



ESCUELA SUPERIOR POLITÉCNICA DECHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y
REDES**

**“IMPLEMENTACION DE TARJETA DE CONTROL DE LAVADORA
INDUSTRIAL DEL HOSPITAL DR. GUSTAVO DOMINGUEZ Z. DE SANTO
DOMINGO DE LOS TSACHILAS.”**

TESIS DE GRADO

PREVIA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

PRESENTADO POR:

JAVIER PATRICIO CENTENO MALDONADO

RIOBAMBA - ECUADOR

2011

A Dios por ser mi fuente de inspiración en todo momento.

A mis padres y mi familia, porque gracias a su esfuerzo, apoyo incondicional, amor y paciencia logre alcanzar esta meta, a mi tutor por transmitirme su forma práctica de resolver los problemas y darme a la vez la libertad de crear e incentivar mis ideas e iniciativas. Y todos aquellos que de una u otra forma me ayudaron a conseguir este logro.

El presente trabajo de titulación está dedicado a mis abuelitos: José Centeno y Rosa Lara que desde el principio de mi carrera estuvieron a mi lado y ahora desde el cielo me bendicen cada día dándome fuerzas para superarme como ellos lo hicieron toda su vida.

NOMBRE

FIRMA

FECHA

**DECANO DE LA FACULTAD DE
INFORMATICA Y ELECTRONICA**

.....

.....

**DIRECTOR DE LA ESCUELA DE
INGENIERIA ELECTRONICA**

.....

.....

DIRECTOR DE TESIS

.....

.....

MIEMBRO DEL TRIBUNAL

.....

.....

**DIRECTOR DEL DPTO DE
DOCUMENTACION**

.....

.....

NOTA DE LA TESIS

.....

“Yo, **JAVIER PATRICIO CENTENO MALDONADO**, soy responsables de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”

Javier Patricio Centeno Maldonado

INDICE DE ABREVIATURAS

ALU	Unidad aritmética lógica
AST	Sistema de humidificación rápida
AVR	Advanced virtual Risc
CISC	Computadores de Juego de Instrucciones Complejo
CPU	Unidad central de procesamiento
EEPROM	Memoria de solo lectura programable y borrable eléctricamente
E/S	Entrada y salida
FLASH	Memoria no volátil de bajo consumo
HS	Cristal de alta potencia
LCD	Liquid cristal display
OTP	One Time Programmable
MCLR	Master clear (Reset)
RAM	Memoria de acceso aleatorio
RISC	Computadores de juego de instrucciones reducido
SISC	Computadores de Juego de Instrucciones Específico
VCCD	Voltaje de corriente continua directa

Índice General

CAPITULO I

MARCO REFERENCIAL.....	15
1.1 INTRODUCCION.....	15
1.2 ANTECEDENTES.....	17
1.3 JUSTIFICACION.....	20
1.4 OBJETIVOS.....	22
1.4.1 OBJETIVO GENERAL.....	22
1.4.2 OBJETIVOS ESPECÍFICOS	22
1.5 HIPÓTESIS.....	23

CAPITULO II

MARCO TEORICO

2.1 LAVADORAS INDUSTRIALES.....	24
2.1.1 Descripción de las partes.	25
2.1.2 Motores.....	27
2.1.3 Datos técnicos	32
2.2 MICROCONTROLADOR.....	33
2.2.1 Recursos similares en todos los microcontroladores.....	34
2.2.2 Recursos Especiales	40
2.3 AVR	45
2.3.1 Comparación ATMEL vs PIC.....	49
2.4 MICROCONTROLADOR ATMEGA164P.....	52
2.4.1 Características:.....	52
2.4.2 Configuración de pines	56
2.4.3 Revisión Global	56
2.6 ORCAD.....	58
2.7 BASCON AVR	62
2.7.1 Características:.....	62
2.7.2 Comandos e instrucciones:	64

2.7.3 Como programar con el Bascom AVR:.....	66
CAPITULO III	
HARDWARE	
3.1 DISEÑO	69
3.1.1 Módulo de entradas.....	70
3.1.2 Módulo de control	75
3.1.3 Módulo de salidas.....	83
3.1.4 Estructura física	86
3.2 IMPLEMENTACIÓN.....	87
3.2.1 Ruteo	88
3.2.2 Atacado	88
3.2.3 Perforado.....	92
3.2.4 Montaje.....	94
3.2.5 Pruebas	95
3.2.6 Protección.....	96
CAPITULO IV	
SOFTWARE	
4.1 Aspectos en el Programa para que funcione el Hardware.....	98
4.2 Programas de prueba de hardware	100
4.3 Contenido	103
4.4 Subrutinas.....	103
4.5 Cuerpo del programa	104
4.6 Interfaz de comunicación PC-Microprocesador	105
CAPITULO V	
EVALUACION Y ANALISIS	
5.1 Resultados.....	112
CONCLUSIONES	
RECOMENDACIONES	

ÍNDICES DE FIGURAS

Figura I.1 Sistemas Electrónicos.....	16
Figura I.2 Hospital “Dr. Gustavo Domínguez Z.”.....	18
Figura I.3 Departamento de lavado.....	19
Figura II.1 Lavadora dañada.....	25
Figura II.2 Motor monofásico de inducción.....	28
Figura II.3 Motor universal.....	29
Figura II.4 Motor Direct-Drive.....	30
Figura II.5 Motor trifásico de inducción.....	31
Figura II.6 Estructura interna de un microcontrolador común.....	35
Figura II.7 Recursos especiales.....	42
Figura II.8 Tecnología Atmel.....	46
Figura II.9 Logos.....	47
Figura II.10 Microcontrolador ATMEGA164p.....	52
Figura II.11 Configuración de pines.....	56
Figura II.12 Interfaz gráfica ORCAD.....	59
Figura II.13 Ejemplo del Capture.....	60
Figura II.14 Ejemplo del Layout.....	61
Figura II.15 interfaz Bascom AVR.....	63
Figura II.16 Funciones especiales de Bascom AVR.....	67

Figura III.1 Diagrama de bloques de la tarjeta.....	70
Figura III.2 Optoacopladores.....	71
Figura III.3 a) Puente <i>común 1 común 2</i> b) <i>común 1 común 2</i> separados.....	72
Figura III.4 a) Diseño de leds b) Led del módulo de entrada.....	73
Figura III.5 Conectores de control.....	74
Figura III.6 Capacitores de adecuación.....	74
Figura III.7 Microprocesador Atmega164p.....	75
Figura III.8 Alimentación de la tarjeta.....	76
Figura III.9 Conexión del cristal externo.....	77
Figura III.10 Conexión del reset.....	78
Figura III.11 Programación serial en el circuito.....	79
Figura III.12 Programador.....	80
Figura III.13 Comunicación PC – Microcontrolador.....	80
Figura III.14 Configuración LCD.....	81
Figura III.15 Back light Key.....	82
Figura III.16 Conectores de control.....	82
Figura III.17 Configuración de Relés.....	83
Figura III.18 Configuración de los leds de salida.....	84

Figura III.19 Integrados ULN2803A.....	85
Figura III.20 Capacitores de adecuación.....	85
Figura III.21 Conectores de control.....	86
Figura III.22 Prototipo de la Estructura física.....	87
Figura III.23 Formato de plaqueta.....	89
Figura III.24 Hoja azul – Atacado.....	90
Figura III.25 Plaqueta.....	91
Figura III.26 Plaqueta con las pistas limpias.....	92
Figura III.27 Perforaciones.....	93
Figura III.28 Elementos necesario.....	94
Figura III.29 Montaje.....	95
Figura III.30 Prueba con multímetro.....	96
Figura III.31 Tarjeta protegida.....	97
Figura IV.1 Contenido del programa.....	102
Figura IV.2 Contenido ciclos de lavado.....	103
Figura IV.3 Contenido de cada proceso de lavado.....	105
Figura IV.4 Pantalla Inicio de sesión.....	106
Figura IV.5 Pantalla Control de tiempos de lavado.....	107
Figura V.1 Error en el LCD.....	109

Figura V.2 Reemplazo de contactores.....	110
Figura V.3 Metodo de trifásico 220 V a 110 V.....	111

ÍNDICES DE TABLAS

Tabla IV.I. Instrucciones de inicialización.....	96
Tabla IV.II. Subrutinas.....	99
Tabla V.I. Proceso de lavado 1.....	108
Tabla V.II. Proceso de lavado 2.....	108
Tabla V.III. Proceso de lavado 3.....	109
Tabla V.IV. Proceso de lavado 4.....	109

ÍNDICES DE ANEXOS

ANEXO A Fotografías

ANEXO B Programa de prueba uno

ANEXO C Programa de prueba dos

ANEXO D Programa de prueba tres

ANEXO E Programa principal

ANEXO F Datasheet ATMEGA164P

ANEXO G Modulo de control

CAPÍTULO I

MARCO REFERENCIAL

1.1 INTRODUCCION

La electrónica desarrolla en la actualidad una gran variedad de tareas. Los principales usos de los circuitos electrónicos son el control, el procesamiento, la distribución de información y la conversión y la distribución de la energía eléctrica, en este caso se controló el funcionamiento industrial de una lavadora con un sistema electrónico diseñado exclusivamente para ello.

Un sistema electrónico es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Una forma de entender los sistemas electrónicos consiste en dividirlos en las siguientes partes:

- **Entradas o Inputs** – Sensores (o transductores) electrónicos o mecánicos que toman las señales (en forma de temperatura, presión, etc.) del mundo físico y las convierten en señales de corriente o voltaje.

Ejemplo: El termopar, la foto resistencia para medir la intensidad de la luz, etc.

- **Circuitos de procesamiento de señales** – Consisten en piezas electrónicas conectadas juntas para manipular, interpretar y transformar las señales de voltaje y corriente provenientes de los transductores.
- **Salidas u Outputs** – Actuadores u otros dispositivos (también transductores) que convierten las señales de corriente o voltaje en señales físicamente útiles. Por ejemplo: un display que nos registre la temperatura, un foco o sistema de luces que se encienda automáticamente cuando este oscureciendo.

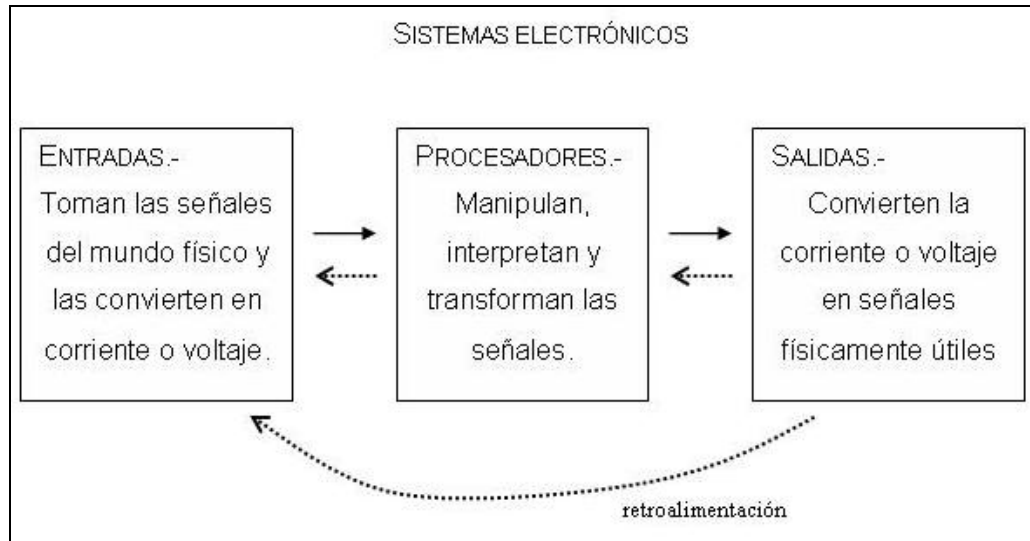


Figura I.1 Sistemas Electrónicos

Básicamente son tres etapas: La primera (transductor), la segunda (circuito procesador) y la tercera (circuito actuador), que se retroalimentan por medio de

sensores u otros dispositivos electrónicos, conformando así el sistema que se va a implementar en el presente trabajo que los acercara a este tema de una manera práctica y guiada en el desarrollo de esta tarjeta de control para lavadoras industriales y que ayudara muy significativamente a una entidad pública como es el caso del hospital “Dr. Gustavo Domínguez Z.” de la ciudad de Santo Domingo de los Tsachilas

1.2 ANTECEDENTES

Por el año 1944 Santo Domingo de los Colorados era un Caserío muy pequeño que estaba ubicado a orillas del río Pobe y le rodeaba selvas vírgenes con un clima inhóspito que tenía lluvias permanentes; estas condiciones climáticas agravaban la salud de los colonos quedando a merced las enfermedades. Años después con el auge comercial que tenía Santo Domingo por la presencia de los caucheros, se funda por primera vez La Casa de salud llamada HOSPITAL DE LOS CAUCHEROS.

En el año 1958 Santo Domingo cuenta con 22.000 habitantes y el HOSPITAL DE LOS CAUCHEROS pasa a denominarse HOSPITAL AUGUSTO EGAS, el cual tenía una capacidad de 30 camas con las especialidades de Clínica, Cirugía, Ginecología, Pediatría, Odontología; Contando con Áreas auxiliares como: LAVANDERÍA, Nutrición, Guardianía, Administración y Contabilidad.

En el año 1977 Santo Domingo cuenta con 115.000 habitantes y se inicia a construir el Nuevo Hospital para Santo Domingo, siendo presidente de la república el Dr. Oswaldo Hurtado.

El 12 de Octubre de 1983 abre sus puertas el Moderno Hospital, con el nombre de “HOSPITAL REGIONAL DE SANTO DOMINGO DE LOS COLORADOS” el mismo que tiene una capacidad de 141 camas, por solicitud de la ciudadanía de Santo Domingo y en honor, a la vida ejemplar del Dr. Gustavo Domínguez, el 12 de Nov. Del 2002 según decreto # 00719 del Ministerio de salud Pública pasa a llamarse Hospital “DR. GUSTAVO DOMÍNGUEZ Z.”.



Figura I.2Hospital “Dr. Gustavo Domínguez Z.”

En la actualidad, Santo Domingo de los Tsachilas cuenta con aproximadamente 500.000 habitantes de los cuales 380.000 son estables y los 120.000 son flotantes, y tiene un solo Hospital donde se atienden a 700 pacientes cada día, 400 por consulta externa y 300 por emergencia, esta alta demanda hace que

todos sus máquinas, equipos e instalaciones trabajen al 100 % de su capacidad, dando como resultado las averías constantes en sus componentes, este es el caso específico del área de LAVANDERIA donde se lava 1650 libras de ropa cada día en cuatro lavadoras industriales y es en esta área donde está ubicado la lavadora KASKADEX 60, objeto de la propuesta de estudio.

Desde la instauración del lavado en serie, la maquina lavadora se ha convertido en una parte vital de los Hospitales e industrias. Las máquinas lavadoras antiguas fueron diseñadas con controles electromecánicos e impulsadas por motores de gran tamaño, con ejes de transmisión, accionadas por poleas y bandas. Para el control de los ciclos de lavado se utilizaba un control electromecánico, este comprendía un conjunto de micro contactos, que se abrían o cerraban con el paso de levas acopladas a un tambor rotativo.



Figura I.3 Departamento de lavado

Estos microcontactos, encendían o apagaban los diferentes dispositivos acoplados a la maquina lavadora. El inconveniente estaba en el desgaste de los micros contactos y de las levas, por lo que su reparación es imposible y se debe efectuar un cambio de todo el programador.

Con la tecnología actual, podemos ya reemplazar este programador de levas, por un programador electrónico, construido y diseñado con la ciencia y técnica de los estudios recibidos en la educación superior, de esta manera podemos implementar un sistema de control para los ciclos de lavado, mejorando la producción y eficiencia de la maquina lavadora de ropa.

La lavadora KASKADEX-60 de procedencia USA fue instalada en el Hospital hace aproximadamente 28 años; y constantemente sufre daños en sus partes electromecánicas por lo que esta propuesta de tesis se basará específicamente en hacer funcionar la máquina con una tarjeta electrónica diseñada y construida para el efecto.

1.2 JUSTIFICACION

El grave problema de paralización constante que tienen las lavadoras del Hospital 'Dr. Gustavo Domínguez Z.' de Santo Domingo de los Tsachilas en su departamento de lavandería, causados por los escasos repuestos electromecánicos existentes en el mercado nacional y consecuentemente el alto costo en sus precios, alientan a dar una solución económica y viable.

Tecnológicamente se puede reemplazar estos elementos electromecánicos obsoletos diseñando y construyendo una tarjeta de control con microprocesadores que automatizará el proceso de lavado, que consisten en varios programas según el tipo de ropa que se va a lavar como ropa contaminada, ropa normal de color, ropa blanca, ropa de cama, etc. y de diferentes ciclos de lavado según necesite el usuario como por ejemplo con ropa contaminada se inicia el proceso con el ingreso del agua fría, prelavado (expulsión y llenado de agua), ingreso de agua caliente, lavado, ingreso de detergente, ingreso de agua fría, centrifugado a alta velocidad, expulsión de agua sucia y finaliza con un enjuague y expulsión del agua sucia.

Posteriormente y de acuerdo a las políticas implantadas por la Dirección del Hospital se implementara este sistema a todas las lavadoras del departamento de lavandería, reduciendo significativamente los costos en la compra de repuestos y minimizando el tiempo de paralización de los equipos.

De esta manera la ESPOCH y particularmente la Escuela de Ingeniería Electrónica fortalecen los vínculos con las instituciones públicas, al brindar el aporte técnico, científico y humanístico en la solución de los problemas presentados en su desenvolvimiento cotidiano, características fundamentales de la educación superior en la formación integral de profesiones comprometidos con la sociedad.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Implementar una tarjeta de control para lavadora industrial del Hospital 'Dr. Gustavo Domínguez Z.' de Santo Domingo de los Tsáchilas con la utilización del Microcontrolador AVR

1.4.2 OBJETIVOS ESPECÍFICOS

- Investigar el funcionamiento, arquitectura física y procesos de la lavadora industrial
- Diseñar una tarjeta electrónica de control capaz de realizar automáticamente los procesos de lavado, enjuague, centrifugado y llenado de agua, requeridos por la lavadora
- Codificar algorítmicamente el microprocesador que va incorporado a la tarjeta electrónica para que funcione cada ciclo de lavado.
- Implementar la tarjeta electrónica de control en la lavadora industrial.
- Implementar una comunicación de la tarjeta con el computador para que se pueda realizar cambios de tiempo en los ciclos de lavado
- Configurar una interfaz simple de comunicación entre la tarjeta y el computador que será instalada en el computador
- Realizar pruebas de funcionamiento de la tarjeta de control en la lavadora industrial.

- Documentar toda la información obtenida desde la investigación hasta la implementación final de la tarjeta de control de la lavadora industrial.

1.5 HIPÓTESIS

La tarjeta de control con microprocesadores que se incorpora a la lavadora industrial pretende reemplazar definitivamente a los elementos electromecánicos logrando así automatizar los ciclos de lavado en forma económica y brindando un respaldo para las demás lavadoras del departamento de lavandería del hospital “Dr. Gustavo Domínguez Z.” de Santo Domingo de los Tsachilas.

CAPÍTULO II

MARCO TEORICO

2.1 LAVADORAS INDUSTRIALES

La lavadora es un aparato eléctrico-electrónico, que puede ser electrodoméstico o de uso industrial, que sirve para lavar la ropa ahorrando esfuerzo, tiempo y agua. Esto lo consigue mezclando agua con jabón o detergente con la ropa sucia y moviendo está a gran velocidad dentro de un tambor.

Actualmente, las lavadoras tienen mecanismos de centrifugado con los que la ropa expulsa parte del agua acumulada durante el proceso de lavado así como diferentes programas de velocidades y temperatura dependiendo del tejido de la ropa.

Básicamente, cuenta con un tambor central con orificios que gira mientras se le introduce agua. Existen mayormente dos grupos de modelos: Las lavadoras horizontales y las verticales. Las horizontales son las que tienen la puerta a un

lado, y el giro del tambor tiene su eje horizontal, de forma que la ropa, al momento de girar, va cayendo permanentemente al ser impulsada por el giro hacia arriba. Las verticales son las que tienen la puerta arriba, y el giro del tambor tiene su eje vertical



Figura II.1 Lavadora dañada

2.1.1 Descripción de las partes.

- **Motor eléctrico:** puede estar situado en la parte inferior de la caja, o bien en centrado en la parte trasera.

El motor recibe energía eléctrica por una entrada de corriente gobernada por un conmutador que proviene de un enchufe y es regido por el microprocesador.

- **Microprocesador:** contiene toda la información programada para controlar el funcionamiento de las distintas operaciones de lavado. Generalmente está situado en la parte superior del lateral derecho de la carrocería. De él sale una línea de mando que se une por debajo con la línea de electricidad hasta el conmutador y hasta el motor decidiendo así la velocidad de este.
- **Tambor:** es aquí donde se introduce la ropa. Es un cilindro de metal con numerosos agujeros para que fluya el agua, muy pequeños para evitar que la ropa se salga por ellos. Es totalmente hueco. En realidad no es cilíndrico ya que de base a base tiene triángulos rebajados en sus puntas para que la ropa gire mejor con el tambor.

En las lavadoras de carga frontal, le falta una de sus bases al cilindro ya que es colocada la puerta. En las de carga superior, el cilindro tiene una apertura que se puede abrir y cerrar con ayuda de unas pequeñas bisagras. El tambor está unido con poleas que atraviesan una cubeta y que conectan con el motor mediante una correa o directamente.

- **Cubeta:** es un cilindro que rodea al tambor y está perforado igual que este, según el tipo de carga de la lavadora. Lógicamente es de mayor tamaño que el tambor.
- **Carter del tambor:** es una pieza con forma de escuadra que sujeta el eje del tambor la cubeta a los laterales de la caja. En su base se encuentran los amortiguadores.

- **Amortiguadores:** son unos muelles que soportan el peso de la cubeta y el tambor. Permiten reducir el sonido y el movimiento o balanceo de la lavadora.
- **Resistencia:** está dispuesto en el espacio que hay entre la cubeta y el tambor, en ocasiones en un depósito condensador.
- **Entradas de agua:** son tuberías de agua caliente y fría que conducen hasta un depósito donde se encuentra la resistencia gobernada por el microprocesador que pone el agua a la temperatura deseada.
- **Filtro:** se encuentra situado en el tubo de desagüe y se encarga de retener objetos que puedan obstruir los conductos. Suele ser extraíble.
- **Depósitos de detergentes:** suele encontrarse en la parte superior izquierda de la lavadora. Tiene varios compartimentos para los diferentes detergentes, suavizantes, etc.
- **Correa:** se encarga de conectar el tambor con el motor cuando no lo hacen directamente.

2.1.2 Motores

El movimiento del tambor es provocado por un motor eléctrico. Los motores más comunes están situados detrás y debajo del tambor y le comunican la tracción a través de poleas y correas.

Actualmente, en el mercado se pueden encontrar lavadoras con uno de los siguientes motores:

- a) Motor monofásico de inducción
- b) Motor universal
- c) Motor “Direct-Drive”
- d) Motor trifásico de inducción

- **Motor Monofásico de inducción**



Figura II.2 Motor monofásico de inducción

En la figura II-02 se puede ver este tipo de motor, aquí se muestra tal y como lo entrega el fabricante sin su polea y con su condensador de arranque.

Este motor es el modelo clásico hasta ahora en todas las lavadoras, y se sigue empleando en modelos de gama baja y poco precio. Es un motor muy veterano, que está siendo sustituido por los motores universales.

Son robustos, de bajo mantenimiento, silenciosos, sin apenas desgaste pues no tienen “escobillas ni contactos en el rotor”, no producen interferencias, su precio es bajo, lo que redunda en una lavadora más económica, su diseño está comprobado en años de fabricación y venta.

Como desventaja se puede acotar que no permiten regulación de velocidad con

triac, su velocidad es baja comparada con otros motores, (400 rpm) lo cual implica centrifugados mediocres y ropa mojada, su eficiencia es del 60% en comparación con otros motores del 80 %, además consumen más energía.

Suelen dar problemas de desgaste de “cojinetes” por el calentamiento inductivo de los mismos.

- **Motor Universal**



Figura II.3 Motor universal

Es el motor más empleado en la actualidad en todos los nuevos modelos de lavadora, su implantación ha sido especialmente alta sobre todo en Europa.

Este tipo de motor puede funcionar tanto con corriente continua como alterna, de ahí su nombre de “universal”.

Son pequeños y robustos, permiten regulación electrónica por triac, pueden girar a muchas revoluciones, (de 900 a 1600 rpm) lo que implica un centrifugado de alta velocidad que deja la ropa más seca. Son más eficientes que otros modelos de hasta un 80%.

En contra se sabe que son modelos recientes en el mercado (no más de 15 años), por tanto adolecen de “enfermedades de diseño”, son mucho más

ruidosos que otro tipo de motor.

Su principal inconveniente es el desgaste de sus escobillas, que produce su parada, así como gran cantidad de chispas y “olor a quemado”. Suelen generar bastantes interferencias en la red, si no están suficientemente filtrados. La principal avería de las lavadoras que emplean este tipo de motor es “triac quemado” por bloqueo del motor en el arranque.

Las lavadoras que usan este tipo de motor deben emplearse en ciclos cortos y no son adecuadas para uso industrial. Dependiendo del uso, suelen necesitar un cambio de escobillas cada 5 años

- **Motor Direct-Drive**

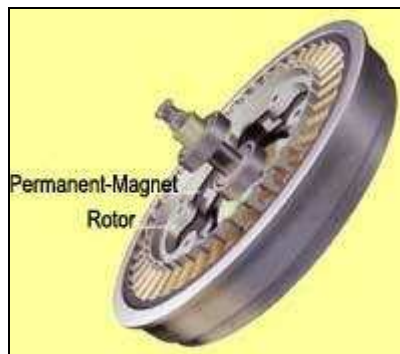


Figura II.4 Motor Direct-Drive

Es un motor sin escobillas, del control directo del tambor, sin poleas o reducción.

Es un motor de reducidas dimensiones y alta calidad, su principal ventaja es el bajo nivel de ruidos y vibración de la lavadora. Tiene un alto par de giro (fuerza) y permite altas revoluciones, lo que implica ropa más seca en el centrifugado. Su velocidad es controlada electrónicamente y no tiene emisión de parásito a la

red eléctrica. Su eficiencia es alta de 85%. Al no tener escobillas su desgaste es mínimo.

Lo negativo de estos motores es que solo lo utilizan por el momento pocas empresas de marca y modelos especiales, lo que implica un alto precio inicial, todo el motor basa su funcionamiento en el control electrónico, por tanto un fallo de la placa inutiliza completamente el motor y eleva el precio de la reparación.

El precio del regulador electrónico, también incrementa el precio final del aparato.

- **Motor trifásico de inducción**

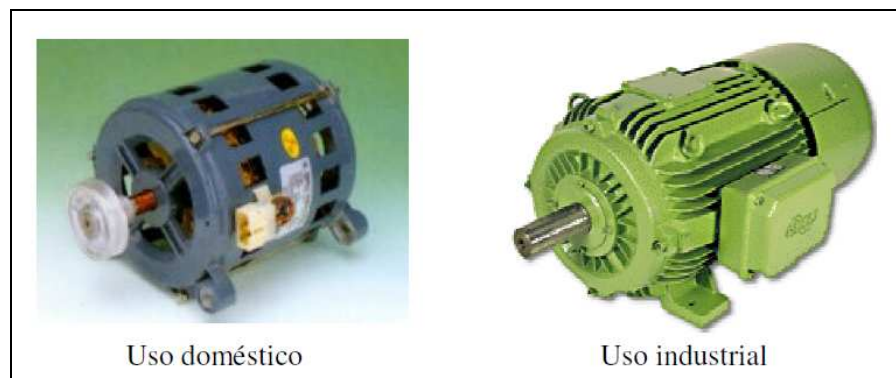


Figura II.5 Motor trifásico de inducción

De aspecto muy parecido al monofásico, tiene importantes ventajas respecto a él. Este tipo de motor, pero más grande, es el empleado en lavadoras industriales y en todo el sector industrial.

Sin duda el mejor de todos, es el motor que se impondrá en todos los modelos futuros de lavadoras.

Es muy eficiente 85%, no tiene escobillas ni apenas desgaste, es silencioso y sin vibraciones, su diseño permite recuperar energía en el frenado, y permite

altas velocidades de centrifugado, dando como resultado ropa muy seca.

Permite la variación de velocidad con “variadores de frecuencia” integrados en la lavadora.

La única deficiencia, pero muy importante, para su funcionamiento y regulación, es que emplea electrónica muy costosa pues su potencia y ventajas dependen del “alimentador” y “variador de frecuencia”, por lo que el precio es alto y su reparación costosa, es por esta razón que la finalidad de este trabajo se enfocó en un ámbito industrial minimizando todas estas desventajas en su funcionamiento y duración.

2.1.3 Datos técnicos

Al vender o comprar lavadoras sean estas industriales o domesticas se debe tener en cuenta algunos aspectos técnicos que se encuentran en catálogos comerciales donde figuran el dato concreto que corresponde a cada modelo de lavadora, estos son:

- Capacidad de carga
- Eficiencia energética: lavado/centrifugado
- Velocidad de centrifugado
- Motor
- Nivel ruidos lavado dBA
- Nivel de ruidos centrifugado dBA
- Potencia de Conexión, W
- Tensión, V

- Frecuencia, Hz
- Control
- Panel textos
- Duración del programa seleccionado
- Pre-programación y Auto-diagnos
- Control automático de carga
- Sistema anti-vibraciones
- Sistema de humidificación rápida (AST)
- Sistema antiespuma
- Consumo de agua
- Consumo energético
- Sistema de desagote
- Medidas:(Altura. x Anchura. x Fondo.)
- Color

2.2 MICROCONTROLADOR

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos y pequeños. Entre ellos los microprocesadores y los microcontroladores.

Los microcontroladores están conquistando el mundo. Están presentes en nuestro trabajo, en nuestra casa y en nuestra vida, en general. Se pueden encontrar controlando el funcionamiento de los ratones y teclados de los computadores, en los teléfonos, en los hornos microondas y los televisores de

nuestro hogar. Pero la invasión acaba de comenzar y el siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabriquemos y usamos los humanos.

Un Microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos y componentes fundamentales de un ordenador, aunque de limitadas prestaciones y que se suele destinar a gobernar una sola tarea.

2.2.1 Recursos similares en todos los microcontroladores

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales: Procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

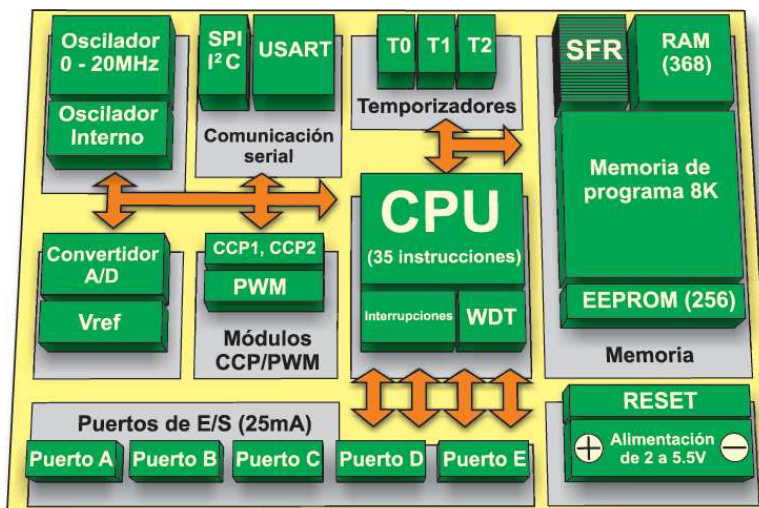


Figura II.6 Estructura interna de un microcontrolador común

- **El procesador o CPU**

Es el elemento más importante del Microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código OP de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están decantándose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de

instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

- **Memoria**

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de los computadores personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes.

Como el Microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadores personales están habituados a manejar Megabytes de memoria, pero, los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1º. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado coste del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2ª. OTP

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es

el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

3ª EPROM

Los microcontroladores que disponen de memoria EPROM (ErasableProgrammableReadOnlyMemory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

4ª EEPROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (ElectricalErasableProgrammableReadOnlyMemory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de un PC. Es muy cómoda y rápida la

operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

Este tipo de memoria es relativamente lenta.

5ª FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta.

- **Puertas de Entrada y Salida.**

La principal utilidad de las patitas que posee la cápsula que contiene un Microcontrolador es soportar las líneas de E/S que comunican al computador interno con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de Microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

- **Reloj principal**

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

2.2.2 Recursos Especiales

Cada fabricante oferta numerosas versiones de una arquitectura básica de Microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo

mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software.

Los principales recursos específicos que incorporan los microcontroladores son:

- **Temporizadores o "Timers"**

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso.

Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patitas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

- **Perro guardián o "Watchdog"**

Cuando el computador personal se bloquea por un fallo del software u otra causa, se pulsa el botón del reset y se reinicializa el sistema. Pero un microcontrolador funciona sin el control de un supervisor y de forma continuada las 24 horas del día. El Perro guardián consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

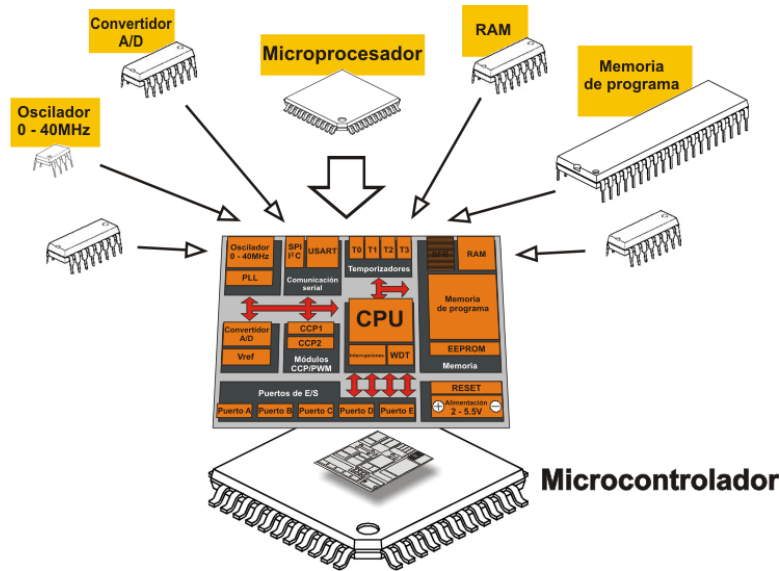


Figura II.7 Recursos especiales

Se debe diseñar el programa de trabajo que controla la tarea de forma que refresque o inicialice al Perro guardián antes de que provoque el reset. Si falla el programa o se bloquea, no se refrescará al Perro guardián y, al completar su temporización, "ladrará y ladRARá" hasta provocar el reset.

- **Estado de reposo o de bajo consumo**

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se "congelan" sus circuitos asociados,

quedando sumido en un profundo "sueño" el microcontrolador. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo.

- **Conversor A/D (CAD)**

Los microcontroladores que incorporan un Conversor A/D (Analógico/Digital) pueden procesar señales analógicas, tan abundantes en las aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patitas del circuito integrado.

- **Conversor D/A (CDA)**

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patitas de la cápsula. Existen muchos efectores que trabajan con señales analógicas.

- **Comparador analógico**

Algunos modelos de microcontroladores disponen internamente de un Amplificador Operacional que actúa como comparador entre una señal fija de referencia y otra variable que se aplica por una de las patitas de la cápsula. La salida del comparador proporciona un nivel lógico 1 ó 0 según una señal sea mayor o menor que la otra.

También hay modelos de microcontroladores con un módulo de tensión de referencia que proporciona diversas tensiones de referencia que se pueden aplicar en los comparadores.

- **Modulador de anchura de impulsos o PWM**

Son circuitos que proporcionan en su salida impulsos de anchura variable, que se ofrecen al exterior a través de las patitas del encapsulado.

- **Puertos de E/S digitales**

Todos los microcontroladores destinan algunas de sus patitas a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de ocho en ocho formando Puertos.

Las líneas digitales de los Puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.

- **Puertos de comunicación**

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

UART, adaptador de comunicación serie asíncrona.

USART, adaptador de comunicación serie síncrona y asíncrona

Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.

USB (Universal Serial Bus), que es un moderno bus serie para los PC.

Bus I²C, que es un interfaz serie de dos hilos desarrollado por Philips.

CAN (ControllerArea Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

2.3 AVR

El significado de AVR no está muy concreto pero según ATMEL el distribuidor de estos microprocesadores, sólo es un nombre que apareció sin ningún significado. Por otra parte dicen que significa Advanced Virtual RISC. Otros dicen que lleva las iniciales de los inventores de los AVR: Alf EgilBogen y VegardWollan es decir Alf VegardRisk (AVR).

Los **AVR** son una familia de microcontroladoresRISC de Atmel. La arquitectura de los AVR fue concebida por dos estudiantes en el NorwegianInstitute of Technology, y posteriormente refinada y desarrollada en **AtmelNorway**, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip.

El AVR es una CPU de arquitectura Harvard. Tiene 32 registros de 8 bits. Algunas instrucciones sólo operan en un subconjunto de estos registros. La

concatenación de los 32 registros, los registros de entrada/salida y la memoria de datos conforman un espacio de direcciones unificado, al cual se accede a través de operaciones de carga/almacenamiento. A diferencia de los microcontroladores PIC, el stack se ubica en este espacio de memoria unificado, y no está limitado a un tamaño fijo.

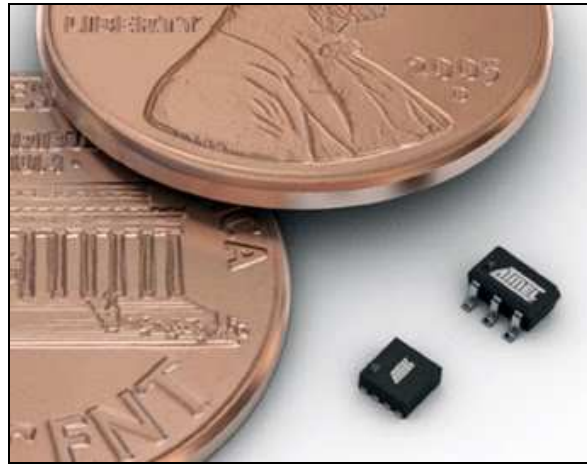


Figura II.8 Tecnología Atmel

El AVR fue diseñado desde un comienzo para la ejecución eficiente de código C compilado. Como este lenguaje utiliza profusamente punteros para el manejo de variables en memoria, los tres últimos pares de registros internos del procesador, son usados como punteros de 16 bit al espacio de memoria externa, bajo los nombres X, Y y Z. Esto es un compromiso que se hace en arquitecturas de ocho bit desde los tiempos de Intel 8008, ya que su tamaño de palabra nativo de 8 bit (256 localidades accedidas) es pobre para direccionar. Por otro lado, hacer que todo el banco superior de 16 registros de 8 bit tenga un comportamiento alterno como un banco de 8 registros de 16 bit, complicaría

mucho el diseño, violando la premisa original de su simplicidad. Además, algunas instrucciones tales como 'suma inmediata' ('addimmediate' en inglés) faltan, ya que la instrucción 'resta inmediata' ('subtractimmediate' en inglés) con el complemento dos puede ser usada como alternativa.



Figura II.9 Logos

El set de instrucciones AVR está implementado físicamente y disponible en el mercado en diferentes dispositivos, que comparten el mismo núcleo AVR pero tienen distintos periféricos y cantidades de RAM y ROM: desde el microcontrolador de la familia *Tiny AVR* ATtiny11 con 1KB de memoria flash y sin RAM (sólo los 32 registros), y 8 pines, hasta el microcontrolador de la familia *Mega AVR* ATmega2560 con 256KB de memoria flash, 8KB de memoria RAM, 4KB de memoria EEPROM, conversor análogo digital de 10 bits y 16 canales, temporizadores, comparador analógico, JTAG, etc. La compatibilidad entre los distintos modelos es preservada en un grado razonable.

Los microcontroladores AVR tienen una cañería ('pipeline' en inglés) con dos etapas (cargar y ejecutar), que les permite ejecutar la mayoría en un ciclo de

reloj, lo que los hace relativamente rápidos entre los microcontroladores de 8-bit.

El set de instrucciones de los AVR es más regular que la de la mayoría de los microcontroladores de 8-bit (por ejemplo, los PIC). Sin embargo, no es completamente ortogonal:

- Los registros punteros X, Y y Z tienen capacidades de direccionamiento diferentes entre sí
- Los registros 0 al 15 tienen diferentes capacidades de direccionamiento que los registros 16 al 31.
- Las registros de I/O 0 al 31 tienen distintas características que las posiciones 32 al 63.
- La instrucción CLR afecta los 'flag', mientras que la instrucción SER no lo hace, a pesar de que parecen ser instrucciones complementarias

Como los PIC, tiene una comunidad de seguidores, principalmente debido a la existencia de herramientas de desarrollo gratuitas o de bajo coste. Estos microcontroladores están soportados por tarjetas de desarrollo de costo razonable, capaces de descargar el código al microcontrolador, y por una versión de las herramientas GNU. Esto último es posible por su uniformidad en el acceso al espacio de memoria, propiedad de la que carecen los procesadores de memoria segmentada o por bancos, como el PIC o el 8051 y sus derivados.

2.3.1 Comparación ATMEL vs PIC

En la actualidad hay dos grandes "casas" de los microcontroladores: el PIC de la familia Microchip y el Atmel de la familia AVR. También está el MSP430 de TI que tiene cosas muy interesantes para el desarrollo en él, pero no lo suficiente como para ponerlo a comparación, además de los micros Silabs que son impresionantes, pero no realmente para el mercado aficionado ni para proyectos especializados.

Por lo que los microprocesadores más populares a nivel global son el ATMEL AT90S1200 de AVR y el 16F84 de Microchip. A continuación se realiza algunos datos comparativos entre el AT90S1200 y el PIC16F84:

Nº de instrucciones

AVR: 89

PIC: 35

Registros RAM

AVR: 32

PIC: 68

Velocidad

AVR: 12MHz

PIC: 20MHz

Memoria de Programa

AVR: 1kByte FLASH (512 líneas de programa, 16bits por inst.)

PIC: 1kx14 (1024 líneas de programa de 14 bit cada una).

Memoria EEPROM libre

AVR: 64Bytes

PIC: 64Bytes

Salidas

AVR: 15

PIC: 13

TIMER:

AVR: 1 de 8bit (con prescaler desde CK hasta CK/1024)

PIC: 1 de 8 bit (con prescaler desde 1:2 hasta 1:256)

Comparador Analógico (NO ADC)

AVR: 1

PIC: NO POSEE

Watchdog: Ambos poseen

Oscilador interno: Ambos poseen, en el AVR sólo habilitable con programación paralela

Niveles de pila (STACK)

AVR: 3

PIC: 8

Interrupciones

AVR: reset, interna, externa, timer y por comparador analógico

PIC: 5 interrupciones

Como pudimos observar los microcontroladores ATMEGA tienen muchas más funcionalidades que los PIC, pero en el momento de elegir se debe tener en cuenta las facilidades que tiene cada uno, además del precio y su utilización, ahora para proyectos grandes se necesita mayor respaldo, seguridad y velocidad de procesamiento teniendo todos estos aspectos en los AVR, por lo que el atmega164p es considerado para profesionales que lo han utilizado como un microprocesador de 40 pines más completo del mercado ya que como veremos en el siguiente apartado está compuesto de comunicación serial, mayor velocidad, gran espacio en memoria gracias a su arquitectura, PWMs (importante para motores), mayor número de instrucciones, trabajo normal con 2,8 V, facilidad de programación, aunque no todo es perfecto porque el atmega164p y en común en la mayoría de AVR's tienen dos desventajas que son: no existe mucha información para poder consultar, como trabajos, proyecto

o publicidad que nos puedan ayudar y si se encuentra alguna información todo está en inglés, el otro problema de los avrs es que después de un tiempo son reemplazados por modelos mejorados, dejando de fabricarlos y entre estos no son compatibles, es decir llegan a cambiar la arquitectura, las instrucciones básicas, la posición de los pines y su función, haciendo difícil poder reemplazarlos en algún proyecto si existe algún daño.

2.4 MICROCONTROLADOR ATMEGA164P

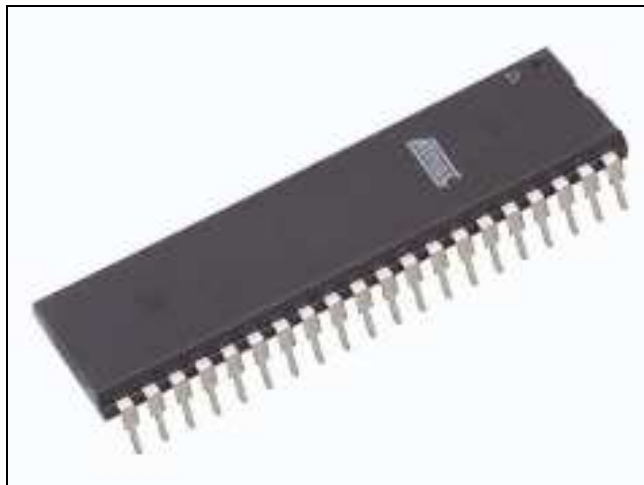


Figura II.10 Microcontrolador ATMEGA164p

2.4.1 Características:

- **Microcontrolador AVR de 8 bits de alto rendimiento y bajo consumo.**
- **Arquitectura Avanzada RISC**
 - 131 instrucciones. La mayoría de un solo ciclo de reloj de ejecución.
 - 32 registros de trabajo de 8 bits para propósito general.
 - Funcionamiento estático total.

- Capacidad de procesamiento de unos 20 MIPS a 20 MHz.
- Multiplicador por hardware de 2 ciclos
- **Memorias de programa y de datos no volátiles de alta duración**
 - 16 K bytes de FLASH auto programable en sistema
 - 512B bytes de EEPROM
 - 1 bytes de SRAM Interna
 - Ciclos de escritura/borrado: 10.000 en Flash / 100.000 en EEPROM
 - Retención de Datos: 20 años a 85°C / 100 años a 25°C
 - Sección opcional de código Boot con bits de bloqueo independientes. Programación en sistema del programa Boot que se encuentra dentro del mismo chip. Operación de lectura durante la escritura.
 - Bloqueo programable para la seguridad del software.
- **Interface JTAG**
 - Capacidades de BoundaryScan de acuerdo con el estándar JTAG
 - Soporte Extendido Debug dentro del chip
 - Programación de FLASH, EEPROM, fusibles y bits de bloqueo a través de la interface JTAG.
- **Características de los periféricos**
 - Dos Timer/Contadores de 8 bits con prescalamiento separado y modo

- Un Timer/Contador de 16 bits con prescalamiento separado, modo comparación y modo de captura.
- Contador en Tiempo Real con Oscilador separado
- 6 Canales para PWM
- ADC de 10 bits y 8 canales
- Modo Diferencial con ganancia seleccionable a x1, x10 o x200.
- Interface serie de dos hilos con byte orientado.
- Dos puertos Seriales USART Programables
- Interfaz Serial SPI maestro-esclavo
- WatchdogTimer programable con oscilador independiente
- Comparador Analógico dentro del mismo Chip
- Interrupt and Wake-up on Pin Change
- **Características especiales del microcontrolador**
 - Power-onReset (en el encendido) y detección de Brown-out (pérdida de polarización) programable.
 - Oscilador RC interno calibrado.
 - Fuentes de interrupción externas e internas.

- 6 modos de descanso: Idle, Reducción de Ruido ADC, Power-save, Power-down, Standby y Standby extendido.

- **Encapsulados para Entradas/Salidas (E/S)**

- 32 líneas de E/S programables.

- PDIP de 40 pines, TQFP y QFN/MLF de 44 pines.

- **Voltajes de Operación**

- 1.8 – 5.5V para el ATMEGA 164P

- 2.7 – 5.5V para el ATMEGA 164P

- **Velocidad de Funcionamiento**

- ATMEGA 164P: 0 – 4MHz @ 1.8 – 5.5V - 10MHz @ 2.7 – 5.5V

- ATMEGA 164P: 0 – 10MHz @ 2.7 – 5.5V - 20MHz @ 4.5 – 5.5V

- **Consumo de energía a 1MHz, 1.8V, 25°C para el ATMEGA 164P**

- Activo: 0.4mA

- Modo Power-down: 0.1uA

- Modo Power-Save: 0.6uA (Incluye RTC de 32 KHz)

2.4.2 Configuración de pines

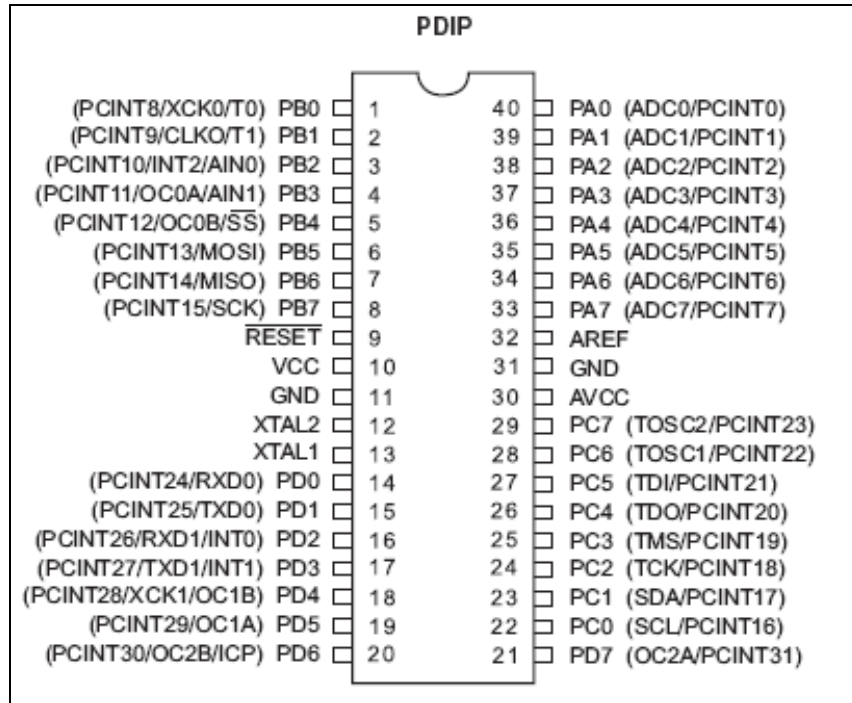


Figura II.11 Configuración de pines

2.4.3 Revisión Global

El ATmega164P es un microcontrolador CMOS de 8 bits de bajo consumo basado en la arquitectura RISC mejorada. Sus instrucciones se ejecutan en un ciclo de máquina, el ATmega164P consigue transferencia de información alrededor de 1 MIPS por MHz admitido por el sistema, permitiendo al diseñador del sistema optimizar el consumo de energía versus la velocidad de procesamiento.

- **Puerto A (PA7:PA0)**

El puerto A sirve como entradas analógicas para el conversor Análogo Digital.

El puerto A también sirve como un puerto bidireccional de 8 bits con resistencias internas de pull up (seleccionables para cada bit). Los buffers de salida del puerto A tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto A están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo. El puerto A también sirve para varias funciones especiales del ATmega164P como la Conversión Análoga Digital.

- **Port B (PB7:PB0)**

El puerto B es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up.

Las salidas de los buffers del puerto B tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto B están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo. El puerto B también sirve para varias funciones especiales del ATmega164P como se menciona en las páginas iniciales.

- **Port C (PC7:PC0)**

El puerto C es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto C tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto C están en tri-estado cuando las condiciones de reset están activadas siempre y cuando el reloj no esté corriendo. El puerto C también sirve para las funciones de Interfaz del JTAG, con funciones especiales del ATmega164P como se menciona en las páginas iniciales.

- **Port D (PD7:PD0)**

El Puerto D es un puerto bidireccional de entradas y salidas con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto D tienen características simétricas controladas con sumideros de fuentes de alta capacidad.

Los pines del Puerto D están en tri-estado cuando llega una condición de reset activa, siempre y cuando el reloj no esté corriendo.

El puerto D también sirve para varias funciones especiales del ATmega164P como se menciona en las páginas iniciales.

2.6 ORCAD

OrCAD es un software para automatizar el diseño de circuitos electrónicos, es uno de los buenos programas para la creación de circuitos. Su uso es sencillo

ya que es muy intuitivo: plasmar el circuito con sus componentes, y diseñar lo que se desee obtener.

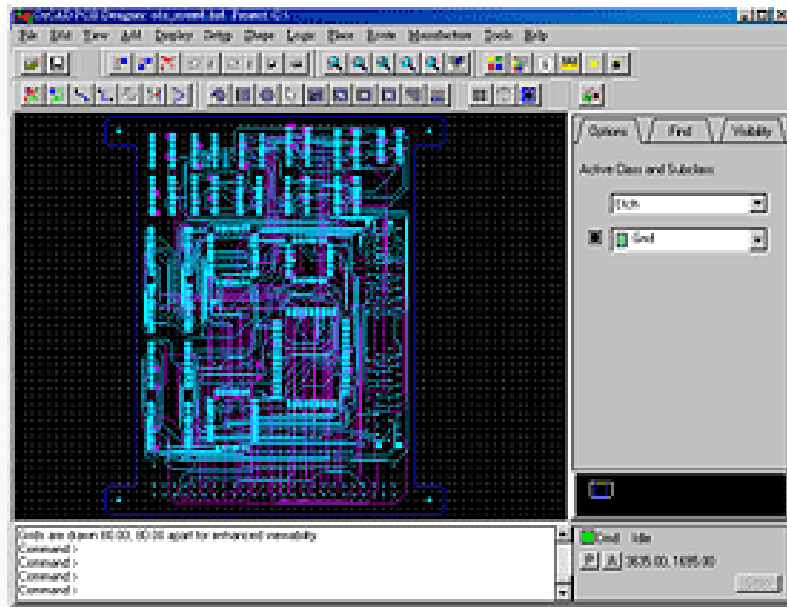


Figura II.12 Interfaz gráfica ORCAD

Ya sea para usuarios inexpertos que simplemente requieran funciones básicas o para expertos que necesiten altos grados de complejidad, este programa permite obtener los objetivos deseados.

Uno de sus puntos más débiles y con más complicación puede ser la utilización del Layout para el diseño de PCB, pero con un poco de paciencia y la ayuda necesaria se puede aprender a utilizar.

OrCAD es una herramienta informática integral, es decir, que integra muchas herramientas con las que hacer de todo en el mundo de la electrónica. Por ejemplo incluye:

- **Pspice:** el simulador analógico - digital –

- **Capture:** el esquemático para hacer todos tus circuitos, tanto digitales como analógicos
- **Layout:** Permite hacer "layouts" de placas, para insolarlas después
- **Editor:** editor de componentes

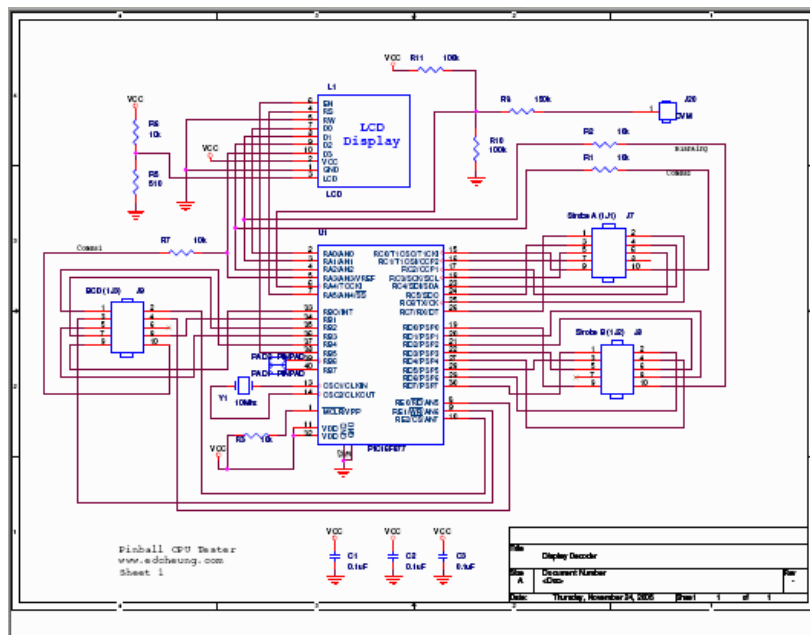


Figura II.13 Ejemplo del Capture

Además de la funcionalidad de todas las herramientas, dispone de un entorno CIS, que permite la intercomunicación entre todos los componentes, para ahorrar trabajo, y no se necesita ninguna otra herramienta para llevar a cabo un proyecto, desde el diseño a la complementación.

Específicamente el que se utiliza más es el Capture, ya que es más fácil de utilizar sin conocerlo muy a fondo. Aun así se recomienda la lectura de algún tutorial del programa o de la ayuda, ya que así se descubre muchas mejoras

para aumentar la productividad de los proyectos. Además todos los componentes tienen una potencia entendida como la capacidad de hacer muchas cosas interesantes. Como el Enrutamiento del Layout, que hace el trazado de pistas automáticamente ahorrando mucho trabajo.

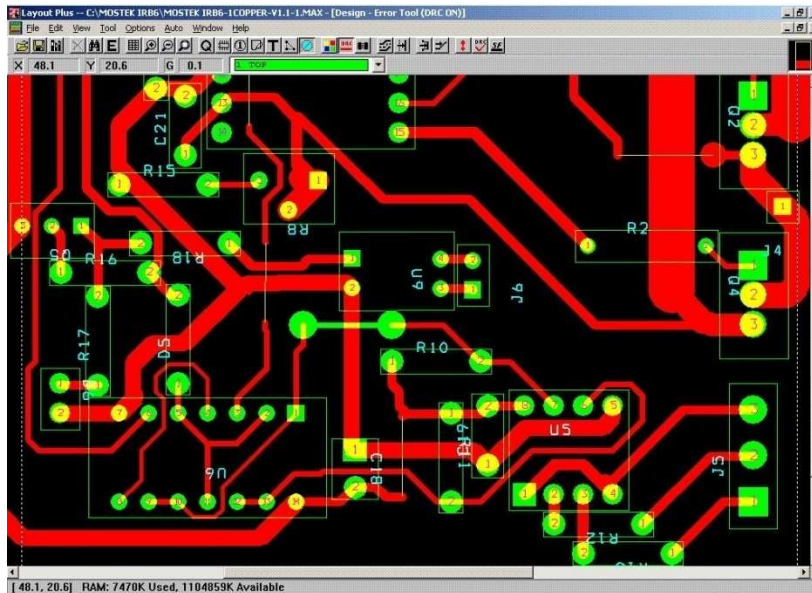


Figura II.14 Ejemplo del Layout

Y además de todos los componentes que tiene, incorpora una gran librería de componentes, que es de inmensa utilidad en el día a día de un Electrónico. Y se pueden añadir componentes a la librería, normalmente de las webs de fabricantes que ponen a disposición librerías suyas para orcad.

Todas las personas que se dedican a la rama eléctrica o electrónica es el programa que más se adapta a las necesidades de los mismos, una vez cogida experiencia tiene una gran facilidad de manejo para la creación de circuitos

eléctricos y electrónicos, con gran facilidad de manejo, grandes aplicaciones y gran cantidad de librerías es un programa indispensable para los que se dedican a esto.

2.7 BASCON AVR

BASCOM-AVR© es un compilador de BASIC para la familia AVR de ATMEL, desarrollado por la empresa Holandesa **MCS Electronic**.

Ha sido desarrollado sobre W95/98/NT y dispone de todas las características de la familia BASCOM

2.7.1 Características:

- **BASIC** estructurado con etiquetas.
- Programación estructurada con sentencias IF-THEN-ELSE-END IF, DO-LOOP, WHILE-WEND, SELECT- CASE.
- Generación de código máquina nativo en lugar de código interpretado.
- Bit, Byte, Entero, Word, Largo, y variables tipo String. (Solo con la Edición Profesional)
- Los programas compilados trabajan con todos los microprocesadores (no-MEGA) de AVR que tienen memoria interior. La edición Profesional apoyará la serie de MEGA también. Puesto que los 1200 no tienen SRAM, no funcionará con los 1200.

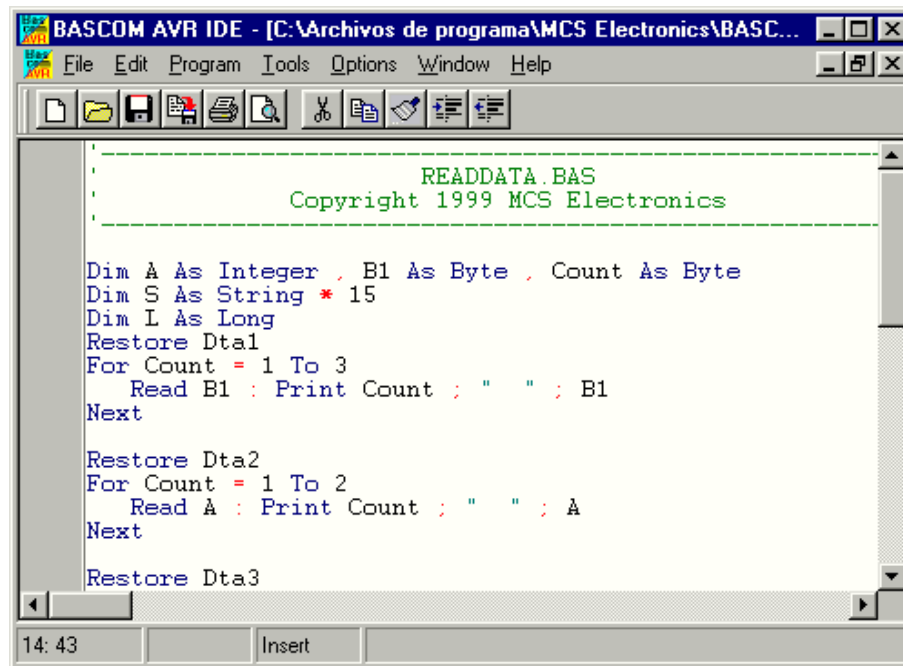


Figura II.15 interfaz Bascon AVR

- Las instrucciones y comandos de este software son bastante similares a las del Visual Basic y QuickBASIC de Microsoft.
- Comandos específicos para el manejo de displays **LCD**, integrados **I2C** e integrados **1WIRE Chips**, teclado de PC, teclado de matriz, recepción **RC5**, software UART. SPI, LCD Gráficos, envió de IR RC5 o código Sony.
- Soporta variables locales, uso de funciones, y librerías
- Emulador terminal integrado con opción de download.
- Simulador integrado.
- Programador de ISP integrado
- Integrado el soporte del programador STK200 y STK300. También soporta el Electronicsprogramme de bajo costo.

- Editor con subrayador de sentencias.
- Ayuda ON-LINE en el editor.

Soporta Las siguientes declaraciones y hay muchas más que se puede revisar en el archivo HELP:

2.7.2 Comandos e instrucciones:

De estructura y condicionales

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNT0, STEP, EXIT FOR, ON, GOTO/GOSUB, SELECT, CASE.

De entrada/salida

PRINT, INPUT, INKEY, PRINT, INPUTHEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, OPEN, CLOSE, DEBOUNCE, SHIF TIN, SHIF TOUT, GETATKBD, SPC

Funcionesnumérica

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD, LOG, EXP, SQR, SIN, COS, TAN, ATN, ATN2, ASIN, ACOS, FIX, ROUND, MOD, SGN, POWER, RAD2DEG, DEG2RAD, LOG10, TANH, SINH, COSH.

12C

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

1WIRE

1WWRITE, 1WREAD, 1WRESET, 1WIRECOUNT, 1WSEARCHFIRST,
1WSEARCHNEXT

SPI

SPIINIT, SPIIN, SPIOUT, SPIMOVE

Gestión de interrupciones

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE,
COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.

Manipulación de bits

SET, RESET, ROTATE, SHIFT, BITWAIT, TOGGLE.

Variables

DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT,
DEFBYTE, DEFINT, DEFWORD.

Varios

REM, SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB,
POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC,
MAKEBCD, INP,OUT, ALIAS, DIM , ERASE, DATA, READ, RESTORE, INCR,

DECR, PEEK, POKE, CPEEK, FUNCTION, READMAGCARD, BIN2GREY, GREY2BIN, CRC8, CRC16, CHECKSUM.

Directivas

\$INCLUDE, \$BAUD y \$CRYSTAL, \$SERIALINPUT, \$SERIALOUTPUT, \$RAMSIZE, \$RAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD, \$EXTERNAL, \$LIB.

Cadenas

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LTRIM, RTRIM, TRIM, LCASE, UCASE, FORMAT, FUSING, INSTR.

Y muchas otras funciones, declaraciones y directivas.

2.7.3 Como programar con el Bascom AVR:

Para realizar un programa con el BASCOM AVR es sencillo, por lo que se deberá realizar las siguientes operaciones:

- Escribir sobre el editor un programa en BASIC.
- Compilarlo a un eficaz código máquina nativo.
- Depurar el resultado con ayuda del simulador integrado, Estudio de AVR
- Programar el microcontrolador con el programador opcional.

El hardware opcional debe ser adquirido opcionalmente. Toda la documentación de este producto y sus periféricos están en inglés.

El programa puede ser escrito sobre un editor MDI intuitivo en color. Este editor, además de las características habituales, soporta Undo, Redo, marcado e indexación de bloques.

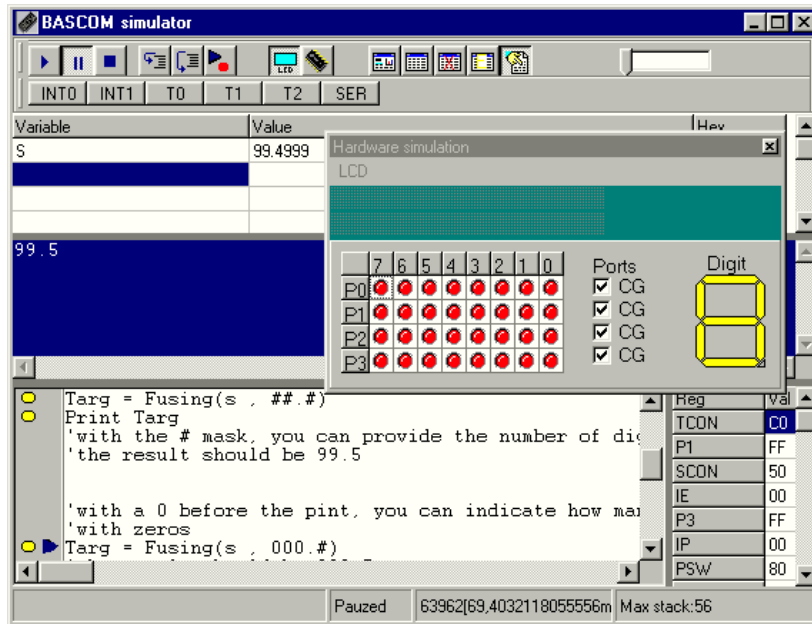


Figura II.16 Funciones especiales de Bascom AVR

El simulador le permite probar su programa antes de escribirlo al microprocesador.

Usted puede mirar variables, puede caminar a través del programa una línea en el momento o puede correr a una línea específica, o usted puede alterar variables.

Para mirar un valor de las variables usted también puede apuntar el cursor del ratón encima de la misma.

Un rasgo poderoso es el emulador del hardware, emula el LCD, y los puertos. Cuando ha terminado de probar el programa en el simulador, llega el momento de llevar el programa al microcontrolador.

CAPÍTULO III

HARDWARE

3.1 DISEÑO

Para el diseño de la tarjeta de control de la lavadora se tuvo muy en cuenta la estructura y arquitectura interna de un PLC básico, ya que según los requerimientos que se obtuvieron en el departamento de lavandería del hospital, se concluyó que para abarcar las salidas y entradas que la lavadora necesita y realización de los diferentes ciclos de lavado, serviría de mucho un PLC, pero es muy costoso, además que su función se podría emular con un microcontrolador, destinándolo a una función específica y un buen diseño integral de la tarjeta propuesta.

Se escogió implementar 10 entradas y 10 salidas, aunque sobren una que otra, es preferible tenerlas por si existe alguna entrada o salida que se necesite en un futuro o si se requiere implementar en otra lavadora con más funciones internas.

Por todo esto, la tarjeta de control para lavadora del hospital “Dr. Gustavo Domínguez Z.” de Santo Domingo de los Tsachilas está conformada por tres módulos (ver **Figura III.1**):

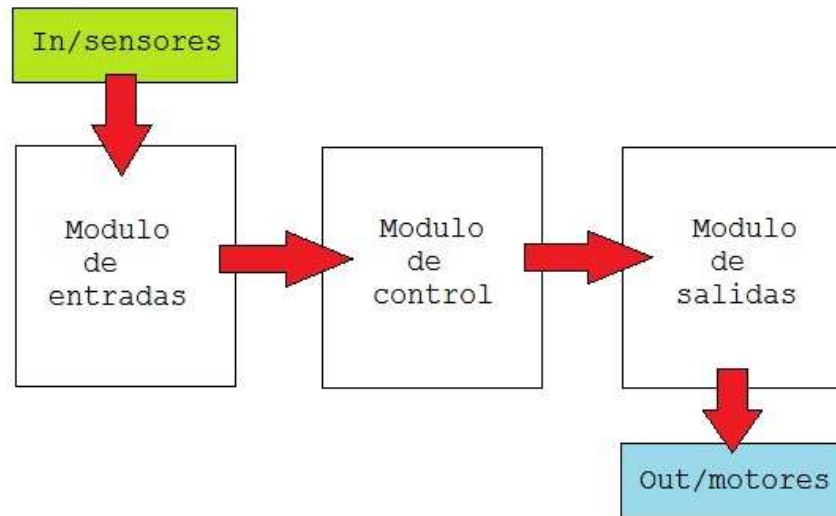


Figura III.1 Diagrama de bloques de la tarjeta

- Módulo de entradas
- Módulo de control
- Módulo de salidas

Cada módulo fue separado para ahorrar espacio físico y para que sea funcionalmente apreciable en el proyecto.

3.1.1 Módulo de entradas

Para el módulo de entradas se tomó la decisión de separar ópticamente el circuito de los diferentes sensores y pulsadores que existe en la lavadora, para

ello se utilizó optoacopladores en cada línea, teniendo así desacoplada ópticamente las dos partes, y sabiendo que si existe algún cambio de corriente o voltaje en la lavadora y por ende en los sensores, se tenga la certeza que no le sucederá nada al circuito gracias a las propiedades que tienen estos elementos.

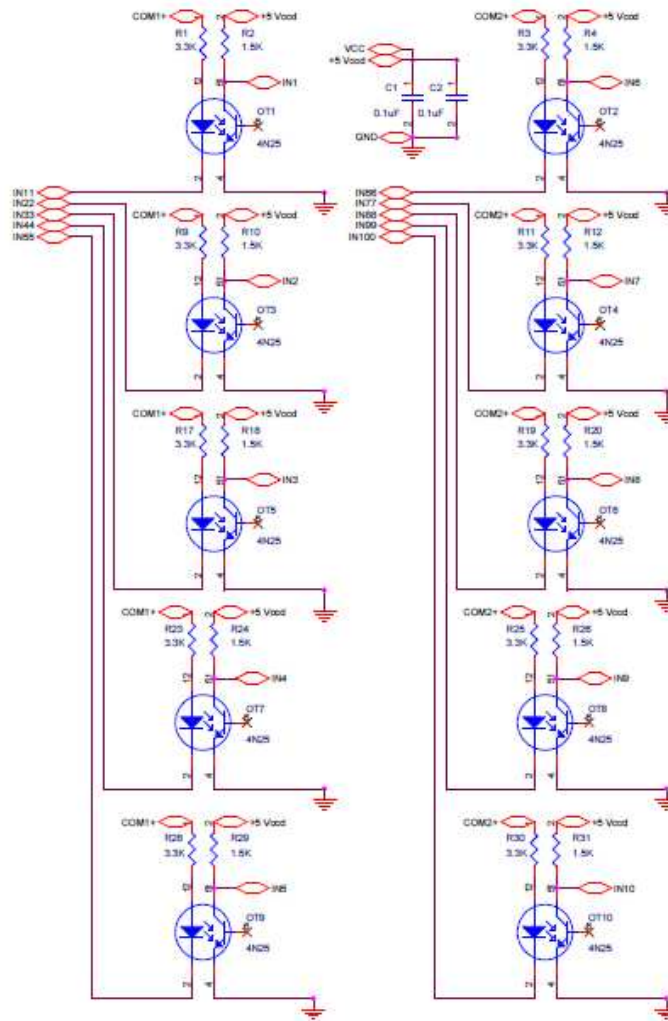


Figura III.2 Optoacopladores

También se dividió en dos grupos de 5 a las entradas, separándolo en dos *comunes* diferentes el *com 1* y el *com 2* que nos ayudara para poder conectar dos fuentes, una interna y una externa, es decir tener una fuente que sirva además de potenciador al circuito, también a los sensores, este sistema de dos fuentes es muy importante ya que se puede tener diversos voltajes que alimenten el circuito si así fuese necesario, en este caso solo necesitamos una fuente de 12 v, por lo que al unir el *com 1* y el *com 2* se tendrá un solo voltaje en el módulo de entradas sin necesidad de realizar cambios en el diseño, este formato de entradas es muy utilizado para prevenir futuros problemas de alimentación a la tarjeta .

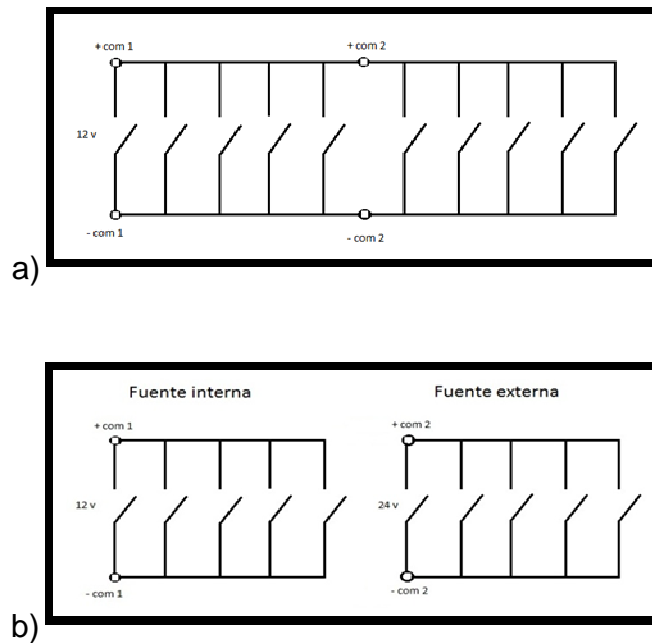


Figura III.3 a) Puente *común 1* y *común 2* b) *común 1* y *común 2* separados

Un aspecto muy importante para el funcionamiento y control de los diferentes ciclos de lavado son los leds indicadores que se pusieron en cada una de las entradas para mostrar que están actuando o que están apagadas, es decir que son 10 leds indicadores de color verde que van a encenderse para conocer qué entrada está funcionando.

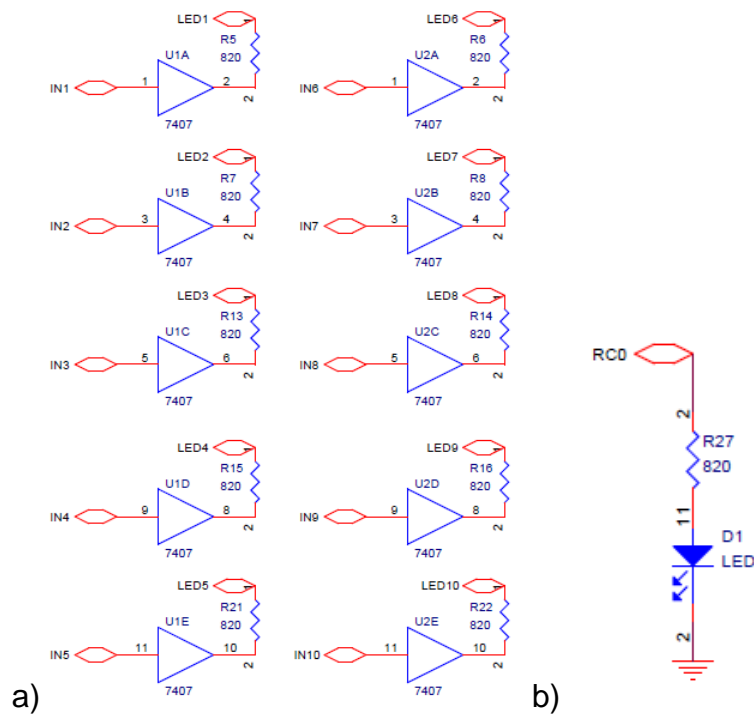


Figura III.4 a) Diseño de leds b) Led del módulo de entrada

Se ha puesto también un led de color blanco que muestra si está o no conectado el módulo de entrada con el módulo de control que le da la alimentación para que funcione, es decir que al prenderse el led se conoce que el módulo de entradas está conectado y funcionando.

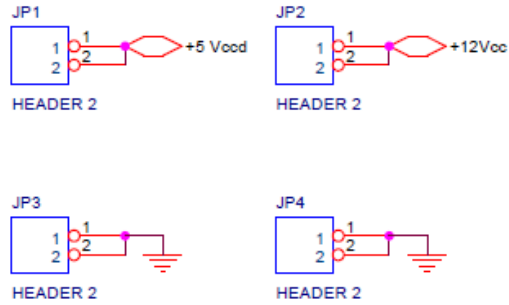


Figura III.5 Conectores de control

Además, para correcciones y pruebas de voltaje y de corriente se integró al módulo varios conectores, en los cuales se pueden incorporar algún multímetro, tomando datos de variación y evaluación del módulo de forma fácil y rápida.

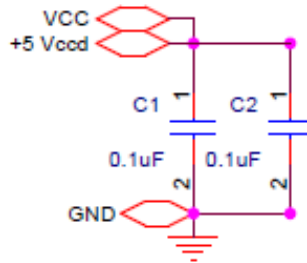


Figura III.6 Capacitores de adecuación

Los capacitores puestos en el módulo ayudan a disipar el ruido de alta y media frecuencia que emiten los elementos del módulo debido a la velocidad de ejecución de las instrucciones.

3.1.2 Módulo de control

El módulo de control es la parte más importante de la tarjeta, ya que aquí está ubicado el microcontrolador que hará funcionar a la lavadora según los ciclos de lavado que se necesite, el sistema de comunicación y el LCD que mostrara lo que está realizando la lavadora.

El microcontrolador es el Atmega164p de la fábrica AVR, este integrado es de los más completos que existe en el mercado. Se ha ocupado casi todos los pines del chip ya que son 20 pines entre entradas y salida, y los demás para la comunicación, el LCD, fuente, el ICSP, el cristal, el reset y otros para comprobación.

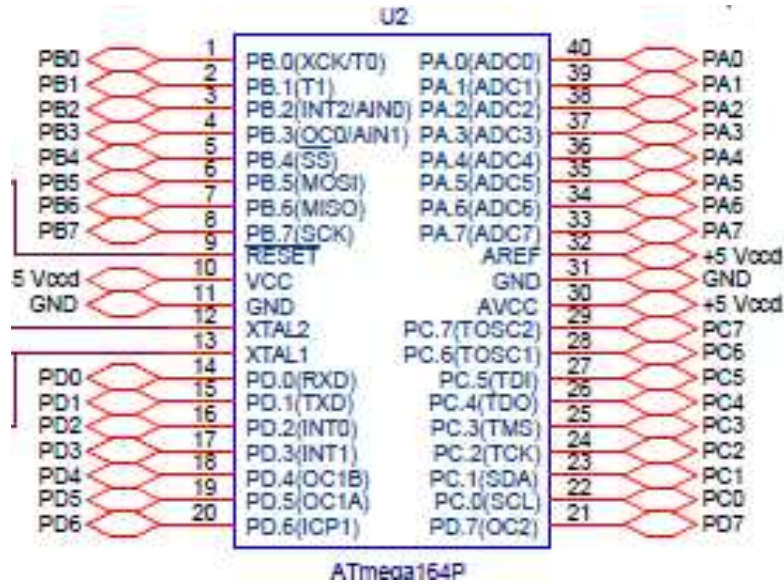


Figura III.7 Microprocesador Atmega164p

Para que el microcontrolador funcione correctamente es necesario tener tres partes conectadas y diseñadas correctamente que son:

- La fuente
- El cristal
- El reset

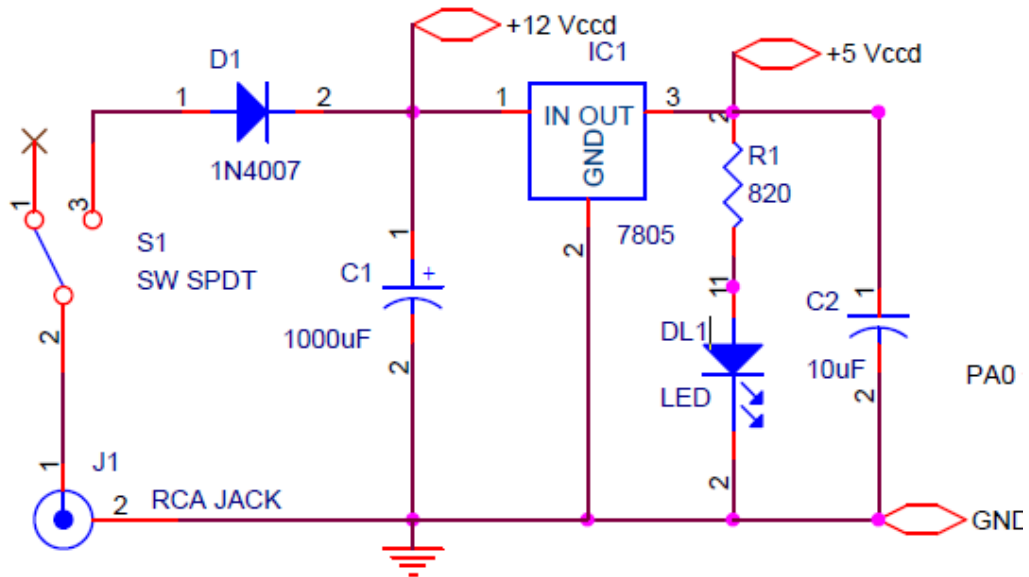


Figura III.8 Alimentación de la tarjeta

La fuente, alimenta el microcontrolador y está diseñado para que no cambie bruscamente de voltaje, cambiando de 12V a 5V al utilizar el integrado 7805, además, se requirió de un led indicador para conocer si está alimentando o no al microcontrolador, otro aspecto muy importante es conocer también la corriente que entra en el jack porque debería potenciar al microcontrolador, al LCD, a los relés, a los optoacopladores y a los demás elementos, es por eso que mínimo se va a colocar un fuente que tenga 2A en la fuente, además el diseño puesto, trata de no causar fallas en el microcontrolador, realizando un cambio de fuente al irse bruscamente la luz ya que tiene una batería que

alimenta a la tarjeta cuando existan cortes de electricidad, hay que recalcar que es solo para la tarjeta y no para la lavadora.

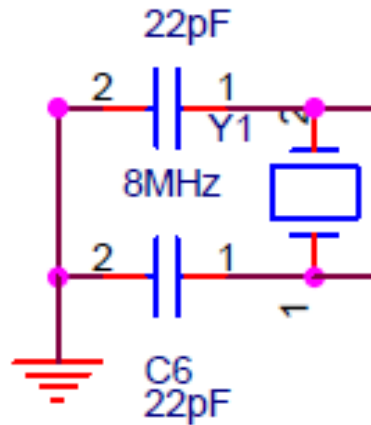


Figura III.9 Conexión del cristal externo

El cristal, donde se utilizó el datasheet para conectarlo, es decir con dos capacitores para que regule su velocidad. El cristal es muy importante también, aunque este elemento se pudo obviar en el módulo, porque el microcontrolador tiene un cristal interno, pero la experiencia al trabajar con ese cristal, sustentado en varias pruebas, indicaba que al utilizar los módulos de comunicación se tendría un retraso en el envío y recepción de los datos, es por eso que se colocó un cristal externo, dando así una mejor precisión al microcontrolador y obteniendo buenos tiempos de procesamiento. Otro aspecto que se debe tomar en cuenta es la temperatura del medio ambiente donde vaya a implementarse el modulo, ya que el cristal interno que tiene el microcontrolador varía según la temperatura del ambiente y como va a funcionar en Santo Domingo de los

Tsachilas que es una ciudad calurosa y húmeda es importante poner un cristal externo.

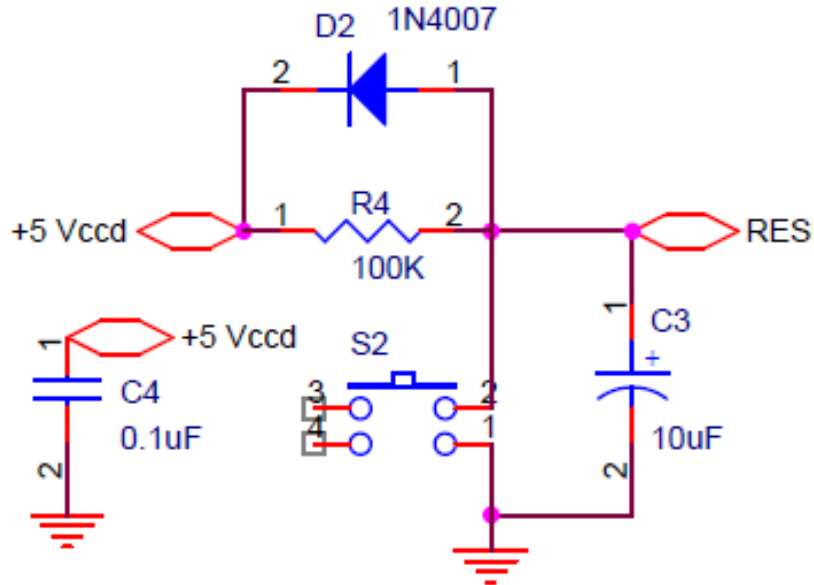


Figura III.10 Conexión del reset

El reset, es utilizado para casos especiales donde el microcontrolador realiza instrucciones indefinidas fuera del procedimiento o ciclos sin fin, además es importante para el ICSP ya que usa el reset para poder funcionar. El diseño del reset es sistemáticamente básico ya que está compuesto de un diodo, una resistencia, un capacitor y un pulsador, estos es para que no haya retrasos o saltos innecesarios en el pulsador y envíe de forma normal el 1 lógico del reset al microcontrolador. El reset va a estar ubicado dentro del módulo de control y no al alcance de las personas, será usado solo en caso de emergencia.

Además de las partes anteriores indicadas, el módulo de control tiene otros elementos que es trascendental realizar una explicación:

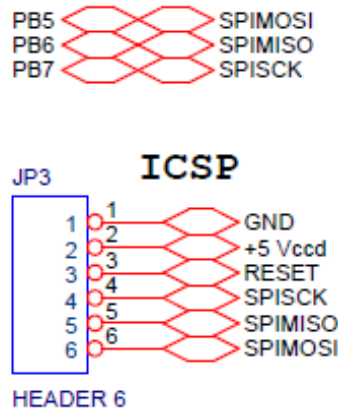


Figura III.11 Programación serial en el circuito

El ICSP, que significa In Circuit Serial Program, programación serial en el circuito, es decir que con este sistema se puede programar al microcontrolador sin retirarlo del módulo y así fácilmente conectar, programar, quemar y probar. Esta forma de programación es utilizada para los microcontroladores de montaje superficial o sea los microcontroladores pequeños que deben ser soldados a la placa y que es imposible sacarlos para programarlos.



Figura III.12 Programador

La programación serial en el circuito cuenta con un programador adaptado para que se conecte al módulo y de allí con el puerto USB del computador.

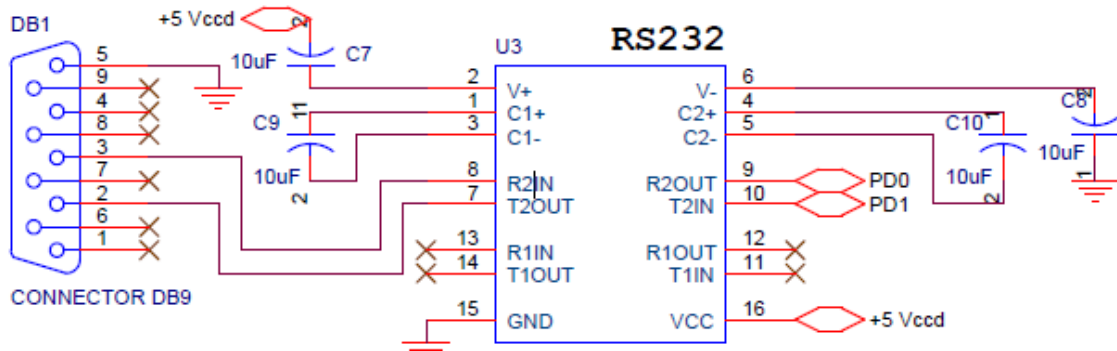


Figura III.13 Comunicación PC - Microcontrolador

El sistema de comunicación está basado en las diferentes instrucciones que el programa tiene, como en qué velocidad trabajar y que palabra reservada de Bascom utilizar para realizar el intercambio de información, conociendo también que el hardware está compuesto de una interfaz que comunica el microcontrolador con la computadora que es el MAX232 que como se vio en el

capítulo 3, es utilizado para enviar en forma serial la información, intercambiando voltajes para que cada terminal pueda reconocer esa información ya que la comunicación va de $\pm 5V$ a $\pm 12V$, además el datasheet explica cómo debe ir la conexión y que elementos se deben incorporar al sistema como son los capacitores que se ven en la figura III.13 .

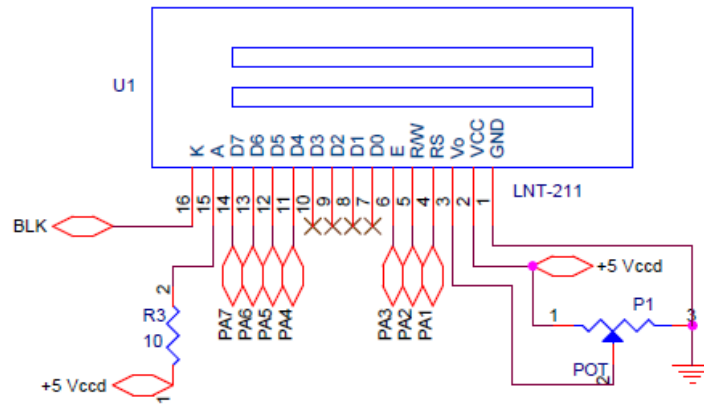


Figura III.14 Configuración LCD

El LCD es el único periférico de salida que tiene la tarjeta de control ya que nos va a mostrar los ciclos que la lavadora está realizando, este elemento es el primero en ser configurado y declarado en el programa por su importancia y por sus características especiales, dándole un tiempo de 500ms para que se inicialice sin errores. El LCD está conectado al microcontrolador por 7 pines y los demás pines al voltaje y para la configuración física.

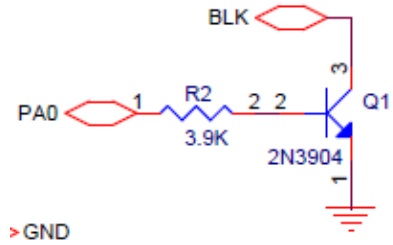


Figura III.15 Back light Key

Para el ahorro de energía en la batería, el LCD tiene un pin llamado BLK que es el Back light, utilizado para apagar la luz de fondo del LCD, este proceso de apagado se realiza mediante software aunque se podría hacerlo con un pulsador, apagando la luz de fondo cuando el usuario lo requiera.

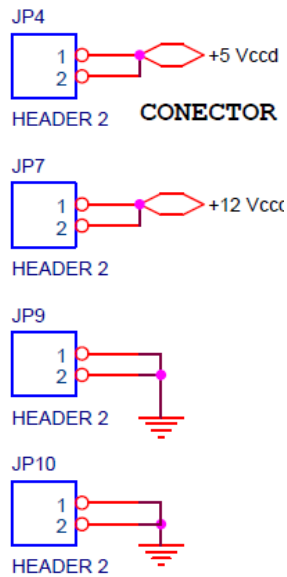


Figura III.16 Conectores de control

Así como en el módulo de entradas, en el módulo de control también se pusieron conectores para corrección y prueba de voltaje y corriente.

3.1.3 Módulo de salidas

Las salidas, así como las entradas también están desacopladas a todo el circuito gracias a los relés, que tienen la propiedad de separar magnéticamente un elemento de otro, es decir que al requerir de una salida el microcontrolador enviara al relé un orden de interrupción o paso, para que funcione o se apague una de las salidas de la lavadora.

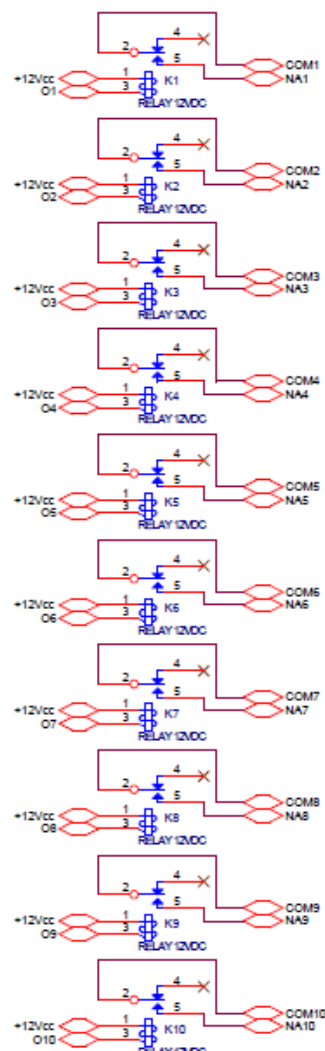


Figura III.17 Configuración de Relés

Los relés se alimentan de 12V cada una, son 10 relés para las 10 salidas.

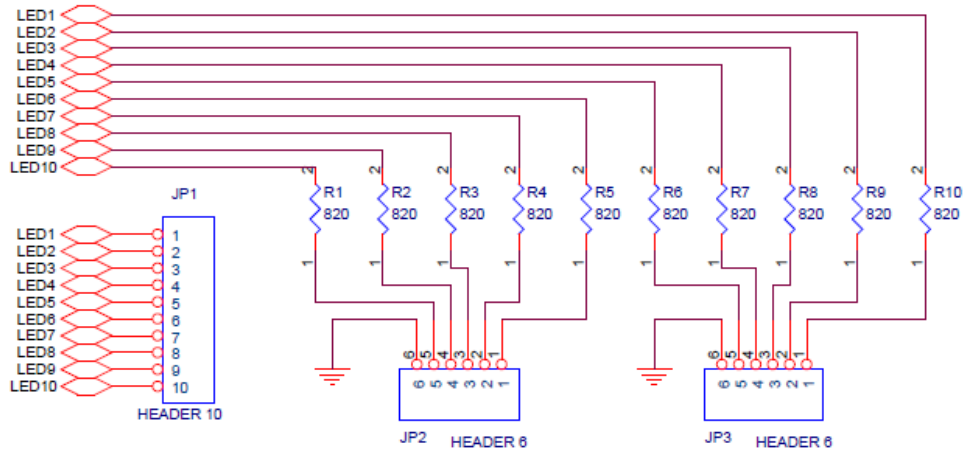


Figura III.18 Configuración de los leds de salida

Para conocer qué salida está activa se agregó un led para cada una de ellas, prendiéndose o apagándose según el ciclo que esté realizando la lavadora. Esta función de la tarjeta en las entradas y en las salidas se pensó, para una mayor comprensión de los usuarios, logrando así un equipo amigable para cualquier persona que quiera saber el funcionamiento de la lavadora.

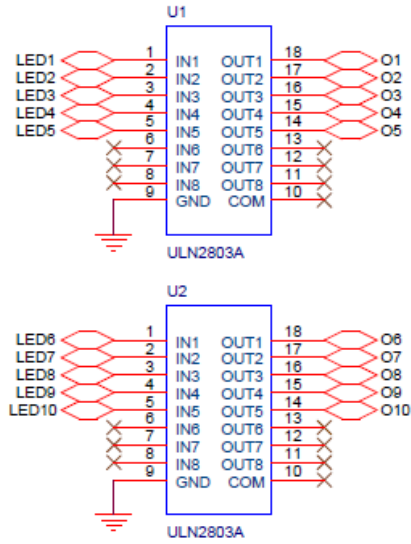


Figura III.19 Integrados ULN2803A

Para que los leds trabajen de una manera óptima se conectó a los relés dos integrados ULN2803A, que en su estructura interna están compuestos de inversores y transistores, ayudando a que no se acumule el módulo de muchos elementos iguales, ocupando poco espacio y minimizando el ruteado de la tarjeta.

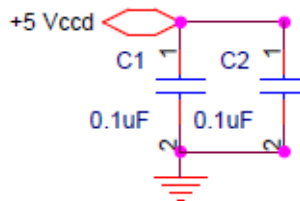


Figura III.20 Capacitores de adecuación

Los capacitores utilizados para disipar el ruido de alta y media frecuencia debido a las altas velocidades de procesamiento de la información, también se agregaron en el módulo de salidas.

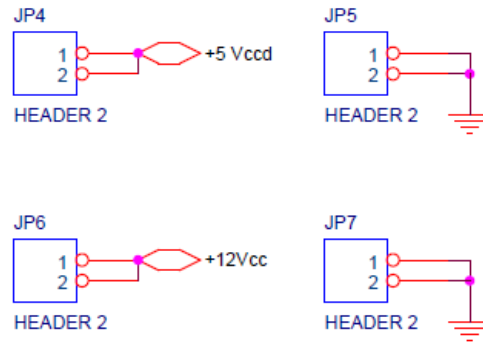


Figura III.21 Conectores de control

Otro elemento que del mismo modo se incorporó al módulo de salidas al igual que al módulo de entradas son los conectores para comprobación de voltajes.

3.1.4 Estructura física

La ubicación de los módulos se realizó siguiendo el mismo esquema físico de un PLC, donde las entradas y las salidas están ubicadas de manera frontal para que no se confundan entre sí, en la mitad de ellos va a estar ubicado el LCD con los leds indicadores de entradas y salidas y otros tres leds incorporados para mostrar el prendido del módulo de entradas, otro del módulo de salidas y el ultimo del encendido de toda la tarjeta cada uno con su respectivo color.

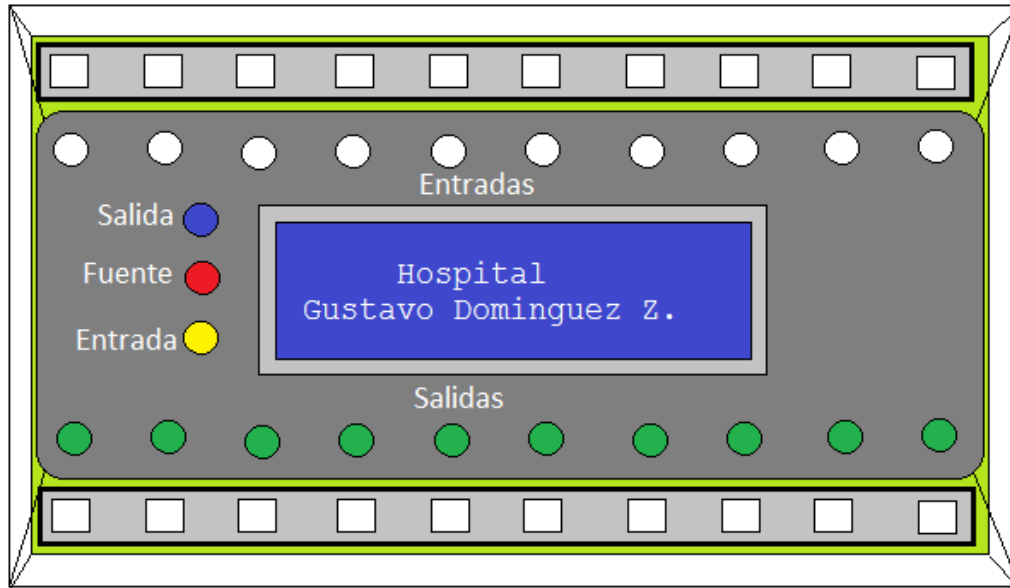


Figura III.22 Prototipo de la Estructura física

En la parte lateral izquierda de la tarjeta estará un interruptor para prender o apagar la tarjeta de control, el jack para conectar la fuente de 12V y la batería, mientras que en la parte lateral derecha estará el conector DB9 para la comunicación

Se pondrá esta tarjeta dentro de una caja metálica de protección para su fácil movilización y adecuación en la lavadora

3.2 IMPLEMENTACIÓN

Para la implementación de la tarjeta de control se planifico realizar de manera continua los tres módulos en el siguiente orden:

- Ruteo

- Atacado
- Perforado
- Montaje
- Pruebas
- Protección

3.2.1 Ruteo

El ruteo se generó en el programa ORCAD y en sus diferentes aplicaciones como el Capture y el Layout, ordenando los elementos de cada módulo para que se ubiquen ahorrando espacio y adaptándose a los conectores de las salidas y de las entradas, en especial con el módulo de control que va encima de los otros dos módulos.

Se trató de ahorrar espacio pero se complicó al adquirir conectores para los calles de entradas y salidas de la lavadora ya que dichos conectores industriales, seguros y resistentes, ocupan un buen espacio en la tarjeta dando un tamaño considerable

3.2.2 Atacado

La impresión de este tipo de circuitos en la placa no se realizó con marcadores o con cinta negra (también llamado PrestTape) ya que es muy complicado y no queda bien el trabajo logrado, esta vez se experimentó con un método muy conocido utilizando un papel llamado, Papel de Transferencia para circuito impreso, o también conocido como, Hoja Azul para circuitos impresos.

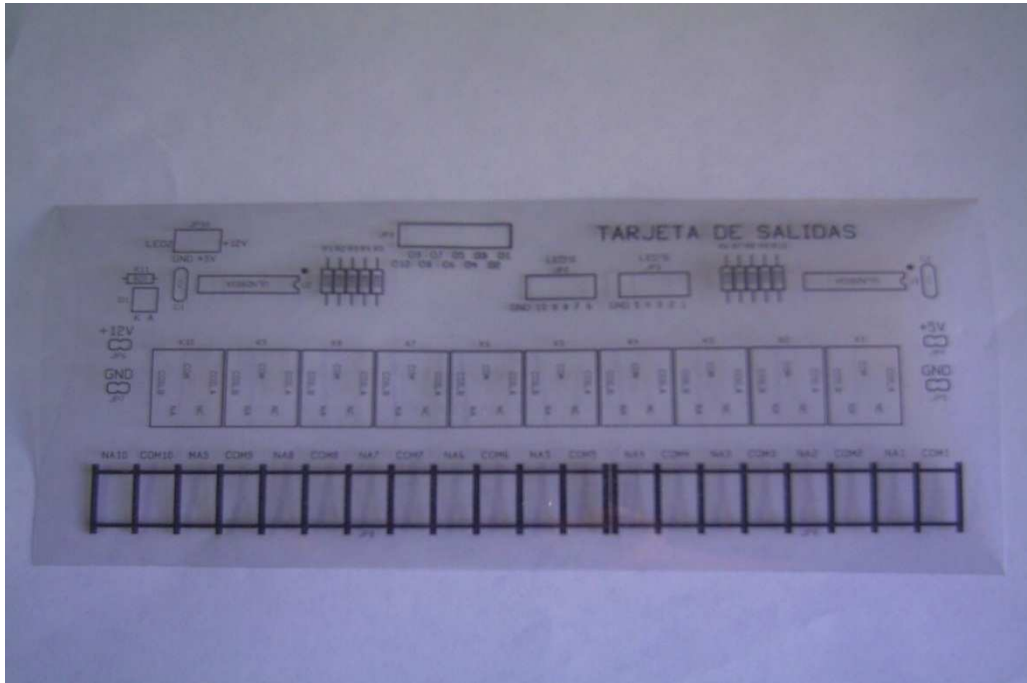


Figura III.23 Formato de plaqueta

Esta hoja es lisa y brillante de un lado y media áspera del otro, el circuito se imprimió por la *parte áspera*, y se usó una impresora láser, en otro tipo de impresora no se debería utilizar este papel, debido a que la impresora laser tiene tonner y las demás no.

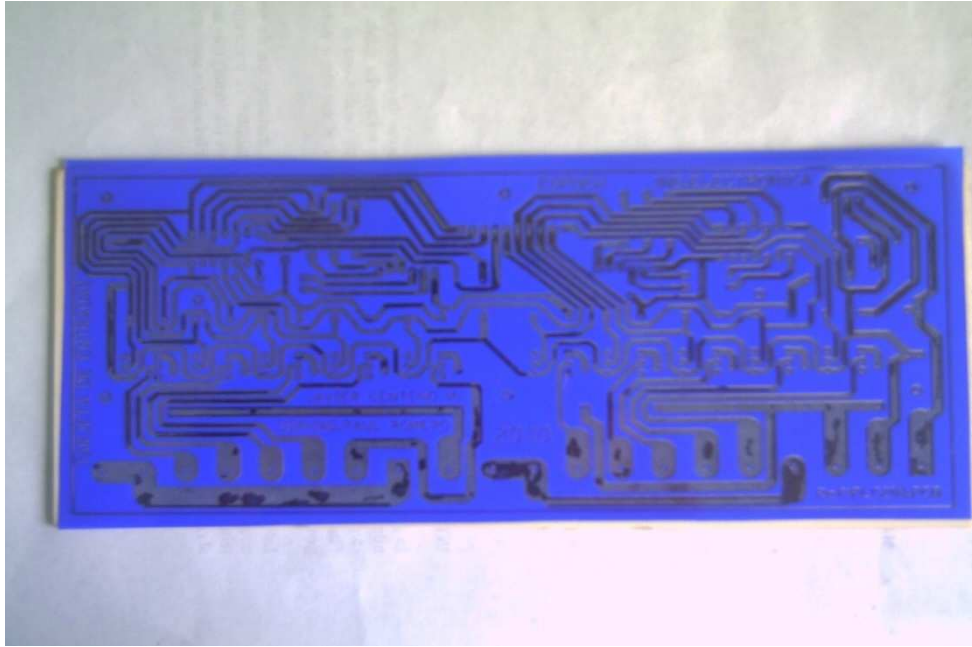


Figura III.24 Hoja azul - Atacado

Hay que tener en claro algo, que se debe planchar el papel azul con la placa de manera *uniforme*, de lo contrario las pistas no pasaran en forma pareja , y si existe algún error se debe corregir con marcadores pero solo para las imperfecciones, teniendo mucho cuidado al hacerlo y evitando que las pistas se toquen porque pueden causar cortocircuito.

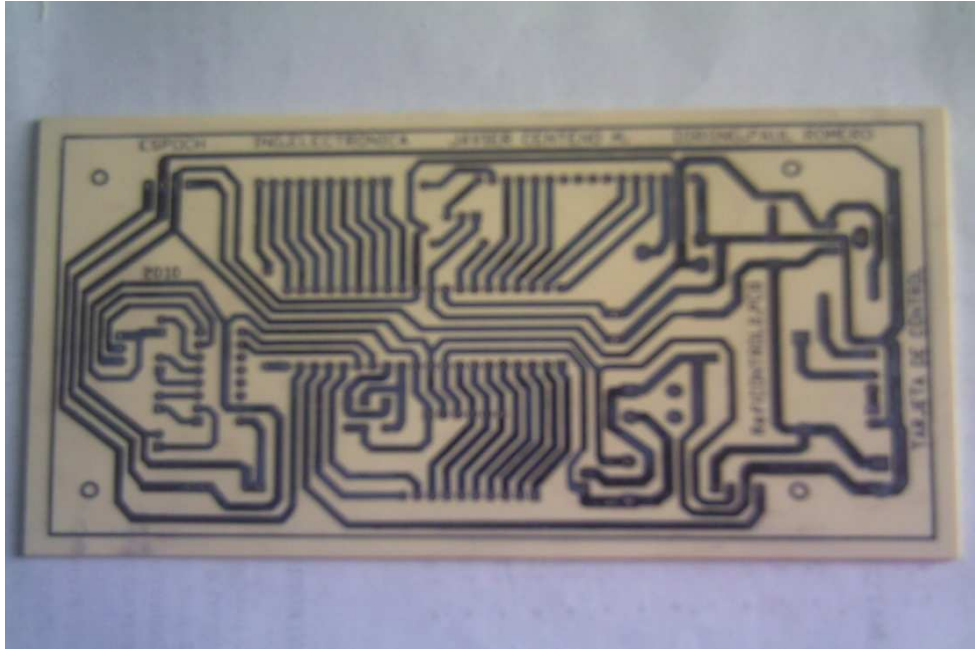


Figura III.25 Plaqueta

A continuación las plaquetas se introdujeron en ácido (percloruro de hierro o ácidoclorhídrico) para que el cobre se disuelva y quede solo las pistas o islas del circuito, además hay que tener claro que para que el ácido funcione correctamente y pueda actuar sobre el cobre debe estar a una temperatura comprendida entre 20 y 50 grados centígrados.

Una vez que el ácido atacó todas las partes no deseadas del cobre se sacaron las 3 plaquetas de la batea, se les colocó en un recipiente lleno de agua y se dejó durante 10 minutos. Luego, se secó y se quitó la impresión con solvente. Al finalizar se necesitó pulir suavemente con viruta de acero cada una de las placas para que queden perfectamente definidas y sin errores.

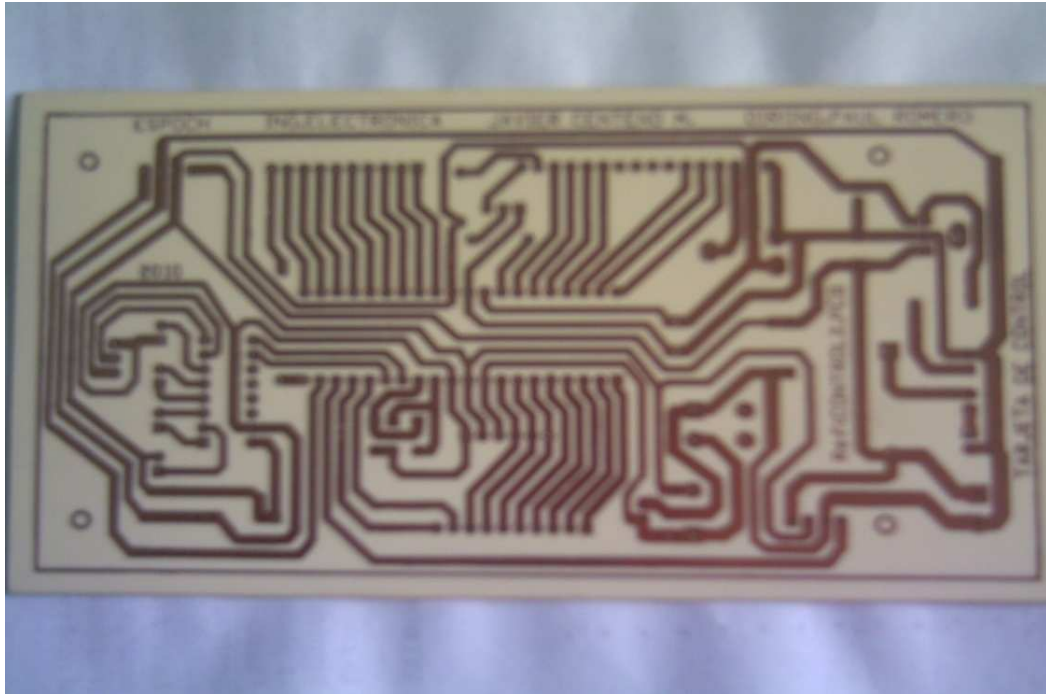


Figura III.26 Plaqueta con las pistas limpias

3.2.3 Perforado

Para que los componentes puedan ser soldados se hicieron orificios en las islas por donde el terminal del componente pasará.

Las perforaciones o vías del circuito se las realizo con un taladro estacionario de broca muy fina y pequeña, ideal para trabajos donde la precisión es lo fundamental.

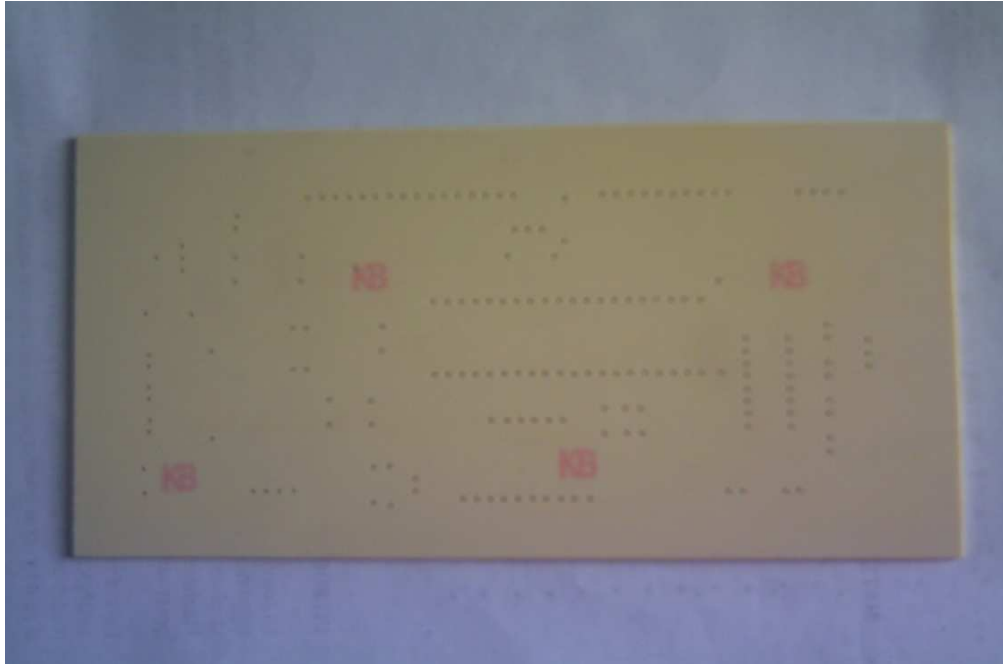


Figura III.27 Perforaciones

Para los orificios de resistencias comunes, capacitores y semiconductores de baja potencia se utilizó una mecha (broca) de 0.75mm de espesor. Para orificios de bornes o donde se soldara espadines o pines se cambia a una broca de 1mm que es adecuada para el tamaño. Aquí además fue de suma utilidad añadir varios orificios centrales y en los bordes de la plaqueta para que quede la hilera de perforaciones lo más pareja que sea posible y se pueda sujetar el módulo de control que va encima de los dos otros módulos.

3.2.4 Montaje



Figura III.28 Elementos necesarios

En el montaje de los elementos de cada módulo se siguió la regla de oro, que es montar primero los componentes de menor espesor, comenzando por los puentes de alambre que fueron muy pocos. Luego se continuó con los diodos, resistencias, pequeños capacitores, transistores, pines de conexión y zócalos de circuitos integrados.

Siempre es bien visto montar zócalos para los circuitos integrados puesto que luego, cuando sea necesario reemplazarlos en futuras reparaciones será un simple quitar uno y colocar otro sin siquiera usar soldador. Además, el desoldar y soldar una plaqueta hace que la pista vaya perdiendo adherencia al plástico y

al cabo de varias reparaciones la isla sede al igual que las pistas que de ella salen.



Figura III.29 Montaje

3.2.5 Pruebas

A la tarjeta sin componentes se le sometió a pruebas al desnudo, donde se verificó cada conexión definida en el netlist en la tarjeta finalizada, es decir que se comprobó por última vez la continuidad eléctrica de las pistas y se reparó lo que fue sea necesario.

Después del montaje se jalo cada elemento para asegurarse que están bien soldados a la plaqueta.



Figura III.30 Prueba con multímetro

Otras pruebas efectuadas en la tarjeta después del montaje fueron las de funcionamiento del microcontrolador, LCD y comunicación, mediante impulsos eléctricos en cada elemento y revisando en el multímetro los resultados obtenidos.

3.2.6 Protección

A todas las plaquetas se les aplicó laca para sellar las pistas, protegiéndolos de los roces causados y previniendo alguna fisura en las islas o desprendimiento del cobre.

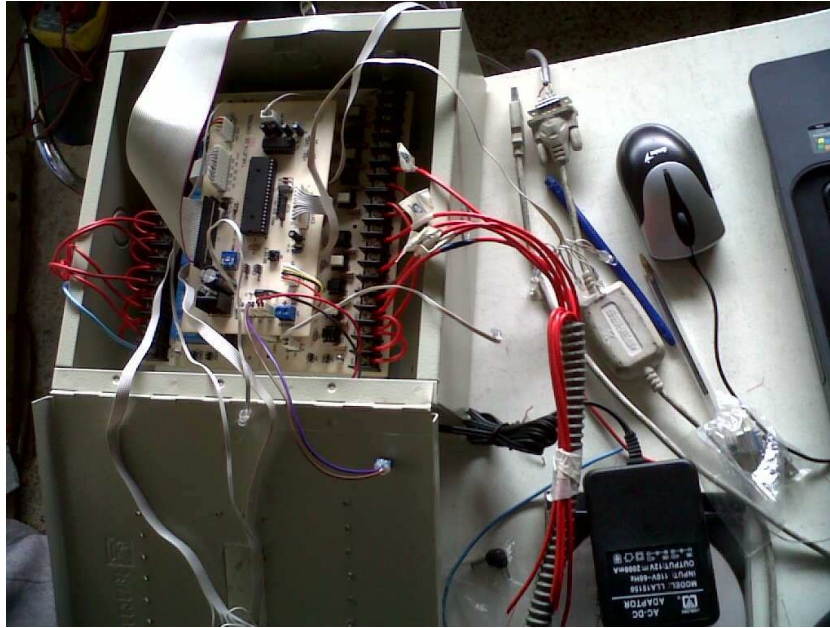


Figura III.31 Tarjeta protegida

La protección más importante fue adecuar una caja impermeable donde se introdujo la tarjeta, facilitando su transportación y preservándola de cualquier descuido o imprevisto que pudiera ocurrir cuando la tarjeta ya sea parte de la lavadora.

CAPÍTULO IV

SOFTWARE

4.1 Aspectos en el Programa para que funcione el Hardware

Antes de empezar a programar las pequeñas rutinas de prueba de hardware o el cuerpo del software, se debe tener en cuenta las diferentes instrucciones que tiene Bascom para inicializar los dispositivos que se va a utilizar dentro de la tarjeta, como el LCD, el microcontrolador, o la interfaz de comunicación.

El LCD tiene un pin llamado R/W el cual es utilizado para que recibir o enviar información, además sirve como señal de control, es por esto que al LCD le enviamos un cero lógico haciendo funcionar esa señal de control con la instrucción *\$initmicro*, otro pin que en la programación se le da importancia es el BLK, ya que se debe enviar un uno lógico para que prenda la luz de fondo del LCD, por todo esto es necesario realizar una configuración por software también al sector de comunicación, enviándole el valor de la velocidad que debe trabajar y realizando una interrupción por software

cuando se requiera realizar tal acción, para ello está la instrucción $\$Baud$ que para la tarjeta se le dio por ejemplo $\$Baud=9600$ donde trabajara a un velocidad de 9600 Hz, y para finalizar se le da un tiempo estimado de 500 ms al LCD para que el encendido de este, se ejecute sin fallas.

Para inicializar el microcontrolador es necesaria la instrucción $\$initmicro$ que ya se comentó anteriormente y que también es utilizado para el microcontrolador funcione, además se deberá configurar el modelo del elemento, escribiendo en el programa que va dentro del integrado el nombre exacto del microcontrolador para que se ejecute de manera correcta.

En resumen:

Tabla IV.I. Instrucciones de inicialización

Instrucción	Valor	Descripción
$\$regfile$	m164pdef.dat	Definición del microcontrolador
$\$crystal$	8000000	Velocidad del cristal
$\$baud$	9600	Velocidad de conexión
$\$initmicro$		Iniciar microcontrolador
Ddra.7	1	Pórtico A.7 como salida

4.2 Programas de prueba de hardware

Terminada la implementación de los módulos y conectados como corresponde cada uno, se realizaron varios programas pequeños de prueba para conocer el funcionamiento que debe realizar los principales elementos de la tarjeta como el LCD, las respuestas del microcontrolador, la comunicación y los leds de entradas y salidas con sus optoacopladores y relés respectivamente.

El primero de los programas fue para mostrar en el LCD frases y comprobar el funcionamiento de las entradas y salidas, activando la entrada uno para accionar el relé uno y por ende el led de la salida uno, y así sucesivamente con las demás entradas y salidas (ver anexo C), demostrando la visualización del LCD, el funcionamiento del microcontrolador, la manera de programar con el ICSP y la velocidad y forma de trabajar de los elementos de las entradas y salidas.

En otra instancia de las pruebas, se incorporó al primer programa varias sentencias de control a la interfaz de comunicación, ya que además de los procesos anteriores se quiso cargar más información al microcontrolador y obtener los mismos resultados. Las sentencias añadidas fueron para que al conectar la tarjeta a la computadora mediante el puerto serial y digitar el número uno (1), en la pantalla del LCD se visualice la palabra "HOLA MUNDO" y si se digitaba el número 2, aparezca "BIENVENIDOS" (ver

anexoD), teniendo así comprobada la comunicación entre la tarjeta y la computadora.

El último programa de pruebas, fue la emulación de pequeños ciclos de lavado en cortos periodos de tiempo, teniendo en cuenta, varias entradas y salidas de la tarjeta y sabiendo que algunas entradas serán sensores que van a prenderse o apagarse según el ciclo donde se encuentre funcionando la lavadora, además se realizó las primeras pantallas de presentación en el LCD, visualizadas al encender la tarjeta.

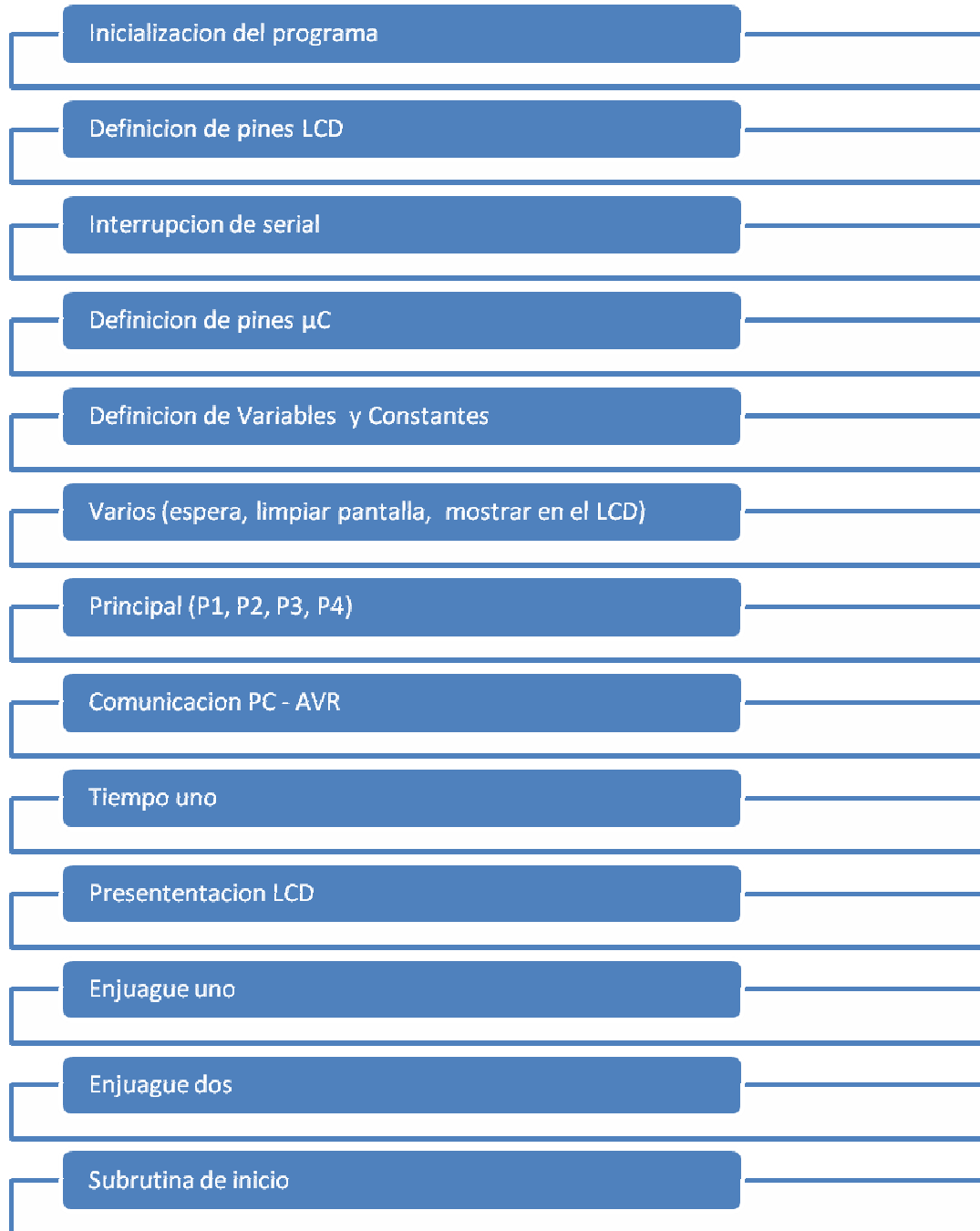


Figura IV.1 Contenido del programa

4.3 Contenido

El contenido del programa muestra una manera fácil de entender la estructura de programación, desde donde empieza, porque instrucción pasa y en que instrucción finaliza, la figura IV.1 indica mediante un gráfico cuales son estas instrucciones.

4.4 Subrutinas

Dentro del programa se creó varias subrutinas para reducir el tamaño de la programación y ahorrar espacio en el microcontrolador. Las cuales se desarrollaron sin orden alguno, ya que el microcontrolador puede leerlas estén al principio, en el medio o al final, no importa el orden, al igual que su uso.

La siguiente tabla describe las subrutinas existentes:

Tabla IV.II Subrutinas

Rutina	Descripción
Interrupción serial y comunicación serial	Realiza el stop del proceso en el Microcontrolador y comunica el PC con la tarjeta
Varios (espera, limpiar pantalla, mostrar en el LCD y subrutina de inicio)	Engloba varias subrutinas como de espera, limpiar pantalla del LCD y proceso de imprimir en pantalla las

	opciones del lavado
Tiempo uno	Instrucciones para el Stop de procesos
Tiempo dos	Instrucciones para el Stop de procesos sin llenado de agua
Presentación LCD	Imprime en el LCD la información del proyecto
Enjuague uno y dos	Procesos de lavado

4.5 Cuerpo del programa

El cuerpo del programa se divide en 4 partes según el tipo de lavado que se requiera realizar, resultando cuatro tipos de ciclos existentes y por ende 4 pulsadores, además de un pulsador adicional que es el de Stop (Parar).

Estas partes están conformadas por sentencias de comparación como el *ifthenelse*, en cuya igualación si es verdadera entrara dentro del proceso y efectuara solo la acción que se requieran en determinado ciclo teniendo en cuenta además varias banderas (flag) que nos servirán de respaldo si no ejecutara bien alguna instrucción.

En la figura IV.2 se describen los diferentes ciclos que realiza cada proceso de lavado al presionar un pulsador donde igualmente aparecerá en el LCD de la tarjeta.

P1 (Pulsador 1)	P2 (Pulsador 2)	P3 (Pulsador 3)	P4 (Pulsador 4)
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Enjuague 1	<input type="checkbox"/> Enjuague 1	<input type="checkbox"/> Enjuague 1	<input type="checkbox"/> Lavado
<input type="checkbox"/> Enjuague 2	<input type="checkbox"/> Lavado	<input type="checkbox"/> Lavado	<input type="checkbox"/> Centrifugado
<input type="checkbox"/> Lavado	<input type="checkbox"/> Enjuague 2	<input type="checkbox"/> Centrifugado	
<input type="checkbox"/> Enjuague 3	<input type="checkbox"/> Centrifugado		
<input type="checkbox"/> Centrifugado			

Figura IV.3 Contenido de cada proceso de lavado

El enjuague 1 en cualquier proceso iniciado por algún pulsador tiene las mismas características, ya que dentro de la programación: revisa el sensor de nivel de agua, si está encendido abre las válvulas de agua fría y caliente, activa el motor de giro de izquierda a derecha, después de un tiempo cambia el giro con un lapso de tiempo entre cambio y cambio, el giro de izquierda a derecha y de derecha a izquierda suman un ciclo que es el que varía según el enjuague realizado: el enjuague 1 tiene 5 ciclos, el enjuague 2 tiene 10 ciclos, el enjuague 3 tiene 7 ciclos y el lavado tiene 15 ciclos. El centrifugado o también llamado exprimido es el aumento de la velocidad del motor para que drene toda el agua que está en la lavadora y por ende en la ropa, este es el último paso que realiza la tarjeta y que es común en todos los procesos.

4.6 Interfaz de comunicación PC-Microprocesador

La interfaz de comunicación PC-Microprocesador es un programa sencillo para modificar los tiempos de ciclo de lavado realizado en Visual Basic 6.0.



Figura IV.4 Pantalla Inicio de sesión

Funciona conectando el cable RS-232 (DB 9) a la tarjeta y a la computadora. Al iniciar el programa es necesario introducir una clave de protección para uso exclusivo de personas que sepan el funcionamiento del software (ver **Figura IV.4**), ya en la ventana principal se tiene tres bloques el primero donde se puede cambiar los tiempos de enjuague 1 el segundo se podrá modificar los tiempos de enjuague 2 y el ultimo está compuesto por la posibilidad de alterar los tiempos de desagüe y de exprimido o centrifugado para esto, dentro de las instrucciones en el AVR está una interrupción donde resetea cualquier acción que esté realizando el microcontrolador al enviar alguna señal desde el computador sea esta cliqueando en “comunicación” o en “modificar”, para que desde allí exista la comunicación o se modifique los tiempos.

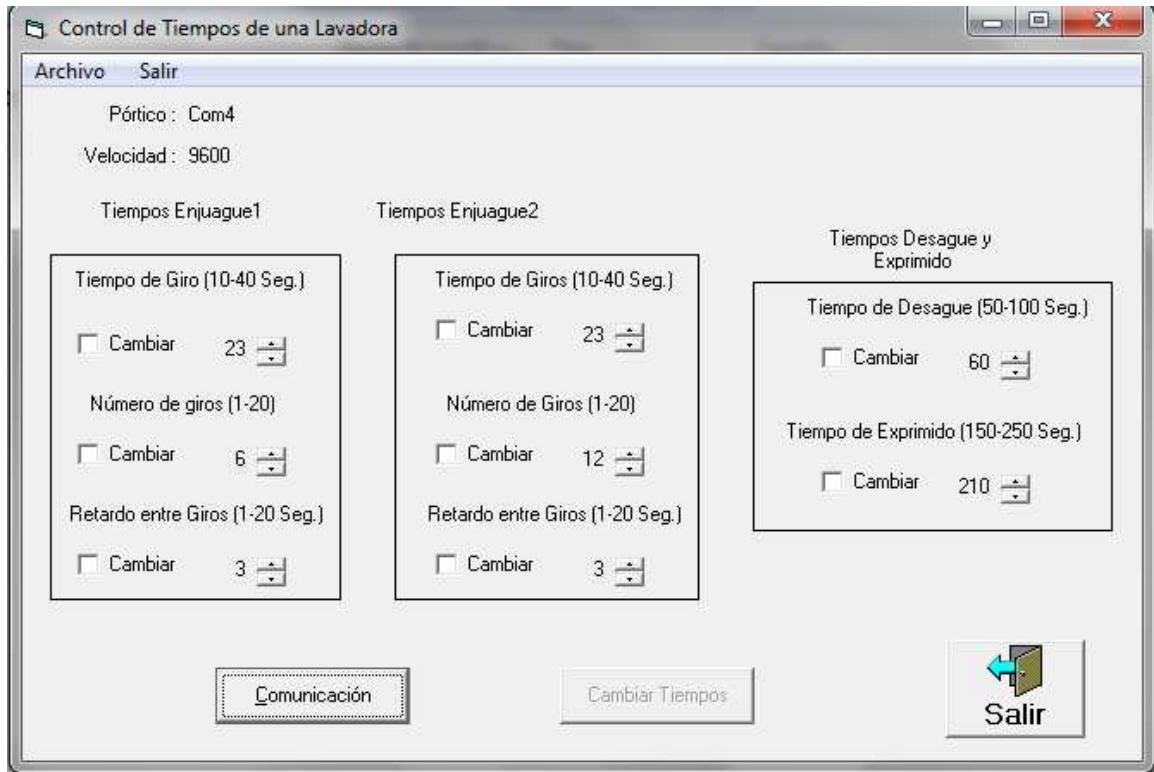


Figura IV.5 Pantalla Control de tiempos de lavado

CAPÍTULO V

EVALUACIÓN Y ANÁLISIS

En el mes de diciembre del 2010 se realizaron las primeras pruebas de funcionamiento de la tarjeta de control en la lavadora Kaskadex del Hospital “Dr. Gustavo Domínguez”, arrojando resultados muy favorables para el avance pero teniendo también algunas dificultades con las fuentes que alimentan a la lavadora y a la tarjeta ya que, en la lavadora hay motores trifásicos que trabajan con 220 V, mientras la tarjeta de control funciona con 110 V por lo que tomar la energía de la misma fuente ocasiono que el microcontrolador se comporte de manera inapropiada teniendo efectos que en la computadora no se da, además de imprimir en la pantalla del LCD símbolos que no estaban programados en el microcontrolador (ver figura V.1).

Para solucionar ese inconveniente se tuvo tres opciones, la primera y la más rápida de efectuar fue el conectar la tarjeta a un UPC cuyo resultado fue muy bueno porque disminuyo el número de símbolos impresos en el LCD, dando a conocer que ese es el problema; para las siguientes pruebas se

optó por comprobar las dos opciones más que se tenía, conseguirse un transformador de 220 V a 110 V y conectar directamente la fuente trifásica (fase, neutro y tierra) tomando la fase y la tierra para obtener 110 V (ver figura V.3).



Figura V.1 Error en el LCD

Cabe recalcar que antes de la primera evaluación de la tarjeta, se encontró inconvenientes con los contactores de la lavadora, los contactores que en ese momento habían estaban viejos y muy deteriorados se los cambio por otros que el hospital tenia de repuesto, no se tuvo ningún gasto en ese aspecto.



Figura V.2 Reemplazo de contactores

Las siguientes pruebas se realizaron durante el mes de enero del 2011; en primeras instancias se reconoció el problema que causaba el funcionamiento de los motores, es decir las salidas, al microprocesador ya que no era solamente la variación de la fuente que causaba ese inconveniente al LCD (ver figura V.1), sino que al no tener un mejor aislamiento con los relés, la variación de voltaje que emanan los motores entraba al microprocesador haciéndolo funcionar de manera incorrecta por lo que sería necesario incorporar al circuito los optoacopladores que están puestos también en las entradas de la tarjeta

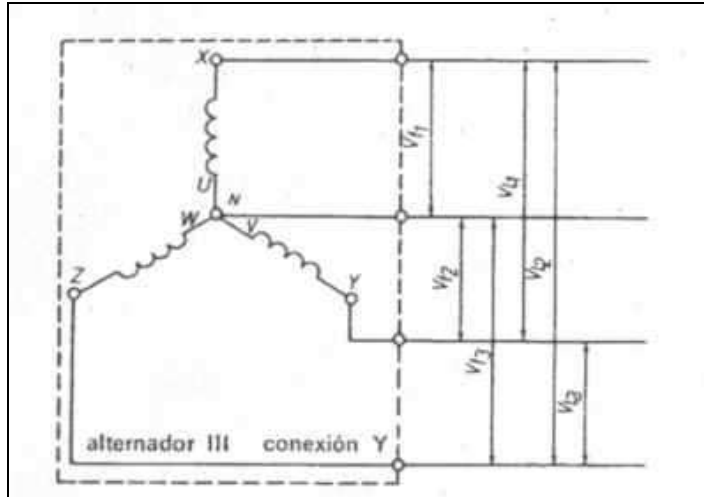


Figura V.3Metodo de trifásico 220 V a 110 V

Febrero y marzo fueron los meses en donde se tuvo grandes respuestas de la tarjeta, ya que se corrigió el error de conexión del ULN2803, se integró capacitores de mayor capacidad a la tarjeta y al LCD y se recomendó al hospital realizar cambios en las conexiones de electricidad, obteniendo buenos resultados en el funcionamiento ya que la tarjeta termino completamente de trabajar en los cuatro programas, sin ninguna falla aunque todavía teniendo pequeños errores en el LCD ya que realizando pruebas con todas las maquinas encendidas la energía que entraba a la tarjeta bajaba hasta 3V, modificando la información que el LCD transmite por otra muy diferente.

5.1 Resultados

Después de la depuración de la tarjeta se consiguió tiempos aceptables para el lavado que están muy apegados a los requerimientos de los trabajadores del departamento de lavado.

El primer pulsador dio los siguientes tiempos:

Tabla V.IProceso de lavado 1

Ciclo	Tiempo (seg)
Enjuague 1	588
Enjuague 2	1117
Lavado	592
Enjuague 3	1120
Centrifugado	150
Total	3567

*3567 segundos son 59 minutos aproximadamente

El pulsador dos obtuvo:

Tabla V.II.Proceso de lavado 2

Ciclo	Tiempo (seg)
Enjuague 1	1118
Lavado	592

Enjuague 2	1121
Centrifugado	148
Total	2979

*2979 segundos son 45 minutos aproximadamente

El tercer pulsador arrojo lo siguiente:

Tabla V.III.Proceso de lavado 3

Ciclo	Tiempo (seg)
Enjuague 1	1119
Lavado	590
Centrifugado	145
Total	1854

*1854 segundos son 30 minutos aproximadamente

El último pulsador es el más corto y se tuvo:

Tabla V.IV.Proceso de lavado 4

Ciclo	Tiempo (seg)
Lavado	592
Centrifugado	148
Total	740

*740 segundos son 15 minutos aproximadamente

CONCLUSIONES

- La lavadora industrial es un aparato eléctrico-electrónico muy completo que ahorra el trabajo manual a las personas con su potencia de lavado optima, que con aditivos de limpieza agregados resulta un producto limpio sin necesidad de esfuerzo físico, además está compuesta de un tambor, motores, tarjeta de control, etc., que funciona iniciando con un proceso de reposo, lavado, exprimido y centrifugado.
- En el diseño del hardware y software se tomó como base fundamental la estructura de un PLC, ya que está compuesto de entradas, salidas y sistema de control, aunque más avanzado y portable sirvió de mucho saber cómo funciona y sus características de arquitectura, para realizar así una tarjeta a medida de los requerimientos del departamento de lavado del hospital.
- La codificación del software y de algunas partes del hardware se realizó con programas especializados como ORCAD, BASCON AVR, ProglSP, los cuales brindan mayor seguridad en los resultados y efectividad en el proceso que cada una tiene, además la mayoría son software libre en algunos de sus módulos.
- El proyecto de tesis se implementó con un microprocesador no estudiado pero muy eficaz al trabajar, el ATMEGA164p, de la familia de los AVRs, proporcionando rapidez de procesamiento y una

estructura similar a los PICs pero con modificaciones importantes como la cantidad de memoria disponible para la programación. La tarjeta está conformada por 3 módulos implementados por separado para que haya mayor soporte en caso de fallo y no tener que volver a reconstruir toda la tarjeta, estos son: módulo de entradas, módulo de control y módulo de salidas.

- Dentro de las pruebas y evaluación de la tarjeta se pudo encontrar algunas falencias que el sistema tenía, dando así una pronta solución aunque retrasando la entrega del proyecto, ya que se sabía que los ensayos hechos por software y utilizando programas de simulación, no serían lo más apegado a la realidad, siendo esto de mucha ayuda.
- El hospital “Dr. Gustavo Domínguez Z.” tendrá una herramienta fundamental en caso de que alguna de sus otras 3 lavadoras presenten anomalías en las tarjetas de control, logrando reemplazar dicha tarjeta dañada por una construida a medida de los requerimientos y ahorrando tiempo y dinero en la adquisición de otra tarjeta del mismo propietario de la lavadora.

RECOMENDACIONES

- La tarjeta de control está dentro de una caja protectora de uso industrial por tal motivo es recomendable que solamente el técnico de mantenimiento tenga las llaves para abrir dicha caja y realizar cualquier modificación o limpieza que se necesite efectuar.
- Es recomendable conocer cómo funciona la interfaz de comunicación entre la tarjeta y el computador para realizar cualquier modificación, por lo cual sería preciso que los trabajadores del departamento de lavado indiquen que necesidades tendrían a futuro, y así realizar las variaciones de las características de los ciclos, cuya operación sería realizado por los técnicos del hospital.
- En diseños futuros, la implementación cambiaría solamente en qué entrada o salida se va a utilizar y que función se le va a dar, siendo esta la manera de programar al microcontrolador sin necesidad de ser expertos en el tema.
- Aplicar un transformador para alimentar la tarjeta y tener niveles de voltajes apropiados como también regularizar todo el departamento de lavado cuando se valla la luz y entre a funcionar la planta generadora, evitando variaciones bruscas de energía.

RESUMEN

Se implementó una tarjeta de control utilizando el microcontrolador ATMEGA de la familia de los AVR, para lavadoras del hospital “Dr. Gustavo Domínguez Z.” de la ciudad de Santo Domingo de los Tsachilas.

Se aplicó las técnicas de observación y entrevista, además del método inductivo para recolectar información sobre la tarjeta que vienen dentro de la lavadora que está dañada, y reemplazarla por un conjunto de módulos que emulan un PLC, basado en la estructura de entradas, control y salidas, formando un sistema correlacionado que funciona para cualquier tipo de lavadora.

La tarjeta se construyó en tres plaquetas diferentes para mayor soporte e introducido en una caja metálica impermeable. La tarjeta tiene 10 entradas y 10 salidas, no todas se utilizaron ya que está hecho para más aplicaciones en futuro, además tiene una entrada de comunicación serial para que desde el computador se puedan cambiar algunos parámetros como el tiempo o el número de ciclos de lavado, esta alimentado con 12 V suficientes para su funcionamiento.

En las pruebas realizadas se obtuvo un 25% más de eficiencia en cada lavado sin ningún inconveniente en la tarjeta, mejorando los tiempos que, según varios usuarios se necesitaba realizar hasta tres lavados para tener un producto limpio.

Al realizar la implementación de la tarjeta de control se concluyó, que permitirá al Hospital dar mayor utilidad a las lavadoras que podrían ser desechadas o dadas de baja por mal funcionamiento del control, dando un ahorro a las empresas públicas y mejorando la atención al público. Por lo que se recomienda que en el Hospital "Dr. Gustavo Domínguez Z." se implemente este proyecto a las demás lavadoras cuando tengan problemas de funcionamiento, ya que no es necesario invertir mucho dinero y se obtiene un rendimiento mayor, gracias a su eficiencia con trabajos a largo tiempo.

SUMMARY

It was implemented a control card, It was used the ATMEGA microcontroller that belongs of the AVRS family. To repair the washing machines that belong the Dr Gustavo Dominguez Hospital of Santo Domingo of the Tsachilas.

The techniques of observation and interview as well as the inductive method were used to collect information about the card that it was in machines damaged. And replace them with a set of modules that emulate a PLC. That is based on the structure of input and control output, forming a correlated system that works for any type of washing machine.

The card was built in three different platelets for more support, and introduced in a waterproof metal box. The card has 10 inputs and 10 outputs, not all were used but it can be use in the future.

Also it has communication serial entry to ensure that the computer can change some parameters such as the time or the number of washing cycles. This fed with twelve Volts sufficient for its operation.

In tests made obtained a twenty-five per cent more than efficiency in each washing without any inconvenient on the card, improving times according to multiple users is needed up to three washes to obtain a clean product

When deploying the control card was concluded, that will allow the hospital can give greater utility to washing machines that were damaged and can be

discarded or given low to cause of the malfunction of the control. This would allow public companies save money and at the same time improve the attention to the public.

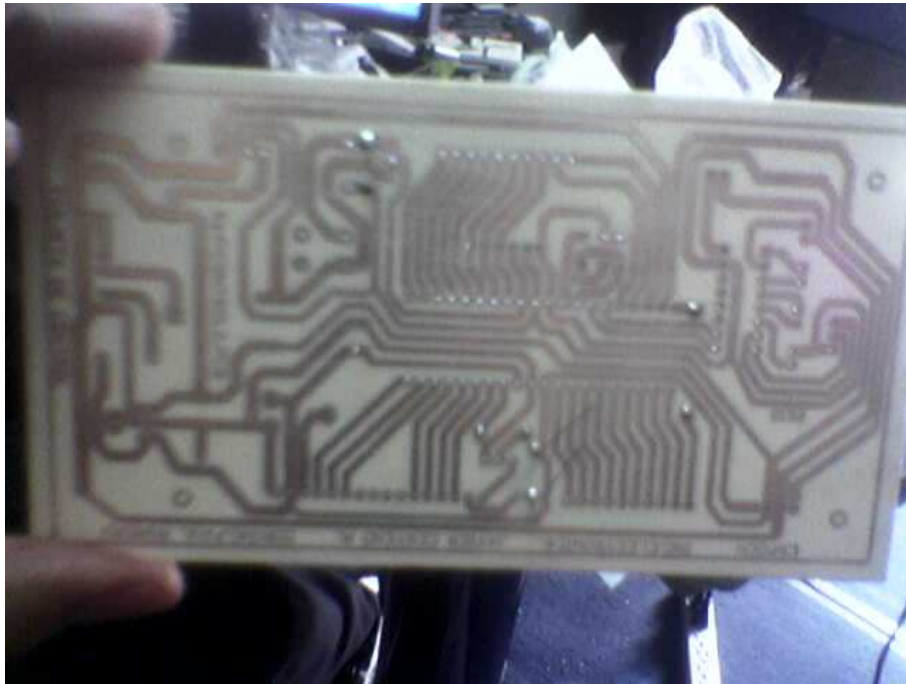
It is recommended to the authorities of the hospital Dr Gustavo Dominguez implement this project to all the washing machines that present problems in its functioning. It is not necessary to invest lots of money and it will can obtain a best performance, thanks to its efficiency to work in the long term.

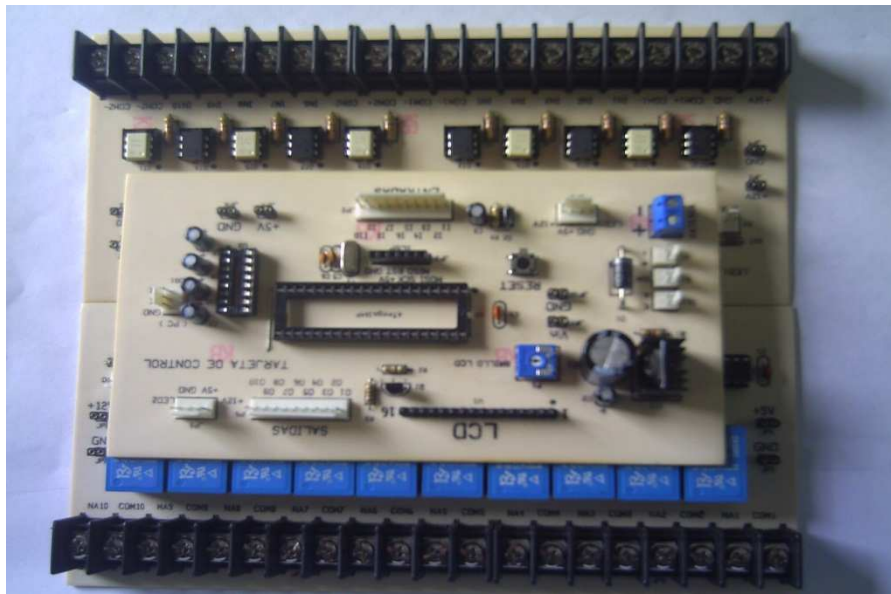
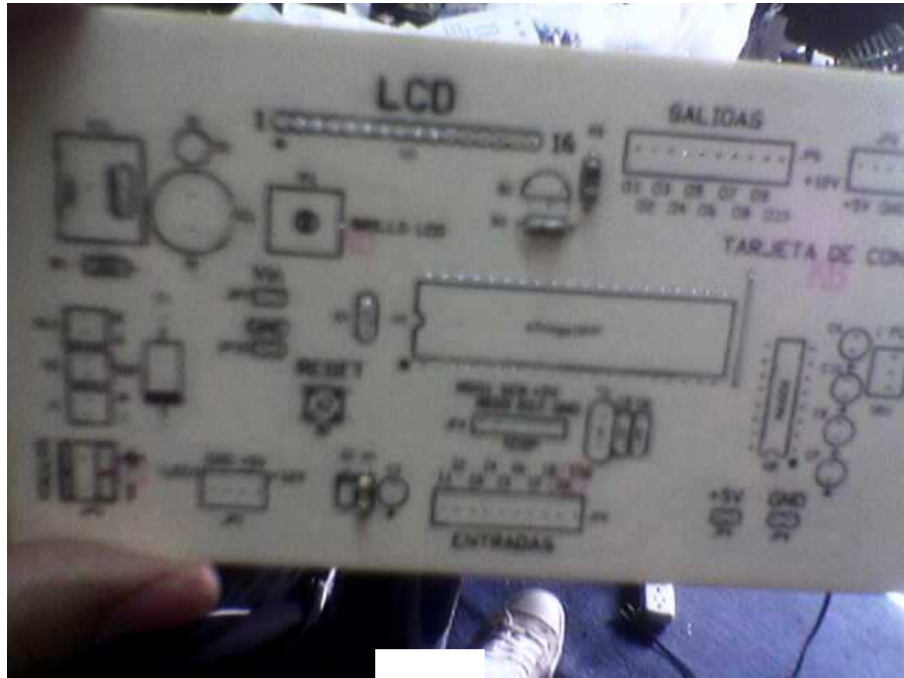
.

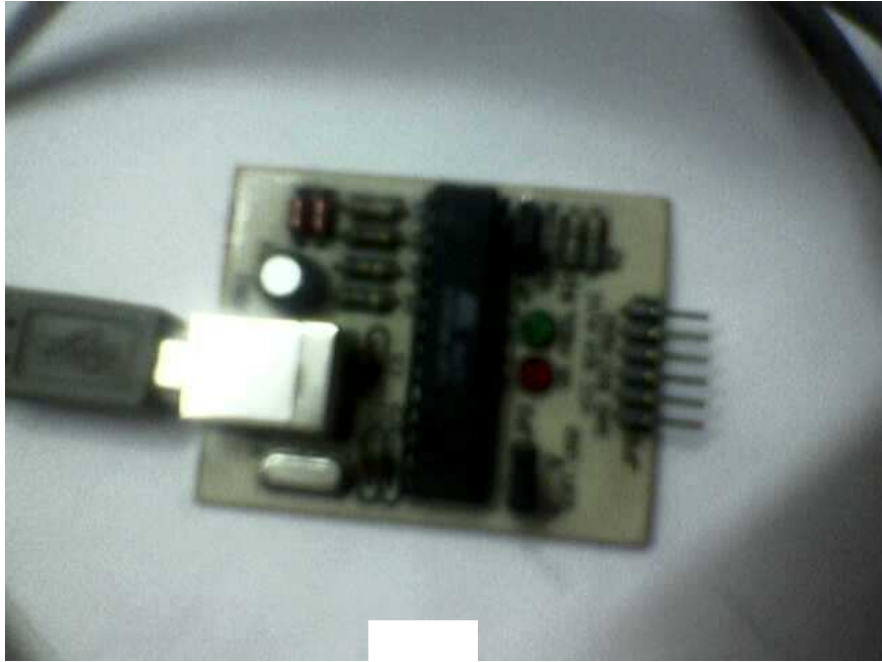
ANEXOS

ANEXO A

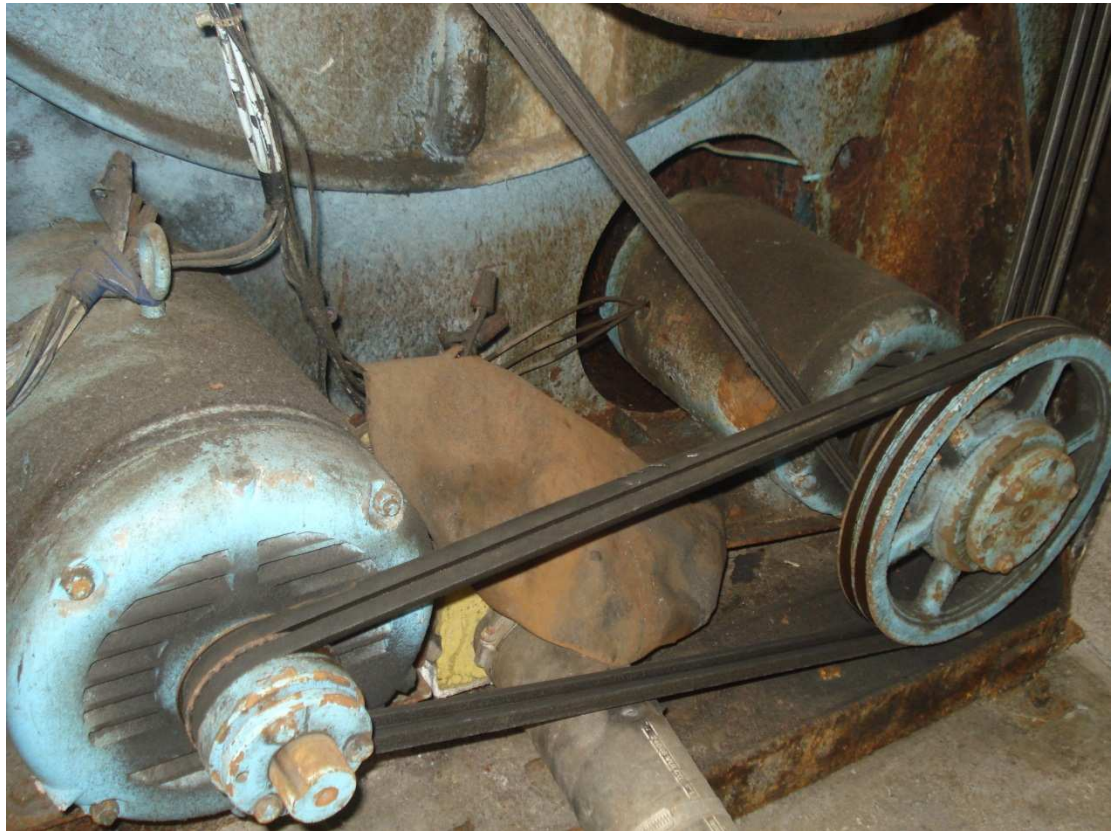
Fotografías













ANEXO B

Programa de prueba uno

```
$regfile = "m16def.dat"
$crystal = 8000000
$baud = 9600
$initmicro
'frecuencia del cristal 8 Mhz

Ddra.7 = 1
Porta.7 = 1
ConfigDebounce = 150

'Portico A.7 como salida
'BK=1

***** LCD ALFANUMERICO *****
ConfigLcdpin = Pin ,Rs = Porta.0 , E = Porta.2 , Db4 = Porta.3 , Db5 = Porta.4 , Db6
= Porta.5 , Db7 = Porta.6
ConfigLcd = 16 * 2

***** INTERRUPCION de serial *****
OnUrxRec_isr
'Define subrutina de Interrupcion Serial1
```


Enable Urxc

```
Const Delay1 = 80
Const Retardo = 3
```

```
Waitms 500
Cursor Off
Cls
'Enable Interrupts
Locate1 , 1
Lcd "hola mundo"
Locate2 , 1
Lcd "Bienvenidos"
```

```
Do
If I01 = 0 Then
    O01 = 1
Else
    O01 = 0
End If
If I02 = 0 Then
    O02 = 1
Else
    O02 = 0
End If
If I03 = 0 Then
    O03 = 1
Else
    O03 = 0
End If
If I04 = 0 Then
    O04 = 1
Else
    O04 = 0
End If
If I05 = 0 Then
    O05 = 1
Else
    O05 = 0
End If
If I06 = 0 Then
    O06 = 1
Else
    O06 = 0
End If
If I07 = 0 Then
    O07 = 1
Else
```

```

    O07 = 0
End If
If I08 = 0 Then
    O08 = 1
Else
    O08 = 0
End If
If I09 = 0 Then
    O09 = 1
Else
    O09 = 0
End If
If I10 = 0 Then
    O10 = 1
Else
    O10 = 0
End If
Gosub Toggle_led1
Loop
Rec_isr:
    Disable Interrupts
    Disable Urx
    Serial1 = ""
    Input Serial1 Noecho
'    Temporal1 = Mid(serial1 , 2 , 12)
'    Serial1 = Temporal1
Cls
Lcd Serial1
Waitms 100
'    Do
'        B = Inkey()
'        Loop Until B = 0
    Enable Interrupts
    Enable Urx
    Bandera1 = 1
Return
*****

Toggle_led1:
    For I = 0 To 2
        Toggle Led1
        Toggle Led2
    Waitms 25
    Next I
    Reset Led1
    Reset Led2
Return
*****

_init_micro:
    Ddra.1 = 1

```

```
Porta.1 = 0  
Return  
End
```

ANEXO C

Programa de prueba dos

```
$regfile = "m16def.dat"
$crystal = 8000000
$baud = 9600                                     'frecuencia del cristal 8 Mhz
$initmicro

Ddra.7 = 1                                       'Portico A.7 como salida
Porta.7 = 1                                       'BK=1
ConfigDebounce = 150

***** LCD ALFANUMERICO *****
ConfigLcdpin = Pin ,Rs = Porta.0 , E = Porta.2 , Db4 = Porta.3 , Db5 = Porta.4 , Db6
= Porta.5 , Db7 = Porta.6
ConfigLcd = 16 * 2

***** INTERRUPCION de serial *****
OnUrxRec_isr                                     'Define subrutina de Interrupcion Serial1
Enable Urx

*****

Const Delay1 = 80
Const Retardo = 3

Waitms 500
Cursor Off
Cls
Enable Interrupts
Locate1 , 1
Lcd "hola mundo"
Locate2 , 1
Lcd "Bienvenidos"

Do
If I01 = 0 Then
  O01 = 1
Else
  O01 = 0
End If
If I02 = 0 Then
  O02 = 1
Else
  O02 = 0
End If
```

```
If I03 = 0 Then
  O03 = 1
Else
  O03 = 0
End If
If I04 = 0 Then
  O04 = 1
Else
  O04 = 0
End If
If I05 = 0 Then
  O05 = 1
Else
  O05 = 0
End If
If I06 = 0 Then
  O06 = 1
Else
  O06 = 0
End If
If I07 = 0 Then
  O07 = 1
Else
  O07 = 0
End If
If I08 = 0 Then
  O08 = 1
Else
  O08 = 0
End If
If I09 = 0 Then
  O09 = 1
Else
  O09 = 0
End If
If I10 = 0 Then
  O10 = 1
Else
  O10 = 0
End If
Gosub Toggle_led1
Loop
*****
Rec_isr:
  Disable Interrupts
  Disable Urx
  Serial1 = ""
  Input Serial1 Noecho
Cls
Lcd Serial1
```

```
Waitms 100
  Enable Interrupts
  Enable Urx
Return
*****

Toggle_led1:
  For I = 0 To 2
    Toggle Led1
    Toggle Led2
  Waitms 25
  Next I
  Reset Led1
  Reset Led2
Return
*****

_init_micro:
  Ddra.1 = 1
  Porta.1 = 0
Return
End
```

ANEXO D

Programa de prueba tres

```
$regfile = "m16def.dat"
$crystal = 8000000
$baud = 9600                                     'frecuencia del cristal 8 Mhz
$initmicro

Ddra.7 = 1                                       'Portico A.7 como salida
Porta.7 = 1                                       'BK=1
ConfigDebounce = 150

***** LCD ALFANUMERICO *****
ConfigLcdpin = Pin ,Rs = Porta.0 , E = Porta.2 , Db4 = Porta.3 , Db5 = Porta.4 , Db6
= Porta.5 , Db7 = Porta.6
ConfigLcd = 16 * 2

***** INTERRUPCION de serial *****
OnUrxRec_isr                                     'Define subrutina de Interrupcion Serial1
Enable Urxc

*****

Const Delay1 = 80
ConstRetardo = 3

Waitms 500
Cursor Off
Cls
'GosubPresentacion
Enable Interrupts
Do
  If I01 = 0 Then
    Waitms 80
    While I01 = 0
      Wend
    Waitms 80
    O04 = 1
    If I06 = 0 Then
      O05 = 1
    Else
      O05 = 0
    End If
```

```
    For I = 0 To 3
O01 = 1
For Z = 1 To 10
Gosub Tiempo1
Next Z
    O01 = 0
For Z = 1 To 3
Gosub Tiempo1
    Next Z
O02 = 1
For Z = 1 To 10
Gosub Tiempo1
Next Z
    O02 = 0
For Z = 1 To 3
Gosub Tiempo1
    Next Z
Next I
O05 = 0
O04 = 0
O02 = 1
Wait 10
'    For Z = 1 To 20
'        Gosub Tiempo1
'    Next Z
O02 = 0
```

```
O04 = 1
If I06 = 0 Then
    O05 = 1
Else
    O05 = 0
End If
```

```
    For I = 0 To 3
O01 = 1
For Z = 1 To 10
Gosub Tiempo1
Next Z
    O01 = 0
For Z = 1 To 3
Gosub Tiempo1
    Next Z
O02 = 1
For Z = 1 To 10
Gosub Tiempo1
Next Z
    O02 = 0
```



```
For Z = 1 To 3
Gosub Tiempo1
  Next Z
Next I
  O05 = 0
  O04 = 0
  O02 = 1
Wait 10
  O02 = 0

O03 = 1
  Wait 5
  O03 = 0
End If

Gosub Toggle_led1
Loop
*****

Rec_isr:
  Disable Interrupts
  Disable Urx
  Serial1 = ""
  Input Serial1 Noecho
Cls
Lcd Serial1
Waitms 100
  Enable Interrupts
  Enable Urx
Return
*****

Toggle_led1:
  For I = 0 To 2
    Toggle Led1
    Toggle Led2
  Waitms 25
  Next I
  Reset Led1
  Reset Led2
Return
*****

Tiempo1:
  For X1 = 0 To 1000
    If I06 = 0 Then
      O05 = 1
    Else
      O05 = 0
    End If
  Waitms 1
  Next X1
Return
```

```
*****
Presentacion:
Cls                               'Limpia la pantalla
Lcd " ESPOCH "
Lowerline
Lcd " RIOBAMBA "
Wait Retardo                       'Esperar 3 seg.
Cls                               'Limpia la pantalla
Lcd " E.I.E "
Lowerline                          'Ir a la Segunda linea
Lcd " "
Wait Retardo                       'Esperar 3 seg.
Cls                               'Limpia la pantalla
Lcd " INGENIERIA "
Lowerline                          'Ir a la Segunda linea
Lcd " ELECTRONICA "
Wait Retardo                       'Esperar 3 seg.
Cls                               'Limpia la pantalla
Lcd " CONTROL "
Lowerline                          'Ir a la Segunda linea
Lcd " LAVADORA "
Wait Retardo                       'Esperar 3 seg.
Cls                               'Limpia la pantalla
Lcd " REALIZADO POR: "
Lowerline                          'Ir a la Segunda linea
Lcd " JAVIER CENTENO"
Wait Retardo                       'Esperar 3 seg.
Cls                               'Limpia la pantalla
Return
*****

_init_micro:
  Ddra.1 = 1
  Porta.1 = 0
Return
End
```

ANEXO E

Programa principal

<pre>(INICIO) Readeeprom Tiempogiroe1 , 1 Readeeprom Girose1 , 2 Readeeprom Retardoe1 , 3 Readeeprom Tiempogiroe2 , 4 Readeeprom Girose2 , 5 Readeeprom Retardoe2 , 6 Readeeprom Tiempodesague , 7 Readeeprom Tiempoexprimido , 8 Waitms 500 Porta.7 = 1 'BK ON Cursor Off Cls Gosub Presentacion Enable Interrupts Flag1 = 0 Do Locate 1 , 1 Lcd " PULSAR " Locate 2 , 1 Lcd " P1 P2 P3 P4 " '----- If I1 = 0 Then 'Pulsador 1 Gosub Toggle_led1 While I1 = 0 Gosub Toggle_led1 Wend Gosub Toggle_led1 Locate 1 , 1 Lcd "PULSADOR 1 ON " Locate 2 , 1 Lcd " LAVADO 1 " Wait 1 If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE "</pre>	<pre>(H) Flag1 = 0 End If End If '----- Gosub Toggle_led1 Loop '***** Toggle_led1: For K = 0 To 4 Toggle Led1 Toggle Led2 Waitms 20 Next K Reset Led1 Reset Led2 Return '***** Tiempo1: For X1 = 1 To 3 If I6 = 0 Then O5 = 1 Gosub Toggle_led1 O6 = 1 Gosub Toggle_led1 Else O5 = 0 Gosub Toggle_led1 O6 = 0 Gosub Toggle_led1 End If If I5 = 1 Then Gosub Toggle_led1 While I5 = 1 Gosub Toggle_led1 Wend</pre>
---	---

<pre>Locate 2, 1 Lcd " ENJUAGUE 1 " Gosub Enjuague1 Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 (A) (A) Lcd " ENJUAGUE 1 " End If If Flag1 = 0 Then Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " ENJUAGUE 2 " Gosub Enjuague2 Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " ENJUAGUE 2 " End If If Flag1 = 0 Then Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " LAVADO " Gosub Enjuague1 Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " LAVADO " End If If Flag1 = 0 Then Gosub Enjuague1 End If If Flag1 = 0 Then Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " ENJUAGUE 3 " Gosub Enjuague1 Locate 1, 1 Lcd " CICLO DE " Locate 2, 1 Lcd " ENJUAGUE 3 " End If If Flag1 = 0 Then</pre>	<pre>Locate 1, 1 Lcd "CICLO DE LAVADO" Locate 2, 1 Lcd "PARADO POR-STOP-" Gosub Toggle_led1 X1 = 255 (I) (I) I = 255 Z = 255 Gosub Toggle_led1 O5 = 0 Gosub Toggle_led1 O6 = 0 Gosub Toggle_led1 O4 = 0 Gosub Toggle_led1 O2 = 0 Gosub Toggle_led1 O1 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 O3 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 Flag1 = 1 For L = 1 To 30 Gosub Toggle_led1 Next L End If Gosub Toggle_led1 Next X1 Gosub Toggle_led1 Return '***** Tiempo2: For X1 = 1 To 3 Gosub Toggle_led1 Gosub Toggle_led1 If I5 = 1 Then Gosub Toggle_led1 While I5 = 1 Gosub Toggle_led1 Wend Locate 1, 1</pre>
--	--

<pre>Gosub Toggle_led1 O3 = 1 Gosub Toggle_led1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " (B) (B) Gosub Toggle_led1 For Z = 0 To Tiempoexprimido Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " Gosub Tiempo2 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " Next Z Gosub Toggle_led1 O3 = 0 Gosub Toggle_led1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " End If If Flag1 = 0 Then Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 1 " Wait 1 Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 1 " Wait 10 Else Flag1 = 0 End If End If ----- If I2 = 0 Then 'Pulsador 2 Gosub Toggle_led1</pre>	<pre>Lcd "CICLO DE LAVADO" Locate 2 , 1 Lcd "PARADO POR-STOP-" Gosub Toggle_led1 X1 = 255 I = 255 Z = 255 (K) (K) Gosub Toggle_led1 O5 = 0 Gosub Toggle_led1 O6 = 0 Gosub Toggle_led1 O4 = 0 Gosub Toggle_led1 O2 = 0 Gosub Toggle_led1 O1 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 O3 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 Flag1 = 1 For L = 1 To 30 Gosub Toggle_led1 Next L End If Gosub Toggle_led1 Next X1 Gosub Toggle_led1 Return ***** Enjuague1: O4 = 1 Gosub Toggle_led1 If I6 = 0 Then O5 = 1 Gosub Toggle_led1 O6 = 1 Gosub Toggle_led1 Else O5 = 0 Gosub Toggle_led1</pre>
--	---

<pre>While I2 = 0 Gosub Toggle_led1 Wend Gosub Toggle_led1 Locate 1 , 1 Lcd "PULSADOR 2 ON " Locate 2 , 1 Lcd " LAVADO 2 "</pre> <p>(C) (C)</p> <pre>Wait 1 If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " ENJUAGUE 1 " Gosub Enjuague2 Locate 1 , 1 Lcd " CICLO DE "</pre> <pre>Locate 2 , 1 Lcd " ENJUAGUE 1 "</pre> <pre>End If If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " LAVADO "</pre> <pre>Gosub Enjuague1 Locate 1 , 1 Locate 1 , 1 Lcd " CICLO DE "</pre> <pre>Locate 2 , 1 Lcd " LAVADO "</pre> <pre>End If If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE "</pre> <pre>Locate 2 , 1 Lcd " ENJUAGUE 2 "</pre> <pre>Gosub Enjuague1 Locate 1 , 1 Lcd " CICLO DE "</pre> <pre>Locate 2 , 1 Lcd " ENJUAGUE 2 "</pre> <pre>End If If Flag1 = 0 Then Gosub Toggle_led1</pre>	<pre>O6 = 0 Gosub Toggle_led1 End If For I = 1 To Girose1 If Flag1 = 0 Then O1 = 1 Gosub Toggle_led1</pre> <p>(L) (L)</p> <pre>For Z = 0 To Tiempogiroe1 Gosub Tiempo1 Next Z O1 = 0 Gosub Toggle_led1 End If If Flag1 = 0 Then For Z = 1 To Retardoe1 Gosub Tiempo1 Next Z End If If Flag1 = 0 Then O2 = 1 Gosub Toggle_led1 For Z = 1 To Tiempogiroe1 Gosub Tiempo1 Next Z O2 = 0 Gosub Toggle_led1 End If If Flag1 = 0 Then For Z = 1 To Retardoe1 Gosub Tiempo1 Next Z End If Next I Gosub Toggle_led1 O5 = 0 Gosub Toggle_led1 O6 = 0 Gosub Toggle_led1 O4 = 0 Gosub Toggle_led1</pre>
--	--

```
O3 = 1
  Gosub Toggle_led1
Locate 1 , 1
  Lcd " CICLO DE "
  Locate 2 , 1
Lcd " CENTRIFUGADO "
  Gosub Toggle_led1
  For Z = 0 To Tiempoexprimido
    Gosub Tiempo2
(D)
(D)
  Next Z
  Gosub Toggle_led1
  O3 = 0
  Gosub Toggle_led1
Locate 1 , 1
  Lcd " CICLO DE "
  Locate 2 , 1
Lcd " CENTRIFUGADO "
  End If
  If Flag1 = 0 Then
Locate 1 , 1
  Lcd " FIN DE "
  Locate 2 , 1
  Lcd " LAVADO 2 "
  Wait 1
  Locate 1 , 1
  Lcd " FIN DE "
Locate 2 , 1
  Lcd " LAVADO 2 "
  Wait 10
  Else
  Flag1 = 0
  End If
End If
-----
If I3 = 0 Then                                'Pulsador 3
  Gosub Toggle_led1
  While I3 = 0
    Gosub Toggle_led1
  Wend
  Gosub Toggle_led1
  Locate 1 , 1
  Lcd "PULSADOR 3 ON "
  Locate 2 , 1
  Lcd " LAVADO 3 "
  Wait 1

O2 = 0
  Gosub Toggle_led1
O1 = 0
  Gosub Toggle_led1
  Gosub Toggle_led1
  Gosub Toggle_led1
O3 = 0
  Gosub Toggle_led1
  Gosub Toggle_led1
(M)
(M)
  Gosub Toggle_led1
  If Flag1 = 0 Then
    Waitms 1
    O1 = 1
    Gosub Toggle_led1
    For Z = 1 To Tiempodesague
  Gosub Tiempo2
    Next Z
    O1 = 0
  Gosub Toggle_led1
  End If
  Waitms 1
Return
*****
Enjuague2:
  O4 = 1
  Gosub Toggle_led1
  If I6 = 0 Then
    O5 = 1
    Gosub Toggle_led1
    O6 = 1
    Gosub Toggle_led1
  Else
    O5 = 0
    Gosub Toggle_led1
    O6 = 0
    Gosub Toggle_led1
  End If

  For I = 1 To Girose2
    If Flag1 = 0 Then
      O1 = 1
      Gosub Toggle_led1
    For Z = 0 To Tiempogiroe2
      Gosub Tiempo1
    Next Z
```

<pre> If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " ENJUAGUE 1 " Gosub Enjuague1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " ENJUAGUE 1 " (E) (E) End If If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " LAVADO " Gosub Enjuague1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " LAVADO " End If If Flag1 = 0 Then Gosub Toggle_led1 O3 = 1 Gosub Toggle_led1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " Gosub Toggle_led1 For Z = 0 To Tiempoexprimido Gosub Tiempo2 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " Next Z Gosub Toggle_led1 O3 = 0 Gosub Toggle_led1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " </pre>	<pre> O1 = 0 Gosub Toggle_led1 End If If Flag1 = 0 Then For Z = 1 To Retardoe2 Gosub Tiempo1 Next Z End If If Flag1 = 0 Then (N) (N) O2 = 1 Gosub Toggle_led1 For Z = 1 To Tiempogiroe2 Gosub Tiempo1 Next Z O2 = 0 Gosub Toggle_led1 End If If Flag1 = 0 Then For Z = 1 To Retardoe2 Gosub Tiempo1 Next Z End If Next I Gosub Toggle_led1 O5 = 0 Gosub Toggle_led1 O6 = 0 Gosub Toggle_led1 O4 = 0 Gosub Toggle_led1 O2 = 0 Gosub Toggle_led1 O1 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 O3 = 0 Gosub Toggle_led1 Gosub Toggle_led1 Gosub Toggle_led1 If Flag1 = 0 Then Waitms 1 </pre>
--	---

<pre> End If If Flag1 = 0 Then Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 3 " Wait 1 Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 3 " (F) (F) Wait 30 Else Flag1 = 0 End If End If End If ----- If I4 = 0 Then 'Pulsador 4 Gosub Toggle_led1 While I4 = 0 Gosub Toggle_led1 Wend Gosub Toggle_led1 Locate 1 , 1 Lcd "PULSADOR 4 ON " Locate 2 , 1 Lcd " LAVADO 4 " Wait 1 If Flag1 = 0 Then Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " LAVADO " Gosub Enjuague1 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " LAVADO " End If If Flag1 = 0 Then Gosub Toggle_led1 O3 = 1 Gosub Toggle_led1 Locate 1 , 1 Lcd " CICLO DE " </pre>	<pre> O1 = 1 Gosub Toggle_led1 For Z = 1 To Tiempodesague Gosub Tiempo2 Next Z O1 = 0 Gosub Toggle_led1 End If Waitms 1 Return ***** (Ñ) (Ñ) Presentacion: Cls 'Limpia la pantalla Lcd " ESPOCH " Lowerline Lcd " RIOBAMBA " Wait Retardo 'Esperar 3 seg. Cls 'Limpia la pantalla Lcd " E.I.E " Lowerline 'Ir a la Segunda linea Lcd " " Wait Retardo 'Esperar 3 seg. Cls 'Limpia la pantalla Lcd " INGENIERIA " Lowerline 'Ir a la Segunda linea Lcd " ELECTRONICA " Wait Retardo 'Esperar 3 seg. Cls 'Limpia la pantalla Lcd " CONTROL " Lowerline 'Ir a la Segunda linea Lcd " LAVADORA " Wait Retardo 'Esperar 3 seg. Cls 'Limpia la pantalla Lcd " REALIZADO POR: " Lowerline 'Ir a la Segunda linea Lcd " JAVIER CENTENO " Wait Retardo 'Esperar 3 seg. Cls 'Limpia la pantalla Return ***** Rec_isr: Disable Interrupts Disable Urxc Serial1 = "" Input Serial1 Noecho </pre>
--	---

<pre>Locate 2 , 1 Lcd " CENTRIFUGADO " Gosub Toggle_led1 For Z = 0 To Tiempoexprimido Gosub Tiempo2 Locate 1 , 1 Lcd " CICLO DE " Locate 2 , 1 Lcd " CENTRIFUGADO " Next Z Gosub Toggle_led1 O3 = 0 (G) (G) Gosub Toggle_led1 End If If Flag1 = 0 Then Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 4 " Wait 1 Locate 1 , 1 Lcd " FIN DE " Locate 2 , 1 Lcd " LAVADO 4 " Wait 30 Else (H)</pre>	<pre>Select Case Serial1 Case "a" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 1 Readeeprom Tiempo Giroe1 , 1 Case "b" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 2 Readeeprom Girose1 , 2 Case "c" (O) (O) Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 3 Readeeprom Retardoe1 , 3 Case "d" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 4 Readeeprom Tiempo Giroe2 , 4 Case "e" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 5 Readeeprom Girose2 , 5 Case "f" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 6 Readeeprom Retardoe2 , 6 Case "g" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 7 Readeeprom Tiempo desague , 7 Case "h" Input Serial2 Noecho DatoSerial = Val(serial2) Writeeprom DatoSerial , 8 Readeeprom Tiempo exprimido , 8 End Select Enable Interrupts Enable Urcx Return</pre>
---	--

	<pre>***** _init_micro: Ddra.1 = 1 Porta.1 = 0 turn End Dta: \$eeprom Data 0 , 23 , 2 , 3 , 23 , 4 , 3 , 60 , 40 \$data (FIN)</pre>
--	---

BIBLIOGRAFÍA

- 1.- BARRETT, F. y PACK, D.,** Atmel Microcontroller Primer: Programming and interfacing., 2da ed., Southern Methodist University: Morgan & Claypool USA., 2008., 39 p.
- 2.- BLAKE, R.,** Sistemas electrónicos de comunicaciones., 2da ed., Inglaterra., Thomson Learning., 2004., 102 p.
- 3.- KRETZINANN, T y ANGULO, D.,** Electrónica y automática aplicadas a la industria., 7ma ed., Madrid- España., Paraninfo., 1979., 87 p.
- 4.- REYES, C.,** Aprenda a programar microcontroladores., Quito-Ecuador., Ayerve., 2004., 55 p.

INTERNET

Elementos electrónicos

es.wikipedia.org/wiki/AVR

2010-08-14

www.datasheetdir.com/ATMEGA164P+AVR-microcontrollers

2010-08-19

rs38.rapidshare.com/files/203310262/atmega164p_espa_ol.pdf

2010-08-23

www.elektroda.pl/rtvforum/topic1363593.html

2010-03-14

micros.mforos.com/1149903/5977027-bascom-avr-compilador-basic-para-avr/

2010-09-12

Lavadoras

www.mcselec.com/index.php?page=shop.downloads&option=com_php

2010-08-16

www.brothersoft.com/bascom-comunicacion-serial-load-45490.html

2010-08-16

Microcontroladores AVR

www.monografias.com/trabajos12/microco/microco.shtml

2010-08-17

www.monografias.com/trabajos27/microcontroladores/microcontroladores.shtml

2010-10-16

www.atmel.com/dyn/products/tools_card.asp?family_id=607&family

2010-10-17

www.forosdeelectronica.com/f24/atmel-vs-microchip-12589/

2010-11-01

proyctoselectronics.blogspot.com/2008/09/planchado-de-circuito-impreso.html

2010-11-06

Atmel164p

robotsperu.org/foros/atmel-vt398.html

2010-09-06

www.dinastiasoft.com.ar/bascomavr.htm

2010-10-28

es.wikipedia.org/wiki/Circuito_impreso

2010-11-16