

MATERIA

Sistemas Programables

CARRERA

Ingeniería en sistemas Computacionales

PRESENTA:

Angely Yazmin Ramírez Martínez

Salvador Suarez Quijas

NOMBRE DE LA MAESTRA:

Ing. Carlos Rafael Levy Rojas

LEÓN, GUANAJUATO

Periodo: agosto - dic 2018

Codigo arduino

```
#include <LiquidCrystal.h>

// pines para pantalla lcd
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int imprimir=0;           // impresion del mensaje
String Mensaje="";        //obtendra el mensaje

//temperatura en la variable c se guardara el resultado de la operacion
byte PIN_SENSOR = A0;     //coneccion sensor lm35
int dato_serial = 0;
float C;
int temp;                 //resultado de temperatura

//tiempo en milisegundos
//variable de tiempo
unsigned long time;
unsigned long t = 0;      //se inicializa t para repetir los mensajes
int Dt = 100;

void setup(){
  pinMode(10,OUTPUT);      //pin que se dirige al led
  digitalWrite(10,HIGH);   //escribe
  lcd.begin(16, 2);        //Inicializa la interfaz a la pantalla LCD (ancho y alto)

  Serial.begin(9600);      // datos en bits por segundo
  //estaes del reloj
  Serial.setTimeout(50);   //tiempo de espera de dato
}

void loop(){
  //reloj
  String text = Serial.readString(); //lee caracteres del buffer serial en una cadena. La función
  finaliza si se agota el tiempo de espera (ver setTimeout ())
  String line1 = text.substring(0, 16); //Obtener una subcadena de una cadena
  String line2 = text.substring(16, 32); //Obtener una subcadena de una cadena

  C = (5.0 * analogRead(PIN_SENSOR) * 100.0)/ 1024; //operacion para determinar la temperatura

  if ( Serial.available() > 0){ //Obtiene la cantidad de bytes (caracteres) disponibles para leer desde el
  puerto serie
    //lectura_datos();
    dato_serial = Serial.read(); //la variable lee los datos de serie entrantes
    comparacion_datos();
  }

  lcd.setCursor(0,0); //nos posicionamos en el 0,0
  lcd.print("C= Grados"); //imprime el mensaje
  lcd.setCursor(2,0); //nos posicionamos en el 2,0
  lcd.print(C); //se imprime la temperatura
  temp = C; //guardamos C en temp
  //
  //
```

```

int cuenta=0;                //variable para llevar conteo de mensaje
int caracteres=0;            //caracteres entrantes

while (Serial.available()>0){    // leer datos desde el puerto serie hara los mandara para
    convertir a ASCII
    Mensaje=Mensaje+Decimal_to_ASCII(Serial.read());
    //text = text + Decimal_to_ASCII(Serial.read());
    cuenta++;                //lleva la cuenta de cada vez que pasa un mensaje en partes
}

caracteres=Mensaje.length();    //cuenta los caracteres que tiene el mensaje
delay(10000);                  //espera un tiempo definido

if(text.length() > 0){          //verifica si text es mayor que cero para poder imprimir
    lcd.setCursor(0,0);        //posicion 0,0
    lcd.print(" ");            //se imprime espacio
    lcd.setCursor(0,1);        //se coloca en la posicion 0,0
    lcd.print(" ");            //se imprime
}

    lcd.setCursor(0,0);        //se coloca en la posicion 0,0
    lcd.print(line1);          //se imprime la linea uno de la pantalla
    lcd.setCursor(0,1);        //se coloca en la posicion 0,1
    lcd.print(line2);          //se imprime la linea dos de la pantalla
    delay(10000);              //tiempo

if(time-t > Dt){                //condicion para que se cicle
    if (caracteres>16){          //condicion que si es mayor a 16 siga para imprimir
        if (Mensaje!=""){        //si es diferente de nulo lo va a imprimir
            lcd.clear();          //limpia la pantalla lcd
            lcd.print(Mensaje.substring(0,16));    //imprime la primera linea, primeros 16 caracteres
            lcd.setCursor(0,1);    //nos posicionamos en la 0,1
            lcd.print(Mensaje.substring(16,caracteres)); //imprime 16 caracteres
        }
    }
    else                          //de lo contrario realiza lo siguiente
    {
        //
        if (Mensaje!=""){        //limpia e imprime
            t = time;
            lcd.clear();
            lcd.print(Mensaje);
        }
    }
    delay(10000);                //espera un tiempo(10 segundos)
    Mensaje="";                  //saca mensaje
    lcd.clear();                  //limpi pantalla
    //temperatura
}

void lectura_dato (void ){      //metodo para leer el serial, cada caractaer
    dato_serial = Serial.read();
}

void comparacion_dato (void){    //si es vacion escribe el temp
    if(dato_serial == ' '){
        Serial.write(temp);
    }
}

```

```
}  
}
```

```
char Decimal_to_ASCII(int entrada){ //recepcion de java y lo convierte a ASCII para poder imprimir en pantalla  
    char salida='';  
    switch(entrada){  
    case 32:  
        salida=' '  
        break;  
    case 33:  
        salida='!';  
        break;  
    case 34:  
        salida='\"';  
        break;  
    case 35:  
        salida='#';  
        break;  
    case 36:  
        salida='$';  
        break;  
    case 37:  
        salida='%';  
        break;  
    case 38:  
        salida='&';  
        break;  
    case 39:  
        salida=' ';  
        break;  
    case 40:  
        salida='(';  
        break;  
    case 41:  
        salida=')';  
        break;  
    case 42:  
        salida='*';  
        break;  
    case 43:  
        salida='+';  
        break;  
    case 44:  
        salida=',';  
        break;  
    case 45:  
        salida='-';  
        break;  
    case 46:  
        salida='.';  
        break;  
    case 47:  
        salida='/';  
        break;  
    case 48:  
        salida='0';
```

```
break;
case 49:
salida='1';
break;
case 50:
salida='2';
break;
case 51:
salida='3';
break;
case 52:
salida='4';
break;
case 53:
salida='5';
break;
case 54:
salida='6';
break;
case 55:
salida='7';
break;
case 56:
salida='8';
break;
case 57:
salida='9';
break;
case 58:
salida='.';
break;
case 59:
salida=',';
break;
case 60:
salida='<';
break;
case 61:
salida='=';
break;
case 62:
salida='>';
break;
case 63:
salida='?';
break;
case 64:
salida='@';
break;
case 65:
salida='A';
break;
case 66:
salida='B';
break;
case 67:
salida='C';
break;
case 68:
```

```
salida='D';  
break;  
case 69:  
salida='E';  
break;  
case 70:  
salida='F';  
break;  
case 71:  
salida='G';  
break;  
case 72:  
salida='H';  
break;  
case 73:  
salida='I';  
break;  
case 74:  
salida='J';  
break;  
case 75:  
salida='K';  
break;  
case 76:  
salida='L';  
break;  
case 77:  
salida='M';  
break;  
case 78:  
salida='N';  
break;  
case 79:  
salida='O';  
break;  
case 80:  
salida='P';  
break;  
case 81:  
salida='Q';  
break;  
case 82:  
salida='R';  
break;  
case 83:  
salida='S';  
break;  
case 84:  
salida='T';  
break;  
case 85:  
salida='U';  
break;  
case 86:  
salida='V';  
break;  
case 87:  
salida='W';  
break;
```

```
case 88:
salida='X';
break;
case 89:
salida='Y';
break;
case 90:
salida='Z';
break;
case 91:
salida='[';
break;
case 92:
salida=' ';
break;
case 93:
salida=']';
break;
case 94:
salida='^';
break;
case 95:
salida='_';
break;
case 96:
salida='\"';
break;
case 97:
salida='a';
break;
case 98:
salida='b';
break;
case 99:
salida='c';
break;
case 100:
salida='d';
break;
case 101:
salida='e';
break;
case 102:
salida='f';
break;
case 103:
salida='g';
break;
case 104:
salida='h';
break;
case 105:
salida='i';
break;
case 106:
salida='j';
break;
case 107:
salida='k';
```

```
break;
case 108:
salida='l';
break;
case 109:
salida='m';
break;
case 110:
salida='n';
break;
case 111:
salida='o';
break;
case 112:
salida='p';
break;
case 113:
salida='q';
break;
case 114:
salida='r';
break;
case 115:
salida='s';
break;
case 116:
salida='t';
break;
case 117:
salida='u';
break;
case 118:
salida='v';
break;
case 119:
salida='w';
break;
case 120:
salida='x';
break;
case 121:
salida='y';
break;
case 122:
salida='z';
break;
case 123:
salida='{';
break;
case 124:
salida='|';
break;
case 125:
salida='}';
break;
case 126:
salida='~';
break;
}
```



```

return salida;          //regresa la salida
}

```

Codigo java

Código de interfaz

```

package mensaje;

import gnu.io.CommPortIdentifier;
import gnu.io.PortInUseException;
import gnu.io.SerialPort;
import gnu.io.UnsupportedCommOperationException;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Enumeration;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Ventana2 extends JFrame {
    //
    private JPanel contentPane;
    private JTextField textField;
    Enumeration puertos_libres =null;
    CommPortIdentifier port=null;
    SerialPort puerto_ser = null;
    OutputStream out = null;
    InputStream in = null;
    int temperatura=10;
    Thread timer;
    JLabel lblNewLabel;
    JButton btnNewButton,btnNewButton_1;
    public Ventana2() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 636, 365);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        timer = new Thread(new ImplementoRunnable());
        timer.start();
        timer.interrupt();
        btnNewButton = new JButton("Iniciar");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                puertos_libres = CommPortIdentifier.getPortIdentifiers();
                int aux=0;
                while (puertos_libres.hasMoreElements())
                {

```

```

        port = (CommPortIdentifier) puertos_libres.nextElement();
        int type = port.getPortType();
        if (port.getName().equals(textField.getText()))
        {
            try {
                puerto_ser = (SerialPort) port.open("puerto serial",
2000);

                int baudRate = 9600; // 9600bps

                puerto_ser.setSerialPortParams(
                    baudRate,
                    SerialPort.DATABITS_8,
                    SerialPort.STOPBITS_1,
                    SerialPort.PARITY_NONE);
                puerto_ser.setDTR(true);
                out = puerto_ser.getOutputStream();
                in = puerto_ser.getInputStream();
                btnNewButton_1.setEnabled(true);
                btnNewButton.setEnabled(false);
                timer.resume();
            } catch ( IOException e1) {
            } catch (PortInUseException e1) {
                e1.printStackTrace();
            } catch (UnsupportedCommOperationException e1) {
                e1.printStackTrace();
            }
        }

        break;
    }
}

});
btnNewButton.setBounds(38, 63, 89, 23);
contentPane.add(btnNewButton);

btnNewButton_1 = new JButton("Ver");
btnNewButton_1.setEnabled(false);
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        timer.interrupt();
        puerto_ser.close();
        btnNewButton_1.setEnabled(false);
        btnNewButton.setEnabled(true);
    }
});
btnNewButton_1.setBounds(205, 63, 128, 23);
contentPane.add(btnNewButton_1);

textField = new JTextField();
textField.setBounds(38, 32, 86, 20);
contentPane.add(textField);
textField.setColumns(10);

JLabel lblPuertoCom = new JLabel("Puerto COM");
lblPuertoCom.setBounds(37, 11, 90, 14);
contentPane.add(lblPuertoCom);

lblNewLabel = new JLabel("Temperatura");

```

```

        lblNewLabel.setBounds(80, 124, 128, 24);
        lblNewLabel.setFont(new java.awt.Font("Arial", 0, 20));
        lblNewLabel.setForeground(Color.blue);
        //contentPane.add(lblNewLabel);
    }public void paint(Graphics g) {
        super.paint(g);
        g.setColor(Color.blue);
        g.fillRect(38, 250-temperatura, 20, temperatura);
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    private class ImplementoRunnable implements Runnable{
        int aux;
        public void run() {
            while(true){
                try {
                    out.write(' ');
                    Thread.sleep(100);
                    aux = in.read();
                    if (aux!=2){
                        temperatura = aux;
                        lblNewLabel.setText(""+temperatura+" °C");
                        System.out.println(aux);
                    }repaint();

                } catch (Exception e1) {
                }
            }
        }
    }

    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(0, 400, Short.MAX_VALUE)
                    .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(0, 300, Short.MAX_VALUE)
                    .addContainerGap())
        );

        pack();
    }
    // </editor-fold>

    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Ventana2().setVisible(true);
            }
        });
    }
}

```

CODIGO MENSAJE

```
package mensaje;

import gnu.io.CommPortIdentifier;
import gnu.io.PortInUseException;
import gnu.io.SerialPort;
import gnu.io.UnsupportedCommOperationException;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Enumeration;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EmptyBorder;

public class Ventana extends JFrame {
    ///////
    private JPanel contentPane;
    private JTextField textField;
    Enumeration puertos_libres = null;
    CommPortIdentifier port = null;
    SerialPort puerto_ser = null;
    OutputStream out = null;
    InputStream in = null;
    int temperatura = 10;
    Thread timer;
    JLabel lblNewLabel;
    JButton btnNewButton, btnNewButton_1;
    public Ventana() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 636, 365);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);
        timer = new Thread(new ImplementoRunnable());
        timer.start();
        timer.interrupt();
        btnNewButton = new JButton("Conectar");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                puertos_libres = CommPortIdentifier.getPortIdentifiers();
                int aux = 0;
                while (puertos_libres.hasMoreElements())
                {
                    port = (CommPortIdentifier) puertos_libres.nextElement();
                    int type = port.getPortType();
                    if (port.getName().equals(textField.getText()))
                    {
                        try {
```

```

        serial", 2000);

        puerto_ser = (SerialPort) port.open("puerto

        int baudRate = 9600; // 9600bps

        puerto_ser.setSerialPortParams(
            baudRate,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);
        puerto_ser.setDTR(true);
        out = puerto_ser.getOutputStream();
        in = puerto_ser.getInputStream();
        btnNewButton_1.setEnabled(true);
        btnNewButton.setEnabled(false);
        timer.resume();
    } catch ( IOException e1) {
    } catch (PortInUseException e1) {
        e1.printStackTrace();
    } catch (UnsupportedCommOperationException e1) {
        e1.printStackTrace();
    }

        break;
    }
}

});
btnNewButton.setBounds(38, 63, 89, 23);
contentPane.add(btnNewButton);

btnNewButton_1 = new JButton("Desconectar");
btnNewButton_1.setEnabled(false);
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        timer.interrupt();
        puerto_ser.close();
        btnNewButton_1.setEnabled(false);
        btnNewButton.setEnabled(true);
    }
});
btnNewButton_1.setBounds(205, 63, 128, 23);
contentPane.add(btnNewButton_1);

textField = new JTextField();
textField.setBounds(38, 32, 86, 20);
contentPane.add(textField);
textField.setColumns(10);

JLabel lblPuertoCom = new JLabel("Puerto COM");
lblPuertoCom.setBounds(37, 11, 90, 14);
contentPane.add(lblPuertoCom);

lblNewLabel = new JLabel("Temperatura");
lblNewLabel.setBounds(80, 124, 128, 24);
lblNewLabel.setFont(new java.awt.Font("Arial", 0, 20));
lblNewLabel.setForeground(Color.blue);
contentPane.add(lblNewLabel);
}public void paint(Graphics g) {

```

```

        super.paint(g);
        g.setColor(Color.blue);
        g.fillRect(38, 250-temperatura, 20, temperatura);
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    private class ImplementoRunnable implements Runnable{
        int aux;
        public void run() {
            while(true){
                try {
                    out.write('T');
                    Thread.sleep(100);
                    aux = in.read();
                    if (aux!=2){
                        temperatura = aux;
                        lblNewLabel.setText(""+temperatura+" °C");
                        System.out.println(aux);
                    }repaint();

                } catch (Exception e1) {
                }
            }
        }
    }
}

```

MAIN

```

package mensaje;

public class main {

    public static void main(String[] args) {
        //    Ventana v= new Ventana();
        //    v.setVisible(true);
        //    v.setSize(400,300);

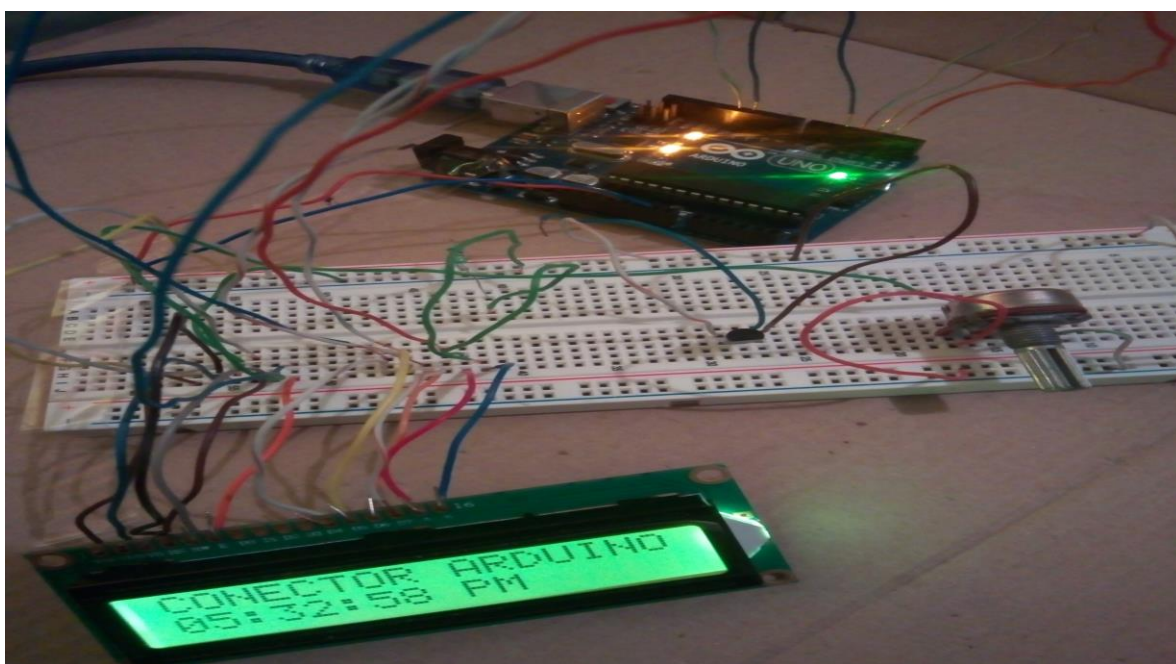
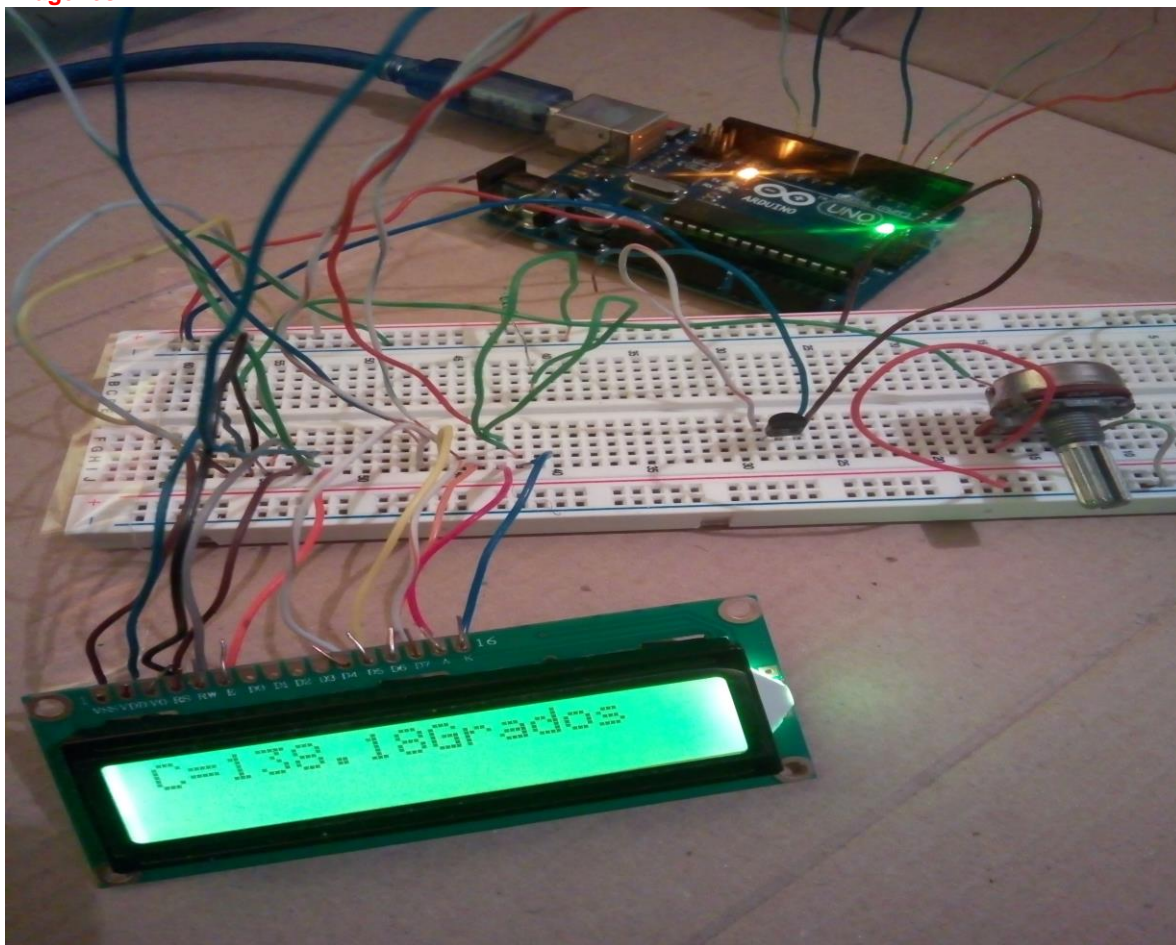
        vista_Mensaje vm = new vista_Mensaje();
        vm.setVisible(true);
        vm.setSize(400,300);

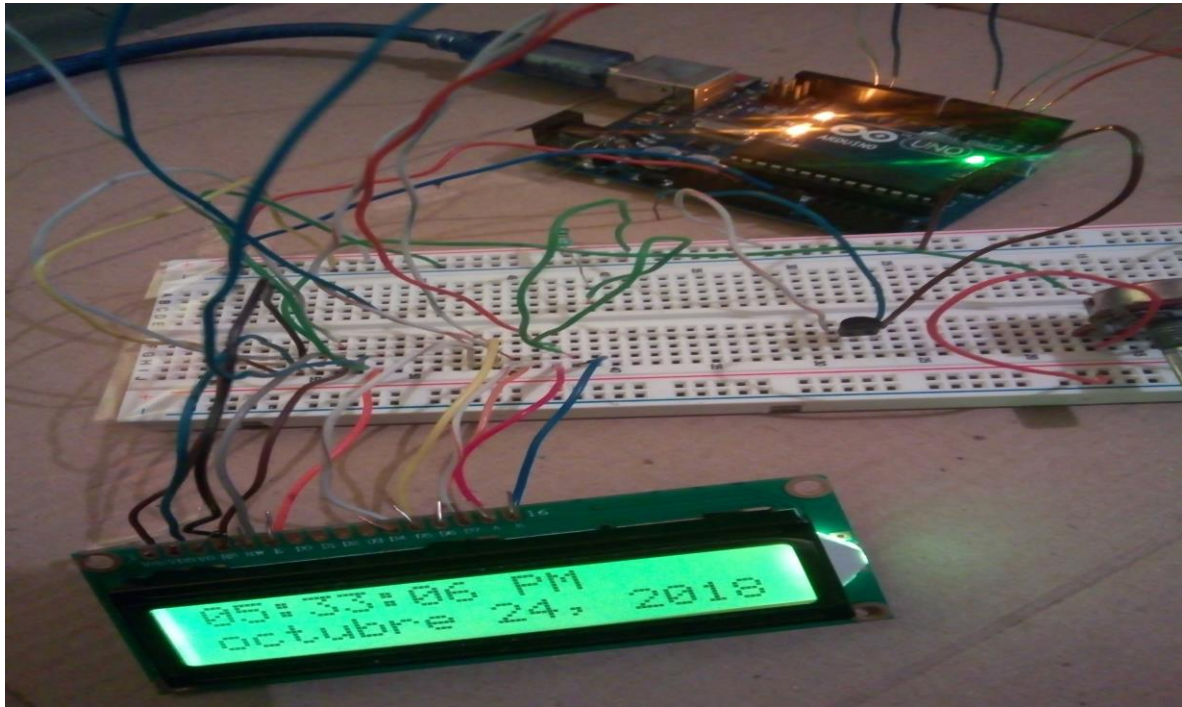
    }

}

```

Imágenes





— □ ×

Escribe un Mensaje

Caracteres disponibles: 32

Ver Mensaje

Limpiar Mensaje

Ver Hora

Ver Temperatura