

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY



DESIGN AND ANALYSIS OF ALGORITHMS BACHELOR OF TECHNOLOGY (B.TECH)

PROJECT IRRIGATION SESSION: 2022-23

COMPUTER SCIENCE AND ENGINEERING

MADE BY:
ROHIT KUMAR SHAW (2021UCS1523)
VEENIT KUMAR (2021UCS1548)
ADITYA GIRI (2021UCS1519)
CHIRAG SAINI (2021UCS1564)

UNDER SUPERVISION OF: DR. GAURAV SINGHAL

PROJECT IRRIGATION

IOT (INTERNET OF THINGS)

IOT is a term that refers to the connection of objects to each other and to humans through the internet.

IOT is a simple concept of basically controlling different devices by establishing a connection and communication from mobile apps or web browsers.

Taking everyday things and embedding them with electronics, software and then connecting them to collect and exchange data without human intervention is called the IOT.

IOT CONCEPTUAL FRAMEWORK AND ARCHITECTURAL FRAMEWORK

Following Equation describes a simple conceptual framework of IOT.

$$[Physical\ object + (Controller, Sensor, Actuator) + Internet = IOT]$$

Actuator:- These are supporting devices which help other devices to work or we can say actuator is a device that makes something move or work.

IOT ARCHITECTURE BY ORACLE

$$Gather + Enrich + Stream + Manage + Acquire + Organize and Analyze = IOT$$

Complete IOT conceptual framework for cloud based process and services:-

$$Gather + Consolidate + connect + collect + assemble + manage and analyze = IOT$$

PROJECT IRRIGATION

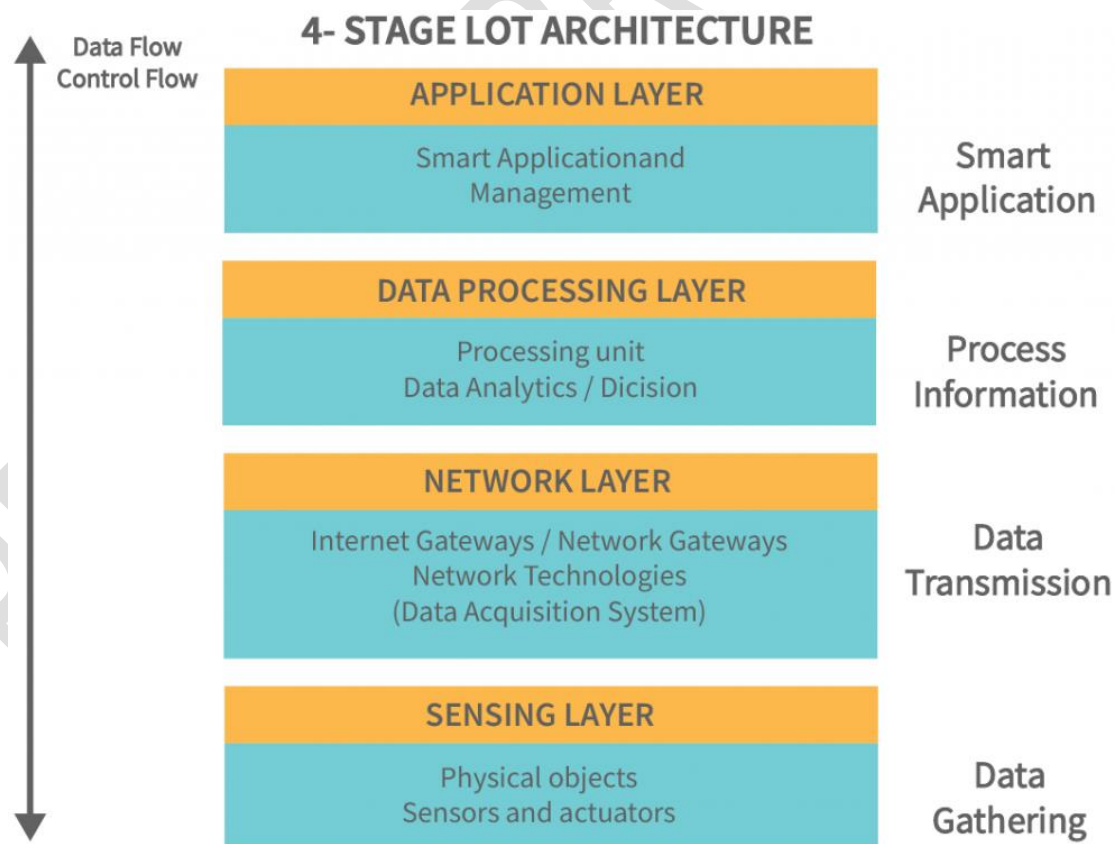
IOT ARCHITECTURE

4 layers of IOT Architecture:-

Layer 1:- Sensing Layer :- This layer is made up of physical objects integrated with sensors (Smart devices) and Actuators. These sensors and Actuators accept data from the atmosphere or place processes data and emits/shares it through IOT gateway.

Layer 2:- Network Layer :- In this layer, Internet/ Network gateways, Data Acquisition System are present. DAS performs data aggregation, collection and conversion functions.

Gateway acts as a carrier between the internet network of sensor nodes with the internet. Gateway also performs many functions like malware protection, filtering, data management, etc.



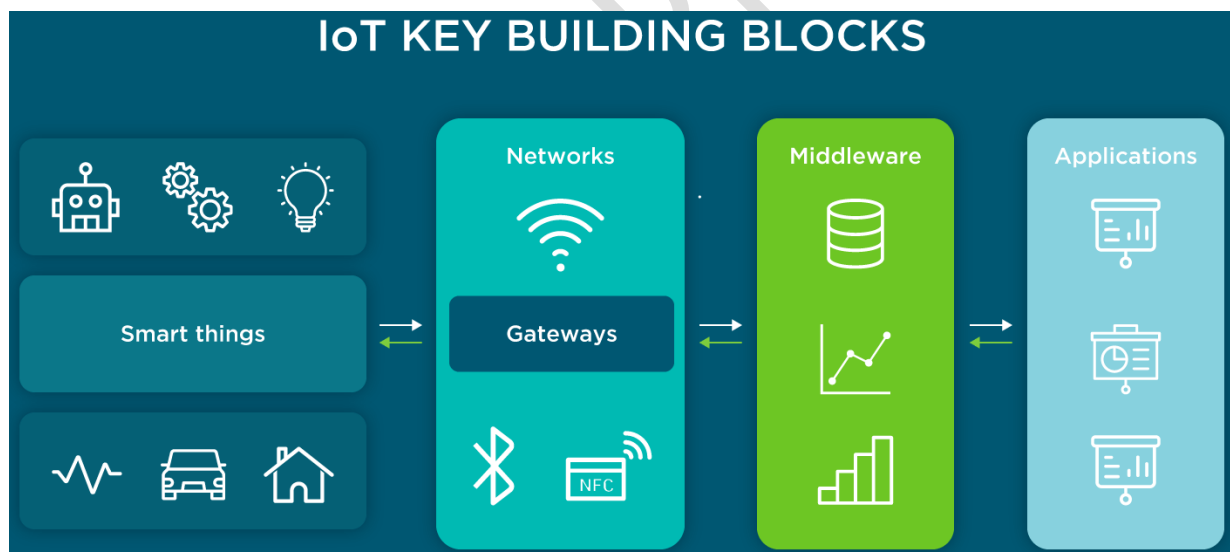
PROJECT IRRIGATION

Layer 3:- Data Processing Layer:- The data transmitted through the gateway is stored and processed securely with the cloud servers from where data is accessed by software application.

The processed data is used to perform intelligent actions that make all our devices smart devices.

Layer 4:- Application layer:- End user Application or mobile apps will help the end-user to control and monitor their device from a remote location. This app takes important information from the cloud and displays it on your smartphone tables. The main tasks here are visualization and management of important information.

With the help of these apps, the user sends commands to sensors to perform some action like changing the temp. Of AC.



PROJECT IRRIGATION

COMPONENTS REQUIRED

Components Used:

1. Arduino Uno R3
2. ESP 8266 WIFI module
3. Solenoid Valve
4. 5V Relay

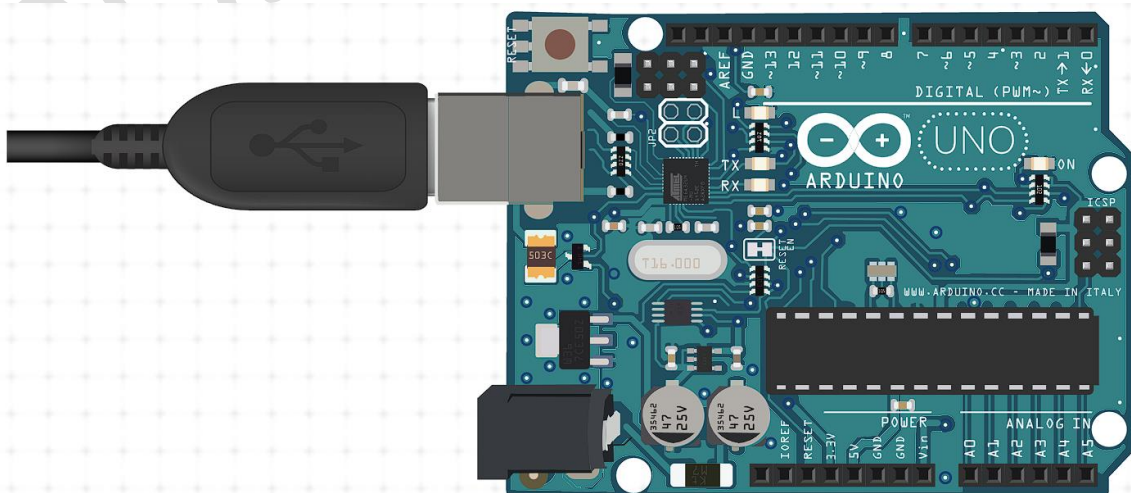
I) ARDUINO UNO R3

Arduino 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 ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller.

It can be simply connected to a computer with a USB cable or powered with an AC-to-DC adapter or battery.

The Arduino Uno R3 can be used to develop applications that operate in a standalone or connected environment.

The device is programmed using the Arduino integrated development environment (IDE).



MADE BY: ROHIT KUMAR SHAW (2021UCS1523) VEENIT KUMAR (2021UCS1548)
ADITYA GIRI (2021UCS1519) CHIRAG SAINI (2021UCS1564)

PROJECT IRRIGATION

ARDUINO UNO R3 SPECIFICATIONS

The Arduino Uno R3 board includes the following specifications.

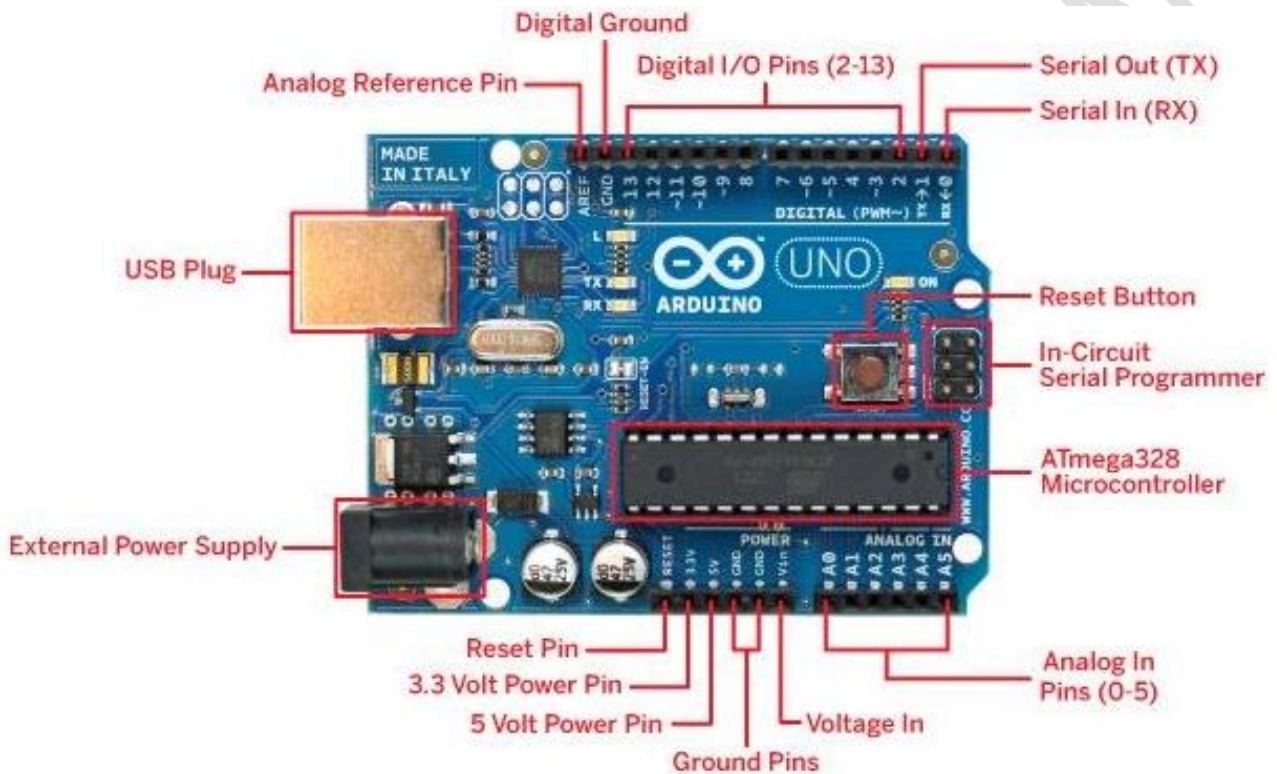
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

PROJECT IRRIGATION

RECOMMENDED OPERATING CONDITIONS

Description	Min	Max
Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C (185°F)

ARDUINO UNO R3 PIN DIAGRAM



POWER SUPPLY

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) mainly includes a battery or an AC to DC adapter.

The connection of an adapter can be done by plugging a centre-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of Vin as well as GND.

PROJECT IRRIGATION

The power pins of an Arduino board include the following.

1) Vin

The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else RPS (regulated power supply). By using this pin, one can supply the voltage.

2) 5 Volts

The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

3) 3V3

A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.

4) GND (ground pins)

MEMORY

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB.

INPUT AND OUTPUT

Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write().

These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm pull up resistor. The maximum current

PROJECT IRRIGATION

on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

SERIAL PINS

The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

EXTERNAL INTERRUPT PINS

The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

PWM PINS

The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write().

SPI (SERIAL PERIPHERAL INTERFACE) PINS

The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the SPI communication with the help of the SPI library.

LED PIN

An arguing board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

TWI (2-WIRE INTERFACE) PINS

The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

PROJECT IRRIGATION

AREF (ANALOG REFERENCE) PIN

An analog reference pin is the reference voltage to the inputs of an analog i/ps using the function like analog Reference().

RESET (RST) PIN

This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

COMMUNICATION

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication.

UART

An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

I2C

An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.

SPI Pins

The SPI communication includes MOSI, MISO, and SCK.

MOSI (Pin 11)

This is the master out slave in the pin, used to transmit the data to the devices.

PROJECT IRRIGATION

MISO (PIN 12)

This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.

SCK (PIN 13)

The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming ATmega microcontroller directly with the boot loader.

II) ESP 8266 WIFI Module

ESP8266 is a very low cost & user-friendly WIFI module, which develops a simple TCP/IP connection and can easily be interfaced with microcontrollers via Serial Port.

It can work as either a slave or a standalone application. If the ESP8266 Wi-Fi runs as a slave to a microcontroller host, then it can be used as a Wi-Fi adaptor to any type of microcontroller using UART or SPI. If the module is used as a standalone application, then it provides the functions of the microcontroller and Wi-Fi network. A set of AT commands are needed by the microcontroller to communicate with the ESP8266 Wi-Fi module.

ESP8266 WI-FI MODULE SPECIFICATIONS

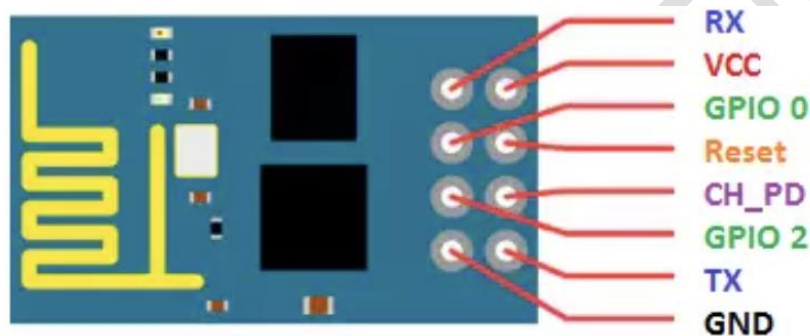
- It is a powerful Wi-Fi module available in a compact size at a very low price.
- It is based on the L106 RISC 32-bit microprocessor core and runs at 80 MHz
- It requires only 3.3 Volts power supply.
- The current consumption is 100 m Amps.
- The maximum Input/Output (I/O) voltage is 3.6 Volts.
- It consumes 100 mA current.

PROJECT IRRIGATION

- The maximum Input/Output source current is 12 mA.
- The frequency of built-in low power 32-bit MCU is 80 MHz.
- The size of flash memory is 513 kb.
- It is a 2.4 GHz Wi-Fi module and supports WPA/WPA2, WEP authentication, and open networks.

ESP8266 PINOUT

ESP8266 Pinout consists of 8 pins in total:



ESP8266 Pinout		
No.	Pin Name	Working
1	RX	Serial Receiver Pin
2	Vcc	Power Pin (+3.3 V; can handle up to 3.6 V)
3	GPIO 0	General-purpose I/O No. 0
4	RST	Reset
5	CH_PD	Chip power-down
6	GPIO 2	General-purpose I/O No. 2
7	TX	Serial Transmitter Pin
8	GND	Ground

MADE BY: ROHIT KUMAR SHAW (2021UCS1523) VEENIT KUMAR (2021UCS1548)
ADITYA GIRI (2021UCS1519) CHIRAG SAINI (2021UCS1564)

PROJECT IRRIGATION

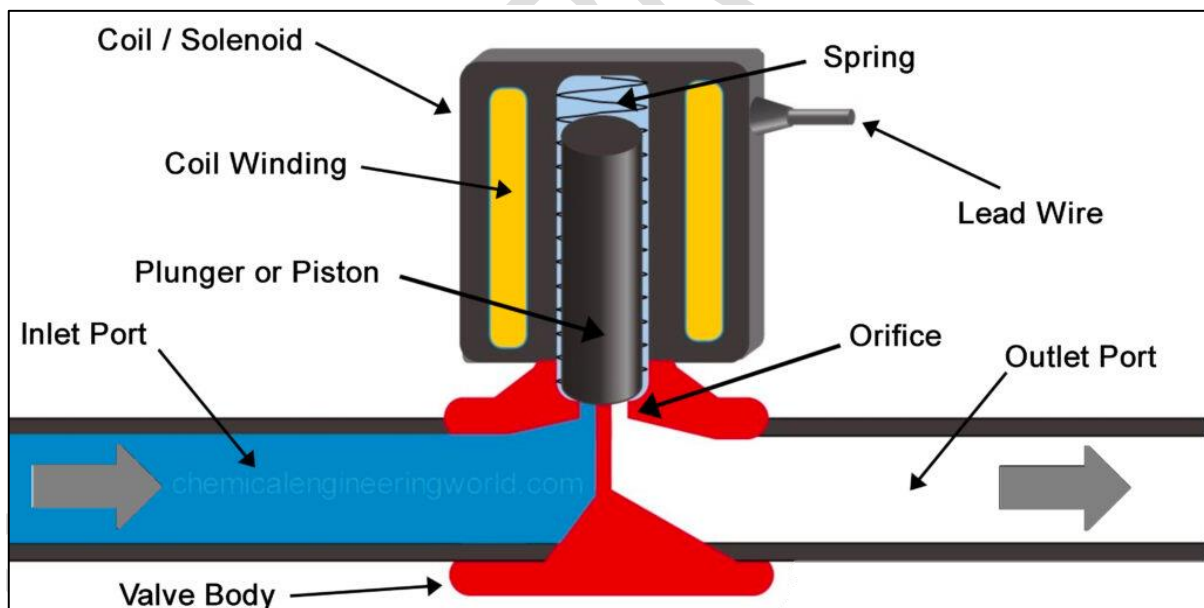
Each pin comes with a specific function associated with it where Vcc and GND are voltage source and ground respectively.

RX and TX are used for communication where TX is dedicated for data transmission and RX is used for receiving data.

SOLENOID VALVE

A solenoid valve is an electrically controlled valve. The valve features a solenoid, which is an electric coil with a movable ferromagnetic core (plunger) in its centre.

In the rest position, the plunger closes off a small orifice. An electric current through the coil creates a magnetic field. The magnetic field exerts an upwards force on the plunger opening the orifice. This is the basic principle that is used to open and close solenoid valves.



WHY IS A SOLENOID VALVE USED?

In most flow control applications, it is necessary to start or stop the flow in the circuit to control the fluids in the system. An electronically operated solenoid valve is usually used for this purpose. By being solenoid actuated,

PROJECT IRRIGATION

solenoid valves can be positioned in remote locations and may be conveniently controlled by simple electrical switches.

III) 5V RELAY

A 5v relay is an automatic switch that is commonly used in an automatic control circuit and to control a high-current using a low-current signal. The input voltage of the relay signal ranges from 0 to 5V. Relay is one kind of electro-mechanical component that functions as a switch. The relay coil is energized by DC so that contact switches can be opened or closed. A single channel 5V relay module generally includes a coil, and two contacts like normally open (NO) and normally closed (NC).

5V RELAY MODULE PIN CONFIGURATION

The pin configuration of the 5V relay module is shown below. This module includes 6-pins where each pin and its functionality are discussed below.



Normally Open (NO): This pin is normally open unless we provide a signal to the relay module's signal pin. So, the common contact pin smashes its link through the NC pin to make a connection through the NO pin.

Common Contact: This pin is used to connect through the load that we desire to switch by using the module.

PROJECT IRRIGATION

Normally Closed (NC): This NC pin is connected through the COM pin to form a closed circuit. However, this NC connection will break once the relay is switched through providing an active high/low signal toward the signal pin from a microcontroller.

Signal Pin: The signal pin is mainly used for controlling the relay. This pin works in two cases like active low otherwise active high. So, in active low case, the relay activates once we provide an active low signal toward the signal pin, whereas, in an active high case, the relay will trigger once we provide a high signal toward the signal pin.

However, these modules generally work on an active high signal which will strengthen the relay coil to make contact with the common terminal with the normally open terminal.

5V VCC: This pin needs 5V DC to work. So 5V DC power supply is provided to this pin.

Ground: This pin connects the GND terminal of the power supply.

PROJECT IRRIGATION

HOW IT WORKS

REMOTEXY

RemoteXY is easy way to make and use a mobile graphical user interface for controller boards to control via smartphone or tablet. The system includes:

- Editor of mobile graphical interfaces for controller boards, located on the site remotexy.com
- Mobile app RemoteXY that allows to connect to the controller and control it via graphical interface.

Distinctive features:

- The interface structure is stored in the controller.
- When connected, there is no interaction with servers to download the interface.
- The interface structure is downloaded to the mobile application from the controller.
- One mobile application can manage all your devices. The number of devices is not limited.

Supported boards:

- Arduino UNO, MEGA, Leonardo, Pro Mini, Nano, MICRO and compatible AVR boards
- ESP8266 boards
- ESP32 boards
- ChipKIT UNO32, ChipKIT uC32, ChipKIT Max32

Supported communication modules:

- Bluetooth HC-05, HC-06 or compatible
- Bluetooth BLE HM-10 or compatible
- ESP8266 as modem
- Ethernet W5100

PROJECT IRRIGATION

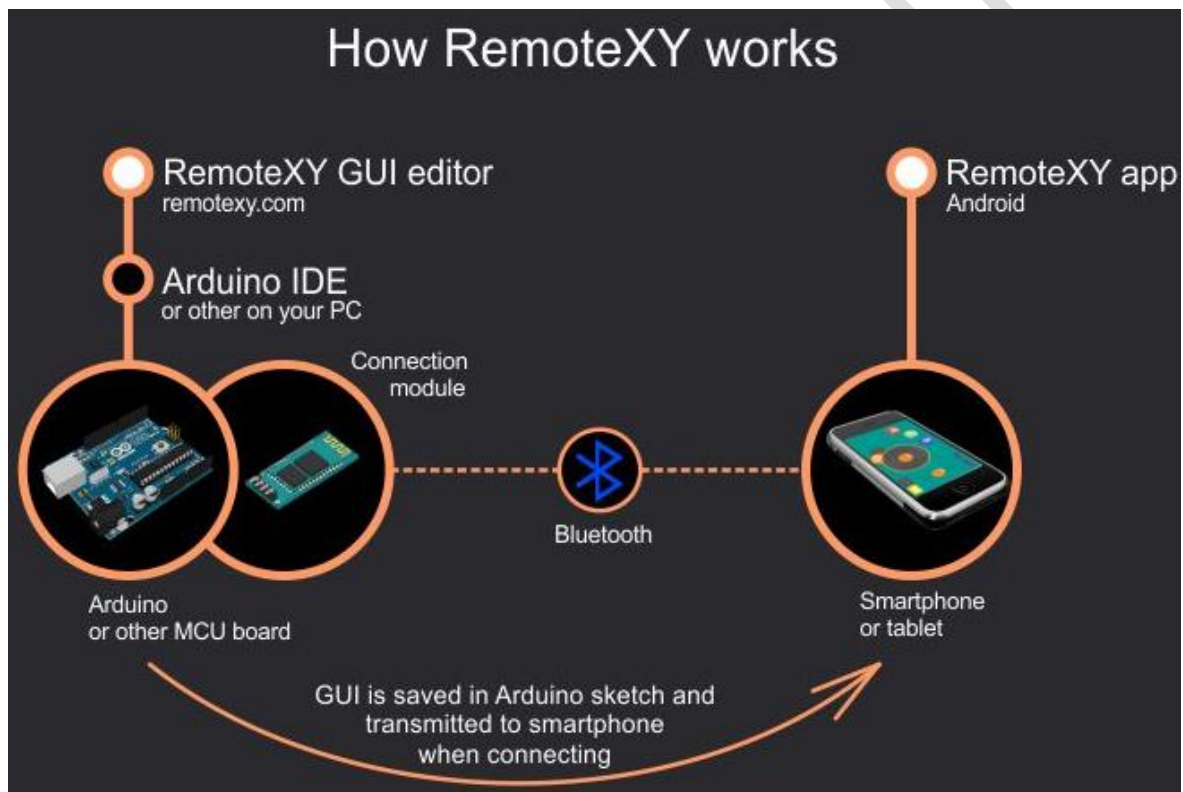
Supported IDE:

- Arduino IDE
- FLProg IDE
- MPIDE

Supported mobile OS:

- Android
- iOS

RemoteXY is easy way to make a unique graphical interface to control microcontroller device via mobile application, Arduino for example.



RemoteXY allows:

- 1) To develop any graphical management interface, using the control, display and decoration elements any combination thereof. You can develop the graphical interface for any task, placing the elements on the screen using the online editor.

PROJECT IRRIGATION

- 2) After the development of the graphical interface, you get the source code for the microcontroller that implements your interface. The source code provides a structure for interaction between your program with the controls and display. Thus you can easily integrate the control system into your task for which you are developing the device.
- 3) To manage microcontroller device using your smartphone or tablet with the graphical interface. For manage used mobile application RemoteXY.
- 4) Using one mobile application, you can manage a large number of devices with different graphical management interfaces. As the interface description is stored on board the microcontroller device.

ESP8266 WiFi MODULE

ESP8266 module allows to use RemoteXY with microcontroller device for Wi-Fi. Implemented support for all Arduino boards. The source code of GUI can be download for the Arduino IDE and PLProg.

To use the ESP8266 module to select this module in the properties configuration settings.



PROJECT IRRIGATION

RemoteXY allows to configure module for operate in one of two modes: access point and client. Client allows to connect the module to an existing Wi-Fi access point.

Access point mode of ESP8266 configures the module as an access point and allows to connect Arduino directly to this point. The access point is available to connect to it from smartphone or tablet within a radius of availability of the radio signal.

It does not require any other network infrastructure. This connection mode can be used in the far away place where there is no computer networks and the Internet. To configure this mode in the configuration properties of editor need to select the type of connection "Wi-Fi access point."

Client mode of ESP8266 configures the module for automatically connect to an existing Wi-Fi access point, such as a home router or enterprise access point. At the same time connected to the Arduino module ESP8266 must be located in the physical availability of the radio signal of access point.

Connecting to the device from smartphone or tablet will be not directly, but through the Ethernet network, an IP address provided by the DHCP server to ESP8266 module. It allows to connect to the Arduino device from anywhere in the local network as well as from the Internet.

Connect from the Internet is possible with the correct configuration of the router, such as the use of the virtual server. To configure this mode in the configuration properties of editor need to select the connection type "Ethernet" and select ESP8266 module.

The module is connected to the microcontroller board via serial interface. You can choose to use software serial or hardware serial. Module connection option is selected in the module interface settings of editor. The module is controlled via AT commands.

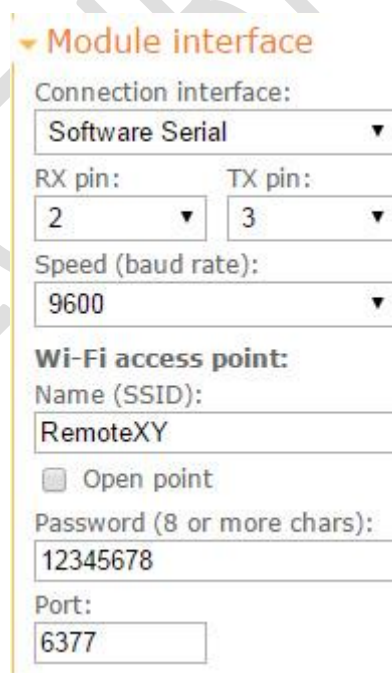
PROJECT IRRIGATION

Power of module can be provided from the DC-DC of Arduino board to 3.3V. Some boards, such as Arduino Nano can not provide sufficient power to output 3.3V, in this case, you need a separate power supply.

Connection via Software Serial

It allows to connect the module to an arbitrary microcontroller pins. The pins used must be select in the module interface settings of editor. Also in the settings must be select the baud rate for serial port. ESP8266 default configured to 115200 baud. To work through SoftwareSerial, must configure the transmission speed ESP8266 to a lower speed.

Stable operation of the module is possible at a speed of no more than 19200 baud. To change the speed of module serial interface, must to connect it to the console to the possibility of the AT commands. Module serial interface speed can be changed with the AT command:
"AT+UART_DEF=19200,8,1,0,0".



Module interface

Connection interface:
Software Serial ▼

RX pin: 2 ▼ TX pin: 3 ▼

Speed (baud rate):
9600 ▼

Wi-Fi access point:

Name (SSID):
RemoteXY

☐ Open point

Password (8 or more chars):
12345678

Port:
6377

There are some limitations on the use of the RX pin for Arduino boards. Limitations associated with do not support to pins interrupt of the microcontroller.

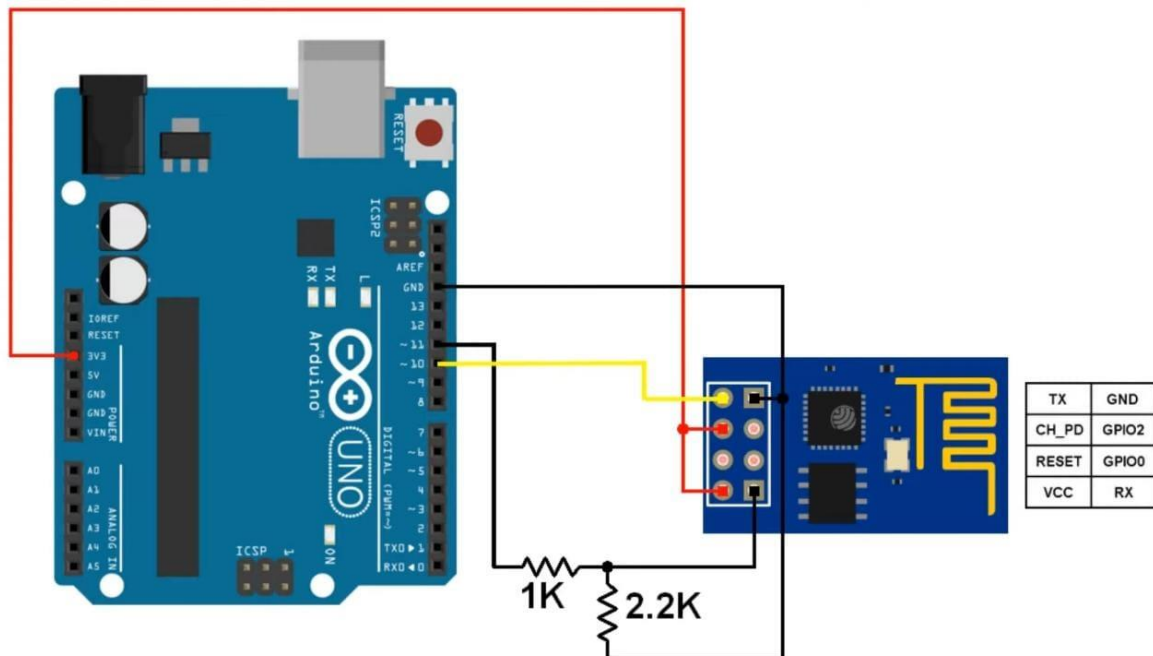
PROJECT IRRIGATION

- On Arduino RX doesn't work on pin 13
- On Mega and Mega2560 only the following can be used for RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8(62), A9(63), A10(64), A11(65), A12(66), A13(67), A14(68), A15(69)
- On Leonardo and Micro only the following can be used for RX: 8, 9, 10, 11, 14, 15, 16

Also, when using SoftwareSerial you should take the following limitations:

- No ability to work at high baud rates. It is not recommended to use a baud rate of more than 19200.
- Some libraries, which also uses interrupt may not work correctly. For example Servo library, the servos will twitch.

Example of ESP8266 module connecting via SoftwareSerial by pins 2(RX) and 3 (TX) on the picture. Please note that you must connect to pins the cross hairs, pin 2(RX) of Arduino to TX of ESP8266 module and pin 3(TX) to the RX pin of module.



Connection via Hardware Serial

It allows to connect the module to the microcontroller pins, supporting one of the hardware serial ports. For different Arduino boards are

MADE BY: ROHIT KUMAR SHAW (2021UCS1523) VEENIT KUMAR (2021UCS1548)
ADITYA GIRI (2021UCS1519) CHIRAG SAINI (2021UCS1564)

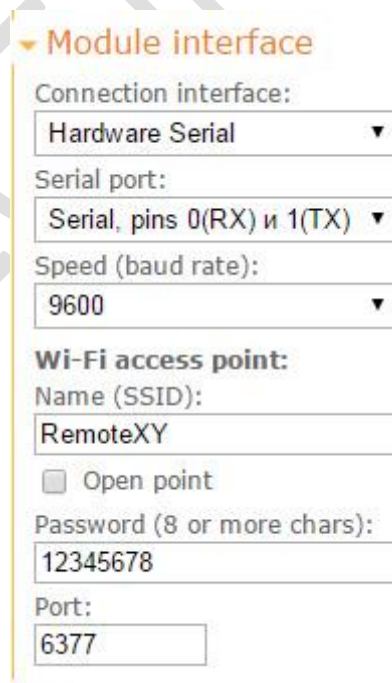
PROJECT IRRIGATION

different ports and contacts. Hardware serial allows to work module at a maximum connection speed of 115200 baud.

The baud rate for Arduino is selected in the module interface settings of editor. ESP8266 must be pre-configured to work at the same speed as the Arduino. The baud rate setting of module can be made via AT commands. Next Arduino serial ports are available:

- **Arduino UNO and Nano:** Serial (RX-0, TX-1)
- **Arduino MEGA and MEGA2560:** Serial (RX-10, TX-11), Serial1 (RX-19 и TX-18), Serial2 (RX-17 и TX-16), Serial3 (RX-15 и TX-14)

Which port to use, and therefore the pins should be connected to the module, you must select in interface settings of editor. It should also be borne in mind that for Arduino boards Serial port enabled for the microcontroller programming, and if you decide to use this port, you must disconnect the ESP8266 module when programming it. A good solution is to use the boards Arduino MEGA any port other than Serial.



Module interface

Connection interface:
Hardware Serial ▼

Serial port:
Serial, pins 0(RX) и 1(TX) ▼

Speed (baud rate):
9600 ▼

Wi-Fi access point:

Name (SSID):
RemoteXY

☐ Open point

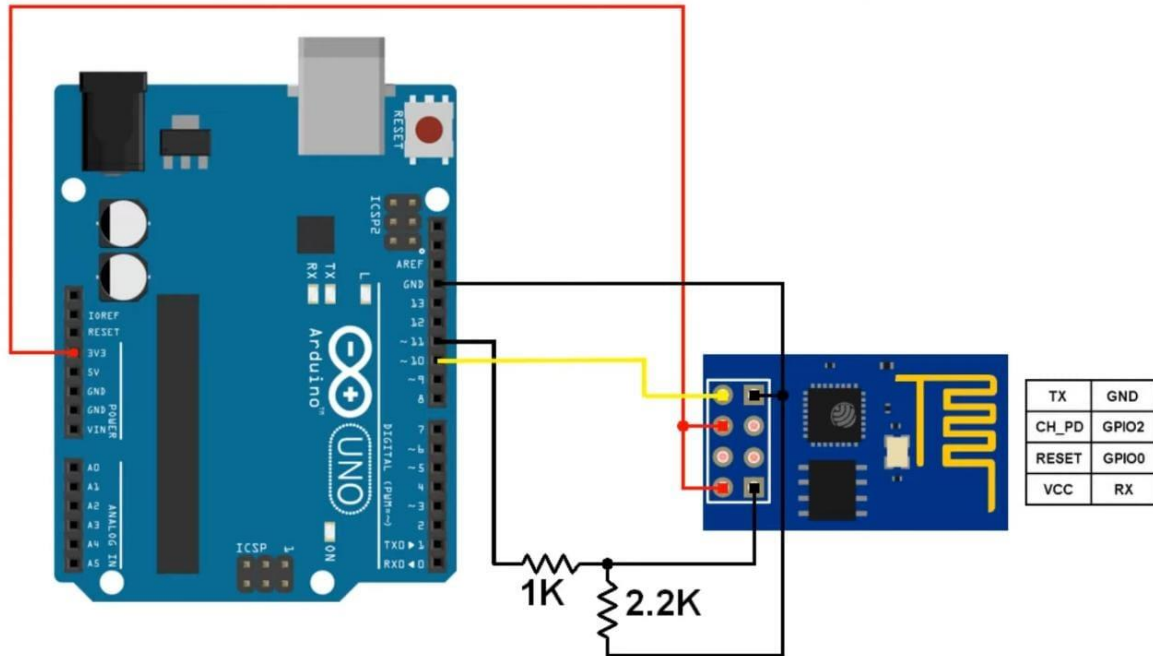
Password (8 or more chars):
12345678

Port:
6377

Example of HC-05(06) module connecting via HardwareSerial by pins 0(RX) and 1(TX) on the picture. Please note that you must connect to pins

PROJECT IRRIGATION

the cross hairs, pin 10(RX) of Arduino to TX of ESP8266 module and pin 11(TX) to the RX pin of module.



ARDUINO UNO + ESP8266 USING CLOUD SERVER

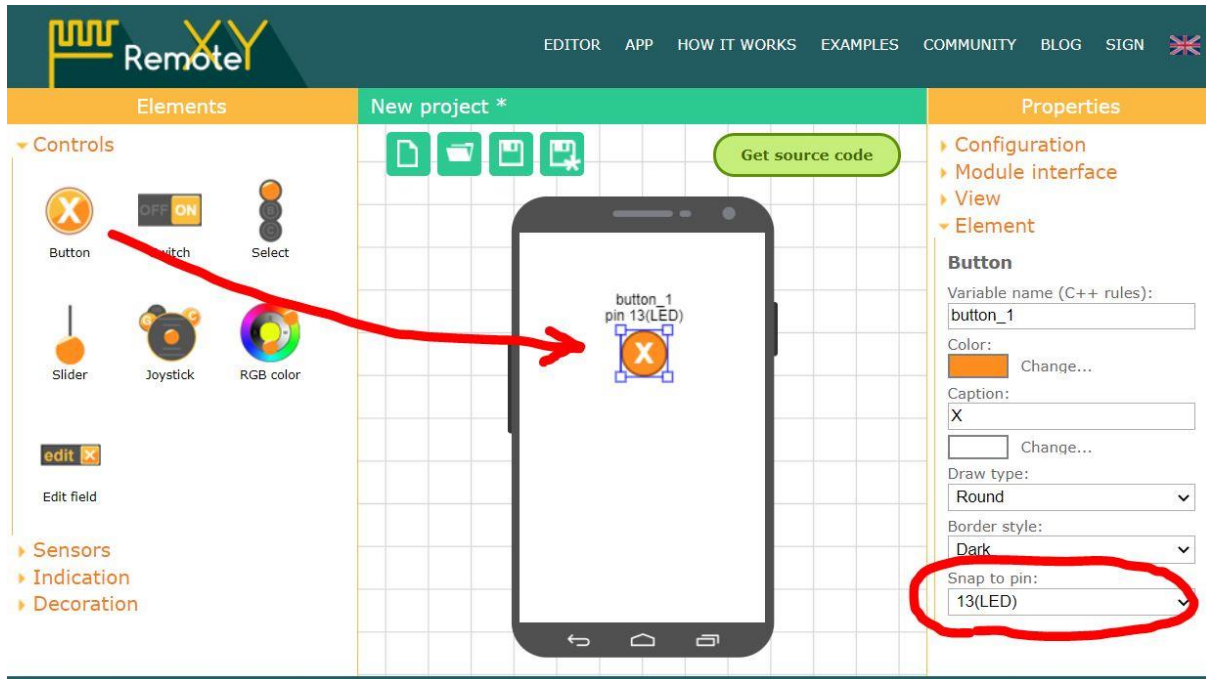
Tutorial to launch the RemoteXY graphical interface on Arduino UNO board via cloud server. External ESP8266 module connected to the hardware serial port. The ESP8266 will be configured to connect to your home WiFi network with internet access.

Note. Connecting using cloud server allows you to connect to your board from anywhere in the world.

Step 1. Create Graphical interface

Go to RemoteXY editor. Find the **Button** element on the left toolbar. Drag it to the phone screen. Select the **button**, you will see a frame around it. With the button selected, in the right toolbar, open the **Element** tab. Find **Snap to pin** setting and select **13(LED)** value.

PROJECT IRRIGATION



Note. The editor allows you to specify pin for some elements that can uniquely determine the state of the pin, such as high or low level. However, the best solution would be to write your own code to control the controller pins.

Step 2. Configure project

Open the **Configuration** tab on the right toolbar. Click on any item in the list, the configuration settings window will open. The configuration setting allows you to specify the options for your board and the connection method. Select the following settings and click the **Apply** button:

- Connection: **Cloud server**
- Board: **Arduino UNO**
- Module: **ESP8266 WiFi module**
- IDE: **Arduino IDE**

The configuration tab should look like the picture:

PROJECT IRRIGATION



Open the **Module interface** tab and set the following settings values:

- Connection interface: Hardware Serial
- Serial port: Serial, pins 10(RX) and 11(TX)
- Speed (baud rate): 115200
- Name (SSID): name of your home WiFi network
- Password: password of your home WiFi network

Module interface

Connection interface:

Software Serial

RX pin:

2

TX pin:

3

Speed (baud rate):

9600

Wi-Fi connection:

Name (SSID):

YOUR_SSID

Password:

YOUR_PASS

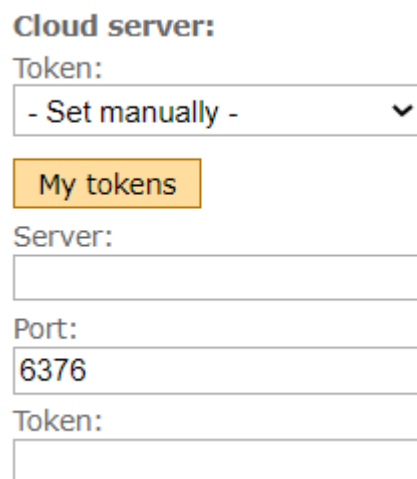
PROJECT IRRIGATION

The settings determine that the ESP8266 module connects to the Arduino through the Hardware Serial at 115200 and uses pins 10 and 11. The settings also include the name of your home WiFi hotspot and password. Through this access point, the board will need to connect to the Internet.

Note. The ESP8266 is configured with a default baud rate of 115200. For this reason, you need to use Hardware Serial for communication. Software Serial cannot run at this speed.

Step 3. Generate a cloud server token

In the same **Module interface** tab, click the **My tokens** button, you will be redirected to the cloud server token page of your account.



Cloud server:

Token:

- Set manually -

My tokens

Server:

Port:

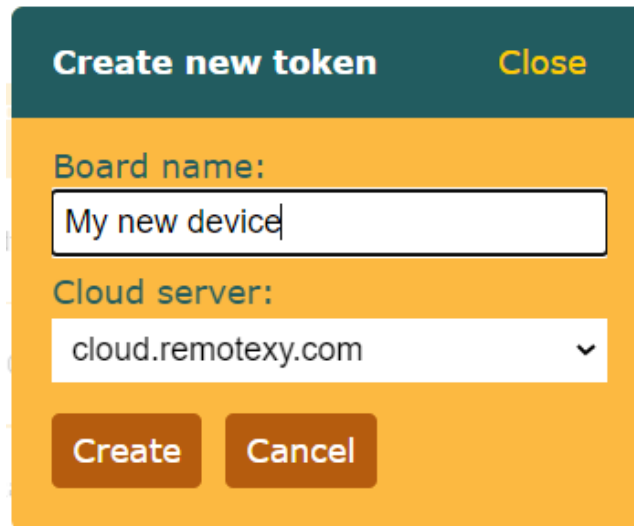
6376

Token:

Note. To create a cloud server token, you need to register on the RemoteXY website.

Click the **Create new token** button. In the window that opens, enter any token name and click the **Create** button.

PROJECT IRRIGATION



The new token should appear in the list of tokens.

My cloud tokens

Nº	Device name	Token	Device state	Server	Device port	App port	
1	My device	391213d8d1c2a6213365529a0f47b327	disconnected	cloud.remotexy.com	6376	6375	Edit Delete

Create new token

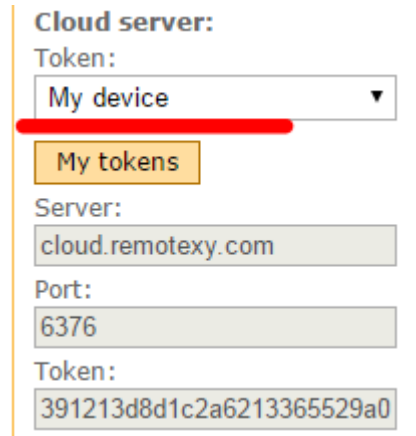
The token is required to connect to the device through the cloud server. Communication via the cloud server allows to control the device from anywhere in the world.

*Note: For each of your board that work using a cloud server, you need to create a separate token. You can control the status of your board in the list of tokens in your account. See the **Device status** column. The state can be one of three:*

- **disconnected** – the board is not connected to the cloud server;
- **connected** – the board has connected to the cloud server and is waiting for a connection from the mobile app;
- **used** — connected to the board from a mobile app.

Return to the editor, open the same **Module interface** tab and select the newly created token in the **Token** field. The token data will be filled in the corresponding fields below.

PROJECT IRRIGATION



Cloud server:
Token:
My device
My tokens
Server:
cloud.remotexy.com
Port:
6376
Token:
391213d8d1c2a6213365529a0

Step 4. Get the source code for Arduino

Press **Get source code** button.

On the page that opens, click the **Download code** link and download the archive of the sketch for Arduino IDE. Unpack this archive. The archive contains a file project.ino, open it in Arduino IDE.

To compile the sketch, you need to add the RemoteXY library to the Arduino IDE. Go to the download link for the library and follow the tutorial to install it.

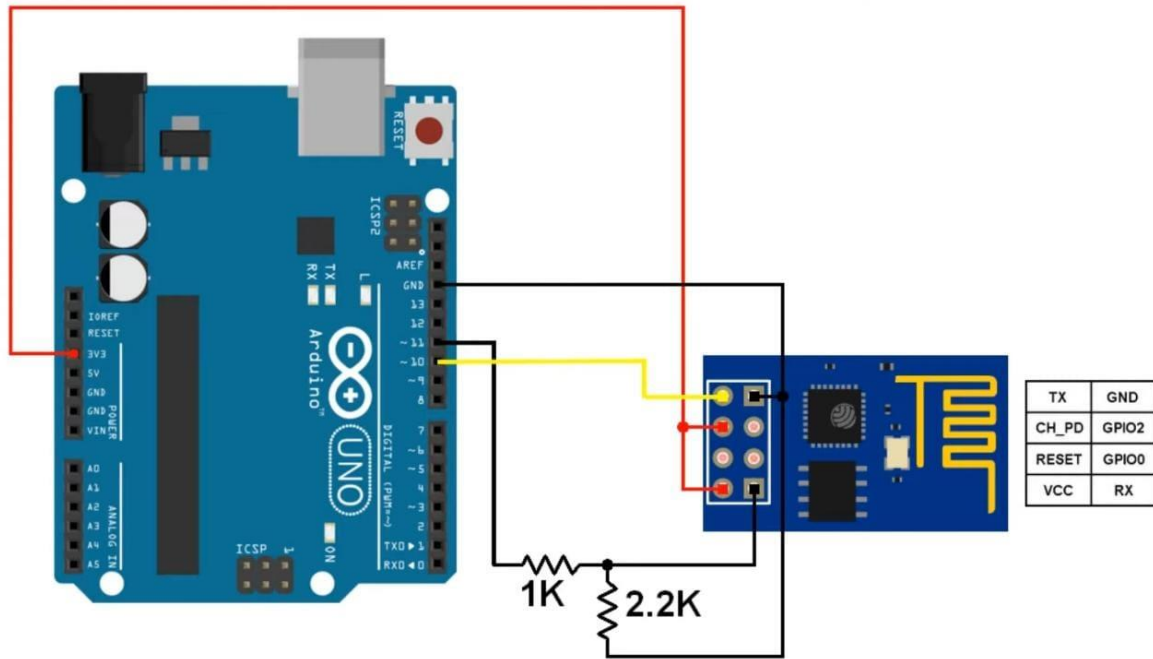
In the Arduino IDE settings, select the **Arduino UNO** board and try compiling the sketch. If you did everything correctly, the sketch should compile without errors.

Step 5. Connect ESP8266 to Arduino Uno

Connect ESP8266 to Arduino Uno according to the diagram below. Note that pin 10 of the Arduino board selected as RX connects to the TX pin of the Bluetooth module, and pin 11 of the Arduino board selected as TX connects to the RX pin of the ESP8266 module.

The signal levels of the ESP8266 module are 3.3V but the Arduino board runs at 5V, you need to use a resistive voltage divider to convert the TX output signal level.

PROJECT IRRIGATION



Note. Some ESP8266 modules are tolerant to the TX signal level and you may not need to use the voltage converter if your module turns out to be like that.

Step 6. Upload the sketch to Arduino board.

Since the ESP8266 module is connected to the main Serial port of the Arduino board, this will interfere with the download of the sketch. Before uploading the sketch, you need to disconnect the ESP8266 from pins 10 and 11 of the Arduino board.

After disconnecting the ESP8266, upload the sketch to the board in the usual way. Connect the Arduino UNO board to your computer, select the COM port to which the board is connected and click the download button. After uploading the sketch, connect the pins of the ESP8266.

Attention. Disconnect and connect the ESP8266 only when there is no power to the board.

If you did everything correctly, then immediately after starting the Arduino you will see a blinking blue LED on the ESP8266 module for about

PROJECT IRRIGATION

half a second. If this happens, then most likely the data exchange between the Arduino board and the ESP8266 is established.

During this time, RemoteXY configures the ESP8266 to work as an access point using AT commands, and the LED blinks during data exchange.

Step 7. Connect to the board using the mobile app.

Install the RemoteXY mobile app to your phone.

Launch the application and click the **+** (Plus) button on the top navigation bar. In the window that opens, select **Cloud server** connection.

Enter your token in the **Device Token** field of the window that opens. You created the token in step 3. Enter the alphanumeric characters of the token, not the name of the token.

Press the **Connect** button. If you did everything correctly, then a graphical interface should open next.

