Electrical Energy Efficient System Design using Automation and Artificial Intelligence for Classrooms

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science & Engineering

under the Supervision of

Mr. Punit Gupta

Assistant Professor, Department of Computer Science

By

ANISHA GUPTA

(Enrollment No. - 111306)

to

Jaypee University of Information and Technology

Waknaghat, Solan, Himachal Pradesh

Certificate

This is to certify that project report entitled “**Electrical Energy Efficient System Design using Automation and Artificial Intelligence for Classrooms**”, submitted by **ANISHA GUPTA** (enrolment number 111306) in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date :

Ms. Punit Gupta

Assistant Professor

Department of Computer Science & Engineering

Jaypee University of Information Technology

Acknowledgement

It is not possible to prepare a project report without the assistance & encouragement of other people and this one is certainly no exception. On the very outset of this report, I would like to extend my sincere obligation towards the personages who have helped me in this endeavor. Without their active guidance, help, cooperation & encouragement, I could not have made headway in the project. First and foremost, I would like to express my sincere gratitude to my project supervisor, Mr. Punit Gupta, Assistant Professor, Department of Computer Science. I am privileged to experience a sustained interest from his side even after his busy schedule, which fuelled my enthusiasm even further. He always inspired me to think broad and out of the box. I would also like to thank the laboratory staff for their invaluable support and infrastructure.

ANISHA GUPTA

Department of Computer Science & Engineering

Jaypee University of Information Technology

Table of Content

List of Figures

List of Tables

Abstract

With the growing need for energy efficiency and conservation of resources, we are in the constant process of developing new ways to utilize the maximum out of the resources we use and to minimize their wastage. Also, with the ever-increasing dependence on technology, we desire to automate everyday tasks so as to reduce human involvement and maximize human comfort. We have seen a recent increase in the popularity of home automation systems and escalating interests in the application domain of 'Internet of Things'. This report presents the design and implementation of an Ethernet-based intelligent automated system for conserving electrical energy which can be deployed in large organizations such as a University or an office. The proposed system works on automation, so that the electrical devices and switches can be remotely controlled and monitored without any human intervention. It uses the available infrastructure in a classroom that includes the Ethernet connectivity so as to minimize the cost criteria. It is monitored and controlled remotely from a web server located at the control room using the Internet or the Intranet connectivity. Unlike home automation systems, it works according to the schedule of the classes, stored in a database which resides at server side, through which commands can be sent to a microprocessor hardware unit installed in the classroom which can toggle the light switches as per the current or the updated schedule of the class. Besides the criteria of a class-schedule, the system also aims to conserve energy using artificial intelligence based on real-time occupancy of the classroom, using motion and proximity sensors, and controls the lights accordingly, so as to minimize energy wastage. The proposed outcome of the project aims as multiple benefits, vis-à-vis, saving on electricity bills of the University or any other organization it is deployed in, eliminating human involvement and manpower which is often required to manually toggle the lights and electrical device on/off, and last but most importantly, conserve the precious natural resources by reducing electrical energy consumption.

CHAPTER 1

INTRODUCTION

1.1 INTERNET OF THINGS (IoT)

The ***Internet of Things (IoT)*** is a scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT has evolved from the convergence of wireless technologies, micro-electromechanical systems (MEMS) and the Internet.

It refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

The IoT extends internet connectivity beyond traditional devices like desktop and laptop computers, smartphones and tablets to a diverse range of devices and everyday things that utilize *embedded technology* to communicate and interact with the external environment, all via the Internet.

Examples of objects that can fall into the scope of Internet of Things include *connected security systems, thermostats, cars, electronic appliances, lights in household and commercial environments, alarm clocks, speaker systems, vending machines* and more.

As far as the reach of the IoT is concerned, there are more than 12 billion devices that can currently connect to the Internet, and researchers at IDC estimate that by 2020 there will be 26 times more connected things than people.

1.1.1 The Visions that define the Internet of Things

The Internet of Things conceptually embodies futuristic visions of automating the minutiae of our lives. Ideally, IoT will optimize every aspect of our morning routine, guided by our preferences for shower temperature, coffee timing, food preparation, etc. It will make our travel arrangements intelligently, using up-to-the-minute traffic and weather data. It will refine our workflows, prioritizing tasks and projects based on ongoing assessments in real time of what is happening throughout our organization. The Internet of Things will maintain our appliances and vehicles, determining when they are next due for service, cleaning, or – in the case of our refrigerators – restocking (and making appropriate arrangements, such as repair appointments and grocery orders). It will enable our cars to communicate with other cars on the road as they self-drive us to and fro. It will regulate our lights, heat, AC, and other home appliances and devices, turning them on and off as we enter and exit rooms and as they "learn" our schedule. And that's not all.

In short, IoT has the power to meet our every whim and every need before we even consciously realize what we want, thanks to intense interconnectedness and automation.

1.2.1 Intel® IoT Platform

The Internet of Things (IoT) has enormous potential to drive economic value and social change. But with 85% of things still unconnected and security threats pervasive, the industry has yet to tap IoT’s enormous potential.

The ***Intel® IoT Platform*** breaks down these obstacles. It provides an end-to-end platform for connecting the unconnected—allowing data from billions of devices, sensors, and databases to be securely gathered, exchanged, stored, and analyzed across multiple industries.

1.2.2 Key Benefits of the Intel IoT Platform

Security: Deliver trusted data with a tight integration of hardware- and software-based security that starts where data is most resilient to attack

Interoperability: Utilize technologies that seamlessly communicate to one another, help accelerate time to market, and reduce the cost of deploying and maintaining IoT solutions

Scalability: Achieve scalable compute from edge to cloud with processors from Intel® Quark™ to Intel® Xeon®, and Intel®-based devices, gateways, and data centres solutions

Manageability: Get advanced data management and analytics from sensor to data center

The Intel IoT Platform helps deliver innovations to market faster, reducing solution complexity and delivering actionable intelligence by offering a defined, repeatable foundation for how devices will connect and deliver trusted data to the cloud. And it allows OEMs, SIs, and vertical industries to to develop and deploy solutions using building blocks on the Intel IoT Platform.

These solutions will help provide a repeatable foundation for the IoT and free up developers' time to focus on building solutions that expertly address specific customer pain points. With the Intel IoT Platform, Intel is helping IoT innovations get to market faster that will allow IoT to reach its potential worldwide.

1.2 AUTOMATION

The dictionary defines ***automation*** as “the technique of making an apparatus, a process, or a system operate automatically.”

We define automation as “the creation and application of technology to monitor and control the production and delivery of products and services.”

Using our definition, the automation profession includes “everyone involved in the creation and application of technology to monitor and control the production and delivery of products and services”; and the automation professional is “any individual involved in the creation and application of technology to monitor and control the production and delivery of products and services.”

Automation is the use of machines or technologies to optimize productivity in the production of goods and delivery of services. The correct incentive for applying automation is to increase productivity, and/or quality beyond that possible with current human labor levels so as to realize economies of scale, and/or realize predictable quality levels. It involves use of various control systems for operating equipment such as machinery, processes in factories, boilers and heat treating ovens, switching in telephone networks, steering and stabilization of ships, aircraft and other applications with minimal or reduced human intervention. Some processes have been completely automated.

The biggest benefit of automation is that it *saves labor*, however, it is also used to *save energy and materials* and to *improve quality, accuracy and precision*.

Automation has been achieved by various means including mechanical, hydraulic, pneumatic, electrical, electronic and computers, usually in combination. Complicated systems, such as modern factories, airplanes and ships typically use all these combined techniques.

Automation encompasses many vital elements, systems, and job functions.
It provides benefits to virtually all of industry. Here are some examples:

- **Manufacturing** , including food and pharmaceutical, chemical and petroleum, pulp and paper
- **Transportation** , including automotive, aerospace, and rail
- **Utilities** , including water and wastewater, oil and gas, electric power, and telecommunications
- **Defense**
- **Facility operations** , including security, environmental control, energy management, safety, and other building automation
- And many others

Automation crosses all functions within industry from installation, integration, and maintenance to design, procurement, and management. Automation even reaches into the marketing and sales functions of these industries.
Automation involves a very broad range of technologies including ***Robotics*** and expert systems, telemetry and communications, electro-optics, Cybersecurity, process measurement and control, sensors, wireless applications, systems integration, test measurement, and many, many more.

2.2 Automation Tools

Engineers can now have numerical control over automated devices. The result has been a rapidly expanding range of applications and human activities. Computer-aided technologies (or CAx) now serve as the basis for mathematical and organizational tools used to create complex systems.

Information technology, together with industrial machinery and processes, can assist in the design, implementation, and monitoring of control systems. One example of an industrial control system is a programmable logic controller (PLC). PLCs are specialized hardened computers which are frequently used to synchronize the flow of inputs from (physical) sensors and events with the flow of outputs to actuators and events.

Different types of automation tools exist:

- ANN - Artificial neural network
- DCS - Distributed Control System
- HMI - Human Machine Interface
- SCADA - Supervisory Control and Data Acquisition
- PLC - Programmable Logic Controller
- Instrumentation
- Motion control
- Robotics

1.3 ARTIFICIAL INTELLIGENCE

Artificial intelligence, or **AI**, is the field that studies *the synthesis and analysis of computational agents that act intelligently*. Let us examine each part of this definition.

An **agent** is something that acts in an environment - it does something. Agents include worms, dogs, thermostats, airplanes, robots, humans, companies, and countries. We are interested in what an agent does; that is, how it **acts**. We judge an agent by its actions.

An agent acts **intelligently** when

- what it does is appropriate for its circumstances and its goals,
- it is flexible to changing environments and changing goals,
- it learns from experience, and
- it makes appropriate choices given its perceptual and computational limitations.

An agent typically cannot observe the state of the world directly; it has only a finite memory and it does not have unlimited time to act.

Artificial intelligence (AI) is the intelligence exhibited by machines or software. It is an academic field of study which studies the goal of creating intelligence. John McCarthy, who coined the term in 1955, defines it as "*the science and engineering of making intelligent machines*".

AI can be implemented in business to improve work processes and add to the functionality of business systems. AI in the business context can refer either to specific software programs or to the ways in which those programs are actually used to help companies and organizations carry out a variety of operations and transactions.

When business AI refers to actual software programs, generally the specific function of an individual program is the focus. For example, if a business uses a translation software program to help handle brief translations, this is an application of artificial intelligence (AI) in business. By executing a task that requires a significant degree of evaluation or higher-order analysis, the translation software is acting as an artificial intelligence.

There are many ***real-world applications*** for artificial intelligence in business. Electronic commerce, better known as e-commerce, is the use of software and machines that have been made to behave like intelligent sales clerks in a store and even like a cashier.

1.4 Difference between Automation, Robotics and Artificial Intelligence

Automation and Robotics are very closely related to each other.

Anything which is done with human efforts is a manual operation. Reduction of human efforts is automation. For example, a semi-automatic washing machine needs human intervention for drying of the clothes while a fully-automatic washing machine needs no human intervention.

Robotics is a part of automation for some specific kind of routine jobs. For example, painting a car in manufacturing takes help of robotics for automating the task.

Artificial intelligence is something much more advanced stage than Automation & Robotics. It is the field where intelligence like humans can be developed in machines/processors. ie. thinking machines. We humans develop our Intelligence by learning that means, studying, making mistakes, remembering the mistakes done, discussing with peers, implementing. Artificial intelligence is applying the same process of learning in machines that means instead of giving them instructions for what to do, we make them capable enough to learn the things on their own like we do.

AI is a programming method in which the machine is able to collect information about the outside world and can then apply that knowledge to best perform its function. Automation and robotics treat this knowledge differently. Automated machines are unable to collect knowledge and can't be programmed with any form of intelligence. Robots can be made with intelligence and they are able to learn from mistakes; this allows the robot to fix problems, if it is exposed to them long enough.

1.5 EMBEDDED SYSTEMS

An embedded system can be broadly defined as a device that contain tightly coupled hardware and software components to perform a single function, forms part of a larger system, is not intended to be independently programmable by the user, and is expected to work with minimal or no human interaction.

Embedded systems often require real time operations and multitasking capabilities. Most embedded systems interact directly with processes or the environment, making decisions based on the inputs. This means it is necessary for the system to be reactive, responding in real time to process inputs to ensure proper operation. These systems operate in constrained environments where memory, computing power, and power supply are limited. Multitasking is the capability to perform many functions in a simultaneous or quasi-simultaneous manner. The embedded controller is often responsible of monitoring several aspects of a system and responding according when need arises.

The concept of embedded system is as old as the concept of an electronic computer, and in a certain way, it can be said to precede the concept of a general purpose computer. If we look at the earliest forms of computing devices, they adhere better to the definition of an embedded system than to that of a general purpose computer.

Embedded systems are found in a variety of common electronic devices, such as consumer electronics (cell phones, digital cameras and calculators), home appliances (microwaves ovens, answering machines, and thermostats), office automation (fax machines, printers, scanners) and automobiles etc. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

1.6 INTEL GALILEO 2nd Generation Development Board

The ***2nd generation Intel Galileo board*** provides a single board controller for maker community, students and professional developers. Based on the Intel Quark SoCX1000, a 32-bit Intel Pentium processor- class system on a chip (SoC), the genuine board provide a full featured offering for a wide range of applications. Arduino-certificated and designed to be hardware, software, and pin compatible with large range of Arduino Uno R3 shields, the board also provides a simple and more cost effective development environment compared to the Intel Atom processor and Intel Core processor based designs. Based on a new micro-architecture, the processor is designed for a two-chip platform consisting of a processor and Platform Controller Hub (PCH). The platform enables higher performance, lower cost, easier validation, and improved x-y footprint. The processor includes Integrated Display Engine, Processor Graphics, PCI Express ports, and Integrated Memory Controller. The processor is designed for desktop platforms. It supports up to 12 Processor Graphics execution units (EUs).

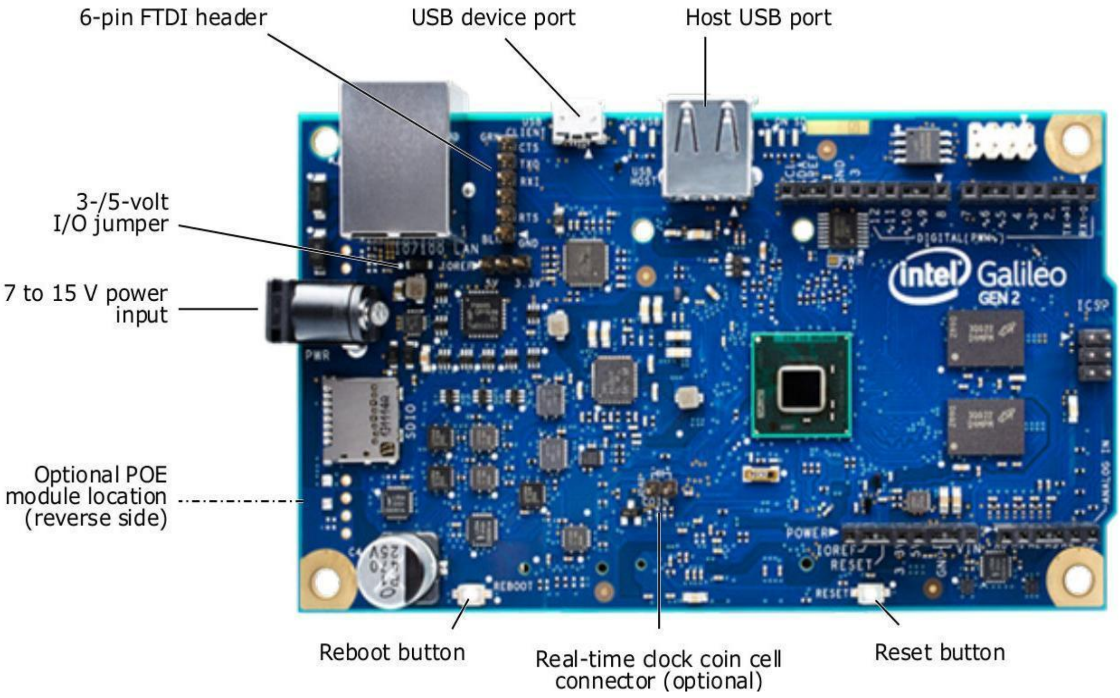


Figure 1: Top view of 2nd Gen Intel Galileo Board

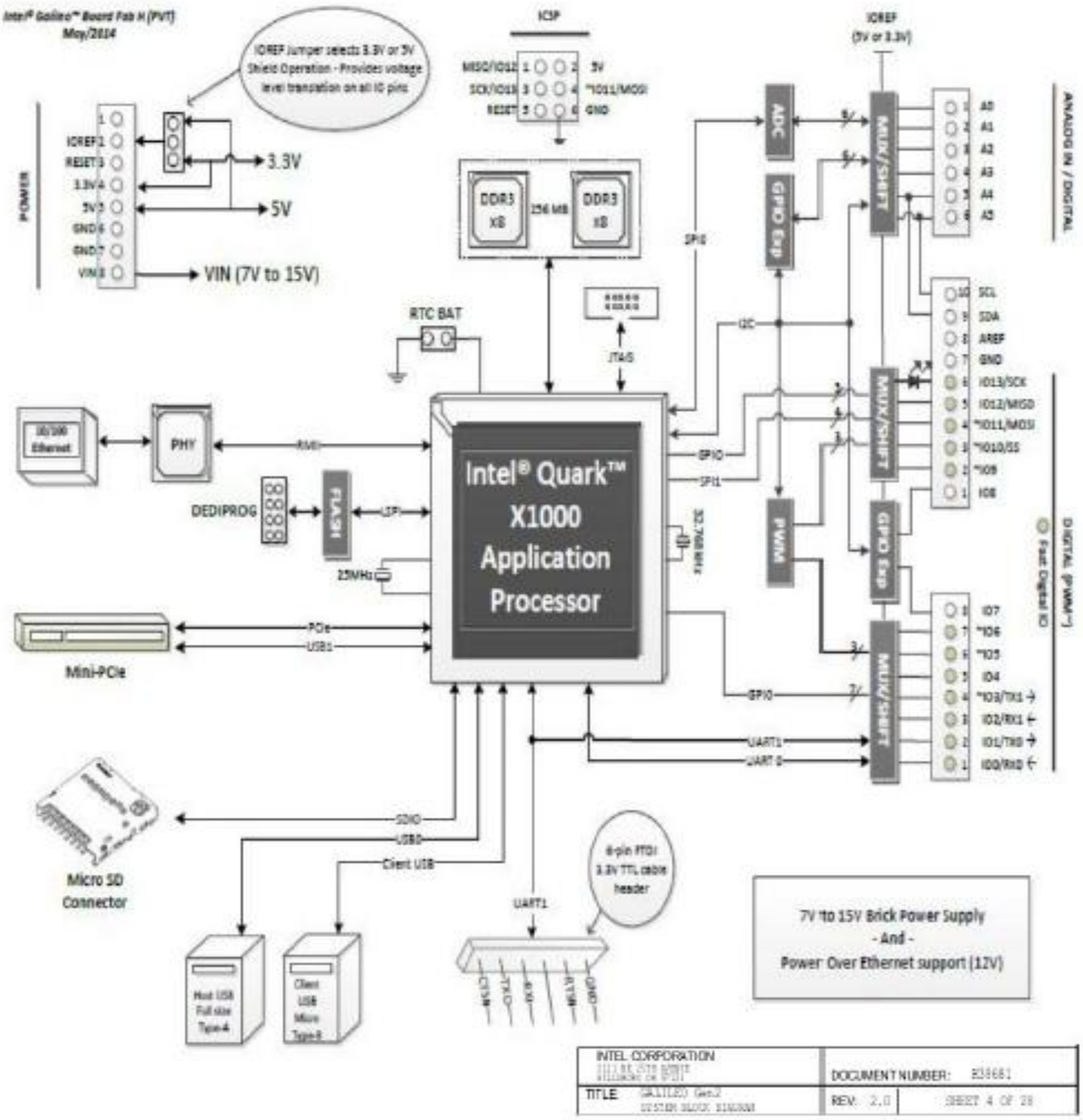


Figure 2: Pin configuration of 2nd Gen Intel Galileo Board

1.6.2 General Characteristics of Intel Galileo 2nd Generation Board

1.6.2.1 Specifications

DIMENSIONS	123.8 mm(L) x 72.0 mm(W)
ATTACHMENTS	Four screw holes 4mm (diameter) Arduino compatible headers containing: • 20 digital I/O • 6 analog inputs • 6 PWMs with 12-bit resolution • 1 SPI master • 2 URATs (1 shared with console URAT) • 1 I^2C master
CONNECTORS	6-pin console URAT (compatible with FTDI USB converters) 6-pin ICSP 10-pin JTAG for debugging RJ45 Ethernet, Power over Ethernet capable USB 2.0 Host(Standard Type A) USB 2.0 Client(micro- USB Type A) Mini-PCI Express 1x slot
POWER	Jack with increased range (7 to 15 V) Supports Power-over-Ethernet (requires PoE module installation) Header for RTC power
BUTTONS	Reset for sketch and attached shield resets 10/100 Mbps Ethernet Reboot to reset the Intel Quark SoC X1000

Table 1: Specifications of Intel Galileo 2nd Gen board

1.6.2.2 Processor Specifications

MODEL	Intel Quark SoC X1000
SPEED	400 MHz
CORES/ THREADS	1/1
INSTRUCTION SET ARCHITECTURE (ISA)	32-bit Intel Pentium processor compatible ISA
L1 CACHE	16 Kb
SRAM	512 Kb on- die, embedded
PACKAGING	15 mm x 15 mm BGA ACPI- compatible with CPU sleep states
TECHNOLOGIES SUPPORTED	Integrated real time clock (RTC) Optional 3 V coin cell battery for operation between turn on cycles

Table 2: Processor Specifications

1.6.2.3 Storage Specifications

FIRMWARE/ BOOT LOADER	8 MB NOR Flash
DRAM	256 MB DDR3, 800 MT/s
SD CARD (OPTIONAL)	Up to 32 GB
USB	Compatible with any USB 2.0 storage device (USB drive/ stick)
EEPROM	8 KB (programmed via the EEPROM library)

Table 3: Storage Specifications

1.6.3 Enhancements

The Intel Galileo board (Gen 2) delivers improved features and functionality in the following areas:

- 12 GPIOs fully native for greater speed and improved drive strength.
- 12-bit PWM for more precise control of servos and smoother response.
- 12 V Power-over-Ethernet capable.
- Power supplies from 7 V to 15 V are supported.
- Serial console URAT header is compatible with FTDI USB converters.
- Console URAT1 can be redirected to Arduino headers in sketches, which can eliminate the need for soft-serial.

1.6.4 Key Component Description

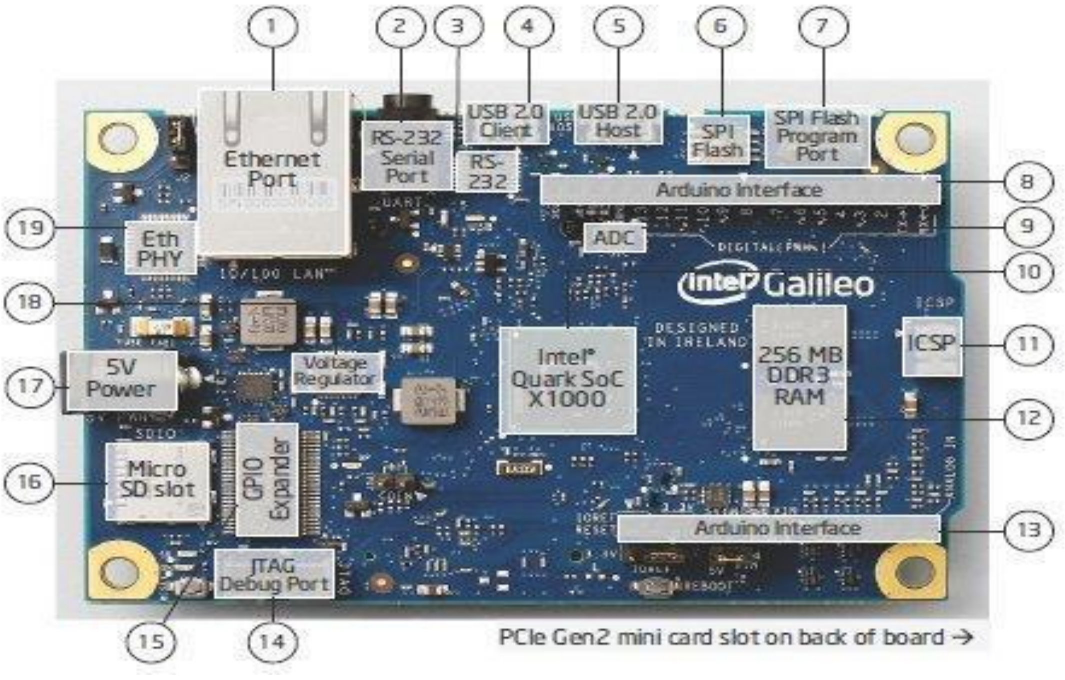


Figure 3: Top view of Galileo Board with Component Specifications

Number	**Component**	**Description**
1	Ethernet Port	10/100 Ethernet connector
2	RS-232 Serial Port	3-pin 3.5 mm jack
3	RS-232	RS-232 transceiver
4	USB 2.0 Client	A fully compliant USB 2.0 Device controller, for programming
5	USB 2.0 Host	Supports up to 128 USB end point devices
6	SPI Flash	To store the bootloader and the latest sketch
7	SPI Flash Program Port	7-pin header for Serial Peripheral Interface (SPI) programming Defaults to 4 MHz to support Arduino Uno shields. Programmable up to 25 MHz.
8	Shield Interface	Complies with Arduino Uno Revision 3 shield pin out.
9	ADC	Analog to Digital Converter
10	Intel Quark SoC X1000	400 MHz 32-bit Intel Pentium instruction set architecture (ISA)-compatible processor
11	ICSP	6-pin in-circuit serial programming (ICSP) header. These pins support SPI communication using the SPI library.
12	256 MB DDR3 RAM	Enabled by the firmware by default.
13	Arduino Interface	Complies with Arduino Uno Revision 3 pinout.
14	JTAG Debug Port	10-pin standard JTAG header for debugging
15	GPIO Expander	GPIO pulse width modulation provided by a single I^2C I/O expander
16	Micro SD slot	Supports micro SD card up to 32 GB
17	5V Power	The board is powered via an AC-to-DC adapter, connected by plugging a 2.1 mm center-positive plug into the board's power jack, output rating of the power adapter is 5V at up to 3A.

18	Voltage Regulator	Generates 3.3 volt supply. Maximum current draw to the shield is 800 mA.
19	Eth PHY	Ethernet Physical layer transceiver

CHAPTER 2

LITERATURE REVIEW

As part of the literature review, here I have reviewed and analyzed the various existing technologies and the products and services based on them, pertaining to one or more domains among those of Automation, Artificial Intelligence and Intelligent systems for energy conservation, Green Technology, and the likes. Here I have specially focussed on Home Automation Systems which aim to conserve energy and automate device control in homes, and the various products recently developed or available in the market which can be easily deployed.

2.1 Z-HOME

ZHOME is a home automation Indian solution based on ***Z-Wave*** *Home Automation* technology offering wireless retrofit solutions to customers who want the best in Automation and Security along with service backing.

- *Retrofit* - Use existing switches, No Re-wiring
- *Cost* - 1/3 of solutions even from far east
- *Complete* - Integrates security, Lighting, AV Control and HVAC all in one
- *Specialties* - Home Automation, Lighting Controls, AV Automation, Building Management System
- *Website* - http://zhome.co.in
- *Industry* - Consumer Electronics

Z-Wave is a wireless communications protocol designed for home automation,

- specifically for remote control applications in residential and commercial environments

- uses a low-power RF radio embedded or retrofitted into electronic devices and systems, such as lighting, access controls, entertainment systems and household appliances.

- has many applications in home automation

- Capabilities being demonstrated even in the roughest & most remote environments like from top of the Himalayas

Zhome allows customers to start small and expand on the go.

They offer custom packages priced low especially for direct customers

The starter kit is the easiest way to start home automation today. The kits have been made to meet the generic requirements of home owners classified as 3BHK, 4BHK and Villas. The basic kit comes with lighting and basic security while the premium packages offer lot more. Including curtain control, AV integration and advanced security features with push notifications on iOS devices. The advanced packages also require additional man hours for installation.

ZHOME
Security & Automation

Basic ZHOME

- 21 Lighting Control
- 1 Home Controller
- 1 WiFi Surveillance camera
- 1 Remote Controller with Scene Capability
- 2 Motion Sensors & 1 Door Sensor

ZHOME Plus

- 39 Lighting Control
- 1 Home Controller
- Premium touch screen with Aluminium flush mount
- 1 Remote Controller with Scene Capability
- 4 Motion sensors & 2 Door Sensors
- IP Camera for Video Door Phone
- 3 AC On/Off Control
- 2 Curtain Controllers

Villa Plus

- 45 Lighting Control
- 2 Home Controllers
- Premium touch screen with Aluminum flush mount
- 1 Remote Controller with Scene Capability
- 5 Motion sensors & 3 Door Sensors
- IP Camera for Video Door Phone
- 1 AV Automation
- 6 AC On/Off Control
- 4 Curtain Controllers
- 1 Yale Door Lock with biometric

AV Room Automation

- 6 Lighting controls
- 1 Handheld Remote
- AV & AC Control (1 Room)

Lighting controls enable you to use your existing switches and add intelligence into them. You can open your main door and the home is lit as per your settings. The night lamps turn on automatically based on outside luminance. The Mosquito coils across your house are turned on and turned off at their preset time.

ZHOME Dimmer & Switch combo devices easily replace existing light switches to offer complete automation of your lights and electronic equipment. The wireless technology used forms a Mesh network allowing you to expand to as many rooms as you want without having to worry about the distance.

ZHOME Combo Switch 100-240V

It contains 2 Relay switches and 1 Dimming control in a single module.

Wireless Switch and Dimmer combo device brings smart lighting capabilities to any home without re-wiring for a variety of lighting and load types. Ideal for upgrading existing lights to smart lighting control, these lighting products offer low cost advanced scene control.

Features & Benefits:

- Compatible with a all variety of switches in India to offer a broad range of design options.
- Fits behind wall boxes even with limited space.
- Support incandescent, halogen, magnetic low voltage, electronic low voltage, fluorescent, compact fluorescent, and LEDs with automatic load sensing.
- Reduce energy waste and lower your electricity bills by setting lights with efficiency settings.
- Create mood lighting and event scenes with ease.
- No problem with power cuts, lights and appliances return to last power state when electricity is restored.
- Support retrofit installations by providing automation through wireless Z Wave technology.

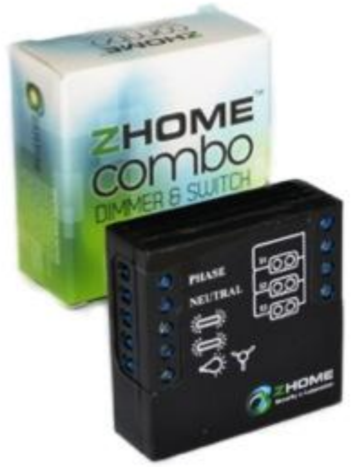


Figure 4: Z HOME Dimmer & Switch

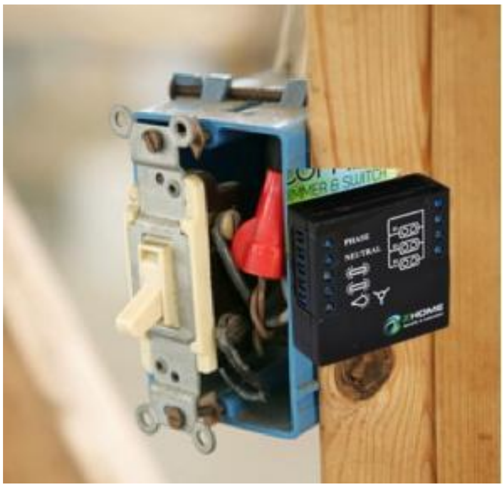


Figure 5: Z HOME Retrofit Solution

2.2 SYNCO LIVING

Synco Living Home automation and control system provide comfort and superior energy efficiency through intelligent home automation. This system provides *Convenient control and switching of HVAC systems, lights, blinds* and more**.**

This system boasts of providing a comfortable home where you save energy costs every month, a pleasant oasis with a perfectly coordinated room climate, and the security of knowing that your home is always monitored even when you are not there.

One system for the entire home

Synco™ living, the intelligent home automation system, turns your four walls into a secure and energy-efficient home. It is a reliable system that thinks along with you and controls many things in the background, including the room temperature and ventilation. Synco living controls your blinds, provides scene control functions, simulates your presence by turning lamps on and off, reports water damage and monitors doors and windows. In addition, a smart phone allows you to access the system at any time – from everywhere.

Comfortable and secure

Synco living's benefits include energy efficient automation that reduces your energy consumption while making your living environment more comfortable. Many functions run automatically so you don't have to think about them or take action. Not only does this eliminate many daily tasks, you also benefit from the system's reliability – extremely accurate values, precise control and great dependability. All components and functions are based on Siemens' many years of experience in building automation.

Adapting to your needs

In addition to easy operation, a comfortable room climate, energy efficiency and security, Synco living also offers a high level of flexibility. For example, if you determine that other scenes would be more suitable, you can customize Synco Living to your new needs, either by programming new scenes or by reprogramming existing ones.

Consistent operation

Synco Living offers operator and display units for different needs. The spectrum ranges from simplest operation from within the room or on the road using a smart phone, operation from a central apartment unit to professional PC tools during commissioning.

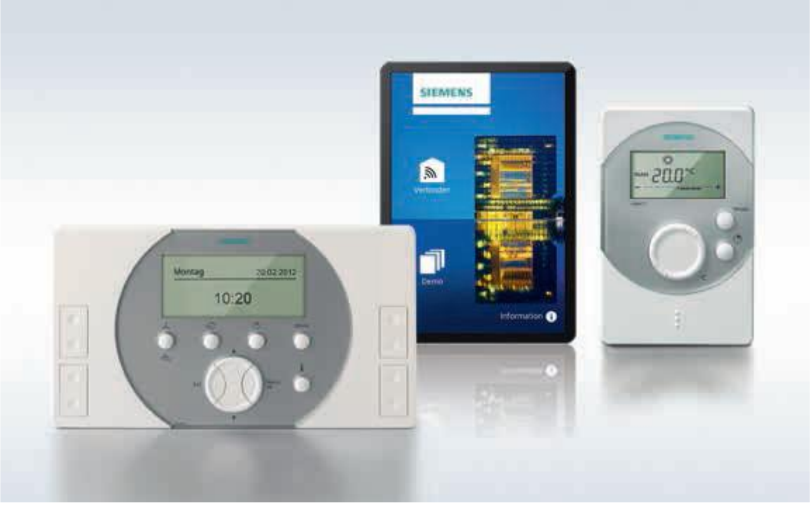
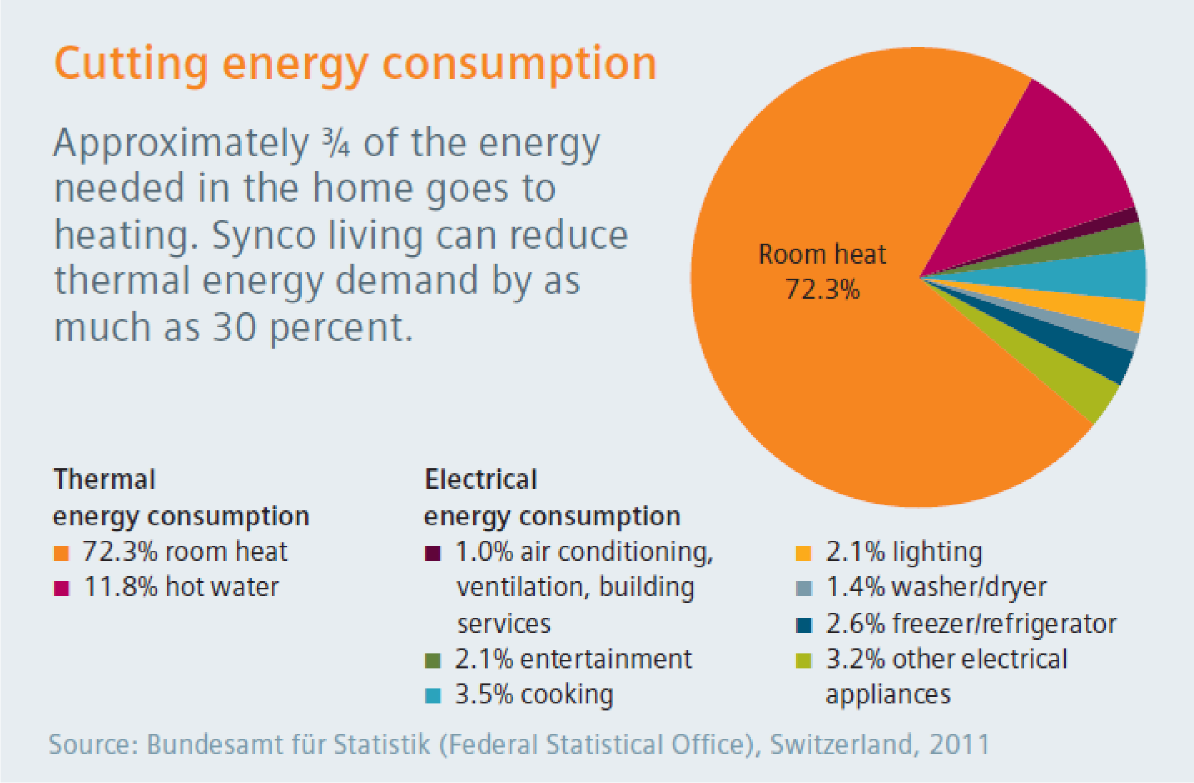


Figure 6: Synco Living Display Units



Cutting energy consumption

Approximately ¾ of the energy needed in the home goes to heating. Synco living can reduce thermal energy demand by as much as 30 percent.

Thermal energy consumption

- 🟧 72.3% room heat
- ■ 11.8% hot water

Electrical energy consumption

- ■ 1.0% air conditioning, ventilation, building services
- 🟩 2.1% entertainment
- 🟦 3.5% cooking
- 🟨 2.1% lighting
- ■ 1.4% washer/dryer
- ■ 2.6% freezer/refrigerator
- ■ 3.2% other electrical appliances

Source: Bundesamt für Statistik (Federal Statistical Office), Switzerland, 2011

Figure 7: Synco Living Statistics

Salient features of Synco Living Home Automation System

- handles many home tasks (HVAC, lights and blinds) that reduce energy consumption and improve comfort level
- up to 30% less heating energy used, lower CO2 emissions
- flexibility of adding components

CHAPTER 3

PROBLEM STATEMENT

"We're data hungry–we want it at our fingertips. At the same time, when it comes to our comfort and convenience, we want it easily and quickly with no distractions." – Geoff Godwin, describing the role of data in the development of cutting-edge home automation systems.

Our world is undergoing changes that force us to think in new ways: demographic change, urbanization, global warming and resource shortages. Maximum efficiency has become the top priority.

On similar lines and after a thorough literature survey, we note that the world today cannot efficiently exist without automation systems, and when talking about such systems, home automation is one such primitive system that comes to our mind, among others.

And it has been observed that often, the extent of such systems has been limited to homes only. The technology used in home automation systems can be easily extended to other realistic scenarios, and organizations such as offices and universities can easily be benefitted by the same.

The motivation behind this project can be summarized as per the following points:

- the need for energy efficiency and conservation of resources,
- the need to automate everyday tasks so as to reduce human involvement,
- the escalating interests in the application domain of "Internet Of Things",
- the growing popularity of home automation systems,
- the need to reduce hefty electricity bills of an organization such as ours, the Jaypee University of Information Technology (with an average bill of about Rs. 12 Lakh per month for the Academic Blocks) .

CHAPTER 4

PROPOSED MODEL

With due consideration to the problem statement, this project proposes to develop an intelligent automated system for conserving electrical energy that can be implemented in large organizations such as this University. This system will work on automation, so that the electrical devices can be remotely controlled and monitored without any human intervention.

4.1 PHASES FOR THE PROPOSED MODEL

PHASE I consists of the following modules :

- Formulation of the project idea keeping in mind the requirements
- Literature survey of the existing models of home automation
- Planning out a schematic approach and defining the proposed model
- Gathering the required hardware components
- Setting up the hardware unit to perform various functionalities
- Setting up client-server communication between the hardware unit and the machine
- Developing a class schedule, which is a time-table of the University according to which the lights of various classrooms would be controlled, and setting up this schedule at the server

PHASE II consists of the following modules :

- Controlling the lights (LEDs an setup on the hardware in this case) from the server machine according to the schedule
- Automating the light control process
- Developing up a Control Panel Web Application to be set up at the main server to perform the functionalities of automating the light control at each classroom through server using relay switches
- Deploying motion detection sensors to further control the lights in classrooms
- Installation of complete setup
- Analysis of energy efficiency after installation

4.2 ARCHITECTURE OF THE PROPOSED MODEL

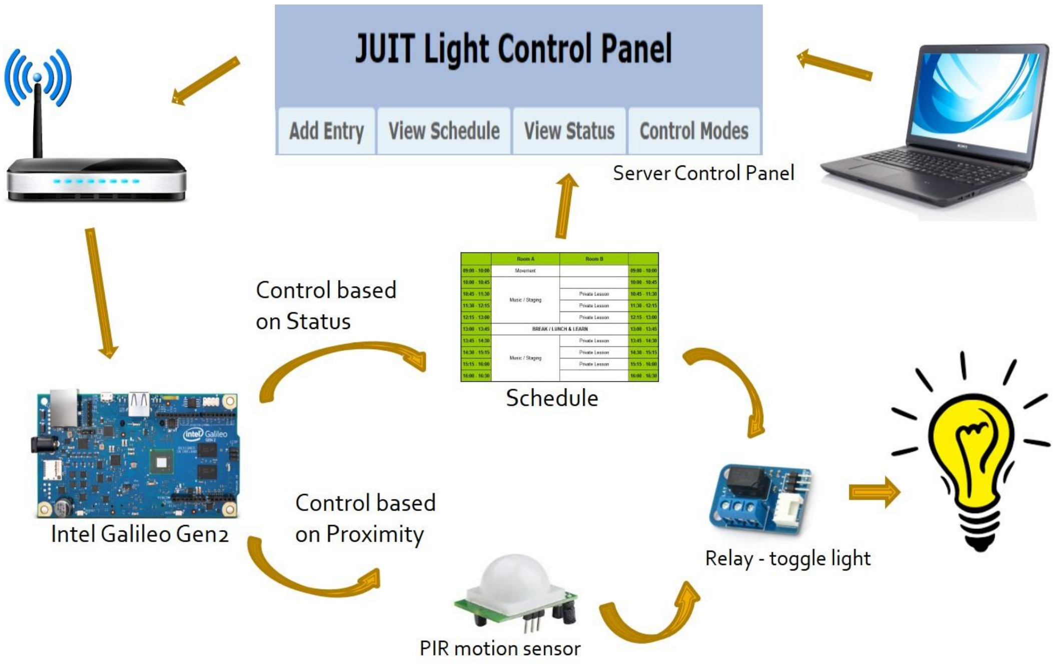


Figure 8 : Architecture of the Proposed Model

4.3 FLOW DIAGRAM

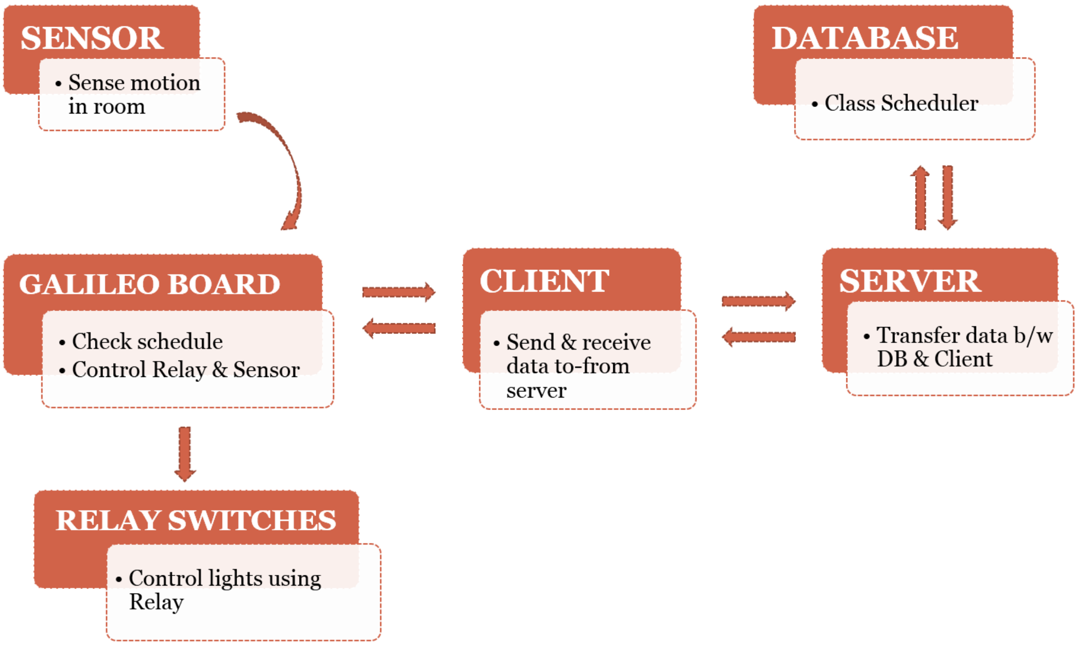


Figure 9 : Flow Diagram of the Proposed Model

4.4 Description and Working of the Proposed Model

Module 1 :

The database consists of a schedule of the ongoing courses, their start and end time, their respective classrooms and their respective IP addresses. Also, another database has a list of the number of lights in the respective classrooms, which are denoted by their IP addresses.

Module 2 :

The Control Panel is a server-side web application which accesses the database,

- to update data of the schedule database, incase there are any changes to be made,
- to retrieve data from the schedule database, to view the current status of the various classrooms and related information,
- to update the status of each classroom according to the stored schedule, both automatically and manually,
- to view the current status in the form of a panel of switches.

Module 3 :

The hardware processor unit, Intel Galileo Gen2, is set-up at the centralized power control unit, and connected to the server through the existing Ethernet connectivity (which may further be extended to wireless connectivity). This hardware unit receives commands from the server to control the lights by toggling them with the help of relay switches at the main power control station.

Module 4 :

Motion detection sensors are strategically set-up in each classroom and connected to the hardware unit through Ethernet connectivity. These sensors send real-time information about the movements in and out of the classroom to the Galileo unit, projecting it as a dataset about the occupancy of the room. The hardware unit then controls the light of the room on the basis of this occupancy data, thus saving maximum electrical energy in real-time.

4.4 SOFTWARE REQUIREMENTS

The various modules of the proposed model require the system to interact with the hardware unit Intel Galileo Gen2, which requires an **Arduino IDE software** to upload various sketches (or program codes for the hardware).

4.4.1 ARDUINO IDE

The Arduino Integrated Development Environment (IDE) is an open-sorce cross-platform application written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a sketch.

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

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

The Arduino IDE uses the GNU tool chain and AVR Libc to compile programs, and uses avrdude to upload programs to the board. Arduino Kit is a software application that provides comprehensive facilities to computer programmers for software development.

We will be using Arduino version 1.5.3 which supports the Intel Galileo Gen 2 board, and can be used for any environment, Windows, Linux or Mac.

4.4.2 FRITZING Electronic Design Software

Fritzing is an open-source initiative to support designers, artists, researchers and hobbyists to take the step from physical prototyping to actual product. We are creating this software in the spirit of Processing and Arduino, developing a tool that allows users to document their Arduino and other electronic-based prototypes, and to create a PCB layout for manufacturing. The complementing website helps users to share and discuss drafts and experiences as well as to reduce manufacturing costs.

Fritzing is essentially an Electronic Design Automation software with a low entry barrier, suited for the needs of designers and artists. It uses the metaphor of the breadboard, so that it is easy to transfer your hardware sketch to the software. From there it is possible to create PCB layouts for turning it into a robust PCB yourself or by help of a manufacturer.

4.5 HARDWARE REQUIREMENTS

As proposed in the block schema, the system intends to use the **Intel Galileo 2nd Generation Development Board** as its main hardware processing unit.

In addition to that, the system requires:

- the Grove Starter Kit Plus – Intel Edition, which would provide components like a Relay, Arduino Shield, LCD display, jumper cables, etc.,
- few LEDs and bulbs,
- few resistors,
- Ethernet cable,
- a router, to create a subnet and simulate the working in a private network,
- PIR sensors or motion detection sensors, proximity sensors, photodiodes etc.

CHAPTER 5

IMPLEMENTATION

5.1 CODE FOR ARDUINO SKETCHES

5.1.1 **Sketch 1 :** Blink.ino

Description :

This sketch is the “Hello World” of Arduino. It causes the on-board LED on pin 13 of Intel Galileo to blink on and off repeatedly, with a delay of one second each.

```
/*
  Sketch to Blink an LED on PIN 13
  Turns on an LED on for one second, then off for one second, repeatedly.
  Pin 13 has an on-board LED connected on the Galileo board.
 */

int led = 13;

void setup()
{
  pinMode(led, OUTPUT);
}

void loop()
 {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

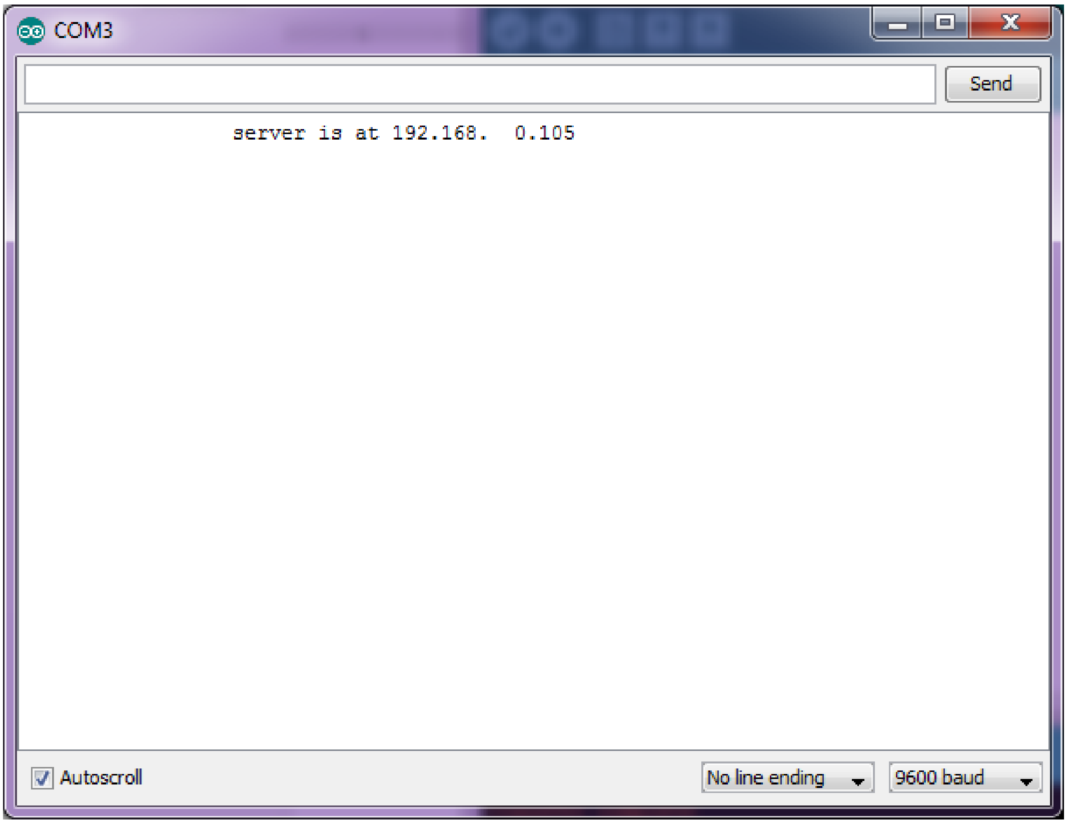
5.1.2 **Sketch 2:** Web Server.ino

Description :

This sketch creates a web server on the host machine which takes in analog input through the pins of the Galileo board, and displays the value of these as the output on the browser. The sketch requires to first establish an Ethernet connection between the host machine and the board, for which it requires an IP and MAC address of the same subnet as that of the host, to be given to the hardware processor. This is achieved by providing the IP and MAC address to the built-in functions as defined under the ‘Ethernet.h’ library.

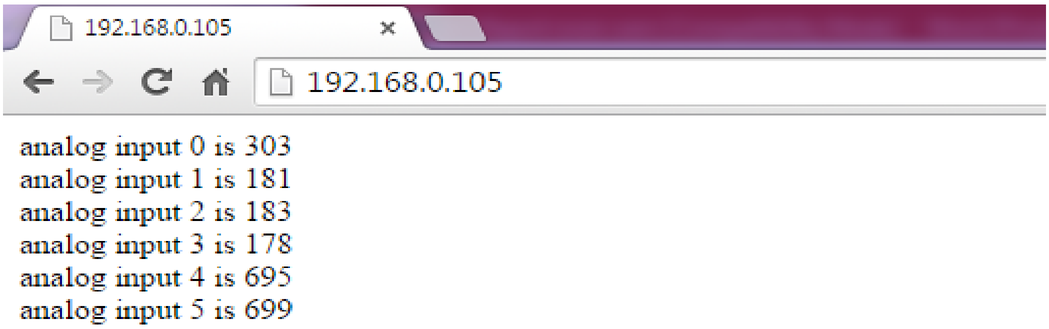
```
/*
  Sketch to create a Web Server –-

 A simple web server that shows the value at the analog input pins.
 Ethernet shield attached to pins 10, 11, 12, 13
*/

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,0,105);
EthernetServer server(80);

void setup()
{
    Serial.begin(9600);
    Ethernet.begin(mac, ip);
    server.begin();
    Serial.print("server is at ");
    Serial.println(Ethernet.localIP());
}

void loop()
{
  EthernetClient client = server.available();
  if (client)
 {
   Serial.println("new client");
   boolean currentLineIsBlank = true;
   while (client.connected())
   {
     if (client.available())
```

```
      {
        char c = client.read();
        Serial.write(c);
        if (c == '\n' && currentLineIsBlank)
        {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close");
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          // add a meta refresh tag, so the browser pulls again every 5 seconds:
          client.println("<meta http-equiv=\"refresh\" content=\"5\">");
          for (int analogChannel = 0; analogChannel < 6; analogChannel++)
           {
            int sensorReading = analogRead(analogChannel);
            client.print("analog input ");
            client.print(analogChannel);
            client.print(" is ");
            client.print(sensorReading);
            client.println("<br />");
          }
          client.println("</html>");
          break;
        }
        if (c == '\n')
       {
          currentLineIsBlank = true;
        }
        else if (c != '\r')
        {
                    currentLineIsBlank = false;
        }
      }
    }
     delay(1);
     client.stop();
        }
}

/*End of program*/
```



```
server is at 192.168.  0.105
```

Figure 11: Connection status on Serial Monitor of Arduino IDE



```
analog input 0 is 303
analog input 1 is 181
analog input 2 is 183
analog input 3 is 178
analog input 4 is 695
analog input 5 is 699
```

Figure 12: Output displayed on the browser

5.1.3 **Sketch 3 :** multipleLED.ino

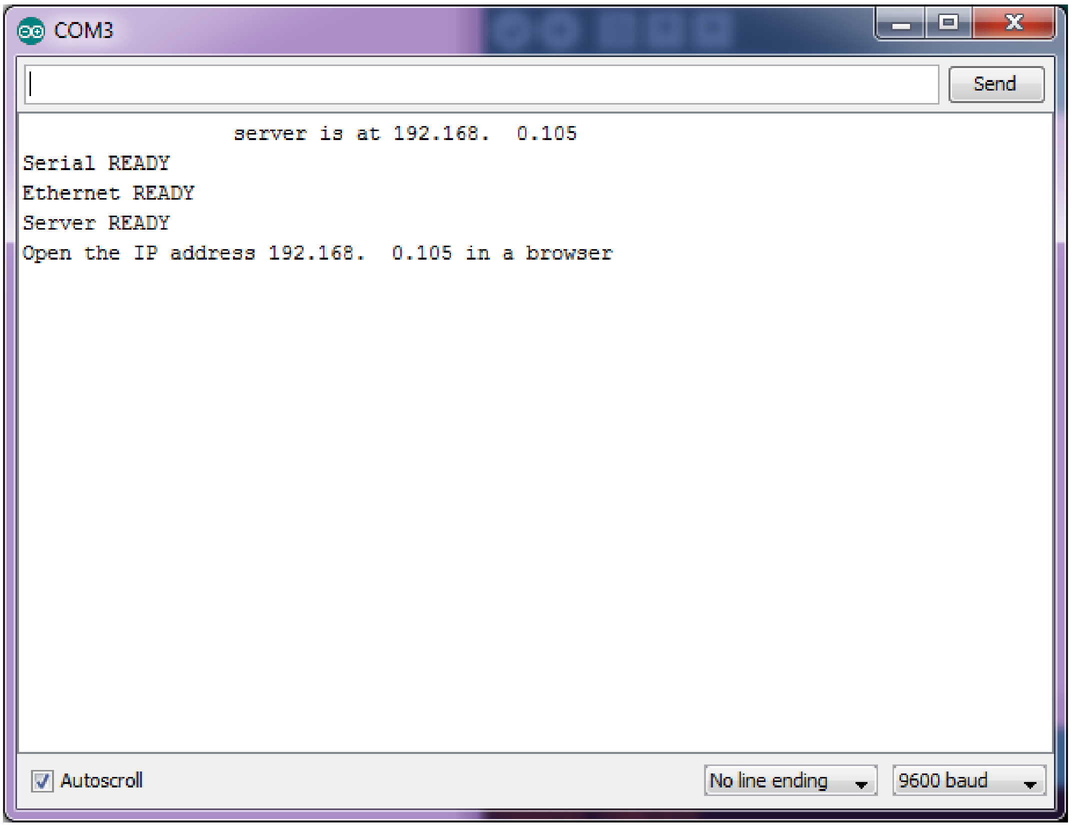
Description :

This sketch creates a web server on the host machine which toggles multiple LEDs connected to the Galileo board, 3 in this case, according to the input given by the user through a web portal on the local IP. The sketch requires to first establish an Ethernet connection between the host machine and the board, for which it requires an IP and MAC address of the same subnet as that of the host, to be given to the hardware unit. This is achieved by providing the IP and MAC address to the in-built functions as defined under the ‘Ethernet.h’ library. The LEDs are controlled through a web control panel on the browser.

```
/*
  Sketch to blink multiple LEDs connected externally on Pins 10, 11 and 12
  IP address extracted from a particular network
*/

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,0,105 };

String inString = String(35);
String Led;

int led[] = {00, 12, 11, 10};
int numofleds = 3;
String value[] = {"on","on","on","on"};
EthernetServer server(80);
String data;

void setup()
{
  Serial.begin(9600);
  Ethernet.begin(mac,ip);
  server.begin();

for (int j = 0; j < (numofleds + 1); j++)
{
   pinMode(led[j], OUTPUT);
}
```

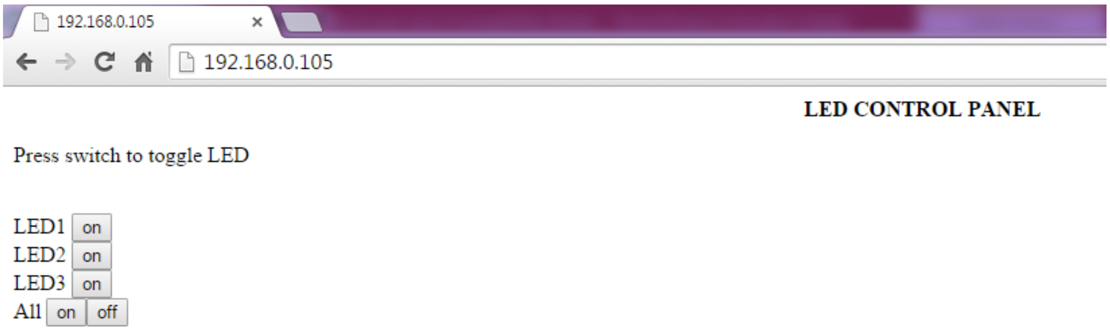
```
Serial.println("Serial READY");
Serial.println("Ethernet READY");
Serial.println("Server READY");
}

void loop()
{
  EthernetClient client = server.available();
  if(client)
  {
    // an http request ends with a blank line
    boolean current_line_is_blank = true;
    while (client.connected())
    {
      if(client.available())
      {
        char c = client.read();
        // if end of the line is read(received a newline
        // character) and since the line is blank, the http request has ended,
        // so a reply can be sent
        if (inString.length() < 35)
        {
          inString.concat(c);
        }
        if (c == '\n' && current_line_is_blank)
        {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();
          client.println("<html><body><form method=get>");
          client.println("<p align=center><b>LED CONTROL PANEL</b></p>");
          client.println("<p> Press switch to toggle LED </p>");
          for(int i=1 ; i < (numofleds + 1) ; i++)
          {
            Led = String("LED") + i;
            if(inString.indexOf(Led+"=on")>0 || inString.indexOf("all=on")>0)
             {
              Serial.println(Led+"on");
              digitalWrite(led[i], HIGH);
              value[i] = "off";
             }
             else if(inString.indexOf(Led+"=off")>0 || inString.indexOf("all=off")>0 )
             {
              Serial.println(Led+"on");
              digitalWrite(led[i], LOW);
              value[i] = "on";
             }
             client.println("<br>"+Led+" <input type=submit name="+Led+
"value="+value[i]+">");
```

```
                }   //end for
              client.println("<br>All <input type=submit name=all value=on><input
type=submit name=all value=off>");
              client.println("</form></body></html>");
              break;
            }    //end if

          if (c == '\n')
               current_line_is_blank = true;
          else if (c != '\r')
               current_line_is_blank = false;
        }    //end if

      }    //end while
      // give the web browser time to receive the data
      delay(10);
      inString = "";
      client.stop();
    }     //end if
}        //end loop function

/*End of program*/
```



```
                 server is at 192.168.  0.105
Serial READY
Ethernet READY
Server READY
Open the IP address 192.168.  0.105 in a browser
```

Figure 13: Connection status on Serial Monitor of Arduino IDE



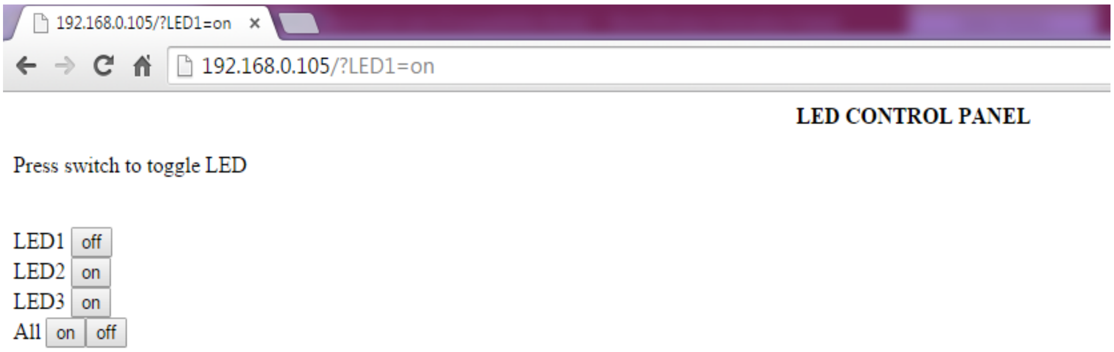
192.168.0.105

LED CONTROL PANEL

Press switch to toggle LED

LED1 on
LED2 on
LED3 on
All on off

Figure 14: Output displayed on the browser when all LEDs are OFF



192.168.0.105/?LED1=on

LED CONTROL PANEL

Press switch to toggle LED

LED1 off
LED2 on
LED3 on
All on off

Figure 15: Output displayed on the browser when LED1 button is pressed ON

5.1.4 Sketch 4 : Relay.ino

Description :

This sketch creates a web server on the host machine which uses a checkbox to toggle a relay connected to Galileo board at PIN 2. The relay is itself connected to AC Power supply and a Bulb. When the user toggles the checkbox button through the web portal, the bulb is switched ON and OFF. The sketch requires to first establish an Ethernet connection between the host machine and the board, for which it requires an IP and MAC address of the same subnet as that of the host.

```
/*
  Sketch to control a Relay connected at PIN2 through a checkbox button in a webpage
  IP address extracted from a particular network
*/

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

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,0,105);
EthernetServer server(80);

String HTTP_req;
boolean r_status = 0;

void setup()
{
   Ethernet.begin(mac, ip);
   server.begin();
   Serial.begin(9600);
   pinMode(2, OUTPUT);      // Relay connected at PIN 2
   Serial.print("server is at ");
   Serial.println(Ethernet.localIP());
}

void loop()
{
   EthernetClient client = server.available();
   if (client)
   {
```

```
boolean currentLineIsBlank = true;
while (client.connected())
 {
    if (client.available())
    {
        char c = client.read();
        HTTP_req += c;
        if (c == '\n' && currentLineIsBlank)
        {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-Type: text/html");
            client.println("Connection: close");
            client.println();
            // send web page
            client.println("<!DOCTYPE html>");
            client.println("<html>");
            client.println("<head>");
            client.println("<title>GALILEO LED Control</title>");
            client.println("</head>");
            client.println("<body>");
            client.println("<h1>RELAY CONTROL</h1>");
            client.println("<p>Click to toggle Bulb on and off through Relay</p>");
            client.println("<form method=\"get\">");
            ProcessCheckbox(client);
            client.println("</form>");
            client.println("</body>");
            client.println("</html>");
            Serial.print(HTTP_req);
            HTTP_req = "";    // finished with request, empty string
            break;
        }
        // every line of text received from the client ends with \r\n
        if (c == '\n') {
            // last character on line of received text
            // starting new line with next character read
            currentLineIsBlank = true;
        }
        else if (c != '\r') {
            // a text character was received from client
            currentLineIsBlank = false;
        }
    } // end if (client.available())
} // end while (client.connected())
```

```
        delay(1);      // give the web browser time to receive the data
        client.stop(); // close the connection
    } // end if (client)
}

// switch bulb and send back HTML for checkbox
void ProcessCheckbox(EthernetClient cl)
{
    if (HTTP_req.indexOf("relay=2") > -1) {  // see if checkbox was clicked
        // the checkbox was clicked, toggle the bulb
        if (r_status) {
            r_status = 0;
        }
        else {
            r_status = 1;
        }
    }

    if (r_status) {    // switch ON
        digitalWrite(13, HIGH);
        // checkbox is checked
        cl.println("<input type=\"checkbox\" name=\"relay\" value=\"2\" \
        onclick=\"submit();\" checked>relay");
    }
    else {             // switch off
        digitalWrite(13, LOW);
        // checkbox is unchecked
        cl.println("<input type=\"checkbox\" name=\"relay\" value=\"2\" \
        onclick=\"submit();\">relay");
    }
}      //end of loop()

/*End of Program */
```

5.1.5 **Sketch 5 :** multiLED_LCD.ino

Description :

This sketch creates a web server on the host machine which creates switches for the LEDs connected to Galileo board. Also, there is an LCD display connected to display the status of LEDs. These LEDs are controlled through the Light Control Panel created at the Server. The sketch requires to first establish an Ethernet connection between the host machine and the board, for which it requires an IP and MAC address of the same subnet as that of the host.

```
/*
  Sketch to control 6 LEDs connected at PINs 13,12,11,10,9,8 through Light Control
Panel at server, and display message on the LCD

*/

#include <Ethernet.h>
#include <SPI.h>
#include <Wire.h>
#include "rgb_lcd.h"



rgb_lcd lcd;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,0,105 };

String inString = String(35);
String Led,Room;

int led[] = {00, 13, 12, 11, 10, 9, 8 }; //Led pins num 0 in arry is not used
int numofleds = 6; //number of leds
String value[] = {"on","on","on","on","on","on","on"}; //startup all led are off
EthernetServer server(80);  // create a server at port 80
String data;

void setup()
{
  lcd.begin(16, 2);  //set the cursor at beginning
  char status[] = {"ON"};
  Serial.begin(9600);
  Ethernet.begin(mac,ip);
  server.begin();

  //set pin mode
  for (int j = 0; j < (numofleds + 1); j++)
  {
    pinMode(led[j], OUTPUT);
  }
   lcd.print("READY TO DISPLAY");
   lcd.setCursor(0, 1);
   lcd.print("STATUS");
   Serial.println("Serial READY");
   Serial.println("Ethernet READY");
   Serial.println("Server READY");
}

void loop()
{
  EthernetClient client = server.available();  // try to get client
  if(client)
  {
    // an http request ends with a blank line
    boolean current_line_is_blank = true;
    while (client.connected())
    {
        if(client.available())
        {
            char c = client.read();
            // if we've gotten to the end of the line (received a newline
            // character) and the line is blank, the http request has ended,
            // so we can send a reply
            if (inString.length() < 35)
              inString.concat(c);

            if (c == '\n' && current_line_is_blank)
            {
              // send a standard http response header
              client.println("HTTP/1.1 200 OK");
              client.println("Content-Type: text/html");
              client.println();
              client.println("<html><body><form method=get>");
              client.println("<p align=center><b>LED CONTROL PANEL</b></p>");
              client.println("<p> Press switch to toggle LED </p>");
```

```
lcd.clear();
for(int i=1 ; i < (numofleds + 1) ; i++)
{

    Led = String("LED") + i;
    Room = String("LT") + i;
    if(i==6)
        Room = String("LAB");

    if( inString.indexOf(Led+"=on")>0 )
    {
      Serial.println(Led+"on");
      lcd.print(Led+" in "+Room+" ON");
      //lcd.setCursor(8, 0);
      digitalWrite(led[i], HIGH);
      value[i] = "off";
    }
    else if( inString.indexOf("all=on")>0 )
    {
      Serial.println(Led+"on");
      lcd.print("ALL LEDs ON");
      //lcd.setCursor(8, 0);
      digitalWrite(led[i], HIGH);
      value[i] = "off";
      break;
    }
    else if( inString.indexOf(Led+"=off")>0 )
    {
      Serial.println(Led+"OFF");
      //lcd.print(Led+" OFF");
      lcd.print(Led+" in "+Room+" OFF");
      digitalWrite(led[i], LOW);
      value[i] = "on";

    }
    else if(inString.indexOf("all=off")>0 )
    {
      Serial.println("ALL LEDs OFF");
      lcd.print("ALL LEDs OFF");
      digitalWrite(led[i], LOW);
      value[i] = "on";
      break;
    }
```

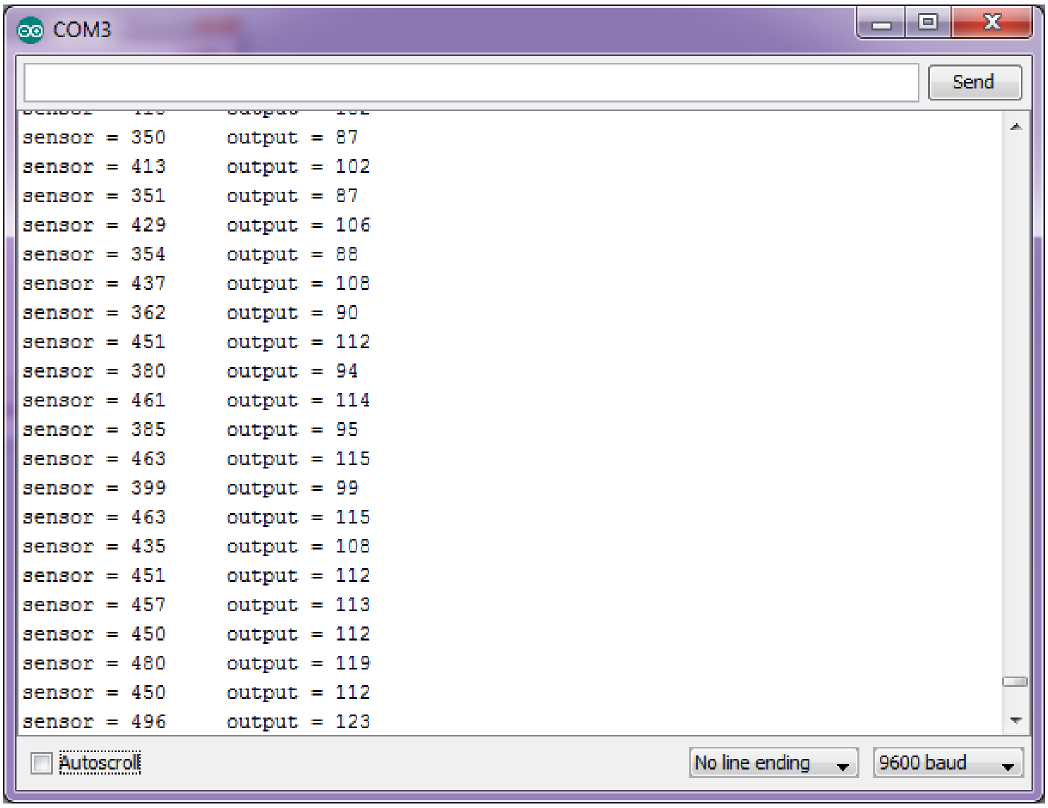
```
                  client.println("<br>"+Led+" <input type=submit name="+Led+"
value="+value[i]+">");
              }

              client.println("<br>All <input type=submit name=all value=on><input
type=submit name=all value=off>");
              client.println("</form></body></html>");
              break;
            }

            if (c == '\n')
            {
              // we're starting a new line
              current_line_is_blank = true;
            }
            else if (c != '\r')
            {
              // we've gotten a character on the current line
              current_line_is_blank = false;
            }

      }   //end of if(cliet)

    }     //end of while
    delay(10);
    inString = "";
    client.stop();

  }
}
```

5.1.6 Sketch 6 : PIRsensor.ino

Description :

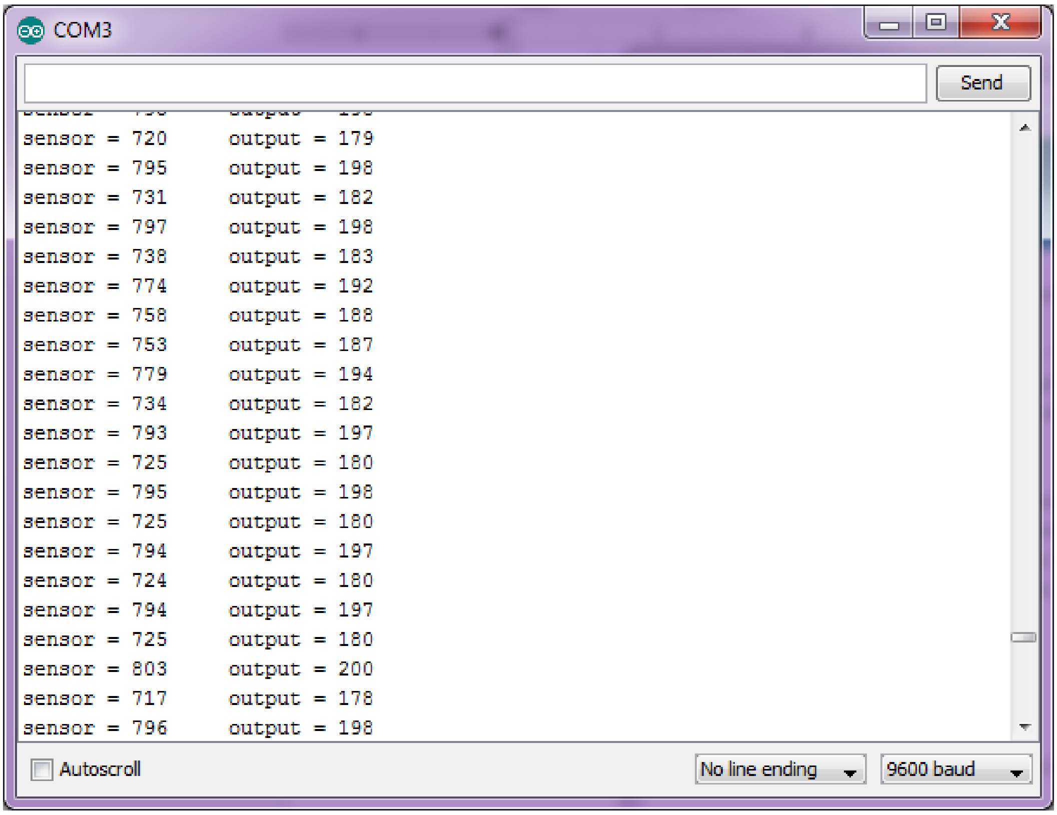
This sketch reads an analog input pin, where the PIR sensor is connected, maps the result to a range from 0 to 255 and uses the result to set the pulsewidth modulation (PWM) of an output pin. It also prints the results to the serial monitor. The PIR sensor is connected to analog pin A0 of Galileo, where its center pin goes to A0 and side pins go to +5V and ground.

```
/*
 Sketch to read analog value from PIR sensor connected to analog pin A0 and print the
values on serial monitor
 */

const int analogInPin = A0;  // Analog input pin that the PIR sensor is attached to
int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop for the analog-to-digital converter to settle
  delay(2);
}
```



```
sensor = 350    output = 87
sensor = 413    output = 102
sensor = 351    output = 87
sensor = 429    output = 106
sensor = 354    output = 88
sensor = 437    output = 108
sensor = 362    output = 90
sensor = 451    output = 112
sensor = 380    output = 94
sensor = 461    output = 114
sensor = 385    output = 95
sensor = 463    output = 115
sensor = 399    output = 99
sensor = 463    output = 115
sensor = 435    output = 108
sensor = 451    output = 112
sensor = 457    output = 113
sensor = 450    output = 112
sensor = 480    output = 119
sensor = 450    output = 112
sensor = 496    output = 123
```

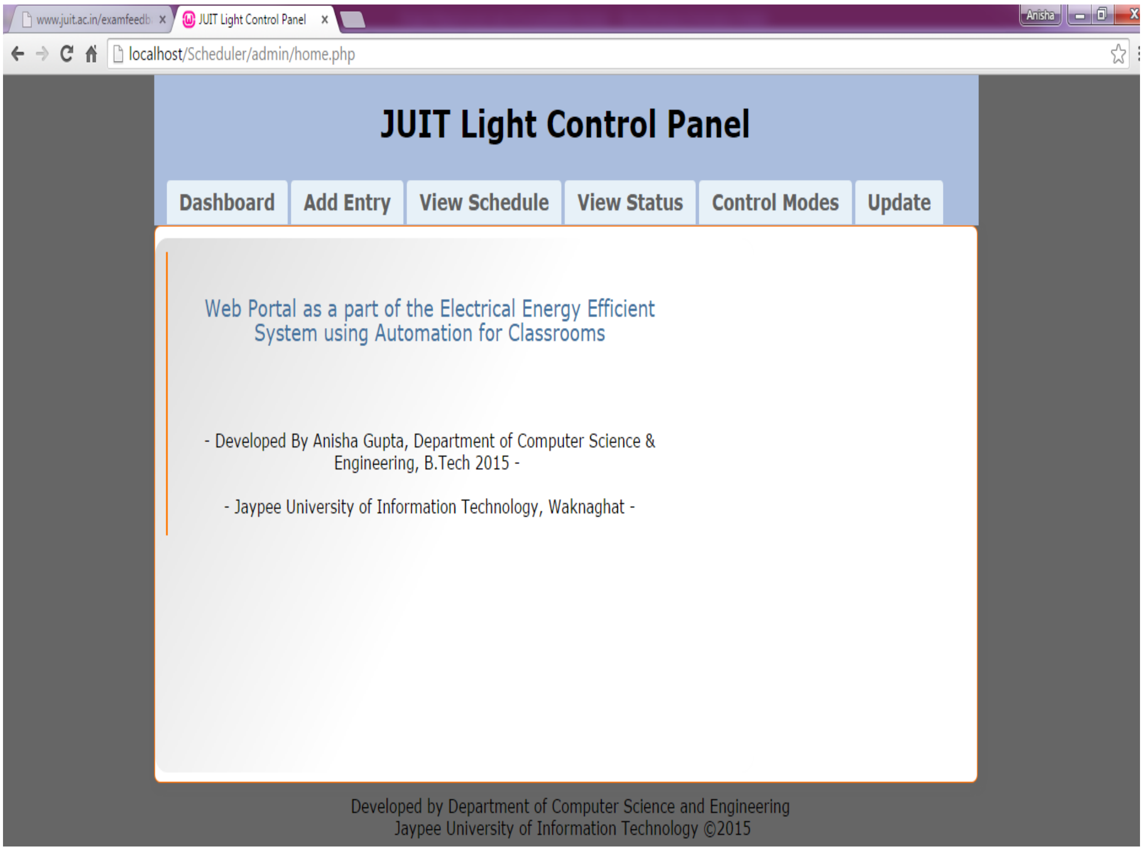
Figure 16: Output analog values on Serial Monitor of Arduino IDE



```
sensor = 720    output = 179
sensor = 795    output = 198
sensor = 731    output = 182
sensor = 797    output = 198
sensor = 738    output = 183
sensor = 774    output = 192
sensor = 758    output = 188
sensor = 753    output = 187
sensor = 779    output = 194
sensor = 734    output = 182
sensor = 793    output = 197
sensor = 725    output = 180
sensor = 795    output = 198
sensor = 725    output = 180
sensor = 794    output = 197
sensor = 724    output = 180
sensor = 794    output = 197
sensor = 725    output = 180
sensor = 803    output = 200
sensor = 717    output = 178
sensor = 796    output = 198
```

5.2 SOFTWARE IMPLEMENTATION

Control Panel Web Application at Server



JUIT Light Control Panel

Web Portal as a part of the Electrical Energy Efficient System using Automation for Classrooms

- Developed By Anisha Gupta, Department of Computer Science & Engineering, B.Tech 2015 -

- Jaypee University of Information Technology, Waknaghat -

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

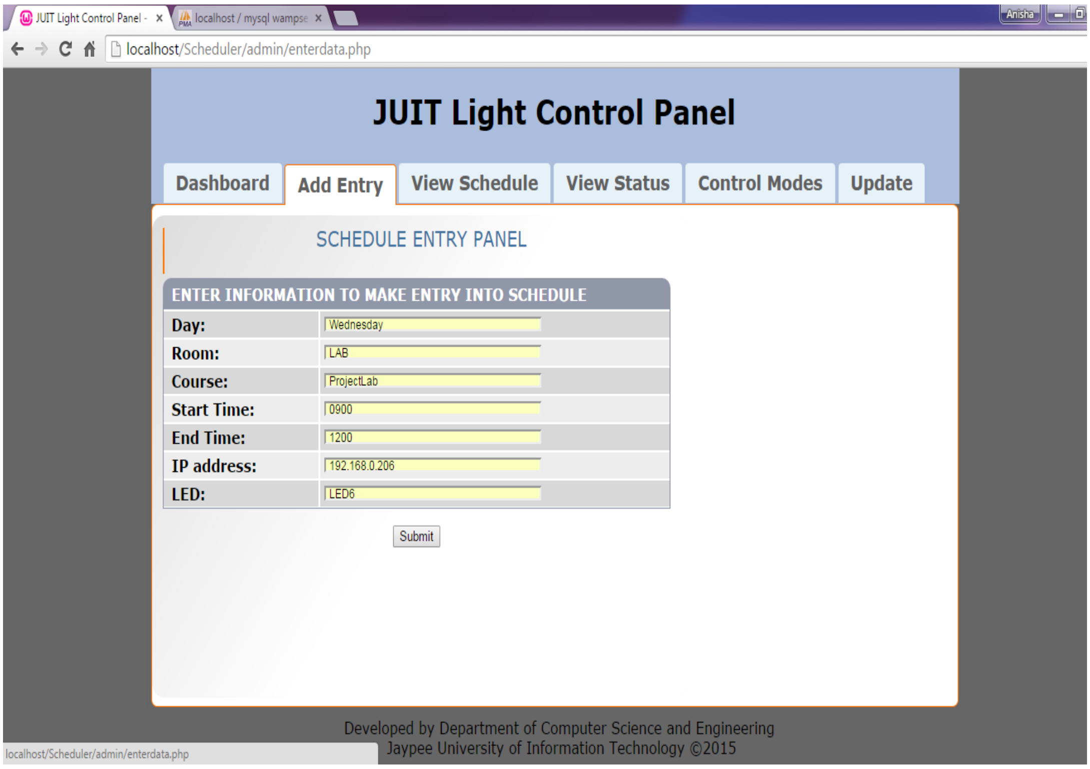
Figure 17: Web Portal for Control Panel – Home Page

The Control Panel Web Application Portal is set up at the server.

This Panel provides various functionalities to the administrator of the system, vis-à-vis :

- add new entry to schedule,
- view day-wise schedule,
- view current status of lights in the various classrooms of the academic block,
- manual and auto control modes,
- update schedule through add and drop entry.

5.2.1 Functionality 1 : ADD ENTRY


JUIT Light Control Panel -
localhost / mysql wampse
Anisha
localhost/Scheduler/admin/enterdata.php
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update
Submit
localhost/Scheduler/admin/enterdata.php

JUIT Light Control Panel

SCHEDULE ENTRY PANEL

ENTER INFORMATION TO MAKE ENTRY INTO SCHEDULE	
Day:	Wednesday
Room:	LAB
Course:	ProjectLab
Start Time:	0900
End Time:	1200
IP address:	192.168.0.206
LED:	LED6

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 18: Web Portal for Control Panel – Add Entry

The ADD ENTRY page provides a form to enter information and make a new entry into the schedule database. Information is entered for each attribute of the database. The admin makes an entry of information about the classes – the day, room number, course, the start and end time of the class, the IP address and the IP address of that particular room.

A sample entry would be –

Day – Wednesday

Room – LAB

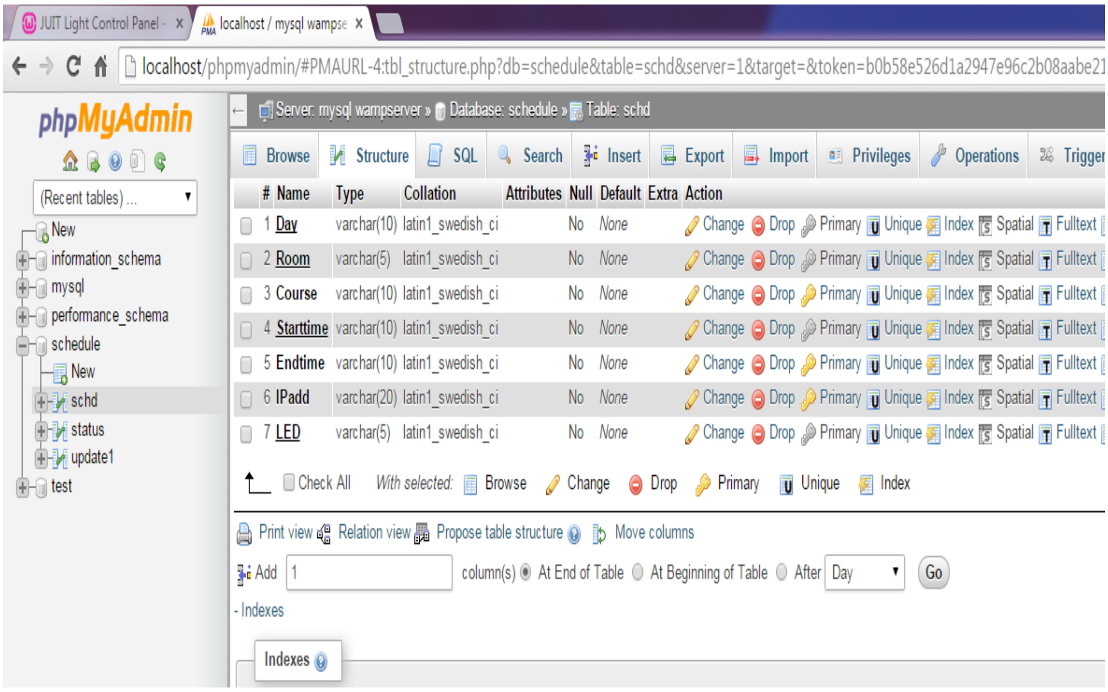
Course – Project Lab

Start Time – 0900 (HRS)

End Time – 1200 (HRS)

IP address – 192.168.0.206

LED# - LED6


JUIT Light Control Panel -
localhost / mysql wampse
localhost/phpmyadmin/#PMAURL-4:tbl_structure.php?db=schedule&table=schd&server=1&target=&token=b0b58e526d1a2947e96c2b08aabe21
phpMyAdmin
(Recent tables) ...
New
information_schema
mysql
performance_schema
schedule
New
schd
status
update1
test
Server: mysql wampserver » Database: schedule » Table: schd
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Trigger

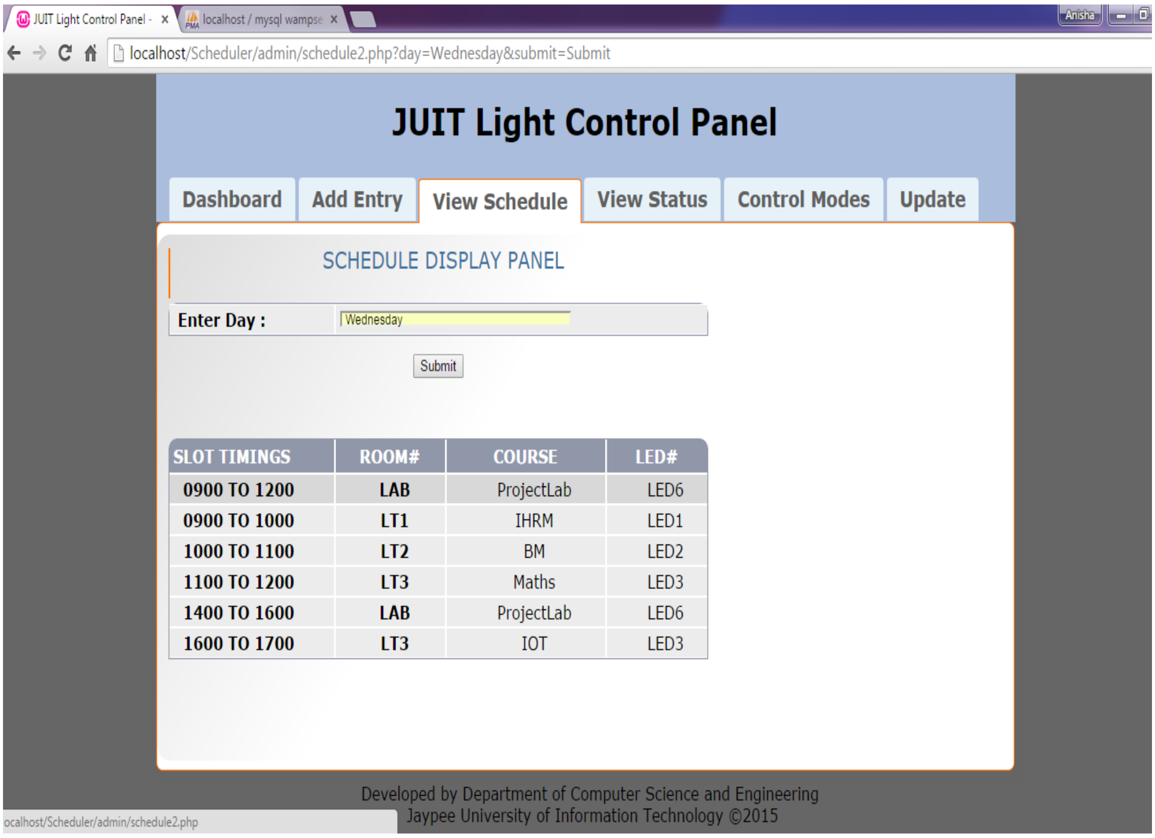
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	Day	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
2	Room	varchar(5)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
3	Course	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
4	Starttime	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
5	Endtime	varchar(10)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
6	IPadd	varchar(20)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext
7	LED	varchar(5)	latin1_swedish_ci		No	None		Change Drop Primary Unique Index Spatial Fulltext

□ Check All With selected: Browse Change Drop Primary Unique Index
Print view Relation view Propose table structure Move columns
Add 1 column(s) ● At End of Table ○ At Beginning of Table ○ After Day
Go
- Indexes
Indexes

Figure 19: Web Portal for Control Panel – Structure of the schedule table in database

The database has a ‘Schedule’ table with the structure as shown above, having the attributes – Day, Room, Course, Start Time, End Time, IP address and LED number.

5.2.2 Functionality 2 : VIEW SCHEDULE



JUIT Light Control Panel -
localhost / mysql wampse
Anisha
localhost/Scheduler/admin/schedule2.php?day=Wednesday&submit=Submit
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update
Submit
ocalhost/Scheduler/admin/schedule2.php

JUIT Light Control Panel

SCHEDULE DISPLAY PANEL

Enter Day : Wednesday

SLOT TIMINGS	ROOM#	COURSE	LED#
0900 TO 1200	LAB	ProjectLab	LED6
0900 TO 1000	LT1	IHRM	LED1
1000 TO 1100	LT2	BM	LED2
1100 TO 1200	LT3	Maths	LED3
1400 TO 1600	LAB	ProjectLab	LED6
1600 TO 1700	LT3	IOT	LED3

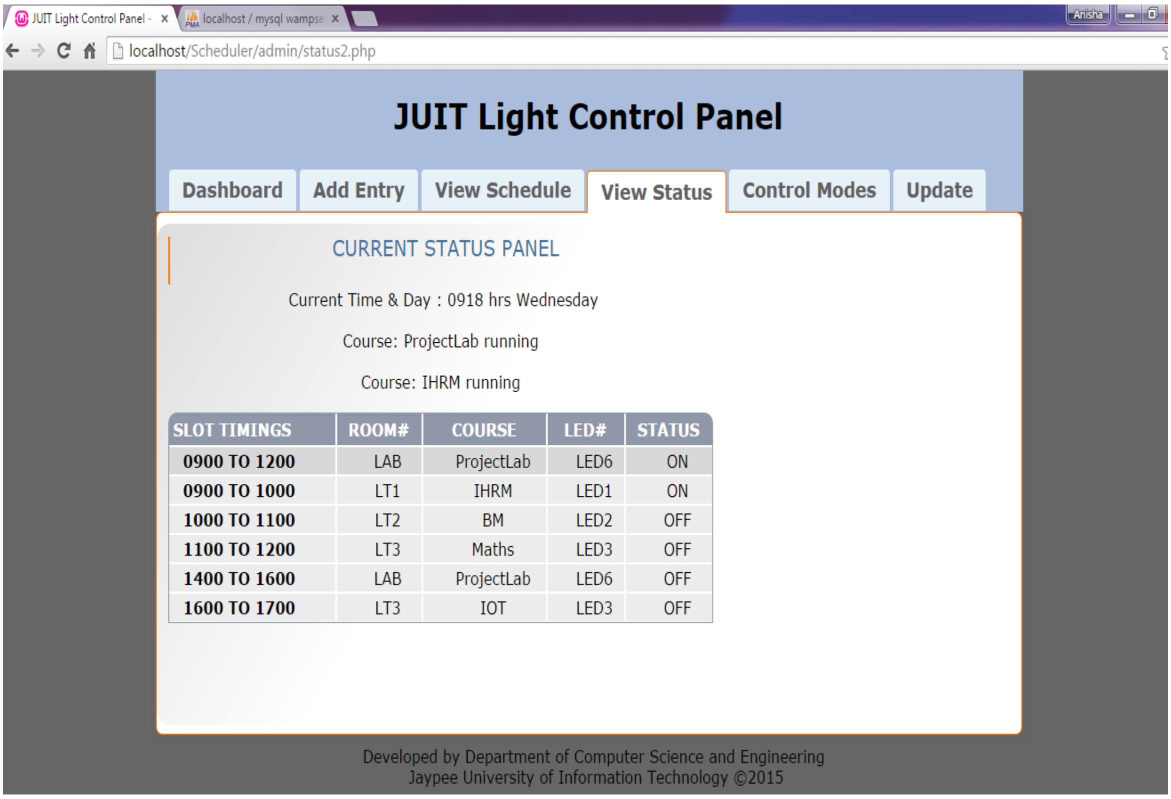
Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 20: Web Portal for Control Panel – Day-wise Schedule Display

This page displays the Schedule of the classes in the university by prompting the user to enter the day.

In a sample entry, the user enters Wednesday in the field and the schedule for Wednesday has been displayed.

5.2.3 Functionality 3 : VIEW CURRENT STATUS


JUIT Light Control Panel -
localhost / mysql wampse
Anisha
localhost/Scheduler/admin/status2.php

JUIT Light Control Panel

CURRENT STATUS PANEL

Current Time & Day : 0918 hrs Wednesday

Course: ProjectLab running

Course: IHRM running

SLOT TIMINGS	ROOM#	COURSE	LED#	STATUS
0900 TO 1200	LAB	ProjectLab	LED6	ON
0900 TO 1000	LT1	IHRM	LED1	ON
1000 TO 1100	LT2	BM	LED2	OFF
1100 TO 1200	LT3	Maths	LED3	OFF
1400 TO 1600	LAB	ProjectLab	LED6	OFF
1600 TO 1700	LT3	IOT	LED3	OFF

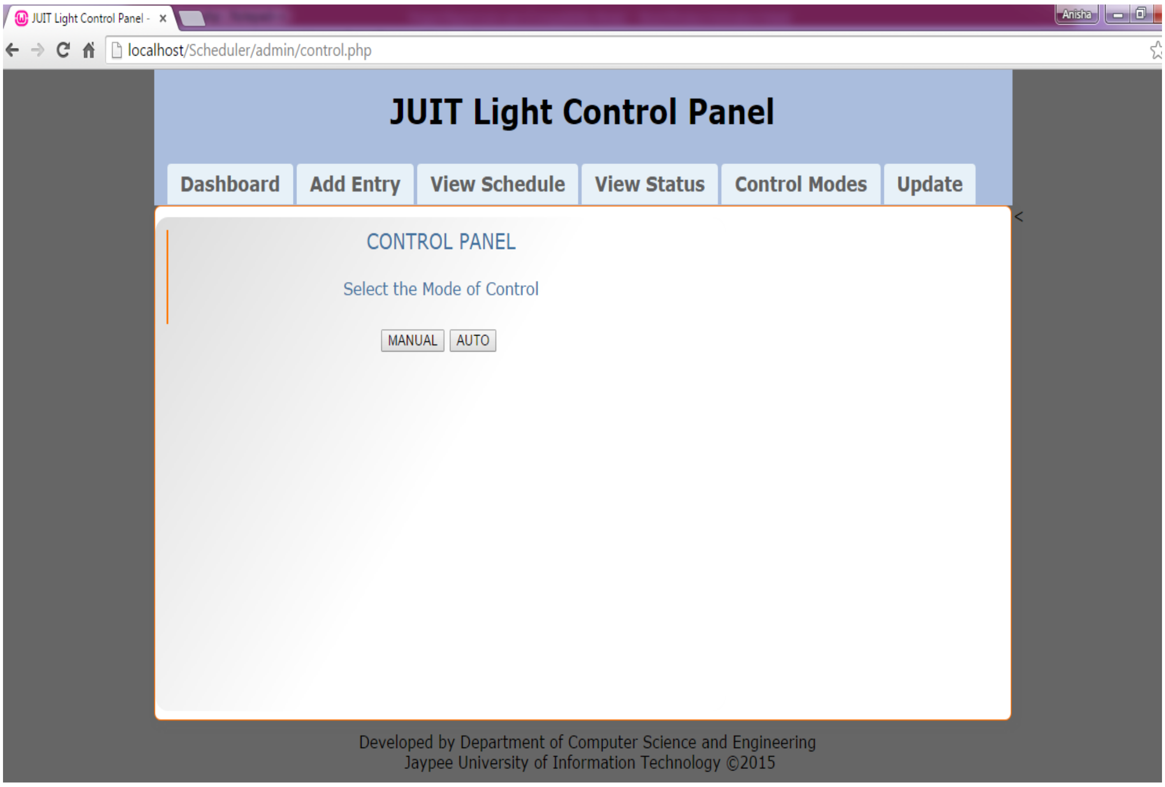
Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 21: Web Portal for Control Panel – Current Status Display

This page displays the Current Status of lights in each room where a class is scheduled for that particular day. It also display the current time and day and the course which is running at that particular moment. Also, the page is automatically refreshed every second, so the data is displayed in real time, and the status is updated according to the schedule.

As shown for the sample data, the courses IHRM and ProjectLab are scheduled on Wednesday 0900 to 1000 hrs and 0900 to 1200 hrs respectively. So, at 0918 hrs on Wednesday, both these courses are running and the LEDs in the respective classrooms/lab are switched ON.

5.2.4 Functionality 4 : CONTROL LIGHTS



CONTROL PANEL

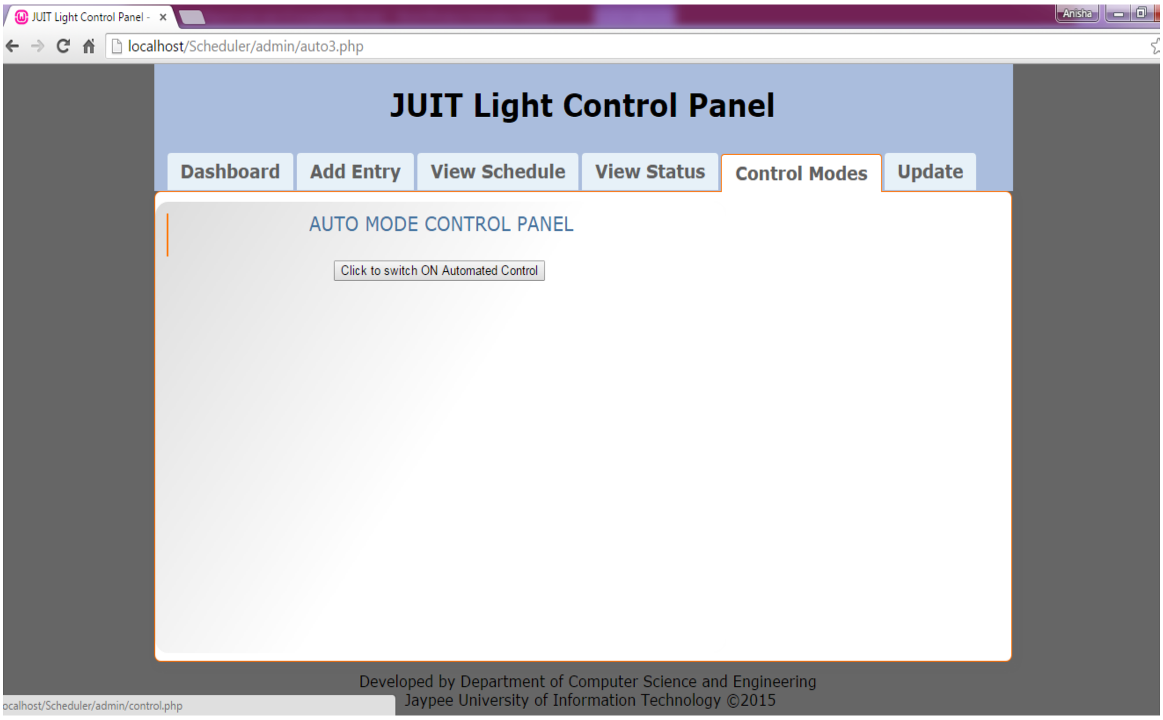
Select the Mode of Control

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 22: Web Portal for Control Panel – Control Options

This page provides two options of controlling lights, vis-à-vis MANUAL and AUTO.

5.2.4.1 Functionality 4a : CONTROL LIGHTS USING AUTO MODE


JUIT Light Control Panel -
localhost/Scheduler/admin/auto3.php
JUIT Light Control Panel
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update
Click to switch ON Automated Control
localhost/Scheduler/admin/control.php

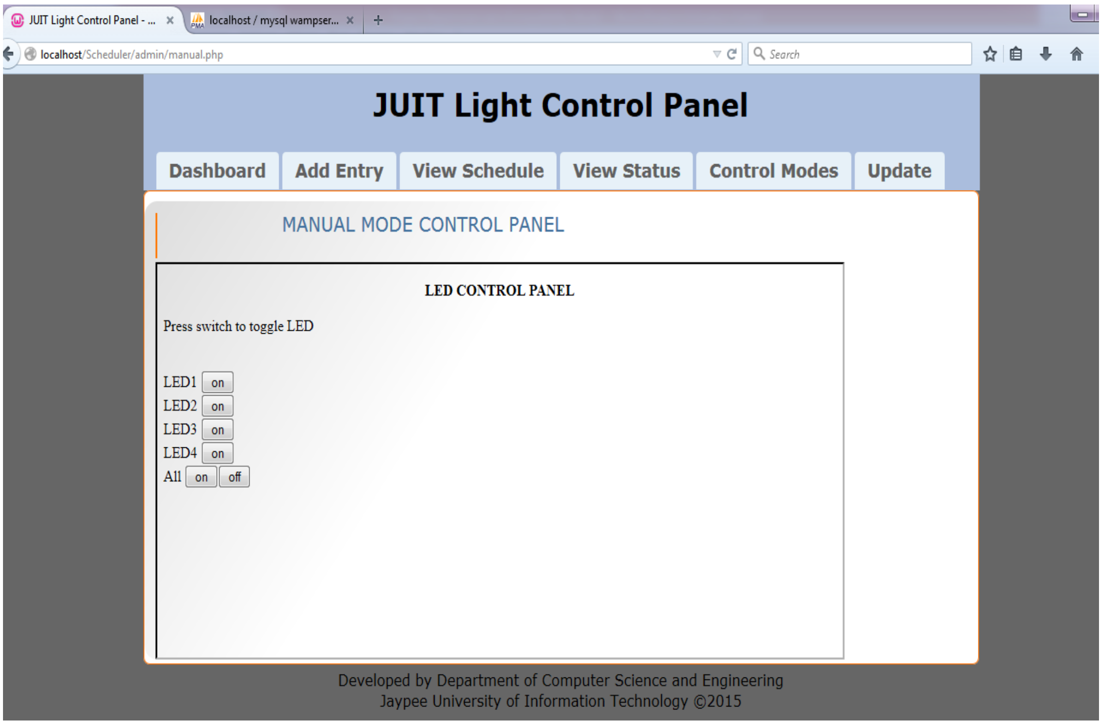
AUTO MODE CONTROL PANEL

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 23: Web Portal for Control Panel – Auto Control Panel

The Auto button on this page, when clicked, initiates the automatic control of lights in the backend according to the current status for that day.

5.2.4.1 Functionality 4b : CONTROL LIGHTS USING MANUAL MODE



JUIT Light Control Panel

MANUAL MODE CONTROL PANEL

LED CONTROL PANEL

Press switch to toggle LED

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 24: Web Portal for Control Panel – Manual Control Panel

On choosing the Manual mode of control, an iframe opens up which calls the IP address of the server provided to the hardware unit. This iframe displays toggle buttons for each light as entered in the schedule. Currently, all four lights are OFF.

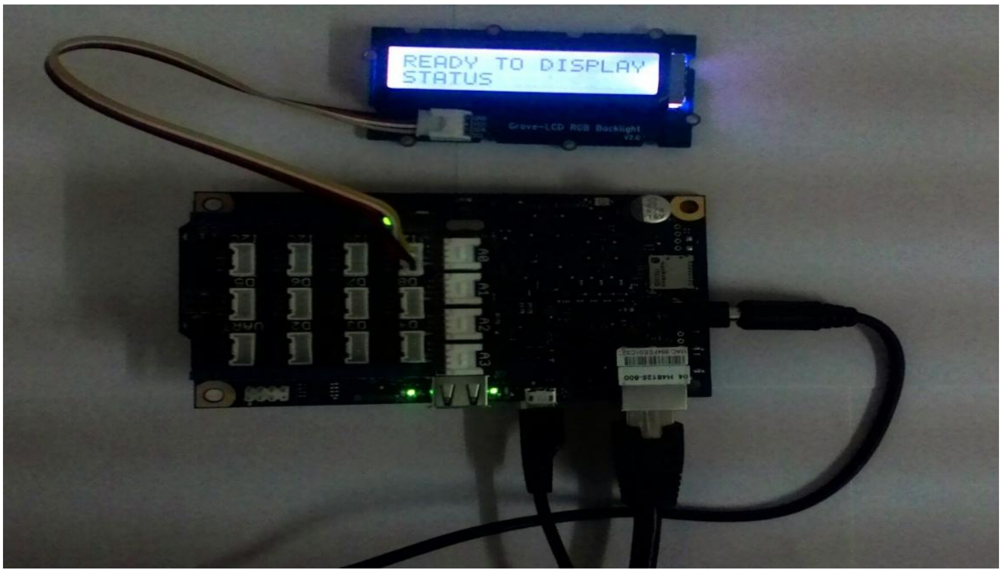
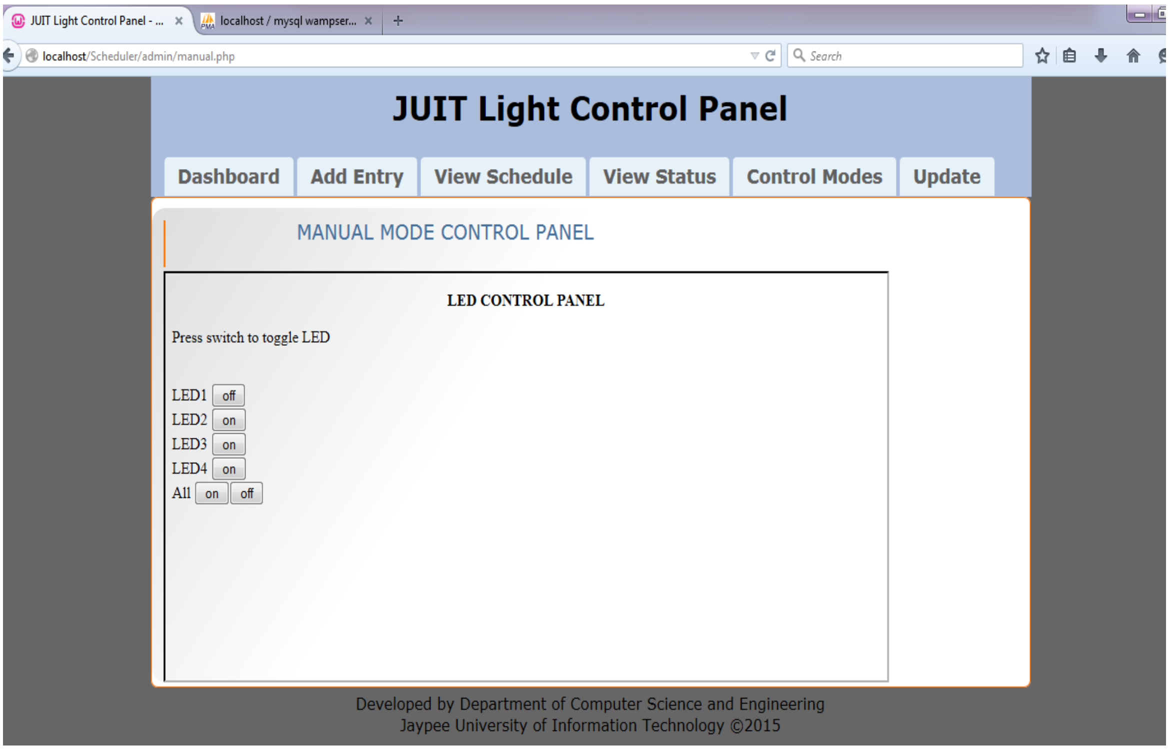


Figure 25: Circuit having LCD component connected to Server through Galileo

The Manual Mode button when pressed activates the LCD display through the server, which shows a "Ready" message.



JUIT Light Control Panel

MANUAL MODE CONTROL PANEL

LED CONTROL PANEL

Press switch to toggle LED

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 26: Web Portal for Control Panel – Manual Control Panel with LED1 ON

LED1 switch is pressed to toggle it ON. Correspondingly, the hardware unit switches ON the Led1 and the LCD displays the status message with the LED and room number.

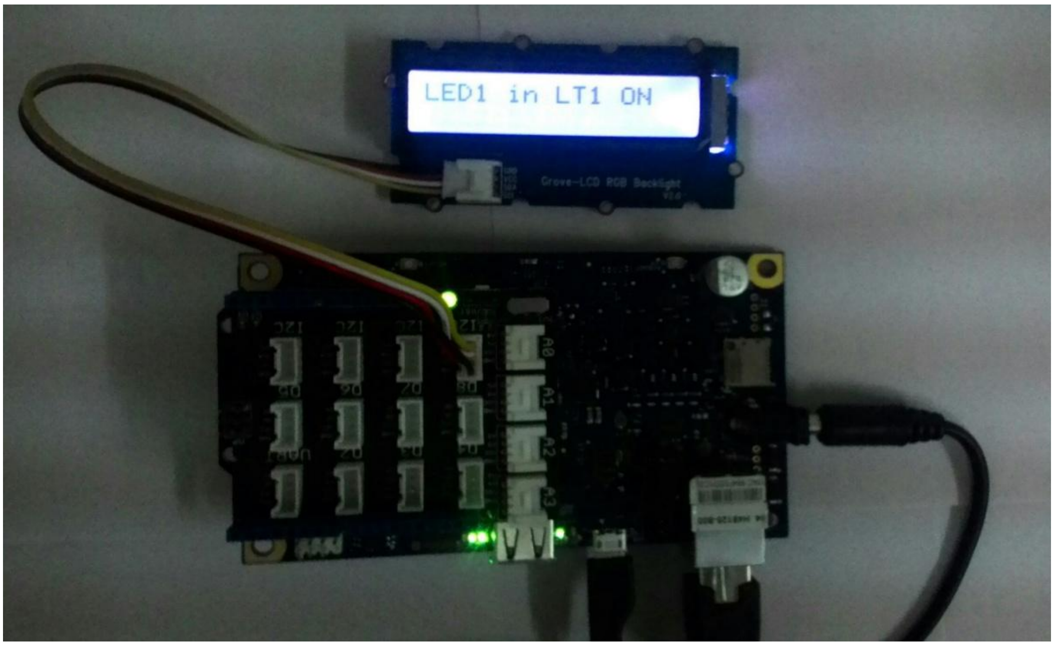
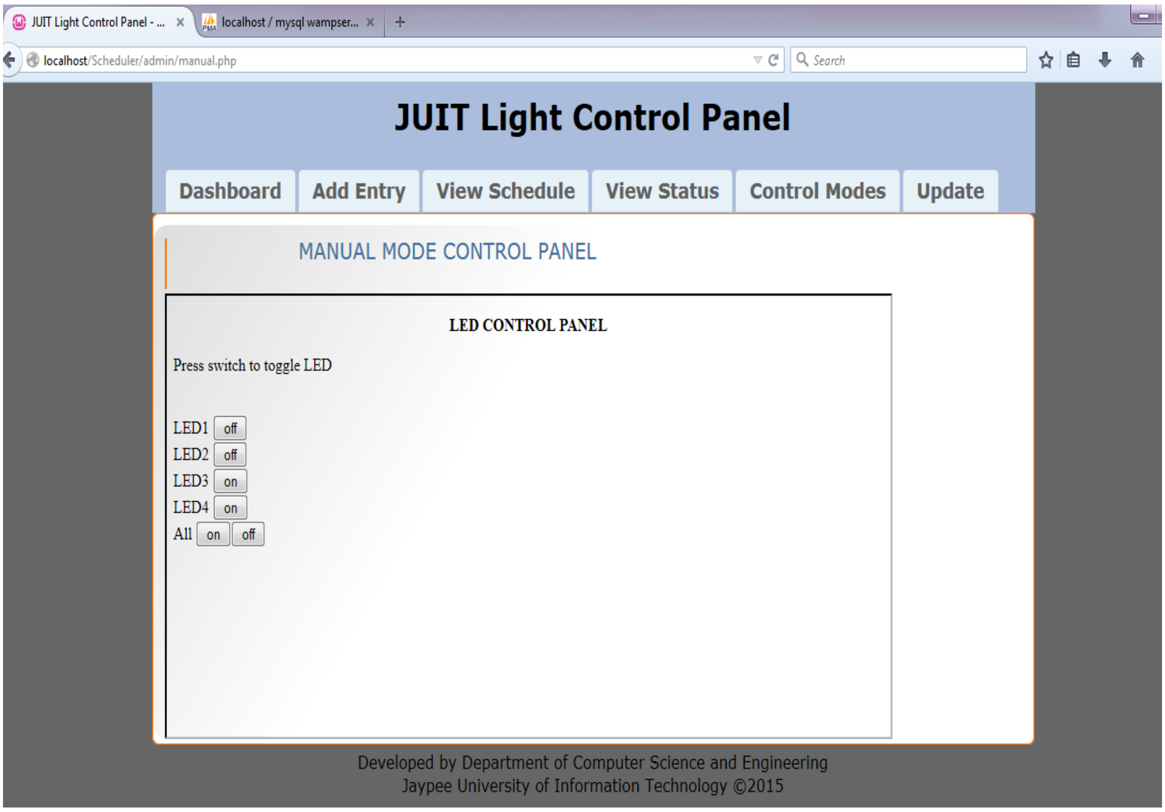


Figure 27: LCD displaying LED1 status when LED1 switch is pressed ON


JUIT Light Control Panel - ...
localhost / mysql wampser...
localhost/Scheduler/admin/manual.php
Search
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update

JUIT Light Control Panel

MANUAL MODE CONTROL PANEL

LED CONTROL PANEL

Press switch to toggle LED

LED1 off
LED2 off
LED3 on
LED4 on
All on off

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 28: Web Portal for Control Panel – Manual Control Panel with LED1 & LED2 ON

After LED1, LED2 switch is pressed to toggle it ON. Correspondingly, the hardware unit switches ON the Led2 and the LCD displays the status.

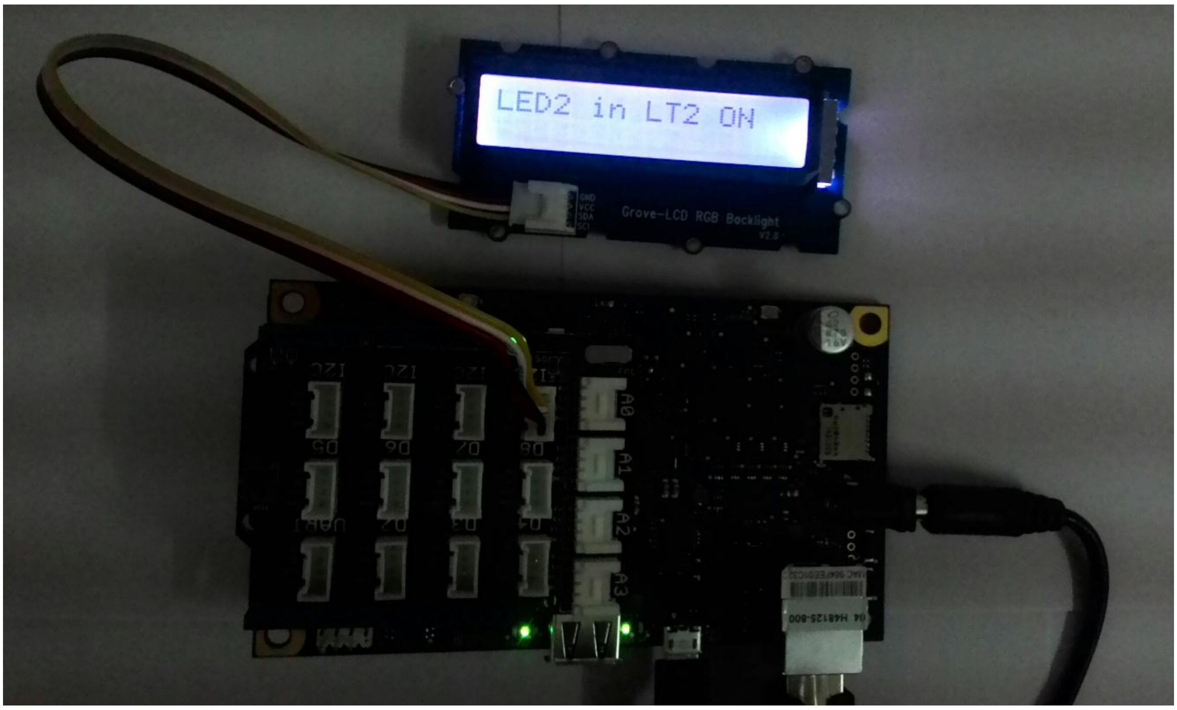
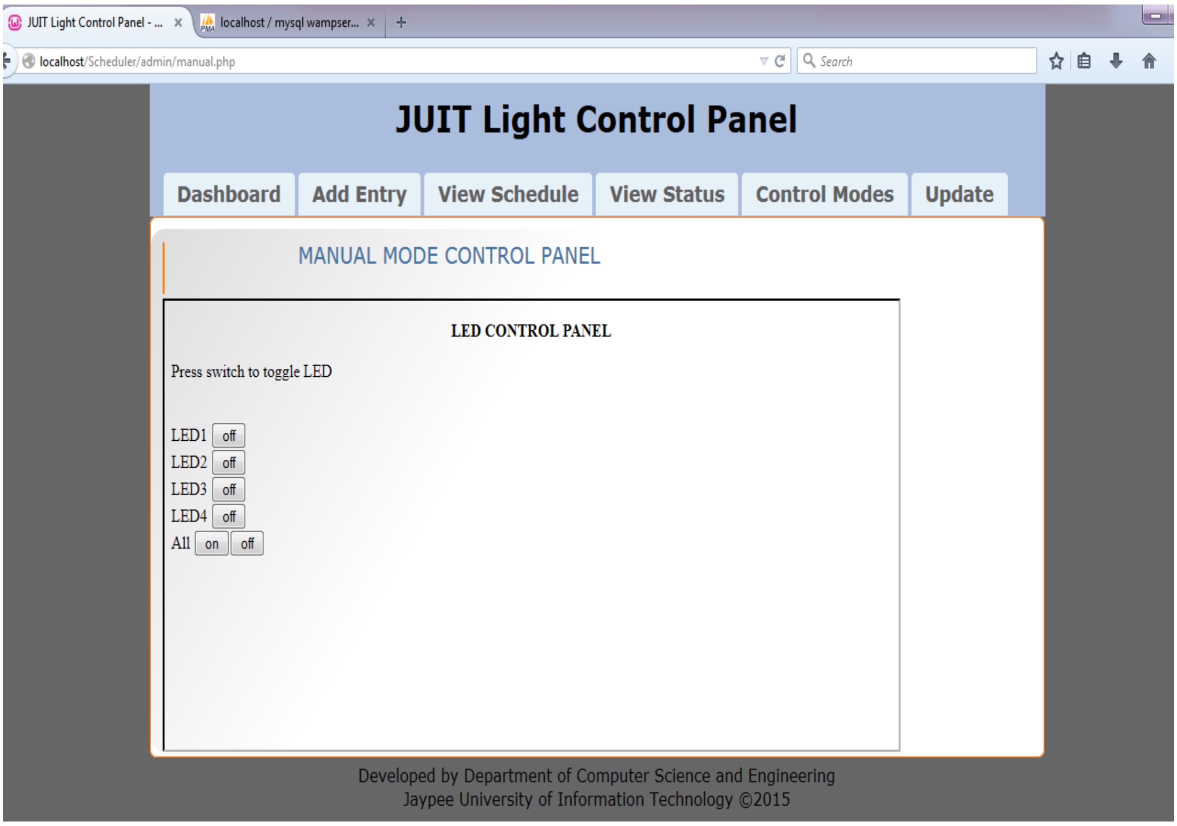


Figure 29: LCD displaying LED2 status when LED2 switch is pressed ON


JUIT Light Control Panel - ...
localhost / mysql wampser...
localhost/Scheduler/admin/manual.php
Search
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update

JUIT Light Control Panel

MANUAL MODE CONTROL PANEL

LED CONTROL PANEL

Press switch to toggle LED

LED1 off
LED2 off
LED3 off
LED4 off
All on off

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 30: Web Portal for Control Panel – Manual Control Panel with all LEDs ON

The ALL button is toggled ON by the admin. Correspondingly, the hardware unit switches ON all LEDs and the LCD displays the status message.

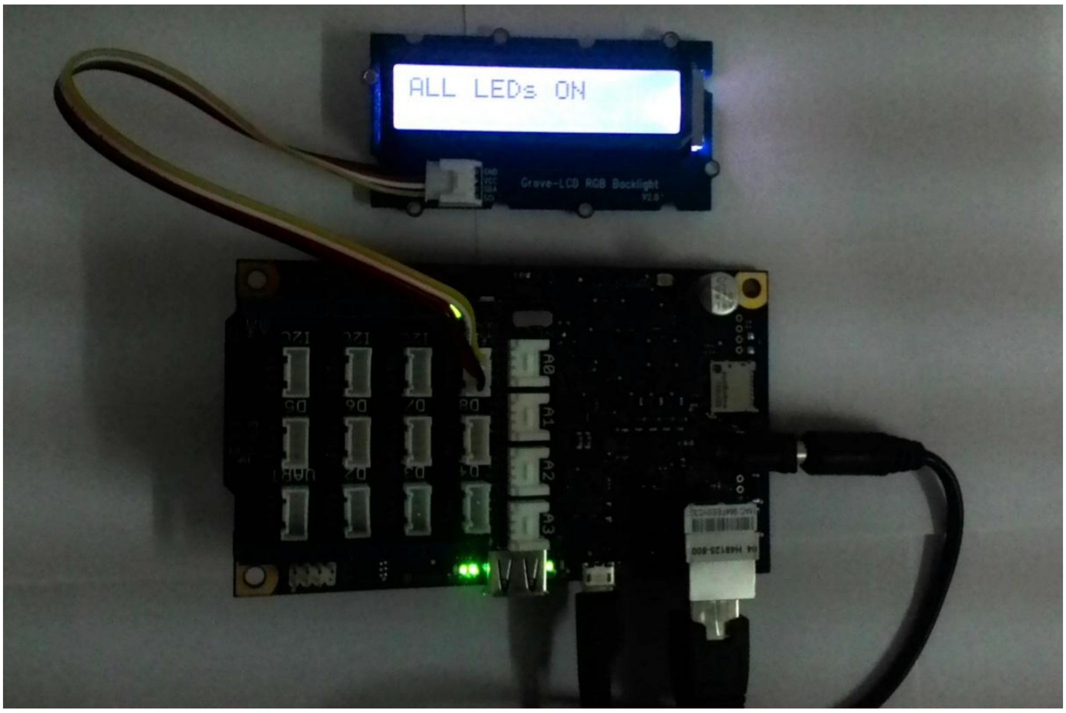
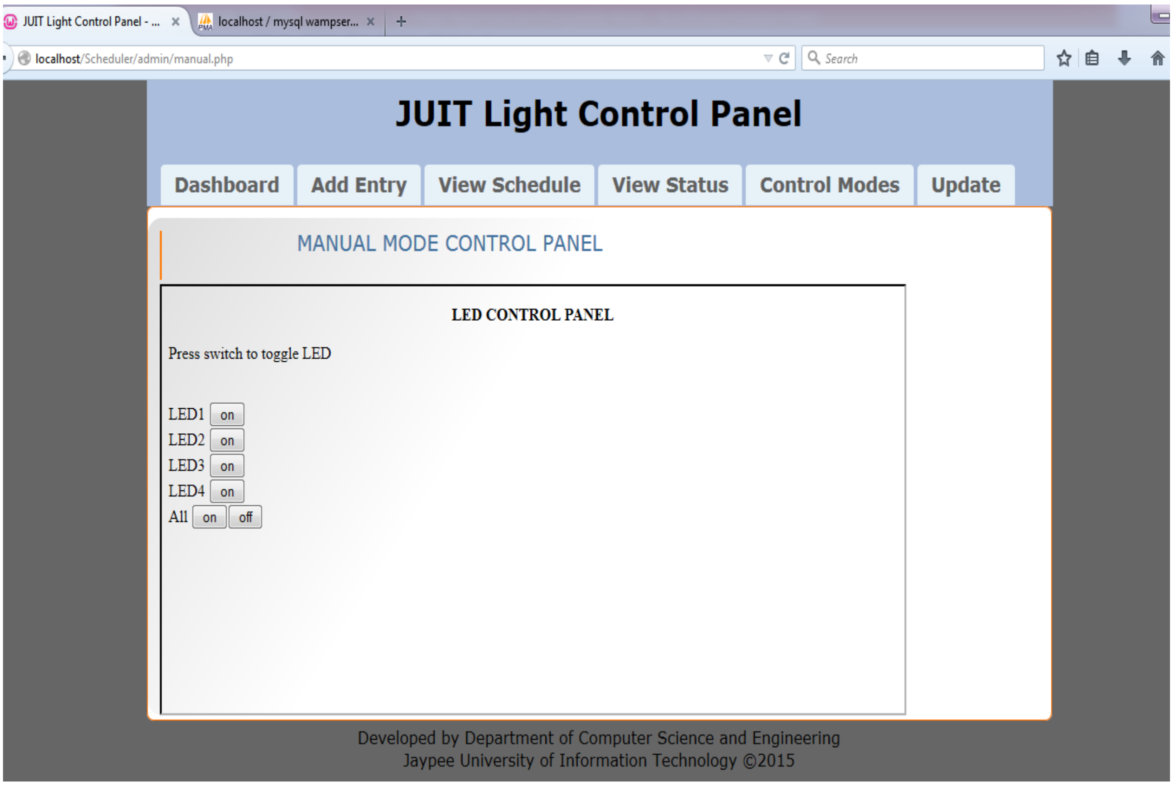


Figure 31: LCD displaying status when the ALL switch is pressed ON



JUIT Light Control Panel - ...
localhost / mysql wampser...
localhost/Scheduler/admin/manual.php
Search
Dashboard
Add Entry
View Schedule
View Status
Control Modes
Update
on
off

JUIT Light Control Panel

MANUAL MODE CONTROL PANEL

LED CONTROL PANEL

Press switch to toggle LED

LED1 on
LED2 on
LED3 on
LED4 on
All on off

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 32: Web Portal for Control Panel – Manual Control Panel with all LEDs OFF

Similarly, the ALL button is toggled OFF by the admin and the hardware unit switches ON all LEDs and the LCD displays the status message.

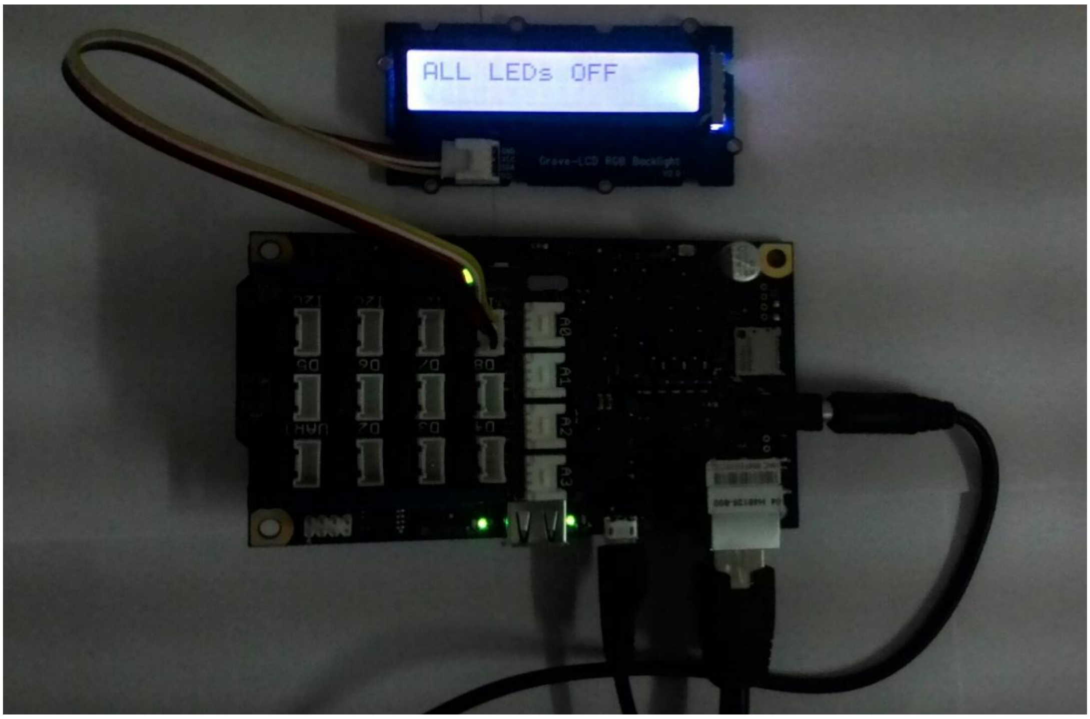
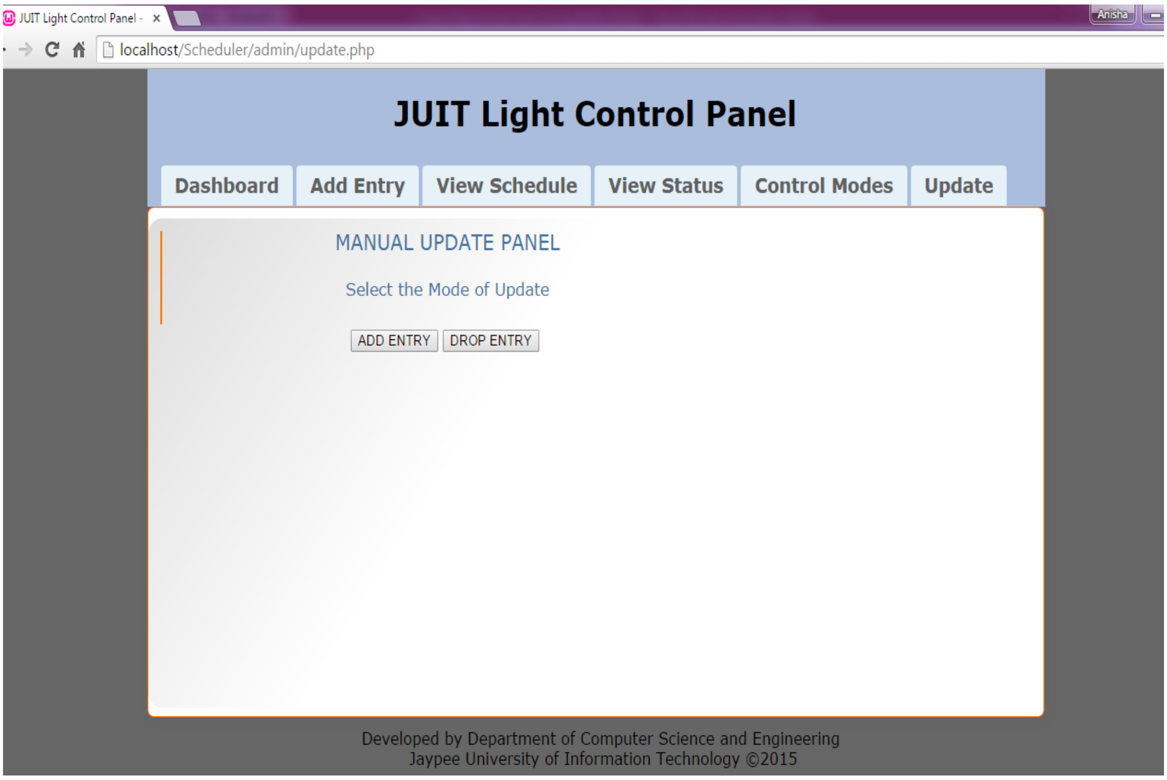


Figure 33: LCD displaying status when the ALL switch is pressed OFF

5.2.5 Functionality 5 : UPDATE SCHEDULE



MANUAL UPDATE PANEL

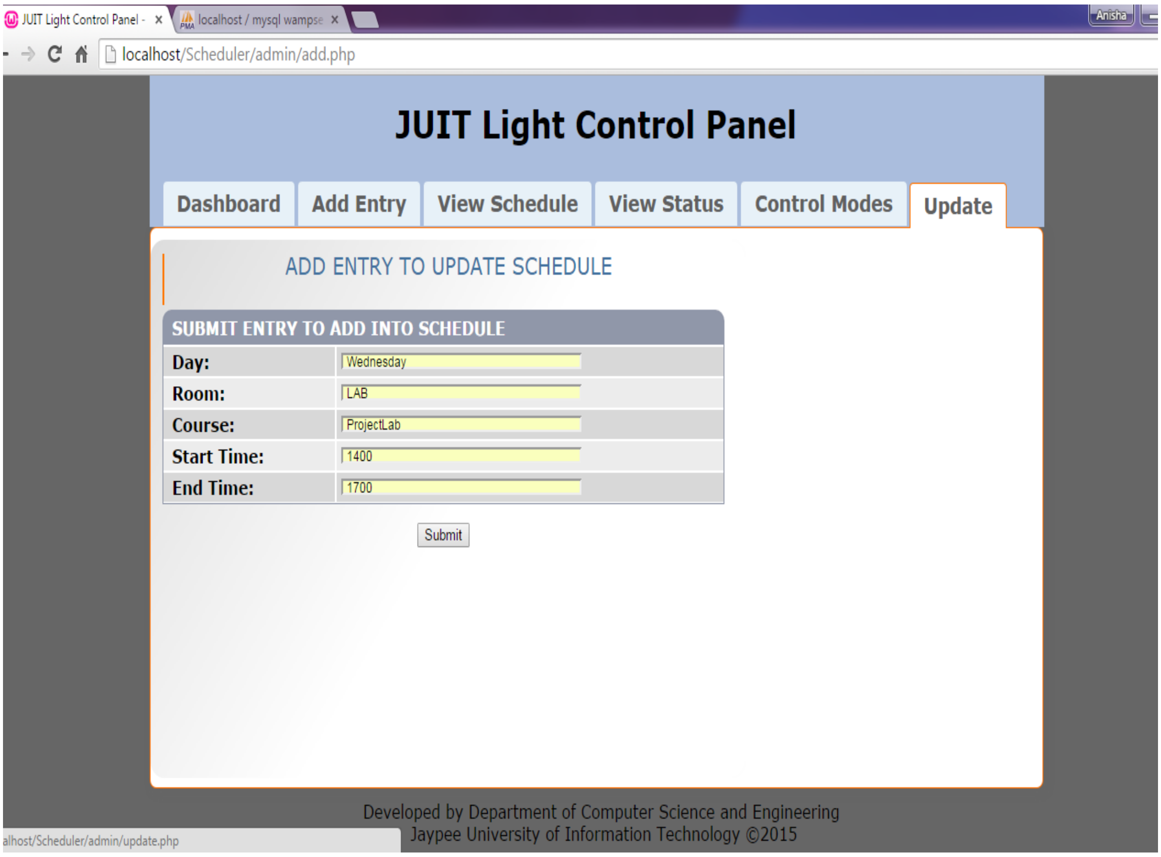
Select the Mode of Update

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

Figure 34: Web Portal for Control Panel – Manual Update Panel

This page provides two modes for updating the Schedule database, vis-à-vis ADD ENTRY and DROP ENTRY.

5.2.5.1 Functionality 5a: UPDATE SCHEDULE – ADD ENTRY



JUIT Light Control Panel

ADD ENTRY TO UPDATE SCHEDULE

SUBMIT ENTRY TO ADD INTO SCHEDULE

Day:	Wednesday
Room:	LAB
Course:	ProjectLab
Start Time:	1400
End Time:	1700

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology ©2015

alhost/Scheduler/admin/update.php

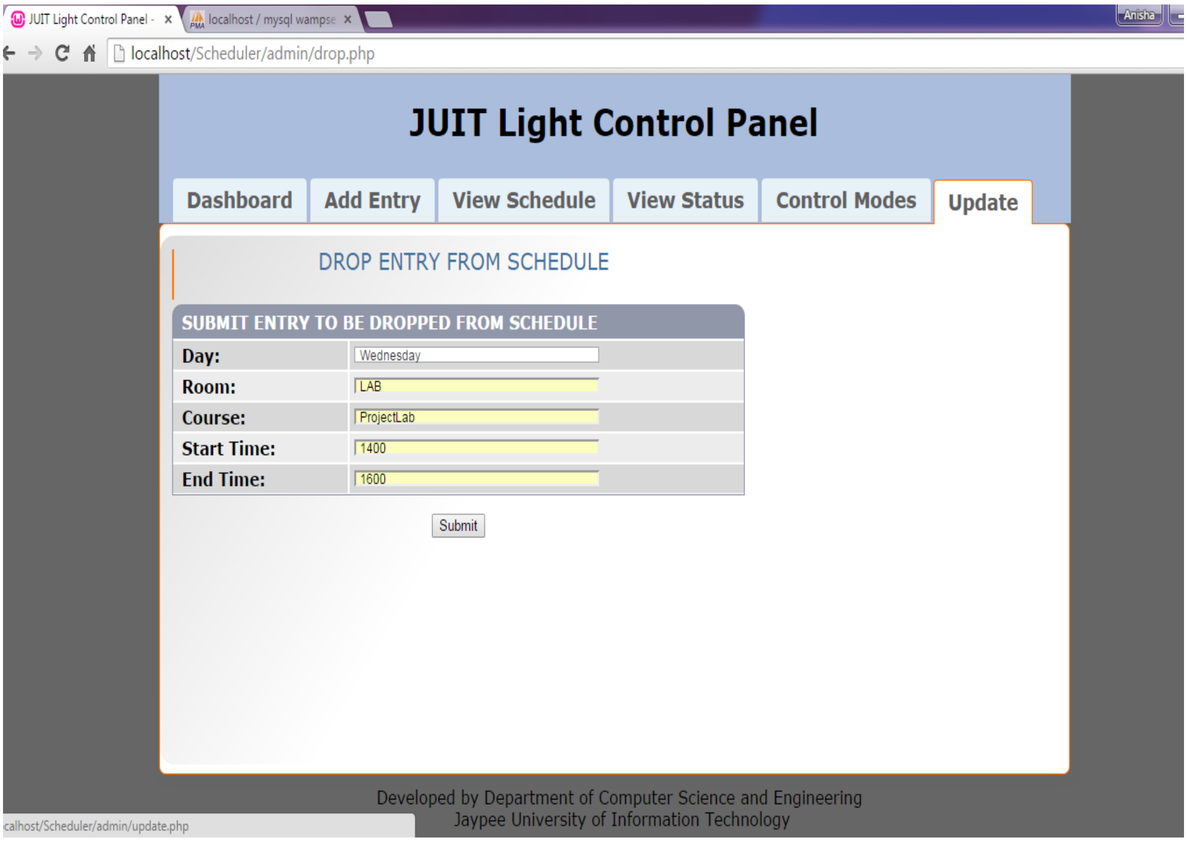
Figure 35: Web Portal for Control Panel – Manual Update Panel – Add Entry

The ADD ENTRY page provides a form to enter information and request to add a new entry into the schedule database, similar to creating a new entry as previously.

The Request Status parameter for each new request is initialized to 0, signifying that the request is in pending state. The server then checks for the validity of the request.

If valid, the requested entry is added to database, and Status is set to 1.

5.2.5.2 Functionality 5b: UPDATE SCHEDULE – DROP ENTRY



JUIT Light Control Panel

DROP ENTRY FROM SCHEDULE

SUBMIT ENTRY TO BE DROPPED FROM SCHEDULE

Day:	Wednesday
Room:	LAB
Course:	ProjectLab
Start Time:	1400
End Time:	1600

Developed by Department of Computer Science and Engineering
Jaypee University of Information Technology

Figure 36: Web Portal for Control Panel – Manual Update Panel – Drop Entry

The DROP ENTRY page provides a form to enter information of an existing entry and delete it from the schedule database. Again, the Request Status parameter is initialized to 0, i.e., pending. The server then looks for the requested entry in the database. If a match is found, it is deleted and Status is set to 1.

These two steps combine to perform an update in the schedule.

5.3 HARDWARE IMPLEMENTATION

5.1.1 **For Sketch 1 :** Blink.ino

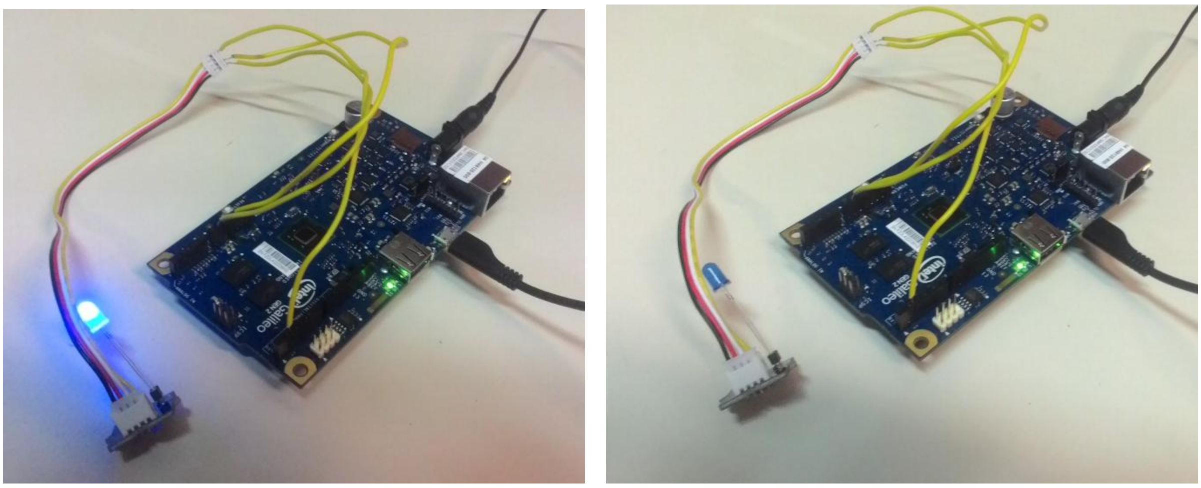


Figure 36: Blinking of LED at Pin 13 on Intel Galileo Gen2

5.3.1 **For Sketch3 :** multipleLED.ino

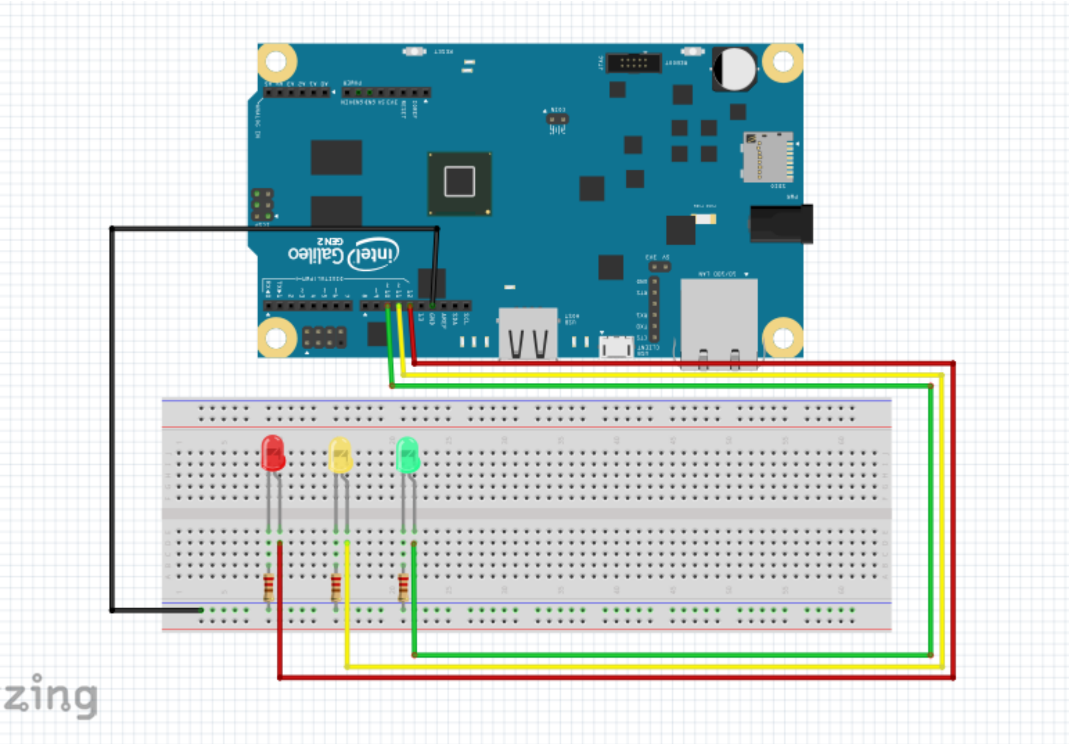


Figure 37: Schematic Circuit showing LEDs connected at Pin 10, 11 & 12 of Galileo Gen2, with anode of each LED connected to pins while negative of each is connected to GND

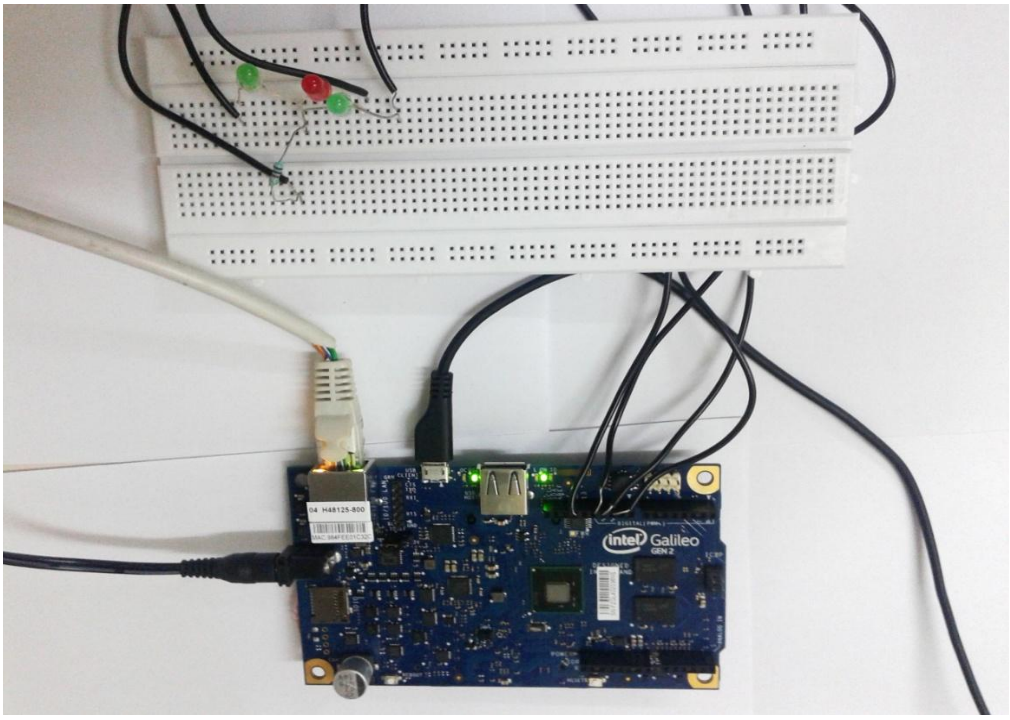


Figure 38: Circuit showing LED on Pin 10 toggled ON while LEDs on Pin 11 & 12 toggled OFF on Intel Galileo

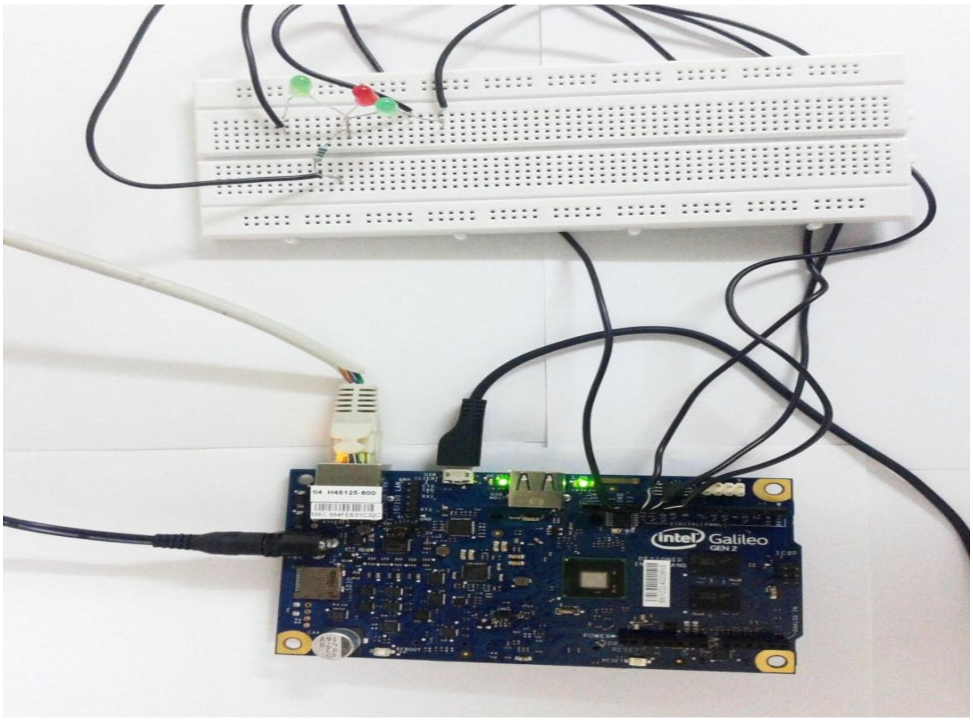


Figure 39: Circuit showing LEDs on Pin 10, 11 & 12 toggled ON

5.3.2 **For Sketch 4 :** relay.ino

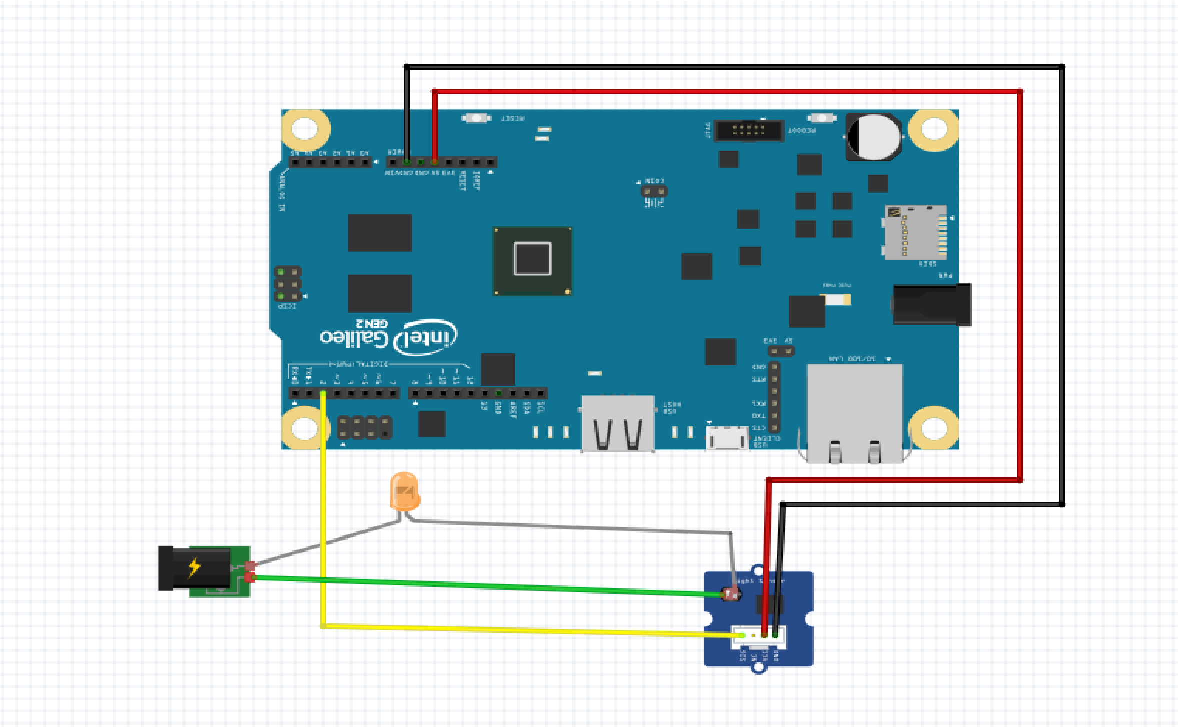


Figure 40: Schematic Circuit showing Relay connected to Galileo at PIN2 with VCC & GND connected at 5V Pin and GND Pin respectively. A bulb is connected to AC power supply at one end and Relay at the other.

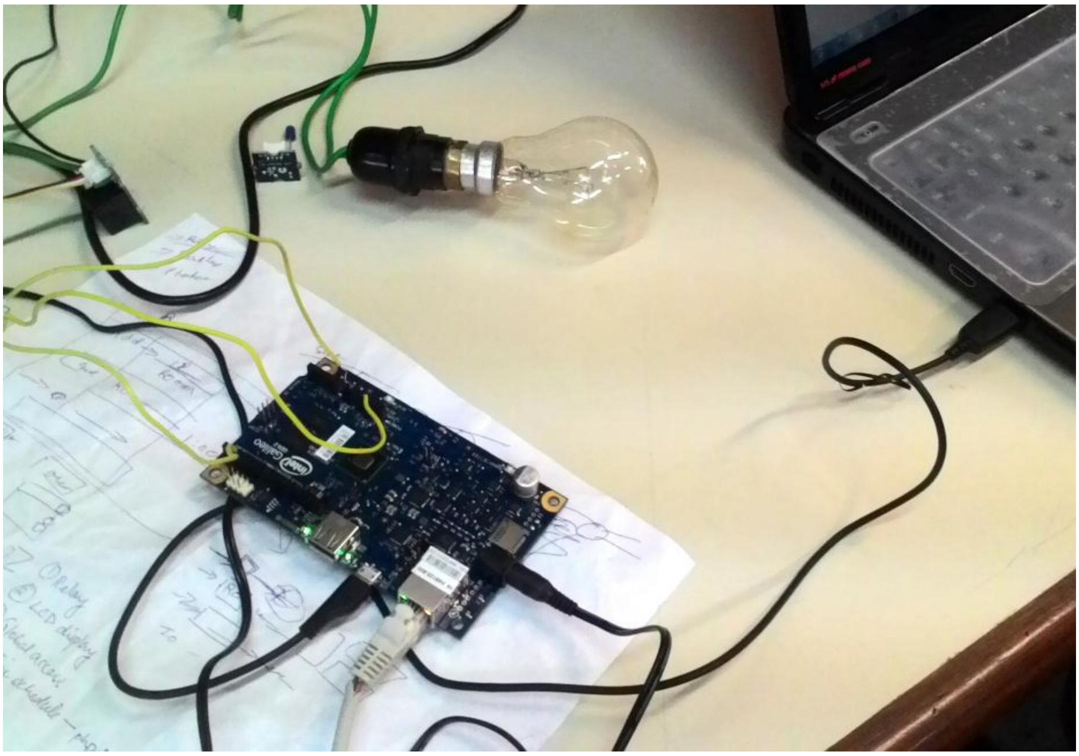
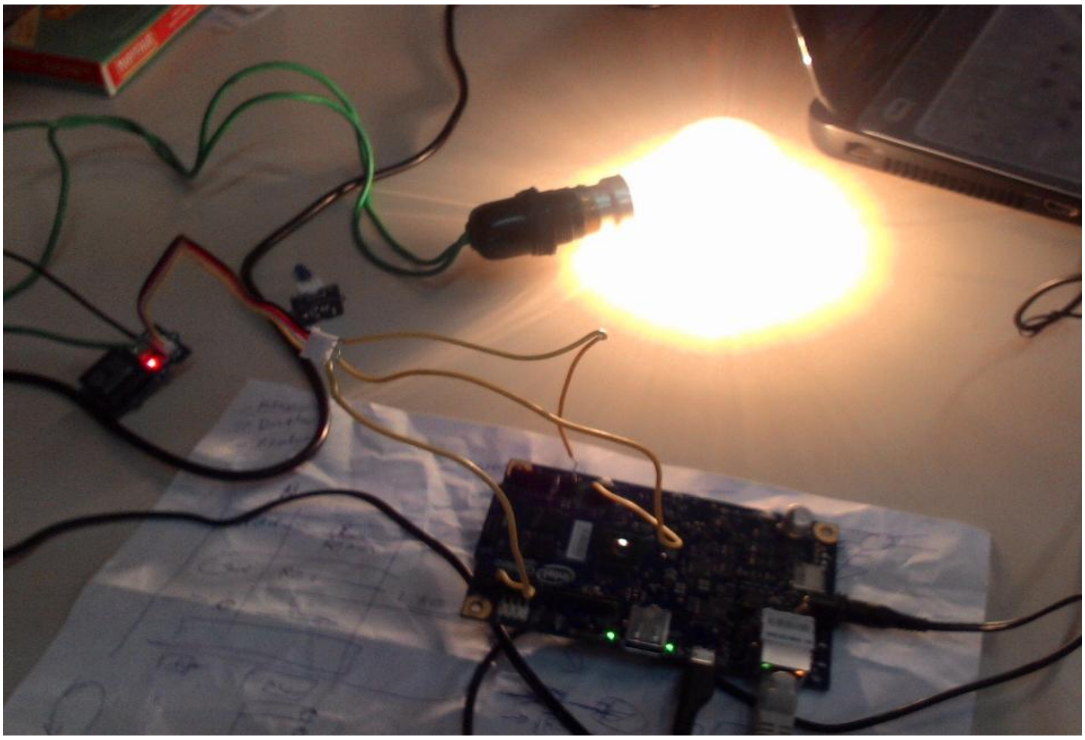


Figure 41: Relay connected to PIN 2 of Galileo controlling a bulb through AC supply to toggle it OFF & ON



5.3.3 For Sketch 5 : multiLED_LCD.ino

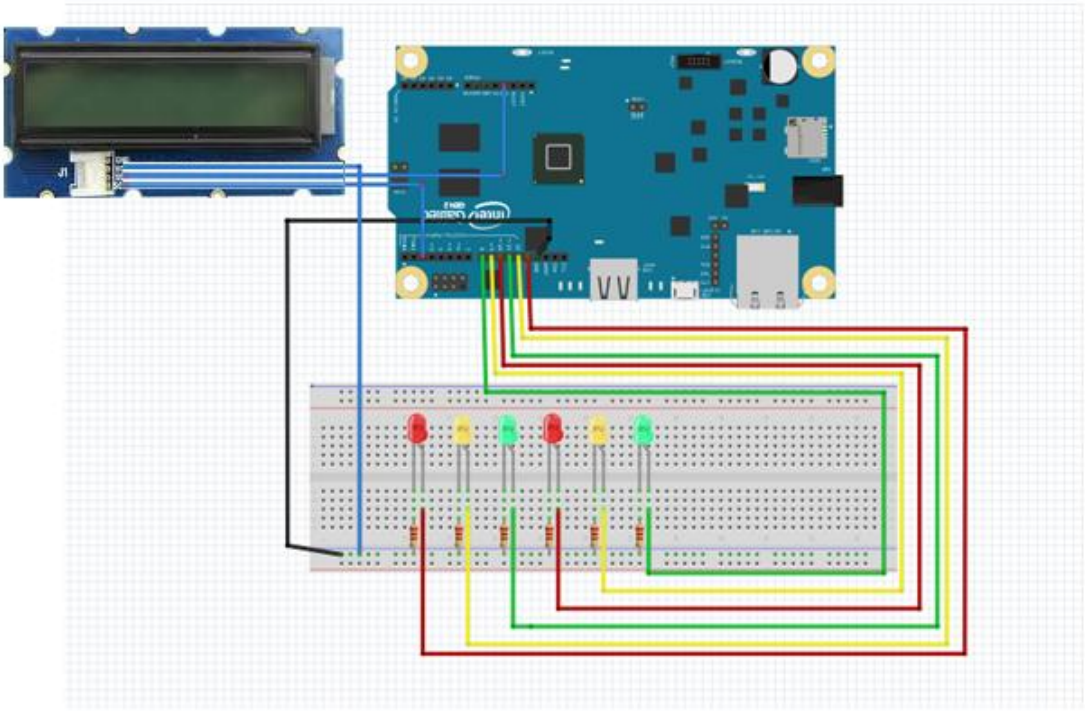


Figure 42: Schematic Circuit showing LCD (RGB Backlight LCD from Grove kit with shield connections) connected to Galileo at PIN3 with VCC & GND connected at 5V Pin and GND Pin respectively. 6 LEDs are connected to PINS 13, 12, 11, 10, 9 and 8 of Galileo.

5.3.4 For Sketch 6 : PIRsensor.ino

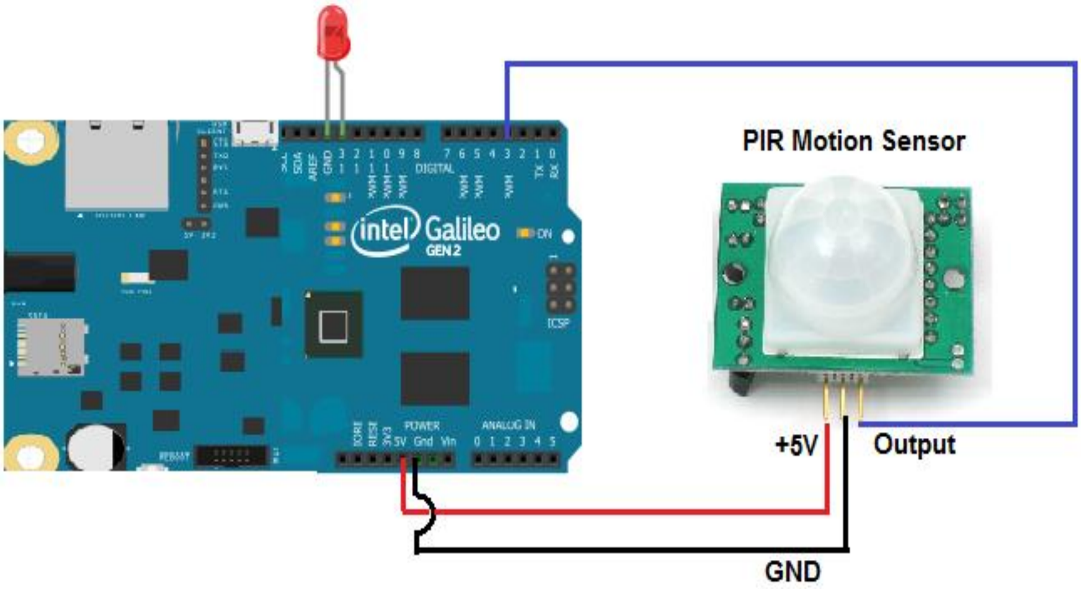


Figure 43: Schematic Circuit showing PIR sensor connected to Galileo with Output pin at Analog A3 & GND and VCC connected at GND Pin and 5V Pin respectively

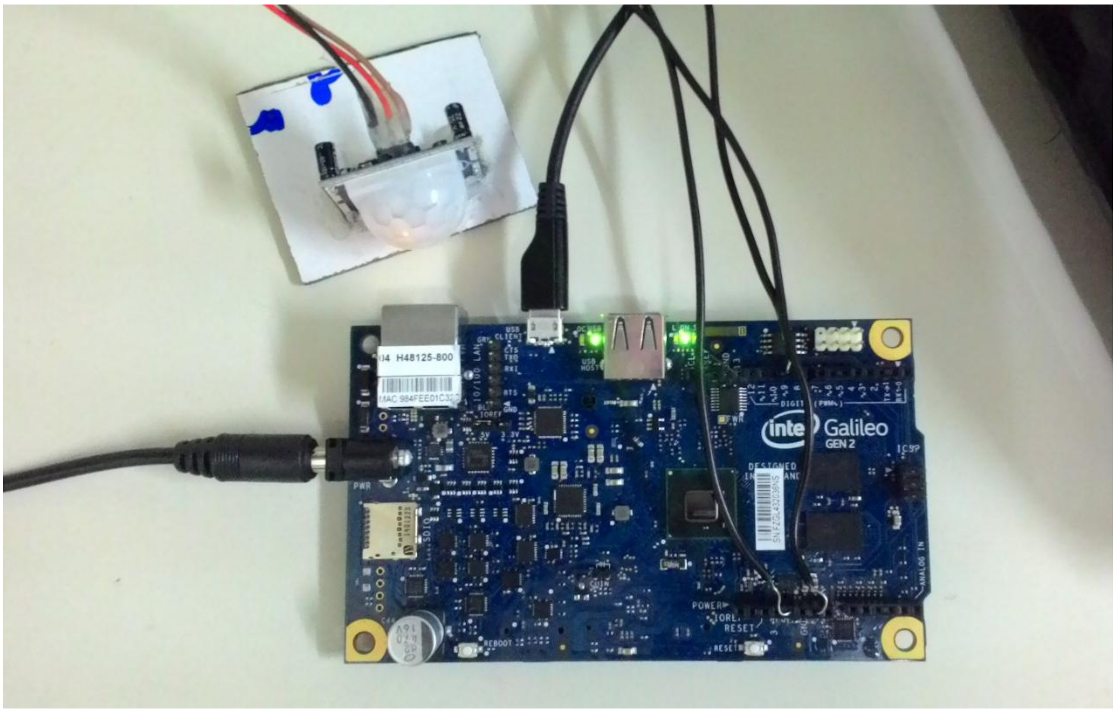


Figure 44: Circuit showing PIR sensor connected to Intel Galileo at PIN A0, GND and 5V PIN

CHAPTER 6

CONCLUSION AND FUTURE WORK

The proposed model to develop an intelligent automated system for conserving electrical energy that can be implemented in a university such as JUIT has been developed and both its phases have been completed, which involve the various modules of creating a class schedule at the server, developing the web application for control panel, controlling the lights in the classroom from the server using the control panel, and automating this process of light control. Further, motion detection sensors (PIR sensors) have been used to detect motion in & out of the classroom to achieve more efficiency in controlling lights.

The actual deployment in a larger organization might pose some ***challenges***, such as connectivity issues that may arise due to the limitations of the Ethernet, difficulty in regulating the schedule if deployed in a workplace scenario, etc., most of which can be overcome through further development on the system.

The proposed system prototype can be extended to be accessed through an Android application through smart phones, to view the current status or update the schedule in case of an emergency, but only through an authorized user.

It is also proposed to work on another criteria of energy saving, which involves real-time sensing of the external environment, like weather, temperature and amount of sunlight entering the room, to decide the required amount of light actually needed and thus control the number of lights needed or the requires voltage.

Furthermore, the system can be extended to work not only on Ethernet, but through the wireless connectivity of the organisation.

CHAPTER 7

REFERENCES

[1] Alkar, A.Z., Hacettepe Univ; Roach, J. ; Baysal, D., "IP based home automation system", *Consumer Electronics, IEEE Transactions* on (Volume:56 ,Issue: 4), November 2010, IEEE

[2] Golzar, M.G. ; AsanPardazan Co. ; Tajozzakerin, H.R., "A New Intelligent Remote Control System for Home Automation and Reduce Energy Consumption", *Mathematical/Analytical Modelling and Computer Simulation* (AMS), 2010, IEEE

[3] Sandeep Patel, Punit Gupta, Mayank Kumar Goyal, "Low Cost Hardware Design of a Web Server for Home Automation Systems", *Conference on Advances in Communication and Control Systems(*CAC2S), 2013

[4] Arduino Software, http://arduino.cc/

[5] “Artificial Intelligence - Foundations of Computational Agents”, http://artint.info/

[6] “What is Automation”, The Automation Federation, http://www.automationfederation.org

[7] “Defining the Internet of Things”, Enterprise Networking Planet, http://www.enterprisenetworkingplanet.com/datacenter

[8] “Introducing the Intel Galileo Development Board”, Intel, http://www.intel.in/content/

[9] “Synco Living Home Automation System”, http://www.buildingtechnologies.siemens.com/synco_living

[10] ZHOME Home Automation India, http:///www.zhome.co.in/

[11] Z-Wave Home Control, http://www.z-wave.com/

[12] Grove starter kit plus – Intel IoT Edition, http://www.seeedstudio.com/depot/Grove-starter-kit-plus-Intel-IoT-Edition-for-Intel-Galileo-Gen-2-and-Edison-p-1978.html; http://www.seeedstudio.com/wiki/Grove_-_Starter_Kit_Plus

[13] Fritzing – About Us – Context, http://fritzing.org/about/context/