WI-FI BASED WIRELESS ADVANCED HOME AUTOMATION SYSTEM

A Project Report

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

N.CHETHAN REDDY	169L1A0440
P.VAMSI KRISHNA RAJU	179L5A0422
M.UTTEJ KUMAR	169L1A0430
S.GANESH	169L1A0456
A.PAVAN KUMAR	169L1A0401

Under the esteemed guidance of

Mr. D. VIJAYA KUMAR REDDY M. Tech.,
Assistant Professor, Department of ECE

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

Near C. Gollapalli, Tirupati - 517 505

APRIL-2020

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS
Near C. Gollapalli, Tirupati - 517 505

CERTIFICATE

This is to certify that the project entitled "**WI-FI BASED WIRELESS ADVANCED HOME AUTOMATION SYSTEM**" *is a bonafide work carried out by*

N.CHETHAN REDDY	**169L1A0440**
P.VAMSI KRISHNA RAJU	**179L5A0422**
M.UTTEJ KUMAR	**169L1A0430**
S.GANESH	**169L1A0456**
A.PAVAN KUMAR	**169L1A0401**

B. Tech students of SEAGI , Affiliated to JNTUA, Anantapuramu in partial fulfillment of the requirements for the award of the Degree of **BACHELOR OF TECHNOLOGY** *with the specialization in* **ELECTRONICS & COMMUNICATION ENGINEERING** *during the Academic year 2019-2020.*

Project guide
Mr. D. VIJAYA KUMAR REDDY,
Assistant. Professor, Dept. of ECE,
SEAGI, Tirupati – 517 505.

Head of the department
Dr. K. PURUSHOTHAM PRASAD,
Associate Professor, Dept. of ECE,
SEAGI, Tirupati – 517 505.

Viva-Voice Conducted on___________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SIDDARTHA EDUCATIONAL ACADEMY GROUP OF INSTITUTIONS

Near C. Gollapalli, Tirupati - 517 505

Declaration

We hereby declare that the project work entitled "**WI-FI BASED WIRELESS ADVANCED HOME AUTOMATION SYSTEM**" is entirely our original work carried out under the guidance of **Mr. D.VIJAYA KUMAR REDDY,** Assistant Professor, Department Electronics and Communication Engineering, Siddartha Educational Academy Group of Institutions, C.Gollapalli, Tirupati, Affiliated to JNTUA, Anantapuramu, A.P, India for the award of the degree of **BACHELOR OF TECHNOLOGY** with the specialization in **ELECTRONICS AND COMMUNICATION ENGINEERING**. The results carried out in this project report have not beensubmitted in a part or full for the award of any degree or diploma of this or any other university or institute.

N.CHETHAN REDDY	**169L1A0440**
P.VAMSI KRISHNA RAJU	**179L5A0422**
M.UTTEJ KUMAR	**169L1A0430**
S.GANESH	**169L1A0456**
A.PAVAN KUMAR	**169L1A0401**

ACKNOWLEDGEMENT

All endeavors over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

We would like to express deep sense of gratitude to our beloved and respected guide **Mr. D VIJAYA KUMAR REDDY**., Assistant Professor, Department of ECE , Siddartha Educational Academy Group of Institutions, Tirupati, for his valuable guidance, suggestions and keen interest enriched throughout the course of project work and encouragement given to us for the successful completion of work.

Our sincere thanks to **Dr. K. PURUSHOTHAM PRASAD**, **Head,** Department of Electronics and Communication Engineering for his valuable advice, guidance and encouragement given to us for the successful completion of this work.

We extend sincere thanks to the **Principal**, Prof. **K. RAJASEKHAR** for his kind co-operation in completing and making the project a success.We would like to thank the **Management** for their kind co-operation and for providing infrastructure facilities.

We extend thanks to all the **Teaching staff** of the Department of ECE for their support and encouragement during the course of our project work. We also thank the **Non-Teaching staff** of ECE department for being helpful in many ways in successful completion of our work.

Finally we thank all those who helped us directly or indirectly in successful completion of this project work.

N.CHETHAN REDDY	**169L1A0440**
P.VAMSI KRISHNA RAJU	**179L5A0422**
M.UTTEJ KUMAR	**169L1A0430**
S.GANESH	**169L1A0456**
A.PAVAN KUMAR	**169L1A0401**

ABSTRACT

The Home Automation System is extension of current activities performed inside the home and this Home Automation System (HAS) can be developed by using powerful computational devices and wireless sensor network (WSN), to provide user friendly and cost fairly home automation system.In this home automation system we are using Wi-Fi technology for communication, and different devices like smart phone, tablet and laptop used for controlling various appliances.

Modern technology Home Automation System is become very useful for handicapped people. It is very useful to the user for control and handles all the appliances that are connected to the system, from controlling devices. “Easy use of appliances” is main motive of this system. In this system home appliances can be monitored and controlled, and the user can interact with the system through a user-friendly interface. The home appliances like fans, lights, switches are remotely controlled through a main control board. By using of the Internet of Things (lot), the developing of home automation are going to become simpler and more popular. Internet of Things (IOT) is nothing but connecting different real world objects to provide proper communication, synchronization, and inter-connecting between various devices or physical appliances is also known as "Things".

We use arduino board as controller to control all the appliances. Relay board and Wi-Fi module is connected to arduino board. Each command is processed by arduino board and control the relay board for switching on/off the appliances.

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
EMBEDDED SYSTEMS

1.1 INTRODUCTION TO EMBEDDED SYSTEMS

An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, sometimes with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use.

Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

Physically embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure.

In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems -such as the operating systems and microprocessors which power them -but are not truly embedded systems, because they allow different applications to be load and peripherals to be connected.

An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular kind of application device. Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines, and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system. Embedded systems that are programmable are provided with a programming interface, and embedded systems programming is a specialized occupation. Certain operating systems or language platforms are tailored for the embedded market, such as Embedded Java and Windows XP Embedded. However, some low-end consumer products use very inexpensive microprocessors and limited storage, with the application and operating system both part of a single program.

1.2 CHARACTERISTICS OF EMBEDDED SYSTEM

• Speed (bytes/sec)	:	Should be high speed
• Power (watts)	:	Low power dissipation
• Size and weight	:	As far as possible small in size and low weight
• Accuracy (%error)	:	Must be very accurate
• Adaptability	:	High adaptability and accessibility
• Reliability	:	Must be reliable over a long period of time

1.3 APPLICATIONS OF EMBEDDED SYSTEMS

We are living in the Embedded World. You are surrounded with many embedded products and your daily life largely depends on the proper functioning of these gadgets. Television, Radio, CD player of your living room, Washing Machine or Microwave Oven in your kitchen, Card readers, Access Controllers, Palm devices of your work space enable you to do many of your tasks very effectively. Apart from all these, many controllers embedded in your car take care of car operations between the bumpers and most of the times you tend to ignore all these controllers.

• Robotics	:	industrial robots, machine tools, Robocop soccer robots
• Automotive	:	cars, trucks, trains
• Aviation	:	airplanes, helicopters
• Aerospace	:	rockets, satellites
• Energy systems	:	windmills, nuclear plants
• Medical systems	:	prostheses, re-validation machine.
• Home and Building Automation		

1.4 MICROCONTROLLER VERSUS MICROPROCESSOR

What is the difference between a Microprocessor and Microcontroller? By microprocessor is meant the general purpose Microprocessors such as Intel's X86 family (8086, 80286, 80386, 80486, and the Pentium) or Motorola's 680X0 family (68000, 68010, 68020, 68030, 68040, etc). These microprocessors contain no RAM, no ROM, and no I/O ports on the chip itself. For this reason, they are commonly referred to as general-purpose Microprocessors.

A system designer using a general-purpose microprocessor such as the Pentium or the 68040 must add RAM, ROM, I/O ports, and timers externally to make them functional. Although the addition of external RAM, ROM, and I/O ports makes these systems bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/O ports needed to fit the task at hand. This is not the case with Microcontrollers.

A Microcontroller has a CPU (a microprocessor) in addition to a fixed amount of RAM, ROM, I/O ports, and a timer all on a single chip. In other words, the processor, the RAM, ROM, I/O ports and the timer are all embedded together on one chip; therefore, the designer cannot add any external memory, I/O ports, or timer to it. The fixed amount of on-chip ROM, RAM, and number of I/O ports in Microcontrollers makes them ideal for many applications in which cost and space are critical.

1.5 MICROCONTROLLERS FOR EMBEDDED SYSTEMS

In the Literature discussing microprocessors, we often see the term Embedded System. Microprocessors and Microcontrollers are widely used in embedded system products. An embedded system product uses a microprocessor (or Microcontroller) to do one task only. A printer is an example of embedded system since the processor inside it performs one task only; namely getting the data and printing it. Contrast this with a Pentium based PC. A PC can be used for any number of applications such as word processor, print-server, bank teller terminal, Video game, network server, or Internet terminal. Software for a variety of applications can be loaded and run. Of course the reason a pc can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM memory and lets the CPU run it.

In this robot as the fire sensor senses the fire, it senses the signal to microcontroller. In an Embedded system, there is only one application software that is typically burned into ROM. An x86 PC contains or is connected to various embedded products such as keyboard, printer, modem, disk controller, sound card, CD-ROM drives, mouse, and so on. Each one of these peripherals has a Microcontroller inside it that performs only one task.

CHAPTER-2

INTRODUCTION TO PROJECT

2.1 INTRODUCTION

“Home automation” refers to the automatic and electronic control of household features, activities, and appliances. The utilities and features of our home can be easily controlled via Internet. There are three main elements of a home automation system: sensors, controllers, and actuators. Home is the place where one desires to be rest after a long tiring day. People come home exhausted after a long hard-working day. Some are way too tired that they find it hard to move once they land on their couch, sofa or bed. So, any small device/technology that would help them switch theirs lights on or off, or play their favorite music etc. on a go with their voice with the aid of their smart phones would make their home more comfortable. Moreover, it would be better if everything such as warming bath water and adjusting the room temperature were already done before they reach their home just by giving a voice command.So, when people would arrive home, they would find the room temperature, the bath water adjusted to their suitable preferences, and they could relax right away and feel cozier and rather, feel more homely. Human assistants like housekeepers were a way for millionaires to keep up their homes in the past. Even now when technology is handy enough only the well to do people of the society are blessed with their new smart home devices, as these devices costs are a bit high. However, not everyone is wealthy enough to be able to afford a human assistant, or some smart home kit. Hence, the need for finding an inexpensive and smart assistant for normal families keeps growing.

2.2 WORKING

The IR sensor which we used in here will detect the objects in the surrounding’s and gives the alarm or buzzer sound.On by giving the voice commands from the android mobile through wi-fi module ,it connect to the server by using adafruit software and the object or devices are controlled through relays .The relays are used to enhance the power to the output components.

2.3 BLOCK DIAGRAM

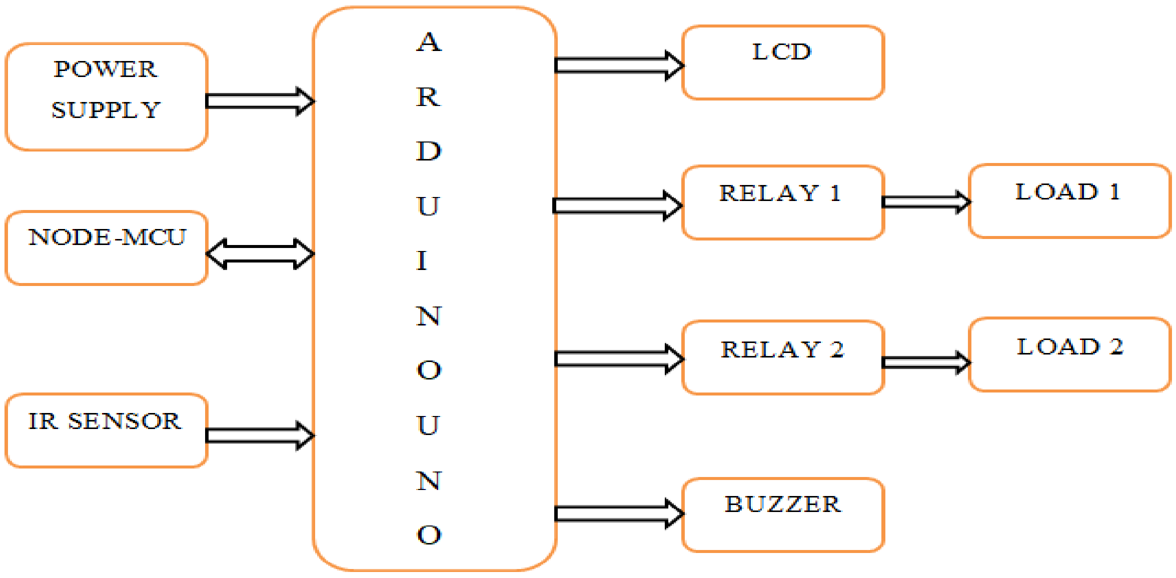


Figure 2.1: Block diagram of wi-fi based wireless advanced home automation system

2.4 APPLICATIONS

1) Home automation.

2) Industrial purpose.

3) Security purpose

2.5 Benefits

1) Control

2) Convenience

3) Safety

4) Flexible

5) Improved energy savings

CHAPTER-3
MICROCONTROLLERS

3.1 MICROCONTROLLER

3.1.1 Introduction

Microcontroller as the name suggests, a small controller. They are like single chip computers that are often embedded into other systems to function as processing/controlling unit. For example, the control you are using probably has microcontrollers inside that do decoding and other controlling functions. They are also used in automobiles, washing machines, microwaves ovens, toys….etc, where automation is needed.

3.1.2 Arduino Uno Microcontroller

The Arduino Uno is a microcontroller board based on the ATmega328 (data sheet). 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 an 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. "Uno" means "One" in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

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 center-positive plug into the board's power jack. Leads from 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 5Vpin 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 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 on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3.3V.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the boot loader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read() 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 attach Interrupt() 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 icon, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provides 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 analogReference() function. Additionally, some pins have specialized.

Functionality

- **I2C: 4 (SDA) and 5 (SCL).** Support I2C (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 analogReference().
- **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 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 USBCOM drivers, and no external driver is needed. However, on Windows,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 Software Serial library allows for serial communication on any of the Uno's digital pins. The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

3.1.3 Arduino UNO Board

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, 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.

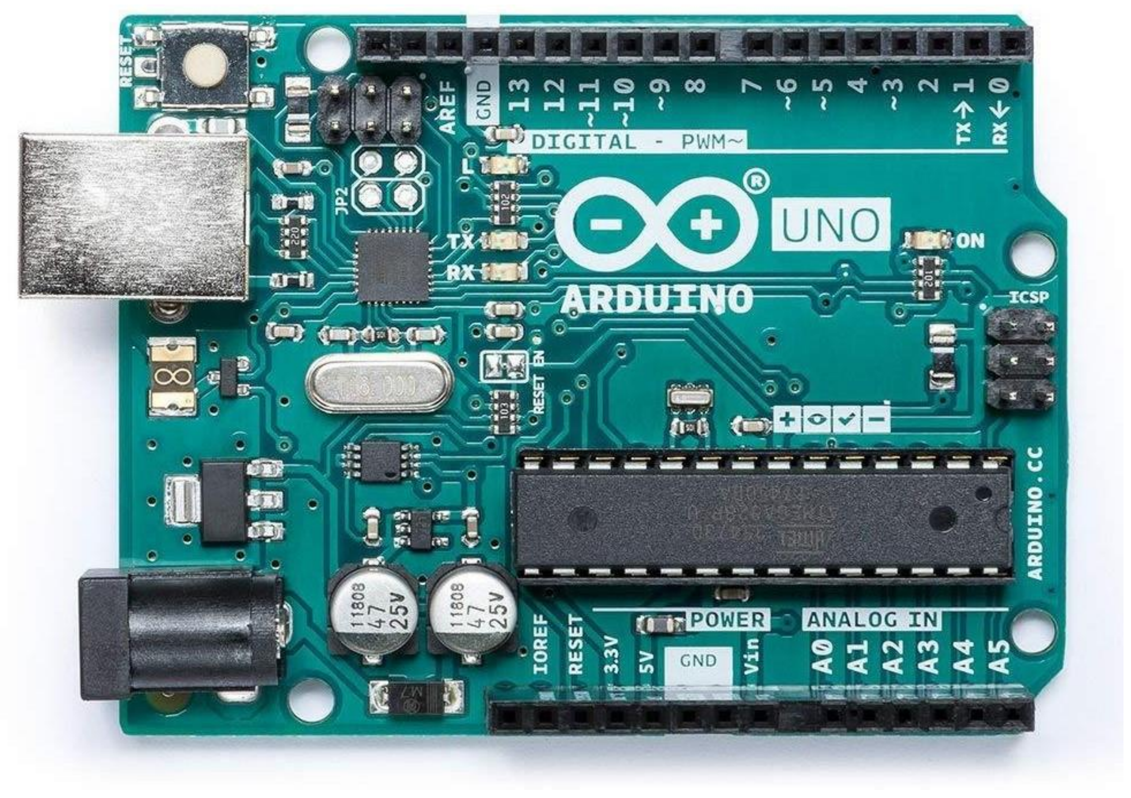


Figure 3.1: Arduino Uno board

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converters.

FEATURE	**SPECIFICATIONS**
Microcontroller	ATmega328
Operating voltage	5v
Input voltage(recommended)	7-12v
Input voltage(limits)	6-20v
Digital I/O pins	14(of which 6 provide PWM output)
Analog input pins	16
DC current per I/O pin	40 mA
DC current for 3.3v	50 mA
Flash memory	32 KB(ATmega328) of which 0.5 KB used by boot loader
SRAM	2KB(ATmega328)
EEPROM	1KB(ATmega328)
Clock speed	16 MHZ

Table 3.1: Arduino uno specifications

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 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

1. USB Interface

Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection **.**

2. External power supply

Arduino boards can be powered directly from the AC mains power supply by connecting it to the power supply (Barrel Jack)

3. Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

4. Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz

5-17.Arduino Reset

It can reset your Arduino board, i.e., starts your program from the beginning. It can reset the UNO board in two ways. First , by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labeled RESET (5).

6-9.Pins (3.3, 5, GND, VIN)

- 3.3V (6): Supply 3.3 output volt
- 5V (7): Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt And 5 volt.
- GND (8) (Ground): There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9): This pin also can be used to power the Arduino board from an External power source , like AC mains power supply.

10. Analog pins

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

11. Main microcontroller

Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

The Atmega8U2 programmed as a USB-to-serial converter. "Uno" means "One" in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards

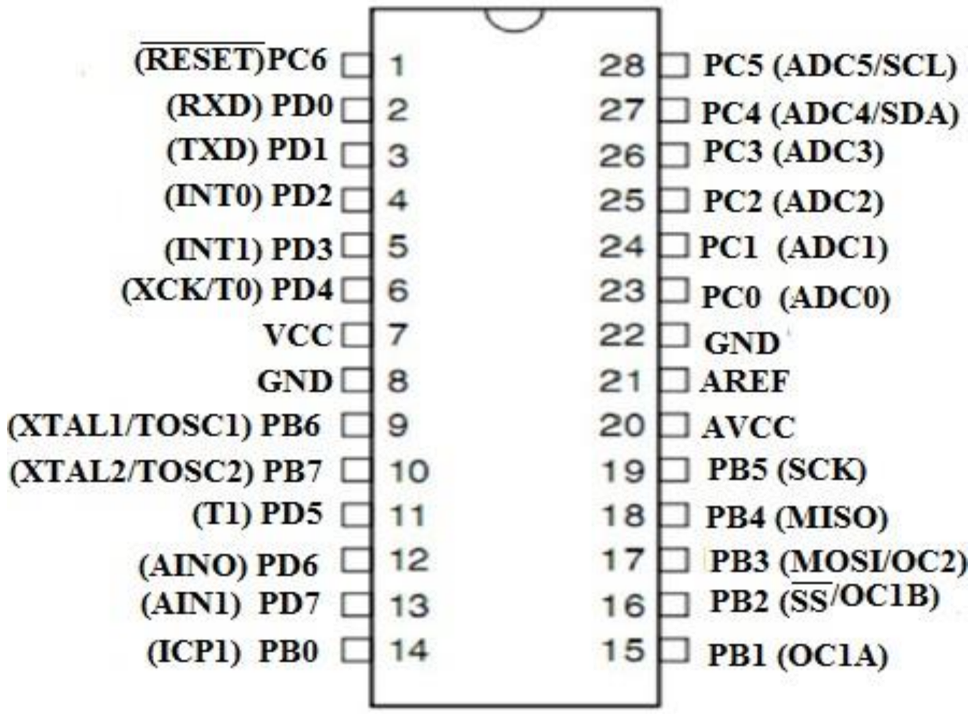


Figure 3.2: ATMEGA328 Arduino Pin diagram

Pin Description

VCC: Digital supply voltage.

GND: Ground.

Port B (PB [7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB [7:6] is used as TOSC [2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC [5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC [5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs,

Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.If the RSTDISBL Fuse is un programmed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

Port D (PD [7:0])

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

AVCC is the supply voltage pin for the A/D Converter, PC [3:0], and PE [3:2]. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC [6:4] use digital supply voltage, VCC.

AREF

AREF is the analog reference pin for the A/D Converter.

ADC [7:6] (TQFP and VFQFN Package Only)

In the TQFP and VFQFN package, ADC [7:6] serves as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

12. ICSP pin

Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.

13. Power LED indicator

This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

14. TX and RX LEDs

On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1 , to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.

15. Digital I / O

The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.

16. AREF

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

CHAPTER-4

HARDWARE COMPONENTS

4.1 POWER SUPPLY BLOCKDIAGRAM

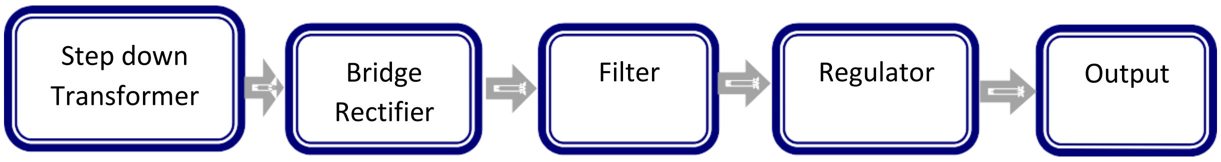


Figure 4.1: Block diagram of power supply

4.1.1 Step-Down Transformer

Step down transformers are designed to reduce electrical voltage. Their primary voltage is greater than their secondary voltage. This kind of transformer "steps down" the voltage applied to it. For instance, a step down transformer is needed to use an 110v product in a country with a 220v supply.

Step down transformers convert electrical voltage from one level or phase configuration usually down to a lower level. They can include features for electrical isolation, power distribution, and control and instrumentation applications. Step down transformers typically rely on the principle of magnetic induction between coils to convert voltage and/or current levels.

Step down transformers are made from two or more coils of insulated wire wound around a core made of iron. When voltage is applied to one coil (frequently called the primary or input) it magnetizes the iron core, which induces a voltage in the other coil, (frequently called the secondary or output). The turn’s ratio of the two sets of windings determines the amount of voltage transformation.

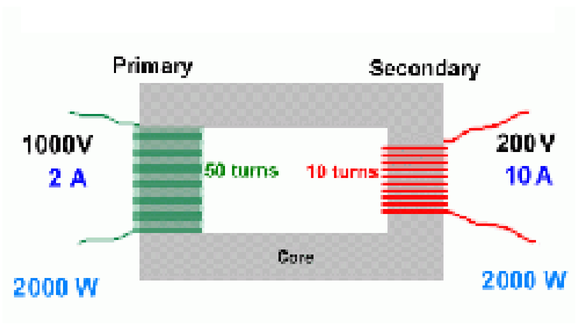


Figure 4.2: Step-Down Transformer

An example of this would be: 100 turns on the primary and 50 turns on the secondary, a ratio of 2 to 1.Step down transformers can be considered nothing more than a voltage ratio device.

With step down transformers the voltage ratio between primary and secondary will mirror the "turn’s ratio". A practical application of this 2 to 1 turn’s ratio would be a 480 to 240 voltage step down. Note that if the input were 440 volts then the output would be 220 volts. The ratio between input and output voltage will stay constant. Transformers should not be operated at voltages higher than the nameplate rating, but may be operated at lower voltages than rated. Because of this it is possible to do some non-standard applications using standard transformers.

Single phase step down transformers 1 kva and larger may also be reverse connected to step-down or step-up voltages single phase step up or step down transformers sized less than 1 KVA should not be reverse connected because the secondary windings have additional turns to overcome a voltage drop when the load is applied. If reverse connected, the output voltage will be less than desired

4.1.2 Rectifier

The purpose of a rectifier is to convert an AC waveform into a DC waveform (OR) Rectifier converts AC current or voltages into DC current or voltage. Here we use bridge rectifier A bridge rectifier circuit is a common part of the electronic power supplies. Many electronic circuits require rectified DC power supply for powering the various electronic basic components from available AC mains supply. We can find this rectifier in a wide variety of electronic AC power devices like home appliances, motor controllers, modulation process, welding applications, etc. The main advantage of bridge rectifier is that it produces almost double the output voltage as with the case of a full wave rectifier using center-tapped transformer.

The bridge rectifier circuit diagram consists of various stages of devices like transformer, Diode Bridge, filtering and regulators. Generally all these blocks combination is called as regulated DC power supply that powers various electronic appliances.The first stage of the circuit is a transformer which is a step-down type that changes the amplitude of the input voltage. Most of the electronic projects uses 230/12V transformer to step-down the AC mains 230V to 12V AC supply

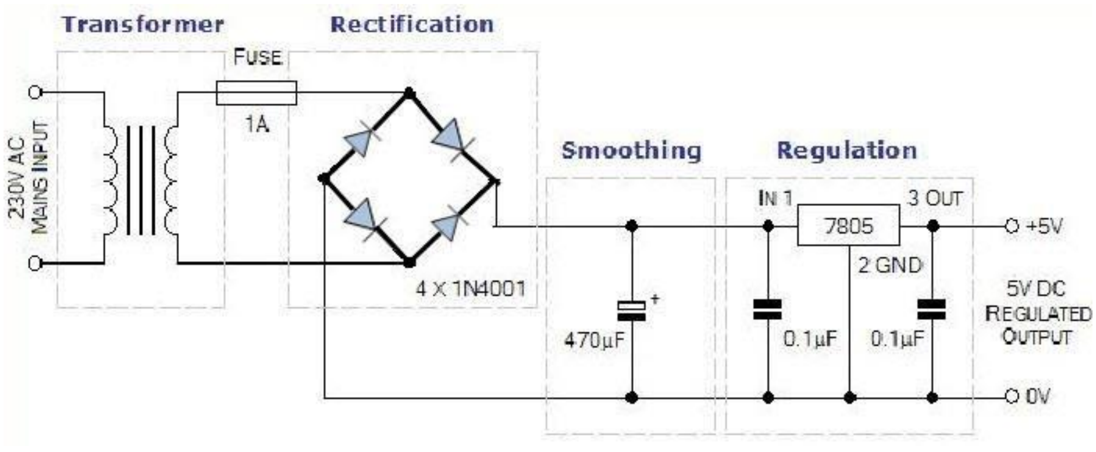


Figure 4.3: Bridge Rectifier

Next stage is a diode-bridge rectifier which uses four or more diodes depending on the type of bridge rectifier. Choosing a particular diode or any other switching device for a corresponding rectifier needs some considerations of the device like Peak Inverse Voltage (PIV), forward current If, voltage ratings, etc. It is responsible for producing unidirectional or DC current at the load by conducting a set of diodes for every half cycle of the input signal.

Since the output after the diode bridge rectifiers is of pulsating nature, and for producing it as a pure DC, filtering is necessary. Filtering is normally performed with one or more capacitors attached across the load, as you can observe in the below figure wherein smoothing of wave is performed. This capacitor rating also depends on the output voltage.

The last stage of this regulated DC supply is a voltage regulator that maintains the output voltage to a constant level. Suppose the microcontroller works at 5V DC, but the output after the bridge rectifier is around 16V, so to reduce this voltage, and to maintain a constant level – no matter voltage changes in input side – a voltage regulator is necessary.

Bridge Rectifier Operation

As we discussed above, a single-phase bridge rectifier consists of four diodes and this configuration is connected across the load. For understanding the bridge rectifier's working principle, we have to consider the below circuit for demonstration purpose. During the Positive half cycle of the input AC waveform diodes D1 and D2 are forward biased and D3 and D4 are reverse biased. When the voltage, more than the threshold level of the diodes D1 and D2, starts conducting – the load current starts flowing through it, as shown as red lines path in the diagram below.

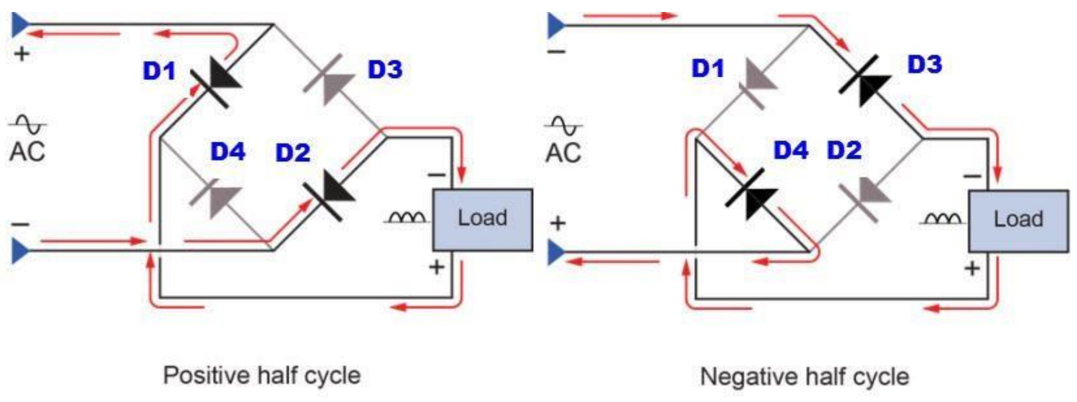


Figure 4.4: Operation of Bridge Rectifier

During the negative half cycle of the input AC waveform, the diodes D3 and D4 are forward biased, and D1 and D2 are reverse biased. Load current starts flowing through the D3 and D4 diodes when these diodes starts conducting as shown in the figure.

We can observe that in both the cases, the load current direction is same, i.e., up to down as shown in the figure so unidirectional, which means DC current. Thus, by the usage of a bridge rectifier, the input AC current is converted into a DC current. The output at the load with this bridge wave rectifier is pulsating in nature, but for producing a pure DC requires additional filter like capacitor. This is all about the bridge rectifier theory its types, circuit and working principles. We hope that this wholesome matter about this topic will be helpful in building students' electronics or electrical projects as well as in observing various electronic devices or appliances. We appreciate you for your keen attention and focus on this article.

And therefore, please do write to us for choosing required component ratings in this bridge rectifier for your application and for any other technical guidance.

4.1.3 Capacitor Filter

The capacitor-input filter, also called "Pi" filter due to its shape that looks like the Greek letter pi, is a type of electronic filter. Filter circuits are used to remove unwanted or undesired frequencies from a signal.

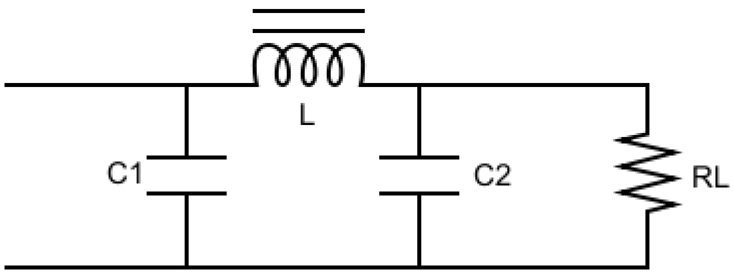


Figure 4.5: Capacitor Filter

A typical capacitor input filter consists of a filter capacitor C1, connected across the rectifier output, an inductor L, in series and another filter capacitor connected across the load.

1. The capacitor C1 offers low reactance to the AC component of the rectifier output while it offers infinite reactance to the DC component. As a result the capacitor shunts an appreciable amount of the AC component while the DC component continues its journey to the inductor L
2. The inductor L offers high reactance to the AC component but it offers almost zero reactance to the DC component. As a result the DC component flows through the inductor while the AC component is blocked.
3. The capacitor C2 bypasses the AC component which the inductor had failed to block. As a result only the DC component appears across the load RL.

4.1.4 Voltage Regulator

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. Depending on the design, it may be used to regulate one or more AC or DC voltages.

There are two types of regulators.They are

- Positive Voltage Series (78xx) and
- Negative Voltage Series (79xx)

78xx:'78' indicate the positive series and 'xx' indicates the voltage rating. Suppose 7805 produces the maximum 5V.'05'indicates the regulator output is 5V.

79xx:'78' indicate the negative series and 'xx' indicates the voltage rating. Suppose 7905 produces the maximum -5V.'05'indicates the regulator output is -5V.

AMS1117 Voltage Regulator

The AMS1117 series of adjustable and fixed regulators are easy to use and are protected against short circuit and thermal overloads. Thermal protection circuitry will shut-down the regulator should the junction temperature exceed 165°C at the sense point.Pin compatible with older three terminal adjustable regulators, these devices offer the advantage of a lower dropout voltage, more precise reference tolerance and improved reference stability with temperature.

Stability

The circuit design used in the AMS1117 series requires the use of an output capacitor as part of the device frequency compensation. The addition of 22mF solid tantalum on the output will ensure stability for all operating conditions.

When the adjustment terminal is bypassed with a capacitor to improve the ripple rejection, the requirement for an output capacitor increases. The value of 22mF tantalum covers all cases of bypassing the adjustment terminal. Without bypassing the adjustment terminal smaller capacitors can be used with equally good results.

To ensure good transient response with heavy load current changes capacitor values on the order of 100mF are used in the output of many regulators. To further improve stability and transient response of these devices larger values of output capacitor can be used.

4.1.5 Output Voltage

The AMS1117 series develops a 1.25V reference voltage between the output and the adjust terminal. Placing a resistor between these two terminals causes a constant current to flow through R1 and down through R2 to set the overall output voltage. This current is normally the specified minimum load current of 10mA. Because I_{ADJ} is very small and constant it represents a small error and it can usually be ignored.

Load Regulation

True remote load sensing it is not possible to provide, because the AMS1117 is a three terminal device. The resistance of the wire connecting the regulator to the load will limit the load regulation. The data sheet specification for load regulation is measured at the bottom of the package. Negative side sensing is a true Kelvin connection, with the bottom of the output divider returned to the negative side of the load.

The best load regulation is obtained when the top of the resistor divider R1 is connected directly to the case not to the load. If R1 were connected to the load, the effective resistance between the regulator and the load would be.

4.2 IR (INFRARED) Sensor:

What Is INFRARED?

Infrared is an energy radiation with a frequency below our eyes sensitivity, so we cannot see it. Even that we cannot "see" sound frequencies, we know that it exist, we can listen them. Even that we cannot see or hear infrared, we can feel it at our skin temperature sensors. When you approach your hand to fire or warm element, you will "feel" the heat, but you can't see it. You can see the fire because it emits other types of radiation, visible to your eyes, but it also emits lots of infrared that you can only feel in your skin.

Introduction

An infrared sensor is an electronic device that emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation , rather than emitting it that is called as a passive IR sensor . Usually in the infrared spectrum, all the objects radiate some form of thermal radiations.

The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received.

IR Sensor Circuit Diagram and Working Principle

An infrared sensor circuit is one of the basic and popular sensor modules in an electronic device. This sensor is analogous to human’s visionary senses, which can be used to detect obstacles and it is one of the common applications in real time. This circuit comprises of the following components

- LM358 IC 2 IR transmitter and receiver pair
- Resistors of the range of kilo ohms.
- LED (Light Emitting Diode).

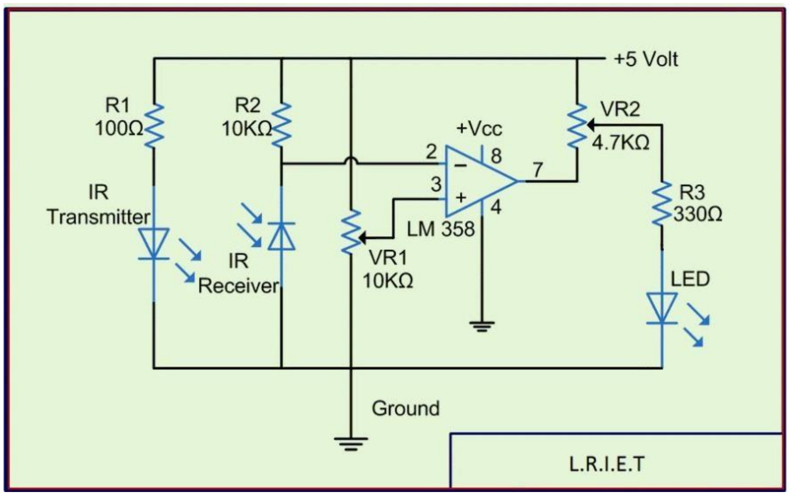


Figure 4.6: IR sensor circuit

In this project, the transmitter section includes an IR sensor, which transmits continuous IR rays to be received by an IR receiver module. An IR output terminal of the receiver varies depending upon its receiving of IR rays. Since this variation cannot be analyzed as such, therefore this output can be fed to a comparator circuit. Here an operational amplifier (op-amp) of LM 339 is used as comparator circuit.

When the IR receiver does not receive a signal, the potential at the inverting input goes higher than that non-inverting input of the comparator IC (LM339). Thus the output of the comparator goes low, but the LED does not glow. When the IR receiver module receives signal to the potential at the inverting input goes low. Thus the output of the comparator (LM 339) goes high and the LED starts glowing. Resistor R1 (100), R2 (10k) and R3 (330) are used to ensure that minimum 10 mA current passes through the IR LED Devices like Photodiode and normal LEDs respectively.

4.3 ESP8266 (NODEMCU)

The ESP8266 Wi-Fi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor. Each ESP8266 module comes pre-programmed with an AT command set firmware, meaning, you can simply hook this up to your Arduino device and get about as much Wi-Fi ability as a Wi-Fi Shield offers (and that's just out of the box)! The ESP8266 module is an extremely cost effective board with a huge, and ever growing, community.

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. Its high degree of on-chip integration allows for minimal external circuitry, including the front-end module, is designed to occupy minimal PCB area. The ESP8266 supports APSD for VoIP applications and Bluetooth co-existance interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions, and requires no external RF parts.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support.

Features:

- 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLLs, regulators, DCXO and power management units
- +19.5dBm output power in 802.11b mode
- Power down leakage current of <10uA
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 1.1 / 2.0, SPI, UART

- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4ms guard interval
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- Default baud-rate: 115200

4.4 LCD (Liquid Crystal Display)

A liquid crystal display (LCD) is a thin, flat display device made up of any number of color or monochrome pixels arrayed in front of a light source or reflector. Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other.

Some of the most common LCDs connected to the controllers are 16X1, 16x2 and 20x2 displays. This means 16 characters per line by 1 line 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.Many microcontroller devices use ‘smart LCD’ displays to output visual Information. LCD displays designed around LCD NT-C1611 module, are inexpensive, easy to use, and it is even possible to produce a readout using the 5X7 dots plus cursor of the display.They have a standard ASCII set of characters and mathematical symbols. For an 8-bit data bus, the display requires a +5V supply plus 10 I/O lines (RS, RW, D7, D6, D5, D4, D3, D2, D1, D0).For a 4-bit data bus it only requires the supply lines plus 6 extra lines (RS, RW, D7, D6, D5, and D4). When the LCD display is not enabled, data lines are tri-state and they do not interfere with the operation of the microcontroller.

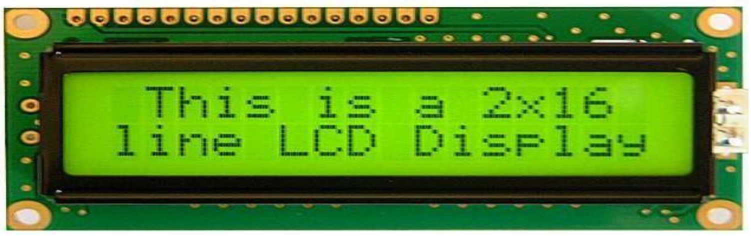


Figure 4.7: 2x16 LCD Display

4.4.1 Uses

The LCD s used exclusively in watches, calculators and measuring instruments is the simple seven-segment displays, having a limited amount of numeric data. The recent advances in technology have resulted in better legibility, more information displaying capability and a wider temperature range. These have resulted in the LCD s being extensively used in telecommunications and entertainment electronics. The LCD s has even started replacing the cathode ray tubes (CRTs) used for the display of text and graphics, and also in small TV applications.

4.4.2 LCD PIN Diagram

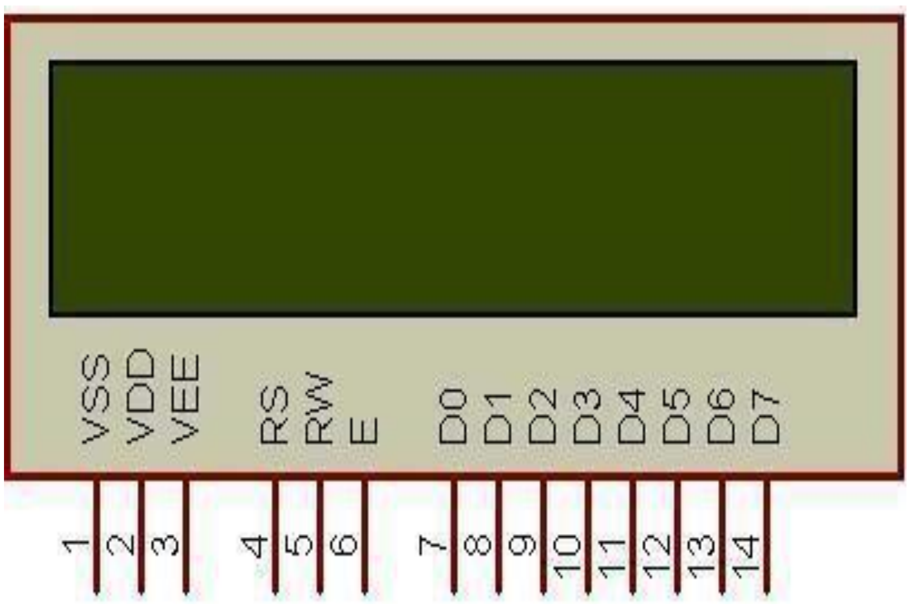


Figure 4.8: Pin Diagram of LCD

PIN	SYMBOL	FUNCTION
1	VSS	Power Supply(GND)
2	VDD	Power Supply(+5V)
3	V_O	Contrast Adjust
4	RS	Instruction/data register select
5	R/W	Data Bus Line
6	E	Enable Signal
7-14	DB0-DB7	Data Bus Line
15	A	Power supply for LED B/L(+)
16	K	Power Supply for LED B/L(-)

Table 4.1: Functions of control lines

4.4.3 Control lines

EN

Line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring EN high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again. Line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.).When RS is high (1), the data being sent is text data which should be displayed on the screen.

RW

Line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands, so RW will almost always be low.

4.4.4 Logic status on control lines

★ R/W - 0 Writing data to LCD.
- 1 Reading data from LCD.

★ RS - 0 Instructions.
-1 Character.

4.4.5 Contrast control

To have a clear view of the characters on the LCD, contrast should be adjusted. To adjust the contrast, the voltage should be varied. For this, a preset is used which can behave like a variable voltage device. As the voltage of this preset is varied, the contrast of the LCD can be adjusted.

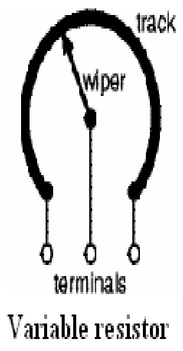


Figure 4.9: variable resistor

4.5 RELAYS

A relay is an electrically controllable switch widely used in industrial controls, automobiles and appliances. The relay allows the isolation of two separate sections of a system with two different voltage sources i.e., a small amount of voltage/current on one side can handle a large amount of voltage/current on the other side but there is no chance that these two voltages mix up.

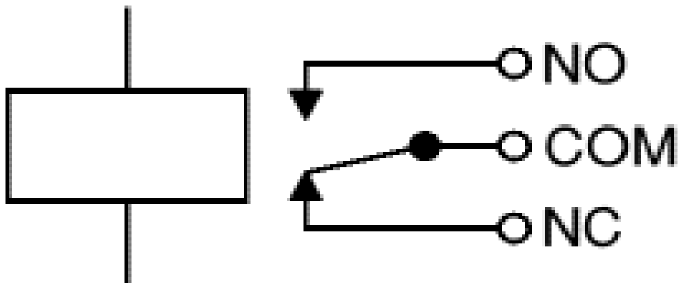


Figure 4.10: Circuit symbol of a relay

4.5.1 Operation:

When current flows through the coil, a magnetic field are created around the coil i.e., the coil is energized. This causes the armature to be attracted to the coil. The armature’s contact acts like a switch and closes or opens the circuit. When the coil is not energized, a spring pulls the armature to its normal state of open or closed. There are all types of relays for all kinds of applications.

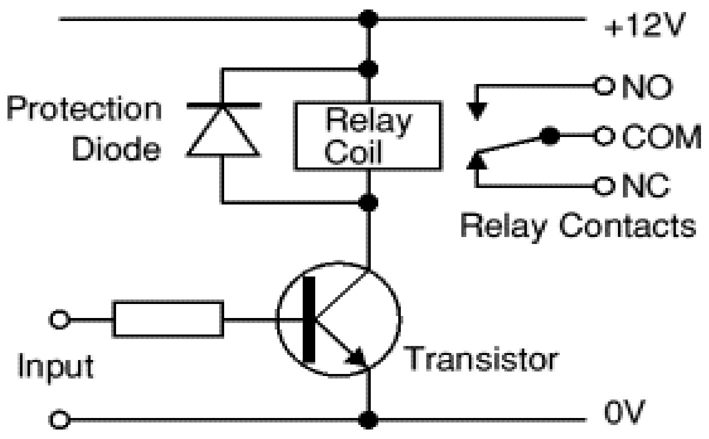


Figure 4.11: Relay Operation and use of protection diodes

Transistors and ICs must be protected from the brief high voltage 'spike' produced when the relay coil is switched off. The above diagram shows how a signal diode (egg 1N4148) is connected across the relay coil to provide this protection. The diode is connected 'backwards' so that it will normally not conduct. Conduction occurs only when the relay coil is switched off, at this moment the current tries to flow continuously through the coil and it is safely diverted through the diode. Without the diode no current could flow and the coil would produce a damaging high voltage 'spike' in its attempt to keep the current flowing.

In choosing a relay, the following characteristics need to be considered:

1. The contacts can be normally open or normally closed In the NC type, the contacts are closed when the coil is not energized. In the NO type, the contacts are closed when the coil is energized.

2. There can be one or more contacts. i.e., different types like SPST (single pole single throw), SPDT (single pole double throw) and DPDT (double pole double throw) relay.

3. The voltage and current required to energize the coil. The voltage can vary from a few volts to 50 volts, while the current can be from a few milli-amps to 20 milli-amps. The relay has a minimum voltage, below which the coil will not be energized. This minimum voltage is called the “pull-in” voltage.

4. The minimum DC/AC voltage and current that can be handled by the contacts. This is in the range of a few volts to hundreds of volts, while the current can be from a few amps to 40A or more, depending on the relay.

4.5.2 Driving A RELAY

An SPDT relay consists of five pins, two for the magnetic coil, one as the common terminal and the last pins as normally connected pin and normally closed pin. When the current flows through this coil, the coil gets energized. Initially when the coil is not energized, there will be a connection between the common terminal and normally closed pin. But when the coil is energized, this connection breaks and a new connection between the common terminal and normally open pin will be established. Thus when there is an input from the microcontroller to the relay, the relay will be switched on. Thus when the relay is on, it can drive the loads connected between the common terminals and normally open pin. Therefore, the relay takes 5V from the microcontroller and drives the loads which consume high currents. Thus the relay acts as an isolation device.

4.6 INTRODUCTION TO INTERNET OF THINGS

4.6.1 Introduction

The Internet of Things may be a hot topic in the industry but it’s not a new concept. In the early 2000’s, Kevin Ashton was laying the groundwork for what would become the Internet of Things (IOT) at MIT’s Auto ID lab. Ashton was one of the pioneers who conceived this notion as he searched for ways that Proctor & Gamble could improve its business by linking RFID information to the Internet. The concept was simple but powerful

“If we had computers that knew everything there was to know about things using data they gathered without any help from us -- we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best. We need to empower computers with their own means of gathering information, so they can see, hear and smell the world for themselves, in all its random glory. RFID and sensor technology enable computers to observe identify and understand the world—without the limitations of human-entered data.”

Today, many of these obstacles have been solved. The size and cost of wireless radios has dropped tremendously. IPv6 allows us to assign a communications address to billions of devices. Electronics companies are building Wi-Fi and cellular wireless connectivity into a wide range of devices. IOT describes a system where items in the physical world, and sensors within or attached to these items, are connected to the Internet via wireless and wired Internet connections. These sensors can use various types of local area connections such as RFID, NFC, Wi-Fi, Bluetooth, and Zigsbee. Sensors can also have wide area connectivity such as GSM, GPRS, 3G, and LTE.

The Internet of Things will

• Connect both inanimate and living things.

• Use sensors for data collection.

• Change what types of item communicate over an IP Network.

4.6.2 Advantages

Here are some advantages of IOT

1. **Data**- The more the information, the easier it is to make the right decision. Knowing what to get from the grocery while you are out, without having to check on your own, not only saves time but is convenient as well.

2. **Tracking-** The computers keep a track both on the quality and the viability of things at home. Knowing the expiration date of products before one consumes them improves safety and quality of life. Also, you will never run out of anything when you need it at the last moment.

3. **Time**- The amount of time saved in monitoring and the number of trips done otherwise would be tremendous.

4. **Money-** The financial aspect is the best advantage. This technology could replace humans who are in charge of monitoring and maintaining supplies.

4.6.3 The Internet of Things applications

- Smart home.
- Smart city.
- Smart grids.
- Industrial internet.

CHAPTER-5

SOFTWARE REQUIREMENTS

5.1 INTRODUCTION TO ARDUINO IDE

Arduino is a prototype platform based on an easy-to-use hardware and software. It consists of a circuit board, which can be programmed and ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are

- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware in order to load a new code onto the board. You can simply use a USB cable.
- the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

After learning about the main parts of the Arduino UNO board, we are ready to learn how-to set up the Arduino IDE. Once we learn this, we will be ready to upload our program on the Arduino board.

5.1.1 Arduino data types

Data types in C refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in the storage and how the bit pattern stored is interpreted.

Void

The void keyword is used only in function declarations. It indicates that the function is expected to return no information to the function from which it was called.

Example

```
Void Loop ( )
{
// rest of the code
}
```

Example

Boolean state= false; // declaration of variable with type Boolean and initialize it with false.

Boolean state = true; // declaration of variable with type Boolean and initialize it with false.

Char

However, characters are stored as numbers. You can see the specific encoding in the ASCII chart. This means that it is possible to do arithmetic operations on characters, in which the ASCII value of the character is used.

Example

Char chorea = ‘a’; //declaration of variable with type char and initialize it with character a.

Char choric = 97; //declaration of variable with type char and initialize it with character 97

Unsigned char

char is an unsigned data type that occupies one byte of memory. The unsigned char data type encodes numbers from 0 to 255.

Example

Unsigned Char y = 121;

Byte

A byte stores an 8-bit unsigned number, from 0 to 255.

Example

Byte m = 25; //declaration of variable with type byte and initialize it with 25

INT

Integers are the primary data-type for number storage. **INT** stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

The **INT** size varies from board to board. On the Arduino Due , for example, an **INT** stores a 32-bit (4-byte) value. This yields a range of -2,147,483,648 to 2,147,483,647 (minimum value of -2^31 and a maximum value of (2^31) - 1).

Example

INT counter = 32; // declaration of variable with type int and initialize it with 32.

Unsigned INT

Unsigned ints (unsigned integers) are the same as int in the way that they store a 2 byte value. Instead of storing negative numbers, however, they only store positive values, yielding a useful range of 0 to 65,535 (2^16) - 1). The Due stores a 4 byte (32-bit) value, ranging from 0 to 4,294,967,295 (2^32 - 1).

Example

Unsigned int counter= 60; // declaration of variable with type unsigned int and initialize it with 60.

Word

On the Uno and other ATMEGA based boards, a word stores a 16-bit unsigned number. On the Due and Zero, it stores a 32-bit unsigned number.

Example

Word w = 1000; //declaration of variable with type word and initialize it with 1000.

Long

Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from 2,147,483,648 to 2,147,483,647.

Example

Long velocity= 102346; //declaration of variable with type Long and initialize it with 102346

Short

A short is a 16-bit data-type. On all Arduinos (ATMega and ARM based), a short stores a 16-bit (2-byte) value. This yields a range of -32,768 to 32,767 (minimum value of -2^15 and a maximum value of (2^15) - 1).

Example

Short val= 13; //declaration of variable with type short and initialize it with 13

Float

Data type for floating-point number is a number that has a decimal point. Floating-point numbers are often used to approximate the analog and continuous values because they have greater resolution than integers.

Example

Float num = 1.352;//declaration of variable with type float and initialize it with 1.352.

Double

On the Uno and other ATMEGA based boards, Double precision floating-point number occupies four bytes. That is, the double implementation is exactly the same as the float, with no gain in precision. On the Arduino Due, doubles have 8-byte (64 bit) precision.

Example

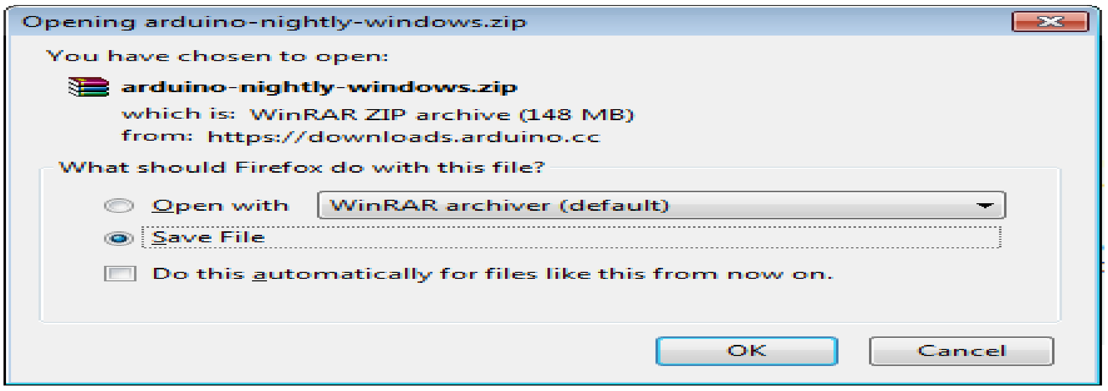
Double num = 45.352; // declaration of variable with type double and initialize it with 45.352.In this section, we will learn in easy steps, how to set up the Arduino IDE on our computer and prepare the board to receive the program via USB cable.

Step 1

First you must have your Arduino board (you can choose your favorite board) and USB cable. In case you use Arduino UNO, Arduino Duemilanove, Nano, Arduino Mega2560, or Diecimila, you will need a standard USB cable (A plug to B plug)

Step 2- Download Arduino IDE Software.

You can get different versions of Arduino IDE from the Download page on the Arduino Official website. You must select your software, which is compatible with your operating system. After your file download is complete, unzip the file.


Opening arduino-nightly-windows.zip

You have chosen to open:

arduino-nightly-windows.zip

which is: WinRAR ZIP archive (148 MB)

from: https://downloads.arduino.cc

What should Firefox do with this file?

Open with WinRAR archiver (default)

[✓ Save File]

Do this automatically for files like this from now on.

OK Cancel

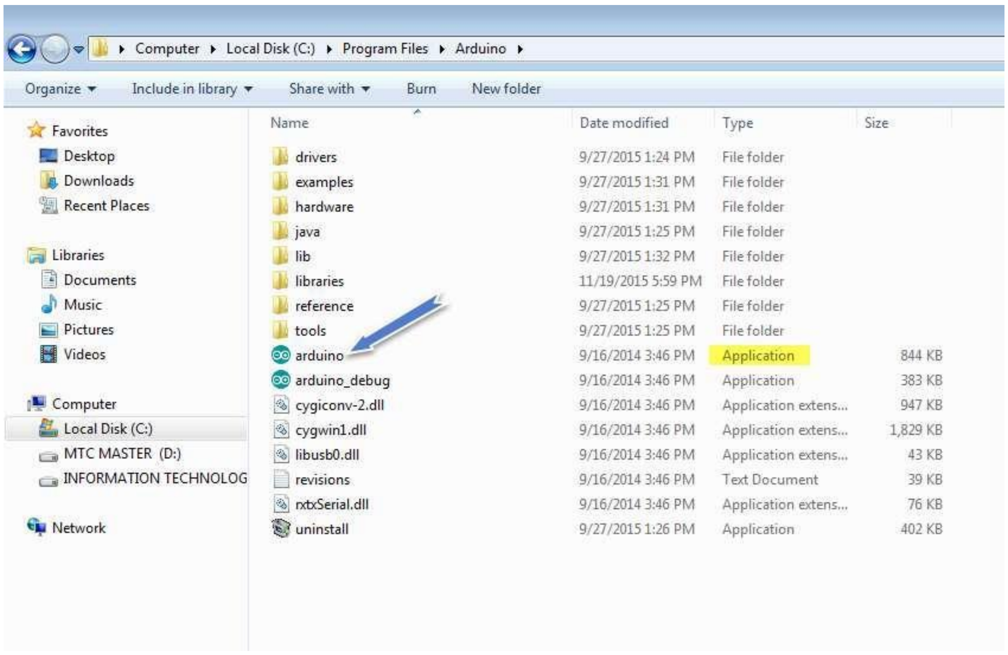
Figure 5.1 : screenshot of opening of Arduino IDE window

Step 3- Power up your board.

The Arduino Uno automatically draw power from either, the USB connection to the computer or an external power supply. If you are using an Arduino Diecimila, you have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port. Connect the Arduino board to your computer using the USB cable. The green power LED should glow.

Step 4- Launch Arduino IDE.

After your Arduino IDE software is downloaded, you need to unzip the folder. Inside the folder, you can find the application icon with an infinity label (application.exe). Double-click the icon to start the IDE.



Name	Date modified	Type	Size
drivers	9/27/2015 1:24 PM	File folder	
examples	9/27/2015 1:31 PM	File folder	
hardware	9/27/2015 1:31 PM	File folder	
java	9/27/2015 1:25 PM	File folder	
lib	9/27/2015 1:32 PM	File folder	
libraries	11/19/2015 5:59 PM	File folder	
reference	9/27/2015 1:25 PM	File folder	
tools	9/27/2015 1:25 PM	File folder	
arduino	9/16/2014 3:46 PM	Application	844 KB
arduino_debug	9/16/2014 3:46 PM	Application	383 KB
cygiconv-2.dll	9/16/2014 3:46 PM	Application extens...	947 KB
cygwin1.dll	9/16/2014 3:46 PM	Application extens...	1,829 KB
libusb0.dll	9/16/2014 3:46 PM	Application extens...	43 KB
revisions	9/16/2014 3:46 PM	Text Document	39 KB
rxtxSerial.dll	9/16/2014 3:46 PM	Application extens...	76 KB
uninstall	9/27/2015 1:26 PM	Application	402 KB

Figure 5.2 : screen-shot of showing arduino IDE icon

Step 5- Open your first project.

Once the software starts, you have two options:

- Create a new project.
- Open an existing project example.

To create a new project, select File -->New. To open

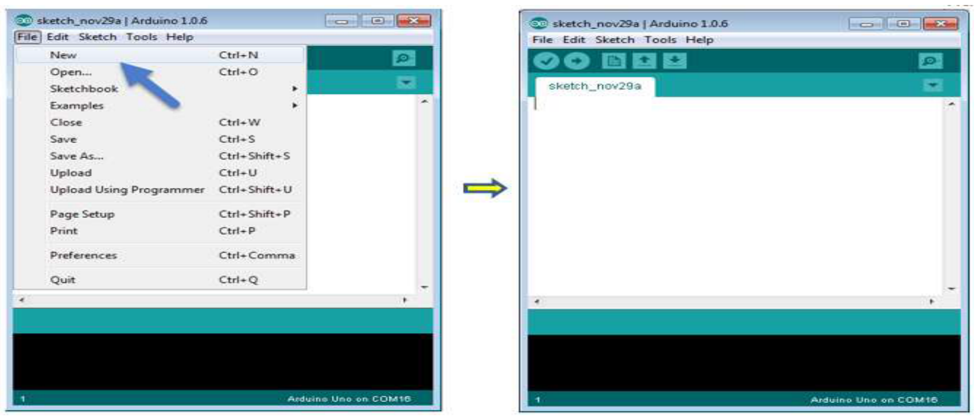
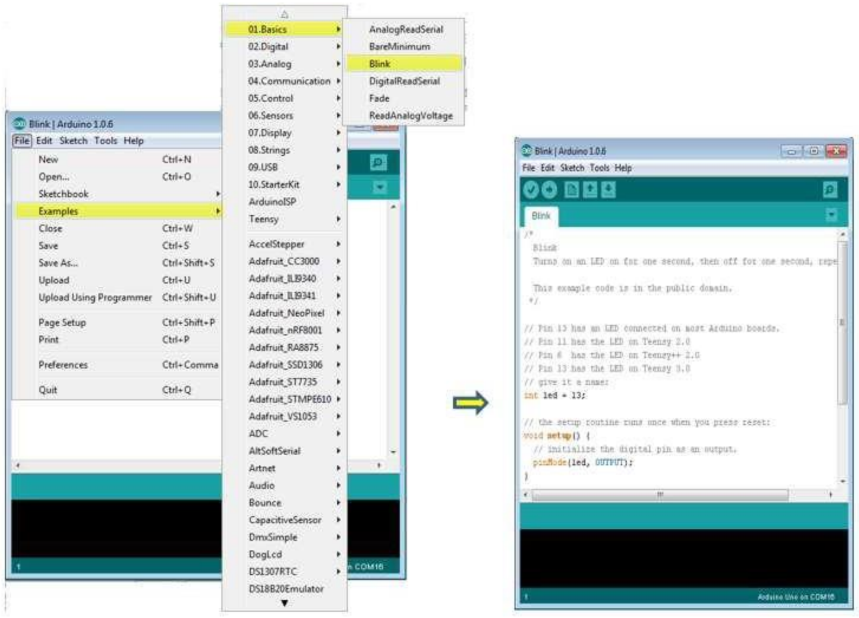


Figure 5.3: creating of New Project

To open an existing project example, select File -> Example -> Basics -> Blink.



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repe

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// Pin 11 has the LED on Teensy 2.0
// Pin 6  has the LED on Teensy++ 2.0
// Pin 13 has the LED on Teensy 3.0
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}
```

Figure 5.4 : Example of opening existing project

Here, we are selecting just one of the examples with the name Blink. It turns the LED on and off with some time delay. You can select any other example from the list.

Step 6- Select your Arduino board.

To avoid any error while uploading your program to the board, you must select the correct Arduino board name, which matches with the board connected to your computer.

Go to Tools -> Board and select your board

Here, we have selected Arduino Uno board according to our tutorial, but you must select the name matching the board that you are using

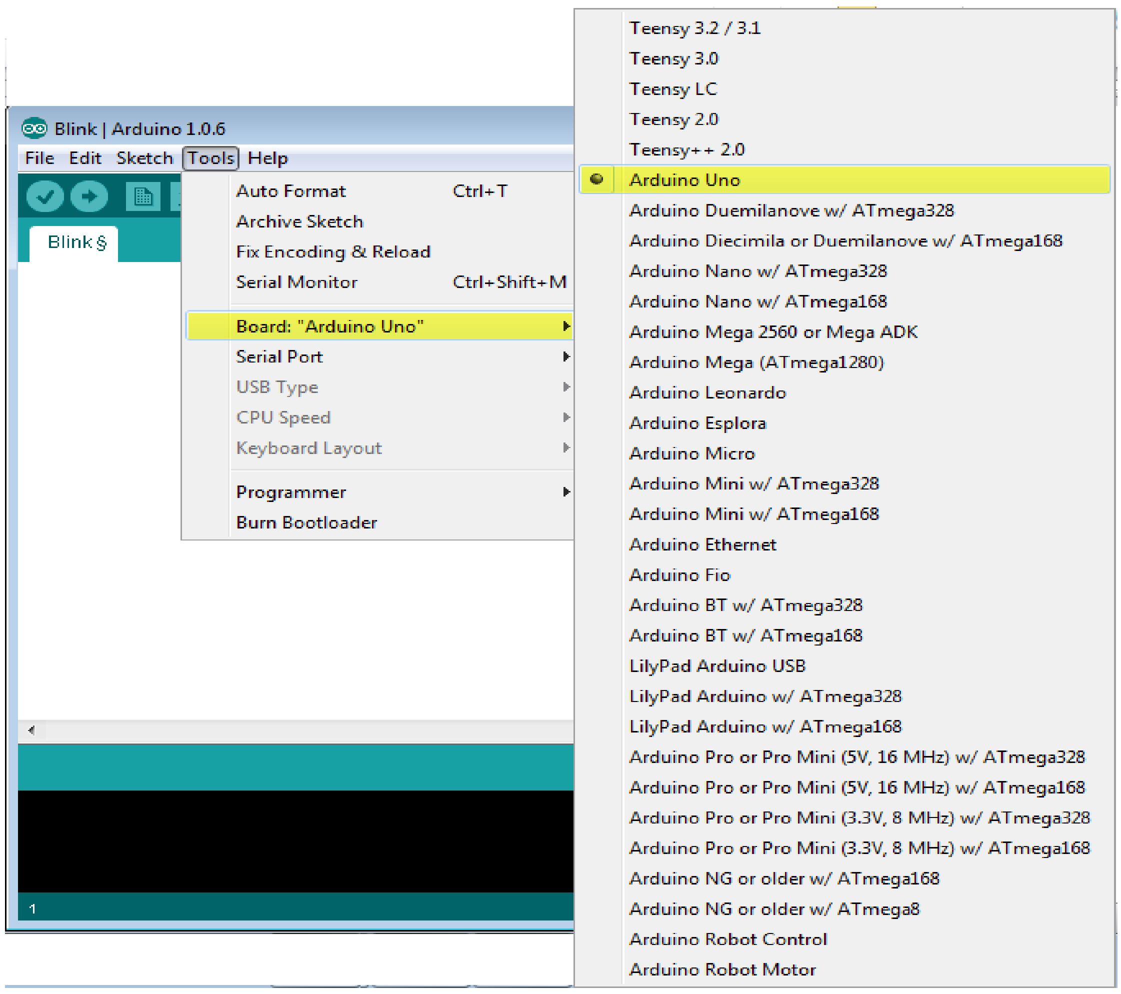

Blink | Arduino 1.0.6
File
Edit
Sketch
Tools
Help
Blink §
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board: "Arduino Uno"
Serial Port
USB Type
CPU Speed
Keyboard Layout
Programmer
Burn Bootloader
Teensy 3.2 / 3.1
Teensy 3.0
Teensy LC
Teensy 2.0
Teensy++ 2.0
● Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega 2560 or Mega ADK
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Esplora
Arduino Micro
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino USB
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
Arduino Robot Control
Arduino Robot Motor
1

Figure 5.5 : Selecting of Arduino board

Step 7- Select your serial port.

Select the serial device of the Arduino board. Go to Tools ->Serial Port menu. This is likely to be COM3 or higher. To find out, you can disconnect your Arduino board and re-open the menu, the entry that disappears should be of the Arduino board. Reconnect the board and select that serial port.

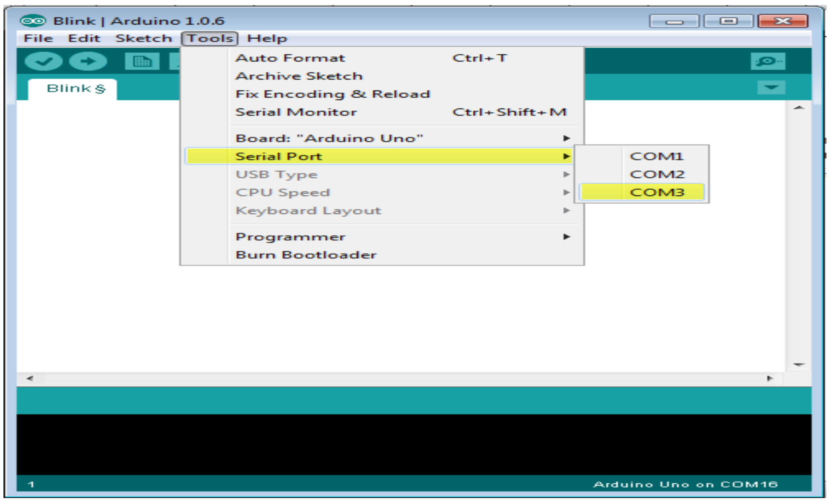


Figure 5.6 : Selecting of serial port

Step 8- Upload the program to your board**.**.

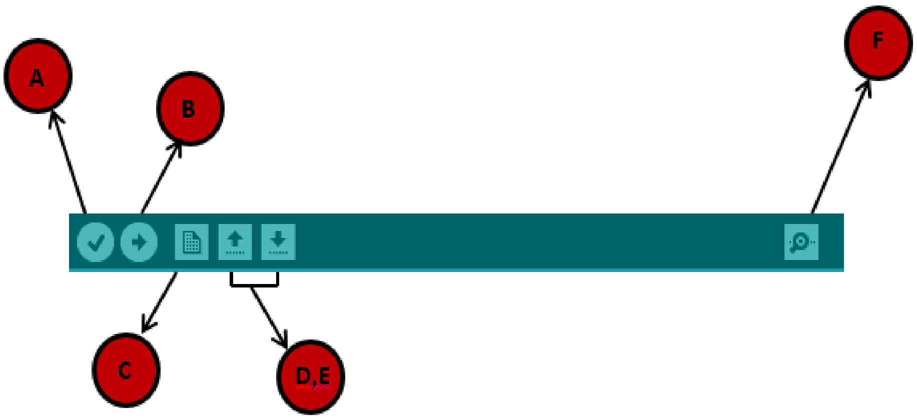


Figure 5.7 : Showing of icon labels

A- Used to check if there is any compilation error.

B- Used to upload a program to the Arduino board.

C- Shortcut used to create a new sketch.

D- Used to directly open one of the example sketch.

E- Used to save your sketch.

F- Serial monitor used to receive serial data from the board and send the serial data to the board.

Now, simply click the "Upload" button in the environment. Wait a few seconds; you will see the RX and TX LED’s on the board, flashing. If the upload is successful, the message "Done uploading" will appear in the status bar.

Arduino programming structure

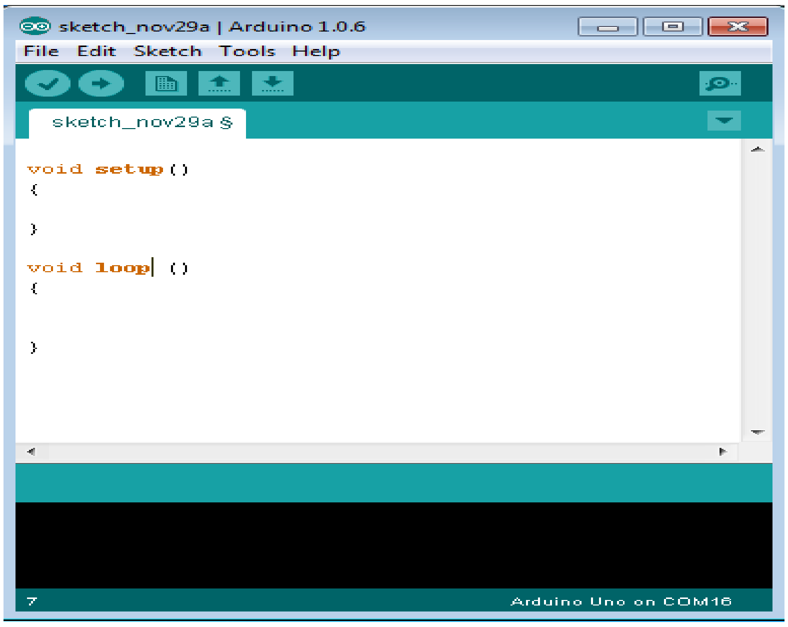
In this chapter, we will study in depth, the Arduino program structure and we will learn more new terminologies used in the Arduino world. The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

Structure

Arduino programs can be divided in three main parts: Structure, Values and Functions. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the Structure. Software structure consist of two main functions:

- Setup( ) function
- Loop( ) function



```
void setup()
{

}

void loop ()
{


}
```

Figure 5.8 : Showing the structure of the Arduino programming

```
Void setup ( )

{

}
```

PURPOSE

The setup( ) function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

INPUT

OUTPUT

RETURN

```
Void Loop ( )

{

}
```

PURPOSE

After creating a setup( ) function, which initializes and sets the initial values, the loop ( ) function does precisely what its name suggests, and loops secutively, allowing your program to change and respond. Use it to actively control the Arduino board.

INPUT

OUTPUT

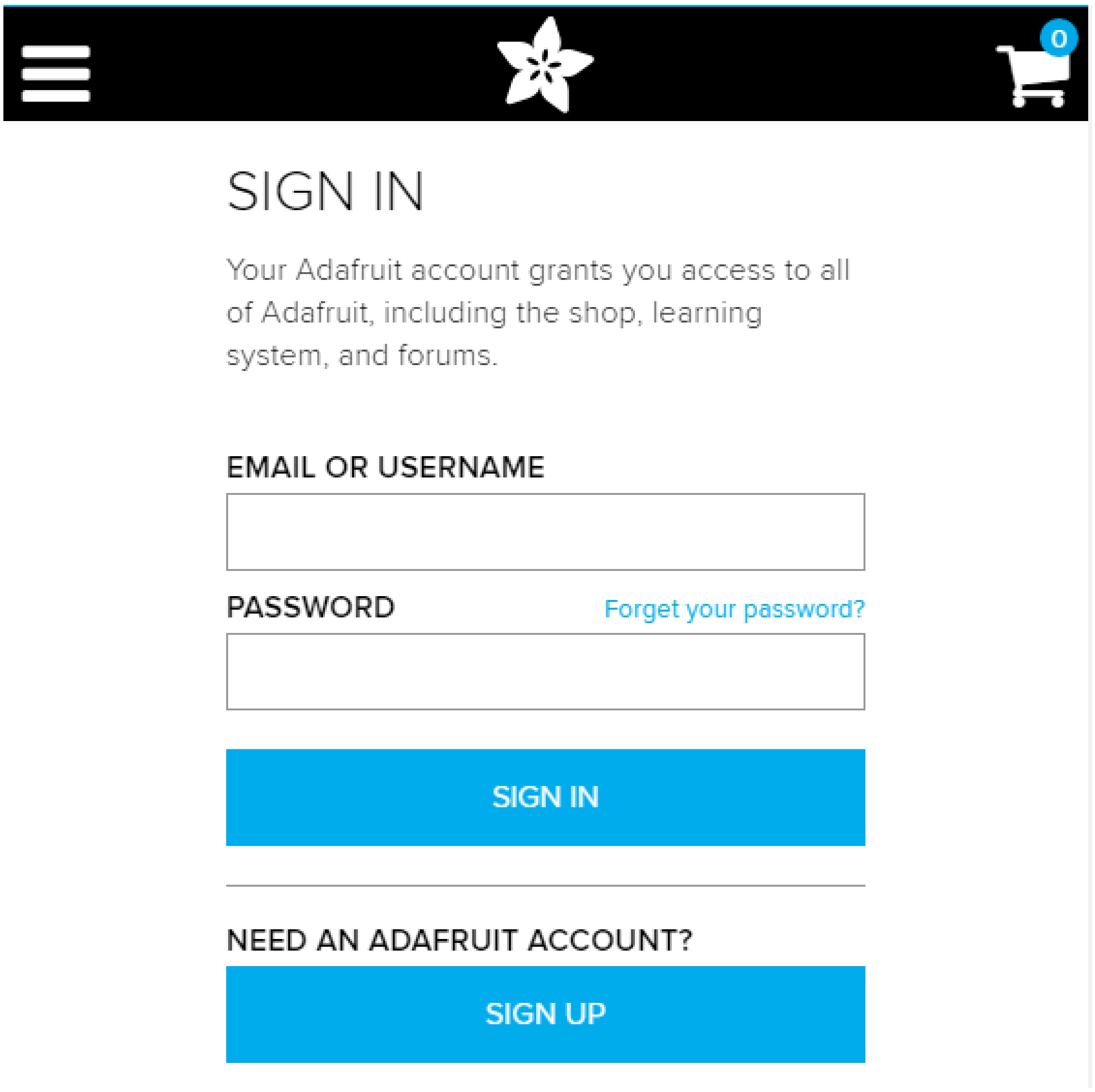
RETURN

5.2 INTRODUCTION TO ADAFRUIT

Adafruit.io is a ***cloud service*** - that just means we run it for you and you don't have to manage it. You can connect to it over the Internet. It's meant primarily for storing and then retrieving data.

5.2.1 Steps for creating the Adafruit Account

First, created account at www.Adafruit.io



SIGN IN

Your Adafruit account grants you access to all of Adafruit, including the shop, learning system, and forums.

EMAIL OR USERNAME

PASSWORD Forget your password?

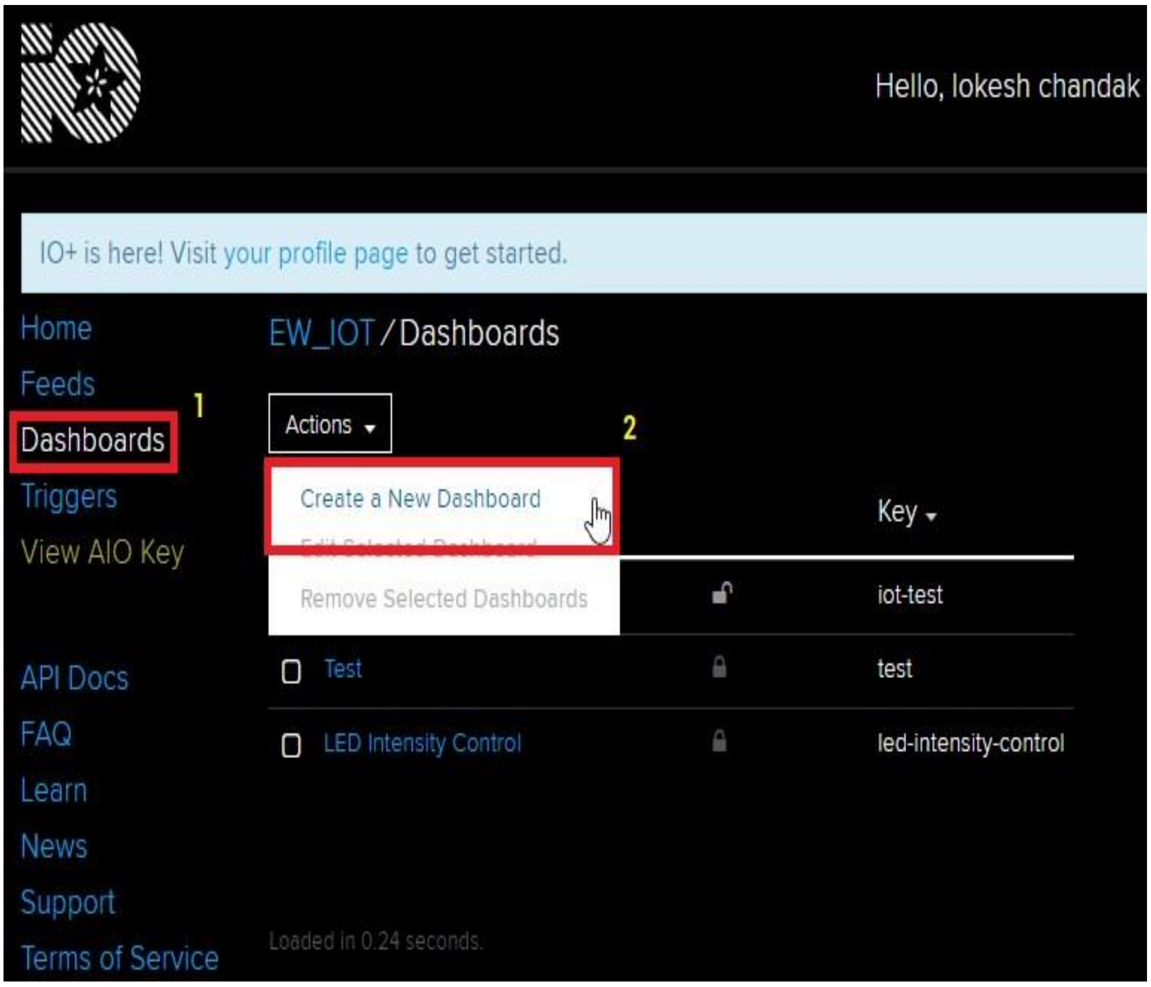
SIGN IN

NEED AN ADAFRUIT ACCOUNT?

SIGN UP

Figure 5.9 : Create the Adafruit Account

Now, create dashboard at Adafruit. This dashboard is a user interface to control things remotely.



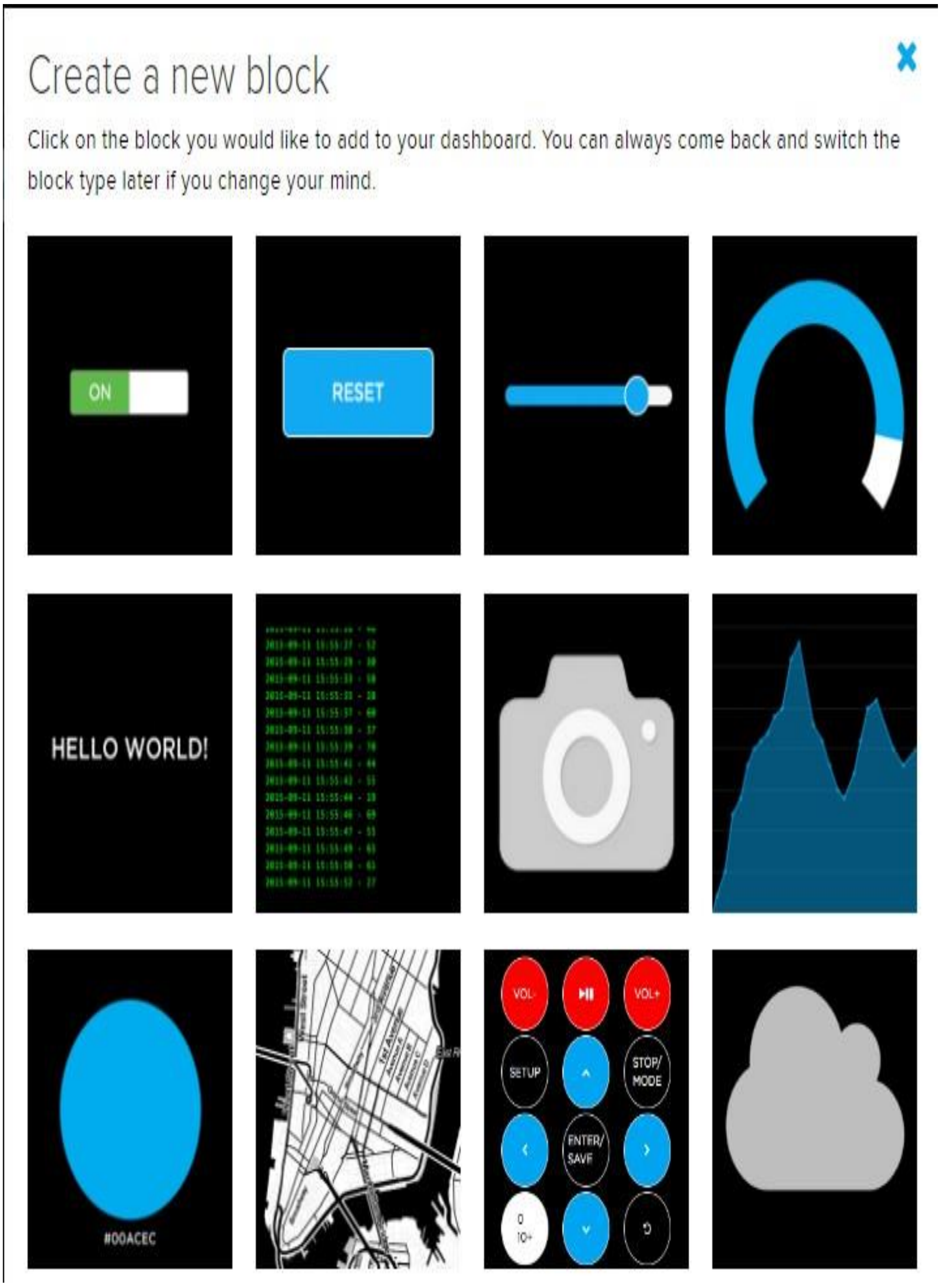
IO+ is here! Visit your profile page to get started.

Loaded in 0.24 seconds.

Figure 5.10 : Creating of a new dashboard

After following above steps, provide name to the dashboard and save it.

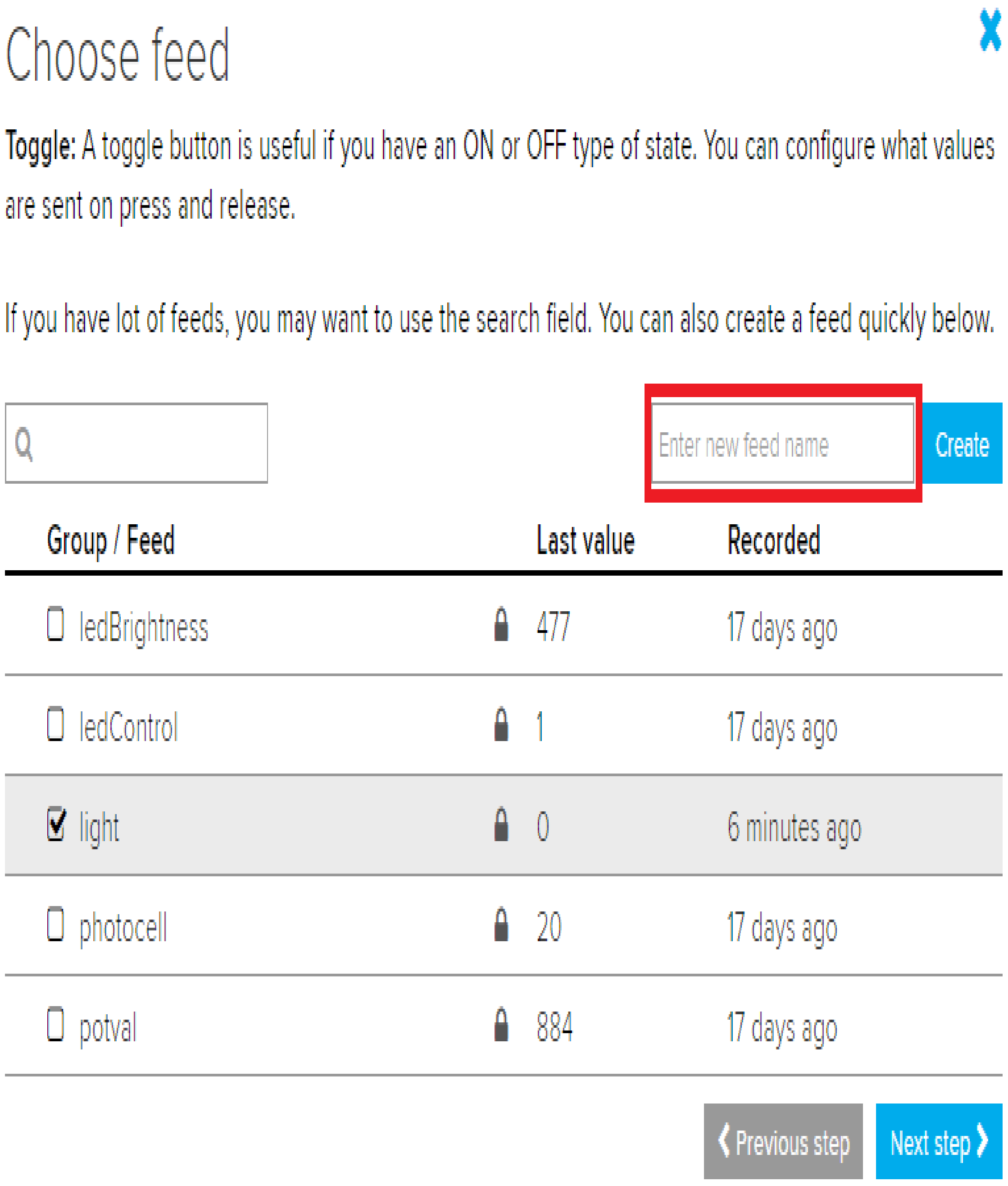
Now, create feed (user interface) to control light On-Off. To create it, just click on '+' symbol and select toggle feed


Create a new block

Click on the block you would like to add to your dashboard. You can always come back and switch the block type later if you change your mind.

Figure 5.11 : Creating of new feeds

After selecting toggle feed, pop-up window appears as shown below.



Choose feed

Toggle: A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

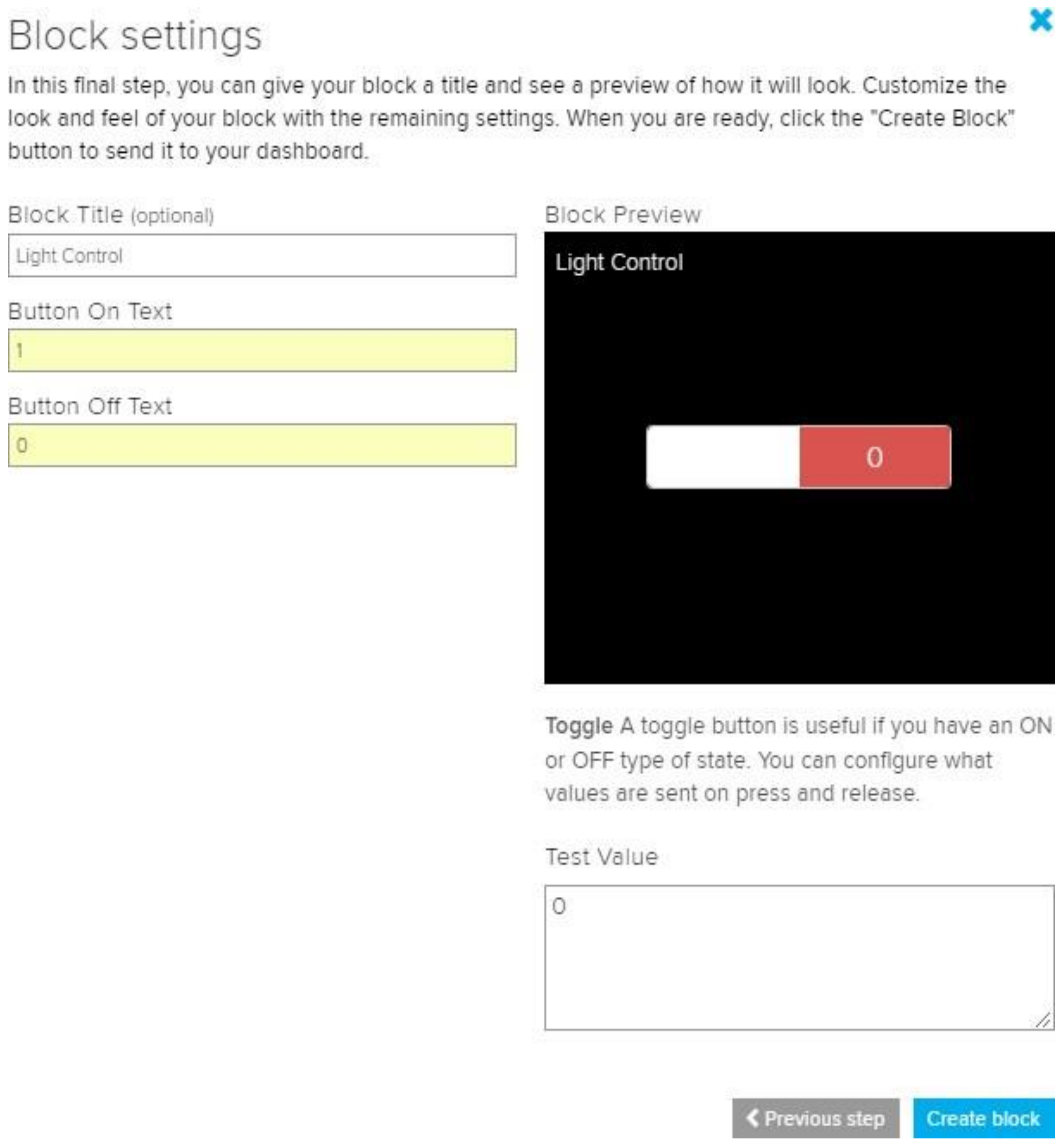
If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

Group / Feed	Last value	Recorded
☐ ledBrightness	477	17 days ago
☐ ledControl	1	17 days ago
☑ light	0	6 minutes ago
☐ photocell	20	17 days ago
☐ potval	884	17 days ago

Figure 5.12: Choosing of feeds

Enter name of our feed (shown in red box) and create it. After creation, select the created feed (here mine is **light**) and then click on **Next step.**

In the next step configure the feed which is shown below,



Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title (optional): Light Control

Button On Text: 1

Button Off Text: 0

Block Preview

Light Control

0

Toggle A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

Test Value: 0

Figure 5.13 : Configuring of feeds

Here, used **0**(OFF) and **1**(ON) text for button and then click on create. This will create toggle button on your dashboard which can be used to control things remotely.

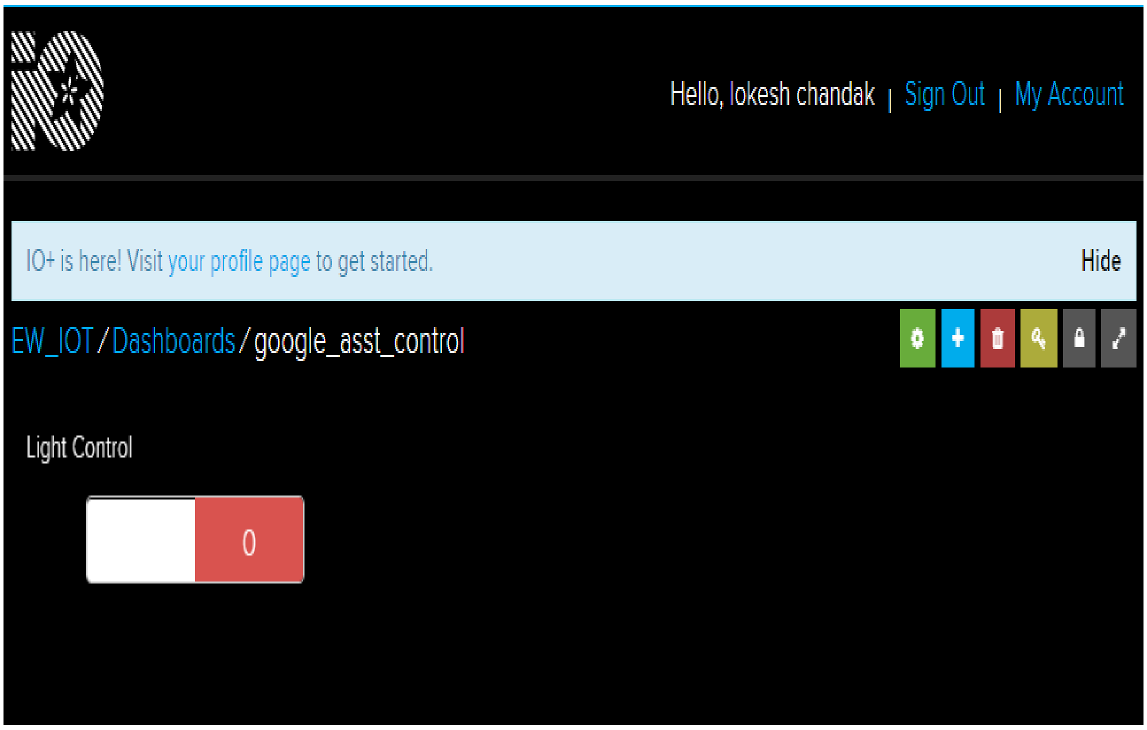

Hello, lokesh chandak | Sign Out | My Account
IO+ is here! Visit your profile page to get started.
Hide
EW_IOT / Dashboards / google_asst_control
Light Control
0

Figure 5.14 : Creating of Toggle buttons

Now, the dashboard is ready for IoT application like home automation.

5.3 INTRODUCTION TO IFTTT (If This Then That)

If This Then That, also known as IFTTT is a free web-based service to create chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as G-mail, Facebook, Telegram, Instagram, or Pinterest.

For example, an applet may send an e-mail message if the user tweets using a hashtag or copy a photo on Facebook to a user's archive if someone tags a user in a photo.

Here, It is used IFTTT to use google assistant service and Adafruit service in chain. So, when the use of google assistant to control light of my home by saying Ok google, turn the light ON or OFF. Then IFTTT interpret the message and can send it to Adafruit's dashboard as a understandable command to the created feed.

5.3.1 Configure IFTTT

First step is creating account on IFTTT.

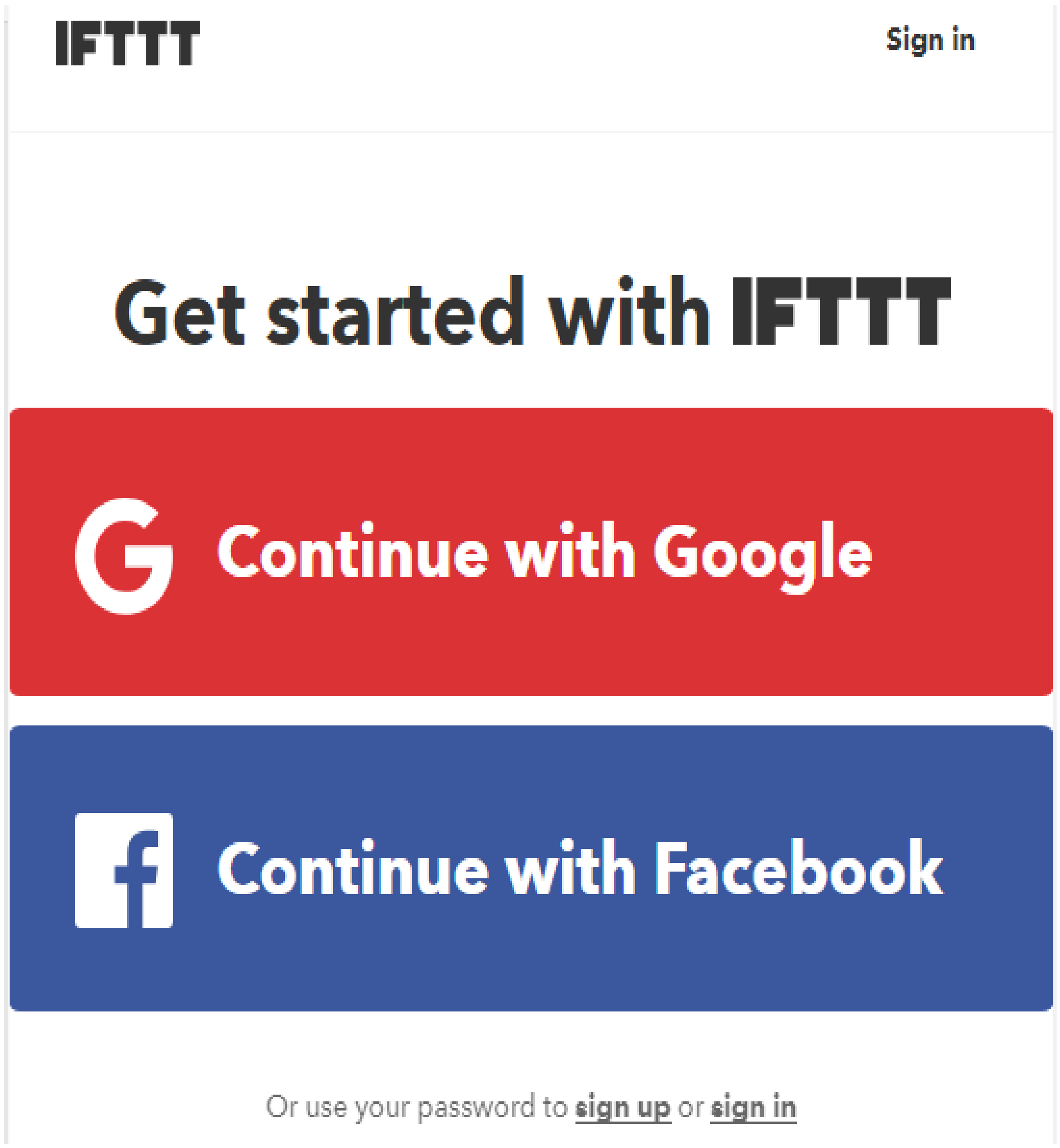


Figure 5.15 : Creating the IFTTT account

Note: Create account on IFTTT by using same e-mail id which you have used for Adafruit.After account creation, click on **My Applets** and then select **New Applet** shown below,

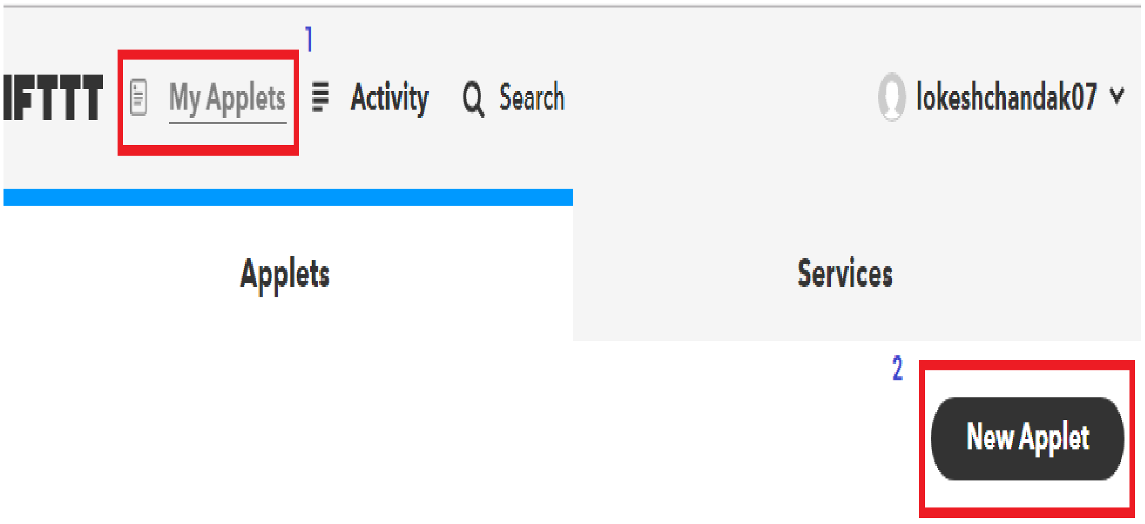


Figure 5.16 : Creating of New Applet

After selecting a new applet, we get a new page in which we should click on to **This** as shown in below image.

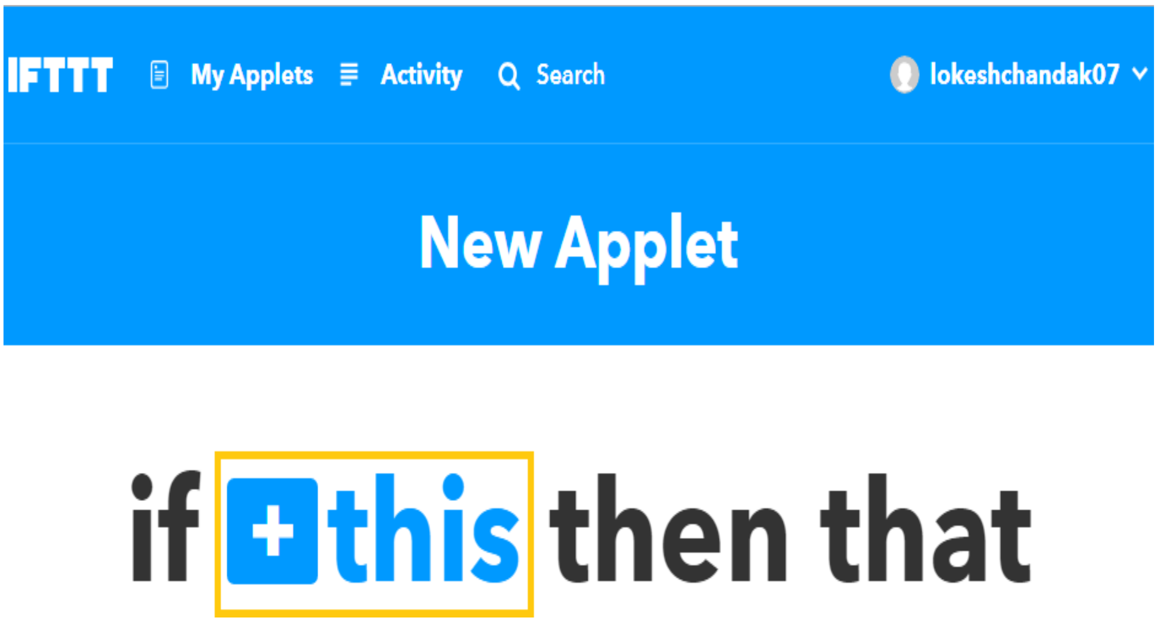


Figure 5.17 : Showing of New Applet icon

Then search for **Google Assistant** and select it.

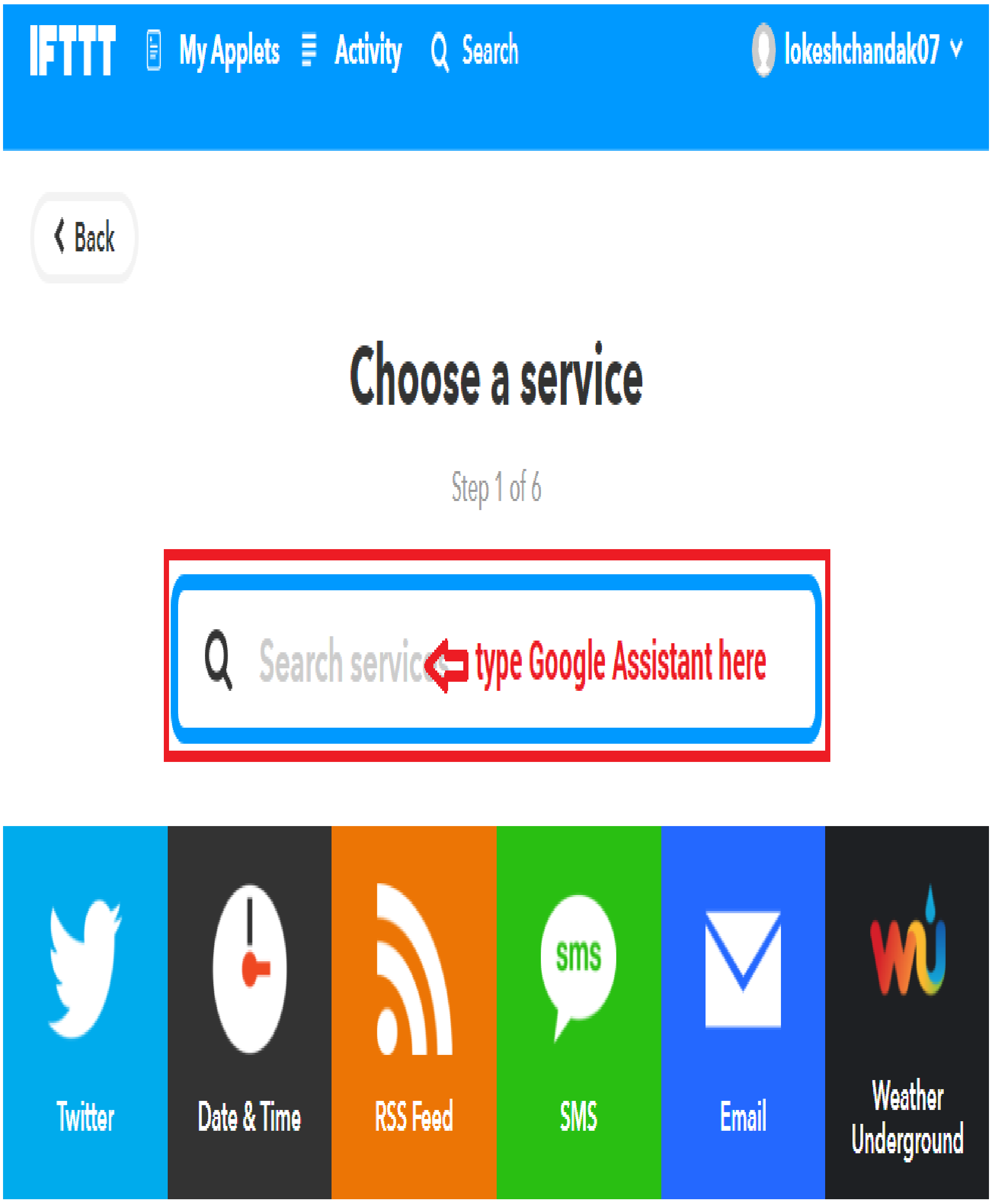
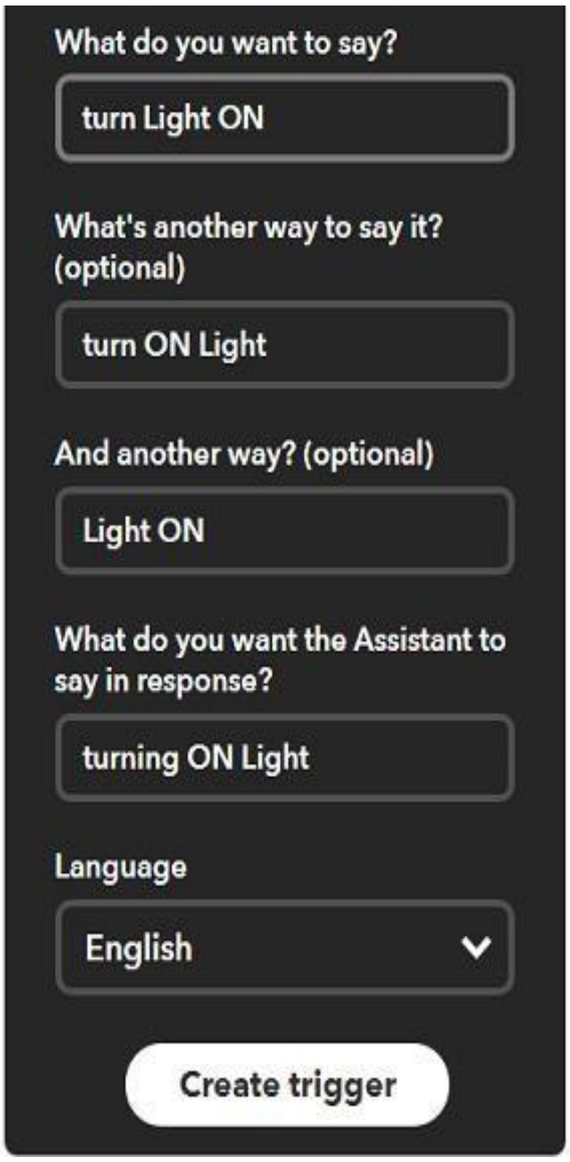


Figure 5.18 : Search for the Google Assistant icon

Now, enter voice phrases which we will use as a command for google assistant.



What do you want to say?

turn Light ON

What's another way to say it? (optional)

turn ON Light

And another way? (optional)

Light ON

What do you want the Assistant to say in response?

turning ON Light

Language

English

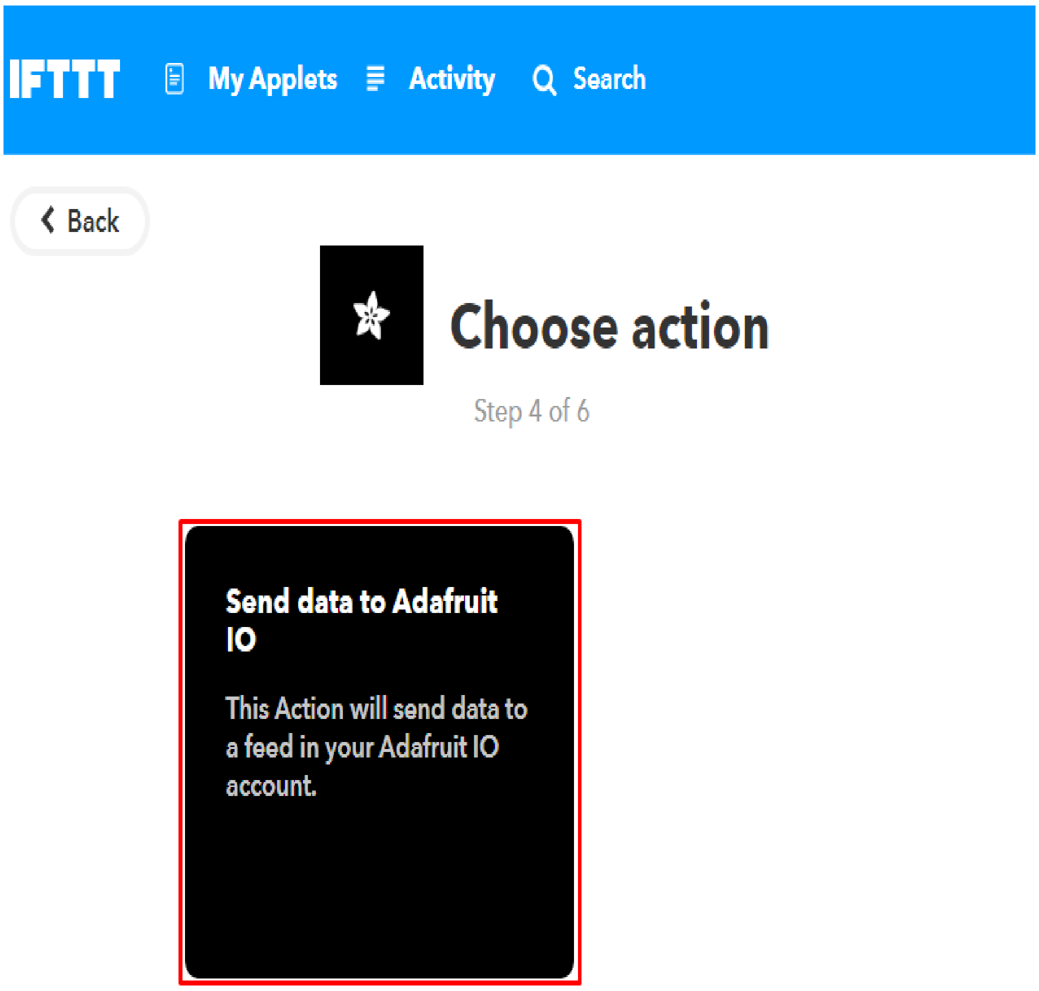
Figure 5.19 : Entering of voice phrases commands

We can enter any phrase as per our application. As you can see, the phrases entered in the above fields is for making **Light ON.** For making **Light OFF**, we have to create another applet with different phrases.

Now, we get another page on which we have to click on **that** option which is used to connect Google Assistant with Adafruit.

Then search for **Adafruit** and select it.

After selecting Adafruit, choose action as shown below,



Choose action

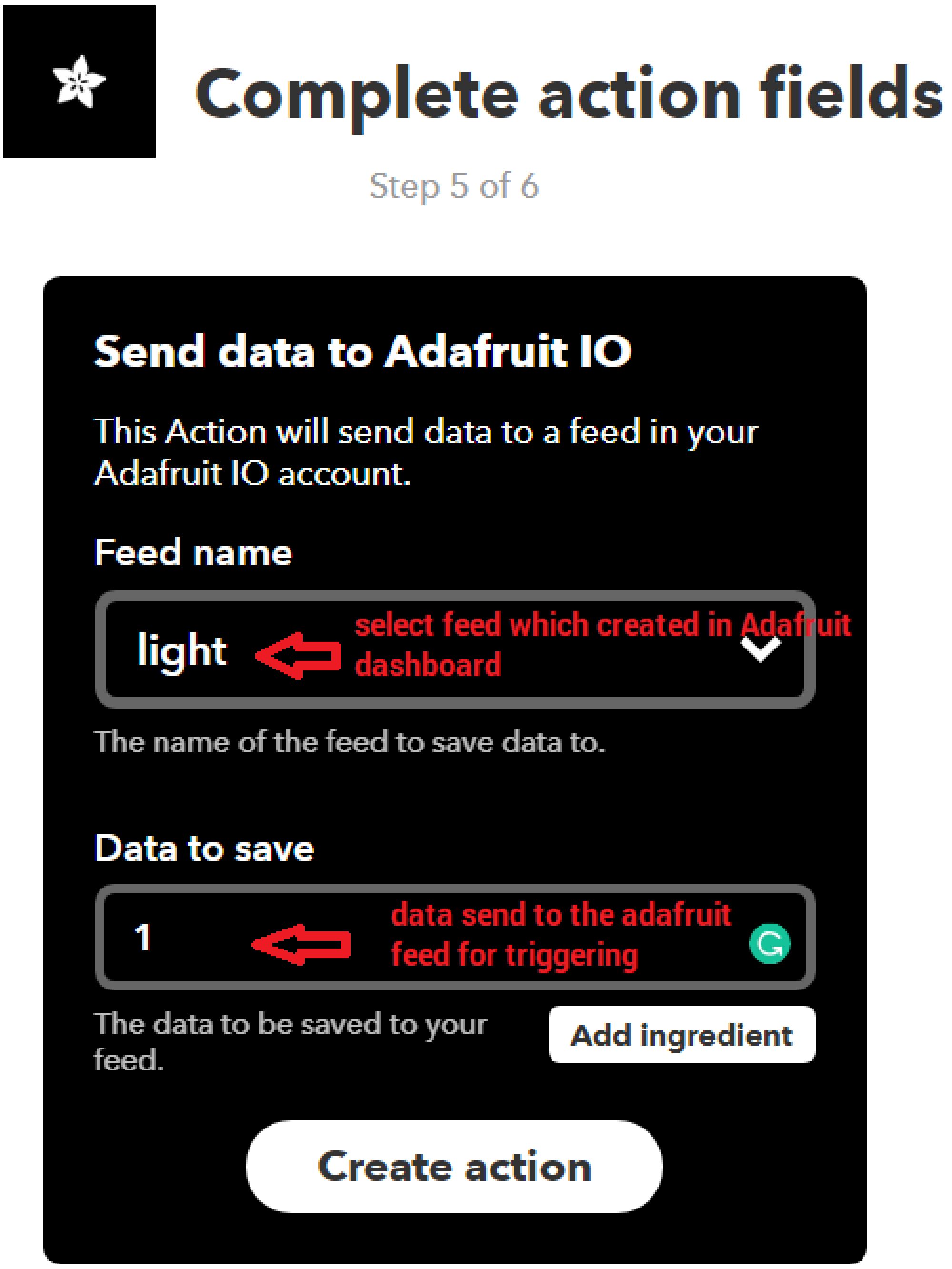
Step 4 of 6

Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

Figure 5.20 : search for the Adafruit

Now enter what data we need to send to which feed of Adafruit dashboard.



Complete action fields

Step 5 of 6

Send data to Adafruit IO

This Action will send data to a feed in your Adafruit IO account.

Feed name

light

select feed which created in Adafruit dashboard

The name of the feed to save data to.

Data to save

1

data send to the adafruit feed for triggering

The data to be saved to your feed.

Figure 5.21 : Enter data to Adafruit dashboard

Click on **Create Action.**

So, when I use Google Assistant on my mobile and give voice command as "Ok Google, Turn LED ON", applet created in IFTTT receive this command and will send data '1' to the Adafruit feed. This will trigger the event on Adafruit dashboard which is continuously monitored by the microcontroller (here NodeMCU). This microcontroller will take action as per the data change on the Adafruit dashboard.

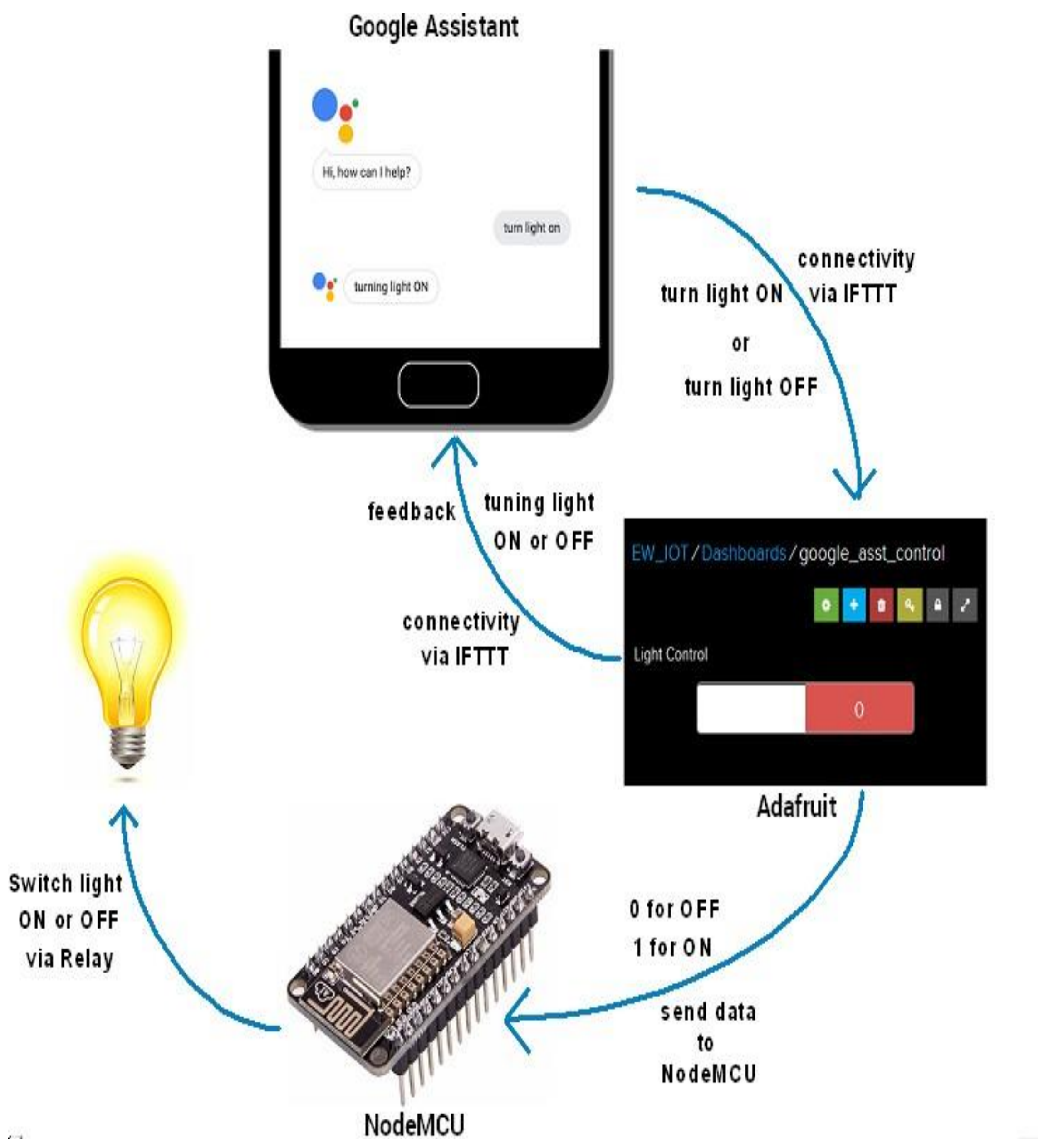


Figure 5.22 : Working mechanism of control of devices

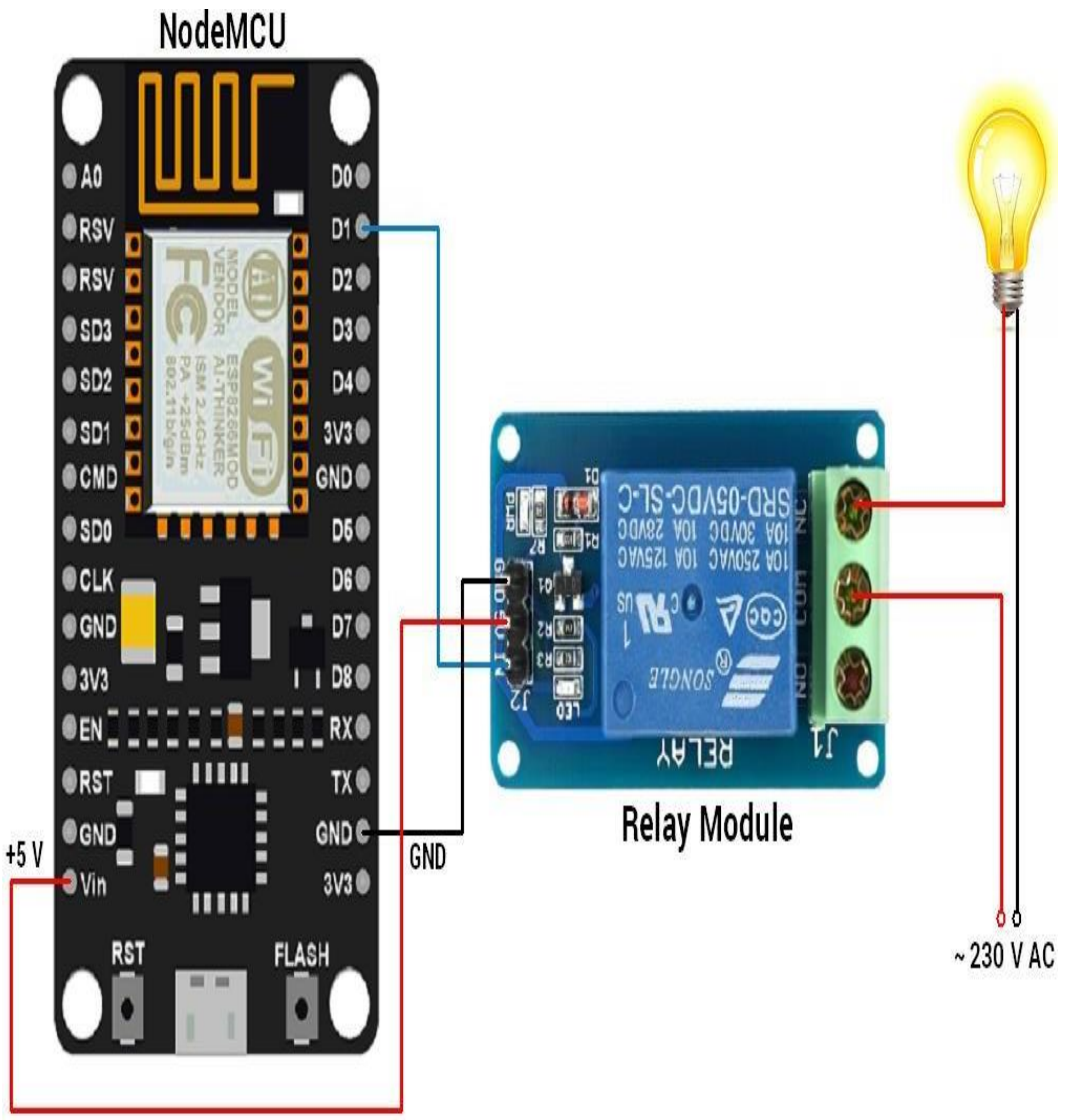


Figure 5.23 : Interfacing Diagram Library

Here, used the Adafruit MQTT library for receiving data from the Adafruit server. To install this library, select option **Sketch -> Include Library -> Manage Libraries.**

In that library, search for Adafruit MQTT and installed it.

Control Home's Light using Google Assistant and NodeMCU

we build an IoT based home automation application in which we control the 100 W bulb at remotely using AI based Google Assistant.

Here, used NodeMCU to read data from Adafruit server and act accordingly. 100 W bulb connected to NodeMCU via relay for controlling it voice command using google assistant.

5.4 GOOGLE ASSISTANT

Google Assistant is an artificial intelligence powered virtual assistant developed by Google which is basically available on mobile and smart home devices. Google Assistant can engage in two-way conversations living behind company's foregoing virtual assistant. Moreover it is available in different languages providing comfort to customer.

CHAPTER-6

RESULT AND ANALYSIS

6.1 RESULT

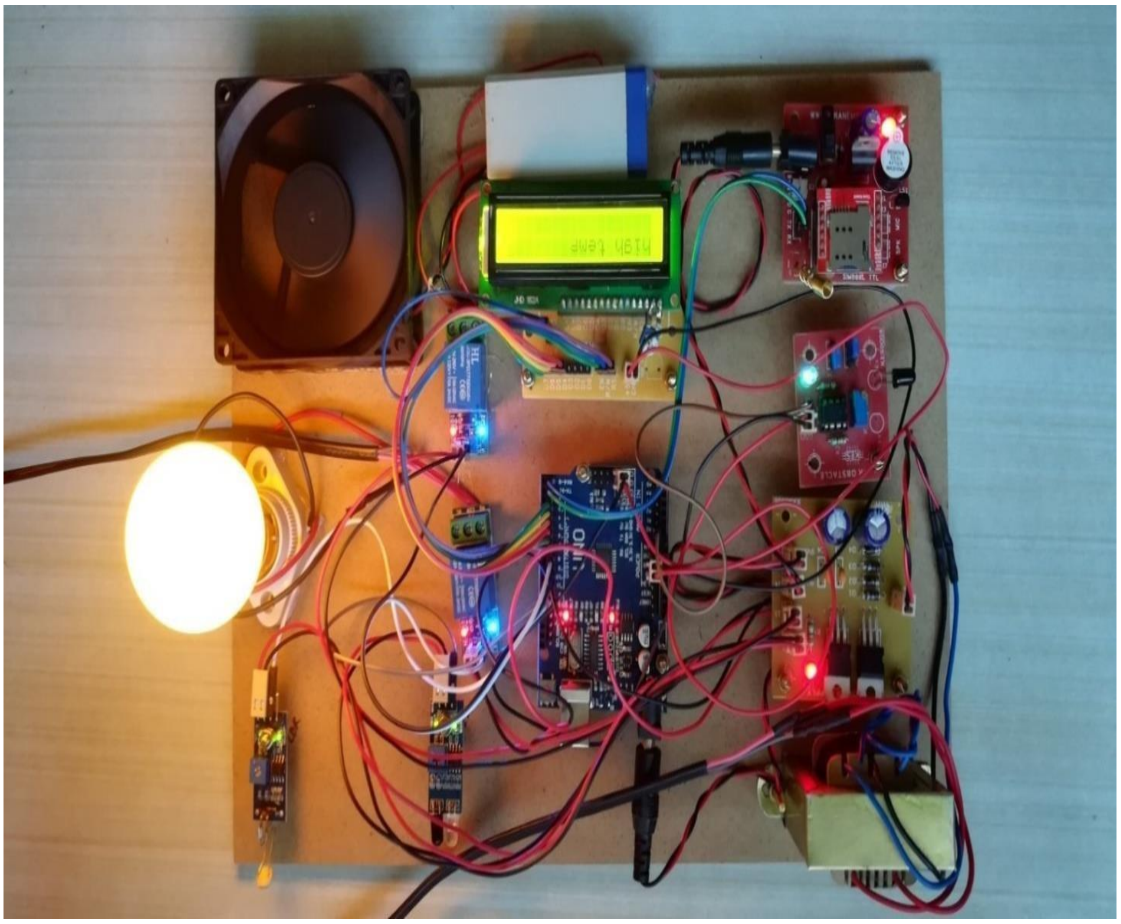


Figure 6.1: Wi-Fi based wireless advanced home automation system

6.2 ANALYSIS

Case 1

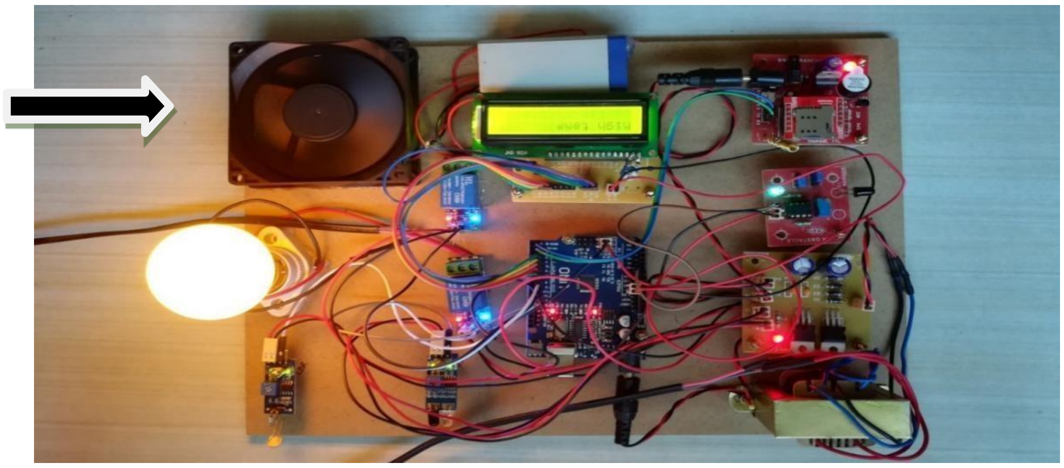


Figure 6.2: Switching -ON the fan

In case 1 when we give voice command as switch on fan through the google assistant then it will update the switch on fan in to the server through internet and simultaneously it switches on the fan.

Case 2

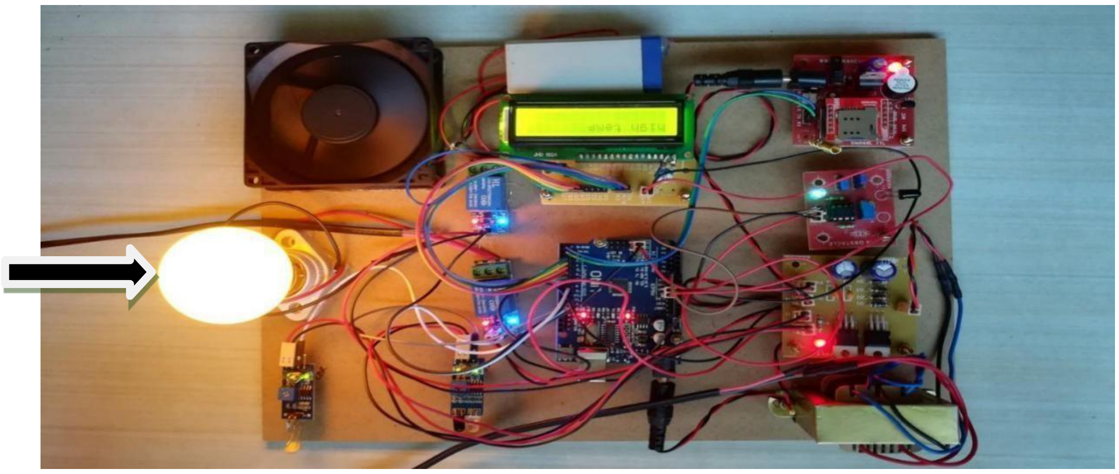


Figure 6.3: Switching -ON the Light

In case 2 when we give voice command as switch on light through the google assistant then it will update the switch on light in to the server through internet and simultaneously it switches on the light.

Case 3

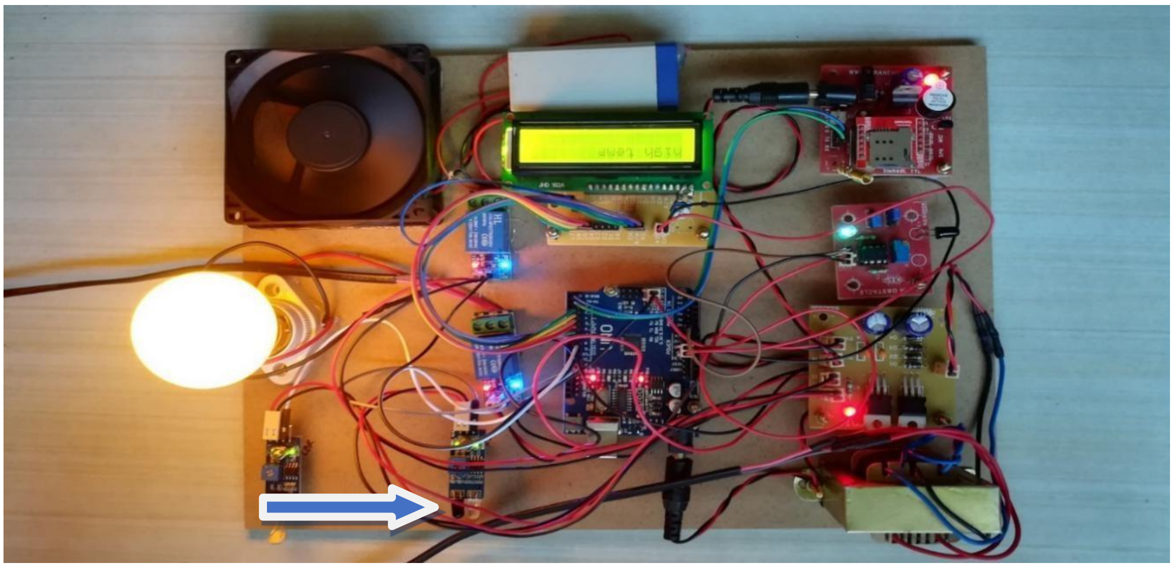


Figure 6.4: Detection of obstracle by IR Sensor

In case 3 the IR sensor will sense whether if any person is entered in to the room. If any person got detected, then it shows object detected and upload as person detected in to the server.

CHAPTER-7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The future of IOT is virtually unlimited due to advances in technology and consumers' desire to integrate devices such as smart phones with household machines. Wi-Fi has made it possible to connect people and machines on land, in the air and at sea. It is critical that both companies and governments keep in ethics in mind as we approach the fourth Industrial Revolution. With so much data traveling from device to device, security in technology will be required to grow just as fast as connectivity in order to keep up with demands. Governments will undoubtable face tough decisions as to how far the private sector is allowed to go in terms of robotics and information sharing. The possibilities are exciting, productivity will increase and amazing things will come by connecting the world.

By keeping the embedded devices in the environment for monitoring enables self-protection (i.e., smart environment) to the environment. To implement this need to deploy the sensor devices in the environment for collecting the data and analysis. By deploying sensor devices in the environment, we can bring the environment into real life i.e. it can interact with other objects through the network. Then the collected data and analysis results will be available to the end user through the Wi-Fi. The smart way to monitor environment and an efficient, low cost embedded system.

7.2 FUTURE SCOPE

Anything that touches consumer industry became a buzz word. So now Internet of Things is one among. What it is - the end goals is to bring all things we use in day to daylife over network and can be accessed across the world over internet. That means every objects/gadgets we use in a day to day life will have identify over network and its information can be consumes via Laptop, Tablet and mobile and including wearable like smart watches. Why we should do it - the first and foremost thing is automation. In a typical day, we all have a 24 hours - 1/3 of time goes in bed, 1/3 of time goes in office/school and 1/3 third of time we have to spend for ourselves. How effectively are we spending this time will have a ripple effect on throughout the life?

Smart Home-you could automate stuff in home to make faster decision, communicate instantly, monitor the stuff which is most important - you can be more effective automating non-value added activities and spend more time where you want to focus (this is more of my personal productive advice when it comes to automating things). Home automation has a great potential for consumers market. Start-ups and Big companies joining together to take a leap forward

Smart Farm-Country like India, we lose lot of money due to lack predictive systems in agriculture fields (excluding natural distastes). In today's environment we also lack the employees' engagement for agriculture fields. IOT has a great potential for smart farming - no matter where you located but you can still control things on field

Smart Transportation-Already companies like Uber, OLA implemented a great connected platform, even IOT can be extended to our own transportation systems like cars, motorcycle and bikes etc. - every use case will have a different stories. Think about you want to travel in a bus from your home town and your mobile phone alerts you the exactly time you should get out of your home.

Smart City-I don't want to write about it this topic and there are enough articles in place to learn more about. Keeping city connected will help in many ways and make people more productive.

Smart Industry - It applies to all industry that exists today, you can think of a new use-case every day to jump in Internet of Things world. It's quite common for every industry. I've also answered different questions related to IOT and you can refer them also follow IOT Geeks to learn more about Internet of Things. You will get some perspective for sure.

Adding of more sensors to monitor other environmental parameters such as soil PH sensor,CO2 and oxygen sensor while allowing the replacing of current sensors if a wider range of measurements is desired. And also Integration of additional monitoring devices such as a WI-FI camera to monitor growth of agricultural product. And also the data can be uploaded to web server continuously.

REFERENCES

[1] Internet of Things : A Hands On Approach Paperback 2015 by Arsheet Bahga (Author), Vijay Madisetti (Author).

[2] IOT (Internet of Things) Programming:A Simple and Fast Way of Learning IOT kindle Edition By David Etter (Author).

[3] Peggy Laramie, "Instruction Level Power Analysis and Low Power Design Methodology of a Core Processor", M.S. Thesis , UC Berkeley, 1998.

[4] P.B.Chu , KSJ Pister – "optical communication using micro corner cube reflectors" 10th IEEE Int'l Workshop on Micro Electro Mechanical Systems.

[5] J. Kahn , R.H.Katz, KSJ Pister – "Mobile Networking for Smart Dust".

[6] Rowinski, D. (2013, November 13). Connected Air: Smart Dust Is The Future Of The Quantified World, Retrieved March 14, 2018.

[7] Sutter, J. D. (2010, May 3). Smart dust' aims to monitor everything. Retrieved March 14, 2018.

[8] https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html

[9] https://www.elprocus.com/bridge-rectifier-circuit-theory-with-working-operation

[10] https://www.ijarcce.com/upload/2016/september-16/IJARCCE%2066.pdf

[11] http://www-ee.stanford.edu/~jmk/pubs/jcn.00.pdf.

[12] http://www.uhisrc.com/FTB/Smart%20Dust/Smart%20Dust.pdf.

[13] https://readwrite.com/2013/11/14/what-is-smartdust-what-is-smartdust-used-for/

[14] http://www.cnn.com/2010/TECH/05/03/smart.dust.sensors/index.html