Executive Summary
Sam Wagner
3/5/21

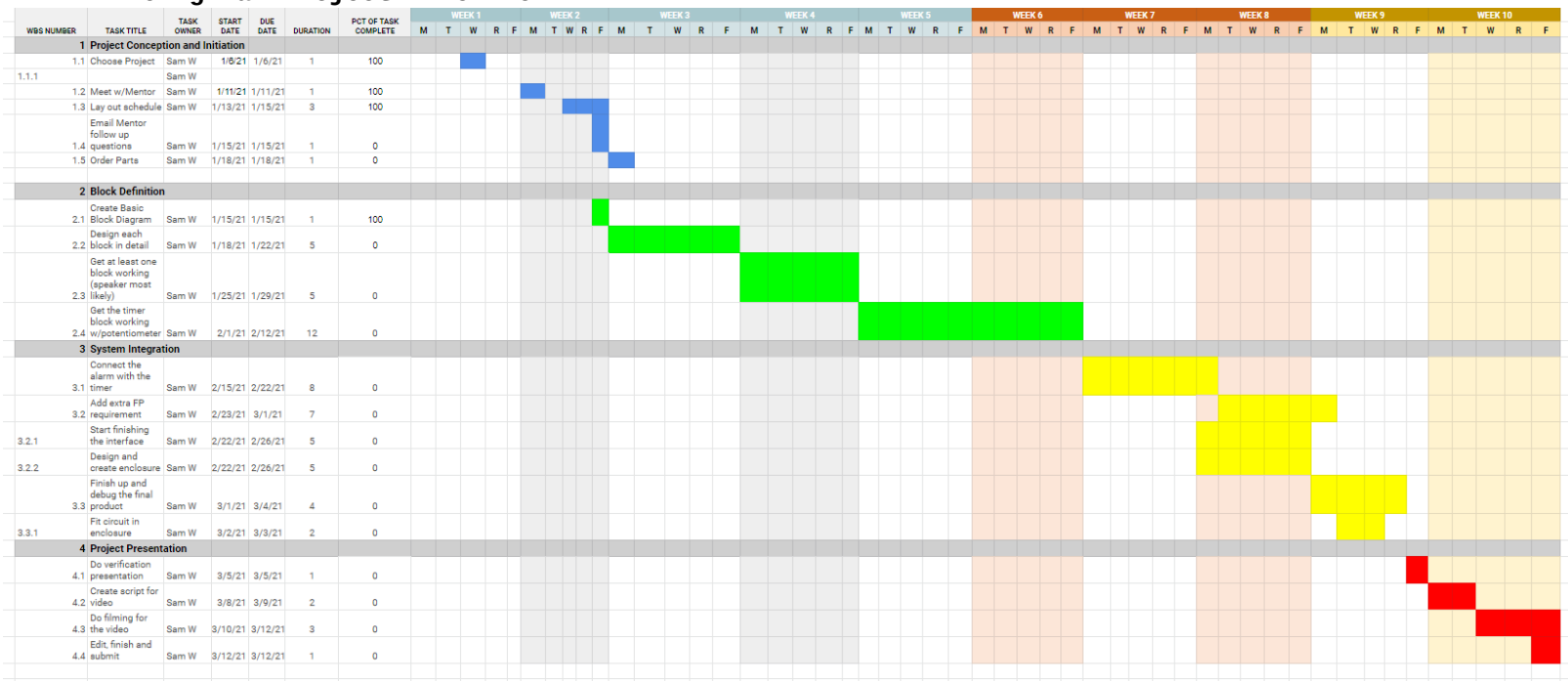
Design Problem: Supposedly a hypothetical client would like to use a timer to hard boil his/her eggs. He/she wants to have two discrete options for 4 and 10 minutes and wants the timer to be accurate. The timer should be accurate within +/-1 second per minute and the display should have 3 discrete brightness levels. When the countdown is finished, the alarm should ring at 440 +/- 1 Hz. In addition the client wants an extra feature which is to add an extra minute in the middle of timing if he/she wants to extend their boiling time a little longer.

Development: I was the only one working on this project. The way I approached was to divide the system into blocks and tackle each block. I used a block diagram to split up the blocks. That was really helpful. I first tackled an easy block, just to start making progress. I chose to tackle dimming feature first. Then I focused on getting the central part of the system, which was the timer countdown, to work. This required coding a long script. When I got the countdown feature to work, I focused on integrating the other blocks into my design. The last step was to do the non-technical parts, such as designing an enclosure and building a mock version of one. I was able to finish the project this way.

Revisions: I was originally going to use USB power to power my timer. However, I began the development with my JD Power Supply which takes in wall power and steps it down to a usable value for the Arduino Nano. I eventually decided that I would abandon the USB power idea and stick with wall power. One improvement for the future I would want to make is to make the timer without the Arduino. The Arduino is a great tool, but it's a little bit overkill. I'd like to design a chip specifically made for the timer and attach it to a PCB. That way I'd have less wires and more sturdy connections.

Key Lessons: One key lesson that I learned is to practice good time management. If you don't spread your work out, you end up with a large pile of it at the end. This can be avoided by doing just a little bit everyday. It doesn't have to be much, it can be just 30 minutes of staring at some code, trying to figure out why it doesn't work. I learned that you should document your project along the way. Even though it's kind of a hassle, it's still better than doing it all in the final week.

Original Project Timeline:



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Conception and Initiation**					
1.1	Choose Project	Sam W	1/5/21	1/6/21	1	100
1.1.1		Sam W				
1.2	Meet w/Mentor	Sam W	1/11/21	1/11/21	1	100
1.3	Lay out schedule	Sam W	1/13/21	1/15/21	3	100
1.4	Email Mentor follow up questions	Sam W	1/15/21	1/15/21	1	0
1.5	Order Parts	Sam W	1/18/21	1/18/21	1	0
2	**Block Definition**					
2.1	Create Basic Block Diagram	Sam W	1/15/21	1/15/21	1	100
2.2	Design each block in detail	Sam W	1/18/21	1/22/21	5	0
2.3	Get at least one block working (speaker most likely)	Sam W	1/25/21	1/29/21	5	0
2.4	Get the timer block working w/potentiometer	Sam W	2/1/21	2/12/21	12	0
3	**System Integration**					
3.1	Connect the alarm with the timer	Sam W	2/15/21	2/22/21	8	0
3.2	Add extra FP requirement	Sam W	2/23/21	3/1/21	7	0
3.2.1	Start finishing the interface	Sam W	2/22/21	2/26/21	5	0
3.2.2	Design and create enclosure	Sam W	2/22/21	2/26/21	5	0
3.3	Finish up and debug the final product	Sam W	3/1/21	3/4/21	4	0
3.3.1	Fit circuit in enclosure	Sam W	3/2/21	3/3/21	2	0
4	**Project Presentation**					
4.1	Do verification presentation	Sam W	3/5/21	3/5/21	1	0
4.2	Create script for video	Sam W	3/8/21	3/9/21	2	0
4.3	Do filming for the video	Sam W	3/10/21	3/12/21	3	0
4.4	Edit, finish and submit	Sam W	3/12/21	3/12/21	1	0

This is my original timeline. Some things changed, but it is a good representation of the progress I made over the course of 8 weeks.