Advanced Research in Electrical and Electronic Engineering
p-ISSN: 2349-5804; e-ISSN: 2349-5812 Volume 5, Issue 1 January-March, 2018, pp. 60-63
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/Publication.html

Arduino – Exploring More Applications

Asif Akhtar[1] **and Nancy Agarwal**[2]

[1,2]*Department of Electronics and Communication SRCEM, Palwal, Haryana, M.D. University*
[1]*asifakhtar56 @gmail.com*, [2]*nancy.aggarwal00@gmail.com*

Abstract—*This technical paper explains how Arduino platform can be used for developing gamut of applications including Arduino based touch pad, Arduino based universal remote control, Arduino based GPS system, Arduino based traffic light controller etc., resulting low NRE (Non-Recurring Engineering) cost and also low time-to-market. Here you can learn not only the hardware but also the running software behind the picture. By following some simple steps, beginners can also develop as well as further modify the application as per need or requirement. Low NRE cost and low time-to-prototype are some of the salient features of Arduino based applications.*

Keywords*: Arduino, open source software, open source hardware, projects, electronics, explore applications, projects*

1. INTRODUCTION

Arduino – designed for multidisciplinary projects/tasks, was initially designed and further developed in Ivrea, Italy. It is open source prototype which is accepted by both professionals as well as with hobbyists. Nowadays Arduino has been accepted for multitude of applications including Arduino based GPS system, Arduino based auto tap controller, Arduino based intelligent hot water controller, Arduino based stepper motor controller, Arduino based solar battery charger, Arduino based traffic light controller, Arduino based Zigbee and many more.

The salient feature of this fantastic concept is "FOSS" (Free and Open Source Software). Now the user is completely free to download different source codes which are by default present in Arduino libraries. We can download respective library of our application and further modify the library so that the new modified library could be used for further applications.

In section I we discuss the hardware of Arduino development board.

Designed on the concept of "Open Source Hardware & Software", Arduino brings flexibility for every level of users. Arduino is simply based on flexible hardware and software concept. In a nutshell, using Arduino multitude of applications can be easily developed as well as further modified.

Board details are as follows:

1. Processor
2. Power Supply
3. Reset Button
4. Digital I/O Pins
5. Programmer
6. Crystal

Let's start discussion one by one.

1. Processor:

Arduino platform is based AVR (Advanced Virtual RISC Processor) development board. AVR microcontroller – 8-bit RISC (Reduced Instruction Set Controller) single chip microcontroller, is a modified Harvard architecture. AVR was discovered by Atmel in the year 1996. Arduino development board consists of numerous of ATmega AVR family including ATmega8, ATmega168, ATmega328, ATmega1280 and lots more. See fig1.

2. Power Supply:

For working faultless, Arduino development board needs 12V DC supply that can easily provided by a power adapter. This simple power supply circuit consists of rectifier circuit, filter capacitor and voltage regulating IC. See fig1.

3. Cable:

You should need a FTDI (Serial-to-USB) cable which will establish a connection of Arduino development board with computer. See fig1.

4. RESET Button:

When the software jumps into endless loop (that will never end) then user can directly press the RESET button. After pressing the RESET button, the whole system restarts itself. This special feature helps new users in any kind of complexity. See fig1.

5. Digital I/O Pins:

For connecting any of output devices including sensors, LCD (Liquid Crystal Display), LED (Light Emitting Device), DC motor, touchscreen and many more devices digital input/output pins are available. See fig1.

6. Programmer:

For burning of codes into any of the ATmega chips, we need programmer. Programmer is a special kind of device which is specially designed for burning the target IC. Arduino development board has on-board programmer resulting reducing extra hardware. See fig1.

7. Crystal:

For converting the frequency into clock we need crystal. Normally we need 11.0592 MHz crystal. In addition to this, some board also consists of 9 MHz, 6 MHz and 12 MHz crystals. See fig1.

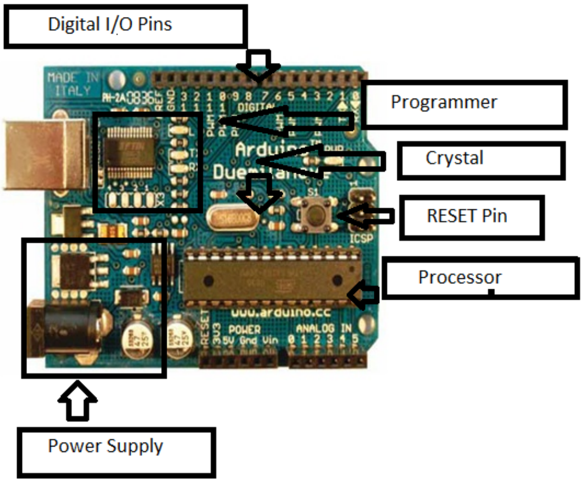


Fig. 1: This figure shows the hardware of Arduino development board

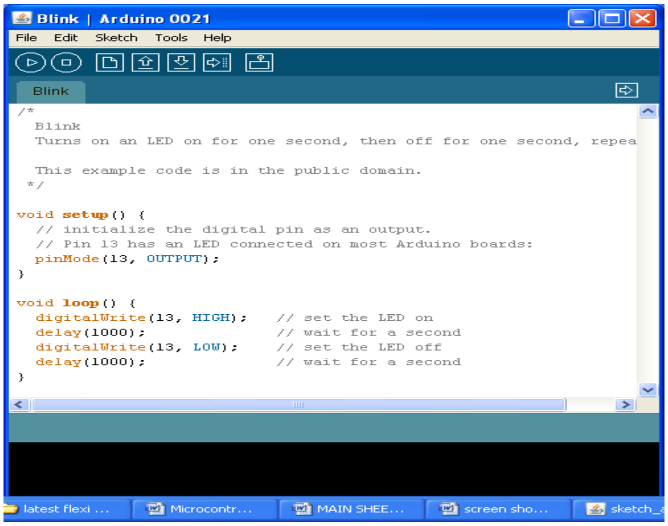
In section II we discuss the software for Arduino development board.

As we know that Arduino is an "Open Source Software" which is completely free to download from the Arduino website or directly from the internet. Since this is open source software i.e., user shouldn't need any kind of license. In addition to this user can also download the "LIBRARY" which is also completely free.

Libraries are some special kind of files which provides the sketch. Sketch is the platform for writing your program. Programs under libraries are either written in C or in C++.

If you are beginner then you can simple run the predefined libraries. But after getting more exposure on Arduino you can develop your own library and add to the predefined main library.

One of the salient features of Arduino software is that it can be run on any operating system including Windows, LINUX and MAC also. Drivers for any operating system (Windows, LINUX, and MAC) can be easily downloaded from Arduino website. See fig. 2:



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. 2: This figure shows sketch which is actually the working environment for Arduino development board.

In section III we discuss various sections various sections of source code.

Arduino Source Code consists of the following sections:

Sketch:

As we know that anything between /* and */ is completely ignored when Arduino runs. Sketch is the platform for writing the target code. It's good to mention the comments at the end of every code. This helps other people to learn or further modify the predefined codes.

For example:

int ledPin = 18; // LED connected to digital pin 18

Function:

Function is a piece of code which is written within the code.

Setup():

Setup() is a kind of function in the source code.

Keyword:

void setup()

pinMode ():

This is also a function which is used for making the pins either in input or output mode.

Keyword:

{

pinMode(ledPin, OUTPUT); // sets the digital pin as output

}

digitalWrite():

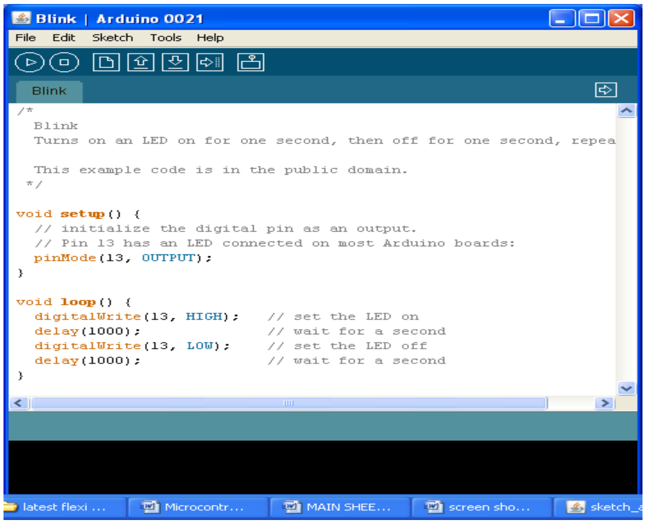
This function writes a output value on the pin of digital I/O ports. The digitalWrite() functions outputs a value on a pin.

For example, the line: digitalWrite(ledPin, HIGH);

In section IV we discuss various sections various sections of source code.

Start building applications

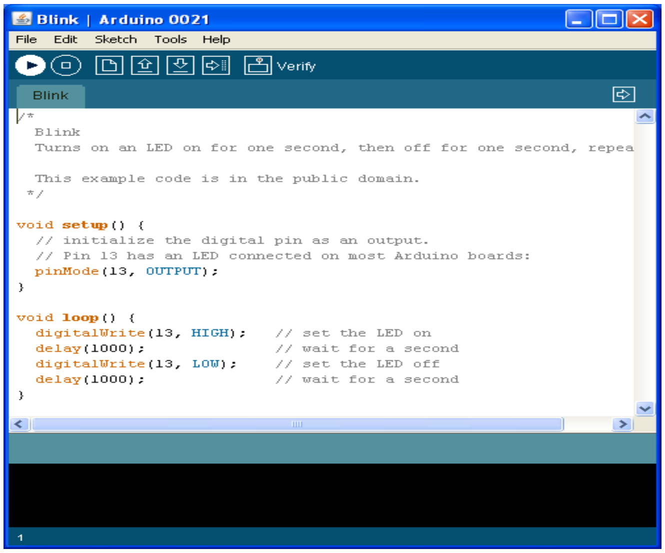
Step1: Directly open Arduino Software by double click at the Arduino icon. See fig1:



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. 3:

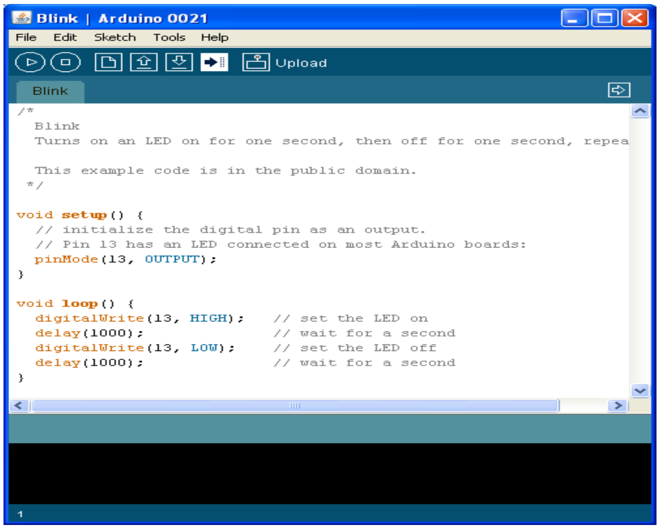
Step2: After writing the code we need to verify that the written code is whether correct or incorrect. For that we need to click at "Verify" icon. See Fig4:



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. 4:

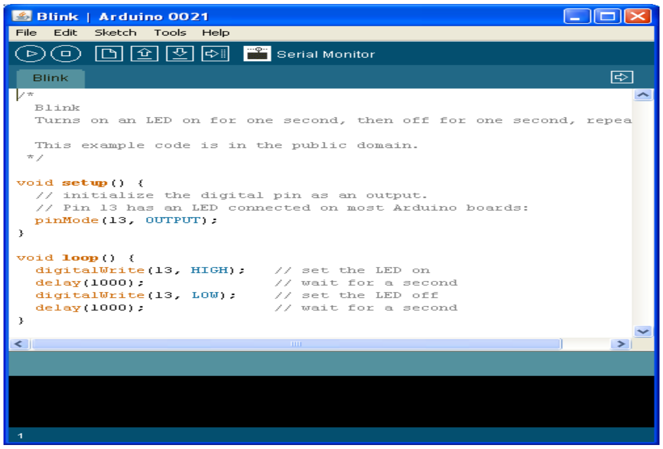
Step3: Next we need to upload our source code into the target IC. For uploading the source code into target IC we need to click at "Upload" icon. See Fig. 5:



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. 5

Step4: Depending upon the program we can either active or deactivate the serial monitor option by double clicking on "Serial Monitor" icon. See Fig. 6:



```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repea

  This example code is in the public domain.
 */

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Fig. 6:

2. CONCLUSIONS

This fantastic concept called "open Source Software" brings new trends and innovations resulting low time-to-market prototypes. Due to free availability on internet nowadays Arduino becomes the first choice amongst its users. Most of the developed countries have already been adopted this state-of-the-art technology and lots of developing countries are in the queue for adoption.

FUTURE WORK

This technology could be further modified and more upgraded as per individual need and interest. We have discussed some basic ideas of this technology. And depending on innovative applications user can upgrade as per requirement.

REFERENCES

[1] http://www.arduino.cc/

[2] http://arduino.cc/en/Tutorial/HomePage

[3] http://www.adafruit.com/tutorials

[4] http://www.instructables.com/id/Arduino-Expermentation-Kit-How-to-get-Started-wi/

[5] http://www.ladyada.net/learn/arduino/