**CHAPTER 3**

**FUNDAMENTALS OF THE SYSTEM**

**3.1INTRODUCTION**

In recent years, robot technology has developed significantly. Most of the traditional robots are still commonly used for industrial applications, such as in car assembly factories Meanwhile, intelligent robots have become popular in daily life applications. Human-friendly robots are now used for taking care of the elderly . The purpose of a human following robot is to improve the relationship between people and the robot . For instance, the robot can carry heavy loads for people in hospitals, airports and shopping centers. The robot can provide services to humans as an assistant in different kinds of situations. In robotic research, vision-based robots have gained growing interests for navigation, however, the tradition method of line following navigation still plays an important role in mobile robot technology. This is because a robot with line following capability requires a lower cost to build and has a simple design . Besides, the application of Radio Frequency Identification (RFID) technology for robots nowadays has become popular, especially in the localization scheme. It is a non-touching recognition system that can tag and send tag data wirelessly at various distances.

**3.2 Hardware Components:**

* + - Arduino
    - LCD
    - RFID Reader
    - RFID tags
    - HC-05

**Software Requirements:**

* + - * Arduino IDE
      * Thingspeak

**ARDUINO UNO**



The Arduino Uno is a microcontroller board based on the ATmega328 [(datasheet)](http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

Arduino is selected because it is an open-source platform, inexpensive and provides sufficient analog/digital I/O pins for customizable applications. It operates at 5V and is powered with Atmel’s ATmega328 micro-controller with a clock speed of 16 MHz’s It has a flash memory of 32kB and Static Random Access Memory (SRAM) of 2kB. It has 6 analog pins and serial ports. One of the serial ports is connected internally to Universal Serial Bus (USB) port.

**3.3 Summar**

**A) Power**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-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 (asopposed 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.** The regulated power supply used to power the microcontroller and other components onthe board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
* **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
* **GND.** Ground pins.

**B) Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode(),](http://arduino.cc/en/Reference/PinMode)[digitalWrite(),](http://arduino.cc/en/Reference/DigitalWrite) and [digitalRead()](http://arduino.cc/en/Reference/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:

**C) Communication**

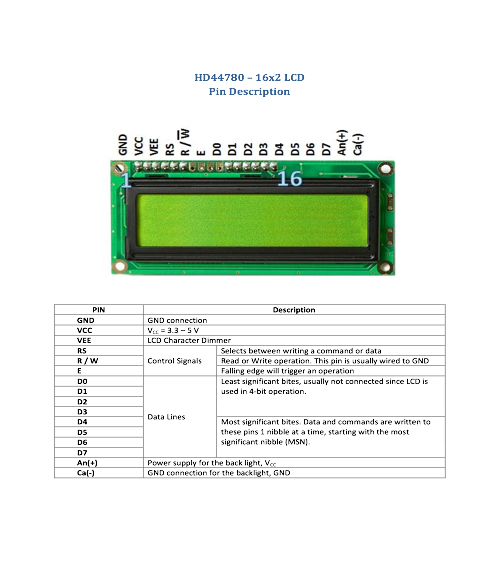
The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [onWindows, a .inf file is required.](http://arduino.cc/en/Guide/Windows#toc4) 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](http://www.arduino.cc/en/Reference/SoftwareSerial) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](http://arduino.cc/en/Reference/Wire) for details. For SPI communication, use the [SPI library.](http://arduino.cc/en/Reference/SPI)

**D) Physical Characteristics**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

**3.4 LCD**



The material “liquid crystal” was discovered accidentally by the botanist FreidrichReinitzer as early as 1888. However the commercially available liquid crystals were not developed until the late 1960’s

We get the definition of LCD from the name “Liquid Crystal” itself. It is actually a combination of two states of matter – the solid and the liquid. They have both the properties of solids and liquids and maintain their respective states with respect to another. Solids usually maintain their state unlike liquids who change their orientation and move everywhere in the particular liquid. Further studies have showed that liquid crystal materials show more of a liquid state than that of a solid. It must also be noted that liquid crystals are more heat sensitive than usual liquids. A little amount of heat can easily turn the liquid crystal into a liquid. This is the reason why they are also used to make thermometers.

The liquid-crystal display has the distinct advantage of having a low power consumption than the LED. It is typically of the order of microwatts for the display in comparison to the some order of milliwatts for LEDs. Low power consumption requirement has made it compatible with MOS integrated logic circuit. Its other advantages are its low cost, and good contrast. The main drawbacks of lcds are additional requirement of light source, a limited temperature range of operation (between 0 and 60° C), low reliability, short oper­ating life, poor visibility in low ambient lighting, slow speed and the need for an ac drive.

**Basic structure of an LCD**

A liquid crystal cell consists of a thin layer (about 10 u m) of a liquid crystal sand­wiched between two glass sheets with transparent elec­trodes deposited on their inside faces. With both glass sheets transparent, the cell is known as transmittive type cell. When one glass is transparent and the other has a reflective coating, the cell is called reflective type. The LCD does not produce any illumination of its own. It, in fact, depends entirely on illumination falling on it from an external source for its visual effect

**3.5 RFID CARDS**

RFID stands for radio frequency identification. RFID tags are small chips (usually comes in a smart card or visiting card shape) that are used in our day to day life for unlocking hotel rooms, entering into cars etc.These tiny chips along with an RFID reader forms the RFID system.

RFID  technology was first used during world war 2 to identify enemy aircrafts. Since then RFID technology has evolved and is now used in many different industries. One clean example is a smart warehouse where the process of warehousing is automated using RFID Technology. Watch the video given below to understand an RFID warehouse.

An RFID system consists of two parts

**1)RFID Reader**

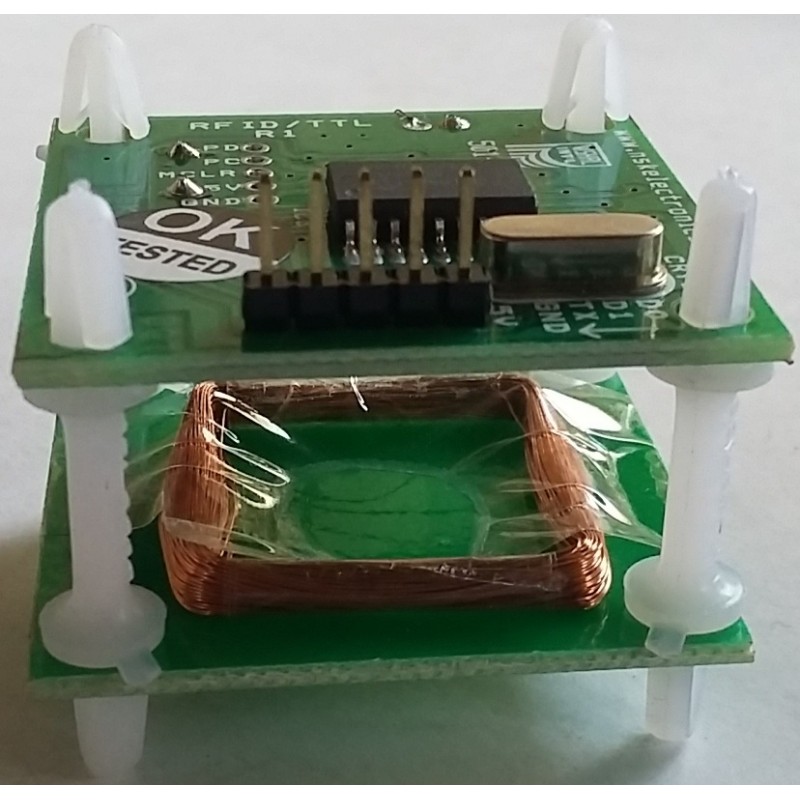
**2)RFID Tag**.

Data is stored in the RFID tag electronically. This data is retrieved by the reader using electromagnetic waves. Tags can store only a few kilo bytes of data.

The operation of an RFID reader is very much similar to barcode scanning method which uses UPC (Universal Product Codes) codes. In some applications, RFID has advantages over the barcode system.

#### ****What is an RFID Reader?****

The image given below is that of an RFID reader module.



The RFID reader has a radio transmitter and receiver inside. It is also called as an interrogator. The reader transmits radio frequency signals continuously upon powering. When an RFID tag is placed inside the range area of a reader, it energizes the tag through electromagnetic induction and collects the information from it.

#### ****What is an RFID Tag****

[](http://www.circuitstoday.com/wp-content/uploads/2017/08/RFID-tag.jpg)

**2.a) RFID Tag**

The image given above is that of an RFID tag (smart card shaped tag). RFID tags are available in different types of size and shapes. The Tag contains an IC for storing the data, an antenna for transmitting and receiving, and also a modulator. Tags are very small in size and they can hold only few bits of data.

#### ****2.b) Types of RFID Tags****

There are two types of RFID tags:

1. **Active**
2. **Passive**

##### **i) Active RFID Tags**

* An RFID tag can be defined as an active tag, when it has its own power source and allows the user to erase and update the data inside the tag.
* Active tags are available with an external battery.
* Active tags are usually operated at different frequencies of 433Mhz and 915 MHZ.
* Generally 433Mhz is preferred because of longer wavelength .
* They can be read from a distance of 100 feet or more, that is they can communicate over longer distances.
* They are expensive compared to passive tags.

**Advantages of Active RFID Tags**

* They can provide long communication range.
* They have highest data bandwidth.

###### **Disadvantages of Active RFID Tags**

* They cannot work without battery.
* Active RFID tags are expensive
* Active tags are larger in size, hence not suitable for all kinds of applications.

**ii) Passive RFID Tags**

* Passive tags are activated only when it receives a signal from the reader.
* This signal activates the internal IC of the tag through antenna.
* Thus the tag sends the required data stored in it to the reader.
* These tags do not use any external battery. They are powered up by using the power from RFID reader.
* The data in the passive tags is pre written. One cannot change the data in these tags.
* These tags are generally used for applications such as file tracking, race timing etc.

###### **Advantages of Passive RFID Tags**

* They are smaller in size.
* They are less expensive.
* They last long as they do not use any battery.

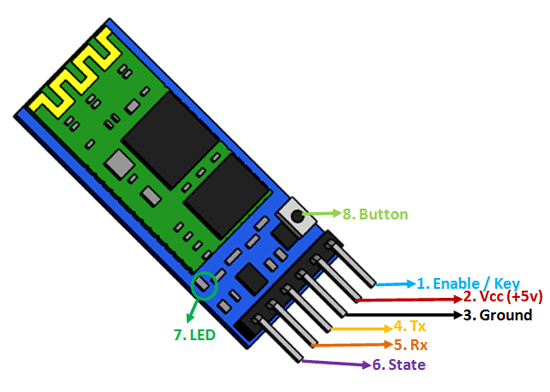
###### **Disadvantages of Passive RFID Tags**

* These tags can be used only for short distance communication.
* The data written in passive tags cannot be updated.

**3.6HC- 05**

****

**Figure 1:HC-05**

****

**Figure 2:HC-05 Pinout**

### ****Pin Configuration****

|  |  |  |
| --- | --- | --- |
| **Pin Number** | **Pin Name** | **Description** |
| 1 | Enable / Key | This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode |
| 2 | Vcc | Powers the module. Connect to +5V Supply voltage |
| 3 | Ground | Ground pin of module, connect to system ground. |
| 4 | TX – Transmitter | Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data. |
| 5 | RX – Receiver | Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth |
| 6 | State | The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly. |
| 7 | LED | Indicates the status of Module   * Blink once in 2 sec: Module has entered Command Mode * Repeated Blinking: Waiting for connection in Data Mode * Blink twice in 1 sec: Connection successful in Data Mode |
| 8 | Button | Used to control the Key/Enable pin to toggle between Data and command Mode |

### ****i) HC-05 Default Settings****

Default Bluetooth Name: “HC-05”

Default Password: 1234 or 0000

### ii) ****HC-05 Technical Specifications****

* Serial Bluetooth module for [Arduino](https://components101.com/microcontrollers/arduino-uno) and other microcontrollers
* Operating Voltage: 4V to 6V (Typically +5V)
* Operating Current: 30mA
* Range: <100m
* Can be easily interfaced with Laptop or Mobile phones with Bluetooth
* Supported baud rate: 9600,19200

**3.7 SOFTWARE REQUIREMENTS**

**What is IoT?**

Internet of Things (IoT) describes an emerging trend where a large number of embedded devices (things) are connected to the Internet. These connected devices communicate with people and other things and often provide sensor data to cloud storage and cloud computing resources where the data is processed and analyzed to gain important insights. Cheap cloud computing power and increased device connectivity is enabling this trend.

IoT solutions are built for many vertical applications such as environmental monitoring and control, health monitoring, vehicle fleet monitoring, industrial monitoring and control, and home automation.

**3.7.a)Arduino IDE**

**Table of Contents**

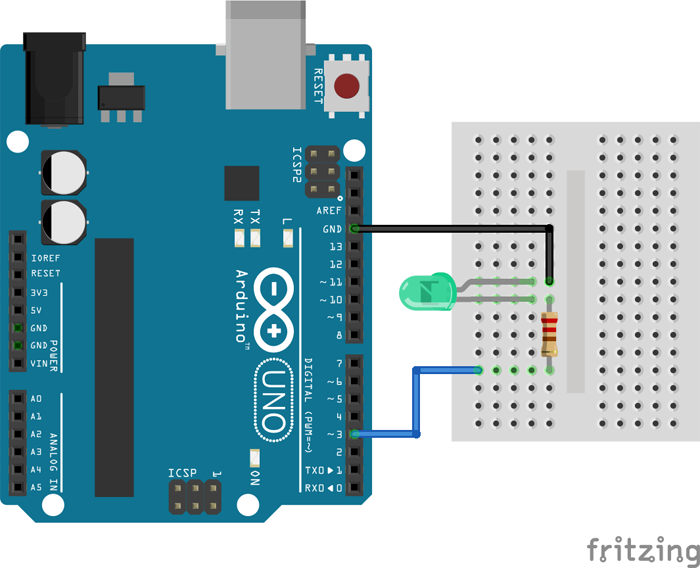
1. Installing Arduino IDE
2. Building the Circuit
3. Writing an Arduino Sketch
4. Uploading the Arduino Sketch and Flashing the LED

## 1) Installing Arduino IDE

Arduino IDE is Arduino’s open-source software integrated development environment. An IDE consists of all the necessary tools for software development. To use your Arduino board you will need to download the Arduino IDE and use it to edit your source code and then upload your code to the board. Arduino IDE is available for Windows, Mac, and Linux..

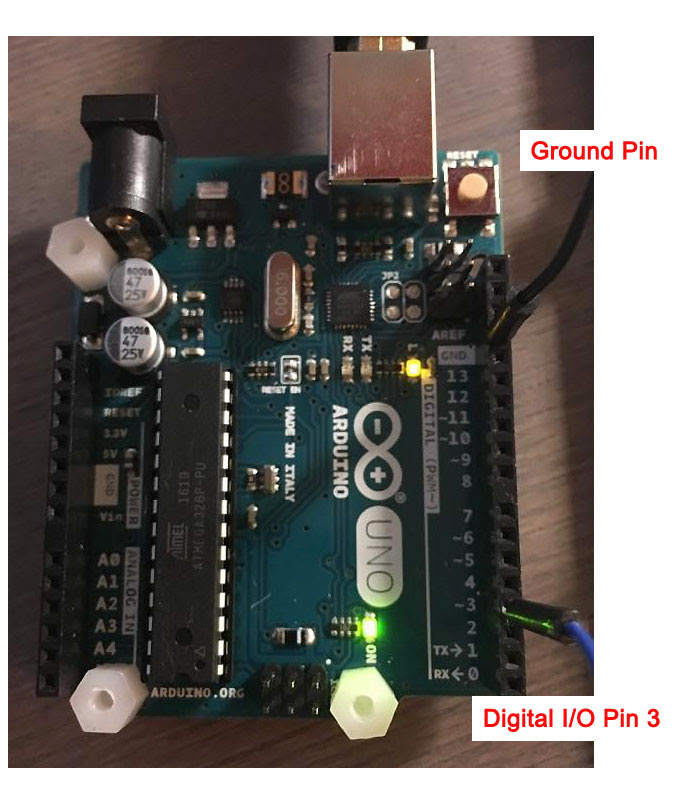
## 2) Building the Circuit

The circuit we are building is really simple. I am using a breadboard to make the circuit; feel free to solder the components together or to make a shield out of a protoboard for your Arduino. I like to make Fritzing schematics of my circuits before building them (Figure 1: Fritzing Schematic). Fritzing is an open-source schematic capture and PCB routing software.



**Figure 1: Fritzing Schematic**

The LED and resistor should be connected in series between Digital I/O Pin 3 and a ground pin (Figure 2: Arduino Board Connections). The resistor is there to limit current through the LED and should be sized accordingly depending on your LED to prevent burning it out.



**Figure 2: Arduino Board Connections**

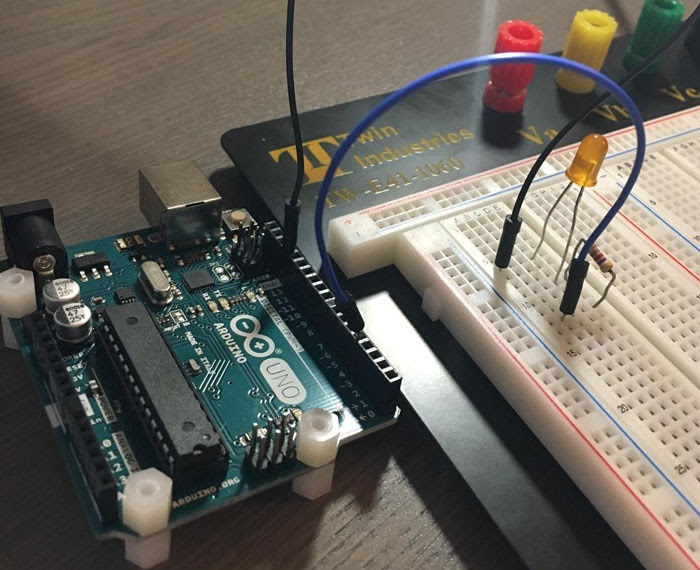
****

Figure 3: LED and Resistor on the Breadboard

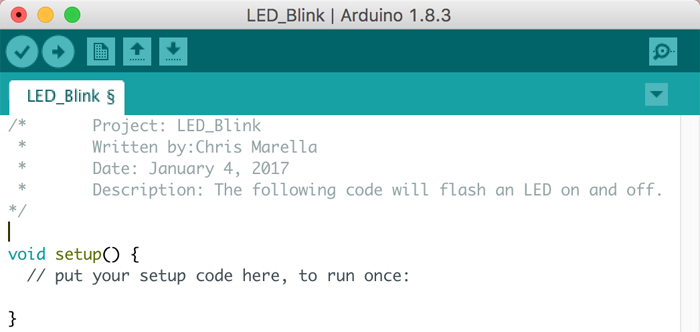
## 3) Writing an Arduino Sketch

Once you’ve successfully installed Arduino IDE, it’s time to start coding. The source code files for Arduino are called sketches. The Arduino programming language is based off C/C++ and is very similar. Open Arduino IDE and a new blank sketch will appear on your screen (Figure 4: New Arduino Sketch).



**Figure 4: New Arduino Sketch**

The sketch is divided into two program parts: a) setup and b) loop. I like to add a header to all my source code, giving the code a title, date, description, and version if necessary (Figure 5: Source Code Header).

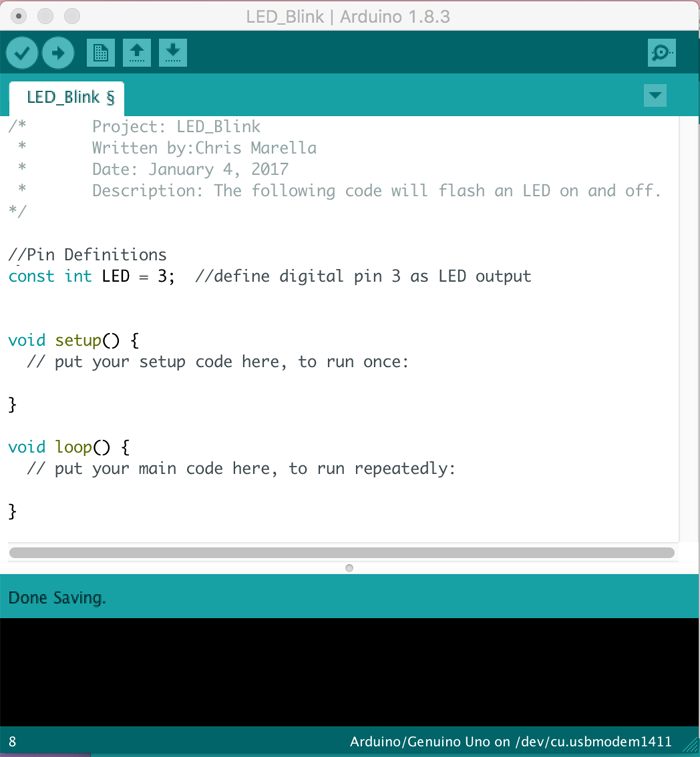


**Figure 5: Source Code Header**

The next step would be to include any necessary libraries, but since our code does not use any libraries we can skip this step.

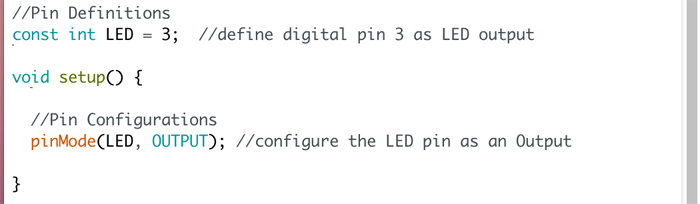
Next you will write any global variable definitions. This step is not vital. However, when working on more complex code, making these definitions can simplify the code and make it easier to edit. When writing a variable definition, you are assigning a value to a variable. In this case I will define Digital I/O Pin 3 on my Arduino Uno as my LED output pin (Figure 6: LED Pin Definition).

In the future if I want to change the pin that outputs to the LED I only have to change this definition; I will not have to change any other code.



**Figure 6: LED Pin Definition**

The setup part of your code is where you make necessary hardware and software configurations. This part of the code runs only once. Since we are driving an LED, we configure the digital I/O pin we have our LED tied to as an output pin (Figure 7: Pin Configuration).

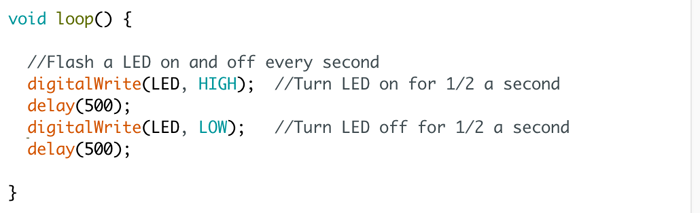


**Figure 7:Pin Configuration**

Now that we have finished setting up the Arduino we can write the main body of the code. This will go under the loop section and will repeat over and over unless otherwise stated or until power is removed from the Arduino.

To flash the LED on and off every second we write the following commands (Figure 8: LED Flash Loop):  
1) Turn LED On  
2) Wait ½ of a second (500 milliseconds)  
3) Turn LED Off  
4) Wait ½ of a second  
5) Repeat

Since the code we write is within the loop function, the Arduino will automatically repeat the code over and over.



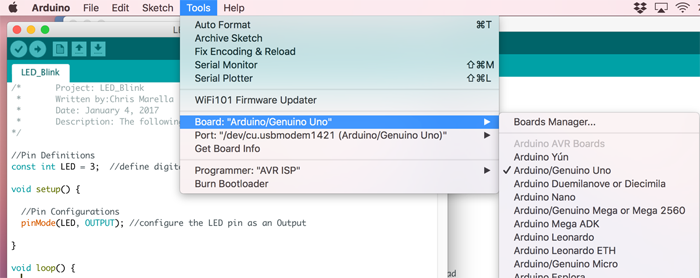
**Figure 8: LED Flash Loop**

## 4) Uploading an Arduino Sketch

Connect the Arduino board to your computer through USB. Once the Arduino is connected, follow these steps to upload the sketch:

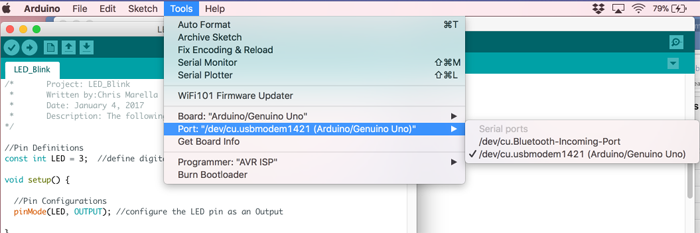
1) Select the target board (Figure 9: Target Board Selection)

* The target board selection tells Arduino IDE which Arduino board you are uploading to
* Note: If you have a different board then the Arduino Uno, select that board



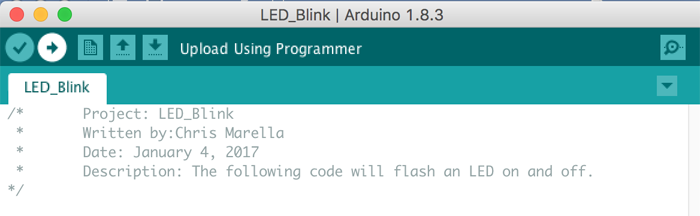
**Figure 9: Target Board Selection**

2) Select the serial port the board is connected to (Figure 10: Serial Port Connection)



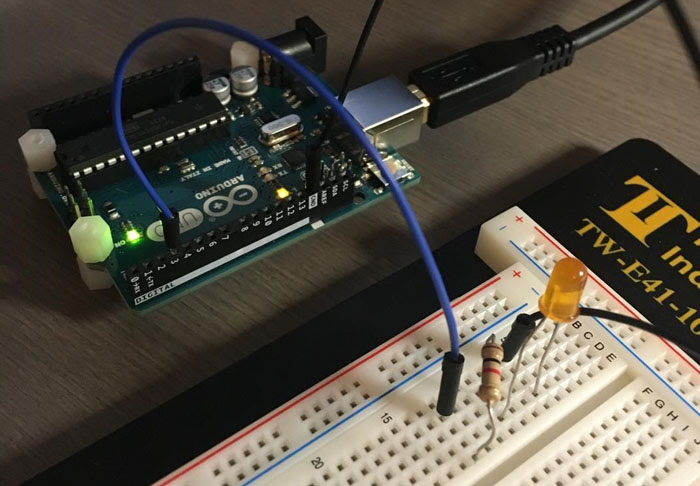
**Figure 10: Serial Port Connection**

3) Press the “Upload” button to upload the sketch to the Arduino (Figure 11: Upload Button Location)

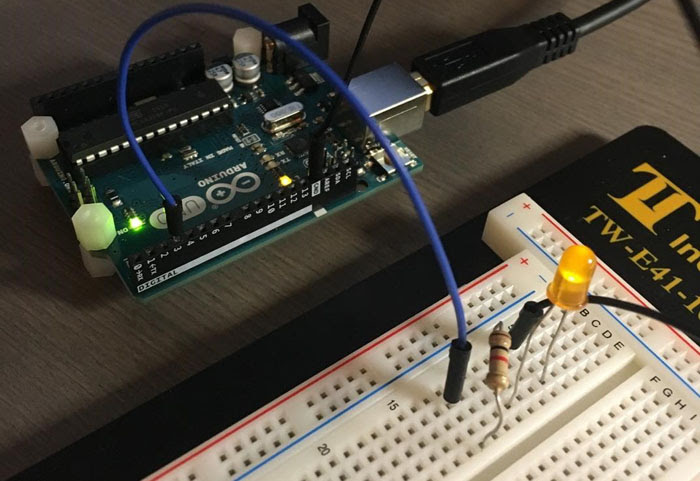


**Figure 11: Upload Button Location**

Congratulations! Your LED should now be flashing OFF (Figure 12: LED Off) and ON (Figure 13: LED On) every second. You have just installed Arduino IDE and used it to successfully write and upload your first Arduino sketch.



**Figure 12: LED Off**

****

**Figure 13: LED On**

**3.7.b) ThingSpeak**

ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB® code in ThingSpeak you can perform online analysis and processing of the data as it comes in. ThingSpeak is often used for prototyping and proof of concept IoT systems that require analytics.

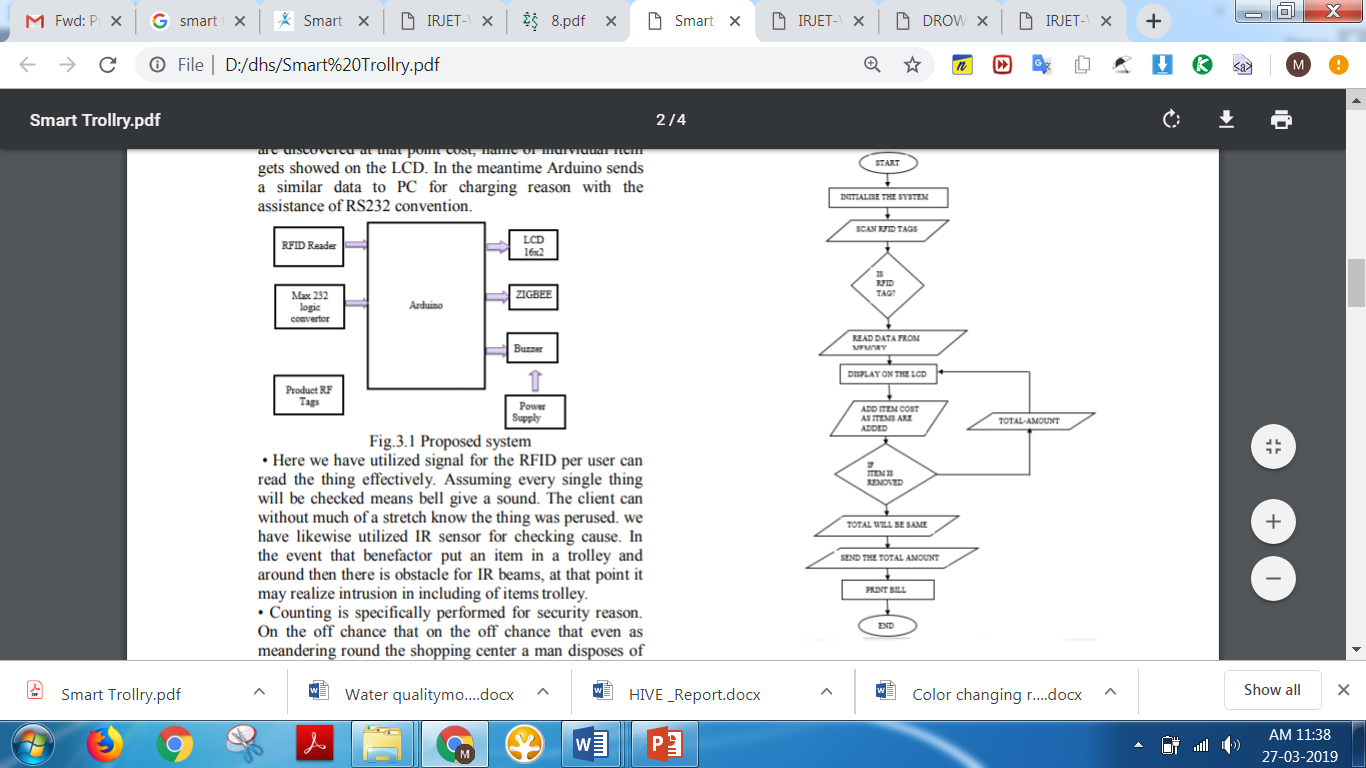
# 3.7.c)ThingSpeak Key Features

ThingSpeak allows you to aggregate, visualize and analyze live data streams in the cloud. Some of the key capabilities of ThingSpeak include the ability to:

* Easily configure devices to send data to ThingSpeak using popular IoT protocols.
* Visualize your sensor data in real-time.
* Aggregate data on-demand from third-party sources.
* Use the power of MATLAB to make sense of your IoT data.
* Run your IoT analytics automatically based on schedules or events.
* Prototype and build IoT systems without setting up servers or developing web software.
* Automatically act on your data and communicate using third-party services like Twilio® or Twitter®.

To learn how you can collect, analyze and act on your IoT data with ThingSpeak, explore the topics below:

**3.8 FLOW CHART**



**Algorithm**

* + Start the process
  + Initialize the system
  + Scan an item in RFID tags
  + Check the RFID tags
  + If the tag is registered or scanned, RFID reader can read the data related from memory
  + Display the data and cost with help of LCD
  + The item is added automatically the item cost also add and produce the total cost
  + If any item is removed, the total cost is subtracted by the particular removed item and again the process will be continuing
  + Send the total amount in the billing system
  + Print the bill
  + The process is end.

**3.9 SUMMARY**

The chapter summarizes the basic requirements of hardware and software of the project .

Introduction to Arduino IDE and its steps and gives an brief idea about IOT(Thinkspeak).

And explain the algorithm steps.