IJCSN International Journal of Computer Science and Network, Volume 5, Issue 3, June 2016
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 1.02

Home Automation Using the Concept of IoT

[1] Tejal Deshpande, [2] Nitin Ahire

[1, 2] Xavier Institute of Engineering, Mahim, Mumbai, India

Abstract - With the rising power of technology, we are able to accomplish things at a much quicker rate. We have at the touch of a button access to large amounts of information due to the capability of computers and the Internet. Not only has technology given us more information, but it also has given us the ability to communicate, organize, and manage our time. The Internet of Things (IoT) is a vision in which the Internet extends into our everyday lives through a wireless network of uniquely identifiable objects. In this paper we present on how to create and build an IoT enabled device and control the said device wirelessly using our smart phones.

Keywords - *Arduino Mega, Blynk app, Internet of Things, Wi-Fi module.*

1. Introduction

The Internet of Things (IoT) is a novel paradigm that is becoming popular with research and industries. The basic idea is that IoT will connect objects around us (electronic and electrical) to provide seamless communication and contextual services provided by them. The objective of this paper is to build an IoT enabled wall outlet using Arduino Mega and ESP Wi-Fi module and control it wirelessly using the smart phone app Blynk.

The Internet of Things (IoT) is huge in several ways. The forces that are driving it and the benefits that are motivating it are increasingly numerous, as more and more organizations, industries, and technologists catch the IoT bug. The number of connected devices on the IoT network will be huge. One estimate says that the number will be nearly 40 billion, which is approximately 30 devices for each and every active social network user in the world. The economic impact and benefits of the IoT will be huge. Analysts define the IoT in terms of connected everyday objects; the nature of the connection remains to be determined. A two-way connection by means of the Internet Protocol constitutes the ideal case, but the originators of the IoT concept appear to have emphasized a simpler model of RFID query and response. The IoT will be inextricable from sensor networks that monitor things but do not control things. Both connected everyday objects and sensor Networks leverage a common set of technological advances toward miniature, power-efficient sensing, processing, and wireless communication.

2. Hardware Description

2.1 Arduino Mega ADK

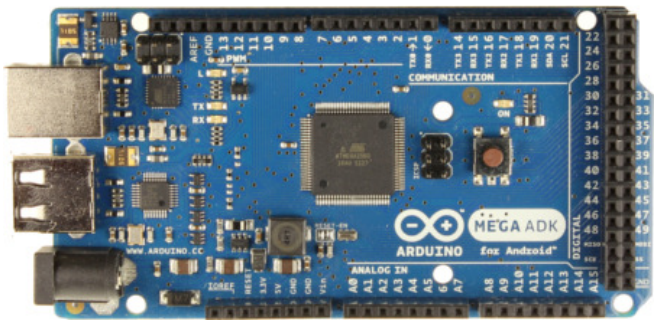

Fig.1. Arduino Mega ADK

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 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 ATMega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Arduino Mega 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 5V,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.

2.2 ESP8266 Wi-Fi Module

The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and microcontroller capability produced by Shanghai based Chinese manufacturer, Espressif. The chip first came to the attention of western makers in August 2014 with the ESP-01 module. This small board allows microcontrollers to connect to a Wi-Fi network and make TCP/IP connections using Hayes-style commands. However, at the time there was almost no English-language documentation on the chip and the commands it accepted. The very low price and the fact that there were very little external components on the module which suggests that it could eventually be very inexpensive in volume, attracted many hackers to explore the module, chip, and the software on it, as well as to translate the Chinese documentation.

Fig.2. ESP8266 module

The ESP8266-01 is the smallest ESP8266 module and has 8 pins only. Of these VCC, GND, RST (reset) and CH_PD (chip select) are not I/O pins but are needed for the operation of the module. This leaves GPIO0, GPIO2, TX and RX available as possible I/O pins, but even these have pre-assigned functions. The GPIO0 and GPIO2 determine what mode the module starts up in and the TX/RX pins are used to program the module for Serial I/O, commonly used for debugging. GPIO0 and GPIO2 need to have pull-up resistors connected to ensure the module starts up correctly.

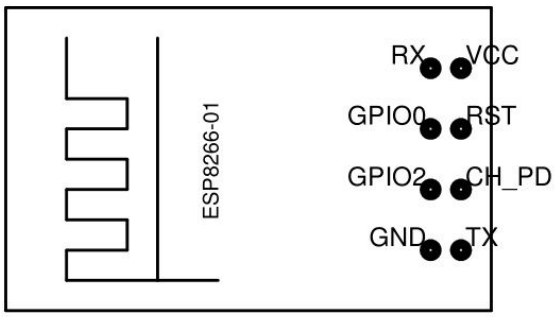

Fig.3. Pin Configuration

2.3 TFT LCD Screen

A 'thin-film-transistor liquid-crystal display' (TFT LCD) is a variant of a liquid-crystal display (LCD) that uses thin-film transistor (TFT) technology to improve image qualities such as addressability and contrast.

The Arduino 2.4" TFT LCD Touch shield is an Arduino UNO/ Mega compatible multicolored TFT display with touch-screen and SD card socket. It is available in an Arduino shield compatible pin out for the attachment. The TFT driver is based on ILI9335 with 8bit data and 4bit control interface. The Arduino 2.4" TFT LCD Touch shield can work with both 3.3V and 5V, so it can display on Chip kit UNO32 and Simple cortex as well.

Fig.4. TFT LCD Screen

3. Software Description

3.1 Arduino IDE

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

3.2 NodeMCU Firmware

NodeMCU is an open source IoT platform. It uses the Lua scripting language. It is based on the eLua project, and built on the ESP8266 SDK 0.9.5. It uses many open

source projects, such as lua-cjson, and spiffs. It includes firmware which runs on the ESP8266 Wi-Fi SoC, and hardware which is based on the ESP-12 module.

NodeMCU was created shortly after the ESP8266 came out. In December 30, 2013, Espressif systems began production of the ESP8266.The ESP8266 is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications. NodeMCU started in 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 1.0. 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 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 project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

3.3 Blynk App

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**.**

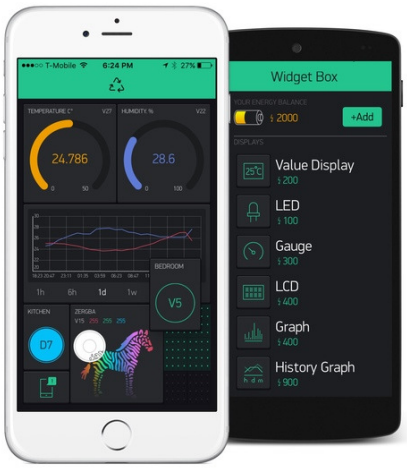

Fig.5. Blynk App

3.4 Blynk Server

Blynk Server is an Open-Source Netty based Java server, responsible for forwarding messages between Blynk mobile application and various microcontroller boards (i.e. Arduino, Raspberry Pi. etc). Blynk Cloud is software written on Java using plain TCP/IP sockets and running on our server. Blynk iOS and Android apps connect to Blynk Cloud by default. Access is free for every Blynk user. To run Blynk Server, all we need is Java Runtime Environment.

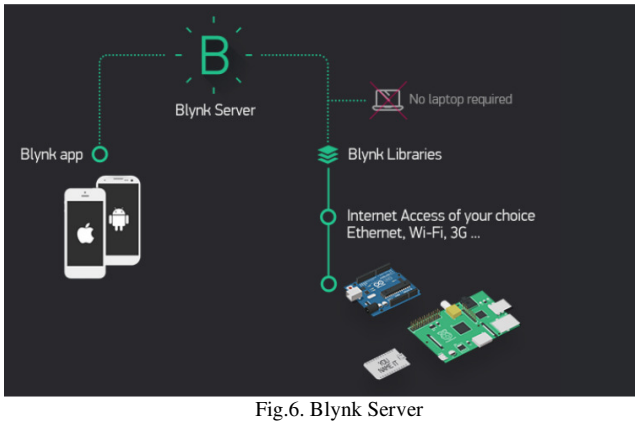

Fig.6. Blynk Server

4. System Description

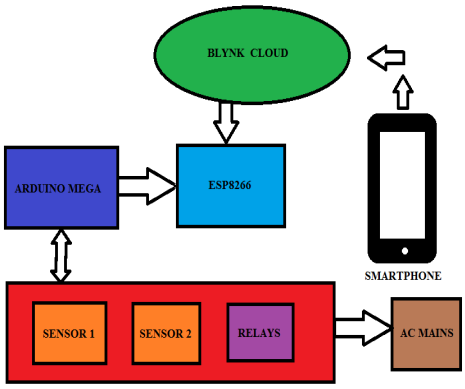

Fig.7. Block Diagram

The TFT LCD acts as a wall outlet: where one would usually find switches. We have a touch screen interface which includes buttons for lights, mood lighting, and A/C components through relays. These components will be connected to the Blynk cloud server through the ESP8266 Wi-Fi module.

The Blynk cloud server not only is connected to the components connected on the Arduino Mega through the ESP8266, but also to the cell phones containing 'Blynk' mobile phone application. Through the mobile phone

application all the components connected to the Arduino Mega can also be controlled.

Blynk mobile phone application allows us to add multiple components onto the Arduino Mega without making any changes to the code for the application.

As this project uses a cloud server, all of these components can be can be controlled from any place on the earth

5. Experimental Results

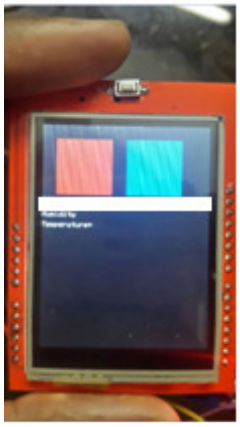

Fig.8 TFT Touch screen

In Fig 8 the TFT LCD is activated and we have designed buttons to remain on the screen. Pressing a button for example, the red button, we move to the next page, where we have the controls for a relay, which can be seen in the next image

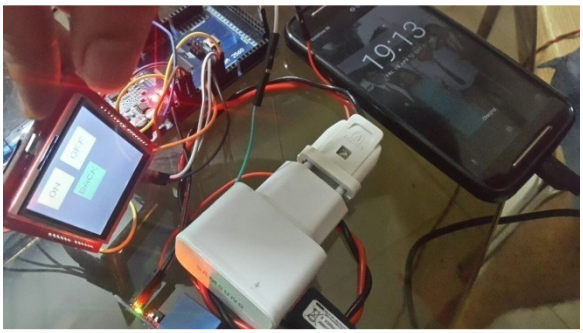

Fig.9. Cell phone in charging mode

The AC current is first supplied to the relay module via its receiving pin which initiates the blinking of the relay. The same AC current is given to a mobile charger. The relay light glows green indicating that it has been activated and hence the cell phone is in charging mode as shown in Fig.9.

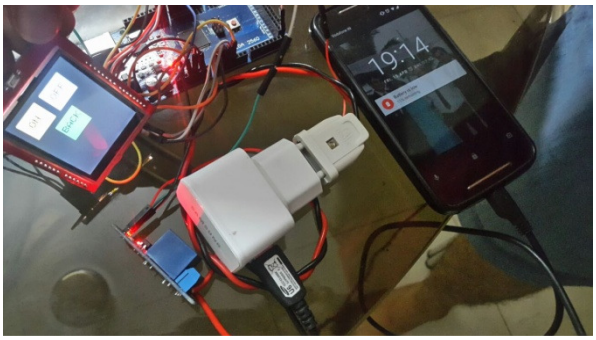

Fig.10. Cell phone not charging

When the relay light on the module glows red, it means that the relay is not in active mode. Hence the cell phone is not charging as shown in Fig.10.

The relay modules can also be turned on or off manually by using the buttons on the TFT LCD screen.

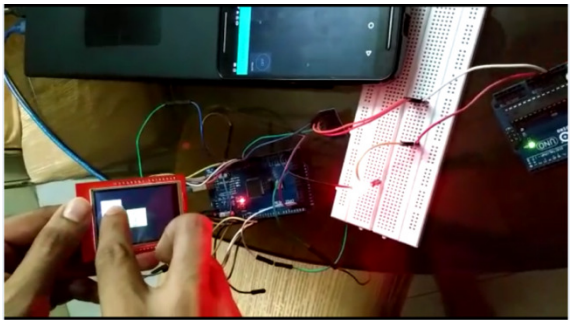

Fig.11. Using Blynk app

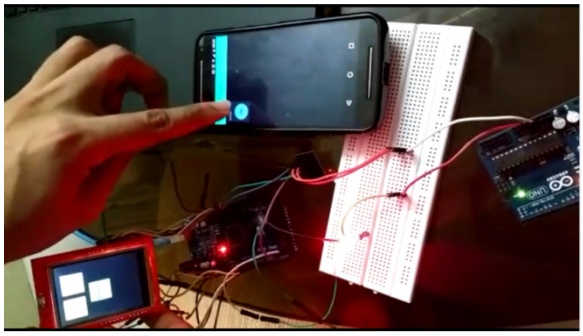

Fig.12.Relay control using Blynk app

Using the Blynk app and touch screen we can control all elements as show in Fig.11 and Fig.12.

6. Limitations

If we send more than 100 values per second we may cause Flood Error and our hardware will be automatically disconnected from the server. Also be careful while

sending a lot of Blynk.virtualWrite commands as most hardware is not very powerful (like ESP8266) so it may not handle many requests.

7. Conclusion

The paper has been experimentally proven to work successfully. We can control the parameters of the AC components using the Blynk app or manually using the touch screen. The designed system not only monitors the sensor data, like temperature, light, mood lighting but also actuates a process according to the requirement. It also stores the sensor parameters in the cloud in a timely manner. This will help the user to analyze the condition of various parameters in home anytime anywhere.

8. Future Scope

Using this paper as a framework, it will give immediate access to information about the physical world and the objects, leading to innovative services and solutions and leading to an increase in efficiency and productivity. In the near future the Internet and wireless technologies will connect different sources of information such as sensors, mobile phones and cars in an ever tighter manner. The number of devices which connect to the Internet is – seemingly exponentially – increasing. These billions of components produce consume and process information in different environments such as logistic applications, factories and airports as well as in the work and everyday lives of people.

References

[1] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, Michele Zorzi, "Internet of Things for Smart Cities," in IEEE Internet Of Things Journal, Vol.1, No.1, February 2014.

[2] Sean Dieter, Tebje Kelly, Nagender Kumar Suryadevara, and Subhas Chandra Mukhopadhyay, "Towards the Implementation of IoT for Environmental Condition Monitoring in Home" in IEEE Sensors Journal, Vol.13, No.10, 2013.

[3] Mohammad Aazam, Pham Phuoc Hung, Eui-Nam Huh, "Cloud of Things: Integrating Internet of Things with Cloud Computing and the Issues Involved" in Proceedings of International Bhurban Conference on Applied Sciences & Technology Islamabad, Pakistan, January 14 – 18, 2014.

[4] Gopinath Marrapan.(2015).Using ESP8266 to power lights.[Online].Available:http://www.instructables.com /id/An-inexpensive-IoT-enabler-using-ESP8266/

[5] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, Marimuthu Palaniswamia, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," in Department of Electrical and Electronic Engineering, University of Melbourne, *Australia (2011).*

[6] Internet of Things. [Online].Available: https://en.wikipedia.org/wiki/Internet_of_Things

[7] Gopinath Marrapan. (2015).An inexpensive IoT enabler using ESP8266. [Online].Available: http://www.instructables.com/id/An-inexpensive-IoT-enabler-using- ESP8266

.