```
#include <ThingSpeak.h>
#include <Wire.h>
#include <SimpleTimer.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

SimpleTimer timer;

int Data[] = {0, 0, 0, 0, 0, 0};//empty array where to put the numbers comming
from the slave
int Data2[] = {0};
int value=1;
String apiKey = "CWCPT6WS57BOWHCW";//  Enter your Write API key from ThingSpeak
String talkBackAPIKey2="8E41GA04PIKO6J29";
char talkBackAPIKey[] = "8E41GA04PIKO6J29"; // TalkBack API Key
char talkBackID[] = "26642"; // TalkBack ID

const char *ssid =  "openlab1";
const char *pass =  "phonglab1";
const char* server = "api.thingspeak.com";
 WiFiClient client;


void setup() {
  Wire.begin(D1, D2); /* join i2c bus with SDA=D1 and SCL=D2 of NodeMCU */
  Serial.begin(9600); /* begin serial for debug */

  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  startTimers();
  ThingSpeak.begin(client);
}

void loop() {

  Wire.beginTransmission (8);
  sendArduino();
  Wire.endTransmission ();
  //timer.run();
```

```
  envoiRequete();
  if (Wire.available())
  {
    for (int i = 0; i < 7; i++) {               // Receive the raw 'float'
data.
      int c = Wire.read();
      Data[i] = c;
    }
    AfficherData();
  }

  if (client.connect(server, 80))  //
  {
    sendDataThingspeak();

    if ( Data[6]==1){
    sendEchoPump();
    Data[6]=0;
  }

  }
  client.stop();
  checkTalkBack();
  Serial.println("Waiting...");
  delay(1000);
}


void startTimers(void)
{
  timer.setInterval(1000, envoiRequete);
}


void envoiRequete(void) {
  Wire.requestFrom(8, 7);
}

void AfficherData(void) {
  Serial.print(" Temp :\t");
  Serial.print(Data[0]);
  Serial.print(" Humidité air :\t");
  Serial.print(Data[1]);
  Serial.print(" Humidité sol: \t");
  Serial.print(Data[2]);
  Serial.print("Luminosité :\t");
  Serial.print(Data[3]);
```

```
  Serial.print(" Reserve eau : \t");
  Serial.print(Data[4]);
  Serial.print(" Pompe activée: \t");
  Serial.print(Data[5]);
  Serial.print(" Bouton activé \t");
   Serial.print(Data[6]);
  Serial.print("  \n");
}




void sendDataThingspeak(void) {
  String postStr = apiKey;
  postStr += "&field1=";
  postStr += String(Data[0]);
  postStr += "&field2=";
  postStr += String(Data[1]);
  postStr += "&field3=";
  postStr += String(Data[2]);
  postStr += "&field4=";
  postStr += String(Data[3]);
  postStr += "&field5=";
  postStr += String(Data[4]);
  postStr += "&field6=";
  postStr += String(Data[5]);
  postStr += "\r\n\r\n";

  client.print("POST /update HTTP/1.1\n");
  client.print("Host: api.thingspeak.com\n");
  client.print("Connection: close\n");
  client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
  client.print("Content-Type: application/x-www-form-urlencoded\n");
  client.print("Content-Length: ");
  client.print(postStr.length());
  client.print("\n\n");
  client.print(postStr);
}

void sendEchoPump(void) {
  String postStr = talkBackAPIKey2;
  Serial.print("------ECHO PUMP TERMINE-----:");

  client.print("POST https://api.thingspeak.com/talkbacks/26642/commands.json?
apikey=8E41GA04PIKO6J29&command_string=TURN_OFF&position=2 HTTP/1.1\n");
  client.print("Connection: close\n");
```

```
  client.print("\n\n");
}
void readLastDataField1(void)
{
int Reponse = ThingSpeak.readFloatField(508790, 7);
Serial.print("------Temp reçue-----:");
Serial.print(Reponse);
Data2[0]=Reponse;
Serial.println("------------:");
}



void sendArduino(void)
{
uint8_t Buffer2[4];
Buffer2[0] = Data2[0];
Wire.write(Buffer2, 1);
}


void checkTalkBack()
{
  char c;
  if (client.connect(server, 80)){
   //Serial.println("Connected to thingspeak");
    client.print("GET /talkbacks/");
    client.print(talkBackID);
    client.print("/commands/execute?api_key=");
    client.print(talkBackAPIKey);
    client.print(" HTTP/1.1\r\n");
    client.print("Host: api.thingspeak.com\r\n");
    client.println("Connection: close\r\n");
    }

    //Serial.println(F("Reading answer..."));
    String currentLine = "";
  while (client.connected()) {
    while (client.available()) {
      c = char(client.read());
      if (c == '\n'){
        //Serial.println(currentLine);
        currentLine = "";
        } else if (c != '\r') {
          currentLine +=c;
        }
    }
  }
```

```
  client.stop();

  Serial.println(currentLine);

  if (currentLine == "TURN_ON")
  {
    Data2[0]=1;
    Serial.println("pump on");
  }
  else
  {
    Data2[0]=0;
    Serial.println("pump off");
  }
}
```