Master’s Thesis 2015

Kishan Prajapati (132359)

Process Control and Monitoring using Arduino and Raspberry Pi

Telemark University College

Faculty of Technology

Department of Technology

Kjølnes ring 56

3918 Porsgrunn

http://www.hit.no

Kishan Prajapati

Telemark University College
Faculty of Technology
M.Sc. Programme

MASTER'S THESIS, COURSE CODE FMH606

Student:	**Kishan Prajapati**		
Thesis title:	**Process Control and Monitoring using Arduino and Raspberry Pi**		
Signature:	. . . .		
Number of pages:	<137>		
Keywords:	**Process, logging, Monitoring, Control, PID, Web service, MySQL/Connector, Data Dashboard**		
Supervisor:	Hans-Petter Halvorsen	Sign.:	. . . .
Censor:	<name>	Sign.:	. . . .
External partner:	National Instruments	Sign.:	. . . .
Availability:	<Open/Secret>		

Archive approval (supervisor signature)**:** Sign.: . . . . **Date:** . . . .

Abstract:

In this 21st century, Monitoring and Controlling is being taken as the main entity of any field which can ensure for the effective performance, hence its importance is rising exponentially in industry field as well. Many parameters of the industry should be within the limit. There can be many factors which can bring variations in those variables. This may cease the efficiency of the industry and destruction of industrial equipment as well. Hence, monitoring and evaluation of the variables is very important and control them whenever it is necessary to.

In this thesis, monitoring part is the most emphasized than the controlling part. Hence, the logging system is being more highlighted here. Two different logging systems are developed. The first one includes the Arduino and Raspberry Pi where Arduino is used for interacting with sensors/process and Raspberry Pi as database server (MySQL). This system can be used in a single industry, however can logged and monitor many sensors/processes using multiple number of Arduino connected to single Raspberry Pi. In this case, website is developed in Raspberry Pi for the managing and monitoring purpose which is displayed in a screen in Telemark University College. This system continuously logged the data in certain interval and can be monitored at the same time. And the second logging system is based on web service in LABVIEW. It is built by the use of Arduino and LABVIEW in laptop. MS SQL database is used for logging the data in this case. SQL toolkit of LABVIEW is used for logging purpose whereas web service and Data Dashboard for LABVIEW application is used to make the data available on Tablets and Smartphones that run on the Android and iOS operating system. Besides, PID controller is developed as an Arduino library. It is used for controlling the process (Air Heater).

Hence, this thesis focus on developing en effective and cheap monitoring and controlling system with the use of Arduino and Raspberry Pi. In addition, web based logging, data publication in web, use of data dashboard, development and use of Arduino library, Arduino-LABVIEW interface, data management, etc are included in this thesis.

Table of Contents

Preface

This thesis is about the development of the logging, monitoring and controlling system by the use of Arduino and Raspberry Pi. This project is conducted as the partial fulfilment of Master of Science in System and Control Engineering at the Telemark University College. The thesis focuses on how we can use the Arduino and Raspberry Pi platforms for industrial applications such as Process Control and Monitoring Applications.

This work is very much related as home automation. Several papers and electronic books are studied which made this project successful. The entire tasks are performed by the use of Raspberry Pi, Arduino UNO, LABVIEW 2013, database (MySQL and MS SQL), PHP, etc. The report is based on database management knowledge as well as the website development experience.

This thesis lasted between January 14 and June 3 2015, was carried out at the Telemark University College in collaboration with National Instrument.

Finally, I would like to thanks to my supervisor, Mr. Hans-Petter Halvorsen who always provide guidance throughout the duration of the thesis. Besides, I would like to thanks the University's IT-department for providing necessary hardware and technical assistance for the thesis.

Porsgrunn, 03- June 2015

Kishan Prajapati

Overview of tables and figures

List of figures

List of Tables

Part I: OVERVIEW

It includes the introduction of the thesis and description of the problem or tasks need to be completed.

1 Introduction

Monitoring and Control is the main entity of the any field which can ensure for effective performance, hence its importance is rising exponentially in this modern era. There are millions of industries in all over the world, where different parameters and measures are to be placed within the limit. Variations on these values may lead to ceasing of performance or even the destruction of the equipments. Hence, those are to be monitor in real time and control whenever it is needed.

Process Control and Monitoring system is developed to monitor the process value and control the values on needed without human interface. Thus, it can be defined as a mechanism removing as much human interaction as technically possible and desirable in various domestic processes and replacing them with programmed electronic systems. For the development of such system, Raspberry Pi is used as the main node which is used for both monitoring and controlling purposes and Arduino can be used to collect the data as it is designed to interact with the physical world. The system is placed in a network so that different devices and components can communicate and interact among each other or with end-users or other entities in the network. Thus, the ‘‘Internet of Things’’ – IoT, can be introduced here along with (Vujovic´ & Maksimovic, 2015). In other words, different devices and the appliances in the industry, as a system, being connected to the Internet, can be controlled remotely or continuously monitored.

Arduino, a microcontroller board, and Raspberry Pi, a fully functional mini-computer, are both cheap solutions for harnessing the Internet of Things at industry. Unlike your regular computer, both devices are very good at reading the world around them as they both include plenty of inputs and outputs for sensory add-ons to test light, temperature, humidity and more.

Besides, PID control is introduced and developed in Arduino as a library which can be easily implemented. This PID control is used for the controlling process (Air heater in this thesis). Hence, this thesis focus on the collecting the data from the process and monitor it using different application (website, data dashboard) and control it.

2 Problem Description

This thesis has certain fixed aim which is briefly introduced in this section. To achieve the aim of the thesis, it is divided in different tasks which are described in following sub-section of this chapter 2. Some main sub tasks may be as listed below.

1) Logging process data from Arduino to database.
2) Developing a data monitoring and managing application, website in this thesis.
3) Developing PID controller in Arduino and implement it as a library.
4) Data publication using web service to access data from dashboard in LABVIEW.

Among all, data monitoring and management is be taken as the most important task in this thesis. In this task, Arduino read the value of process and save in database. And data monitoring and management system is used for monitoring and managing data.

2.1 System Description

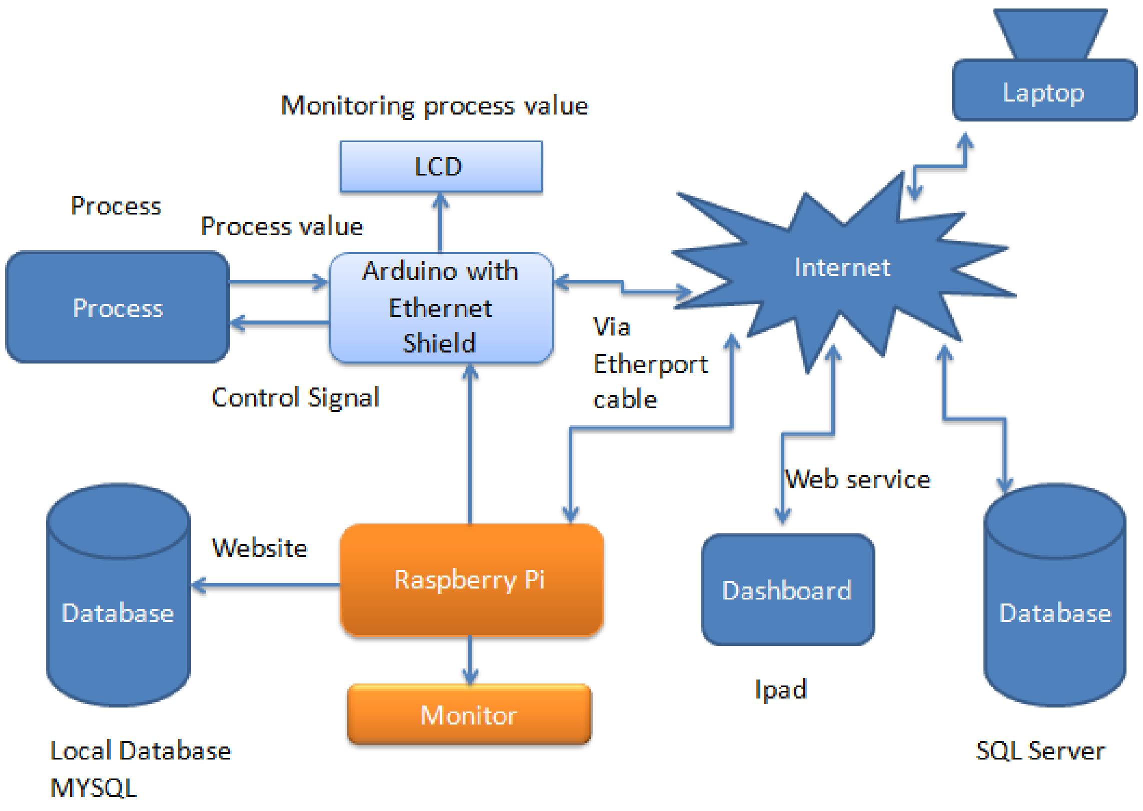


Figure 2-1: System Overview

The Figure 2-1 shows the system overview of the thesis which consists of different units to achieve the objectives of the thesis.

In the system, the main controlling and manipulating device can be taken as the Raspberry Pi as all the logical and mathematical tasks are perform in it. The monitoring application as well as the controlling application is developed in Raspberry Pi as it is the only computer in the system to do the tasks. The laptop in the system is just to monitor the website developed in Raspberry Pi through network. All the necessary applications like Arduino IDE, databases, Apache, PHP and others are installed in the Raspberry Pi. The website is also developed in it for monitoring purpose using PHP and HTML. And the next important device is taken as Arduino which is connected to the physical sensors; process in the system. Arduino consists of lot of input/output ports to interact with outside world. In this system, Arduino is connected to process which reads all the required information and value and it is logged in the local database (MySQL) in the Raspberry Pi. Arduino is attached with its one of the shield (Ethernet shield in this case) and the shield is connected to internet via Ethernet cable. Similarly, Raspberry Pi is connected to internet using Ethernet cable.

For controlling process, PID controller is developed using Arduino in Raspberry Pi and made it as a library so that it will be easy to use. The PID controller provides the control signal according to the error which is difference between process value and set point. It is iteration process which reduces the difference between the process value and set point.

And next is the use of web service using LABVIEW to share the process value to the internet so that it can be access by any clients from anywhere through internet. In this thesis, data dashboard is used for monitoring process data. Data dashboard is an application which is run in windows 8 or IOS and Android mobile.

2.2 Data logging

The main and first task of the thesis is to log the data which is read from the process to a local database (MySQL) with timestamp.

Arduino is physical connector which interacts with the external world to get data. In this system, it is connected to a process with the use of its I/O port and read the value of the process. MySQL/Connector is used for the connection of Arduino and database directly without an intermediate medium. It is nothing, just a library which makes connection between them. But, it is very important that both the Arduino and Raspberry Pi (database server) should be in network. Hence, both of them are connected to internet using Ethernet cable. And Arduino and Raspberry Pi are physically connected just during uploading the Arduino code to Arduino which is coded in Raspberry pi.

In the thesis, there are three scenarios for the logging of process/sensor data which are discussed as follow.

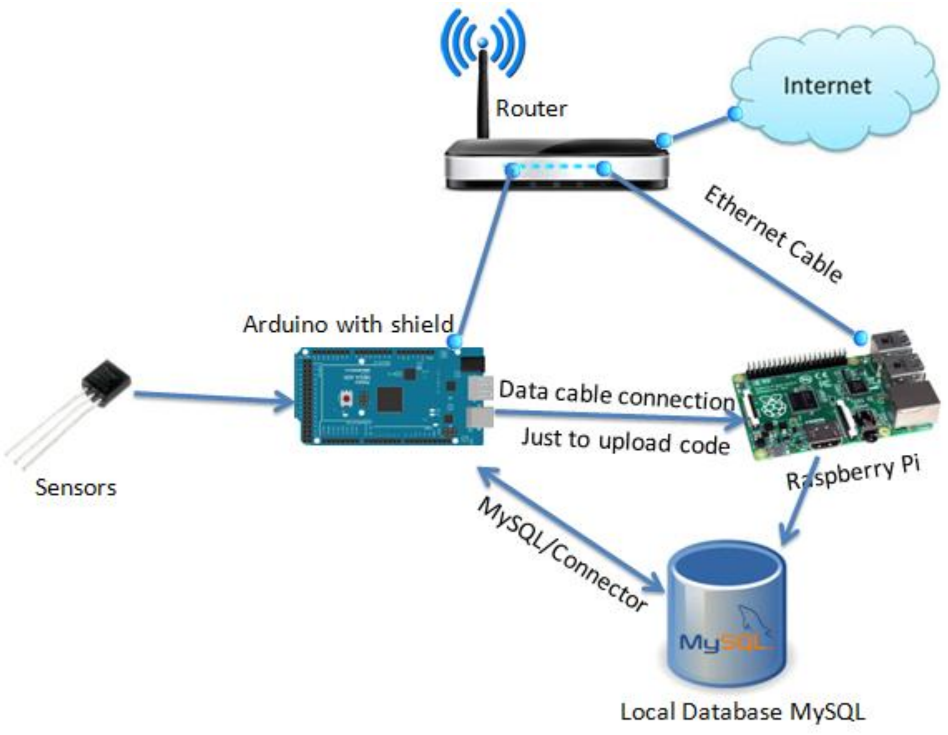


Figure 2-2: Logging process data to the database (Scenario 1)

Figure 2-2 shows the first scenario where a single Arduino is connected to the Raspberry pi. This is used for the logging the data from a single process.

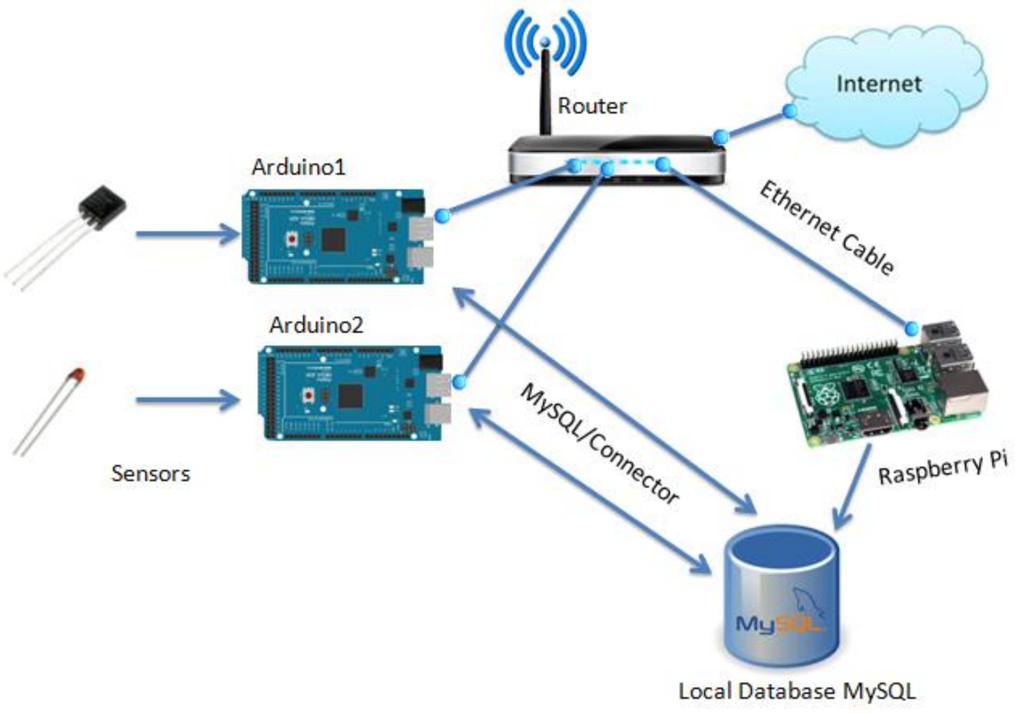


Figure 2-3: Logging process data to the database (Scenario 2).

The second scenario consists of multiple Arduinos connected to the Raspberry Pi and single sensor is connected to each Arduino. Hence, this system can read and log the data of multiple numbers of process or sensors. The scenario is shown in Figure 2-3.

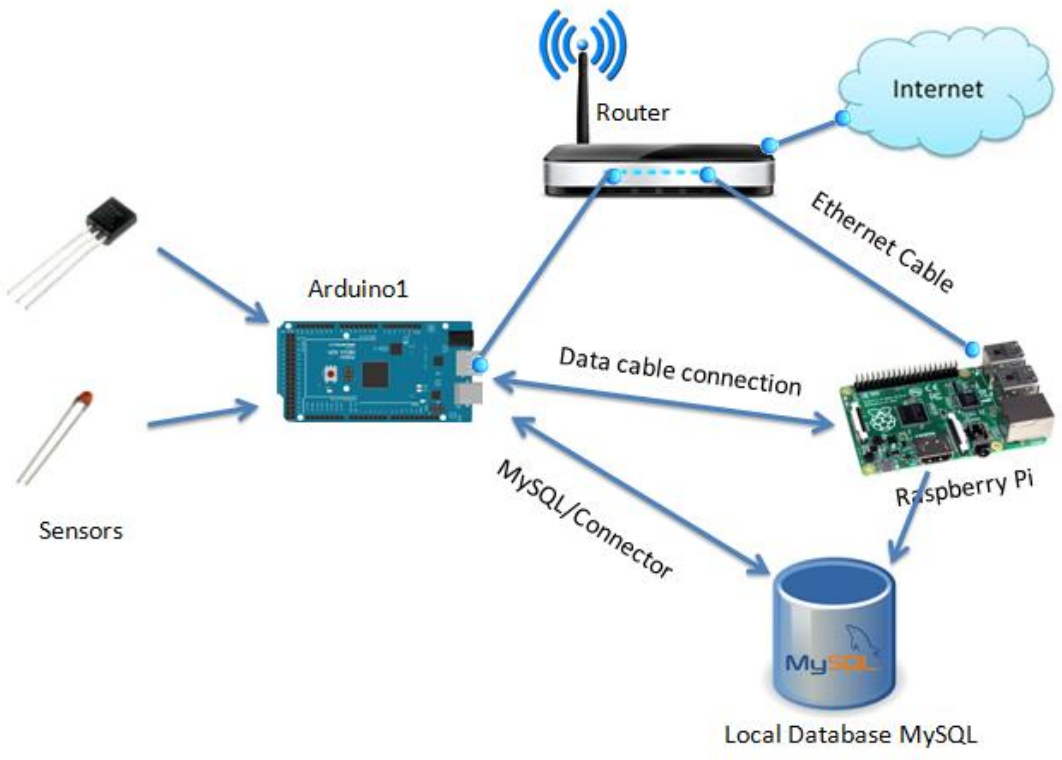


Figure 2-4: Logging process data to the database (Scenario 3).

This scenario is the best one as multiple numbers of sensors are connected to a single Arduino and it is shown in Figure 2-4. With the use of single Arduino and Raspberry Pi, it is possible to log the data of one than one sensor/process. The maximum number of sensors that can connect to an Arduino is limited as there is limited number of I/O ports in the Arduino board.

2.3 Data Monitoring and Managing

The second main task of the thesis is to monitor and manage the data which is stored in database. In this thesis, it is most important portions and the block diagram for this task is shown in Figure 2-5.

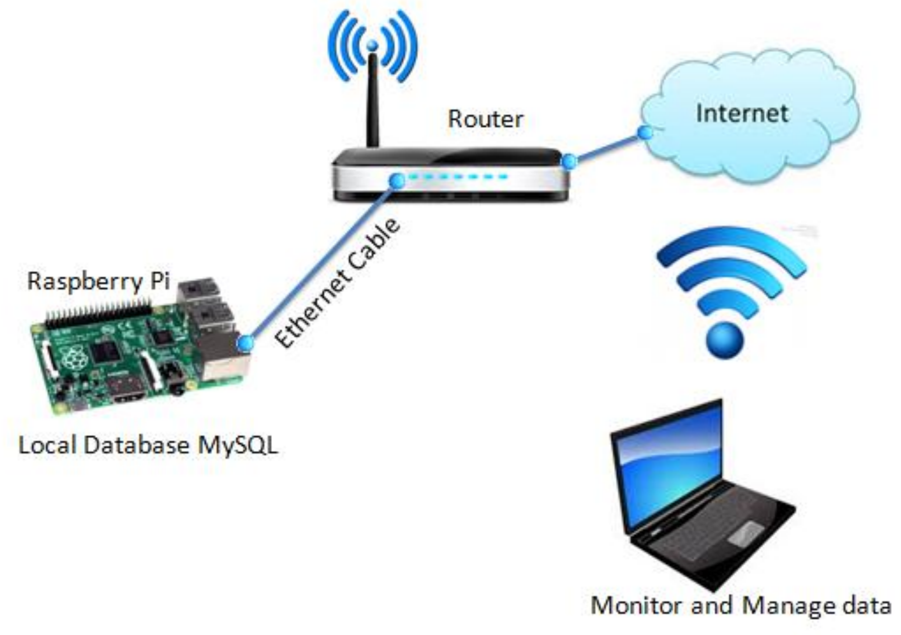


Figure 2-5: Block diagram for data monitoring and management

Website is developed in Raspberry Pi using PHP which is used for data presentation and data management as well. It is already mentioned that database is built in Raspberry Pi, so the website use the local database. For data presentation among the outside world, trends is used.

Data management includes the adding, deleting and editing devices and tags. It helps to provides information about the number of devices and tags used and their inter-connection. We can easily get information about devices and their connection with others devices or tags. Besides, alarm configuration and management is also done via the website. All the active alarms are displayed and give an option for acknowledging it. For security, login facility is provided in the website and users are assigned as different level. For different levels of user, different authorities or access to website is provided.

Figure 2-6 shows the architecture for the data monitoring and management system. The architecture shows that the first layer is device layer. It consists of devices needed for interfacing the tags and processes which includes Arduino, Ethernet shield, etc. The second layer is logging software which read the value from process and saves in database and the sketch is run in Arduino. Process and database can be treated as another layer.

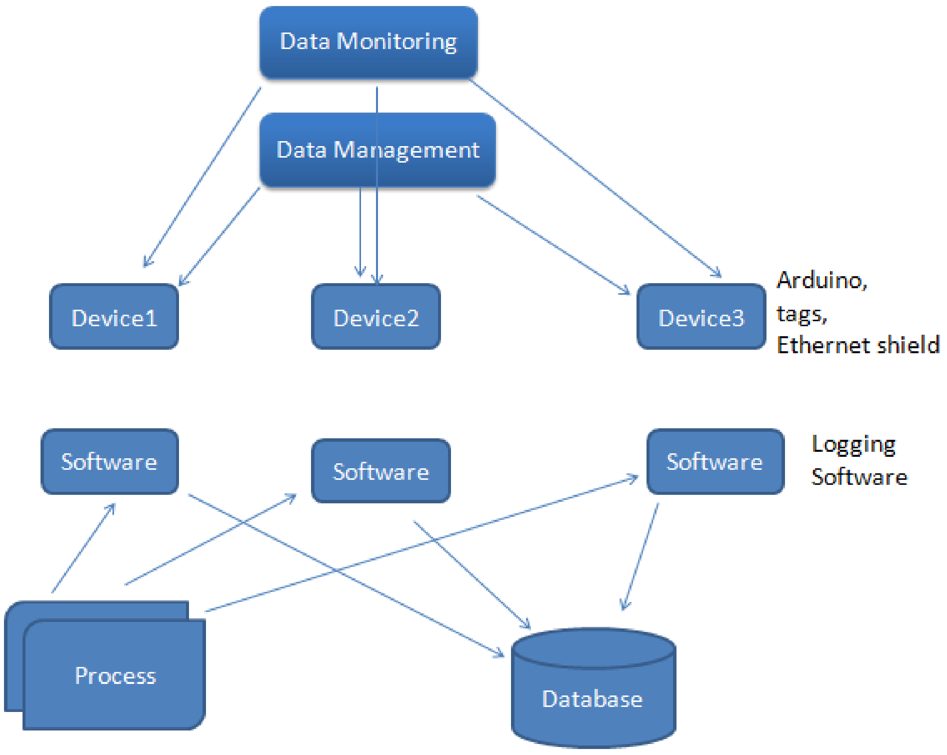


Figure 2-6: Architecture of the Monitoring and Management System

2.4 Web Based Data logging and Data Dashboard

Data or database cannot be directly accessed by client via internet. There may be security problem or compatible issues which makes the direct connection between data and client over internet near to impossible as shown in Figure 2-7.

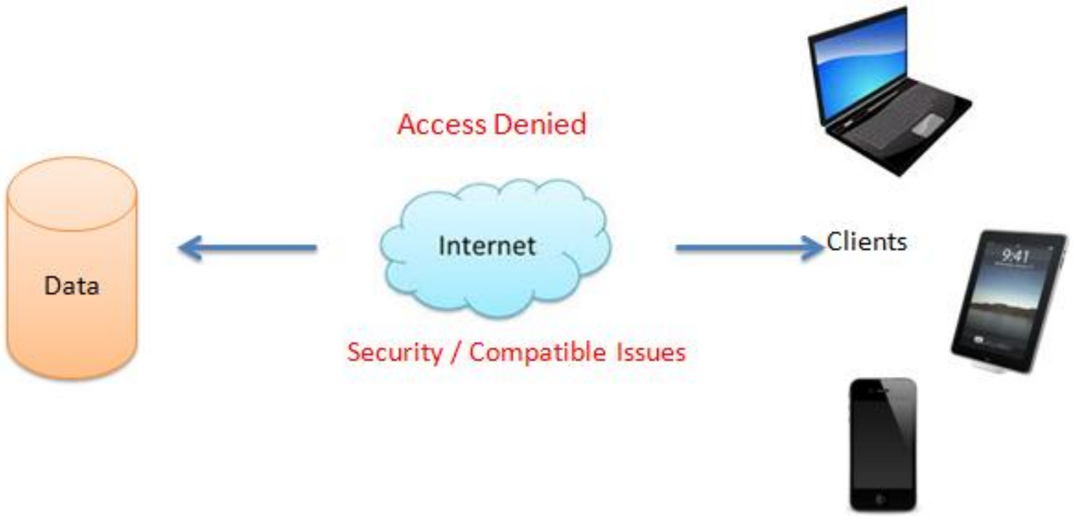


Figure 2-7: Problem in Direct Connection

One of the solutions to share the data in internet so that it can be access by other client is web service. Web service makes the connection between the data/database and clients via internet.

In this thesis, information from the sensors/process is stored to the by the use of SQL toolkit in LABVIEW. It is a simple and easy to use SQL toolkit. The Toolkit consists of 4 simple VIs for Database Communication. It is easy to include in the functions palette in LABVIEW. The overview for the logging system using LABVIEW is shown in Figure 2-8.

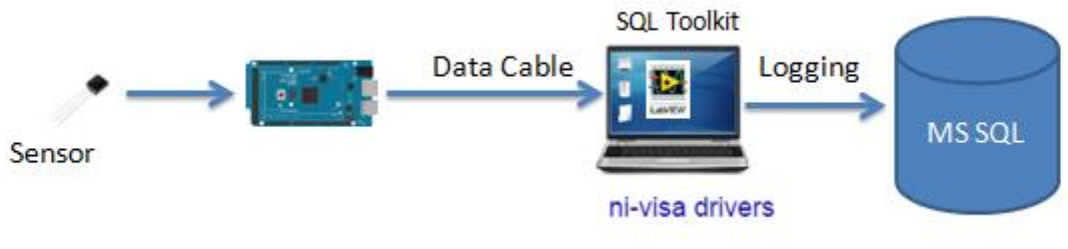


Figure 2-8: Data Logging using LABVIEW SQL Toolkit

Finally, web service is use to publish the data stored in database to the web so that it can be access by others through internet. It is shown in Figure 2-9.

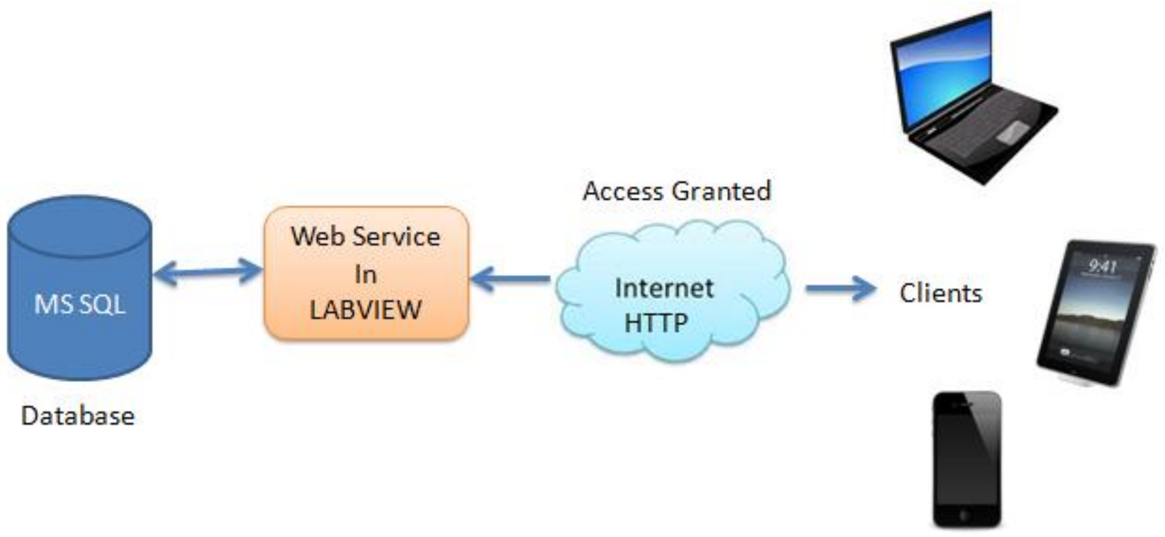


Figure 2-9: Web service Connecting data and clients

Data Dashboard is an application developed for android, IOS and windows 8 which is used to monitor the data using web service. In this thesis, web service will be developed in LABVIEW.

2.5 Arduino PID Control

PID control is a widely-used method to achieve and maintain a process set point. The PID control equation may be expressed in various ways, but a general formulation is shown in Equation 1.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dx} \qquad \text{Equation 1}$$

Where e(t) is the difference between the current value of the process variable (temperature) and the desired set point, usually written as e(t) = (Value-Set Point); $\int_0^t e(\Gamma)d\Gamma$ is the summation of previous Error values; and $\frac{de(t)}{dt}$ is the time rate of change of the process variable being controlled, or of the error itself. The proportional coefficient K_p, the integral coefficient K_i, and the derivative coefficient K_d are gain coefficients which tune the PID equation to the particular process being controlled. Drive is the total control effort (often a voltage or current) applied to actuators (heater) to achieve and hold the set point.

Here, PID control is developed in Arduino and it is implemented as a library so that it can be used by other users as well. In this thesis, it is used to control a process (Air heater).

3 Objectives

In this thesis, one of the prototype platforms which enable end-user programming in order to build a complete open-source process control and monitoring system is considered. This makes devices, components and appliances web aware so that managing and monitoring can be done by web browser (Website[1]). The main objectives of this thesis are to log process data, monitoring it via website or data dashboard, data management and control.

In order to achieve those objectives following sub topics also considered.

1) Literature review on Raspberry Pi, Arduino, databases, database connector, etc
2) Study about different scenario for logging data and select the best one.
3) Log the process data in a database connecting Arduino with MySQL using Connector/Arduino[2].
4) Developed a website using PHP in Raspberry Pi for data monitoring and data management.
5) Networking the devices so that website can be accessed from anywhere via internet.
6) Developed PID controller on Arduino for controlling data and implement as an Arduino library.
7) Web based data logging is performed to MS SQL and web service is created in LABVIEW which access data from MS SQL and publish which makes possible to monitor using dashboard in LABVIEW.
8) Documentation of the whole work.

[1] Website is created on Raspberry Pi but can be accessed from any computer via internet.
[2] database connector made specifically for the Arduino and MySQL

4 Some Remaining Tasks

The task description for this thesis contains altogether 11 tasks, some of tasks are relatively less important. During inception of thesis, we (me and my supervisor) had agreed upon first completing important tasks because we were quite sure of insufficiency of time for completing all listed task .The important tasks were shortlisted by my supervisor. These shortlisted tasks are completed by the end of thesis but few less important tasks are not done.

Part II: Theory

It includes the theory for all the devices and professional tools for completing the thesis.

5 Raspberry Pi

5.1 Overview

Raspberry Pi seems to be new in the world and many people really don't know what the Raspberry Pi is. Raspberry Pi can be defined as a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It’s capable of doing everything you would expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. It is great bonding with Arduino and can do a lot with Arduino.

5.2 Exploring the Raspberry Pi Board

There are two models of Raspberry Pi, model A and model B. These two are bit similar with few advance features on model B compared to model A. Model B has 512 MB RAM, two USB port where as Model A has 256 MB RAM and just a USB port. Besides, Model B has Ethernet port while Model A does not.

The overview of the Raspberry Pi Model B is shown in Figure 5-1.

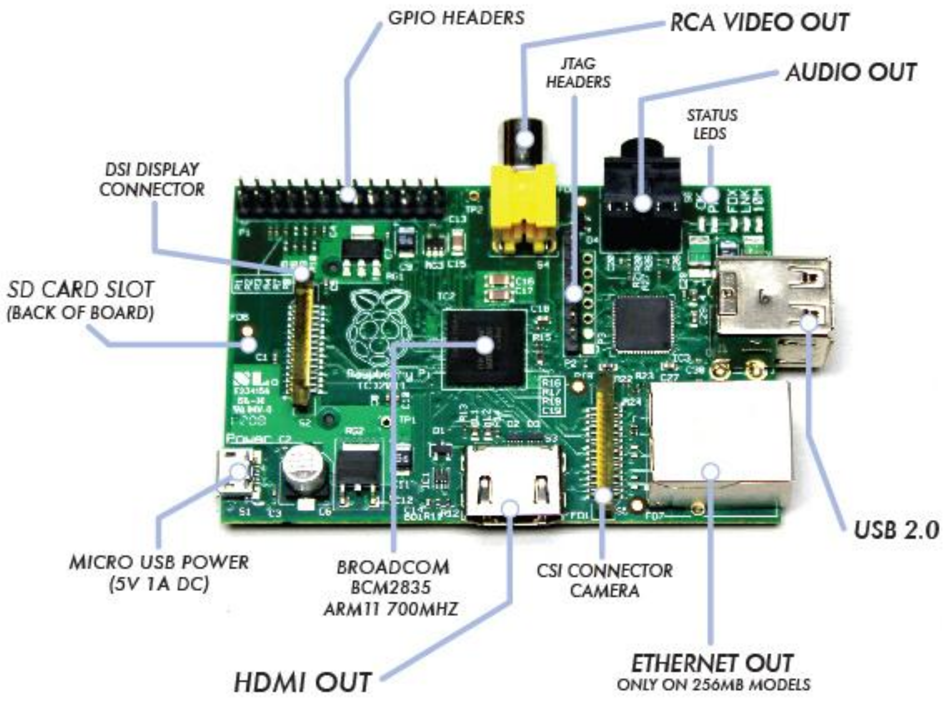

GPIO HEADERS
RCA VIDEO OUT
JTAG HEADERS
AUDIO OUT
STATUS LEDS
DSI DISPLAY CONNECTOR
SD CARD SLOT (BACK OF BOARD)
MICRO USB POWER (5V 1A DC)
BROADCOM BCM2835 ARM11 700MHZ
CSI CONNECTOR CAMERA
USB 2.0
ETHERNET OUT ONLY ON 256MB MODELS
HDMI OUT

Figure 5-1: Raspberry Pi Model B(Sanders, 2013)

Different components of the Pi are named in the Figure 5-1 and brief description on each component is given in following sections.

5.2.1 SD Card Slot

Raspberry Pi doesn’t have the real hard drive as in laptop and computer, SD card is taken as solid state drive (SSD) which is used to install operating system and all others software and store everything. This card is needed to insert into the slot for using the Raspberry Pi. SD card may be 2GB, 4GB or 16GB.

5.2.2 Micro USB Power

The power port is a 5V micro-USB input and supply should be exactly 5v as it doesn’t have onboard power regulator**.** So, power supply shouldn’t exceed than 5V.

5.2.3 HDMI Out

This output port is used to connect the Raspberry Pi with a monitor via HDMI (High Definition Multimedia Interface). Hence, any screen or TV can be connected to it which consists of HDMI port.

5.2.4 Ethernet and USB port

Both the Ethernet port and USB port on Model B are supplied via the onboard LAN9512 chip. It is a high-speed USB 2.0 hub with a 10/100 Ethernet controller (Donat, 2014). USB ports are used to connect the inputs (keyboard, mouse). Almost everything that can connect to computer via USB also can connect with Raspberry Pi.

5.2.5 RCA Video Out and Audio Out

Audio and RCA video jacks are present on the board for audio and video out.
The Raspberry Pi does support sound over its HDMI output, but there is a standard 3.5-mm audio jack to plug in headphones but USM mikes may work or not.

For video, the RCA jack sends video to any connected RCA video device.

5.2.6 GPIO Headers(Pins)

GPIO pins stands for general purpose of input output pins. These pins are used to connect any number of physical extensions with the Raspberry Pi. Raspberry Pi has pre-installed libraries that allow us to access the pins using programming languages like C, C++ or python.

5.2.7 ChIPs (Broadcom)

The most important component in a Raspberry Pi is chip that is Broadcom which is placed at the middle of the board. The chip consists of ARM11 processor running at 700 MHz and a Videocore4 GPU and can be over clocked to at least 800 MHz without a problem.

5.3 Raspberry Pi 2 Model B

Recently, Raspberry Pi 2 Model B has been lunched recently which Broadcom BCM2836 ARM Cortex-A7 Quad Core Processor has powered Single Board Computer running at 900MHz, 1GB RAM and 4 Quad USB ports. It is the advanced version of Model B and is 6 times faster than Model B Raspberry Pi. In addition, it has combined 4-pole jack for connecting your stereo audio out and composite video out and advanced power management. Figure 5-2 shows the top view of the board with labels of some important components (raspberrypi.org, 2015a).

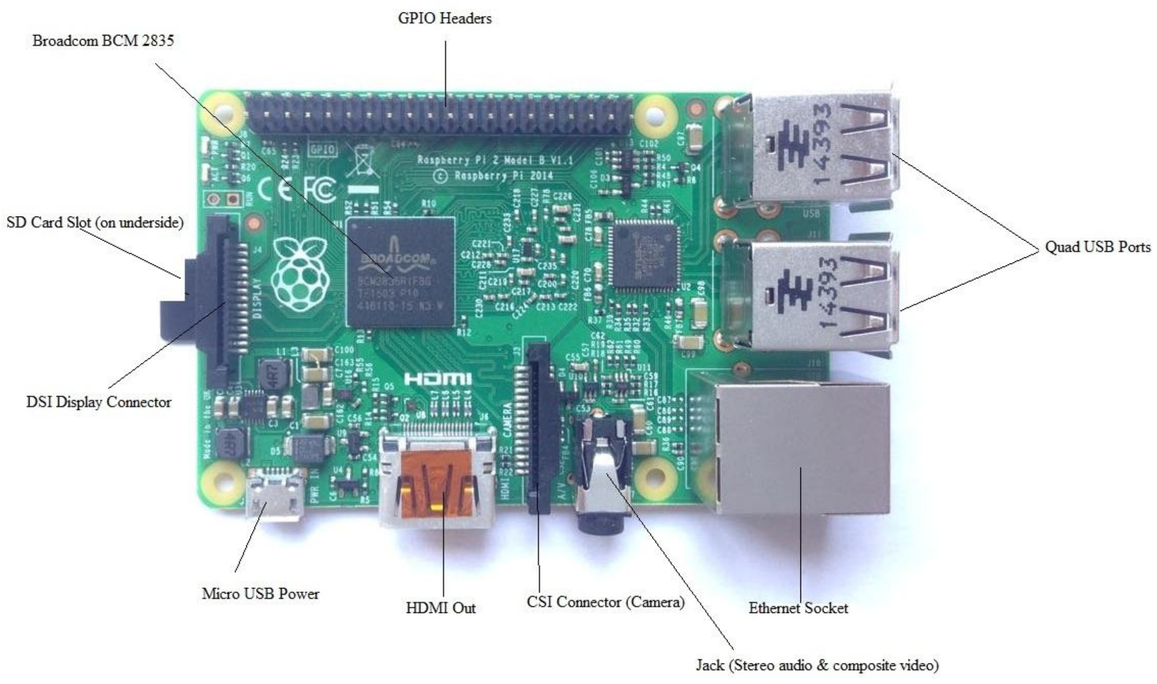


Figure 5-2: Raspberry Pi 2 Model B

5.4 Hardware Required for Raspberry Pi

Raspberry Pi can’t start alone, it needs many others peripherals (hardware). There is brief description of the hardware requirements in the following section (Bates, 2014).

5.4.1 Power Supply

As mentioned already in above theory portion, Raspberry Pi needs 5V power supply. If supply exceeds 5V then it can't guaranteed to work properly. And the power supply also need to supply at least 500 milliamps (mA), and preferably more like 1 amp (A). If the supply is 500 mA or less, it is likely to have the mal-function of keyboard and mouse. It is not good idea to power the Raspberry Pi from USB port of computer and hub as they mostly provide current less than required. Hence, the Raspberry Pi requires a Micro-USB connection which is capable of supplying at least 700 mA (or 0.7 A) at 5V.

5.4.2 Storage

A separate hardware is required for the storage purpose in Raspberry Pi. For this, SD card is used, mostly 4 GB and 8 GB if needed. The operating system and all files are stored in the card. We can buy blank SD card and install operating system or buy a pre-installed one.

5.4.3 Input

External keyboard and mouse are required to provide input to the Raspberry Pi. No any additional software is needed to use the keyboard and mouse.

5.4.4 Monitor

We can use monitor or TV with HDMI port or DVI inputs as the screen for the Raspberry Pi. For DVI inputs, HDMI-to-DVI converters are required which can be finding easily in a market. Monitor is most important for the Raspberry Pi as it is the only way to see what we have done on it.

5.4.5 Network

As in laptop or computer, we can access to internet and network in Raspberry Pi as well. For that, we can use wired Ethernet connection which is easier option or Wi-Fi module to access Wi-Fi in the Raspberry Pi.

6 Arduino

Arduino is a popular open-source physical computing platform which is very common for communicating with the physical world. It is very easy and effective to use Arduino to sense and control most of the sensors and equipment. It is based on a simple microcontroller board, and a development environment for writing software for the board. It is very popular for developing electronically thesis.

Arduino can be used to develop interactive objects which take inputs from a variety of switches or sensors, and control varieties of lights, motors, and other physical outputs and Arduino projects can be stand-alone, or they can communicate with software running on other computer (arduino.cc, 2015b). Arduino can be divided into two part, physical programmable circuit board and software (Arduino IDE[3]).

6.1 Arduino Board

It is the physical programmable circuit board which is connected to real world. It can get inputs from varieties of sensors and switches and on return, can control others physical devices. Program is uploaded in the board via IDE in a computer using USB cable. There are different types of Arduino board available in the market such as Arduino UNO, Arduino Mega 2560, Arduino Mega ADK, etc. Arduino UNO is shown in Figure 6-1.

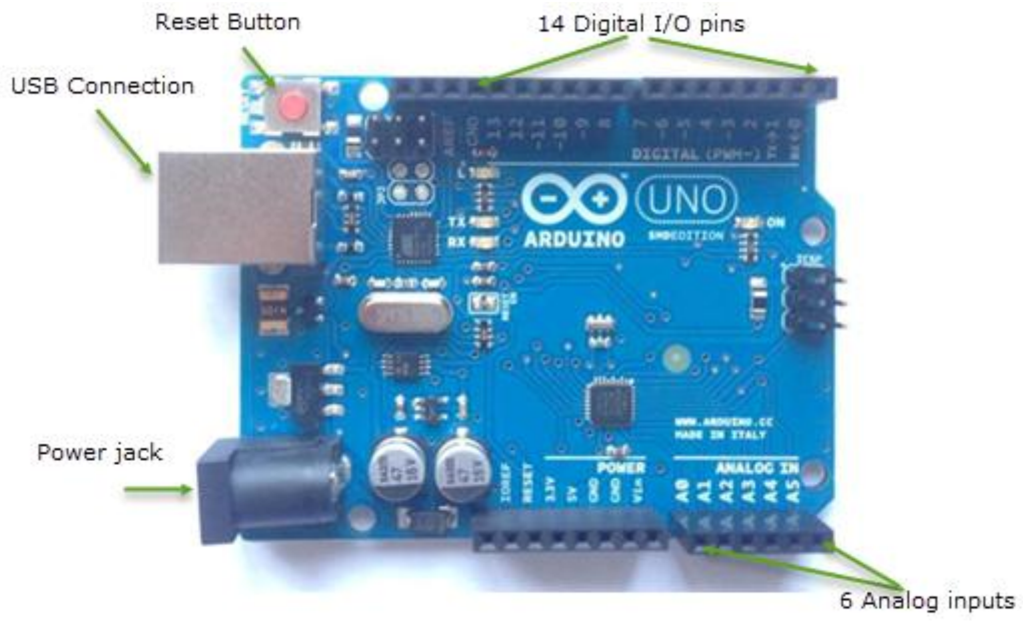


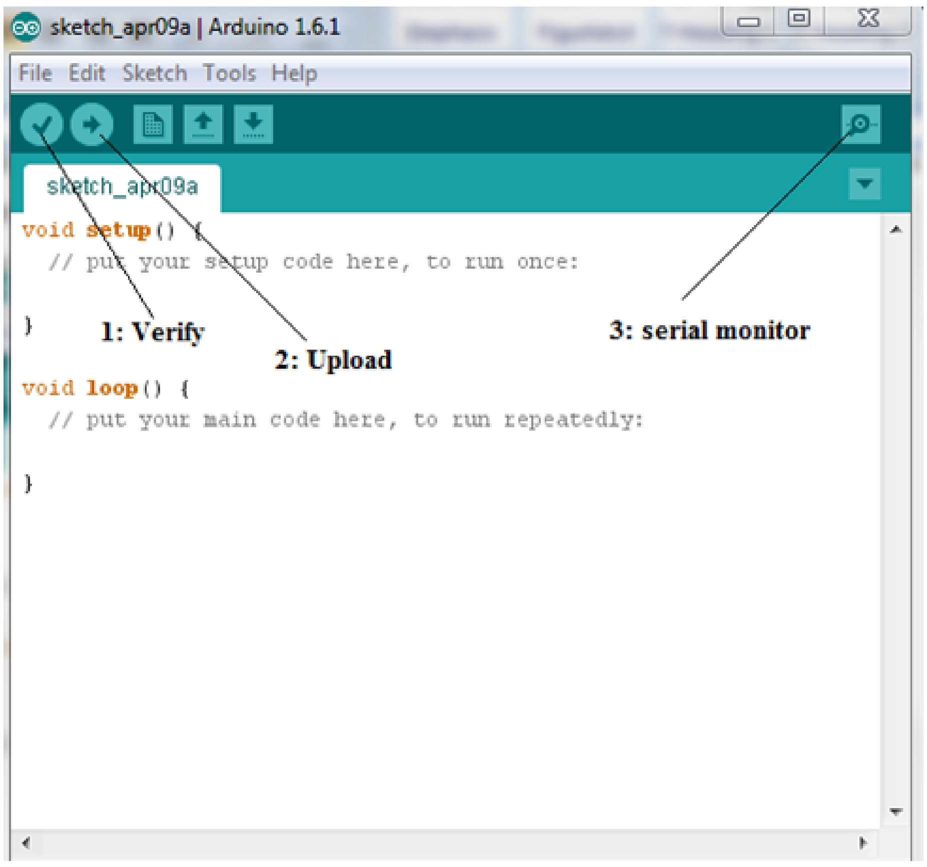
Figure 6-1: Arduino UNO

[3] Integrated Development Environment

It consists of 14 digital I/O[4] pins, 6 analog inputs, a USB connection, a power jack, an ICSP header, a reset button and a 16 MHz crystal oscillator. Among 14 digital I/O pins, 6 pins can be used as PWM outputs which is indicated by ~. It contains everything needed to support the microcontroller, we just need to connect it to a computer with a USB cable or just power it with an AC-to-DC adapter or battery to get started.

6.2 Arduino IDE

Arduino IDE is software which is needed to install in a computer. And, it uses a simplified version of C++, making it easier to learn to program.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Figure 6-2: Arduino IDE

The Figure 6-2 illustrates the three main steps that carried out in the IDE software. First of all, program is coded and clicks the “verify” button. This is compiling the code and if there is no any error then ”upload” button should be clicked. This is done to upload the program to the Arduino board. And, finally out is display in the serial monitor which can be viewed by clicking on serial monitor button.

[4] input/output

As mentioned in Figure 6-2, code which needs to run just a once is kept under ”setup” function and which needs to be run repeatedly in certain interval of time should be kept under ”loop” function. For defining the interval, delay(x) function is used where x is time interval defined in millisecond.

6.3 Internet Connection on Arduino

Arduino, itself cannot connect to internet it needs either Ethernet shield or Wi-Fi shield. Either one of the shield should be mounted over the Arduino board for the internet connection.

6.3.1 Arduino Ethernet Shield

Arduino Ethernet Shield allows the Arduino board to connect with internet using Ethernet library. The library can serve as either a server accepting incoming connections or a client making outgoing ones. The library supports up to four concurrent connection (incoming or outgoing or a combination)(arduino.cc, 2015a). For details information, you can reference to this link, Ethernet Library.

For internet connection, first the shield should mounted over the Arduino and the shield should be connected to the internet using standard Ethernet cable then provide a power. The next is network setting. For that MAC (Media Access Control) address and fixed IP address should be assigned to the Ethernet shield using the function, Ethernet.begin(). This function initializes the Ethernet library and network settings. If the network has DHCP enabled then Ethernet.begin (MAC_address) is enough because IP address is assigned automatically. But in case of static IP, fixed static IP address should be added as the argument of the function as Ethernet.begin (MAC_address, IP). Here in the function, MAC_address is MAC address of the Ethernet shield which is array of 6 bytes where as IP is IP address of the Ethernet shield and it is array of 4 bytes.

Nowadays, MAC address of Ethernet shield is printed in its back. So we should use that during internet connection. But, there is no MAC address printed in old Ethernet shield. In such case, unique array of 6 byte should be assigned manually.

The Arduino Ethernet shield is shown in Figure 6-3 and MAC address of the device is printed at back side on white sticker.

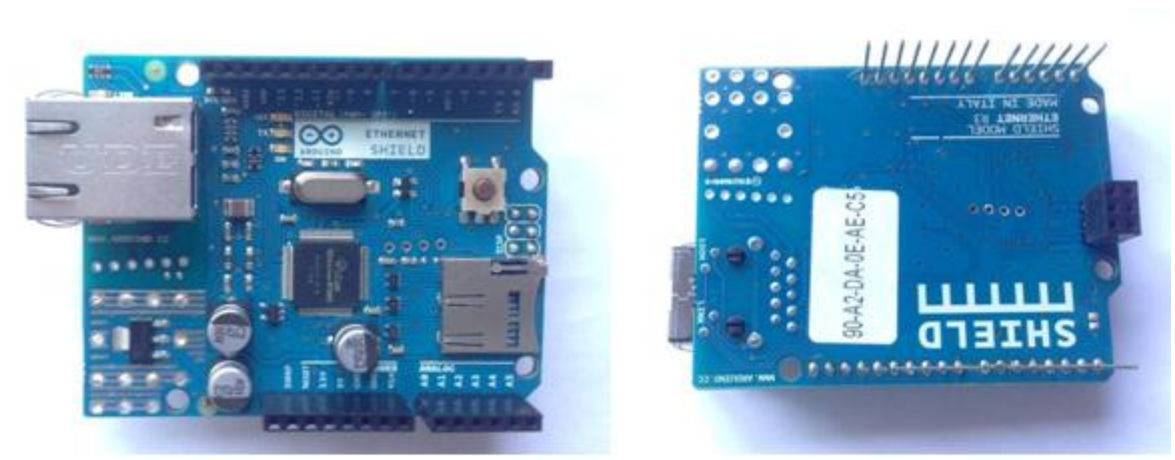


Figure 6-3: Ethernet shield

6.3.2 Arduino Wi-Fi Shield

The next option for the internet connection on Arduino is the use of Arduino Wi-Fi Shield using the 802.11 wireless specifications (Wi-Fi). Similarly as Ethernet shield, Wi-Fi shield is also mounted over the Arduino board. It uses Wi-Fi library to connect Arduino to internet. The Wi-Fi Shield can connect to encrypted networks that use either WPA2 Personal or WEP encryption and can connect to open networks as well but the network must broadcast its SSID for the Wi-Fi Shield to be able to connect(arduino.cc, 2015a).

Wi-Fi Shield is not used in this thesis so for more information and use of Wi-Fi Shield, you can get information about the Shield in the link given as Arduino Wi-Fi Shield.

Figure 6-4 shows the diagram of Arduino Wi-Fi Shield for the Arduino.

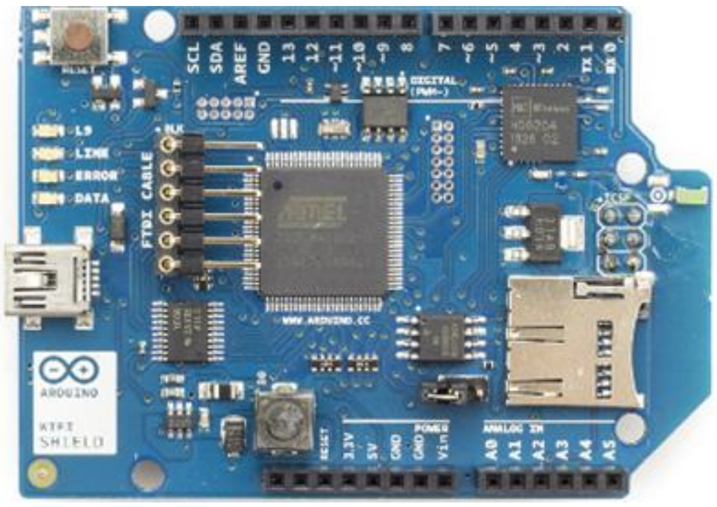


Figure 6-4: Arduino WiFi Shield

7 Air Heater

Air Heater is lab station developed for the purpose of effectively demonstrating and/or learning basic PID control skills in Telemark University College (Haugen, Fjelddalen, Dunia, & Edgar, 2007). Air heater is taken as the process of an industry for monitoring and controlling purpose. The Figure 7-1 shows the Air heater.

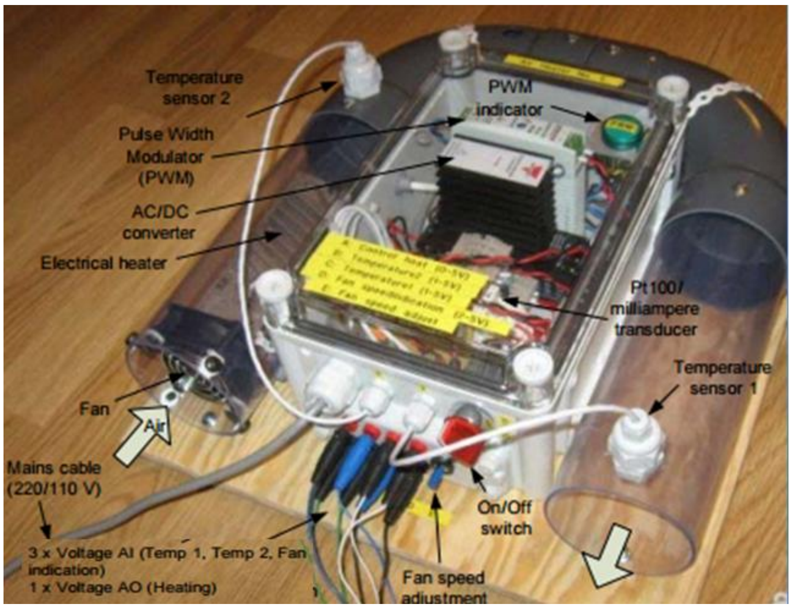


Figure 7-1: Air Heater(Haugen et al., 2007)

Some features of the Air Heater are explained as follow.

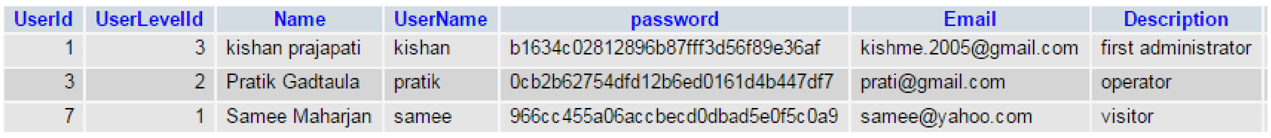
1) Fan: A fan is used to make air flow within a tube and it is operated manually with a knob. The fan position is measured by a voltage signal across the two terminals which is in the range 2 - 5 V (min, max fan speed). The normal fan speed is defined to be the maximum speed.
2) Heater: The air inside the tube is heated by the help of electric heater. The power supplied to tube is controlled by an external signal ranging from 0-5 V which must be applied between two terminals. The output of this controller is used to control Pulse Width Modulator (PWM) which connect/disconnect the main voltage to the heater.
3) Temperature sensor: In an air heater two PT100 temperature elements are available which have been equally calibrated. We can use any of these, but we have used Temperature sensor 2 for the thesis. The sensor signals are available as voltage signals at their respective terminals. The range is 1 - 5 V, and this voltage range corresponds to the temperature range 20^{o}C - 50^{o}C (with a linear relation). The normal sensor position is defined to be the outermost position in the tube.

4) Analog I/O device: Any analog I/O device supporting the voltage ranges defined above can be used in measurement and control applications. In this thesis, Arduino is used as analog I/O device.

8 Database

Database is the organized structured collection of data for future use. Databases can be stored in a computer or in web and can be managed by a program named as database management system (DMS). Data is stored in such an organized way so that it can be easy to store and to extract in future. There are many ways to organize the computerized data; these ways are known as database models. One of the most common and popular models is relational model and the program which use this model is known as relational database management system (RDMS) which is discussed in details in following sections (ucl.ac.uk, 2000).

Computer based database consists of one or more tables with rows and column to store the data. The number of tables will vary according to the size of data to be stored. Tables are 2-dimentional array which consists of rows and columns. Each column contains various types of attributes or also named as field name, while each row corresponds to a single record. The Figure 8-1 illustrates the tables for storing the information about the users. It consists of field name or attributes as “UserId”, ” UserLevelId”, ”Name”, “UserName”, “Password”, “Email” and “Description” are the attributes or field names of the table “user” whereas the remaining three rows are records of the three users.



UserId	UserLevelId	Name	UserName	password	Email	Description
1	3	kishan prajapati	kishan	b1634c02812896b87fff3d56f89e36af	kishme.2005@gmail.com	first administrator
3	2	Pratik Gadtaula	pratik	0cb2b62754dfd12b6ed0161d4b447df7	prati@gmail.com	operator
7	1	Samee Maharjan	samee	966cc455a06accbecd0dbad5e0f5c0a9	samee@yahoo.com	visitor

Figure 8-1: Table to Store Information of Users

8.1 Database model

Database model can be defined as a way to manage the computerized data so that it will be easy and effective to extract in future. The quality of database performances is mostly depends on the database design and model used. There are so many types of database models such as relational model, flat model, objects database model, etc. Among all, relational model is the most popular and common database model.

Relational Model

A common and powerful method for organizing data for computerization is the relational data model. In this method, records are saved in different tables according to the category of information and different tables are inter-related. Tables are connected to each other so that during the extraction of data, they can be joined and extract all the required information from those tables. In this method, there will be many tables but small in size.

Hence, it will be easy to understand and easy to save the information. The most important advantage of relational model is; it is easy to manage and get required information though they are stored in different tables.

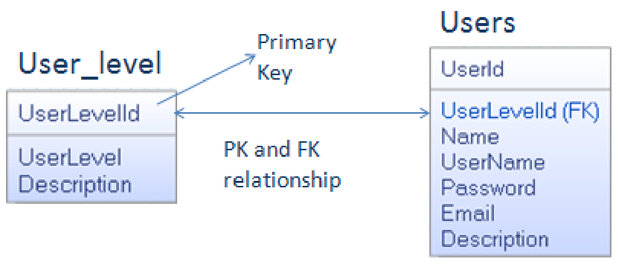


Figure 8-2: Two Tables Linked by PK-FK

Figure 8-2 shows the relation between two tables; User_level and Users. UserLevelId as primary key in User_level is related to the UserLevelId as FK (Foreign Key) in Users. In this case, all the information from both tables can be extracted by joining them using one SQL. This is most powerful technique for processing and accessing the data in the database. Primary key must be integer type, auto-increment, not NULL and must be unique as well where as FK must be the same data type of its primary key (PK).

Primary and Foreign keys

Every row of a table should be unique in relational database so one or more columns of the table should be designed as primary key. Primary key is also known as unique identifier. There must not be two identical so at least one primary key is introduced in a table. And Foreign keys are columns in a table which provide a link to another table (ucl.ac.uk, 2000).

In Figure 8-2, UserLevelId and UserId is primary key of table User_level and Users respectively where as UserLevelId in Users is foreign key as this field is used to connect two tables.

8.2 Database Design

Before creating physical database for any application or project, it is very important to think and collect the information about the full requirement of the project, types of information need to be stored, purposes of the project and future enhancement. According to the data collected, database should be design so that it will be easy to stored different category of information in different tables and link the tables which have related information. We should be able to create simple and small tables and inter-connect each other rather than creating a large table with much information. This creates difficulty during accessing data from the database. Database can be design using software CA Erwin.

CA Erwin

CA Erwin is a software tool for data modeling of custom developed information systems, including databases of transactional systems and data marts. Erwin’s data modeling engine is based upon the IDEF1X method, although it now supports diagrams displayed with information engineering notation as well. (wikipedia.org, 2014)

A design of a table includes the table name, attributes and indication of primary key. If any two tables are needed to connect, foreign key should be introduced. In this design, it is possible to show the connection among the tables.

8.3 MySQL

MySQL is the most popular open source relational database management system and it is named after the name of co-founder Michael Widenius’s daughter (wikipedia.org, 2015). Database consists of structured collection of data in large amount. The size of data may be varies in large extend. There will be low size of data in small project but there will be vast amounts of information in a corporate network. Whatever the size of information, data should be managed (add, access and process). For that database management system such as MySQL is needed.

MySQL databases are relational database in nature; hence it stores the data in many tables rather than in a big table. The database structures are organized in a file form to increase the speed. The features of the MySQL such as database, table, columns, rows, view, etc provides flexibility in programming world (mysql, 2015). Here, view is a virtual table which is generated by the combination of two or more tables which are discussed in later section.

And MySQL is open source so anyone can use it and modify it. It can be downloading directly from internet and use it. Besides, it is very easy, fast, reliable and scalable and can be use in any desktop, laptop and in Raspberry Pi as well.

Some important parts of MySQL are described in brief in following sections.

8.3.1 SQL

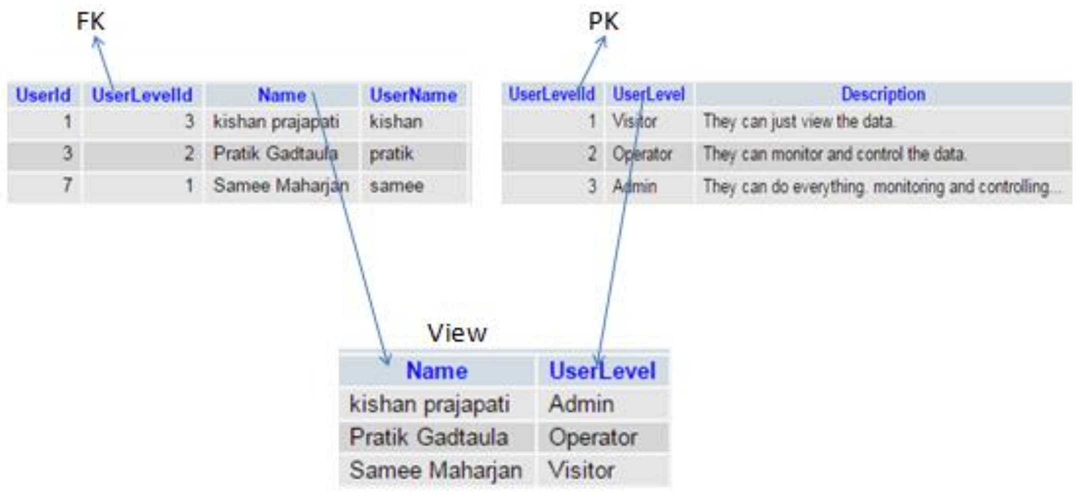
SQL stands for Structured Query Language which is used to add, access and process the information stored in the database. There are different types of queries which are used for different purposes. The Table 8-1: List of Popular Queries with their Function shows the list of most used queries with their function.

Table 8-1: List of Popular Queries with their Function

S.N	Queries	Funtion
1	select	Extracts information from a database
2	delete	Deletes information from a database
3	update	updates information in a database
4	insert	Saves information in a database
5	create	Creates table in a database
6	join	Joins two or more tables in a database

8.3.2 View

View is a virtual table which is created by joining two or more tables. This table includes some specific column of the connected tables. Any number of columns of any table can be combined with any number of columns of any table to create a virtual table but there is a condition to be fulfilled; those tables should be linked by PK and FK. The concept of view will be clear from the Figure 8-3.



UserId	UserLevelId	Name	UserName
1	3	kishan prajapati	kishan
3	2	Pratik Gadtaula	pratik
7	1	Samee Maharjan	samee

UserLevelId	UserLevel	Description
1	Visitor	They can just view the data.
2	Operator	They can monitor and control the data.
3	Admin	They can do everything, monitoring and controlling...

Name	UserLevel
kishan prajapati	Admin
Pratik Gadtaula	Operator
Samee Maharjan	Visitor

Figure 8-3: View Table Creating from Two Tables

8.3.3 SQL Trigger

A SQL trigger is a set of SQL statements stored in the database catalog. A SQL trigger is executed or fired whenever an event associated with a table occurs such as insert, update or delete. A SQL trigger is a special type of stored procedure. It is special because it is not called directly like a stored procedure. The main difference between a trigger and a stored procedure is that a trigger is called automatically when a data modification event is made against a table whereas a stored procedure must be called explicitly (mysqltutorial, 2015).

9 Phpmyadmin

Phpmyadmin is manager for whole MySQL server and a single database. It bring MySQL to a browser hence it is easy to manage database in phpmyadmin. Phpmyadmin can do lot of things in MySQL but some of the main and important things are listed as follow.

1) Browse and drop databases, tables, views, columns and indexes[5].
2) Import and Export database
3) Create, update, delete, rename and alter databases, tables, columns and indexes.
4) Manage MySQL users and privileges
5) Create and read dumps[6] of tables
6) Test and create queries.

The requirements for the installation of phpmyadmin are listed as follow.

1) Web server: Phpmyadmin's interface is based entirely in a browser, it is needed to install web server like apache.
2) PHP: PHP 5.3.0 or newer version is needed.
3) Web browser: To access phpMyAdmin you need a web browser with cookies and JavaScript enabled.
4) Database: phpMyAdmin supports MySQL-compatible databases (MySQL 5.5 or newer) (phpmyadmin.net, 2014).

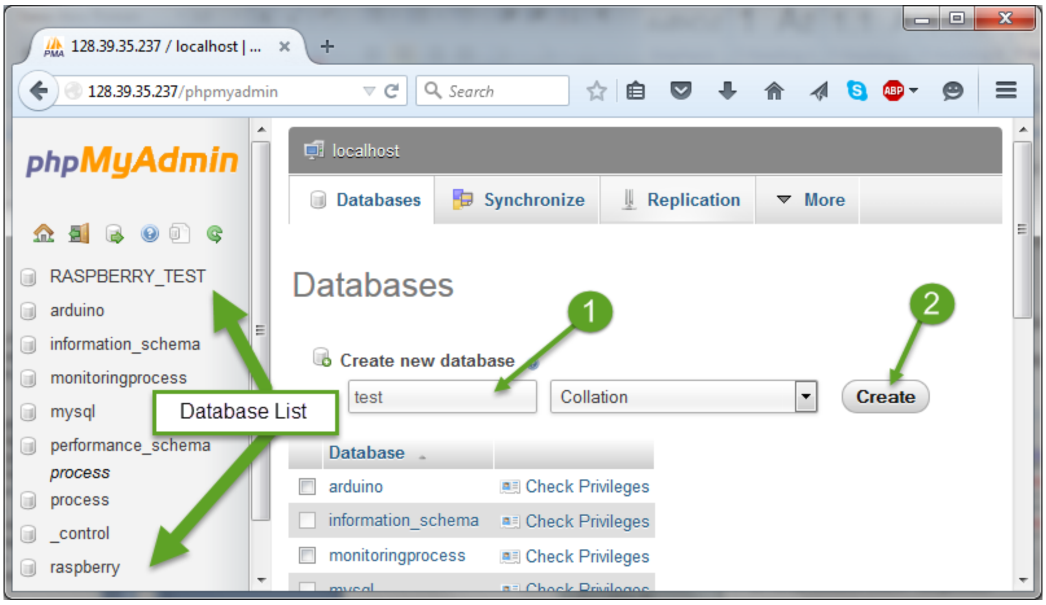


Figure 9-1: Graphical Representation of phpmyadmin in Web Browser

[5] data structure that improves the speed of data retrieval operations on a database table.

[6] dump contains a record of the table structure and/or the data from a database and is usually in the form of a list of SQL statements

The Figure 9-1 shows the graphical representation of phpmyadmin and two steps for creating a new database. As shown in Figure 9-1, phpmyadmin can be access using the link http://localhost/phpmyadmin/ or http://128.39.35.237/phpmyadmin where “128.39.35.237” is the IP address of the MySQL server. And a table can be create by the use of CREATE statement as shown in Figure 9-2 where database name is “test” and table name is “test_table”.

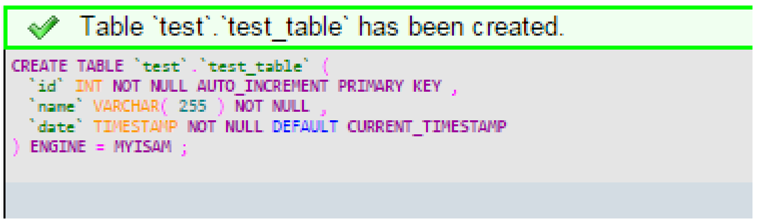


Table `test`.`test_table` has been created.

```
CREATE TABLE `test` . `test_table` (
  `id` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
  `name` VARCHAR( 255 ) NOT NULL ,
  `date` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE = MYISAM ;
```

Figure 9-2: Creating Table in Phpmyadmin

INSERT statement is used to insert data into the database whereas SELECT statement is used to extract data from the database.

10 MySQL Connector/Arduino

10.1 Introduction

MySQL Connector/Arduino is a database connector which can be simply defined as the library for the connection between the Arduino and MySQL Server directly without using an intermediate computer or a web-based service. This library allows the users to have direct access to MySQL Server through Arduino projects. Having direct access to a database server means you can store data acquired from your project as well as check values stored in tables on the server (Bell, 2012). Saving data to the database doesn't give the opportunity to just re-use the data in future but also can feed the data to the more complex applications.

MySQL Connector/Arduino implements the MySQL client communication protocol (called a database connector) in a library built for the Arduino platform. Sketches on Arduino written to use the library permit to encode SQL statements to insert data and retrieve data from the database. The protocol for communicating with a MySQL server is well known and documented not only this; it is also specifically designed to be lightweight. Hence, it is being possible to insert data from Arduino to database and retrieve data from database though Arduino has very less memory and limited processing power.

The Arduino must be connected to a network in order to connect with MySQL server. After the Arduino connected to the server, it will be able to communicate with database. Arduino can be connected to internet or be in network with the use the Ethernet shield or Wi-Fi shield as a result it can connect to the database server. The library is compatible with most new Arduino Ethernet, Wi-Fi, and compatible clone shields that support the standard Ethernet library (Charles Bell, 2013).

The Connector/Arduino library allows us to issue queries to the database server in much the same manner as you would through the MySQL client application. We can insert, delete, and update data, call functions, create objects, etc. Issuing SELECT queries are also possible but they incur[7] a bit more thought concerning memory management (Chuck Bell, 2013).

[7] to come into or acquire (some consequence, usually undesirable or injurious)

10.2 Limitations

Arduino has a limited memory so there are some limitations on using this complex library. The Connector/Arduino library is not a small library; it may consume a lot of memory so there is high chance to be out of memory. Though library uses dynamic memory to keep memory use to a minimum, some amount of memory is consumed according to the sketch and depends on how the library is used. When issuing SELECT queries, the query strings must fit into memory and the largest row of a result set must also fit in memory. This is because result sets are read one row at a time and the class uses an internal buffer for building data packets to send to the server. The connector reads one packet-at-a-time and since the Arduino has a limited data size, the combined length of all fields must be less than available memory (Chuck Bell, 2013).

However, if the latest Arduino Due is used, this may not be an issue. But there are other considerations. The following are the known limitations of the Connector/Arduino:

1) Query strings (the SQL statements) must fit into memory. It is suggested that long strings be stored in program memory using PROGMEM.
2) Result sets are read one row at a time and one field at a time.
3) The combined length of a row in a result set must fit into memory (Charles Bell, 2013).

11 PID Controller

In Figure 11-1 a schematic of a system with a PID (Proportional, Integral, and Derivative) controller is shown. The PID controller compares the measured process value with a reference set point value. The difference or error, e, is then processed to calculate a new process input, u. This input will try to adjust the measured process value back to the desired set point.

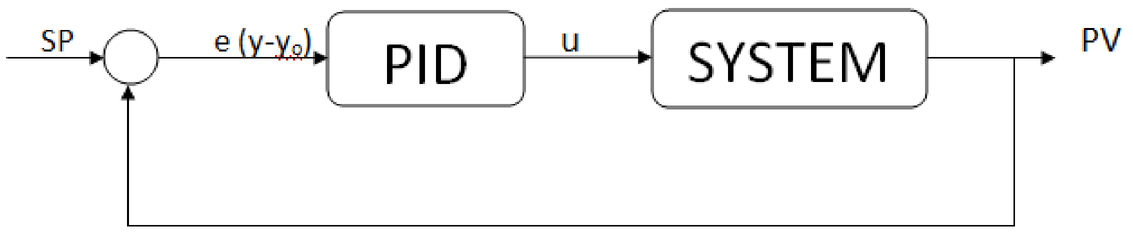


Figure 11-1: Closed Loop System with PID controller

PID is a control algorithm that tries to compensate for characteristics in your system. There are three primary components to think about in a PID control loop. Each component is pre-fixed with a gain constant, and when added together, give you the instantaneous control value that you use to drive your system. Typically, you are generating a voltage to control your system, so each component can be thought of as contributing a particular voltage to your final output. You will have voltage corresponding to the current state of your system (position, temperature, etc) that is called your “Process Variable” or PV. The PV is the value you pass to your PID control loop to tell it about the state of the system. You also have a set point (SP) voltage, corresponding to the state you wish your PV to reach. Basically, you want your PID loop to drive your system so that SP and PV are equal. Third, you have a control voltage u, which corresponds to the instantaneous voltage value you are using to drive your system towards its SP voltage. Your control voltage u can be thought of as what is actually sent to the system to steer it where you want it to go.

The PID algorithm is show in Equation 1. There is a proportional, integral and differential part. The constants K_P, K_i, and K_d are used to set the sign and contribution gain of each part of this equation. e (t) is the proportional “error” corresponding to SP - PV. The variable t corresponds to the current time in our system, and τ is simply a variable of integration. The proportional portion of the equation takes into account how far away our PV is from our SP. The differential part takes into account how fast we are moving (if we move to fast near our SP , we will over shoot), and can be used to reduce the proportional portion if we are moving too fast, or speed us up if we are experiencing resistance despite our proportional contribution. The integral part of the equation takes into account how long we have been off of the set point, contributing more to our output the longer we are missing the SP. This is important because our P and D contributions will typically lead our PV to sag slightly above or below our SP variable (Comnes & Rosa, 2015).

Part III: Implementation

This chapter includes all the exercises done by me to complete the tasks listed in this thesis description.

12 Start up with Raspberry Pi

12.1 Installation of operating system on Raspberry Pi

Raspberry Pi is a small computer; hence operating system (OS) should be installed. As the Raspberry doesn't have hard drive, OS is installed in the external memory. For that, memory card (SD card) is used for the installation of operating system and all the required software and supporting files are stored in the same SD card.

There are different types of operating system but we preferred to talk about NOOBS (New Out of the Box Software) as it is suitable for the beginners. We can either buy a pre-installed SD card or empty SD card. In pre-installed SD card, NOOBS is already copied and ready to boot.

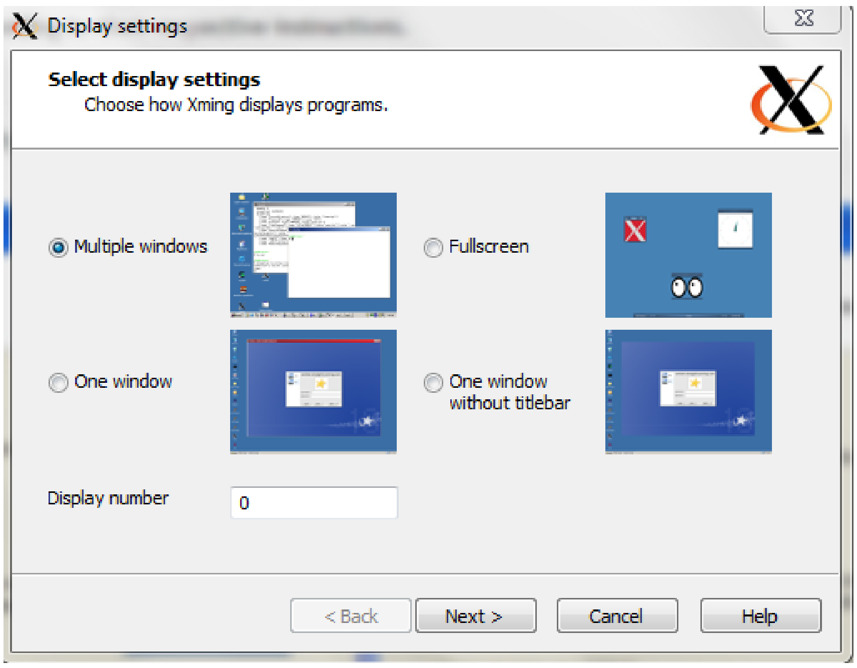
The detail description of installation of operating system in a blank SD card in step wise is given in Appendix B.

12.2 Use of laptop screen, keyboard and mouse for Raspberry Pi

If you don't have suitable screen for the Raspberry Pi, we can use the monitor of a laptop. Besides, we can use keyboard and mouse of laptop as well for Raspberry Pi. For that, two software are needed to install in a laptop which screen is going to use for Raspberry Pi. Installation and configuration of the software is discussed briefly in the following sections.

12.2.1 Xming

This is the first software need to be installed which can be downloaded from the link, Download Xming and install it in the laptop. After completion of installation, run the application called 'XLaunch' and verify that the settings are as shown in three following figures; Figure 12-1, Figure 12-2 and Figure 12-3.

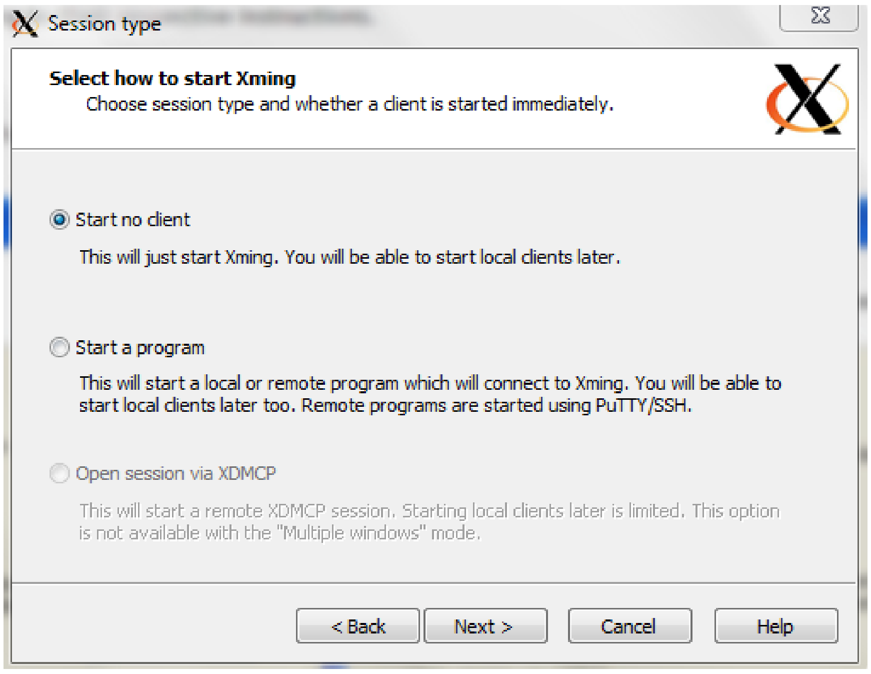


Select display settings
Choose how Xming displays programs.

[✓ Multiple windows] Fullscreen
One window — One window without titlebar

Display number: 0

Figure 12-1: Xming Configuration step 1



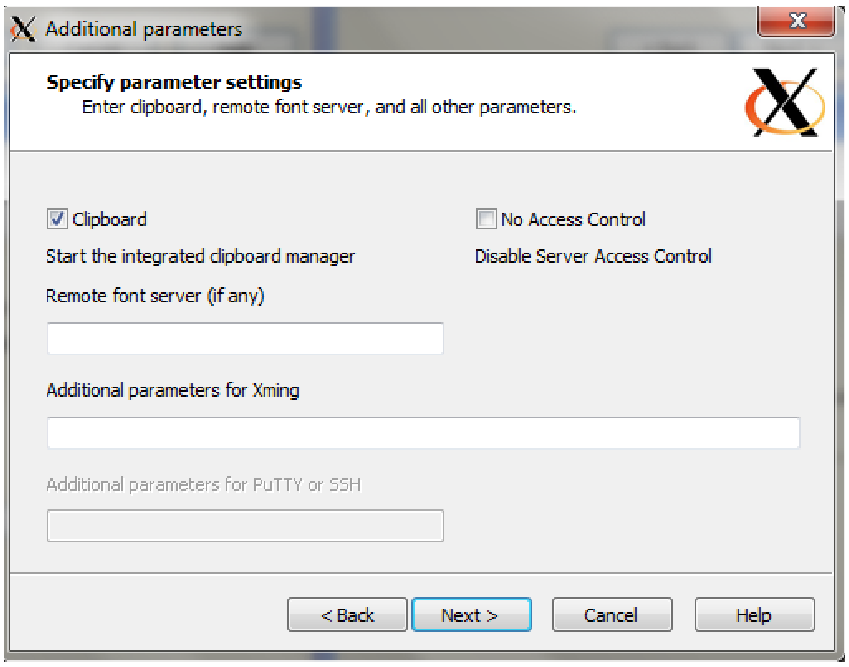
Select how to start Xming
Choose session type and whether a client is started immediately.

[✓ Start no client]
This will just start Xming. You will be able to start local clients later.

Start a program
This will start a local or remote program which will connect to Xming. You will be able to start local clients later too. Remote programs are started using PuTTY/SSH.

Open session via XDMCP
This will start a remote XDMCP session. Starting local clients later is limited. This option is not available with the "Multiple windows" mode.

Figure 12-2: Xming Configuration step 2


Additional parameters

Specify parameter settings
Enter clipboard, remote font server, and all other parameters.

[✓] Clipboard
Start the integrated clipboard manager

[ ] No Access Control
Disable Server Access Control

Remote font server (if any)

Additional parameters for Xming

Additional parameters for PuTTY or SSH

Figure 12-3: Xming Configuration step 3

Finally, click ”Next” button shown in Figure 12-3, it goes to last configuration dialogue box where ”Finish” button is needed to click for the completion of the setting.

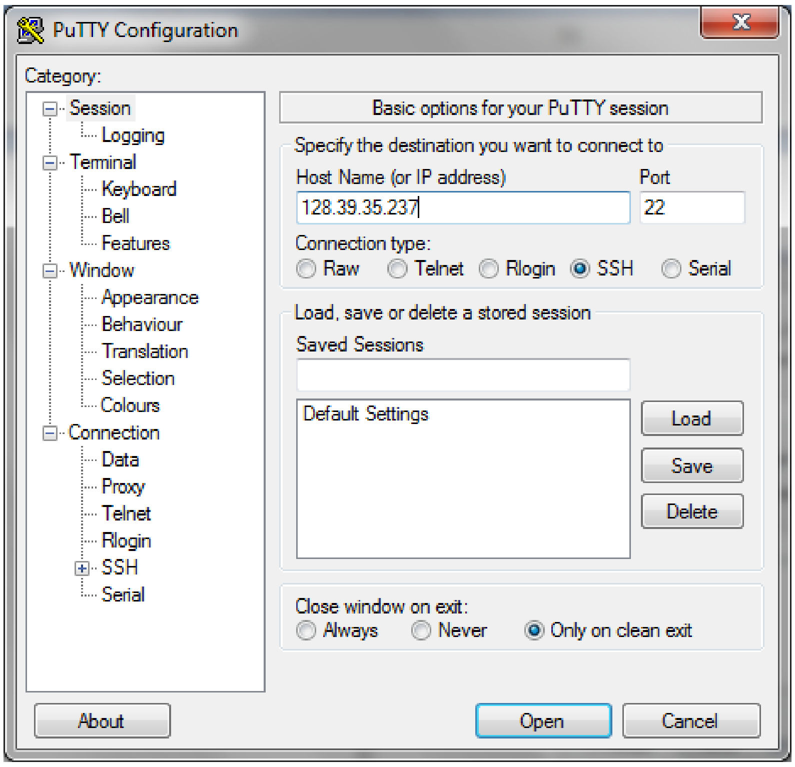
After Completion of configuration, double click the application named as”Xming”.

12.2.2 PuTTY

This is the primary software need to be installed. It can be downloaded in the provided link as follow, Download PuTTy.

As, it is downloaded, it needs to be installed following some few normal steps of installation.

For Configuration, double click the icon of PuTTy after the completion of installation and enter the IP address of Raspberry Pi as shown in Figure 12-4.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 128.39.35.237 Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

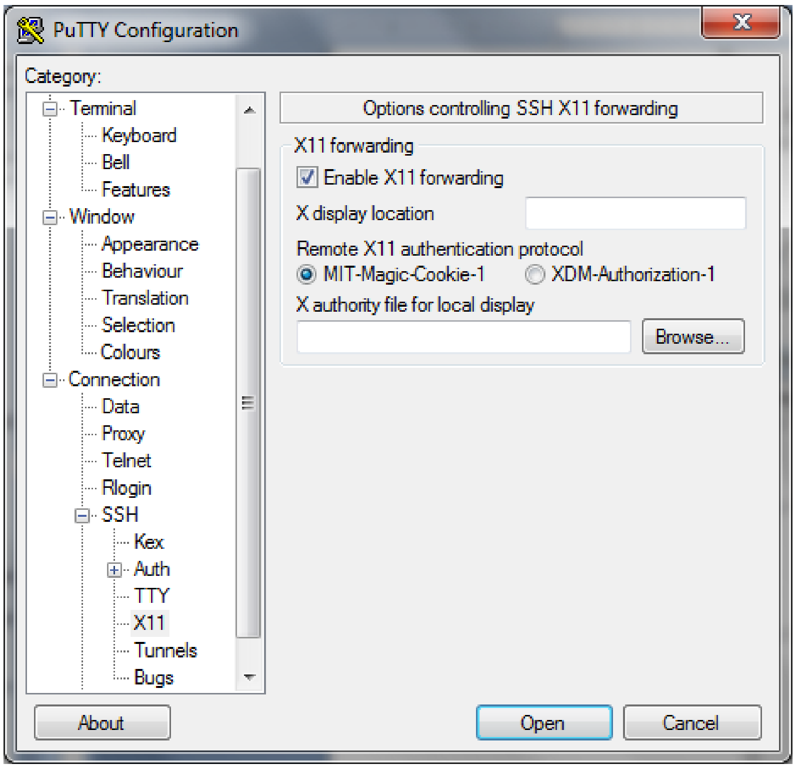
Saved Sessions

Default Settings

Close window on exit: Always Never [✓ Only on clean exit]

Figure 12-4: PuTTy Configuration Step 1

And next is click the ”plus” sign on SSH and click on ”X11” and enable X11 forwarding as shown in Figure 12-5.



Options controlling SSH X11 forwarding

X11 forwarding

[✓ Enable X11 forwarding]

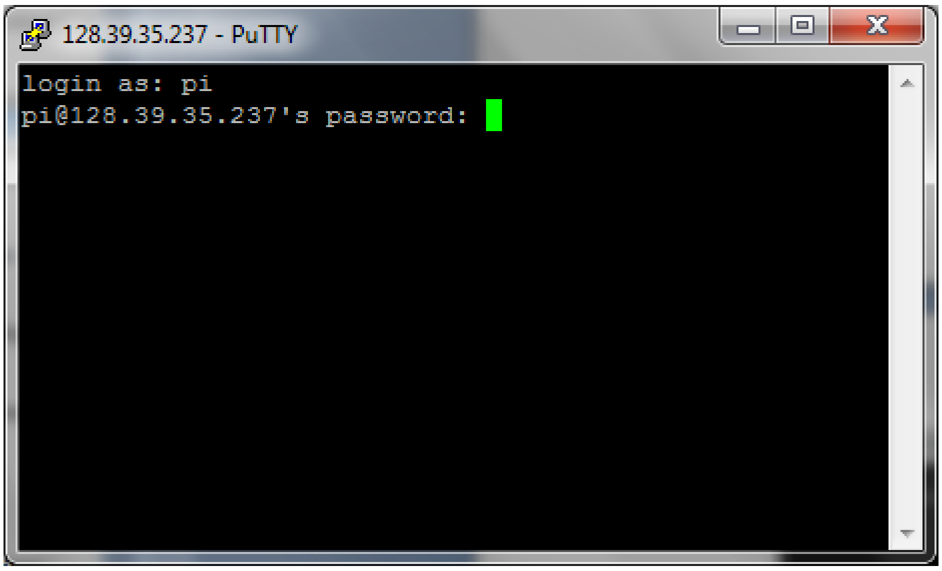
X display location:

Remote X11 authentication protocol: [✓ MIT-Magic-Cookie-1] XDM-Authorization-1

X authority file for local display:

Figure 12-5: Putty Configuration step 2

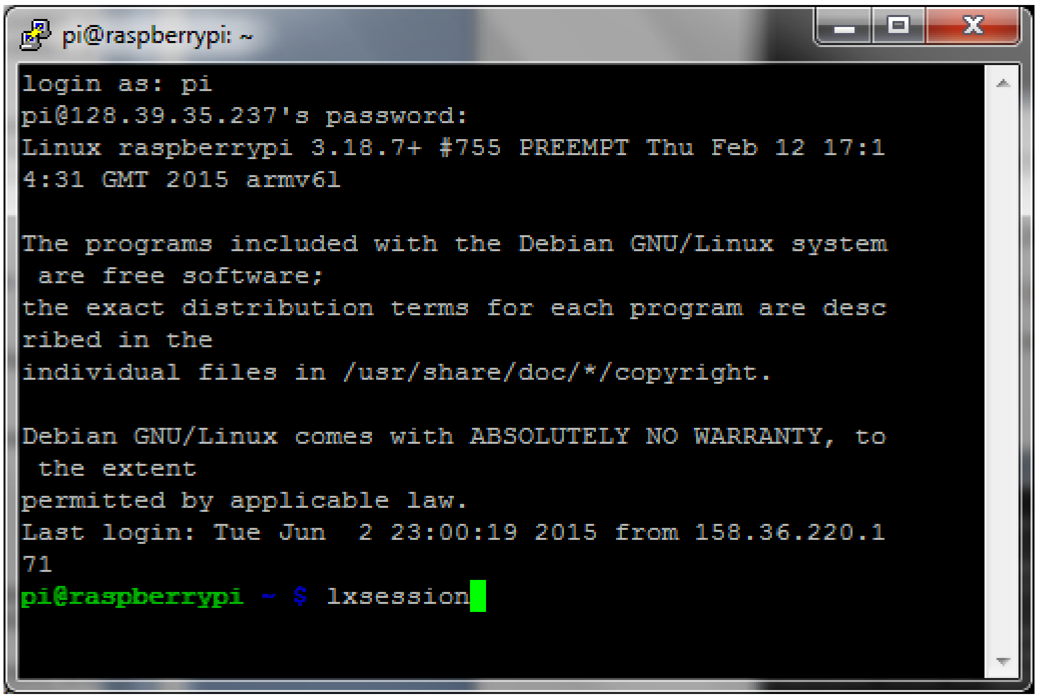
Finally, click on “open” button and as a result; window will be displayed as shown in Figure 12-6.



```
login as: pi
pi@128.39.35.237's password:
```

Figure 12-6: PuTTy login windows

In this window, username and password of the Raspberry Pi should be entered. The default username for a Raspberry Pi is pi. Press enter after entering correct username and password and you will see the windows where you should enter the text ”lxsession” as shown in Figure 12-7.



```
login as: pi
pi@128.39.35.237's password:
Linux raspberrypi 3.18.7+ #755 PREEMPT Thu Feb 12 17:1
4:31 GMT 2015 armv6l

The programs included with the Debian GNU/Linux system
 are free software;
the exact distribution terms for each program are desc
ribed in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to
 the extent
permitted by applicable law.
Last login: Tue Jun  2 23:00:19 2015 from 158.36.220.1
71
pi@raspberrypi ~ $ lxsession
```

Figure 12-7: PuTTy success windows

Then, you will be able to see the desktop of Raspberry Pi in the laptop as shown in Figure 12-8 and mouse and keyboard can be used for the Raspberry Pi.

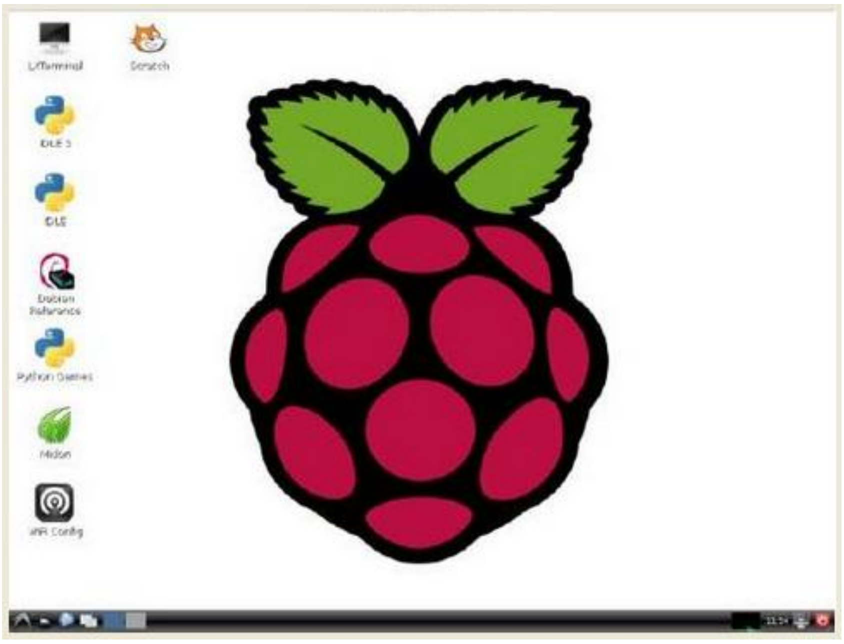


Figure 12-8: Raspberry Pi Desktop on your Laptop Display

12.3 Installation of required applications on Raspberry Pi

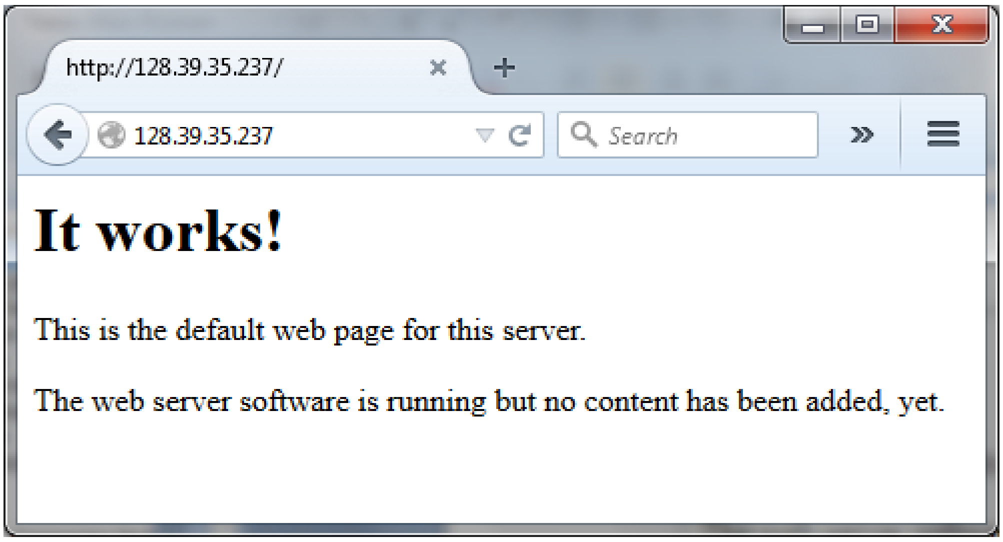
There are many applications that are needed to install in the Raspberry Pi for the completion of the thesis. For data logging, MySQL apache5 and phpmyadmin are needed to install whereas for the web-page development, PHP is needed to install. Web page is used for the monitoring and managing purpose.

12.3.1 Installation of Apache, PHP and MySQL

For the installation of apache, PHP and MySQL, first open the LX Terminal and start typing the command given as follow in sequence.

1) First it should be sure that the Raspberry Pi is up-to-date. For that type "*sudo apt-get update*" in LX Terminal. It may takes some times.
2) To install Apache type the command which is written in italic form, "*sudo apt-get installs apache2 apache2-doc apache2-utils*". Then, the installation asks a confirmation where you should type "y" as "yes". It may take 2 to 3 minutes. After installation of Apache2, Apache puts a test HTML file in the web folder by default. The default web page can be viewed by entering url as http://localhost/ in Raspberry Pi or http://128.39.35.237 (whatever the Raspberry Pi's IP address is) from another computer on the network (raspberrypi.org, 2015b). The default web page looks as shown in Figure 12-9. The IP address can be obtained by typing

"*ifconfig*" in LX Terminal and IP address of Raspberry Pi is displayed as"inet addr".



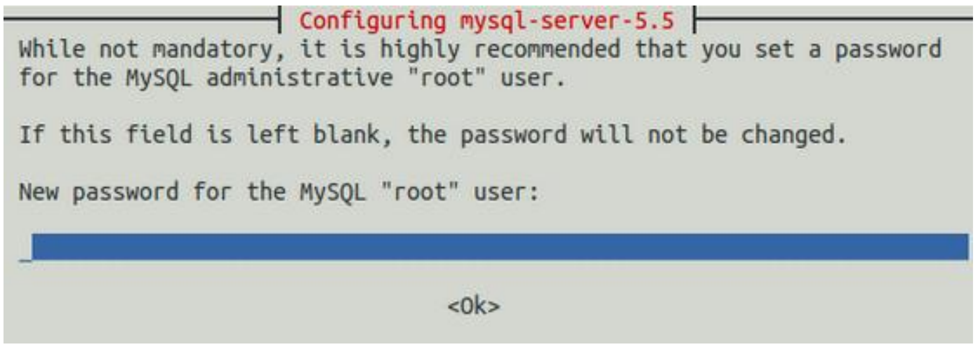
It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Figure 12-9: Default web page

3) Once the apache is installed, we need to install few support packages including PHP. For that, type the command as"*sudo apt-get install libapache2-mod-php5 php5 php-pear php5-xcache*". In this case, it is probable to get the error indicated as "404 Not Found", you need to type the command as "*apt-get update --fix-missing*" and re-enter the above command to install PHP (stackexchange, 2015).
4) This shouldn't take long time to install PHP as well. After this, we should follow up with installing the support package for database connectivity. The command for this part is as follow. "*sudo apt-get install php5-mysql*"
5) Now, finally to install MySQL server, type the command as"*sudo apt-get install mysql-server mysql-client*" in the LX Terminal. During the installation of MySQL server, you will be asked to enter the password for MySQL as shown in Figure 12-10 and re-enter the password for confirmation. It is recommended to give the password and this will be the password for"root" user. This is the last step for the configuration of Apache, PHP and MySQL in Raspberry Pi (raspipress, 2015a).



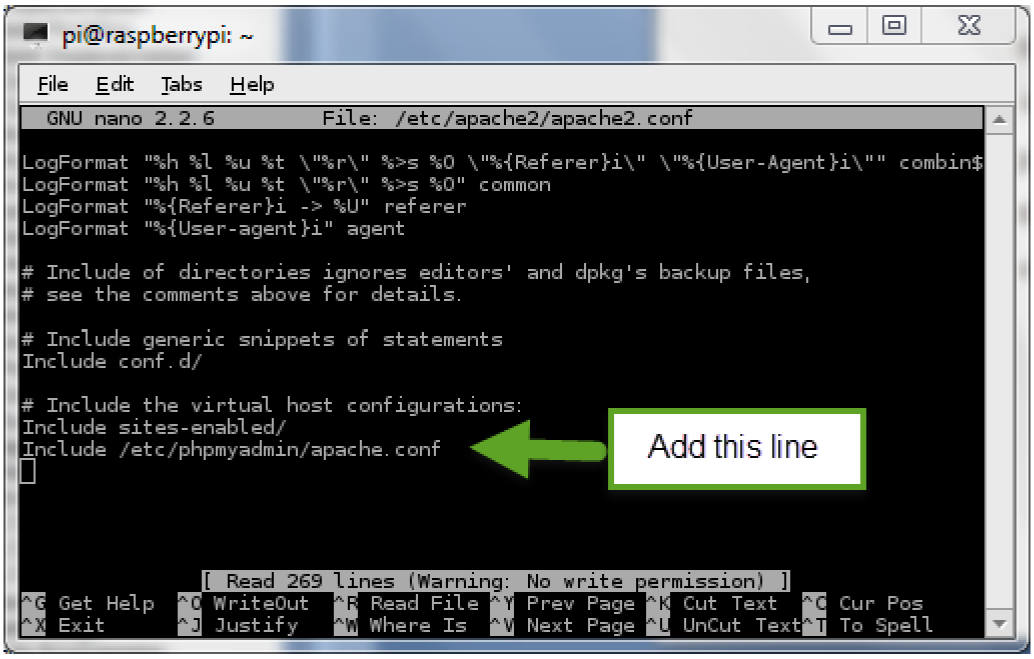
```
Configuring mysql-server-5.5
While not mandatory, it is highly recommended that you set a password
for the MySQL administrative "root" user.

If this field is left blank, the password will not be changed.

New password for the MySQL "root" user:

                         <Ok>
```

Figure 12-10: Password for MySQL

12.3.2 Installation of phpmyadmin

PhpMyAdmin is a handy web interface for managing local MySQL databases, and can make database queries, management and backups easy (RaspiPress, 2015b). The installation of phpmyadmin is explained in following section in stepwise.

1) Install the PhpMyAdmin package by entering the command, "*sudo apt-get install phpmyadmin*". During installation, you will be asked to choose a web server where apache2 should be chosen.
2) The next thing is to configure for dbconfig-common. Here, you should select Yes", this will ask you the administrative password which is the root password that was set during the MySQL installation in section 12.3.1. Next, you will be asked to enter password for"phpmyadmin". The password for MySQL and Phpmyadmin is different thing but you can keep same for the both. This is the completion of the installation of phpmyadmin.
3) Next is to configure the apache to work with phpmyadmin. For that, we need to run the command," *sudo nano /etc/apache2/apache2.conf*" which leads to open the configuration file in Nano. Keep pressing CTRL + V to navigate to the bottom of the configuration file and add a new line as "*Include /etc/phpmyadmin/apache.conf*" which is shows as in Figure 12-11 and save it (CTRL + X and press Y when prompted). This is completion of configuration and need to restart the apache to use this service. To restart the apache, the command "*sudo service apache2 restart*" should be run.



```
  GNU nano 2.2.6            File: /etc/apache2/apache2.conf

LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combin$
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see the comments above for details.

# Include generic snippets of statements
Include conf.d/

# Include the virtual host configurations:
Include sites-enabled/
Include /etc/phpmyadmin/apache.conf


                  [ Read 269 lines (Warning: No write permission) ]
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 12-11: Configuration of apache file

Now, it is ready to use the phpmyadmin. For that, type http://localhost/phpmyadmin in browser of Raspberry Pi itself or http://128.39.35.237/phpmyadmin in browser of a computer under the network (RaspiPress, 2015b). The phpmyadmin in a browser (login page) is shown in Figure 12-12.

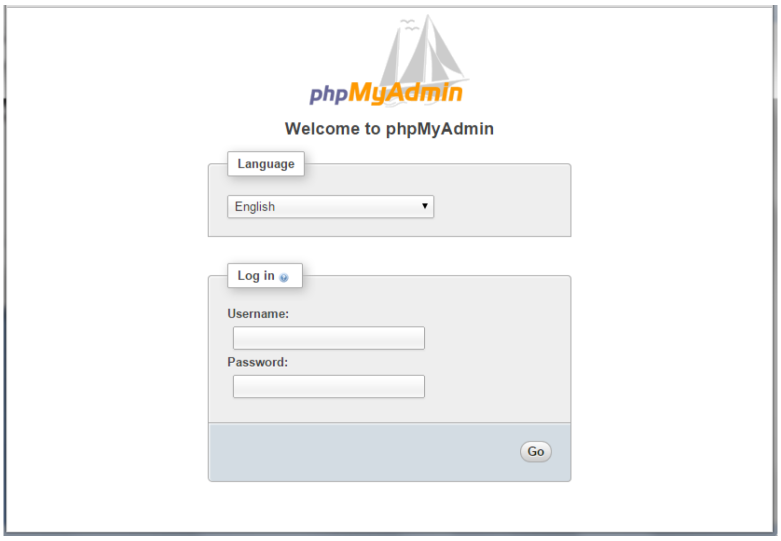


Figure 12-12: phpmyadmin in browser

12.3.3 Installation of Arduino Ide

It is very easy to install Arduino Ide on Raspberry Pi. A simple command is enough for its installation. The command is as follow. *sudo bash*" and then"*apt-get install arduino*". The version of Arduino IDE installed is Arduino 1.0.1 which is not the latest version but it works on Arduino Uno.

13 Database Implementation

Before the implementation of database, it is very important to know about all the requirements of the project. The main requirements of this thesis are to log the process data, manage it and monitor on it. Database comes first to play its role in such application. For data logging, database should be designed and implemented according to the structure of the data and information that need to be stored and retrieve in future.

13.1 Database Modeling

Database modeling is the first phase for the development of the application. As per the requirement of the application, database model is designed in such way that all necessary information can be stored in database and possible to manage and monitor in effective manner. Database is designed using the software, CA Erwin and it is shown in Figure 13-1.

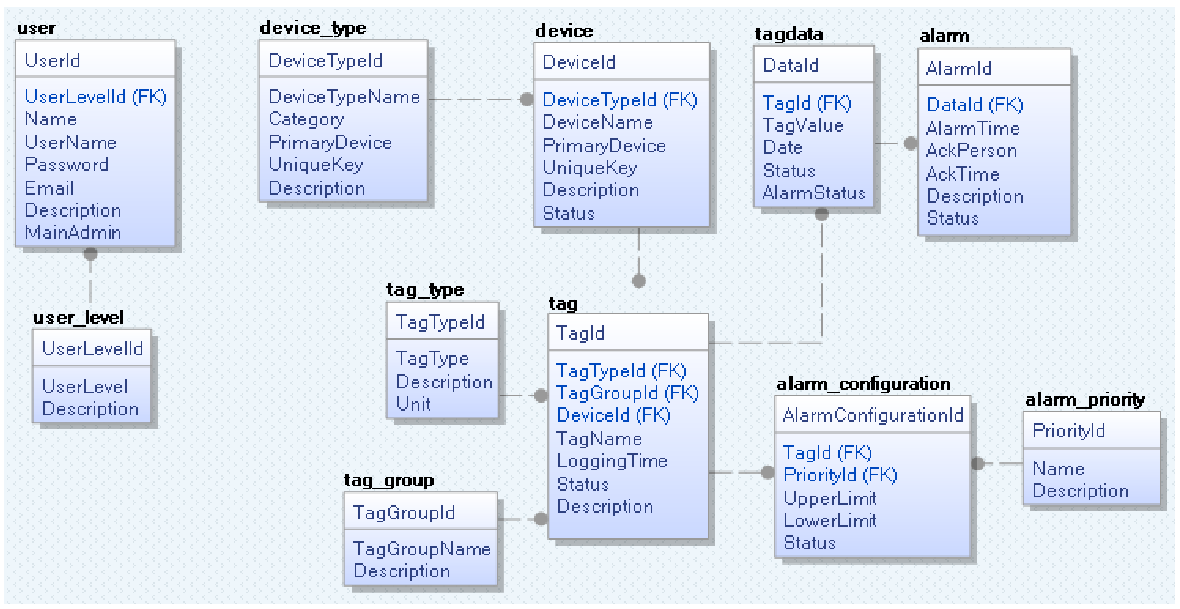


Figure 13-1: Database Modeling in CA Erwin

There are four groups of tables in database as per their purpose in the application. These four groups include user management for login purpose, device and tag management which includes the adding and deleting devices, data monitoring and alarm management.

User management includes two tables; USER and USER_LEVEL. The table, USER_LEVEL records the information of the level of the user of the application whereas the table, USER consist the detail information about the user. According to the level, access to the application varies. User level may be admin, operator or visitor.

Device and tag management consists of five tables; DEVICE_TYPE, DEVICE, TAG_TYPE, TAG_GROUP and TAG. It allows the addition of device and tags in the system and can store all related information.

For monitoring purpose, TAGDATA table is used. All the process values are logged in this table hence it is used for monitoring purpose as well.

Three tables are used for the alarm management and the tables are ALARM, ALARM_CONFIGURATION and ALARM_PRIORITY. The table, ALARM consists the tag value, alarm occurring time, acknowledge time, etc where other two tables consist configuration and priority of alarms.

As you can see in Figure 13-1, there is inter-connection among the tables of a group and among the group as well. These interconnections of the table make for the effective management and presentation of data. These connections makes possible to join two or more tables to extract the data from multiple tables using a single SQL.

13.2Database Implementation

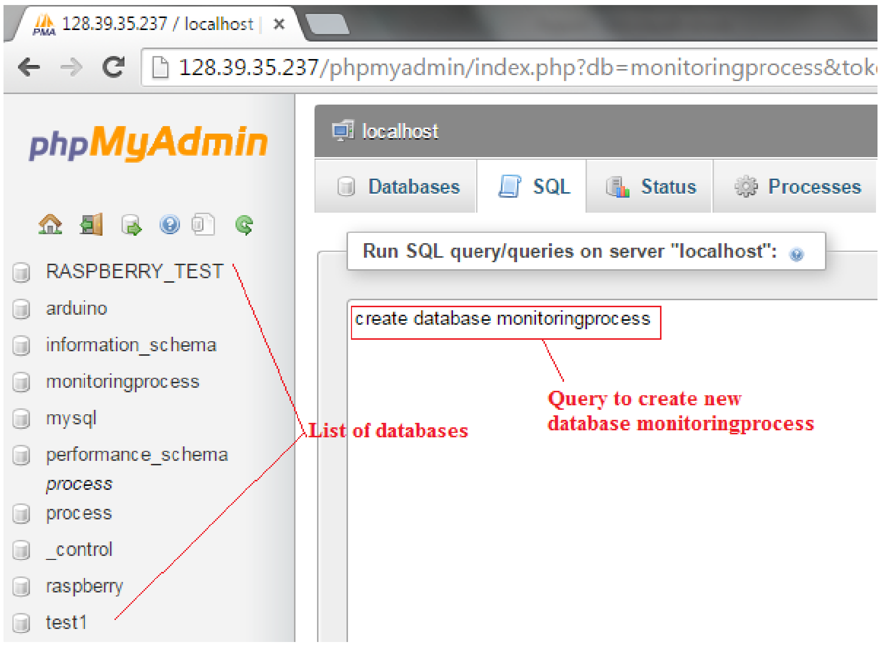
After the completion of database design, it is the time for its creation. MySQL is the database server used for the purpose of logging the data in the project. And phpmyadmin is used for the management of the database. Phpmyadmin is manager for whole MySQL server and a single database and it brings MySQL to a browser so that it will be easier for management.

13.2.1 Creating Database

It is very simple to create a new database in phpmyadmin. For that, we just need to run a create query as shown in Figure 13-2 and the database will appear in the database list.

In this thesis, a database is created with the name as ”monitoringprocess” which runs on MySQL Server built in Raspberry Pi. After the creation of the database, all the tables as mentioned in database design are to be created which is explained in the following section 13.2.2.

The list of the database shown in the Figure 13-2 is used for other purposes. We will concern to only one database, “monitoringprocess”.

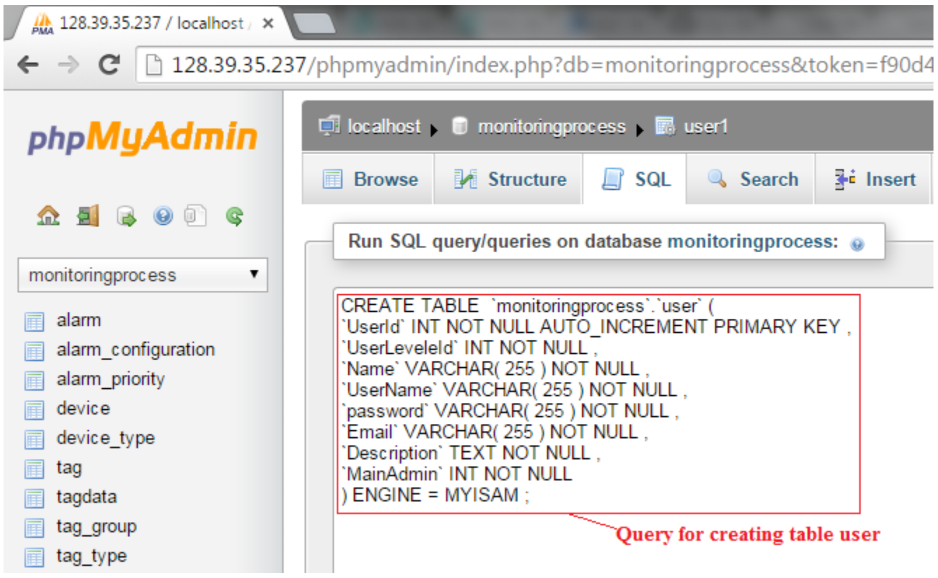


```
create database monitoringprocess
```

Figure 13-2: Creating a database

13.2.2 Creating Table

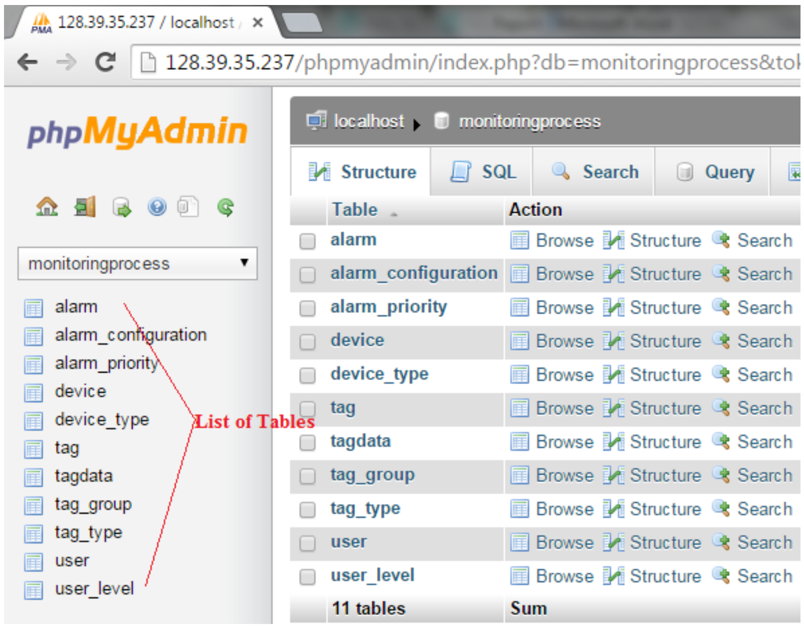
After the creation of a database, the particular database is selected by clicking on it which is displayed in the database list and all the required tables are created. Tables can be created by running the create query. The create query for creating a new table is shown in Figure 13-3 and the table displays in the list.


128.39.35.237 / localhost
128.39.35.237/phpmyadmin/index.php?db=monitoringprocess&token=f90d4
phpMyAdmin
localhost
monitoringprocess
user1
Browse
Structure
SQL
Search
Insert
Run SQL query/queries on database monitoringprocess:
monitoringprocess
alarm
alarm_configuration
alarm_priority
device
device_type
tag
tagdata
tag_group
tag_type
Query for creating table user

```
CREATE TABLE  `monitoringprocess`.`user` (
`UserId` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`UserLevelerId` INT NOT NULL ,
`Name` VARCHAR( 255 ) NOT NULL ,
`UserName` VARCHAR( 255 ) NOT NULL ,
`password` VARCHAR( 255 ) NOT NULL ,
`Email` VARCHAR( 255 ) NOT NULL ,
`Description` TEXT NOT NULL ,
`MainAdmin` INT NOT NULL
) ENGINE = MYISAM ;
```

Figure 13-3: Creating a new table

Similarly, all the tables are created and database is implemented as per it is designed in CA Erwin. The list of all the created tables of the database “monitoringprocess” is shown in Figure 13-4. All together, there are 11 numbers of tables which is indicated by a number.



alarm
alarm_configuration
alarm_priority
device
device_type
tag
tagdata
tag_group
tag_type
user
user_level

Table	Action
alarm	Browse Structure Search
alarm_configuration	Browse Structure Search
alarm_priority	Browse Structure Search
device	Browse Structure Search
device_type	Browse Structure Search
tag	Browse Structure Search
tagdata	Browse Structure Search
tag_group	Browse Structure Search
tag_type	Browse Structure Search
user	Browse Structure Search
user_level	Browse Structure Search
11 tables	Sum

Figure 13-4: List of tables

All the database and tables are creating in Raspberry Pi using the keyboard and screen of laptop as explained in section 12.2.

13.3 Database Table descriptions

The database consists of 11 tables in total and all the tables are created with their individual purposes. Each and every table is used to store different information as rows which in future can be useful. The Table 13-1 shows the list of table names along with their function.

Table 13-1: Table Name with their functions

S.N	Table Name	Funtion
1	userlevel	Stores the level of users; admin, operator or visitor
2	user	Stores information about users; name, username, password, email, etc.
3	device_type	Stores the type of device used in the thesis; Raspberry Pi, Arduino, Ethernet shield, etc
4	device	Saves information about the device used.
5	tag_type	Stores the type of tag used; temperature sensor, pressure sensor, etc
6	tag_group	Stores the group of tag belonging; industry A, industry B, etc
7	tag	Stores details about the tag used.
8	tagdata	Main tables which stores the tag values according to the tag.
9	alarm	Stores information about alarm; status, acknowledge status, alarm time, etc.
10	alarm_priority	Set priority for alarm.
11	alarm_configuration	Saves configuration parameters of alarms.

14 Data Logging

Data logging is the primary task of the thesis which includes the reading data from process using Arduino and inserting these values to the database along with the timestamp. It is not that easy and had lots of things to be done and configured to make it work. The set up that need to be taken are explained in details in following sections.

14.1 Preliminary Setup

14.1.1 Adding MySQL Connector/Arduino library

For data insertion, a database connector is needed which is used for the connection of Arduino and database and is introduced in section 10. In this thesis, MySQL Connector/Arduino is used for the connection of Arduino and MySQL.

As it is already mention in the section 10, MySQL Connector/Arduino is just a library of Arduino and it is need to be added in the library so that it can be used. Following steps are to be followed for adding new library in Arduino in raspberry Pi.

1 The library, MySQL Connector/Arduino can be downloaded in this link, Download Library and the library is downloaded in zip file.

2 Extract the zip file, and copy the mysql_connector and sha1 folders to the Arduino Libraries folder. In case of Raspberry pi, the directory is */usr/share/arduino/libraries*. For the first time, permission will be denied when the file is tried to copy. For the solution, just run the command *sudo chown pi /usr/share/arduino/libraries* in LX-Terminal.

3 Open the Arduino and click on sketch menu and hover over import library then libraries are displayed. The library is marked by red box are MySQL Connector/Arduino library in Figure 14-1. If it is not shown in the list, it is just need to restart the Arduino IDE.

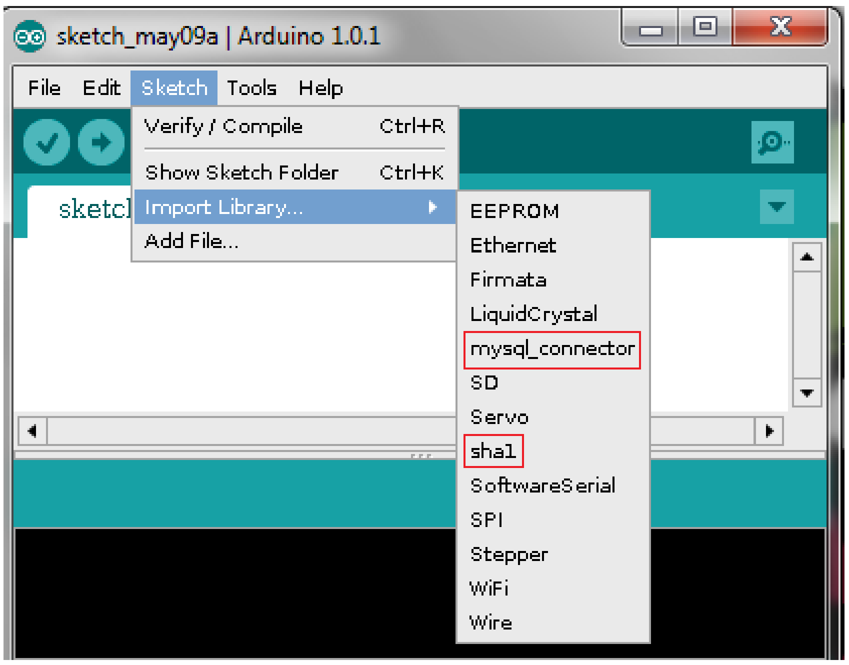


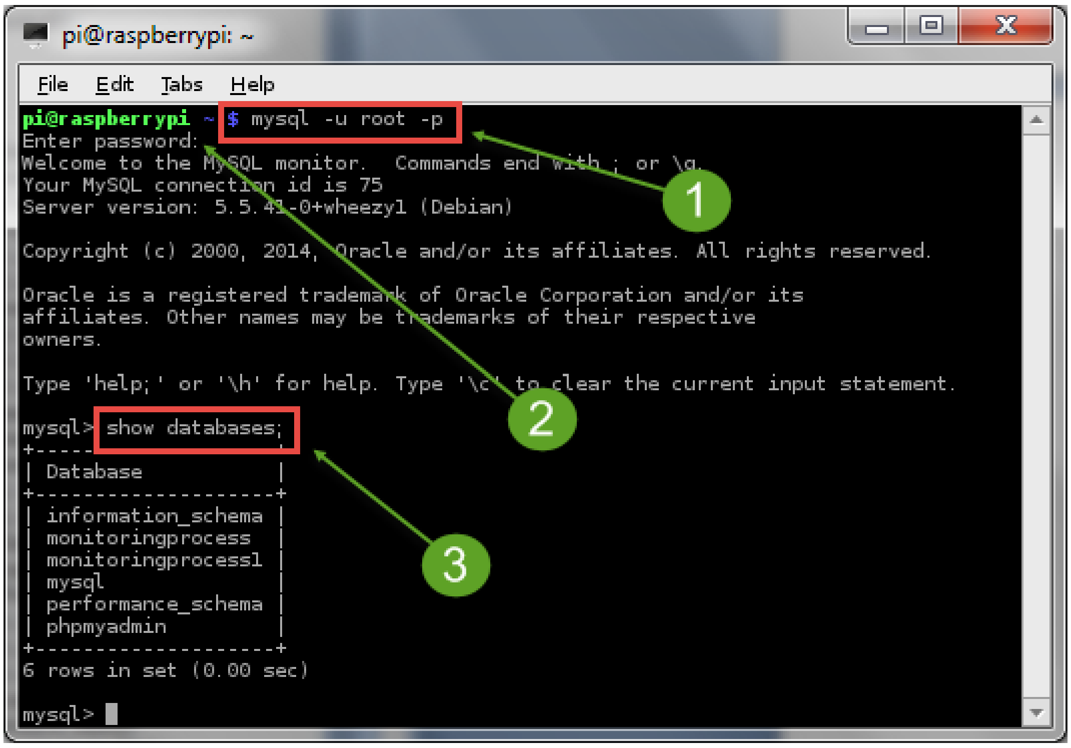
Figure 14-1: Addition of MySQL/Connector library

Now the Connector/Arduino library is installed, and it is ready to start writing database-enabled sketches.

14.1.2 Database Setup

This is the second step that needs to be taken for data logging purpose. We had already created a database and create all the necessary tables for the project. Now, database is needed to be set up so that Arduino can connect to database. All privileges should be granted to user of database to get access to the database. The following things are need to be tested in sequential order.

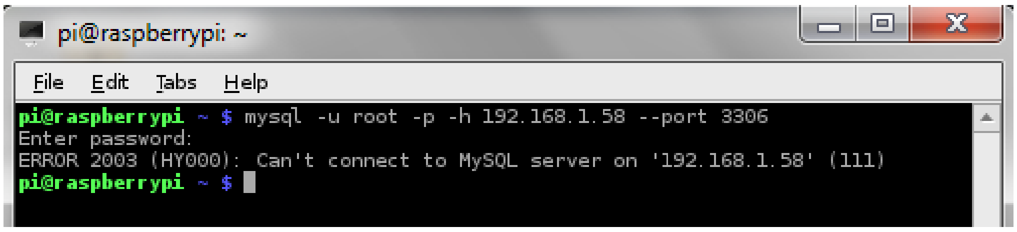
1) First login the MySQL from LX Terminal and view the databases. There are three steps need to be carried out as shown in Figure 14-2. First, query is run for login to MySQL server which is denoted as number 1. For that, password is asked and password is the same password which is set during the installation of MySQL server. Finally, "*show databases*" is run to view the databases. In this case, we can view all the databases. Here, root is the username for database login.



```
pi@raspberrypi ~ $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 75
Server version: 5.5.41-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| monitoringprocess  |
| monitoringprocess1 |
| mysql              |
| performance_schema |
| phpmyadmin         |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

Figure 14-2: Login to database server using LX-Terminal

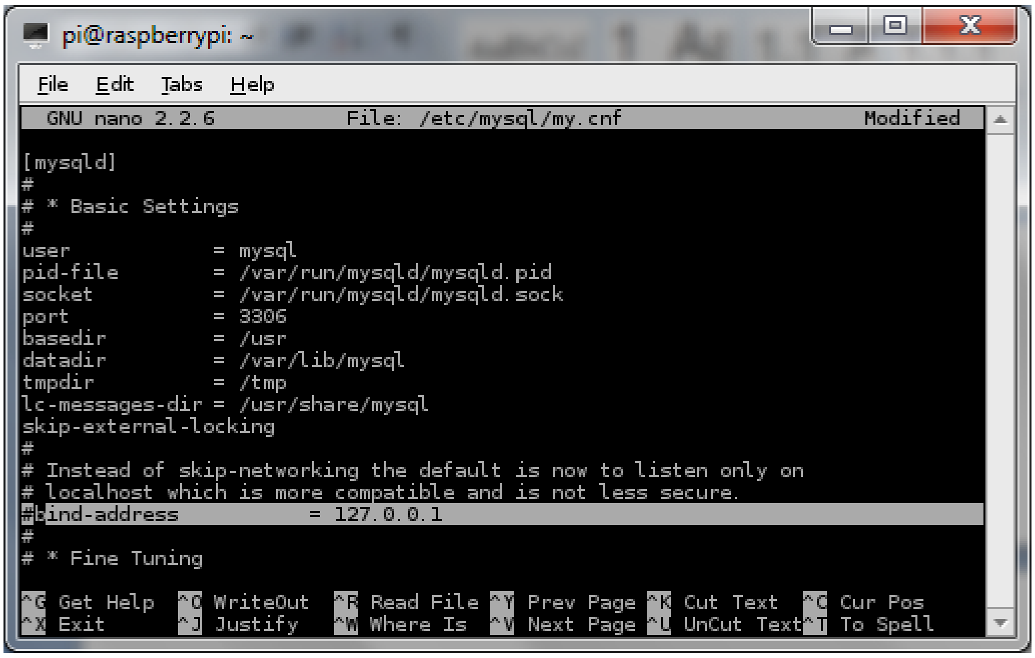
2) The second step is to login MySQL using network. In this case, the command for login to MySQL is changed to, *mysql –u root –p –h 128.39.35.237 port 3306.* Here, 128.39.35.237 stands for the IP address for the MySQL server. In the contest of the thesis, MySQL is in Raspberry Pi; hence it is the IP address of Raspberry Pi. And, 3306 is the listening port. In this case, it may encounter a problem which is shown in Figure 14-3.



```
pi@raspberrypi ~ $ mysql -u root -p -h 192.168.1.58 --port 3306
Enter password:
ERROR 2003 (HY000): Can't connect to MySQL server on '192.168.1.58' (111)
pi@raspberrypi ~ $
```

Figure 14-3: Error during login to MySQL Server

- “my.cnf” file is needed to be edited. For that, two successive commands are to be run in LX Terminal. The commands are: *sudo su* and *nano /etc/mysql/my.cnf* which open the file as shown in Figure 14-4. Bind-address of “my.cnf” file should be commented (put # in front of the code) and save it. For saving, press ctrl + X and press “y” and finally hit enter in the keyboard.



```
  GNU nano 2.2.6            File: /etc/mysql/my.cnf                    Modified

[mysqld]
#
# * Basic Settings
#
user            = mysql
pid-file        = /var/run/mysqld/mysqld.pid
socket          = /var/run/mysqld/mysqld.sock
port            = 3306
basedir         = /usr
datadir         = /var/lib/mysql
tmpdir          = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address           = 127.0.0.1
#
# * Fine Tuning

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Figure 14-4: Commenting bind-Address in my.cnf

- Grant privileges to the user “root”, for that, run the command
  GRANT ALL PRIVILEGES ON monitoringprocess. TO 'root'@'%' IDENTIFIED BY’ kishanpratik’.*
  This command provides all privileges to root on database, monitoringprocess or run the command
  *GRANT ALL ON *.* TO 'root'@'%' IDENTIFIED BY 'kishanprati'*
  This command provides all privileges to the user ”root” on all the databases (stackoverflow, 2013). Here, “kishanpratik” is password for MySQL server.

14.1.3 Arduino Setup

There is nothing much to be set up in Arduino. Arduino is just need to mount with one of its shield, may be Ethernet shield or Wi-Fi shield. Ethernet shield is used in this thesis. Shield is needed to connect the Arduino in network.

14.2 Logging System Setup

The system set up which is made for logging data of sensor/process is shown in Figure 14-5. Instead of process, temperature sensor (TMP 36) is used for logging temperature. Data cable is connected just to upload the Arduino sketch to Arduino board from Raspberry Pi. After the code is uploaded, Arduino can power up by external power supply as indicated in the system setup and data cable can be removed. Hence, the Arduino and Raspberry Pi are not necessary to connected by any kind of cable physically for communication. MySQL/Connector is the Arduino library which is used for connecting the Arduino and database server (MySQL) in the Raspberry Pi. The two Ethernet wire connects both the Arduino and Raspberry Pi to router to be in network.

If process value is needed to log in the database, the temperature sensor is replaced by the process (Air Heater). For that, the positive terminal of voltage analog input (Temp 1) is connected to analog input of Arduino and negative terminal to ground of Arduino.

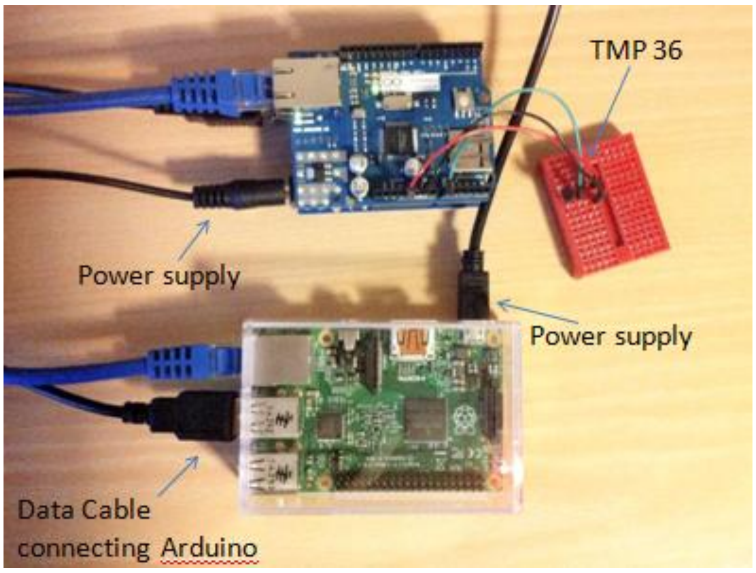


Figure 14-5: Logging System Setup

14.3Arduino Sketch

After the setup of logging system, it is time to code the sketch for data logging in Arduino platform. Some important portion of the sketch is explained in this section and the full Arduino sketch is placed in Appendix C.

14.3.1 Include Files

To use the Connector/Arduino library, it is already explained in section 14.2 that it requires an Ethernet shield and therefore the Ethernet library. The Ethernet library also requires the SPI library. And Connector/Arduino also requires the SHA1 library. Thus, they are needed to be included in order. The following shows all the library header files you need to include at a bare minimum for a MySQL database-enabled sketch.

```
 #include <SPI.h>
#include <Ethernet.h>
#include <sha1.h>
#include <mysql.h>
```

14.3.2 Premilinary Declarations

With the include files set up, it is the time for some preliminary declarations for the Ethernet library and Connector/Arduino.

The Ethernet library requires the set up of MAC address and IP address of Ethernet shield and the IP address of the server. The MAC address is a string of hexadecimal digits and need not be anything special, but it should be unique among the machines on your network. In case of latest Ethernet shield, the MAC address is printed in backside of it. It uses Dynamic Host Control Protocol (DHCP) to get an IP address, DNS, and gateway information. If the network is not enabled for DHCP, it is necessary to give a static IP address for Ethernet shield as well and it should be unique. The IP address of the server is defined using the IPAddress class (which stores the value as an array of four integers, just as you would expect) (Charles Bell, 2013).

```
/* Setup for Ethernet Library */

byte mac_addr[] = {0x90,0xA2, 0xDA, 0x0F, 0xE4, 0xF7};

IPAddress server_addr (128,39,35,237);

byte Ethernet_IP[] = { 128,39,35,238 };
```

Next, some variables for Connector/Arduino are needed to set up. It is needed to define a reference to the library and some strings to use for the data. At a minimum, these include a string for the username, another for the password, and one for the query you use. The declaration is given as follow.

```
/* Setup for the Connector/Arduino */
Connector my_conn;        // The Connector/Arduino reference
 char user[] = "root";
char password[] = "kishanpratik";
char INSERT_SQL[] = "Write your insert query";
```

The last string which is used for storing the SQL may vary according to the need.

14.3.3 Connecting to a MySQL Server

The code for connecting to MySQL Server is written in *setup()* as it is just need to connect to the database for a time. The code for the connection is given as follow.

```
Serial.begin(115200);

 Ethernet.begin(mac_addr,Ethernet_IP);

 If (my_conn.mysql_connect (server_addr, 3306, user, password))

Serial.print(“Connected”);

Else
```

Serial.print(“Not Connected”);

The code begins with a call to the Ethernet library to initialize the network connection with the use of *Ethernet.begin()* method, passing both the MAC address and IP address of Ethernet shield as shown in the example. If the network is enabled DHCP, we just can use *Ethernet.begin(mac_addr);* it causes the Ethernet library to use DHCP to obtain an IP address.

And the next step is the call for serial monitor. Though it is not compulsory to use but it is good idea to use so that user can view the connection message or error message if any in serial monitor sent by the library.

As already mentioned that it is just needed to connect to database just for a time so connecting to the server is a single call to the Connector/Arduino library named as *mysql_connect()*. The IP address of the MySQL database server (IP address of Raspberry Pi in our case), the port the server is listening on (3306), and the user name and password of the database server are to be passed in the function *mysql_connect() (Charles Bell, 2013)*.

14.3.4 Running the query

Usually, query is needed to run continuously, hence it is kept in *loop()* method in the Arduino sketch. The cod for running a query is shown as follow.

my_conn.cmd_query(INSERT_SQL);

It consists of simply a method named *cmd_query()* and pass it the query defined earlier as per the need. It is very important to give *delay ()* function in each loop.

14.3.5 Appending Float variable to the Query

In most of the project, it is necessary to save the real time float values from sensors to a database. Similarly, it is needed to log the real time process value to database in certain fix interval which in float in nature in the project. For this, it is important to append the float variable to the insert query. A simple example for this purpose is placed as follow.

char charTemp[6];

float temp = 20.5; // process value for example

/ conversion from float to char array */*

dtostrf(temp, 4, 3, charTemp);

```
/* conversion from char array to string */
for (int i=0; i<sizeof(charTemp); i++)
 {
        stringTemp+=charTemp[i];
}
String query="insert into monitoringprocess.tagdata (TagValue) values (“;
/* concading strings */
 query += stringTemp;
query += “)”;
/* Get length of string to create the char of same */
 int str_len = query.length() + 1;
char INSER_SQL[str_len];
/* conversion of string to char */
query.toCharArray(INSER_SQL,str_len);
/* Run the query INSER_SQL */
my_conn.cmd_query(INSER_SQL);
```

At the last of this bunch of code, the string query results as:

String query="insert into monitoringprocess.tagdata (TagValue) values (20);

This string can be easily converted to char variable by the use of *toCharArray()* function and ready to use in *cmd_query()* function as shown in the example. Hence, it is being able to insert the real time sensor value (float in nature) to a table in the database. In the example, monitoringprocess is database name whereas “tagdata” is the table name. I had stacked for a long time in this part, so I had included this as well in my report.

14.3.6 Performing SELECT Query

SELECT query is being very useful in this thesis. There are many tags saved in the database which values are to be logged and monitored. Hence, SELECT query is used to distinguish which tag is being used so that tag values are logged along with its respective “TagId”. Unique “TagId” is assigned to each and every tag in database so it is use for distinguishing the particular tag from the group of tags.

The example for performing SELECT query is given as follow according to this library code.

```
const char QUERY_SEL[] = "SELECT TagId FROM monitoringprocess.tag WHERE
`DeviceId` = 1 ";
my_conn.cmd_query(QUERY_SEL);
  my_conn.get_columns();
/* Now we read the rows */
 row_values *row = NULL;
  do {
            row = my_conn.get_next_row();
            if (row != NULL)
              {
                String Id = row->values[0]; // first value returned in the row in string form
                 long id = atol (row->values[0]);  // in the form of integer
              }
} while (row != NULL);
my_conn.free_columns_buffer();
my_conn.free_row_buffer();
```

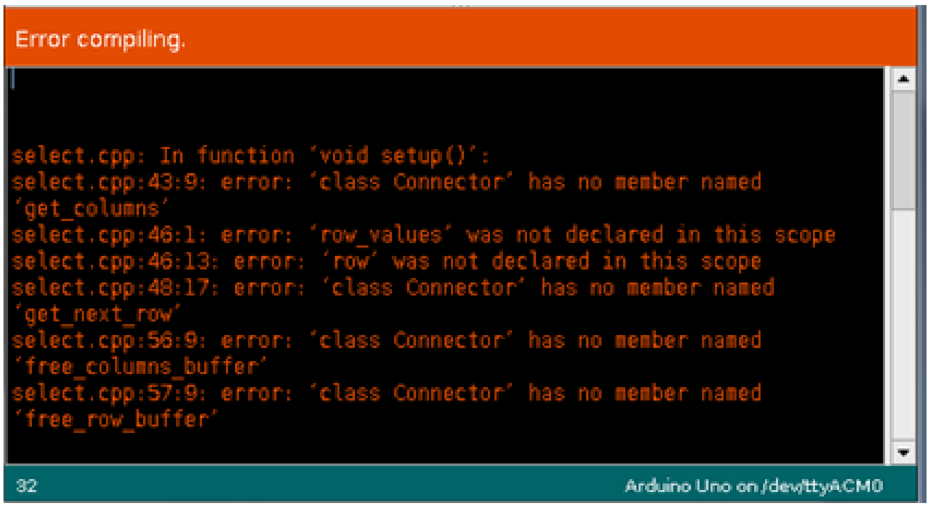
This code is used for that SELECT query which returns just a row. *get_columns()* is used for retrieving the column names and *get_next_row()* is used to read rows which is an iterator whereas *free_columns_buffer()* and *free_row_buffer()* are memory-cleanup methods. We should call the *free_row_buffer()* method after processing the data for the row and the *free_columns_buffer()* once all the rows are read. And a function, *atol()* is used for converting string to long integer.

In this thesis, I have made some modification in the MySQL/Connector Library. I had added one function for performing SELECT query in the library so that I don't need to include the above long code in the main Arduino file. In my code, I just need to call a added function "RunQuery()" to perform the SELECT query. The added function in the library is attached in Appendix D.

14.3.7 Error encountered

During executing the code, it is high possible to get some error due to some files in the library itself. It is easy to solve as it is solved with very simple modification in the file.

14.3.7.1 Mysql_connector Library

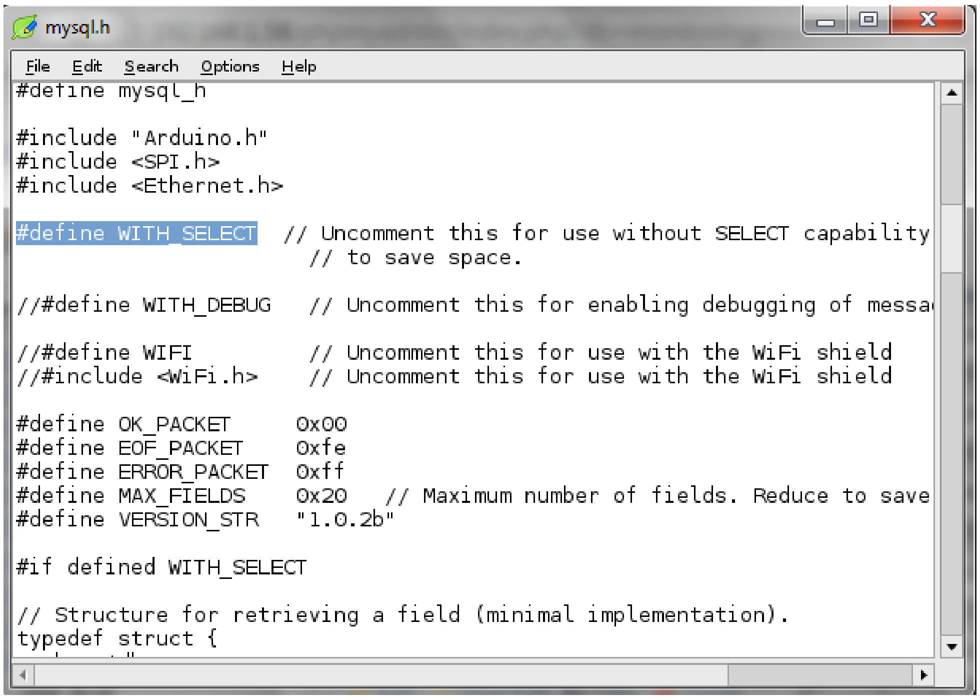


```
select.cpp: In function 'void setup()':
select.cpp:43:9: error: 'class Connector' has no member named
'get_columns'
select.cpp:46:1: error: 'row_values' was not declared in this scope
select.cpp:46:13: error: 'row' was not declared in this scope
select.cpp:48:17: error: 'class Connector' has no member named
'get_next_row'
select.cpp:56:9: error: 'class Connector' has no member named
'free_columns_buffer'
select.cpp:57:9: error: 'class Connector' has no member named
'free_row_buffer'
```

Figure 14-6: Possible Error during SELECT Query

During performing SELECT query, it is probable to get errors for missing methods related to SELECT query as shown in Figure 14-6 (Charles Bell, 2013).

The cause for these errors is that “WITH_SELECT” is commented in the file, mysql.h in connector library. It is commented initially to reduce the program memory. For the solution, it is needed to uncomment “WITH_SELECT” as shown in Figure 14-7.

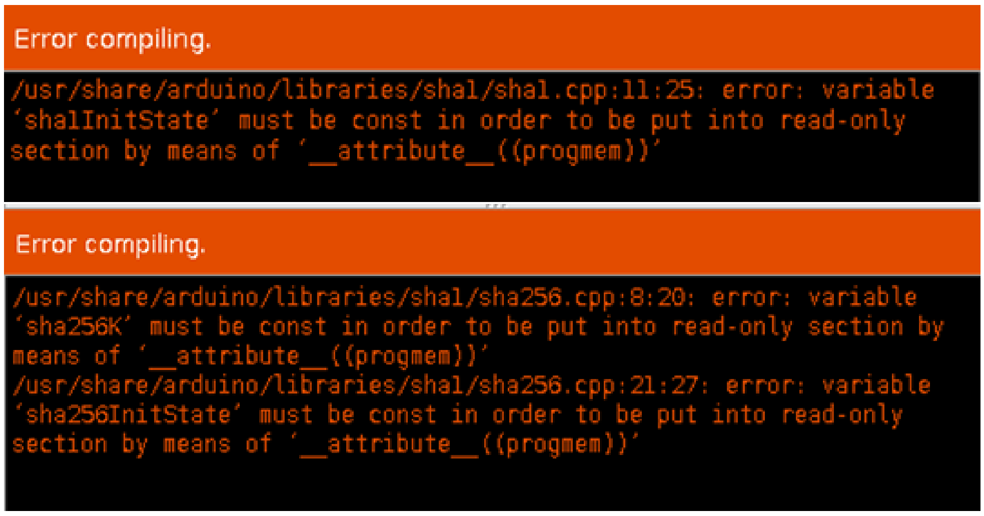


```
#define mysql_h

#include "Arduino.h"
#include <SPI.h>
#include <Ethernet.h>

#define WITH_SELECT   // Uncomment this for use without SELECT capability
                        // to save space.

//#define WITH_DEBUG    // Uncomment this for enabling debugging of messa

//#define WIFI          // Uncomment this for use with the WiFi shield
//#include <WiFi.h>     // Uncomment this for use with the WiFi shield

#define OK_PACKET     0x00
#define EOF_PACKET    0xfe
#define ERROR_PACKET  0xff
#define MAX_FIELDS    0x20   // Maximum number of fields. Reduce to save
#define VERSION_STR   "1.0.2b"

#if defined WITH_SELECT

// Structure for retrieving a field (minimal implementation).
typedef struct {
```

Figure 14-7: Un-commenting WITH_SELECT for SELECT query

14.3.7.2 Sha1 Library

There are some variables in Sha1 library where a keyword “const” should be added in front of them. Before adding the keyword, following errors may encounter which is shown in Figure 14-8.



```
Error compiling.
/usr/share/arduino/libraries/sha1/sha1.cpp:11:25: error: variable
'sha1InitState' must be const in order to be put into read-only
section by means of '__attribute__((progmem))'
```

```
Error compiling.
/usr/share/arduino/libraries/sha1/sha256.cpp:8:20: error: variable
'sha256K' must be const in order to be put into read-only section by
means of '__attribute__((progmem))'
/usr/share/arduino/libraries/sha1/sha256.cpp:21:27: error: variable
'sha256InitState' must be const in order to be put into read-only
section by means of '__attribute__((progmem))'
```

Figure 14-8: Error on sha1 library

The error can be solved just by adding a keyword “const” in front of all the variables; sha1InitState in sha1.cpp and sha256k and sha256InitState in sha256.cpp.

14.3.7.3 Memory Problem

Remember that the Arduino (certainly the Uno) is very limited on memory and program space, so having the Arduino connect direct, and doing anything else, might be mutually exclusive. It shows nothing in serial monitor. If so, just cut off some strings from the code to minimize the program size and run it. If, it runs well now then it is confirm that it is out of memory.

15 Data Monitoring and Management

Data is logged in a database for monitoring and future analysis or evaluation. In this thesis, sensor data and process data are logged in the database which is monitored and managed with the use of web site and if any value is out of limit, alarm is activated. Alarm information is stored in database and can be acknowledge by the users. Website is taken as the monitoring system and it is developed in the Raspberry Pi using PHP.

The designing and programming of the website is done in laptop and after the completion, it is copied to the Raspberry Pi. Raspberry Pi supports all the platforms and programming language used for the development of the website. It is easy and effective to develop the site in laptop as it is very fast as compared to Raspberry Pi.

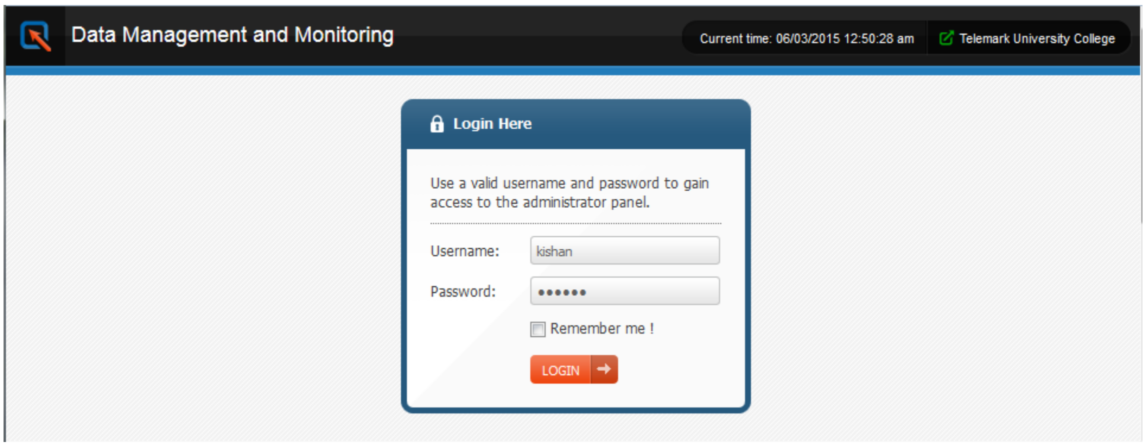
15.1 Website Design

As per the requirement, website is designed. Requirements are different for two cases; monitoring and management.

In case of data monitoring, the design should be able to present the data for the viewer in easy and understandable way. In this system, all the data of are displayed in a table with its logged timestamp and trends is shown in day wise. It is able to show the data per day in a graph and date is able to change to view the trends of the particular date and in case of data management, different tables are designed to display all the tags, devices and data in separate and systematic manner. All the related devices such are Arduino, sensors, Ethernet shield, etc are displayed in a table and tag type and tag group are separated in different one. Different information are displayed in different tables so that it will be easy to manage and understandable as well.

For the development of the website, Dreamweaver C6 is installed in the laptop so that it will be easy to program. Dreamweaver is a well-known web editor (a type of computer program) used by both experienced and novice webmasters to design websites (Heng, 2014). For design, HTML, CSS, JavaScript, etc are used. Different pages are designed for different purposes.

Some designed main pages are shown in following figures. Figure 15-1 shows the login page and Figure 15-2 shows the home page whereas Figure 15-3 shows the tag management page.


Data Management and Monitoring
Current time: 06/03/2015 12:50:28 am
Telemark University College
Login Here
LOGIN

Use a valid username and password to gain access to the administrator panel.

Username:	kishan
Password:	••••••
☐ Remember me !	

Figure 15-1: Login Page

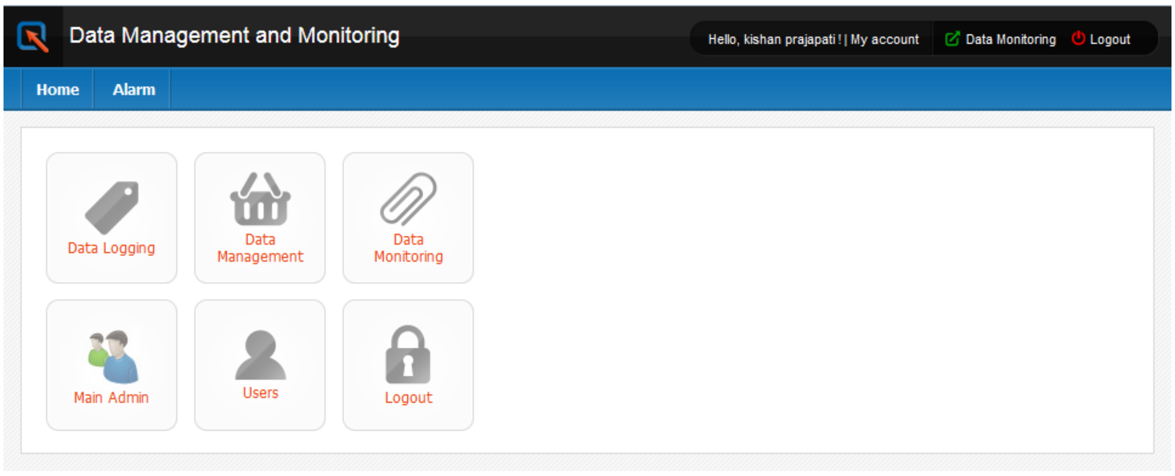
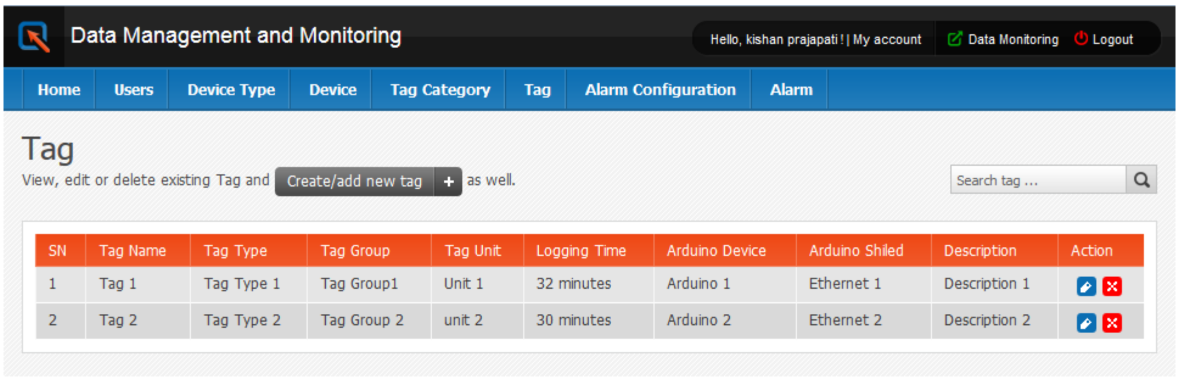


Figure 15-2: Home Page



Tag

View, edit or delete existing Tag and Create/add new tag as well.

SN	Tag Name	Tag Type	Tag Group	Tag Unit	Logging Time	Arduino Device	Arduino Shiled	Description	Action
1	Tag 1	Tag Type 1	Tag Group1	Unit 1	32 minutes	Arduino 1	Ethernet 1	Description 1	
2	Tag 2	Tag Type 2	Tag Group 2	unit 2	30 minutes	Arduino 2	Ethernet 2	Description 2	

Figure 15-3: Tag Management Table

15.2 Website Programming

After the completion of design, functions are to be added to the pages. Website should be function as per the requirements of the thesis. The requirement includes the presentation of data in table and as a trend, adding, deleting and edit devices, etc. For that, programming is to be done in the web pages. PHP is used for the programming and make the web pages work and function as the need. Besides PHP, JavaScript, jQuery and Ajax are used to make the web page more effective and attractive. These add animations and some interesting features in the webpage.

15.3 Copying to Raspberry Pi

The website should be in Raspberry Pi rather than in a laptop as per the requirement of the thesis. The folder of the website is copied to Raspberry Pi with the help of pen-drive. The directory for the folder in a laptop is as follow, *C:\xampp\htdocs* whereas the directory for the folder in the Raspberry Pi is */var/www.*

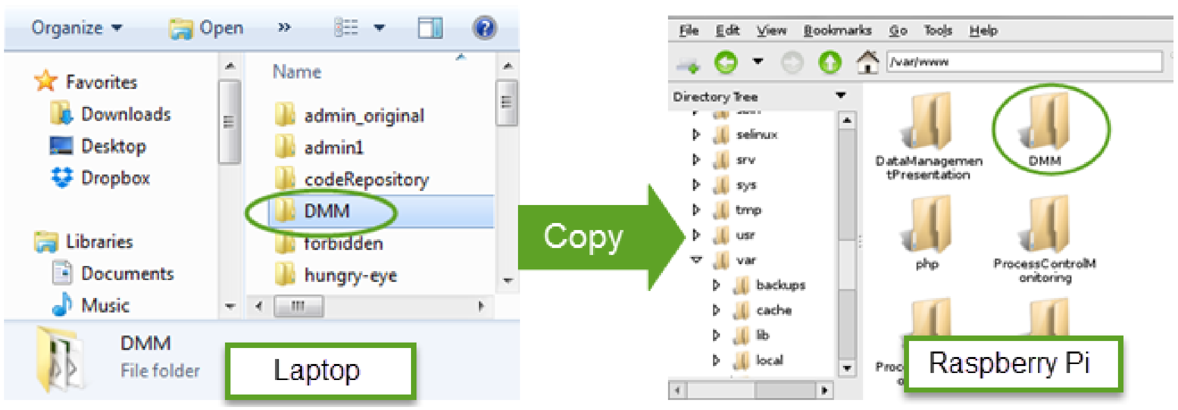


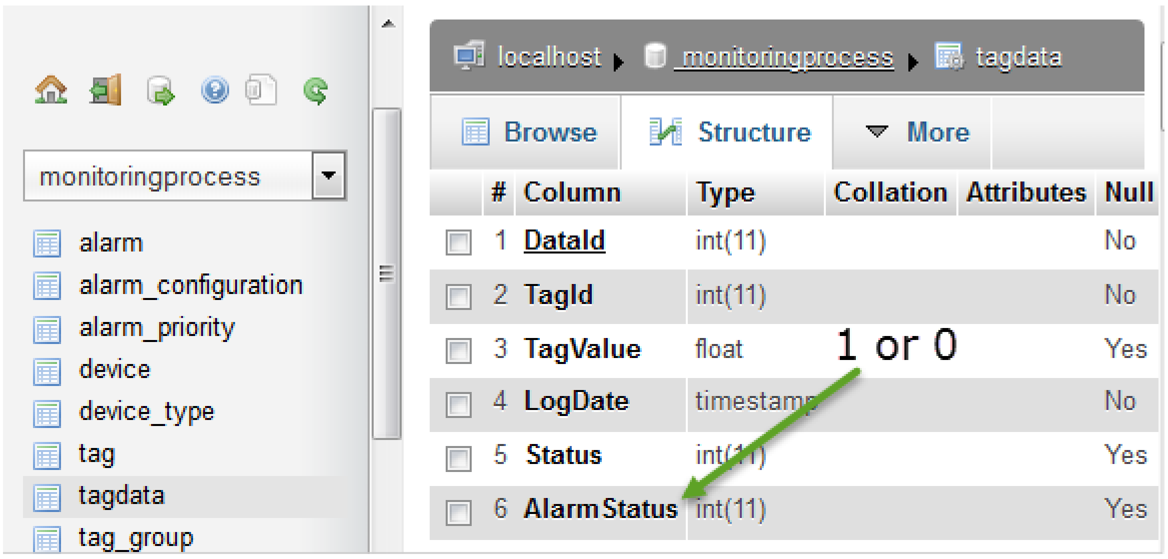
Figure 15-4: Copying file from laptop to Raspberry Pi.

Before copying the file to the directory in the Raspberry Pi, it is necessary to grant permission to copy. For that, a command needs to be run in LX Terminal. The command is as follow, *sudo chown pi /var/www*. Finally, the website can be accessed in a browser with the url as http://localhost/DMM.

16 Alarm System

In data management system, lower value and upper value for each tag is assigned and saved in database. During data logging, if the value exceeds the upper limit or doesn't meet the lower limit, alarm is activated. If any value is detected as the out of range or limit, a row is inserted in alarm table which displays in the alarm section in website and it is possible to acknowledge the alarm from the website.

Table "tagdata" is used for logging the value of tags where "AlarmStatus" is used for indication of alarm activation. If the tag value exceeds the upper limit or doesn't meet the lower limit, the "AlarmStatus" field is set as 1 else it is set 0. SQL Trigger is created in database to insert alarm information in the table "alarm". This trigger checks if the "AlarmStatus" of "tagdata" is 1 or 0, if it finds as 1, a row is inserted to the "alarm" table as the information of the alarm. The active alarms can be viewed in the website by the users (operator and admin).



	#	Column	Type	Collation	Attributes	Null
☐	1	DataId	int(11)			No
☐	2	TagId	int(11)			No
☐	3	TagValue	float			Yes
☐	4	LogDate	timestamp			No
☐	5	Status	int(11)			Yes
☐	6	AlarmStatus	int(11)			Yes

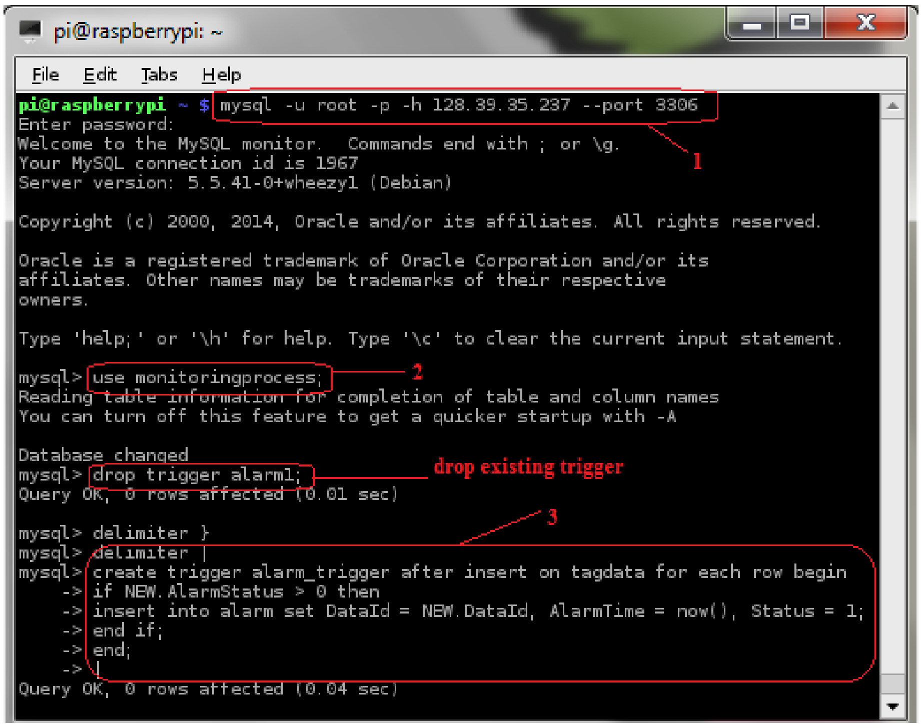
Figure 16-1: Table tagdata.

Figure 16-1 shows the table; "tagdata" in database "monitoringprocess". During data logging, if "AlarmStatus" set as 1 then the trigger will invoke which insert alarm information in table "alarm".

Creating SQL Trigger

It is very simple to create a SQL trigger in MySQL database. The process to create SQL trigger is shown in Figure 16-2. It shows the three steps for creating trigger indicated as 1, 2 and 3. Step 1 is the command to connect MySQL server and step 2 is command for selecting database. Here, database "monitoringprocess" is chosen. And, command for creating trigger is run which is indicated at 3. Finally, a message is get as "QUERY OK".

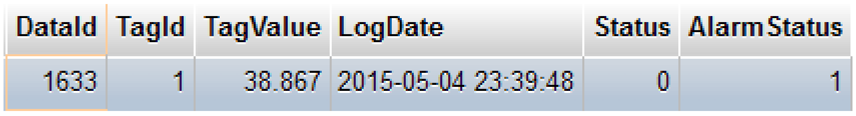
This is the only notification for confirmation of creation of trigger. The name of the trigger is alarm_trigger.



```
pi@raspberrypi ~ $ mysql -u root -p -h 128.39.35.237 --port 3306
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1967
Server version: 5.5.41-0+wheezy1 (Debian)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use monitoringprocess;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> drop trigger alarm1;
Query OK, 0 rows affected (0.01 sec)

mysql> delimiter }
mysql> delimiter |
mysql> create trigger alarm_trigger after insert on tagdata for each row begin
    -> if NEW.AlarmStatus > 0 then
    -> insert into alarm set DataId = NEW.DataId, AlarmTime = now(), Status = 1;
    -> end if;
    -> end;
    -> |
Query OK, 0 rows affected (0.04 sec)
```

Figure 16-2: Creation of SQL Trigger in MySQL.

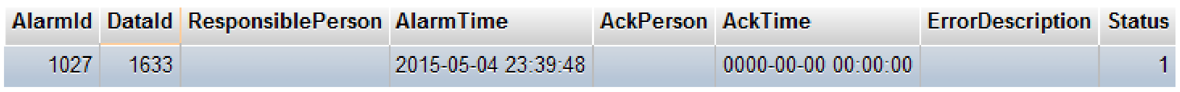
The Figure 16-3 shows the table “tagdata” with one record with AlarmStatus as 1.



DataId	TagId	TagValue	LogDate	Status	AlarmStatus
1633	1	38.867	2015-05-04 23:39:48	0	1

Figure 16-3: Table tagdata with one record

And the Figure 16-4 shows the table “alarm” with one record.



AlarmId	DataId	ResponsiblePerson	AlarmTime	AckPerson	AckTime	ErrorDescription	Status
1027	1633		2015-05-04 23:39:48		0000-00-00 00:00:00		1

Figure 16-4: Table alarm with one record

It is clearly seen that AlarmTime in table “alarm” and LogDate in table “tagdata” contain the exactly same date and time. It means, both the data is inserted at the same time. As, a data with AlarmStatus = 1 is inserted, trigger is invoked then a new record is inserted to table “alarm” automatically at the same time. Hence, trigger works. And it is noticeable that the DataId is same in both records in two tables. Hence, the alarm is related to the same data.

All the records inserted to the table “tagdata” are displayed as an Active Alarm in the website. In this case, the field, “Status” is set as 1 which denotes Active. For each active alarm, there is button to acknowledge it. If the “Acknowledge” button is clicked, the field “Status” is updated to 0 from 1. Then, these alarms are listed in the website as Acknowledged Alarm. Besides, updating the field “Status” in the database, it also provides information about the alarm to an operator when the “Acknowledge” button is clicked. The information about the alarm is send via Email. Information may include tag value, tag name, tag type, tag group, Arduino used, alarm priority, etc.

17 Data logging and Web Service

In this section, sensor values are logged in SQL server built in a laptop with the use of SQL toolkit of LABVIEW and those data are monitored by data dashboard in LABVIEW using web service.

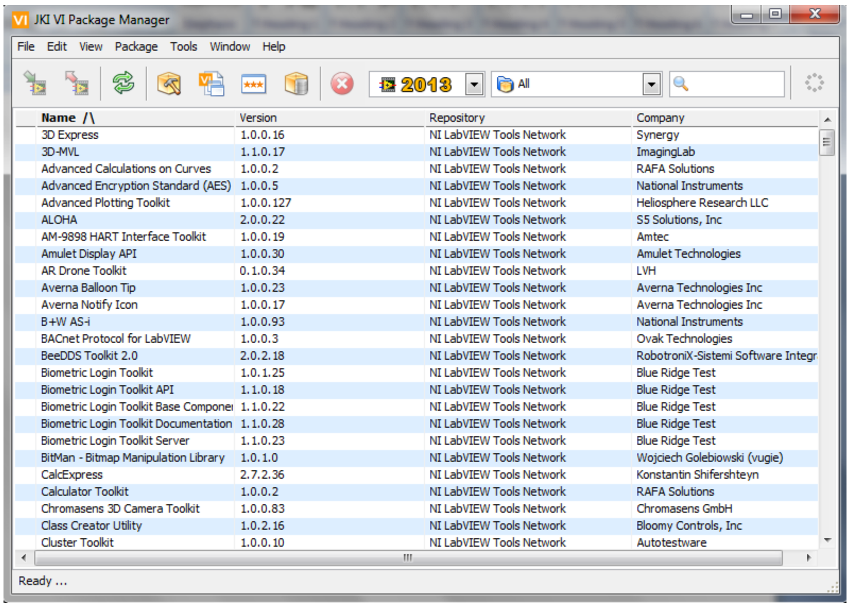
17.1 Data Logging Using LABVIEW

This includes the data logging of sensor values using Arduino and LABVIEW to SQL database. It includes the interfacing Arduino and LABVIEW and connection between LABVIEW and SQL server.

17.1.1 LABVIEW Interface for Arduino

Sensor/process is attached to Arduino which reads its value and the information should be accessed in LABVIEW so that the data can be logged to the database. Setting up the LABVIEW Interface for Arduino is a six step process that needs to complete only once. The following instructions below should be followed to start creating applications with the LABVIEW Interface for Arduino (Sammy_K, 2011).

1) Install LABVIEW
2) Install the NI-VISA drivers as per the operating system in the laptop
   The NI-VISA driver can be downloaded from the link, Download NI-VISA and it should be installed in the laptop.
3) Install JKI VI Package Manager (VIPM) Community Edition (Free):
   VI Package Manager is the add-ons in the LABVIEW and it can be installed by clicking "Find LABVIEW Add-ons" in the LABVIEW Getting Started window or visit the JKI site at Download VIPM.
4) Install the LABVIEW Interface for Arduino as described follow:
   - Launch installed VIPM and it displays list of packages as shown in Figure 17-1.



JKI VI Package Manager

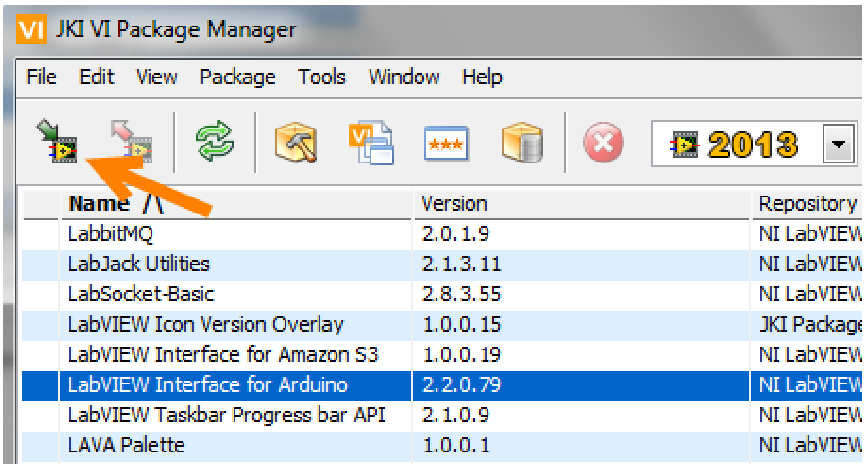
File Edit View Package Tools Window Help

Name /\	Version	Repository	Company
3D Express	1.0.0.16	NI LabVIEW Tools Network	Synergy
3D-MVL	1.1.0.17	NI LabVIEW Tools Network	ImagingLab
Advanced Calculations on Curves	1.0.0.2	NI LabVIEW Tools Network	RAFA Solutions
Advanced Encryption Standard (AES)	1.0.0.5	NI LabVIEW Tools Network	National Instruments
Advanced Plotting Toolkit	1.0.0.127	NI LabVIEW Tools Network	Heliosphere Research LLC
ALOHA	2.0.0.22	NI LabVIEW Tools Network	S5 Solutions, Inc
AM-9898 HART Interface Toolkit	1.0.0.19	NI LabVIEW Tools Network	Amtec
Amulet Display API	1.0.0.30	NI LabVIEW Tools Network	Amulet Technologies
AR Drone Toolkit	0.1.0.34	NI LabVIEW Tools Network	LVH
Averna Balloon Tip	1.0.0.23	NI LabVIEW Tools Network	Averna Technologies Inc
Averna Notify Icon	1.0.0.17	NI LabVIEW Tools Network	Averna Technologies Inc
B+W AS-i	1.0.0.93	NI LabVIEW Tools Network	National Instruments
BACnet Protocol for LabVIEW	1.0.0.3	NI LabVIEW Tools Network	Ovak Technologies
BeeDDS Toolkit 2.0	2.0.2.18	NI LabVIEW Tools Network	RobotroniX-Sistemi Software Integr
Biometric Login Toolkit	1.0.1.25	NI LabVIEW Tools Network	Blue Ridge Test
Biometric Login Toolkit API	1.1.0.18	NI LabVIEW Tools Network	Blue Ridge Test
Biometric Login Toolkit Base Compone	1.1.0.22	NI LabVIEW Tools Network	Blue Ridge Test
Biometric Login Toolkit Documentation	1.1.0.28	NI LabVIEW Tools Network	Blue Ridge Test
Biometric Login Toolkit Server	1.1.0.23	NI LabVIEW Tools Network	Blue Ridge Test
BitMan - Bitmap Manipulation Library	1.0.1.0	NI LabVIEW Tools Network	Wojciech Golebiowski (vugie)
CalcExpress	2.7.2.36	NI LabVIEW Tools Network	Konstantin Shifershteyn
Calculator Toolkit	1.0.0.2	NI LabVIEW Tools Network	RAFA Solutions
Chromasens 3D Camera Toolkit	1.0.0.83	NI LabVIEW Tools Network	Chromasens GmbH
Class Creator Utility	1.0.2.16	NI LabVIEW Tools Network	Bloomy Controls, Inc
Cluster Toolkit	1.0.0.10	NI LabVIEW Tools Network	Autotestware

Ready ...

Figure 17-1: VIPM launched

- Browse to "LABVIEW Interface for Arduino" in the list of packages as shown in Figure 17-2.



JKI VI Package Manager

File Edit View Package Tools Window Help

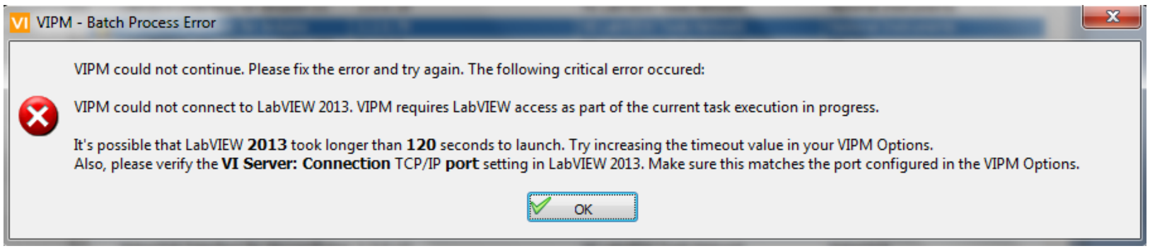
Name /\	Version	Repository
LabbitMQ	2.0.1.9	NI LabVIEW
LabJack Utilities	2.1.3.11	NI LabVIEW
LabSocket-Basic	2.8.3.55	NI LabVIEW
LabVIEW Icon Version Overlay	1.0.0.15	JKI Package
LabVIEW Interface for Amazon S3	1.0.0.19	NI LabVIEW
LabVIEW Interface for Arduino	2.2.0.79	NI LabVIEW
LabVIEW Taskbar Progress bar API	2.1.0.9	NI LabVIEW
LAVA Palette	1.0.0.1	NI LabVIEW

Figure 17-2: Selection of LABVIEW Interfaces for Arduino

- Click on the "Install &Upgrade Packages" button as shown by the arrow in Figure 17-2.

Problem: VIPM can't connect to LABVIEW

When I click the **"**install and upgrade**"** button shown in Figure 17-2, VI Package Manager attempts to connect to LABVIEW. A wait time appears at the bottom. After time elapses, I see the following error dialog box as shown in Figure 17-3.



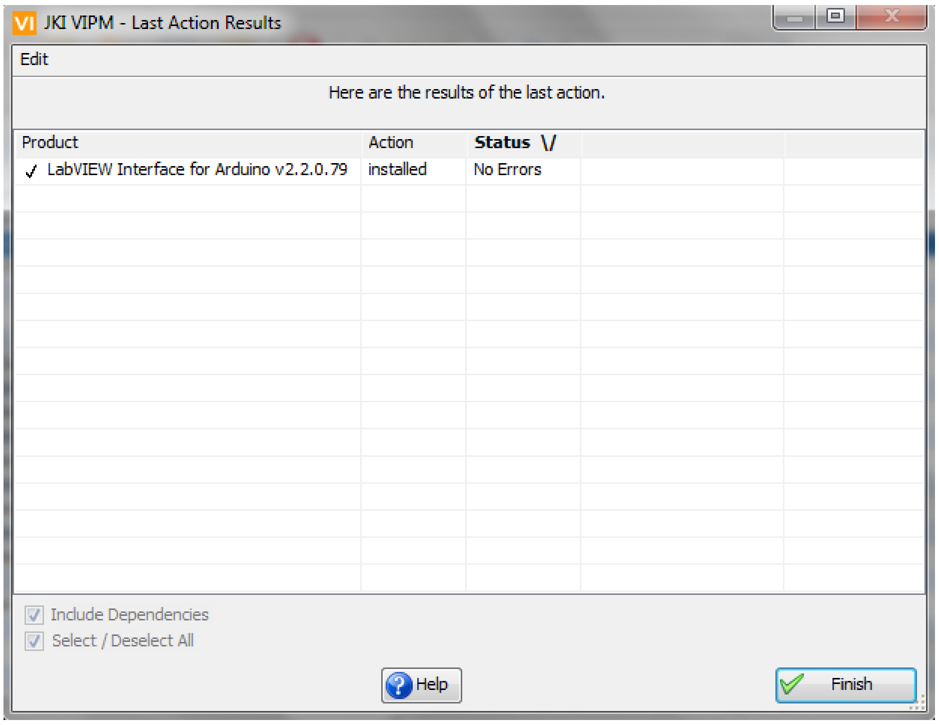
VIPM could not continue. Please fix the error and try again. The following critical error occured:

VIPM could not connect to LabVIEW 2013. VIPM requires LabVIEW access as part of the current task execution in progress.

It's possible that LabVIEW **2013** took longer than **120** seconds to launch. Try increasing the timeout value in your VIPM Options.
Also, please verify the **VI Server: Connection** TCP/IP **port** setting in LabVIEW 2013. Make sure this matches the port configured in the VIPM Options.

Figure 17-3: Error during connecting VIPM and LABVIEW

The reason for this error is that the port number in VIPM and LABVIEW is not same. For the solution, you can refer to the link, Solution for the problem. On connection to LABVIEW it opens a message box where "Continue" button needed to click. You will see a screen indicating that LABVIEW will be restarted. When you click okay, LABVIEW will prompt you to save any open VIs as it is restarted. After the completion of the installation, a screen is displayed as shown in Figure 17-4.

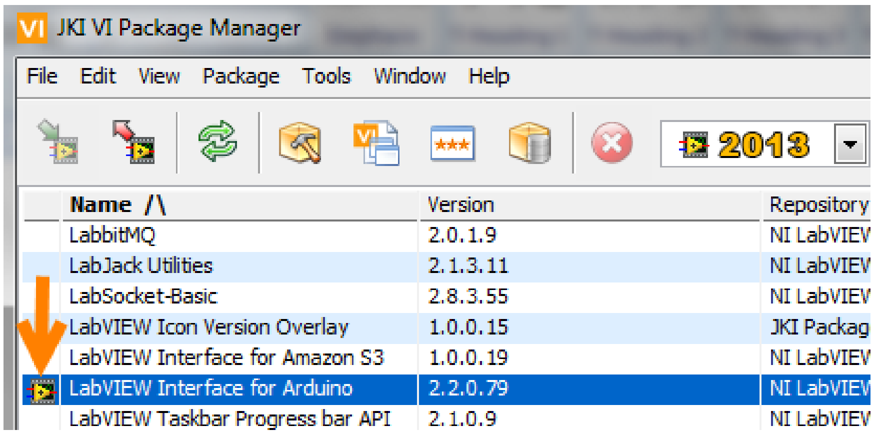


Here are the results of the last action.

Product	Action	Status
✓ LabVIEW Interface for Arduino v2.2.0.79	installed	No Errors

Figure 17-4: Installation Success of LABVIEW Interface for Arduino

Here, "Finish" button is needed to be clicked for the completion of the installation. Once the toolkit is installed, an icon will be displayed in the VPIM as shown in Figure 17-5.


JKI VI Package Manager

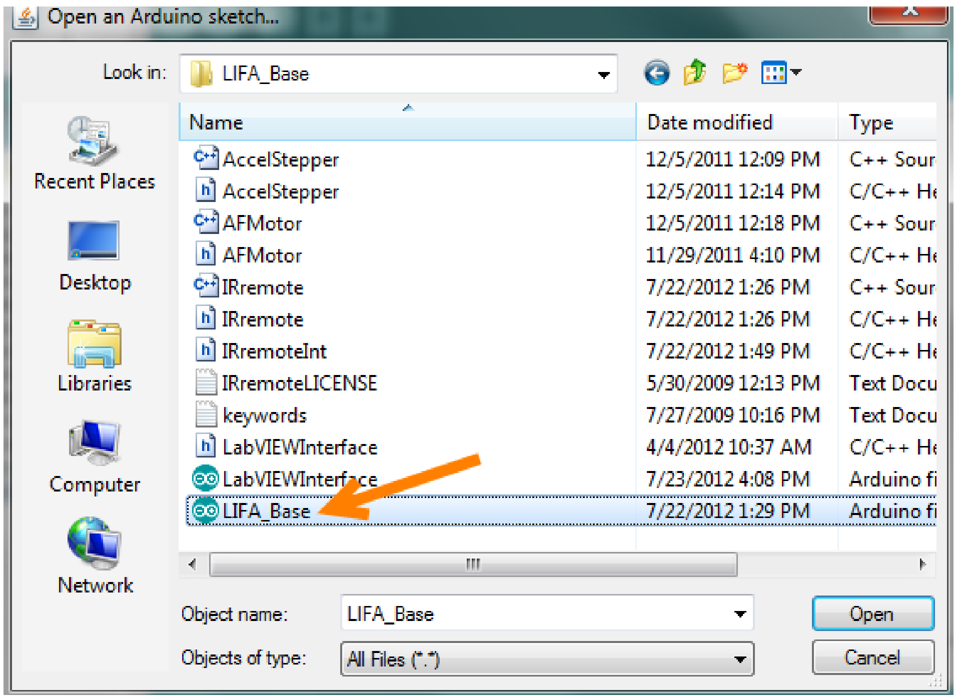
File Edit View Package Tools Window Help

2013

Name /\	Version	Repository
LabbitMQ	2.0.1.9	NI LabVIEV
LabJack Utilities	2.1.3.11	NI LabVIEV
LabSocket-Basic	2.8.3.55	NI LabVIEV
LabVIEW Icon Version Overlay	1.0.0.15	JKI Packag
LabVIEW Interface for Amazon S3	1.0.0.19	NI LabVIEV
LabVIEW Interface for Arduino	2.2.0.79	NI LabVIEV
LabVIEW Taskbar Progress bar API	2.1.0.9	NI LabVIEV

Figure 17-5: Installation completion of the toolkit

5) Connect the Arduino UNO to the PC.
6) Load the LABVIEW Interface for Arduino Firmware onto the Arduino UNO. The firmware can be found in <LabVIEW>\vi.lib\LabVIEW Interface for Arduino\Firmware\LVIFA_Base and arduino IDE is used to deploy this firmware to the Arduino. The steps for loading the firmware are as follow.
   - Open the Arduino IDE
   - Click File » Open and browse to LIFA_Base.ino found in *C:\Program Files\National Instruments\LabVIEW 2013\vi.lib\LabVIEW Interface for Arduino\Firmware\LIFA_Base*


Open an Arduino sketch...

Look in: LIFA_Base

Recent Places
Desktop
Libraries
Computer
Network

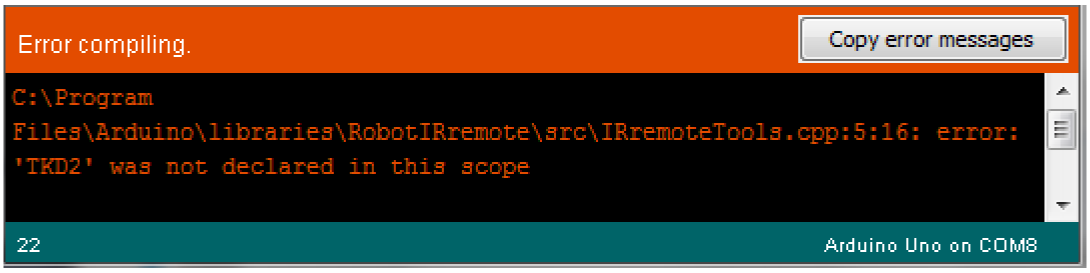
Name	Date modified	Type
AccelStepper	12/5/2011 12:09 PM	C++ Sour
AccelStepper	12/5/2011 12:14 PM	C/C++ He
AFMotor	12/5/2011 12:18 PM	C++ Sour
AFMotor	11/29/2011 4:10 PM	C/C++ He
IRremote	7/22/2012 1:26 PM	C++ Sour
IRremote	7/22/2012 1:26 PM	C/C++ He
IRremoteInt	7/22/2012 1:49 PM	C/C++ He
IRremoteLICENSE	5/30/2009 12:13 PM	Text Docu
keywords	7/27/2009 10:16 PM	Text Docu
LabVIEWInterface	4/4/2012 10:37 AM	C/C++ He
LabVIEWInterface	7/23/2012 4:08 PM	Arduino fi
LIFA_Base	7/22/2012 1:29 PM	Arduino fi

Object name: LIFA_Base
Objects of type: All Files (*.*)

Open
Cancel

Figure 17-6: LABVIEW Interface for Arduino Firmware

When I tried to load the LABVIEW Interface for Arduino Firmware onto the Arduino UNO, I had got an error as shown in Figure 17-7.



```
C:\Program
Files\Arduino\libraries\RobotIRremote\src\IRremoteTools.cpp:5:16: error:
'TKD2' was not declared in this scope
```

Figure 17-7: Error during Uploading Arduino Firmware

The solution for this error is as follow;

1) Go to computer –> go to (C:) –>program files –> Arduino –> libraries –> RobotIRremote
2) Move files IRemote.cpp, IRemote.h, IRemoteInt.h, IRemoteTools.cpp and IRemoteTools.h to the desktop or somewhere out of primary Arduino map (you can even delete them but rather not maybe you will need those files later)

Note: If Arduino map is not in program files than it's where you have saved it.

17.1.2 Database implementation

Database is implemented in SQL server for the logging of the value from sensor/process. In this task, the sensor value is needed to save with the timestamp so that it can be monitored by using Data Dashboard using LABVIEW. So, a single table is enough for the purpose. A table named as "TAGDATA" is created.

The design for the table is shown in Figure 17-8 which is prepared in CA Erwin.

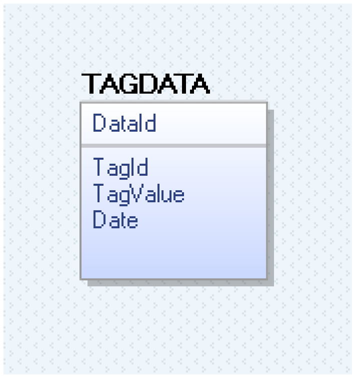


Figure 17-8: Database design in CA Erwin

A database named as "DATALOGGING" is created in SQL server and the table "TAGDATA" is implemented in SQL server. Microsoft SQL Server Management Studio is used for the managing the database.

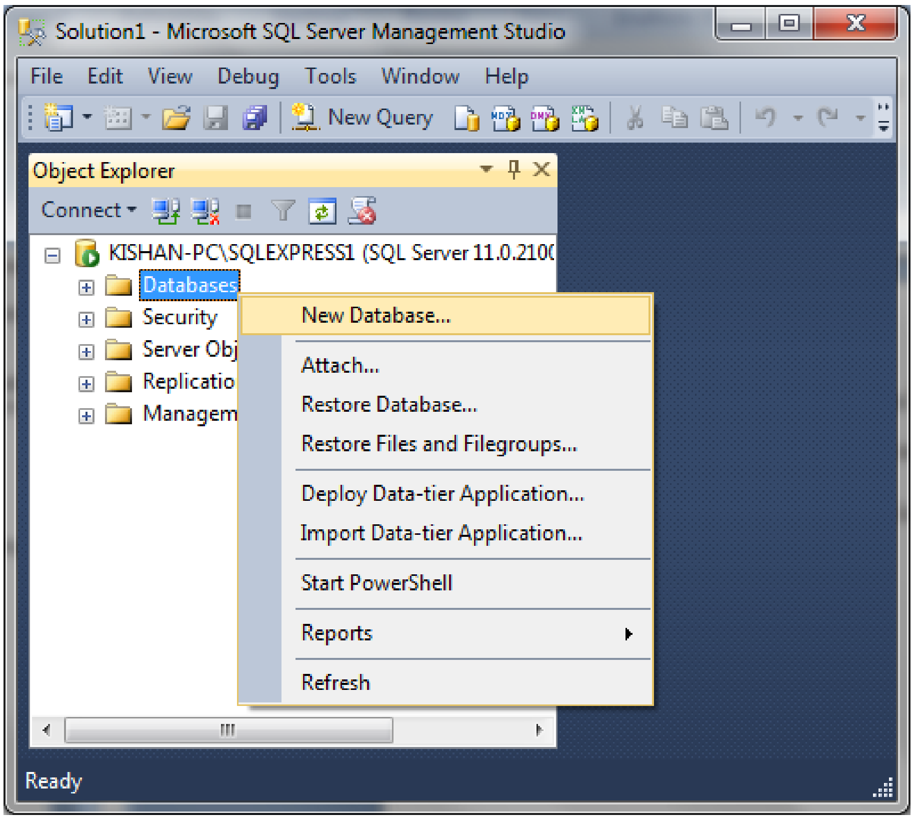
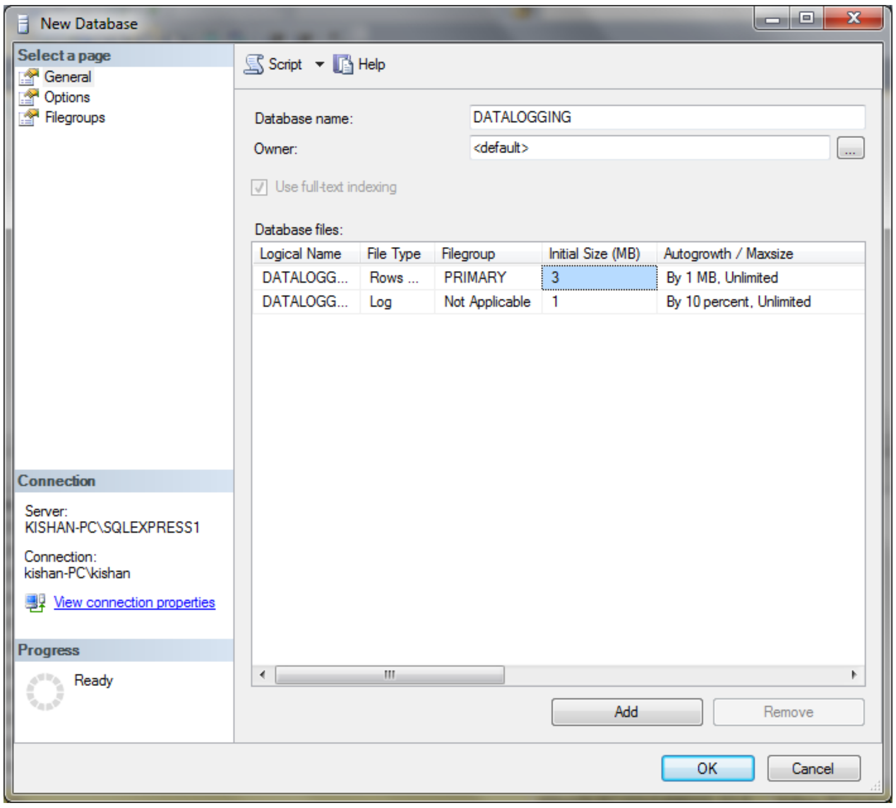

Solution1 - Microsoft SQL Server Management Studio
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
KISHAN-PC\SQLEXPRESS1 (SQL Server 11.0.210
Databases
Security
Server Obj
Replicatio
Managem
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh
Ready

Figure 17-9: Create New Database

As shown in Figure 17-9, for creating a new database, right click on "Databases" and click on "New Database..." which result the page where name of database can be given as shown in Figure 17-10.



Database name: DATALOGGING

Owner: <default>

Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
DATALOGG...	Rows ...	PRIMARY	3	By 1 MB, Unlimited
DATALOGG...	Log	Not Applicable	1	By 10 percent, Unlimited

Connection

Server:
KISHAN-PC\SQLEXPRESS1

Connection:
kishan-PC\kishan

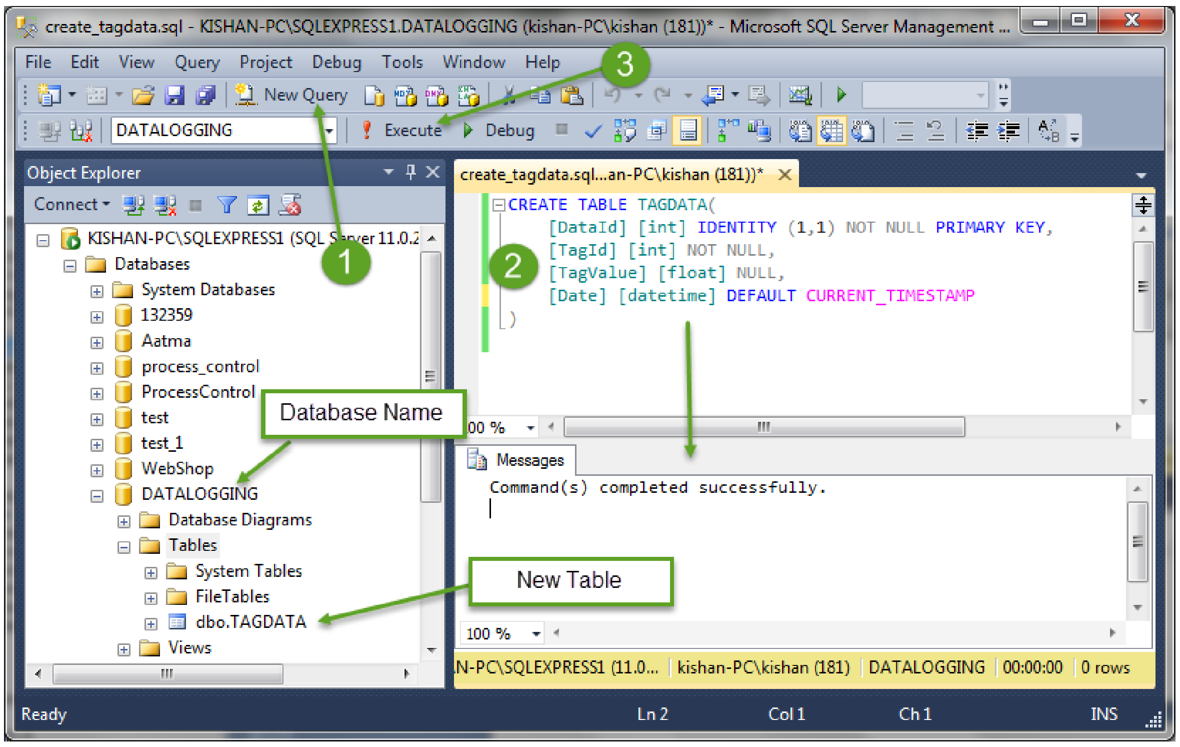
View connection properties

Progress

Ready

Figure 17-10: Screen for Creating New Database

The CREATE TABLE statement is used to create a table in a database and the process to create a table is shown in Figure 17-11.


create_tagdata.sql - KISHAN-PC\SQLEXPRESS1.DATALOGGING (kishan-PC\kishan (181))* - Microsoft SQL Server Management ...
File Edit View Query Project Debug Tools Window Help
New Query
DATALOGGING
Execute
Debug
Object Explorer
Connect
KISHAN-PC\SQLEXPRESS1 (SQL Server 11.0.2
Databases
System Databases
132359
Aatma
process_control
ProcessControl
test
test_1
WebShop
DATALOGGING
Database Diagrams
Tables
System Tables
FileTables
dbo.TAGDATA
Views
create_tagdata.sql...an-PC\kishan (181))*
1
2
3
Database Name
New Table
Messages
N-PC\SQLEXPRESS1 (11.0... kishan-PC\kishan (181) DATALOGGING 00:00:00 0 rows
Ready Ln 2 Col 1 Ch 1 INS

```sql
CREATE TABLE TAGDATA(
    [DataId] [int] IDENTITY (1,1) NOT NULL PRIMARY KEY,
    [TagId] [int] NOT NULL,
    [TagValue] [float] NULL,
    [Date] [datetime] DEFAULT CURRENT_TIMESTAMP
)
```

```
Command(s) completed successfully.
```

Figure 17-11: Process to Create a New Table

17.1.3 Database Connection in LABVIEW

This section includes the database connection in LABVIEW. SQL Toolkit is used in LABVIEW for the database connection. For this, following sections need to be followed.

17.1.3.1 Installation of SQL Toolkit

LABVIEW needs to be installed prior to this Toolkit. It is simple and easy to-use and can run any queries (select, insert, etc). For the installation, following steps are needed to be followed.

1) Download the SQL toolkit. It can be downloaded from the link, Download it as a zip file.
2) Unzip the file
3) Copy "SQLToolkit.mnu" to C:\Program Files\National Instruments\LabVIEW 2013\menus\Categories
4) Copy "SQLToolkit.llb" to C:\Program Files\National Instruments\LabVIEW 2013\vi.lib\
5) The SQL Toolkit is ready to use and in the Functions palette in LabVIEW a new palette named "SQL" will appear (Halvorsen, 2011).

The SQL Toolkit appears in the LABVIEW as shown in Figure 17-12 and can be found as the name “SQL”.

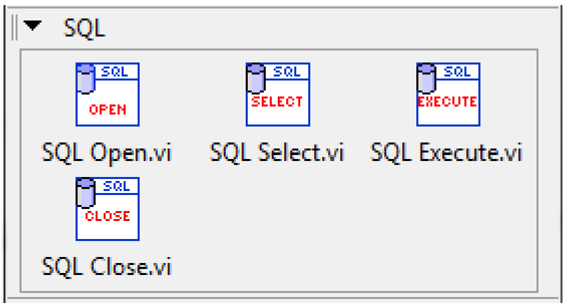
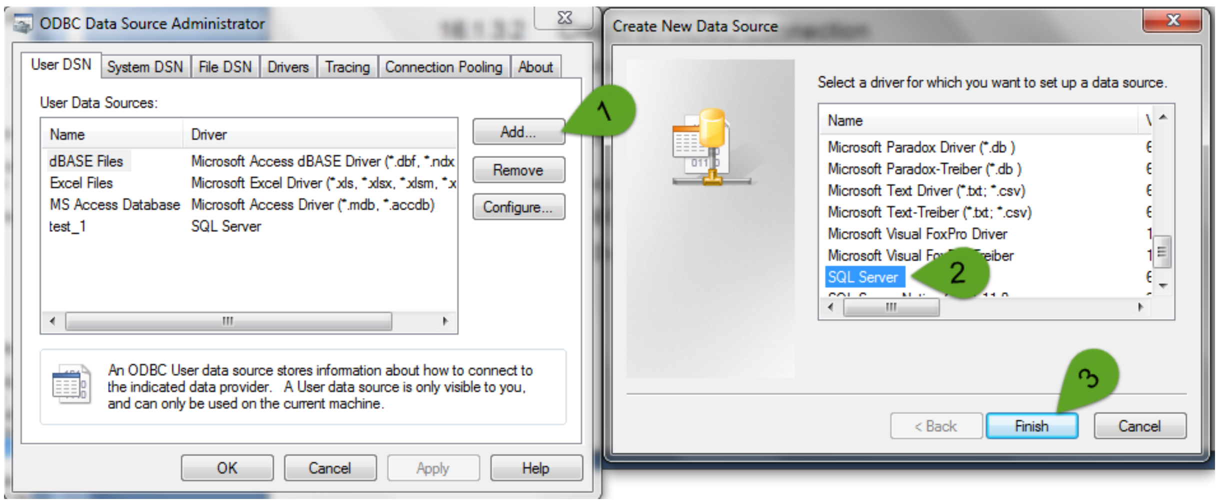


Figure 17-12: SQL Toolkit in LABVIEW

17.1.3.2 Create an ODBC Connection

ODBC (Open Database Connectivity) is a widely accepted application programming interface (API) for database access (Microsoft, 2015). This section includes the steps for the creating the ODBC connection (Halvorsen, 2011).

1) Go to “Control Panel” –> “Administrative Tools” and click to “Data Source(ODBC)”
2) Add a new Data Source, select the SQL Server driver and finally click on “Finish” as shown in Figure 17-13.



User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.x
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
test_1	SQL Server

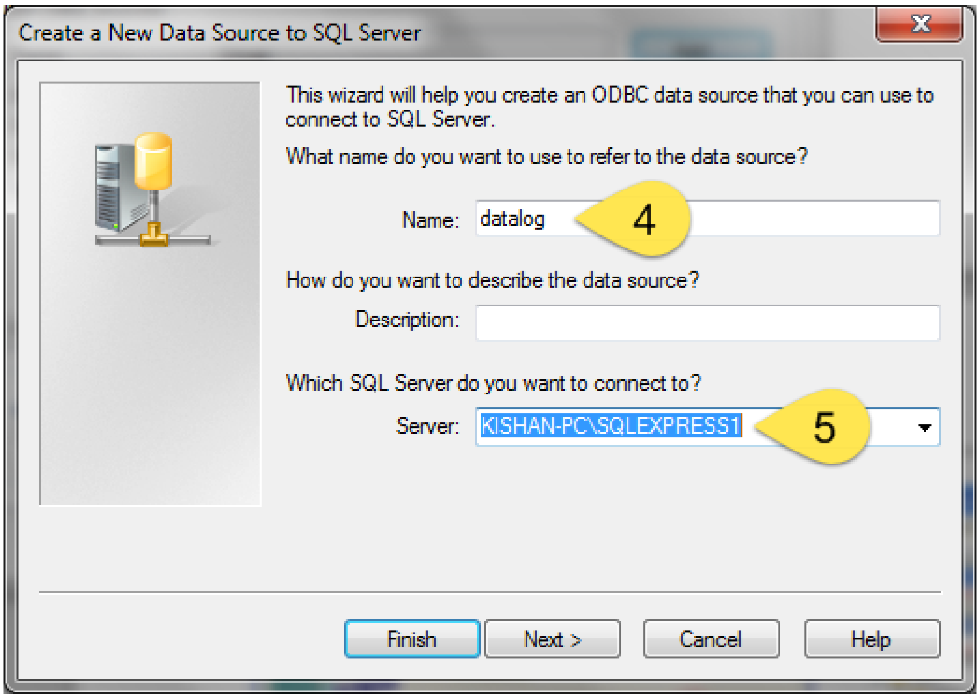
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Select a driver for which you want to set up a data source.

Name
Microsoft Paradox Driver (*.db)
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual Fo[illegible]reiber
SQL Server

Figure 17-13: Add New Data Source

3) Type the data source name and choose the SQL server as shown in Figure 17-14. Here, the data source name is datalog.


Create a New Data Source to SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: datalog

How do you want to describe the data source?

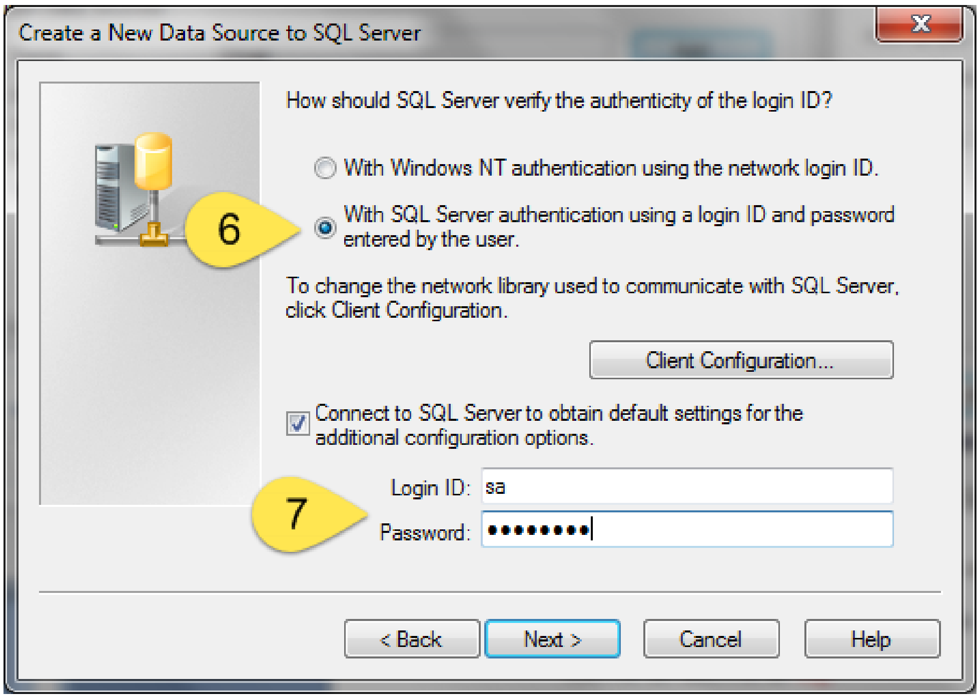
Description:

Which SQL Server do you want to connect to?

Server: KISHAN-PC\SQLEXPRESS1

Figure 17-14: Data Source Name and SQL Server

4) Select SQL Server authentication and type the "sa" (System Administrator) and the password for the sa user are defined during the setup procedure of SQL Server as shown in Figure 17-15.


Create a New Data Source to SQL Server

How should SQL Server verify the authenticity of the login ID?

With Windows NT authentication using the network login ID. [✓ With SQL Server authentication using a login ID and password entered by the user.]

To change the network library used to communicate with SQL Server, click Client Configuration.

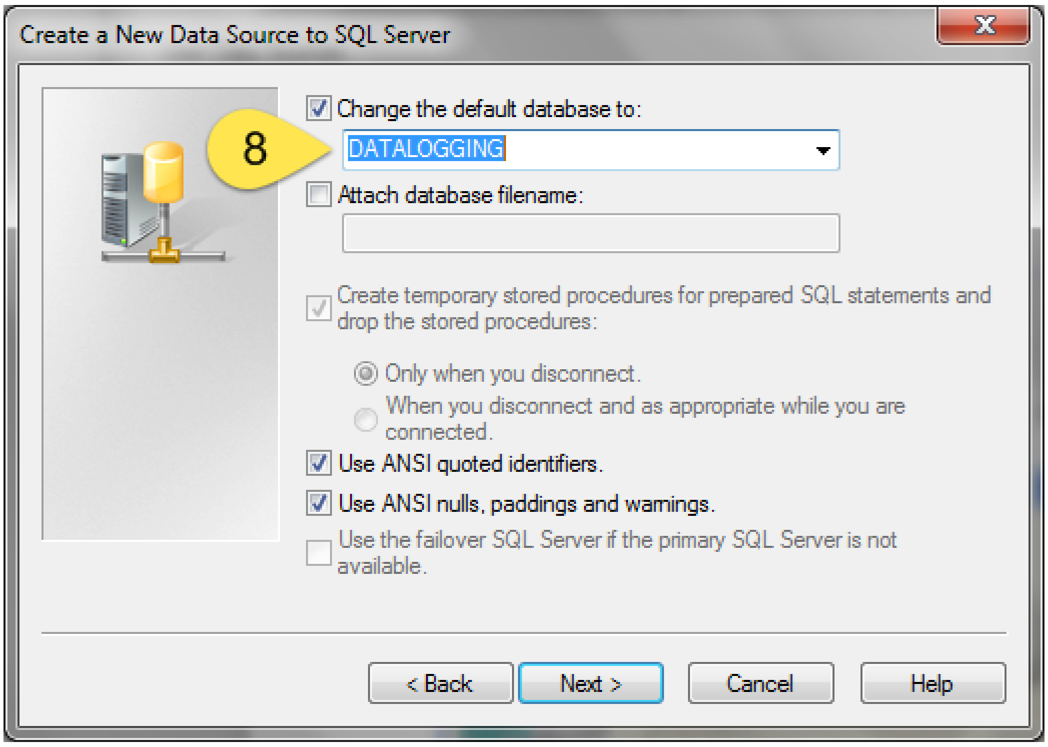
[✓] Connect to SQL Server to obtain default settings for the additional configuration options.

Login ID: sa

Password: ●●●●●●●●

Figure 17-15: SQL Server Authentication and connection to SQL Server

5) Choose the default database and click on button “Next” as shown in Figure 17-16.


Create a New Data Source to SQL Server

[✓] Change the default database to:

DATALOGGING

[ ] Attach database filename:

[✓] Create temporary stored procedures for prepared SQL statements and drop the stored procedures:

(●) Only when you disconnect.

( ) When you disconnect and as appropriate while you are connected.

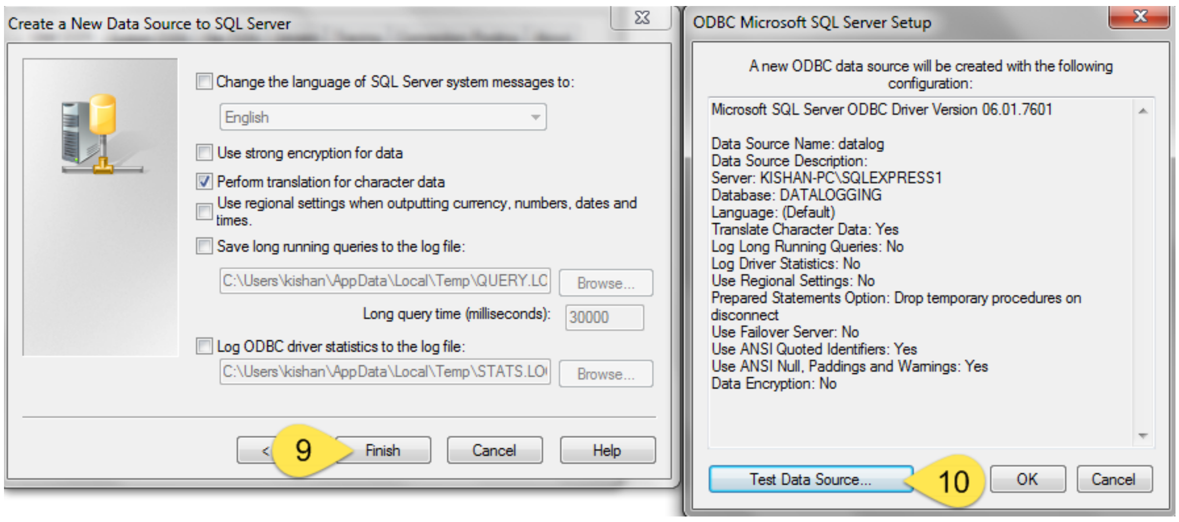
[✓] Use ANSI quoted identifiers.

[✓] Use ANSI nulls, paddings and warnings.

[ ] Use the failover SQL Server if the primary SQL Server is not available.

Figure 17-16: Selection of Defaul Database

6) Finally, complete the configuration and Test the data source to see if it is OK as shown in Figure 17-17.


Create a New Data Source to SQL Server

[ ] Change the language of SQL Server system messages to:

English

[ ] Use strong encryption for data

[✓] Perform translation for character data

[ ] Use regional settings when outputting currency, numbers, dates and times.

[ ] Save long running queries to the log file:

C:\Users\kishan\AppData\Local\Temp\QUERY.LC

Long query time (milliseconds): 30000

[ ] Log ODBC driver statistics to the log file:

C:\Users\kishan\AppData\Local\Temp\STATS.LO

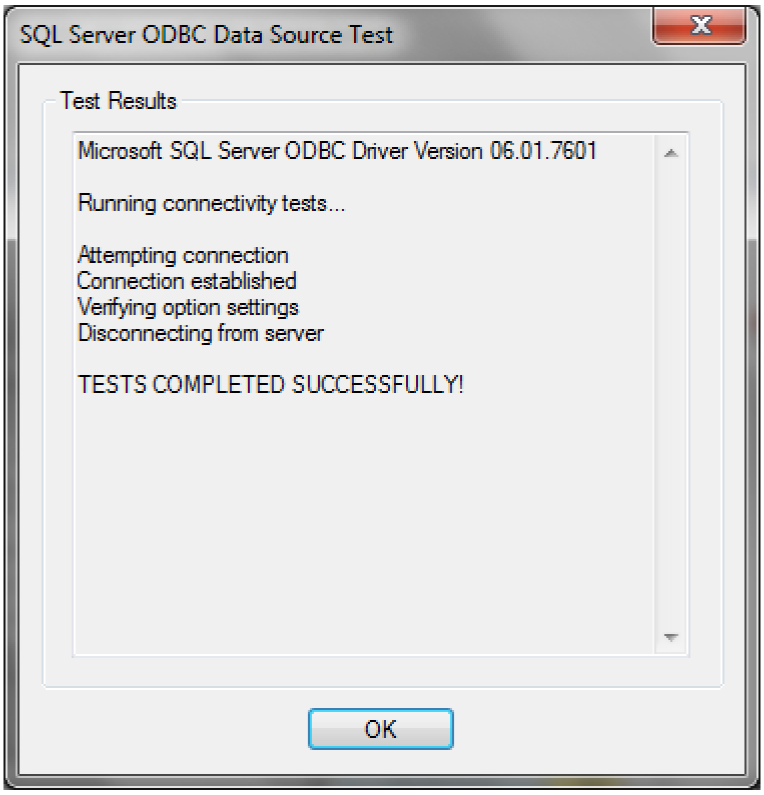
ODBC Microsoft SQL Server Setup

A new ODBC data source will be created with the following configuration:

```
Microsoft SQL Server ODBC Driver Version 06.01.7601

Data Source Name: datalog
Data Source Description:
Server: KISHAN-PC\SQLEXPRESS1
Database: DATALOGGING
Language: (Default)
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Regional Settings: No
Prepared Statements Option: Drop temporary procedures on
disconnect
Use Failover Server: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes
Data Encryption: No
```

Figure 17-17: Completion of Configuration and Test the Data Source

7) At last, click the “Test Data Source” button and the success message will be displayed as shown in Figure 17-18.



```
Microsoft SQL Server ODBC Driver Version 06.01.7601

Running connectivity tests...

Attempting connection
Connection established
Verifying option settings
Disconnecting from server

TESTS COMPLETED SUCCESSFULLY!
```

Figure 17-18: Success Message

This is the completion for the creation of an ODBC Connection and it is ready to be used.

17.1.4 Arduino Connection and LABVIEW Block Diagram

The Arduino setup for connecting sensor is shown in Figure 17-19. Here, I have used thermistor as a temperature sensor to read the temperature of the surroundings and resistor used is 10 kohm.

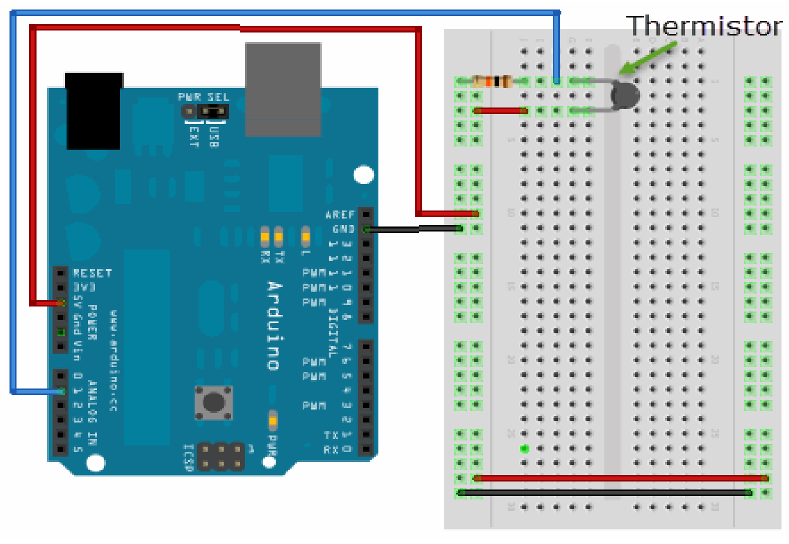


Figure 17-19: Arduino Connection with sensor

After the Arduino connected to the sensor, it is the time to build a LABVIEW program. The LABVIEW block diagram as shown in Figure 17-20 which interfaces the Arduino and reads value of the sensor which is finally stores in database. Hence, it is the logging system built in LABVIEW.

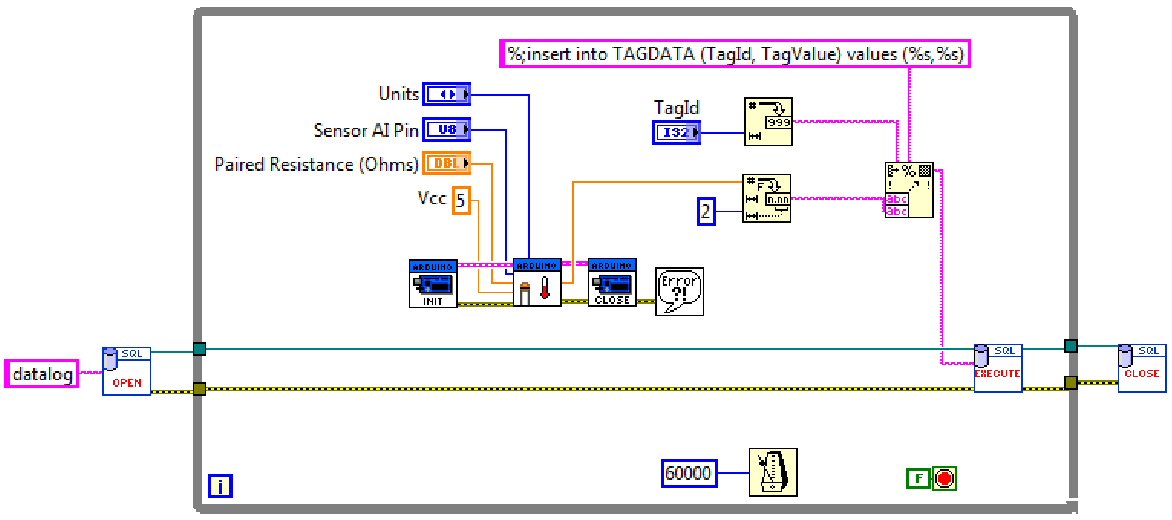


Figure 17-20: Block Diagram in LABVIEW for Logging data from Arduino

It has two parts of blocks; Arduino interfacing part and logging part. The Arduino interfacing part of block diagram is shown in Figure 17-21.

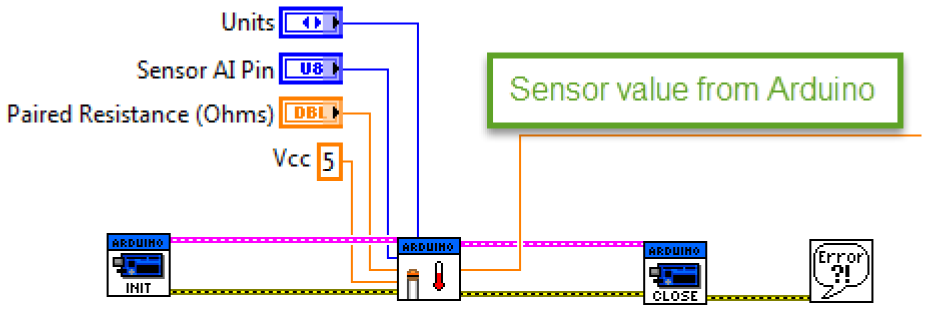


Figure 17-21: Arduino Interfacing Block Diagram

Sensor value read from Arduino is passed to logging part which is shown in Figure 17-22.

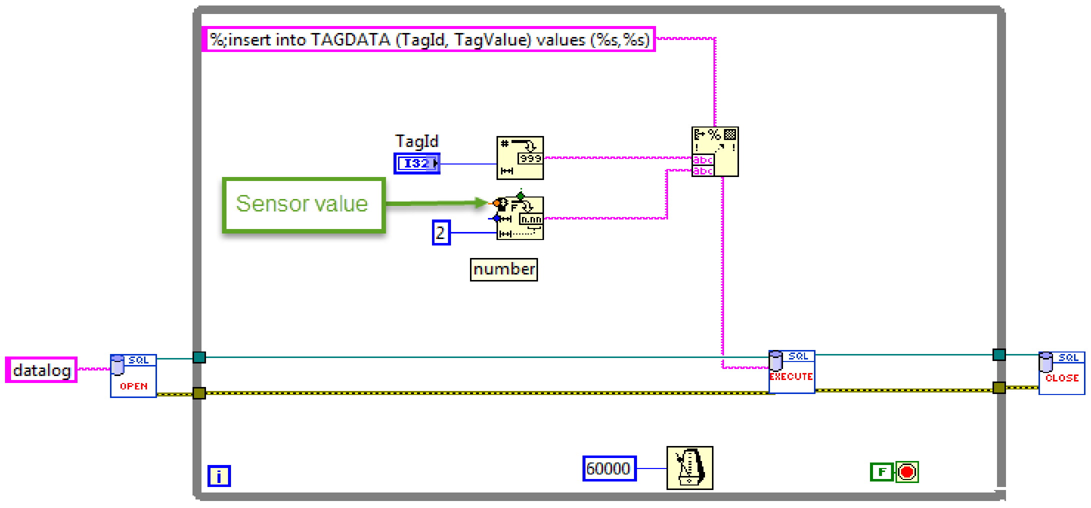
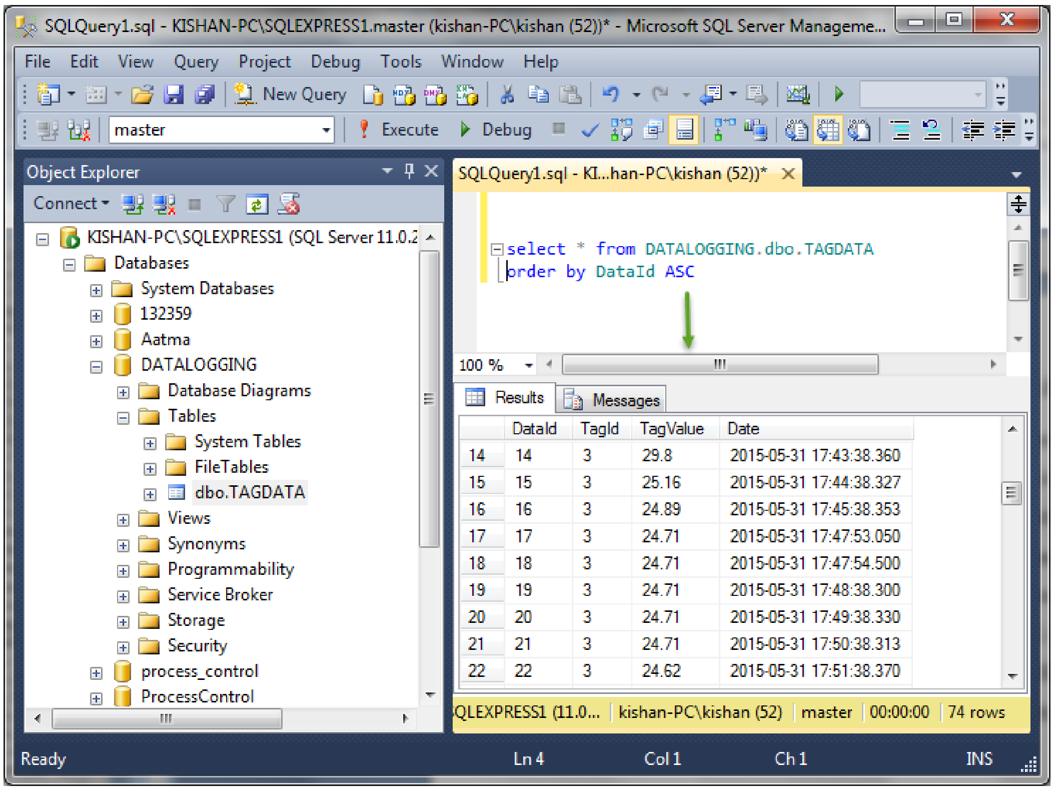


Figure 17-22: Logging Part of the Block Diagram

The point indicated as “sensor value” in the Figure 17-22 is the node where the output (sensor value) of Arduino interfacing part is passed. “SQL OPEN.vi” is used to open a connection to the database specified in the Connection string. The connection string is replaced here by the server name “datalog”. ODBC server is setup for the connection of LABVIEW and database and named as “datalog”. “SQL CLOSE.vi” closes the connection to the database opened by "SQL OPEN.vi" where as “SQL EXECUTE.vi” executes a Query with no return Data, e.g., an INSERT statement. Hence, the INSERT statement is shown in fig is executed by the use of “SQL EXECUTE.vi”. This results the logging of data in the database which is shown in Figure 17-23.



```sql
select * from DATALOGGING.dbo.TAGDATA
order by DataId ASC
```

	DataId	TagId	TagValue	Date
14	14	3	29.8	2015-05-31 17:43:38.360
15	15	3	25.16	2015-05-31 17:44:38.327
16	16	3	24.89	2015-05-31 17:45:38.353
17	17	3	24.71	2015-05-31 17:47:53.050
18	18	3	24.71	2015-05-31 17:47:54.500
19	19	3	24.71	2015-05-31 17:48:38.300
20	20	3	24.71	2015-05-31 17:49:38.330
21	21	3	24.71	2015-05-31 17:50:38.313
22	22	3	24.62	2015-05-31 17:51:38.370

QLEXPRESS1 (11.0... | kishan-PC\kishan (52) | master | 00:00:00 | 74 rows

Figure 17-23: Data Logging in the Database

17.2 Web Service

Web service is created for data publication in the web. The data is extracted from the database by the use of SQL toolkit and finally share on the web by the use of web service in LABVIEW. The published data can be accessed from anywhere by the use of internet and it can be accessed by the use of data dashboard from Apple Ipad, Iphone and Android mobile. In this thesis, web service named as “datalog” is created.

17.2.1 Creating Web Service

The steps for creating a web service in LABVIEW are described as follows (Instruments, 2013).

1) Create a LABVIEW project to organize the Web service files. Save the project as “datalog”.
2) Right-click “My Computer” and select “New−>Web Service”. LABVIEW adds a Web service project item and folders under the target.
3) Rename web service as “datalog” by right-clicking on the web service and selecting “Rename”. Hence, a web service named as “datalog” is created as shown in Figure 17-24.

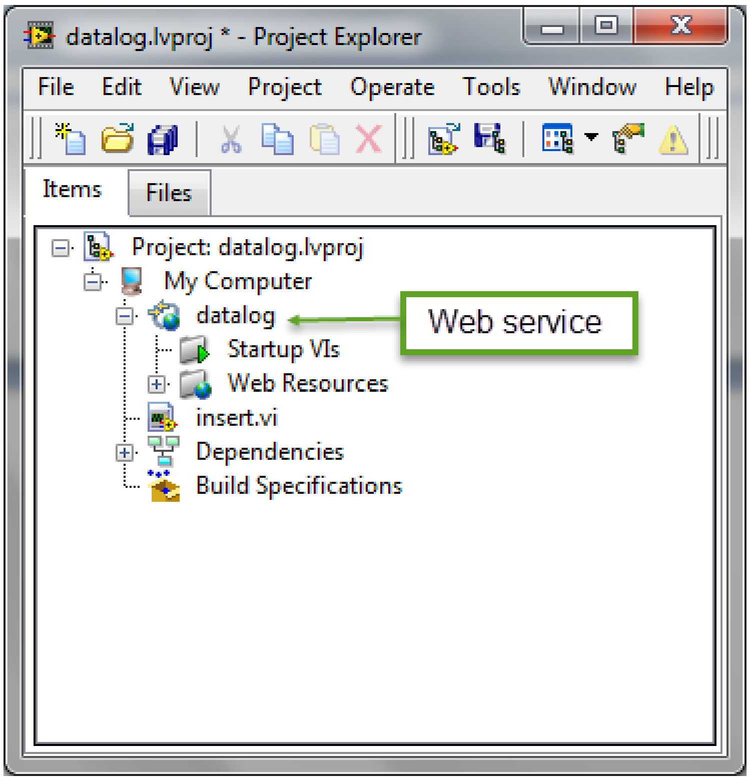
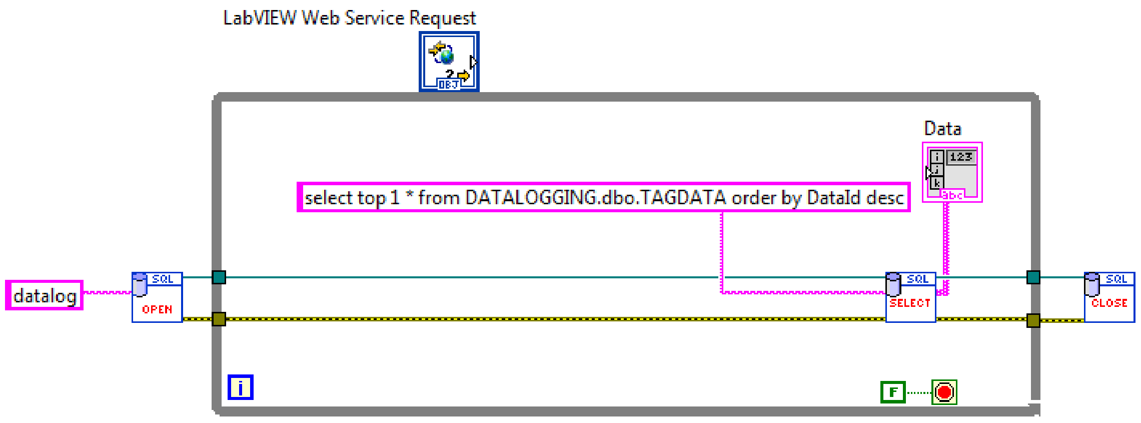


Figure 17-24: Creating a Web Service "datalog"

4) Right-click “Web Resources" and select “New VI” to create a new HTTP method VI, which is a VI that receives HTTP requests from clients and returns data to clients.
5) The step 4 opens a VI with LABVIEW web service request and the required block diagram is created for extracting data from database in this same VI. The VI for the HTTP method is named as “select.vi”. This includes the block diagram, front panel and connector pane. The block diagram of the HTTP method.vi is shown in Figure 17-25.



LabVIEW Web Service Request
Data
select top 1 * from DATALOGGING.dbo.TAGDATA order by DataId desc
datalog
OPEN
SELECT
CLOSE

Figure 17-25: Block Diagram of HTTP Method.vi

The front panel is shown in Figure 17-26.

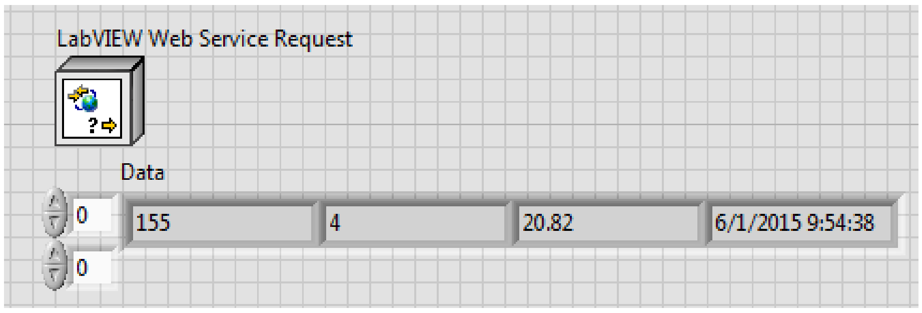


Figure 17-26: Front Panel of HTTP Method.vi

And the connector pane is connected to the data indicator. The Figure 17-27 shows the process to link the indicator with the connector pane. For the connection, click on the pane and click on the respective indicator.

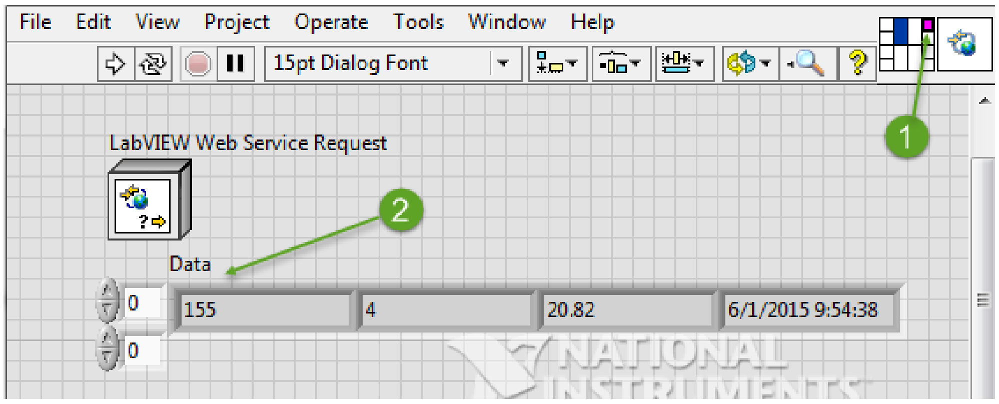


Figure 17-27: Connector Pane of HTTP Method.vi

6) Save the VI as select.vi and the project looks as shown in Figure 17-28.

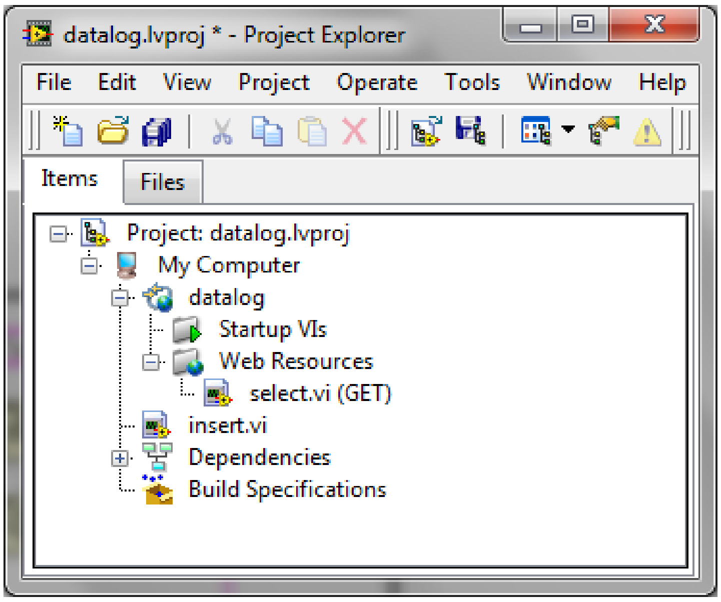


Figure 17-28: Project Final out Look

17.2.2 Testing and Debugging the Web Service

Before publishing the Web service to a target, it is better to test the HTTP method VI if it communicates with clients as expected or not. The steps for testing briefly described as follows (Instruments, 2013).

1) Right-click the Web service project item, "datalog" and click "Start".
2) Open select.vi and check the Run button. There we can see something like arrow that indicates the VI is reserved for execution. It is shown in Figure 17-29. And, finally close the select.vi.

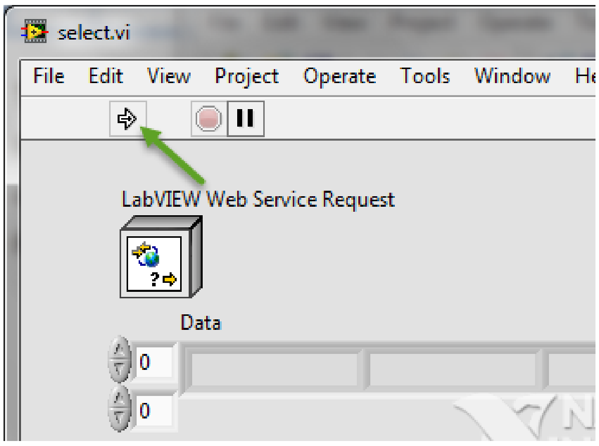
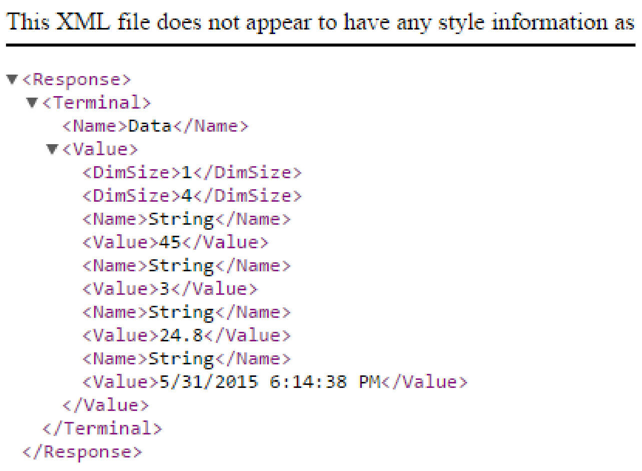

select.vi
File Edit View Project Operate Tools Window He
LabVIEW Web Service Request
Data
0
0

Figure 17-29: Indication for execution of select.vi

3) Right-click select.vi and select "Show Method URL" to display the HTTP Method URL dialog box. In the Available Servers drop-down menu, select "Local Debugging" and then click the "Copy URL" button and close the dialog box.
4) Paste the URL from step 3 into a standard web browser and see the result. It must displays the XML response with the name and output value of each indicator. The URL in this thesis is such as: http://127.0.0.1:8002/datalog/select and the XML response is shown in Figure 17-30.


This XML file does not appear to have any style information as

```
▼<Response>
  ▼<Terminal>
      <Name>Data</Name>
    ▼<Value>
        <DimSize>1</DimSize>
        <DimSize>4</DimSize>
        <Name>String</Name>
        <Value>45</Value>
        <Name>String</Name>
        <Value>3</Value>
        <Name>String</Name>
        <Value>24.8</Value>
        <Name>String</Name>
        <Value>5/31/2015 6:14:38 PM</Value>
      </Value>
    </Terminal>
  </Response>
```

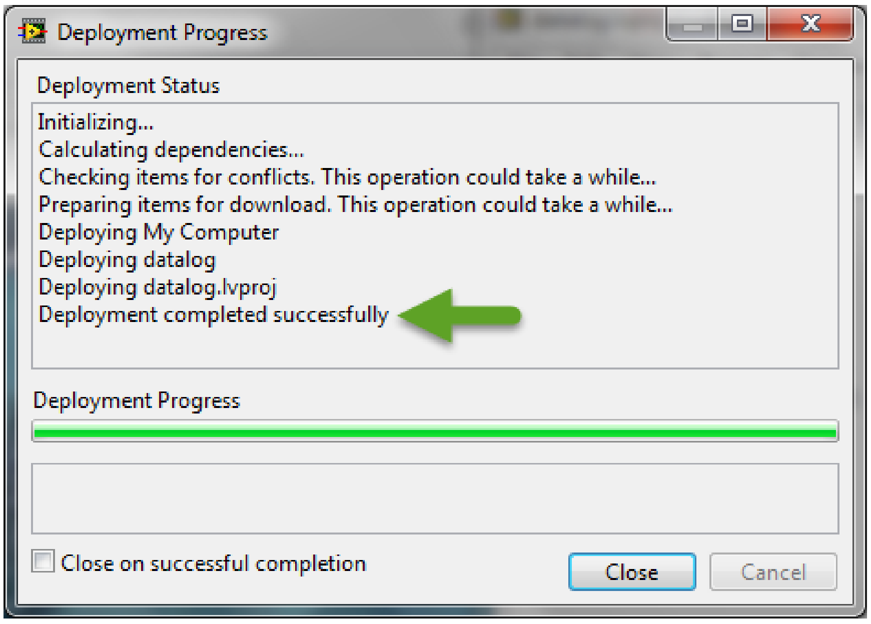
Figure 17-30: XML response

5) To stop it, right-click the Web service project item, “datalog” and select “Stop”.

17.2.3 Publishing a Web Service

LABVIEW publishes stand-alone Web services called the Application Web Server. The Application Web Server hosts stand-alone Web service applications on a network and provides multiple security-related features to protect the network data exchange, including Secure Socket Layer (SSL) encryption (Instruments, 2013). We should follow the following steps to publish the Web service to the Application Web Server.

1) Right-click the Web service project item and hover over “Application Web Server” and click on “Publish” which results a dialog box showing progress.
2) If the LABVIEW publishes web service successfully, close the dialog box. The success dialog box is shown in Figure 17-31.



```
Initializing...
Calculating dependencies...
Checking items for conflicts. This operation could take a while...
Preparing items for download. This operation could take a while...
Deploying My Computer
Deploying datalog
Deploying datalog.lvproj
Deployment completed successfully
```

Figure 17-31: Success of Deployment Progress

After the completion of deployment, clients can access the data from anywhere using internet.

17.2.4 Accessing the Web Service with a Client

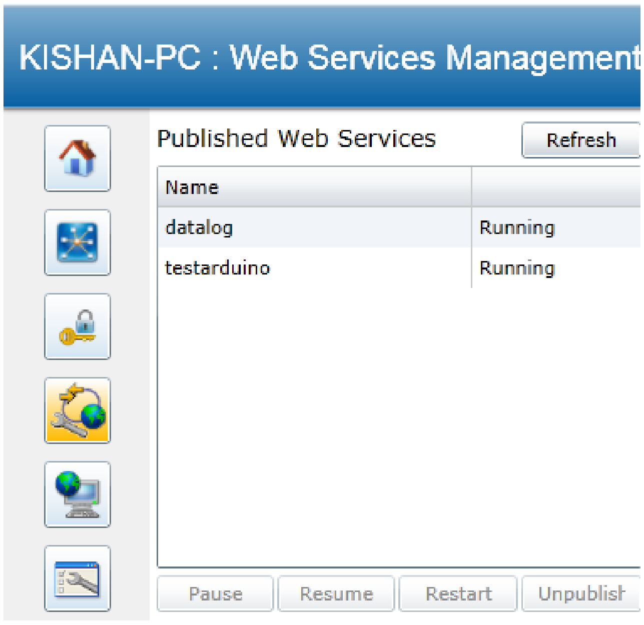
For accessing the web service with a client, following steps are needed to be followed.

1) Right-click select.vi and select “Show Method URL” to display the HTTP Method URL dialog box.
2) Select “Application” in “Available Servers” drop-down menu and copy the URL by clicking on “Copy” button. The port number in the URL is different than the URL for the debugging server. This port number is used for communication with the Web service files running on the Application Web Server (Instruments, 2013).
3) Close the dialog box.
4) Paste the copied URL to the standard browser. The URL for the project is: http://127.0.0.1:8080/datalog/select. The browser returns the same XML response as shown in Figure 17-30S. If the web service is needed to access from another computer then “127.0.0.1” must be replaced by the IP address of the server computer where web service is installed.

17.2.5 Monitoring the Web Service

For monitoring the web service, following steps are needed to be followed.

1) Right click on the Web service project item “datalog” and hover over “Application Web Server” and click on “Manage Web Server” which leads to open a default browser.
2) Click on “Web Services Management” on the left side of the open page in the browser. Then, all the published web services are listed as shown in the Figure 17-32. If the list is not displayed, then click on “Refresh” button.



KISHAN-PC : Web Services Management

Published Web Services

Name	
datalog	Running
testarduino	Running

Figure 17-32: List of Published Web Services

3) Click the “datalog” to see the mapping and all the buttons (Pause, Resume, Restart and Unpublish) will be activated which are used to pause, resume, restart, and unpublish Web services respectively.

17.3 Data Dashboard

Data dashboard is an application developed for apple and android IOS. It supports in Aple and android mobile and Apple Ipad. More features can be viewed in Apple Ipad. It displays the values of network-published shared variables and deployed LABVIEW Web services on indicators, such as charts, gauges, textboxes, and LEDs.

For the use of dashboard, following steps are must be followed.

1) Download the application in the device (mobile or Ipad). In my case, I had downloaded in android mobile.
2) Open the application and the first page of the application is shown in Figure 17-33.

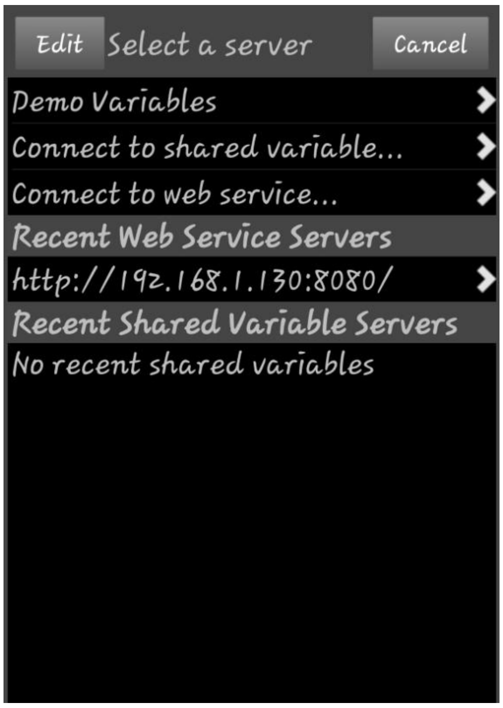

Edit
Select a server
Cancel
Demo Variables
Connect to shared variable...
Connect to web service...
Recent Web Service Servers
http://192.168.1.130:8080/
Recent Shared Variable Servers
No recent shared variables

Figure 17-33: Home page for Data Dashboard application in Android Mobile

3) Click the option “connect to web service” and a screen will appear where we can enter the IP address. The screen shot is shown in Figure 17-34.

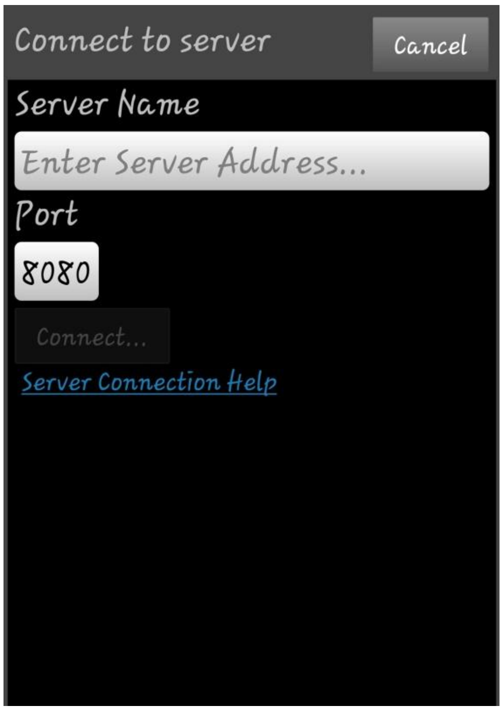


Figure 17-34: Screen to Enter Server Address

Here, IP address of the server where web service is installed must be enter and click on “connect” button to view the web services.

18 PID Controller Library in Arduino

PID controller is developed in Arduino and the sketch is created as a library in Arduino. And finally, the library is used for controlling a process (Air heater). The file for the PID Arduino library is attached in Appendix E.

18.1 Developing PID Controller Code

For the development of Arduino library, it needs the header file and CPP file. The header file consists of a class with all the constants and functions declaration whereas the CPP file consists of the definition of all the functions. The files are developed in Microsoft Visual Studio 2010.

The main functions in PID controller are listed in the Table 18-1 with their tasks.

Table 18-1: List of functions in PID controller

PID	It is a constructor, sets some parameter manually such as PID tuning parameters, sampling time, etc
Compute	Compute the new PID output and decides whether to calculate new PID output or not. It is run in loop with sampling time as an interval.
SetTunningParameter	Set tuning parameters of PID controller, K_p, K_i and K_d and can set dynamically as well.
SetSamplingTime	Sets the sampling time for the control loop in mili-seconds and it is set in constructor.
SetOutputLimits	Set output limits (0-255)
Setmode	Set mode for the controller. Automatic or manual
Initialize	Does all the things need to be happened to ensure a bump less transfer from manual to automatic mode
SetControllerDirection	Set the direction for the controller. DIRECT or REVERSE DIRECT means output increase with input increase INVERSE means output increase with input decrease

18.2Use PID controller as Library in Arduino

As the coding is finished and both the header and CPP file are copied to a folder and named it as "PID". To add this "PID" library in Arduino, it is needed to the directory of library as shown in Figure 18-1. The directory for the library in Raspberry Pi is */usr/share/arduino/libraries/*.

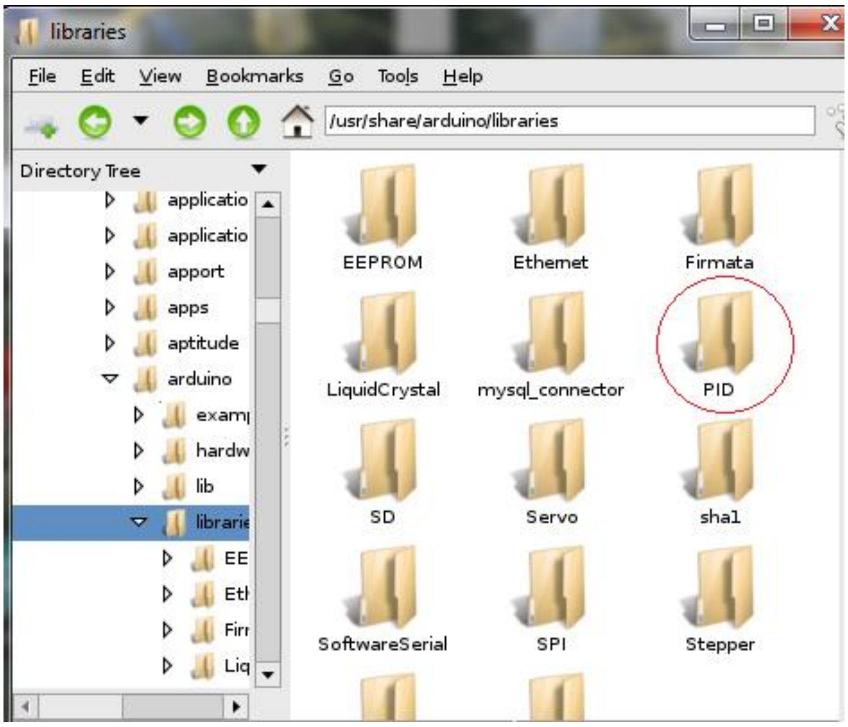


Figure 18-1: Directory for library of Arduino

18.3Use of PID Controller

The analog input of the Arduino is used to read the value from the Air heater. Voltage Analog Input (Temp 1) of Air Heater is connected to A_0 and GND in the Arduino. So, analog input (A_0) is read in Arduino sketch which is converted to required unit (Degree Celsius in this project). This converted value will be the input for the PID controller. Set point is set manually whereas tuning parameters are set manually but they can be changed while program is running. As, all the parameters are set for PID functions, PID functions are called. As a result, PID output is generated which is written to PWM digital pin 9 (Air_Heater) in Arduino and this pin is connected to voltage analog output pin of Air Heater. This voltage (output of PID) controls the heating element of the Air heater so that the process value (input to PID controller) becomes closer to desired Set point. This process is set in loop until the process value is very close to the set point.

The connection between the Air Heater and the Arduino is shown in Figure 18-2.

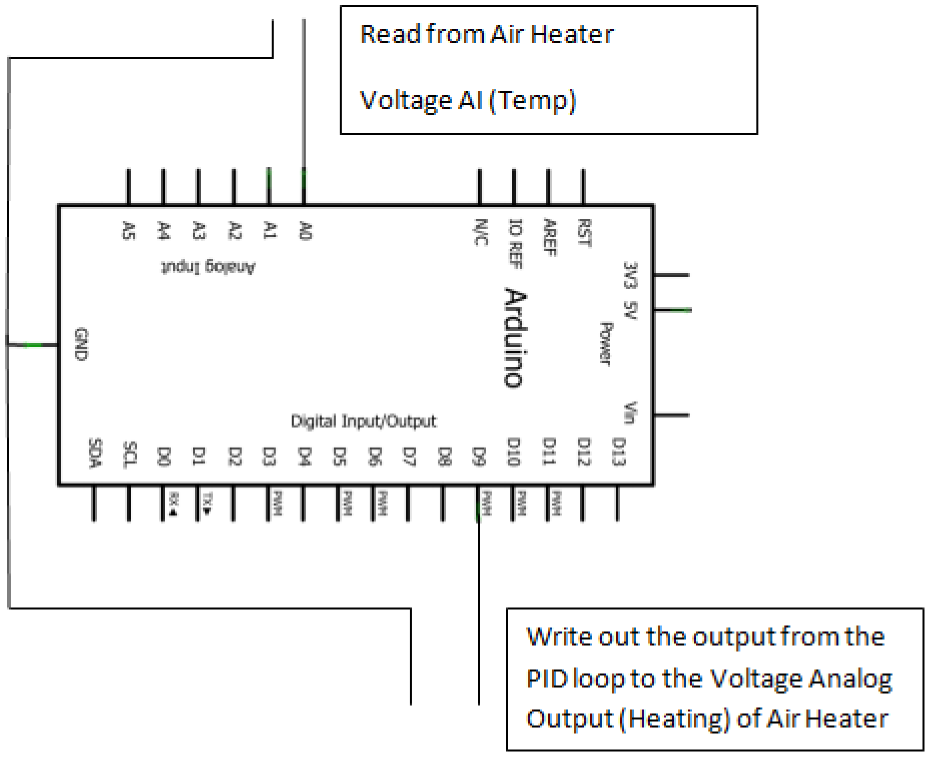
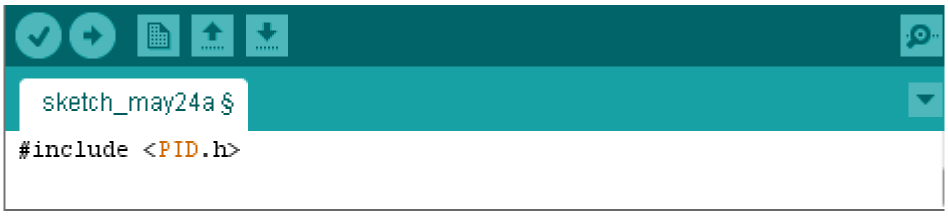


Figure 18-2: Connection of Arduino and Air Heater for PID controller

PID library is used to control the process (Air Heater). To include it in the Arduino sketch, open Arduino IDE, click on “Sketch” and import library “PID”. You will get a code added in the sketch as shown in Figure 18-3. Now, it will be able to use the functions of PID controller.



```
#include <PID.h>
```

Figure 18-3: Including PID library in a sketch

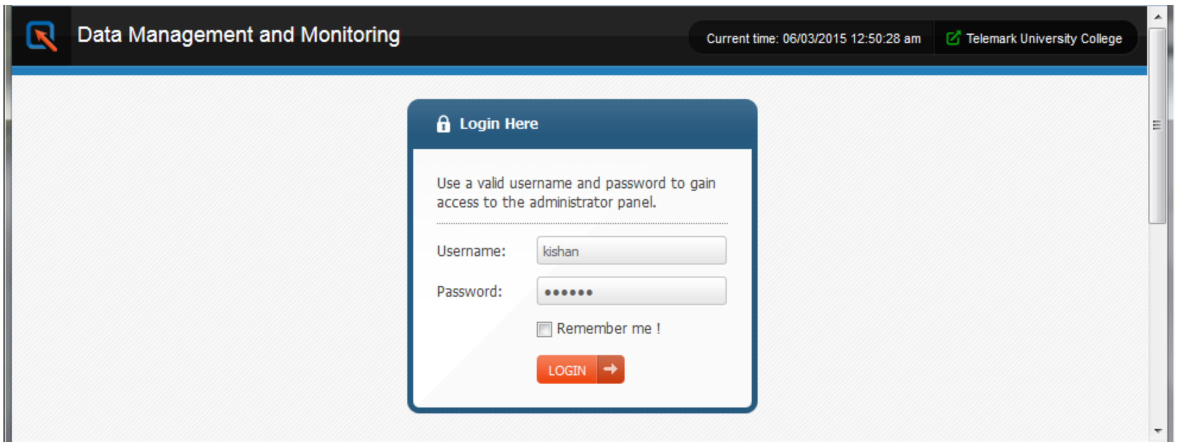
Part IV: Result

This chapter includes the results of the tasks of the thesis.

19 Website

Website is developed for the data monitoring and management purposes. The process value is logged in the database by the use of Arduino and Raspberry Pi and website is developed in Raspberry Pi for the purposes. Website is available in this link: http://128.39.35.237/DMM/.

There is login system in the website and need to login for accessing it. The login page is shown in Figure 19-1.



Data Management and Monitoring
Current time: 06/03/2015 12:50:28 am
Telemark University College
Login Here
Use a valid username and password to gain access to the administrator panel.
Username: kishan
Password: ••••••
Remember me !
LOGIN

Figure 19-1: Log in prompt for website

If the correct username and password is given, it will be redirected to the home page of the website which is shown in Figure 19-2.

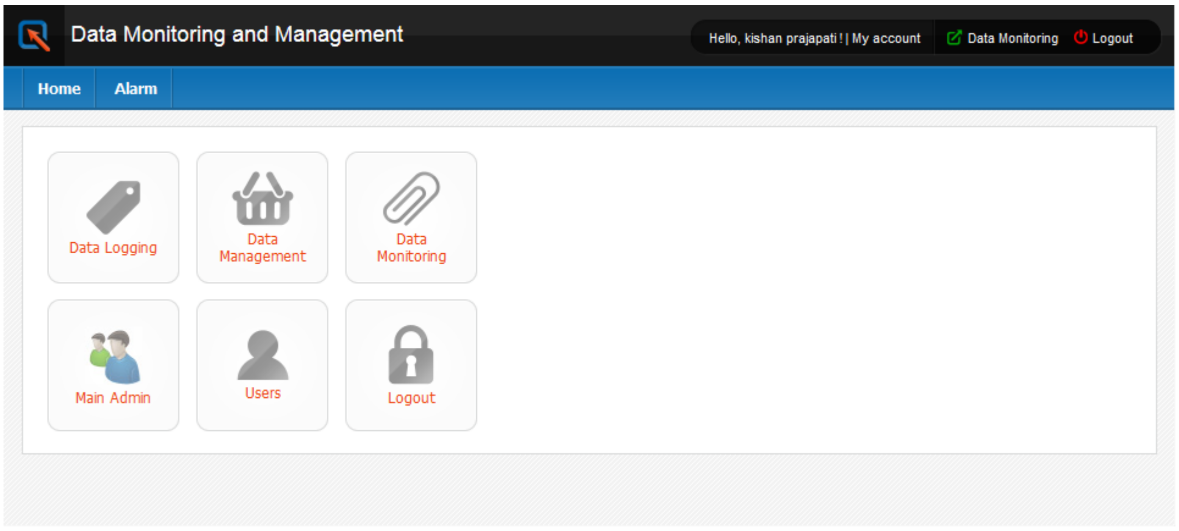


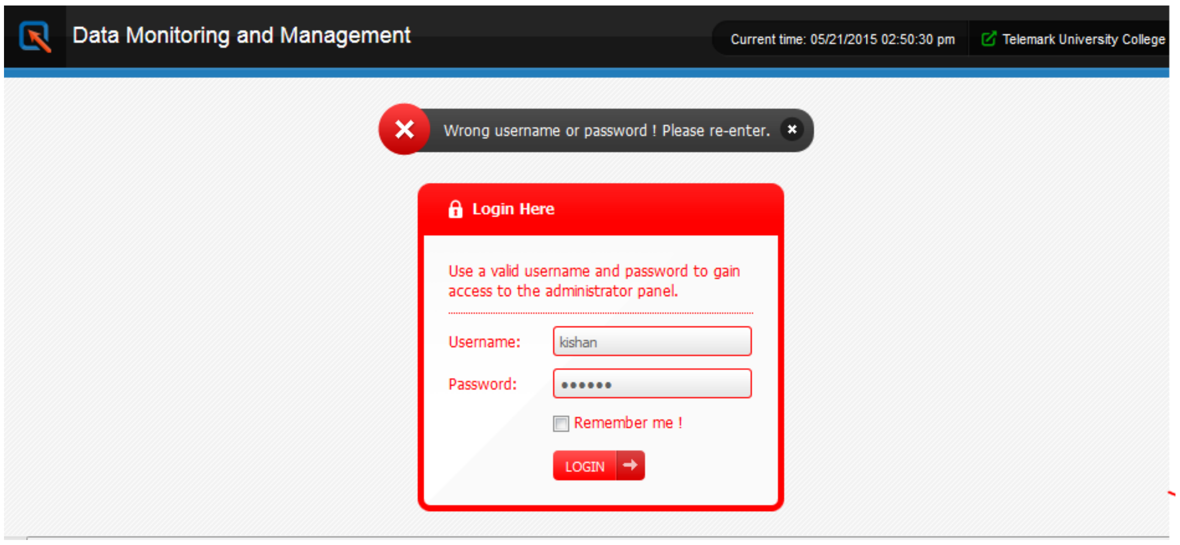
Figure 19-2: Home page of the website

In home page, there are 5 main sub systems and they are as follow.

1) User Management: Manage the users of the system.
2) Data Logging: Configure logging time for a tag and check the status.
3) Data Monitoring: Monitor the tag value in table and as a trend.

4) Data Management: Manage the tag and device for the system.
5) Alarm Management: List the active alarm and option to acknowledge it.

And if there is error then it will redirect to error page. The error page is shown in Figure 19-3.


Data Monitoring and Management
Current time: 05/21/2015 02:50:30 pm
Telemark University College
Login Here
Remember me !
LOGIN

Wrong username or password ! Please re-enter.

Use a valid username and password to gain access to the administrator panel.

Username: kishan

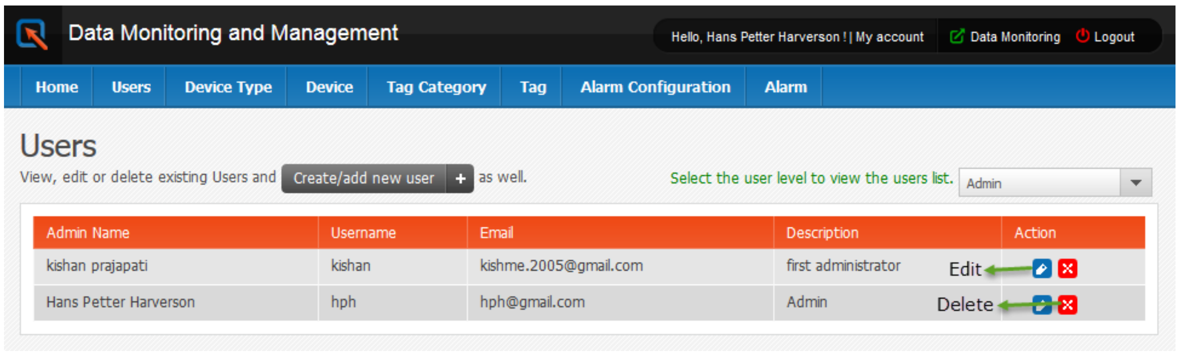
Password: ••••••

Figure 19-3: Error page during log in

19.1 Users management

There are three levels of the users for the system and they are;

1) Admin: Can access all the sections, user management, data management, alarm management and monitoring part.
2) Operator: Can access alarm management and data monitoring part only.
3) Visitor: Can monitor data only.



View, edit or delete existing Users and Create/add new user as well.

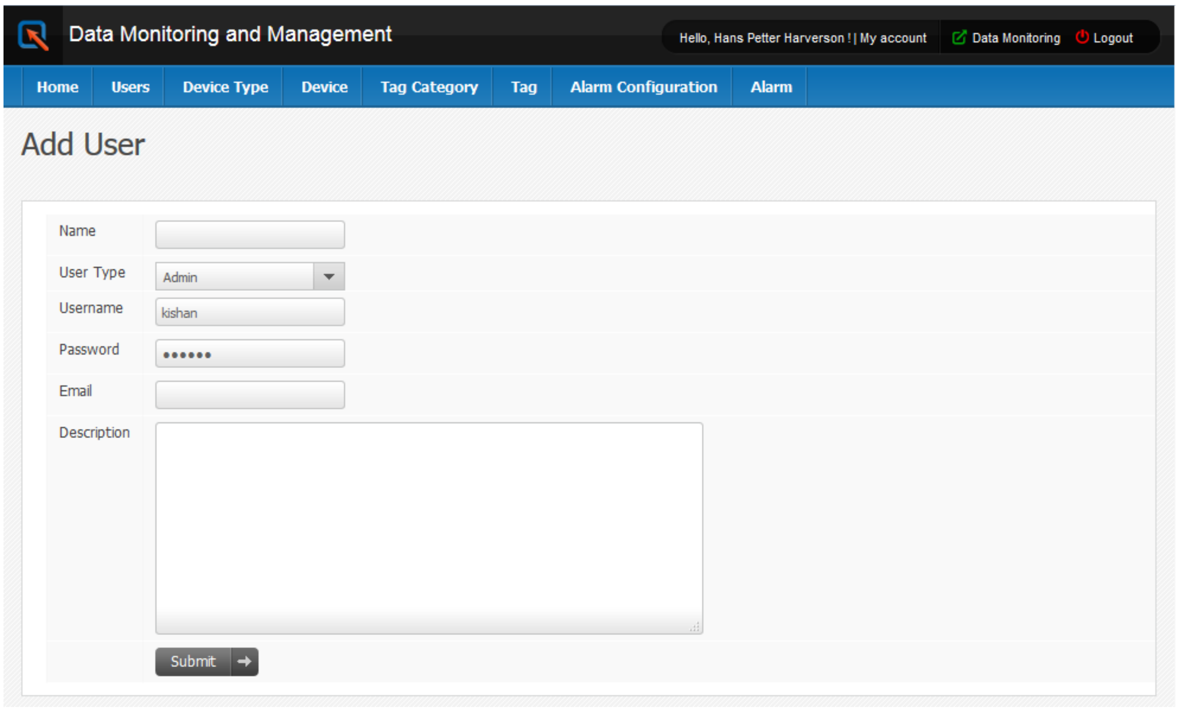
Select the user level to view the users list. Admin

Admin Name	Username	Email	Description	Action
kishan prajapati	kishan	kishme.2005@gmail.com	first administrator	
Hans Petter Harverson	hph	hph@gmail.com	Admin	

Figure 19-4: User Management System

The users' management page is shown in Figure 19-4 where two users (admin user type) are listed in the table. There is button "Create/add new user" which is used for creating a new users and each row consists two buttons for editing and deleting the records under

“Action” column which are used for editing and deleting a record. And, there is a select button at the right of the page where it is possible to choose the user type. According to user type chosen, the list of users is displayed in the table.



Add User

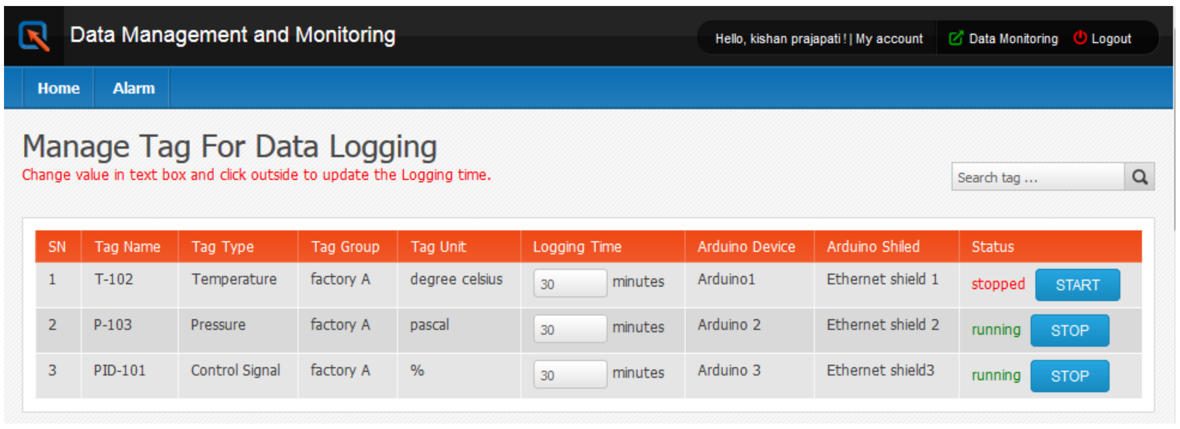
Name	
User Type	Admin
Username	kishan
Password	••••••
Email	
Description	

Figure 19-5: Creating New User

Figure 19-5 shows the page for creating new user for the system. Username should be unique in this case, so during entering username, it is checked in database using jQuery. If it is unique, “okay” key word is displayed otherwise “Already used” is displayed and asked to enter other username.

19.2 Configuration for Data Logging

Before the data logging, some configurations are needed to be made; logging time (Sampling time) and alarm limit. For configuration of logging time, there is a table shown in Figure 19-6 which displays after the click of “Data logging” button in home page. To change the logging time, we just need to change the value in text box and click outside the page. And two buttons can be noticed as “START” and “STOP” which enable and disable the logging status of the respective tag. As logging starts, the status changes as “Running” and after the click of “STOP” button, logging stops.


Data Management and Monitoring
Hello, kishan prajapati ! | My account
Data Monitoring
Logout
Home
Alarm
Search tag ...

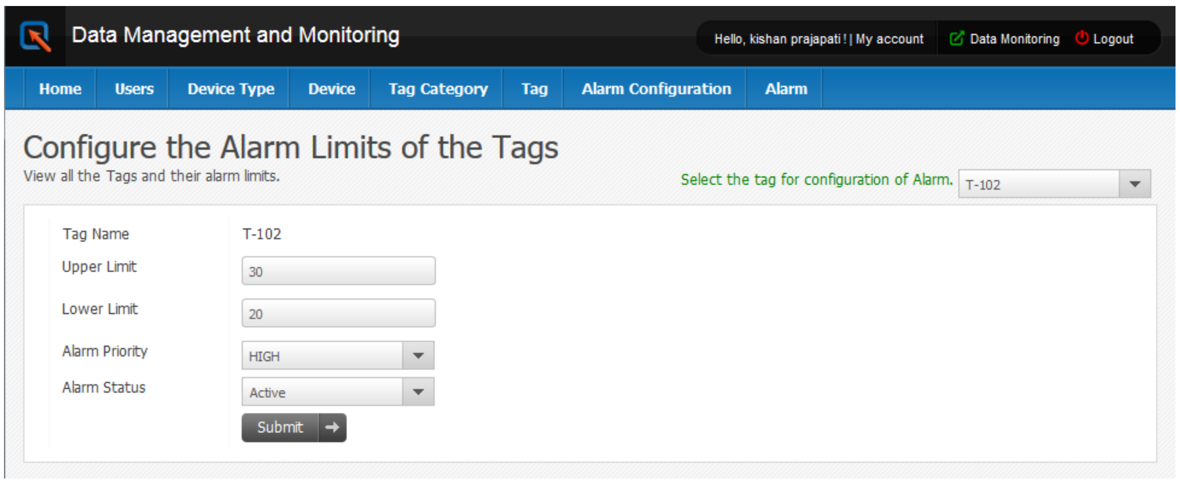
Manage Tag For Data Logging

Change value in text box and click outside to update the Logging time.

SN	Tag Name	Tag Type	Tag Group	Tag Unit	Logging Time	Arduino Device	Arduino Shiled	Status
1	T-102	Temperature	factory A	degree celsius	30 minutes	Arduino1	Ethernet shield 1	stopped START
2	P-103	Pressure	factory A	pascal	30 minutes	Arduino 2	Ethernet shield 2	running STOP
3	PID-101	Control Signal	factory A	%	30 minutes	Arduino 3	Ethernet shield3	running STOP

Figure 19-6: Configuration of logging time of tags

And next, that need to be configured is alarm limit. For that, we should click on “Data Management” option in home page and go to “Alarm Configuration” for the configuration where all the tags are listed in table with a button “Configure” in each row. For configuration, we should click on the “Configure” button which redirect to the page as shown in Figure 19-7.



Configure the Alarm Limits of the Tags

View all the Tags and their alarm limits.

Select the tag for configuration of Alarm. T-102

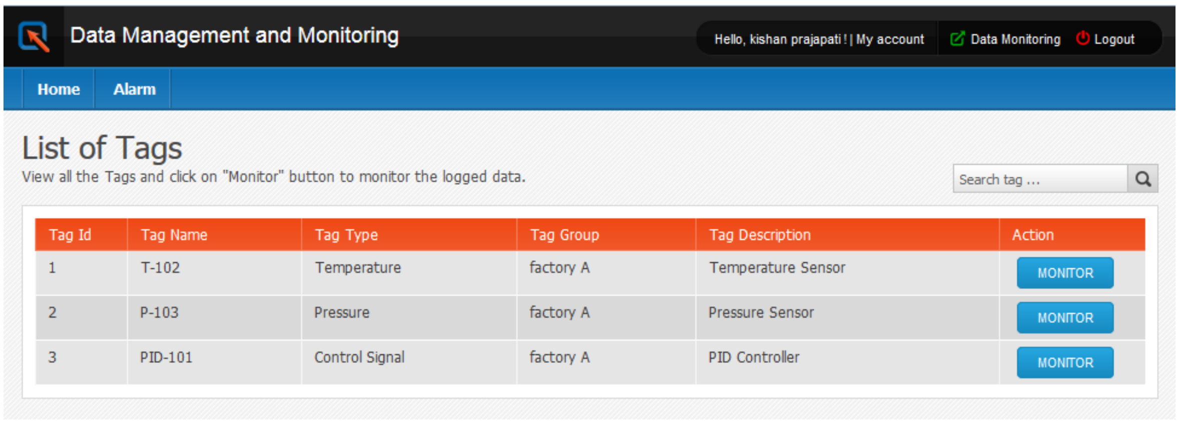
Tag Name	T-102
Upper Limit	30
Lower Limit	20
Alarm Priority	HIGH
Alarm Status	Active

Figure 19-7: Alarm Configuration of a tag

In this alarm configuration page, submit button should be clicked after entering the entire required field. This is completion of the alarm configuration of the particular tag.

19.3 Data Monitoring

All the active tags are listed in the table as the button; “Data Monitoring” is clicked in the home page. The table with all the tags will be displayed along with a button in each row and named as “Monitor” which is shown in Figure 19-8.


Data Management and Monitoring

Hello, kishan prajapati ! | My account | Data Monitoring | Logout

Home | Alarm

List of Tags

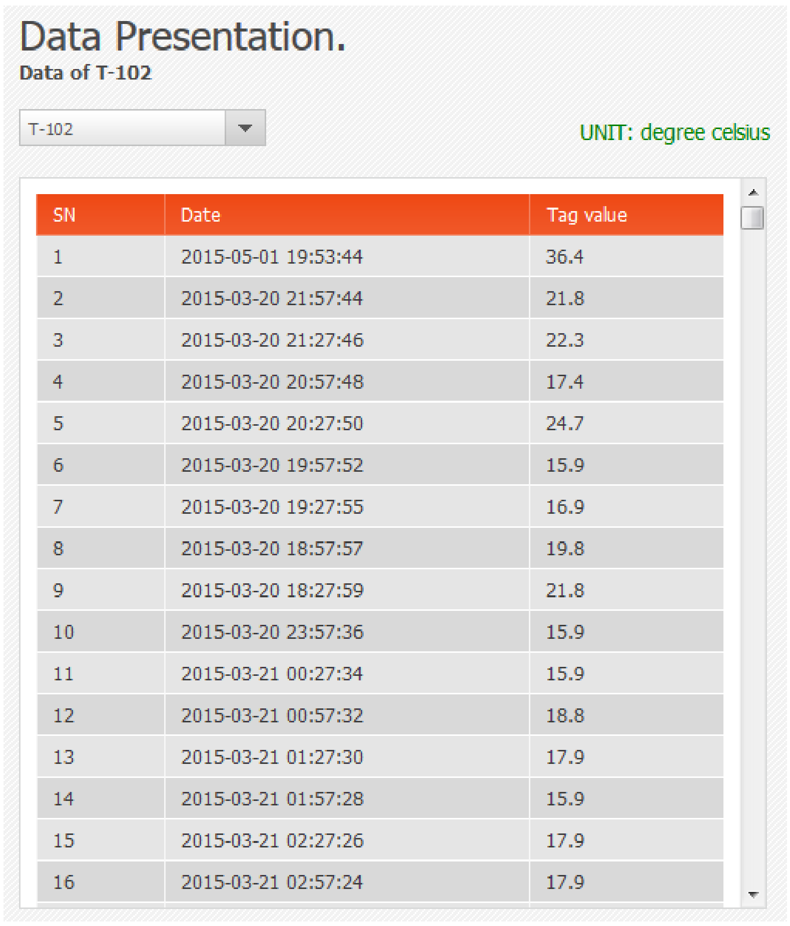
View all the Tags and click on "Monitor" button to monitor the logged data.

Search tag ...

Tag Id	Tag Name	Tag Type	Tag Group	Tag Description	Action
1	T-102	Temperature	factory A	Temperature Sensor	MONITOR
2	P-103	Pressure	factory A	Pressure Sensor	MONITOR
3	PID-101	Control Signal	factory A	PID Controller	MONITOR

Figure 19-8: List of Active Tags ready to monitor

The button, "Monitor" is needed to click to monitor the respective tags. As the button is clicked, it is directed to a monitoring page and the monitoring page is shown in Figure 19-9. The data of a tag is displayed in table. Besides, data is presented in graph as well per day. There is facility of changing the date to view the data of the corresponding day as a trend.


Data Presentation.

Data of T-102

T-102

UNIT: degree celsius

SN	Date	Tag value
1	2015-05-01 19:53:44	36.4
2	2015-03-20 21:57:44	21.8
3	2015-03-20 21:27:46	22.3
4	2015-03-20 20:57:48	17.4
5	2015-03-20 20:27:50	24.7
6	2015-03-20 19:57:52	15.9
7	2015-03-20 19:27:55	16.9
8	2015-03-20 18:57:57	19.8
9	2015-03-20 18:27:59	21.8
10	2015-03-20 23:57:36	15.9
11	2015-03-21 00:27:34	15.9
12	2015-03-21 00:57:32	18.8
13	2015-03-21 01:27:30	17.9
14	2015-03-21 01:57:28	15.9
15	2015-03-21 02:27:26	17.9
16	2015-03-21 02:57:24	17.9

Figure 19-9: Data presentation in table

Figure 19-9 shows the data presentation of the selected tags with the tag value and timestamp along with its unit. It is possible to change the tags in select button to monitor its data. The table displays all the logged value in the database.

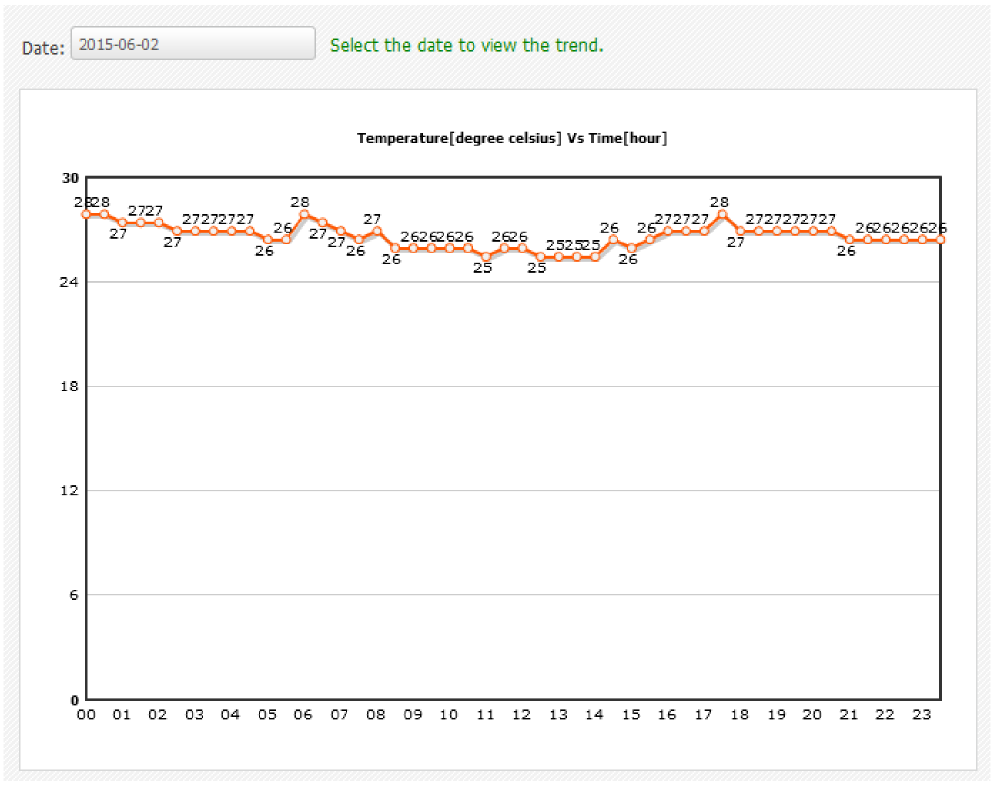
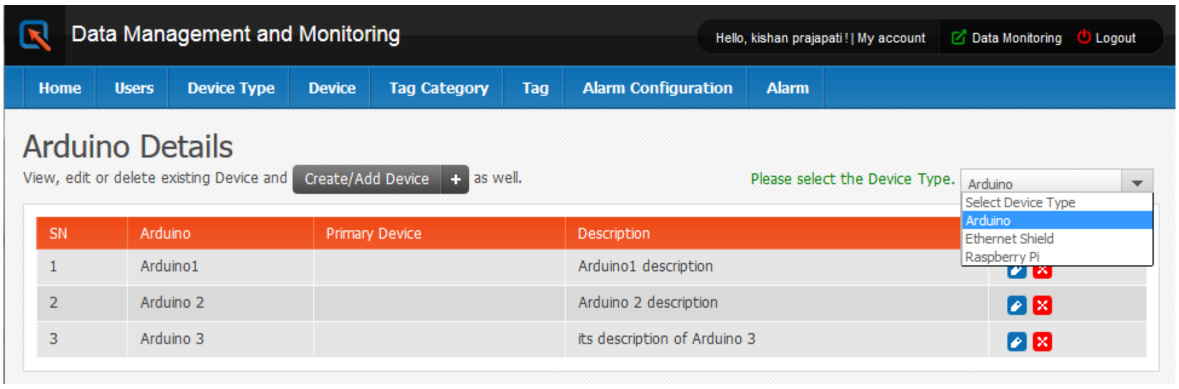


Figure 19-10: Data presentation as a trend

Figure 19-10 shows the trend of the data of the selected tags of the current date. Tag value is displayed in the graph and the time interval for the value is 30 minutes. Date can be change by click on date to see the trend of the data in the particular day.

19.4 Data Management

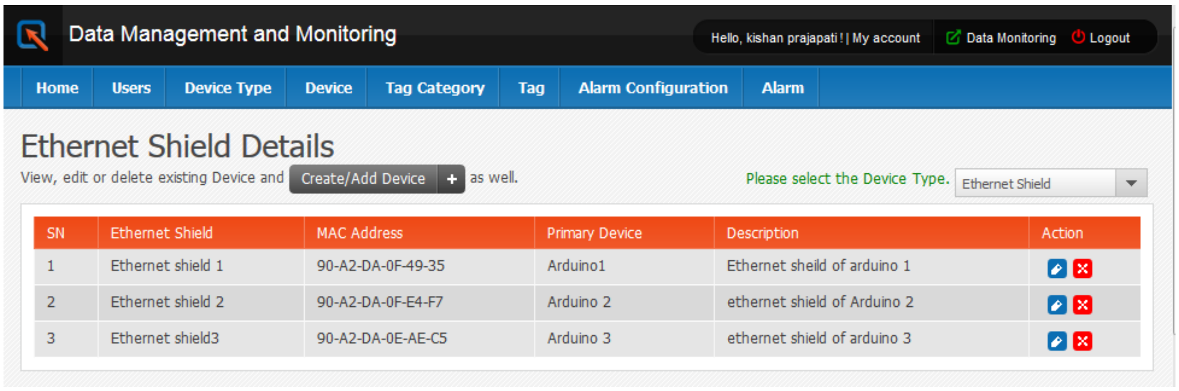
Data management includes the management of tag and other devices like Arduino, Ethernet Shield whatever needed in the system. For data management, the icon named as “Data Management” is needed to click in home page of the website. As the icon is clicked then the menu for managing device will appear in the main menu bar. Management means the adding, editing and deleting the records of the devices.



SN	Arduino	Primary Device	Description
1	Arduino1		Arduino1 description
2	Arduino 2		Arduino 2 description
3	Arduino 3		its description of Arduino 3

Figure 19-11: Management of a Device, Arduino

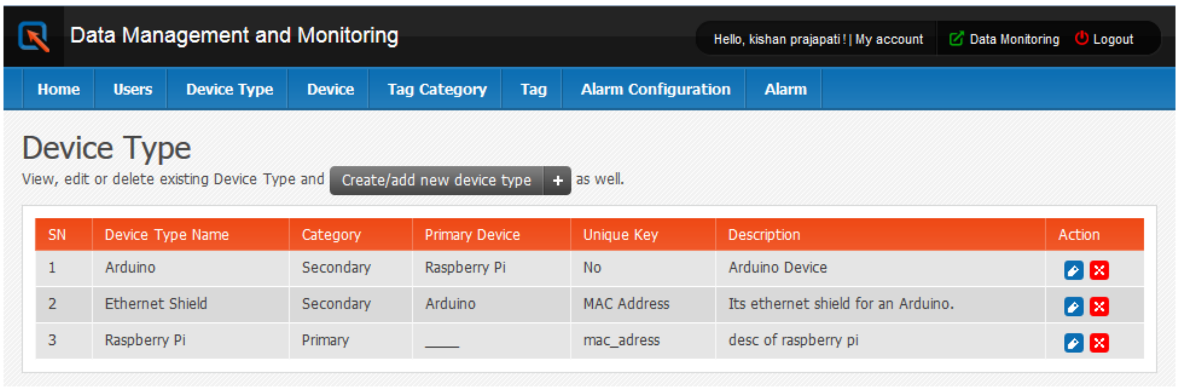
Figure 19-11 shows the table for the management of device, Arduino. There is a button to add a new Arduino and two icons in each row to edit and delete the respective row. Similarly, management of Ethernet Shield for Arduino is shown in Figure 19-12. The field, "Primary Device" is another device linked with it. For example, Ethernet shield 1, Ethernet shield 2 and Ethernet shield 3 are linked to Arduino 1, Arduino 2 and Arduino 3 respectively and if a device doesn't have a primary device, it is left blank.



SN	Ethernet Shield	MAC Address	Primary Device	Description	Action
1	Ethernet shield 1	90-A2-DA-0F-49-35	Arduino1	Ethernet sheild of arduino 1	
2	Ethernet shield 2	90-A2-DA-0F-E4-F7	Arduino 2	ethernet shield of Arduino 2	
3	Ethernet shield3	90-A2-DA-0E-AE-C5	Arduino 3	ethernet shield of arduino 3	

Figure 19-12: Management of Ethernet shield for Arduino

There is select button in the right top of the page where device type can be selected to view the list of the device of the respective type. For managing the device type, we just need to click on "Device Type" menu in the menu bar. The page for managing the device type is shown in Figure 19-13.


Data Management and Monitoring
Hello, kishan prajapati ! | My account
Data Monitoring
Logout
Home
Users
Device Type
Device
Tag Category
Tag
Alarm Configuration
Alarm
Create/add new device type

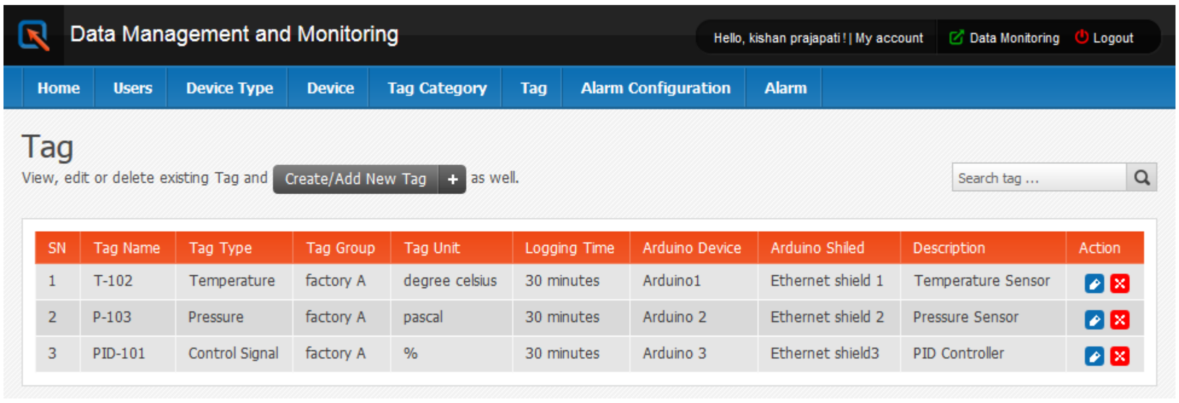
Device Type

View, edit or delete existing Device Type and Create/add new device type as well.

SN	Device Type Name	Category	Primary Device	Unique Key	Description	Action
1	Arduino	Secondary	Raspberry Pi	No	Arduino Device	
2	Ethernet Shield	Secondary	Arduino	MAC Address	Its ethernet shield for an Arduino.	
3	Raspberry Pi	Primary	____	mac_adress	desc of raspberry pi	

Figure 19-13: Managing Device Type

Similarly, the tags needed for the system are managed. The table for managing tags is shown in Figure 19-14. There are two different category of tags named as “Tag Type” and “Tag Group”. These two categories are managed in similar ways and they can be accessed by two sub menu under “Tag Category”. These all tags and devices are inter-related in this system.



Tag

View, edit or delete existing Tag and Create/Add New Tag as well.

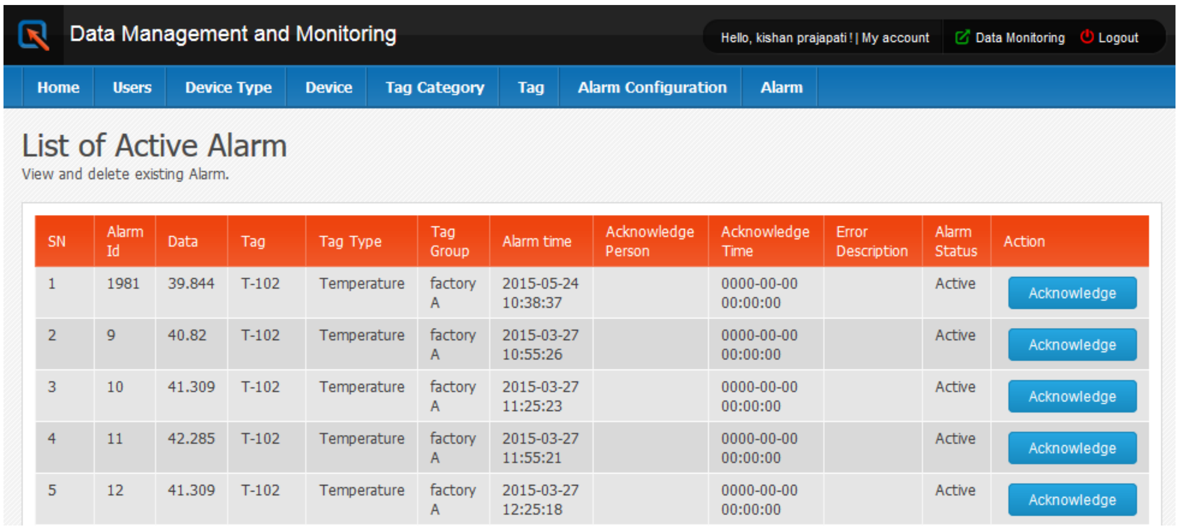
SN	Tag Name	Tag Type	Tag Group	Tag Unit	Logging Time	Arduino Device	Arduino Shiled	Description	Action
1	T-102	Temperature	factory A	degree celsius	30 minutes	Arduino1	Ethernet shield 1	Temperature Sensor	
2	P-103	Pressure	factory A	pascal	30 minutes	Arduino 2	Ethernet shield 2	Pressure Sensor	
3	PID-101	Control Signal	factory A	%	30 minutes	Arduino 3	Ethernet shield3	PID Controller	

Figure 19-14: Management of Tags

All the information related to a tag is displayed. Tag type, group, unit, logging time, Arduino device, Ethernet shield and the description of each tag is displayed as a row in the table.

19.5 Alarm Management

Each and every tag has its own upper and lower limit. During data logging, if these limits don’t meet, alarm is activated and is listed in the website as an active alarm. Website has an option to inform about the alarm to an operator so that he/she can acknowledge the alarm. Alarm of the system can be viewed under the menu “Alarm”. There are two types of alarm, active alarm and acknowledged alarm. If the information about the active alarm is provide to an operator, the alarm turns to acknowledged one.



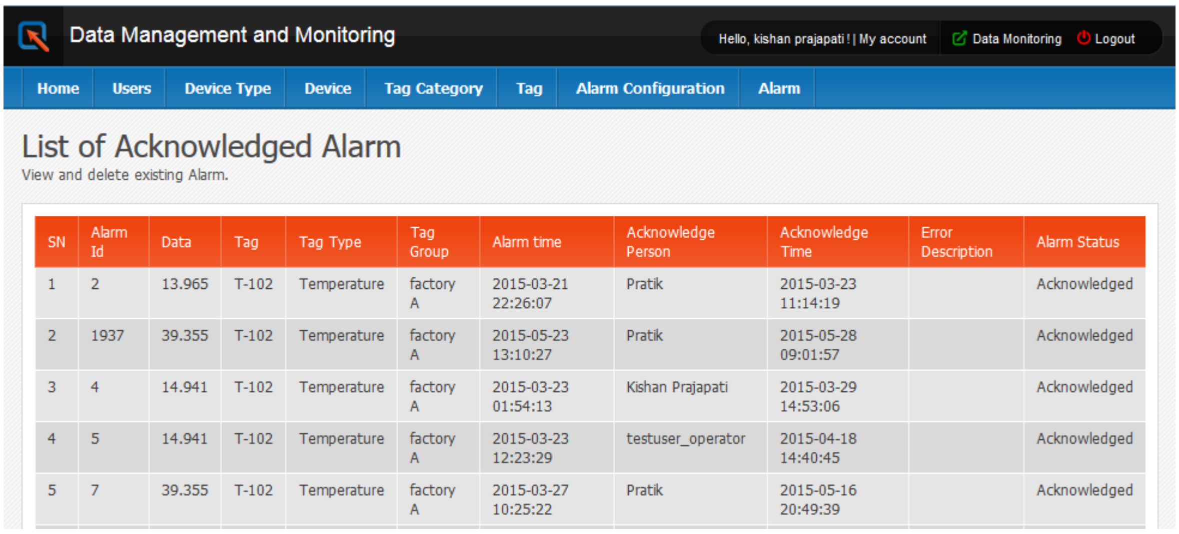
List of Active Alarm

View and delete existing Alarm.

SN	Alarm Id	Data	Tag	Tag Type	Tag Group	Alarm time	Acknowledge Person	Acknowledge Time	Error Description	Alarm Status	Action
1	1981	39.844	T-102	Temperature	factory A	2015-05-24 10:38:37		0000-00-00 00:00:00		Active	Acknowledge
2	9	40.82	T-102	Temperature	factory A	2015-03-27 10:55:26		0000-00-00 00:00:00		Active	Acknowledge
3	10	41.309	T-102	Temperature	factory A	2015-03-27 11:25:23		0000-00-00 00:00:00		Active	Acknowledge
4	11	42.285	T-102	Temperature	factory A	2015-03-27 11:55:21		0000-00-00 00:00:00		Active	Acknowledge
5	12	41.309	T-102	Temperature	factory A	2015-03-27 12:25:18		0000-00-00 00:00:00		Active	Acknowledge

Figure 19-15: List of Active Alarm

The table displaying the active alarm in the website is shown in Figure 19-15. Each row consists of a button “Acknowledge” which provides information to an operator about the respective alarm via email. The list of acknowledged alarm is shown in Figure 19-16.



List of Acknowledged Alarm

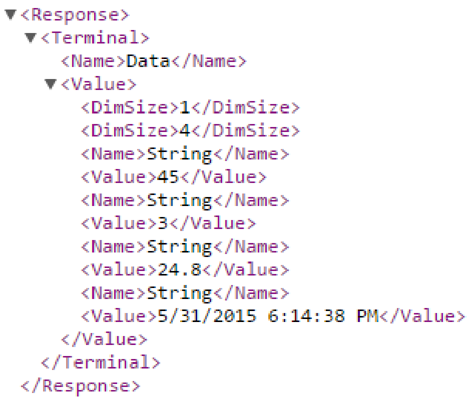
View and delete existing Alarm.

SN	Alarm Id	Data	Tag	Tag Type	Tag Group	Alarm time	Acknowledge Person	Acknowledge Time	Error Description	Alarm Status
1	2	13.965	T-102	Temperature	factory A	2015-03-21 22:26:07	Pratik	2015-03-23 11:14:19		Acknowledged
2	1937	39.355	T-102	Temperature	factory A	2015-05-23 13:10:27	Pratik	2015-05-28 09:01:57		Acknowledged
3	4	14.941	T-102	Temperature	factory A	2015-03-23 01:54:13	Kishan Prajapati	2015-03-29 14:53:06		Acknowledged
4	5	14.941	T-102	Temperature	factory A	2015-03-23 12:23:29	testuser_operator	2015-04-18 14:40:45		Acknowledged
5	7	39.355	T-102	Temperature	factory A	2015-03-27 10:25:22	Pratik	2015-05-16 20:49:39		Acknowledged

Figure 19-16: List of Acknowledge Alarm

20 Web Service and Data Dashboard

Web service was created to publish the latest record of the database (SQL Server). SQL toolkit in LABVIEW is used to extract the value from the database where as web service created in LABVIEW is used for publishing the data. The latest record includes the DataId, TagId, TagValue and Timestamp. The XML output of the web service is shown in Figure 20-1.



```
▼<Response>
  ▼<Terminal>
     <Name>Data</Name>
    ▼<Value>
       <DimSize>1</DimSize>
       <DimSize>4</DimSize>
       <Name>String</Name>
       <Value>45</Value>
       <Name>String</Name>
       <Value>3</Value>
       <Name>String</Name>
       <Value>24.8</Value>
       <Name>String</Name>
       <Value>5/31/2015 6:14:38 PM</Value>
     </Value>
   </Terminal>
 </Response>
```

Figure 20-1: XML Response of Web Service

Data dashboard can be made connection with the web service and access those public data. In this case, the data extracted from the database is in the form of array. But, the data dashboard doesn’t support this type of data type. Hence, it is changed to scalar and only temperature is displayed. The data dashboard displaying the present value of sensor/process is shown in Figure 20-2.

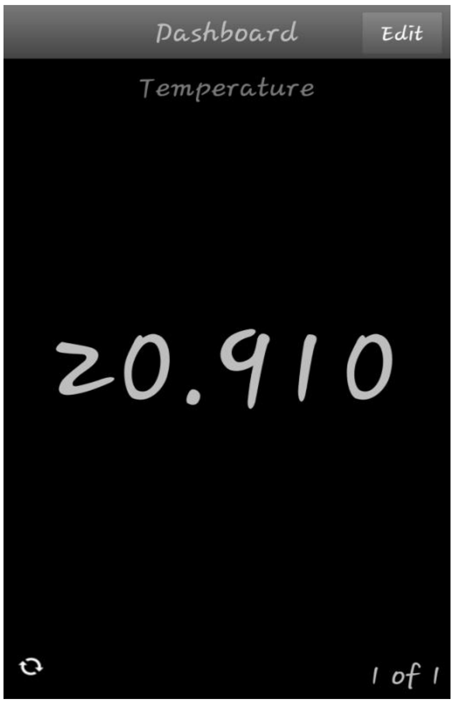


Figure 20-2: Data Dashboard accessing data of Web Service

Part V: Summary

It gives the summary of the thesis. It includes the discussion, conclusion and suggestion for further work.

21 Discussion and Suggestion

This includes the discussion and suggestions on different tasks of the thesis.

21.1 Logging System

In the first logging system, Ethernet shield for Arduino is used which is more efficiency on connecting to internet. However, Ethernet cable is needed for connecting to the network for Ethernet shield. This can make the system bit messy. So, instead of Ethernet shield, it will be better to use Wi-Fi shield.

21.2 Website

The website developed is mostly fit and seems good when used in laptop (normal size screen). The website may show some errors in design when viewed in large or small screen. As a suggestion, HTML5 and CSS3 can be used for website design as they are the latest and the most advanced one. They introduce a bunch of new elements that will make our pages more semantic and make it a lot easier for search engines (Kjaer, 2009). Besides, they allow adding more interesting animations and seem more advance. Besides design, there is some limitation in monitoring part as well in website. It allows viewing the value of sensor/process for duration of a day only. Instead, it will be better if the trend is more flexible. Besides, multiple graphs can replace the single graph so that we can monitor the multiple tags and provides the feature of comparison. And, website can be developed by some another strong programming language as ASP.net than PHP.

21.3 Data Dashboard for LABVIEW

Data dashboard is an application which just runs in apple IOS and android mobile. It will be better if an application can be built which runs in windows as well and have more features than data dashboard. In this thesis, just a recent sensor value (temperature) is displayed. A suggestion for future work is to work hard on it to increase the data information. More trends, figures, etc can be added so that monitoring will be clearer, informative and advance as well.

21.4 Central Server

It is the most important suggestion for the students. Central server should be installed in separate device so that all the information of different units (industries) can be stores collectively as backup and central management. This central server is the best idea to link the different units into one. So, it is important to build a system that copies all the data from the local database to main central server in real time.

21.5 Other Suggestions for further work

1) Improve the website for data management and monitoring. Add more features in case of monitoring part.
2) Add more features in data dashboard.
3) Implement other Control strategies, such as MPC (Model Predictive Control), Feed forward Control, Cascade Control, etc. And State Estimation methods like Kalman Filter.
4) Use the latest Raspberry Pi which supports the latest operating system, windows 10.
5) Use Arduino Due which solves the memory problem in Arduino UNO.

22 Conclusion

In this thesis, I have developed a logging, monitoring and controlling application using Arduino and Raspberry Pi. This application is used for industrial purposes such as process control and monitoring. This technical report is the documentation of the thesis, “Process Control and Monitoring using Arduino and Raspberry Pi”.

The very initial phase of the thesis is to design the relational database model using CA Erwin and startup with the Raspberry Pi. As, I started the thesis using new Raspberry Pi, installation of operating system is the first startup with Raspberry Pi. All the software such as MySQL Server, phpmyadmin, apache, Arduino IDE, etc are installed. Hence, the local database server for a system is built in Raspberry Pi itself. Arduino is the next important element of the application. It is the one interfacing with outer world (sensors and processes). MySQL Connector is Arduino library which is used for the connection between Arduino and MySQL server. This made the logging part successful in this thesis. For monitoring purpose, website is developed in Raspberry Pi using PHP. Besides monitoring, it is used for data management as well.

The next important task was web based logging and monitoring using data dashboard with LABVIEW. In this part, Arduino and LABVIEW are interfaced by the use of LABVIEW Interface for Arduino toolkit which is installed using VI Package Manager. SQL toolkit is used in LABVIEW for connecting the LABVIEW and database for logging purpose. Finally, web service is created in LABVIEW which is published in web. Data dashboard with LABVIEW is used to access the published web service.

PID controller is also implemented in this thesis for controlling process. PID is developed in Arduino and it is used as a library in the Arduino sketch.

The entire tasks are performed along with documentation is performed in around 18 months. And hence, this is the completion of the thesis.

References

arduino.cc. (2015a). Arduino WiFi Shield. Retrieved April 10, 2015, from http://arduino.cc/en/Main/ArduinoWiFiShield

arduino.cc. (2015b). What is Arduino? Retrieved April, 2015, from http://arduino.cc/en/guide/introduction

Bates, D. (2014). *Raspberry Pi Projects for Kids* (pp. 96).

Bell, C. (2012). MySQL Connector/Arduino. Retrieved 03/05, 2015, from https://launchpad.net/mysql-arduino

Bell, C. (2013). *Beginning Sensor Networks with Arduino and Raspberry Pi*. http://proquest.safaribooksonline.com.ezproxy.hit.no/book/hardware/arduino/9781430258247

Bell, C. (2013). Introducing MySQL Connector/Arduino. Retrieved 05/09, 2015, from http://drcharlesbell.blogspot.no/2013/04/introducing-mysql-connectorarduino_6.html

Comnes, B., & Rosa, A. L. (2015). Arduino PID Example Lab

Donat, W. (2014). *Learn Raspberry Pi Programming with Python* (pp. 256). Retrieved from http://proquest.safaribooksonline.com.ezproxy.hit.no/book/programming/python/9781430264248

Halvorsen, H.-P. (2011). Database Communication in LabVIEW. Retrieved from http://home.hit.no/~hansha/documents/labview/training/Database%20Communication%20in%20LabVIEW/Database%20Communication%20in%20LabVIEW.pdf

Haugen, F., Fjelddalen, E., Dunia, R., & Edgar, T. F. (2007). Demonstrating PID Control Principles using an Air Heater and LabVIEW.

Heng, C. (2014). How to Design a Website with Dreamweaver CS6. Retrieved 05/16, 2015, from http://www.thesitewizard.com/dreamweaver/dreamweaver-cs6-tutorial-1.shtml

Instruments, N. (2013). Tutorial: Creating and Accessing a LabVIEW Web Service (Real-Time, Windows). Retrieved 05/30, 2015, from http://zone.ni.com/reference/en-XX/help/371361K-01/lvhowto/build_web_service/

Kjaer, M. (2009). HTML 5 and CSS 3: The Techniques You'll Soon Be Using. Retrieved 06/02, 2015, from HTML 5 and CSS 3: The Techniques You'll Soon Be Using

Microsoft. (2015). ODBC Overview. Retrieved 05/31, 2015, from https://msdn.microsoft.com/en-us/library/ms710220(v=vs.85).aspx

mysql. (2015). Overview of the MySQL Database Management System. Retrieved April 11, 2015, from https://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html

mysqltutorial. (2015). Introduction to SQL Trigger. Retrieved 05/16, 2015, from http://www.mysqltutorial.org/sql-triggers.aspx

phpmyadmin.net. (2014). Welcome to phpMyAdmin's documentation! Retrieved April 11, 2015, from http://docs.phpmyadmin.net/en/latest/

raspberrypi.org. INSTALLING OPERATING SYSTEM IMAGES USING WINDOWS. Retrieved March 20, 2015, from https://www.raspberrypi.org/documentation/installation/installing-images/windows.md raspberrypi.org. NOOBS SETUP.   Retrieved March 20, 2015, from https://www.raspberrypi.org/help/noobs-setup/

raspberrypi.org. (2015a). Raspberry Pi 2 - Model B Retrieved April 10, 2015, from https://www.raspberrypi.org/products/raspberry-pi-2-model-b/

raspberrypi.org. (2015b). SETTING UP AN APACHE WEB SERVER ON A RASPBERRY PI.   Retrieved 4/15, 2015, from https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md

raspipress. (2015a). Tutorial – Install Apache, PHP and MySQL on Raspberry Pi. Retrieved 04/15, 2015, from http://www.raspipress.com/2012/09/tutorial-install-apache-php-and-mysql-on-raspberry-pi/

RaspiPress. (2015b). Tutorial – Install PhpMyAdmin on your Raspberry Pi.   Retrieved 04/15, 2015, from http://www.raspipress.com/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/

Sammy_K. (2011). LabVIEW Interface for Arduino Setup Procedure.   Retrieved 05/29, 2015, from https://decibel.ni.com/content/docs/DOC-15971

Sanders, B. (2013). Starting OOP in PHP: Raspberry Pi Users Welcome! Retrieved from http://www.php5dp.com/starting-oop-in-php-raspberry-pi-users-welcome/

stackexchange. (2015). apt-get installation doesn't work on raspberry pi.   Retrieved 04/15, 2015, from http://raspberrypi.stackexchange.com/questions/9307/apt-get-installation-doesnt-work-on-raspberry-pi

stackoverflow. (2013). ERROR 2003 (HY000): Can't connect to MySQL server (111). Retrieved 05/10, 2015, from http://stackoverflow.com/questions/11758339/error-2003-hy000-cant-connect-to-mysql-server-111

ucl.ac.uk. (2000). An introduction to databases.   Retrieved April 11, 2015, from http://www.ucl.ac.uk/archaeology/cisp/database/manual/node1.html#extable

Vujovic´, V., & Maksimovic, M. (2015). Raspberry Pi as a Sensor Web node for home automation. *Comput Electr Eng.*

wikipedia.org. (2014, 06/05/2014). CA ERwin Data Modeler.   Retrieved May, 2014, from http://en.wikipedia.org/wiki/CA_ERwin_Data_Modeler

wikipedia.org. (2015, 01/02/2015). MySQL.   Retrieved Feb, 2015, from http://en.wikipedia.org/wiki/MySQL

Part VI: Appendix

There are lists of appendix. They are listed as follow.

1) Appendix A: Thesis description
2) Appendix B: Installation of operating system in Raspberry Pi
3) Appendix C: Arduino logging Code
4) Appendix D: Added code in MySQL/Connector Library
5) Appendix E: Arduino PID library
6) Appendix F: Website code

Appendix A

Telemark University College

Faculty of Technology

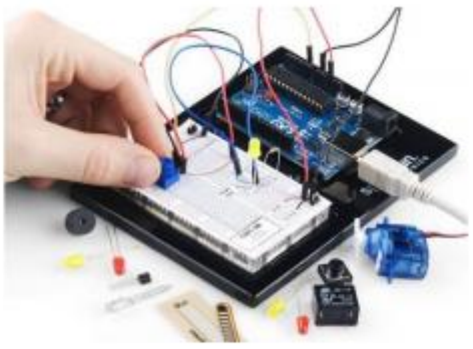
FMH606 Master's Thesis

Title: Process Control and Monitoring using Arduino and Raspberry Pi

TUC supervisor: Hans-Petter Halvorsen

External partner: National Instruments

Task Description: The Arduino and Raspberry Pi platforms have become very popular today among hobbyists and in education. In this project we will explore how we can use the Arduino and Raspberry Pi platforms for industrial applications such as Process Control and Monitoring Applications.



The following topics could be investigated in this project:

- Explore the Arduino and the Raspberry Pi platform in general and explore how they can be integrated in Process Control and Monitoring Applications.
- Create an Arduino/Rapberry Pi embedded PID Controller (replacing e.g., the Fuji PXG5 PID controllers or myRIO devices that we have at TUC) and do practical experiments on small-scale process models. The code should be implemented as an Arduino Library.
- Implement other Control strategies, such as MPC (Model Predictive Control), Feedforward Control, Cascade Control, etc. And State Estimation methods like Kalman Filter.
- Using Arduino and Raspberry Pi within MATLAB and Simulink
- Using Arduino/Raspberry Pi for Process Monitoring and Datalogging using Web Services and “Data Dashboard for LabVIEW” (App for Smartphones and Tablets)

- A web-based logging service similar to e.g., Xively or Temboo should be developed. A Web Service should be created together with an Arduino Library for communication with the library. • Using Arduino within LabVIEW; LabVIEW LINX
- Explore Wireless Communication, such as WiFi, Bluetooth, XBee, RFID together with Raspberry Pi and Arduino for Publishing and Monitoring Process Data
- Explore possibilites for using Raspberry Pi together with the LabVIEW platform
- Explore the possibilities to combine Raspberry Pi and Arduino, i.e. control the arduino from the Raspberry Pi device together with e.g., Python, using Raspberry Pi as a Web Server, etc. Based on the students interest, he should select some of the topics above in collaboration with the supervisor for further investigation.

References:

- The Arduino platform: http://arduino.cc
- The Raspberry Pi platform: http://www.raspberrypi.org
- LabVIEW LINX (using Arduino with the LabVIEW platform): https://www.labviewhacker.com/doku.php?id=learn:libraries:linx:getting_started
- Arduino Support from MATLAB: http://www.mathworks.se/hardware-support/arduino-matlab.html • Arduino Support from Simulink: http://www.mathworks.se/hardware-support/arduino-simulink.html
- Raspberry Pi Support from MATLAB: http://www.mathworks.se/hardware-support/raspberry-pi-matlab.html
- Raspberry Pi Support from Simulink: http://www.mathworks.se/hardware-support/raspberry-pi-simulink.html
- Learning Python with Raspberry Pi: http://www.raspberrypi.org/learning-python-with-raspberry-pi/
- Xively: https://xively.com

Task background:

Arduino is a low cost open-source electronics prototyping platform. The Arduino has analog and digital I/O. The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and may be used with a standard keyboard and mouse. It does not include a built-in hard disk or solid-state drive, but it uses a SD card for booting and persistent storage.

Student category: SCE students

Practical arrangements: None

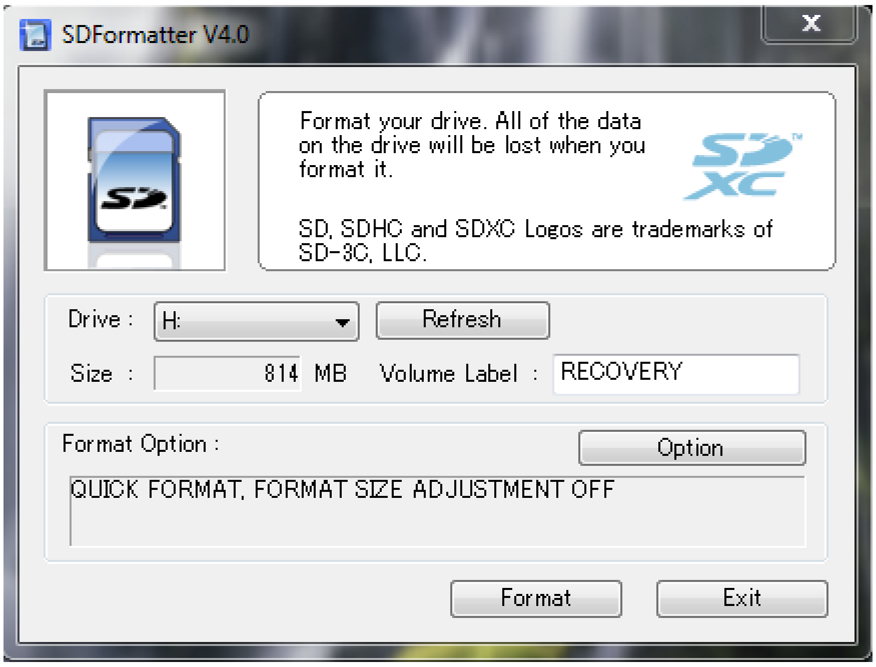
Signatures: Supervisor (date and signature):

Students (date and signature):

Appendix B

To set up a blank SD card with NOOBS:

1) Buy a new SD card (16 GB recommended).
2) Format the SD card using SD formatter as FAT.
   - ➢ You can download the SD formatter as per the operating system on a laptop or computer. You can refer to the link provided. Download SD Formatter
   - ➢ Install it on a laptop.
   - ➢ Insert the SD card in the laptop SD card reader.
   - ➢ Run the SD formatter software and choose the drive. The SD Formatter seems as shown in Figure A-1.



Format your drive. All of the data on the drive will be lost when you format it.

SD, SDHC and SDXC Logos are trademarks of SD-3C, LLC.

Drive : H:

Size : 814 MB Volume Label : RECOVERY

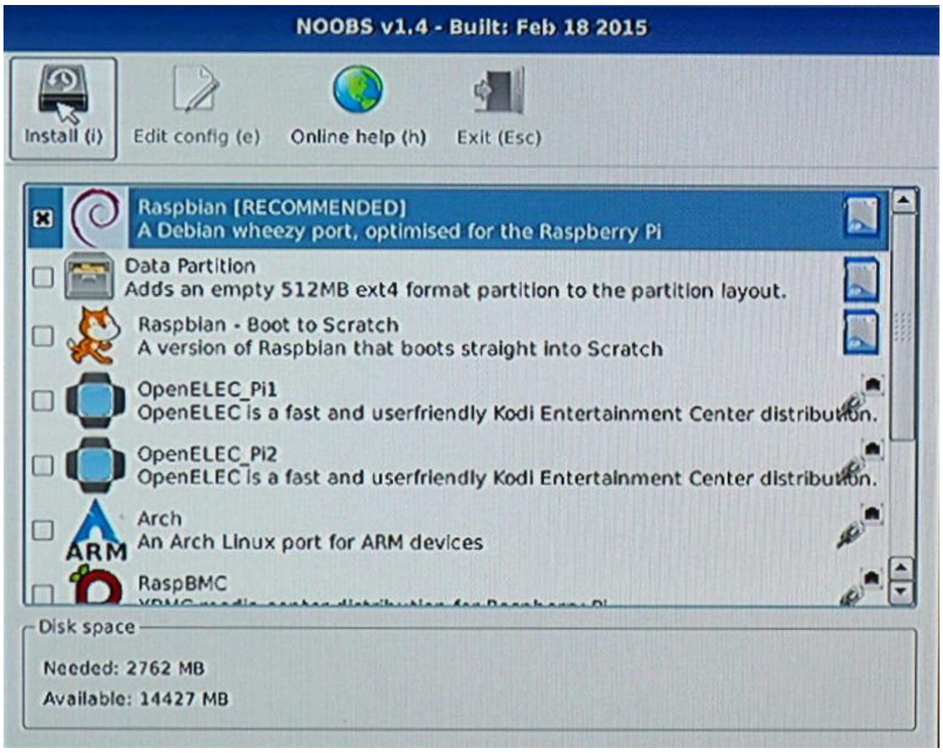
Format Option :

QUICK FORMAT, FORMAT SIZE ADJUSTMENT OFF

Figure A-1: SD Formatter

   - ➢ Click on the option and make “FORMAT SIZE ADJUSTMENT" option to "ON" to ensure that the entire SD card volume is formatted, and not just a single partition.
   - ➢ Finally click on”Format” button to start.
3) Download the latest version of NOOBS (zip file) in a laptop; you can download it in the link Download NOOBS.
4) Extract the file and copy the file in the formatted SD card. The file size may be around 800 MB.
5) Connect keyboard, mouse, monitor to the raspberry Pi and insert the memory stick on it and finally plug-in the power to it. Then, the installation of operating system is started.

6) In the first, there is list of operating system listed as shown Figure A-2. You should choose one of them. It is recommended to choose “Raspbian” and click on the “install” icon as shown in screen shot. It will take around 30 minutes to complete.



NOOBS v1.4 - Built: Feb 18 2015

[✓] Raspbian [RECOMMENDED]
A Debian wheezy port, optimised for the Raspberry Pi

Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.

Raspbian - Boot to Scratch
A version of Raspbian that boots straight into Scratch

OpenELEC_Pi1
OpenELEC is a fast and userfriendly Kodi Entertainment Center distribution.

OpenELEC_Pi2
OpenELEC is a fast and userfriendly Kodi Entertainment Center distribution.

Arch
An Arch Linux port for ARM devices

RaspBMC

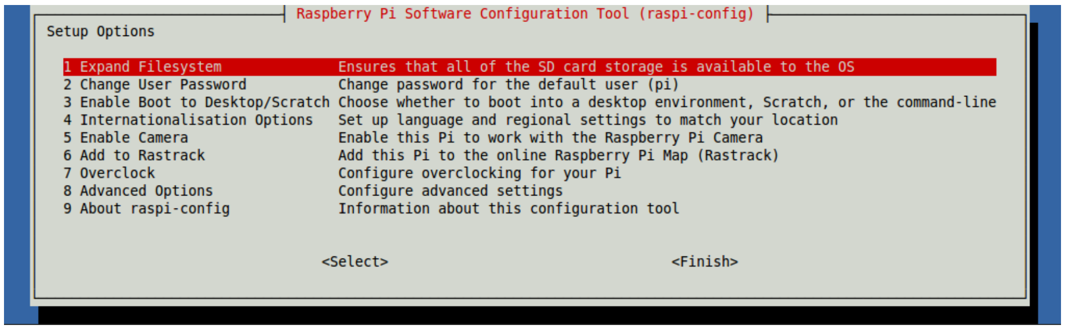
Disk space
Needed: 2762 MB
Available: 14427 MB

Figure A-2: List of operating system

7) During installation, numbers of figures are displayed with some tips. It may be helpful later. Hence, recommended to go through the information provided.

8) After completion, Raspberry Pi software configuration tool is opened where you can configure most things to use on the way we want them. There are some useful configurations needed to be done. Click on “Advance Option” and enable SSH. Click on “Enable Boot to Desktop/Scratch” to enable “Boot to Desktop/Scratch” and select “Desktop Log in” option. For time zone and language setup, click on the “Internationalisation option. You can return to this menu any time by typing sudo raspi-config as a command line. For details on all the Setup options, you can go through to the link raspi-config.

The configuration tool setup option is shown in Figure A-3.



```
Raspberry Pi Software Configuration Tool (raspi-config)

Setup Options

  1 Expand Filesystem              Ensures that all of the SD card storage is available to the OS
  2 Change User Password           Change password for the default user (pi)
  3 Enable Boot to Desktop/Scratch Choose whether to boot into a desktop environment, Scratch, or the command-line
  4 Internationalisation Options   Set up language and regional settings to match your location
  5 Enable Camera                  Enable this Pi to work with the Raspberry Pi Camera
  6 Add to Rastrack                Add this Pi to the online Raspberry Pi Map (Rastrack)
  7 Overclock                      Configure overclocking for your Pi
  8 Advanced Options               Configure advanced settings
  9 About raspi-config             Information about this configuration tool

                        <Select>                    <Finish>
```

Figure A-3: Configuration tool (raspi-config)

9) After completion of configuration, click on the "Finish" button.
10) For completion of installation, you will get an alert as "OS (es) installed successfully" and the graphical desktop is seen as shown in Figure A-4.
11) If you don't choose the option as "Desktop log in" during configuration, you are asked to enter text. Then enter "startx" at the command prompt to start the graphical desktop (raspberrypi.org).

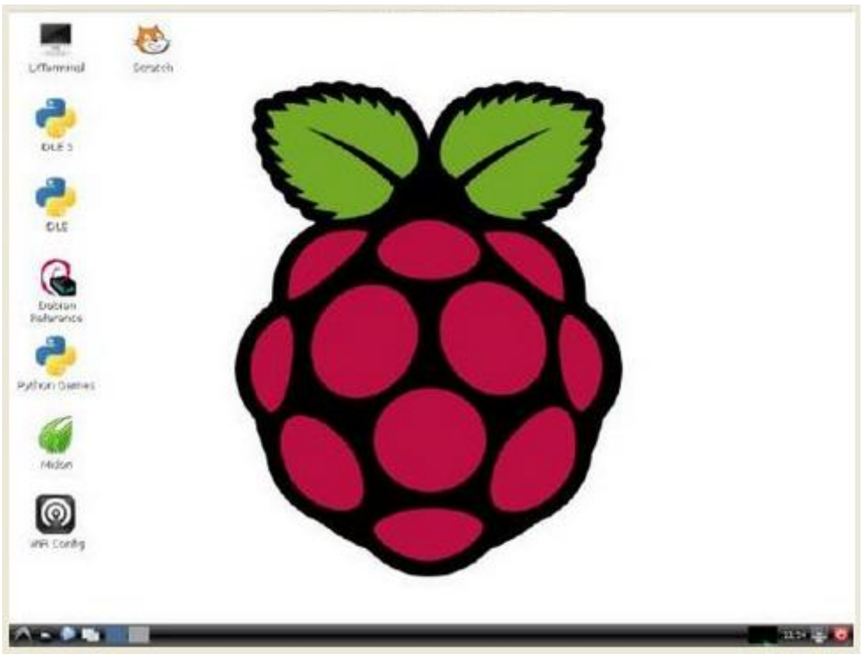


Figure A-4: Graphical Desktop of Raspberry Pi

To install operating system images without NOOBS

Advanced users wish to install a specific operating system image. In this case, you don't need to install the NOOBS in the SD card. In this case, you should follow the following steps.

1) Download an operating image as per your interest in a laptop. In my project, I used the RASPBIAN. You can download from the link below. [Download Operating Image]

2) You need to write this image file in a formatted SD card. It is already explained for formatting the SD card as FAT above.

3) For writing image file, following steps should be followed.

   - Download Win32DiskImager utility from Download Win32DiskImager
   - Extract the file in the laptop and run as administrator. The running program looks as shown in Figure A-5.

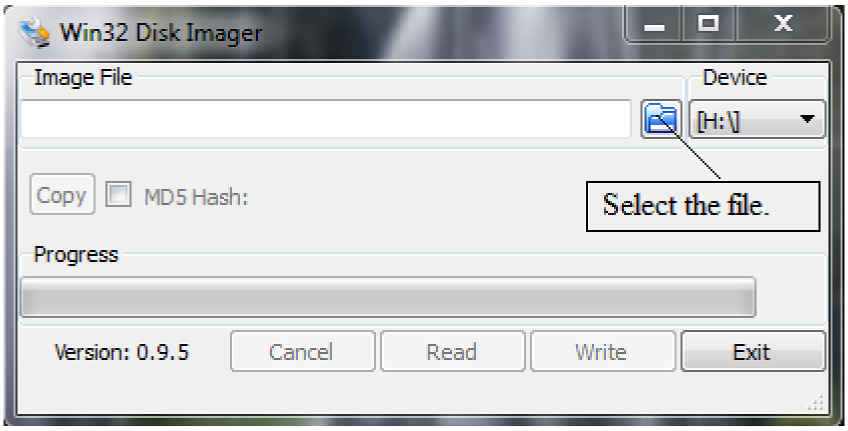


Figure A-5: Win 32 Disk Imager

   - Select the image file to be copied to the SD card.
   - Select the drive letter of the SD card, as H:\ as shown in Figure A-5. You should choose the correct one because it may damage the drive if you choose a wrong one.
   - Click on "Write" button to start writing which will be highlighted after the selection of file and drive (raspberrypi.org).

Exit the imager after finish and eject the SD card from the card reader and follow the steps from 5 to the end as in "**To set up a blank SD card with NOOBS**".

Appendix C

```
#include <sha1.h>

#include <SPI.h>

#include <Ethernet.h>

#include <mysql.h>

// MAC Address of Ethernet shield

byte mac_addr[] = {0x90,0xA2, 0xDA, 0x0E, 0xAE, 0xC5};

// IP address of Raspberry pi (MySQL Server)

IPAddress server_addr (128,39,35,237);

// IP Address of Ethernet shield

byte ethernet_ip[] = { 128,39,35,238 };

Connector my_conn;

float temp;

float LowerLimit = 20, UpperLimit = 50;

String tagid;

char user[] = "user";

char password[] = "password";

EthernetClient client;

int sensorPin = A0;

void setup()

{

  Serial.begin(115200);

  Ethernet.begin(mac_addr,ethernet_ip);

  my_conn.mysql_connect (server_addr, 3306, user, password);

   // select  device from MAC address of ethernet shield

  String Deviceid, query22, query33;

  // selecting TagId
```

```
 query22 = "SELECT PrimaryDevice FROM monitoringprocess.device WHERE
`UniqueKey` = '90-A2-DA-0E-AE-C5'";
 Deviceid = my_conn.RunQuery(query22);
 query33 = "SELECT TagId FROM monitoringprocess.tag WHERE `DeviceId` = ";
 query33 += Deviceid;
 tagid = my_conn.RunQuery(query33);
}
void loop()
 {
   const String INSER_SQL;
   String stringTemp;
   float volt;
   char charTemp[6];
   String query1, query2, query3;
   int read = analogRead(sensorPin);
   volt = (read/1024.0)*5.0;
   temp = (volt - 0.5)*100.0;
   Serial.println(temp);
   // conversion from float to char array
   dtostrf(temp, 4, 3, charTemp);
   // conversion to string from char array
    for(int i=0;i<sizeof(charTemp);i++)
    {
      stringTemp+=charTemp[i];
     }
    if(temp > UpperLimit || temp <LowerLimit)
     {
         query1="insert into monitoringprocess.tagdata
(AlarmStatus,TagId,LogDate,TagValue) values (1,";

         query1 += tagid;

         query1 += ",now(),";

        }

        else

        {

        query1="insert into monitoringprocess.tagdata (TagId,LogDate,TagValue) values (";

        query1 += tagid;

        query1 += ", now(),";

        }

      query2=stringTemp;

    query3= ")";

    //concading strings.

   query1 += query2;

   query1 += query3;

   int str_len = query1.length() + 1;

   char charINSER_SQL[str_len];

   query1.toCharArray(charINSER_SQL,str_len);

   my_conn.cmd_query(charINSER_SQL);

   delay(1800000); // delay for 30 minuites

   }
```

Appendix D

mysql.h

```
String RunQuery(String s);
```

mysql.cpp

```
String Connector::RunQuery(String s) // To perform the SELECT query
{
  String id;
  int str_len1 = s.length() + 1;
  char query300[str_len1];
  s.toCharArray(query300,str_len1);
  cmd_query(query300);
  get_columns();

            // Now we read the rows.
            row_values *row = NULL;
            do {
              row = get_next_row();

              if (row != NULL)
              {
                id=row->values[0]; // in the form of string

              }

            } while (row != NULL);

            free_columns_buffer();
            free_row_buffer();
            return id;

}

#endif
```

Appendix E

PID.h

```
#ifndef PID_h
#define PID_h

class PID
{

  public:

  //Constants used in some of the functions below
  #define AUTOMATIC 1
  #define MANUAL    0
  #define DIRECT  0
  #define REVERSE  1

  //commonly used functions
**************************************************************************
    PID(double*, double*, double*,        // * constructor.  links the PID to the
Input, Output, and
        double, double, double, int);     //   Setpoint.  Initial tuning parameters
are also set here

    void SetMode(int Mode);               // * sets PID to either Manual (0) or Auto
(non-0)

    bool Compute();                       // * performs the PID calculation.  it
should be
                                          //   called every time loop() cycles.
ON/OFF and
                                          //   calculation frequency can be set
using SetMode
                                          //   SetSampleTime respectively

    void SetOutputLimits(double, double); //clamps the output to a specific range.
0-255 by default, but
                                                                     //it's likely
the user will want to change this depending on
                                                                     //the
application



  //available but not commonly used functions
********************************************************
    void SetTunningparameter(double, double,       // * While most users will set
the tunings once in the
                    double);              //   constructor, this function gives the
user the option
                                          //   of changing tunings during runtime
for Adaptive control
       void SetControllerDirection(int);   // * Sets the Direction, or "Action" of
the controller. DIRECT
                                                                     //   means the
output will increase when error is positive. REVERSE
                                                                     //   means the
opposite.  it's very unlikely that this will be needed
                                                                     //   once it
is set in the constructor.
    void SetSampleTime(int);              // * sets the frequency, in Milliseconds,
with which
```

```
                                          //   the PID calculation is performed.
default is 100


  //Display functions
****************************************************************
      int GetMode();                                           //  inside the PID.
      int GetDirection();                               //

  private:
      void Initialize();

      double dispKp;                          // * we'll hold on to the tuning
parameters in user-entered
      double dispKi;                          //   format for display purposes
      double dispKd;                          //

      double kp;                  // * (P)roportional Tuning Parameter
    double ki;                  // * (I)ntegral Tuning Parameter
    double kd;                  // * (D)erivative Tuning Parameter

      int controllerDirection;

    double *myInput;              // * Pointers to the Input, Output, and Setpoint
variables
    double *myOutput;             //   This creates a hard link between the
variables and the
    double *mySetpoint;           //   PID, freeing the user from having to
constantly tell us
                                  //   what these values are.  with pointers we'll
just know.

      unsigned long lastTime;
      double ITerm, lastInput;

      unsigned long SampleTime;
      double outMin, outMax;
      bool inAuto;
};
#endif
```

PID.cpp

```
#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

#include <PID.h>

/*Constructor (...)*********************************************************
 *    The parameters specified here are those for for which we can't set up
 *    reliable defaults, so we need to have the user set them.
 ***************************************************************************/
PID::PID(double* Input, double* Output, double* Setpoint,
        double Kp, double Ki, double Kd, int ControllerDirection)
{

    myOutput = Output;
    myInput = Input;
    mySetpoint = Setpoint;
```

```
       inAuto = false;

       PID::SetOutputLimits(0, 255);                         //default output limit
corresponds to

       //the arduino pwm limits

    SampleTime = 100;                                               //default
Controller Sample Time is 0.1 seconds

    PID::SetControllerDirection(ControllerDirection);
    PID::SetTunningparameter(Kp, Ki, Kd);

    lastTime = millis()-SampleTime;
}


/* Compute() **********************************************************************
 *     This, as they say, is where the magic happens.  this function should be
called
 *   every time "void loop()" executes.  the function will decide for itself whether
a new
 *   pid Output needs to be computed.  returns true when the output is computed,
 *   false when nothing has been done.
 **********************************************************************************/
bool PID::Compute()
{
   if(!inAuto) return false;
   unsigned long now = millis();
   unsigned long timeChange = (now - lastTime);
   if(timeChange>=SampleTime)
   {
      /*Compute all the working error variables*/
         double input = *myInput;
      double error = *mySetpoint - input;
      ITerm+= (ki * error);
      if(ITerm > outMax) ITerm= outMax;
      else if(ITerm < outMin) ITerm= outMin;
      double dInput = (input - lastInput);

      /*Compute PID Output*/
      double output = kp * error + ITerm- kd * dInput;

         if(output > outMax) output = outMax;
      else if(output < outMin) output = outMin;
         *myOutput = output;

      /*Remember some variables for next time*/
      lastInput = input;
      lastTime = now;
         return true;
   }
   else return false;
}


/*
SetTunningparameter(...)************************************************************
*
 * This function allows the controller's dynamic performance to be adjusted.
 * it's called automatically from the constructor, but tunings can also
 * be adjusted on the fly during normal operation
 ******************************************************************************/
void PID::SetTunningparameter(double Kp, double Ki, double Kd)
{
   if (Kp<0 || Ki<0 || Kd<0) return;
   dispKp = Kp; dispKi = Ki; dispKd = Kd;

   double SampleTimeInSec = ((double)SampleTime)/1000;
   kp = Kp;
   ki = Ki * SampleTimeInSec;
   kd = Kd / SampleTimeInSec;

  if(controllerDirection ==REVERSE)
   {
      kp = (0 - kp);
      ki = (0 - ki);
      kd = (0 - kd);
   }
}

/* SetSampleTime(...) *********************************************************
 * sets the period, in Milliseconds, at which the calculation is performed
 ******************************************************************************/
void PID::SetSampleTime(int NewSampleTime)
{
   if (NewSampleTime > 0)
   {
      double ratio  = (double)NewSampleTime
                      / (double)SampleTime;
      ki *= ratio;
      kd /= ratio;
      SampleTime = (unsigned long)NewSampleTime;
   }
}

/* SetOutputLimits(...)****************************************************
 *     This function will be used far more often than SetInputLimits.  while
 *  the input to the controller will generally be in the 0-1023 range (which is
 *  the default already,)  the output will be a little different.  maybe they'll
 *  be doing a time window and will need 0-8000 or something.  or maybe they'll
 *  want to clamp it from 0-125.  who knows.  at any rate, that can all be done
 *  here.
 **************************************************************************/
void PID::SetOutputLimits(double Min, double Max)
{
   if(Min >= Max) return;
   outMin = Min;
   outMax = Max;

   if(inAuto)
   {
         if(*myOutput > outMax) *myOutput = outMax;
         else if(*myOutput < outMin) *myOutput = outMin;

         if(ITerm > outMax) ITerm= outMax;
         else if(ITerm < outMin) ITerm= outMin;
   }
}

/* SetMode(...)****************************************************************
 * Allows the controller Mode to be set to manual (0) or Automatic (non-zero)
 * when the transition from manual to auto occurs, the controller is
 * automatically initialized
 ******************************************************************************/
void PID::SetMode(int Mode)
{
    bool newAuto = (Mode == AUTOMATIC);
    if(newAuto == !inAuto)
    {  /*we just went from manual to auto*/
        PID::Initialize();
    }
    inAuto = newAuto;
}

/* Initialize()****************************************************************
 *     does all the things that need to happen to ensure a bumpless transfer
 *  from manual to automatic mode.
 ******************************************************************************/
void PID::Initialize()
{
   ITerm = *myOutput;
   lastInput = *myInput;
   if(ITerm > outMax) ITerm = outMax;
   else if(ITerm < outMin) ITerm = outMin;
}

/* SetControllerDirection(...)*************************************************
 * The PID will either be connected to a DIRECT acting process (+Output leads
 * to +Input) or a REVERSE acting process(+Output leads to -Input.)  we need to
 * know which one, because otherwise we may increase the output when we should
 * be decreasing.  This is called from the constructor.
 ******************************************************************************/
void PID::SetControllerDirection(int Direction)
{
   if(inAuto && Direction !=controllerDirection)
   {
         kp = (0 - kp);
      ki = (0 - ki);
      kd = (0 - kd);
   }
   controllerDirection = Direction;
}

int PID::GetMode(){ return  inAuto ? AUTOMATIC : MANUAL;}
int PID::GetDirection(){ return controllerDirection;}
```

Appendix F

The website code is uploaded as a zip file in the fronter. It is separate file.