29.11.2018-06.12.2018



**MIDDLE EAST TECHNICAL UNIVERSITY**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

EE493 – Weekly Progress Report #8

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 started to put our full concentration on the preparations for demo. In this manner, as a PiTech team we concentrated onto three area main branches. The first and the most important branch for the demo is the development of the communication and telecontroller block. The second branch is image processing block. Moreover, we not only work on the demo but also, we continue to the research and development of the other submodules. In this week we worked on motion subsystem and power subsystem. In the following part we provided the detailed explanation of the development process.

**Communication and Telecontroller Block**

This week, we were supposed to implement the band bass filters that were designed previous week, however, as a company, we decided to go with Tiva (Texas instrument microprocessor evaluation board) or an Arduino to count the frequencies of the input signal due to the fact that the analog filters will be too sensitive to environmental effects and implementing them correctly will be quite challenging. Therefore, this week, we worked on sending different pulses with different inputs over FM channel that I was implementing. First, we have managed to use Raspberry Pi’ GPIO pins as an input buttons using WiringPi library and depending on the states, I have managed to send specific sine waves with different frequencies. In other words, we did manage to send commands to wireless location. Further optimization will be done in the transmitter program in the following week. On the other hand, Bahadır made a better antenna (it was a simple jumper wire before) using a think satellite receiver coaxial cable to achieve even better range and noise performance. The new antenna can be seen in Figure 1. Testing on the new antenna, we as a team observed that the signal power is way above the level it was before. We believe that with this antenna design, we won’t need an amplifier to boost the signal. However, the antenna can be further improved in the following weeks.

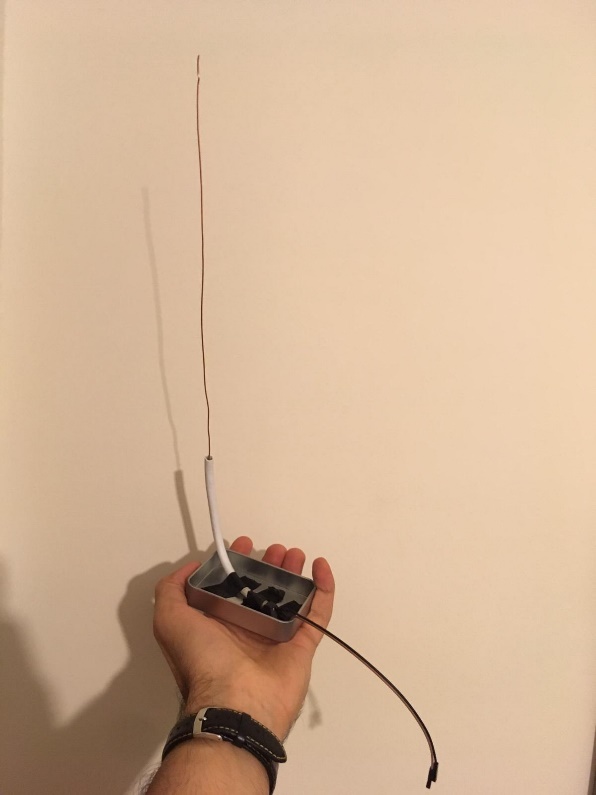


Figure 1: The antenna

For the receiver part of the telecommunication system, frequency measuring code implemented on Assembly. We do not believe Assembly is the most efficient language for this task but since most of our members have a better control on this language we decided to give this a chance. During our tests TIVA TM4C123GH6PM microcontroller used. This option is actually cheaper because TIVA board is $12.99 while a Raspberry Pi3 is more than $30.00. During the tests we worked with two different frequency signal which are 10kHz and 20kHz. These frequencies represent different commands like left and right. Transmission of this signals was already has been made so, we worked with signal generator. We managed to give different outputs for different frequency levels. This means that we have a fully functional system to identify the different commands and give different outputs respectively. These outputs are going to be connected to Motors later. We are waiting for our shipments to arrive in order to try the whole communication system. Figure 2 below shows the Assembly implementation of commands for different frequencies.

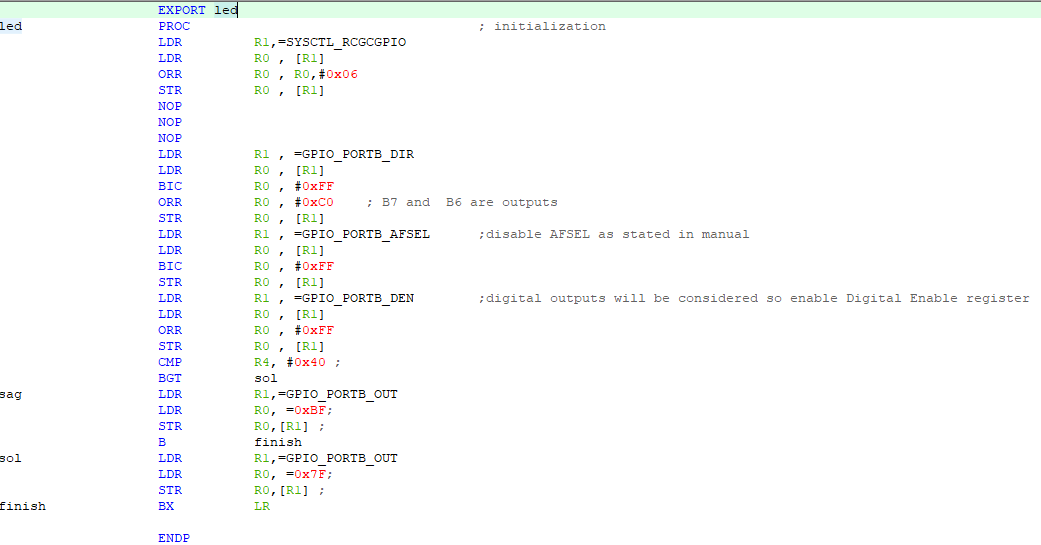


Figure 2: Assembly implementation for identifying different inputs and give outputs respectively

**Image Processing Block**

Due to the plan, this week we are supposed to develop the logical basis of the for a ball detection algorithm. As stated in the previous weekly report, we already developed a color detection algorithm, which omits the undesired colors from the original image and creates a reduced version of it. Starting from this code, in this week we took a step further on the development of the image processing algorithm and successfully implemented a ball detection algorithm.

Due to the practical reasons the test demo, which is provided below, is implemented on the MATLAB. In the first step we worked on two sample images shown in Figure 3. As observed in these images there are two balls in the provided images, which colored red and green respectively. The main aim of the developed algorithm is deciding is there any green ball. If no green ball present, red screen is shown. If ball is present the original image is shown. The results of the given images are provided in Figure 4. Moreover, when the ball is present it also finds the approximate center of this ball. The closeup for marker is shown in Figure 5.

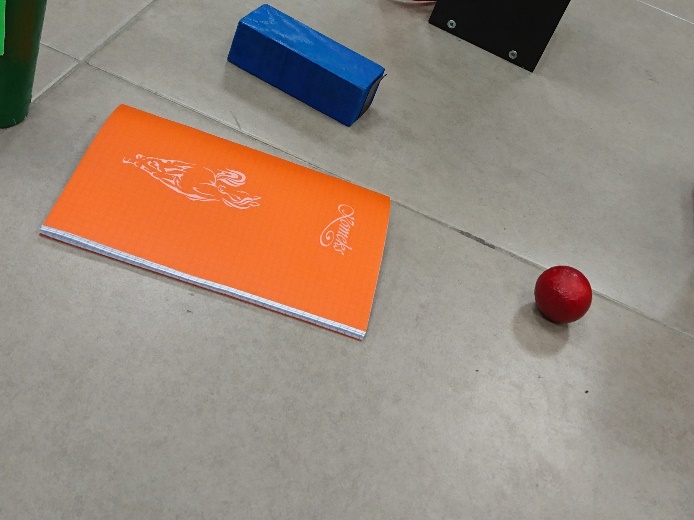


Figure 3: Sample images with red(left) and green(right) ball

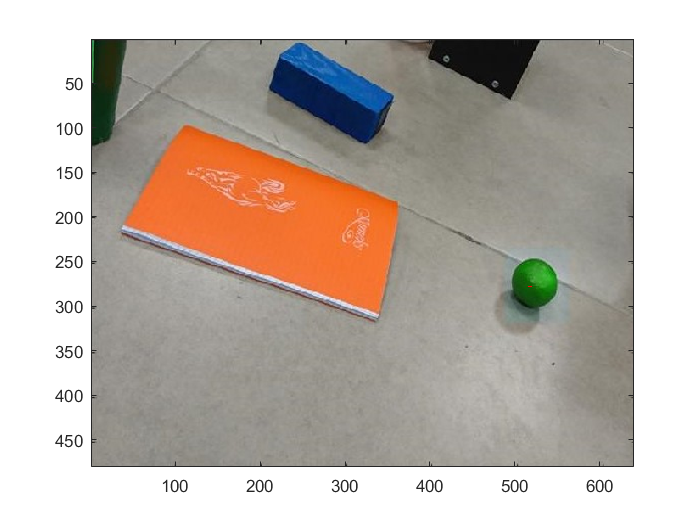
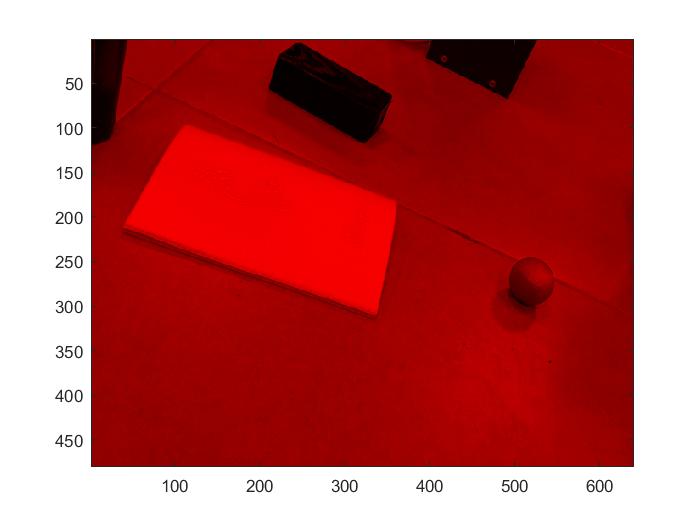


Figure 4: The resultant image of algorithm with sample images with red(left) and green(right) ball



Figure 5: The closeup image of marker indicating the approximate center of ball

**Motion Submodule**

Further progress is achieved in motion submodule one of which is wheel research. There are different variety of wheels in the market such as **standard, orientable, ball, and omnidirectional wheels.** Even though the wheel configuration of our robot is not decided yet, most likely we will use different combinations of the following wheels:

1. **Standard Wheel**

Most common wheels used in robotics is standard wheels. This type of wheels is characterized by two degrees of freedom and can run in front and back.



*Figure 6: Basic Toy Wheel [1]*

Most likely our primary tire will be standard one. However, this on itself may not be enough to accurately move our robot.

1. **Orientable Wheel**

Based on two types including center and off-centered oriented, orientable wheels are standard wheels mounted to a fixed or omni-directional fork designed to catch the wheel.



*Figure 7: Caster Wheel [1]*

This the type we think will be used the least likely. Since these tires do not have direct contact to the shaft of the motor. However, it may be used as a secondary type of tire used for balancing.

1. **Ball Wheel**

Based on a ball, these wheels have a total freedom of 360 degrees and like orientable wheels are used to balance a robot.



*Figure 8: Pololu Ball Caster with 1″ Plastic Ball [1]*

This type of wheel can be used if we need to stabilize the motion of our robot.

1. **Omnidirectional Wheel**

Small wheels are attached on a large wheel to build a multi-directional system designed for robots for movement in any direction.



*Figure 9: Multi-direction Roller [1]*

Considering the movement capabilities of our future robot using Multi-direction rollers can be challenging for us since controlling these wheels are quite different than the standard wheels. However, if we can manage to control these it will be much easier to move our robot to the desired destination. Another drawback is that Multi-directional wheels are more expensive than the regular ones.

The most vital component of the motion subsystem is the motors. Considering the motors in the market in order to drive the robot and shoot the ball via shooting subsystem possible motors are as below:

1. **Servo Motors**

Using servo motors for driving the robots seems unreasonable since the working principle of servo motors are rotating the shaft to a desired angle with respect to its input duty cycle. However, for shooting subsystem it is more likely to be used. On the other hand, for shooting subsystem we may need some torque to send the ball further. Servo motors with higher torque can be heavier than the small ones we used in our circuit laboratories therefore we need to consider this drawback. Another drawback is that even though controlling as servo is easy in theory for our specific situation it may be harder to hit the ball to the desired location with servo motors. Suitable shooter selection is highly important at this stage if we decide to use servo motors.

1. **Stepper Motors**

Since steppers move in precise repeatable steps, they excel in applications requiring precise positioning such as 3D printers. However, we don’t need precise positioning in motion subsystem. It will be good for shooting subsystem though. Normal DC motors don't have very much torque at low speeds. A stepper motor has maximum torque at low speeds, so they are a good choice for applications requiring low speed with high precision. This feature is also not applicable for motion subsystem. However, obtaining high torque with low speed can be used for shooting subsystem. One drawback of stepper motors is that unlike servo motors, most steppers do not have integral feedback for position.

1. **DC Motors**

DC motors in the market are the ones that we will most likely use. We can modify these with respect to our torque and speed regulations. These are also easy to control with a simple encoder integration. Other than reading how many turns have been made by the motors we may also consider using an electronic speed controller to control how fast our motor turns as well. For brushed motors ESC simply turns the voltage on and off to vary the speed of the motor. This is suitable design for driving the robot. PWM is used as ESC input which can be easily produced using an Arduino. For shooting the ball an encoder will do the trick is we decide use DC motors for both of the system. However, we must also consider our budget since we will need an ESC for each robot in this scenario.

**Power Subsystem**

This week, we focused on LiPo batteries for battery selection. The reason of focusing on the LiPO battery is, it is lighter than other possible battery types, and its efficiency was also higher. The main work of this week about power supply was the use of the battery rather than the choice of battery. We've researched the chargers to charge the battery.

In addition, we understand that we may be able to change the input and output voltages when using the battery. That's why we decided to use the switching converter.

Switching converters (such as buck converters) provide much greater [power efficiency](https://en.wikipedia.org/wiki/Power_efficiency) as DC-to-DC converters than [linear regulators](https://en.wikipedia.org/wiki/Linear_regulator), which are simpler circuits that lower voltages by dissipating power as heat, but do not step up output current. For this reason, more detailed research has been done on buck converter usage. A buck converter (step-down converter) can be seen in Figure 10, is a [DC-to-DC power converter](https://en.wikipedia.org/wiki/DC-to-DC_converter) which steps down voltage (while stepping up current) from its input (supply) to its output (load).

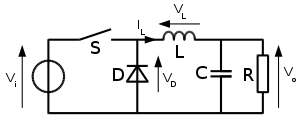


Figure 10: Buck Converter Schematic [2]

**Next Week:**

* We will try to blink a LED at a remote location using complete communication link that is Raspberry pi, FM radio and Tiva or Arduino.
* Arduino for frequency measuring also going to be tested for plan B.
* Complete command test on actual DEMO place (E and D buildings) will be tested.
* More command options will be added to both transmitter and receiver systems for having better control over our robot.
* Optimization on already working systems will be made.
* Also, for the image transfer task, we will try a new drone kit that we had ordered as a plan B that we are waiting it to be shipped.
* The ball detection algorithm is going to be implemented on C++.
* Further research on batteries will continue without being highest priority.
* In the demo, we will try to determine the voltage and current that the battery should provide.

**References**

## [1] Overview of Wheels Used in Robotics, Retrieved from

## https://www.intorobotics.com/overview-of-wheels-used-in-robotics/ [2] Buck Converter Schematic, Retrieved from

## https://en.wikipedia.org/wiki/Buck\_converter#cite\_note-2