



# **UNIVERSIDAD TECNOLÓGICA ISRAEL**

## **TRABAJO DE TITULACIÓN**

**CARRERA:** ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

**TEMA:** Estudio, diseño e implementación de un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia.

**AUTOR:** Edwin Fernando Segura Martínez

**TUTOR:** PhD. René Alberto Cañete Bajuelo.

**AÑO 2015**

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“ESTUDIO, DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA ELECTRÓNICO DE ADVERTENCIA DE PRESENCIA DE AGUA A CIERTO NIVEL EN LOS PASOS DEPRIMIDOS DE LA CIUDAD DE QUITO, PARA REEMPLAZAR O COMPLEMENTAR LOS ACTUALES SISTEMAS DE ADVERTENCIA.”**, presentado por el Sr. Edwin Fernando Segura Martínez, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M. Marzo del 2015

TUTOR

.....

PhD. René Alberto Cañete Bajuelo.

# UNIVERSIDAD TECNOLÓGICA ISRAEL

## AUTORÍA DEL TRABAJO DE TITULACIÓN

El abajo firmante, en calidad de estudiante de la Carrera de Electrónica Digital y Telecomunicaciones, declaro que los contenidos de este Trabajo de Titulación, requisito previo a la obtención del Grado de Ingeniería en Electrónica Digital y Telecomunicaciones, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito D.M. Marzo del 2015

.....

Edwin Fernando Segura Martínez

CC: 1713702964

**UNIVERSIDAD TECNOLÓGICA ISRAEL**

**APROBACIÓN DEL TRIBUNAL DE GRADO**

Los miembros del Tribunal de grado, aprueban el trabajo de titulación para la graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Israel para títulos de pregrado.

Quito D.M. Marzo del 2015

Para constancia firma:

TRIBUNAL DE GRADO

.....  
PRESIDENTE

.....  
MIEMBRO 1

.....  
MIEMBRO 2

## **AGRADECIMIENTO**

Agradezco a Dios quien me ha guiado en las diferentes etapas de mi vida dándome las fuerzas y la sabiduría necesarias para levantarme cuando he caído, a mi esposa Mary, compañera y amiga que ha estado siempre brindándome el apoyo incondicional para que pueda llevar a cabo con satisfacción la culminación de mi carrera, a mi familia y a la familia de mi esposa quienes brindan su apoyo para cumplir las metas y objetivos planteados, a los profesores de la Universidad Tecnológica Israel quienes han impartido sus conocimientos y experiencia.

## **DEDICATORIA**

Dedico a Dios porque sin él nada es posible, a mí esposa Mary quien ha comprendido mi ausencia a causa del tiempo dedicado a los estudios, especialmente a mi hija Arianna quien es la niña de mis ojos y mi inspiración para esforzarme cada días más.

## ÍNDICE

APROBACIÓN DEL TUTOR .....	II
AUTORÍA DEL TRABAJO DE TITULACIÓN .....	III
APROBACIÓN DEL TRIBUNAL DE GRADO .....	IV
AGRADECIMIENTO.....	V
DEDICATORIA.....	VI
ÍNDICE DE FIGURAS .....	X
ÍNDICE DE TABLAS .....	XI
INTRODUCCIÓN .....	1
Descripción.....	1
Antecedentes.....	1
Clima.....	2
Intercambiadores de tráfico y pasos deprimidos o a desnivel .....	3
Problema de Investigación .....	3
OBJETIVOS .....	4
Objetivo General.....	4
Objetivos Específicos:.....	4
CAPÍTULO I .....	6
1.    FUNDAMENTACIÓN TEÓRICA.....	6
1.1.    Introducción .....	6
1.2.    Marco Teórico.....	6
1.2.1.    Microcontroladores.....	6
1.2.2.    Programación.....	8
1.2.3.    Arduino .....	8
1.2.3.1.    Cuerpo de un programa en Arduino.....	9
1.2.4.    Microcontroladores Atmel (AVR 8) .....	9
1.2.5.    Reloj en Tiempo Real DS1307 .....	10
1.2.6.    Protocolo I2C .....	10
1.2.6.1.    Características del protocolo: .....	10
1.2.7.    Sensor de Temperatura DS18B20 .....	11
1.2.8.    Modem Celular WAVECOM M1306B .....	11
1.2.9.    Comunicación Serial RS232.....	13
1.2.10.    Circuito de Interface serial MAX232 .....	13
1.2.11.    Motores Paso a Paso .....	14
1.2.12.    Display LCD 20 x 4 .....	14
1.2.13.    LED IR .....	14
1.2.14.    Fototransistor .....	15

1.2.15.	Switch de Nivel.....	15
1.2.16.	OrCAD .....	15
1.2.17.	Proteus .....	15
1.2.18.	Software Arduino.....	16
1.2.19.	Software BASCOM – AVR .....	16
1.2.20.	Medición de Nivel.....	17
1.2.20.1.	Medidores de nivel de líquidos .....	17
1.2.20.2.	Sensores de nivel .....	17
CAPÍTULO II .....		19
2.	BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO PARA LA IMPLEMENTACIÓN DE UN SISTEMA ELECTRONICO DE SEGURIDAD Y ADVERTENCIA PARA LOS PASOS A DESNIVEL.....	19
2.1.	Problema Principal .....	19
2.2.	Por Qué y Para Qué de los Objetivos.....	19
2.3.	Hipótesis o Idea a Defender.....	19
2.3.1.	Variables .....	20
2.3.1.1.	Independientes:.....	20
2.3.1.2.	Dependientes:.....	20
2.4.	Los métodos que se utilizaron para el desarrollo del proyecto son:.....	20
2.4.1.	Análisis y Síntesis: .....	20
2.4.2.	Modelación.....	20
2.4.3.	Experimental .....	20
2.5.	Resultados esperados del Proyecto .....	21
CAPÍTULO III .....		22
3.	PRESENTACIÓN DE RESULTADOS .....	22
3.1.	Requisitos del Proyecto .....	22
3.1.1.	Requisitos Generales .....	22
3.1.2.	Requisitos Específicos.....	22
3.2.	Diagrama de Bloques General .....	22
3.3.	Diagrama de Bloques del Hardware.....	23
3.4.	Diagrama Esquemático del Sistema.....	24
3.5.	Flujograma del Sistema.....	25
3.6.	Simulaciones (ISIS - Proteus) .....	26
3.6.1.	Simulación Switch de Nivel con Control Arduino .....	26
3.6.2.	Simulación Control de motor PaP .....	27
3.6.3.	Simulación Sensor de Temperatura DS18B20 y RTC DS1307 .....	28
3.6.4.	Simulación Emisor Infrarrojo (IR) .....	28



3.7.	Diseño Esquemático de Tarjetas.....	29
3.7.1.	Tarjeta Etapa Arduino – AVR 8 – Modem Celular .....	29
3.7.2.	Tarjeta Etapa Control de motor PaP – Sensores IR - Alarma .....	30
3.8.	Circuito Impreso de Tarjetas (PCB).....	31
3.8.1.	PCB Etapa Arduino – AVR 8 – Modem Celular .....	31
3.8.2.	PCB Etapa Control de motor PaP – Sensores IR - Alarma.....	32
3.8.3.	PCB Emisor IF .....	33
3.8.4.	PCB Receptor IF .....	33
3.9.	Implementación.....	34
3.9.1.	Procedimiento para la elaboración de las PCB .....	34
3.10.	Interconexión de Tarjetas e Instalación en Maqueta .....	39
3.11.	LEDS Indicadores del Estatus del Sistema .....	40
3.12.	Prototipo Final.....	41
3.12.1.	Descripción de partes:.....	42
3.13.	Evaluación Técnica del Sistema.....	43
3.14.	Pruebas de Funcionamiento.....	43
3.15.	Análisis de Resultados .....	44
3.15.1.	Análisis de Resultados Tabla 3.2 .....	44
3.15.2.	Análisis de Resultados Tabla 3.3 .....	45
3.15.3.	Análisis de Resultados Tabla 3.4 .....	45
4.	CONCLUSIONES Y RECOMENDACIONES.....	46
4.1.	Conclusiones .....	46
4.2.	Recomendaciones .....	46
1.	BIBLIOGRAFÍA .....	48
ANEXOS	.....	1

## ÍNDICE DE FIGURAS

Figura 0.1 Parámetros climáticos promedio de Quito .....	2
Figura 1.1 Microcontrolador.....	6
Figura 1.2 Diagrama de Bloques del Microcontrolador .....	7
Figura 1.3 Pines de un Microcontrolador PDIP .....	8
Figura 1.4 Programación .....	8
Figura 1.5 Logotipo de Arduino .....	9
Figura 1.6 Circuito típico del RTC.....	10
Figura 1.7 Circuito típico Sensor de Temperatura .....	11
Figura 1.8 Arquitectura funcional del Modem .....	12
Figura 1.9 Fastrack M1306B descripción General.....	13
Figura 1.10 Modelo conceptual de un Motor Paso a Paso.....	14
Figura 1.11 Espectro de Luz Visible .....	14
Figura 1.12 Switch de Nivel.....	15
Figura 1.13 Logotipo de OrCAD (Diseño de PCB).....	15
Figura 1.14 Logotipo Proteus (Simulación).....	16
Figura 1.15 Logotipo Software Arduino .....	16
Figura 1.16 Información de la versión Instalada .....	17
Figura 1.17 Ejemplo de medidores de nivel por flotador .....	18
Figura 3.2.1 Diagrama de Bloques General.....	23
Figura 3.3.1 Diagrama de Bloques Hardware.....	23
Figura 3.4.1 Diagrama Esquemático del Sistema.....	24
Figura 3.5.1 diagrama de flujo del programa .....	25
Figura 3.6.1.1 Simulación Switch de nivel con control Arduino .....	26
Figura 3.6.2.1 Simulación control de motor PaP.....	27
Figura 3.6.3.1 Simulación DS18B20 y DS1307 .....	28
Figura 3.6.4.1 Simulación Emisor Infrarrojo.....	28
Figura 3.7.1.1 Diseño Tarjeta Arduino – AVR 8 – Modem Celular .....	29
Figura 3.7.2.1 Esquemático Tarjeta de control motor PaP – Sensor IR – Alarma.....	30
Figura 3.8.1.1 Distribución de elementos PCB etapa Arduino .....	31
Figura 3.8.1.2 Pistas PCB Etapa Arduino.....	31
Figura 3.8.2.1 Distribución de elementos PCB etapa control PaP .....	32
Figura 3.8.2.2 Pistas PCB etapa control PaP .....	32
Figura 3.8.3.1 Distribución de elementos PCB Emisor IR.....	33
Figura 3.8.3.2 Pistas PCB Emisor IR.....	33
Figura 3.8.4.1 Distribución de elementos PCB Receptor IR .....	33

Figura 3.8.4.2 Pistas PCB Receptor IP.....	34
Figura 3.9.1.1 Impresión de Pistas en papel térmico .....	34
Figura 3.9.1.2 Corte de Baquelita.....	35
Figura 3.9.1.3 Termografía de pistas en baquelita.....	35
Figura 3.9.1.4 Ejemplo de corrección de fallas .....	36
Figura 3.9.1.5 Remoción de cobre excedente con ácido .....	36
Figura 3.9.1.6 PCB terminada .....	37
Figura 3.9.1.7 Perforación en placa y distribución de elementos .....	37
Figura 3.9.1.8 Vista posterior de la tarjeta Terminada .....	38
Figura 3.9.1.9 Vista Frontal Tarjeta Terminada .....	38
Figura 3.10.1 Interconexión Tarjeta de Control Arduino.....	39
Figura 3.10.2 Interconexión Tarjeta Control PaP .....	39
Figura 3.10.3 Interconexión Tarjetas Emisor – Receptor IR .....	40
Figura 3.10.4 Interconexión Modem Celular .....	40
Figura 3.11.1 Indicadores de estatus del sistema.....	41
Figura 3.11.2 Indicadores de estatus de la barrera.....	41
Figura 3.12.1 Producto Terminado .....	42

## ÍNDICE DE TABLAS

Tabla 3.1 Parámetros de Evaluación Técnica .....	43
Tabla 3.2 Pruebas de Funcionamiento en modo Automático.....	44
Tabla 3.3 Pruebas de Funcionamiento en modo Remoto.....	44
Tabla 3.4 Pruebas de Funcionamiento en modo Manual.....	44

## INTRODUCCIÓN

Con la finalidad de mejorar el sistema de señalización y seguridad de los pasos a desnivel o pasos deprimidos del Distrito Metropolitano de Quito ante una inundación, choque o mantenimiento, se realiza un análisis del sistema de señalética actual considerando su funcionalidad, ventajas y limitaciones de su operatividad.

### **Descripción**

Se plantea implementar un prototipo de sistema electrónico de señalización y advertencia automático que mejorará la seguridad para los usuarios de los pasos a desnivel o deprimidos.

El circuito tendrá la capacidad de detectar 5 niveles de agua dentro del paso a desnivel, cada uno de estos niveles serán mostrados en la pantalla (Display) indicando al usuario el nivel (altura) de agua en centímetros para que el usuario decida si su vehículo puede o no circular a determinada altura de agua dentro del paso a desnivel, en el último nivel considerado como crítico o imposible para la circulación se activará una barrera que impedirá el paso de cualquier vehículo al interior del paso a desnivel.

Considerando que este sistema únicamente funcionaría o se activaría cuando el paso a desnivel se inunde, es decir cuando se produzcan torrenciales lluvias, se incrementaron ciertas funcionalidades al circuito para que sea útil mientras está en modo de espera por así decirlo, una de sus funcionalidades es que en la pantalla se estaría mostrando todo el tiempo la hora y fecha actualizada, la temperatura ambiente y mensajes pregrabados de advertencia o consejos para manejar con precaución.

Eventualmente se puede producir la necesidad de bloquear el acceso al paso a desnivel ya sea por un accidente vehicular o por mantenimientos programados, para estos casos la barrera de acceso al puente puede ser accionada manualmente o remotamente mediante un mensaje enviado al modem GSM el cual le indicará al microcontrolador que ejecute una instrucción para bajar la barrera y así impedir el acceso al paso a desnivel.

### **Antecedentes**

Quito, capital del Ecuador cuenta con 2'240.000 habitantes en el Distrito Metropolitano, de las ciudades del Ecuador está en segundo lugar en número de habitantes. Es también cabecera cantonal del Distrito Metropolitano de Quito. Debido a su alto índice de desarrollo

humano, Quito será la ciudad más poblada del país en el 2020. (VisitaQuito, 2014)

En Quito se concentra la política del Ecuador, en esta ciudad se asientan los principales organismos del gobierno, de cultura, de finanzas al ser la sede de la mayoría de bancos de la Nación, administrativos y comerciales del país un alto porcentaje de empresas nacionales y transnacionales tienen su matriz en la ciudad de Quito. Quito, de acuerdo al censo 2010 es la capital económica del Ecuador a más de ser la capital administrativa. Quito posee uno de los centros históricos mejor conservados de Latino América y casi sin alteraciones respecto a su construcción original, lo que hizo que la Unesco declare a Quito como Patrimonio Cultural de la Humanidad, el 18 de septiembre de 1978.

Quito es la sede de la Unión de Naciones Suramericanas (UNASUR) desde el 2008, convirtiéndose en el centro de reuniones oficiales de los países de América del Sur.

## Clima

Quito posee un clima subtropical; el sur es la zona con más frío por ser una zona alta, el norte tiene un clima templado y el centro es caliente. La ciudad tiene dos etapas o estaciones, cuatro meses es una estación seca con las temperaturas más altas y casi ocho meses existe una estación o etapa de invierno con lluvias constantes. La temperatura de Quito está entre 10 y 27 °C haciéndole una ciudad de clima templado.

Parámetros climáticos promedio de Quito													
Mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Anual
Temperatura máxima absoluta (°C)	29	26	32	25	30.4	29	31	25.3	29	25.2	29	29	32
Temperatura máxima media (°C)	18.9	18.9	18.8	19.1	19.2	19.4	19.7	20.2	20.3	19.8	19.3	19.1	19.4
Temperatura media (°C)	14.4	14.5	14.5	14.7	14.6	14.5	14.4	14.8	14.8	14.7	14.4	14.4	14.6
Temperatura mínima media (°C)	9.9	10.1	10.2	10.2	10	9.6	9.1	9.3	9.3	9.5	9.6	9.7	9.7
Temperatura mínima absoluta (°C)	1	0	0	0	0.7	0	0.6	0.6	1.1	1	0	1.1	0
Lluvias (mm)	59.0	60.8	82.7	58.2	52.4	16.4	10.5	15.4	49.8	60.8	60.2	47.2	573.4
Días de lluvias (≥ 1 mm)	1.9	2.2	2.7	1.9	1.7	0.5	0.3	0.5	1.7	2.0	2.0	1.5	18.9
Días de nevadas (≥ 1 cm)	0	0	0	0	0	0	0	0	0	0	0	0	0

Fuente: Meowweather<sup>53</sup>

Figura 0.1 Parámetros climáticos promedio de Quito

Fuente: Meowweather

## **Intercambiadores de tráfico y pasos deprimidos o a desnivel**

El distrito Metropolitano de Quito cuenta con varias alternativas para facilitar el tránsito, estas son los intercambiadores de tráfico y los pasos deprimidos o a desnivel, uno de los intercambiadores más importantes es El Trébol que une el valle de los Chillos con el centro y sur de Quito. En el 2008 la naturaleza se encargó de destruir una de las "hojas" de El Trébol ante un desbordamiento del río Machángara desde entonces pasó a ser un redondel,

A lo largo de la Avenida 10 de Agosto se encuentran varios pasos deprimidos o a desnivel, algunos de ellos son: El paso a desnivel o deprimido de la Y en el que se conectan las principales avenidas, Galo Plaza Lazo, Av. América, La Prensa, y como avenidas secundarias la Gaspar de Villarroel y la Av. Brasil, al ser un paso a desnivel de alto tráfico es crítico si este falla o colapsa ya que se generaría un caos vehicular, otro paso a deprimido está en las Avenidas 10 de Agosto y NNUU, también se encuentra el de la Avenida Atahualpa y Avenida 10 de Agosto, el de la Avenida Eloy Alfaro y Av. 10 de Agosto. Con la construcción del nuevo aeropuerto en Tababela se construyeron a la par varios pasos deprimidos en los cuales se está utilizando el mismo sistema de señalización para advertir la presencia de agua a un nivel crítico para los usuarios.

Para prevenirlas, la Empresa Metropolitana de Agua Potable (EMAP) limpió en este año 70.000 de los casi 100.000 sumideros que tiene Quito. Las alcantarillas que están en los pasos deprimidos y los sectores aledaños a los mercados son los más propensos a taparse. (ANDES Cotopaxinoticias.com, 2012)

Las inundaciones que suelen ocurrir en la capital, tienen que ver con abundantes precipitaciones en períodos cortos de tiempo y que pueden presentarse como chubascos, tormentas eléctricas y granizadas. Los lugares más propensos a estas circunstancias son los intercambiadores que existen en la capital. Las alcantarillas que están en los pasos deprimidos y los sectores aledaños son los más propensos a taparse, provocando que estos pasos colapsen por las inundaciones, lo que ha generado varios incidentes de vehículos del servicio público como el trole bus y a vehículos privados. El Distrito Metropolitano de Quito con el objetivo de advertir cuando el nivel de agua en un paso deprimido está muy alto, instaló señalética como advertencia para los conductores indicándoles que no ingresen cuando existe cierto nivel de agua en el paso deprimido o a desnivel. (ANDES Cotopaxinoticias.com, 2012)

## **Problema de Investigación**

En los últimos años las fuertes lluvias han provocado que los pasos a desnivel en la ciudad de Quito se inunden debido a que el sistema de alcantarillado y sumideros colapsa por la

presencia de basura o por que la lluvia viene acompañada de granizo, esto ha traído como consecuencia que se produzcan accidentes porque vehículos particulares y de servicio público como el trolebús quedan atrapados dentro de los pasos deprimidos que se encuentran inundados ya que los usuarios de los pasos deprimido o a desnivel no tienen la información del nivel de agua exacto por lo tanto no pueden determinar si el vehículo puede circular dentro del paso deprimido.

La solución a este problema que ha dado el Municipio de Quito es la limpieza de sumideros y alcantarillas además de colocar señalética de advertencia.

En otros países, los pasos deprimidos o a desnivel presentan problemas de inundaciones similares a los de la ciudad de Quito, en estos países también se ha optado por realizar limpiezas de los sistemas de drenaje, construir más ramales de drenaje y colocar señalética de advertencia manual, inclusive se ha llegado a la decisión de suspender definitivamente la circulación por estos pasos.

Existen varios sistemas y proyectos electrónicos principalmente a nivel estudiantil que permiten detectar el nivel de agua, ninguno de estos sistemas son aplicados a detectar el nivel de agua en los pasos a desnivel y peor aún a controlar el ingreso a los mismos ante la presencia de un excesivo nivel de agua.

El sistema de señalización y advertencia instalado en los pasos deprimidos o pasos a desnivel del Distrito Metropolitano de Quito no funciona adecuadamente y no cumple con su objetivo de advertir a los usuarios de que no ingresen con sus vehículos ante una inundación. La señalética utilizada pasa desapercibida para los usuarios, además con el tiempo la línea de referencia de nivel de agua en el que el paso es peligroso se ha borrado.

No existe un sistema automático que permita suspender el acceso al paso a desnivel ante eventuales situaciones como accidentes de tránsito o mantenimiento programados.

## **OBJETIVOS**

### **Objetivo General**

- Estudiar, diseñar e implementar un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de señalización.

### **Objetivos Específicos:**

- Estudiar la tecnología necesaria que permita determinar niveles de agua.

- Diseñar un sistema electrónico capaz de detectar niveles de agua y advertir con señales visuales a los usuarios de vehículos que circulan por los pasos deprimidos.
- Implementar un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos que permita bloquear el ingreso de los vehículos al paso deprimido o desnivel mediante una barrera controlada automática o manualmente.
- Comprobar el funcionamiento del prototipo.



# CAPÍTULO I

## FUNDAMENTACIÓN TEÓRICA

### 1.1. Introducción

Para realizar el presente trabajo es necesario conocer y entender varios temas sobre electrónica básica, electrónica digital, microcontroladores, sensores, dispositivos de visualización como los LCD, módems de tecnología celular y programación, además de software especializados para simulaciones, programación y elaboración de tarjetas electrónicas .

### 1.2. Marco Teórico

#### 1.2.1. Microcontroladores

Los microcontroladores están ganando cada vez más campos de aplicación, mejorando constantemente el manejo y la seguridad en sus funciones. Su trabajo es interno, secreto y silencioso, sin que se les pueda reconocer desde afuera. Algunos ejemplos de aplicaciones son una lavadora en donde el microcontrolador es el que se encarga de lavar con varios programas de lavados diferentes, algunos tienen la capacidad de detectar la cantidad de agua y detergente. (Harth, 2008)

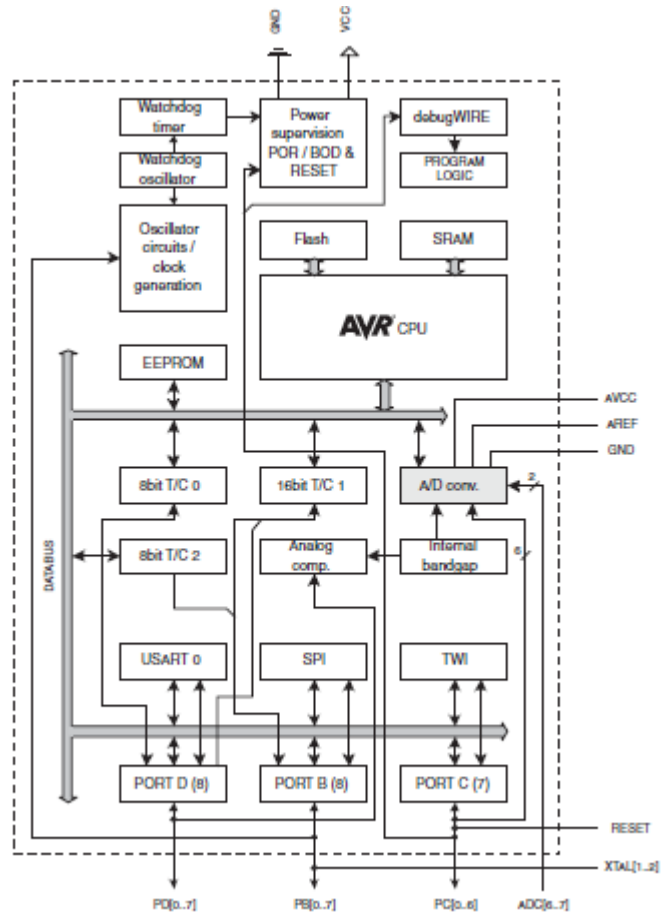


**Figura 1.1 Microcontrolador**

Fuente: (Atmel, 2010)

En los sistemas antibloqueo (ABS), airbags, o aparatos médicos, los microcontroladores pueden salvar vidas y vigilar las constantes vitales. Es fácil comprender que para este tipo de aparatos se requiere seguridad en sus funciones fuera de lo común. Y esa seguridad no se consigue sólo través del hardware, sino, sobre todo, a través del software, normalmente invisible y, por lo mismo, injustamente no tomado en cuenta, por eso es necesario para garantizar el funcionamiento conocer el hardware y software.

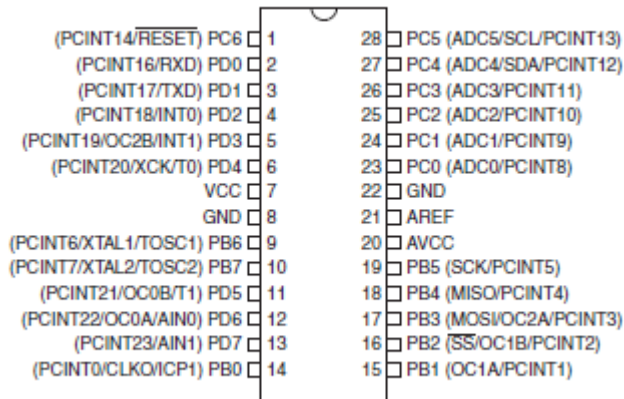
El uso creciente de los microcontroladores se debe, por último, a que a menudo permiten incorporar nuevas funciones sin necesidad de modificar el hardware y con pocos cambios en el software.



**Figura 1.2 Diagrama de Bloques del Microcontrolador**

Fuente: (Atmel, 2010)

Para que pueda controlar algún sistema o proceso es necesario crear y luego grabar en la memoria EEPROM del microcontrolador algún programa, generalmente se escribe en lenguaje ensamblador u otro lenguaje que entiendan los microcontroladores; se crea un archivo ejecutable el cual es quemado o grabado en el microcontrolador para que este ejecute las instrucciones, rutinas o subrutinas que controlarán el proceso.



**Figura 1.3 Pines de un Microcontrolador PDIP**

Fuente: ATMEL

### 1.2.2. Programación

La programación es un gran recurso que permite crear diversas secuencias de pasos lógicos que van a satisfacer nuestras necesidades y las de nuestros sistemas. Es el proceso de diseñar, escribir, probar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado. (Robótica, 2012)



**Figura 1.4 Programación**

Fuente: Libro Básico de Arduino

### 1.2.3. Arduino

Arduino es una plataforma de software y hardware libre para la creación de proyectos, flexible y fácil de usar. (Arduino, 2014).

Se creó para estudiantes, profesionales o simplemente aficionados que quieren incursionar en la electrónica y control.

“Arduino puede ser programado para tomar información del mundo que le rodea a través de sus pines de entrada, para esto existe una gran variedad de sensores que puede ser usada y de esta manera puede controlar su entorno mediante luces, pantallas motores y

otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos con Arduino pueden ejecutarse sin la necesidad de que éste permanezca conectado a un computador". (Robótica, 2012)

Los proyectos de Arduino pueden ser autónomos o comunicarse con un programa (software) que se ejecute en un computador (ej. Flash, Processing, MaxMSP). El software de desarrollo es abierto y se puede descargar gratis.



**Figura 1.5 Logotipo de Arduino**

Fuente: <http://.arduino.cc/>

#### **1.2.3.1. Cuerpo de un programa en Arduino**

Arduino se programa en el lenguaje de alto nivel C/C++ y generalmente tiene los siguientes componentes para elaborar el algoritmo:

- Estructuras
- Variables
- Operadores matemáticos, lógicos y booleanos
- Estructuras de control (Condicionales y ciclos)
- Funciones

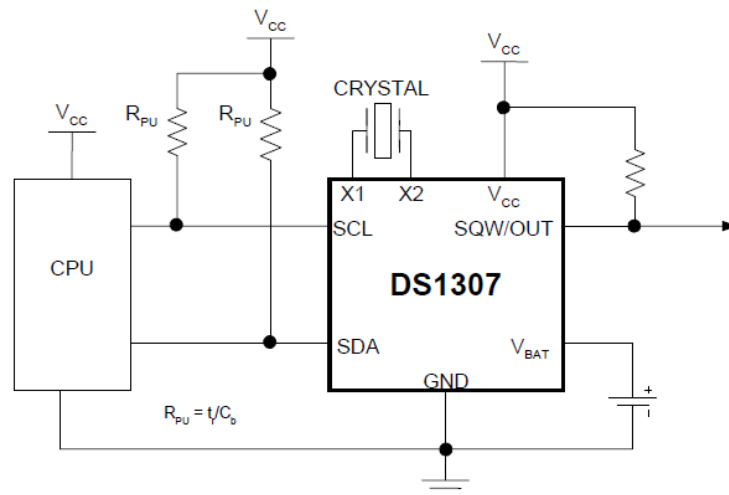
#### **1.2.4. Microcontroladores Atmel (AVR 8)**

Rendimiento, eficiencia y flexibilidad como ningún otro, de uso fácil y de bajo consumo de energía, y un alto nivel de integración, Atmel® AVR® de 8 y 32 bits microcontroladores complementan el Atmel línea de SMART de microcontroladores y microprocesadores basados en ARM ® para ofrecer una combinación única de rendimiento, eficiencia energética y flexibilidad de diseño. (Corporation, 2015)

Este microcontrolador está optimizado para aplicaciones que requieren alto rendimiento y facilidad de uso. Todos los dispositivos AVR se basan en la misma arquitectura y son compatibles con otros modelos AVR, tienen el ADC integrado y suficiente memoria.

### 1.2.5. Reloj en Tiempo Real DS1307

El DS1307 serial real-time clock (RTC) es de bajo consumo, utiliza código decimal binario (BCD) para el reloj y calendario más 56 bytes de NV SRAM. Los datos y direcciones son transferidos en serie a través de un bus bidireccional en  $I^2C$ . El reloj y calendario provee segundos, minutos, horas, la fecha con día mes y año. El reloj opera en los formatos 24 - horas o 12 - horas con el indicador de AM / PM.



**Figura 1.6 Circuito típico del RTC**

Fuente: (Semiconductor, 2006)

### 1.2.6. Protocolo $I^2C$

$I^2C$  Es un protocolo de comunicación serie diseñado por Philips que se utiliza esencialmente entre dispositivos que pertenecen al mismo circuito, por ejemplo, sensores con un microcontrolador.

#### 1.2.6.1. Características del protocolo:

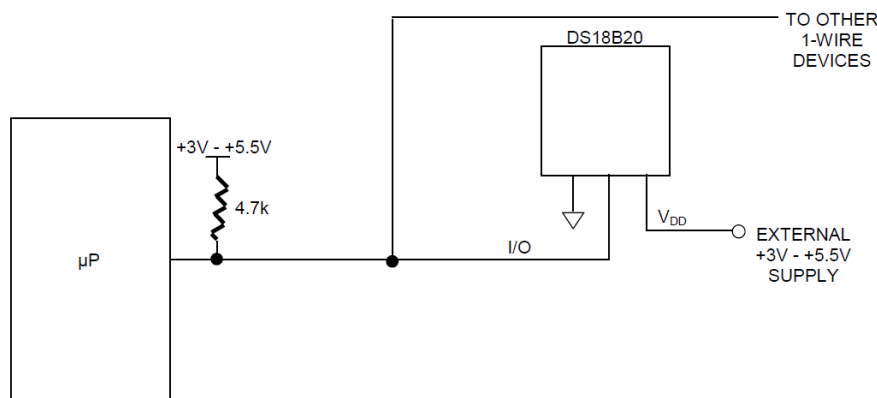
- Es una comunicación de tipo half duplex. Comunicación bidireccional por la misma línea pero no simultáneamente bidireccional.
- Velocidad standard de 100Kbit/s (100kbaudios). Se puede cambiar al modo de alta velocidad (400Kbit/s)
- Configuración maestro/esclavo. La dirección del esclavo se configura con software
- Sólo se necesitan dos líneas:

- SDA (Serial Data Line): Línea de datos.
- SCL/CLK (Serial Clock Line): Línea de reloj, será el que marque el tiempo de RW (Lectura/Escritura)
- La comunicación siempre tiene la estructura siguiente:
  - Transmisor: Byte de datos (8 Bits)
  - Receptor: Bit llamado ACK de confirmación.

### 1.2.7. Sensor de Temperatura DS18B20

El termómetro digital DS18B20 provee 9 a 12 bits (configurable) de lecturas de temperatura que indican la temperatura del dispositivo.

La información se envía a y desde el DS18B20 a través de una interfaz 1-Wire, por lo que se necesita sólo un hilo (y tierra) ser conectado desde un microcontrolador central a un DS18B20. La energía para la lectura, la escritura, y realizar conversiones de temperatura se pueden derivar de la propia línea de datos sin necesidad de una fuente externa. (Semiconductor, 2006)



**Figura 1.7 Circuito típico Sensor de Temperatura**

Fuente: (Semiconductor, 2006)

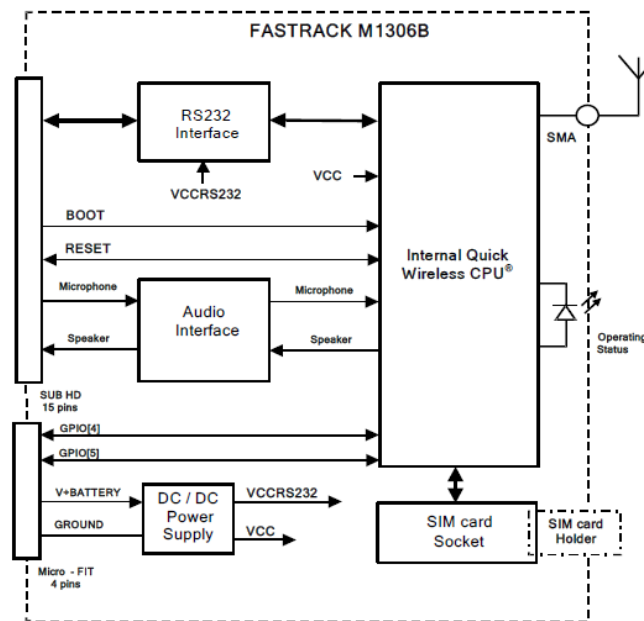
### 1.2.8. Modem Celular WAVECOM M1306B

El módem multibanda WAVECOM M1306B GSM-GPRS-SMS, está diseñado para aplicaciones de comunicación de datos, fax, SMS y voz. Puede operar en 4 bandas: 850, 900, 1800 y 1900 Mhz, según estas bandas de operación podría utilizarse con cualquiera de las operadoras existentes en el Ecuador.

La aplicación típica del modem es: en sistemas de telemetría y control supervisado, o bien

envío programado de mensajes SMS ya sea a móviles o a otras computadoras, utilizando software residente en la PC o en un microcontrolador.

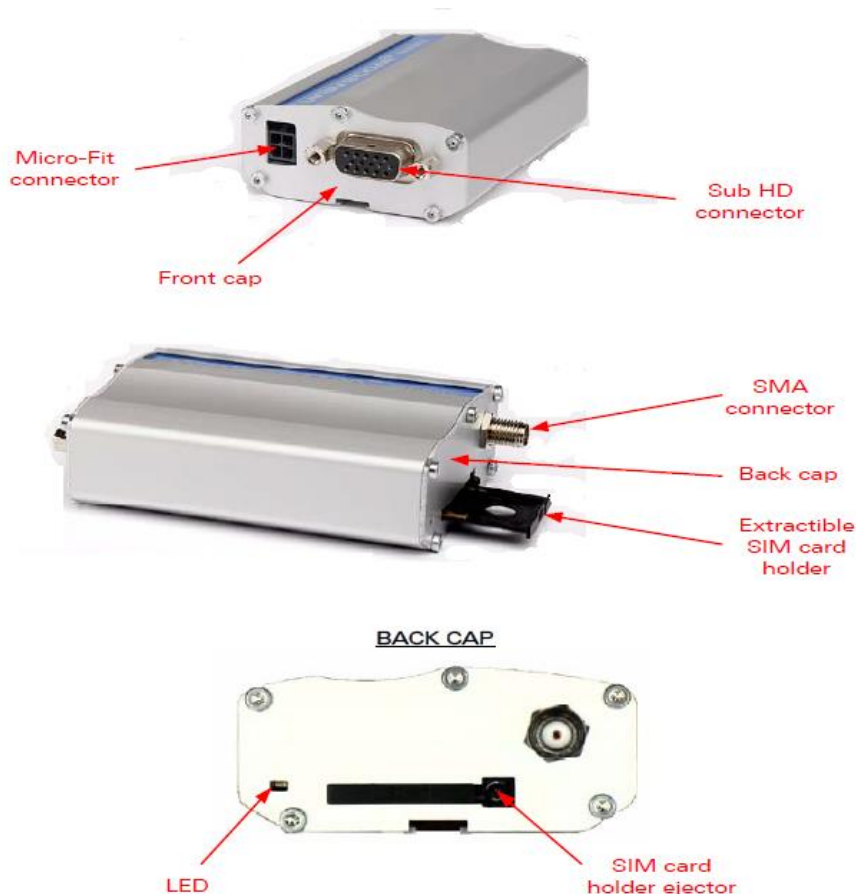
El módem se controla a través de un conector DB9, con interfaz estándar RS232, y baud rate programable desde 300 a 115200 bps. Desde la computadora PC-Windows, se conecta al puerto USB a través de un cable USB-Serial, y puede emplearse el programa Hyperterminal para la configuración inicial y pruebas preliminares del módem, ya que opera con un set extendido de comandos AT.



**Figura 1.8 Arquitectura funcional del Modem**

Fuente: (WAVECOM, 2006)

El módem GSM funciona como un teléfono celular y ya incluye su receptáculo para chip SIM. El módem WAVECOM puede operarse a través de un microcontrolador conectado a su puerto serial. Entre sus aplicaciones con microcontroladores, están el control remoto supervisando dispositivos como motores, actuadores, válvulas, sistemas de riego, puertas eléctricas. Todos estos dispositivos pueden operarse por medio de comandos remotos enviados a través de un móvil, con mensajes cortos SMS.



**Figura 1.9 Fastrack M1306B descripción General**

Fuente: (WAVECOM, 2006)

### 1.2.9. Comunicación Serial RS232

Un dispositivo serial utiliza un protocolo de comunicación que es estándar para casi cualquier PC, y este fue el protocolo de comunicación que se utilizó para la comunicación entre el Modem Celular y el microcontrolador. “Las placas Arduino tienen un puerto serial (también conocido como UART o USART). Este se comunica a través de los dos pines de conexión RX (pin 0) y TX (pin 1).” (Arduino, 2014)

### 1.2.10. Circuito de Interface serial MAX232

Es un circuito integrado de Maxim que convierte las señales de un puerto serie RS-232 a señales compatibles con los niveles TTL de circuitos lógicos. El MAX232 sirve como interfaz de transmisión y recepción para las señales RX, TX, CTS y RTS. (Fundación Wikimedia, Inc., 2013)



### 1.2.11. Motores Paso a Paso

Los motores Paso a Paso son diferentes a los motores de Corriente Continua, estos giran a un ángulo determinado según se requiera, es decir, en cada paso hace girar su eje cierto grado y se queda en esa posición hasta que no se cambie el voltaje en las bobinas. Estos tienen una fuerza considerable por lo que no es necesario un mecanismo de reducción.

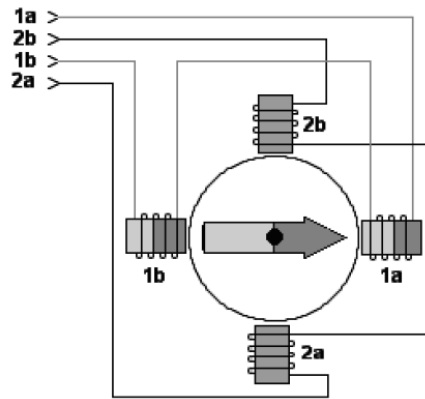


Figura 1.10 Modelo conceptual de un Motor Paso a Paso

Fuente: [www.EasyBots.com](http://www.EasyBots.com)

### 1.2.12. Display LCD 20 x 4

Es un dispositivo que permite visualizar en una pantalla de 20 caracteres por 4 líneas, los mensajes generados por un usuario o por las condiciones del circuito.

### 1.2.13. LED IR

El diodo emisor de luz infrarroja, emite un tipo de radiación electromagnética infrarroja, que no se puede ver con el ojo humano porque su longitud de onda es mayor al espectro visible para el ojo humano.

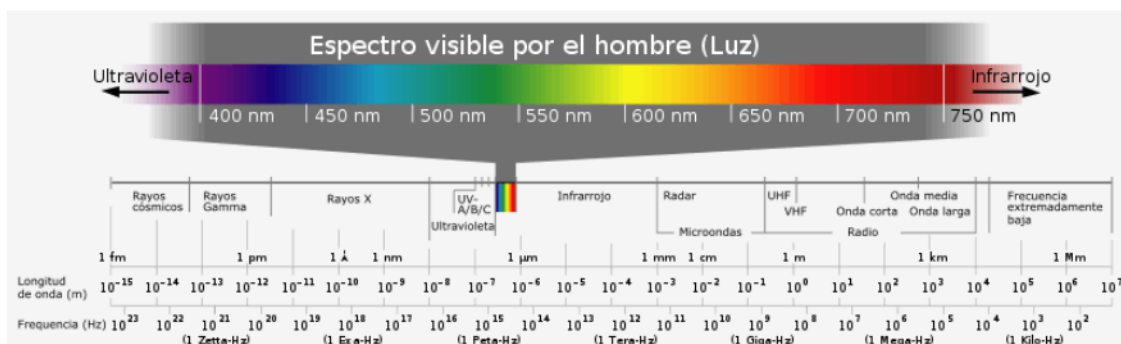


Figura 1.11 Espectro de Luz Visible

Fuente: [www.circuitoselectronicos.org](http://www.circuitoselectronicos.org)

#### 1.2.14. Fototransistor

Se diferencia de un transistor común por su base ya que se sustituye por un cristal fotosensible que regula el flujo de corriente colector – emisor según la luz que incide sobre él, para este trabajo luz infrarroja.

#### 1.2.15. Switch de Nivel

El SW de nivel escogido permite sensor la presencia de agua y este cambia su estado de NO a NC o al contrario dependiendo la posición en que se lo instale y la necesidad del diseño.



**Figura 1.12 Switch de Nivel**

Fuente: Fabricante

#### 1.2.16. OrCAD

OrCAD es un software propietario que se utiliza para crear diagramas electrónicos e imprimir los diagramas en las tarjetas y de esta manera crear las PCB.



**Figura 1.13 Logotipo de OrCAD (Diseño de PCB)**

Fuente: Fabricante

#### 1.2.17. Proteus

Proteus es un software especializado principalmente para electrónica, dentro de sus principales funciones esta la simulación de circuitos que se realiza con el módulo ISIS.



**Figura 1.14 Logotipo Proteus (Simulación)**

Fuente: Fabricante

### 1.2.18. Software Arduino

Arduino dispone de su propio software en lenguaje ensamblador. Arduino se programa en lenguaje C/C++.



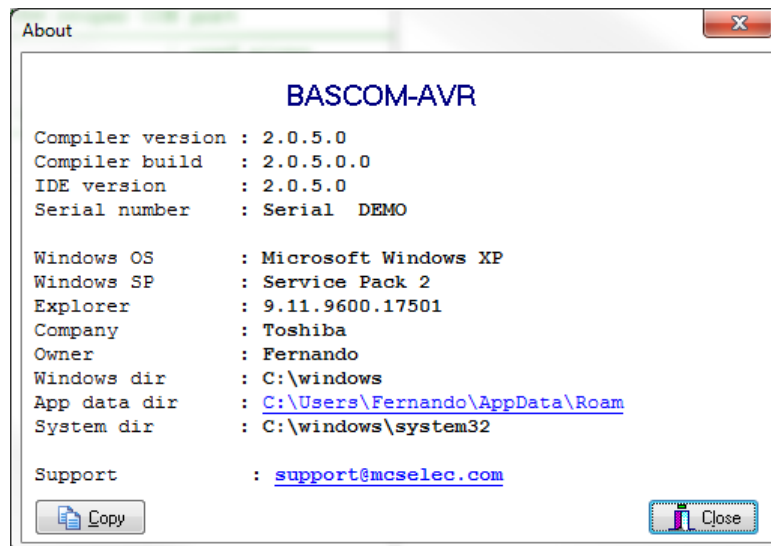
**Figura 1.15 Logotipo Software Arduino**

Fuente: Fabricante

### 1.2.19. Software BASCOM – AVR

Este software es el compilador para los microcontroladores de la familia AVR de Atmel, funciona en varios sistemas operativos, dispone de una programación estructurada con sentencia, genera el código máquina y las variables y etiquetas pueden tener una longitud de 32 caracteres.

Este software dispone de un simulador que sirve para verificar que el programa este correcto antes de cargarlo al microcontrolador.



**Figura 1.16 Información de la versión Instalada**

Fuente: Software Instalado

### **1.2.20. Medición de Nivel**

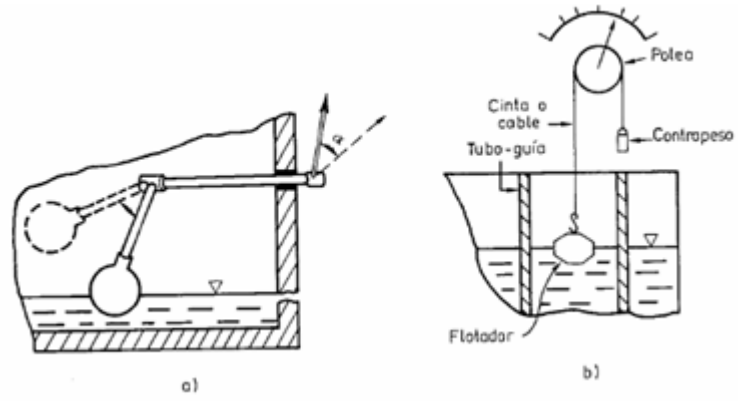
En general la medición de nivel es importante, principalmente en la industria o en aplicaciones donde está de por medio la seguridad de las personas o la infraestructura para el correcto funcionamiento de un proceso. Los instrumentos de nivel pueden dividirse en medidores de nivel de líquidos y de sólidos, que son dos mediciones claramente diferenciadas.

#### **1.2.20.1. Medidores de nivel de líquidos**

Los medidores de nivel de líquidos trabajan midiendo, bien directamente la altura del líquido sobre una línea de referencia, bien la presión hidrostática, bien el desplazamiento producido en un flotador por el propio líquido contenido, bien aprovechando características eléctricas del líquido o bien utilizando otros fenómenos. (Rojas, 2008)

#### **1.2.20.2. Sensores de nivel**

La mayoría de estos dispositivos se basan en un flotador que se conecta de diversas formas a sensores de movimiento o también pueden funcionar con base en sensores de presión diferencial. La Figura 17 muestra 2 tipos distintos de sensores de nivel, en Figura (a) y Figura (b) se trata de sensores basados en flotadores.



**Figura 1.17 Ejemplo de medidores de nivel por flotador**

Fuente: (Rojas, 2008)

## CAPÍTULO II

### **BREVE DESCRIPCIÓN DEL PROCESO INVESTIGATIVO REALIZADO PARA LA IMPLEMENTACIÓN DE UN SISTEMA ELECTRONICO DE SEGURIDAD Y ADVERTENCIA PARA LOS PASOS A DESNIVEL.**

#### **2.1. Problema Principal**

El sistema de señalización y advertencia instalado en los pasos deprimidos o pasos a desnivel del Distrito Metropolitano de Quito no funciona adecuadamente y no cumple con su objetivo de advertir a los usuarios de que no ingresen con sus vehículos ante una inundación. La señalética utilizada pasa desapercibida para los usuarios, además con el tiempo la línea de referencia de nivel de agua en el que el paso es peligroso se ha borrado.

No existe un sistema automático que permita suspender el acceso al paso a desnivel ante eventuales situaciones como accidentes de tránsito o mantenimiento programados.

#### **2.2. Por Qué y Para Qué de los Objetivos**

En este proyecto se diseñará e implementará un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, y se reemplazará o complementará los actuales sistemas de advertencia porque en los actuales momentos donde la tecnología ha ido evolucionando vertiginosamente no es concebible que se mantengan sistemas manuales en aplicaciones donde está involucrada la seguridad y bienestar de las personas.

En este proyecto se diseñará un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia y mejorar de esta manera la seguridad de los usuarios así como brindar una herramienta de control adicional para los entes encargados de la seguridad, vigilancia y mantenimiento en los pasos deprimidos o a desnivel.

#### **2.3. Hipótesis o Idea a Defender**

Se puede demostrar que al implementar el sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, y reemplazar o complementar los actuales sistemas de advertencia, reducirá el riesgo de accidentes

causados por inundaciones de los puentes y brindará una herramienta efectiva para bloquear el acceso al paso a desnivel en caso de requerir un mantenimiento.

### **2.3.1. Variables**

#### **2.3.1.1. Independientes:**

Sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia.

#### **2.3.1.2. Dependientes:**

Demostrar que se puede mejorar la seguridad y bienestar para los usuarios de los pasos deprimidos o a desnivel en el distrito Metropolitano de Quito. Dotar de una herramienta de control para los entes encargados de la seguridad, vigilancia y mantenimiento de los pasos a desnivel.

## **2.4. Los métodos que se utilizaron para el desarrollo del proyecto son:**

### **2.4.1. Análisis y Síntesis:**

Para investigar en el mercado la disponibilidad de elementos electrónicos que tengan la capacidad de detectar la presencia de agua, de dispositivos que permitan controlar remotamente un sistema y que estas señales sean procesadas, se utilizó el método de análisis y síntesis

### **2.4.2. Modelación**

Para estudiar la factibilidad de desarrollar un sistema electrónico para advertir la presencia de agua a cierto nivel se utilizó el método de modelación.

### **2.4.3. Experimental**

Para diseñar e implementar un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia se utilizó el método experimental.

En el proceso investigativo se indagó en la información presentada por la prensa escrita, recopilando información de las páginas WEB de los periódicos de mayor circulación del país, con la información obtenida se determinó que el sistema de advertencia de presencia de agua en los pasos deprimidos o a desnivel del Distrito Metropolitano de Quito es ineficiente y que no ha permitido evitar accidentes causado por las inundaciones.

## **2.5. Resultados esperados del Proyecto**

Este prototipo puede ser puesto a consideración de los personeros del Municipio de Quito, para que sea utilizado en los puentes a desnivel o pasos deprimidos del Distrito Metropolitano de Quito para mejorar los sistemas de advertencia actuales y de esta manera mejorar la seguridad de los usuarios así como proveer un sistema de control para los entes de seguridad, vigilancia y mantenimiento.



## **CAPÍTULO III**

### **PRESENTACIÓN DE RESULTADOS**

El estudio y diseño de un prototipo de sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia, cumple lo siguiente:

#### **3.1. Requisitos del Proyecto**

##### **3.1.1. Requisitos Generales**

El prototipo tendrá la capacidad de:

- ✓ Detectar cinco niveles de agua.
- ✓ Mostrar cada nivel de agua en la pantalla (Display).
- ✓ En el quinto nivel de agua accionará una barrera para impedir el ingreso al paso a desnivel.
- ✓ Mostrar en la pantalla la hora, la fecha y la temperatura ambiente.

##### **3.1.2. Requisitos Específicos**

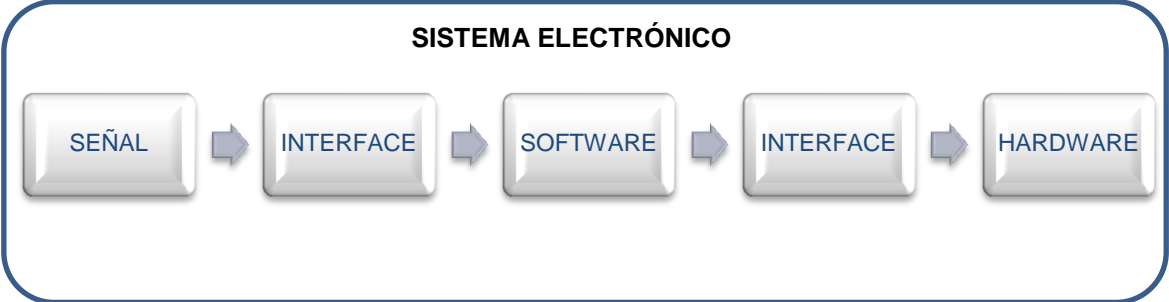
Los requerimientos específicos del proyecto son los siguientes:

- La barrera no se accionará si detecta la presencia de un vehículo.
- El control de la barrera se puede cambiar de automático a manual.
- La barrera puede ser accionada remotamente mediante una instrucción recibida por un modem celular.
- Se pueden enviar mensajes cortos al modem celular para ser presentados en la pantalla.
- El sistema estará compuesto por:
  - Un Microcontrolador Arduino Uno
  - Dos microcontroladores AVR 8
  - Un motor paso a paso
  - Un Modem Celular WAVECOM M1306B
  - Un CI RTC (Real Time Clock)
  - Un sensor de temperatura
  - Fuente de Alimentación 12 V DC
  - Batería libre mantenimiento de 12 V DC

#### **3.2. Diagrama de Bloques General**

El sistema está compuesto por sensores cuyas señales son recibidas por los

microcontroladores mediante protocolos de comunicación, estas señales son procesadas para ejecutar subrutinas que determinan las acciones a tomar según las condiciones.

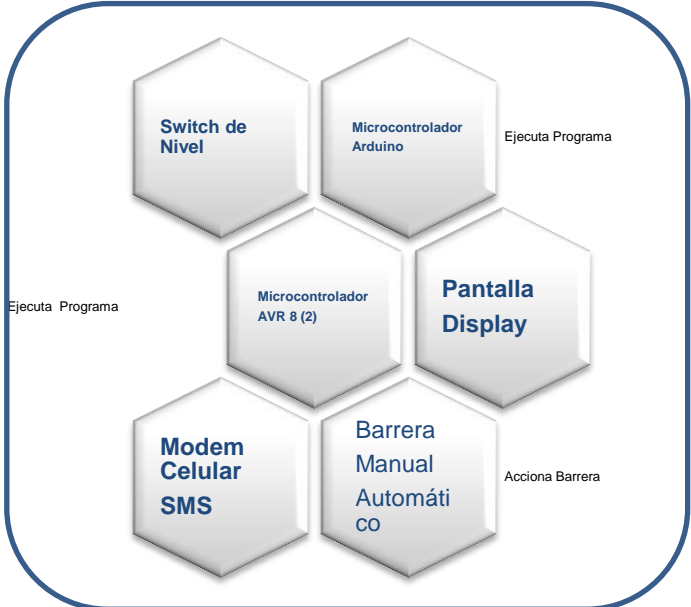


**Figura 3.2.1 Diagrama de Bloques General**

Fuente: (Segura, 2015)

**3.3. Diagrama de Bloques del Hardware**

El sistema electrónico de advertencia de presencia de agua a cierto nivel en los pasos deprimidos de la ciudad de Quito, para reemplazar o complementar los actuales sistemas de advertencia consta de tres etapas principales, la primera etapa la controla el microcontrolador Arduino que recibe la señal de los Switch de nivel y presenta la información en el Display, cuando existe el nivel crítico de agua acciona el motor PaP (barrera) de bloqueo que es controlada por un AVR 8 (segunda etapa), la tercera etapa está formada por un modem celular controlada por otro AVR 8 el cual puede accionar la barrera remotamente.



**Figura 3.3.1 Diagrama de Bloques Hardware**

Fuente: (Segura, 2015)

### 3.4. Diagrama Esquemático del Sistema

La figura 21 representa el esquemático del sistema completo, en este se encuentran los principales componentes del circuito, se observa el Switch de nivel que cambiará de estado cuando detecta la presencia de agua, esta información es recibida y procesada por el Arduino quien presenta en el Display la información, si se acciona el Switch de nivel crítico, mediante la comunicación con el ACVR que controla el motor PaP activa la barra que bloquearía el acceso al paso a desnivel. También se encuentra el modem celular quien puede recibir una instrucción mediante un SMS y activar remotamente la barrera para subir o bajar, la barrera también puede ser operada de forma manual mediante un selector y pulsadores.

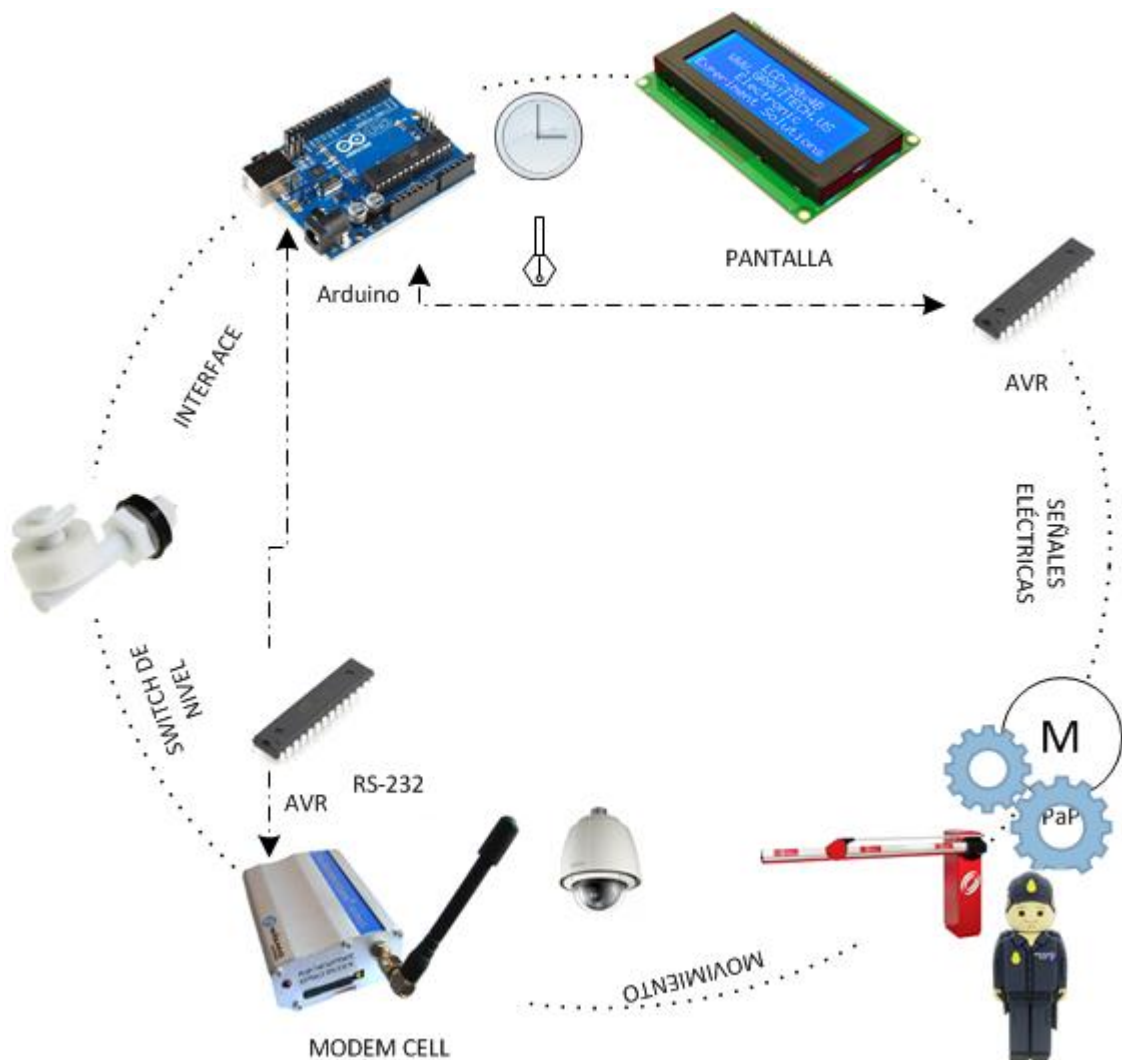


Figura 3.4.1 Diagrama Esquemático del Sistema

Fuente: (Segura, 2015)

### 3.5. Flujoograma del Sistema

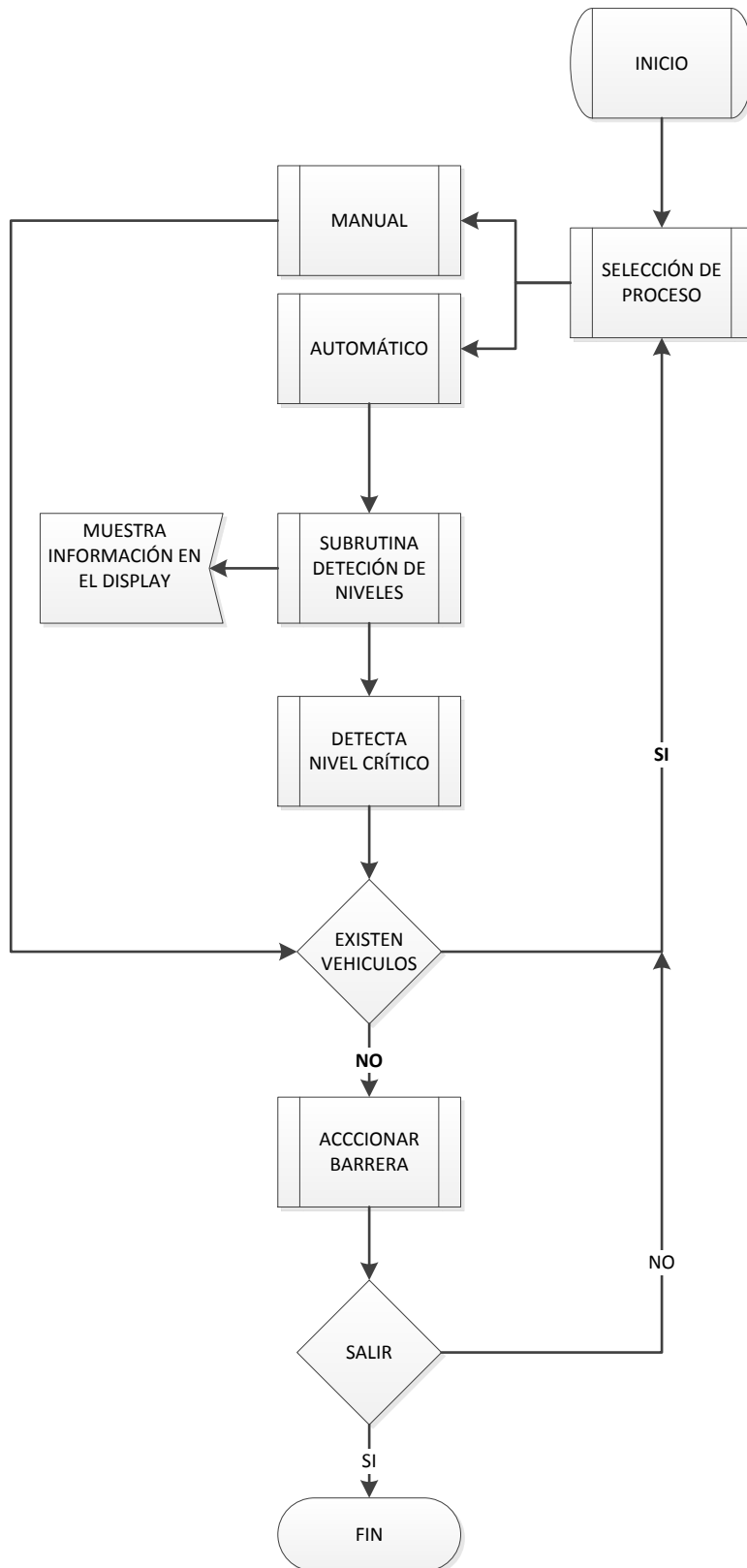


Figura 3.5.1 diagrama de flujo del programa

Fuente: (Segura, 2015)

### 3.6. Simulaciones (ISIS - Proteus)

Para garantizar el funcionamiento del sistema completo, se realizaron simulaciones de cada una de sus etapas de forma separada, para esta simulación se utilizó el software ISIS de Proteus.

#### 3.6.1. Simulación Switch de Nivel con Control Arduino

En esta simulación se pudo verificar la programación de las subrutinas en el Arduino al momento de accionar cada uno de los cinco Switch que representan los cinco niveles de agua a ser medidos dentro del paso a desnivel.

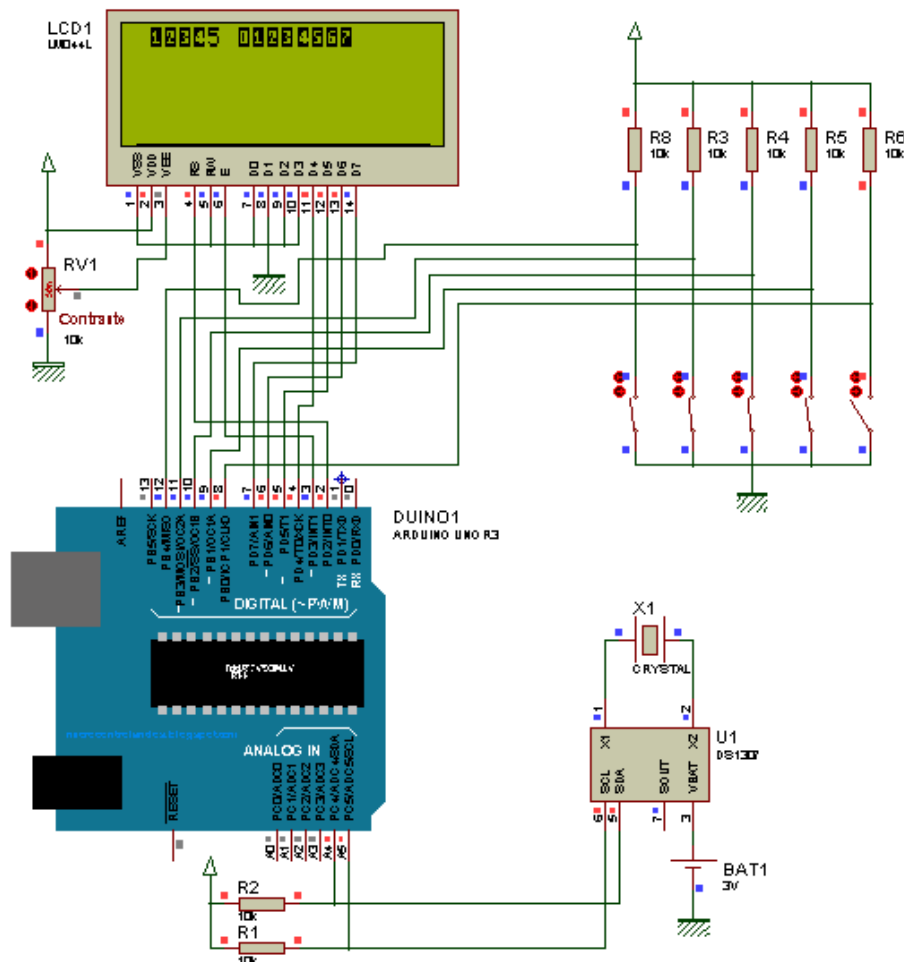


Figura 3.6.1.1 Simulación Switch de nivel con control Arduino

Fuente: (Segura, 2015)

### 3.6.2. Simulación Control de motor PaP

En esta etapa se hizo la simulación del control de motor PaP que es controlado por el AVR 8, también se incluye la simulación del manejo en forma manual de la barrera que impide el ingreso al paso a desnivel, el AVR también controla la alarma que se acciona al mismo tiempo que la barrera, existen Leds indicativos de la forma en que está trabajando la barrera.

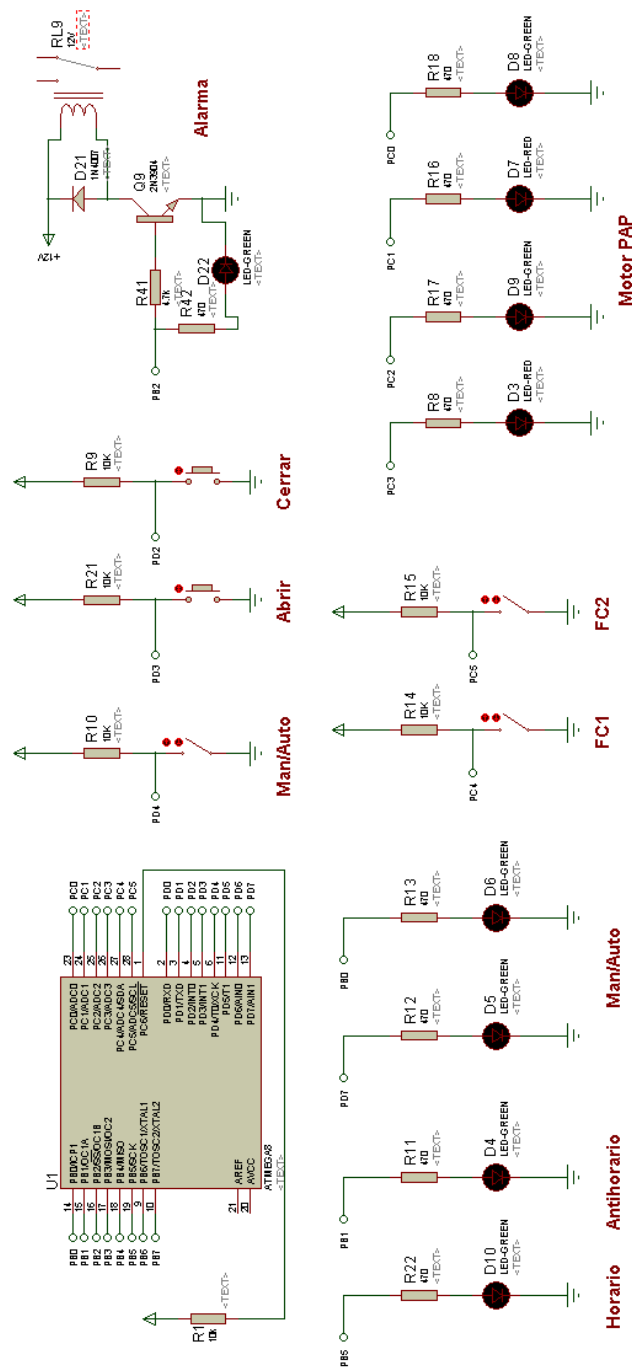


Figura 3.6.2.1 Simulación control de motor PaP

Fuente: (Segura, 2015)

### 3.6.3. Simulación Sensor de Temperatura DS18B20 y RTC DS1307

En la simulación del sensor de temperatura y del reloj en tiempo real se puede verificar la comunicación de estos dispositivos con el Arduino y que los datos se presentan en el Display.

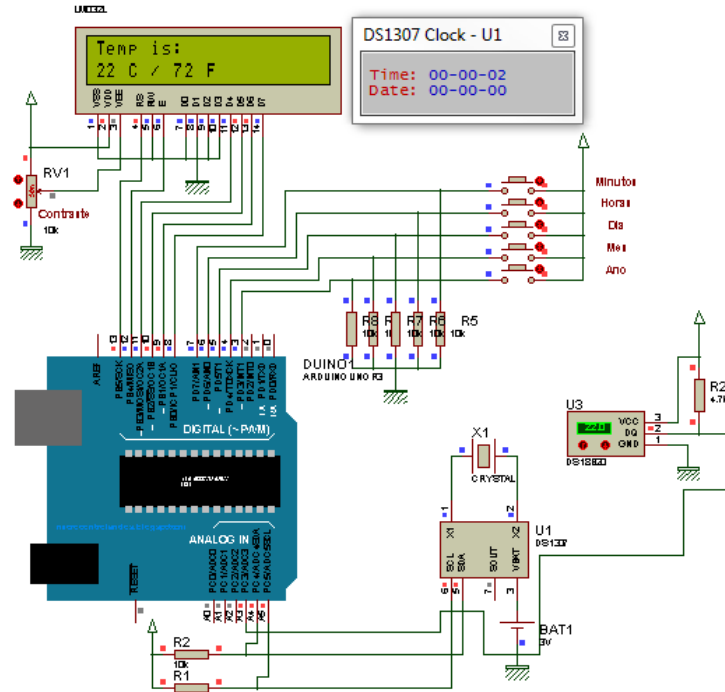


Figura 3.6.3.1 Simulación DS18B20 y DS1307

Fuente: [microcontrolandos.blogspot.com.br](http://microcontrolandos.blogspot.com.br)

### 3.6.4. Simulación Emisor Infrarrojo (IR)

En esta simulación se verifica que el emisor IR emita la luz a cierta frecuencia, esta puede ser regulada con el potenciómetro de precisión.

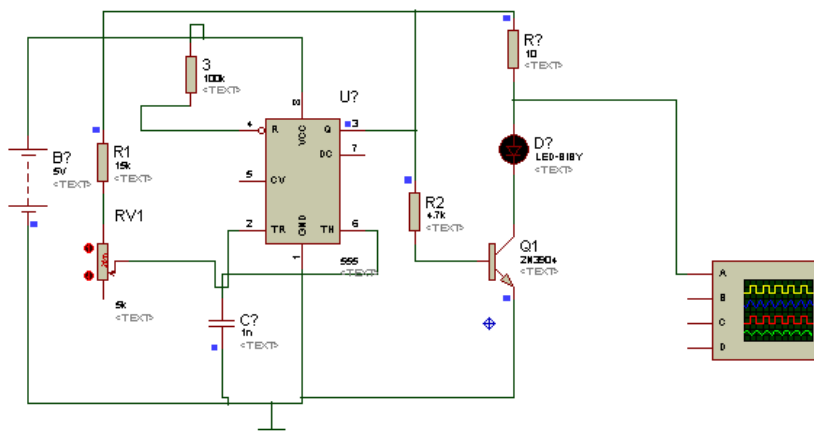


Figura 3.6.4.1 Simulación Emisor Infrarrojo

Fuente: (Segura, 2015)

### 3.7. Diseño Esquemático de Tarjetas

#### 3.7.1. Tarjeta Etapa Arduino – AVR 8 – Modem Celular

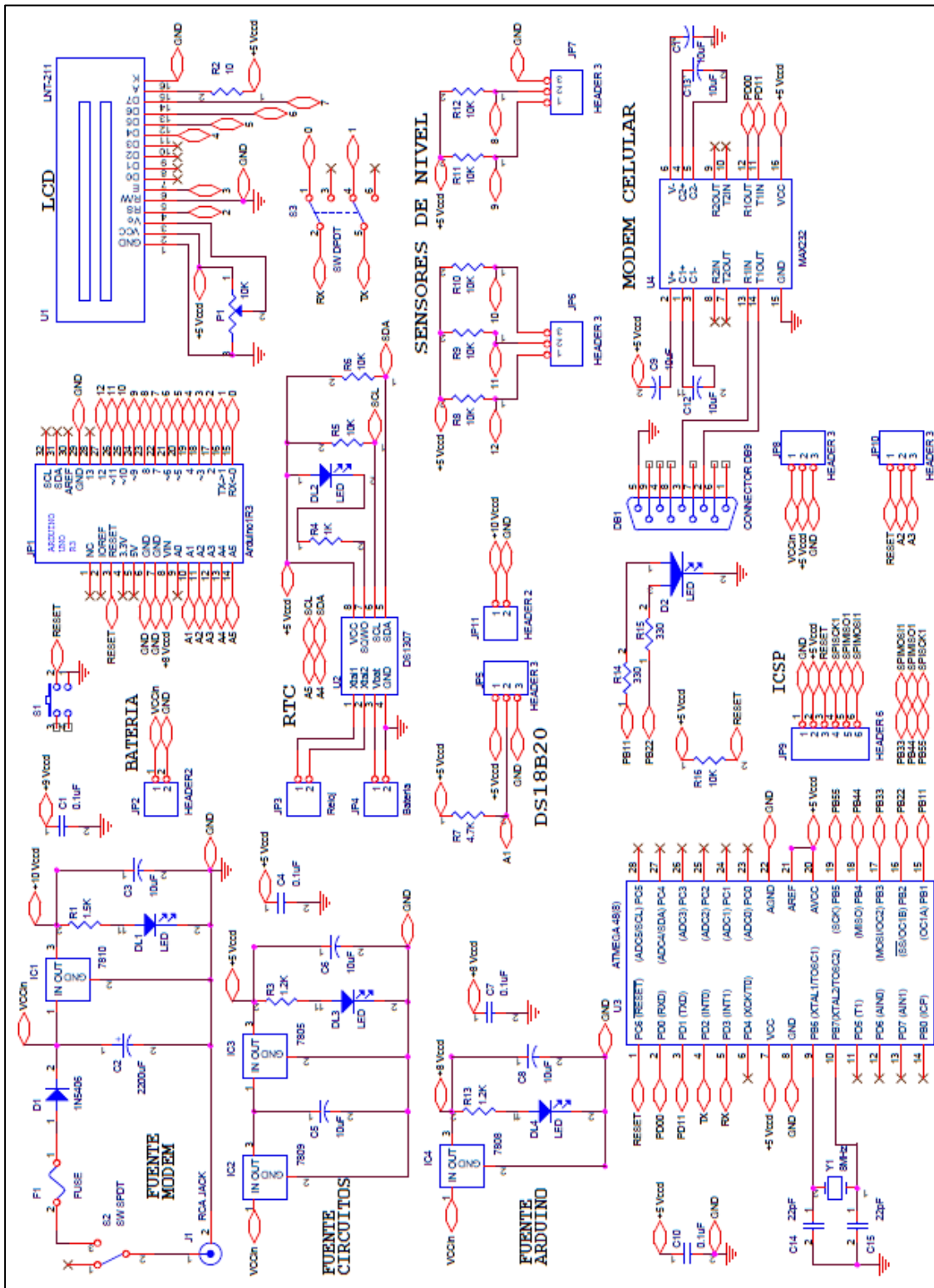


Figura 3.7.1.1 Diseño Tarjeta Arduino – AVR 8 – Modem Celular

Fuente: (Segura, 2015)



### 3.7.2. Tarjeta Etapa Control de motor PaP – Sensores IR - Alarma

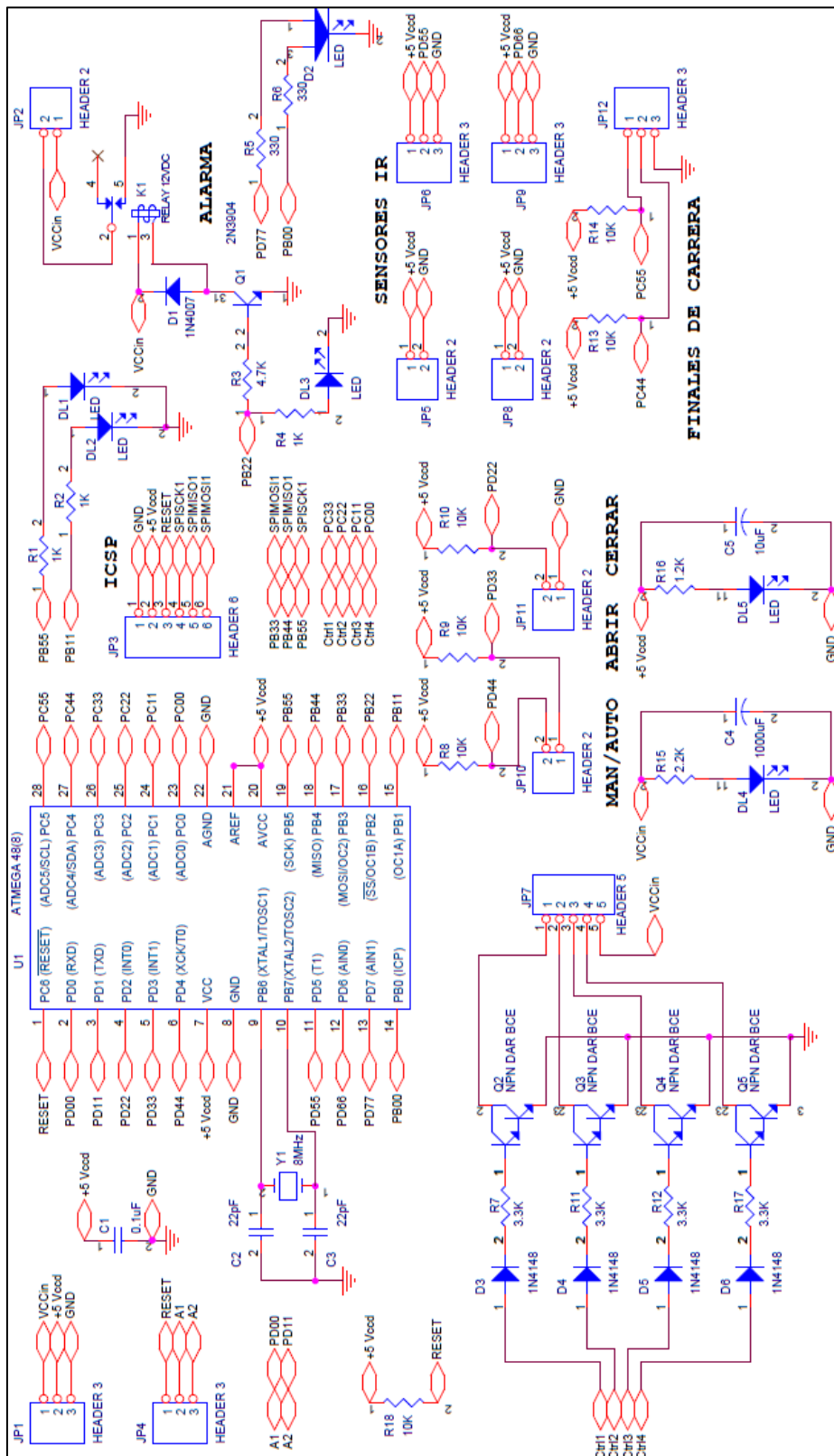


Figura 3.7.2.1 Esquemático Tarjeta de control motor PaP – Sensor IR – Alarma

Fuente: (Segura, 2015)

### 3.8. Circuito Impreso de Tarjetas (PCB)

El diseño de las PCB de cada etapa se lo realizó con el software ORCAD.

#### 3.8.1. PCB Etapa Arduino – AVR 8 – Modem Celular

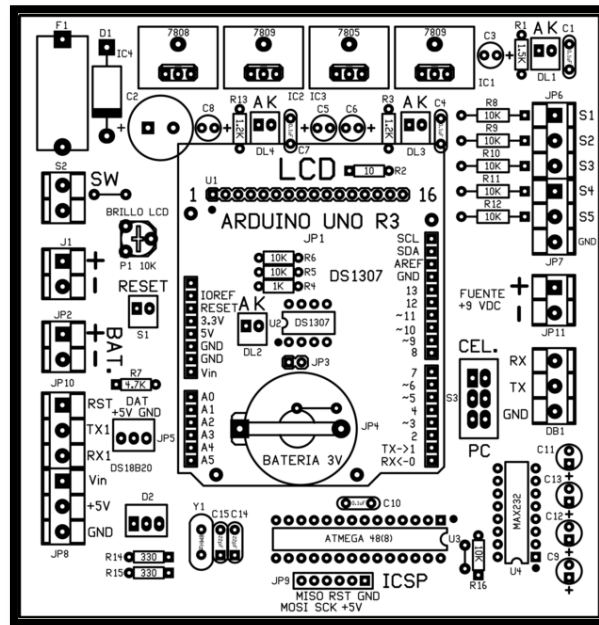


Figura 3.8.1.1 Distribución de elementos PCB etapa Arduino

Fuente: (Segura, 2015)

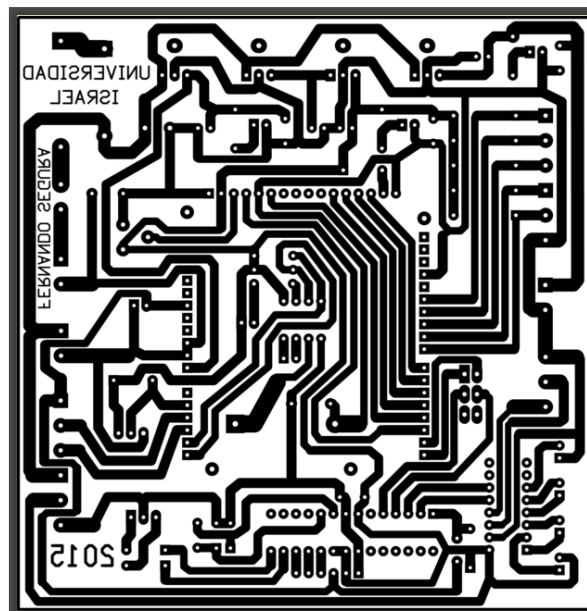


Figura 3.8.1.2 Pistas PCB Etapa Arduino

Fuente: (Segura, 2015)

### 3.8.2. PCB Etapa Control de motor PaP – Sensores IR - Alarma

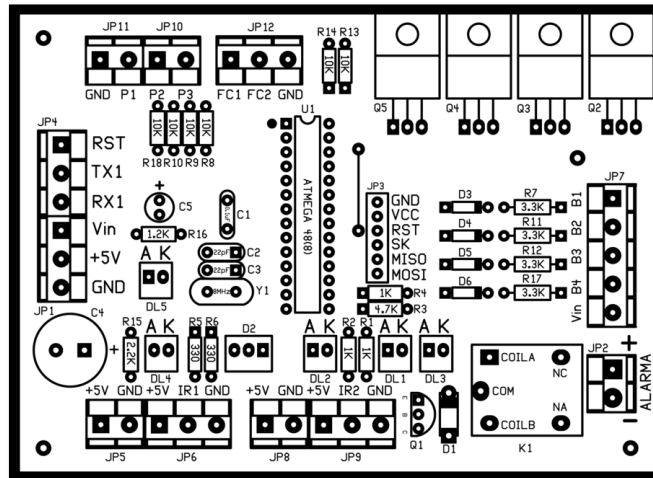


Figura 3.8.2.1 Distribución de elementos PCB etapa control PaP

Fuente: (Segura, 2015)

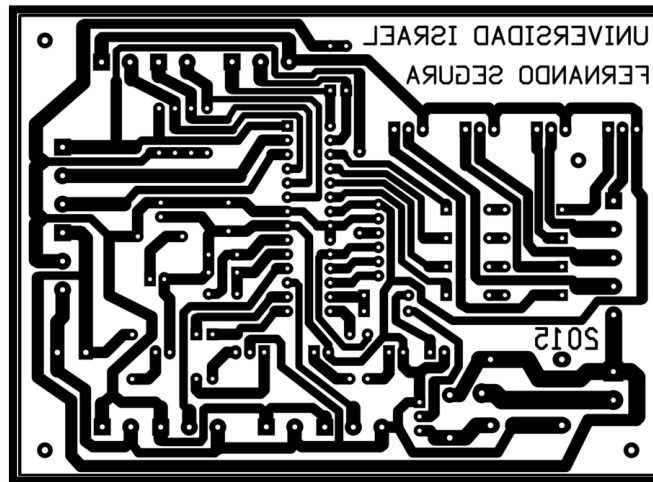


Figura 3.8.2.2 Pistas PCB etapa control PaP

Fuente: (Segura, 2015)

### 3.8.3. PCB Emisor IF

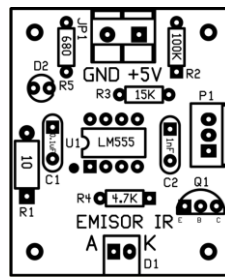


Figura 3.8.3.1 Distribución de elementos PCB Emisor IR

Fuente: (Segura, 2015)

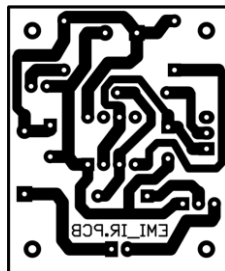


Figura 3.8.3.2 Pistas PCB Emisor IR

Fuente: (Segura, 2015)

### 3.8.4. PCB Receptor IF

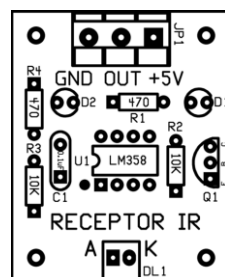
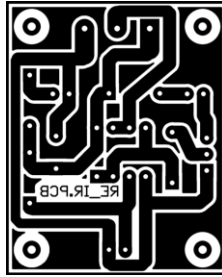


Figura 3.8.4.1 Distribución de elementos PCB Receptor IR

Fuente: (Segura, 2015)



**Figura 3.8.4.2 Pistas PCB Receptor IP**

Fuente: (Segura, 2015)

### **3.9. Implementación**

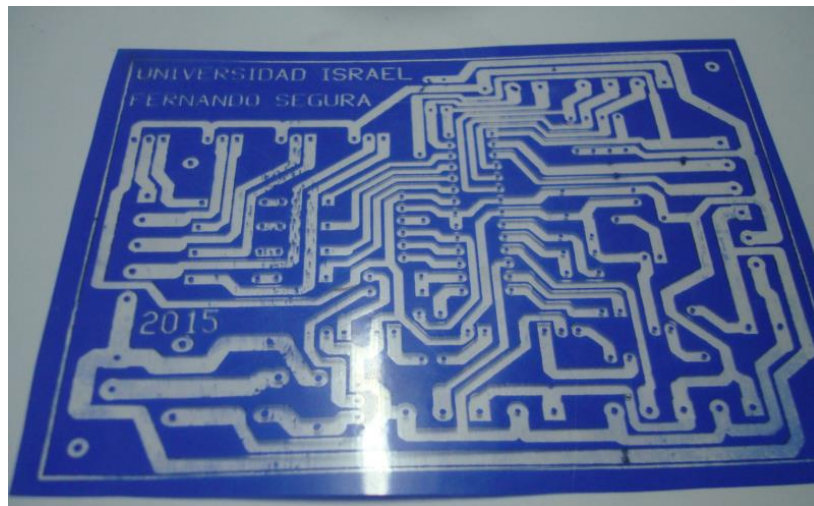
Una vez que se superó la etapa de simulación y diseño de las tarjetas, se procedió a la implementación de las mismas.

Se diseñó la placa principal de tal manera que tenga conexión directa con los pines del Arduino y así evitar la instalación de cables de comunicación.

#### **3.9.1. Procedimiento para la elaboración de las PCB**

Para cada placa se siguió el siguiente procedimiento:

- ✓ Impresión de las pistas en papel térmico.



**Figura 3.9.1.1 Impresión de Pistas en papel térmico**

Fuente: (Segura, 2015)

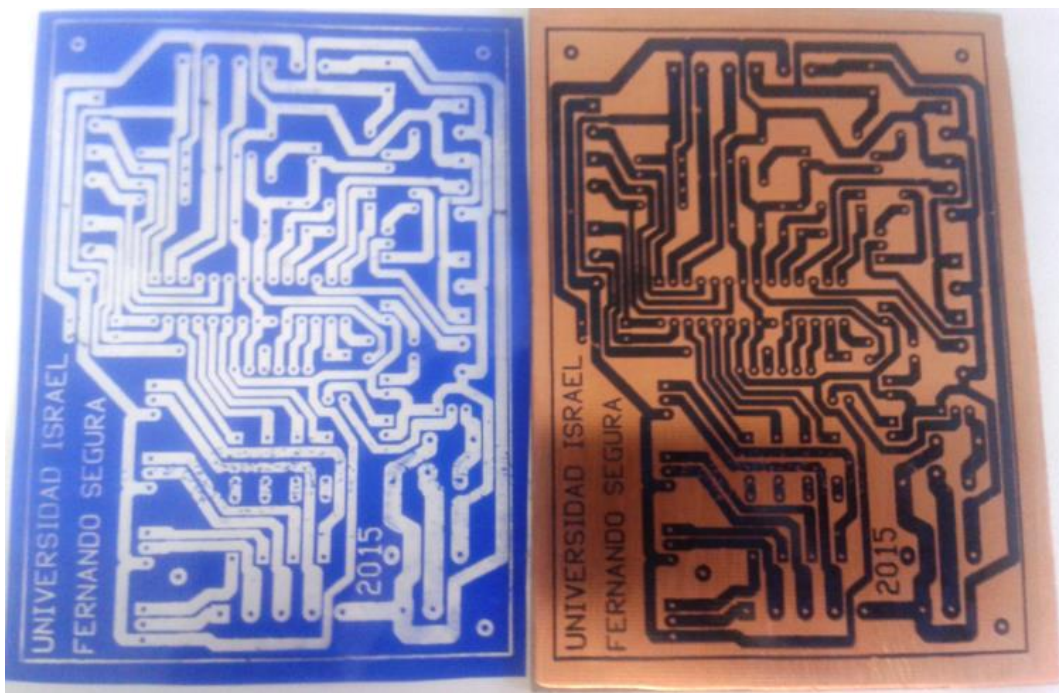
- ✓ Cortar la baquelita de acuerdo a la medida de la tarjeta diseñada



**Figura 3.9.1.2 Corte de Baquelita**

Fuente: (Segura, 2015)

- ✓ Termo-grabar las pistas en la tarjeta con una fuente de calor, en este caso se utilizó una plancha doméstica.

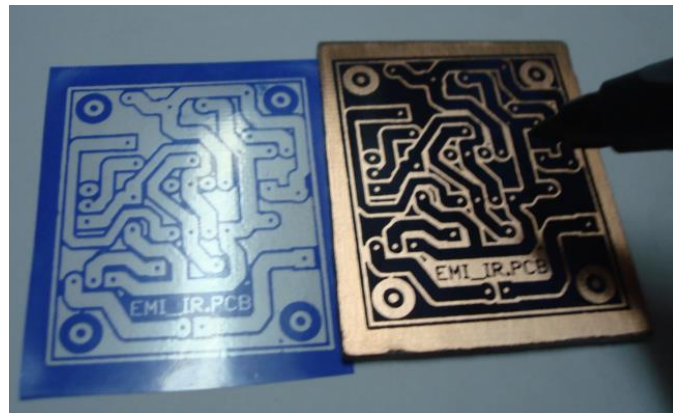


**Figura 3.9.1.3 Termografía de pistas en baquelita**

Fuente: (Segura, 2015)



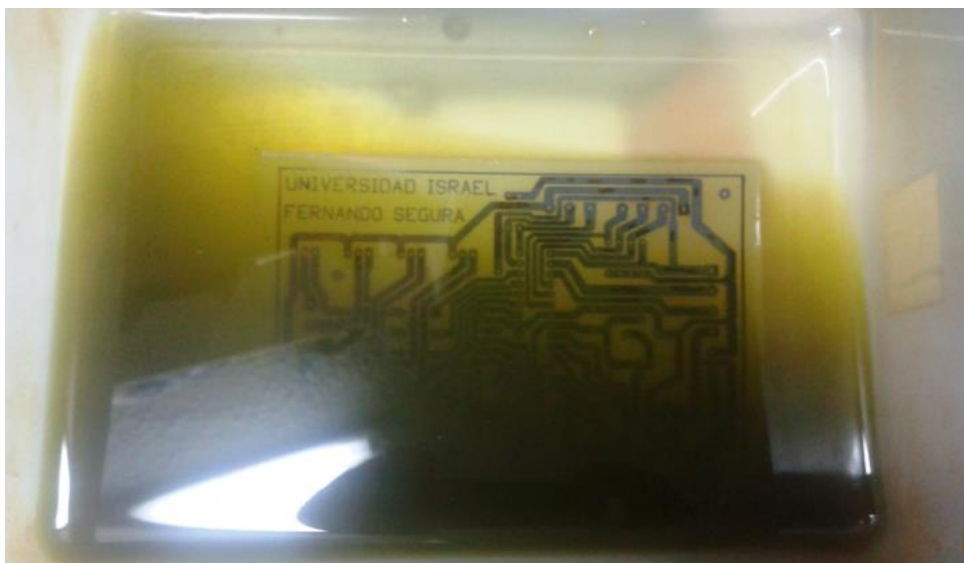
- ✓ Corregir fallas en las pistas que se pudieron haber presentado el momento de la Termografía.



**Figura 3.9.1.4 Ejemplo de corrección de fallas**

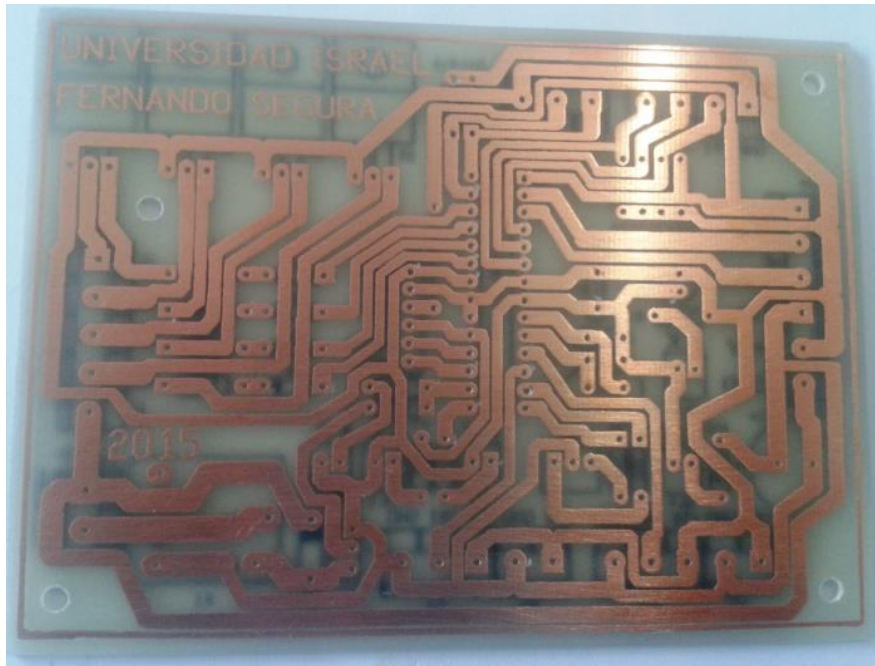
Fuente: (Segura, 2015)

- ✓ Una vez que se tienen las pistas en la placa se procede a remover el cobre excedente con ácido.



**Figura 3.9.1.5 Remoción de cobre excedente con ácido**

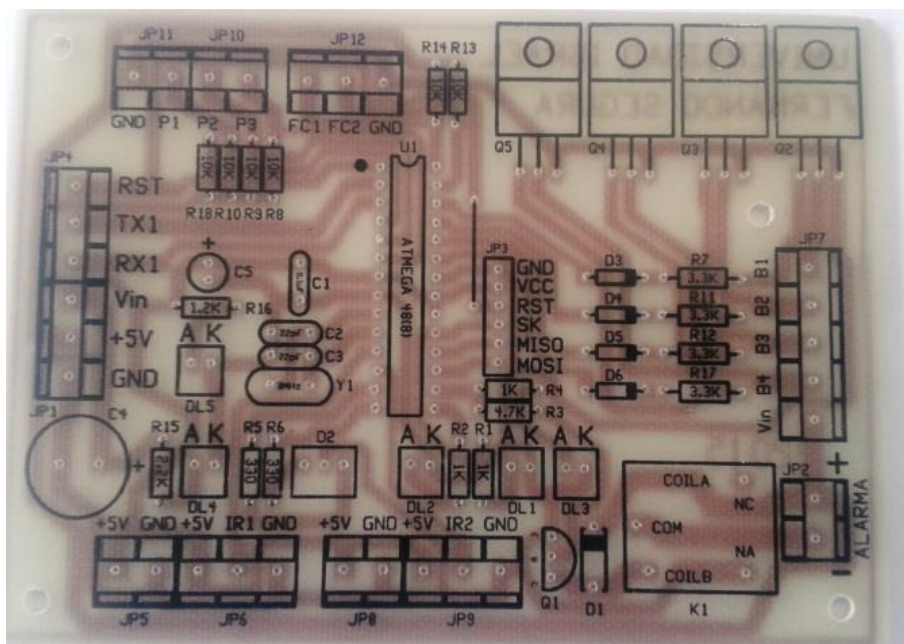
Fuente: (Segura, 2015)



**Figura 3.9.1.6 PCB terminada**

Fuente: (Segura, 2015)

- ✓ Se imprime la distribución de elementos en la baquelita del lado plástico para lograr un mejor acabado.
- ✓ Se realiza las perforaciones para cada elemento.

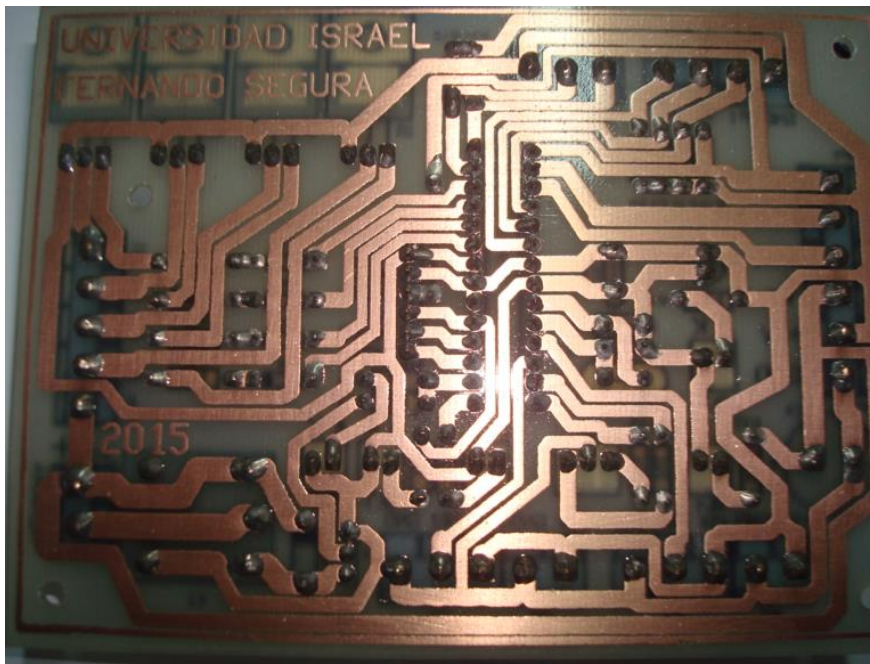


**Figura 3.9.1.7 Perforación en placa y distribución de elementos**

Fuente: (Segura, 2015)



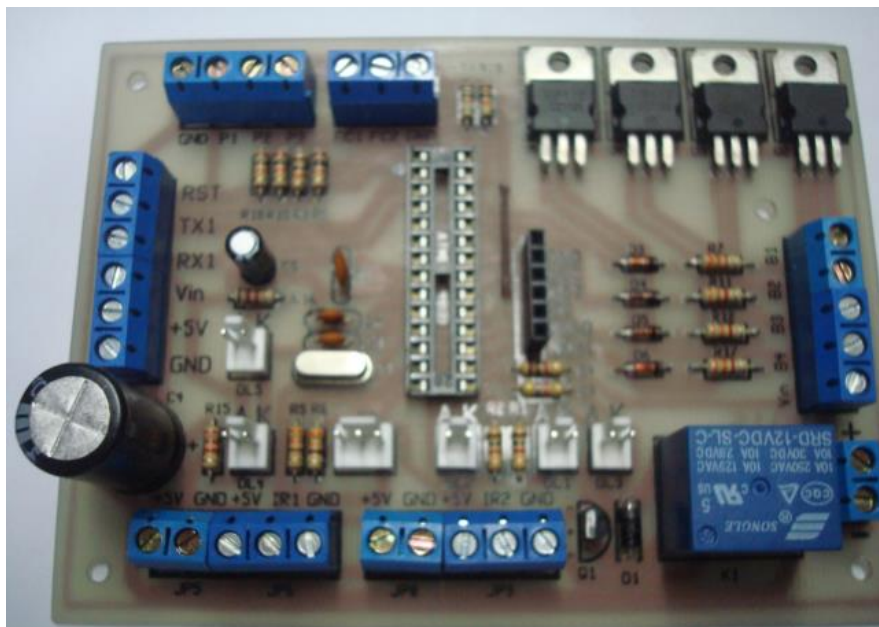
- ✓ Se suelda cada uno de los elementos a la tarjeta.



**Figura 3.9.1.8 Vista posterior de la tarjeta Terminada**

Fuente: (Segura, 2015)

- ✓ Tarjeta Terminada

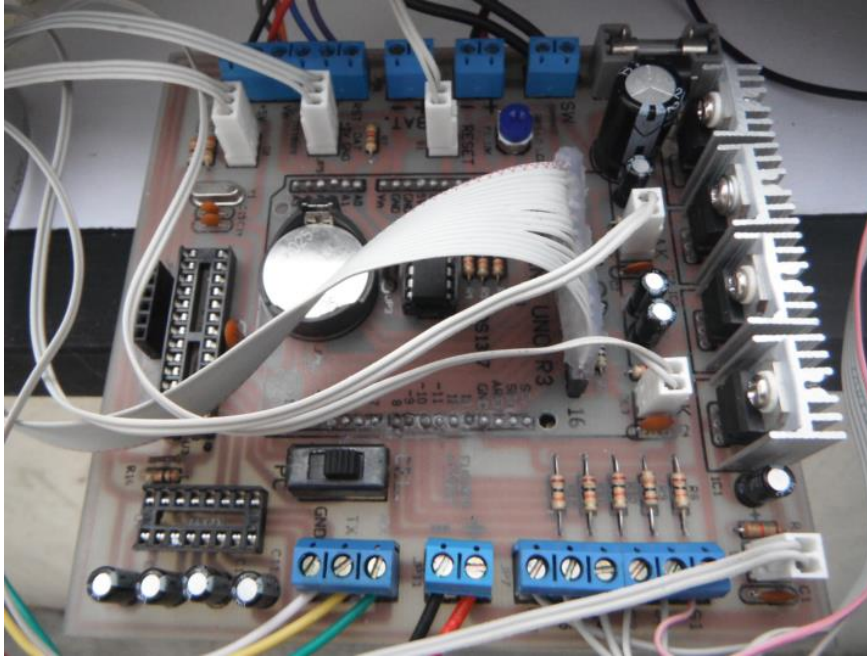


**Figura 3.9.1.9 Vista Frontal Tarjeta Terminada**

Fuente: (Segura, 2015)

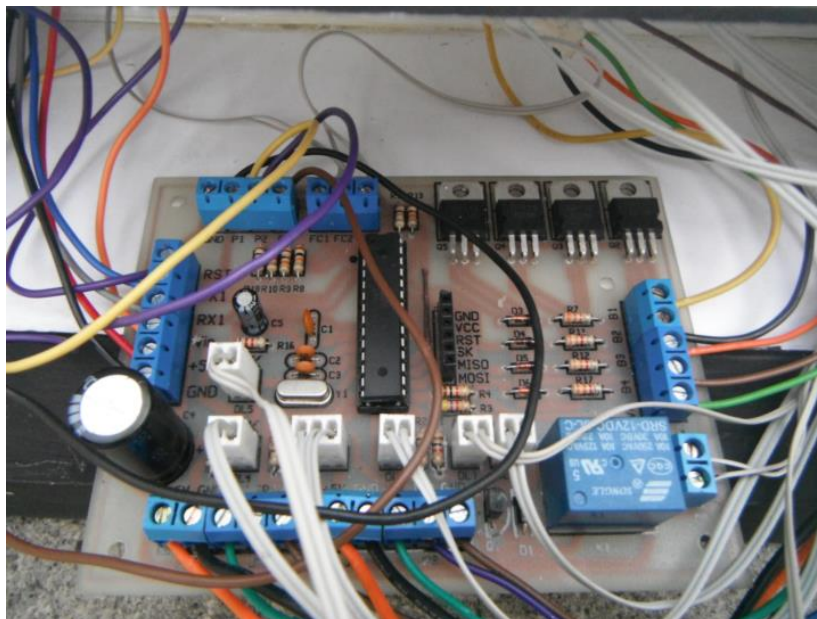
### 3.10. Interconexión de Tarjetas e Instalación en Maqueta

Para las conexiones de las tarjetas y elementos que van fuera de la tarjeta como leds, Switch y pulsadores se utilizó cable eléctrico y principalmente cable de bus de datos.



**Figura 3.10.1 Interconexión Tarjeta de Control Arduino**

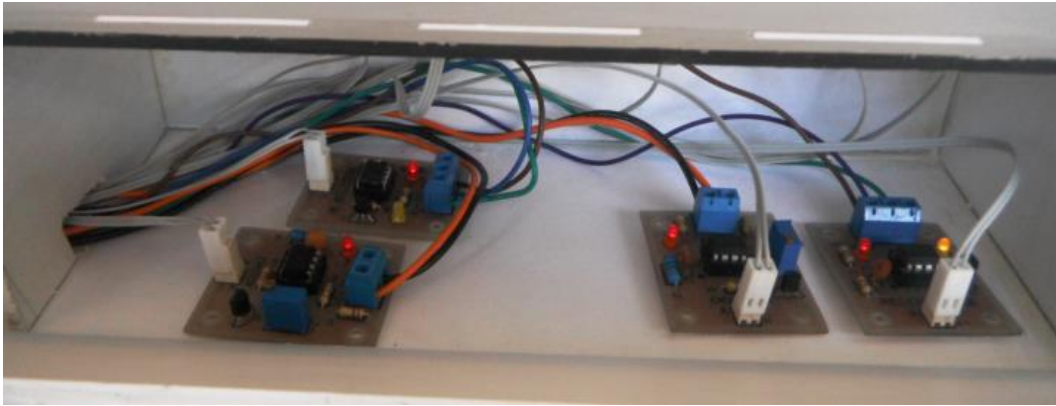
Fuente: (Segura, 2015)



**Figura 3.10.2 Interconexión Tarjeta Control PaP**

Fuente: (Segura, 2015)

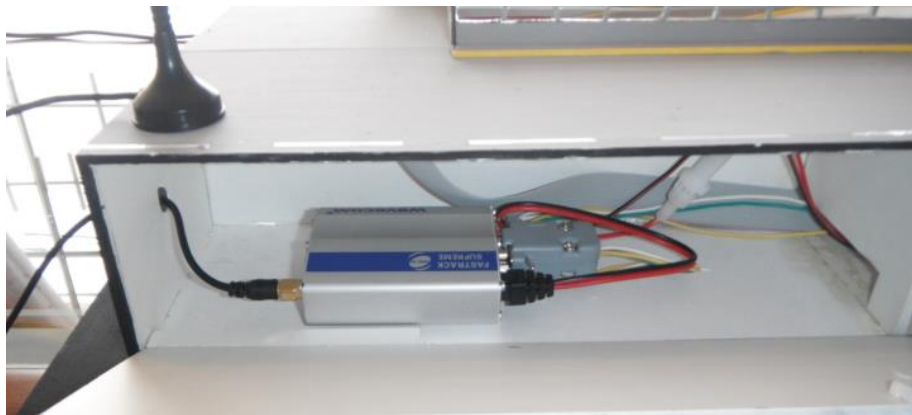
La figura 3.10.3 muestra las conexiones de las tarjetas del emisor y receptor de los sensores infrarrojo.



**Figura 3.10.3 Interconexión Tarjetas Emisor – Receptor IR**

Fuente: (Segura, 2015)

La figura 3.10.4 muestra la ubicación y conexión del modem celular.



**Figura 3.10.4 Interconexión Modem Celular**

Fuente: (Segura, 2015)

### **3.11. LEDS Indicadores del Estatus del Sistema**

El prototipo dispone de un conjunto de leds que indican el estatus del sistema, principalmente la presencia de los diferentes voltajes que maneja cada uno de los circuitos y componentes, además del estatus del modo de operación de la barrera.

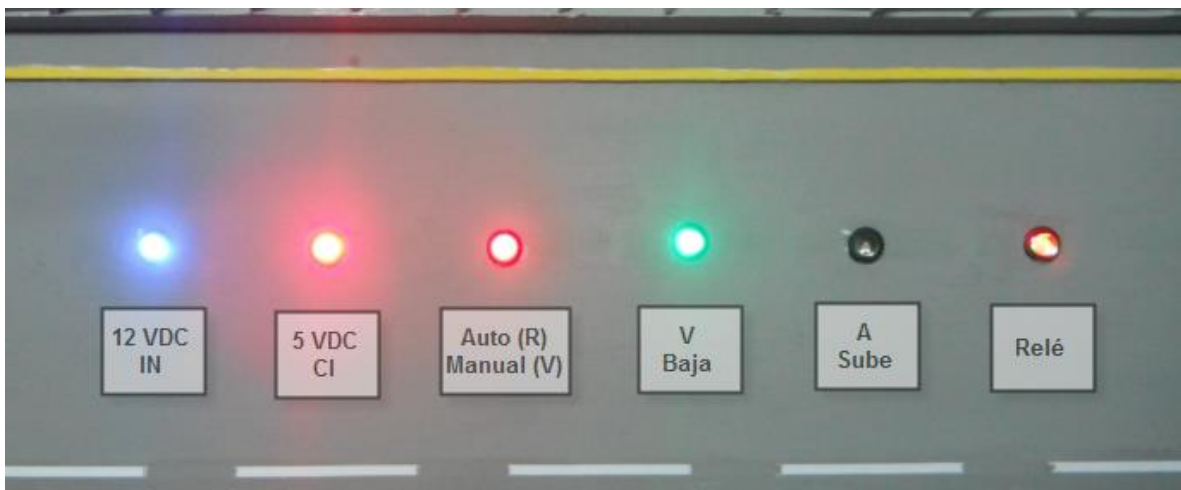




**Figura 3.11.1 Indicadores de estatus del sistema**

Fuente: (Segura, 2015)

La figura 3.11.2 muestra los leds que representan el estatus del sistema en lo que corresponde al control de la barra que impide el ingreso al paso a desnivel, estos leds sirven para determinar de forma inmediata la forma en que está trabajando el sistema y sus componentes.

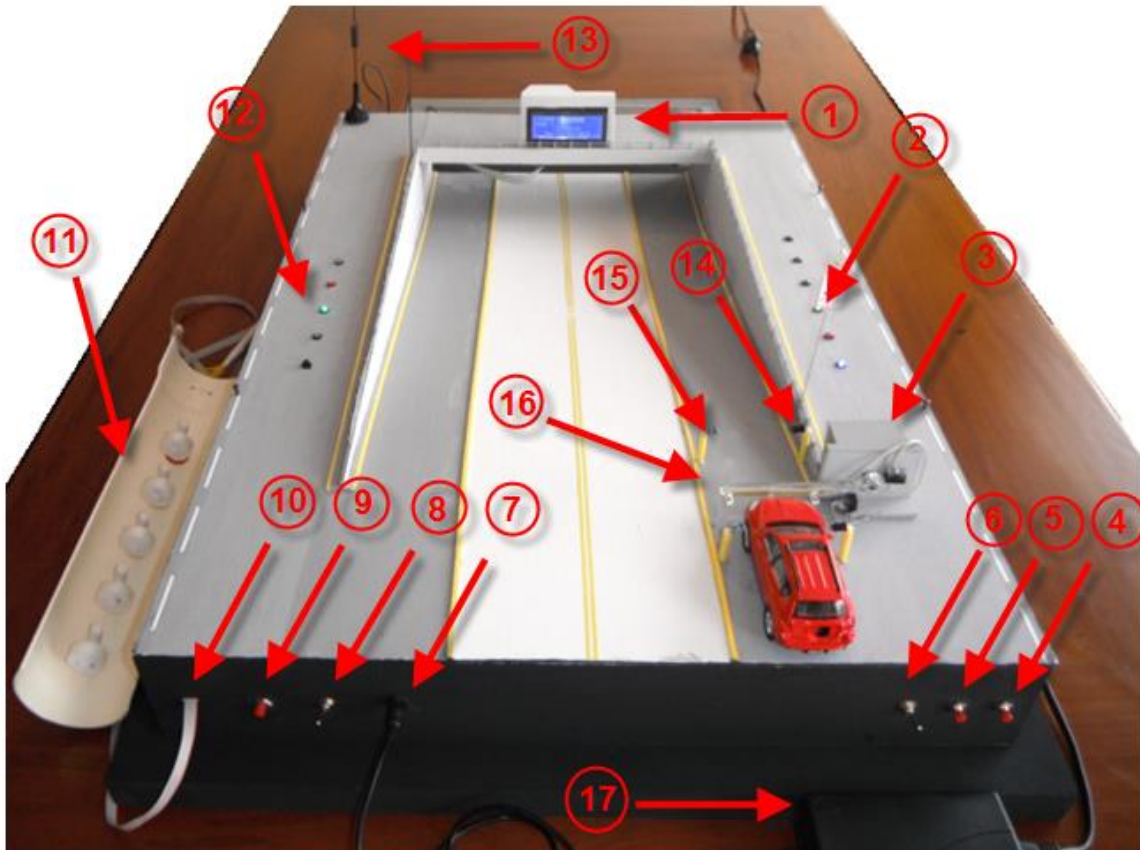


**Figura 3.11.2 Indicadores de estatus de la barrera**

Fuente: (Segura, 2015)

### 3.12. Prototipo Final

La figura 3.12.1 muestra el prototipo final terminado con la descripción de sus partes externas.



**Figura 3.12.1 Producto Terminado**

Fuente: (Segura, 2015)

### 3.12.1. Descripción de partes:

1. Display (pantalla)
2. Leds indicadores de estatus tarjeta de control motor PaP (barrera)
3. Motor PaP (barrera).
4. Pulsador para bajar barrera en operación manual.
5. Pulsador para subir barrera en operación manual.
6. Selector modo manual / automático.
7. Plug para alimentación eléctrica.
8. Switch encendido / apagado.
9. Pulsador para resetear el sistema.
10. Conexión para los switchs de nivel.
11. Cinco Switch de nivel
12. Leds indicadores de estatus tarjeta de control Arduino.

- 13. Antena del modem celular
- 14. Receptor IR
- 15. Emisor IR
- 16. Barrera
- 17. Fuente

### 3.13. Evaluación Técnica del Sistema

Se realizó la evaluación técnica en base a la tabla 3.1, en donde se verifica el cumplimiento de las especificaciones generales y específicas previamente definidas para el sistema:

**Tabla 3.1 Parámetros de Evaluación Técnica**

Parámetro	Cumple (SI/NO)	Observaciones
Sistema Automático	SI	Se puede escoger entre operación manual o automática
Detecta 5 niveles de agua	SI	El quinto nivel es el crítico
Se acciona la barrera en el nivel crítico	SI	
Se muestra en la pantalla los niveles	SI	Todos los niveles
Muestra hora y fecha en pantalla	SI	
Muestra temperatura en la pantalla	SI	Temperatura ambiente
Los sensores de presencia impiden que la barrera baje.	SI	Ante la presencia de un vehículo
Se presentan mensajes de manejo prudente en la pantalla	SI	Mensajes cortos cada cierto tiempo
Se pueden enviar mensajes SMS para ser presentados en la pantalla	SI	Mensajes cortos de hasta 20 caracteres
Se puede accionar la barrera remotamente mediante un mensaje SMS	SI	Únicamente con una instrucción específica

Fuente: (Segura, 2015)

### 3.14. Pruebas de Funcionamiento

Las pruebas de funcionamiento del sistema se realizaron en base a sus modos de operación, es decir, se probó el funcionamiento en modo automático, remoto y manual, ejecutando y verificando el funcionamiento de cada una de sus partes y características del sistema.

Las pruebas realizadas se presentan en las tablas 3.2, 3.3 y 3.4

**Tabla 3.2 Pruebas de Funcionamiento en modo Automático**

<b>Funcionamiento Automático</b>					
Nivel de Agua	Detecta Agua	Presenta nivel en el Display	Acciona barrera (Baja Barrera)	Acciona Luz de Advertencia	Envía mensaje
10 cm	SI	SI	NO	NO	-
20 cm	SI	SI	NO	NO	-
30 cm	SI	SI	NO	NO	-
40 cm	SI	SI	NO	NO	-
50 cm	SI	SI	SI	SI	SI

(Segura, 2015)

**Tabla 3.3 Pruebas de Funcionamiento en modo Remoto**

<b>Funcionamiento Remoto</b>					
Dispositivo Remoto	Envía - Recibe SMS de nivel Crítico	Recibe - Envía SMS para ser presentado en el Display	Sube Barrera con instrucción SMS	Baja Barrera con instrucción SMS	Acciona Luz de Advertencia
Modem Celular	SI	SI	SI	SI	SI
Teléfono Celular	SI	SI	SI	SI	SI

(Segura, 2015)

**Tabla 3.4 Pruebas de Funcionamiento en modo Manual**

<b>Funcionamiento Manual</b>						
Dispositivo	Prende el Sistema	Reinicia los Microcontroladores	Cambia entre operación Manual y Automático	Sube Barrera	Baja Barrera	Impide que baje la barrera
Switch On/Off	SI	-	-	-	-	-
Pulsador Reset	-	SI	-	-	-	-
Switch M / A	-	-	SI	-	-	-
Pulsador 1	-	-	-	SI	-	-
Pulsador 2	-	-	-	-	SI	-
Sensor IR	-	-	-	-	-	SI

(Segura, 2015)

### 3.15. Análisis de Resultados

#### 3.15.1. Análisis de Resultados Tabla 3.2

La tabla 3.2 muestra las pruebas realizadas del sistema operando en modo automático, en ésta se evidencia que el sistema mide los cinco niveles de agua mostrando cada uno de ellos en el Display (Pantalla), únicamente al quinto nivel considerado como crítico acciona la barrera, activa la luz de advertencia y envía el mensaje de tipo SMS al número del celular definido en la programación.

### **3.15.2. Análisis de Resultados Tabla 3.3**

La tabla 3.3 muestra las pruebas realizadas del sistema operando en modo remoto, estas pruebas consistieron en probar las funciones previamente definidas para el modem y el teléfono celular, comprobando las siguientes funciones:

- El modem celular envía el mensaje “Nivel de agua CRÍTICO” cuando el Switch del quinto nivel se acciona ante la presencia de agua.
- El modem celular recibe un mensaje SMS enviado por el teléfono celular y lo presenta en el Display
- El modem celular recibe el mensaje “Barrera Subir”, la instrucción es interpretada por el microcontrolador quien sube la barrera y acciona la luz de advertencia.
- El modem celular recibe el mensaje “Barrera Bajar”, la instrucción es interpretada por el microcontrolador quien baja la barrera y acciona la luz de advertencia.

### **3.15.3. Análisis de Resultados Tabla 3.4**

La tabla 3.4 muestra las pruebas realizadas del sistema operando en modo manual, estas pruebas consistieron en probar las funciones previamente definidas para el sistema cuando es operado por una persona independientemente del estatus de los switch de nivel y las instrucciones del modem o el teléfono celular, comprobando las siguientes funciones:

- El switch de encendido – apagado, prende y apaga el sistema.
- El pulsador para resetear reinicia los microcontroladores y el proceso.
- El switch M/A cambia el modo de operación entre manual y automático.
- Se sube la barrera mediante el pulsador “Subir Barrera”
- Se baja la barrera mediante el pulsador “Bajar Barrera”
- Los sensores infrarrojos impiden que la barrera baje cuando detecta la presencia de un vehículo.



## **CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- Es posible mejorar la seguridad vehicular en pasos a desnivel o deprimidos con sistemas electrónicos, es así que el presente proyecto constituye una solución que reemplaza procedimientos o sistemas manuales mejorando el bienestar de los usuarios.
- Es necesario dimensionar la carga y determinar la fuente con la capacidad suficiente para suministrar la corriente que demandan las tarjetas, el motor PaP y el modem celular, para este sistema la corriente máxima es 1.5 Amperios.
- Para que el modem se enlace con la red celular GSM es necesario que tenga una intensidad de señal entre 11 y 31, este nivel se determina mediante comandos AT conectándose al celular de forma serial con el Hyperterminal.
- Para garantizar la comunicación entre el microcontrolador y el modem celular se escoge los pines de comunicación serial que vienen definidos en el Hardware del microcontrolador AVR 8, para la comunicación con el Arduino se puede configurar mediante software otros pines.
- Ante una falla en el software del sistema es necesario que se reinicien los tres microcontroladores al mismo tiempo mediante el pulsador reset para garantizar el correcto funcionamiento.

### **4.2. Recomendaciones**

- El modem celular depende de la red de la operadora y esta puede ser una causa de lentitud en el procesamiento que no se puede evitar, por lo que se sugiere utilizar un microcontrolador exclusivo para el manejo del modem y reducir de esta manera el tiempo en el procesamiento de los datos.

- Se sugiere que el número celular al que deben llegar los mensajes de advertencia de nivel crítico de agua sea a un número definido por el Servicio Integrado de Seguridad ECU 911 en Quito.
- Realizar una regulación de voltaje progresiva para la alimentación de los circuitos, de esta manera se evita el sobrecalentamiento.
- Realizar el respaldo de los programas cargados a los microcontroladores.
- Etiquetar los puntos de manipulación del usuario.

## 1. BIBLIOGRAFÍA

- ANDES Cotopaxinoticias.com. (09 de octubre de 2012). *Cotopaxi noticias.com*. Obtenido de Cotopaxi noticias.com:  
<http://www.cotopaxinoticias.com/seccion.aspx?sid=13&nid=9409>
- Arduino. (1 de Noviembre de 2014). *ARDUINO*.
- Atmel. (julio de 2010). *Atmel AVR 8-bit and 32-bit Microcontrollers*. Recuperado el 10 de junio de 2014, de <http://www.atmel.com/Images/doc2466.pdf>
- Bayle, J. (2013). *C Programming for Arduino*. Birmingham: Packt Publishing Ltd.
- Braza, C. (05 de Enero de 2015). *Quadruino.COM*. Obtenido de <http://www.quadruino.com/guia-2/sensores/protocolo-i2c-twi>
- Corporation, A. (1 de enero de 2015). *Atmel*. Obtenido de Atmel:  
<http://www.atmel.com/products/microcontrollers/avr/default.aspx?tab=overview>
- Creus, A. (2011). *INSTRUMENTACIÓN INDUSTRIAL*. México D.F.: Alfaomega .
- Fundación Wikimedia, Inc.,. (11 de MAR de 2013). *WIKIPEDIA*. Recuperado el 01 de junio de 2014, de <http://es.wikipedia.org/wiki/MAX232>
- Harth, S. L. (2008). *MICROCONTROLADORES PIC PRÁCTICAS DE PROGRAMACIÓN*. Barcelona: Marcombo.
- Karolina. (19 de febrero de 2009). *Karolina Blogspot*. Obtenido de <http://karolina-quito.blogspot.com/2009/02/transporte-en-quito.html>
- Robótica. (01 de abril de 2012). *tiendaderobotica*. Obtenido de [www.tiendarobotica.com](http://www.tiendarobotica.com)
- Rojas, A. A. (julio de 2008). *Universidad de Costa Rica*. Recuperado el 25 de abril de 2014, de [http://eie.ucr.ac.cr/uploads/file/proybach/pb2009/pb2009\\_022.pdf](http://eie.ucr.ac.cr/uploads/file/proybach/pb2009/pb2009_022.pdf)
- SA, P. F. (2 de Diciembre de 2015). *Punto Flotante*. Obtenido de <http://www.puntofotante.net/MODEM-WAVECOM.htm>
- Semiconductor, D. (01 de eNERO de 2006). *ALLDATASHEET*. Obtenido de <http://www.alldatasheet.com>
- VisitaQuito. (01 de mayo de 2014). *VisitaQuito*. Obtenido de VisitaQuito: <http://visitaquito.net/>
- WAVECOM. (11 de Noviembre de 2006). *manualslib*. Obtenido de <http://www.manualslib.com/manual/530980/Wavecom-Fastrack-M1306b.html>

**ANEXOS**

## ANEXO 1

### Costos del Prototipo

Tarjeta de Control Arduino					
Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	4	C1,C4,C7,C10	0.1uF	\$ 0,15	\$ 0,60
2	1	C2	2200uF	\$ 0,40	\$ 0,40
3	8	C3,C5,C6,C8,C9,C11,C12,C13	10uF	\$ 0,15	\$ 1,20
4	2	C15,C14	22pF	\$ 0,15	\$ 0,30
5	1	DB1	CONNECTOR DB9	\$ 0,40	\$ 0,40
6	5	DL1,DL2,D2,DL3,DL4	LED	\$ 0,30	\$ 1,50
7	1	D1	1N5406	\$ 0,30	\$ 0,30
8	1	F1	FUSE	\$ 1,00	\$ 1,00
9	1	IC1	7810	\$ 0,80	\$ 0,80
10	1	IC2	7809	\$ 0,80	\$ 0,80
11	1	IC3	7805	\$ 0,80	\$ 0,80
12	1	IC4	7808	\$ 0,80	\$ 0,80
13	1	JP1	Arduino1R3	\$ 2,00	\$ 2,00
14	1	JP2	HEADER2	\$ 0,30	\$ 0,30
15	1	JP3	Reloj	\$ 1,00	\$ 1,00
16	1	JP4	Batería	\$ 1,00	\$ 1,00
17	5	JP5,JP6,JP7,JP8,JP10	HEADER 3	\$ 0,40	\$ 2,00
18	1	JP9	HEADER 6	\$ 0,50	\$ 0,50
19	1	JP11	HEADER 2	\$ 0,30	\$ 0,30
20	1	J1	RCA JACK	\$ 0,30	\$ 0,30
21	8	R5,R6,R8,R9,R10,R11,R12,R16	10K	\$ 0,05	\$ 0,40
22	1	P1	10K	\$ 0,70	\$ 0,70
23	1	R1	1.5K	\$ 0,05	\$ 0,05
24	1	R2	10	\$ 0,05	\$ 0,05
25	2	R13,R3	1.2K	\$ 0,05	\$ 0,10
26	1	R4	1K	\$ 0,05	\$ 0,05
27	1	R7	4.7K	\$ 0,05	\$ 0,05
28	2	R14,R15	330	\$ 0,05	\$ 0,10
29	1	S1	SW PUSHBUTTON	\$ 0,30	\$ 0,30
30	1	S2	SW SPDT	\$ 0,30	\$ 0,30
31	1	S3	SW DPDT	\$ 0,60	\$ 0,60
32	1	U1	LNT-211	\$ 0,80	\$ 0,80
33	1	U2	DS1307	\$ 0,20	\$ 0,20
34	1	U3	ATMEGA 8	\$ 0,40	\$ 0,40
35	1	U4	MAX232	\$ 0,30	\$ 0,30
36	1	Y1	8MHz	\$ 1,00	\$ 1,00
37	1	Arduino	UNO	\$ 30,00	\$ 30,00
38	1	Ds1307		\$ 4,00	\$ 4,00
39	1	Batería 2025		\$ 1,50	\$ 1,50
40	1	Atmega8		\$ 6,00	\$ 6,00
41	1	Max232		\$ 3,50	\$ 3,50
42	1	Ds18b20		\$ 4,00	\$ 4,00
43	1	Fusible 5A.		\$ 0,20	\$ 0,20

44	1	SW ON/OFF		\$ 0,60	\$ 0,60
45	1	SW reset		\$ 0,60	\$ 0,60
46	5	Led's		\$ 0,25	\$ 1,25
47	1	Led bicolor		\$ 0,30	\$ 0,30
48	4	Disipadores		\$ 1,00	\$ 4,00
49	1	LCD 20*4		\$ 20,00	\$ 20,00
50	1	Modem GSM		\$134,40	\$ 134,40
51	2	Cable IDE 20		\$ 5,00	\$ 10,00
52	2	Cable 10 hilos		\$ 2,50	\$ 5,00
53	1	Cable DB15		\$ 3,00	\$ 3,00
54	5	Switch de Nivel	SW NC/ NO	\$ 5,90	\$ 29,50
<b>Subtotal (1)</b>					<b>\$ 279,55</b>

### Tarjeta de Control Motor PaP

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	1	C1	0.1uF	\$ 0,15	\$ 0,15
2	2	C3,C2	22pF	\$ 0,15	\$ 0,30
3	1	C4	1000uF	\$ 0,40	\$ 0,40
4	1	C5	10uF	\$ 0,15	\$ 0,15
5	6	DL1,DL2,D2,DL3,DL4,DL5	LED	\$ 0,30	\$ 1,80
6	1	D1	1N4007	\$ 0,20	\$ 0,20
7	4	D3,D4,D5,D6	1N4148	\$ 0,20	\$ 0,80
8	5	JP1,JP4,JP6,JP9,JP12	HEADER 3	\$ 0,40	\$ 2,00
9	5	JP2,JP5,JP8,JP10,JP11	HEADER 2	\$ 0,30	\$ 1,50
10	1	JP3	HEADER 6	\$ 0,50	\$ 0,50
11	1	JP7	HEADER 5	\$ 0,70	\$ 0,70
12	1	K1	RELAY 12VDC	\$ 1,50	\$ 1,50
13	1	Q1	2N3904	\$ 0,15	\$ 0,15
14	4	Q2,Q3,Q4,Q5	NPN DAR BCE	\$ 0,80	\$ 3,20
15	3	R1,R2,R4	1K	\$ 0,05	\$ 0,15
16	1	R3	4.7K	\$ 0,05	\$ 0,05
17	2	R5,R6	330	\$ 0,05	\$ 0,10
18	4	R7,R11,R12,R17	3.3K	\$ 0,05	\$ 0,20
19	6	R8,R9,R10,R13,R14,R18	10K	\$ 0,05	\$ 0,30
20	1	R15	2.2K	\$ 0,05	\$ 0,05
21	1	R16	1.2K	\$ 0,05	\$ 0,05
22	1	U1	ATMEGA 8	\$ 0,40	\$ 0,40
23	1	Y1	8MHz	\$ 1,00	\$ 1,00
24	1	Atmega8		\$ 6,00	\$ 6,00
25	1	SW MAN/AUTO		\$ 0,60	\$ 0,60
26	2	Pulsadores		\$ 0,60	\$ 1,20
27	5	Led's		\$ 0,25	\$ 1,25
28	1	Led bicolor		\$ 0,30	\$ 0,30
<b>Subtotal (2)</b>					<b>\$ 25,00</b>

### Tarjeta Emisor IR

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	4	D2,D1	LED	\$ 0,30	\$ 1,20
2	2	JP1	HEADER 2	\$ 0,30	\$ 0,60
3	2	P1	20K	\$ 0,70	\$ 1,40
4	2	Q1	2N3904	\$ 0,15	\$ 0,30
5	2	R1	10	\$ 0,05	\$ 0,10
6	2	R2	100K	\$ 0,05	\$ 0,10
7	2	R3	15K	\$ 0,05	\$ 0,10
8	2	R4	4.7K	\$ 0,05	\$ 0,10
9	2	R5	680	\$ 0,05	\$ 0,10
10	2	U1	LM555	\$ 0,20	\$ 0,40
11	2	Diodo emisor IR		\$ 0,50	\$ 1,00
12	2	LM555		\$ 0,50	\$ 1,00
13	2	Led's		\$ 0,15	\$ 0,30
<b>Subtotal (3)</b>					<b>\$ 6,70</b>

### Tarjeta Receptor IR

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	2	C1	0.1uF	\$ 0,15	\$ 0,30
2	4	D1,D2		\$ 0,15	\$ 0,60
3	2	DL1	LED	\$ 0,30	\$ 0,60
4	2	JP1	HEADER 3	\$ 0,40	\$ 0,80
5	2	Q1	2N3904	\$ 0,15	\$ 0,30
6	4	R4,R1	470	\$ 0,05	\$ 0,20
7	4	R2,R3	10K	\$ 0,05	\$ 0,20
8	2	U1	LM358	\$ 0,20	\$ 0,40
9	2	Diodo receptor IR		\$ 1,00	\$ 2,00
10	2	LM358		\$ 0,50	\$ 1,00
<b>Subtotal (4)</b>					<b>\$ 6,40</b>

### Maqueta Puente a Desnivel

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	1	Maqueta Puente	N/A	\$150,00	\$ 150,00
2	1	Consumibles	N/A	\$ 10,00	\$ 10,00
<b>Subtotal (5)</b>					<b>\$ 160,00</b>

### Horas Hombre

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	15,189	Diseño	N/A	\$ 5,00	\$ 75,95
2	17,189	Programación	N/A	\$ 5,00	\$ 85,95
3	12,189	Implementación	N/A	\$ 5,00	\$ 60,95
<b>Subtotal (6)</b>					<b>\$ 222,84</b>

**TOTAL**

**\$ 700,49**

### Costo Sistema Real

Ítem	Cantidad	Referencia	Parte	Valor Unitario	Subtotal
1	1	Etapa de Control	N/A	\$ 466,45	\$ 466,45
2	1	Barrera Automática (Instalada)	FAAC 615 EST.	\$ 2.830,41	\$ 2.830,41
3	1	Pantalla led 6 x 1.5		\$10.000,00	\$ 10.000,00

<b>Total</b>	<b>\$ 13.296,86</b>
--------------	---------------------

(Segura, 2015)



## ANEXO 2

### PROGRAMACIÓN MICROCONTROLADORES

#### ARDUINO

```
#include <OneWire.h>
#include <DS1307.h>
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <Wire.h>

#define RXPIN A3
#define TXPIN A2
#define MOTORBAUD 9600
SoftwareSerial uart_motor(RXPIN, TXPIN); // Comunicacion por Software con el MOTOR
PAP.

void setup();
void loop();
void Print(int number);
void DiaSemana(int number);
void RevisarSensores();
void RevisarSensores1();
void printTime(void);
void getTemp();
void printTemp(void);
LiquidCrystal lcd(2,3,4,5,6,7); //Configura el LCD
int clock[7];
int Sensor1 = 12;    // Sensor1
int Sensor2 = 11;    // Sensor2
int Sensor3 = 10;    // Sensor3
int Sensor4 = 9;     // Sensor4
int Sensor5 = 8;     // Sensor5
//Crea nuevos caracteres para el LCD
byte caracter0[8] = {31,17,21,21,21,17,31,31};
byte caracter1[8] = {31,27,19,27,27,17,31,31};
byte caracter2[8] = {31,17,21,29,27,17,31,31};
byte caracter3[8] = {31,17,29,27,29,17,31,31};
byte caracter4[8] = {31,29,25,21,16,29,31,31};
byte caracter5[8] = {31,17,23,25,29,17,31,31};
byte caracter6[8] = {31,25,23,17,21,17,31,31};
byte caracter7[8] = {14,10,14,32,32,32,32,32};

boolean EstadoSensor1;
boolean EstadoSensor2;
boolean EstadoSensor3;
```

```

boolean EstadoSensor4;
boolean EstadoSensor5;
OneWire ds(A1);    // ds18b20 pin A1
byte i;
byte j=0;
byte k=0;
byte l=0;
byte m=0;
int x=0;
byte present = 0;
byte data[12];
byte addr[8];
byte nivel = 0;
char cadena[30];  //se crea un array que almacenará los caracteres que se escribe en la
LCD
byte posicion=0;  //Variable para cambiar la posición de los caracteres del array
int HighByte, LowByte, SignBit, Whole, Fract, TReading, Tc_100, FWhole;
//-----
void setup()
{
  uart_motor.begin(MOTORBAUD); //Inicilaiza comunicación por Software con el MOTOR
  PAP
  Serial.begin(9600);          //Inicilaiza comunicación por Hardware para comunicarse con el
  micro del Celular
  for(int i=2;i<8;i++)
  {
    pinMode(i,OUTPUT);
  }
  for(int i=8;i<13;i++)
  {
    pinMode(i,INPUT);
  }
  digitalWrite(Sensor1,HIGH);
  digitalWrite(Sensor2,HIGH);
  digitalWrite(Sensor3,HIGH);
  digitalWrite(Sensor4,HIGH);
  digitalWrite(Sensor5,HIGH);

//Se crean los caracteres (simbolos)
  lcd.createChar(0,caracter0);
  lcd.createChar(1,caracter1);
  lcd.createChar(2,caracter2);
  lcd.createChar(3,caracter3);
  lcd.createChar(4,caracter4);
  lcd.createChar(5,caracter5);
  lcd.createChar(6,caracter6);
  lcd.createChar(7,caracter7);

```

```

    lcd.begin(20,4);
    lcd.clear();
    DS1307.begin();    //Inicilaiza comunicacion con el DS1307
//
DS1307.setDate(15,2,25,3,8,33,00);//a\u00f1o,mes,d\u00eda,d\u00edasemana,horas,minuto
s,segundos
//Inicilaiza comunicacion con el DS18B20
    if ( !ds.search(addr))
    {
        lcd.clear(); lcd.print("No more addr");
        delay(1000);
        ds.reset_search();
        return;
    }
    if ( OneWire::crc8( addr, 7) != addr[7]) {
        lcd.clear(); lcd.print("CRC not valid!");
        delay(1000);
        return;
    }
    lcdMensaje();    //Elige cual mensaje va a imprimirse en el LCD
}
//-----
void loop()
{
    if (Serial.available())    //Recibe datos del micro Celular
    {
        while(Serial.available()>0) //Mientras haya datos en el buffer ejecuta la función
        {
            delay(5);    //Poner un pequeño delay para mejorar la recepción de datos
            cadena[posicion]=Serial.read();//Lee un carácter del string "cadena" de la "posicion",
luego lee el siguiente carácter con "posicion++"
            posicion++;
        }
        posicion=0;
        x=0;
        if (cadena[0]=='A')    //Si mensaje recibido es "A" sube la barrera
        {
            uart_motor.print("A");
            m=1;
        }
        if (cadena[0]=='B')    //Si mensaje recibido es "B" baja la barrera
        {
            uart_motor.print("B");
            m=1;
        }
        if (m==0)    //En otro caso imprime en el LCD el mensaje
        {
            lcd.setCursor(0,0);

```

```

    lcd.print("          ");
    lcd.setCursor(0,0);
    lcd.print(cadena);
    posicion=0;
    x=0;
}
m=0;
for (int i=0; i <= 25; i++)
{
    cadena[i]=' ';
}
}
lcd.setCursor(0,1);
getTemp();
printTemp();          //Imprime en el LCD la Temperatura
lcd.setCursor(9,1);
RevisarSensores1();    //Imprime en el LCD la medida de los sensores de nivel
DS1307.getDate(clock); //Imprime en el LCD la Hora y la Fecha
lcd.setCursor(0,2);
printTime();
x=x+1;
if (x >=250)
{
    x=0;
    k=k+1;
    lcdMensaje();      //Elige cual mensaje va a imprimirse en el LCD
}
}
//-----
//Funcion para para ajustar los digitos a: 00,01,02,...
void Print(int number)
{
    lcd.print(number/10);//imprime o digito mais significativo
    lcd.print(number%10);//imprime o digito menos significativo
}
//-----
//Funcion para día de la semana
void DiaSemana(int number)
{
    switch (number)
    {
        case 0:
            lcd.print("Dom-");
            break;
        case 1:
            lcd.print("Lun-");
            break;
        case 2:

```

```

    lcd.print("Mar-");
    break;
case 3:
    lcd.print("Mie-");
    break;
case 4:
    lcd.print("Jue-");
    break;
case 5:
    lcd.print("Vie-");
    break;
case 6:
    lcd.print("Sab-");
    break;
case 7:
    lcd.print("Dom-");
    break;
}
}
//-----
void RevisarSensores()
{
    EstadoSensor1=digitalRead(Sensor1);
    if (EstadoSensor1==LOW)
    {
        lcd.setCursor(1, 0); // Se posiciona en el primer caracter de la primer línea 0,0
        lcd.print((char)1); // Se escribe en la primer línea del display
    }
    else
    {
        lcd.setCursor(1, 0); // Se posiciona en el primer caracter de la primer línea 0,0
        lcd.print("1"); // Se escribe en la primer línea del display
    }
    EstadoSensor2=digitalRead(Sensor2);
    if (EstadoSensor2==LOW)
    {
        lcd.setCursor(2, 0); // se posiciona en el primer caracter de la primer línea 0,0
        lcd.print((char)2); // Se escribe en la primer línea del display
    }
    else
    {
        lcd.setCursor(2, 0); // se posiciona en el primer caracter de la primer línea 0,0
        lcd.print("2"); // Se escribe en la primer línea del display
    }
    EstadoSensor3=digitalRead(Sensor3);
    if (EstadoSensor3==LOW)
    {
        lcd.setCursor(3, 0); // se posiciona en el primer caracter de la primer línea 0,0

```

```

    lcd.print((char)3); // Se escribe en la primer línea del display
}
else
{
    lcd.setCursor(3, 0); // se posiciona en el primer caracter de la primer línea 0,0
    lcd.print("3"); // Se escribe en la primer línea del display
}
EstadoSensor4=digitalRead(Sensor4);
if (EstadoSensor4==LOW)
{
    lcd.setCursor(4, 0); // se posiciona en el primer caracter de la primer línea 0,0
    lcd.print((char)4); // Se escribe en la primer línea del display
}
else
{
    lcd.setCursor(4, 0); // se posiciona en el primer caracter de la primer línea 0,0
    lcd.print("4"); // Se escribe en la primer línea del display
}
EstadoSensor5=digitalRead(Sensor5);
if (EstadoSensor5==LOW)
{
    lcd.setCursor(5, 0); // se posiciona en el primer caracter de la primer línea 0,0
    lcd.print((char)5); // Se escribe en la primer línea del display
}
else
{
    lcd.setCursor(5, 0); // se posiciona en el primer caracter de la primer línea 0,0
    lcd.print("5"); // Se escribe en la primer línea del display
}
}
//-----
void RevisarSensores1()
{
    EstadoSensor1=digitalRead(Sensor1);
    if (EstadoSensor1==LOW)
    {
        bitSet(nivel,4);
    }
    else
    {
        bitClear(nivel, 4);
    }
    EstadoSensor2=digitalRead(Sensor2);
    if (EstadoSensor2==LOW)
    {
        bitSet(nivel, 3);
    }
    else

```

```

{
    bitClear(nivel, 3);
}
EstadoSensor3=digitalRead(Sensor3);
if (EstadoSensor3==LOW)
{
    bitSet(nivel, 2);
}
else
{
    bitClear(nivel, 2);
}
EstadoSensor4=digitalRead(Sensor4);
if (EstadoSensor4==LOW)
{
    bitSet(nivel, 1);
}
else
{
    bitClear(nivel, 1);
}
EstadoSensor5=digitalRead(Sensor5);
if (EstadoSensor5==LOW)
{
    bitSet(nivel, 0);
}
else
{
    bitClear(nivel, 0);
}

switch (nivel)
{
    case 0:
        lcd.print("NIVEL:00 cm");
        j=0;
        if (l==1)
        {
            l=0;
            uart_motor.print("A");
        }
        break;
    case 1:
        lcd.print("NIVEL:10 cm");
        j=0;
        if (l==1)
        {
            l=0;

```

```

    uart_motor.print("A");
}
break;
case 3:
    lcd.print("NIVEL:20 cm");
    j=0;
    if (l==1)
    {
        l=0;
        uart_motor.print("A");
    }
    break;
case 7:
    lcd.print("NIVEL:30 cm");
    j=0;
    break;
case 15:
    lcd.print("NIVEL:40 cm");
    j=0;
    break;
case 31:
    lcd.print("NIVEL:50 cm");
    if (j==0)
    {
        j=1;
        l=1;
        uart_motor.print("B");
        Serial.print("1");
        while(Serial.available()==0) //Espera recibir un dato
        {
            delay(5); //Poner un pequeño delay para mejorar la recepción de datos
            Serial.print("1");
        }
        char cadena1=Serial.read();
        Serial.flush();
    }
    break;
}
}
//-----
void printTime(void)
{
    lcd.print(" ");
    Print(clock[4]);
    lcd.print(":");
    Print(clock[5]);
    lcd.print(":");
    Print(clock[6]);
}

```



```

lcd.print(" ");
lcd.setCursor(0,3);
lcd.print(" ");
DiaSemana(clock[3]);
Print(clock[2]);
lcd.print("/");
Print(clock[1]);
lcd.print("/");
lcd.print("20");
lcd.print(clock[0]);
lcd.print(" ");
}
//-----
void getTemp()
{
  int foo, bar;
  ds.reset();
  ds.select(addr);
  ds.write(0x44,1);
  present = ds.reset();
  ds.select(addr);
  ds.write(0xBE);

  for ( i = 0; i < 9; i++)
  {
    data[i] = ds.read();
  }

  LowByte = data[0];
  HighByte = data[1];
  TReading = (HighByte << 8) + LowByte;
  SignBit = TReading & 0x8000; // Prueba el bit mas significativo

  if (SignBit)
  {
    TReading = -TReading;
  }
  Tc_100 = (6 * TReading) + TReading / 4; // multiplica por (100 * 0.0625) or 6.25
  Whole = Tc_100 / 100; // separa las partes de la fraccion
  Fract = Tc_100 % 100;
  if (Fract > 49)
  {
    if (SignBit)
    {
      --Whole;
    } else
    {
      ++Whole;
    }
  }
}

```

```

    }
}

if (SignBit)
{
    bar = -1;
} else
{
    bar = 1;
}
foo = ((Whole * bar) * 18);    // seccion de conversion a Celsius o Fahrenheit
FWwhole = (((Whole * bar) * 18) / 10) + 32;
if ((foo % 10) > 4)
{
    // redondea si es necesario
    ++FWwhole;
}
}
//-----
void printTemp(void)
{
    lcd.setCursor(0,1);
    lcd.print("T:");
    if (SignBit)
    {
        lcd.print("-");
    }
    lcd.print(Whole);
    lcd.print((char)7);
    lcd.print("C");
    lcd.print(" ");
}
//-----
void lcdMensaje(void)
{
    if (k>=3)
    {
        k=0;
    }
    switch (k)
    {
        case 0:
            lcd.setCursor(0,0);
            lcd.print(" VELOCIDAD 50Km/h ");
            break;
        case 1:
            lcd.setCursor(0,0);
            lcd.print("UTILICE EL CINTURON");
            break;
    }
}

```

```

case 2:
    lcd.setCursor(0,0);
    lcd.print(" NO USE EL CELULAR ");
    break;
}
}
//-----

```

## AVR 8 MODEM CELULAR

```

*****
!*      DEFINICION MICROCONTROLADOR, CRISTAL Y VELOCIDAD SERIAL      *
*****

$sim
$regfile = "m8def.dat"                'MICROCONTROLADOR A UTILIZAR
$crystal = 8000000                    'Frecuencia del cristal 8000000hz
$baud = 9600                          'VELOCIDAD COMUNICACION Celular

*****
!*      DECLARACIÓN DE SUBFUNCIONES                                  *
*****

Declare Sub Config_inicial()           'SUBFUNCION PARA CONFIGURACION
INICIAL DEL CELULAR
Declare Sub Getok(s As String)         'SUBFUNCION PARA OBTENER OK
Declare Sub Limpiarbuffer()           'SUBFUNCION PARA LIMPIAR BUFFER
Declare Sub Enviarmensaje(s As String , N As String) 'SUBFUNCION PARA ENVIAR
MENSAJE
Declare Sub Recibirmensaje()          'SUBFUNCION PARA RECIBIR
MENSAJE
Declare Sub Validarmensaje(s As String) 'SUBFUNCION PARA VALIDAR
MENSAJE

*****
!*      LED BICOLOR                                               *
*****

Ddrb.1 = 1
Portb.1 = 0
Ddrb.2 = 1
Portb.2 = 0
Ledrojo Alias Portb.1
Ledverde Alias Portb.2

***** Uart por Software *****
Ddrd.3 = 1

```

```
Portd.3 = 0
Ddrd.2 = 0
Portd.2 = 1
'Now open a pin for output para MICRO CEL.
Open "comd.3:9600,8,n,1" For Output As #2
'Now open a pin for input para MICRO CEL.
Open "comd.2:9600,8,n,1" For Input As #3
```

```
'+++++
```

```
Dim A As Byte
Dim B As Byte
Dim C As Byte
Dim D As Byte
Dim E As Byte
Dim F As Byte
Dim G As Byte
Dim H As Byte
Dim I As Byte
Dim I2 As Byte
Dim V As Byte
Dim W As Word
Dim X As Byte
Dim Zz As Byte
Dim Z1 As Byte
Dim V1 As Byte
Dim Tempora As String * 1
Dim Temporal As String * 1
Dim Temporal1 As String * 1
Dim Bandera0 As Bit
```

```
Dim Flag1 As Byte
Dim Flag2 As Bit
Dim Flag3 As Bit
Dim Flag4 As Bit
Dim Flag5 As Bit
Dim Clave1(10) As String * 1
Dim Clavea1(10) As String * 1
Dim Clavea2(10) As String * 1
Dim Clavea3(10) As Byte
Dim Serial1 As String * 10
```

```
Dim Sret As String * 50
("OK", "ERROR")
Dim Mensa As String * 150
Dim Mensaje As String * 150
Dim Numero1 As String * 12
Dim Numero2 As String * 12
```

```
'GUARDA LA RESPUESTA DEL CELULAR
```

```
'Guarda El Mensaje A Enviar
```

```
'Se Guarda EL MENSAJE RECIBIDO
```

```
Const Tiempo = 120           'Tiempo en seg
Const Tiempo1 = 30          'Tiempo en seg
Const Mil = 1000            '1000
Const Retardo = 3           'Tiempo en seg.
Const Vermensaje = 2        'Tiempo en seg.
```

```
Numero1 = ""
For I = 1 To 10
  Readeeprom B , I
  Temporal = Chr(b)
  Numero1 = Numero1 + Temporal
Next I
```

```
Numero2 = ""
For I = 13 To 22
  Readeeprom B , I
  Temporal = Chr(b)
  Numero2 = Numero2 + Temporal
Next I
```

```
For X = 1 To 5
  Ledrojo = 1
  Ledverde = 0
  Waitms 200
  Ledrojo = 0
  Ledverde = 1
  Waitms 200
Next X
```

```
Ledrojo = 0
Ledverde = 0
Bandera0 = 0
```

```
Do
  B = Inkey()
Loop Until B = 0
Do
  B = Inkey()
Loop Until B = 0
```

```
'I1 = Chr(&Hf7)
'Print #2 , I1;
'I1 = Chr(&H1)
'Print #2 , I1;
'Do
' B = Inkey(#3)
'Loop Until B = 0
```

Ledrojo = 1

Config\_inicial

Ledverde = 1

Ledrojo = 0

\*\*\*\*\* Programa Principal \*\*\*\*\*

Do

    Gosub Revisarmensaje

'If Flag5 = 0 Then

'    Print #2 , "1" ;                                    'On1 ; Chr(13);

'    Flag5 = 1

'Else

'    Print #2 , "2" ;                                    ' On1 ; Chr(13);

'    Flag5 = 0

'End If

'Wait 2

'    If Bandera0 = 0 Then

'        Toggle Ledverde

'        Ledrojo = 0

'    Else

'        Toggle Ledrojo

'        Ledverde = 0

'    End If

'    Gosub Revisarmensaje

'        Mensaje = "Prueba Mensajes"

'        Mensa = Mensaje

'        Enviarmensaje Mensa , Numero1

'        Limpiarbuffer

'        Mensa = ""

Loop

\*\*\*\*\*

Revisarmensaje:

    If Ischarwaiting() = 1 Then

        Recibirmensaje

'    Cls

'    Lcd Mensaje

    Waitms 500

    Print #2 , Mensaje ;                                    'On1 ; Chr(13);

'    Validarmensaje , Mensaje

    End If

Return

\*\*\*\*\*

Rec\_isr1:

'INTERRUPCION DEL XBEE

    Disable Interrupts

    Disable Urx

    Input Serial1 Noecho

```

    Temporal1 = Mid(serial1 , 1 , 1)
    Do
        B = Inkey()
        Loop Until B = 0
    Enable Interrupts
    Enable Urxc
Return
*****
'*          DESARROLLO DE SUBRUTINAS          *
*****

*****

'*  SUBROUTINA DE CONFIGURACION INICIAL  *
*****

' SE ENVIA AT          -> SE ESPERA "OK" -> INICIAR COMUNICACION
' SE ENVIA ATE0        -> SE ESPERA "OK" -> APAGAR EL ECO DE RESPUESTA
' SE ENVIA AT+CMGF=1   -> SE ESPERA "OK" -> Se configura RESPUESTA EN
MODO TEXTO
' SE ENVIA AT+CNMI=1,2,0,0,0 -> SE ESPERA "OK" -> CUANDO ENTRA MENSAJE
ENVIARLO POR CABLE

Sub Config_inicial()
    Wait 10
    Limpiarbuffer

    Do
        Print "AT" ; Chr(13);          ' Se envía "AT" + ENTER
        Getok Sret
        Loop Until Sret = "OK"
        Limpiarbuffer
        Waitms 25
    Do
        Print "ATE0" ; Chr(13);
        Getok Sret
        Loop Until Sret = "OK"
        Limpiarbuffer
        Waitms 25
    Do
        Print "AT+IPR=9600" ; Chr(13);  ' COMUNICACION A 9600 BPS
        Getok Sret
        Loop Until Sret = "OK"
        Limpiarbuffer
    Do
        Print "AT+CMGF=1" ; Chr(13);
        Getok Sret
        Loop Until Sret = "OK"
        Limpiarbuffer

```

```

Waitms 25
Do
  Print "AT+CNMI=2,2,0,0,0" ; Chr(13);
  Getok Sret
  Loop Until Sret = "OK"
  Limpiarbuffer
  Waitms 25
  Do
    Print "AT&W" ; Chr(13);          ' GUARDAR LA CONFIGURACION LA
CONFIGURACION ACTUAL
    Getok Sret
    Loop Until Sret = "OK"
    Limpiarbuffer
    Wait 1
  End Sub

```

```

*****
'*  SUBROUTINA DE ESPERA RESPUESTA OK  *
*****

```

```

' Se espera LA RESPUESTA DEL CELULAR -> "OK"

```

```

Sub Getok(s As String)
  S = ""
  Do
    B = Inkey()
    Select Case B
      Case 0
      Case 13
      Case 10
        If S <> "" Then Exit Do
      Case Else
        S = S + Chr(b)
    End Select
  Loop
End Sub

```

```

*****
'*  SUBROUTINA PARA LIMPIAR BUFFER UART2  *
*****

```

```

' DEJAR LIMPIO LOS CANALES DE COMUNICACION ENTRE MICRO Y CELULAR
Sub Limpiarbuffer()
  Waitms 10
  Do
    B = Inkey()
  Loop Until B = 0
End Sub

```



```

*****
'*  SUBROUTINA PARA ENVIAR MENSAJES  *
*****
' AT          -> RESP: "OK"
' AT+CMGS = "NUMERO"      -> RESP: ">"
' Se envía MENSAJE + CTRL Z -> RESP: "OK"
Sub Enviarmensaje(s As String , N As String)
  Do
    Print "AT" ; Chr(13);
    Getok Sret
    Loop Until Sret = "OK"
  Limpiarbuffer
  Do
    Print "AT+CMGS=" ; Chr(34) ; N ; Chr(34) ; Chr(13);
    Sret = ""
  Do
    B = Inkey()
    Select Case B
      Case 0
      Case 13
      Case 10
        If Sret <> "" Then Exit Do
      Case 62
        Goto Envio1
      Case Else
        Sret = Sret + Chr(b)
    End Select
  Loop
  Loop Until Sret = ">"

  Envio1:
  Limpiarbuffer
  Waitms 100
  Limpiarbuffer
  Print S ; Chr(26);
  Do
    Getok Sret
    Loop Until Sret = "OK"
  Limpiarbuffer
  Waitms 500
End Sub

*****
'*  SUBROUTINA PARA RECIBIR MENSAJES  *
*****
Sub Recibirmensaje()
  Do
    B = Inkey()

```

```

Select Case B
  Case 13:
  Case 10:
  Case 45:
    Exit Do
  Case Else
    Mensaje = Mensaje + Chr(b)
End Select
Loop
' Print #1 , Mensaje          '/////
Do
  B = Inkey()
Loop Until B = 10
Mensaje = ""
Do
  B = Inkey()
  Select Case B
    Case 13:
    Case 10:
      Exit Do
    Case Else
      Mensaje = Mensaje + Chr(b)
  End Select
Loop
' Print #1 , Mensaje          '/////
Do
  B = Inkey()
Loop Until B = 0
End Sub
'++++++
+++++
Sub Validarmensaje(s As String)
Dim Zz1 As String * 150
  Zz1 = Ucase(s)
  Waitms 100
  Mensaje = ""
End Sub
'++++++
+++++
!*****

Toggle_ledverde:
' For I1 = 0 To 5
  Toggle Ledverde
  For I2 = 1 To 25
    Waitms 1
  Next I2
' Next I1
' Reset Ledverde

```

Return

\*\*\*\*\*

Toggle\_ledrojo:

```
' For I1 = 0 To 5
  Toggle Ledrojo
  For I2 = 1 To 25
    Waitms 1
  Next I2
' Next I1
' Reset Ledrojo
```

Return

\*\*\*\*\*

End

Dta:

\$eeprom

Data 0 , "0999914345"

Data 0 , "0996206772"

Data 0 , "2356"

Data 0 , "101011111212"

Data 0 , "0922"

\$data

## AVR 8 CONTROL MOTOR PaP

\$regfile = "m8def.dat "

\$crystal = 8000000

\$baud = 9600

'frecuencia del cristal 8 Mhz

\*\*\*\*\* INTERRUPCION de serial \*\*\*\*\*

On Urxc Rec\_isr

'Define subrutina de Interrupcion XBEE

Enable Urxc

\*\*\*\*\*

\* LedH \*

\*\*\*\*\*

Ddrb.5 = 1

Portb.5 = 0

Ledh Alias Portb.5

\*\*\*\*\*

\* LedAH \*

\*\*\*\*\*

Ddrb.1 = 1

Portb.1 = 0

Ledah Alias Portb.1

\*\*\*\*\*

!\* BicolorRojo \*

\*\*\*\*\*  
Ddrd.7 = 1  
Portd.7 = 0  
Ledrojo Alias Portd.7

!\* BicolorVerde \*

\*\*\*\*\*  
Ddrb.0 = 1  
Portb.0 = 0  
Ledverde Alias Portb.0

!\* IR1 \*

\*\*\*\*\*  
Ddrd.5 = 0  
Portd.5 = 1  
Ir1 Alias Pind.5

!\* IR2 \*

\*\*\*\*\*  
Ddrd.6 = 0  
Portd.6 = 1  
Ir2 Alias Pind.6

!\* SW MAN/AUTO \*

\*\*\*\*\*  
Ddrd.2 = 0  
Portd.2 = 1  
Swma Alias Pind.2

!\* BOTONabrir \*

\*\*\*\*\*  
Ddrd.3 = 0  
Portd.3 = 1  
Babrir Alias Pind.3

!\* BOTONcerrar \*

\*\*\*\*\*  
Ddrd.4 = 0  
Portd.4 = 1  
Bcerrar Alias Pind.4

!\* FIN DE CARRERA 1 \*

\*\*\*\*\*  
Ddrc.5 = 0  
Portc.5 = 1  
Fc1 Alias Pinc.5

\*\*\*\*\*

\* FIN DE CARRERA 2 \*

Ddrc.4 = 0  
Portc.4 = 1  
Fc2 Alias Pinc.4

\* MOTOR PAP \*

Ddrc.3 = 1  
Portc.3 = 0  
B1 Alias Portc.3  
Ddrc.2 = 1  
Portc.2 = 0  
B2 Alias Portc.2  
Ddrc.1 = 1  
Portc.1 = 0  
B3 Alias Portc.1  
Ddrc.0 = 1  
Portc.0 = 0  
B4 Alias Portc.0

\* Alarma \*

Ddrb.2 = 1  
Portb.2 = 0  
Alarma Alias Portb.2

Dim Flag1 As Bit 'variable tipo bit  
Dim Flag2 As Bit 'variable tipo bit  
Dim Flag3 As Bit 'variable tipo bit  
Dim Flag4 As Bit 'variable tipo bit  
Dim Flag5 As Bit 'variable tipo bit  
Dim Flag6 As Bit 'variable tipo bit  
Dim Clave1(5) As Byte  
Dim B As Byte  
Dim I As Byte  
Dim J As Byte  
Dim Rx\_rev As Byte  
Dim Xx As Byte  
Dim Yy As Byte  
Dim X As Word  
Dim Y As Word  
Dim Z As Word  
  
Dim Temporal As String \* 1  
Dim Temporal1 As String \* 10  
Dim Serial1 As String \* 20

```
Dim Serial2 As String * 20
```

```
Const Delay1 = 80
```

```
Const Retardo = 20
```

```
Waitms 500
```

```
Flag1 = 0
```

```
Flag2 = 0
```

```
Flag3 = 0
```

```
Flag4 = 0
```

```
Flag5 = 0
```

```
Flag6 = 0
```

```
Do
```

```
    B = Inkey()
```

```
Loop Until B = 0
```

```
Enable Interrupts
```

```
'Alzar barrera
```

```
    If Swma = 0 Then
```

```
        Ledrojo = 1
```

```
        Ledverde = 0
```

```
    Else
```

```
        Ledrojo = 0
```

```
        Ledverde = 1
```

```
    End If
```

```
Gosub Abrirpuerta
```

```
'-----
```

```
Do
```

```
    If Swma = 0 Then
```

```
        Ledrojo = 1
```

```
        Ledverde = 0
```

```
'////////////////////////////////////
```

```
    If Babrir = 0 Then
```

```
        Gosub Toggle_led1
```

```
        While Babrir = 0
```

```
            Gosub Toggle_led1
```

```
        Wend
```

```
        While Ir1 = 1 Or Ir2 = 1
```

```
            Gosub Toggle_led1a
```

```
        Wend
```

```
        Gosub Toggle_led1
```

```
        Gosub Abrirpuerta
```

```
    End If
```

```
'////////////////////////////////////
```

```
    If Bcerrar = 0 Then
```

```
        Gosub Toggle_led2
```

```

While Bcerrar = 0
  Gosub Toggle_led2
Wend
While Ir1 = 1 Or Ir2 = 1
  Gosub Toggle_led2a
Wend
Gosub Toggle_led2
Gosub Cerrarpuerta
End If
Else
  Ledrojo = 0
  Ledverde = 1
'//////////////////////////////////////////////////////////////////
  If Flag1 = 1 Then
    Flag1 = 0
    While Ir1 = 1 Or Ir2 = 1
      Gosub Toggle_led1a
    Wend
    Gosub Abrirpuerta
  End If
'//////////////////////////////////////////////////////////////////
  If Flag2 = 1 Then
    Flag2 = 0
    While Ir1 = 1 Or Ir2 = 1
      Gosub Toggle_led2a
    Wend
    Gosub Cerrarpuerta
  End If
End If
Loop
!*****

Toggle_led1:
  Toggle Ledh
  For Xx = 1 To 250
    Waitms 1
    If Fc2 = 0 Then
      Xx = 255
      Flag5 = 1
    End If
  Next Xx
Return
!*****

Toggle_led2:
  Toggle Ledah
  For Yy = 1 To 250
    Waitms 1
    If Fc1 = 0 Then
      Yy = 255

```

```
    Flag6 = 1
  End If
Next Yy
Return
*****
```

```
Toggle_led1a:
  Toggle Ledh
  Waitms 20
Return
*****
```

```
Toggle_led2a:
  Toggle Ledah
  Waitms 20
Return
*****
```

```
Abrirpuerta:
  While Fc2 = 1 And Flag4 = 0
    Alarma = 1
    B1 = 1
    If Flag5 = 0 Then
      Gosub Toggle_led1
    End If
    B1 = 0 : B2 = 1
    If Flag5 = 0 Then
      Gosub Toggle_led1
    End If
    B2 = 0 : B3 = 1
    If Flag5 = 0 Then
      Gosub Toggle_led1
    End If
    B3 = 0 : B4 = 1
    If Flag5 = 0 Then
      Gosub Toggle_led1
    End If
    B4 = 0
  Wend
  Flag5 = 0
  Ledh = 0
  Alarma = 0
Return
*****
```

```
Cerrarpuerta:
  While Fc1 = 1 And Flag4 = 0
    Alarma = 1
    B4 = 1
    If Flag6 = 0 Then
      Gosub Toggle_led2
    End If
```



```

B4 = 0 : B3 = 1
If Flag6 = 0 Then
    Gosub Toggle_led2
End If
B3 = 0 : B2 = 1
If Flag6 = 0 Then
    Gosub Toggle_led2
End If
B2 = 0 : B1 = 1
If Flag6 = 0 Then
    Gosub Toggle_led2
End If
B1 = 0
Wend
Flag6 = 0
Ledah = 0
Alarma = 0
Return
*****
Rec_isr:                                'INTERRUPCION SERIAL
    Disable Interrupts
    Disable Urx

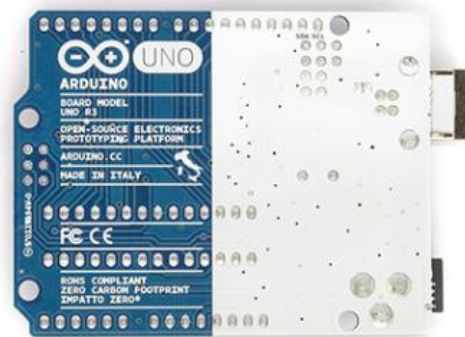
    Rx_rev = Inkey()
    Temporal = Chr(rx_rev)

'    Input Serial1 Noecho
'    Temporal = Mid(serial1 , 1 , 1)
    Select Case Temporal
        Case "A"
            Flag1 = 1
        Case "B"
            Flag2 = 1
        Case Else
            Flag2 = 1
    '        'Print #1 , Temporal1;
    End Select
    Do
        B = Inkey()
    Loop Until B = 0
    Enable Interrupts
    Enable Urx
Return
*****
End

```

## ANEXO 3

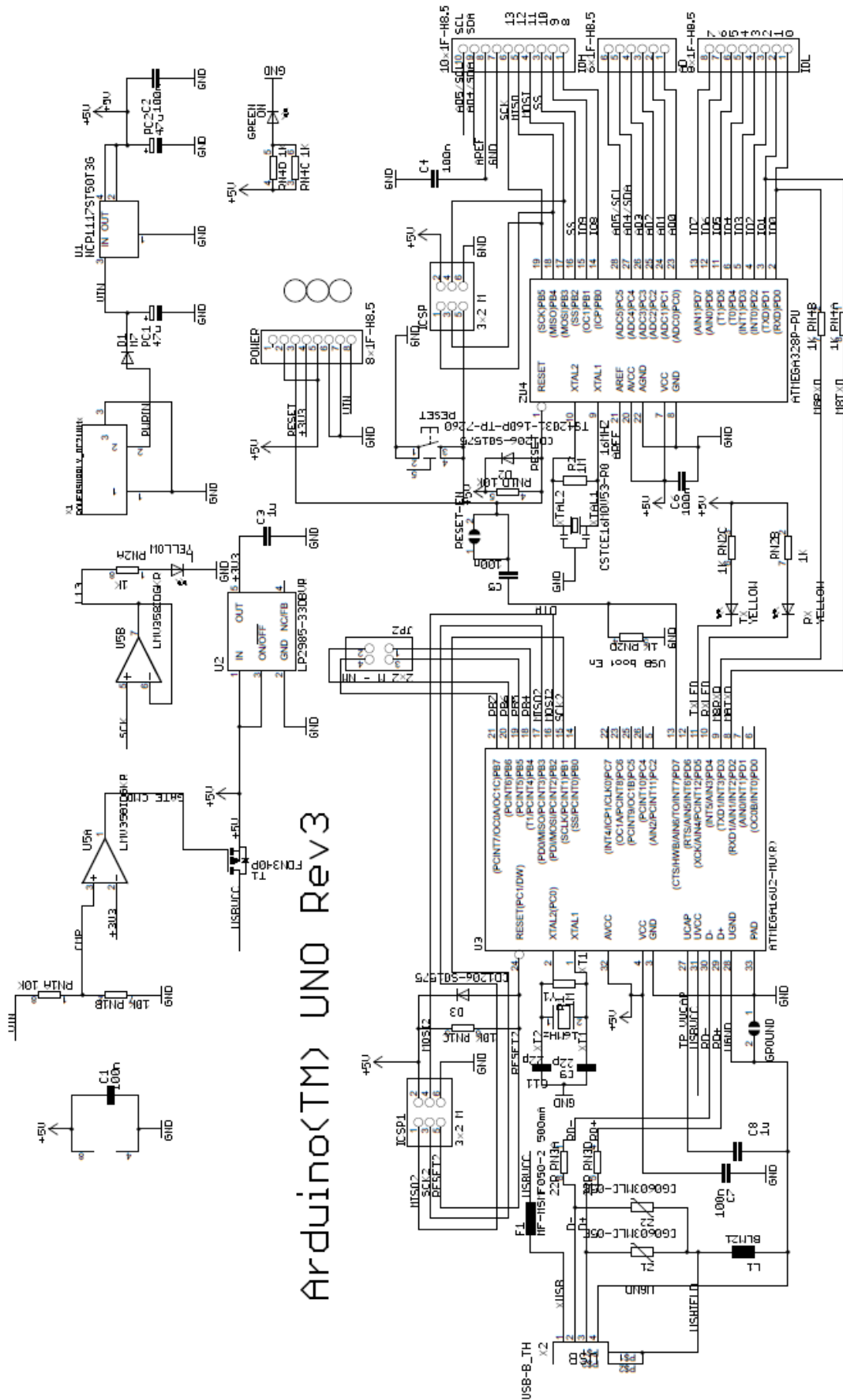
### Arduino UNO



## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

(Arduino, 2014)



# Arduino(TM) UNO Rev3

(Arduino, 2014)

## ANEXO 4

### Microcontrolador AVR 8

#### Features

- High performance, low power Atmel® AVR® 8-bit microcontroller
- Advanced RISC architecture
  - 131 powerful instructions – most single clock cycle execution
  - 32 × 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
  - 4/8/16 Kbytes of in-system self-programmable flash program memory
  - 256/512/512 bytes EEPROM
  - 512/1K/1 Kbytes internal SRAM
  - Write/erase cycles: 10,000 flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>0</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
- QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix 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
  - 6-channel 10-bit ADC in PDIP Package
  - Programmable serial USART
  - Master/slave SPI serial interface
  - Byte-oriented 2-wire serial interface (Philips I<sup>2</sup>C compatible)
  - Programmable watchdog timer with separate on-chip oscillator
  - On-chip analog comparator
  - Interrupt and wake-up on pin change
- Special microcontroller features
  - DebugWIRE on-chip debug system
  - Power-on reset and programmable brown-out detection
  - Internal calibrated oscillator
  - External and internal interrupt sources
  - Five sleep modes: Idle, ADC noise reduction, power-save, power-down, and 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:
  - 1.8V - 5.5V for Atmel ATmega48V/88V/168V
  - 2.7V - 5.5V for Atmel ATmega48/88/168
- Temperature range:
  - -40°C to 85°C
- Speed grade:
  - ATmega48V/88V/168V: 0 - 4MHz @ 1.8V - 5.5V, 0 - 10MHz @ 2.7V - 5.5V
  - ATmega48/88/168: 0 - 10MHz @ 2.7V - 5.5V, 0 - 20MHz @ 4.5V - 5.5V
- Low power consumption
  - Active mode:
    - 250µA at 1MHz, 1.8V
    - 15µA at 32kHz, 1.8V (including oscillator)
  - Power-down mode:
    - 0.1µA at 1.8V

Note: 1. See "Data Retention" on page 7 for details.



**8-bit Atmel  
Microcontroller  
with 4/8/16K  
Bytes In-System  
Programmable  
Flash**

**ATmega48V  
ATmega88V  
ATmega168V**

**Summary**

Rev. 2545TS-AVR-05/11

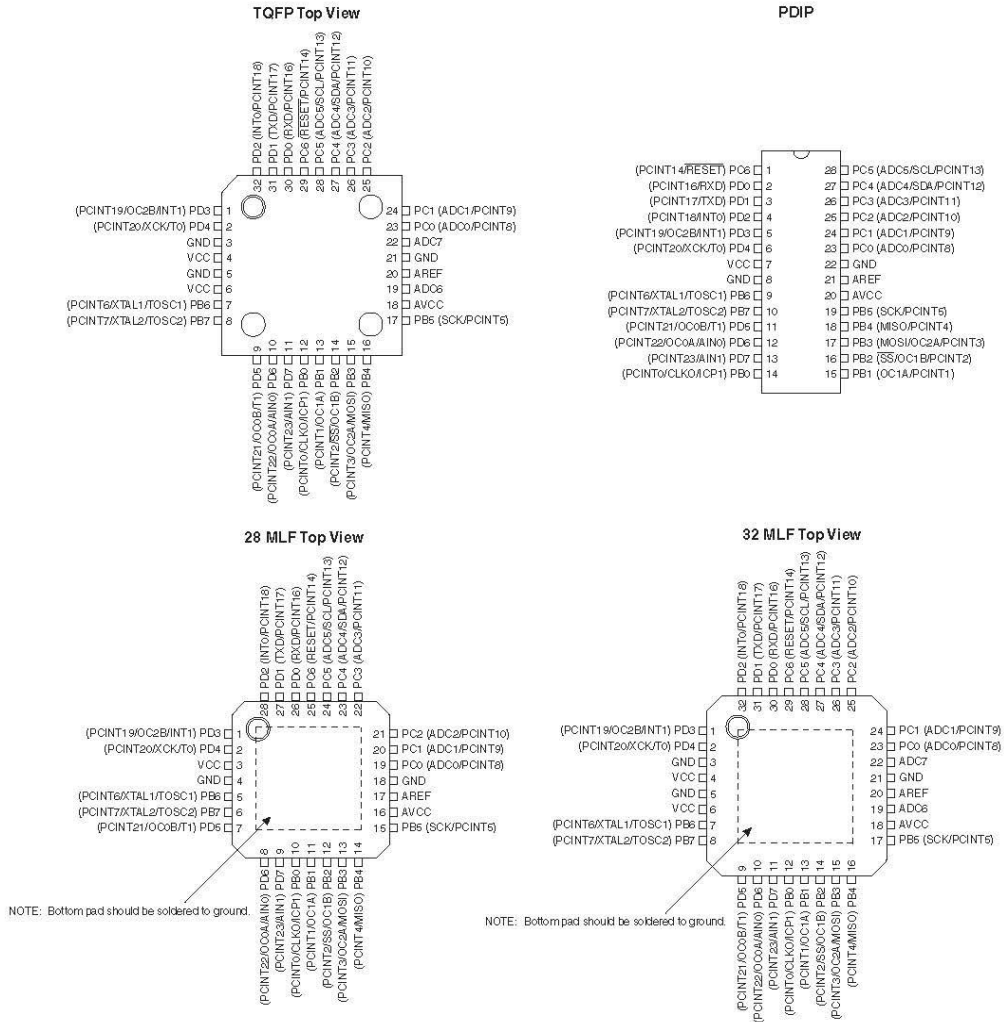


(Atmel, 2010)



# 1. Pin configurations

Figure 1-1. Pinout Atmel ATmega48/88/1682545TS



## 1.1 Pin descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in [“Alternate functions of port B” on page 77](#) and [“System clock and clock options” on page 26](#).

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 1.1.5 PC6/ $\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 306](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in [“Alternate functions of port C” on page 80](#).

### 1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up



resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate functions of port D" on page 83.

**1.1.7 AV<sub>CC</sub>**

AV<sub>CC</sub> is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V<sub>CC</sub>, even if the ADC is not used. If the ADC is used, it should be connected to V<sub>CC</sub> through a low-pass filter. Note that PC6..4 use digital supply voltage, V<sub>CC</sub>.

**1.1.8 AREF**

AREF is the analog reference pin for the A/D Converter.

**1.1.9 ADC7:6 (TQFP and QFN/MLF package only)**

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.



## ANEXO 5 Sensor de Temperatura DS18B20



[www.dalsemi.com](http://www.dalsemi.com)

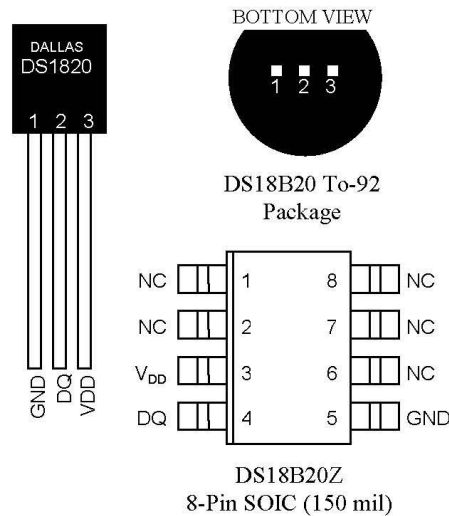
PRELIMINARY

### DS18B20 Programmable Resolution 1-Wire® Digital Thermometer

#### FEATURES

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +257°F
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

#### PIN ASSIGNMENT



#### PIN DESCRIPTION

- GND - Ground
- DQ - Data In/Out
- V<sub>DD</sub> - Power Supply Voltage
- NC - No Connect

#### DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.



**DETAILED PIN DESCRIPTION** Table 1

PIN 8PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
5	1	GND	Ground.
4	2	DQ	<b>Data Input/Output pin.</b> For 1-Wire operation: Open drain. (See "Parasite Power" section.)
3	3	V <sub>DD</sub>	<b>Optional V<sub>DD</sub> pin.</b> See "Parasite Power" section for details of connection. V <sub>DD</sub> must be grounded for operation in parasite power mode.

DS18B20Z (8-pin SOIC): All pins not specified in this table are not to be connected.

## OVERVIEW

The block diagram of Figure 1 shows the major components of the DS18B20. The DS18B20 has four main data components: 1) 64-bit lasered ROM, 2) temperature sensor, 3) nonvolatile temperature alarm triggers TH and TL, and 4) a configuration register. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS18B20 may also be powered from an external 3 volt - 5.5 volt supply.

Communication to the DS18B20 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out a specific device if many are present on the 1-Wire line as well as indicate to the bus master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS18B20 to perform a temperature measurement. The result of this measurement will be placed in the DS18B20's scratch-pad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of 1 byte EEPROM each. If the alarm search command is not applied to the DS18B20, these registers may be used as general purpose user memory. The scratchpad also contains a configuration byte to set the desired resolution of the temperature to digital conversion. Writing TH, TL, and the configuration byte is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

## ANEXO 6 RTC DS1307



### DS1307 64 x 8, Serial, I<sup>2</sup>C Real-Time Clock

#### GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I<sup>2</sup>C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

#### FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, Nonvolatile (NV) RAM for Data Storage
- I<sup>2</sup>C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratory (UL) Recognized

*Typical Operating Circuit and Pin Configurations appear at end of data sheet.*

#### ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z/T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN/T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+ Denotes a lead-free/RoHS-compliant device.

\*A "+" anywhere on the top mark indicates a lead-free device.

*Note: Some revisions of this device may incorporate deviations from published specifications known as errata. Multiple revisions of any device may be simultaneously available through various sales channels. For information about device errata, click here: [www.maxim-ic.com/errata](http://www.maxim-ic.com/errata).*

**ABSOLUTE MAXIMUM RATINGS**

Voltage Range on Any Pin Relative to Ground.....	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial.....	0°C to +70°C
Industrial.....	-40°C to +85°C
Storage Temperature Range.....	-55°C to +125°C
Soldering Temperature (DIP, leads).....	+260°C for 10 seconds
Soldering Temperature (surface mount).....	See JPC/JEDEC Standard J-STD-020

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

**RECOMMENDED DC OPERATING CONDITIONS**

(T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>		4.5	5.0	5.5	V
Logic 1 Input	V <sub>IH</sub>		2.2		V <sub>CC</sub> + 0.3	V
Logic 0 Input	V <sub>IL</sub>		-0.3		+0.8	V
V <sub>BAT</sub> Battery Voltage	V <sub>BAT</sub>		2.0	3	3.5	V

**DC ELECTRICAL CHARACTERISTICS**

(V<sub>CC</sub> = 4.5V to 5.5V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I <sub>LI</sub>		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I <sub>LO</sub>		-1		1	μA
Logic 0 Output (I <sub>OL</sub> = 5mA)	V <sub>OL</sub>				0.4	V
Active Supply Current (f <sub>SCL</sub> = 100kHz)	I <sub>CCA</sub>				1.5	mA
Standby Current	I <sub>CCS</sub>	(Note 3)			200	μA
V <sub>BAT</sub> Leakage Current	I <sub>BATLKG</sub>			5	50	nA
Power-Fail Voltage (V <sub>BAT</sub> = 3.0V)	V <sub>PF</sub>		1.216 x V <sub>BAT</sub>	1.25 x V <sub>BAT</sub>	1.284 x V <sub>BAT</sub>	V

**DC ELECTRICAL CHARACTERISTICS**

(V<sub>CC</sub> = 0V, V<sub>BAT</sub> = 3.0V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V <sub>BAT</sub> Current (OSC ON); SQW/OUT OFF	I <sub>BAT1</sub>			300	500	nA
V <sub>BAT</sub> Current (OSC ON); SQW/OUT ON (32kHz)	I <sub>BAT2</sub>			480	800	nA
V <sub>BAT</sub> Data-Retention Current (Oscillator Off)	I <sub>BATDR</sub>			10	100	nA

**WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.**

ANEXO 7  
Modem Celular



**Plug and Play Wireless CPU®**

**FASTRACK M1306B  
User Guide**

Reference: WM\_PRJ\_M13\_UGD\_001

Revision: 003

Date: November 11, 2006



**PRODUCT FEATURES**

Dual Band GSM modem (EGSM900/1800 MHz) designed for data, fax, SMS and voice applications

Fully Type Approved

Fully compliant with ETSI GSM Phase 2 + Small MS

**Output power:**

Class 4 (2W @ 900 MHz)

Class 1 (1W @ 1800 MHz)

Power supply:

**Input voltage:** 5V-32V

- 5mA in idle mode, 140mA in communication GSM 900 @ 12V

- 5mA in idle mode, 100mA in communication GSM 1800 @ 12V

- Peak 1,7A @ 5.5V

Overall dimensions: 73 x 54 x 25mm

Weight: 82g

- Closed User Group
- Explicit Call Transfer

**OTHER FEATURES**

- Advanced Open Software Platform: MUSE Platform Open AT
- Fixed Dialling Number
- SIM Toolkit Class 2
- SIM, network and service provider locks
- Real Time Clock
- Alarm management
- Software upgrade through Xmodem protocol
- UCS2 character set management

**INTERFACES**

- RS-232 and audio through mini sub-D 15-pin connector supporting:
  - Serial link autoshtutdown controlled by software (AT)

(WAVECOM, 2006)

## ANEXO 8 Display 20 x 4

# LCD-20x4B

## 20x4 Liquid Crystal Display with HD44780 driver (Blue Backlight and White Character)

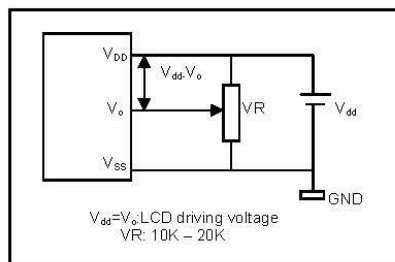
### Specifications

Display Format	20 characters x 4 lines
Drive Method	1/16 duty, 1/5 bias
LCD Type	STN Blue
LED Backlight	White
Viewing direction	6: 00
Operating Temperature	-20°C to 60°C

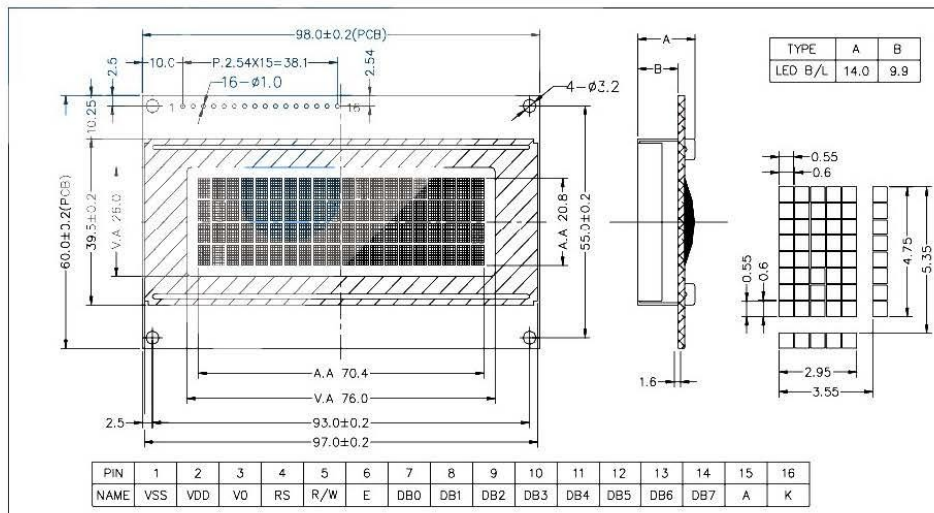
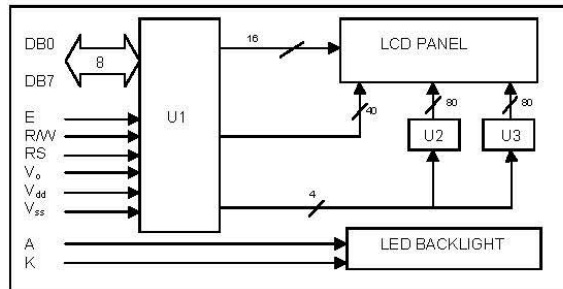
### Electrical Characteristics

Parameter	Symbol	Condition	Stand value			UNIT
			Min.	Typ.	Max.	
Supply Voltage for Logic	$V_{DD}-V_{SS}$	-	4.5	5.0	5.5	V
Supply Voltage for LCD	$V_{DD}-V_o$	-	-	4.7	-	V
Input Logic HIGH	$V_{IH}$	-	2.2	-	$V_{DD}$	V
Input Logic LOW	$V_{IL}$	-	-0.3	-	0.6	V
Output Logic HIGH	$V_{OH}$	$I_{OH}=0.2\text{mA}$	2.4	-	-	V
Output Logic LOW	$V_{OL}$	$I_{OL}=1.2\text{mA}$	-	-	0.4	V
Power Supply Current without backlight	$V_{DD}$	$V_{DD}=5.0\text{V}$	-	2.0	5	mA
Power Supply Current with backlight	$I_{LED}$	$V_{LED}=5.0\text{V}$	-	72	80	mA

### Power Supply



### Block Diagram



(Semiconductor, 2006)

# ANEXO 9 MAX 232



MAX3232

www.ti.com

SLLS410J – JANUARY 2000 – REVISED JANUARY 2014

## MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver With $\pm 15$ -kV ESD Protection

Check for Samples: [MAX3232](#)

### FEATURES

- **RS-232 Bus-Terminal ESD Protection Exceeds  $\pm 15$  kV Using Human-Body Model (HBM)**
- **Meets or Exceeds the Requirements of TIA/EIA-232-F and ITU V.28 Standards**
- **Operates With 3-V to 5.5-V  $V_{CC}$  Supply**
- **Operates up to 250 kbit/s**
- **Two Drivers and Two Receivers**
- **Low Supply Current: 300  $\mu$ A Typical**
- **External Capacitors:  $4 \times 0.1 \mu$ F**
- **Accepts 5-V Logic Input With 3.3-V Supply**
- **Alternative High-Speed Terminal-Compatible Devices (1 Mbit/s)**
  - SN65C3232 ( $-40^{\circ}\text{C}$  to  $85^{\circ}\text{C}$ )
  - SN75C3232 ( $0^{\circ}\text{C}$  to  $70^{\circ}\text{C}$ )

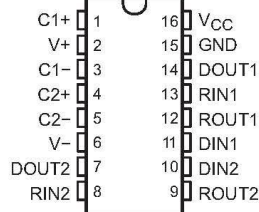
### DESCRIPTION

The MAX3232 device consists of two line drivers, two line receivers, and a dual charge-pump circuit with  $\pm 15$ -kV ESD protection terminal to terminal (serial-port connection terminals, including GND). The device meets the requirements of TIA/EIA-232-F and provides the electrical interface between an asynchronous communication controller and the serial-port connector. The charge pump and four small external capacitors allow operation from a single 3-V to 5.5-V supply. The devices operate at data signaling rates up to 250 kbit/s and a maximum of 30-V/ $\mu$ s driver output slew rate.

### APPLICATIONS

- **Battery-Powered Systems**
- **PDA's**
- **Notebooks**
- **Laptops**
- **Palmtop PCs**
- **Hand-Held Equipment**

D, DB, DW, OR PW PACKAGE  
(TOP VIEW)



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2000–2014, Texas Instruments Incorporated

(TEXAS Instruments 2010)



## ANEXO 10

### Transistores de Potencia TIP-110



**TIP110/112**  
**TIP115/117**

### COMPLEMENTARY SILICON POWER DARLINGTON TRANSISTORS

- STMicroelectronics PREFERRED SALESTYPES
- COMPLEMENTARY PNP - NPN DEVICES
- MONOLITHIC DARLINGTON CONFIGURATION
- INTEGRATED ANTIPARALLEL COLLECTOR-EMITTER DIODE

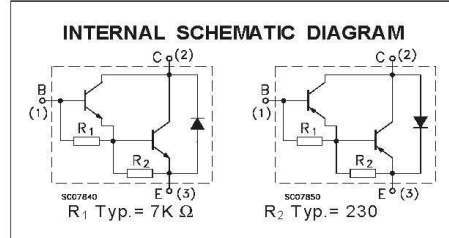
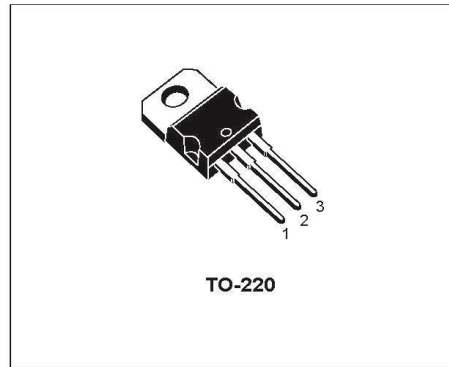
**APPLICATIONS**

- LINEAR AND SWITCHING INDUSTRIAL EQUIPMENT

**DESCRIPTION**

The TIP110 and TIP112 are silicon Epitaxial-Base NPN transistors in monolithic Darlington configuration mounted in Jedec TO-220 plastic package. They are intended for use in medium power linear and switching applications.

The complementary PNP types are TIP115 and TIP117.



**ABSOLUTE MAXIMUM RATINGS**

Symbol	Parameter	Value		Unit	
		NPN	TIP110		TIP112
		PNP	TIP115	TIP117	
$V_{CBO}$	Collector-Base Voltage ( $I_E = 0$ )		60	100	V
$V_{CEO}$	Collector-Emitter Voltage ( $I_B = 0$ )		60	100	V
$V_{EBO}$	Emitter-Base Voltage ( $I_C = 0$ )		5		V
$I_C$	Collector Current		2		A
$I_{CM}$	Collector Peak Current		4		A
$I_B$	Base Current		50		mA
$P_{tot}$	Total Dissipation at $T_{case} \leq 25^\circ C$ $T_{amb} \leq 25^\circ C$		50		W
			2		W
$T_{stg}$	Storage Temperature		-65 to 150		$^\circ C$
$T_J$	Max. Operating Junction Temperature		150		$^\circ C$

\* For PNP types voltage and current values are negative

# ANEXO 11

## LED IR



### Technical Data Sheet

### 5mm Infrared LED , T-1 3/4

---

#### IR333-A

#### Features

- High reliability
- High radiant intensity
- Peak wavelength  $\lambda_p=940\text{nm}$
- 2.54mm Lead spacing
- Low forward voltage
- Pb free
- The product itself will remain within RoHS compliant version.



#### Descriptions

- EVERLIGHT'S Infrared Emitting Diode(IR333-A) is a high intensity diode , molded in a blue transparent plastic package.
- The device is spectrally matched with phototransistor , photodiode and infrared receiver module.

#### Applications

- Free air transmission system
- Infrared remote control units with high power requirement
- Smoke detector
- Infrared applied system

#### Device Selection Guide

LED Part No.	Chip	Lens Color
	Material	
IR	GaAlAs	Blue



## ANEXO 11

### Proforma Barrera Automática

2015 Presupuesto – 0309–025

Sistematic  
ventas@sistematic.ec  
0984252990  
Quito-Ecuador



Fecha: lunes, 9 de marzo de 2015

Cliente: Fernando Segura

Referencia: Barrera FAAC 615 STD

Producto	Precio Unitari	Cantidad	Valor total
BARRERA FAAC 615 ESTANDAR -Barrera automática para barras de hasta 5 metros de largo -Operador oleodinámico con bloqueo en apertura y cierre -Tiempo de apertura 6 segundos -Alimentación 220V ac -Tasca Rectangular	\$1.519,70	1	\$1.519,70
BARRA RECTANGULAR FAAC -Aluminio -4 metros	\$225,20	1	\$225,20
DETECTOR LOOP MAGNETC AUTOCONTROL - Sensor de masa magnética - Espira en la vía	\$292,25	1	\$292,25
INSTALACIÓN - Anclaje de equipos - Conexiones - Puesta en Marcha	\$490,00	1	\$490,00

<b>Subtotal</b>	<b>\$2.527,15</b>
<b>IVA (12%)</b>	<b>\$303,26</b>
<b>Total</b>	<b>\$2.830,41</b>

Forma de Pago: 50% anticipo, 50% contraentrega.

Plazo de entrega: 20 días a partir del anticipo.

Validez de la oferta: 30 días

Garantía 12 meses contra defectos de fabricación.

Garantía no cubre daños que sean ocasionados por: choques, golpes o raspones a los equipos, variaciones de voltaje, ya sean ocasionados por la Empresa Eléctrica o por fenómenos naturales (rayos), manipulación de los equipos por personal no autorizado por Sistematic.

Servicio técnico y repuestos hasta 5 años, tiempo de respuesta para reparaciones y mantenimientos de acuerdo a disponibilidad en un máximo de 5 días laborables.

En caso de requerir una solución de protección de datos en el servidor, para que, en caso de falla de hardware y/o software, los datos de cobro se puedan recuperar; se la debe cotizar por separado.

2015 Presupuesto – 0309–025

Instalación incluye anclaje de los equipos descritos, conexiones y pruebas operativas del sistema.

No incluye obra civil, canales o tuberías para los cables ni acometidas de luz; a menos que estén especificados en la oferta.

El cliente deberá proveer las acometidas de luz: 110V ac y/o 220 V ac.

Cualquier cambio solicitado se aplicará un REGISTRO DE CAMBIOS que podrá afectar el valor final y el plazo de entrega.