



# TECHNICAL PROJECT REPORT

**TITLE OF INVENTION / PROJECT: WIFI/BLEETOOTH BASED DATA UPLOADING TO SERVER**  
(Digital Attendance)

**NICKNAME : OMNI RECEIVER**

**TEAM MEMBERS / INVENTORS:**

S. No.	Name	Department	Designation	Mobile	E-Mail
1.	Jaskaran Singh	CSE	Student	9872448480	gurjas9872448480@gmail.com
2.	Karan Trehan	CSE	Student	9468303640	karantrehn3@gmail.com
3.	Abhishek	CSE	Student	6284619253	nda1392137@gmail.com
4.	Khushal Thakur	ECE	Mentor	9646030764	khushal.thakur@cumail.in
5.	Anshul Sharma	ECE	Mentor	9478697475	anshulsharma.ece@cumail.in
6.	Kiran Jot Singh	ECE	Mentor	9463909689	kiranjotsingh.ece@cumail.in
7.	Divneet Singh Kapoor	ECE	Mentor	9878422653	divneet.ece@cumail.in

## *Section – 1 (IPR Related)*

### **BRIEF ABSTRACT (500 WORDS):**

We here propose the idea of a digital attendance, portable wireless receiver along with an ID card , based on the concept of wirelessly receiving the inputs from the students through bluetooth connection , with the receiver acting as the Master. Along with a 8 pin piezoelectric system embedded on the card to make the bluetooth chip work.

The idea is to develop a system that first monitors the pressure applied in pascals over few entries and thus enter that data in ID cards to ensure only the real bearer can transmit his/her attendance. Thus preventing multiple entries by an individual, thereby overcoming the problem of proxy.

Whatever information received by the Receiver ( Omni Receiver ) is to be kept in the flash of the Micro Controller rather than using any external Card , thus will prevent any modification or discrepancy



afterwards ,maintaining correctness along with can still maintain the record of an entire month , if for some reason the internet connectivity isn't available.

The ID card has a Bluetooth embedded on it , along with a small battery , which will only transmit signal when all the 8 piezoelectric switches are pressed at the same time and would only work if the sequence of the pressure applied matches to at least the 75% to the recorded values. Along with a LED to notify when the transmission is completed successfully.

The receiver is made up of a MicroController having flash memory and a wifi module and a bluetooth module. It is very portable and can easily be carried by the teacher along with him/her. With the ability to work at the baud rate of 115200 bluetooth can parallelly connect to multiple devices and receive the data simultaneously but before saving getting through a check to ensure no repetition of data.

It will automatically upload the attendance to the excel sheet through server once it comes in the range of the predefined wifi network , and there it can be further processed to be used to mark attendance.

This will prevent any delay, time consumption or the problem of carrying any heavy machinery to carry out this task. Providing a more secure, less hassle and portable way of carrying out daily attendance.

This can further be used to carry other digital information in the ID CARD like account data , banking details etc. to carry transactions with the security of encryption provided by the institute's server.

## EXISTING STATE-OF-THE-ART AND DRAWBACKS IN EXISTING STATE-OF-THE-ART

S. No.	Existing state of art	Drawbacks in existing state of art
1	<b>US6513015B2</b>	Less portability , requires big hardware. Not cheap or Student Centered  Uses visual recognition which is very costly as compared to a Piezoelectric System
2	<b>KR20050039100A</b>	Purposes the idea of installing fixed hardware in class, for receiving and transmitting data, thus lacks portability and has higher rate of risk of getting bugged.

## NOVEL/ADDITIONAL MODIFICATIONS THAT YOU CAN PROPOSE TO IMPROVE UPON DRAWBACKS

- Providing an easy to carry receiver rather than a fixed one..
- Directly saving data in flash rather than SD Card.
- Using other means , such as piezoelectric system which are cost-effective

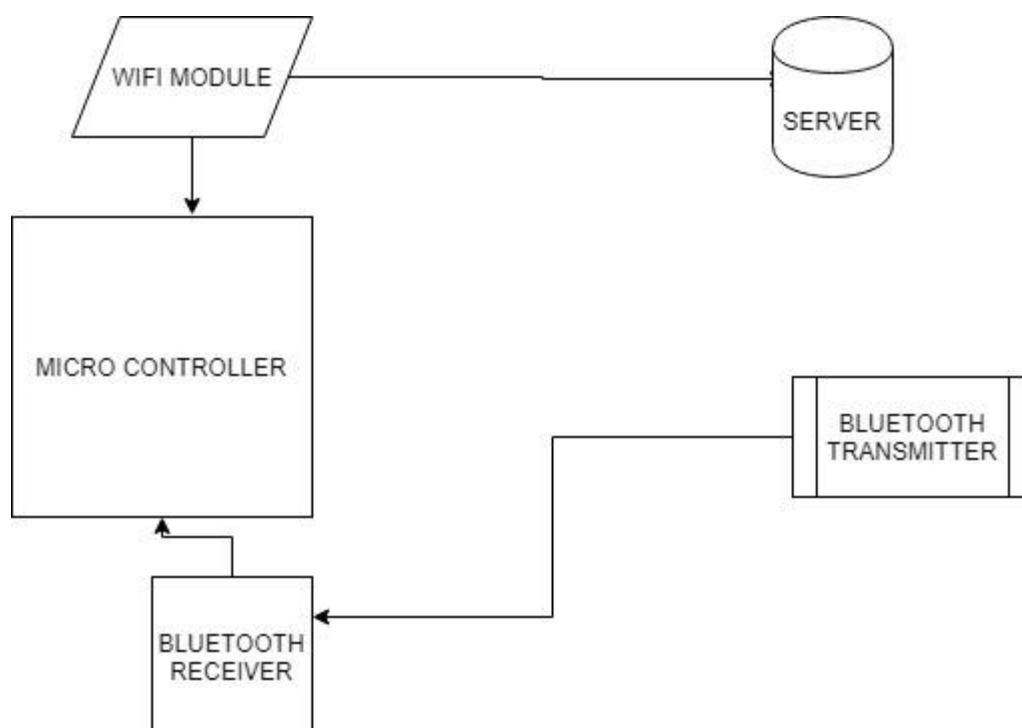


- Pressure-Sensitivity Sequencing minimises redundancy..
- Able to make parallel connection with multiple slave Bluetooth devices at once.

## ADVANTAGES

- Making the receiver portable makes it easy for the bearer to take along plus provides more security as it could not be bugged in his absence.
- Directly saving data in flash , reduces the change of post-editing in original data to minimal,thus will provide more security and correctness.
- Using piezoelectric system on ID card, will make sure an individual person cannot transmit data from 2 cards, as it needs full points to be in contact for creating signal, thus prevents any proxy.

## BLOCK DIAGRAM





## Section – 2 (Real Project)

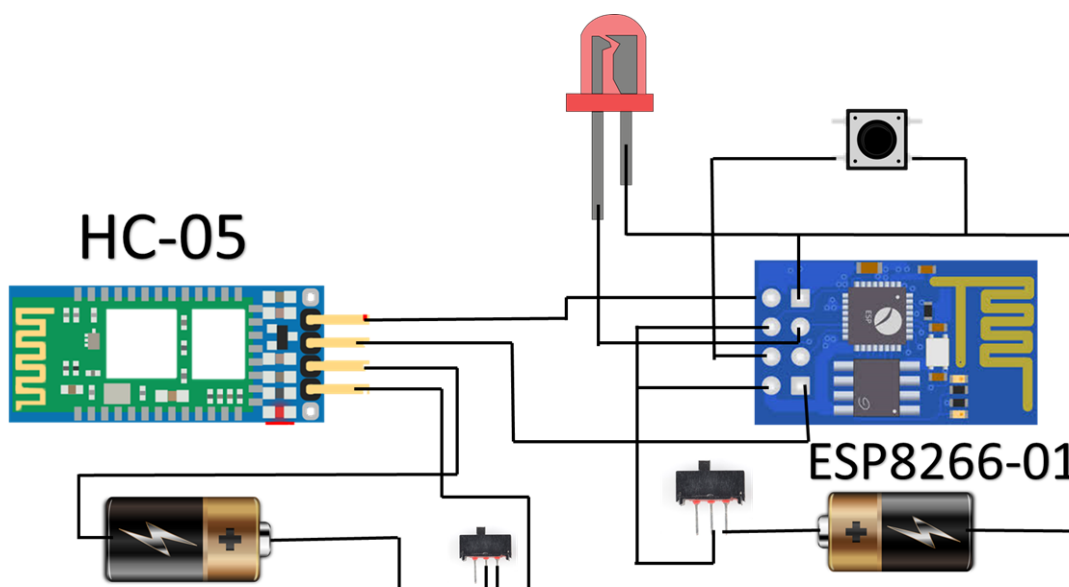
### MATERIALS

(List down the Components, Equipment, etc. actually used in the project)

WIFI MODULE(ESP8266-01), BLUETOOTH MODULE(HC-05) , BATTERY (3.3v, JUMPER WIRES, CASE, LED,RESISTOR

DEVICE CATEGORY	Model No. / Value	Price (INR)/(Per Piece)
WIFI MODULE	ESP8266-01	210/-
BLUETOOTH MODULE	HC-05	270/-
BATTERY	3.3 Volts DC	20/-
	5 Volts DC	30/-
RESISTOR	220 Ohm	1/-
LED	3 Volts	1/-
On/Off Switch	-----	25/-
Push Button	-----	10/-

### CIRCUIT DIAGRAM





## STEPS OF CIRCUIT COMPLETION



1. Take a case, punch the holes required for LED's And Connections of Push Buttons and Switches.

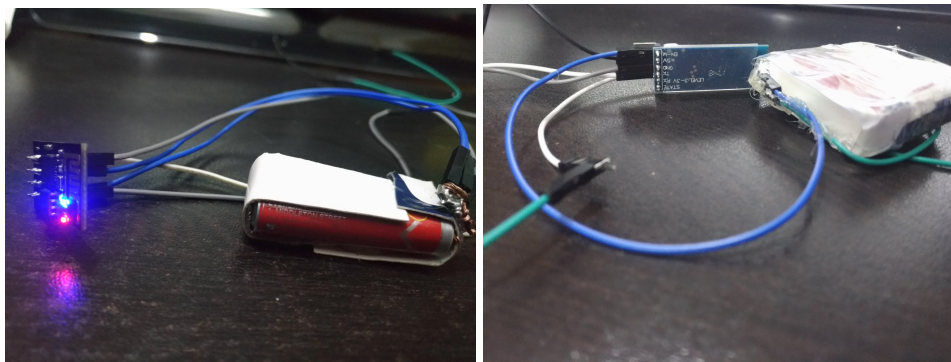


2. Solder The Switches, Led's and Push Button for the Circuit Output needed.





3. Take 1.5 V cells together to form batteries required, i.e. 3.3 V , closing to 3 V and for 5V closing to 4.5V.



4. Connect the circuit of WiFi- Module to the 3.3 V battery and Bluetooth Module to 5V battery , and after which connect both the modules together through RX and TX pins, as per the Circuit Diagram.





5. Arrange the modules and battery in the case , completing the connections.

## PROGRAM CODE

[https://github.com/JSN1999/JSN\\_CSE](https://github.com/JSN1999/JSN_CSE)

CODE:--

```
#include <ESP8266WiFi.h> // valid for both node-mcu and esp8266-01

#include <WiFiClientSecure.h>

const char* ssid = "JSN"; // Enter the name of Wifi-Router with whom you wanna connect
const char* password = "jsnjsnjsn"; // Enter the Router Password to build the connection
String data1="",data2="",data3="";

const char* host = "maker.ifttt.com"; //if using third party app for developing an applet here using IFTTT.com applets

const int httpsPort = 443; // if data to be sent is under http use 80, or if https then use 443

void setup() {

  Serial.begin(115200);

  pinMode(2, OUTPUT);

  pinMode(1,OUTPUT);

  digitalWrite(1,LOW);

  digitalWrite(2,LOW);

  /*Baud Rate , either 9600 or 115200 but 115200 is preferred as it is faster

  9600 baud rate is used as many HC-05 i.e. bluetooth module are not flashed to work for 115200

  Can be updated using AT commands to 115200

  */

  while(!Serial.available()){

    // Send data only when you receive data:
```



```
{  
  
    data1 = Serial.readString();    //Read the incoming data & store into data  
  
    Serial.print(data1);          //Print Value inside data in Serial monitor  
  
    Serial.print("\n");  
  
    {digitalWrite(2,HIGH);  
  
        delay(100);  
  
        digitalWrite(2,LOW);  
  
        delay(50);  
  
        digitalWrite(2,HIGH);  
  
        delay(100);  
  
        digitalWrite(2,LOW);  
  
        delay(50);  
  
        digitalWrite(2,HIGH);  
  
        delay(100);  
  
        digitalWrite(2,LOW);  
  
        delay(50);  
  
        digitalWrite(2,HIGH);  
  
        delay(100);  
  
        digitalWrite(2,LOW);  
  
        delay(50);  
  
    }  
  
}  
  
while(!Serial.available()){  
  
    // Send data only when you receive data:  
  
    {  
  
        data2 = Serial.readString();    //Read the incoming data & store into data  
  
        Serial.print(data2);          //Print Value inside data in Serial monitor  
  
        Serial.print("\n");  
  
    }  
  
}
```





```
digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

delay(50);

digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

delay(50);

digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

delay(50);

digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

delay(50);

}

while(!Serial.available()){

    // Send data only when you receive data:

    {

        data3 = Serial.readString();    //Read the incoming data & store into data

        Serial.print(data3);    //Print Value inside data in Serial monitor

        Serial.print("\n");

        digitalWrite(2,HIGH);

        delay(100);

        digitalWrite(2,LOW);

        delay(50);

        digitalWrite(2,HIGH);
```



```
    delay(100);

    digitalWrite(2,LOW);

    delay(50);

    digitalWrite(2,HIGH);

    delay(100);

    digitalWrite(2,LOW);

    delay(50);

    digitalWrite(2,HIGH);

    delay(100);

    digitalWrite(2,LOW);

    delay(50);

    }

Serial.println();

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.mode(WIFI_STA);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

    delay(500);

    Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// Use WiFiClientSecure class to create TLS connection

WiFiClientSecure client;

Serial.print("connecting to ");
```



```

Serial.println(host);

if (!client.connect(host, httpsPort)) {

    Serial.println("connection failed");

    return;

}

digitalWrite(2,HIGH);

delay(100);

digitalWrite(2,LOW);

String url =
"/trigger/test_data/with/key/oMUA5XRBNIATGHt7Ig_3RnUl3a-10jNyQC4C5D5GAwm";//here give the
link of your resource of applet

// Usually of the form
/trigger/test_data/with/key/oMUA5XRBNIATGHt7Ig_3RnUl3a-1kkayQC4C5D5GAwm

//(Key is not correct, given to just explain the syntax)

Serial.print("requesting URL: ");

Serial.println(url);

String jsonObject = String("{\"value1\":\"\" + data1 + "\",\"value2\":\"\"
+ data2 + "\",\"value3\":\"\" + data3 + "\"}");

//for free acoount values to be passed at once limited to three

//you may replicate the query to other records

digitalWrite(2,HIGH);

client.println(String("POST ") + url + " HTTP/1.1"); //not be changed, it is to be kept at HTTP/1.1 for both
http and https

client.println(String("Host: ") + host);

client.println("Connection: close\r\nContent-Type: application/json");

client.print("Content-Length: ");

client.println(jsonObject.length());

```



```
client.println();

client.println(jsonObject);

Serial.println("request sent");

while (client.connected()) {

    String line = client.readStringUntil('\n');

    if (line == "\r") {

        Serial.println("headers received");

        break;

    }

}

if(!client.available()) {

    Serial.println("No response...");

}

while(client.available()){

    Serial.write(client.read()); // Gives the status after passing the values to website i.e. output shown on
    website

}

delay(50);

digitalWrite(2,LOW);

delay(50);

digitalWrite(2,HIGH);

delay(50);

digitalWrite(2,LOW);

delay(50);

digitalWrite(2,HIGH);

Serial.println("closing connection");

}

void loop() {

}
```