CONTENT

Abstract

The IoT Enabled Water Distribution system is proposed for managing, monitoring and efficient distribution of water. With the increase in population, the urban residential areas were increased because of this reasons water has become a crucial problem which affects the problem of water distribution, interrupted water supply, water conservation, water consumption. In order to fulfill the water demand of the continuously growing population, it is essential to provide the sufficient and uniform quantity of water to each and every area. IoT Enabled Water Distribution System as the name says it all about managing water supply throughout the scale, including small societies, townships, urban infrastructure and also for irrigational use. The water supply with continuous monitoring makes a proper distribution so that, we can have a record of the available amount of water in tanks, flow rate, abnormality in the distribution line. IoT is a network of physical objects which are embedded with electronic devices, sensors, software, and network connectivity. Monitoring can be done from anywhere as a central office. As per the scarcity of the water, it will be indicated by sending short message service notification as well as notification of water supply timing to societies.

There are many problems in existing way of water distribution, a few of them are mentioned below,

Improper Management At present the municipal corporation system is supplying the water without any proper management that is the people who are having the water resources they get more amount of water and for others, the water is not sufficient. So, for supplying the equal amount of water to each and every area we have to maintain proper management for the distribution of water.

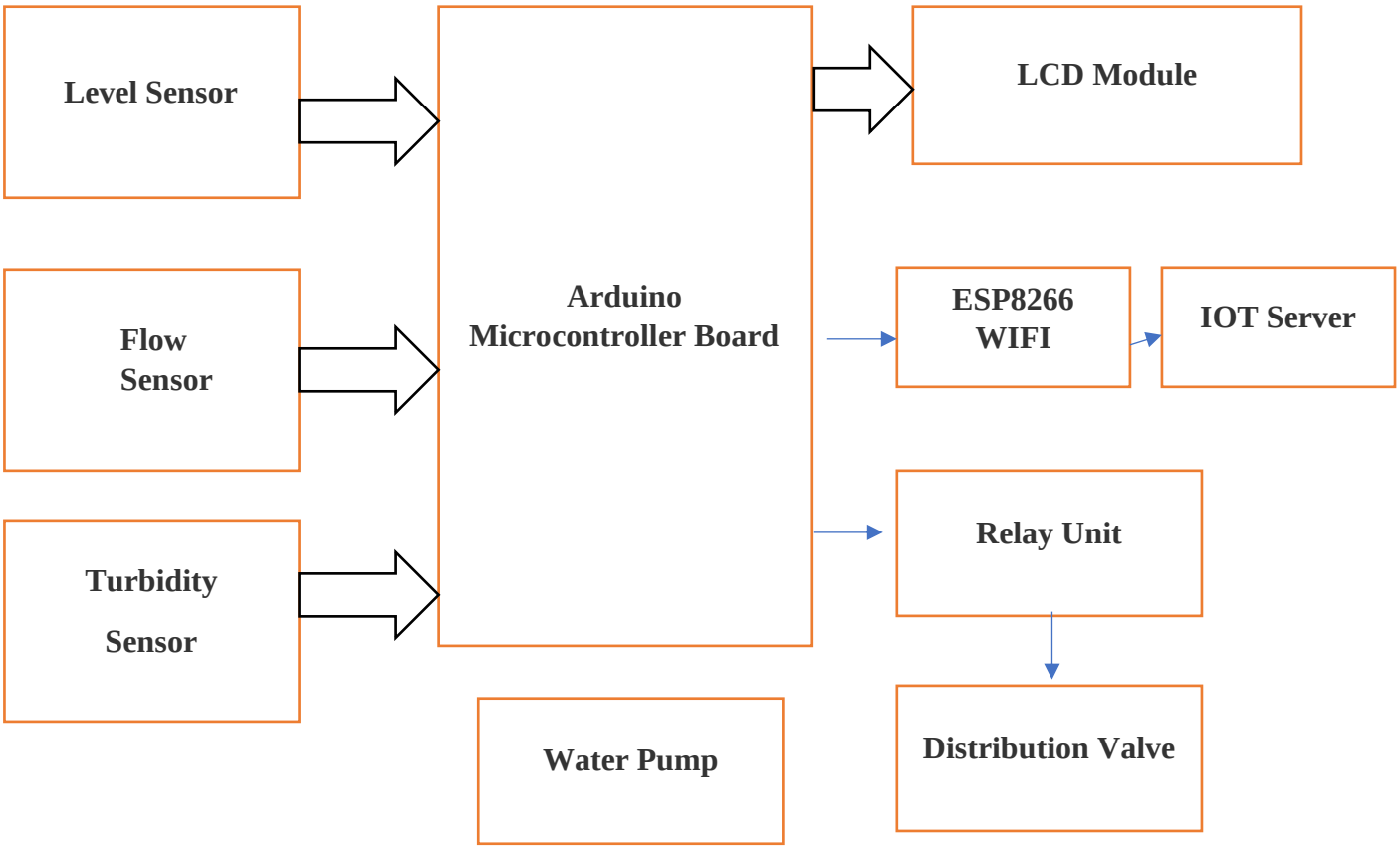
Wasting of water Now a day's water is wasted in many ways due to the irresponsibility of the people because we have more amount of water than we need. It mostly takes place in the areas where the water resources are mostly available. So, due to our negligence, we are wasting the water at the public taps and at our home, etc.

Water Leakage The leakage of water is the most common problem in the existing system. While the water is supplied through the pipelines from the municipal tank, due to the breaking of the pipes or improper installation the water is leaked and it is very difficult to easily identify the leakages by the maintenance people. So, more amount of water is wasted unnecessarily.

The main objective of our methodology is "Providing efficient distribution, management, and conservation of water to sustain for the future generation". This methodology is used to avoid all the problems mentioned in the existing system and also it has more advanced features. In

this method, we are adding some additional circuitry to the existing system for effective usage of water. By adding our circuitry to the existing system, we have to face those two challenges mentioned above. In this methodology, we are using the latest technology called as Interne of Things. In this, the entire circuitry is controlled using the Internet as mentioned in the title. So, human interference is reduced by using this technology.

Block Diagram:



Float Sensors

Float Sensor is an electrical ON/OFF Switch, which operates automatically when liquid level goes up or down with respect to specified level. The Signal thus available from the Float Sensor can be utilized for control of a Motor Pump or an allied electrical element like Solenoid, Lamps, and Relays etc.

Float Sensors contain hermetical sealed Reed Switch in the stem and a permanent Magnet in the Float. As the Float rises or falls with the level of liquid the Reed Switch is activated by Magnet in the Float.

Hamilton Float Sensors are available in wide range according to operation / material/ mounting methods to suit variety of individual application. These Float sensors are rugged, accurate and reliable in operation.

Fig Float Sensor

The major applications of Magnetic Float Switch Applications are as under:

Automotive

Wind shield washer, Tank brake, Oil Tank, Fuel Tank.

Appliances

Vending Machine, Water Purifier, Humidifier, Dish Washer, Coffee Machines and Washing Machine

Others

Power Generators, Oil Pressure Machine, Solar Systems, Boiler, Water Level Controllers
and Indicators for Overhead Water Tanks. s : Power Generators, Oil Pressure Machine, Solar

Systems, Boiler, Water Level Controllers and Indicators for Overhead Water Tanks, Air conditioner Chiller Units.

Some of the Salient Features are:

- Suitable for Any type of Pump Controller
- When Water Tank Level goes down, pump turns ON automatically
- When Water Tank gets empty, pump turns OFF automatically
- Rust-free Magnetic Sensors
- Manufactured in Electric Shock-proof Plastic body
- Detects water level at overhead water tank
- Hussle-free simple Installation
- Operation based on 220V AC as well as 12V DC
- Maintenance free product

Working Principle:

Most float switches utilize a magnetic reed switch to open or close the circuit. The reed is encased in a glass tube, which is cemented into a plastic or stainless steel stem with epoxy. The illustration to the left demonstrates how a magnet can be used to open or close a circuit by moving it closer to or farther away from a reed switch.

When the magnet comes close to the two contacts, they draw together and touch, allowing current to pass through. When the magnet is moved away, the contacts demagnetize and separate, breaking the circuit.

In a float switch, the magnetic reed switch is hermetically sealed in a stem, most often made from plastic or stainless steel. The float encases a sealed magnet, which moves up and down the length of the stem as a fluid level rises and falls.

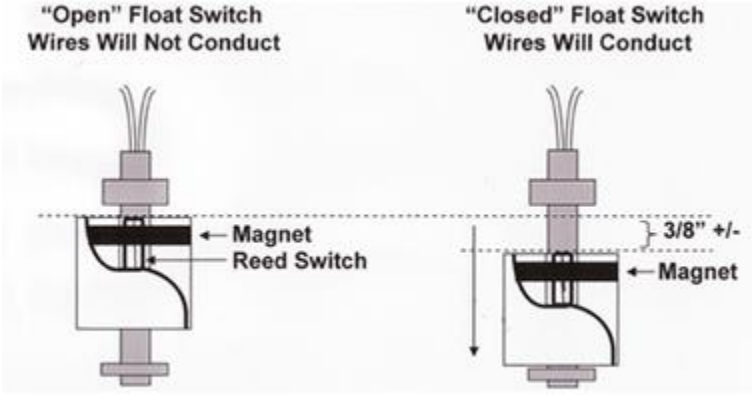


Fig Internal Diagram of Float Switch

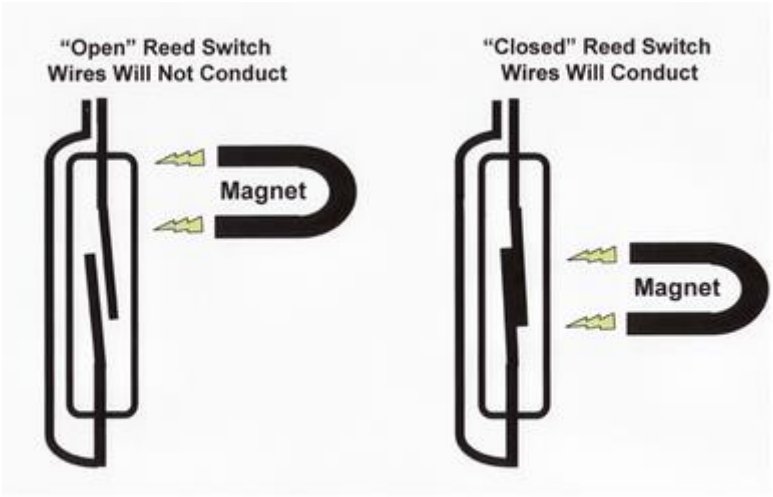


Fig Float Switch action towards magnet

Hall-Effect Flow Sensor

Effective water management involves supplying water according to the real requirement, and thus measuring water is very essential step in water management systems. There are many water flow measurement techniques as well as different types of water flow meters used to measure the volume of water flow in pipelines but these all are too costly. This article describes ideas for design and development of low cost automatic water flow meters, with the help of readily-available and low-cost water flow sensors.

YF-S201 Hall-Effect Flow Sensor

Accurate flow measurement is an essential step both in the terms of qualitative and economic points of view. Flow meters have proven excellent devices for measuring water flow, and now it is very easy to build a water management system using the renowned water flow sensor

YF-S201

. This sensor sits in line with the water line and contains a pinwheel sensor to measure how much water has moved through it. There is an integrated magnetic Hall-Effect sensor that outputs an electrical pulse with every revolution. The “

YFS201

Hall Effect Water Flow Sensor” comes with three wires: Red/VCC (5-24V DC Input), Black/GND (0V) and Yellow/OUT (Pulse Output). By counting the pulses from the output of the sensor, we can easily calculate the water flow rate (in litre/hour – L/hr) using a suitable conversion formula.

Fig Flow Sensor

Fig Internal view of a Flow Sensor

Hardware

Connecting the water flow sensor to arduino requires minimal interconnection. Connect the VCC (Red) and GND (Black) wires of the water flow Sensor to the 5v and Gnd of Arduino, and link Pulse Output (Yellow) wire of the water flow sensor to Arduino's digital pin 2. Note that the water flow sensor is not a power-hungry type; it draws a maximum of 15-20mA at 5V DC input!

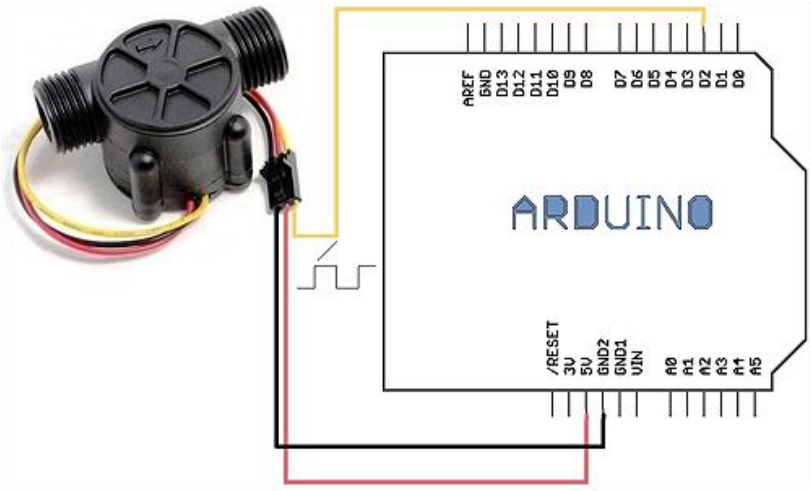


Fig Interfacing Flow sensor with Arduino

Software Preparation

The Arduino Sketch (code) uses the external interrupt (int 0) on Arduino's digital pin 2 (D2). This is used to read the output pulses coming from the water flow sensor. When Arduino detects the pulse, it immediately triggers the pulseCounter() function.

Applications of Water Flow Sensor

Water flow sensors can measure the rate of flow of water either by measuring velocity or displacement. These sensors can also measure the flow of water like fluids such as measuring milk in a dairy industry etc…

There are various types of water flow sensors available based on their diameter and method of measuring. A cost-effective and most commonly used water flow sensor is Paddlewheel sensor. It can be used with water-like fluids.

For the type of applications where a straight pipe is not available for inlet, Positive displacement flow meter is used. This type of water flow sensor can be used for viscous liquids also.

For working with dirty water and wastewater which may be conductive, Magnetic flow meter is used. For applications such as sewage water, slurries, and other dirty liquids Ultrasonic flow meters are used.

The LCD display is used to display the measurements. The magnetic hall effect water flow sensor outputs a pulse of every revolution of the rotor. The hall effect sensor present in the device is sealed from water to keep it safe and dry.

Turbidity sensor

Introduction

Turbidity is the cloudiness or haziness of a fluid caused by large numbers of individual particles that are generally invisible to the naked eye, similar to smoke in the air. The measurement of turbidity is a key test of water quality.

Turbidity is caused by particles **suspended** or **dissolved** in water that scatter light making the water appear **cloudy** or **murky**. Particulate matter can include sediment, especially clay and silt, fine organic and inorganic matter, soluble colored organic compounds, algae, and other microscopic organisms.

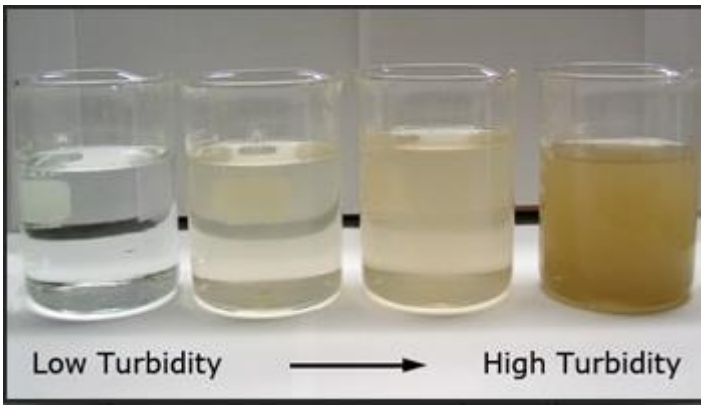


Fig Various level of turbidity in water

Impact of Turbidity

High turbidity can significantly reduce the **aesthetic quality** of lakes and streams. It can increase the cost of water treatment for drinking and food processing. It can **harm fish** and other **aquatic life** by reducing food supplies, degrading spawning beds, and affecting gill function.

Turbidity Sources

Sediment often tops the list of substances or pollutants causing turbidity. Natural sources can include erosion from upland, riparian, stream bank, and stream channel areas. Algae that grow with nourishment from nutrients entering the stream through leaf decomposition or other naturally occurring decomposition processes can also be a source of turbidity.

Stream channel movement can also release sediment. Organic matter from sewage discharges, especially during treatment plant bypasses, can contribute to turbidity. Human activities that disturb the land, such as construction, mining, and agriculture, can lead to high sediment levels entering water bodies during rainstorms due to storm water runoff.

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.

The unit of measurement is called a **Nephelometric Turbidity Unit (NTU)**, which comes in several variations. The greater the scattering of light, the higher the turbidity. Low turbidity values indicate high water clarity; high values indicate low water clarity.

Interface Description:

1. "D/A" Output Signal Switch
2. "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity
3. "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer
4. Threshold Potentiometer: you can change the trigger condition by adjusting the threshold potentiometer in digital signal mode.

Fig Turbidity Sensor

The turbidity sensor detects water quality by measuring the levels of turbidity. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate, which changes with the amount of total

suspended solids (TSS) in water. As the TTS increases, the liquid turbidity level increases.

Turbidity sensors are used to measure water quality in rivers and streams, wastewater and effluent measurements, control instrumentation for settling ponds, sediment transport research and laboratory measurements.

This sensor provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

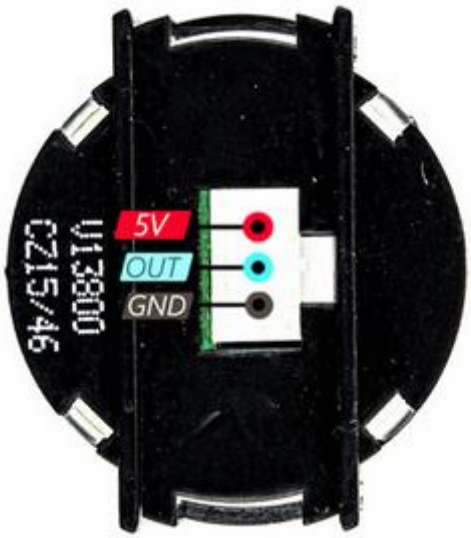


Fig Turbidity Sensor Pinout

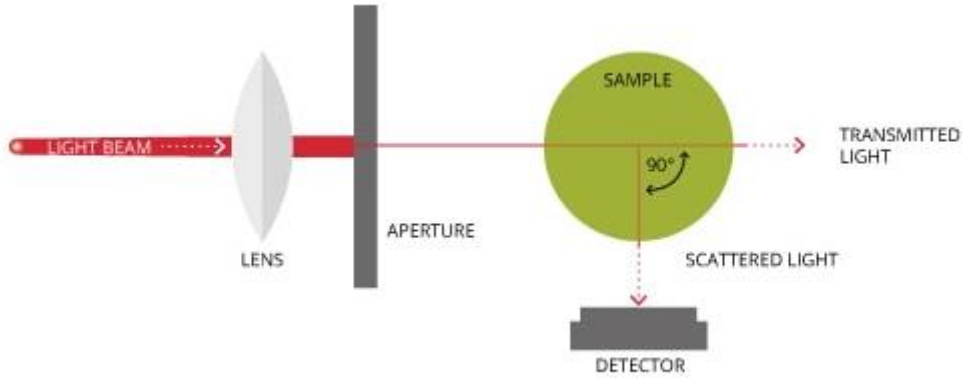


Fig Turbidity Working Principle

The sensor operates on the principle that when the light is passed through a sample of water, the amount of light transmitted through the sample is dependent on the amount of soil in the water. As the soil level increases, the amount of transmitted light decreases. The turbidity sensor measures the amount of transmitted light to determine the turbidity of the wash water.

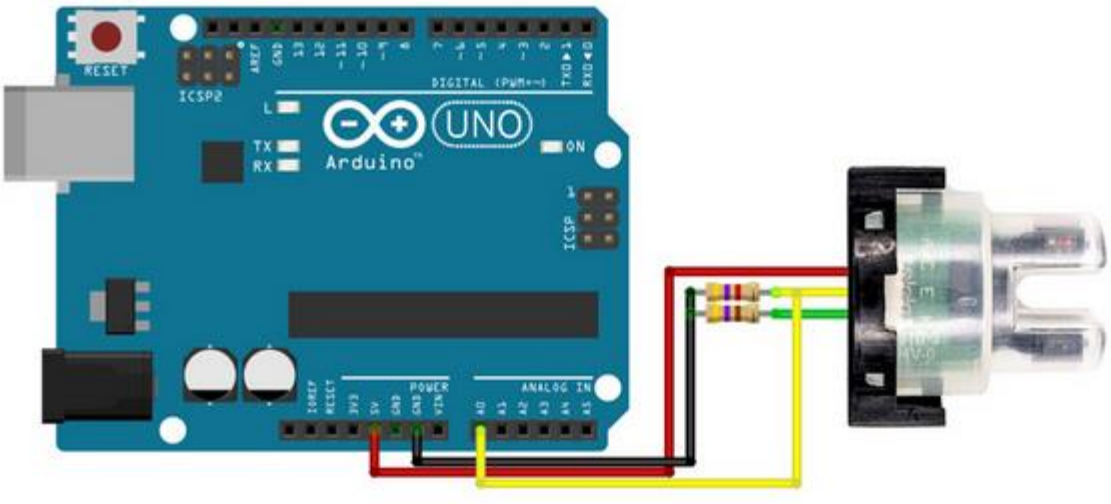


Fig Arduino Turbidity Sensor Connection

Connect the VCC of the Turbidity Sensor with Arduino 5V, GND to GND & Analog Output to Arduino A0

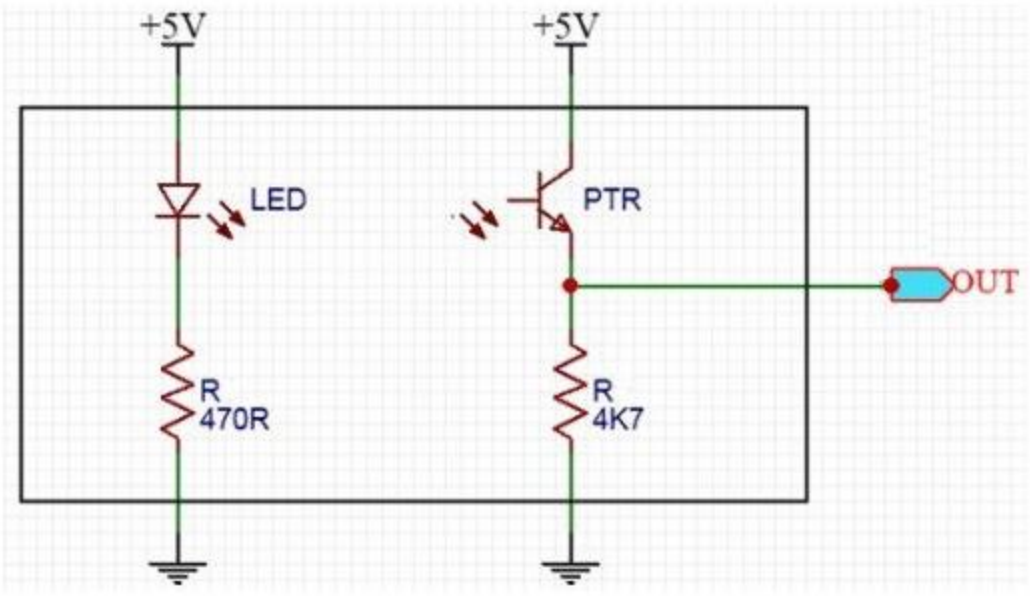


Fig Turbidity Sensor Circuit Connection

Specification

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance:100M (Min) Output Method:
- Analog output: 0-4.5V
- Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5 ~90
- Storage Temperature: -10 ~90

SOLENOID VALVE

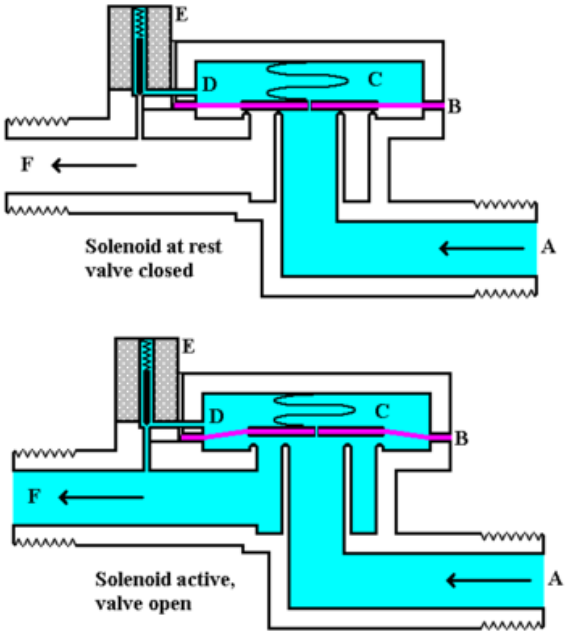
A solenoid valve is an electromechanical valve for use with liquid or gas. The valve is controlled by an electric current through a solenoid: in the case of a two-port valve the flow is switched on or off; in the case of a three-port valve, the outflow is switched between the two outlet ports. Multiple solenoid valves can be placed together on a manifold.

Solenoid valves are the most frequently used control elements in fluidics. Their tasks are to shut off, release, dose, distribute or mix fluids. They are found in many application areas. Solenoids offer fast and safe switching, high reliability, long service life, good medium compatibility of the materials used, low control power and compact design.

Besides the plunger-type actuator which is used most frequently, pivoted-armature actuators and rocker actuators are also used.

A solenoid valve has two main parts: the solenoid and the valve. The solenoid converts electrical energy into mechanical energy which, in turn, opens or closes the valve mechanically. A direct acting valve has only a small flow circuit, shown within section E of this diagram. This diaphragm piloted valve multiplies this small flow by using it to control the flow through a much larger orifice.

Solenoid valves may use metal seals or rubber seals, and may also have electrical interfaces to allow for easy control. A spring may be used to hold the valve opened or closed while the valve is not activated.



The diagram to the right shows the design of a basic valve. At the top figure is the valve in its closed state. The water under pressure enters at A. B is an elastic diaphragm and above it is a weak spring pushing it down. The function of this spring is irrelevant for now as the valve would stay closed even without it. The diaphragm has a pinhole through its center which allows a very small amount of water to flow through it. This water fills the cavity C on the other side of the diaphragm so that pressure is equal on both sides of the diaphragm, however the compressed spring supplies a net downward force. The spring is weak and is only able to close the inlet because water pressure is equalised on both sides of the diaphram.

In the previous configuration the small conduit D was blocked by a pin which is the armature of the solenoid E and which is pushed down by a spring. If the solenoid is activated by drawing the pin upwards via magnetic force from the solenoid current, the water in chamber C will flow through this conduit D to the output side of the valve. The pressure in chamber C will drop and the incoming pressure will lift the diaphragm thus opening the main valve. Water now flows directly from A to F.

When the solenoid is again deactivated and the conduit D is closed again, the spring needs very little force to push the diaphragm down again and the main valve closes. In practice there is often no separate spring, the elastomer diaphragm is moulded so that it functions as its own spring, preferring to be in the closed shape.

From this explanation it can be seen that this type of valve relies on a differential of pressure between input and output as the pressure at the input must always be greater than the

pressure at the output for it to work. Should the pressure at the output, for any reason, rise above that of the input then the valve would open regardless of the state of the solenoid and pilot valve.

In some solenoid valves the solenoid acts directly on the main valve. Others use a small, complete solenoid valve, known as a pilot, to actuate a larger valve. While the second type is actually a solenoid valve combined with a pneumatically actuated valve, they are sold and packaged as a single unit referred to as a solenoid valve. Piloted valves require much less power to control, but they are noticeably slower. Piloted solenoids usually need full power at all times to open and stay open, where a direct acting solenoid may only need full power for a short period of time to open it, and only low power to hold it.

Types

Many variations are possible on the basic, one way, one solenoid valve described above:

- one or two solenoid valves;
- direct current or alternating current powered;
- different number of ways and positions;

Common uses

Solenoid valves are used in fluid power pneumatic and hydraulic systems, to control cylinders, fluid power motors or larger industrial valves. Automatic irrigation sprinkler systems also use solenoid valves with an automatic controller. Domestic washing machines and dishwashers use solenoid valves to control water entry to the machine. In the paintball industry, solenoid valves are usually referred to simply as "solenoids." They are commonly used to control a larger valve used to control the propellant (usually compressed air or CO_2).Solenoid valves are used in dental chairs to control air flow. In the industry,"solenoid" may also refer to an electromechanical solenoid commonly used to actuate a sear.

Besides controlling the flow of air and fluids solenoids are used in pharmacology experiments, especially for patch-clamp, which can control the application of agonist or antagonist.

ARDUINO UNO

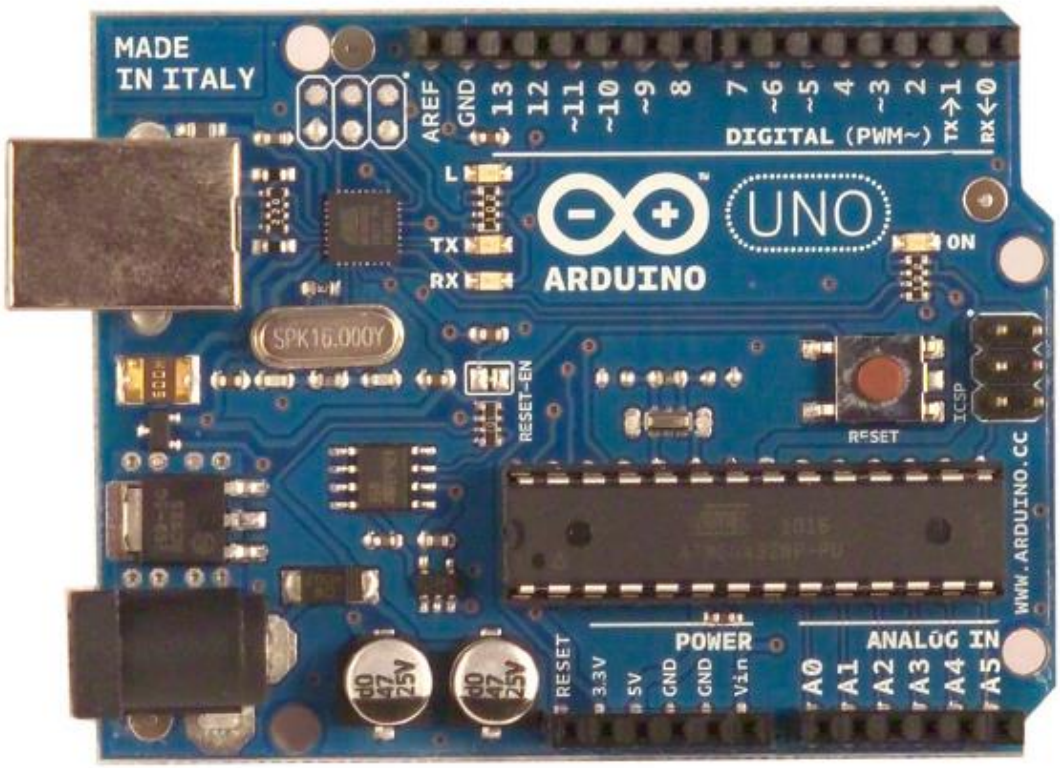


Fig1.1:Arduino UNO Board

Arduino is a single-board microcontroller to make using electronics in multidisciplinary projects more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Arduino boards can be purchased pre-assembled or as do-it-yourself kits. Hardware design information is available for those who would like to assemble an Arduino by hand. It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced.

Histroy

Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. At that time program students used a "BASIC Stamp" at a cost of $100, considered expensive for students. Massimo Banzi, one of the founders, taught at Ivrea.[2]

A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community. The school eventually closed down, so these researchers, one of them David Cuartielles, promoted the idea.[2]

A 3rd-party Arduino board with a RS-232 serial interface (upper left) and an Atmel ATmega8 microcontroller chip (black, lower right); the 14 digital I/O pins are located at the top and the six analog input pins at the lower right.

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits. An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as shields. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to

specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

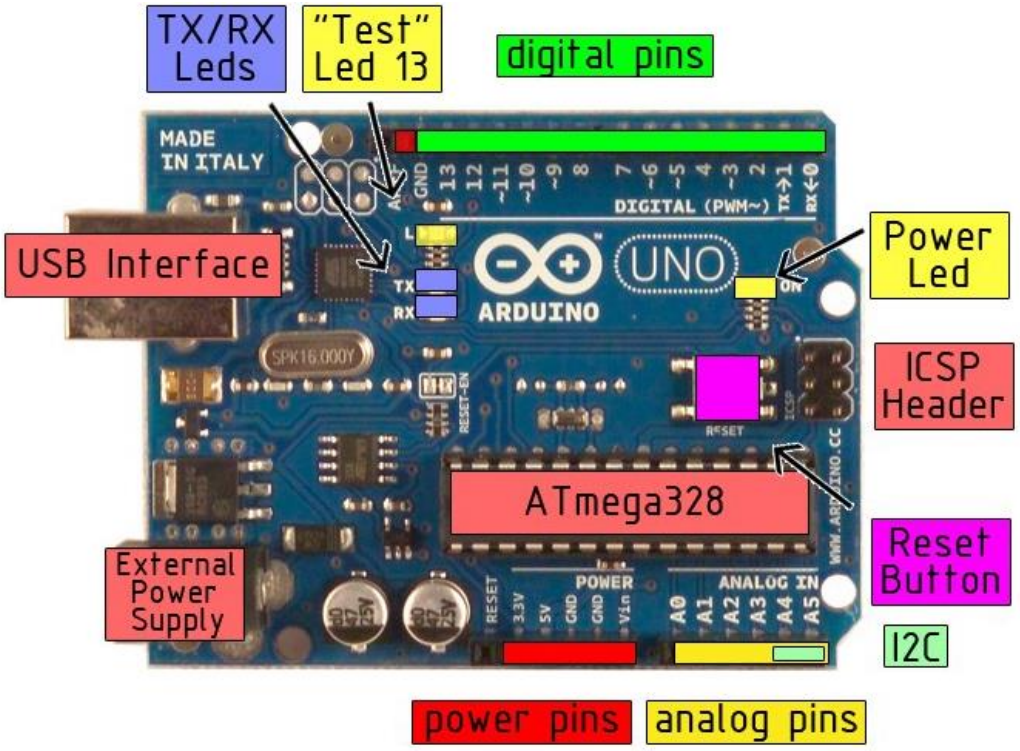


Fig 1.2: Arduino Uno board Description

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (nonUSB) power can come either from an AC to- DC adapter (wall -wart) or battery. The adapter can be connected by plugging a 2.1mm center -positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7-12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board.

3V3.A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins. The Arduino Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board to be plugged into solderless breadboards.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and may be used interchangeably. Many are the basic Arduino with the addition of commonplace output drivers, often for use in school-level education to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor, sometimes permitting the continued use of Shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

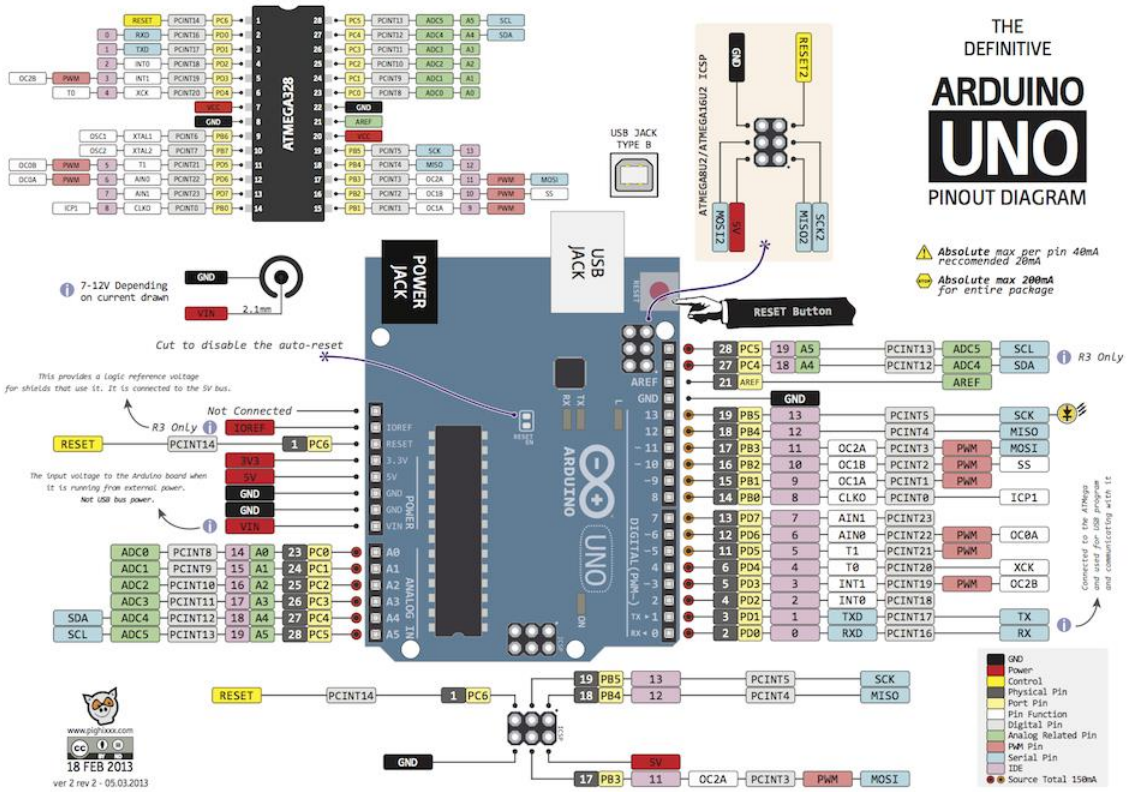


Fig1.3: Arduino Uno Pin Diagram

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM

INPUT AND OUTPUT PINS

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .

·**External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which,although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure

from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference () function. Additionally, some pins have specialized functionality:

• TWI: A4 or SDA pin and A5 or SCL pin.Support TWI communication using the Wire library

.

There are a couple of other pins on the board:

•AREF.Reference voltage for the analog inputs. Used with analog Reference

•Reset. Bring this line LOW to reset the microcontroller.Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required . The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB to serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also supports I2C (TWI) and SPI ommunication. The Arduino software includes a Wire library to simplify use of the I2C bus;
For SPI communication, use the SPI library

.

Programming

The Arduino Uno can be programmed with the Arduino

Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol You can also bypass the bootloader and program the microcontroller through the ICSP (In - Circuit Serial Programming) header;

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2 /8U2 is loaded with a DFU bootloader, which Can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

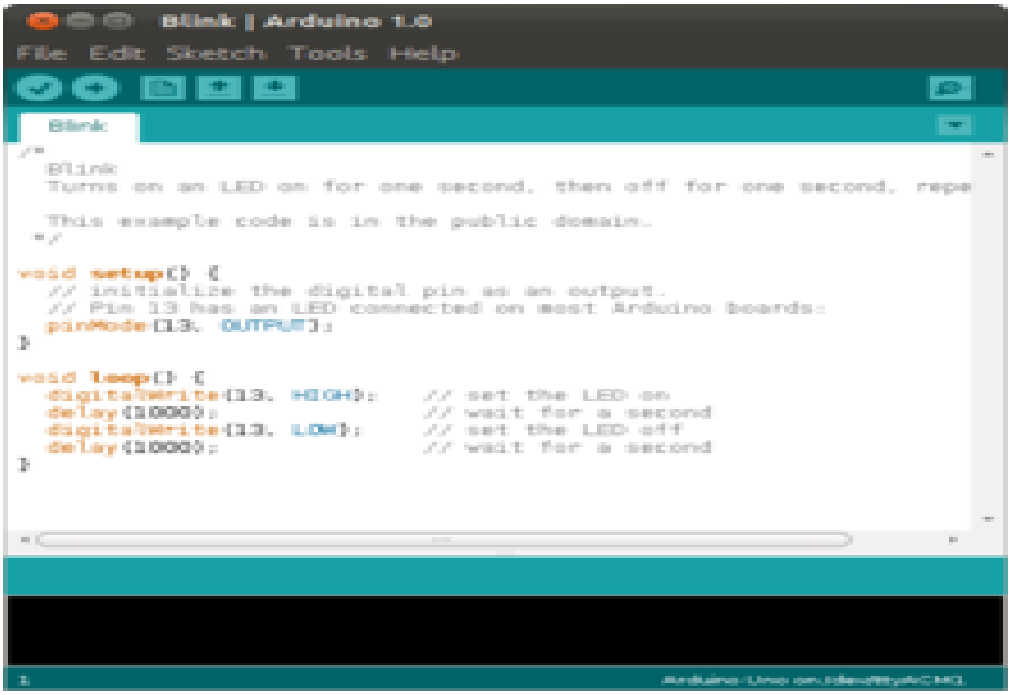


Fig 1.4: A screenshot of the Arduino IDE

Developer(s)	Arduino Software
Stable release	1.0.5 / May 15, 2013
Preview release	1.5.4 Beta / September 10, 2013
Written in	Java, C and C++
Operating system	Cross-platform
Type	Integrated development environment
License	LGPL or GPL license
Website	arduino.cc

The Arduino integrated development environment (IDE) is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce

programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a "sketch".

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

A typical first program for a microcontroller simply blinks an LED on and off. In the Arduino environment, the user might write a program like this:

```
The integrated pin 13 LED
#define LED_PIN 13

void setup () {
  pinMode (LED_PIN, OUTPUT); // Enable pin 13 for digital output
}

void loop () {
  digitalWrite (LED_PIN, HIGH); // Turn on the LED
  delay (1000); // Wait one second (1000 milliseconds)
  digitalWrite (LED_PIN, LOW); // Turn off the LED
  delay (1000); // Wait one second
```

}

It is a feature of most Arduino boards that they have an LED and load resistor connected between pin 13 and ground; a convenient feature for many simple tests.[11] The previous code would not be seen by a standard C++ compiler as a valid program, so when the user clicks the "Upload to I/O board" button in the IDE, a copy of the code is written to a temporary file with an extra include header at the top and a very simple main() function at the bottom, to make it a valid C++ program.

The Arduino IDE uses the GNU toolchain and AVR Libc to compile programs, and uses avrdude to upload programs to the board.

As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.[12][13]

Development

The core Arduino developer team is composed of Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis and Nicholas Zambetti. Massimo Banzi was interviewed on the March 21st, 2009 episode (Episode 61) of FLOSS Weekly on the TWiT.tv network, in which he discussed the history and goals of the Arduino project.[14] He also gave a talk at TEDGlobal 2012 Conference, where he outlined various uses of Arduino boards around the world.

Arduino is open source hardware: the Arduino hardware reference designs are distributed under a Creative Commons Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE is available and released under the GNU General Public License, version 2.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derivative works without permission. The official policy document on the use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the "Arduino" name by using "-duino" name variants.

Applications

- Xoscillo: open-source oscilloscope
- Scientific equipment
- Arduinome: a MIDI controller device that mimics the Monome
- OBDuino: a trip computer that uses the on-board diagnostics interface found in most modern cars
- The Humane Reader and Humane PC from Humane Informatics: low-cost electronic devices with TV-out that can hold a five thousand book library (e.g. offline Wikipedia compilations) on a microSD card
- Ardupilot: drone software / hardware
- ArduinoPhone

LCD Display

LCD stands for liquid crystal; this is a output device with a limited viewing angle. The choice of LCD as an output device was Because of its cost of use and is better with alphabets when compared with a 7-segment LED display. We have so many kinds of LCD today and our application requires a LCD with 2 lines and 16 characters per line, this gets data from the microcontroller and displays the same. It has 8 data lines, 3 control line, a supply voltage Vcc (+5v and a GND. This makes the whole device user friendly by showing the balance left in the card. This also shoes the card that is currently being used.

In recent years the LCD is finding widespread use replacing LED's. This is due to the following reasons:

1. The declining prices of LCD's.
2. The ability to display numbers, characters and graphics. This is in contrast to LED's, which are limited to numbers and few characters.
3. Incorporation of a refreshing controller into the LCD, there by relieving the CPU of the task of refreshing the LCD .in contrast, the Led must be refreshed by the CPU to keep displaying the data.
4. Ease of programming for characters and graphics.

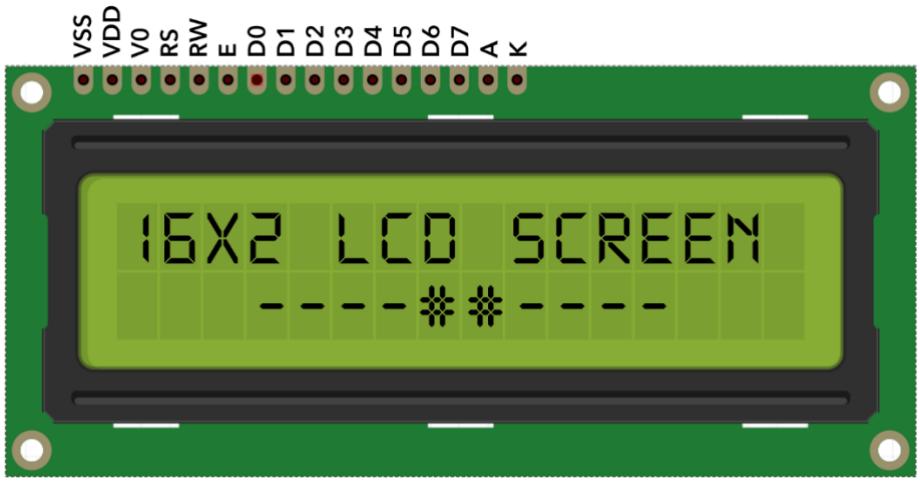


Fig 1.1 – 16x2 LCD Screen

LCD Pin Descriptions

VCC, VSS and VEE

While VCC and VSS provide +5v and ground respectively, VEE is used for controlling LCD contrast.

RS, Register Select

There are two very important registers inside the LCD. The RS pin used for their selection as follows. If RS=0, the instruction command code register is selected, allowing the user to sent a command such as clear display, cursor at home ,etc .IF RS=1 the data register is selected, allowing the user to sent data to be displayed on the LCD.

R/W Read/Write

R/W input allows the user to write information to the LCD or read information from it.

R/W=1 when reading; R/W=0 when writing.

E, ENABLE

The enable pin is used by the LCD to latch information present to its data pins. When data is supplied to data pins, a high to low pulse must be applied to this pin in order for the LCD to latch in the data present at the data pins. This pulse must be a minimum of 450ns wide.

D0-D7

The 8-bit data pins, D0-D7, are used to sent information to LCD or read the contents of the LCD’s internal registers.

The LCD commands codes are as shown in table.4. To display letters and numbers, we send ASCII codes for the letters A-Z, a-z, and numbers 0-9 to these pins while making RS=1.

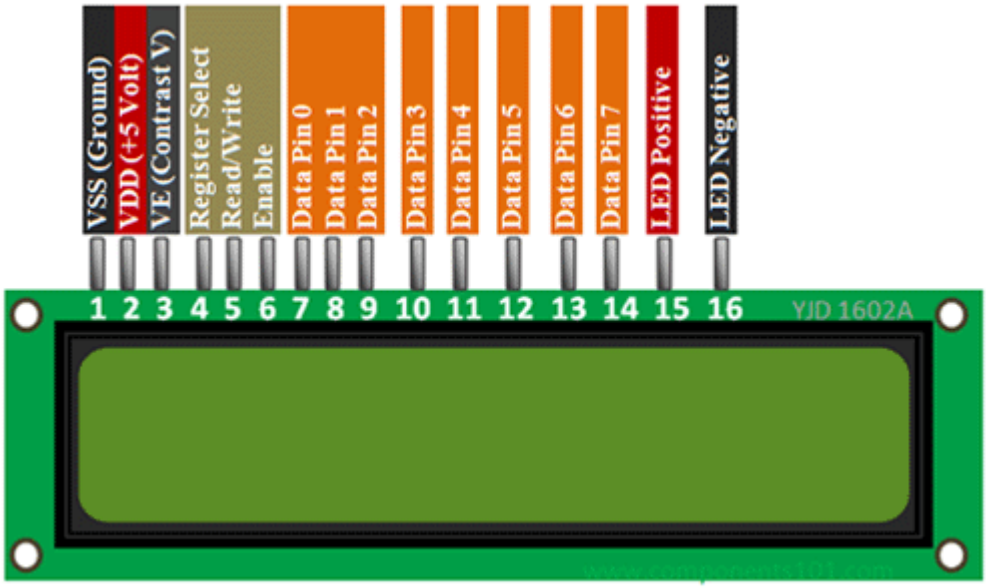


Fig 1.2 LCD Pin Description

LCD PROGRAMMING CHART

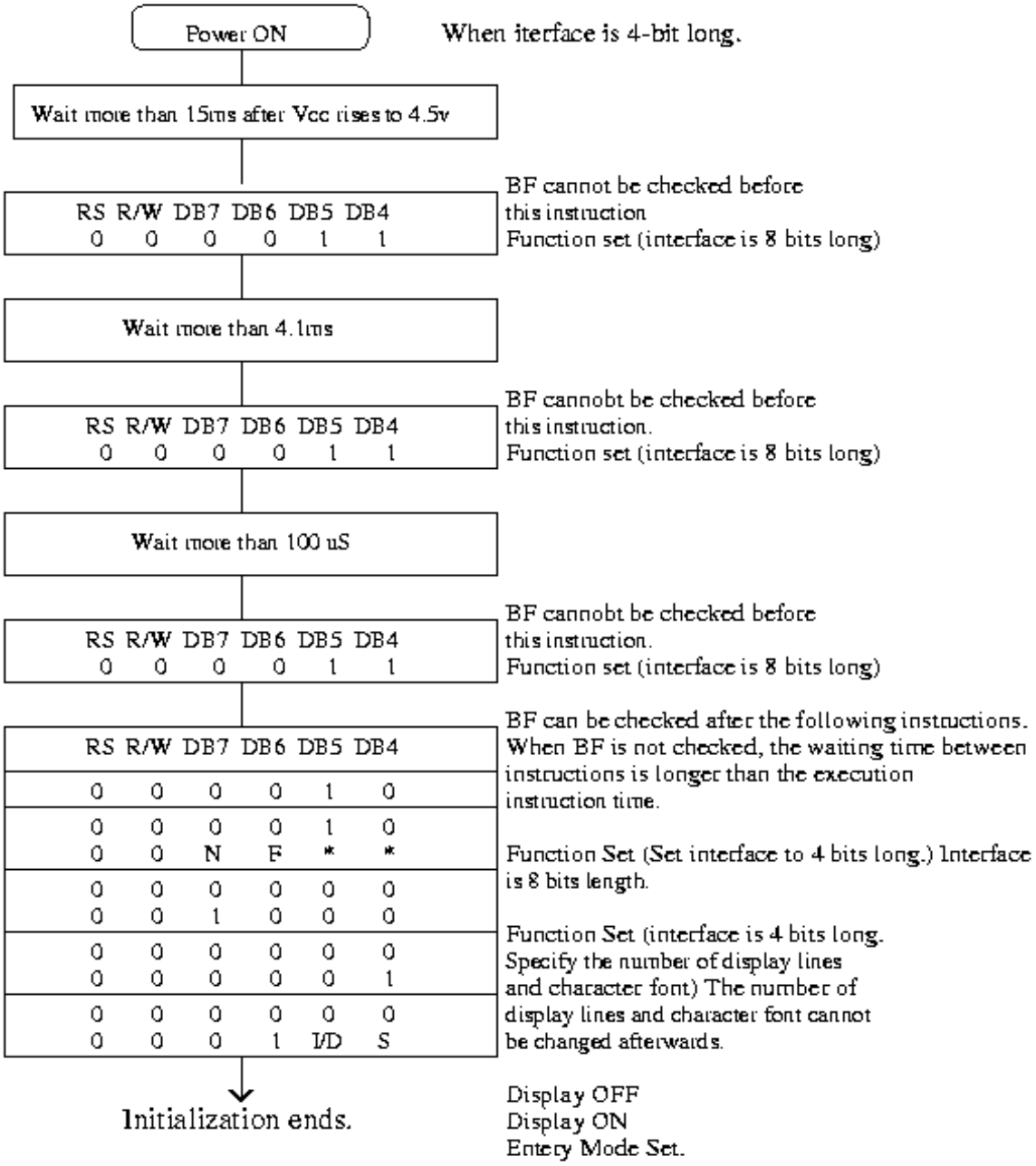


FIG 1.3.. LCD PROGRAMMING CHART

Programming the LCD

Before you may really use the LCD, you must initialize and configure it. This is accomplished by sending a number of initialization instructions to the LCD.

The first instruction we send must tell the LCD whether we'll be communicating with it with an 8-bit or 4-bit data bus. We also select a 5x8 dot character font. These two options are selected by sending the command 38h to the LCD as a command. As you will recall from the last section, we mentioned that the **RS** line must be low if we are sending a command to the LCD. The second byte of the initialization sequence is the instruction 0Eh. Thus we must repeat the initialization code from above, but now with the instruction. Thus the next code segment is: The last byte we need to send is used to configure additional operational parameters of the LCD. We must send the value 06h.

Thus, the first character in the upper left-hand corner is at address 00h. The following character position (character #2 on the first line) is address 01h, etc. This continues until we reach the 16th character of the first line which is at address 0Fh.

However, the first character of line 2, as shown in the memory map, is at address 40h. This means if we write a character to the last position of the first line and then write a second character, the second character will not appear on the second line. That is because the second character will effectively be written to address 10h--but the second line begins at address 40h.

Thus we need to send a command to the LCD that tells it to position the cursor on the second line. The "Set Cursor Position" instruction is 80h. To this we must add the address of the location where we wish to position the cursor.

ESP8266 NODEMCU WIFI DEVKIT

Fig 1.1 – NodeMCU Board

The ESP8266 is the name of a micro controller designed by Espressif Systems. The ESP8266 itself is a self-contained Wi-Fi networking solution offering as a bridge from existing micro controller to WiFi and is also capable of running self-contained applications. This module comes with a built in USB connector and a rich assortment of pin-outs. With a micro USB cable, you can connect NodeMCU devkit to your laptop and flash it without any trouble, just like Arduino. It is also immediately breadboard friendly.

Specification:

- Voltage:3.3V.
- Wi-Fi Direct (P2P), soft-AP.
- Current consumption: 10uA~170mA.
- Flash memory attachable: 16MB max (512K normal).
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz.
- RAM: 32K + 80K.
- GPIOs: 17 (multiplexed with other functions).

- Analog to Digital: 1 input with 1024 step resolution.
- +19.5dBm output power in 802.11b mode
- 802.11 support: b/g/n.
- Maximum concurrent TCP connections: 5.

The most basic way to use the ESP8266 module is to use serial commands, as the chip is basically a WiFi/Serial transceiver. However, this is not convenient. What we recommend is using the very cool Arduino ESP8266 project, which is a modified version of the Arduino IDE that you need to install on your computer. This makes it very convenient to use the ESP8266 chip as we will be using the well-known Arduino IDE. Following the below step to install ESP8266 library to work in Arduino IDE environment.

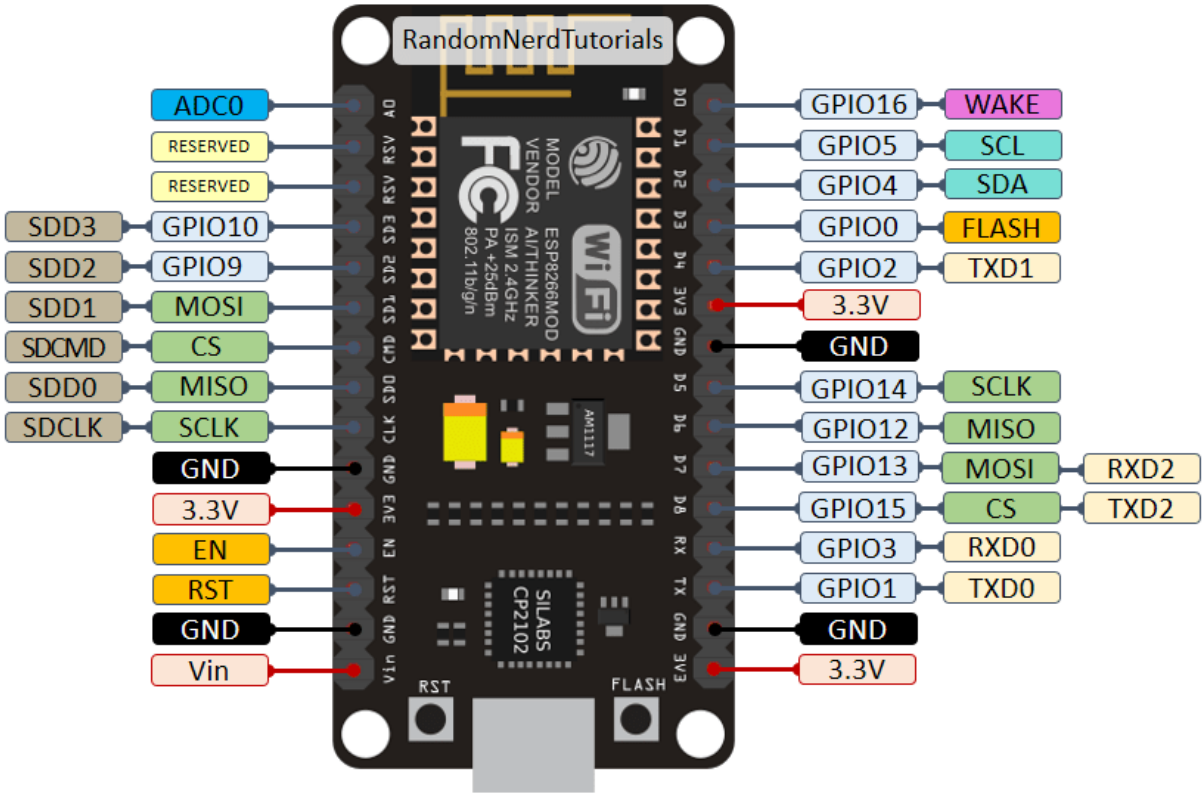


Fig 1.2 NodeMCU pinout Diagram

Internet of Things (IoT)

Definition

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

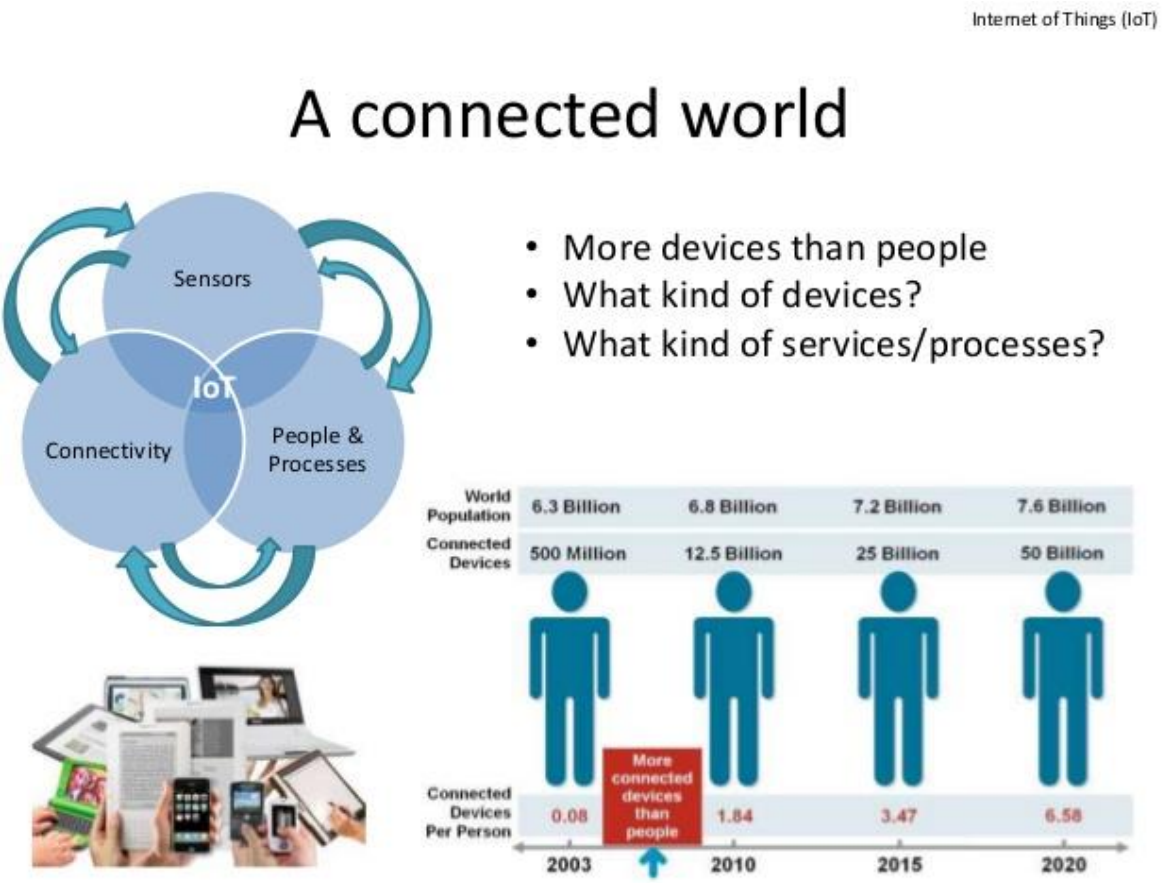


Fig 1.1 – IOT Structure

It is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and network connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

Experts estimate that the IoT will consist of about 30 billion objects by 2020. It is also estimated that the global market value of IoT will reach $7.1 trillion by 2020.

The IoT allows objects to be sensed or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit in addition to reduced human intervention. When IoT is augmented with sensors and actuators, the technology becomes an instance of the more general class of cyber-physical systems, which also encompasses technologies such as smart grids, virtual power plants, smart homes, intelligent transportation and smart cities.

"Things", in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, cameras streaming live feeds of wild animals in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring, or field operation devices that assist firefighters in search and rescue operations. Legal scholars suggest regarding "things" as an "inextricable mixture of hardware, software, data and service".These devices collect useful data with the help of various existing technologies and then autonomously flow the data between other devices.

History

As of 2016, the vision of the Internet of things has evolved due to a convergence of multiple technologies, including ubiquitous wireless communication, real-time analytics, machine learning, commodity sensors, and embedded systems.[13] This means that the traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things.

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold. Mark Weiser's seminal 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of IoT.[20] In 1994 Reza Raji described the concept in *IEEE Spectrum* as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1996 several companies proposed solutions like Microsoft's at Work or Novell's NEST. However, only in 1999 did the field start gathering momentum. Bill Joy envisioned Device to Device (D2D) communication as part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999

What is the Internet of Things? Key characteristics

You can define the Internet of Things by looking at the various characteristics in the broader context. We see all of these characteristics coming back in most Internet of Things definitions out there (further below is an overview with some of these IoT definitions).

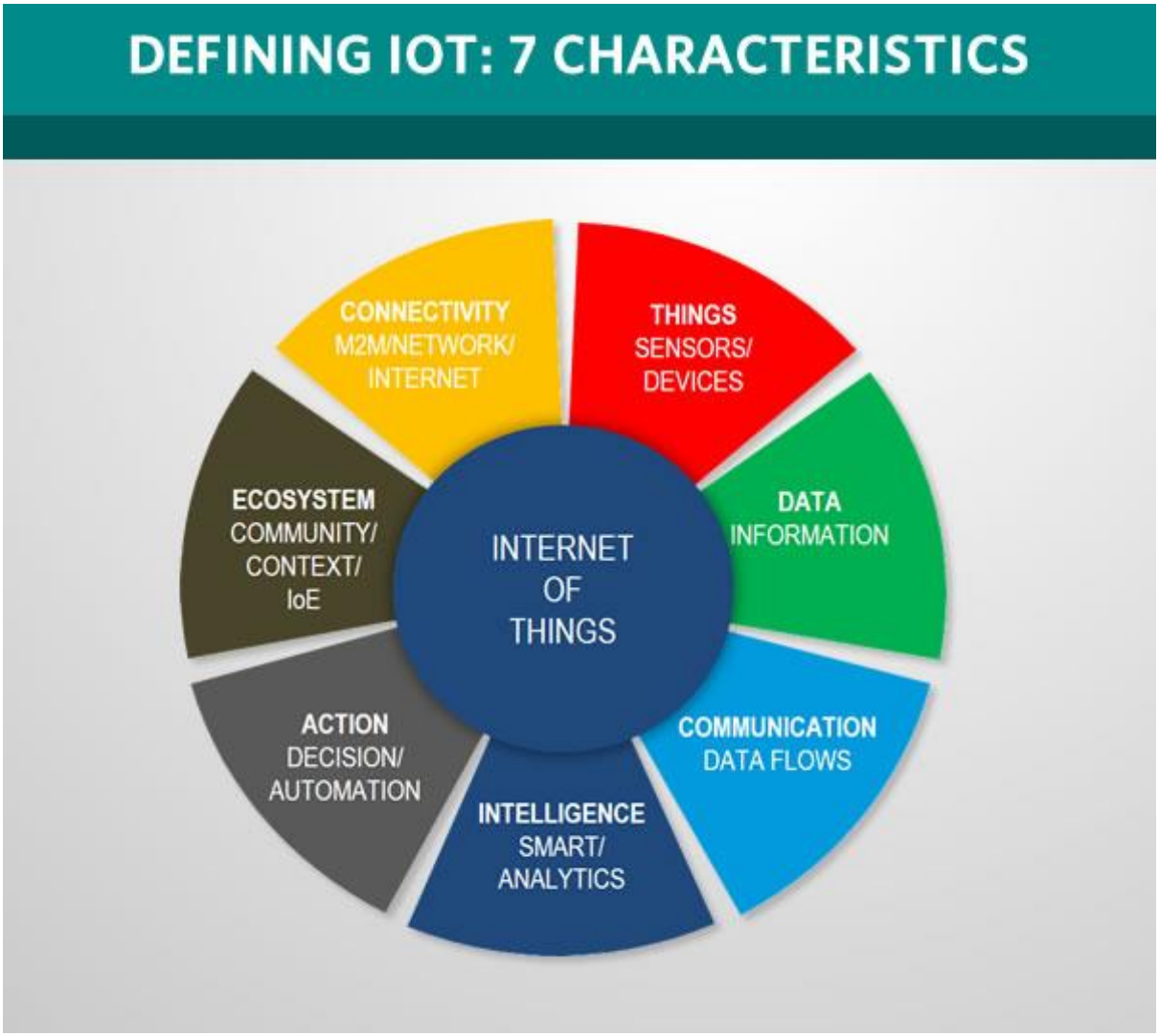


Fig 1.2 IOT Characteristics

There are 7 crucial Internet of Things characteristics:

1. **Connectivity.** This doesn't need much further explanation. Devices, sensors, they need to be connected: to an item, to eachother, actuators, a process and to 'the Internet' or another network.

2. **Things**. Anything that can be tagged or connected as such as it's designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items.
3. **Data**. Data is the bond of the Internet of Things, the first step towards action and intelligence.
4. **Communication**. Devices get connected so they can communicate data and this data can be analyzed.
5. **Intelligence**. The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from data analytics (also artificial intelligence).
6. **Action**. The consequence of intelligence. This can be manual action, action based upon decisions concerning phenomena (for instance in climate change decisions) and automation, often the most important piece.
7. **Ecosystem**. The place of the Internet of Things from a perspective of other technologies, communities, goals and the picture in which the Internet of Things fits. The Internet

Defining the Internet of Things with a consumer part and an industrial/business segment

The first distinction people started to make was between a consumer IoT and an Internet of Things for industrial applications or Industrial IoT as a way to distinguish between many types of IoT use cases and applications. Yet, as said and as with all terminology there were certainly overlaps in the definitions of these forms of IoT.

This is why some organizations and individuals, for instance, rather talk about the Internet of Everything, while others opt to drop the term IoT alltogether and mention it in terms of specific use cases and contexts such as smart cities, smart metering, smart buildings, smart wearables, Industrial Internet or smart homes, all of course with their own meaning and, again, with more subdivisions.

I.Consumer Internet of Things definition (CIoT)

The Consumer Internet of Things is what almost everybody knows. It's what the media talk about most.

The difference between the Industrial Internet of Things and Consumer Internet of Things as depicted by Vector Software – source – courtesy Vector Software

The Consumer Internet of Things or CIoT is where you will find applications and use cases to track your personal ‘assets’ (asset tracking), from your pet to your skateboard. Or where you will find the connected ‘smart appliances’ such as connected refrigerators, washing machines, light bulbs, etc.

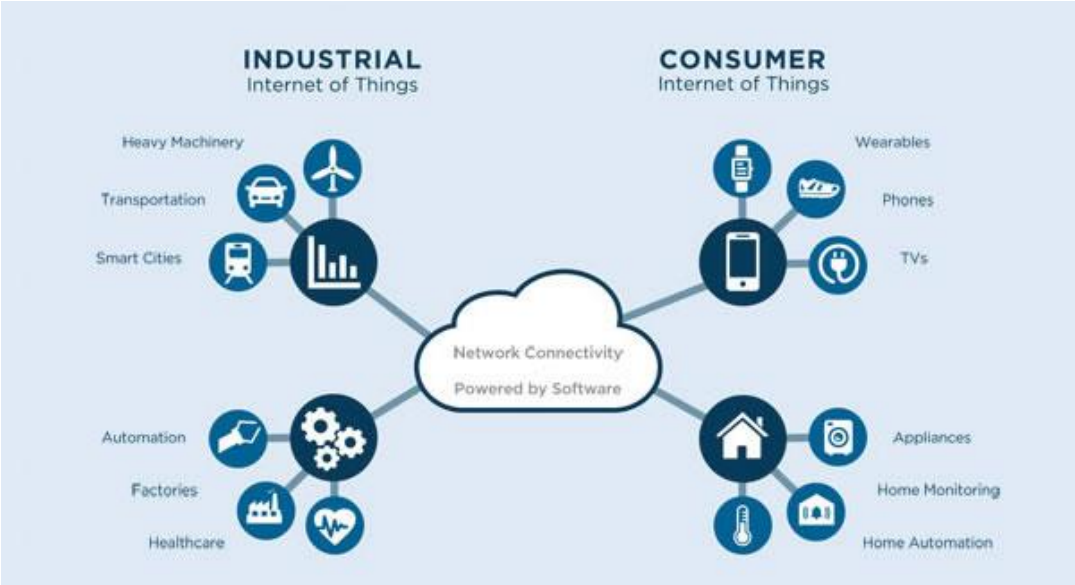


Fig 1.5 IOT Applications

Also wearables for consumer use (wearables are also used in healthcare and in factories, to name just two) and all sorts of consumer electronics such as smart wristwear belong to this category, along with all sorts of smart home appliances like thermostats or connected parking door openers.

The applications get better and smarter. They also get more independent from other devices such as smartphones. This is certainly the case with smart wearables. A simple definition of the Consumer Internet of Things is all we need: the Internet of Things as it’s used for consumer applications and consumer-oriented services.Typically, in Consumer IoT, data volumes and data communication needs are low and limited. That’s why there are many technologies of which some are specifically designed for consumer applications, ranging from smart home connectivity standards to special operating systems for wearables.

II.The Industrial Internet of Things

The Industrial Internet of Things describes typical industry use cases across a range of sectors. Some people see the Industrial Internet of Things more in a context of ‘heavy’ industries like manufacturing or utilities. But it is also used for use cases in, for example smart cities and smart metering.

If we look at it as a sort of ‘Business Internet of Things’ it is clear that there are some overlaps with the Consumer Internet of Things. For instance: if you have a smart thermostat and smart energy consumption meter in your house they are on one hand consumer applications because they are for personal usage. But from the perspective of the company that uses it to send you invoices and to help optimize energy consumption it is a business matter. So, the terms are not that good but that’s how it is and it’s better to look at use cases than at these broad categories because just as there are many different applications in the Consumer Internet of Things, there are also many in IIoT and some are hard to compare.

Manufacturing

Network control and management of manufacturing equipment, asset and situation management, or manufacturing process control bring the IoT within the realm of industrial applications and smart manufacturing as well. The IoT intelligent systems enable rapid manufacturing of new products, dynamic response to product demands, and real-time optimization of manufacturing production and supply chain networks, by networking machinery, sensors and control systems together.

Digital control systems to automate process controls, operator tools and service information systems to optimize plant safety and security are within the purview of the IoT. But it also extends itself to asset management via predictive maintenance, statistical evaluation, and measurements to maximize reliability. Smart industrial management systems can also be integrated with the Smart Grid, thereby enabling real-time energy optimization. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by a large number of networked sensors

Applications

One key application of smart home is to provide support and aid for disabled and elderly individuals. These smart home systems utilize assistive technology to accommodate user’s specific disabilities. Voice control can help users with sight and mobility limits while alert systems can be connected directly to Cochlear implants worn by hearing impaired users. The

system can also be equipped with additional safety features. These features can include sensors systems that monitor for medical emergencies. Smart home technology applied in this way can provide users with more liberty and a higher quality of life.

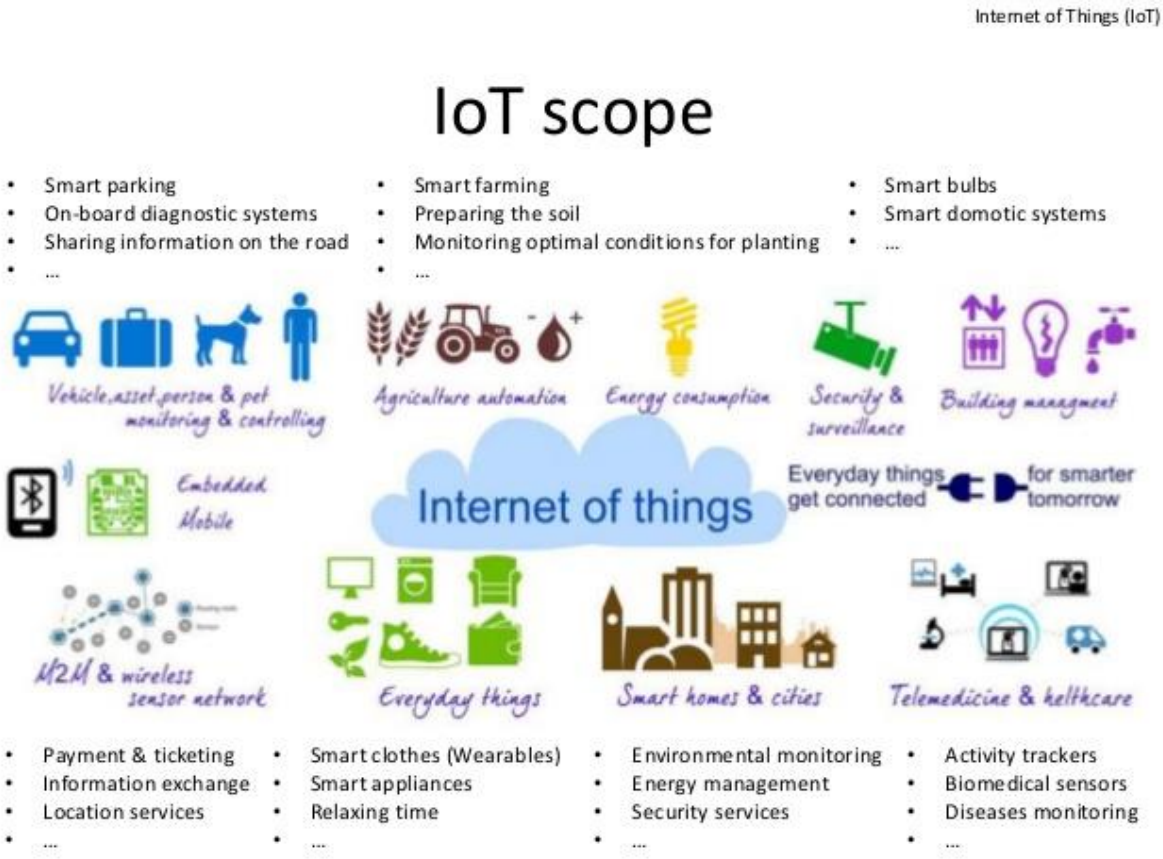


Fig 1.3 – Scope of IOT in Different Fields

A second application of smart home is even more complicated. One can guide the connected device at home even from far away. For example if a user leaves the office, it is possible to tell a connected air conditioner device via smart phone to cool down the house to a certain temperature.

Another example would be to use smart devices as for examples Amazon`s Alexa to get the most recent and most important news of the day while cutting the vegetables for the meal you are cooking at the moment. In general, Smart Home devices make life easier at home and give user possibility to make several things at the same time.

Agriculture

The IoT contributes significantly towards innovating farming methods. Farming tasks caused by population growth and climate change have made it one of the first industries to utilize the

IoT. The integration of wireless sensors with agricultural mobile apps and cloud platforms helps in collecting vital information pertaining to the environmental conditions – temperature, rainfall, humidity, wind speed, pest infections, soil humus content or nutrients, besides others – linked with a farmland, can be used to improve and automate farming techniques, take informed decisions to improve quality and quantity of agriculture, and minimize risks and wastages. The app-based field and crop monitoring also reduces the hassles of managing crops at multiple locations. For example, farmers can now detect areas that have been fertilised (or mistakenly missed), if the land is too dry and predict future yields.

Environmental monitoring

Environmental monitoring applications of the IoT naturally use sensors to assist in environmental protection by monitoring air quality or water quality, atmospheric or soil conditions, and the even in areas like monitoring the movements of animals in wildlife and their habitats. Development of specific-constrained devices connected to the Internet also means that other applications like earthquake or tsunami early-warning systems which can also be used by emergency services to provide more effective aid. IoT devices in this application typically extent a large geographic area and can also be mobile. It has been argued that the regularity of IoT brings to wireless sensing will revolutionize this area.

Building and home automation

IoT devices can be used in various types of buildings for monitoring and controlling the mechanical, electrical and electronic systems (e.g., public and private, industrial, institutions, or residential) in home automation and building automation systems. In this framework, three main areas are being covered in literature:

- The integration of the internet with building energy management systems in order to create energy efficient and IOT driven “smart buildings”.
- The possible means of real-time monitoring for reducing energy consumption and monitoring occupant behaviors.
- The integration of smart devices in the built environment and how they might be used in future applications.

Medical and healthcare

IoT devices have main usage in medical fields due to their high speed and reliability. They can be used in remote health monitoring and emergency notification systems. These health monitoring devices are blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants, such as pacemakers, Fitbit electronic wristbands, or advanced hearing aids. In some hospitals there are "smart beds" that can detect when they are occupied and when a patient is attempting to get up. It can also adjust itself to an appropriate pressure and support is applied to the patient without the manual interaction of nurses.

Specialized sensors can also be used within living spaces to monitor the health and general well-being of senior citizens, while also ensuring that proper treatment is being administered and assisting people regain lost mobility via therapy as well. Other end user devices to encourage healthy living, such as, connected scales or wearable heart monitors, are also a possibility with the IoT. More and more health monitoring IoT platforms are coming up for antenatal and chronic patients, helping one manage health vitals and chronic medication requirements.

Blynk App – IOT Platform

Blynk is an IOT Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.

It is like a digital dashboard where one can build a graphic interface for the project by simply dragging and dropping widgets.

Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the **Internet Of Your Things**.

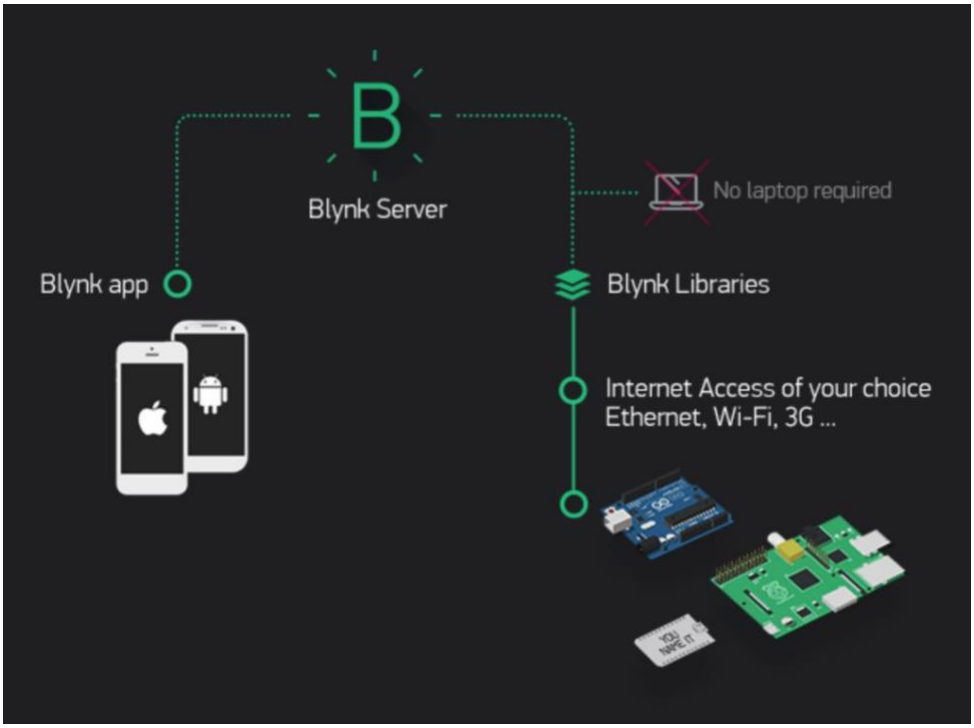


Fig - IOT Platform in Blynk App

Circuit Diagram:

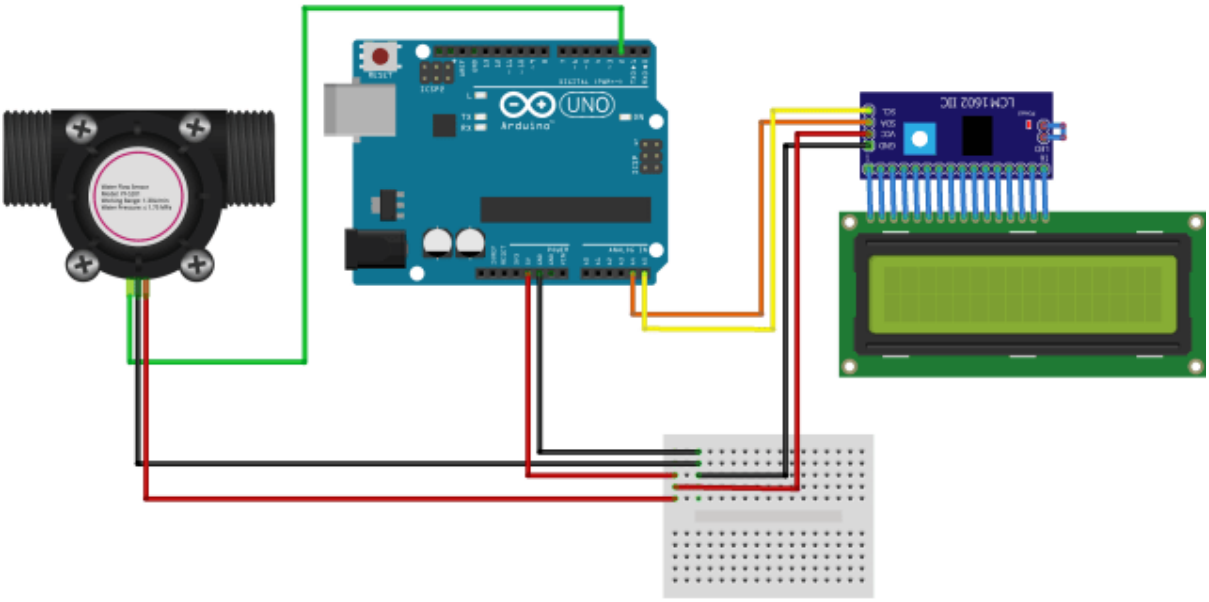


Fig Arduino with Flow Sensor

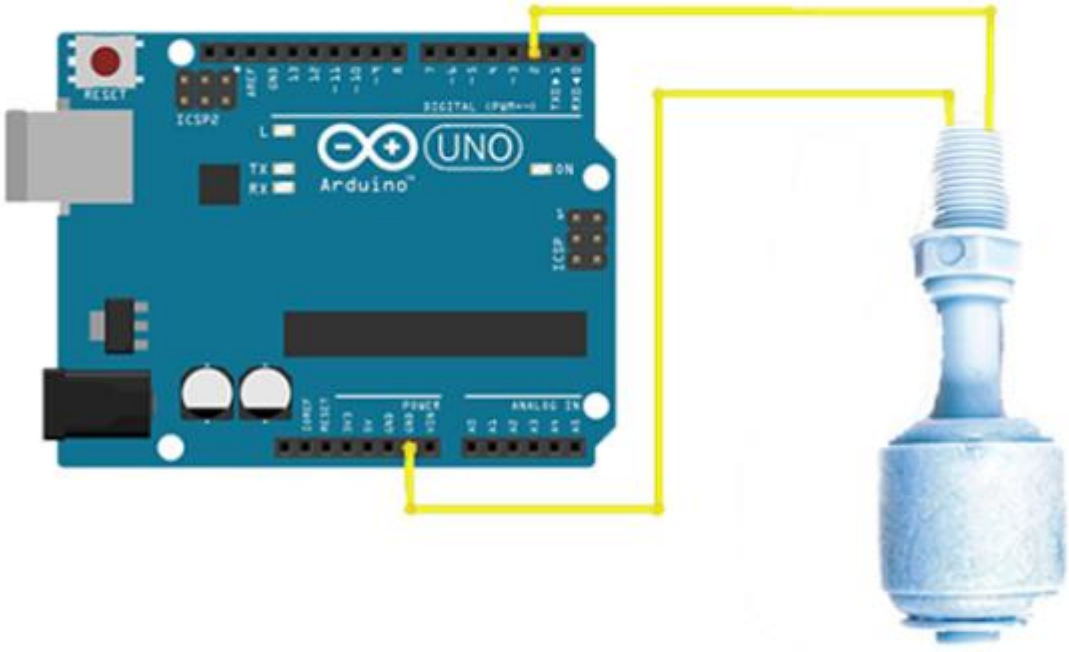


Fig Arduino with Float Sensor

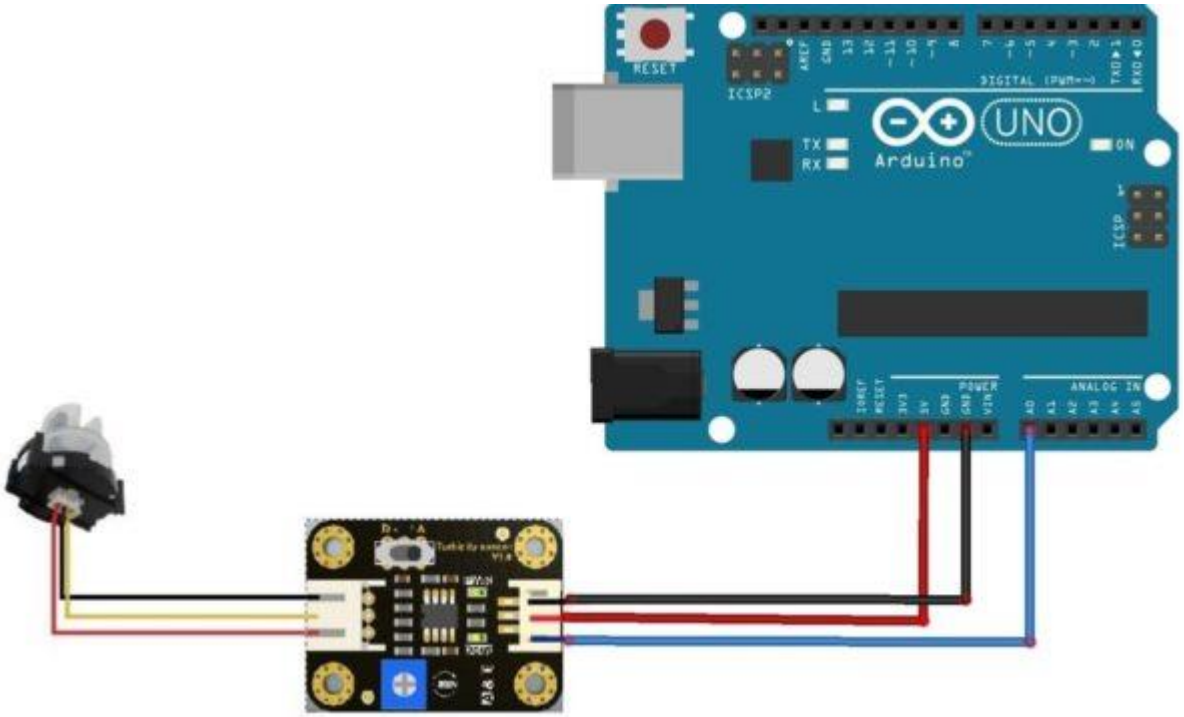


Fig Arduino with Turbidity Sensor

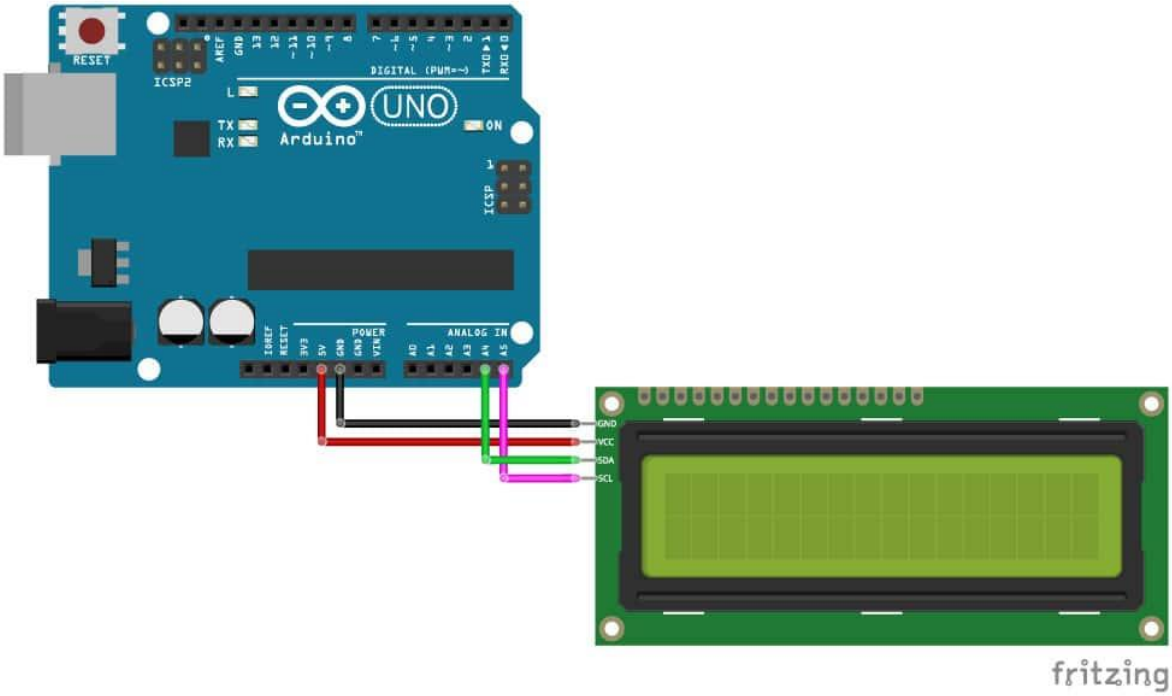


Fig Arduino with LCD

Program:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9,8);

#include <SoftwareSerial.h>
SoftwareSerial mySerial(4,3);


int X;
int Y;
float TIME = 0;
float FREQUENCY = 0;
float WATER = 0;
float TOTAL = 0;
float LS = 0;
const int input = 2;

#define sol1 7
#define sol2 6

#define float_sw1 A1
#define float_sw2 A2

#define sensorPin A0
int turbidity,count=0;
int sensorValue;


void send_data()
{
    mySerial.print("Turbidity:");
    mySerial.print(turbidity);
    mySerial.print('\n');
    delay(300);

    mySerial.print("Liters:");
    mySerial.print(TOTAL);
    mySerial.print('\n');
    delay(300);



}


void setup()
{
Serial.begin(9600);
mySerial.begin(9600);
```

```
lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("IOT Based Water");
lcd.setCursor(0,1);
lcd.print("Dist - System  ");
delay(2000);
pinMode(input,INPUT);
pinMode(sol1,OUTPUT);
pinMode(sol2,OUTPUT);

pinMode(float_sw1,INPUT_PULLUP);
pinMode(float_sw2,INPUT_PULLUP);

digitalWrite(sol1,HIGH);
digitalWrite(sol2,HIGH);
}


void loop()
{
  sensorValue = analogRead(sensorPin);
  turbidity = map(sensorValue, 0, 750, 100, 0);
  Serial.print("Turbidity:");
  Serial.println(turbidity);
  lcd.setCursor(11,0);
  lcd.print("T:");
  lcd.print(turbidity);
  delay(500);
if(!digitalRead(float_sw1))
{
  digitalWrite(sol1,LOW);
  delay(200);
}
if(digitalRead(float_sw1))
{
  digitalWrite(sol1,HIGH);
  delay(200);
}

if(!digitalRead(float_sw2))
{
  digitalWrite(sol2,LOW);
  delay(200);
}
if(digitalRead(float_sw2))
{
  digitalWrite(sol2,HIGH);
  delay(200);
```

```
}

X = pulseIn(input, HIGH);
Y = pulseIn(input, LOW);
TIME = X + Y;
FREQUENCY = 1000000/TIME;
WATER = FREQUENCY/7.5;
LS = WATER/60;
if(FREQUENCY >= 0)
{
if(isinf(FREQUENCY))
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("VOL. :0.00");
lcd.setCursor(0,1);
lcd.print("TOTAL:");
lcd.print( TOTAL);
lcd.print(" L");
}
else
{
TOTAL = TOTAL + LS;
Serial.println(FREQUENCY);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("VOL.: ");
lcd.print(WATER);
lcd.print(" L/M");
lcd.setCursor(0,1);
lcd.print("TOTAL:");
lcd.print( TOTAL);
lcd.print(" L");
}
}
delay(500);
count=count+1;
if(count>=10)
{
  count=0;
  send_data();
}

}
```

ADVANTAGE:

- EASY TO MONITER
- WE GET PURE WATER
- EASY TO FIND THE WATER FLOW
- TO AVOID WATER WASTE
- NO NEED HUMAN POWER
- LOW ENGERY CONSUPTION

CONCLUSION

This project addresses the environmental features like durability, affordability, prevention against leakage and maintenance issues. Hence this system avoids excess us age of water, wastage of water and promises efficient water management system. The future scope is to develop the system using PLC. This helps in improving the reliability and durability. A greater number of nodes can be included and also replacement for a particular component can be made easily. To develop an app in which the local residents can monitor their consumption of water

REFERENCE:

- Poonam J. Chavan, Manoj Mechkul “IoT Based Water quality Monitoring” IJMTER Journal, Vol 3, pp.746-750,April 2016.
- Mithali Borbade, Shruti Danve “Real Time Water Quality monitoring system” IJIRCE journal, Vol 3, pp. 5046-5068, June 2015.
- Vaishnavi V, Daigovane, Dr.M.A. Gaikwad “Water quality monitoring system based on IoT” Research India publicationon Vol 10, pp. 1107 1116, 2017.
- Pradeepkumar M. Manisha J.Praveen Sha R. Proiserin V. Suganya Devi K “The real timemonitoring of water quality in IoT Environment” IJIRSET, Vol 5, pp. 4419-4427, March 2016.
- Aaina Venkteshwaran, Harsha Menda P. Prof. Priti Bodar “An IoT based system for water quality monitoring” IJRCCE journal, pp. 2510-2515, April 2017.
- S.Geeta, S.Goutami “Internet of Things enabled real time water quality monitoring system” Springer Open journal Vol 5, pp. 1-19, 2017.
- Jyoti Bhatt, Jignesh Pataliya “IoT based water quality monitoring system”IJIEE journal, Vol 4, pp. 44-48, april 2016.
- N Vijaykumar ,R Ramyas, “The real time monitoring of water quality in IOT environment”,IEEE sponsored 2nd international conference on innovations in information, embedded and communication systems (Iciiecs)2015
- Saima Maqbool , Nidhi Chandra, “Real Time Wireless Monitoringand Control of Water Systems using Zigbee 802.15.4” 5th International Conference on Computational Intelligence and Communication Networks., 2013
- Prachet Varma, Akshay Kumar, Nihesh Rathod, Pratik Jain,Mallikarjun S,Renu Subramaniyam,Bhardhwaj Amrutur,M.S.Mohan kumar,Rajesh Sundresan, IoT based water management System for a Campus IEEE,IEEE First International Smart Cities Conference (ISC2),2015

PHOTOGRAPHY