



**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

---

---

**FACULTAD DE ESTUDIOS SUPERIORES  
ARAGÓN**

**IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN  
ARTIFICIAL MEDIANTE PROCESADORES ARM  
PARA LA MANIPULACIÓN DE UNA SILLA DE  
RUEDAS.**

**TESIS**

**QUE PARA OBTENER EL TÍTULO DE  
INGENIERO ELÉCTRICO ELECTRÓNICO**

**P R E S E N T A:**

**DIEGO REYES QUIJADA**

**ASESOR:**

**MTRO. EN I. MAURICIO ONTIVEROS SALGADO**



**CIUDAD NEZAHUALCÓYOTL, ESTADO DE MÉXICO, 2018**



Universidad Nacional  
Autónoma de México



**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 mis padres Nicolás y Eugenia, quienes siempre me han dado todo lo que está en sus manos, por su cariño y enseñanzas. Este trabajo es el resultado de su amor. Siempre los llevo en mis pensamientos.*

*A mis hermanos Nicolás y Sergio, con quienes disfruto cada momento de la vida. A mi hermano Hugo, por ser la inspiración de este trabajo y ser la persona que nos mantiene más unidos como familia a pesar de las adversidades.*

*A mi esposa Valeria, por ir de la mano conmigo siempre y darme los momentos más hermosos desde que la conocí, uno de ellos darme la dicha de ser padre de Natalia y Diego. Te amo.*

*A la familia Nava Quijada, quienes nos han apoyado en momentos difíciles. Estoy en deuda con ellos.*

*A la Universidad Nacional Autónoma de México y Facultad de Estudios Superiores Aragón, la casa de estudios que me ha dado la oportunidad de ser alguien en la vida basado en la formación académica que me ha brindado, por el apoyo y las facilidades como estudiante. Estoy orgulloso de pertenecer a esta institución.*

*A mi asesor de tesis, M. en I. Mauricio Ontiveros Salgado por dedicarme el tiempo para sacar adelante este proyecto y por sus consejos. También agradezco especialmente a los revisores, M. en C. Arturo Ocampo Álvarez, Ing. Isaac I. López Trujano, Ing. Ramón Patiño Rodríguez e Ing. Oscar Guadalupe Moreno Espinoza por sus observaciones acertadas en este trabajo. Gracias por la dedicación de su tiempo en la revisión de este trabajo.*

# Contenido

Resumen.....	i
Capítulo 1 Introducción.....	1
1.1 Planteamiento del problema .....	2
1.2 Justificación.....	2
1.3 Propuesta de solución .....	3
1.4 Objetivo .....	3
1.5 Metodología.....	3
Capítulo 2 Marco teórico.....	5
2.1 Parálisis cerebral .....	5
2.1.1 Etiología de la parálisis cerebral.....	7
2.1.2 Clasificación de la parálisis cerebral .....	8
2.1.3 Incidencia general de parálisis cerebral en México.....	10
2.2 Silla de ruedas .....	14
2.2.1 Tipos de silla de ruedas.....	15
2.2.2 Direccionamiento diferencial .....	17
2.3 Visión artificial.....	17
2.3.1 Elementos de un sistema de visión artificial .....	18
2.3.2 Etapas de un sistema de visión artificial .....	28
2.3.3 Niveles de visión artificial .....	31
2.4 <i>Advanced RISC Machine (ARM®)</i> .....	31
2.4.1 Características de la arquitectura <i>ARM®</i> .....	31
2.4.2 Set de instrucciones de <i>ARM®</i> .....	32
2.4.3 Procesadores <i>ARM Cortex®</i> .....	33
2.5 Transceptores de radiofrecuencia.....	36
2.5.1 Módulo <i>NRF24L01®</i> .....	37
Capítulo 3 Consideraciones del sistema de control .....	38
3.1 Movilidad de la silla de ruedas.....	38
3.2 Consideraciones del sistema de visión artificial para el direccionamiento .....	40

3.3 Consideraciones del sistema manual para el direccionamiento.....	42
3.4 Control de motores para el direccionamiento.....	43
Capítulo 4 Implementación de software y diseño de hardware .....	45
4.1 Funcionamiento del sistema .....	45
4.2 Algoritmo del sistema de visión artificial.....	46
4.2.1 Adquisición.....	47
4.2.2 Pre proceso .....	48
4.2.3 Segmentación.....	49
4.3 Algoritmo del control manual .....	51
4.4 Algoritmo del sistema de control de motores .....	55
4.4.1 Selección de los modos de operación.....	63
4.5 Diseño de hardware del control manual .....	64
4.6 Diseño de hardware del control de motores .....	66
4.6.1 Criterio de selección de los elementos de potencia .....	66
4.7 Análisis de costos .....	74
Capítulo 5 Pruebas del sistema y resultados.....	77
5.1 Sistema de visión artificial .....	77
5.1.1 Adquisición.....	78
5.1.2 Pre proceso .....	78
5.1.3 Segmentación.....	79
5.1.4 Transmisión de datos.....	92
5.2 Prueba del sistema manual.....	94
5.2.1 Prueba de la inclinación en el eje x,y.....	94
5.2.2 Transmisión de información.....	97
5.3 Prueba del control de motores .....	99
5.3.1 Direccionamiento de avance.....	101
5.3.2 Direccionamiento de reversa .....	103
5.3.3 Direccionamiento de giro.....	104
5.4 Circuito impreso del control manual.....	106
5.5 Circuito impreso del control de motores.....	107
Capítulo 6 Conclusiones .....	110
6.1 Trabajo a futuro .....	111

Referencias bibliográficas .....	112
----------------------------------	-----

# Resumen

En este trabajo se describe el proceso de desarrollo e implementación de un sistema de visión artificial para el direccionamiento de una silla de ruedas, dirigido a personas con parálisis cerebral totalmente dependientes. El trabajo se desarrolla principalmente en una microcomputadora de placa reducida, en inglés, *Single Board Computer (SBC)*[25], la cual tiene integrado un procesador *Broadcom*®. También se utiliza una tarjeta de desarrollo *FRDM-KL25Z*® la cual tiene incorporado un micro controlador *Kinetis L Series*®. Ambos dispositivos de arquitectura *Advanced RISC Machine*®, por sus siglas en inglés *ARM*® [25].

Con una cámara digital de interfaz *USB*® [25], en inglés, *Universal Serial Bus*, se obtiene una serie de imágenes digitales en las cuales se realiza el procesamiento, reconocimiento y seguimiento de un objetivo particular. El reconocimiento del objetivo se realiza en la microcomputadora bajo el entorno *Linux*® [25] y su distribución *Raspbian*® utilizando la biblioteca *OpenCV*®. Con los recursos mencionados se realiza el reconocimiento y se obtiene la coordenada "x" y el área del objetivo en la imagen para realizar el seguimiento.

Se realiza un escalamiento del dato obtenido en el rango de -10 a 10, teniendo como 0 el centro de la imagen. Estos 20 valores se envían a la tarjeta de desarrollo para generar las señales de modulación por ancho de pulso, en inglés *Pulse Width Modulation (PWM)* [25] y poder controlar dirección del giro y velocidad de los motores eléctricos de corriente directa mediante un puente H.

Con dos transceptores de radiofrecuencia se desarrolla un control manual controlado por un segundo usuario, el cual envía las señales provenientes del acelerómetro incorporado en una segunda tarjeta de desarrollo *FRDM-KL25Z*®. Con los datos generados por el sistema de visión o el control manual, se logra el control de los motores y direccionamiento de la silla de ruedas.

# Capítulo 1

## Introducción

La ingeniería ocupa un campo muy extenso en sus aplicaciones, desde el diseño y fabricación de máquinas y construcciones monumentales hasta el desarrollo de un chip nano métrico que puede ser implantado en el cerebro. La mayor parte de estas aplicaciones están dirigidas a facilitar las actividades en beneficio de la población, es por esto que ingenieros de todas partes del mundo aplican sus conocimientos en campos particulares para mejorar la calidad de vida de la población.

La visión por computadora o visión artificial es una disciplina mediante la cual se otorga el sentido de la vista a una máquina, se analiza y extrae información de objetos específicos en la escena capturada y con ello se ejecutan acciones o tareas específicas, tal y como lo hacemos los seres biológicos.

El campo de aplicación de la visión artificial es muy extenso, se aplica en procesos industriales como el control de calidad, actividades en el campo de la medicina como análisis de imágenes de Rayos X, aplicaciones militares como detección y seguimiento de objetivos, análisis de plantaciones en la agricultura, vigilancia de edificios en aplicaciones de seguridad o aplicaciones de robótica muy complejos como el cálculo de una ruta en un entorno desconocido.

En la actualidad existen diferentes plataformas de desarrollo y una amplia variedad de aplicaciones de reconocimiento y seguimiento de objetos. La tendencia en los avances tecnológicos es diseñar y construir sistemas eficientes, de bajo costo y dimensiones. Estas características son comunes en varias tarjetas de desarrollo y sistemas embebidos como las micro computadoras de placa reducida o *Single Board Computer* [25], que son computadoras construidas en una mínima expresión ya que sus dimensiones físicas son comparables a las de una tarjeta de crédito y su composición electrónica es la misma que la de una computadora normal, poseen memoria de acceso aleatorio, en inglés, *Random Access Memory (RAM)* [25], un microprocesador capaz de trabajar bajo un sistema operativo, así como periféricos de entrada y salida para la conexión de teclado, mouse, pantalla etc.

Se implementará un sistema de visión artificial y un sistema de control manual a una silla de ruedas mecánica convencional para que funcione con energía eléctrica proveniente de baterías selladas de ácido plomo, en inglés *Valve Regulated Lead Acid (VRLA)*. Para el procesamiento de imágenes la micro computadora utilizada en este proyecto es la tarjeta *Raspberry Pi 2 Model B*® por sus funcionalidades, ya que cuenta con un procesador Broadcom® BCM2836 quad-core ARM Cortex-A7 de 900 MHz hasta 1 GHz, 1GB en RAM, bastantes pines con funcionalidades GPIO y de propósito específico [35].



El controlador de los motores se realiza a mediante la tarjeta de desarrollo *FRDM-KL25Z*® desarrollada por NXP®, la cual tiene entre sus principales componentes un microcontrolador MKL25Z® de 32 bits, un acelerómetro digital de 3 ejes, un diodo emisor de luz (*LED*) indicador, un *slider* capacitivo así como 7 puertos de comunicación y una gran cantidad de pines GPIO [39] con los que podremos manipular la dirección y velocidad de los motores con señales de modulación de ancho de pulso (*PWM*) y dos puente H de potencia.

## 1.1 Planteamiento del problema

En base a la experiencia personal adquirida en las visitas a la Asociación Pro Personas Con Parálisis Cerebral (APAC), a la experiencia en el hogar, y al análisis de operación de las sillas de ruedas comerciales controladas por *joystick*, se detecta que las personas con esta capacidad especial no serían aptas para la manipulación de este tipo de sillas de ruedas, ya que los movimientos musculares de este tipo de usuario no permitirían ejecutar el control adecuado.

En APAC se observó también que un porcentaje importante de los pacientes es de una edad promedio de treinta años, y que deben ser acompañados por algún familiar quien en muchos casos son adultos mayores los que realizan el esfuerzo físico para trasladar al usuario, subiendo rampas con inclinaciones de 20° aproximadamente y de gran longitud.

La principal contribución de este trabajo sería darle movilidad a una persona con parálisis cerebral suprimiendo la acción de empuje propia del usuario o de un tercero, lo cual conlleva cierta comodidad para el traslado. Además, los gastos en la elaboración no superarían a los precios comerciales de las sillas de ruedas eléctricas, que en general controlan su direccionamiento mediante un *joystick*.

## 1.2 Justificación

La tendencia en la fabricación de circuitos integrados e implementación de sistemas electrónicos, tarjetas de desarrollo y sistemas embebidos es mejorar la funcionalidad, reducir dimensiones y costos. En el mercado de las sillas de ruedas eléctricas controladas por *joystick*, el costo comercial se encuentra a partir de los \$26,599 pesos [41] hasta los \$69,900 pesos [42], implicando un gran gasto para las familias de bajos recursos, con la limitante de que personas totalmente dependientes no pueden operar este tipo de sillas.

El objetivo primordial de este trabajo es darle movilidad una silla de ruedas mecánica convencional de una persona con parálisis cerebral totalmente dependiente a través de un sistema de visión artificial, un sistema de control manual y motores eléctricos reduciendo considerablemente el costo respecto a una silla de ruedas eléctrica comercial controlada por *joystick*, lo que implica ahorro en la inversión y aumento en la comodidad de quien realiza el empuje del usuario de la silla.

### 1.3 Propuesta de solución

La propuesta consiste en desarrollar e integrar un sistema de visión artificial para identificar y seguir un objetivo de características particulares haciendo uso de la computadora de placa reducida *Raspberry Pi 2 Model B*® mediante la biblioteca *OpenCV*® para el procesamiento de imágenes. El control de los motores de la silla se realiza a través una tarjeta de desarrollo *FRDM-KL25Z*®, la cual establece los modos de operación “Cámara” y “Radio” mediante el algoritmo del micro controlador. Los elementos de salida son los dos motores eléctricos de corriente directa que son controlados mediante un puente H completo y modulación por ancho de pulso (*PWM*) para cada motor.

### 1.4 Objetivo

El objetivo principal es desarrollar un sistema de visión artificial para el reconocimiento y seguimiento de un objetivo particular e implementarlo en una silla de ruedas convencional.

Como segundo objetivo se implementa un dispositivo de control de radiofrecuencia el cual envía señales provenientes de un acelerómetro para dar movilidad a la silla, obteniendo así un sistema de control manual.

El tercer objetivo es la implementación de un sistema electrónico de potencia para manipular el funcionamiento de los motores de la silla a través de la información recibida del sistema de visión artificial y del sistema de control manual.

### 1.5 Metodología

En primera instancia se realiza una revisión bibliográfica mediante libros, tesis y artículos científicos de diferentes autores sobre el campo de la visión artificial. Es necesario estudiar el proceso de adquisición de una imagen digital, las variables que componen un escenario, dispositivos de captura, las componentes de una imagen digital y los métodos para realizar de forma adecuada el procesamiento de las imágenes.

Posteriormente se debe elaborar un algoritmo en lenguaje *Python* para realizar el tratamiento de las imágenes capturadas por la cámara *USB*. En este programa se realiza el reconocimiento del objetivo a seguir mediante sus características físicas, las cuales se definirán durante el desarrollo de este proyecto en base a los resultados de las pruebas realizadas.

Una vez detectado el objetivo, se debe localizar su posición espacial en el plano 2D de la imagen en la coordenada “x” y calcular su área, referencias que nos ayudarán a calcular el desplazamiento de la silla de ruedas en el sentido de izquierda o derecha, y de avance o retroceso.

En el modo “Cámara” se envía información de la posición del objetivo de la computadora *Raspberry Pi 2 Model B*® al micro controlador de la tarjeta de desarrollo *FRDM-KL25Z*® quien realiza la interpretación de los datos mencionados, teniendo como salida cuatro señales de modulación por ancho de pulso con las que podremos controlar la dirección y

velocidad de los motores con ayuda del puente H, para generar el direccionamiento de la silla de ruedas.

Se integrará un transceptor de radio frecuencia en la tarjeta de desarrollo que corresponde con la función "Radio" así como diodos emisor de luz, en inglés, *Light Emitting Diode (LED)* indicadores de movimiento para ejecutar un control manual. La tarjeta que realiza el control de los motores también tiene incorporado un módulo de radiofrecuencia, el cual se activa como receptor cuando se selecciona el modo "Radio".

# Capítulo 2

## Marco teórico

### 2.1 Parálisis cerebral

William Little, médico y director del hospital de Londres escribió por primera vez en 1984 sobre la influencia de un parto anormal en un cuadro clínico en el que había alteraciones del movimiento, del tono muscular y babeo en los productos, que con el paso de los años se nombró “enfermedad de Little” para posteriormente llamarse “Parálisis Cerebral” [11].

Este término fue introducido por William Osler en 1889; Freud de igual forma en 1897 lo describe como el resultado de encefalitis y aporta la idea de que se produce frecuentemente por factores congénitos y complicaciones perinatales. A lo largo del siglo XX y sobre todo de la segunda guerra mundial, se incrementó el estudio de la parálisis cerebral. Numerosos estudios aportaron los conocimientos que actualmente se tienen en el campo de la medicina. En general estos trabajos se concentraron en el estudio de problemas neurológicos, principalmente en la patología, la etiología y la clasificación de los mismos [11].

En estos tiempos “Parálisis Cerebral” se interpreta como un conglomerado de deficiencias emocionales, neuromusculares, y sensoriales causado por el daño o la ausencia de estructuras cerebrales. Estudiosos de esta condición, la definieron en su momento como *“trastorno persistente pero no invariable de la postura y del movimiento debido a una lesión no evolutiva del encéfalo antes de que su crecimiento y desarrollo se completen”*. Esta definición es comúnmente aceptada en los Congresos sobre Parálisis Cerebral celebrados en Oxford en 1958 y Berlín en 1966. Otras definiciones comparten la misma idea pero destacando, según sea el caso, de una lesión en un cerebro inmaduro y que tiene lugar antes o después del parto (hasta los tres o cinco años, según algunos autores) y sobre todo que se trata de un daño no progresivo [11].

Los autores franceses en la materia lo mencionan como una “enfermedad motriz cerebral” o de una “incapacidad motriz cerebral” restringiendo el término para personas con niveles mentales normales. Otros autores señalan que *“el término IMOC (incapacidad motriz cerebral) permite englobar todas las lesiones cerebrales que ocurren antes de los seis años, casi únicamente en el periodo perinatal, pero en las cuales el elemento esencial es el elemento motor, sean cuales fueren las invalideces asociadas a dichas lesiones intelectuales, sensoriales, epilepsia etc. Cuando en el cuadro clínico predomina el elemento intelectual y el elemento motor es nulo o secundario, debe utilizarse el término ‘encefalopatía’”*. La característica que señalan todos los autores tanto ingleses o francos, es que se trata de un déficit motor [11].

El Dr. Eric Denhoff propone hablar de Parálisis Cerebral en el contexto de los síndromes de disfunción cerebral, que contemplan dificultades de desarrollo, como: retardo mental orgánico, epilepsia, determinados problemas de conducta y dificultades de aprendizaje, las cuales podrían encontrarse por separado o en conjunto. Esta delimitación está influenciada por la época, pues en los años sesenta se prefería hablar de “dificultades de aprendizaje” para designar anteriores [11].

A partir de los años setenta, se comenzó a utilizar el término “*discapacidad en el desarrollo*”, que según la federación norteamericana incluía: retardo mental, epilepsia, parálisis cerebral, afasia y similares problemas clínicos. Cabe señalar que la palabra “desarrollo” hace alusión a varias etapas de la infancia, mientras que estos problemas se desarrollan a lo largo de toda la vida; por otra parte, hay un problema de delimitación ya que el término parálisis cerebral nunca se ha llegado a definir del todo sino como una lista de categorías de problemas clínicos asociadas a incapacidades relacionadas con el sistema neurológico. En el fondo lo que resalta es un problema en las clasificaciones las cuales tienen diversas finalidades: para los profesionales en el campo permite dar ideas en la investigación; para los políticos permite clasificar quienes son susceptibles de recibir cierto tipo de ayuda, para los educadores saber qué tipo de educación es la apropiada, etc. [11].

Sin embargo, tales clasificaciones tienen una repercusión social que sirve como detonante de una marginación para todas las personas con esta condición. El uso inadecuado de estas clasificaciones pueden dar lugar a etiquetas deshumanizantes. Por ende es preciso tener cuidado en el empleo de estos mismos [11].

Por esta situación, la OMS elaboró en el año 1980, la “*Clasificación Internacional de Deficiencias, Discapacidades y Minusvalías. Manual de las consecuencias de la enfermedad*”. En este documento se delimitan conceptos como “deficiencia” orientada al ámbito de la medicina, “discapacidad” a aspectos rehabilitadores y “minusvalía” a aspectos sociales: “*la deficiencia hace referencia a las anomalías de la estructura corporal, de la apariencia y a la función de un órgano o sistema, cualquiera que sea su causa; en principio, las deficiencias presentan trastornos a nivel de órgano...*” “*La discapacidad refleja las consecuencias de la deficiencia desde el punto de vista del rendimiento funcional y de las actividades del individuo, las discapacidades representan, por tanto, trastornos a nivel de la persona*”. “*La minusvalía hace referencia a las desventajas que experimenta el individuo como consecuencia de las deficiencias y discapacidades; así pues, las minusvalías reflejan una interacción y adaptación del individuo al entorno*” [5, 11].

Según estos criterios, una enfermedad provocaría una deficiencia orgánica, y esta provocaría una discapacidad para realizar determinadas acciones y a su vez, esta discapacidad provocaría la minusvalía. Pero entonces ¿Qué es la Parálisis Cerebral? [11].

Los investigadores médicos son los encargados de discutir sobre el tema y determinar si es útil la rehabilitación o no. Desde el ámbito psicoeducativo, hacer un análisis funcional del rendimiento motor y cognitivo de cada niño para realizar una mejor intervención pedagógica [11].

### 2.1.1 Etiología de la parálisis cerebral

Los factores etiológicos han sido objeto de numerosos estudios médicos. El Dr. Eric Denhoff condensó los siguientes factores posibles en donde las causas de la parálisis cerebral se pueden confundir con las que producen una deficiencia mental. Diversos estudios coinciden en señalar que los motivos perinatales son la causa más importante de parálisis cerebral, y por el contrario, se encuentran los factores genéticos [11].

Otros especialistas del tema inciden en la misma idea pero señalando que el bajo peso y sobre todo la prematuridad del niño son los mayores causantes de la parálisis cerebral. A continuación se muestran las principales según la investigación del Dr. Eric Denhoff:

- I. Factores hereditarios:
  - i. Estáticos: Atetosis familiar, Paraplejía familiar, temblor familiar.
  - ii. Progresivos: Enfermedad desmielinizante de origen viral o indeterminado (los desórdenes cromosómicos son raros en parálisis cerebral, así como los metabólicos) [11].
  
- II. Factores congénitos (*adquiridos en el útero*):
  - i. Rubeola, toxoplasmosis, citomegalia, herpes simple y otros agentes infecciosos o virales.
  - ii. Anoxia materna, envenenamiento por monóxido de carbono, estrangulación, anemia, hipotensión asociada con anestesia espinal, infarto placentar, placenta repentina.
  - iii. Hemorragia cerebral prenatal, toxemia materna, trauma directo, diátesis hemorrágicas maternas.
  - iv. Anoxia prenatal, torsión o enredo del cordón.
  - v. Toxinas diversas, drogas [11].
  
- III. Perinatal (durante el parto)
  - i. Anoxia mecánica: Obstrucción respiratoria, sobredosificación con drogas, placenta previa o repentina, hipotensión asociada con anestesia espinal, parto de glúteos con retraso de la cabeza del recién nacido.
  - ii. Traumatismo: Hemorragia asociada con distocia, desproporciones y malposiciones, aplicaciones imprudentes del fórceps, resistencia de la cabeza, parto inducido, cambios súbitos de presión, alumbramiento precipitado, parto por cesárea.
  - iii. Complicaciones del nacimiento: Prematuridad, inmadurez, dismadurez, postmadurez, hiperbilirrubinemia y factores de isoimmunización, trastornos emolíticos, sífilis, meningitis y otras infecciones, reacciones de la drogadicción, reacciones de la hipoglucemia, reacciones de la hipocalcemia [11].
  
- IV. Posnatal y primera infancia
  - i. Traumatismo (hematoma subdural, fractura craneal, contusión craneal).
  - ii. Infecciones (meningitis, encefalitis, abscesos cerebrales).

- iii. Accidentes vasculares (aneurisma cerebral congénito, trombosis, embolia, encefalopatía hipertensa, súbitos cambios de presión).
- iv. Toxinas (plomo, arsénico, alquitrán y derivados...).
- v. Anoxia (envenenamiento por monóxido de carbono, estrangulación, anoxia por bajas presiones y extrema altitud, hipoglucemia).
- vi. Neoplasia y neuro desarrollo tardío (tumores, quiste, hidrocefalia progresiva [11]).

En cualquiera de los casos, hay que señalar que el estudio de la etiología de la parálisis cerebral, por ser de gran importancia desde la perspectiva médica, aporta poco al planeamiento rehabilitador que se lleve a cabo. Para el planteamiento educativo la etiología también tiene poca importancia, ya que, para hacer una adecuada programación de las actividades con el paciente, es de mayor importancia conocer las necesidades que tienen, que las causas que la provocaron [11].

### 2.1.2 Clasificación de la parálisis cerebral

En 1942 se realizó una clasificación que hoy es clásica debido a su claridad lógica. Esta clasificación abarca los siguientes criterios: tipo, topografía, grado, tono y posibles disfunciones asociadas [11].

- I. Clasificación basada en el tipo  
Está basada en el carácter de los movimientos disociados. Según los cuales tendríamos:
  - i. Espasticidad: Se trata de una des armonía en los movimientos musculares producida por hipertonia, es decir, por un aumento exagerado en el tono muscular. El desequilibrio entre excitación de unos músculos e inhibición de otros no se presenta, observándose una exagerada contracción cuando los músculos están extendidos. La espasticidad interfiere con la relación del movimiento voluntario y se caracteriza por espasmos musculares cuando el individuo desea realizar una acción. Por lo general el niño espástico presenta un movimiento lento y explosivo. Se observa la oposición a movimientos rápidos cuando un examinador mueve al niño con rapidez, y por el contrario, la espasticidad disminuye cuando lo mueve lentamente. Esta resistencia es mayor en los músculos que ayudan a mantener la posición erguida.
  - ii. Atetosis: Se caracteriza por extraños movimientos involuntarios sobre todo en las extremidades (muñeca y dedos) aunque, dependiendo de la gravedad, pueden estar también afectados el control de la cabeza y el tronco. Estos movimientos pueden aparecer cuando el niño está relativamente quieto, incrementarse cuando aumenta la tensión emocional y cesar cuando está dormido o en reposo. En algunos casos pueden estar implicados los músculos de la boca, lengua, faringe y laringe, que puede interferir en la deglución, masticación y el habla.

“El atetósico, con sus movimientos involuntarios sobreañadidos, produce un lenguaje extraordinariamente variable, desde pequeños fallos en la articulación hasta la ausencia total del habla”.

- iii. Ataxia: Se trata de una descoordinación de los movimientos voluntarios debida a una alteración del balance, de la postura o de la retroalimentación kinestésica. Se caracteriza por la inestabilidad en la marcha con descoordinación motora tanto fina como gruesa.
- iv. Tipos mixtos: Se pueden dar combinaciones de los tipos anteriores, atetosis combinada con espasticidad, rigidez con ataxia etc. En el campo real, se han encontrado más tipos mixtos que puros [11].

## II. Clasificación basada en la topografía

La clasificación topográfica es usada como complemento en el diagnóstico clínico. Comúnmente se utiliza como un diagnóstico funcional ya que hace alusión a la parte del cuerpo afectada en el movimiento:

- i. Hemiplejía: Afectación a la mitad lateral del cuerpo (izquierda o derecha).
- ii. Diplejía: Las piernas están más afectadas que los brazos.
- iii. Cuadriplejía: Parálisis en los cuatro miembros.
- iv. Paraplejía: Sólo las piernas están afectadas.
- v. Monoplejía: Solamente un miembro está paralizado.
- vi. Triplejía: Tres miembros están afectados [11].

Esta clasificación debe ser ampliada con el concepto de “paresia”, término que debería de ser utilizado en el campo de la parálisis cerebral, en lugar de plejía, ya que designa una “parálisis con restos de movimiento”, y en la parálisis cerebral es muy raro encontrar una parálisis completa [11].

## III. Clasificación basada en el grado

Dependiendo del grado de severidad tanto en movilidad como en comunicación, se puede distinguir:

- i. Leve: La afectación sólo está en la precisión de los movimientos finos. El niño puede realizar una pinza o una señalización con el dedo índice. Su desplazamiento es autónomo aunque puede haber dificultades en el equilibrio o coordinación. El habla es comprensible aunque puede haber ciertos problemas articulatorios.
- ii. Moderado: Los movimientos finos y los gruesos están afectados, puede caminar pero requiere de ayuda parcial (andaderas, bastones, sillas de ruedas etc.). La manipulación de objetos puede realizarse mediante el pulgar y el resto de los dedos, también la señalización con toda la mano. De igual forma, el habla se ve afectada, se le entiende pero tiene graves problemas de comunicación. Las actividades de la vida diaria las pueden realizar con autonomía o con ayuda parcial.
- iii. Severo: No se pueden ejecutar acciones de la vida diaria como el caminar (es necesario que utilice silla de ruedas y no la puede conducir por sí mismo), usar las manos (no tiene pinza ni es capaz de señalar) o hablar (tiene serias dificultades tal que no se le entiende o no habla) [11].



#### IV. Clasificación basada en el tono

También se realizó una clasificación de acuerdo al estado muscular en reposo:

- i. Isotónico: El tono es normal.
- ii. Hipertónico: El tono se encuentra incrementado.
- iii. Hipotónico: Se da una disminución de tono.
- iv. Variable: Tono inconsistente [11].

Esta clasificación tiene mucha relación con la primera, ya que el carácter de los movimientos tiene que ver en su mayor parte, con el tono muscular.

#### V. Disfunciones asociadas

- i. Sensoriales: de visión, de oído, de olfato, táctiles, kinestésicas, propioceptivas.
- ii. Convulsiones epilépticas: Varían según el tipo y grado de la afectación.
- iii. Intelectuales: De tipo y grado.
- iv. Conductuales: Hiperquinesia, ansiedad, neurosis.
- v. Aprendizaje: Dificultades específicas, secundarias a los desórdenes.
- vi. Emocionales: Asociados primariamente con factores familiares o biológicos, secundariamente a los desórdenes [11].

Es un hecho evidente que al tratarse de un daño en el cerebro puede haber otros trastornos fuera de los trastornos motores. Cada caso de parálisis cerebral puede ser diferente, por tanto hay que ser precavidos a la hora de considerar los trastornos anteriores ya que pueden aparecer en conjunto pero no tienen que tener forzosamente la misma causa. Cuando se hace una clasificación de un síndrome en el ámbito médico, es permisible clasificar a los sujetos, pero esa clasificación pierde validez y se reciente más al aplicarlo en el ámbito social desde cualquier perspectiva [11].

### 2.1.3 Incidencia general de parálisis cerebral en México

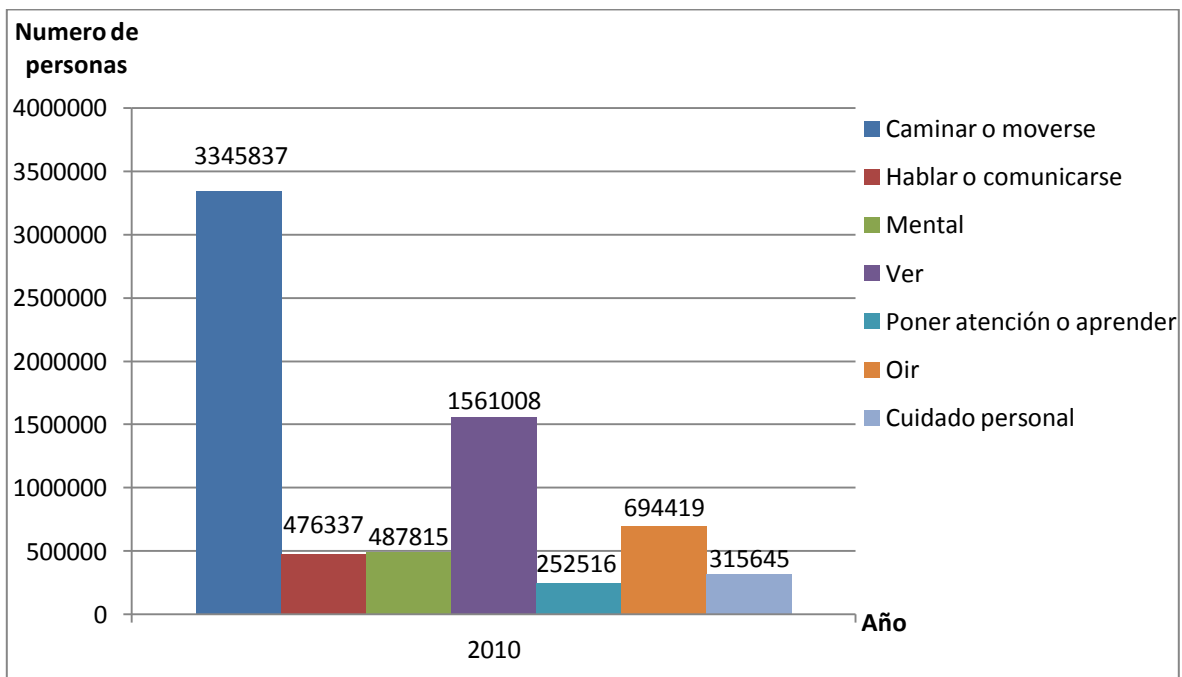
La parálisis cerebral se presenta a nivel mundial, en todos los países y grupos étnicos, siendo la población masculina la mayor afectada. Existen estudios que han demostrado que existen factores como el nivel socioeconómico, y dentro de este, el de los recién nacidos de peso normal donde se registra una mayor incidencia.

Durante los años setenta, en México surgieron distintas asociaciones que prestaron atención a personas con discapacidad, Asociación Pro personas con Parálisis Cerebral (APAC, I.A.P.) en 1970, el Centro de Integración Psicología y Aprendizaje A.C. (CIPAAC) en 1976 y la Confederación Mexicana de Organizadores en Favor de Personas con Discapacidad Intelectual A.C (CONFIE) en 1978. La creación de estas organizaciones se debió al incremento detectado en esta década de niños prematuros con parálisis cerebral [5].

En el año 1998 al 2002, la Secretaría de Salud emitió un estudio en donde se dice que de cada 10,000 niños recién nacidos se reportan 3 casos con parálisis cerebral espástica. En el Sistema Nacional para el Desarrollo Integral de la Familia (SNDIF), se prestó atención a

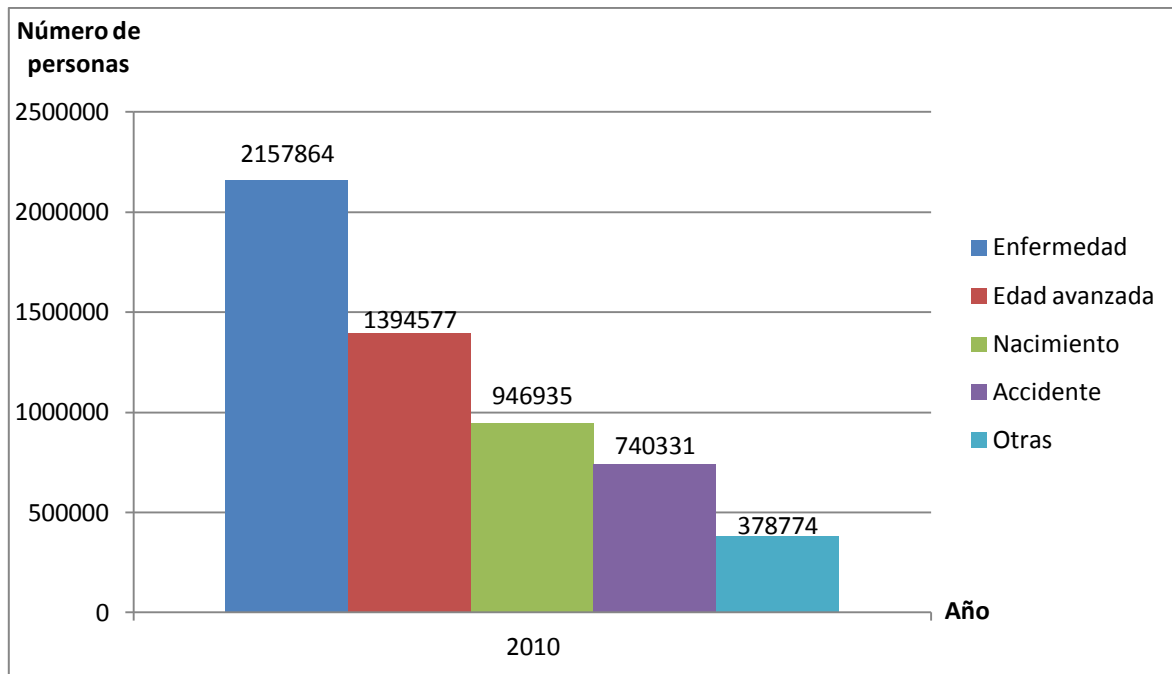
22, 213 niños en el periodo del 2000 al 2008, siendo esta la novena causa de atención a la población [5].

En el año 2010, como resultado del censo de población elaborado por el INEGI (Instituto Nacional de Estadística y Geográfica), se registraron 5 millones 739 mil personas con discapacidad en todo el país. En la figura 2.1 se muestra las principales dificultades de las personas con discapacidad [5].



**Figura 2.1**  
*Principales dificultades de las personas con discapacidad*

La mayor incidencia de las causas que provocaron las discapacidades analizadas, en la mayoría de los casos se reflejan por enfermedad con un 36.7%, edad avanzada con un 24.3% y causas de nacimiento con el 16.5%, es por eso que se debe poner mucha atención en el periodo prenatal y posnatal, para llevar a cabo un diagnóstico y tratamiento correcto. En la figura 2.2 se plasma las principales causas de las personas con discapacidad [5].



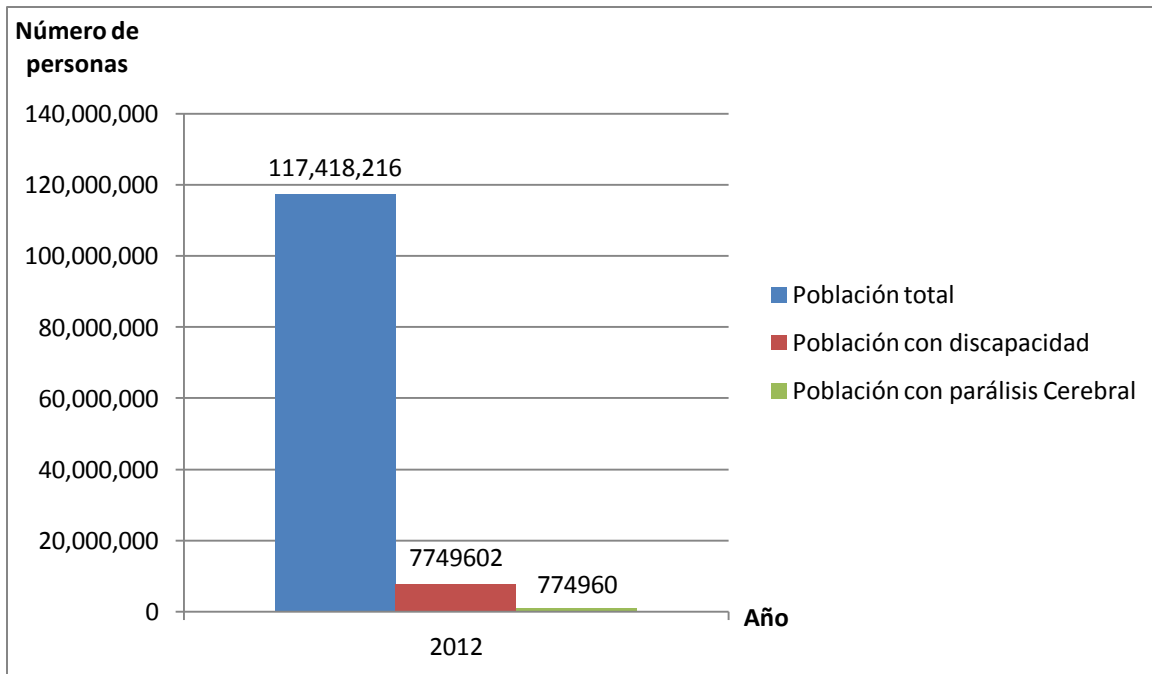
**Figura 2.2**  
**Principales causas de discapacidad**

En el año 2012, la ENIGH (Encuesta Nacional de Ingresos y Gastos de los Hogares), determinó que el 6.6% de la población total del país reporta tener discapacidad (figura 2.3), de este porcentaje, según el CCEM (Centro de Cirugía Especial de México) en el país:

- I. Alrededor del 10% de los discapacitados tienen parálisis cerebral
- II. De lo anterior, más del 60% son del tipo espástico
- III. 40% de los casos se desconoce la causa
- IV. 20% están asociados a problemas de prematurez
- V. El 85% de los casos se origina en el nacimiento

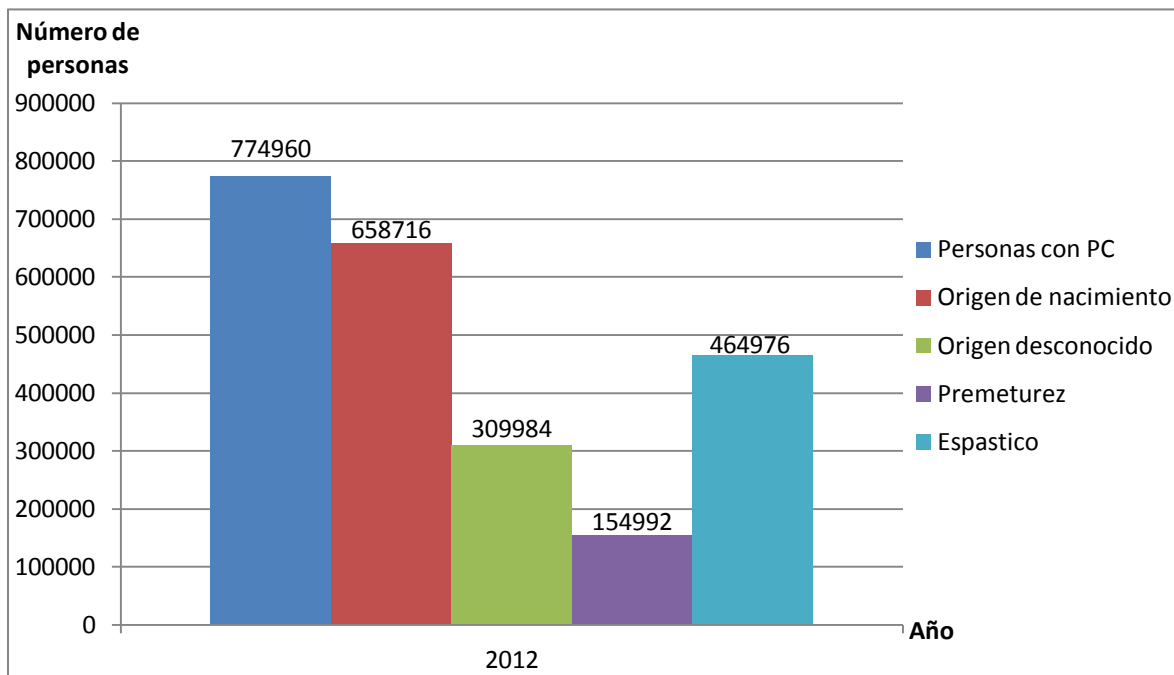
De los datos anteriores se estima que:

- VI. El 25% de los niños con parálisis cerebral nunca podrá caminar
- VII. El 75% que pueden caminar, necesitarán algún tipo de ayuda
- VIII. El 50% de los niños con parálisis cerebral tienen problemas visuales
- IX. Entre el 5% y 15% de los niños con parálisis cerebral tiene problemas auditivos
- X. El 70% de los casos con parálisis cerebral presentan alguna reducción en su capacidad mental [3].



**Figura 2.3**  
**Población con parálisis cerebral en 2012**

De las cifras anteriores, exclusivamente en la población con parálisis cerebral, se derivan otros factores generales como el tipo, etiología, causa, que se muestra en el figura 2.4.



**Figura 2.4** Factores predominantes en personas con parálisis cerebral

## 2.2 Silla de ruedas

La evolución de la parálisis cerebral en los niños ha dado origen a deformidades en su fisionomía debido a la atrofia de algunos músculos. Esto conlleva a un estado de sedentario, en donde es necesario cuidar la postura para evitar lesiones más severas. Por esto, en los casos en que la parálisis cerebral es severa los médicos prescriben una silla de ruedas para garantizar la mejor postura del paciente y su movilidad en la vida cotidiana y así evitar el esfuerzo extremo del acompañante.

Las primeras sillas eran fabricadas de madera, pesadas y con respaldos altos. Proveían movilidad limitada y la mayoría de los usuarios no podían empujarse a sí mismos y tenían que depender de otros para empujarlos [7].

Posteriormente se empezaron a usar sillas de ruedas con un rotor que tenía una cinta enrollada a su alrededor la cual transmitía la energía a las ruedas. Con el avance tecnológico empezaron a aparecer las primeras sillas eléctricas que eran de operación manual con baterías y mecanismos de engranaje hechos de cualquier forma. Eran abultadas y difíciles de navegar ya que usaban la energía eléctrica sólo para mover las ruedas hacia adelante [7].

Las sillas de hoy en día usan transmisión directa, en donde la energía mecánica del motor se transmite directamente a las ruedas a través de un sistema de engranes o moto reductor así como ajustes motorizados para los asientos, descansos de los pies, respaldos y reposacabezas [7].

La industria ortopédica ha ido de la mano con el estudio de la parálisis cerebral. Actualmente hay más de doscientos tipos de sillas de ruedas que tienen como finalidad una buena sedestación del paciente y la movilidad del mismo. Las sillas que solo se utilizan para adaptarse al paciente no tienen funcionalidad en él, pues se deben saber datos como:

- I. Objetivos posturales (encaminados a una mejor postura)
  - i. Reducir la aparición de reflejos para poder estudiar o comer
  - ii. Normalización de la postura para evitar posturas que faciliten la hipertensión
  - iii. Control de movimientos anormales
  - iv. Facilitar el desarrollo neuromotor normal
  - v. Evitar la aparición de deformidades, la cual se tiene que estimular mediante una buena postura a través del auxilio de los padres
  - vi. Evitar la aparición de zonas de hipertensión, lo cual se consigue con el 70% del peso en isquion, trocánter, glúteos coxis y muslos. El 30% restante con los pies (19%), columna vertebral (4%) y antebrazos (2%). La falta de apoyo en este 30% es lo que origina una hiperpresión y úlceras de decúbito
- II. Objetivos funcionales (encaminados a desarrollar habilidades motrices y manipulativas)
  - i. Mejor control de la cabeza mediante una buena alineación del tronco
  - ii. Mejorar los movimientos de las extremidades superiores con la opción de acoplar una mesa para este objetivo

- iii. Facilitar el trabajo de los padres o cuidadores en actividades cotidianas como comer, beber, cambiar el pañal
- iv. Mejorar la actividad cardiorrespiratoria mediante la postura alineada
- v. Confort y seguridad
- vi. Desarrollo de habilidades cognitivas [8].

### 2.2.1 Tipos de silla de ruedas

Para la mayoría de las personas con parálisis cerebral, las sillas de ruedas suelen ser propulsadas por un acompañante, ya que como se vio en los temas anteriores, los movimientos finos y gruesos de las manos son irregulares.

#### I. Manuales

- i. Tipo portabebés: Son prescritas para niños menores de tres años aunque no tienen funcionalidad.
- ii. Sillas tipo paraguas: Son ligeras, fáciles de plegar, pero no favorecen la postura y no tiene adaptaciones. Actualmente ya no se prescriben
- iii. Sillas que permiten pocas adaptaciones: Son de asiento y respaldo rígido reclinables de 3-4 posiciones y reposapiés regulables en altura.
- iv. Sillas de tipo paseo: Son prescritas para niños de cuatro a doce años en función del peso y talla del paciente, son desmontables con reposapiés regulable en altura, respaldo para cabeza, freno en ruedas posteriores y equipo de seguridad.
- v. Silla convencional: Con asiento y respaldo semirrígidos: Permiten cambiar la posición de todo el cuerpo y cuenta con respaldo en cabeza, reposabrazos y reposapiés regulable en altura y mesa de trabajo



**Figura 2.5 Silla tipo paraguas**  
**Fuente:**  
<https://www.sunrisemedical.es/sillas-de-ruedas/zippie/silla-pci-plegable-paraguas>



**Figura 2.6 Silla tipo paseo**  
**Fuente:** <https://www.sci-geriatria.com/catalogo/sillas-ruedas/plegables/aluminio-stagi-reclinable/>



## II. Eléctricas

- i. En general, este tipo de sillas posee motores eléctricos y baterías recargables. La velocidad está relacionada con el motorreductor y la potencia de las baterías, las cuales pueden ser húmedas, de gel o selladas de ácido plomo (VRLA). En la mayoría de los modelos comerciales el control de esta silla se realiza mediante un *joystick* ubicado en el descanso brazos. Para recargar las baterías se debe de utilizar un cargador externo o sólo enchufarse a la red eléctrica en los modelos más modernos. Pueden incluir accesorios como: sistema antivuelco, capacidad para superar bordillos de 5-10 cm, subir pendientes de 15°-20°, amortiguadores, luces, claxon, indicadores de batería, asiento y respaldos reclinables manual o eléctricamente [8].



**Figura 2.8 Silla de ruedas eléctrica**  
**Fuente:** <https://ramsan.mx/es/sillas-de-ruedas-electricas-nuevas/17-silla-de-ruedas-electrica-shoprider.html>

### 2.2.2 Direccionamiento diferencial

El control de móviles a través de este tipo de direccionamiento se rige por la diferencia de velocidades en las ruedas laterales (figura 2.5), la tracción se consigue con las mismas y existen una o más ruedas de soporte para el móvil. Este tipo de sistema se caracteriza por ser de fácil implementación debido a su diseño simple, aunque posee las desventajas de ser difíciles de controlar en trayectorias rectas y la distorsión en el control cuando alguna de las dos ruedas de tracción disminuye su diámetro [12].

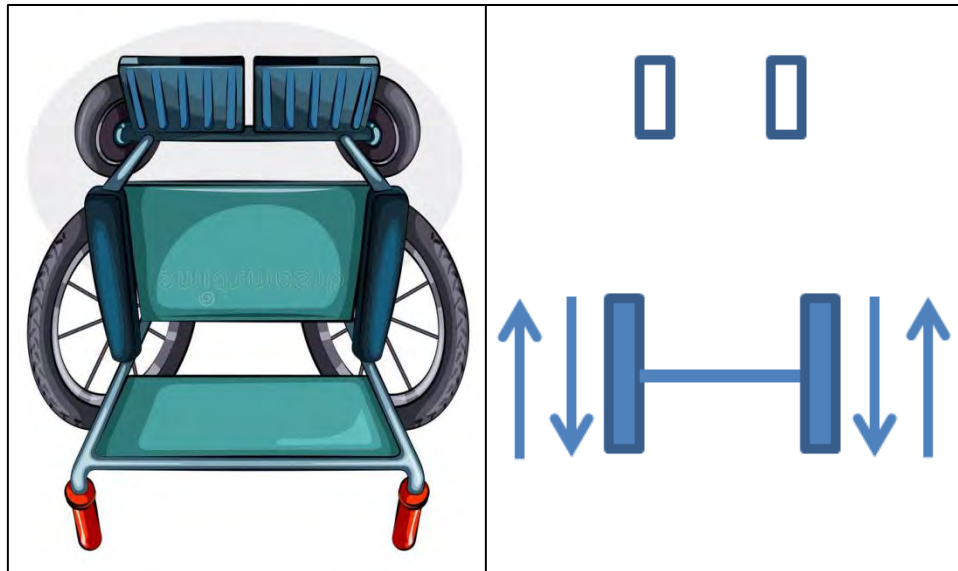


Figura 2.9 Vista aérea de las ruedas de una silla. Las ruedas traseras proporcionan la tracción al mismo tiempo que determinan la dirección  
Fuente: <https://es.dreamstime.com/imagen-de-archivo-vista-a%C3%A9rea-de-una-silla-de-ruedas-image37439891>

### 2.3 Visión artificial

Los seres humanos y las especies animales interactúan con su entorno a través de un medio que conocemos como *sistema sensorial*: *olfato*, *gusto*, *tacto*, *oído* y *vista* en el caso de los seres humanos. Mediante este sistema sensorial se transforman distintas variables físicas a través de los receptores sensoriales en impulsos eléctricos para que las neuronas sensitivas realicen el procesamiento de la información en la corteza cerebral y con ello obtener una respuesta.

El sentido de la vista es el sentido más complejo debido a que la información obtenida por este es mucho mayor que los otros sentidos en conjunto, pues podemos obtener información muy compleja como las características físicas y espaciales de un objeto (color, tamaño, distancia o trayectoria), cálculo de caminos libres, o presencia de amenazas relevantes, entre otras.



### 2.3.1 Elementos de un sistema de visión artificial

Un sistema de visión artificial o visión por computadora (figura 2.6) es un conjunto de elementos mediante el cual se obtiene una imagen digital bidimensional del escenario capturado. Para obtener esta imagen, el primer paso es la adquisición mediante un sensor, el cual podría ser monocromático o a color [17, 18].

En segunda instancia, la tarjeta de adquisición y procesamiento es la encargada de recibir las señales eléctricas provenientes del sensor para realizar el tratamiento y procesamiento de la imagen. En esta tarjeta se puede ejecutar el proceso inverso (conversión digital-analógica) y generar señales de control para manipular un elemento actuador que realizará una tarea específica [18].

Un elemento primordial en la disciplina de visión artificial y el procesamiento de imágenes es el sistema de iluminación. Dependiendo de las condiciones de este, el procesamiento se puede simplificar ahorrando bastantes procesos o tratamientos de la imagen con una iluminación adecuada y uniforme [14, 18, 20].

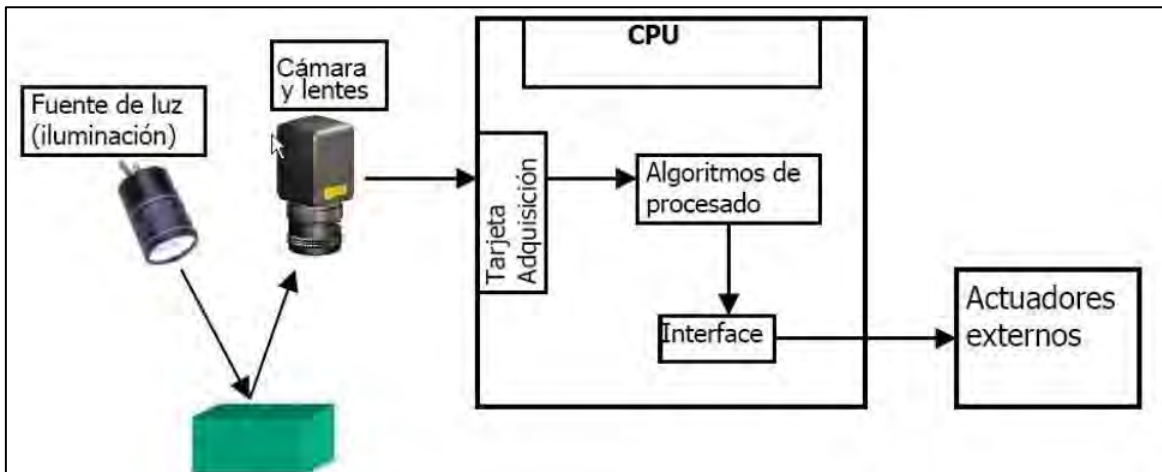


Figura 2.10 Principales elementos de un sistema de visión artificial  
Fuente: [http://visionartificial.fpcat.cat/wpcontent/uploads/UD\\_1\\_didac\\_Conceptos\\_previos.pdf](http://visionartificial.fpcat.cat/wpcontent/uploads/UD_1_didac_Conceptos_previos.pdf)

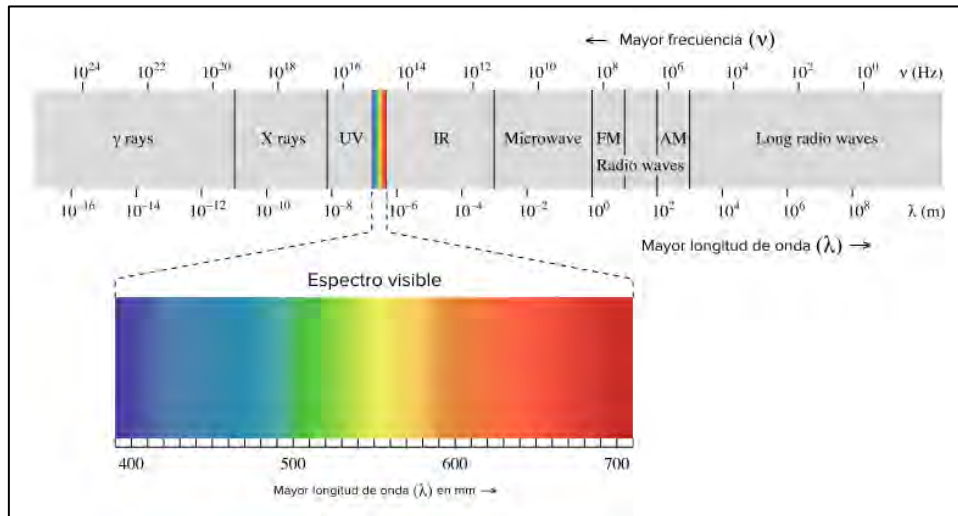
#### 2.3.1.1 Imagen

##### Luz

La luz visible corresponde a las ondas electromagnéticas que tienen longitud de onda de entre 430 y 690 nm (figura 2.7). Fuera de este rango se ubican otras ondas electromagnéticas de constitución similar pero de diferente longitud de onda, por ejemplo las ondas de radio, televisión, rayos gamma, luz ultravioleta, luz infrarroja, etc.

En el procesamiento de imágenes digitales, la luz ultravioleta e infrarroja no son útiles para la elaboración de una imagen digital, es por eso que las cámaras

fotográficas tienen un filtro IR para quitar estas ondas que están fuera del rango de la luz visible; de no tener este filtro, la imagen se contaminaría con ruido [18].



**Figura 2.11** Rango de luz visible dentro del espectro electromagnético  
Fuente: <https://es.khanacademy.org/science/physics/light-waves/introduction-to-light-waves/a/light-and-the-electromagnetic-spectrum>

## Color

El espectro de luz visible o luz blanca a su vez se puede descomponer en ondas electromagnéticas de diferente longitud, los llamados colores. Debido a la amplia gama de colores resultantes de los componentes más básicos, se han realizado diversos modelos o clasificaciones llamados “*espacios de color*”, los cuales se clasifican en dos ramas:

- I. Modelos sensoriales: Estos modelos tienen aplicaciones principalmente en equipos de proyección o de impresión, como los televisores o impresoras respectivamente (figura 2.8).
  - i. Modelo *RGB* (*Red, Green, Blue*): Este modelo se basa en los tres colores que percibe el ojo humano, rojo (*Red*), verde (*Green*) y azul (*Blue*), el modelo considera la combinación lineal de estos colores forman el color blanco es la combinación ecuánime en tonalidad de los colores RGB.
  - ii. Modelo *CMYK* (*Cyan, Magenta, Yellow, Key*): Se basa en los colores secundarios del modelo anterior. Cian (*Cyan*), magenta (*Magenta*), amarillo (*Yellow*) y negro (*Key*) son los elementos que rigen este modelo, el cual se basa en la absorción de luz.
- II. Modelos perceptuales: Estos modelos son los más cercanos a la percepción humana y son los de mayor aplicación en la visión artificial.
  - i. Modelo *HSV* (*Hue, Saturation, Value*): Este modelo es el que más se aproxima a la apreciación del color del ser humano. Es una transformación no lineal del modelo *RGB* pero en forma de cono con el cual se puede hacer progresiones de color. Matiz

(Hue), saturación (Saturation) y valor (Value) son los elementos en los que se basa este modelo.

- ii. Modelo HSL (Hue, Saturation, Lightness): Es similar al modelo HSV, pero en forma de doble cono. Los parámetros matiz (Hue), saturación (Saturation) y luminosidad (Lightness), en donde los dos vértices opuestos corresponden a la iluminancia (blanco, negro) [14,18].

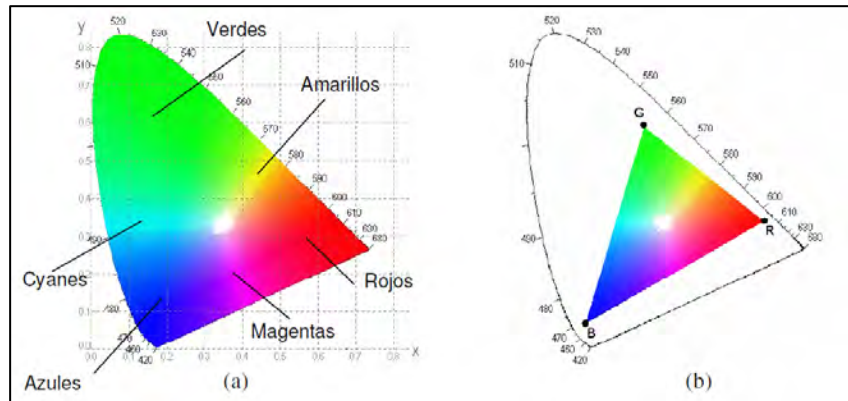


Figura 2.12  
a) Diagrama cromático RGB y CMY  
b) Triangulo cromático RGB  
Fuente: [18]

## Píxel

El píxel es la unidad mínima de color que constituye la imagen digital y que puede adquirir valores entre 0 y 255, en tres parámetros, R, G y B (rojo, azul y verde). Al adquirir valores entre 0 y 255 se puede notar que el tamaño de este píxel es de ocho bits ( $2^8$ ) como normalmente se denota, aunque también se tienen píxeles de 24 bits ( $2^{24}$ ) que son 8 bits por canal de color y 48 bits ( $2^{48}$ ) en fotografía avanzada [14].

## Imagen de mapa de bits

Las imágenes de mapa de bits o imágenes digitales son un arreglo o matriz de  $(x, y)$  resultado del muestreo de una función analógica (escenario). La imagen de mapa de bits se describe como una función analógica multiplicada por un arreglo o matriz bidimensional de funciones delta, como la siguiente ecuación:

$$f_s(x, y) = \iint_{-\infty}^{\infty} f(x, y), \delta(x - x_0, y - y_0), dx, dy \quad (2.1)$$

El resultado de la ecuación anterior ( $f_s(x, y)$ ) es el valor de la intensidad que se encuentra en un rango de 0 a 255 (imagen de 8 bits) para cada píxel (*picture element*) [14].

Una imagen monocromática es una imagen de mapa de bits donde cada píxel que la conforma posee un valor de 0 a 255, el cual representa el valor de la intensidad de luz, donde 0 es el valor para el color negro y 255 para el blanco puro. Los valores intermedios corresponden al color gris en diferentes tonalidades con valores de 1 a 254, formando una imagen en escala de grises (figura 2.9).



Figura 2.13

Imagen de mapa de bits monocromática

Fuente: <https://rebelreactor.bandcamp.com/releases>

### Imagen multi espectral

Este tipo de imágenes, a diferencia de las monocromáticas de mapas de bits, son representadas por una función vectorial ( $f_1, f_2 \dots f_n$ ) donde el resultado representa la intensidad de la longitud de onda en los componentes RGB de una imagen a color, está representada por la ecuación:

$$f(x, y) = [f_{Red}(x, y), f_{Green}(x, y), f_{Blue}(x, y)] \quad (2.2)$$

Donde  $f(x, y)$  es la imagen resultado de la intensidad de sus componentes dentro de la matriz  $(x, y)$  (figura 2.10) [14].

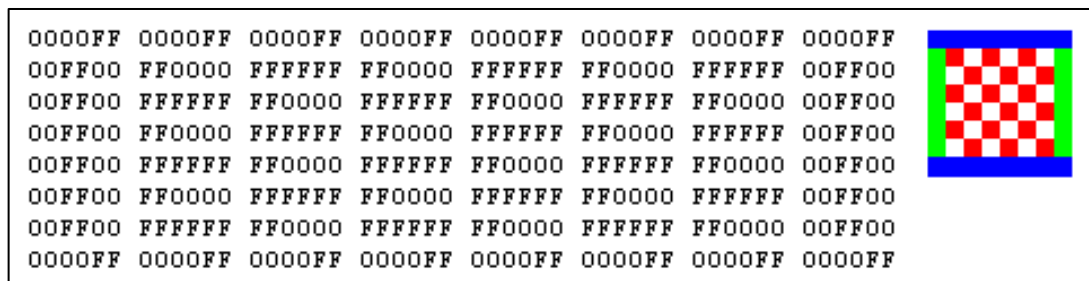


Figura 2.14

Ejemplo básico de imagen de mapa de bits de 8x8 píxeles multi espectral.

Fuente: <https://docs.microsoft.com/es-es/dotnet/framework/winforms/advanced/types-of-bitmaps>

### 2.3.1.2 Iluminación

Como se mencionó con anterioridad, la iluminación de una escena en el campo de la visión artificial es un factor importante, ya que influye de manera drástica en el procesamiento de una imagen. Por ejemplo, un escenario con la iluminación arbitraria (irregular) se puede generar un bajo contraste, nitidez, o sombras indeseadas. Por el contrario, un escenario bien iluminado puede reducir en un 60% el trabajo de la tarjeta de procesamiento.

Si se presenta un cambio brusco en la iluminación, un sistema de visión artificial funciona de una forma diferente, por lo que se recomienda que el ambiente del objetivo sea lo más estable en cuanto a luminosidad. En caso de que el ambiente no sea estable se tiene que implementar un algoritmo adaptativo, esto eleva el nivel de complejidad en cuanto a programación y procesamiento [20].

Existen diferentes técnicas de iluminación la cual va en función del tipo de análisis que se desea hacer, por ejemplo:

- I. Iluminación difusa: Se procura que los haces luminosos incidan sobre el objeto en todas direcciones. Se suele emplear en aplicaciones donde las características de la superficie del objeto sean importantes.
- II. Iluminación a contraluz: En esta técnica el objeto a analizar se pone entre la cámara y la fuente de luz. Se utiliza en aplicaciones en las que se necesita tener información de la localización o el análisis dimensional donde solo la silueta del objeto es necesaria para el análisis.
- III. Iluminación estructural: En este tipo de iluminación se trazan puntos, líneas o una cuadrícula en la escena, e donde se establece un patrón conocido sobre la escena. Realizando un proceso algebraico de “resta” de imágenes (*actual-anterior*) la cual da como resultado la detección de objetos [33].

### 2.3.1.3 Cámara

Es el elemento el cual mediante sus componentes es posible la transformación analógica a digital de los componentes de la escena (rayos luminosos) a señales eléctricas para su tratamiento. Si bien la cámara es importante, esta a su vez tiene elementos propios los cuales tienen la misma importancia en el sistema de detección [20].

#### Lentes

La función de este elemento es captar los rayos luminosos de la escena y concentrarlos sobre el elemento sensor. La elección de la lente influye en la calidad de la imagen de la escena respecto a las condiciones de la misma, como la luminosidad, distancia, etc.

Resalta una característica importante dentro de los tipos de lentes y sus modelados: el punto focal y distancia focal. El punto focal es donde convergen todos los rayos luminosos que pasan a través de la lente. La distancia entre el eje transversal de la lente y el punto focal es lo que se denomina distancia focal (figura 2.11). La distancia focal es el parámetro principal para calcular la posición y el tamaño del objetivo capturado por la cámara. La distancia focal también se relaciona con la cantidad de luz refractada por la lente [33].

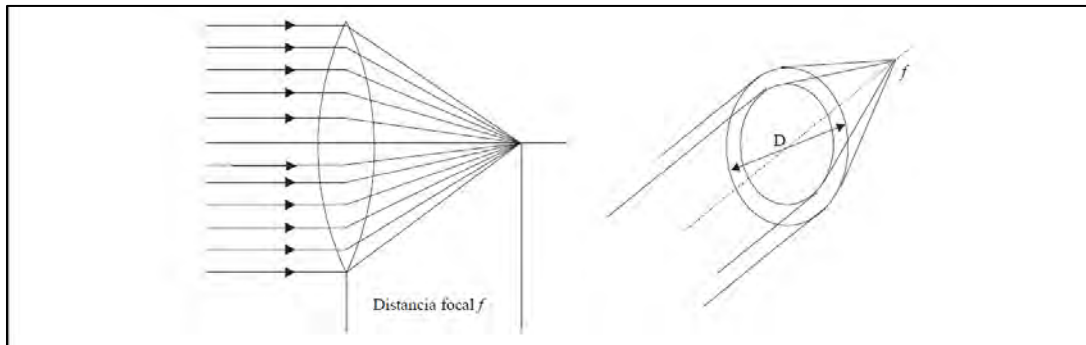


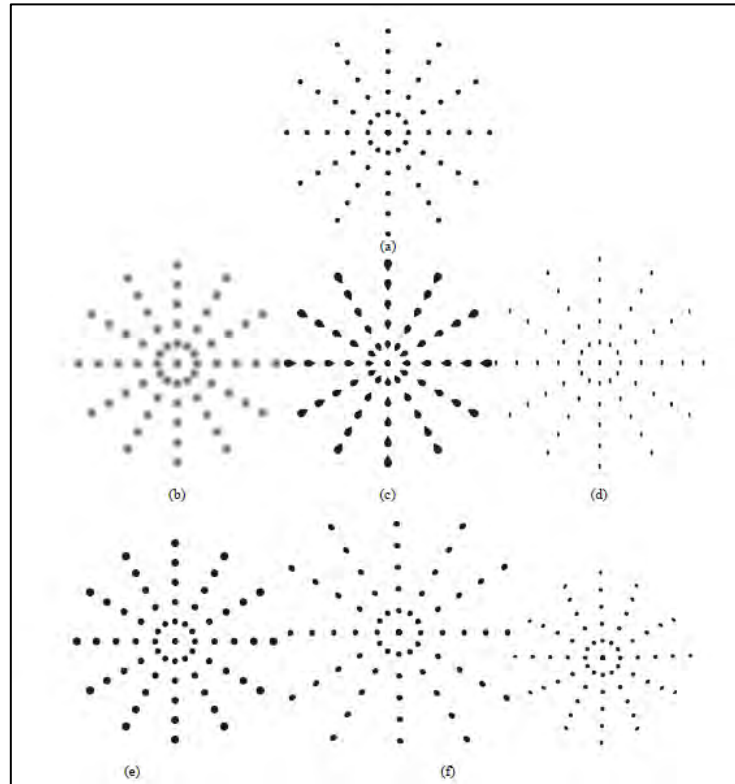
Figura 2.15  
Punto focal y distancia focal [33]

### Aberraciones ópticas

Son las imperfecciones que presentan los sistemas ópticos e introducen en la imagen capturada. Estas imperfecciones no se deben a una imperfección en las lentes, sino a que en ocasiones la luz proveniente de un objetivo no converge en el punto focal y estas aberraciones aparecen. Estas aberraciones se clasifican en dos tipos:

- I. Aberraciones cromáticas: Esta aberración se presenta por los diferentes índices de refracción de los rayos según su color o longitud de onda.
- II. Aberraciones geométricas o de Seidel: Estas se presenta por los defectos geométricos en la constitución de las lentes. Estas aberraciones son nombradas por el investigador Seidel, quien fue quién las clasificó:
  - a. Esférica:
  - b. Coma
  - c. Astigmatismo
  - d. Curvatura de campo
  - e. Distorsión

Las cuatro primeras afectan la nitidez de la imagen y la última, la forma geométrica de los objetivos como se muestra en la figura 2.12 [33].



**Figura 2.16**  
a) *Imagen sin aberraciones*  
b) *aberración esférica*  
c) *coma*  
d) *astigmatismo*  
e) *curvatura del campo*  
f) *distorsión*  
[33]

## Sensor

Son los elementos capaces de transformar las variables físicas de la escena en pulsos eléctricos para formar las imágenes digitales. Estos sensores también determinan la clasificación de las cámaras mediante la tecnología del sensor [33].

- I. **Sensor CCD (*Charged-Coupled Device*):** Los sensores CCD funcionan mediante elementos sensibles a la luz los cuales son activados por un pulso de reloj para crear “huecos” de potencial. Cuando los rayos de luz inciden sobre estos pozos se producen electrones los cuales quedan atrapados. La cantidad de electrones producidos es proporcional a la cantidad de luz que incide en el sensor (figura 2.13). Utilizan registros de desplazamiento para leer línea a línea la información proveniente del sensor y posteriormente es tratada por un convertor analógico-digital para conseguir la intensidad en cada píxel en un valor digital de 8 bits (0-255) y así formar la imagen [33].

- II. Sensor C-MOS (*Complementary Metal Oxide Semiconductor*): Los sensores C-MOS son más modernos que los CCD (figura 2.14). Estos sensores realizan la conversión analógica-digital mediante los transistores que están distribuidos en las celdas independientes de la matriz. A comparación del sensor CCD, el sensor CMOS se caracteriza por ser más barato y de menor consumo, lo que ha propiciado su reciente fama en cámaras digitales de distintas marcas [33].

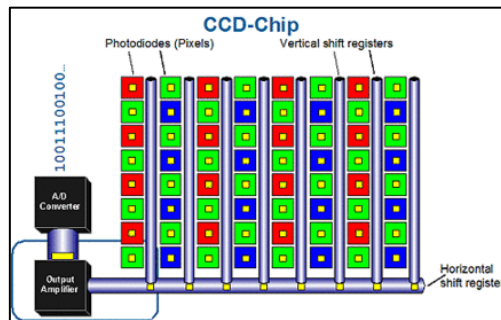


Figura 2.17 Sensor CCD

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

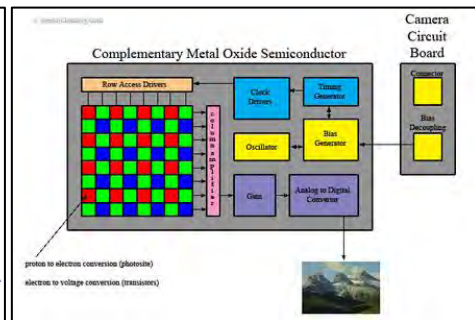


Figura 2.18 Sensor C-MOS

Fuente: [http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

### 2.3.1.4 Computadora

Es el módulo donde se realiza el procesamiento de la imagen proveniente de la cámara mediante un algoritmo. Se puede utilizar desde una computadora de escritorio hasta una computadora de placa reducida. Adicional a los elementos anteriores, se puede tener un monitor periférico mediante el cual podemos ver el resultado del procesamiento de las imágenes.

#### Raspberry Pi 2 Model B®

Es un sistema embebido muy popular debido que presentan dos ventajas muy significativas: gran procesamiento de información casi igual una computadora convencional, volumen reducido, y puertos de entrada/salida, en inglés, *Input/Output (I/O)* (figura 2.15).

Las componentes de este sistema son muy similares a las de una laptop o computadora de escritorio. Los elementos que la integran son: procesador *Broadcom® BCM2836 quad-core ARM Cortex®-A7* de 900 MHz, 1 Gb de memoria RAM, cuatro puertos *USB*, *Ethernet* 10/100, 40 pines de propósito general, en inglés, *General Purpose Input and Output (GPIO)*, conector *CSI* (cámara) y *DSI* (display), *jack* de audio de 3.5mm (audio), salida de video *HDMI* y sistema operativo Linux (alguna versión) o Windows 10 IoT (*Internet of Things*) a través del slot para *MicroSD* [35].





Figura 2.19 Raspberry Pi 2 Model B+  
Fuente: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

### OpenCV®

Existe en el mercado una variedad de software de análisis de imagen que están enfocados a distintas aplicaciones, como detección de color, contorno, movimiento, etc. *Open Source Computer Vision®* son las siglas de *OpenCV®*, una biblioteca de conjuntos de utilidades para el procesamiento de imágenes y video, multiplataforma y de código abierto ya que puede utilizarse en aplicaciones comerciales y no comerciales, es adaptable a distintos entornos de programación como *C/C++*, *Python* o *Java* por ejemplo y está disponible en plataformas como *Windows*, *Linux*, *OS X*, *Android*, *iOS* etc. [29].

### Python

Es un lenguaje de programación que se caracteriza por la simplicidad y legibilidad de su sintaxis, creado por Guido van Rossum a finales de los años ochenta. La velocidad de procesamiento es más lenta que otros lenguajes de programación como *C/C++*, pero esta desventaja se puede contrarrestar ya que se pueden utilizar código escrito en *C/C++* y utilizarlo como un módulo de *Python*. Con esto podemos obtener una ventaja importante: el código escrito en *Python* con módulos de *C/C++* puede ser tan rápido como un programa elaborado totalmente con sintaxis de *C/C++* [36].

### Numpy (Numerical Python)

Dentro del campo de la visión artificial, los cálculos numéricos en las operaciones con imágenes son bastantes complejas debido a que se trabaja con arreglos matriciales o (*arrays*), que son la representación de una imagen digital.

*Numerical Python* es una biblioteca de *Python* constituida por funciones matemáticas de alto nivel para realizar operaciones más rápido con vectores y matrices principalmente. Esta biblioteca es tan poderosa como otros programas de cómputo científico como *Wolfram Mathematica*® o *Matlab*® y al igual que *Python*® se puede utilizar en distintos entornos de programación y plataformas [37].



Figura 2.19  
OpenCV + Python

### 2.3.1.5 Elementos de salida

Son los elementos de salida del sistema mediante el cual se realizará una tarea específica. Estos por lo general son elementos eléctricos debido a la versatilidad y simplicidad en comparación con los hidráulicos o neumáticos. Las aplicaciones pueden ser diversas, por ejemplo abrir una puerta corrediza o dar un mensaje de voz de bienvenida a una persona en específico después de realizar una detección de rostro.

La cantidad, complejidad y tipos de actuadores dependen de la necesidad del sistema definida por el usuario. El uso de optoacopladores es necesario debido a que se debe mantener a los elementos de salida “aislados” del resto del sistema, pues no trabajan con la misma tensión y en muchos casos manejan un nivel de corriente muy superior. De ocurrir una falla con los elementos de salida, puede haber afectación en el resto del sistema en caso de que se tengan conexiones directas entre la parte lógica y la salida del sistema.

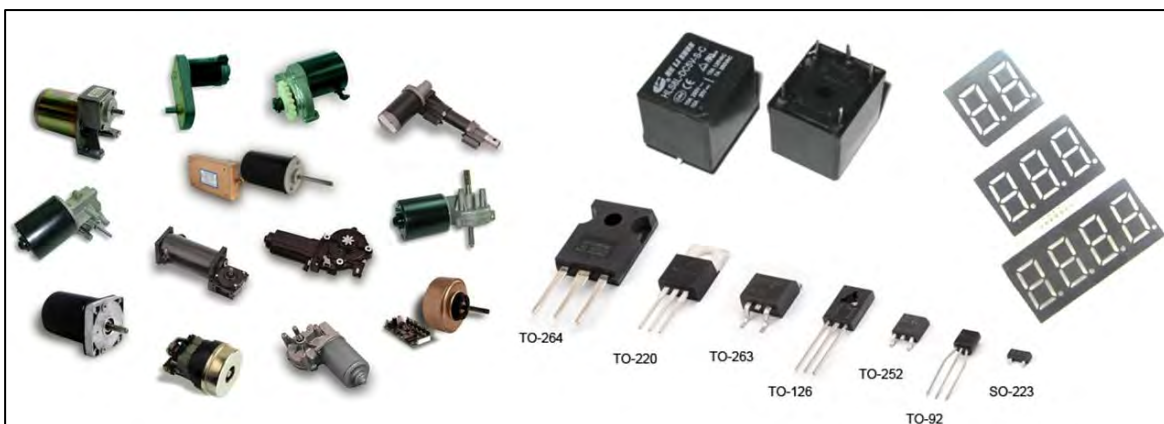


Figura 2.20  
Dispositivos de salida más comunes en sistemas electrónicos

### 2.3.2 Etapas de un sistema de visión artificial

Los sistemas de visión artificial tratan de asemejarse a la visión humana que, aunque nosotros no lo percibimos, también se realiza un proceso mediante el cual se obtiene una reacción derivada de lo que vemos. De igual forma, la visión artificial se constituye de varias etapas, como se muestra en la figura 2.17 [18].

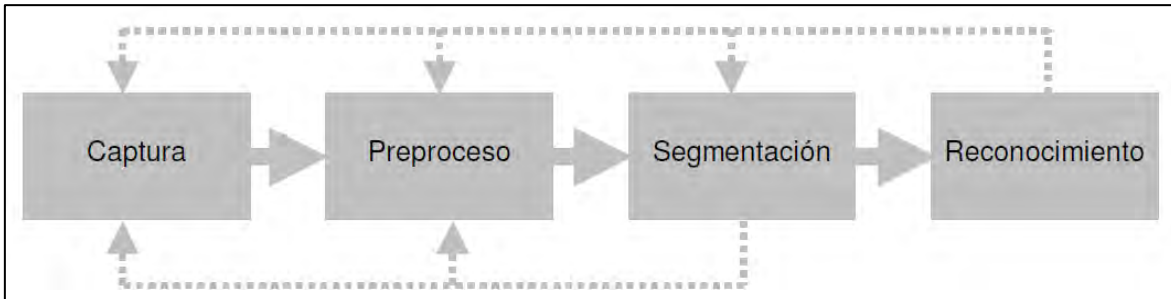


Figura 2.21  
Proceso general de la visión artificial [18]

#### Adquisición

La formación de una imagen ocurre cuando un sensor capta las ondas electromagnéticas que se reflejan de un objeto. Esta imagen es una función bidimensional  $(x, y)$  con origen en el extremo superior izquierdo de esta matriz, y cuyo valor de esta función es la brillantez o intensidad en cada punto de esta función. Una imagen se puede describir con la siguiente ecuación:

$$I = f(x, y) \quad (2.3)$$

La representación gráfica de esta función da como resultado un sistema de tres dimensiones (figura 2.18), dos corresponden a las coordenadas de la imagen y la otra a la intensidad lumínica de la misma [18].

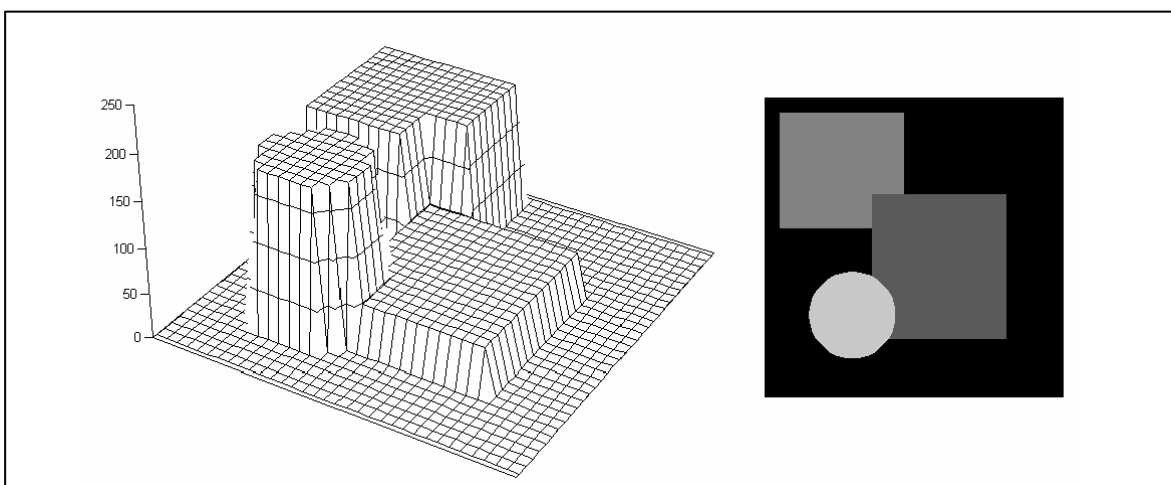


Figura 2.22  
Representación gráfica de una imagen en función de su intensidad [18]

## Pre proceso

El objetivo de esta etapa es hacer que la extracción de datos sea más simple para el ordenador. Pero ello se debe realizar algunos “tratamientos previos” de la imagen capturada para que la extracción de datos sea más sencilla.

- I. Escala de grises: En el proceso de visión artificial las imágenes a color, y el color mismo no aporta mucha información, por lo que se debe transformar esta imagen en una imagen de tonos grises mediante algoritmos de programación. Con este proceso los datos empiezan a ser más sencillos para procesar.
- II. Binarización por umbral (*Threshold binarization*): En este proceso la imagen en tonos grises obtenida del anterior, se transforma la imagen a píxeles únicamente de color negro y blanco, o sea, una imagen binaria. Se le llama binarización por umbral porque si el tono de un píxel gris se encuentra por encima o debajo de cierto umbral, se convertirá en blanco o negro respectivamente. Como resultado de esto, la imagen se puede reducir hasta 24 veces el tamaño de la imagen original.
- III. Sustracción del fondo (*Background subtraction*): Esta técnica consiste en aplicar un sencillo algoritmo, una resta. La resta consiste en sustraer el fondo de la imagen actual. Realizado esto sólo quedan los objetos que han variado respecto al fondo, obteniendo así información sobre movimiento en la escena capturada [20].



Figura 2.23

De izquierda a derecha:

I. Conversión a escala de grises, II. Binarización, III. Sustracción de fondo

Fuente: <https://www.tumblr.com/search/young%20david%20gilmour>

## Segmentación

En esta etapa se aplican algoritmos para extraer información de la imagen previamente tratada. Se puede obtener información sobre patrones geométricos, formas específicas, medida, posición o movimiento.

Igual que en la etapa de pre proceso, en esta también hay ciertas técnicas para conocer información de la imagen:

- I. Seguimiento del movimiento (*Frame difference - movement tracking*): Con esta técnica se obtienen datos sobre el movimiento y dirección del objetivo en la escena. Esto se consigue restando la escena actual capturada con la escena anterior.
- II. Seguimiento por color (*Color tracking*): Con esta técnica se puede hacer el seguimiento de una determinada área de color. Para realizarla se debe de aislar el color objetivo y obtener una imagen binaria en la que el color aislado aparecerá como una mancha blanca.
- III. Seguimiento de caras (*Face tracking*): Esta técnica se basa en reconocer patrones lo más cercano a una “cara”, de la que se puede extraer información como las medidas, posición o aspectos morfológicos.
- IV. Seguimiento de características (*Feature tracking*): Esta técnica se basa en el reconocimiento de características específicas en una escena y que son fácilmente reconocibles por los algoritmos, como los bordes y las esquinas.
- V. Reconocimiento de regiones (*Bloob detection*): Es el reconocimiento de blobs o regiones más luminosas que las que lo rodean, que en consecuencia, forman una mancha. A partir de esta técnica se puede obtener información de la escena como la cantidad de objetos, medidas, contorno o centro de masa [20].

## Reconocimiento

En esta etapa se debe realizar una transformación del o los objetos a analizar en vectores “x” de “N” componentes para realizar una clasificación automática. Los “N” componentes de los vectores usualmente se denominan como características discriminantes o rasgos. Estas características son las que determinan a qué clase de conjuntos pertenece cada objeto, representándose esta clasificación en la siguiente expresión:

$$X = (X_1, X_2, X_3, \dots, X_N,) \text{ con } N \in \mathbb{N}, \quad X_N \in \mathbb{R}, \quad \forall_N = 1 \dots N \quad (2.4)$$

Una vez definidos los rasgos, la clasificación de los objetos comienza por su patrón, que es la instancia particular del vector “x” mencionado. Posteriormente se determina la proximidad de este patrón al conjunto de características o rasgos; las funciones discriminantes o funciones de decisión son las encargadas de realizar esta asignación del grado de semejanza del objeto (figura 2.20) [18].

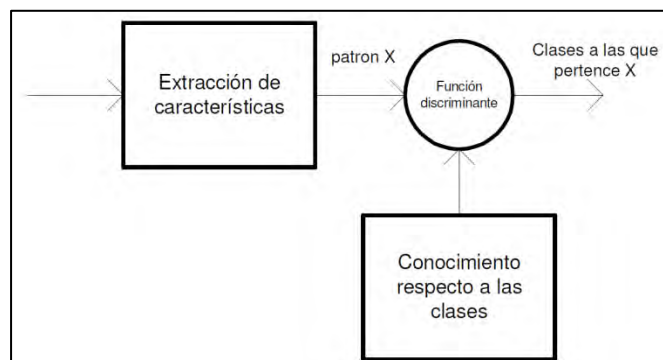


Figura 2.24  
Diagrama básico del proceso de clasificación [18]

### 2.3.3 Niveles de visión artificial

La visión artificial es un proceso complejo. Debido a esta complejidad se ha clasificado en diferentes niveles de visión, en los cuales la información y se va reduciendo hasta llegar a la descripción deseada para ejecutar la tarea específica.

Generalmente se consideran tres niveles:

- I. Procesamiento de bajo nivel: En esta etapa el procesamiento se hace directamente con los píxeles de la imagen. La información más común que suele extraerse de una imagen digital es el color, profundidad, textura etc.
- II. Procesamiento de nivel intermedio: En este nivel la información obtenida del nivel anterior se utiliza para obtener los contornos, líneas o regiones con el propósito de segmentar la imagen. Esta segmentación consiste en diferenciar los escenarios y determinar la posición de un objeto a través de sus características físicas como la forma o el color.
- III. Procesamiento de alto nivel: El propósito de este nivel es la interpretación de la información obtenida de los niveles anteriores. Las estructuras de datos son aún más complejas y se requiere de un trabajo de cómputo muy grande, por lo que son sistemas no tan comunes debido a la respuesta no instantánea [14].

## 2.4 Advanced RISC Machine (ARM®)

ARM® es una arquitectura RISC, las siglas en inglés de *Reduced Instruction Set Computer* creada en 1985 por *Acorn Computers Limited*® en el Reino Unido. Las características por las que se hizo muy popular de 1985 a la fecha es que utiliza menos transistores para economizar espacio y la reducción en el consumo eléctrico.

Existen diversos fabricantes como *Apple Inc.*®, *Atmel*®, *Broadcom*®, *Freescale*® entre otros, que son titulares de licencias de esta arquitectura bajo la cual diseñan sus circuitos integrados. Desde el año 2009 la popularidad de ARM® pasó de 90% a 98% en chips de 32-bits de esta arquitectura con aplicaciones en la telefonía móvil, calculadoras, reproductores de audio y video y videojuegos [21].

### 2.4.1 Características de la arquitectura ARM®

ARM® mantiene algunas características heredadas del diseño *Reduced Instruction Set Computer (RISC)* de sus primeros procesadores:

- I. Arquitectura de carga-almacenamiento: Las instrucciones de carga-almacenamiento son operaciones que se aplican solamente a la memoria como la copia de valores de memoria a un registro (instrucción de carga) o valores de registro copiados a la memoria (instrucción de almacenamiento). Estas instrucciones son independientes una de la otra.

- II. Instrucciones de ancho fijo: Instrucciones de ancho de 32-bits: Con esta longitud se simplifica la carga y decodificación de manera rápida y simple ya que no es necesario conocer el ancho de la instrucción que se está ejecutando para comenzar a decodificar la siguiente en el pipeline.
- III. Formatos de instrucción de tres direcciones: El formato cuenta con “f” bits correspondientes al código de operación, “n” bits para especificar la dirección en registros del primer operando, “n” bits para la dirección del segundo operando y “n” bits para el resultado o destino. Los bits “f” y “n” (fuente-destino) son independientes uno del otro [21].

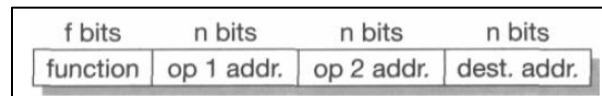


Figura 2.25  
Instrucción de tres direcciones [21]

Adicional a las características heredadas de la arquitectura *RISC*, también se integraron características propias de *ARM®* que hacen más eficientes a los procesadores por el bajo consumo de energía, menor cantidad de silicio para la fabricación y el reducido tamaño del código.

- I. Arquitectura *Thumb*: Esta arquitectura consiste en usar instrucciones de 16 bits del repertorio original de instrucciones de 32 bits para mejorar la densidad de código. Para establecer este modo de operación se debe poner 1 en el bit 5 de la instrucción estándar de 32 bits. No todos los procesadores son capaces de ejecutar este modo de operación, por lo que se denota con una “T” a los procesadores capaces de ejecutar este modo de operación, por ejemplo el *ARM7TDMI®*.
- II. Técnica *pipeline*: Para economizar tiempo *ARM®* implemento esta arquitectura que básicamente consiste en ejecutar la siguiente instrucción antes de terminar la instrucción actual. Con ello se realiza más de una instrucción en un ciclo de reloj.
- III. Ejecución rápida: Todas las instrucciones se establecen en un ciclo de reloj en el *pipeline*.
- IV. Modos de direccionamiento: Se establecen incrementos y decrementos en punteros para mejorar los saltos de programa.
- V. Set de instrucciones simétrico: Es un set de instrucciones de propósito general. [21]

#### 2.4.2 Set de instrucciones de *ARM®*

Hay alrededor de 4 billones de instrucciones binarias diferentes en *ARM®* de 32-bits las cuales todas caen dentro de alguna de las siguientes categorías quienes pertenecen a la arquitectura carga-almacenamiento:

- I. Instrucciones de procesamiento: Son instrucciones usadas solamente en los valores de registros, como una instrucción que suma dos registros y coloca el resultado en un registro.
- II. Instrucciones de transferencia: Estas instrucciones copian valores de la memoria dentro de un registro, copian valores de un registro dentro de la memoria o intercambian un valor de memoria con un valor de registro. Esta última función es útil solamente usando código de sistema.
- III. Instrucciones de control de flujo: Son un tipo de instrucciones que permiten el cambio de dirección del proceso que se está ejecutando. El cambio de dirección puede ser permanente o temporal, guardando la dirección de retorno para volver a la secuencia original [21, 22].

### 2.4.3 Procesadores ARM Cortex®

La marca Cortex® de ARM® tiene una gran variedad de procesadores que a su vez se clasifican en subfamilias:

- I. Cortex®-A: Procesadores de aplicación (*Cortex®-Application*) de alto rendimiento, con instrucciones de 32-bits, así como *Thumb2* con instrucciones de 16/32bits con soporte para datos de 64 bits. Soporte para sistemas operativos como *Linux®*, *Android®*, *Chrome®* y *MontaVista®*. Las aplicaciones de estos procesadores son principalmente en *smartphones*, *tablets*, *routers*, sistemas embebidos y aplicaciones automotrices.
- II. Cortex®-M: Procesadores para aplicaciones de microcontroladores (*Cortex®-Microcontroller*) de 32-bits con *Thumb2* incorporado para maximizar el rendimiento de la memoria Flash. Fácil desarrollo por ser completamente programables en C y compatibilidad para transmisión de datos de 8, 16 y 32 bits. La serie M se utiliza principalmente en aplicaciones embebidas de bajo costo, reducido consumo y alto rendimiento como instrumentación médica, dispositivos inalámbricos, aplicaciones automotrices e industriales.
- III. Cortex®-R: Procesadores de aplicaciones en tiempo real (*Cortex®-Real Time*) principalmente en aplicaciones críticas que requieren respuesta en tiempo real, tolerancia a fallos y cumplimiento con los tiempos de sincronización, para evitar situaciones de riesgo. Se utilizan principalmente en aplicaciones médicas o sistemas autónomos [38].

#### 2.4.3.1 Procesador BCM2836®

Proviene de la familia Cortex®-A7 que tiene soporte para micro computadoras de placa reducida, en inglés *Single Board Computer*. Es un procesador de alto desempeño y bajo consumo de cuatro núcleos con una velocidad de 900 MHz hasta 1 GHz, fabricado por la empresa estadounidense *Broadcom®* (figura 2.22). Este procesador es la unidad central de procesamiento de la micro computadora Raspberry Pi 2 Model B, haciéndola una



computadora versátil por la variedad de aplicaciones en diferentes sistemas operativos, así como la cantidad de periféricos disponibles [24].

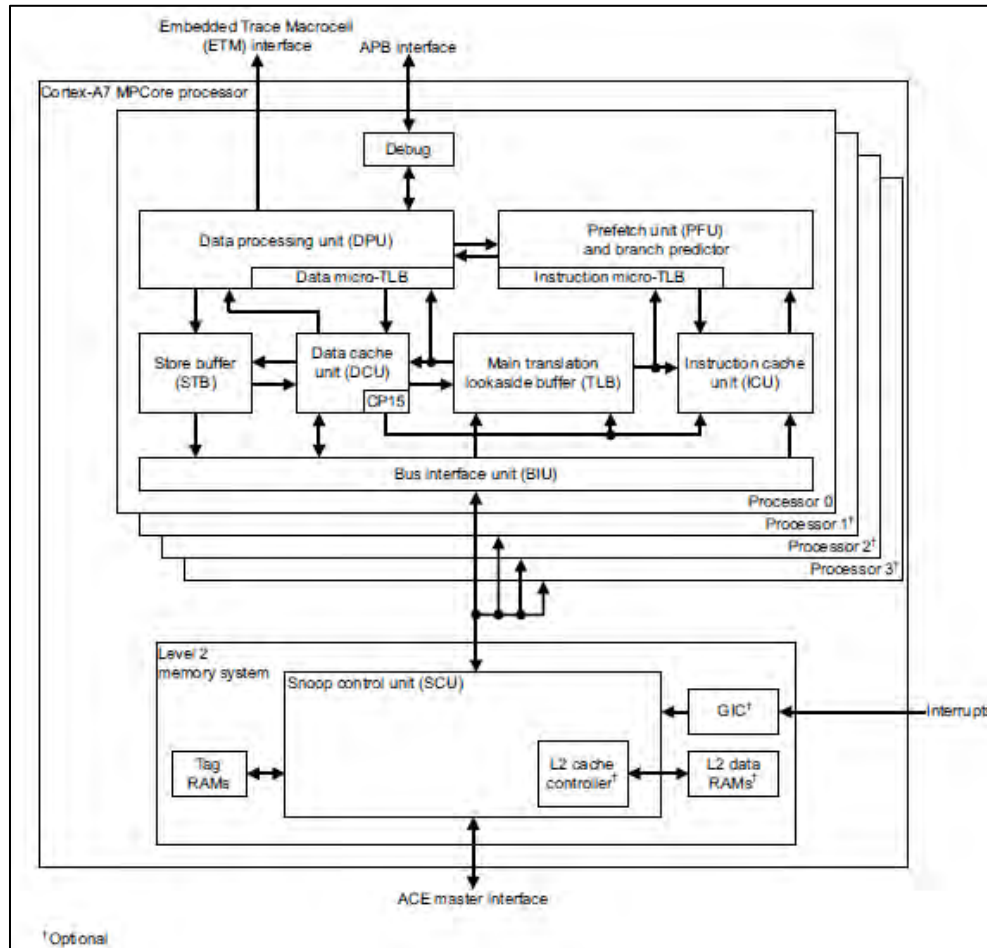


Figura 2.26  
Estructura general del procesador BCM2836 [24].

### 2.4.3.2 Micro controlador MKL25Z128VLK4®

Es un micro controlador de la familia *Cortex®-M0+* de 32-bits de ultra-bajo consumo y alta eficiencia fabricado por la empresa *Freescale®* (ahora *NXP®*). Las cualidades más importantes son: 48 MHz de reloj, 128 KB de memoria Flash, 15 KB de memoria RAM estática (*Static RAM*), tres puertos serie, dos convertidores analógico/digital, un puerto de comunicación *SPI* de 8 bits, dos puertos *I<sup>2</sup>C*, seis puertos *Timer/PWM* y un reloj de tiempo real.

Este micro controlador se encuentra en la tarjeta de desarrollo *FRDM-KL25Z®* explotando sus características al máximo en conjunto con el acelerómetro digital de tres ejes, un *LED RGB*, un *slider* capacitivo y el firmware *OpenSDA®* para la programación [39].

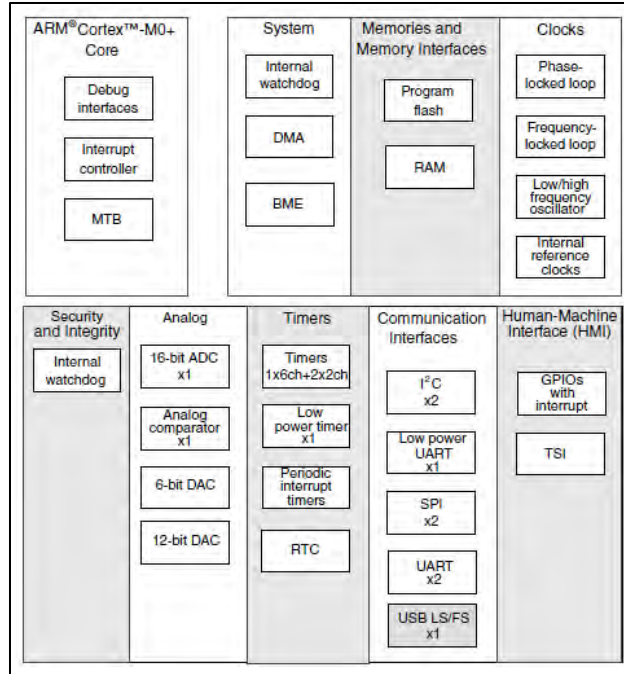


Figura 2.27  
Estructura general del microcontrolador MKL25Z128VLK4 [39]

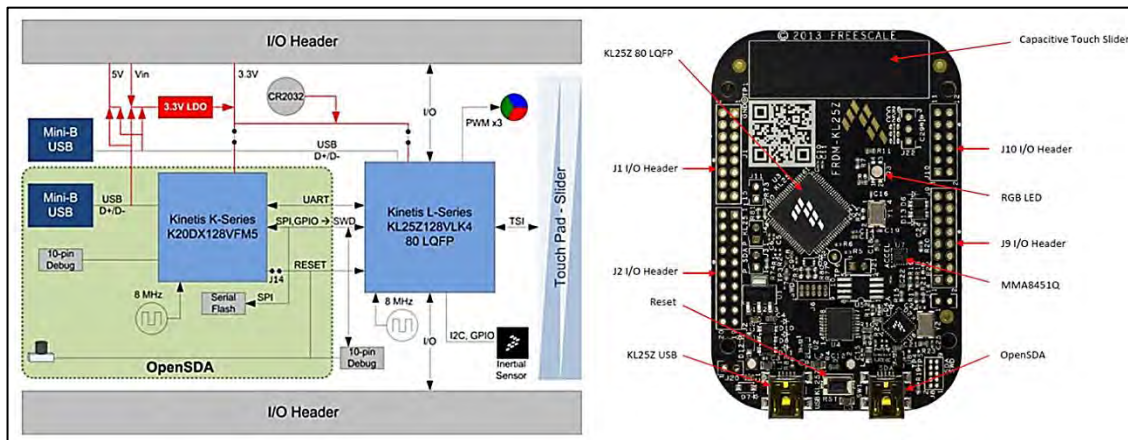


Figura 2.28  
Diagrama de bloques la tarjeta FRDM KL25Z  
Fuente: <https://www.nxp.com/docs/en/user-guide/FRDMKL25ZUM.zip>

## 2.5 Transceptores de radiofrecuencia

Los transceptores, en inglés, *Transceiver* (*Trans*-mitter, re-*ceiver*), de radiofrecuencia son dispositivos integrados por un transmisor y receptor dentro del mismo encapsulado. Las funciones de transmisión y recepción se realizan de manera alternada y nunca al mismo tiempo. Cuando el transmisor envía información es posible que sea “escuchado” por uno o varios receptores, cuando se trata de la topología punto a punto o multipunto [44].

El proceso de transmisión (figura 2.29) se resume en las siguientes etapas:

- I. Codificación: Adecuación de la información a transmitir, puede ser del tipo analógica o digital.
- II. Generar onda portadora: Producción de la señal a través de la cual se enviará la información.
- III. Modulación de señal: Modulación de la onda portadora.
- IV. Amplificación: Brindar la potencia necesaria a la onda portadora para que pueda llegar hasta los receptores.
- V. Filtrado: Eliminar rangos de frecuencia innecesarios para el receptor.
- VI. Radiación: Transmisión de la onda portadora mediante la antena.

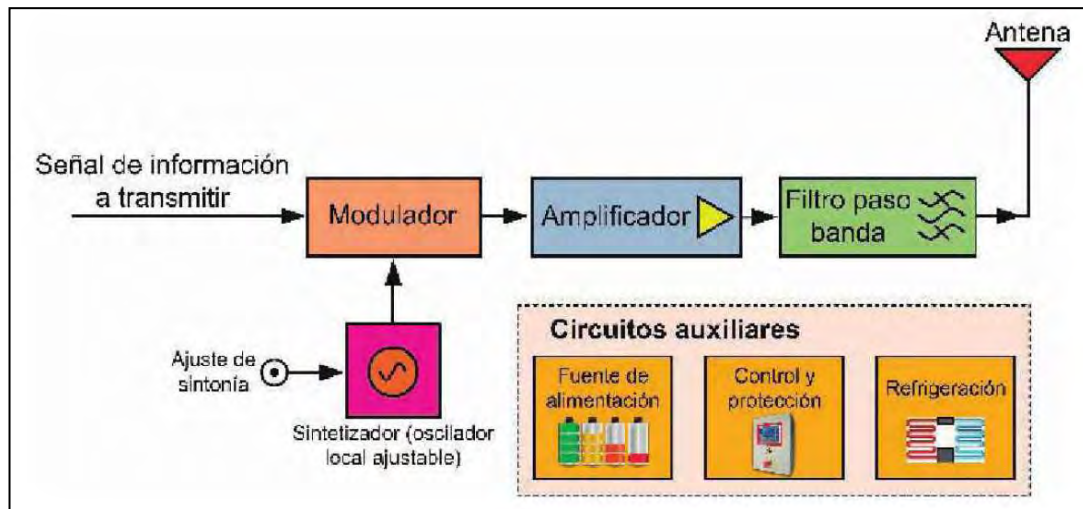


Figura 2.29  
Diagrama de bloques de un transmisor de radiofrecuencia [45]

El proceso de recepción (figura 2.30) se conforma de las siguientes etapas:

- I. Captación: Entrada de la señal modulada la cual genera una corriente a través de la antena.
- II. Filtrado: Eliminar ruido y otro tipo de interferencias para evitar alguna alteración en la información.
- III. Demodulación: Recuperación de información de la señal modulada.
- IV. Amplificación: Elevar la potencia de la señal entrante la cual sufrió una atenuación a consecuencia de la distancia entre receptor y transmisor.

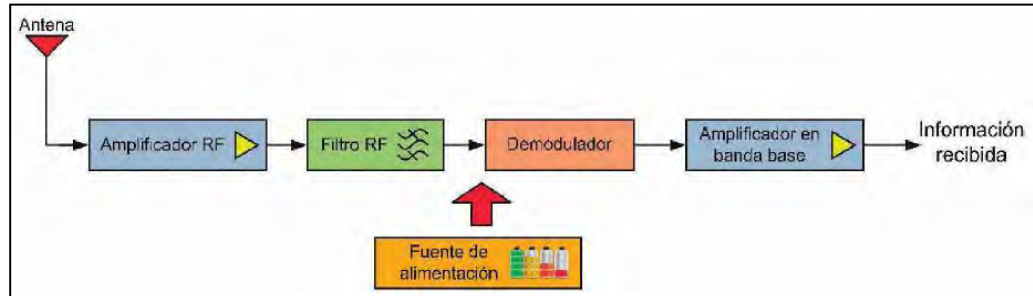


Figura 2.30  
Diagrama de bloques de un receptor de radiofrecuencia [45]

### 2.5.1 Módulo NRF24L01®

Es un módulo de radiofrecuencia (figura 2.31) de hardware libre, el cual tiene como elemento principal el transceptor *NRF24L01®* del fabricante *Nordic Semiconductor®*. Las principales cualidades de este dispositivo son: frecuencia de operación de 2.4 GHz, 126 canales, envío de información de 1 hasta 32 bytes, regulador de voltaje integrado, voltaje de operación de 1.9 V<sub>CD</sub> hasta 3.6 V<sub>CD</sub>, bajo consumo de 12.3 mA, interfaz periférica serial, en inglés, *Serial Peripheral Interface (SPI)*.

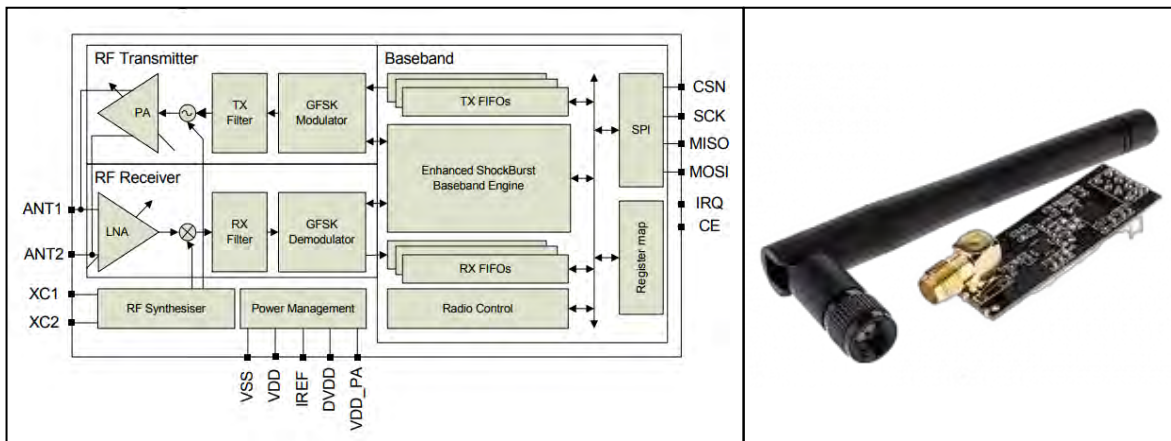


Figura 2.31  
Diagrama de bloques y módulo transceptor NRF24L01®

# Capítulo 3

## Consideraciones del sistema de control

Un sistema de control basa su funcionamiento en variables de entrada y salida que son definidas de acuerdo a la aplicación requerida. La funcionalidad del sistema de control de un vehículo con ruedas se compone de varios subsistemas necesarios para poder ejecutar un direccionamiento eficaz. Si alguno de estos elementos falla, se generan movimientos indeseados o la inmovilidad total.

- I. Software de aplicación: Software o programa que realiza el procesamiento de la información de las variables de entrada e indica el movimiento y las condiciones para realizar el mismo.
- II. Controlador de movimiento: Encargado de interpretar los datos del software de aplicación y emitir señales las cuales deben cumplir las condiciones del movimiento requerido.
- III. Amplificador o drive: Elemento que magnifica las señales de control emitidas en el sistema anterior para mover los actuadores de salida del sistema.
- IV. Motor: Elemento que convierte energía eléctrica en mecánica y proporciona el movimiento del sistema.
- V. Elementos mecánicos: Elementos adicionales que son manipulados por el motor.
- VI. Dispositivos de retroalimentación: Elementos que se encargan del medir la variable de salida en el caso de un sistema cerrado [40].

### 3.1 Movilidad de la silla de ruedas

En este tipo de vehículo, la movilidad se consigue en base al direccionamiento diferencial, el cual se define por la diferencia de velocidades en las ruedas traseras, generando la tracción y permite a la silla de ruedas tener tres direcciones de avance, tres de retroceso y dos movimientos rotacionales. El par de ruedas delanteras son orientables y de libre movimiento.

El resultado del direccionamiento diferencial se puede trazar en un plano cartesiano en el cual se tiene acceso a los cuatro cuadrantes tomando como referencia al vehículo como origen (figura 3.1) y mediante el cual se definen las condiciones para lograr los siguientes movimientos.

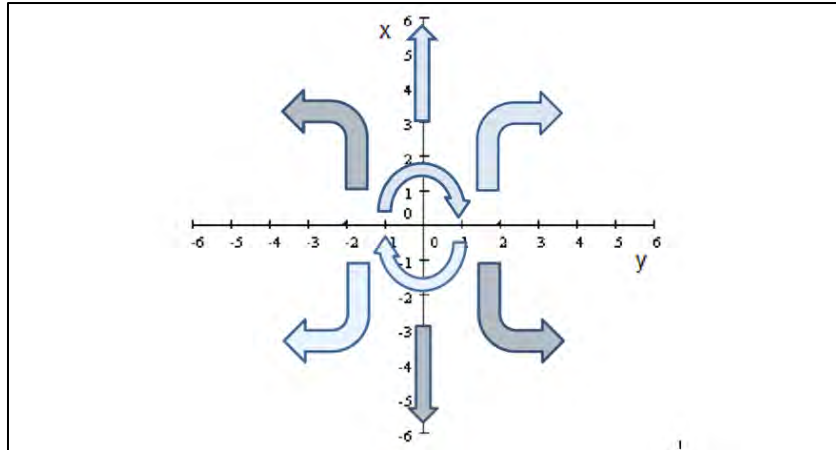


Figura 3.1  
*Movimientos resultantes del direccionamiento diferencial*

#### Movimientos primarios.

- I. Adelante: Ambas ruedas giran a la misma velocidad y en el mismo sentido.
- II. Atrás: Ambas ruedas giran a la misma velocidad y en reversa
- III. Giro a la izquierda: La rueda izquierda gira en reversa y la derecha hacia adelante, creando con esto un giro sobre su propio eje.
- IV. Giro a la derecha: La rueda derecha gira en reversa y la derecha hacia adelante, creando con esto un giro sobre su propio eje.

#### Movimientos en cuadrantes

- I. Movimiento en cuadrante I: Ambas ruedas giran hacia adelante. La izquierda gira más rápido que la derecha para tener un avance hacia adelante y a la derecha.
- II. Movimiento en cuadrante II: Ambas ruedas giran hacia adelante. La derecha gira más rápido que la izquierda para tener un avance hacia adelante y la izquierda.
- III. Movimiento en cuadrante III: Ambas ruedas giran hacia atrás. La derecha gira en reversa más rápido que la izquierda para provocar un movimiento en reversa hacia la izquierda.
- IV. Movimiento en cuadrante IV: Ambas ruedas giran hacia atrás. La izquierda gira en reversa más rápido que la derecha para provocar un movimiento en reversa hacia la derecha.

### 3.2 Consideraciones del sistema de visión artificial para el direccionamiento

El sistema de visión artificial de la silla tiene como objetivo darle movilidad al usuario con parálisis cerebral y evitar el esfuerzo de una segunda persona para empujar la silla. Se realizan procedimientos básicos de un sistema de visión artificial y en base a predicciones del comportamiento del objetivo en el escenario se obtiene el área y movimiento sobre el eje x, parámetros con los que se definen los siguientes movimientos.

#### Movimientos primarios

- I. Adelante: El objetivo va reduciendo su tamaño, esto indica que gana profundidad en la imagen y se mueve hacia adelante, en consecuencia ambas ruedas se mueven hacia el frente a la misma velocidad.
- II. Atrás: El objetivo aumenta su tamaño en la escena, esto nos indica que se dirige hacia atrás. Ambas ruedas se mueven en retroceso a la misma velocidad.
- III. Giro a la izquierda: El objetivo, sin cambiar su tamaño, se mueve hacia la derecha de la escena. La rueda diestra gira hacia delante y la izquierda hacia atrás.
- IV. Giro a la derecha: El objetivo, sin cambiar su tamaño, se mueve hacia la izquierda de la escena. La rueda correspondiente gira hacia adelante y la derecha hacia atrás.

#### Movimiento en cuadrantes

- I. Movimiento en cuadrante I: El objetivo reduce su tamaño y se desplaza a la derecha de la escena, por lo que ambas ruedas giran hacia adelante pero la rueda izquierda gira más rápido que la derecha para tener un avance hacia la derecha.
- II. Movimiento en cuadrante II: El objetivo reduce su tamaño y se desplaza a la izquierda de la escena, por lo que ambas ruedas giran hacia adelante pero la rueda derecha gira más rápido que la contraria para tener un avance hacia la izquierda.
- III. Movimiento en cuadrante III: El objetivo aumenta su tamaño y se desplaza hacia la izquierda de la escena, por lo que ambas ruedas giran hacia atrás, pero la rueda derecha gira en reversa más rápido que la izquierda para provocar un movimiento en reversa hacia la izquierda.
- IV. Movimiento en cuadrante IV: El objetivo aumenta su tamaño y se desplaza hacia la derecha de la escena, por lo que ambas ruedas giran hacia atrás, pero la rueda izquierda gira en reversa más rápido que la contraria para provocar un movimiento en reversa hacia la izquierda.

El direccionamiento de las ruedas depende de los siguientes movimientos del objetivo en la escena capturada:

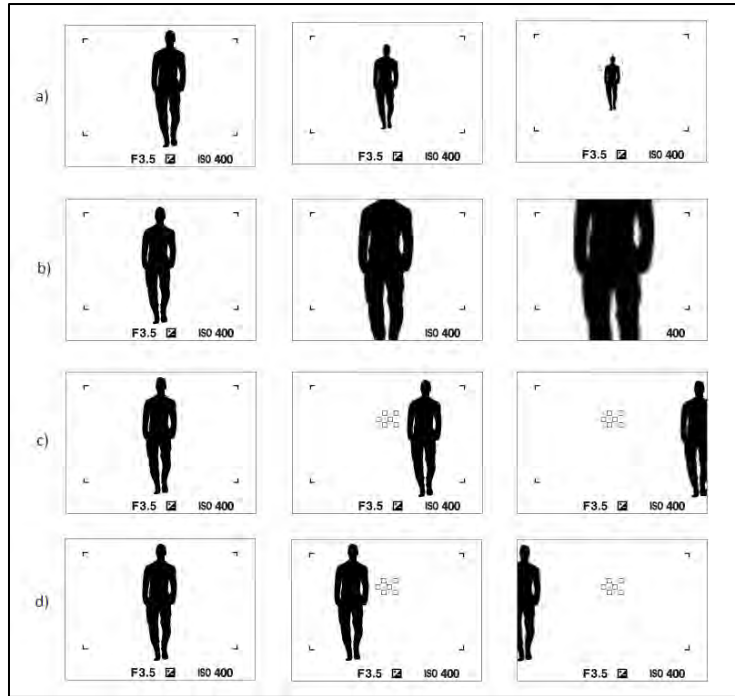


Figura 3.2

Movimientos primarios:

a) adelante, b) atrás, c) giro a la derecha, d) giro a la izquierda

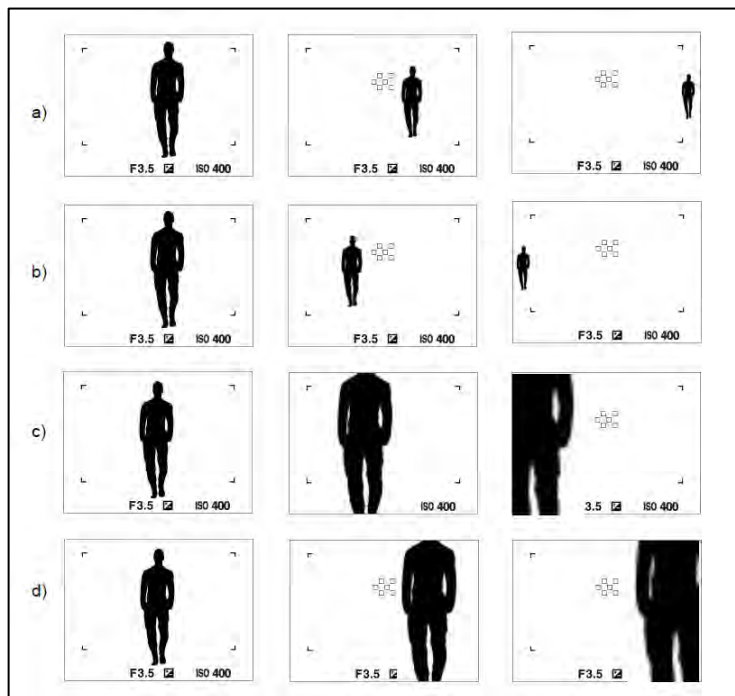


Figura 3.3

Movimientos compuestos:

a) adelante-derecha, b) adelante-izquierda, c) atrás-izquierda, d) atrás-derecha



Las predicciones anteriores se obtienen a partir de un estado de inmovilidad, el cual se presenta cuando el objetivo está en el centro y con un área definida en la imagen capturada.

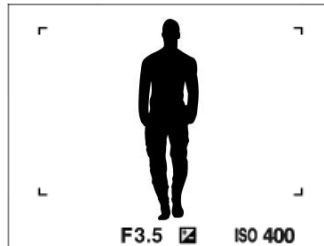


Figura 3.4 Patrón de inmovilidad de la silla

### 3.3 Consideraciones del sistema manual para el direccionamiento

El sistema de direccionamiento manual se basa la operación del acelerómetro integrado en la tarjeta *FRDM-KL25Z*®. El resultado de la inclinación de este elemento es un direccionamiento similar al sistema de visión artificial con los mismos resultados, movimientos primarios y en cuadrantes del plano cartesiano.

Respecto la posición del acelerómetro en la tarjeta (figura 3.6), solamente se utiliza los ejes (x, y) del mismo para generar los movimientos del direccionamiento diferencial inclinando la tarjeta como se muestra en la siguiente figura.

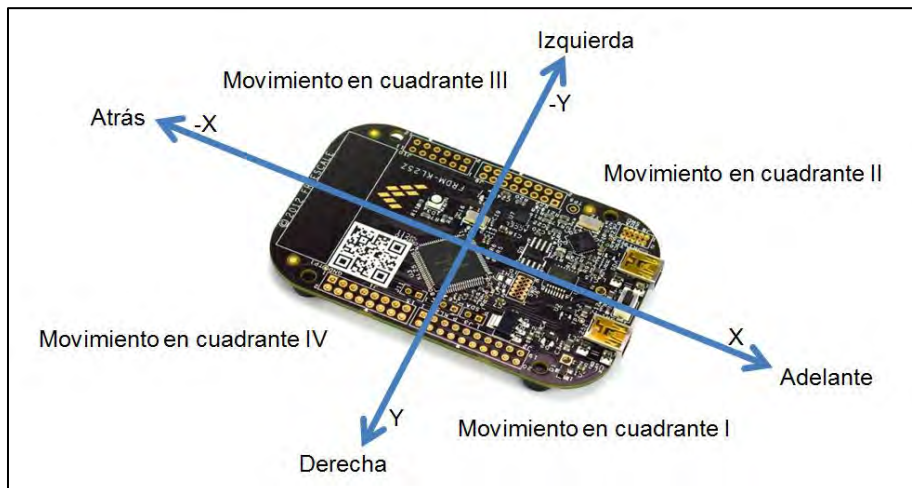
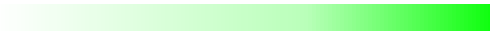









Figura 3.5  
Inclinaciones de la tarjeta

Se incorpora también un indicador de color para expresar el sentido y la velocidad de la silla de ruedas mediante el *LED RGB* incorporado en la tarjeta *FRDM-KL25Z®*.

- Adelante: 
- Atrás: 
- Giro a la izquierda: 
- Giro a la derecha: 
- Movimiento compuesto I: 
- Movimiento compuesto II: 
- Movimiento compuesto III: 
- Movimiento compuesto IV: 

### 3.4 Control de motores para el direccionamiento

La velocidad de un motor de C.D. depende de la corriente que circula en el devanado de la armadura. Para controlar la velocidad de un motor es necesario controlar el voltaje que se aplica a la armadura. La modulación por ancho de pulso o (*PWM*) es la técnica más común para variar la velocidad de un motor mediante dispositivos digitales.

Para cambiar el sentido de giro de un motor, se utiliza un puente H (figura 3.7), el cual se compone principalmente de cuatro transistores, tal como se muestra en la siguiente figura [10]:

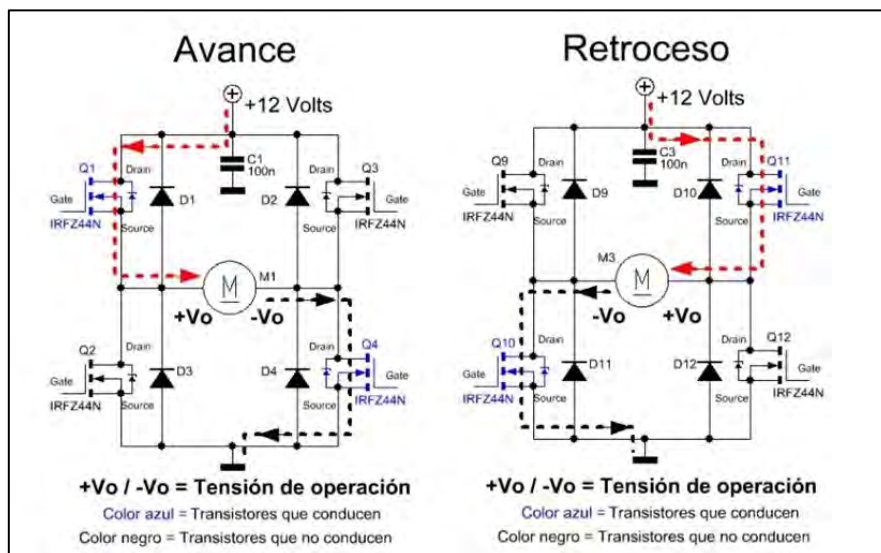


Figura 3.6  
Estructura y funcionamiento del puente H  
Fuente: <https://www.neoteo.com/wp-content/images/A0DC.jpg>

Las ventajas de utilizar transistores de efecto de campo de metal óxido, en inglés, *Metal-oxide-semiconductor Field-Effect Transistor (MOSFET)* es que los estados de conducción se controlan mediante voltaje, el coeficiente de temperatura es positivo (se calienta menos), tienen un valor de resistencia bajo en la salida ( $R_{DS}$ ) y poseen un diodo de protección (Damper) incorporado que evita la retroalimentación de corrientes destructivas para el circuito.

Este tipo de control considera condiciones de operación que permanecen constantes y sin perturbaciones, como el voltaje de alimentación del motor o la carga desplazada [10].

# Capítulo 4

## Implementación de software y diseño de hardware de hardware

### 4.1 Funcionamiento del sistema

Los principales componentes del sistema de control son: la tarjeta *Raspberry Pi 2 Model B*® la cual tiene ejecuta las tareas de adquisición, procesamiento de imágenes y transmisión de datos para el seguimiento del objetivo. La tarjeta *FRDM-KL25Z*® controladora, realiza la interpretación de los datos generados por el sistema de visión artificial y sistema manual para generar las señales modulación por ancho de pulso (*PWM*) y que con ayuda de dos puente H, establecer la dirección de giro y velocidad en los motores (figura 4.1).

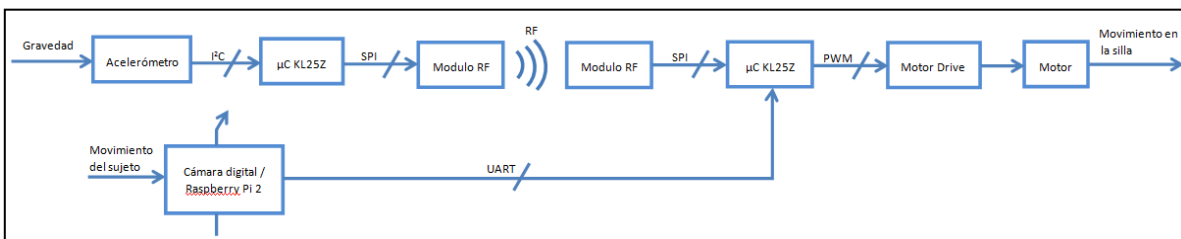


Figura 4.1  
Diagrama de bloques del sistema de control.



Figura 4.2  
Ubicación de los componentes de visión artificial, control de motores, radiofrecuencia y motores.

### 4.2 Algoritmo del sistema de visión artificial

Los sistemas de visión artificial se constituyen de etapas las cuales tratan de mantener una semejanza con la visión humana obteniendo como resultado una acción o tarea específica. En la micro computadora *Raspberry Pi 2 model B®* se implementa la adquisición, pre proceso, segmentación y reconocimiento (etapas de un sistema de visión artificial) en un algoritmo de lenguaje *Python*. El diagrama de flujo del algoritmo y sus procesos se muestran en la figura 4.2.

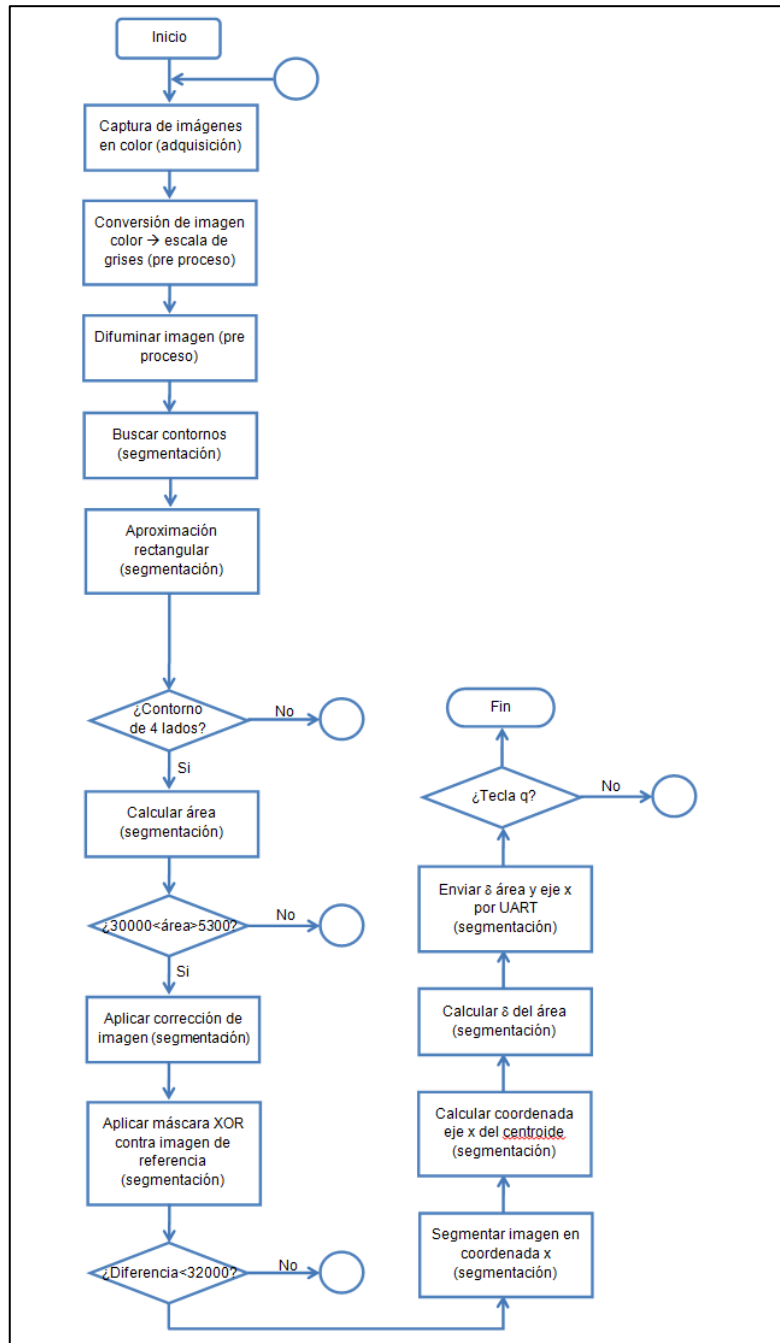


Figura 4.3 Diagrama de flujo del algoritmo de visión artificial.

### 4.2.1 Adquisición

Consiste en la captura de un sujeto portador de una imagen sobre cualquier prenda que cubra su torso, mediante una cámara web *Logitech C270®* conectada a la tarjeta *Raspberry Pi 2 model B®* a través del puerto *USB*. Utilizando funciones de las bibliotecas de *OpenCV®* en lenguaje *Python*, se habilita la cámara y da inicio al proceso de adquisición, como se muestra en el siguiente fragmento de código.

```
1. import cv2
2. import numpy as np
3. cap=cap.open()
4. while (True):
5.     ret,frame=cap.read()
6.     cv2.imshow("Original",frame)
7. cap.release()
```

Figura 4.4

Código para la adquisición de la imagen

1. La instrucción *import cv2* realiza la importación de instrucciones de la biblioteca de *OpenCV®* para su utilización en el algoritmo de lenguaje *Python*, nombrándola como *cv2*.
2. Se importa la librería de *Numpy*, el acelerador de operaciones matriciales para las operaciones algebraicas entre imágenes, la cual se nombra como *np*.
3. Se accede al driver de la cámara mediante *cap.open()* para empezar la captura de imágenes en forma de video almacenándose en la variable *cap*.
4. La estructura *while (True)* ejecuta un ciclo indefinido hasta que alguna condición dentro de esta provoca la salida.
5. Se leen las imágenes con la instrucción *cap.read()* la cual mediante la instrucción *ret (return)* se almacenan en la variable *frame*.
6. Mediante la instrucción *cv2.imshow()* se muestra en pantalla la imagen almacenada en la variable *frame*.
7. Al salir de la estructura *while*, la captura de imágenes se detiene mediante la instrucción *cap.release()*.

### 4.2.2 Pre proceso

Se realiza el tratamiento de la información que se almacenó en la variable *frame* de la etapa anterior para minimizar la carga al procesador, convirtiendo la imagen multi espectral de 3 bytes (*B,G,R*) en una imagen monocromática de 1 byte (escala de grises). Para reducir aún más la carga al procesador se realiza la binarización por umbral para transformar la imagen en escala de grises en una imagen binaria de 1 bit.

El tamaño de la imagen es:

$$frame_{size} = 640 \times 480 = 307200 \text{ pixel} \quad (2.5)$$

La imagen a color requiere de 3 bytes por píxel para su codificar su color en los canales R(0-255),G(0-255),B(0-255):

$$frame_{color} = 307200 \times 3 \text{ bytes} = 921600 \text{ bytes} \approx \boxed{.87 \text{ MB}} \quad (2.6)$$

```
1. gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
2. blur=cv2.GaussianBlur(gray,(3,3),0)
```

Figura 4.5  
Código de pre proceso de la imagen

1. Se realiza la transformación de color a escala de grises con la instrucción *cv2.cvtColor()*. La imagen en escala de gris requiere 1 byte por píxel para codificar su intensidad luminosa (0-255), por lo que el tamaño resultante es:

$$frame_{gris} = 307200 \times 1 \text{ byte} = 307200 \text{ bytes} \approx \boxed{300 \text{ Kb}} \quad (2.7)$$

2. Se filtra la imagen de escala de grises con la función *cv2.GaussianBlur()* para eliminar el ruido y suavizar la imagen. El objetivo del filtrado es para delimitar todos los contornos presentes en la imagen.

### 4.2.3 Segmentación

Se extraen las características del objeto que serán útiles para realizar el reconocimiento. Posteriormente se realiza el conteo de contornos, detección de un polígono, cálculo del área así, corrección de la proyección del polígono, aplicación de una máscara booleana y el conteo de píxeles para determinar si la imagen capturada del objetivo corresponde con la imagen de referencia.

```

1.  edges=cv2.Canny(blur,127,255)
2.  contours,hierarchy=cv2.findContours(edges,cv2.RETR_TREE,
    cv2.CHAIN_APPROX_SIMPLE)
3.  for cnt in contours:
4.      approx=cv2.approxPolyDP(cnt,0.01*cv2.arcLength(cnt,True),True)
5.      if len(approx)==4:
6.          area=cv2.contourArea(approx)
7.          if area>30000:
8.              warped=perspective(frame,approx.reshape(4,2))
9.              warped=cv2.resize(warped,(320,320))
10.             diffImg=cv2.bitwise_xor(warped,img_ref)
11.             diff=cv2.countNonZero(diffImg)

```

Figura 4.6  
Código de segmentación (extracción de características) de la imagen

1. Se dibujan todos los contornos con la función `cv2.Canny()` a la imagen en escala de grises obtenida anteriormente, guardándose en la variable `edges`.
2. Se realiza el reconocimiento de todos los contornos con la función `cv2.FindContours()` a la variable `edges`.
3. Con el bucle `for` se realiza el conteo de los contornos encontrados en la línea anterior, almacenándolos en la variable `cnt`.
4. Se realiza el reconocimiento de polígonos con la función `cv2.approxPolyDP()` de la variable `cnt`.
5. Mediante la estructura `if` se comprueba si algún polígono tiene cuatro lados. En caso de ser falso se no se realiza ninguna acción, provocando la salida del bucle `while()`.
6. Si la estructura `if` es verdadera, se realiza el cálculo del área en píxeles del polígono encontrado en la línea 4 mediante la función `cv2.contourArea()`, almacenándose en la variable `área`.
7. Con la estructura `if` se compara el área calculada con anterioridad. Si la sentencia es falsa, no se realiza ninguna acción provocando la salida del bucle `while()`.



8. Se implementa la función “*perspective*” para realizar la corrección del ángulo de proyección de la imagen.
9. Se reajustan las dimensiones de la corrección realizada mediante la instrucción *cv2.resize()*. El reajuste se realiza a razón tamaño de la imagen de referencia para que la máscara sea más precisa, almacenándose en la variable *warped*.
10. Se aplica una máscara *xor* entre la imagen corregida *warped* y la imagen de referencia *Img\_ref* para obtener una diferencia de píxeles entre las dos imágenes. El resultado se almacena en la variable *diffImg*.
11. Se realiza el conteo de los píxeles blancos en la máscara mediante la función *cv2.countNonZero()*. El resultado se guarda en la variable *diff* para su comparación.

```
12.   if diff<32000:  
13.       x,y,w,h=cv2.boundingRect(cnt)  
14.       mov=int((area-30000)/1300)*-1  
15.       cx=(x/27)
```

`port.write(str(a[mov]+b[cx]))` *Figura 2.7 Silla convencional*  
*Fuente: [https://es.made-in-china.com/co\\_carerehab/image\\_Children-Aluminium-Wheelchair-for-Cerebral-Palsy-People-Wheelchair\\_eouinenug\\_KNBtmPQdQvcs.html](https://es.made-in-china.com/co_carerehab/image_Children-Aluminium-Wheelchair-for-Cerebral-Palsy-People-Wheelchair_eouinenug_KNBtmPQdQvcs.html)*

*Figura 4.7*  
*Código del cálculo del área, del eje “x” y transmisión de datos*

12. Si el resultado del conteo de píxeles es menor a 32000, se ejecuta el código dentro de la sentencia *if*.
13. Mediante la función *cv2.boundingRect()* se realiza una aproximación poligonal de cuatro vértices que corresponden a la forma del objetivo, almacenándose en las variables *x,y,h,w*.
14. Se realiza la segmentación respecto al área del objetivo en 18 partes en el rango de  $0 \rightarrow 18$  para determinar el avance o retroceso, almacenándose en la variable *mov*.
15. Se realiza el cálculo y segmentación del desplazamiento horizontal del objetivo en función de la variable *x*, dentro del rango de  $0 \rightarrow 18$ , almacenándose en *cx*.
16. Mediante *port.write* se envía el número ( $0 \rightarrow 18$ ) referenciado a la matriz *a* y *b*, las cuales contienen una lista de caracteres correspondientes al código ASCII y corresponden a la magnitud del desplazamiento frontal y lateral del objetivo.

### 4.3 Algoritmo del control manual

En la tarjeta *FRDM-KL25Z*® se implementa un algoritmo de transmisión para el control manual, el cual basa su funcionamiento en la inclinación del acelerómetro integrado en la tarjeta así como en el módulo de radiofrecuencia *NRF24L01*. El algoritmo se realiza en lenguaje C a través del compilador en línea *MBED*, basando su funcionamiento en el diagrama de la siguiente figura.

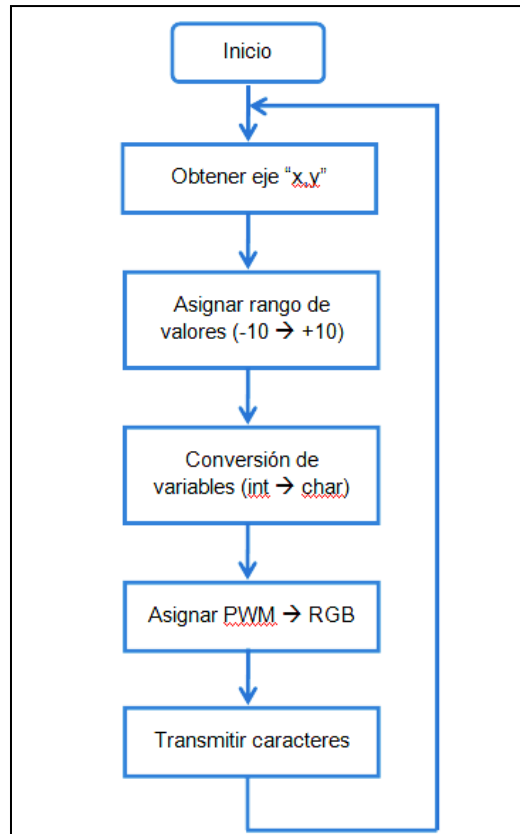


Figura 4.8  
Diagrama de flujo del control manual

En primera instancia, se realiza el mapeo de los pines de la tarjeta *FRDM-KL25Z*® que están habilitados para la comunicación con otros dispositivos mediante el protocolo de interfaz periférica serial, en inglés, *Serial Peripheral Interface (SPI)* y asignar correctamente estos pines en el algoritmo. A continuación se muestra un condensado de los pines asignados (tabla 4.1) para comunicación *SPI* basado en la hoja de datos del fabricante.

FRDM KL25Z Pin Name	Function Enable		LED	NRF24L01
PTA13	SPI0			CE
PTA5	PWM		LED R	
PTA4			LED G	
PTA12			LED B	
PTD0	SPI0	PCS0		IRQ
PTD1		SCKL		SCK
PTD2		MOSI		MOSI
PTD3		MISO		MISO
PTD5	SPI1	SCKL		CSN

Tabla 4.1 Asignación de funciones para control manual [39]

El desarrollo del código se muestra a continuación:

```

1. #include "mbed.h"
2. #include "MMA8451Q.h"
3. #include "nRF24L01P_PTX.h"

```

Figura 4.9  
Declaración de archivos de cabecera (header files)

Se incluyen los archivos de cabecera para poder realizar la compilación en plataforma *MBED®*, así como del acelerómetro (*MMA8451Q*) y el módulo de radiofrecuencia en la modalidad de transmisor. Con estas cabeceras se tiene acceso a los registros necesarios para el funcionamiento de estos dispositivos.

```

4. PinName const SDA = PTE25;
5. PinName const SCL = PTE24;

```

Figura 4.10  
Declaración de pines de comunicación de *I<sup>2</sup>C* (Inter-Integrated Circuit)

Se definen los pines del puerto E del microcontrolador con los que se establecerá la comunicación con el acelerómetro utilizando el protocolo *Inter-Integrated Circuit (I<sup>2</sup>C)*.

```

6.  MMA8451Q acc(SDA, SCL, MMA8451_I2C_ADDRESS);
7.  nRF24L01P Device(PTD2, PTD3, PTC5, PTD5); // MOSI, MISO, SCK, CSN
8.  nRF24L01P_PTX PTX(Device, PTA13, PTD0); // CE, IRQ
9.  PTX.Initialize();
10. PTX.SetDataRate(1000);
11. PTX.PowerUp();
12. PwmOut rled(LED1);
13. PwmOut gled(LED2);
14. PwmOut bled(LED3);

```

Figura 4.11  
Inicialización del acelerómetro, módulo de RF y pines de salida PWM

Dentro de la función *main()*, se accede a los registros del acelerómetro para establecer la comunicación con el micro controlador. Se definen también los pines de comunicación que utiliza el módulo de radiofrecuencia, el cual trabaja con el protocolo *SPI*. Posteriormente se define el ancho de banda y se inicializa el módulo. Además, se configura el *LED RGB* de la tarjeta ya que funcionará como indicador del direccionamiento propuesto.

```

15. float x, y;
16. int8_t x2, y2;
17.     x = abs(acc.getAccX());
18.     y = abs(acc.getAccY());
19.     x2 = acc.getAccX()*10;
20.     y2 = acc.getAccY()*(-10);
21. char a,b;
22.     a = (x2 + 48);
23.     b = (y2 + 106);

```

Figura 4.12  
Obtención de los datos del acelerómetro conversión a caracteres

En la función *while()* se implementa el código de funcionamiento manual de la tarjeta *FRDM-KL25Z®*. Se declaran las variables *x*, *y* para almacenar los valores de inclinación

que toma el acelerómetro y que se utilizan para variar la intensidad y color en el *LED RGB*.

Las variables *x2*, *y2* se utilizan para el control de los motores convirtiendo los valores de (-10 → +10) en caracteres del código *ASCII* almacenándose en las variables *a*, *b*. Se requiere la conversión de variable *int8\_t* a *char* por requerimiento de la instrucción para transmitir los datos.

```

24. PTX.TransmitPacket(&a,1);
25. PTX.TransmitPacket(&b,1);
    
```

Figura 4.13  
Transmisión de datos en forma de caracter

La transmisión de datos se ejecuta mediante la instrucción *PTX.TransmitPacket()* la cual se encuentra en el archivo cabecera *nRF24L01P\_PTX.h*. Las siguientes tablas muestran la correspondencia entre los valores de inclinación y los caracteres transmitidos.

Eje X (Avance-Reversa)		
Inclinación (int_8t)	Conversión (+48)	ASCII (char)
-10	38	&
-9	39	'
-8	40	(
-7	41	)
-6	42	*
-5	43	+
-4	44	,
-3	45	-
-2	46	.
-1	47	/
0	48	0
1	49	1
2	50	2
3	51	3
4	52	4
5	53	5
6	54	6
7	55	7
8	56	8
9	57	9
10	58	10

Tabla 4.2 Conversión de variable x

Eje Y (Izquierda-Derecha)		
Inclinación (int_8t)	Conversión (+106)	ASCII (char)
-10	96	'
-9	97	a
-8	98	b
-7	99	c
-6	100	d
-5	101	e
-4	102	f
-3	103	g
-2	104	h
-1	105	i
0	106	j
1	107	k
2	108	l
3	109	m
4	110	n
5	111	o
6	112	p
7	113	q
8	114	r
9	115	s
10	116	t

Tabla 4.3 Conversión de variable y

#### 4.4 Algoritmo del sistema de control de motores

Se implementa un algoritmo para la interpretación y manejo de los datos recibidos por la computadora *Raspberry Pi 2 model B*® del sistema de visión artificial y la tarjeta *FRDM-KL25Z*® del control manual a través del compilador *MBED*®. El objetivo de este algoritmo es emitir cuatro señales *PWM* para controlar la dirección de giro y velocidad de los motores de la silla de ruedas.

Se consulta la asignación de funciones en los pines con ayuda del manual del fabricante para establecer la funcionalidad en el algoritmo. La siguiente tabla es un condensado de las los pines del micro controlador y sus funciones específicas.

FRDM KL25Z Pin Name	Function Enable		Raspberry Pi 2	NRF24L01	PWM
PTA13	SPI0			CE	
PTB0 PTB1 PTB2 PTB3	PWM				PWM CH1 PWM CH2 PWM CH3 PWM CH4
PTC5	SPI0	SCKL		SCK	
PTD0 PTD2 PTD3 PTD5	SPI0	PCS0 MOSI MISO		IRQ MOSI MISO	
	SPI1	SCKL		CSN	
PTE 1	UART	RX	TX		

Tabla 4.4  
Asignación de pines para control de motores [39]

El diagrama de flujo que describe el comportamiento del algoritmo que controla la velocidad y sentido de rotación de los motores se muestra en la siguiente imagen:

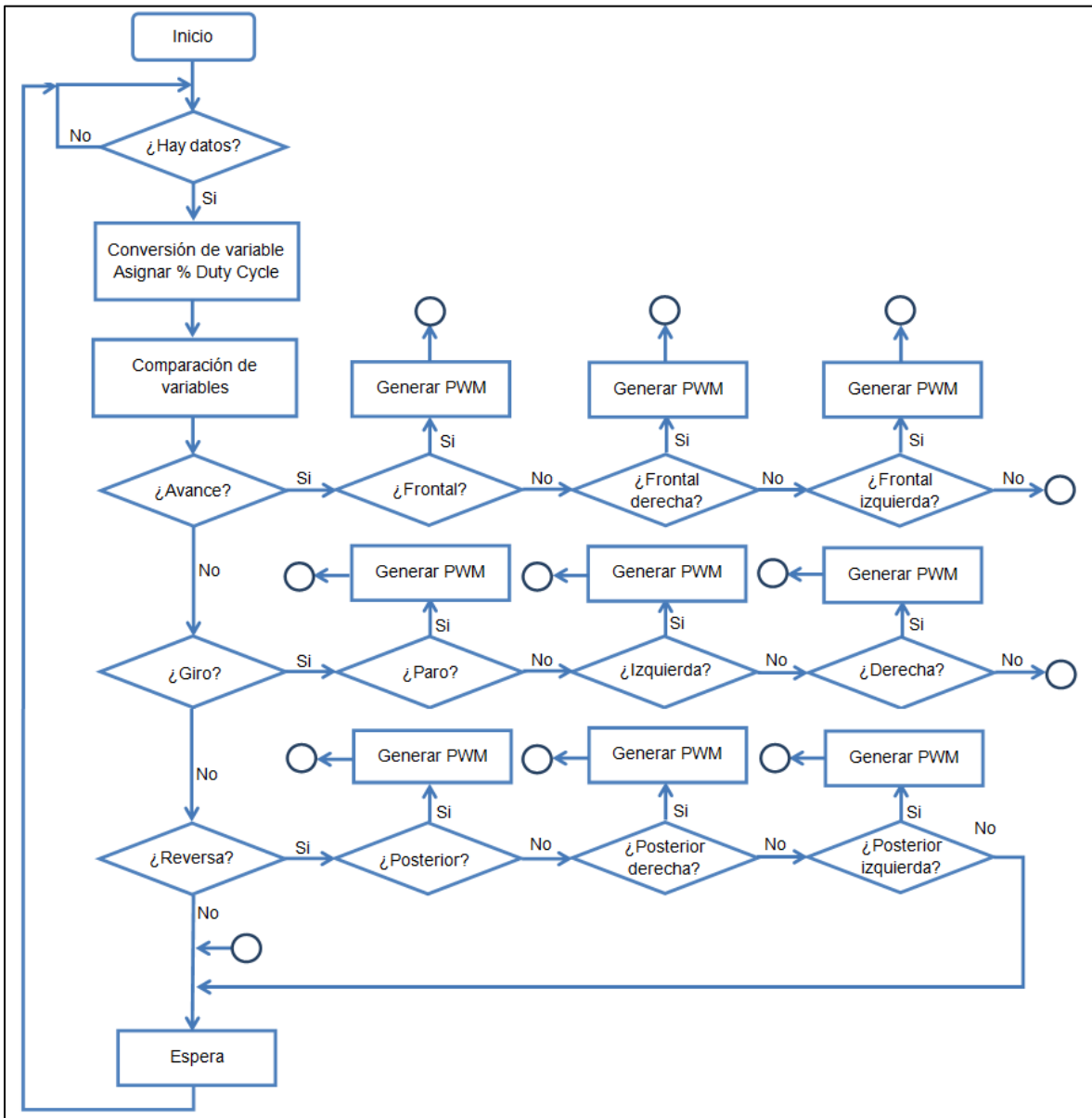


Figura 4.14  
Diagrama de flujo general del algoritmo para realizar el control de motores

Una vez elaborado y analizado del diagrama de flujo, se inicia la redacción de código mediante el cual se realiza el control de los motores.

```

1. #include "mbed.h"
2. #include "nRF24L01P_PRX.h"
    
```

Figura 4.15  
Declaración de archivos de cabecera

Se incluyen los archivos de cabecera del compilador *MBED*® y del módulo de radiofrecuencia para el acceso a los registros del módulo de radio y del micro controlador.

```

3. Serial RPi(NC,PTE1);    //(Tx,Rx)
4. nRF24L01P Receiver(PTD2, PTD3, PTC5, PTD5); // MOSI, MISO, SCK, CSN
5. nRF24L01P_PRX PRX(Receiver, PTA13, PTD0);    //CE, IRQ
6. PRX.Initialize();
7. PRX.SetDataRate(1000);
8. PRX.SetPayloadSize(1);
9. PRX.PowerUp();
10. PRX.StartReceive();
    
```

Figura 4.16  
Inicialización del puerto serie y del módulo de radiofrecuencia

Dentro del *main()*, se declaran los pines de transmisión y recepción de datos entre el micro controlador y el sistema de visión artificial y micro controlador con el módulo de radiofrecuencia. Con las instrucciones definidas en los archivos de cabecera que determinan el funcionamiento y la configuración (transmisor-receptor) del módulo de radiofrecuencia. Posteriormente se establece el tamaño del paquete de datos a 1 *byte*, ya que los caracteres transmitidos se encuentran en el rango entre 0 y 255. Se enciende el módulo y se habilita la recepción.

Se declara los pines [*PTB0, PTB1, PTB2, PTB3*] (figura 4.5) como pines de salida en modalidad *PWM* los cuales determinarán el sentido de giro y la velocidad en los motores.

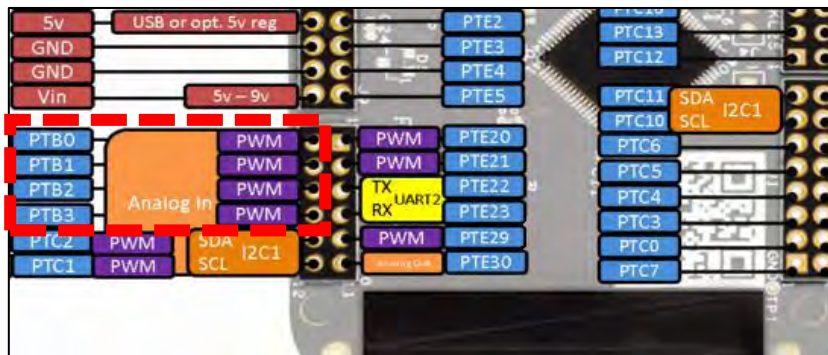


Figura 4.17  
Puerto B del microcontrolador y pines utilizados



Las características de las señales de modulación por ancho de pulso se definen como se muestra en la siguiente tabla:

Pin	Nombre	Frecuencia	Motor	Dirección
PTB0	CH1	10MHz	Izquierdo	Avance
PTB1	CH2	10MHz	Derecho	Avance
PTB2	CH3	10MHz	Izquierdo	Reversa
PTB3	CH4	10MHz	Derecho	Reversa

*Tabla 4.5  
Características de las señales PWM, pines de salida, asignación y sentido de giro en los motores*

Las líneas de código en la plataforma de programación quedan de la siguiente manera:

```

11. PwmOut CH1(PTB0);
12. PwmOut CH3(PTB2);
13. PwmOut CH2(PTB1);
14. PwmOut CH4(PTB3);

15. CH1.period(0.00010);
16. CH3.period(0.00010);
17. CH2.period(0.00010);
18. CH4.period(0.00010);

```

*Figura 4.18  
Declaración de los pines del puerto B y asignación de frecuencia de operación*

El fragmento de código de la figura 4.18 muestra cómo queda la declaración de los pines [PTB0-PTB3], los cuales se nombran como CH1, CH2, CH3 y CH4. Se establece la frecuencia de las señales de modulación por ancho de pulso a 10 MHz (10µs).

```

19. char c[3];
20. char x,y;
21. int val[2];
22. float a,b;

```

*Figura 4.19  
Declaración de variables globales*

Se realiza la declaración de variables globales en el *main()* de la función. La variable *char c[3]* es el buffer de dos caracteres donde se almacenarán los datos que provienen de la

tarjeta *Raspberry Pi 2 model B®* y el módulo de radiofrecuencia. Es una cadena de caracteres de tres *bytes* que posee la siguiente estructura:

```
char c['valorX', 'valorY', '\0']
```

Siendo:

$c[0] = \text{'valorX'}$

$c[1] = \text{'valorY'}$

$c[2] = \text{'\0'}$  o *caracter null*

Las variables  $x,y$  se utilizarán para almacenar independientemente los caracteres  $c[1]$  y  $c[0]$  respectivamente. La variable  $val [2]$  tiene una estructura semejante a  $char c[3]$ , a diferencia de que es una cadena de enteros en la cual después de la conversión de  $char \rightarrow int$  se almacenan los números de  $-10 \rightarrow +10$ . Las variables de tipo flotante  $a,b$  se utilizan para almacenar el *% Duty Cycle* para el control de los motores.

```
23.  if(RPi.readable()){
      //if(PRX.IsPacketReady())//
24.      RPi.gets(c,3);
25.      x=c[1];
26.      y=c[0];
27.      led=!led;
28.      val[0] = (x - 48);
29.      val[1] = (y - 106);
30.      a = abs(val[0] * 0.000010);
31.      b = abs(val[1] * 0.000010);
```

Figura 4.20

Condicional de la recepción de datos

Mediante la sentencia *if* se verifica si hay datos disponibles en el buffer de entrada y se almacenan en la variable  $c[3]$  mediante la instrucción *gets()*. Posteriormente se reubica el dato en las variables  $x,y$  y se realiza la conversión de carácter de texto a entero conforme a las tablas antes explicadas. Finalmente, en las variables  $a,b$  se guarda el valor absoluto del *%Duty Cycle* (estado alto) para regular la velocidad en los motores.

```

32.  if(val[0] > 0.0f) {
33.    if(val[1] == 0.0f) {
34.      CH1.pulsewidth(a);
35.      CH2.pulsewidth(a); }
36.    else {
37.      if(val[1] > 0.0f) {
38.        CH1.pulsewidth(a);
39.        CH2.pulsewidth(a - b);
40.        CH3.pulsewidth(0.0f);
41.        CH4.pulsewidth(0.0f);}
42.      else {
43.        if(val[1] < 0.0f) {
44.          CH1.pulsewidth(a - b);
45.          CH2.pulsewidth(a);
46.          CH3.pulsewidth(0.0f);
47.          CH4.pulsewidth(0.0f);}
48.        else {
49.          }
50.        }
51.      }
52.    }

```

Figura 4.21

Sentencia que ejecuta el avance de la silla de ruedas

Se accede a la condición de “avance” si la variable  $val[0]$  es mayor que 0. Posteriormente se evalúa si el avance es totalmente frontal ( $val[1] == 0$ ), si está cargado a la derecha ( $val[1] > 0$ ) o a la izquierda ( $val[1] < 0$ ). En caso de que el avance no sea frontal, la secuencia salta a la siguiente sentencia de comparación.

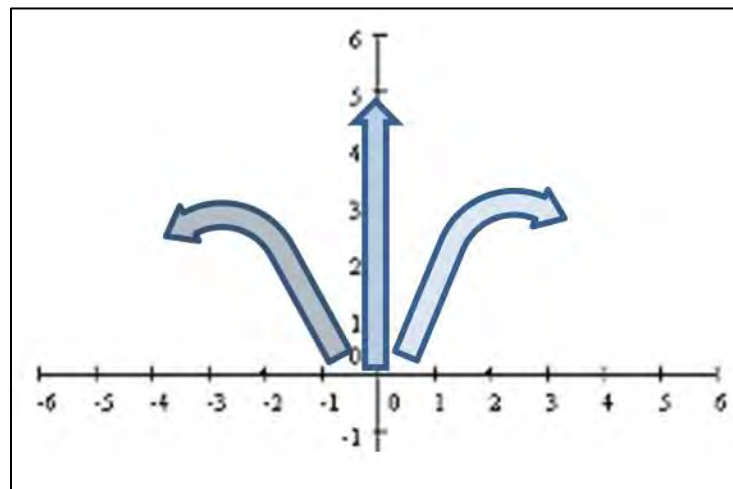


Figura 4.22

Direccionamiento de la condición “avance”

```

53.  if(val[0] == 0.0f) {
54.      if(val[1] == 0.0f) {
55.          CH1.pulsewidth(0.0f);
56.          CH2.pulsewidth(0.0f);
57.          CH3.pulsewidth(0.0f);
58.          CH4.pulsewidth(0.0f);}
59.      else {
60.          if(val[1] < 0.0f) {
61.              CH1.pulsewidth(0.0f);
62.              CH4.pulsewidth(0.0f);
63.              CH2.pulsewidth(b);
64.              CH3.pulsewidth(b);}
65.          else {
66.              if(val[1] > 0.0f) {
67.                  CH2.pulsewidth(0.0f);
68.                  CH3.pulsewidth(0.0f);
69.                  CH1.pulsewidth(b);
70.                  CH4.pulsewidth(b);}
71.              else {
72.                  }
73.              }
74.          }
75.      }

```

Figura 4.23  
Sentencia que ejecuta el paro, giro a la izquierda o a la derecha

La condición de paro y giro a la izquierda o derecha se ejecuta si la variable  $val[0] == 0$ . Cuando  $val[1] == 0$  la silla de ruedas permanece en reposo. Si  $val[1] < 0$  se ejecuta el giro hacia la izquierda y si  $val[1] > 0$  se ejecuta el giro a la derecha, como se muestra en la siguiente figura.

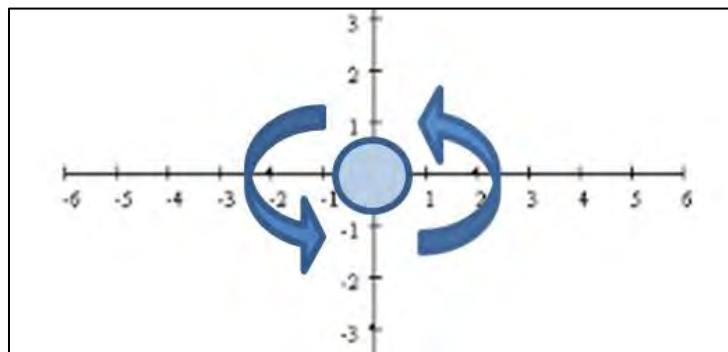


Figura 4.24 Direccionamiento de la condición “paro y giro”

```

76.  if(val[0] < 0.0f) {
77.    if(val[1] == 0.0f) {
78.      CH3.pulsewidth(a);
79.      CH4.pulsewidth(a);}
80.    else {
81.      if(val[1] < 0.0f) {
82.        CH3.pulsewidth(a - b);
83.        CH4.pulsewidth(a);
84.        CH1.pulsewidth(0.0f);
85.        CH2.pulsewidth(0.0f);}
86.      else {
87.        if(val[1] > 0.0f) {
88.          CH3.pulsewidth(a);
89.          CH4.pulsewidth(a - b);
90.          CH1.pulsewidth(0.0f);
91.          CH2.pulsewidth(0.0f);}
92.        else {
93.          }
94.        }
95.      }
96.    }

```

Figura 4.25  
Sentencia que ejecuta la reversa de la silla de ruedas

El direccionamiento de “reversa” (figura 4.6) se basa en una estructura de código similar al de avance. Se accede a esta rutina si la variable  $val[0] < 0$ , posteriormente se evalúa si la variable  $val[1] == 0$ , el direccionamiento es reversa en línea recta, si  $val[1] < 0$  el direccionamiento es reversa cargado a la izquierda y si  $val[1] > 0$  el movimiento es reversa cargado a la izquierda.

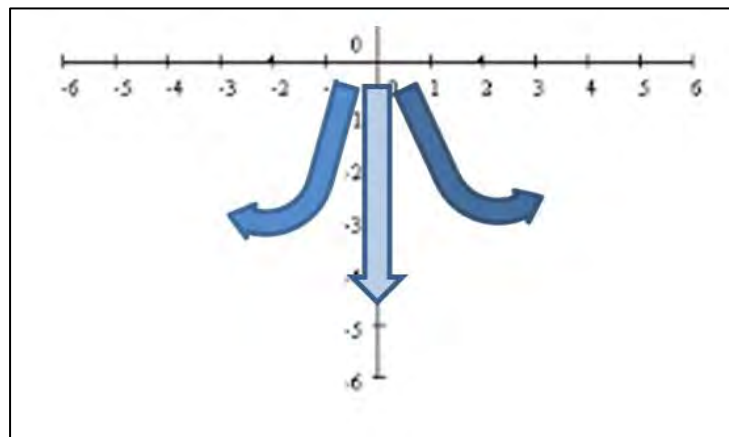


Figura 4.26  
Direccionamiento de la condición “reversa”

#### 4.4.1 Selección de los modos de operación

Para establecer los modos de funcionamiento (cámara-radiofrecuencia) en la silla de ruedas, se realiza la lectura del estado de pin PTD7 para ejecutar un modo u otro. Posteriormente el programa se re direcciona ciertas localidades en la memoria donde está almacenado el código, como se muestra en la siguiente figura:

```

1.  int main(){
2.      while(1) {
3.          if(NRF == 0){
4.              Radio();
5.          }
6.          else if(Raspi == 0){
7.              Camera();
8.          }
9.      }
10. }

```

*Figura 4.27*

*Sentencia que ejecuta la selección del modo de operación*

### 4.5 Diseño de hardware del control manual

El sistema de control manual está formado principalmente por una tarjeta *FRDM-KL25Z*®, un módulo de radiofrecuencia *NRF24L01*®, cuatro *LED* indicadores del direccionamiento y opto acopladores para aislar eléctricamente a la tarjeta *FRDM-KL25Z*® de los demás componentes.

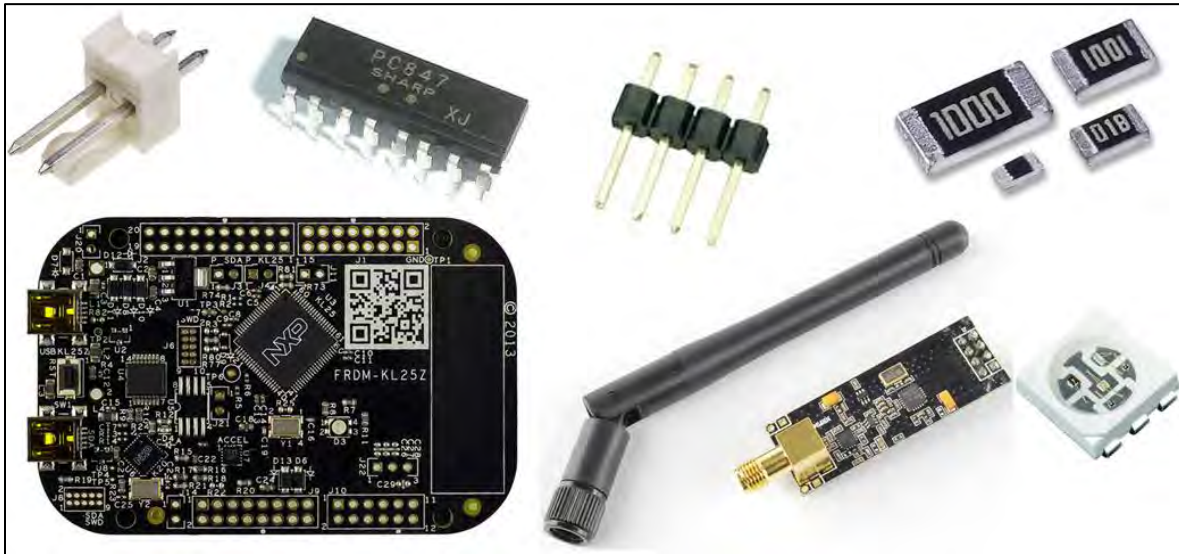


Figura 4.28  
Componentes del circuito de control manual

Las conexiones eléctricas entre la tarjeta *FRDM-KL25Z*®, el módulo de radiofrecuencia *NRF24L01*® y los *LED* indicadores del direccionamiento se determinan de acuerdo a la tabla de asignación de la implementación del software.

Tomando en cuenta la asignación de pines entre el módulo de radiofrecuencia y el microcontrolador del control manual, se diseña el diagrama esquemático con ayuda del software de diseño *Proteus*® v8.0, utilizando los componentes de la siguiente lista, los cuales se encuentran en el catálogo del distribuidor de componentes electrónicos *AG Electrónica*®:

Category	Quantity	References	Value
Resistors	3	R1-R3	330
Resistors	4	R4-R7	1.0K
Integrated Circuits	1	U1	PC847AB
Microcontroller	1	None	None
Miscellaneous	1	J2	CONN-SIL6
Miscellaneous	1	J4	CONN-SIL9
Miscellaneous	3	J5-J7	CONN-SIL8
Miscellaneous	4	LED1-LED4	LED_SMD 5050
Miscellaneous	2	NRF3.3,NRFGND	CONN-SIL4
Miscellaneous	1	POWER	CONN-H2
Miscellaneous	1	RESET	JUMPER

Tabla 4.6  
Componentes del circuito de control manual

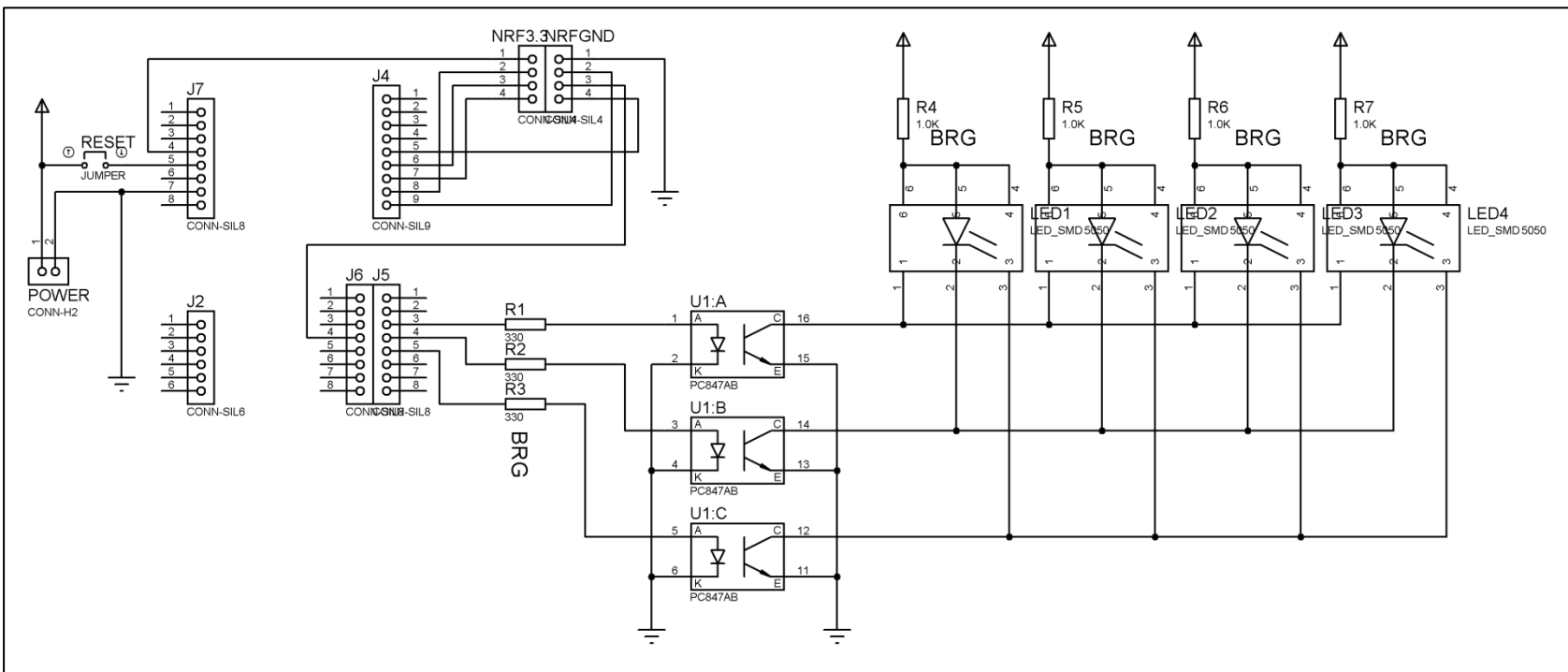


Figura 4.29  
Diagrama esquemático del control manual



## 4.6 Diseño de hardware del control de motores

El circuito de direccionamiento de la silla está conformado por una tarjeta *FRDM-KL25Z*®, un módulo receptor de radiofrecuencia *NRF24L01*®, dos convertidores *CD-CD* para excitar los transistores *MOSFET* que conforman los dos puente H y opto acopladores para aislar la etapa de control y potencia debido a que los voltajes y corrientes necesarios para manipular el sentido de giro y velocidad de los motores pueden causar daños irreversibles en los dispositivos digitales si existen conexiones eléctricas directas entre estos componentes.

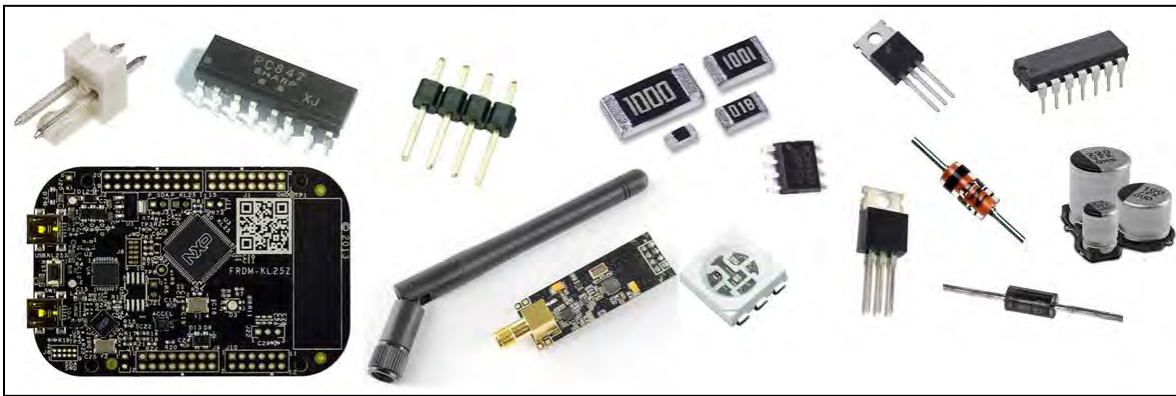


Figura 4.30  
Componentes del circuito controlador de motores

### 4.6.1 Criterio de selección de los elementos de potencia

Los dispositivos digitales de este trabajo son importantes para determinar el tipo de comportamiento del sistema en base a los parámetros de entrada. Para realizar una acción determinada en función de los parámetros de entrada, es necesario que el sistema cuente con dispositivos de control de potencia y actuadores de salida, como lo son el puente H y los motores de corriente directa.

Para el caso de los transistores del puente H, es de suma importancia utilizar dispositivos con características eléctricas que sean compatibles con las del elemento de salida, ya que si no se consultan en las hojas de datos, el sistema de control puede llegar a ser disfuncional o presentar daños permanentes en sus elementos al momento de ponerlo en funcionamiento.

Los motores típicamente utilizados para aplicación en sillas de ruedas eléctricas. son los de motores de corriente directa de cuatro polos e imán permanente con reductor.

Consultando la variedad de fabricantes de motores eléctricos, se encuentra que *ElectroCraft*® es uno de los fabricantes líder en motores de corriente directa con aplicación en sillas de ruedas. El motor consultado que corresponde con la aplicación es el modelo *MP36*® del cual es importante consultar dos parámetros eléctricos y en base a ellos, elegir el transistor adecuado para el puente H; el voltaje de operación y la corriente a plena carga del motor. En la siguiente tabla se muestran las especificaciones del fabricante de este modelo:

<b>MP36 Motor Parameters</b>								
Gear Ratio	32.00:1	29.25:1	26.82:1	24.67:1	21.33:1	19.50:1	17.88:1	16.44:1
S2 Torque (15 min) (Nm)	38.6	35.2	32.3	29.7	25.7	23.5	21.6	19.8
Torque Constant (Nm/A)	0.05918	0.05918	0.05918	0.05918	0.05918	0.05918	0.05918	0.05918
Max Torque (Nm)	108.5	99.2	90.9	83.6	72.3	66.1	60.6	57.7
Operational Voltage (Volts DC)	24	24	24	24	24	24	24	24
Nominal Speed at 24 VDC (RPM)	115	126	138	130	173	179	200	228
Back EMF Constant (V/KRPM)	6.2	6.2	6.2	6.2	6.2	6.2	6.2	6.2
Motor Mass (kg)	6.26	6.26	6.26	6.26	6.26	6.26	6.26	6.26
Motor Inertia (Armature) (kg.mm2)	1386.10	1386.10	1386.10	1386.10	1386.10	1386.10	1386.10	1386.10
Motor Winding Resistance (Ohms)	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3
Motor Winding Inductance (µH)	79	79	79	79	79	79	79	79
Motor Max Winding Temp (°C)	155	155	155	155	155	155	155	155
Motor Poles	4 Poles							
Max Motor Current (A)	80							
S2 motor power (15 min) (W)	425							
Max Motor Power (Watts)	850							

*Continuous rating based on nominal operating voltage 24 VDC.*

Tabla 4.7  
Especificaciones del fabricante del motor *MP36*®

De la tabla anterior, se observa estas características eléctricas importantes:

- i. Fabricación en 4 polos
- ii. Voltaje de operación de 24 V<sub>CD</sub>
- iii. Corriente máxima de trabajo de 80 A
- iv. Temperatura máxima de operación de 155 °C
- v. Potencia máxima de 850 W

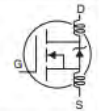
Tomando en cuenta las características eléctricas del motor, se encuentra en el catálogo del distribuidor de componentes electrónicos *AG Electrónica*®, que el transistor MOSFET más adecuado para trabajar con los parámetros eléctricos del motor *MP36*® y añadiendo el 20% de tolerancia en la capacidad de corriente, es el transistor con número de parte *IRF3205PBF* del fabricante *International Rectifier*®. La siguiente tabla muestra los parámetros máximos eléctricos de operación de este transistor.

### Absolute Maximum Ratings

	Parameter	Max.	Units
$I_D @ T_C = 25^\circ\text{C}$	Continuous Drain Current, $V_{GS} @ 10\text{V}$	110 <sup>⑤</sup>	A
$I_D @ T_C = 100^\circ\text{C}$	Continuous Drain Current, $V_{GS} @ 10\text{V}$	80	
$I_{DM}$	Pulsed Drain Current <sup>①</sup>	390	
$P_D @ T_C = 25^\circ\text{C}$	Power Dissipation	200	W
	Linear Derating Factor	1.3	W/ $^\circ\text{C}$
$V_{GS}$	Gate-to-Source Voltage	$\pm 20$	V
$I_{AR}$	Avalanche Current <sup>②</sup>	62	A
$E_{AR}$	Repetitive Avalanche Energy <sup>①</sup>	20	mJ
dv/dt	Peak Diode Recovery dv/dt <sup>③</sup>	5.0	V/ns
$T_J$	Operating Junction and	-55 to +175	$^\circ\text{C}$
$T_{STG}$	Storage Temperature Range		
	Soldering Temperature, for 10 seconds	300 (1.6mm from case )	
	Mounting torque, 6-32 or M3 srew	10 lbf•in (1.1N•m)	

### Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	55	—	—	V	$V_{GS} = 0\text{V}, I_D = 250\mu\text{A}$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.057	—	V/ $^\circ\text{C}$	Reference to $25^\circ\text{C}, I_D = 1\text{mA}$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	—	8.0	m $\Omega$	$V_{GS} = 10\text{V}, I_D = 62\text{A}$ <sup>④</sup>
$V_{GS(th)}$	Gate Threshold Voltage	2.0	—	4.0	V	$V_{DS} = V_{GS}, I_D = 250\mu\text{A}$
$g_{fs}$	Forward Transconductance	44	—	—	S	$V_{DS} = 25\text{V}, I_D = 62\text{A}$ <sup>④</sup>
$I_{DSS}$	Drain-to-Source Leakage Current	—	—	25	$\mu\text{A}$	$V_{DS} = 55\text{V}, V_{GS} = 0\text{V}$ $V_{DS} = 44\text{V}, V_{GS} = 0\text{V}, T_J = 150^\circ\text{C}$
$I_{GSS}$	Gate-to-Source Forward Leakage	—	—	100	nA	$V_{GS} = 20\text{V}$
	Gate-to-Source Reverse Leakage	—	—	-100	nA	$V_{GS} = -20\text{V}$
$Q_g$	Total Gate Charge	—	—	146	nC	$I_D = 62\text{A}$
$Q_{gs}$	Gate-to-Source Charge	—	—	35	nC	$V_{DS} = 44\text{V}$
$Q_{gd}$	Gate-to-Drain ("Miller") Charge	—	—	54	nC	$V_{GS} = 10\text{V}$ , See Fig. 6 and 13
$t_{d(on)}$	Turn-On Delay Time	—	14	—	ns	$V_{DD} = 28\text{V}$ $I_D = 62\text{A}$ $R_G = 4.5\Omega$ $V_{GS} = 10\text{V}$ , See Fig. 10 <sup>④</sup>
$t_r$	Rise Time	—	101	—		
$t_{d(off)}$	Turn-Off Delay Time	—	50	—		
$t_f$	Fall Time	—	65	—		
$L_D$	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
$L_S$	Internal Source Inductance	—	7.5	—		
$C_{iss}$	Input Capacitance	—	3247	—	pF	$V_{GS} = 0\text{V}$ $V_{DS} = 25\text{V}$ $f = 1.0\text{MHz}$ , See Fig. 5
$C_{oss}$	Output Capacitance	—	781	—		
$C_{rss}$	Reverse Transfer Capacitance	—	211	—		
$E_{AS}$	Single Pulse Avalanche Energy <sup>②</sup>	—	1050 <sup>⑥</sup>	264 <sup>⑦</sup>		



**Tabla 4.8**  
**Parámetros eléctricos del transistor IRF3205PBF**

De la tabla anterior, se observa estas características eléctricas importantes:

- i. Corriente continua máxima en *Drain* de 80 A bajo condiciones específicas de prueba
- ii. Voltaje *Drain-Source* hasta 55 V bajo condiciones específicas de prueba
- iii. Resistencia *Drain-Source* de 8 mΩ bajo condiciones específicas de prueba
- iv. Disipación de potencia del encapsulado de 200 W
- v. Umbral de voltaje de 2V-4V en *Gate-Source* bajo condiciones específicas de prueba
- vi. Tiempo de los flancos de subida y bajada <101 ns

Otro elemento importante de la sección de potencia es el control de los transistores *MOSFET*, ya que para accionarlos de forma correcta es necesario considerar un elemento que los haga trabajar dentro de los parámetros recomendados del fabricante y no llevarlos a las zonas máximas de trabajo. El elemento requerido es un manejador de puerta, en inglés, *Gate Drive*.

Básicamente el funcionamiento de este elemento es adecuar el voltaje que se tiene en el *Gate* del transistor para trabajar dentro del umbral entre *Gate-Source* ( $V_{GS(th)}=2V-4V$ ). El no respetar el valor recomendado para  $V_{GS(th)}$  puede ocasionar daño físico en el transistor por sobrecalentamiento.

En base a la asignación de pines del micro controlador, se elabora el diagrama esquemático que comprende las conexiones entre la tarjeta de desarrollo *FRDM-KL25Z*®, el módulo de radiofrecuencia *NRF24L01*®, la micro computadora *Raspberry Pi model B*®, los conectores de baterías, convertidores *DC-DC*, los *gate drive* de los transistores *MOSFET* de los puente H y los conectores de motores, con ayuda del software de diseño *Proteus*® v8.0. Se genera el *Bill Of Materials* en el programa de diseño para obtener los tipos de componentes, cantidad, nombre de referencia en el circuito impreso y valor del componente en el caso de los componentes pasivos. En la siguiente tabla se muestran los componentes utilizados:

Category	Quantity	References	Value
Capacitors	1	C1	1u
Capacitors	1	C2	.1u
Capacitors	7	C3,C5,C10-C14	0.1u
Capacitors	2	C4,C6	0.47u
Capacitors	1	C7	1000u
Capacitors	2	C8,C15	470u
Capacitors	1	C9	1.0u
Resistors	1	R	CONN-SIL1
Resistors	8	R1-R4,R6-R8,R23	10
Resistors	2	R5,R24	0R1
Resistors	8	R9-R16	180
Resistors	2	R17-R18	1.0K
Resistors	1	R19	330
Resistors	1	R20	100K
Resistors	2	R21-R22	10K
Integrated Circuits	4	U1-U2,U6,U10	IR2101
Integrated Circuits	1	U3	74126
Integrated Circuits	1	U4	PC847AB
Integrated Circuits	1	U5	74125
Integrated Circuits	1	U7	NE556
Integrated Circuits	1	U8	74LS00
Transistors	8	Q1-Q8	IRFZ44N
Transistors	1	Q9	2N2222
Diodes	1	D1	1N4001
Diodes	8	D2-D3,D5-D10	20BQ030TR_1
Diodes	4	D4,D11-D13	BAT46W
Diodes	8	D14-D21	1N4148WS-7_1
Miscellaneous	14	5V,+,-,BAT+,BAT-,BATT,GND,J6,L,L+,R+,VGATE,VIN+,VIN-	CONN-SIL1
Miscellaneous	2	J1,J3	CONN-SIL8
Miscellaneous	1	J4	CONN-SIL6
Miscellaneous	1	J5	CONN-SIL10
Miscellaneous	3	NRF(3.3),NRF(GND),RPI	CONN-SIL4
Miscellaneous	2	PTD6-PTD7,SELECTOR	CONN-SIL2
Miscellaneous	1	RL1	G6B-2214P-US-DC-5

*Tabla 4.9*  
*Componentes utilizados en el circuito de control de motores*

El diagrama esquemático del circuito se divide en varias partes para facilitar la elaboración e interpretación, como se muestra a continuación:

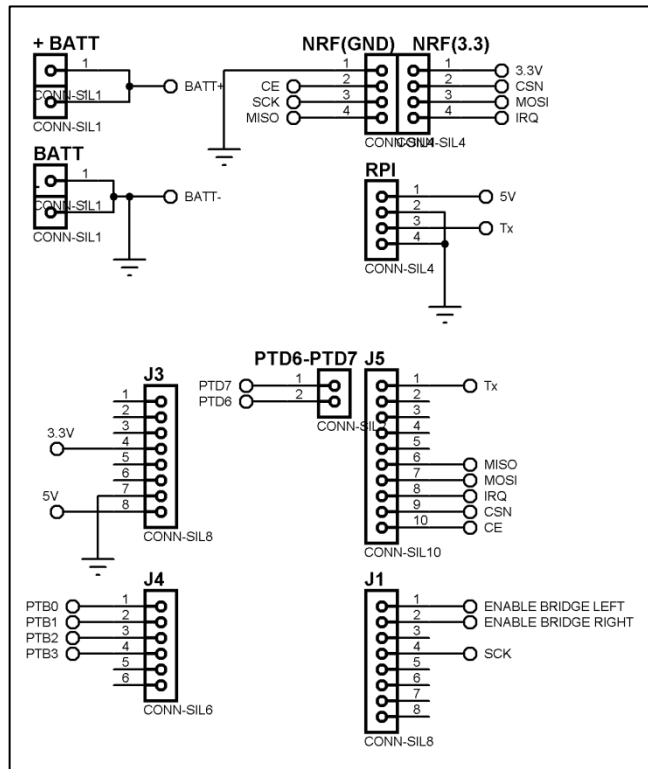


Figura 4.31  
Diagrama esquemático del control manual. Conectores

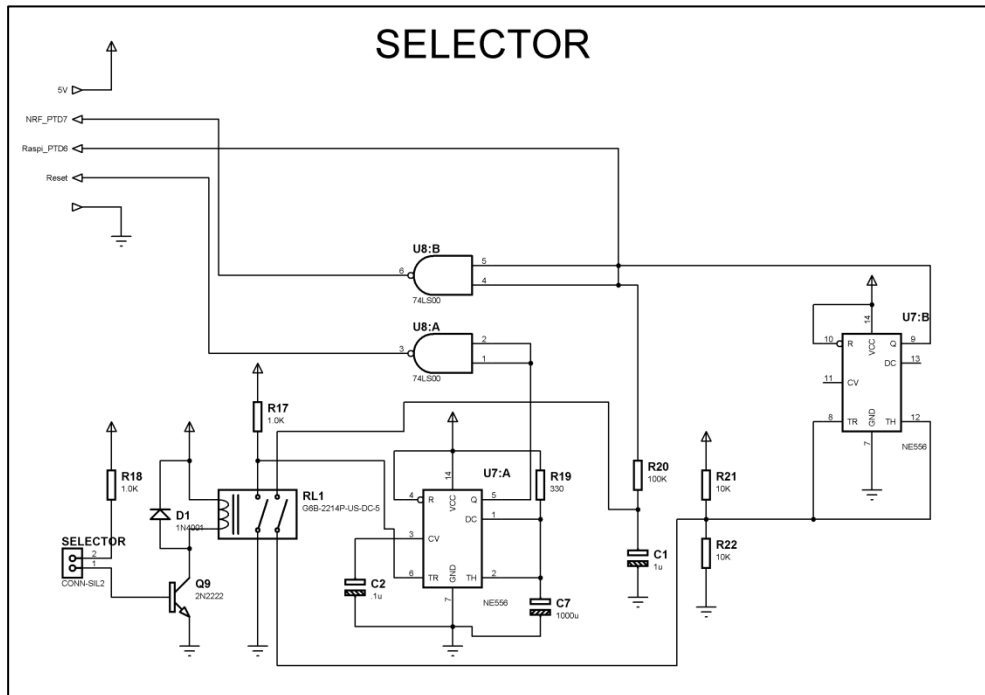


Figura 4.32  
Diagrama esquemático del circuito de selección

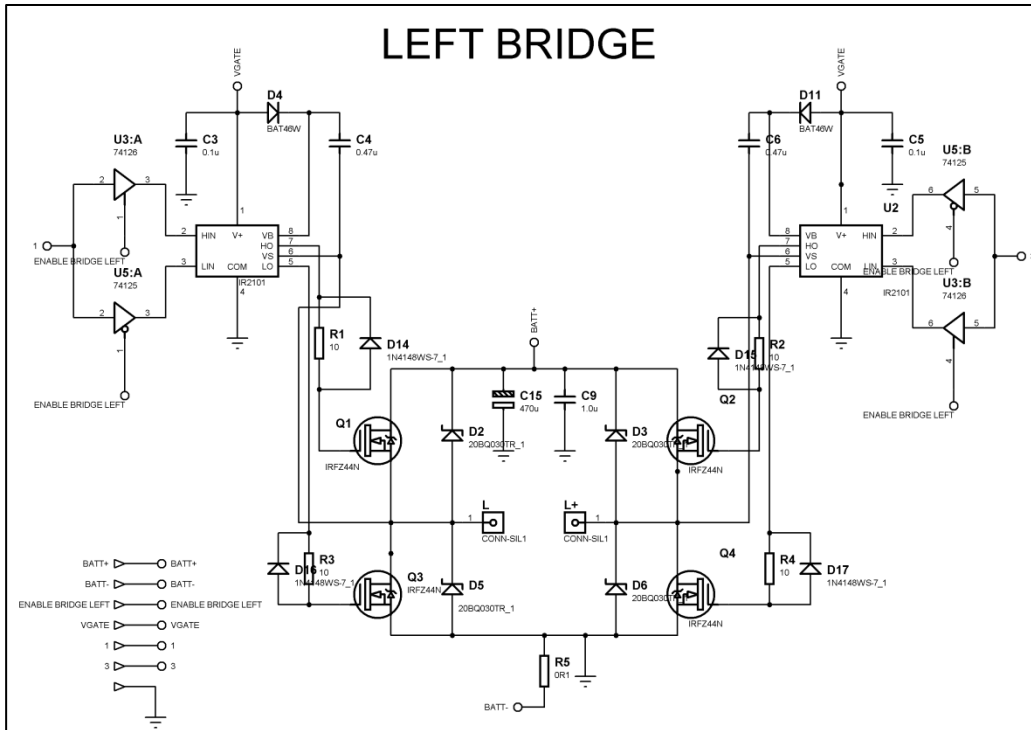


Figura 4.33  
Diagrama esquemático del puente H izquierdo

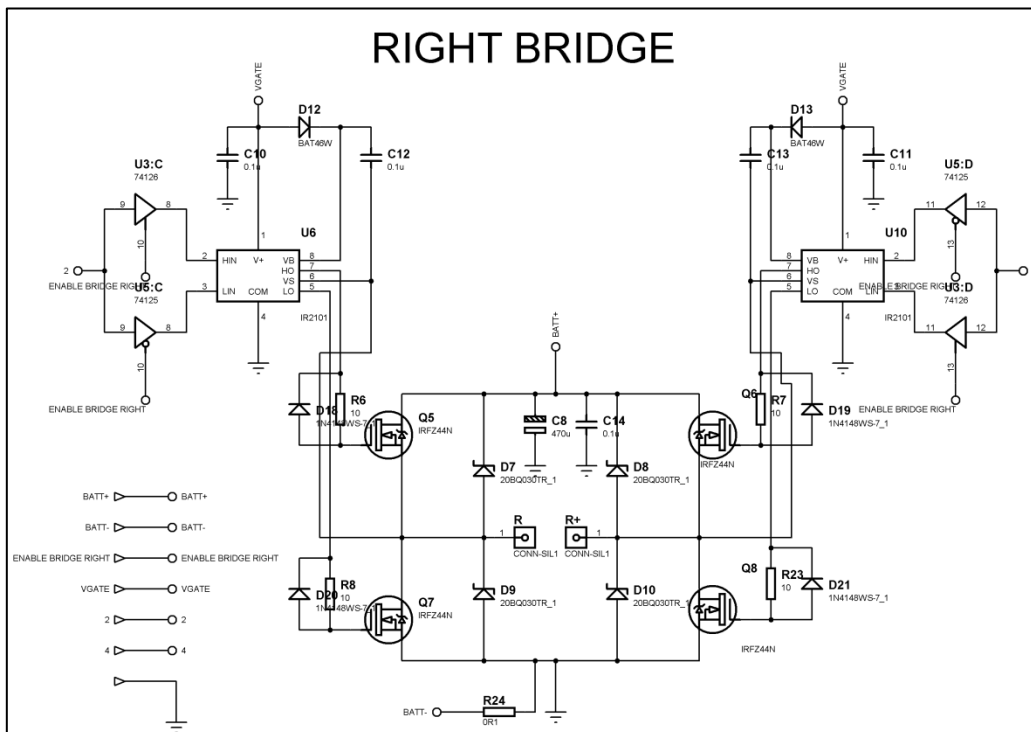


Figura 4.34  
Diagrama esquemático del puente H izquierdo

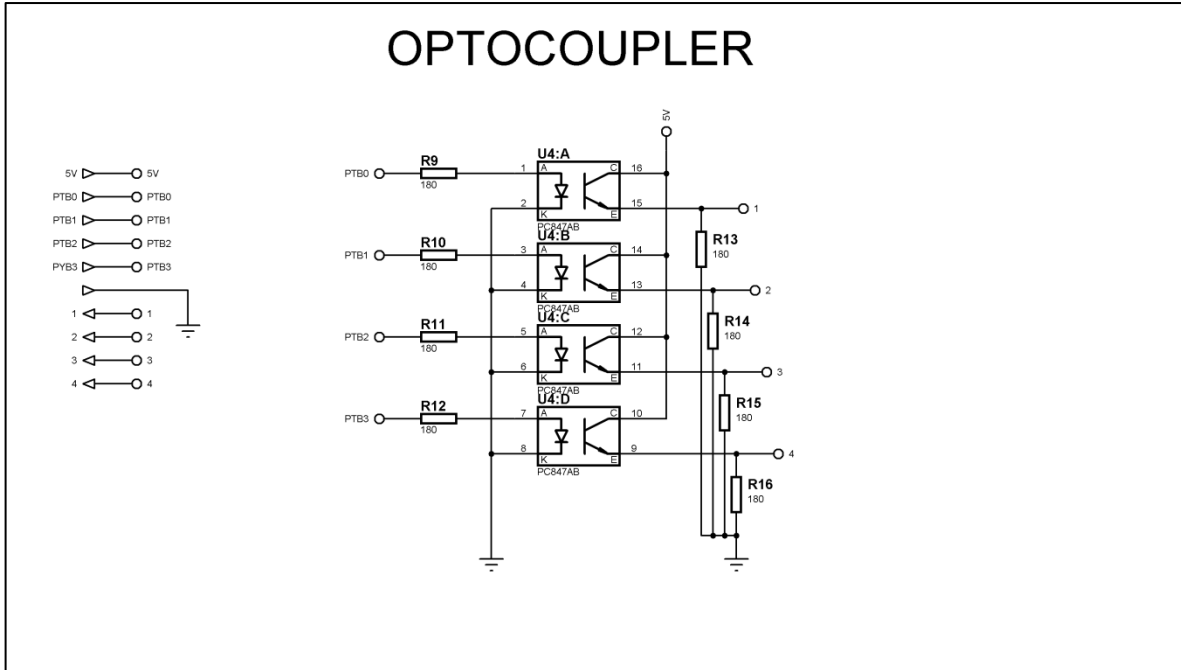


Figura 4.34  
Diagrama esquemático del puente H izquierdo

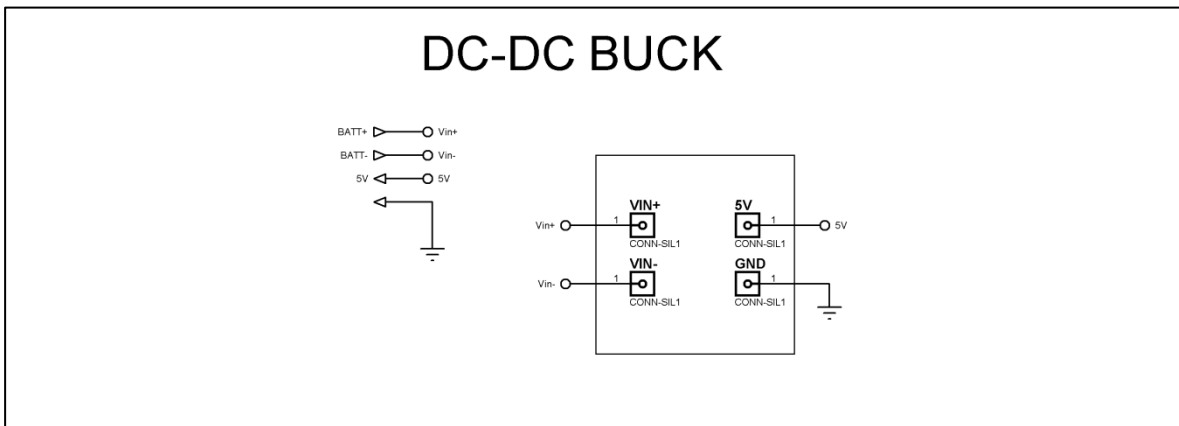


Figura 4.35  
Diagrama esquemático del reductor de voltaje para el Gate Drive



### 4.7 Análisis de costos

A partir de la elaboración de los diagramas esquemáticos del modulo de control manual y el módulo controlador de motores en el software *Proteus® v8*, se puede generar el listado de costos de los componentes utilizados en cada circuito. Este listado comprende la cantidad de elementos utilizados, el valor en el caso de componentes pasivos (ohms, faradios o henrios), circuitos integrados y módulos (para el caso de las tarjetas de desarrollo y micro computadora).

Es importante considerar los recursos destinados ya que en la comparativa con las sillas de ruedas eléctricas comerciales, el costo de este proyecto debe ser menor que el de una silla comercial para que la posibilidad de adquisición aumente sobre todo en las familias de mediano y bajos recursos.

En la tabla 4.10 se organizan los elementos del sistema de control manual y los detalles que se mencionan con anterioridad:

Bill Of Materials						Totals		
Design Title		Transmitter						
Author		Diego Reyes Quijada						
Total Parts In Design		26						
Category	Quantity	References	Value	AG Electrónica Stock Code	Unit Cost	Category	Quantity	Net Cost
Resistors	3	R1-R3	330	RMS-330E	\$2.59	Modules	2	\$504.00
Resistors	4	R4-R7	1.0K	RMS-1K/2512	\$2.59	Capacitors	0	\$0.00
Integrated Circuits	1	U1	PC847AB	PC847X	\$17.24	Resistors	7	\$18.13
Modules	1			FRDM-KL25Z	\$384.48	Integrated Circuits	1	\$0.00
Modules	1			NRF24L01	\$50.00	Transistors	0	\$0.00
Miscellaneous	1	J2	CONN-SIL6	HEADER-1	\$6.03	Diodes	0	\$0.00
Miscellaneous	1	J4	CONN-SIL9	HEADER-1	\$6.03	Miscellaneous	16	\$220.95
Miscellaneous	3	J5-J7	CONN-SIL8	HEADER-1	\$6.03	<b>Total</b>	<b>26</b>	<b>\$743.07</b>
Miscellaneous	4	LED1-LED4	LED_SMD 5050	LED-RGB5050-AC-F120/C	\$4.31			
Miscellaneous	2	NRF3.3,NRFGND	CONN-SIL4	HEADER-1	\$6.03			
Miscellaneous	1	POWER	CONN-H2	HEADER-1	\$6.03			
Miscellaneous	1	RESET	JUMPER	HEADER-1	\$6.03			
Miscellaneous	1			ALCOHOL-ISOPROPILICO	\$62.07			
Miscellaneous	1			MC-050	\$38.79			
Miscellaneous	1			FR-10X10	\$18.10			

Tabla 4.10  
Desglose de costos del circuito de control manual

Los elementos del sistema controlador de motores se enlistan en la tabla 4.11. Los componentes electrónicos se cotizan en el catálogo de *AG Electrónica®*, a excepción de la cámara WEB, la cual se adquirió en la tienda *OfficeMax®*.

Design Title Receiver						Totals		
Author Diego Reyes Quijada						Category	Quantity	Net Cost
Total Parts In Design 106						Category	Quantity	Net Cost
Category	Quantity	References	Value	Stock Code	Unit Cost	Category	Quantity	Net Cost
Capacitors	1	C1	1u	CE-1/100V-TEAPO	\$4.31	Modules	3	\$1,554.99
Capacitors	1	C2	.1u	CE-.1/100V-TEAPO	\$4.31	Capacitors	15	\$104.01
Capacitors	7	C3,C5,C10-C14	0.1u	CEMS-.1/50V	\$4.31	Resistors	24	\$57.98
Capacitors	2	C4,C6	0.47u	CEMS-.47/50V	\$4.31	Integrated Circuits	9	\$301.03
Capacitors	1	C7	1000u	CE-1000/16V-TEAPO	\$8.62	Transistors	9	\$254.98
Capacitors	2	C8,C15	470u	CEMS-470/16V	\$14.66	Diodes	21	\$117.09
Capacitors	1	C9	1.0u	CEMS-1/50V	\$4.31	Miscellaneous	25	\$2,774.87
Resistors	8	R1-R4,R6-R8,R2	10	RMS-10E/2512	\$2.59	<b>Total</b>	<b>106</b>	<b>\$5,164.95</b>
Resistors	2	R5,R24	0R1	RMS-1E/2512	\$2.59			
Resistors	8	R9-R16	180	RMS-180E	\$1.72			
Resistors	2	R17-R18	1.0K	RMS-1K	\$1.72			
Resistors	1	R19	330	RMS-330E	\$1.72			
Resistors	1	R20	100K	RMS-100K	\$1.72			
Resistors	2	R21-R22	10K	RMS-10K	\$1.72			
Integrated Circuits	4	U1-U2,U6,U10	IR2101	IR2101PBF	\$43.97			
Integrated Circuits	1	U3	74126	DM74LS126AN	\$23.28			
Integrated Circuits	1	U4	PC847AB	PC847X	\$17.24			
Integrated Circuits	1	U5	74125	HD74LS125AP	\$20.69			
Integrated Circuits	1	U7	NE556	NE556N	\$10.35			
Integrated Circuits	1	U8	74LS00	HD74LS00P	\$12.07			
Transistors	8	Q1-Q8		IRF3205PBF	\$25.86			
Transistors	1	Q9	2N2222	2N2222	\$12.93			
Diodes	1	D1	1N4001	1N4001-M1	\$0.86			
Diodes	8	D2-D3,D5-D10		1N5819W/F	\$2.59			
Diodes	4	D4,D11-D13	BAT46W	10BF10	\$14.66			
Diodes	8	D14-D21	1N4148WS-7_1	RS1MW/F	\$2.59			
Modules	1			RASPBERRYPI-2-MODB-1C	\$906.03			
Modules	1			FRDM-KL25Z	\$384.48			
Modules	1			MD0040	\$50.00			
Miscellaneous	1			LOGITECH C270	\$463.79			
Miscellaneous	1			NP24-12	\$1,622.41			
Miscellaneous	2		DC Motor	RB-04M022	\$83.62			
Miscellaneous	14	TT,GND,J6,L,L+,	CONN-SIL1	HEADER-1	\$6.03			
Miscellaneous	2	J1,J3	CONN-SIL8	HEADER-1	\$6.03			
Miscellaneous	1	J4	CONN-SIL6	HEADER-1	\$6.03			
Miscellaneous	1	J5	CONN-SIL10	HEADER-1	\$6.03			
Miscellaneous	3	(3.3),NRF(GND)	CONN-SIL4	HEADER-1	\$6.03			
Miscellaneous	2	J6-PTD7,SELECT	CONN-SIL2	HEADER-1	\$6.03			

**Tabla 4.11**  
*Desglose de costos del circuito de control de motores*

Los motores ocupados en este circuito son motores de corriente directa con reductor utilizados principalmente en la elaboración de prototipos debido a la dificultad de conseguir piezas similares a los del modelo *MP36®*, ideal para esta aplicación. Adquiriendo los modelos de motor adecuados, el costo total puede aumentar hasta \$4000 para el caso de motores usados. Se cotiza también una batería de ácido plomo de 12V/24Ah para alimentar al sistema y que es ideal para la aplicación de vehículos eléctricos al ser libre de mantenimiento.

De las tablas anteriores se obtiene el costo total aproximado de la elaboración del proyecto. El cual se resume en la siguiente:

Category	Quantity	Net Cost	Category	Quantity	Net Cost
Modules	2	\$504.00	Modules	3	\$1,554.99
Capacitors	0	\$0.00	Capacitors	15	\$104.01
Resistors	7	\$21.03	Resistors	24	\$57.98
Integrated Circuits	1	\$20.00	Integrated Circuits	9	\$301.03
Transistors	0	\$0.00	Transistors	9	\$254.98
Diodes	0	\$0.00	Diodes	21	\$117.09
Miscellaneous	16	\$220.95	Miscellaneous	25	\$354.88
<b>Total</b>	<b>26</b>	<b>\$765.97</b>	<b>Total</b>	<b>106</b>	<b>\$5,164.95</b>
<b>GRAN TOTAL</b>					<b>\$5,930.92</b>

*Tabla 4.12  
Resumen de gastos del proyecto*

En conclusión, los recursos destinados para la elaboración de este proyecto es mucho menor a los que se podrían destinar en la compra de una silla de ruedas comercial cuyo precio oscila entre los \$26,599 pesos [41] hasta los \$69,900 pesos [42], teniendo como principales ventajas:

- I. Precio por debajo del rango comercial
- II. Maniobrabilidad sin esfuerzo
- III. Adaptación a cualquier silla de ruedas mecánica
- IV. Configuración mediante software de usuarios que controlan el direccionamiento

# Capítulo 5

## Pruebas del sistema y resultados

### 5.1 Sistema de visión artificial

La adquisición, el pre procesamiento, segmentación y reconocimiento de una imagen, se realiza mediante una cámara *Logitech® C270* con una conexión *USB* hacia la tarjeta *Raspberry Pi 2 model B®*. Se hace una configuración en la tarjeta *Raspberry Pi 2 model B®* para trabajar en modalidad “*Desktop*” a través de un monitor genérico. Al ejecutarse el algoritmo de *Python*, se visualizan las imágenes resultado de los procesos mediante la instrucción `cv2.imshow(“Nombre”,source)`.

La imagen de referencia seleccionada para realizar la adquisición, pre proceso y segmentación, se muestra en la figura 5.1. Para realizar las pruebas se utiliza tres imágenes adicionales (figura 5.2) para realizar la prueba de reconocimiento en la cual el sistema ejecuta el algoritmo completo solo si la imagen capturada es igual a la imagen de referencia.

:

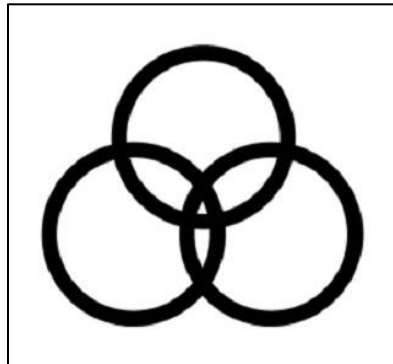


Figura 5.1  
Imagen de referencia

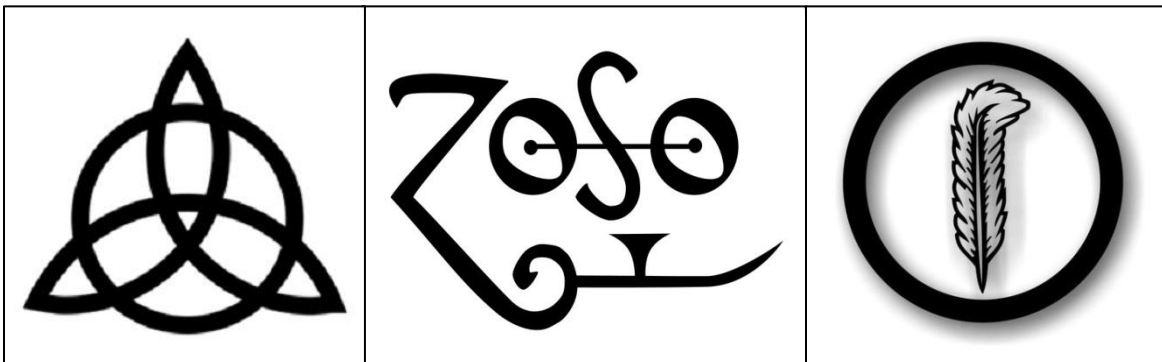


Figura 5.2  
Imágenes para la prueba de reconocimiento

Fuente: <https://ugc.kn3.net/i/origin/http://www.tedelmecanico.com/wp-content/uploads/2009/01/zoso.gif>

### 5.1.1 Adquisición

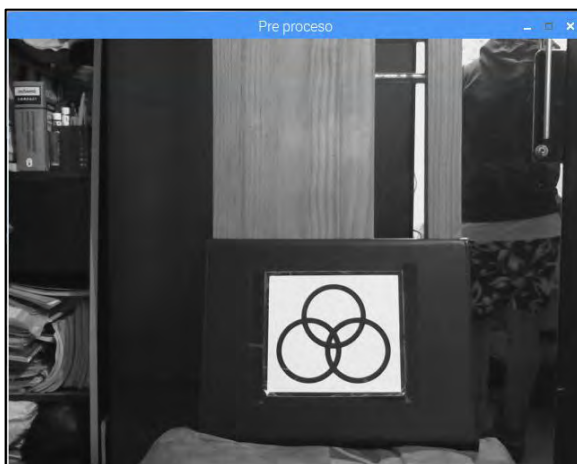
Consiste en capturar una secuencia de imágenes a color con una resolución de 640x480 píxeles (figura 5.3). La captura aparece mediante la instrucción `cv2.imshow("Adquisición",frame)`.



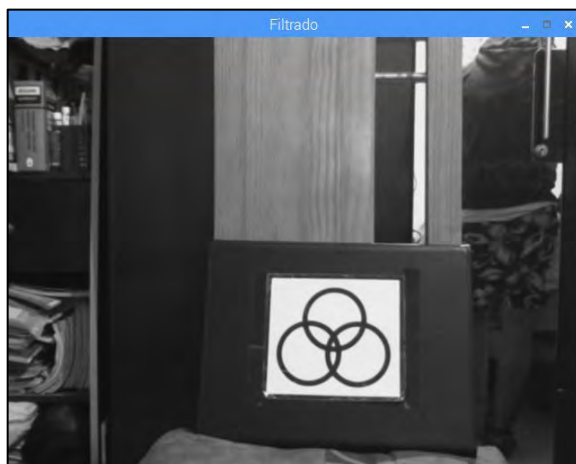
*Figura 5.3  
Imagen original de la figura a reconocer*

### 5.1.2 Pre proceso

Mediante la conversión a escala de grises (figura 5.4) de la imagen a color obtenida en el proceso de adquisición, la carga del procesador disminuye aproximadamente un 70%. A la imagen resultante en escala de grises se le aplica un filtro para eliminar el ruido y delimitar aún más los contornos presentes (figura 5.5).



*Figura 5.4  
Conversión a escala de grises*

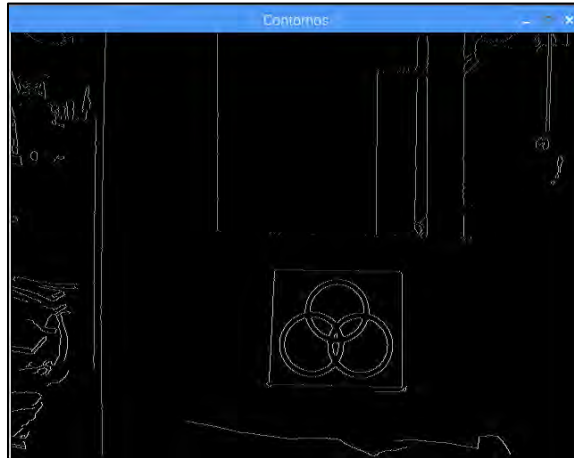


*Figura 5.5  
Filtrado de la imagen en escala de grises*

### 5.1.3 Segmentación

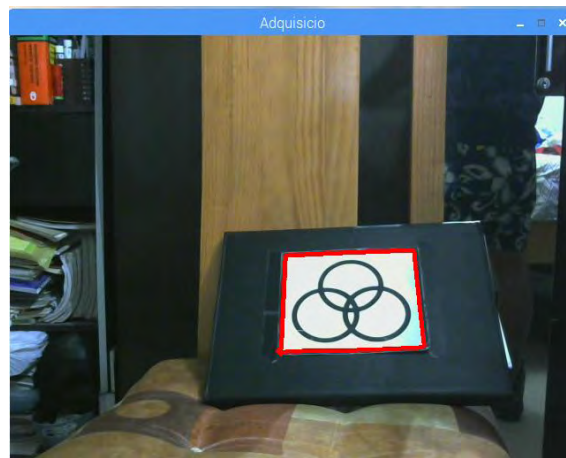
En este proceso se extrae la información necesaria para determinar la ubicación de la imagen de referencia dentro del escenario de trabajo. Para ello se requiere la ejecución de los siguientes sub procesos:

- I. Dibujar todos los contornos de la imagen, como se muestra en la siguiente figura:



*Figura 5.6*  
*Contornos del escenario de trabajo*

- II. Ejecutar una aproximación poligonal de cuatro lados a todos los contornos presentes en la imagen (figura 5.7).



*Figura 5.7*  
*Polígono de cuatro lados detectado*

- III. Determinar el área del polígono. Si el área del polígono calculado se encuentra entre los 5300 y 30000 píxeles, se ejecuta el siguiente proceso. La siguiente gráfica contiene las muestras del área del polígono en donde se observa que el código no se ejecuta si el área excede los límites establecidos.

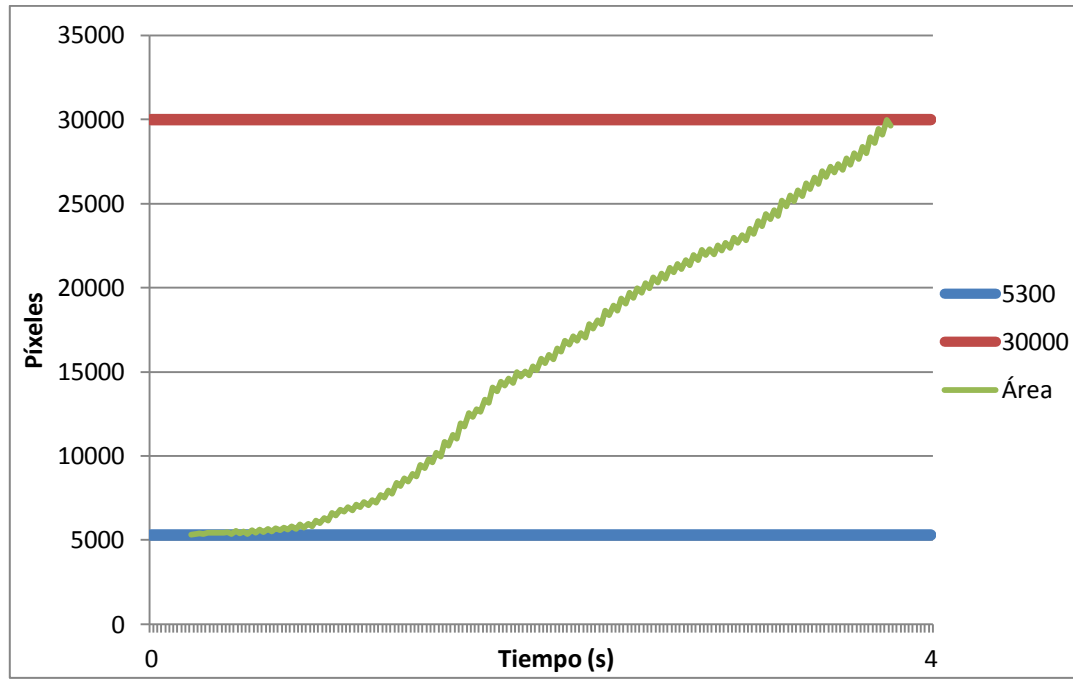


Gráfico 5.1  
Muestreo del área de la imagen de referencia

- IV. Corrección de perspectiva de la imagen en caso de no encontrarse exactamente en posición horizontal y totalmente frontal. La tolerancia de corrección es de una inclinación de  $45^\circ$  en sentido horario y anti horario, como se observa en las siguientes imágenes:

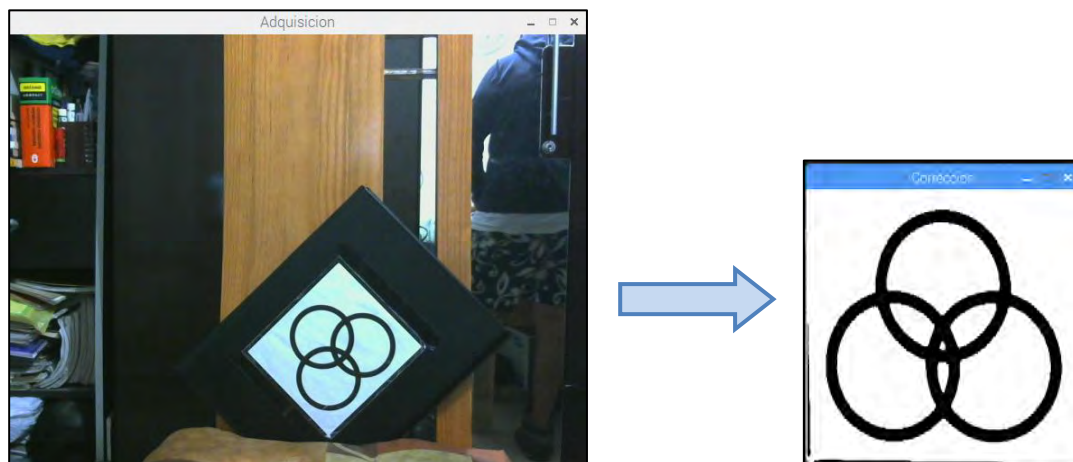
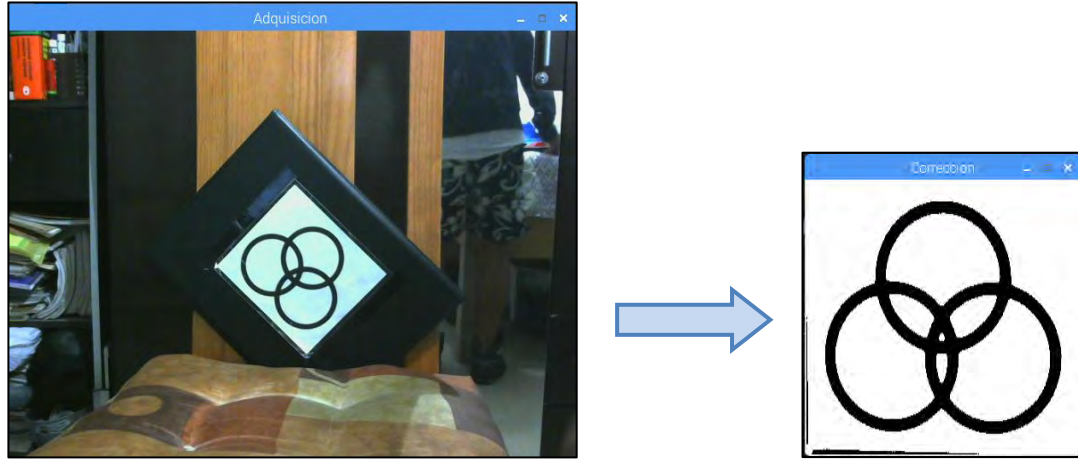
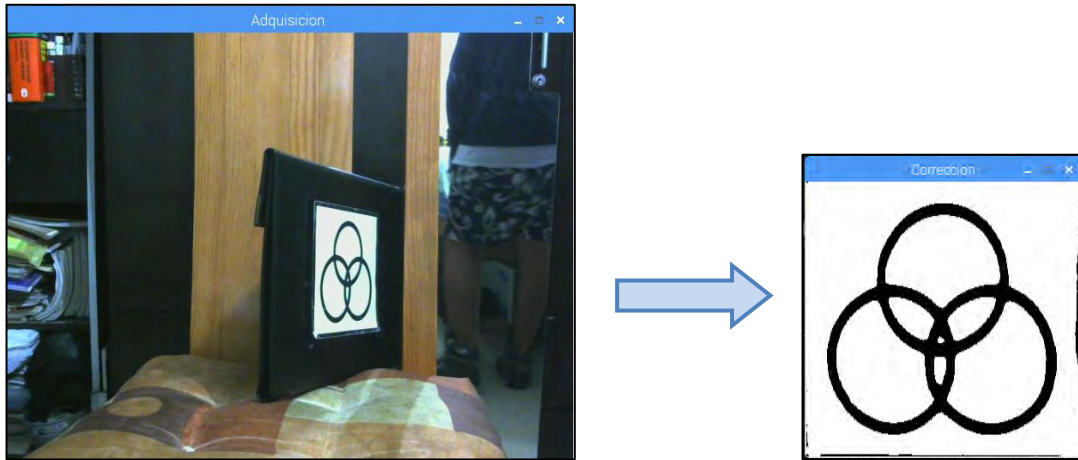


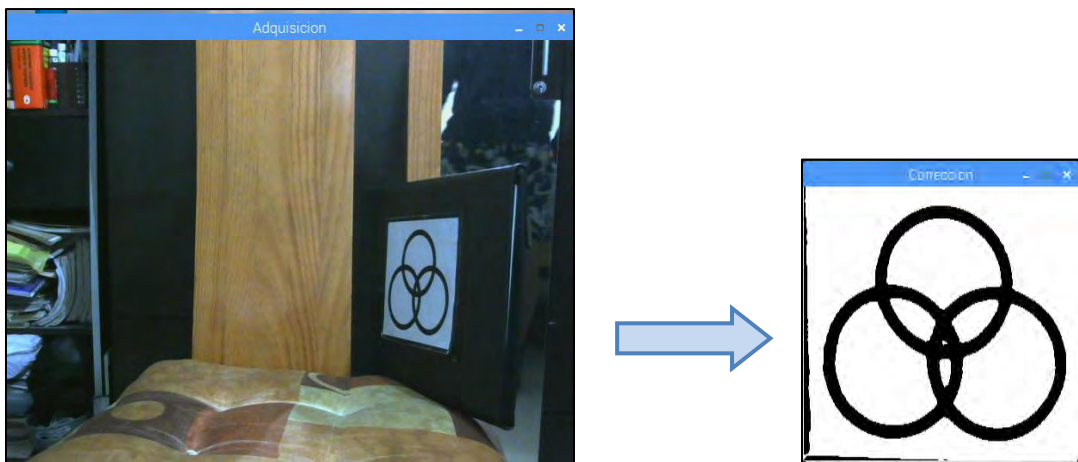
Figura 5.8  
Corrección de la inclinación a  $45^\circ$  sentido anti horario



*Figura 5.9*  
Corrección de la inclinación a 45° sentido horario



*Figura 5.10*  
Corrección de la perspectiva con punto de fuga derecho



*Figura 5.11*  
Corrección de la perspectiva con punto de fuga izquierdo



Posteriormente se debe re escalar la imagen corregida a una dimensión de 320x320 píxeles para poder implementar la máscara XOR, ya que la imagen de referencia tiene este tamaño.

- V. Aplicación de una máscara XOR (figura 5.12) entre la imagen de referencia y la proyección del polígono corregida.

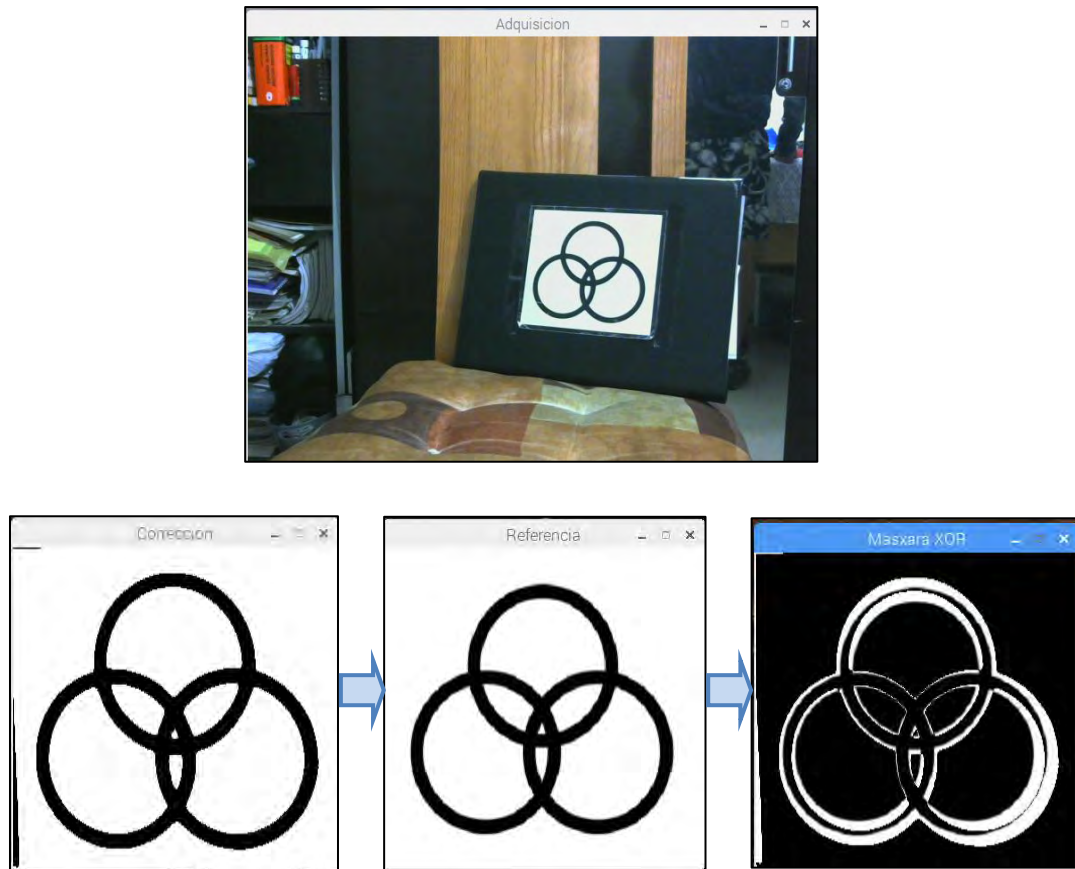


Figura 5.12  
Aplicación de la máscara XOR

- VI. Calcular la diferencia XOR entre la imagen corregida y la imagen de referencia. Se usarán las tres imágenes de prueba para el reconocimiento la imagen de referencia. La siguiente tabla contiene los datos obtenidos de la máscara XOR, dentro del rango en que se ejecuta el algoritmo (5300-30000 píxeles).

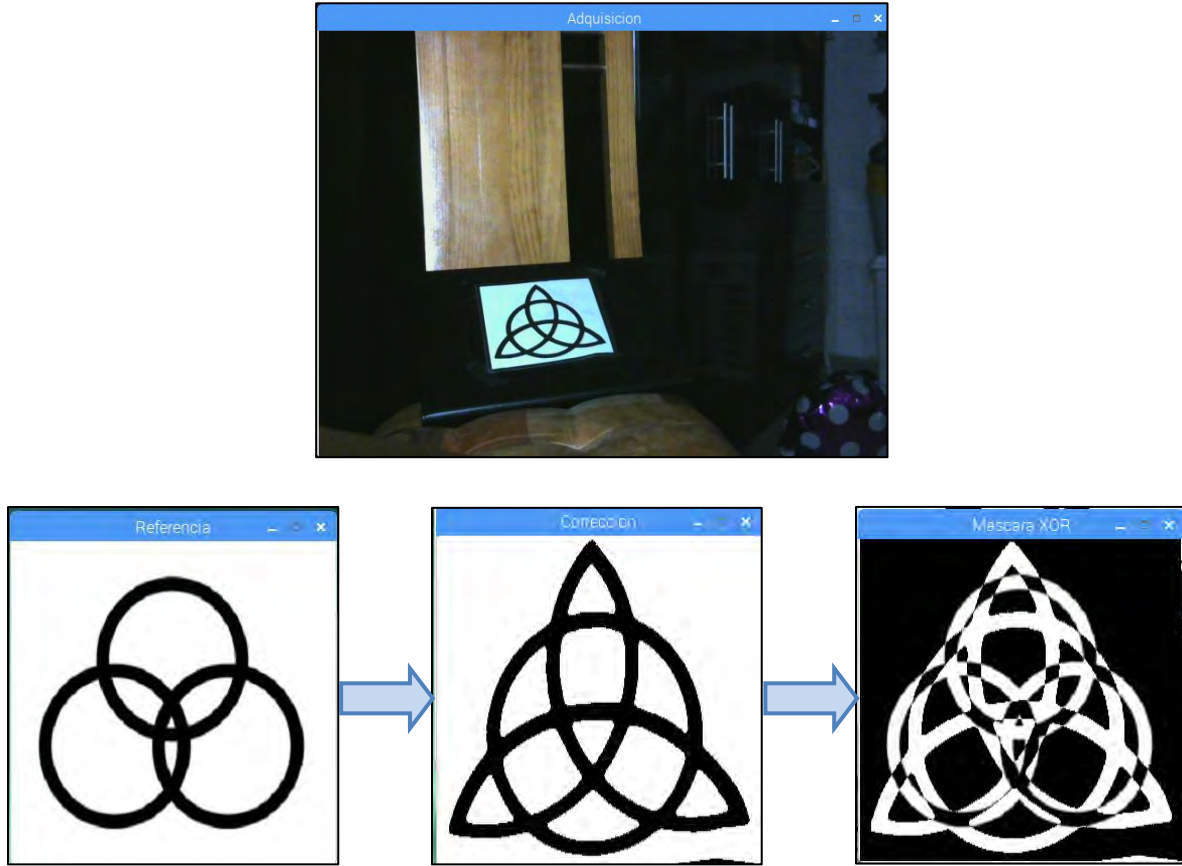


Figura 5.13  
Proceso de reconocimiento de la imagen de prueba 1

Gráfico de la imagen de prueba 1:

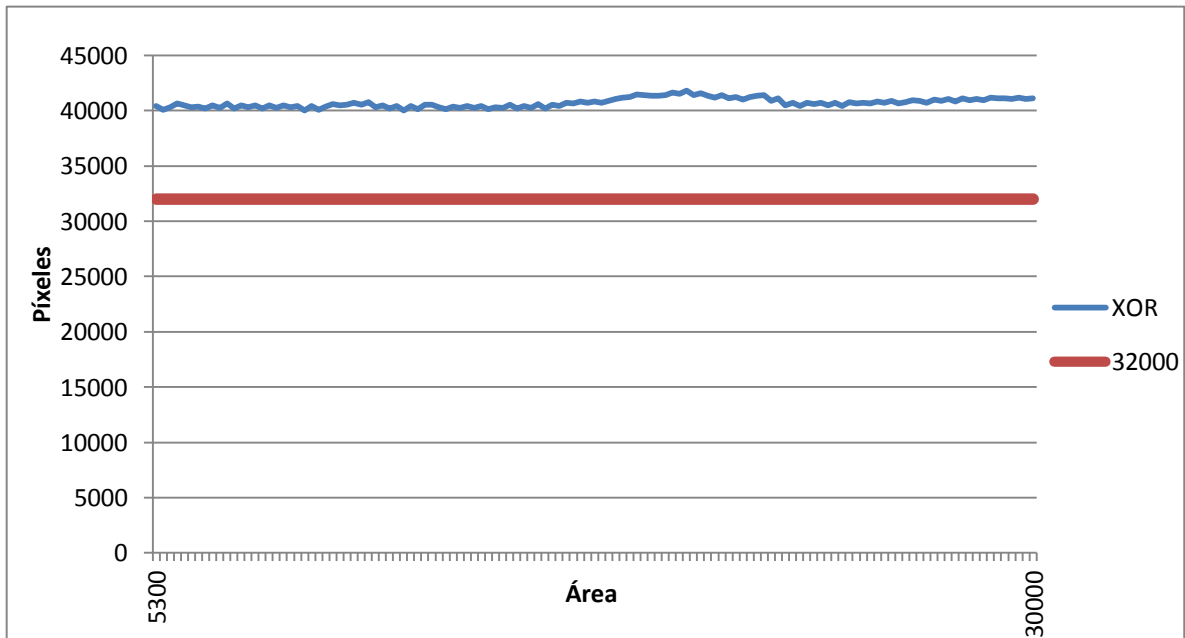


Gráfico 5.2 Píxeles de la imagen de prueba 1

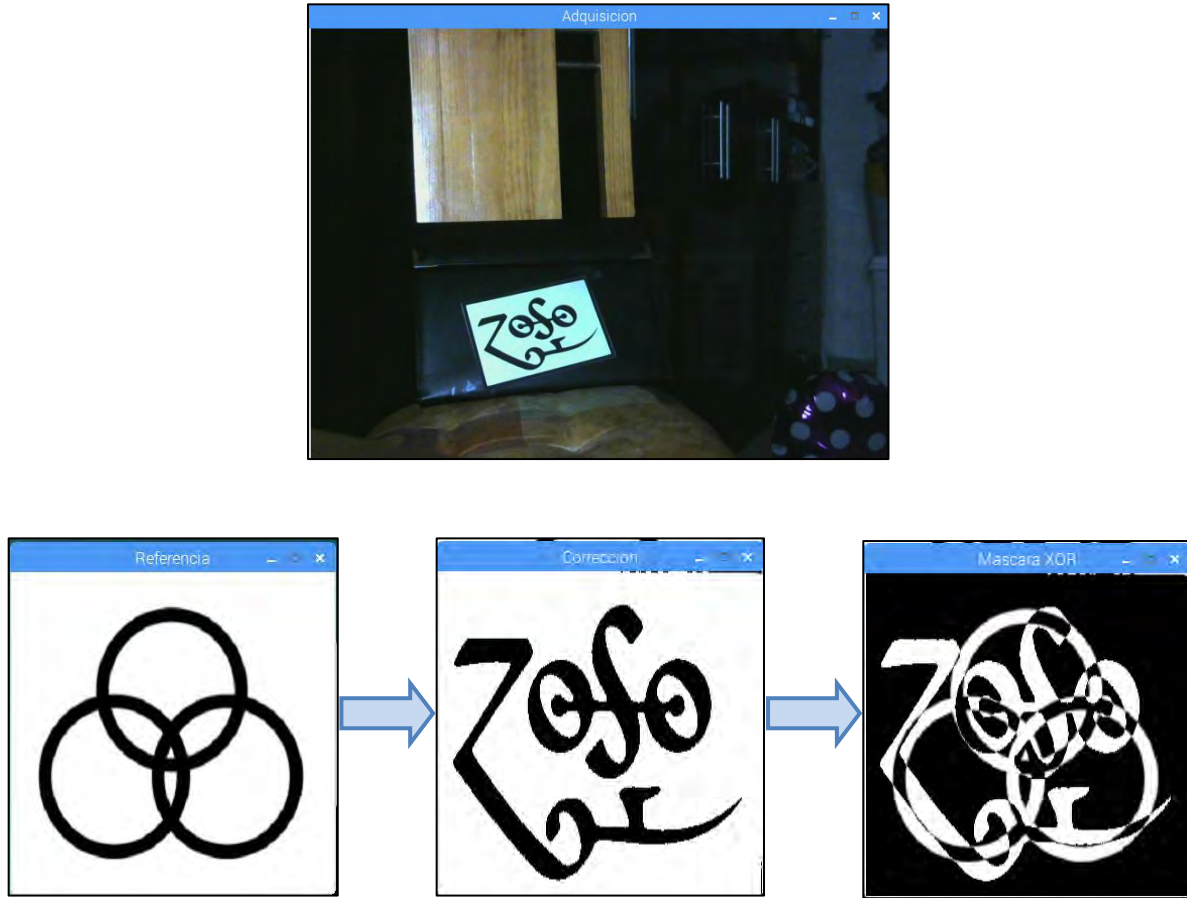


Figura 5.14  
Proceso de reconocimiento de imagen 2

Gráfico de la imagen de prueba 2:

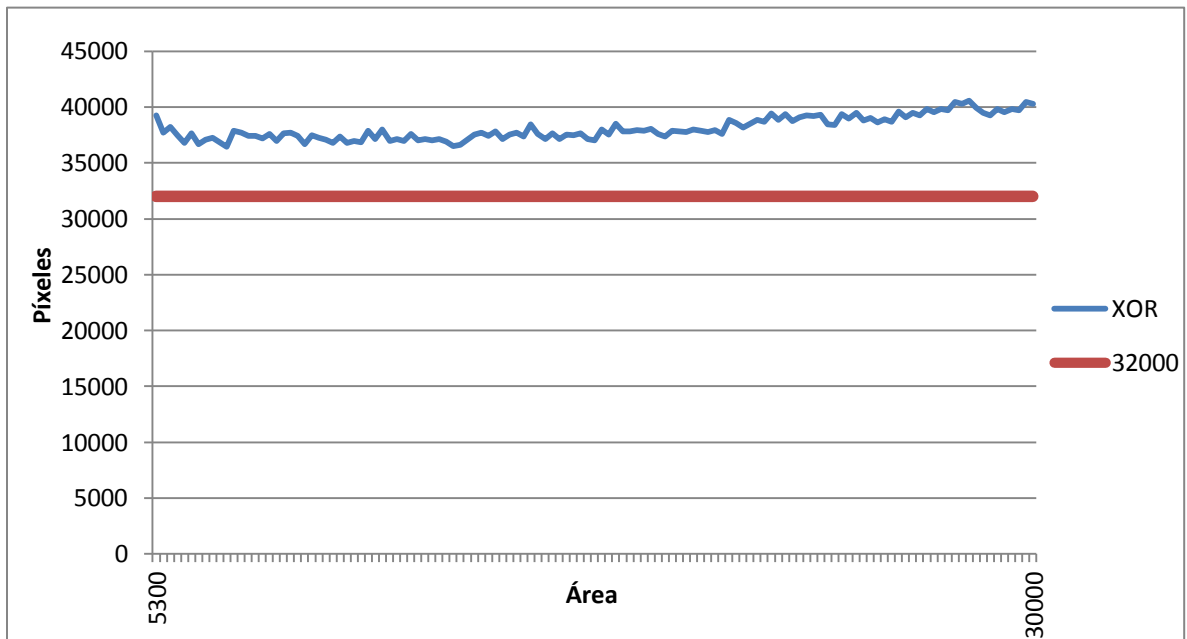


Gráfico 5.3 Píxeles de la imagen de prueba 2

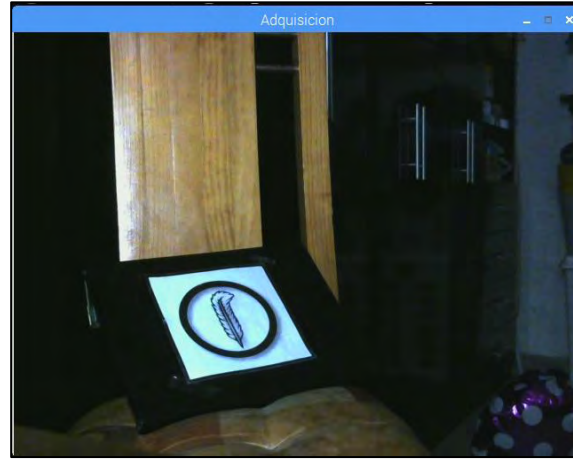


Figura 5.15  
Proceso de reconocimiento de imagen 3

Imagen de prueba 3:

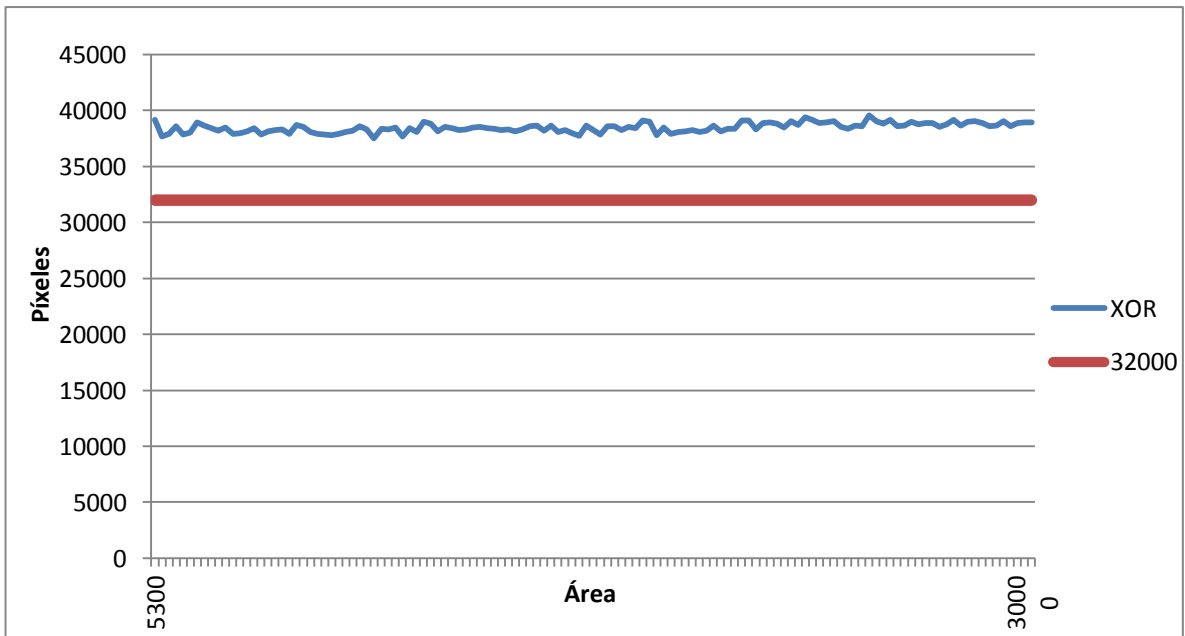


Gráfico 5.4 Píxeles de la imagen de prueba 3

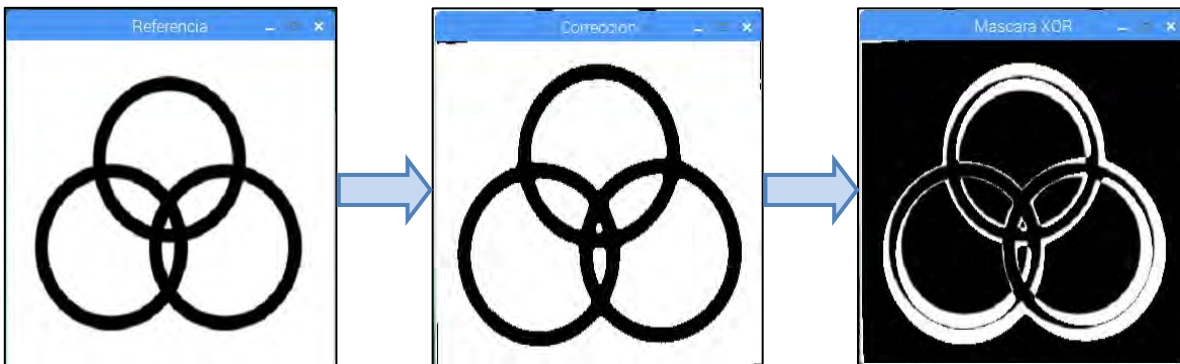
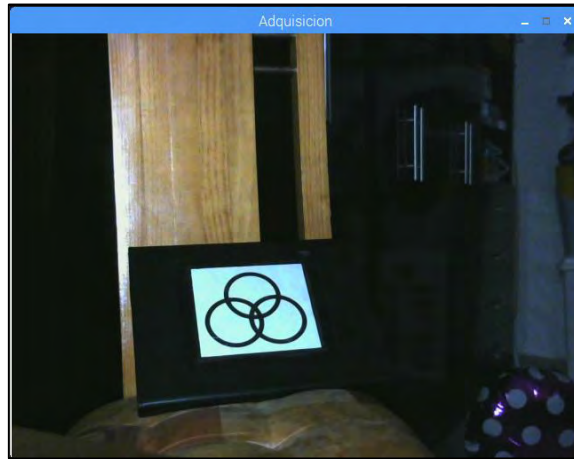


Figura 5.16 Proceso de reconocimiento de imagen de referencia

Imagen de referencia:

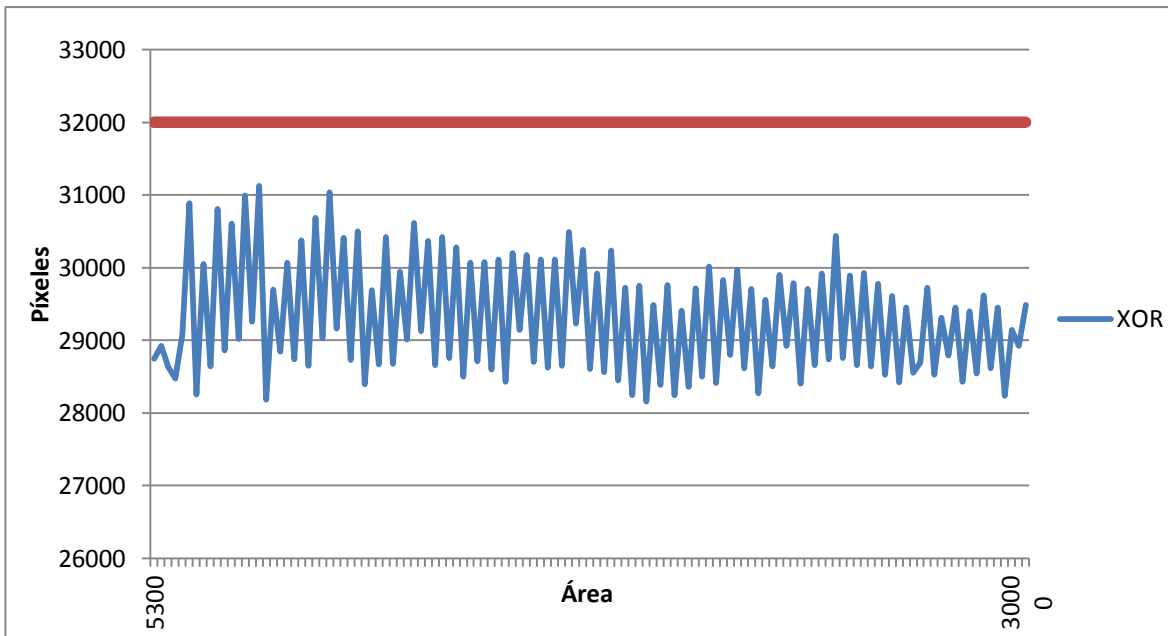


Gráfico 5.5 Píxeles de la imagen de prueba de referencia

La prueba de reconocimiento realizada con la imagen 1 muestra que los píxeles resultantes de la máscara XOR sobrepasan el límite establecido (32000 píxeles) debido a que su valor se encuentra alrededor de los 40000 píxeles y no se puede ejecutar la siguiente instrucción del algoritmo.

La prueba con la imagen 2 indica que tampoco se puede ejecutar la siguiente instrucción debido a que el resultado de la máscara se encuentra entre los 36000 y los 41000 píxeles.

La diferencia de píxeles en la imagen 3 también se encuentra sobre el límite de los 32000 y no es posible ejecutar la siguiente instrucción.

Utilizando la imagen de referencia, el resultado de la máscara XOR (entre 28000 y 31000 píxeles) se encuentra por debajo del límite establecido, por lo que se ejecutan las siguientes líneas del algoritmo que calculan el movimiento frontal y posterior en función de la variación de su área (figura 5.17) y el desplazamiento en el eje x del escenario de trabajo.

- VII. Dentro del área de reconocimiento determinada entre 5300 y 30000 píxeles, se realiza un escalamiento para trabajar solamente con números enteros en un rango de 0 → 18 y posteriormente realizar la conversión de variable *int* → *char* (entero → carácter ASCII) haciendo referencia a la siguiente matriz:

$$a = [",", "(", ")", "*", "+", ",", "-", ":", "/", "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]$$

En la siguiente tabla se muestra la correspondencia entre la cantidad de píxeles de la máscara resultante, su valor en entero y el carácter asignado.

Área (Pixel)	Escala (int)	ASCII (char)
30000	0	,
28700	1	(
27400	2	)
26100	3	*
24800	4	+
23500	5	,
22200	6	-
20900	7	.
19600	8	/
18300	9	0
17000	10	1
15700	11	2
14400	12	3
13100	13	4
11800	14	5
10500	15	6
9200	16	7
7900	17	8
6600	18	9

Tabla 5.1  
Tabla de conversión en función del área

Los resultados obtenidos se muestran en las siguientes figuras. Mediante la siguiente instrucción se imprime los datos en la consola para verificar la correspondencia en la tarjeta de control de motores:

```
print "Area(pixel) =",area,"/ Escala(int) =",mov,"/ ASCII(char) =",a[mov]
```

Dónde:

*area* = área calculada de la imagen (píxeles)

*mov* = Escala de tipo *int* (entero)

*a[mov]* = Dato para transmitir de tipo *char* (caracter)

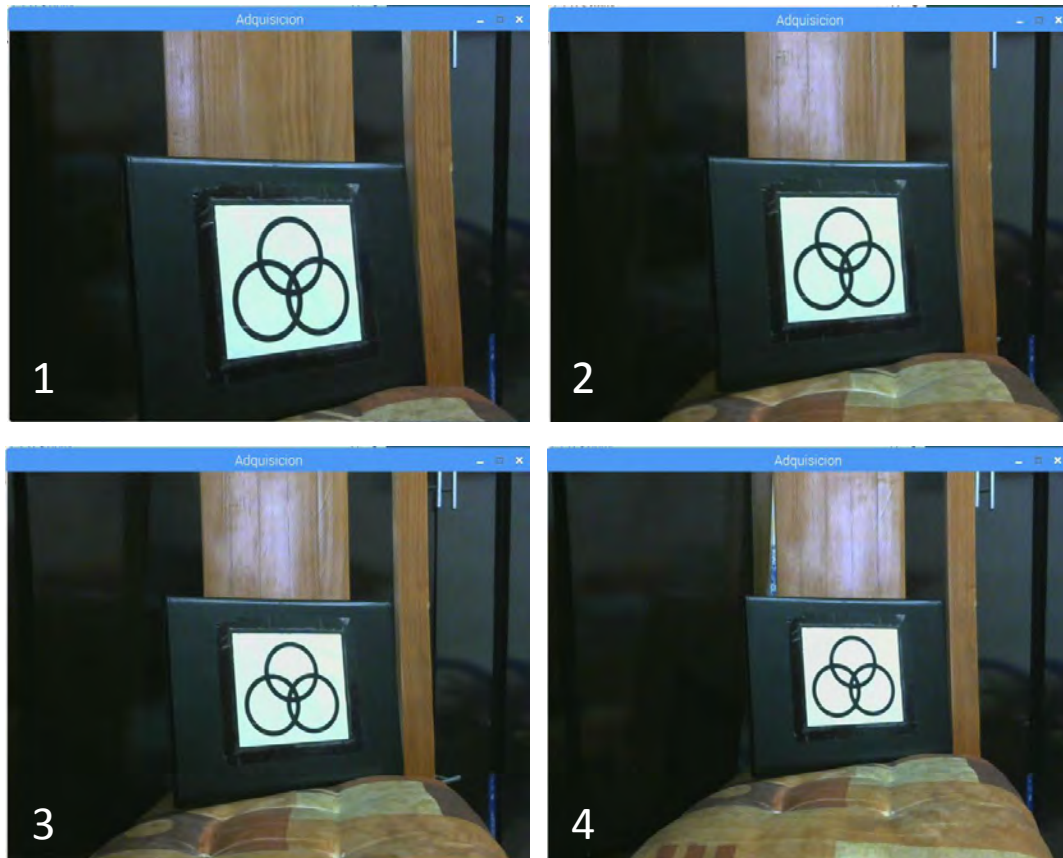


Figura 5.17  
Secuencia de variación del área de la imagen de referencia

La correspondencia del área calculada y valor de tipo entero se muestra en la siguiente gráfica:

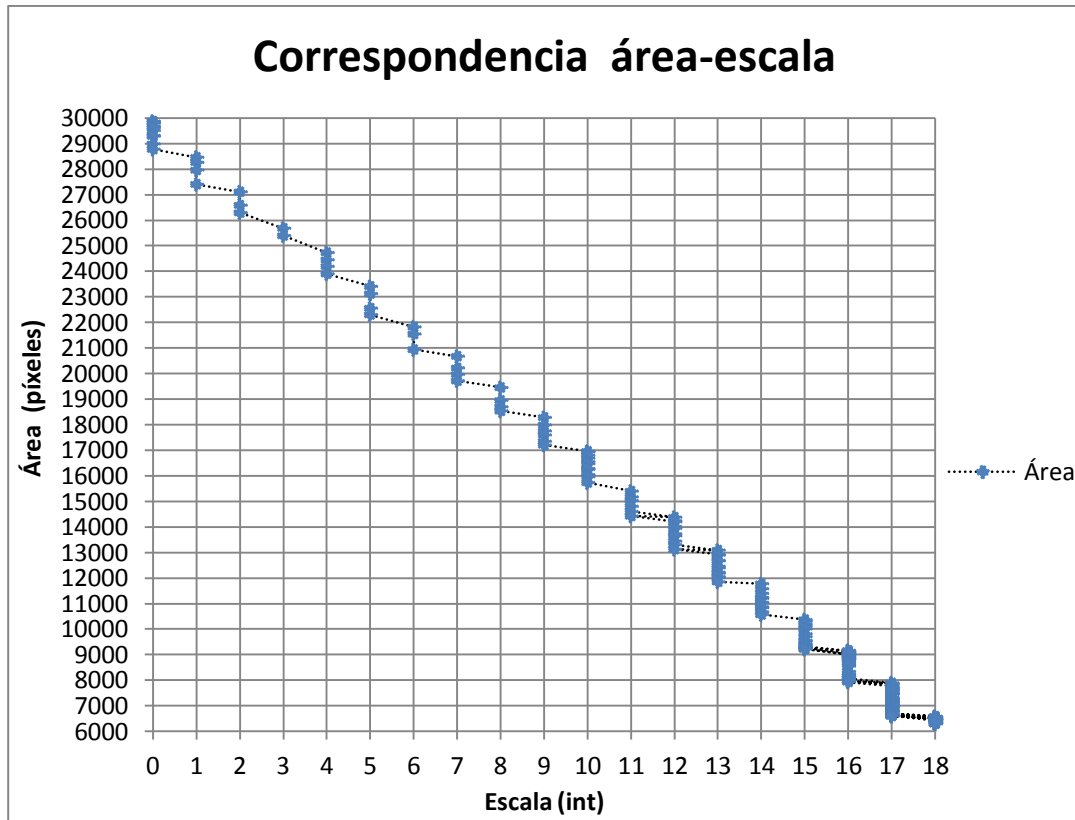


Gráfico 5.6  
Distribución del área en los valores de escala

VIII. Se calcula la ubicación de la imagen en el eje x del escenario de trabajo el cual se divide en 18 segmentos. Cada segmento hace referencia a un valor entero que a su vez, hace referencia a la siguiente matriz:

$$b = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s"]$$

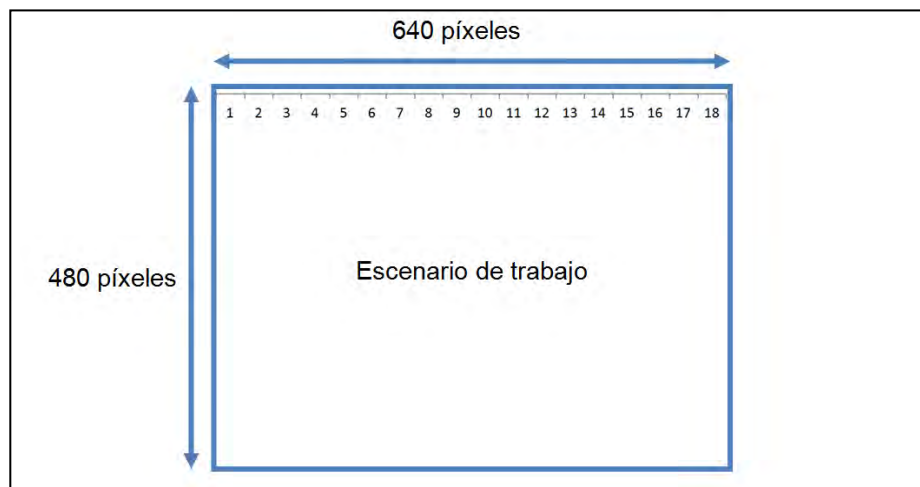


Figura 5.18  
Segmentación del escenario de trabajo



La correspondencia de variables se determina conforme a la siguiente tabla:

Eje X (Píxel)	Escala (int)	ASCII (char)
0	0	a
36	1	b
71	2	c
107	3	d
142	4	e
178	5	f
213	6	g
249	7	h
284	8	i
320	9	j
355	10	k
391	11	l
426	12	m
462	13	n
497	14	o
533	15	p
568	16	q
604	17	r
639	18	s

Tabla 5.2  
Tabla de conversión en función del eje x de la imagen

Los resultados obtenidos se muestran en la figura 18. Se imprime los datos en la consola mediante la instrucción:

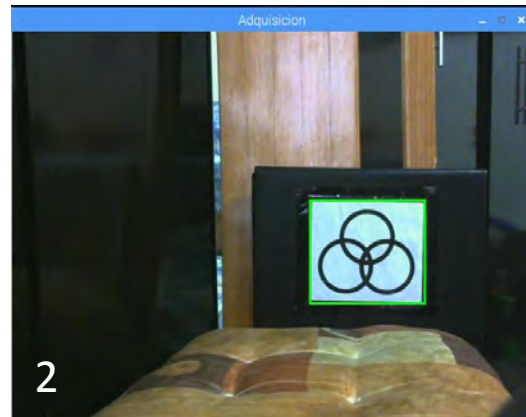
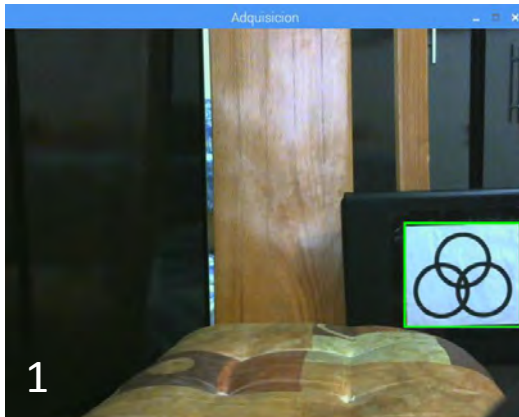
```
print "Eje X(pixel) =",x,"/ Escala(int) =",cx,"/ ASCII(char) =",b[cx]
```

Dónde:

**x** = posición de la imagen en el eje x del escenario de trabajo (píxeles)

**cx** = Escala de tipo *int* (entero)

**b[cx]** = Dato para transmitir de tipo *char* (caracter)



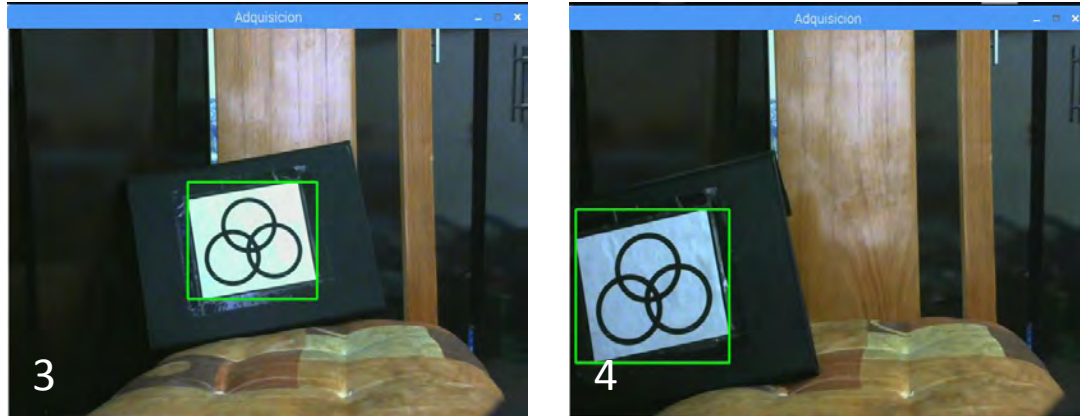


Figura 5.19  
Variación de la posición de la imagen de referencia en el eje x

De los datos de la consola se expresa la convergencia de la posición calculada y el valor de escalamiento de tipo entero en la siguiente gráfica:

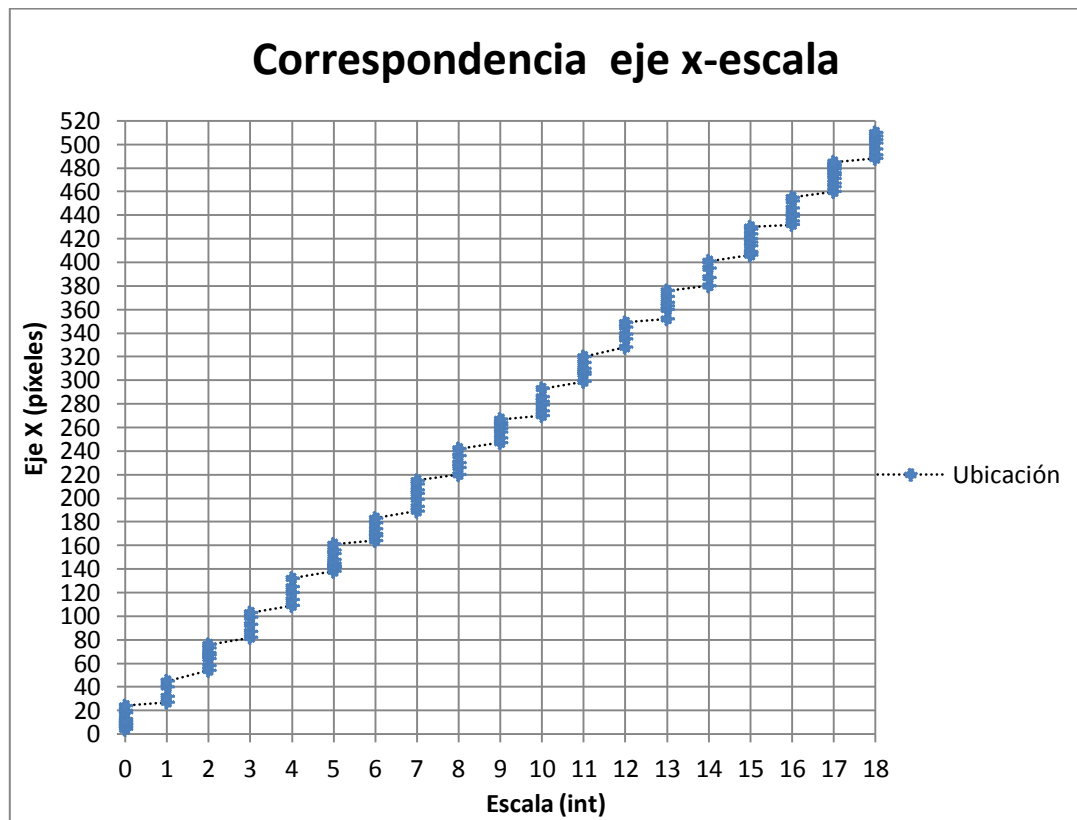


Gráfico 5.6  
Ubicación de la imagen en la coordenada x

### 5.1.4 Transmisión de datos

El sistema de visión artificial finaliza con el envío de los datos obtenidos durante el proceso de reconocimiento y ubicación de la imagen de referencia en el escenario de trabajo. A través del pin de transmisión de la tarjeta *Raspberry Pi 2 model B*® se transmite un *string* por el puerto serial de la siguiente forma:

*String = [a[mov],b[cx],\0]*

Dónde:

*a[mov]* = Direccionamiento de avance- reversa

*b[cx]* = Direccionamiento de giro izquierda - derecha

*\0* = Caracter null de término de cadena

En la consola de *Python* se imprime los datos enviados. Para comprobar que no hay errores en la transmisión, se utiliza un emulador serial virtual y el siguiente algoritmo en la tarjeta *FRDM-KL25Z*®. Este algoritmo recibe un *string* de tres caracteres mediante la instrucción *gets(c,3)*. El primer y segundo caracter se almacenan en las variables *x,y* respectivamente y se imprimen en el emulador serial mediante la instrucción:

*.printf("\n\rString RPi2: %c%c",x,y);*

```
#include "mbed.h"
Serial RPi(NC,PTE1);  //(Tx,Rx)
Serial pc(USBTX, USBRX); // tx, rx

int main()
{
char c[3];
char x,y;

while(true) {
    if (RPi.readable()) {
        RPi.gets(c,3);
        x=c[1];
        y=c[0];
        pc.printf("\n\rString RPi2: %c%c",x,y);
    }
}
}
```

*Figura 5.20*  
Código de prueba para la recepción de datos

Los resultados se muestran en las siguientes imágenes. Se observa que la tarjeta *FRDM-KL25Z®* recibe la misma información que se envía de la tarjeta *Raspberry Pi 2 model B®* y no se presentan errores en la transmisión de datos.

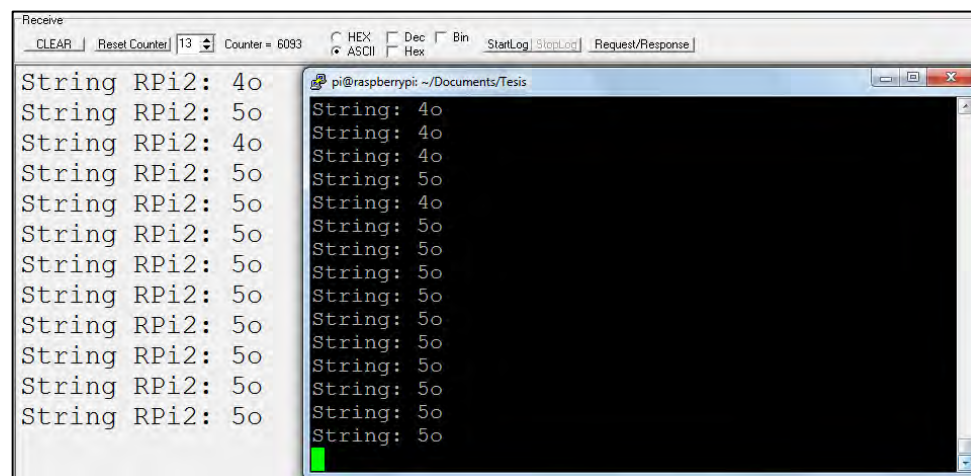
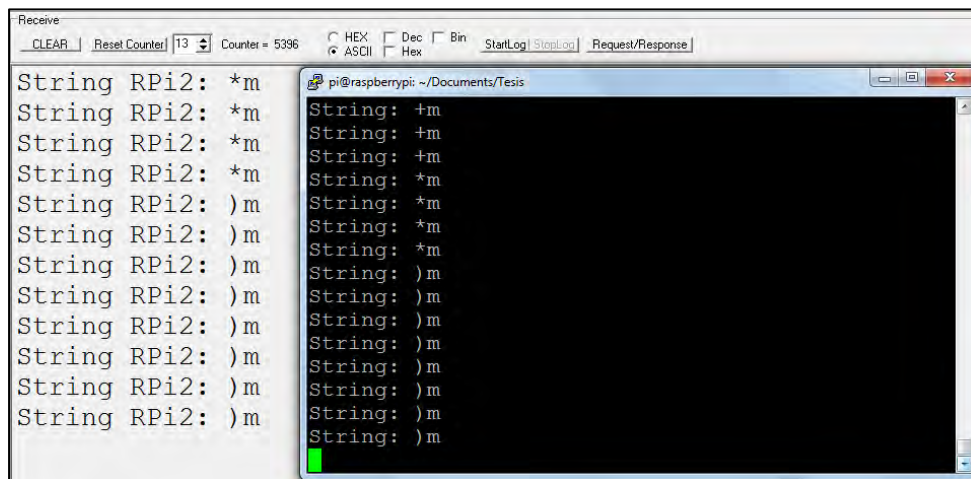
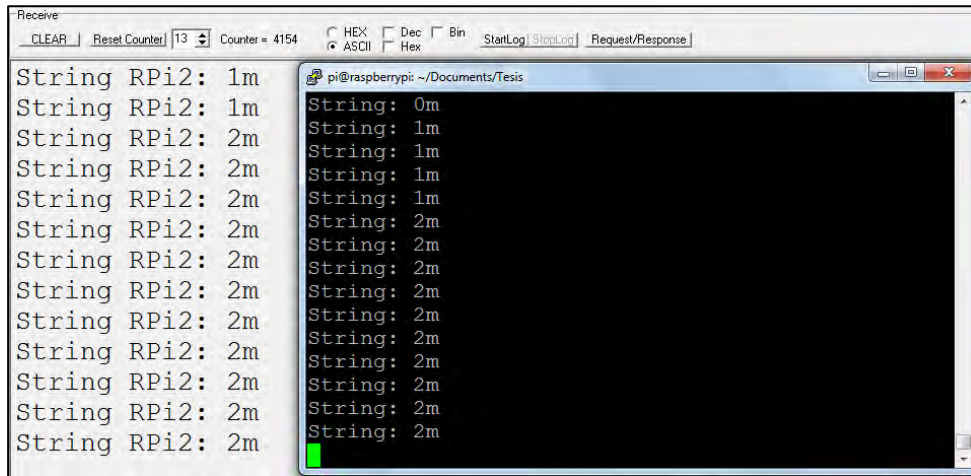


Figura 5.21  
Verificación de datos del emulador de Windows (gris) y consola de Python (negro)

## 5.2 Prueba del sistema manual

El direccionamiento de la silla es posible debido a la inclinación del acelerómetro incorporado en la tarjeta *FRDM-KL25Z*®, la interpretación de la inclinación, la transmisión de datos entre la tarjeta de control manual y la tarjeta de control de motores y mediante el módulo *NRF24L01*®.

Con el algoritmo implementado se envía dos caracteres del código ASCII y se manipula el LED RGB de la tarjeta como referencia visual del usuario en el direccionamiento aplicado a la silla.

### 5.2.1 Prueba de la inclinación en el eje x,y

La inclinación de la tarjeta *FRDM-KL25Z*® sobre el eje x determina el avance o retroceso. Cuando el acelerómetro se encuentra totalmente horizontal se ajusta el valor 0 el cual modifica su valor de 0 → 9 y de 0 → -9 proporcionalmente a la inclinación.

Se implementa el siguiente algoritmo de prueba en la tarjeta *NRF24L01*® para imprimir la información del acelerómetro por el puerto serie visualizándolo en la PC a través del emulador serial, como se muestra en la figura 5.22 y 5.23.

```
#include "mbed.h"
#include "MMA8451Q.h"
    PinName const SDA = PTE25;
    PinName const SCL = PTE24;

#define MMA8451_I2C_ADDRESS (0x1d<<1)

int main()
{
    MMA8451Q acc(SDA, SCL, MMA8451_I2C_ADDRESS);
    while (true) {
        float x, y;
        int8_t x2, y2;
        x2 = acc.getAccX()*10;
        y2 = acc.getAccY()*(-10);
        char a,b;
        a = (x2 + 48);
        b = (y2 + 106);
        printf("\n\rx = %d, y = %d",x2,y2);
    }
}
```

*Figura 5.22*  
*Algoritmo de prueba para la transmisión – recepción de datos del sistema manual*

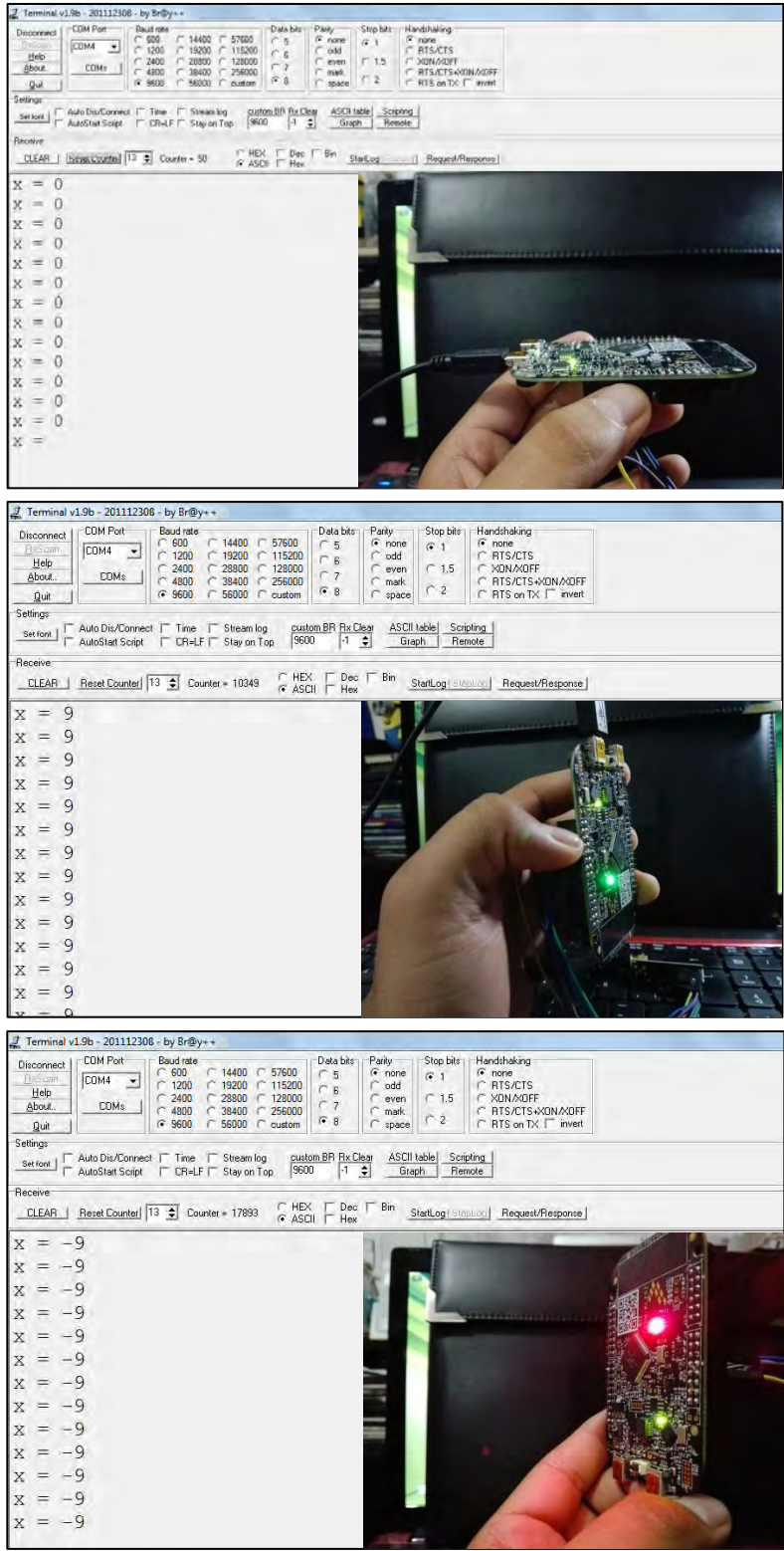


Figura 5.23  
Pruebas de inclinación sobre el eje x

La inclinación de la tarjeta sobre el eje  $y$  (figura 5.24) determina la dirección de giro en la silla. La figura 5.22 muestra la prueba cuando la tarjeta se encuentra en posición horizontal y cuando modifica su valor de  $0 \rightarrow 9$  y de  $0 \rightarrow -9$ , que son valores proporcionales a la inclinación.

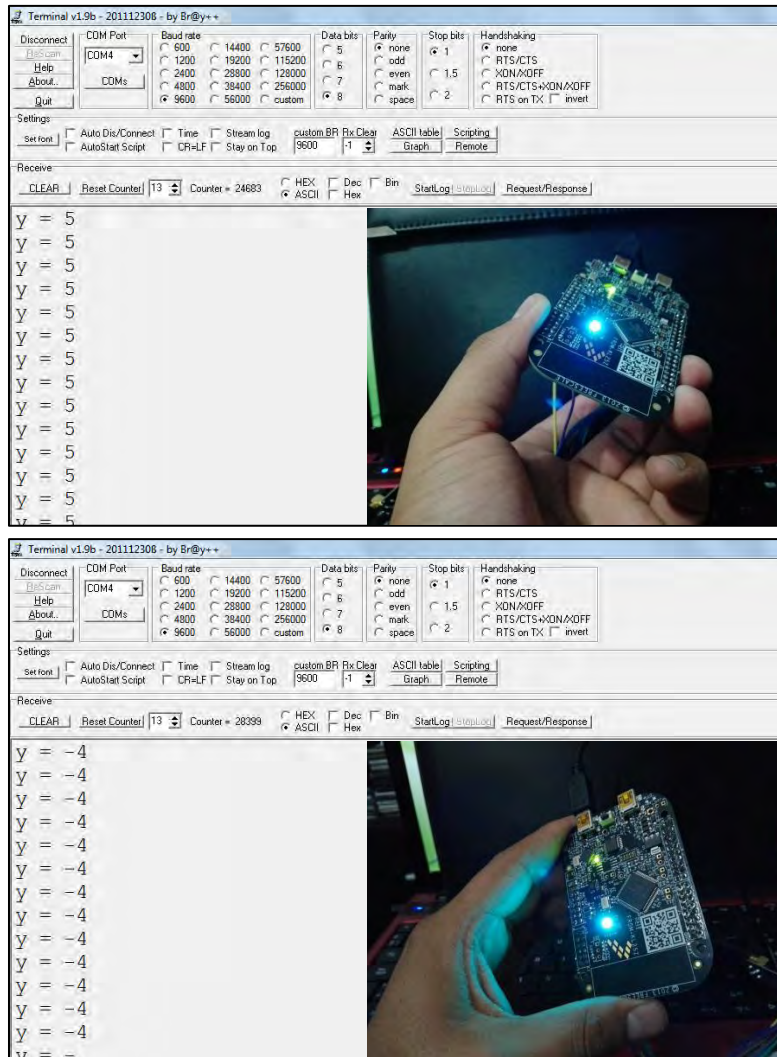


Figura 5.24  
Pruebas de inclinación sobre el eje  $y$

De la prueba de inclinación en el eje  $x,y$  se observa que el micro controlador de la tarjeta *FRDM-KL25Z*® interpreta correctamente los datos generados por el acelerómetro y que la escala de la inclinación corresponde con los valores propuestos para lograr el direccionamiento manual.

## 5.2.2 Transmisión de información

El enlace entre el sistema de direccionamiento manual y el controlador de motores se realiza mediante los módulos de radiofrecuencia *NRF24L01*®. El sistema manual se encarga de transmitir información de la inclinación del acelerómetro y el controlador de motores de recibir e interpretar la información para generar cuatro señales de modulación por ancho de pulso.

Es necesario vincular los dispositivos transmisor y receptor para determinar si hay errores de comunicación entre los componentes. Re escribiendo el código de prueba de del eje x,y como transmisor, se implementa un algoritmo en la tarjeta *FRDM-KL25Z*® que controla los motores configurándolo como receptor, se transmite la información entre módulos y se imprimen los datos enviados y recibidos en el emulador serial, como se muestra en la figura 5.25.

```
#include "mbed.h"
#include "nRF24L01P_PRX.h"
Serial pc(USBTX, USBRX); // tx, rx

int main()
{
    nRF24L01P Receiver(PTD2, PTD3, PTC5, PTD5); // MOSI, MISO, SCK, CSN
    nRF24L01P_PRX PRX(Receiver, PTA13, PTD0); //CE, IRQ
    PRX.Initialize();
    PRX.SetDataRate(1000);
    PRX.SetPayloadSize(1);
    PRX.PowerUp();
    PRX.StartReceive();

    char c;
    int val[2];
    float a,b;

    while (true)
    {
        if(PRX.IsPacketReady())
        {
            PRX.ReadPacket(&c);
            if(c < 68)
            {
                val[0] = (c - 48);
            }
            else if(c > 76)
            {
                val[1] = (c - 106);
            }
            pc.printf("\n\rx = %f, y = %f",val[0],val[1]);
        }
    }
}
```

Figura 5.25  
Algoritmo de transmisión del sistema manual



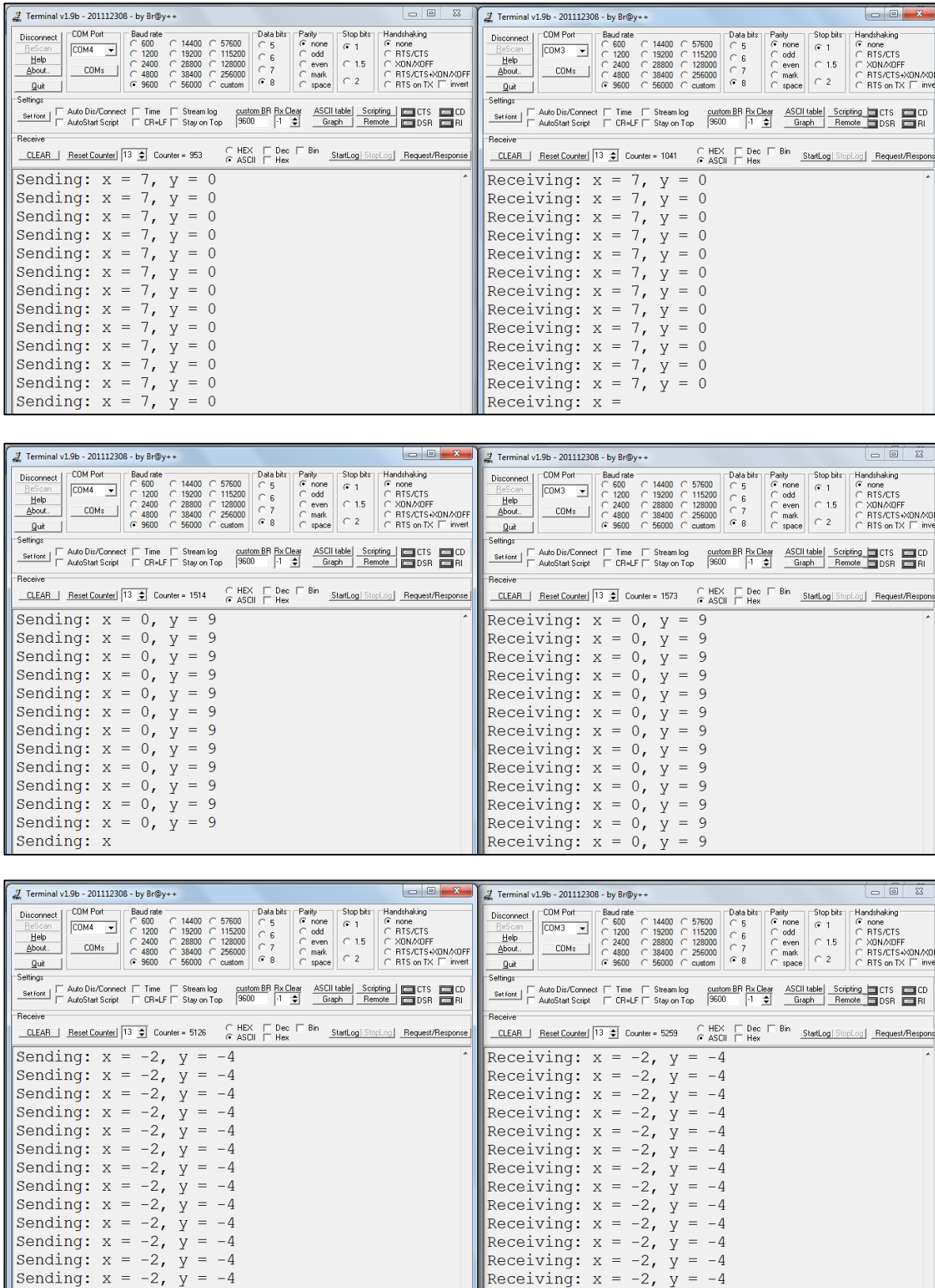


Figura 5.26  
Transmisión y recepción de datos a través del emulador serial

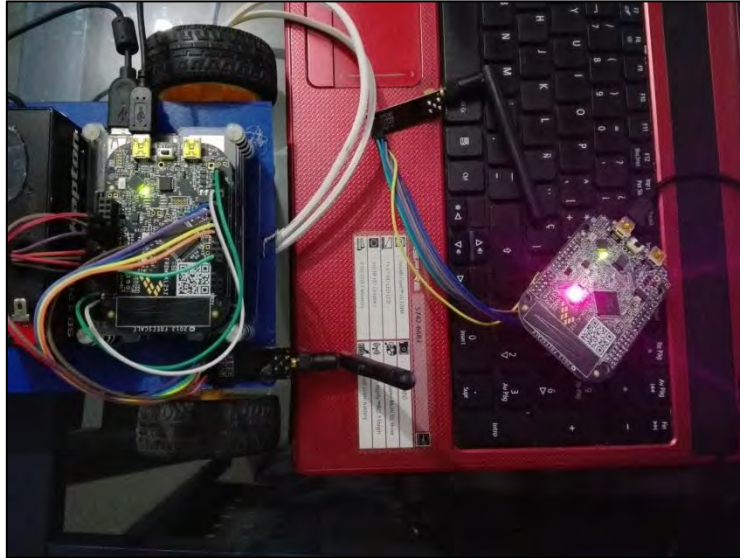


Figura 5.27  
Prototipo de comunicación entre tarjetas *FRDM-KL25Z*® del sistema manual

La transmisión de datos entre las tarjetas *FRDM-KL25Z*® de direccionamiento manual y la del control de motores funciona conforme a lo propuesto. El paquete de datos enviado y recibido es el mismo sin presentar interferencia con otro tipo de señales de la misma frecuencia ya que durante las pruebas había ondas Wi-Fi en el medio ambiente así como dispositivos Bluetooth habilitados.

### 5.3 Prueba del control de motores

La función de la tarjeta *FRDM-KL25Z*® que controladora es recibir e interpretar los datos generados en el sistema de visión artificial y el sistema manual para posteriormente generar cuatro señales modulación de ancho de pulso con ciclo de trabajo variable el cual va en función del dato recibido.

Con el algoritmo final en la tarjeta se puede visualizar mediante el emulador serial los datos recibidos y la interpretación de los mismos. Para verificar los resultados se realiza la medición de las señales resultantes mediante un osciloscopio portátil de la marca *Fluke*® modelo *ScopMeter 123*®.

Las siguientes tablas muestran el porcentaje del ciclo de trabajo presente en los pines del micro controlador para lograr el direccionamiento.

Dirección	x	Duty Cycle	Dirección	y	Duty Cycle (y)	PTB0	PTB1	PTB2	PTB3
Avance	1	10%	Izquierda	-9	90%	x-y	x	0%	0%
				-8	80%	x-y	x	0%	0%
	2	20%		-7	70%	x-y	x	0%	0%
				-6	60%	x-y	x	0%	0%
	3	30%		-5	50%	x-y	x	0%	0%
				-4	40%	x-y	x	0%	0%
	4	40%		-3	30%	x-y	x	0%	0%
				-2	20%	x-y	x	0%	0%
	5	50%		-1	10%	x-y	x	0%	0%
			Línea Recta	0	0%	x	x	0%	0%
	6	60%	Derecha	1	10%	x	x-y	0%	0%
				2	20%	x	x-y	0%	0%
				3	30%	x	x-y	0%	0%
				4	40%	x	x-y	0%	0%
				5	50%	x	x-y	0%	0%
				6	60%	x	x-y	0%	0%
				7	70%	x	x-y	0%	0%
	8	80%		8	80%	x	x-y	0%	0%
9				90%	x	x-y	0%	0%	

Tabla 5.3  
Asignación de ciclo de trabajo para el avance de la silla

Dirección	x	Duty Cycle	Dirección	y	Duty Cycle	PTB0	PTB1	PTB2	PTB3
Reversa	1	10%	Izquierda	-9	90%	0%	0%	x-y	x
				-8	80%	0%	0%	x-y	x
	2	20%		-7	70%	0%	0%	x-y	x
				-6	60%	0%	0%	x-y	x
	3	30%		-5	50%	0%	0%	x-y	x
				-4	40%	0%	0%	x-y	x
	4	40%		-3	30%	0%	0%	x-y	x
				-2	20%	0%	0%	x-y	x
	5	50%		-1	10%	0%	0%	x-y	x
				0	0%	0%	0%	0%	x
	6	60%	Derecha	1	10%	0%	0%	x	x-y
				2	20%	0%	0%	x	x-y
				3	30%	0%	0%	x	x-y
				4	40%	0%	0%	x	x-y
				5	50%	0%	0%	x	x-y
				6	60%	0%	0%	x	x-y
				7	70%	0%	0%	x	x-y
	8	80%		8	80%	0%	0%	x	x-y
9				90%	0%	0%	x	x-y	

Tabla 5.4  
Asignación de ciclo de trabajo para la reversa de la silla

Dirección	x	Duty Cycle (x)	Dirección	y	Duty Cycle	PTB0	PTB1	PTB2	PTB3
Giro	0	0%	Izquierda	-9	90%	0%	y	y	0%
				-8	80%	0%	y	y	0%
				-7	70%	0%	y	y	0%
				-6	60%	0%	y	y	0%
				-5	50%	0%	y	y	0%
				-4	40%	0%	y	y	0%
				-3	30%	0%	y	y	0%
	0	0%	0%	0%	0%	0%	0%		
	0	0%	Derecha	1	10%	y	0%	0%	y
				2	20%	y	0%	0%	y
				3	30%	y	0%	0%	y
				4	40%	y	0%	0%	y
				5	50%	y	0%	0%	y
				6	60%	y	0%	0%	y
				7	70%	y	0%	0%	y
				8	80%	y	0%	0%	y
				9	90%	y	0%	0%	y

Tabla 5.5  
Asignación de ciclo de trabajo para el giro de la silla

### 5.3.1 Direccionamiento de avance

El direccionamiento de avance se ejecuta cuando el valor del eje x es positivo, con el cual obtenemos tres posibles direcciones: avance-izquierda, avance-línea recta, avance-derecha a través de los pines *PTB0* (canal A del osciloscopio) y *PTB1* (canal B del osciloscopio).

Conforme a la tabla de asignación se tiene que:

- $PTB0 = (x - y) = (50\% - 20\%) = 30\% \approx 300nseg$  (5.1)
- $PTB1 = x = 50\% \approx 500nseg$  (5.2)

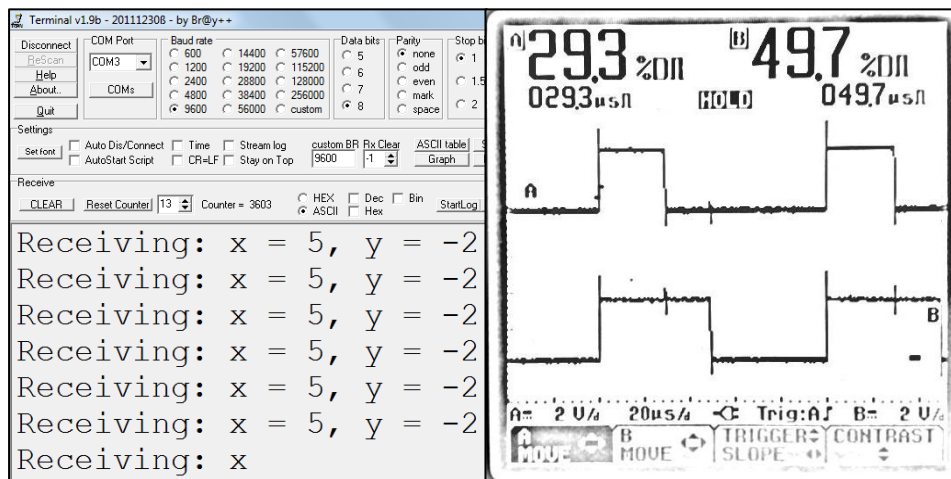


Figura 5.28 Dirección avance a la izquierda

El direccionamiento avance-línea recta se ejecuta solamente si la variación se presenta sobre el eje x y es positiva, como se muestra en la figura 5.29.

Dónde:

- $PTB0 = x = 40\% \approx 400nseg$  (5.3)

- $PTB1 = x = 40\% \approx 400nseg$  (5.4)

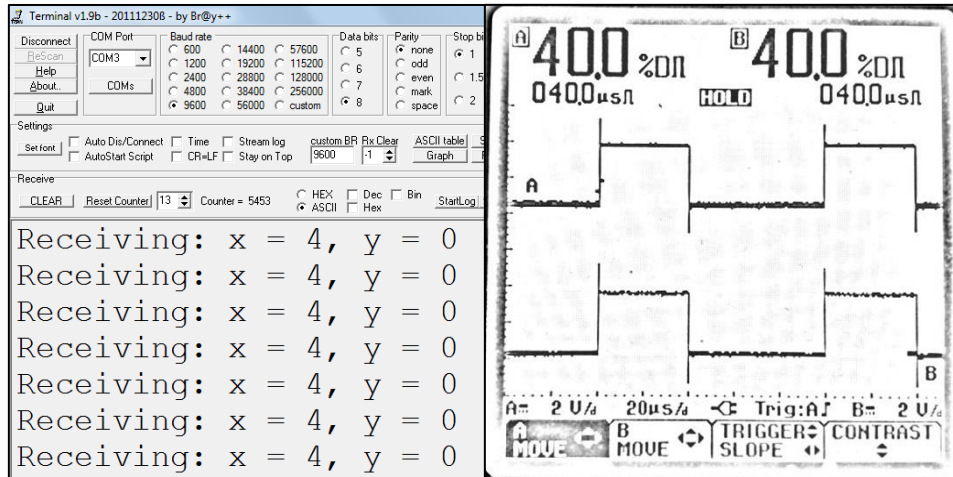


Figura 5.29  
Dirección avance línea recta

El avance-derecha se ejecuta cuando los datos x,y recibidos por la tarjeta *FRDM-KL25Z*® son positivos (figura 5.30).

Dónde:

- $PTB0 = x = 70\% \approx 700nseg$  (5.5)

- $PTB1 = (x - y) = (70\% - 10\%) = 60\% \approx 600nseg$  (5.6)

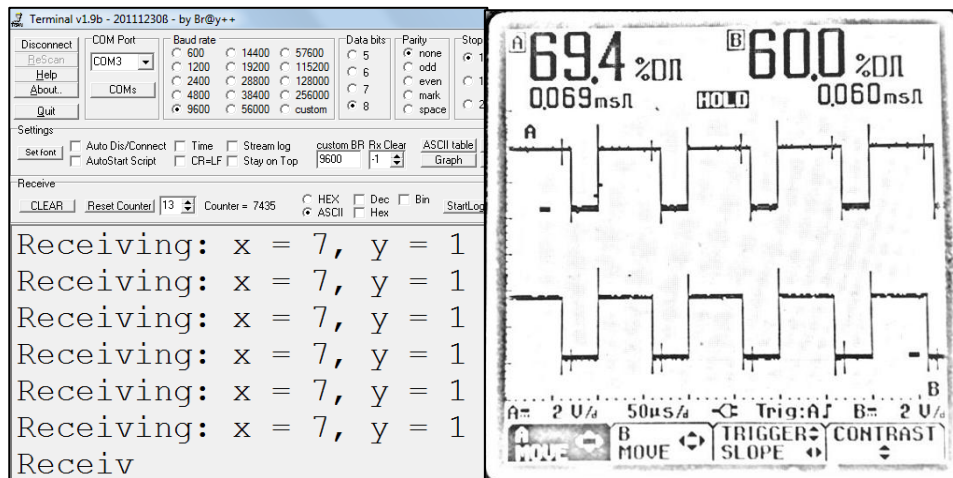


Figura 5.30  
Dirección avance a la derecha

### 5.3.2 Direccionamiento de reversa

Cuando el valor de  $x$  es negativo se ejecuta el direccionamiento de reversa del cual se puede ejecutar reversa-izquierda, reversa-línea recta, reversa-derecha. Los pines configurados para lograr estar trayectorias son  $PTB2$  (canal A del osciloscopio) y  $PTB3$  (canal B del osciloscopio). El direccionamiento de reversa-izquierda se ejecuta si los valores  $(x,y)$  son negativos.

Siendo:

- $PTB2 = (x - y) = (70\% - 30\%) = 40\% \approx 400nseg$  (5.7)

- $PTB3 = x = 70\% \approx 700nseg$  (5.8)

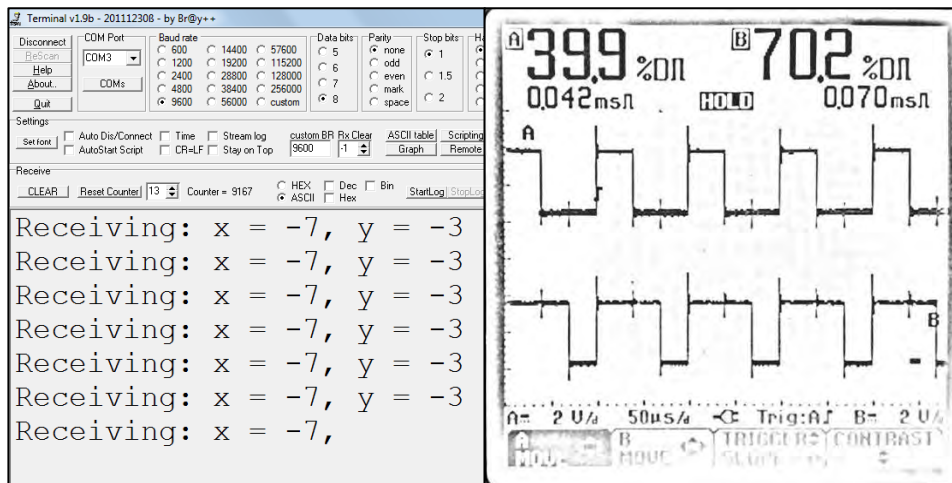


Figura 5.31  
Dirección reversa a la izquierda

La reversa-línea recta se ejecuta si la variación se presenta solamente sobre el eje  $x$  y es negativa, como se muestra en la figura 5.32, dónde:

- $PTB2 = x = 60\% \approx 600nseg$  en estado alto del PWM (5.9)

- $PTB3 = x = 60\% \approx 600nseg$  en estado alto del PWM (5.10)

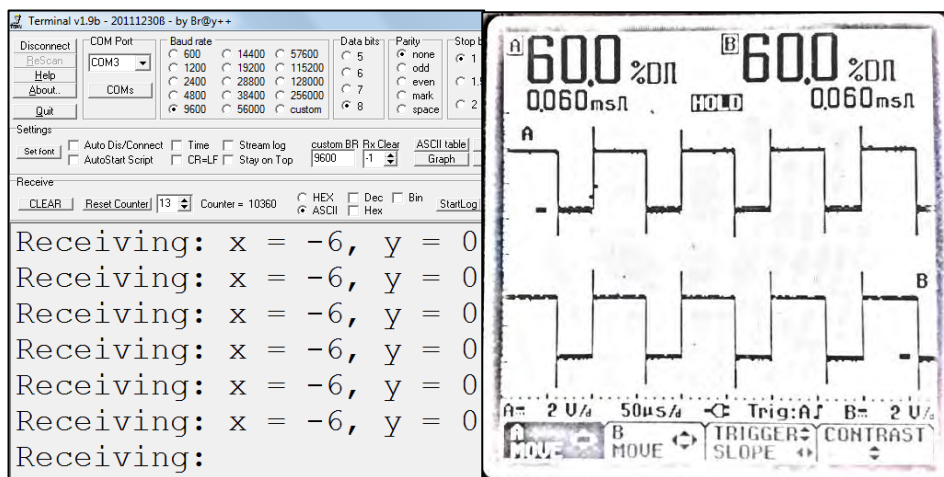


Figura 5.32  
Dirección reversa en línea recta

El direccionamiento reversa-derecha se ejecuta cuando el valor recibido de x es negativo y el valor de y es positivo (figura 5.33).

Siendo:

- $PTB2 = x = 60\% \approx 600nseg$  (5.11)

- $PTB3 = (x - y) = (60\% - 40\%) = 20\% \approx 600nseg$  (5.12)

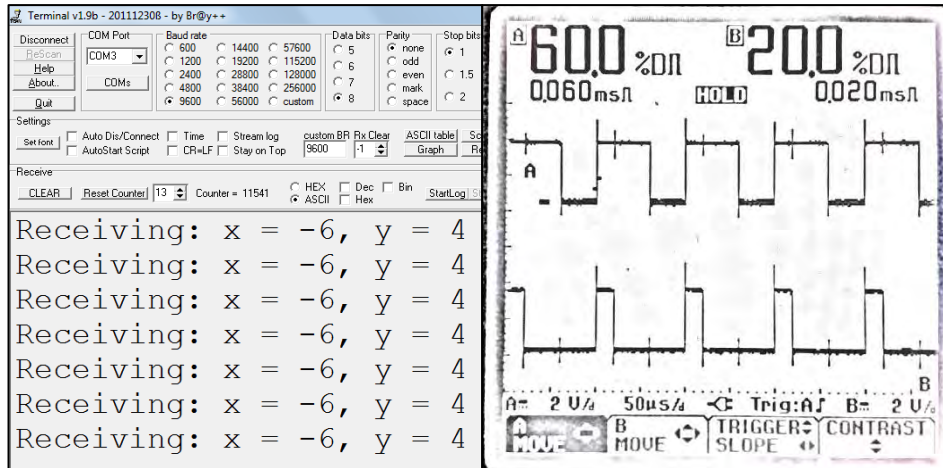


Figura 5.33  
Dirección reversa a la derecha

### 5.3.3 Direccionamiento de giro

El giro a la izquierda-derecha sólo se ejecuta cuando el valor de x cero. El giro a la izquierda se ejecuta cuando y es negativo.

Dónde:

- $PTB1(\text{canal A}) = y = 40\% \approx 400nseg$  (5.13)

- $PTB2(\text{canal B}) = y = 40\% \approx 400nseg$  (5.14)

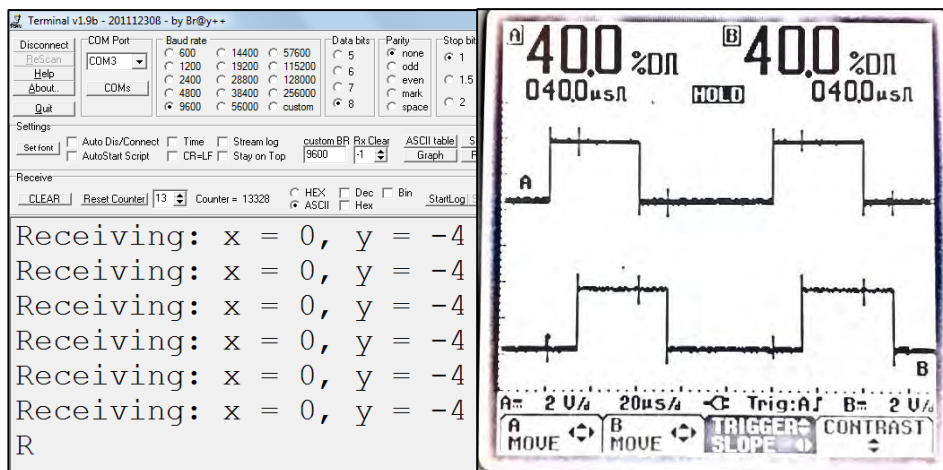


Figura 5.34  
Dirección giro a la izquierda

El giro-derecha se ejecuta cuando el valor de  $y$  es positivo, como se muestra en la figura 5.35.

Dónde:

- $PTB0(\text{canal A}) = y = 60\% \approx 600\text{nseg}$  (5.15)

- $PTB3(\text{canal B}) = y = 60\% \approx 600\text{nseg}$  (5.16)

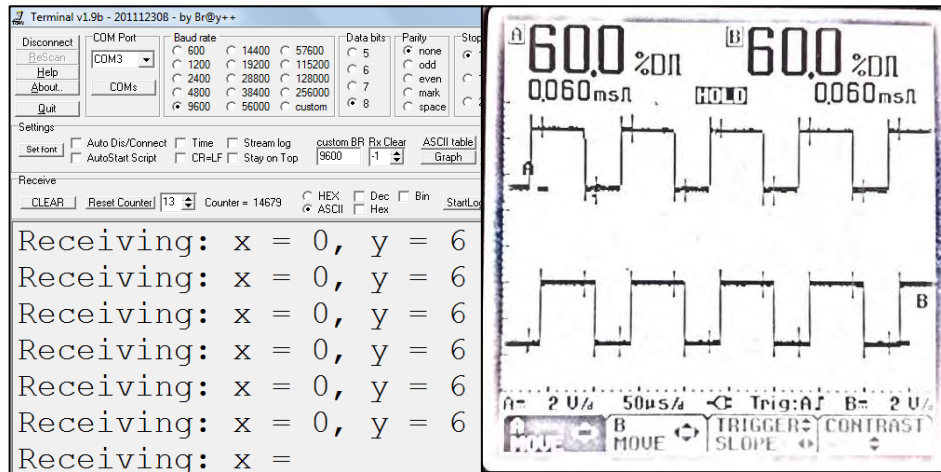


Figura 5.35  
Dirección giro a la derecha

La generación de señales de modulación por ancho de pulso de la tarjeta *FRDM-KL25Z*® que controla los motores se ejecuta conforme a lo propuesto. Los pines corresponden a los asignados así como la frecuencia y el ciclo de trabajo.



### 5.4 Circuito impreso del control manual

A partir del diagrama esquemático propuesto para el sistema de control manual, se genera el circuito impreso con ayuda del software de diseño y simulación *Proteus v8.0*®. Las principales características del diseño son:

- Diseño en doble cara (*Top-Bottom*)
- Componentes de montaje superficial, en inglés, *Surface Mount Device (SMD)* y *Throuh Hole*.
- Dimensión de 6x5 cm.
- Plano de masa para aislar el circuito de ruido electromagnético y disminuir la cantidad de cobre a retirar en la elaboración.

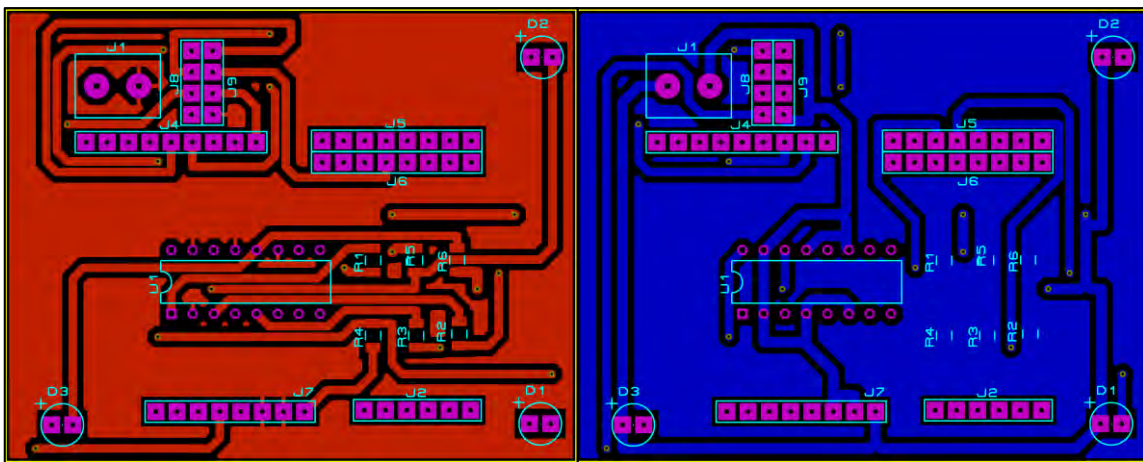


Figura 5.36  
Circuito impreso del sistema manual

La visualización 3D del circuito impreso propuesto se muestra en la siguiente imagen:

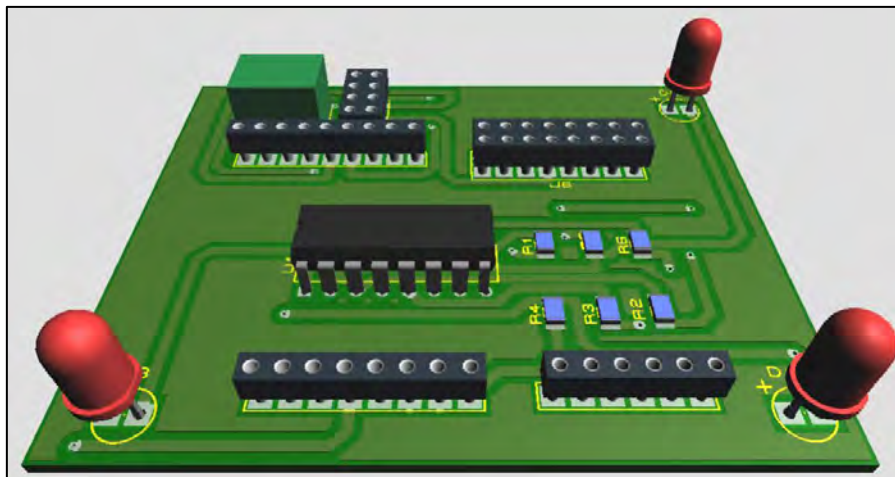


Figura 5.37  
Vista en 3D del circuito impreso (Top)

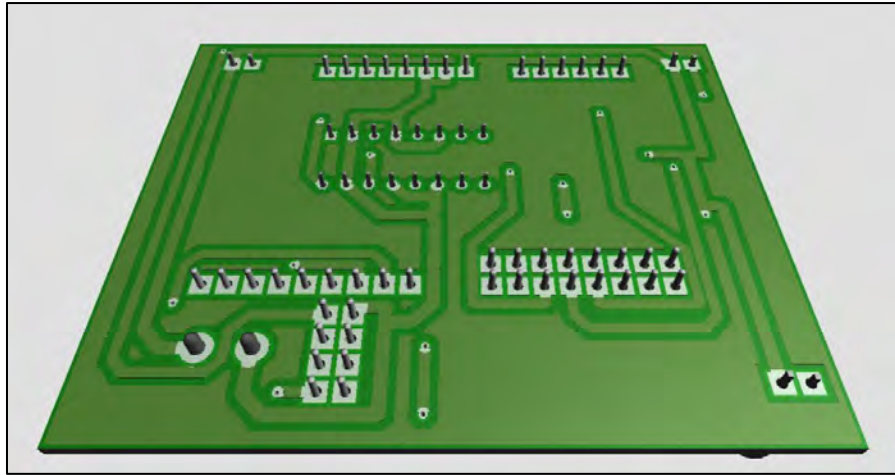


Figura 5.38  
Vista en 3D del circuito impreso (Bottom)

### 5.5 Circuito impreso del control de motores

El circuito impreso del control de motores es más robusto en cuanto a dimensión y cantidad de componentes debido a que en éste se encuentra la etapa de control de motores y la etapa de potencia. En base al diagrama esquemático propuesto y su enrutamiento, se genera el circuito impreso con las siguientes características:

- Diseño a doble cara (*Top-Bottom*)
- Componentes de montaje superficial y *Throuh Hole*
- Dimensión de 20x10 cm
- Plano de masa

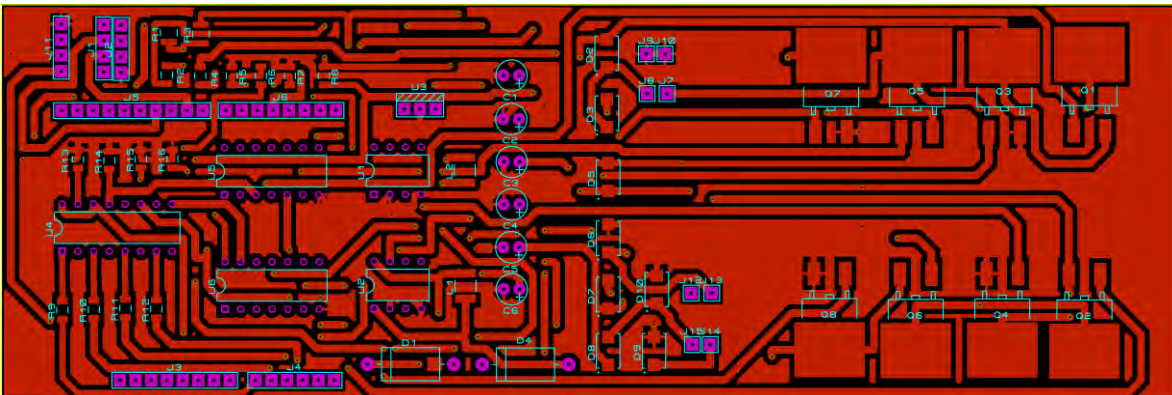


Figura 5.39  
Circuito impreso del control de motores (Top)

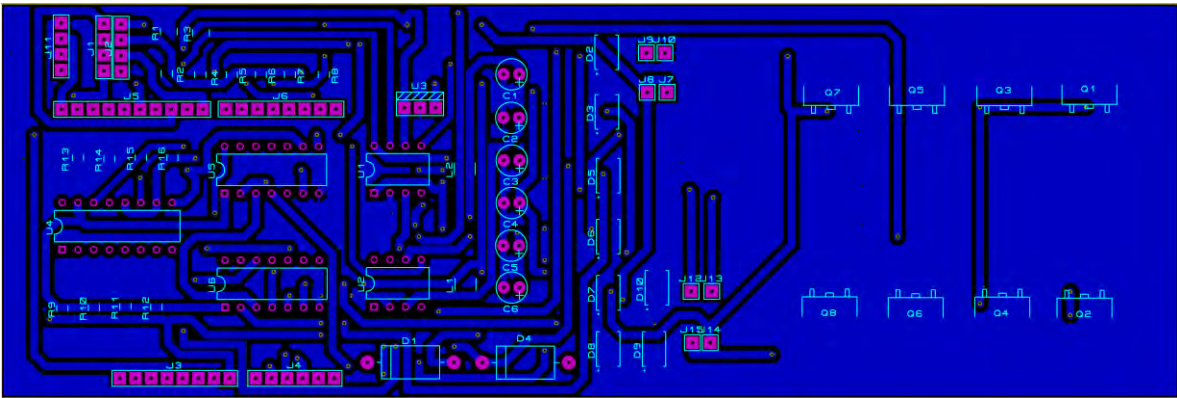


Figura 5.40  
Circuito impreso del control de motores (Bottom)

Y posteriormente se genera la vista del circuito en 3D:

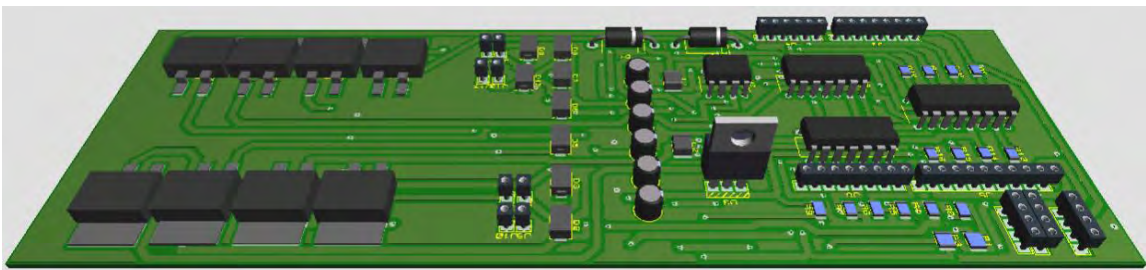


Figura 5.41  
Vista en 3D del circuito de control de motores (Top)

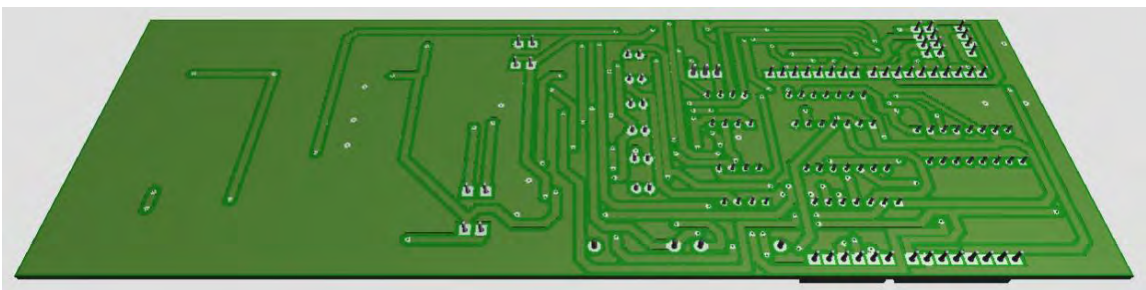
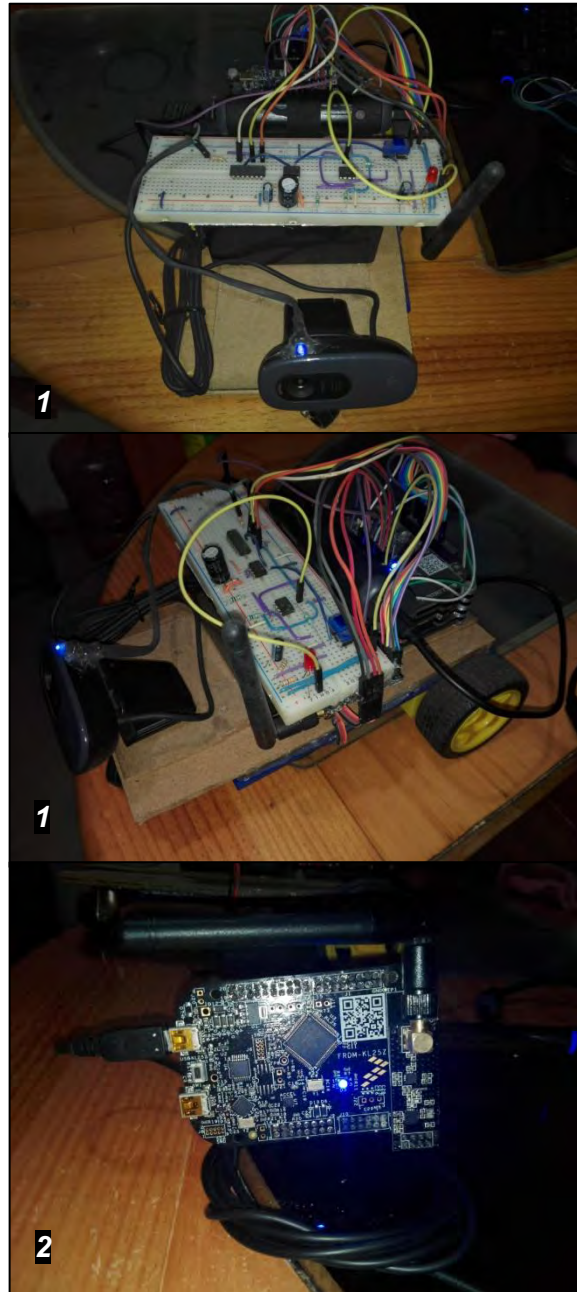


Figura 5.42  
Vista en 3D del circuito de control de motores (Bottom)

Las siguientes imágenes muestran el prototipo de la silla de ruedas y sus componentes



*Figura 5.43*  
1) Prototipo del sistema de visión artificial y controlador de  
2) Prototipo del sistema manual

# Capítulo 6

## Conclusiones

De acuerdo al objetivo planteado, el cual conlleva a desarrollar un sistema de visión artificial para el control de una silla de ruedas, se comprueba que se puede hacer de una manera práctica y de bajo costo utilizando una computadora de placa reducida y dispositivos comerciales de fácil acceso como una cámara *WEB*.

Las pruebas realizadas al sistema de visión artificial demuestran que los procesos previos al reconocimiento de la imagen de referencia se realizan correctamente a pesar de las condiciones de luminosidad del escenario y la cantidad de objetos presentes en el mismo. La corrección de proyección de la imagen de referencia se realiza satisfactoriamente a pesar de que las variaciones implementadas en el proceso de pruebas tienen una probabilidad baja de presentarse en la persona que actúa como guía, la cual es quien posee la imagen que se utiliza como referencia en su torso.

De igual forma, interpretar los movimientos producidos por la imagen de referencia mediante su posición sobre el eje  $x$  así como el cálculo de su área, se ejecuta de forma correcta en el algoritmo el cual envía esta información a la tarjeta de control de motores.

Respecto al control de dirección manual de la silla, éste se desarrolló satisfactoriamente conforme al objetivo trazado, siendo un sistema estable y funcional respecto a la interpretación de la inclinación y transmisión de datos mediante el módulo de radiofrecuencia, probando el sistema en espacios cerrados con interferencias como paredes de concreto.

En cuanto a la recepción e interpretación de los datos provenientes del sistema de visión artificial y del control manual, éste proceso funciona sin anomalías. Las mediciones realizadas mediante el osciloscopio muestran la convergencia entre la información recibida, el pin del micro controlador utilizado y la señal de modulación por ancho de pulso generada para lograr el sentido de giro y velocidad de los motores a través del puente H.

Por lo que, como conclusión final, es factible adaptar este sistema de control a cualquier silla de ruedas mecánica convencional para una persona con parálisis cerebral totalmente dependiente, ya que el bajo costo de los elementos utilizados no superan el de una silla de ruedas eléctrica comercial.

## 6.1 Trabajo a futuro

Se pretende incorporar un sistema de gestión de baterías selladas (*Valve Regulated Lead Acid*) las cuales son libres de mantenimiento y por sus características eléctricas, son las ideales para aplicaciones de vehículos eléctricos.

También se tiene planeado implementar una interfaz HMI con un *display* capacitivo de 7" conectado a la tarjeta *Raspberry Pi 2 model B*® y desarrollar una interfaz gráfica de control con la librería de *QT Creator*®, en donde el usuario pueda elegir entre los modos de operación de la silla y a la vez monitorear los parámetros eléctricos de las baterías como el voltaje, corriente, tiempo de carga y descarga.

Se puede sustituir la micro computadora actual por una *Raspberry Pi 3*® y aprovechar sus dispositivos *Wi-Fi* y *Bluetooth* incorporados, utilizándolos como medios de control o monitoreo de la ubicación, o variables ambientales como el clima predominante.

## Referencias bibliográficas

- [1] Centro Nacional de Excelencia Tecnológica en Salud (CENETEC), *Abordaje y manejo del niño con PARÁLISIS CEREBRAL INFANTIL con Comorbilidades Neurológicas y Músculo Esqueléticas*, [página web en línea], México, Centro Nacional de Excelencia Tecnológica en Salud (CENETEC) , [citado 20/02/16], disponible en internet: [http://www.cenetec.salud.gob.mx/descargas/gpc/CatalogoMaestro/420\\_GPC\\_ParalisisCerebrallInfantil/IMSS-420-10-GER\\_ParalisisCerebrallInfantil\\_CN.pdf](http://www.cenetec.salud.gob.mx/descargas/gpc/CatalogoMaestro/420_GPC_ParalisisCerebrallInfantil/IMSS-420-10-GER_ParalisisCerebrallInfantil_CN.pdf)
- [2] Centro Nacional de Excelencia Tecnológica en Salud (CENETEC), *Evaluación diagnóstica del niño con parálisis cerebral en el tercer nivel de atención*, [página web en línea], México, Centro Nacional de Excelencia Tecnológica en Salud (CENETEC), [citado 20/02/16], disponible en internet: [http://www.cenetec.salud.gob.mx/descargas/gpc/CatalogoMaestro/332\\_DIF\\_09\\_Paralisis\\_cerebral/EyR\\_DIF\\_332\\_09.pdf](http://www.cenetec.salud.gob.mx/descargas/gpc/CatalogoMaestro/332_DIF_09_Paralisis_cerebral/EyR_DIF_332_09.pdf)
- [3] Centro de Cirugía Especial de México (CCEM) IAP, *PARÁLISIS CEREBRAL Y OTRAS FORMAS DE ESPASTICIDAD, ESTADÍSTICAS*, [página web en línea], México, Centro de Cirugía Especial de México (CCEM) IAP, [citado 20/02/16], disponible en internet: <http://www.ccem.org.mx/pci/>
- [4] Instituto Nacional de Estadística Geográfica e Informática (INEGI) 2003, *Estadísticas del Sector Salud y Seguridad Social edición 2003*, [página web en línea], México, Instituto Nacional de Estadística Geográfica e Informática (INEGI), [citado 20/02/16], disponible en internet: [http://www.inegi.gob.mx/prod\\_serv/contenidos/espanol/bvinegi/productos/continuas/social/es/salud/2003/cuaderno20a.pdf](http://www.inegi.gob.mx/prod_serv/contenidos/espanol/bvinegi/productos/continuas/social/es/salud/2003/cuaderno20a.pdf)
- [5] Instituto Nacional de Estadística Geográfica e Informática (INEGI) 2010, *Las personas con discapacidad en México, una visión al 2010*, [página web en línea], México, Instituto Nacional de Estadística Geográfica e Informática (INEGI), [citado 25/02/16], disponible en internet: [http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod\\_serv/contenidos/espanol/bvinegi/productos/censos/poblacion/2010/discapacidad/702825051785.pdf](http://internet.contenidos.inegi.org.mx/contenidos/Productos/prod_serv/contenidos/espanol/bvinegi/productos/censos/poblacion/2010/discapacidad/702825051785.pdf)
- [6] Instituto Nacional de Estadística Geográfica e Informática (INEGI) 2011, *ESTADÍSTICA A PROPÓSITO DEL DÍA DEL NIÑO*, [página web en línea], México, Instituto Nacional de Estadística Geográfica e Informática (INEGI), [citado 26/02/16], disponible en internet: [http://www.finanzas.df.gob.mx/pppeg/docs/Estadisticas\\_dia\\_del\\_nino\\_DF\\_2011.pdf](http://www.finanzas.df.gob.mx/pppeg/docs/Estadisticas_dia_del_nino_DF_2011.pdf)
- [7] CHAIRDEX 2016, *HISTORIA*, [página web en línea], CHAIRDEX, [citado 28/02/16], disponible en internet: <http://www.chairdex.com/shistory.htm>

- [8] Elsevier, septiembre 1999, La silla de ruedas en la parálisis cerebral. Criterios de selección. Asesoramiento para la familia, [página web en línea], Elsevier, [citado 28/02/16], disponible en internet: <http://www.elsevier.es/es-revista-rehabilitacion-120-articulo-las-sillas-ruedas-paralisis-cerebral--13004979>
- [9] International Society of Wheelchair Professionals, 2015, *Motors, Drive trains and External Drive Motor Systems*, [documento PDF en línea], International Society of Wheelchair Professionals, [citado 13/03/16], disponible en internet: [http://www.wheelchairnet.org/wcn\\_wcu/research/stakeholderdocs/pdfs/motors.pdf](http://www.wheelchairnet.org/wcn_wcu/research/stakeholderdocs/pdfs/motors.pdf)
- [10] Neoteo, 2010, Mario, *Puente H con MOSFET para motores CC*, [citado 13/03/16], disponible en internet: <https://www.neoteo.com/puente-h-con-mosfet-para-motores-cc>
- [11] Rosa Rivero Alberto, Montero García-Celay Ignacio y Cruz Lorente María, *El Niño Con Parálisis Cerebral; Enculturación, Desarrollo E Intervención*, [citado 06/03/16], Madrid 1993, Ed. CIDE.
- [12] Ollero Baturone, Aníbal, *Robótica: manipuladores y robots móviles*, [citado 18/04/16], Barcelona 2001, Ed. Marcombo.
- [13] Kelly Rafael, Santibañez Víctor, *Control de movimiento de robots manipuladores*, [citado 25/04/16], Madrid 2003, Ed. Prentice Hall.
- [14] Sucar, L. Enrique y Gómez Giovani, *Visión Computacional*, [documento PDF en línea], *Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México* [citado 15/09/2016], disponible en internet: <http://ccc.inaoep.mx/~esucar/Libros/vision-sucar-gomez.pdf>
- [15] Cristian Andrés Ordoñez Santiago, Revista Digital Universitaria, 2005, *Formatos de Imagen Digital*, [revista digital en línea], [citado 15/09/2016] disponible en internet: [http://www.revista.unam.mx/vol.6/num5/art50/may\\_art50.pdf](http://www.revista.unam.mx/vol.6/num5/art50/may_art50.pdf)
- [16] Davies E.R., *Computer & Machine Vision: Theory, Algorithms, Practicalities, Fourth Edition*, [citado 13/05/16], 2012, Ed. Elsevier
- [17] C. González Rafael, E. Woods Richard, *Digital Image Processing, Second Edition*, [citado 30/06/16], Ed. Prentice Hall.
- [18] Vélez Serrano J.F., *Visión por computador*, [publicación en línea] [citado 6/07/16], disponible en internet: [http://www.researchgate.net/publication/31759521\\_Vision\\_por\\_computador\\_JF\\_Velez\\_Serrano\\_et\\_al](http://www.researchgate.net/publication/31759521_Vision_por_computador_JF_Velez_Serrano_et_al)
- [19] A. Forsyth David, Ponce Jean, *Computer Vision: A Modern Approach*, [citado 15/07/16], 2015, Ed. Pearson Education Limited.
- [20] Eloi Maduell i García, *Visión Artificial*, [publicación en línea], [citado 05/08/16], Universitat Oberta de Catalunya, disponible en internet: [https://www.exabyteinformatica.com/uoc/Informatica/Diseno\\_de\\_interaccion/Diseno\\_de\\_interaccion\\_\(Modulo\\_5\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Diseno_de_interaccion/Diseno_de_interaccion_(Modulo_5).pdf)



- [21] Marta Canael Susana, *Microcontroladores ARM: Advanced RISC Machine*, [publicación en línea] [citado 29/08/16], disponible en internet: [http://www.electron.frba.utn.edu.ar/upload/Materias/950429/archivos/Cap10\\_2009\\_ARM7\\_apunte.pdf](http://www.electron.frba.utn.edu.ar/upload/Materias/950429/archivos/Cap10_2009_ARM7_apunte.pdf)
- [22] *Bo Furber Stephen, ARM System-on-Chip Architecture*, [citado 16/09/16], 2000, Ed. Addison-Wesley
- [23] *Perles Ángel, ARM Cortex-M: Introducción a los microcontroladores STM32 de ST*, [documento PDF en línea], Universitat Politècnica de Valencia, [citado 10/10/16], disponible en internet: [http://www.disca.upv.es/aperles/arm\\_cortex\\_m3/libre/libro-ARM-Cortex-M.pdf](http://www.disca.upv.es/aperles/arm_cortex_m3/libre/libro-ARM-Cortex-M.pdf)
- [24] Broadcom Corporation®, *BCM2835 ARM Peripherals*, [documento PDF en línea], Broadcom Corporation®, [citado 23/10/16], disponible en internet: <https://www.raspberrypi.org/app/uploads/2012/02/BCM2835-ARM-Peripherals.pdf>
- [25] Ganssle Jack, Barr Michael, *Embedded Systems Dictionary*, [citado 11/11/16], 2003, Ed. Elsevier
- [26] Joshi Prateek, Godoy Vinicius, Millan Escrivá David, *OpenCV By Example*, [citado 11/11/16], 2016, Ed. Packt Publishing
- [27] Howse Joseph, *OpenCV Computer Vision With Python*, [citado 30/11/16], 2013, Ed. Packt Publishing
- [28] Solem Jan Erik, *Programming Computer Vision with Python*, [citado 15/12/16], 2012, Ed. O'Reilly
- [29] OpenCV, *OpenCV API reference*, [página web en línea], [citado 20/12/16], disponible en internet: <https://docs.opencv.org/2.4/modules/refman.html>
- [30] OpenCV, *OpenCV-Python Tutorials*, [página web en línea], [citado 29/12/16], disponible en internet: [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_tutorials.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html)
- [31] Rosebrock Adrian, *4 Point OpenCV getPerspective Transform Example*, [artículo en línea], [citado 05/02/17], disponible en internet: <https://www.pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/>
- [32] Shaikh Izhar, *Si-Fo ...The Autonomous Sign Follower*, [blog en línea], [citado 13/01/17], disponible en internet: <https://robosifo.wordpress.com/>
- [33] De La Escalera Arturo, Armingol José María, García Fernando, Gómez David Martín, H. Al-Kaff Abdulla, *Introducción a la visión por computador: desarrollo de aplicaciones con OpenCV*, Universidad Carlos III de Madrid, [curso en línea], [citado 2/02/16], disponible en internet: <https://courses.edx.org/courses/course-v1:UC3Mx+ISA.1x+1T2016/course/>

[34] Cognex, *Introducción a la visión artificial: Una guía para la automatización de procesos y mejoras de calidad*, [publicación en línea] [citado 10/08/16], disponible en internet:

[http://www.ikusmen.com/documentos/descargas/3cbb38\\_Introduction%20to%20Machine%20Vision.pdf](http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf)

[35] Raspberry®, *Buy A Raspberry Pi*, [página web en línea], [citado 15/10/16], disponible en internet: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

[36] Downey Allen, Elkner Jeffrey, Meyers Chris, *Aprenda a pensar como un programador con Python [traducido]*, [citado 15/12/16], 2002, Ed. Green Tea Press

[37] NumPy, *NumPy*, [página web en línea], [citado 15/08/16], disponible en internet: <http://www.numpy.org/>

[38] ARM®, *Processors*, [página web en línea], [citado 8/11/16], disponible en internet: <https://www.arm.com/products/processors>

[39] NXP®, *Kinetis KL25 Sub-Family*, [documento PDF en línea], NXP®, [citado 23/12/16], disponible en internet: <https://www.nxp.com/docs/en/data-sheet/KL25P80M48SF0.pdf>

[40] National Instruments®, *Fundamentos de control de movimiento*, [publicación en línea] [citado 10/08/16], 2016, disponible en internet: <http://www.ni.com/white-paper/3367/es/#toc1>

[41] Costco, [página web en línea], [citado 20/1/17], disponible en internet: [https://www.costco.com.mx/Salud-y-Belleza/Ortopedia-y-Rehabilitacion/Sillas-de-Ruedas-y-Andaderas/c/cos\\_7.7.1](https://www.costco.com.mx/Salud-y-Belleza/Ortopedia-y-Rehabilitacion/Sillas-de-Ruedas-y-Andaderas/c/cos_7.7.1)

[42] Mercado Libre, [página web en línea], [citado 20/1/17], disponible en internet: [https://articulo.mercadolibre.com.mx/MLM-613614979-silla-de-ruedas-electrica-de-gran-capacidad-\\_JM](https://articulo.mercadolibre.com.mx/MLM-613614979-silla-de-ruedas-electrica-de-gran-capacidad-_JM)

[43] ElectroCraft®, *MP36 ElectroCraft MobilePower™*, [documento PDF en línea], ElectroCraft® [citado 15/09/2016], disponible en internet: <https://www.electrocrafter.com/pdf.php?pdf=files/downloads/Datasheets/gearmotors/MP36-DataSheet-US.pdf>

[44] Guías Prácticas.com, *Transceptor*, [página web en línea], [citado 15/5/18], disponible en internet: <http://www.guiaspracticas.com/walkie-talkies/transceptor>

[45] Gallardo Vázquez, Sergio, *Elementos de Sistemas de Telecomunicaciones*, [citado 15/5/18], 2002, Ed. Paraninfo