



Development Plan

Cyclops Ride Assist: Real-Time Monitoring System

Team 9

Aaron Li (lia79)

Amos Cheung (cheuny2)

Amos Yu (yua25)

Brian Le (leb7)

Manny Lemos (lemosm1)

Table of Contents

- [1. Revision History](#)
- [2. Version Control](#)
 - [2.1. Contributing Workflow Plan](#)
 - [2.2. Regression Testing Workflow Plan](#)
- [3. Roles and Responsibilities](#)
 - [3.1. Aaron Li](#)
 - [3.2. Amos Cheung](#)
 - [3.3. Amos Yu](#)
 - [3.4. Brian Le](#)
 - [3.5. Manny Lemos](#)
- [4. Team Communication Plan](#)
- [5. Team Meeting Plan](#)
- [6. Overall System Workflow Plan](#)
- [7. Demonstration Plans](#)
 - [7.1. Proof Of Concept Demonstration](#)
 - [7.2. Rev0 Demonstration](#)
 - [7.3. Rev1 Demonstration](#)
- [8. Software Development and Engineering Tools and Technology](#)

List of Tables

- [Table 1.1: Revision History](#)
- [Table 6.1: System Workflow Plan](#)
- [Table 7.1.1: Proof Of Concept Topics](#)
- [Table 7.2.1: Rev0 Topics](#)
- [Table 7.3.1: Rev1 Topics](#)
- [Table 8.1: Software Technologies](#)
- [Table 8.2: Hardware Technologies](#)

List of Figures

1. Revision History

Table 1: Revision History

Date	Developer(s)	Change
2022-09-22	Brian Le	Document created
2022-09-23	Brian Le	Detailed steps and tools
2022-10-20	Amos Yu	Improved formatting
2022-11-09	Aaron Li	Updated content
2023-04-04	Amos Yu	Updated Development Plan for Rev1

2. Version Control

Members are expected to use the public GitHub repository for all version control and collaboration. Members will also be expected to maintain the repository by contributing clean and accurate code. Members are required to create new branches when developing different aspects of software before merging into the main branch once functionality has been verified, approved, and is stable. Changes are highly encouraged to be made through multiple commits with a new commit made per module or issue. All commits made should have an associated issue number and issue that describes what needed to be improved. Pull requests are to be processed with the validation of at least two other members within the team with discussions encouraged to be held within their respective topics. Other larger scope and big picture discussions, such as meeting minutes, should take place in the Discussions tab of the GitHub repository.

2.1. Contributing Workflow Plan

When contributing new features, team members of Cyclops are expected to follow this general outline as a workflow:

1. Before beginning work on a new module, pull any new changes from the master branch.
2. Create a new branch to develop on if a branch doesn't already exist for this module.
3. Create a detailed plan of the structure of the module.
4. Implement the modules/functions.
5. Perform unit testing on the modules and its functions.
6. Push any changes to the current working branch.
7. Repeats steps 4 through 6 towards the completion of the module.
8. Merge new functionality to the main branch after code review from 2 other team members.

This workflow is fully outlined in greater detail in the repository [Wiki](#).

2.2. Regression Testing Workflow Plan

When fixing bugs, team members of Cyclops are expected to follow this general outline as a workflow:

1. Create an issue in the Github page.
2. Create a new branch if necessary with the name of the change.
3. Perform unit testings to verify that the issue is still prevalent.
4. Perform regression testing to ensure that there are no further issues appearing.
5. Merge to the Master branch.
6. Close the issue.

This workflow is fully outlined in greater detail in the repository [Wiki](#).

3. Roles and Responsibilities

3.1. Aaron Li

1. To be the main point of contact between the team and the course instructors.
2. To oversee the writing of clean, concise, and consistent documentation.
3. To prepare meeting minutes before and after weekly meetings.
4. To collaborate on the mechanical design of the project.

3.2. Amos Cheung

1. To address peer review feedback given by course instructors or other teams.
2. To maintain a list of team expenses.
3. To be the main point of contact between the team and suppliers of third-party components.
4. To collaborate on the software for the rear vehicle detection module of the system.

3.3. Amos Yu

1. To lead the development of the mechanical design of the project.
2. To oversee the smooth integration of all the components of the system.
3. To collaborate on the software for the crash identification and video export modules of the system.
4. To produce all team media (logos, charts, slides, videos).

3.4. Brian Le

1. To collaborate on the testing software of the project.
2. To conduct market research on currently available products and technologies.
3. To collaborate on the mechanical design of the project.
4. To address peer review feedback given by course instructors or other teams.

3.5. Manny Lemos

1. To lead the development of the software of the project.
2. To lead the development of the electrical design of the project.
3. To oversee the prototyping and field testing process.
4. To plan a team workflow that ensures that deadlines for deliverables are met on time.

4. Team Communication Plan

The team will be communicating primarily using a Facebook Messenger group chat. This application allows the team to instantly send messages to one another. Furthermore, notifications have been enabled to ensure that each team member receives an alert. Communication with course instructors will take place over email and Microsoft Teams. In-person meetings will be held as well as a way to discuss progress and issues as the development of the project continues. This will be further discussed in [Section 5](#). Team members will be expected to communicate frequently and in a timely manner with one another. In the case of an emergency, personal contact information has been distributed between the team members. In the case of conflicts, the team will discuss one-on-one to determine the course of action.

5. Team Meeting Plan

The team will be meeting once a week officially every Tuesday from 2:30 - 3:30pm on the McMaster campus or in a team member's house. During these meetings, there will be a time to present the progress that has been done on the previous meeting's agenda and checklist. Furthermore, next steps for the development will be discussed and addressed with each member receiving an equal amount of work for the week. All members are expected to be done their part by the next meeting.

All members are expected to be timely for each meeting. In the case that a member cannot make a meeting, they must use the official team communication application, Facebook Messenger, to alert the team of their

absence. They will still be expected to complete their portion of work for the week. At the team meeting, each member will have an opportunity to ask questions to one another in order to further the understanding of the overall project. One of the team members will be responsible for taking notes using the formatting outlined in the repository Wiki. These notes will be made available in the repository Discussions tab.

6. Overall System Workflow Plan

The following are steps that the Cyclops team will aim to complete.

Table 6.1: System Workflow Plan

Step	Goal	Technologies	Soft Deadline
1	Source all components	LiDAR, Camera, ADXL345, Raspberry Pi, Buttons	Early October 2022
2	Build a working rear vehicle detection system on a breadboard	Python	Late October 2022
3	Create CAD designs for the system housing	Autodesk Inventor	Early November 2022
4	Build a working crash identification system on a breadboard	Python, ffmpeg	Early December 2022
5	Fabricate all the CAD designs	Prusa 3D Printer	Early January 2023
6	Assemble the full project		Early February 2023
7	Conduct field testing to optimize the monitoring algorithms	Bicycle	Early March 2023
8	Finalize software	Python	Late March 2023

7. Demonstration Plans

Cyclops Ride Assist is a complex electromechanical system with software and hardware interfacing. There are many risks that will need to be addressed before the Proof of Concept Demonstration. Some of these risks include learning Python and the Raspberry Pi IDE and using the OpenCV library, creating datasets and training machine learning models to accurately detect vehicles, and merging mechanical and electrical components to create one unified design and product.

7.1. Proof Of Concept Demonstration

The Proof of Concept demonstration will consist of the following main components.

Table 7.1.1: Proof Of Concept Topics

Topic	Description
Physical Components	The first iteration of the design will be shown to house the Raspberry Pi safely. The housing and the files will be presented.
Electrical Components	The breadboard will be shown with the existing wires and connections to the Raspberry Pi. The breadboard will be neat and exhibit a high level of understanding of electrical concepts.
Crash Identification	The accelerometer will be wired correctly to the Raspberry Pi and will be able to output data into a Python graph using numpy.
Rear Vehicle Detection	The sensor will be able to detect the presence of objects and output the distance to the command line.
Video Capture and Export	The front-facing camera will be able to record and save footage.
Power Source	The Raspberry Pi will be plugged into the wall.

7.2. Rev0 Demonstration

The Rev0 demonstration will improve on the [Proof Of Concept Demonstration](#) and consist of the following main components.

Table 7.2.1: Rev0 Topics

Topic	Description
Physical Components	The first iteration of the design will be printed and contain the entire system.
Electrical Components	The electrical system will be transferred from the breadboard into the physical housing.
Crash Identification	The accelerometer will provide data to a rudimentary crash detection algorithm.
Rear Vehicle Detection	The sensor will be able to detect the presence of objects and consistently output the distance to an array of LEDs.
Video Capture and Export	There will be a demonstration of the front-facing camera saving a certain amount of footage on crash detection or user input.
Power Source	The Raspberry Pi will be plugged into the wall.

7.3. Rev1 Demonstration

The Rev1 demonstration will improve on the [Rev0 Demonstration](#) and consist of the following main components.

Table 7.3.1: Rev1 Topics

Topic	Description
Physical Components	The second iteration of the design will be printed and assembled. The entire system will be securely mounted on a bicycle.
Electrical Components	The electrical system will be fully contained within the system.
Crash Identification	The crash detection algorithm will reliably detect crashes during a live demonstration.
Rear Vehicle Detection	The sensor will be able to detect the presence of objects behind the bike and consistently output the distance to an array of LEDs.
Video Capture and Export	There will be a demonstration of the front-facing camera saving a certain amount of footage on crash detection or user input to a USB flash drive.
Power Source	The Raspberry Pi will be powered by a battery bank mounted on the bicycle.

8. Software Development and Engineering Tools and Technology

The following are environments, tools, and technologies that team members will be using for the creation, development, and debugging of the system.

Table 8.1: Software Technologies

Software Technology	Use Cases
Python	Python will be the main programming language used.
ffmpeg	ffmpeg will be the primary library leveraged for interfacing with the front-facing camera.
NumPy	NumPy will be a Python library used to develop the graphs of the accelerometer data.
PyTest	PyTest will be used as the main testing tool for the software components.
Visual Studio Code (VSCode)	VSCode will be the preferred Python and Github IDE and will be utilized for all software development.
MobaXTerm	MobaXTerm will be used for SSH purposes for remote connections between computers and the Raspberry Pi.
GitHub	GitHub will be the version control and issue tracking platform utilized.
Markdown	Markdown will be the primary text-based language used to create documentation.
Filmora	Video editing software for creating videos for demonstrations.

Table 8.2: Hardware Technologies

Hardware Technology	Use Cases
Prusa MK3	Will be used to convert 3D models into gcode to be 3D printed.
Autodesk Inventor	Inventor will be utilized for creating 3D models for the physical aspects of Cyclops including the mount and housing.
Fritzing	Fritzing will be used to design the breadboarding electrical schematic.
Bike	Tester bike will be used to simulate and test the functionalities of Cyclops in a real time environment.
Webcam	A webcam will be used as a the front-facing camera.
ADXL345	A common accelerometer used to detect three-axis acceleration.
RCmall TF-Luna LiDAR	An all-condition distance sensor for detecting the distance of a vehicle behind the bicycle.
Adafruit Neopixel Strip	An array of LEDs that will interface with the LiDAR sensor to display distance information.
Buttons	Used for user input for the system.
Machining Tools	A variety of machining tools will be used in order to create the electro-mechanical system. For example, power drills, screwdrivers, soldering irons, etc.