Automated Irrigation system Applied in Crop Farming (ITC's Green House)

Lecturer : HEL CHANTHON

Student : PROEUNG BUNRONG (e20191346)

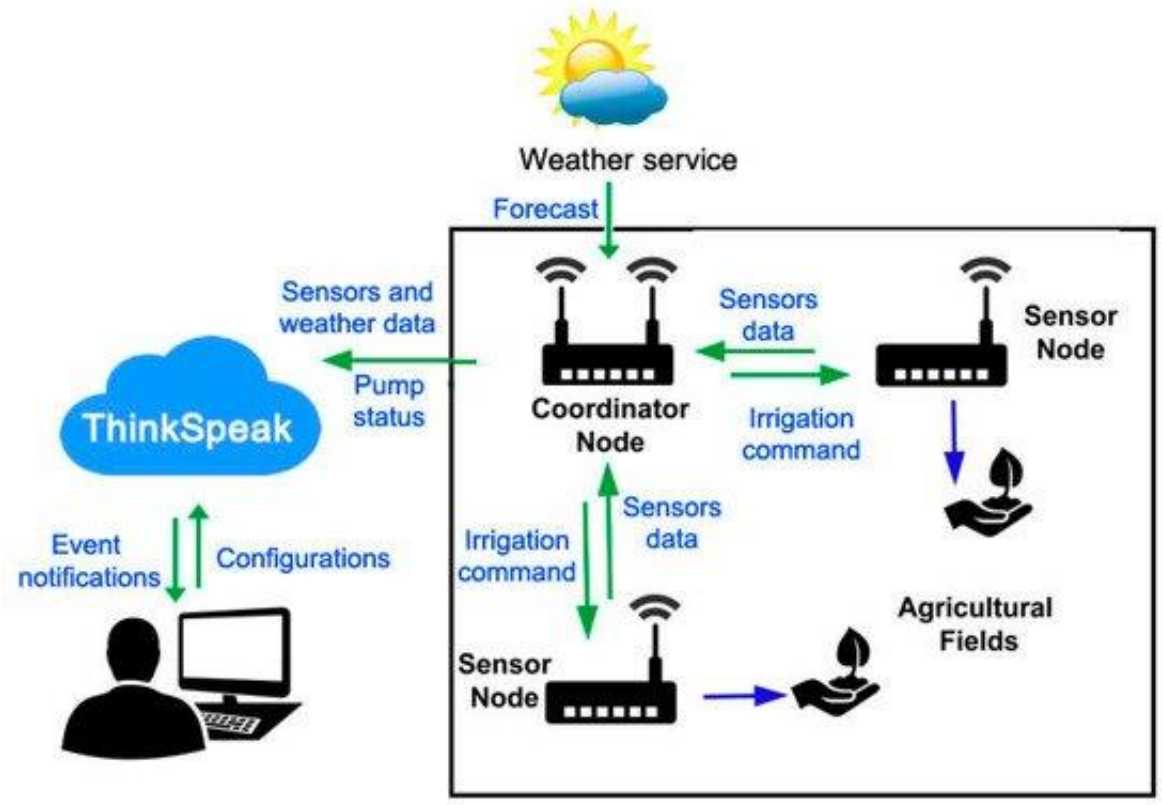
Department : I3 GTR

Project Analysis

❖ The Composition of the system

The system is composed mainly of three parts:

Cloud Platform, Coordinator Node, and Sensor Nodes …



Project Analysis

❖ Planning for first month

Week 1 6 Aug – 13 Aug	Week2 14 Aug – 20 Aug	Week3 21 Aug – 27 Aug	Week4 28 Aug – 3 Sep
Researching Code and test with dht11 Create charnel Thingspeak and testing.	To understand problem of code, Using Wi-Fi with username Combine code with coordinator and testing then send data to Thingspeak.	Collect data from Greenhouse send to Thingspeak And show all the result of Temperature (T1+T2) Or another Sensor on thingspeak.	Need to Complete planning of first month and continue working for second month to use Camera in greenhouse.
Missing: testing some error, problem to understand something in code	Not ready combine code with coordinator yet.		

Project Analysis

❖ Problems and solutions

ESP8266: Poor performance at hot place.

Code has corrected and Decide to combine with coordinator code.

Can’t make code to connect with username of WIFI yet.

Project Analysis

❖ Testing code with DHT11

esp8266_and_DHT11_sensor §

```
#include <SoftwareSerial.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define DHTPIN 13     // what pin we're connected to
#define DHTTYPE DHT22   // DHT 22  (AM2302)
#define RX 11 // set rx pin
#define TX 12 // set tx pin
String AP = "GTR_Lab";       // AP NAME
String PASS = "@gtrlab@"; // AP PASSWORD
String API = "88OOFGWCTW33Q0I6";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
String field1 = "field1";
String field2 = "field2";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial esp8266(TX,RX);

void setup() {
  Serial.begin(9600);
  esp8266.begin(9600);
  sendCommand("AT",5,"OK");  //send AT command
  sendCommand("AT+CWMODE=3",5,"OK"); // set MODE 1
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK"); // configure esp8266 to
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(5000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
```

Done compiling.

esp8266_and_DHT11_sensor §

```
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  String getData = "GET /update?api_key="+ API +"&field1="+String(t)+"&field2="+String(h);

  //String getData1 = "GET /update?api_key="+ API +"&"+ field1 +"="+String(t);// set string for send data to thing speak
  sendCommand("AT+CIPMUX=1",5,"OK"); // set to single connection
  sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK"); // to connect to the Thingspeak API using TCP protocol
  sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");// for read data and start sending data
  esp8266.println(getData); // send data to Thingspeak
  //delay(1000);
  countTrueCommand++;
  sendCommand("AT+CIPCLOSE=0",5,"OK");// for end and close transmission
  //delay(10000);

  //String getData2 = "GET /update?api_key="+ API +"&"+ field2 +"="+String(h);
  //sendCommand("AT+CIPMUX=1",5,"OK"); // set to single connection
  //sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK"); // to connect to the Thingspeak API using TCP protocol
  //sendCommand("AT+CIPSEND=0," +String(getData2.length()+4),4,">");// for read data and start sending data
  //esp8266.println(getData2);
  //delay(1500);
  //countTrueCommand++;
  //sendCommand("AT+CIPCLOSE=0",5,"OK");// for end and close transmission

//  Serial.print("Humidity: ");
//  Serial.print(h);
//  Serial.print(" %\t");
//  Serial.print("Temperature: ");
```

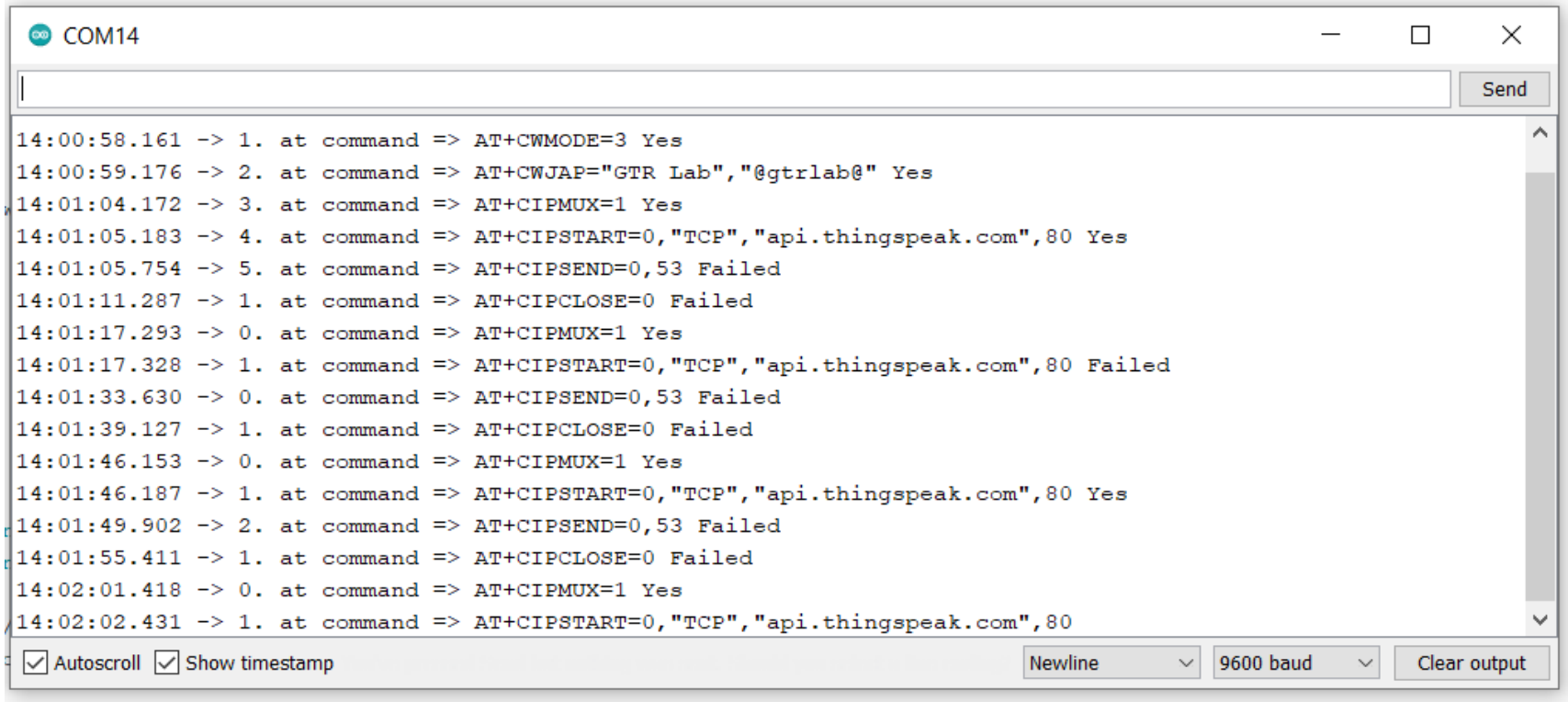
Done compiling.

Project Analysis

esp8266_and_DHT11_sensor §

```
70 //  Serial.print(t);
71 //  Serial.println(" .,*C ");
72 }
73
74 void sendCommand(String command, int maxTime, char readReplay[]) {
75   Serial.print(countTrueCommand);
76   Serial.print(". at command => ");
77   Serial.print(command);
78   Serial.print(" ");
79   while(countTimeCommand < (maxTime*1))
80   {
81     esp8266.println(command);//at+cipsend
82     if(esp8266.find(readReplay))//ok
83     {
84       found = true;
85       break;
86     }
87
88     countTimeCommand++;
89   }
90
91   if(found == true)
92   {
93     Serial.println("Yes");
94     countTrueCommand++;
95     countTimeCommand = 0;
96   }
97   if(found == false)
98   {
99     Serial.println("Failed");
100    countTrueCommand = 0;
101    countTimeCommand = 0;
102  }
103  found = false;
104 }
```

Done compiling.



Project Analysis

❖ Combine Code not ready yet

coordinator_with_ESP8266

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <XBee.h>
#include <SoftwareSerial.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#define DHTPIN 13     // what pin we're connected to
#define DHTTYPE DHT22   // DHT 22  (AM2302)
#define RX 11 // set rx pin
#define TX 12 // set tx pin
String AP = "GTR_Lab";       // AP NAME
String PASS = "@gtrlab@"; // AP PASSWORD
String API = "88OOFGWCTW33Q0I6";   // Write API KEY
String HOST = "api.thingspeak.com";
String PORT = "80";
int countTrueCommand;
int countTimeCommand;
boolean found = false;
DHT dht(DHTPIN, DHTTYPE);
SoftwareSerial esp8266(TX,RX);
//Define XBee
SoftwareSerial XBee(0,1);

//Define package structure
/* | header | node_id | humidity | temperature | moisture | rdiation |
   |   2    |    2    |     4    |      4      |     4    |     4    | */

typedef struct
{
  uint8_t header;
  uint8_t node_id;

  float humidity;
  float temperature;
```

Done compiling.

coordinator_with_ESP8266

```
  float temperature;
  float moisture;
  float radiation;
} data_struct_t;

typedef union
{
  data_struct_t data_struct;
  uint8_t data_byte[20];
} packet_t;

//Define LCD
LiquidCrystal_I2C lcd (0x3F, 20, 4);

//Define relay pin
#define relayPin 5

//Define mois1 and mois2, mois1 and 2 should more than 50 in order to prevent relay activate in case one node is not available
float mois1 = 50;
float mois2 = 50;
void setup() {
  XBee.begin(9600);
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();

  //Set relay pin
  pinMode(relayPin, OUTPUT);

  //Set title
  lcd.setCursor(4,1);
  lcd.print("NEVER GIVE UP");
  delay(1000);
  lcd.clear();
  esp8266.begin(9600);
```

Done compiling.

Project Analysis

coordinator_with_ESP8266

```
  lcd.clear();
  esp8266.begin(9600);
  sendCommand("AT",5,"OK");  //send AT command
  sendCommand("AT+CWMODE=3",5,"OK"); // set MODE 1
  sendCommand("AT+CWJAP=\""+ AP +"\",\""+ PASS +"\"",20,"OK"); // configure esp8266 to WiFi
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(5000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  String getData = "GET /update?api_key="+ API +"&field1="+String(t)+"&field2="+String(h);

  //String getData1 = "GET /update?api_key="+ API +"&"+ field1 +"="+String(t);// set string for send data to
  sendCommand("AT+CIPMUX=1",5,"OK"); // set to single connection
  sendCommand("AT+CIPSTART=0,\"TCP\",\""+ HOST +"\","+ PORT,15,"OK"); // to connect to the Thingspeak API us
  sendCommand("AT+CIPSEND=0," +String(getData.length()+4),4,">");// for read data and start sending data
  esp8266.println(getData); // send data to Thingspeak
  //delay(1000);
  countTrueCommand++;
  sendCommand("AT+CIPCLOSE=0",5,"OK");// for end and close transmission
  //delay(10000);
  //Check if XBee available
```

Done compiling.

coordinator_with_ESP8266

```
packet_t coord;
if (XBee.available() > 0)
{
  //Set seperate display
  lcd.setCursor(9,0);
  lcd.print("|");
  lcd.setCursor(9,1);
  lcd.print("|");
  lcd.setCursor(9,2);
  lcd.print("|");
  lcd.setCursor(9,3);
  lcd.print("|");

  coord.data_byte[0] = XBee.read();
  if (coord.data_struct.header == 0x55)
  {
    Serial.print(coord.data_byte[0], HEX);
    Serial.print(";");
    for (int i = 1; i <= 20; i++)
    {
      coord.data_byte[i] = XBee.read();
      Serial.print(coord.data_byte[i], HEX);
      Serial.print(";");
    }
  }

  //Print data from Node 1 on LCD
  if (coord.data_struct.node_id == 1)
  {
    lcd.setCursor(0,0);
    lcd.print("H1:");
    lcd.setCursor(3,0);
    lcd.print(coord.data_struct.humidity ,2);
    lcd.setCursor(0,1);
    lcd.print("T1:");
```

Done compiling.

Project Analysis

coordinator_with_ESP8266

```
137       lcd.print("T1:");
138       lcd.setCursor(3,1);
139       lcd.print(coord.data_struct.temperature, 2);
140       lcd.setCursor(0,2);
141       lcd.print("M1:");
142       lcd.setCursor(3,2);
143       lcd.print(coord.data_struct.moisture, 2);
144
145       mois1 = coord.data_struct.moisture;
146     }
147
148     //Print data from Node 2 on LCD
149     if (coord.data_struct.node_id == 2)
150     {
151       lcd.setCursor(11,0);
152       lcd.print("H2:");
153       lcd.setCursor(14,0);
154       lcd.print(coord.data_struct.humidity, 2);
155       lcd.setCursor(11,1);
156       lcd.print("T2:");
157       lcd.setCursor(14,1);
158       lcd.print(coord.data_struct.temperature, 2);
159       lcd.setCursor(11,2);
160       lcd.print("M2:");
161       lcd.setCursor(14,2);
162       lcd.print(coord.data_struct.moisture, 2);
163       lcd.setCursor(11,3);
164       lcd.print("R1:");
165       lcd.setCursor(14,3);
166       lcd.print(coord.data_struct.radiation, 1);
167
168       mois2 = coord.data_struct.moisture;
169     }
170
171     //Define when to start the Valve
```

Done compiling.

coordinator_with_ESP8266

```
171     //Define when to start the Valve
172     if (mois1 <= 32.00 or mois2 <= 32.00)
173     {
174       digitalWrite(relayPin, HIGH);
175     }
176
177     //Define when to stop the Valve
178     if (mois1 >= 34.00 and mois2 >= 34.00)
179     {
180       digitalWrite(relayPin, LOW);
181     }
182
183     /*
184     Serial.println("Mois1 is: ");
185     Serial.print(mois1);
186     Serial.println("Mois2 is: ");
187     Serial.print(mois2);
188     */
189   }
190 }
191
192 void sendCommand(String command, int maxTime, char readReplay[]) {
193   Serial.print(countTrueCommand);
194   Serial.print(". at command => ");
195   Serial.print(command);
196   Serial.print(" ");
197   while(countTimeCommand < (maxTime*1))
198   {
199     esp8266.println(command);//at+cipsend
200     if(esp8266.find(readReplay))//ok
201     {
202       found = true;
203       break;
204     }
205
```

Done compiling.

coordinator_with_ESP8266

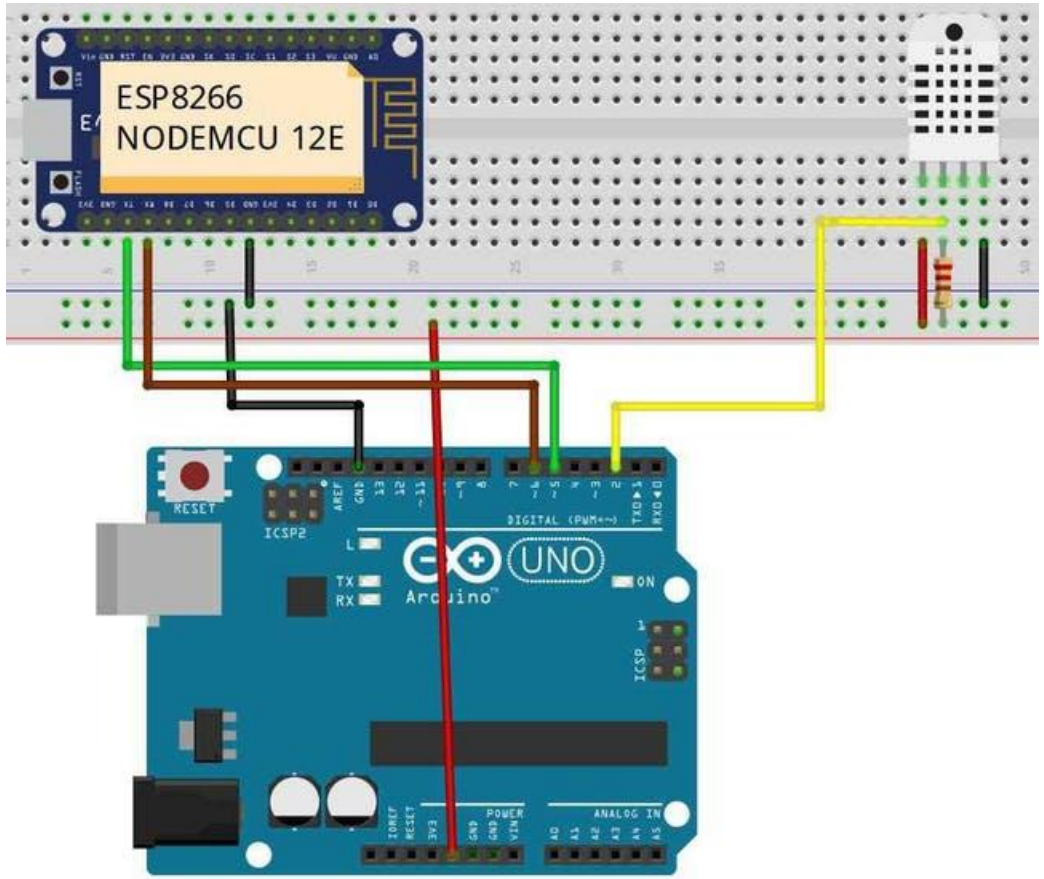
```
189   }
190 }
191
192 void sendCommand(String command, int maxTime, char readReplay[]) {
193   Serial.print(countTrueCommand);
194   Serial.print(". at command => ");
195   Serial.print(command);
196   Serial.print(" ");
197   while(countTimeCommand < (maxTime*1))
198   {
199     esp8266.println(command);//at+cipsend
200     if(esp8266.find(readReplay))//ok
201     {
202       found = true;
203       break;
204     }
205
206     countTimeCommand++;
207   }
208
209   if(found == true)
210   {
211     Serial.println("Yes");
212     countTrueCommand++;
213     countTimeCommand = 0;
214   }
215   if(found == false)
216   {
217     Serial.println("Failed");
218     countTrueCommand = 0;
219     countTimeCommand = 0;
220   }
221   found = false;
222 }
223
```

Done compiling.

Project Analysis

❖ Send Data from Arduino to ESP8266 NodeMCU

Code for Arduino:
Select Arduino Board and Arduino Port before uploading the code.




```
#include "DHT.h"
#include <SoftwareSerial.h>
#define DHTPIN 2
// Uncomment whatever type you're using!
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
SoftwareSerial espSerial(5, 6);
DHT dht(DHTPIN, DHTTYPE);
String str;
void setup(){
Serial.begin(115200);
espSerial.begin(115200);
dht.begin();
delay(2000);
}
void loop()
{
float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
Serial.print("H: ");
Serial.print(h);
Serial.print("% ");
Serial.print(" T: ");
Serial.print(t);
Serial.println("C");
str =String("coming from arduino: ")+String("H= ")+String(h)+String("T= ")+String
(t);
espSerial.println(str);
delay(1000);
}
```

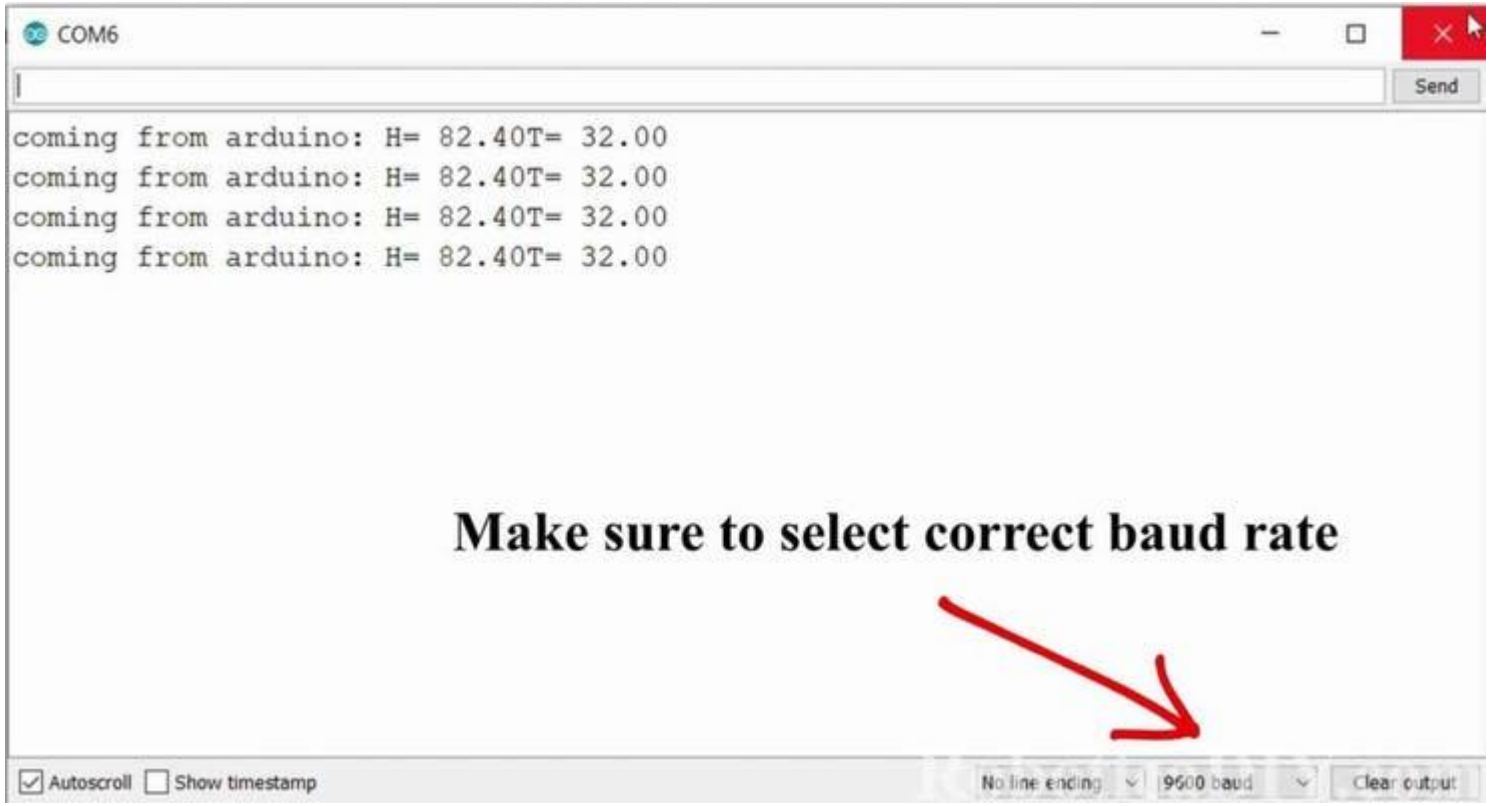
Project Analysis

Code for ESP8266:
Select NodeMCU 1.0 Board and ESP8266 Port before uploading the code.

```
void setup() {
// Open serial communications and wait for port to open:
Serial.begin(115200);
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
}
void loop() { // run over and over
if (Serial.available()) {
Serial.write(Serial.read());
}
}
```

Project Analysis

❖ Result



Thank You