CONSTRUCTION OF ANDROID AND ARDUINO FOR HOME APPLIANCES SWITCHING SYSTEM

Si Thu
Physics Department,
Lashio University, Myanmar

Hlaing Htake Khaung Tin
Research and Development Department
University of Computer Studies, Yangon, Myanmar

Abstract— **Android and arduino based switching system uses bluetooth module to connect each other. The RN-41 (from Sparkfun electronics) bluetooth module receives the data from android phone or android operating system based tablet to control the system remotely. After getting the command data, bluetooth module sends the data to the arduino by the way of serial communication. Arduino receives the data from bluetooth module and then it controls the external circuit. The programming in this research project used is C++ language. Some types of microcontrollers, development boards, some operating systems of smart phones and general knowledge about android systems are introduced in introduction section. After then background theories and using devices are explained. And then design and construction of the Android and arduino based switching system is mentioned.**

Keywords— **Remote Switching, Bluetooth, Home Appliances, Android, Arduino**

I. INTRODUCTION

Smart phone controlled home applications are being populared and smart phones, tablets are well known and are popular devices nowadays. Recently almost all of the phones used Java based operating system. From then, electronic device production companies tried to use their own operating systems in their smart phones. For example, iPhone used iOS operating system. Sam Sung used Android operating system. Huawei and many other China made phones also used Android operating system. Black berry used Black berry operating system and Nokia used Symbian operating system. Some smart phones used windows operation system. Even though most of the brands used their own operating systems, Android operating systems and applications are very common and well known. There are millions and tons of Android applications are in the market currently. We used the android application in this project. The modern electronics introduced 32 bit microcontrollers after developing 8 bit and 16 bits microcontrollers. There are many types of microcontrollers by various electronics companies. For instance, 8051, ARM, Atmel, AVR(8bit), AVR(32bit), Microchip technology PIC(8bit, 16bit, 32bit), STMicroelectronics [STM8(8bit),

ST10(16bit) and STM32(32bit)], etc. Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target microcontrollers. These languages are either designed especially for the purpose, or version of general purpose language such as C programming language. Compilers for general purpose languages will typically have some restrictions as well as enhancements to better support the unique characteristics of microcontrollers.

Interpreter firmware is also available for some microcontrollers. For example BASIC programming language. After compiling, computer gives out hexa decimal file output. Microcontroller ICs needed to put that haxa file via programmer board. For more convenient, today's modern electronics world is being used development board/ evaluation board instead of microcontroller ICs only. A Development Board is an amalgamation of hardware and software. It is nothing but a printed circuit board containing a microcontroller or microprocessor with other peripherals as hardware and support packages in the form of tool chains, operating system to make up the software part. Evaluation boards are same as Development Boards however apart from nomenclature they differ in power, features and cost. These are used mainly to evaluate how better the particular solution using the MPU or MCU chip on board is. Most people opt for these in initial stages and then move towards the proper Development Boards having lots of peripherals and feature loaded.

From the point of very small computer, among programmers, yet another is an idiomatic qualifier in the name of a computer program, organization, or event that is confessedly unoriginal. These are the boards which were made with the vision of providing small computer in size with enough power to let newbie and students learn computing in a new way : For example: **Android PC System**: This is new Android targeted board from VIA. APC runs a custom Android system, built for keyboard and mouse input. A basic selection of applications is preinstalled. Also included is a full set of consumer I/O ports, enabling APC to connect to your PC monitor or TV. Software Android 2.3 (PC System) and Chip VIA 800MHz Processor. **Raspberry Pi** is the newest, cheapest and advanced development board in market. It is the

result of collaboration at University of Cambridge's Computer Laboratory. By default the SD Card comes with Debian loaded in it to run directly on the board. Else all ARM linux can be tried.

II. BACKGROUND THEORY

Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. Pre-programmed into the on-board microcontroller chip is a boot loader that allows uploading programs into the microcontroller memory without needing a chip (device) programmer. Arduino boards can be purchased pre-assembled or as do-it-yourself kits. Hardware design information is available for those who would like to assemble an Arduino by hand. It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced. Arduino started in 2005 as a project for students at the Interaction Design Institute Ivrea in Ivrea, Italy. Massimo Banzi, one of the founders, taught at Ivrea.

A hardware thesis was contributed for a wiring design by Colombian student Hernando Barragan. After the wiring platform was complete, researchers worked to make it lighter, less expensive, and available to the open source community. An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components to facilitate programming and incorporation into other circuits.

An important aspect of the Arduino is the standard way that connectors are exposed, allowing the CPU board to be connected to a variety of interchangeable add-on modules known as *shields*. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I²C serial bus, allowing many shields to be stacked and used in parallel. Official Arduinos have used the megaAVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles.

Most boards include a 5 volt linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the onboard voltage regulator due to specific form-factor restrictions. An Arduino's microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDI FT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

III. DESIGN AND CONSTRUCTION

Android and Arduino based switching system essentially depends on the wireless system such as wi-fi or bluetooth wireless system. In this project, there are 4 remote switches used. Switch no. 1,2,3 for device 1, 2, 3 respectively. Switch no. 4 is for auxiliary use. Device 1, 2, 3 is represented by three 5W lamps red, yellow and blue. Even 12 ports or 12 devices can control, only four devices are used for demonstration only.

Android programming Arduino programming also need in such a remote switch system. There are many many android softwares available on the internet. Some are selling and some are free. In this project, readymade software is used to control the devices.
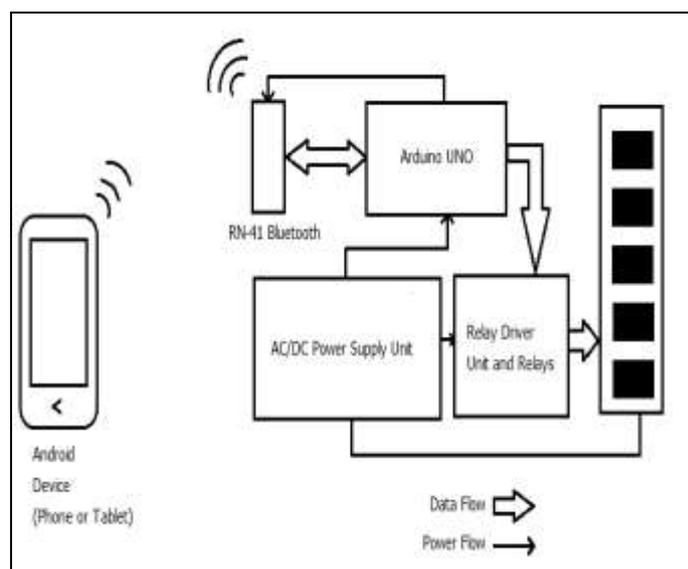

Figure 1. Block Diagram for Android and Arduino Based Switching System

The software is downloaded from Techbitar website. After putting softwares in Android based device (tablet or phone) and Arduino, the device is ready to use. First the project device must be powered. The RN-41 Bluetooth module is ready and waiting to connect devices by means of red LED blinking. Phone or tablet which use android operating system must open pre installed software whic called the name "Ardudroid".

Figure 2. ArduDroid Software logo and running in phone

RN-41 Bluetooth module and Telephone bluetooth module connect each other and make data transmitting and receiving. RN-41 Bluetooth also connect with Arduino from UART(Universal Asynchronous Transmitting and Receiving) line. On the other hand, RN-41 receives datan sends these data to Arduino. Arduino then drive relay drivers to control external AC devices. Relays control external devices . All external devices get AC power from the input line of transformer. DC power distributes to Arduino and relay driver unit and relays. RN-41 bluetooth module gets power from Arduino.

A. Circuit Operation

The Arduino operates at 5V since it is TTL Logic. Therefore 9V source is used in this circuit for Arduino. Arduino UNO contains the built in regulator for 5V supply. The center tap transformer which contain both 12V and 9V is used to distribute power to all unit of the device. All electromagnetic relays use 12V. So rectifier is designed to get output for 12V and 9V. It uses half-wave rectification method. Two capacitors C_1 and C_2 is 1000uF/25V. D_1 through D_8 are 1A, 1N4007 rectifier diodes. Transistors Q_1, Q_2, Q_3, Q_4 are C945, npn, general purpose transistors. When the power is turned ON, power distribution system delivers the DC power to Arduino for 9V and all electromagnetic relays for 12V. All relay contacts connect with 220V phase line and all under controlled AC devices connect one end of their leads to neutral lines. Another leads are connected to NO(Normally open) lead of relay. When the data are come from the air by serial communication, RN-41 Bluetooth module collects data and sends these data to Arduino what to do. After getting the command data from RN-41, Arduino switchs ON/OFF to transistors. Q_1, Q_2, Q_3, Q_4 get saturation condition if the Arduino gives enough voltage. In this case a transistor may be saturate if V_{BE} voltage greater than 0.7V because of a silicon transistor. In the moment of saturate region, transistor draws the collector current to ground via emitter lead. Electromagnetic relays are connected in collector. So if the

collector is in saturate region, the electromagnetic reay becomes energize and open the switch.
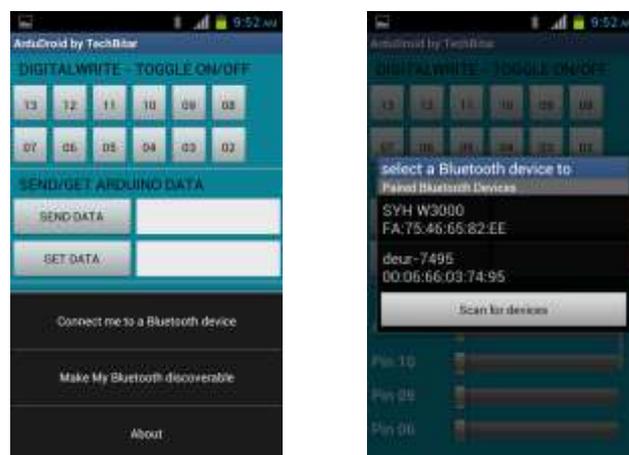
Figure 3. Right click menu of Ardudroid software and Scanning Device

RN-41 bluetooth module gets the power from Arduino. Transmitt and Receive leads of RN-41 are connected to Receive and Transmitt Leads of Arduino. On the other word, one Tx lead of a dcevice must be connected with one Rx lead of other. As soon as opening the ArduDroid software in phone or tablet, it will ask to open bluetooth connection on the phone or tablet. After then three line key or right key in phone must be pressed to connect and search device. Then press the button name "Connect me to a bluetooth device" and it will find out bluetooth devices around the phone. After scanning the bluetooth device, it will show the RN-41 bluetooth module, here in this case (deur-7495) with mac address. By pressing the deur-7495, phone will connect RN-41 Bluetooth module. Before connecting condition, the RN-41 will blink with red colour. After getting connection, the Gren LED will turn ON from the RN-41. If so, the device can use to control external electronic or electrical devices.
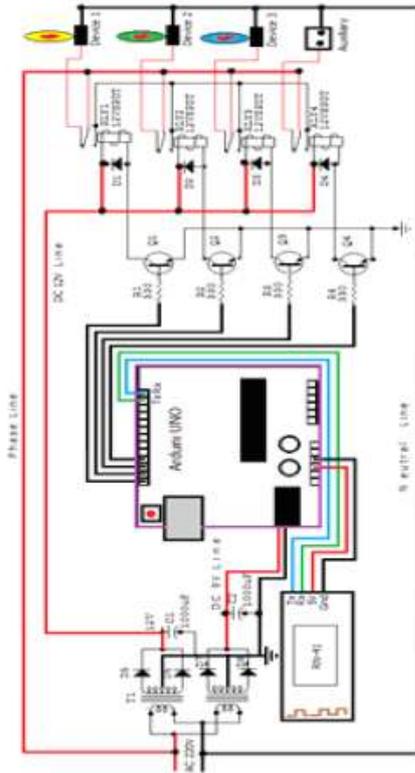
Figure 4.Complete circuit diagram

B. *Program Source Code*

```
#define START_CMD_CHAR '*'
#define END_CMD_CHAR '#'
#define DIV_CMD_CHAR '|'
#define CMD_DIGITALWRITE 10
#define CMD_ANALOGWRITE 11
#define CMD_TEXT 12
#define CMD_READ_ARDUDROID 13
#define MAX_COMMAND 20  // max command number
code. used for error checking.
#define MIN_COMMAND 10 // minimum command number
code. used for error checking.
#define IN_STRING_LENGHT 40
#define MAX_ANALOGWRITE 255
#define PIN_HIGH 3
#define PIN_LOW 2

String inText;
void setup() {
  Serial.begin(9600);
  Serial.println("ArduDroid 0.12 Alpha by TechBitar
(2013)");
  Serial.flush();
}
void loop()
```

```
{
  Serial.flush();
  int ard_command = 0;
  int pin_num = 0;
  int pin_value = 0;
  char get_char = ' '; //read serial
  // wait for incoming data
  if (Serial.available() < 1) return; // if serial empty, return to
loop().
  // parse incoming command start flag
  get_char = Serial.read();
  if (get_char != START_CMD_CHAR) return;
  // parse incoming command type
  ard_command = Serial.parseInt(); // read the command
  pin_num = Serial.parseInt(); // read the pin
  pin_value = Serial.parseInt();  // read the value
  // 1) GET TEXT COMMAND FROM ARDUDROID
  if (ard_command == CMD_TEXT){
    inText =""; //clears variable for new input
    while (Serial.available())  {
      char c = Serial.read(); //gets one byte from serial buffer
      delay(5);
      if (c == END_CMD_CHAR) { // if we the complete string
has been read
        break;
      }
      else {
        if (c != DIV_CMD_CHAR) {
          inText += c;
          delay(5);
        }
      }
    }
  }

  // 2) GET digitalWrite DATA FROM ARDUDROID
  if (ard_command == CMD_DIGITALWRITE){
    if (pin_value == PIN_LOW) pin_value = LOW;
    else if (pin_value == PIN_HIGH) pin_value = HIGH;
    else return; // error in pin value. return.
    set_digitalwrite( pin_num,  pin_value); // Uncomment this
function if you wish to use
    return; // return from start of loop()
  }

  // 3) GET analogWrite DATA FROM ARDUDROID
  if (ard_command == CMD_ANALOGWRITE) {
    analogWrite( pin_num, pin_value );
    return;  // Done. return to loop();
  }

  // 4) SEND DATA TO ARDUDROID
  if (ard_command == CMD_READ_ARDUDROID) {
    Serial.print(" Analog 0 = ");
```

```
  Serial.println(analogRead(A0));   // Example: Read and
send Analog pin value to Arduino
  return; // Done. return to loop();
  }
}

void set_digitalwrite(int pin_num, int pin_value)
{
 switch (pin_num) {
 case 13:
  pinMode(13, OUTPUT);
  digitalWrite(13, pin_value);
  break;
 case 12:
  pinMode(12, OUTPUT);
  digitalWrite(12, pin_value);
  break;
 case 11:
  pinMode(11, OUTPUT);
  digitalWrite(11, pin_value);
  break;
 case 10:
  pinMode(10, OUTPUT);
  digitalWrite(10, pin_value);
  break;
 case 9:
  pinMode(9, OUTPUT);
  digitalWrite(9, pin_value);
  break;
 case 8:
  pinMode(8, OUTPUT);
  digitalWrite(8, pin_value);
  break;
 case 7:
  pinMode(7, OUTPUT);
  digitalWrite(7, pin_value);
  break;
 case 6:
  pinMode(6, OUTPUT);
  digitalWrite(6, pin_value);
  break;
 case 5:
  pinMode(5, OUTPUT);
  digitalWrite(5, pin_value);
  break;
 case 4:
 pinMode(4, OUTPUT);
 digitalWrite(4, pin_value);
break;
 case 3:
  pinMode(3, OUTPUT);
  digitalWrite(3, pin_value);
 break;
 case 2:
  pinMode(2, OUTPUT);
```

```
  digitalWrite(2, pin_value);
  break;
 }
}
```

IV. CONCLUSIONS

Android and Arduino based remote switching system is completely constructed. Accuracy is also good. Limiting range is about 15m in radial distance. This program is only for Arduino UNO version. If other versions (for example Arduino micro, Arduino nano, YUM, Mega) are wanted to use, the software from the Android must be repaired. In this project, readymade program is used. For those who want to write their own program shall be good in Android programming. There are many readymade android application programs are available on the internet base on the requirements. Although this project used readymade application, it is powerful and has many functions. This application has 10 port toggle switch, 4 channel pulse width modulation, data transmission and receiving display facilities. For the pulse width modulation part, the system needs powerful SCRs (Silicon Controlled Rectifier) to control the output interface.

In this paper, only four channel interfacing is used for the demonstration purpose only. It provided for three 5W lamps and one auxiliary socket. In this socket, the other electrical applications can be used such as Television, DVD, Fan, Hair dryer etc. Even though the socket may use many applications, the electrical appliance which is more than 1000W should not be used to avoid from damage. Because the contact rating of resistive load is maximum 5A and 250VAC. For the data transmission and receiving part, the project used RN-41 Bluetooth module from sparkfun electronics. It has only 4 pins to operate and very easy to use. When power is turned on, the red LED is blinking and it means ready to connect. When it is connected with other device, the LED color changes to green color and not blinking under operation. According the reference of the Sparkfun electronic Bluetooth module RN-41, it can be used for 106m or 350ft. In actually, after testing under this project, only 15m or about 45ft can be safely use. It may differ from the baud rate of the Bluetooth module. Serial communication is used in the project. For the further extension research, wifi modules will be used instead of Bluetooth module because of the long range distance. In conclusion, even the system is good in performance and use, it has shorter range. To get longer range communication, wifi module and Ethernet modules should be used.
.

V. REFERENCES

[1] Floyd T.L., "**Electronic Devices",** Prentice-Hall International, Inc., 1996.

[2] Floyd T.L., **"Digital Fundamentals",** Prentice-Hall International, Inc., 1997.

[3] Julien B, **"C Programming For Arduino",** Packt Publishing, May 2013.

[4] http://en.wikipedia.org/wiki/

[5] wwwhttps://www.sparkfun.com/products/12575

[6] https://www.techbitar.com/projects

[7] http://arduino.cc/en/Main/ArduinoBoardUNO

[8] www.shoutmeloud.com/top-mobile-os-overview.html

[9] http://electronics.wikia.com/wiki/Rectifier_diode