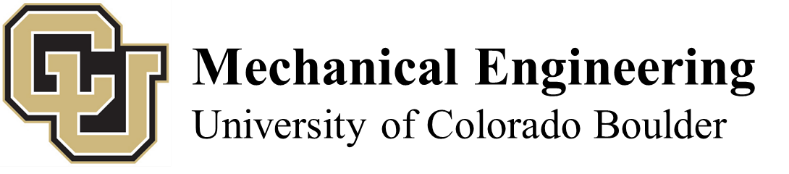
**– TANK BOT –**

**PROJECT REPORT**

**BY: NEBIYU TADESSE**

**MCEN 4115: Mechatronics**

Date: 05/04/2020



**Table of Contents**

Introduction and Project Requirements [3]

Code [4]

1. Flow Diagram [4]
2. Libraries Explanation [4]
3. Systems Integration [4]

Results [6]

1. Progress and Results [4]
2. Future Work [4]
3. Advice for future kits users [4]

Citation [7]

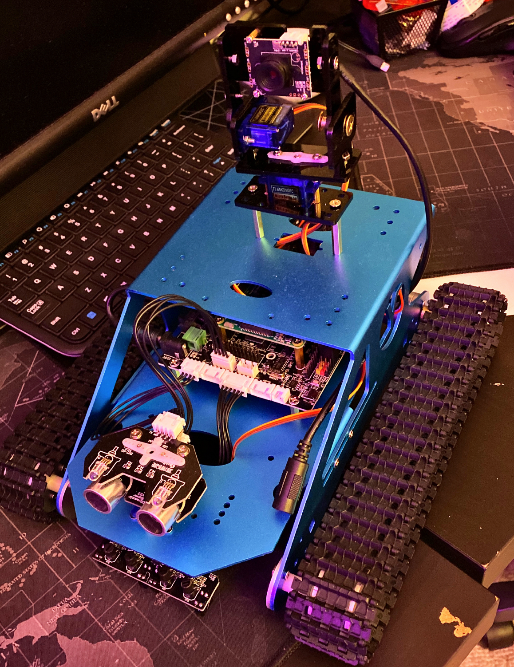
Appendices [17]

1. Code [4]
2. Other technical support [4]
3. **Introduction and Project Requirements**

The original project for this course was to build and program a Police Academy Robot that can travel through a custom-made map and strike down a simple enemy target, by shooting it down with NERF foam bullets or by other means that fit the scenario and were safe. However, considering the recent pandemic, and school shutting down, the project was changed such that it was more feasible for student to work from home.

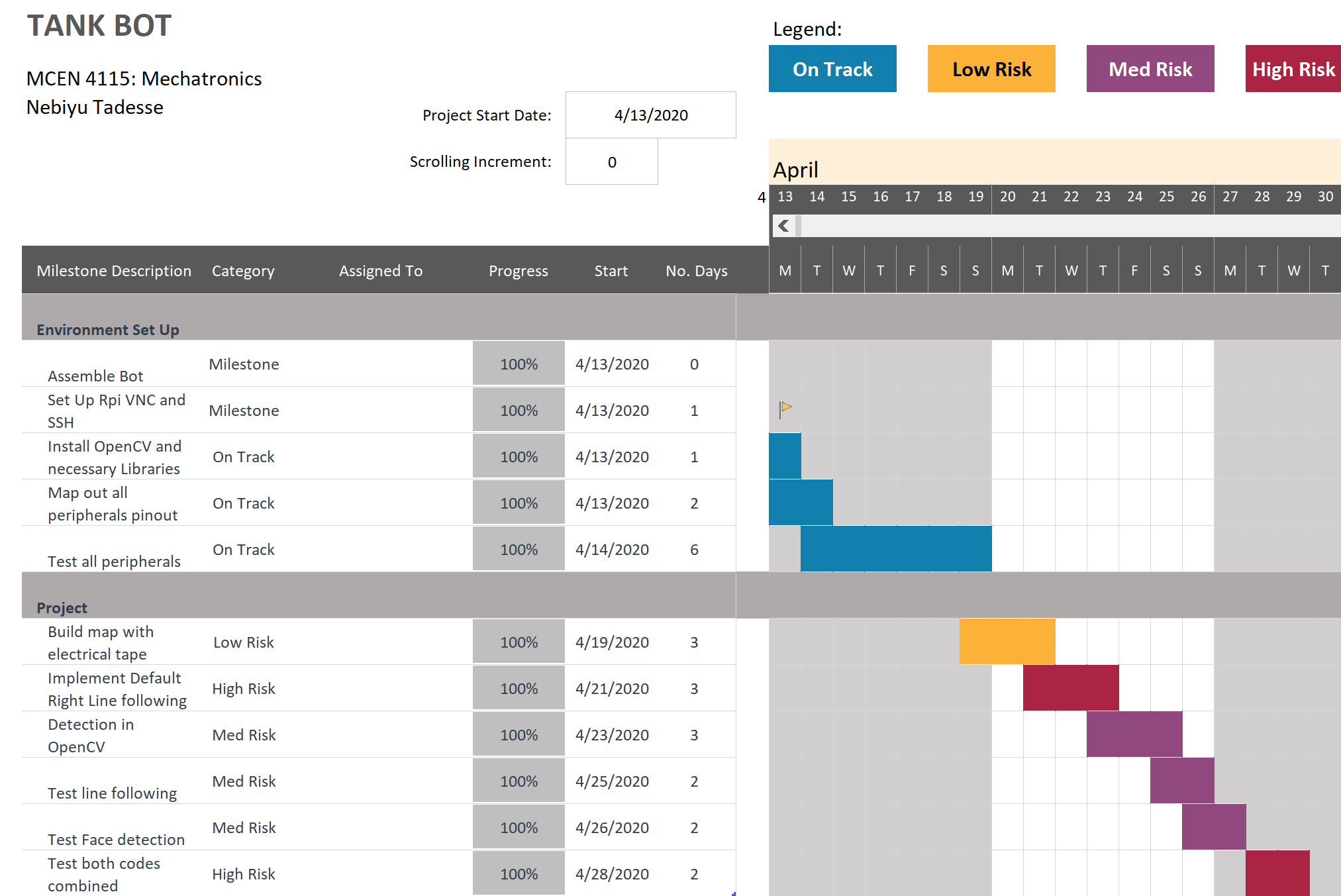
I opted in for the individual robotics kit project where the professor bough us a robot kit from online, and we tried to do closely approximate the goal of the original project but with the hardware at hand. The requirements of this project, as presented by the professor, go as follow:

1. Lay out a course with an intersection or two using tape on the floor or cardboards to make walls
2. Get you robots to navigate through your course, using vision or sensors (not timed or hardcoded)
3. Find an object you think your robot can easily identify. Place it at the end of the course.
4. Put a lance (stick/ ruler/ extension) on the front of your robot and have your robot drive up to the object and knock it down with the lance.

****The robot that I got was the Raspberry Pi Tank Smart Robot Kit by Yahboom, **Fig.1,** and the first step of this project was to assemble to robot since it came disassembled, **Fig.2**, then make a timeline, **Fig.3**, in order to complete the project on time.

**Figure 2**: Assembled Robot Kit

**Figure 1**: Robot Kit from Yahboom

****

**Figure 3**: Project Timeline

1. **Code**

The coding for this project was done in python and is broken into 3 separate files, which are:

1. Module that includes functions for all low-level interfaces
2. Module that includes functions for all high-level Algorithms and main function
3. Module that sets up all functions and uses them as they were intended to be used to allow easily running tests on all modules and functions created.
   1. **Flow Diagram**

The flow diagram of the project is quite straight forward,

All it does, is follow the line it is on until it arrives at an intersection. Once it is at the intersection, the robot uses its camera to check all three directions, forward, left, and right to see if the enemy can be seen. If the enemy is seen in a certain direction, the robot turns toward that direction till it knocks it down. However, if the enemy is not seen, it will always try to default to a right turn, if not possible, go forward, and if that is not possible either, it will take a left turn. If we reach a dead end, the robot will just turn and continue its quest to eliminate the enemy.

* 1. **Libraries Explanation**

The external libraries used for this project are:

1. Open-CV – which is “a library of programming functions mainly aimed at real-time computer vision” [1]
2. Pigpio – which is “a library for the Raspberry Pi which allows control of the General Purpose Input Outputs (GPIO)”

Open-CV was chosen because it is open source and has functions that allow to easily implement computer vision into the project. It also has a big user community behind it.

I chose Pigpio, pronounced as pi gpio and not pig pio, because of strong focus on meeting timing requirements for PWM pulses and others which is very useful for the servos and ultrasonic sensor. More can be found online for both.

* 1. **Modules Explanation**
     1. **Robot\_control.py**

The robotics

* + 1. **Camera\_test.py**
    2. **Ultra\_test.py**
    3. **Main.py**
    4. **Main\_functions\_test.py**

* 1. **Systems Integration**

1. **Results**
   1. **Progress and Result**
   2. **Future Work**
   3. **Advice for Future Kit Users**
2. **Citations**
3. **Appendices**
   1. **Code**
   2. **Other Technical Support**