

# Smart Irrigation Sytem using Think Speak IoT Platform

**Shyamal Kumar Roy<sup>1</sup>, Suman Ghosh<sup>2</sup>, Aveek Chattopadhyaya<sup>3</sup>,  
Madhumita Chakraborty<sup>4</sup>, Amit Debnath<sup>5</sup>**

*Guru Nanak Institute of Technology  
Shyamal.roy@gnit.ac.in*

## Abstract

This paper describes the smart irrigation system using the concept of IoT. The paper uses a wifi module (ESP8266) which connects the system to the internet. This module controls the motor for supplying water to the field on the information obtained from a soil moisture sensor. This whole system is monitored by Thing speak IoT Platform. The primary focus of this paper is to help the farmers and reduce their work and the module can be implemented in perennial plant irrigation and gardening land.

***Keywords: IoT, Think Speak IoT, ESP8266, Arduino Uno, and Irrigation***

## 1. Introduction

India is the country of village and agriculture plays an important role for development of country. In our country, agriculture depends on the monsoons which has insufficient source of water. So, the irrigation is used in agriculture field. In Irrigation system, depending upon the soil type, water is provided to plant. In agriculture, two things are very important, first to get information of about the fertility of soil and second to measure moisture content in soil. Nowadays, for irrigation, different techniques are available which are used to reduce the dependency of rain. And mostly this technique is driven by electrical power and on/off scheduling. In this technique, water level indicator placed in water reservoir and soil moisture sensors are placed root zone of plant and near the module and gateway unit handles the sensor information and transmit data to the controller which in turns the control the flow of water through the valves. For continuously increasing demand and decrease in supply of food necessities, it's important to rapid improvement in production of food technology. Agriculture is only the source to provide this. This is the important factor in human societies to growing and dynamic demand in food production. Agriculture plays the important role in the economy and development, like India. Due to lack of water and scarcity of land water result the decreasing volume of water on earth, the farmer use irrigation. Irrigation may be defined as the science of artificial application of water to the land or soil that means depending on the soil type, plant is to be provided with water.

## 2. Methodology and Implementation of IoT

The Internet of Things (IoT) is the network of physical objects devices, vehicles, buildings and other items embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer based systems, and resulting in improved efficiency, accuracy and economic benefit, when IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber physical systems, which also encompasses technologies such as smart grids, smart homes, intelligent transportation and smart cities. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. So, Internet of Things or IoT is an architecture that comprises specialized hardware boards, Software systems, web APIs, protocols which together creates a seamless environment which allows smart embedded devices to be connected to internet such that sensory data can be accessed and control system can be triggered over internet. Also devices could be connected to internet using various means like Wi-Fi, Ethernet and so on. Furthermore devices may not needed to be connected to internet independently. Rather a cluster of devices could be created (for example a sensor network) and the base station or the cluster head could be connected to internet. This leads to more abstract architecture for communication protocols which ranges from high level to low level. Most interestingly, these devices must be uniquely discovered. For unique discovery of the devices in a Network, they need to have unique IP address. IoT devices essentially have IPv6 addressing scheme. All these devices have either fixed or Subnet masked IP addresses of type v6. Unique IP addresses makes IoT devices discoverable in the internet as independent node. This is the most important concept to have in mind to understand IoT. Since IoT are essentially embedded systems and smart objects connected to internet with unique IP address which can be discovered and communicated over internet. We have also seen that the IoT devices may have external peripheral like Actuators and Sensors. Following figure.1 explain what IoT is all about.

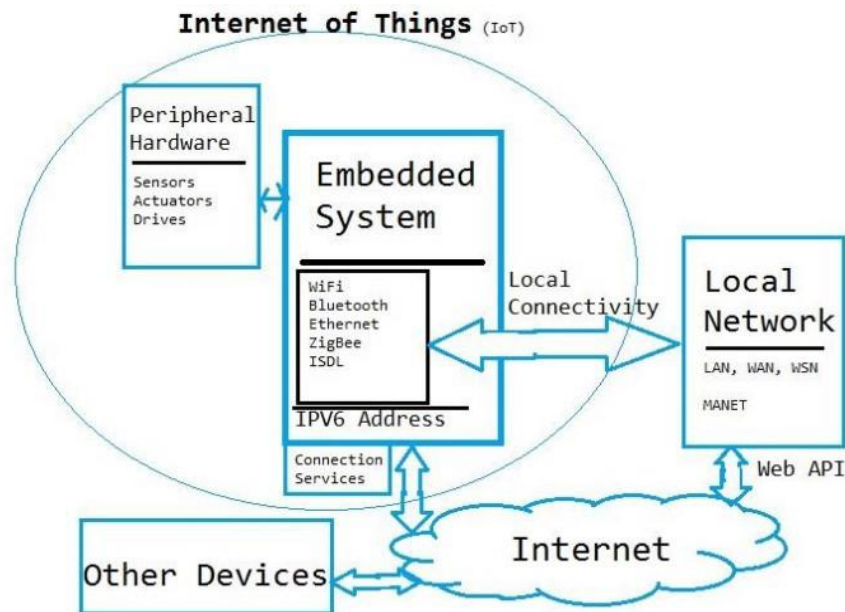


Figure 1. Internet of Things (IoT) basic Architecture

## 2.1 Introduction to ESP8266

ESP8266 is a complete and self-contained Wi-Fi network solutions that can carry software applications, or through another application processor uninstall all Wi-Fi networking capabilities. ESP8266 when the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, and the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference. The block diagram of ESP8266 illustrated in figure 2.

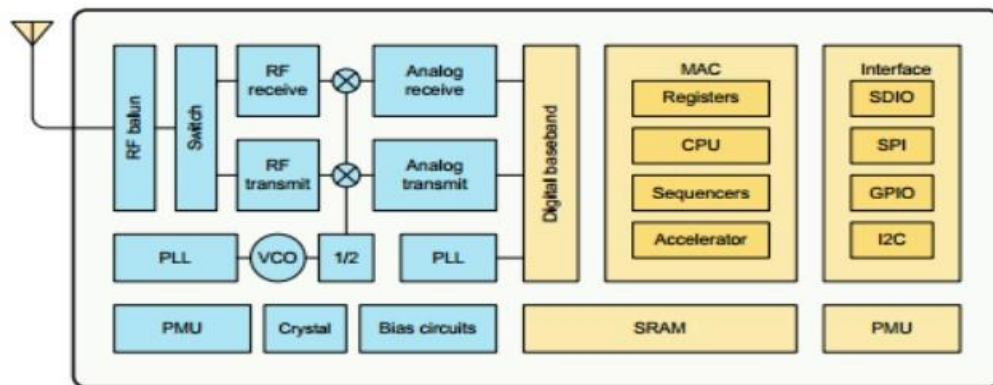


Figure 2. Block Diagram of ESP8266

### 3. Arduino Uno

The Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. Each of the 14 digital pins can be used as an input or output, using pin Mode(), digital Write(), and digital Read() functions. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller.

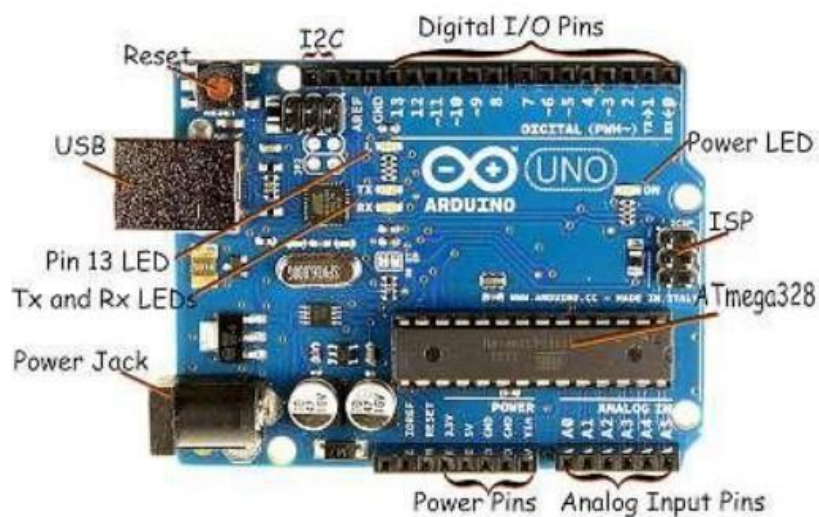


Figure 3. Arduino Uno

### 4. Schematic Diagram of Explore ESP8266 Wi-Fi Module

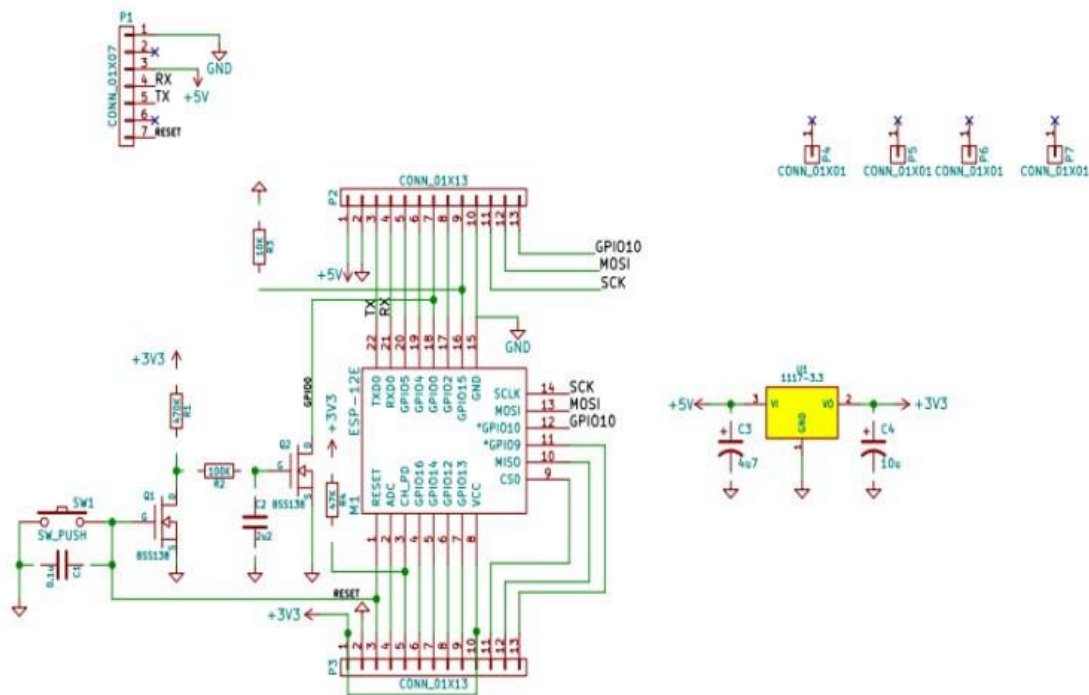


Figure 4. Schematic Diagram of Explore ESP8266 Wi-Fi Module

ESP8266, in its default configuration, boots up into the serial modem mode. In this mode you can communicate with it using a set of AT commands. AT commands are based on the Hayes Command Set. Index of all known AT commands is given in table 4.1

Basic	Wi-Fi layer	TCPIP Layer
AT	AT+CWMODE	AT+CIPSTATUS
AT+RST	AT+CWJAP	AT+CIPSTART
AT+GMR	AT+CWLAP	AT+CIPSEND
AT+GSLP	AT+CWQAP	AT+CIPCLOSE
ATE	AT+CWSAP	AT+CIFSR
	AT+CWLIF	AT+CIPMUX
	AT+CWDHCP	AT+CIPSERVER
	AT+CIPSTAMAC	AT+CIPMODE
	AT+CIPAPMAC	AT+CIPSTO
	AT+CIPSTA	AT+CIUPDATE
	AT+CIPAP	+IPD

#### 4.1. Software and programming tools

A program for Arduino may be written in any programming language with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their microcontrollers, AVR Studio and the newer Atmel Studio. The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. A program written with the IDE for Arduino is called a sketch. Sketches are saved on the development computer as text files with the file extension. ino. Arduino Software (IDE) pre-1.0 saved sketches with the extension. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main () into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program argued to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader


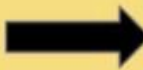

program in the board's firmware. The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

### 4.2. Think Speak

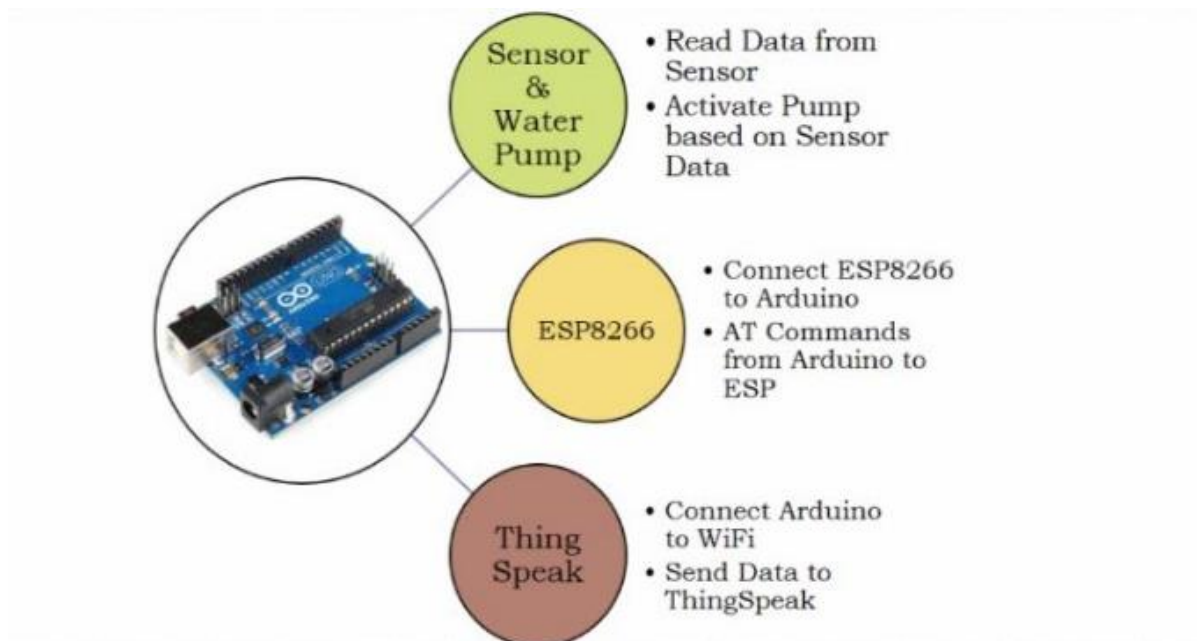
According to its developers, "Thing Speak is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. Thing Speak enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates". Thing Speak was originally launched by io Bridge in 2010 as a service in support of IoT applications. Thing Speak has integrated support from the numerical computing software MATLAB from MathWorks, allowing Thing Speak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a MATLAB license from Math works. Thing Speak has a close relationship with MathWorks, Inc. In fact, all of the Thing Speak documentation is incorporated into the MathWorks' Matlab documentation site and even enabling registered MathWorks user accounts as valid login credentials on the Thing Speak website. The terms of service and privacy policy of ThingSpeak.com are between the agreeing user and MathWorks, Inc. Thing Speak has been the subject of articles in specialized "Maker" websites like Instructible, Code project, and Channel 9.

### 4.2. Working

First all the connections are made between the ARDUINO UNO board, ESP8266 WIFI module, L293D motor driver, YL38 soil sensor and the motor as per the connections given below:

Component 1	Pin	Pin Description		Pin	Pin Description	Component 2
Arduino Uno	A0	Analog Read		A0	Analog Data	YL - 38 Soil Moisture Sensor Module
	5V	Vcc		Vcc	-	
	GND	Ground		GND	Ground	
Arduino Uno	8	Digital I/O		A2	Input 2	L293D Motor Driver connected to Water Pump
	GND	Ground		A1	Input 1	
	5V	Vcc		ENA	Enable	
	GND	Ground		GND	Ground	
Arduino Uno	0 (Rx)	Receiver		Tx	Transmitter	ESP8266
	1 (Tx)	Transmitter		Rx	Receiver	
	3V3	3.3 V		Vcc	-	

### 4.3. Programing Logic



- We read data from the sensor using the function 'analogRead (pin)'.
- After that using the map function, we enter the range of the sensor and the data.
- We connect the ESP8266 Wi-Fi module to Arduino Uno board and create an instance of an object for serial communication using the function 'SoftwareSerial(rxpin,txpin)'.
- We then send AT commands to ESP8266 Wi-Fi module to check for available networks and access points using the function 'sendAT'.
- Then using the function 'connectwifi' we connect to a local hotspot.
- Then using the 'updateTS' function we send data to ThingSpeak



### 4.3. Code

The code given below was first typed in the Arduino compiler and then uploaded to the ARDUINO UNO board.

```
#include <SoftwareSerial.h> //include the software serial library  int sensor; //variable to store
sensor values
int data; //variable to stored mapped sensor values
int pump_status; //variable to store pump status values
SoftwareSerial esp8266(3, 4); //set the software serial pins RX pin = 3, TX pin = 4
//definition of variables
#define DEBUG true //show messages between ESP8266 and Arduino in serial port, when set
to true
#define SSID "ASR" //replace x with your wifi network name
#define PASS "Asr16896" //replace x with your wifi network password

String sendAT(String command, const int timeout, boolean debug)
{
String response = ""; esp8266.print(command); long int time = millis();
while ((time + timeout) > millis())
{
while (esp8266.availa
{
char c = esp8266.read(); response += c;
}
}
if (debug)
{
Serial.print(response);
}
return response;
}

void setup()
{
Serial.begin(9600);// begin the serial communication with baud of 9600  esp8266.begin(9600);//
begin the software serial communication with baud rate 9600

sendAT("AT+RST\r\n", 2000, DEBUG); // call sendAT function to send reset AT command
sendAT("AT\r\n", 1000, DEBUG);
    sendAT("AT+CWMODE=1\r\n", 1000, DEBUG); //call sendAT function to set ESP8266 to
station mode
    sendAT("AT+CWJAP=\\\"SSID\\\",\\\"PASS\\\"\r\n", 2000, DEBUG); //AT command to
connect with the wifi network
while(!esp8266.find("OK")) { //wait for connection
```

```

}
sendAT("AT+CIFSR\r\n", 1000, DEBUG); //AT command to print IP address on serial monitor
sendAT("AT+CIPMUX=0\r\n", 1000, DEBUG); //AT command to set ESP8266 to multiple
connections

}

void loop(){
// put your main code here, to run repeatedly:
sensor=analogRead(A0); //read raw sensor data and store it in variable sensor
data=map(sensor,0,1023,100,0); //map the raw sensor data and store the result in data variable
String sensor_value=String(data); //convert integer to string data type
Serial.print("Soil Moisture: "); //print soil moisture on serial monitor Serial.println(data); //print
soil moisture value on serial monitor
if(data<40){ //check if sensor value is less than 76 digitalWrite(8,HIGH); //switch on the water
pump pump_status=100; //update pump status variable value to 100
}
else{
digitalWrite(8,LOW); //switch off the water pump pump_status=0; //update pump status
variable value to 0
}

String pump = String(pump_status); //convert integer to string data type
updateTS(sensor_value, pump); //call the function to update ThingSpeak channel delay(3000);
}

void updateTS(String T,String P){ Serial.println("");
sendAT("AT+CIPSTART=\"TCP\", \"api.thingspeak.com\", 80\r\n", 1000, DEBUG);
delay(2000);
String cmdlen;
String cmd="GET
/update?key=BN6SB18BU9OY51U5&field1="+T+"&field2="+P+"\r\n"; // update the
temperature and humidity values on ThingSpeak URL, replace xxxxxxxx with your write api
key
cmdlen = cmd.length();

```

```

sendAT("AT+CIPSEND="+cmdlen+"\r\n", 2000, DEBUG);
esp8266.print(cmd); Serial.println("");
sendAT("AT+CIPCLOSE\r\n", 2000, DEBUG);
Serial.println(""); delay(150);
}

```

After uploading the code to the Arduino Uno board we create an account on ThingSpeak API and give the API keys to identify the calling program. Then we run the program using our preferred range of sensor and observations were taken.

The screenshot shows the ThingSpeak website interface for a channel named "smart irrigation using IOT". The channel ID is 395793, the author is rin2ray, and the access is public. The page is divided into several sections:

- Write API Key:** A text input field contains the key "BN6SB18BU90Y51U5". Below it is an orange button labeled "Generate New Write API Key".
- Read API Keys:** A text input field contains the key "JJL1VEPSKPFUXR21". Below it is a text area labeled "Note".
- Help:** A section explaining that API keys enable writing data to a channel or reading data from a private channel. It notes that API keys are auto-generated when creating a new channel.
- API Keys Settings:** A list of instructions:
  - Write API Key:** Use this key to write data to a channel. If compromised, click "Generate New Write API Key".
  - Read API Keys:** Use this key to allow other people to view private channel feeds and charts. Click "Generate New Read API Key" for additional keys.
  - Note:** Use this field to enter information about channel read keys. Add notes to keep track of users with access.
- API Requests:** A section titled "Update a Channel Feed" showing a code snippet:
 

```
GET https://api.thingspeak.com/update?api_key=BN6SB18BU90Y51U5&field
```

Figure 5. Smart Irrigation Using IoT Thinkspeak

The soil and the pump status could be seen in the ThingSpeak account that we made earlier.

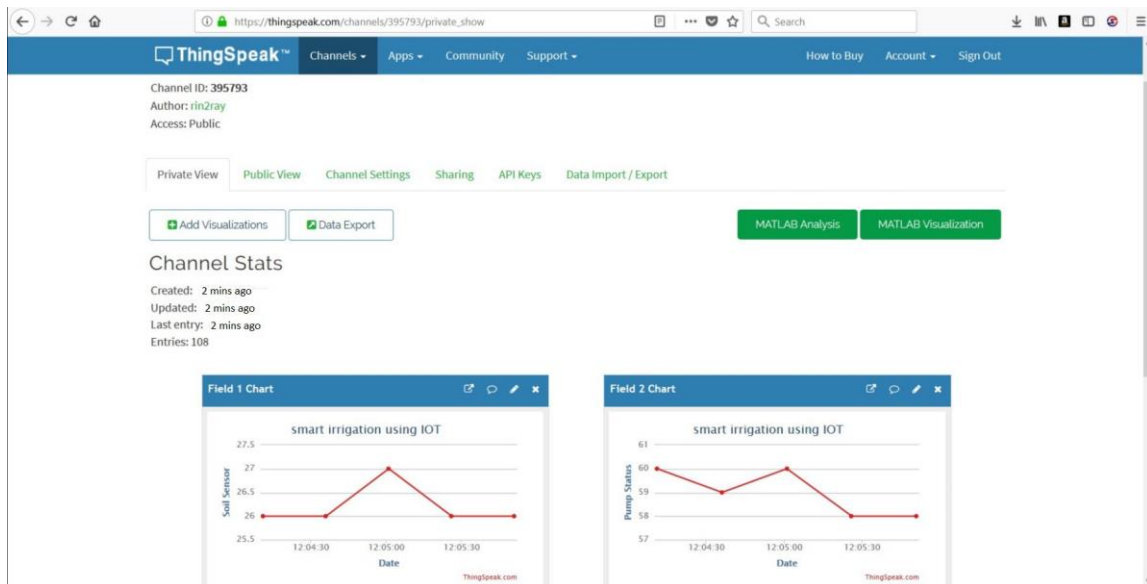


Figure 6. Smart Irrigation Using IoT Thingspeak Results

The soil moisture sensor sensed that the soil was dry and so it gave data to the pump to switch on and supply water till our defined range.

## 5. Conclusion

The agriculture field is being monitored by ThingSpeak API. The ESP8266 is the device at field end which receives the messages from ThingSpeak API network and manipulates it and will perform the function mentioned in message. After it will send the messages to ThingSpeak API network and in turn it will be published to the Client (user end). The ESP8266 is the best device for IoT projects. Since it is small, compact, lightweight, easily programmable, and easily installable and have enough GPIO pins to use them. Thus, the smart irrigation system is feasible and cost effective for optimizing the water resources for agricultural production. This type of irrigation system allows cultivation in places with water scarcity thereby improving sustainability.

## References

- [1] Mozdalifah Ali, Guo Ling, Hisham Elmouazen, "Design and Implementation of an Embedded System for Water Quality Monitoring (WQM) Based on Internet of Things (IOT)", 2023 7th International Conference on Robotics, Control and Automation (ICRCA), pp.21-25, 2023
- [2] A.Jaya Lakshmi, Ratnasunder Dasari, Manikanta Chilukuri, Yashwanth Tirumani, Hiral Dwaraka Praveena, A.Pramod Kumar, "Design and Implementation of a Smart Electric

- Fence Built on Solar with an Automatic Irrigation System", 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp.1553-1558, 2023.*
- [3] Prathibha S R, Anupama Hongal, Jyothi M P "IOT Based Monitoring System In Smart Agriculture 2017 International Conference on Recent Advances in Electronics and communication Technology 978-1-5090-6701-5/17 \$31.00 © 2017 IEEEDOI10.1109/ICRAECT.2017.5281
- [4] Ram Krishna Jha, Santosh Kumar, Kireet Joshi, Rajneesh Pandey "Field Monitoring Using IoT in Agriculture" 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)
- [5] Abdul Salam and Syed Shah Internet of Things in Smart Agriculture: Enabling Technologies 2019 IEEE 5th World Forum on the Internet of Things (WF-IoT)
- [6] I. U. Siloko, K. E. Ukhurebor, E. A. Siloko et al., "Effects of some meteorological variables on cassava production in Edo state, south-south, Nigeria via density estimation," *Scientific African*, vol. 13, Article ID e00852, 2021.
- [7] K. E. Ukhurebor, C. O. Adetunji, O. T. Olugbemi et al., "Precision agriculture: weather forecasting for future farming," *AI, Edge and IoT-Based Smart Agriculture*, Elsevier, Amsterdam, Netherlands, pp. 101–121, 2022.