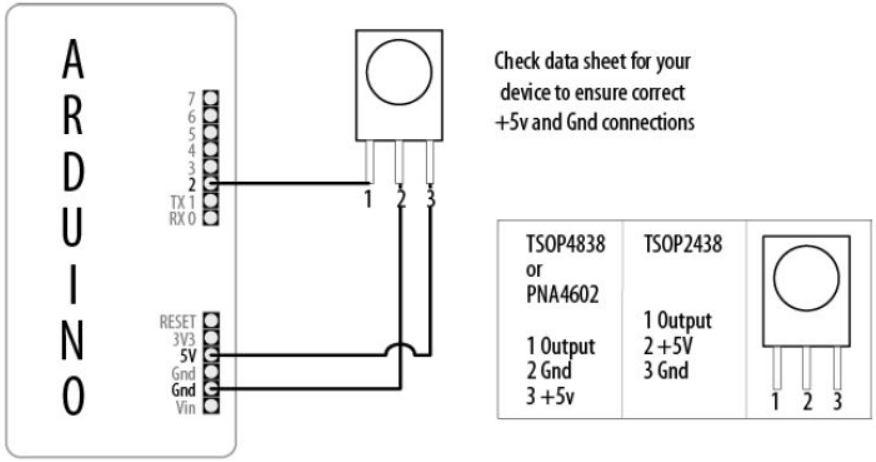
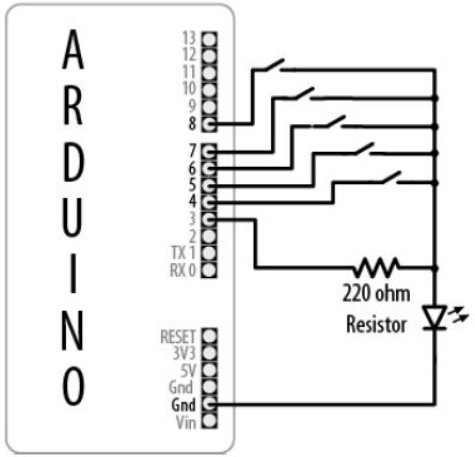
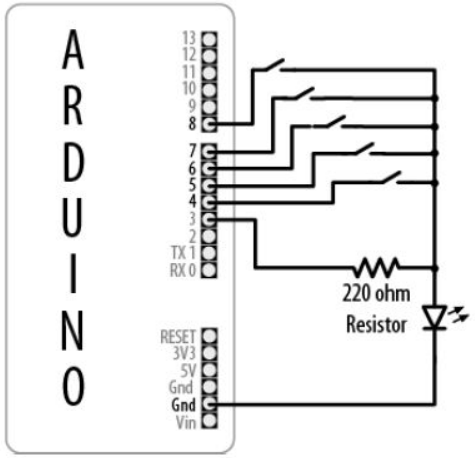
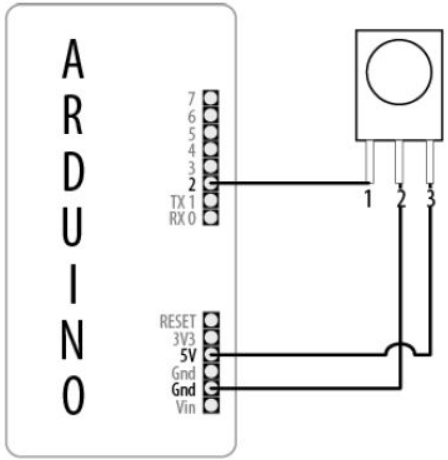




IR Communication

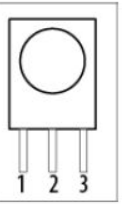
Most remote controls work by sending digital data from a transmitter to a receiver using infrared light (IR). Different protocols (signal patterns) are used to translate key presses into a digital signal





Check data sheet for your device to ensure correct +5v and Gnd connections

TSOP4838 or PNA4602	TSOP2438
1 Output 2 Gnd 3 +5v	1 Output 2 +5V 3 Gnd

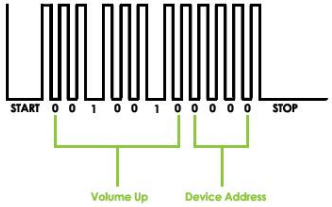


Sender

Remote Control Key Pressed

↓

IR Signals

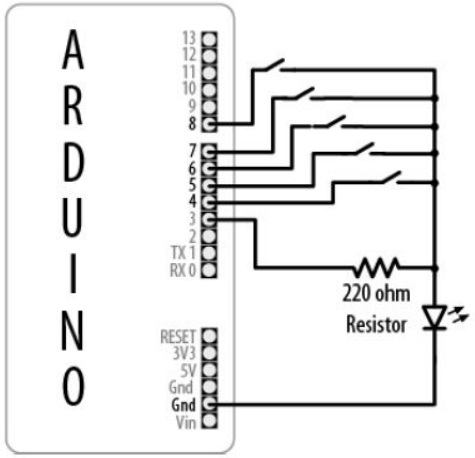


Receiver

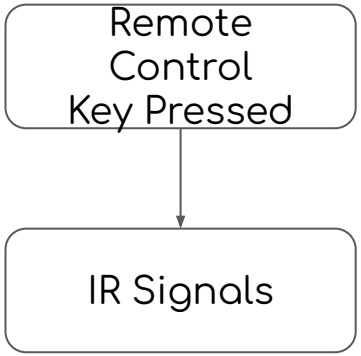
Decode IR Signals

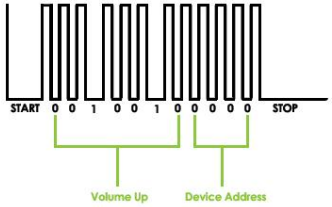
↓

Apply Action



Sender





Config

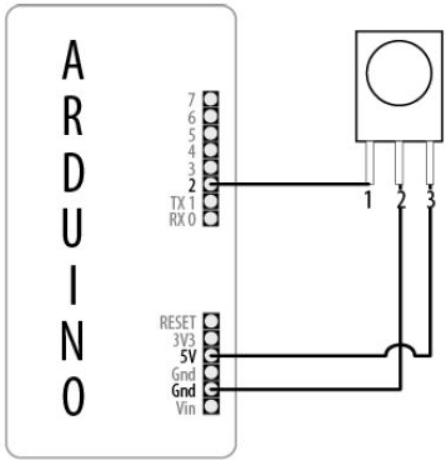
```
#define IR_SEND_PIN 3
#include <IRremote.h>
IRsend sender;
```

Send

```
sender.sendSony(0xC90,12);
```

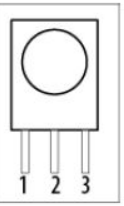
Command = "0010011" Address = "00001"

1	1	0	0	1	0	0	1	0	0	0	0

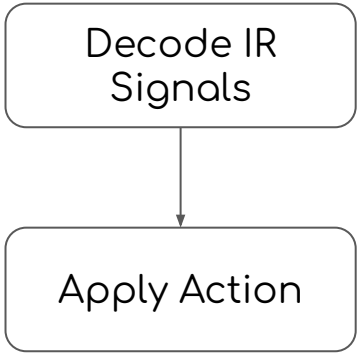


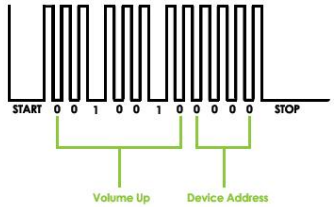
Check data sheet for your device to ensure correct +5v and Gnd connections

TSOP4838 or PNA4602	TSOP2438
1 Output 2 Gnd 3 +5v	1 Output 2 +5V 3 Gnd



Receiver





Command = "0010011" Address = "00001"

1	1	0	0	1	0	0	1	0	0	0	0

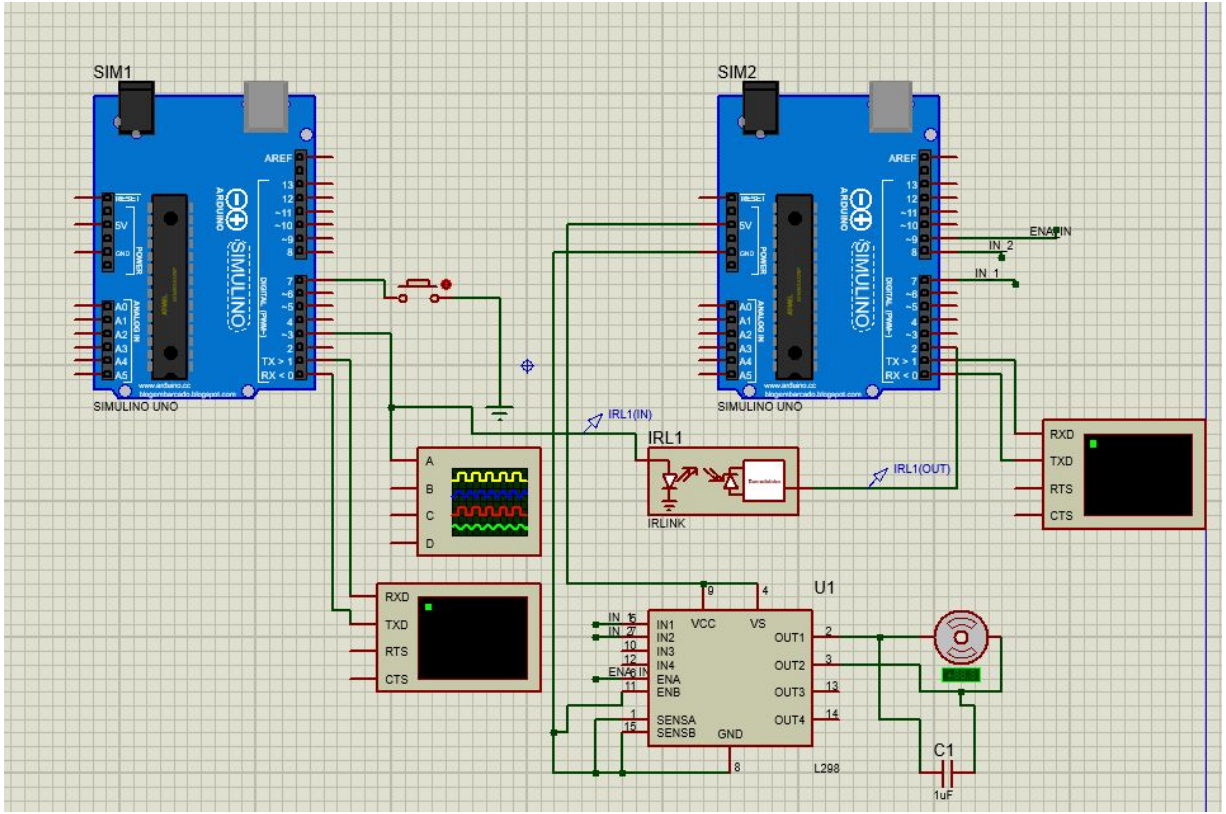
Config

```
#define IR_RECEIVE_PIN 2
#include <IRremote.h>
IRrecv receiver(IR_RECEIVE_PIN);
void setup() {
receiver.enableIRIn();
}
```

Receive

```
void loop(){
  if(receiver.decode()){
    if(receiver.decodedIRData.command==19){
      // Make Action
    }
  receiver.resume();
}}
```

It is required to control the fan speed using a remote control with infrared sensor, develop two embedded systems (sender/receiver) that allow user to supply four speed levels (Off, Slow, Medium, Fast).



It is required to control the fan speed using a remote control with infrared sensor, develop two embedded systems (sender/receiver) that allow user to supply four speed levels (Off, Slow, Medium, Fast).

OFF → Command = "0010011"

1	1	0	0	1	0	0	1	0	0	0	0

Slow → Command = "0010100"

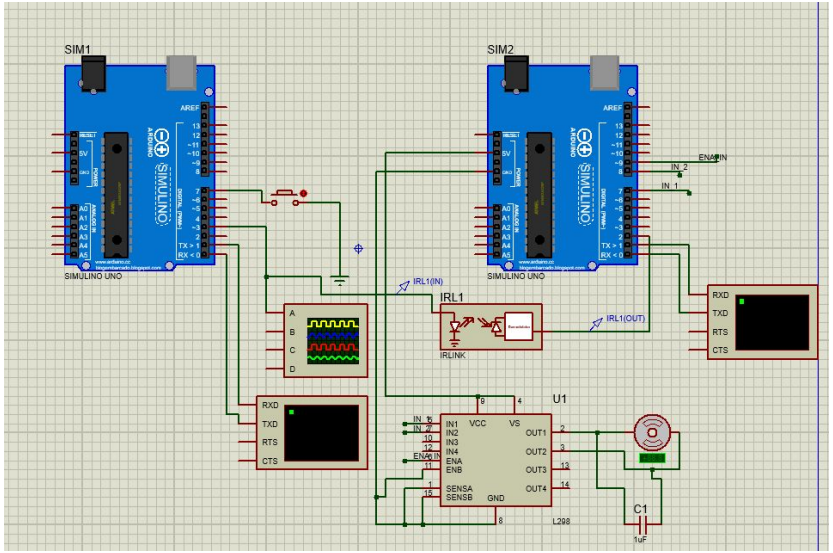
0	0	1	0	1	0	0	1	0	0	0	0

Medium → Command = "0010101"

1	0	1	0	1	0	0	1	0	0	0	0

Fast → Command = "0010110"

0	1	1	0	1	0	0	1	0	0	0	0

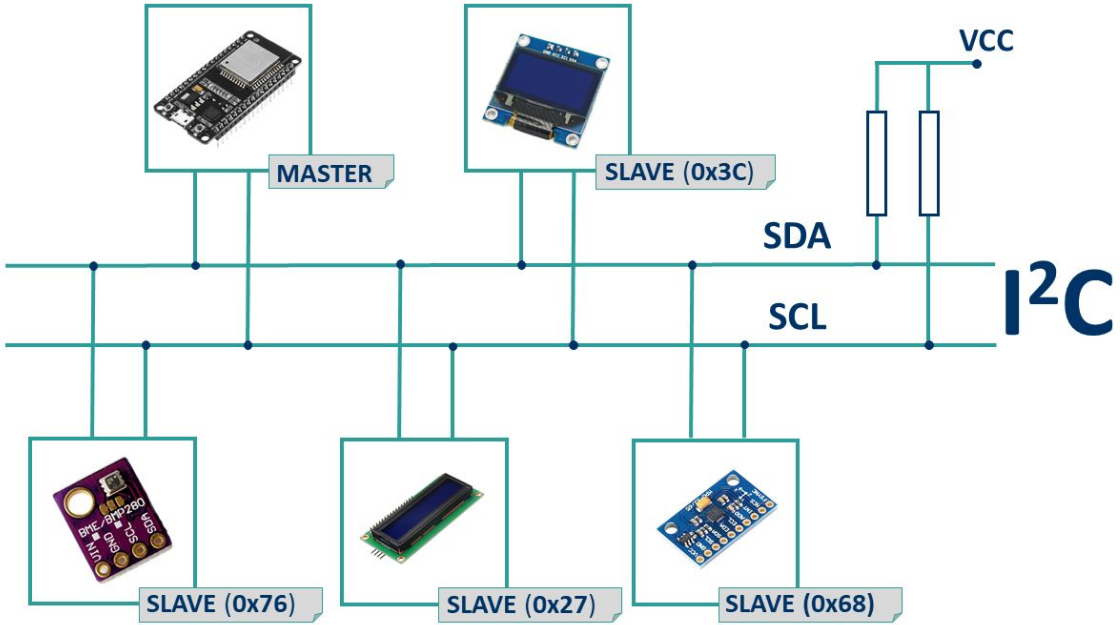


I2C

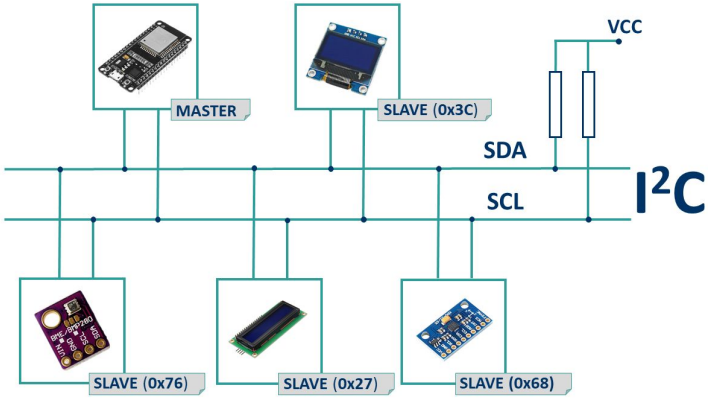
It is a kind of serial communication technology which is originally designed for exchanging data between multiple IC chips.

I2C is implemented by only Two Wires (SDA, SCL), it is also called TWI (Two Wire Interface).

A4 SDA, A5 SCL



I2C



Send

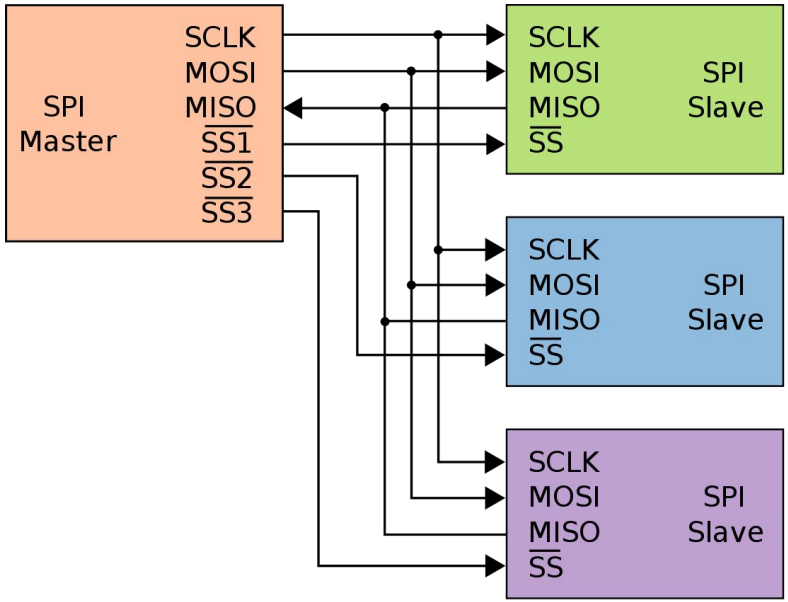
```
#include <Wire.h>
void setup() {
Wire.begin();
}
void loop()
{
      Wire.beginTransmission(ADDRESS);
      Wire.write(data);
      Wire.endTransmission();
}
```

Receive

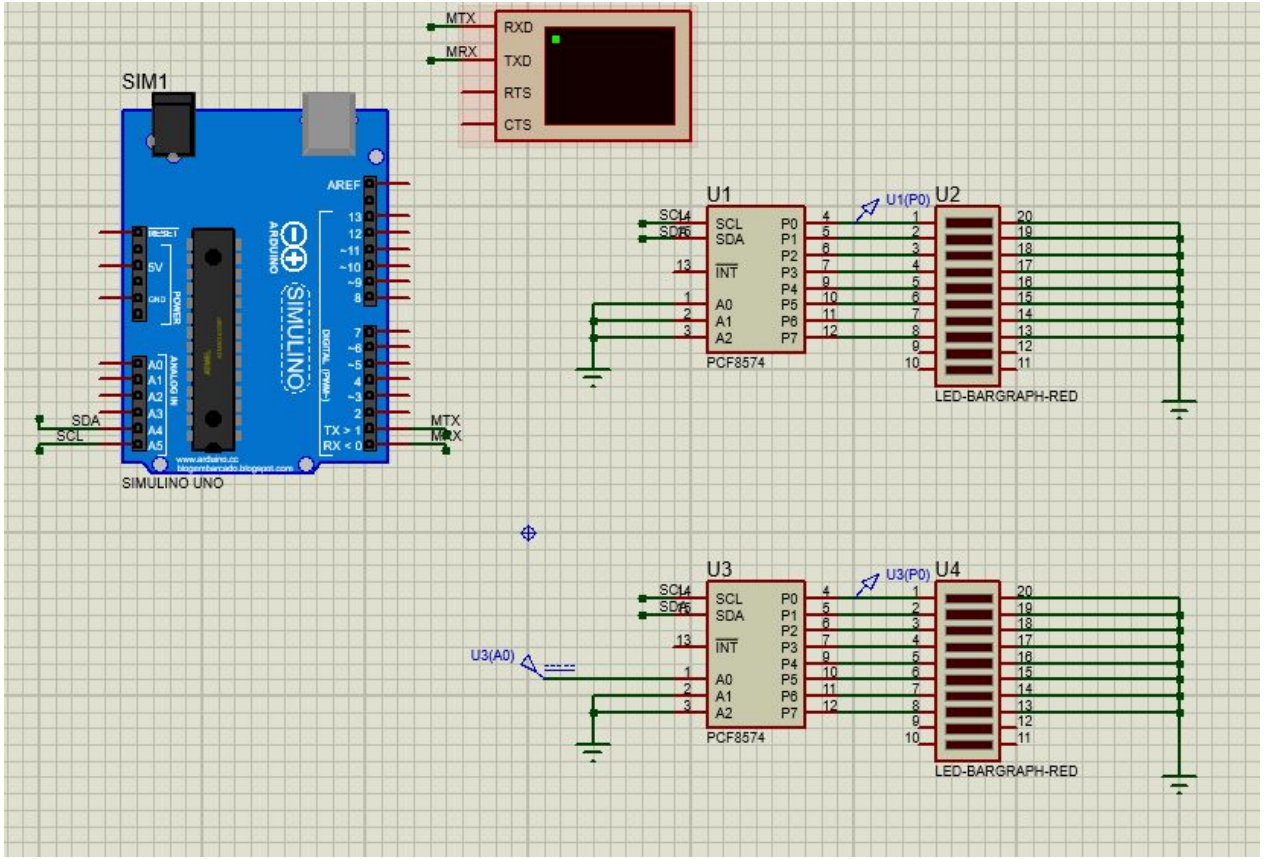
```
#include <Wire.h>
void setup() {
Wire.begin();
}
void loop()
{
      Wire.requestFrom(ADDRESS,1);
      byte data = Wire.read();
}
```

SPI

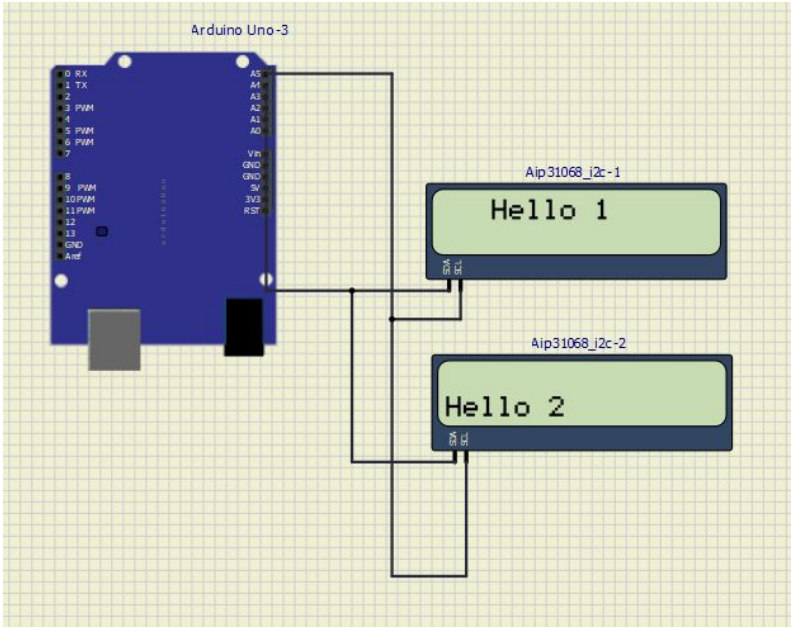
Serial Peripheral Interface (SPI) is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two microcontrollers. Arduino UNO (SCLK : 13, MISO : 12, MOSI0 : 11, SS : 10)



Using I2C port expander to create a bouncing light system with 16 LED.

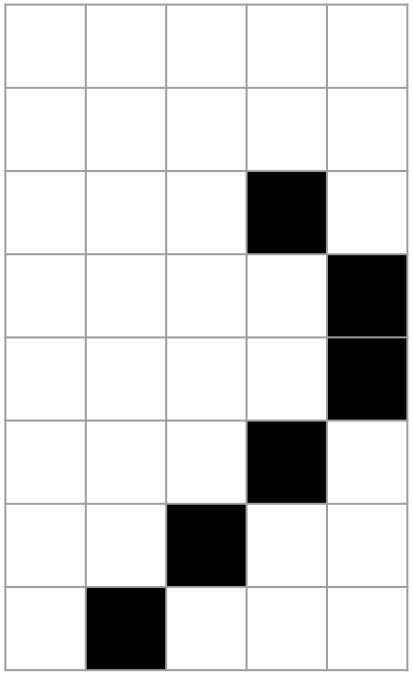


LCD

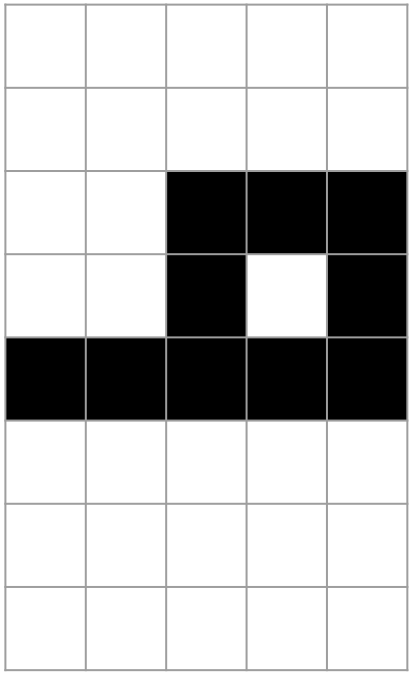



```
#include <LiquidCrystal_AIP31068_I2C.h>
LiquidCrystal_AIP31068_I2C lcd1(0x3E,20,2);
LiquidCrystal_AIP31068_I2C lcd2(0x3F,20,2);
void setup()
{
        lcd1.init();
        lcd1.setCursor(3,0);
        lcd1.print("Hello 1");
        lcd2.init();
        lcd2.setCursor(0,1);
        lcd2.print("Hello 2");
}
void loop()
{
}
```

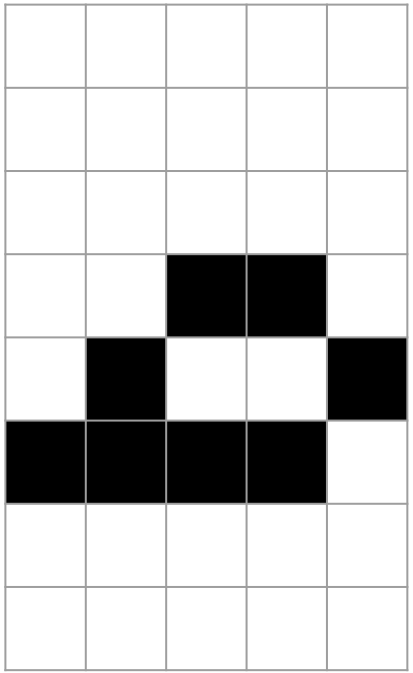
Arabic Characters



```
byte raa[8] =
{
  0b00000,
  0b00000,
  0b00010,
  0b00001,
  0b00001,
  0b00010,
  0b00100,
  0b01000
};
```

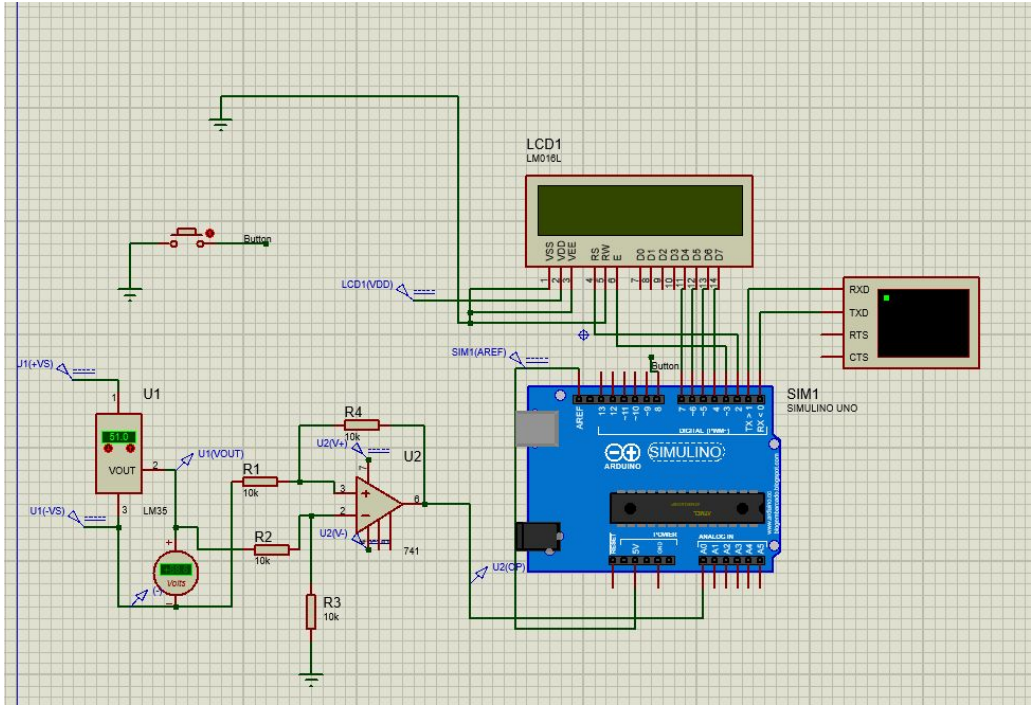


```
byte sad[8] = {
  0b00000,
  0b00000,
  0b00111,
  0b00101,
  0b11111,
  0b00000,
  0b00000,
  0b00000
};
```

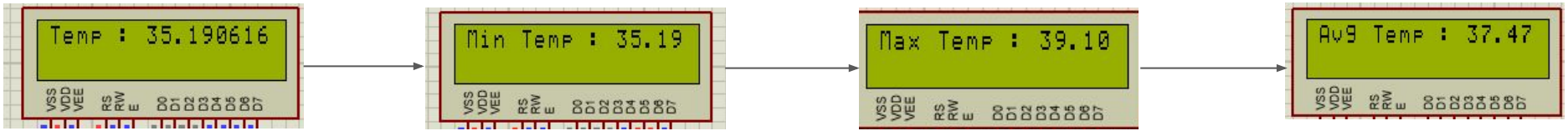


```
byte mem[8] =
{
  0b00000,
  0b00000,
  0b00000,
  0b00110,
  0b01001,
  0b11110,
  0b00000,
  0b00000,
};
```

Develop an embedded system that shows temperature value in an LCD display. Also the system allow user to show min, max, average temperature value in the last hour. User can switch between those values using a single push-button. If the user leaves the button for 30 second the system return to the main screen and show the current temperature value again.



Develop an embedded system that shows temperature value in an LCD display. Also the system allow user to show min, max, average temperature value in the last hour. User can switch between those values using a single push-button. If the user leaves the button for 30 second the system return to the main screen and show the current temperature value again.



Temp sample array : 1 hour / 10 sec = 360 samples

34	33	35	37	40	39	…