



UNIVERSIDAD NACIONAL AUTÓNOMA  
DE MÉXICO

---

---

**FACULTAD DE INGENIERÍA**

**RE-DISEÑO DE CONTROL  
PARA LA TRAMPA DE  
SEDIMENTOS S/MT 234**

**TESIS**

Que para obtener el título de:

**INGENIERO MECATRÓNICO**

**PRESENTA:**

Daniel Martínez Cruz

**DIRECTOR DE TESIS:**

M. en C. Arturo Ronquillo Arvizu

Ciudad Universitaria, Cd. Mx., 2016





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

Quiero agradecer a la Universidad Nacional Autónoma de México por abrirme sus puertas, darme una formación profesional y permitirme un desarrollo integral como ingeniero y como ser humano. Mil gracias por otorgarme el honor de pertenecer a una gran familia universitaria.

Agradezco a la Facultad de Ingeniería, por ser la gran institución en la cual adquirí las bases de las diferentes disciplinas que integran a la ingeniería mecatrónica. Gracias, mi amada Facultad, por permitirme encontrar personas muy valiosas; que marcaron mi desarrollo y cuyas enseñanzas atesoraré el resto de mi vida.

Agradezco al Instituto de Ciencias del Mar, particularmente a la Doctora Elva Escobar, directora del instituto; ya que este trabajo no podría haber sido realizado sin su apoyo incondicional, su constante velar por la oficina de instrumentación, así como su apoyo a este trabajo.

# DEDICATORIAS

Quiero agradecer a la Universidad Nacional Autónoma de México por abrirme sus puertas, darme una formación profesional y permitirme un desarrollo integral como ingeniero y como ser humano. Mil gracias por otorgarme el honor de pertenecer a una gran familia universitaria.

Agradezco a la Facultad de Ingeniería, por ser la gran institución en la cual adquirí las bases de las diferentes disciplinas que integran a la ingeniería mecatrónica. Gracias, mi amada Facultad, por permitirme encontrar personas muy valiosas, que marcaron mi desarrollo y cuyas enseñanzas atesoraré el resto de mi vida.

Agradezco al Instituto de Ciencias del Mar, particularmente a la Doctora Elva Escobar, directora del instituto, ya que este trabajo no podría haber sido realizado sin su apoyo incondicional, su constante velar por la oficina de instrumentación, así como su apoyo a este trabajo.

# ÍNDICE GENERAL

Agradecimientos.....	3
Dedicatorias.....	3
Índice general.....	4
Índice de figuras.....	5
Índice de Tablas.....	7
Resumen.....	8
Objetivos.....	9
<b>1 Introducción.....</b>	<b>10</b>
1.1 ANTECEDENTES.....	10
1.2 MUESTREADORES DE SEDIMENTO.....	10
1.2.1 Multicorer.....	11
1.2.2 Dragas.....	11
1.2.3 Trampas de sedimentos.....	12
1.3 ANÁLISIS DE LA TRAMPA DE SEDIMENTOS.....	13
1.3.1 ¿Por qué se utilizan las trampas de sedimento?.....	13
1.3.2 ¿Cómo funcionan las trampas de sedimento?.....	13
1.3.3 ¿Qué se ha aprendido usando las trampas de sedimentos?.....	13
1.3.4 Nieve marina.....	14
1.3.5 Contenido de la nieve marina.....	15
1.3.6 Problemas de muestreo.....	16
1.3.7 Diseños típicos.....	16
1.3.8 ¿Cómo se despliegan las trampas de sedimentos?.....	17
1.3.9 Limitaciones.....	18
1.3.10 Liberadores acústicos.....	18
1.4 TRAMPAS DE SEDIMENTO CON LAS QUE SE CUENTA.....	19
1.5 MICROCONTROLADORES.....	21
1.5.1 ¿Qué es un microcontrolador?.....	21
1.5.2 Aplicaciones de los microcontroladores.....	21
1.6 COMPONENTES DE UN MICROCONTROLADOR.....	22
1.6.1 Unidad Central De Procesamiento (Central Processor Unit - CPU).....	22
1.6.2 Puertos de entrada y salida.....	23
1.6.3 Registros.....	24
1.6.4 Registros SFR.....	24
1.6.5 Memoria RAM (Random Access Memory) - Memoria de acceso aleatorio.....	25
1.6.6 Memoria EEPROM (ELECTRICALLY ERASABLE PROGRAMMABLE ROM).....	26
1.6.7 Memoria ROM.....	27
1.6.8 Memoria flash.....	27
1.7 ATMEL.....	28
1.8 ARDUINO.....	29
1.8.1 ¿Qué es Arduino?.....	29
1.8.2 ¿Por qué emplear Arduino?.....	29
1.8.3 Bibliotecas de Arduino.....	29
1.9 ATMEGA328P.....	30
1.9.1 RISC.....	30
1.9.2 Características del ATMEGA328P.....	30
1.10 RELOJ DE TIEMPO REAL (RTC).....	31
1.11 RTC PCF8583P.....	32
<b>2 Planteamiento del problema.....</b>	<b>40</b>
2.1 ANÁLISIS DEL FUNCIONAMIENTO DEL CIRCUITO DE CONTROL ORIGINAL.....	40
2.1.1 Tarjeta 1.....	41

2.1.2 Tarjeta 2.....	42
2.1.3 Tarjeta 3.....	43
2.1.4 Tarjeta 4.....	44
2.1.5 Tarjeta 5.....	45
2.2 ANÁLISIS DEL FUNCIONAMIENTO DEL PROGRAMA DE CONTROL ORIGINAL.....	46
2.2.1 Puesta en funcionamiento del software.....	46
2.2.2 Ejecución del software original.....	48
<b>3 Desarrollo.....</b>	<b>52</b>
3.1 PROPUESTA DE DISEÑO DEL CIRCUITO.....	52
3.1.1 Control de motor de CD.....	52
3.1.2 Circuito de RTC.....	53
3.1.3 Conexión RS232.....	54
3.1.4 Circuito de control/almacenamiento de datos.....	55
3.2 PROPUESTA DE DISEÑO DEL PROGRAMA.....	58
3.2.1 Descripción del funcionamiento.....	58
3.3 ANÁLISIS DE BATERÍAS.....	64
3.3.1 Vida de las baterías y consumo de corriente.....	64
3.3.2 Consumo de corriente teórico.....	65
3.4 PROGRAMACIÓN DEL MICROCONTROLADOR.....	67
3.4.1 Componentes del programador.....	68
3.4.2 Conector de ISP en el circuito.....	69
3.4.3 Programación del circuito.....	69
<b>4 Resultados.....</b>	<b>74</b>
4.1 CÁLCULOS DE CONSUMO ELÉCTRICO.....	74
4.2 CIRCUITO RE-DISEÑADO IMPLEMENTADO.....	75
<b>Conclusiones.....</b>	<b>78</b>
<b>Recomendaciones.....</b>	<b>79</b>
<b>Bibliografía.....</b>	<b>80</b>
<b>Anexos.....</b>	<b>83</b>
ANEXO 1 FOTOGRAFÍAS DEL CIRCUITO ORIGINAL.....	83
ANEXO 2 INSTALACIÓN DEL PROGRAMA.....	86
ANEXO 3 PROBLEMAS EN INSTALACIÓN DE ADAPTADOR USB-SERIAL.....	88
ANEXO 4 CÓDIGO FUENTE DE LA INTERFAZ GRÁFICA Y DEL MICROCONTROLADOR.....	91
ANEXO 5 DISEÑO ESQUEMÁTICO DEL CIRCUITO.....	92
ANEXO 6 DISEÑO DE PCB.....	94
ANEXO 7 PROGRAMADOR DE AVR SPARKFUN.....	95
ANEXO 8 DATASHEETS.....	97

## ÍNDICE DE FIGURAS

FIGURA 1 TARJETA DEL CIRCUITO DE CONTROL SIN DAÑO (IZQUIERDA) TARJETA DEL CIRCUITO DE CONTROL DAÑADA POR EL INGRESO DE AGUA (DERECHA).....	9
FIGURA 2 TUBOS DE MULTICORER CON MUESTRA DE SEDIMENTOS.....	11
FIGURA 3 - DRAGA.....	12
FIGURA 4 - TRAMPA DE SEDIMENTOS, DE LA COMPAÑÍA KC.....	14
FIGURA 5 - EJEMPLOS DE NIEVE MARINA.....	16
FIGURA 6 - DIFERENTES TIPOS DE TRAMPAS DE SEDIMENTOS.....	17
FIGURA 7 - EQUIPO PARA COMUNICACIÓN CON LIBERADORES ACÚSTICOS, MARCA EDGETECH.....	19
FIGURA 8 - LIBERADOR ACÚSTICO, MARCA EDGETECH.....	19

FIGURA 9 - RECUPERACIÓN DE TRAMPA DE SEDIMENTOS, CAMPAÑA CANEK-10 (INVESTIGADOR RESPONSABLE: DR. JULIO CANDELA) NOVIEMBRE DE 2005 B/O JUSTO SIERRA.....	21
FIGURA 10 - TRAMPA DE SEDIMENTOS EN CUBIERTA, CAMPAÑA CANEK-10, NOVIEMBRE 2005, B/O JUSTO SIERRA.....	21
FIGURA 11 - IMAGEN QUE MUESTRA LOS COMPONENTES INTERNOS DE UN MICROCONTROLADOR.....	22
FIGURA 12 - CIRCUITOS QUE COMPONEN A LA CPU.....	23
FIGURA 13 - PUERTOS DEL MICROCONTROLADOR.....	24
FIGURA 14 - REGISTROS DEL MICROCONTROLADOR.....	25
FIGURA 15 - REGISTROS DE FUNCIONES ESPECIALES.....	25
FIGURA 16 - CELDA MEMORIA FLASH, 0 LÓGICO.....	28
FIGURA 17 - CELDA MEMORIA FLASH, 1 LÓGICO.....	28
FIGURA 18 - DIAGRAMA DE BLOQUES DE ATMEGA328P.....	31
FIGURA 19 - DIAGRAMA DE BLOQUES PCF8583P, MUESTRA EL CONTENIDO DE LOS DIFERENTES REGISTROS.....	32
FIGURA 20 - CONTENIDO DEL REGISTRO DE STATUS/CONTROL.....	33
FIGURA 21 - CONTENIDO DE LOS REGISTROS 0x00 A 0x06.....	34
FIGURA 22 - REGISTRO DE HORA.....	36
FIGURA 23 - CONTENIDO DEL REGISTRO DE FECHA / AÑO.....	37
FIGURA 24 - CONTENIDO DEL REGISTRO DE NÚMERO DE DÍA/NÚMERO DE MES.....	37
FIGURA 25 - CONTENIDO DEL REGISTRO DE LA ALARMA.....	39
FIGURA 26 - TARJETAS QUE COMPONEN AL CIRCUITO DE CONTROL.....	41
FIGURA 27 - TARJETA 1.....	41
FIGURA 28 - TARJETA 2.....	42
FIGURA 29 - TARJETA 3.....	43
FIGURA 30 - TARJETA 4.....	44
FIGURA 31 - TARJETA 5.....	45
FIGURA 32 - CREACIÓN DE ACCESO DIRECTO.....	46
FIGURA 33 - SELECCIÓN DE RUTA PARA CREACIÓN DE ACCESO DIRECTO.....	47
FIGURA 34 - RUTA DE ACCESO A DOSBox.....	47
FIGURA 35 - PANTALLA DE BIENVENIDA.....	48
FIGURA 36 - MENÚ DE CONFIGURACIÓN.....	49
FIGURA 37 - ELECCIÓN DEL NÚMERO DE MUESTRAS.....	49
FIGURA 38 - DATOS DE LANZAMIENTO.....	50
FIGURA 39 - ELECCIÓN DE FECHAS PARA LA MUESTRA.....	50
FIGURA 40 - TRAMPA DE SEDIMENTOS TRATANDO DE CONECTARSE AL PROGRAMA.....	50
FIGURA 41 - ERROR EN LA CONEXIÓN.....	51
FIGURA 42 - MENÚ BASE DE DATOS.....	51
FIGURA 43 - MENÚ DE UTILIDADES.....	51
FIGURA 44 - CIRCUITO L298N CON DISIPADOR DE CALOR.....	52
FIGURA 45 - CIRCUITO DE CONTROL DEL MOTOR L298N.....	53
FIGURA 46 - CIRCUITO DE CONTROL RTC PCF8583P.....	54
FIGURA 47 - CIRCUITO RS232.....	55
FIGURA 48 - CIRCUITO DE CONTROL DISEÑADO.....	57
FIGURA 49 - NUEVO SOFTWARE PARA PROGRAMAR LA TRAMPA DE SEDIMENTOS.....	58
FIGURA 50 - BÚSQUEDA DE PUERTOS.....	59
FIGURA 51 - CONEXIÓN AL PUERTO COM8.....	60
FIGURA 52 - CONEXIÓN AL PROGRAMA.....	60
FIGURA 53 - CONEXIÓN EXITOSA DEL CIRCUITO DE LA TRAMPA.....	60
FIGURA 54 - ELECCIÓN DE FECHAS.....	60
FIGURA 55 - CASILLA DE VERIFICACIÓN DE LA FECHA.....	61
FIGURA 56 - ALMACENANDO LAS FECHAS EN LA TRAMPA.....	61
FIGURA 57 - PROGRESO EN EL ALMACENAMIENTO DE LAS FECHAS.....	62

FIGURA 58 - VERIFICACIÓN DE FECHAS.....	62
FIGURA 59 - LISTA CON LAS FECHAS LEÍDAS DIRECTAMENTE DE LA TRAMPA DE SEDIMENTOS.....	63
FIGURA 60 - BOTÓN PARA GUARDAR LAS FECHAS EN UN ARCHIVO DE TEXTO.....	63
FIGURA 61 - CREACIÓN DE ARCHIVO DE TEXTO.....	64
FIGURA 62 - ARCHIVO DE TEXTO GENERADO.....	64
FIGURA 63 - GRÁFICA DE DESCARGA DURACELL, 250mA Y 500mA.....	65
FIGURA 64 - GRÁFICA DE DESCARGA DURACELL A 250mA, A 0°C Y A 21°C.....	65
FIGURA 65 - CORRIENTE DE REPOSO MC7805CTG.....	66
FIGURA 66 - CONSUMO DE CORRIENTE A DIFERENTES VOLTAJES Y CORRIENTES.....	67
FIGURA 67 - CONSUMO DE CORRIENTE EN REPOSO DE L298.....	67
FIGURA 68 - PROGRAMADOR ISP MARCA SPARKFUN. FIGURA 69 - COMPONENTES DE PROGRAMADOR ISP.....	68
FIGURA 70 - CONECTORES DE ISP.....	68
FIGURA 71 - UBICACIÓN DE HEADER ISP.....	69
FIGURA 72 - CONFIGURACIÓN PARA PROGRAMACIÓN ISP.....	70
FIGURA 73 - CARGA DEL CÓDIGO VÍA ISP.....	70
FIGURA 74 - PROGRAMACIÓN VÍA CMD.....	70
FIGURA 75 - CÓDIGO CARGADO VÍA CONSOLA DE COMANDOS.....	71
FIGURA 76 - COMPILACIÓN MEDIANTE ATMEL STUDIO.....	72
FIGURA 77 - COMPILACIÓN CORRECTA.....	72
FIGURA 78 - CONFIGURACIÓN DE PROGRAMADOR COMO HERRAMIENTA EXTERNA.....	72
FIGURA 79 - AÑADIR HERRAMIENTA EXTERNA.....	73
FIGURA 80 - PARÁMETROS DE HERRAMIENTA EXTERNA.....	73
FIGURA 81 - CIRCUITO DE CONTROL ORIGINAL.....	76
FIGURA 82 - REDUCCIÓN EN EL TAMAÑO CON EL CIRCUITO DISEÑADO.....	76
FIGURA 83 - TARJETA PCB, CARA SUPERIOR.....	77
FIGURA 84 - TARJETA PCB, CARA INFERIOR.....	77
FIGURA 85 - PCB CON TODOS LOS COMPONENTES SOLDADOS.....	77
FIGURA 86 - COMPARACIÓN DE TAMAÑO ENTRE PCB Y BACKPLANE ORIGINAL.....	77

## ÍNDICE DE TABLAS

TABLA 1 - ESPECIFICACIONES DE LA TRAMPA DE SEDIMENTOS S/MT-234.....	20
TABLA 2 - CONTENIDO DE UNA CELDA DE LA EEPROM.....	26
TABLA 3 - REPRESENTACIÓN DE NÚMEROS DECIMALES EN FORMATO BCD.....	35
TABLA 4 - EJEMPLO DE CODIFICACIÓN BCD, NÚMERO 95.....	35
TABLA 5 - EJEMPLO DE CODIFICACIÓN BCD, NÚMERO 37.....	35
TABLA 6 - NÚMERO DE DÍAS PROGRAMABLES DE ACUERDO AL MES.....	36
TABLA 7 - CICLO DE CONTEO DE MESES Y DÍA DE LA SEMANA.....	37
TABLA 8 - CONTENIDO DE LA EEPROM.....	56
TABLA 9 - DIFERENTES ESCENARIOS DE CONSUMO. ....	74
TABLA 10 - CÁLCULO DE TIEMPO DE FUNCIONAMIENTO PROMEDIO.....	75



## RESUMEN.

El Instituto de Ciencias del Mar y Limnología cuenta con tres trampas de sedimentos de la marca KUM, modelo S/MT 234; sin embargo, debido a una falla que permitió el ingreso de agua a los cilindros que contienen a la electrónica, estas trampas no se encuentran funcionales (Figura 1).

Para poner nuevamente en funcionamiento las trampas de sedimentos se tendrá que re-diseñar el circuito de control, así como el programa por medio del cual se realizará la configuración de las mismas; esto debido a que la documentación con la que se cuenta no es suficiente para permitir diagnosticar la falla; además de eso, el software de control y la electrónica tienen casi 20 años, por lo cual se encuentran obsoletos.

Las trampas de sedimentos son instrumentos que utilizan los investigadores para recolectar muestras de sedimento con el fin de determinar sus propiedades físicas y químicas. Analizar estas muestras ayuda a comprender cómo los nutrientes y oligoelementos se mueven desde el océano poco profundo hasta las profundidades del mismo.

## OBJETIVOS.

- Se diseñará un circuito electrónico que servirá de reemplazo para las trampas de sedimento modelo S/MT 234.
- Se realizará un programa a través del cual se pueda configurar la trampa de sedimentos modelo S/MT 234 para su posterior uso. El programa se podrá emplear en equipos con una versión de Windows XP o superior.
- Se verificará el funcionamiento de la trampa al emplear el circuito de control diseñado.
- Se realizarán los cálculos de consumo de corriente, así como de la descarga de las baterías para garantizar el funcionamiento de la trampa de sedimentos.

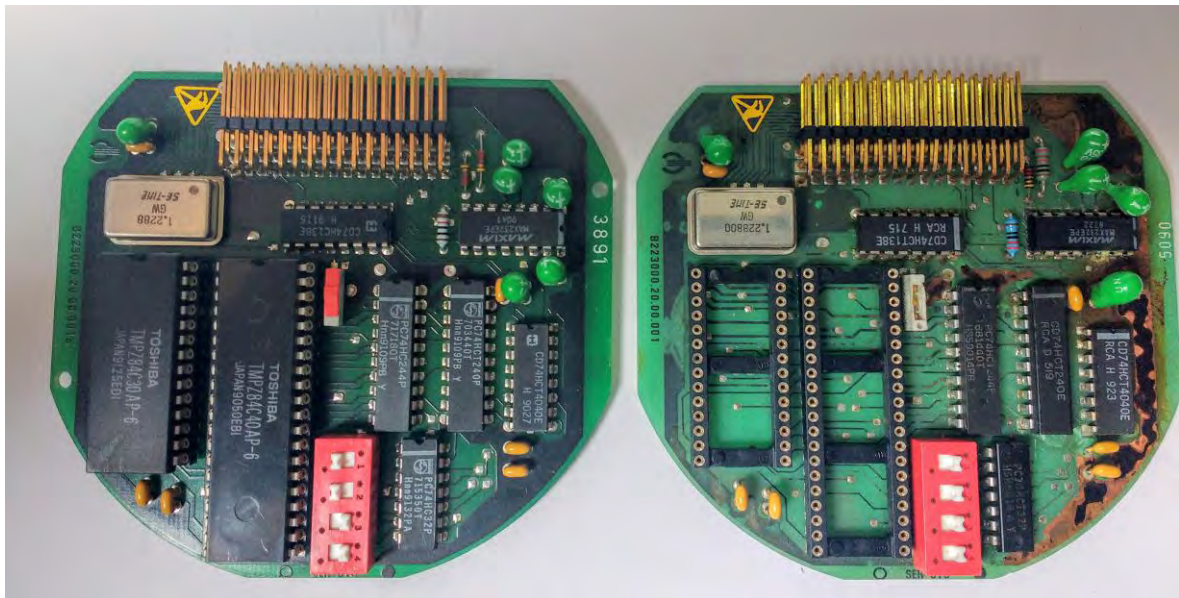


Figura 1 Tarjeta del circuito de control sin daño (izquierda). Tarjeta del circuito de control dañada por el ingreso de agua (derecha).

# 1 INTRODUCCIÓN

---

## 1.1 ANTECEDENTES.

El hundimiento de materia orgánica en el mar es un fenómeno que afecta de gran manera a la demanda metabólica marina (Fowler y Knauer, 1986). La sedimentación de materia orgánica de la capa superficial del océano está íntimamente ligada con eventos biológicos de la zona fótica<sup>1</sup> (Silver, MW, y MM Gowing, 1991). En la capa superficial del mar, el carbono inorgánico disuelto es incorporado a la cadena alimenticia marina por el fitoplancton; posteriormente es transformado, principalmente por el zooplancton y por otros procesos no biológicos, en agregados mayores de materia orgánica particulada, en forma de agregados mucosos y pelotillas fecales (Fowler y Knauer, 1986). Esta materia muerta transporta además otros componentes, tales como esqueletos de organismos planctónicos, minerales provenientes de la superficie terrestre y posiblemente contaminantes. Debido al pequeño tamaño de las partículas las técnicas usadas comúnmente para recolectar materia orgánica en suspensión no son adecuadas para un muestreo idóneo; en consecuencia, el uso de las trampas de sedimentos ha proliferado en las últimas 2 a 3 décadas como un medio para obtener información cuantitativa y cualitativa del flujo de partículas en hundimiento.

En particular, los estudios de series de tiempo han revelado patrones de sedimentación tales como máximos flujos asociados con aportes terrígenos, estimulación de la productividad por el aporte de cenizas volcánicas ricas en hierro, y brotes estacionales de aguas ricas en nutrientes (Thunell 1998, Wong et al. 1999, Takahashi et al. 2000).

## 1.2 MUESTREADORES DE SEDIMENTO.

Los muestreadores de sedimentos se utilizan para recoger muestras con el fin de determinar sus propiedades; su objetivo principal es determinar la cantidad de

---

<sup>1</sup> La zona fótica (*euphotic zone*) es la profundidad marina en la que la intensidad de la luz aún es suficiente para permitir la fotosíntesis producida por el fitoplancton y plantas. *Photoc zone - Encyclopædia Britannica Online.*

sólidos en suspensión o sedimento contenido en una muestra, así como su tamaño y composición química y física. Algunos de los equipos disponibles para el muestreo de sedimentos son las trampas de sedimentos, dragas, multicorers, etc.

### 1.2.1 Multicorer.

El *Multicorer* está diseñado para obtener muestras múltiples de sedimento de los fondos marinos a grandes profundidades. No tiene conexiones eléctricas ni requiere una configuración por computadora; funciona simplemente por acción de la fuerza hidrostática y de mecanismos de alta precisión. Se compone de varios tubos transparentes con un diámetro de 6 a 10 cm, al llegar al fondo de la superficie marina se dispara la carga de un resorte, cerrando el tubo para preservar la muestra hasta que el multicorer sea sacado del agua. Se muestra un multicorer en la Figura 2.



*Figura 2 Tubos de multicorer<sup>2</sup> con muestra de sedimentos.*

### 1.2.2 Dragas.

Las *dragas* son muestreadores de sedimentos utilizados para recoger muestras del fondo marino. Las mandíbulas se colocan abiertas, sujetas por medio de un gancho, y posterior a esto se hacen descender lentamente en el agua; tan pronto como toca el fondo, el gancho se suelta y las mandíbulas se cierran, quedando dentro la muestra. Se muestra en la Figura 3 una draga.

<sup>2</sup> Imagen tomada de: <http://www.sidmar.es/Multicorer.html>

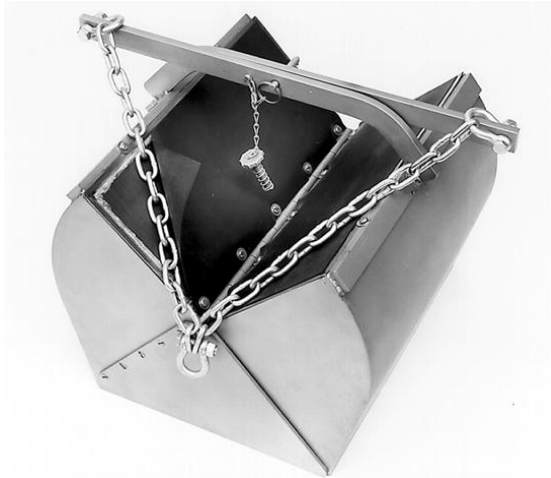


Figura 3 - Draga.<sup>3</sup>

### 1.2.3 Trampas de sedimentos.

Las trampas de sedimento son contenedores que los científicos ponen en el agua para recoger partículas que caen hacia el fondo del mar. Las trampas recogen pequeños cúmulos de granos de sedimentos o escombros más grandes, también llamada nieve marina; compuesta de materia orgánica, plancton muerto, pequeñas conchas, polvo y minerales.

Analizar estas muestras ayuda a comprender cómo los nutrientes y oligoelementos<sup>4</sup> como el carbono, el nitrógeno, fósforo, calcio, silicio, y el uranio se mueven desde el océano poco profundo hasta las profundidades del mismo. Estos materiales son también los que utiliza casi toda la vida en alta mar como alimento (ya que las plantas no pueden crecer en la oscuridad). Otros investigadores analizan el rastro de los elementos en las conchas para obtener pistas sobre la circulación oceánica hace miles de años.

---

<sup>3</sup> Imagen tomada de: <http://www.rickly.com/as/bottomgrab.htm>

<sup>4</sup> Se conocen como *oligoelementos* a las sustancias químicas que se encuentran en pequeñas cantidades en el organismo para intervenir en su metabolismo. Estos son esenciales para el buen funcionamiento de los organismos.

## **1.3 ANÁLISIS DE LA TRAMPA DE SEDIMENTOS.**

### **1.3.1 ¿Por qué se utilizan las trampas de sedimento?**

La principal característica por la cual son utilizadas las trampas de sedimento es porque permiten realizar mediciones a través del tiempo, con lo que es posible obtener una tasa de sedimentación; es decir, la velocidad con la que cae el sedimento al fondo del mar.

### **1.3.2 ¿Cómo funcionan las trampas de sedimento?**

La trampa de sedimentos básica se compone de un amplio embudo con un frasco de recogida, en la cima cuenta con una rejilla para evitar que objetos muy grandes puedan tapar el embudo. Cualquier cosa que cae en el embudo se dirige en el frasco de recogida en la parte inferior; en el caso de trampas de aguas profundas, que a menudo se quedan en el agua durante periodos de alrededor de un año, tienen varios frascos en una bandeja circular motorizada. Después de un período determinado de tiempo, o si otros instrumentos en la trampa detectan un cambio en las condiciones del agua, un nuevo frasco de recogida gira en su lugar y se sella el frasco anterior.

### **1.3.3 ¿Qué se ha aprendido usando las trampas de sedimentos?**

Antes de la utilización de las trampas de sedimentos, los científicos asumieron que los nutrientes y los pequeños organismos del plancton se hundían muy lentamente, tomando siglos o milenios para finalmente llegar al fondo del mar. Se pensaba que durante el largo descenso, gran parte del material se disolvía en el agua o era consumido por la vida marina de manera que nunca llegaba al fondo. Se convirtió en un acertijo el poder entender cómo podría sobrevivir la vida en el fondo del mar con tan pocos nutrientes.

Las trampas de sedimento mostraron que las partículas mucho más grandes (de 12 mm [0.5 pulgadas] de largo), sí llegaban a la parte inferior. Esta nieve marina (llamada así debido a su color y textura esponjosa) se hunde a una velocidad de hasta 200 metros por día y puede llegar al fondo en tan sólo unas semanas. A esa velocidad, las condiciones de la superficie cambiante pueden

afectar el fondo del mar con relativa rapidez en la forma de una ventisca de nieve marina. Se muestra una trampa de sedimento en la Figura 4.



*Figura 4 - Trampa de Sedimentos, de la compañía KC.*

#### **1.3.4 Nieve marina.**

Gran parte de la materia derivada de la superficie que entra al océano, viaja al fondo marino como nieve marina.

Las autoras Alice Alldredge y Mary Silver definen a la nieve marina como el agregado de partículas en el océano con un tamaño mayor a 500 micrómetros (Alldredge, AL, and MW Silver. 1988).

Como microhabitat, la nieve marina alberga comunidades microbiales enriquecidas y gradientes químicos. En este microhabitat los procesos de fotosíntesis, descomposición, y regeneración de nutrientes se producen a muy elevados niveles. Estas partículas tienen un rango de tamaño desde micrones a varios centímetros de diámetro

Las plantas y los animales muertos no son los únicos componentes de la nieve marina; también incluye materia fecal, arena, hollín y otro polvo inorgánico. Esta nieve va creciendo a medida que va cayendo y en ocasiones alcanza varios

centímetros. Esta lluvia continua de nieve marina provee alimento para muchas criaturas del fondo del mar.

Los avances en el entendimiento de las características y abundancia de nieve marina han sido limitados por problemas asociados con la toma de muestras y la cuantificación de estos frágiles agregados en la naturaleza (*Allredge, AL, y MW Silver. 1988*). Históricamente los estudios de sedimentación se habían desarrollado de dos formas distintas; una de ellas es la observación *in situ*, ejemplificada por investigadores japoneses que en los años 50 captaron acumulados visibles desde sumergibles tanto en aguas costeras como en aguas profundas. Los investigadores Suzuki y Kato en 1954 fotografiaban las muestras mientras caían. El segundo método de investigación, empleado por investigadores norteamericanos, consistía en estudiar las partículas obtenidas de botellas de agua; estos investigadores examinaban los agregados que se quedaban atrapados en filtros que se colocaban en las botellas, definiendo estas partículas como “agregados orgánicos”, hojuelas y escombros de los organismos. Posteriormente en los años 70s y 80s ambas técnicas fueron combinadas con lo cual se determinó la abundancia de la nieve marina en el agua poco profunda; esto al realizar la investigación *in situ*, así como con la recolección de partículas frágiles, permitiendo las primeras mediciones cuantitativas del contenido en aguas poco profundas. (*Allredge, AL, and MW Silver. 1988*).

### **1.3.5 Contenido de la nieve marina.**

La nieve marina es una categoría genérica en la cual se encuentran englobados diferentes agregados con diversas morfologías y características. Sin embargo, las características varían; pueden ser frágiles, porosas o con alta cohesión. Muchos agregados contienen matrices de mucosa en las cuales se encuentran incrustados, materia orgánica y heces fecales. Se muestra en la Figura 5 un ejemplo del contenido de la nieve marina, conteniendo diferentes agregados de sedimentos, así como restos fecales de macro crustáceos.



Examples of marine snow. (a) loosely associated aggregate formed by the flocculation of living, chain-forming diatoms. Scale = 1cm; (b) abandoned filter net or "house" of an appendicularian identifiable by the presence of 2 incurrent filters and a large, U-shaped internal filter surrounded by an envelope of particle-studded mucus; (c) typical comet-shaped aggregate; (d) irregularly-shaped aggregate of unknown origin containing numerous macrocrustacean fecal pellets. Scale of b-d = 1mm.

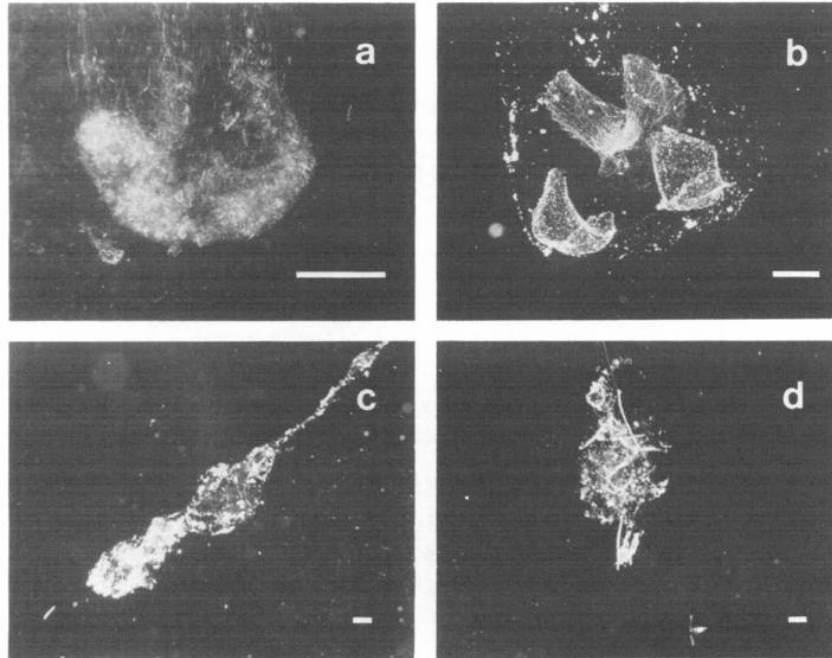


Figura 5 - Ejemplos de nieve marina.<sup>5</sup>

### 1.3.6 Problemas de muestreo.

La existencia de nieve marina había pasado inadvertida hasta la llegada de la observación submarina directa, ya que la nieve marina es frágil y los métodos típicos de recolección causan su fractura al ascender a la superficie. (Aldredge, AL, y MW Silver. 1988). Los efectos combinados de estos problemas derivan en subestimar la cantidad de nieve marina.

### 1.3.7 Diseños típicos.

Existen tres diseños típicos de trampas de sedimento:

- *De superficie* (atada o *tethered*): Este tipo de trampa se utiliza para atrapar el sedimento que se encuentra en la superficie marina.

<sup>5</sup> Tomado de Aldredge, AL, and MW Silver. "Characteristics, Dynamics and Significance of Marine Snow." Progress in Oceanography (1988).

- *Flotando libremente*: Se utiliza para profundidades medias.
- *Aterrada al fondo*: Se utiliza en mar profundo; para recoger los sedimentos utiliza un cono de gran tamaño que capta la mayor cantidad posible de elementos.

En la Figura 6 se muestran los diferentes tipos de trampas de sedimentos.

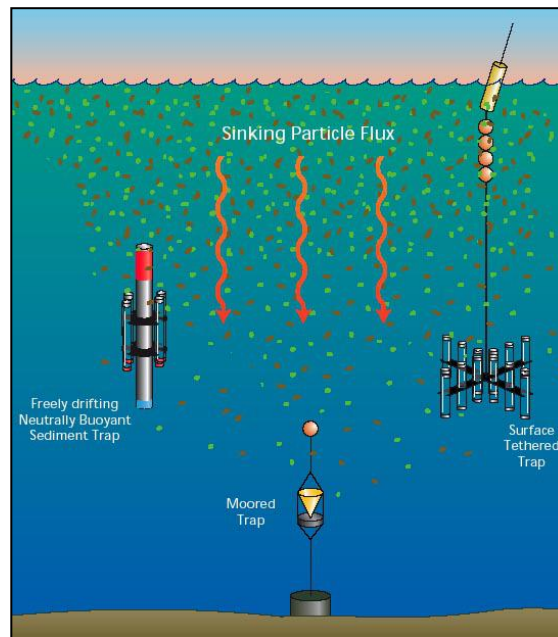


Figura 6 - Diferentes tipos de trampas de sedimentos.<sup>6</sup>

### 1.3.8 ¿Cómo se despliegan las trampas de sedimentos?

Las trampas de sedimento generalmente se montan en un cable que está anclado al fondo del mar y son levantadas por una boya en la superficie o semi sumergida. Las trampas se atornillan en el cable a profundidades específicas, donde permanecen hasta por un año antes de que un barco de investigación vuelva a recoger las muestras. Cuando un barco vuelve a recuperar la trampa, la tripulación activa un dispositivo de control remoto llamado liberador acústico. La liberación separa la línea entre la trampa y su anclaje, y la trampa flota a la superficie.

<sup>6</sup> Tomada de <http://www.whoi.edu/instruments/gallery.do?mainid=19735&iid=10286>

### **1.3.9 Limitaciones.**

Las trampas de sedimentos tienen que permanecer de forma vertical en el agua para que funcione correctamente; por lo cual, las trampas desplegadas en corrientes fuertes deben registrar continuamente su ángulo de inclinación para que los científicos pueden saber si las muestras se han comprometido.

En aguas profundas toma mucho tiempo recolectar suficientes sedimentos para una muestra; especialmente porque cuando se recupera la trampa de sedimentos, científicos de una gran variedad de disciplinas están esperando por una parte de la muestra.

Las muestras pueden sesgarse a sí mismas. La comida en el fondo del mar es muy poca, por lo cual un frasco de recolección lleno de sedimentos es un buffet para el zooplancton local. En algunas trampas la mayor parte de la muestra viene de “los nadadores” en busca de comida, y no directamente del sedimento. Los diseñadores de trampas están trabajando en formas de mantener a los organismos vivos fuera de las muestras.

### **1.3.10 Liberadores acústicos.**

Un liberador acústico<sup>7</sup> (en inglés, *acoustic release*) es un dispositivo oceanográfico para el despliegue y la recuperación posterior de la instrumentación del fondo del mar, en el que el sistema de recuperación se activa de forma remota mediante una señal de comando acústica. Cuando el periodo de medición termina, el barco regresa a la posición exacta donde fue liberado el equipo. Ya en este punto, los técnicos emplean un transductor<sup>8</sup>, el cual se comunica con el liberador acústico, accionándolo y liberando la carga anclada a él.

En la Figura 7 se muestra el equipo que se utiliza para la enviar la señal de comando acústica; en la Figura 8 se muestra un liberador acústico.

---

<sup>7</sup> <http://cascoantiguoceanografia.com/categoria-producto/boyas-y-fondeo/liberadores-acusticos/>

<sup>8</sup> <http://www.edgetech.com/products/acoustic-releases/model-8011m-acoustic-transceiver/#productTab4>



Figura 7 - Equipo para comunicación con liberadores acústicos, marca Edgetech.<sup>9</sup>



Figura 8 - Liberador acústico, marca EdgeTech.<sup>10</sup>

#### **1.4 TRAMPAS DE SEDIMENTO CON LAS QUE SE CUENTA.**

Las trampas de sedimento con las que cuenta el Instituto de Ciencias del Mar son de la marca **K.U.M.**, una empresa alemana que se encuentra en la Bahía de Kiel.

El modelo de la trampa de sedimentos es el **S/MT 234**.

De acuerdo a su hoja de especificaciones<sup>11</sup>, se tienen los siguientes datos:

---

<sup>9</sup> Imagen tomada de: <http://www.edgetech.com/products/acoustic-releases/model-8011m-acoustic-transceiver/#productTab4>

<sup>10</sup> Imagen tomada de: <http://www.edgetech.com/products/acoustic-releases/model-8242xs-acoustic-release-transponder/>

<sup>11</sup> [http://www.kum.homepage.t-online.de/eng/sediment\\_trap.pdf](http://www.kum.homepage.t-online.de/eng/sediment_trap.pdf)

<b>Modelo</b>	<b>S/MT 234</b>
<b>Área de recolección</b>	0.5m <sup>2</sup>
<b>Número de botellas de muestra</b>	21
<b>Capacidad de cada botella</b>	400ml (14oz)
<b>Altura del cono</b>	1313mm
<b>Diámetro del cono</b>	800mm
<b>Altura total</b>	1900mm
<b>Ancho</b>	1079mm
<b>Profundidad de operación</b>	Hasta 6000m
<b>Tiempo de operación</b>	12 meses con baterías nuevas
<b>Fuente de alimentación</b>	9V DC, 6 pilas Alcalinas 1.5V (Tipo D /LR20)
<b>Peso</b>	Aproximadamente 115kg

*Tabla 1 - Especificaciones de la trampa de sedimentos S/MT-234*

En las Figuras 9 y 10 se pueden ver fotografías de las trampas de sedimentos:

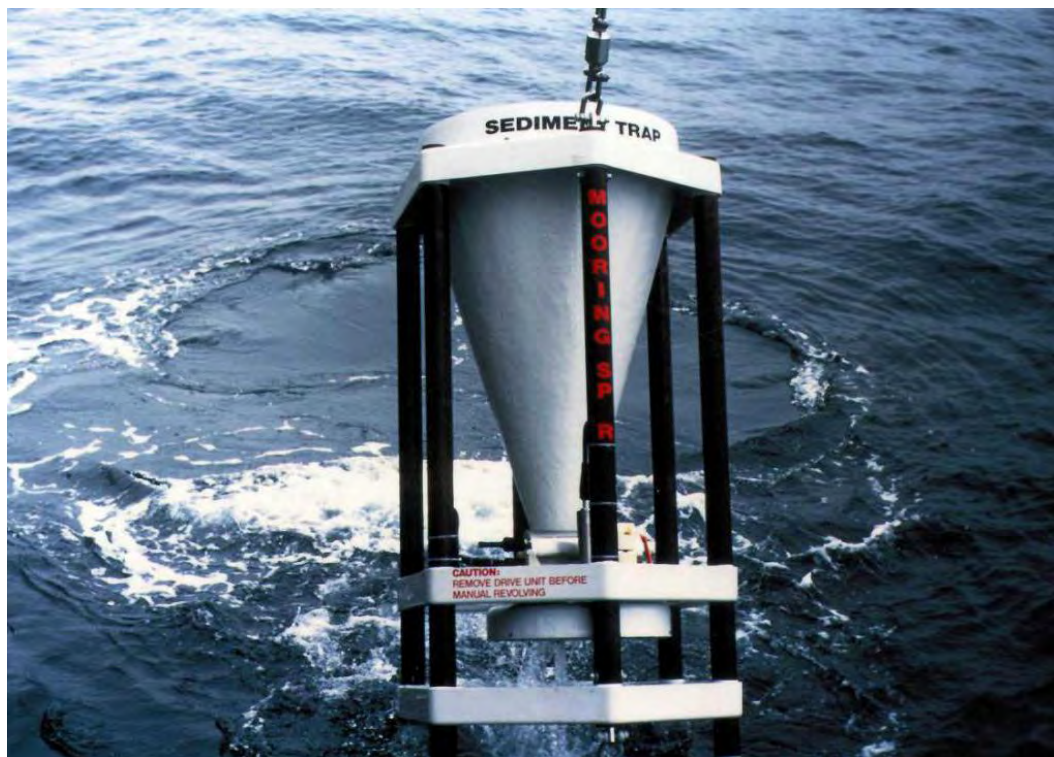


Figura 9 - Recuperación de trampa de sedimentos, campaña CANEK-10 (Investigador responsable: Dr. Julio Candela) Noviembre de 2005 B/O Justo Sierra.

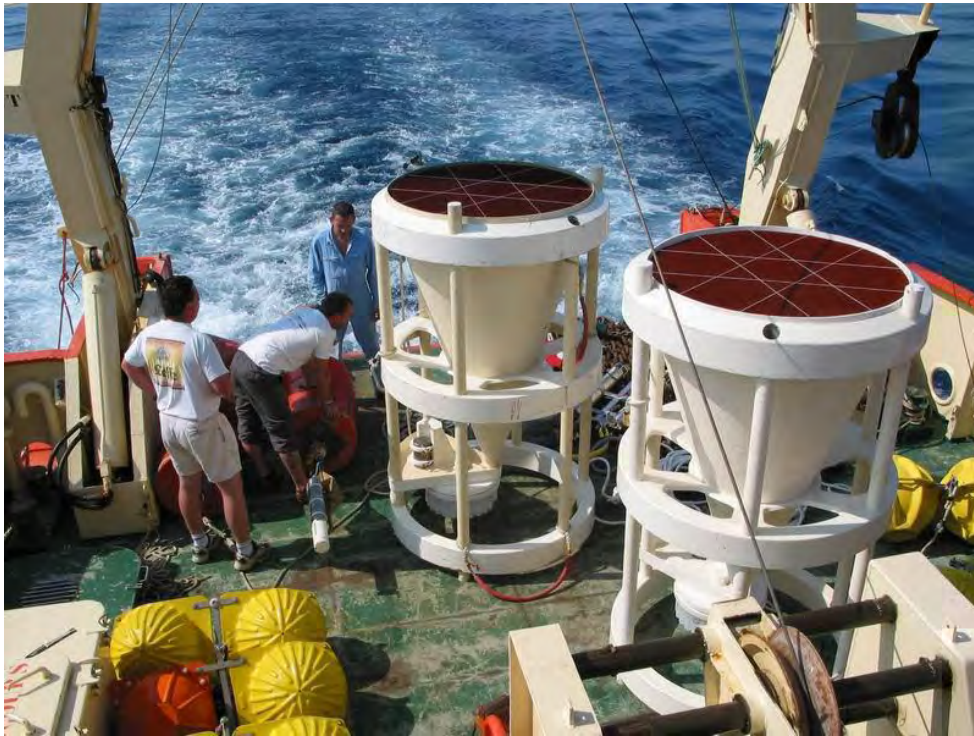


Figura 10 - Trampa de sedimentos en cubierta, campaña CANEK-10, Noviembre 2005, B/O Justo Sierra.

## **1.5 MICROCONTROLADORES.**

### **1.5.1 ¿Qué es un microcontrolador?**

Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento (*CPU*), unidades de memoria (*RAM* y *ROM*), puertos de entrada y salida, así como otros periféricos. En la Figura 11 se muestran los componentes principales de un microcontrolador, los cuales serán descritos posteriormente.

### **1.5.2 Aplicaciones de los microcontroladores.**

Los microcontroladores se pueden encontrar en una amplia gama de productos y soluciones, su aplicación solo está limitada por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y la automatización, en la industria del entretenimiento, en las telecomunicaciones, en la instrumentación, en el hogar, en la industria automotriz, etc.

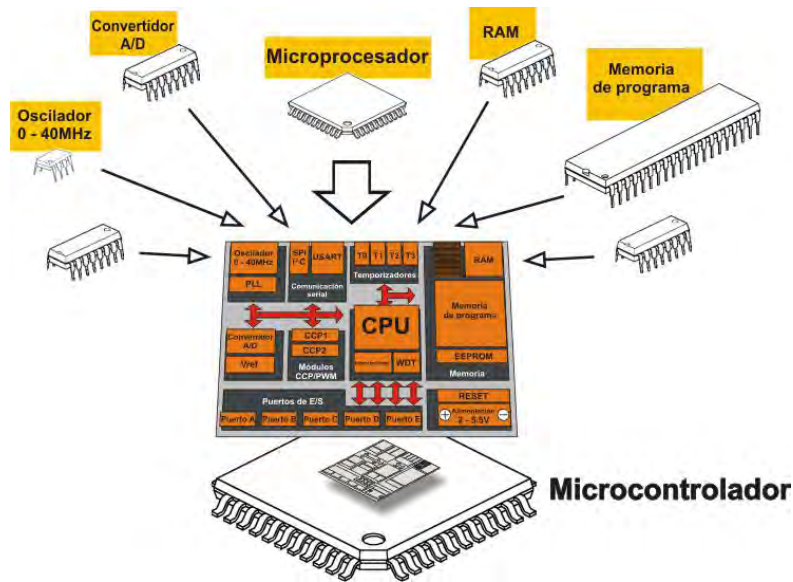


Figura 11 - Imagen que muestra los componentes internos de un microcontrolador.<sup>12</sup>

## 1.6 COMPONENTES DE UN MICROCONTROLADOR.

Los principales componentes de un microcontrolador son:

- Unidad Central de Procesamiento (CPU).
- Puertos de entrada y salida.
- Registros.
- Registros *SFR*.
- Memoria *RAM*.
- Memoria *EEPROM*.
- Memoria *ROM / In-System Reprogrammable Flash Program Memory*.
- Memoria flash.

### 1.6.1 Unidad Central De Procesamiento (Central Processor Unit - CPU).

Como indica su nombre, esto es una unidad que controla todos los procesos dentro del microcontrolador. Consiste en varias unidades más pequeñas, las más importantes son:

- **Decodificador de instrucciones:** es la parte que decodifica las instrucciones del programa accionando otros circuitos con base en estas instrucciones. El “conjunto de instrucciones” que es diferente para cada familia de microcontrolador expresa las capacidades de este circuito.

<sup>12</sup> Imagen tomada de <http://learn.mikroe.com/ebooks/microcontroladorespicc/chapter/introduccion-al-mundo-de-los-microcontroladores>

- **Unidad lógica aritmética (*Arithmetical Logical Unit - ALU*):** realiza todas las operaciones matemáticas y lógicas sobre datos. En la Figura 12 se muestran los componentes del CPU.
- **Registro de trabajo:** También llamado acumulador. Es un registro de función especial que se encuentra estrechamente relacionado con el funcionamiento de la *ALU*. Se utiliza para almacenar todos los datos sobre los que se debe realizar alguna operación lógica o matemática (sumar, mover); de igual manera almacena resultados preparados para un procesamiento futuro. Uno de los registros de función especial, denominado *Registro de Estado* (o registro PSW), está estrechamente relacionado con el registro de trabajo; muestra el “estado” de un número almacenado en el registro de trabajo (el número es mayor o menor que cero etc.) en cualquier instante dado.



Figura 12 - Circuitos que componen a la CPU.

### 1.6.2 Puertos de entrada y salida.

Los microcontroladores tienen uno o más registros (denominados puertos), conectados a los pines en el microcontrolador. Estos pines pueden ser configurados como entrada o salida; esto se puede establecer al inicio del programa o durante su ejecución, y al realizarse por medio de software, esta función puede ser cambiada durante la ejecución del programa. En la Figura 13 se muestra una representación de los pines en un microcontrolador.

Una de las características más importantes a destacar es la corriente máxima que pueden entregar/recibir. En la mayoría de los microcontroladores la corriente obtenida de un pin es suficiente para activar un LED u otro dispositivo de



baja corriente (10-20 mA); mientras más pines de E/S haya, más baja es la corriente máxima de un pin. En otras palabras, todos los puertos de E/S comparten la corriente máxima declarada en la hoja de especificación técnica del microprocesador.

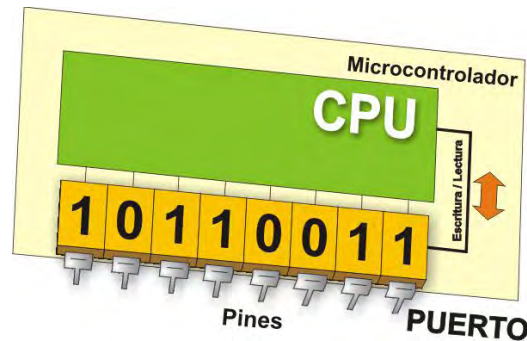


Figura 13 - Puertos del microcontrolador.

Cada puerto de E/S, normalmente está bajo el control de un registro especializado; lo que significa que cada bit de ese registro determina el estado del pin correspondiente en el microcontrolador. Por ejemplo, al escribir un uno lógico (1) a un bit del registro de control, el pin apropiado del puerto se configura automáticamente como salida; eso significa que el voltaje llevado a ese pin se puede leer como 0 o 1 lógico. En caso contrario, al escribir 0 al registro de función especial, el pin apropiado del puerto se configura como salida. Su voltaje (0V o 5V) corresponde al estado del bit apropiado del registro del puerto.

### 1.6.3 Registros.

Un registro es un circuito electrónico que puede memorizar el estado de un byte. Estos registros se pueden utilizar para almacenar información (en los registros generales) o para controlar el estado de los diferentes puertos por medio de los registros de función especial. En la Figura 14 se muestran los registros del microcontrolador, así como su conexión con el CPU.

### 1.6.4 Registros SFR.

Los registros de funciones especiales (SFR) tienen una función predeterminada por el fabricante, estos registros están conectados a los circuitos internos del microcontrolador (también llamados periféricos) tales como temporizadores, convertidores A/D, osciladores entre otros; lo que significa que directamente

manejan el funcionamiento de estos circuitos, es decir, del microcontrolador. En la Figura 15 se muestra uno de los registros de funciones especiales.

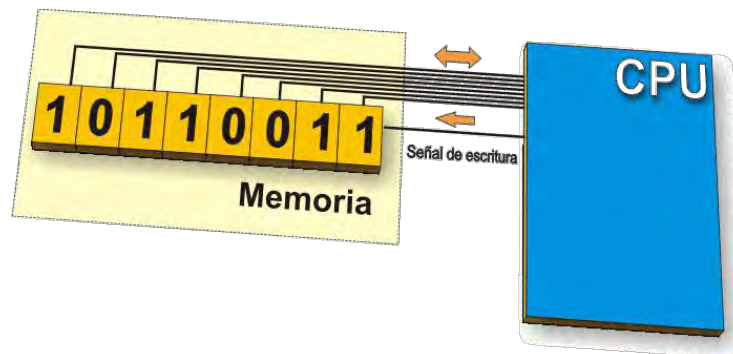


Figura 14 - Registros del microcontrolador.

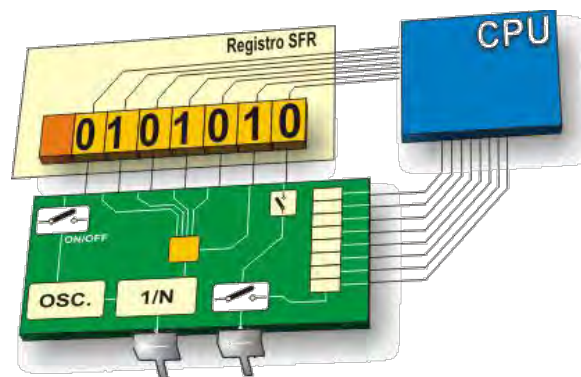


Figura 15 - Registros de funciones especiales.

### 1.6.5 Memoria RAM (*Random Access Memory*) - Memoria de acceso aleatorio.

La memoria RAM es un dispositivo para almacenamiento de información digital; una memoria RAM permite la lectura y escritura de los elementos almacenados en ella en la misma cantidad de tiempo, sin importar la posición física en la que se encuentre almacenada la información.

Se utiliza para almacenar temporalmente los datos y los resultados inmediatos creados y utilizados durante el funcionamiento del microcontrolador. Esta memoria volátil es utilizada comúnmente ya que generalmente es más rápida

que la memoria ROM. Su desventaja es que, al apagar la fuente de alimentación, se pierde el contenido de la memoria.

### 1.6.6 Memoria EEPROM (*ELECTRICALLY ERASABLE PROGRAMMABLE ROM*).

La memoria EEPROM consiste en celdas independientes cada una representando un bit. Estas celdas están basadas en la tecnología de transistor de compuerta flotante (*floating-gate transistor technology*): Una carga eléctrica contenida en la compuerta del transistor determina el nivel lógico de la celda. Se puede definir su funcionamiento de la siguiente manera:

Cuando se borra una celda, una carga eléctrica es colocada en la compuerta del transistor, por lo que se lee como un uno lógico. Programar una celda es equivalente a descargar la celda, causando una lectura de cero lógico. Es sólo posible programar (descargar) una celda que ha sido borrada (cargada). Esto es importante de destacar ya que si no se borra previamente la memoria EEPROM se puede caer en un error de programación al no poder almacenar nuevos valores en la memoria.

El contenido de la EEPROM se puede cambiar durante el funcionamiento (similar a la RAM), pero se queda permanentemente guardado después de la pérdida de la fuente de alimentación (similar a la ROM). Por lo tanto, la EEPROM se utiliza con frecuencia para almacenar los valores creados durante el funcionamiento, que tienen que estar permanentemente guardados.

El microcontrolador ATMEGA328P contiene 1Kbyte de memoria EEPROM organizado en un espacio de memoria diferente, esto significa que se tienen 1024 registros disponibles para almacenar hasta 1 byte en cada uno (255 en sistema decimal)

8	7	6	5	4	3	2	1	Dec
1	1	1	1	1	1	1	1	255

Tabla 2 - contenido de una celda de la EEPROM.

El fabricante además asegura un ciclo de vida de al menos 100,000 ciclos de escritura/borrado. Una de las desventajas de este tipo de memoria es que es una memoria de lento acceso ya que únicamente se puede acceder a un registro de memoria a la vez. (Atmel Corp. *AVR103: Using the EEPROM Programming Modes*,2005).

### **1.6.7 Memoria ROM.**

La memoria ROM se utiliza para guardar permanentemente el programa que se está ejecutando. El microcontrolador que vamos a utilizar, el *ATMEGA328P*, tiene 32 Kilobytes de memoria Flash re-programable, denominada por ellos *In-System Reprogrammable Flash Program Memory*. Esta memoria tiene un ciclo de vida de 10,000 ciclos de escritura/borrado<sup>13</sup>.

### **1.6.8 Memoria flash.**

La memoria flash es un tipo de memoria derivada de la *EEPROM*, permite la lectura y escritura de múltiples posiciones de memoria en la misma operación; por lo cual ésta tecnología permite velocidades superiores a la *EEPROM*.

La unidad básica del circuito de memoria es una celda que almacena un bit. Para ello se cuenta con dos transistores conocidos como compuerta flotante y compuerta de control. La memoria flash tiene una cuadrícula de columnas, llamadas *bitline* y filas llamadas *wordline*, y la unión entre estas es una celda que tiene dos transistores separados entre sí por una delgada capa de óxido.

El transistor de compuerta flotante solo se encuentra unido a los renglones (*wordline*) y la compuerta de control se encuentra unida a las columnas (*bitline*). Para cambiar el valor de 0 a 1 se requiere de un proceso llamado *Fowler-Nordheim tunneling*, este proceso se usa para alterar el lugar de los electrones en la compuerta flotante. Los electrones almacenados en la puerta flotante se drenan al lado de la fuente del transistor. La memoria flash funciona mucho más rápido que las memorias *EEPROM* tradicionales, porque en lugar de borrar un byte a la vez, se borra un bloque o todo el chip y luego se vuelve a escribir. En las

---

<sup>13</sup> Tomado de datasheet: pp 17

Figuras 16 y 17 se muestra cómo se programa un 0 o un 1 lógico en la celda de memoria.

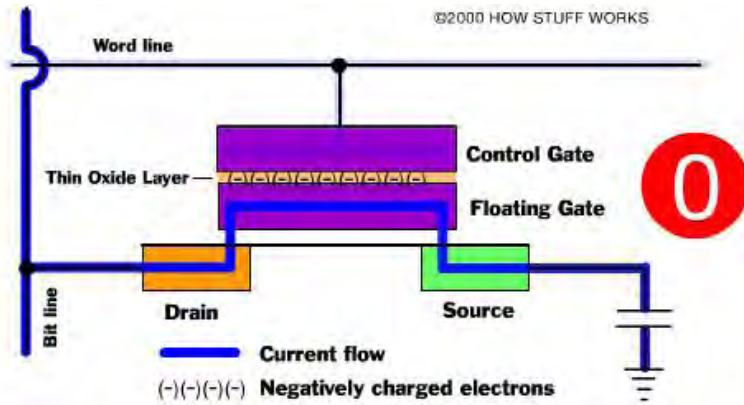


Figura 16 - Celda memoria flash, 0 lógico.

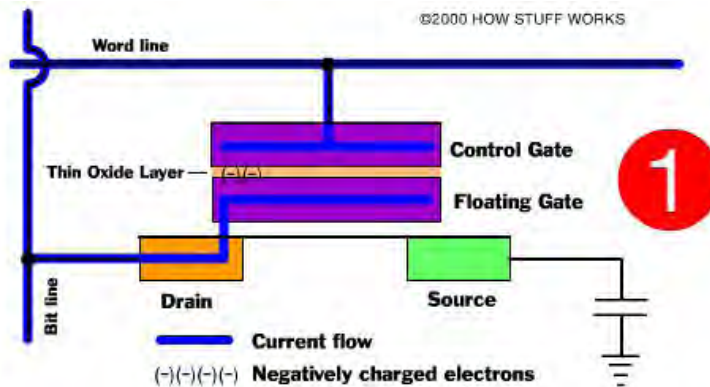


Figura 17 - Celda memoria flash, 1 lógico.

## 1.7 ATMEL.

Atmel Corp. Es una compañía de semiconductores con sede en San José, California, en Estados Unidos. Atmel era el segundo proveedor de microcontroladores (después de Microchip Technology) hasta que este año, el 19 de enero de 2016, Microchip Technology compró a la empresa de Atmel Corp. por \$3,600 millones de dólares (Picker, L, 2016).

## 1.8 ARDUINO.

### 1.8.1 ¿Qué es Arduino?

Arduino es un proyecto *open-source* fundado por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, y David Mellis. Fue inicialmente desarrollado en el *Interaction Design Institute Ivrea* al Norte de Italia. Las tarjetas iniciales de Arduino fueron diseñadas por Massimo Banzi y David Cuartielles. David Mellis desarrolló el primer software basado en *Wiring*. Arduino toma su nombre de un bar en Ivrea, el cual a su vez es nombrado por un rey de Italia de Ivrea. (Kushner, D., 2011).

El sistema de Arduino consta de dos componentes: Una tarjeta de prototipado rápido y un entorno de desarrollo integrado (de sus siglas en inglés *IDE*).

La tarjeta más popular de Arduino es la tarjeta de Arduino Uno. La tarjeta de Arduino Uno emplea como unidad central de procesamiento **un microcontrolador de la familia ATMEL (ATMEGA328P)**.

### 1.8.2 ¿Por qué emplear Arduino?

Arduino se puede considerar como un punto de partida para comenzar a desarrollar proyectos de manera rápida; al ser un proyecto *open source* cualquier persona puede contribuir en el desarrollo del mismo. Arduino puede ser extendido mediante bibliotecas en lenguaje C++, además de esto es posible cambiar de un entorno simplificado como el IDE de Arduino a un software de desarrollo más especializado como Atmel Studio. No solo el software se puede modificar, Arduino también publica los planos de los circuitos para poder adaptar nuestro diseño a nuestras necesidades.

### 1.8.3 Bibliotecas de Arduino

Una biblioteca es un conjunto de funciones que han sido pre compiladas en lo que se conoce como archivo de biblioteca. (Purdum, J., 2012)

## 1.9 ATMEGA328P.

El ATMEGA328P es un microcontrolador de tecnología CMOS de 8 bits, basado en la arquitectura RISC con 131 instrucciones. En la Figura 18 se muestra el diagrama de bloques que compone al microcontrolador.

### 1.9.1 RISC

El acrónimo RISC, proviene de acrónimo RCISC (del inglés *Reduced Complexity Instruction Set Computer*), pero debido a que el acrónimo no es tan sencillo de decir se optó por cambiarlo a RISC (*Reduced Instruction Set Computer*). RISC es un tipo de diseño de CPU generalmente utilizado en microprocesadores o microcontroladores. El diseño RISC está basado en la premisa de que la mayoría de las instrucciones que una computadora decodifica y ejecuta son simples. Como resultado de este concepto, la arquitectura RISC limita el número de instrucciones incorporadas en el microprocesador, pero optimiza cada una de ellas de forma que se ejecuten muy rápidamente (generalmente en un solo ciclo de reloj). Por lo tanto, los chips RISC ejecutan las instrucciones simples más rápidamente que los microprocesadores que cuentan con un conjunto más amplio de instrucciones.

### 1.9.2 Características del ATMEGA328P

Entre las características principales se encuentran:

- 32 KBytes de memoria flash no volátil llamada In-System Self-Programmable Flash program Memory.
- 1 KByte de memoria EEPROM.
- 2 KBytes de memoria SRAM.
- 10,000 ciclos de borrado/escritura de memoria flash.
- 100,000 ciclos de borrado/escritura de memoria EEPROM.
- 2 Timer de 8 bits.
- 1 Timer de 16 bits.
- 6 canales de PWM.
- 6 canales de convertidor analógico-digital de 10 bits, 15 kSPS (15,000 muestras por segundo).
- 1 interfaz de puerto serie USART programable.
- 1 interfaz I<sup>2</sup>C 2-wire.

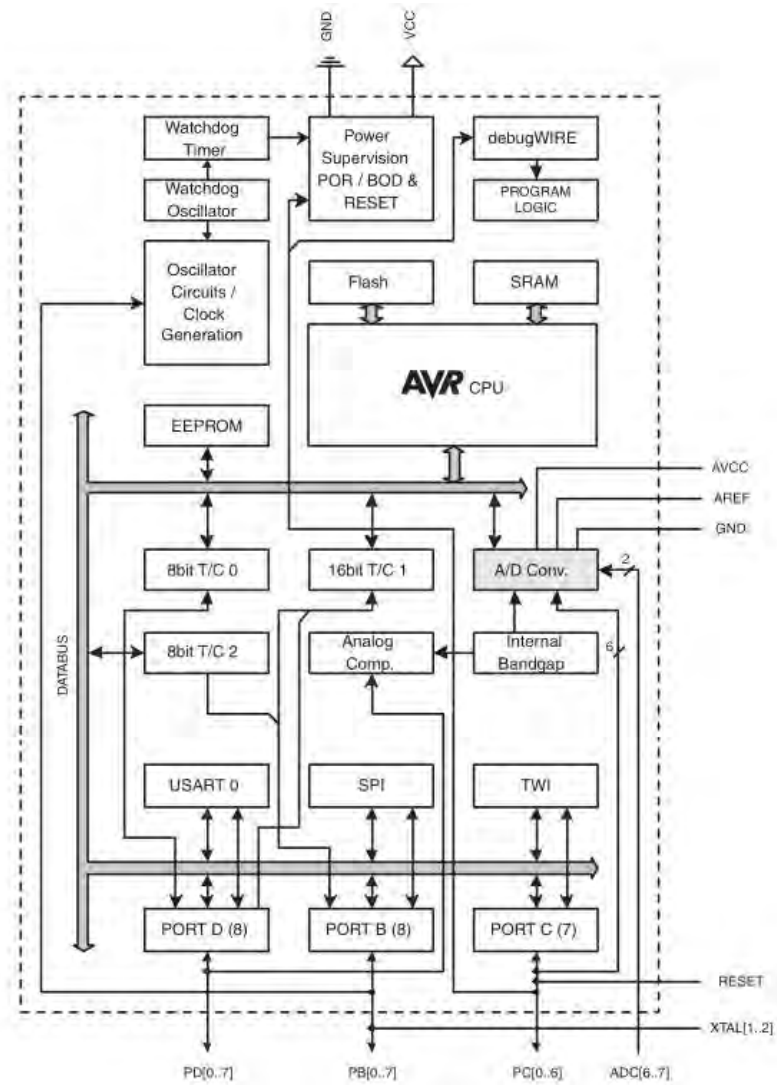


Figura 18 - Diagrama de bloques de ATMEGA328P.<sup>14</sup>

## 1.10 RELOJ DE TIEMPO REAL (RTC).

Se denomina reloj de tiempo real (o RTC) a un circuito integrado con la función de mantiene la hora actual, esto es con la finalidad de que nuestro circuito tenga nociones de tiempo reales; nos es útil tanto para mantener la hora como para detonar eventos con base en tiempo (alarmas).

<sup>14</sup> Diagrama de bloques obtenido de Atmel Corporation. (2016). 8-bit AVR Microcontrollers ATmega328/P DATASHEET COMPLETE, (rev A 06-2016),



## 1.11 RTC PCF8583P

Este circuito integrado cuenta con 256 x 8-bits de RAM con un registro auto incrementable. Además de eso este circuito posee la función de alarma en cuatro diferentes presentaciones:

- Alarma diaria.
- Alarma calendarizada.
- Alarma semanal.
- Temporizador.

El circuito es un integrado de 8 pines; el intercambio de información se realiza mediante el protocolo I<sup>2</sup>C. Como se mencionó previamente, este dispositivo cuenta con 256 registros, donde los primeros 16 son de propósito específico; dejando libre al usuario los siguientes 240 registros, desde la dirección **0x10** hasta la dirección **0xFF**.

Los primeros ocho registros (**0x00-0x07**) contienen la información relacionada al reloj y los siguientes 8 registros (**0x08-0x0F**) solo son válidos cuando se habilita la alarma, de lo contrario se pueden considerar como memoria disponible. En la Figura 19 se muestra el diagrama de bloques para el circuito de RTC PCF8583P.

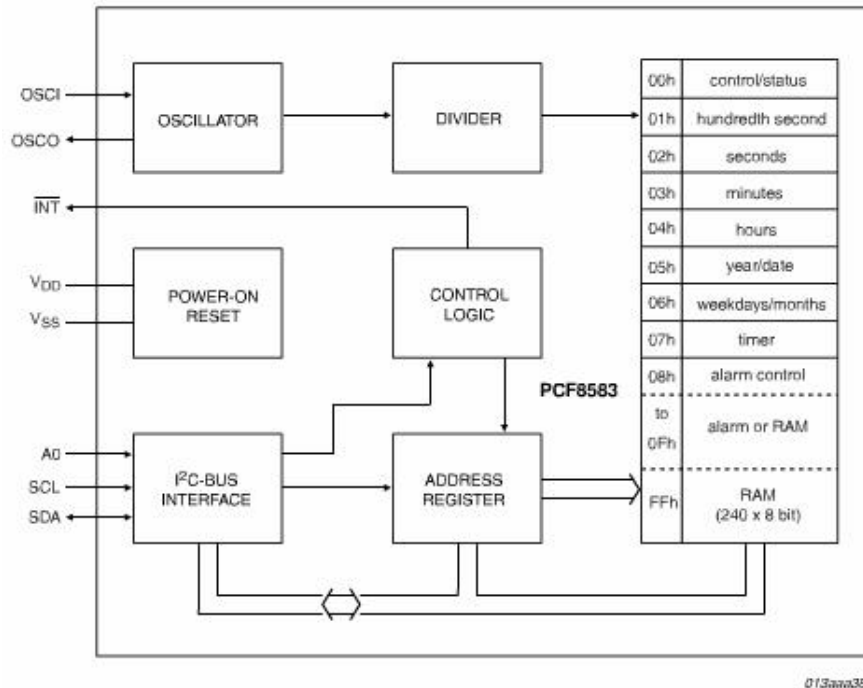


Figura 19 - Diagrama de bloques PCF8583P, muestra el contenido de los diferentes registros.<sup>15</sup>

<sup>15</sup> Tomado del datasheet pp 2 [http://www.nxp.com/documents/data\\_sheet/PCF8583.pdf](http://www.nxp.com/documents/data_sheet/PCF8583.pdf)

## Registro de control (Dirección 0x00)

El registro de la memoria **0x00** es el registro de *Control/Status* que es usado para definir el comportamiento general del chip. El comportamiento de cada bit es el siguiente:

- 0 - Bandera de timer
- 1 - Bandera de la alarma
- 2 - Bandera de máscara
- 3 - Activar alarma
- 4,5 - Modo de función
  - 00 - Reloj a 32.768kHz
  - 01 - Reloj a 50 Hz
  - 10 - Modo contador de eventos
  - 11 - Modo de prueba/modo de fábrica
- 6 - Bandera de retención de última cuenta
  - 0 - Cuenta
  - 1 - Almacena y retiene la última cuenta
- 7 - Bandera de detención de cuenta
  - 0 - Contar pulsos
  - 1 - Detener cuenta, reiniciar divisor

En la Figura 20, se muestra el contenido del registro de memoria 0x00 (Control/Status).

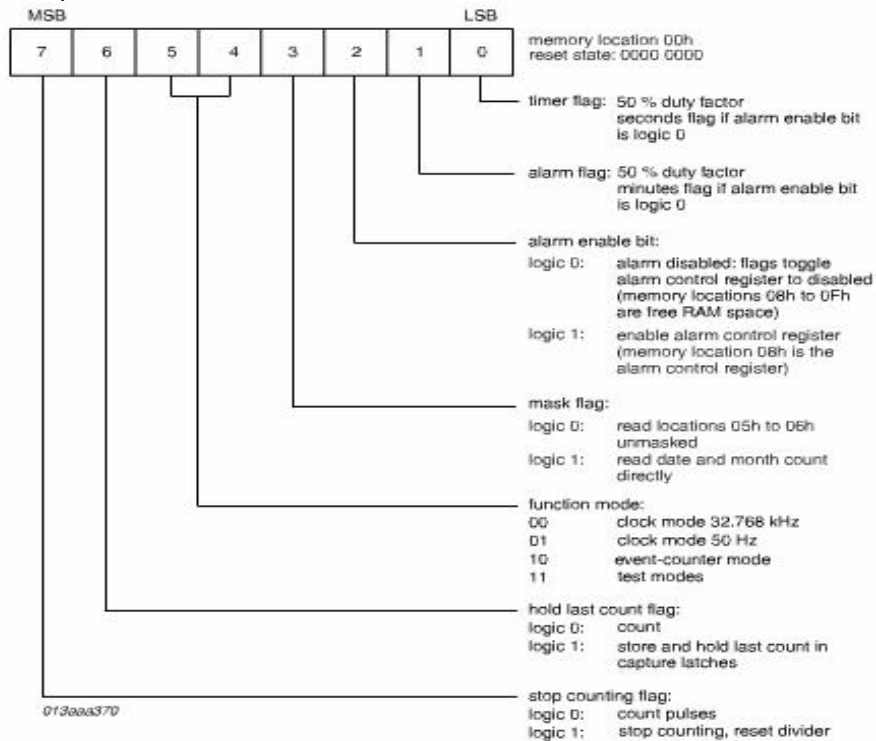


Figura 20 - Contenido del registro de Status/Control.

## Registros de conteo.

Los registros **0x01** a **0x06** son registros de conteo autoincrementables y en estos es donde se almacena la hora y fecha de nuestro circuito. El contenido de estos registros se encuentra almacenado en formato BCD. Se muestra en la Figura 21 los diferentes registros de conteo, los cuales se deben de programar empleando el formato BCD.

control/status	00h
hundredth of a second 1/10 s   1/100 s	01h
seconds 10 s   1 s	02h
minutes 10 min   1 min	03h
hours 10 h   1 h	04h
year/date 10 day   1 day	05h
weekdays/months 10 month   1 month	06h

Figura 21 - Contenido de los registros 0x00 a 0x06.

## Formato BCD.

El formato BCD (Decimal Codificado en Binario) es un estándar para representar números decimales en el sistema binario, en donde cada dígito decimal es codificado con una secuencia de 4 bits. Con esta codificación especial de los dígitos decimales en el sistema binario, se pueden realizar operaciones aritméticas como suma, resta, multiplicación y división de números en representación decimal, sin perder en los cálculos la precisión ni tener las inexactitudes en que normalmente se incurre con las conversiones de decimal a binario puro y de binario puro a decimal.

## Representación en BCD.

Cada dígito decimal tiene una representación binaria codificada con 4 bits, en la Tabla 3 se muestra la codificación para los diferentes dígitos.

Decimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Tabla 3 - Representación de números decimales en formato BCD.

Los números decimales, se codifican en BCD con los bits que representan sus dígitos.

Por ejemplo, en la Tabla 4 tenemos la codificación en BCD del número decimal 95 que es:

Decimal	9	5
BCD	1001	0101

Tabla 4 - Ejemplo de codificación BCD, número 95.

De esta forma, podemos leer el número 95 como el número binario **10010101**, que en formato hexadecimal sería el **0x95**.

De igual manera, para otro número como el 37, tenemos en la Tabla 5 su codificación.

Decimal	3	7
BCD	0011	0111

Tabla 5 - Ejemplo de codificación BCD, número 37.

De igual forma este número en su representación binaria es el número **00110111**, y en su formato hexadecimal sería el **0x37**.

### Registro de conteo (1/100 s) 0x01

Almacena desde el número **0 (BCD 0x00)** hasta el número **99 (BCD 0x99)**. Al superar el número 99, regresa a 0, y suma 1 al siguiente registro, registro de segundos (**0x02**).

### Registro de segundos 0x02

Almacena desde el número **0 (BCD 0x00)** hasta el número **59 (BCD 0x59)**. Al superar el número 59, regresa a 0, y suma 1 al siguiente registro, registro de minutos (**0x03**).

### Registro de minutos 0x03

Al igual que el registro de segundos, almacena desde el número **0 (BCD 0x00)** hasta el número **59 (BCD 0x59)**. Al superar el número 59, regresa a 0, y suma 1 al siguiente registro, registro de horas (**0x04**).

## Registro de hora 0x04

Almacena desde el número **0 (BCD 0x00)** hasta el número **23 (BCD 0x23)**, pero además en este registro se controla si se utiliza el formato de 12/24 horas, así como la bandera de a.m. y p.m. Se muestra en la Figura 22 este registro, el registro 0x04.

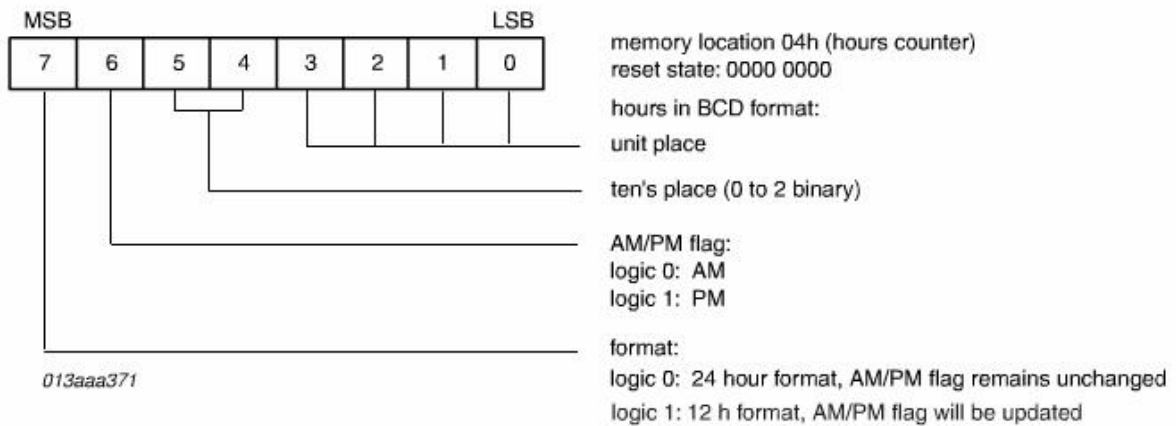


Figura 22 - Registro de hora.

Si, por ejemplo, queremos que sean 13 horas, el contenido del registro debe de ser **0001,0011<sub>2</sub>=0x13**.

De igual forma, si queremos que sea la 1 p.m., el contenido del registro debe de ser **1100,0001<sub>2</sub>=0xC1**.

## Registro año/fecha 0x05

En este registro se almacenan los días y el año del circuito. En la Tabla 6 se muestran los diferentes números del día del mes:

Unit	Counting cycle	Carry to next unit	Contents of month calendar
date	01 to 31	31 to 01	1, 3, 5, 7, 8, 10, and 12
	01 to 30	30 to 01	4, 6, 9, and 11
	01 to 29	29 to 01	2, year = 0
	01 to 28	28 to 01	2, year = 1, 2, and 3
year	0 to 3	-	-

Tabla 6 - Número de días programables de acuerdo al mes.

Se muestra en la Figura 23 el registro de año/fecha:

El año es almacenado en un valor de 0 a 3, donde 0 es año bisiesto, y ya que el año de 2016 es un año bisiesto, el año a almacenar será 0

De esta forma, si tenemos el día 23 del año 2016, el contenido del registro será:

**0010,0011<sub>2</sub>=0x23**

Para otra fecha, por ejemplo, el día 12 del año 2018; el contenido del registro sería:

**1001,0010<sub>2</sub>=0x92**

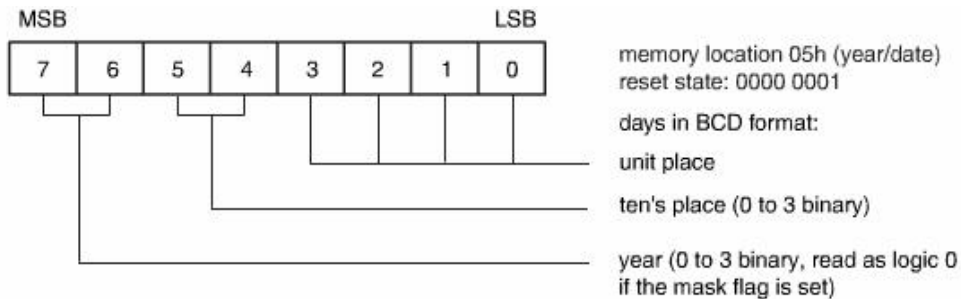


Figura 23 - Contenido del registro de fecha / año.

### Registro de día de la semana / número de mes 0x06

Este registro almacena el número de día de la semana (de 0 a 6) y el número de mes (de 1 a 12). En la Tabla 7 se muestran las diferentes fechas que se pueden almacenar, así como si cuentan con acarreo y en la Figura 24 se muestra el contenido del registro a nivel de bit para el almacenamiento de la información en este registro.

Unit	Counting cycle	Carry to next unit	Contents of month calendar
months	01 to 12	12 to 01	-
weekdays	0 to 6	6 to 0	-
timer	00 to 99	no carry	-

Tabla 7 - Ciclo de conteo de meses y día de la semana.

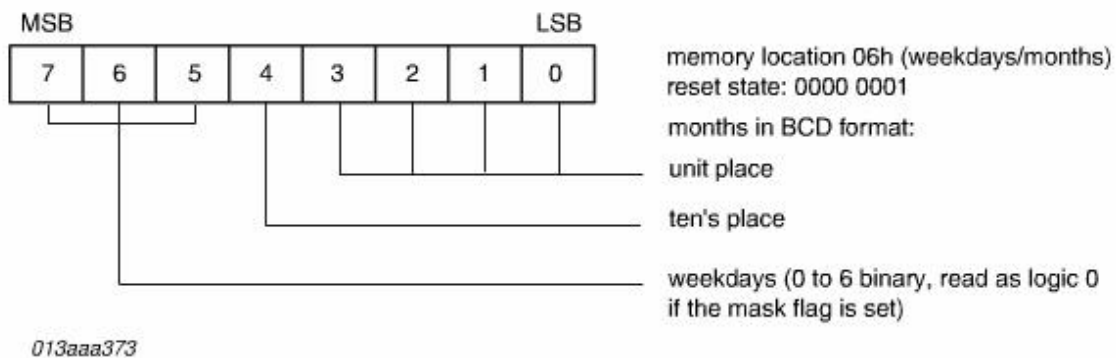


Figura 24 - Contenido del registro de número de día/número de mes.

## Registro de control/alarma (Dirección 0x08)

Para habilitar los registros de la alarma (**0x08-0x0F**) el **bit 2** del registro de control (**0x00**) debe estar en **1** lógico. Cuando se habilita la alarma, ésta se configura en el registro 0x08

Se requiere un reloj sin timer y con la alarma calendarizada, con interrupción. Para programar esto, se almacena en el registro **0x08** el valor de **10110000<sub>2</sub>=0xB0**

Cuando un evento de alarma ocurre, la bandera de alarma del registro de control y del registro de estado es habilitada (puesto a 1).

Este modo requiere la siguiente configuración:

- **Registro de Control/Estado**
  - Bit **2** habilitado para habilitar alarmas.
- **Registro de la alarma**
  - Bits **4-5** habilitados en alguno de los modos de reloj.
  - Bit **7** habilitado para activar las interrupciones por alarma

El ejemplo en pseudocódigo para programar el reloj es:

- *Detener el contador* – En el registro de *Control/Status* **bit 7 = 0**
- *Ajustar tiempo del reloj* – Con los registros *Clock time/date/* **0x01 - 0x06**
- *Ajustar tiempo de la alarma* – Con los registros *Alarm time/date* **0x09 - 0x0e**
- *Habilitar alarma* – En el registro *Control/Status* **bit 2 = 1**
- *Comenzar Contador* – En el registro *Control/Status* **bit 7 =0**

En la Figura 25 se muestra el registro a nivel de bit para el almacenamiento de la información en este registro.

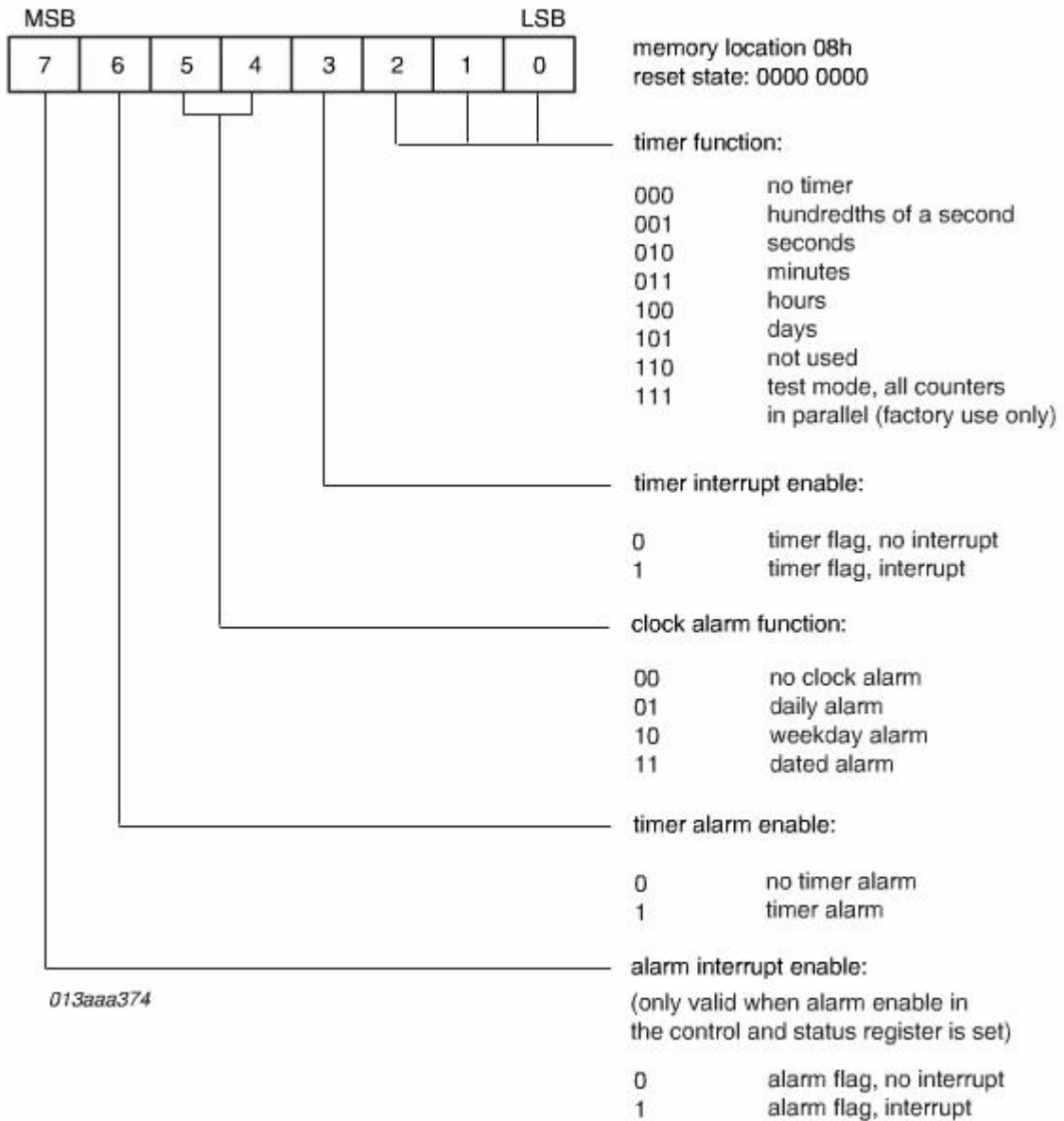


Figura 25 - Contenido del registro de la alarma.



## **2 PLANTEAMIENTO DEL PROBLEMA.**

---

Las trampas de sedimentos presentan diversos problemas, los cuales se listan a continuación:

- Debido a un sellado imperfecto en los cilindros que contenían la electrónica las trampas de sedimentos presentaron fallas. Tras la limpieza del circuito de control se pudo verificar el daño ocasionado por el ingreso de agua.
- Para poder medir la magnitud del daño se trató de verificar el funcionamiento de las trampas mediante el software original con el que operaban las trampas de sedimentos, después de verificar la conexión con la trampa no se pudo obtener la información de la misma, puesto que solo mostraba un error no documentado en el software.
- Las trampas de sedimentos son equipos con una edad mayor a 20 años; esto significa que el software se encontraba obsoleto, ya que requería de una computadora con Windows 98 o anterior.
- Debido a la limitada documentación del software de control con la que contábamos se optó por un re-diseño del circuito de control, así como del software, esto con la finalidad de poder utilizar las trampas de sedimentos con equipos de cómputo más actuales. Se requiere diseñar un programa que pueda ser utilizado en computadoras actuales; se desea incluir soporte para computadoras con el Sistema Operativo Windows XP y superiores. El Hardware tendrá que ser reemplazado por completo al tener información muy limitada acerca del Hardware original.

### **2.1 ANÁLISIS DEL FUNCIONAMIENTO DEL CIRCUITO DE CONTROL ORIGINAL.**

Para diseñar el circuito de control se hizo un análisis del funcionamiento del circuito original. Para comenzar se verificaron las diferentes tarjetas que componían al circuito. En total son cinco tarjetas.

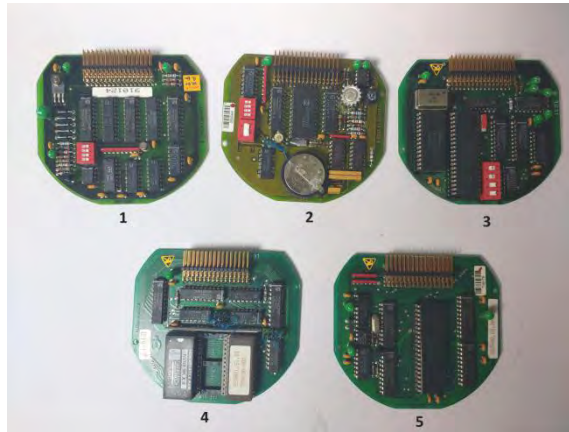


Figura 26 - Tarjetas que componen al circuito de control.

### 2.1.1 Tarjeta 1.

Esta tarjeta contiene:

- 1) Convertidor analógico digital ADC0804L.
- 2) Buffer 74HCT244.
- 3) Receptor de bus 74HCT245.
- 4) Demultiplexor 74HCT138.
- 5) Puente H L293B.
- 6) Detector de voltaje 5D438.

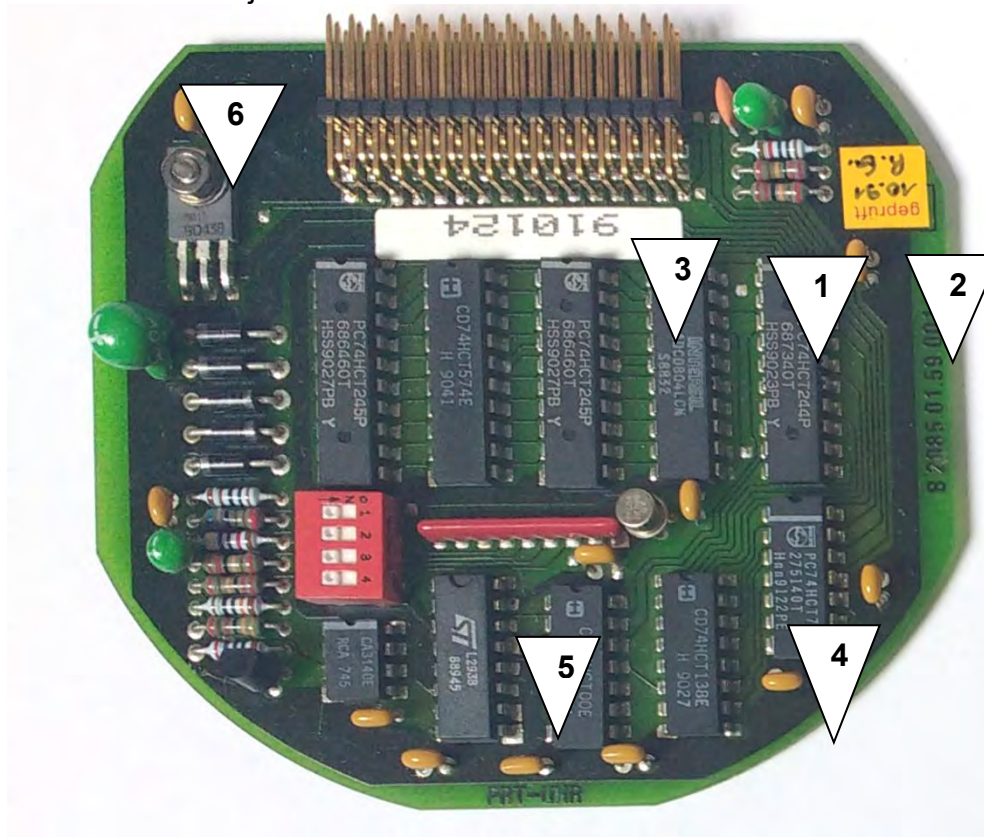


Figura 27 - Tarjeta 1.

Se pueden asumir las siguientes funciones:

- Control de motor de CD, esto ya que se emplea un puente H.
- Medición de voltaje en la trampa de sedimentos, esto de acuerdo al circuito convertidor de análogo a digital, y del detector de voltaje; envía ésta información y es almacenada en otra tarjeta; para que al final ésta información sea desplegada en el archivo de texto tras recuperar la trampa de sedimentos.

### 2.1.2 Tarjeta 2.

Esta tarjeta contiene:

- 1) Reloj de tiempo real ICM7170.
- 2) Buffer 74HCT244.
- 3) Receptor de bus 74HCT245.
- 4) Arreglo de transistores general NPN CA3045.
- 5) Batería de 3.3V.
- 6) Comparador de magnitud de 4 bits 75HCT85.

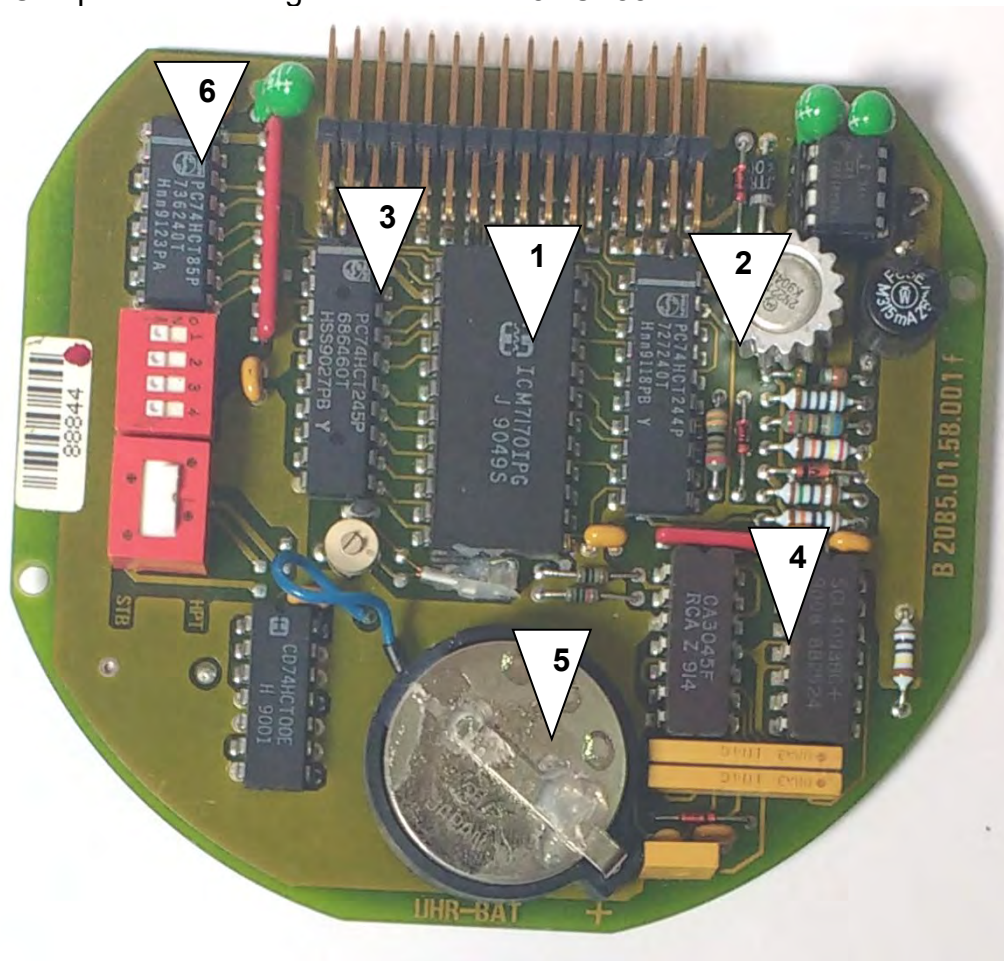


Figura 28 - Tarjeta 2.

De esta tarjeta se puede asumir que se trata de un circuito RTC, se compara el valor actual del RTC con el valor almacenado en la tarjeta 4; no hay información acerca de los diferentes switches del lado izquierdo de la tarjeta.

### 2.1.3 Tarjeta 3.

Esta tarjeta contiene:

- 1) Microprocesador Toshiba familia 8080 TMPZ84.
- 2) Microprocesador Toshiba familia 8080 TMPZ84.
- 3) Buffer 74HCT244.
- 4) Driver serial MAX232.
- 5) Demultiplexor 74HCT138.

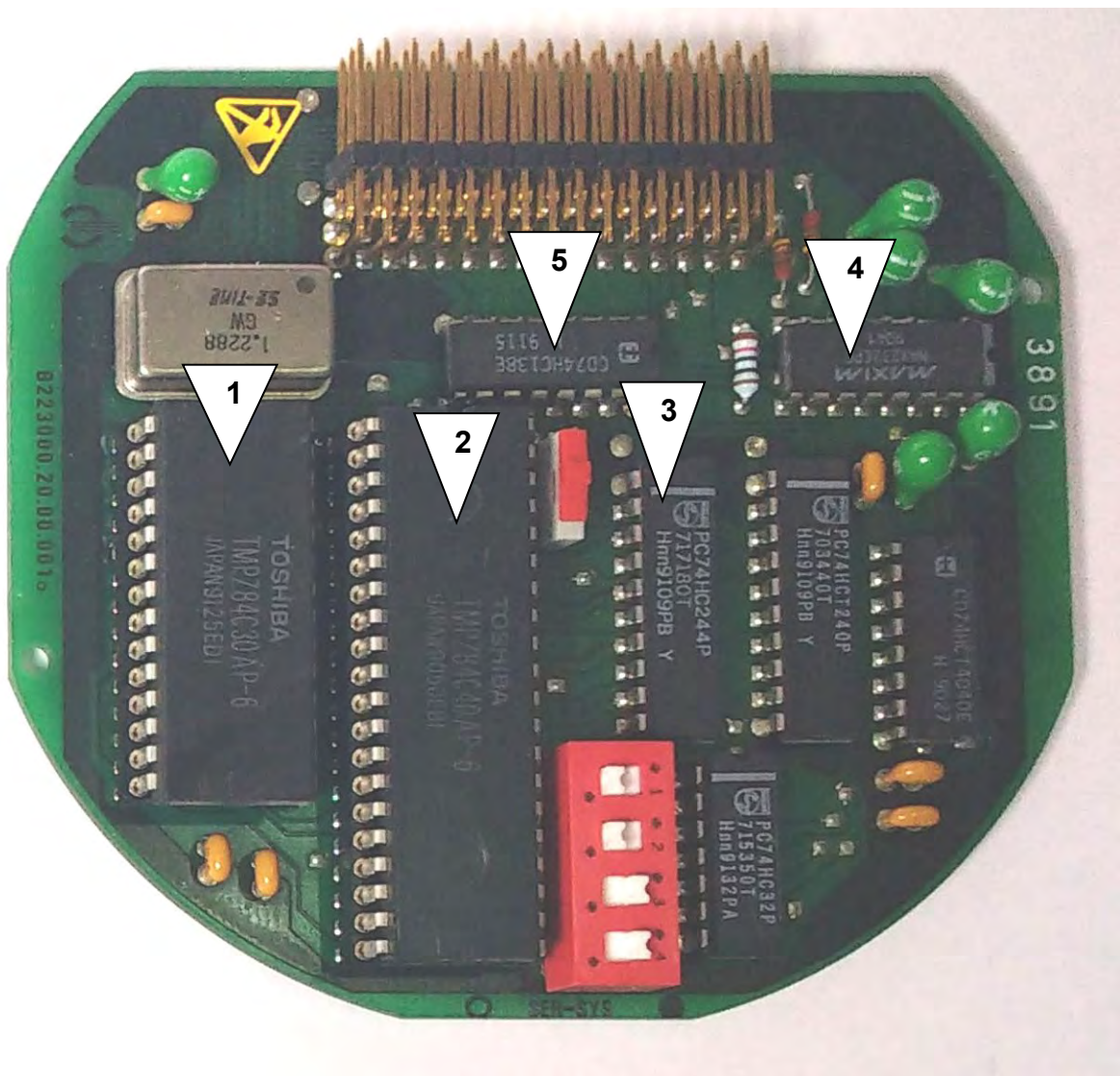


Figura 29 - Tarjeta 3.

Esta tarjeta es la que realiza el control de la trampa de sedimentos; debido a que utiliza microprocesadores, requiere de los diferentes periféricos (buffer, demultiplexor, etc.) De igual forma, ésta tarjeta es la que permite la comunicación con la PC mediante el driver MAX232.

#### 2.1.4 Tarjeta 4.

Esta tarjeta contiene:

- 1) RAM no volátil.
- 2) Compuerta lógica.
- 3) Receptor de bus 74HCT245.
- 4) Demultiplexor 74HC138.
- 5) No identificado.

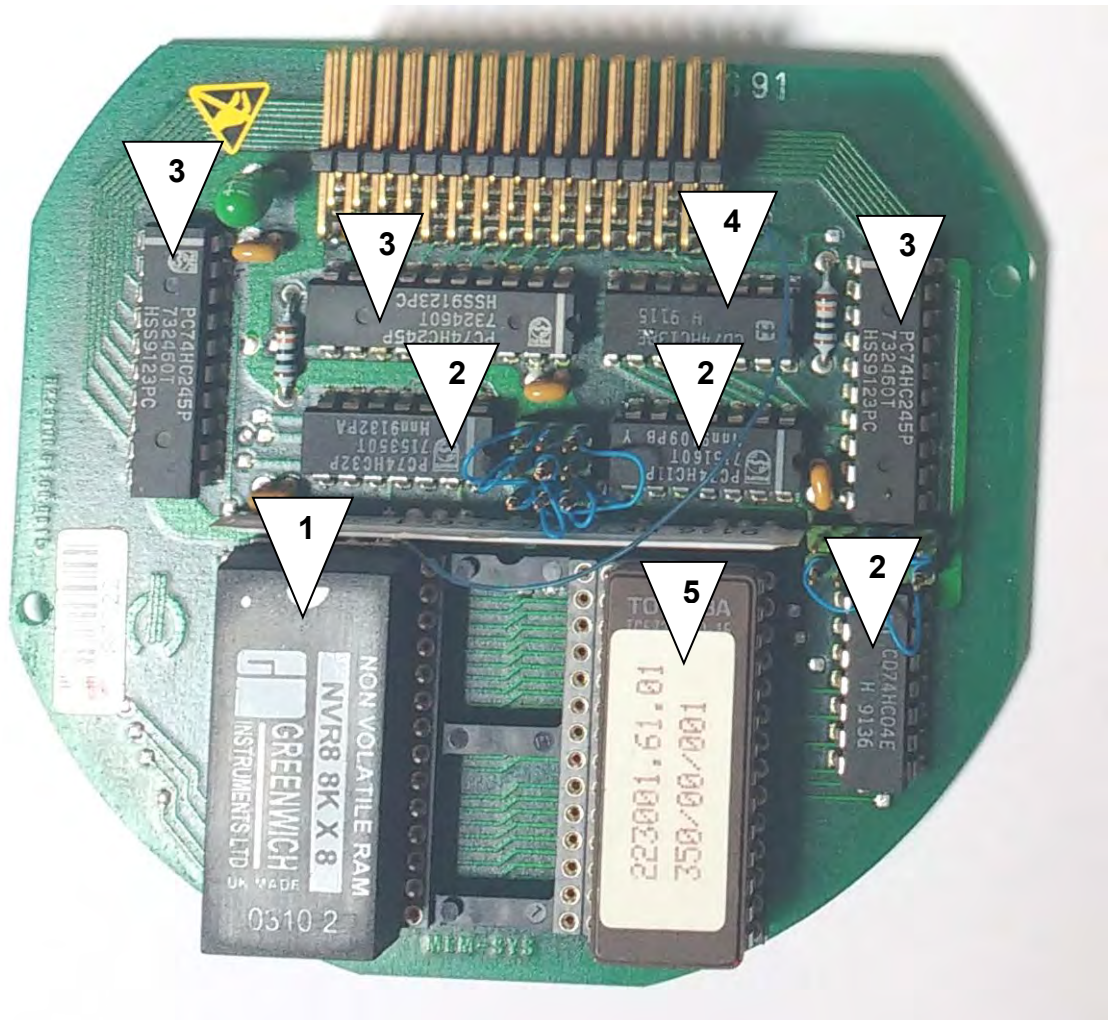


Figura 30 - Tarjeta 4.

Esta tarjeta contiene una RAM no volátil, eso significa que aquí es dónde se almacena la configuración de la trampa (horas, fechas, voltajes, etc.).

### 2.1.5 Tarjeta 5.

Esta tarjeta contiene:

- 1) Microprocesador Toshiba familia 8080 TMPZ84.
- 2) Receptor de bus 74HCT245.
- 3) Buffer 74HCT244.
- 4) Microprocesador Toshiba familia 8080 TMPZ84.
- 5) Compuerta lógica.
- 6) Supervisor de voltaje TL7705.

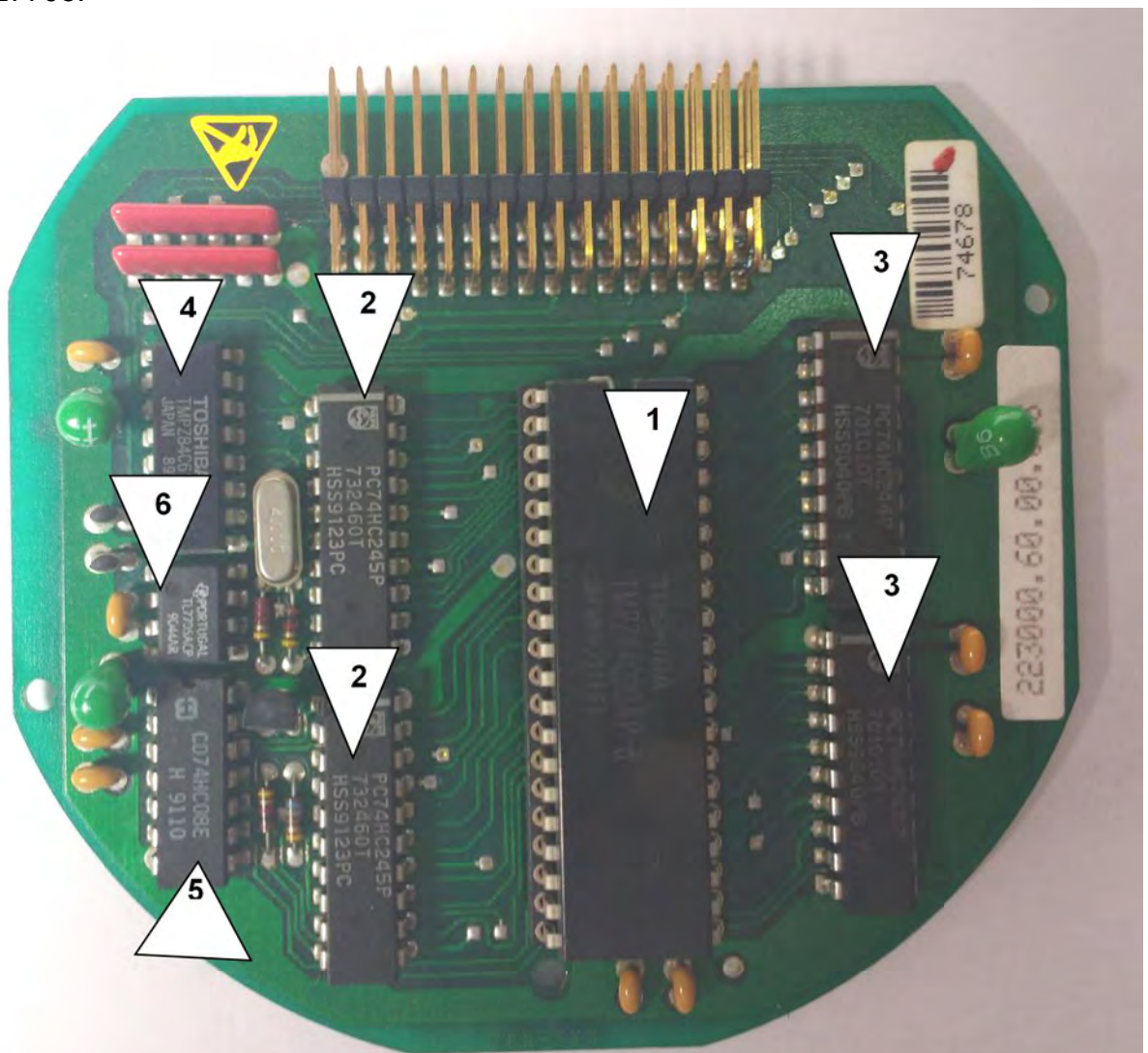


Figura 31 - Tarjeta 5.

Esta tarjeta funciona como control de la trampa de sedimentos; es sensible a la variación de voltaje, por eso es que tiene un supervisor de voltaje, para asegurar 5V en todo el circuito.

## 2.2 ANÁLISIS DEL FUNCIONAMIENTO DEL PROGRAMA DE CONTROL ORIGINAL.

### 2.2.1 Puesta en funcionamiento del software

Para analizar el funcionamiento del software original de control se requirió la emulación del entorno DOS de Windows (originalmente disponible en versiones de Windows 98 y anteriores). Para la simulación se empleó el software libre **DOSBox**, una aplicación que funciona en cualquier sistema operativo que puede ser descargado desde la siguiente dirección:

<http://www.dosbox.com/download.php?main=1>

Posterior a la instalación se debe configurar la ruta de ejecución del software de la siguiente manera:

- Se crea un nuevo acceso directo.

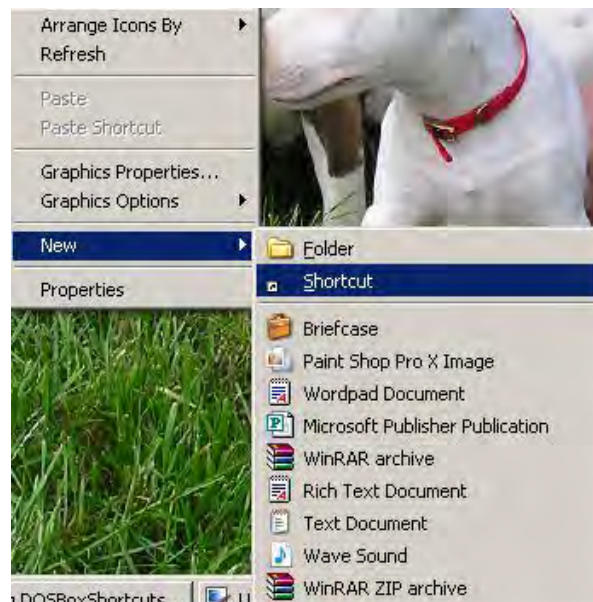


Figura 32 - Creación de acceso directo.

Con lo que aparece la siguiente ventana:

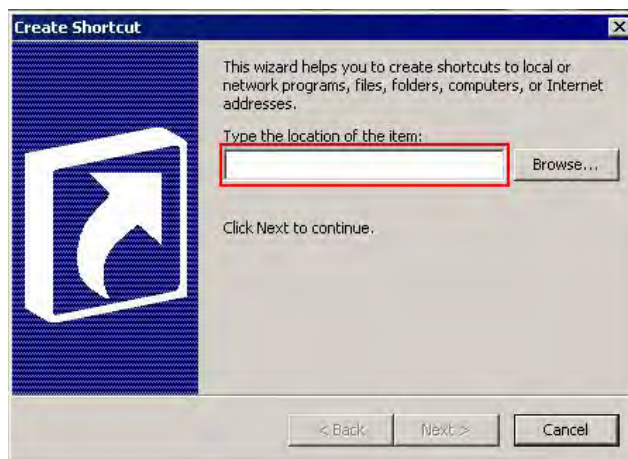


Figura 33 - Selección de ruta para creación de acceso directo.

En el recuadro seleccionado se debe de colocar la ruta completa del software DOSBox:

- Para la versión de 64 bits de Windows la ruta es: **C:\Program Files (x86)\DOSBox-0.74**
- Para la versión de 32 bits la ruta es: **C:\Program Files\DOSBox-0.74**

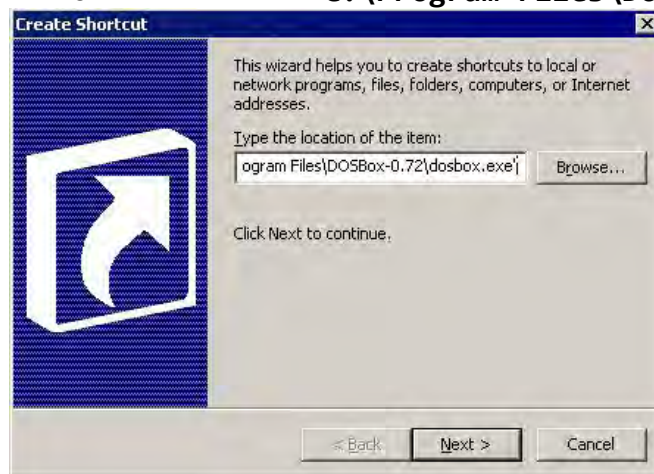


Figura 34 - Ruta de acceso a DOSBox.

Después se escribe un espacio y se coloca entre comillas la ruta del programa de DOS que se desee ejecutar, en este caso, la ruta que se empleó es la siguiente:

**"C:\Program Files (x86)\DOSBox-0.74\DOSBox.exe" "C:\sediment\SNK.EXE"**



Después solo se selecciona un nombre para el ejecutable y esto permitirá ejecutar el software de DOS sin importar la versión de Windows que se esté utilizando. Se puede comprobar que todo esté correcto al verificar en las propiedades del archivo y al ejecutar el programa se puede ver el software original en funcionamiento.

### 2.2.2 Ejecución del software original

Se procede a ejecutar el software original; la pantalla de bienvenida es la siguiente:



Figura 35 - Pantalla de bienvenida.

Como se puede observar, el programa se compone de 5 pestañas: Configuración, Lanzamiento, Recolección, Base de datos y utilidades.

#### 2.2.2.1 Menú de configuración.

En este menú se realiza la configuración principal de la trampa; las opciones disponibles son:

- **Número de muestras (*Number of samples*):** Permite elegir entre un mínimo de 2 y un máximo de 21 muestras.
- **Datos de lanzamiento (*Launching data*):** Permite almacenar un texto con información que se considere relevante.
- **Configuración de las muestras (*Sampling list manual*):** Permite programar las diferentes fechas para cada muestra.

- **Configuración automática de las muestras (*Sampling list semi-automatic*):** Permite programar de manera rápida las diferentes muestras, al elegir una fecha inicial y un intervalo de repetición entre cada muestra.
- **Leer archivo de configuración (*Read configuration file*):** Permite leer la configuración de la trampa cargando un archivo formato \*.CFG
- **Guardar archivo de configuración (*Save configuration file*):** Permite guardar la configuración de la trampa en un archivo formato \*.CFG
- **Configuración del puerto COM (*Port to sediment trap*):** Permite seleccionar el puerto COM con el que se conectará la trampa de sedimentos.
- **Imprimir configuración (*Print configuration*):** Permite imprimir la configuración de la trampa de sedimentos.
- **Guardar configuración (*Configuration to textfile*):** Permite guardar la configuración de la trampa de sedimentos en un archivo de texto.



Figura 36 - Menú de configuración.

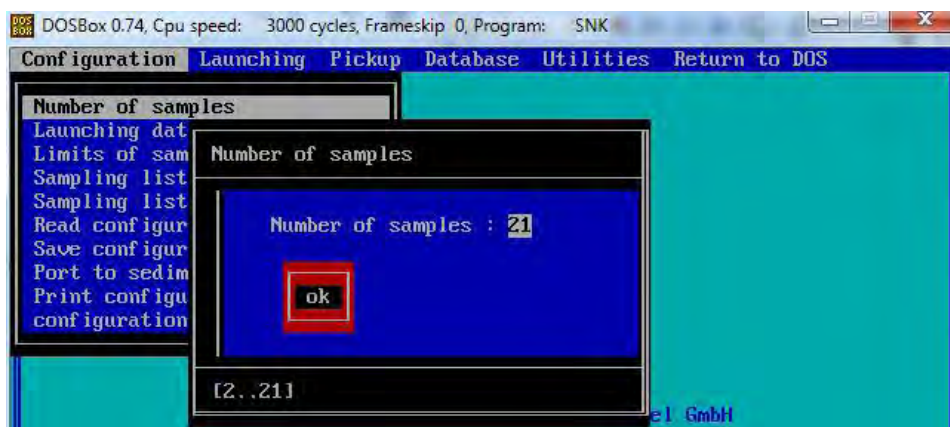


Figura 37 - Elección del número de muestras.

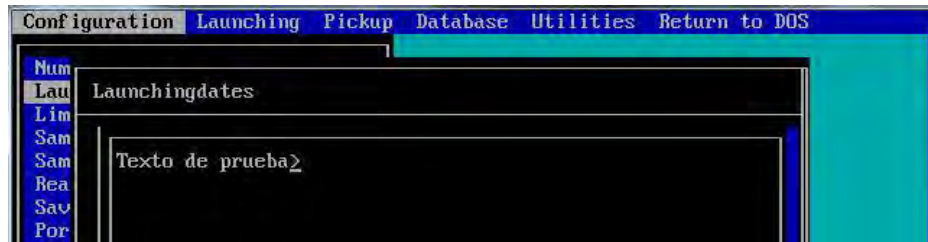


Figura 38 - Datos de lanzamiento.



Figura 39 - Elección de fechas para la muestra.

### 2.2.2.2 Menú de lanzamiento y de recolección.

Estos menús no están disponibles, ya que la trampa de sedimentos no se puede conectar con el programa.



Figura 40 - Trampa de sedimentos tratando de conectarse al programa.



Figura 41 - Error en la conexión.

### 2.2.2.3 Menú Base de Datos.

En el menú de base de datos se pueden almacenar las fechas en un archivo de texto.

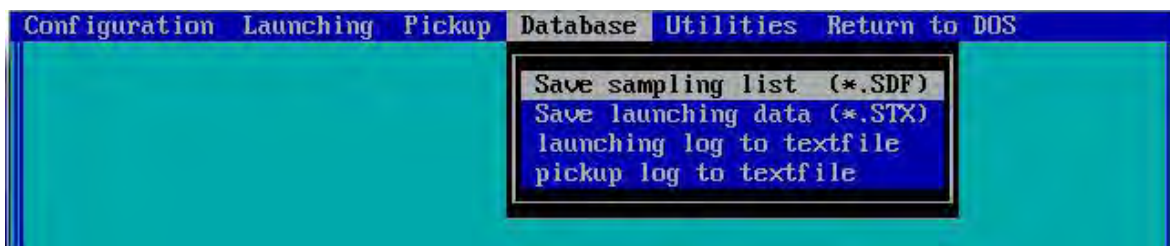


Figura 42 - Menú base de datos.

### 2.2.2.4 Menú Utilidades.

En el menú de utilidades se puede observar: Acerca de del programa, imprimir las fechas de las muestras, girar a la siguiente muestra, verificar el voltaje de la trampa de sedimentos y ponerle un nombre a la trampa de sedimentos.



Figura 43 - Menú de utilidades.

## 3 DESARROLLO.

### 3.1 PROPUESTA DE DISEÑO DEL CIRCUITO.

Después de analizar las diferentes tarjetas se tiene una noción de los requisitos del circuito:

- Debe contar con el control de un motor de corriente directa.
- Debe contar con un circuito de RTC para poder activar eventos basados en tiempo (alarmas).
- Debe contar con la capacidad de almacenar datos, aún sin alimentación continua.
- Debe ser capaz de conectarse con una PC por medio de una conexión RS232.
- Debe contar con un circuito de control principal.

#### 3.1.1 Control de motor de CD.

Para el control de motores originalmente se utilizaba un puente H L293B, el cual puede manejar una corriente de hasta 1A, pero al revisar el diseño de la tarjeta se puede ver que no cuenta con un disipador de calor; lo que pudo haber causado una falla en la misma, por eso se optó por utilizar un circuito más avanzado. El circuito será el L298N, un puente H capaz de manejar hasta 2A, y con la capacidad de agregar un disipador de calor de forma externa. En la Figura 44 se muestra el circuito L298N con un disipador de calor externo.



Figura 44 - Circuito L298N con disipador de calor.

Por lo cual el diseño propuesto será el siguiente:

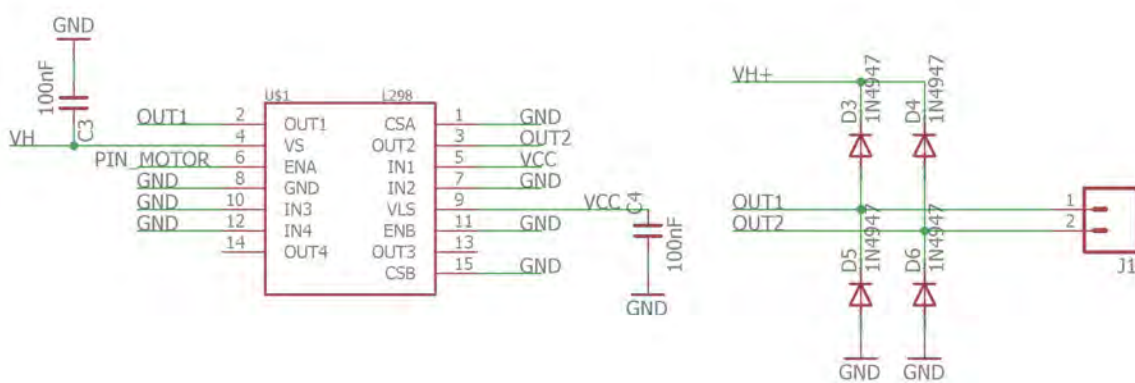


Figura 45 - Circuito de control del motor L298N.

- **OUT1** y **OUT2** representan las conexiones del motor.
- **PIN\_MOTOR** representa la señal de control del motor.
- **GND** es conexión a tierra.
- **VCC** es conexión a 5V.
- **VH** es el voltaje de alimentación del motor 9V+.

Cabe destacar que se emplean dos capacitores de 100nF en la alimentación para desacoplo de señales, por recomendación del fabricante.

También se emplean diodos de recuperación rápida (tiempo de recuperación menor a 250 nseg); esto con finalidad de evitar el retorno de corriente generada por el movimiento del motor. Estos diodos serán los **1N4947**.

### 3.1.2 Circuito de RTC.

El funcionamiento del circuito está basado en la activación con base en tiempos. Para el circuito de RTC se emplea el **PCF8583P**. El protocolo de comunicación con el circuito es el protocolo I<sup>2</sup>C. Este circuito requiere de alimentación de respaldo, para ellos se utiliza una batería CR2032, y debido a que no tiene un pin dedicado a la alimentación de respaldo se emplean un conjunto de diodos. Además, este circuito requiere de un oscilador de 32.768kHz, así como dos resistencias *pull up*, una en **A0**, para configurar la dirección de lectura/escritura, y otra en el **pin de interrupción**, ya que este pin es una salida de **drenaje abierto activa baja**. Esto se puede ver como un switch, cuando está activado no permite el paso de corriente, y cuando se desactiva permite el paso de corriente.

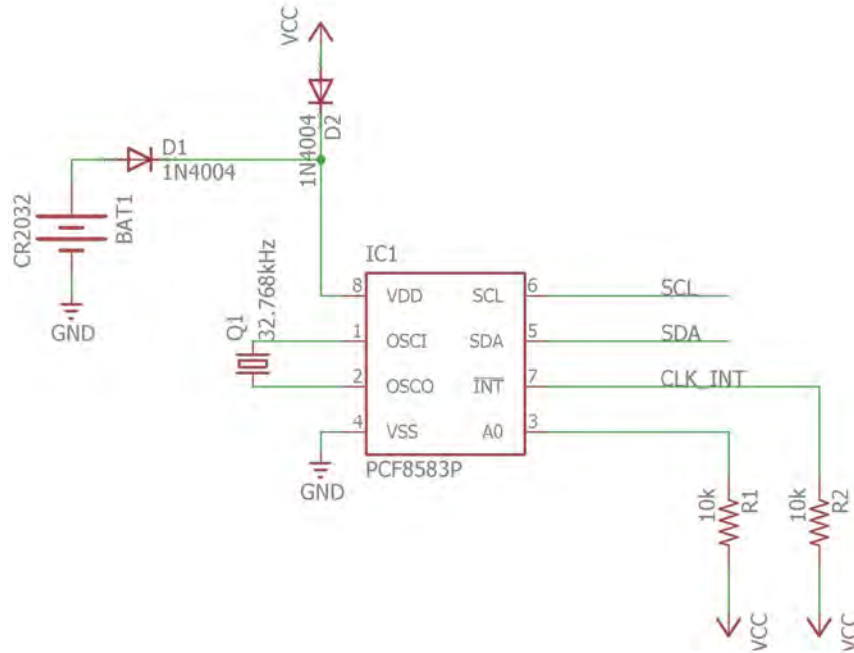


Figura 46 - Circuito de control RTC PCF8583P.

- **SCL** es la línea de reloj para la comunicación I<sup>2</sup>C.
- **SDA** es la línea de comunicación serial para la comunicación I<sup>2</sup>C.
- **CLK\_INT** es el pin por el cual se crea una interrupción de la alarma (Cuando la alarma se activa, este pin se activa permitiendo el paso de corriente eléctrica).

### 3.1.3 Conexión RS232.

Para la programación se diseñó una interfaz la cual se comunica con el microcontrolador por medio de una serie de comandos definidos; para la comunicación se utiliza la comunicación RS-232, y los comandos se reciben por medio de una cadena de caracteres y se analizan para decidir las acciones causadas por el comando.

Para realizar la conexión serial se emplea el circuito MAX233, ésta es una versión modificada del circuito MAX232 de forma que no requiera de capacitores externos para funcionar.

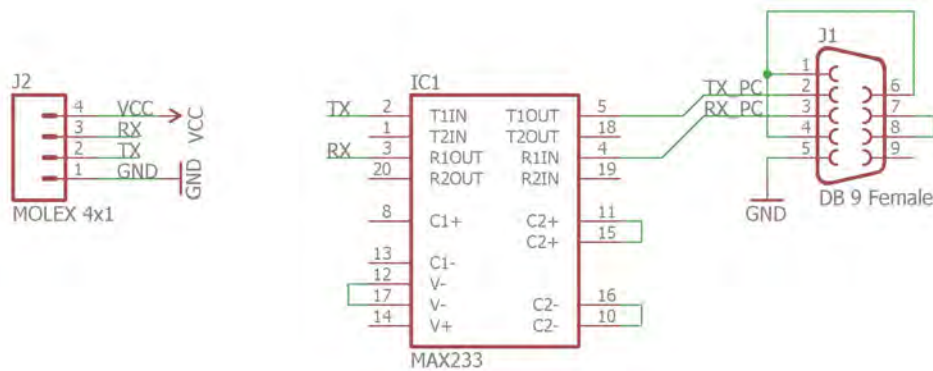


Figura 47 - Circuito RS232.

### 3.1.4 Circuito de control/almacenamiento de datos.

Para el almacenamiento de datos se utilizará la sección de memoria EEPROM del microcontrolador ATMEGA 328P y la información se almacena en los primeros 100 registros de los 1024 disponibles; la información a almacenar será:

- Días [0..20].
- Mes [0..20].
- Año [0..20].
- Fecha de programación inicial (día, mes, año, hora, minuto, segundo).
- Posición inicial de la trampa.
- Posición actual de la trampa.
- Próxima alarma a programar.

Se utiliza un circuito que puede almacenar sólo una alarma, de manera que el procedimiento para programar varias alarmas será el siguiente:

1. Grabar en la memoria EEPROM todas las fechas de la alarma.
2. Programar primer alarma en RTC.
3. Esperar a la activación de la alarma.
4. Cuando se active la alarma, leer de memoria EEPROM la siguiente alarma.
5. Programar siguiente alarma.
6. Repetir el proceso hasta que se activen todas las alarmas.

El ordenamiento de estos datos será:



n	day[i]	n	month[i]	n	year[i]	n	Clock
0	day[0]	25	month[0]	50	year[0]	75	day_
1	day[1]	26	month[1]	51	year[1]	76	month_
2	day[2]	27	month[2]	52	year[2]	77	year_
3	day[3]	28	month[3]	53	year[3]	78	hour_
4	day[4]	29	month[4]	54	year[4]	79	minute_
5	day[5]	30	month[5]	55	year[5]	80	second_
6	day[6]	31	month[6]	56	year[6]	81	paInicial
7	day[7]	32	month[7]	57	year[7]	82	pdInicial
8	day[8]	33	month[8]	58	year[8]	83	alarm_day
9	day[9]	34	month[9]	59	year[9]	84	alarm_month
10	day[10]	35	month[10]	60	year[10]	85	
11	day[11]	36	month[11]	61	year[11]	86	alarm_hour
12	day[12]	37	month[12]	62	year[12]	87	alarm_minute
13	day[13]	38	month[13]	63	year[13]	88	alarm_sec
14	day[14]	39	month[14]	64	year[14]	89	numalarm
15	day[15]	40	month[15]	65	year[15]	90	
16	day[16]	41	month[16]	66	year[16]	91	
17	day[17]	42	month[17]	67	year[17]	92	
18	day[18]	43	month[18]	68	year[18]	93	
19	day[19]	44	month[19]	69	year[19]	94	
20	day[20]	45	month[20]	70	year[20]	95	
21		46		71		96	
22		47		72		97	
23		48		73		98	
24		49		74		99	
25		50		75		100	

Tabla 8 - Contenido memoria EEPROM.

Para el circuito de control se utiliza el microcontrolador ATMEGA328P; el circuito es el siguiente:

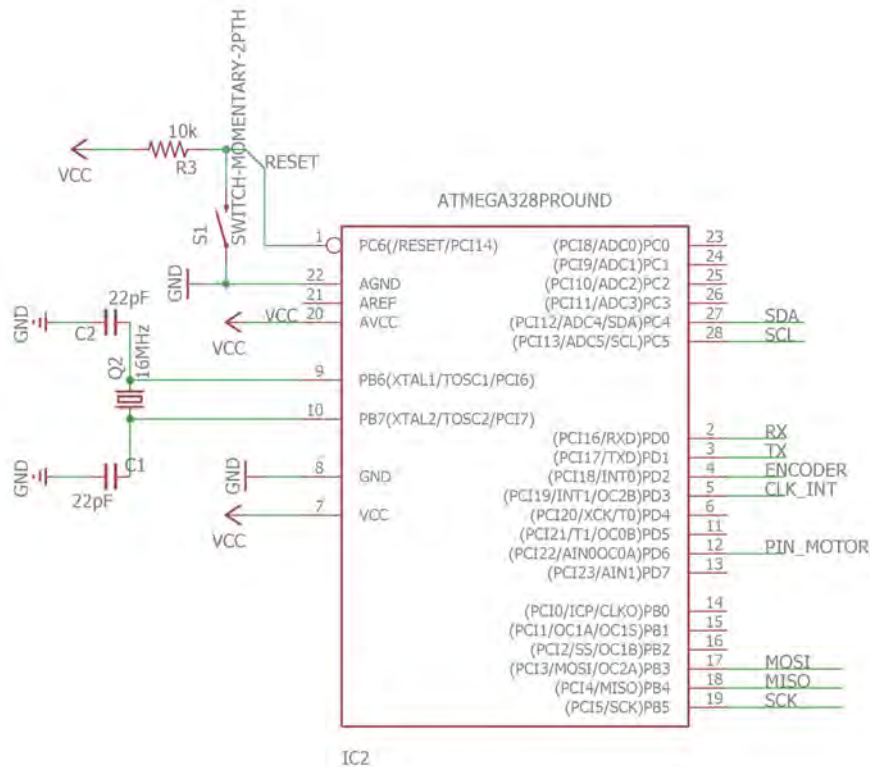


Figura 48 - Circuito de control diseñado.

- **PIN\_MOTOR** representa la señal de control del motor.
- **GND** es conexión a tierra.
- **VCC** es conexión a 5V.
- **SCL** es la línea de reloj para la comunicación I<sup>2</sup>C.
- **SDA** es la línea de comunicación serial para la comunicación I<sup>2</sup>C.
- **CLK\_INT** es el pin por el cual se crea una interrupción de la alarma (Cuando la alarma se activa, este pin se activa permitiendo el paso de corriente eléctrica).
- **ENCODER** es la señal de entrada del encoder del motor.
- **RX** es la señal de entrada de la comunicación RS232.
- **TX** es la señal de salida de la comunicación RS232.
- **MOSI**, **MISO** y **SCK** son pines para la capacidad de programación *In System Programming (ISP)*.

## 3.2 PROPUESTA DE DISEÑO DEL PROGRAMA.

Para el desarrollo del programa se utilizó el software Visual Studio 2015, y se empleó el lenguaje de programación C#. Se empleó el lenguaje de C# debido a sus prestaciones, entre las que se cuentan: Lenguaje orientado a objetos, con lo cual es posible heredar clases, sobrecarga de métodos, así como permitir la modularidad del software. Además de eso, la comunicación con puertos COM es muy sencilla, puesto que solo se debe de incluir la clase en la que se encuentran ya definidos.

El software de Visual Studio además nos permite crear fácilmente los paquetes de instalación para equipos usando el sistema operativo de Windows.

### 3.2.1 Descripción del funcionamiento.

Después de analizar el software original de la trampa de sedimentos se llegó a la creación del programa. Se muestra el programa en la Figura 49.

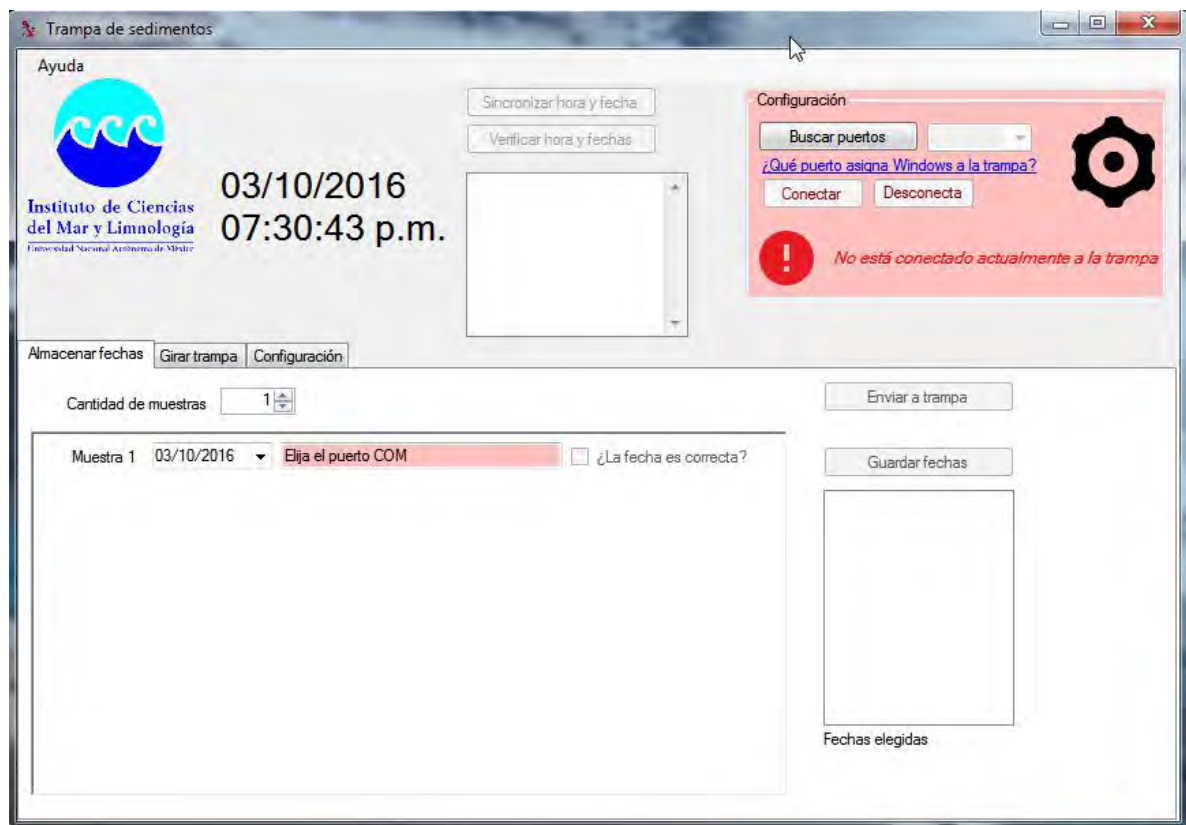


Figura 49 - Nuevo software para programar la trampa de sedimentos.

El programa consiste de tres módulos:

- **Almacenar fechas:** En esta sección se realiza toda la programación de la trampa de sedimentos.
- **Girar trampa:** En esta sección se puede realizar el giro de la trampa de sedimentos a la muestra deseada.
- **Configuración:** Permite modificar los intervalos máximos entre fechas y otras configuraciones.

Para realizar la programación de la trampa de sedimentos el primer paso es realizar la conexión de la trampa de sedimentos con el programa, esto se realiza en la sección de configuración marcada en un recuadro rojo. Como se muestra en la Figura 50, primero se presiona el botón de búsqueda de puertos; si no se sabe en qué puerto se ha conectado la trampa de sedimentos abajo hay un link directo al administrador de dispositivos desde donde se puede ver el puerto que le corresponde. Como se muestra en la Figura 51, la trampa estará conectada en COM8, después de debe de presionar “Conectar” como se muestra en la Figura 52.

Tras la conexión exitosa con el programa el recuadro rojo cambiará a un color verde y se desplegará el mensaje de conexión correcta, como es mostrado en la Figura 53.

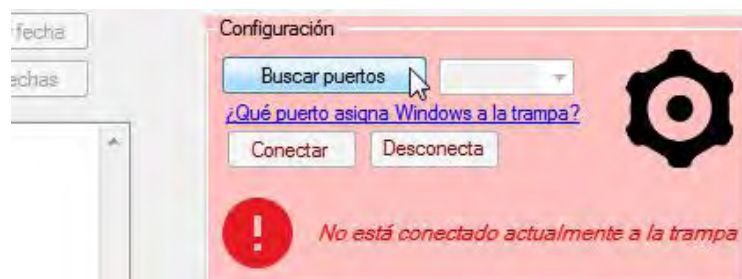


Figura 50 - Búsqueda de puertos.

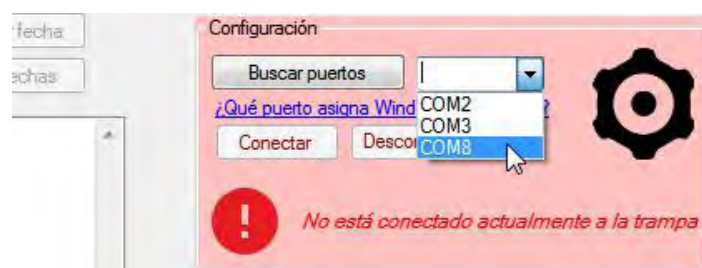


Figura 51 - Conexión al puerto COM8.

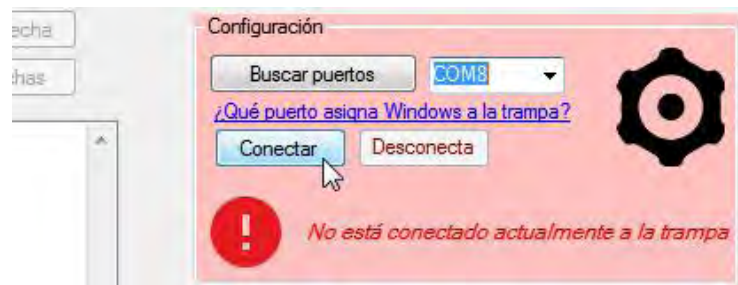


Figura 52 - Conexión al programa.



Figura 53 - Conexión exitosa del circuito de la trampa.

Posterior a esto se procede a la programación de la trampa.

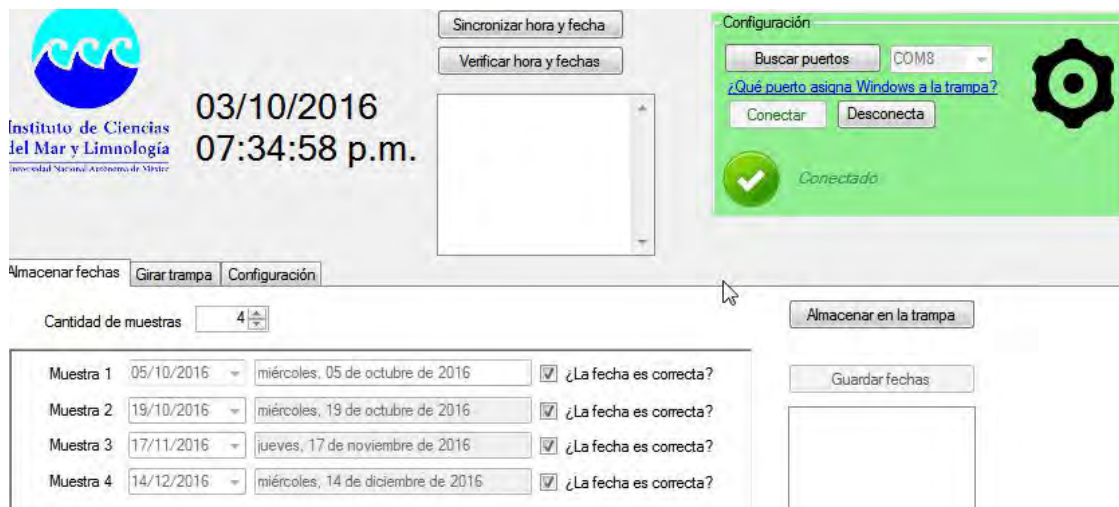


Figura 54 - Elección de fechas.

Se procede a elegir el número de muestras, en este caso elegimos 4, de forma automática se mostrarán la cantidad de muestras a programar. La fecha mínima elegible para la fecha un día posterior a la fecha de ejecución del programa, en el caso del ejemplo, la primera fecha elegible será el 04/10/2016, para la fecha máxima no hay límites en el caso de la fecha 1; para fechas

subsecuentes la fecha mínima a elegir será la fecha anterior más un día, de forma que si en la fecha 1 se programa el 15/08/2016 en la fecha 2 la fecha mínima elegible será 16/08/2016, y así subsecuentemente con las demás fechas. El intervalo entre fechas elegibles es de un mes; de forma que si la primera fecha elegida es el 25/08/2016, la segunda fecha puede ser máximo el 25/09/2016. Al terminar de elegir una fecha se debe de verificar la casilla que dice *¿La fecha es correcta?*, de esta forma se podrá habilitar la siguiente fecha a programar. Al terminar de seleccionar las fechas se procede a su almacenamiento en la trampa de sedimentos.

### 3.2.1.1 Almacenamiento de fechas en la trampa de sedimentos.

Al seleccionar la última casilla de verificación (Figura 55) ,procedemos al almacenamiento de las fechas en la trampa de sedimentos, para ello se debe presionar el botón de Almacenar en la trampa (Figura 56).

Muestra	Fecha	Día	¿La fecha es correcta?
Muestra 1	05/10/2016	miércoles, 05 de octubre de 2016	<input checked="" type="checkbox"/>
Muestra 2	19/10/2016	miércoles, 19 de octubre de 2016	<input checked="" type="checkbox"/>
Muestra 3	17/11/2016	jueves, 17 de noviembre de 2016	<input checked="" type="checkbox"/>
Muestra 4	14/12/2016	miércoles, 14 de diciembre de 2016	<input checked="" type="checkbox"/>

Figura 55 - Casilla de verificación de la fecha.

03/10/2016  
07:36:52 p.m.

Almacenar en la trampa

Muestra	Fecha	Día	¿La fecha es correcta?
Muestra 1	05/10/2016	miércoles, 05 de octubre de 2016	<input checked="" type="checkbox"/>
Muestra 2	19/10/2016	miércoles, 19 de octubre de 2016	<input checked="" type="checkbox"/>
Muestra 3	17/11/2016	jueves, 17 de noviembre de 2016	<input checked="" type="checkbox"/>
Muestra 4	14/12/2016	miércoles, 14 de diciembre de 2016	<input checked="" type="checkbox"/>

Figura 56 - Almacenando las fechas en la trampa.

Al presionar el botón de Almacenar en la trampa, debajo de éste aparecerá una barra de progreso, al completarse se podrá verificar que se almacenaron las diferentes fechas en la ventana de texto como se puede ver en el recuadro amarillo en la Figura 57.

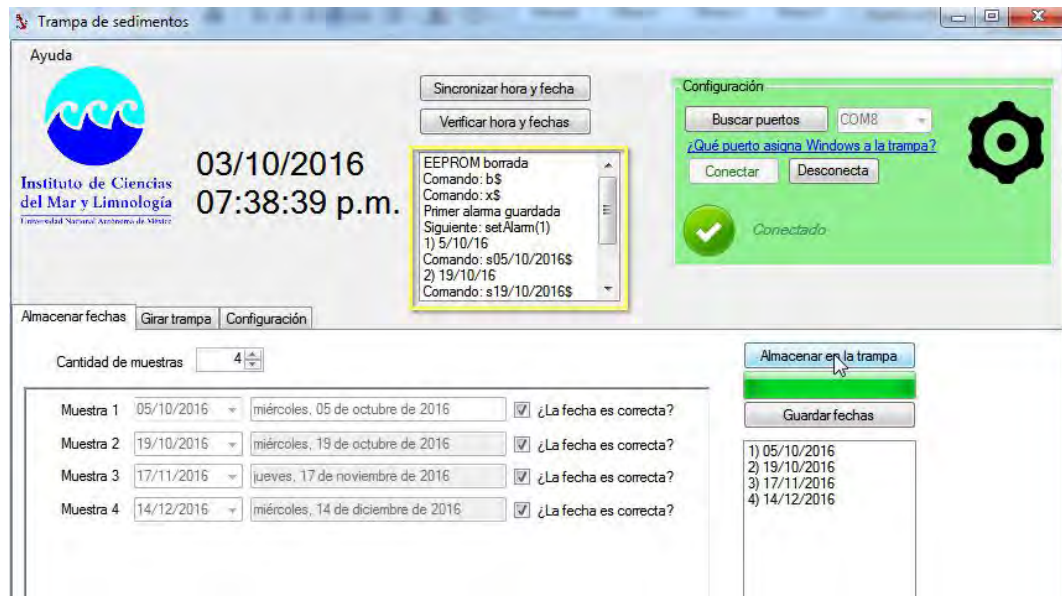


Figura 57 - Progreso en el almacenamiento de las fechas.

Para verificar que las fechas ya se encuentren almacenadas en la trampa se presiona el botón verificar hora y fechas: este botón despliega la hora de la trampa (aún no programada) y las diferentes fechas (ya programadas).

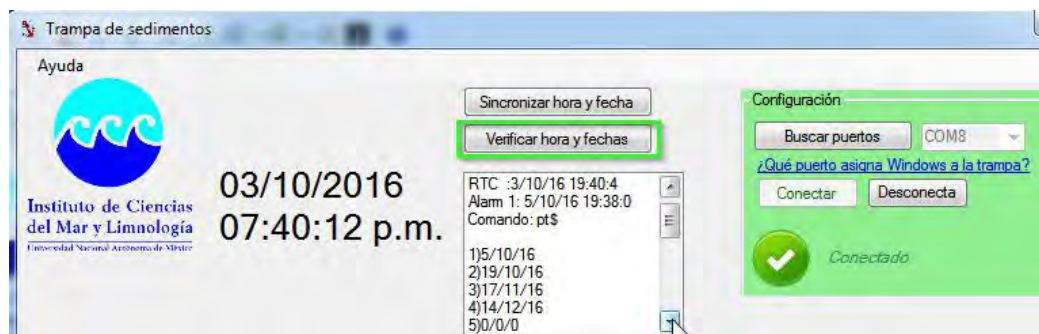


Figura 58 - Verificación de fechas.

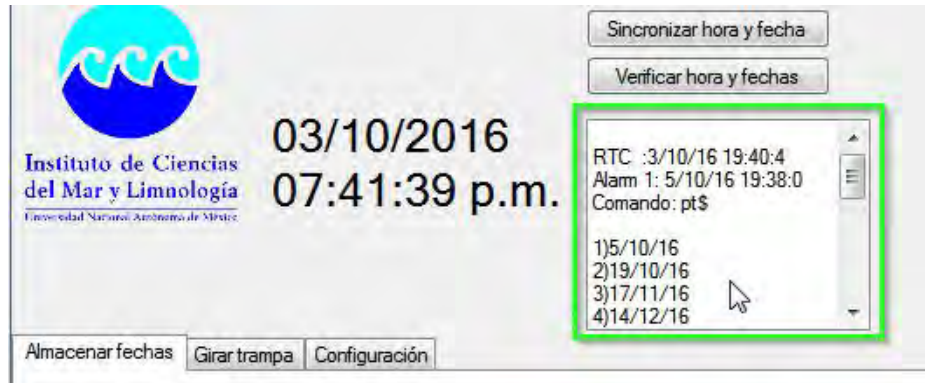


Figura 59 - Lista con las fechas leídas directamente de la trampa de sedimentos.

### 3.2.1.2 Almacenamiento de fechas en un archivo de texto.

Posterior a almacenar las fechas en la trampa de sedimentos se procede a almacenarlas en un archivo de texto; para ello sólo es necesario presionar el botón de guardar fechas como se muestra en la Figura 60.

Al presionar el botón aparecerá un cuadro de diálogo que permite elegir dónde guardar las fechas, y en caso de que exista el archivo, permite decidir si sobrescribir o no; al terminar de guardar se muestra un mensaje confirmando que se guardó correctamente el archivo de texto, tal y como se muestra en la Figura 61.

Se puede abrir el archivo y ver las fechas almacenadas en él; el archivo en automático genera un encabezado y agrega la hora y fecha en que se generó el mismo, esto se muestra en la Figura 62.

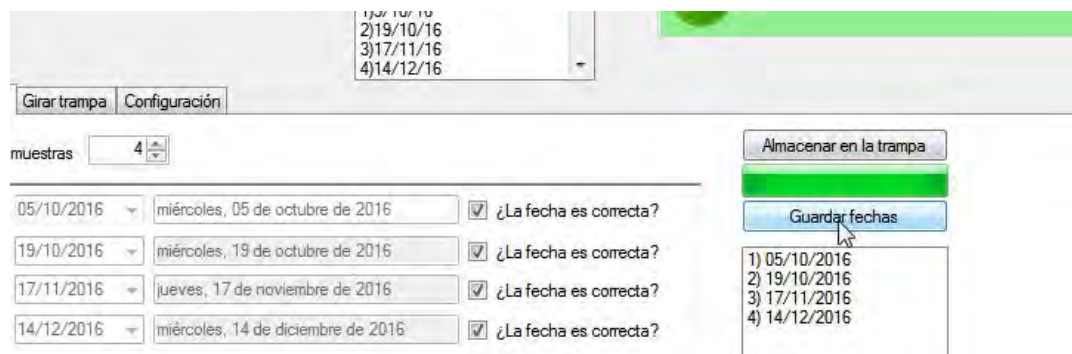


Figura 60 - Botón para guardar las fechas en un archivo de texto.



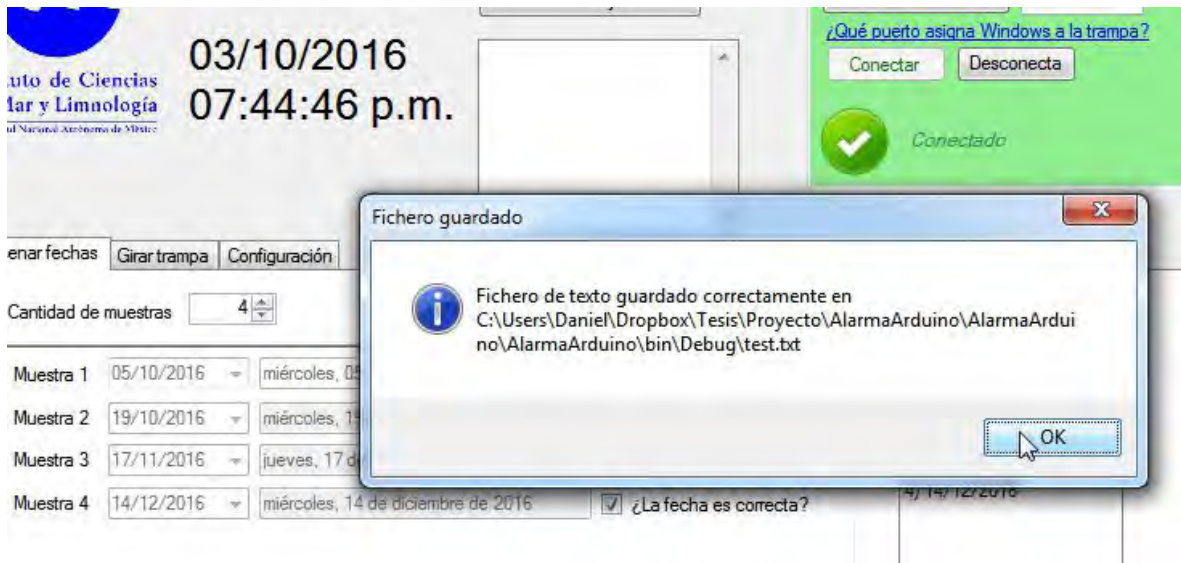


Figura 61 - Creación de archivo de texto.

```

1 Fechas de control de la trampa de sedimentos
2 Trampa de Sedimentos S/MT 234
3 Instituto de Ciencias del Mar y Limnología, UNAM
4
5 Fecha de creación del archivo:
6 03/10/2016 07:44:34 p.m.
7 Lista de fechas programadas en la trampa de sedimentos
8 1) 05/10/2016
9 2) 19/10/2016
10 3) 17/11/2016
11 4) 14/12/2016

```

Figura 62 - Archivo de texto generado.

### 3.3 ANÁLISIS DE BATERÍAS.

#### 3.3.1 Vida de las baterías y consumo de corriente.

La capacidad de las baterías se mide en miliamperes-hora (abreviado mAh), la cual es una medida de cuántas horas una batería puede sostener una descarga de corriente constante; por ejemplo, una batería de 800 mAh, puede proveer 800 mA de corriente por una hora. Sin embargo, en la realidad, a una descarga continua de corriente se tiene una menor capacidad de la batería. Por eso es que cada proveedor ofrece gráficas de descarga para diferentes corrientes como las siguientes:

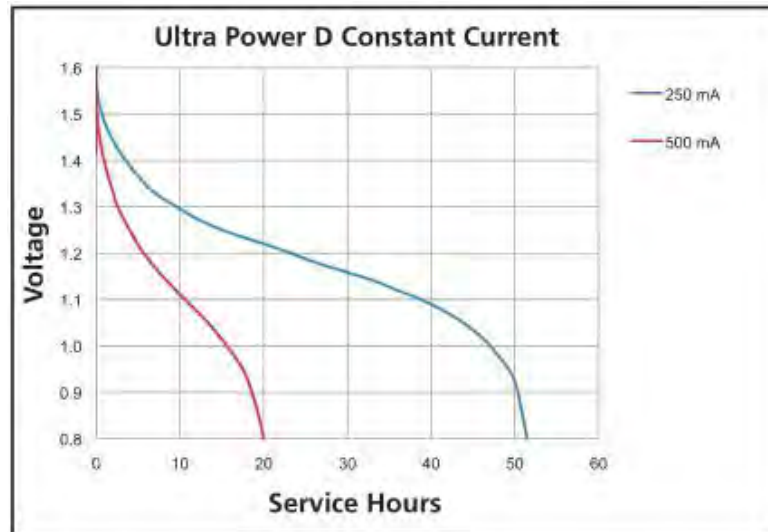


Figura 63 - Gráfica de descarga Duracell, 250mA y 500mA.



Figura 64 - Gráfica de descarga Duracell a 250mA, a 0°C y a 21°C.

Como podemos ver, a un mayor consumo de corriente, una menor capacidad de la batería; sin embargo, también hay que notar que las baterías también son afectadas por la temperatura; a una menor temperatura, una menor capacidad de la batería.

### 3.3.2 Consumo de corriente teórico

Con el consumo de corriente promedio y la capacidad de la batería se puede calcular el tiempo que nuestro circuito puede permanecer alimentado.

Tomando como referencia las baterías del tipo Alcalinas (Duracell Coppertop<sup>16</sup>) con el número de producto MN1300, tenemos una capacidad de carga de 15,000 mAh, a un voltaje de 1.2V y a una temperatura de 21°C.

El consumo de corriente se dividirá en tres diferentes elementos:

1. Circuito de regulación.
2. Circuito de control.
3. Circuito de potencia.

### Circuito de regulación.

El regulador empleado es el **MCP7805CTG**; este integrado nos ofrece una regulación a +5V y hasta 1A de corriente. De su hoja de especificaciones podemos destacar el consumo de corriente conocido como *quiescent current* (o corriente de reposo)

**ELECTRICAL CHARACTERISTICS** ( $V_{in} = 10\text{ V}$ ,  $I_O = 500\text{ mA}$ ,  $T_J = T_{low}$  to  $125^\circ\text{C}$  (Note 1), unless otherwise noted)

Characteristic	Symbol	MC7805B, NCV7805B			MC7805C			Unit
		Min	Typ	Max	Min	Typ	Max	
Output Voltage ( $T_J = 25^\circ\text{C}$ )	$V_O$	4.8	5.0	5.2	4.8	5.0	5.2	Vdc
Output Voltage ( $5.0\text{ mA} \leq I_O \leq 1.0\text{ A}$ , $P_D \leq 15\text{ W}$ ) $7.0\text{ Vdc} \leq V_{in} \leq 20\text{ Vdc}$ $8.0\text{ Vdc} \leq V_{in} \leq 20\text{ Vdc}$	$V_O$	- 4.75	- 5.0	- 5.25	4.75 -	5.0 -	5.25 -	Vdc
Line Regulation (Note 4) $7.5\text{ Vdc} \leq V_{in} \leq 20\text{ Vdc}$ , $1.0\text{ A}$ $8.0\text{ Vdc} \leq V_{in} \leq 12\text{ Vdc}$	$Reg_{line}$	- -	5.0 1.3	100 50	- -	0.5 0.8	20 10	mV
Load Regulation (Note 4) $5.0\text{ mA} \leq I_O \leq 1.0\text{ A}$ $5.0\text{ mA} \leq I_O \leq 1.5\text{ A}$ ( $T_A = 25^\circ\text{C}$ )	$Reg_{load}$	- -	1.3 0.15	100 50	- -	1.3 1.3	25 25	mV
Quiescent Current	$I_B$	-	3.2	8.0	-	3.2	6.5	mA

Figura 65 - Corriente de reposo<sup>17</sup> MC7805CTG.

### Circuito de control

El circuito de control consta básicamente del microcontrolador ATMEGA328P; el consumo de corriente no está explícitamente indicado en su hoja de especificaciones, puesto que, su consumo de corriente depende de su configuración. A una velocidad de 16MHz, y a 5V, puede consumir 20mA en promedio. Sin embargo, puede ser configurado para consumir corriente en el orden de microamperes reduciendo la velocidad de operación, el voltaje de

<sup>16</sup> <http://www.alliedelec.com/m/d/21cc16a9fe1a0f147764266cdfb21305.pdf>

<sup>17</sup> Tomado de: <http://www.mouser.com/ds/2/308/MC7800-D-94905.pdf>

alimentación, la cantidad de periféricos, así como la configuración de registros SFR que controlan el consumo de corriente.

Frequency	mA @ 5V	mA @ 3.3V
1MHz	6.75	0.92
8MHz	11.68	4.16
16MHz (LP)	16.32	7.40
16MHz (FS)	17.52	7.94
20MHz (LP)	19.71	8.86
20MHz (FS)	21.12	9.46

(LP) Indicates the Low Power Crystal Oscillator  
(FS) Indicates the Full Swing Crystal Oscillator

Figura 66 - Consumo de corriente a diferentes Voltajes y Corrientes.

### Circuito de potencia

Para la parte de potencia se emplea el integrado L298, el cual es un doble puente H, con una corriente máxima de 2A, y hasta 42V de alimentación.

#### ELECTRICAL CHARACTERISTICS ( $V_S = 42V$ ; $V_{SS} = 5V$ , $T_j = 25^\circ C$ ; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_S$	Supply Voltage (pin 4)	Operative Condition	$V_{RH} + 2.5$		46	V
$V_{SS}$	Logic Supply Voltage (pin 9)		4.5	5	7	V
$I_S$	Quiescent Supply Current (pin 4)	$V_{en} = H$ ; $I_L = 0$ $V_i = L$ $V_i = H$		13 50	22 70	mA mA
		$V_{en} = L$ $V_i = X$			4	mA
$I_{SS}$	Quiescent Current from $V_{SS}$ (pin 9)	$V_{en} = H$ ; $I_L = 0$ $V_i = L$ $V_i = H$		24 7	36 12	mA mA
		$V_{en} = L$ $V_i = X$			6	mA

Figura 67 - Consumo de corriente en reposo de L298.

### 3.4 PROGRAMACIÓN DEL MICROCONTROLADOR.

Para la programación del circuito de control se utilizó un programador de la compañía Sparkfun<sup>18</sup>; se muestra el programador en la Figura 68. Por medio de este programador podremos programar cualquier microcontrolador de Atmel con una memoria flash de 64KB o menos (el microcontrolador empleado tiene 32 KB de memoria flash).

<sup>18</sup> <https://www.sparkfun.com/products/9825>

### 3.4.1 Componentes del programador.

El programador consiste en:

- **Conector USB** - Emplea un cable tipo mini USB, permite la alimentación y el envío de información del programador.
- **Header ISP 2x5** - Junto con el cable de programación, ésta es la conexión que permite la programación de nuestro microcontrolador.
- **Switch de alimentación** - Si nuestro microcontrolador no se encuentra alimentado, podemos alimentarlo al activar este switch; si el microcontrolador se encuentra alimentado, es necesario apagar este interruptor.
- **ATtiny2313** - Circuito integrado que convierte los comandos entre el USB y el programador SPI.
- **Buffer 74AC125** - Circuito integrado con la finalidad de proteger al programador.

Podemos observar los diferentes componentes del programador en la Figura 69.

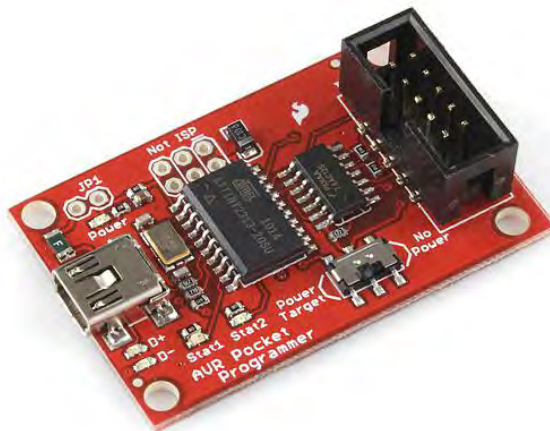


Figura 68 - Programador ISP marca Sparkfun.

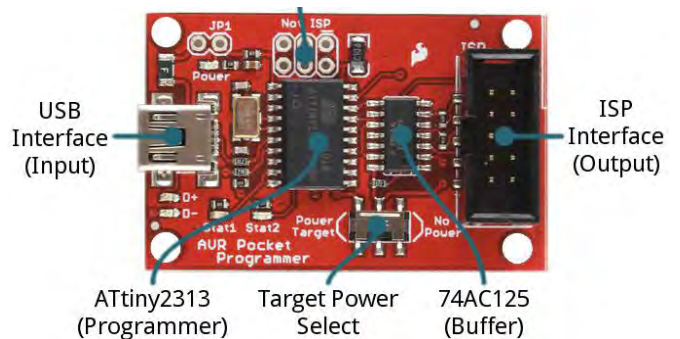


Figura 69 - Componentes de programador ISP.

Para conectar las señales entre dispositivos se cuenta con dos conectores estandarizados, uno de 6 pines (2x3) y uno de 10 pines (2x5), como se muestra en la Figura 70.



Figura 70 - Conectores de ISP.

### 3.4.2 Conector de ISP en el circuito

En el circuito fue implementado un conector ISP para poder programarlo en caso de que sea requerido. Este circuito de ISP tiene un conector de 6 pines (2x3) y se muestra en la Figura 71.

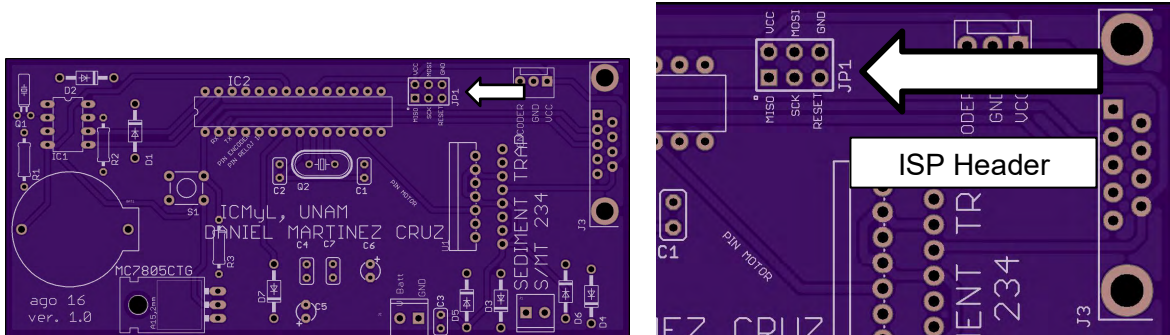


Figura 71 - Ubicación de header ISP.

### 3.4.3 Programación del circuito.

Para la programación existen tres métodos:

- IDE de Arduino
- Línea de comandos
- Atmel Studio.

#### 3.4.3.1 Programación mediante Arduino IDE.

Abrimos el IDE de Arduino, y vamos a Herramientas>Programador y elegimos USBtinyISP como se muestra en la Figura 72.

Para subir nuestro código vamos a Archivo y subir usando programador. Este proceso tarda aproximadamente un minuto. Nota: Al realizar ésta operación es borrado el bootloader de nuestro microcontrolador. El procedimiento se muestra en la Figura 73.

#### 3.4.3.2 Programación mediante línea de comandos.

Para la programación por medio de línea de comandos es necesario generar el archivo **.HEX**

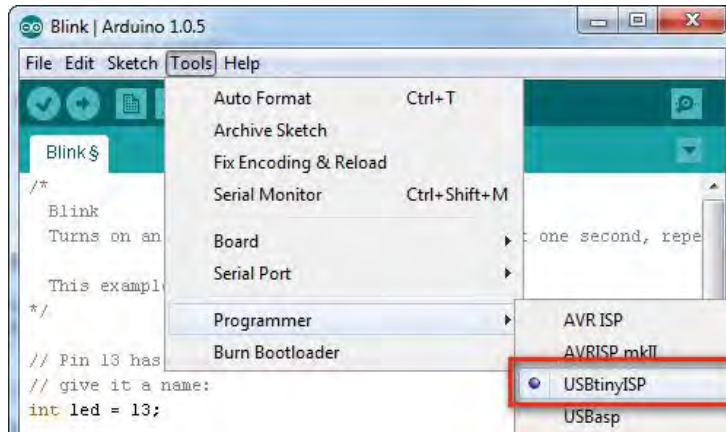


Figura 72 - Configuración para programación ISP.

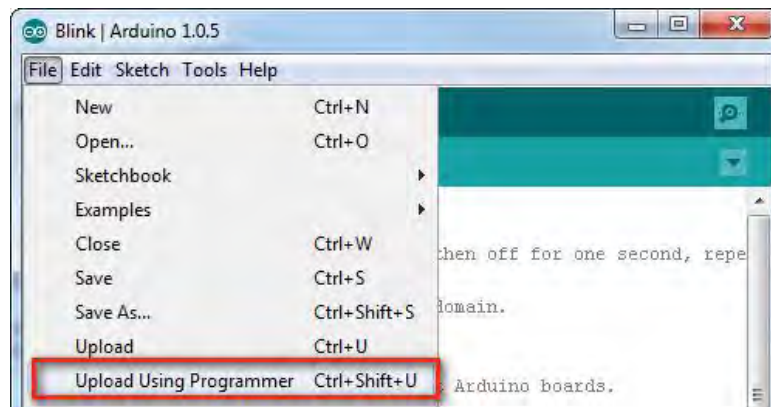


Figura 73 - Carga del código vía ISP.

Se ejecuta la línea de comandos de Windows con el programador conectado, y se ejecuta el siguiente comando de verificación:

```
avrdude -c usbtiny -p atmega328p
```

Con lo cual se muestra una ventana similar a la mostrada en la Figura 74.

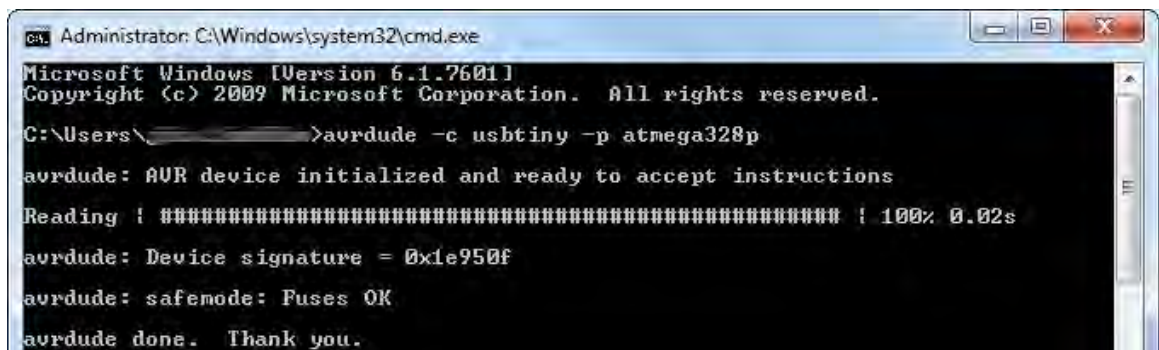


Figura 74 - Programación vía CMD.

Ahora que se ha verificado que todo está en orden, se puede cargar el archivo .HEX con el siguiente comando:

```
avrdude -c usbtiny -p atmega328p -U flash:w:NAME.hex
```

Con NAME como el nombre de nuestro archivo HEX. La opción `-c` nos dice qué programador se usará, (se usa el `usbtiny`), con `-p` se elige el micro a programar, con la opción `-U` se indica que se necesita leer y escribir la memoria, luego le decimos qué escribiremos en la memoria `flash`; con `w` se escribe, y luego se dice que archivo se escribirá. Este procedimiento se muestra en la Figura 75.

```
Administrator: C:\Windows\system32\cmd.exe
C:\>avrdude -c usbtiny -p atmega328p -U flash:w:blink.hex
avrdude: AVR device initialized and ready to accept instructions

Reading | ##### | 100% 0.02s
avrdude: Device signature = 0x1e950f
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
        To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "blink.hex"
avrdude: input file blink.hex auto detected as Intel Hex
avrdude: writing flash (3098 bytes):

Writing | ##### | 100% 9.67s

avrdude: 3098 bytes of flash written
avrdude: verifying flash memory against blink.hex:
avrdude: load data flash data from input file blink.hex:
avrdude: input file blink.hex auto detected as Intel Hex
avrdude: input file blink.hex contains 3098 bytes
avrdude: reading on-chip flash data:

Reading | ##### | 100% 5.51s

avrdude: verifying ...
avrdude: 3098 bytes of flash verified

avrdude: safemode: Fuses OK
avrdude done. Thank you.
```

Figura 75 - Código cargado vía consola de comandos.

### 3.4.3.3 Programación mediante Atmel Studio

Atmel Studio es un entorno de desarrollo basado en Visual Studio, y a diferencia del IDE de Arduino, Atmel Studio permite un mayor control sobre el proyecto a diseñar. Además, Atmel permite la importación de proyectos iniciados en el IDE de Arduino, y los convierte en un proyecto C++.



Para la programación mediante Atmel Studio se requiere de la importación de un proyecto de Arduino, o iniciar un proyecto nuevo. Tras cargar correctamente el proyecto, se procede a la compilación como se muestra en la Figura 76.

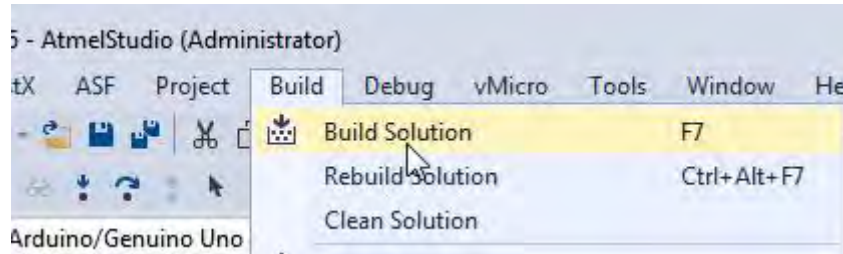


Figura 76 - Compilación mediante Atmel Studio.

Si la compilación resulta exitosa, Atmel Studio mostrará la siguiente imagen (Figura 77).

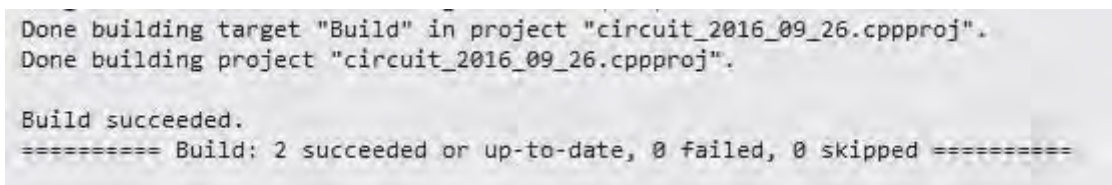


Figura 77 - Compilación correcta.

Tras compilar correctamente el proyecto, hay que configurar el programador. Como se muestra en la Figura 78, ir a Herramientas > Herramientas externas

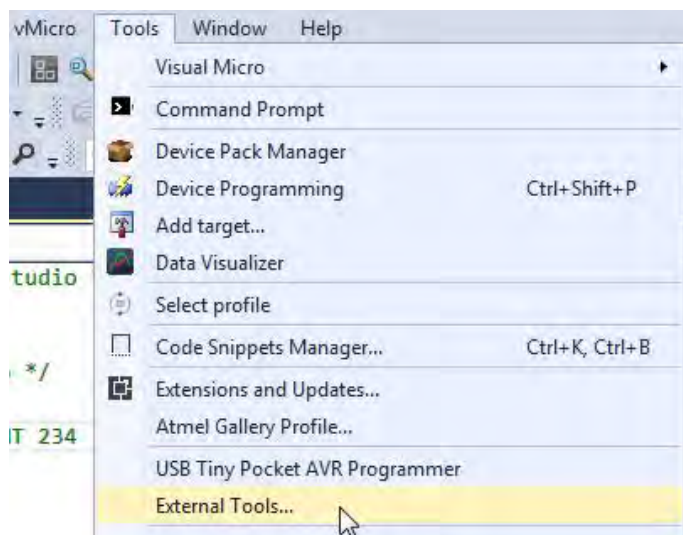


Figura 78 - Configuración de programador como Herramienta Externa.

Se abre la siguiente ventana:

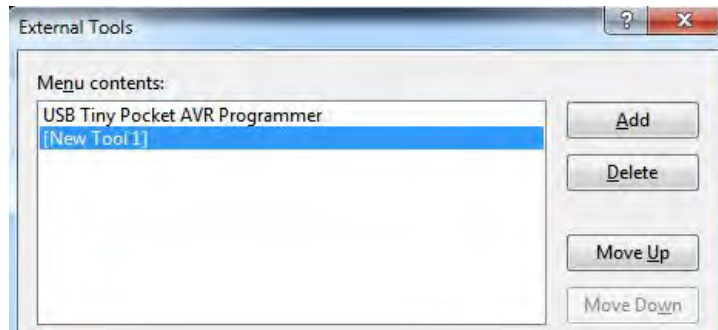


Figura 79 - Añadir herramienta externa.

Primero se debe presionar el botón de Add, con lo cual se crea la siguiente configuración:

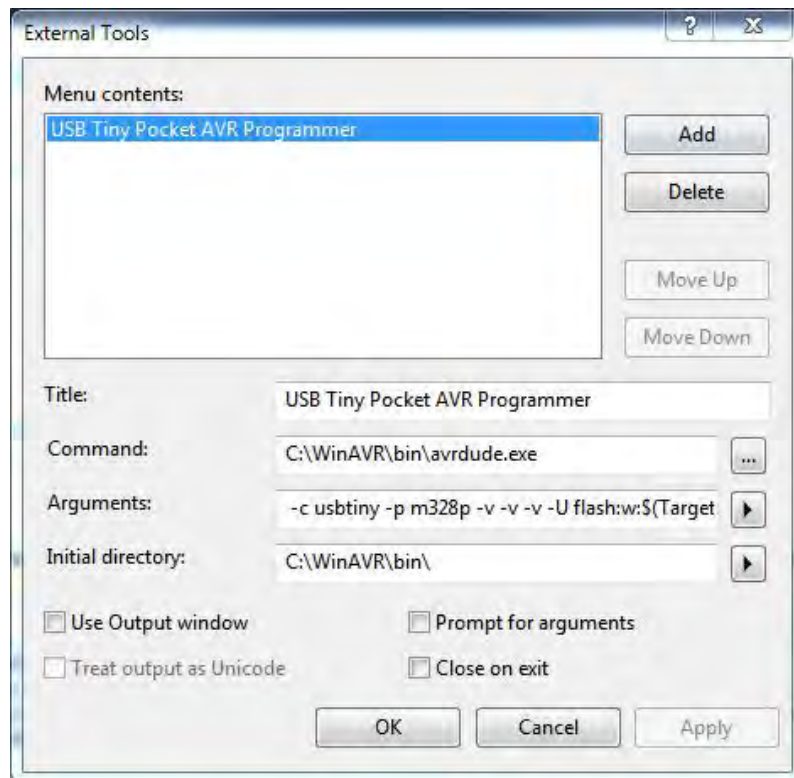


Figura 80 - Parámetros de Herramienta externa.

Title: **USB Tiny Pocket AVR Programmer**  
Command: **C:\WinAVR\bin\avrdude.exe**  
Arguments:  
**-c usbtiny -p m328p -v -U flash:w:\$(TargetDir)\$(TargetName).hex:i**  
Initial directory: **C:\WinAVR\bin\**

La configuración se tomó del manual de usuario de AVRDUDE<sup>19</sup>

No hay que seleccionar ninguna casilla. Tras realizar esto, nuestro programador se encuentra correctamente configurado. Para cargar nuestro código hay que ir a Herramientas y elegir el **USB Tiny Pocket AVR Programmer**.

## 4 RESULTADOS.

En ésta tesis se diseñó, se fabricó y se ensambló un circuito electrónico que sirviera de reemplazo para el circuito de control de las trampas de sedimentos de la marca KUM modelo S/MT 234.

De igual forma se desarrolló una interfaz de programación que sirviera como reemplazo a la antigua interfaz (originalmente para MS-DOS), de manera que la nueva versión se puede utilizar en equipos de cómputo con una versión de Windows XP o superior. Se procede a realizar los cálculos correspondientes al consumo eléctrico del circuito.

### 4.1 CÁLCULOS DE CONSUMO ELÉCTRICO

En la Tabla 9 se muestra una tabla con los diferentes tipos de consumo eléctrico que se pueden encontrar tras la implementación del circuito.

Elemento	Consumo mínimo	Consumo típico [mA]	Consumo Máximo [mA]
Regulador	3.2	3.2	6.5
Microprocesador	0.1	18	22
Puente H	4	4	36
Total	7.3	25.2	64.5

Tabla 9 - Diferentes escenarios de consumo.

Tomando como referencia las baterías del tipo Alcalinas (Duracell Coppertop<sup>20</sup> número de producto MN1300), tenemos una capacidad de carga de 15,000 mAh, a un voltaje de 1.2V y a una temperatura de 21°C. Considerando esto, se presentan los diferentes escenarios posibles.

<sup>19</sup> [http://www.nongnu.org/avrdude/user-manual/avrdude\\_4.html#Option-Descriptions](http://www.nongnu.org/avrdude/user-manual/avrdude_4.html#Option-Descriptions)

<sup>20</sup> <http://www.alliedelec.com/m/d/21cc16a9fe1a0f147764266cdfb21305.pdf>

Consumo mínimo	Consumo promedio	Consumo máximo
$\frac{15,000mAh}{7.3mA} = 2055hrs$	$\frac{15,000mAh}{25.2mA} = 595hrs$	$\frac{15,000mAh}{64.5mA} = 233hrs$

Tabla 10 - Cálculo de tiempo de funcionamiento promedio.

Con el consumo de corriente promedio y la capacidad de la batería se puede calcular el tiempo que nuestro circuito puede permanecer alimentado. Si se deseara que nuestra trampa funcione al menos 6 meses se tiene que:

- 6 meses= 183.5 días.
- 183.5 días= 4,404 hrs.

Así que el circuito debe de funcionar por lo menos cuatro mil cuatrocientas horas. Considerando esto, tenemos el siguiente cálculo.

$$\frac{15,000mAh}{4404h} = 3.405mA$$

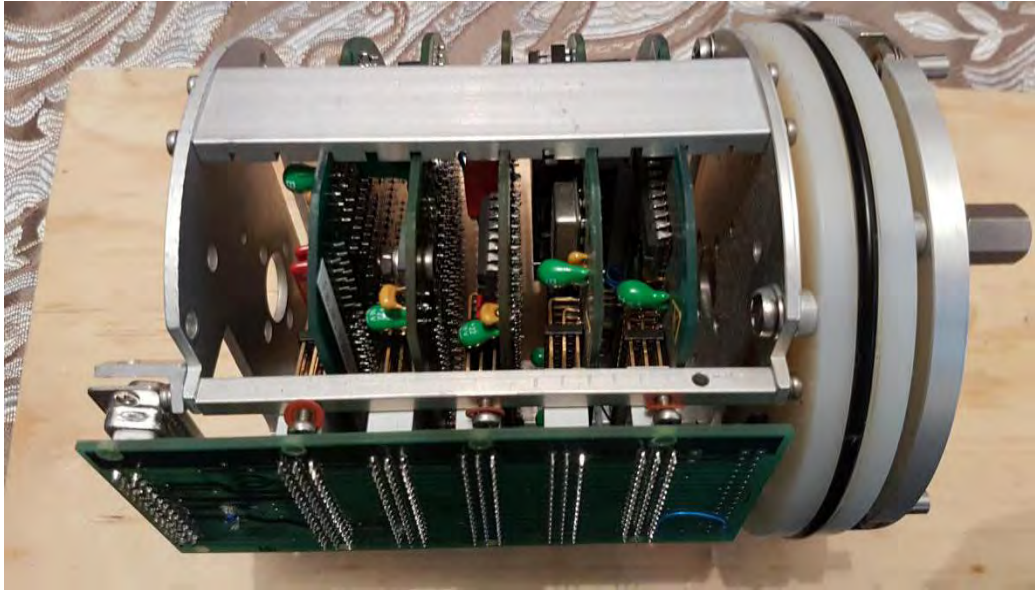
Para un consumo de corriente continua, podemos disponer de hasta 3.4mA de forma continua, asegurando un funcionamiento de hasta 6 meses.

## 4.2 CIRCUITO RE-DISEÑADO IMPLEMENTADO.

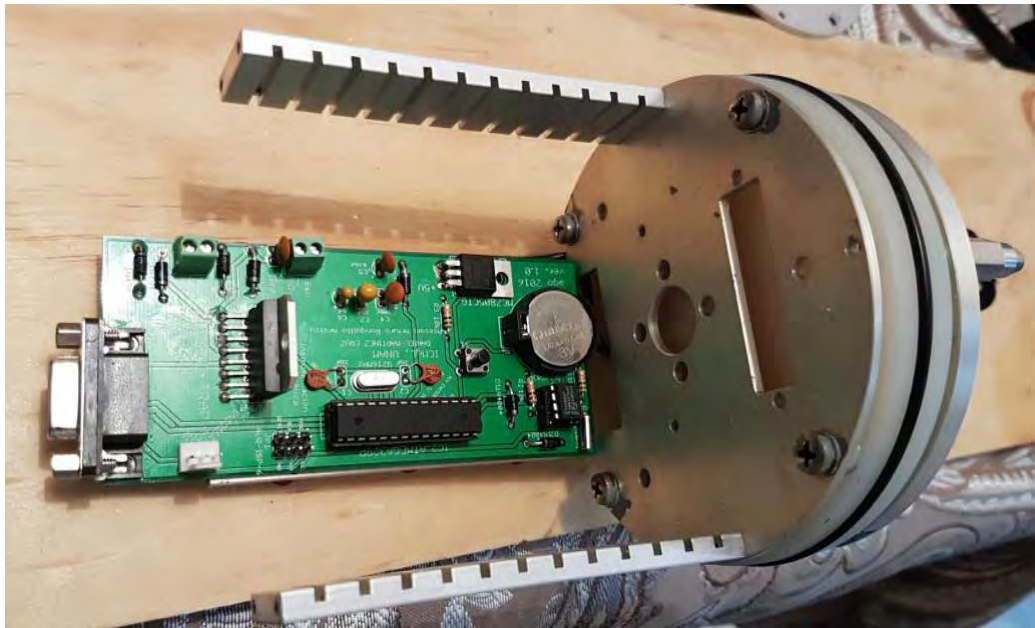
En la Figura 81 se muestra el circuito de control original, y en la Figura 82 se muestra el circuito re-diseñado. Se destaca la diferencia volumétrica del diseño.

En las Figuras 83 y 84 se muestran respectivamente las caras superior e inferior de la PCB diseñada. Y en la Figura 85 se muestra la PCB con los diferentes componentes soldados en la misma.

En la Figura 86 se muestra una comparación entre la PCB original (arriba) contra la PCB re-diseñada, se destaca la similitud de tamaño entre las mismas, puesto que esta fue la base para realizar su diseño.



*Figura 81 - Circuito de control original.*



*Figura 82 - Reducción en el tamaño con el circuito diseñado.*

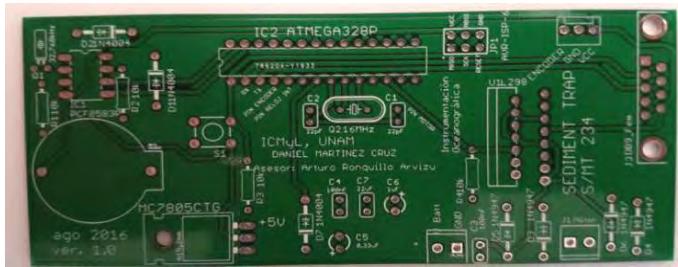


Figura 83 - Tarjeta PCB, cara superior.

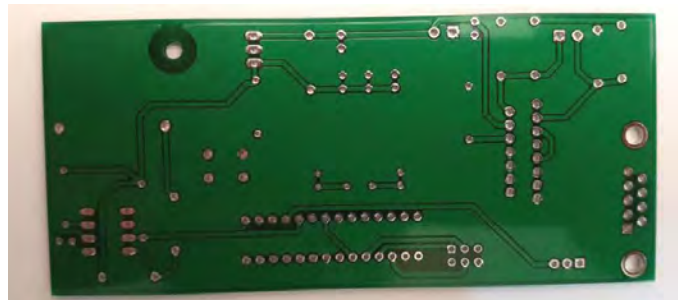


Figura 84 - Tarjeta PCB, cara inferior.



Figura 85 - PCB con todos los componentes soldados.

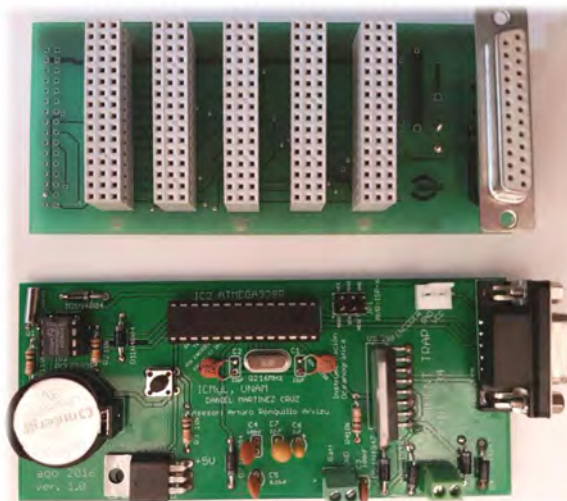


Figura 86 - Comparación de tamaño entre PCB y backplane original.

## CONCLUSIONES.

La nueva interfaz de programación permite su uso en equipos con una versión de Windows XP y posterior. Se dio especial importancia al sistema operativo Windows XP ya que, por experiencia personal, en los distintos buques con los que cuenta el Instituto de Ciencias de Mar, el buque oceanográfico Puma y el Buque oceanográfico Justo Sierra, el sistema operativo principal de los diferentes equipos es Windows XP.

La interfaz diseñada permite al usuario desplazarse a través del programa empleando teclado y mouse, a diferencia de la versión original que sólo permitía el ingreso de información por medio del teclado.

Además de esto, este nuevo programa permite el almacenamiento de la información relativa a las fechas de programación con un formato de archivo de texto que es visible y editable con cualquier programa de edición de texto (formato TXT); esto es relevante, ya que la versión original del software de control almacenaba la información en un formato de archivo no editable con ningún software. Solo permitía la impresión del archivo por medio del software original, de forma que si se quería revisar la información era necesario instalar el programa original.

Como es posible ver, el circuito diseñado reduce el tamaño del circuito original en más del 80% del espacio volumétrico, puesto que en lugar de emplear las cinco tarjetas originales más el backplane, el circuito rediseñado solo utiliza el espacio asignado al backplane. Al emplear una sola tarjeta, se reduce la dificultad de mantenimiento. Menos tarjetas significan menos componentes por revisar, además de eso el circuito fue diseñado de forma modular, de forma que se tienen localizadas diferentes zonas (Circuito de alimentación, circuito de reloj, circuito de control y circuito de potencia). Los principales circuitos integrados cuentan con una base que permite reemplazar rápidamente los circuitos si se encuentran dañados; además de esto, el circuito cuenta con un conector ISP que permite la programación del microcontrolador para permitir la modificación del firmware, en caso de ser requerido.

La reducción en el tamaño del circuito permite almacenar más baterías, con lo que es posible asegurar un mayor tiempo de funcionamiento. No es necesario cambiar la base original del circuito para adecuarlo al nuevo, pero sí se requiere de cambiar el conector original de 25 pines. La base tiene las mismas medidas que el circuito original; y del conector de 25 pines, solo se emplean 4 cables de conexión, de forma que el reemplazo se puede realizar de una forma sencilla.

## RECOMENDACIONES.

El diseño del circuito puede ser optimado empleando un regulador del tipo bomba de carga, ya que este tipo de regulador nos permite obtener un voltaje fijo desde una fuente de voltaje menor con un consumo de corriente muy bajo. Estos reguladores entregan 5V con solo 2.5V de entrada; y la corriente de reposo es mínima, puesto que estos reguladores pueden ser apagados mientras no se utilicen. Se pueden activar mediante un pulso, como sería el pulso generado por el RTC.

De igual forma, para reducir el consumo de corriente eléctrica en el diseño original no se incorporó ningún LED indicador de encendido; se puede colocar un LED con un jumper, de manera que mientras uno realice pruebas, el LED permanezca encendido.

Las trampas de sedimentos son sensibles a la inclinación, de manera que si la trampa se encuentra inclinada, la muestra puede estar sesgada; se puede implementar la medición de ángulo mediante un acelerómetro, que almacene el valor registrado en cada eje y con base en la información de éste se tome la decisión de si tomar o no esa muestra como válida.



# BIBLIOGRAFÍA.

## Artículos.

Fowler, SW, & Knauer, GA. (1986). Role of large particles in the transport of elements and organic compounds through the oceanic water column.

*Progress in Oceanography*.

Obtenido de <http://www.sciencedirect.com/science/article/pii/0079661186900327>

Silver, MW, & Gowing, MM. (1991). The “particle” flux: origins and biological components.

*Progress in Oceanography*.

Obtenido de <http://www.sciencedirect.com/science/article/pii/0079661191900079>

Allredge, AL, & Silver, MW. (1988). Characteristics, dynamics and significance of marine snow.

*Progress in oceanography*.

Obtenido de <http://www.sciencedirect.com/science/article/pii/0079661188900535>

Lane, P. V. Z., Smith, S. L., Urban, J. L., & Biscaye, P. E. (1994). Carbon flux and recycling associated with zooplanktonic fecal pellets on the shelf of the Middle Atlantic Bight.

*Deep Sea Research Part II: Topical Studies in Oceanography*, 41(2-3), 437–457.

Doi:10.1016/0967-0645(94)90031-0

Buesseler, KO, Antia, AN, & Chen, M. (2007). An assessment of the use of sediment traps for estimating upper ocean particle fluxes.

*Journal of Marine ...*

Obtenido

de

<http://www.ingentaconnect.com/content/jmr/jmr/2007/00000065/00000003/art00002>

## Libros.

Oxer, J., & Blemings, H. (2010). *Practical Arduino*.

Springer.

Doi:10.1007/978-1-4302-2478-5

Trevennor, A. (2012). *Practical AVR Microcontrollers*.

Springer.

Doi:10.1007/978-1-4302-4447-9

Purdum, J., & Purdum, J. (2012). *Arduino Libraries* (pp. 211–230).

springer.

Doi:10.1007/978-1-4302-4777-7\_12

## Notas de aplicación.

ATMEL Corporation. Appl. Note (2005). *AVR103: Using the EEPROM Programming Modes*.

Obtenido de: <http://www.atmel.com/images/doc2578.pdf>

Atmel Corporation. Appl. Note (2008). *AVR910: In-System Programming*.

Obtenido de: <http://www.atmel.com/images/doc2578.pdf>

Svendli, OJ. (2013). *Atmel's Self-Programming Flash Microcontrollers*.

Obtenido de <http://www.atmel.com/images/doc2464.pdf>

## Hojas de especificaciones.

Atmel Corporation. (2016). *8-bit AVR Microcontrollers ATmega328/P DATASHEET SUMMARY, (rev A 06-2016), 24 páginas*.

Obtenido el el 23 de Agosto de 2016, de:

[http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P\\_Summary.pdf](http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Summary.pdf)

Atmel Corporation. (2016). *8-bit AVR Microcontrollers ATmega328/P DATASHEET COMPLETE, (rev A 06-2016), 444 páginas*.

Obtenido el el 23 de Agosto de 2016, de:

[http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_datasheet.pdf)

NXP Semiconductors (2010) *PCF8583 Clock and calendar with 240 x 8-bit RAM, (Rev. 06 — 6 October 2010)*.

Obtenido el 23 de Agosto de 2016, de:

[http://www.nxp.com/documents/data\\_sheet/PCF8583.pdf](http://www.nxp.com/documents/data_sheet/PCF8583.pdf)

ST Microelectronics (2000). *L298 Dual Full Bridge Driver*.

Obtenido el 23 de Agosto de 2016, de:

<http://www.st.com/content/ccc/resource/technical/document/datasheet/82/cc/3f/39/0a/29/4d/f0/CD00000240.pdf/files/CD00000240.pdf/jcr:content/translations/en.CD00000240.pdf>

## Páginas Web.

NOAA, *What Is Marine Snow?* (n.d.)

Obtenido el 23 de Agosto de 2016, de:

<http://oceanservice.noaa.gov/facts/marinesnow.html>

UPIICSA IPN, *Arquitectura CISC y RISC*. (n.d.).

Obtenido el 23 de Agosto de 2016, de:

[http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P\\_terminados/PolilibroFC/Unidad\\_VI/Unidad%20VI\\_31.htm](http://www.sites.upiicsa.ipn.mx/polilibros/portal/Polilibros/P_terminados/PolilibroFC/Unidad_VI/Unidad%20VI_31.htm)

Picker, L. (2016, Enero 19). *Microchip Technology to Buy Atmel for Nearly \$3.6 Billion*.

Obtenido del 31 de Agosto de 2016, de:

<http://www.nytimes.com/2016/01/20/business/dealbook/microchip-technology-to-buy-atmel-for-nearly-3-6-billion.html>

Kushner, D. (2011, Octubre 26). *The Making of Arduino*.

Obtenido del 31 de Agosto de 2016, de: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino>

Atmel Corporation. *Advanced Architecture Optimizes the Atmel AVR CPU*. (n.d.).

Obtenido el 23 de Agosto de 2016, de:

[http://www.atmel.com/technologies/cpu\\_core/avr.aspx](http://www.atmel.com/technologies/cpu_core/avr.aspx)

# ANEXOS.

## ANEXO 1 FOTOGRAFÍAS DEL CIRCUITO ORIGINAL.

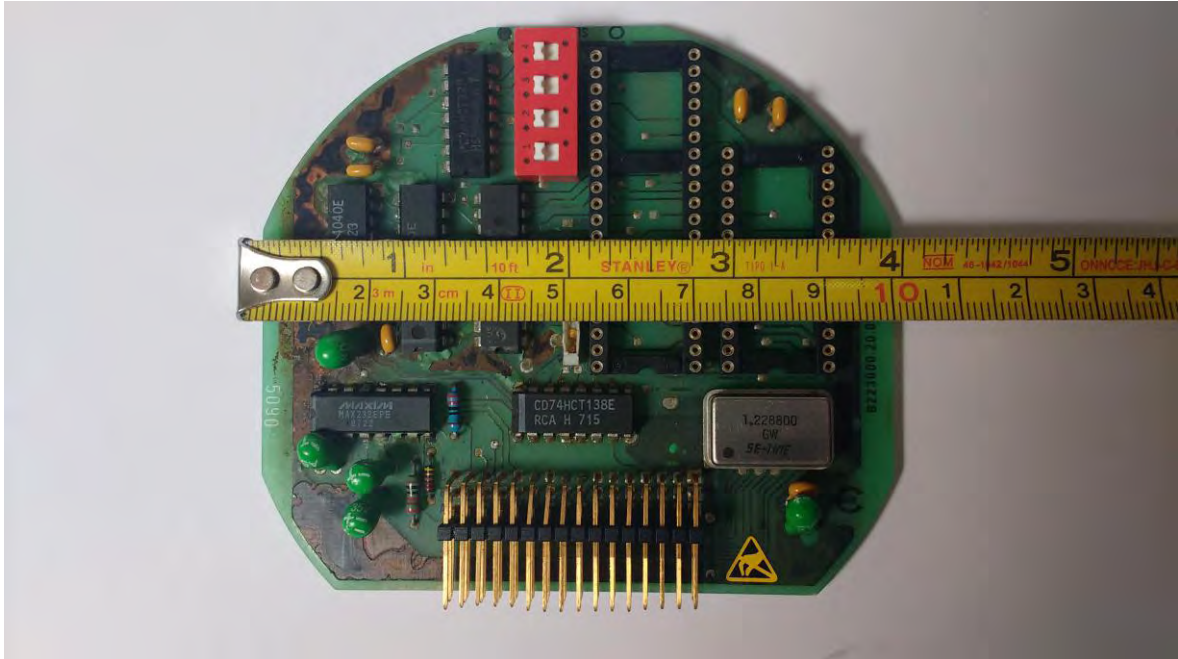


Imagen 1 - Circuito dañado por el agua, vista superior

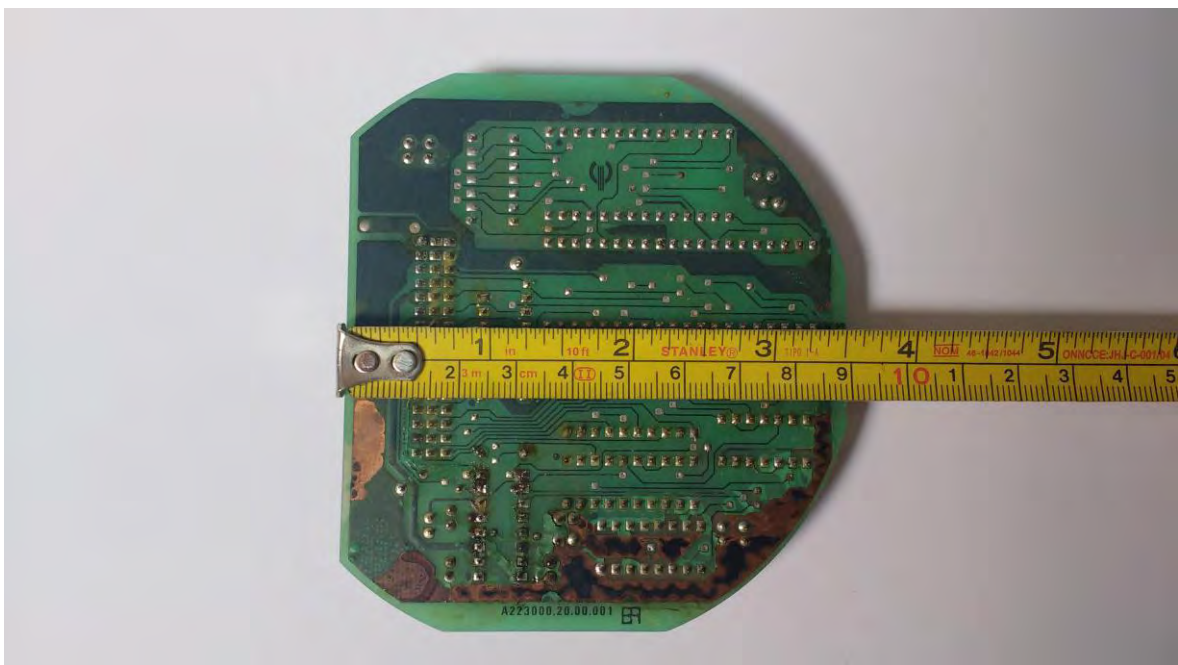
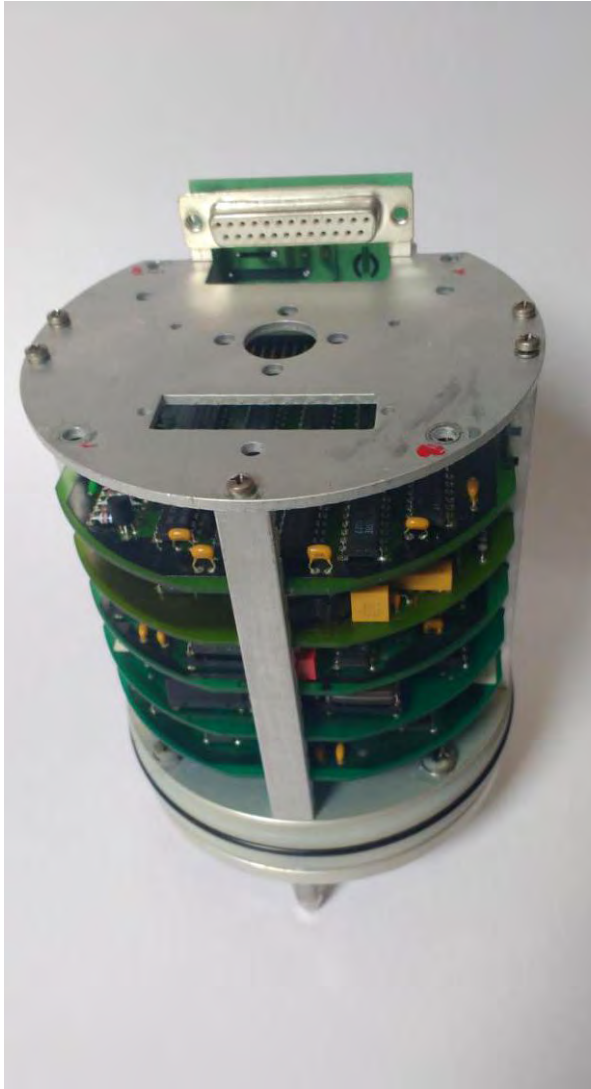


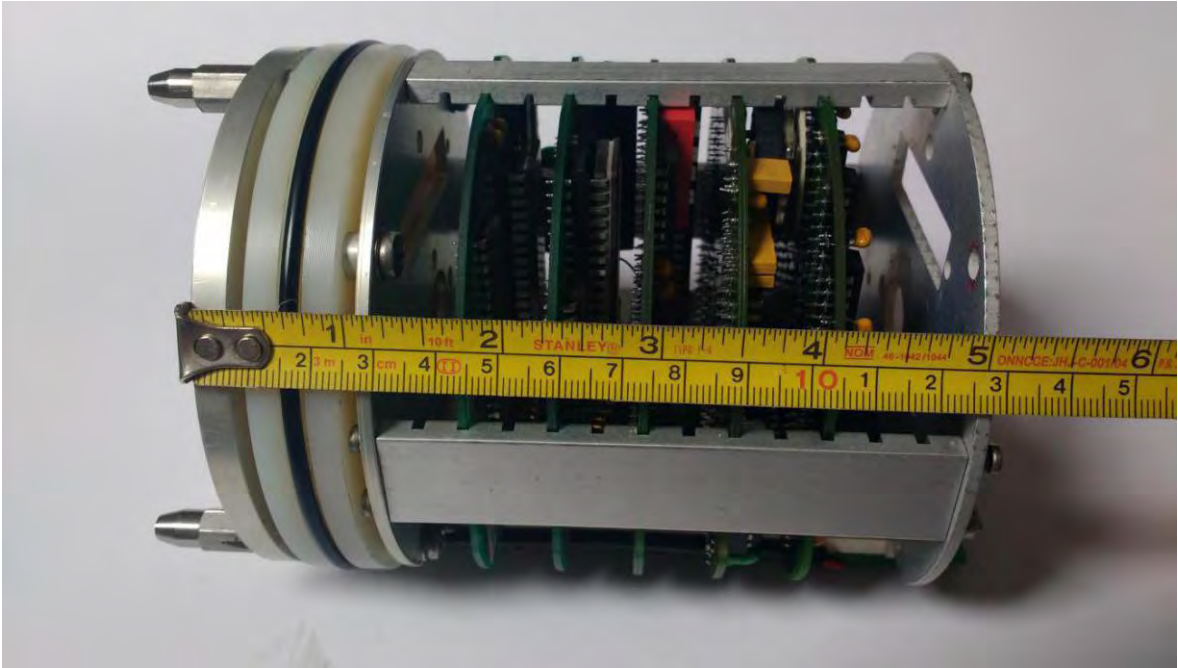
Imagen 2 - Circuito dañado por el agua, vista inferior.



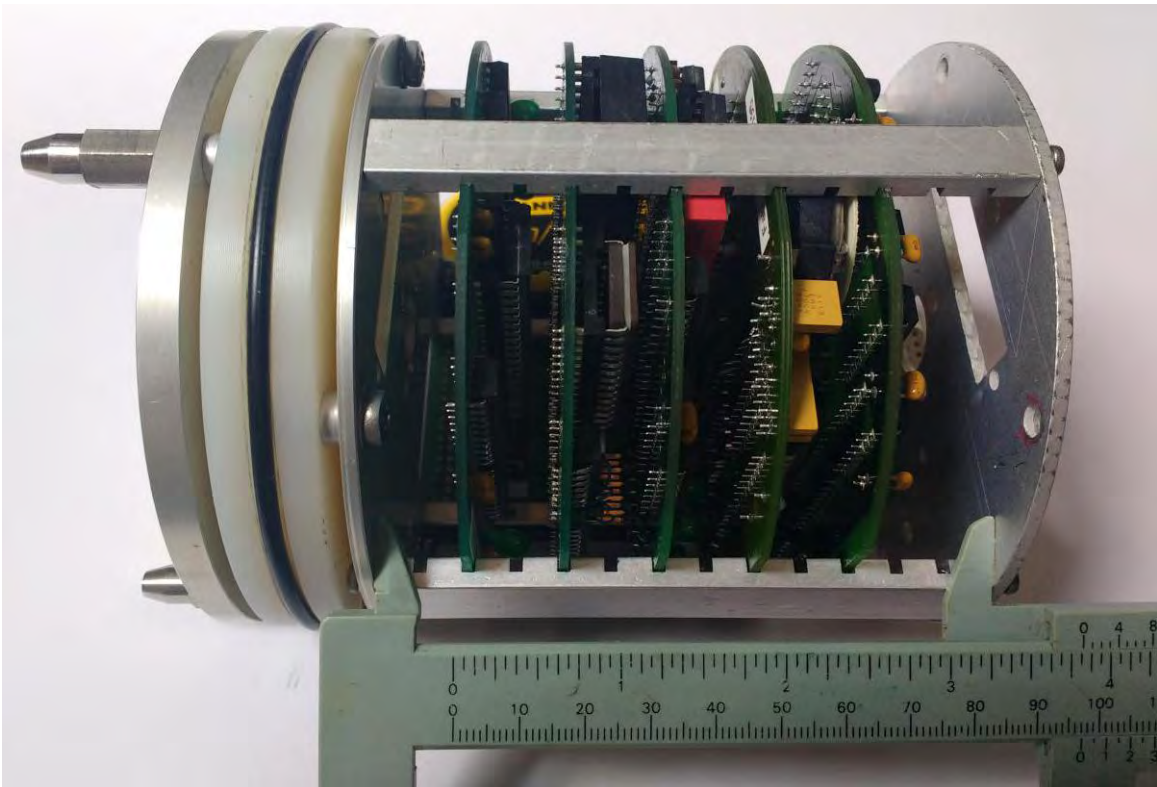
*Imagen 3 - Circuito de control original, vista posterior.*



*Imagen 4 - Circuito de control original, vista anterior.*



*Imagen 5 - Circuito de control original, vista lateral.*



*Imagen 6 - Circuito de control, vista lateral (2).*

## **ANEXO 2 INSTALACIÓN DEL PROGRAMA.**

Para la instalación del programa se requiere de una computadora con una versión de Windows XP o superior, ya sea con un sistema operativo de 32 o de 64 bits.

### **Requisitos de instalación:**

#### **Windows XP**

##### **Sistemas operativos compatibles:**

- ✓ Windows XP Service Pack 3.

##### **Software requerido:**

- ✓ .Net Framework 4.0 (incluido en el disco de instalación, o disponible desde: <https://www.microsoft.com/es-mx/download/details.aspx?id=17851> )

#### **Windows Vista y Windows 7**

##### **Sistemas operativos compatibles:**

- ✓ Windows Vista SP2 (x86 y x64)
- ✓ Windows 7 SP1 (x86 y x64)
- ✓ Windows Server 2008 R2 SP1 (x64)
- ✓ Windows Server 2008 SP2 (x86 y x64)

##### **Software requerido:**

.Net Framework 4.5 o superior (incluido en el disco de instalación, o disponible desde: <https://www.microsoft.com/es-mx/download/details.aspx?id=30653> ).

#### **Windows 8, 8.1 y Windows 10**

##### **Sistemas operativos compatibles:**

- ✓ Windows 8 (x86 y x64)
- ✓ Windows 8.1 (x86 y x64)
- ✓ Windows 10 (x86 y x64)

##### **Software requerido:**

.Net Framework 4.5 o superior (El cual se encuentra pre instalado en estas versiones de sistema operativo).

### **Requisitos de Hardware:**

- ✓ Procesador a 1 GHz o más rápido.
- ✓ 512 MB de RAM.
- ✓ 50MB de almacenamiento en disco duro.
- ✓ Puerto serie o adaptador USB Serial.

### **Instrucciones de instalación.**

#### **Windows XP**

Primero se debe instalar .NET Framework 4.0, no intentar instalar la versión de Framework 4.5 ya que ésta no es compatible con el sistema operativo de Windows XP.

#### **Windows Vista y Windows 7**

Se debe instalar la versión de Framework 4.5

## Windows 8, 8.1 y Windows 10

No hay que instalar ningún software adicional, solo ejecutar la instalación del programa.

Después de instalar el Framework (si es necesario), se regresa a la carpeta de instalación; dentro de la carpeta de instalación se encuentran cuatro carpetas:

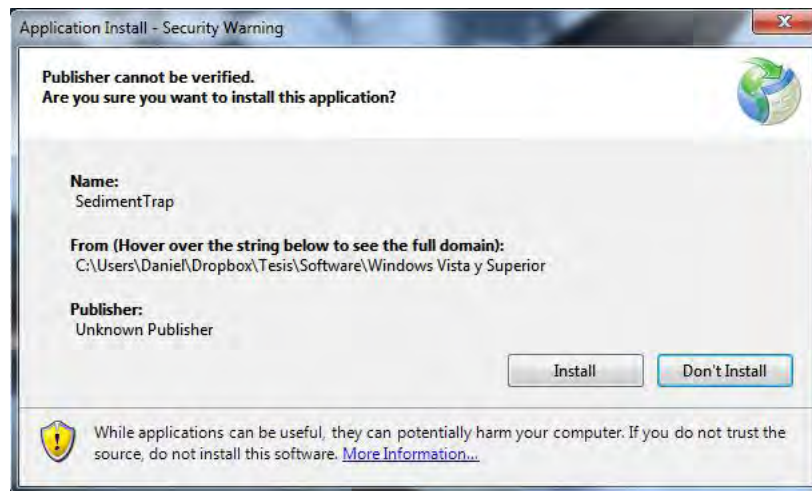
Name	Date modified	Type
.NET Frameworks	05/09/2016 07:40 ...	File folder
Driver USB-Serial	05/09/2016 07:38 ...	File folder
Windows Vista y Superior	05/09/2016 07:38 ...	File folder
Windows XP	05/09/2016 07:38 ...	File folder

Primero se elige la versión del software a instalar, en este caso la versión de **Windows Vista y Superior**.

Dentro de la carpeta se encuentran los siguientes archivos:

Name	Date modified	Type	Size
Application Files	05/09/2016 07:38 ...	File folder	
SedimentTrap	05/09/2016 07:38 ...	ClickOnce Applica...	6 KB
setup	05/09/2016 07:38 ...	Application	532 KB

Se ejecuta el archivo de **setup** que mostrará la siguiente pantalla:



Se presiona en instalar, y el software se encontrará instalado mostrando el menú principal, con lo que la instalación se encontrará finalizada.

Para ejecutar de nuevo el programa, simplemente hay que buscar el acceso directo al programa en el menú de inicio de Windows.



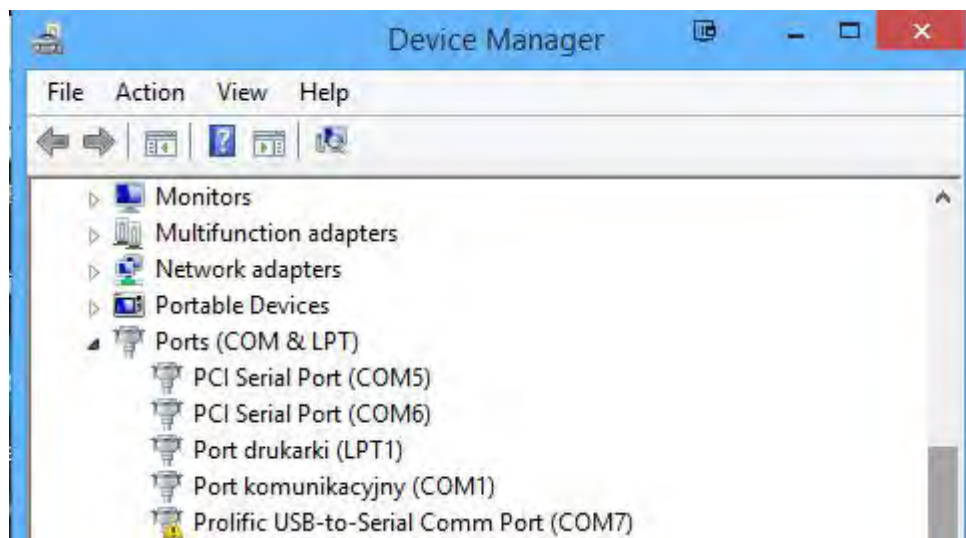
## ANEXO 3 PROBLEMAS EN INSTALACIÓN DE ADAPTADOR USB-SERIAL.

### Instalación de adaptador de puerto serie a USB.

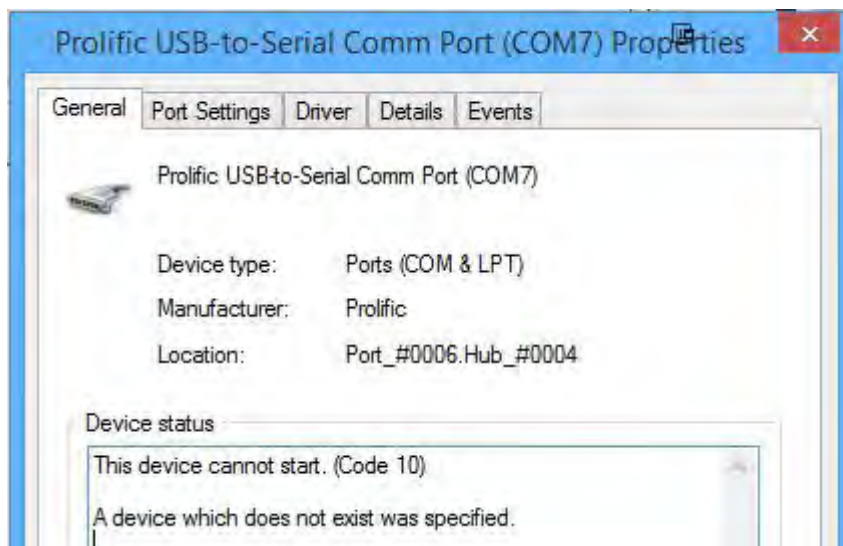
En caso de que la computadora no cuente con un puerto serie instalado, la opción más común es el uso de un adaptador de puerto USB a serie como el siguiente:



Desafortunadamente, la mayoría de cables de este tipo son fabricados por proveedores chinos, los cuales utilizan una copia no auténtica de estos circuitos; en este caso se utiliza una copia del circuito *PL-2303HX Rev A USB to Serial Controller*<sup>21</sup>, esto con la finalidad de reducir sus costos. A partir del año 2012; la compañía Prolific, fabricante del circuito original, comenzó a bloquear los controladores de este tipo de cables ocasionando un error en el administrador de dispositivos de Windows. El error es el código 10.



<sup>21</sup> Warning Letter, tomado de: [http://www.prolific.com.tw/US/ShowProduct.aspx?p\\_id=155&pcid=41](http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=155&pcid=41)



A pesar de que debería ser una obligación para el proveedor el distribuir estos convertidores con la licencia de funcionamiento correcta, no lo hace. Con la finalidad de evitar este problema, se opta por el uso de un **controlador previo a 2009**, en este caso el uso del **driver versión 3.3.2.105**<sup>22</sup>

Para ello se desinstala cualquier driver relacionado al adaptador; después se instala el paquete de drivers llamado:

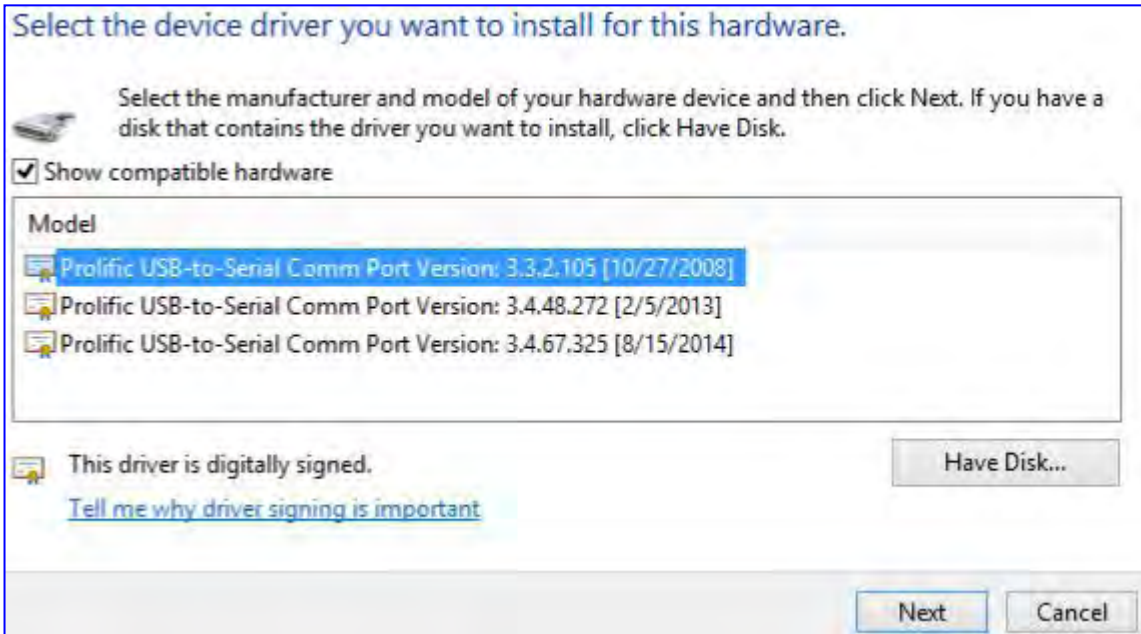
*IO-Cable PL-2303 Drivers-Generic Windows PL2303 Prolific*

Posteriormente hay que ir al Administrador de Dispositivos, clic derecho en **Prolific-to-Serial Comm Port** y seleccionar *Actualizar Driver*.

Una vez abierta esta ventana hay que elegir la opción *Buscar en mi computadora el driver* -> *Escoger de una lista de dispositivos en mi computadora*-> elegir de la lista de modelos la opción **Prolific USB-to\_Serial Comm Port version 3.3.2.105 model [10/27/2008]** y hacer clic en siguiente.

---

<sup>22</sup> Se puede descargar desde: [http://wp.brodzinski.net/wp-content/uploads/2014/10/IO-Cable\\_PL-2303\\_Drivers-Generic\\_Windows\\_PL2303\\_Prolific.zip](http://wp.brodzinski.net/wp-content/uploads/2014/10/IO-Cable_PL-2303_Drivers-Generic_Windows_PL2303_Prolific.zip)



Después de esto se puede utilizar sin problemas el adaptador USB-Serial.

## ANEXO 4 CÓDIGO FUENTE DE LA INTERFAZ GRÁFICA Y DEL MICROCONTROLADOR.

Debido a la extensión del código, el mismo se puede revisar desde la siguiente dirección: [https://github.com/escaleraalcielo/sediment\\_ctrl\\_234/](https://github.com/escaleraalcielo/sediment_ctrl_234/)

The screenshot shows the GitHub repository page for 'escaleraalcielo/sediment\_ctrl\_234'. The repository is private and has 12 commits, 2 branches, 0 releases, and 1 contributor. The latest commit is from 11 days ago. The repository contains several folders and files, including PCB Diseño, Programa Microcontrolador/Arduino, Programa Windows/SedimentTrapSoftware, Software Requerido/Driver USB-Serial, .gitattributes, .gitignore, and README.md. The README.md file is displayed below the repository list, showing the title 'sediment\_ctrl\_234' and the description 'Repositorio en Github con todo el contenido relacionado con el rediseño de la trampa de sediemntos S/MT 234'.

Repository in Github con todo el contenido relacionado con el rediseño de la trampa de sediemntos S/MT 234 — Edit

12 commits 2 branches 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

escaleraalcielo actualización al 9 de septiembre Latest commit e2094d4 11 days ago

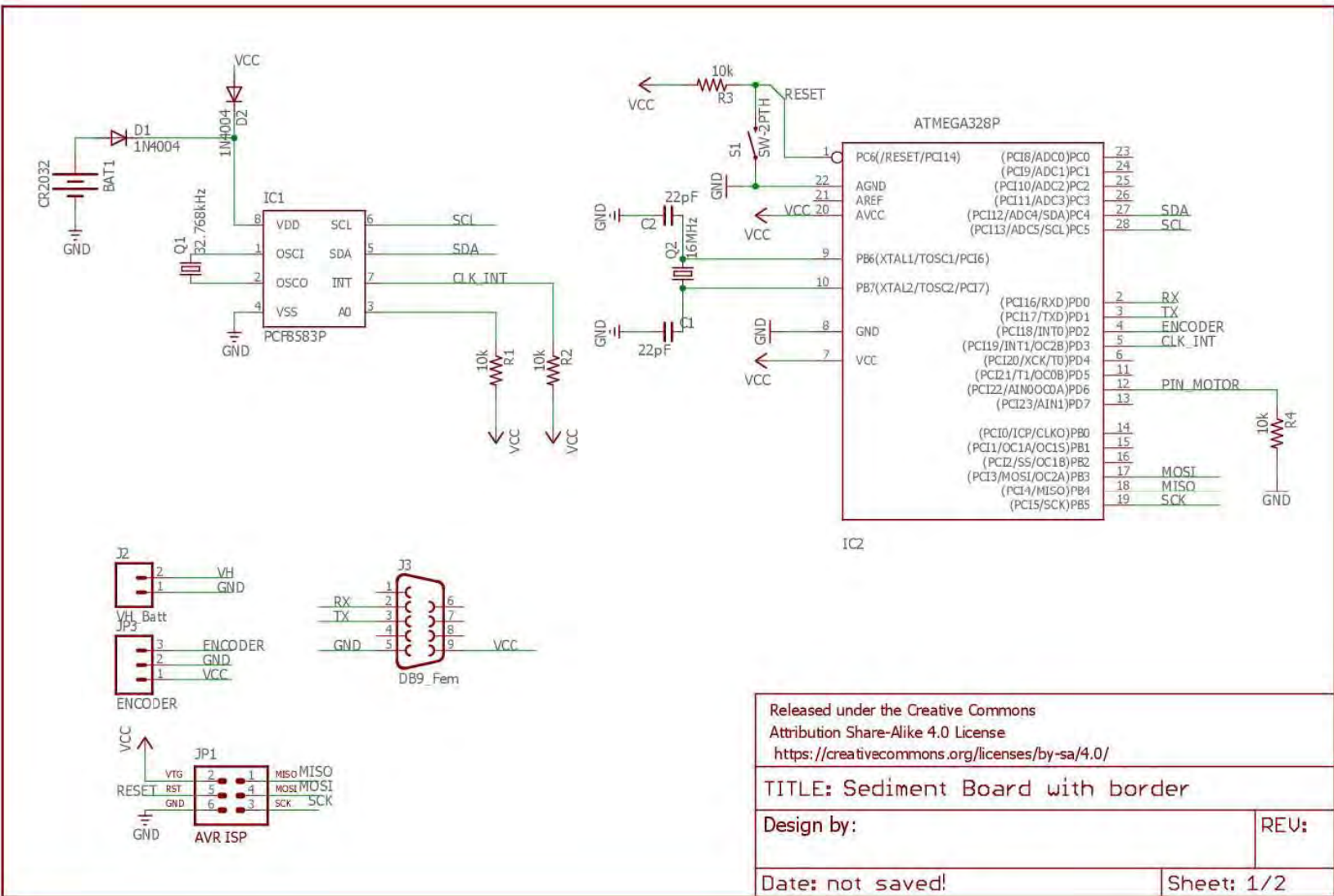
File/Folder	Commit Message	Time
PCB Diseño	actualización al 9 de septiembre	11 days ago
Programa Microcontrolador/Arduino	actualización al 9 de septiembre	11 days ago
Programa Windows/SedimentTrapSoftware	actualización al 9 de septiembre	11 days ago
Software Requerido/Driver USB-Serial	actualización al 9 de septiembre	11 days ago
.gitattributes	Added .gitattributes & .gitignore files	15 days ago
.gitignore	Added .gitattributes & .gitignore files	15 days ago
README.md	Create README.md	15 days ago

README.md

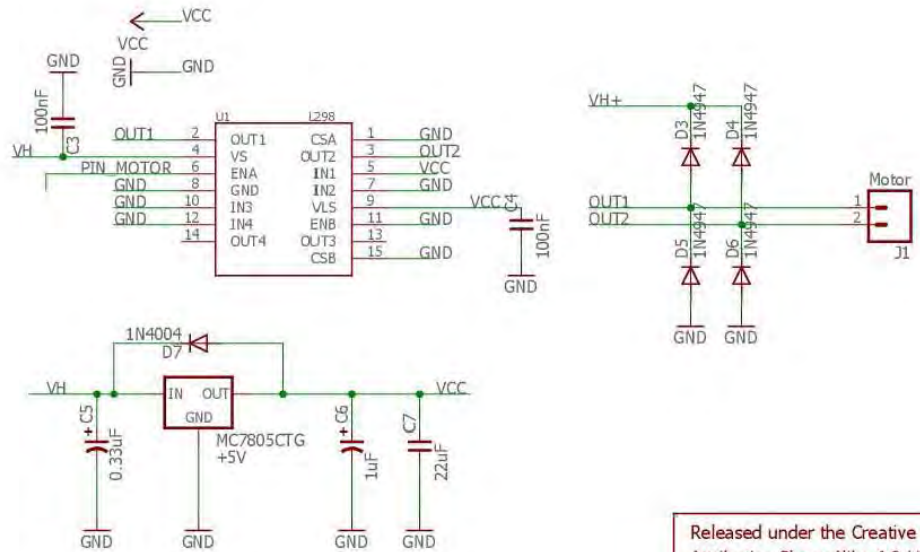
### sediment\_ctrl\_234

Repositorio en Github con todo el contenido relacionado con el rediseño de la trampa de sediemntos S/MT 234

# ANEXO 5 DISEÑO ESQUEMÁTICO DEL CIRCUITO



Released under the Creative Commons Attribution Share-Alike 4.0 License <a href="https://creativecommons.org/licenses/by-sa/4.0/">https://creativecommons.org/licenses/by-sa/4.0/</a>	
TITLE: Sediment Board with border	
Design by:	REV:
Date: not saved!	Sheet: 1/2



Released under the Creative Commons Attribution Share-Alike 4.0 License <a href="https://creativecommons.org/licenses/by-sa/4.0/">https://creativecommons.org/licenses/by-sa/4.0/</a>	
TITLE: Sediment Board with border	
Design by:	REV:
Date: not saved!	Sheet: 2/2

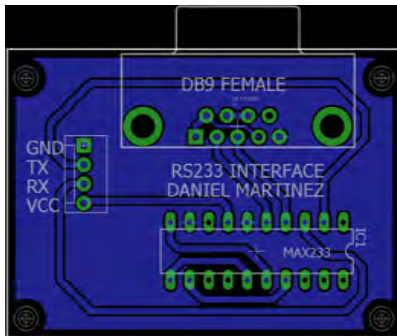
## ANEXO 6 DISEÑO DE PCB.

Para diseñar los diferentes circuitos de las PCB se utilizó el software de Cadsoft Eagle<sup>23</sup>, se generaron dos diseños; uno llamado **RS233**, el cual es un convertidor de TTL a RS232, y uno llamado **Sediment\_trap**, el cual es el circuito de la trampa de sedimentos. Los archivos generados fueron los siguientes:

### ✓ **RS233**

Dimensiones:

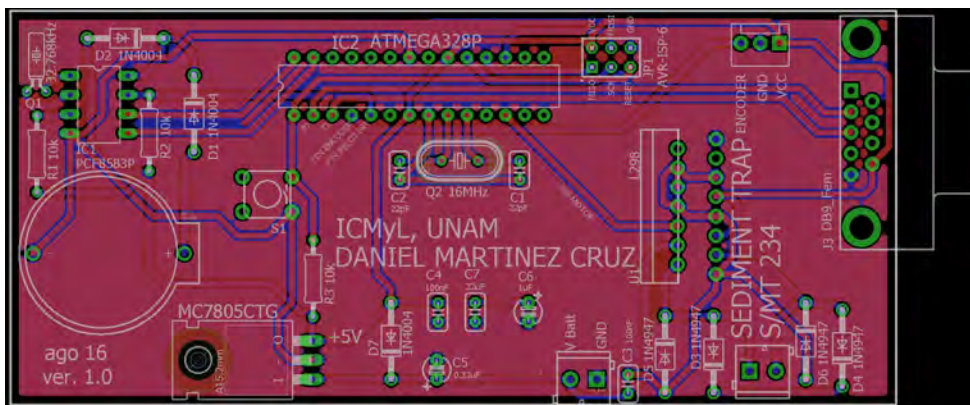
52mm x 38mm



### ✓ **Sediment\_trap**

Dimensiones:

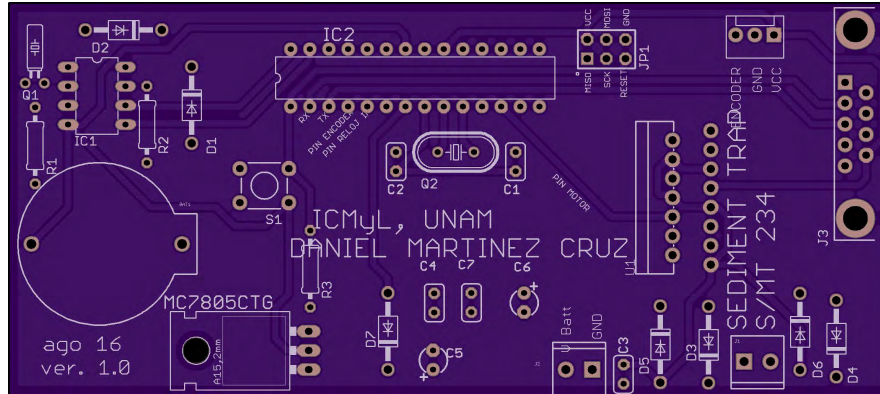
52mm x 117 mm



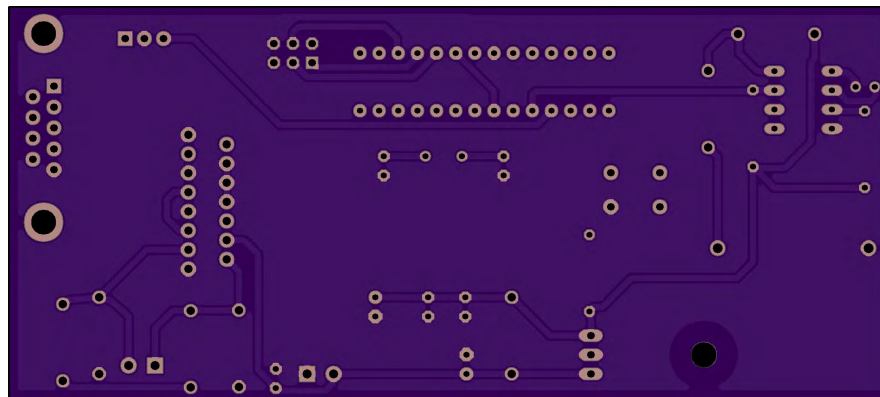
<sup>23</sup> <https://cadsoft.io/>

El diseño final de la tarjeta es el siguiente:

Vista superior, tamaño real.



Vista inferior, tamaño real.



## ANEXO 7 PROGRAMADOR DE AVR SPARKFUN.

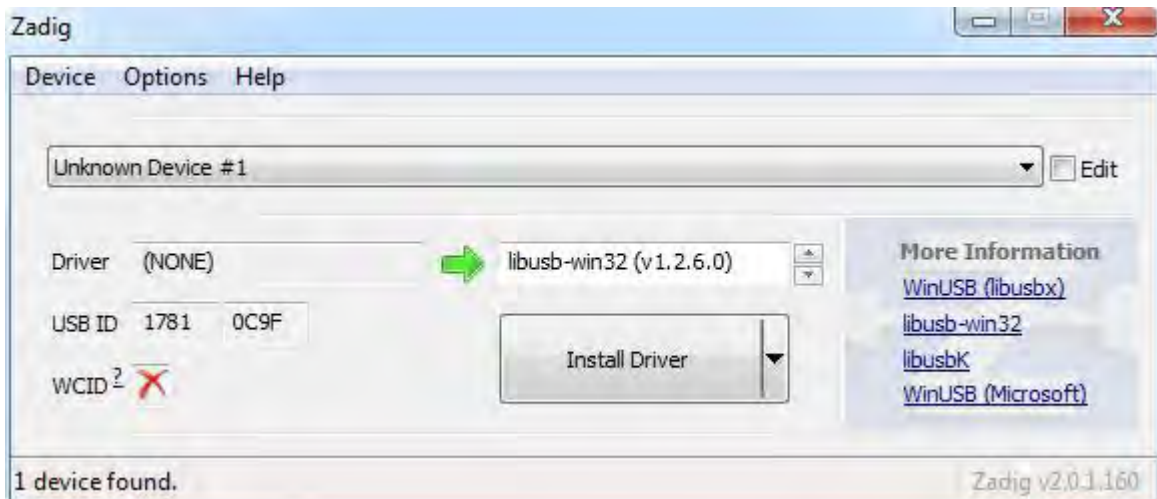
El programador nos permite subir nuestro código al microcontrolador por medio de la interface de programación *ISP (In System Programming Interface)*.

Para instalar el programador primero debemos de conectarlo a la computadora. Después de conectarlo, Windows tratará de buscar el driver sin éxito. Posterior a esto, se deben de instalar los drivers del programador llamados **Zadig USBtiny Drivers**<sup>24</sup>. Después de extraer el archivo, se ejecuta el archivo denominado **zadig.exe** el cual muestra la siguiente pantalla:

<sup>24</sup> Drivers disponibles desde:

[https://cdn.sparkfun.com/assets/learn\\_tutorials/2/1/4/zadig\\_v2.0.1.160.zip](https://cdn.sparkfun.com/assets/learn_tutorials/2/1/4/zadig_v2.0.1.160.zip)





Primero se debe seleccionar el dispositivo denominado *Unknown Device #1*. Posterior a eso, se debe seleccionar de la lista el driver denominado **libusb-win32 (vx.x.x.x)**. Después de elegir estas dos opciones se presiona en *Install Driver*, seguidamente, tras un par de minutos nos muestra en la pantalla un mensaje que dice que el driver fue instalado correctamente con lo que ya es posible utilizar el programador.

## ANEXO 8 DATASHEETS.



8-bit AVR Microcontrollers

ATmega328/P

DATASHEET SUMMARY

### Introduction

The Atmel<sup>®</sup> picoPower<sup>®</sup> ATmega328/P is a low-power CMOS 8-bit microcontroller based on the AVR<sup>®</sup> enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega328/P achieves throughputs close to 1MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

### Feature

High Performance, Low Power Atmel<sup>®</sup>AVR<sup>®</sup> 8-Bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 32KBytes of In-System Self-Programmable Flash program Memory
  - 1KBytes EEPROM
  - 2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data Retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel<sup>®</sup> QTouch<sup>®</sup> Library Support
  - Capacitive Touch Buttons, Sliders and Wheels
  - QTouch and QMatrix<sup>®</sup> Acquisition
  - Up to 64 sense channels

- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Two Master/Slave SPI Serial Interface
  - One Programmable Serial USART
  - One Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - One On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 105°C
- Speed Grade:
  - 0 - 4MHz @ 1.8 - 5.5V
  - 0 - 10MHz @ 2.7 - 5.5V
  - 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA
  - Power-down Mode: 0.1µA
  - Power-save Mode: 0.75µA (Including 32kHz RTC)

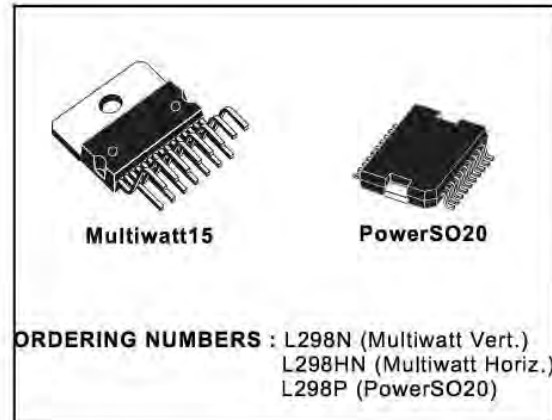


## DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

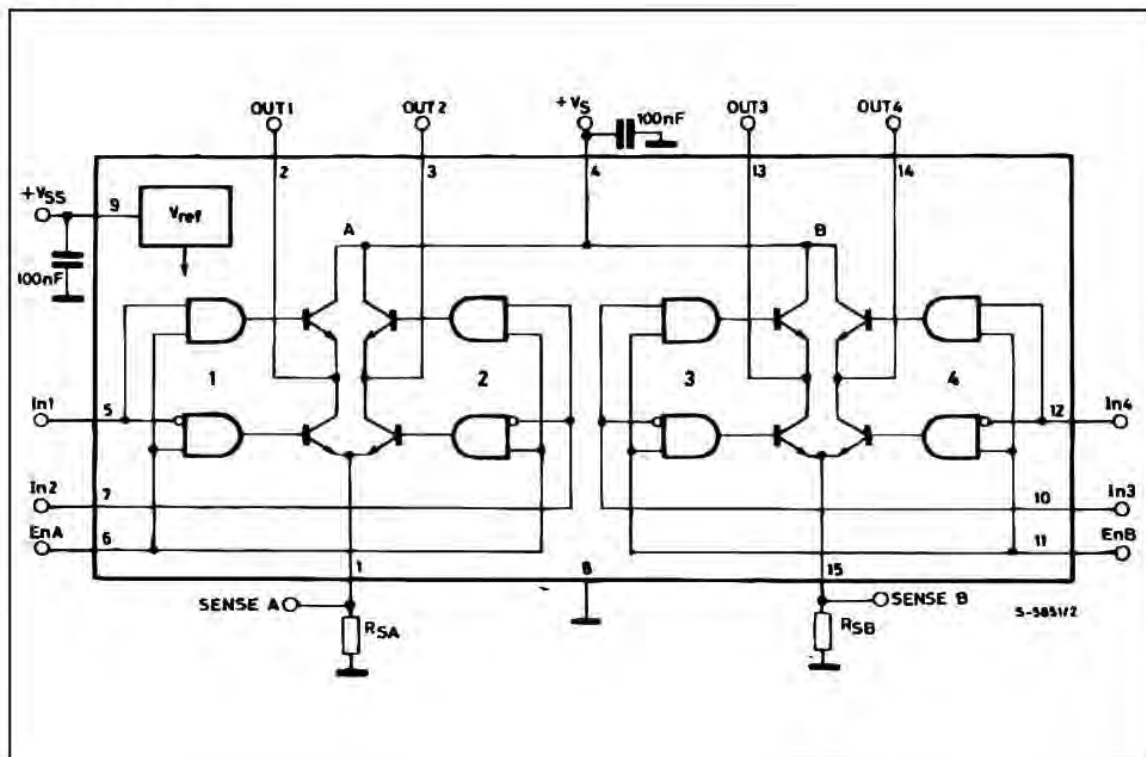
### DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

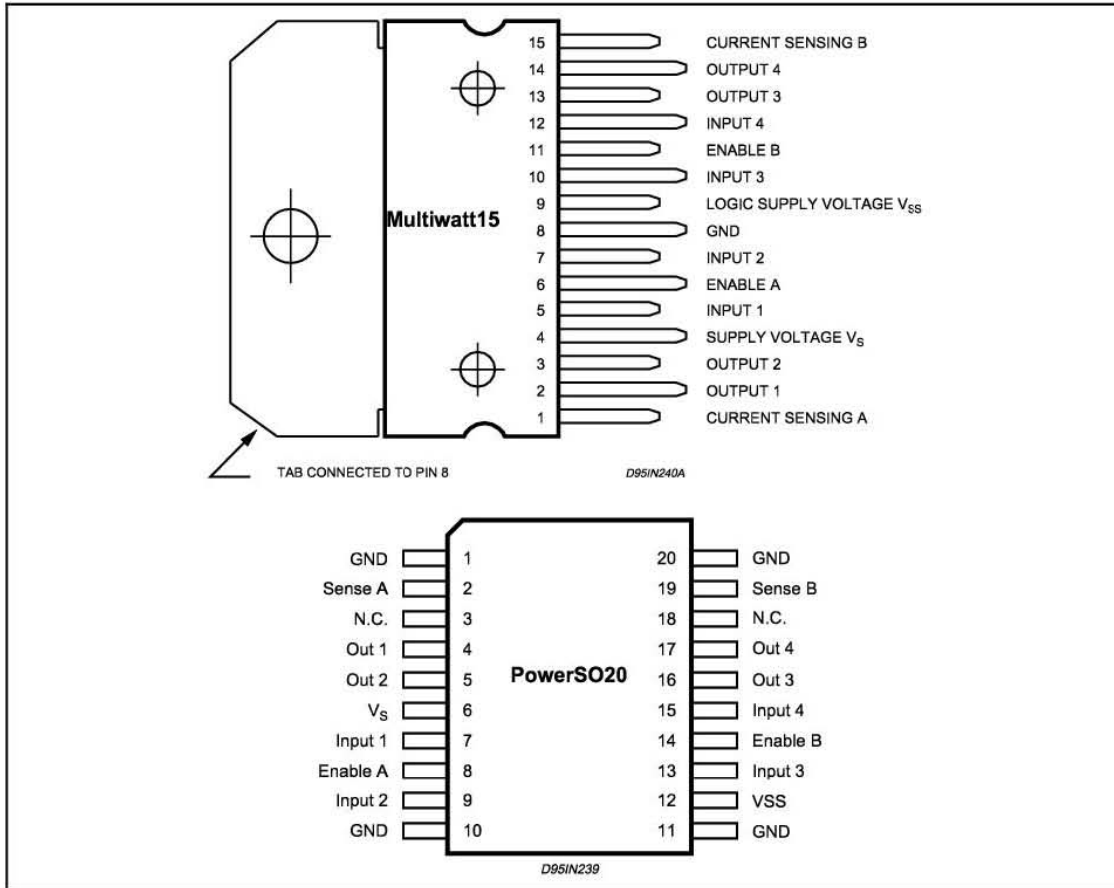
### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value	Unit
$V_S$	Power Supply	50	V
$V_{SS}$	Logic Supply Voltage	7	V
$V_i, V_{en}$	Input and Enable Voltage	-0.3 to 7	V
$I_o$	Peak Output Current (each Channel)		
	- Non Repetitive ( $t = 100\mu s$ )	3	A
	- Repetitive (80% on -20% off; $t_{on} = 10ms$ )	2.5	A
	-DC Operation	2	A
$V_{sens}$	Sensing Voltage	-1 to 2.3	V
$P_{tot}$	Total Power Dissipation ( $T_{case} = 75^\circ C$ )	25	W
$T_{op}$	Junction Operating Temperature	-25 to 130	$^\circ C$
$T_{stg}, T_j$	Storage and Junction Temperature	-40 to 150	$^\circ C$

**PIN CONNECTIONS (top view)**



**THERMAL DATA**

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max.	-	3	$^\circ C/W$
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	$^\circ C/W$

(\*) Mounted on aluminum substrate



**PIN FUNCTIONS** (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V <sub>S</sub>	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V <sub>SS</sub>	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

**ELECTRICAL CHARACTERISTICS** (V<sub>S</sub> = 42V; V<sub>SS</sub> = 5V, T<sub>J</sub> = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V <sub>S</sub>	Supply Voltage (pin 4)	Operative Condition	V <sub>IH</sub> +2.5		46	V
V <sub>SS</sub>	Logic Supply Voltage (pin 9)		4.5	5	7	V
I <sub>S</sub>	Quiescent Supply Current (pin 4)	V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L	13	22	mA
			V <sub>i</sub> = H	50	70	mA
		V <sub>en</sub> = L	V <sub>i</sub> = X		4	mA
I <sub>SS</sub>	Quiescent Current from V <sub>SS</sub> (pin 9)	V <sub>en</sub> = H; I <sub>L</sub> = 0	V <sub>i</sub> = L	24	36	mA
			V <sub>i</sub> = H	7	12	mA
		V <sub>en</sub> = L	V <sub>i</sub> = X		6	mA
V <sub>IL</sub>	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V <sub>IH</sub>	Input High Voltage (pins 5, 7, 10, 12)		2.3		V <sub>SS</sub>	V
I <sub>IL</sub>	Low Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = L			–10	μA
I <sub>IH</sub>	High Voltage Input Current (pins 5, 7, 10, 12)	V <sub>i</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>en</sub> = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V <sub>en</sub> = H	Enable High Voltage (pins 6, 11)		2.3		V <sub>SS</sub>	V
I <sub>en</sub> = L	Low Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = L			–10	μA
I <sub>en</sub> = H	High Voltage Enable Current (pins 6, 11)	V <sub>en</sub> = H ≤ V <sub>SS</sub> –0.6V		30	100	μA
V <sub>CEsat (H)</sub>	Source Saturation Voltage	I <sub>L</sub> = 1A I <sub>L</sub> = 2A	0.95	1.35 2	1.7 2.7	V V
V <sub>CEsat (L)</sub>	Sink Saturation Voltage	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V <sub>CEsat</sub>	Total Drop	I <sub>L</sub> = 1A (5) I <sub>L</sub> = 2A (5)	1.80		3.2 4.9	V V
V <sub>sens</sub>	Sensing Voltage (pins 1, 15)		–1 (1)		2	V



# PCF8583

Clock and calendar with 240 x 8-bit RAM

Rev. 06 — 6 October 2010

Product data sheet

## 1. General description

The PCF8583 is a clock and calendar chip, based on a 2048 bit static CMOS<sup>1</sup> RAM organized as 256 words by 8 bits. Addresses and data are transferred serially via the two-line bidirectional I<sup>2</sup>C-bus. The built-in word address register is incremented automatically after each written or read data byte. Address pin A0 is used for programming the hardware address, allowing the connection of two devices to the bus without additional hardware.

The built-in 32.768 kHz oscillator circuit and the first 8 bytes of the RAM are used for the clock, calendar, and counter functions. The next 8 bytes can be programmed as alarm registers or used as free RAM space. The remaining 240 bytes are free RAM locations.

## 2. Features and benefits

- I<sup>2</sup>C-bus interface operating supply voltage: 2.5 V to 6 V
- Clock operating supply voltage 1.0 V to 6.0 V at 0 °C to +70 °C
- 240 × 8-bit low-voltage RAM
- Data retention voltage: 1.0 V to 6.0 V
- Operating current (at f<sub>SCL</sub> = 0 Hz): max 50 μA
- Clock function with four year calendar
- Universal timer with alarm and overflow indication
- 24 hour or 12 hour format
- 32.768 kHz or 50 Hz time base
- Serial input and output bus (I<sup>2</sup>C-bus)
- Automatic word address incrementing
- Programmable alarm, timer, and interrupt function
- Slave addresses: A1h or A3h for reading, A0h or A2h for writing

1. The definition of the abbreviations and acronyms used in this data sheet can be found in [Section 14](#).





### 3. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
PCF8583P	DIP8	plastic dual in-line package; 8 leads (300 mil)	SOT97-1
PCF8583T	SO8	plastic small outline package; 8 leads; body width 7.5 mm	SOT176-1
PCF8583BS	HVQFN20	plastic thermal enhanced very thin quad flat package; no leads; 20 terminals; body 5 × 5 × 0.85 mm	SOT662-1

### 4. Marking

Table 2. Marking codes

Type number	Marking code
PCF8583P	PCF8583P
PCF8583T	8583T
PCF8583BS	8583S

### 5. Block diagram

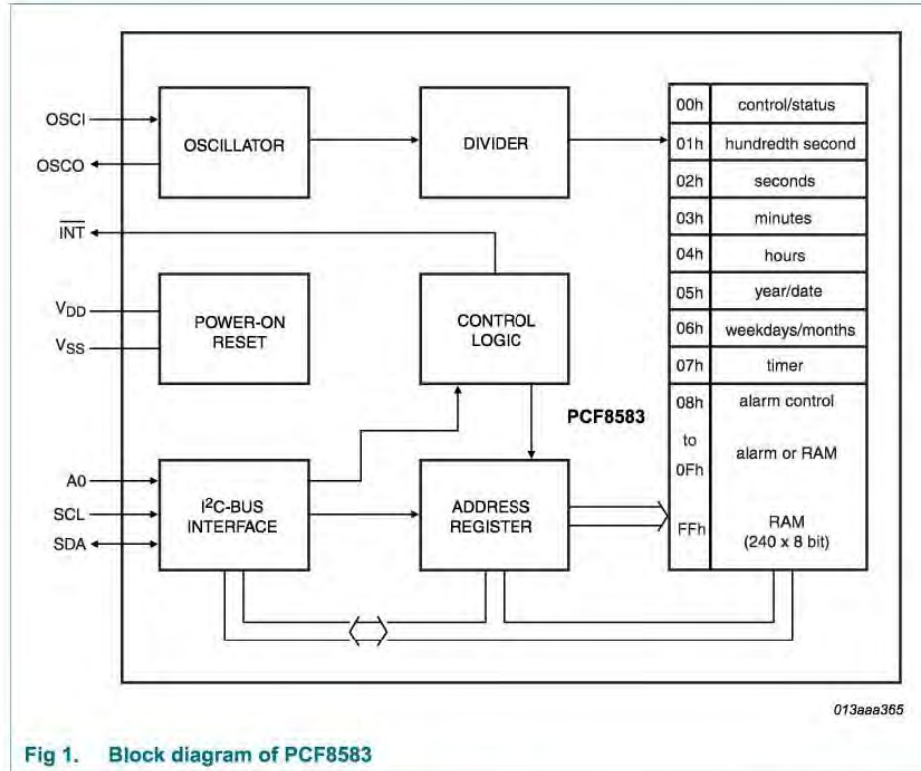
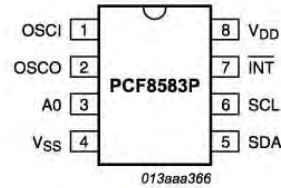


Fig 1. Block diagram of PCF8583

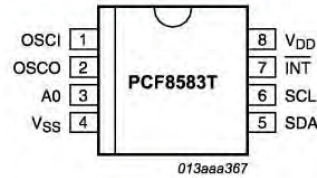
## 6. Pinning information

### 6.1 Pinning



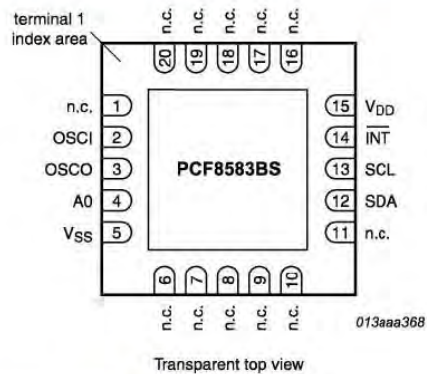
Top view. For mechanical details, see [Figure 24](#).

**Fig 2. Pin configuration for DIP8 (PCF8583P)**



Top view. For mechanical details, see [Figure 25](#).

**Fig 3. Pin configuration for SO8 (PCF8583T)**



For mechanical details, see [Figure 26](#).

**Fig 4. Pin configuration for HVQFN20 (PCF8583BS)**

this mode, the timer (location 07h) increments once for every one, one hundred, ten thousand, or 1 million events, depending on the value programmed in bits 0, 1 and 2 of the alarm control register. In all other events, the timer functions are as in the clock mode.

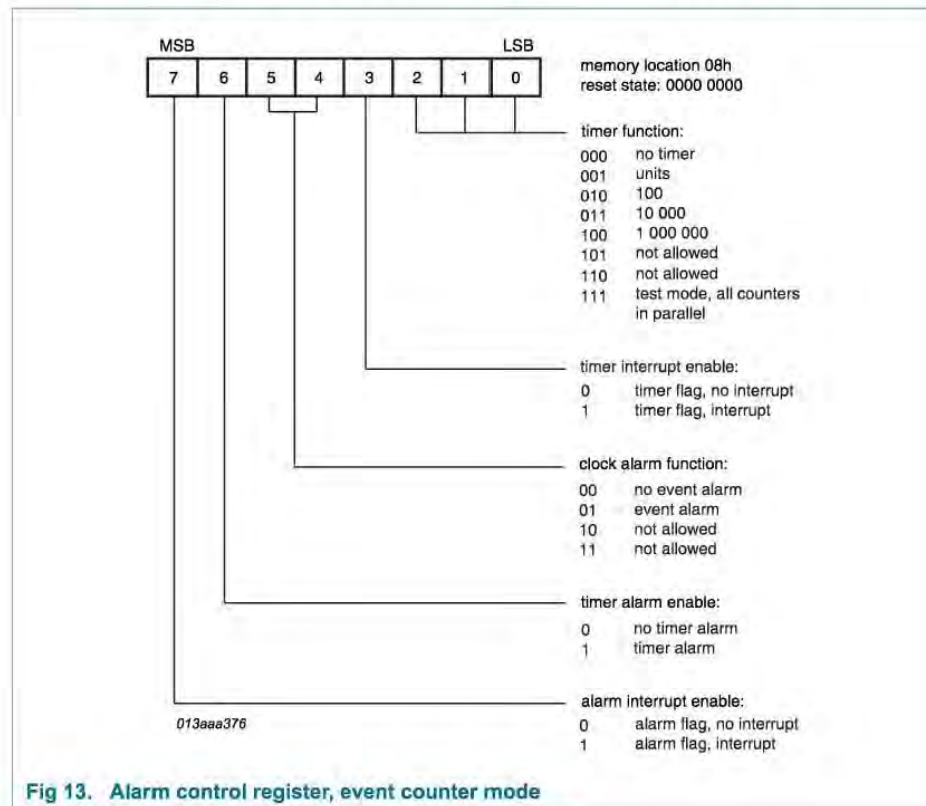


Fig 13. Alarm control register, event counter mode

### 7.9 Interrupt output

The conditions for activating the output  $\overline{\text{INT}}$  (active LOW) are determined by appropriate programming of the alarm control register. These conditions are clock alarm, timer alarm, timer overflow, and event counter alarm. An interrupt occurs when the alarm flag or the timer flag is set, and the corresponding interrupt is enabled. In all events, the interrupt is cleared only by software resetting of the flag which initiated the interrupt.

In the clock mode, if the alarm enable is not activated (alarm enable bit of the control and status register is logic 0), the interrupt output toggles at 1 Hz with a 50 % duty cycle (may be used for calibration). This is the default power-on state of the device. The OFF voltage of the interrupt output may exceed the supply voltage, up to a maximum of 6.0 V. A logic diagram of the interrupt output is shown in [Figure 12](#).

### 7.10 Oscillator and divider

A 32.768 kHz quartz crystal has to be connected to OSCI and OSCO. A trimmer capacitor between OSCI and  $V_{DD}$  is used for tuning the oscillator (see [Section 11.1](#)). A 100 Hz clock signal is derived from the quartz oscillator for the clock counters.