TABLES OF FIGURES

ABSTRACT

In the present world thefts have become more and we are lagging security. Though we are so careful about our valuable things they are getting stolen. As we are living in a world where we cannot trust someone else for the safety of our things, we can design a small project which is affordable.

In this project we are using Node MCU board. Node MCU is an open source IoT device. It includes firm fare which runs on the ESP8266 WiFi SoC from Espressif systems, and hardware which is based on the ESP-01 Module. For this Node MCU we connect our WiFi ssid by using the IoT platforms for output of our project.

In this project we are going to get a notification whenever our valuable things are being touched by other persons. So we can get alert through an email and a buzzer is also kept and it will make a sound to notify if someone touches it. So we can easily get alerted when the theft is going to occur and we can immediately take the actions. In this project we will use a nodemcu, a buzzer, bread board, bread board power supply, smps adapter, touch sensor. This project is very useful as its is of low cost, high quality and accurate results.

CHAPTER-1
PROJECT OVERVIEW

INTRODUCTION:

In this project we are going to make a theft detection project using node mcu and blynk platform. As we all know that, we are in a world where we cannot trust anyone. There are many people who are becoming thieves and stealing things. This has become a major problem now a days. People are very afraid to keep the valuable things in their houses as they can be stolen easily. Though there are number of security system existing they are very costly and all cannot afford that. So here we are going to make a simple project where we can get an email and a message alert when some one touches our things. We are even keeping a buzzer as to threaten the thief and also the people around. This project is very useful and very cheap.

CHAPTER-2

PROJECT REQUIREMENTS AND ITS DESCRIPTION

PROJECT REQUIREMENTS

Software Requirements:

Operating System : WINDOWS 8/10

Software :Arduino IDE

Hardware Requirements:

NodeMCU

Bread board power supply

Bread board

SMPS Adapter

Touch sensor

buzzer

Jumper wires

Platforms:

Blynk platform

ARDUINO IDE SOFTWARE

ARDUINO:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, 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 out the functions of the micro-controller into a more accessible package. The Uno is one of the more popular boards in the Arduino family and a great choice for beginners. We'll talk about what's on it and what it can do later in the tutorial.

ARDUNIO IDE:

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java, c and based on Processing and other open-sourcesoftware. Its an easy coding software. We can control our hardware by writing the code in this. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communication s interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.The Arduino project started in 2003 as a program for students at the in Ivrea, Italy, aiming to provide a low-cost and easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.

MQTT:

MQTT (MQ Telemetry Transport or Message Queue Telemetry Transport) is an ISO standard (ISO) publish-subscribe-based "lightweight" messaging protocol for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker. The broker is responsible for distributing messages to interested clients based on the topic of a message. Andy Stanford-Clark and Arlen Nipper of Cirrus Link authored the first version of the protocol in 1999.

The specification does not specify the meaning of "small code footprint" or the meaning of "limited network bandwidth". Thus, the protocol's availability for use depends on the context. In 2013, IBM submitted MQTT v3.1 to the OASIS specification body with a charter that ensured only minor changes to the specification could be accepted. MQTT-SN is a variation of the main protocol aimed at embedded devices on non-TCP/IP networks, such as ZigBee.

Historically, the "MQ" in "MQTT" came from IBM's MQ Series message queuing product line. However, queuing itself is not required to be supported as a standard feature in all situations. Alternative protocols include the Advanced Message

Queuing Protocol (AMQP), Streaming Text Oriented Messaging Protocol (STOMP) the IETF Constrained Application Protocol, XMPP and Web Application Messaging Protocol (WAMP).

Real-world applications:

MQTT is designed to support wireless networks with varying levels of latency due to occasional bandwidth constraints or unreliable connections. There are several projects that implement MQTT.

Examples:

- Facebook Messenger. Facebook has used aspects of MQTT in Facebook Messenger for online chat. However, it is unclear how much of MQTT is used or for what.
- IECC Scalable, Delta Rail's latest version of their IECC Signaling Control System uses MQTT for communications within the various parts of the system and other components of the signaling system. It provides the underlying communications framework for a system that is compliant with the cenelec standards for safety-critical communications.

- The EVRYTHNG IoT platform uses MQTT as an M2M protocol for millions of connected products .Amazon Web Services announced Amazon IoT based on MQTT in 2015.
- The Open Geospatial Consortium Sensor Things API standard specification has a MQTT extension in the standard as an additional message protocol binding. It was demonstrated in a US Department of Homeland Security IoT Pilot.
- Adafruit launched a free MQTT cloud service for IoT experimenters and learners called Adafruit IO in 2015.Microsoft Azure IoT Hub uses MQTT as its main protocol for telemetry messages.
- XIM, Inc. launched an MQTT client called MQTT Buddy in 2017. It's a MQTT app for Android and iOS, but not F-Droid, users available in English, Russian and Chinese languages.
- Node-RED supports MQTT nodes as of version 0.14, in order to properly configure TLS connections. Open-source software home automation platform Home Assistant is MQTT enabled and offers four options for MQTT brokers.

Pimatic home automation framework for Raspberry Pi and based on Node.js offers MQTT plugin providing full support for MQTT protocol

NODEMCU

NodeMCU is an open source IoT platform.]It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware] which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the dev kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

History:

NodeMCU was created shortly after the ESP8266 came out. On December 30, 2013, Espressif Systems *b*egan production of the ESP8266. The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core widely used in IoT applications (see related projects). NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the gerber file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported MQTT client library from Contiki to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another

important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to NodeMCU .

Developer	**ESP8266** **Open source community**
type	Single board microcontroller
Operating system	XTOS
CPU	ESP8266(LX106)
memory	128kBytes
storage	4MBytes
power	USB

website	www.nodemcu.con

Pins of NodeMCU:

NodeMCU provides access to the GPIO (General Purpose Input/Output) and for developing purposes below pin mapping table should be referenced.

IO index	ESP8266 pin	IO index	ESP8266 pin
0	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9

5	GPIO14	12	GPIO10
6	GPIO12		

ESP8266 Arduino Core

As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU and used in the Arduino Due, they needed to modify the Arduino IDE so that it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled down to these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file down to the target MCU's machine language. Some creative ESP8266 enthusiasts have developed an Arduino core for the ESP8266 WiFi SoC that is available at the GitHub ESP8266 Core webpage. This is what is popularly called the "ESP8266 Core for the Arduino IDE" and it has become one of the leading software development platforms for the various ESP8266 based modules and development boards, including NodeMCUs. For more information on all things ESP8266, check out the ESP8266 Community Forum on GitHub.

The Button

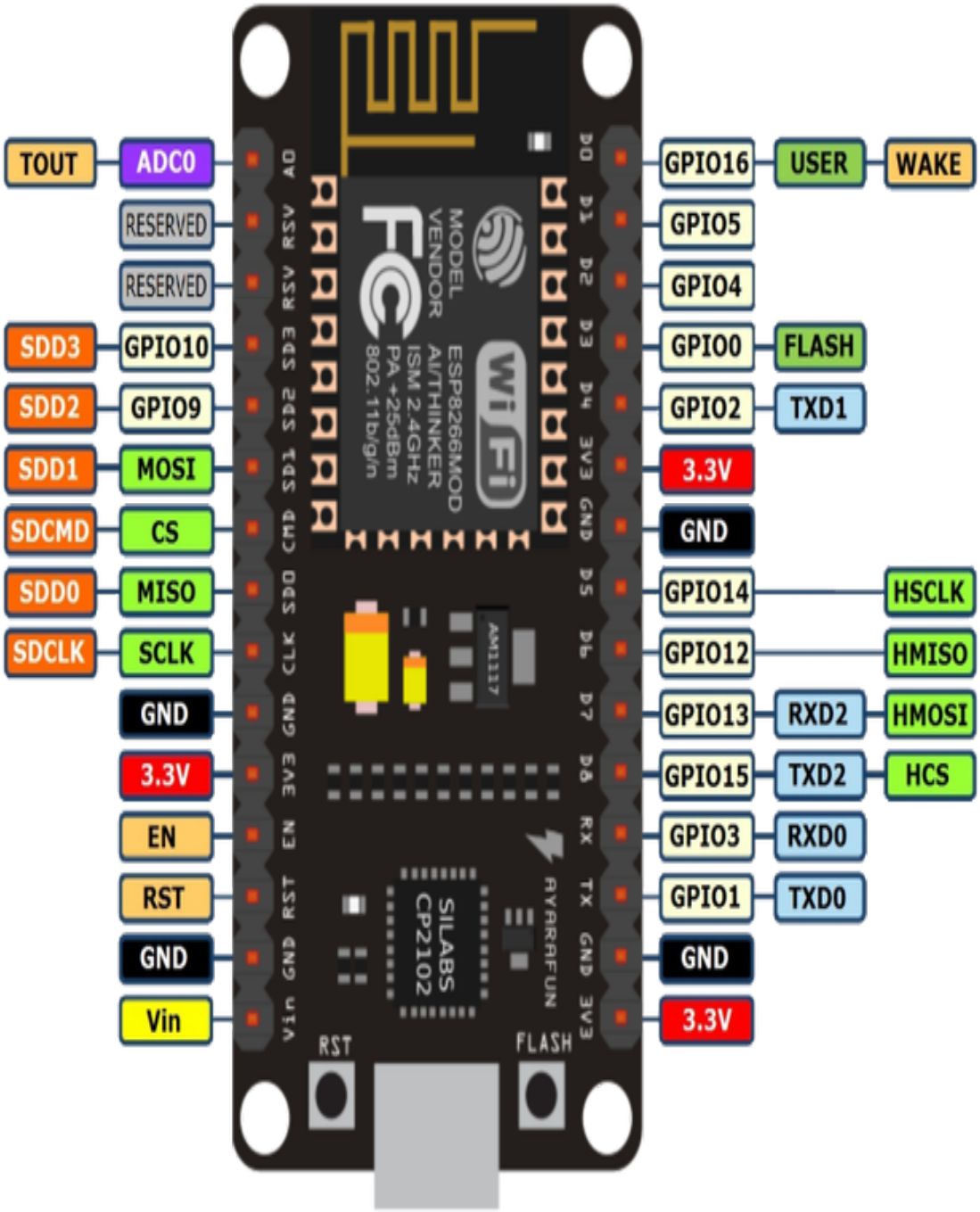
The Button is a Wi-Fi connected push button designed by Peter R Jennings The Button is designed for single-purpose, internet-enabled functions. When the button is pressed, a connection is made to a web server which will perform the desired task. Applications include a doorbell or panic button.

NodeUSB

NodeUSB is an open IoT platform about the size of a standard USB stick. It was designed to leverage NodeMCU (Lua) for easy programming and has the extra feature of USBcapability. It is ideal for Plug-n-Play solutions, allowing easy prototyping for developers.

ijWatch

ijWatch is an open-hardware and open-source Wi-Fi smartwatch, using an OLED screen and running NodeMCU firmware. The author believes it may be the first smartwatch. (As in, the watch itself is fully functional without the pairing of another bluetooth device such as a smartphone.)



Node mcu and its pins

SENSORS:

In the broadest definition, a **sensor** is a device, module, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics, frequently a computer processor. A sensor is always used with other electronics, whether as simple as a light or as complex as a computer. Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in micro machinery and easy-to-use microcontroller platforms, the uses of sensors have expanded beyond the traditional fields of temperature, pressure or flow measurement, for example into MARG sensors. Moreover, analog sensors such as potentiometers and force-sensing resistors are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine, robotics and many other aspects of our day-to-day life.

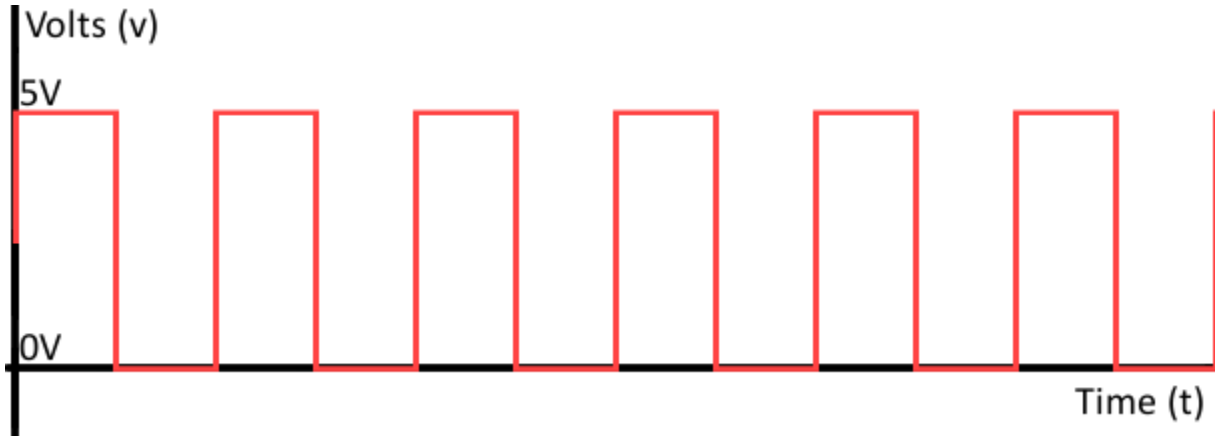
A sensor's sensitivity indicates how much the sensor's output changes when the input quantity being measured changes. For instance, if the mercury in a thermometer moves 1 cm when the temperature changes by 1 °C, the sensitivity is 1 cm/°C (it is basically the slope Dy/Dx assuming a linear characteristic). Some sensors can also affect what they measure; for instance, a room temperature thermometer inserted into a hot cup of liquid cools the liquid while the liquid heats the thermometer. Sensors are usually designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages. Technological progress allows more and more sensors to be manufactured on a microscopic scale as micro sensors using MEMS technology. In most cases, a micro sensor reaches a significantly higher speed and sensitivity compared with macroscopic approaches.

TYPES OF SENSORS:

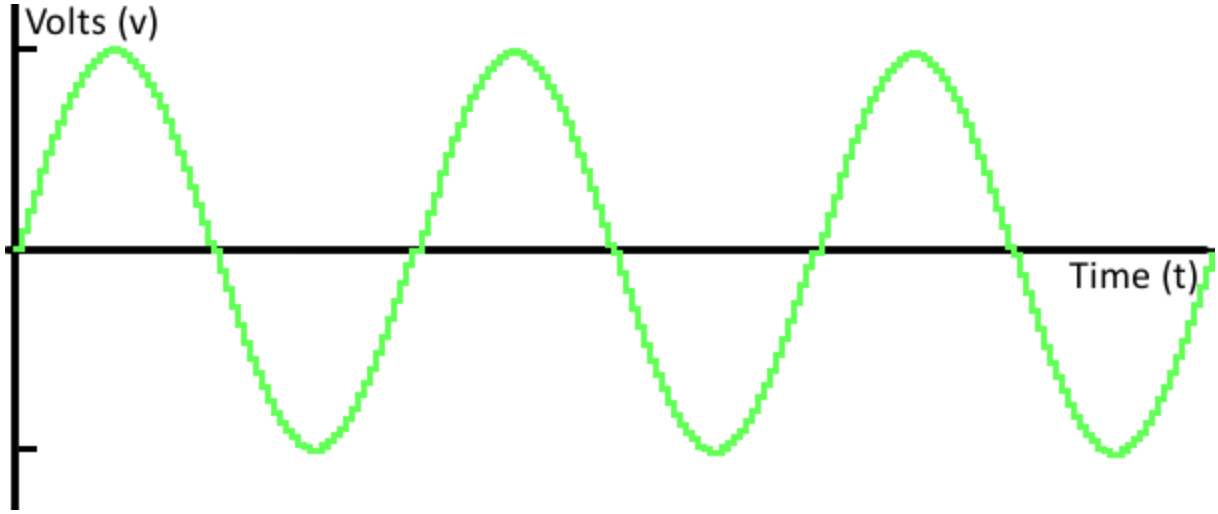
The sensors are majorly classified into analog sensors and digital sensors.

Digital Signals:

Digital signals must have a finite set of possible values. The number of values in the set can be anywhere between two and a-very-large-number-that's-not-infinity. Most commonly digital signals will be one of **two values** – like either 0V or 5V. Timing graphs of these signals look like **square waves**.



Or a digital signal might be a discrete representation of an analog waveform. Viewed from afar, the wave function below may seem smooth and analog, but when you look closely there are tiny discrete **steps** as the signal tries to approximate values:



That's the big difference between analog and digital waves. Analog waves are smooth and continuous, digital waves are stepping, square, and discrete.

Analog sensors use discrete (discontinuous) values to represent information for input. Where as digital sensors use only binary values i.e 1 and 0 to represent information for input

TOUCH SENSOR (DIGITAL SENSOR):

In this project we are going to use a touch sensor which is a digital sensor. Now let us see how it works and its specifications.

Today, capacitive touch sensors are widely used in consumer products like MP3 players, mobile phones and other portable devices. More and more the technology is utilized in further application fields such as household appliances as well as automotive and industrial applications. There are several reasons for this development.

1)Attractive product design: Users are able to design appealing products to distinguish from competitors. Touch sensors allow much more flexible designs as conventional mechanical buttons and sliders.

2) Durability: Touch elements do not contain moving parts like mechanical input devices. Thus no wearing out of these parts.

3)Robust housing design: It is easier to design devices for rough environments. Compared to conventional solutions, no holes or other openings are necessary where humidity and dust could enter the device

4)Cost: The manufacturing of the housing is simpler and cheaper because no openings and sealings are necessary. As manifold as the application fields for capacitive touch sensors, are the technical solutions. It is possible to implement a touch sensor in software with a Microcontroller together with some external components. The other end of the range is defined by fully integrated hardware

products. All methods detect the change of capacitance, if an electrode is touched by a human finger. Though this principle is the same for all methods, every approach feature specific advantages and drawbacks. The paper presents various capacitive sensing approaches and discusses the properties current consumption, response time, cost and reliability of these methods. An important aspect of touch user interfaces is reliable and robust operation. Since environmental parameters like humidity and temperature can influence the sensor behavior, it might be necessary to compensate these factors. Otherwise the system might become instable and false touches are detected. Methods to overcome this problem in actual applications will be presented in the paper. Since more and more applications feature a display, touch sensors in combination with screens are discussed in a third part of the paper. If the sensing electrodes are transparent, they can be placed onto a display and the user is able to control a device by pressing virtual buttons directly on the screen. Capacitive touch screens offer some advantages compared to resistive solutions which are very common in the industry and widely used already. In many cases, machines, home appliances and electronic devices have to be controlled by human beings. It is part of everyone's daily life and we are familiar with switches, push buttons, keyboards, knobs and slider controls. Since some time, a new species of control elements invades our life. It started in consumer products like mobile phones and MP3 players but moves into all kind of devices now. Those talks are about touch sensors. Simple electrodes

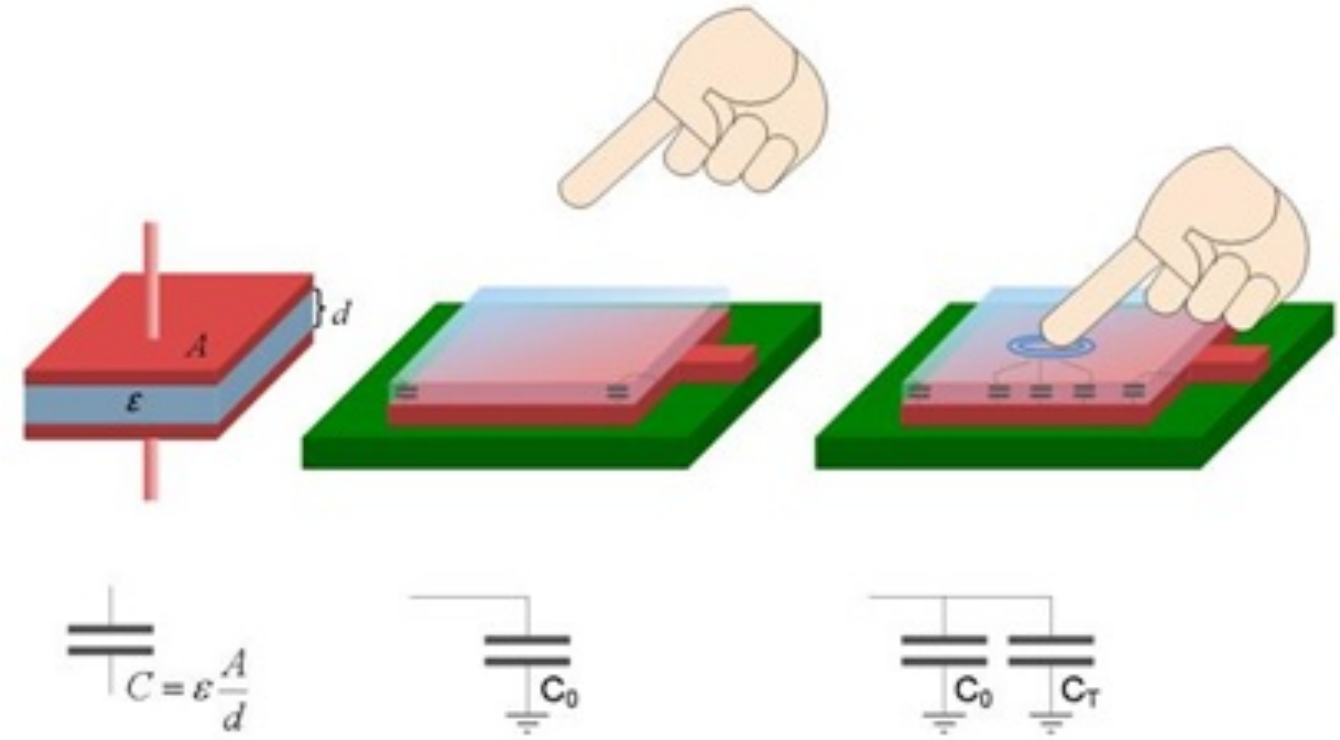
underneath the housing replace mechanical input devices with moving elements. The shape and layout of these sensor electrodes can be designed in a very flexible way, leading to appealing, modern product designs with enhanced usability. A wide range of elements can be implemented with touch sensor electrodes: simple buttons and keyboards, linear or circular sliders, transparent touch elements on displays or even buttons on wooden surfaces. As the sensor electrodes are placed inside the device, no openings are required. The housing is more robust and cost-effective and ideally suited for rough environments where dust and moisture could creep into the device. Especially for medical applications or devices used in clean environments like in the food industry, capacitive touch control enables hygienic casings. Conventional mechanical buttons or potentiometers with moving parts have a certain lifetime. Sooner or later they are worn out and do not work reliable anymore. Due to the lack of moving parts, capacitive touch is much more durable. However for highly stressed elements, the surface resp. overlay cover material of the touch electrode has to be considered. Glass or acryl may be better suited as plastics. For single, isolated buttons, even metal can be used, not for complete front covers however as the sensor pads must be isolated from each other. One positive aspect of the moving parts in conventional push buttons or switches is the tactile feedback to the user. By touching a surface, the user does not “feel” if the push button was triggered. This can be compensated by using optical, acoustical or, a bit more complex, vibration feedback.

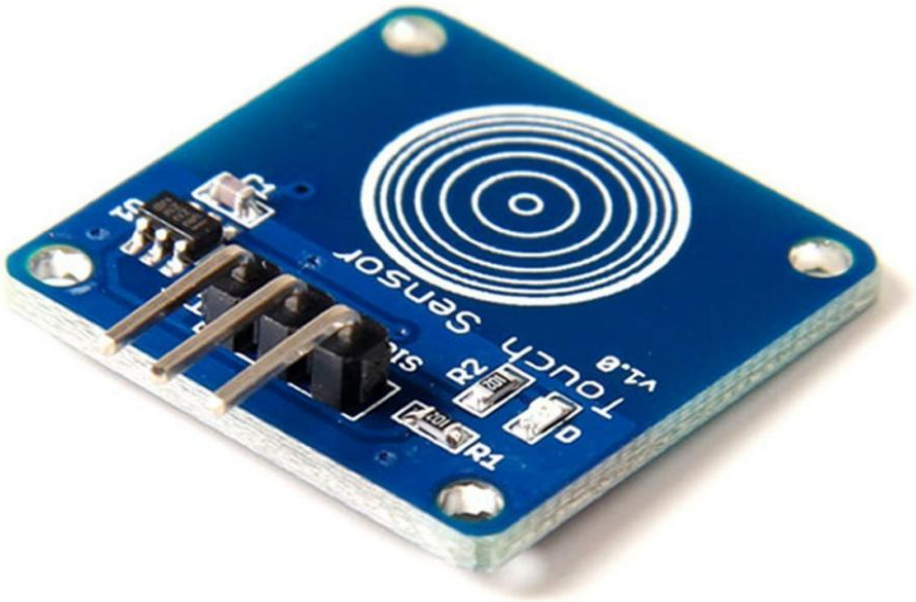
In the simplest case, the sensor circuit just detects a touch of the sensor pad and performs a yes/no decision. Buttons can be implemented in this way. For more sophisticated elements like sliders with higher resolutions, the “strength” of the touch, i.e. the quantity of CT has to be evaluated as well. In a linear slider design, several sensor pads are located close to each other. By touching the slider, more than one pad is influenced by the finger. Typically distributions of CT may be similar to the one shown in figure 4, are measured on the slider channels. Depending on the pad size and layout, the resolution of such a 5 channel slider can be as high as 100 steps by applying interpolation to the measurement results. The described delay to digital converter method features sensor channels with a CT resolution of 78fF and a dynamic range of 7.8pF, leading to 100 steps per channel. Hence it is excellently suited for interpolation algorithms. Usually the complete dynamic range of several pF is not used in practice. A finger may add a capacitance between 1 to 2 pF maximum, depending on the cover thickness and material.

Reliable operation:

In contrast to mechanical contacts, capacitive touch sensors do not require force to trigger a button. The presence of conductive material is sufficient. Therefore, the risk of unintended, false triggers is higher. In particular, water and moisture, which is a good conductor, are potential problems. Many capacitance measurement methods require a reference ground plane, located nearby the sensing pad. A

fingertip forms the capacitance between the sensing pad and the reference ground. Since the human body consists of around 70% water, a drop of water on such a pad layout is very similar to the fingertip and leads to false triggers. Reducing the sensitivity is no option because intended touches shall be detected reliable. There are proposals to distinguish between water and intended touches by additional guard sensing pads, special pad layouts with shield electrodes and software algorithms. However the best solution to overcome this problem is to get rid of the reference ground electrode. The differential measurement approach discussed earlier and implemented in Fujitsu's FMA1127/25 touch sensor controllers, use single pads only. With a resolution of 78fF, the sensitivity of the circuit is high enough to sense the self capacitance of the touch pad, which is influenced by the human body. No reference ground plane is required. Furthermore all sensor channels of the device are operated synchronously. This avoids cross coupling effects. Another aspect of reliable operation is the compensation of long term effects. Parameters like temperature, humidity etc. influence the measurement result. If these effects are not taken into account and be compensated, the sensitivity of the sensor changes over time. Under bad conditions, this can even lead to false triggers. Some methods register slow capacitance changes and assure a stable sensitivity. This feature is available on FMA1127 and called AIC – automatic impedance calibration.


A
ε
d
$C = \varepsilon \frac{A}{d}$
C0
C0
CT



Touch sensor

PINS:

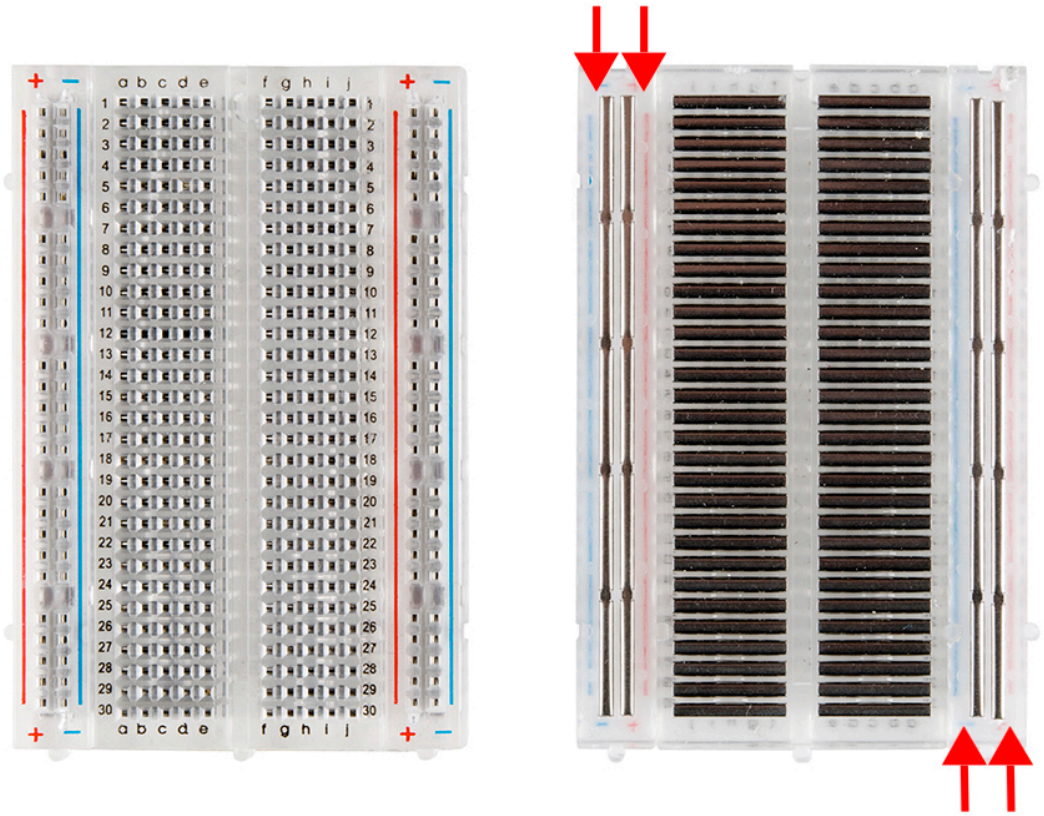
There are 3 pins in touch sensor.

One is vcc pin,

Second one is ground pin,

Third one is digital input signal pin.

BREAD BOARD:



A breadboard is a circuit board that is used to make temporary circuits. It is a device having electronics and test circuit designs. The electronic elements inside the electronic circuits can be interchanged by inserting the terminals and leads into holes and later connecting it with the help of appropriate wires. The device has stripes of metal below the board that connects the holes placed on the top of the board. The connections of the breadboard are mostly temporary and the elements can further be reassembled and reused without any damage. Breadboards

are generally used in electrical engineering. Engineers make use of breadboards in order to test different products made by them. Using breadboard is the most efficient way of testing and also they are cost effective. They can be reused again and again for the purpose of testing. Today, starting from tiny analog, digital circuits to big complicated CPU's everything can be tested with the help of this. Breadboards earlier were made of copper wires or terminal strips. These days it is made up of white plastic and is a breadboard that can be plugged. Breadboards are solderless and they are made of two kinds of strips i.e. terminal and bus strips. Terminal strips help in holding the electronic elements while the bus strip is used to power electric power to all the electronic components. You can find manufacturers selling solderless breadboards very easily, some manufactures sell the bust and terminal strips separately and some sell it together. A breadboard is a circuit which if of a temporary nature used for the purpose of testing and prototyping circuits. It is easy to prototype circuits with the help of breadboards because it is fast and easy. Breadboards are generally used to test circuits. As this device have holes in it. In order to form a circuit, wires are inserted simply inside the holes. Basically, a chunk of plastic with a bunch of holes. However, something special is going on inside the breadboard! Although you can't see it, inside the breadboard are many strips of metal that connect the rows and columns together. If you look on the back of your breadboard, there's a yellow waxy paper covering some sticky foam. If you were to peel back that foam you'd see dozens of these

metal rows. For really big projects, give yourself some room to work in, with a massive 2250-point breadboard - equivalent in size to three full sized breadboards side by side. The breadboards are mounted onto a metal plate, and comes with 4 colored posts you can use with a bench-top supply. Four bumpers are included, to keep the board from slipping around your desk. Like nearly all large breadboards, the 'power rails' are split in the middle. That means that if you want to plug in a voltage at the top of the board, it wont appear at the bottom. Aside from horizontal rows, breadboards usually have what are called power rails that run vertically along the sides. These power rails are metal strips that are identical to the ones that run horizontally, except they are, typically*, all connected. When building a circuit, you tend to need power in lots of different places. The power rails give you lots of easy access to power wherever you need it in your circuit. Usually they will be labeled with a '+' and a '-' and have a red and blue or black stripe, to indicate the positive and negative side.

Jumper Wires

Jump wires (also called jumper wires) for solderless. Bread boarding can be obtained in ready-to-use jump wire sets or can be manually manufactured. The latter can become tedious work for larger circuits. Ready-to- use jump wires come in different qualities, some even with tiny plugs attached to the wire ends. Jump wire material for ready-made or homemade wires should usually be 22 AWG (0.33 mm2) solid copper, tinplated wire - assuming no tiny plugs are to be attached to the wire ends. The wire ends should be stripped 3⁄16 to 5⁄16 in (4.8 to 7.9 mm). Shorter stripped wires might result in bad contact with the board's spring clips (insulation being caught in the springs). Longer stripped wires increase the likelihood of short-circuits on the board. Needle-nose pliers and tweezers are helpful when inserting or removing wires, particularly on crowded boards.

Differently colored wires and color-coding discipline are often adhered to for consistency. However, the number of available colors is typically far fewer than the number of signal types or paths. Typically, a few wire colors are reserved for the supply voltages and ground (e.g., red, blue, black), some are reserved for main signals, and the rest are simply used where convenient. Some ready-to-use jump wire sets use the color to indicate the length of the wires, but these sets do not

allow a meaningful color-coding schema.

Jumper wires with clips:

TYPES OF JUMPER WIRES:

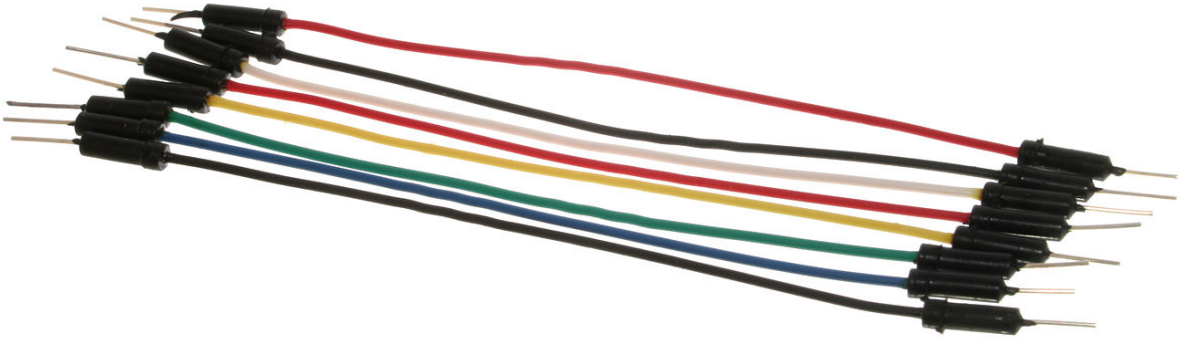
We have many 3 types of jumper wires namely,

1. male to male jumper wires.

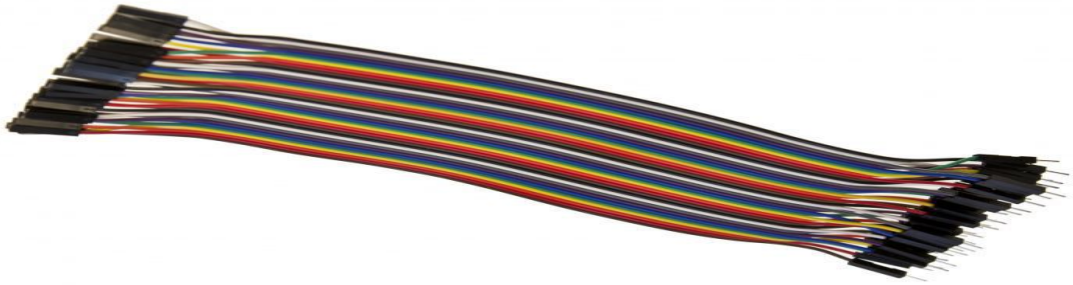
2. female to female jumper wires.

3. male to female jumper wires.

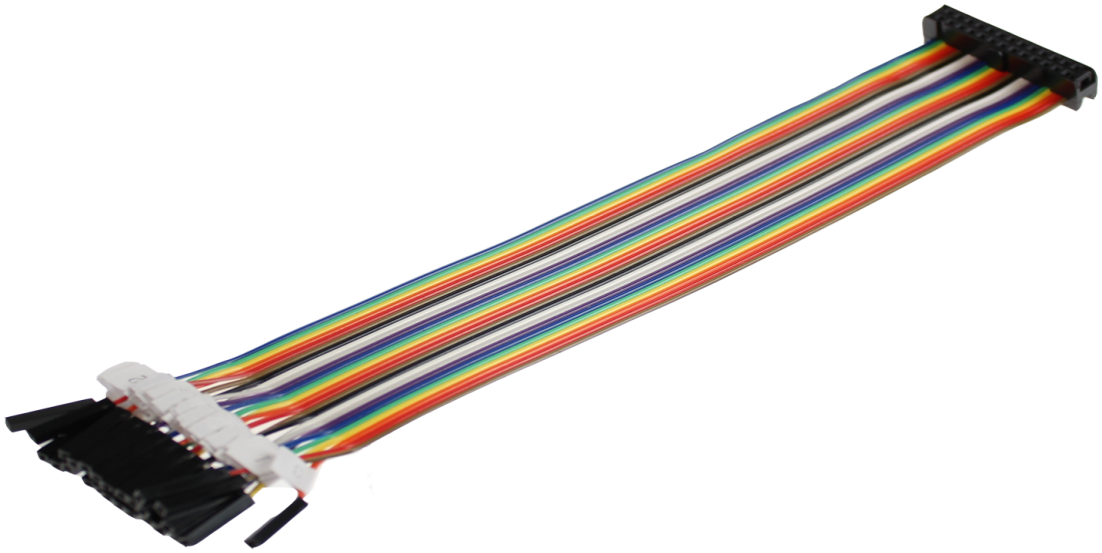
We can use these jumper wires according to our needs. The images of the types of jumper wires are as shown below.



Male to male jumper wires



Male to female jumper wires



Female to female jumper wires

CHAPTER – 3

TECHNOLOGIES INVOVLED

BLYNK PLATFORM

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.It's really simple to set everything up and you'll start tinkering in less than 5 mins.Blynk is not tied to some specific board or shield. Instead, it's supporting hardware of your choice. Whether your Arduino or Raspberry Pi is linked to the Internet over Wi-Fi, Ethernet or this new ESP8266 chip, Blynk will get you online and ready for the Internet Of Your Things. Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

There are three major components in the platform:

1.Blynk App - allows to you create amazing interfaces for your projects using various widgets we provide.

2.Blynk Server - responsible for all the communications between the smart phone and hardware. You can use our Blynk Cloud or run your locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

3.Blynk Libraries - for all the popular hardware platforms - enable

communication with the server and process all the incoming and outcoming commands.

Now imagine: every time you press a Button in the Blynk app, the message travels to space the Blynk Cloud, where it magically finds its way to your hardware. It works the same in the opposite direction and everything happens in a blynk of an eye.


T-Mobile
6:24 PM
27%
TEMPERATURE C°
V27
24.786
HUMIDITY. %
V22
28.6
BEDROOM
V5
1h
6h
1d
1w
KITCHEN
D7
ZERGBA
V15 255 255 255
Widget Box
YOUR ENERGY BALANCE
2000
+Add
DISPLAYS
Value Display
200
LED
100
Gauge
300
LCD
400
Graph
400
History Graph
900

FEATURES:

- Similar API & UI for all supported hardware & devices
- Connection to the cloud using:
  - WiFi
  - Bluetooth and BLE
  - Ethernet
  - USB (Serial)
  - GSM

- Set of easy-to-use Widgets
- Direct pin manipulation with no code writing
- Easy to integrate and add new functionality using virtual pins
- History data monitoring via History Graph widget
- Device-to-Device communication using Bridge Widget
- Sending emails, tweets, push notifications, etc. new features are constantly added.

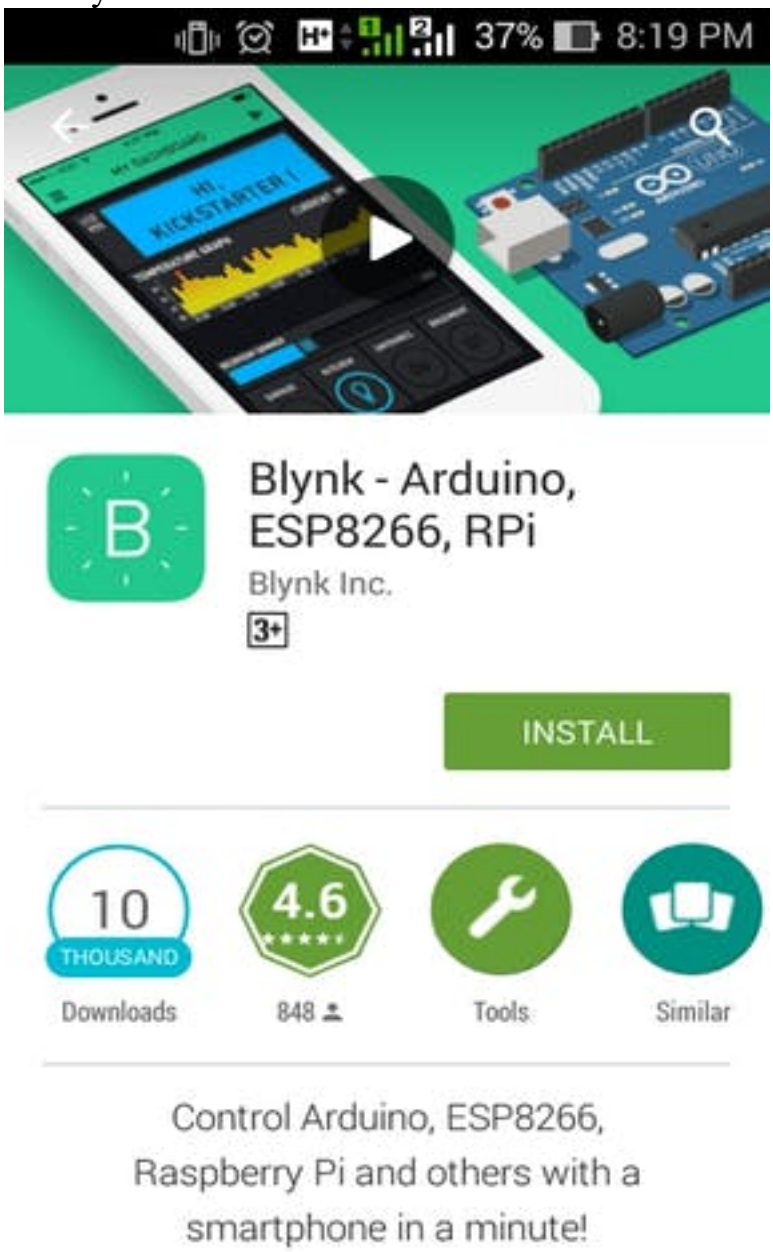
You can find example sketches covering basic Blynk Features. They are included in the library. All the sketches are designed to be easily combined with each other.

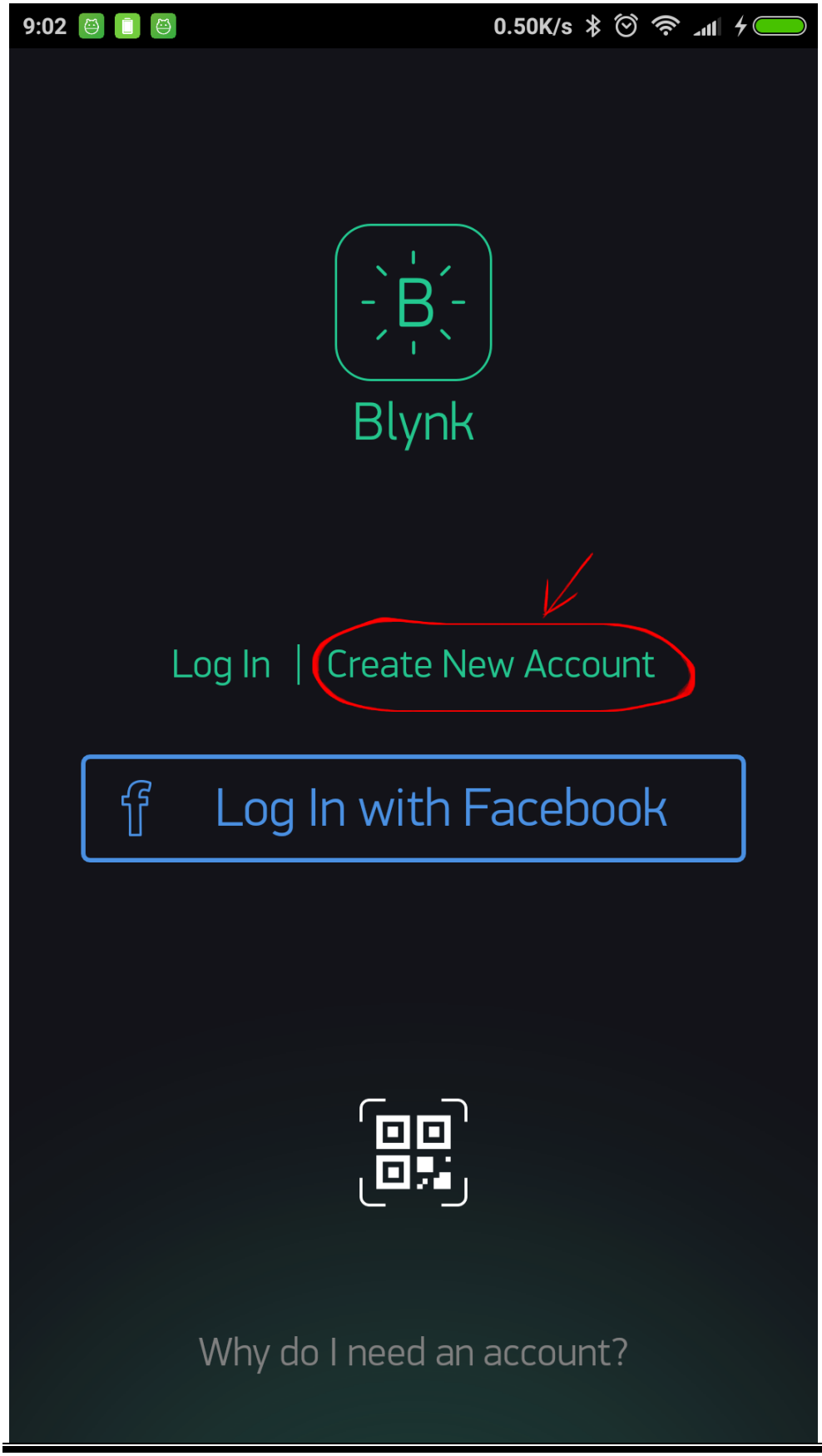
Blynk works over the Internet**.** This means that the hardware you choose should be able to connect to the internet. Some of the boards, like Arduino Uno will need an Ethernet or Wi-Fi Shield to communicate, others are already Internet-enabled: like the ESP8266, Raspberri Pi with WiFi dongle, Particle Photon or SparkFun Blynk Board. But even if you don't have a shield, you can connect it over USB to your laptop or desktop (it's a bit more complicated for newbies, but we got you covered). What's cool, is that the list of hardware that works with Blynk is huge and will keep on growing.

creating an account:

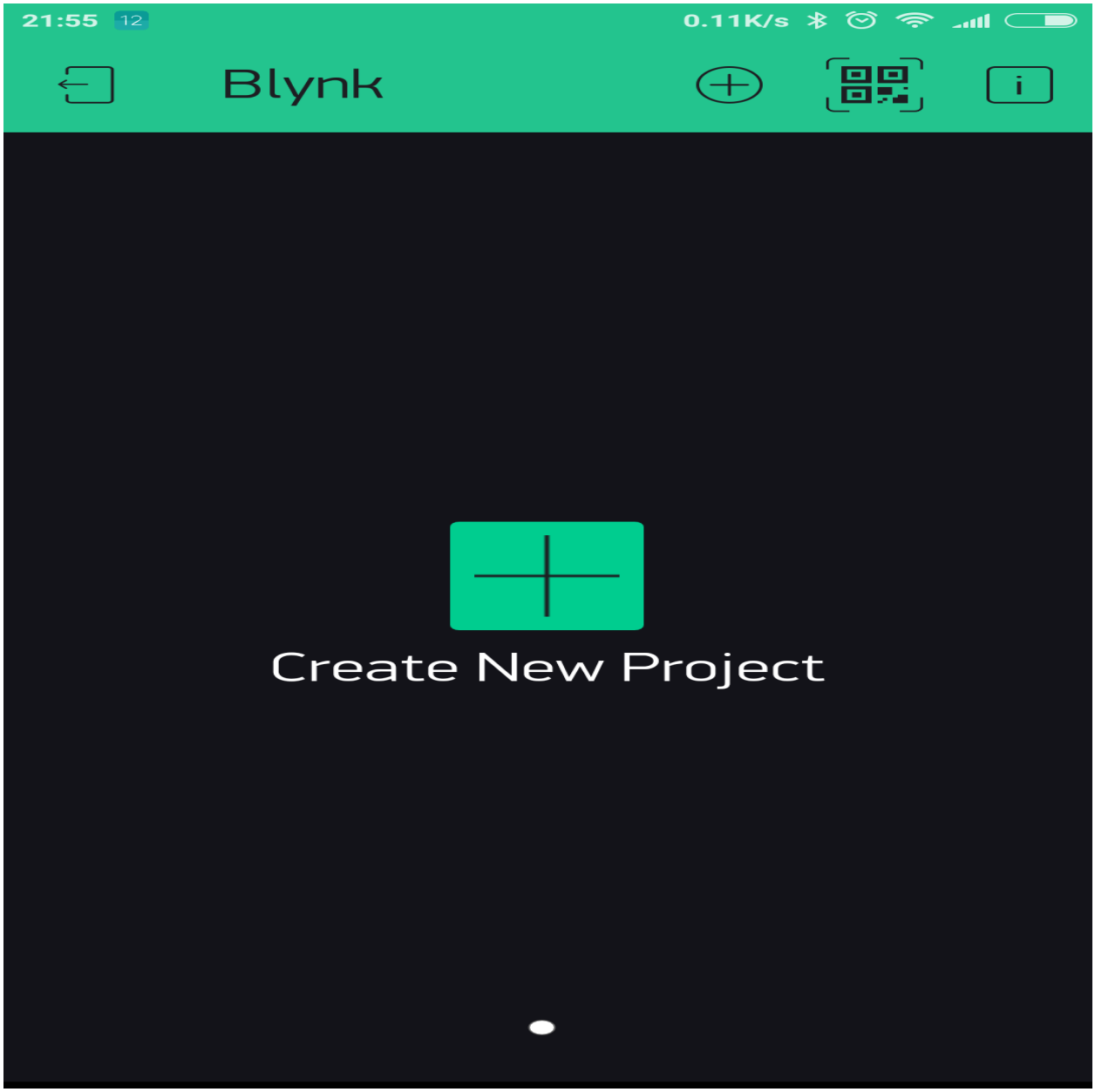
After you download the Blynk App, you'll need to create a New Blynk account.

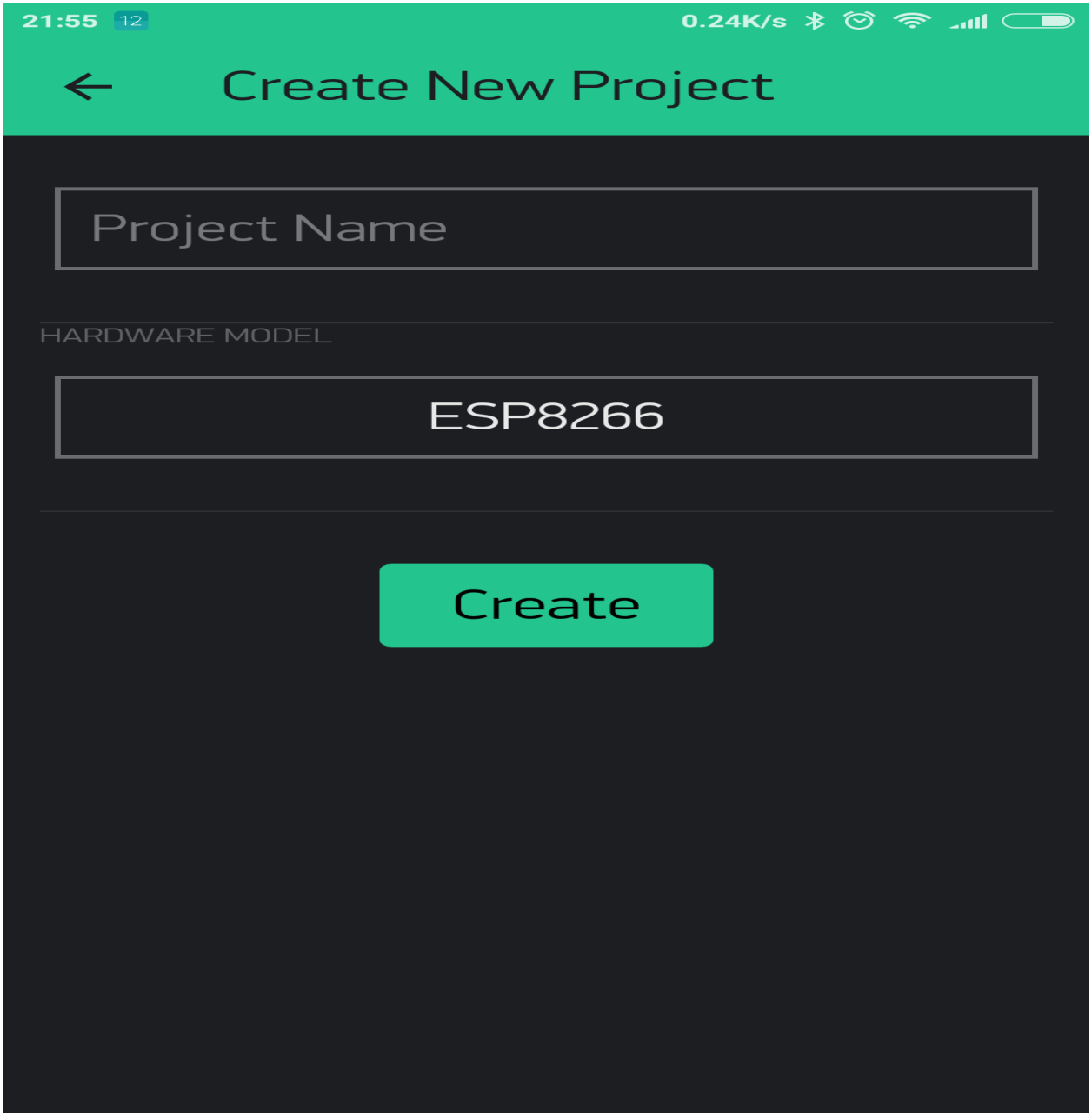
This account is separate from the accounts used for the Blynk Forums, in case you already have one.



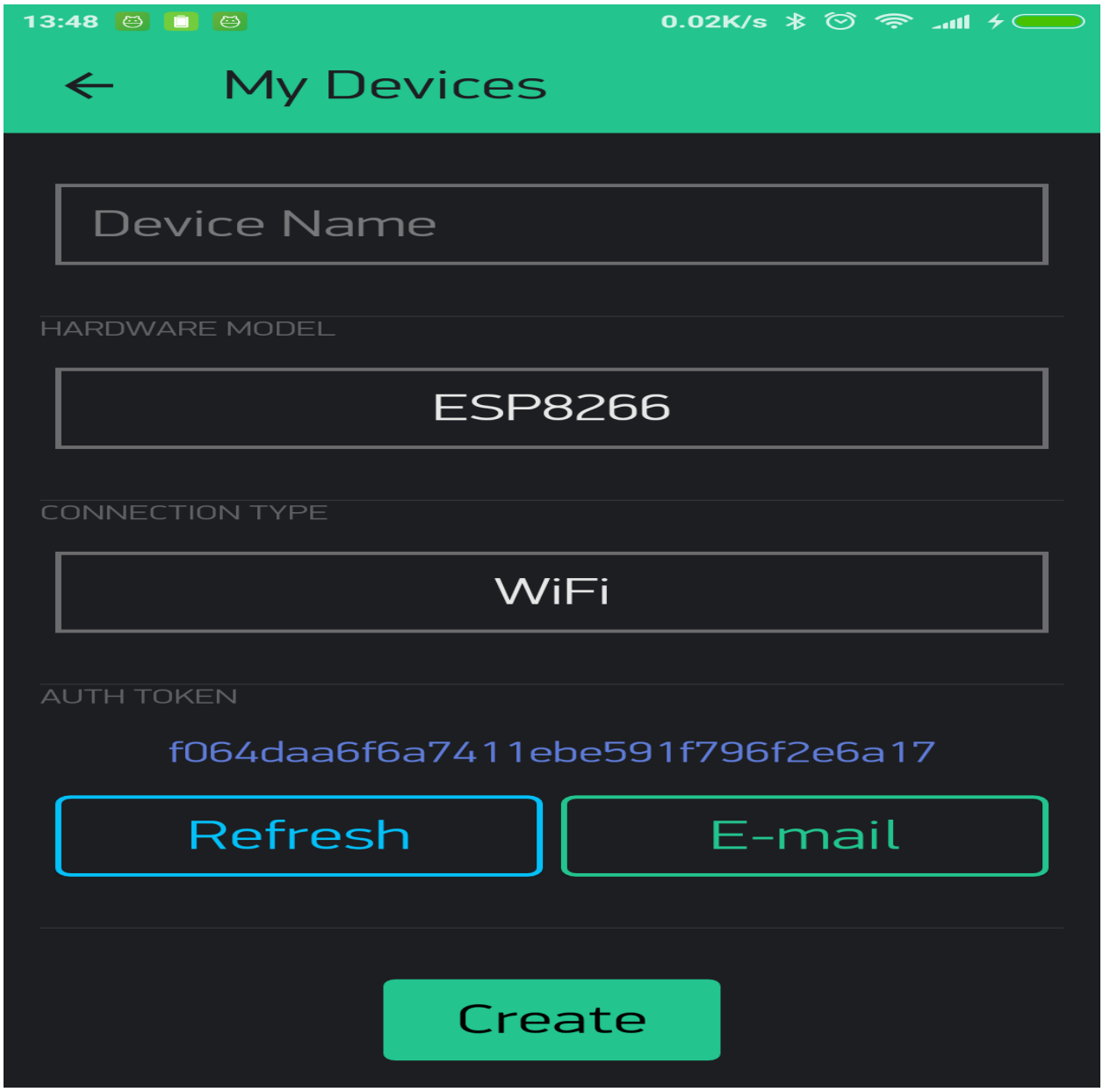

9:02
0.50K/s
Blynk
Log In | Create New Account
Log In with Facebook
Why do I need an account?

CREATE A NEW PROJECT:



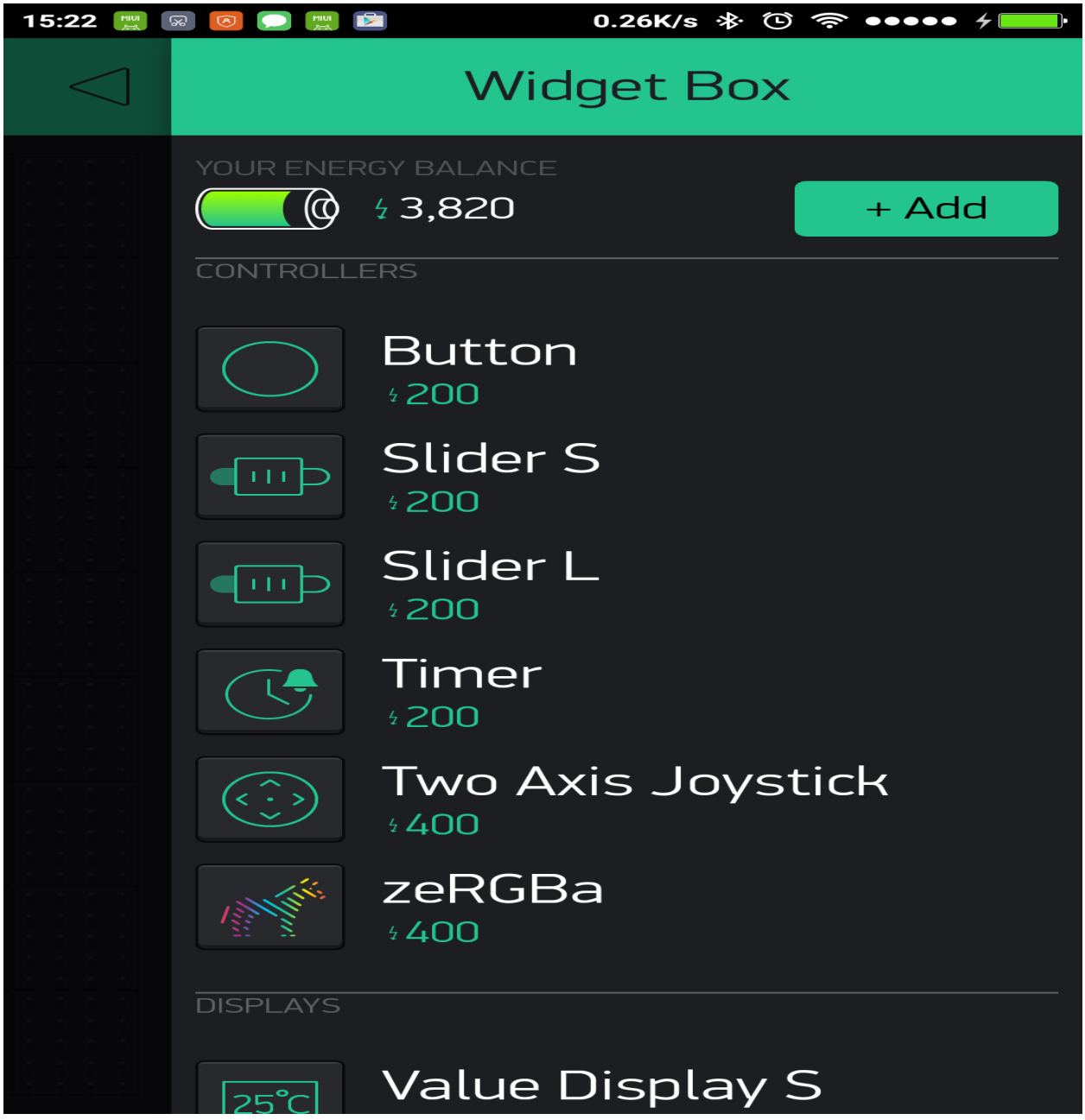


Select the hardware model and name the project



After creating, we will get a auth token to our mail, which should not be shared.

Tap anywhere on the canvas to open the widget box. All the available widgets are located here. select an email widget and if u want an notification widget also.



CHAPTER-4

PROJECT DESIGN

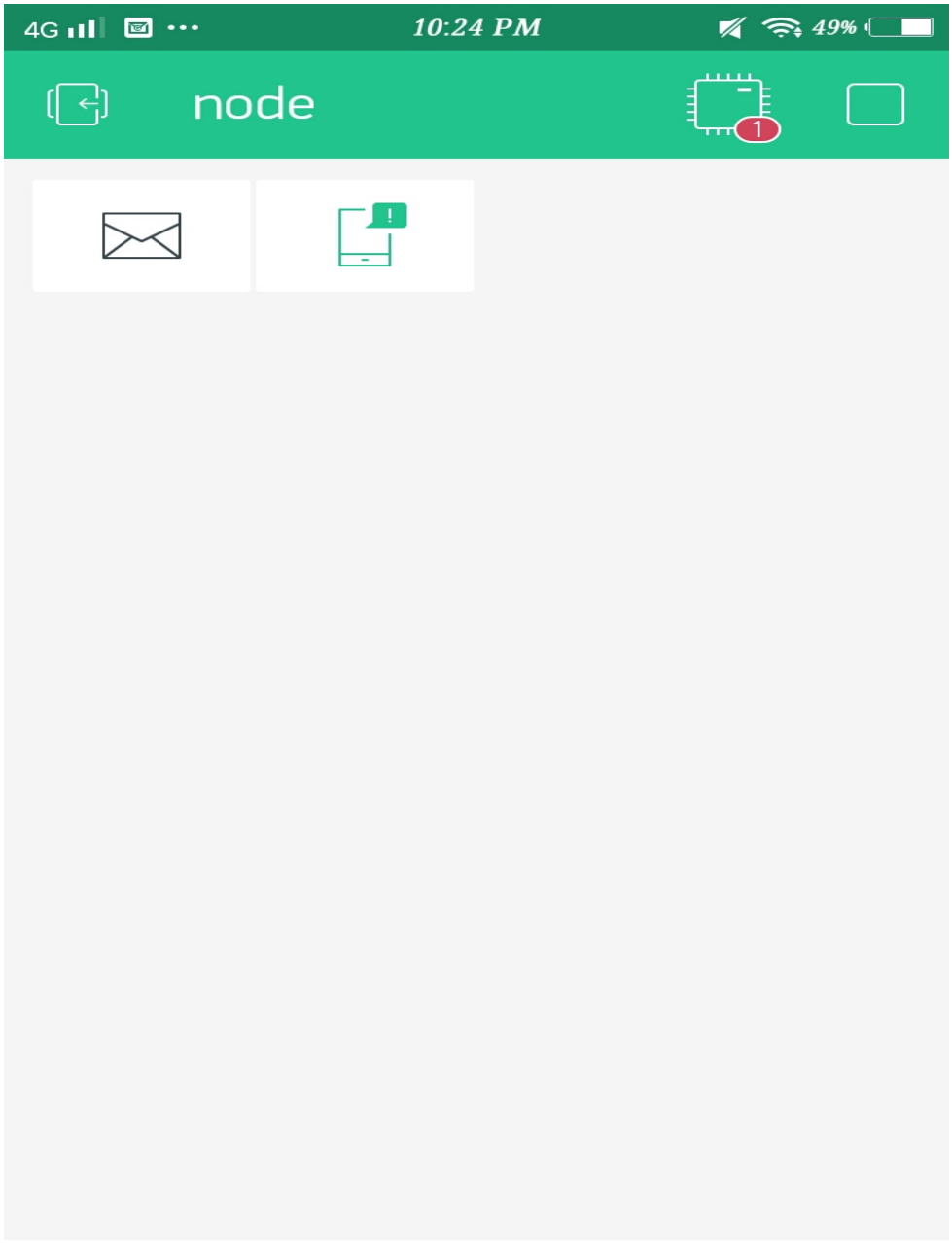
HARDWARE CONNECTIONS:

Coming to hardware connections as we don't have 5v in nodemcu we can connect the vcc pin and ground pin either to the arduino board of 5v and ground pin or to the bread board power supply. The bread board power supply is connected to power by SPMS adapter. The digital pin of the sensor is connected to the digital pin of nodemcu. Coming to buzzer, the buzzer works on 3.3v. so we will connect the vcc pin of buzzer to 3.3v, ground pin to ground pin of nodemcu and finally the signal pin of the buzzer to the digital pin of nodemcu. And finally we should upload the code written in arduino. The code is as follows:

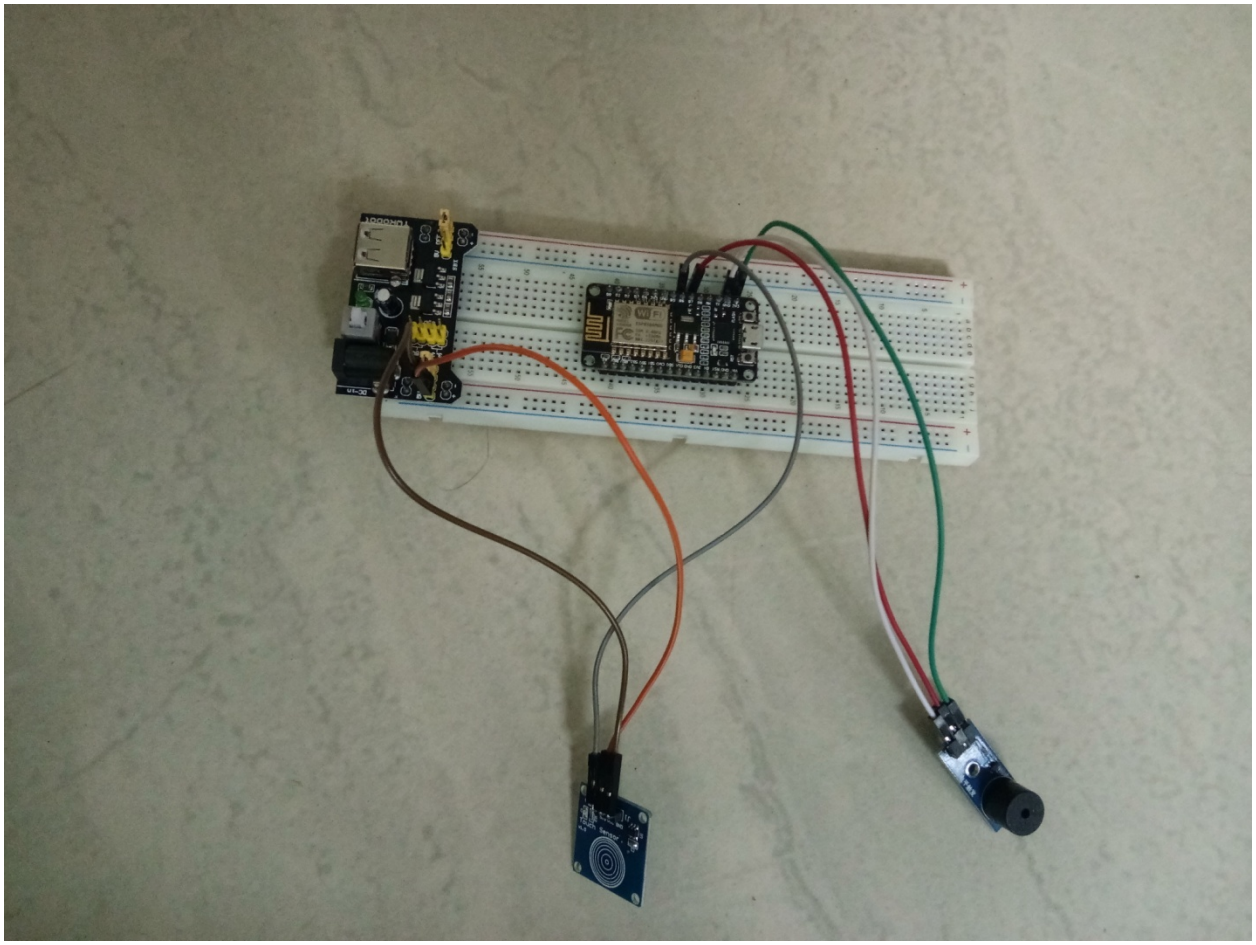
```
#include <Blynk.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[]="xxxx"; //enter your auth//
char ssid[]="xxx"; //enter your wifi username//
char pass[]="xxx"; //enter your wifi password//
void setup() {
  Serial.begin(9600);
  Blynk.begin(auth,ssid,pass);
  pinMode(D6,INPUT);
  pinMode(D7,OUTPUT);
```

```
}
void loop() {
  int sensorValue=digitalRead(D6);
  if (sensorValue==1)
  {
    digitalWrite(D7,HIGH);
    Blynk.email("xxxxx@gmail.com","high","some one touched your sensor");
  }
  else
  {
    digitalWrite(D7,LOW);
  }
 Blynk.run();
}
```

Add a email and notification widget:



After connections:

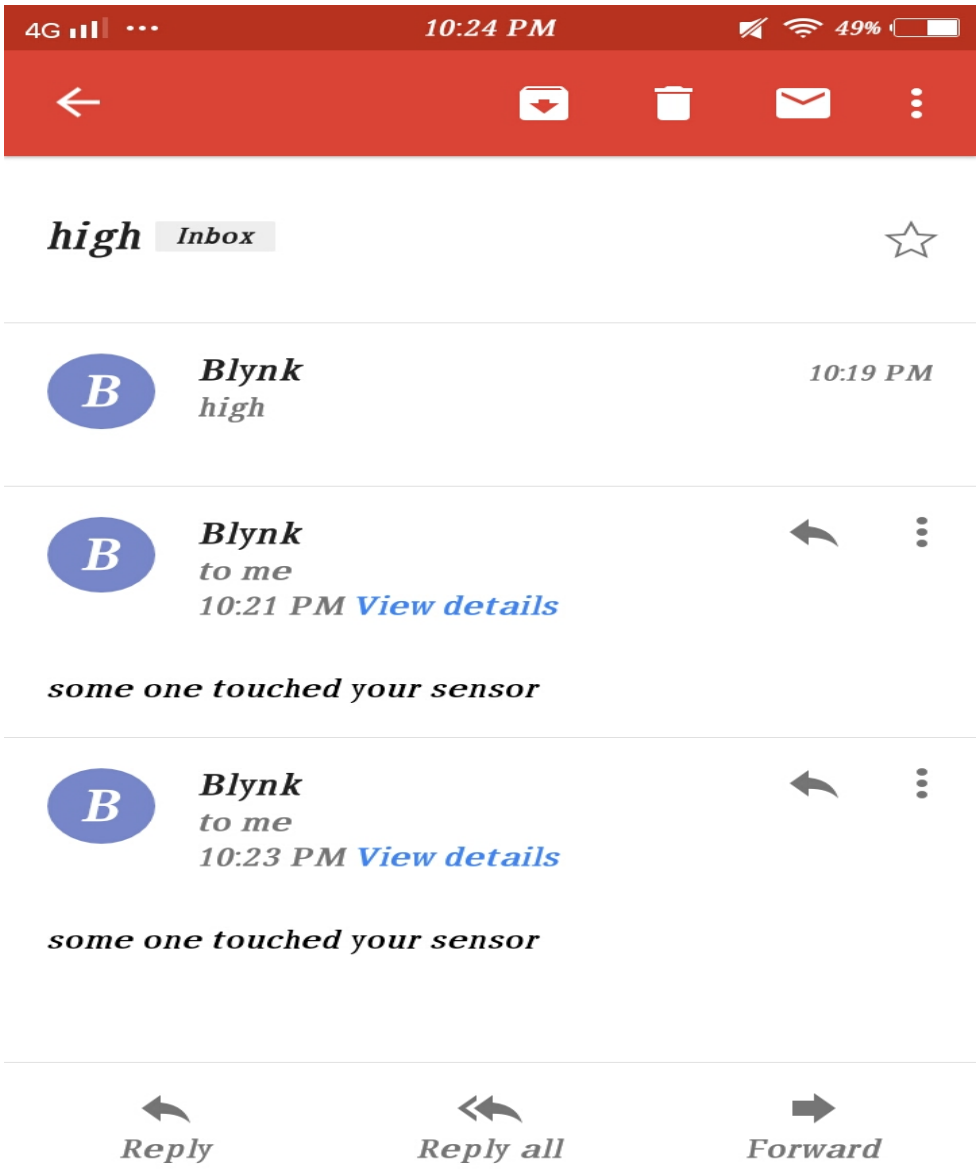


CHAPTER-5

RESULTS

RESULT:

The project named “THEFT DETECTION USING BLYNK PLATFORM” is done. Its will display as follows:



Chapeter-6

Advantages, disadvantages and applications

ADVANTAGES:

- This project is very useful to know theft detections.
- This will save our the time.
- By knowing the theft happening, we can take the necessary steps to prevent the theft.

DISADVANTAGES:

- Once the reset button is pressed in the NodeMcu,it doesn't work and we should again upload the code for controlling the appliances.
- As we are using wifi,it works upto some distance only in our house.
- The memory of NodeMcu device is limited to 4Kbytes.It can't be raised.

APPLICATIONS:

- It has many household applications.
- accurate quality
- low operating costs .
- capacitive touch sensors are user interface controllers that manage multiple configurations of touch pads, sliders, rotary positions and mechanical keys.
- high quality data.

CHAPTER-7

CONCLUSION

CONCLUSION:

Thus theft detection is made easy by this project because we are reducing the mankind labour and we are reducing all the needs and use of only some money in using this project and hence we are saving money in implementing this project and we are also using modern technology in day to day life and hence this IoT projects we will bring a revolution in our life.

REFERENCES:

TEXT BOOKS:

1. Systems, Espressif, ”Espressif Systems”, EspressifWikiDevi. Retrieved 3 june 2017.
2. Jump up Brain Benchoff. “A DEV BOARD FOR THE ESP LUA INTERPRETER”. Hackday. Retrieved 2 April 2015.

WEB PORTALS:

1. https://www.youtube.com/playlist?list
2. Http://easycoding.tn/index.php/resource