

Real-Time Execution of UOB-iLabs Shared Architectures

Prof. Dr. Bahaa Ibraheem Kazim Mazin Rahman Ali
Mechatronics Engineering Department / Al-Khwarizmi Engineering College
University of Baghdad

Abstract

Laboratories availability is one of the key factor for the success of any academic program in the universities .Due to large number of students and the limitation in space, resource and time for using the laboratories in the universities a new style of laboratories had been developed that allows students to conduct real experiments remotely through internet environment (iLab).In this work we suggest a new design for internet laboratory that can be used by the student at mechatronics engineering department at the University of Baghdad. This internet laboratory can be used with limited internet bandwidth and several concurrent accesses of users and administrators. The suggested design for University of Baghdad internet laboratory (UOB-iLab) depends on the Arduino Microcontroller to establish the interface with various laboratory devices using C and C++ program language to control input/output signals for a specific lab device.The Web interface for UOB-iLab has been developed using HTML, PHP and AJAX in addition to JAVA language to build graphical real time interfaces for the experiment. Several experiments were designed to the check the reliability of the suggested UOB-iLab architectures. Different types of input- output configurations were tested like (Controlling a DC motor using Arduino, Python and PHP).The suggested design of UOB-iLab Weblab server shows good reliability for remotely operated laboratory experiments through web environment in comparison with other available platforms. Also, the Arduino Microcontroller can be easily used as a controlling platform for iLab since it is compatible with different types of sensors and actuators. The designed web interface using HTML-JAVA- PHP- JavaScript code is used successfully to be interfaced with Arduino Microcontroller, and Processing Language was very helpful in building Applet Platform directly. The suggested design of UOB-iLab can be used to design several types of experiments to improve the lab skill of mechatronics engineering department at University Of Baghdad and other Universities.

Keywords: internet laboratory, remote control, weblab

1. Introduction:

Internet technology has made it possible for schools, universities and other educational institutions to offer Web-based online laboratories as an alternative to the traditional laboratory experience. Online laboratories are remotely-operated experiments that are conducted in real-time over the internet.

Educational institutions that would have struggled or been unable to provide a sufficient traditional laboratory experience and equipment for itself , such that a distance-learning technology give universities to use online laboratories to incorporate laboratory experiences in their course curriculum.

Consequently, more students in technical and science courses are able to participate in valuable laboratory experiences. Even universities without such concerns scan benefits from online laboratories as a complement to their existing hands-on facilities [1].

Online laboratories are flexible and can be accessed from anywhere at any time with large number of users. The flexibility lets students conduct their experiments at a convenient time and location. It also allows institutions to keep their laboratories open much longer than typically possible with the traditional hands on laboratory. Furthermore, universities can use online laboratories to expand their offering of available laboratory experiments with minimal additional cost so that the entire class of students can perform the same experiment on shared equipment. Certainly, online laboratories are not a perfect replacement for traditional laboratories, be can the students do not get hands-on experience with setting up experiments and have limited debugging abilities.

2. iLab Project

The iLab project was started at MIT in 1998 by one of the authors (del Alamo) [1]. It was several years before the project acquired its final name. The initial inspiration for the first iLab came from the frustration that MIT's courses on semiconductor devices did not contain a laboratory component. Traditionally, students in these courses were exposed only to theoretical device models presented in lectures and course texts. At the same time, an Agilent 4155B semiconductor parameter analyzer as shown in Figure 1, an expensive piece of equipment bought under a research contract, was sitting in a graduate research lab with spare capacity available. While the underutilization of the Agilent equipment seemed to provide an opportunity to have this tool also used in education, there was no way to accommodate the students taking courses using a single piece of equipment in the crowded research lab.



Figure 1. Agilent 4155B analyzer [2]

A small initial grant from the Microsoft Corporation allowed del Alamo to explore the potential of remote access to the 4155B. He hired an undergraduate student who wrote a Java applet that enabled students using a standard Web browser to submit descriptions of semiconductor device characterization routines for execution by a server connected directly to the Agilent instrument.

Students in an upper level electrical engineering course were the first to try this system, Microelectronics WebLab, in the fall of 1998. By the following spring, the hardware and software combination had proved its reliability to the point that an undergraduate class of nearly 100 students employed the online lab for an assignment [1].

3. iLab Overview

iLab are traditionally designed by electronic and control engineers who naturally tend to place a major attention on the hardware side of the system they usually follow a three step process as shown in Figure 2:

- a. Choose a programmable device.
- b. Attach it to a server, accessible through the web or simply a TCP/IP socket.
- c. Design a simple protocol to record program in the remote device (Service Broker) which is regard Database management system.

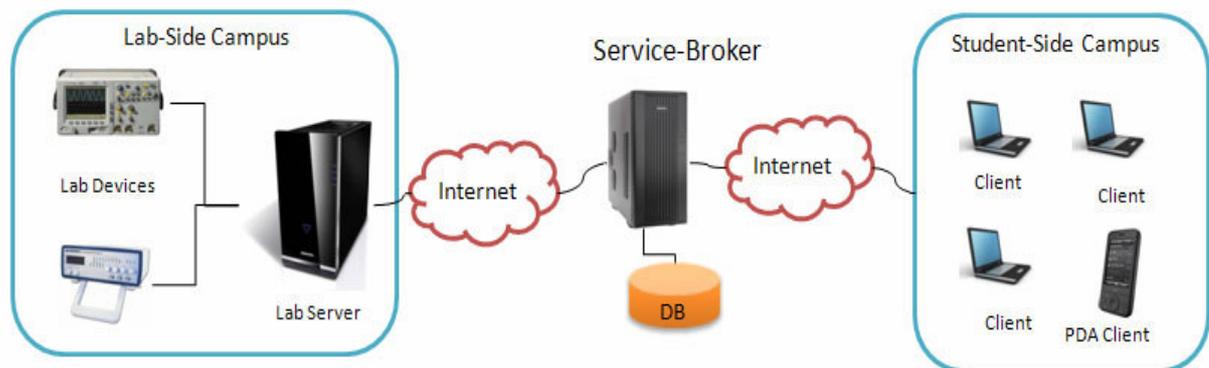


Figure 2. iLab Shared Architecture[3]

From the perspective of the ISA (Internet Shared Architecture), online experiments fall into two broad categories.

1. Batch experiment:

The batch experiment is an online experiment with the facility that an entire course of the experiment can be specified before the experiment gets running. Experiment execution takes place in machine time.

2. Interactive experiment :

The interactive experiment is an online experiment in which the user monitors and controls one or more aspects of the experiment during execution .Experiment execution takes place in human time. It should appreciate that UOB-iLab project belongs to interactive experiments.

4. iLab Architecture:

iLab architecture consists of three layers ,Client , WebLab server, and experiment device .WebLab server is responsible for management of lab server experiments for the students who will perform experiments , as shown in Figure 3.

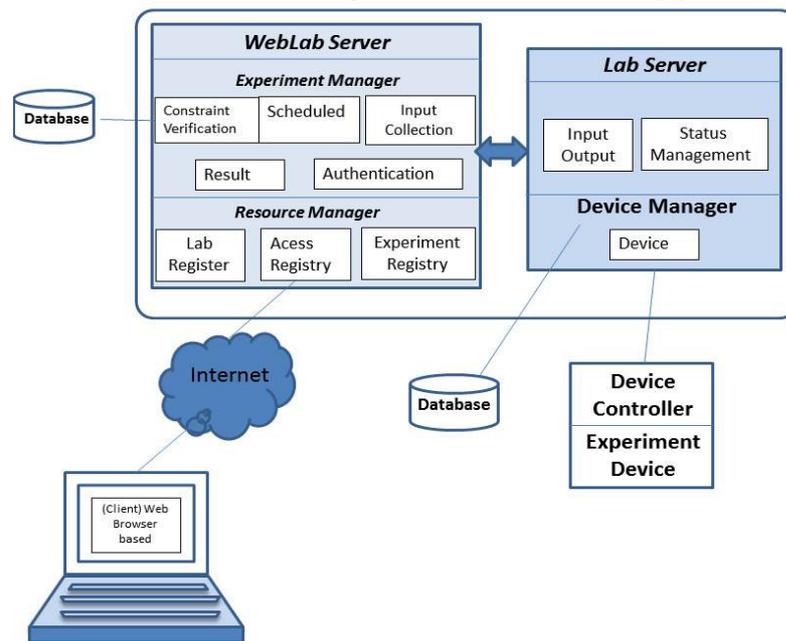
The WebLab server plays an important role in the management of the relationship between the lab server at laboratory side and client (user) at the other side.

The role of weblab server is:

- a. Management of authorization
- b. Authentication

- c. Registration of users
- d. Scheduling and periodization of users.

Figure 3. WebLab Architecture[4]



The lab server role is controlled the lab devices to implement a specific set of experiments according to the client (user) need.

5. Advantages of UOB-iLab:

The design and use of UOB-iLab in the Colleges of Engineering has the following advantages:

- UOB-iLab has better performance for using laboratory equipment since they are available to students during 24 hours a day and 365 days a year. This means that both students and professors can better organize their study time.
- UOB-iLab can be considered as an add value for distance learning and to face to face learning models.
- Integration of handicapped students: since all the hardware equipment is controlled by a computer, they may be used by handicapped students with software / hardware techniques specially designed for their particular needs.
- UOB-iLab is flexible and can be accessed from anywhere at any time with large number of users.
- UOB-iLab can be used to integrate the laboratory resources between several universities around Iraq.
- UOB-iLab can be integrated with other international resources of iLab around the world.

6. Research Objectives :

The main objectives of the present work can be summarized as:

- a. Suggesting a new design for iLab system that can be used by students at Mechatronics Engineering Department at the University of Baghdad (iLab -UOB). The suggested design can provide secured authentication for client and administers, low cost, and can work with limited internet Band-width.
- b. Implementing the suggested design for UOB-iLab using Arduino microcontroller platform.
- c. Designing the required web interface to control and interact with electronic and mechanical devices.

7. UOB-iLab Architecture

The suggested design for UOB-iLab must be reliable and can be implemented easily at the University of Baghdad with following design features:

- a. The system can be used by mechatronics engineering students at Mechatronics Engineering Department at the University of Baghdad (UOB-iLab).
- b. The design of (UOB-iLab) architecture can be used to support secured authentication for clients and administers, with low cost, and limited internet Band-width.
- c. The suggested design for (UOB-iLab) can be implemented using Arduino microcontroller platform with

web interface.

- d. The design and implementation of the required web interface can be used to control and interact with electronic and mechanical devices.

Figure 5 represents the input/output connection pins and the function of them for Arduino board.

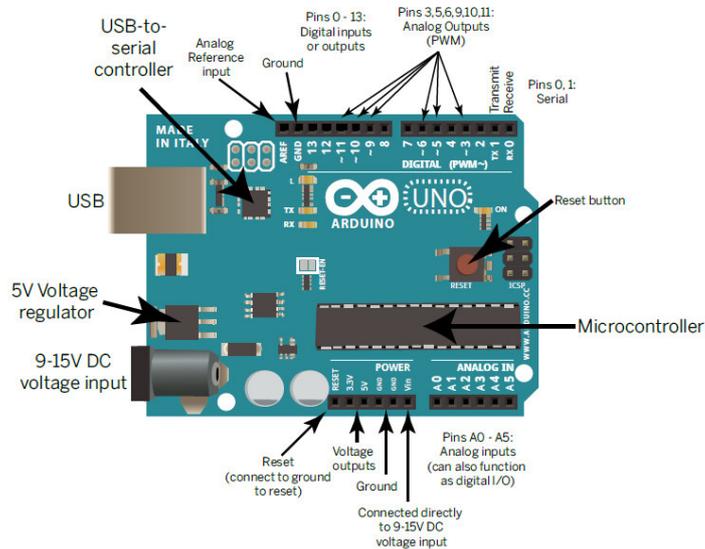


Figure 4. Functional parts of an Arduino. Most microcontrollers have the same or similar parts: power connections, digital and analog inputs, and serial communications

The UOB-iLab platform can be divided to two parts:

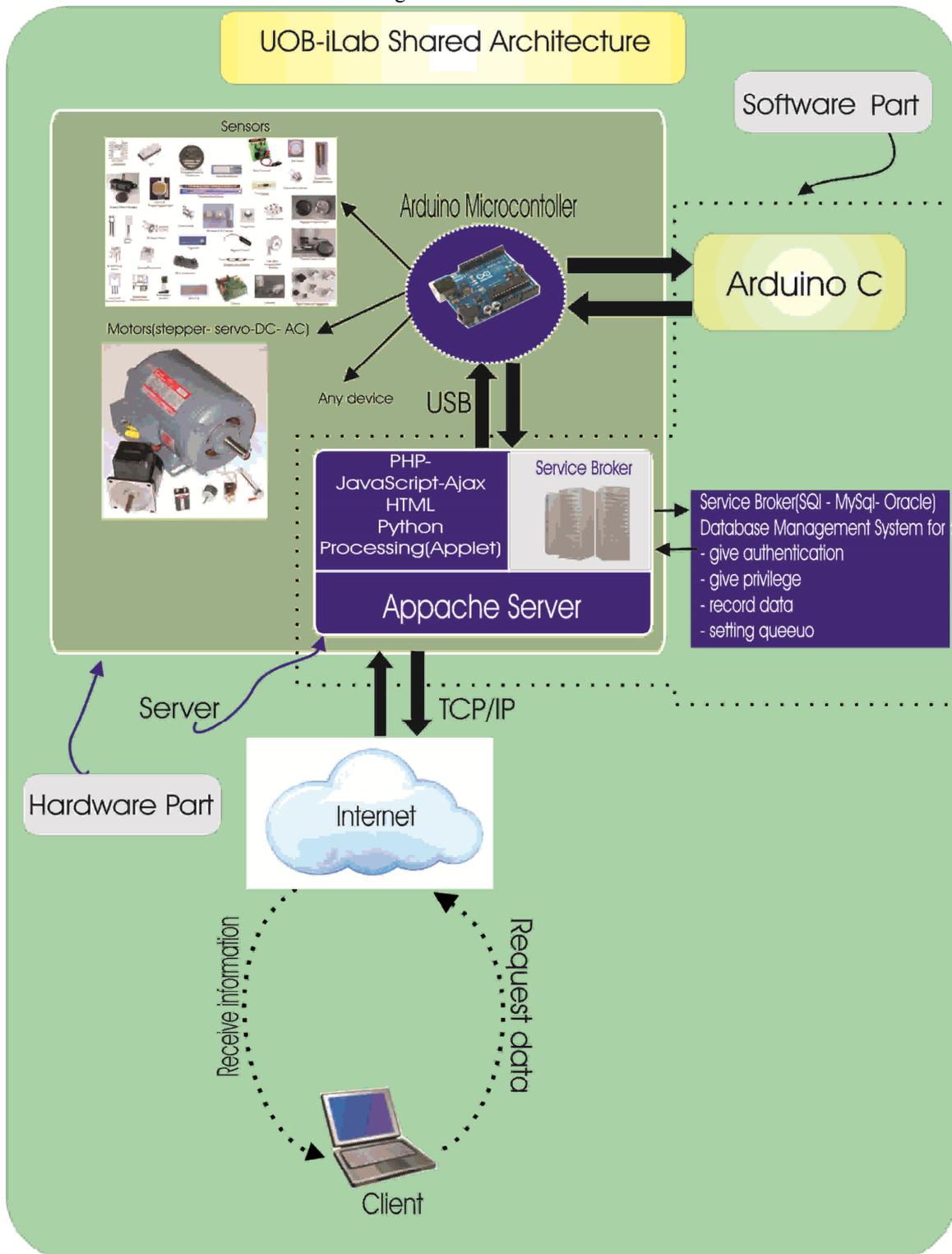
1. The Software platform
2. The Hardware platform

Figure 5 shows the basic design for each part and the required programming and interface languages to complete data transformation and acquisition between the user interface and laboratory device. In this research, been dealing the most important needs in electronics laboratory and to achieve the completion of testing in an integrated manner, such as :

1. Switching process of equipment and devices.
2. Getting the results value from sensors.
3. Modifying the value of process variables.
4. Achieving the concept of feedback control system.

5. Monitoring of experiments in real time.

Figure 5. UOB-iLab Architecture



8. Arduino microcontrollers:

Arduino is a small microcontroller board with a USB plug to connect to our computer as shown in Figure 5 and a number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors, laser diodes, loudspeakers, microphones, etc. They can either be powered through the USB connection from the computer or from a 9V battery. They can be controlled from the computer or programmed by the computer and then disconnected and allowed to work independently [5].

Arduino is an open source code, that environment makes it easy to write code and upload it to the I/O board. It runs on Windows, Mac OS X, and Linux.

The environment is written in Java and based on Processing, avr-gcc, and other open source software [6]. Physical computing platform is based on a simple input/output (I/O) board and a development environment that implements the Processing, Arduino is composed of two major parts:

Part 1: the Arduino board, which is the piece of hardware you work on when you build your objects.

Part 2: the Arduino IDE, the piece of software you run on your computer [7].

9. Arduino Software Platform:

The Arduino integrated development environment (IDE) shown in Figure 6 is a cross-platform application written in Java, and is derived from the IDE for the Processing programming language and the wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. There is typically no need to edit make files or run programs on a command-line interface. A program or code written for Arduino is called a sketch.[8]

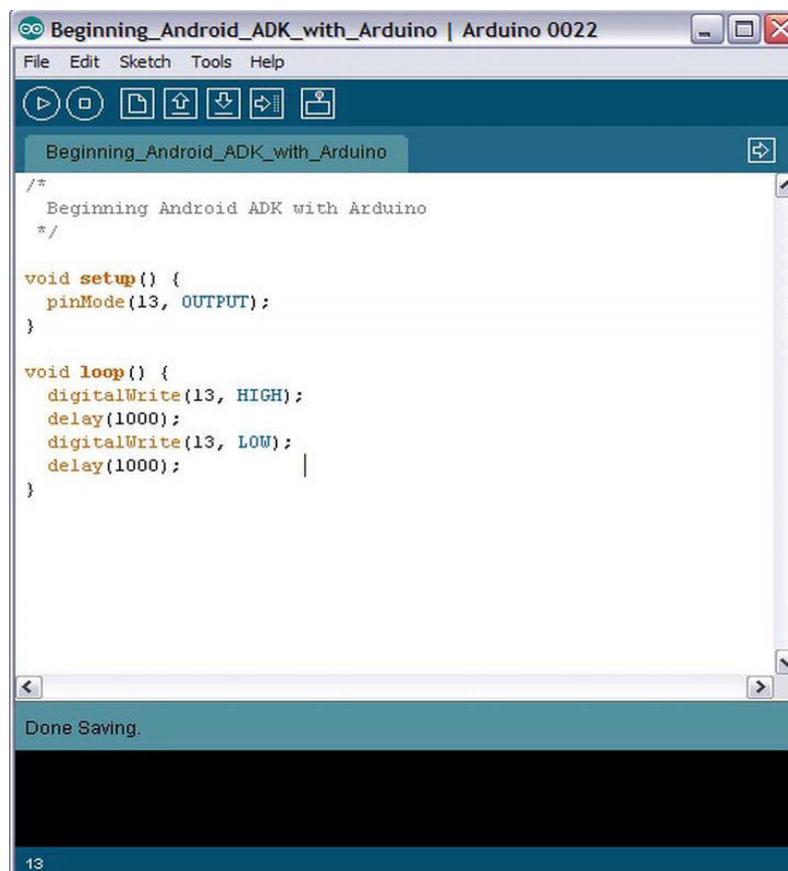


Figure 6. Arduino IDE main window

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original wiring project, which makes many common input/output operations much easier. Users only need to define two functions to make a runnable cyclic executive program:

- `setup()`: a function runs once at the start of a program that can initialize settings
- `loop()`: a function called repeatedly until the board powers off

10. Integrated relation between Arduino Microcontroller and Python in UOB-iLab:

Python programs are indeed often deployed in the context of larger applications. For instance, to test hardware devices, Python programs may call out to components that give low-level access to a device. Similarly, programs may run bits of Python code at strategic points to support end-user product customization without the need to ship and recompile the entire system's source code. Python's simplicity makes it a naturally flexible control tool. Technically, though, this is also just a common Python role; many (perhaps most) Python programmers code

standalone scripts without ever using or knowing about any integrated components. It is not just a control language [9].

In order to support communication with the Arduino board a method of interfacing is required. Standard Arduino boards provide RS232-style serial connection older boards over a physical RS232 connection but with such a port that is no longer common to modern computers, recent boards include USB for serial functionality. The approach of connecting Arduino over a serial interface is pretty trivial in Python. On Windows-like systems we can read and write to the serial device as if it was a file, but there is also a wrapper library called (pySerial) that works well across all operating systems [9]. After installing (pySerial), reading data from Arduino is straightforward:

```
>>> import serial
>>> ser = serial.Serial('COM11', 9600)
>>> while True:
...     print ser.readline()
'1 Hello world!\r\n'
'2 Hello world!\r\n'
'3 Hello world!\r\n'
```

Writing data to Arduino is easy too:

```
>>> import serial # if you have not already done so
>>> ser = serial.Serial('COM11', 9600)
>>> ser.write('5')
```

Note that we will need to connect to the same device that we connect to from within the Arduino development environment. We created a symlink between the longer-winded device name and COM11 to cut down on keystrokes. It is worth noting that the example above will not work on a windows machine; the Arduino serial device takes some time to load, and when a serial connection is established it resets the Arduino.

Any (write() commands) issued before the device initialization will be lost. A robust server side script will read from the serial port until the arduino declares itself ready, and then issue write commands. Alternatively it is possible to work around this issue by simply placing a 'time.sleep(2)' call between the serial connection and the write call. The major role to use python program in UOB-iLab is to convert the result data from Arduino microcontroller from binary form to another form such as (CSV – TXT- XCL... etc.) which enables the other program to analyze it easily as shown in Figure.6.

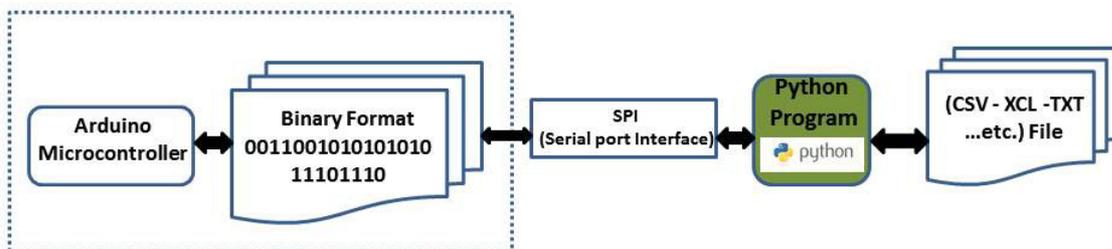


Figure 7. Algorithm of mechanism procedure building of Arduino Experiment

Figure 8 shown clearly the algorithm or flowchart of mechanism build of Arduino microcontroller experiments in UOB-iLab.

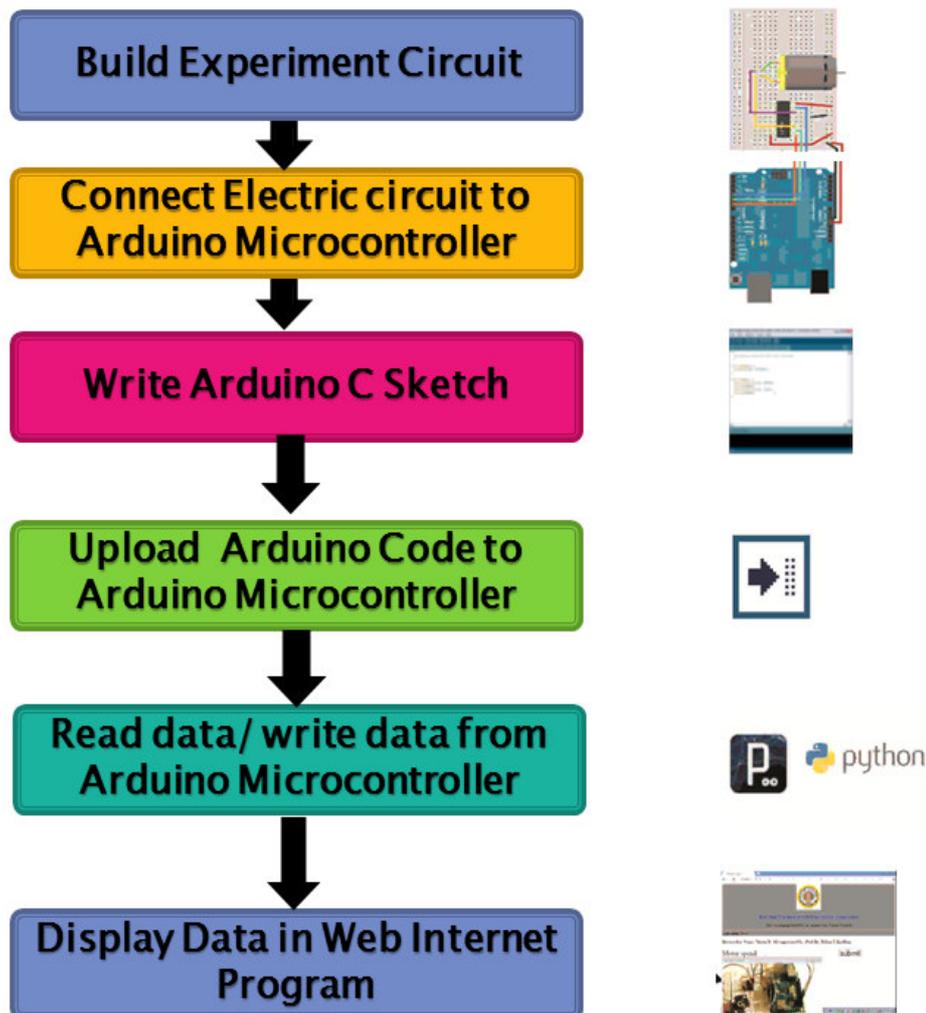


Figure 8. Schematic diagram of python program with Arduino Microcontroller

11. Experimental work:

(Controlling a DC motor using Arduino, Python and PHP)

a. Objective of the Experiment:

The aim of this experiment is to show the capability of Arduino in controlling the speed of a DC motor via internet in UOB-iLab project, and the method of using python program to read and write data to Arduino platform using USB port by converting the results from Binary to text file.

b. Hardware Equipment:

- Arduino Uno board
- Breadboard
- DC motor (5 Volt)
- L293D motor driver
- Wires

c. Hardware configuration:

We need to configure to correctly wire the motor and the motor driver IC. We need to plug the L293D motor driver first, in the middle of our breadboard. We can start by connecting the power for this integrated circuit; we connect pins 8 and 9 to the 5V of the Arduino board, and the pin 5 to the ground pin of the Arduino board.

Then, there are 3 inputs pin we need to connect, and 2 output pins. The output part is easy, we just need to connect the two output pins to the terminals of the DC motor. The output pins we want to use are pins 3 and 6.

The first input pin to connect is pin number 1, which is called the Enable pin. This is the pin we use to set the motor on and off, and to change the speed of the motor. We connect this pin to the pin number 6 of the

Arduino board.

Finally, we want to connect pins number 2 and 7 of the L293D to pins number 4 and 5 of the Arduino board. These pins will be used to change the direction of the motor.

d. Software requirements:

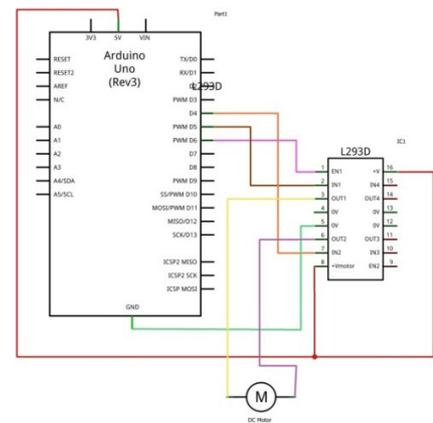
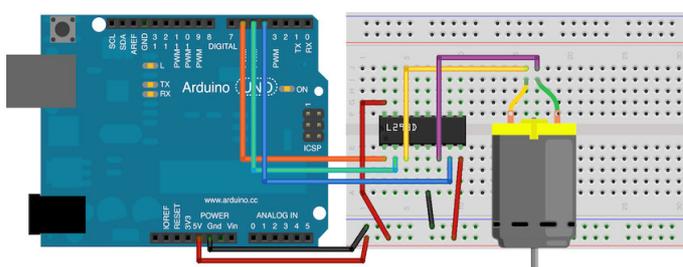
- Arduino C IDE
- Python (with pyserial library)
- PHP
- HTML
- JavaScript

e. Procedure

This complete schematic of experiment is shown in Figure 9:

Figure 9. Schematic diagram for experiment (Controlling a DC motor using Arduino, Python and PHP)

The L293D circuit had needed to make sure working properly, before doing any remote command operation.



To check that, we will do something quite simple, like making the motor accelerate and decelerate, and we will also make it change direction. This is the script we will adopt and write by Arduino C:

```

1      int motorPinPlus = 4;
2      int motorPinMinus = 5;
3      int motorPinEnable = 6;
4      // Setup
5      void setup()
6      {
7          pinMode(motorPinPlus, OUTPUT);
8          pinMode(motorPinMinus, OUTPUT);
9          pinMode(motorPinEnable, OUTPUT);
10         Serial.begin(9600);
11     }
12     // Loop
13     void loop()
14     {
15         // Accelerate forward
16         for (int i = 100; i < 250; i++)
17         {
18             setMotor(true, i);
19             delay(10);
20         }
21         // Decelerate forward for (int i = 255; i > 100; i--) {
22             setMotor(true, i);
23             delay(10);
24         }
25         // Accelerate reverse
26         for (int i = 100; i < 250; i++){
27             setMotor(false, i);
28             delay(10);
29         }
    
```

```
30 // Decelerate reverse
31 for (int i = 255; i > 100; i--){
32     setMotor(false, i);
33     delay(10);
34 }
35 }
36 // Function to control the dc motor
37 void setMotor(boolean forward, int speed){
38     digitalWrite(motorPinPlus, forward);
39     digitalWrite(motorPinMinus, !forward);
40     analogWrite(motorPinEnable, speed);
41 }
```

The core of this script is the function **setMotor**, and we will also use it in the rest of this project, so let's spend more time on it:

```
1 // Function to control the motor
2 void setMotor(boolean forward, int speed){
3     digitalWrite(motorPinPlus, forward);
4     digitalWrite(motorPinMinus, !forward);
5     analogWrite(motorPinEnable, speed);
6 }
```

It takes two inputs, for the direction and the speed. The first step consists in doing two `digitalWrite()` operations to set the direction of the motor: one pin of the L293D circuit will receive 5V, and the other one will be set at 0V. Finally, we use the `analogWrite()` command on the enable pin as we did in the last project, to change the motor speed using PWM. With this function, it is easy to command the motor in both directions. For example, to accelerate in reverse, we can just do the following:

```
1 // Accelerate reverse
2 for (int i = 100; i < 250; i++){
3     setMotor(false, i);
4     delay(10);}
```

Upload the script above to the Arduino board, and the motor should successively accelerate, decelerate, accelerate in the reverse direction, and decelerate in the reverse direction. If everything is going well at this point, we can move to the next part of this experiment.

At this point, we know how to control the DC motor from the Arduino. Let's now put everything we know together and interface the system we designed in this part to the web. There are three different software parts we have to develop. First, we need to slightly modify the **Arduino sketch** so that it can receive data from the host computer. The second part will be written in **Python**, which will make the interface between the computer and the Arduino board, by reading and writing data using text files. Finally, we will use the **HTML, PHP and JavaScript** languages together to have a nice web interface for the project.

f. Analysis :

The web page can be opened the experiment no.3 that has text button to input the value of speed and has button to send it as in Figure(10).The web page permit us to change the speed of DC motor .

We will show how to send commands to devices and control them remotely, and we will take a simple example: a DC motor. This is a very useful experiment, this is basically nothing but a larger DC motor to command, everything will be done using open-source components, and no proprietary libraries or services, by using Arduino, Python, and some HTML, PHP & JavaScript.

Figure (11) show the architecture of experiment. And the role of python program for converting data by USB Port from (Hex) to (txt) type, and PHP program can read it perfectly.

From the above experiment, we conclude that the Arduino Microcontroller compatible with internet programs such as (HTML- JavaScript, PHP, python, etc.), and it is an appropriate to use it to achieve the (Modifying the value of process variables via internet) on any devices or actuators in iLab-UOB.



Figure 10. DC Motor Control Site

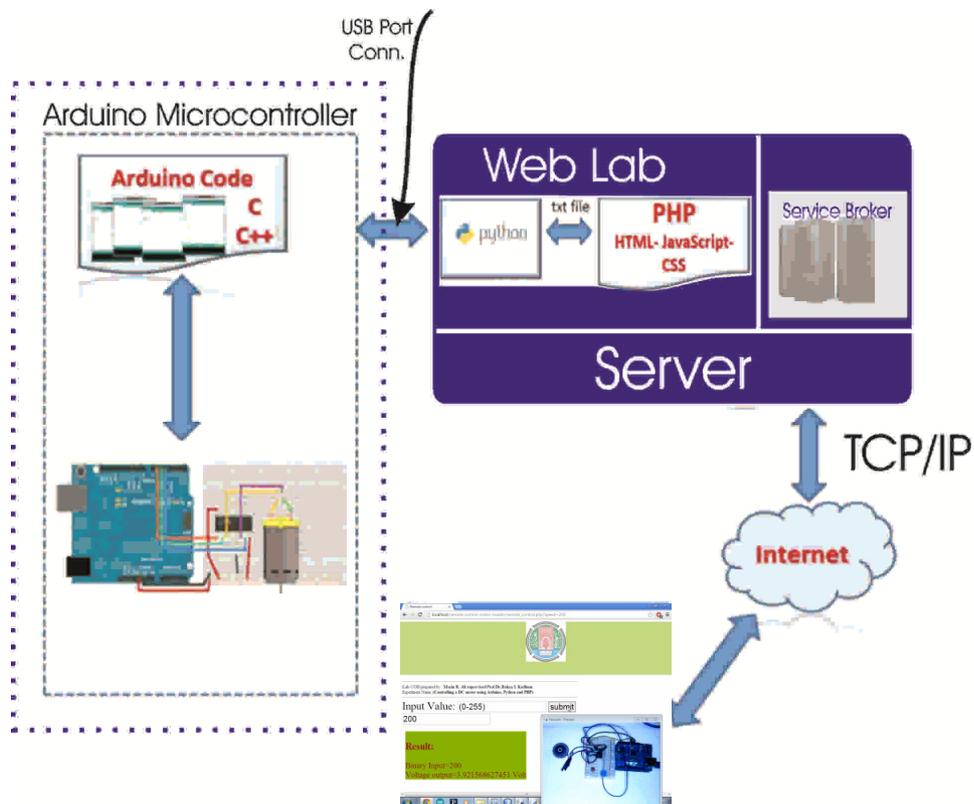


Figure 11. Architecture of experiment (Controlling a DC motor using Arduino, Python and PHP) with UOB-iLab

12. Conclusion

The Arduino Microcontroller can be used as reliable controlling platform for iLab since it compatible with different type of sensor and actuators, the designed web interface using HTML-JAVA- PHP-JavaScript code can be used successfully with proposal Arduino Microcontroller, the developed application for UOB-iLab has several advantages as following:

- a. Can be work at any web interface, and need any runtime software.
- b. Can be work at any operating system platform (windows- Linux- OS) at PC and mobile device.
- c. Can be encrypted easily to prevent the information from spy and hackers.
- d. Easily for development new design of UOB-iLab using the available server like (Apache or IIS).

From the experiment we see the (Python program) one of is the best program especially at reading data from USB interface with microcontroller and has the ability to convert data to different form which we need (txt , csv , xcl...etc.) And work as Synchronous with web programs, and have small code compare with others, and available for all major platforms (Windows, MacOS , Linux ,Unix ,BeOs). Arduino Microcontroller and its family can deals with any device that derive with more than 5 Volt by using relay, and can make calibration to make compatible control., the embedded code that written in (Arduino C IDE) such as (HTML- JavaScript) make the access to Ethernet microcontroller by web is slow, because arduino Microcontroller don't have speed processor.

References

- [1] V. Judson Harward and Jesus A. del Alamo ,The iLab Shared Architecture: AWeb Services Infrastructure to Build Communities of Internet Accessible Laboratories, Vol. 96, No. 6, June 2008, Fellow IEEE,pp 931 - 950.
- [2] A. Fjeldl , “Lab on the Web: Running Real Electronics Experiments Via the Internet “,First edition ,2003.
- [3] Doru Ursutiu and Daniel T. Cotfas, WEB Instruments ,IEEE EDUCON Education Engineering 2010 – The Future of Global Learning Engineering education ,April 14-16,2010, Madrid,SPAIN,pp589-611.
- [4] Awkash Agrawal and Sanjay Srivastava,WebLab:A Generic Architecture for Remote Labrotaries,15th International Conference on Advanced Computing and Communications ,June 2007, pp.301-306.
- [5] Simon Monk, 30 Arduino Projects for Evil Genius,pix ,First edition,2010.
- [6] “Arduino IDE”, Mar-2013, [Online] Available: <http://arduino.cc/en/main/software>
- [7] Massimo Banzi , “Getting Started with Arduino”, First edition , 2009.
- [8] ”Blink experiment by Arduino Microcontrioller”, Mar-2013, [Online] Available: <http://arduino.cc/en/Tutorial/Blink>
- [9] “Arduino and Python”, May-2013, [Online] Available: <http://playground.arduino.cc/interfacing/python>.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:

<http://www.iiste.org>

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: <http://www.iiste.org/journals/> All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: <http://www.iiste.org/book/>

Academic conference: <http://www.iiste.org/conference/upcoming-conferences-call-for-paper/>

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

