Shiram Vaish
1BM18CS152
5th Sem 'B'

IOT
02/12/2020

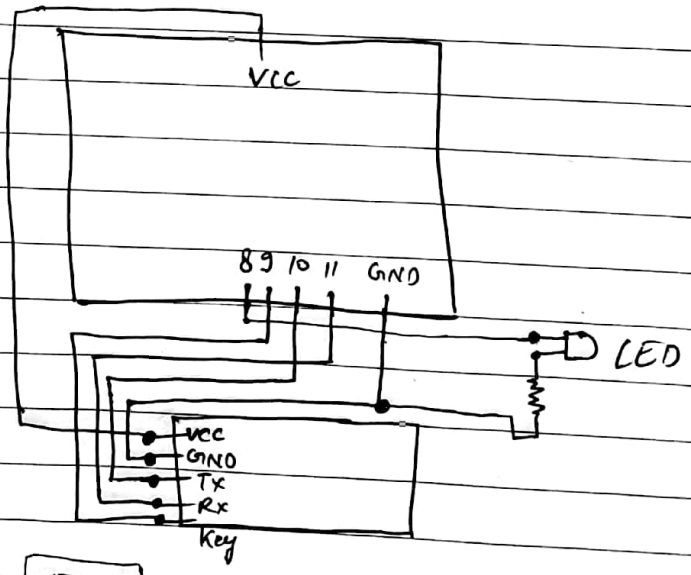
(3) → Aim: To control the led in master device by client device through Bluetooth communication

Component Req:

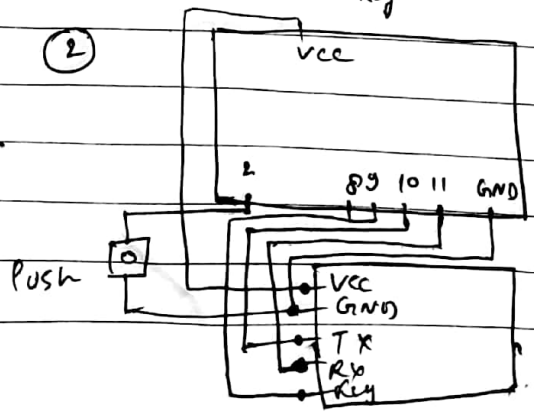
- 2 x Arduino Board
- 2 x Bluetooth Module HC-05
- Jumper Wires
- LED
- Push Button
- Resistors

Circuit Diagram:

(1) Master Circuit →



(2)



Shivam Vaish

classmate
Date
Page

To Connect both the circuits →

- Slave config :-

```
AT+ RMAAD   (clear paired)
AT+ ROLE=0  ( for slave)
AT+ ADDR    (To get bluetooth ID)
AT+ UART= 38400,0,0
```

- Master config →

```
AT+ RMAAD
AT+ ROLE=1
AT+ CMODE=0  (To connect to bluetooth device)
AT+ BIND= bluetooth_id
AT+ UART= 38400,0,0
```

Code in master →

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial (10,11);
int State=0;
const int led=8;
Void setup(){ pinMode(led, OUTPUT); BTSerial.begin(38400)
}
Void loop()
{
    if (BTSerial.available()>0)
    {
        State= BTSerial.read();
    }
    if (State == 1)
    {
        digitalWrite (led, HIGH)
        State=0;
    }
```

Shivam Vaish

```
    else if (State == 0)
    {
        digitalWrite (led, LOW);
        State = 0;
    }
}
```

Code in Slave →

```
#include <SoftwareSerial.h>
SoftwareSerial BTSerial (10,11);
int buttonpin = 2;
int buttonstate = 1;

void setup()
{
    BTSerial.begin(38400);
    pinMode (buttonPin, INPUT);
}

void loop()
{
    if
buttonstate = digitalRead(buttonpin);
if (buttonState == LOW)
    { BTSerial.write("1");
    }
else
    { BTSerial.write("0");
    }
}
```

Thiruvalar

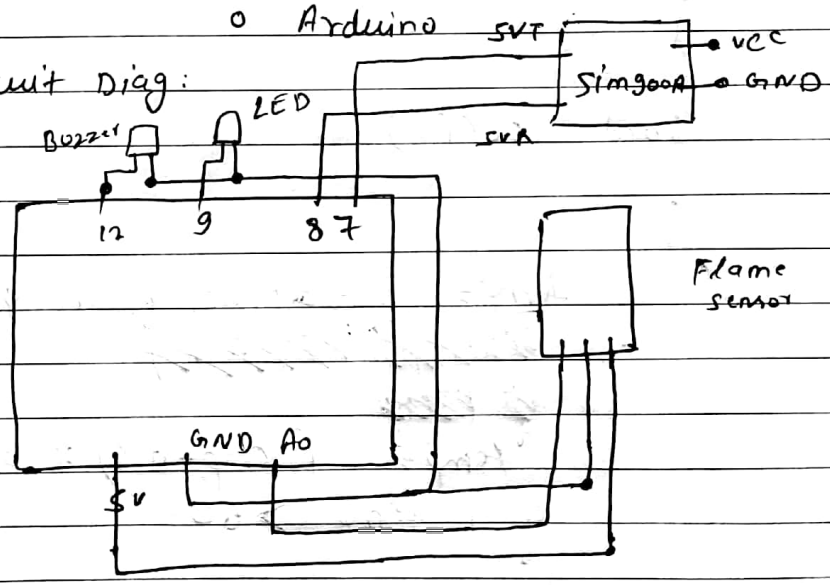
classmate
Date
Page

(9)(b) Aim: To build a Fire Alert System using GSM module.

Components:

- GSM module (SIM900A)
- Flame Sensor
- Sim Card
- LED
- Buzzer
- Arduino

Circuit Diag:



Code:

```
#include <SoftwareSerial.h>
# define flame A0;
# define led 9
# define buzzer 12
#  *
   int temp=0;
   SoftwareSerial SIM900A(7,8)
```

SW-on voice

```
void setup()
{
    PinMode( led, OUTPUT);
    PinMode ( buzzer, OUTPUT);
    SIM900A.begin (9600);
    Serial.begin (9600);
    delay (1000);
}
void loop()
{
    temp= analogRead( flame)* 0.00488;
    temp= (temp-0.5) *100.0;
    if ( temp >60)
    {
        digitalWrite ( LED, HIGH);
        digitalWrite ( Buzzer, HIGH);
        Serial.println ("Fire Fire");
        SIM900A.println("AT+CMGF=1");
        delay (500);
        SIM900A.println("AT+CMGS=\+91
            7691027248");
        delay (100);
        SIM900A.println("FIRE ALERT");
        delay (100);
        SIM900A.println((char)26);
        delay(100);
```

Thiruvalla

classmate
Date
Page

```
SIM900A.println("ATD+917690272848");
Serial.println("All the Alerts Sent");
delay(500);

}
else
{

digitalWrite(LED,LOW);
digitalWrite(Buzzer,LOW);
SIM900A.println("ATH");
Serial.println("Everything is fine");
}

}
```

Shivaraj

classmate
Date
Page

(1) Need of IoT Protocal :-

Iot protocal are critical part of the IOT technology stack, without them, hardware would be rendered useless, as the iot protocol enables it to exchange data in a structured and meaningful way.

In IoT, we always talk about communication interaction b/w sensors, devices, gateways, servers and user Application but what enable all these stuff to talk and interact are protocals.

IoT protocal in each application layer :-

- HTTP : Http is the foundation of the client server model used for web.
- Web socket : WebSocket is a protocal that provides full-duplex communication over a single TCP connection b/w client and server.
- XMPP : Xmpp has its root in instant messaging and prsince information.
- COAP : It is designed for use of low power and constrained network.
- MQTT : It is an open source protocol for constrained devices and low bandwidth, high latency network.

Swarnvaishe

2→ⓐ A Networking system is built to reliably move data. The data is in motion. Prior to level, data is moving through network at the rate and organization determined by the devices generating it.

However some computational activities could occur at level 2 such as protocol translation or application of network security. Level 3, packet inspection.

Most applications ~~such as~~ can't or don't need to process data at network wire speed. Applications typically assume that data is at rest or unchanging. in memory or disk. At Level 4, data communication, data in motion is converted to data at rest.

2ⓑ Facebook Messenger:

Uses Web Realtime Communication (WebRTC) protocol, It is a free, open source project that provide web browser and mobile applications with realtime communication via APIs. It allows audio and video communication to work inside web pages by allowing direct peer to peer Communication, eliminating the need to install plugins or download native apps.

Amazon Web Services (AW) →

It uses standard ~~protocol~~ communication protocols HTTP, MQTT and WebSockets.

Communication is secured using TLS and uses SOAP protocol for older APIs and exclusively JSON for newer ones.

2 (C) → Run IPV6 instead of IPV4 on network to use 6LOWPAN

(i) Most efficient routing: reduces the size of routing tables and makes routing more efficient and hierarchical.

(ii) Most efficient packet processing: IPV6's simplified packet header makes packet processing more efficient compared to IPV4. IPV6 contains no IP level checksum, so the checksum doesnot need to be recalculated at every router hop.

(iii) Different data flows: IPV6 supports multicast rather than broadcast. Multicast allow bandwidth intensive packet flows to be sent to multiple destination simultaneously

(iv) Simplified Network Configuration: Address auto configured is built in IPV6.

(v) Support for new services

(vi) Security