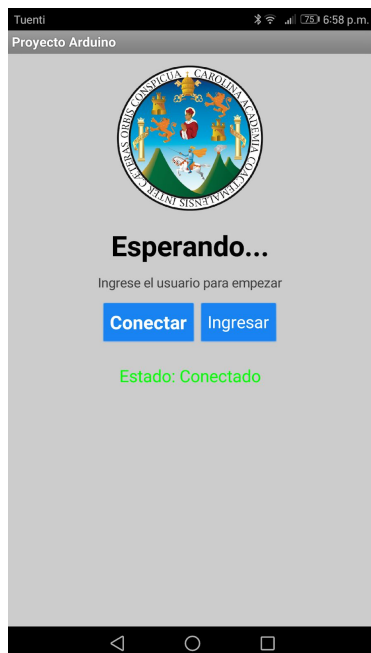


Documentación Proyecto Arduino

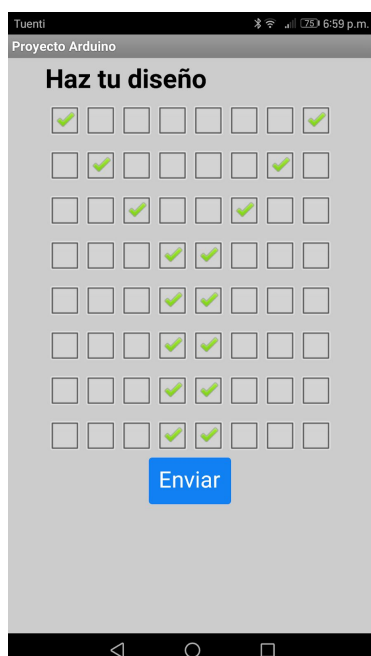
Aplicación Móvil

- Pantalla de inicio:



En esta pantalla podremos conectar la aplicación móvil con la simulación de proteus, para eso se debe pulsar el botón “conectar” se desplegará una lista con los dispositivos bluetooth disponibles se deberá conectar con el dispositivo que esté corriendo la simulación de proteus, si la conexión es exitosa el estado cambia a “conectado” y solo se deberá de esperar a que el usuario ingrese la contraseña para seguir.

- Pantalla de diseño



En esta pantalla se deberá ingresar el diseño deseado para que la simulación empiece el proceso de manufactura, en la pantalla se despliega una matriz de checkbox de 8*8 cada checkbox representa un led, por lo tanto si es checkbox está seleccionado corresponde a el led en esa posición encendido.

Una vez el diseño deseado esté plasmado en la matriz se deberá pulsar el botón enviar para seguir el proceso de control.

- pantalla de control



En esta pantalla estará el control del proceso de la simulación de deberá pulsar una sola vez cada estación. al pulsar la estación deseada la simulación empezará a trabajar en la estación que se pulso en la aplicación móvil, en cualquier momento se podrá iniciar el movimiento de la cinta transportadora y también se podrá detener el movimiento pulsando el mismo botón.

Se desplegará la temperatura de la simulación y el tiempo transcurrido, una vez todas las estaciones sean accionadas el botón aceptar será habilitado para concluir el proceso y volver a iniciar de nuevo en la pantalla de diseño. Por último el botón de emergencia("33-12") detiene todo el proceso y vuelve a iniciar en la pantalla de diseño.

Diagrama de proteus

Módulo de contraseña: El módulo de contraseña consta de un pad numérico, donde al ingresar la contraseña correcta (123456) ingresa al sistema dando una bandera verde en la aplicación para seleccionar un diseño.

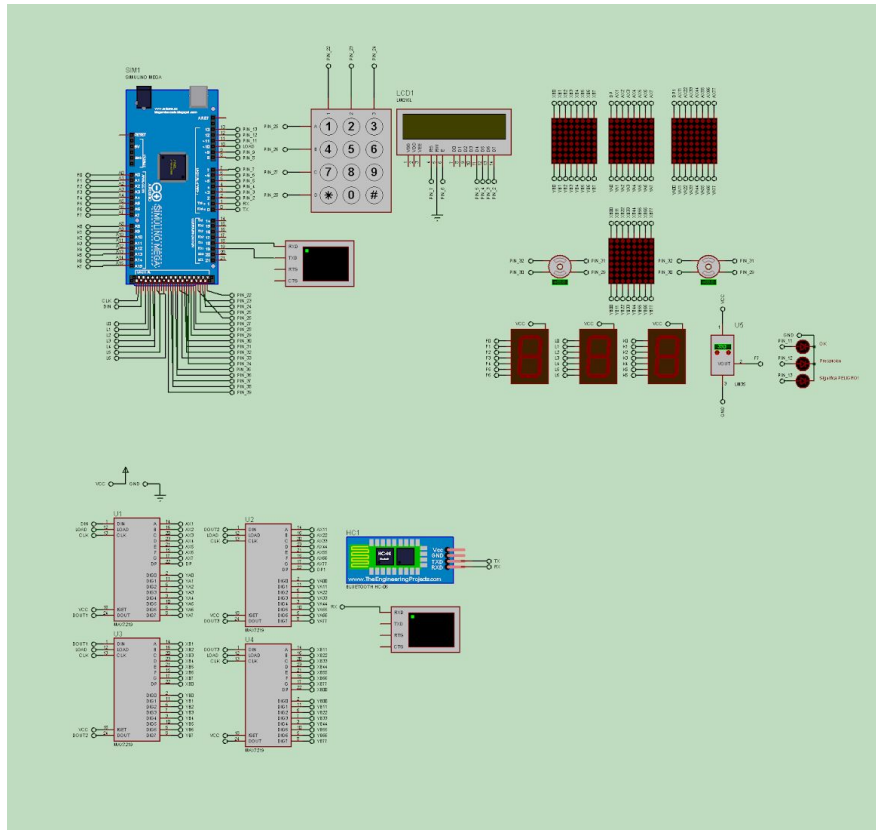
Módulo LCD: Muestra las distintas acciones de un usuario, al ingresar un usuario, al mover la banda o mostrar un error al ingresar al sistema.

Modulo de display de matrices: Muestra el diseño del producto en porcentaje según el display, para el primero un 33%, el segundo 66% y el tercero un 100%, el cuarto display muestra que display está mostrando y al finalizar los 3 muestra el producto terminado.

Modulo Banda trasportadora: Se puede mover y parar según la señal de la aplicación.

Modulo reloj: Se activa al seleccionar un diseño en la aplicación y envía a la aplicación en tiempo real su valor.

Modulo temperatura: La temperatura se puede variar en tiempo real y mostrar en la aplicación tiene también 3 alertas con verde funcionamiento correcto, precaución y error fatal donde se detiene todas las funciones que se están realizando actualmente.



Código de Arduino

```
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

//Variables de temperatura
int senLM35 = A7;
float temp;
float vt;
int rojo = 13;
int amarillo = 12;
int verde = 11;
float tempAnterior = 0;
float temperatura;
bool flag_emergencia = false;
```

```

//Variables para el reloj
byte digitOne[10] = {B01000000, B01111001, B00100100, B00110000,
B00011001, B00010010, B00000010, B01111000, B00000000, B00010000};
//digitos del 0 al 9
byte contador3 = 0; //Centenas
byte contador = 0; //decenas
byte contador2 = 0; //unidades
unsigned long tiempo1 = 0;
unsigned long tiempo2 = 0;
unsigned long tiempo3 = 0;
String value;

//crear un objeto LiquidCrystal (rs,enable,d4,d5,d6,d7)
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

//Variables teclado
const int ROW_NUM = 4; //four rows
const int COLUMN_NUM = 3; //three columns

char keys[ROW_NUM][COLUMN_NUM] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'}
};

//Variable temporal que almacena las cadenas recibidas
String cadena = "";

//variable que guarda la estacion
int estacion = 0;

//Variable que guarda el tope
int topar = 3;

//Matriz que almacena la matriz del diseño
int matriz[8][8];

//configuracion panel
int pinCS = 10; // Attach CS to this pin, DIN to MOSI and CLK to
SCK (cf http://arduino.cc/en/Reference/SPI )

```

```

int numeroMatricesHorizontales = 4;
int numeroMatricesVerticales = 1;

Max72xxPanel matrizLED = Max72xxPanel(pinCS,
numeroMatricesHorizontales, numeroMatricesVerticales);

byte pin_rows[ROW_NUM] = {25, 26, 27, 28}; //connect to the row
pinouts of the keypad
byte pin_column[COLUMN_NUM] = {22, 23, 24}; //connect to the
column pinouts of the keypad

const String password = "1"; // change your password here
String input_password;
String autorizacion = "";

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column,
ROW_NUM, COLUMN_NUM );
int estado = 1;
String estacion_actual;
bool flag_parcearCadena = false;
int contador_estaciones = 0;
char caracter;
char cinta;
char caracter_emergencia;
bool estado_emergencia = false;

String texto_fila = "ACYE 1 G2";

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);

  //Variables del reloj
  DDRK = B11111111; //declaramos como salida
  DDRL = B11111111;
  DDRF = B01111111;
  tiempo1 = millis();
  PORTK = digitOne[0];
  PORTL = digitOne[0];
  PORTF = digitOne[0];

```

```
//Variables de temperatura
pinMode(verde, OUTPUT);
pinMode(amarillo, OUTPUT);
pinMode(rojo, OUTPUT);
pinMode(senLM35, INPUT);

//Motores Stepper
pinMode(29, OUTPUT);
pinMode(30, OUTPUT);
pinMode(31, OUTPUT);
pinMode(32, OUTPUT);

//pantalla
lcd.begin(16, 2); //16 columnas y 2 filas
lcd.setCursor(1, 0);
lcd.print("  INGRESE SU  ");
lcd.setCursor(1, 1);
lcd.print(" USUARIO:  ");
input_password.reserve(32);

//Emulacion cadena recibida
cadena += "00100100"; //Fila 1
cadena += "00100100"; //Fila 2
cadena += "00100100"; //Fila 3
cadena += "00100100"; //Fila 4
cadena += "00000000"; //Fila 5
cadena += "11100111"; //Fila 6
cadena += "00100100"; //Fila 7
cadena += "00011000"; //Fila 8

//Configuracion de los paneles
matrizLED.setIntensity(7);
matrizLED.setPosition(1, 0, 0);
matrizLED.setPosition(0, 1, 0);
matrizLED.setRotation(0, 3);
matrizLED.setRotation(1, 3);
matrizLED.setRotation(2, 3);
matrizLED.setRotation(3, 3);

//Escribe en los paneles y setea todos apagados
```

```

matrizLED.fillScreen(LOW);
matrizLED.write();

}

void Mensaje_Mov() {
    int tam_texto = texto_fila.length();

    for (int i = tam_texto; i > 0 ; i--)
    {
        String texto = texto_fila.substring(i - 1);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(texto);
        delay(75);
    }

    for (int i = 1; i <= 16; i++)
    {
        lcd.clear();
        lcd.setCursor(i, 0);
        lcd.print(texto_fila);
        delay(75);
    }
}

void Estacion() {
    if (Serial.available()) {
        character = Serial.read();
        if (character == '1') {
            estacion_actual = "1";
            lcd.clear();
            lcd.setCursor(1, 0);
            lcd.print(" Estacion 1 ");
            lcd.setCursor(1, 1);
            lcd.print("-----");
        } else if (character == '2') {
            estacion_actual = "2";
            lcd.clear();
            lcd.setCursor(1, 0);
            lcd.print(" Estacion 2 ");
            lcd.setCursor(1, 1);

```

```

        lcd.print("-----");
    } else if (caracter == '3') {
        lcd.clear();
        lcd.setCursor(1, 0);
        lcd.print("  Estacion 3  ");
        lcd.setCursor(1, 1);
        lcd.print("-----");
        estacion_actual = "3";
    } else if (caracter == '4') {
        estado_emergencia = true;
        Serial1.println("EMERGENCIA!");
    } else if (caracter == '5') {
        cinta = '5';
        Serial1.println("StepperOn: " + (String) cinta);
    } else if (caracter == '6') {
        cinta = '6';
        Serial1.println("StepperOff: " + (String) cinta);
    }
    } else {
        estacion_actual = "error";
    }
    reloj();
}

String Recibiendo_Cadena() {
    String text;
    while (Serial.available()) {
        delay(5);
        char c = Serial.read();
        text += c;
    }
    if (text.length() > 0) {
        flag_parcearCadena = true;
        return text;
    }
}

void loop() {
    switch (estado) {
        case 1:
            //Mensaje_Mov();
            USUARIO();

```



```

    delay(50);
    break;
case 2:
    cadena = Recibiendo_Cadena();
    delay(50);
    if (flag_parcearCadena == true) {
        parcearCadena(cadena);
    }
    PORTK = PORTL = PORTF = digitOne[0];
    contador = contador2 = contador3 = 0;
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("  ACYE 1 G2  ");
    lcd.setCursor(1, 1);
    lcd.print("-----");
    break;
case 3:
    stepper();
    Estacion();
    if (estado_emergencia == true) {
        estado_emergencia = false;
        cadena = "";
        estacion = 0;
        topar = 3;
        estacion_actual = "";
        estado = 2;
        contador_estaciones = 0;
        matrizLED.fillScreen(LOW);
        matrizLED.write();
        Serial1.println("Limpia Valores y Resetea");
        break;
    }
    if (estacion_actual != "error") {
        contador_estaciones++;
        serial_Estacion(estacion_actual);
        estacion_actual == "";
        Serial1.println(contador_estaciones);
        if (contador_estaciones == 3) {
            delay(50);
            serial_Estacion("4");
            estacion_actual == "";
            contador_estaciones = 0;
        }
    }

```

```

        estado = 2;
        delay(1000);
        matrizLED.fillScreen(LOW);
        matrizLED.write();
    }
}
reloj();
Temperatura();
if (flag_emergencia) {
    Serial.println("PELIGRO");
} else {
    Serial.println("Temperatura: " + (String) temperatura +
"C" + " " + value);
}
break;
case 4:
    break;
}
}

void USUARIO() {
    lcd.setCursor(1, 0);
    lcd.print("  INGRESE SU  ");
    lcd.setCursor(1, 1);
    lcd.print("  USUARIO:  ");

    Verificar_Usuario();
    if (estado == 2) {
        for (int i = 0; i < 25; i++) {
            Serial.println(autorizacion);
            delay(50);
        }
    }
}

void Verificar_Usuario() {
    char key = keypad.getKey();

    if (key) {
        Serial1.print(key);
        if (key == '#') {
            input_password = ""; // clear input password

```

```

    } else if (key == '*') {
        if (password == input_password) {
            autorizacion = "Autorizado";
            estado = 2;
            lcd.clear();
            lcd.setCursor(1, 0);
            lcd.print("  Bienvenido  ");
            lcd.setCursor(1, 1);
            lcd.print("                ");
            delay(800);
            lcd.clear();
        } else {
            lcd.clear();
            lcd.setCursor(1, 0);
            lcd.print("                ");
            lcd.setCursor(1, 1);
            lcd.print("  ERROR  ");
            autorizacion = "Denegado";
            delay(800);
        }
        input_password = ""; // clear input password
    } else {
        input_password += key; // append new character to input
        password string
    }
}

//Convierte una cadena string recibida del diseño a matriz
void parpearCadena(String cadenas) {
    int contador = 0;
    for (int a = 0; a < 8; a++) {
        for (int b = 0; b < 8; b++) {
            matriz[a][b] = cadenas[contador];
            contador++;
        }
    }
    flag_parpearCadena = false;
    Serial1.println("Conversion con exito.");
    estado = 3;
}

```

```

//Convierte la cadena recibida de la estacion actual a int
void parsearEstacion(String cadena) {
    if (cadena[0] == 49) {
        estacion = 1;
    } else if (cadena[0] == 50) {
        estacion = 2;
    } else if (cadena[0] == 51) {
        estacion = 3;
    } else if (cadena[0] == 52) {
        estacion = 4;
    }
    if (estado_emergencia == true) {
        estado_emergencia = false;
        cadena = "";
        estacion = 0;
        topar = 3;
        estacion_actual = "";
        estado = 2;
        contador_estaciones = 0;
        matrizLED.fillScreen(LOW);
        matrizLED.write();
        Serial1.println("Limpia Valores y Resetea");
    }
}

//Imprime en el primer panel como parametro recibe hasta donde
// grafica
void imprimirPanel1(int tope) {
    //Tope = 8 = 100%
    //Tope = 6 = 66%
    //Tope = 3 = 33%

    for (int b = 0; b < 8; b++) {
        for (int a = 0; a < tope; a++) {
            //LOW
            if (matriz[a][b] == 48) {
                matrizLED.drawPixel(b, a, LOW);
                matrizLED.write();
            }
            //HIGH
            else {
                matrizLED.drawPixel(b, a, HIGH);
            }
        }
    }
}

```

```

        matrizLED.write();
    }
}

}

}

//Imprime en el segundo panel como parametro recibe hasta donde
grafica
void imprimirPanel2(int tope) {
    //Tope = 8 = 100%
    //Tope = 6 = 66%
    //Tope = 3 = 33%

    for (int x = 0; x < 8; x++) {
        for (int y = 0; y < tope; y++) {
            //LOW
            if (matriz[y][x] == 48) {
                matrizLED.drawPixel(x + 8, y, LOW);
                matrizLED.write();
            }
            //HIGH
            else {
                matrizLED.drawPixel(x + 8, y, HIGH);
                matrizLED.write();
            }
        }
    }
}

//Imprime en el tercer panel como parametro recibe hasta donde
grafica
void imprimirPanel3(int tope) {
    //Tope = 8 = 100%
    //Tope = 6 = 66%
    //Tope = 3 = 33%

    for (int x = 0; x < 8; x++) {
        for (int y = 0; y < tope; y++) {

```

```

        //LOW
        if (matriz[y][x] == 48) {
            matrizLED.drawPixel(x + 16, y, LOW);
            matrizLED.write();
        }
        //HIGH
        else {
            matrizLED.drawPixel(x + 16, y, HIGH);
            matrizLED.write();
        }
    }
}

//Va actualizando el valor de topar para la proxima vez que se
necesite
void verificarTopar() {
    if (topar == 6) {
        topar = topar + 2;
    } else if (topar == 8) {
        topar = 3;
    }
    else {
        topar = topar + 3;
    }
}

void limpiarPanel(int numeroPanel) {

    switch (numeroPanel) {
        case 1://Limpiar panel 1
            for (int b = 0; b < 8; b++) {
                for (int a = 0; a < 8; a++) {
                    //LOW
                    matrizLED.drawPixel(b, a, LOW);
                    matrizLED.write();

                }
            }
    }
}

```

```

        break;
    case 2:
        for (int x = 0; x < 8; x++) {
            for (int y = 0; y < 8; y++) {
                //LOW
                matrizLED.drawPixel(x + 8, y, LOW);
                matrizLED.write();
            }
        }
        break;
    case 3:
        for (int x = 0; x < 8; x++) {
            for (int y = 0; y < 8; y++) {
                //LOW
                matrizLED.drawPixel(x + 16, y, LOW);
                matrizLED.write();
            }
        }
        break;
    }
}

//Panel 4 controla a los otros 3 segun el numero que se ingrese
muestra el numero
//En su panel y el panel correspondiente.
void imprimirPanel4(int numero) {
    int x = 26;
    int y = 0;
    int veces = 0;
    Estacion();
    if (estado_emergencia == true) {
        estado_emergencia = false;
        cadena = "";
        estacion = 0;
        topar = 0;
        estacion_actual = "";
    }
}

```

```

    estado = 2;
    contador_estaciones = 0;
    matrizLED.fillScreen(LOW);
    matrizLED.write();
    Serial1.println("Limpia Valores y Resetea");
}
if (flag_emergencia) {
    Serial.println("PELIGRO");
} else {
    Serial.println("Temperatura: " + (String) temperatura + "C" +
"      " + value);
}
switch (numero) {
    case 1: matrizLED.drawChar(x, y, '1', HIGH, LOW, 1);
        matrizLED.write();
        //Ejecutamos la estacion que recibio
        //Aqui hacemos que parpadea

        veces = 0;
        //El numero de veces son la veces que parpadea (veces * 200)
= Milisegundos
        while (veces < 8) {
            imprimirPanel1(topar);
            delay(100);

            //Apagamos todos los led
            limpiarPanel(1);
            delay(100);
            veces++;
            Estacion();
            if (estado_emergencia == true) {
                estado_emergencia = false;
                cadena = "";
                estacion = 0;
                topar = 0;
                estacion_actual = "";
                estado = 2;
                contador_estaciones = 0;
                matrizLED.fillScreen(LOW);
                matrizLED.write();
                Serial1.println("Limpia Valores y Resetea");
                break;
            }
        }
    }
}

```



```

    }
    if (flag_emergencia) {
        Serial.println("PELIGRO");
    } else {
        Serial.println("Temperatura: " + (String) temperatura +
"C" + "      " + value);
    }
}
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("  ACYE 1 G2  ");
lcd.setCursor(1, 1);
lcd.print("-----");
verificarTopar();

//verificarTopar();
break;
case 2: matrizLED.drawChar(x, y, '2', HIGH, LOW, 1);
matrizLED.write();
//Ejecutamos la estacion que recibio
//Aqui hacemos que parpadea

veces = 0;
//El numero de veces son la veces que parpadea (veces * 200)
= Milisegundos
while (veces < 8) {
    imprimirPanel2(topar);
    delay(100);

    //Apagamos todos los led
    limpiarPanel(2);
    delay(100);
    veces++;
    Estacion();
    if (estado_emergencia == true) {
        estado_emergencia = false;
        cadena = "";
        estacion = 0;
        topar = 0;
        estacion_actual = "";
        estado = 2;
    }
}

```

```

        contador_estaciones = 0;
        matrizLED.fillScreen(LOW);
        matrizLED.write();
        Serial1.println("Limpia Valores y Resetea");
        break;
    }
    if (flag_emergencia) {
        Serial.println("PELIGRO");
    } else {
        Serial.println("Temperatura: " + (String) temperatura +
"C" + "      " + value);
    }
}
}
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("  ACYE 1 G2  ");
lcd.setCursor(1, 1);
lcd.print("-----");
verificarTopar();
//verificarTopar();
break;
case 3: matrizLED.drawChar(x, y, '3', HIGH, LOW, 1);
        matrizLED.write();
        //Ejecutamos la estacion que recibio
        //Aqui hacemos que parpadea

        veces = 0;
        //El numero de veces son la veces que parpadea (veces * 200)
= Milisegundos
        while (veces < 8) {
            imprimirPanel3(topar);
            delay(100);

            //Apagamos todos los led
            limpiarPanel(3);
            delay(100);
            veces++;
            Estacion();
            if (estado_emergencia == true) {
                estado_emergencia = false;
                cadena = "";
                estacion = 0;

```

```

        topar = 0;
        estacion_actual = "";
        estado = 2;
        contador_estaciones = 0;
        matrizLED.fillScreen(LOW);
        matrizLED.write();
        Serial1.println("Limpia Valores y Resetea");
        break;
    }
    if (flag_emergencia) {
        Serial.println("PELIGRO");
    } else {
        Serial.println("Temperatura: " + (String) temperatura +
"C" + "      " + value);
    }
}
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("  ACYE 1 G2  ");
lcd.setCursor(1, 1);
lcd.print("-----");
verificarTopar();
//verificarTopar();
break;
case 4:
    matrizLED.fillScreen(LOW);
    matrizLED.write();
    veces = 0;
    for (int x = 0; x < 8; x++) {
        for (int y = 0; y < 8; y++) {
            //LOW
            if (matriz[y][x] == 48) {
                matrizLED.drawPixel(x + 24, y, LOW);
                matrizLED.write();
            }
            //HIGH
            else {
                matrizLED.drawPixel(x + 24, y, HIGH);
                matrizLED.write();
            }
        }
        Estacion();
        if (estado_emergencia == true) {

```

```

        estado_emergencia = false;
        cadena = "";
        estacion = 0;
        topar = 0;
        estacion_actual = "";
        estado = 2;
        contador_estaciones = 0;
        matrizLED.fillScreen(LOW);
        matrizLED.write();
        Serial1.println("Limpia Valores y Resetea");
        break;
    }
}
Estacion();
if (estado_emergencia == true) {
    estado_emergencia = false;
    cadena = "";
    estacion = 0;
    topar = 0;
    estacion_actual = "";
    estado = 2;
    contador_estaciones = 0;
    matrizLED.fillScreen(LOW);
    matrizLED.write();
    Serial1.println("Limpia Valores y Resetea");
    break;
}
}
break;
}
}

void serial_Estacion(String recibir) {
    //Parseamos la estacion que recibimos en el serial
    parsearEstacion(recibir);
    //Llamamos al panel cuatro que muestra el panel correcto
    imprimirPanel4(estacion);
}

void stepper() {

    if (cinta == '5') {

```

```

    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("CINTA TRANSPORTA");
    lcd.setCursor(1, 1);
    lcd.print("  MOVIENDOSE  ");
    digitalWrite(29, HIGH);
    digitalWrite(30, LOW);
    digitalWrite(31, LOW);
    digitalWrite(32, LOW);
    delay(180);
    digitalWrite(29, LOW);
    digitalWrite(30, HIGH);
    digitalWrite(31, LOW);
    digitalWrite(32, LOW);
    delay(180);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31, HIGH);
    digitalWrite(32, LOW);
    delay(180);
    digitalWrite(29, LOW);
    digitalWrite(30, LOW);
    digitalWrite(31, LOW);
    digitalWrite(32, HIGH);
    delay(180);
  }
}

void reloj() {
  tiempo2 = millis(); //tiempo trascurrido
  if (tiempo2 > (tiempo1 + 350)) { // indica que ha pasado 1
segundo de tiempo en arduino
    value = "Tiempo: " + (String)contador3 + ":" +
(String)contador + "" + (String) contador2; //centenas, decenas ,
unidades
    //Serial.println(value); //lo mandamos por puerto serial
    tiempo1 = millis(); //actualizamos el tiempo 1
    PORTK = digitOne[contador2]; //escribimos las unidades
    contador2 = contador2 + 1;

    if (contador2 == 10 ) {
      contador2 = 0; //volvemos a 0 las unidades
    }
  }
}

```

```

        contador = contador + 1; //aumentamos decenas
    }
}
if (contador2 == 1) {
    PORTL = digitOne[contador];
    if (contador == 6) { //si ya llegó a 6 las decenas, ya pasó 1
minuto, y hay que volver a iniciar
        contador = 0;
        contador3++;
    }
    PORTF = digitOne[contador3];
}
}

void Temperatura() {
    vt = analogRead(senLM35);
    //El sensor nos devuelve de 0 a 1.5V
    float mv = (vt / 1024.0) * 5000; //miliVoltios
    temperatura = mv / 10; //temperatura en grados celsius

    if (temperatura < 37) {
        digitalWrite(verde, HIGH); //Enciende el verde
        digitalWrite(amarillo, LOW); //Se apaga el amarillo
        digitalWrite(rojo, LOW); //Se apaga el rojo
        lcd.setCursor(1, 0);
        lcd.print("  ACYE 1 G2  ");
        lcd.setCursor(1, 1);
        lcd.print("-----");
        flag_emergencia = false;
    } else if (temperatura > 36 && temperatura < 46) {
        digitalWrite(verde, LOW); //Se apaga el verde
        digitalWrite(amarillo, HIGH); //Enciende el amarillo
        digitalWrite(rojo, LOW); //Se apaga el rojo
        lcd.setCursor(1, 0);
        lcd.print("  WARNING  ");
        lcd.setCursor(1, 1);
        lcd.print("-----");
        flag_emergencia = false;
    } else if (temperatura > 45) {
        digitalWrite(verde, LOW); //Se apaga el verde
        digitalWrite(amarillo, LOW); //Se apaga el amarillo

```

```
digitalWrite(rojo, HIGH); //Enciende el rojo
flag_emergencia = true;
lcd.setCursor(1, 0);
lcd.print("  DANGER  ");
lcd.setCursor(1, 1);
lcd.print("-----");
Serial.println("PELIGRO");
}
}
```