

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

#define air 13

int sensor_pin = A0;

int output_value ;

void setup() {

  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only

  }

  pinMode (air, OUTPUT);

  Serial.println("Thingspeak Data Sender");

  // set the data rate for the SoftwareSerial port
  mySerial.begin(4800);

}

void loop() { // run over and over

  output_value = analogRead(sensor_pin);
```

```
Serial.print("Mositure : ");
```

```
Serial.print(output_value);
```

```
Serial.println("%");
```

```
delay(1000);
```

```
if (output_value >= 500 ) {
```

```
    Serial.println ("kering");
```

```
    digitalWrite (air, LOW);
```

```
}
```

```
else {
```

```
    Serial.println ("basah");
```

```
    digitalWrite (air, HIGH);
```

```
}
```

```
//String payload =
```

```
"AT+HTTTPARA=\"URL\", \"https://api.thingspeak.com/update?api_key=77HZS  
3IERJQV7LRR&field1=\" + String(val) + "\"";
```

```
String payload =
```

```
"AT+HTTTPARA=\"URL\", \"https://api.thingspeak.com/update?api_key=27CM1  
31U2UJMX4TV&field1=\" + String(output_value) + "\"";
```

```
Serial.println("AT+HTTTPARA=\"URL\", \"https://api.thingspeak.com/update?api_key=27CM131U2UJMX4TV&field1=\" + String(output_value) + "\");  
  
    sending(payload);  
  
    // for (int u = 0; u < payload.length(); u++) {  
    //   mySerial.write(payload.charAt(u));  
    // }  
  
    mySerial.println();  
  
    Serial.print("data yanbg dikirim : "); // + String(coba);  
  
    Serial.println(output_value); //  
  
    delay(2000);  
  
    mySerial.println("at+httpaction=0");  
  
    //Serial.println("httpaction");  
  
    while (mySerial.available() > 0) {  
        Serial.write(mySerial.read());  
    }  
  
    Serial.println();  
  
    delay(5000);  
  
    mySerial.println("at+httpread");  
  
    while (mySerial.available() > 0) {  
        Serial.write(mySerial.read());  
    }  
  
    delay(15000);
```

```
}
```

```
void sending(String payload) {  
    sendData("AT", "OK", 10000);  
    sendData("ATE0", "OK", 10000);  
    sendData("AT+SAPBR=0,1", "OK" , 10000);  
  
    sendData("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\"", "OK", 10000);  
    sendData("AT+SAPBR=3,1,\"APN\", \"3data\"", "OK", 10000);  
    sendData("AT+SAPBR=1,1", "OK" , 10000);  
    sendData("AT+SAPBR=2,1", "OK" , 10000);  
    sendData("AT+HTTPINIT", "OK" , 1000);  
    sendData("AT+HTTPSSL=1", "OK" , 10000);  
    sendData("AT+HTTTPARA=\"CID\",1", "OK" , 1000);  
    sendData(payload , "OK" , 10000);  
  
    sendData("AT+HTTPACTION=0", "+HTTPACTION:" , 2500000);  
    sendData("AT+HTTPREAD", "OK", 10000);  
    sendData("AT+HTTPTERM", "OK", 1000);  
  
}
```

```

void sendData(String command, String balasan, long int timeout )
{

    String response = "";
    mySerial.println(command); // send the read character to the gsm
//Serial.println(command);
    long int time = millis();
    bool rec = false;
    bool selesai = false;
    while ( (time + timeout) > millis() && selesai == false ) {
        if (mySerial.available()) {
            // The esp has data so display its output to the serial window
            char c = mySerial.read(); // read the next character.
            response += c;
            Serial.write(c);
        }
        int a = response.indexOf(balasan);
        if ( a > 0) {
            // Serial.print("response : " + response);
            Serial.println(" == == == == == == > ok");
            selesai = true;
        }
    }
}

```