# ARDUINO, HARDWARE AND PROCESSING: A VISTA

[1]Dambarudhar Das,[2]Ratikanta Sahoo,[3]Subhashree Pothal,[4]Maheswar Sahoo
Department of Mechanical Engineering
Raajdhani Engineering College, Bhubaneswar, India

***Abstract:*** In this article, a recent development of small-sized experimental devices for control research and education, with a special focus on the use of open-source technologies such as Arduino and Processing has been introduced. Arduino is a pronominal open-source hardware whose architecture, implementation and other necessary resources are accessible to every user, while Processing is its software counterpart which supports rapid development of controller/interface programs without much expertise. This work modestly approaches to unwind the hardware and processing inherent to some extent concern to Arduino.

***Index Terms*** -Control education, Ardiomp, Integrated development environment (IDE), GPL (Gnu General Public License)

## I. INTRODUCTION

With no doubt, experiencing laborious try-and-error processes in control experiment is a crucial part of advanced control education. Principles of control theory could be truly understood when they are associated with suitable experience of mathematical modeling, system identification, computer-aided controller design and its hardware implementation. Thus the control experiment projects are commonly assigned in most of the college courses, such as the inverted pendulum, wheeled mobile robots and the magnetic levitation system for instance. However, preparing control experiment system is usually a demanding task. Typical experimental setup consists of a desktop computer with sufficient bus card slots, A/DD/A conversion and pulse counter boards, external power supply, motor drivers, real-time operating system, and so on. Generally such a system tends to large and heavy, time-consuming if it is built from scratch, or expensive if it is dealt in as a commercial product. This is not only a burden for educational organizations, but also an obstacle for students those who intend to build their own experimental systems.

## II. ARDUINO- THE HARDWARE UNIT

The power of the Arduino is not its ability to crunch code, but rather its ability to interact with the outside world through its input-output (I/O) pins. The Arduino has 14 digital I/O pins labeled 0 to 13 that can be used to turn motors and lights on and off and read the state of switches. Each digital pin can sink or source about 40 mA of current. This is more than adequate for interfacing to most devices, but does mean that interface circuits are needed to control devices other than simple LED's. In other words, you cannot run a motor directly using the current available from an Arduino pin, but rather must have the pin drive an interface circuit that in turn drives the motor. A later section of this document shows how to interface to a small motor. To interact with the outside world, the program sets digital pins to a high or low value using C code instructions, which corresponds to +5 V or 0 V at the pin. The pin is connected to external interface electronics and then to the device being switched on and off. The sequence of events is shown in this figure no1.

To determine the state of switches and other sensors, the Arduino is able to read the voltage value applied to its pins as a binary number. The interface circuitry translates the sensor signal into a 0 or +5 V signal applied to the digital I/O pin. Through a program command, the Ardiomp interrogates the state of the pin. If the pin is at 0 V, the program will read it as a 0 or LOW. If it is at +5 V, the program will read it as a 1 or HIGH. If more than +5 V is applied, you may blow out your board, so be careful. The sequence of events to read a pin is shown in the above.

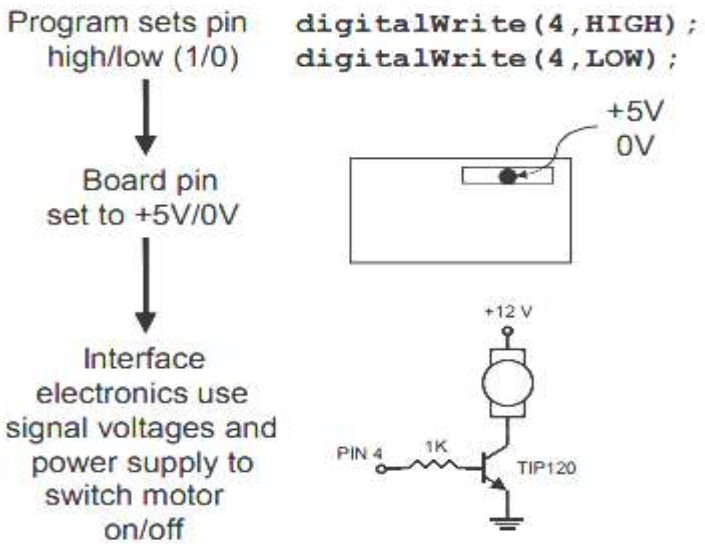


**Figure 1: Sequence of events**

## III. ARDUINO- THE PROCESSING UNIT

Open-source hardware implies electronic product whose architecture is free and open to everyone and the movements along this direction as well. Though the terminology is not rigorously established yet, it has been widely used as in the same manner as the so-called open-source software technology. Among the recent evolution of these activities, Arduino, a family of micro-computing platform, is considered a pro-nominal open-source hardware.


**Figure 2: - Duemilanove; The current flagship of Arduino**

As for software development, a free, cross-platform, easy to-use environment called Arduino IDE is available. A code editor, compiler and serial communication interface with the Arduino boards, are all integrated in a single java application. The programming language mostly resembles the C/C++ (supported by the "Wiring" library), quite easy to understand for novice programmers, and supports common features of modern object-oriented programming languages. These features are of much value since they enable the language to be suitable for students at various stages of learning. For example, details of hardware implementation can either be hidden or exploited in accord with their skill level without any customization of the language. The typical program, which blinks an LED, is as follows.

```
void setup () {
/* initialization procedure */
pinMode(ledPin, OUTPUT);
}
void loop () {
/* online tasks (blink a LED) */
digital Write(ledPin, HIGH);
delay(1000);
digitalWrite(ledPin, LOW);
delay(1000);
}
```

The API set is pretty straightforward and the implemented code is easy-to-follow. Also, the API set is equipped with essential time functions, such as delay () and millis (), which can be utilized to implement real-time controllers. Processing is free and open source software developed by Reas and Fry, which supports powerful and easy programming for visualization, education and electronic arts (Fig. 2). By virtue of its outstanding user-friendliness, users can quickly develop visual application without special experience in graphical software (Arduino IDE itself is indeed developed based on this technology). The programming language is essentially a simplified version of java with certain expansions specialized for visualization. Following is a typical Processing code.

```
void setup() {
size(400, 400);
stroke(255);
}
void draw() {
ellipse(mouseX, mouseY, 5, 5);
}
```

As it has been experienced, the code is very clear by virtue of its easy to-use API set, and complicated initialization procedures, which often distress beginners, are not required at all. Since the API includes functions to communicate with the Arduino boards through the serial (with USB/UART conversion if necessary), it is also suitable for building graphical interface for hardware control.

## IV.CONCLUSION

Arduino is based on easy-to-use hardware (including Atmel AVR microcontroller and simple I/O interface) and software development environment, partially intended for non-technical users such as artists and designers. Every detail of the hardware architecture is exposed, and its license is free (released under GPL (Gnu General Public License) based on the copyleft principle). Several shipped versions of Arduino are available at relatively low cost. Some of which are called official hardware, featuring Atmel mega AVR series as central processor. Notably, everyone may create his/her own Arduino at will; this results in a number of clones at various compatibility levels.

## V. ACKNOWLEDGMENT

We would like to acknowledge the faculty members of mechanical department and electronics department of Raajdhani Engineering College for their support and motivation to carry out the study.

## REFERENCES

[1]. Brown, S. W., Santana, C. & Eppeldauer, G. P. “Development of a tunable LED-based colorimetric source”, J. Res. Nat. Inst. Stand. Technol. **107**, 363–371 (2002).
[2]. Eka Cahyaprima, “Heat Transfer Lab Kit using Temperature Sensor based Arduino for Educational Purpose”, Department of Science Education, Faculty of Mathematics and Science Education, Universitas Pendidikan, Indonesia, 134, 265–278 (2017).
[3]. Everdell, N., Styles, I., Claridge, E., Hebden, J. & Calcagni, “A.Multispectral imaging of the ocular fundus using LED illumination”, In European Conferences on Biomedical Optics(2009).
[4]. Pedro Acevedo, “A Pulse Generator Based on an Arduino Platform for Ultrasonic Applications”, Universidad Nacional Autonoma de México, Av. Universidad No 3000. Col. Universidad Nacional, 89, 271–282 (2015).
[5]. Petteri Teikari, “An inexpensive Arduino-based LED stimulator system for vision research, Stem Cell and Brain Research Institute”, INSERM U846, 18 avenue Doyen Lépine, 69500 Bron, France, 33, 103–115 (2012).