H&M Configuration Dongle

EE464 – Senior Design I

Senior Design Progress Report

Mohamed Ayoub
Harsh Dubey

12/17/2019

Department of Electrical Engineering and Computer Science
South Dakota State University

Project Approval

Project Governance		
Role *(Add/delete as applicable)*	**Name(s)** *(First initial. Last name)*	**Approval Signature / Date**[1]
Customer	*M. Sickler*	Matt Sickler 10/23/2019 ★
Sr. Engineer 1	*R. Fourney*	10/23/19
Project Manager	*S. Tan*	
Project Team		
Name(s) *(Last name, First name, Middle initial.)*		**Approval Signature / Date**
Ayoub, Mohamed, M.		Mohamed, 10/23/19
Dubey, Harsh		Harsh Dubey 10/23/19

★ **See electronic signature in Appendix A.**

Table of Contents

PROJECT APPROVAL **2**

TABLE OF CONTENTS **3**

PROPOSAL SUMMARY **4**

INTRODUCTION **5**

DEFINITION OF PROBLEM 5
BACKGROUND 5
SCOPE OF PROPOSAL 5
In Scope *6*
Out of Scope *6*

FUNCTIONAL DESIGN REQUIREMENTS **6**

OBJECTIVES 6
SPECIFICATIONS 6
DESIGN CONSTRAINTS 7

PROPOSED TECHNICAL SOLUTION **7**

OVERVIEW 7
SUBMODULE DESCRIPTION 7
Microcontroller *7*
LCD Screen *8*
Push Buttons *8*
USB Port & Power *8*
Communication Protocol with the Host Device *8*
ANTICIPATED REQUIRED RESOURCES 9
System level component list and cost estimate *9*
Test and Laboratory Equipment *9*
PROPOSED PERFORMANCE VERIFICATION 9

PROPOSED PROJECT SCHEDULE **13**

PLAN OF ACTION 13

REFERENCES **15**

Proposal Summary

Title: *Daktronics Configuration Dongle*

Team Members: *Mohamed Ayoub*

Harsh Dubey

Advisor: *Dr. Robert Fourney*

Daktronics manufactures display controllers which are accessed over a network. If the network fails or the network settings are incorrectly configured, the remote display controller becomes inaccessible. The primary objective of the project is to develop and build a device (Configuration Dongle) that will communicate with Daktronics's host device using a Universal Serial Bus (USB) to read and to configure the host device's network settings. This would require designing the dongle, with an appropriate user interface and a communication protocol that the dongle can use to communicate with the display controller.

The Configuration Dongle would then be used to read the current system settings of the display controller and then display them on the Configuration Dongle's LCD/OLED screen. Using push buttons installed on the Configuration Dongle, the user can navigate through the settings and reconfigure the display controller network setting.

The team will collaborate with each other to design the communication protocol that the Configuration Dongle will use to communicate with the display controller. After the protocol is designed, the team will design the actual Configuration Dongle. The Configuration Dongle components will include a microcontroller/microprocessor, an LCD/OLED screen, and five push buttons. After building and developing the Configuration Dongle, the team will place the Configuration Dongle into a housing ad showcase it at South Dakota State's Engineering EXPO.

Introduction

Definition of Problem

Daktronics is a company based in the U.S. that designs, manufactures, sells, and services video displays, scoreboards, digital billboards, and related products. The display controllers that Daktronics produces are Linux based machines that are normally accessed over a network using a web browser to configure and control the video displays. If the network fails or the display controller's network settings are corrupted, the display controller becomes inaccessible. This prevents the user from accessing the display controller to implement necessary changes to the video displays. Currently older models of the display controllers have keyboard and display support, which allows technicians to change the network configuration of the display controller through directly through the terminal. Newer models of the display controllers, however, do not have keyboard and display support and are shipped back to Daktronics. Daktronics then either recovers those units using complex methods or marks the units as "broken" units and disposes them. This wastes time and money for Daktronics. A device is needed to gain physical access to the network settings of those controllers (host devices) and to reconfigure these network settings as needed.

Background

One possible solution makes use of a Wi-Fi dongle, having the display controller connect to the ad hoc network generated by the Wi-Fi dongle, [5]. However, this solution is of no use if the network settings are corrupted. In this case it becomes difficult to access the display controller through any kind of network, [4].

A second possible solution makes use of a Bluetooth dongle to communicate with the display controller [1]. In this case, the technician would still have to change the configuration settings which is a complex procedure according to Daktronics. The technician could also use a smartphone and connect to the display controller using a USB cable and then make use of specialized application software in the smartphone to change the network settings, [2]. However, Daktronics company policies preclude the use of personal smartphones to change the settings of company products.

Therefore, developing a Configuration Dongle would be the preferred solution. The dongle should be cheap, compact, easy to carry, powered by the host device, and does not need a network connection to communicate with the host device.

Scope of Proposal

The remainder of the proposal is divided into three major sections. The first section defines the design project by clearly stating a set of objectives, specifications, and design constrains. This section is followed by the technical solution section which first presents an overview of the design and associated submodules followed by a detailed description of each submodule and how it achieves the objectives of the design project. The third subsection of the technical solution section presents a detailed list of the anticipated resources needed and their anticipated cost. Finally, the last section presents the design project schedule and the plan of action of the design project.

The scope of this project focuses on the initial development of a system, called the Configuration Dongle, to display and configure the network settings of the host device. The scope of the project is fully defined by separating tasks into "in scope" and "out of scope" categories which are listed below.

<u>In Scope</u>

Upon completion, the following items will be implemented in the design project:

- The Configuration Dongle will communicate with the host device by enumerating a standard USB.
- The Configuration Dongle will read the system name and type of the host device.
- The Configuration Dongle will be able to change the network mode of the host device.
- The Configuration Dongle will configure the network settings of the host device.

<u>Out of Scope</u>

These are the concerns that are related to the project that the team is not responsible for, and are thus out of the scope of the project but will be considered as optional objectives:

- ProLink Master control.
- Read link-local Internet Protocol (IP) v6.
- Device component identification.
- Develop a firmware for the host device.

Functional Design Requirements

Objectives

The primary objective of the project is to develop and build a device (Configuration Dongle) that will communicate with Daktronics's host device using USB to read and to configure the host device's network settings. Another objective of the project is to design the communication protocol of the Configuration Dongle so that the beginning and the end of a message can be detected unambiguously. Additionally, the device must be able to detect dropped or corrupted data and recover that data. Finally, the communication protocol of the device should be designed so it is backward and forward compatible and should have the Capability Detection feature.

Specifications

The specifications for the project are listed below:

1. The Configuration Dongle must include at least a 16x2 LCD/OLED display or a matrix OLED display to display the system information and current network settings of the host device.
2. The Configuration Dongle must include at least 3 push buttons on the right-hand side of the PCB.
3. The Configuration Dongle must include one USB port.
4. The Configuration Dongle must be a minimum of 1.5" by 2.5" in size but not larger than 3" by 6".
5. The Configuration Dongle must be able to read and write the IPv4 address of the host device under static mode.
6. The Configuration Dongle must be able to read and write IPv4 Netmask in Classless Inter-Domain Routing (CIDR) format.
7. The communication protocol between the Configuration Dongle and the host device must use at least 16-bit Cyclic Redundancy Check (CRC).

Design Constraints
The design constraints of the project are listed below:

- The Configuration Dongle must be powered by a USB.
- Team does not have prior experience with computer networks.
- Team does not have prior experience working with Linux.

Proposed Technical Solution

Overview
The proposed solution to the problem in accordance with the Functional Design Requirements is divided into several submodules. The LCD/OLED screen will be used to display the host device's system information along with its network settings. The navigation module (push buttons) will be required for the user to configure the network settings of the host device. A custom communication protocol will be required to receive and send data between the host and the host device. The USB port will serve as a physical connection between the host device and the Configuration Dongle to initiate the communication protocol and to power the Configuration Dongle. The system level block diagram is shown in Fig. 1.

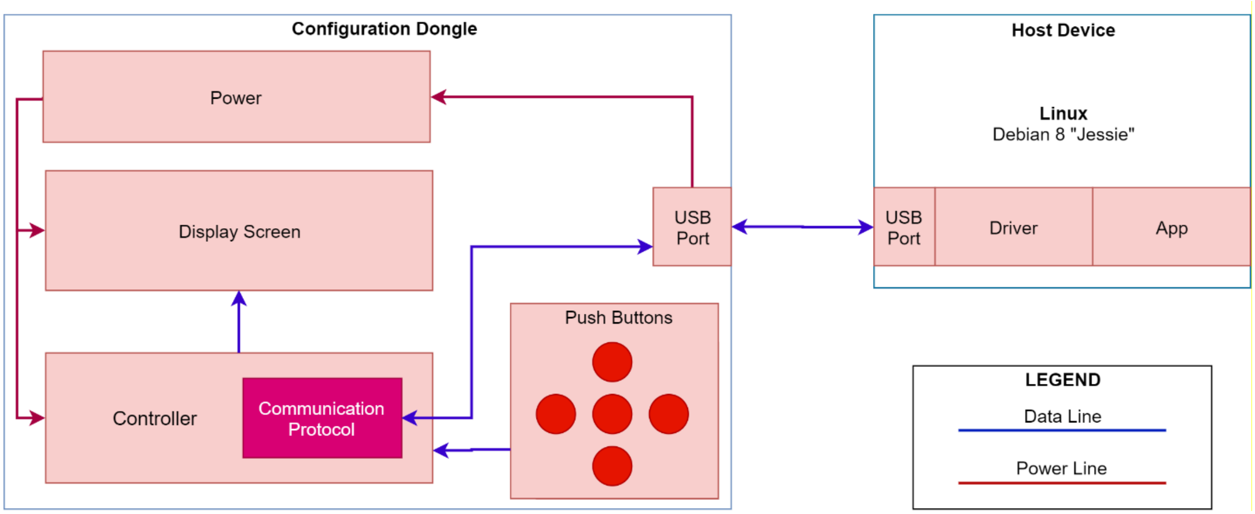

Figure 1. Design Project Block Diagram

Submodule Description
This section is a detailed description of each submodule in the proposed block diagram, Fig. 1.

<u>Controller</u>
A microprocessor/microcontroller will be used as the control unit in the Configuration Dongle instead of a field programmable gated array (FPGA). Although microcontrollers/microprocessors are slower than FPGAs, however, microcontrollers/microprocessors are usually developed with the purpose to achieve a specific task. There are a variety of microcontrollers/microprocessors that are geared towards implementing communication protocols using USB stack which makes them suitable for the purposes of this project. FPGAs are usually faster overall, since they don't use fetch/decode style of sequential instruction execution, however, this is also why they are not suited for the Dongle. Also, microcontrollers/microprocessors are more affordable than FPGAs.

To accomplish the objectives stated in the functional design requirements section, a 32-bit microcontroller/microprocessor with a built-in USB stack will be used. The built-in USB stack will be used to enumerate the designed serial communication protocol, so that the Configuration Dongle can communicate with the host device using a USB port. A 32-bit microcontroller/microprocessor will make the transmission of data with the framing bits and 16-bit CRC faster.

Display

An LCD/OLED screen that can display at least 16x2 characters, according to the specifications, will be used to provide an interactive user interface. The Configuration Dongle will receive current configuration settings from the host device and display them for the user. LCD/OLED screen will also display user commands and the current network settings of the host device, this would achieve the first specification of the project.

User Input

Push buttons will be used to allow the user to change the network settings of the host device. Push buttons are also compact in size when compared to a keypad. Additionally, the functionality of a keyboard is not required. A typical three by three numerical keypad is of size 2.75” by 3”, therefore it cannot be used in the Configuration Dongle design. Since the dongle can only be a maximum size of 3” by 6” with all the sub-modules included.

Incorporating pushbuttons will achieve the second specification of the project and will be used to send the user’s desired network settings for the host device to the microcontroller/microprocessor. The microcontroller/microprocessor will then communicate with the host device to overwrite the host device’s network settings with the desired network settings provided by the user.

USB Port & Power

Since Daktronics’ host devices have USB ports built-in, the ports will be used to communicate with the Configuration Dongle and power the Configuration Dongle.

USB ports traditionally have four channels, two differential channels “D+” and “D-” which are used to receive and transmit data, while the other two channels, “Vbus” and “GND” are used to supply 0V-5V power. This will achieve the third specification of the project.

Communication Protocol with the Host Device

A serial communication protocol will be used to communicate with the host device. The protocol structure will be designed such that the message can be detected unambiguously. The protocol will be able to recover dropped data, will be extendable, and will be updateable.

For unambiguous data detection, the protocol will have message framing. The framing will denote the beginning and the end of the message. A 16-bit CRC polynomial will be used for error detection. To make the protocol extendable, an “extension” tag will be used to denote the message size of 8-bit, 16-bit, 32-bit and 64-bit. To make the protocol updateable, a “version” tag will be used to denote the protocol version. The “version” tag will make the protocol structure such that the future versions of the protocol will not disrupt older implementations. If a future version of the protocol introduces a new message type, then the older version of the system will be able to recognize it through “version” tag and will return an error without losing the message framing.

Anticipated Required Resources

These are the resources that will be required to build and test the Configuration Dongle:

<u>System Level Component List and Cost Estimate</u>

Table 1: Proposed Budget

Item Description	Retail ($)	Anticipated Expense ($) *
Microcontroller/Microprocessor	-----------	$15.86 x2
LCD screen	-----------	$31.76 x2
USB Connector	-----------	$3.84 x2
PCB	-----------	$42.00
Subtotal	-----------	$144.92
Total	-----------	$173.90

* The actual expense may have been after an educational discount.

<u>Test and Laboratory Equipment</u>

- PC with a Debian 8 “Jessie” operating system.
- Oscilloscope
- 5V power supply
- Signal generator
- Bread boards
- Microcontroller/Microprocessor development board
- 3D Printer
- PCB design software
- Light-emitting diodes (LEDs)
- Voltmeter

Proposed Performance Verification

The oscilloscope will be used to debug the communication between the LCD/OLED screen and the microcontroller/microprocessor. The integrated development environment (IDE) will be used to primarily develop and debug the microcontroller/microprocessor code. The LCD/OLED screen along with LEDs will also be used to debug the microcontroller/microprocessor code. The user input will be tested by verifying that the input provided by the user is read into the memory of the microcontroller/microprocessor r which will also be verified by displaying the input to the LCD/OLED screen. The communication protocol will be tested by sending data to the host device and verifying that the host device has received that data. The LCD/OLED screen will also be used to verify that the data being read by the microcontroller/microprocessor from the host device is being displayed to the LCD/OLED screen. A voltmeter will then be used to verify that the USB port is supplying 0V-5V to the microcontroller/microprocessor.

For the system level testing, a generic CDC driver will be installed on the host device (Linux Machine), for the Configuration Dongle to be identified by the host device. Additionally, changes will be made to the generic CDC driver to make the driver compatible with the Configuration Dongle. Then, an application will be developed on the host device to accept instructions from the Configuration Dongle. The application will then be used to test the functionality of the Configuration Dongle. This will include reading the current network settings of the host device and successfully displaying them on the LCD/OLED screen. Then allowing the Configuration Dongle to change the IPv4 address, netmask, and gateway using the push buttons and verifying that the new network settings are successfully updated in the host device.

Progress and Analysis

Controller

For the control unit of the Configuration Dongle, Raspberry Pi, PIC 32, STM32 and ATSAMD21 were considered. After comparing all the above options, ATSAMD21 ARM processor with Feather M0 development board was determined to be the best fit for our project, Fig. 2.

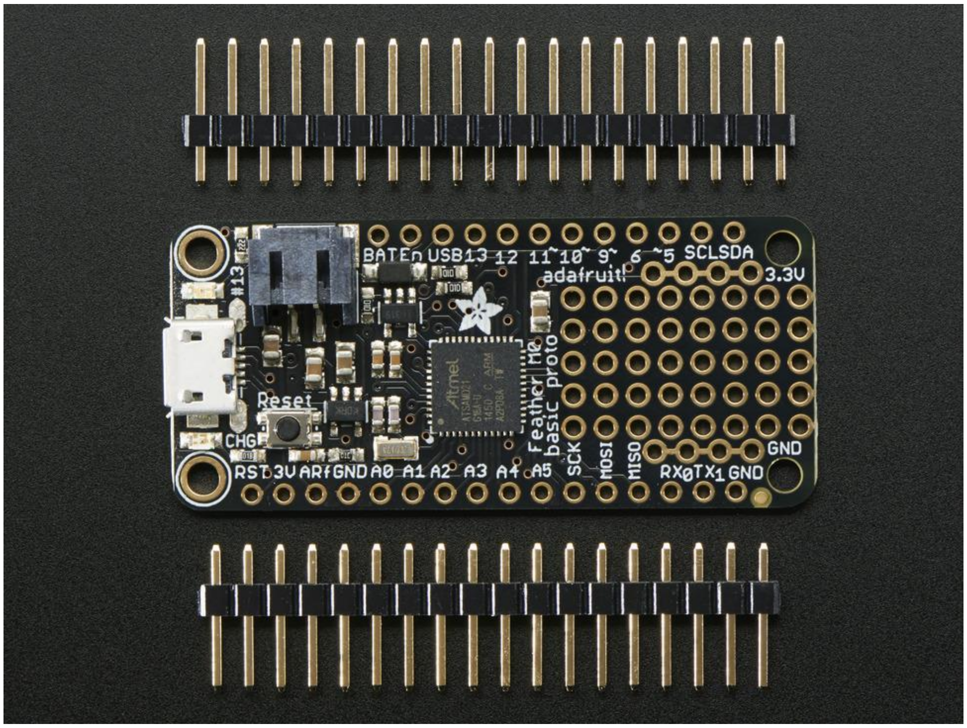

Figure 2. Feather M0 with ATSAMD21

From a manufacturability perspective, Raspberry Pi (RPI), PIC32/PIC18 and STM32 are not suitable for this project. The Configuration Dongle would be faster to develop and prototype using an RPI, however, the end-product will not be as efficient as a purpose-built design. There are LCD/OLED panel “hats” that are available for RPI, but those kits are more expensive than the purpose-built design. Additionally, using “hats” will have more points of failure due to having more components and interconnections. The software running on the RPI is orders of magnitude more complicated than what runs on a microcontroller that makes it harder to debug when something goes wrong with the Configuration Dongle. Additionally, an RPI will be physically larger than the desired size of this project as specified in the specifications section.

The physical size and layout of the LCD/OLED and push buttons "hat" is restricted by the manufacturers of those "hats". Additionally, if the manufacturers of the LCD/OLED and push buttons "hats" stop manufacturing those "hats", the manufacturability of the Configuration Dongle would be at risk. Therefore, a custom design was determined to be the ideal solution for this project

PIC18 is an architecture specific to the PIC microcontrollers. Meaning MPLAB would be required to develop the code for the microcontroller. Additionally, open source toolchains for the PIC microcontrollers currently does not exist. PIC32 is better as it is Microprocessor without Interlocked Pipeline Stages (MIPS) based, but it still requires a proprietary toolchain.

Hence, we decided to go with ATSAMD21 (ARM Cortex-M0 based). STM32F was also considered as a valid option, but it was more expensive as compared to ATSAMD21. The Feather M0 devkit is Arduino compatible, which simplifies the development process. The ATSAMD21 chips are compatible with MPLAB and Atmel Studio as well. Additionally, open source toolchains are widely available for the ATSAMD21 chips.

Display

An OLED screen, Fig. 3, was chosen for this project. The OLED screen has operating voltage of 2.2V-5V and has current draw of 75mA. The microcontroller selection as mentioned above can only supply 3.3V, which makes the OLED screen a perfect fit for our project. Additionally, since Daktronics is in Brookings, SD, the display screen had to be able to sustain harsh weather conditions and the OLED screen is rated at -40ºC which makes it an ideal choice for this project.

Figure 3. OLED Screen

The OLED screen was successfully interfaced using 8-bit SPI using the command Table 2.

Table 2: OLED Instruction Commands

Instruction	Byte
Clear display	0x01
Return cursor to home	0x02
Display on, cursor off	0x0C
Display on, steady cursor	0x0E
Shift cursor left	0x10
Shift cursor right	0x14

User Input

For user input, a tactile 5-button navigation module with LEDs is used instead of discrete normally open single pole single throw (SPST) push buttons. Five different Shapnu's navigation modules were considered, and for the Configuration Dongle SPD5K2 module was selected. Since, SPD5K2 had a smaller size as compared to the other available modules. Additionally, the SPD5K2 had the longest mechanical life of 1,000,000 cycles and had higher range of operating temperature from -25ºC to 60ºC, Fig. 4.

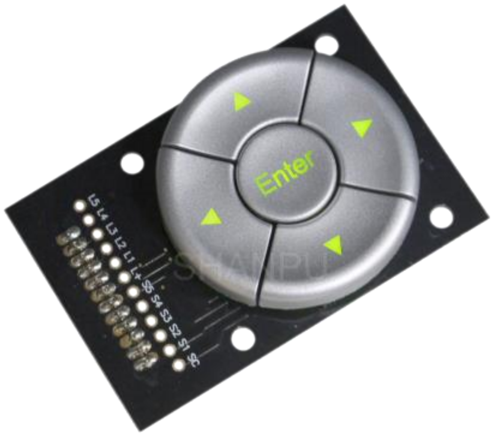

Figure 4. Navigation Module, SPD5K2

Communication Protocol

The communication protocol is designed to be a byte-oriented protocol. This would eliminate the intrinsic endianness issues if the protocol is updated to send more than one byte. The size of the CRC polynomial is chosen to be 16 bits in size, according to the specs, Fig. 4.

Max length at HD / **Polynomial**	**CRC Size (bits)**													
	3	4	5	6	7	8	9	10	11	12	13	14	15	16
HD=2	**0x5**	**0x9**	**0x12**	**0x33**	**0x65**	**0xe7**	**0x119**	**0x327**	**0x5db**	**0x987**	**0x1abf**	**0x27cf**	**0x4f23**	**0x8d95**
HD=3	4 **0x5**	11 **0x9**	26 **0x12**	57 **0x33**	120 **0x65**	247 **0xe7**	502 **0x119**	1013 **0x327**	2036 **0x5db**	4083 **0x987**	8178 **0x1abf**	16369 **0x27cf**	32752 **0x4f23**	65519 **0x8d95**
HD=4			10 **0x15**	25 **0x23**	56 **0x5b**	119 **0x83**	246 **0x17d**	501 **0x247**	1012 **0x583**	2035 **0x8f3**	4082 **0x12e6**	8177 **0x2322**	16368 **0x4306**	32751 **0xd175**
HD=5					4 **0x72**	9 **0xeb**	13 **0x185**	21 **0x2b9**	26 **0x5d7**	53 **0xbae**	52 **0x1e97**	113 **0x212d**	136 **0x6a8d**	241 **0xac9a**
HD=6						4 **0x9b**	8 **0x13c**	12 **0x28e**	22 **0x532**	27 **0xb41**	52 **0x1e97**	57 **0x372b**	114 **0x573a**	135 **0x9eb2**
HD=7								5 **0x29b**	12 **0x571**	11 **0xa4f**	12 **0x12a5**	13 **0x28a9**	16 **0x5bd5**	19 **0x968b**
HD=8									4 **0x4f5**	11 **0xa4f**	11 **0x10b7**	11 **0x2371**	12 **0x630b**	15 **0x8fdb**
HD=9													5 **0x5a47**	6 **0xe92f**
HD=10														5 **0xed2f**
HD=11														

Figure 5. CRC-16 polynomial

It is evident the table above that the Hamming Distance (HD) is the number of detectable errors in a single message, the columns are the CRC size, and the intersection is the maximal message length that the given CRC can detect the given number of errors for. Since the protocol is byte oriented (0 –255), therefore, CRC-16 at HD of 5 and word length of 241 is chosen. This will work perfect with our message of size 8-bits and will detect errors up to length 241.

Furthermore, similarly like UART, the protocol will also have framing so that the message can be unambiguously detected.

<u>Test bench</u>
An Apple Mac Pro PC was obtained by Dr. Fourney, and a Debian 8 “Jessie” Linux OS was successfully setup and installed on the PC. Connecting the PC to the Internet in the Senior Design Lab remains a work in progress. The team is working Mr. Flaskey to connect the PC to the Internet.

Project Schedule

Plan of Action

To manage the tasks required to complete the project and to keep track of the objectives, a Gantt chart was created, Fig. 6.

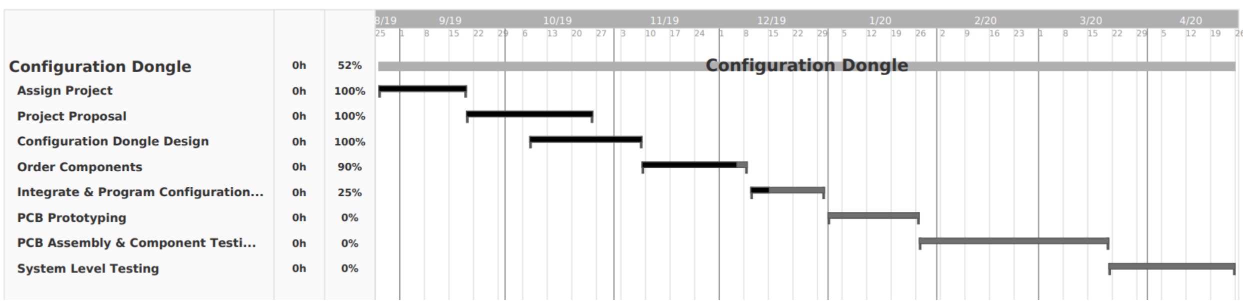

Figure 6. Plan in the form of a Gantt chart

The team will continue to collaborate and work together a minimum of 20 hours each week and to accomplish all the objectives of the project. Additionally, the team will set milestones for each consecutive month to achieve long term goals and to stay ahead of schedule. To further help recover time, the team will continue to work on the project during winter break.

Conclusion

The team is currently done ordering all the components for this project and were able to interface the OLED and the pushbuttons with the ATSAMD21. The team will finish setting up the test bench before the end of finals week and will start working on the USB stack and the communication protocol during winter break. The team will then work on designing a prototype PCB. After the prototype PCB have been tested and all the changes have been finalized, the team will order a final version of the PCB. The final version of the PCB will be assembled, and each submodule will be tested to make sure all the submodules accomplish the objectives and the specification of the project. Finally, the team will enter the system level testing. In this phase a performance verification document will be created to test and verify that our system level testing achieves the objectives and specifications of the project.

Team Contribution

This section of the report highlights the team members contribution to the project, which was recorded in Table 3.

Table 3: Team Members Contribution

Name	Moe	Harsh	Total
Test Bench Setup	100%	0%	100%
Controller Selection	30%	70%	100%
Display Screen Selection	60%	40%	100%
Push Buttons Selection	40%	60%	100%
Communication Protocol Polynomial	0%	100%	100%
Setup RPI Interface Push Buttons & LCD/OLED	100%	0%	100%
Setup ATSAMD21 Interface Push Buttons & LCD/OLED	0%	100%	100%
Soldering	40%	60%	100%
Demo: Interface Push Buttons & LCD/OLED	80%	20%	100%
Testing/Debugging	50%	50%	100%
Weekly Memos	100%	0%	100%
Documentation: Proposal & Report	70%	30%	100%
Presentation	80%	20%	100%

Appendix A

Matt Sickler's Signature

Matt Sickler
to me

3:26 PM (1 hour ago)

Hello Harsh,

I am at an all day planning event today and tomorrow. Does this need my actual signature, or can I just say in email that I sign off on it? If that's enough, consider this email such a sign off. 😊
If not I'll have to go back to Daktronics tonight to print/sign/scan the document.

Thanks,
Matt

From: Harsh Dubey <had7143@gmail.com>
Sent: Tuesday, October 22, 2019 6:01 PM
To: Matt Sickler <Matt.Sickler@daktronics.com>
Subject: Re: Status report and proposal - Senior Design

This email originated outside of Daktronics. Use caution when opening links or attachments. Report questionable emails to ITHelp.

...

Harsh Dubey <had7143@gmail.com>
to Matt

4:14 PM (51 minutes ago)

Hey Matt,

This email would work for Dr. Tan, I'll type in your name in the report and show Dr. Tan this email. You don't have to go to Daktronics. Also, thanks for responding.

Regards
Harsh

References

[1] Ferguson, Greyson, et al. “How to Connect a Printer Using Bluetooth.” *Techwalla*. Accessed on: Oct. 8, 2019. [Online]. https://www.techwalla.com/articles/how-to-connect-a-printer-using-bluetooth.

[2] Jonas. “Change Windows IP Address Settings Remotely with Psexec and Netsh.” *Jonamiki.com*, 26 Jan. 2018. Accessed on: Oct. 8, 2019. [Online]. http://jonamiki.com/2014/12/19/change-windows-ip-address-settings-remotely-with-psexec-and-netsh/.

[3] Mitchell, Bradley. “How to Change Your IP Address (And Why You Might Want To).” *Lifewire*, Lifewire, 4 Oct. 2019. Accessed on: Oct. 8, 2019. [Online]. https://www.lifewire.com/change-your-ip-address-818150.

[4] “Network Configuration.” *Network Configuration - an Overview | ScienceDirect Topics*. Accessed on: Oct. 8, 2019. [Online]. https://www.sciencedirect.com/topics/computer-science/network-configuration.

[5] Ruddy. “WiFi Dongle - What Is It and How Does It Work?” My Webspot - Pocket WiFi Rental, 10 Jan. 2019. Accessed on: Oct. 8, 2019. [Online]. https://www.my-webspot.com/blog/wifi-dongle-how-it-work/.