UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

ARQUITECTURA DE COMPUTADORES Y ENSAMBLADORES 1

AUX: ROBINSÓN HERNÁNDEZ

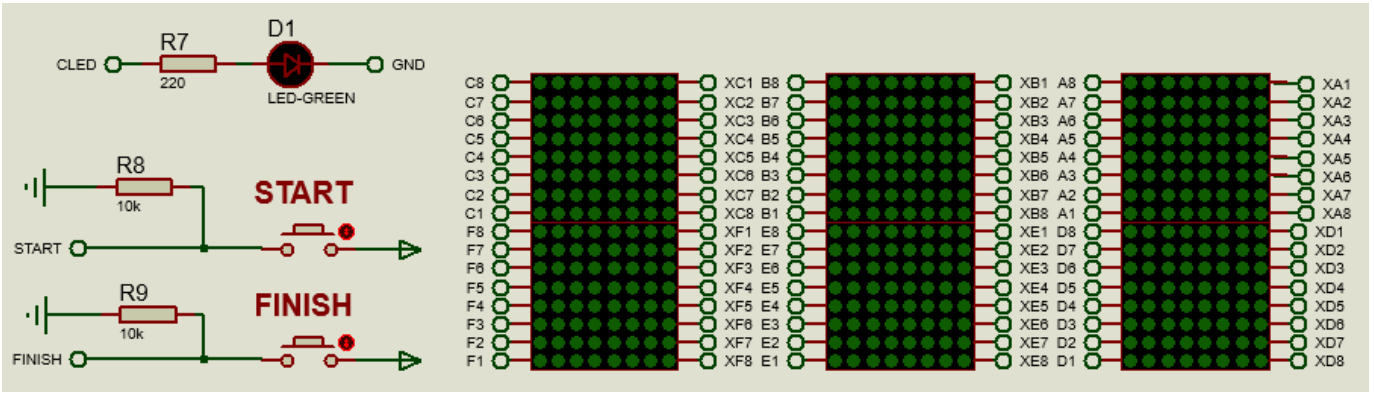
VACACIONES DE DICIEMBRE 2022

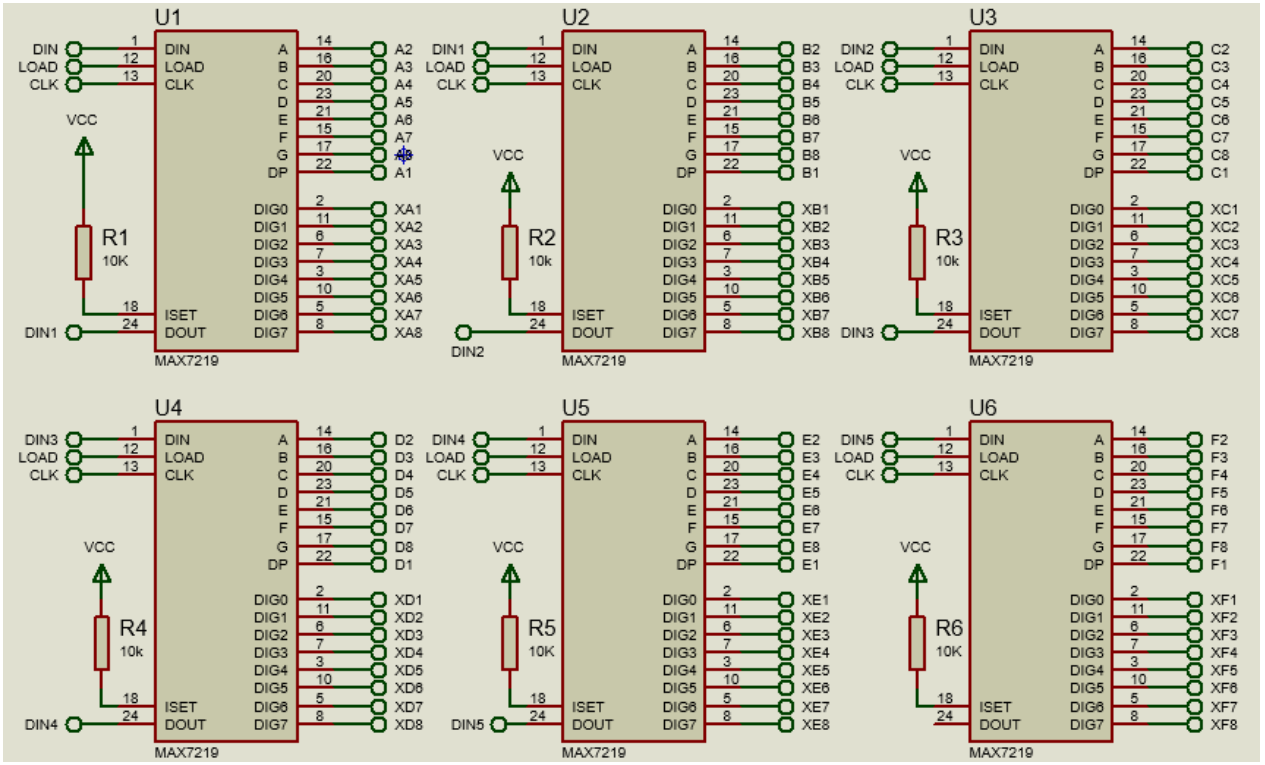
GRUPO 2

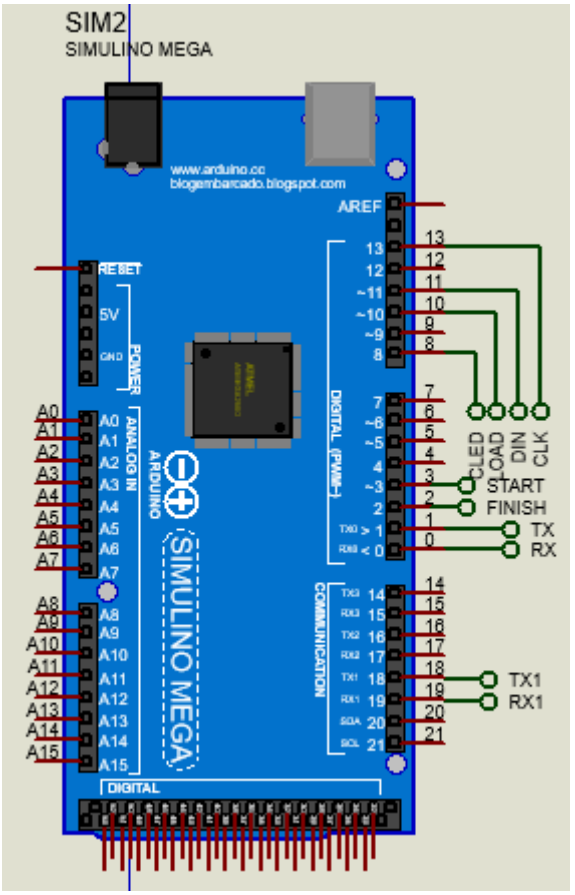
PRÁCTICA 1

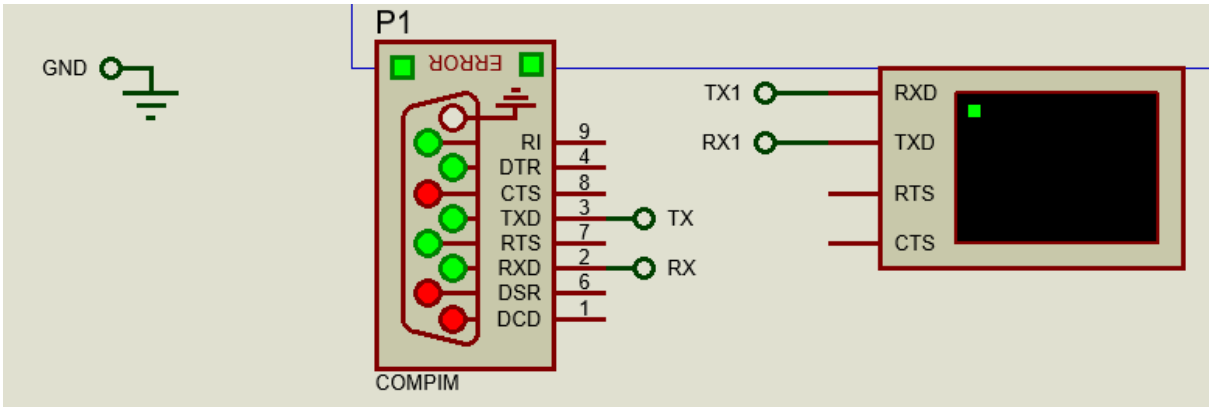
NOMBRE	CARNET
Rodrigo Antonio Porón De León	201902781
Marcos Geovani Josías Pérez Secay	201903878
Mario Cesar Moran Porras	202010793
Cristian Daniel Pereira Tezagüic	202010893
María Zucely Hernández García	201701160
Eliezer Abraham Zapeta Alvarado	201801719
Marvin Alexis Estrada Florian	201800476
Roxana Madahí Carías Paredes	201800672

DIAGRAMA


R7
D1
CLED
220
LED-GREEN
GND
R8
10k
START
START
R9
10k
FINISH
FINISH
C8 C7 C6 C5 C4 C3 C2 C1 F8 F7 F6 F5 F4 F3 F2 F1
XC1 B8 XC2 B7 XC3 B6 XC4 B5 XC5 B4 XC6 B3 XC7 B2 XC8 B1 XF1 E8 XF2 E7 XF3 E6 XF4 E5 XF5 E4 XF6 E3 XF7 E2 XF8 E1
XB1 A8 XB2 A7 XB3 A6 XB4 A5 XB5 A4 XB6 A3 XB7 A2 XB8 A1 XE1 D8 XE2 D7 XE3 D6 XE4 D5 XE5 D4 XE6 D3 XE7 D2 XE8 D1
XA1 XA2 XA3 XA4 XA5 XA6 XA7 XA8 XD1 XD2 XD3 XD4 XD5 XD6 XD7 XD8


U1
U2
U3
U4
U5
U6
DIN
LOAD
CLK
VCC
R1
10K
R2
10k
R3
10k
R4
10k
R5
10K
R6
10K
ISET
DOUT
MAX7219
DIN1
DIN2
DIN3
DIN4
DIN5
A B C D E F G DP
DIG0 DIG1 DIG2 DIG3 DIG4 DIG5 DIG6 DIG7
A2 A3 A4 A5 A6 A7 A8 A1
XA1 XA2 XA3 XA4 XA5 XA6 XA7 XA8
B2 B3 B4 B5 B6 B7 B8 B1
XB1 XB2 XB3 XB4 XB5 XB6 XB7 XB8
C2 C3 C4 C5 C6 C7 C8 C1
XC1 XC2 XC3 XC4 XC5 XC6 XC7 XC8
D2 D3 D4 D5 D6 D7 D8 D1
XD1 XD2 XD3 XD4 XD5 XD6 XD7 XD8
E2 E3 E4 E5 E6 E7 E8 E1
XE1 XE2 XE3 XE4 XE5 XE6 XE7 XE8
F2 F3 F4 F5 F6 F7 F8 F1
XF1 XF2 XF3 XF4 XF5 XF6 XF7 XF8


SIM2
SIMULINO MEGA
www.arduino.cc
blogembarcado.blogspot.com
AREF
RESET
5V
GND
POWER
ANALOG IN
DIGITAL (PWM~)
COMMUNICATION
DIGITAL
ARDUINO
SIMULINO MEGA
CLK
DIN
LOAD
CLED
START
FINISH
TX
RX
TX1
RX1


GND
P1
ERROR
RI
DTR
CTS
TXD
RTS
RXD
DSR
DCD
TX
RX
COMPIM
TX1
RX1
RXD
TXD
RTS
CTS

LIBRERÍAS DE JAVA UTILIZADAS

Código	Descripción
7 import com.panamahitek.ArduinoException; 8 import com.panamahitek.PanamaHitek_Arduino; 9 import java.util.logging.Level; 10 import java.util.logging.Logger; 11 import jssc.SerialPortException;	**Línea 7-8:** Librerías utilizadas para facilitar la comunicación entre Arduino y Java. **Línea 11:** Importando de la biblioteca de comunicación serial JSSC (para uso del puerto) especialmente las Excepciones.

CÓDIGO JAVA

Código: Interfaz.java

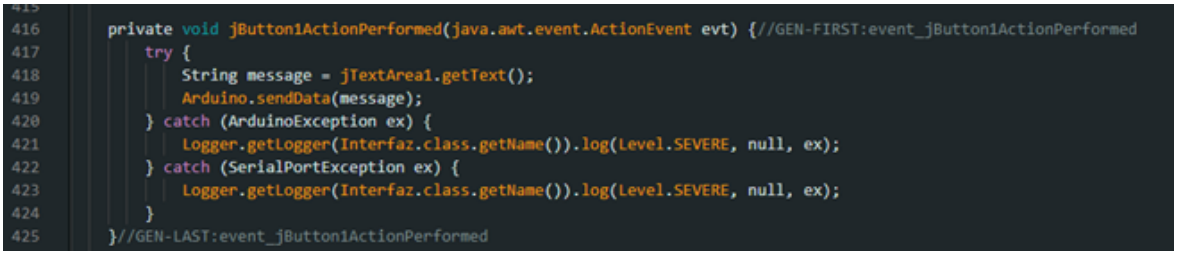
```
19      public static PanamaHitek_Arduino Arduino = new PanamaHitek_Arduino();
```

```
23      public Interfaz() {
24          initComponents();
25
26          try {
27              Arduino.arduinoTX("COM2", 9600);
28          } catch (ArduinoException ex) {
29              System.err.println("No hay arduino para comunicar");
30              Logger.getLogger(Interfaz.class.getName()).log(Level.SEVERE, null, ex);
31          }
32      }
```

Descripción

Línea 19: El objeto “Arduino” de PanamaHitek el cual nos ayudará a realizar la comunicación con arduino.

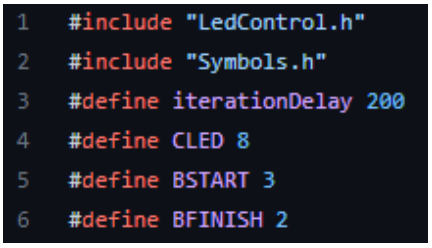
Línea 26-32: Se Encuentra el constructor de la clase Interfaz.java.
arduinoTX define el puerto serial con velocidad de 9600 Baudios.

Código

Descripción
Línea 416-425: Obtenemos la cadena ingresada en el TextArea y lo enviamos hacia Arduino y Proteus mediante la comunicación serial.

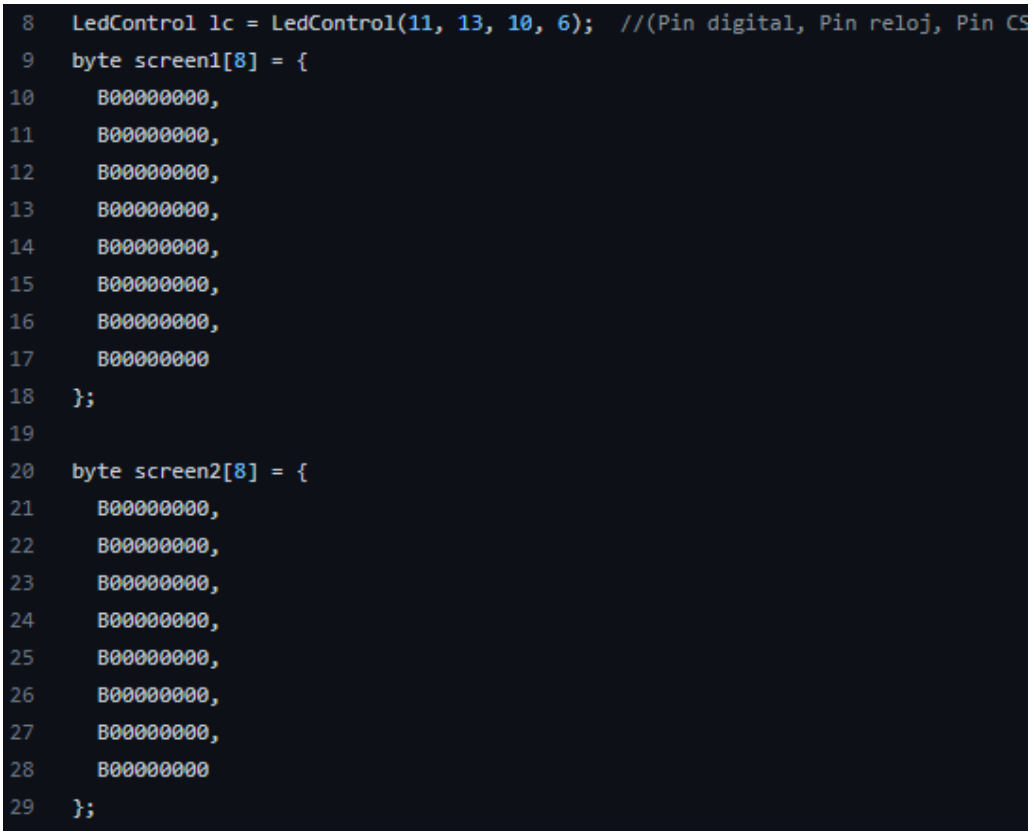
```
415
416     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_jButton1ActionPerformed
417         try {
418             String message = jTextArea1.getText();
419             Arduino.sendData(message);
420         } catch (ArduinoException ex) {
421             Logger.getLogger(Interfaz.class.getName()).log(Level.SEVERE, null, ex);
422         } catch (SerialPortException ex) {
423             Logger.getLogger(Interfaz.class.getName()).log(Level.SEVERE, null, ex);
424         }
425     }//GEN-LAST:event_jButton1ActionPerformed
```

Notas: Utilización de ApacheNetbeas 15 y JDK 19

CÓDIGO ARDUINO

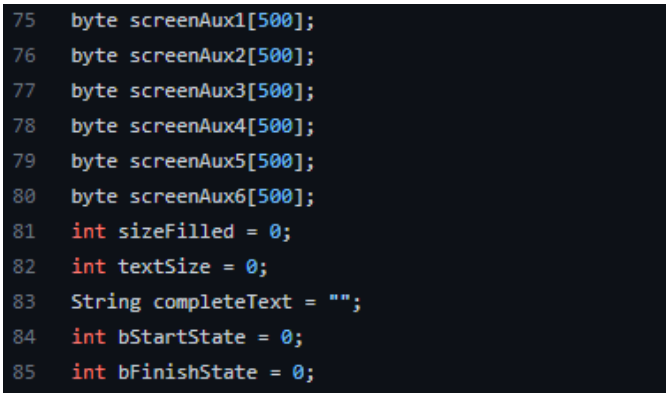
Código	Descripción
	**Línea 1:** Para controladores de pantalla led MAX7221 y MAX7219. **Línea 2:** Pre definidos para setear en la matrices de leds. **Línea 3:** Definido para el retraso en la animación. **Línea 4-6:** Definidos para botones de Start y Finish.

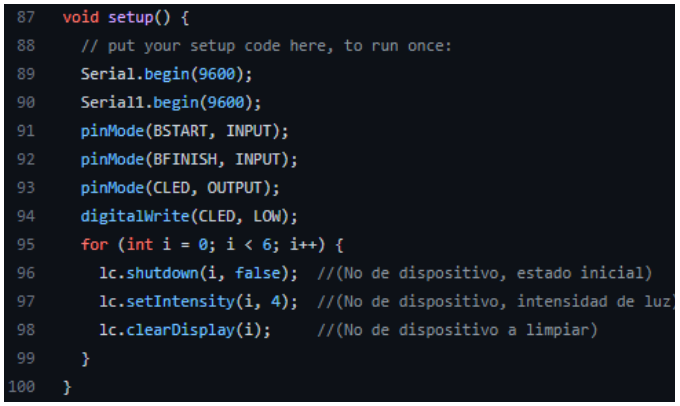
```
1  #include "LedControl.h"
2  #include "Symbols.h"
3  #define iterationDelay 200
4  #define CLED 8
5  #define BSTART 3
6  #define BFINISH 2
```

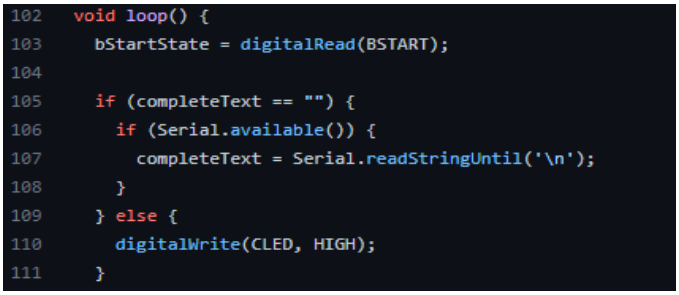
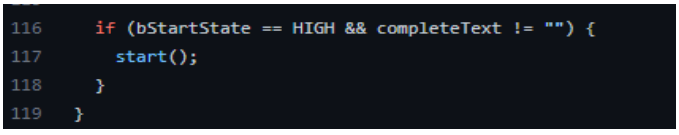
Código


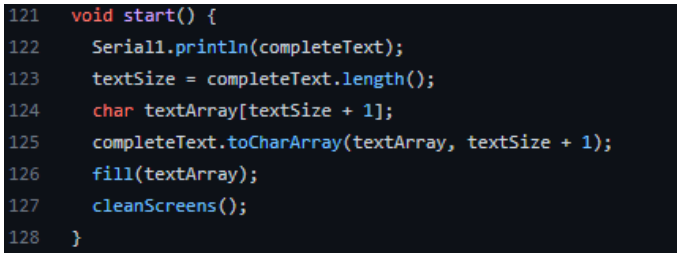
```
8  LedControl lc = LedControl(11, 13, 10, 6);  //(Pin digital, Pin reloj, Pin CS
9  byte screen1[8] = {
10   B00000000,
11   B00000000,
12   B00000000,
13   B00000000,
14   B00000000,
15   B00000000,
16   B00000000,
17   B00000000
18 };
19
20 byte screen2[8] = {
21   B00000000,
22   B00000000,
23   B00000000,
24   B00000000,
25   B00000000,
26   B00000000,
27   B00000000,
28   B00000000
29 };
```

Descripción
Línea 8: Crear un control para los dispositivos. **Línea 9-73:** Matrices de bytes para apagar las matrices de leds.

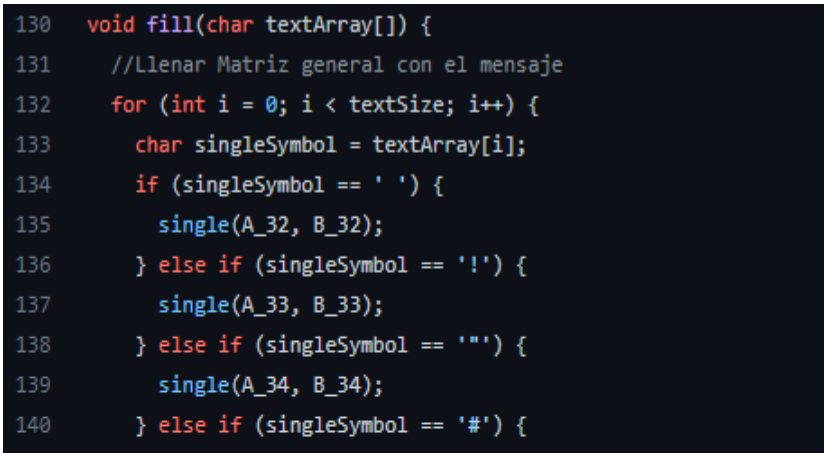
Código	Descripción
	**Línea 75-83:** declaración de las posiciones auxiliares para la matriz de leds. Estados para los botones de inicio / finalizar y variables auxiliares para la cadena de texto.

Código	Descripción
	**Línea 87-100:** Se realiza la configuración de la comunicación serial a 9600 Baudios y se configuran los pines especificados para entrada en el pin 3,2 y 8 para salida respectivamente. ***digitalWrite(CLED, LOW)*** envía al pin CLED (8) a un estado bajo. Describe los estados del dispositivo con un for iterado de 0 a 6.

Código	Descripción
 	**Línea 102-119:** Realiza la lectura de un pin específico si este está activo realiza la comunicación serial y muestra el resultado en la matriz de leds de lo de lo contrario apaga todas las matrices de LED

Código	Descripción
	**Línea 121-128:** Recibe y convierte a texto para la función. ***fill()*** realiza el llenado de las matrices de leds. ***cleanScreens()*** finaliza y limpia las matrices de leds.

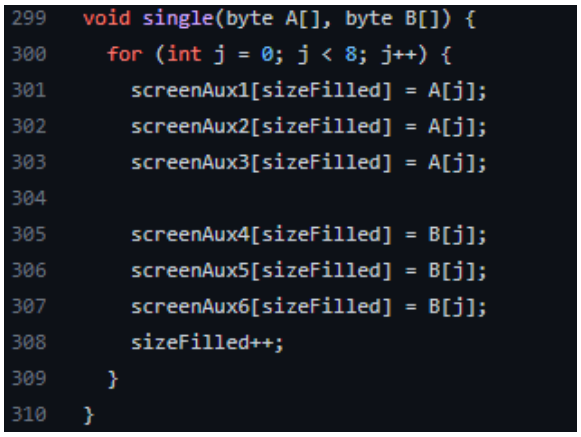
Código



Descripción

Línea 130-331: For que realiza la clasificación de cada uno de los caracteres enviados y mediante ***single()*** envía las matrices declaradas en ***Simbols.h*** para su posterior impresión en las matrices de leds.

Línea 330: Realiza el llamado a la función que realiza la animación en la matriz de leds.

Código	Descripción
	**Línea 299-310:** Recorre las matrices de bytes declarados en “***simbols.h***” y las almacena en las posiciones auxiliares.

```
299  void single(byte A[], byte B[]) {
300    for (int j = 0; j < 8; j++) {
301      screenAux1[sizeFilled] = A[j];
302      screenAux2[sizeFilled] = A[j];
303      screenAux3[sizeFilled] = A[j];
304
305      screenAux4[sizeFilled] = B[j];
306      screenAux5[sizeFilled] = B[j];
307      screenAux6[sizeFilled] = B[j];
308      sizeFilled++;
309    }
310  }
```

Código

```
383  void cleanScreens() {
384    for (int i = 0; i < 500; i++) {
385      screenAux1[i] = B00000000;
386      screenAux2[i] = B00000000;
387      screenAux3[i] = B00000000;
388      screenAux4[i] = B00000000;
389      screenAux5[i] = B00000000;
390      screenAux6[i] = B00000000;
391    }
392    for (int i = 0; i < 8; i++) {
393      screen1[i] = B00000000;
394      screen2[i] = B00000000;
395      screen3[i] = B00000000;
396      screen4[i] = B00000000;
397      screen5[i] = B00000000;
398      screen6[i] = B00000000;
399    }
400    for (int j = 0; j < 8; j++) {
401      lc.setRow(0, j, screen1[j]);  //(No de dispositivo, fila, valor)
402      lc.setRow(1, j, screen2[j]);  //(No de dispositivo, fila, valor)
403      lc.setRow(2, j, screen3[j]);  //(No de dispositivo, fila, valor)
404      lc.setRow(3, j, screen4[j]);  //(No de dispositivo, fila, valor)
405      lc.setRow(4, j, screen5[j]);  //(No de dispositivo, fila, valor)
406      lc.setRow(5, j, screen6[j]);  //(No de dispositivo, fila, valor)
407    }
408    sizeFilled = 0;
409  }
```

Descripción
Línea 383-409: ***CleanScreens()*** función utilizada para recorrer las matrices y dejarlas en estado 0.