# An Advanced Networking Outreach Activity for Kids

DESIGN DOCUMENT


sddec20-05

Dr. Tom Daniels

Grayson Cox | UI Developer | Agile Project Manager

Austin Dvorak | Network Systems Manager

Ryan Newell | System Admin

Spencer Parry | UI Developer

Ross Thedens | Hardware Systems Manager | Meeting Secretary

Team Email: sddec20-05@iastate.edu

Team Website: http://sddec20-05.sd.ece.iastate.edu/



Revised: February 23, 2020 (V1)

# Executive Summary

## Development Standards & Practices Used

- Standards
  - Quality management (ISO 9001)
  - Developing information for users in an agile environment (ISO/IEC/IEEE 26515)
- Practices
  - Agile development
  - Code reviews
  - Weekly meetings
  - Rigorous testing for security and reliability

## Summary of Requirements

- Support mesh and dynamic ad-hoc networking capabilities
- Ability to stream data between nodes
- Easy to use and set up
- Hardware should be mobile and durable
- Can be operated without the need for an Internet connection
- Range between nodes should support a 40 foot radius (when in eye-sight)
- Nodes will consist of a single-board computer, rechargeable battery, and camera/sensors

## Applicable Courses from Iowa State University Curriculum

- CprE 489 - Computer Networking and Data Communications
- CprE 430 - Network Protocols and Security
- CprE 537 - Wireless Network Security
- CprE 230 - Cyber Security Fundamentals
- ComS 252 - Linux Operating System Essentials
- SE 319 - Construction of User Interfaces
- SE 329 - Software Project Management
- ComS 309 - Software Development Practices

## New Skills/Knowledge acquired that was not taught in courses

- Web Development with Angular/Typescript
- Ad-Hoc/Mesh Networking and Router Configurations
- Raspberry Pi Configuration

# Table of Contents

1. Introduction **4**

1.1 Acknowledgement 4

1.2 Problem and Project Statement 4

1.3 Operational Environment 4

1.4 Requirements 5

1.5 Intended Users and Uses 5

1.6 Assumptions and Limitations 6

1.7 Expected End Product and Deliverables 6

2. Specifications and Analysis **7**

2.1 Proposed Approach 7

2.2 Design Analysis 7

2.3 Development Process 7

2.4 Conceptual Sketch 9

3. Statement of Work **13**

3.1 Previous Work And Literature 13

3.2 Technology Considerations 13

3.3 Task Decomposition 13

3.4 Possible Risks And Risk Management 13

3.5 Project Proposed Milestones and Evaluation Criteria 13

3.6 Project Tracking Procedures 13

3.7 Expected Results and Validation 13

4. Project Timeline, Estimated Resources, and Challenges **14**

4.1 Project Timeline 14

4.2 Feasibility Assessment 14

4.3 Personnel Effort Requirements 14

4.4 Other Resource Requirements 14

4.5 Financial Requirements 14

5. Testing and Implementation **15**

5.1 Interface Specifications 15

5.2 Hardware and software 15

5.3 Functional Testing 15

5.4 Non-Functional Testing 15

5.5 Process 15

5.6 Results 15

6. Closing Material **16**

6.1 Conclusion 16

6.2 References 16

6.3 Appendices 16

# Figures and Tables

## Figures

1: Conceptual Sketch 9

2: System Modules 11

# 1. Introduction

## 1.1 Acknowledgement

We would like to acknowledge our client and advisor Dr. Tom Daniels for his guidance and support on this project.

## 1.2 Problem and Project Statement

Wireless networking has become a fundamental part of many people's everyday lives. This is especially true for students across grades 4-12, who are beginning to spend hours each day on school-issued internet-connected devices. However, few students understand the technology that underpins these networks. The internet is often portrayed as an amorphous portal granting access to huge amounts of information. In reality, this information is stored on servers around the world and delivered to end users via routing protocols. By creating a teaching tool that demonstrates routing protocols, we can illustrate the challenges involved in creating a reliable wireless network, increase technology literacy, and stimulate students' interest in computer networking careers.

Our approach involves creating a teaching toolkit consisting of a set of portable wireless network nodes, a network monitor/configuration application, and a set of lesson plans. A grade school instructor with no computer networking training will be able to  operate it. The nodes will automatically form an ad-hoc network with one another when powered on (and within range of one another). Network data will be provided by video cameras and various sensors on the nodes, such as temperature sensors. The data will travel the network to reach a master node which is directly connected to a network monitor application running on an instructor's laptop. The monitor application will control which sensor/video streams are delivered to the master node and display the data received. The lessons will challenge students to transmit data across a long distance (beyond the range of a single node; from the main office to a classroom, for instance) to the network monitor application. As students add nodes into the network and position them to attain connectivity, they will observe the routing of the data (which nodes are reached) in the monitor application. They will use this information to figure out where data is unable to travel and rearrange the nodes accordingly.

Ultimately, our toolkit will act as a miniature mock-up of the real internet. Students will be able to appreciate how data from a web server hops from node to node over the internet before reaching their devices. Following the activities, the instructor will discuss grade level-appropriate topics related to networking from the lesson plan with the class. For younger students, this may involve the fact that wireless signals have a limited range, while for older students it may include a discussion of the routing decisions made by the ad-hoc protocol. Our project will provide a flexible way to teach computer networking to grade school students.

## 1.3 Operational Environment

A collection of networking nodes will compose the main aspect of the system. It is expected that these nodes will be used indoors, but they should be able to operate outdoors as well. They do not need to be weatherproof, however, they will need to be durable enough to protect the internal components.

There will be a graphical user interface aspect to the system, which will allow users to view sensor and video data transmitted from the nodes. This can be accessed via any web browser on a separate computer, and no internet connection should be necessary as long as there is a connection between the computer and the base node.

### 1.4 Requirements

Nodes will need to be powered by rechargeable batteries so that they do not require a power cord when in use. They will also connect wirelessly to other nodes. Both of these requirements ensure that nodes are mobile and can be used without the restriction of cords.

Each node should support ad-hoc networking capabilities. Multiple networks should be able to coincide and stay independent of each other. Nodes should use their respective networks and safely handle interference and other network errors. To demonstrate functionality, nodes will send data across the network. Camera feed, sensor data, and control signals will need to be transmitted through nodes to their destination. The graphical user interface will display this data and give a visualization of the network.

The packaging of each node will need to be durable enough to survive drops since they will be moved around quite frequently. Each node will consist of a single-board computer, a wireless network adapter, and a rechargeable battery. Some nodes will contain sensors and cameras to serve as data sources for the network. Components will be bundled together in a package that makes it intuitive to charge and easy to use and set up.

Besides being wireless and easy to place, the software should also be user-friendly and intuitive to use. The instructor may not have much knowledge on computer systems so the monitoring application should be simple enough that most kids and adults can understand the interface. To aid in the ease of set up, nodes will not need to be connected to an existing network. The only network that they will use is the one they create themselves.

### 1.5 Intended Users and Uses

The project has three main types of users: students, instructors, and system administrators. The goal of the project is to educate students on how mesh networking works, therefore, the students would be the target audience. The students would not interact with the GUI as much as the instructors or teachers would, but the GUI should be user friendly enough for anyone in the three target audience groups to use. The students largely will be moving or placing the nodes in different places, and the instructors will use the GUI to interpret the data to the class.

The instructors would have knowledge on how to use the nodes and the GUI so they can teach their students some of the principle lessons that these devices could teach, such as range and interference. They would know how to position the nodes to display on the GUI how different forms of interference can cause the video to buffer more or less than other forms of interference. The instructor would also use the GUI to change settings on the nodes, like grouping nodes into two separate groups.

The system administrators would have access to the file system and command line of the nodes to manage updates or to diagnose problems with the nodes. The administrators would only have to be involved if there is a problem with one of the nodes, or a mandatory update needs to take place. Students and instructors would not have access to the system administrator settings, however, system administrator would have full system access, as well as access to the student and instructor GUI.

### 1.6 Assumptions and Limitations

Assumptions

- The user owns a laptop computer with a web browser
- Person running this activity is able to charge the nodes when batteries run low

Limitations

- The project will be completed by the end of the fall 2020 semester.
- Nodes will need to be mobile
- Internet connection may not be available
- Each node will be powered by a single-board computer
- Storage of nodes is tight
- Battery life should last at least two hour

### 1.7 Expected End Product and Deliverables

The main deliverables will be a set of at least eight wireless, rechargeable nodes. Some of which will be sensor nodes and the rest will be relay nodes whose purpose is only to conduct data from the sensor nodes. Each sensor node will have one sensor for collecting data from an external source and sending it into the network to be viewed by the users on visual display. We expect that at least one of these sensor nodes will have a camera; the selection of other sensors will be made at a later date. We anticipate delivery of functioning relay nodes by October 1 because of the simplicity in their design. As for the sensor nodes, we expect to deliver a functioning set by October 15.

In order for the users to access the data within the network, there will be a base station, which acts as the gateway between the instructor's machine and the node network. The instructor will be able to connect to this base station and access the user interface component. This base station component is to be delivered along with the set of sensor nodes by October 15.

The user interface component will be in the form of a web-based application that can be accessed through any web browser. The UI will offer at least two displays: one large display for visualizing a specific stream of data and one display for controlling the nodes and the large display. The idea is that the instructor can control the system on a small screen and show the large display on a projector or television for the students to see. We anticipate delivery of a functional UI by the end of October.

Lastly, to assist educators in using this technology, we will prepare a set of lesson plans and activities which demonstrate the workings of wireless networking. This deliverable serves as a vehicle for demonstrating our project, so we set the delivery date to be sometime during the last week before our final presentation of the project.

# 2. Specifications and Analysis

## 2.1 Proposed Approach

Recalling the Summary of Requirements, there are some major objectives that this project needs to meet. The most limiting requirement is the budget. Due to wanting to keep each node at $65, limits the hardware. This means that our approach to what operating system, networking protocols, etc. we use changes due to the limited processing power of the device.

A single board computer was deemed the correct choice due to the low cost ~$35 per board, but also the higher processing power and IO abilities over embedded boards, along with the high mobility. To increase this mobility, a battery pack would be included to allow the nodes to work without wall power. The next step was deciding how the networking would be handled. Certain distributions of Linux aimed at computer networking come with support for mesh networking protocols, so the decision has to be made about which one to use. These distributions should allow for automatic connecting and disconnecting of links to aid with ease of use. A web server would be set up on one of the links to allow for a cross-platform GUI that would show all of the required information coming from the network, along with easy configuration of the network and nodes.

## 2.2 Design Analysis

Deciding on the hardware of the project seemed to be relatively easy. Due to the widespread support and price of the board, Raspberry Pi's will be used as the nodes. This also makes figuring out which mesh-networking protocol and linux distribution easy. After research, it was found that IPFire and OpenWRT were the best distributions to use, as there were already examples of these distributions working on the Raspberry Pi. Selecting the ad-hoc networking protocol was the next step.

There are 3 main types of protocols: proactive, reactive, and hybrid. Proactive protocols create a list of connections and their routes at the expense of latency when adding and subtracting nodes, while Reactive floods the network with packets at the expense of latency when sending packets. Hybrid routing creates a table initially and floods only when it cannot find a path to the destination. Due to how much bandwidth flooding a network stream with video packets would require, the decision was made to go with a proactive protocol.

There are 5 main proactive routing protocols: OSLR, Babel, DSDV, DREAM, and BATMAN. BATMAN and OSLR are the two most prominent protocols and are both supported by OpenWRT and IPFIRE. Further testing is required to decide on which distribution and protocol would be best for our project.

For the GUI, It was decided that Angular/Typescript would be used for the UI due to the ease of use and since the members working on the UI have had previous experience in Angular development.

### 2.3 Development Process

Because of the lack of certainty in the design of this project, our team requires an iterative development method that allows for design decisions to take place at any point in the project lifecycle. And given the vagueness of our initial project description, we need an approach that will assist us in reaching our final design starting with a high-level understanding of the project. Therefore, we will conform to an Agile development process with a top-down design approach because it mitigates the risks associated with the inherent variability of this project.

The development lifecycle of our project will consist of a number of sprints, each lasting two weeks. Prior to each sprint, the team will hold a sprint planning meeting in which we choose tasks/features to implement during the coming sprint. Following each sprint, the team will meet for a sprint retrospective meeting, where we will discuss our progress and think critically about our performance.

We will be using the tools provided by GitLab for our artifacts. Individual units of work (i.e., tasks, features, or user stories) will be represented as Issues in GitLab, which can be created by and assigned to any team member. Each Issue is associated with a single Git branch, so Issues are loosely coupled, allowing team members to work independently.

To control the quality of our implementations, all code will go through peer reviews before going into production. These code reviews will be facilitated via merge requests in GitLab. Once a team member finishes work on an Issue, he will open a merge request and assign at least one other team member to review the work.

To track the progress of the team, Issues will appear on sprint boards. We will use different boards to denote the status of every Issue. The board in which an Issue resides will indicate whether it is Open, InProgress, InReview, or Closed.

## 2.4 Conceptual Sketch

Our conceptual sketch of the project is shown in Figure 1.

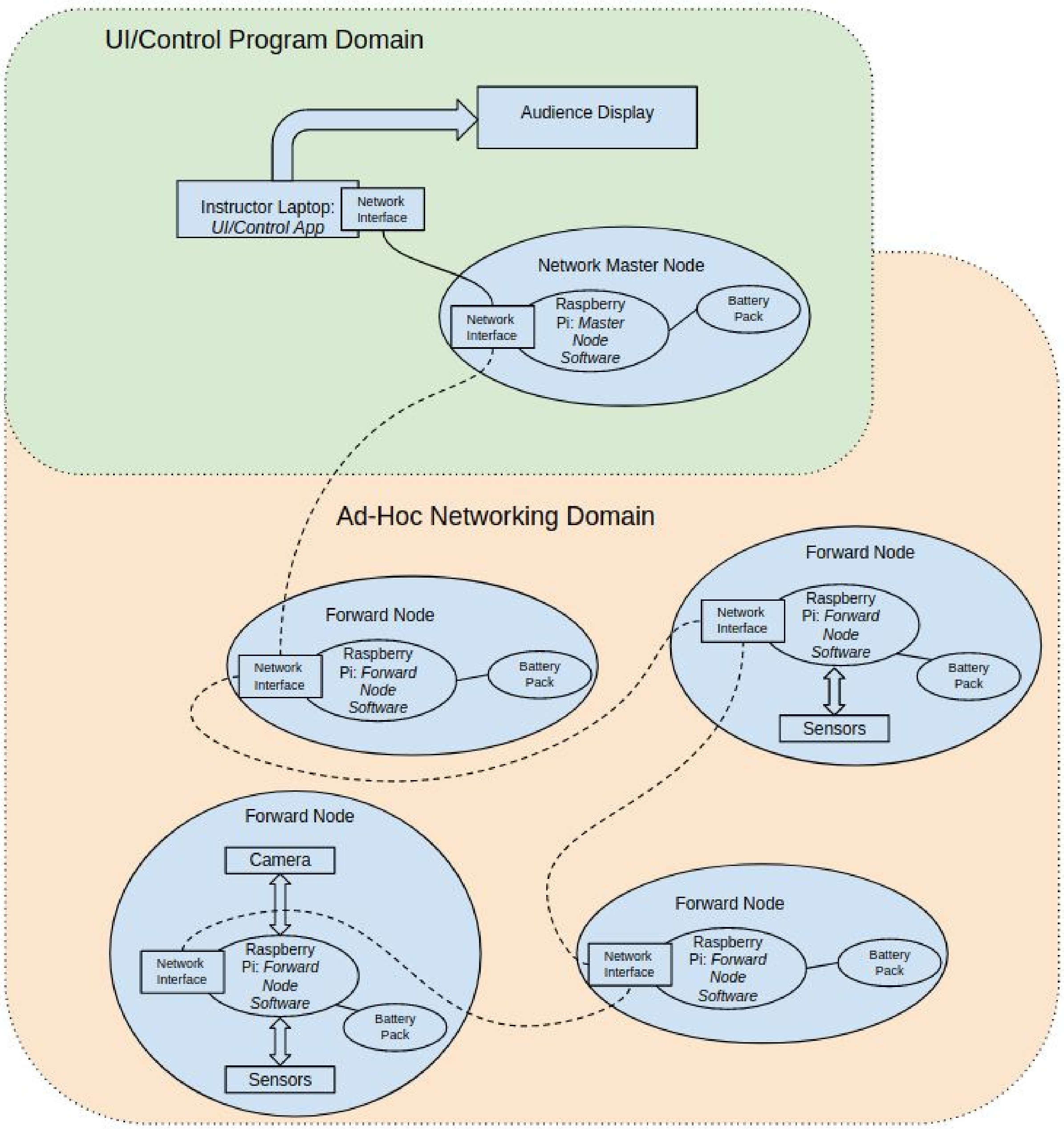


*Figure 1: Conceptual Sketch*

Our project consists of two major domains: a UI/Control Program domain and an Ad-Hoc Networking domain. In the UI/Control Program domain, the instructor's laptop runs the UI/Control frontend software, which allows users (students and the instructor) to initiate data transfers through the network and observe the routing information (nodes reached, timing information, etc.). The routing information is shown on the Audience Display, which may be the laptop screen or an external display (such as a projector). The instructor's laptop is connected via a dedicated network interface (ethernet or a USB WiFi adapter) to the network interface of the network master node. This allows it access to the ad-hoc network of nodes, which are not internet-connected. This connection will be open for the duration of the system's operation. The master node will have a special software load that allows it to participate in the ad-hoc networking of the other nodes while providing data/controls to the instructor's laptop via the UI/Control backend.

In the Ad-Hoc Networking domain, each forward node will consist of a Raspberry Pi board with a wireless network interface that supports ad-hoc networking. Each board will run a Linux distribution specialized for routing, such as OpenWrt. The connections from node to node will be managed by an ad-hoc networking protocol such as B.A.T.M.A.N or OLSR. On top of the Linux operating system, there will be other drivers and software to capture data from cameras and sensors connected to some of the nodes (via USB). This data will be transmitted back to the master node and to the instructor's laptop at the behest of the UI/Control Program, which sends control signals to the nodes in the network to initiate transfers. All nodes (forward and network master) will be powered via rechargeable battery packs and stored in portable, durable, drop-proof casings. Each set of nodes distributed to instructors will be programmed with a unique identifier so that all node sets form separate ad-hoc networks, even when used within range of one another.

Figure 2 (next page) shows the system modules involved in our project for each physical computing device involved. This includes the forward nodes, the network master node, and the instructor laptop.

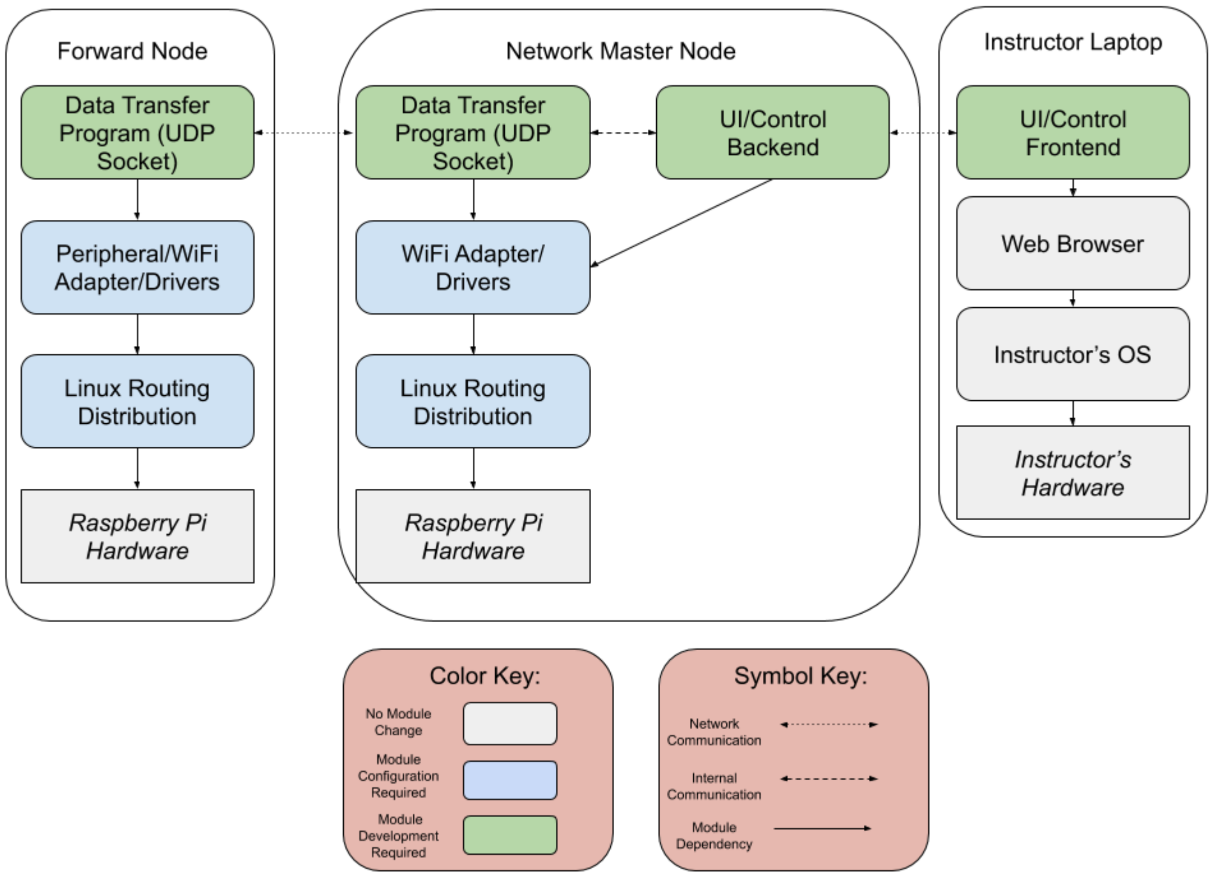


*Figure 2: System Modules*

The sharp-cornered blocks represent the base hardware layers of each device, while rounded-cornered blocks represent primarily software modules. As shown by the color key, the gray modules already exist and require no special configuration. The blue modules exist but will need to be configured/installed as part of the project. The green modules do not exist and will need to be developed as part of the project.

The arrows indicate the dependencies between layers. For forward/network master nodes, the data transfer program will rely on device drivers for the network interfaces and sensor/camera peripherals. The drivers will be installed on top of a Linux routing distribution, which will be installed on the Raspberry Pi hardware. Network communication will be initiated in the data transfer program, although the actual data transfer and routing is handled by the routing distribution and network interfaces. There will be a special connection between the data transfer program and the UI/Control program backend on the network master node; this will allow the UI/Control program to send and receive network data. The data transfer program and UI/Control program backend will run concurrently on the network master node. Since the frontend runs in a browser, no special configuration is required on the instructor's laptop. Once the master node is connected and started, the instructor will simply have to launch a web browser and load the app's webpage.

# 3. Statement of Work

## 3.1 Previous Work And Literature

Include relevant background/literature review for the project

– If similar products exist in the market, describe what has already been done

– If you are following previous work, cite that and discuss the **advantages/shortcomings**

– Note that while you are not expected to "compete" with other existing products / research groups, you should be able to differentiate your project from what is available.

Detail any similar products or research done on this topic previously. Please cite your sources and include them in your references. All figures must be captioned and referenced in your text.

## 3.2 Technology Considerations

Highlight the strengths, weakness, and trade-offs made in technology available.

Discuss possible solutions and design alternatives

## 3.3 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and to understand interdependence among tasks.

## 3.4 Possible Risks And Risk Management

Include any concerns or details that may slow or hinder your plan as it is now. These may include anything to do with costs, materials, equipment, knowledge of area, accuracy issues, etc.

## 3.5 Project Proposed Milestones and Evaluation Criteria

What are some key milestones in your proposed project? Consider developing task-wise milestones. What tests will your group perform to confirm it works?

## 3.6 Project Tracking Procedures

What will your group use to track progress throughout the course of this and next semester?

## 3.7 Expected Results and Validation

What is the desired outcome?

How will you confirm that your solutions work at a **High level**?

# 4. Project Timeline, Estimated Resources, and Challenges

## 4.1 Project Timeline

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 3.3) and associated subtasks versus the proposed project calendar. The Gantt chart shall be referenced and summarized in the text.

– Annotate the Gantt chart with when each project deliverable will be delivered

• Completely compatible with an Agile development cycle if that's your thing

How would you plan for the project to be completed in two semesters? Represent with appropriate charts and tables or other means.

Make sure to include at least a couple paragraphs discussing the timeline and why it is being proposed. Include details that distinguish between design details for present project version and later stages of project.

## 4.2 Feasibility Assessment

Realistic projection of what the project will be. State foreseen  challenges of the project.

## 4.3 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be based on the projected effort required to perform the task correctly and not just "X" hours per week for the number of weeks that the task is active.

## 4.4 Other Resource Requirements

Identify the other resources aside from financial, such as parts and materials that are required to conduct the project.

### 4.5 Financial Requirements

If relevant, include the total financial resources required to conduct the project.

# 5. Testing and Implementation

Testing is an extremely important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested
3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

### 5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

### 5.2 Hardware and software

– Indicate any hardware and/or software used in the testing phase

– Provide brief, simple introductions for each to explain the usefulness of each

### 5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

### 5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

### 5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

### 5.6 Results

– List and explain any and all results obtained so far during the testing phase

– Include failures and successes

– Explain what you learned and how you are planning to change it as you progress with your project

– If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-Modeling and Simulation: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the implementation Issues and Challenges.

# 6. Closing Material

## 6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

## 6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

## 6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.