Mini-Projects A & B

Terrarium Logger

EEE3096S

Embedded Systems II

University of Cape Town

Chapter 1

The Terrarium Mini-logger

1.1 Overview

The mini-project is a chance for you to use what you have learnt in this course and test your problem-solving skills. The projects use everything learnt in the previous practicals - so here's to hoping that you paid attention and remembered those! All students need to do Mini-Project A. Only CS students (EEE3095S students) need to do Mini-Project B.

1.1.1 Scenario

Your uncle has heard of you achieving great things in an embedded systems course at university. He approaches you and requests your assistance in building an environment logger for his terrarium to assist with taking care of his pet bearded dragon. As they are ridiculously adorable (bearded dragons, not uncles) - you agree to assist.

Your uncle hearing about how hard you are working despite how difficult things can be and generally being proud of you
Your uncle when you agree to help him set up a live monitor of his terrarium

Figure 1.1.1: You can read more about this adorable creature here

Your uncle needs to monitor temperature and light levels at varying intervals, which needs to be adjusted by a button press. He also wants to enable and disable logging by pressing a second button. Your uncle has a spare system which they can use to log into the Pi remotely and monitor console output, which he lists as one of the requirements. However, they wish to also monitor the terrarium remotely when they are not there. Their son, who plays with Arduinos, has spoken a lot about Blynk for IoT devices. You decide to look into these options for remote monitoring.

The following requests were also made:

- the most recent 20 samples are stored on an EEPROM chip as backup, in the case that internet access is lost.

1.1.2 Basic Details

Your objective for Mini-Project A environment logger. Mini-Project B is an add-on to Mini-Project A which involves minimal work to extend the functionality of the monitoring system. If you are in a group with both CS and EE members, you can both work together on Mini-Project A. For Mini-Project B, CS students can chose to work alone or partner up with another CS student to form new groups of 2.

Generally speaking, an environment logger interacts with the world around it by measuring various factors from GPS location to air pollution. While a single sensor is useful to monitor a single location, many can be scattered around a broader area to monitor that area as well as any patterns emerging in that broader area. This data can be used to track changes over time to forecast effects or make inferences between locations and how they might affect each other. In these mini-projects there is not much data to track and there aren't many potential effects to forecast, though the principles remains the same.

Project A and B differ in how that data is presented to end users. Project A is simply accessed through a terminal. Project B will add reporting data to Blynk, an app that can be installed on smartphones. The environment logger can also be controlled via the Blynk app.

NOTE: FOR EACH SUBMISSION, BOTH STUDENTS NEED TO INCLUDE A SIGNED PLAGIARISM DECLARATION AT THE END OF THE SUBMISSION, WITH THE SUBMISSION AND BOTH PLAGIARISM DECLARATIONS IN A SINGLE PDF. THE FORM CAN BE FOUND AT THE END OF THIS MANUAL.

Chapter 2

Mini-Project A

This project examines ELO5.2, which requires the student to develop a representative embedded system. It requires hardware/software interfacing, so an ADC is sampled by software, and processing operations are applied to read the data.

2.1 Project Requirements

The Project is split into Milestones in order to help you plan time and ensure you aren't overburdened with work a few days before the final deadline. Dates for submission will be put on Vula. The milestones are as follows:

A. Planning and System Design
   For this milestone we expect to see the following:

   - A Gantt Chart indicating your planned activities with the project
   - A circuit diagram for the system
   - A choice on how you plan to approach the project (e.g. Spiral, waterfall, etc.) as well as a justification as why
   - A flowchart detailing how the system operates

B. Code and Report
   The final hand in will be the report and your code submission. Details on these can be found in the marking guides.

2.2 Outcomes

In addition to creating a working system, you will learn about the following in this project:

- Technical outcomes
  - ADC - MCP3008
  - Temperature sensor - MCP9700A
- Other outcomes
  - Team Work
  - Design methodologies, such as the Waterfall, Spiral and V Models.
  - Planning and time management

2.3 Deliverables

At the end of this project, you must:

- Submit all milestones
- For information on the marks and what sections to cover, refer to the marking guide in Section 2.8

2.4 System Overview

Figure 2.4.1 shows the system overview for the mini-project.

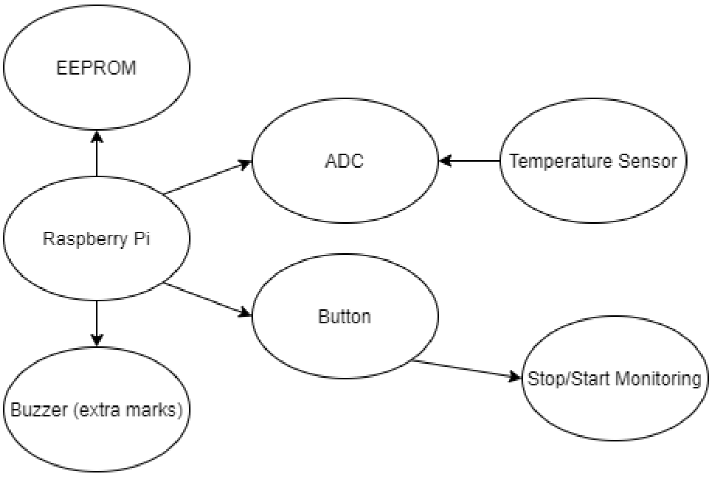

Figure 2.4.1: Components of Mini-Project A

2.5 Hardware requirements

You will need to build your designed circuit on a breadboard and demonstrate it in the video you create.

2.6 Software requirements

You need to do the following:

- Set up the EEPROM.
- Create a thread for reading from the ADC. The value read from the temperature sensor needs to be converted to degrees Celsius. See the datasheet (linked above) for the formula.
- Create interrupts for all the button functionality (don't forget to debounce your inputs)
- Create an output signal to trigger the buzzer (bonus marks)
- In the main() loop, print values to the screen as described in Section 2.7

2.7 Description

- By default, the system continuously monitors the sensors every 5s using this format:

Time	Sys Timer	Temp	Buzzer
10:17:15	00:00:00	25 C	*
10:17:20	00:00:05	25 C	
10:17:25	00:00:10	25 C	
10:17:30	00:00:15	25 C	
10:17:35	00:00:20	25 C	
10:17:35	00:00:20	25 C	*

- The stop switch stops or starts the monitoring of the sensors. The system timer is not affected by this functionality. The screen must also be cleared when logging is stopped, and a message printed to display to inform the user that the logging has stopped.

2.8 Marking Guide

Table I: Project A marking Guide

Heading	**Report**
Introduction	Provide a short introduction ($\sim \frac{1}{2}$ page) to your project explaining your main design choices and how the report has been structured [**15 marks**]
Requirements	The requirements section should provide a refined UML Use Case diagram of the system, according to your implementation, and any accompanying text that is needed to clarify the requirements. Highlight any departures or additions that you may have made compared to the original project description given in this document. [**15 marks**]
Specification and Design	This section should provide a UML State Chart describing the main operation. Add a UML class or deployment diagram (or other suitable diagram) to indicate the structuring of your implementation (e.g. code modules/classes you may be using). You don't need to provide fine detail of the system, the diagram(s) can be e.g. at the level of functions. You should also include a circuit diagram [**20 marks**]
Implementation	This section should give some snippets of important code and explanations for this (or referring to particular functions in code files). The point here is elaborating any parts of the State Chart that are not so straightforward to turn into code. [**20 marks**]
Validation and Performance	Provide at least a paragraph or two explaining the performance of the system. A snapshot could be included and you could show test cases where you have tested that the system works reliably (e.g. using a powersupply to set the value given to the ADC). [**20 marks**]
Conclusion	Give a summary of the extent that the system was found to be successful. Discuss if you think that a system working in this way might be considered a potentially useful product. [**10 marks**]
References	Provide a few references if relevant.
Total	100

Chapter 3

Mini-Project B

This project is for CS students only.

3.1 Scenario

Your uncle appreciates the work you have done developing a rough prototype. Because you did such a good job of developing the base system, your uncle has asked you to develop a means of a system where they can access their data remotely. You decide to do this as simply as possible, making use of an application called Blynk.

3.1.1 Overview

Essentially, this is a repeat of the first project, but adds a button to change the interval reading and makes the data accessible through Blynk.

3.1.2 Deliverables

At the end of this practical, you must

- Submit a short write up. See the marking guide in section 3.1.4

3.1.3 Hardware Required

The hardware from mini-project A and an additional button.

Software Requirements

- You need to add Blynk.

- You need to add functionality for changing the reading interval between 2, 5 and 10 seconds.

Figure 3.1.1 shows an example of what the app you create in Blynk might look like. This is the example from last year's project, but you can see there's a terminal for accessing the output of the print statements and a space for values as read from the ADC. Last year's project also had an alarm, which is not a feature in this year's project so you need not include one. There are various widgets in Blynk. It's suggested you play with your energy budget to develop the most intuitive and aesthetically pleasing design possible.

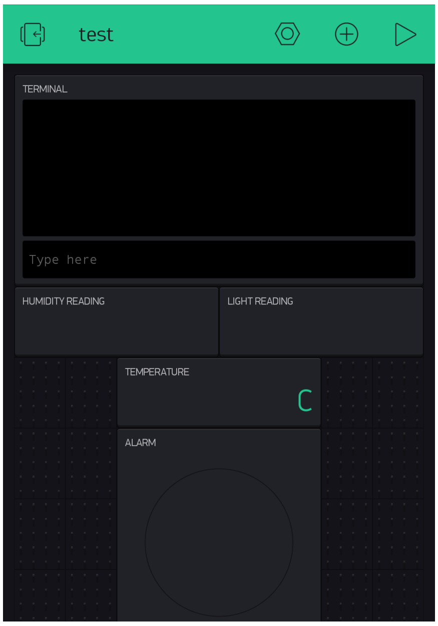

Figure 3.1.1: Example Blynk Project

3.1.4 Marking Guide

Table I: The Write Up Format For mini-project B

Section of Report	**Description**	**Marks**
Introduction	An introduction to what’s new in this project	10
Design	Design of the system. Design of the server. Talk about the stack used for development. Include UML and block diagrams, and mention any hardware/software interfacing issues.	30
Implementation/ Build Proces	Some code snippets and steps in building the device and server.Can think of this as a simple methodology.	15
Instructions for use	How to operate the system. You should include some screenshots and photos.	15
Testing/Results	How did you ensure your system works? What did you do to test the functionality, and what were the results? You can include screenshots or photos, but you need to talk about them (it’s not enough to just have a photo).	20
Conclusions	How well did your system work? Did you achieve the objectives? What could you do to improve the system?	10
TOTAL		**100**

Chapter 4

Acknowledgements

1. The front page image is take from onlineplantsindubai.weebly.com
2. The meme template used in Figure 1.1.1 is of Butter the bearded dragon, which is accessible through a Google search

Declaration of non-plagiarism

- I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
- I have used the ____________________ convention for citation and referencing. Each contribution to, and quotation in, this essay/report/project/submission from the work(s) of other people has been attributed, and has been cited and referenced.
- This essay/report/project/submission is my own work.
- I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

Name ____________________ Student Number ____________________

Signature ____________________ Date ____________________