

Ex. No: 6	PIR SENSOR BASED OBJECT DETECTION USING LPC2148
Date:	

AIM:

To implement PIR Sensor based object detection using LPC2148 microcontroller.

SOFTWARES/HARDWARES REQUIRED:

S.No	Description	Specification	Quantity
1	ARM Development Kit	LPC 2148	1
2	Keil μ Vision3 IDE	-	-
3	Flash Magic	-	-
4	PIR Sensor	-	1

THEORY:

The PIR sensor itself has two slots in it, each slot is made of a special material that is sensitive to IR. The lens used here is not really doing much and so we see that the two slots can 'see' out past some distance (basically the sensitivity of the sensor). When the sensor is idle, both slots detect the same amount of IR, the ambient amount radiated from the room or walls or outdoors. When a warm body like a human or animal passes by, it first intercepts one half of the PIR sensor, which causes a *positive differential* change between the two halves. When the warm body leaves the sensing area, the reverse happens, whereby the sensor generates a negative differential change. These change pulses are what is detected.

Applications

- Intruder alarms
- Automatic ticket gates
- Entryway lighting
- Security lighting
- Hand dryers
- Automatic door

PROCEDURE:

- Open Keil uVision 4.
- Click Project in tool bar then select New uVision Project, then choose the desired location and give the name of the project and click Save.
- In the pop-up menu, select NXP, then select LPC2148 in the drop-down menu and click
- OK.
- Again, in the pop up menu click Yes to Copy the Startup code to project folder.
- Click File ->New to create a new file and save it as main.c .
- Copy the required Header Files to your project folder.
- In project workspace bar in left, open your Project: name -> Target 1 -> Source Group 1.
- Right click on the Source Group 1 folder and select Add Existing Files to Group 'Source Group 1'... and select the created main.c file and the required header files.
- Write the required program in main.c file and save it.
- Right click on Target 1 in left side project workspace and select Options for Target 'Target 1'.
- In the pop up menu, select Output tab and check Create Hex File. Also select Linker
- tab and check Use Memory Layout from Target Dialog and click OK.
- Compile and Built the program.
- Open Flash Magic.
- Click Select Device and select ARM7 -> LPC2148.
- Set the correct COM PORT where Embedded Kit is connected.
- Set the Baud Rate as 9600 and Oscillator (MHz) as 12.00.
- Check Erase all Flash+Code Rd Prot.
- Click Browse and select the required HEX File of your program from your project folder.
- Check Verify after Programming.
- Connect the Embedded Kit to PC via USB Cable.
- Click Start and check for the software to finish uploading.
- Press Reset button in the embedded kit and check the relative output in the Embedded Kit.

SOURCE CODE:

```
#include<lpc214x.h>
# define bit(x) (1<<x)
#define delay for(i=0;i<7000;i++);
#define PIR (IO1PIN & (1<<24))
unsigned int i;
void lcd_int();
void dat(unsigned char);
void cmd(unsigned char);
void string(unsigned char *);
void main()
{
IO0DIR =0XFFF;
IO1DIR = 0x0;
lcd_int();
cmd(0x80);
string("EMBETRONICX.COM ");
while(1) {
if(PIR == 0) {
string("Intruder Detcted");
}
delay;delay;
cmd(0x01);
}
}
void lcd_int()
{
cmd(0x38);
cmd(0x0c);
cmd(0x06);
cmd(0x01);
cmd(0x80);
}
void cmd(unsigned char a)
{
IO0PIN&=0x00;
```

```

IO0PIN|=(a<<0);
IO0CLR|=bit(8); //rs=0
IO0CLR|=bit(9); //rw=0
IO0SET|=bit(10); //en=1
delay;
IO0CLR|=bit(10); //en=0
}
void dat(unsigned char b)
{
IO0PIN&=0x00;
IO0PIN|=(b<<0);
IO0SET|=bit(8); //rs=1
IO0CLR|=bit(9); //rw=0
IO0SET|=bit(10); //en=1
delay;
IO0CLR|=bit(10); //en=0
}
void string(unsigned char *p)
{
while(*p!='\0') {
dat(*p++);
}
}

```

RESULT:

Thus the PIR Sensor based object detection using LPC2148 microcontroller implemented successfully.

Ex. No: 7	IOT BASED GAS MONITORING SYSTEM USING MQ5 SENSOR
Date:	

AIM:

To implement IoT based gas monitoring system using MQ5 Sensor in Arduino Uno board.

SOFTWARES/HARDWARES REQUIRED:

S.No	Description	Specification	Quantity
1	Breadboard	-	1
2	Arduino Uno	-	1
3	MQ5 Sensor	-	1
4	Connecting wires	-	As Required

THEORY:

Gas sensor is used to detect the gas leakage. The electrical properties of the sensor would change with variations in gas concentration. Sensitive material of MQ-6 gas sensor is SnO₂, which with lower conductivity in clean air. When the target combustible gas exist, the sensor's conductivity is higher along with the gas concentration rising. MQ-6 gas sensor has high sensitivity to Propane, Butane and LPG, also response to Natural gas. The sensor could be used to detect different combustible gas, especially Methane; it is with low cost and suitable for different application

PROCEDURES:

- Open Arduino IDE.
- Click File, select new project. Type the program, then choose the desired location and give the name of the project and click Save.
- Select Board as Arduino UNO and desired Port and press OK
- Compile the program by clicking Sketch and select compile. Compilation is done by clicking Ctrl+R or by using shortcut ✓
- After compilation, check any error present in the program and correct if any error is present and compile again till no errors.
- Click Sketch and select Upload to load the code in Arduino
- Do the connections as per the above circuit in bread board and check the LED toggling.
- Check Verify after Programming.
- Connect the Embedded Kit to PC via USB Cable.
- Click Start and check for the software to finish uploading.
- Press Reset button in the embedded kit and check the relative output in the Embedded Kit

SOURCE CODE:

```
// MQ-2 Gas Sensor Interface with Arduino Uno
```

```
const int sensorPin = A0; // Analog pin connected to the sensor
const int threshold = 400; // Adjust this threshold value according to your needs
```

```
void setup() {
  Serial.begin(9600); // Initialize serial communication
}
```

```
void loop() {
  int sensorValue = analogRead(sensorPin); // Read the analog sensor value
  Serial.print("Sensor Value: ");
  Serial.println(sensorValue);
```

```
  if (sensorValue > threshold) {
    Serial.println("Gas Detected!");
  } else {
    Serial.println("No Gas Detected");
  }
```

```
  delay(1000); // Delay for a second
}
```

RESULT:

Thus the IoT based gas monitoring system using MQ5 Sensor in Arduino Uno board is implemented successfully.

