Journal of Engineering Sciences

Vol 12, Issue 9, Sep/2021
ISSN NO:0377-9254

APPLICATION DEPLOYMENT FOR USER ESSENTIALS

K. Hari Krishna[1], N.Sahiti[2], M.Vasudha Reddy[3], E.Vineeth Kumar Varma[4], M.Amith Varma[5]

[1]Assistant Professor,[2345]UG Students

Electronics and Communication Engineering

SATYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT VIZIANAGARAM

Abstract

Now a day's developing apps can be fun and is potentially lucrative, but it is also quickly becoming a core skill in the information technology field. Businesses are increasingly looking to mobile apps to enhance their relationships with their customers and improve their internal processes. They need individuals skilled in developing the mobile apps that support these initiatives. This invention is intended as an introduction to mobile development for both Android and iOS. Although this provides everything you need to know to begin creating apps on both platforms, it is not intended to be a comprehensive work on the subject. Mobile development introduces issues and concerns not associated with traditional development, but at its core requires the ability to program. MIT App Inventor is an online platform designed to teach computational thinking concepts through development of mobile applications. Students create applications by dragging and dropping components into a design view and using a visual blocks language to program application behavior. MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing. As the part of this we came up with the idea of developing an app which can be used for controlling or handling devices. This MIT app inventor help in building that apps and that helps in a proper understanding of how the MIT app inventor platform works for the user.

Keywords:
Computational thinking · Computational action · Educational technology · Programming languages · Block-based programming · Mobile learning

I. INTRODUCTION

Humans Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this work. The smartphone is the ultimate personal computer. Mobile devices are always with us and have become an essential part of personal productivity and lifestyle needs. We use them for messaging, social media, Google searches, games, picture taking, and, of course, phone calls.

The Android operating system powers most of the world's smartphones, bringing an extensive app catalog to these devices. According to Google, more than 1 billion active devices are currently running Android. Perhaps you have reached the point at which using mobile apps on your smartphone isn't enough—it is time to create one. You might just want to tinker and program a simple app, or maybe you have thought of a new concept that doesn't exist yet.

Whatever the case, MIT App Inventor is an excellent place to start. App Inventor is an easy-to-use tool for building both simple and complex Android applications. The apps can easily be ported to your phone, shared with others, or even sent to the Google Play Store for distribution to all Android devices worldwide. For those looking to learn a programming language, MIT App Inventor can serve as an excellent bridge to acquiring more complex coding skills. Instead of presenting new users with frustrating messages and unfamiliar commands, App Inventor has a visually friendly interface that uses the methods of dragging, dropping, and connecting puzzle pieces to program applications

Even though App Inventor does not require using code, it builds on the same kinds of principles that successful programmers need to write good applications. Whether you go no further with programming or you use App Inventor to launch a new career, using App Inventor can be a highly engaging and challenging experience. Additionally, the open and flexible nature of Android makes it the perfect place to start.

II. BACK GROUND WORK

In these days we are clearly observe that everyone needs secure their personal data. No one is ready to share their data. And, everyone wants to build an app without having complete knowledge of programming's here we are using MIT app inventor. In this we can easily code by using block editor. No one can access our personal data and, we can easily operate the app and if any problem occurs we can find out it and solve easily.

In earlier methods, the installation might be difficult for a user without proper knowledge.There are many other apps for Operating home switches, but it may cost and time during installation. Complexity in technology is also includes. Not all the systems are compatible with each other. One system is different with another, it is difficult to connect. Even though the price of automation systems has become much more affordable in recent years, the cost to purchase and install a device can still add up. To truly leverage the convenience of home controlling, you may need to invest in centralized platform technology to control all systems and devices from one location. These apps access user's personal data.

III. PROPOSED METHODOLOGY

Most smart phone users consume technology without being able to produce it, even though local problems can often be solved with mobile devices. How then might they learn to leverage smart phone capabilities to solve real-world, everyday problems? MIT App Inventor is designed to democratize this technology and is used as a tool for learning computational thinking in a variety of educational contexts, teaching people to build apps to solve problems in their communities,sowe have come up with an idea of developing an application for user requirements.

In our proposed method we are going to develop an application in which user can easily access and can perform tasks i.e user can control devices using mobile. We had used an open source platform MIT app inventor. By using this we can create our own app for Operating home switches. By using component designer and block editor the app creation is done. We can maintain it so easily. The compatibility is also included. We can use multiple requests of users without affecting the system. And we are also using the Arduino IDE for coding the Arduino board.

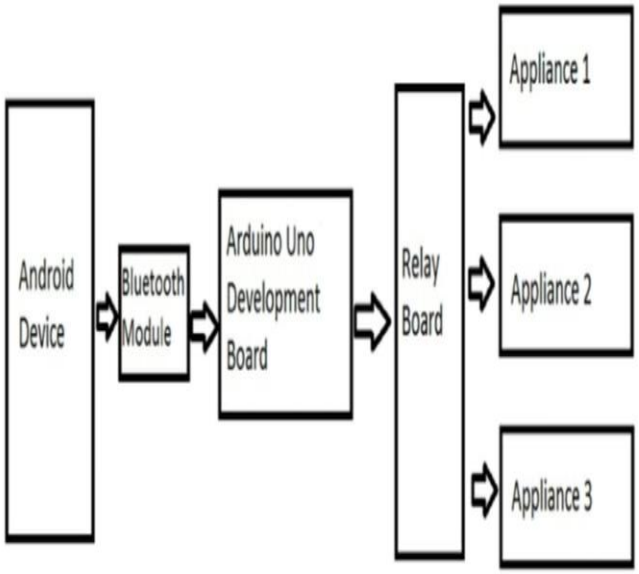


fig.2 Block diagram of the system

IV. SYSTEM IMPLEMENTATION

Our system is implemented by following hardware components and software tools:

General Power Supply:

The power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

***Transformer* :**

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected

to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.

***Bridge rectifier* :**

Bridge rectifier is used to maintain the proper DC polarity at the input to the circuit, irrespective of telephone line polarity. It comprises of four diodes connected to form a bridge. It uses the entire AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting.

MIT App Inventor:

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create application software(apps) for two operating systems (OS): Android, and iOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software released under dual licensing: a Creative Commons Attribution ShareAlike 3.0 Unported license, and an Apache License 2.0 for the source code. It uses a graphical user interface (GUI) very similar to the programming languages Scratch (programming language) and the StarLogo, which allows users to drag and drop visual objects to create an application that can run on android devices, while a App-Inventor Companion (The program that allows the app to run and debug on) that works on iOS running devices are still under development. In creating App Inventor, Google drew upon significant prior research in educational computing, and work done within Google on online development environments. App Inventor and the other projects are based on and informed by constructionist learning theories, which emphasize that programming can be a vehicle for engaging powerful ideas through active learning.

Component Designer

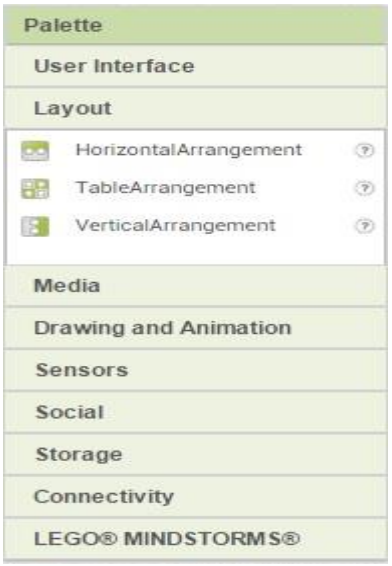
Component designer window having different components to use while building the app. The components are dragged to phone screen and the user can change the properties of each and every component. In component designer there are three main blocks 1. Palette 2. Viewer 3. Properties. The above fig shows the component designer window.

Palette

Palette is located at the left of the page. It contains all those components which can be added to our project. Components are classified in various categories like User Interface components, Media Components, Storage Components, and Social Components etc.

- **User Interface :** User interface includes all the visible components that you see on screens of any apps. It includes buttons, text boxes, labels, images etc.
- **Layout :** Layout gives us the power to arrange the components in anorder like horizontal or vertical.
- **Media :** Camera, Music Player, Voice Recorder, Video Player all comes under media category. In app inventor we have ready components available for all these which you can directly add to your projects.

- **Sensors :** Sensors play important roles specially during gaming applications to control the user actions. App inventor gives accessibility to use various sensors like Accelerometer Sensor, Barcode Scanner, Location Sensor, Orientation Sensor and Proximity Sensor.



Viewer

Viewer is the actual representation of how our app will look like on the phone after installation. It is on the middle of the page. As you drag any components to the viewer it will be automatically added to your project. If the physical property of a component is not visible, then it will not be visible in viewer but will be placed under viewer with non-visible tag.

Components

Components tab lists all those components which you add to your project. Those all are listed under it in tree structure. You can also perform nesting like you will be using horizontal arrangement and then various buttons it will show buttons listed under horizontal arrangement.

Properties

Properties tab allows you to set different properties for the components like height, width, colour, text, background etc. When you add any component to your project and select it on screen its respective properties are shown under the properties tab. These properties can be configured by you as per the requirement.

Block Editor

Fig Block Editor

In block editor we have already built behaviours for all the components available in the palette and we can directly pick them and use as per the need. These behaviours are independent java bundles which are presented in easy to use socket form so that you can group 3-4 behaviours all together to make a complex activity. It is that much simple so that an English reader can better under what will happen when this block will be executed.

SCAN QR CODE

ANDROID PHONE

This is the app developed by us using the MIT app inventor platform

Arduino

Arduino is open source hardware.Most Arduino boards consist of an Atmel 8-bit AVR r (ATmega8, ATmega168, Tmega328,ATmega1280, ATmega2560) with varying amounts of flash memory, pins, and features. The 32-bit Arduino due based on the Atmel SAM3X8E was introduced in 2012. The boards use single or double-row pins or female headers that facilitate connections for programming and incorporation into other circuits. These may connect with add-on modules termed *shields*. Multiple and possibly stacked shields may be individually addressable via an I2C serial bus. Most boards include a 5 V linear regulator r and a 16 MHz crystal oscillator or ceramic resonator . Some designs, such as the Lily Pad, run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions.

PIN DIAGRAM

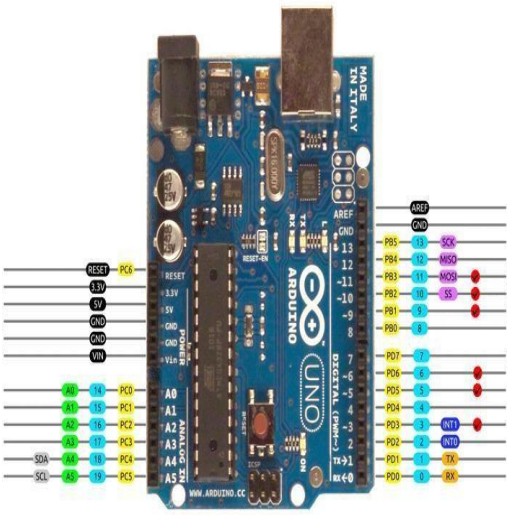
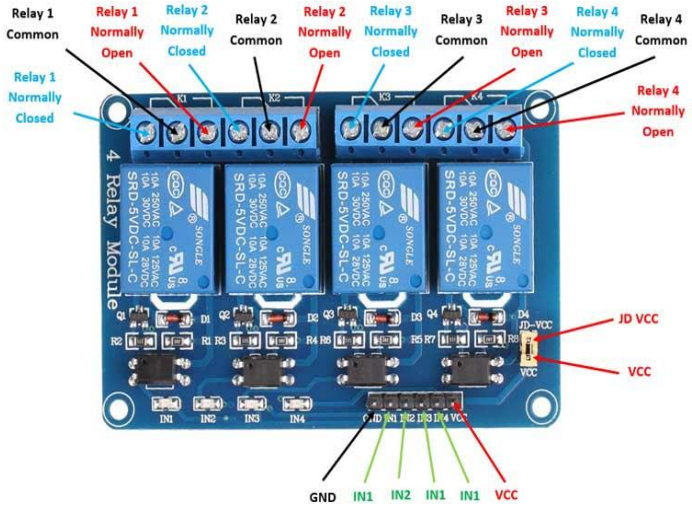


Fig Arduino UNO

4-CHANNEL RELAY MODULE:

The **four-channel relay module** contains four 5V relays and the associated switching and isolating components, which makes interfacing with a microcontroller or sensor easy with minimum components and connections. The contacts on each relay are specified for



250VAC and 30VDC and 10A in each case, as marked on the body of the relays.

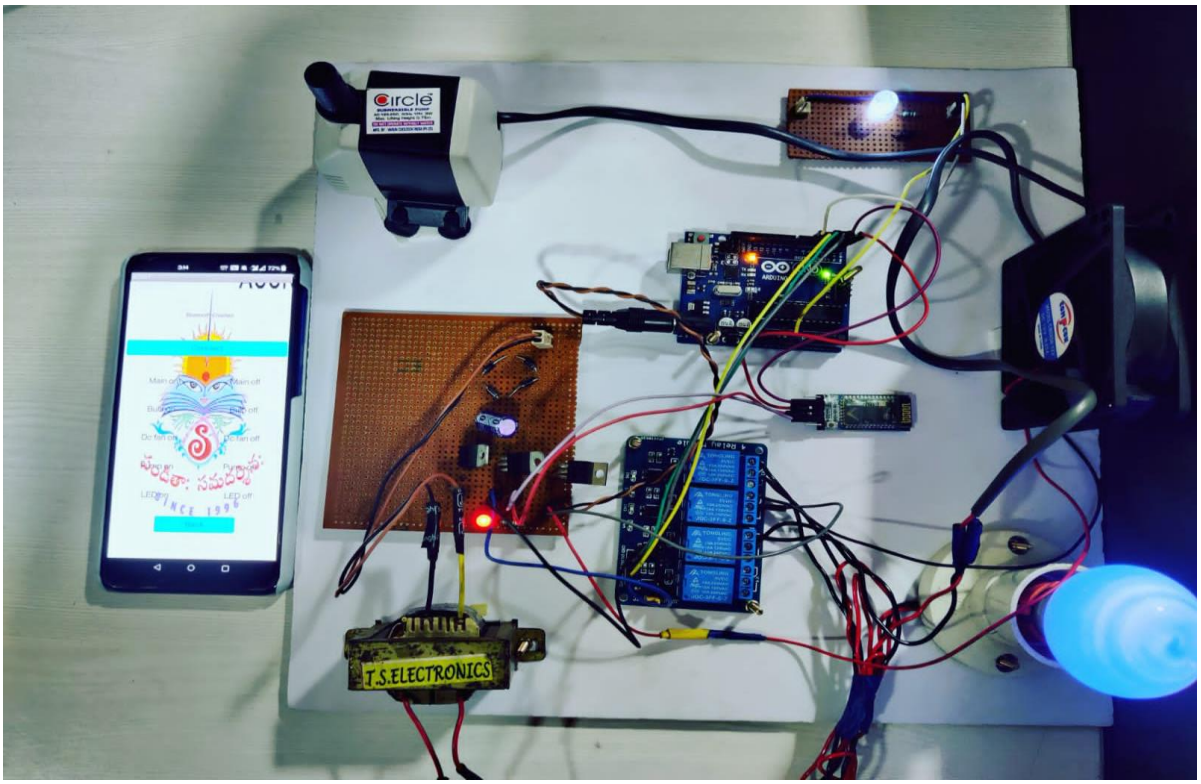
BLUETOOTH MODULE:

In communication, the Bluetooth wireless technology has become very popular and it is one of the fastest growing fields in the wireless technology. HC 05 Bluetooth is a wireless communication protocol; it is used in two devices as a sending and receiving the information. The Bluetooth is free to use in the wireless communication protocol as the range of the Bluetooth is less than the other **wireless communication protocols like Wi-Fi and Zigbee**. The Bluetooth operates at the frequency of the 2.41 GHz and used in many small ranges of applications.

Journal of Engineering Sciences
Vol 12, Issue 9, Sep/2021
ISSN NO:0377-9254

RESULTS

The real time view of our project as shown in figure. We are using four loads and those are controlled by the mobile app. We can ON or OFF any load in the circuit through mobile app which is designed by using MIT APP INVENTOR. In below figure all loads are in ON condition, because here we are using main on switch, i.e., all loads should in ON condition.



FUTURE SCOPE

Things are changes when time passes, so, it is better to follow latest guidelines of Android for App Inventor. Because App Inventor apps are even working on first releases of Android which is only 5% of people using.So I think dropping support for old versions and focusing for latest versions is better for App Inventor, so App Inventor can add new features to the platform like full implemented Material Design, because newly created apps on other platforms are already supports Material Design. Or Background services, or better Adaptive Icons support. I know other App Inventor forks/clones working for latest features but seeing that for App Inventor itself would be better. Maybe these things will require a major redesign/remade for App Inventor, but I think it will grant a lot of benefits for MIT Team about adding new features easily and users who are using App Inventor for creating apps.

REFERENCES

1. *Hardesty, Larry (August 19, 2010).* "The MIT roots of Google's new software". *MIT News Office.*
2. "On the Shoulders of Giants!". *Archived from* the original *on August 11, 2010*. Retrieved August 10, 2010.
3. "The FirebaseDB Component (Experimental)". *ai2.appinventor.mit.edu*. Retrieved 2019-02-14.
4. *Wolber, David;* Abelson, Hal*;* Spertus, Ellen*; Looney, Liz (May 2011),* App Inventor for Android: Create Your Own Android Apps, *O'Reilly*, ISBN 978-1-4493-9748-7
5. "App Inventor @ MIT".
6. *Clark, Andrew (December 30, 2013),* "App Inventor launches second iteration", *MIT News*, retrieved 7 July 2019
7. App Inventor Classic, *December 3, 2013*
8. "MIT AI2 Companion". *May 25, 2019*. Retrieved 8 July 2019