PROJECT REPORT

Project Name **:** SMARTFARMER - IOT ENABLED SMART FARMING APPLICATION

Team Id **:** PNT2022TMID30027

Team leader **: Manojkumar. M**

Team members **: Kiran. V**

Dhanush. U

Raghul. k

TABLE OF CONTENTS

ABSTRACT

In daily operations related to farming of field, watering is the most important practice and intensive task. No matter whichever weather it is, either too hot and dry or too cloudy and wet, you want to be able to control the amount of water that reaches in plants, crops etc., For implementation of automatic smart gardening system, we have used combination of electromechanical relay and pump/motor. The main objective of this project is to provide auto irrigation system based on IoT with IBM Watson cloud computing to sense the soil moisture level and to give water to field. And this level of sensing is done by soil moisture sensor which detects the moisture level through the DHT11 sensor humidity and temperature can also be identified through the atmosphere condition. When the moisture level of the soil is decreased below a certain level then the sensor sends the detected value to the Node MCU. According to the value that is sensed by the sensor, the water is supplied to the field automatically to the desired level, temperature and humidity in order to maintain the moisture content in the soil by IoT with Watson cloud computing technology. The theme of this project is to reduce the human intervention by using this smart field system by IBM Watson cloud technology system. The overall system controlled by the Node MCU and the data are stored in cloud. The moisture in the soil is sensed and based on which the motor operation takes place. This system is environment-friendly and cost effective.

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW

An automation of irrigation systems has several positive effects. Once installed, the water distribution on fields or small-scale of agriculture is easier and does not have to be permanently controlled by an operator of cloud computing. There are several solutions to design automated irrigation systems. Modern big-scale systems allow big areas to be managed by one operator only. Sprinkler, drip or subsurface drip irrigation systems require pumps and some high tech-components and if used for large surfaces skilled operators are also required. Extremely high-tech solutions also exist using GIS and satellites to automatically measure the water needs content of each crop parcel and optimize the irrigation system. But automation of irrigation can sometimes also be done with simple, mechanical appliances: with clay pot or porous capsule irrigation networks or bottle irrigation (see also manual irrigation). We all know that plants and crop cultivation are very need to all human beings in many aspects. Plants and crops helps in keeping the environment healthy by cleaning air naturally and producing oxygen. Many people love to have plants in their backyard. But due to civilization and insufficiency of place many people used to grow plants in mound or dirt, pot, and placed on the windowsill. These plants and crops are dependent on conventional breeding - watering and provide the right amount of sun to sustain life and growth. In busy schedule of day-to-day life, many time people forget to water their plants and due to this, plants suffer many disorders and ultimately died. In addition, the world's biggest problem in modem society is the shortage of water resources, agriculture is a demanding job to consume large amounts of water. It is very essential to utilize the water resources in proper way. Here the data are stored in Watson cloud and can be collected at any

time and from anywhere it is needed. Mostly the temperature and humidity are being detected by the DHT11 sensor. Cloud storage data are the advance technology to store data and to collect it.

1.2 OBJECTIVE

- The main objective of this work is to control the water application. Once installed, the water distribution on fields or small-scale agriculture is easier and does not have to be permanently controlled by an operator of cloud data.
- This System consists of a water pump, electromechanical relay, soil moisture sensor and IoT based cloud technology. if the soil moisture level is detected and based on the sensor data the electromechanical relay activates the water pump.
- Water conservation in garden is controlled using Node MCU with soil moisture sensor and IBM Watson cloud technology of IoT.
- To save the plants from being dry and improve their lifetime.
- Humidity and temperature of filed location are very important for agriculture method, which can also be identified.

CHAPTER 2
LITERATURE SURVEY

[1] S. Harishankar, R. Sathish Kumar, Sudharsan K.P, U. Vignesh and T.Viveknath, 2020, "Smart Irrigation System", Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 4, Number 4, pp. 341-346, Research India Publications.

The entire system can be solar powered to make the system environment friendly because solar energy is widely available energy source on the earth. Solar power is good in the view of the economy and also environment friendly form of the energy. Now days this energy is used in street lighting and in other domestic loads. In today's life due to advance technologies the cost of solar panels is decreased. One of the applications of solar energy is in irrigation system. In India there is major problem of energy, therefore solar energy is best solution for Indian farmer.

[2] Alsayid B, Jallad J, Dradi M and Al-Qasem O „Automatic Irrigation System ",

There are different irrigation systems that are used nowadays to reduce dependency on rain. Due to the lack of electricity and mismanagement in the manual control irrigation system many times crops become dry or flooded with water. So to avoid this problem sensor base irrigation system is used. In manual system, farmers usually control the electric motors observing the soil, crop and weather conditions by visiting the sites which is time consuming and the simple analysis by observation of condition of soil may or may not be correct.

[3] Arduino Based Automatic Water Planting System using Soil Moisture Sensor [Hriday Chawla, Praveen Kumar March 15,2019]

This research paper is about the automatic water planting system using a moisture sensor which senses the humidity level of the soil. Depending on the moisture or humidity level of the soil, water pump is being set on or off. This research is being done using Arduino on Arduino ide. This research has increasing demands in agriculture sector. Using this system farmer can easily monitor usage of water according to crops they use. By using this method, they can cultivate crops more easily and it reduces the labor. It also helps to maintain the health of the crops and also increase the production by farmers. In this research, we also tested this system on soil for few days and noted its effective results.

[4].A Review and Proposed Automated Irrigation System using Soil Moisture Sensor and Android App In this author (Laxmikant Jayprakash Goud)

proposed "A Review and Proposed Automated Irrigation System using Soil Moisture Sensor and Android App" This Application will work with the help of Hardware. Hardware which required is Sensors, Controller, GSM Kit (Global system for mobile communication). First, the sensor will place in the field. As per its name it will detect the moisture present in the soil. This produces a voltage level in terms of output. As Machine will only understand the language of voltage, this voltage will be acquired by the controller and it will produce output. In this way sensor and controller will work together to produce the moisture level. After getting the voltage levels, it decides how much water supply needed by the soil. The work of GSM is to allow user to handle the water Sources at the remote location. As well as it will notify user at the time whether actually water supply is

started or not. In this way user can handle water supply if he is not physically present there.

[5] A survey on Automatic Irrigation System using Wireless Sensor Network Nattapol Kaewmard et al., 2019.

describe the design of an automated irrigation system using WSN including soil moisture sensor, air temperature sensor and air humidity sensor in order to collect environmental data and controlling the irrigation system. By using smart phone, the irrigation system uses values to turn on/off the solenoid valve. The irrigation system control water by sending and receiving control commands from smart phone application via the internet. Result shows that proposed AIS is useful, cost effective and provides better performance than conventional system.

CHAPTER 3

EXISTING SYSTEM & PROPOSED SYSTEM

3.1 EXISTING SYSTEM

- Knowing when and how much amount to water plants, are two major important aspects of plant watering.
- During day to-day activities many people have many work often forget to water their plants and crop to field and thus it becomes challenging for them to keep their plants healthy and alive.
- Based on the above background, we thought that it is necessary to implement the automated system which will take care of plants and crops considering all the different aspects of agriculture system (for system based on household purpose) and helps them to grow healthy.

PROPOSED METHOD

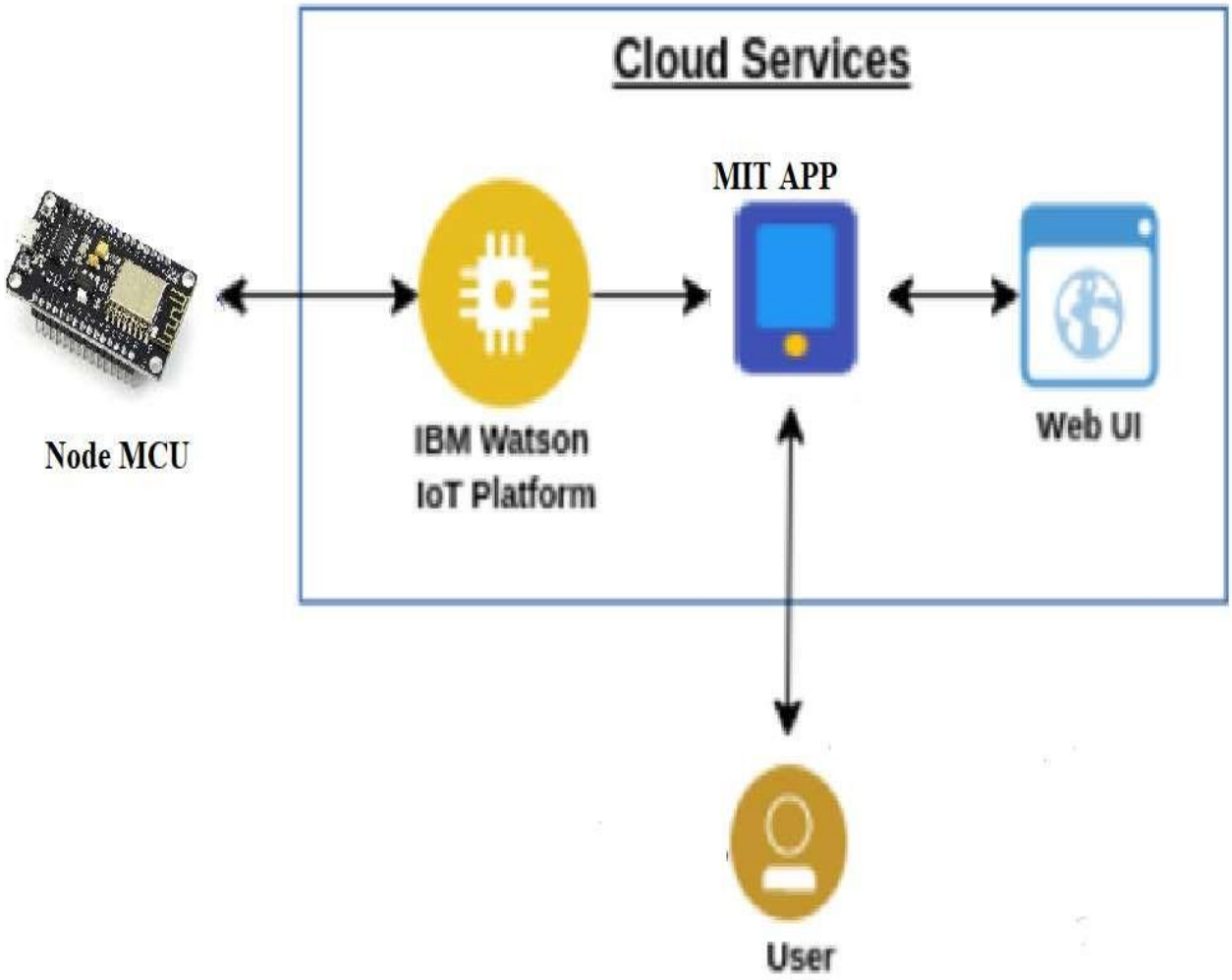
- The smart agriculture system uses the soil moisture sensor to sense the soil moisture level in the garden. The Node MCU act as a control circuit to control the overall process and cloud data will store all the data and can be recollected whenever it is needed from anywhere.
- The level of the soil moisture sensing is done by soil moisture sensor which detects the moisture level, programming for respective sensor value are programmed in the Node MCU.
- Signal from microcontroller to 12V relay are operate to on/off the motor pump.
- Water flow from the pump are depends upon the signal from Node MCU. When the soil moisture sensor senses the low level of the soil moisture then a

signal is send to the Node MCU then the controller check for the condition given in program.

- The irrigation is automated with Soil moisture sensor and the relay unit. When soil moisture level is low then a signal sends to the relay to switch ON the motor and when the soil is wet then motor is in OFF condition. Relay gives the ON/OFF condition to the motor.
- The DHT11 sensor will send the information of humidity and temperature of particular filed area to Node MCU and the data will be stored in IBM Watson cloud, through this cloud technology we can access data at any time and anywhere.

CHAPTER 4

4.1 ARCHITECURE DESIGN



CHAPTER 5

HARDWARE REQUIERMENTS & SPECIFICATION

5.1 ESP-32 CONTROLLER

The Node MCU (Node Microcontroller Unit) is open source software and hardware development environment that is built around a very inexpensive System-on-a-Chip (SoC) called the ESP 32. The ESP 32, designed and manufactured by Espressif Systems, contains all crucial elements of the modern computer: CPU, RAM, networking (Wi-Fi), and even a modern operating system and SDK.

However, as a chip, the ESP 32 is also hard to access and use. You have to solder wires, with the appropriate analog voltage, to its PINs for the simplest tasks such as powering it on or sending a keystroke to the "computer" on the chip. And, you have to program it in low-level machine instructions that can be interpreted by the chip hardware. While this level of integration is not a problem when the ESP 32 is used as an embedded controller chip in mass-produced electronics, it is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

Borrowing a page from the successful playbooks of a Raspberry Pi, the Node MCU project aims to simplify ESP 32 development. It has two key components.

1. An open source ESP 32 firmware that is built on top of the chip manufacturer's proprietary SDK. The firmware provides a simple programming environment based on eLua (embedded), which is a very simple and fast scripting language with an established developer community. For new comers, the Lua scripting language is easy to learn.

2. A board that incorporates the ESP 32 chip on a standard circuit board. The board has a built-in USB port that is already wired up with the chip, a hardware reset button, wifi antenna, LED lights, and standard-sized GPIO (General Purpose Input Output) pins that can plug into a bread board.

5.1.1 The Node MCU pin schema:

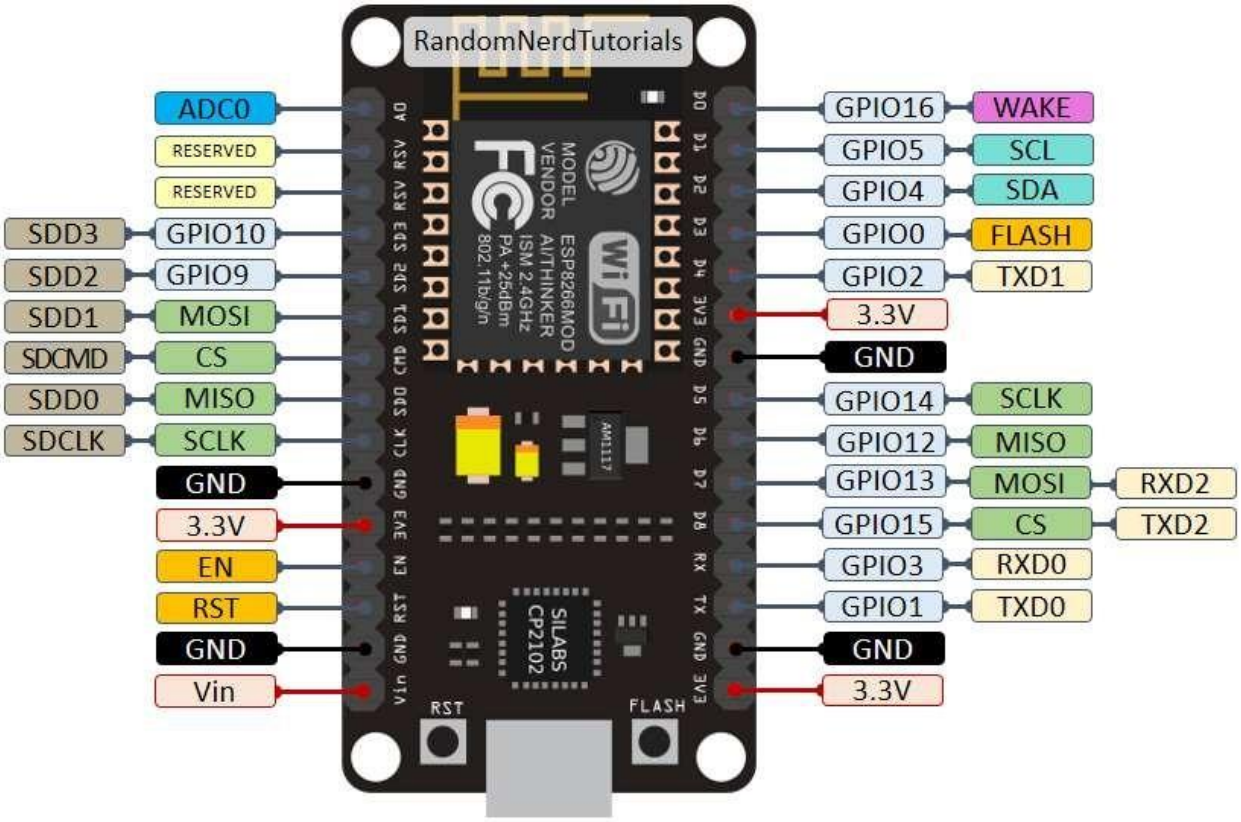


FIG NO: PIN OUT DIAGRAM

The project creates an open source hardware design and software SDK for a versatile IoT controller. Similar to Node MCU, the hardware is a microcontroller board with a ready USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards.

But unlike Node MCU, the board can have different types of CPU chips (typically an ARM or Intel x 86 chips) with memory chips, and a variety of programming environments. In fact, there is a reference design for the ESP 32 chip as well. However, the flexibility of also means significant variations across different vendors. For example, most boards do not have wifi capabilities and some even have a serial data port instead of a USB port. I feel that Node MCU provides a more consistent and accessible experience for IoT developers.

5.1.2 Node MCU programming basics

Thus far, you have seen how to load and run Lua applications on Node MCU. In this section, let's review some basic techniques for running Node MCU apps.

```
-- setup Wifi
wifi.setmode(wifi.STATION)
wifi.sta.config("SSID","password")
```

Notice that you will need to know the network name and password in order to join. A common technique is to loop through a list of known network name and password pairs if the device might be placed in several different environments.

5.1.2 Connecting to the internet

The Node MCU SDK contains an HTTP module for making HTTP requests over the internet. When you build your Node MCU firmware, you will need to select the HTTP options to include this module. The code snippet below shows how to make an HTTP GET request, and execute some code upon

completion of this request. In the callback function, the code argument is the HTTP return value (for example, 200 indicates success and 404 indicates that the URL is not accessible), and the data argument is the content in the HTTP response message.

```
geturl = "http://www.ibm.com/"
    http.get(geturl, nil, function(code, data)
        -- Turn off the red LED on NodeMCU board
end)
```

5.1.3 Accessing the GPIO pins

The General Purpose Input Output (GPIO) pins are digital pins on the Node MCU DEVKIT board. Each pin can have only two states: a low voltage state and a high-voltage state, representing 0 and 1 respectively. From the Node MCU Lua application, you can read the state from each pin, and then set the state.

```
-- Read the state of GPIO PIN #5. The val value is 0 or 1
gpio.mode(5, gpio.INPUT)
val = gpio.read(5)


-- Set the state of GPIO PIN #5 to be HIGH
gpio.mode(5, gpio.OUTPUT)
gpio.write(5,gpio.HIGH)
```

5.1.4 Reading analog signals

While the GPIO pins are digital pins, some IoT sensors send in data as analog signals. That is, the voltage of the input wire represents the data. For example, the actual voltage level from a temperature sensor might indicate the temperature reading. On the Node MCU DEVKIT board, the A0 pin can function

as an ADC (Analog to Digital Convertor) pin. When an input wire is connected to the A0, its voltage level between 0 to 3.3V will be converted to an integer number between 0 and 1024. The code snippet below shows how to read an analog value from A0 pin.

5.2 WATER PUMP

The water pump can be defined as a pump which uses the principles like mechanical as well as hydraulic throughout a piping system and to make sufficient force for its future use. They have been approximately in one structure otherwise another because of early civilization. At present these pumps are utilized within a wide range of housing, farming, municipal, and manufacturing applications.

5.2.1 Water Pump Working Principle

The working principle of a water pump mainly depends upon the positive displacement principle as well as kinetic energy to push the water. These pumps use AC power otherwise DC power for energizing the motor of the water pump whereas others can be energized other kinds of drivers like gasoline engines otherwise diesel.

The water pump is a portable device and can be applied in several household applications. These pumps are used for pumping the huge amount of water from one place to another. The main purpose of a water pump is versatile. A quality pump which can be selected carefully may be perfect for draining water from a low flooded region, refilling the swimming pool, and bathtub, circulating pesticides otherwise fertilizers.

The collection of water pumps is very large, therefore, while selecting a strong and consistent one, one should think about the requirement.

5.2.2 Centrifugal Water Pump

Centrifugal pumps are designed with a rotating impeller which can be used for supplying the water into the pump and force the discharge flow. These pumps come in several types which includes trash, submersible, and standard models. By using these pumps, all types of liquids can be pumped with low-viscosity. And also these pumps work fine with thin fluids & gives high flow rates.

Fig no: water pump

5.3 RELAY:

The Relay module allows a wide range of microcontroller such as AVR ,PIC, ARM with digital outputs to control larger loads and devices like AC or DC Motors, electromagnets, solenoids, and incandescent light bulbs. This module is designed to be integrated with 4 relays that it is capable of control 4 relays. The relay shield use one QIANJI JQC-3F high-quality relay with rated load 7A/240VAC,10A/125VAC,10A/28VDC.The relay output state is individually indicated by a light-emitting diode.

A relay is an electrical switch that opens and closes under the control of another electrical circuit. In the original form, the switch is operated by an

electromagnet to open or close one or many sets of contacts. It was invented by Joseph Henry in 1835. Because a relay is able to control an output circuit of higher power than the input circuit, it can be considered to be, in a broad sense, a form of an electrical amplifier.

Fig.no: diagram for relay

A simple electromagnetic relay, such as the one taken from a car in the first picture, is an adaptation of an electromagnet. It consists of a coil of wire surrounding a soft iron core, an iron yoke, which provides a low reluctance path for magnetic flux, a moveable iron armature, and a set, or sets, of contacts; two in the relay pictured. The armature is hinged to the yoke and mechanically linked to a moving contact or contacts. It is held in place by a spring so that when the relay is de-energized there is an air gap in the magnetic circuit. In this condition, one of the two sets of contacts in the relay pictured is closed, and the other set is open. Other relays may have more or fewer sets of contacts depending on their function. The relay in the picture also has a wire connecting the armature to the yoke. This ensures continuity of the circuit between the moving contacts on the armature, and

the circuit track on the Printed Circuit Board (PCB) via the yoke, which is soldered to the PCB.

When an electric current is passed through the coil, the resulting magnetic field attracts the armature, and the consequent movement of the movable contact or contacts either makes or breaks a connection with a fixed contact. If the set of contacts was closed when the relay was de-energized, then the movement opens the contacts and breaks the connection, and vice versa if the contacts were open. When the current to the coil is switched off, the armature is returned by a force, approximately half as strong as the magnetic force, to its relaxed position. Usually this force is provided by a spring, but gravity is also used commonly in industrial motor starters. Most relays are manufactured to operate quickly. In a low voltage application, this is to reduce noise. In a high voltage or high current application, this is to reduce arcing.

If the coil is energized with DC, a diode is frequently installed across the coil, to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate a voltage spike dangerous to circuit components. Some automotive relays already include that diode inside the relay case. Alternatively a contact protection network, consisting of a capacitor and resistor in series, may absorb the surge. If the coil is designed to be energized with AC, a small copper ring can be crimped to the end of the solenoid. This "shading ring" creates a small out-of-phase current, which increases the minimum pull on the armature during the AC cycle.

By analogy with the functions of the original electromagnetic device, a solid-state relay is made with a thyristor or other solid-state switching device. To

achieve electrical isolation an optocoupler can be used which is a light-emitting diode (LED) coupled with a photo transistor.

5.3.1 PIN DEFINITION:

- COM- Common pin
- NC- Normally Closed, in which case NC is connected with COM when INT1 is set low and disconnected when INT1 is high;
- NO- Normally Open, in which case NO is disconnected with COM1 when INT1 is set low and connected when INT1 is high.

Terminal 2-4 is similar to terminal 1, Except that the control port is INT2-4

- INT 1-4: Relay 1-4 control port

5.4 SOIL MOISTURE SENSOR:

The moisture of the soil plays an essential role in the irrigation field as well as in gardens for plants. As nutrients in the soil provide the food to the plants for their growth. Supplying water to the plants is also essential to change the temperature of the plants. The temperature of the plant can be changed with water using the method like transpiration. And plant root systems are also developed better when rising within moist soil. Extreme soil moisture levels can guide to anaerobic situations that can encourage the plant's growth as well as soil pathogens. This article discusses an overview of the soil moisture sensor, working and its applications.

What is a Soil Moisture Sensor

The soil moisture sensor is one kind of sensor used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil

moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content.

The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology.

5.4.1 Soil-moisture-sensor-device

These sensors normally used to check volumetric water content, and another group of sensors calculates a new property of moisture within soils named water potential. Generally, these sensors are named as soil water potential sensors which include gypsum blocks and tensiometer.

Soil Moisture Sensor Pin Configuration:

- The FC-28 soil moisture sensor includes 4-pins

Fig no: soil moisture sensor

Soil-moisture-sensor

- VCC pin is used for power
- A0 pin is an analog output
- D0 pin is a digital output
- GND pin is a Ground

This module also includes a potentiometer that will fix the threshold value, & the value can be evaluated by the comparator-LM393. The LED will turn on/off based on the threshold value.

5.4.2 Working Principle

This sensor mainly utilizes capacitance to gauge the water content of the soil (dielectric permittivity). The working of this sensor can be done by inserting this sensor into the earth and the status of the water content in the soil can be reported in the form of a percent.

This sensor makes it perfect to execute experiments within science courses like environmental science, agricultural science, biology, soil science, botany, and horticulture.

5.4.3 Specifications

The specification of this sensor includes the following.

- The required voltage for working is 5V
- The required current for working is <20mA
- Type of interface is analog
- The required working temperature of this sensor is 10°C~30°C

5.4.4 Soil Moisture Sensor Applications

The applications of moisture sensor include the following.

- Agriculture
- Landscape irrigation
- Research
- Simple sensors for gardeners

5.5 7805 IC Voltage Regulator:

In this tutorial, we will see about one of the most commonly used regulator ICs', the 7805 Voltage Regulator IC. A regulated power supply is very much essential for several electronic devices due to the semiconductor material employed in them have a fixed rate of current as well as voltage. The device may get damaged if there is any deviation from the fixed rate.

One of the important sources of DC Supply are Batteries. But using batteries in sensitive electronic circuits is not a good idea as batteries eventually drain out and lose their potential over time.

Also, the voltages provided by batteries are typically 1.2V, 3.7V, 9V and 12V. This is good for circuits whose voltage requirements are in that range. But, most of the TTL IC's work on 5V logic and hence we need a mechanism to provide a consistent 5V Supply.

Here comes the 7805 Voltage Regulator IC to the rescue. It is an IC in the 78XX family of linear voltage regulators that produce a regulated 5V as output.

They are available in several IC Packages like TO-220, SOT-223, TO-263 and TO-3. Out of these, the TO-220 Package is the most commonly used one (it is the one shown in the above image).

Some of the important features of the 7805 IC are as follows:

- It can deliver up to 1.5 A of current (with heat sink).
- Has both internal current limiting and thermal shutdown features.
- Requires very minimum external components to fully function.

5.5.1 Pin Diagram of 7805 Voltage Regulator IC

As mentioned earlier, 7805 is a three-terminal device with the three pins being 1. INPUT, 2. GROUND and 3. OUTPUT. The following image shows the pins on a typical 7805 IC in To-220 Package.

Fig no: Pin Diagram of 7805 Voltage Regulator IC

5.5.2 Basic Circuit of 7805

As I have previously talked about regulated power supply as a device that works on DC voltages and it can uphold its output accurately at a fixed voltage all the time even if there is a significant alteration in the DC input voltage.

As per the datasheets of 7805 IC, the basic circuit required for 7805 to work as a complete regulator is very simple. In fact, if the input supply is an unregulated DC Voltage, then all you need are two capacitors (even those are not mandatory depending on the implementation).

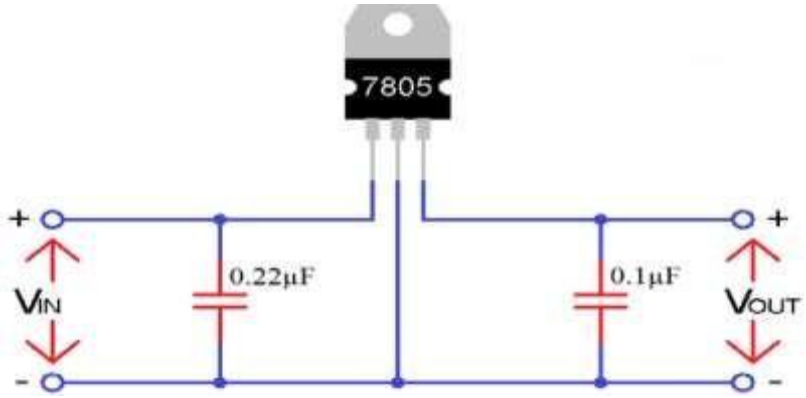


Fig no: Basic Circuit of 7805

The above circuit shows all the components required for a 7805 IC to work properly. The 0.22µF Capacitor near the input is required only if the distance between the regulator IC and the power supply filter is high. Also, the 0.1µF Capacitor near the output is optional and if used, it helps in the transient response.

In this circuit, VIN is the input voltage to the 7805 IC and the source can be from either a battery of an unregulated DC. VOUT is the output of the 7805 IC, which is a Regulated 5V.

Get Constant DC Power Supply from AC:

Although batteries can be used as input to the 7805 Voltage Regulator IC, we face certain bumps like frequent discharge of batteries and reduction of battery voltage levels over a period of time.

The best alternative to using Batteries is to provide an unregulated but rectified DC Voltage from an AC Source. Since AC Source is easily available as mains supply,

we can design a circuit to convert AC Mains to DC and provide it as input to the 7805 Voltage regulator IC.

5.5.4 Circuit Diagram

The following image shows the circuit diagram of producing a regulated 5V from AC Mains supply.

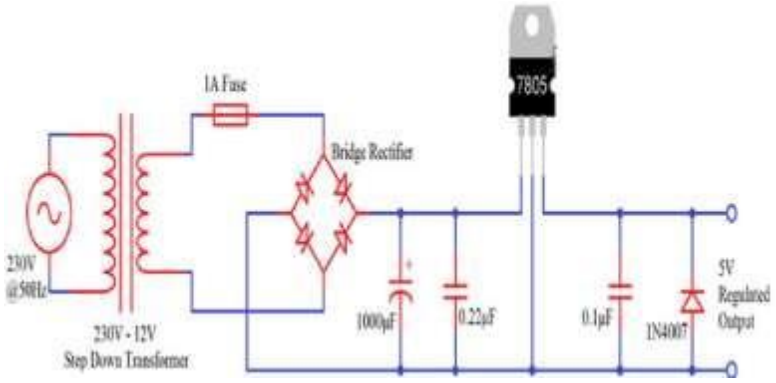


Fig no: Basic Circuit diagram of 7805

Components Required

- 230V-12V Step Down Transformer
- Bridge Rectifier (or 4 PN Diodes - 1N4007)
- 1A Fuse
- 1000µF Capacitor
- 7805 Voltage Regulator IC
- 0.22µF Capacitor
- 0.1µF Capacitor
- 1N4007 Diode

5.5.5 Working

The AC power supply from mains first gets converted into and unregulated DC and then into a constant regulated DC with the help of this circuit. The circuit

is made up of transformer, bridge rectifier made up from diodes, linear voltage regulator 7805 and capacitors.

If you observe, the working of the circuit can be divided into two parts. In the first part, the AC Mains is converted into unregulated DC and in the second part, this unregulated DC is converted into regulated 5V DC. So, let us start discussing the working with this in mind.

Initially, a 230V to 12V Step down transformer is taken and its primary is connected to mains supply. The secondary of the transformer is connected to Bridge rectifier (either a dedicated IC or a combination of 4 1N4007 Diodes can be used).

A 1A fuse is placed between the transformer and the bridge rectifier. This will limit the current drawn by the circuit to 1A. The rectified DC from the bridge rectifier is smoothened out with the help of 1000µF Capacitor.

So, the output across the 1000µF Capacitor is unregulated 12V DC. This is given as an input to the 7805 Voltage Regulator IC. 7805 IC then converts this to a regulated 5V DC and the output can be obtained at its output terminals.

5.6 DHT11 SENSOR

The **DHT11** is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

DHT11 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)

- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

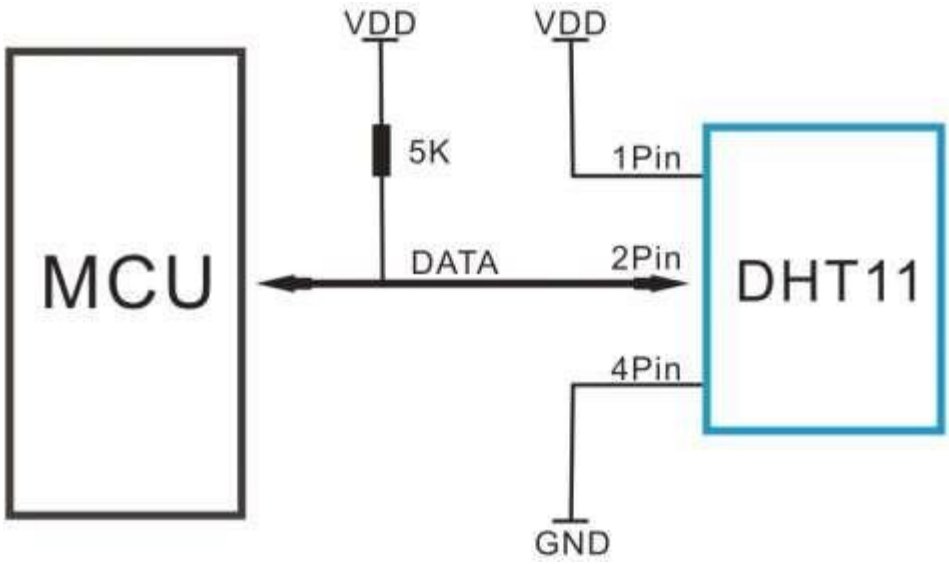
Where to use DHT11 Sensors

The **DHT11** is a commonly used **Temperature and humidity sensor.** The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%. So if you are looking to measure in this range then this sensor might be the right choice for you.

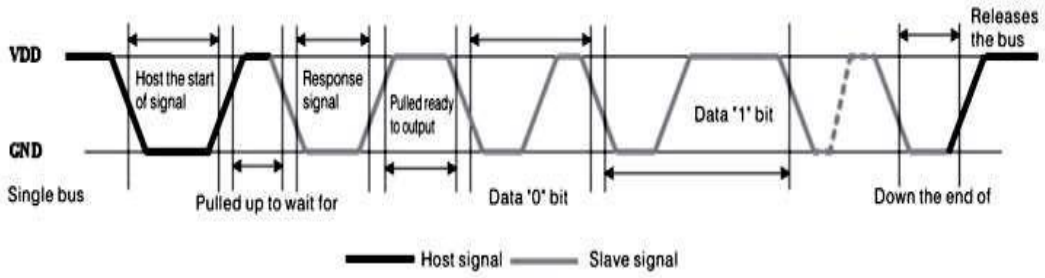
How to use DHT11 Sensor

The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown below.



As you can see the data pin is connected to an I/O pin of the MCU and a 5K pull-up resistor is used. This data pin outputs the value of both temperature and humidity as serial data. If you are trying to interface DHT11 with Arduino then there are ready-made libraries for it which will give you a quick start.

If you are trying to interface it with some other MCU, then the datasheet given below will come in handy. The output given out by the data pin will be in the order of 8bit humidity integer data + 8bit the Humidity decimal data +8 bit temperature integer data + 8bit fractional temperature data +8 bit parity bit. To request the DHT11 module to send these data the I/O pin has to be momentarily made low and then held high as shown in the timing diagram below



The duration of each host signal is explained in the DHT11 datasheet, with neat steps and illustrative timing diagrams

Applications

- Measure temperature and humidity
- Local Weather station
- Automatic climate control
- Environment monitoring

CHAPTER 6

SYSTEM SPECIFICATION

6.1 Hardware Specification

1) Node MCU
2) Moisture sensor
3) Relay
4) Water pumping motor
5) Power supply

6.2 Software Specification

- Embedded C
- Java
- IBM Watson
- Ardinuode

6.3.1 INTRODUCTION TO EMBEDDED C

Looking around, we find ourselves to be surrounded by various types of embedded systems. Be it a digital camera or a mobile phone or a washing machine, all of them has some kind of processor functioning inside it. Associated with each processor is the embedded software. If hardware forms the body of an embedded system, embedded processor acts as the brain, and embedded software forms its soul. It is the embedded software which primarily governs the functioning of embedded systems.

During infancy years of microprocessor based systems, programs were developed using assemblers and fused into the EPROMs. There used to be no mechanism to find what the program was doing. LEDs, switches, etc. were used to check correct

execution of the program. Some ‘very fortunate’ developers had In-circuit Simulators (ICEs), but they were too costly and were not quite reliable as well.

As time progressed, use of microprocessor-specific assembly-only as the programming language reduced and embedded systems moved onto C as the embedded programming language of choice. C is the most widely used programming language for embedded processors/controllers. Assembly is also used but mainly to implement those portions of the code where very high timing accuracy, code size efficiency, etc. are prime requirements.Initially C was developed by Kernighan and Ritchie to fit into the space of 8K and to write (portable) operating systems. Originally it was implemented on UNIX operating systems. As it was intended for operating systems development, it can manipulate memory addresses. Also, it allowed programmers to write very compact codes. This has given it the reputation as the language of choice for hackers too.

As assembly language programs are specific to a processor, assembly language didn’t’t offer portability across systems. To overcome this disadvantage, several high level languages, including C, came up. Some other languages like PLM, Modula-2, Pascal, etc. also came but couldn’t’t find wide acceptance. Amongst those, C got wide acceptance for not only embedded systems, but also for desktop applications. Even though C might have lost its sheen as mainstream language for general purpose applications, it still is having a strong-hold in embedded programming. Due to the wide acceptance of C in the embedded systems, various kinds of support tools like compilers & cross-compilers, ICE, etc. came up and all this facilitated development of embedded systems using C.

Subsequent sections will discuss what Embedded C is, features of C language, similarities and difference between C and embedded C, and features of embedded C programming.

6.3.4. EMBEDDED C PROGRAM

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

Embedded C use most of the syntax and semantics of standard C, e.g., main () function, variable definition, data type declaration, conditional statements (if, switch. case), loops (while, for), functions, arrays and strings, structures and union, bit operations, macros, unions, etc.

6.4 Java programming:

Java is a popular programming language.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games

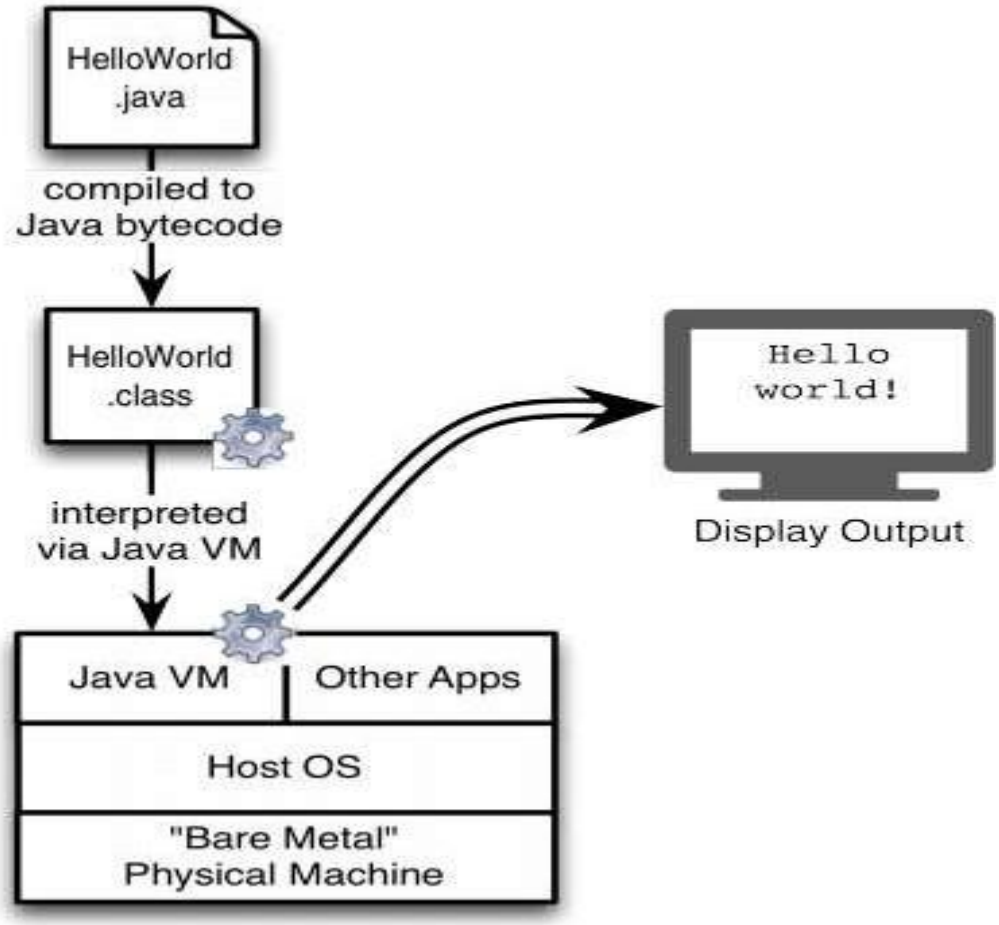
- Database connection
- And much, much more!

6.4.1 Use Java

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

6.4.2 Java Programming Language

- Java: general-purpose language: "write code once, run anywhere"
- The key: Java Virtual Machine (JVM) - Program code compiled to JVM byte code - JVM byte code interpreted on JVM
- We'll focus on Java .



6.5 ARDUINO USER INTERFACE

Arduino user interface and their functions.

- In order to verify your code, make use of this button. It will indicate any error if present in your code before uploading the sketch on Arduino board.
- After verification of your code use this button to upload the sketch.
- If you want to open a new editor window in place of your current code editing window then this button comes handy.
- To open a new file, this button is used.
- Use this button to save your sketch.

- To open a serial monitor for debugging purpose, this button is used.
- It gives you the option to add a sketch in your current project i.e. opening a new tab in your current project.

STRUCTURE OF ARDUINO SKETCH

Arduino programs made in Arduino IDE are known as sketches. So if we talk about the structure of basic Arduino sketch, then we can say that it consists of two mandatory functions known as setup () and loop () functions. Whenever we open a new window in Arduino IDE we can see that these two functions are already present in new sketch. So let's talk about the basic necessities of functions and will explain it using hello world problem as a tradition to introduce beginners with programming.

FUNCTION

In order to define a function, we need to follow these instructions as a must:

v All function should have a unique name. For example setup () is an example of unique function name. It can be any name but it must be unique i.e. not more than one function can have same name in the same sketch otherwise there will be an error. The two main programs we discussed above that constitute the body of Arduino sketch i.e. setup () and loop () functions are special functions of Arduino program.

v The function name should be followed by closing and opening parentheses (). It is not mandatory to write something in function's parentheses. Basically in function's parentheses we can pass reference to some variables whose value is going to alter. These references to variables can be by name or by value. We will discuss detail of this later somewhere.

v All functions of Arduino sketch should have a return type. Mostly return types that are used are return 1, return 0 or void. The mandatory functions of Arduino sketch i.e. setup () and loop () are having void return type.

v The body of function is enclosed in opening and closing braces i.e. { }. The action we need our function to perform is present in these braces.

So let's discuss all these requirements of function formation with some example.

HELLO WORLD SKETCH EXAMPLE:

So by following tradition set by programmers let's start with hello world program in order to introduce beginners to programming in Arduino IDE. So if we talk about this function. It only prints the text "hello world" on the screen whenever we run this program. It is mostly used to ensure that our programming environment is installed properly and is in working form. If this "hello world" program runs smoothly without any errors then we can surely say that you are ready to start with the learning of new programming language. As we know that Arduino don't come up with a screen so if we write this "hello world" program then where we will print it? Here we can make use of serial monitor window and the USB port.

RUNNING OF SKETCH:

We can run this sketch by plugging our Arduino in our PC by making use of a USB cable. After connecting it with PC, click the upload button in order to load program in our Arduino. Now after successful uploading of this code in our Arduino, open the Arduino serial monitor to see the sketch running and you can see the printed text over there.

ERROR FINDING:

The most errors or faults we can encounter are

v Programming errors

v Setup faults

v Baud rate settings fault

So let's briefly discuss about these types of errors and faults.

PROGRAMMING ERRORS:

If we do some mistake while typing the above program in Arduino IDE then we have to face error. This type of error is known as compile error. If you want to make sure that you have typed everything correctly then you can verify that by using verify button in Arduino IDE. The compile error can be check in the bottomed Arduino IDE. Suppose you missed semicolon or parentheses or stuff like that then error will be displayed in some meaningful language when you will verify your program or upload it in your Arduino board. So you can open that error and can correct it by understanding your mistake and re-run that program

SETUP FAULTS:

Now coming to the second type of error which can be caused when we have verified our program and it's not giving any sort of compile errors but still it's not getting uploaded on our Arduino board. In case of this problem you have to make sure that you have selected the right board and it can be verified under tools -> board and also make sure that serial port is also correct by checking in tools -> serial port.

BAUD RATE SETTINGS FAULT:

The last type of error that can be expected is baud rate setting fault. At this point we are expecting that there are no compile or syntax errors and our program is uploaded successfully on Arduino board but still as expected from Hello World program we are not getting hello world printed on serial monitor. So in this case the only error that can prevent text from being printed / shown in serial monitor is that the baud rate settings at bottom right of our serial monitor window is not set to the value 9600 as we have declared that rate in our setup () as 9600. So because of this mismatch of values of baud rate we cannot see hello world printed in serial monitor so it can be corrected by just changing this value in our Arduino IDE.

SETUP () AND LOOP () FUNCTIONS:

Setup (): It is always executed first. If we take example of our hello world sketch we are just setting baud rate to 9600 so our serial.begin (9600) will run first and then our printIn () function. This function is only executed once every time that sketch is run and it will start it's execution of instruction present in it once it has been programmed in our Arduino board. It can be re-run by using reset button or by connecting Arduino again by disconnecting it

Loop (): Statements present in loop () function will continuously run from top to bottom and then back to top. In our Hello World sketch we have nothing in loop function so our sketch will end up there. But even if we have nothing to do in loop () we still have to use this function in our sketch otherwise our microcontroller will start executing whatever it finds next in memory. So in order to prevent this issue we have to define loop function in our sketch. The operational functionality of Arduino takes place in loop () function.

6.6 IBM Watson

The IBM Watson Studio learning path demonstrates various ways of using IBM Watson Studio to predict customer churn. It ranges from a semi-automated approach using the AutoAI Experiment tool to a diagrammatic approach using SPSS Modeler Flows to a fully programmed style using Jupyter notebooks for Python.

All of the tutorials within this learning path follow the main steps of methods for data science (and data mining), such as Cross Industry Standard Process for Data Mining (CRISP-DM) and the IBM Data Science Methodology. The tutorials focus on tasks for data understanding, data preparation, modeling, evaluation, and deployment of a machine learning model for predictive analytics. They use as a

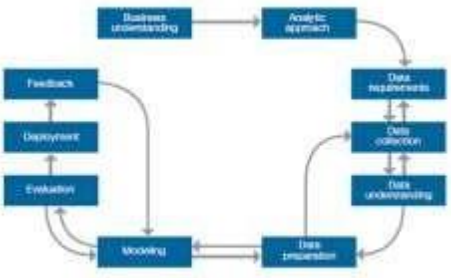
basis a data set and notebook for customer churn available on Kaggle, and then demonstrate alternative ways of solving the same problem by using AutoAI, the SPSS Modeler, and the IBM Watson Machine Learning service provided by the IBM Watson Studio suite of tools. The learning path explains the use of the profiling tool and the dashboards of IBM Watson Studio to support data understanding, as well as the Refine tool to solve straightforward data preparation and transformation tasks.

Data Science Methodology

IBM has defined a lightweight IBM Cloud Garage Method that includes a process model to map individual technology components to the reference architecture. This method does not include any requirement engineering or design thinking tasks. Because it can be hard to initially define the architecture of a project, this method supports architectural changes during the process model.

Each stage plays a vital role in the context of the overall methodology. At a certain level of abstraction, it can be seen as a refinement of the workflow outlined by the CRISP-DM method for data mining.

Cross Industry Standard Process for Data Mining



IBM Data Science Methodology

According to both methodologies, every project starts with *Business understanding*, where the problem and objectives are defined. This is followed in the IBM Data Science Method by the *Analytical approach* phase, where the data scientist can define the approach to solving the problem. The IBM Data Science Method then continues with three phases called *Data requirements*, *Data collection*, and *Data understanding*, which in CRISP-DM are presented by a single *Data understanding* phase.

After the data scientist has an understanding of the data and has sufficient data to get started, they move to the *Data preparation* phase. This phase is usually very time consuming. A data scientist spends about 80% of their time in this phase, performing tasks such as data cleansing and feature engineering. The term "data wrangling" is often used in this context. During and after cleansing the data, the data scientist generally performs exploration, such as descriptive statistics to get an overall feel for the data, and clustering to look at the relationships and latent structure of the data. This process is often iterated several times until the data scientist is satisfied with their data set.

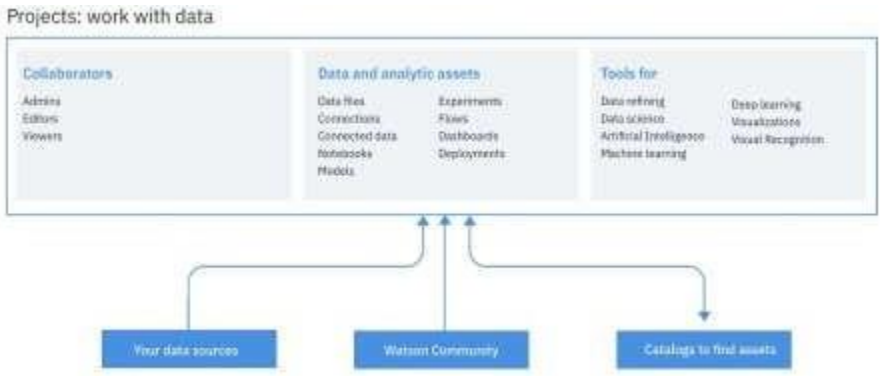
The model training stage is where machine learning is used in building a predictive model. The model is trained and then evaluated by statistical measures such as prediction accuracy, sensitivity, and specificity. After the model is deemed sufficient, it is deployed and used for scoring on unseen data. The IBM Data Science Methodology adds an additional *Feedback* stage for obtaining feedback from using the model, which is then used to improve the model. Both methods are highly iterative by nature.

In this learning path, we will focus on the phases starting with data understanding, and then continue on to preparing the data, building a model, evaluating the model,

and then deploying and testing the model. The purpose is to develop models to predict customer churn. Aspects related to analyzing the causes of these churns to improve the business is out of the scope of this learning path. This means that we will be working with various kinds of classification models that can, given an observation of a customer defined by a set of features, give a prediction as to whether this specific client is at risk of churning.

6.7 IBM Watson Studio

For all of the tasks we use IBM Watson Studio. It gives you the **environment** and **tools** to solve business problems by collaboratively working with data. You can choose the tools needed to analyze and visualize data; to cleanse and shape the data; to ingest streaming data; or to create, train, and deploy machine learning models.



MIT APP INVENTOR

MIT App Inventor is an online platform designed to teach computational thinking concepts through development of mobile applications. Students create applications by dragging and dropping components into a design view and using a visual blocks language to program application behavior. The smartphone is an information nexus

in today's digital age, with access to a nearly infinite supply of content on the web, coupled with rich sensors and personal data. However, people have difficulty harnessing the full power of these ubiquitous devices for themselves and their communities. Most smartphone users consume technology without being able to produce it, even though local problems can often be solved with mobile devices. How then might they learn to leverage smartphone capabilities to solve real-world, everyday problems? MIT App Inventor is designed to democratize this technology and is used as a tool for learning computational thinking in a variety of educational contexts, teaching people to build apps to solve problems in their communities. In the design of MIT App Inventor, introducing mobile app development in educational contexts was a central goal. Prior to its release, most development environments for mobile applications were clunky, only accessible with expertise in systems level or embedded programming, or both. Even with Google's Android operating system and the Java programming language, designing the user interface was a complex task. Further, use of the platform required familiarity with Java syntax and semantics, and the ability to debug Java compilation errors (e.g., misspelled variables or misplaced semicolons) for success. These challenges presented barriers to entry for individuals not versed in computer science, App Inventor's target demographic. We briefly highlight and discuss design goals for the App Inventor project, specifically, the use of *components* to abstract some of the complexity of platform behavior, and the use of *blocks* to eliminate complexity of the underlying programming language. These goals can be further explained as aligning the visual language to the mental models of young developers and enabling exploration through fast, iterative design.

Benefits of MIT app inventor:

- Everything is done through a select and drop manner. This means we can select a particular chunk of code and drop in our code. Hence, no typing.
- Easy to test your app. We can check the app developed on desktop or laptop with the app inventor application on our mobile phones.
- MIT provides the user with some basic lessons which help in building that apps and that helps in a proper understanding of how the MIT app inventor platform works for the user.
- Useful for novices.
- Power of native apps with a simple UI.

CHAPTER 7

RESULTS

CHAPTER 8

CONCLUSION

By implementing above smart agriculture systems there are various benefits. By continuously monitoring the status of the soil in the field, we can control the flow of water and thereby reduce the wastage. Conservation of water and labor: Since the systems are automatic, they do not require continuous monitoring by labor. The design is low cost, small size, robust and highly versatile. This system avoids over irrigation, under irrigation, top soil erosion and reduce the wastage of water. By implementing this system, agricultural, horticultural lands, parks, gardens, golf courses can be irrigated. The temperature and humidity of particular place can also be identified which a necessary part of agriculture area.

CHAPTER 9

REFERENCE

[1] M. Wenz and H. Worn, "Event-based Production Rules for Data Aggregation in Wireless Sensor Networks," 2019 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Heidelberg, 2006, pp. 59-64.doi: 10.1109/MFI.2006.265587

[2] K. Doddapaneni, F. A. Omondi, E. Ever, P. Shah, O. Gemikonakli and R. Gagliardi, "Deployment Challenges and Developments in Wireless Sensor Networks Clustering," Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on, Victoria, BC, 2019, pp. 227-232. doi: 10.1109/WAINA.2014.46

[3] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," in IEEE Communications Magazine, vol. 40, no. 8, pp. 102-114, Aug 2019.doi: 10.1109/MCOM.2002.1024422

[4] M. Nesa Sudha, M.L. Valarmathi, Anni Susan Babu, Energy efficient data transmission in automatic irrigation system using wireless sensor networks, Computers and Electronics in Agriculture, Volume 78, Issue 2, September 2020, Pages 215-221, ISSN 0168-1699, http://dx.doi.org/10.1016/j.compag.2011.07.009

[5] A. K. Singh, J. Tiwari, A. Yadav and S. Gupta, "Real Time Autonomous Irrigation Module Design," Computational Intelligence and Communication Networks (CICN), 2021 International Conference on, Bhopal, 2014, pp. 1209-1213. doi: 10.1109/CICN.2014.251

[6] M. Abu-El Magad, E. Kamel and K. Kamel, "Economic assessment of an irrigation canal automation and control project," Intelligent Control, 2019. Proceedings of the 2019 IEEE International Symposium on, Istanbul, 2019, pp. 205-208. doi: 10.1109/ISIC.1997.626454

[7] Y. X. Dong, C. P. Chang and F. D. Li, "The influence of groundwater quality on irrigation: A case study in Dezhou, Shandong Province," Multimedia Technology (ICMT), 2020 International Conference on, Hangzhou, 2011, pp. 1812-1815. doi: 10.1109/ICMT.2011.6003275

[8] J. Gutiérrez, J. F. Villa-Medina, A. NietoGaribay and M. Á Porta-Gándara, "Automated Irrigation System Using a Wireless Sensor Network and GPRS Module," in IEEE Transactions on Instrumentation and Measurement, vol. 63, no. 1, pp. 166-176, Jan. 2020.

[9] A. Nayak, G. Prakash and A. Rao, "Harnessing wind energy to power sensor networks for agriculture," Advances in Energy Conversion Technologies (ICAECT), 2020 International Conference on, Manipal, 2020, pp. 221-226.

[10] J. J. Baviskar, A. Y. Mulla, A. J. Baviskar, K. Dsouza and M. Khan, "Designing of mobile controlled automatic Interactive Voice Response irrigation system," Recent Advances and Innovations in Engineering (ICRAIE), 2020, Jaipur, 2020, pp. 1-6. doi: 10.1109/ICRAIE.2014.6909238

[11] S. Khriji, D. E. Houssaini, M. W. Jmal, C. Viehweger, M. Abid and O. Kanoun, "Precision irrigation based on wireless sensor network," in IET Science, Measurement & Technology, vol. 8, no. 3, pp. 98-106, May 2021.

[12] R. K. Kodali and L. Boppana, "Irrigation with grid architecture sensor network," Advances in Computing, Communications and Informatics (ICACCI, 2021 International Conference on, New Delhi, 2021, pp. 2110-2116.

[13] R. G. Vishwakarma and V. Choudhary, "Wireless solution for irrigation in agriculture," Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, Thuckafay, 2020, pp. 61-63.

[14] V. Ahmed and S. A. Ladhake, "Design of Ultra Low Cost Cell Phone Based Embedded System for Irrigation," Machine Vision and Human-Machine Interface (MVHI), 2021 International Conference on, Kaifeng, China, 2021, pp. 718-721.

[15] Y. Jiber, H. Harroud and A. Karmouch, "Precision agriculture monitoring framework based on WSN," 2011 7th International Wireless Communications and Mobile Computing Conference, Istanbul, 2019.

[16] H. N. and K. Kant, "Local weather interpolation using remote AWS data with error corrections using sparse WSN for automated irrigation for Indian farming," Contemporary Computing (IC3), 2019 Seventh International Conference on, Noida, 2019, pp. 478-483. doi: 10.1109/IC3.2014.6897220

[17] M. Dursun and S. Ozden, "A prototype of PC based remote control of irrigation," 2020 International Conference on Environmental Engineering and Applications, Singapore, 2010, pp. 255-258.

[18] S. K. Lenka and A. G. Mohapatra, "Gradient Descent with Momentum based Neural Network Pattern Classification for the Prediction of Soil Moisture Content in Precision Agriculture," 2021 IEEE International Symposium on Nanoelectronic and Information Systems, Indore, 2021, pp. 63-66. doi: 10.1109/iNIS.2015.56

[19] J. Haule and K. Michael, "Deployment of wireless sensor networks (WSN) in automated irrigation management and scheduling systems: a review," Science, Computing and Telecommunications (PACT), 2014 Pan African Conference on, Arusha, 2021, pp. 86-91. doi: 10.1109/SCAT.2014.7055144

[20] M. H. Pham, S. K. Saleem and N. Okello, "Real-time optimization of irrigation scheduling in agriculture," 2021 25th Chinese Control and Decision Conference (CCDC), Guiyang, 2021, pp. 4435-4439. doi: 10.1109/CCDC.2013.6561733.