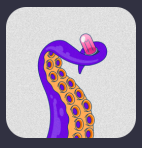
classpert



Making Embedded Systems

Final Project Evaluation

Orange Stars Cohort

Dodeca Timer

by Graeme Gets

Reviewers

Thomas Fuller

Elecia White

1 Overview

The objective of this document is to assess and give you high-level feedback on your final project. Completing it and receiving a passing grade is a prerequisite for the certificate of course conclusion to be issued. Your project was reviewed and graded by mentor **Thomas Fuller** and instructor **Elecia White**.

1.1 Project Details

Project Title
Dodeca Timer

Student Name
Graeme Gets

Enrollment ID
graemegets

Deliverables	Links
📄 Report	https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fraw.githubusercontent.com%2Fgraeme-gets%2Fmaking-embedded-projects%2Fmain%2Ffinal-project%2FTime%2520Tracker%2520-%2520Final%2520project.docx&wdOrigin=BROWSELINK
💻 Code	https://github.com/graeme-gets/making-embedded-projects/tree/main/final-project
▶️ Video	Class Presentation Recording & https://youtu.be/gA9Q5SyaInY

2 Final Evaluation

For each criteria, a score was given according to the grading rubric (see appendix). The total achievable score was **24**, of which **18** are common credits and **6** are bonus credits.

Criteria	Score	Notes
Project meets minimum project goals	** breakdown 3.2	This goes beyond a 3. This is a viable proof of concept for a consumer product.
Completeness of deliverables	**	The report is 25 pages, but every page has good info about background, design, fabrication, software, hardware and assembly. Everything needed to recreate this prototype is included.
Clear intentions and working code	**	The features are well modularized and can be easily navigated. The report and video add insight into the design.
Reusing code	**	Reused, built interfaces for and made upstream improvements to a few libraries. I think from other licences by the author and the text at the end of the blog post that the DMA Addressable LED library intends to be MIT, but it isn't in the repo. If you wanted to sell something with the library, you might want to clarify.
Originality and scope of goals	**	This is an immense amount of output for 10 weeks of work
Self-assessment (mentor category only)	**	Self evaluation of 14.5 points, mentor evaluation of 21 points. 6.5/21 is a 31% difference.
Power analysis, firmware update, or system profiling	**	Power analysis on average current of the LEDs in actual operation and report details the operating implications of the highest power consumer in the system. Examined memory footprint of lighting configuration into the WS8212 output
Version control was used	**	Great use of version control. You could consider uploading your Fusion360 and 3D print STLs into version control, so you can rebuild everything from a clone of the repo.

Total	**	PASSED

With a total of ** your project **PASSED**.

*Grades have been hidden

2.1 Reviewers' Feedback

2.2.1 Mentor Comments

by Thomas Fuller

This is a remarkably polished prototype. The suggestions I make are nitpicks. It is astounding that in 10 weeks you did mechanical design, fabrication and assembly for both the electrical housing and the dodecahedron faces.

You also wrote a report that did an excellent job showing how everything fits together. Consider adding a known issues section. I noticed in the video that your frame would fit together in a way that calculated the wrong upright face. Adding that note into the report will be helpful to yourself in the future when you forget

> "A slow flashing LED can be used to indicate the need for a charge."
Just a note to be sure to measure the power draw from this feature. In a past project, waking up the processor to handle flashing an LED consumed a surprising amount of remaining power and shortened the low battery operational life considerably more than I expected.

The method of storing config in an area of memory that's not overwritten on a gdb load is a great idea. You should consider adding a schema version field to that config area. If your config schema changes and new firmware is loaded, the config area may still pass the CRC check, but not be interpreted by the firmware correctly.

Overall you've got a great path forward. Please let me know when you end up shipping a product or kit. I'm very interested in a physical object that can help me do pomodoros.

2.2.2 Instructors Comments

by Elecia White

Graeme, you did a good job. A very good job. Look at your score: the only place we aren't impressed is where you didn't recognize that you did a good job.

No, it isn't shippable yet but it is usable and useful. I can think of all kinds of uses for it:

- Activities (as you have it)
- Exercises if I'm doing different ones

- Tracking foods (yes, I really only eat about a dozen different foods)
- Pomodoro timer

You made something I want to build and you documented it well enough that I could build it. You've created flexibility for the future for yourself and for other people to use. I don't even want it to be smaller, the larger size would remind me to move it when I change tasks.

The one thing I really want is a sleep face without LEDs set so its batteries will last. Oh, and I want the LEDs to turn off after a little bit, coming on as I turn it so I see it works but shutting off to save batteries.

I can see how you'd want an internet connected piece for tracking long term. That would be nifty. On the other hand, a technical audience would be just as happy to not have their data go to the cloud.

It is an engaging project and I'm happy I got to look at it. Thank you!

3 Appendix

3.1 Grading Rubric

Criteria	Score		
	1 - Needs Improvement	**2 - Meets Expectation**	**3 - Exceeds Expectation**
Project meets minimum project goals	All project goals not met	All project goals are met. The state machine may be basic	Additional sensors, actuators Well documented and implemented state machine Comprehensive command line on serial port
Completeness of deliverables	Lacks report, video or code Report does not cover all sections Code has obvious errors that would cause it not to compile	Report covers all sections but some are answered incompletely leaving questions for the reader Code is readable given the report as a description Video shows code working	Code is readable on its own, without the report Report addresses each point thoroughly, demonstrating understanding as it related to the course Video demonstrates the project and is explanatory
Clear intentions and working code	What the system is supposed to do (based on the report or code) doesn't seem to be what the system does in the video	The system performs approximately as described in the report and code	The system performs as described in the report in a manner that is professionally polished The code shows how it works in a way that is easy for a maintainer to see

Reusing code	No code was used from other sources or it is unclear what code was used from other sources	Student code was identified	Versioning of reused code was included along with a license document that describes the license for the student's code and the reused code as well as shipping implications Reader is confident they could rebuild the student's system
Originality and scope of goals	The student did the bare minimum to meet the goals No originality	Some areas of interest were noted in the report but they were minor extensions of the existing examples	The student has gone far beyond the requirements to make something novel and awesome
Self-assessment (mentor category only)	Self-assessment was significantly different from mentor assessment	Self assessment was +/- 25% of mentor assessment	Self-assessment was +/- 10% of mentor assessment
Power analysis, firmware update, or system profiling	None	Described	Described, has graphs, and is accurate
Version control was used	None or a single commit		The log shows the project being built, though the messages may be terse but should be descriptive

3.2 Requirements

3.2.1 Project

🟢 Delivered 🟡 Partially Delivered 🔴 Not Delivered * Not Required ** Extra Credit

Features	Delivered	Note
Video turned in	🟢	Nice video demonstration. Seeing the inside with the wiring harness was really impressive.
Link to code	🟢	
Report turned in	🟢	
Use a Cortex-M processor	🟢	STM32
Button with interrupt	🟢	Button to change
Has serial port output	🟢	Extensive serial console, and extended command parsing
Implements a state machine	🟢	State machine to control operation between timing and orientation detection
Algorithmic piece	🟢	Face orientation detection, CRC on config
Peripheral 1	🟢	Accelerometer/Gyroscope
Peripheral 2	🟢	Addressable LED's controlled via PWM fed with DMA
Peripheral 3	🟢	USART & I2C Comms
Other*	🟢	Timers
Other*	🟢	RTC
Uses a HAL*	🟢	STM32 HAL
Analysis of Power**	🟡	Good analysis of the potential current draw from that sea of LEDs.
Firmware update**	🟡	Setting the config section to NOLOAD means that reflashing the controller does not overwrite data
System Profiling**	🟡	Examined memory throughput and configuration storage benefits for DMA based library.
Version control with history	🟢	

3.2.2 Report

🟢 Delivered 🟡 Partially Delivered 🔴 Not Delivered

Features	Delivered	Note
Application Description	🟢	You ended up with a lot of polish for a project done in 10 weeks
Hardware Description	🟢	The hardware description does a great job of showing how everything logically fits together and has a wiring diagram so you can recreate these if you want to
Software Description	🟢	Breakdown of all the libraries and software modules, configuration and operation is comprehensive
Identify written vs reused code	🟢	Good discussion of the licenses used
Architecture Diagrams	🟢	I like the colour coding of the architecture diagrams. Helps to cram extra and useful info into a smaller space. Tiny and insignificant nitpick, the colour for main in the SW diagram isn't defined in the key. Doesn't detract from understanding and there's a strong argument against making the key bigger for a single box.
Build Instructions (HW)	🟢	Wiring diagram provided with specific pins called out
Build Instructions (SW)	🟢	Build system specifies Windows only due to build ID script. Instructions are clear
Debug Instructions	🟢	Able to use the console to do many debugging and configuration options
Future Plans	🟢	Good list of both hardware and software future development tasks to take this past the current phase.
Self Assessment	🟢	Under-assesed in multiple categories.