EMBEDDED SYSTEMS

MOTION SENSOR

SENSOR	**ARDUINO**
VCC	5 V
GND	GND
DATA PIN	3
LED	+→13, - → GND

CODE (MS):

```
int calibrationTime = 30;

long unsigned int lowIn;

long unsigned int pause = 5000;

boolean lockLow = true;

boolean takeLowTime;

int piroutpin = 3;

int ledPin = 13;

/////////////////////////////

//SETUP

void setup(){

Serial.begin(9600);

pinMode(piroutpin, INPUT);

pinMode(ledPin, OUTPUT);

digitalWrite(piroutpin, LOW);

Serial.print("calibrating sensor ");

for(int i = 0; i < calibrationTime; i++){

Serial.print(".");

delay(1000);

}

Serial.println(" done");

Serial.println("SENSOR ACTIVE");

delay(50);

}
```

```
////////////////////////////

//LOOP

void loop(){

if(digitalRead(piroutpin) == HIGH){

digitalWrite(ledPin, HIGH);

if(lockLow){

lockLow = false;

Serial.println("---");

Serial.print("motion detected at ");

Serial.print(millis()/1000);

Serial.println(" sec");

delay(50);

}

takeLowTime = true;

}

if(digitalRead(piroutpin) == LOW){

digitalWrite(ledPin, LOW);

if(takeLowTime){

lowIn = millis();

takeLowTime = false;

}

if(!lockLow && millis() - lowIn > pause){

lockLow = true;

Serial.print("motion ended at ");

Serial.print((millis() - pause)/1000);

Serial.println(" sec");

delay(50);

}

}

}
```

FLAME SENSOR

SENSOR	**ARDUINO**
D0	A2
GND	GND
VCC	5 V
BUZZER :	**ARDUINO**
GND	GND
VCC	11
I/O	8
LED :	**ARDUINO**
+	13
-	GND

CODE(FS) :

```
const int ledpin=13; // ledpin,flamepin and buzpin are not changed throughout the process
const int flamepin=A2;
const int buzpin=11;
const int threshold=200;// sets threshold value for flame sensor
int flamesensvalue=0; // initialize flamesensor reading
void setup() {
Serial.begin(9600);
pinMode(ledpin,OUTPUT);
pinMode(flamepin,INPUT);
pinMode(buzpin,OUTPUT);
}
void loop() {
flamesensvalue=analogRead(flamepin); // reads analog data from flame sensor
if (flamesensvalue<=threshold) { // compares reading from flame sensor with the threshold value
digitalWrite(ledpin,HIGH); //turns on led and buzzer
tone(buzpin,100);
delay(1000); //stops program for 1 second
}
else{
digitalWrite(ledpin,LOW); //turns led off led and buzzer
noTone(buzpin);
}
}
```

TEMPERATURE SENSOR

- RED : VCC
- BLACK : GND
- YELLOW : DATA WIRE

SENSOR	**ARDUINO**	**RESISTOR**
VCC	VCC	VCC
GND	GND	-
DATA PIN	2/8	DATA PIN
LCD DISPLAY	**ARDUINO**	**BREADBOARD**
SDM	A4	-
SCL	A5	-
VCC	-	RED(VCC)
GND	-	GND

Sensor TO arduino
GND →GND
VCC → 5V
DATA PIN →A0

LCD	**ARDUINO**
1, 3, 5, 16	GND
2, 15	VCC
4	D8
6	D9
11	D0
12	D11
13	D12
14	D13

CODE(TS) :

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4); //The LCD address and size. You can change according you yours
// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 2 //pin for sensor
// Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
/*
The setup function. We only start the sensors here
*/
void setup(void)
{
// start serial port
Serial.begin(9600);
Serial.println("Dallas Temperature IC Control Library Demo");
// Start up the library
sensors.begin();
lcd.init();
// Print a message to the LCD.
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("DS18B20 TEST!");
lcd.setCursor(0, 1);
lcd.print("by miliohm.com");
delay(2000);
lcd.clear();
}
/*
Main function, get and show the temperature
*/
void loop(void)
{
// call sensors.requestTemperatures() to issue a global temperature
// request to all devices on the bus
Serial.print("Requesting temperatures...");
sensors.requestTemperatures(); // Send the command to get temperatures
Serial.println("DONE");
// After we got the temperatures, we can print them here.
// We use the function ByIndex, and as an example get the temperature from the first sensor only.
float tempC = sensors.getTempCByIndex(0);
// Check if reading was successful
if (tempC != DEVICE_DISCONNECTED_C)
{
Serial.print("Temperature for the device 1 (index 0) is: ");
Serial.println(tempC);
lcd.setCursor(0, 0);
lcd.print("Temperature:");
lcd.setCursor(0, 1);
lcd.print(tempC);
lcd.print((char)223);
lcd.print("C");
lcd.print(" | ");
lcd.print(DallasTemperature::toFahrenheit(tempC));
lcd.print(" F");
}
else
{
Serial.println("Error: Could not read temperature data");
}
}
```