

Programación en Arduino:

Arduino maestro

CodigoArduino

```
1 #include <LiquidCrystal_I2C.h>
2 #include <Servo.h>
3 #include <Keypad.h>
4 #include <Wire.h> /*esto lo agrego osman*/
5
6 /*****VARIABLES*****/
7 char Contra;
8
9 /*****KEYPAD*****/
10 // Se definen las filas y columnas del Keypad
11 const byte Filas = 4;
12 const byte Columnas = 3;
13
14 //Definimos la estructura del keypad
15 char keys [Filas][Columnas]={
16   {'1','2','3'},
17   {'4','5','6'},
18   {'7','8','9'},
19   {'*','0','#'}
20 };
21 // Estructura de pines que van conectados al keypad
22 byte rowPins[Filas] = {10,9,8,7}; //PINES DE LAS COLUMNAS DEL TECLADO
23 byte colPins[Columnas] = {13,12,11}; //PINES DE LAS FILAS DEL TECLADO
24 //Teclado
25 Keypad pad = Keypad( makeKeymap(keys), rowPins, colPins, Filas, Columnas ); //METODO PARA OBTENER QUE TECLA SE
26
27
28
29
30 /*****PANTALLA LCD*****/
31 //Crear el objeto lcd dirección 0x20 y 16 columnas x 2 filas
32 LiquidCrystal_I2C lcd(0x20,16,2); //
33
34
35
36 #define S1 6
37 #define S2 5
38
39 void setup() {
```

En esta parte, tenemos el código Arduino que conecta al lado del Arduino maestro, quien es el que inicia la simulación. Se incluyeron 4 librerías para el funcionamiento correcto del código, se definieron las teclas del pad numérico y también los pines del lcd1 que va conectado al Arduino maestro.

```

55
56
57 void loop() {
58   // put your main code here, to run repeatedly:
59   lcd.clear();
60   digitalWrite(S2,LOW);
61   lcd.setCursor(3,0);
62   lcd.print("CASA ACYE1");
63   lcd.setCursor(4,1);
64   lcd.print("B-G04-S2");
65   delay(100);
66   String Contraseña = "";
67   int Temp = 0;
68   while(true){
69     Contra = pad.getKey();
70     if (Contra == '*'){ // * SIMBOLIZA EL FIN DE LA CONTRASEÑA
71       break;
72       //Serial.print('*');
73     }
74     if (Contra != NO_KEY){
75       Contraseña += Contra;
76       Serial.print(Contraseña);
77       Serial.print(" ");
78       Temp++;
79     }
80     if (Contra == '#'){ // SIMBOLIZA EL APAGADO DE LOS MOTORES
81       digitalWrite(S2,HIGH);
82     }
83   }
84   if (Contraseña == "202104"){
85     //ContadorErrores=0;
86     lcd.clear();
87     lcd.setCursor(3,0);
88     lcd.print("BIENVENIDO");
89     lcd.setCursor(5,1);
90     lcd.print("A CASA");
91     digitalWrite(S1,HIGH);
92     delay(500);
93

```

En el método setup() tenemos la inicialización de los pines y entradas del Arduino, en el método loop() tenemos el código que pide la contraseña por medio del pad numérico y procede a verificar que esta sea correcta para poder darle ingreso a la casa inteligente y poder mostrar un mensaje de bienvenida en el lcd1.

Arduino esclavo

```
Prueba
1 #include <Wire.h>           // include Arduino Wire library (required for I2C devices)
2 #include <LiquidCrystal.h> // include Arduino LCD library
3
4 // LCD module connections (RS, E, D4, D5, D6, D7)
5 LiquidCrystal lcd(13,12,11,10,9,8);
6
7 // define DS1621 I2C slave address (1001+A2+A1+A0)
8 // A2, A1 & A0 connected to GND --> 1001000 = 0x48
9 #define DS1621_ADDRESS 0x48
10 const int E1 = 7;
11 const int E2 = 6;
12 int state = 0;
13
14
15 //----- de 1 motor-----
16 #define M3 3
17 #define M4 4
18
19
20
21 void setup(void) {
22   lcd.begin(16, 2); // set up the LCD's number of columns and rows
23   lcd.setCursor(3, 0); // move cursor to column 0, row 0 [position (0, 0)]
24   lcd.print("APAGADO ");
25   pinMode (E1, INPUT);
26   pinMode (E2, INPUT);
27   Wire.begin(); // join i2c bus
28   // initialize DS1621 sensor
29   Wire.beginTransmission(DS1621_ADDRESS); // connect to DS1621 (send DS1621 address)
30   Wire.write(0xAC); // send configuration register address (Access Config)
31   Wire.write(0); // perform continuous conversion
32   Wire.beginTransmission(DS1621_ADDRESS); // send repeated start condition
33   Wire.write(0xEE); // send start temperature conversion command
34   Wire.endTransmission(); // stop transmission and release the I2C bus
35
36 //----- de 1 motor-----
37 pinMode(M3,OUTPUT) ;
38 pinMode(M4,OUTPUT) ;
39 }
40
```

En esta parte, tenemos el código Arduino que conecta al lado del Arduino esclavo, quien es el que da seguimiento a la simulación. Se incluyeron 2 librerías para el funcionamiento correcto del código, se definieron las entradas para el sensor, motores y también los pines del lcd2 que va conectado al Arduino esclavo.

```

45 // ----- loop -----
46 void loop() {
47     digitalWrite(M3,HIGH) ;
48     digitalWrite(M4,LOW) ;
49     delay(3000) ;
50     digitalWrite(M3,LOW) ;
51     digitalWrite(M4,HIGH) ;
52     delay(3000) ;
53     //-----fin de l motor-----
54
55     state = digitalRead(E1);
56     delay(100);
57     if(state == HIGH){
58         // // wait a second
59         // get temperature in tenths °C
60         int16_t c_temp = get_temperature();
61
62         if(c_temp <= 180) { // if temperature < 18 °C
63             if (c_temp < 0 ) { // if temperature < 0 °C
64                 c_temp = abs(c_temp); // absolute value
65                 sprintf(c_buffer, "-%02u.%1u°C", c_temp / 10, c_temp % 10, 223);
66             }
67             else{
68                 sprintf(c_buffer, " %02u.%1u°C", c_temp / 10, c_temp % 10, 223);
69             }
70             lcd.clear();
71             lcd.setCursor(1,0);
72             lcd.print("TEMP: ");
73             lcd.print(c_buffer); // print c_buffer (temperature in °C)
74             lcd.setCursor(1,1);
75             lcd.print("NIVEL : 1");
76         }
77         if ((c_temp > 180) && (c_temp < 250)){
78             c_temp = abs(c_temp); // absolute value
79             sprintf(c_buffer, " %02u.%1u°C", c_temp / 10, c_temp % 10, 223);
80             lcd.clear();
81             lcd.setCursor(1,0);
82             lcd.print("TEMP: ");
83             lcd.print(c_buffer); // print c buffer (temperature in °C)

```

En el método setup() tenemos la inicialización de los pines y entradas del Arduino, en el método loop() tenemos el código que va verificando la temperatura y, a su vez, va mostrando la temperatura obtenida del sensor y el nivel en el que se encuentra. También tenemos el código donde indicamos a los motores en que momento deben girar.

SIMULACION DEL CIRCUITO

