06.12.2018-13.12.2018



**MIDDLE EAST TECHNICAL UNIVERSITY**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

EE493 – Weekly Progress Report #9

POTATO INTEGRATED TECHNOLOGIES

A close up of a clock

Description generated with high confidence

**Design Studio Coordinator:** Arzu KOÇ

**Partners and Contact Information:**

Fatma Nur ARABACI 2030047 0533 602 75 27 arabaci.fn@gmail.com

İrem COŞKUN 2030419 0539 414 59 69 iremcoskun0@gmail.com

Aycan BEYENİR 2030328 0539 330 05 70 abeyenir@gmail.com

Berkay GÖKSU 2030757 0505 635 81 45 berkaygoksu1@gmail.com

Furkan Bahadır ELİK 2030518 0554 963 94 10 elik.bahadir@metu.edu.tr

**What has been done:**

In this week we put our full concentration on the preparations for demo. In this manner, as a PiTech team we concentrated onto the four main branches. Firstly, we worked on the optimization of the data transmission system. Secondly, we created a test procedure for our robot. Thirdly, we worked on the image processing unit. At last, we worked on the development of the shooting subsystem. In the following part we provided the detailed explanation of the development process.

**Optimization of the Data Transmission:**

On optimization issues on transmission delay of commands due to transmitter program running on Raspberry Pi, we worked on solving the delay problem. We had about 1 seconds of delay before which was unacceptable for the project. To solve the issue, first we debugged the program and find which part is causing such a delay. It was important to determine that the bottleneck is whether our code or the Raspberry Pi’s processor speed. Because if it was the RPi’s speed, we would have a serious problem. However, luckily, we were able to identify the problem and solve it. The problem was a non-optimized piece of the algorithm that we were implemented. After making the necessary changes, now, we are able to send the commands at instant. We tested the new code and saw that the LEDs blink at the instant we press the buttons.

**Test Procedure:**

During the Demonstration we will demonstrate some steps for Communication, Telecontroller and Detection subsystems. This document covers these steps.

In order to transfer commands, communication and telecontroller subsystems will be tested. Test procedure for command transfer is as follows:

1. Different sine waves created in C++ will be transmitted from our transmitter Antenna to our receiver radio or module.
2. Different frequencies should be distinguishable with human ear from different distances or audio wave should be observed with Oscilloscope.
3. For different user inputs from buttons different sine waves will be heard from radio’s headphone or audio wave should be observed.
4. For different user inputs we should observe different outputs from LEDs which corresponds to different commands.
5. Step 5 will be tested for different distances at DEMO place.

For detection and transferring the image data we are still considering more than one option and working on both of the options. So, just in case of any unprecedented problem we decided to write test procedure for both plans. The following two test procedures are for different problem-solving methods.

In order to detect image information and send it to the user, detection and communication subsystems will be tested. Test procedure for image processing method is as follows:

1. Existence of the ball will be understood by the Raspberry Pi.
2. Location data as left or right will be understood, and corresponding LED will light up.
3. Location data will be transferred as a sine wave to user radio or receiver module. Different sine waves should be heard from radio with human ear or audio wave should be observed.
4. The data will be transferred using the same method as stated in the previous test procedure. The steps 3-5 will be implemented for different ball positions rather than different user inputs.

In order to detect image information and send it to the user, detection and communication subsystems will be tested. Test procedure for drone kit option is as follows:

1. The connections will be checked.
2. Video transmission will be tested for 0-meter distance.
3. Video transmission will be tested for 10-meter distance.
4. Video transmission will be tested for 30-meter distance.

**Image Processing Unit:**

In this week, in the development of the image processing algorithm we are behind the schedule. Due to the unforeseen events and problems, our process is slowed down. On the other hand, with the completion of the principal structure of communication subsystem as a whole team we can put our full concentration on this unit.

The main hardship we faced in this unit is, taking a picture and uploading it to C++. If we achieve to take an image on C++, we can complete this unit more easily.

**Development of Shooting Subsystem:**

In this week we started to learn required drawing programs for the design of shooting subsystem. The program which is called “Rhinoceros” is used for 3d modeling of the shooting subsystem and “Keyshot” is used to get the end product image of the system.

According to our experiences, there should be two angular sticks that are parallel to ground for holding the ball closer to rotational shooting part.

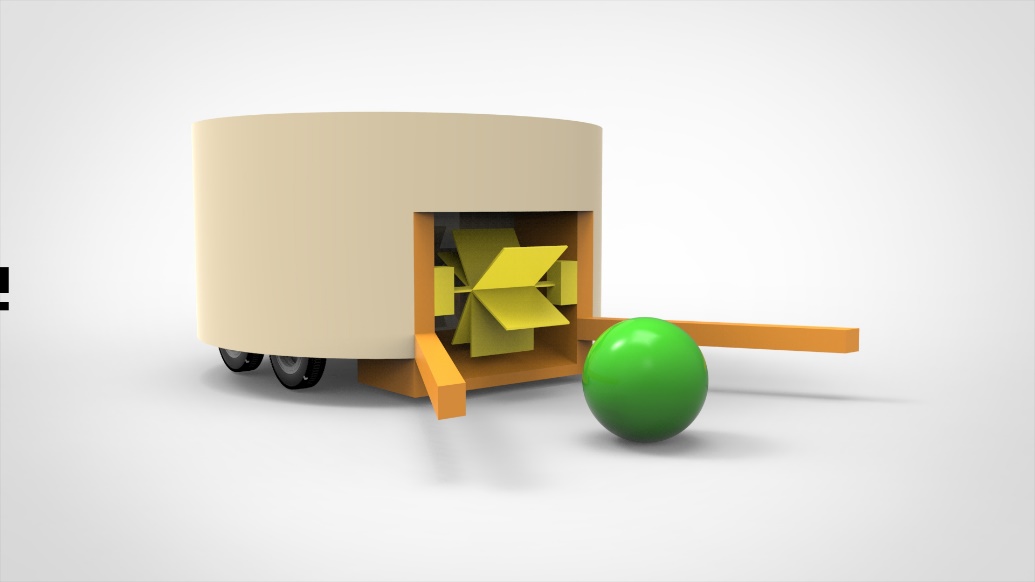


Figure 1: Shooting Subsystem Design

In the figure 1, the orange ones are represented with these sticks. The radius of the ball is 4 cm and it gives reference for the rest of the model. The length of the sticks are 25 cm and the radius of larger cylindrical part is around 20 cm. The yellow parts represent the rotational shooting system.

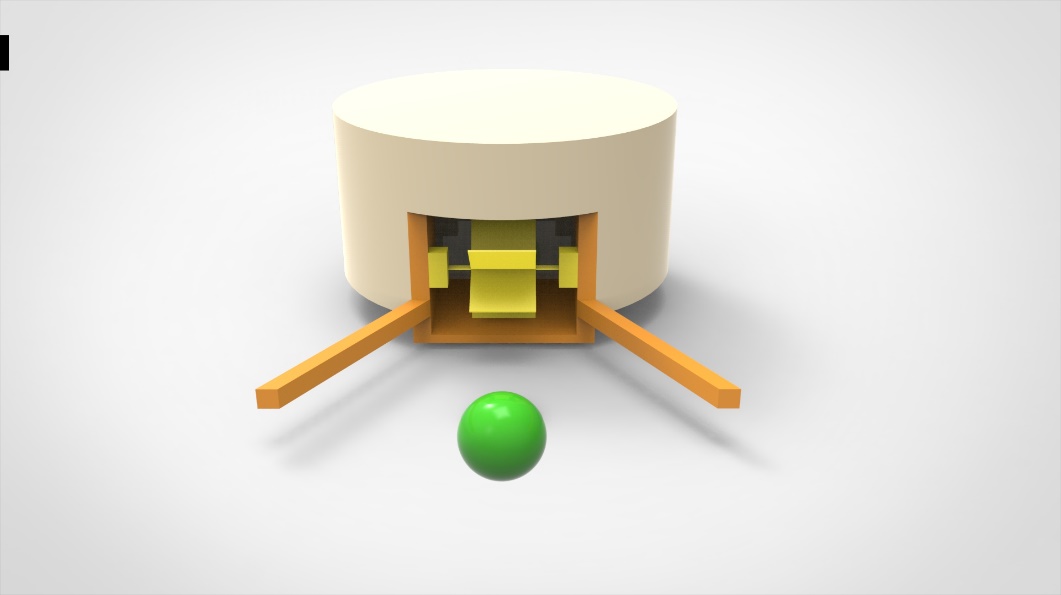


Figure 2: Shooting Subsystem Design (front view)

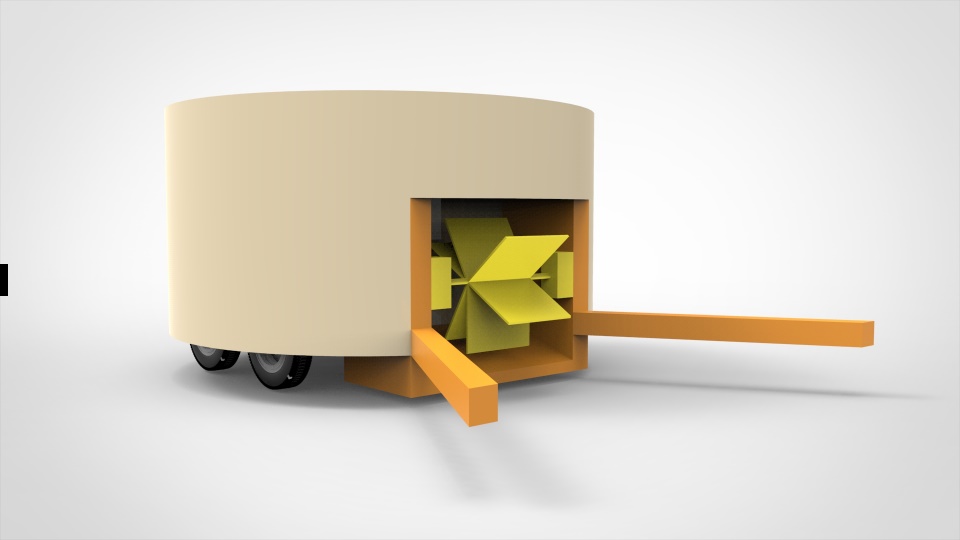


Figure 3: Shooting Subsystem Design (Alternative 1)

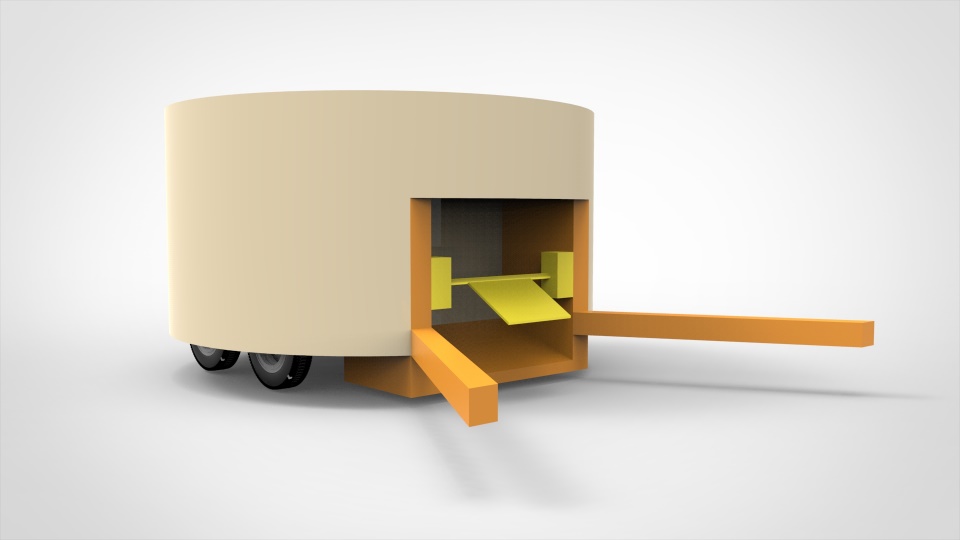


Figure 4: Shooting Subsystem Design (Alternative 2)

As a first alternative of rotational shooting system, there can be one piece and it shoots the ball by rotating around itself, or the second alternative is multiple pieces connected to a single part. One of them can be chosen in later stages.

**Frequency Counting via Arduino Uno:**

We tried measuring the frequency of a given squared wave input via Tiva board last week but after some research we found that we may also use Arduino for the same purpose. The code we executed was also for square wave. However, we first tried it with sine wave with some offset. For different frequency levels and for different offset values we read the value calculated by the code and keep the data to calculate the max error percentage which is the deviation from the actual frequency value. The results are as in Figure 5:

Figure 5: Error Rate (%) Data of the Code for Different Offset Values

After figuring out the working principle of the code we finalized our frequency counter design with 2Vpp and 2V offset. By achieving this we optimized our couniting module since Arduino will also be used for other purposes such as driving the motors with the help of driver shields.

**Next week’s plan**:

Since next week is the demo week, in this week we are going to work on the unfinished parts and in the meantime prepare a demo setup for demo session. Also, if we found a chance, we will try to optimize our system for better response time.