



**Materia: Proyecto y Diseño Electrónico**

**Profesor: Juan Manuel Mercadin**

**Colegio: E.E.S.T. N°3." Domingo Faustino Sarmiento"**

**Año y División: 7°3**

-----  
**Actividad N°1: Central de control para portón automático**

**Grupo: Mauricio Choque y Jesús Mella Soria**

**Fecha: 6/7/2024**

### **Indicé general**

<b>PORTADA.....</b>	<b>1</b>
---------------------	----------

<b>INDICÉ GENERAL .....</b>	<b>2</b>
<b>OBJETIVOS.....</b>	<b>2</b>
<b>FUNCIONAMIENTO.....</b>	<b>3</b>
<b>GRAFICO DE GANTT.....</b>	<b>4</b>
<b>CODIGO DEL PROYECTO.....</b>	<b>5</b>
<b>PRESUPUESTO.....</b>	<b>9</b>
<b>VISTAS.....</b>	<b>10</b>

## **Introducción**

Las estaciones de soldadura son herramientas esenciales para los aficionados y profesionales de la electrónica. Permiten realizar soldaduras precisas y controladas, lo que es fundamental para obtener resultados de alta calidad. Las estaciones de soldadura tradicionales utilizan un potenciómetro para ajustar la temperatura, pero esto puede ser impreciso y poco confiable.

En este informe, se presenta el diseño y construcción de una estación de soldadura controlada por Arduino Uno y una termocupla. Esta estación permite ajustar la temperatura con precisión y mantenerla estable durante el proceso de soldadura.

## **Objetivos**

1. Explorar el área de diseño de instrumentos y objetos propuestos por el técnico.
2. Articular los conocimientos en los tres años de especialización.
3. Realizar trabajos vinculados con la electrónica industrial.
4. Poder ajustarse a los tiempos establecidos, como en la práctica profesional.
5. Resolver problemáticas referidas al campo de la Electrónica.
6. Utilización y perfeccionamiento de circuitos ya diseñados.

## **Requisitos y requerimientos:**

1. Deberá poder accionar un motor de fase partida de 220V como también así poder controlar el sentido de giro del motor.

2. Deberá contar con entradas para poder detectar el accionamiento de los finales de carrera del portón.
3. Tendrá que tener la posibilidad de accionamiento local a través de un pulsador externo y remoto mediante un control remoto RF.
4. Poseer la habilidad de poder registrar y almacenar usuarios para poder controlar la central (opcional).
5. Deberá contar con 2 modos de funcionamiento; el primero es el modo manual (se abre y cierra cada vez que se presiona el pulsador o el control remoto) y el segundo es el modo temporizado (se abre el portón y al cabo de un cierto tiempo configurable se debe cerrar automáticamente).
6. Utilizar en la medida de lo posible materiales reciclados.

### **Funcionamiento**

La central deberá ser controlada de forma local y remota. Mediante un pulsador en la placa se accederá al modo de funcionamiento manual y al temporizado. Con otro botón se deberá poder hacer los registros de nuevos usuarios (en caso de incluir el registro de usuarios). Cuando el motor es accionado solo podrá ser detenido si alguna de las siguientes condiciones se cumple.

1. Se acciona el final de carrera correspondiente hacia donde el motor fue accionado.
2. Se vuelve a presionar el botón del control remoto RF o el botón externo.
3. Se cumple el tiempo en caso de estar trabajando en modo temporizado.

Una vez que se cumple alguna de las condiciones anteriores, el motor deberá quedar preparado para que el siguiente accionamiento, haciendo que se mueva en la dirección contraria a la anterior.

### **Componentes de Entrega**

Se deberá entregar la plaqueta de la central de control, deberá ser completamente funcional, además, deberá contar con un manual de instrucciones para el usuario, así como también un informe correspondiente a todo el proceso de diseño, desarrollo e implementación junto con los esquemáticos y códigos correspondientes. La etapa de potencia para controlar el motor será provista por el docente.

### **Gráfico de Gantt**

ETAPAS	Inicio	Final
--------	--------	-------

<b>Manual</b>	<b>11/3/2024</b>	<b>15/4/2024</b>
<b>Temporización</b>	<b>15/4/2024</b>	<b>13/5/2024</b>
<b>Control RF</b>	<b>13/5/2024</b>	<b>10/6/2024</b>
<b>Entrega final</b>	<b>1/7/2024</b>	<b>1/7/2024</b>

<b>Tiempo Trabajados</b>																		
<b>Etapas</b>	<b>Marzo</b>				<b>Abril</b>				<b>Mayo</b>				<b>Junio</b>				<b>Julio</b>	
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	
<b>Manual</b>																		
<b>Temporización</b>																		
<b>Control RF</b>																		
<b>Entrega final</b>																		

**Codigo del proyecto**



```
1 //Este código se encarga de controlar el funcionamiento de un portón eléctrico automático.
2 // Implementa la lógica necesaria para abrir, cerrar y detener el movimiento
3 //en respuesta a diferentes señales de entrada.
4 //Librerías
5 #include <RCSwitch.h>
6
7 //Control rf
8 const uint32_t CODIGO = 16242856;
9 long dato;
10 bool estado = false;
11 bool codigoValido = false;
12
13
14 // Constantes pines: botones, sensores y actuadores
15 const uint8_t BOTON = 9; //Accion
16 const uint8_t BOTON2 = 12; //Configuracion
17 const uint8_t SensorFCA = 6; //Final de Carrera Abierto
18 const uint8_t SensorFCC = 7; //Final de Carrera Cerrado
19 const uint8_t led0 = 10; //direccion
20 const uint8_t led1 = 11; //Activacion
21
22 //Variables estado: porton, botones, sensores y tiempo
23 uint8_t estado_porton = LOW;
24 bool direccion = HIGH; //"0" = Abriendo, "1" = Cerrando
25 bool activacion = LOW; //"0" = Apagado, "1" = Encendido
26 bool modos = LOW;
27
28 bool estado_boton = HIGH; //Manual
29 bool estado_boton2 = HIGH; //Tempo
30 bool estado_boton3 = HIGH; //MOD
31 bool anterior_estado_boton = HIGH;
32 bool anterior_estado_boton2 = HIGH;
33 bool anterior_estado_boton3 = HIGH;
34
35 bool estado_FCA = HIGH;
36 bool estado_FCC = HIGH;
37
38 bool condicion = HIGH;
39 bool condicion2 = LOW;
40
41 unsigned long anterior_tiempo = 0;
42 unsigned long anterior_tiempo2 = 0;
43 unsigned long anterior_tiempo3 = 0;
44 const uint16_t espera = 1000;
45 const uint16_t espera_temporizada = 10000;
46 const uint16_t INTERVALO = 1000;
47
48 //Creacion objetos
49 RCSwitch mySwitch = RCSwitch();
50
51 enum { // Constantes para los estados del porton
52     Frenado = 0,
53     Apertura,
54     Cerraje,
55 };
56
```

```

1 void abrir_porton() { // Funcion para abrir el porton
2   activacion = HIGH;
3 }
4 void cerrar_porton() { // Funcion para cerrar el porton
5   activacion = HIGH;
6 }
7 void parar_porton() { // Funcion para parar el porton
8   activacion = LOW;
9 }
10 void mensajes(String mensaje) { //Prender leds
11   digitalWrite(led0, direccion);
12   digitalWrite(led1, activacion);
13   Serial.print("Porton: ");
14   Serial.println(mensaje);
15 }
16 // Funcion verificacion estados
17 void verificacion_estado_porton() {
18
19   if (modos) {
20     if (estado_porton == Frenado && !estado_FCA && condicion) {
21       anterior_tiempo2 = millis();
22       condicion = LOW;
23       Serial.println("321");
24     }
25     if (estado_porton == Frenado && !estado_FCA && estado_FCC) {
26       if (millis() - anterior_tiempo2 >= espera_temporizada) {
27         anterior_tiempo2 = millis();
28         estado_porton = Cerraje;
29         direccion = HIGH;
30         condicion = HIGH;
31         Serial.println("123");
32       }
33     }
34   }
35
36   switch (estado_porton) {
37     case Apertura:
38       if (!estado_FCA) estado_porton = Frenado; //Si el porton se esta ABRRIENDO y llego al FCA FRENAR el porton
39       else estado_porton = Apertura; //Abrir el porton hasta nuevo aviso
40       break;
41
42     case Cerraje:
43       if (!estado_FCC) estado_porton = Frenado; //Si el porton se esta CERRANDO y llego al FCC FRENAR el porton
44       else estado_porton = Cerraje; //Cerrar el porton hasta nuevo aviso
45       break;
46
47     case Frenado:
48       if (!estado_FCC && estado_FCA && direccion == LOW) estado_porton = Apertura; //Si el porton esta QUIETO y llego al FCC y se presiono el boton ABRIR el porton
49       else if (estado_FCC && !estado_FCA && direccion == HIGH) estado_porton = Cerraje; //Si el porton esta QUIETO y llego al FCA y se presiono el boton CERRAR el porton
50       else if (estado_FCA && estado_FCC && direccion == LOW && condicion2) estado_porton = Apertura;
51       else if (estado_FCA && estado_FCC && direccion == HIGH && condicion2) estado_porton = Cerraje;
52       else estado_porton = Frenado; //Frenar Porton hasta nuevo aviso
53       break;
54
55     default:
56       parar_porton();
57       break;
58   }
59 }
60

```



```
1 void setup() {  
2   Serial.begin(9600);  
3   mySwitch.enableReceive(0); // Recepción por interrupción 0 => Eso es en el pin 2  
4   //pinMode(LED, OUTPUT);  
5   pinMode(BOTON, INPUT_PULLUP);  
6   pinMode(BOTON2, INPUT_PULLUP);  
7   pinMode(SensorFCA, INPUT_PULLUP);  
8   pinMode(SensorFCC, INPUT_PULLUP);  
9   pinMode(led0, OUTPUT);  
10  pinMode(led1, OUTPUT);  
11  Serial.begin(9600);  
12  estado_porton = Frenado;  
13 }  
14
```



```
1  if (millis() - anterior_tiempo3 > INTERVALO && codigoValido == true) {
2      codigoValido = false;
3      mySwitch.enableReceive(0);
4  }
5
6  estado_FCA = digitalRead(SensorFCA);
7  estado_FCC = digitalRead(SensorFCC);
8  verificacion_estado_porton();
9
10 if (millis() - anterior_tiempo >= espera) {
11     anterior_tiempo = millis();
12     switch (estado_porton) {
13         case Apertura: // Abre el porton
14             abrir_porton();
15             mensajes("Abriendo");
16             break;
17
18         case Cerraje: // Cierra el porton
19             cerrar_porton();
20             mensajes("Cerrando");
21             break;
22
23         case Frenado: // Frena el porton
24             parar_porton();
25             mensajes("Frenado");
26             break;
27
28         default:
29             Serial.println("cosas raras");
30             break;
31     }
32     Serial.print("Modo: ");
33     Serial.print(modos);
34     Serial.print("    |||    direccion: ");
35     Serial.print(direccion);
36     Serial.print("    |||    activacion: ");
37     Serial.println(activacion);
38     Serial.println("");
39 }
40 }
```

### Presupuesto

<u>Materiales</u>	<u>Cantidad</u>	<u>Precio C/U</u>	<u>Total</u>
-------------------	-----------------	-------------------	--------------



Arduino Uno	1	\$18.500	X
Placa Experimental	1	\$2.100	X
Cables	10	X	X
Fuente switching 12V	1	\$30.000	X
Resistencias	2	\$81,94	X
LED's	2	\$186	X
Switches	3	\$ 170	X
Control RF	1	\$9.499	X
Borneras	4	\$392	X
Gabinete	1	X	X
Modulo VHF 433mhz	1	\$2.232	X
Bateria 9V	1	\$3.919	X
X	X	X	\$68.863,88

## Vistas











