

Beginners Approach to the Open Source Programming: Case Study Arduino with ESP32

Pravin Bhadane^{1*}, Aparna Lal²

^{1,2}Department of Electronics, Nowrosjee Wadia College, Pune, India-411001

Corresponding Author: pravin16163@gmail.com, Tel.:9823855341

Available online at: www.ijcseonline.org

Accepted: 19/Oct/2018, Published: 31/Oct/2018

Abstract: Open source programming of arduino with case study of ESP32 has been studied. Open source programming is suitable for both students as well as for beginners. The basic concept of Open System Software (OSS), particularly its need, application and suitability for IoT has been studied with suitable example. One of the popular OSS platform i.e. Arduino is considered as a case study. Open source programming of Arduino for the beginners in the field of embedded system as well as IoT is carried out with the sample microcontroller ESP32. The reasons for popularity of ESP32 microcontroller, particularly integration of radio frequency module on its silicon chip, are discussed. It includes concept of open source programming, downloading and installation of open source Integrated development Environment (IDE) software, understanding of IDE functions, coding and finally burning of hex code into the flash memory of microcontroller. It also includes some technical insights of programming process. The present work is suitable for beginners as well as for new students in the field of electronics and computer science.

Keywords: Open source programming, Arduino, ESP32.

I. INTRODUCTION

Embedded system and IOT [1] are basically some applications which include both hardware and software. Hardware is an electronic circuit constructed from electronic components like sensors, transistors and integrated circuits(IC). Microcontroller is one of the important IC of IoT necessary for data manipulation and communication [2]. Hardware can be constructed by purchasing electronic components from the market and connecting them to each other by using soldering gun on printed circuit board (PCB). Hardware is never free of cost and it is available either from manufacturer or dealer. Most of the hardware is license free i.e. no need to take permission of manufacturer for its use in some application. Commercial use of some hardware components needs permission of manufacturer, but such components are less in number.

Software is a necessary part of any IoT or embedded system and without its use the hardware has no value. Software is a program written in programming language such as Assembly or C language [3] Its code is different for different applications. In case of embedded system and IOT, the software is developed on some other machine like PC or laptop and then copied or burned into the flash memory of microcontroller of an embedded system. To develop a software i.e. to write a program, following software tools are required: editor, linker, library, compiler, burner. These

tools are developed by some software industries working in the field of system programming [4] and low level programming. The tools developed by them are freely available or purchased by paying some charges. The freely available softwares are of following two types.

Evaluation version is for the promotion of software but it is valid for some period or its code capacity is limited. In some evaluation versions some water marks are included, some are not giving complete results. Such evaluation versions are suitable for study purpose and they are generally used by students and educational institutes for the study purpose and not for the commercial use. After learning of evaluation version sometimes user purchases the full version of software which has all facilities that are expected in that software. The purchased software is licensed to the user which is valid for one year or lifetime period. Some licence software are perceptual i.e. for one year they can be upgraded and after that they can be used without upgradation. The cost of software depends on its type and importance. For example, antivirus softwares are of low cost, on the other hand the operating system software is expensive.

In the present work, we have tried to understand the basics of open source programming which are necessary for beginners in the field of electronics. The article is divided into following sections. Section II contains work related to

B) Open source software

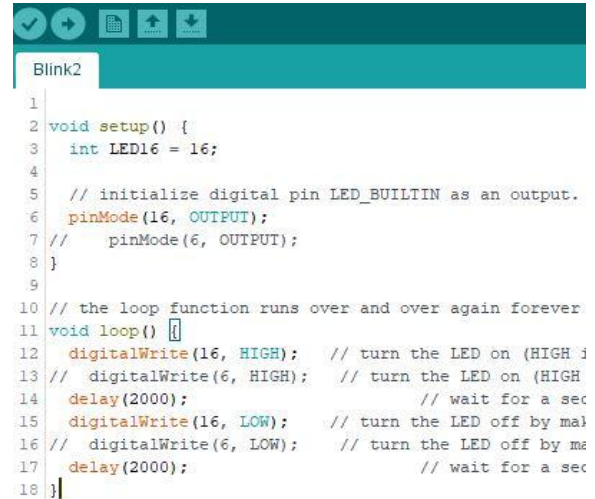
Arduino has its own software, i.e Arduino Nightly which is open source, and reasons for its popularity are given below: freely downloadable from the site of developer, easy to install, simple to use, facility to include libraries of different microcontrollers, direct download of hex code into the flash memory of microcontroller, no need of external programmer or burner, lot of reference study material available on internet, etc.

C) Installation procedure of open source software

- Search for OSS on Net, for example Arduino IDE.
- Download it from developers site or third party software provider for example <https://www.arduino.cc/en/Main/Software>.
- Install the OSS by running the Setup application program, e.g. arduino-1.8.5-windows.
- During installation procedure, answer the questions and proceed to the next stage by clicking on 'Next' button.
- Add the library of desired microcontroller device from the online store e.g. library of ESP32 microcontroller can be included by running GIT GUI. Download GIT 2.16.2 and install it [10].
- After installation of Git, search for GIT GUI and click on it. Then next window appears with three options, select option 'clone existing repository'.
- In the next window write the source and target locations. In Source location write <https://github.com/espressif/arduino-esp32.git> and in target location write [ARDUINO_SKETCHBOOK_DIR]/hardware/espressif/esp32, where ARDUINO-SKETCHBOOK_DIR can be found in the pop down option 'preference' on the GUI screen of Arduino IDE e.g. C:/Users/ruisantos/Documents/Arduino /espressif/esp32 [11].
- Now open the Arduino IDE and go to the Tool section and select the microcontroller board from board manager.
- Set the port in Tool section to a value shown by Device manager in the manage option of 'My computer' icon on Desktop.
- Select the required programmer in the Tool section.
- This completes the installation procedure and now program can be edited and executed in the IDE.

D) Editing of source code

Open source software sometime provides editor for typing the program e.g. Arduino has its own editor which is a part of IDE. The program can be typed in an editor with some rules e.g. initialisation of variables is done in the setup section and repetitive code is typed in a loop section. Figure 3 shows the screen shot of Arduino editor window.



```

1
2 void setup() {
3   int LED16 = 16;
4
5   // initialize digital pin LED_BUILTIN as an output.
6   pinMode(16, OUTPUT);
7   //   pinMode(6, OUTPUT);
8 }
9
10 // the loop function runs over and over again forever
11 void loop() {
12   digitalWrite(16, HIGH); // turn the LED on (HIGH is
13 //   digitalWrite(6, HIGH); // turn the LED on (HIGH
14   delay(2000); // wait for a sec
15   digitalWrite(16, LOW); // turn the LED off by ma
16 //   digitalWrite(6, LOW); // turn the LED off by ma
17   delay(2000); // wait for a sec
18 }

```

Figure3: Screenshot of Arduino open source IDE.

E) Compilation of program

Compilation is a main part of OSS. Generally, compiler part of OSS is not allowed to modify to the users as well as contributors. In Arduino IDE, program is compiled by selecting the verify/compile option. The lower part of Arduino workspace shows the error window, where the errors occurred during compilation are displayed. The errors should be removed from the program by writing the correct syntax of statements.

F) Uploading of the object or hex file

After compilation, hex file is generated which is a binary code of typed programs. After pressing the upload button, the hex file is uploaded or burned into the flash memory of microcontroller. Execution of program starts immediately on the microcontroller system. As soon as the hex file is burned into the flash memory, its execution starts and the LED starts blinking as soon in figure 4.

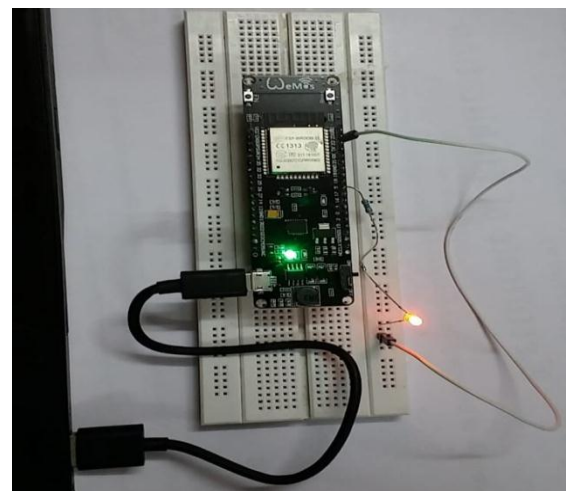


Figure 4: Assembly of ESP32 microcontroller and LED on bread board

IV. CONCLUSIONS

Open source software is a program which is open to modify to the contributors. It is freely available to the users for non commercial use. The main purpose of OSS is to enhance its features and increase the applications. They are popular among the students community for the completion of their project work. Arduino is one of the open source platform which includes both hardware and software. It is open source because its software supports another manufacturers microcontrollers e.g. ESP32. It has facility to include the libraries of different microcontrollers. By adding such libraries, Arduino can be used as a cross-compiler and hex code burner for the particular microcontroller. The present work is for beginners in the field of electronics and computer science which gives brief knowledge about use of open source software.

V. FUTURE SCOPE

In future it is possible to contribute in the development of open source software by writing header files in a given format. Furthermore, it is possible to write another open source software which has similar features like Arduino IDE or with new features.

REFERENCES

- [1] Anusha Bharati, Ritika Thakur, Kavita Mhatre, "Protection of Industrial and Residential areas by Wireless Gas Leakage Detector using IoT and WSN", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.3, pp.62-67, 2017.
- [2] K.M. Rudrappa, B.P. Divakar, "A Programmable System on Chip (PSOC) for Active Power Filter (APF) Based on Cortex M3", International Journal of Scientific Research In Network Security And Communication, Volume-5, Issue-2, 2017.
- [3] Bhojaraj Barhate, Dipashri Sisodiya, Rakesh Deore, "Application of Speech Recognition: For Programming Languages", International Journal of Scientific Research in. Computer Science and Engineering, Vol.06, Issue.01, pp.6-8, 2018
- [4] D.M. Dhamdhere, "Systems programming", Edition 1(2011).
- [5] Hemant Malviya, Sita Solanki, Survey of Quality "Models for Evaluating Quality in OSS (Open Source Software)", International Journal of Scientific Research in Computer Science and Engineering, Vol.4, Issue.1, pp.15-20, 2016.
- [6] Krupali Mistry, Avneet Saluja, "An Introduction to Open CV using Python with Ubuntu", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 1 ,Issue 2 ,2016.
- [7] Kumari Seema Rani, "Open Source Software: A Prominent Requirement of Information Technology", International Journal of Scientific Research In Network Security And Communication, Volume-6, Issue-2, 2018.
- [8] Leo Louis, "Working principle of arduino and using it as a tool for study and research", International Journal Of Control, Automation, Communication And System, Vol.1, No.2, 2016.
- [9] Programming Aurdino Getting started with sketches- McGraw Hill, 2011.

- [10] John D. Blischak, Emily R. Davenport, Greg Wilson, "A Quick Introduction to Version Control with Git and GitHub", PLOS Computational Biology, 2016.
- [11] Neil Kolban, Kolbans book on ESP32, 2017.

Authors Profile

Pravin K. Bhadane is Head of the Electronics Department at Nowrosjee Wadia College, Pune (India). His area of research is in the field of electronics.



Aparna Lal is post graduate student of Electronics at Nowrosjee Wadia College, Pune (India).

