University of Cincinnati
Department of Electrical Engineering and Computing Systems

Felix: A *Better* Embedded Systems Development Kit

Ryan Michael Norton

April 18, 2019

Submitted in partial fulfillment of the degree of

Bachelors of Science in
Electrical Engineering

Ryan Michael Norton	Jason Heyl	Dr. Carla Purdy
Team Member	Technical Advisor	Technical Advisor

DEDICATION

To the various students, teachers, and mentors who have helped me along the way. I am lucky in that you are too numerous for me to thank each and every one of you individually.

To Dr. Carla Purdy, thank you for your kindness and flexibility in allowing me to work on this project remotely and especially thank you for sharing such wonderful stories of your career and life.

To Dr. Jason Heikenfeld, thank you for your support and guidance. Without it, I would not have been able to undertake this project while traveling and studying abroad.

To my friends, Nicholas and Sarah Twine, Tristan Morphew, Adina McClelland, Dr. Andrew Jajack, Alva Webster, Mikaila Wenker, Sarah Frey, and Michael Takahashi, thank you all for your constant support when I was down.

To Zach Matson, thank you for always being the voice of reason and pushing me to improve education by simplifying resources and expanding educational opportunities. And thank you for being my mentor and close friend.

To my parents, thank you for your constant support and guidance when times were hard.

To Dr. Teri Murphy, thank you for your kind and ample enthusiasm for this project, even in the midst of personal challenges (and triumphs!).

To Dr. Rui Dai, thank you for taking a risk on me and allowing your students to be test subjects for this project as part of their curriculum and for even considering this as a viable replacement.

To my students, thank you for being test subjects and for giving honest feedback.

… and to everyone whose name I could not mention. Your sacrifice does not go unrecognized.

Thank you all.

Dedication 2
TABLE OF CONTENTS **3**

Abstract 4

INTRODUCTION **4**
Problem/Need 4
Solution 4
Credential 5
Project Goals 6
Methodology 6

DISCUSSION **8**
Project Concept 8
Design Objectives 8
Technical Approach 10
Budget / Bookkeeping 11
Bill of Materials 12
Timeline 13
Problems Encountered 14
Future Recommendations 14

CONCLUSION **15**
Annotated Bibliography 15
Appendix A: Supporting Documents 16
Appendix B: Design Files 29
Appendix C: Datasheets 32
Appendix D: Test Protocols 45

LIST OF FIGURES

Figure 1. Objective Tree 12
Figure 2. Bookkeeping 13
Figure 3. Bill of Materials 14
Figure 4. Gantt Chart 15
Figure 5. Panel A Front and Back 31
Figure 6. Panel B Front and Back 31
Figure 7. 4x7 Segment Display 33
Figure 8. LED Array 33
Figure 9. Rotary Encoder 33
Figure 10. GPIO Array 33

Figure 11. Potentiometer 33
Figure 12. Real Time Clock 33

Abstract

At the University of Cincinnati, all electrical and computer engineers have to take a course titled Embedded Systems Development that requires them to purchase an expensive and out of date system. The proposed system is dramatically cheaper and more powerful and expands the educational opportunities of the course.

INTRODUCTION

Problem/Need

Students in Embedded Systems Design currently pay too much for a rigid system that limits the educational opportunities of the course. The current system involves two development kits (Parallax BasicStamp HomeWork Board and Microchip Curiosity Board) that are limited in power and architecture.

Ten undergraduate electrical and computer engineering students, a mix of both current and alumni, were asked about their opinions on the current system and suggestions for a suitable platform. All resented the current kits for their cost and limitations, such as computational power constraints. Of the three professors surveyed, there were many requests: more computational power, specifically enough for an FFT; further life beyond the course, specifically in senior design; and cost reduction, so students can work individually on projects.

When given better tools to learn, students are set up for greater success. Keeping costs low and supplying better, more powerful tools are ways for both students and the university to stay competitive as education becomes more demanding.

The current system is suboptimal simply due to lack of maintenance and the lack of resources that could be devoted to solving this issue. This is the result of time constraints due to faculty and staff responsibilities, which transfers to the students as a monetary burden.

Solution

The proposed embedded systems development kit is a better solution because it uses a modern architecture chip that is likely to be on the market longer and has greater computational capabilities (operating frequency, memory, amount of GPIO, peripherals). Its modularity helps to reduce cost, teaches a new skill to students (soldering), and extends the use of individual components beyond the course. The board assembly process allows students to wire their device once, and then focus on the programming and architecture aspects of embedded systems instead of continually diagnosing wiring problems.

The implemented microcontroller should have the following capabilities:

- Hardware communication (UART, I2C, SPI, etc.)
- Analog to digital conversion

- Adequate number of GPIO pins
- Low power/sleep
- EEPROM/hard memory storage
- Programmable with Assembly and C/C++

Multiple microcontrollers were considered for this application, including the ARM Cortex-M0, ATmega328P, Cypress PSoC, Intel 8051, PIC 16F1619, and PIC 18F24K40[3]. They were evaluated based on cost, computational power, memory size, peripherals, lifetime, applicability to coursework, industrial applications, and usability. The PIC 18F24K40 was selected for its broad superiority in these categories.

The boards will be composed entirely of through-hole components so that students with limited to no soldering experience will be able to assemble their boards with limited difficulty. The soldering should not affect the current timeline of the course, as lab currently does not take place the first few weeks of the semester. This time could be used for teaching soldering and assembling boards. Since students will assemble their own boards, they will have the necessary skills to replace individual components in the event of a failure. Having a modular design furthers this compartmentalization of failure by limiting what parts can individually break in the system and reducing the likelihood and cost of a failure. If a part does break, it will not compromise the entire system, and the individual component or module is likely replaceable instead of replacing the whole system.

The solution will be a set of design files representing all the printed circuit boards, a bill of materials listing all the necessary components, documentation of all circuit schematics and layouts, and a set of laboratories to test students' abilities to perform specific skills. These items could be sold as a complete kit at scale or purchased at the individual level by students direct from manufacturers/suppliers. All components are easily sourced through major suppliers such as Digikey, Mouser, or Element14, and PCBs can be ordered directly from a multitude of low-volume board houses such as JLB PCB, OSH Park, or PCB Way.

The university could also purchase components at reduced prices, dropping the overall price for students significantly and providing an immediate stock of components if a replacement part is needed.

Credential

This project will require more than basic engineering design skills, but also an understanding of the logistics and psychology involved in the educational process. Knowledge of useful pedagogies and empathic design processes will assist in producing a product that is useful to the final user, the students.

Ryan Michael Norton:

University of Cincinnati, Class of 2019: BS. Electrical Engineering (Minor: Mathematics)

- Lecture Assistant for Calculus II

- Embedded System Design (Grade Received: A)
- User-Centered Design (Grade Received: A-)
- STEM Education and Practices (Grade Received: A)

Big Ass Fans: Research and Development Engineer for Control Systems (Summer 2016)
- Led multidisciplinary team to develop a high bandwidth data logging platform/kit for non-engineering researchers within the company
- Developed hardware solutions for customs communication with industrial systems

Happy Nerd Company, LLC: Owner / Operator (Spring 2016 – Present)
- Develop hardware solutions for low risk medical environments and data logging
- Full stack web developer/designer with focus in eCommerce and business growth

University of Cincinnati, Novel Device Lab: Engineering Research Assistant (Spring 2017)
- Built test rigs for measuring and limiting effects of humidity on hydrophilicity of chemically functionalized surfaces
- Award for Best Co-op Research Presentation because of innovative style of presentation

Tesla Motor Company, Gigafactory 1: Design Management Engineer (Summer 2018)
- Developed statistical models to determine factory failure modes and likelihoods
- Developed advanced staffing model to predict hiring needs over the next decade

Project Goals

Students who take embedded system design are currently required to purchase kits that have little to no use beyond the scope of the course, are expensive, and limit the educational opportunity the course can offer.

The proposed solution is a cheaper, more flexible, more powerful kit that helps keep student cost low, improves the educational opportunities, and encourages use beyond the scope of the course.

Methodology

Pedagogical Approach

The Dreyfus model of learning[2] states that, when learning a new subject, there are five stages of thought development all people exhibit:

1. Novice – needs exact instructions
2. Advanced Beginner – can work on their own, has difficulty troubleshooting/diagnosing
3. Competent – works from concepts, troubleshoots, and solves problems
4. Proficient – understands larger context within the problem
5. Expert – seeks new knowledge, writes books, does research

Ideally, students would go from novice to expert in the span of three months, but that is just too much to ask of any individual. It would be considered a resounding success to get all students

from novice to competent. With enough information, students can then make informed decisions to continue their studies in embedded systems or pursue other paths.

The point of this model is not to create a pedagogical goal, rather to show that there are steps that all students will naturally follow on their path to competency. The educational system should facilitate this kind of growth, not fight it.

Current System

The BasicStamp Homework board is a small surface mounted system with 16 digital-only IO pins that exclusively runs PBASIC, a version of BASIC designed to run on the PIC system. The ideology behind this board is that students do not have to learn complex programming early on and can focus on embedded systems control schemas. Unfortunately, because of the simplicity of the board, it is slow, can not hold a large program, and is limited in capability. The nature of this board ultimately limits students down the line by wasting their time and money on a limited system.

The Microchip Curiosity board runs modern C++/Assembly instead of BASIC. Unfortunately, the development board trades off convenience for flexibility and power. The In-Circuit Serial Programmer (ICSP), which can be expensive, is built into the board, limiting its use to program only the chip on the Curiosity board. This limitation, combined with others such as pre-wired peripherals (i.e. sensors/output permanently wired to certain IO), can unnecessarily confuse students and make important topics seem abstract.

What the BasicStamp Homework board gets is that students require an elementary understanding in order to begin to learn. While its simplicity allows for a low barrier for entry, it also means that as students progress and enter the ‘Advanced Beginner’[2] stage, they quickly outgrow the board, requiring more control of and complexity in the system. The current response to this is to then transition to using the Curiosity board.

The problem is that the Curiosity board is too much of a jump, as it is really designed for use by those in the ‘Competent’[2] stage and above. Predetermined IO and a built-in programmer are two examples of things that can frustrate and confuse students only in the ‘Advanced Beginner’[2] stage of their embedded systems education. The system needs to grow as the student does and present new challenges as the student reaches new successes. It has to match them at every stage. Too far ahead and students won’t be able to engage, too far behind, and the students will become disinterested.

The proposed system provides better value and educational opportunity for students. The lower cost of the components helps students achieve a greater return on investment for their education.

Modularity increases the value of the kit as well as the flexibility and number of uses, even at its reduced cost. This compartmentalizes failures of the system and facilitates replacement of individual parts, allowing for repair of the system.

Since the system is has more capability, teachers can challenge students further, improving the educational experience for both the teacher and the student. Students will be exposed to the important skill of soldering, while teachers are free to add/remove different modules in order to improve and refine the system without creating significant changes to the curriculum.

DISCUSSION

Project Concept

A cheaper, custom development kit designed with cost and power in mind would help alleviate financial strain on students and provide a more open platform for development, exploration, and education within the purview of embedded systems. Given the low cost of custom printed circuit boards and electrical components, a kit could easily be designed to significantly lower costs and improve capabilities.

Design Objectives

Objectives

- Lower cost from current system
- Raise value compared to current system
- Increase computational power (for students to explore more)
- Be safe and easy to use
- Expand educational goals of the course
- Keep students engaged with low level programming techniques and principles

Functions

- Reading Datasheets
- Assembly
- C/C++
- Controlling I/O
    - Digital Inputs
        - Debouncing
        - Reading States
    - Digital Out
    - Analog In
        - Analog to Digital Conversion
    - Analog Out
        - CCP/PWM
- Interrupts
- Communication

    - Serial (USB to Serial)
    - I2C
- Memory
    - EEPROM
- Timers
- Power Consumption
    - Sleep Mode
    - Battery Life
- Skills
    - Linear Mapping of Numbers
    - Bit Shifting/Bit Masking
    - Soldering
    - Control Schemas
        - Multiplexing
        - Servos
    - PCB Layout, Milling, and Sourcing
- Easily portable
- Useful beyond the scope of educational courses (Reusable)
- High Computational Power

Constraints

- Low Cost (<$50)
- Safe to use
- Solderable by Amateurs

Means

- Modular design through PCB compartmentalization
- Only through hole components and large surface mounted components
- Simple circuits without extraneous protection
- Low voltage (5V-3V3)
- PIC Processor

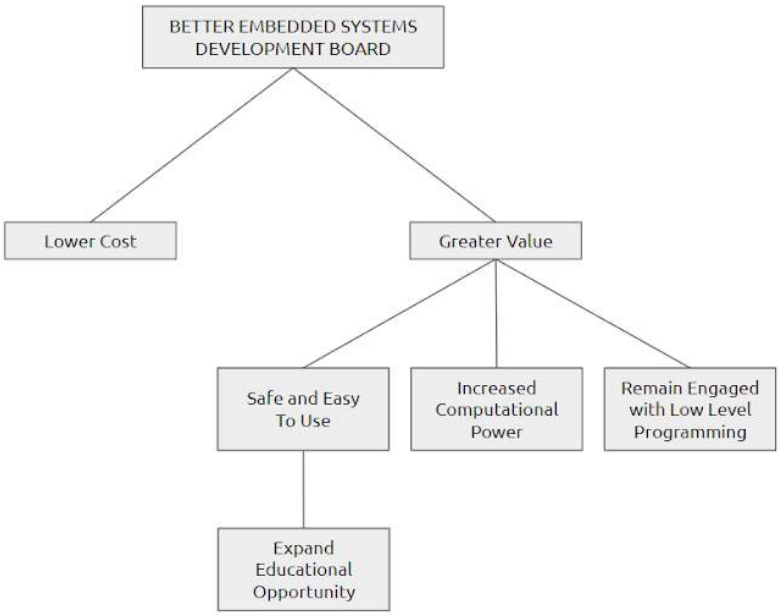


Figure 1. Objective Tree

Technical Approach

Felix was designed to improve a course that relied on an old and inflexible development system to teach embedded systems. The goal was a more powerful, more intuitive, and cheaper embedded systems development kit. It is similar to an Arduino Uno but with more power and control.

Over the course of several months, constraints and goals were specified, the board layout was paper crafted, countless design iterations, renderings, and critique sessions were conducted. The modular system that was developed is already generating excitement from professors and engineering students alike.

To test this system, a small group of students will be introduced and observed using it, with the help of Dr. Carla Purdy and Dr. Rui Dai at the University of Cincinnati. This project will improve the education of electrical and computer engineering students as they engage with embedded systems and their applications. Not only will the proposed system provide more power and capabilities, it does so at a lower cost for students.

This helps to lower the barrier for entry and potentially open the course to students that otherwise could not afford to participate. Improving the education process ripples out into industry *and* academia.When people are given the tools and opportunities to be successful, society at large reaps the rewards of their discoveries and innovations.

Budget / Bookkeeping

Order Description	Purchaser	Supplier	Cost	Ship	Tax	Tarriff	Total	Category
Breadboards and Male Header Pins	Ryan Norton	Amazon	$ 18.98	$ -	$ -	$ -	$ 18.98	Circuit Boards
Various Components	Happy Nerd Company, LLC	Digi-Key Electronics	$ 222.14	$ 14.55	$ 14.20	$ -	$ 250.89	Components
Various Components	Happy Nerd Company, LLC	Farnell UK	$ 81.62	$ 13.66	$ -	$ -	$ 95.28	Components
Circuit Boards version 2	Ryan Norton	JLC PCB	$ 51.00	$ 35.93	$ 45.03	$ -	$ 131.96	Circuit Boards
Circuit Boards version 3	Ryan Norton	JLC PCB	$ 44.00	$ 20.87	$ -	$ -	$ 64.87	Circuit Boards
Circuit Boards version 4	Ryan Norton	JLC PCB	$ 7.00	$ 24.97	$ -	$ -	$ 31.97	Circuit Boards
Screwdriver	Ryan Norton	Wilko	$ 0.46	$ -	$ -	$ -	$ 0.46	Tools
Various Components	Happy Nerd Company, LLC	Digi-Key Electronics	$ 50.89	$ 3.06	$ -	$ 24.72	$ 78.67	Components
Circuit Boards version 1	Happy Nerd Company, LLC	JLC PCB	$ 3.97	$ 37.34	$ -	$ -	$ 41.31	Circuit Boards
Digital Multimeter	Ryan Norton	Wilko	$ 15.63	$ -	$ -	$ -	$ 15.63	Tools
220V Soldering Iron, Screwdrivers	Ryan Norton	Wilko	$ 27.94	$ -	$ -	$ -	$ 27.94	Tools
Flush Cutters	Ryan Norton	Amazon	$ 7.81	$ -	$ -	$ -	$ 7.81	Tools
Various Components	Ryan Norton	Digi-Key Electronics	$ 141.41	$ 14.85	$ -	$ -	$ 156.26	Components
						Total Project Cost	***$ 922.03***	
						Budget	*$ 1,000.*	
						Remaining	*$ 77.97*	

Figure 2. Bookkeeping

Bill of Materials

Part	Value	Manufacturer Part Number	Unit Cost	No.	Cost
Display	4x7 Segment	LTC-4727JR	$3.96	1	$3.96
Female Header	6 Pin	PPTC061LFBN-RC	$0.52	7	$3.64
Resistors	1k, Small	CF18JT1K00	$0.10	30	$3.00
UART USB Bridge	MCP 2221	MCP2221-I/P	$2.60	1	$2.60
Smoothing Caps	0.1 uF	RDER71H104K0P1H03B	$0.29	7	$2.03
Male Header	6 Pin	M20-9770646	$0.25	8	$2.00
LEDs	Green	151031VS06000	$0.16	12	$1.92
USB Port	Mini B	548190519	$1.64	1	$1.64
LEDs	Red	151031SS04000	$0.18	9	$1.62
Microcontroller	PIC18F24K40	PIC18F24K40-I/SP	$1.52	1	$1.52
Rotary Encoder	EN11-HSM1AF15	EN11-HSM1AF15	$1.25	1	$1.25
Switch	2PDT	MHSS1104	$0.54	2	$1.08
Real Time Clock	MCP 7940M	MCP7940M-I/P	$0.66	1	$0.66
Potentiometer	10k	P120PK-Y25BR10K	$0.63	1	$0.63
Oscaillltor Caps	15 pF	SR152A150KAR	$0.31	2	$0.62
Decoder	3x8 Active Low	CD74HC238E	$0.60	1	$0.60
Buzzer	WT-1205	WT-1205	$0.54	1	$0.54
Boards	Assorted	---	$0.50	1	$0.50
Crystal Oscaillator	32.768 kHz	CFS-20632768EZBB	$0.46	1	$0.46
Voltage Regulator	3V3 LDO	AZ1117EH-3.3TRG1	$0.43	1	$0.43
Angled Male Header	6 Pin	M20-9960645	$0.35	1	$0.35
IC Socket	28 DIP	A 28-LC-TT	$0.34	1	$0.34
DIP Socket	16 DIP	A 16-LC-TT	$0.25	1	$0.25
IC Socket	14 DIP	A 14-LC-TT	$0.22	1	$0.22
Mom. Button	SPST	1825910-7	$0.11	2	$0.22
IC Socket	8 DIP	A 8-LC-TT	$0.18	1	$0.18
				Total	***$32.26***

Figure 3. Bill of Materials

Timeline

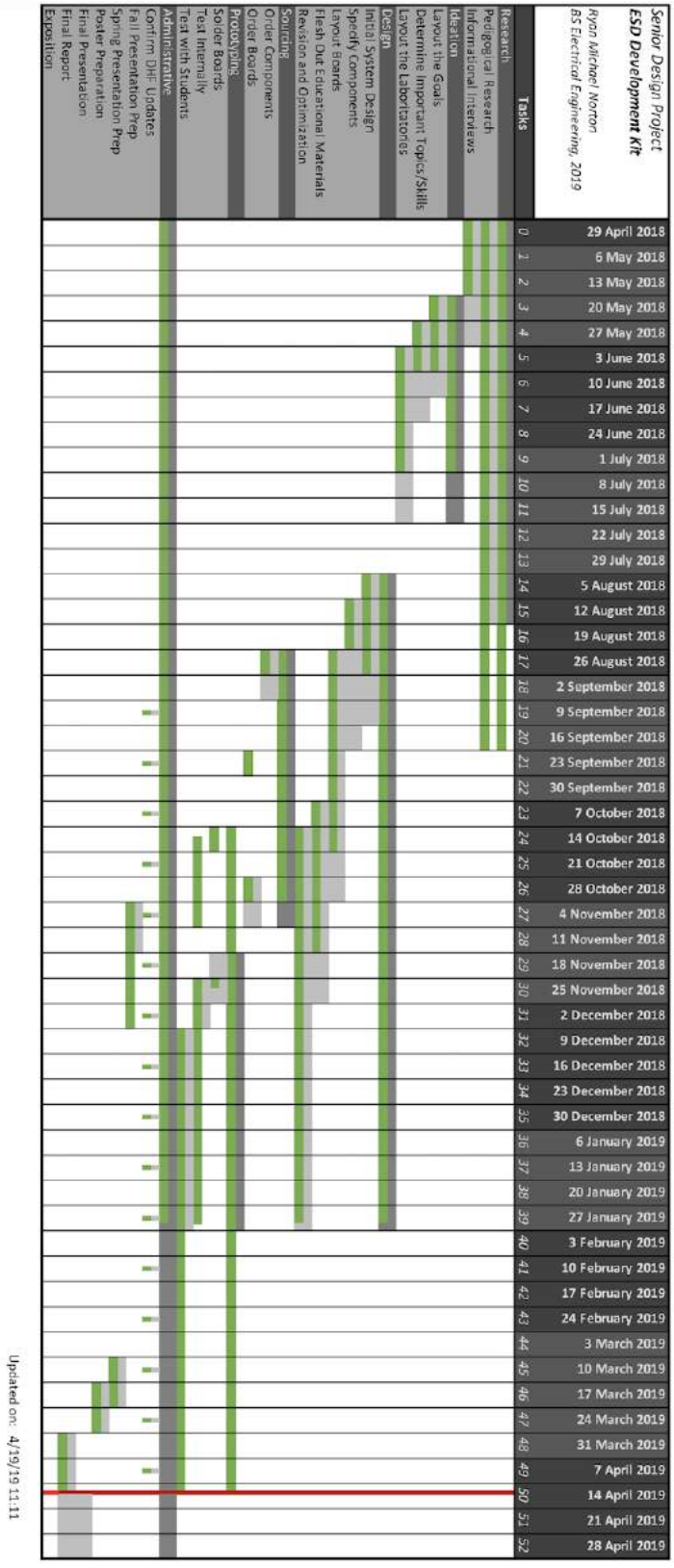


Figure 4. Gantt Chart

Problems Encountered

Wireless Communication

It would have been ideal to have a Bluetooth or Wifi module to interface via a serial port but it conflicted with the need to have all through hole components that students could easily solder by hand with little to no soldering experience.

Voltage Regulator

An economic low voltage dropout regulator could not be found in a suitable through hole format so a large format surface mounted device was chosen.

Boardhouse Limitations

Most low-volume board houses for printed circuit board manufacture limit board size to 10 cm by 10 cm. This conflicted with early designs that had integrated sensors on board as well as a breadboard for easy prototyping. This limitation was eventually overcome when the size of the board shifted to a 2.2” square with 1” square modules.

PICkit3 Power Limitations

The Felix board cannot have every sensor port occupied when programming the device at 5V or 3.3V. The PICkit3 can only provide 200 mA of current to the board during programming and this limit can be exceeded especially when LED arrays are in use as each LED consumes a large amount of current despite a large current limiting resistor.

Four Digit Seven Segment Display Brightness

When current limiting resistors were used with the 4x7 segment display, segments appeared too dim during multiplexing. To counter this, no current limiting resistors were used and current limiting now relies on abusing the decoder output driver limitations, which may shorten the life of the integrated circuit.

Rotary Encoder Noise

The low cost rotary encoders purchased had too much noise on the data lines to be used reliably. By adding a small RC filter to each of the data lines, most of the noise is mitigated.

Future Recommendations

Increased Complexity with Subassemblies

To increase complexity of the board and lower cost, it is recommended to switch to surface mounted subassemblies. Students could still be required to solder headers and other large components onto the subassemblies, but it would expand the kinds of circuits they could utilize (wireless communications, gyro/accelerometers, etc) while also shrinking the amount of time it takes to put a kit together.

CONCLUSION

Annotated Bibliography

[1] Ferris, Timothy (2012) *The 4-Hour Chef: The Simple Path to Cooking Like a Pro, Learning Anything, and Living the Good Life.* Seattle, WA: New Harvest Publishing

Specifically in the "Meta-Learning" section of this text, Ferriss discusses how students learn and points out specific pitfalls to the learning process and how to avoid them. While not every challenge can be mitigated, by considering them at the conception of this project, the end product is more conducive to learning and aims to avoid common learning mistakes. This is especially true of Ferriss' "DiSSS" breakdown :

- Deconstruction - defining the minimum viable product to be successful
- Selection - Find the 20% of actions that will result in 80% of successes
- Sequencing - Prioritize and plan what to learn and how
- Stakes - Create consequences for not following the plan that hold personal value

[2] Hunt, Andy (2008) *Pragmatic Thinking and Learning: Refactor Your Wetware*. Raleigh, NC: The Pragmatic Bookshelf

The Dreyfus model of learning states that, when learning a new subject, there are five stages of thought development all people exhibit:

1. Novice – needs exact instructions
2. Advanced Beginners – can work on their own, has difficulty troubleshooting/diagnosing
3. Competent – works from concepts, troubleshoots, and solves problems
4. Proficient – understands larger context within the problem
5. Expert – seeks new knowledge, writes books, does research

Ideally, students would go from novice to expert in the span of three months, but that is just too much to ask of any individual. It would be considered a resounding success to get all students from novice to competent. With enough information, students can then make informed decisions to continue their studies in embedded systems or pursue other paths.

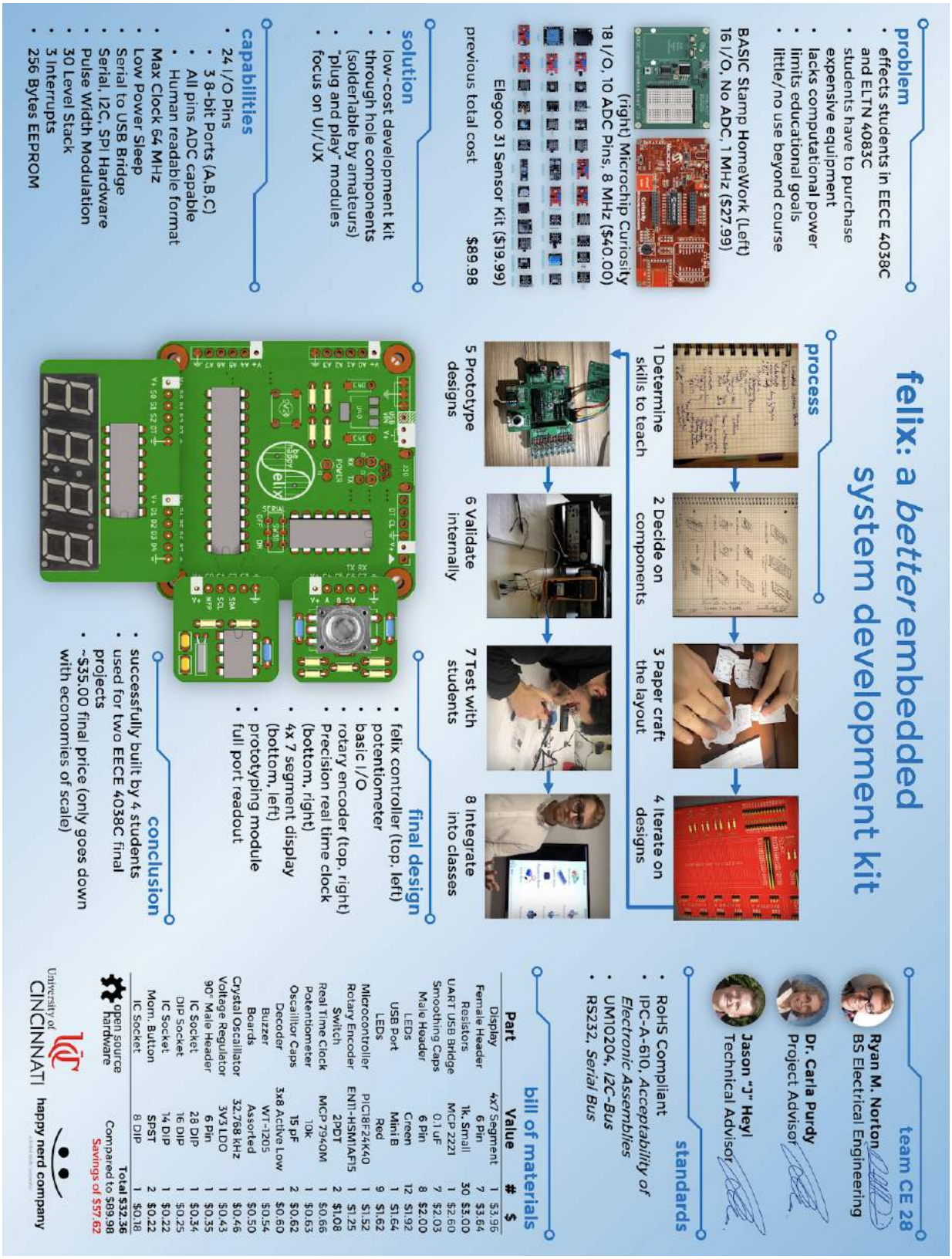
[3] Microchip (2012-2018) *PIC18F25K40.* Chandler, AZ: Microchip Technologies, Inc.
Retrieved from: https://www.microchip.com/wwwproducts/en/PIC18F24K40

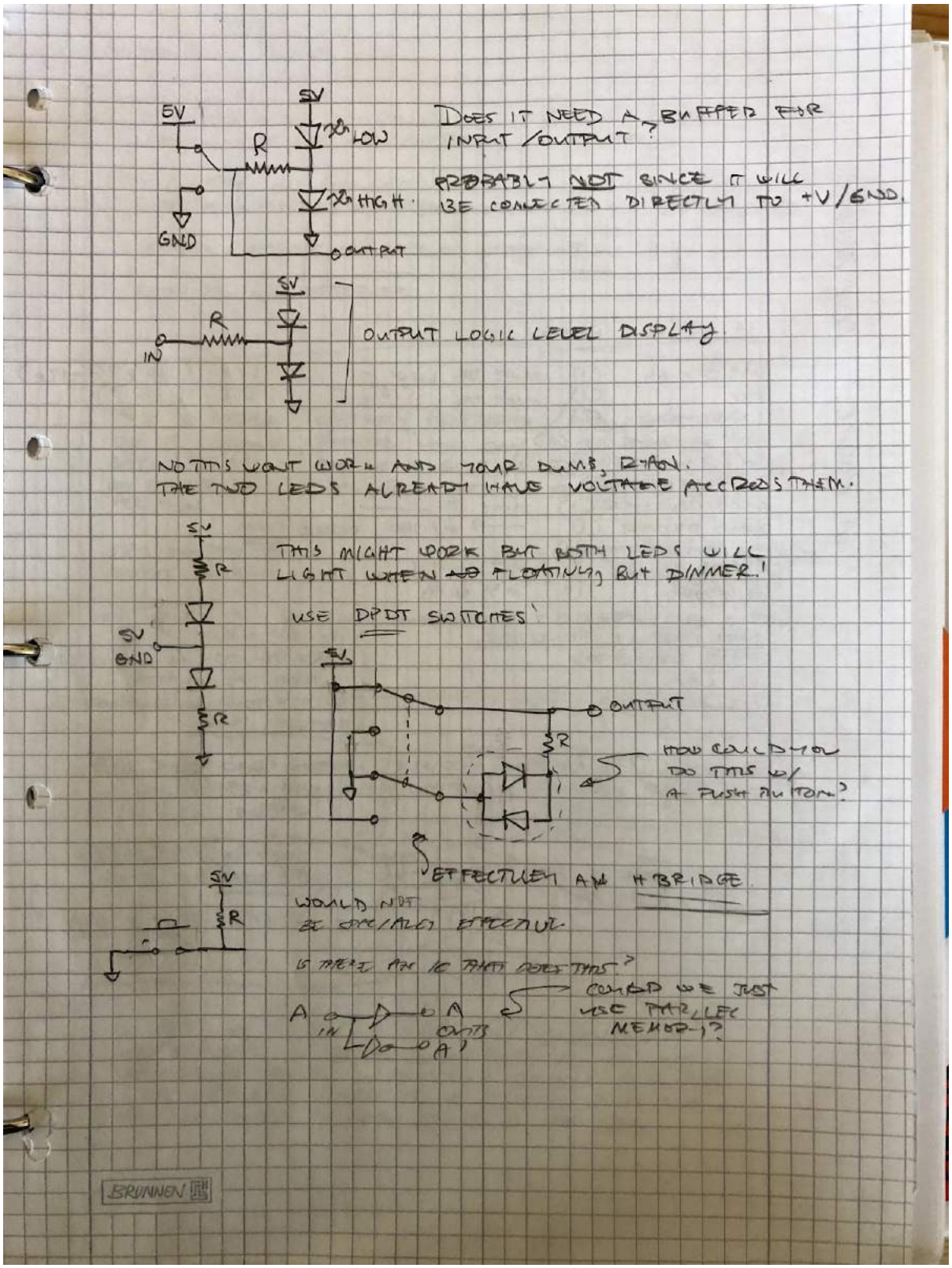
This document was used as a reference for planning out hardware interactions, potential sensors, and determining the placement of components on the circuit board. It also allowed for the determination of capabilities to validate in the device.

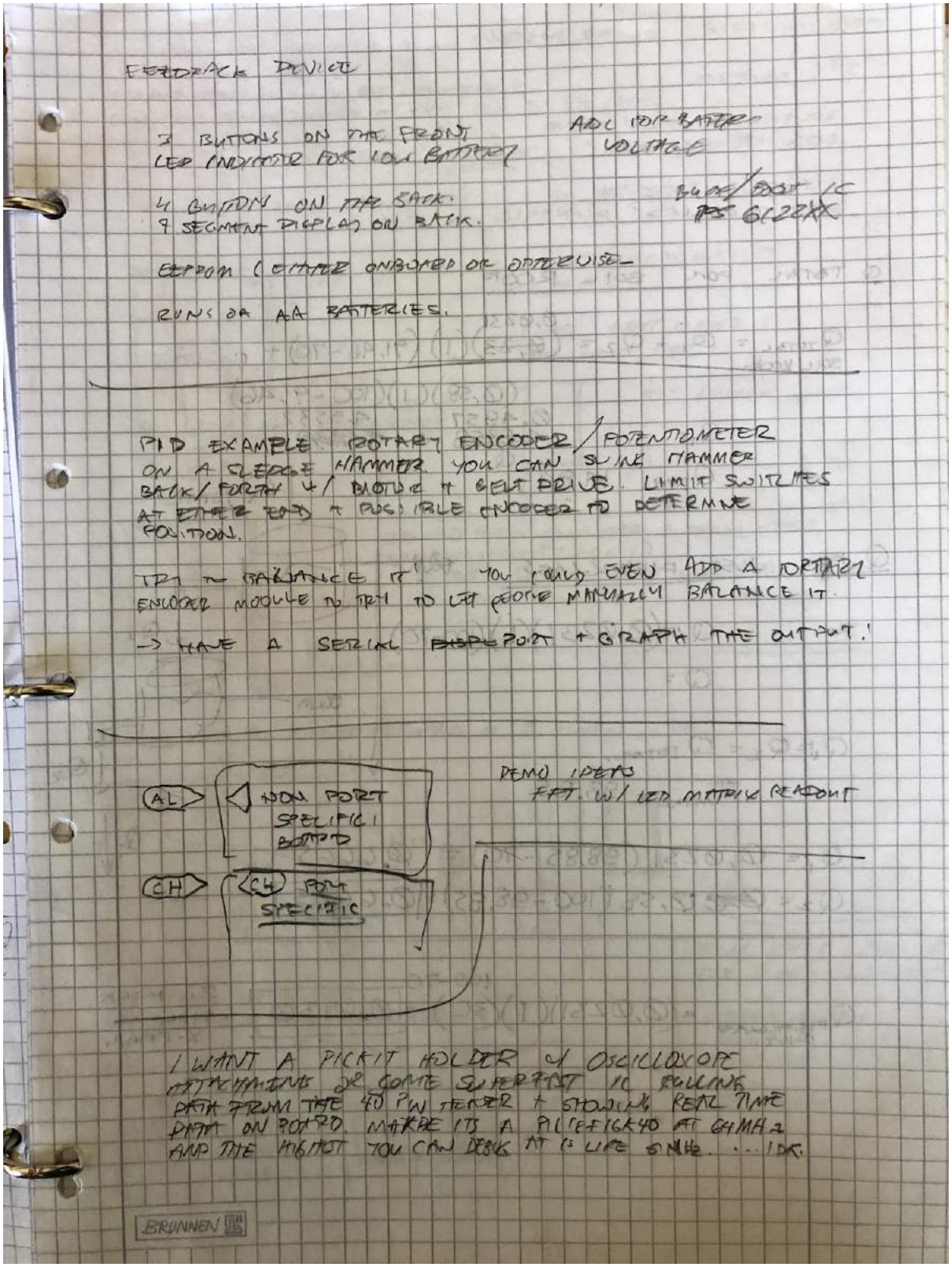
[4] Microchip (2012-2018) *MCP 7940M: Low-Cost I^2C Real-Time Clock/Calendar with SRAM*
Chandler, AZ: Microchip Technologies, Inc.
Retrieved from: https://www.microchip.com/wwwproducts/en/MCP7940M

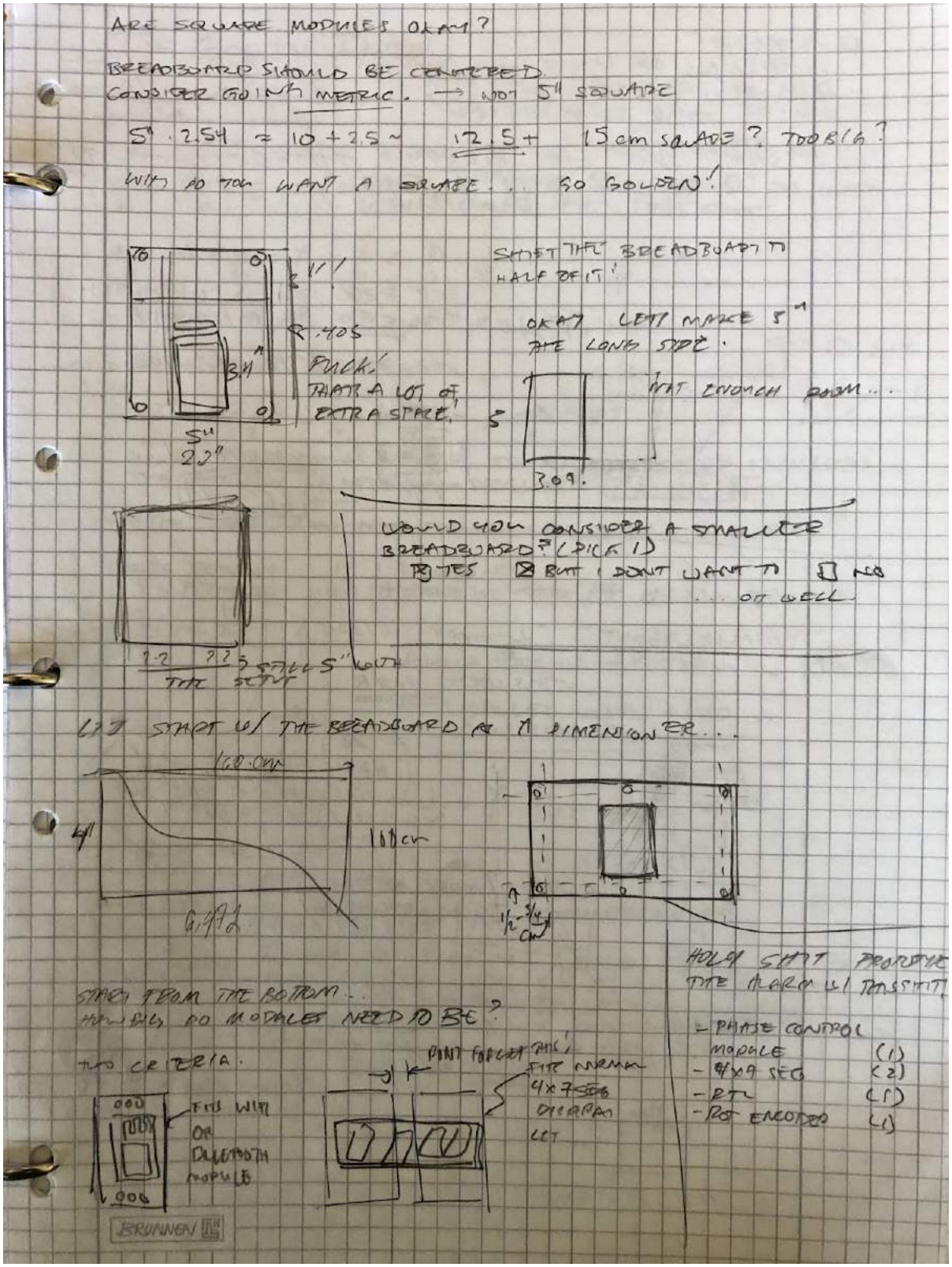
This document provided technical specifications and most importantly memory register maps (see Appendix C) for use in interfacing and validating with the real time clock module.

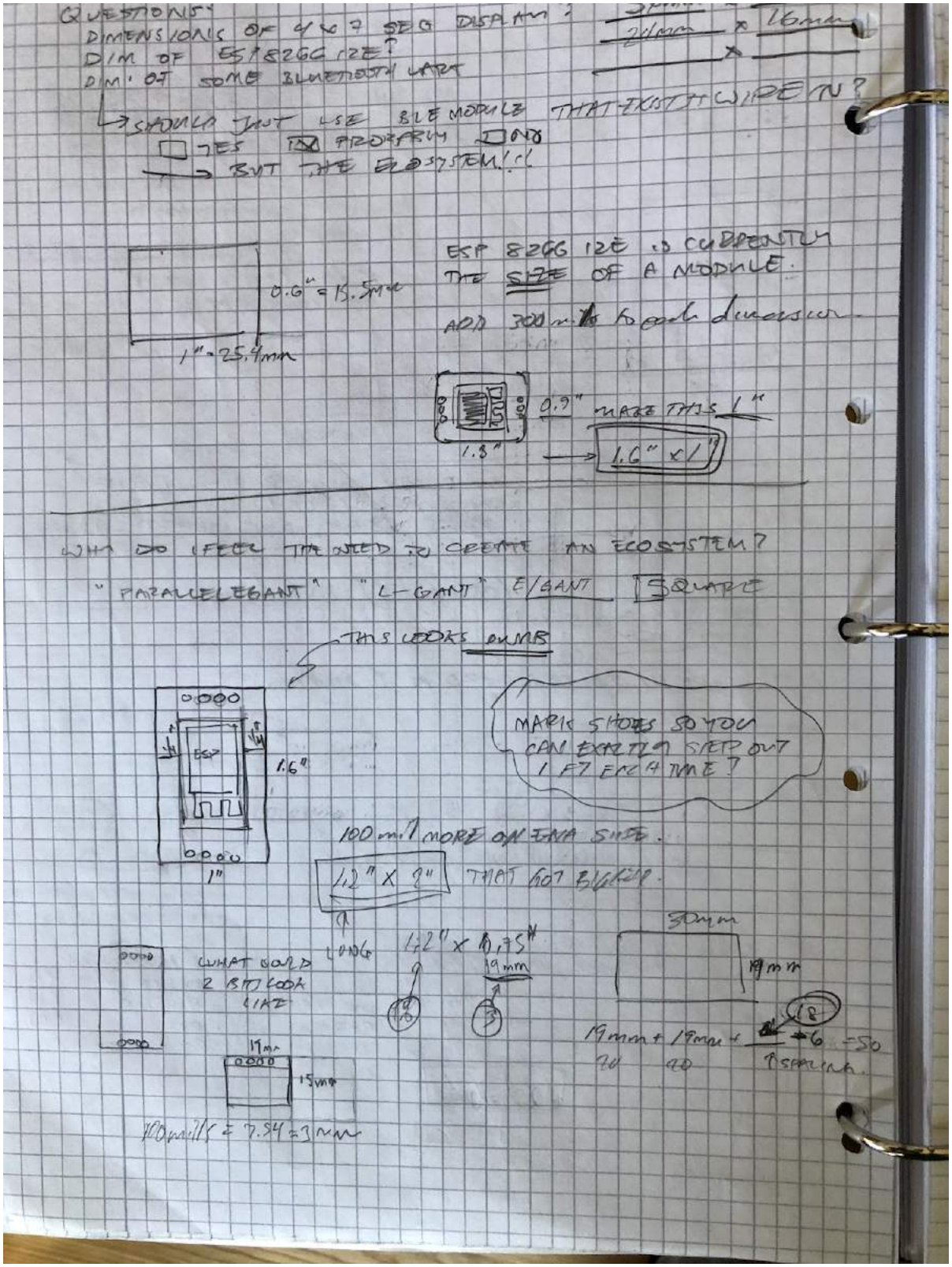
APPENDIX A: Supporting Documents



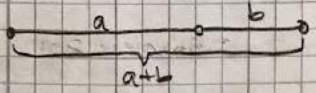

5V
R
LOW
HIGH
GND
OUTPUT
DOES IT NEED A BUFFER FOR INPUT/OUTPUT?
PROBABLY NOT SINCE IT WILL BE CONNECTED DIRECTLY TO +V/GND.
5V
R
IN
OUTPUT LOGIC LEVEL DISPLAY
NO THIS WON'T WORK AND YOUR DUMB, RYAN.
THE TWO LEDS ALREADY HAVE VOLTAGE ACCROSS THEM.
5V
R
5V
GND
THIS MIGHT WORK BUT BOTH LEDS WILL LIGHT WHEN ~~AD~~ FLOATING BUT DIMMER!
USE DPDT SWITCHES!
5V
OUTPUT
R
HOW COULD YOU DO THIS W/ A PUSH BUTTON?
EFFECTIVELY AN H BRIDGE
5V
R
WOULD NOT BE IDEAL/VERY EFFECTIVE.
IS THERE AN IC THAT DOES THIS?
A
IN
OUTS
A'
COULD WE JUST USE PARALLEL MEMORY?


FEEDBACK DEVICE
3 BUTTONS ON THE FRONT
LED INDICATOR FOR LOW BATTERY
4 BUTTONS ON THE BACK.
9 SEGMENT DISPLAY ON BACK.
EEPROM (EITHER ONBOARD OR OTHERWISE-
RUNS ON AA BATTERIES.
ADC FOR BATTERY VOLTAGE
BUCK/BOOST IC
PIC 6C22XX
PID EXAMPLE. ROTARY ENCODER / POTENTIOMETER ON A SLEDGE HAMMER. YOU CAN SWING HAMMER BACK/FORTH + / MOTOR + BELT DRIVE. LIMIT SWITCHES AT EITHER END + POSSIBLE ENCODER TO DETERMINE POSITION.
TRY TO BALANCE IT! YOU COULD EVEN ADD A ROTARY ENCODER MODULE TO TRY TO LET PEOPLE MANUALLY BALANCE IT.
-> HAVE A SERIAL DISPLAYPORT + GRAPH THE OUTPUT!
DEMO IDEAS
FAT. W/ LED MATRIX READOUT
AL
NON PORT SPECIFIC BOARD
CH
CH PORT SPECIFIC
I WANT A PICKIT HOLDER + OSCILLOSCOPE ATTACHMENT OR SOME SUPERFAST IC BLINKING PATH FROM THE 40 PIN HEADER + SHOWING REAL TIME DATA ON BOARD. MAKE IT A PIC16F1640 AT 64MHz AND THE HIGHEST YOU CAN CLOCK AT IS LIKE 5MHz. ... IDK.
BRUNNEN


ARE SQUARE MODULES OKAY?
BREADBOARD SHOULD BE CENTERED.
CONSIDER GOING METRIC. → NOT 5" SQUARE
5" · 2.54 = 10 + 2.5 ~ 12.5+ 15cm SQUARE? TOO BIG?
WHY DO YOU WANT A SQUARE... SO GOLDEN!
PUCK!
THAT'S A LOT OF EXTRA SPACE.
5"
2.5"
SHIFT THE BREADBOARD TO HALF OF IT!
OKAY LET'S MAKE 5" THE LONG SIDE.
3.09"
WHY ENOUGH ROOM...
WOULD YOU CONSIDER A SMALLER BREADBOARD? (PICK 1)
YES
BUT I DON'T WANT TO
NO
... OH WELL
2.2
2.2
5 STILL 5" WIDE
THE SETUP
LET'S START W/ THE BREADBOARD AS A DIMENSIONER...
168 cm
66"
168 cm
6.492
1/2 - 3/4 cm
HOLD SHIFT PROTOTYPE THE ALARM W/ TRANSMITT
- PHASE CONTROL MODULE (1)
- 4x4 SEG (2)
- RTC (1)
- ROT ENCODER (1)
STACK FROM THE BOTTOM.
HOW BIG DO MODULES NEED TO BE?
TWO CRITERIA.
FITS WITH OR BLUETOOTH MODULE
DON'T FORGET THIS, FITS INSIDE 4x7 SEG DIAGRAM LCT
BRUNNEN


QUESTIONS:
DIMENSIONS OF 4x6 SEG DISPLAY?
DIM OF ESP32GC IZE?
DIM. OF SOME BLUETOOTH UART
↳ SHOULD JUST USE BLE MODULE THAT EXISTS WIDELY?
☐ YES ☒ PROGRAM ☐ NO
→ BUT THE ECOSYSTEM!!
0.6" = 15.5mm
1" = 25.4mm
ESP 32GG IZE :D CURRENTLY
THE SIZE OF A MODULE.
ADD 300mil to each dimension
0.9"
MAKE THIS 1"
1.3"
1.6" x 1"
WHY DO I FEEL THE NEED TO CREATE AN ECOSYSTEM?
"PARALLELEGANT" "L-GANT" E/GANT SQUARE
THIS LOOKS OWNS
MAKE SHOES SO YOU CAN EXACTLY STEP OUT 1 FT EACH TIME?
1.6"
1"
100 mil MORE ON EACH SIDE.
1.2" X 1"
THAT GOT BIGGER.
LONG
1.2" x 0.75"
19mm
30mm
19mm
WHAT WOULD 2 BITS LOOK LIKE
19mm
15mm
100mils = 2.54 = 3mm
19mm + 19mm + #6 = 50
40 40 spacing.

THE GOLDEN RATIO



a IS TO b AS a+b IS TO a.

$$\frac{a}{b} = \frac{a+b}{a} \equiv \varphi \text{ (GOLDEN RATIO)} = \varphi, \phi$$

(CONJUGATE) (DEPENDING ON AUTHOR'S PREFERENCE)

$$\frac{a+b}{a} = 1 + \frac{b}{a} = \varphi = \frac{a}{b} \therefore \frac{b}{a} = \frac{1}{\varphi} \therefore 1 + \frac{1}{\varphi} = \varphi$$

$$\rightarrow \varphi + 1 = \varphi^2 \therefore \varphi^2 - \varphi - 1 = 0$$

BY QUADRATIC FORMULA

$$\varphi = \frac{1 \pm \sqrt{5}}{2} = \{1.618033\ldots, -0.618033\ldots\}$$

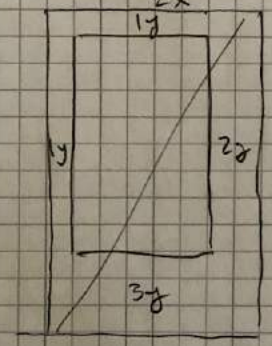
$\phi = 0.618$

$\phi = \varphi - 1$

IN A PREHISTORIC ATTEMPT TO AVOID NEGATIVE NUMBERS THE GOLDEN RATIO WAS DECLARE $\varphi \equiv 1.618$

→ THE VETRUVIAN MAN ONLY USES WHOLE # RATIOS

BOOK DESIGN 1:1:2:3 MARGINS 2:3 PAGES 1:√3 USED IN BWELL



ADOLF ZEISING — PATTERNS IN NATURE

→ CALLED GOLDEN RATIO A "UNIVERSAL LAW"

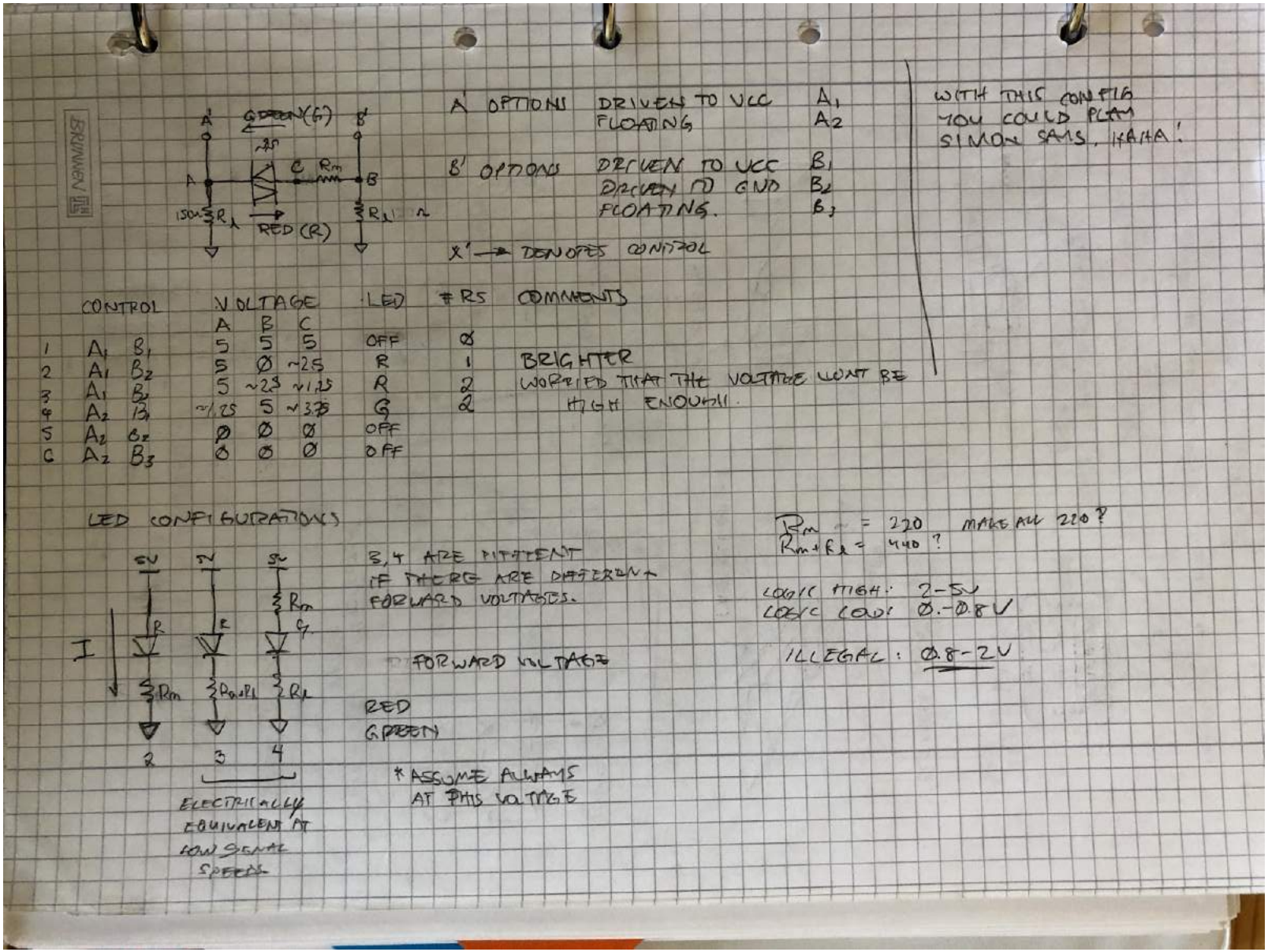
$$\frac{1}{\varphi} = \varphi - 1 \; , \; \frac{1}{\phi} = \phi + 1$$

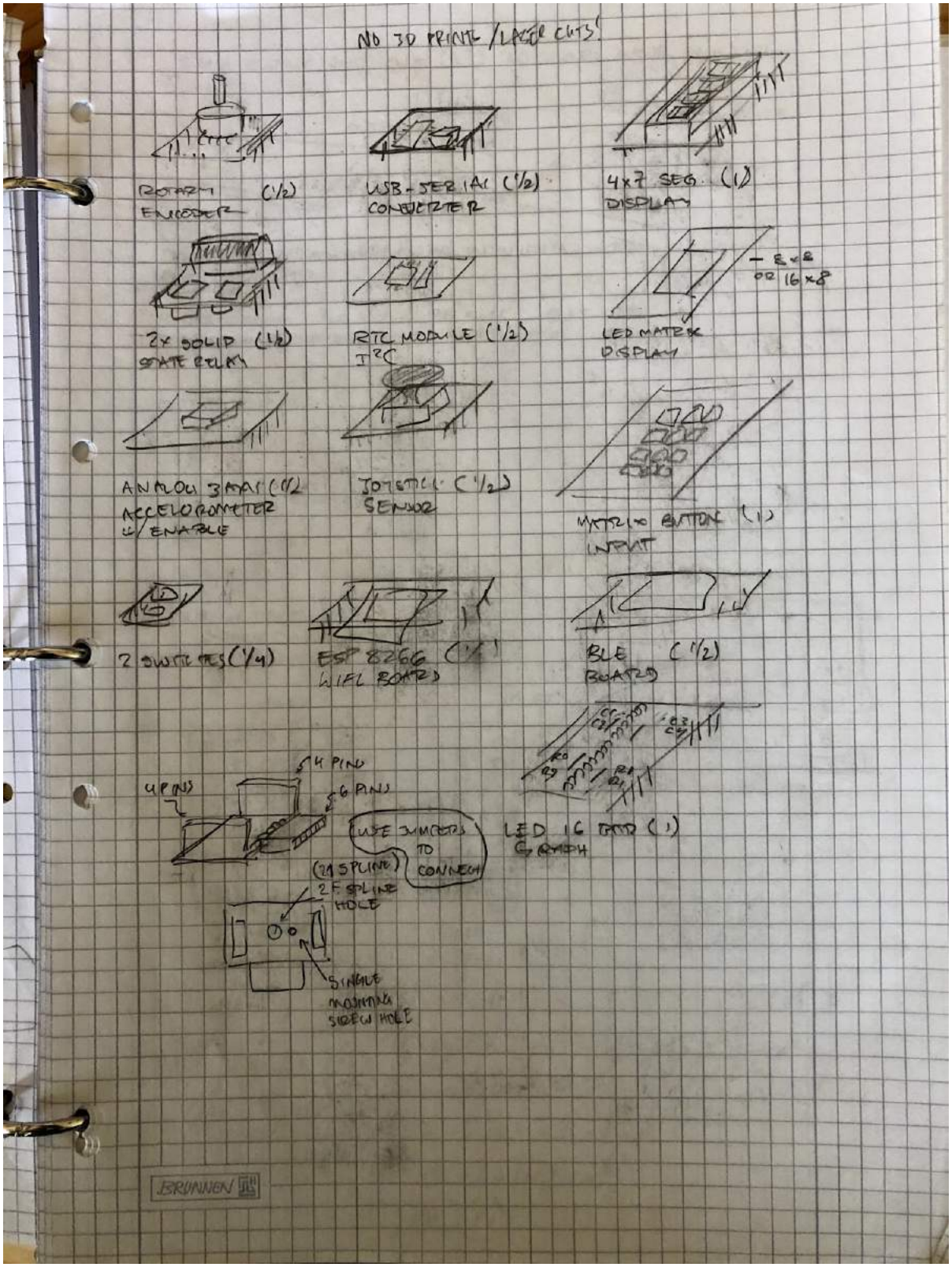
$$\phi : 1 = 1 : \varphi$$

→ TURNS UP IN PENTAGONAL SYMMETRY

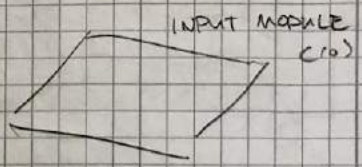
COMES INTO PLAY W/ THOMSON PROBLEM OF EVENLY DISTRIBUTING POINTS ON A SPHERE
→ DIVIDE SPHERE INTO LATITUDES CONTAINING EQUAL SURFACE AREA + ONE NODE ON EACH BAND AT A GOLDEN RATIO OFFSET OF ANOTHER
360° φ = 222.5°

BRUNNEN


A
GREEN (G)
B'
~25
C
Rm
A
B
ISOLATED
RL
RED (R)
RL
A' OPTIONS
DRIVEN TO VCC
FLOATING
A1
A2
B' OPTIONS
DRIVEN TO VCC
DRIVEN TO GND
FLOATING.
B1
B2
B3
X' → DENOTES CONTROL
WITH THIS CONFIG YOU COULD PLAY SIMON SAYS. HAHA!
CONTROL
VOLTAGE
LED
RS
COMMENTS
A B C
1 A1 B1 5 5 5 OFF Ø
2 A1 B2 5 Ø ~2.5 R 1 BRIGHTER
3 A1 B3 5 ~2.3 ~1.25 R 2 WORRIED THAT THE VOLTAGE WONT BE HIGH ENOUGH!
4 A2 B1 ~1.25 5 ~3.75 G 2
5 A2 B2 Ø Ø Ø OFF
6 A2 B3 Ø Ø Ø OFF
LED CONFIGURATIONS
5V
5V
5V
Rm
R
R
G
I
Rm
Rm||RL
RL
2
3
4
ELECTRICALLY EQUIVALENT AT LOW SIGNAL SPEEDS
3,4 ARE PITTIENT IF THEIR ARE DIFFERENT FORWARD VOLTAGES.
FORWARD VOLTAGE
RED
GREEN
*ASSUME ALWAYS AT THIS VOLTAGE
Rm = 220 MAKE ALL 220?
Rm+RL = 440 ?
LOGIC HIGH: 2-5V
LOGIC LOW: Ø.-Ø.8V
ILLEGAL: Ø.8-2V


NO 3D PRINTS / LASER CUTS!
ROTARY ENCODER (1/2)
USB-SERIAL CONVERTER (1/2)
4x7 SEG. (1) DISPLAY
2x SOLID STATE RELAY (1/2)
RTC MODULE (1/2) I²C
8x8 OR 16x8 LED MATRIX DISPLAY
ANALOG 3 AXIS (1/2) ACCELEROMETER W/ ENABLE
JOYSTICK (1/2) SENSOR
MATRIX BUTTON (1) INPUT
2 SWITCHES (1/4)
ESP 8266 WIFI BOARD (1/2)
BLE BOARD (1/2)
4 PINS
6 PINS
USE JUMPERS TO CONNECT
(2) SPLINE 2F SPLINE HOLE
SINGLE MOUNTING SCREW HOLE
LED IC DRIVER (1)
BRUNNEN

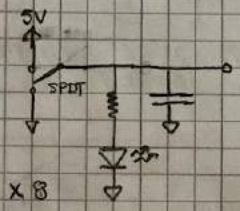
EACH STUDENT GETS ONE (2?) SO GROUPS HAVE MULTIPLES!



DO WE USE LED ARRAYS OR INDIVIDUAL LEDS FOR THE OUTPUT

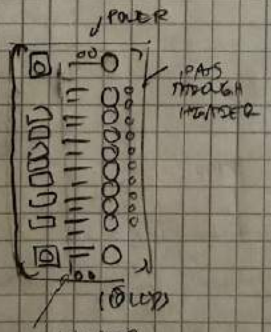
BIDIRECTIONAL WOULD BE NICE BUT IND. IS PROBABLY CHEAPER + FINE FOR DIGITAL WORK

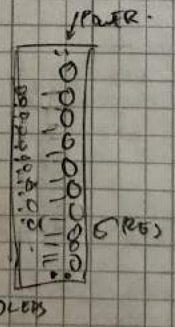
OUTPUT

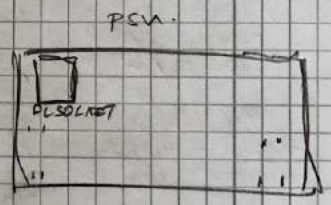


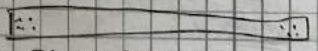
x 8

x 2 w/ TACTILE SWITCH (CLOCK) (RESET)

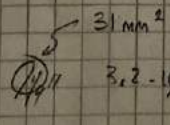








x 2



$$P_1 + \rho g h + \frac{1}{2}\rho v^2 = P_2 + \rho g h + \frac{1}{2}\rho v^2$$

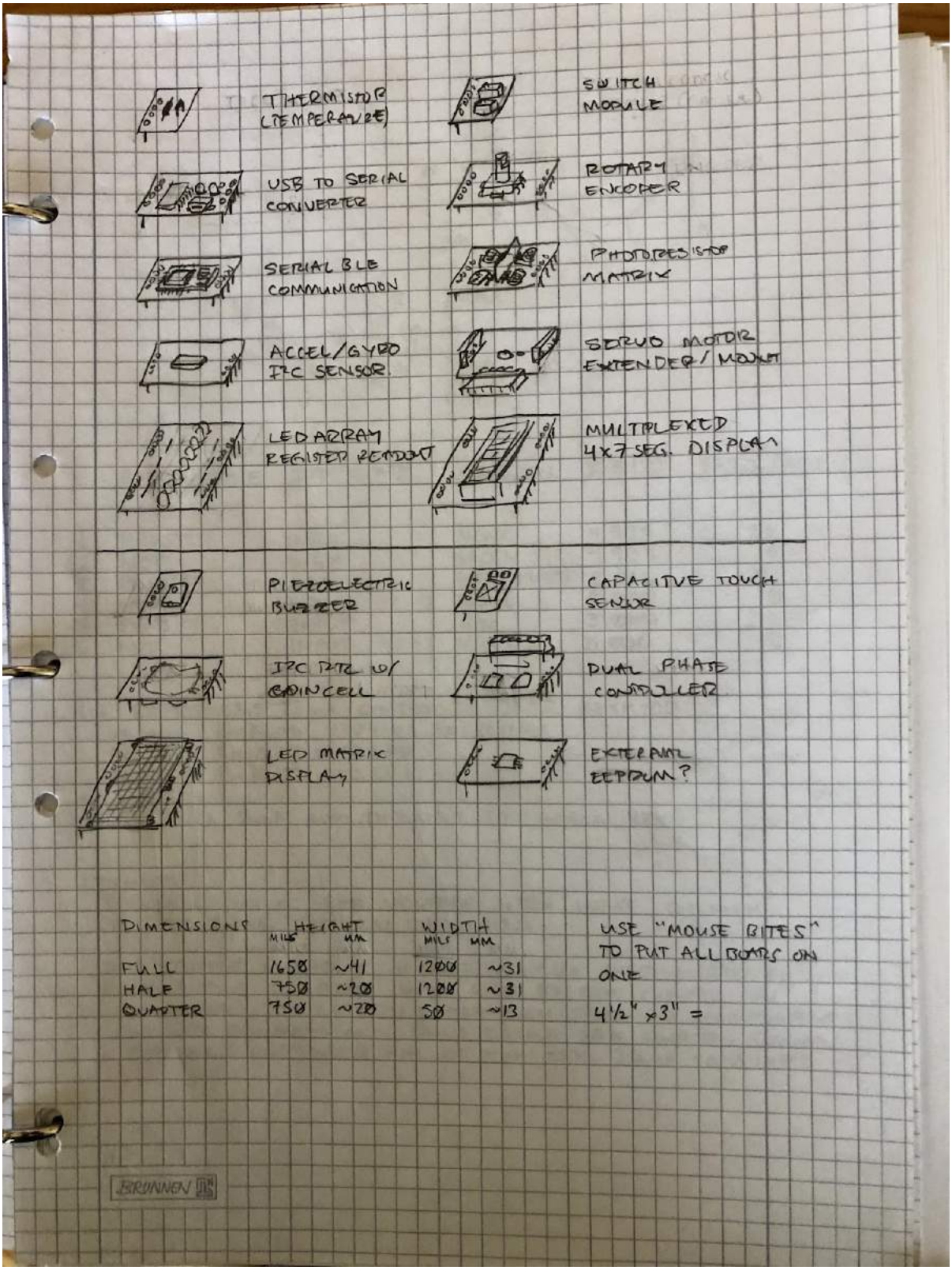
const

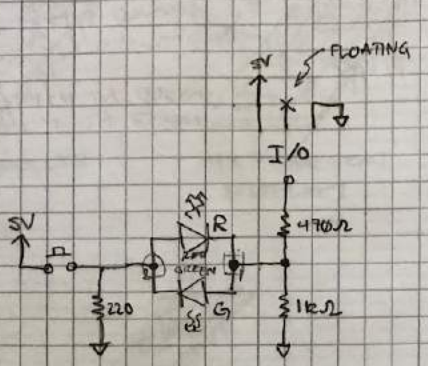
$$P_1 + \frac{1}{2}\rho v_1^2 = P_2 + \frac{1}{2}\rho v^2$$

$$\Delta P = \frac{1}{2}\rho (v_2^2 - v_1^2)$$

$$\sqrt{\frac{2\Delta P}{\rho}} = v$$

BRUNNEN


THERMISTOR (TEMPERATURE)
SWITCH MODULE
USB TO SERIAL CONVERTER
ROTARY ENCODER
SERIAL BLE COMMUNICATION
PHOTORESISTOR MATRIX
ACCEL/GYRO I²C SENSOR
SERVO MOTOR EXTENDER / MOUNT
LED ARRAY REGISTER REMOVED
MULTIPLEXED 4x7 SEG. DISPLAY
PIEZOELECTRIC BUZZER
CAPACITIVE TOUCH SENSOR
I²C RTC W/ COINCELL
DUAL PHASE CONTROLLER
LED MATRIX DISPLAY
EXTERNAL EEPROM?
DIMENSIONS
HEIGHT MILS MM
WIDTH MILS MM
FULL 1650 ~41 1200 ~31
HALF 750 ~20 1200 ~31
QUARTER 750 ~20 500 ~13
USE "MOUSE BITES" TO PUT ALL BOARDS ON ONE
4½" x 3" =
BRUNNON



	BUTTON STATE	I/O STATE	LEDS ON	I/O (V)	CKT I (mA)
1	PRESSED	HIGH	NONE	4.925	26
2	NOT	HIGH	GREEN	4.983	5
3	PRESSED	LOW	RED (BRIGHT)	0.016	32
4	NOT	LOW	NONE	0.000	0
5	PRESSED	INPUT	RED	3.040	26
6	NOT	INPUT	NONE	0.000	0

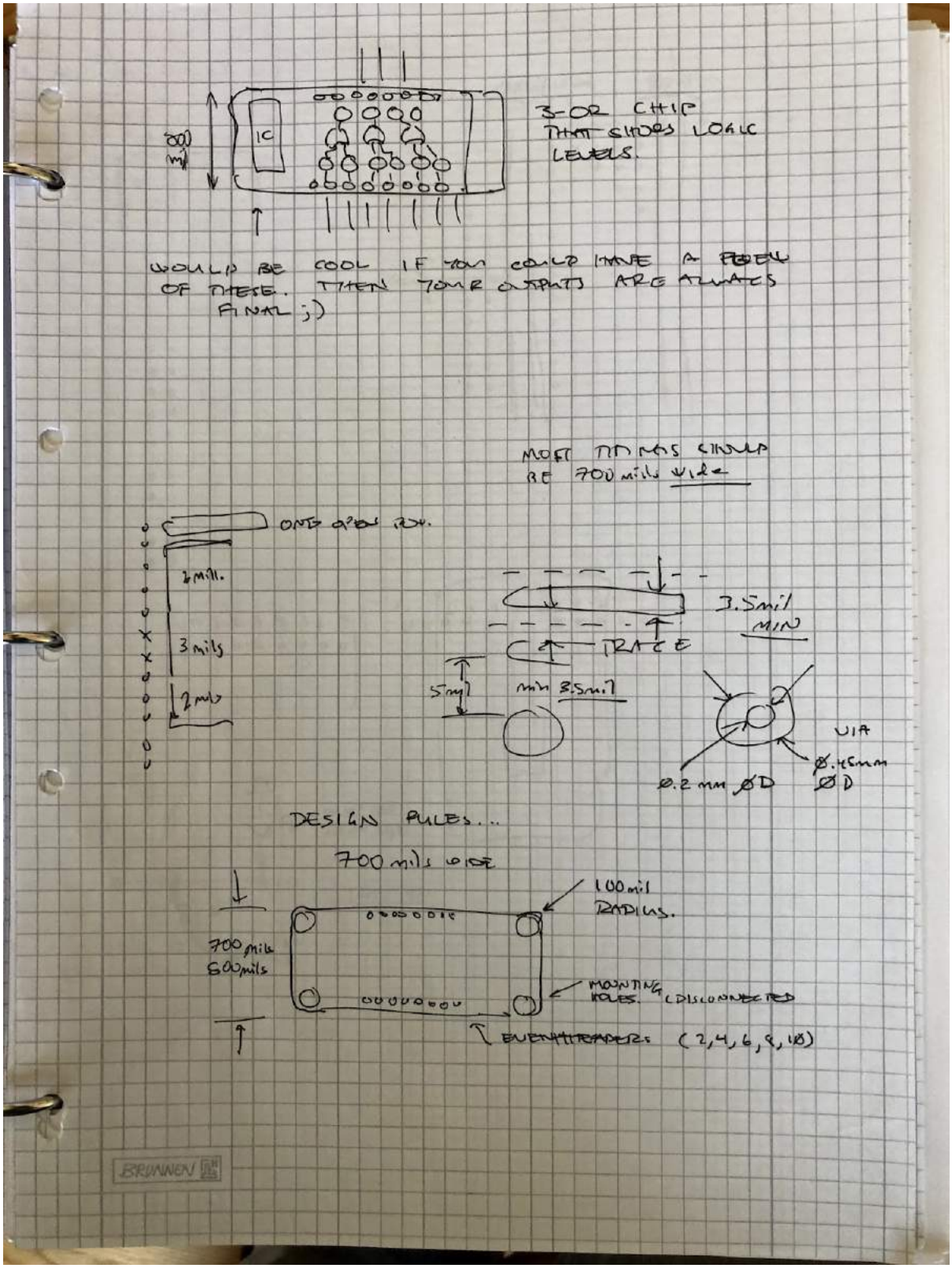
IN STATES 1,3,5 23mA GO THROUGH 220Ω RESISTOR
IN 1 .. 3mA TO 470+1kΩ RESISTORS
IN 5 .. 3mA TO RED LED
IN 3 .. 9mA TO RED LED

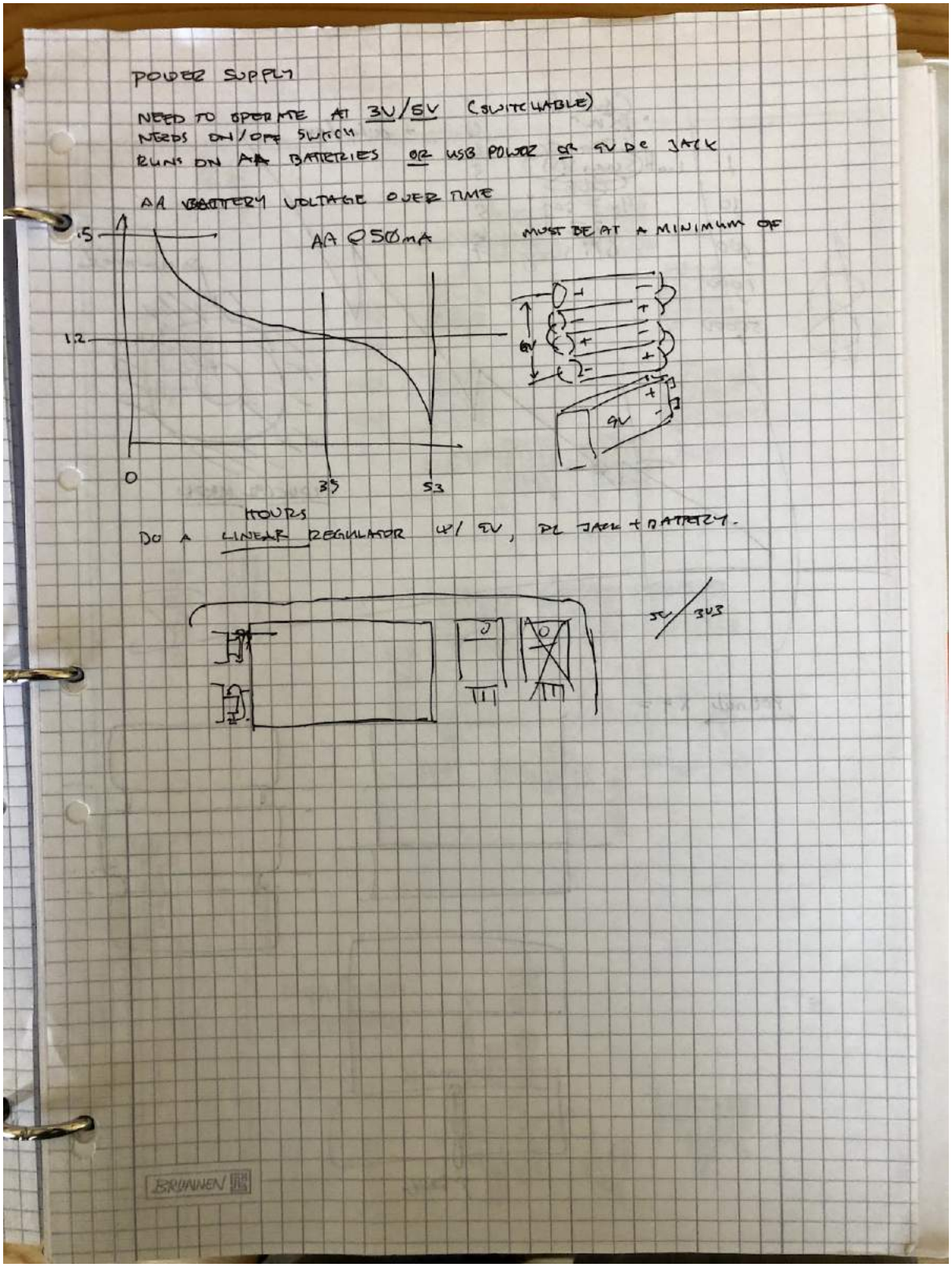
IN 2 .. ALL 5mA GOES THROUGH THE GREEN LED.

RESULTS:
THIS WILL WORK. THE VOLTAGE WHEN INPUT ON HIGH IS WELL ABOVE THE 2V THRESHOLD.

RED TRIPLE CURRENT IS NOT VISIBLY DISTINGUISHABLE FROM SINGLE CURRENT W/O SIDE BY SIDE COMPARISON. HOWEVER THE CURRENTS ARE SUCH THAT NO COMPONENTS WILL BE DAMAGED AND ONLY 33mA ARE SOURCED FROM THE MICROCONTROLLER PIN OF THE TOTAL 26mA CONSUMED.

BRUNNEN


IC
3-OR CHIP THAT SHOWS LOGIC LEVELS.
WOULD BE COOL IF YOU COULD HAVE A POWER OF THESE. THEN YOUR OUTPUTS ARE ALWAYS FINAL ;)
MOST TRACES SHOULD BE 700 mils WIDE
ONE OZ COPPER
2 mil.
3 mils
2 mils
3.5mil MIN
TRACE
5mil
min 3.5mil
VIA
0.2 mm ØD
ØD
DESIGN RULES...
700 mils WIDE
700 mils
600 mils
100 mil RADIUS.
MOUNTING HOLES. (DISCONNECTED)
EVEN NUMBERS: (2, 4, 6, 8, 10)


POWER SUPPLY
NEED TO OPERATE AT 3V/5V (SWITCHABLE)
NEEDS ON/OFF SWITCH
RUNS ON AA BATTERIES OR USB POWER OR 5V DC JACK
AA BATTERY VOLTAGE OVER TIME
AA @ 50mA
MUST BE AT A MINIMUM OF
1.5
1.2
0
3.5
5.3
HOURS
6V
9V
DO A LINEAR REGULATOR W/ 5V, DC JACK + BATTERY.
5V / 3V3

APPENDIX B: Design Files

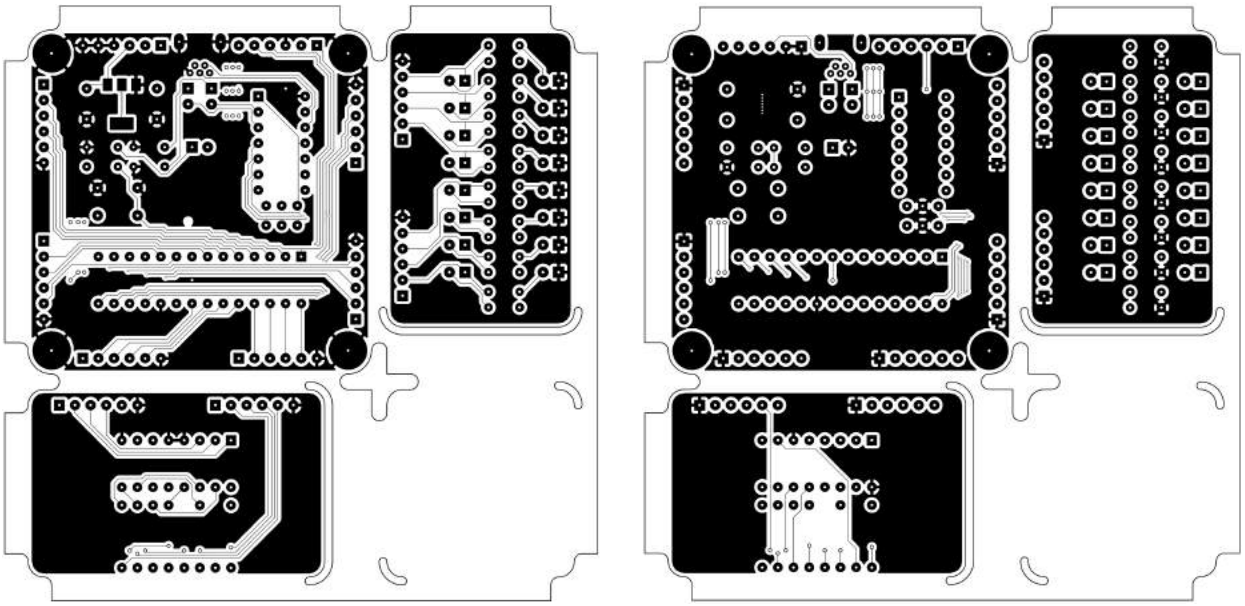


Figure 5. Panel A Front and Back
(Felix Controller, 4x7 Segment Display, LED Array)

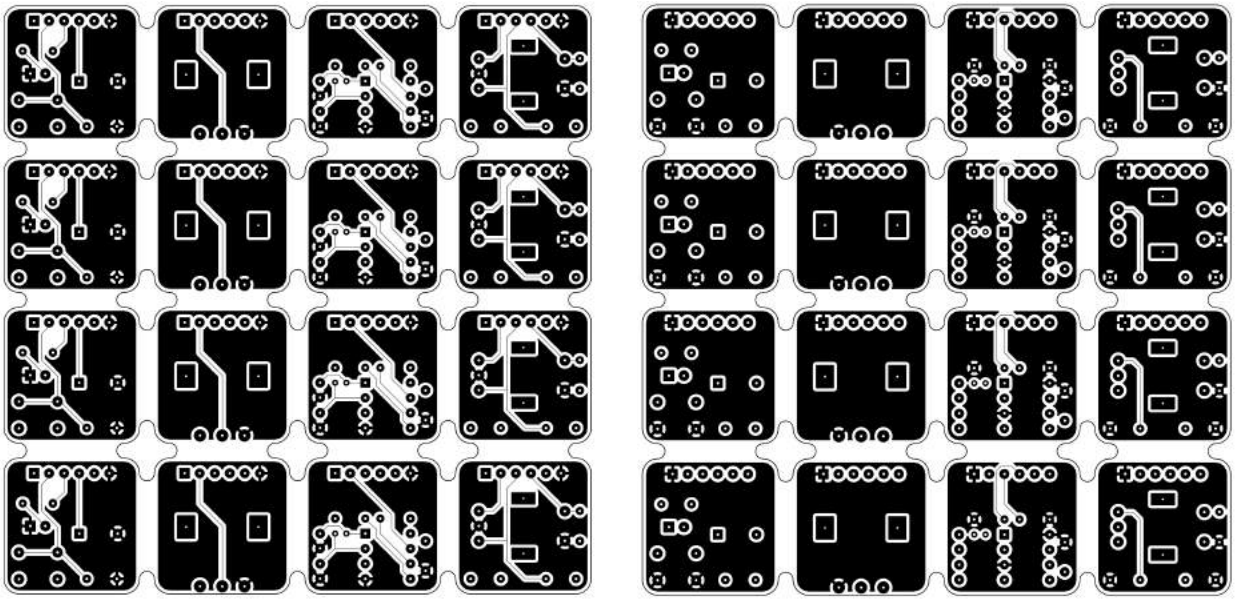
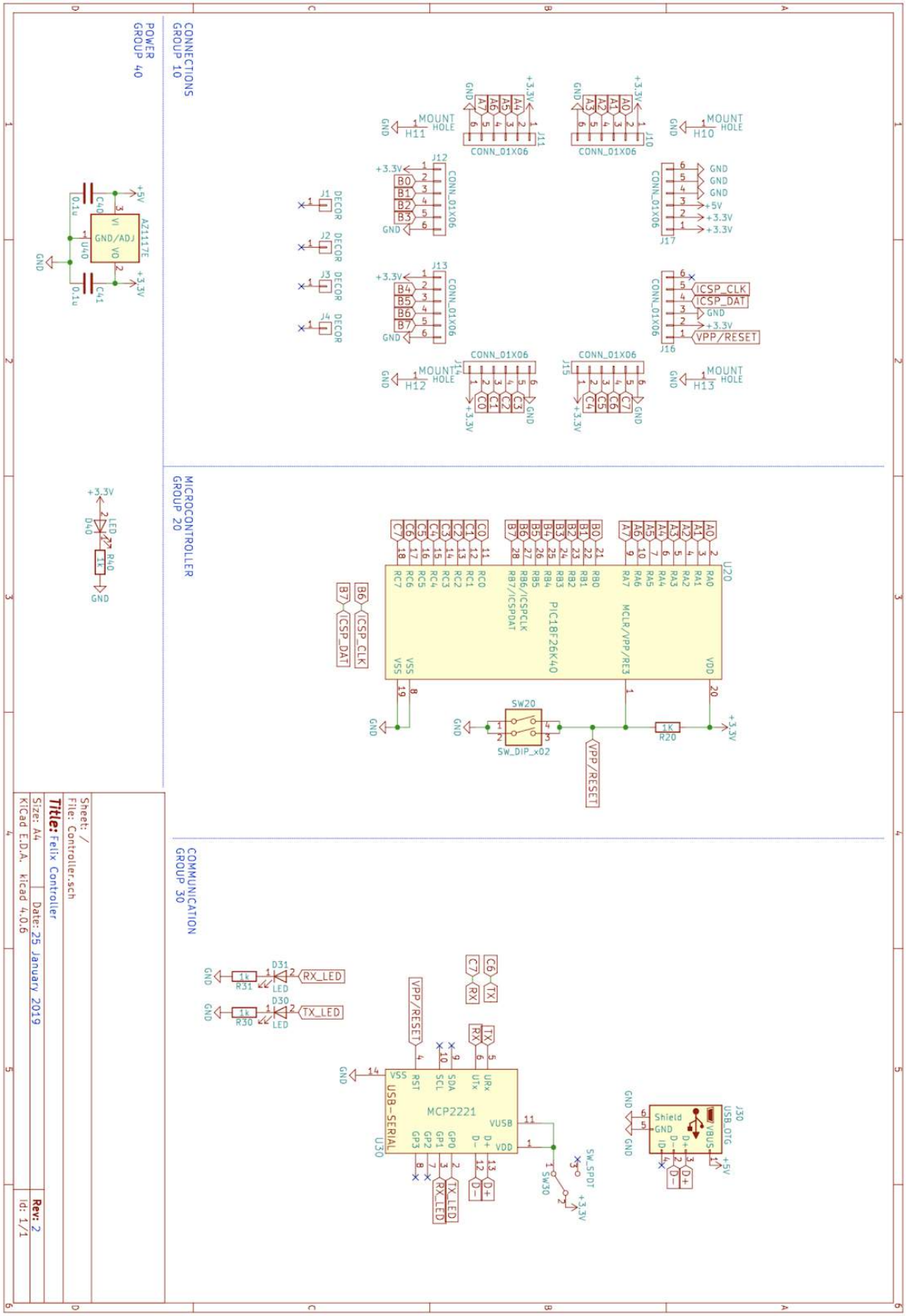


Figure 6. Panel B Front and Back
(GPIO Array, Potentiometer, Real Time Clock, Rotary Encoder)


POWER
GROUP 40
CONNECTIONS
GROUP 10
MICROCONTROLLER
GROUP 20
COMMUNICATION
GROUP 30
AZ1117E
GND/ADJ
PIC18F26K40
MCP2221
USB-SERIAL
SW_DIP_x02
SW_SPDT
CONN_01X06
MOUNT HOLE
ICSP_CLK
ICSP_DAT
VPP/RESET
RX_LED
TX_LED
Title: Felix Controller
Sheet: /
File: Controller.sch
Size: A4
Date: 25 January 2019
KiCad E.D.A. kicad 4.0.6
Rev: 2
Id: 1/1

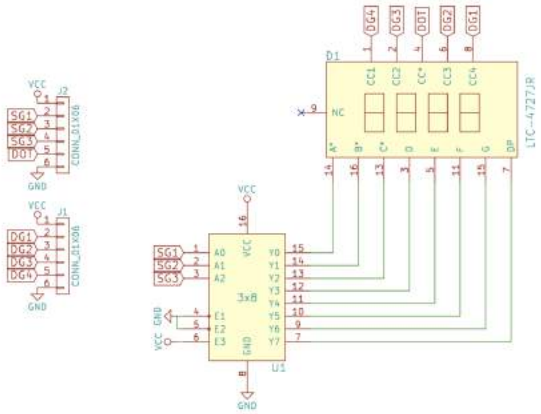


Figure 7. 4x7 Segment Display

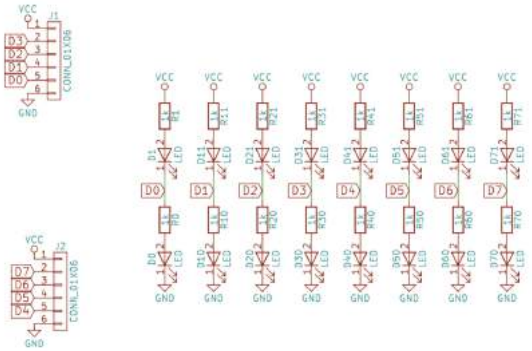


Figure 8. LED Array

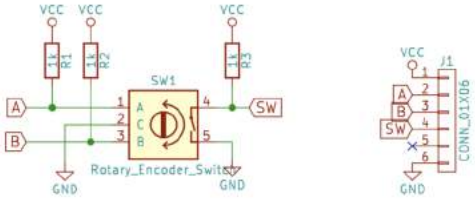


Figure 9. Rotary Encoder

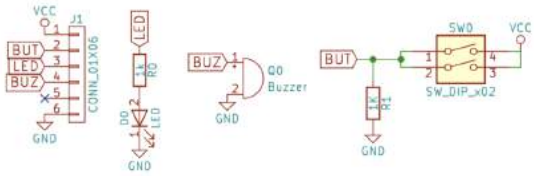


Figure 10. GPIO Array

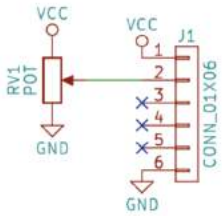


Figure 11. Potentiometer

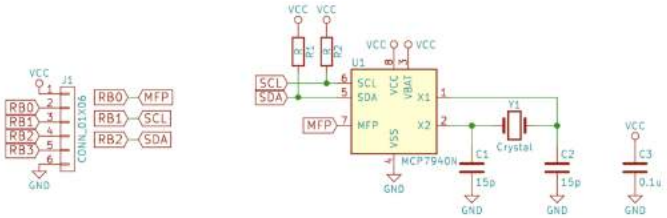


Figure 12. Real Time Clock

APPENDIX C: Datasheets

MICROCHIP

PIC18(L)F24/25K40

28-Pin, Low-Power, High-Performance Microcontrollers with XLP Technology

Description

These PIC18(L)F24/25K40 microcontrollers feature Analog, Core Independent Peripherals and Communication Peripherals, combined with eXtreme Low-Power (XLP) technology for a wide range of general purpose and low-power applications. These 28 -pin devices are equipped with a 10-bit ADC with Computation (ADCC) automating Capacitive Voltage Divider (CVD) techniques for advanced touch sensing, averaging, filtering, oversampling and performing automatic threshold comparisons. They also offer a set of Core Independent Peripherals such as Complementary Waveform Generator (CWG), Windowed Watchdog Timer (WWDT), Cyclic Redundancy Check (CRC)/Memory Scan, Zero-Cross Detect (ZCD) and Peripheral Pin Select (PPS), providing for increased design flexibility and lower system cost.

Core Features

- C Compiler Optimized RISC Architecture
- Operating Speed:
  - DC – 64 MHz clock input over the full V_{DD} range
  - 62.5 ns minimum instruction cycle
- Programmable 2-Level Interrupt Priority
- 31-Level Deep Hardware Stack
- Three 8-Bit Timers (TMR2/4/6) with Hardware Limit Timer (HLT)
- Four 16-Bit Timers (TMR0/1/3/5)
- Low-Current Power-on Reset (POR)
- Power-up Timer (PWRT)
- Brown-out Reset (BOR)
- Low-Power BOR (LPBOR) Option
- Windowed Watchdog Timer (WWDT):
  - Watchdog Reset on too long or too short interval between watchdog clear events
  - Variable prescaler selection
  - Variable window size selection
  - All sources configurable in hardware or software

Memory

- Up to 32K bytes Program Flash Memory
- Up to 2048 Bytes Data SRAM Memory

© 2017 Microchip Technology Inc.

- 256 Bytes Data EEPROM
- Programmable Code Protection
- Direct, Indirect and Relative Addressing modes

Operating Characteristics

- Operating Voltage Ranges:
  - 1.8V to 3.6V (PIC18LF24/25K40)
  - 2.3V to 5.5V (PIC18F24/25K40)
- Temperature Range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C

Power-Saving Operation Modes

- Doze: CPU and Peripherals Running at Different Cycle Rates (typically CPU is lower)
- Idle: CPU Halted While Peripherals Operate
- Sleep: Lowest Power Consumption
- Peripheral Module Disable (PMD):
  - Ability to selectively disable hardware module to minimize active power consumption of unused peripherals
- Extreme Low-Power mode (XLP)
  - Sleep: 500 nA typical @ 1.8V
  - Sleep and Watchdog Timer: 900 nA typical @ 1.8V

eXtreme Low-Power (XLP) Features

- Sleep mode: 50 nA @ 1.8V, typical
- Windowed Watchdog Timer: 500 nA @ 1.8V, typical
- Secondary Oscillator: 500 nA @ 32 kHz
- Operating Current:
  - 8 uA @ 32 kHz, 1.8V, typical
  - 32 uA/MHz @ 1.8V, typical

Digital Peripherals

- Complementary Waveform Generator (CWG):
  - Rising and falling edge dead-band control
  - Full-bridge, half-bridge, 1-channel drive
  - Multiple signal sources
- Capture/Compare/PWM (CCP) modules:
  - Two CCPs
  - 16-bit resolution for Capture/Compare modes
  - 10-bit resolution for PWM mode
- 10-Bit Pulse-Width Modulators (PWM):

  - Two 10-bit PWMs
- Serial Communications:
  - One Enhanced USART (EUSART) with Auto-Baud Detect, Auto-wake-up on Start. RS-232, RS-485, LIN compatible
  - SPI
  - I^2C, SMBus and PMBus™ compatible
- Up to 25 I/O Pins and One Input Pin:
  - Individually programmable pull-ups
  - Slew rate control
  - Interrupt-on-change on all pins
  - Input level selection control
- Programmable CRC with Memory Scan:
  - Reliable data/program memory monitoring for Fail-Safe operation (e.g., Class B)
  - Calculate CRC over any portion of Flash or EEPROM
  - High-speed or background operation
- Hardware Limit Timer (TMR2/4/6+HLT):
  - Hardware monitoring and Fault detection
- Peripheral Pin Select (PPS):
  - Enables pin mapping of digital I/O
- Data Signal Modulator (DSM)

Analog Peripherals

- 10-Bit Analog-to-Digital Converter with Computation (ADC^2):
  - 24 external channels
  - Conversion available during Sleep
  - Four internal analog channels
  - Internal and external trigger options
  - Automated math functions on input signals:
    - Averaging, filter calculations, oversampling and threshold comparison
  - 8-bit hardware acquisition timer
- Hardware Capacitive Voltage Divider (CVD) Support:
  - 8-bit precharge timer
  - Adjustable sample and hold capacitor array
  - Guard ring digital output drive
- Zero-Cross Detect (ZCD):
  - Detect when AC signal on pin crosses ground
- 5-Bit Digital-to-Analog Converter (DAC):
  - Output available externally
  - Programmable 5-bit voltage (% of V_{DD},[V_{Ref+} - V_{Ref-}], FVR)
  - Internal connections to Comparators and ADC
- Two Comparators (CMP):
  - Four external inputs
  - External output via PPS

- Fixed Voltage Reference (FVR) module:
  - 1.024V, 2.048V and 4.096V output levels
  - Two buffered outputs: One for DAC/CMP and one for ADC

Clocking Structure

- High-Precision Internal Oscillator Block (HFINTOSC):
  - Selectable frequencies up to 64 MHz
  - ±1% at calibration
- 32 kHz Low-Power Internal Oscillator (LFINTOSC)
- External 32 kHz Crystal Oscillator (SOSC)
- External High-frequency Oscillator Block:
  - Three crystal/resonator modes
  - Digital Clock Input mode
  - 4x PLL with external sources
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if external clock stops
- Oscillator Start-up Timer (OST)

Programming/Debug Features

- In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) with Three Breakpoints via Two Pins
- Debug Integrated On-Chip

PIC18(L)F24/25K40 Family Types

Table 1. Devices included in this data sheet

Device	Program Memory Flash (bytes)	Data SRAM (bytes)	Data EEPROM (bytes)	I/O Pins	16-bit Timers	Comparators	10-bit ADC^2 with Computation (ch)	5-bit DAC	Zero-Cross Detect	CCP/10-bit PWM	CWG	SMT	Low Voltage Detect (LVD)	8-bit TMR with HLT	Windowed Watchdog Timer	CRC with Memory Scan	EUSART	I^{2}C/SPI	PPS	Peripheral Module Disable	Temperature Indicator	Debug(1)
PIC18(L)F24K40	16k	1024	256	25	4	2	24	1	1	2/2	1	0	1	3	Y	Y	1	1	Y	Y	Y	I
PIC18(L)F25K40	32k	2048	256	25	4	2	24	1	1	2/2	1	0	1	3	Y	Y	1	1	Y	Y	Y	I

Table 2. Devices not included in this data sheet

Device	Program Memory Flash (bytes)	Data SRAM (bytes)	Data EEPROM (bytes)	I/O Pins	16-bit Timers	Comparators	10-bit ADC² with Computation (ch)	5-bit DAC	Zero-Cross Detect	CCP/10-bit PWM	CWG	SMT	Low Voltage Detect (LVD)	8-bit TMR with HLT	Windowed Watchdog Timer	CRC with Memory Scan	EUSART	I²C/SPI	PPS	Peripheral Module Disable	Temperature Indicator	Debug(1)
PIC18(L)F26K40	64k	3615	1024	25	4	2	24	1	1	2/2	1	0	1	3	Y	Y	2	2	Y	Y	Y	I
PIC18(L)F27K40	128k	3615	1024	25	4	2	24	1	1	2/2	1	0	1	3	Y	Y	2	2	Y	Y	Y	I
PIC18(L)F45K40	32k	2048	256	36	4	2	35	1	1	2/2	1	0	1	3	Y	Y	2	2	Y	Y	Y	I
PIC18(L)F46K40	64k	3615	1024	36	4	2	35	1	1	2/2	1	0	1	3	Y	Y	2	2	Y	Y	Y	I
PIC18(L)F47K40	128k	3615	1024	36	4	2	35	1	1	2/2	1	0	1	3	Y	Y	2	2	Y	Y	Y	I
PIC18(L)F65K40	32k	2048	1024	60	5	3	45	1	1	5/2	1	2	1	4	Y	Y	5	2	Y	Y	Y	I
PIC18(L)F66K40	64k	3562	1024	60	5	3	45	1	1	5/2	1	2	1	4	Y	Y	5	2	Y	Y	Y	I
PIC18(L)F67K40	128k	3562	1024	60	5	3	47	1	1	5/2	1	2	1	4	Y	Y	5	2	Y	Y	Y	I

Note: Debugging Methods: (I) – Integrated on Chip.

Data Sheet Index:

1. DS40001843 PIC18(L)F24/25K40 Data Sheet, 28-Pin, 8-bit Flash Microcontrollers
2. DS40001816 PIC18(L)F26/45/46K40 Data Sheet, 28/40/44-Pin, 8-bit Flash Microcontrollers
3. DS40001844 PIC18(L)F27/47K40 Data Sheet, 28/40/44-Pin, 8-bit Flash Microcontrollers
4. DS40001842 PIC18(L)F65/66K40 Data Sheet, 64-Pin, 8-bit Flash Microcontrollers
5. DS40001841 PIC18(L)F67K40 Data Sheet, 64-Pin, 8-bit Flash Microcontrollers

Pin Diagrams

Figure 1. 28-pin SPDIP, SSOP, SOIC

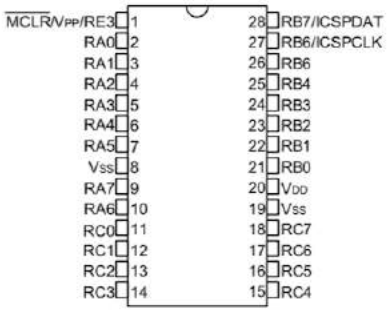
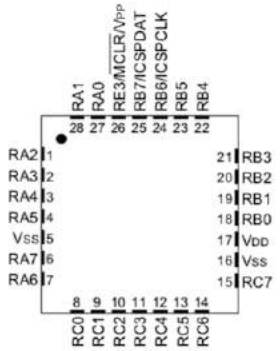


Figure 2. 28-pin QFN, UQFN



Note: It is recommended that the exposed bottom pad be connected to V_{SS}, however it must not be the only V_{SS} connection to the device.

Pin Allocation Tables

Table 1. 28-Pin Allocation Table

I/O(2)	28-Pin SPDIP, SOIC, SSOP	28-Pin (U)QFN	A/D	Reference	Comparator	Timers	CCP	CWG	ZCD	Interrupt	EUSART	DSM	MSSP	Pull-up	Basic
RA0	2	27	ANA0	—	C1IN0- C2IN0-	—	—	—	—	IOCA0	—	—	—	Y	—
RA1	3	28	ANA1	—	C1IN1- C2IN1-	—	—	—	—	IOCA1	—	—	—	Y	—
RA2	4	1	ANA2	DAC1OUT1 Vref- (DAC) Vref- (ADC)	C1IN0+ C2IN0+	—	—	—	—	IOCA2	—	—	—	Y	—
RA3	5	2	ANA3	Vref+ (DAC) Vref+ (ADC)	C1IN1+	—	—	—	—	IOCA3	—	MDCARL(1)	—	Y	—
RA4	6	3	ANA4	—	—	T0CKI(1)	—	—	—	IOCA4	—	MDCARH(1)	—	Y	—
RA5	7	4	ANA5	—	—	—	—	—	—	IOCA5	—	MDSRC(1)	$\overline{SS1}$(1)	Y	—
RA6	10	7	ANA6	—	—	—	—	—	—	IOCA6	—	—	—	Y	CLKOUT OSC2
RA7	9	6	ANA7	—	—	—	—	—	—	IOCA7	—	—	—	Y	OSC1 CLKIN
RB0	21	18	ANB0	—	C2IN1+	—	—	CWG1(1)	ZCDIN	IOCB0 INT0(1)	—	—	—	Y	—
RB1	22	19	ANB1	—	C1IN3- C2IN3-	—	—	—	—	IOCB1 INT1(1)	—	—	—	Y	—
RB2	23	20	ANB2	—	—	—	—	—	—	IOCB2 INT2(1)	—	—	—	Y	—

I/O(2)	28-Pin SPDIP, SOIC, SSOP	28-Pin (U)QFN	A/D	Reference	Comparator	Timers	CCP	CWG	ZCD	Interrupt	EUSART	DSM	MSSP	Pull-up	Basic
RB3	24	21	ANB3	—	C1IN2- C2IN2-	—	—	—	—	IOCB3	—	—	—	Y	—
RB4	25	22	ANB4	—	—	T5G(1)	—	—	—	IOCB4	—	—	—	Y	—
RB5	26	23	ANB5	—	—	T1G(1)	—	—	—	IOCB5	—	—	—	Y	—
RB6	27	24	ANB6	—	—	—	—	—	—	IOCB6	—	—	—	Y	ICSPCLK
RB7	28	25	ANB7	DAC1OUT2	—	T6IN(1)	—	—	—	IOCB7	—	—	—	Y	ICSPDAT
RC0	11	8	ANC0	—	—	T1CKI(1) T3CKI(1) T3G(1)	—	—	—	IOCC0	—	—	—	Y	SOSCO
RC1	12	9	ANC1	—	—	—	CCP2(1)	—	—	IOCC1	—	—	—	Y	SOSCIN SOSCI
RC2	13	10	ANC2	—	—	T5CKI(1)	CCP1(1)	—	—	IOCC2	—	—	—	Y	—
RC3	14	11	ANC3	—	—	T2IN(1)	—	—	—	IOCC3	—	—	SCK1(1) SCL1(3,4)	Y	—
RC4	15	12	ANC4	—	—	—	—	—	—	IOCC4	—	—	SDI1(1) SDA1(3,4)	Y	—
RC5	16	13	ANC5	—	—	T4IN(1)	—	—	—	IOCC5	—	—	—	Y	—
RC6	17	14	ANC6	—	—	—	—	—	—	IOCC6	CK1(1,3)	—	—	Y	—
RC7	18	15	ANC7	—	—	—	—	—	—	IOCC7	RX1/ DT1(1,3)	—	—	Y	—
RE3	1	26	—	—	—	—	—	—	—	IOCE3	—	—	—	Y	Vpp/MCLR
VSS	19	16	—	—	—	—	—	—	—	—	—	—	—	—	VSS
VDD	20	17	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	8	5	—	—	—	—	—	—	—	—	—	—	—	—	VSS
OUT(2)	—	—	ADGRDA ADGRDB	—	C1OUT C2OUT	TMR0	CCP1 CCP2 PWM3 PWM4	CWG1A CWG1B CWG1C CWG1D	—	—	TX1/ CK1(3) DT1(3)	DSM	SDO1 SCK1	—	—

Note:

1. This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to the peripheral input selection table for details on which PORT pins may be used for this signal.
2. All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in the peripheral output selection table.
3. This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
4. These pins are configured for I^2C logic levels; The SCLx/SDAx signals may be assigned to any of these pins. PPS assignments to the other pins (e.g., RB1) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

3.6 Register Summary - Configuration Words

Offset	Name	Bit Pos.								
0x300000	CONFIG1	7:0		RSTOSC[2:0]			FEXTOSC[2:0]			
		15:8			FCMEN		CSWEN			CLKOUTEN
0x300002	CONFIG2	7:0	BOREN[1:0]		LPBOREN				PWRTE	MCLRE
		15:8	XINST		DEBUG	STVREN	PPS1WAY	ZCD	BORV[1:0]	
0x300004	CONFIG3	7:0		WDTE[1:0]		WDTCPS[4:0]				
		15:8			WDTCCS[2:0]			WDTCWS[2:0]		
0x300006	CONFIG4	7:0					WRT3	WRT2	WRT1	WRT0
		15:8			LVP	SCANE		WRTD	WRTB	WRTC
0x300008	CONFIG5	7:0							CPD	CP
		15:8								
0x30000A	CONFIG6	7:0					EBTR3	EBTR2	EBTR1	EBTR0
		15:8							EBTRB	

3.7 Register Definitions: Configuration Words

Figure 10-1. Program and Data Memory Map

<table>
<tr><th rowspan="2">Address</th><th colspan="5">Device</th></tr>
<tr><th>PIC18(L)Fx4K40</th><th>PIC18(L)F25/45K40</th><th>PIC18(L)F65K40</th><th>PIC18(L)Fx6K40</th><th>PIC18(L)Fx7K40</th></tr>
<tr><td>Note 1</td><td colspan="5">Stack (31 Levels)</td></tr>
<tr><td>00 0000h</td><td colspan="5">Reset Vecor</td></tr>
<tr><td>...</td><td colspan="5">...</td></tr>
<tr><td>00 0008h</td><td colspan="5">Interrupt Vecor High</td></tr>
<tr><td>...</td><td colspan="5">...</td></tr>
<tr><td>00 0018h</td><td colspan="5">Interrupt Vecor Low</td></tr>
<tr><td>...</td><td colspan="5">...</td></tr>
<tr><td>00 001Ah
to
00 3FFFh</td><td>Program Flash Memory (8 KW)</td><td>Program Flash Memory (16 KW)</td><td>Program Flash Memory (16 KW)</td><td>Program Flash Memory (32 KW)</td><td rowspan="2">Program Flash Memory (64 KW)</td></tr>
<tr><td>00 4000h
to
00 7FFFh</td><td rowspan="4">Not Present(2)</td><td rowspan="3">Not Present(2)</td><td rowspan="3">Not Present(2)</td><td rowspan="2">Not Present(2)</td></tr>
<tr><td>00 8000h
to
00 FFFFh</td></tr>
<tr><td>01 0000h
to
01 FFFFh</td><td rowspan="2">Not Present(2)</td></tr>
<tr><td>02 0000h
to
1F FFFFh</td></tr>
<tr><td>20 0000h
to
20 000Fh</td><td colspan="5">User IDs (8 Words)(3)</td></tr>
<tr><td>20 0010h
to
2F FFFFh</td><td colspan="5">Reserved</td></tr>
<tr><td>30 0000h
to
30 000Bh</td><td colspan="5">Configuration Words (6 Words)(3)</td></tr>
<tr><td>30 000Ch
to
30 FFFFh</td><td colspan="5">Reserved</td></tr>
<tr><td>31 0000h
to
31 00FFh</td><td colspan="2">Data EEPROM (256 Bytes)</td><td colspan="3" rowspan="2">Data EEPROM (1024 Bytes)</td></tr>
<tr><td>31 0100h
to
31 01FFh</td><td colspan="2">Unimplemented</td></tr>
<tr><td>30 000Ch
to
30 FFFFh</td><td colspan="5">Reserved</td></tr>
<tr><td>3F FFFCh
to
3F FFFDh</td><td colspan="5">Revision ID (1 Word)(4)</td></tr>
<tr><td>3F FFFEh
to
3F FFFFh</td><td colspan="5">Device ID (1 Word)(4)</td></tr>
</table>

Note 1: The stack is a separate SRAM panel, apart from all user memory panels.
2: The addresses do not roll over. The region is read as '0'.
3: Not code-protected.
4: Device/Revision IDs are hard-coded in silicon.

Figure 10-2. Memory Map and Code Protection Control

Region	Address	Device				
		PIC18(L)Fx4K40	PIC18(L)F25/45K40	PIC18(L)F65K40	PIC18(L)Fx6K40	PIC18(L)Fx7K40
PFM	00 0000h to 00 07FFh	Boot Block 1 KW CP, WRTB, EBTRB	Boot Block 1 KW CP, WRTB, EBTRB	Boot Block 1 KW CP, WRTB, EBTRB	Boot Block 1 KW CP, WRTB, EBTRB	Boot Block 1 KW CP, WRTB, EBTRB
	00 0800h to 00 1FFFh	Block 0 3 KW CP, WRT0, EBTR0	Block 0 3 KW CP, WRT0, EBTR0	Block 0 3 KW CP, WRT0, EBTR0	Block 0 7 KW CP, WRT0, EBTR0	Block 0 7 KW CP, WRT0, EBTR0
	00 2000h to 00 3FFFh	Block 1 4 KW CP, WRT1, EBTR1	Block 1 4 KW CP, WRT1, EBTR1	Block 1 4 KW CP, WRT1, EBTR1		
	00 4000h to 00 5FFFh	Not Present	Block 2 4 KW CP, WRT2, EBTR2	Block 2 4 KW CP, WRT2, EBTR2	Block 1 8 KW CP, WRT1, EBTR1	Block 1 8 KW CP, WRT1, EBTR1
	00 6000h to 00 7FFFh		Block 3 4 KW CP, WRT3, EBTR3	Block 3 4 KW CP, WRT3, EBTR3		
	00 8000h to 00 BFFFh		Not Present	Not Present	Block 2 8 KW CP, WRT2, EBTR2	Block 2 8 KW CP, WRT2, EBTR2
	00 C000h to 00 FFFFh				Block 3 8 KW CP, WRT3, EBTR3	Block 3 8 KW CP, WRT3, EBTR3
	01 0000h to 01 3FFFh				Not Present	Block 4 8 KW CP, WRT4, EBTR4
	01 4000h to 01 7FFFh					Block 5 8 KW CP, WRT5, EBTR5
	01 8000h to 01 BFFFh					Block 6 8 KW CP, WRT6, EBTR6
	01 C000h to 01 FFFFh					Block 7 8 KW CP, WRT7, EBTR7
CONFIG	30 0000h to 30 000Bh	6 Words WRTC				
Data EEPROM	31 0000h to 31 00FFh	256 Words CPD, WRTD		1 KW CPD, WRTD		
	31 0100h to 31 01FFh	Unimplemented				

MICROCHIP

MCP7940M

Low-Cost I²C Real-Time Clock/Calendar with SRAM

Timekeeping Features

- Real-Time Clock/Calendar (RTCC):
  - Hours, Minutes, Seconds, Day of Week, Day, Month, Year
  - Leap year compensated to 2399
  - 12/24 hour modes
- Oscillator for 32.768 kHz Crystals:
  - Optimized for 6-9 pF crystals
- On-Chip Digital Trimming/Calibration:
  - ±1 PPM resolution
  - ±129 PPM range
- Dual Programmable Alarms
- Versatile Output Pin:
  - Clock output with selectable frequency
  - Alarm output
  - General purpose output

Low-Power Features

- Wide Voltage Range:
  - Operating voltage range of 1.8V to 5.5V
- Low Typical Timekeeping Current:
  - Operating from Vcc: 1.2 µA at 3.3V

User Memory

- 64-byte SRAM

Operating Ranges

- 2-Wire Serial Interface, I²C Compatible
  - I²C clock rate up to 400 kHz
- Temperature Range:
  - Industrial (I): -40°C to +85°C

Packages:

- 8-Lead SOIC, MSOP, TSSOP, PDIP and 2x3 TDFN

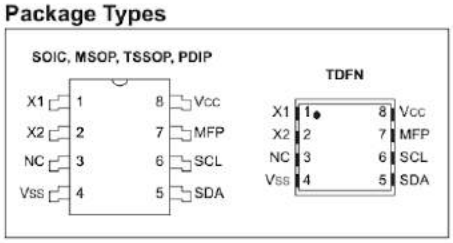
General Description

The MCP7940M Real-Time Clock/Calendar (RTCC) tracks time using internal counters for hours, minutes, seconds, days, months, years, and day of week. Alarms can be configured on all counters up to and including months. For usage and configuration, the MCP7940M supports I²C communications up to 400 kHz.

The open-drain, multi-functional output can be configured to assert on an alarm match, to output a selectable frequency square wave, or as a general purpose output.

The MCP7940M is designed to operate using a 32.768 kHz tuning fork crystal with external crystal load capacitors. On-chip digital trimming can be used to adjust for frequency variance caused by crystal tolerance and temperature.

Package Types



© 2012-2018 Microchip Technology Inc.

TABLE 4-1: DETAILED RTCC REGISTER MAP

Addr.	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Section 4.3 "Timekeeping"									
00h	RTCSEC	ST	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
01h	RTCMIN	—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
02h	RTCHOUR	—	12/24	AM/PM HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
03h	RTCWKDAY	—	—	OSCRUN	—	—	WKDAY2	WKDAY1	WKDAY0
04h	RTCDATE	—	—	DATETEN1	DATETEN0	DATEONE3	DATEONE2	DATEONE1	DATEONE0
05h	RTCMTH	—	—	LPYR	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
06h	RTCYEAR	YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
07h	CONTROL	OUT	SQWEN	ALM1EN	ALM0EN	EXTOSC	CRSTRIM	SQWFS1	SQWFS0
08h	OSCTRIM	SIGN	TRIMVAL6	TRIMVAL5	TRIMVAL4	TRIMVAL3	TRIMVAL2	TRIMVAL1	TRIMVAL0
09h	Reserved	Reserved – Do not use							
Section 4.4 "Alarms"									
0Ah	ALM0SEC	—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
0Bh	ALM0MIN	—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
0Ch	ALM0HOUR	—	12/24(2)	AM/PM HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
0Dh	ALM0WKDAY	ALMPOL	ALM0MSK2	ALM0MSK1	ALM0MSK0	ALM0IF	WKDAY2	WKDAY1	WKDAY0
0Eh	ALM0DATE	—	—	DATETEN1	DATETEN0	DATEONE3	DATEONE2	DATEONE1	DATEONE0
0Fh	ALM0MTH	—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
10h	Reserved	Reserved – Do not use							
Section 4.4 "Alarms"									
11h	ALM1SEC	—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
12h	ALM1MIN	—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
13h	ALM1HOUR	—	12/24(2)	AM/PM HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
14h	ALM1WKDAY	ALMPOL(3)	ALM1MSK2	ALM1MSK1	ALM1MSK0	ALM1IF	WKDAY2	WKDAY1	WKDAY0
15h	ALM1DATE	—	—	DATETEN1	DATETEN0	DATEONE3	DATEONE2	DATEONE1	DATEONE0
16h	ALM1MTH	—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
17h-1Fh	Reserved	Reserved – Do not use							

Note 1: Grey areas are unimplemented.

2: The 12/24 bits in the ALMxHOUR registers are read-only and reflect the value of the 12/24 bit in the RTCHOUR register.

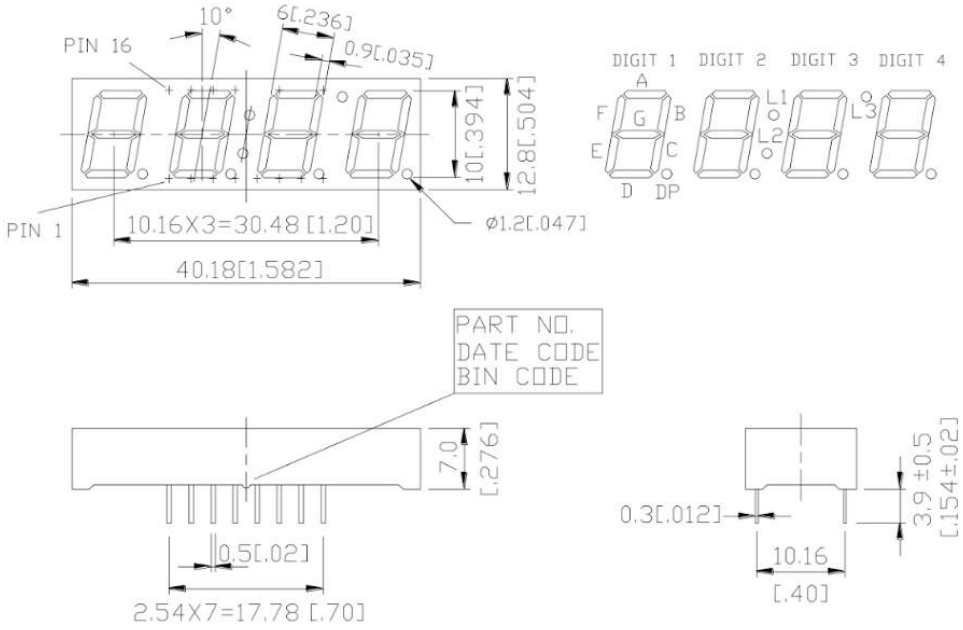
3: The ALMPOL bit in the ALM1WKDAY register is read-only and reflects the value of the ALMPOL bit in the ALM0WKDAY register.

© 2012-2018 Microchip Technology Inc.

LITE-ON TECHNOLOGY CORPORATION

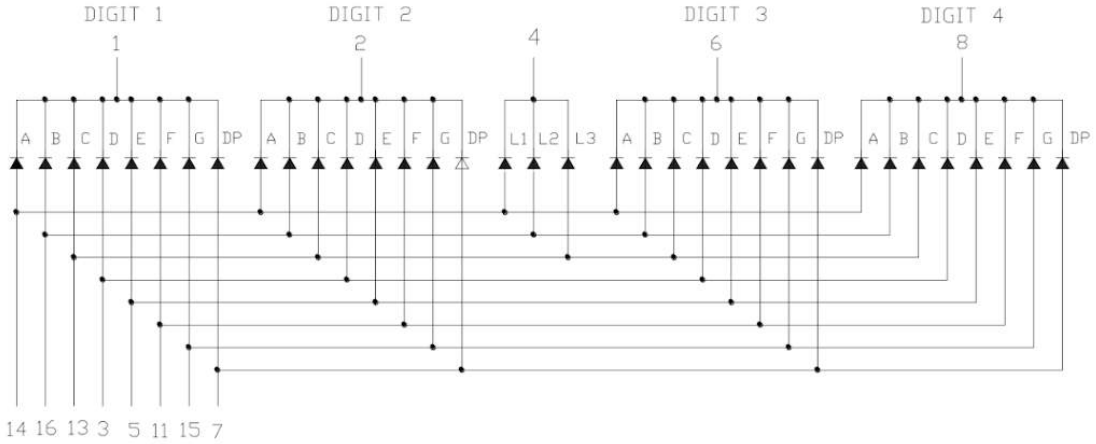
Property of Lite-On Only

PACKAGE DIMENSIONS



NOTES: All dimensions are in millimeters. Tolerances are± 0.25 mm (0.01“) unless otherwise noted.

INTERNAL CIRCUIT DIAGRAM

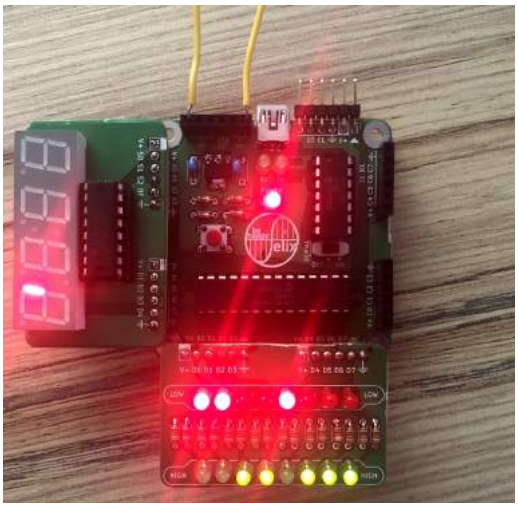


PART NO.: LTC-4727JR	PAGE: 2 of 5

APPENDIX D: Test Protocols

Four Digit Seven Segment Display
Date Tested: 28 November 2018
Tested By: Ryan Michael Norton

The LED array and 4x7 segment display were tested in unison. With the controller configured to have registers A and B as outputs, a script was developed to push specific values to both LATA and LATB, cycling through all the segments on the 4x7 segment display and showing the raw inputs to that display on the LED array.



main.c

```
#include "mcc_generated_files/mcc.h"
void main(void)
{
   // Initialize the device
   SYSTEM_Initialize();

   LATA = 0;
   LATB = 0;

   uint8_t port = 0;
   while (1) {
      for (uint8_t digit = 0; digit < 5; digit++) {
         for (uint8_t segment = 0; segment < 8; segment++) {
            port = 0x01;
            port = port << (digit + 3);

            port |= 0x07 & ~segment;
            port = ~port;

            LATA = port;
            LATB = port;
            __delay_ms(100);
```

```
      }
    }
  }
```

The code was written to cycle through each segment of the display, which can be controlled according to the following diagram. Pins 0-2 representing a 3 bit number that determines which segment is active on a given digit (A-G and dot). Pins 3-7 are directly connected to the ground pins of the five digits (4 numeric, one for extra dots) and are active when low. If a digit is not supposed to be displayed, the associated pin should be held high so no voltage drop is present across undesired LEDs.

Pin	Rx0	Rx1	Rx2	Rx3	Rx4	Rx5	Rx6	Rx7
Function	Decoder Input 0	Decoder Input 1	Decoder Input 2	Digit Dots	Digit 0	Digit 1	Digit 2	Digit 3
Active	High	High	High	Low	Low	Low	Low	Low

The delay in the code is so that individual LED segments can be observed lighting up to confirm individual control, and therefore multiplexing are possible. As the delay is lowered and then removed, all segments appear to be active, even if dimmer.

Conclusion: The LED array and 4x7 segment display function as desired. The green LEDs on the array are certainly dimmer than the red ones, and attention is drawn to the red. It should be considered to use only red LEDs for this specific module, but subject testing will be required to be sure.

EUSART to USB Bridge Test
Date Tested: 7 November 2018
Tested By: Ryan Michael Norton

After configuring the serial port using the MCC tool in MPLAB X, a script was written to display the classic “Hello, World!” repeatedly on the serial port. The code was flashed and then the USB port connected to a laptop. Ther serial port at the designated baud rate was opened in the Arduino IDE for simplicity and “Hello, World!” was repeatedly displayed on the screen.

main.c

```
#include "mcc_generated_files/mcc.h"
void main(void)
{
    // Initialize the device
    SYSTEM_Initialize();

    while (1)
```

```
  {
    // Add your application code
    char str[] = "Hello, World!\n";
    for (int i=0; i<14; i++) {
      EUSART_Write(str[i]);

    }
  }
}
```

Conclusion: MCP2221 serial bridge and PIC18F24K40 hardware serial port both function without issue.

LED Current Consumption Test
Date Tested: 18 November 2018
Tested By: Ryan Michael Norton

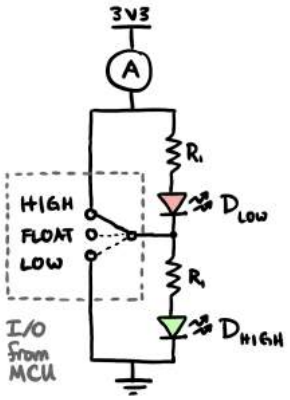
4x7 Segment Display
Multimeter: INNOVA 3320 Auto-Ranging Digital Multimeter
Display: LTC-4727JR (Part No. 160-1551-5-ND)
Decoder: CD74HC238E (Part No. 296-25983-5-ND)

Current was tested by connecting a single LED segment to a 3.3 V regulated power supply in line with an ammeter, decoder, and listed resistance. Visibility was noted by multiplexing through all possible segments using the prototype Felix controller version 1.0.

Nominal Resistance [ohm]	Measured Resistance [ohm]	Current Consumption [mA]	Visibility at Full Multiplexing
1k	986.0	1.56	Hardly visible
220	119.2	5.53	Somewhat visible
0 (short)	0 (short)	16.01	Comfortably visible

Conclusion: By taking advantage of the decoders output current limiting feature, no resistors on display produces the best results and is below the displays continuous forward current of 25 mA (see datasheet for LTC-4727JR in appendix).

LED Array Module
The following circuit was constructed and connected in the states listed in the table below. It represents a single unit of the LED array. Current through the ammeter (shown in schematic) was measured and visibility of each LED noted.



Input Pin State (TTL tristate)	Nominal Resistance [ohm]	Measured Resistance [ohm]	Current Consumption [mA]	Red Visibility	Green Visibility
High (~3.3 V)	1k	986	1.64	On	Off
	220	119.2	7.08	On	Off
Low (0 V)	1k	986	1.51	Off	On (Dim)
	220	119.2	6.27	Off	On
Not Connected (Floating)	1k	986	0.03	Barely On	Off
	220	119.2	0.06	Barely On	Off

Conclusion: While 220 ohm resistance produces a better user experience result, it comes at the cost of higher current consumption and having to source resistors of multiple values when everything else relies on 1k only. Even though the 220 ohm resistor uses more current, its still below the maximum forward current rating for both LEDs at 30 mA for both. Based on this, it is nominal to use 1k ohm resistors for the array.

Potentiometer Test
Date Tested: 7 November 2018
Tested By: Ryan Michael Norton

Code was written to connect the potentiometer pin to an ADC channel and export the received value to the serial port. After connecting the USB port to a laptop and opening the serial port using the Arduino IDE, turning the knob resulted in values ranging from 0 to 255.

main.c

```
#include "mcc_generated_files/mcc.h"
#include <math.h>
void main(void)
{
   SYSTEM_Initialize();

   ADPCH = 0b000000;
   ADCON0bits.ADFM = 0;
   ADCON0bits.ADON = 1;

   while (1) {
      ADCON0bits.ADGO = 1;
     while (ADCON0bits.ADGO == 1);
     PORTB = ADRESH;
    __delay_us(600);

    uint8_t result = ADRESH;
    char newline = '\r';
    uint8_t dig[] = {0, 0, 0};
    dig[1] = result % 10;
    result = (result - dig[1]) / 10;

    dig[2] = result % 10;
    result = (result - dig[2]) / 10;

    dig[3] = result;

    dig[1] += 48;
    dig[2] += 48;
    dig[3] += 48;

    EUSART_Write(dig[3]);
    __delay_ms(10);
    EUSART_Write(dig[2]);
    __delay_ms(10);
    EUSART_Write(dig[1]);
    __delay_ms(10);
    EUSART_Write(newline); // new line
    __delay_ms(10);
   }
}
```

Conclusion: Potentiometer works as designed.

Real Time Clock Test
Date Tested: 28 November 2018
Tested By: Ryan Michael Norton

Using the Arduino code generated by Kevin Darrah, code was ported to C in order to communicate with the real time clock module. MCC generated code was used to facilitate the

I2C protocol in hardware. Specialized functions were written to export data to the serial port in certain formats (strings, hexadecimal, etc).

A test time was loaded into the clock module at start and the real time clock was occasionally polled to determine the time. The time was then formatted and displayed on the serial port. An timer was set for 24 hours on a cell phone and the reset button hit just as the timer was started. When the timer went off the times were compared and not a second had been missed.

main.c

```
#include "mcc_generated_files/mcc.h"

#define RTCSEC     0x00
#define RTCMIN     0x01
#define RTCHOUR    0x02
#define RTCWKDAY   0x03
#define RTCDATE    0x04
#define RTCMTH     0x05
#define RTCYEAR    0x06
#define CONTROL    0x07
#define OSCTRIM    0x08
#define ALM0SEC    0x0A
#define ALM0MIN    0x0B
#define ALM0HOUR   0x0C
#define ALM0WKDAY  0x0D
#define ALM0DATE   0x0E
#define ALM0MTH    0x0F
#define ALM1SEC    0x11
#define ALM1MIN    0x12
#define ALM1HOUR   0x13
#define ALM1WKDAY  0x14
#define ALM1DATE   0x15
#define ALM1MTH    0x16
#define PWRDNMIN   0x18
#define PWRDNHOUR  0x19
#define PWRDNDATE  0x1A
#define PWRDNMTH   0x1B
#define PWRUPMIN   0x1C
#define PWRUPHOUR  0x1D
#define PWRUPDATE  0x1E
#define PWRUPMTH   0x1F

#define RTC        0xDE >> 1

char hex[] = "0123456789ABCDEF";
void print(uint8_t data) {
  EUSART_Write(data);
}
void printh (uint8_t data) { // print hexadecimal
  print('0');
  print('x');
  print(hex[data >> 4]);
  print(hex[data & 0x0F]);
}
void printn (uint8_t data) {
  data += 48;
  print(data);
}
void printl (char *str[]) {
```

```
   for (uint8_t index = 0; index < sizeof(str); index++) {
      print(str[index]);
   }
}
void println (char *str[]) {
   printl(str);
   print('\n');
}

I2C1_MESSAGE_STATUS pflag;
uint8_t buffer[2];

void write(uint8_t addr, uint8_t reg, uint8_t val) {
   buffer[0] = reg;
   buffer[1] = val;

   I2C1_MasterWrite (&buffer, 2, addr, &pflag);
   while(pflag == I2C1_MESSAGE_PENDING);
}

uint8_t read (uint8_t addr, uint8_t reg, uint8_t length) {
   buffer[0] = reg;
   buffer[1] = 0x00;

   I2C1_MasterWrite (&buffer, 1, addr, &pflag);
   while(pflag == I2C1_MESSAGE_PENDING);

   I2C1_MasterRead (&buffer, 1, addr, &pflag);
   while(pflag == I2C1_MESSAGE_PENDING);

   return buffer[0];
}

uint8_t seconds;
uint8_t minutes;
uint8_t hours;
uint8_t weekday;
uint8_t day;
uint8_t month;
uint8_t year;
uint8_t pm;

void getTime () {
   seconds = read(RTC,RTCSEC,  1);
   minutes = read(RTC,RTCMIN,  1);
   hours   = read(RTC,RTCHOUR, 1);
   weekday = read(RTC,RTCWKDAY,1);
   day     = read(RTC,RTCDATE, 1);
   month   = read(RTC,RTCMTH,  1);
   year    = read(RTC,RTCYEAR, 1);
}

char dow[7] = "UMTWRFS";
void printTime () {
   seconds &= 0x7F;
   minutes &= 0x7F;
   hours   &= 0x7F;
   weekday &= 0x07;
   day     &= 0x1F;
   month   &= 0x1F;

   printn((hours >> 4) & 0x01);
   printn(hours & 0x0F);
   print(':');
   printn(minutes >> 4);
```

```
   printn(minutes & 0x0F);
   print(':');
   printn(seconds >> 4);
   printn(seconds & 0x0F);
   print(' ');
   if (((hours >> 5) & 0x01) == 1) {
      print('P');
   } else {
      print('A');
   }
   print('M');
   print(' ');
   print(dow[(weekday & 0x07) -1]);
   print(' ');
   printn((day >> 4) & 0xFF);
   printn(day & 0x0F);
   print('/');
   printn(month>>4 & 0x01);
   printn(month & 0x0F);
   print('/');
   printn((year >> 4) & 0xFF);
   printn(year & 0x0F);
   print('\n');
}

uint8_t bin2bcd (uint8_t binary) {
   uint8_t low = 0;
   uint8_t high = 0;

   if (binary > 99) {
      binary = 99;
   }

   low = binary % 8;
   binary -= low;
   high = binary / 10;

   return (high << 4) | low;
}

uint8_t value;
void rtc_init() {
   write(RTC,CONTROL,0x00);

   value = read(RTC,RTCHOUR,1) | 0x40;
   write(RTC,RTCHOUR,value);

   value = read(RTC,RTCWKDAY,1) | 0x08;
   write(RTC,RTCWKDAY,value);

   value = read(RTC,RTCSEC,1) | 0x80;
   write(RTC,RTCSEC,value);
}

void set_time () {
   value = read(RTC,RTCSEC,1) | ((bin2bcd(seconds) & 0x7F));
   printh(value);
   write(RTC,RTCSEC,value);

   value = read(RTC,RTCMIN,1) | ((bin2bcd(minutes) & 0x7F));
   printh(value);
   write(RTC,RTCMIN,value);

   hours = bin2bcd(hours) | pm << 5;
   value = read(RTC,RTCHOUR,1) | (hours & 0x3F);
```

```
   printh(value);
   write(RTC,RTCHOUR,value);
}

void main(void)
{
   SYSTEM_Initialize();
   INTERRUPT_GlobalInterruptEnable();
   INTERRUPT_PeripheralInterruptEnable();

   __delay_ms(5000);
   for (int i = 0; i<20; i++) {
      print(':');
      __delay_ms(100);
   }

   seconds = 00;
   minutes = 12;
   hours   = 04;
   pm      =  1;


   set_time();


   rtc_init();

   uint8_t n = 0;

   while (1) {
      getTime();
      printTime();
   }

   while (1) {

      if (n % 8 == 0) {
         if (n == 0) {
            print('\n');
            getTime();
            printTime();

            print('\n');
             __delay_ms(1000);
         }
         print('\n');
         printh(n);
         print('\t');
         print('-');
         print('-');
         print('>');
      }
      print('\t');
      printh(read(RTC,n,1));

      n++;
      if (n > 0x1F) n = 0x00;
   }
}
```

Conclusion: As the real time clock was able to stably keep time over a 24 hour period, it is reasonable to conclude that its other capabilities function as desired and that the module is stable enough for educational use.

Rotary Encoder Test
Date Tested: 7 November 2018
Tested By: Ryan Michael Norton

Code was written to connect the rotary encoder pins to external interrupts and increment a register value displayed by the LED array on Port B. After uploading the code, the device was successfully able to increment and decrement the register value using the rotary capability.

main.c

```
#include "mcc_generated_files/mcc.h"
#include <math.h>

char value = 128;

void rotA (void) {
   if (PORTAbits.RA5) {
      value++;
   } else {
      value--;
   }
}
void main(void) {
   SYSTEM_Initialize();
   INT0_AttachInterrupt(rotA);
   while (1) {
      LATB = value;
   }
}
```

Conclusion: Rotary encoder works as designed.

SMD LDO Soldering Test
Date Tested: 9 November 2018
Tested By: Ryan Michael Norton

The subject was given a rudimentary soldering iron, solder, an empty controller board, and the SMD LDO. The subject was instructed to solder the regulator to the board. The subject was recorded being given instructions and soldering the regulator on. Experimenter was present and answered questions the subject had during the process. After the subject was comfortable with the job, the regulator was connected to a 5V power supply and the voltage measured.

Test Subject: Laura Stegner
Limited previous soldering experience. No surface mounted soldering experience.

See *SMD LDO Soldering Test - Soldering.mov* and *SMD LDO Soldering Test - Validation.mov* for more details.

Conclusion: SMD LDO is large enough to be soldering by amateurs successfully.