Brunel University

School of Engineering and Design

Final year project

Smart system for control and customisation of office building energy consumption

Author:
Dionysios Kalantzis

Supervisor:
Dr. M. Abbot

April 6, 2012

Abstract

"Apollo the god who sees and foresees everything."

Acknowledgements

Thanks Mum!

Contents

1 Introduction **5**

1.1 Project Aims and Objectives 5

2 Literature review **7**

2.1 Energy efficiency 7

2.2 Ventilation and air quality standards 7

2.3 Internet of Things 8

2.4 Wireless Sensor Network 8

3 System Architecture **10**

4 System Specifications **12**

4.1 The sensor board 12

4.1.1 MQ - 4 Methane (Compressed Natural Gas) sensor 13

4.1.2 MQ - 7 Carbon Monoxide sensor 13

4.1.3 LDR - Light Dependent Resistor 14

4.1.4 PIR - Passive InfraRed motion sensor 14

4.1.5 LM335 - Precision Temperature Sensor 15

4.2 The control board 15

4.2.1 Arduino Pro Mini 16

4.2.2 XBee RF Module 17

4.3 The action board 18

4.3.1 Relay SPDT Sealed 18

4.4 Software 19

4.4.1 Arduino IDE 19

4.4.2 CoolTerm 20

4.4.3 Eagle PCB Software 21

5 System Design **22**

5.1 Sensor board 23

5.1.1 Schematic implementation 23

5.1.2 Board implementation 26
5.2 Control board 27
5.3 Action board 30
5.4 Arduino code 31
5.4.1 Initialisation 32
5.4.2 setup() 32
5.4.3 loop() 32
5.5 Database 33

6 Results **34**
6.1 PIR 36
6.2 Light 37
6.3 Temperature 38
6.4 CH_4 38
6.5 CO 39

7 Conclusions & Future improvements **40**

List of Figures

3.1 Overall system design - Block diagram 10

4.1 MQ - 4 Methane - Compressed Natural Gas sensor 13
4.2 MQ - 4 sensor's drive circuit 13
4.3 MQ - 7 Carbon Monoxide sensor 14
4.4 MQ - 7 Dimensions 14
4.5 LDR - Light Dependent Resistor 14
4.6 PIR - Passive InfraRed motion sensor 15
4.7 LM335 - Precision Temperature Sensor 15
4.8 Arduino Pro Mini 16
4.9 XBee RF module 17
4.10 Relay SPDT Sealed 18
4.11 Led 19
4.12 Arduino IDE software development environment 19
4.13 CoolTerm - serial port communication 21
4.14 Eagle - Schematic design 21
4.15 Eagle - Board design 21

5.1 Sensor board - Adding all the parts (Stage 1) 23
5.2 Sensor board - (Stage 2) 24
5.3 Sensor board - (Stage 3) 24
5.4 Sensor board - (Stage 4) 25
5.5 Sensor board - (Stage 5) 25
5.6 Sensor board - (Stage 6) 26
5.7 Sensor board - (Stage 7) 26
5.8 Sensor board - (Stage 8) 27
5.9 Control board - (Stage 1) 28
5.10 Control board - (Stage 2) 28
5.11 Control board - (Stage 3) 29
5.12 Control board - (Stage 4) 29
5.13 Arduino header 30
5.14 Action board - (Stage 1) 30

5.15 Action board - (Stage 2) 31
5.16 Action board - (Stage 3) 31

6.1 Sensor board - Completed with all parts soldered 34
6.2 Control board - Completed with all parts soldered 35
6.3 Action board - Completed with all parts soldered 35
6.4 *Apollo* - All boards are connected 36
6.5 *Überdust* chart for for PIR sensor in room 0.I.1 36
6.6 *Überdust* chart for for PIR sensor in room 0.I.3 37
6.7 *Überdust* chart for for PIR sensor in room 0.I.9 37
6.8 *Überdust* chart for LDR sensor in room 0.I.11 37
6.9 *Überdust* chart for LDR sensor in room 0.I.1 37
6.10 *Überdust* chart for LDR sensor in room 0.I.3 37
6.11 *Überdust* chart for LDR sensor in room 0.I.9 37
6.12 *Überdust* chart for LDR sensor in room 0.I.11 37
6.13 *Überdust* chart for Temperature sensor in room 0.I.11 38
6.14 *Überdust* chart for Temperature sensor in room 0.I.11 38
6.15 *Überdust* chart for Temperature sensor in room 0.I.11 38
6.16 *Überdust* chart for Temperature sensor in room 0.I.11 38
6.17 *Überdust* chart for PIR sensor in room 0.I.11 38
6.18 *Überdust* chart for PIR sensor in room 0.I.11 38
6.19 *Überdust* chart for PIR sensor in room 0.I.11 39
6.20 *Überdust* chart for PIR sensor in room 0.I.11 39

Chapter 1

Introduction

Air is one of life's essentials for all creatures on Earth. Humans live, work and prosper in air. The quality of air affects humans performance. Throughout the years various methods have been devised to assure the good air quality. For example miners were using canary birds as warning systems until 1987. The presence of toxic gases such as carbon monoxide, methane or carbon dioxide could have deadly consequences. Therefore the notice of any distress from the canary birds would act as a warning. Moreover, high air quality standards improve the performance of the people in a working environment.

Nowadays another main issue, in the fast growing world, is power consumption, and how to minimise it. More and more devices are being developed and are becoming available to consumers. Regardless if those devices are mechanical or electrical, they need energy to operate.

This project presents a system devised to improve the air quality in office buildings and other closed working environments. It can also minimise the energy consumption by using the building's resources only whenever they are needed. This is achieved by constantly monitoring the environmental conditions in a room and operating the lighting and air-conditioning according to the presence and the preferences of the occupants.

1.1 Project Aims and Objectives

This projects aims to build a system that monitors the environmental conditions of a room as well as operating the lighting and/or air-conditioning of the room,

according to the user(s) preferences.

Chapter 2

Literature review

2.1 Energy efficiency

A lot of attention has been focused on the energy use of new buildings through regulations and various standards. These just address only part of the problem as almost 70% of the buildings which will be needed by 2050 have already been built. This becomes even more dramatic as such buildings have been constructed before 1985, the year during which the energy efficiency requirements in the Building Regulations were first introduced [6, ?].

Modifyin an existing building can be very expensive and extremely inconvenient, in particular if it is currently being used. Therefore it seems only reasonable to invest into new technologies that could reduce the energy consumption without altering the construction. *Apollo* is a plug and play 3 piece system which can control the energy use of the room it is installed in.

2.2 Ventilation and air quality standards

Clean air is very important for a working environment. Productivity may be reduced by up to 20%, as studies has shown. The pollution, which human and machines produces can make the air feel heavy and humid. This is due to the fact that both continuously produce vapours and heat. The result of operating in this environment could be tiredness and discomfort from the increased $CO2$ and temperature levels. Studies has shown that in order to maintain a good indoor climate in an office, the air should be replaced at least three times per hour. In many rooms, such as laboratories, cooling is also essential if just the ventilation

is not enough [1].

2.3 Internet of Things

The *Internet of Things* (IoT) is the concept of globally interconnected devices – *things* with RFID *(Radio-Frequency IDentification)* technology with virtual representation in an Internet-like structure. With the rapid technology developments today, the interconnection of the *things* is feasible between heterogeneous devices. Once connected to the Internet these devices can report in real time from anywhere on the globe. Moreover due to their uniqueness, it is easy to map where the data are coming from and most importantly what they are reporting[14].

IoT vision is not limited to just reporting. As they would be part of global infrastructure, more and more smart processes and services could be available. Those processes and services could have environmental and, ultimately, an economic impact. *Apollo* could clearly be included into IoT, as it monitors environmental changes and by acting accordingly energy is saved effectively contributing to cost reduction in terms of annual consumption [5].

An example of such IoT project is Pachube.

> Pachube ("patch-bay") connects people to devices, applications, and the Internet of Things. As a web-based service built to manage the world's real-time data, Pachube gives people the power to share, collaborate, and make use of information generated from the world around them [9].

Moreover *Pachube* is a platform where sensors report in real time and their data are accessible anywhere on the world. This is due to the fact that *Pachube* traffics its data over the Internet.

2.4 Wireless Sensor Network

Wireless sensor network (*WSN*) is a concept that makes IoT vision be more and more real. The first appearance of the term dates back to the 1990s.

> Researcher named Kris Pister dreamed up a wild future in which people would sprinkle the Earth with countless tiny sensors, no larger than grains of rice.

These particles were called *"smart-dust"* [13]. WSN is a network consisting of distributed devices that provide sensing features. Lately systems applied on WSN are becoming more and more noticeable. Arduino, which is one of the most common hardware platforms, would make the perfect wireless sensor network node, due to its small size (Arduino Pro Mini), low cost and modularity.

Uberdust holds a somewhat executive role to IoT as well as to WSN. Uberdust is one of the promising outcomes of the research program **SPITFIRE** (*Semantic Service Provisioning for the Internet of Things using Future Internet Research by Experimentation*). SPITFIRE's main aim is to investigate unified concepts methods and software infrastructures that facilitate the efficient development of applications that span and intergrate the Internet and the embedded world. Uberdust is a data storage and brokerage web service for connecting IoT data providers and actuators with data consumers and application developers providing real time as well as historical data.It's main focus is to provide storage, sharing and discovery of real-time data from objects, devices and building installations around the world via the Web.

Additionally a unique experiment research facility is currently being proposed by *Smart Santander*. A city-scale experiment for applications and services of a smart city. *Apollo* could easily be adapted into this experiment, providing information on the energy consumption of the office buildings and not only. This way, the city's resources use would be optimised, achieving an greater energy efficiency.

Chapter 3

System Architecture

Apollo is an automatic system which detects the environmental conditions in the office. As a smart system it reacts, accordingly to the preferences of the users in the office, to achieve the optimum environment to maximise the work performance.

In Figure 3.1 below there is a visual representation of *Apollo*. The communication between the board is shown with the use of the arrows.

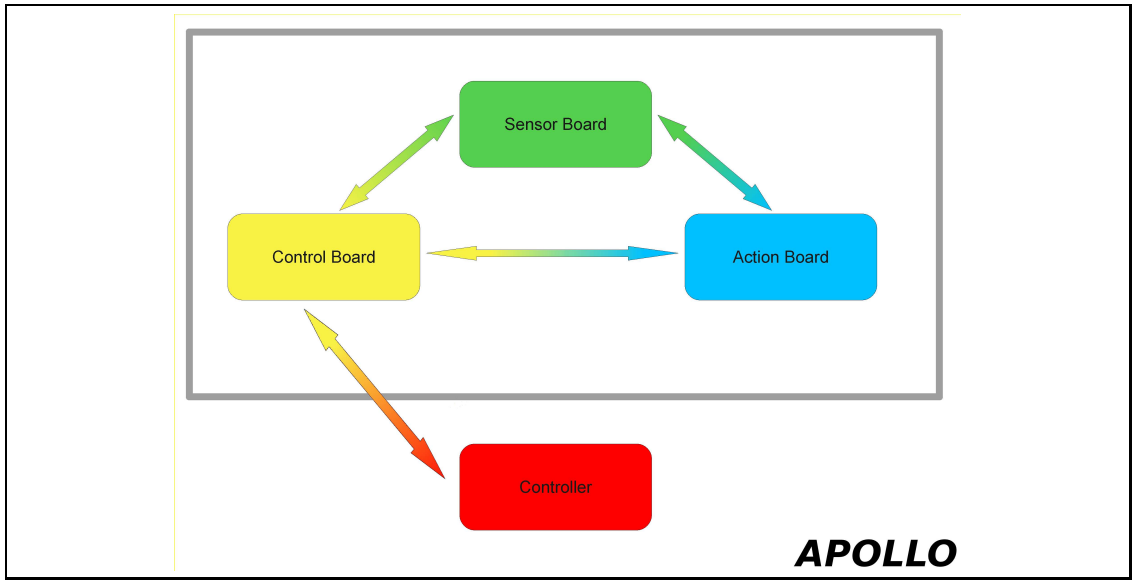

Figure 3.1: Overall system design - Block diagram

The *Controller* is a SunSPOT. SunSPOT (Sun Small Programmable Object Technology) is a wireless sensor network mote developed by Sun Microsystems [15]. Due to is computational power it is preferred over the Arduino to make the decisions on the operation of the lights and/or the air-conditioning.

The system works in two stages – the *passive* process and the *active* process. In order for the system to work on the later stage it is necessary to go through the *passive*. The difference between these two stages is, during the former (*passive*) stage, the system collects the user(s) preferences on the environmental changes. It also observes the environmental values when the user(s) choose to switch on the lights and/or the air conditioning.

Those marks will set the threshold on when the system will operate on the *active* stage. During this stage (*active*) the system operates the lights and the air conditioning. And this is done with the preferences the users set on the *passive* stage.

As the weather conditions change over the year it is advised to use the system on the *passive* stage approximately for one to two weeks every four months. This is to calibrate the system with reference to the four different seasons (*spring, summer, autumn, winter*) over the year. In general this is needed to be done only once after the installation of the system. From then onwards the *passive* stage is needed to be run only when the users feel the need to make any core modifications on the thresholds.

Regarding the modifications, user(s) are also able to use the web interface of the system to manually control the lighting and/or air conditioning. The web interface is a medium where user(s) can remotely access and interact with the system. This could mean operating manually the lighting and/or air conditioning remotely. The user(s) are also capable to check the status of the system via the web interface.

The status of the system is generally used for security reasons as *Apollo* can inform the owner of the office of any unusual changes in environmental conditions. This is useful for example to detect fires. Moreover every time the system recognise an unauthorised person using the office can remotely inform its owner through the internet with an email.

Chapter 4

System Specifications

Apollo is a three (3) piece system:

- The sensor board
- The control board
- The action board

The main reason for the existence of the 3 different pieces is to minimise the mean time of repair. Each board can easily removed and individually checked for defect(s).

4.1 The sensor board

The sensor board is a PCB(*Printed Circuit Board*) responsible for collecting the environmental changes of the space it is installed in. To achieve the above, the board is equipped with the following:

- Gas sensors:
  - MQ - 4 Methane - *Compressed Natural Gas* sensor
  - MQ - 7 Carbon Monoxide sensor
- Light sensor:
  - LDR - Light Dependent Resistor
- Motion Sensor:
  - PIR - Passive InfraRed motion sensor
- Temperature Sensor:
  - LM335 - Precision Temperature Sensor

4.1.1 MQ - 4 Methane (Compressed Natural Gas) sensor

This is an easy to use *compressed natural gas* sensor capable of sensing natural gas, which is mainly composed of methane (CH_4). The detection range for this sensor is between 200 to 10000ppm of methane - compressed natural gas [11].

Figure 4.1: MQ - 4 Methane - Compressed Natural Gas sensor

From the following diagram it is easy to see the simplicity of sensor's drive circuit.

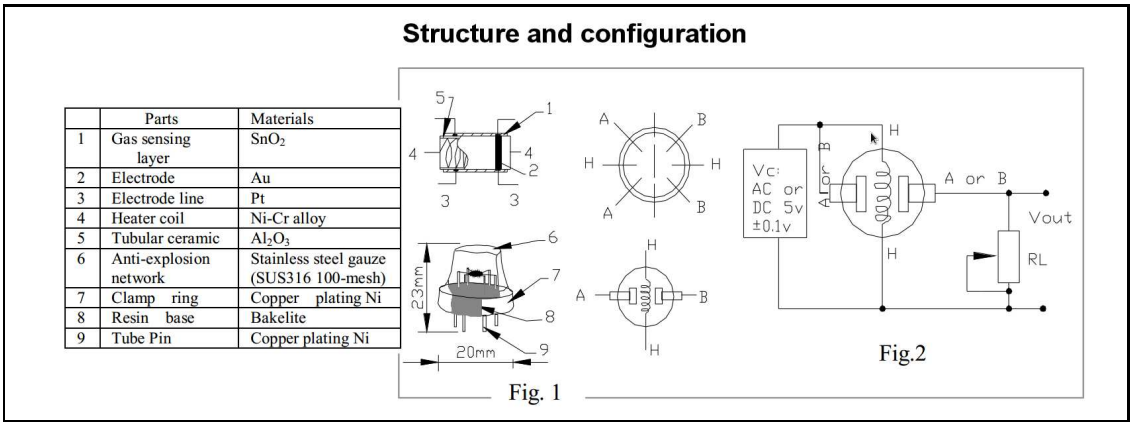

Figure 4.2: MQ - 4 sensor's drive circuit

The sensor requires an input of 5 Volt and its output is an analogue resistance. The output is inversely proportional to the gas concentration. Moreover the greater the gas concentration the higher the output value.

4.1.2 MQ - 7 Carbon Monoxide sensor

The *Carbon Monoxide* sensor is as well an easy to use sensor. The detection range is between 20 to 2000ppm of CO [10].

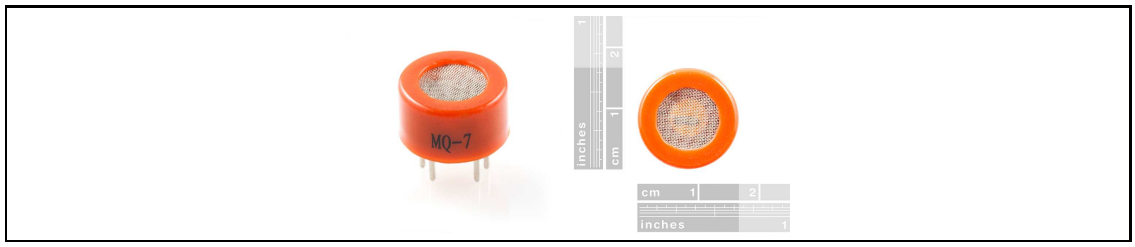

Figure 4.4: MQ - 7 Dimensions

The MQ - 7 has a slightly more complicated operation technique comparing to the MQ - 4. It detects by cycle high (5 Volts) and low (1.5 Volts) temperature and detects CO when low temperature is applied. When sensor is at high temperature (heated by 5 Volts) cleans the other gases absorbed under the low temperature heating (1.5 Volts). The conductivity of the sensor is higher along the gas concentration rising [7].

4.1.3 LDR - Light Dependent Resistor

This is the simplest light sensor. It is a variable resistance of a range between 10 M(Ω) down to 100 Ω.

Figure 4.5: LDR - Light Dependent Resistor

The LDR, in dark condition, has a very high resistance allowing to a small amount of current to pass. As the light (lux) increases, the resistance decreases proportionally.

4.1.4 PIR - Passive InfraRed motion sensor

PIR *Passive InfraRed* motion sensor is a ready to use integrated circuit. It can detect movement up to approximately 6 meter.

Figure 4.6: PIR - Passive InfraRed motion sensor

The sensor has a 3-pin connection interface. The red wire is the power (5 Volts), the brown wire is the ground (*GND*) and finally the black wire is the output. The alarm pin (output) is an open collector and therefore the introduction of a pull up resistor is essential. In a no movement state the sensor outputs a high signal (5 Volts). When on movement detection state the alarm pin goes low (0 Volt). The open drain setup allows multiple motion sensors to be connected on a single line. When one of the sensors detects movement the alarm pin will be pulled low.

4.1.5 LM335 - Precision Temperature Sensor

The LM335A is a analog temperature sensor. The LM335A works like a Zener diode with a breakdown voltage proportional to absolute temperature at 10mV/°K. When a resistor is connected between the 5V and the GND, and the LM335A will output an analog voltage of 2.98V (298 Kelvin is 25°C). The output of the sensor is linear, and when calibrated at 25°C the LM335A has typically less than 1°C error over a 100C temperature range. The sensor can operate continuously from 40°C to 100°C.

Figure 4.7: LM335 - Precision Temperature Sensor

4.2 The control board

The control board is the core of the *Apollo*. It is a PCB board which is equipped with:

- ▷ Arduino Pro Mini
- ▷ XBee RF Module

The board is connected through pin cable with the sensor board. The collected values are then prepared and sent. This is done with the use of the XBee RF Module.

4.2.1 Arduino Pro Mini

Arduino is an open-source platform based on a micro-controller and a simple software development environment. It can be used to build interactive applications, receiving inputs from several types of sensors and interacting with the environment controlling various actuators.

Arduino Pro Mini board is based on the ATMEL AVR 8-bit micro-controller, specifically on the ATmega168. Despite its size, it is equipped with both analogue as well digital input/outputs. More specifically it has 14 digital input/output pins, of which 6 can be used as PWM (*Pulse-width modulation*) outputs. In addition it has 6 analogue inputs, as well as a reset button. Each pin is a hole allowing pin headers to be mounted (soldered).

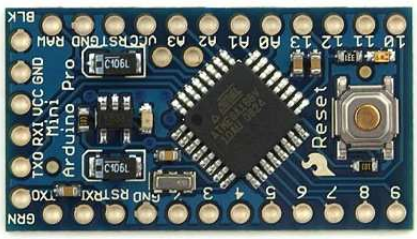

Figure 4.8: Arduino Pro Mini

There are two versions of the Arduino Pro Mini. One, which runs at 3.3 Volts and 8 MHz and another which runs at 5 Volts and 16 MHz. The former one is used for *Apollo*. Arduino Pro Mini has three different memories embedded on it; one **Flash** of 16 MB, one **SRAM** of 1 KB and one **EEPROM** 512 bytes.

It can be powered up by either USB connection or a external power supply. The recommended input voltage is between 5 to 12 V.
Specifications:

Micro-controller	ATmega168
Operating Voltage	5 Volts
Input Voltage	5-12 Volts
Digital I/O Pins	14 (6 provide PWM output)
Analogue Pins	6
DC Current per I/O Pin	$40mA$
Flash Memory	16 KB (2 KB dedicated to the bootloader)
SRAM	1 KB
EEPROM	512 bytes
Clock Speed	16 MHz

Table 4.1: Arduino Pro Mini Specifications

4.2.2 XBee RF Module

Arduino does not provide built-in wireless connectivity. Therefore the wireless communication is achieved by connecting an IEEE 802.15.4 module, the XBee RF.

XBee RF is an embedded module providing wireless connectivity to devices. The module uses the IEEE 802.15.4 standards for a low-rate wireless PAN (*Personal Area Network*) [**?**]. This makes it ideal for a low-cost, low-power wireless sensor network, such as *Apollo*.

Figure 4.9: XBee RF module

XBee is connected to the Arduino via a serial port and adds wireless capabilities by forwarding any data received from the Arduino to the wireless network and forwarding received wireless to the Arduino respectively.

4.3 The action board

The action board is a PCB board equipped with holes to fit up to 4 relays. The relays used are fully sealed SPDT (*Single Pole Double Throw*).

4.3.1 Relay SPDT Sealed

This is a high quality SPDT sealed relay. Suitable for high voltage and high current devices. [12]

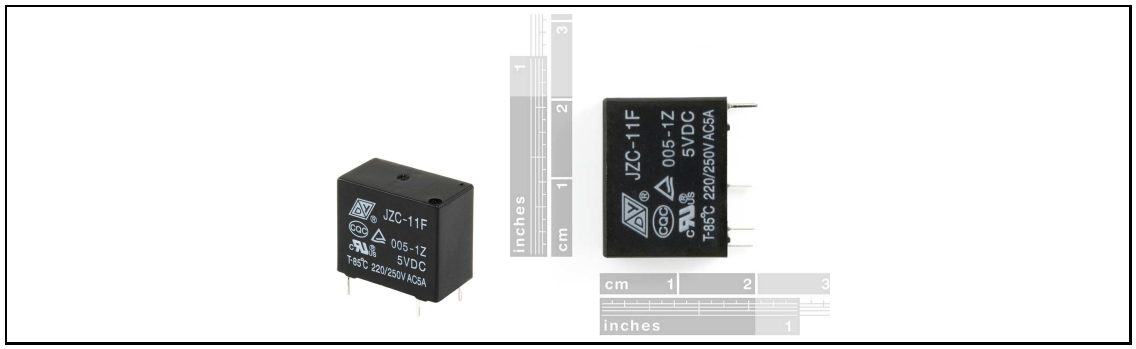

Figure 4.10: Relay SPDT Sealed

Specifications:

Max. Operating Current	5 A
Max. Operating Voltage	220 V AC / 30 V DC
Max. Operating Power	1250 V AC / 150 W
Contact resistance	100mΩ
Ambient Temperature	-40 ∼ +85°C
Mechanical life	10000000 Ops
Electrical life	100000 Ops

Table 4.2: SPDT Specifications

The action board is also equipped with **LEDs** (*light emitting diodes*) which are connected to the relays. This provides a visual and fast inspection for defect(s) on the board.

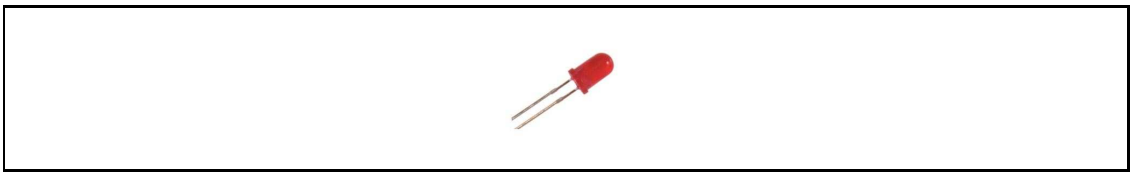

Figure 4.11: LEDs

4.4 Software

4.4.1 Arduino IDE

Arduino IDE is a software development environment for program applications for the Arduino platform. It is comprises a toolbar, which includes the most common functions, a text editor for writing the code and a message area. Arduino IDE is also used to upload programs and communicate with the Arduino boards [2].

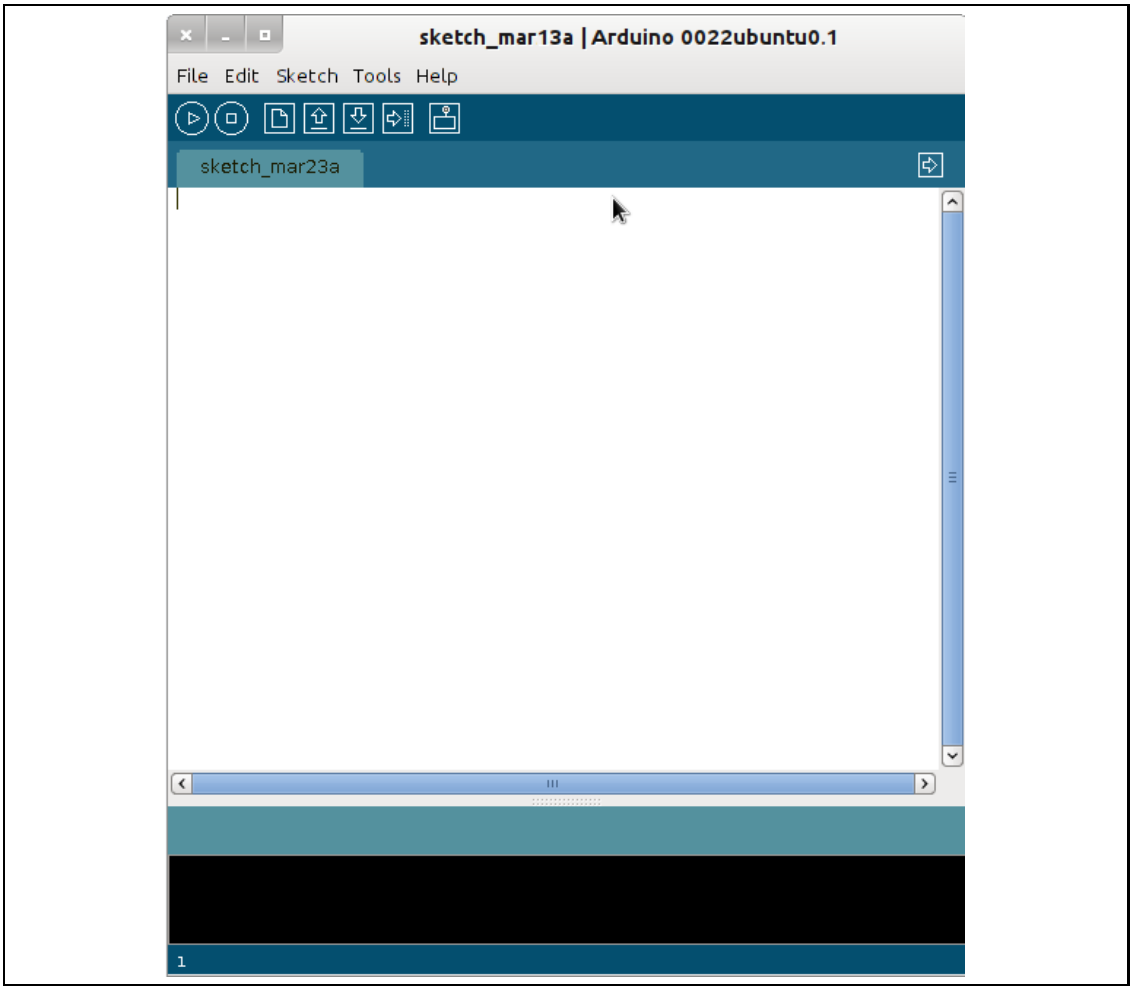

Figure 4.12: Arduino IDE software development environment

The toolbar buttons allow the user to verify programs, create new, open existing, save, upload to the Arduino board and the serial monitor. The programs written using Arduino Software are called **Sketches**. The serial monitor displays serial data being sent from the Arduino board it also can send data to the board. The message area is used to display errors and to provide feedback at the process of saving and uploading the programs on the board. The console displays text output by the Arduino environment including full error messages [2].

Sketches are written in C/C++ but developers need to define two methods:

- **setup():** runs only once in the beginning of the program and initialises the variables.
- **loop():** runs repeatedly inside the program until the power of the board be cut.

Beside these two basic methods Arduino IDE includes others which can be split in **structure**, **values**, **functions** [2]. These are defined below:

- **Structure:** include all the control flow structures of numerical and logical operations, such as: **if, if...else, and more**,
- **Values:** include all the data structure, such as: **boolean, char, int, byte, long, and more**,
- **Functions:** include all the management methods for the pins and the general internal function of the micro-controller, such as: **pinMode(), digitalWrite(), digitalRead(), analogRead(), tone(), interrupts(), and more**.

4.4.2 CoolTerm

CoolTerm is a software for communication with hardware connected to serial ports. It is a serial port terminal application for the exchange of data with hardware connected to serial ports such as micro-controllers. It has the same concept as the serial monitor of the Arduino IDE. The main differences that made it more suitable are the capability of multiple concurrent connections if multiple serial port connections are presented. It can also display the received data in plain text

as well as hexadecimal format. In addition, it can capture the received data to a text file [8].

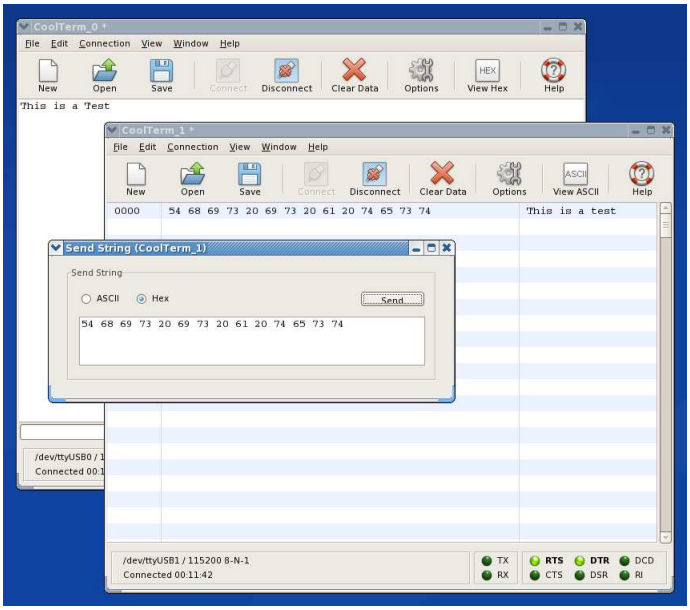

Figure 4.13: CoolTerm - serial port communication

4.4.3 Eagle PCB Software

Eagle PCB Software is a powerful and flexible PCB design program. Eagle is an acronym for Easily Applicable Graphical Layout Editor. It was used to design and print the PCB designs for this project (*Apollo*). To do this the schematic design of the three different boards first had to be drawn. The board design was then produced by arranging the components appropriately [3].

Figure 4.15: Eagle - Board design

Chapter 5

System Design

This section presents the overall design of the system as well as of the individual components implementation of *Apollo*. In addition, the design choices are explained and justified in detail and their advantages are analysed in depth.

The main guides for designing *Apollo* were the air quality standards for office and laboratory buildings. In addition the installation of the system should have the minimum impact on the building's structure. The maintenance time and cost of all three boards was another major factor for their design. Moreover the system's capacity for heterogeneous connection and interaction with other systems was one of the crucial characteristics for its design.

Apollo has passed various stages before taking the shape and form that has on the date of this report. The original approach included two boards; ***control*** and the ***sensor*** board. After a series of prototype testing seemed cogent to break the *control* board in two discrete ones. The outcome of this was the three boards; ***sensor, control, action***.

The significant difference of the two is the separation of the Arduino platform from the control board with the relays. The reason for that was the minimisation of the repair ($MTTR$). This increased the cost of the system. The cost increase is of approximately 1, according to the School technician. It is not possible to calculate the exact cost increase, as for the development of the PCBs the facilities of the School were used.

For the design of the boards, CadSoft Eagle CAD software was used. Eagle is equipped with an Electrical Rule Check ERC as well as with a Design Rule Check DRC . This is integrated functions that check the schematics for electrical and design errors respectively. If any error(s) are detected, an error dialogue window

pops with details on the errors.

5.1 Sensor board

5.1.1 Schematic implementation

Below, it will be presented the process of designing the sensor board with the use of Eagle CAD software.
To begin with, the development starts with the creation of the schematic. In the schematic all the necessary parts are being placed. 5.1

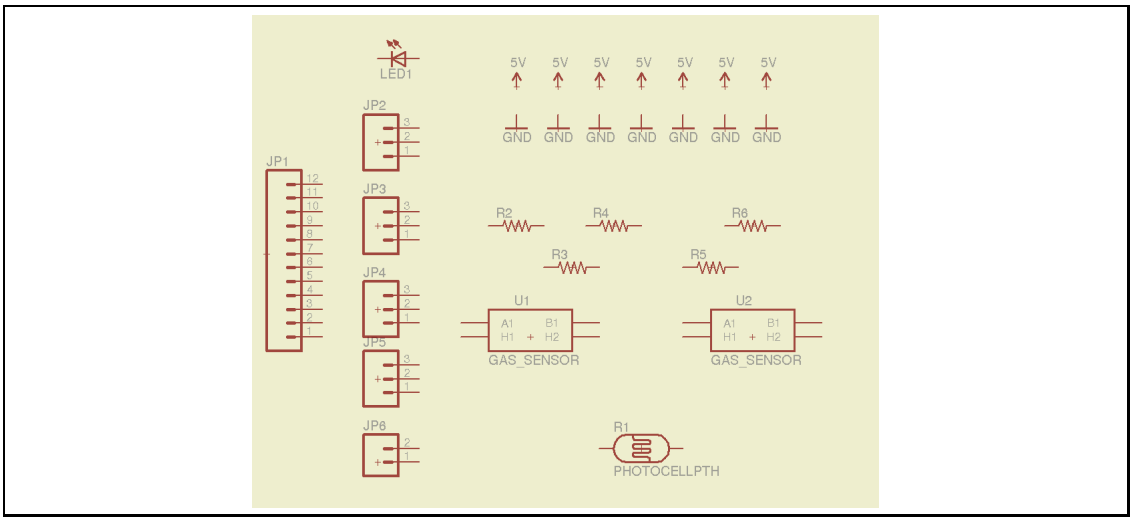

Figure 5.1: Sensor board - Adding all the parts (Stage 1)

For the schematic, and the board later, to be easier and clear what is what, the parts are named accordingly. 5.2

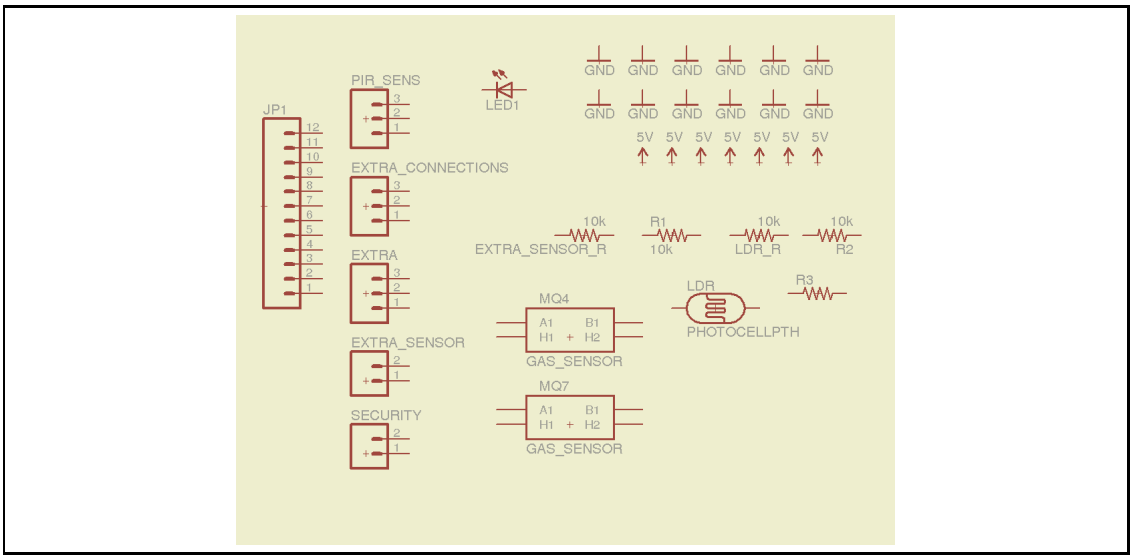

Figure 5.2: Sensor board - (Stage 2)

At this stage the parts are being spread. The reason for this is for the following step to be easier and more efficient.

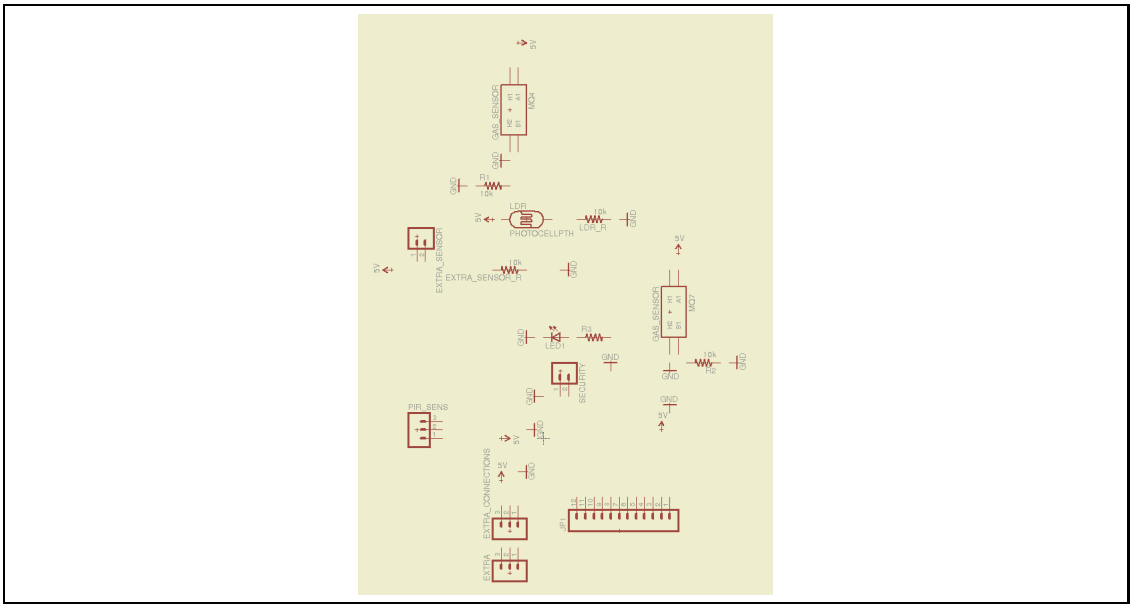

Figure 5.3: Sensor board - (Stage 3)

In the following Figure 5.4 the wiring of the parts of the sensor board is being made.

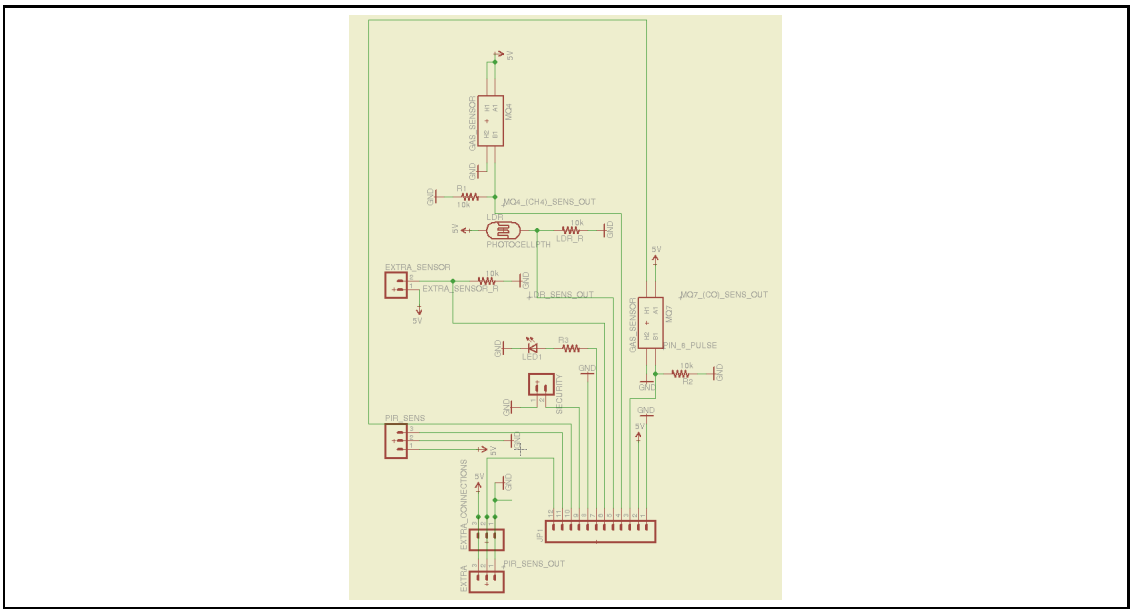

Figure 5.4: Sensor board - (Stage 4)

To check if the wiring is properly done, and there is neither short-circuit nor unconnected parts and **ERC** was performed. The result is shown in Figure 5.5. There were no errors reported back. The warnings are of somewhat minimum importance as the parts used are universal. On the other hand it is of importance, all pins to be connected either to other parts and/or to the power supply and/or to the ground.

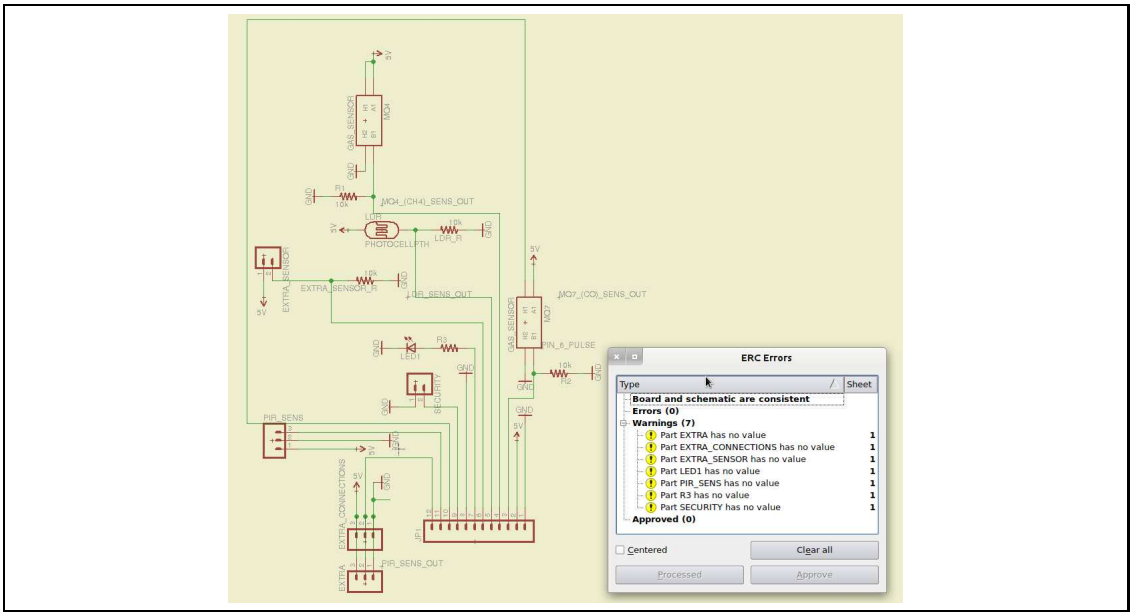

Figure 5.5: Sensor board - (Stage 5)

5.1.2 Board implementation

This is the stage where the parts implemented on the previous part will be take shape on the board. As the board will be installed in a visible part in a room, it is important to be as discreet as possible. In the Figure 5.6, below, the components are ready to be spread in the available space. The size of the board will be 5cm wide and 5cm high.

Figure 5.6: Sensor board - (Stage 6)

Once all components were spread with sufficient space between them the *Ratsnest* tool was used to minimise the path of the of the wiring. This time the wiring is the conductive medium,of electric current between the parts of the board, the copper.

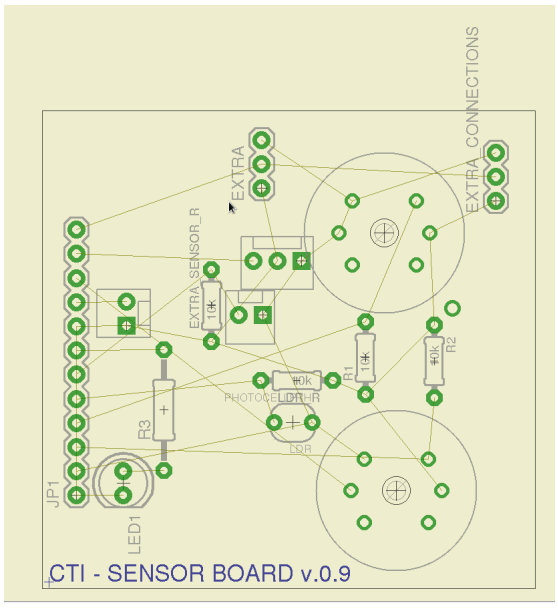

Figure 5.7: Sensor board - (Stage 7)

The wiring route were finally generated with the use of the *Autoroute* tool. To achieve the optimal result, in this case the minimum copper paths *Autoroute* has an *Optimiser* option which was used. Moreover the settings of the *Optimiser* were set accordingly to achieve the least possible layers and bias.

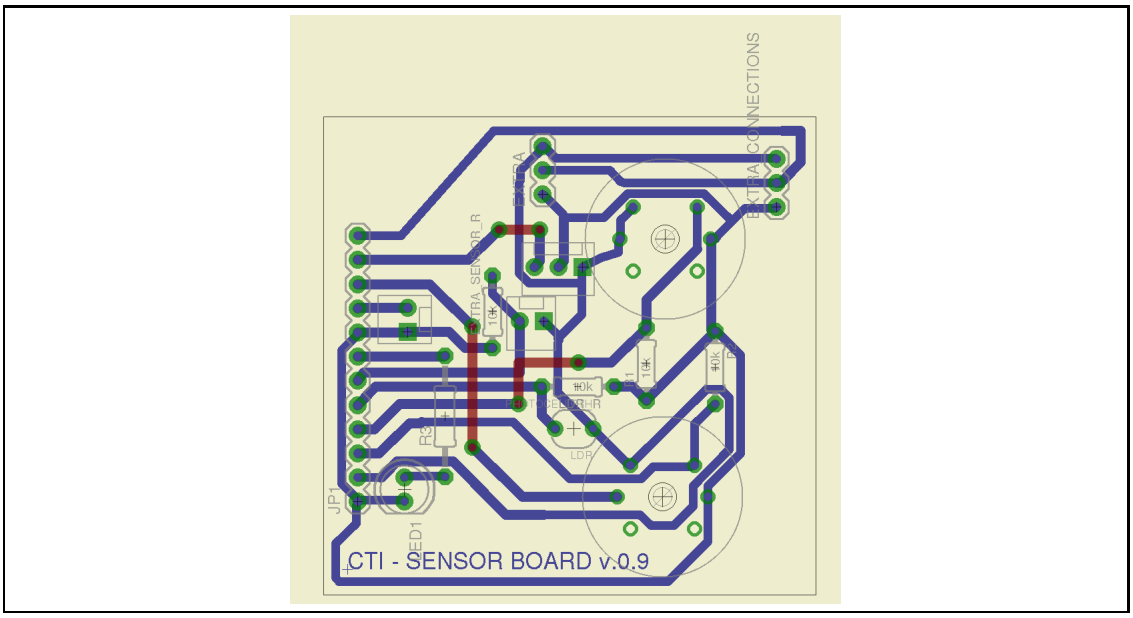

Figure 5.8: Sensor board - (Stage 8)

Layers are the sides of the board used for the creating the copper paths. In Figure 5.8 the top layer wire paths are marked with maroon color, where the bottom layer paths are on blue. Bias are called the points where the wire path changes layer. Bias add extra difficulty on soldering the board and therefore are being avoided when possible.

5.2 Control board

The same method as with *Sensor* was used for designing and developing the *Control* board. The following Figures 5.9 5.10 5.11 5.12 show the process.

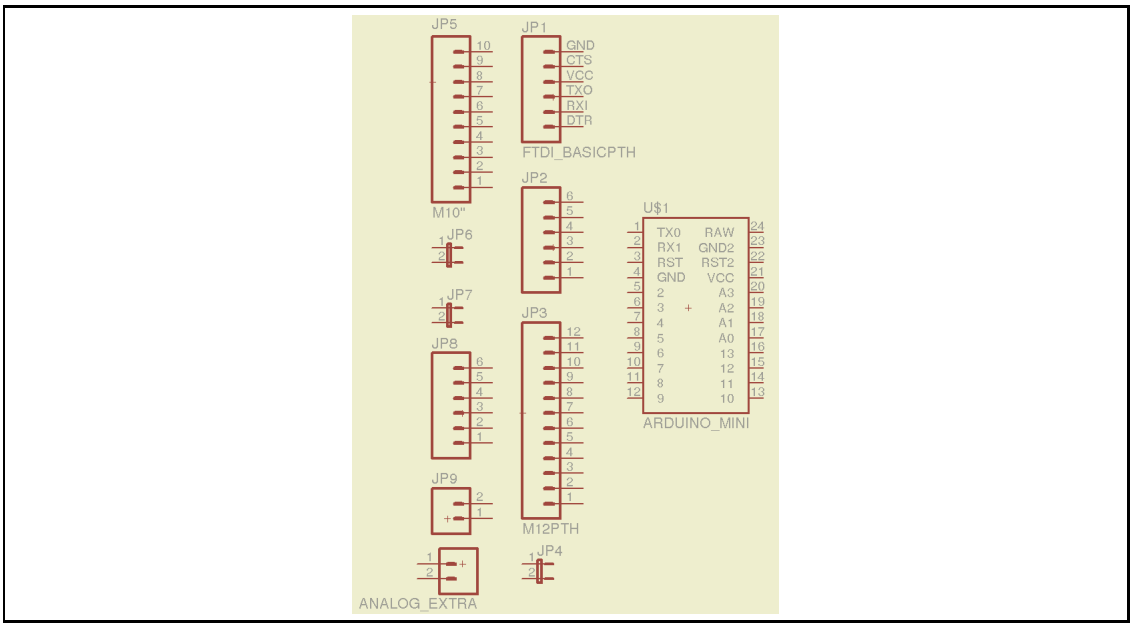

Figure 5.9: Control board - (Stage 1)

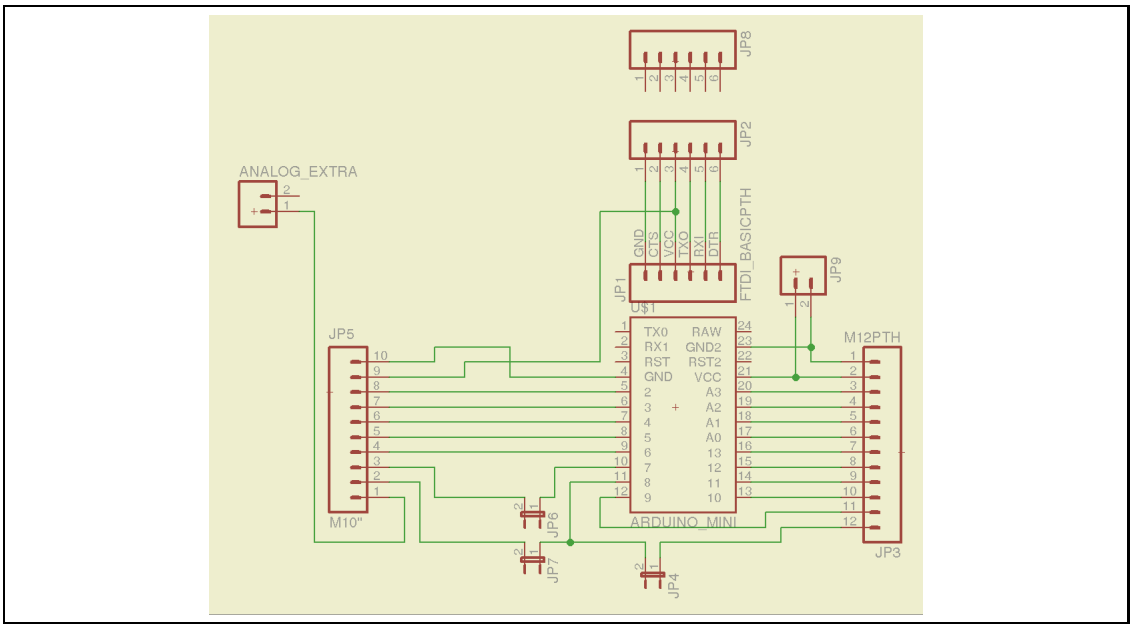

Figure 5.10: Control board - (Stage 2)

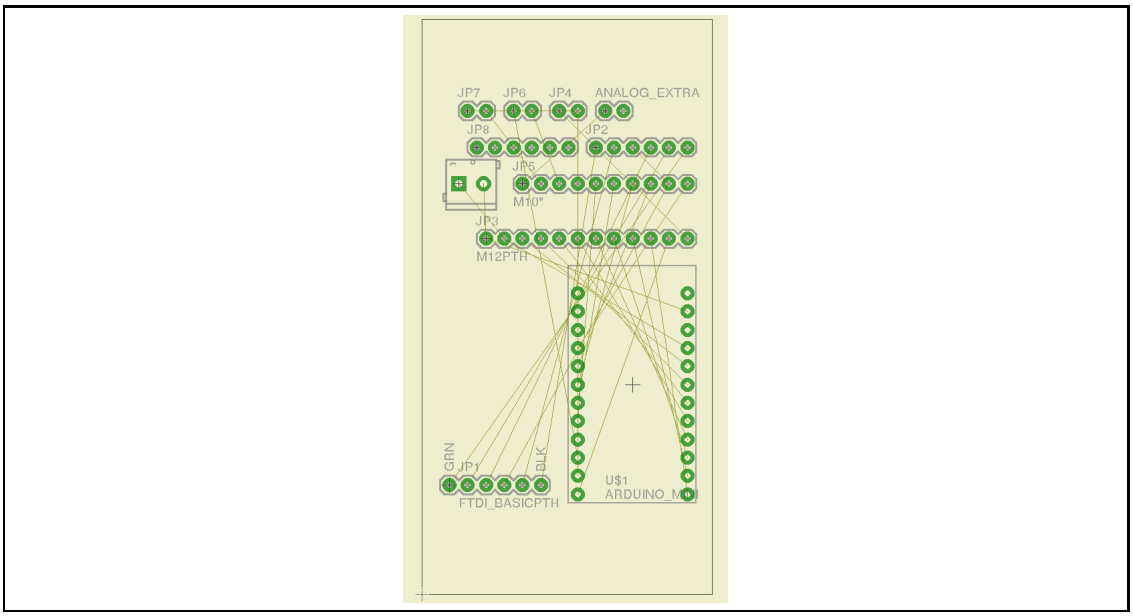

Figure 5.11: Control board - (Stage 3)

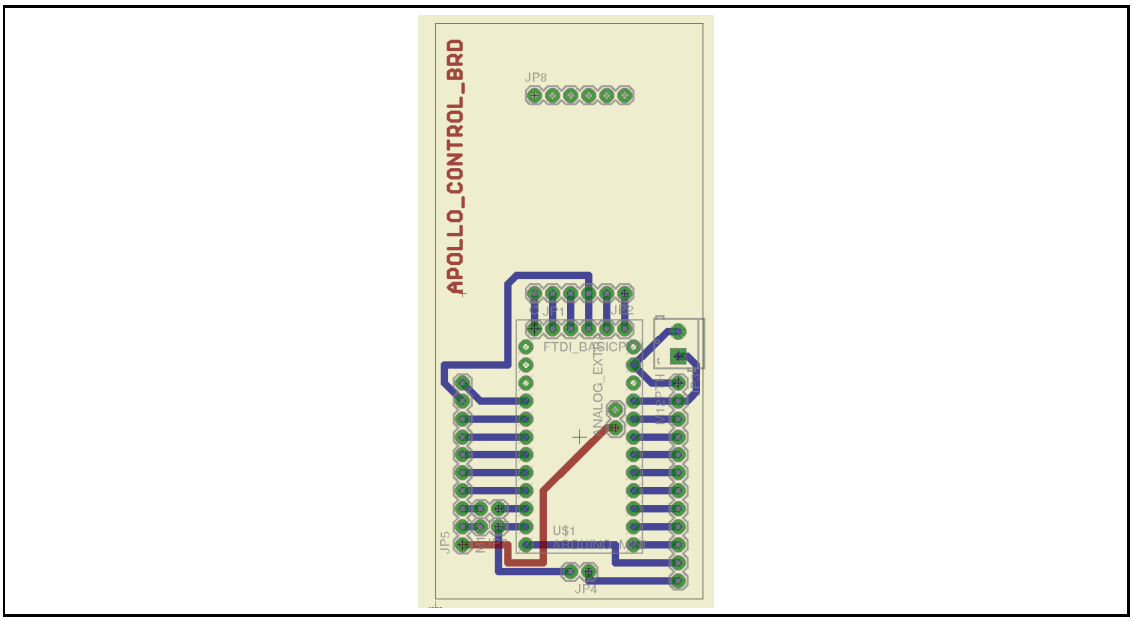

Figure 5.12: Control board - (Stage 4)

At this stage is essential to clarify the existence of the six pin holes (*JP8*) in the above Figure 5.12. Those six pin holes will host a six pin header (Figure 5.13). This is one of many practical ways to hold the *XBee* shield straight. With this method the shield's pins will not be bended, therefore any damage will be avoided.

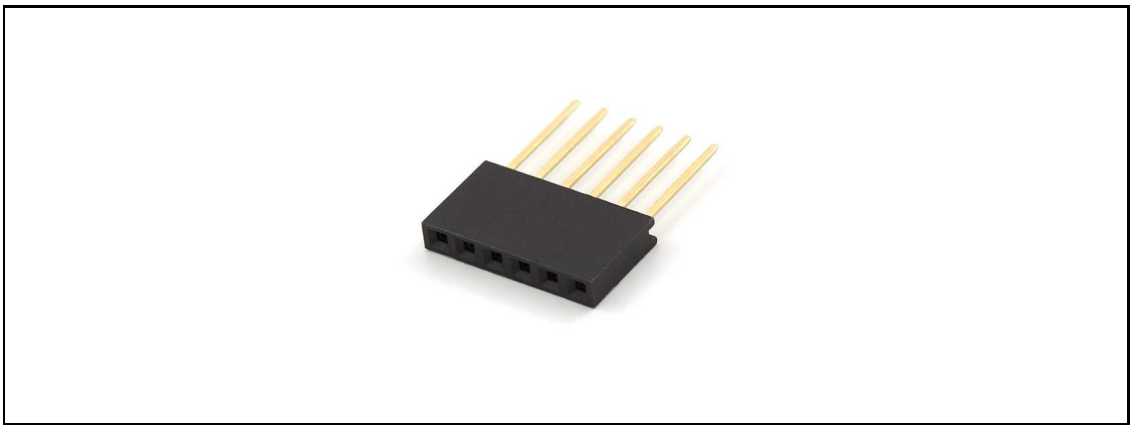

Figure 5.13: Arduino header

5.3 Action board

For the development of the *Action* board was used the same method as of the *Control*. At this point, it is important to explain the existence of the four *LEDs* on the board. Their purpose is to have a quick visual check of the signal going into the relay. In this way, it is possible to detect if any or more of the relays has/have fail. With this technique, precious time is saved from removing the *action* board.

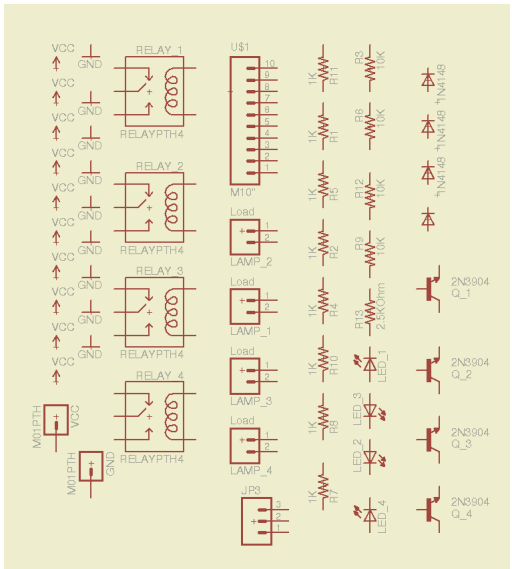

Figure 5.14: Action board - (Stage 1)

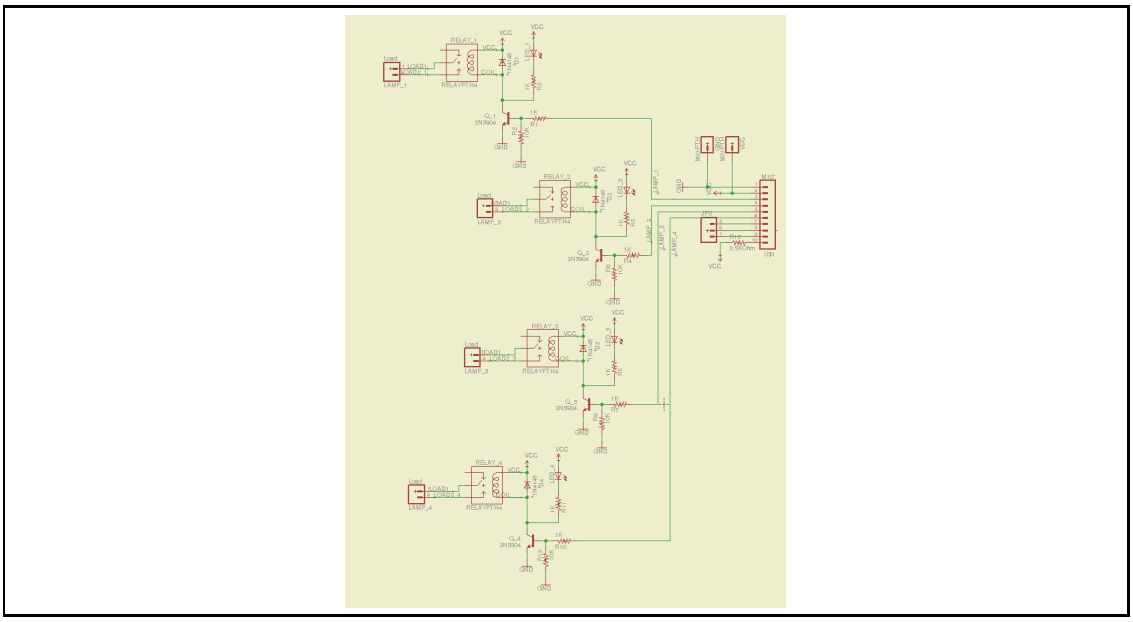

Figure 5.15: Action board - (Stage 2)

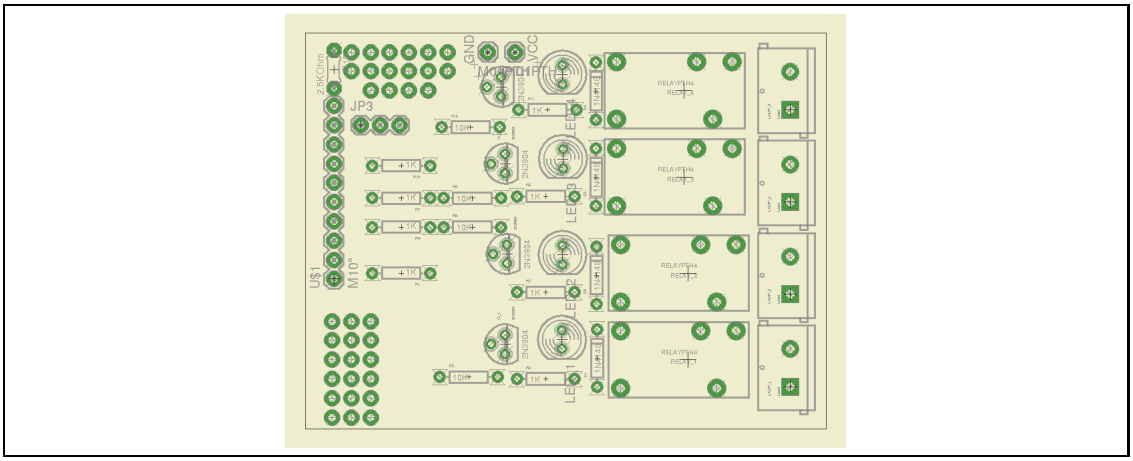

Figure 5.16: Action board - (Stage 3)

5.4 Arduino code

In this section an extensive description of the code that was created for *Apollo* will be carried out. The code is divided in three parts; ***initialisation, setup(), loop()***

5.4.1 Initialisation

The very first part of the code consist the *initialisation*. This includes the libraries as well as the global variables, which will be used across all the program.

The heterogeneous communication protocol which allows the Arduino to communicate with different devices is included as a header in this part. This is the *XbeeDario.h* and was provided by the Computer Technology Institute and Press "Diophantus".

Furthermore the global variables were declared at this stage as well. Global variables include the Arduino's pins and to what they are connected to.

5.4.2 setup()

The connected pins can be set as input or output depending of their purpose. This is done in the *setup()* part of the code.

In addition the communication between the Arduino and the XBee is set up. Moreover the XBee is connected to the pins of the Arduino. Therefore it is compulsory to specify the communication serial port between them two.

5.4.3 loop()

Loop() is the final and longest part of the code. To start with Arduino checks if the other two boards (*sensor, action*) are connected to it. This way the system do not have to be removed in case any of the boards has a defect and is down.

If the *action* board (relays) are connected, the number of relays are being checked. As the specifications of the room that the system is installed may vary therefore the *action* board my not have all four relays installed. To check how many relays are on the *action* board different resistance is used for each case. Therefore different current values are being measured in the *relayCheckPin*. This values are mapped in the code and the system reports back how many relays are installed.

The next function controls the output of the relays. The received data from the Controller is checked if it contains any value for the relay(s) and in what state does it set it. The next step is to check which of the lamps are on and which are off and report this information. Also reports any change on the status of the lamps to the Controller.

The following group of statements check the status of the motion sensor (*PIR - Passive InfaRed*). Every half of a second the status of the PIR sensor is being checked and his status is being reported as well.

Following, the light sensor is called every three seconds and its value is being reported at the time of measurement.

On the same rate of every three seconds, the value of temperature is being measured. Explain the temp function.

The value of methane (CH_4) is also measured every three seconds. This is simply being done by reading the analogue value of sensor's (*MQ-4*) output pin.

It was not possible to measure the CO levels as there was an ambiguity on the return values of the *MQ-7* sensor. The issue was determined around the heating time and the working cycles before sensor's values were evaluated.

5.5 Database

Überdust stores data to an RDBMS using the WiseDB component of the WiseML library based on the state of the art O/RM framework, Hibernate. Hibernate is probably the most well establish relational persistence framework for Java. Hibernate facilitates the storage and retrieval of Java domain objects via Object/Relational Mapping (ORM). Although *Überdust* was originally developed for storing sensor data from wireless sensor networks, any source can provide data as long as it is in a simple timestamp-based format and a small set of metadata is already provided in the persistence store. and was provided by the Computer Technology Institute and Press "Diophantus". [4]

Chapter 6

Results

Apollo was built completely as it can be seen in the following Figures 6.1 6.2 6.3.

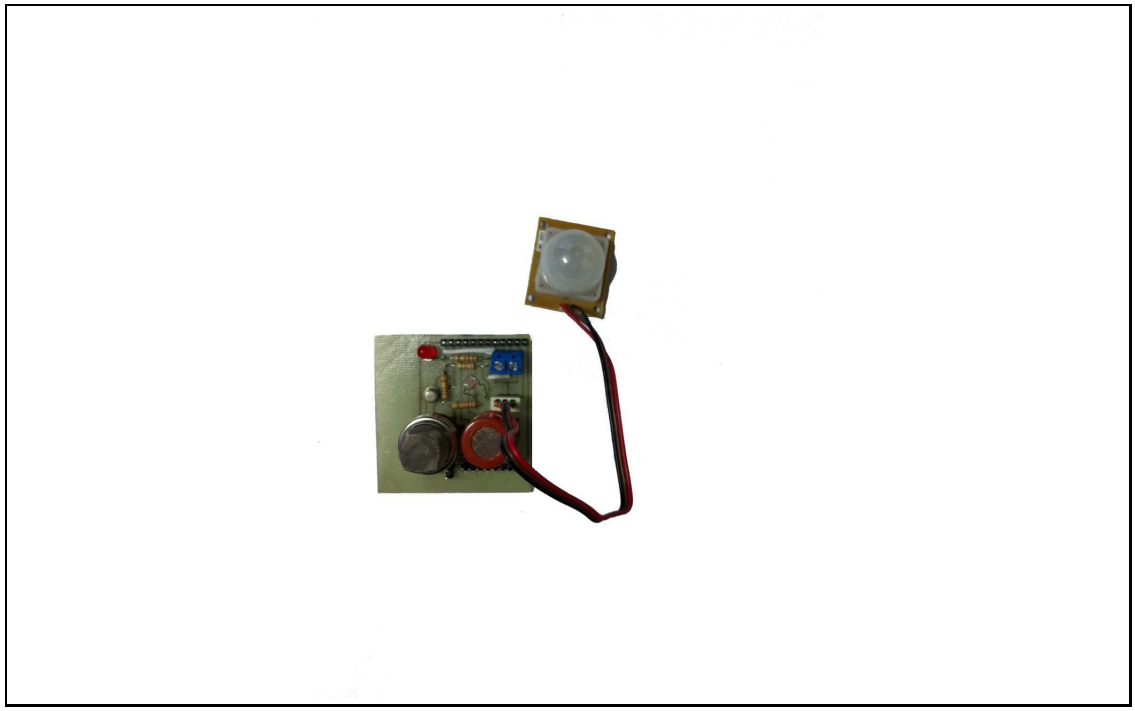

Figure 6.1: Sensor board - Completed with all parts soldered

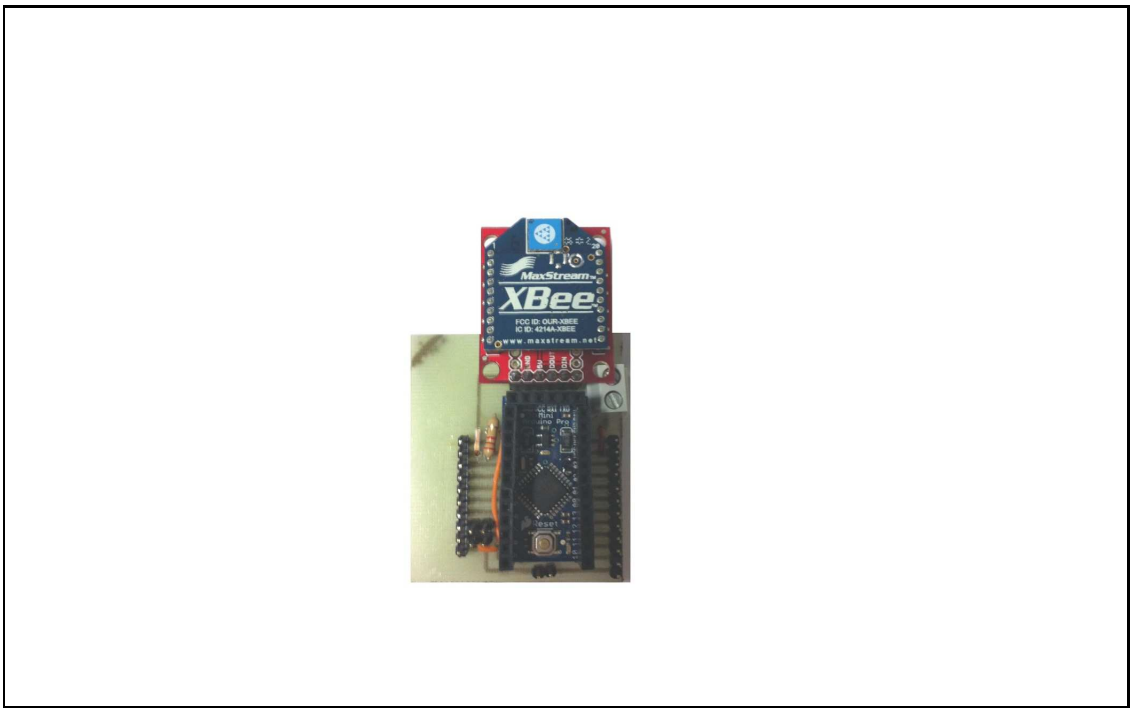

Figure 6.2: Control board - Completed with all parts soldered

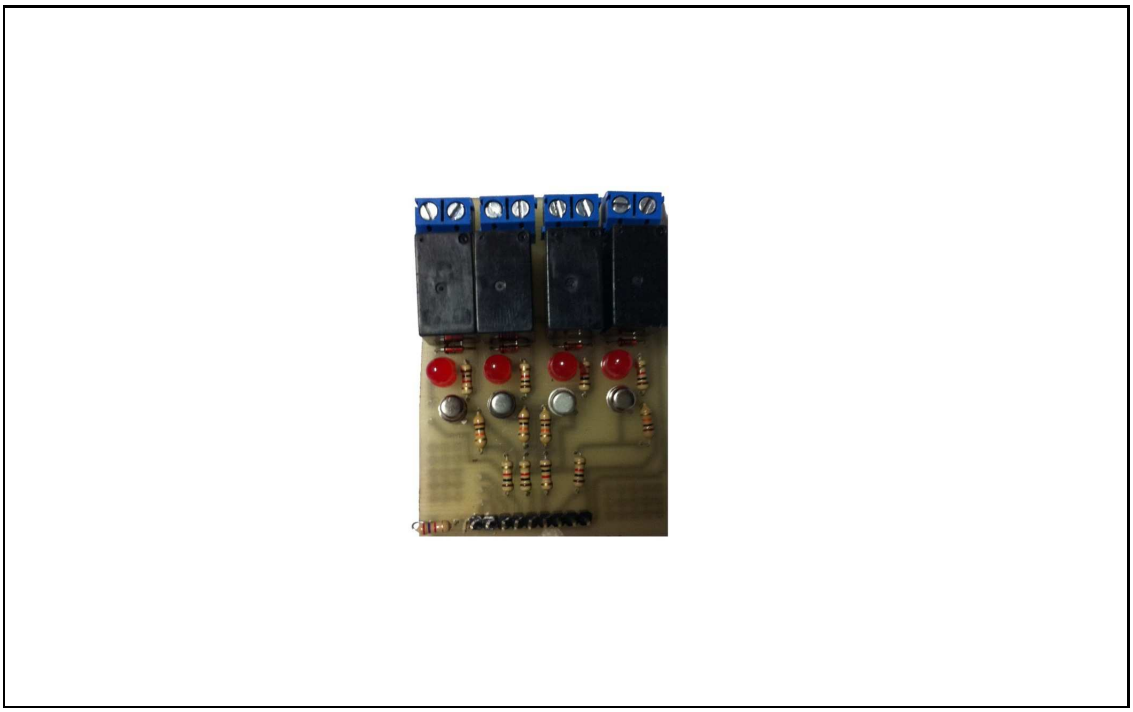

Figure 6.3: Action board - Completed with all parts soldered

In the next figure is shown the connections between the three boards.

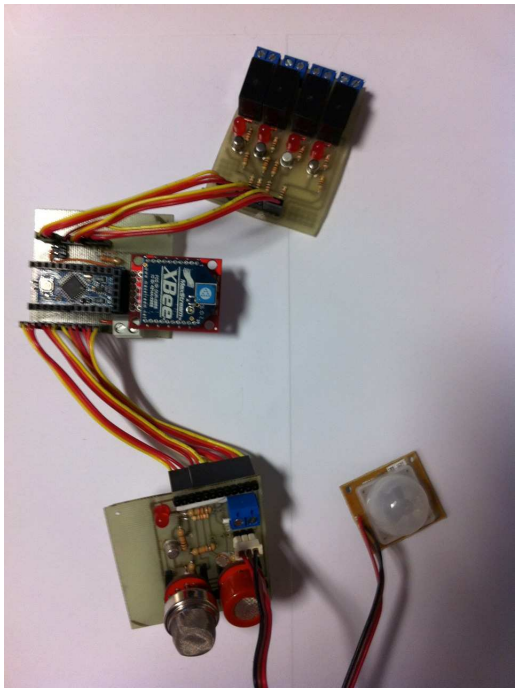

Figure 6.4: *Apollo* - All boards are connected

The system was installed into the rooms 0.I.1, 0.I.3, 0.I.9 and 0.I.11 of the lower ground floor of the Computer Technology Institute and Press "Diophantus" building in Greece. The selection of the rooms was made on the fact that they are occupied by different number of people. This was an extra test of the reliability of the system. Thus the size of the rooms defers as well as the number of lamps and air-conditioning units.

The moment each system went live it started reporting the environmental values and feed the *Uberdust* database. Below are shown the data collected into the database for each room separately, categorised by the type of sensor.

6.1 PIR

Room 0.I.1

Figure 6.5: *Überdust* chart for for PIR sensor in room 0.I.1

Room 0.I.3

Figure 6.6: *Überdust* chart for for PIR sensor in room 0.I.3

Room 0.I.9

Figure 6.7: *Überdust* chart for for PIR sensor in room 0.I.9

Room 0.I.11

Figure 6.8: *Überdust* chart for LDR sensor in room 0.I.11

6.2 Light

Room 0.I.1 From the following Figure 6.9 is clear the pattern of the light emittance that is followed each day.

Figure 6.9: *Überdust* chart for LDR sensor in room 0.I.1

Room 0.I.3 In the expanded chart in Figure 6.10, is visible the light's strength during a day.

Figure 6.10: *Überdust* chart for LDR sensor in room 0.I.3

Room 0.I.9 In this case the illuminance of different room is being presented.

Figure 6.11: *Überdust* chart for LDR sensor in room 0.I.9

Room 0.I.11 The occupants of this room have a preference of low light emittance. This can be seen from the low values from the light sensor.

Figure 6.12: *Überdust* chart for LDR sensor in room 0.I.11

6.3 Temperature

Room 0.I.1 The temperature show a steady value around $24°C$ during the whole day. This is due to the fact of electronics dissipating heat.

Figure 6.13: *Überdust* chart for Temperature sensor in room 0.I.11

Room 0.I.3 As all rooms are installed with computers with high computing power the levels of dissipating heat are quit high.

Figure 6.14: *Überdust* chart for Temperature sensor in room 0.I.11

Room 0.I.9 The same situation is seen in this room as well.

Figure 6.15: *Überdust* chart for Temperature sensor in room 0.I.11

Room 0.I.11 In Figure 6.16 it is indicated that around 14:00 the ventilation was turned on. This is due to just a small drop of temperature.

Figure 6.16: *Überdust* chart for Temperature sensor in room 0.I.11

6.4 CH_4

The *Methane* CH_4 levels are constant and at a low value. The *glitch* noticed between 08:00 and 12:00 is due to sensor's fault return value. Room 0.I.1

Figure 6.17: *Überdust* chart for PIR sensor in room 0.I.11

Room 0.I.3

Figure 6.18: *Überdust* chart for PIR sensor in room 0.I.11

Room 0.I.9

Figure 6.19: *Überdust* chart for PIR sensor in room 0.I.11

Room 0.I.11

Figure 6.20: *Überdust* chart for PIR sensor in room 0.I.11

6.5 CO

Unfortunately as it has been explained in section ***5.4.3 loop ()*** the *CO* sensor is not yet funtioning properly.

Chapter 7

Conclusions & Future improvements

To the author's opinion the project tends to be characterized as successful due to the fact that the aims have been achieved. The system is installed and working in the Computer Technology Institute and Press "Diophantus" building, lower ground floor. Evidence provided by the database feeding and everyday working status clearly supports the success of the system.

Moreover *Apollo* is able to collect the environmental values from the room is installed in and report them back to the *Controller* and in extension to the database. It is also capable of operating the lighting as well as the air-conditioning of the room. This is a novel approach of such a system, as with the use of the IEEE 802.15.4 protocol the communication with other systems is feasible. On that point the system can work along with other systems such as TelosB, SunSpot, iSense and any other system that uses the same communication protocol.

Apollo has a great potential of expansion. As a first step, the development of a pressure sensor system. This addition it would enable the option of personalising the system according to the each user. On that note the pressure sensor could wireless send information of the presence of a specific user. Moreover the system could respond to the user's preferences for lighting and temperature. With this technique if more than one user occupied the room, each can have a separately preferences for his working space.

Additionally to the last suggestion the presence of a blue-tooth module connected to the *action* board could expand system's personalisation. Nowadays every mobile device has a blue-tooth connectivity option. The system could recognise the

presence of a specific user and would set the working environment accordingly. This would mean the light along with the temperature level would be set upon user's arrival in the building.

Introduction of security alerts via e-mail could be considered a further achievement. Additionally the system would recognise an ambiguity activity. The owner of the office would be notified for this activity via e-mail.

Most office buildings are equipped with a Building Management System (*BMS*). BMS is a computer based control system, which operates the electrical and mechanical equipment, such as ventilation, lighting, power systems, fire systems and security systems of a building. Priceless energy could be saved on the overall building. This could easily achieved if the electrical and mechanical systems of each floor or section (depending on the BMS abilities) would operate only when necessary.

Bibliography

[1] AirMaster. Fresh air is commonn sense. http://goo.gl/2AVsV.

[2] Arduino. Arduino development environment. http://arduino.cc/en/Guide/Environment.

[3] CadSoft. Home of cadsoft eagle pcb design software. http://www.cadsoftusa.com/?language=en.

[4] Chatzigiannakis, I. Wiseml. https://github.com/ichatz/wiseml/wiki.

[5] Coetzee, L., and Eksteen, J. The internet of things promise for the future? an introduction. pp. 1–9.

[6] for Communities, D., and Government, L. Part l of the building regulations. http://goo.gl/O2Okb.

[7] HANWEI, Electronics, C. Technical data mq-7 gas sensor. http://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf.

[8] Meier, R. Roger meier's freeware. http://freeware.the-meiers.org/.

[9] Pachube. About us - pachube. https://pachube.com/about_us.

[10] Sparkfun. Carbon monoxide sensor - mq-7. http://www.sparkfun.com/products/9403.

[11] Sparkfun. Methane cng gas sensor - mq-4. http://www.sparkfun.com/products/9404.

[12] Sparkfun. Relay spdt sealed. http://www.sparkfun.com/products/100.

[13] Sutter, J. D. 'smart dust' aims to monitor everything. http://goo.gl/KdK7q.

[14] Wikipedia. Internet of things. http://en.wikipedia.org/wiki/Internet_of_Things.

[15] Wikipedia. Sun spot. http://en.wikipedia.org/wiki/Sun_SPOT.

Appendices