

Universidad
Politécnica
de Cartagena



industriales
etsii UPCT

Diseño de la tarjeta de control de un sistema de planchado industrial

Titulación:	I.T.I. especialidad en Electrónica Industrial
Alumno/a:	Adrián Rivera Sánchez
Director/a/s:	Pedro Díaz Hernández

Cartagena, a 26 de septiembre de 2013

ÍNDICE

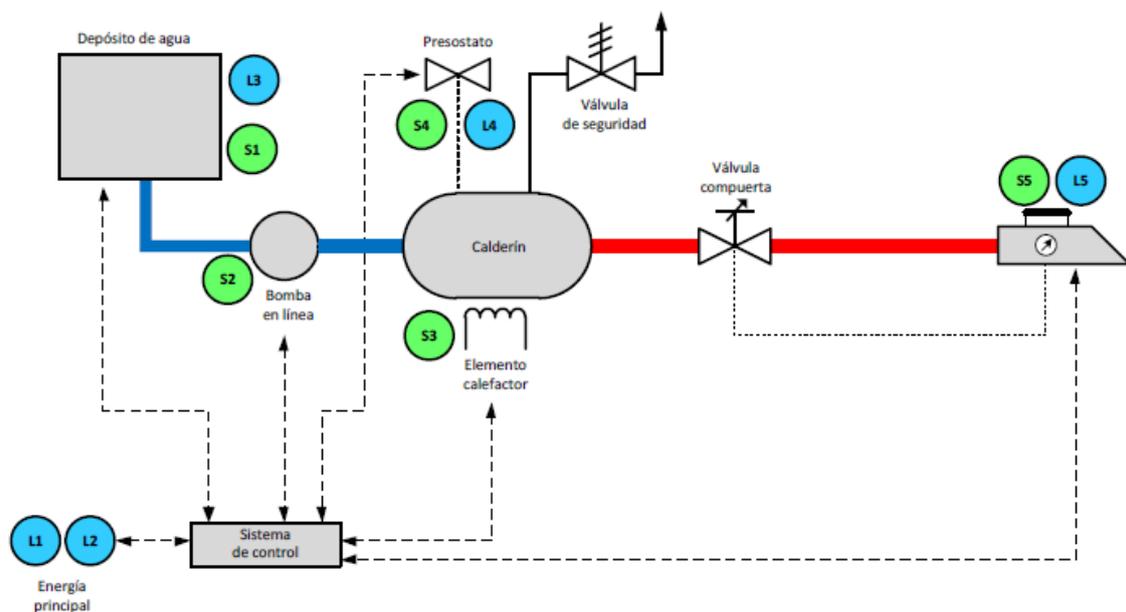
1. MEMORIA.	5
1.1. Objetivo.	6
1.2. Elementos principales que componen el proyecto.	6
1.2.1. Calderín.	9
1.2.2. Plancha.	9
1.2.3. Depósito de agua.	10
1.2.4. Bomba de agua.	10
1.2.5. Sensor de nivel del depósito.	11
1.2.6. Sensor de nivel del calderín.	11
1.2.7. Resistencia calefactora del calderín.	12
1.2.8. Sensor de presión.	12
1.2.9. Sensores de funcionamiento	13
1.2.10. Relé de estado sólido	13
1.2.11. Pantalla LCD de 16x4.	14
1.2.12. Diodos Led de alarma y funcionamiento.	14
2. PLANOS	16
3. PLIEGO DE CONDICIONES	22
3.1. Características del calderín y dimensionamiento de las tuberías a usar.	23
3.2. Funcionamiento de los sensores.	25
3.2.1. Sensor de nivel del depósito.	25

3.2.2. Sensor de presión.	26
3.2.3. Sensor de nivel del calderín.	28
3.2.4. Sensores de funcionamiento.	30
3.2.4.1. Sensor de funcionamiento en corriente continua.	30
3.2.4.2. Sensor de funcionamiento en corriente alterna.	31
3.3. Funcionamiento de la pantalla LCD.	32
3.4. Funcionamiento de la placa Arduino UNO.	35
3.5. Entorno de programación de Arduino.	36
3.6. Programa de control de la placa Arduino UNO.	39
3.7. Puesta en marcha del sistema.	44
4. PRESUPUESTO.	45
5. BIBLIOGRAFÍA.	47

1. MEMORIA

1.1. Objetivo.

El objetivo del proyecto es encontrar una solución viable y práctica para un sistema de planchado industrial formado además del sistema de control, por un calderín y una plancha con selector de temperatura, cuyo esquema se corresponde al del siguiente diagrama:



En la propuesta del proyecto se detallan los siguientes elementos sensores e indicadores lumínicos:

- S1: Sensor de existencia de agua.
- S2: Sensor de funcionamiento de la bomba de aporte de agua al calderín.

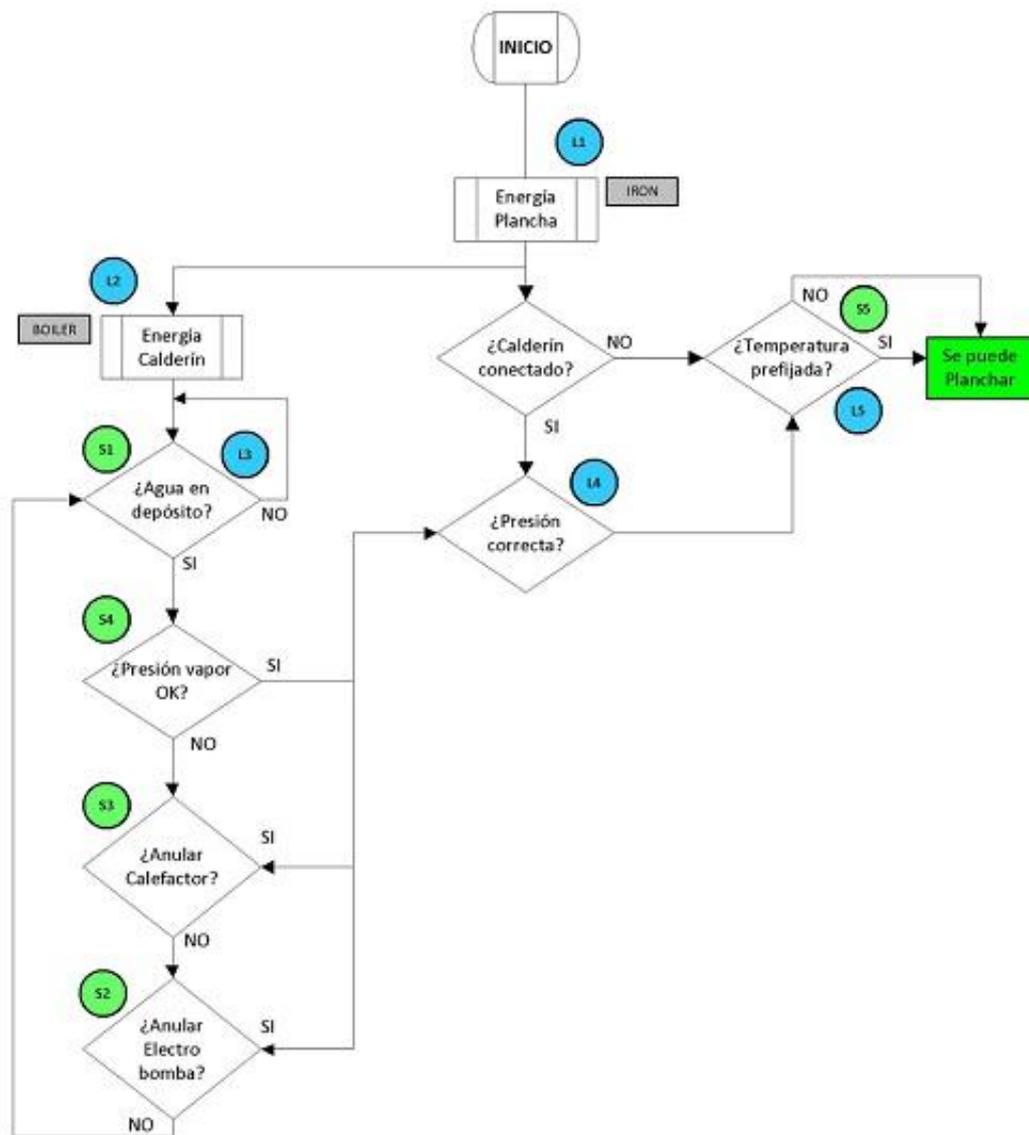
- S3: Sensor de funcionamiento de la resistencia calefactora del calderín.
- S4: Sensor de presión alcanzada en el interior del calderín.
- S5: Sensor de temperatura del elemento de planchado (En este caso se ha optado por reemplazarlo por un sensor de efecto Hall).

- L1: Activación del elemento calefactor de la caldera.
- L2: Activación del elemento de planchado.
- L3: Indicación visual de falta de agua en el depósito.
- L4: Presión alcanzada en el calderín.
- L5: Temperatura alcanzada por el elemento de planchado.

Además se deben cumplir también las siguientes condiciones:

- Depósito de agua destilada de 5 litros.
- El agua deberá aportarse caliente o calentarse previamente a su aportación a la caldera.
- Bomba de impulsión en línea para aportar el agua a la caldera. Deberá poseer un caudal adecuado para alimentar sin problemas a la caldera. El funcionamiento de esta bomba puede ser temporizada, o controlando el caudal real aportado a la caldera.
- La caldera tendrá un consumo máximo de 4Kw.
- El presostato de la caldera estará tarado a 3.5 bares de presión. Al alcanzar esta presión estará preparado para su funcionamiento.
- La válvula de seguridad estará tarada a 5 bares de presión. Por encima de esta presión, el sistema expulsará al aire el exceso de presión, para evitar la rotura de manguitos o agrietamiento/explosión de la caldera.
- El elemento de planchado, en contacto con el tejido, tendrá un consumo máximo de 2Kw.
- Las indicaciones de estado del conjunto y los avisos visuales se indicarán mediante un LCD de 4x16 líneas retroiluminado.
- Si se instala algún avisador acústico se realizará con un zumbador piezoeléctrico.
- La activación de las cargas de alto consumo se realizarán con relés de estado sólido, no siendo posible la utilización de relés electromecánicos.
- Las indicaciones visuales individuales se realizarán mediante diodos LED's de bajo consumo del color que se estime adecuado.

Para la implementación de este control nos serviremos como base central del proyecto de una placa de desarrollo *Arduino UNO*, sobre la que montaremos una placa externa conocida normalmente como *shield* en la que colocaremos los elementos sensores y auxiliares necesarios, para el correcto funcionamiento de la tarjeta de control. Una idea aproximada, aunque puede variar a gusto del diseñador del modo de trabajo del sistema de planchado lo podemos ver en el siguiente flujograma:



1.2. Elementos principales que componen el proyecto.

1.2.1. Calderín.

Es el encargado de producir el vapor necesario para realizar el planchado gracias a la resistencia eléctrica o elemento calefactor de su interior, tendrá un consumo máximo de 4 kW y una presión de trabajo de 3,5 bares, además de contar con una válvula de seguridad que liberará la presión al exterior cuando ésta supere los 5 bares, dadas estas características será necesario encargar su construcción a una empresa especializada.



1.2.2. Plancha.

La plancha a utilizar es la *Jolly P+Guard* del fabricante *Strima* con una potencia de 800 W (230 V, 50 Hz); un peso de 2,7 kg y con control e indicador de temperatura adecuada según el tejido, indicada para cualquier fuente externa de vapor, se recomienda su limpieza regular con el producto *DECAPFER* fabricado también por *Strima*.



1.2.3. Depósito de agua.

El depósito de agua necesario de 5 litros es del fabricante *Sunbox* con referencia BCG-5, construido en polietileno de alta densidad, con junta tórica de seguridad para asegurar estanqueidad total, se alimentará siempre con agua previamente tratada y destilada para prevenir incrustaciones y deterioros en los elementos que constituyen el centro de planchado, irá conectada por medio de la abertura del grifo a una tubería conectada a su vez a la bomba de agua.



1.2.4. Bomba de agua

Es la encargada de llevar el agua desde el depósito de 5 litros hasta el calderín, se ha elegido para el proyecto el modelo 2088-514-145-12v con un consumo máximo de 5,6 A y una presión de trabajo de 10 psi (0,7 bares) del fabricante *Shurflo* de funcionamiento silencioso y con un voltaje de trabajo de 12 V, se accionará cada

vez que el nivel de agua en el calderín se encuentre fuera del rango marcado por las sondas de nivel.



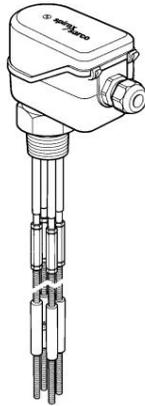
1.2.5. Sensor de nivel del depósito.

Lo colocaremos en el depósito de agua de 5 litros, es de la casa *CEBEK* con referencia *C-7235*, trabaja con dos niveles, alto y bajo, que llevaremos a una de las entradas analógicas de la placa *Arduino UNO* para controlar la existencia de agua, está compuesto de una parte móvil (flotador) que contiene interiormente un imán.



1.2.6. Sensor de nivel del calderín.

Se trata de la sonda de nivel *LP10-4* del fabricante *Spirax Sarco*, de sus cuatro terminales utilizaremos dos para detectar el nivel correcto de agua que se encontrará entre dos valores (máximo y mínimo) dentro del calderín y hacer funcionar de esta manera la bomba de agua.



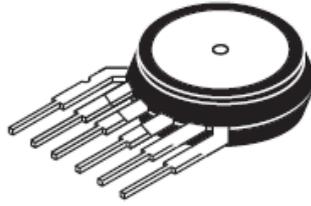
1.2.7. Resistencia calefactora del calderín.

La resistencia elegida tiene como referencia *12449* y será suministrada por *Iberital*, se alimenta a 220 V y tiene un consumo de 2000 W, es decir dentro de los márgenes aceptables de consumo que se exigen en el proyecto.



1.2.8. Sensor de presión.

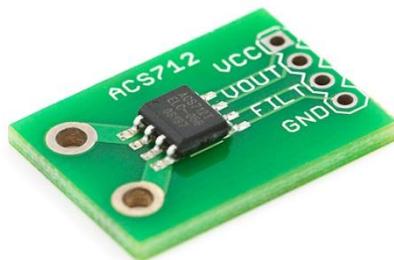
Es necesario saber cuándo se ha alcanzado la presión de vapor correcta de 3,5 bares en el calderín, para este propósito el sensor adecuado es el *MPX5500D* de *Freescale Semiconductors* con un rango de medida de entre 0 y 5 bares, cuando se llegue al valor prefijado se mostrará un mensaje en la pantalla LCD y se encenderá un diodo led verde conectado a una de las salidas digitales de *Arduino UNO*.



MPX5500D
CASE 867-08

1.2.9. Sensores de funcionamiento.

Serán necesarios tres sensores de este tipo para poder comprobar el correcto funcionamiento de la bomba de agua, el elemento calefactor del calderín y la plancha. Para los tres casos usaremos el *ACS712ELCTR-20A-T* de *Allegro MicroSystems*. El método de funcionamiento es el siguiente: la corriente que circula a través del circuito integrado generará un campo magnético que el sensor transforma en un voltaje proporcional que nos indicará la corriente que circula por cada uno de los tres elementos anteriormente referenciados.



1.2.10. Relé de estado sólido.

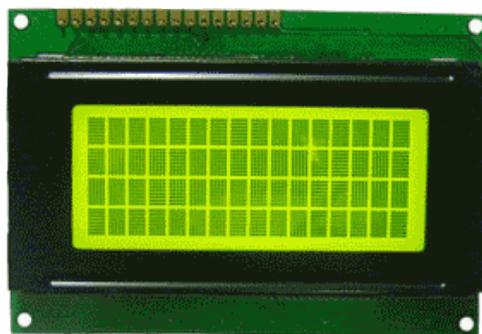
Es el encargado de accionar o apagar la bomba de agua, está controlado por la señal proveniente de los dos sensores de nivel del calderín, conocidas las características de voltaje y consumo de corriente de la bomba se ha optado por el relé *84134750* de

Crouzet que como características generales cuenta con un voltaje mínimo de activación de control de solo 3 V y un manejo de carga de hasta 10 A, lo que lo convierte en el componente ideal para funcionar entre la placa *Arduino UNO* y la bomba. Al tratarse de un componente robusto y de aplicación industrial y sobre todo por el tamaño del componente la instalación se hará fuera de la placa.



1.2.11. Pantalla LCD de 16x4.

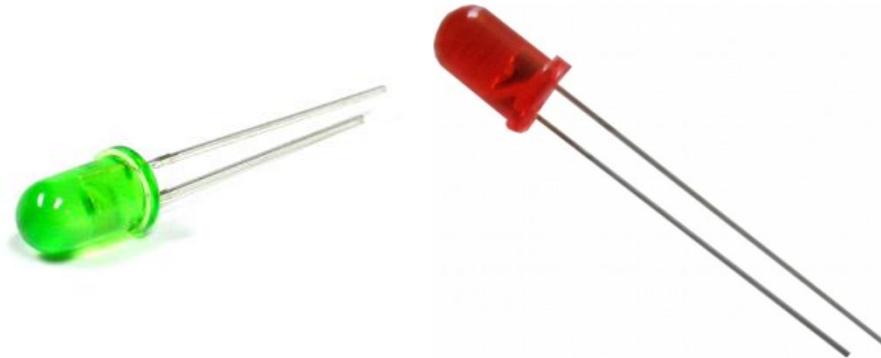
Es la encargada de mostrar todos los mensajes de alarma y de funcionamiento del sistema, esto lo hará mediante mensajes alternativos con una frecuencia aproximada de 1 segundo en los que se llevará el control de todos los sensores y actuadores de la placa *Arduino UNO*.



1.2.12. Diodos Led de alarma y funcionamiento.

Se utilizarán un total de 4 diodos led de bajo consumo, 1 de color rojo de alarma ante la falta de agua en el depósito y 3 de color verde que indicarán lo siguiente:

- Funcionamiento de la resistencia del calderín.
- Funcionamiento de la plancha.
- Presión de trabajo alcanzada en el calderín.



2. PLANOS

0

0

1

1

2

2

3

3

4

4

5

5

6

6

7

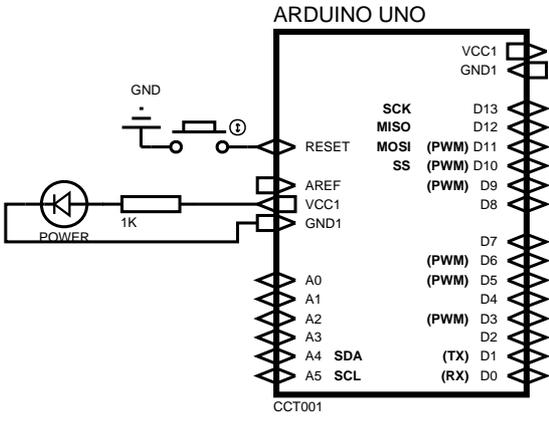
7

8

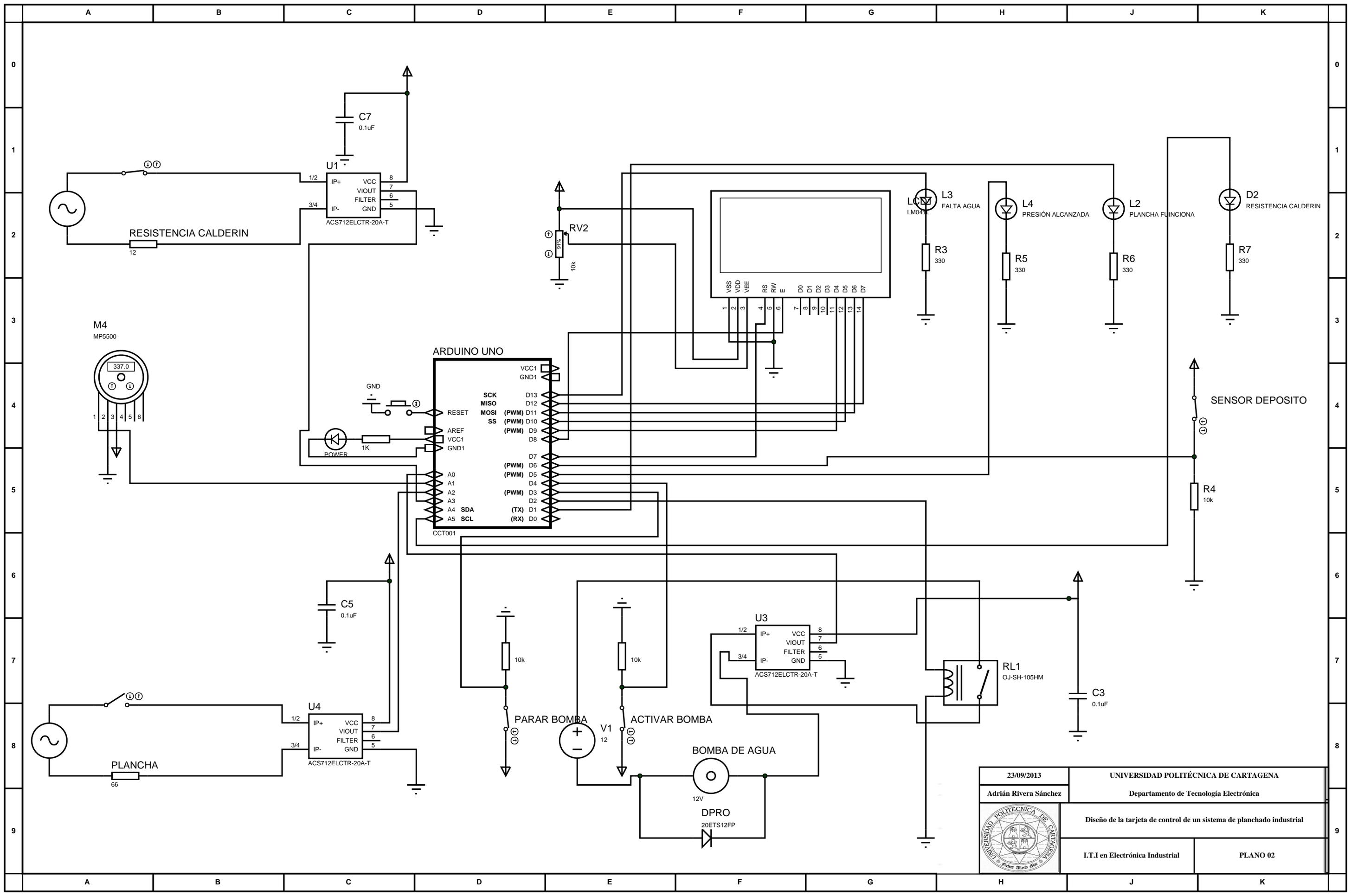
8

9

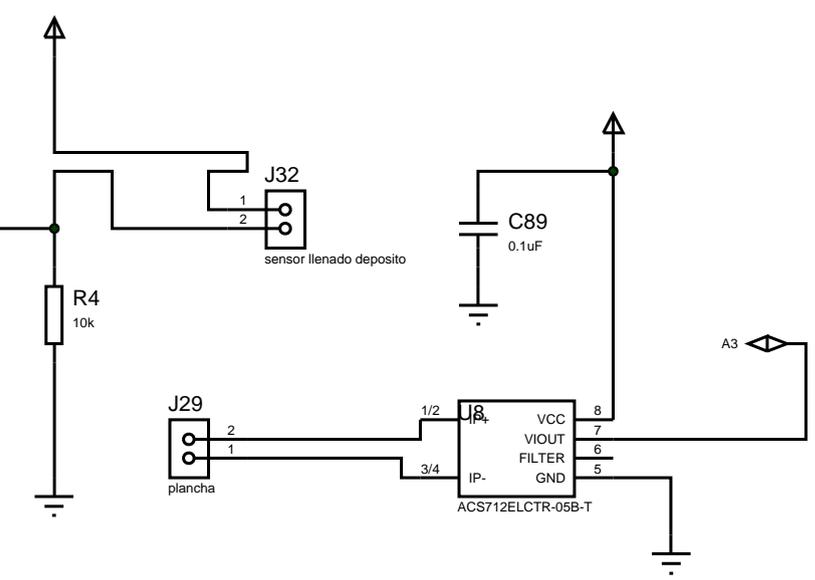
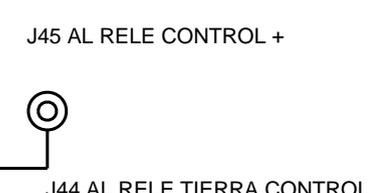
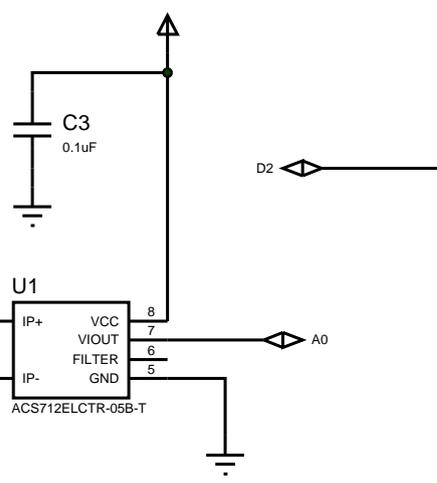
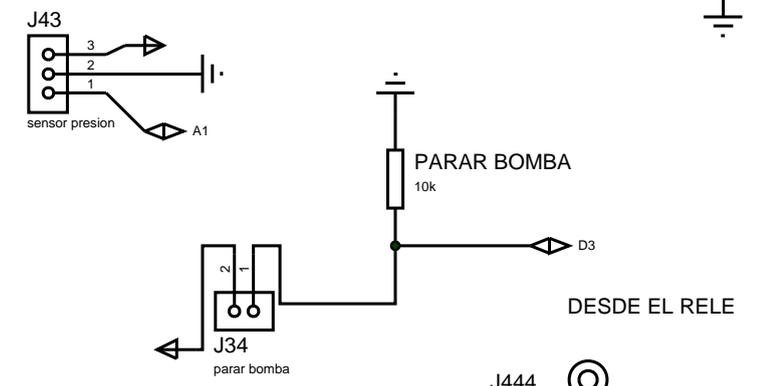
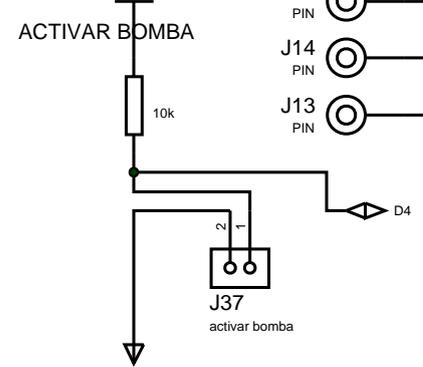
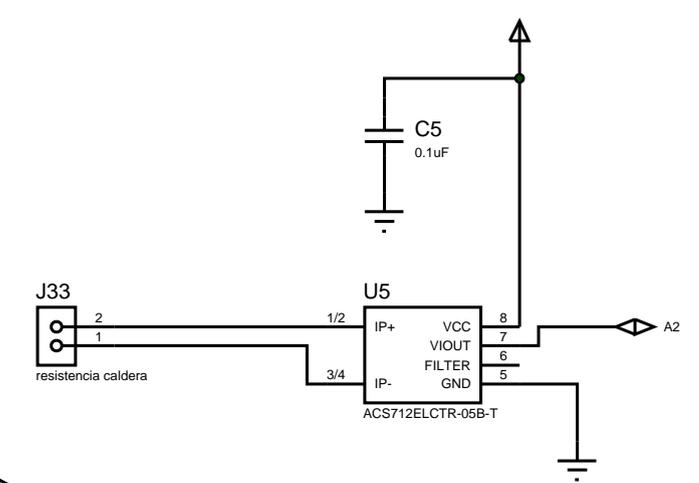
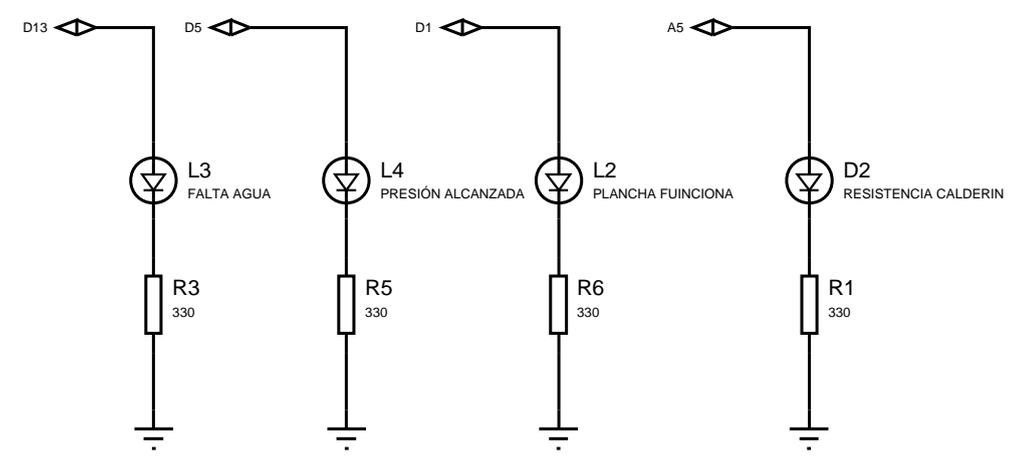
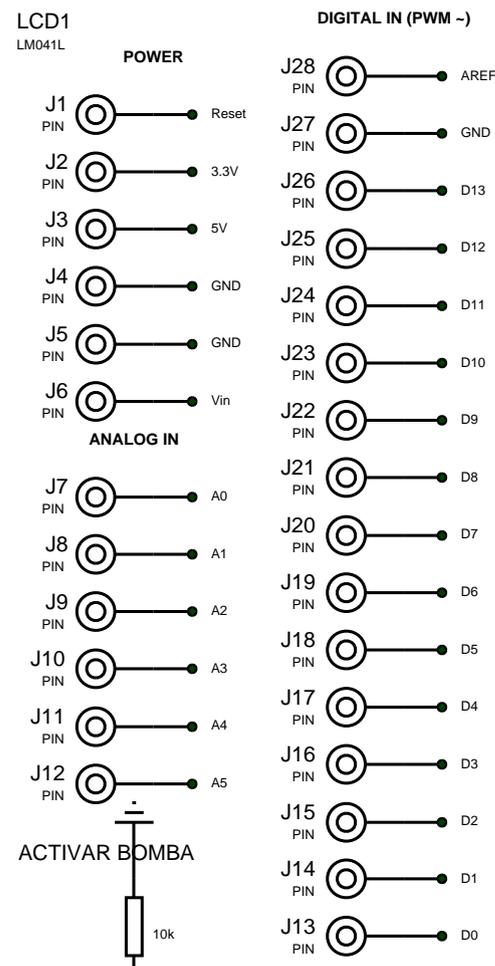
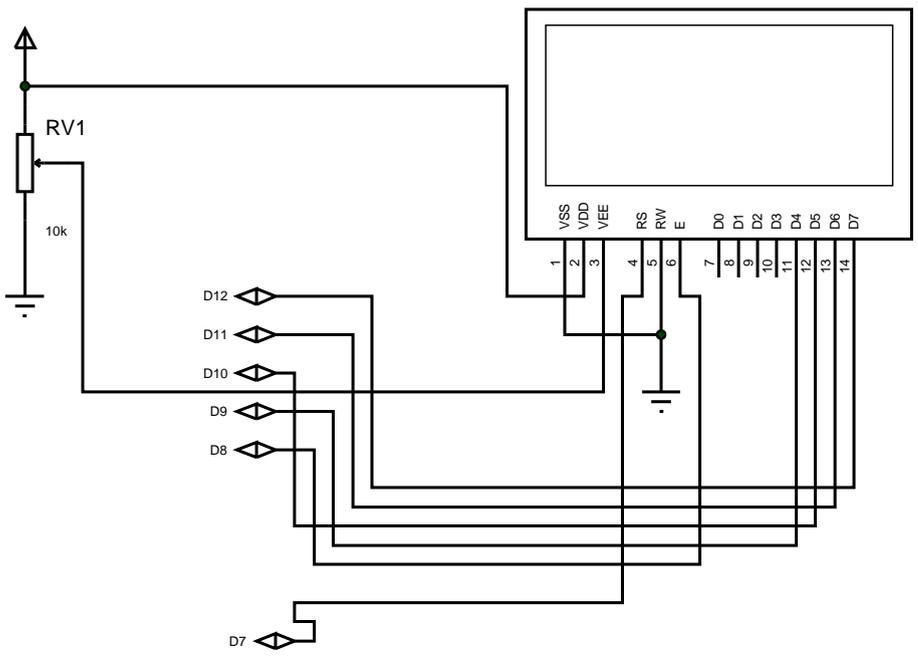
9



23/09/2013	UNIVERSIDAD POLITÉCNICA DE CARTAGENA	
Adrián Rivera Sánchez	Departamento de Tecnología Electrónica	
	Diseño de la tarjeta de control de un sistema de planchado industrial	
	I.T.I en Electrónica Industrial	PLANO 01



23/09/2013	UNIVERSIDAD POLITÉCNICA DE CARTAGENA	
Adrián Rivera Sánchez	Departamento de Tecnología Electrónica	
	Diseño de la tarjeta de control de un sistema de planchado industrial	
	I.T.I en Electrónica Industrial	PLANO 02



23/09/2013	UNIVERSIDAD POLITÉCNICA DE CARTAGENA	
Adrián Rivera Sánchez	Departamento de Tecnología Electrónica	
	Diseño de la tarjeta de control de un sistema de planchado industrial	
	I.T.I en Electrónica Industrial	PLANO 03

0

0

1

1

2

2

3

3

4

4

5

5

6

6

7

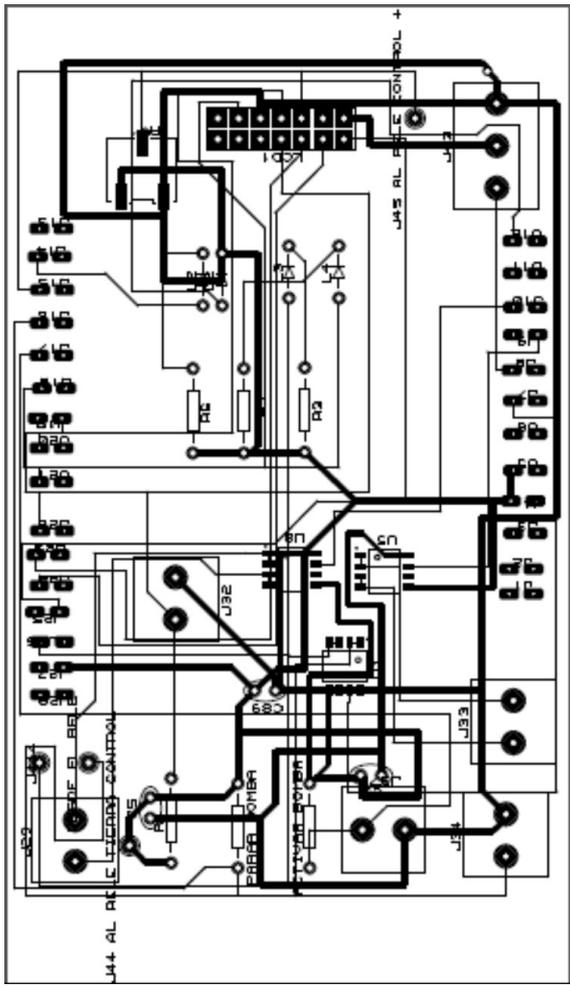
7

8

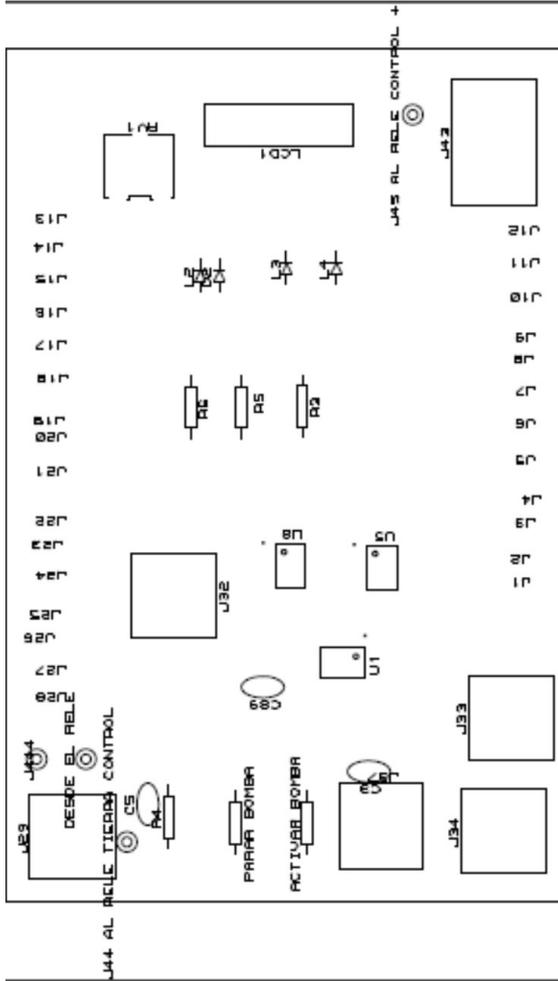
8

9

9



23/09/2013	UNIVERSIDAD POLITÉCNICA DE CARTAGENA	
Adrián Rivera Sánchez	Departamento de Tecnología Electrónica	
	Diseño de la tarjeta de control de un sistema de planchado industrial	
	I.T.I en Electrónica Industrial	PLANO 04



23/09/2013	UNIVERSIDAD POLITÉCNICA DE CARTAGENA	
Adrián Rivera Sánchez	Departamento de Tecnología Electrónica	
	Diseño de la tarjeta de control de un sistema de planchado industrial	
	I.T.I en Electrónica Industrial	PLANO 05

3. PLIEGO DE CONDICIONES

3.1. Características del calderín y dimensionamiento de las tuberías a usar.

Las calderas de vapor son recipientes a presión en los cuales se produce la vaporización de una masa de agua a alta temperatura a través de una aportación de calor. El gran incremento de volumen que se produce en el cambio de estado junto a la presión de trabajo y la temperatura, hacen que presenten una apreciable peligrosidad hacia personas y bienes.

Es por ello que estas instalaciones están sometidas a una amplia y en ocasiones compleja Reglamentación Oficial. Esta Reglamentación contempla todas las fases por las que pasa el equipo como son:

- Diseño
- Fabricación
- Controles de calidad y conformidad
- Ubicación
- Instalación
- Revisiones Oficiales
- Mantenimiento periódico

Así mismo la Comunidad Europea hace que se tienda a la unificación de la Normativa en todos los países miembros. El primer paso ha sido la aprobación de la Directiva del Parlamento Europeo y Consejo N° 97/23/CE recogida en el Real Decreto 769/1999. Esta directiva contempla los aspectos relacionados con el diseño, fabricación y evaluación de conformidad, quedando derogado el anterior Reglamento de Aparatos a Presión español (RD 1244/1979) con fecha 29 de mayo de 2002. En cuanto al resto de normas aplicables, como son las ITC, estas se mantienen en vigor.

Los aspectos reglamentarios más importantes en cuanto a las seguridades y al mantenimiento y conservación de las calderas, se dictan en la **ITC MIE AP1**. A través de esta ITC se describen el tipo y número de seguridades que debe incorporar una caldera dependiendo de la categoría y el tipo de vigilancia a la cual este sometida. También se reglamentan entre otros, aspectos como las revisiones e inspecciones periódicas que debe realizar el usuario, las condiciones que debe tener la sala de calderas o la calificación de los operadores y las obligaciones del usuario.

A modo de resumen las principales normas de aplicación actual y sus aspectos básicos reglamentados son:

NORMATIVA	PRICIPALES ASPECTOS REGLAMENTADOS
<i>Directiva Europea 97/23/CE</i>	<ul style="list-style-type: none"> - Diseño - Fabricación - Evaluación de conformidad
<i>ITC MIE AP1 (28/3/85), relativa a Calderas,</i>	<ul style="list-style-type: none"> - Pruebas oficiales y periódicas - Salas de Calderas - Elementos de seguridad (Válvulas, control de nivel, de presión...) - Homologación de Operadores de Calderas - Obligaciones del Usuario - Tratamiento del agua
<i>Norma UNE 9075/</i>	<ul style="list-style-type: none"> - Tratamientos y Calidad del agua para Calderas
<i>Norma UNE 9001</i>	<ul style="list-style-type: none"> - Terminología y definiciones de calderas y accesorios
<i>Norma UNE 9-100</i>	<ul style="list-style-type: none"> - Prescripciones para válvulas de seguridad
<i>Disposiciones Vigentes sobre Combustión</i>	<ul style="list-style-type: none"> - Emisiones de contaminantes
<i>ITC MIE AP2</i>	<ul style="list-style-type: none"> - Tuberías de fluidos relativos a calderas

Para el cálculo de las tuberías según la norma **ITC MIE AP1** será necesaria saber la potencia térmica Q de la caldera en Kcal/h, ésta se calcula aplicando la siguiente fórmula:

$$0.8604206496 * \text{Potencia(en vatios)} = \text{Potencia Térmica(en Kcal/h)}$$

$$0.8604206496 * 4000 \text{ W} = 3441,68 \text{ Kcal/h}$$

Con este valor podemos calcular el diámetro interior de la tubería según la siguiente fórmula:

$$D = 15 + 1,5 \sqrt{\frac{Q}{1000}} \text{ en milímetros}$$

$$D = 17,78 \text{ mm}$$

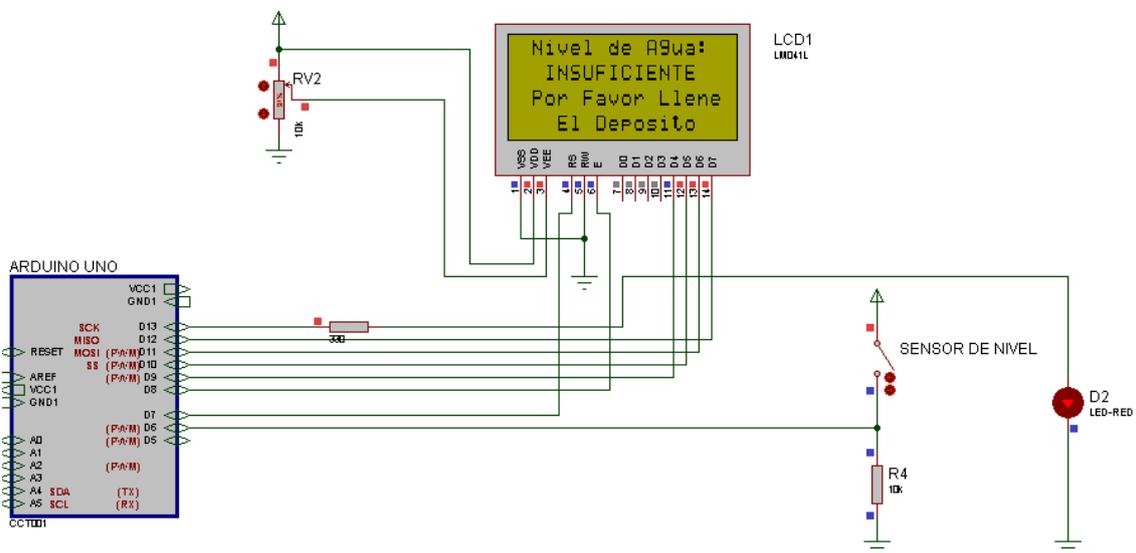
Por seguridad y según recomendaciones de la norma anteriormente citada se tomará como diámetro interior el valor 25 mm.

3.2. Funcionamiento de los sensores.

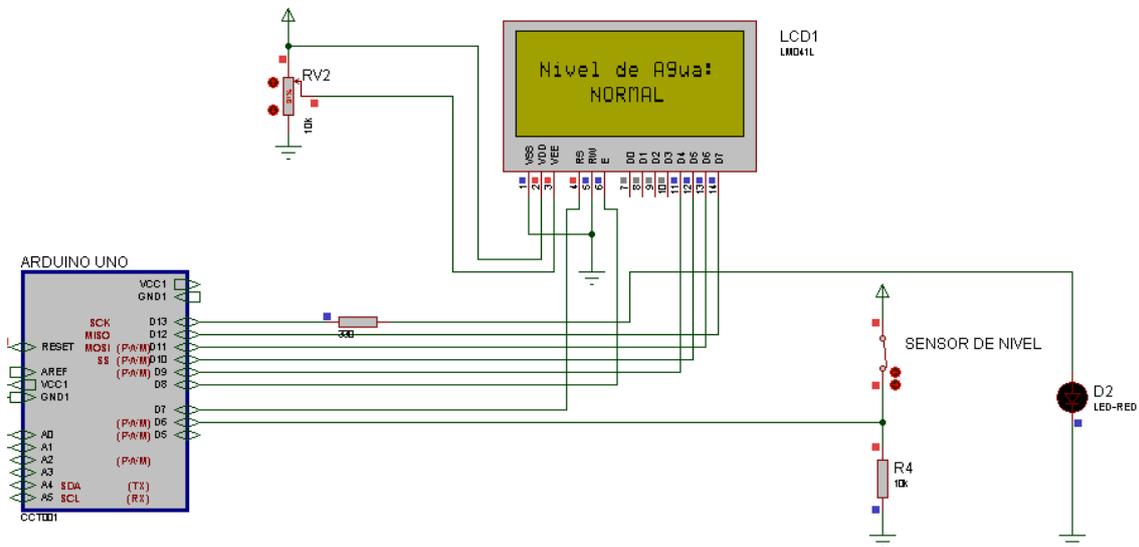
3.2.1. Sensor de nivel del depósito.

El sensor C-7235 según la norma ITC MIE AP1 se debe colocar a 70 mm de altura del depósito, dado que la alimentación de agua destilada se hará manualmente, lo interesante es saber cuándo el nivel de agua esté por debajo de este nivel de seguridad de 70 mm, para ello se utilizará una resistencia *pull-down* de 10 kΩ para ajustar el valor a un cero lógico o a un uno lógico.

La señal obtenida del sensor de nivel del depósito será llevada al pin digital número 6 de *Arduino UNO*, así cuando el nivel de agua esté por debajo del nivel que deseamos además de encenderse el diodo led de alarma D2 conectado al pin digital número 13, aparecerá en la pantalla LCD el siguiente mensaje:



En el caso contrario el led D2 se mantiene apagado y aparecerá en pantalla el siguiente mensaje:



3.2.2. Sensor de presión.

El sensor de presión *MPX5500* opera en un rango de medida de 0 a 500 kPa (100 kPa = 1 bar), con una sensibilidad de 9 mV/kPa y un error de *offset* que varía entre 0,08 y 0,313 V, para este caso y según los datos obtenidos en el simulador comprobamos que este error corresponde a un valor de 0,13 V, así que para una presión de trabajo de 3,5 bares (350 kPa) obtenemos el voltaje de salida del sensor según la siguiente relación:

$$\text{Salida del sensor} = 0,13 \text{ V} + 0,009 \text{ V} * 350 \text{ kPa}$$

$$\text{Salida del sensor} = 3,28 \text{ V}$$

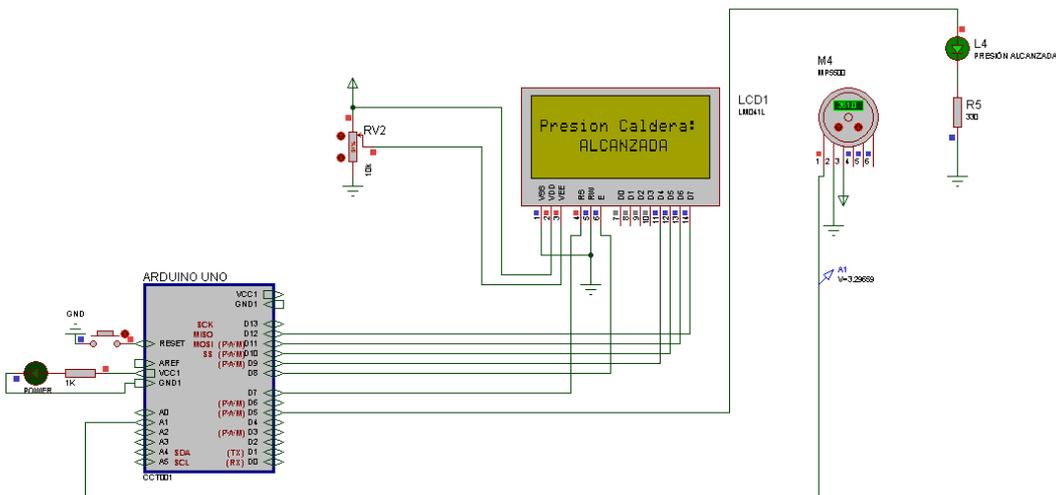
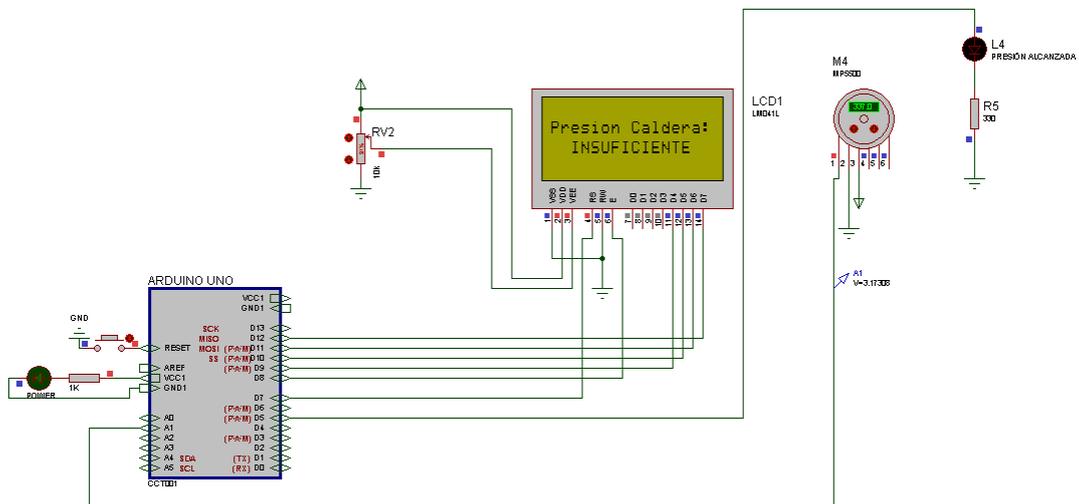
Este voltaje será llevado al pin analógico *A1* de *Arduino UNO*, que incorpora un convertor analógico digital de 10 bits, esto significa que convertirá voltajes de entre 0 y 5 voltios a un número entero entre 0 y 1023. Esto proporciona una resolución en la lectura de: 5 voltios / 1024 unidades, es decir, 0,0049 voltios (4.9 mV) por unidad. El convertor tarda aproximadamente 100 microsegundos en leer una entrada analógica por lo que se puede llevar una tasa de lectura máxima aproximada de 10.000 lecturas por segundo.

Aplicando una sencilla regla de 3 tenemos que:

$$X = \frac{1024 * 3,28 V}{5 V}$$

$$X = 672$$

El valor 672 es el que usaremos en el programa para indicar por medio de un mensaje en la pantalla y del encendido de un diodo led el momento en el que lleguemos a la presión de funcionamiento adecuada.



3.2.3. Sensor de nivel del calderín.

La sonda de nivel usada es la *LP10-4*, puede ser usada para proporcionar control de nivel *on/off* y alarmas de nivel en calderas de vapor, tanques y otros depósitos.

Las cuatro varillas son también adecuadas para uso como indicador eléctrico de nivel o alarmas con 3 señales luminosas. El cuerpo de la sonda está conectado a tierra a través de la conexión roscada de 1" BSP. La caldera o tanque cierra el circuito eléctrico conductivo. La sonda puede ser usada en tanques de material no conductivo (plástico, cemento) usando una de las varillas como masa o instalando una varilla o platina de tierra independiente. La sonda *LP10-4* dispone de cuatro varillas roscadas que detectan el nivel, que deben cortarse durante la instalación para proporcionar los niveles requeridos. Cuando el extremo de la varilla está sumergido se cierra el circuito eléctrico a tierra. Cuando la varilla está por encima del nivel, la resistencia a tierra aumenta, indicando al controlador que la varilla está fuera del líquido.

Las conexiones se realizan fácilmente a través de las conexiones rápidas suministradas. Los terminales están identificados por colores. Los terminales pueden ser extraídos fácilmente tirando hacia arriba del portaprensa, sin tener que manipular el prensacables. No se requiere mantenimiento especial, aunque los controles de nivel de caldera requieren comprobación periódica.

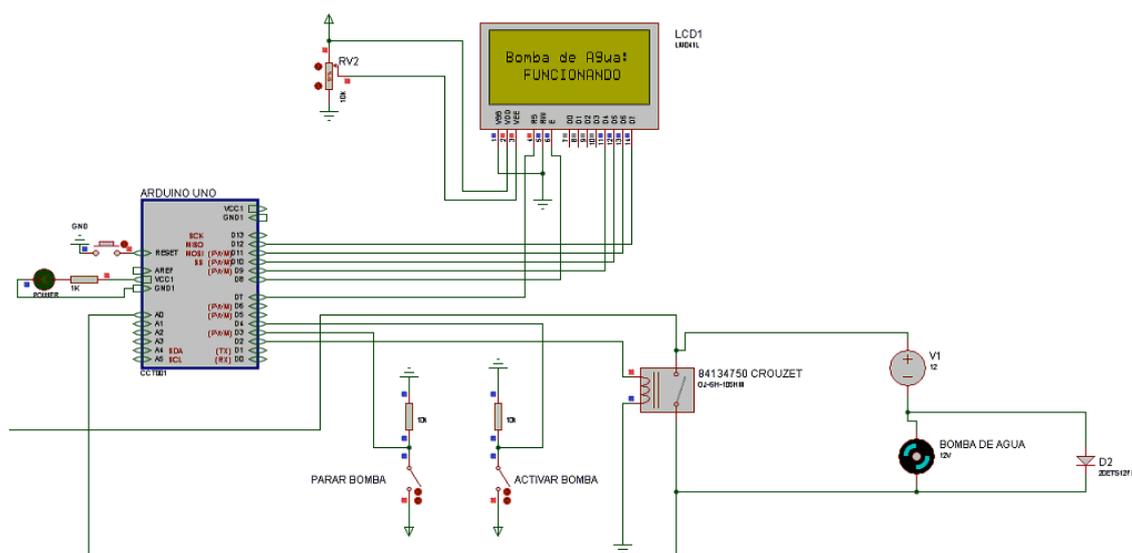
La forma de trabajo será como sigue, se colocará una sonda a una altura de 70 mm como indica la norma **ITC MIE API** que en el diagrama de simulación se define como *ACTIVAR BOMBA* e irá conectada al pin digital 4 de la placa y la otra sonda se colocará a $\frac{3}{4}$ de la altura del calderín y se definirá como *PARAR BOMBA* que estará en el pin digital 3 de la placa *Arduino UNO*.

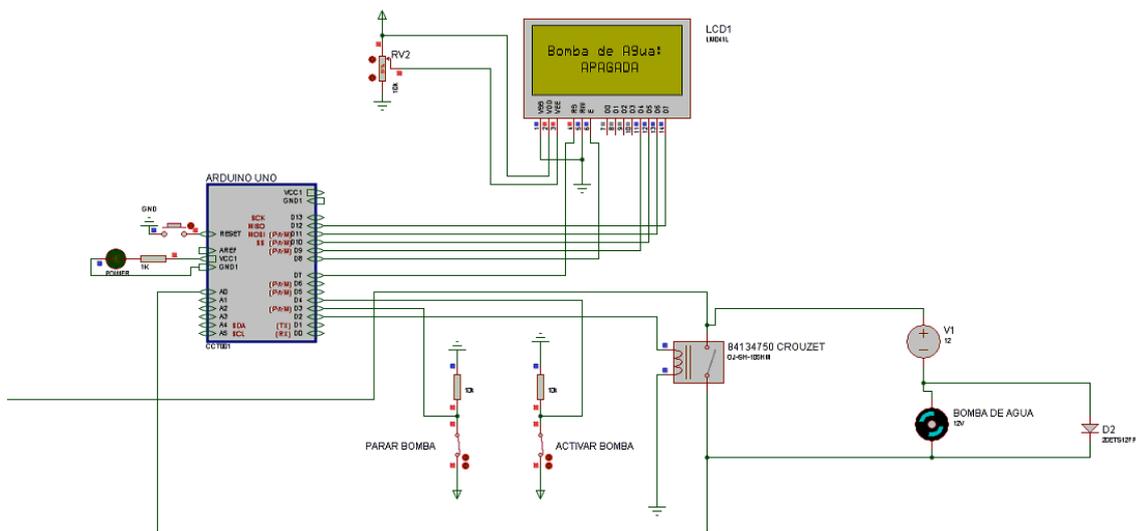
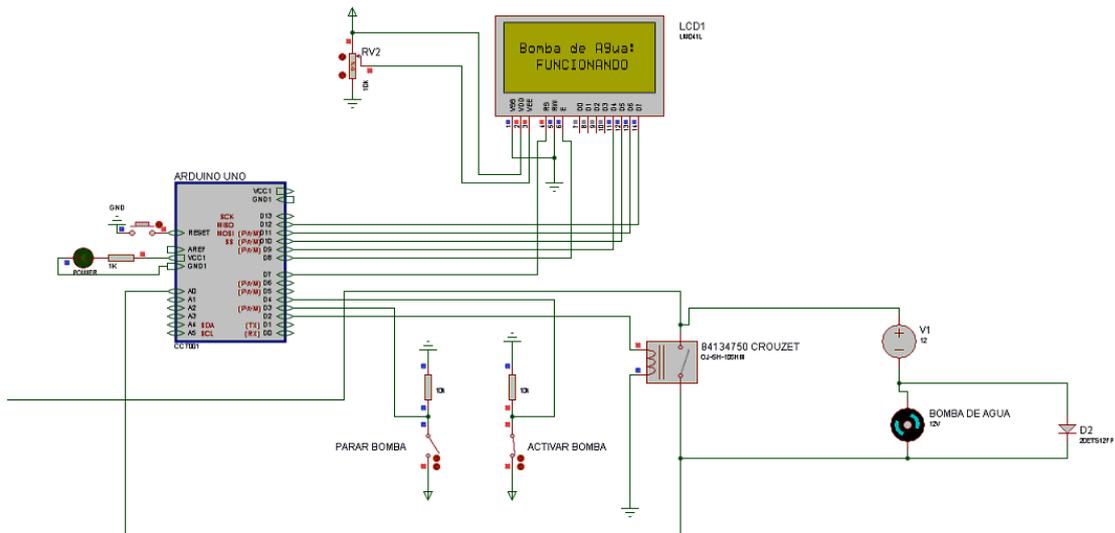
Ambas sondas ejercerán el control sobre el relé de estado sólido *84134750* y lograrán que se active o cierre controlando de esta manera el funcionamiento de la bomba de agua. La tabla de verdad de las sondas es la siguiente:

PARAR BOMBA	ACTIVAR BOMBA	SALIDA (D2)
0	0	1
0	1	1
1	0	0 (ERROR)
1	1	0

La condición PARAR BOMBA = 1 y ACTIVAR BOMBA = 0, no se puede dar, ya que es imposible físicamente que la sonda inferior no esté en contacto con el agua y la superior sí, en este caso la salida digital 2 se mantendrá en 0 como medida de seguridad. Para asegurar el correcto funcionamiento tenemos que conectar a cada sonda una resistencia *pull-up*.

Aunque no se aprecia en las siguientes capturas, el sensor de funcionamiento de la bomba está conectado, pero se ha preferido dejarlo fuera de la imagen para poder explicarlo con más detenimiento en la parte de los sensores de efecto Hall. Los mensajes de la pantalla y funcionamiento de la bomba son como siguen:





3.2.4. Sensores de funcionamiento.

Utilizaremos tres sensores *ACS712ELCTR-20A-T*, que presentan diferencias en la programación según el tipo de corriente que midamos, el modo de solucionar el problema según cada caso se presenta a continuación.

3.2.4.1. Sensor de funcionamiento en corriente continua.

Este es el caso más sencillo y el que usaremos en el caso de la bomba de agua, al tratarse de corriente continua, y sabiendo que la salida del sensor según su hoja de características es de 2,5 V para una corriente de 0 A, con una sensibilidad de 100 mV/A, la salida del sensor de funcionamiento se registrará por la siguiente fórmula:

$$\textit{Salida del sensor de funcionamiento} = 2,5 V + 0,100 V * \textit{Corriente (A)}$$

Así, y dado que el consumo medio de la bomba de agua es de 5,6 A, tomaremos como valor de referencia 2 A de consumo, así sabremos que la bomba está funcionando cuando en el sensor tengamos como mínimo el siguiente voltaje:

$$\textit{Salida del sensor de funcionamiento} = 2,5 V + 0,100 V * 2 A$$

$$\textit{Salida del sensor de funcionamiento} = 2,7 V$$

Este voltaje será llevado al pin analógico A0 de *Arduino UNO*, que incorpora un conversor analógico digital de 10 bits, esto significa que convertirá voltajes de entre 0 y 5 voltios a un número entero entre 0 y 1023 como habíamos mencionado anteriormente en la toma de datos del sensor de presión.

Aplicando una sencilla regla de 3 tenemos que:

$$X = \frac{1024 * 2,70 V}{5 V}$$

$$X = 553$$

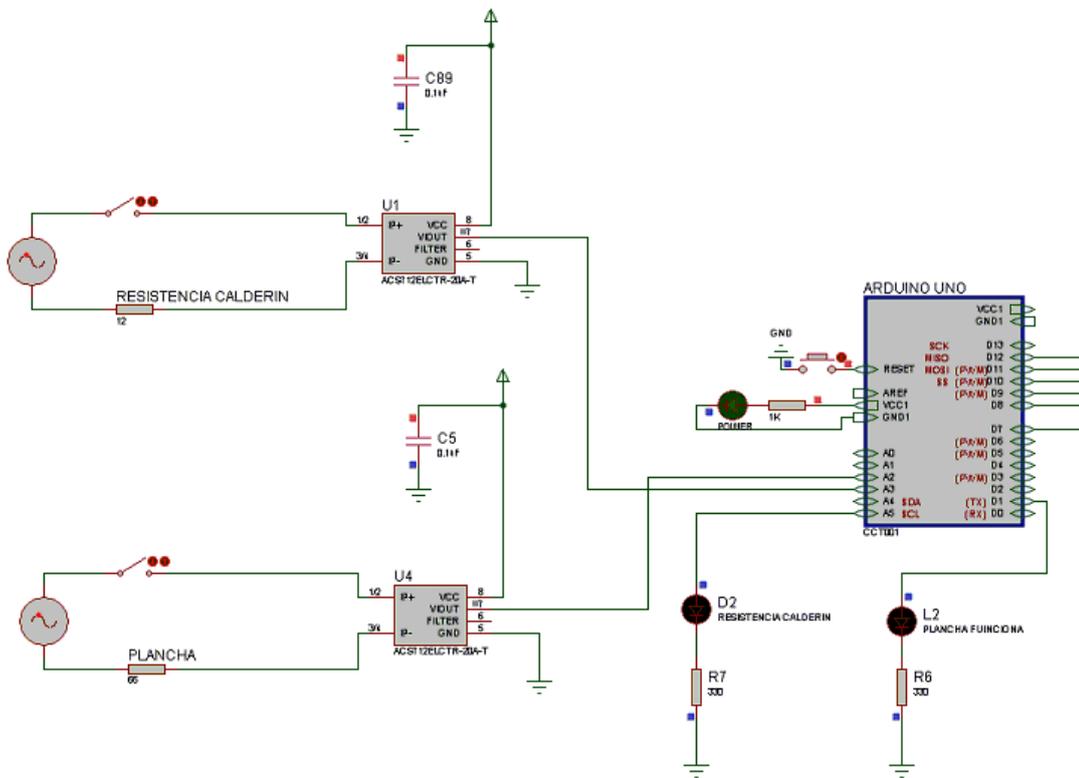
Lo que significa que para valores iguales o superiores a 553 (siempre teniendo en cuenta una circulación positiva de la corriente) la bomba funciona activando los mensajes mostrados anteriormente en pantalla.

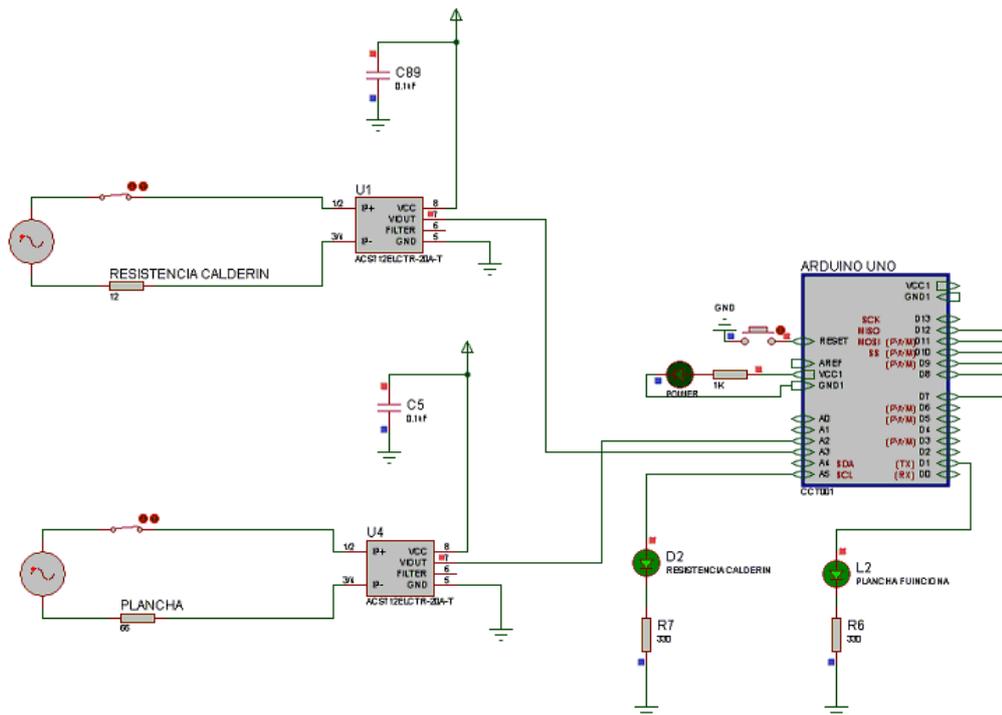
3.2.4.2. Sensor de funcionamiento en corriente alterna.

Este modo de funcionamiento es el utilizado para poder medir la corriente circulante en el caso de la plancha y de la resistencia del calderín, al trabajar en corriente alterna es necesario medir la corriente *I_{rms}*, almacenarla luego en una variable interna del programa y compararla con un valor mínimo de funcionamiento que hemos definido en los dos casos con el valor de 0,15, cuando sobrepase este valor tendremos una corriente

circulante y podremos decir sin ninguna duda que la plancha y la resistencia del calderín están funcionando.

Los sensores de funcionamiento se conectarán en los pines analógicos A2 y A3 de *Arduino UNO* y las luces indicadoras de funcionamiento en el pin digital 1 y el pin analógico A5 (usado en este caso como salida digital) y los mensajes en pantalla serán como sigue:





3.3. Funcionamiento de la pantalla LCD.

La pantalla LCD elegida, es una pantalla de 4x16 líneas con retroiluminación controlado por un potenciómetro conectado entre los pines 2 y 3 para regular el contraste, funciona con el controlador *Hitachi HD44780*, uno de los controladores para pantallas de cristal líquido más comunes, con las siguientes características principales:

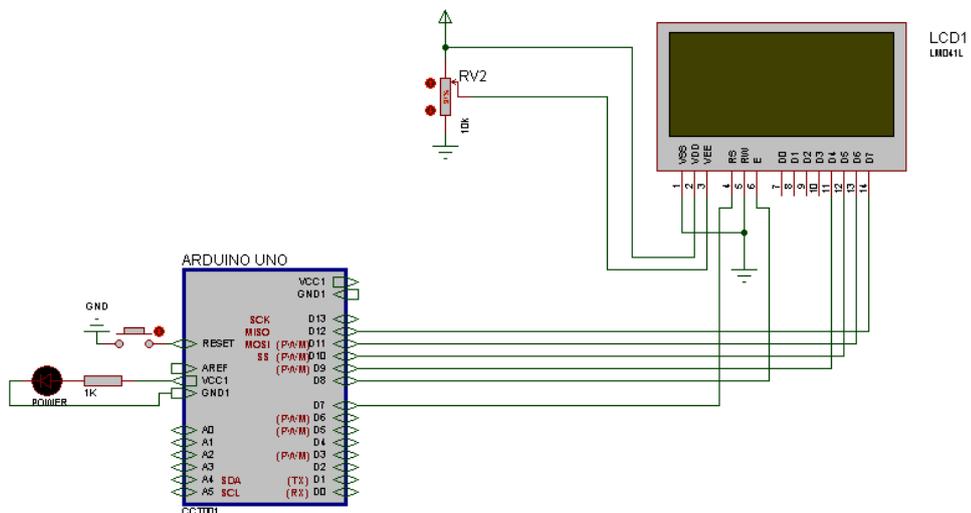
- 16 caracteres por cada línea hasta un total de 64 (los hay con más y menos líneas).
- Encendido/apagado del display.
- Parpadeo del cursor.
- Desplazamiento izquierda/derecha.
- Reconocimiento de ASCII estándar.
- Soporta 132 caracteres alfanuméricos y 32 de control.

Se ha elegido este controlador dado que la plataforma *Arduino* cuenta con una librería específica llamada *LiquidCrystal* que le permite controlar displays LCD basados en el chipset *Hitachi HD44780*. La biblioteca trabaja en modo 4-bit o en 8-bit (es decir, por medio de 4 u 8 líneas de datos, además de RS, ENABLE, y, opcionalmente, las líneas de control RW). En nuestro caso para poder dejar la mayor cantidad de pines posible se

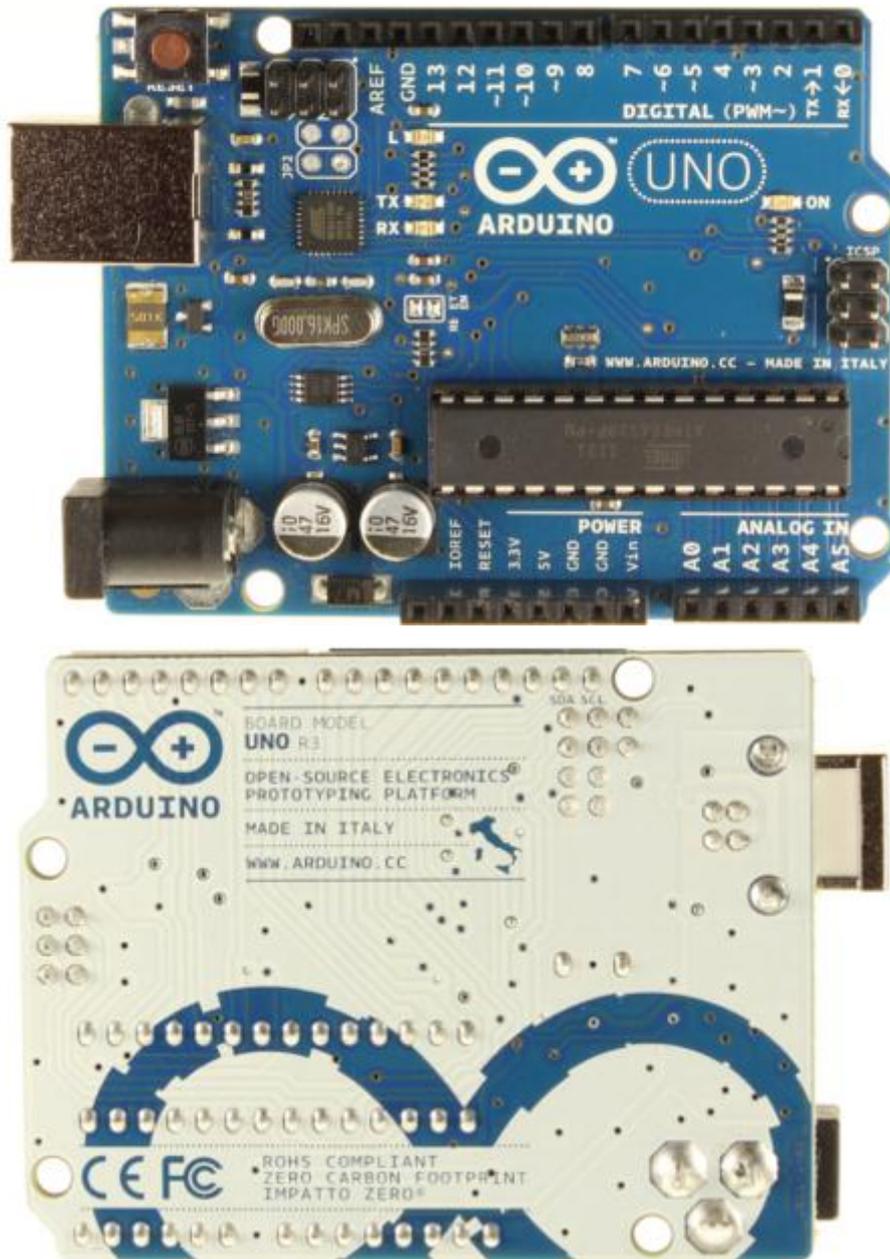
elegirá el modo de trabajo de 4 bits, teniendo en cuenta la siguiente información de conexionado.

Nº PIN	SÍMBOLO	DESCRIPCIÓN
1	V _{SS}	Alimentación negativa (masa)
2	V _{DD}	Alimentación positiva (+5Vdc)
3	V _O	Ajuste del contraste
4	RS	Selección de Registro (Register Select) RS = 1 ⇒ Registro de Datos RS = 0 ⇒ Registro de Instrucciones
5	R/ \bar{W}	Lectura / Escritura (Read/Write) R/ \bar{W} = 1 ⇒ Lectura R/ \bar{W} = 0 ⇒ Escritura
6	E	Habilitación del display (Enable) E = 1 ⇒ Habilitado E = 0 ⇒ Deshabilitado
7-14	DB[0..7]	Pines del 0 al 7 del bus de datos bidireccional

Y la conexión con la placa *Arduino UNO* será:



3.4. Funcionamiento de la placa Arduino UNO.



Se ha elegido como base fundamental del proyecto la plataforma de desarrollo *Arduino* por su filosofía de código abierto, bajo coste, facilidad de desarrollo y sobre todo por la existencia de una comunidad muy grande de aficionados y profesionales de la electrónica que trabajan constantemente en el desarrollo de nuevas aplicaciones y proyectos.

Dentro de todas las posibles elecciones dentro de la familia *Arduino* se ha elegido la placa *Arduino UNO*, que por sus características que se detallarán a continuación es la elección más adecuada para el desarrollo de la tarjeta de control.

La placa *Arduino UNO* está basada en el microcontrolador *ATmega328* de *Atmel* tiene autoselección del voltaje de alimentación (DC/USB) gracias a un chip MOSFET incluido en la placa. Además, dispone del nuevo *bootloader OptiBoot* que permite cargar programas a 115Kbps. El *bootloader* ocupa **512bytes**. Incluye función de auto-reset, protección de sobrecargas, conector USB para programarlo, totalmente montado con componentes miniatura.

Cuenta con 14 entradas / salidas digitales (de las cuales 6 pueden utilizarse como salidas PWM), 6 entradas analógicas, un oscilador de 16, conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reset.

3.5. Entorno de programación de Arduino.

Arduino utiliza para escribir el software un programa al que denomina *sketch*. Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie.

Se encontrarán otros comandos en los cinco menús: *File*, *Edit*, *Sketch*, *Tools*, *Help*. Los menús son sensibles al contexto, lo que significa que estarán disponibles sólo los elementos relevantes para la tarea que esté realizando en ese momento:

-Edit

Copy for Discourse: Copia el código del *sketch* en el portapapeles con el formato adecuado para publicarlo en un foro, incluyendo la sintaxis coloreada.

Copy as HTML: Copia el código de un programa al portapapeles en formato HTML, adecuándolo para incrustarlo en una página web.

-Sketch

Verify/Compile: Verifica los errores del programa.

Import Library: Añade una librería al *sketch* insertando la sentencia *#include* en el código.

Show Sketch Folder: Abre la carpeta de programas en el escritorio.

Add File: Añade un fichero fuente al programa (se incluirá desde su ubicación actual). El fichero aparece en una nueva pestaña en la ventana del programa. Los ficheros pueden ser quitados del programa utilizando el menú *tab*.

-Tools

Auto Format: Da formato al código proporcionando estética: por ejemplo realiza tabulaciones entre la apertura y cierre de llaves, y las sentencias que tengan que ser tabuladas lo estarán.

-Board

Selecciona la placa que se esté usando. En este caso *Arduino UNO*.

Serial Port: Este menú contiene todos los dispositivos de serie (reales o virtuales) del equipo. Se refrescará automáticamente cada vez que se abra el menú *Tools*.

Burn Bootloader: Este elemento del menú le permite grabar un gestor de arranque (bootloader) dentro del micro-controlador de la placa Arduino. Aunque no es un requisito para el normal funcionamiento de la placa Arduino, será útil si se compra un nuevo ATmega (el cual viene normalmente sin gestor de arranque). Hay que asegurarse de haber seleccionado la placa correcta en el menú *Boards* antes de grabar el *bootloader*.

-Sketchbook

El entorno de Arduino incluye el concepto de *sketchbook* que es el lugar estándar para el almacenamiento de sus programas. Los sketches dentro de su *sketchbook* pueden abrirse desde el menú *File > Sketchbook* o desde el botón de la barra de herramientas *Open*. La primera vez que se arranque el software Arduino, se creará un directorio para el *sketchbook*. Se puede visualizar o cambiar su localización dentro de *sketchbook location* desde el apartado *Preferences*.

-Tabs, Multiple Files, and Compilation (Pestañas, ficheros múltiples y compilación)

Permite manejar *sketches* con más de un fichero (cada uno de los cuales aparece en su pestaña). Pueden ser normalmente ficheros de código Arduino (no extensiones), ficheros C (extensiones .c), ficheros c++ (.cpp), o ficheros de cabecera (.h).

-Uploading (Volcado)

Antes de volcar su sketch, necesitará seleccionar los elementos correspondientes desde los menús *Tools > Board y Tools > Serial Port*.

-  **Verify/Compile**
Chequea el código en busca de errores.
-  **Stop**
Finaliza la monitorización serie y oculta otros botones
-  **New**
Crea un nuevo *sketch*.
-  **Open**
Presenta un menú de todos los programas *sketch* de su "sketchbook", (*librería de sketch*). Un click sobre uno de ellos lo abrirá en la ventana actual.
Nota: Debido a un error *bug* en Java, la barra de desplazamiento *scroll* no funciona en este menú; si necesita abrir un programa que no se vea en la lista, utilice *File | Sketchbook* en el menú.
-  **Save**
Salva el programa *sketch*.
-  **Upload to I/O Board**
Compila el código y lo vuelca en la placa E/S de Arduino. Ver volcado más detalles abajo.
-  **Serial Monitor**
Inicia la monitorización serie Monitorización Serie.

3.6. Programa de control de la placa Arduino UNO.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
int sen_llenado = 6;
int hall_01 = A0;
int presion = A1;
boolean motor_on = 4;
boolean motor_off = 3;
int estadosensor = 0;
int corrientemotor = 0;
int presioncaldera = 0;
int activacion = 0;
int apagado = 0;

const int currentPin = A2;
const unsigned long
tiempomuestreo=100000UL;
const unsigned long mummuestras =
250UL;
const unsigned long
intervalomuestras=tiempomuestreo/mummuestras;
const int adc_zero = 510;

const int currentPin2 = A3;
const unsigned longtiempomuestreo2=100000UL;
const unsigned long mummuestras2
=250UL;
const unsigned long
intervalomuestras2=tiempomuestreo2/mummuestras2;
const int adc_zero2 = 510; )

void setup ()
{

lcd.begin(16, 4);
pinMode(13, OUTPUT);
```

```

pinMode(sen_llenado, INPUT);
pinMode(5, OUTPUT);
pinMode(hall_01, INPUT);
pinMode(presion, INPUT);
pinMode(4, INPUT);
pinMode(3, INPUT);
pinMode(2, OUTPUT);
pinMode(1, OUTPUT);
pinMode(A5, OUTPUT);
}

```

```

void loop()
{
  estadosensor=digitalRead(sen_llenado); //usar resistencia
  de pull down//
  if (estadosensor == HIGH)
  {
    digitalWrite(13, LOW);

    lcd.setCursor(0, 0);
    lcd.print("                ");
    lcd.setCursor(0, 1);
    lcd.print(" Nivel de Agua: ");
    lcd.setCursor(-4, 2);
    lcd.print("    NORMAL    ");
    lcd.setCursor(-4, 3);
    lcd.print("                ");
    delay(400);
  }

  if (estadosensor == LOW)
  {
    digitalWrite(13, HIGH);

    lcd.setCursor(0, 0);
    lcd.print(" Nivel de Agua: ");
    lcd.setCursor(0, 1);
    lcd.print("  INSUFICIENTE  ");
    lcd.setCursor(-4, 2);
    lcd.print(" Por Favor Llene");
    lcd.setCursor(-4, 3);
    lcd.print("   El Deposito  ");
    delay(400);
  }
  corrientemotor=analogRead(hall_01);
  if (corrientemotor <= 553)
  {

```

```

    delay(1000);
    lcd.setCursor(0, 0);
        lcd.print("                ");
    lcd.setCursor(0, 1);
        lcd.print(" Bomba de Agua: ");
    lcd.setCursor(-4, 2);
        lcd.print("      APAGADA      ");
    lcd.setCursor(-4, 3);
        lcd.print("                ");
    delay(400);
}
if (corrientemotor > 553)
{
    delay(1000);
    lcd.setCursor(0, 0);
        lcd.print("                ");
    lcd.setCursor(0, 1);
        lcd.print(" Bomba de Agua: ");
    lcd.setCursor(-4, 2);
        lcd.print("      FUNCIONANDO  ");
    lcd.setCursor(-4, 3);
        lcd.print("                ");
    delay(400);
}

presioncaldera=analogRead(presion);

if (presioncaldera > 672) //350 bares, 3,29 voltios
{
    digitalWrite(5, HIGH);
    lcd.setCursor(0, 0);
        lcd.print("                ");
    lcd.setCursor(0, 1);
        lcd.print("Presion Caldera:");
    lcd.setCursor(-4, 2);
        lcd.print("      ALCANZADA    ");
    lcd.setCursor(-4, 3);
        lcd.print("                ");
    delay(400);
}

if (presioncaldera <= 672)//350 bares 3,29 voltios
{
    digitalWrite(5, LOW);
    lcd.setCursor(0, 0);
        lcd.print("                ");
    lcd.setCursor(0, 1);
        lcd.print("Presion Caldera:");
    lcd.setCursor(-4, 2);
        lcd.print("      INSUFICIENTE ");
}

```

```

        lcd.setCursor(-4, 3);
        lcd.print("                ");
        delay(400);
    }

    activacion = digitalRead(motor_on);
    apagado = digitalRead(motor_off);
    if (apagado == LOW && activacion == LOW)
    {
        digitalWrite(2, HIGH);
        delay(100);
    }
    if (apagado == LOW && activacion == HIGH)
    {
        digitalWrite(2, HIGH);
        delay(100);
    }
    if (apagado == HIGH && activacion == LOW) //error no se
    puede dar esta condicion sonda arriba cubierta, abajo sin
    agua
    {
        digitalWrite(2, LOW); //bomba apagada por
    seguridad
        delay(100);
    }
    if (apagado == HIGH && activacion == HIGH)
    {
        digitalWrite(2, LOW);
        delay(100);
    }

    unsigned long currentAcc = 0;
    unsigned int count = 0;
    unsigned long prevMicros = micros() - intervalomuestras ;
    while (count < mummuestras)
    {
        if (micros() - prevMicros >= intervalomuestras)
        {
            int adc_raw = analogRead(currentPin) - adc_zero;
            currentAcc += (unsigned long)(adc_raw * adc_raw);
            ++count;
            prevMicros += intervalomuestras;
        }
    }

    float rms = sqrt((float)currentAcc/(float)mummuestras) *
    (75.7576 / 1024.0);
    if(rms > 0.15)
    digitalWrite(1, HIGH);
    delay(50);
    if(rms <= 0.15)

```

```

digitalWrite(1, LOW);
delay(50);

unsigned long currentAcc2 = 0;
unsigned int count2 = 0;
unsigned long prevMicros2 = micros() - intervalomuestras2
;
while (count2 < mummuestras2)
{
    if (micros() - prevMicros2 >= intervalomuestras2)
    {
        int adc_raw2 = analogRead(currentPin2) - adc_zero2;
        currentAcc2 += (unsigned long) (adc_raw2 * adc_raw2);
        ++count2;
        prevMicros2 += intervalomuestras2;
    }
}

float rms2 = sqrt((float)currentAcc2/(float)mummuestras2)
* (75.7576 / 1024.0);
if(rms2 > 0.15)
digitalWrite(A5, HIGH);
delay(50);
if(rms2 <= 0.15)
digitalWrite(A5, LOW);
delay(50);
}

```

3.7. Puesta en marcha del sistema.

Se detallan a continuación los pasos a seguir para la puesta en marcha del sistema de planchado.

1. Activar el interruptor del calderín y de la plancha en el panel de control.

**Después de que se haya activado los siguientes pasos son automáticos.*

2. El indicador de la bomba de agua se enciende. La bomba empieza a estar operativa y a alimentar al calderín hasta que reciba la información de que está lleno, entonces para.

3. El indicador de nivel de falta de agua se mantiene apagado. El calentador está encendido. Después de 3-5 minutos el vapor de agua llega a 3 bares y está en su etapa operativa.

4. Durante la operación, los indicadores de la bomba de agua, del nivel de agua normal y del calentador se iluminarán y apagarán repetidamente.

5. Si el nivel de agua en el calderín está bajando se activará la bomba, en caso de que el agua falte en el depósito el indicador (led rojo de alarma) se iluminará. El agua alimentará al calderín automáticamente.

6. En caso de que falte agua en el depósito por favor, añádala rápidamente.

4. PRESUPUESTO

PRESUPUESTO DE GASTOS
 Universidad Politécnica de
 Cartagena

26/09/2013

Centro de Planchado	Cantidad	Precio Unidad	Total
Calderín	1	232,00 €	232,00 €
Plancha	1	130,00 €	130,00 €
Resistencia Calefactora	1	56,00 €	56,00 €
Bomba de agua Shurflo	1	160,00 €	160,00 €
Depósito BCG-5	1	14,00 €	14,00 €
Circuito de Control	Cantidad	Precio Unidad	Total
Arduino Uno	1	13,00 €	13,00 €
Sensor ACS712-20A	3	1,20 €	3,60 €
Pantalla LCD 16x4	1	7,00 €	7,00 €
Diodo LED verde 20 mA	3	0,30 €	0,90 €
Diodo LED rojo 20 mA	1	0,30 €	0,30 €
Resistencia 330 Ω 1/4 W	4	0,30 €	1,20 €
Resistencia 10 k Ω 1/4 W	4	0,30 €	1,20 €
Sensor de presión MPX5500	1	12 €	12 €
Conectores placa	1	7 €	7 €
Relé Crouzet 84134750	1	34,30 €	34,30 €
Sensor C-7235	1	13 €	13 €
Sonda LP 10-4	1	27 €	27 €
Condensador cerámico 0,3 uF	3	0,20 €	0,60 €
Gastos totales			713,10 €

5. BIBLIOGRAFÍA

-Manual de programación Arduino.
Traducido y adaptado por José Manuel Luis Gutierrez.

-Prácticas con Arduino Nivel I.
José Manuel Luis Gutierrez.

-30 Arduino Projects for the Evil Genius.
Simon Monk.

-Learn Diy Electronics with the Sparkfun Inventor's Guide.
Sparkfun Electronics.

Varias webs:

[http://forum.arduino.cc/index.php?PHPSESSID=5beb3d3e7324733514e4d92e4e3710bb
&board=32.0](http://forum.arduino.cc/index.php?PHPSESSID=5beb3d3e7324733514e4d92e4e3710bb&board=32.0)

<https://www.dropbox.com/sh/79k6wxoxwuj7164/lxwNWqK4yK>

<http://www.dtic.upf.edu/~jlozano/interfaces/interfaces5.html>

<http://playground.arduino.cc/Main/InterfacingWithHardware>