

**Materia: Electrónica Aplicada**

**Profesor: Gustavo Terradas**

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

**Año y División: 7°3**

**------------------------------------------------------------------------------------------**

**Proyecto Tecnológico: Estación de Soldado**

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

**Fecha: 29/6/2024**

**Indicé general**

**PORTADA......................................................................................................................1**

**INDICÉ GENERAL.......................................................................................................1**

**OBJETIVO......................................................................................................................1**

**FUNCIONAMIENTO SOFTWARE............................................................................3**

**FUNCIONAMIENTO....................................................................................................4**

**ETAPAS DEL PROYECTO..........................................................................................6**

**PRESUPUESTO.............................................................................................................7**

**GABINETE.....................................................................................................................8**

**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. Los objetivos principales de este proyecto son:
4. Diseñar y construir una estación de soldadura controlada por Arduino Uno y termocupla.
5. Implementar un sistema de control PID para regular la temperatura del soldador.
6. Desarrollar una interfaz de usuario para ajustar la temperatura deseada y monitorear la temperatura actual.
7. Probar y evaluar el funcionamiento de la estación de soldadura.

**Requisitos y requerimientos:**

**Funcionamiento**

La estación de soldadura funciona de la siguiente manera:

1. **Sensor de temperatura:** La termocupla se conecta al Arduino Uno y mide la temperatura del soldador.
2. **Control PID:** El Arduino Uno utiliza un algoritmo de control PID para ajustar la potencia del soldador y mantener la temperatura deseada.
3. **Interfaz de usuario:** La estación de soldadura cuenta con una pantalla LCD y botones para ajustar la temperatura deseada y monitorear la temperatura actual.
4. **Fuente de alimentación:** La estación de soldadura se alimenta con una fuente de alimentación de corriente continua que proporciona la potencia necesaria para el soldador.

**Funcionamiento del Software**

El software de la estación de soldadura controla la temperatura con precisión y gestiona la interacción del usuario.

**Funciones principales:**

* **Control de temperatura:**
  + Algoritmo PID ajusta la potencia del soldador para alcanzar y mantener la temperatura deseada.
  + Monitorea la temperatura actual con la termocupla.
* **Interfaz de usuario:**
  + Permite ajustar la temperatura deseada y visualizar información de interés.
  + Puede incluir opciones adicionales como guardar configuraciones o registrar datos.
* **Monitoreo y seguridad:**
  + Supervisa el sistema para detectar fallos y garantizar la seguridad.
  + Registra eventos para diagnóstico y resolución de problemas.

**En resumen, el software garantiza un funcionamiento preciso, seguro y confiable de la estación de soldadura.**

**Etapas del proyecto**

**Etapa 1: Recopilación de materiales**

En esta etapa, se recopilan todos los materiales necesarios para el proyecto, incluyendo:

* Arduino Uno
* Termocupla (Max6675)
* Pantalla LCD
* Botones
* Soldador
* Fuente de alimentación de corriente continua
* Cables
* Carcasa

**Etapa 2: Diseño del circuito**

En esta etapa, se diseña el circuito electrónico de la estación de soldadura. Se debe considerar la conexión de la termocupla al Arduino Uno, la conexión del soldador a la fuente de alimentación y la interfaz de usuario con la pantalla LCD y los botones.

**Etapa 3: Programación del Arduino Uno (Regulacion de Temperatura)**

En esta etapa, se programa el Arduino Uno para controlar la temperatura del soldador. Se debe implementar un algoritmo de control PID y desarrollar la interfaz de usuario para la pantalla LCD y los botones.

#include "LiquidCrystal.h"

#include "max6675.h"

// CONFIGURACION DE LOS PINES UTILIZADOS PARA LA COMUNICACIÓN CON EL MAX6675

#define CONFIG\_TCSCK\_PIN 10 // SPI SCK

#define CONFIG\_TCCS\_PIN 9 // SPI CS

#define CONFIG\_TCDO\_PIN 8 // SPI SO

// CONSTRUCTOR PARA LA PANTALLA LCD 16X2

// AQUI SE CONFIGURAN LOS PINES PARA LA COMUNICACION CON LA PANTALLA

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

// OBJETO UTILIZADO PARA LA COMUNICACION CON EL MAX6675

MAX6675 thermocouple(CONFIG\_TCSCK\_PIN, CONFIG\_TCCS\_PIN, CONFIG\_TCDO\_PIN);

void setup() {

// PREPARAR LA INTERFAZ SERIAL

Serial.begin(9600);

// IMPRIMR MENSAJE INICIAL A LA TERMINAL

Serial.println(F("----------------------------------------------------"));

Serial.println(F(" SENSOR DE TEMPERATURA MAX6675 CON ARDUINO "));

Serial.println(F(" ELECTROMAN YOUTUBE "));

Serial.println(F("----------------------------------------------------"));

// INDICAMOS QUE TENEMOS CONECTADA UNA PANTALLA DE 16X2

// IMPRIMIR MENSAJE INICIAL EN PANTALLA

lcd.begin(16, 2);

lcd.clear();

lcd.print(F("TERMOPAR ARDUINO"));

lcd.setCursor( 0, 1 );

lcd.print(F(" CON MAX6675 "));

// ESPERAR UN SEGUNDO

delay(1000);

}

void loop() {

// LEER EL TERMOPAR Y ALMACENAR EL VALOR EN UNA VARIABLE

double t = thermocouple.readCelsius();

// PRIMERO LIMPIAMOS LA PANTALLA Y LUEGO IMPRIMIMOS LA TEMPERATURA

lcd.clear();

lcd.print(F("->TEMPERATURA<-"));

lcd.setCursor( 3, 1 );

lcd.print(t);

// IMPRIMIR LA TEMPERATURA EN LA TERMINAL SERIAL

Serial.print("ªC = ");

Serial.println(t);

// ESPERAR UN SEGUNDO ENTRE LAS LECTURAS

delay(1000);

}

**Etapa 4: Desarrollo aplicación móvil (EN PROCESO)**

#Region Project Attributes

#ApplicationLabel: PrenderLED

#VersionCode: 1

#VersionName:

'SupportedOrientations possible values: unspecified, landscape or portrait.

#SupportedOrientations: portrait

#CanInstallToExternalStorage: False

#End Region

#Region Activity Attributes

#FullScreen: False

#IncludeTitle: False

#End Region

Sub Process\_Globals

'Creacion de variables.

Private xui As XUI

Dim ble As BleManager2

ble.Initialize("ble")

Dim connect As String

connect ( As String)

ble.readData (service As String)

ble.Scan (ServiceUUIDs As List)

ble.stopScan

SetNotify (Service As String, Characteristic As String, Notify As Boolean) As Boolean

WriteData (Service As String, Characteristic As String, Data() As Byte)

End Sub

Sub Globals

'These global variables will be redeclared each time the activity is created.

connexion

'ID de los puertos de transmision de daatos de un ESP32.

Dim UART\_UUid ="6e400001-b5a3-f393-e0a9-e50e24dcca9e" As String

Dim UART\_TX = "6e400003-b5a3-f393-e0a9-e50e24dcca9e" As String

Dim UART\_RX = "6E400002-B5A3-F393-E0A9-E50E24DCCA9E" As String

miembros visibles

'Declaracion de los objetos.

Private Botonconectado As Button

Private Botondesconectado As Button

Private LabelEstado As Label

Private ToggleButton1OnOf As ToggleButton

Private SeekBar1 As SeekBar

Private Label1 As Label

Private Label2 As Label

Dim tg As BitmapDrawable

'------------------------------------

' Mienbros de Datos

Private LblHumedad As Label

Private LblTemperatura As Label

Private Pnl\_1\_Datos As Panel

'------------------------------------

' Mienbros

Dim Control\_BLE As BleManager2

Dim PERMISO As RuntimePermissions

Private BC As ByteConverter

'Control\_BLE.Initialize("Control\_BLE")

'------------------------------------

'

Dim IDs(15) As String

Dim x=0 As Int

Dim xx As Boolean

Dim currentStateText As String

'Private CustomListView1 As CustomListView

End Sub

Sub Activity\_Create(FirstTime As Boolean)

Activity.LoadLayout("1")

Label1

Label2

ToggleButton1OnOf

Botonconectado

Botondesconectado

SeekBar1

LabelEstado

'visibles

tg.Initialize(LoadBitmap(File.DirAssets,"off2.png"))

tg.Gravity = Gravity.FILL

ToggleButton1OnOf.Background = tg

End Sub

Sub Activity\_Resume

End Sub

Sub Activity\_Pause (UserClosed As Boolean)

End Sub

Sub Button1\_Click

xui.MsgboxAsync("Hello world!", "B4X")

End Sub

Private Sub Botonconectado\_Click

End Sub

Private Sub Botondesconectado\_Click

Control\_BLE.Disconnect

BtnDesconexion.Enabled=False

TogBtnConexion.Enabled=True

ToastMessageShow("Desonectado", True)

End Sub

Private Sub ToggleButton1OnOf\_CheckedChange(Checked As Boolean)

'Visuales de los objetos

If Checked = True Then

tg.Initialize(LoadBitmap(File.DirAssets,"on.png"))

Else

tg.Initialize(LoadBitmap(File.DirAssets,"off2.png"))

End If

tg.Gravity = Gravity.FILL

ToggleButton1OnOf.Background= tg

End Sub

Private Sub SeekBar1\_ValueChanged (Value As Int, UserChanged As Boolean)

End Sub

**Etapa 5: Construcción de la estación de soldadura**

En esta etapa, se construye la estación de soldadura. Se debe montar el circuito electrónico en una carcasa, conectar todos los componentes y soldar los cables.

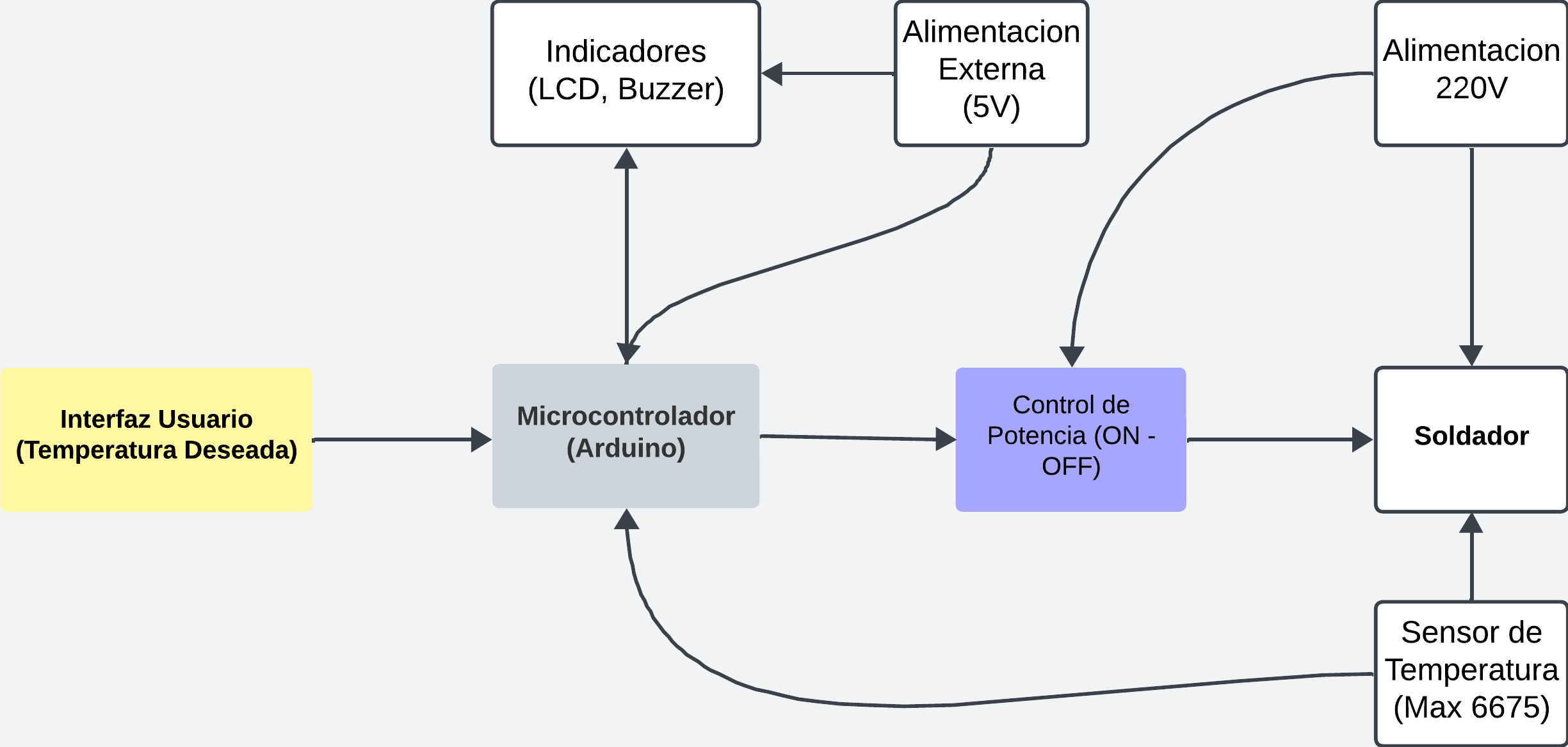
**Etapa 6: Pruebas y evaluación**

En esta etapa, se prueba el funcionamiento de la estación de soldadura. Se debe verificar que la temperatura del soldador se regule correctamente y que la interfaz de usuario funcione correctamente.

(**Se tendrá que realizar un test, e ir anotando los datos de dicho test en tiempo real**)

**Etapa 7: Documentación del proyecto**

En esta etapa, se documenta el proyecto, incluyendo el diseño del circuito, el código fuente del Arduino Uno y las instrucciones de construcción.



**Presupuesto**

|  |  |  |
| --- | --- | --- |
| **Materiales** | **Cantidad** | **Precio(c/u)** |
| Arduino Uno | 1 | $9500 |
| Termocupla | 1 | $8.500 |
| Soldador | 1 | $10.138 |
| Cables | 10 | $10.000 |
| Resistencias | 2 | $ 81,94 |
| LED’s | 2 | $186 |
| Switches | 3 | $2100 |
| Modulo Control de Potencia | 1 | $3000 |
| Borneras | 2 | $520 |
| Pantalla LED | 1 | $11,239.00 |
|  |  |  |