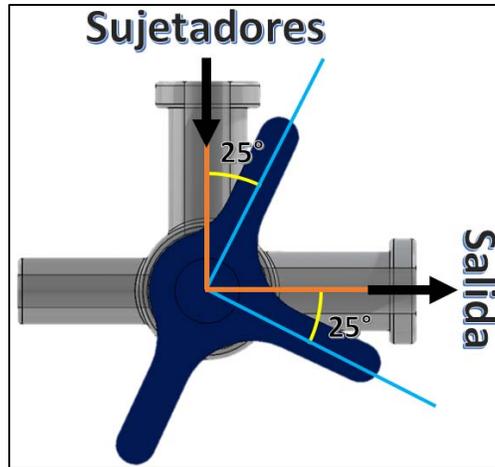


Figura 3.12. Posición de mínima salida de aire desde los sujetadores.

El flujo máximo de salida de aire desde los sujetadores hacia el exterior o medio ambiente, se obtiene al girar al mango 25° más en sentido contrario de las manecillas del reloj (figura siguiente). El valor de la duración del ciclo de trabajo para el PWM es de $575 \mu\text{s}$. Esto genera un rango de $250 \mu\text{s}$ desde $825 \mu\text{s}$ para el flujo mínimo de salida hasta los $575 \mu\text{s}$ de flujo máximo. Se debe de tomar en cuenta que el ciclo de trabajo puede tomar cualquier valor dentro de este rango, dependiendo de la acción del controlador PI.

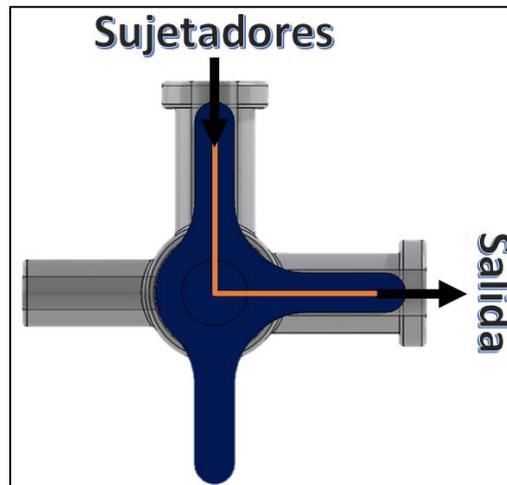


Figura 3.13. Posición de máxima salida de aire desde los sujetadores.

Se debe señalar que la válvula de tres vías debe ser modificada para poder colocarla dentro del socket. La modificación principal es la eliminación de los tres brazos del mango de la válvula (en color azul). Esto es con el fin de que pueda girar libremente con ayuda del servomotor. En las figuras anteriores, que muestran las posiciones de la válvula, se dejaron los brazos, con el fin de que se entienda con mayor facilidad los movimientos de la servoválvula.

Los conductos de conexión entre la servoválvula, los sujetadores y la bomba seleccionados son los mismos que se utilizaron en el trabajo previo del diseño del sistema de autoajuste [23]. Se trata de mangueras de uso médico de látex de 3 mm de diámetro interno y 4 mm de diámetro externo. La decisión de mantener el mismo tipo de conductos fue debido a la facilidad que tienen para acoplarse a los distintos elementos del sistema neumático; además, permitirán acomodarse sin problema en el socket gracias a su flexibilidad.

3.2.3 Bomba neumática

La bomba para el sistema de autoajuste del socket (figura siguiente) se seleccionó tomando en cuenta las necesidades planteadas previamente en este trabajo. Se trata de una bomba de diafragma (alternativa y de desplazamiento positivo). Es de origen chino, de la marca OEM que cuenta con la aprobación de la norma ISO 9001. Sus características son las siguientes:

- Presión máxima: 65 kPa (487 mmHg)
- Dimensiones: 57 mm (largo), 27 mm (diámetro)
- Nivel de sonido: 35 dB
- Alimentación: 6 V
- Consumo: 300 mA

Estas características permiten tener una bomba neumática más eficiente que la que se utilizó previamente en el sistema de autoajuste [23]. Alcanza una presión mayor que la anterior bomba, al ser alimentada a 6 V. Por otra parte, presenta unas dimensiones menores a su antecesora. Y uno de los parámetros más importantes, que es el nivel de sonido, es inferior casi en un 50 %, de 60 dB a 35 dB, aproximadamente.

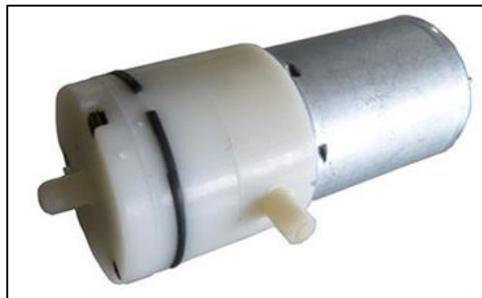


Figura 3.14. Bomba neumática.

3.3 Control del sistema de autoajuste

Para el correcto funcionamiento del sistema autoajutable del socket con la instrumentación, es necesario analizar el control a ser implementado en el microcontrolador. En este caso se deben de realizar los ajustes en cada etapa del diseño del controlador, dependiendo de los elementos que se tengan en el sistema a controlar. Este sistema incluye, como ya se mencionó previamente, los sujetadores, la servoválvula, la bomba y el sensor de presión. Debido a que se realizaron ciertos cambios respecto al trabajo

previo [23] realizado del sistema de autoajuste, los valores del controlador PI presentan cambios.

3.3.1 Ajuste en la planta del sistema de autoajuste

La función de transferencia de la planta total del socket es la relación entre la presión en los sujetadores (en pascales) y la posición angular del servomotor de la servoválvula (en grados).

$$H(s) = \frac{P(s)}{A(s)}$$

Como se mencionó con anterioridad, esta función de transferencia resultó ser la siguiente [23]:

$$H(s) = \frac{1319.86}{s}$$

Esta planta fue obtenida a partir de tres funciones de transferencia intermedia:

$$H(s) = H_{Pq}(s)H_{qA}(s)H_{A\theta}(s)$$

En el caso de la función de transferencia entre el área de flujo de aire y la posición angular del servomotor, esta no sufrirá cambios. Esto es debido a que la llave de tres vías seleccionada en la servoválvula, es la misma que la que se utilizó en el análisis de esta función de transferencia. Así que la relación entre estas dos variables se mantiene igual.

$$H_{A\theta}(s) = \frac{A}{\theta} = 1 \times 10^{-7}$$

La función de transferencia entre el área de flujo y el área de paso de aire tampoco se verá afectada, pues los valores de los que ésta depende, se mantienen iguales en el nuevo diseño. El coeficiente de descarga tiene el mismo valor unitario, y la temperatura de entrada del sistema es la misma de 289 K, considerando condiciones ideales.

$$H_{qA}(s) = 286.74$$

Por otro lado, la relación entre la presión en los sujetadores y el flujo de aire sí sufre un cambio. Esto se debe a que el volumen considerado en este trabajo, es diferente al que se tomó en el trabajo previo [23]. Este cambio, genera que la capacitancia neumática en los sujetadores se modifique.

Mediante ayuda de un software de CAD, se obtuvo el volumen de los sujetadores diseñados en el trabajo “Diseño de un socket autoajustable para prótesis de miembro inferior” [44]. El volumen interno del total de los sujetadores fue:

$$V = 2.2343 \times 10^{-3} m^3$$

Este volumen se sustituyó en la fórmula de la capacitancia neumática, con los otros valores definidos previamente, para obtener lo siguiente:

$$C = \frac{V}{n * R_{aire} * T} = \frac{2.2343 \times 10^{-3}}{(1) * (287) * (310.5)} \frac{kg}{N/m^2}$$

$$C = 25.0729 \times 10^{-9} \frac{kg}{N/m^2}$$

Así que, la función de transferencia entre la presión y el flujo de aire resulta:

$$H_{Pq}(s) = \frac{1/C}{s} = \frac{1/25.0729 \times 10^{-9}}{s} = \frac{3.9884 \times 10^7}{s}$$

Para facilitar el manejo de la presión, se manejarán las unidades en milímetros de mercurio, por lo que la función de transferencia anterior queda de la siguiente manera.

$$H_{Pq}(s) = \left(\frac{3.9884 \times 10^7}{s} \right) \left(\frac{1 \text{ mm Hg}}{133.322 \text{ Pa}} \right)$$

$$H_{Pq}(s) = \frac{299156.8803}{s}$$

Este cambio, genera una modificación en la función de transferencia total de la planta:

$$H(s) = \left(\frac{299156.8803}{s} \right) (286.74) (1 \times 10^{-7})$$

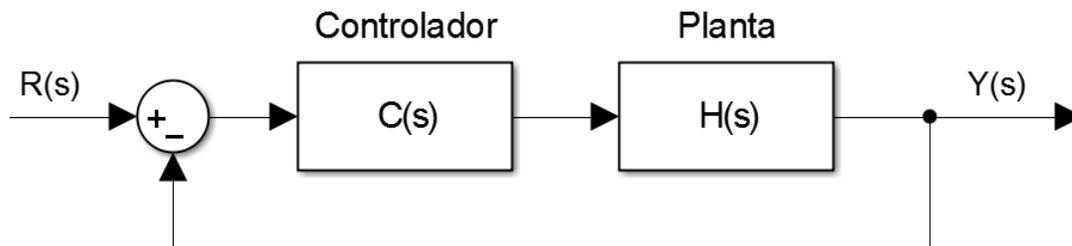
$$H(s) = \frac{8.578}{s}$$

3.3.2 Ajuste en el controlador del sistema de autoajuste

Como ya se mencionó previamente, un controlador PI es suficiente para que el control del sistema de autoajuste funcione adecuadamente. Este tipo de controlador incrementa el porcentaje de sobrepaso, el tiempo de asentamiento, y elimina el error en estado permanente.

Es importante que exista cierto porcentaje de sobrepaso, ya que el sistema del socket es un sistema neumático. Cuando existe un error entre la referencia de presión y la presión en los

sujetadores, el control actúa aumentando el flujo de aire hacia los sujetadores, hasta alcanzar la presión de referencia. Si no se presentara el sobrepaso, significaría que se llega al nivel de presión sin tener un valor mayor a la presión de referencia. Sin embargo, debido al material de los sujetadores, que presenta cierta elasticidad, la presión alcanzada comienza a disminuir lentamente, provocando que se aplique el control de manera constante hasta alcanzar el valor deseado nuevamente. Es por eso que un sobrepaso permitiría hacer más estable al sistema de control. Idealmente el volumen de los sujetadores es constante; no obstante, pueden llegar a presentar un bajo porcentaje de deformación elástica. Luego de que los sujetadores se deforman momentáneamente, disminuyen de manera natural la presión de los sujetadores. Al existir un sobrepaso, la presión excedente comenzará a disminuir, no solo por la acción del controlador, sino también por la naturaleza de los sujetadores. Esto reduce el consumo energético considerando que se llegaría al valor deseado de presión de referencia más rápidamente.



En el diagrama superior, se encuentra el control del sistema de autoajuste, en el que se tienen las siguientes definiciones:

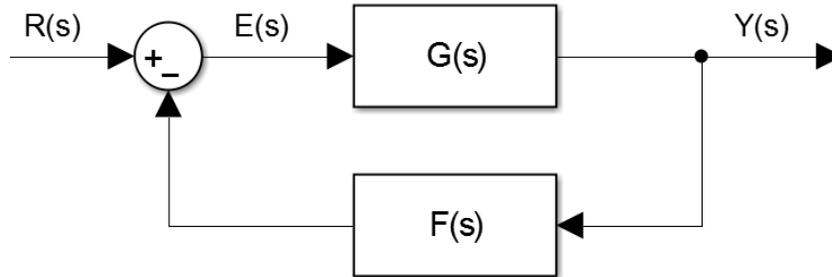
- $R(s) = \text{Presión de referencia del socket en mmHg}$
- $C(s) = k_p + \frac{k_i}{s}$ Control PI
- $H(s) = \frac{8.578}{s}$ Función de transferencia de la planta
- $Y(s) = \text{Presión de los sujetadores (salida del sistema)}$

El rango de valores permitidos para la referencia de la presión del sistema es de 70 a 90 mmHg, por lo que es más sencillo trabajar en esas unidades que con pascales. Así es que en la programación del microcontrolador, el error obtenido es la diferencia entre la presión de referencia y la presión medida, en milímetros de mercurio. A diferencia del diseño anterior [23], que utilizó la diferencia de voltajes como error de entrada al controlador, en este trabajo se usa la diferencia de presiones del sistema. Esto lo convierte en un control más confiable pues compara directamente la salida y referencia de presiones en vez de hacerlo indirectamente.

Para comprobar la estabilidad del sistema de control, se puede utilizar el método del Lugar Geométrico de las Raíces. Este método permite modificar la planta del sistema para tener una respuesta deseada. Mediante este método se puede observar el comportamiento de los polos de lazo cerrado cuando se modifica una constante de proporcionalidad. El

movimiento de estos polos en el plano complejo se le conoce como el Lugar Geométrico de las Raíces. El primer paso es obtener la ecuación característica de la función de transferencia de lazo cerrado.

Para obtener la función de transferencia de lazo cerrado, se debe reducir el diagrama bloques retroalimentado, como el que se muestra a continuación.



Como se muestra en el diagrama de arriba, la salida y el error, se definen de la siguiente manera:

$$Y(s) = G(s)E(s)$$

$$E(s) = R(s) - F(s)Y(s)$$

Al sustituir $E(s)$ en la primera ecuación:

$$Y(s) = G(s)R(s) - G(s)F(s)Y(s)$$

$$Y(s) = \frac{G(s)}{1 + G(s)F(s)} R(s)$$

Así que, la función de transferencia de lazo cerrado es:

$$G_{LC}(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)F(s)}$$

Debido a que en el sistema del control del socket $G(s) = C(s)H(s)$, la función de transferencia de lazo cerrado se puede definir de la siguiente manera:

$$G_{LC}(s) = \frac{C(s)H(s)}{1 + C(s)H(s)}$$

La ecuación característica de lazo cerrado es el denominador de la función de transferencia de arriba igualado a cero:

$$1 + C(s)H(s) = 0$$

Las raíces de esta ecuación, son los polos de lazo cerrado, y mediante el LGR (Lugar Geométrico de las Raíces), se pueden observar aquellos que se encuentren en el lado estable del plano complejo. La parte real de estos polos debe ser negativa para asegurar que el sistema permanezca siendo estable.

Al sustituir los valores de las funciones de transferencia $C(s)$ y $H(s)$ se obtiene la siguiente ecuación característica.

$$1 + \left(k_p + \frac{k_i}{s}\right) \left(\frac{8.578}{s}\right) = 0$$

Si se reacomoda el controlador:

$$1 + k_p \left(\frac{s + k_i/k_p}{s}\right) \left(\frac{8.578}{s}\right) = 0$$

Como se puede observar, al ser un controlador PI, se tienen dos constantes sin definir; sin embargo, el método del LGR funciona mediante la variación de solo una constante proporcional. Es por ello que habrá que definir el valor de k_i para que se puede manejar k_p como la constante del método.

Un controlador PI proporciona un cero y un polo al sistema. Si se fija un valor al cero agregado al sistema, se puede proceder a realizar el método del LGR. Una manera de hacerlo es mediante el tiempo de asentamiento T_s (tiempo que tarda el sistema en alcanzar el valor de referencia dentro de una tolerancia). Para un 2% de tolerancia, el valor aproximado del tiempo de asentamiento tiene la siguiente fórmula:

$$T_s = \frac{4}{P_1}$$

En donde P_1 es el polo dominante de la ecuación característica. Este polo debe estar en la parte negativa de los valores reales en el plano complejo y ser el más cercano al origen, similar a como se planteó previamente en el diseño del socket [23]. Además, debe de estar lo suficientemente alejado de los demás polos.

La respuesta de los sistemas neumáticos es más lenta que la de sistemas electrónicos y eléctricos, por lo que el tiempo de asentamiento planteado para este trabajo se planteará de un segundo, así que el polo debe tener un valor cercano a -4. Para lograr esto, se debe recordar que los polos del LGR siempre tienden a algún cero. Entonces, para el diseño del controlador, se definirá un cero en -4, con la finalidad de que el polo dominante se acerque a este valor y tener un tiempo de asentamiento cercano a un segundo.

El cero ubicado en -4 se puede definir a través del controlador PI. Si se reacomoda la función de transferencia del controlador, se puede observar, que se aporta un polo en el origen y un cero en $-k_i/k_p$ al sistema de control.

$$C(s) = k_p + \frac{k_i}{s} = \frac{k_p s + k_i}{s} = \frac{k_p(s + k_i/k_p)}{s}$$

De esta manera, al sustituir $k_i/k_p = 4$ en la anterior ecuación, con el fin de agregar al cero en -4, aportado por el controlador PI, se obtiene lo siguiente:

$$C(s) = \frac{k_p(s + 4)}{s}$$

Para seleccionar el valor de la constante, se tomó en cuenta que el sistema, al ser de primer orden y agregarle el control PI, se convierte en un sistema de segundo orden. Esto se puede observar al sustituir los valores de $C(s)$ y $H(s)$ en la función de transferencia de lazo cerrado $G_{LC}(s)$, en donde se obtiene lo siguiente:

$$G_{LC}(s) = \frac{8.578(k_p s + k_i)}{s^2 + 8.578k_p s + 8.578k_i}$$

Aquí se puede observar que el sistema en lazo cerrado es de segundo orden, y su polinomio característico es el siguiente:

$$s^2 + 8.578k_p s + 8.578k_i$$

Este nuevo sistema tiene un polinomio característico de segundo orden como el que se muestra a continuación [77]:

$$s^2 + 2\zeta\omega_0 s + \omega_0^2$$

En el que ζ es el factor de amortiguamiento, el cual determina la forma que tendrá la respuesta del sistema. Y ω_0 es la frecuencia natural del sistema, la cual ayuda a la velocidad de respuesta. Al comparar los coeficientes de los dos anteriores polinomios, se puede parametrizar a las constantes del control PI de la siguiente manera [77]:

$$k_p = \frac{2\zeta\omega_0}{8.578} \dots \dots \dots (A)$$

$$k_i = \frac{\omega_0^2}{8.578} \dots \dots \dots (B)$$

En este sistema de control de presión se necesita que la respuesta del sistema sea subamortiguada ($0 < \zeta < 1$) para que la presión del sistema pueda presentar un sobrepaso, cuya razón fue explicada en la sección 3.3.2. Es por ello, que se buscará que las constantes del controlador queden parametrizadas en función del factor de amortiguamiento. Para lograr esto, se sabe que uno de los ceros se ubica en -4, y que en consecuencia se tiene la relación $k_i/k_p = 4$ planteada previamente. De esta última relación, se puede despejar k_i y sustituirla en la ecuación B:

$$4k_p = \frac{\omega_0^2}{8.578}$$

Si se divide entre 4 la anterior ecuación, se obtiene una nueva ecuación de la constante k_p , la cual puede ser igualada con la ecuación A:

$$\frac{\omega_0^2}{(4)(8.578)} = \frac{2\zeta\omega_0}{8.578}$$

Al despejar a la frecuencia natural del sistema se obtiene $\omega_0 = 8\zeta$. Por lo que, ahora las constantes del controlador PI pueden ser parametrizadas en función del factor de amortiguamiento al sustituir la anterior ecuación en A y en B.

$$k_p = \frac{16\zeta^2}{8.578} \quad k_i = \frac{64\zeta^2}{8.578}$$

Dado que para que exista un sobrepaso, el factor de amortiguamiento debe estar entre cero y uno, se asignará un valor que permita que el sistema sea estable. Mediante el método del LGR, se comprobará la estabilidad, dejando como constante a k_p con diferentes valores de ζ . Con ayuda de la herramienta rltool de Matlab, se agregará el cero en -4 y un integrador ubicado en el origen para generar al controlador PI. La función de transferencia del controlador es la siguiente, en la que la constante C es igual a $4k_p$, como se muestra en la siguiente figura:

$$C(s) = \frac{k_p(s + 4)}{s} = \frac{C(1 + 0.25s)}{s}$$

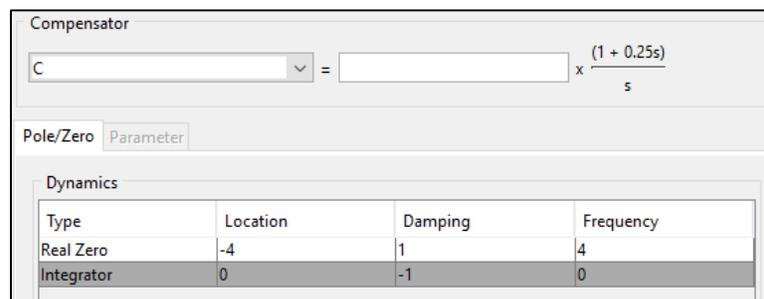


Figura 3.15. Utilización de la herramienta rltool de Matlab.

Para un factor de amortiguamiento de cero y uno, se tienen las siguientes constantes k_p respectivamente:

$$k_p(0) = 0 \quad k_p(1) = 1.8652$$

Por lo tanto, el la constante k_p debe estar arriba de de cero y menor que 1.8652 para una respuesta subamortiguada. La siguiente gráfica muestra el LGR, en el que se observan las trayectorias que toman los dos polos en lazo cerrado, los cuales siempre están en el lado negativo del plano complejo. Uno de los polos tiene una trayectoria que tiene a infinito negativo (azul) y el otro polo tiene al cero en -4 (rojo).

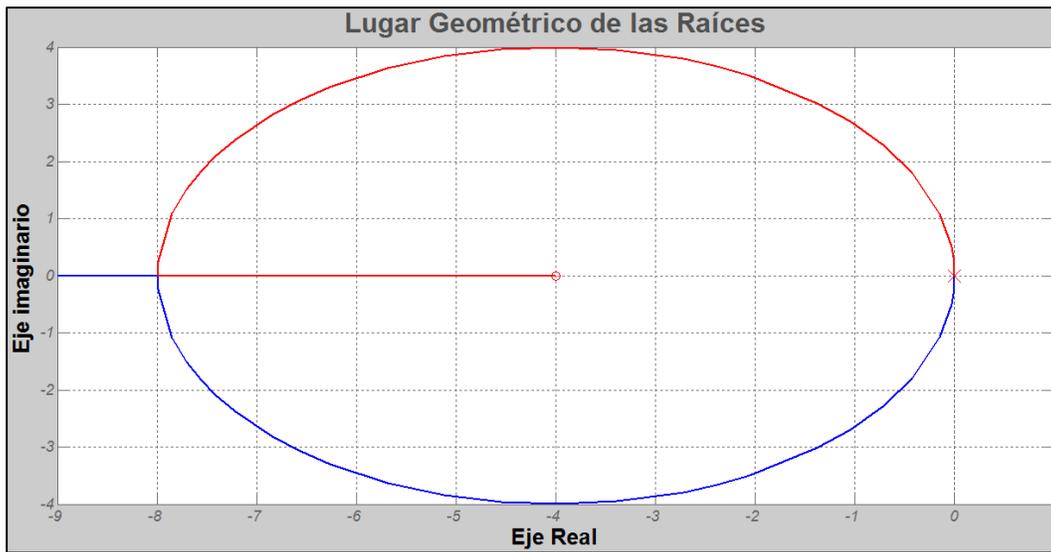


Figura 3.16. LGR mediante la herramienta rltool de Matlab.

Con esto se concluye que el sistema es estable ante cualquier valor de la constante k_p . Sin embargo, al ser un sistema subamortiguado, los polos son complejos conjugados, lo que significa que no se podrá lograr tener un tiempo de asentamiento de un segundo exactamente. Así que se buscará un valor de factor de amortiguamiento alto, para intentar que el polo de la trayectoria en color rojo, se acerque lo más posible a -4. Dicho valor se fijará en 0.95 y los resultados de las constantes se muestran a continuación.

$$k_p = 1.6834 \quad k_i = 6.7335$$

El LGR, se muestra a continuación, (obtenido mediante la herramienta rltool), en el que se observan los dos polos de la función de transferencia de lazo cerrado, para los valores obtenidos de k_p y k_i . Los dos polos de lazo abierto se encuentran en el origen. Además, se muestran los dos polos en lazo cerrado obtenidos a partir de definir las constantes proporcional e integral del controlador.

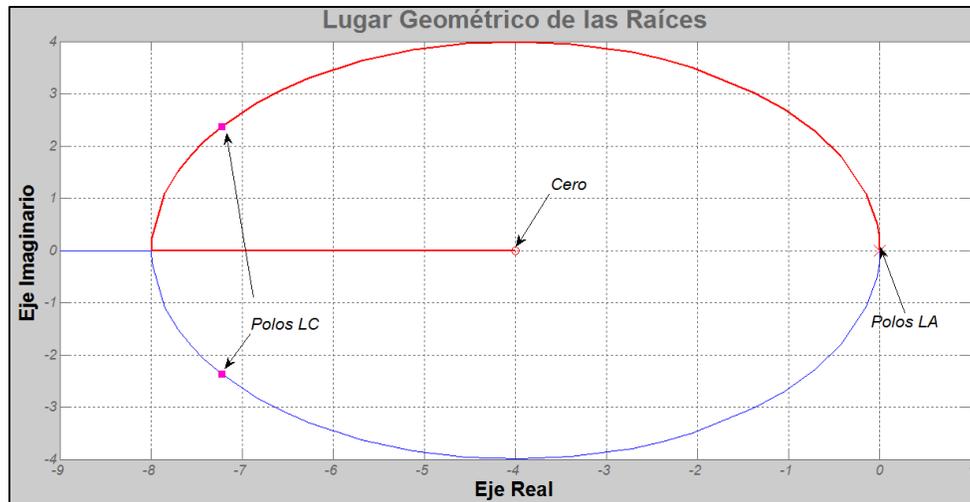


Figura 3.17. Polos de lazo abierto mostrados en el LGR mediante la herramienta rtool.

Para comprobar que el sistema de control presenta un sobrepaso, se utilizó la herramienta Simulink de Matlab, utilizando los siguientes bloques:

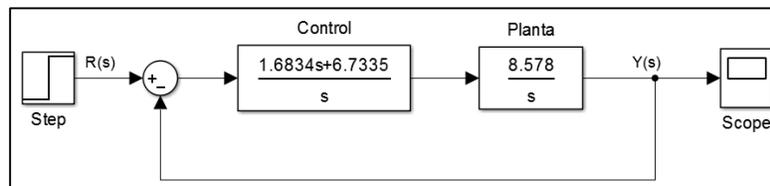


Figura 3.18. Diagrama de bloques del control del socket mediante Simulink.

La referencia del sistema se definió como una función escalón con un valor final de 80 mmHg, similar a como se realizó en el diseño previo [23]. En la siguiente figura, se muestra la respuesta de la presión en los sujetadores, al aplicarle el control PI. Se puede observar en la gráfica que se presenta un sobrepaso menor a 100 mmHg, y que el tiempo de asentamiento (desde cero hasta alcanzar el valor de 80 mmHg) es cercano a un segundo, como se había diseñado el controlador.

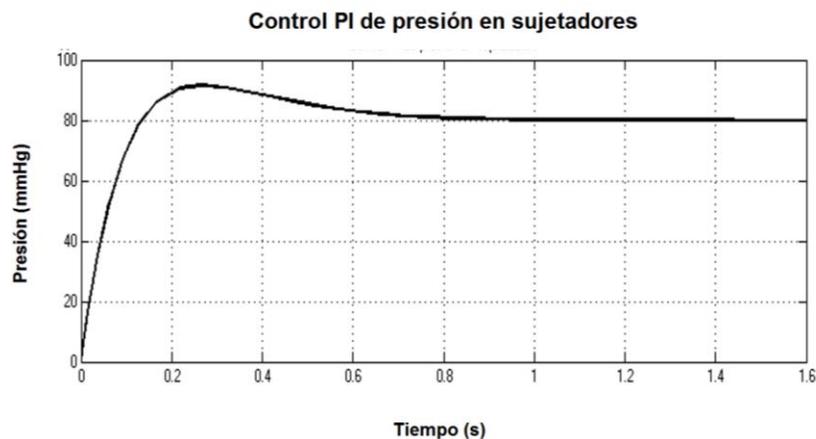


Figura 3.19. Comportamiento de la presión con el control PI.

3.3.3 Discretización del control del sistema de autoajuste

La discretización es el proceso por el cual se obtiene un sistema en tiempo discreto a partir de su forma representada en tiempo continuo.

El control diseñado, se implementó en el microcontrolador PIC16F887. Para ello se realizó la discretización del sistema de control. Debido a que el microcontrolador es un elemento digital, la señal del sensor de presión debe ser muestreada y además, retenida. El voltaje obtenido desde el sensor, es una señal en tiempo continuo; sin embargo, a través del microcontrolador, y el convertidor analógico digital, se transforma esa señal en un tren de pulsos cada cierto tiempo de muestreo. La retención de estos valores implica convertir la señal de tiempo discreto (tren de pulsos), en una señal de tiempo continuo. Esto se logra mediante un retén de orden cero (ROC), el cual mantiene el valor del pulso hasta el siguiente muestreo.

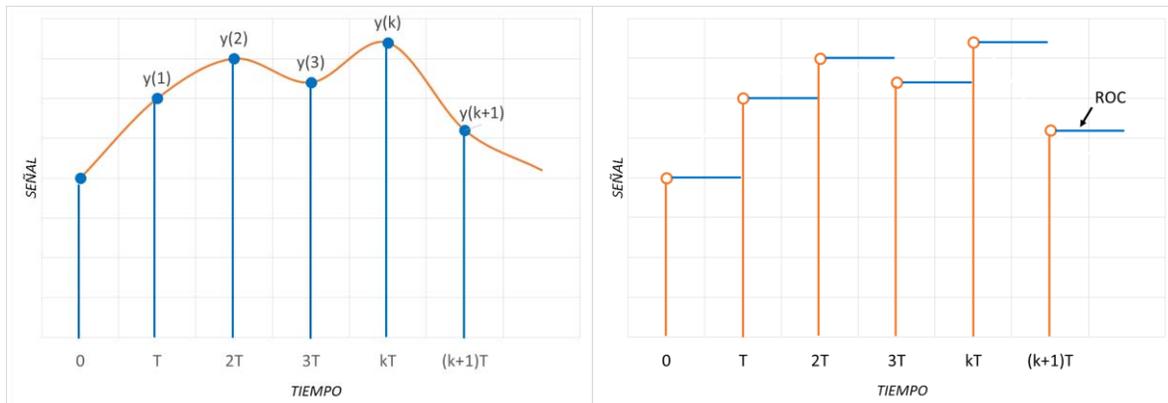


Figura 3.20. Discretización de una señal en tiempo continuo.

Como se observa en la figura anterior, la señal es muestreada cada T , por lo que el tiempo ahora se representa como $t = kT$, donde k es el número de muestra y T es el tiempo de muestreo. Esta manera de discretización del sistema, presenta facilidad en el cálculo del controlador; sin embargo, si el tiempo de muestreo T no es suficientemente corto, se pueden degradar la respuesta y la aproximación obtenida.

Para que el microcontrolador pueda implementar al controlador PI del sistema de autoajuste. Cuando la referencia $R(t)$ del sistema es fijada a un cierto valor, y la presión de los sujetadores lo alcanza, cualquier variación en la presión diferente causará un error. El controlador implementado en el dominio del tiempo tiene la siguiente ecuación.

$$C(t) = k_p E(t) + k_i \int_0^t E(t) dt$$

En este sistema, la señal que produce la salida del controlador, determinará la posición de la servoválvula. Sin embargo, directamente en el microcontrolador, esta señal del

controlador define la magnitud del ciclo de trabajo del módulo PWM, para controlar la posición angular del servomotor.

La presión del sistema es medida a través del sensor MPX5050DP, que entrega una señal analógica de voltaje. Este voltaje tiene una relación proporcional con la presión manométrica. Al estar alimentado con 5V, presenta una respuesta muy cercana a ser lineal, como se ha mencionado previamente. Además, presenta un offset que puede ser despreciado, y asumir que parte desde el origen, como se realizó en el diseño previo [23]. La señal de voltaje analógica que entrega el sensor, es procesada por el convertidor analógico digital del microcontrolador PIC16F887. Posteriormente ese valor es transformado a milímetros de mercurio, con base en la función de transferencia de la hoja de datos y en la definida en el trabajo anterior [23] del sistema de autoajuste.

Para poder implementar el controlador, se deben realizar ciertas aproximaciones. El primer paso que se debe realizar, para lograr la discretización, es aproximar la parte integral mediante de la siguiente ecuación.

$$\int_0^t E(t)dt \approx T_m \sum_0^N E(n)$$

En la que $E(n)$ es el error actual, y T_m es el tiempo de muestreo. Con esta aproximación, se puede definir a $C(t)$ como lo siguiente:

$$C(n) = k_p E(t) + k_i T_m \sum_0^N E(n)$$

O de la siguiente manera:

$$C(n) = k \left(E(t) + \left(\frac{T_m}{T_i} \right) \sum_0^N E(n) \right)$$

En donde $k = k_p$ y además, $k_i = k/T_i$

En la siguiente figura, se muestra el mapa de flujo de la rutina que sigue el microcontrolador para implementar al control PI. Posteriormente se detallará cada parte con el pseudocódigo.

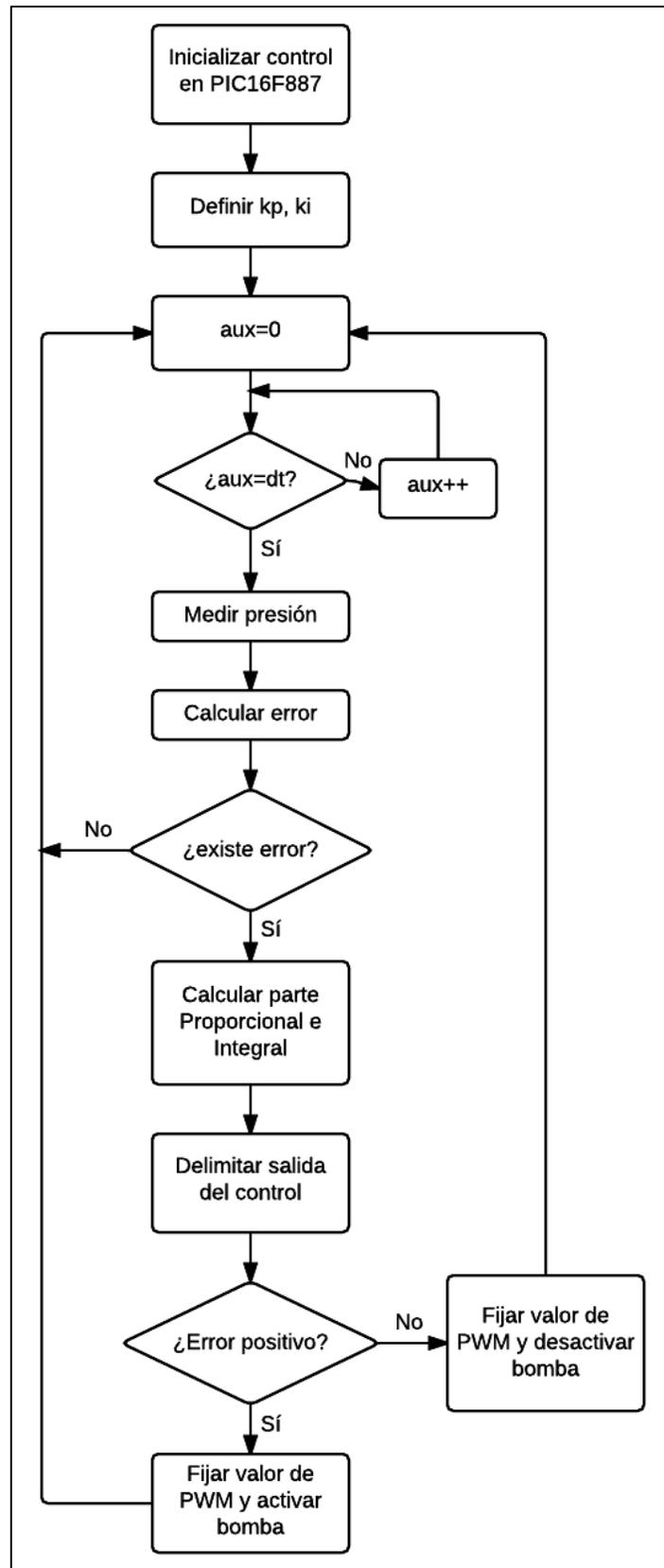


Figura 3.21. Mapa de flujo reducido de la rutina del control PI en el microcontrolador.

Pseudocódigo del control PI:

1. Configuración del microcontrolador. Se configuran los puertos de entrada y salida para el sensor de presión, el servomotor y la bomba. También se configuran el convertidor analógico digital para el tiempo de muestreo. Y finalmente se configura el módulo PWM para el servomotor.
2. Cargar en la memoria k_p y k_i .
3. Obtener el valor de referencia $R(k)$.
4. Obtener $Y(k)$ desde el ADC.
5. Convertir $Y(k)$ a presión en mmHg.
6. Calcular el error $E(k)$.
7. Calcular el control $C(k)$.
8. Escribir $C(k)$ en el ciclo de trabajo del PWM.
9. Esperar hasta el siguiente tiempo de muestreo T_m y regresar al punto 3.

En los anexos, se encuentra el código en lenguaje C desarrollado en el PIC16F887 para implementar el control PI.

3.4 CAD del socket protésico con la instrumentación

Para poder realizar un diseño asistido por computadora del socket protésico con los elementos de la instrumentación y el sistema de autoajuste, se utilizó el software SolidWorks.

El diseño del socket protésico con la instrumentación del socket se desarrolló tomando en cuenta los requerimientos planteados en el capítulo anterior. Este diseño pretendió permitirle al usuario contar con un mantenimiento sencillo, por lo que cada parte se colocó dentro del socket para que pudiese ser accedida con facilidad.

Los elementos del sistema electrónico fueron diseñados para poder ser removidos sin tener algún inconveniente y sin tener que dañar al socket para hacerlo.

La batería, el sensor de presión y la servoválvula se colocaron en la parte inferior, dentro del socket. El sensor de presión es fijado mediante dos barras en forma cilíndrica, ubicadas en la base del socket de 3 mm de diámetro para que pudiesen ajustarse a los agujeros del sensor. La servoválvula está fijada a la base del socket gracias a dos barras laterales separadas 22.5 mm, entre las que se coloca el servomotor. La batería es ubicada en el espacio restante, y puede ser fijada a la base mediante algún pegamento industrial.

En la siguiente figura se puede observar la posición de estos elementos en la base del socket.

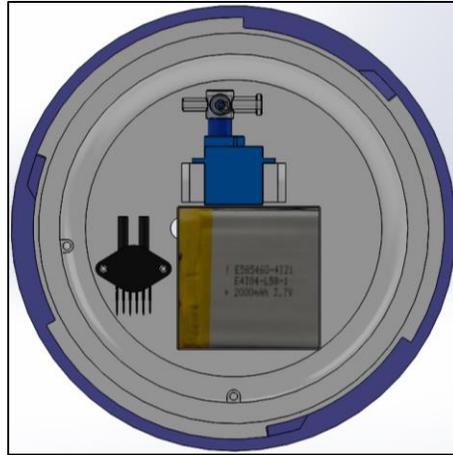


Figura 3.22. Vista de sección superior del socket con la ubicación de la servoválvula, el sensor de presión y la batería en la base del socket.

Posteriormente se coloca la bomba neumática por encima de la servoválvula. La fijación de esta última a la base, se logra mediante dos barras paralelas a las que sujetan al servomotor. La parte superior de estas barras tienen una curva con el radio externo de la bomba, esto facilita que no presente algún tipo de movilidad y se mantenga en una sola posición. La tarjeta del circuito con todos sus elementos en ella se une a la base mediante dos tornillos, como se puede observar en la siguiente figura.

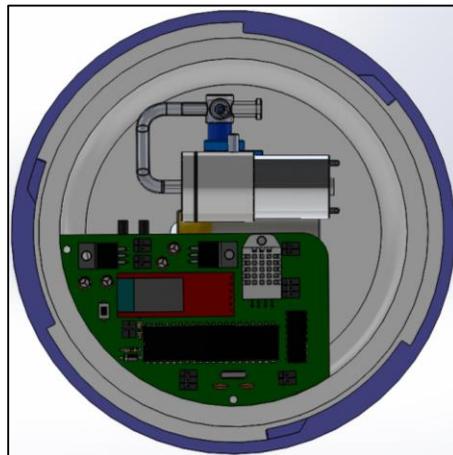


Figura 3.23. Vista de sección superior del socket con la ubicación de la bomba y el circuito.

Para poder realizar este acomodo, se debió modificar el espacio disponible en la base, el cual, originalmente no era suficiente para albergar a todos los elementos ya mencionados con anterioridad. En la tesis de licenciatura “Metodología de pruebas estáticas y dinámicas para un socket protésico de miembro inferior a nivel transfemoral”, se establece que la malla elástica del socket debe estar constituida por el PVC flexible Apex90NT. Esto permitiría que la malla sufriera un desplazamiento máximo no mayor a 18.4 mm, al aplicarle una carga de 981 N (considerando el peso de una persona cercano a 100 kg) [46]. Con base en esta consideración y al tamaño de los elementos internos del socket, se aumentó la

altura de la base 2.45 cm, lo que permite que la distancia entre la malla elástica y el elemento más cercano (la bomba neumática) sea cercana a los 28 mm. Tanto el cambio en la base como la distancia entre la malla y su elemento más cercano se muestran en las siguientes dos figuras.

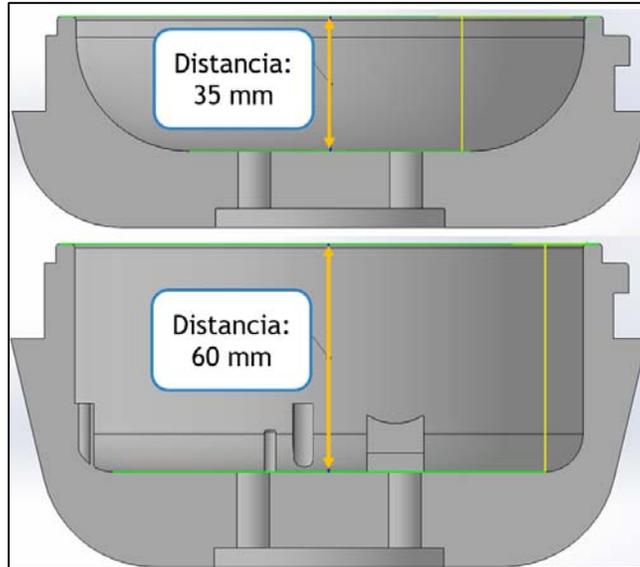


Figura 3.24. Aumento de 25 mm de largo en la base del socket.

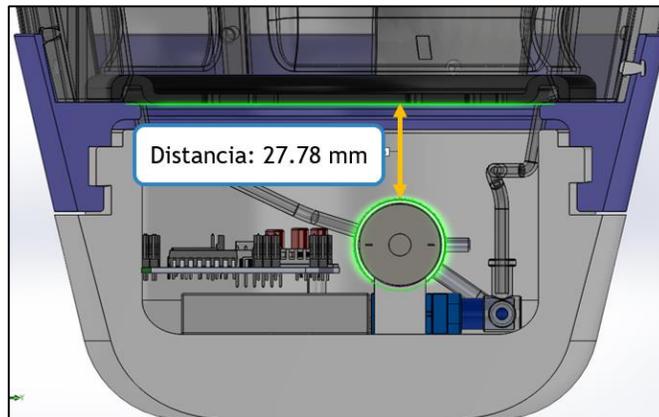


Figura 3.25. Distancia mínima entre la malla y el elemento más cercano en la parte inferior.

Finalmente el acomodo de la parte interna del socket con los elementos de cada sistema, se muestran en la siguiente figura, en la que también se observa las conexiones de los elementos neumáticos mediante los conductos para el flujo de aire. Estas conexiones son el sensor de presión con el sujetador lateral, la servoválvula con el sujetador medial y la bomba con la servoválvula.

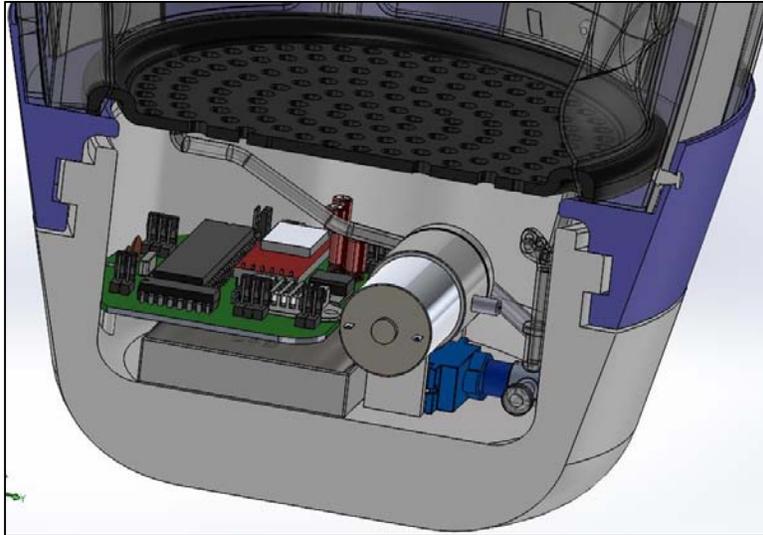


Figura 3.26. Vista de sección que muestra el acomodo de los elementos de cada sistema en la parte interna de la base del socket.

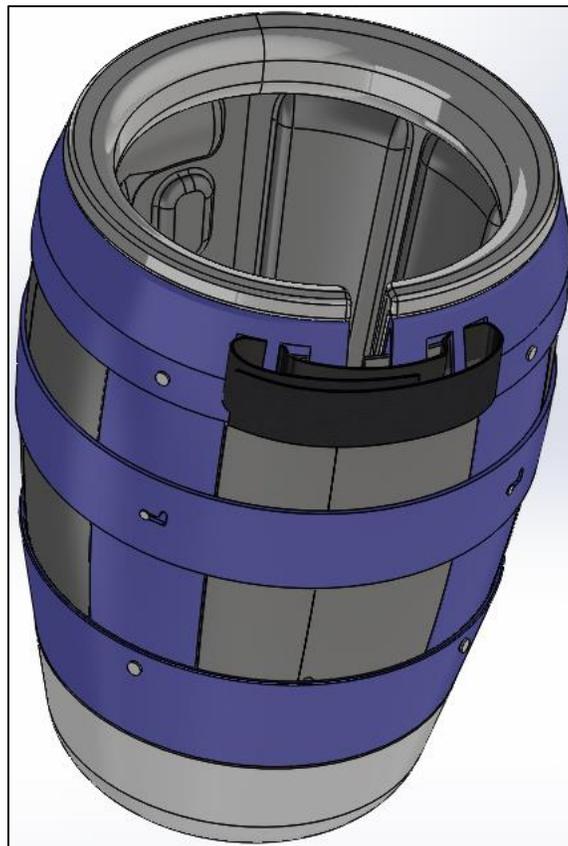


Figura 3.27. Diseño final del socket.

3.5 Programación de la interfaz de usuario

La aplicación para el socket protésico con su instrumentación fue diseñada con base en los objetivos y los requerimientos previamente planteados y analizados. La herramienta utilizada para la creación de esta interfaz fue MIT App Inventor, el ambiente de desarrollo de Android en la red del Instituto Tecnológico de Massachusetts (MIT por sus siglas en inglés).

El diseño de esta interfaz de usuario (aplicación) le permite al usuario visualizar los datos de temperatura, humedad y presión en los sujetadores de manera sencilla y ordenada. Por otro lado, el usuario puede comprobar el funcionamiento del control automático del sistema de autoajuste del socket, a través de una gráfica en tiempo real, que muestre los valores de la presión en los sujetadores comparados con la referencia que se haya fijado.

Además de las características mencionadas previamente, la aplicación permite almacenar los datos de temperatura, humedad y presión de una manera gráfica en la memoria del dispositivo móvil.

Posteriormente, mientras se muestre la disposición de la aplicación, se describirán las características y funciones que ésta posee.

3.5.1 Disposición del diseño de la interfaz

La visualización de la interfaz desarrollada en Android, está compuesta de elementos que le permiten al usuario interactuar con la aplicación. Una actividad en Android es cada una de las visualizaciones y que son comúnmente conocidas como “pantallas”. La actividad principal de la aplicación para el socket protésico, la primera vez que es instalada en el dispositivo móvil puede ser alguna de las siguientes:



Figura 3.28. Dos de las posibles primeras visualizaciones de la interfaz de usuario una vez instalada la aplicación en el dispositivo móvil.

Si no tiene activado el Bluetooth, se mostrará la visualización de la izquierda en la figura de arriba, y posteriormente aparecerá la visualización de la derecha, una vez activado el Bluetooth. Esta primera actividad consta de cinco botones, ordenados verticalmente. El primero de ellos es el botón iniciar (color azul oscuro), el cual permite cambiar a la siguiente actividad, en la que el usuario podrá decidir si desea consultar los datos de presión o de temperatura y humedad. Si el usuario da clic en este botón por primera vez, se mostrará un mensaje en la pantalla que diciéndole que es necesario conectar el socket (siguiente figura). Esto indica que primero que el módulo de Bluetooth del socket y el del dispositivo móvil deben estar conectados para poder mostrar la siguiente actividad.



Figura 3.29. Indicación de la aplicación para conectar al dispositivo móvil con el socket.

Al ser la primera vez que la aplicación es utilizada en el dispositivo móvil, el módulo de Bluetooth del socket debe ser emparejado con el de dicho dispositivo. Si se da clic al botón de conectar, se mostrará una lista de los dispositivos emparejados con el dispositivo móvil; sin embargo, no aparecerá el módulo de Bluetooth del socket. Debido a esto, el botón al que debe darse clic es al botón de emparejar, teniendo en cuenta que el sistema del socket debe estar encendido. Una vez habiendo dado clic en el botón emparejar, se abrirán los ajustes del Bluetooth del dispositivo, en el que se mostrarán los dispositivos emparejados, entre los que no se encuentra el módulo de Bluetooth del socket (siguiente figura). Al buscar algún dispositivo, se puede encontrar al del socket, precisamente con el mismo nombre "Socket", al cual se le debe dar clic para comenzar con el emparejamiento.

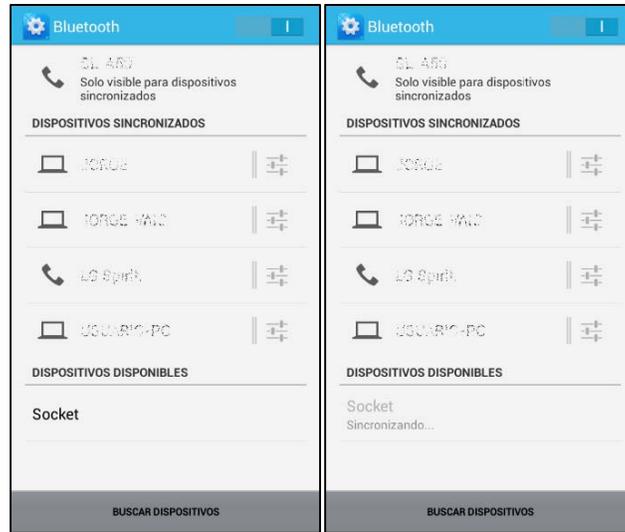


Figura 3.30. Ajustes del Bluetooth del dispositivo móvil.

Para comenzar con el emparejamiento, es necesario escribir el PIN del socket, el cual en este caso es 1234, y se le da clic en aceptar. Una vez habiendo realizado estos pasos, se habrá logrado emparejar al módulo de Bluetooth del socket, el cual se mostrará en la lista de dispositivos sincronizados (siguiente figura).



Figura 3.31. Emparejamiento entre el socket y el dispositivo móvil.

Posteriormente se deberá regresar a la visualización inicial para poder conectar al módulo de Bluetooth del dispositivo con el del socket. El botón de ayuda le permitirá al usuario poder consultar, de manera general, las instrucciones para el correcto funcionamiento de la aplicación (figura siguiente).



Figura 3.32. Aviso de ayuda de la primera visualización de la interfaz.

Una vez que el socket está emparejado con el dispositivo móvil, se procede a conectarlos mutuamente. Esto se logra con el botón conectar, el cual despliega una lista de los dispositivos sincronizados, y se seleccionará aquel con el nombre “Socket” (figura siguiente). Después de unos segundos, se logrará la conexión exitosa entre ambos dispositivos, como se muestra en la imagen de la derecha de la siguiente figura. Se debe mencionar que es necesario que el socket esté encendido para lograr la conexión.



Figura 3.33. Conexión entre el socket y el dispositivo móvil.

Como se puede observar en la figura superior, los botones de emparejar y conectar han desaparecido, y en su lugar ahora se encuentra el botón desconectar (color naranja). Esto indica que se logró una conexión correcta entre el módulo de Bluetooth del dispositivo móvil y el del socket protésico y, que puede comenzar el envío de datos entre ambos elementos. En caso de que se desee desconectar el socket del dispositivo móvil, únicamente será necesario presionar el botón en color naranja, con el mismo nombre de su función. El

último botón, antes de continuar a la siguiente actividad, es el de salir, el cual está constituido por una flecha curva en color blanco dentro de un cuadro rojo. Este botón cerrará la aplicación del socket protésico.

Ahora, para acceder a la siguiente actividad, se presiona el botón iniciar, una vez conectados los dispositivos. Con esto, se mostrará la siguiente visualización, la cual contiene esta vez cuatro botones, con diferentes funciones (siguiente figura). El primer elemento es el de la presión, el cual permite acceder a la actividad para el monitoreo y control de la presión. El segundo elemento es el de temperatura y humedad, con el cual se puede visualizar la actividad del monitoreo de la temperatura en el socket y la humedad. El tercer elemento es un elemento de información, el cual muestra un aviso con ciertas instrucciones de uso tanto de la aplicación como del socket. El último botón es el de salir, con el mismo símbolo que el de la primera actividad, el cual permite regresar a la actividad principal de la aplicación.



Figura 3.34. Visualizaciones donde se muestran los botones de presión, temperatura y humedad, información y salir. Además se muestra el aviso de información al presionar el botón con el mismo nombre.

Al presionar el botón de presión, se logra acceder a la actividad del monitoreo y control de la presión en el sistema de autoajuste del socket, mostrada en la figura siguiente. En esta se pueden observar diversos elementos. En la parte superior se muestra el nivel de presión de ajuste en milímetros de mercurio para los sujetadores. Por defecto se encuentra en 70 mmHg, lo que indica que el socket se encuentra ajustado al usuario con esa presión. Si se desea aumentar la presión de referencia, se puede utilizar el elemento de desplazamiento circular en color agua, el cual permite ser arrastrado horizontalmente de izquierda a derecha. Dependiendo de la posición del elemento de desplazamiento, se enviará al microcontrolador el nivel de referencia de la presión de ajuste de los sujetadores.



Figura 3.35. Visualización inicial de la actividad para el monitoreo y control de la presión.

Si se desea liberar el aire en los sujetadores completamente, se puede utilizar al botón desinflar que aparece en la visualización, esto envía datos al microcontrolador para indicarle que se debe de ajustar la presión al nivel mínimo de referencia.

Para acceder al monitoreo del control de presión en el socket, se presiona el botón PID en color amarillo. Este desplegará una gráfica de presión por tiempo de la presión de referencia y la presión actual en los sujetadores cada 500 ms, esto en concordancia con el tiempo de muestreo del control digital diseñado en el microcontrolador. La línea en color azul es la referencia del sistema de control, y la línea en color naranja es la respuesta del sistema ante los cambios de referencia. En la siguiente figura se muestra la visualización de la actividad con la gráfica, siendo modificada la presión de referencia (mediante el elemento de desplazamiento o mediante el botón desinflar/inflar) a distintos niveles.

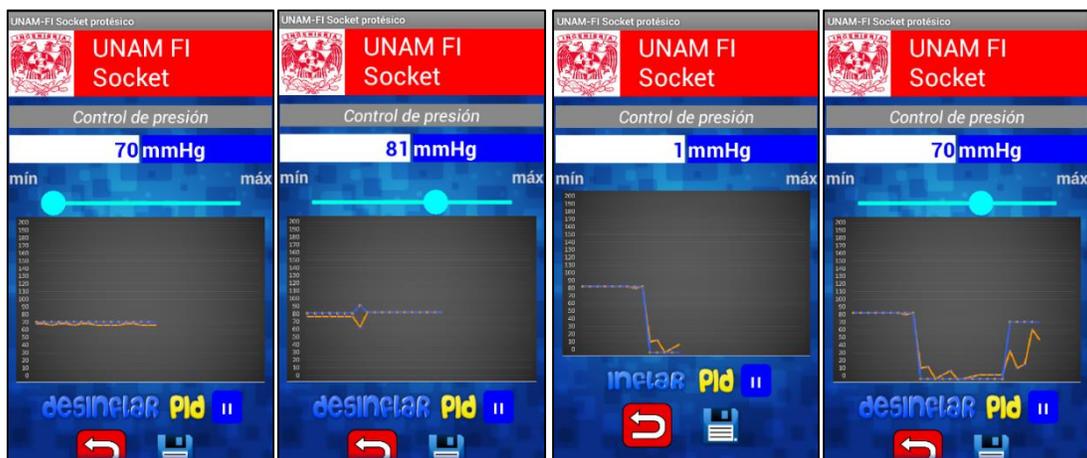


Figura 3.36. Diferentes ajustes de presión y la respuesta del sistema de control, mostrados en la gráfica.

El botón en color azul con dos líneas verticales blancas, es el botón de pausa, que detiene o reanuda la recepción y graficación de los datos de la presión desde el microcontrolador. Si se desea guardar los datos de la gráfica, se puede presionar el botón con el símbolo de un diskette. Este desplegará un mensaje para confirmar que se desea guardar la gráfica en el dispositivo móvil. Al responder afirmativamente, se mostrará la ruta de destino de la gráfica, que por defecto es almacenada en la memoria externa del dispositivo móvil en la carpeta llamada "Graphics". El nombre de la gráfica en forma de imagen, comienza con la hora y fecha de almacenaje y con la terminación "graphPID.jpg".



Figura 3.37. Almacenaje de la gráfica del control y monitoreo de la presión en los sujetadores.

Para regresar a la actividad anterior, se puede presionar el botón de color rojo con la flecha blanca. Y posteriormente se puede acceder al monitoreo de la temperatura y la humedad presionando el botón temperatura y humedad, el cual desplegará una nueva actividad, con los elementos mostrados a continuación.

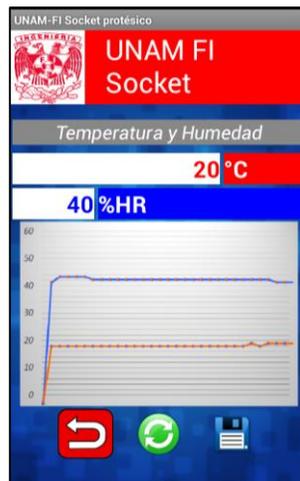


Figura 3.38. Visualización de la actividad para el monitoreo de los datos de la temperatura y la humedad en el socket.

El primer dato mostrado de arriba hacia abajo es la temperatura en grados centígrados, y debajo de este, la humedad relativa existente en el socket. Luego se puede observar la gráfica de la temperatura y la humedad. En color rojo se encuentra la línea de temperatura y en azul la de humedad. Esta gráfica muestra los datos de las últimas dos horas de uso del socket. Si se desea actualizar los datos de temperatura, se debe de presionar el botón en color verde. El botón de guardar realiza la misma función que el existente en la actividad del control y monitoreo de la presión. La única variación es la terminación del archivo guardado, la cual es "graphTyH.jpg" para diferenciarlo de las gráficas de presión.



Figura 3.39. Almacenaje de los datos de temperatura y humedad.

Finalmente, si se desea regresar a la actividad principal, se debe presionar el botón rojo en cada actividad, hasta llegar a la visualización de la figura 3.32. Al desconectar el dispositivo móvil y al socket, se mostrará la siguiente visualización, en la que no será necesario el botón de emparejar debido a que el socket y el dispositivo móvil ya se encuentran emparejados (figura siguiente). La siguiente ocasión que se acceda a la aplicación, no será necesario emparejar o conectar manualmente al socket con el dispositivo móvil, debido a que la aplicación lo realizará automáticamente. Solo se tendrá que conectar al socket de manera manual cuando no se haya activado el Bluetooth en el dispositivo o cuando no se tenga encendido el socket.



Figura 3.40. Actividad principal luego de desconectar al socket y al dispositivo móvil.

Capítulo 4. RESULTADOS Y CONCLUSIONES

El desarrollo de la instrumentación del socket protésico se llevó hasta la etapa del diseño de configuración en la que se lograron cumplir los objetivos y las necesidades planteadas originalmente. En este capítulo se abordan las cuestiones referentes a las pruebas de la instrumentación del socket autoajutable. Se comprueba el funcionamiento del control de presión en los sujetadores, a través de la interfaz gráfica en la aplicación diseñada. Además se comprueba que los datos de temperatura y humedad pueden ser mostrados en la interfaz de la aplicación. Finalmente se concluye este trabajo, analizando el aporte que se le da a este proyecto del socket protésico a nivel transfemoral.

El modelo utilizado para la realización de las pruebas permitió comprobar el funcionamiento del socket protésico con la instrumentación. En la siguiente figura se muestra una imagen del modelo en la que aparecen los componentes de la instrumentación, como el circuito, el módulo de Bluetooth, el sensor de temperatura y humedad, el sensor de presión, la bomba neumática y la servoválvula. Además, para poder simular a los sujetadores, se usó una bolsa de aire de un baumanómetro digital, la cual fue conectada a la servoválvula para el ingreso, retención o salida del aire.

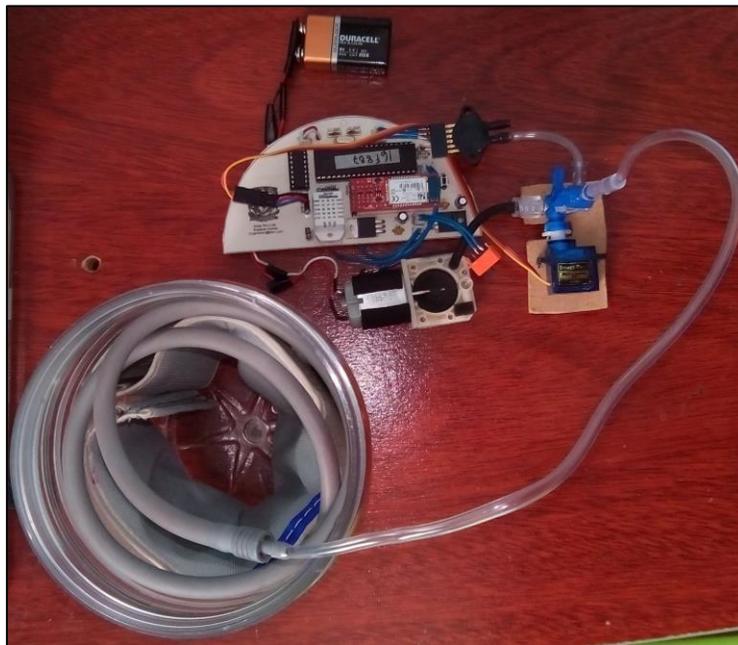


Figura 4.1. Modelo para las pruebas de la instrumentación en el socket.

Para simular al muñón, y poder realizar las pruebas del funcionamiento de la instrumentación en el socket, se colocó una bolsa de aire de un baumanómetro dentro de los sujetadores, como se observa en la figura siguiente. Las pruebas realizadas con este modelo pretenden comprobar los objetivos planteados al inicio de este trabajo, correspondientes a la instrumentación del socket. Entre ellos se encuentran que el socket sea capaz de medir temperatura, humedad y presión. Además, que transfiera estos datos

hacia un dispositivo móvil con sistema operativo Android. Y mediante una aplicación se pueda almacenar esta información. De igual forma que controle al sistema de autoajuste del socket. En adición a lo anterior, se corroborará que el control de presión cumpla con sus requerimientos planteados.



Figura 4.2. Modelo con simulación de un muñón mediante una bolsa de aire, listo para realizar las pruebas.

4.1 Pruebas y resultados

Prueba 1

En esta primera prueba se corroboró que el control de la presión cumpliera con los requerimientos planteados. Primero, de manera similar a como se realizó en un trabajo previo del socket [23], se comprobó el funcionamiento del sensor de presión al conectar el manómetro baumanómetro a los sujetadores. La presión obtenida del sensor, mostrada en la gráfica de la aplicación, fue comparada con la presión que indicaba el manómetro, que presentó un error no mayor a 1 mmHg.

Posteriormente se estableció la presión de referencia en 0 mmHg, en los sujetadores. Una vez alcanzada esta presión y con la bolsa de aire del baumanómetro simulando al muñón, se procedió (a través de la aplicación en Android) a cambiar la presión de referencia a 70 mmHg. En ese momento, el sistema de autoajuste permitió alcanzar a la referencia, como se muestra a continuación en la gráfica obtenida de la aplicación.

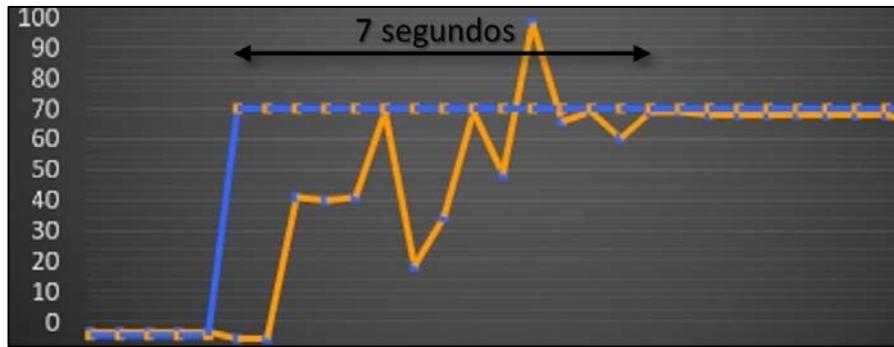


Figura 4.3. Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms).
 La línea azul representa la señal de referencia de 0 a 70 mmHg.
 La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.

En la figura anterior se observa que el tiempo de asentamiento fue de 7 segundos, con un sobrepaso aproximado de 20%, y que no existe error en estado estable. La respuesta del sistema no cumple fehacientemente los requerimientos del sistema. Sin embargo, este diseño es más eficiente que el diseño propuesto anteriormente [23], el cual tardaba más de un minuto de tiempo de asentamiento en llegar a la presión de referencia, en contraste con los 7 segundos observados en esta prueba realizada.

Además, se puede concluir que la razón por la cual el tiempo de asentamiento obtenido es diferente a lo establecido en los requerimientos de control, puede deberse a las características de los sujetadores. En la caracterización de la planta, se consideró que los sujetadores sufren un cambio en el volumen despreciable, algo que en realidad es diferente. Cuando aumenta el flujo de aire hacia los sujetadores, el volumen de ellos aumenta, y esto hace que se retarde la eliminación del error en el control de la presión. Por otro lado, la discretización del sistema de control, con un tiempo de muestreo de 500 ms, también puede afectar al tiempo de respuesta y al sobrepaso obtenido. Esto se debe a que el sistema de autoajuste pudo haber llegado a alcanzar la presión de referencia antes del siguiente tiempo de muestreo. Lo cual se observa en la figura anterior, en la que la presión alcanza los 90 mmHg y se debió disminuir automáticamente.

Prueba 2

La siguiente prueba consistió en observar el comportamiento del control de la presión ante los cambios en el nivel de referencia. Para ello se estableció la presión de referencia en 70 mmHg, y se cambió posteriormente a 90 mmHg, en donde se encontró que el control nuevamente permitió que la presión en los sujetadores alcanzara este valor y conseguir un error nulo en el estado estable. Además, la respuesta nuevamente fue rápida al alcanzar este valor en aproximadamente cuatro segundos. El sobrepaso fue aceptable ante este cambio como se observa en la siguiente figura.

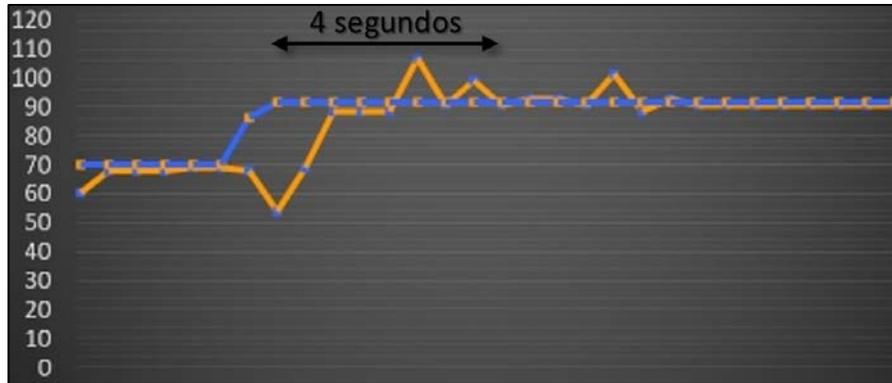


Figura 4.4. Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms).
 La línea azul representa la señal de referencia con un cambio de 70 a 90 mmHg.
 La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.

Prueba 3

La tercera prueba involucró el comportamiento del control de presión ante cambios en el muñón. Para lograr la simulación de este evento, se aumentó y disminuyó la presión en la bolsa de aire del baumanómetro. La referencia se fijó en 70 mmHg y se pudo observar que el error existente intentaba ser eliminado por el controlador PI. Como se muestra en la siguiente figura, la presión en los sujetadores aumentaba y disminuía ante los cambios de volumen del muñón (variación de la presión en la bolsa de aire del baumanómetro).

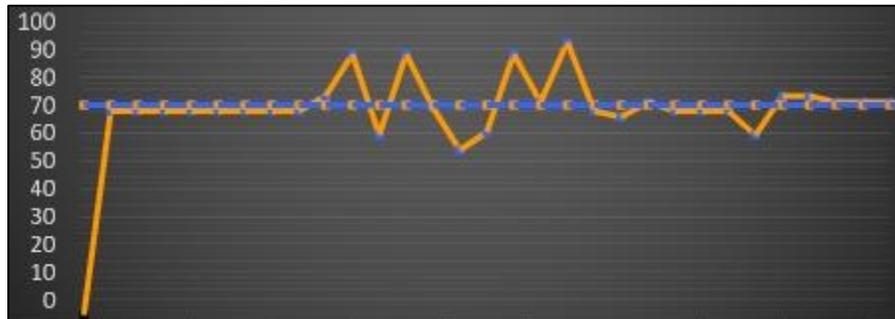


Figura 4.5. Gráfica de presión (mmHg) contra el tiempo (cada punto equivale a 500 ms).
 La línea azul representa la señal de referencia de 70 mmHg.
 La línea naranja es la señal obtenida del sensor MPX5050 cada 500 ms.

Prueba 4

La cuarta prueba consistió en comprobar que la temperatura y la humedad fuesen monitoreadas, almacenadas en el microcontrolador, y posteriormente observadas gráficamente en la aplicación móvil. Para ello se dejó funcionar al sistema durante más de una hora, debido a que la temperatura y humedad son medidas mediante el sensor DHT22 por 30 veces cada dos minutos. Luego de ello, se accedió al monitoreo de la temperatura y la humedad en la aplicación, en donde se pudo obtener los valores obtenidos desde la memoria EEPROM del microcontrolador. Estos datos representaron los datos de la temperatura y la humedad en el socket existentes en la última hora de su utilización. La figura siguiente muestra lo descrito en este párrafo.

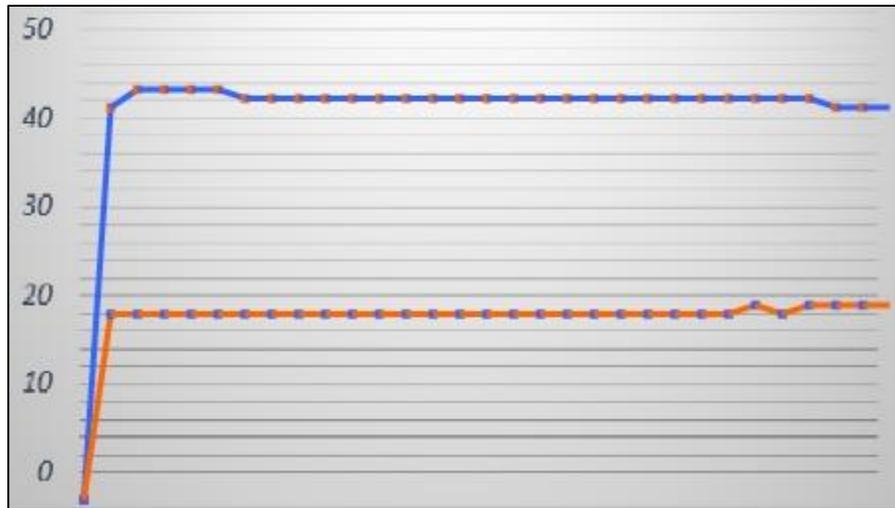


Figura 4.6. Gráfica de temperatura (°C) y humedad (%HR) contra el tiempo (cada punto equivale 2 minutos). La línea azul representa los valores de humedad en el socket cada dos minutos en la última hora de utilización de su utilización. La línea naranja indica los datos de temperatura en el socket en el mismo intervalo de tiempo que los obtenidos de la humedad.

En la gráfica anterior se puede notar que la temperatura no alcanza siquiera los 20 °C, algo que indicaría una anomalía en el muñón. Esto se debe a la sustitución de este último mediante una bolsa de aire de un baumanómetro, para realizar las pruebas pertinentes.

También se pudo observar que la humedad relativa sí se encontró en el rango del confort térmico como se planteó en el capítulo 2. De igual manera, los valores obtenidos de la humedad no representan los valores que se podrían llegar a obtener en el socket, una vez que éste se encuentre siendo utilizado por el usuario de la prótesis. Sin embargo, una vez habiendo completado la manufactura y procedido a realizar las pruebas con la prótesis, estos datos le serán de gran apoyo al diseñador para analizar el comportamiento del muñón ante dichas pruebas. Además, la temperatura y la humedad en el socket le permitirán a un médico especialista darle indicaciones al usuario de la prótesis en caso de encontrar una anomalía, como se describió en el capítulo 1.

4.2 Conclusiones

Una vez realizadas las pruebas, se pudo concluir que los objetivos planteados al inicio de este trabajo se cumplieron. La medición de la temperatura, la humedad en el socket, así como la presión en los sujetadores, se realizó de manera adecuada gracias a cada uno de los sensores utilizados para la instrumentación.

Los datos de temperatura y humedad mostrados gráficamente a través de la aplicación diseñada, le permitirán al médico realizar conclusiones respecto al comportamiento del muñón. Además, servirán como apoyo para prevenir lesiones, abscesos o algún otro

problema en el miembro residual del usuario, como isquemia, necrosis o la presencia de hongos. Con base en el monitoreo, al observar algún tipo de irregularidad en la temperatura y la humedad en el socket, el usuario decidirá si es necesario dejar de utilizar su prótesis por un tiempo determinado hasta consultar a un especialista o hasta que la anomalía desaparezca.

La aplicación desarrollada en Android permite realizar el monitoreo de estas mediciones y del control de presión del sistema de autoajuste del socket, de manera integral. La transferencia de estos datos desde el microcontrolador hacia el dispositivo móvil con un sistema operativo Android también se logró de manera exitosa mediante un módulo de Bluetooth seleccionado en la instrumentación.

Por otro lado, la aplicación desarrollada en este trabajo, ahora permite al usuario controlar el nivel de ajuste referencia y observar el comportamiento del control de la presión en socket. Gracias a esto, el encargado de diseñar el socket a futuro, podrá realizar ajustes en el sistema neumático y observar cómo se comporta el control PI ante dichos cambios.

El control del sistema de autoajuste del socket también mejoró en la rapidez de respuesta, como se planteó en las necesidades, comparado con el diseño previo del socket [23]. La mejora es significativa al reducir el tiempo de asentamiento de un minuto a siete segundos, aproximadamente. El sobrepaso es alto; sin embargo, esto no representa una desventaja. Al contrario, cuando la presión en los sujetadores presenta un sobrepaso alto, como el que se observó en las pruebas, los sujetadores llegan más rápidamente a su presión de referencia. Esto se debe a la característica de los sujetadores, que pierden presión paulatinamente, debido a que están hechos de un material elástico. Así que, al presentarse el sobrepaso, esta última característica, combinada con la acción de control, hacen que la presión en los sujetadores alcance su valor de referencia más rápidamente que en el caso contrario (sin sobrepaso). En la figura de la prueba 2 se puede observar lo descrito previamente, de mejor manera, ya que una vez que se dio el sobrepaso, el sistema llega a su referencia en un segundo, aproximadamente.

Además, se pudo observar que el control de la presión responde satisfactoriamente ante los cambios de volumen en el muñón, los cuales fueron simulados mediante la variación de presión en la bolsa de aire del baumanómetro. La presión de los sujetadores se ajusta automáticamente ante dichos cambios, como se observó en la prueba 3.

En el caso del sistema neumático, éste fue mejorado al reducir el tamaño y peso de la servoválvula. Además, la bomba neumática mejoró su nivel de sonido de manera significativa, comparada con la utilizada en trabajos previos.

La instrumentación del socket protésico para amputación a nivel transfemoral permite ahora integrar dos funciones fundamentales en un solo microcontrolador. Éstas son el monitoreo y almacenamiento de la temperatura y la humedad, así como el control de la presión en los sujetadores. Esto no se había logrado en los anteriores trabajos del diseño

del socket, lo que representa, ahora, un gran paso para la posterior implementación en un prototipo funcional. Y se puede concluir, por lo tanto, que el sistema de autoajuste del socket fue mejorado.

Con este trabajo de la instrumentación del socket, se puede asegurar, que el usuario de prótesis se sentirá más confiado de contar con un sustituto de su miembro inferior que le brinde la seguridad de autoajuste y el confort. Además ahora puede reemplazar los liners por los sujetadores de este socket, reduciendo la temperatura y la humedad generadas, y ahora monitoreadas en paralelo con el control del sistema de autoajuste.

Capítulo 5. RECOMENDACIONES PARA EL TRABAJO A FUTURO Y EL DISEÑO DE DETALLE

Las recomendaciones para el diseño del socket protésico para amputación a nivel transfemoral se plantean a continuación:

- **Sistema electrónico.** En este sistema se puede mejorar el diseño del circuito al utilizar elementos de montaje superficial para el microcontrolador, el sensor de presión, el módulo de Bluetooth y algunos otros elementos. Es necesario también, agregar al circuito la parte necesaria para poder cargar la batería LiPo seleccionada, sin la necesidad de removerla del socket para conectarla a un cargador externo.
- **Sistema de programación.** En cuanto a la aplicación desarrollada en Android, ésta tiene diversas áreas de oportunidad, entre ellas el diseño gráfico, y el aumento de nuevas funciones. Además, la aplicación de este proyecto fue desarrollada mediante App Inventor del MIT, por lo que en trabajos a futuros se recomienda realizar el diseño en algún otro IDE como Android Studio, mediante Java. Esto le permitirá a la aplicación contar con ciertas funciones que no son posibles de implementar mediante App Inventor (hasta el momento) como el zoom en las gráficas, por ejemplo. Y sobre todo, reducir el tamaño de la aplicación, el cual puede ser excesivo en algunos dispositivos móviles.
- **Sistema neumático.** Este sistema puede mejorar en el diseño de la servoválvula posiblemente para reducir aún más el tamaño y peso de ésta última en el socket. Además, la bomba neumática seleccionada en este trabajo, puede ser sustituida por otra de similares características pero con un menor tamaño y nivel de sonido, en caso de ser encontrada.
- El socket debe ser analizado mediante un análisis de elemento finito con la instrumentación, para poder posteriormente tener la certeza de que no fallará y que puede ser manufacturado.
- Por otro lado, en caso de ser necesario, se recomienda integrar un algoritmo que permita medir la presión arterial del usuario y ser monitoreada en la aplicación diseñada.
- En el caso de la humedad y la temperatura se puede implementar un sistema en el que se controlen estas variables físicas, posiblemente haciendo uso de la bomba neumática con la que ya cuenta el socket.
- También se recomienda realizar un prototipo para poder realizar las pruebas físicas con un usuario de prótesis y utilizar las herramientas desarrolladas en el diseño del socket.
- Finalmente, es necesario realizar la protección del diseño del socket mediante derechos de autor y patente.

Apéndice

- A1. Microcontrolador PIC16F887
- A2. Módulo de Bluetooth RN-42
- A3. Sensor DHT22
- A4. Sensor MPX5050DP
- A5. Servomotor SG90
- A6. Esquema del circuito del sistema
- A7. Código de programación del microcontrolador
- A8. Código de programación de la aplicación

A1. Microcontrolador PIC16F887



PIC16F882/883/884/886/887

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 Instructions to Learn:
 - All single-cycle instructions except branches
- Operating Speed:
 - DC – 20 MHz oscillator/clock input
 - DC – 200 ns instruction cycle
- Interrupt Capability
- 8-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
 - Factory calibrated to $\pm 1\%$
 - Software selectable frequency range of 8 MHz to 31 kHz
 - Software tunable
 - Two-Speed Start-up mode
 - Crystal fail detect for critical applications
 - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide Operating Voltage Range (2.0V-5.5V)
- Industrial and Extended Temperature Range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with Software Control Option
- Enhanced Low-Current Watchdog Timer (WDT) with On-Chip Oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with Pull-up/Input Pin
- Programmable Code Protection
- High Endurance Flash/EEPROM Cell:
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - Flash/Data EEPROM retention: > 40 years
- Program Memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
 - 50 nA @ 2.0V, typical
- Operating Current:
 - 11 μ A @ 32 kHz, 2.0V, typical
 - 220 μ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
 - 1 μ A @ 2.0V, typical

Peripheral Features:

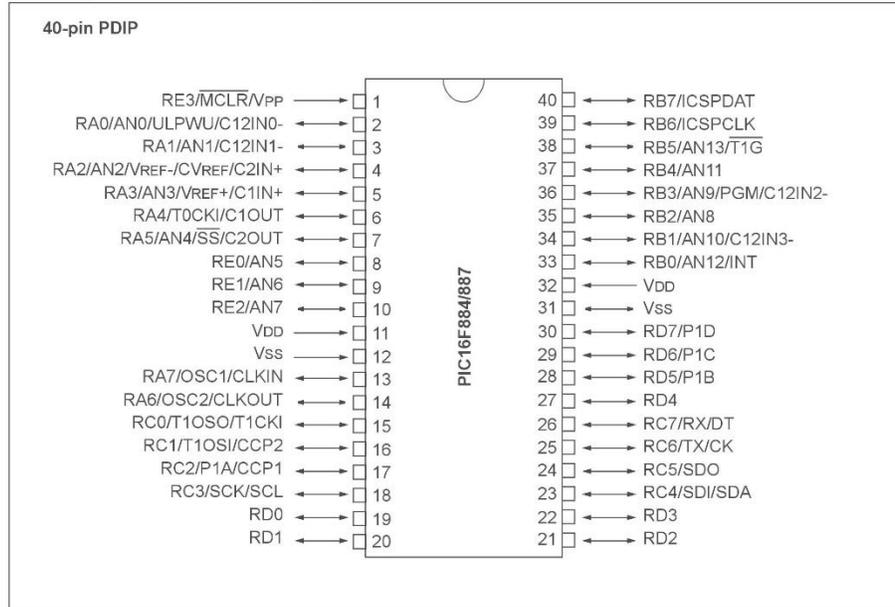
- 24/35 I/O Pins with Individual Direction Control:
 - High current source/sink for direct LED drive
 - Interrupt-on-Change pin
 - Individually programmable weak pull-ups
 - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator Module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (CVREF) module (% of VDD)
 - Fixed voltage reference (0.6V)
 - Comparator inputs and outputs externally accessible
 - SR Latch mode
 - External Timer1 Gate (count enable)
- A/D Converter:
 - 10-bit resolution and 11/14 channels
- Timer0: 8-bit Timer/Counter with 8-bit Programmable Prescaler
- Enhanced Timer1:
 - 16-bit timer/counter with prescaler
 - External Gate Input mode
 - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit Timer/Counter with 8-bit Period Register, Prescaler and Postscaler
- Enhanced Capture, Compare, PWM+ Module:
 - 16-bit Capture, max. resolution 12.5 ns
 - Compare, max. resolution 200 ns
 - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
 - PWM output steering control
- Capture, Compare, PWM Module:
 - 16-bit Capture, max. resolution 12.5 ns
 - 16-bit Compare, max. resolution 200 ns
 - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART Module:
 - Supports RS-485, RS-232, and LIN 2.0
 - Auto-Baud Detect
 - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- Master Synchronous Serial Port (MSSP) Module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave Modes with I²C Address Mask

PIC16F882/883/884/886/887

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	MSSP	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F882	2048	128	128	24	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1

PIC16F882/883/884/886/887

Pin Diagrams – PIC16F884/887, 40-Pin PDIP



PIC16F882/883/884/886/887

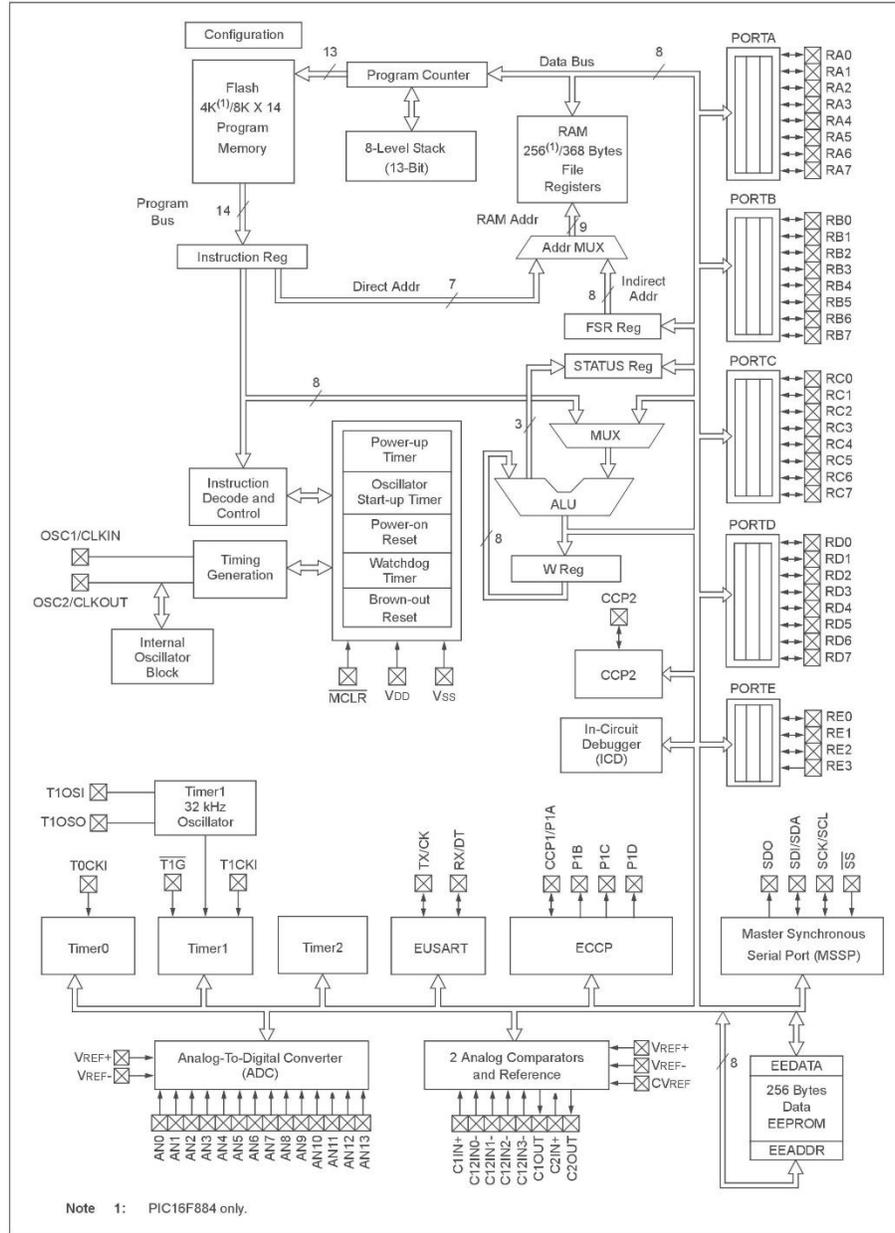
TABLE 3: PIC16F884/887 40-PIN SUMMARY (PDIP)

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	14	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	13	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	33	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	34	AN10	C12IN3-	—	—	—	—	IOC	Y	—
RB2	35	AN8	—	—	—	—	—	IOC	Y	—
RB3	36	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	37	AN11	—	—	—	—	—	IOC	Y	—
RB5	38	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	39	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	40	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	15	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	16	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	17	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	18	—	—	—	—	—	SCK/SCL	—	—	—
RC4	23	—	—	—	—	—	SDI/SDA	—	—	—
RC5	24	—	—	—	—	—	SDO	—	—	—
RC6	25	—	—	—	—	TX/CK	—	—	—	—
RC7	26	—	—	—	—	RX/DT	—	—	—	—
RD0	19	—	—	—	—	—	—	—	—	—
RD1	20	—	—	—	—	—	—	—	—	—
RD2	21	—	—	—	—	—	—	—	—	—
RD3	22	—	—	—	—	—	—	—	—	—
RD4	27	—	—	—	—	—	—	—	—	—
RD5	28	—	—	—	P1B	—	—	—	—	—
RD6	29	—	—	—	P1C	—	—	—	—	—
RD7	30	—	—	—	P1D	—	—	—	—	—
RE0	8	AN5	—	—	—	—	—	—	—	—
RE1	9	AN6	—	—	—	—	—	—	—	—
RE2	10	AN7	—	—	—	—	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	11	—	—	—	—	—	—	—	—	VDD
—	32	—	—	—	—	—	—	—	—	VDD
—	12	—	—	—	—	—	—	—	—	VSS
—	31	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.

PIC16F882/883/884/886/887

FIGURE 1-2: PIC16F884/PIC16F887 BLOCK DIAGRAM



A2. Módulo de Bluetooth RN-42



RN-42/RN-42-N Data Sheet

www.rovingnetworks.com

DS-RN42-V1.0 6/21/2011

Class 2 Bluetooth® Module



Features

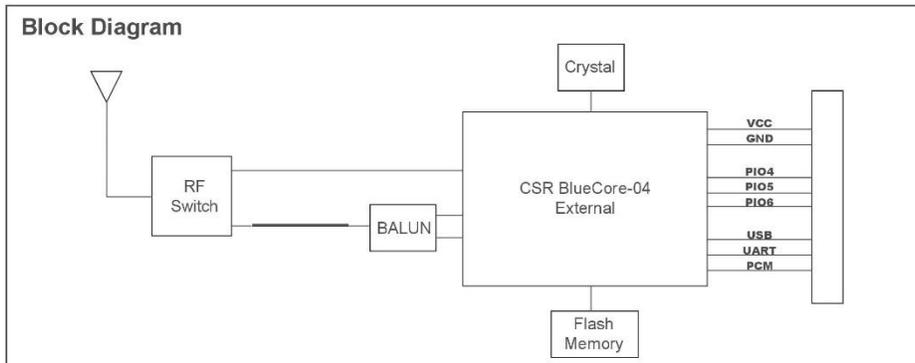
- Fully qualified Bluetooth 2.1/2.0/1.2/1.1 module
- Bluetooth v2.0+EDR support
- Available with on board chip antenna (RN-42) and without antenna (RN-42-N)
- Postage stamp sized form factor, 13.4mm x 25.8 mm x 2mm (RN-42) and 13.4mm x 20 mm x 2 mm (RN-42-N)
- Low power (*26uA sleep, 3mA connected, 30mA transmit*)
- UART (SPP or HCI) and USB (HCI only) data connection interfaces.
- Sustained SPP data rates - 240Kbps (slave), 300Kbps (master)
- HCI data rates - 1.5Mbps sustained, 3.0Mbps burst in HCI mode
- Embedded Bluetooth stack profiles included (*requires no host stack*): GAP, SDP, RFCOMM and L2CAP protocols, with SPP and DUN profile support.
- Bluetooth SIG certified
- Castellated SMT pads for easy and reliable PCB mounting
- Certifications: FCC, ICS, CE
- Environmentally friendly, RoHS compliant

Applications

- Cable replacement
- Barcode scanners
- Measurement and monitoring systems
- Industrial sensors and controls
- Medical devices
- Barcode readers
- Computer accessories

Description

The RN42 is a small form factor, low power, highly economic Bluetooth radio for OEM's adding wireless capability to their products. The RN42 supports multiple interface protocols, is simple to design in and fully certified, making it a complete embedded Bluetooth solution. The RN 42 is functionally compatible with RN 41. With its high performance on chip antenna and support for Bluetooth® Enhanced Data Rate (EDR), the RN42 delivers up to 3 Mbps data rate for distances to 20M. The RN-42 also comes in a package with no antenna (RN-42-N). Useful when the application requires an external antenna, the RN-42-N is shorter in length and has RF pads to route the antenna signal.



809 University Avenue • Los Gatos, CA 95032 • Tel (408) 395-6539 • info@RovingNetworks.com

~ 1 ~



RN-42/RN-42-N Data Sheet

DS-RN42-V3.2 6/21/2011

Overview

- Baud rate speeds: 1200bps up to 921Kbps, non-standard baud rates can be programmed.
- Class 2 radio, 60 feet (20meters) distance, 4dBm output transmitter, -80dBm typical receive sensitivity
- Frequency 2402 ~ 2480MHz,
- FHSS/GFSK modulation, 79 channels at 1MHz intervals
- Secure communications, 128 bit encryption
- Error correction for guaranteed packet delivery
- UART local and over-the-air RF configuration
- Auto-discovery/pairing requires no software configuration (instant cable replacement).
- Auto-connect master, IO pin (DTR) and character based trigger modes

Digital I/O Characteristics

2.7V ≤ VDD ≤ 3.0V	Min	Typ.	Max.	Unit
Input logic level LOW	-0.4	-	+0.8	V
Input logic level HIGH	0.7VDD	-	VDD+0.4	V
Output logic level LOW	-	-	0.2	V
Output logic level HIGH	VDD-0.2	-	-	V
All I/O's (except reset) default to weakpull down	+0.2	+1.0	+5.0	uA

Environmental Conditions

Parameter	Value
Temperature Range (Operating)	-40 °C ~ 85 °C
Temperature Range (Storage)	-40 °C ~ 85 °C
Relative Humidity (Operating)	≤90%
Relative Humidity (Storage)	≤90%

Electrical Characteristics

Parameter	Min	Typ.	Max.	Unit
Supply Voltage (DC)	3.0	3.3	3.6	V
Average power consumption				
Radio ON* (Discovery or Inquiry window time)		40		mA
Connected Idle (No Sniff)		25		mA
Connected Idle (Sniff 100 milli secs)		12		mA
Connected with data transfer	40	45	50	mA
Deep Sleep Idle mode		26		uA

* If in SLAVE mode there are bursts of radio ON time which vary with the windows. Depending on how you set the windows that determines your average current.



RN-42/RN-42-N Data Sheet

DS-RN42-V3.2 6/21/2011

Radio Characteristics

Parameter	Freq. (GHz)	Min	Typ	Max	Bluetooth Specification	Units
Sensitivity @ 0.1%BER	2.402	-	-80	-86	≤ -70	dBm
	2.441	-	-80	-86		dBm
	2.480	-	-80	-86		dBm
RF Transmit Power	2.402	0	2	4	≤ 4	dBm
	2.441	0	2	4		dBm
	2.480	0	2	4		dBm
Initial Carrier Frequency Tolerance	2.402	-	5	75	75	kHz
	2.441	-	5	75		kHz
	2.480	-	5	75		kHz
20dB bandwidth for modulated carrier		-	900	1000	≤ 1000	kHz
Drift (Five slots packet)		-	15	-	40	kHz
Drift Rate		-	13	-	20	kHz
Δf _{1avg} Max Modulation	2.402	140	165	175	>140	kHz
	2.441	140	165	175		kHz
	2.480	140	165	175		kHz
Δf _{2avg} Min Modulation	2.402	140	190	-	115	kHz
	2.441	140	190	-		kHz
	2.480	140	190	-		kHz

Range Characteristics (Approximate range in office environment)

Range	RN-42
After One Wall	55 feet
After Two Walls	60 feet
After Three Walls	36 feet

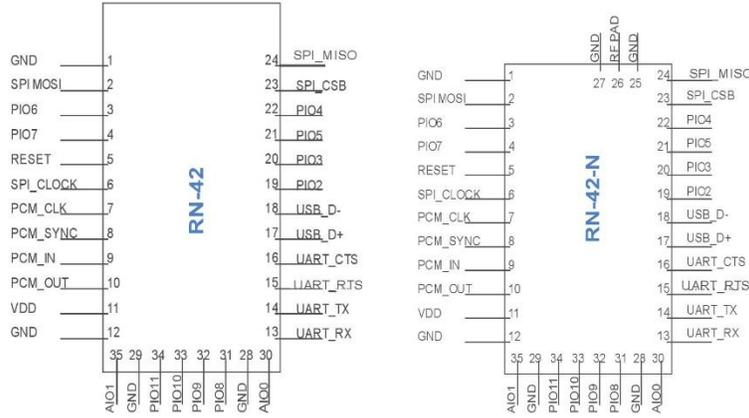
The above readings are approximate and may vary depending upon the RF environment. Bluetooth hops in a pseudo-random fashion over the 79 frequencies in the ISM band to adapt to the interference. Data throughput and range vary depending on the RF interference environment.



RN-42/RN-42-N Data Sheet

DS-RN42-V3.2 6/21/2011

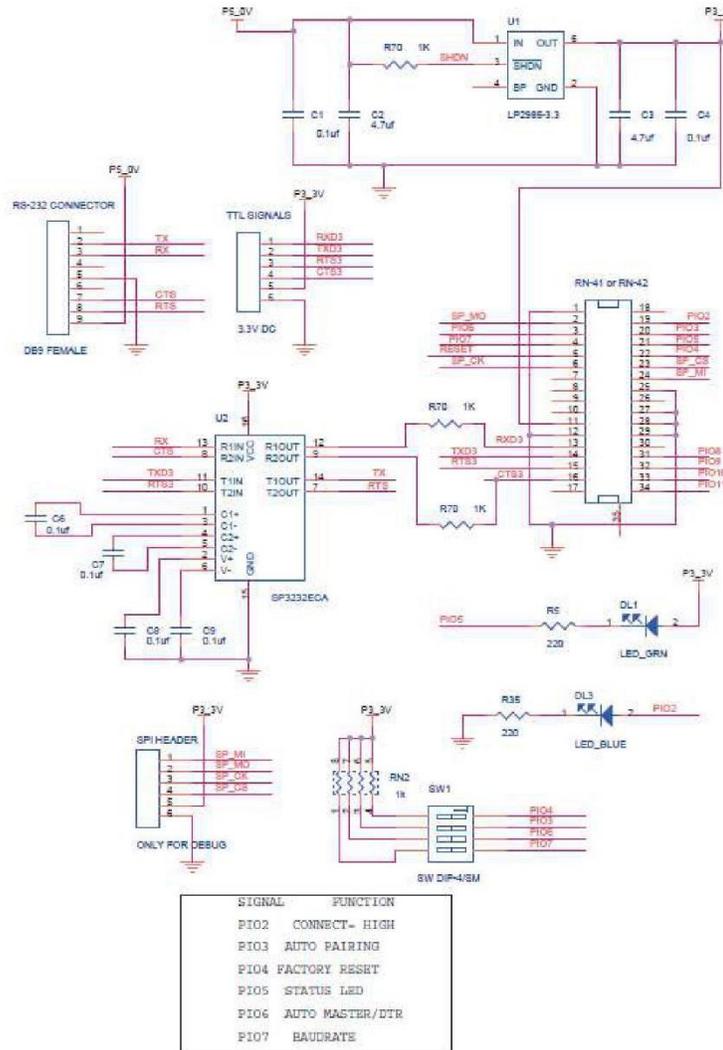
Pin Description



Pin	Name	Description	Default	Voltage
1	GND			0V
2	SPI_MOSI	Programming only	No Connect	3V
3	PIO6	Set BT master (HIGH=auto-master mode)	Input to RN42with weak pulldown	0V-3.3V
4	PIO7	Set Baud rate (HIGH = force 9600, LOW = 115K or firmware setting)	Input to RN42 with weak pulldown	0V-3.3V
5	RESET	Active LOW reset	Input to RN42 with 1K pullup	
6	SPI_CLK	Programming only	No Connect	
7	PCM_CLK	PCM interface	No Connect	
8	PCM_SYNC	PCM interface	No Connect	
9	PCM_IN	PCM interface	No Connect	
10	PCM_OUT	PCM interface	No Connect	
11	VDD	3.3V regulated power input		
12	GND			
13	UART_RX	UART receive input	Input to RN42	0V-3.3V
14	UART_TX	UART transmit output	High level output from RN42	0V-3.3V
15	UART_RTS	UART RTS, goes HIGH to disable host transmitter	Low level output from RN42	0V-3.3V
16	UART_CTS	UART CTS, if set HIGH, disables transmitter	Low level input to RN42	0V-3.3V
17	USB_D+	USB port	Pull up 1.5K when active	0V-3.3V
18	USB_D-	USB port		0V-3.3V
19	PIO2	Status, HIGH when connected, LOW otherwise	Output from RN42	0V-3.3V
20	PIO3	Auto discovery = HIGH	Input to RN42 with weak pulldown	0V-3.3V
21	PIO5	Status, toggles based on state, LOW on connect	Output from RN42	0V-3.3V
22	PIO4	Set factory defaults	Input to RN42 with weak pulldown	0V-3.3V
23	SPI_CSB	Programming only	No Connect	
24	SPI_MISO	Programming only	No Connect	
25	GND	GND for RN42-N		
26	RF Pad	RF Pad for RN42-N		
27	GND	GND for RN42-N		
30	AIO0	Optional analog input	Not Used	
31	PIO8	Status (RF data rx/tx)	Output from RN42	0V-3.3V
32	PIO9	IO	Input to RN42 with weak pulldown	0V-3.3V
33	PIO10	IO (remote DTR signal)	Input to RN42 with weak pulldown	0V-3.3V
34	PIO11	IO (remote RTS signal)	Input to RN42 with weak pulldown	0V-3.3V
35	AIO1	Optional analog input	Not Used	

Typical Application Circuit

Since the RN 41 and RN 42 are functionally compatible, this application diagram applies to RN 41 and RN 42.



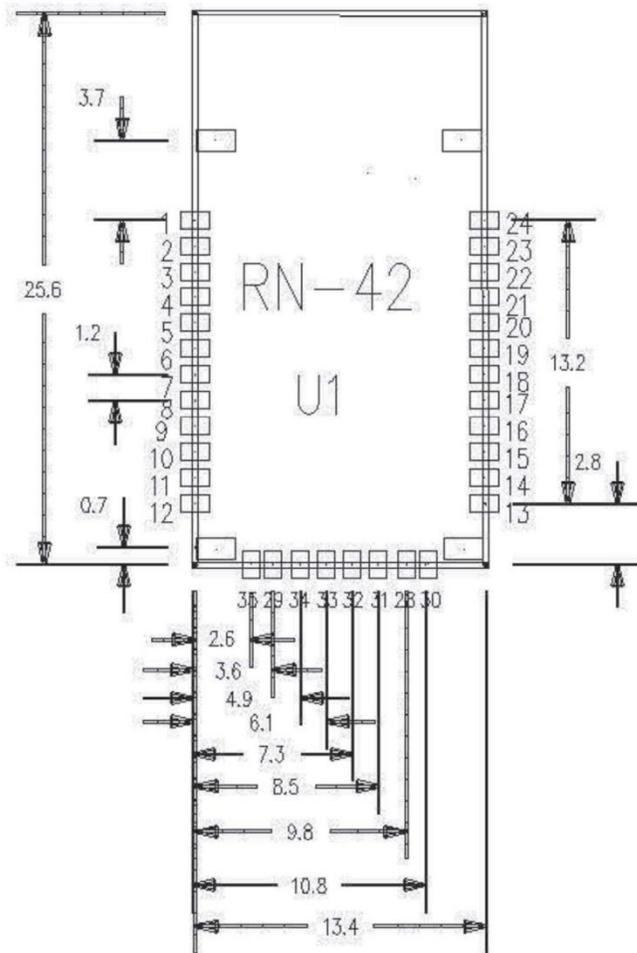


RN-42/RN-42-N Data Sheet

DS-RN42-V3.2 6/21/2011

RN-42 Module Dimensions

PAD SIZE = 0.8 x 1.30 mm



NOTE: All dimensions are in mm

A3. Sensor DHT22



Your specialist in innovating humidity & temperature sensors



Digital relative humidity & temperature sensor RHT03

1. Feature & Application:

- *High precision
- *Capacitive type
- *Full range temperature compensated
- *Relative humidity and temperature measurement
- *Calibrated digital signal
- *Outstanding long-term stability
- *Extra components not needed
- *Long transmission distance, up to 100 meters
- *Low power consumption
- *4 pins packaged and fully interchangeable

2. Description:

RHT03 output calibrated digital signal. It applies exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(100m) enable RHT03 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

3. Technical Specification:

Model	RHT03
Power supply	3.3-5.5V DC
Output signal	digital signal via MaxDetect 1-wire bus
Sensing element	Polymer humidity capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH (Max +-5%RH); temperature +-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Interchangeability	fully interchangeable

4. Dimensions: (unit---mm)

MaxDetect Technology Co., Ltd.

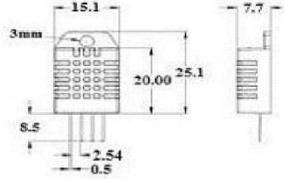
- 1 -

<http://www.humiditycn.com>

Thomas Liu (Sales Manager)
 Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com



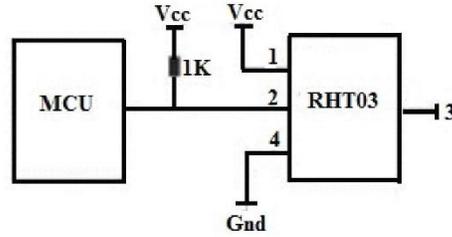
Your specialist in innovating humidity & temperature sensors



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	GND
4	GND

5. Electrical connection diagram:



6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3.3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

(2) Communication and signal

MaxDetect 1-wire bus is used for communication between MCU and RHT03. (MaxDetect 1-wire bus is specially designed by MaxDetect Technology Co., Ltd. , it's different from Maxim/Dallas 1-wire bus, so it's incompatible with Dallas 1-wire bus.)

Illustration of MaxDetect 1-wire bus:

DATA=16 bits RH data+16 bits Temperature data+8 bits check-sum

Example: MCU has received 40 bits data from RHT03 as

0000 0010 1000 1100 0000 0001 0101 1111 1110 1110

16 bits RH data 16 bits T data check sum

Here we convert 16 bits RH data from binary system to decimal system,

0000 0010 1000 1100 → 652

Binary system Decimal system

RH=652/10=65.2%RH

Here we convert 16 bits T data from binary system to decimal system,

0000 0001 0101 1111 → 351

Binary system Decimal system

T=351/10=35.1°C

When highest bit of temperature is 1, it means the temperature is below 0 degree Celsius.

Example: 1000 0000 0110 0101, T= minus 10.1°C

16 bits T data

Sum=0000 0010+1000 1100+0000 0001+0101 1111=1110 1110

Check-sum=the last 8 bits of Sum=1110 1110

MaxDetect Technology Co., Ltd.

- 2 -

<http://www.humiditycn.com>

Thomas Liu (Sales Manager)

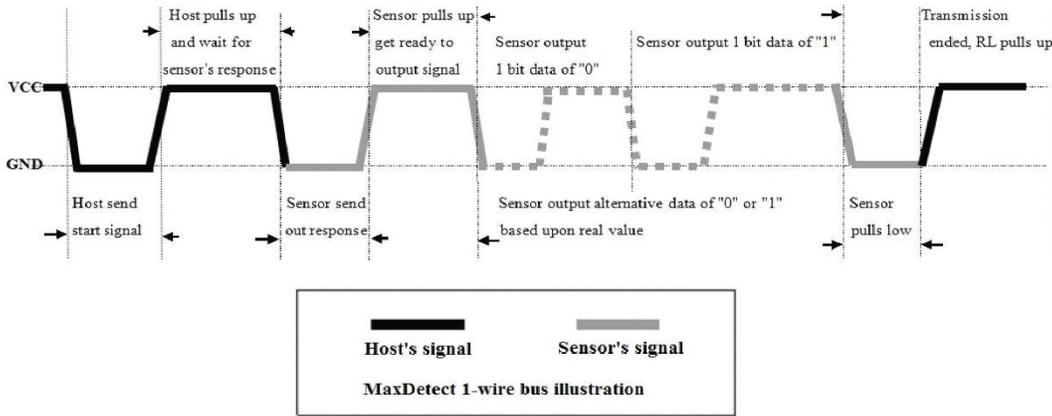
Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com



Your specialist in innovating humidity & temperature sensors

When MCU send start signal, RHT03 change from standby-status to running-status. When MCU finishes sending the start signal, RHT03 will send response signal of 40-bit data that reflect the relative humidity and temperature to MCU. Without start signal from MCU, RHT03 will not give response signal to MCU. One start signal for one response data from RHT03 that reflect the relative humidity and temperature. RHT03 will change to standby status when data collecting finished if it don't receive start signal from MCU again.

See below figure for overall communication process, **the interval of whole process must beyond 2 seconds.**



1) Step 1: MCU send out start signal to RHT03 and RHT03 send response signal to MCU

Data-bus's free status is high voltage level. When communication between MCU and RHT03 begins, MCU will pull low data-bus and this process must beyond at least 1~10ms to ensure RHT03 could detect MCU's signal, then MCU will pulls up and wait 20-40us for RHT03's response.

When RHT03 detect the start signal, RHT03 will pull low the bus 80us as response signal, then RHT03 pulls up 80us for preparation to send data. See below figure:

MaxDetect Technology Co., Ltd.

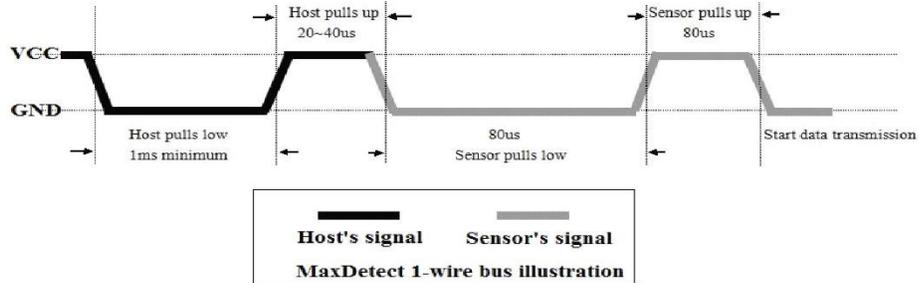
- 3 -

<http://www.humiditycn.com>

Thomas Liu (Sales Manager)
 Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

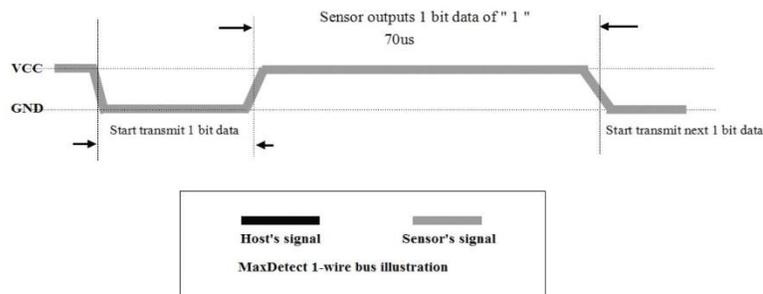
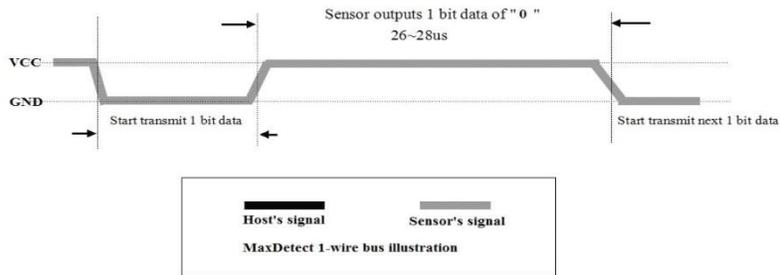


Your specialist in innovating humidity & temperature sensors



2). Step 2: RHT03 send data to MCU

When RHT03 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0". See below figures:



MaxDetect Technology Co., Ltd.

<http://www.humiditycn.com>

Thomas Liu (Sales Manager)
Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com



Your specialist in innovating humidity & temperature sensors

Attention:

If signal from RHT03 is always high-voltage-level, it means RHT03 is not working properly, please check the electrical connection status.

7. Electrical Characteristics:

Items	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	6	V
Current supply	Measuring	1		1.5	mA
	Stand-by	40	Null	50	uA
Collecting period	Second		2		Second

8. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The RHT03 sensor can recover after working in abnormal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere RHT03's sensitive-elements and debase RHT03's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the RHT03 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the RHT03 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

RHT03 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase RHT03's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius.

* Avoid using the sensor under dew condition.

* Don't use this product in safety or emergency stop devices or any other occasion that failure of RHT03 may cause personal injury.

MaxDetect Technology Co., Ltd.

- 5 -

<http://www.humiditycn.com>

Thomas Liu (Sales Manager)

Email: thomasliu198518@yahoo.com.cn , sales@humiditycn.com

A4. Sensor MPX5050DP

Freescale Semiconductor

MPX5050
Rev 11, 03/2010

**Integrated Silicon Pressure Sensor
On-Chip Signal Conditioned,
Temperature Compensated and
Calibrated**

The MPXx5050 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 2.5% Maximum Error over 0° to 85°C
- Ideally suited for Microprocessor or Microcontroller-Based Systems
- Temperature Compensated Over -40° to +125°C
- Patented Silicon Shear Stress Strain Gauge
- Durable Epoxy Unibody Element
- Easy-to-Use Chip Carrier Option

**MPX5050
MPXV5050
MPVZ5050
Series**
0 to 50 kPa (0 to 7.25 psi)
0.2 to 4.7 V Output

ORDERING INFORMATION

Device Name	Case No.	# of Ports			Pressure Type			Device Marking
		None	Single	Dual	Gauge	Differential	Absolute	
Unibody Package (MPX5050 Series)								
MPX5050D	867	*				*		MPX5050D
MPX5050DP	867C			*		*		MPX5050DP
MPX5050GP	867B		*		*			MPX5050GP
MPX5050GP1	867B		*		*			MPX5050GP
Small Outline Package (MPXV5050 Series)								
MPXV5050GP	1369		*		*			MPXV5050GP
MPXV5050DP	1351			*		*		MPXV5050DP
MPXV5050GC6U	482A		*		*			MPXV5050G
MPXV5050GC6T1	482A		*		*			MPXV5050G
Small Outline Package (Media Resistant Gel) (MPVZ5050 Series)								
MPVZ5050GW7U	1560		*		*			MZ5050GW

Operating Characteristics

Table 1. Operating Characteristics ($V_S = 5.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P1 > P2$. Decoupling circuit shown in Figure 4 required to meet electrical specifications.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾	P_{OP}	0	—	50	kPa
Supply Voltage ⁽²⁾	V_S	4.75	5.0	5.25	Vdc
Supply Current	I_o	—	7.0	10	mAdc
Minimum Pressure Offset ⁽³⁾ @ $V_S = 5.0$ Volts	V_{off}	0.088	0.2	0.313	Vdc
Full Scale Output ⁽⁴⁾ @ $V_S = 5.0$ Volts	V_{FSO}	4.587	4.7	4.813	Vdc
Full Scale Span ⁽⁵⁾ @ $V_S = 5.0$ Volts	V_{FSS}	—	4.5	—	Vdc
Accuracy ⁽⁶⁾	—	—	—	±2.5	% V_{FSS}
Sensitivity	V/P	—	90	—	mV/kPa
Response Time ⁽⁷⁾	t_R	—	1.0	—	ms
Output Source Current at Full Scale Output	I_{o+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms
Offset Stability ⁽⁹⁾	—	—	±0.5	—	% V_{FSS}

1. 1.0 kPa (kiloPascal) equals 0.145 psi.

2. Device is ratiometric within this specified excitation range.

3. Offset (V_{off}) is defined as the output voltage at the minimum rated pressure.

4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.

5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.

6. Accuracy (error budget) consists of the following:

Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.

Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.

Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from the minimum or maximum rated pressure at 25°C.

TcSpan: Output deviation over the temperature range of 0° to 85°C, relative to 25°C.

TcOffset: Output deviation with minimum pressure applied, over the temperature range of 0° to 85°C, relative to 25°C.

Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} at 25°C.

7. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.

8. Warm-up Time is defined as the time required for the product to meet the specified output voltage after the Pressure has been stabilized.

9. Offset Stability is the product's output deviation when subjected to 1000 hours of Pulsed Pressure, Temperature Cycling with Bias Test.

Pressure

Maximum Ratings

Table 2. Maximum Ratings⁽¹⁾

Rating	Symbol	Value	Unit
Maximum Pressure (P1 > P2)	P_{max}	200	kPa
Storage Temperature	T_{stg}	-40° to +125°	°C
Operating Temperature	T_A	-40° to +125°	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

Figure 1 shows a block diagram of the internal circuitry integrated on a pressure sensor chip.

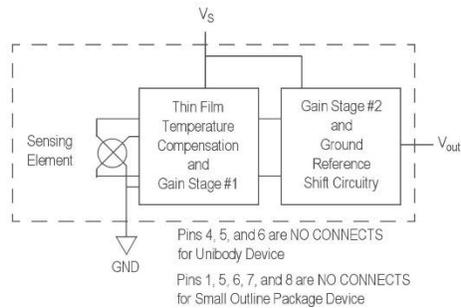


Figure 1. Fully Integrated Pressure Sensor Schematic

On-chip Temperature Compensation and Calibration

Figure 3 illustrates the Differential/Gauge Sensing Chip in the basic chip carrier (Case 867). A fluorosilicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the sensor diaphragm.

The MPX5050/MPXV5050G series pressure sensor operating characteristics, and internal reliability and qualification tests are based on use of dry air as the pressure media. Media, other than dry air, may have adverse effects on sensor performance and long-term reliability. Contact the

factory for information regarding media compatibility in your application.

Figure 2 shows the sensor output signal relative to pressure input. Typical, minimum, and maximum output curves are shown for operation over a temperature range of 0° to 85°C using the decoupling circuit shown in Figure 4. The output will saturate outside of the specified pressure range.

Figure 4 shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

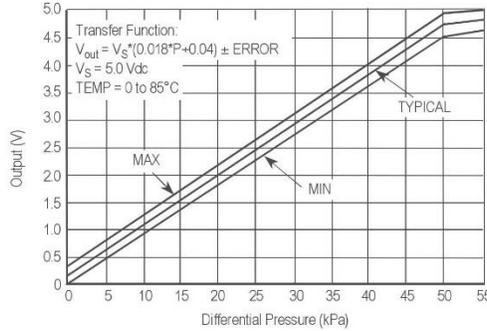


Figure 2. Output vs. Pressure Differential

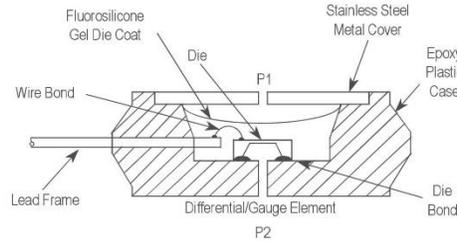


Figure 3. Cross-Sectional Diagram (not to scale)

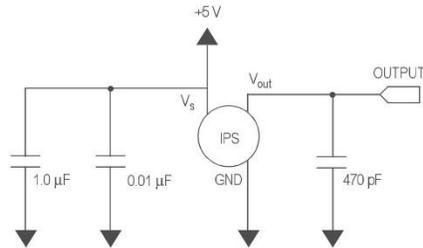


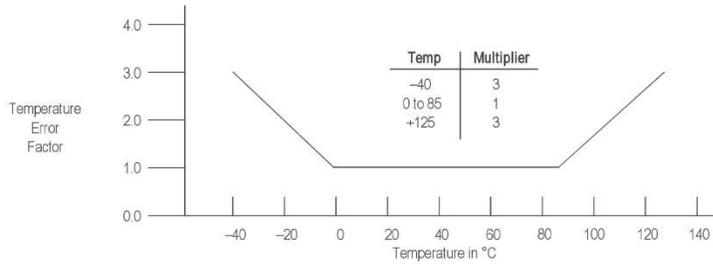
Figure 4. Recommended Power Supply Decoupling and Output Filtering
 (For additional output filtering, please refer to Application Note AN1646)

Pressure

Transfer Function

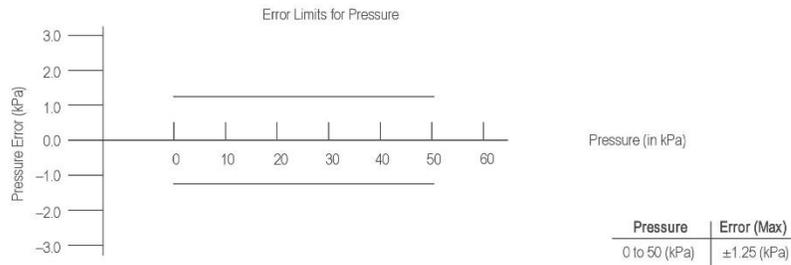
Nominal Transfer Value: $V_{out} = V_S (P \times 0.018 + 0.04)$
 $\pm (\text{Pressure Error} \times \text{Temp. Factor} \times 0.018 \times V_S)$
 $V_S = 5.0 \text{ V} \pm 0.25 \text{ Vdc}$

Temperature Error Band



NOTE: The Temperature Multiplier is a linear response from 0° to -40°C and from 85° to 125°C.

Pressure Error Band



PRESSURE (P1)/VACUUM (P2) SIDE IDENTIFICATION TABLE

Freescale designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing fluorosilicone gel which protects the die from harsh media. The MPX pressure

sensor is designed to operate with positive differential pressure applied, P1 > P2.

The Pressure (P1) side may be identified by using the table below:

Part Number	Case Type	Pressure (P1) Side Identifier
MPX5050D	867	Stainless Steel Cap
MPX5050DP	867C	Side with Part Marking
MPX5050GP	867B	Side with Port Attached
MPXV5050GP	1369	Side with Port Attached
MPXV5050DP	1351	Side with Part Marking
MPXV5050GC6U/T1	482A	Vertical Port Attached

PACKAGE DIMENSIONS

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. 867B-01 THRU -3 OBSOLETE, NEW STANDARD 867B-04.

STYLE 1:

- PIN 1: V OUT
 2: GROUND
 3: VCC
 4: V1
 5: V2
 6: V EX

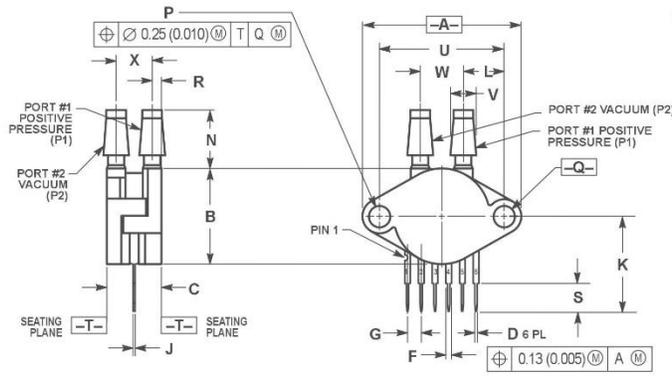
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE	
TITLE: SENSOR, 6 LEAD UNIBODY CELL, AP & GP 01ASB09087B	DOCUMENT NO: 98ASB42796B	REV: G	
	CASE NUMBER: 867B-04	28 JUL 2005	
	STANDARD: NON-JEDEC		

PAGE 2 OF 2

**CASE 867B-04
 ISSUE G
 UNIBODY PACKAGE**

Pressure

PACKAGE DIMENSIONS



NOTES:
 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982
 2. CONTROLLING DIMENSION: INCH

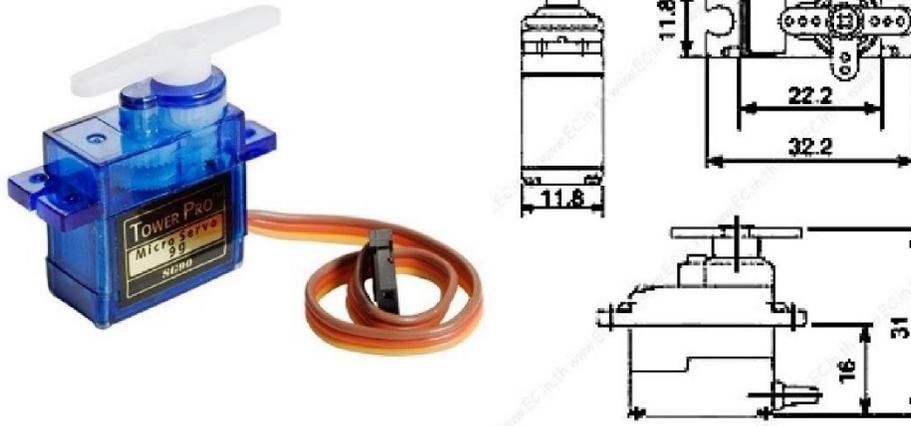
DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.145	1.175	29.08	29.65
B	0.685	0.715	17.40	18.16
C	0.405	0.435	10.29	11.05
D	0.027	0.033	0.68	0.84
F	0.048	0.064	1.22	1.63
G	0.100 BSC		2.54 BSC	
J	0.014	0.016	0.36	0.41
K	0.695	0.725	17.65	18.42
L	0.290	0.305	7.37	7.62
N	0.420	0.440	10.67	11.18
P	0.153	0.159	3.88	4.04
Q	0.153	0.159	3.88	4.04
R	0.083	0.083	1.60	2.11
S	0.220	0.240	5.59	6.10
U	0.910 BSC		23.11 BSC	
V	0.162	0.164	4.62	4.93
W	0.310	0.330	7.67	8.38
X	0.248	0.276	6.30	7.06

STYLE 1:
 PIN 1: V_{OUT}
 2: GROUND
 3: V_{CC}
 4: V_I
 5: V₂
 6: V_{EX}

CASE 867C-05
 ISSUE F
 UNIBODY PACKAGE

A5. Servomotor SG90

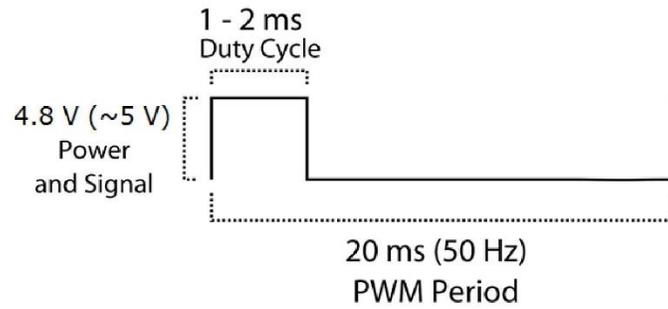
SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

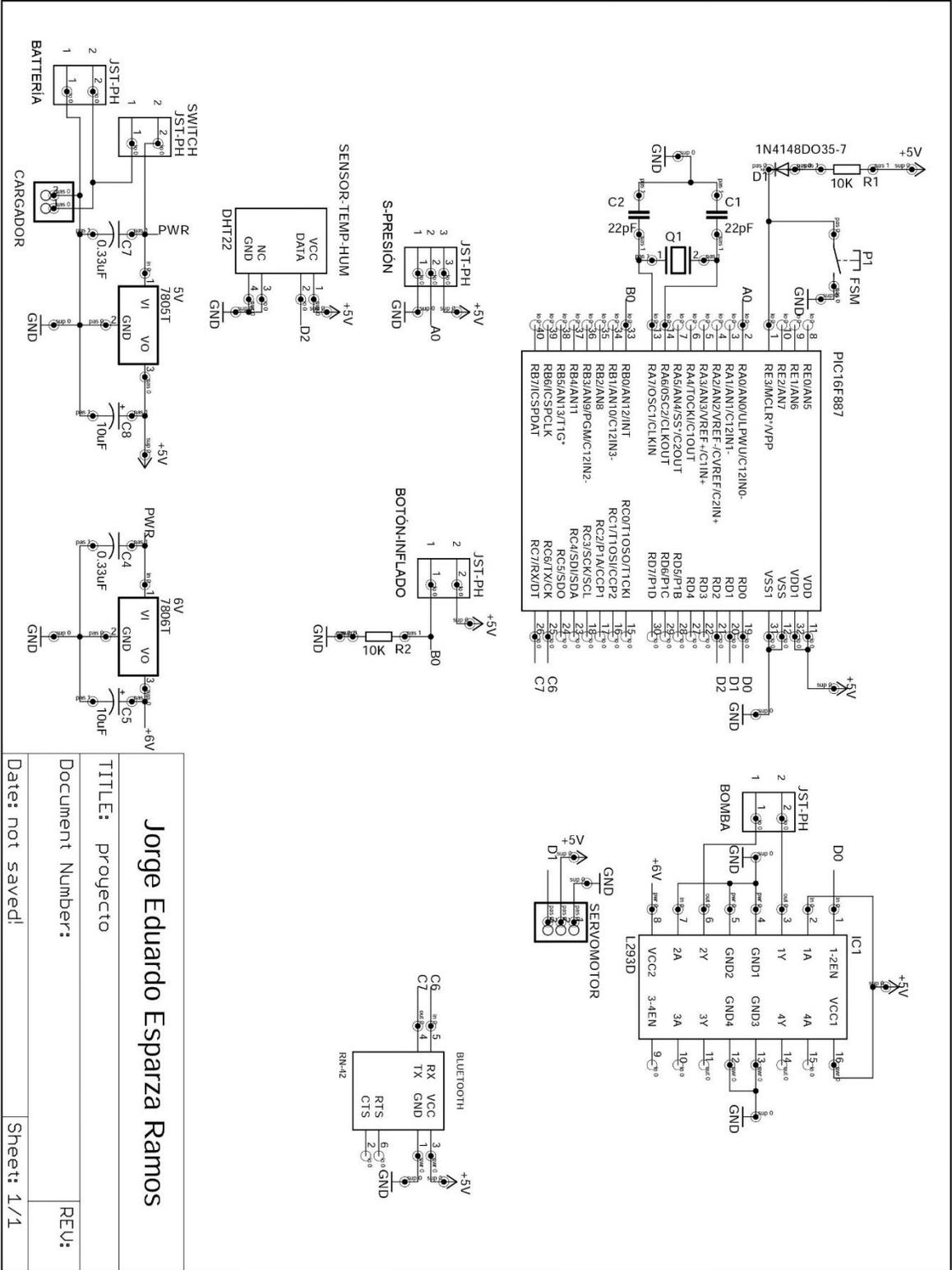
Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10 μ s
- Temperature range: 0 °C – 55 °C



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

A6. Esquema del circuito del sistema



Jorge Eduardo Esparza Ramos

TITLE: proyecto	REV: 1
Document Number: 1	REV: 1
Date: not saved!	Sheet: 1/1

A7. Código de programación del microcontrolador

La programación del algoritmo para la instrumentación del socket en el microcontrolador se llevó a cabo en lenguaje C. El código de programación se escribió en múltiples archivos dentro de un solo proyecto, para facilitar su comprensión. Se utilizó un archivo principal en el que se incluyeron los nombres de los archivos de cabecera necesarios para que funcionase el código, tanto de los archivos incluidos en el compilador, como de los creados específicamente para este trabajo. Para incluirlos se utilizó la directiva #include, seguida del nombre del archivo cabecera con terminación “.h”. En los archivos cabecera se han declarado las variables externas y los prototipos de las funciones, contenidas en los archivos librería, que son utilizadas dentro del programa desde cualquier otro archivo que lo requiera.

Además del archivo principal, se crearon tres archivos librería. El primero es el archivo “CONFIG.c”, el cual contiene el código necesario para establecer la configuración de los puertos, los timers, las interrupciones, etc.

El segundo archivo librería, creado en este proyecto, es el “DHT22.c”, el cual permite realizar el algoritmo para la lectura de la temperatura y humedad, además de almacenar esos datos en la memoria EEPROM.

El último archivo librería diseñado en este trabajo fue el “PI.c”, que contiene todo lo necesario para llevar a cabo el control PI de la presión en los sujetadores.

Cada uno de estos tres archivos tiene su propio archivo cabecera para poder utilizar el código de estos en el archivo principal. Sus nombres son “CONFIG.h”, “DHT22.h” y “PI.h” respectivamente.

La función principal del proyecto se encuentra en el archivo “Principal.c”, el cual contiene el siguiente código:

```
#include <16f887.h>
#include "CONFIG.h"
#include "DHT22.h"
#include "PID.h"
#device ADC=8
#use delay(internal=4MHz)
#fuses XT, NOWDT, MCLR
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#bit TMR2ON = 18.2 // T2CON.TMR2ON
#bit Servo = 8.1 // PORTD.1
#define TMR_TH 3636 // Se compara con t0int en INT_TIMER0 para controlar cuándo es llamada DHT22()
// Si TMR_TH = 1818, DHT22() es llamada cada 1 minuto aprox.
// Si TMR_TH = 303, DHT22() es llamada cada 10 segundos
int1 y = 0; // Variable auxiliar en TMR0 que permite a DHT22() ser llamada
extern int1 TOUT; // Variable auxiliar que indica desbordamiento del TMR2
extern int i; // Variable auxiliar para ciclos for
extern float T_Byte; // Guardan temperatura y
extern float RH_Byte; // Guarda humedad respectivamente
```

```

int getP = 70;           // Guarda el byte recibido para la presión
char a = 'A';          // Guardan caracteres recibidos desde el módulo EUSART
char b = 'E';          // Guardan caracteres recibidos desde el módulo EUSART
char c = 'E';          // Guardan caracteres recibidos desde el módulo EUSART
long duty = 1475;      // Guarda el ciclo de trabajo en  $\mu$ s para crear PWM
long max = 270;        // Guarda el valor máximo que puede sumarse al ciclo de trabajo "duty"
long muestreo = 0;     // Variable para bucle de tiempo de muestreo del PID
long t0int = 0;        // Variable auxiliar para el TMR0
float ref = 70f;        // Guarda la referencia de presión en milímetros de mercurio
float kp = 1.6834;     // Constante proporcional del control PI
float ki = 6.7335;     // Constante integral del control PI
float Tm = 0.5;        // Tiempo de muestreo

//===== Interrupción al recibir datos por RS232 =====
#INT_RDA
void serial_interrupt(){
    if(a=='A') a=getc();
    if(b=='P'){
        getP=getc();
        if((getP=='C')&&(c=='E')) c=getP; // Para comprobar PID
        if(getP=='A') a = getP;
    }
    b=a;
}

//===== Interrupción por desbordamiento de TMR0 =====
#INT_TIMER0
void interruptT0(){
    set_timer0(0);
    if(t0int==TMR_TH){
        t0int = 0;
        y = 1; // Esta declaración permite que la función DHT22() pueda ser llamada
    }
    t0int++;
}

//===== Interrupción por desbordamiento del TMR1 =====
#INT_TIMER1
void PWM(){
    set_timer1(63040);
    Servo = 1;
    delay_us(duty);
    Servo = 0;
}

//===== Interrupción por desbordamiento del TMR2 =====
#INT_TIMER2
void interruptT2(){
    TOUT = 1;
    TMR2ON = 0; // Detiene el TMR2
}

//===== Interrupción externa =====

```

```

#INT_EXT
void inflar(){
    delay_ms(500); // Filtro
    if (getP<60)    getP = 70;
    else           getP = 1;
}

//=====
//===== Método main =====
//=====
void main(){
    config();
    while(1){
        if(muestreo==11780){
            pid(getP, ref, kp, ki, Tm, max, &duty);
            muestrero=0;
        }
        muestrero++;

        if((a=='A')&&(y==1)){ // Entra cada n TMRO (y==1) y cada que a=='A'
            DHT22();
            y=0;
        }

        if(a=='T'){
            for (i=0; i<30; i++){
                putc(read_eeprom(i));
                putc(read_eeprom(i+30));
            }
            a='A';
        }

        if(a=='B'){ // Si se presiona el botón actualizar (el cual manda 'B')
            putc((int)RH_Byte);
            putc((int)T_Byte);
            a='A';
        }
    }
}

```

Como se puede observar, este último código contiene los nombres de los archivos cabecera ya mencionados con anterioridad. El primer archivo cabecera “CONFIG.h” y su archivo librería “CONFIG.c” correspondiente, se muestran a continuación.

Archivo cabecera “CONFIG.h”:

```

#ifndef CONFIG_H
#define CONFIG_H

void config();

```

#endif

Archivo librería "CONFIG.c":

```
#include <16f887.h>
#use delay(internal=4MHz)
#fuses XT, NOWDT, MCLR
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#bit PumpDir = 136.0 //TRISD.0
#bit ServoDir = 136.1 //TRISD.1
#bit Pump = 8.0 //PORTD.0
#bit Servo = 8.1 //PORTD.1

//===== Método de configuración =====
void config(){
    ServoDir = 0; // Se configura al pin del Servo como salida
    Servo = 0; // Pone en bajo al pin del Servo a ser usado como módulo PWM
    PumpDir = 0; // Se configura al pin de la bomba como salida
    Pump = 0; // Pone en bajo al pin de la bomba

    //Configuración del ADC
    setup_adc(ADC_CLOCK_INTERNAL); //Reloj de conversión
    setup_adc_ports(sANO); //PA.0, es analógico
    set_adc_channel(0); //Seleccionar canal 0 para leer
    delay_us(10); //Espera 10 µs para conversión

    //Configuración de timers
    setup_timer_0(RTCC_INTERNAL | T0_DIV_128); // Configura al timer0
    setup_timer_1(T1_INTERNAL | T1_DIV_BY_8); // Configura al timer1
    set_timer1(63040); // Valor inicial del TMR1 (interrupción cada 20 ms)

    //Configuración de interrupciones
    disable_interrupts(GLOBAL); // Permiso global de interrupción

    clear_interrupt(INT_EXT); // Limpia bandera de interrupción EXT
    enable_interrupts(INT_EXT); // Permiso de interrupción EXT

    clear_interrupt(INT_TIMER0); // Limpia bandera de interrupción del TMRO
    enable_interrupts(INT_TIMER0); // Permiso de interrupción del TMRO
    set_timer0(0);

    clear_interrupt(INT_TIMER1); //Limpia bandera de interrupción del TMR1
    enable_interrupts(INT_TIMER1); //Permiso de interrupción del TMR1

    clear_interrupt(INT_RDA); // Limpia bandera de interrupción por EUSART
    enable_interrupts(INT_RDA); // Permiso de interrupción por EUSART

    enable_interrupts(GLOBAL); // Permiso global de interrupción
}

```

Los siguientes dos archivos corresponden al archivo cabecera "DHT22.h" y al archivo librería "DHT22.c" del sensor de temperatura y humedad, y son mostrados a continuación.

Archivo cabecera "DHT22.h":

```
#ifndef DHT22_H
#define DHT22_H

int1 TOUT;    // Variable auxiliar que indica desbordamiento del TMR2
int i;        // Variable auxiliar para ciclos for
float T_Byte; // Guardan temperatura y
float RH_Byte; // Guarda humedad respectivamente

void DHT22();

// Señal de inicio del MCU hacia el DHT22
void StartSignal();

// Revisión de respuesta del DHT22 hacia MCU
int1 CheckResponse();

// Lectura de Bytes del DHT22 hacia MCU
int ReadByte();

#endif
```

Archivo librería "DHT22.c":

```
#include <16f887.h>
#use delay(internal=4MHz)
#fuses XT, NOWDT, MCLR
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#bit DataDir = 136.2 //TRISD.2
#bit Data = 8.2 //PORTD.2
#bit TMR2ON = 18.2 //T2CON.TMR2ON

int1 check; // Variable para confirmar respuesta del sensor DHT22
int1 TOUT = 0; // Variable auxiliar que indica desbordamiento del TMR2
int i; // Variable auxiliar para ciclos for
int CheckSum; // Guarda el byte de CheckSum enviado por el DHT22
int ep = 0; // Utilizada como índice de arreglos para guardar datos en la EEPROM
int RHep[60], Tep[60]; // Guardan múltiples valores de T y H para guardarlos en la EEPROM
int T_Byte2, RH_Byte2; // MSB recibidos de temperatura y humedad respectivamente
long T_Byte1, RH_Byte1; // LSB recibidos de temperatura y humedad respectivamente
float T_Byte = 0; // Guardan temperatura y
float RH_Byte = 0; // Guarda humedad respectivamente

//===== Señal de inicio del MCU hacia el DHT22 =====
void StartSignal(){
    setup_timer_2(T2_DIV_BY_4,255,1); // prescala 1:4, PR2=255, pos 1:1
    DataDir = 0; // Se configura al puerto de datos como salida
```

```

Data = 0;           // Pone en bajo la línea de datos hacia el DHT22
TOUT = 0;          // Variable auxiliar que indica desbordamiento del TMR2 es puesta a cero
clear_interrupt(INT_TIMER2); // Limpia bandera de interrupción del TMR2
enable_interrupts(INT_TIMER2); // Permiso de interrupción del TMR2
TMR2ON = 1;        // Activa el TMR2
set_timer2(0);     // Se pone el TMR2 a cero
while(TOUT==0);   // Espera al menos 1 ms antes de poner en alto a Data
disable_interrupts(INT_TIMER2); // Permiso de interrupción del TMR2
Data = 1;          // Pone en alto la línea de datos hacia el DHT22
delay_us(22);      // Espera entre 20 y 40 µs antes de configurarlo como entrada
DataDir = 1;       // Se configura al puerto de datos como entrada
}

//===== Revisión de respuesta del DHT22 hacia MCU =====
int1 CheckResponse(){
    setup_timer_2(T2_DIV_BY_1,255,1); // prescala 1:1, PR2=255, pos 1:1
    TOUT = 0; // Variable auxiliar que indica desbordamiento del TMR2 es puesta a cero
    clear_interrupt(INT_TIMER2); // Limpia bandera de interrupción del TMR2
    enable_interrupts(INT_TIMER2); // Activa permiso de interrupción del TMR2
    set_timer2(0); // Se pone el TMR2 a cero
    while(!Data && !TOUT); // Espera respuesta del sensor o desbordamiento de TMR2
    if (TOUT) return 0; // Si hubo desbordamiento (TOUT = 1), regresa cero
    else { // Si hubo respuesta del sensor (Data = 1)
        set_timer2(0); // Se pone el TMR2 a cero
        while(Data && (!TOUT)); // Espera respuesta de sensor o desbordamiento de TMR2
        if (TOUT) return 0; // Si hubo desbordamiento (TOUT = 1), regresa cero
        else { // Si hubo respuesta del sensor (Data = 0)
            TMR2ON = 0; // Se apaga el TMR2
            return 1; // Regresa uno
        }
    }
}

//===== Lectura de Bytes del DHT22 hacia MCU =====
int ReadByte(){
    int num = 0; // Variable auxiliar que almacena el Byte recibido es puesta a cero
    DataDir = 1; // Se configura al puerto de datos como entrada
    for (i=0; i<8; i++){ // Ciclo for para almacenar 8 bits recibidos
        while(!Data); // Se espera a que el sensor cambie a alto
        set_timer2(0); // Se pone el TMR2 a cero
        TMR2ON = 1; // Se enciende el TMR2
        while(Data); // Se espera a que el sensor cambie a bajo
        TMR2ON = 0; // Se apaga el TMR2
        if(get_timer2() > 45){ // Si el DHT22 dura en alto más de 28 us, entonces el bit es 1
            num|=(1<<(7-i)); // Corrimiento de bit a izquierda, OR inclusiva con num, guardado en num
        }
    }
    return num; // Regresa el Byte recibido en la variable num
}

//===== Método DHT22 =====
void DHT22(){
    StartSignal(); // Envía de la señal de inicio hacia el sensor
}

```

```

check = CheckResponse(); // Revisa si hubo respuesta del sensor

if(check){ // Si sí hubo respuesta del DHT22
    clear_interrupt(INT_TIMER0); // Limpia bandera de interrupción del TMRO
    disable_interrupts(INT_TIMER0); // Permiso de interrupción del TMRO
    clear_interrupt(INT_TIMER1); // Limpia bandera de interrupción del TMR1
    disable_interrupts(INT_TIMER1); // Permiso de interrupción del TMR1
    clear_interrupt(INT_RDA); // Limpia bandera de interrupción por EUSART
    disable_interrupts(INT_RDA); // Permiso de interrupción por EUSART
    clear_interrupt(INT_EXT); // Limpia bandera de interrupción EXT
    disable_interrupts(INT_EXT); // Permiso de interrupción EXT

    RH_Byte1 = ReadByte(); // MCU lee y guarda Bytes enviados del DHT22
    RH_Byte2 = ReadByte();
    T_Byte1 = ReadByte();
    T_Byte2 = ReadByte();
    CheckSum = ReadByte(); // Revisa si hubo error al recibir los datos

    clear_interrupt(INT_TIMER0); //Limpia bandera de interrupción del TMRO
    enable_interrupts(INT_TIMER0); //Permiso de interrupción del TMRO
    clear_interrupt(INT_TIMER1); //Limpia bandera de interrupción del TMR1
    enable_interrupts(INT_TIMER1); //Permiso de interrupción del TMR1
    clear_interrupt(INT_RDA); // Limpia bandera de interrupción por EUSART
    enable_interrupts(INT_RDA); // Permiso de interrupción por EUSART
    clear_interrupt(INT_EXT); // Limpia bandera de interrupción EXT
    enable_interrupts(INT_EXT); // Permiso de interrupción EXT

    disable_interrupts(INT_TIMER2); // Desactiva permiso de interrupción del TMR2

    // Los datos fueron recibidos de la siguiente forma: 16 bits de RH + 16 bits
    // de Temperatura + 8 bits de CheckSum
    // Si la transmisión de datos fue correcta, CheckSum es los 8 últimos bits
    // de: RH_Byte1 + RH_Byte2 + T_Byte1 + T_Byte2
    if (CheckSum == ((RH_Byte1 + RH_Byte2 + T_Byte1 + T_Byte2) & 0xFF)){
        // Si no hubo error, manda los datos por USART
        RH_Byte = RH_Byte2 + (RH_Byte1<<7); // 16 bits de RH
        RH_Byte = RH_Byte / 10; // El valor real de RH es la décima parte
        // del dato de los 16 bits de RH
        T_Byte = T_Byte2 + (T_Byte1<<7); // 16 bits de T
        T_Byte /= 10; // El valor real de T es la décima parte
        // del dato obtenido de los 16 bits de T

        if(ep<30){
            RHep[ep]=RH_Byte;
            Tep[ep]=T_Byte;
            ep++;
        }
        else{ // Guarda datos de T y RH en EEPROM
            ep=0;
            for (i=0; i<30; i++){
                write_eeprom(i, RHep[i]);
                write_eeprom(i+30, Tep[i]);
            }
        }
    }
}

```

```

    }
}
}

```

Los últimos dos archivos son relativos al control del sistema de autoajuste, y corresponden al archivo cabecera "PI.h" y al archivo librería "PI.c" del sensor de temperatura y humedad, los cuales son mostrados a continuación.

Archivo cabecera "PI.h":

```

#ifndef PID_H
#define PID_H

void pid(int getP, float ref, float kp, float ki, float Tm, long max, long *duty);

#endif

```

Archivo librería "PI.c":

```

#include <16f887.h>
#include delay(internal=4MHz)
#include XT, NOWDT, MCLR
#include rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#include bit Pump = 8.0 // PORTD.0
extern char c; // Guardan caracteres recibidos desde el módulo EUSART
float press; // Guarda valor de presión en milímetros de mercurio leído por el sensor
float press_prev = 0; // Variable que sirve para filtrar el sensor MPX5050DP
float eT = 0f; // Guarda el error entre la referencia y la salida
float integral = 0f; // Guarda la integral discretizada del error
float uT; // Guarda la salida del control PI

void pid(int getP, float ref, float kp, float ki, float Tm, long max, long *duty){

    if((getP>69)&&(getP<91) || (getP==1)) ref=getP; // Si getP está en el rango de 70 a 90 mmHg,
                                                    // se guarda en ref

    // Conversión de ADC a presión en mmHg sabiendo que V=(0.09*P(kPa) + 0.2)
    press = 0.32682426364835*(5.0*read_adc() - 51.0);
    if((press_prev + 100)<press) press = press_prev; // Filtro del sensor de presión

    if (press>=0){ // En caso que no exista alguna falla en el sensor
        press_prev = press; // Guarda la presión actual en press_prev
        eT=ref-press; // Se calcula el error
    }
    else press = 0;

    if(eT!=0){ // Si existe error, se aplica el control
        integral += ki*Tm*eT; // Se calcula la integral del error
        eT = abs(eT); // Obtiene el valor absoluto del error
    }
}

```

```

if(eT<20) uT = kp*eT + integral; // Si el error es pequeño, aplica PI
else{ // Si el error es grande,
    uT = kp*eT; // aplica control P
    integral = 0; // El valor de la integral del error se limpia
}

uT = abs(uT); // Obtiene el valor absoluto de la salida

if (uT>((float)max)) uT=max; // Si uT supera el max permitido, se iguala a max

if((ref>press)&&(eT>2f)){ //Inflar // Si falta presión y hay error mayor a 5,
    Pump=1; // se activa la bomba y
    //uT debe estar entre 0 y 250 // se abre la válvula para inflar
    duty = 2125f + uT;
}
else if((press>ref)&&(eT>5f)){ //Desinflar // Si sobra presión y hay error mayor a 10,
    Pump=0; // se desactiva la bomba y
    //uT debe estar entre 0 y 250
    duty = 775f - uT; // se abre la válvula para desinflar
}
else{ // Si no sobra ni falta mucha presión
    Pump = 0; // apaga la bomba y
    duty = 1425; // bloquea la válvula
}
}
else{ // Si no hay error
    Pump = 0; // apaga la bomba y
    if(press>=0) duty = 1425; // si no hay problemas con el sensor, bloquea la válvula
}
if (c == 'C'){ // si se recibe una 'C' desde el Bluetooth
    putc((int)press); // se envían los datos de presión
    c = 'E';
}
}
}

```

A8. Programación de la aplicación desarrollada en App Inventor

La aplicación desarrollada en Android, mediante la herramienta de MIT App Inventor, consta de la parte del diseño visual y los bloques para la programación. Para poder comprender la programación en bloques es necesario definir qué es lo que representa cada variable.

El diseño de la aplicación en App Inventor complicó la utilización del módulo de Bluetooth del dispositivo móvil al momento de cambiar entre las diferentes pantallas, conocidas como actividades. La complicación principal fue que en el momento de cambiar de una pantalla a otra, el Bluetooth del teléfono seguía conectado al módulo del socket, pero el cliente de Bluetooth en la aplicación era inaccesible. Una posible solución hubiese podido ser el desconectar los módulos de Bluetooth durante el tiempo en el que se cambiara de pantalla, y posteriormente volver a conectarlos. Sin embargo, este proceso duraba muchos segundos, lo que hacía que no fuese práctico esperar hasta que se estableciera una nueva conexión. La solución implementada en este trabajo, fue el simular la existencia de pantallas (actividades), a través de layouts. Las layouts en este diseño, contienen los elementos como los botones, las gráficas, los textos y las imágenes que una pantalla tendría. Para poder simularlas se activa o desactiva la visibilidad de las distintas layouts haciendo que se eficiente la velocidad de la aplicación y la conexión entre los módulos de Bluetooth, sin tener que cambiar entre pantallas. A continuación se describen las diferentes layouts presentes en el diseño de la aplicación, mediante la herramienta de App Inventor; además, se especifican los nombres utilizados para la programación en bloques.

- Para desplegar la pantalla principal, mostrada en la figura 3.27, se activa la visualización de la layout llamada en el área de bloques como “ventana_1”. La visualización de las demás layouts permanece desactivada.
- La siguiente pantalla o actividad, que aparece al presionar el botón iniciar, y en la que aparecen los botones de presión, temperatura y humedad, e información, aparece con el nombre de “ventana_2”. De igual manera que la anterior layout, se desactivan las visualizaciones de las otras layouts y se cambia a activa la de esta layout.
- La pantalla del control y monitoreo de la presión aparece con el nombre de “ventana_presión”, y para poder visualizarla, se realiza el procedimiento análogo al de las anteriores layout, descrito previamente.
- La última visualización es en la que aparece el monitoreo de la temperatura y la humedad, y es nombrada “ventana_TyH” para ser utilizada así en el área de bloques.

En la visualización principal de la aplicación, como ya se mencionó previamente, existen cuatro botones, en el área de bloques, cada uno tiene un nombre específico:

- El botón “iniciar” aparece en el área de bloques como “Iniciar”.
- El botón “conectar socket” aparece en el área de bloques como “Conectar”. Existe otra variable llamada “lp_Conectar” la cual hace referencia a un botón con el mismo símbolo de “conectar socket” en la visualización de la actividad principal, pero que al ser presionado, despliega una lista de elementos a seleccionar, ya que es un

selector de lista. Este botón aparece únicamente cuando es necesario seleccionar el módulo de Bluetooth del socket para conectarse. Visualmente no se distingue la diferencia entre estos dos botones pues cumplen funciones similares y nunca están visibles al mismo tiempo.

- Una vez que se conectó el dispositivo móvil con el Bluetooth del socket (después de presionar el botón “lp_Conectar”), la MAC del módulo es almacenada en una pequeña base de datos que aparece en el área de bloques como “db_MAC”.
- El botón “emparejar” aparece en el área de bloques como “EmparejarBT”, e invoca a una acción llamada “as_EmparejarBT” que abre la configuración del Bluetooth en el dispositivo móvil mediante el código android.settings.BLUETOOTH_SETTINGS
- El botón “desconectar” aparece en el área de bloques como “desconectar”.
- El botón “ayuda” aparece en el área de bloques como “Ayuda”. Este botón invoca a un notificador con el nombre de “Aviso”.
- El botón “salir” de la visualización principal aparece en el área de bloques como “Salir”.
- También se define un Cliente Bluetooth con el nombre “BluetoothClient1”.

La siguiente visualización es donde aparecen los botones para seleccionar la opción de ver los datos de presión, o los datos de temperatura y humedad, o solicitar información, y cada uno tiene su nombre específico en el área de bloques:

- El botón “presión” aparece en el área de bloques como “Presion”.
- El botón “temperatura y humedad” aparece en el área de bloques como “TyH”.
- El botón “información” aparece en el área de bloques como “informacionPTH”, el cual invoca a un notificador con el nombre de “Aviso”.
- El botón “salir” de esta visualización, aparece en el área de bloques como “retornar1”.

La visualización correspondiente a la presión tiene diversos elementos que serán descritos a continuación, para el correcto entendimiento de la programación en bloques:

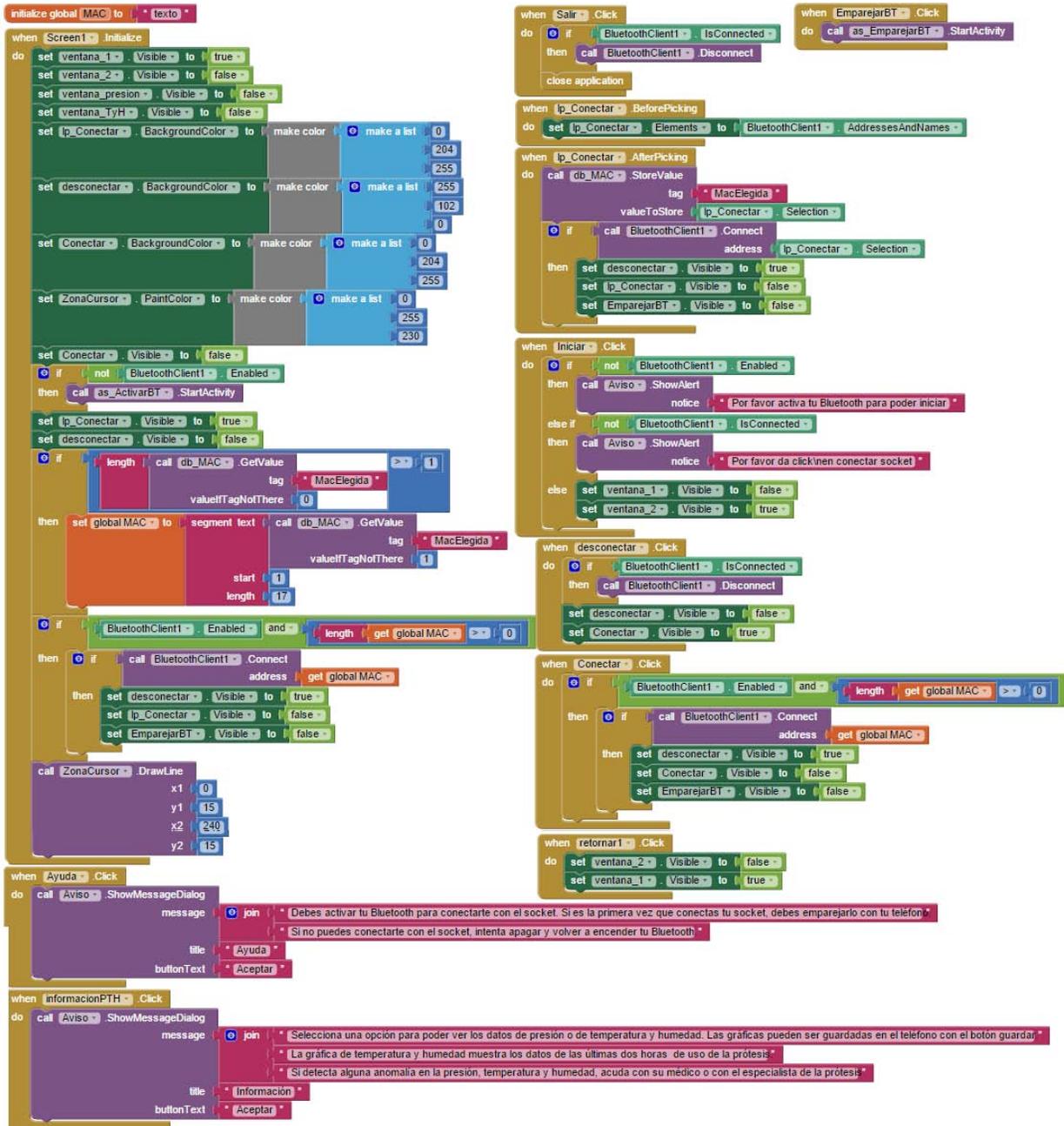
- Para mostrar el valor de referencia de la presión en milímetros de mercurio se utiliza una etiqueta con el nombre “valorPresion” para ser utilizada en el área de bloques.
- El elemento de desplazamiento en color agua se logra mediante un canvas (lienzo) y una esfera o pelota. En el área de bloques, se les llamó “ZonaCursor” y “Cursor”, respectivamente. Este elemento es utilizado para ajustar la presión de referencia dentro de un rango de 70 a 90 mmHg, como se mencionó en el capítulo del diseño de configuración.
- La gráfica de la presión, para verificar el funcionamiento del control, se logra mediante otro canvas llamado “GraphPID”. Los valores de la gráfica son mostrados cada 500 ms gracias al reloj “ClockPID”.

- En la parte inferior de esta visualización se encuentra el botón inflar, que originalmente aparece oculto hasta presionar al botón desinflar. El nombre con el que aparece en el área de bloques es “inflar”.
- El otro botón, ya mencionado previamente, es el de desinflar, el cual desaparece cuando es presionado el botón de inflar. En el área de bloques, aparece con el nombre de “desinflar”.
- Para visualizar u ocultar la gráfica de presión, se utiliza el botón PID, el cual aparece con el mismo nombre de “PID” en el área de bloques.
- Con estos últimos botones, al presionarlos, se realiza también la comunicación con el módulo de Bluetooth del socket, por lo que para evitar error en el envío de datos, se implementó un reloj que funciona como retraso, y aparece en área de bloques con el nombre de “DelayClockEnvío”.
- Otro botón que aparece en esta visualización es el de pausa, que detiene la graficación de la presión de referencia y la presión actual en los sujetadores. Su nombre en el área de bloques es “Pausa”.
- Para guardar la gráfica en el dispositivo móvil, se utiliza el botón con el símbolo de un diskette, y lleva el nombre de “guardarPID” en el área de bloques. Antes de realizar la acción de guardar la gráfica, le pregunta al usuario si desea guardarla. Para esto, se despliega un notificador con el nombre de “aviso_guardarPID”.
- El último botón que aparece es el de retornar, que precisamente regresa a la visualización anterior, y aparece en área de bloques con el nombre de “retornar_a_v2_desde_presion”.

La última visualización es la del monitoreo de la temperatura y la humedad, la cual tiene los siguientes elementos:

- El valor de la temperatura y de la humedad, se muestran en el área de bloques mediante los nombres “valor_T” y “valor_H”, respectivamente.
- La gráfica de temperatura y humedad es un canvas (lienzo) que aparece en el área de bloques con el nombre de “GraphTyH”.
- El botón en color verde, es el botón para actualizar los valores de temperatura y humedad en las etiquetas “valor_T” y “valor_H”. Para hacerlo, le pide al microcontrolador que le envíe dichos datos mediante el módulo de Bluetooth. Este botón aparece con el nombre de “actualizar_TH”.
- Para guardar la gráfica de temperatura y humedad, se utiliza el botón “guardarTH”, el cual tiene un símbolo de un diskette, y realiza la misma función que la del botón guardar en la pantalla de la presión. De igual forma, al presionar este botón, se despliega un aviso para que el usuario decida si realmente desea guardar la gráfica de temperatura y humedad mediante el elemento “aviso_guardarTH”.
- El botón de color rojo con la flecha, sirve para activar la visibilidad del layout “ventana_2” y desactivar los demás layouts que simulan a las actividades (pantallas), y aparece en el área de bloques con el nombre de “retornar_a_v2_desde_tyh”.

Los bloques de la programación en Android, a través de la herramienta MIT App Inventor, se muestran a continuación:



INSTRUMENTACIÓN DE UN SOCKET PROTÉSICO A NIVEL TRANSFEMORAL

```

when Presion - Click
do
  set ventana_2 - Visible to false
  set GraphPID - Visible to false
  call BluetoothClient1 - SendText
  text: P
  set ventana_presion - Visible to true
  initialize global PresionRef to 70
  initialize global PrevPresion to 0
  initialize global PrevPresionRef to 70
  initialize global auxP to 0
  initialize global envioBT to 0
  initialize global GraphPID to 0
  initialize global Xindex_PID to 25
  initialize global Pausa to 0
  initialize global TiempoFecha to *
  initialize global Ubicacion to 0

when ClockPID - Timer
do
  if [ventana_presion - Visible - and - GraphPID - Visible - and - get global envioBT == 0]
  then
    call BluetoothClient1 - SendText
    text: G
    if [get global Xindex_PID == 320]
    then
      call GraphPID - Clear
      set global Xindex_PID to 25
      initialize local Presion to call BluetoothClient1 - ReceiveUnsigned1ByteNumber
      in call DrawLineSegment_PID
      LineColor:
      CurrentDegrees: get Presion
      PrevDegrees: get global PrevPresion
      Offset: 197
      set global PrevPresion to get Presion
      set global auxP to get Presion
      call DrawLineSegment_PID
      LineColor:
      CurrentDegrees: get global PresionRef
      PrevDegrees: get global PrevPresionRef
      Offset: 197
      set global PrevPresionRef to get global PresionRef
      set global Xindex_PID to get global Xindex_PID + 9
      to AdjustPID value Offset
      result: round (get Offset - 0) / 1 * get value

when Infiar - Click
do
  set global envioBT to 1
  set DelayClockEnvio - TimerEnabled to true
  set ZonaCursor - Visible to true
  set Infiar - Visible to false
  set desinfiar - Visible to true
  set global PresionRef to 70
  set Crusor - X to 1
  set ValorPresion - Text to get global PresionRef
  call BluetoothClient1 - Send1ByteNumber
  number: get global PresionRef

when desinfiar - Click
do
  set global envioBT to 1
  set DelayClockEnvio - TimerEnabled to true
  set ZonaCursor - Visible to false
  set desinfiar - Visible to false
  set Infiar - Visible to true
  set global PresionRef to 1
  set Crusor - X to 1
  set ValorPresion - Text to get global PresionRef
  call BluetoothClient1 - Send1ByteNumber
  number: get global PresionRef

when DelayClockEnvio - Timer
do
  set global envioBT to 0
  set DelayClockEnvio - TimerEnabled to false

when retornar_a_v2_desde_presion - Click
do
  call BluetoothClient1 - SendText
  text: A
  set ventana_presion - Visible to false
  set ventana_2 - Visible to true
  set GraphPID - Visible to false
  set PID - Visible to true
  set ClockPID - TimerAlwaysFires to false
  set Pausa - Visible to false
  set guardarPID - Visible to false
  set global GraphPID to 0

when aviso_guardarPID - AfterChoosing
choice
do
  if [get choice == * SI *]
  then
    call Aviso - ShowAlert
    notice: Guardado en:
    call GraphPID - SaveAs
    fileName: join (/Graphics/
    call DelayClockEnvio - Hour
    instant: get global TiempoFecha
    call DelayClockEnvio - Minute
    instant: get global TiempoFecha
    call DelayClockEnvio - Second
    instant: get global TiempoFecha
    *
    call DelayClockEnvio - DayOfMonth
    instant: get global TiempoFecha
    call DelayClockEnvio - Month
    instant: get global TiempoFecha
    call DelayClockEnvio - Year
    instant: get global TiempoFecha
    * graphPID *
    *.pp *
  else
  end

when ZonaCursor - Touched
x y touchedAnySprite
do
  if [get x <= get global PresionRef]
  then
    if [get global Ubicacion >= 0]
    then
      set global Ubicacion to get global Ubicacion - 1
    else
    if [get global Ubicacion <= 255]
    then
      set global Ubicacion to get global Ubicacion + 1
      set Crusor - X to get global Ubicacion
      set global PresionRef to get global Ubicacion / 12 + 70
      set ValorPresion - Text to get global PresionRef
  when ZonaCursor - Dragged
startX startY prevX prevY currentX currentY draggedAnySprite
do
  set global envioBT to 1
  if [get draggedAnySprite]
  then
    set Crusor - X to get currentX
    if [get currentX <= 241]
    then
      set global Ubicacion to get currentX
    else
      set global Ubicacion to 240
      set global PresionRef to round (get global Ubicacion / 12) + 70
      set ValorPresion - Text to get global PresionRef
      set global envioBT to 1
      set DelayClockEnvio - TimerEnabled to true
      call BluetoothClient1 - Send1ByteNumber
      number: get global PresionRef
      set global envioBT to 0
  to DrawLineSegment_PID LineColor CurrentDegrees PrevDegrees Offset
do
  set GraphPID - LineWidth to 3
  call GraphPID - DrawPoint
  x: get global Xindex_PID
  y: call AdjustPID
  value: get PrevDegrees
  Offset: get Offset
  set GraphPID - LineWidth to 2
  set GraphPID - PaintColor to get LineColor
  call GraphPID - DrawLine
  x1: get global Xindex_PID
  call AdjustPID
  value: get PrevDegrees
  Offset: get Offset
  x2: get global Xindex_PID + 9
  y2: call AdjustPID
  value: get CurrentDegrees
  Offset: get Offset
  when PID - Click
do
  if [get global GraphPID == 0]
  then
    set global GraphPID to 1
    set GraphPID - Visible to true
    set ClockPID - TimerEnabled to true
    set global Xindex_PID to 25
    set guardarPID - Visible to true
    set Pausa - Visible to true
  else
    set global GraphPID to 0
    set ClockPID - TimerEnabled to false
    set GraphPID - Visible to false
    call GraphPID - Clear
    set guardarPID - Visible to false
    set Pausa - Visible to false
  end
  
```

```

initialize global PrevTemperatura to 0
initialize global PrevHumedad to 0
initialize global XIndex to 25
initialize global auxT to 0
initialize global auxH to 0

to DrawLineSegment_TH LineColor CurrentDegrees PrevDegrees Offset
do
  set GraphTyH . LineWidth to 3
  call GraphTyH . DrawPoint
  x get global XIndex
  y call AdjustYTemp
  value get PrevDegrees
  Offset get Offset
  set GraphTyH . LineWidth to 2
  set GraphTyH . PaintColor to get LineColor
  call GraphTyH . DrawLine
  x1 get global XIndex
  y1 call AdjustYTemp
  value get PrevDegrees
  Offset get Offset
  x2 get global XIndex + 9
  y2 call AdjustYTemp
  value get CurrentDegrees
  Offset get Offset

to AdjustYTemp value Offset
result round (get Offset - (49 / 15) * get value)
  
```

```

when actualizar_TH Click
do
  call BluetoothClient1 . SendText
  text "B"
  set global auxH to call BluetoothClient1 . ReceiveUnsigned1ByteNumber
  set global auxT to call BluetoothClient1 . ReceiveUnsigned1ByteNumber
  if (get global auxH == 0) and (get global auxT == 0)
  then
    call Aviso . ShowAlert
    notice "Por favor espera 1 minuto para volver a actualizar"
  else
    set valor_H . Text to get global auxH
    set valor_T . Text to get global auxT
  end if
end do
  
```

```

when aviso_guardarTH . AfterChoosing
choice
do
  set global TiempoFecha to call DelayClockEnvio . Now
  if (get choice == "SI")
  then
    call Aviso . ShowAlert
    notice join "Guardado en:"
    call GraphTyH . SaveAs
    fileName join "/Graphics/"
    call DelayClockEnvio . Hour
    instant get global TiempoFecha
    call DelayClockEnvio . Minute
    instant get global TiempoFecha
    call DelayClockEnvio . Second
    instant get global TiempoFecha
    " "
    call DelayClockEnvio . DayOfMonth
    instant get global TiempoFecha
    call DelayClockEnvio . Month
    instant get global TiempoFecha
    call DelayClockEnvio . Year
    instant get global TiempoFecha
    "graphTyH"
    ".jpg"
  end if
else
end do
end choice
  
```

```

when TyH Click
do
  set ventana_2 . Visible to false
  call BluetoothClient1 . SendText
  text "T"
  set ventana_TyH . Visible to true
  while test (get global aux <= 30)
  do
    initialize local Humedad to call BluetoothClient1 . ReceiveUnsigned1ByteNumber
    in call DrawLineSegment_TH
    LineColor
    CurrentDegrees get Humedad
    PrevDegrees get global PrevHumedad
    Offset 197
    set global PrevHumedad to get Humedad
    set global auxH to get Humedad
  end do
  initialize local Temperatura to call BluetoothClient1 . ReceiveSigned1ByteNumber
  in call DrawLineSegment_TH
  LineColor
  CurrentDegrees get Temperatura
  PrevDegrees get global PrevTemperatura
  Offset 197
  set global PrevTemperatura to get Temperatura
  set global auxT to get Temperatura
  set global XIndex to (get global XIndex + 9)
  set global aux to (get global aux + 1)
end while
set valor_H . Text to get global auxH
set valor_T . Text to get global auxT
set global aux to 0
end do
  
```

```

when retornar_a_v2_desde_tyh Click
do
  call GraphTyH . Clear
  set global XIndex to 25
  call BluetoothClient1 . SendText
  text "A"
  set ventana_TyH . Visible to false
  set ventana_2 . Visible to true
end do
  
```

```

when guardarTH Click
do
  call aviso_guardarTH . ShowChooseDialog
  message "¿Desea guardar esta gráfica?"
  title " "
  button1Text "SI"
  button2Text "No"
  cancelable false
end do
  
```

Referencias

- [1] H. G. Liddell, R. Scott, “πρόσθεσις”, *A Greek–English Lexicon at the Perseus Project*. [En línea]. Disponible en:
<http://www.perseus.tufts.edu/hopper/text?doc=Perseus:text:1999.04.0057:entry=pro/sqesis>
[Consultado: 13 de marzo de 2014].
- [2] L. Farah Simon, H. L. García Guerra, R. Rodríguez López, “Diseño de un socket ajustable para prótesis de miembro inferior”, Tesis Profesional, Facultad de Ingeniería, UNAM, Ciudad de México, México, 2006.
- [3] S. Lam, “Types of Prosthesis – Myoelectric Prosthesis”, The University of California, 2010. [En línea]. Disponible en:
<http://bme240.eng.uci.edu/students/10s/slam5/types.html>
[Consultado: 13 de marzo de 2014].
- [4] “Prostheses Prosthetics and Artificial Limbs”, *Disabled World*. [En línea]. Disponible en: <http://www.disabled-world.com/assistivedevices/prostheses/>
[Consultado: 16 de marzo de 2014].
- [5] S. G. Karlsson, “User Control of Lower Limb Prosthesis”, Tesis Profesional, Faculty of Industrial, Mechanical Engineering and Computer Science, University of Iceland, 2010. [En línea]. Disponible en:
http://skemman.is/stream/get/1946/4396/12811/2/SGK_MSc_Thesis_fixed.pdf
[Consultado: 16 de marzo de 2014].
- [6] KH Ha, HA Varol, M Goldfarb, “Volitional Control of a Prosthetic Knee Using Surface Electromyography” *Biomedical Engineering, IEEE Transactions on*, 2011. Disponible en:
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5559403>
[Consultado: 16 de marzo de 2014].
- [7] J. Angelico, “Prosthetic Primer: Transfemoral Prosthetics, Above Knee”, Amputee Coalition, 1999. [En línea]. Disponible en:
http://www.amputee-coalition.org/inmotion/may_jun_99/transfemoral.html
[Consultado: 17 de marzo de 2014].
- [8] J. F. Ramírez, E. J. Muñoz, J. A. Vélez, “Algoritmo para la predicción de fuerzas reactivas en sockets de amputados transfemorales”, 2012, vol.79, n.173. [En línea]. Disponible en:
<http://www.revistas.unal.edu.co/index.php/dyna/article/view/30749/39025>
[Consultado: 17 de marzo de 2014].
- [9] R. Borjian Tesis Profesional, University of Waterloo, Waterloo, Ontario, Canada, 2008. [En línea]. Disponible en:
https://uwspace.uwaterloo.ca/bitstream/handle/10012/4082/Roozbeh_Borjian_MASc_thesis.pdf?sequence=1
[Consultado: 20 de marzo de 2014].
- [10] A. L. Muilenburg, A. B. Wilson, “The Definitive Above-Knee Prosthesis – A Manual for Above-Knee Amputees”, *Orthotics & Prosthetics*, 1996. [En línea]. Disponible en:
<http://www.oandp.com/resources/patientinfo/manuals/ak6.htm>

- [Consultado: 20 de marzo de 2014].
- [11] L. Puglisi, H. Moreno, “Prótesis Robóticas”, Universidad Politécnica de Madrid, Departamento de Automática, Ingeniería electrónica e Informática Industrial, Junio de 2010. [En línea]. Disponible en:
http://www.disam.upm.es/~barrientos/Curso_Robots_Servicio/R_servicio/Protesis_files/Protesis%20roboticas.pdf
 [Consultado: 25 de marzo de 2014].
- [12] R. Bowers, “Prosthetic Primer: Transfemoral Prosthetics, Above Knee”, Amputee Coalition, 2002. [En línea]. Disponible en:
http://www.amputee-coalition.org/spanish/inmotion/mar_apr_02/cosmesis.html
 [Consultado: 25 de marzo de 2014].
- [14] L. Guerrero, D. M. De la Torre, F. Ortiz, V. Alejandro, “Prótesis de rodilla policéntrica, un diseño biónico”, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnología Avanzada del IPN, México, 2007. [En línea]. Disponible en:
http://www.imbiomed.com/1/1/articulos.php?method=showDetail&id_articulo=46654&id_seccion=508&id_ejemplar=4727&id_revista=26
 [Consultado: 28 de marzo de 2014].
- [15] B. Dupes, “Prosthetic Knee Systems”, Amputee Coalition, Estados Unidos, 2005. [En línea]. Disponible en:
<http://www.amputee-coalition.org/military-instep/knees.html>
 [Consultado: 28 de marzo de 2014].
- [16] I. Y. Campos, “Diseño de un mecanismo policéntrico para prótesis de rodilla”, Tesis Profesional para obtener el grado de Maestría en Ciencias de la Ingeniería Mecánica, Escuela Superior de Ingeniería Mecánica y Eléctrica, IPN, Ciudad de México, México, 2008. [En línea]. Disponible en:
<http://www.biblio-sepi.esimez.ipn.mx/mecanica/2008/Dise%C3%83%C2%B1o%20de%20un%20mecanismo%20policentrico%20para%20protesis%20de%20rodilla.pdf>
 [Consultado: 28 de marzo de 2014].
- [17] D. Mc Cormack, “Prosthetic Knee Designs: Biomechanics And Functional Classification”, *International Fellow in Pediatric Orthopedic Surgery*, Georgia, Estados Unidos, 1997. [En línea]. Disponible en:
<http://www.iol.ie/~rcsiorth/journal/volume2/february/prostics.htm>
 [Consultado: 28 de marzo de 2014].
- [18] M. Pardo, “La rodilla C-Leg”, Adampi Cyl – Asociación de Amputados. [En línea]. Disponible en:
http://www.adampicyl.es/j15new/index.php?option=com_content&view=article&id=258&Itemid=246
 [Consultado: 30 de marzo de 2014].
- [19] “élan”, Endolite North America. [En línea]. Disponible en:
<http://www.endolite.com/products/elan>
 [Consultado: 30 de marzo de 2014].
- [20] “KX06”, Endolite North America, [En línea]. Disponible en:
<http://www.endolite.com/products/kx06>

- [Consultado: 30 de marzo de 2014].
- [21] “Knees and Legs”, Össur Americas, 2014. [En línea]. Disponible en: <http://www.ossur.com/prosthetic-solutions/products/knees-and-legs> [Consultado: 30 de marzo de 2014].
- [22] Freedom Innovations, “Plié® 2.0 Microprocessor Knee”, 2014. [En línea]. Disponible en: <http://www.freedom-innovations.com/knees/> [Consultado: 30 de marzo de 2014].
- [23] J. A. Hernández Angulo, Diseño de un sistema de autoajuste para un socket protésico a nivel transfemoral. Tesis Profesional, Facultad de Ingeniería, UNAM, Ciudad de México, México 2013.
- [24] Organización Mundial de la Salud, “Clasificación Internacional del Funcionamiento, de la Discapacidad y de la Salud”, 2001. Ministerio de Trabajo y Asuntos Sociales. Secretaría de Estado de Servicios Sociales, Familias y Discapacidad. Instituto de Mayores y Servicios Sociales, España.
- [25] Organización Mundial de la Salud, “Resumen Informe mundial la discapacidad”, 2011. [En línea]. Disponible en: http://www.who.int/disabilities/world_report/2011/summary_es.pdf?ua=1 [Consultado: 7 de agosto de 2015].
- [26] INEGI “Estadísticas a propósito del día internacional de las personas con discapacidad”, 2012. [En línea]. Disponible en: http://www.conadis.gob.mx/doc/contenidos/inegi_2012.pdf [Consultado 7 de agosto de 2015].
- [27] Centro Nacional de Excelencia Tecnológica en Salud “Rehabilitación del Paciente Adulto Amputado de Extremidad Inferior por Diabetes Mellitus, en el Segundo y Tercer Nivel de Atención”, 2010. [En línea]. Disponible en: http://www.cenetec.salud.gob.mx/descargas/gpc/CatalogoMaestro/257_DIF_09_Amputado_DM/EyR_DIF_257_09.pdf [Consultado: 8 de agosto de 2015].
- [28] Salud 180 “Amputación de extremidades inferiores”. [En línea]. Disponible en: <http://www.salud180.com/salud-z/amputacion-de-extremidades-inferiores> [Consultado: 8 de agosto de 2015].
- [29] MedlinePlus, “Amputación traumática”, 9 de agosto de 2014. [En línea]. Disponible en: <https://www.nlm.nih.gov/medlineplus/spanish/ency/article/000006.htm> [Consultado: 8 de agosto de 2015].
- [30] J. L. Gómez, “Las Prótesis: Restauración del Individuo - El pionero de las prótesis en nuestro país”, Ciencia y Desarrollo – CONACYT, Junio de 2006. [En línea]. Disponible en: <http://www.cyd.conacyt.gob.mx/196/Articulos/Lasprotesis/Lasprotesis03.htm> [Consultado: 9 de agosto de 2015].
- [31] P. Pérez Morales, “Tecnología CAD-CAM”, Sistemas Ortopédicos del Noroeste. [En línea]. Disponible en: http://sistemasortopedicosdelnoroeste.es.tl/CAD_CAM.htm [Consultado: 9 de agosto de 2015].

- [32] M. Arnette, "Responses to Genium/Cleg comparison", OANDP-L, 19 de febrero de 2013. [En línea]. Disponible en: <http://www.oandp.com/oandp-l/message.asp?frmMessageId=2AE2BD61-F063-4775-ACD3-1ED7EF80EFDB>
[Consultado: 9 de agosto de 2015].
- [33] "Genium controlada por microprocesador", Ottobock 2013. [En línea]. Disponible en: <http://www.ottobock.com.mx/prosthetics/lower-limb/solution-overview/genium-bionic-prosthetic-system/>
[Consultado: 9 de agosto de 2015].
- [34] D. García Jurado, "Líner – Hablemos de Prótesis", 7 de marzo de 2013. [En línea]. Disponible en: <http://danigaro.com/2013/03/07/liner/>
[Consultado: 11 de agosto de 2015].
- [35] F. Rojas, "Sockets", Ortho-Mex Sistemas Protésicos y Ortésicos, 2010. [En línea]. Disponible en: <http://www.ortho-mex.com.mx/protesis-de-pierna/sockets.html>
[Consultado: 11 de agosto de 2015].
- [36] A. Cavaco, L. Durães, S. Pais, A. García, "Liners en las prótesis: revisión de la literatura", ORTO 2012 Congress, 28 de septiembre de 2012. [En línea]. Disponible en: http://www.ortomedicalcare.com/pdf/2012/ponencias/P_Adriana_Cavaco.pdf
[Consultado: 11 de agosto de 2015].
- [37] W. Root, "Exo Prosthetic Leg", 22 de octubre de 2014. [En línea]. Disponible en: <https://www.behance.net/gallery/20696469/Exo-Prosthetic-Leg>
[Consultado: 12 de agosto de 2015].
- [38] G. Pirouzi, N. A. Abu Osman, A. A. Oshkour, S. Ali, H. Gholizadeh, W.A.B Wan Abas, "Development of an Air Pneumatic Suspension System for Transtibial Prostheses" Sensors 2014, 14 p., 16754-16765 pp.
- [39] G. M. Street, L. Mosler, R. E. Finlinson, W. M. Clover Jr., E. Laatsch, "Pneumatic connections for prosthetic socket" EUA, Patente US8496715 B2, 30 de julio de 2013.
- [40] L. Porten, "Pneumatic cushion socket with a porous filler" EUA, Patente US3309714 A, 21 de marzo de 1967.
- [41] J. B. Price Jr., "Adjustable prosthetic socket" EUA, Patente US6991657 B1, 31 de enero de 2006.
- [42] W. S. Patterson, W. E. Patterson, "Prosthetic socket with self-contained vacuum reservoir" EUA, Patente US 7427297 B2, 23 de septiembre de 2008.
- [43] R. C. Dean Jr., M. B. Mayor, D. F. Nelson, C. S. Braley, M. W. Blanchard, "Dynamic variable geometry fitting system for use with a body appliance" EUA, Patente US 6585774 B2, 1 de julio de 2003.
- [44] H. L. García Guerra, "Diseño de un socket autoajustable para prótesis de miembro inferior", Tesis de Maestría, Facultad de Ingeniería, UNAM, Ciudad de México, México, 2009.
- [45] M. Almanza Mazas, "Diseño e implementación de una aplicación móvil para monitoreo de un socket autoajustable", Tesis Profesional, Facultad de Ingeniería, UNAM, Ciudad de México, México, 2014.

- [46] G. Sáenz Ortiz, “Metodología de pruebas estáticas y dinámicas para un socket protésico de miembro inferior a nivel transfemoral”, Tesis Profesional, Facultad de Ingeniería, UNAM, Ciudad de México, México, 2014.
- [47] “7 de cada 10 smartphones en México son Android”, CNNExpansión.com, 27 de Octubre de 2014. [En línea]. Disponible en:
<http://www.cnnexpansion.com/tecnologia/2014/10/27/siete-de-cada-10-mexicanos-usan-android>
 [Consultado: 30 de septiembre de 2015].
- [48] R. Alamilla, R. Camargo, “Radiografía del Mercado de Smartphones 2T15”, The Competitive Intelligence Unit S.C. [En línea]. Disponible en:
http://www.the-ciu.net/nwsltr/419_2Distro.html
 [Consultado: 1 de octubre de 2015].
- [49] Y. Çengel and M. Boles, “Energía, transferencia de energía y análisis general de energía” en *Termodinámica*, México: McGraw-Hill, 2009, 21-30 pp.
- [50] Y. Çengel and M. Boles, “Propiedades de las sustancias puras” en *Termodinámica*, México: McGraw-Hill, 2009, 21-30 pp.
- [51] Y. Çengel and M. Boles, “Mezclas de gas-vapor y acondicionamiento de aire” en *Termodinámica*, México: McGraw-Hill, 2009, 733-735 pp.
- [52] D. Ibrahim, “Microcontroller based applied digital control” Chichester: Wiley, 2006.
- [53] L. J. Rodríguez Aragón, “Tema 4: Internet y Teleinformática”, *Informática Básica*. Departamento de Informática, Estadística y Telemática. Universidad Rey Juan Carlos. [En línea]. Disponible en:
<http://www.uclm.es/profesorado/licesio/Docencia/IB/IBTema4.pdf>
 [Consultado: 3 de octubre de 2015].
- [54] “Bluetooth Fast Facts”, Bluetooth Technology Website. [En línea]. Disponible en:
<http://www.bluetooth.com/Pages/Fast-Facts.aspx>
 [Consultado: 3 de octubre de 2015].
- [55] “Technology Basics”, Bluetooth Technology Website. [En línea]. Disponible en:
<http://www.bluetooth.com/Pages/Basics.aspx>
 [Consultado: 3 de octubre de 2015].
- [56] “Bluetooth Protocol”, Nokia Developer Website. [En línea]. Disponible en:
http://developer.nokia.com/community/wiki/Bluetooth_Protocol
 [Consultado: 3 de enero de 2015].
- [57] R. Gutierrez-Osuna, “Sensor characteristics – Intelligent Sensor Systems”, Clase. Wright State University, Estados Unidos. [En línea]. Disponible en:
http://research.cs.tamu.edu/prism/lectures/iss/iss_l2.pdf
 [Consultado: 5 de octubre de 2015].
- [58] K. Kalantar-zadeh, “Sensor Characteristics and Physical Effects” en *Nanotechnology-enabled sensors*, New York: Springer, 2007, 492 p., 13-17 pp.
- [59] Aosong (Guangzhou) Electronics Co., Ltd “Temperature and humidity module DHT11 Product Manual”, datasheet. [En línea]. Disponible en:
<http://akizukidenshi.com/download/ds/aosong/DHT11.pdf>
 [Consultado: 5 de octubre de 2015].

- [60] I. Kosow and V. González Pozo, “Máquinas eléctricas y transformadores”, México: Prentice Hall Hispanoamericana, 1993.
- [61] “Servos – Mecánica”, X-Robotics – Robótica & μ Controladores Pic. [En línea]. Disponible en: <http://www.x-robotics.com/motorizacion.htm> [Consultado: 6 de octubre de 2015].
- [62] “Pilas y baterías”, MCI Ltda. - Olimex Chile. [En línea]. Disponible en: <http://www.olimex.cl/tutorials.php?page=tutorial-baterias> [Consultado: 6 de octubre de 2015]
- [63] I. Marchant, “Futuro Digital México 2015”, comScore Webinar, 1 de junio de 2015. [En línea]. Disponible en: <https://www.comscore.com/lat/Insights/Presentations-and-Whitepapers/2015/2015-Mexico-Digital-Future-in-Focus> [Consultado: 8 de octubre de 2015].
- [64] “One-Quarter of Mexico's Population to Use Smartphones in 2014”, eMarketer, 22 de diciembre de 2014. [En línea]. Disponible en: <http://www.emarketer.com/Article/One-Quarter-of-Mexicos-Population-Use-Smartphones-2014/1011753> [Consultado: 8 de octubre de 2015].
- [65] “Tools Help”, Android Developers. [En línea]. Disponible en: <http://developer.android.com/intl/es/tools/help/index.html> [Consultado: 9 de octubre de 2015].
- [66] Z. Honig, “Google intros Android Studio, an IDE for building apps”, Engadget – Technology News, Advice and Features, 15 de mayo de 2013. [En línea]. Disponible en: <http://www.engadget.com/2013/05/15/google-android-studio/> [Consultado: 9 de octubre de 2015].
- [67] T. Claburn, “Google App Inventor Simplifies Android Programming”, InformationWeek, 12 de julio de 2010. [En línea]. Disponible en: <http://www.informationweek.com/mobile/google-app-inventor-simplifies-android-programming/d/d-id/1090674?> [Consultado: 11 de octubre de 2015].
- [68] “ADT Plugin Release Notes”, Android Developers. [En línea]. Disponible en: <http://developer.android.com/intl/es/tools/sdk/eclipse-adt.html> [Consultado: 20 de octubre de 2015].
- [69] “Application Fundamentals”, Android Developers. [En línea]. Disponible en: <http://developer.android.com/intl/es/guide/components/fundamentals.html> [Consultado: 20 de octubre de 2015].
- [70] G. Pacheco H., A. Rojas T., A. Hernández Q., “Gasto másico y potencia de una bomba”, Laboratorio de Termodinámica, División de Ciencias Básicas, Facultad de Ingeniería, UNAM. [En línea]. Disponible en: <http://dcb.fi-c.unam.mx/CoordinacionesAcademicas/FisicaQuimica/Termodinamica/gastompbo mba.pdf> [Consultado: 20 de octubre de 2015].
- [71] K. Ogata, “Modern control engineering”, Boston, MA: Prentice-Hall, 2010. 894 p.

- [72] K. Ogata, "Dinámica de sistemas" Prentice-Hall Hispanoamericana, 1987. 619 p.
- [73] "Standards Internacionales – Artículos Técnicos e Informes de Servicio", Technician's Friend. [En línea]. Disponible en: <http://www.techniciansfriend.com/standards.htm> [Consultado: 6 de noviembre de 2015].
- [74] "SparkFun Bluetooth Mate Silver", SparkFun Electronics®. [En línea]. Disponible en: <https://www.sparkfun.com/products/12576> [Consultado: 6 de noviembre de 2015].
- [75] B. Valdez, "El periodismo necesita inversión", Grupo Milenio: Política. 8 de enero de 2015. [En línea]. Disponible en: http://www.milenio.com/politica/mil-amputaciones-pie-diabetico-Mexico_0_223177697.html [Consultado: 6 de noviembre de 2015].
- [76] "Settlement and cross-license agreement between Otto Bock and Össur", Össur Corporate Website. [En línea]. Disponible en: <http://www.ossur.com/corporate/about-ossur/ossur-news/1231-settlement-and-cross-license-agreement-between-otto-bock-and-ossur> [Consultado: 6 de noviembre de 2015].
- [77] K. J. Åström, "Simple Control Systems". Notas de Lectura. Departamento de Ingeniería Mecánica y Ambiental, Universidad de California. 2002. <http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom-ch4.pdf> [Consultado: 26 de febrero de 2016].