INTERNET of THINGS (IoT): INTEGRATION of BLYNK for DOMESTIC USABILITY

HIRAL S. DOSHI #1, MINESH S. SHAH #2, UMAIR S A. SHAIKH #3

DEPARTMENT OF COMPUTER ENGINEERING, VIIT, PUNE

Abstract: Internet of things is just a hype in the market. There was internet of machines, internet of smart phones and now the internet of things. We are now in the era of internet of things and in this era hardware platforms like Arduino, Raspberry Pi, Orange Pi, etc. and Cloud based services like AWS (Amazon Web Services), Firebase, Blynk, Canne, etc. make up the internet of things come to life. Using internet of things has not only made naïve devices smarter but also has reduced the human intervention while keeping the accountability higher and keeping monthly bill in check. In this paper, we are going to implement a simple yet effective approach to build the IoT platform for remote monitoring and sensing of data, home automation, industrial automation and many more.

Keywords: IoT, Arduino, Blynk, Cloud, Remote sensing and monitoring, Home automation, Industrial automation, Smart systems.

I. INTRODUCTION

The internet of things(IoT) is network of naïve devices communicating with each other over the internet to perform a unified task in a smarter way. The IoT architecture consists of wirelessly embedded sensors, nodes, actuators and a micro-controller to perform and take intelligent decisions. Here, the object, devices, machines, animals, humans are provided a unique identification tag which enables it to be identified from a remotely located area and transfer data without requiring HCI (Human-Computer Interaction).

IoT has become a way of life in recent years, as it has a great potential in helping human life and focus on the tasks that usually needs human intelligence. IoT has a very broad spectrum that covers every aspect of human interaction in daily life, from buying goods to health care, even monitoring of resources and remotely handling them. With IoT insight human resources can be much efficiently utilized while sparing the burden over natural resources.

Imagine a world where your refrigerator becomes your helper by creating a checklist of all the items required in the next few days based on your present consumption and preferences and forwards it to your handheld devices.
your storage cabinets of the kitchen and suggesting you the recipes based on the available ingredients. Now, your refrigerator is communicating with your microwave after you select a recipe based on the recommendations and your microwave starts preparing meal and serves you hot meal when you reach home after a tiring day. Doesn’t this seem to be a part of magical world but NO its just a part of the IoT World which is not far from the real world.

Now let’s come to another scenario which might seem hypothetical but is a trending application in IoT. Smart Health Care, a domain that affects all our lives. Imagine that we are injected with smart implants in our body that can analyse our blood for virus and bacteria contents and send this report to centralized system that automatically delivers you the medicine as per the analysis of blood content, sounds too good to be true but with the advancement in technology it could be the reality of near future, so now more value for your prestigious time rather than waiting in a long queue to get diagnosed. Health just got a new upgrade through IoT.

II. RELATED TECHNOLOGIES

A. Arduino Uno [2]

When we talk about IoT there is basic involvement of certain technologies. So here we will showcase these essential building blocks for IoT. We need to have a good internet connection which forms the backbone for data transfer. Internet is a necessity so that you can transfer the sensed data from the nodes to the master hub.

Conversely, if internet is not present, technologies like radio frequency, Bluetooth and Wi-Fi can be used for low proximity data transfer and for long range distance communications, IoT technologies like ZigBee, Lora etc. can be implemented to cater the data transfer. The above technologies only cover’s the communication part, but also, we need some type of device to process the data. This is where the microcontroller devices come’s in picture. The industry is flooded with vast range of devices ranging from raspberry pi, to BeagleBones to nodeMCU, but the king of IoT based devices has been and is ARDUINO.

The reason behind ARDUINO’s popularity is easy to use IDE, open source documentation and a very vast community of active Arduino developers. Another key reason behind its wide acceptability is its low expensiveness and miniature footprint. The programming of Arduino is done in C, C++. The IDE provides various built in examples and codes that make working with Arduino an effortless task.

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, 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. You can tinker with your UNO without worrying too much about doing something wrong. In the worst-case scenario, you can replace the chip for a few dollars and start over again.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino
boards, and the reference model for the Arduino platform.

B. Blynk [3]
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, visualize it and do many other cool things.

There are three major components in the platform:

**Blynk App:** – It allows you to create amazing interfaces for your projects using various widgets which are provided.

**Blynk Server:** – It is responsible for all the communications between the smartphone and hardware. You can use the Blynk Cloud or run your private Blynk server locally. It’s open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

**Blynk Libraries:** – It enables communication, for all the popular hardware platforms, 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 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.

![Fig 3: Blynk app overview.](image)

**Fig 3: Blynk app overview.**

![Fig 4: Blynk cloud architecture.](image)

**Fig 4: Blynk cloud architecture.**
Characteristics of Blynk are:

- Similar API & UI for all supported hardware & devices
- Connection to the cloud can be done using Ethernet, Wi-Fi, Bluetooth, BLE and USB (Serial)
- 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.

C. Sensors

1. Temperature Sensor (LM35):
   The temperature sensor commonly known as LM35 is used to sense the present temperature of the environment. The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling.

   The proposed system is about how to create a simple yet effective Internet of Things environment which is capable of monitoring and remotely controlling actuators and sensors.

   The temperature sensor interfaced with the Arduino senses the environment temp & forwards it to the analog pin on the Arduino. The Arduino compares the sensed value with the predefined threshold value & if the value is greater than the threshold value then it sends the email to the registered account. The email recommends the user to turn on his cooling device at least half an hour before he steps into the house. The user can do so by using the blynk app as it provides the controlling functionality from a remote place.
2. **LDR:**
   A photo resistor (or light-dependent resistor, LDR, or photocell) is a light-controlled variable resistor.
   The resistance of a photo resistor decreases with increasing incident light intensity; in other words, it exhibits photoconductivity.

   **A. LDR and Arduino interfacing.**

   In this part, we will discuss how LDR sensor needs to be connected with Arduino and to display the sensed value on your mobile app. To begin with, we need to connect the Arduino to the computer and run the Arduino IDE.

   The LDR needs to be connected in the following sequence:

   1. First select any leg of the resistor and connect it to the 5V pin on the Arduino Uno.
   2. Now, the next leg of resistor need to be connected to any one of the leg of LDR sensor.
      This point of contact also needs to be connected to any of the Analog pin on Arduino Uno.
   3. The remaining pin of LDR sensor needs to be connected to ground pin on Arduino.
   4. Once connected we can write code on Arduino that will read the LDR value and display it on serial monitor.

   **B. LM35 and Arduino interfacing**

   In this part, we will discuss how LM35 sensor needs to be connected with Arduino and to display the sensed value on your mobile app. To begin with, we need to connect the Arduino to the computer and run the Arduino IDE.

   The LM35 needs to be connected in the following sequence:

   1. The sensors flat side should be faced towards us and then the left pin should be connected to +5V pin on Arduino.
   2. The centre pin is the output pin of the sensor. The output provided by this sensor is in analog format. This value is handled by Arduino’s analog pin and later it can be converted to digital values.
   3. The right pin should be connected to the any of the ground pin on Arduino Uno.
IV. GENERAL STEPS FOR CONNECTION

1. Connect the Arduino to pc and run the Arduino IDE.
2. Download the blynk library from www.blynk.cc and add the library to the ide.
4. After selecting the example, the example code will show up.
5. In this example, we need to make some changes, for this we need to login on Blynk app using android app.
6. After logging in, create a new project and select the type of board.
7. In this type, it will be Arduino uno. After this we will receive an authentication key, select this authentication key and paste it in the example sketch under authentication token.
8. After this step click compile and upload button to burn it on the Arduino. After receiving “Done uploading”, the example sketch is successfully loaded an script should be executed which is located under Arduino-libraries-blynk-scripts.
9. The file name is blynk-ser.bat. In order to run the script type cmd on search tab then run it as an administrator. When cmd is opened type cd and path name of the script file when on the right path type blynk-ser.bat -c COM4 where COM is a port where your Arduino is connected at.
10. Make changes according to your own setup, once this script is up and running, the Arduino is connected to the blynk server over the usb without any shields. This is the easiest way to make your device go online without much hardware.
11. Here we can connect the sensor discussed in the previous discussion. Considering the ldr we will connect it in the same way we discussed and the output to pin A0.
12. Now on the app, we have created our project, so now click on the widget box icon which is a circle with a plus sign on it.
13. Then select on the gauge icon, the icon will appear onto the blynk workspace area. To make this gauge work, click on the gauge on the workspace then under input, select pin analog A0 and reading frequency A0.
14. With this we have setup the gauge to be connected to pin A0 of Arduino where we have connected our LDR sensor, click on the play button on the app to run this code.
15. When we click on the play button the Arduino and Blynk app have established connection and the gauge shows the value of the LDR sensor.
16. Now click on the stop button, then go on the widget box and select button. The button shows on the workspace, click on it and select the output to be on pin D13, there is an option to keep the button as push type or switch.
17. After this go back and select the play button now we can see the LDR value and have a button from which we can control the gpio D13 on Arduino as well as the on board led connected to pin D13.
18. VOLA!! you have successfully made an internet of things (IoT) device that you can control over the internet as well as get sensor readings.

V. EXPERIMENT
The objective of this experiment is to trigger an event in this case an email whenever the temperature sensor senses the value greater than threshold.
VI. SOLUTION

For the above proposed system to be accomplished we have divided it into following 2 modules:

1. Hardware side implementation:
   The sensor will send the data to the Arduino
   On the Arduino side the calculations will be performed. Using the installed blynk server script, the data will be sent to the blynk server.

![Fig 11: Interfacing of LDR with Arduino uno.](image1)

![Fig 12: Bulb relay “ON” using blynk and arduino.](image2)
Fig 13: Bulb relay “OFF” using blynk and arduino.

With reference to section III-A we have established the proposed hardware infrastructure in above figures.

2. Software side implementation:
Install the blynk on your mobile device.

Fig 14: Blynk widget box.

VII. CONCLUSION

This paper describes the general overview of IoT and provides very basic and easy setup to start working in IoT in just few minutes using blynk server.
REFERENCES


