Android App for Industrial Automation

Mayan Kumar
Student, Bachelor of engineering
Electronics and Telecommunication
Priyadarshini college of engineering
Nagpur, India

Nikita Wadekar
Student, Bachelor of engineering
Electronics and Telecommunication
Priyadarshini college of engineering
Nagpur, India

Sakshi Sinha
Student, Bachelor of engineering
Electronics and Telecommunication
Priyadarshini college of engineering
Nagpur, India

DhanashriRaut
Student, Bachelor of engineering
Electronics and Telecommunication
Priyadarshini college of engineering
Nagpur, India

Dr.V.K. Taksande
HOD, Bachelor of engineering
Electronics and Telecommunication
Priyadarshini college of engineering
Nagpur, India

Abstract– With the technology a boon to the world as it leads to advancement of industries, infrastructures etc. around the globe. In the era of the new upcoming technologies there is a need to reduce the human action for the convenience of every person. Keeping this in mind, we have chosen to design an Android App for Industrial Automation. The objective of this project is to design an android app using MIT App Inventor for operating industrial machinery. In order to achieve this we have made an app (accessible from mobile) using MIT App Inventor wherein all the Industrial Automation equipment can be easily operated and controlled using this app.The app is develop on MIT app inventor and to test this app we have designed the prototype of hardware (containing ESP module & Relay) which takes the signal from this app using the Wi-Fi network(IP Address)which is a Wi-Fi router and can work accordingly. Also you can send direct messages using the app to the desired user. The users can login in and out the app and the data filled in by the user can be stored using the Firebase software by Google.

***Key Words*: MIT App Inventor, ESP Module, Relay, Firebase.**

1. INTRODUCTION

In our daily activities, we prefer to use an app instead of browser-based access. For example, it's possible to connect to a search site, type in the word "Maps," and open up the corresponding web-based application to find directions. Instead, most people have a navigation app installed on their phone, and they simply click on it to find their way. Apps are preferred over browser-based access because they are faster to access, quicker to open, work better with low latency connections, and are faster and easier for interactions. This is because they are custom designed to perform only one task and do it well, as compared to a web-based connection which must stand ready to provide access an enormous range of applications spanning the internet.

For industrial applications, apps have an even more pronounced advantage over browser-based access because speed of interaction is generally more important in the industrial world than in the consumer arena.

2. LITERATURE REVIEW

As per survey machines typically found in a plant or a facility are usually controlled by a PLC, with a local HMI providing interactions with the PLC and local operator interface. For plants or facilities with many machines, it often doesn't make sense for each to have its own full-featured HMI because this adds costs, takes up space, and increases the need for spares and maintenance.

Instead, smaller machines and those not requiring constant operator interaction can be supplied with just a few pushbuttons and lights, with all other HMI functionality provided by an app running on a smartphone or tablet [1-2]. For machines requiring a full-featured and permanently-mounted HMI, an app can be used to provide remote access, whether this access is required locally to the plant, or many miles away.

In either case, the app needs to provide functionality similar to an HMI, with the capabilities listed in the Table.

Table, Required App Functionality for PLC Interface and HMI

i) Runs on any iOS or Android mobile device

ii) Provides a range of wireless connectivity options

iii) Updates PLC firmware

iv) Downloads and uploads PLC programs

v) Includes role-based access

vi) Monitors PLC parameters

vii) Changes PLC setpoints and other values

viii) Extracts stored data from the PLC

ix) Sends data via text or email

3. Hardware And Software Requirement

3.1 Hardware

ESP8266
ESP8266 is a Wi-Fi microchip, with full of TCP/IP stack and microcontroller capability, developed by ESPRESSIF SYSTEMS in shanghai china.

Fig(A): ESP-12

Esp8266 is a low cost, small module and has the ability to perform Wi-Fi related activities hence it is widely used as Wi-Fi module. This small module allows microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections. It is used for establishing a wireless network connection.

ii)NodeMCU

NodeMCU is a firmware on ESP8266 and basically a System on Chip (SoC), and hardware is based on ESP-12 module.
The firmware uses the Lua scripting language. Supports for the 32-bit ESP32 has been implemented.
It is a low-cost open source IoT platform. It is easy to work on this inexpensive SoC. It has analog and digital pins, which are used to interface sensors and get the data over internet.
It has all the ESP-12 pins, it also includes a micro USB connector with onboard programmer IC, voltage regulator, reset and program buttons and LEDs. In this project we used NodeMCU to blink LEDs.

iii)Relay

A Relay is made up of electromagnets which in general used as a switch. Relay means the act of passing something from one thing to another. Signal received from one side of the device controls the switching operation on the other side, so Relay is a switch which controls (open and close) circuits electromechanically.

Fig(B)Relay

The main operation of this device is to make or break contact with the help of a signal without any human involvement in order to switch it ON or OFF. We used Relay to blink the LEDs.

It is mainly used to control a high-powered circuit using a low power signal. The Relay is used for transmitting and receiving the information where the input signals used to be either 1 or 0, these change in signals were mechanically noted in terms of ON and OFF of a light bulb or a beep sound[3], it means those pulses of 1s and 0s are converted as mechanical ON and OFF using electromagnets.

(iv)Working

There are several IoT Projects based on ESP8266 Wi-Fi module but Home Automation i.e.

- Sends data to the cloud

controlling different electrical appliances through Wi-Fi (or Internet) has always been a trending and in demand project. So, in this project, A Simple Relay Module is connected to the ESP8266 Wi-Fi module and Relay is controlled using ESP8266.For this project, we created an Android App which can control the Relay. In order to develop the Android App., we used MIT App Inventor[5-6]. The .apk file is the one to install onto your phone. The .aia file is uploaded to MIT app inventor to change it.

3.2 SOFTWARE:

MIT App Inventor

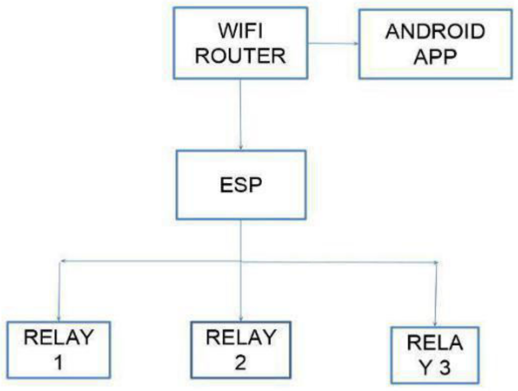
The MIT App Inventor user interface includes two main editors: the design editor and blocks editor. The design editor, or designer, is a drag and drop interface to layout the elements of the applications user interface. The blocks editor is an environment in which app inventors can visually layout the logic of their apps using color-coded blocks connect together like puzzle pieces to describe the program.

In MIT App Inventor, users code application behaviour using a block-based programming language. There are two types of blocks in App inventor: built-in blocks and component blocks. Developers use component blocks (properties, method and events) to respond to system and user events, interact with device hardware.

Firebase Real-Time Database

Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Fire-base's cloud. The company provides client libraries

That enable integration with **Android iOS JavaScript Java Objective C, Swift and Node.js applications**. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as **AngularJS, React, Ember.js and Backbone.js.** The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real time database can secure their data by using the company's server-side-enforced security rules.



Fig(C): Block Diagram

4. APPLICATIONS:

- This app can be used in home or industry to control the AC devices from mobile phone.
- It is used in educational institutes to power on and off the lights or fans in the classrooms wirelessly sitting in one place.
- Can be used to control temperature, pressure and power of big machines in industries.
- Also used to send text messages to the user at the other end who is keeping a watch on the machinery.

5. FUTURE SCOPE

- Can be used to control temperature, pressure and power of big machines in industries
- Images and videos can be shared to the users who have logged in.
- Locations can be searched in maps and live locations can be send to the user

6. CONCLUSIONS

The ultimate aim of the paper was to design an Android App for Industrial Automation. Basically, developing an Android app using MIT app inventor via block programming and enabling various functions. This provides small scale industrialists to synchronize their machinery to the app and hence allows the admin to supervise the production and working via the app.

7. REFERENCES

[1] M. Chan, D. Esteve, C. Escriba, E. Campo, “A review of smart homes—Present state and future challenges”, Computer methods and programs in biomedicine, Elsevier, 9I, pp. 55-81, 2008.

[2] J. Haartsen, "BLUETOOTH—The universal radio interface for ad hoc, wireless connectivity", Ericsson Review No. 3, pp. 110-117, 1998.

[3] Why Android?
http://www.android.com/about/, last seen on

October 2013.

[4] R. Llamas, R Reith, M. Shiere, “Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC” 7

August 2013, IDC Press Release,
http://www.idc.com/getdoc.jsp?containerId=prUS24257413.

[5] N. Sriskanthan, F. Tan, A. Karande, "Bluetooth based home automation system", Microprocessors and Microsystems, Elsevier, 26, pp. 281-289, 2002.

[6] S. Panth, M. Jivani, “Designing Home Automation system (HAS) using Java ME for Mobile Phone”, International Journal of Electronics