People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of
The Requirements of the Degree of

'MASTER'

In: **Computer Engineering**

Title:

FPGA-based Climate Controller for a Greenhouse

Presented By:

- **Fatima MEKLOUT**

Supervisor:

Dr. A. BENZEKRI

Registration Number: 2021/2022

Dedication

“ This project is wholeheartedly dedicated to my parents, without whom I don’t know where I would be, what values would I hold and what path would I follow. I would like to extend my gratitude towards my sister Narimene who always has my back no matter the circumstances, my brothers Amine and Yanis who express very little and show just beyond, Meriem, Melissa and Ismail for everything he did and still does for me.
I would also like to dedicate this work to all of my friends, classmates, teachers and the ISC family who have encouraged and supported me when I needed it most; this journey would have never been what it is without you taking part of it.”

Fatima

Abstract

This report describes the design and implementation of a climate controller for a greenhouse automation system. It deals with controlling different environmental parameters inside a greenhouse that ensure the growth requirements of plants.

The approach to design this digital controller is the SoPC technique and it is implemented using both hardware and software components.

The prototype for this project consists of four parts, the On-chip system integrating the Nios II soft core processor along with an on-chip memory, I/O peripherals, UART port and custom VHDL blocks. The off-chip system that includes different motor driving units, actuators and data acquisition unit. The smartphone that runs an android application for data display and the laptop.

The digital controller is developed using the Quartus II V13.0 web edition software development tool on a Cyclone II EP2C35F672C6 Field Programmable Gate Array and the android application is developed using Kodular tool.

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Dr.Benzekri, who provided me with valuable guidance, help and support throughout the implementation of this project.

The fulfilment of this project would not have been possible without the cooperation and contribution of several individuals; therefore, I would like to acknowledge them all for their insightful support and encouragement. I would also like to give special thanks and credit to Anis Slimatni who guided me and provided me with valuable advice throughout this entire process and Mohamed Yanis Hiou who was of great help whenever needed.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
ARM	Advanced RISC Machine
ASIC	Application-Specific Integrated Circuit
AI	Artificial Intelligence
CAD	Computer-Aided Design
CLB	Configurable Logic Block
CMOS	Complementary Metal Oxide Semiconductor
CPLD	Complex Programmable Logic Device
CPU	Central Processing Unit
DC	Direct Current
DE2	Board Development and Education board cyclone 2
DSP	Digital Signal Processor
DFF	D Flip Flop
EDR	Enhanced Data Rate EOC End Of Conversion
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
IC	Integrated Circuit
I/O	Input/Output
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
LAB	Logic Array Block
LDR	Light Dependent Resistor
LE	Logic Elementxi
LUT	Look Up Table
NDIR	Non Dispersive Infra Red
NRE	Non-Recurring Expense
OEM	Original Equipment Manufacturer

OS	Operating System PC Personal Computer
PLL	Phase-Locked Loop
PWM	Pulse Width Modulation
PC	Portable Computer
RAM	Random Access Memory
RISC	Reduced Instruction Set Computer
SDRAM	Synchronous Dynamic Random-Access Memory
SoC	System on Chip
SoPC	System on Programmable Chip
SPP	Serial Port Protocol TCO Total Cost of Ownership
TCO	Total Cost of Ownership
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VHDL	VHSIC Hardware Description Language
VHSIC	Very High-Speed Integrated Circuit

Chapter 01

Introduction

1.1 Overview

Greenhouse farming is essential in increasing domestic crop production. Smart greenhouse development is even more important to achieve high levels of food security. While the main aim of greenhouses is to offer an appropriate environment for high-yield production while protecting crops from adverse climate conditions, smart greenhouses provide precise regulation and control of the microclimate variables by utilizing the latest control techniques and communication infrastructures. Smart management systems thus provide the optimal environment for crop development [1].

With the growth of population, more and more greenhouse spaces are becoming necessary to assure food security and with that increase, it is becoming harder to monitor and control greenhouses from a human level and as days are passing by, there is a growing emphasis on developing data-driven agricultural solutions, as well as shifting-in the industry towards controlled environment agriculture.

An environment controller automation system for a Greenhouse is a self-regulating, micro-climate controlled environment for optimal plant growth that is based on sensors, actuators, monitoring and control systems. Climatic conditions inside the greenhouse, such as, temperature, light intensity, soil moisture.. directly affect the growth of crops inside a greenhouse and thus these parameters must be continuously monitored. Small variations in these climatic conditions trigger automated actions. The automated actions evaluate change and take corrective actions to maintain optimal conditions for plant growth. **Figure1.1** Show an illustrative example of different climate parameters that we ought to control inside a greenhouse.

Figure 1.1 Illustration of climate parameters aimed to be controlled inside a greenhouse.

1.2 Project Objectives

The main objective of this project is to design and implement a Nios II microprocessor-based environment controller for a greenhouse that will monitor the growth of crops inside the greenhouse by controlling the actuators that adjust environment parameters inside the greenhouse. The system is intended to be automatic and to be implemented using the Cyclone II family Field Programmable Gate array and the Nios II soft core processor along with a smartphone android application that serves as a Graphical User Interface for the user to check the status of different climate parameters consisting of a roof, lights, fan and irrigation pumps. In addition to checking the security inside the greenhouse from potential fires detection. This is aimed to be done via wireless communication from the FPGA to the android application using a Bluetooth Module.

1.3 Structure of the Project

This project was built in the laboratory for controlling the climate inside a greenhouse.
It consists of four parts:

- **The On-Chip Hardware**

Implemented on an FPGA, it contains both the SoPC system -including the Nios II processor, on-chip memory and I/O peripherals- and VHDL control blocks

- **The Off-Chip Hardware**

It is implemented on an external protoboard and it contains the different sensors with the data acquisition unit, actuators, drivers, ICs, communication unit, and connections with the FPGA through the expansion header.

- **The Smartphone**

The smartphone is used to run the android application where it displays different sensor readings coming through the Bluetooth module

- **The PC**

The laptop is an essential part of our project as it is responsible for providing different design platforms such as the Quartus II V13.0 and the Nios II software build tools for Eclipse as well as the kodular design platform; it is also responsible for conducting system compilation, downloading, implementing and running the on-Chip hardware and software parts of this project.

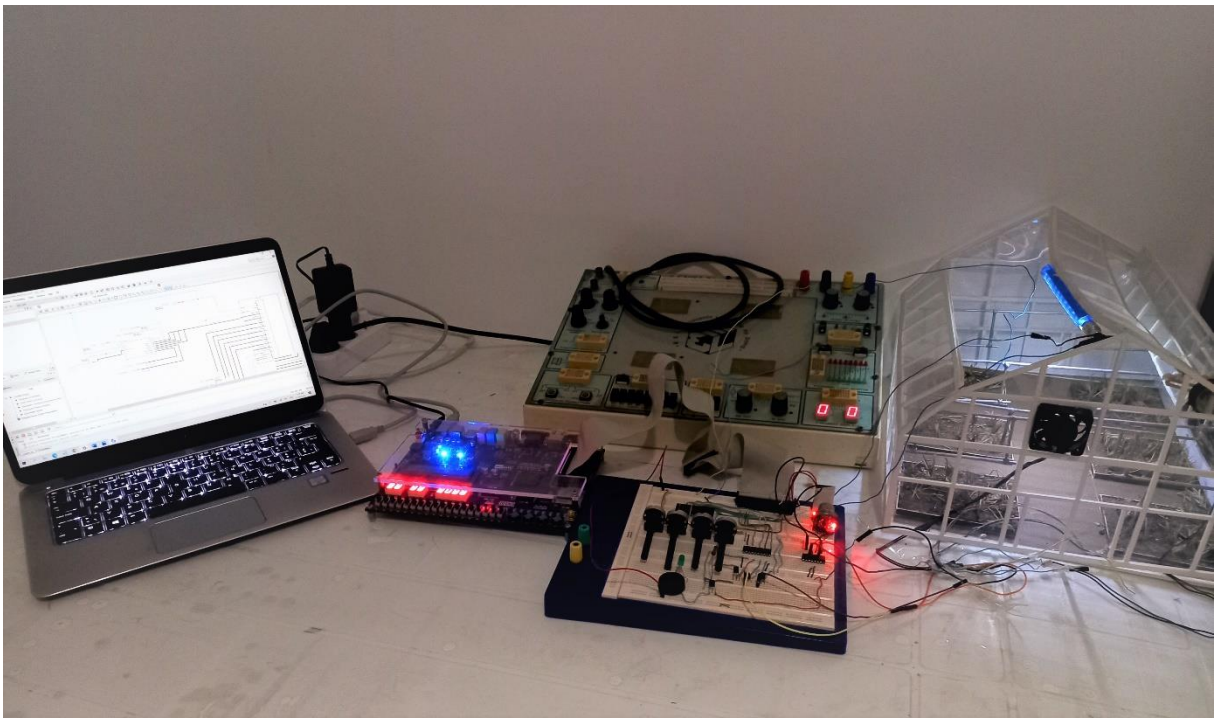

Figure 1.2 The Overall Prototype Built in the Laboratory.

1.4 Organization of the Report

This report is divided into four chapters. The first chapter is a general introduction to the project and the report. The second chapter deals with the theoretical background required for the implementation of this project; it describes different components and materials used in the course of the project along with their working principal and special requirements. The third chapter is the core of the work done in this project, it describes in details the design and implementation of the hardware with its different parts. The fourth and last chapter describes the software development flow of this project, it includes the firmware and the android application design description. A general conclusion is added along some suggested future work and the report ends with a set of references.

Chapter 02

Theoretical Background

2.1 Overview of Embedded Systems

An Embedded system is a computer system - a combination of a computer processor, computer memory and input/output peripheral devices - that has a dedicated function within a larger mechanical or electronic system [2-3]. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts [4].

Embedded systems are used in a wide variety of applications and their design depends on the different requirements of those applications and they include cost, general/specific computation speed/need, real-time constraint, reliability and power consumption [5].

2.11 Field Programmable Gate Array

2.11.2 General Structure

Field Programmable Gate Arrays (FPGAs) are semiconductor devices based on I/O blocks which are located around a two-dimensional array of a core programmable fabric known as Configurable logic blocks CLBs and a reconfigurable interconnect consisting of wires and switches. FPGAs are designed to be electrically configured by the customer after manufacturing to become almost any customized digital system.

Logic blocks can be configured to execute sophisticated combinational operations or to behave as simple logic gates such as AND and XOR. Memory components, which might be simple flip-flops or larger memory blocks, are included in most FPGA logic blocks. [1] Many FPGAs may be reprogrammed to perform various logic functions, allowing for flexible reconfigurable computing similar to that done in computer software [6].

FPGAs were first introduced in 1983 by Altera as concurrents of SPLDs and CPLDs to serve the purpose of glue logic i.e. connecting chips togethers. As their capabilities and size grew dramatically, it allowed the integration of entire systems on a single chip. Today FPGAs play a significant role in embedded system development (HW) Due to their ability to start system software (SW) development concurrently with hardware development.
Figure 2.1 Depicts the conceptual framework of an FPGA device.

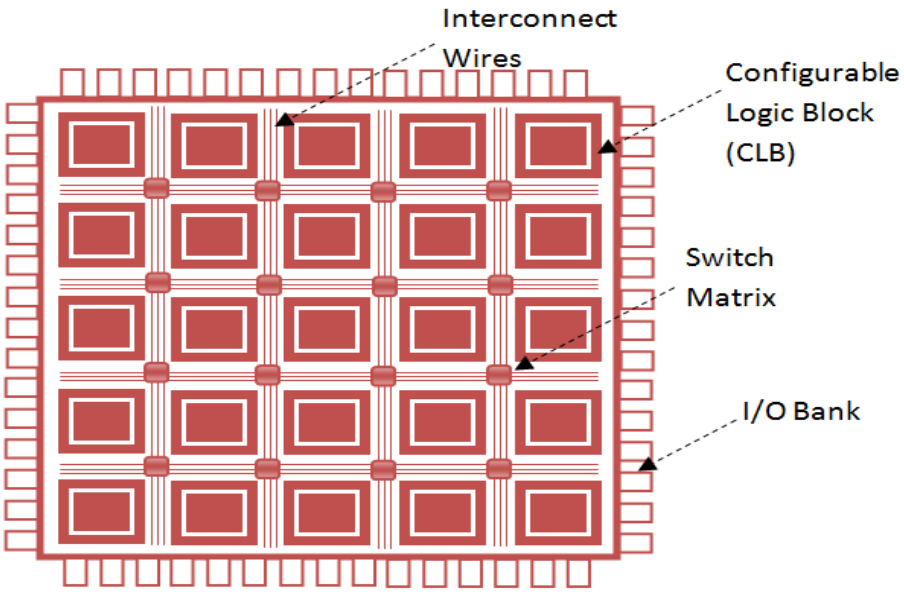

Figure 2.1 Conceptual Framework of an FPGA Device.

So as described earlier, an FPGA is a matrix of CLBs, each CLB consists in a generic way of a small configurable combinational circuit known as the Look Up Table LUT with a D-type flip-flop DFF where the design varies from manufacturer to the other. **Figure 2.2** shows an example architecture of a CLB [7].

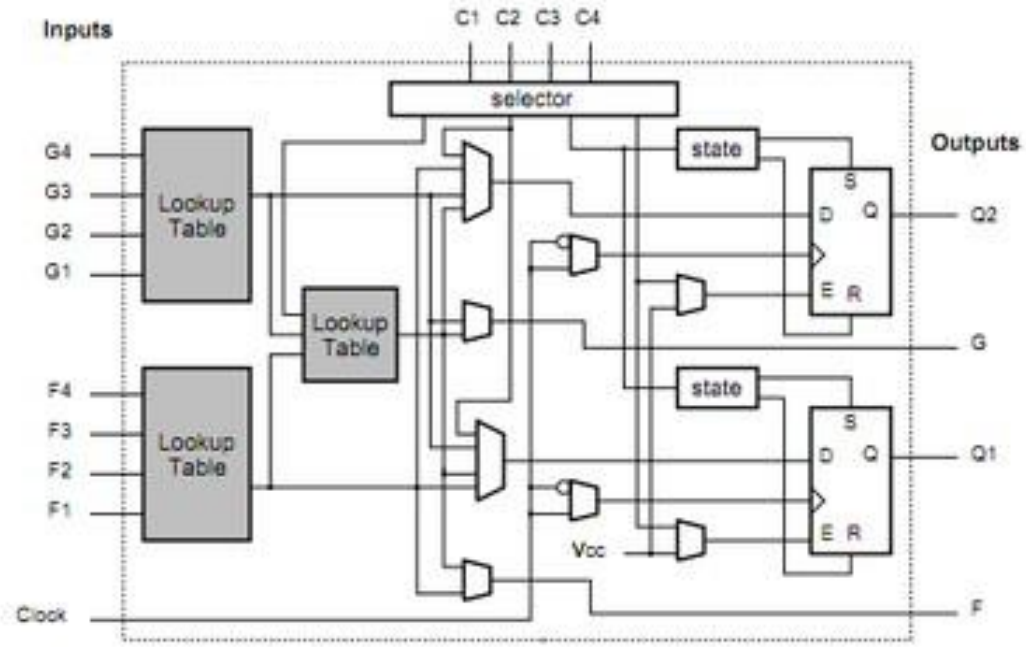

Figure 2.2 Example architecture of a CLB.

Not only are the logic blocks configurable, but the routing matrix, as well as the input and output cells, are also programmable, offering FPGAs great flexibility in terms of meeting the needs of each design [7].

2.11.3 FPGA Advantages

FPGAs present major advantages over early CPLDs and even application specific integrated circuits ASIC. Some of the benefits are detailed in the following points:

- FPGA functionality can change upon every power-up of the device.
- Acceleration: get products to market quicker and/or increase system performance.
- Integration: today's FPGAs include on-die processors, transceiver I/O's at 28 Gbps (or faster), RAM blocks, DSP engines, and more.
- Total Cost of Ownership (TCO): while ASICs may cost less per unit than an equivalent FPGA, building them requires a non-recurring expense (NRE), expensive software tools, specialized design teams, and long manufacturing cycles [8].

2.11.4 Applications

Many applications rely on the parallel execution of identical operations; the ability to configure the FPGA's CLBs into hundreds or thousands of identical processing blocks has applications in image processing, artificial intelligence (AI), data center hardware accelerators, enterprise networking and automotive advanced driver assistance systems (ADAS). Other FPGA uses include aerospace and defense, medical electronics, digital television, consumer electronics, industrial motor control, scientific instruments, cybersecurity systems and wireless communications [9].

2.12 Overview of VHDL Coding

VHDL stands for Very-High-Speed Integrated Circuit Hardware description language. It is an industry standard language used to describe hardware from the abstract to the concrete level. VHDL resulted from work done in the '70s and early '80s by the U.S. Department of Defense [10].

The cooperation of companies such as IBM and Texas Instruments led to the release of VHDL's first version in 1985. Xilinx, which invented the first FPGA in 1984, soon supported VHDL in its products. Since then, VHDL has evolved into a mature language in digital circuit design, simulation, and synthesis [11].

In 1986, VHDL was proposed as an IEEE standard. It went through a number of revisions and changes until it was adopted as the IEEE 1076 standard in December 1987 [10].

VHDL is a programming language that is commonly used to create text models of logic circuits. Only if the model is part of the logic design is it handled by a synthesis software. The logic design is tested using a simulation tool that uses simulation models to represent the logic circuits

that interface with the design. A testbench is a term used to describe a collection of simulation models.

The key advantage of VHDL, when used for systems design, is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires).

Another benefit is that VHDL allows the description of a concurrent system. VHDL is a dataflow language in which every statement is considered for execution simultaneously, unlike procedural computing languages such as BASIC, C, and assembly code, where a sequence of statements is run sequentially one instruction at a time.

VHDL usage has risen rapidly since its inception and is used by literally tens of thousands of engineers around the globe to create sophisticated electronic product; It is a powerful language with numerous language constructs that are capable of describing very complex behavior [10].

2.13 Overview of Altera Cycolne II Devices

The Cyclone II is a high-density, low-cost FPGA from Altera. Following the immensely successful first-generation Cyclone device family, Altera Cyclone II FPGAs extend the low-cost FPGA density range to 68,416 logic elements (LEs) and provide up to 622 usable I/O pins and up to 1.1 Mbytes of embedded memory.

Cyclone II devices can support complex digital systems on a single chip at a cost that rivals that of ASICs. Unlike other FPGA vendors who compromise power consumption and performance for low-cost, Altera's latest generation of low-cost FPGAs - Cyclone II FPGAs - offer 60% higher performance and half the power consumption of competing 90-nm FPGAs. The low cost and optimized feature set of Cyclone II FPGAs make them ideal solutions for a wide array of automotive, consumer, communications, video processing, test and measurement, and other end-market solutions [12]. Figure 2.3 depicts a cyclone II FPGA Chip.

Figure 2.3 Cyclone II FPGA Chip.

Cyclone II devices support the Nios II embedded processor which allows one to implement custom-fit embedded processing solutions. Cyclone II devices can also expand the peripheral set, memory, I/O, or performance of embedded processors. Single or multiple Nios II embedded

processors can be designed into a Cyclone II device to provide additional co-processing power or even replace existing embedded processors in one's system. Using Cyclone II and Nios II together allow for low-cost, high-performance embedded processing solutions, which allow users to extend their product's life cycle and improve time to market over standard product solutions [12]. **Figure 2.4** shows the Cyclone II logic elements block diagram

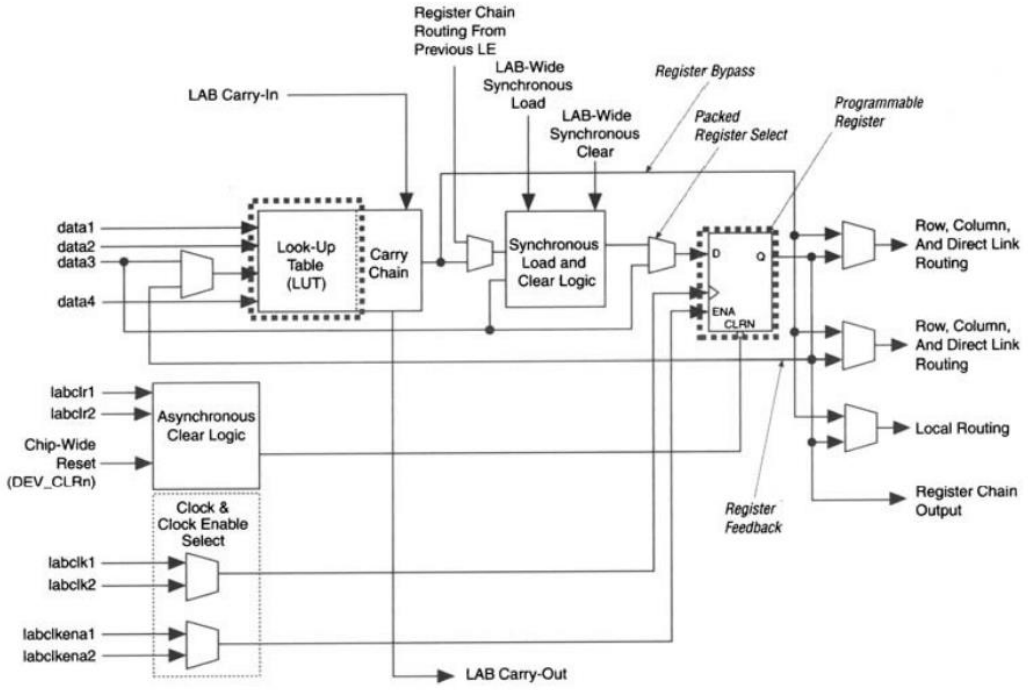

Figure 2.4 Block diagram of a Cyclone II logic element.

2.14 The DE2 Development and Education Board

The Altera DE2 Development and Education board's goal is to make learning about digital logic, computer organization, and FPGAs as easy as possible. It employs cutting-edge hardware and CAD tools to introduce students and professionals to a variety of disciplines. The board has a rich set of features that make it ideal for use in a laboratory setting for university and college courses, as well as for a variety of design projects and the implementation of sophisticated digital systems.

The DE2 board shown in **Figure 2.5** features a state-of-the-art Cyclone® II 2C35 FPGA in a 672-pin package. All important components on the board are connected to pins of this chip, allowing the user to control all aspects of the board's operation. For simple experiments, the DE2 board includes a sufficient number of robust switches (of both toggle and push-button type), LEDs, and 7-segment displays. For more advanced experiments, there are SRAM, SDRAM, and Flash memory chips, as well as a 16 x 2-character display. For experiments that

require a processor and simple I/O interfaces, it is easy to instantiate Altera's Nios II processor and use interface standards such as RS-232 and PS/2 [13].

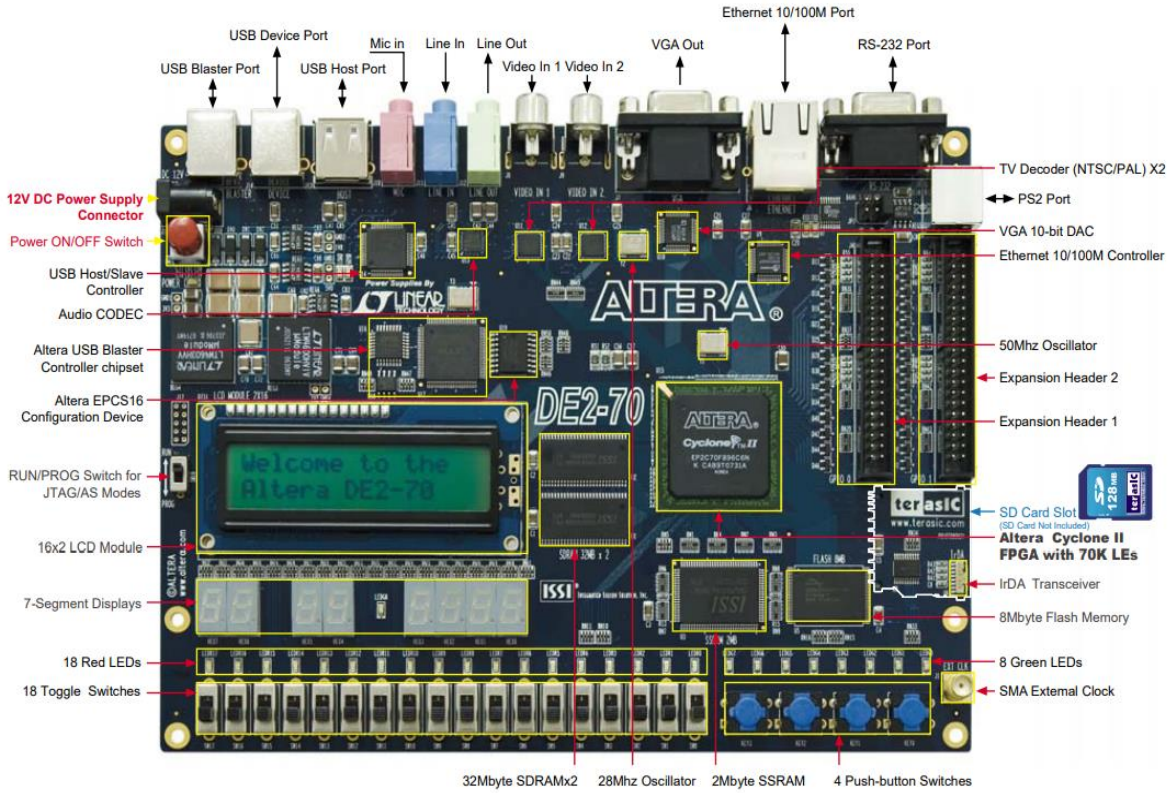

Figure 2.5 DE2 Board.

2.15 Block Diagram of the DE2 Board

Figure 2.6 Shows the block diagram of the DE2 board. To provide maximum flexibility for the user, all connections are made through the Cyclone II FPGA device. Thus, the user can configure the FPGA to implement any system design.

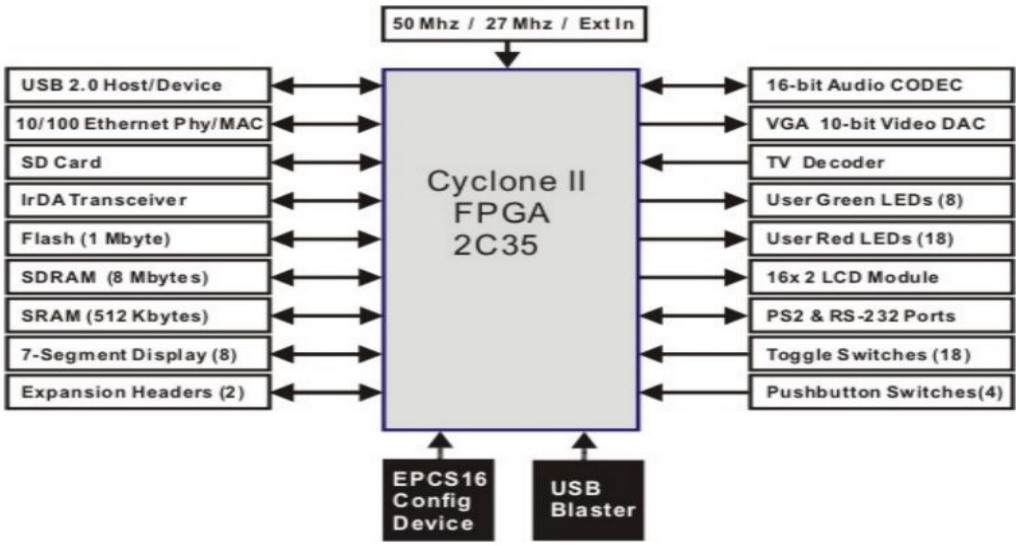

Figure 2.6 DE2 Board Block Diagram.

2.16 Overview of Quartus II Software

The Altera® Quartus® II design software is a multiplatform design environment that easily adapts to specific needs throughout the FPGA design process. For Altera FPGAs, CPLDs, and HardCopy ASICs, Quartus II software provides the optimum productivity and performance.
The Quartus II software provides superior synthesis, placement, and routing, leading in a reduction in compilation time [14].

The Quartus II software has many versions and updates each year, the software version used in this project is the Intel web edition Quartus II 13.0sp1 and it is compatible with the provided DE2 board and cyclone II device family.
Figure 2.7 and **Figure 2.8** represent the Quartus II design flow and the graphical user interface GUI is provided for the user to use throughout the entire design flow as they can use different options at different stage [14].

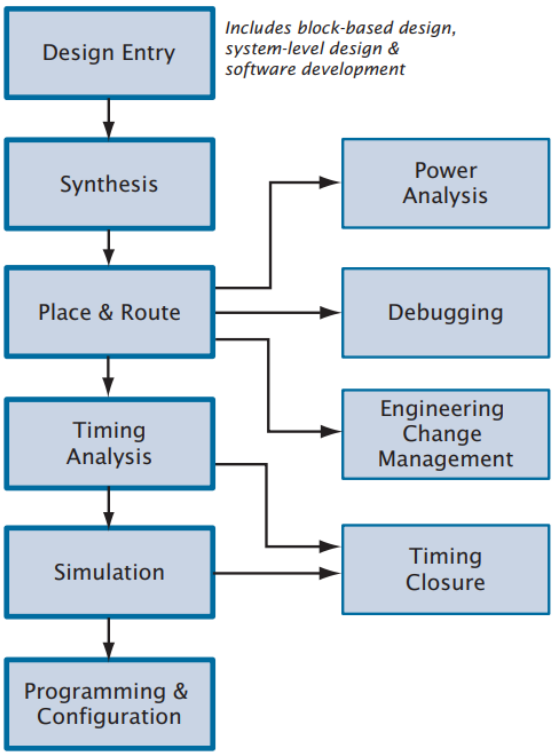

Figure 2.7 Quartus II Design Flow.

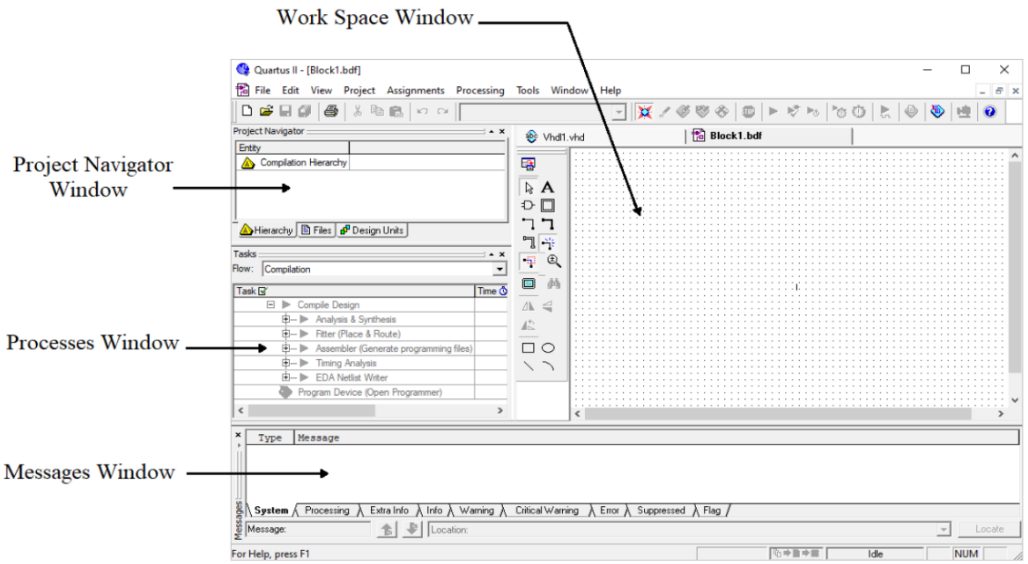

Figure 2.8 Quartus Graphical User Interface.

2.17 Qsys integration tool

The Qsys tool is used in conjunction with the Quartus II software, it is used to design digital hardware systems that contain components such as processors, memories, input/output interfaces, timers, and the like. The Qsys tool allows a designer to choose the components that are desired in the system by selecting these components in a graphical user interface and customize their parameters according to the user's specific design requirements. It then automatically generates the hardware system that connects all of the components together [15]
Figure 2.9 Represents the Graphical user interface in the Qsys tool

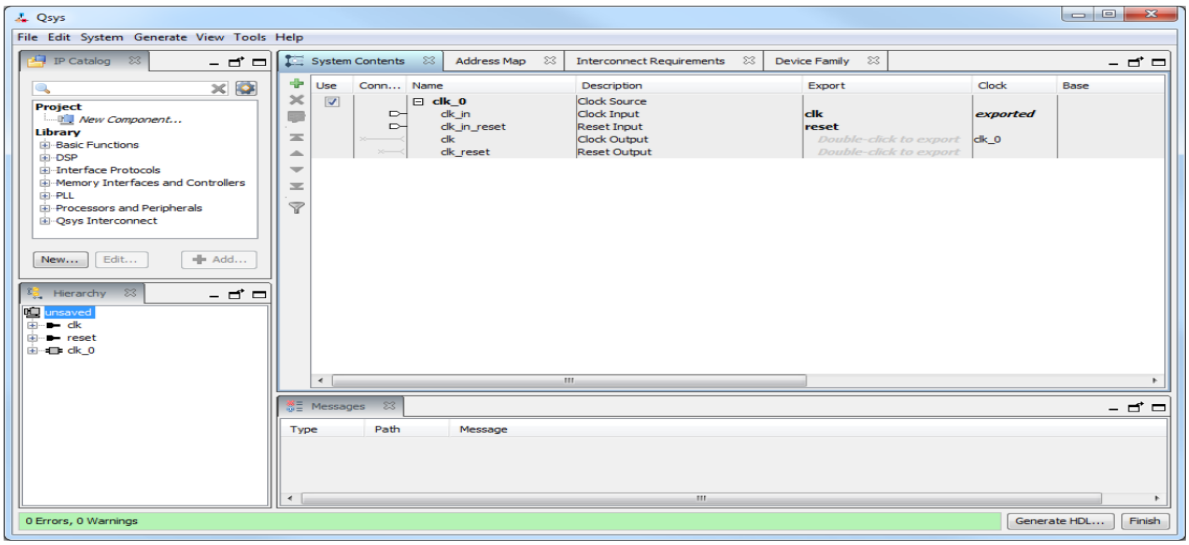

Figure 2.9 Qsys Graphical User Interface.

The Qsys system integration tool saves significant time and effort in the FPGA design process by automatically generating interconnect logic to connect custom HDL design blocks, commonly referred to as design modules; intellectual property (IP) cores; and components. Qsys is powered by a new FPGA-optimized NoC–based technology delivering higher performance. Qsys also raises the level of abstraction and increases productivity by enabling design reuse and scalable systems [14].
Figure 2.10 is a Qsys Interconnect Fabric Example.

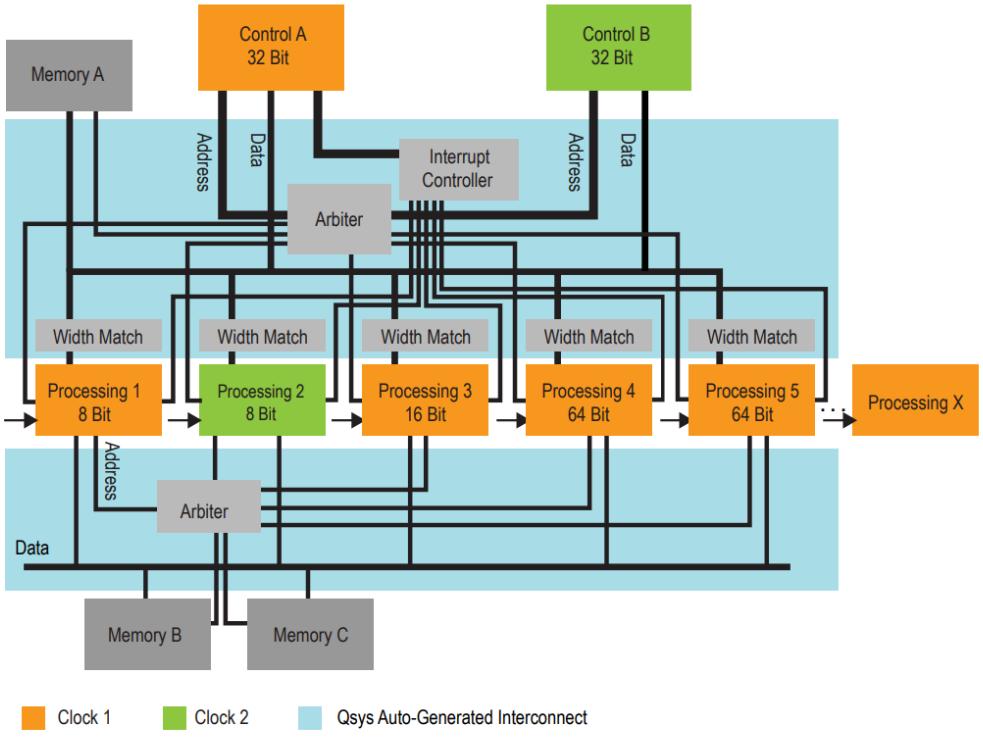

Figure 2.10 Qsys Interconnect Fabric Example.

2.18 NIOS II

2.18.2 Embedded SOPC design

A system on a programmable chip is referred to a system that integrates different hardware components into a single chip, these systems have the capability of being modified according to the requirements of the application they are designed for. These systems can be implemented on an FPGA using either SOPC builder or in the case of this project, Qsys tool.
An SOPC design's central component is called a soft-core processor, it is a microprocessor fully described in software, usually in HDL, which can be synthesized in programmable hardware, such as FPGAs. A soft-core processor targeting FPGAs is flexible because its parameters can be changed at any time by reprogramming the device [16].

We can tailor the processor, select only the needed I/O peripherals, create a custom I/O interface, and develop specialized hardware to match specific needs [5].

2.18.3 NIOS II processor

The Nios II is a 32-bit Reduced instruction set computing RISC embedded processor architecture designed specifically for the Altera family of FPGA integrated circuits [17].

The Nios II processor and the interfaces needed to connect to other chips on the DE2 board are implemented in the Cyclone II FPGA chip. These components are interconnected by means of the interconnection network called the Avalon Switch Fabric. Memory blocks in the Cyclone II device can be used to provide an on-chip memory for the Nios II processor. They can be connected to the processor either directly or through the Avalon network.

The processor can be implemented in three different configurations:

• Nios II/f is a "fast" version designed for superior performance. It has the widest scope of configuration options that can be used to optimize the processor for performance.
• Nios II/s is a "standard" version that requires less resources in an FPGA device as a trade-off for reduced performance.
• Nios II/e is an "economy" version which requires the least amount of FPGA resources, but also has the most limited set of user-configurable features.

The Nios II software development environment is called The Nios II integrated development environment (IDE). It is based on the GNU (GNU's not unix) C/C++ compiler and the Eclipse IDE and provides a familiar and established environment for software development in addition to a command line interface [18].

2.19 Android OS

2.19.2 Overview

Android is an open-source mobile operating system that was created primarily for touchscreen mobile devices like smartphones and tablets. It is based on a modified version of the Linux kernel. It was developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google [19].

The first Android smartphone was released in the market in September 2008, it quickly took over the mobile industry thanks to features like user-friendliness, widespread community support, customizability, and widespread production of Android devices by big businesses.
Android is a powerful operating system and it supports a large number of applications on Smartphones. These applications are more comfortable and advanced for users. The hardware that supports android software is based on the Advanced RISC Machine ARM architecture platform [20]. **Figure 2.11** shows Android's most recent Logo

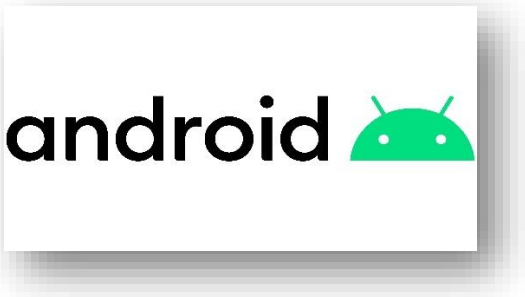

Figure 2.11 Android OS Logo.

2.19.3 Kodular App inventor

2.19.3.2MIT App inventor

MIT App Inventor is a web application integrated development environment originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT) [21].
It is an intuitive, visual block-based programming environment to build fully functional applications compatible for android phones and tablets. The blocks-based tool facilitates the creation of simple to complex high-impact apps in less time than traditional programming environments.
For the front end, App inventor allows users to drag and drop visual objects and edit them while an App-inventor companion can be connected to the development environment for testing purposes [22].

2.19.3.3Kodular

Kodular tool was launched in June 2018, it is in simple terms a baked version of MIT App inventor with an improved and more modern user experience [23]. **Figure 2.12** Shows part of the graphical user interface for the development of applications.

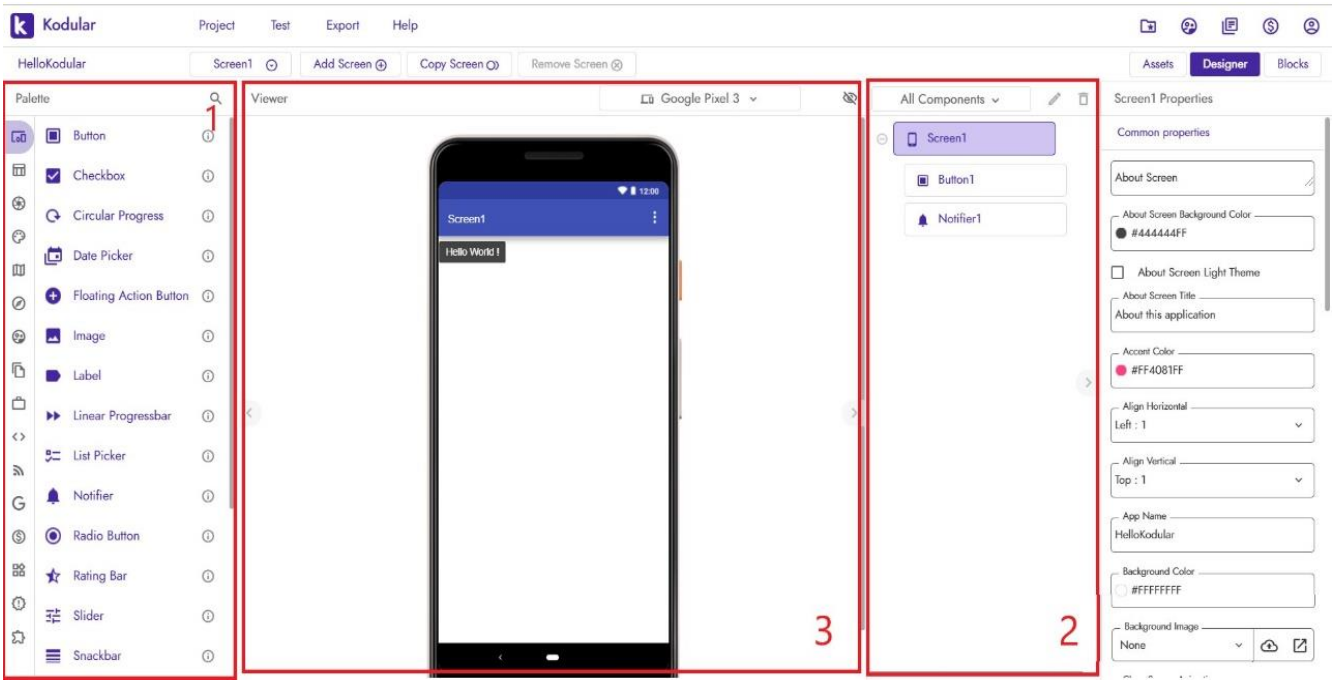

Figure 2.12 View of the Kodular Designer Workplace Window.

2.20 Bluetooth Wireless Technology

2.20.2 Overview

Bluetooth is an omnidirectional wireless technology that provides limited-range data transmission over the unlicensed 2.4-GHz frequency band, allowing connections with a wide variety of fixed and portable devices that normally would have to be cabled together. Up to eight devices can-one master and seven slaves- can communicate with one another in a so called piconet at distances of up to 30 feet [24].

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 35,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard.
Since its development in 1994 by the Swedish telecommunications firm Ericsson, more than 35000 companies worldwide have signed on as members of the Bluetooth SIG to build products to the wireless specification and promote the new technology in the marketplace [25]. **Figure 2.13** Shows the Bluetooth logo

Figure 2.13 Bluetooth Wireless Logo.

2.20.3 HC-05 Bluetooth Module

HC-05 Bluetooth Module is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with microcontrollers.

The communication is done via a Universal Asynchronous Receiver Transmitter UART interface which is a physical circuit in a microcontroller used to receive and transmit data.
Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband [26].
Figure 2.14 below show the HC-05 Bluetooth module

Figure 2.14 HC-05 Bluetooth Module.

2.21 Analog to Digital Converter

2.21.2 Overview

An analog-to-digital converter is any device that converts analog signals (continuous quantity) into digital signals (discrete time digital representation). The analog signal is a continuous sinusoidal wave form that cannot be read by a computer, hence the need for conversion. By converting the analog signal, data can be amplified, added or taken from the original signal [27].

Analog to Digital Converter samples the analog signal on each falling or rising edge of the sample clock. In each cycle, the ADC gets the analog signal, measures it, and converts it into a digital value. The ADC converts the output data into a series of digital values by approximates the signal with fixed precision.

Figure 2.15 below figure depicts how analog to digital conversion takes place. Bit rate decides the resolution of digitized output and you one can observe in the below figure where a 3-bit ADC is used for converting the analog signal [27].

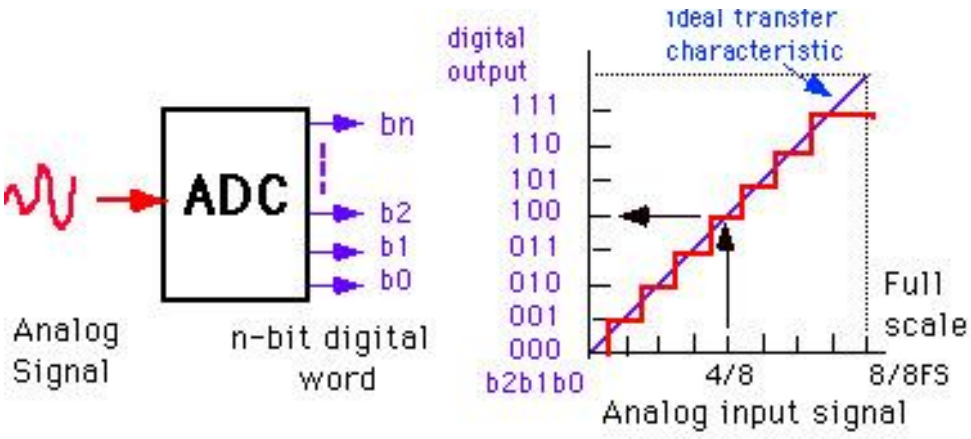

Figure 2.15 Analog to Digital Conversion Process.

ADC Integrated circuits ICs differ from each other by means of bit resolution which is the number of discrete digital values it produces and the step size which is the change in analog input to cause a unit change in the output of ADC.

2.21.3 ADC0808 IC

ADC0808 data acquisition component is a monolithic CMOS device with an 8- bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The ADC0808 offers high speed, high accuracy and minimum power consumption. The input channel from which to convert data can be selected using a three-input address line and voltage references can also be set through Vref- and Vref+ pins. The step size is by default set to 19.53mv corresponding to a 5v reference voltage [28]. **Figure 2.16** shows the block diagram of the ADC0808 and **Figure 2.17** shows its corresponding pin assignments diagram.

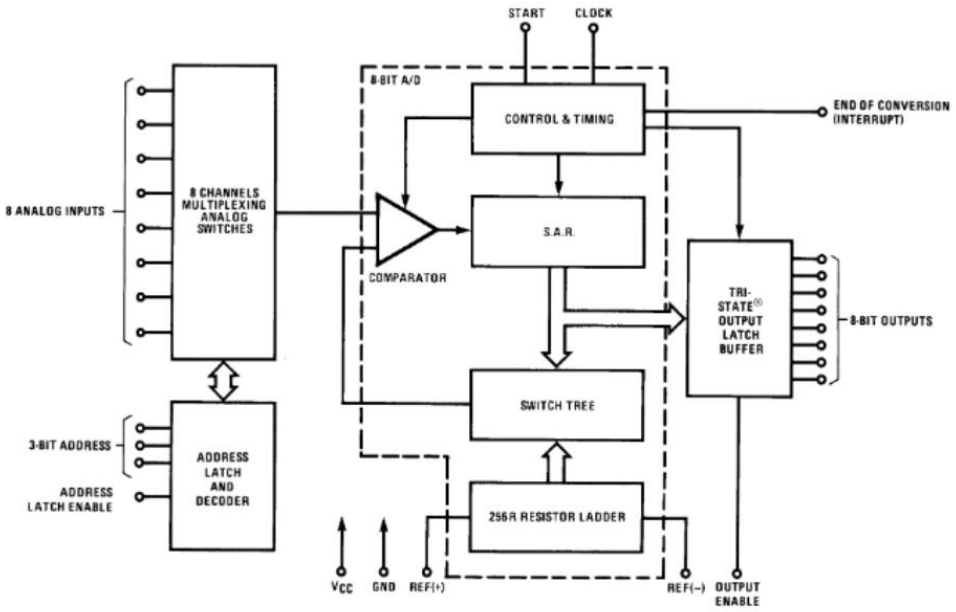

Figure 2.16 Block Diagram of the ADC0808.

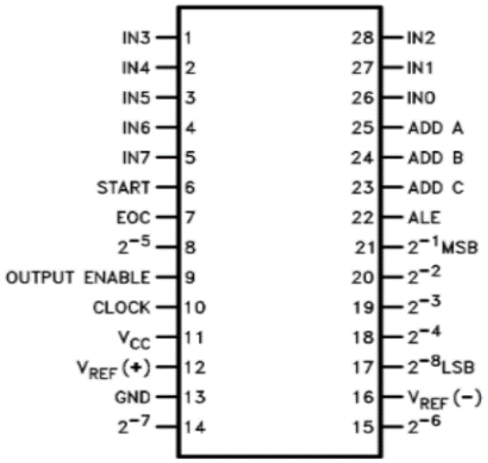

Figure 2.17 Pin Assignments of the ADC0808.

2.22 DC Motor

2.22.2 Overview

The electric motor in its simplest terms is a converter of electrical energy to useful mechanical energy. The electric motor has played a leading role in the high productivity of modern industry An electric motor's principle of operation is based on the fact that a current-carrying conductor, when placed in a magnetic field, will have a force exerted on the conductor proportional to the current flowing in the conductor and to the strength of the magnetic field [29].

Figure2.18 Typical Small DC Motor

2.23 H-bridge

A H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are used to control DC motors to run forwards or backwards [30].

H-bridges are available as integrated circuits, or can be built from discrete components

The term H-bridge is derived from the typical graphical representation of such a circuit. An H-bridge is built with four switches as shown in **Figure 2.19** (solid-state or mechanical). When the switches S1 and S4 (according to the first figure) are closed (and S2 and S3 are open) a positive voltage is applied across the motor. By opening S1 and S4 switches and closing S2 and S3 switches, this voltage is reversed, allowing reverse operation of the motor.

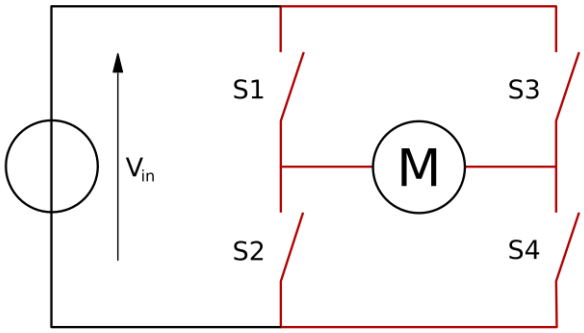

Figure 2.19 Structure of an H Bridge.

In this project, the L293D integrated circuit motor driver was used which is a quadruple high-current half-H driver designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5v to 36v [31].

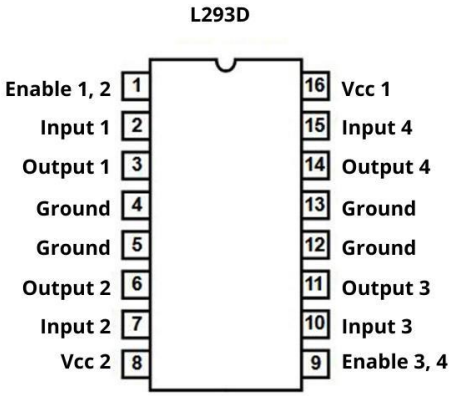

Figure 2.20 Pin Diagram of the L293D Motor Driver.

2.24 Octal buffer

A buffer IC provides electrical impedance transformation from one circuit to another, with the aim of preventing the signal source from being affected by whatever currents (or voltages, for a current buffer) that the load may be produced with. The signal is 'buffered from' load currents [32].

In this project, a SN74LS244 was used which is a bidirectional octal buffer designed to be employed as a memory address driver, clock driver and bus-oriented transmitter/receiver that provides improved PC board density [33]
The Truth Table for the SL74LS244 is shown below.

SN54/74LS244

INPUTS		OUTPUT
$\overline{1G}$, $\overline{2G}$	D	
L	L	L
L	H	H
H	X	(Z)

Table 2.1 Truth Table Illustration of the Octal Buffer.

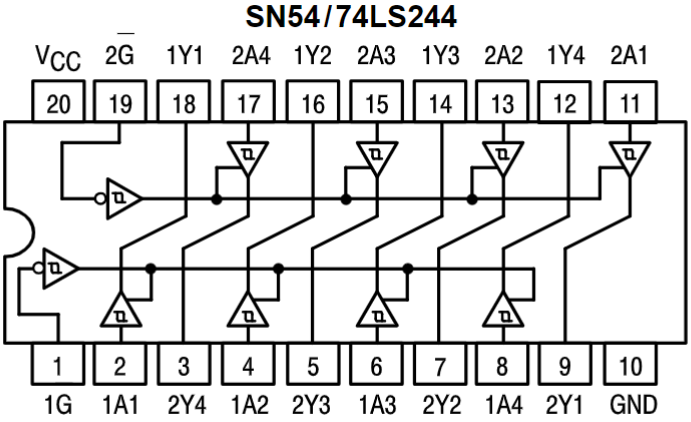

Figure 2.21 The 74LS244 Pin Diagram.

2.25 BC546B Transistor

The BC546 is a high frequency NPN Transistor commonly used in low noise amplifier designs. It has a maximum gain value of 800 and a high collector to emitter voltage of 65V making it also suitable for high voltage audio amplifier applications [34].

In this project, it is used as a switching component. When this transistor is fully biased, it can allow a maximum of 100mA to flow across the collector and emitter, this stage is called Saturation Region**.** When base current is removed the transistor becomes fully off, this stage is called the Cut-off Region. **Figure 2.22** shows a diagram of this transistor.

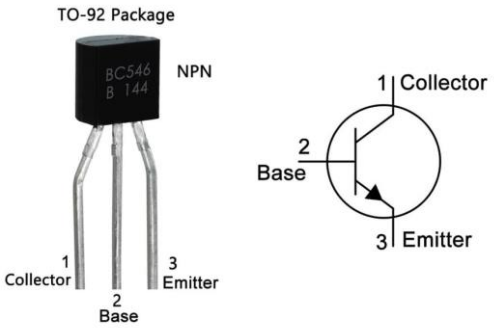

Figure 2.22 BC546B Transistor.

2.26 Buzzer

a buzzer is a type of electronic device that's used to produce a tone, alarm or sound, it can be mechanical, electromechanical, or piezoelectric. When certain piezoelectric materials are subjected to an alternating field of electricity, the piezo buzzer element — often a manmade piezoceramic material — stretches and compresses in sequence with the frequency of the current. As a result, it produces an audible sound [35].

Figure 2.23 Piezoelectric Buzzer.

2.27 Soil Moisture Sensor

Soil moisture sensor measures the volumetric water content in soil. Since the direct gravimetric measurement of free soil moisture requires removing, drying, and weighing of a sample, soil moisture sensors measure the volumetric water content indirectly by using some other property of the soil, such as electrical resistance, dielectric constant, or interaction with neutrons

Features:

- Soil moisture sensor based on soil resistivity measurement
- Easy to use
- 2.0cmX6.0cm grove module [36].

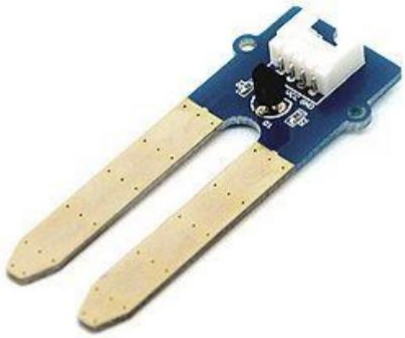

Figure 2.24 Soil Moisture Sensor

2.28 Rain Sensor

Rain sensor module is an easy tool for rain detection. It can be used as a switch when raindrops fall through the raining board and also for measuring rainfall intensity [37]. The raindrop sensors can be used in the automobile sector to control the windshield wipers automatically, in the agriculture sector to sense rain and it is also used in home automation systems**. Figure 2.25** shows a rain drop sensor module.

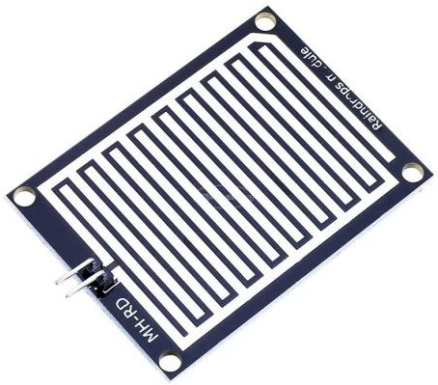

Figure 2.25 Rain Drop Sensor Module.

2.29 Light Sensor

Light Sensors are photoelectric devices that convert light energy (photons) whether visible or infra-red light into an electrical (electrons) signal. It is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material [38]. So an LDR as in **Figure 2.26** acts like a variable resistor whose value decreases with increasing incident light intensity.

Figure 2.26 Light Dependent Resistor

2.30 Temperature Sensor

LM35 is a temperature measuring device having an analog output voltage proportional to the temperature. It provides output voltage in Centigrade (Celsius) and does not require any external calibration circuitry. The sensitivity of LM35 is 10 mV/degree Celsius. As temperature increases, output voltage also increases. As shows in **Figure 2.27** It is a 3-terminal sensor used to measure surrounding temperature ranging from -55 °C to 150 °C [40].

Figure 2.27 LM35 Temperature Sensor

2.31 Gas Sensor

The MQ2 Gas sensor is a Metal Oxide Semiconductor (MOS) type Gas Sensor mainly used to detect gases like Methane**,** Butane**,** LPG**,** Smoke**,** etc. It is also known as a Chemiresistor as the gas detection is based on the change of resistance of the sensing material when the Gas comes in to contact. MQ2 Gas sensor module works on 5V DC and uses around 800mW. It can detect gases within a concentrations range of 200 to 10000 ppm.

MQ2 Gas Sensor shown in **Figure 2.28** has the following features:

- Operating Voltage is +5V
- Analog output voltage: 0V to 5V
- Digital Output Voltage: 0V or 5V (TTL Logic)
- Preheat duration 20 seconds
- Can be used as a Digital or analog sensor
- The Sensitivity of Digital pin can be varied using the potentiometer [40].

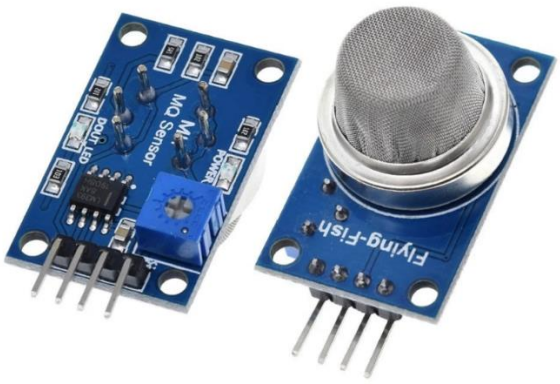

Figure 2.28 MQ2 Gas Sensor.

Chapter 03

Hardware System Design

3.1 Overview

The hardware system design of this project constitutes of two major parts, the Off-chip and the On-chip system design.
The Off-chip hardware system design consists of the external circuit that is implemented in a breadboard, in includes different sensors, the data acquisition unit, buffers, the motors driving units and the communication unit that is composed of a Bluetooth module along the android application. The whole circuit design is connected to the FPGA through GPIO header 01.

The On-chip hardware system design consists of two parts, the SOPC system and the non-SOPC system. The SOPC system is implemented using Qsys tool on Quartus and it includes the Nios II microprocessor, the on-chip memory, the JTAG UART to communicate with the laptop and the Bluetooth UART. The non-SOPC system consists of different custom blocks developed with VHDL and that serve a certain control functionality. **Figure 3.1** shows a diagram of the overall hardware system.

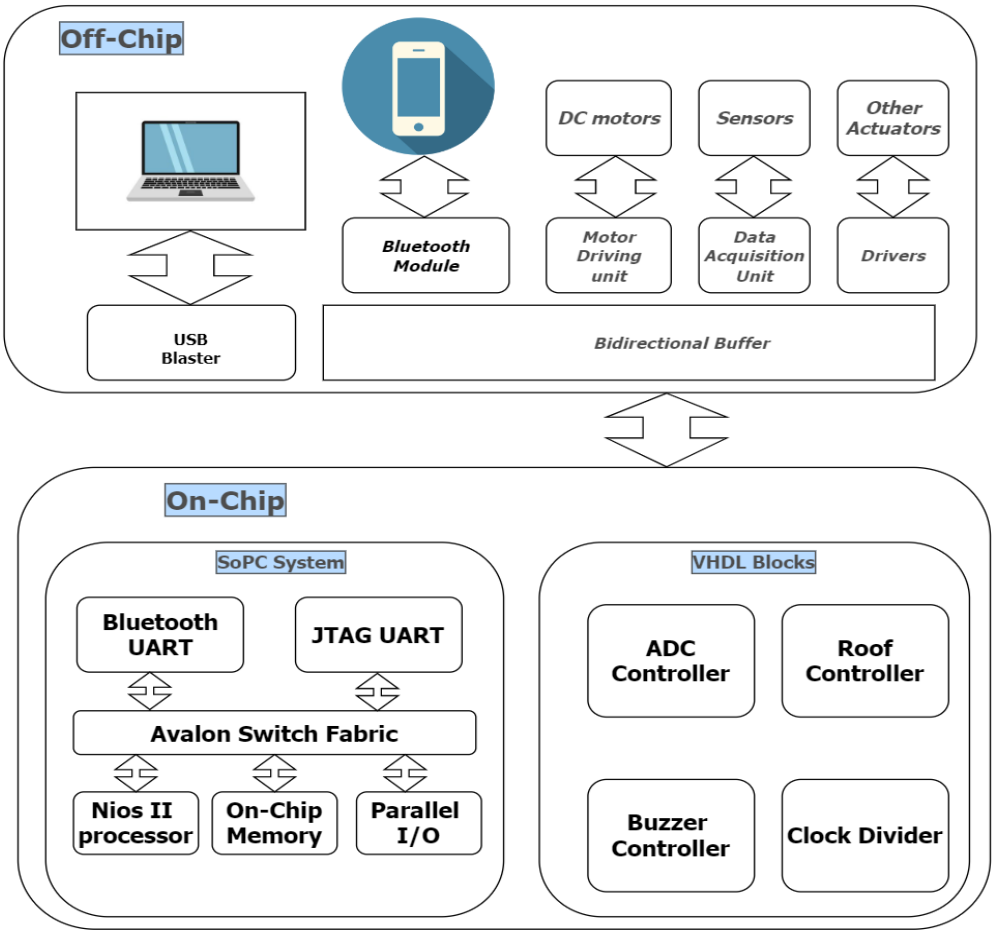

Figure 3.1 Overall System Block Diagram.

3.2 The Off-chip Hardware Design

The principal of operation of the environment controller is to automatically control physical/environmental parameters inside a greenhouse to provide a climate where crops can live according to their specific needs of water, temperature, cooling and light. The Off-chip design contains the different sensors connected to the analog-to-digital converter and transmitted over the DE2 board header to the Nios II to be processed. In addition to that, we have the different motor driving units that will control the roof, irrigation pumps and fan as well as a gas sensor that would lunch an alarm in case of a fire or smoke detection. Finally, we have a Bluetooth module that serially communicates with the FPGA and sends data to the android app for it to be displayed.

3.2.1 Data Acquisition Unit

In order to convert the sensor readings from their Analog original format to digital values suitable for processing by the microprocessor, an ADC0808 IC has been used for this purpose. It is an 8-bit 8-channels Analog-to-digital converter that uses successive approximation as a conversion technique. The sensors are connected to the input of the ADC and for proper data conversion, the ADC requires several control signals; a clock signal that has been set to 1Mhz, START and ALE signals that are generated by the ADC controller block and sent to the ADC chip, End-of-conversion signal EOC that is generated by the ADC0808 and sent to the ADC controller and three address signals A, B and C to select the input channel whose data is to be converted from.
After conversion, data is sent out to the FPGA ADC controller block to be organized in different buffers and further sent to the processing unit.
Figure 3.2 illustrates the circuit diagram of the data acquisition unit interfaced with the ADC controller block on the FPGA and that shall be described later on in this report.

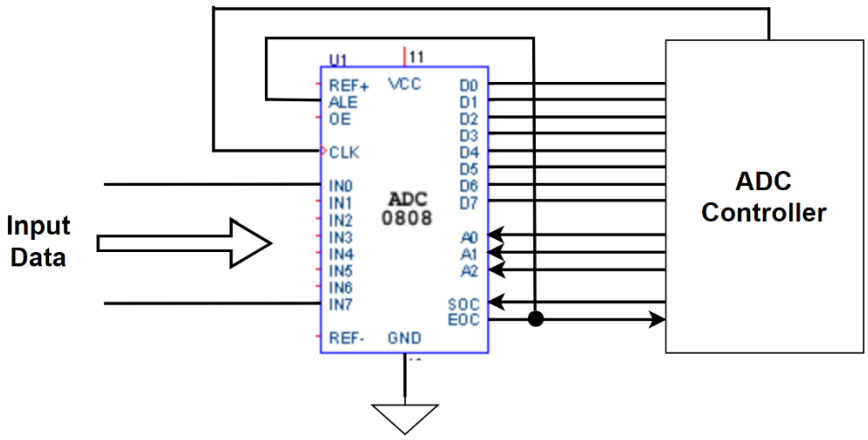

Figure 3.2 Data Acquisition Unit.

3.2.2 Motor Driving Unit

The motor driving unit is used to drive DC motors of the Fan, irrigation pumps and the roof. The fan and irrigation pumps do not require bidirectional rotation and thus only require an amplifier circuit whereas the roof requires rotation in both clock-wise and anti-clockwise directions and thus requires an H-Bridge for direction control.
The system takes automatic control of these motors based on the data coming from the sensors

3.2.2.1 Roof Driving Unit

The roof in our greenhouse prototype can be opened or closed and is then controlled using a DC motor interfaced with a transistor-based H-bridge that allows it to rotate in both directions On the edge of the roof, there are two limit switches in opposite site that are used to indicate to us whether the opening or closing of the roof has reached its maximum or not.
The roof controller block controls the signals that are sent to the H-bridge and they are generated based on the data coming from the soil moisture and rain drop sensors; if the crops inside the greenhouse are in need of water, we check whether there is rain to open the roof or keep it closed. **Figure 3.3** shows the roof driving unit diagram.

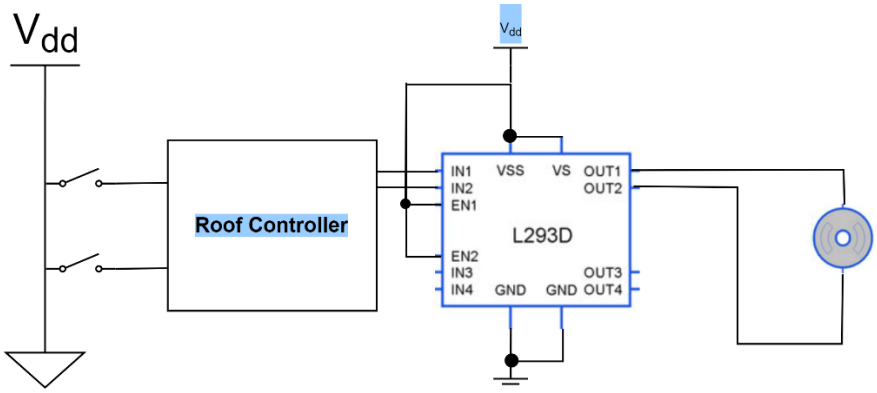

Figure 3.3 Roof Driving Unit.

3.2.2.2 Fan/Pumps driving unit

The Fan is turned On or Off based on the temperature sensor readings; when the temperature exceeds a certain set point, the Fan is turned On to allow cool air to enter inside the greenhouse. On the other hand, the irrigation pumps are turned On and Off based on the soil moisture and rain drop sensor readings; when crops are in need of water and there is no rain, we turn On the water pumps.
Both the fan and pumps use a DC motor that rotates in one direction only so there is no need to use an H-bridge; a tri-state octal buffer is placed as a first stage component to hold the control signals sent from the FPGA and are then outputted to the base of a transistor to obtain the appropriate current to drive the DC motors.

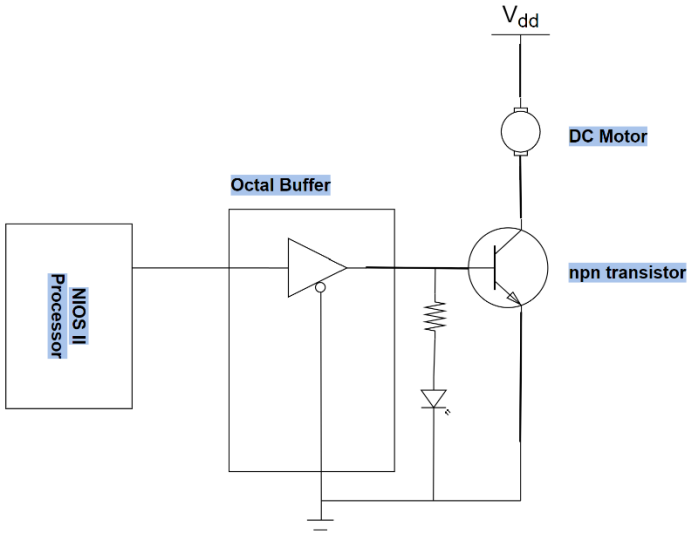

Figure 3.4 Fan/pumps Driving Unit.

3.2.3 Light Driving Unit

When the crops inside the greenhouse require light and there is not enough light outside due to different reasons, the system will turn On artificial light.
To drive the lights unit, a npn transistor was used alongside a tri-state buffer. **Figure 3.5** below shows a circuit diagram of the lights driving unit.

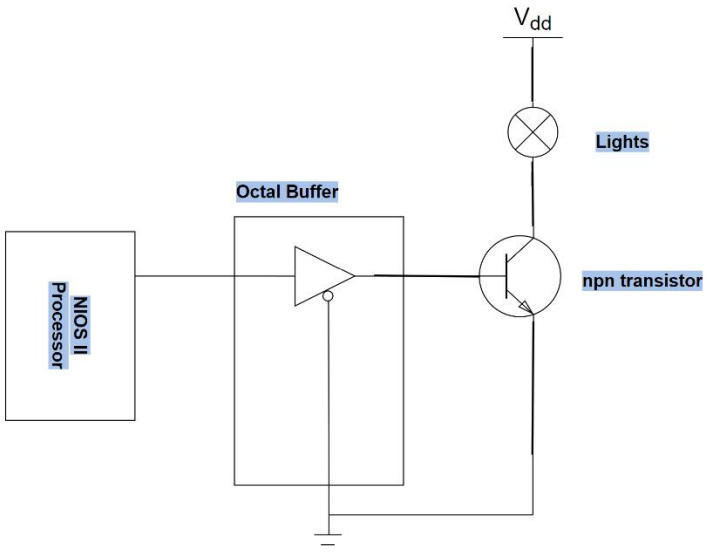

Figure 3.5 Light Driving Unit.

3.2.4 Smoke Sensing Unit

A gas sensor is placed inside the greenhouse for protection purposes; in case of a fire, smoke or methane gas detection, the gas sensor will send a digital signal to the SOPC system and would automatically lunch an alarm and alert the owner on the android application.

3.2.5 Alarm Driving Unit

The alarm unit is implemented using a piezo buzzer that is interfaced with a control block that tunes it at an appropriate frequency. The piezo buzzer can be used to create sound waves when provided with an electrical signal. In order to produce a tone, it is driven by a transistor and it requires a square wave whose frequency will determines the pitch of the generated sound.
Figure 3.6 Shows the Alarm Driving Unit.

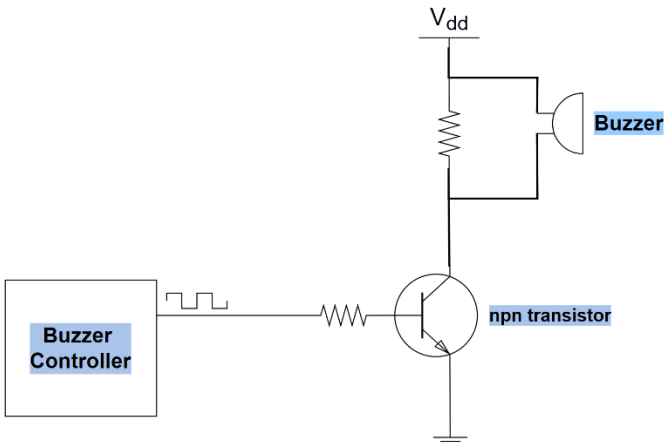

Figure 3.6 Alarm Driving Unit.

3.2.6 Communication Unit

3.2.6.1 Overview

The communication unit allows different devices and components to communicate between each other.
The Off-chip hardware design is connected to the SOPC system in the DE2 board by means of the JP1 expansion header and the DE2 board is connected to the laptop through a USB blaster port while the HC-05 Bluetooth module is the linking device between the Android application and the greenhouse. It uses the UART as a communication protocol to send and receive data.
The laptop is used to run the software and implement On-chip hardware parts of the system while the smartphone runs the android application that serves as a means for data visualization and monitoring.

3.2.6.2 UART Protocol

A universal asynchronous receiver transmitter is the most common protocol used for full-duplex serial communication. It is a hardware device on a single chip designed to perform asynchronous communication where it sends and receives data from one system to another by converting bytes of source data into individual bits in a sequential fashion [41].

3.2.6.3 RS-232 Serial Communication Port

The RS232 UART Core implements a method for communication of serial data. The core provides a simple register mapped Avalon interface. Master peripherals (such as a Nios II processor) communicate with the core by reading and writing control and data registers [42].

The RS232 UART core can be instantiated in a system using Qsys and in this project, it allows connection between the SOPC system on the DE2 board and the HC-05 Bluetooth module through two external pins, Tx and Rx. Tx pin is used for data transmission and Rx pin is used for data reception. The HC-05 uses a baud rate defined at 9600 Bps and the UART core can be adjusted to meet the latter specification. **Figure 3.7** shows the overall off-chip hardware system.

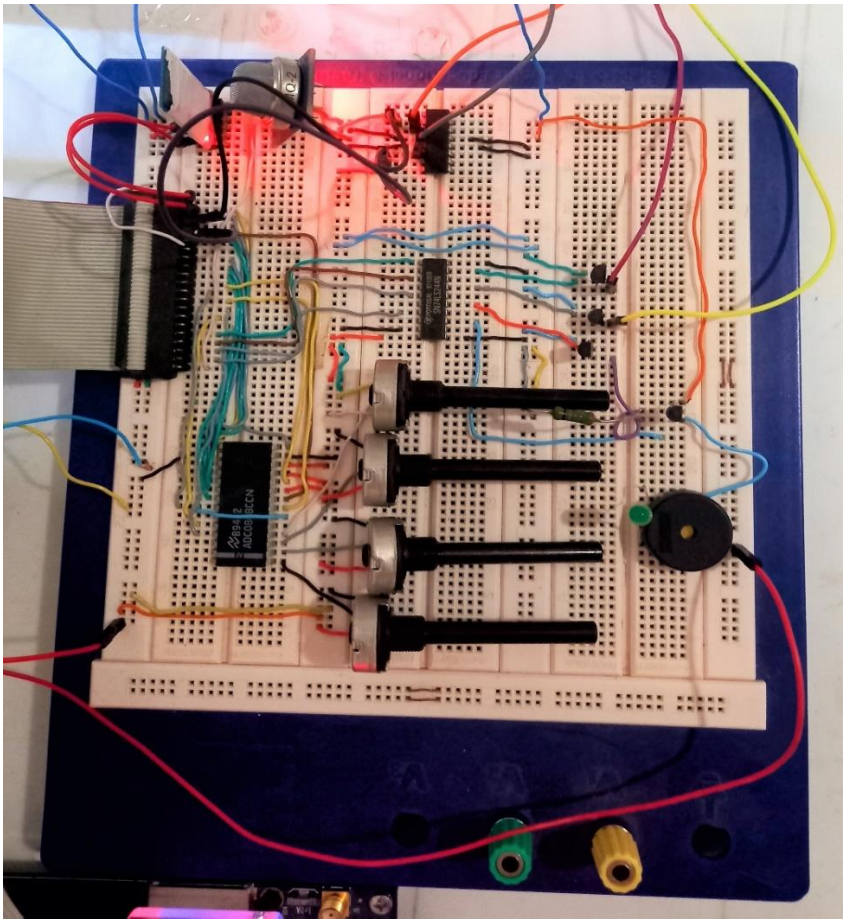

Figure3.7 Overall Off-chip Hardware System.

3.3 The On-Chip Hardware System

The On-chip hardware system of this project has been implemented on the Cyclone II EP2C35F672C6 FPGA available on the DE2 development board and is divided into two parts: The SOPC System and the Custom VHDL Blocks.

3.3.1 SOPC System Implementation

The SOPC System has been implemented using Qsys integration tool of the Quartus II IDE
The following components have been selected and properly connected

- Clock source
- Nios II processor
- On chip memory
- I/O peripherals: Input and output data and control signals interfaced with the custom VHDL blocks in the non-SoPC system

    - Ldr_rain_soil_temperature: 32-bit input port coming from a custom VHDL control block
    - Gas_iout_port: 1-bit input port interfaced with the gas sensor
    - Fan: 1-bit output port interfaced with the fan's DC motor
    - Roof: 2-bit output port interfaced with a VHDL control block
    - Pumps: 1-bit output port interfaced with the pumps' DC motor
    - Lights: 1-bit output port interfaced with the greenhouse's interior lamp
    - Alarm: 1-bit output port interfaced with the buzzer control block

- JTAG UART port to exchange data serially with the laptop
- Bluetooth UART port to communicate with the HC-05 Bluetooth module
- System ID is added to maintain the consistency between the hardware configuration file and software image.

Figure 3.8 shows the overall SoPC components.

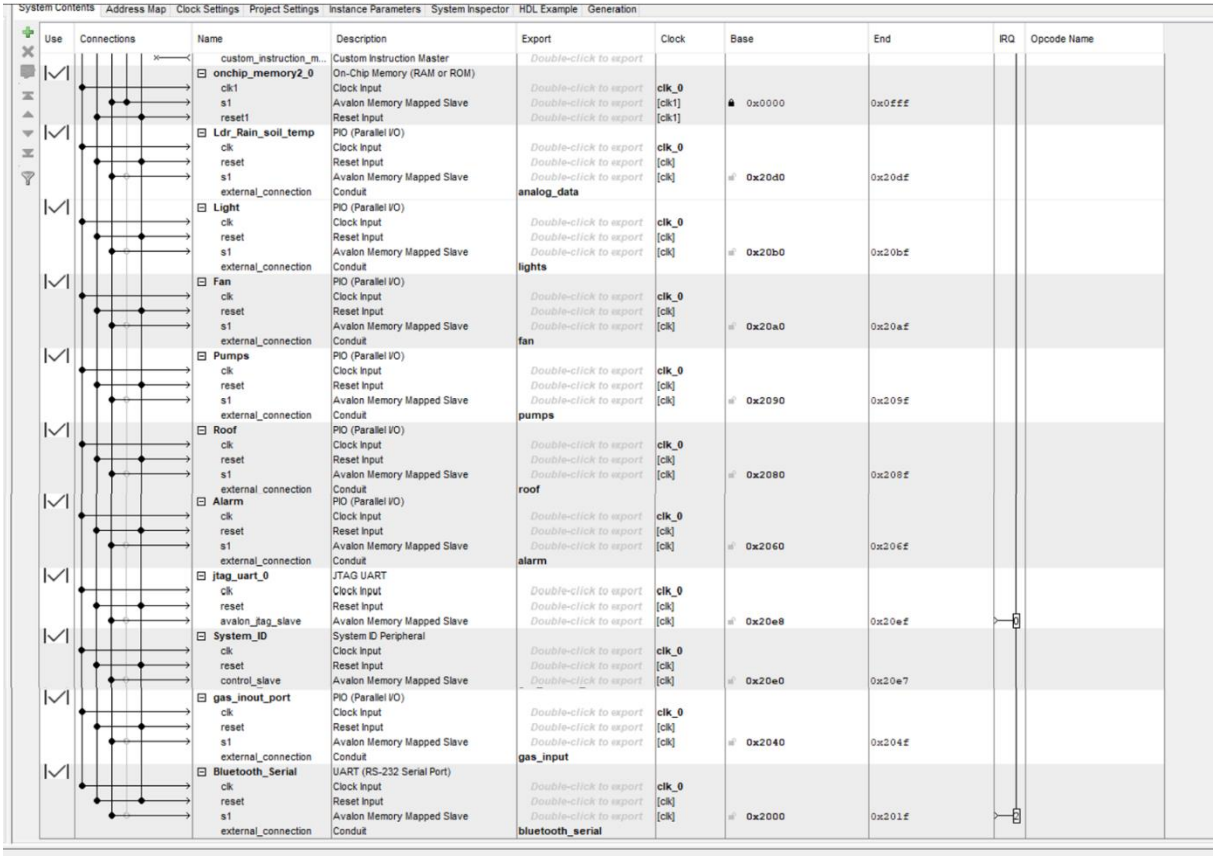

Figure 3.8 Overall SOPC Components.

The SOPC system has been generated and at this point it was possible for it to be instantiated as a block diagram from Quartus II as shows in **Figure 3.9**.

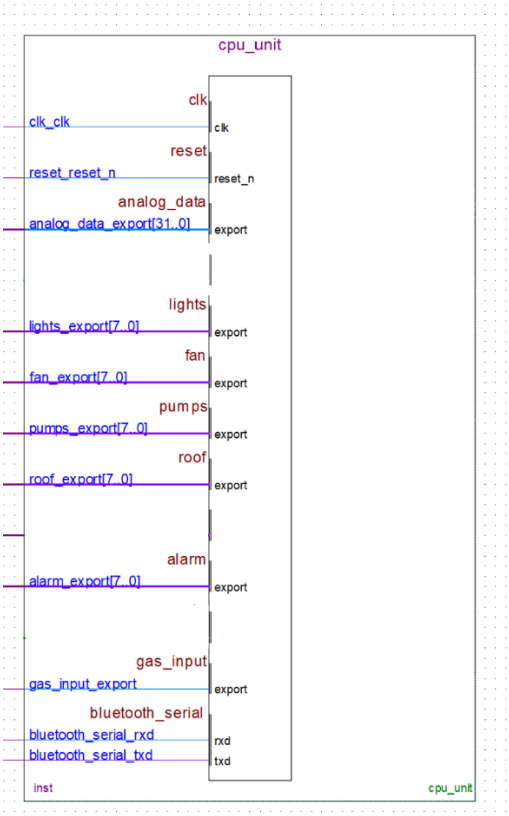

Figure 3.9 SoPC Block Diagram.

3.3.2 Custom VHDL Blocks

Some of the overall system functionalities require special handling which is why control blocks are needed. They are custom blocks developed using VHDL and instantiated as block diagrams in the overall system design where they serve as an interfaced between the SoPC block peripherals and GPIO pins of the DE2 board.

These VHDL blocks involve a clock divider, the ADC controller block, The Roof controller, and the buzzer controller.

3.3.2.1 ADC Controller Block

The ADC0808 chip has special control signals that need to be properly handled. There is a total of six signals that need to be sent from the FPGA to the ADC chip: three address signals to choose the channel from which data is converted from, a clock signal that needs to be tuned at 1Mhz frequency, ALE and START signals.
The START and ALE signals should be pulsed for at least 100ns in order to clear internal registers of the ADC, load proper addresses and start conversion.

In order to tune the ADC to a proper frequency, we have developed a frequency divider in order to divide the on-board clock source of 50Mhz to 1Mhz and send it out to the ADC chip.

The ADC controller block works as an FSM having four states:

Reset state: when reset switch is high, FSM goes to reset state where it sets the address lines to the corresponding channel address, sets Start signal to low and at the next rising edge of the clock, it moves to the next state.
Conversion start: the controller sets the Start signal to high and moves to the next state
Converting state: the controller sets the Start signal to low and moves to the next state
End-of-conversion: controller fetches converted data and saves it in its corresponding buffer, increments the address line signal to read from the next channel and goes back to the reset state.
Figure 3.10 Show the block diagram of the ADC controller block

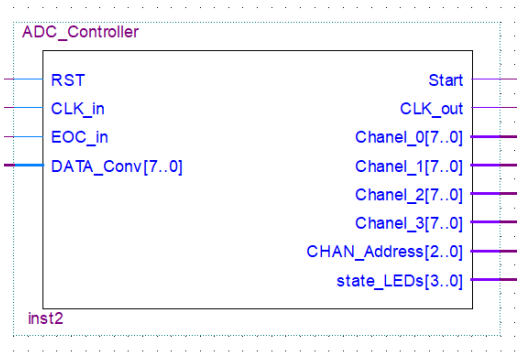

Figure 3.10 ADC Controller Diagram

3.3.2.2 Roof Controller

The roof controller shown in **Figure 3.11** is an interface between the Nios II processor and the DE2 output pin signals, it controls the signals going from the FPGA to the Motor driver IC in order to control the direction of the DC motor for the purpose of opening and closing the roof.

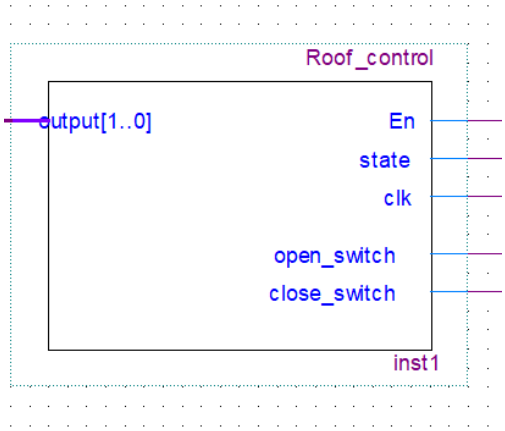

Figure 3.11 Roof Controller Diagram

3.3.2.3 Buzzer Controller

The Alarm system is implemented using a simple piezo buzzer that requires a square wave to generate a sound. The buzzer controller is an interface between the Nios II processor and external DE2 board pins; it creates a sound by generating a PWM signal – a square wave signal, which Is nothing more than a sequence of logic zeros and ones.
Figure 3.12 below shows a block diagram of the Buzzer controller where PWM pulse is applied with a 2second duty cycle.

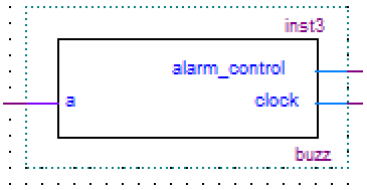

Figure 3.12 Buzzer Controller

3.3.3 Top Level File Compilation

The overall System design is shown in **Figure 3.14** It contains both the SOPC and the Non-SOPC systems illustrated in the Top-level file block diagram.

The compilation of the entire system was successful and the summary of the compilation is shown in **Figure 3.13**

Flow Summary	
Flow Status	Successful - Wed Sep 21 23:19:14 2022
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	cpu_unit
Top-level Entity Name	full_design
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	2,539 / 33,216 (8 %)
Total combinational functions	2,292 / 33,216 (7 %)
Dedicated logic registers	1,533 / 33,216 (5 %)
Total registers	1533
Total pins	32 / 475 (7 %)
Total virtual pins	0
Total memory bits	44,032 / 483,840 (9 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 3.13 Compilation Summary

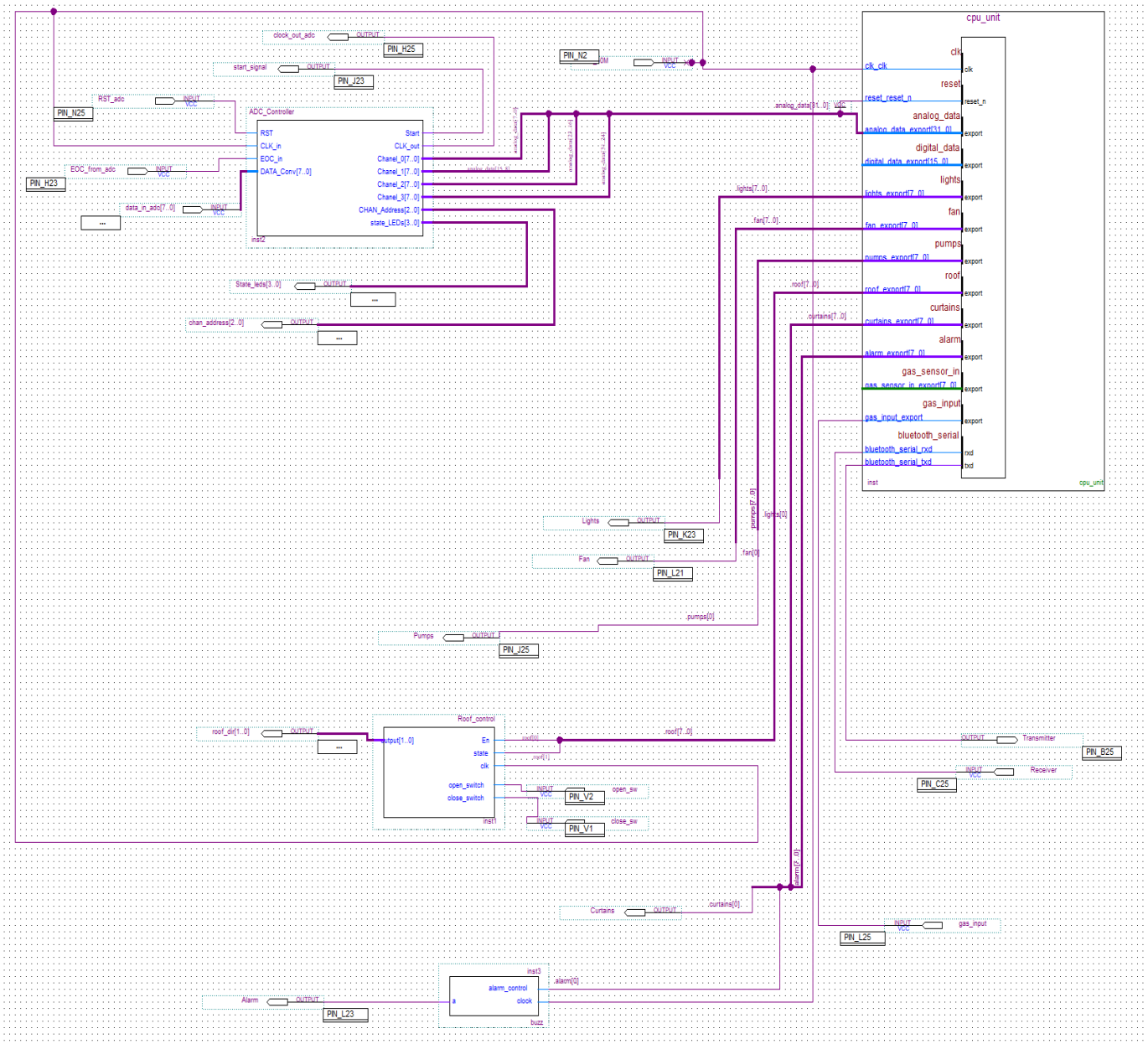

Figure 3.14 Overall Block Diagram of the On-Chip System

Chapter 04

Software System Design

4.1 Overview

There are multiple components involved in the design of an embedded system and they can be classified into two categories, hardware components and software components. In this chapter we will be describing the software components developed for this project.

After building the on-chip and off-chip hardware parts, software comes in to allow hardware to run according to the specifications of our project functionalities.
The software part of this project is organized into two main parts:

- The firmware, which is a program on the on-chip memory of the FPGA system that controls the behavior of our system and the kind of instructions we want to send. It is developed using high level C language through the Nios II software build tools for eclipse
- The Android Application, which serves as a graphical user interface in order to visualize and monitor our data collected from the input sensors. It allows the user to read the status of the overall system and it is developed using the Kodular tool for android applications

4.2 The Android Application

The main function of the android application is to display real-time information about the status of different parameters of our greenhouse prototype that are based on the sensor readings.

The information is sent throughout the Bluetooth module and is received by the android application.

The android application is developed using Kodular tool that was previously described in chapter one. It has two design features, the “front end” which is a visible user interface and the “back-end” which is non visible and applies a block-based development flow in which we can use pre-built functions and manipulate them according to our desired functionality of the application.

4.2.1 Kodular Designer Platform

The workflow to develop the android application is to first start a new project on the Kodular online platform, after that we can drag-and-drop components from the available components library and adjust their different parameters.

For the application to serially communicate with the system, a correct Bluetooth connection has to be made. For this task and as shows in Figure 4.1 a button on top of the application’s screen is designed to display a list of nearby available Bluetooth devices. The selection of the HC-05 Bluetooth module yields to viewing the status “Connected” on the screen. The user is then able to check the greenhouse’s environment parameters; it presents the temperature level inside the greenhouse as well as the weather condition outside the greenhouse where it informs the user whether it is raining or not. In addition to that, it shows the soil moisture level of the crops and the light intensity level. Finally the app informs the user about the security of the greenhouse.

If there are any fire risks, the page will display a warning for the user; else it will simply display “SAFE”.

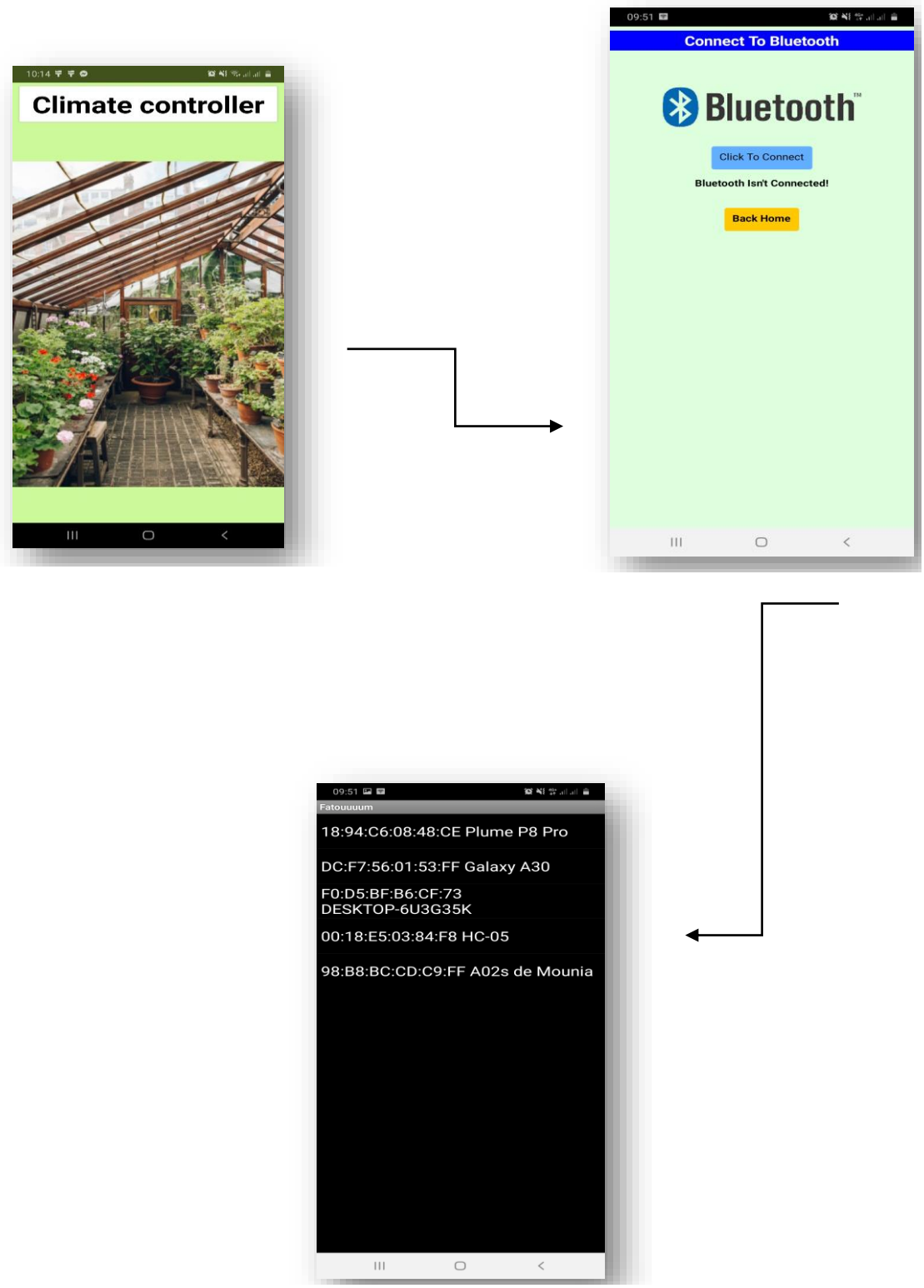

Figure 4.1 Steps to Connect a Bluetooth Device

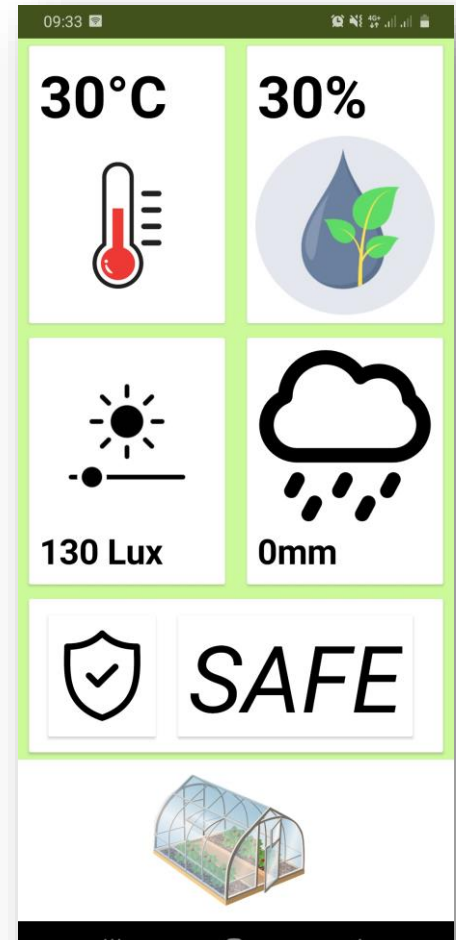

Figure 4.2 Workflow of the Android Application

4.2.2 Kodular Blocks Editor

The second stage of the design process is the blocks editor, the Editor uses OpenBlocks, an open-source java library. These blocks are considered as visual programming codes which can be dragged and cemented with other blocks to create a desired functional behavior of the application. **Figure 4.3** depicts a segment of the application's blocks editor.

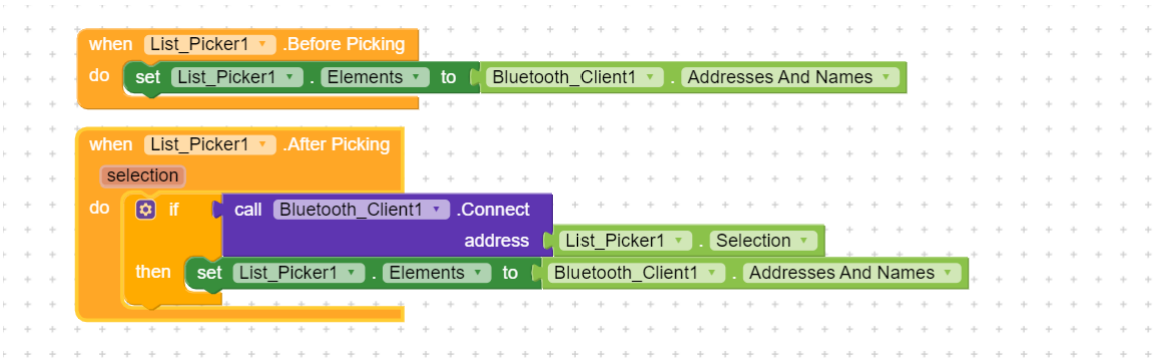

Figure 4.3 Segment of the Application's Blocks Editor

Testing on an Android smartphone is the final part of the design. Live testing of the application can be done throughout the design process through what is known as the 'Kodular Companion App' available on app store; it can be connected via QR code with the online platform and we can perform simultaneous tests. Once the development of the app is done, we can download directly a .apk file and the application is run on the smartphone and ready for use.

4.3 The Firmware

In microprocessor-based systems, a firmware is a computer software that provides the low-level control for a device's specific hardware. The hardware setup itself cannot provide any functionality without a firmware that manipulates all incoming data, stores data in the corresponding variables or registers, performs conditional tasks and communication between the processor and the different peripherals. It can be programmed using Nios II Assembly language or high-level language containing multiple routines shared between different tasks.
In the case of our project, C language has been used to develop the Firmware and Figure 4.4 shows a firmware's flowchart of this project.

It starts off by initializing the actuators into a desired behavior like turning off lights, dc motors of pumps and fan and closing off the roof. The main instruction sets depend on the data coming from the sensors via the data acquisition unit where data is constantly read from the PIOs and is saved in an array variable. The system operates in an automatic mode through pre-set setpoints where it performs a series of conditional if statements; The system then generates output signals that are sent to the control blocks and then to the actuators in the external off-chip hardware system in order to control the different environment parameters of the greenhouse. Finally, it sends off the sensor data readings to the smartphone android application to be displayed via the UART port and the Bluetooth module.

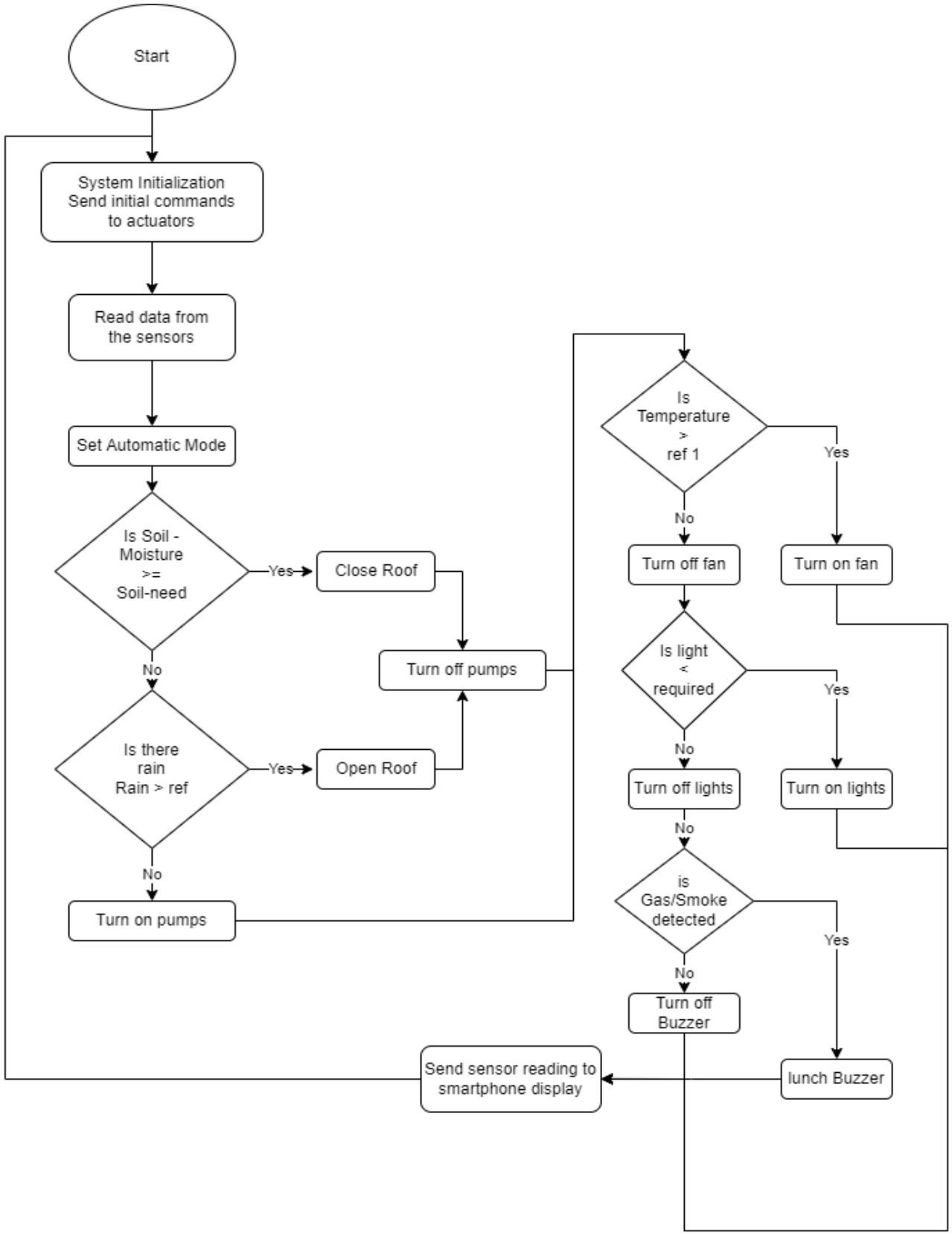

Figure 4.4 The Firmware's Flowchart

Figure 4.5 shows the Main function of the program, where it runs non-stop in a while loop. The main program collects, manages and sends data every fixed period of time.

```
int main()
{
    int delay_count ;
    alt_putstr("Hello from Nios II!\n");

  while (1)
  {

      getMeasurements();       // get measurements from sensors
      set_automatic_mode();  // set the control logic of each mode
      write_data(); // write data in PIOs
      send_bt_data(); // send data to bluetooth uart

      for(delay_count = 100000; delay_count!=0; --delay_count )  {/*read and write data every 20 seconds}*/}

  }

  return 0;
}
```

Figure 4.5 Screenshot of the Main Function.

Conclusion

The design and implementation of a Nios II microprocessor-based controller for a greenhouse environment has been realized in this project. An android application serving as a graphical user interface and a data visualization tool has been developed in order to allow the user to check the status of the environment inside the greenhouse.

Despite the numerous bugs in both the hardware and software development flow, we can successfully say that this project has met its initial goals and the implementation of the SoPC environment controller for a greenhouse was carried and the final prototype was fully functional.

This project can be enhanced by improving many features such as developing a more sophisticated User interface and enabling remote control of the environment controller through the web.

References

[1] Smart greenhouses as the path towards precision agriculture in the food-energy and water nexus: case study of QatarTheodora Karanisa, Yasmine Achour, Ahmed Ouammi & Sami Sayadi

[2] Michael Barr, "Embedded Systems Glossary". Neutrino Technical Library. Retrieved 2007-04-21.

[3] Heath, Steve (2003). “Embedded systems design. EDN series for design engineers” (2 ed.). Newnes. p. 2. ISBN 978-0-7506-5546-0.

[4] Michael Barr; Anthony J. Massa (2006), "Introduction". Programming embedded systems: with C and GNU development tools. O'Reilly. pp. 1–2. ISBN 978-0-596-00983-0.

[5] John Wiley & Sons,Inc “Embedded SoPC design with Nios II processor and VHDL examples”

[6] “FPGA Architecture for the Challenge". toronto.edu. University of Toronto.

[7] [Online]: https://www.generatecnologias.es/en/fpga_architecture.html]

[8] [Online]: https://www.intel.com/content/www/us/en/products/details/fpga.html

[9] [Online]:https://www.arrow.com/en/research-and-events/articles/fpga-basics-architecture-applications-and-uses

[10] Perry, Douglas L, “VHDL: Programming by Example”.

[11] [Online]:https://www.allaboutcircuits.com/technical-articles/hardware-description-langauge-getting-started-vhdl-digital-circu

[12] Altera Cyclone II Device Handbook, Volume 1

[13] DE2 Board Manual

[14] Altera, “Introduction to the Quartus® II Software”.

[15] Altera, “Introduction to Altera Qsys tool”

[16] Franjo Plavec, “Soft Core Processor Design”.

[17] Altera. "Nios II Embedded Processor Backgrounder".

[18] Altera Nios II Processor Reference Handbook.

[19] [online]: "Is Android Really Open Source? And Does It Even Matter?". MakeUseOf. March 28, 2016.

[20] [online]: https://www.elprocus.com/what-is-android-introduction-features-application

[21] [Online]: https://fr.wikipedia.org/wiki/App_Inventor

[22] [Online]: https://appinventor.mit.edu

[23] [Online]: https://www.kodular.io

[24] **De** Nathan J. Muller**, "**Networking A to Z**".**

[25] [Online]: https://www.bluetooth.com/about-us

[26] "HC-05 Bluetooth to Serial Port Module" User manual.

[27] [Online]: https://www.techopedia.com/definition/572/analog-to-digital-converter-adc

[27] [Online]: https://www.elprocus.com/analog-to-digital-converter/

[28] Texas Instruments, "ADC0808/ADC0809" datasheet

[29] Leeson electric, "Basic training for electric motors"

[30] Al Williams (2002). Microcontroller projects using the Basic Stamp (2nd ed.). Focal Press. p. 344. ISBN 978-1-57820-101-3.

[31] [Online]: datasheet LD293D quadruple half H bridge texas instruments]

[32] [Online]: https://mulloverthing.com/what-is-the-purpose-of-a-buffer-amplifier/

[33] SN74LS244N Octal Tri-Sate Buffer Motorola, Inc datasheet

[34] "BC546B Datasheet" Motorola, Inc datasheet.

[35] [Online]: https://www.americanpiezo.com/standard-products/buzzers.html

[36] Seed Studio, "Gove – Moisture Sensor User Manuel"

[37] "Rain drop sensor" e-gizmo manual.

[38] [Online]: https://www.electronics-tutorials.ws/io/io_4.html

[39] [Online]: https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor

[40] [Online]: https://quartzcomponents.com/products/mq-2-gas-sensor-module

[41] [Online]: https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/