Smart Greenhouse Project

Computer Engineer

Computer System Engineering I

Group 1

Rahaf Darouich

Sara Alterkawi

Contents

Abstract 3
1. Introduction 4
2. Method (Project requirements) 5
3. User Guide 10
4. Conclusion 10
.5 References 11

Abstract

This project aims to create a comprehensive system for managing the climate in a **Smart Greenhouse** that provides an environment for healthy plant growth. To achieve this, we will be utilizing various elements such as a photosensor (ADC), temperature sensor (temp), and RC servo motor (PWM) to monitor and regulate the temperature and lighting conditions.

In order to complete this project, we were provided with several essential components such as the Microchip (Atmel) SAM3X8E-ARM, the Arduino Due embedded computer platform, a Keypad (Port I/O), a display (Port I/O), and LEDs. The introduction of the project provides all the necessary prerequisites for the successful implementation of the project, along with a functional block diagram that illustrates how the embedded system works.

Additionally, there is a dedicated section for first-time users that provides detailed instructions on how to use the project effectively. The C programming language was used, and the datasheet of each element was studied and researched. Overall, this project offers a robust and functional solution for managing the climate in a **Smart Greenhouse**, making it an essential tool for agricultural and environmental researchers.

1. Introduction

The very useful means that helped us a great deal to accomplish this project were the hardware and the software labs in which we could learn the working mechanisms of the embedded system that works as an implementation unit with different registers and peripherals units, such as the ADC unit, PWM unit, etc. and that we also could know the way of linking them together. These registers and units are supplementary to each other, and they all lead to improve an embedded system and perform a variety of tasks, such as managing a particular system. However, the main aim of this project is to imitate making a **smart greenhouse** which offers the plants an appropriate environment that enables them to develop healthily through making use of some sensors and actuators that help doing this task. Each system, in general, consists of three elements, which are the sensors, the implementation unit and the actuators, that activate its working mechanism. As for the sensors in this system, they work to receive the signals then send all the data connected to the system's tasks, such as the light intensity and the temperature, to the system. Concerning the implementation unit, it is the motherboard of the system in which all the system's tasks are accomplished. The actuators, however, work as a means that shows the results of the tasks' implementation, such as displaying something, setting an alarm, etc. To be put into consideration, the essential things, which were very beneficial, and which enabled us to understand the pointers and the ways of using them in the code that will help us when we encode and wire the peripherals and which contributed to give us general ideas concerning initiating the project and improving it, were the software labs. Our first step in this project was wiring the elements to connect them with each other as shown in diagram (Figure 1) which is available on Blackboard. In addition, in this project the different elements; namely the servo motor, the temperature sensor and the LED, all work automatically to promote the keypad that will connect with the display to offer a calendar and a list of different emerging variants.

As a beginning, we had to create a calendar, which displays the date in the DD/MM/YYYY format and the time in the hh:mm:ss format with the 24-clock system, and which gives the user the permanent opportunity to check and edit it.

Then the temperature had to be recorded within just one minute and for a whole week to be time-stamped and so to know exactly when it was recorded because the recording should start as soon as the buffer of the system is filled.

After that, the recorded data had to be displayed on LCD to show the data in minimum, maximum and average temperature every day, but the minimum and maximum temperature values should show up along with their recorded time.

Next, the greenhouse is designed with a large glass roof that is horizontally oriented and equipped with motorized mirrors and a shading system. These mirrors are controlled by a motor and are strategically positioned to face the sun, reflecting sunlight through the glass roof and into the interior where the plants are located. A photosensor helps track the movement of the sun to ensure optimal reflection. The shading system allows for precise control of the amount of light entering the greenhouse by adjusting the shades to be opened or closed as needed.

Managing light and darkness with an aim to ascertain the availability of the healthy environment that offers the plants their full need of these two important elements to develop properly. In general, the natural system for every plant to develop healthily is getting sunlight for 16 hours and darkness for 8 hours daily. However, because the sun in Sweden doesn't often arise due to the weather conditions in this country, we had to replace it with the LEDs' light system that can be as beneficial for this purpose as the real sun, and to display the position of the sun in degrees, the number of sun light and the number of hours of Lighting system, the number of hours of darkness.

To provide the optimal light to darkness ratio necessary for healthy plant growth, it is essential to implement a comprehensive system that can accurately track the movements of the sun and control the positioning of mirrors and shades. Alongside this, a lighting system that contains LED technology can also be hired to imitate the natural rays of the sun.

When these diverse factors are combined, the ensuing system can ensure that plants receive the appropriate levels of light and darkness required for their development and well-being. This is particularly crucial in indoor environments where access to natural sunlight may be limited or inconsistent.

By simulating the natural movements of the sun and carefully controlling the amount and direction of light that is provided, this system can create a constant and optimized growing environment that is perfectly tailored to the needs of the plants in question. Through these means, it becomes possible to maximize the growth and yield of plants, even in difficult indoor environments. Then we had to set an alarm to alert us if the **greenhouse's** temperature surpasses its lowest or highest defined levels that were established by the user.

Finally, a fast mode system, which equals thirty minutes of our current time, was created to enable us to have a check over all the previous tasks and to know if they are working properly.

2. Method (Project requirements)

As shown in the diagrams below (Figure 1), we began with wiring the keypad and the display into the Arduino) in the hardware lab2, and to define this project we used of the C programming language and in order to see if the keypad and the display are appropriately connected together, we started by writing just a simple code after getting a considerable help from studying and researching the datasheet of each element like (sam3x8e[2], display [3], keypad [4], temperature sensor [5], photosensor [6], and servo motor [7], and so after all the elements were wired, we were able to initiate the project's assigned missions. Furthermore, we wanted to create on the display of this project a list that enables the user to choose whatever he wants from it. In this list that can show the date and time there will be options that enable the user to set the wanted time and date by himself and that also enables him to have a daily look over all the temperature data, its maximum, minimum or average levels. Moreover, in this list there is a specific part for the alarm and a specific option that gives the user the choice to enable or disable the fast mode system.

In requirement 1, we used a Systick as a base for coding the created calendar, and a screCase variable of the char kind, which informs the user about everything that pops up on the screen, was also created to establish this system. For example, when the screCase is 'm' the display shows the main list, and the display will exhibit the time and date when the user presses the button 1, and there the screCase will be 't'. Also * can be pressed by the user if he wants to go back to the main list, and to set the calendar he can press #. However, a setTime function was created to enable the user to configure the date and time in the calendar through just a simple press on the buttons of the keypad which consist of 17 digits and that exhibits the date and time in the format he wants along with an indicator that goes to the following digit whenever the user presses a button until he reaches the last digit that make the indicator get a reset before it vanishes from the screen.

In the requirements 2 and 3, the temperature had to be recorded every minute to be shown on the display, and this recorded temperature data is saved from the beginning of its recording in the linkedList file that enables us to calculate the maximum, minimum and average temperature at the end of every single day and to be able then in the following day to start again recording the data of the new temperature. However, to do these requirements we had to create four functions that can do the tasks which are temperatureReading, signalReading, weekData and printData. To be more clear, if we want to start recording the data every minute, the temperatureReading task can do this by sending an order to the temperature sensor, and in order know every second if there is any response from the temperature sensor, the signalReading can also attend this need, and the weekData can simply keep and calculate all the everyday data then goes directly to the following day, and lastly, the printData to print the records on display , Date minimum temperature with its time-stamp maximum temperature with its time-stamp and the temperature average.

In requirement 4, we had to manage the horizontal glass roof surface, using photosensor to track the movement of the sun, ensuring sunlight reflection. Additionally, the shading system enables precise control over the amount of light that enters the greenhouse, with the ability to adjust the shades to open or close as needed.

And we can manage turning from light to darkness to provide the plants with their natural healthy environment that requires 16 hours of lightness and 8 hours of darkness as mentioned above. To be able to do this and to solve the problem of the scarcity of light in winter and darkness in summer, in this project we got to work on two systems; the light system that requires using the LEDs to provide the plants with enough hours of light in winter, and the shading system that uses the servo motor to provide them with enough hours of darkness. However, the greenhouse in this project contains a counter that can calculate a period of 960 minutes that is the same amount of the 16 hours in our current day (16*60 = 960), and if this counter did not get this 960 minutes of lightness (photosensor), it will activate the LEDs that replaces the sun until it gets the required amount of time for lightness and move after this to activate the shading system instead, and after that to print the sun position, number of hours of the sun, the lightning and darkness hours. By creating three functions that can do the tasks: PWM_rotate to rotate a servo motor to different angles based on the sun position, houseLight to control the lighting system with help of SysTick_Handler, and printLight to print the current day light, darkness, and sun position on a display.

In requirement 5, we had to set an alarm that works to alert user if the temperature exceeds its highest or lowest levels, and to ensure that this works well, the setLimits and the alarmCheck function were created to turn the LEDs on or off when need requires. At last, to spend one day in 48 seconds of our real time in this **greenhouse** we had to create the fast mode system, which the user can enable and disable it in the time he wants through just a simple press on the keypad, to make the one-hour increments one every two seconds.

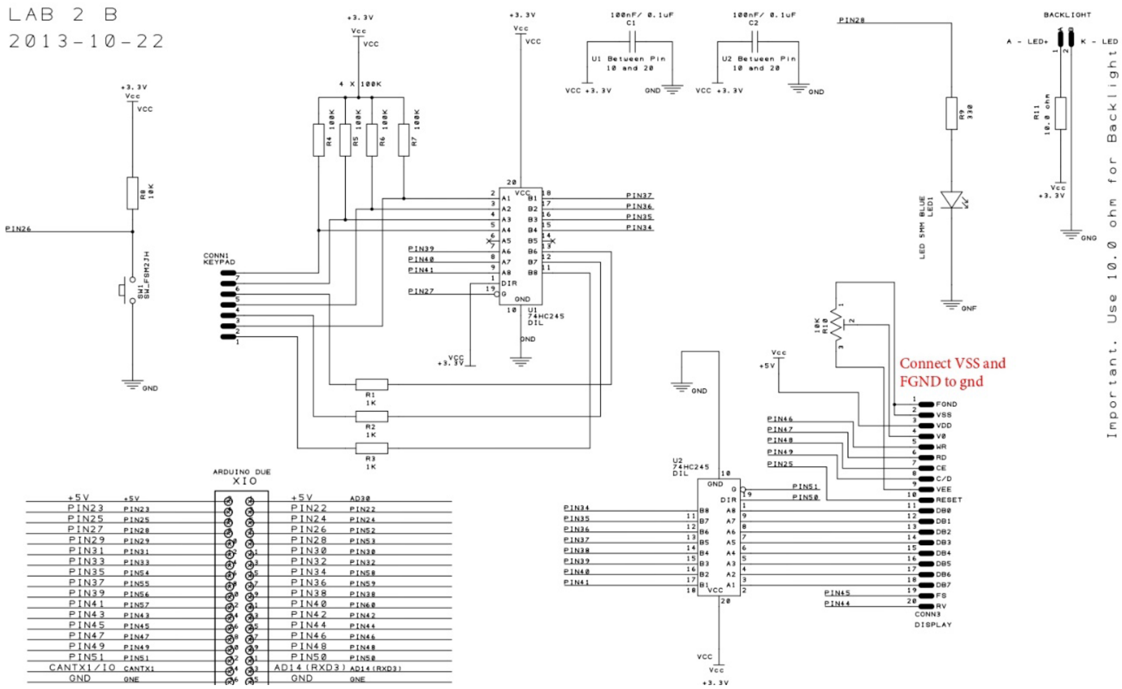

Figure 1 Wiring diagrams for keypad and display

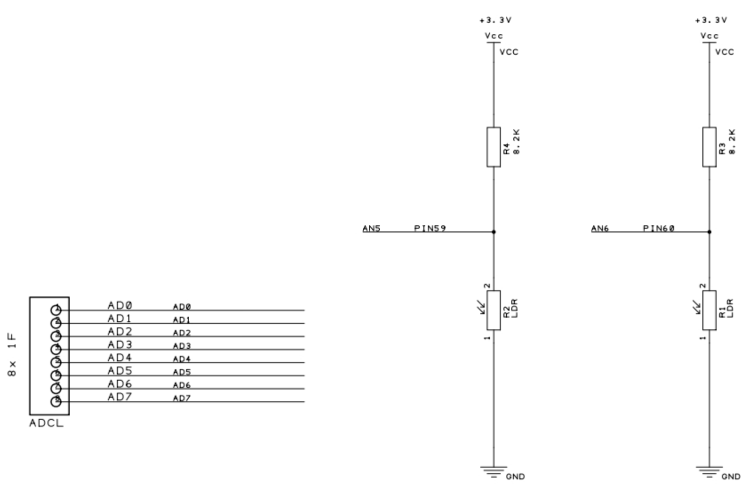

Figure 2 Wiring diagram for the photosensor.

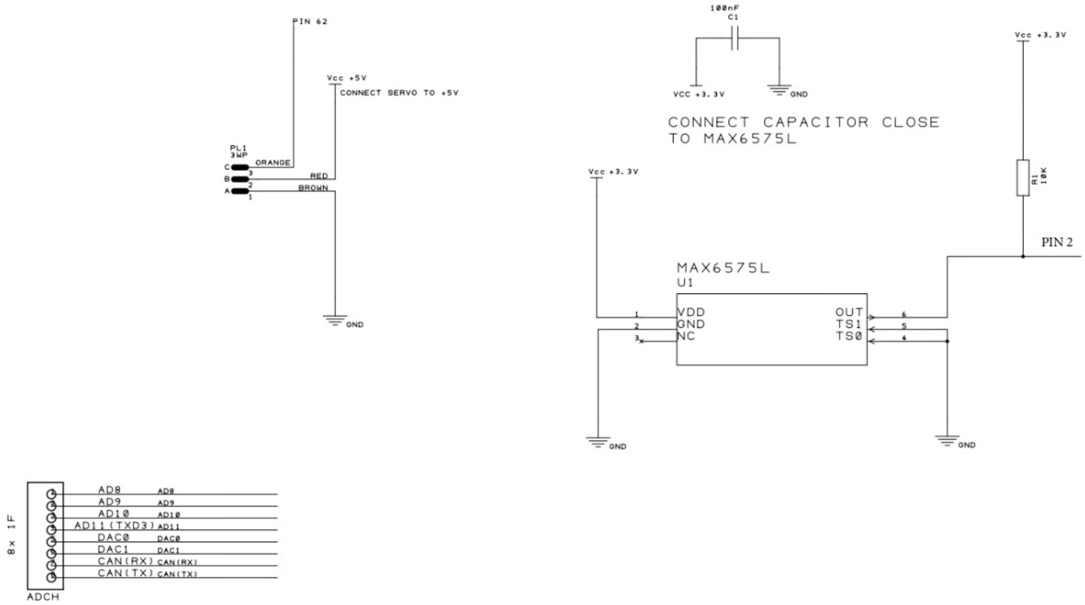

Figure 3Wiring diagrams for servo motor and temperature sensor.

In the following diagram a general overview is provided to show the working mechanism of the embedded system.

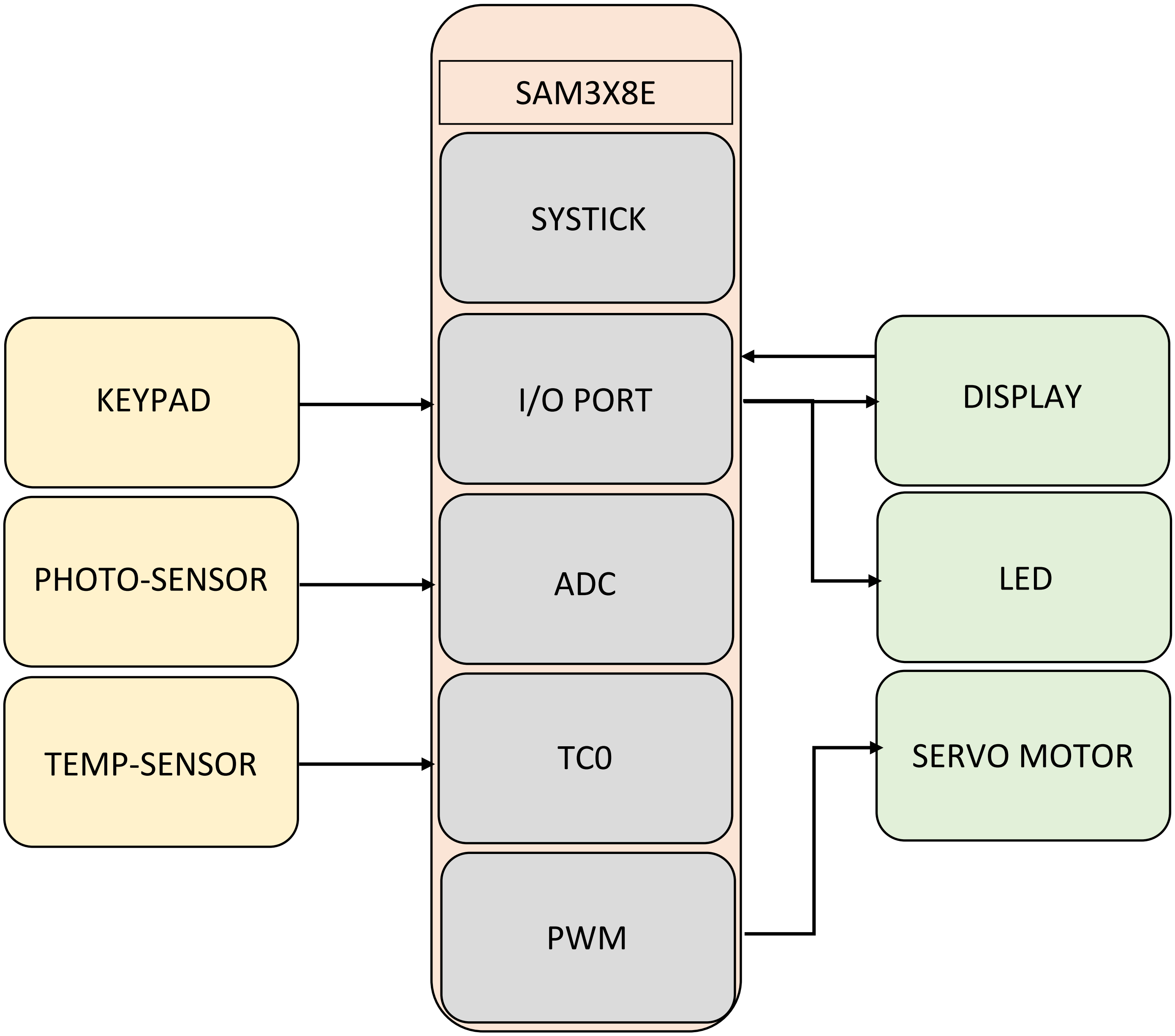

Figure 4 Functional block diagram

The purpose of this project is to make a **greenhouse** that provides the perfect healthy environment for the plants to develop healthily. The embedded system of this **greenhouse** has three levels as said previously; the photosensor, temperature sensor and the keypad that informs the system of any pressing on the buttons, the brain is sam3x8e that gets the input signals and, at the same time, controls the output circuit along with the various other parts, and the outcome field.

3. User Guide

The first screen that includes the project and group names, to go to the main list press #
Main list has various choices, and to go back to the main display whenever you like press *.

Main list choices:

1. Time and date display: To have a look at the time and date press 1! And if you want to set them press # and enter the digit from keypad (for going to the following digit without any change also press #).
2. Start temperature recording: To record the temperature, press 2!
3. View temperature records: To see the recorded data press 3!
4. View temperature limits: To configure the highest and lowest temperature levels press 4! press # and enter the digit from keypad (for going to the following digit without any change also press #).
5. View lightning records: To show the sun position, hours of lightning and darkness press 5!
6. Start / Stop fast mode: To enable the fast mode, press 6, to disable the fast mode, press 6 again.

If you have the warning light on press * to stop it or go to set limit screen and fix a new temperature limit.

4. Conclusion

To sum up, in an aim to attain a grade 3, all the previously mentioned requirements to make this project successful are practically implemented and applied to their full to make the **greenhouse** work out properly as it was planned for it.

This project aimed to manage the climate of a "**Smart Greenhouse**" using various elements such as photosensors, temperature sensors, and servo motors to observe the temperature and light intensity. To accomplish this, elements such as Microchip (Atmel) SAM3X8E-ARM, Arduino Due, Keypad, Display, and LEDs were used. The introduction provided a functional block diagram explaining the working mechanism of the embedded system. The hardware and software labs were beneficial for understanding the pointers and ways of using them in the code. The project involved creating for example, the recorded daily temperature data accompanied with its timestamp and the calendar are displayed in the screen clearly, the darkness and lightness systems, which work by the sun's tracking and the shades' preserving, are working properly, the alarm that helps us a lot to avoid any possible harm for the plants by ensuring that they acquire their sufficient amount of temperature is also set efficiently, and the fast mode proved its great usefulness in speeding and facilitating the checking processes. However, while working on this project, we run into some troubles concerning the memory, the matter which informed us about the great importance of this factor for projects like this. At last, a considerable lot of time was taken to understand and being familiar with the wiring diagrams that provided us with the suitable way of wiring the project.

Overall, this project successfully demonstrated how to make a **smart greenhouse** and could be beneficial for those who want to develop a similar system.

5. References

1. Wiring diagrams from Blackboard.
2. SAM3X8E Datasheet
   https://ww1.microchip.com/downloads/en/devicedoc/atmel-11057-32-bit-cortex-m3-microcontroller-sam3x-sam3a_datasheet.pdf

3. Display Datasheet "NEWHAVEN Display"
   https://newhavendisplay.com/content/specs/NHD-240128WG-ATMI-VZ.pdf

4. KeyPad Datasheet
   http://www.accordia.com.tw/spec.html

5. Temperature Sensor Datasheet
   https://www.analog.com/media/en/technical-documentation/data-sheets/max6575-max6575l.pdf

6. PhotoSensor Datasheet
   https://www.electrokit.com/uploads/productfile/40850/40850001.pdf

7. Servo Motor Datasheet
   https://datasheetspdf.com/pdf-file/791970/TowerPro/SG90/1
8. SAM3X-Arduino Pin Mapping
   https://fides.fe.uni-lj.si/~jernejo/modulB_due/pinMapping.pdf