**Project Report**

**OF**

**IIoT**

**SUB-CODE:** IIOT 2

[COMPUTER SCIENCE AND ENGINEERING]



|  |  |
| --- | --- |
| **Submitted To:** | **Submitted By:** |
| Mr. *Dileep Kuamr Tiwari* | *Shahnawaz Alam* |
| (Advisor IIoT) | Roll no: 11202722 (B1)  Class: B. Tech. (2nd Y)  Session: 2021-2022 |

**MAHARISHI MARKANDESHWAR ENGINEERING COLLAGE**

**MAHARISHI MARKANDESHWAR (DEEMED TO BE UNIVERSITY)**

**MULLANA-AMBALA, HARYANA (INDIA) – 133207**

(Established Under Section 3 of the UGC Act, 1956)

(Accredited by NAAC with Grade ‘**A++**’)

Catalog

[Project 1: Smart Water Management 3](#_Toc10865)

[Why Smart Technologies required? 3](#_Toc29388)

[Technical Requirements: 4](#_Toc1233)

[Block Diagram 4](#_Toc25972)

[Sensors 5](#_Toc14250)

[1. pH Sensor : 5](#_Toc31770)

[2. Turbidity 6](#_Toc30531)

[Sensor Specification - 6](#_Toc32590)

[Circuit Diagram of turbidity sensor 7](#_Toc16838)

[Circuit Diagram for Turbidity Meter 8](#_Toc21767)

[ARDUINO TURBIDITY METER SOURCE CODE PROGRAM - 8](#_Toc3067)

[3. Temperature Sensor 9](#_Toc21226)

[Circuit diagram Temperature Sensor 10](#_Toc16616)

[Implementation code: 10](#_Toc23546)

[4. Flow Sensor: 11](#_Toc4950)

[Circuit Diagram Flow sensor 11](#_Toc27772)

[Implementation code: 11](#_Toc22664)

[5. Ultrasonic Sensor: 13](#_Toc3675)

[Circuit diagram of Ultrasonic Sensors 13](#_Toc7595)

[Implementatoin code: 13](#_Toc16722)

[Wired communication protocol - I2C, SPI, UART 15](#_Toc30151)

[1. I2C (Inter-Integrated Circuit): 15](#_Toc32155)

[2. SPI : 15](#_Toc11321)

[3. UART: 16](#_Toc27903)

[Iot Hardware/Controller (Modules) - ESP8266 17](#_Toc10239)

[Arduino 17](#_Toc5493)

[NodeMCU ESP8266: 20](#_Toc28253)

[Arduino Actuators - Motorized valve, Relays, Thrusters 21](#_Toc18872)

[Motorized valve, 21](#_Toc26527)

[Relay 21](#_Toc6004)

[Wireless Communication Protocol - HTTP, MQTT 22](#_Toc24329)

[HTTP: 22](#_Toc26622)

[MQTT: 22](#_Toc16182)

[Cloud Server – ThingWorx 23](#_Toc28598)

[ThingWorx 23](#_Toc22129)

# Project 1: Smart Water Management

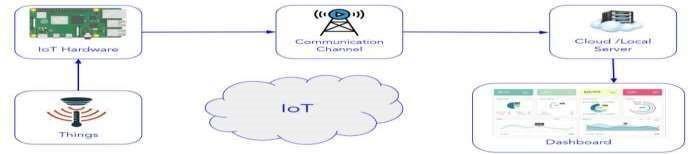
# **Why Smart Technologies required?**

* The ability to detect, sense, measure, and record data.
* The ability to community and interact with system operators and managements.
* The ability to analyze the situation, enable quick response and optimize troubleshooting solutions.
* High-technology solutions

# **Technical Requirements:**

* Sensors - pH, Turbidity, Temperature, Flow, Ultrasonic
* Wired communication protocol - I2C,SPI,UART
* Iot Hardware/Controller(Modules) - ESP8266,
* Arduino Actuators - Motorized valve, Relays, Thrusters
* Wireless Communication Protocol - HTTP, MQTT
* Cloud Server - ThingWorx
* Dashboard - Visualization, Analysis

# **Block Diagram**

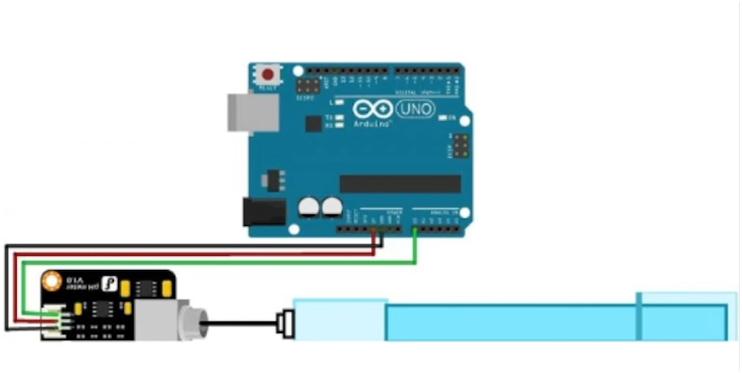


# **Sensors**

# **1. pH Sensor :**

* The PH value is an important indicator for monitoring water quality.
* In smart water management, the most suitable pH range is 6.5-7.5.
* When the pH is lower than 6.5, the fungus starts to compete with the bacteria.
* When the pH reaches 4.5, the fungus will have a complete advantage in the biochemical tank, which will seriously affect the settlement of the sludge.
* When the pH exceeds 9, the metabolic rate of microorganisms will be hindered.

## Circuit diagram pH Sensors



## Implementation code:

int sensorValue = 0;

float voltage = 0.00f;

float pHvalue = 0.00f;

float calibrationFactor = 3.50f;

void setup()

{

    Serial.begin(9600);

}

void loop()

{

    int sensorValue = analogRead(A0);

    voltage = (float)sensorValue \* (5 / 1023.0);

    pHvalue = voltage \* calibrationFactor;

    Serial.print("Ph: ");

    Serial.println(pHvalue);

    delay(1000);

}

# Roinco Turbidity Sensor : Amazon.in: Industrial & Scientific**2. Turbidity**

* Turbidity is the measurement of water clarity.
* The measurement of turbidity is a key test of water quality.

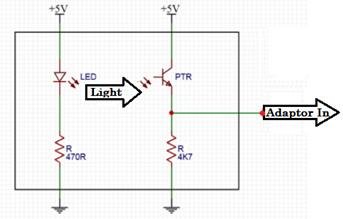
## Measuring turbidity:

Turbidity is measured using specialized optical equipment in a laboratory or in the field. Light is directed through a water sample, and the amount of light scattered is measured.

# **Sensor Specification -**

1. Operating Voltage: 5V DC
2. Operating Current: 40mA (MAX)
3. Response Time: <500ms
4. Insulation Resistance: 100M (Min)
5. Output Method: Analog
6. Analog output: 0-4.5V
7. Digital Output: High/Low-level signal (you can adjust the threshold value by adjusting the potentiometer)

# **Circuit Diagram of turbidity sensor**



**Basic Turbidity Sensor Arduino interfacing code**

This code will read the analog value from the sensor and display it on the Serial Monitor.

void setup()

{

    Serial.begin(9600);

}

void loop()

{

    int sensorValue = analogRead(A0);

    float voltage = sensorValue \* (5.0 / 1024.0);

    Serial.println("Sensor Output (V):");

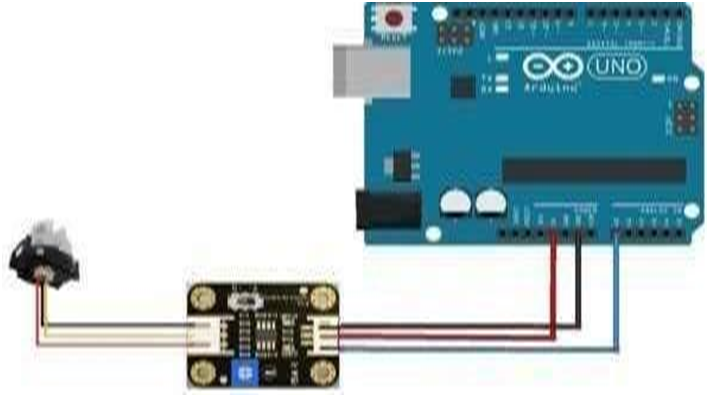
    Serial.println(voltage);

    Serial.println();

    delay(1000);

}

# **Circuit Diagram for Turbidity Meter**



# **ARDUINO TURBIDITY METER SOURCE CODE PROGRAM -**

#include <LiquidCrystal\_I2C.h>

    LiquidCrystal\_I2Clcd(0x27, 16, 2);

Int sensorPin = A0;

float volt;

float ntu;

void setup()

{

    Serial.begin(9600);

    lcd.begin();

    lcd.backlight();

}

void loop()

{

    volt = 0;

    for (int i = 0; i < 800; i++)

    {

        volt += ((float)analogRead(sensorPin) / 1023) \* 5;

    }

    volt = volt / 800;

    volt = round\_to\_dp(volt, 2);

    if (volt < 2.5)

    {

        ntu = 3000;

    }

    else

    {

        ntu = -1120.4 \* square(volt) + 5742.3 \* volt - 4353.8;

    }

    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print(volt);

    lcd.print(“V”);

    lcd.setCursor(0, 1);

    lcd.print(ntu);

    lcd.print(“NTU”);

    delay(10);

}

float round\_to\_dp(float in\_value, int decimal\_place)

{

    float multiplier = powf(10.0f, decimal\_place);

    in\_value = roundf(in\_value \* multiplier) / multiplier;

    return in\_value;

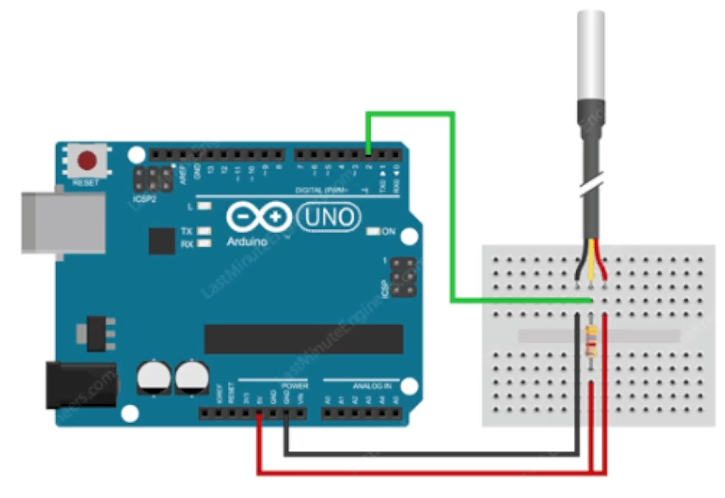
}

Thus, the final arduino turbidity meter is ready now and can be used for water quality monitoring.

# **3. Temperature Sensor**

* Temperature sensor is a device, typically a thermocouple or resistance temperature detector .
* It provides temperature measurement in readable form through an electrical signal.
* A thermonuclear is the most basic form of a temperature meter that is used to measure the degree of hotness and coolness.

# **Circuit diagram Temperature Sensor**



# **Implementation code:**

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 2

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(& oneWire);

void setup(void)

{

  Serial.begin(9600);

  Serial.println("DallasTemperature IC Demo");

  sensors.begin();

}

void loop()

{

  Serial.print("Requesting remperatures... ");

  sensors.requestTemperatures();

  Serial.println("Done");

  Serial.println(sensors.getTempCByIndex(0));

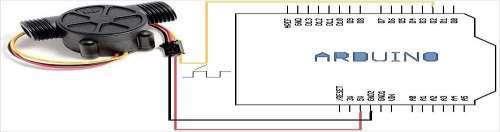
  delay(100);

}

# **4. Flow Sensor:**

* A flow sensor is an electronic device that measures or regulates the flow rate of liquids and gasses within pipes and tubes.
* Flow sensors are generally connected to gauge to render their measurements, but they can also be connected to computers and digital interfaces.

# **Circuit Diagram Flow sensor**



# **Implementation code:**

#include <FlowMeter.h>

interrupt pin 2 FlowMeter \*Metre;

const unsigned long period = 200;

void MeterS()

{

    //  metre count the pulses

    Metre->count();

}

void setup()

{

    Serial.begin(115200);

    pinMode(8, OUTPUT);

    Metre = new FLowMeter(digitalPin ToInterrupt(2),

                          Uncalibrated Sensor, MeterISR, RISING);

}

void loop()

{

    delay(period);

    Metre->tick(period);

    Serial.println("Currenly" + String(Metre->getCurrent Flow Rate()) + "l/min," + String(Metre->getTotalVolume()) + "total.");

  if (Meter->getCurrent Flow Rate() {

        digitalWrite(8, HIGH);

  }

  else

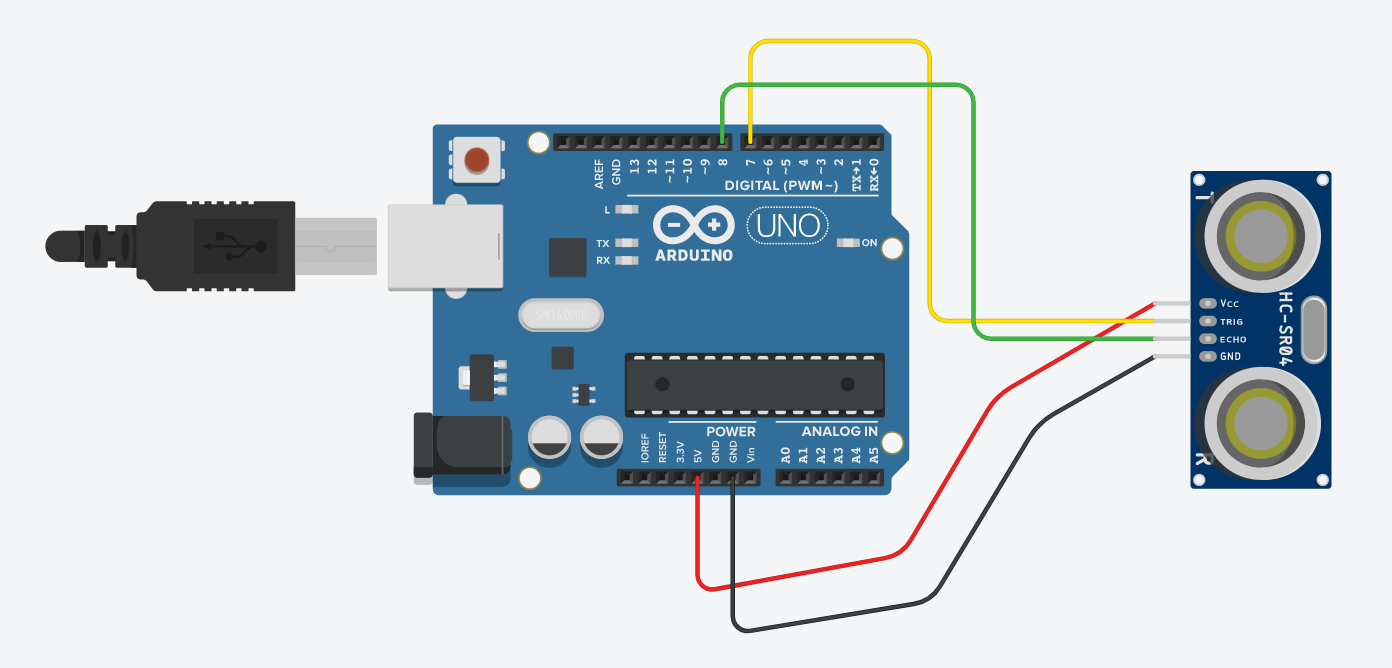
    digitalWrite(8, LOW);

}

# **5. Ultrasonic Sensor:**

* An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves.
* An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object’s proximity.
* High -frequency sound waves reflect from boundaries to produce distinct echo patterns.
* There are following wired communication protocol which we have to
* followed:

# **Circuit diagram of Ultrasonic Sensors**



# **Implementatoin code:**

int cm = 0;

int inches = 0;

void setup()

{

  pinMode(7, OUTPUT);

  pinMode(8, INPUT);

  Serial.begin(9600);

}

void loop()

{

  digitalWrite(7, 0);

  //  delayMicrosecond(2);

  delay(2 / 1000000);

  //  delayMicrosecond(10);

  digitalWrite(7, 1);

  delay(1 / 100000);

  cm = 0.01732 \* pulseIn(8, 1);

  inches = (cm / 2.54);

  Serial.print(cm);

  Serial.print(" cm, ");

  Serial.print(inches);

  Serial.println(" Inches");

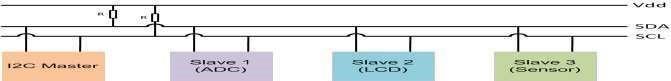
  delay(1000);

}

# **Wired communication protocol - I2C, SPI, UART**

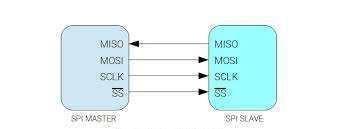
# **1. I2C (Inter-Integrated Circuit):**

* I2C stands for Inter - Integrated Circuit.
* It is a bus interface connection protocol incorporated into devices for serial communication.
* It was originally designed by Philips Semiconductor in 1982.
* It was widely used protocol for short distance communication.
* It is also known as Two wired interface.



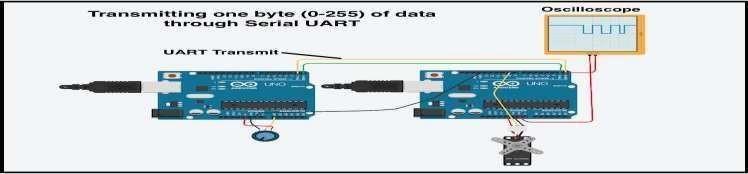
# **2. SPI :**

* Serial peripheral Interface or SPI
* It is synchronous serial communication protocol that provide full - duplex communication at very high speeds.
* Serial peripheral Interface (SPI) is a master -slave type protocol that provides a simple and low cost interface between a microcontroller and its peripherals.
* SPI is used to authorise a microcontroller to communicate with peripheral devices single master, single slave



# **3. UART:**

* UART stands for universal Asynchronous Transmitter Receiver
* It is the most common protocol used for full- duplex serial communication.
* It is a single LSI (large-scale-integration) chip designed to perform asynchronous communication.
* This device sends and receives data from one system to another system.



# **Iot Hardware/Controller (Modules) - ESP8266**

# **Arduino**



## **Requirement of Arduino**

* Required embedded hardware to test POC
* Strong connectivity between hardware and software
* Sot effective hardware within programmer
* Simple, clear programming environment
* Open source and extensible software and hardware
* It his used to build low cost scientific instruments, robotics, innovative ideas which enhance thinking and analytical skill set

**History of Arduino**

* The Arduino project began in 2005 as a tool for students at the interaction design institute lvrea, Italy.
* Founder - Massimo Banzi
* The very first Arduino board released (in 2006) was the Arduino Serial.
* Popular in mid 2011.
* The name Arduino comes fro a bar in lvera, Italy.
* The bar was named after Arduino of lvera, who was the margrave of the march of lvrea and king of Italy (1002 to 1014).
* Arduino --> German --> Harduwin --> Hardu --> Strong --> Wini --> Friend.

**Concepts : Input vs output**

* Inputs is a signal/ information going into the board.
* Examples: buttons, witches, light sensors, flex sensors, humidity sensors, temperature sensors…
* Output is any signal exiting the board.
* Examples: LEDs, DC motor, servo motor, a piezo buzzer, relay, an RGB LED

**What is Arduino?**

* Arduino is an open-source platform based on easy to use hardware and software.
* Inputs: light on a sensors, a finger on a button
* Output: activating a motor, turning on an LED, publishing something online.
* Low cost 500 rs (around)
* Easy to program via USB

**Features of Arduino UN**

* Microcontroller: ATmega328
* Operating Voltage: 5v
* Input Voltage (recommended): 7-12V
* Input Voltage (limits): 6-20V
* Digital 1/O Pins: 14 (of which 6 provide PWM output)
* Analog Input Pins:6
* DC Current per /O Pin: 40 mA
* DC Current for 3.3V Pin: 50 mA
* Flash Memory: 32 KB of which 0.5 KB used by bootloader
* SRAM: 2 KB (ATmega328)
* EEPROM: 1 KB (ATmega328)
* Clock Speed: 16 MHz

**Features of ATmega328**

* Program Memory Type: Flash
* Program Memory Size (KB): 32
* MICROCH ATmega32
* CPU Speed (MIPS/DMIPS): 20
* SRAM (bytes): 2,048 or 2kb
* Data EEPROM/HEF (bytes): 1024 or 1kb
* Digital Communication Peripherals: 1-UART, 2-SPI, 1-12C
* Capture/Compare/PWM Peripherals: 1 Input Capture, 1 CCP, 6PWM
* Number of comperators: 1
* Temperature Range C : -40 to 85
* Operating Voltage Range (V): 1.8 to 5.5
* Total Pin: 28
* I/O Pin: 23
* ADC Module: 6 channels, 10-bit resolution ADC
* External Oscillator:
  + (a) 0-4 MHz 1.8V to 5.5V
  + (b) 0-10 MHz@ 2.7V to 5.5V
  + (c) 0-20 MHz @ 4.5V to 5.5V
* Internal Oscillator: 8MHz Calibrated Internal Oscillator

**Arduino - Hardware Description**

1. Arduino UNO board because it is the most popular board in the Arduino board family.

2. It is the best board to get started with electronics and coding.

3. Most of the Arduinos board common Components like this board.

**Crystal Oscillator**

It provide clock to the processor which is work as heart beat for the processoor.

**Arduino Reset**

To start your program trom the beginning required to reset. In Arduino UNO board two types of reset you can

1. By using the reset button 17 on the board.

2. By connecting an external reset button to the Arduino pin labelled RESET

**Pins (3.3, 5, GND, Vin)**

3.3V6- Supply 3.3 output volt

1. 5V- Supply 5 output volt
2. GND 8- To provide GND signal to the different circuit 3.
3. Vin-To power the Arduino board from an external power source.

**Analog pins**

Arduino UNO board has six analog input pins A0 through A5. It takes the analog signal trom sensor and convert it into a digital value that can be read by the microcontroller.

**Main microcontroller**

Arduino board has its own microcon troller (11). That work as a brain of your board and all program burn in it.

After testing the POC, this controller embed to the application

**ICSP pin**

ICSP is in circuit serial programmer. It's used for the SPI (Serial Peripheral Interface)

**Power LED indicator**

Power LED should light up when you plug your Arduino.

If this light does not turn on, then there is something wrong with the connection.

**TX and RX LEDs**

Labels: TX (transmit) and RX (receive).

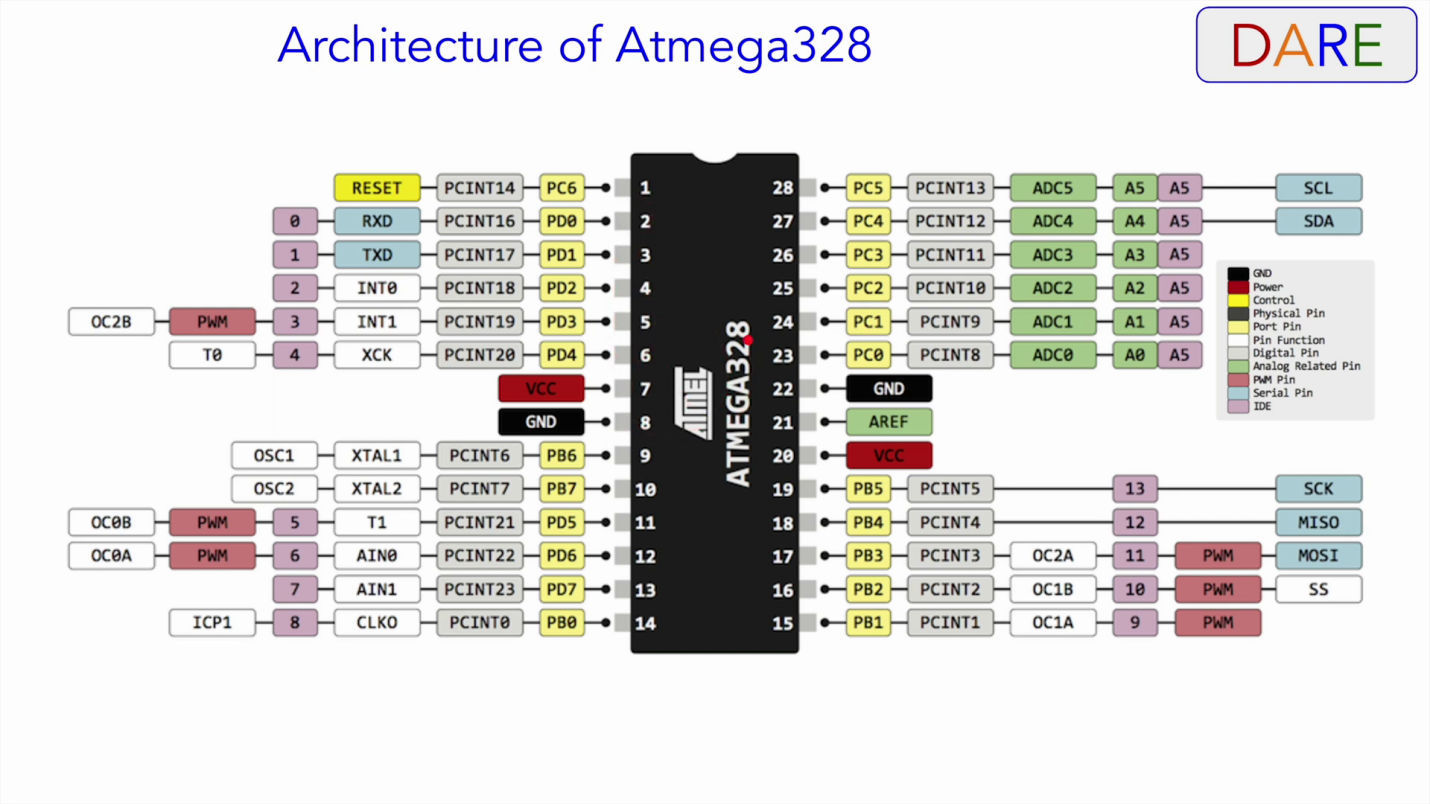
These LED’s blink when the data are transmitted and receiving through the Arduino to external

**Digital I/O**

Arduino UNO board has 14 digital I/O pins in which 6 provides PWM (Pulse Width Modulation) output. The pins labeled “~” can be used to generate PWM

**AREF**

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.



# **NodeMCU ESP8266:**

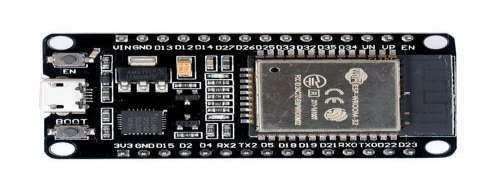
The ESP8266 WiFi Module is a self contained SOC with integrated

TCP/IP protocol stack that can give any microcontroller access to your WiFi network

The ESP8266 is capable of either hosting an application or offloading all WiFi networking functions from another application processor.

Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much WiFi-ability as a WiFi Shield

The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.



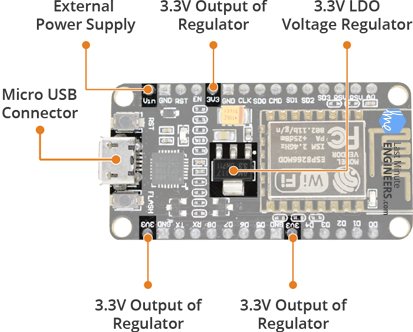
**Requirement of NodeMCU ESP8266**

Operating Voltage: 2.5V to 3.6V

On-board 3.3V 600mA regulator

80mA Operating Current

20 µA during Sleep Mode



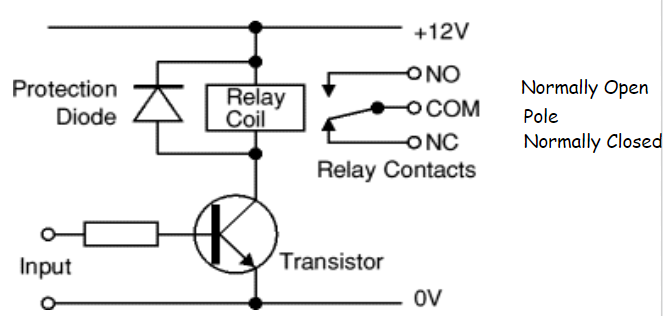
# **Arduino Actuators - Motorized valve, Relays, Thrusters**

# **Motorized valve,**

* To control the valve via electrically
* To control the flow of water/ liquid
* Precise controlling of flow
* Motor operated valves are also called as on-off valves
* Motors sever the purpose of fully opening or fully closing values in pipelines.
* It can be used to fully allow or fully stop the fluid flow.

# **Relay**

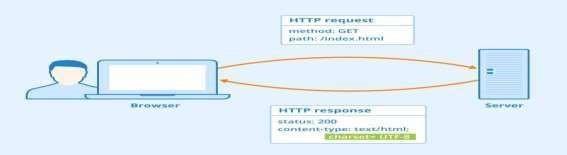
* Relay is electromagnetic switch.
* When DC suppl y provide to the relay it energized the coil and attract the pole
* According to the pole it shift to the normally close to normally open



# **Wireless Communication Protocol - HTTP, MQTT**

# **HTTP:**

* HTTP stands for Hypertext Transfer Protocol.
* Hypertext Transfer Protocol is the best example of Iot network protocol.
* This protocol has formed the foundation of data communication over the web.
* It is the most common protocol that is used for IoT devices when there is a lot of data to be published.
* However, the HTTP protocol is not preferred because of its cost, battery life, energy saving, and more constraints.
* Additive manufacturing/3D printing is one of the use cases of the HTTP protocol. It enables computers to connect 3D printers in the network and print three-dimensional objects and pre-determined process prototypes.



# **MQTT:**

* Message Queue Telemetry Transport (MQTT).
* One of the most preferred protocols for IoT devices.
* MQTT collects data from various electronic devices and supports remote device monitoring.
* It is a subscribe/publish protocol that runs over Transmission Control Protocol (TCP), which means it supports event-driven message exchange through wireless networks.
* MQTT is mainly used in devices which are economical and requires less power and memory.
* For instance, fire detectors, car sensors, smart watches, and apps for text based messaging

# **Cloud Server – ThingWorx**

# **ThingWorx**

* ThingWorx is end-to-end technology platform.
* It is designed for the industrial Internet of Things (IIoT).
* ThingWorx enable to manage the development lifecycle for IoT Application.
* It provides many types of features to manage the IoT devices.
* It helps to develop and deploy of smart, connected devices rapidly

