Auto Sports

March 1st, 2016

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440W Design Specification for the SmartPitcher

Dear Dr. Rawicz:

Attached is the Design Specification document for the SmartPitcher with technical specification which outlines the full design procedure of the SmartPitcher. We are designing and implementing a smart pitching machine that can automatically fire the ball to the athlete's exact location. The project's purpose is to increase the practice output of an athlete and to introduce a more efficient and convenient method of practicing.

The following document provides a proof of concept model of the SmartPitcher. It also provides an overview of both the hardware and software procedures necessary to implement a functional prototype of the SmartPitcher. The technologies described in this document are intended for development of a proof-of-concept prototype. In order to market our product, additional design work will be required to meet the market standards of a competent and cost effective design.

If the reader has any questions or comments, please feel free to contact us via email at youjungk@sfu.ca.

Sincerely,

Youjung Kim

Enclosure: Design Specification for SmartPitcher

Design Specification for the SmartPitcher

Project team: Youjung Kim
Sion Park
Joel Kim
Jeff Yoo
Fereshteh Sana

Contact Person: Youjung Kim
youjungk@sfu.ca

Submitted to: Dr. Andrew Rawicz
Steve Whitmore
School of Engineering Science
Simon Fraser University

Issue Date: March 1st, 2016

EXECUTIVE SUMMARY

The design specification for the SmartPitcher by Auto Sports delivers a set of explicit information of the design and detailed technical descriptions of the proof-of-concept, demonstrating its feasibility and principles of the concept in different aspects. The purpose of this documentation is to provide a clear presentation of the design decisions in details for the reader in terms of software and hardware mechanisms along with functionality and usability consideration. This documentation will be referenced in the future as an instruction tool or inspection guide for readers that can be used to an external.

The SmartPitcher is a ball pitching machine that has devised a motion and distance tracking system, combining with ball-loading mechanism for a user to receive the ball whenever and wherever by simply pushing a button of a peripheral device that is attached on an armband. Our unique approach utilizes the efficient exercise training without the need of an extra labor, passing a ball from one player to another player. The main hardware and software of SmartPitcher design consists of three main components:

- Motor Speed Control
- Motion Tracking System
- Automatic Ball Loader

This document describes the thorough knowledge of the listed subjects with specification and justification. Additionally, this design specification is mainly focused on the technical side of hardware, software, and firmware design along with the compatibility and sustainability of our chosen parts. Each aspect of the design section is provided with the technical data with characteristics of the feature.

The remainder of the document presents the test plans to deliver a set of test procedures, ensuring the development to meet our requirements and safety.

Table of Contents

List of Figures and Tables

Figures

Tables

Glossary

API	Application Programming Interface
DC	Direct Current
USB	Universal Serial Bus: A cable for connection, communication, and power supply between computers and external device.
IDE	Integrated Development Environment
Controller	Arduino Microcontroller, Bluetooth transceiver, battery, power supply, LED, LCD, motor shield, and its enclosure.
Mobile Application	Software application called, "Bluetooth spp pro," supported by Android: A simple 2-way Bluetooth-based Android Controller for Arduino UNO.
Linear Gear	Stationary gear, powered by Servo Motor, and linear gear racks use rotation to transmit torque to push the ball to the spinning wheel.
Arduino LCD	Liquid-Crystal-Display: produces image and desired information regarding to the motor speed and the distant measurement.
Motor Driver	Type of current amplifier to take a low current control signal to produce higher-current signal in order to drive a motor.
Ultrasonic	A device emits high frequency sound wave that measures the distance between the device and an object.
Bluetooth	Short range wireless (connectionless) interconnection of a mobile device to the receiver mounted on microprocessor to receive a signal.
Loading	A design part where the ball and the linear gear encounter.
Camera	Webcam that is compatible to a laptop and Roborealm.
Wheel	Stationary 6 inch diameter of two spinning wheels engulfs a ball to shoot.

RPM	Revolutions per minute, unit of motor’s rotational speed
g∗cm	Gram centimeter, unit of torque
I/O	Input/Output
PWM	Pulse Width Modulation meaning that digital or analog data is encoded through the use of square wave electrical signals, where the width of the square wave determines the value of the data.
SRAM	Static Random Access Memory
Flash	Flash memory
CPU	Central Processing Unit
C/C++	Type of object-oriented programming language
GCC C/C++ Compiler	GNU Compiler Collection, a compiler which supports wide variety of programming languages, including C/C++
IDE	Interactive Development Environment
SM	Servo Motor
RR	Roborealm
AU	Arduino Uno Microcontroller
Rack and pinion Gears	Assembly of gears found in automotive steering system
LPF	Low Pass Filter

1. Introduction

The SmartPitcher by Auto Sports is an innovative pitching machine that will track an athlete's movement and provide feedback by firing the ball to the exact location. The first-perspective operating SmartPitcher is designed to bring easy, fast, reliable, and safe access for athletes to have convenient but expeditious exercise. The firmly affixed mobile device on a user's armband will interface with Bluetooth transmitter which will release a ball to the accelerated wheel, delivering the ball to where the user is standing. More specifically, the machine consists of a ball loader which is triggered by a Bluetooth signal, Roborealm which is a software that is used to track motion and color, and wheel speed control which is controlled by a microcontroller with a motor shield. The design specifications for the SmartPitcher's components are described in further detail below. Additionally, this document shall cover all the technical details for each component and provide overview of high level design of the system for the proof-of-concept prototype.

1.1 Scope

The scope of this document is to fully describe the design specifications and the features of the SmartPitcher that must be satisfied by the Auto Sports team. More specifically, this document describes and outlines design specifications for the motor, the microcontroller that controls the wheels' speed, the bluetooth communication between existing mobile application and the microcontroller, the motion tracking software, and the linear gear. The listed design specifications will describe the functionalities of the prototype and will outline potential modifications that must be considered before manufacturing. Furthermore, this document will provide appropriate test plans for all aspects including: software, firmware, and hardware.

1.2 Intended Audience

The design specification is intended to be used by all Auto Sports team members, as well as potential stakeholders. The specifications in this document shall be used to understand the implementation and design of the SmartPitcher. Also the testers are advised to refer to this document to ensure that the SmartPitcher is functioning in accordance with the design.

2. System Specification

The following figure displays the high level overview of the SmartPitcher. Once the SmartPitcher initiates its operation, the webcam will detect the user by applying color code identification feature from the computer vision/image analysis software, Roborealm, and the ultrasonic range finder will measure the distance between the machine and the user. Depending on the distance, the revolution speed of the pitching machine wheels also changes. Furthermore, the SmartPitcher allows the user to receive the balls which are being fired from the machine through the utilization of Smartphone application and Bluetooth transceiver communication system.

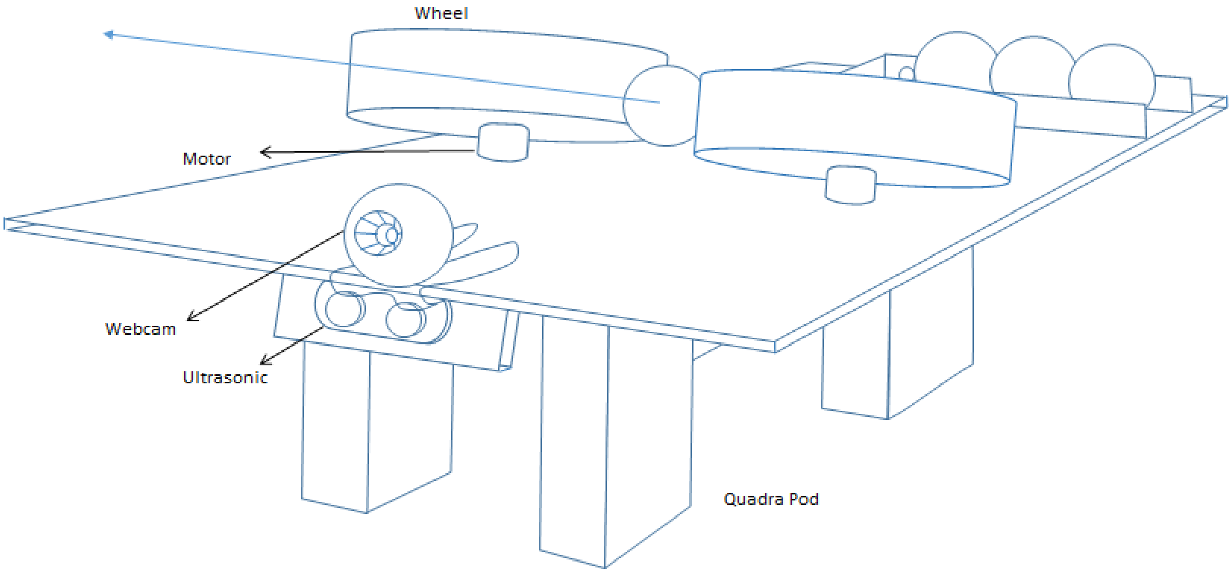


Figure 1: Smart pitcher front view

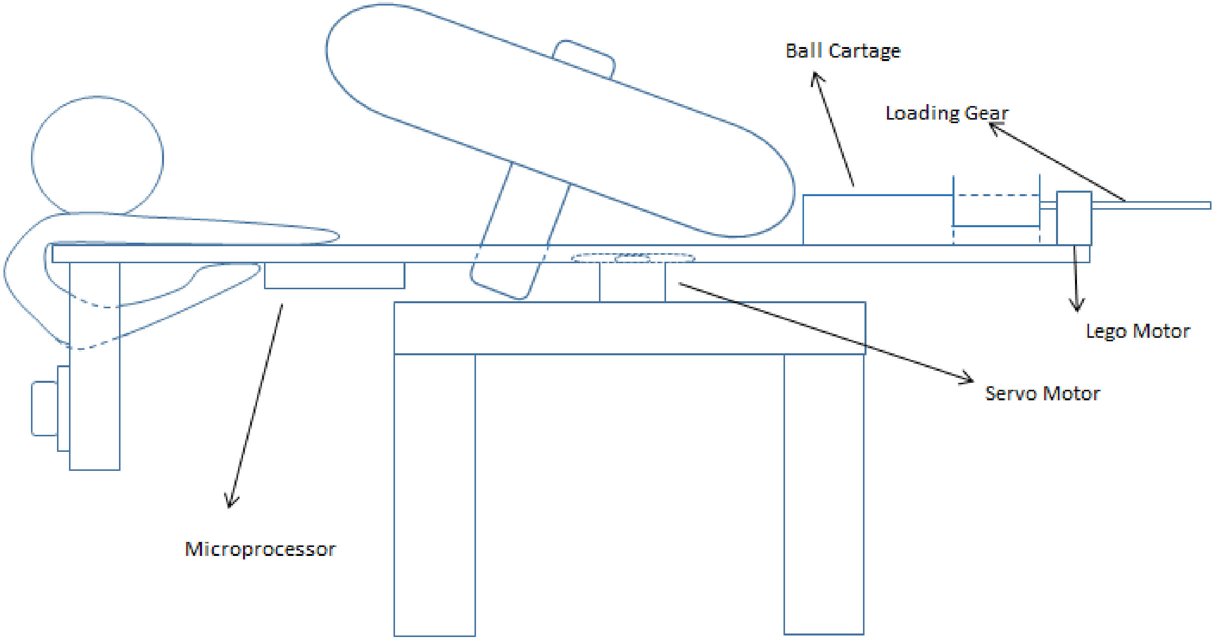


Figure 2: Smart pitcher side view

This section also details the general system specification for the SmartPitcher. The product, SmartPitcher, consists of three important mechanisms: Motor Speed Control, Motion Tracking, and automated Bluetooth ball loading system. First of all, Motor speed control mechanism consists of the Arduino Uno microcontroller, motor driver shield, two DC motors, and 12V rechargeable battery. The second significant mechanism is the motion tracking system which consists of the following components: webcam, servo motor, Arduino Uno Microcontroller, and AAA sized batteries as 6~7.8V power supply. Lastly, these components, circular/linear gears, Bluetooth Shield, Arduino Uno Microcontroller, and LEGO Stepper motor are going to be utilized for the complete design of automated ball loading mechanism.

The following figures display the high-level block diagrams for each significant mechanism of the SmartPitcher. Even though each mechanism utilizes Arduino Uno Microcontroller for the successful design/operation, only 2 microcontrollers are planning to be used, since there exist sufficient pins for the completion of Motion Tracking and automated Bluetooth ball loading system part.

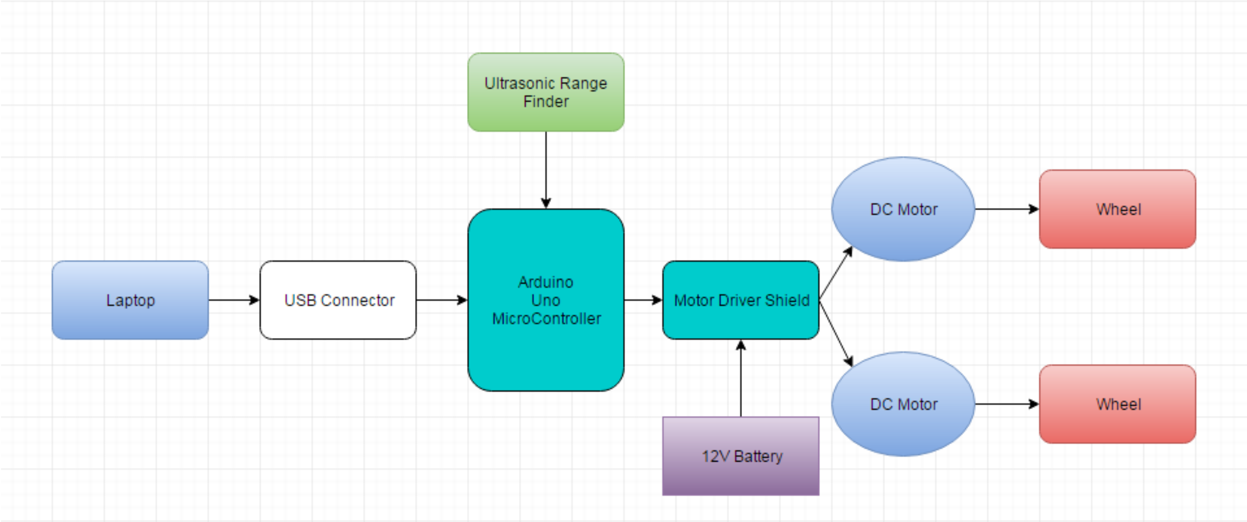


Figure 3: Block Diagram of Motor Speed Control system

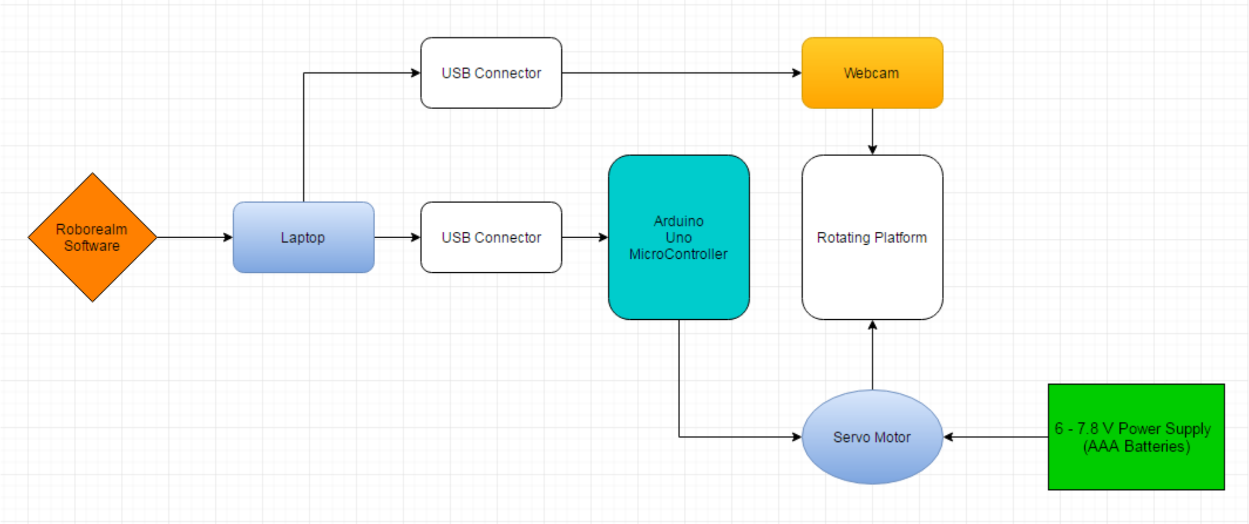


Figure 4: Block Diagram of Motion Tracking Mechanism

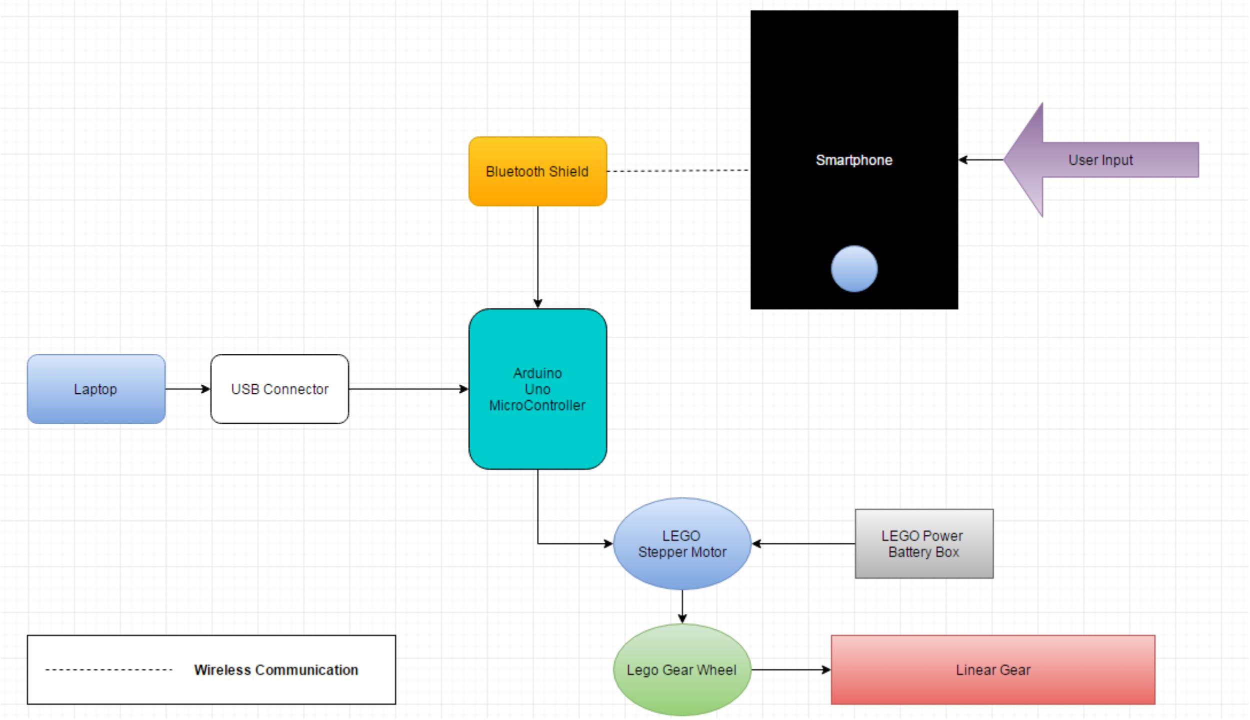


Figure 5: Block Diagram of Ball Loading Mechanism

As mentioned in Figure 5, the smartphone with the installed application, Bluetooth spp pro, will be used in order to provide efficient signal communication between the user and the SmartPitcher. The application assists the user to receive the ball when the user transmits the signal to the machine by tapping the smartphone.

3. Hardware Design

This section lists the general requirements applicable to the SmartPitcher.

3.1 Sensor + LCD Unit

The particular module, HC-SR04, is chosen as an ultrasonic range finder to give distance measurement to the Arduino Uno microcontroller. The module sends eight 40 kHz signal and detect whether there is a retrieving pulse signal. The range of this sensor is from 2 cm ~ 500 cm. Even though the range is not great, due to the budget and compatibility constraints, we decided that this sensor was the best fit to the design. The sensor module is wired directly to the Arduino microcontroller. The interconnection between the pins of the microcontroller and the sensor is listed in the table below.

Sensor Module		Microcontroller
VCC	↔	5V
GND	↔	GND
ECHO	↔	Pin 11
TRIG	↔	Pin 12

Table 1: Interconnection table for the sensor module

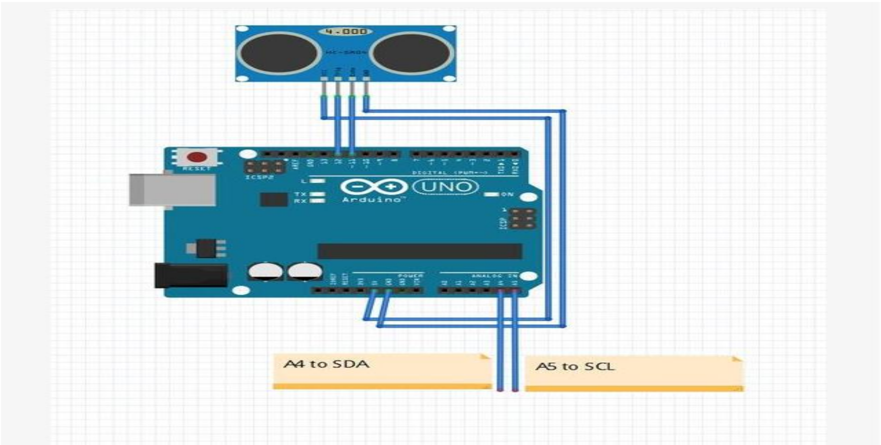


Figure 6: Basic wiring diagram for sensor, LCD, and microcontroller

In order to actively see the distance that it is measuring, we have connected LCD display to the microcontroller. In the figure above, the diagram shows the schematic of how LCD and sensor are connected to the microcontroller. The interconnection between LCD module and microcontroller is listed in the table below.

LCD Display	Microcontroller
VCC	5V
GND	GND
ECHO	A4
TRIG	A5

Table 2: Interconnection table for the display module

3.2 Motor Unit

Motor is one of the most crucial components for the full development of the SmartPitcher prototype, since 3 different types of motors are going to be operated in each significant mechanism of the pitching machine. While the DC motor will be essential for the completion of Motor Speed Control section, Servo and LEGO Stepper Motor will be applied in the development of Motion Tracking mechanism and automated Bluetooth ball loading system.

3.2.1 DC Motor Unit

DC motor unit was selected to accomplish the flexible movement and smooth rotation of the pitching machine wheel at a variety of rotational speed. The unit itself, as shown in Table 3 below is the Gear Motor 12V 500 RPM 25GA, which is a DC geared Motor and possesses unloaded current of 70 mA and max torque of 1000 g∗cm. Even though the motor, Motor – 12V DC, 160mA, was also selected as another option, DC geared motor has been determined as the ideal component due to large torque value and decent rotational speed. Table 3 displays the feature of 2 motors that our group have chosen for building the motor speed control system.

Motor Name	Gear Motor 12V 500 RPM 25GA	Motor – 12V DC, 160mA
Image		
Typical/Ideal Operating Voltage	12 V	12 V
Revolution Speed	500 rpm	11500 rpm
Current (Unloaded)	70 mA	180 mA
Torque	1000 g∗cm	62.5 g∗cm

Table 3: DC Geared Motor and DC Motor features

One of the most common features of every pitching machine is the rotating wheels. Although 12 V DC motor in the table above possesses high rotational speed, the machine will never pull out its full potential if the motor cannot support the mass of wheels that are applied and attached above. Thus, our team verified torque as the most important element and identified that the DC Geared motor is capable of holding more weight at a radial distance of 1 cm.

3.2.2 Servo Motor Unit

Servo Motor (SM) is a rotational motor that provides precise control of angular orientation including position, time, velocity and acceleration. The SM system can be implanted as a rotary actuator which is purely mechanical that produces a rotary motion or torque. Rotary actuator is an idea of angular motion that is produced by electrical power, either to rotate continuously or to a fixed angular position similarly to a stepper motor. Additionally, the linear actuator creates motion in a straight line to give a push motion. It applies to industrial machinery such as computer disk driver with valves and dampers. Figure 7 shows the SM model LS-8101F, which is assembled with a series of gears.

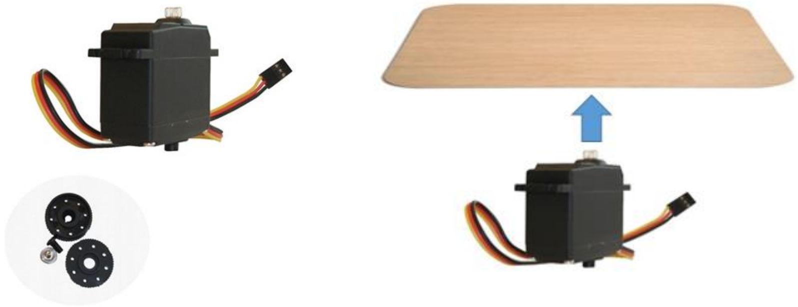


Figure 7: Servo Motor LS-8101F and Platform Design Preview

SM utilizes the horizontal rotation in the SmartPitcher, placing at the lowest part of the foundational platform to act as a neck. However, the size of the SM LS-8101F is relatively too small to maintain the balance of the platform that is shown in the Figure 8. Therefore, a quadra-pod is essential to stabilize and support the balance to avoid tilting shown in Figure 9. Consideration must also be given to the ground level, adjusting the length of each pod to receive a perfect horizon.

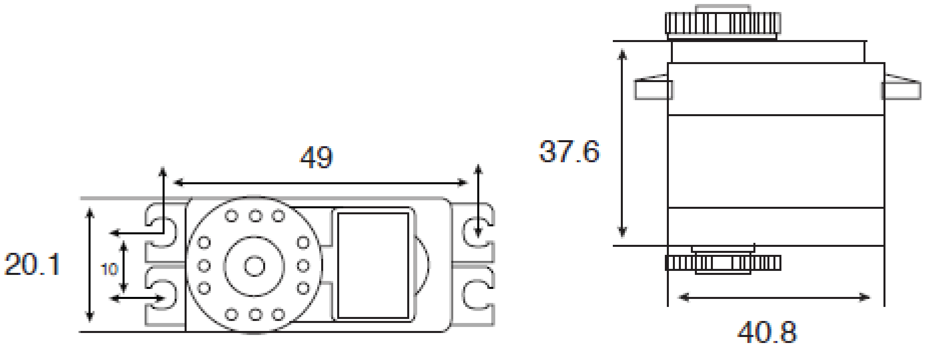


Figure 8: Servo Motor LS-81801F Mechanical Specification

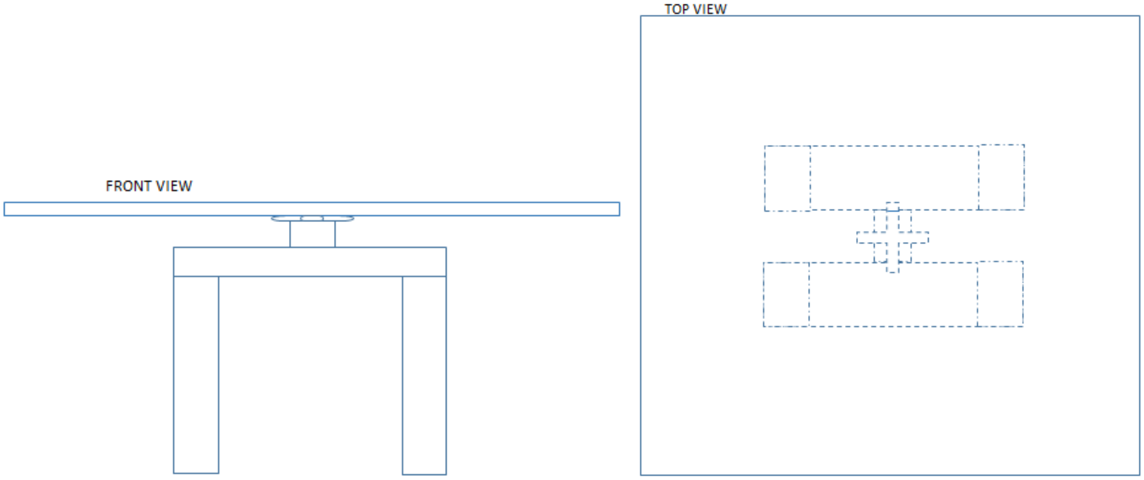


Figure 9: 2D Graphical Communication Design of Servo, Platform, and Quadra-pot

LS-8101F SM uses 5 volts - 7.2 volts to operate. Although the speed of the motor will be affected by the load on the servo, it can be powered by Arduino Uno through 5 volts pin since the board is provided a 5 volts supply on a single wire that is connected from a personal computer. The ideal voltage is 6 volts to operate the SmartPitcher which takes 0.96 seconds in one cycle with a stall torque value of 12 kilogram-centimeter. The higher the torque rotating, the heavier the load the servo can rotate. In other words, the higher current needs to draw to obtain higher torque. However, LS-8101F SM is digital servo that has an embedded microcontroller that provides additional functionality over traditional analog servos. Such features include faster response time, programmable speed, accurate positioning, and feedback. The trade off, however, is higher power consumption and a more complex interface. The following tables display the specification values of environmental condition and electrical/mechanical inspection.

Item	Specification
Overall Dimensions	40.8*20.1*37.6
Limit Angle	180°± 10°
Weight	56± 1 g
Connector Wire Length	300 ± 5 mm
Horn Type	Disc
Operating Temperature Range	10°C - 50°C
Operating Voltage Range	5V - 7.2V
Operating Speed (6V)	0.96 sec/360°
Running Current (6V)	400 mA
Stall torque (6V)	12 kg-cm

Table 4: Servo's Technical Data Sheet of Specification

The power is generated by a series of AAA batteries, and each of them contains nominal voltage 1.5 volts, connected to a mini breadboard. The positive 6 volts is connected to the red wire of SM and to the microprocessor pin Vin. The negative of the voltage source is connected to the black wire of SM and to the microprocessor pin GND. The yellow wire of SM connects to pin 9. The following table and figure show the interconnection of the microprocessor with the SM:

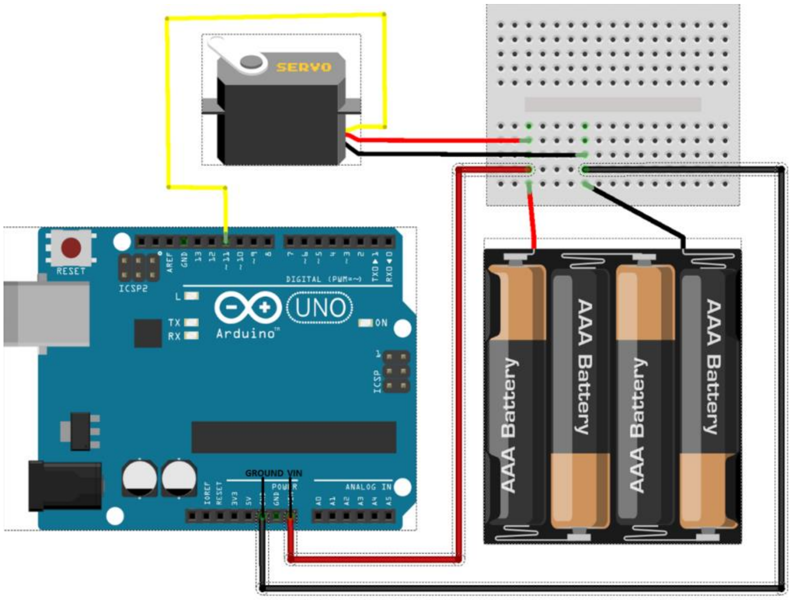


Figure 10: Circuit Design Connection between Servo with Arduino Uno Microprocessor

Terminals		Arduino Uno Pins
Positive Battery	↔	AU Pin [Vin]
Negative Battery	↔	AU Pin [GND]
Red wire SM	↔	AU Pin [Vin]
Black wire SM	↔	AU Pin [GND]
Yellow wire SM	↔	AU Pin [9]

Table 5: Interconnection between microprocessor to servo motor

3.2.3 LEGO Power Function XL-Motor Unit

The LEGO Power Function XL-Motor was chosen to be part of loading mechanism; it will be working with a linear gear like rack-and-pinion. This unit is shown in Figure 11 as well as its technical data in Figure 12 below. As it can be observed from the technical data, efficiency is not so great comparing with other metal stepper motors. However, this motor was chosen because of our gears. Our loading mechanism will be made of LEGO gears, and it will not fit in regular motors. Thus, LEGO Motor is used. Also, even with relatively low torque, it will not have a problem pushing a tennis ball. Finally, it fits well within our budget.

The LEGO Power Function XL-Motor delivers a maximum torque of 90,4 m*Nm (600 mA). Without load, its rotation speed is around 220 rotations per minute.

The current consumption will depend heavily on the load it is driving. Under normal conditions it can be around 300 mA and we do not recommend a continuous use above 600 mA.

If the motor is blocked the current will exceed the maximum limit of the LPF system. Turn the Power OFF, unblock the motor and turn Power ON again.

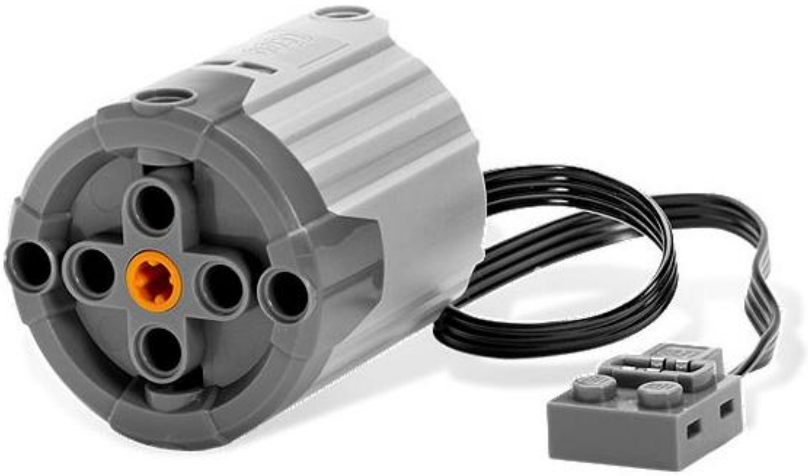


Figure 11: LEGO Power Function XL-Motor

MOTOR TECHNICAL DATA

VOLTAGE		NO LOAD		AT MAXIMUM EFFICIENCY					STALL	
Operating range	Nominal	Speed	Current	Speed	Current	Torque	Output	Efficiency	Torque	Current
		Rpm	mA	Rpm	mA	mNm	W	%	mNm	mA
0-9V	9V	224	105	181	406	90,4	1,71	49	628,6	2019

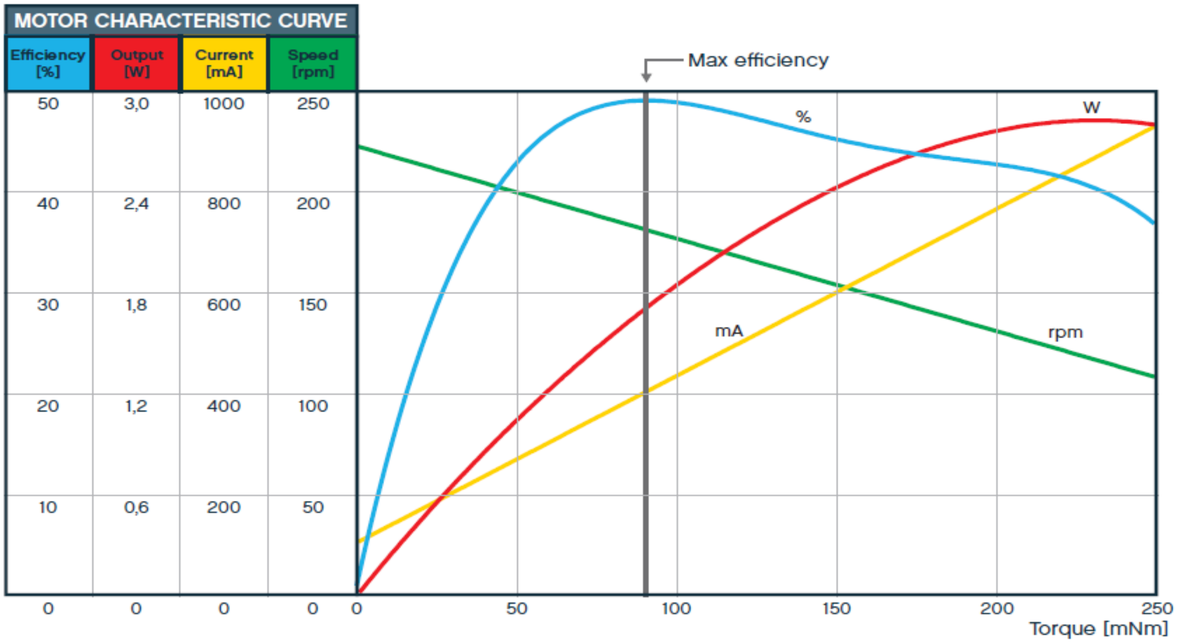


Figure 12: Technical Data from Official Data Sheet

3.3 Microcontroller Unit

In order to control all the components including: sensor, DC/servo motors, and linear gears, we chose a microcontroller, called Arduino Uno. The reason for choosing this board is because it offers a reliable and inexpensive solution to the design. Furthermore, Arduino is built as open-source hardware and it provides open-source software developing platform in C/C++. The board layout is shown in the figure below. However during our development phase, we found that the number of required pins exceeds the number of pins given on the board. Thus, we have decided to use two boards to increase the number of available pins. Additionally, the microcontroller is compatible with many Bluetooth shield modules which allows the user to control the board wirelessly. The technical specifications of the board are mentioned in the table below.

Microcontroller Module :	Uno
Processor	ATmega328P
Operating Voltage/Input Voltage	5 V / 7 – 12 V
CPU speed	16 MHz
Analog Input/output	6/0
Digital IO/PWM	14 / 6
SRAM	2 kB
Flash	32 kB

Table 6: Technical specifications of Arduino Uno

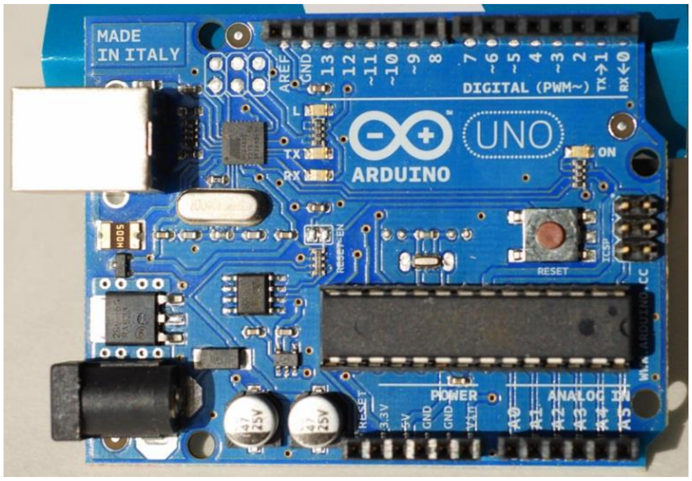


Figure 13: Arduino Uno board layout

3.4 Motor Shield

The motor driver shield is defined as the interface between the microcontroller and 2 DC motors. Even though there exists the power from the laptop to activate the microcontroller and 12 V power supply to operate 2 DC motors, the microcontroller itself is unable to change the DC motors’ revolution speed which depends on the measured distance between the user and the SmartPitcher. The motor driver shield selected for the development of motor control system mechanism is Pololu Dual VNH5019 Motor Driver Shield which supports Arduino library and assists the user to manipulate 2 bidirectional and high-powered DC motors through Arduino board [1]. After the distance value was successfully measured by the ultrasonic range finder, this chip provides a smooth manipulation to the rotational speed of the DC motors.

While the input voltage of 12V is going to be used for the full operation of 2 DC motors, the voltage of 5V will be applied to both the microcontroller and the motor shield through the shorting jumper block.

Like the following figure, Figure 14, the Pololu motor driver shield will be connected directly on the top of the Arduino Uno Microcontroller with the use of soldered single row pin header connector.

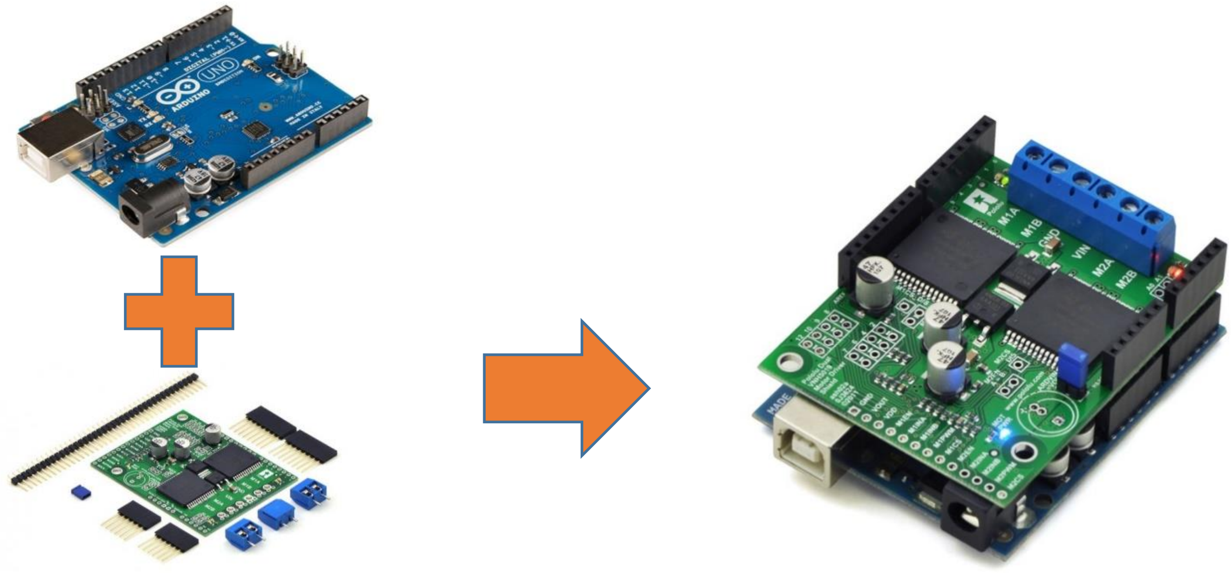


Figure 14: Pololu Motor Driver Shield integrated above the Arduino Uno Microcontroller

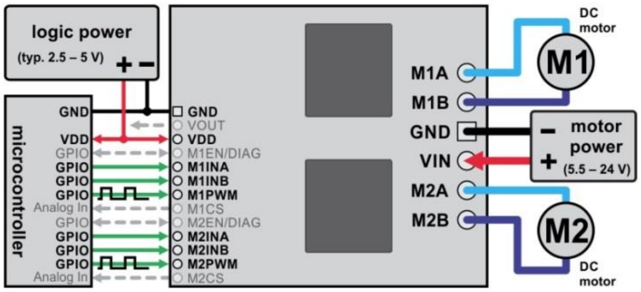


Figure 15: Wiring diagram of DC motors, power supply, and Arduino UNO

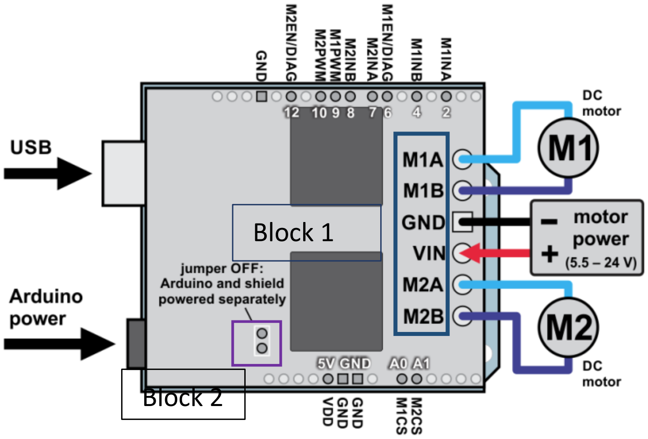


Figure 16: Upper view wiring diagram between Motor shield and Arduino UNO

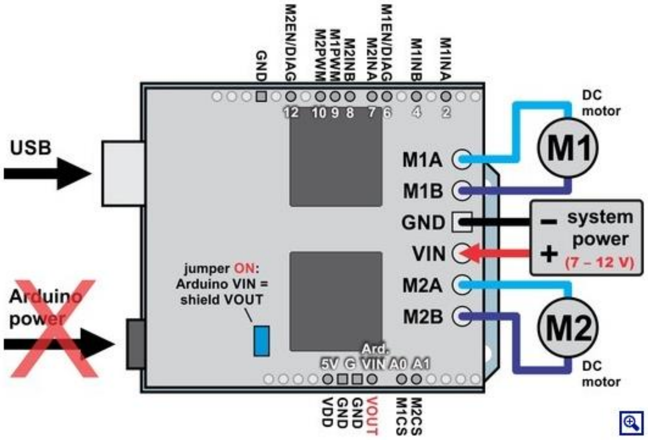


Figure 17: Upper view wiring diagram with inserted Jumper block

The last three figures above indicates the detailed interconnection schematic between DC motors, 12 V power supplies, and Arduino Uno Microcontroller. Even though most of the pins within the motor shield are connected to all pins within the Arduino board, only 8 pins will be utilized for the manipulation of DC motors' speed. 6 Pins provided within Block 1 consists of motor outputs and power inputs, where connections of DC motors and 12V power supply are being enacted through soldering 5mm terminal [1]. Moreover, 2 pins in Block 2 represents Arduino power jumper where Arduino V_{IN} = motor shield V_{OUT}, powering Arduino Uno microcontroller and motor shield using the same source, takes place. Like Figure 17, removable jumper will be inserted into Block 2 to power the Arduino directly from the motor shield [1].

3.5 Bluetooth Unit

3.5.1 Seed Studio Bluetooth Shield V2

As it was discussed earlier, our SmartPitcher requires wireless communication between the machine itself and the user. In order to enable this wireless communication, Seed Studio Bluetooth Shield V2 was found after a comprehensive research. This product is shown in Figure 18 below. The main reason for choosing this product is that it made the wiring process simpler and less time consuming. All the pins are perfectly aligned to pinholes on the Arduino board as shown in Figure 19, and it can simply be attached to Arduino like a shell which simplifies the circuit design process.

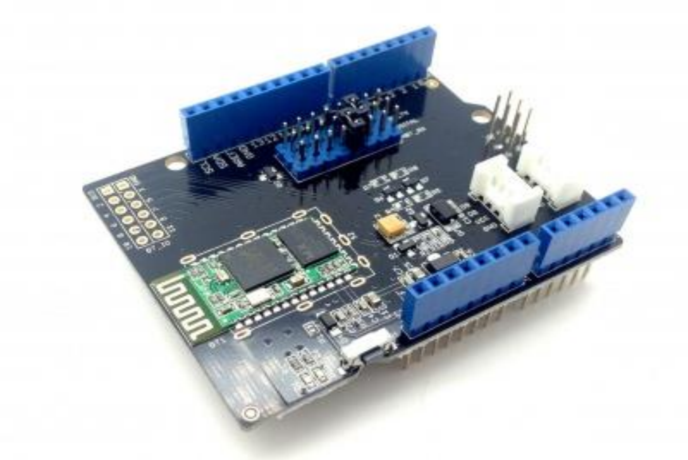


Figure 18: Seed Studio Bluetooth Shield V2

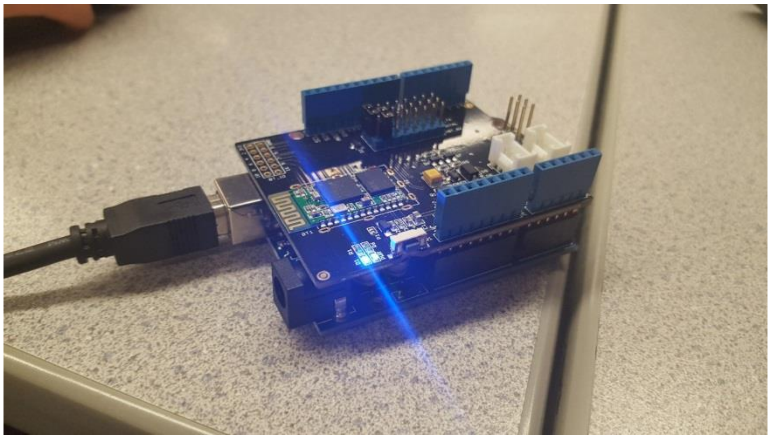


Figure 19: Bluetooth Shield Connected to Arduino

Features	
Input Voltage	3.3V
BaudRate	9600
Compatibility	Seeeduino/Arduino
Effective Distance	10m
Interface	UART interface with programmable baud rate
Default PINCODE	1234
Support	A full set of configuration commands
Antenna	On board PCB Antenna

Table 7: Bluetooth Shield Features

Android Device (Samsung Galaxy S6 Edge) was selected to be our wireless device or remote control. It will be attached to the user's arm, send data to Arduino, and trigger the pitcher system to shoot the ball. This requires an Android Application that has user interface with keypads larger than user's fingertip. Bluetooth spp pro is selected because it is a free application that provides an interface with 9 keypads, and it can be modified to send the desired data by touching any keypad. Bluetooth spp pro will be consistently running on smartphone that is attached the user's arm. When any keypad is touched, smartphone sends a signal to Bluetooth Shield attached to Arduino on the SmartPitcher platform to initiate the ball loading mechanism.

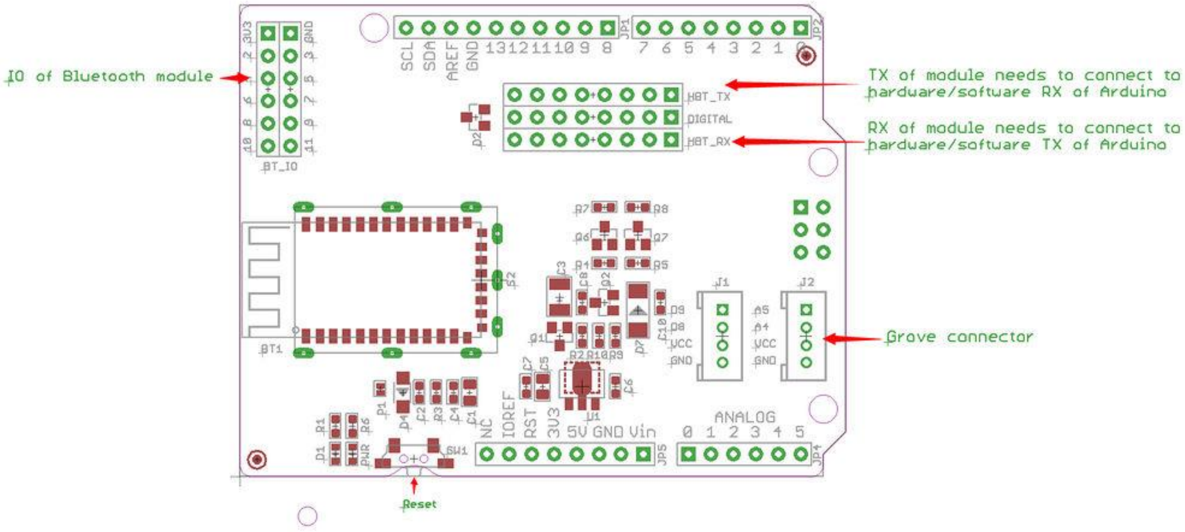

IO of Bluetooth module
TX of module needs to connect to hardware/software RX of Arduino
RX of module needs to connect to hardware/software TX of Arduino
Grove connector
Reset
BT_IO
BT1
HBT_TX
DIGITAL
HBT_RX
SCL SDA AREF GND 13 12 11 10 9 8
7 6 5 4 3 2 1 0
JP1
JP2
J1
J2
NC IOREF RST 3U3 5V GND Vin
ANALOG 0 1 2 3 4 5
JP5
JP4
SW1
PWR

Figure 20: Seed Studio Bluetooth Shield V2 layout

There is one critical factor when working with this Bluetooth Shield; there are two jumpers, and the Arduino code may not work depending on their connection. Especially for our Seeed Studio Bluetooth Shield V2, pin number for Rx and Tx was not clearly written on the board as observed in the Figure 21 below. Therefore, there were many tests to find out which pin had which number.

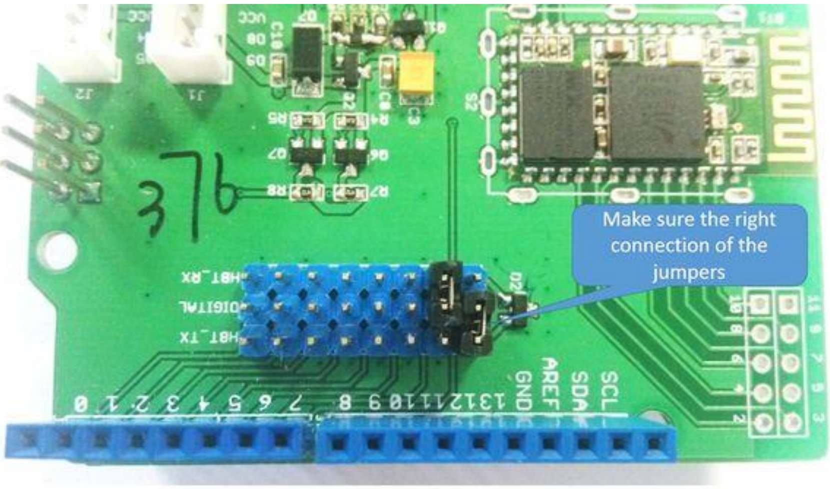
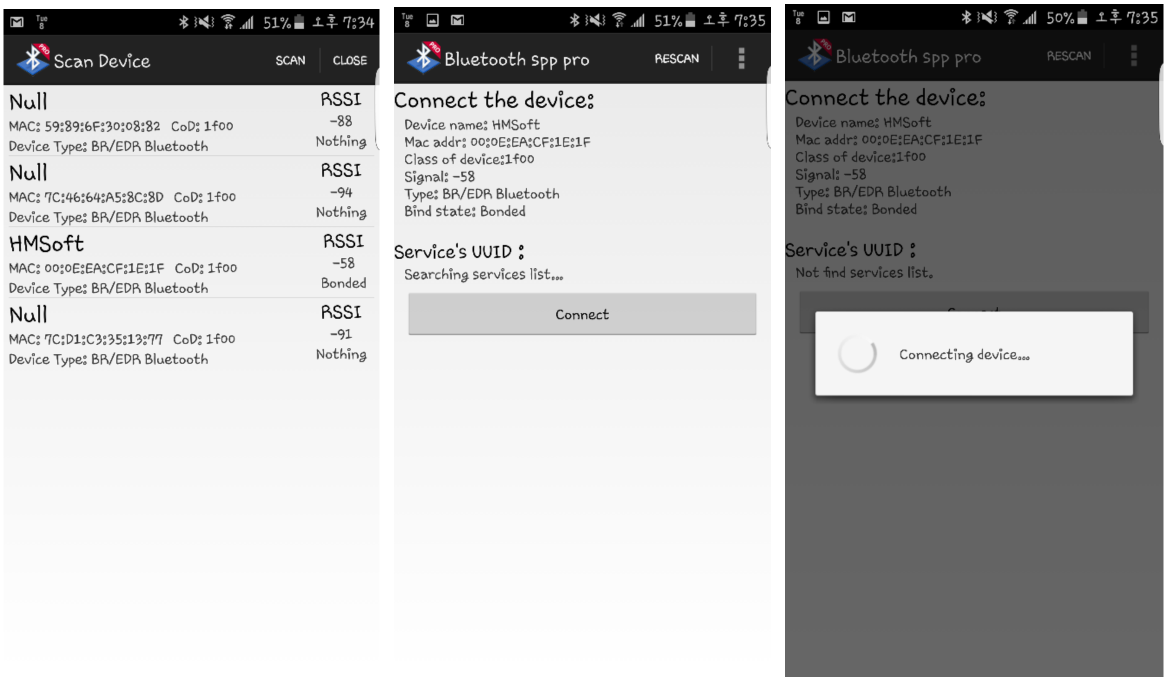


Figure 21: Seed Studio Bluetooth Shield V2 layout

3.5.2 Bluetooth Spp Pro

This is an Android Application which will be installed on the selected Android device. Our Bluetooth Shield can be connected through this App as shown in Figure 22.



Null
MAC: 59:89:6F:30:08:82 CoD: 1f00
Device Type: BR/EDR Bluetooth
RSSI -88 Nothing

Null
MAC: 7C:46:64:A5:8C:8D CoD: 1f00
Device Type: BR/EDR Bluetooth
RSSI -94 Nothing

HMSoft
MAC: 00:0E:EA:CF:1E:1F CoD: 1f00
Device Type: BR/EDR Bluetooth
RSSI -58 Bonded

Null
MAC: 7C:D1:C3:35:13:77 CoD: 1f00
Device Type: BR/EDR Bluetooth
RSSI -91 Nothing

Connect the device:
Device name: HMSoft
Mac addr: 00:0E:EA:CF:1E:1F
Class of device:1f00
Signal: -58
Type: BR/EDR Bluetooth
Bind state: Bonded

Service's UUID :
Searching services list...

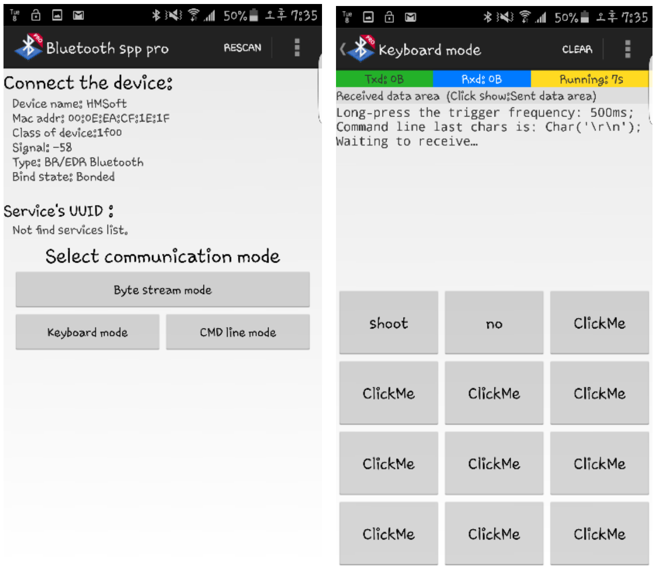
Connect the device:
Device name: HMSoft
Mac addr: 00:0E:EA:CF:1E:1F
Class of device:1f00
Signal: -58
Type: BR/EDR Bluetooth
Bind state: Bonded

Service's UUID :
Not find services list.

Connecting device...

Figure 22: Bluetooth Spp Pro, Scanning and Connecting Interface

After the Android device and Bluetooth Shield are successfully paired and connected, Bluetooth spp pro gives choices of interface: Byte stream mode, keyboard mode and command line mode as shown in Figure 23.



Bluetooth spp pro

Connect the device:

Device name: HMSoft
Mac addr: 00:0E:EA:CF:1E:1F
Class of device:1f00
Signal: -58
Type: BR/EDR Bluetooth
Bind state: Bonded

Service's UUID :

Not find services list.

Select communication mode

Keyboard mode

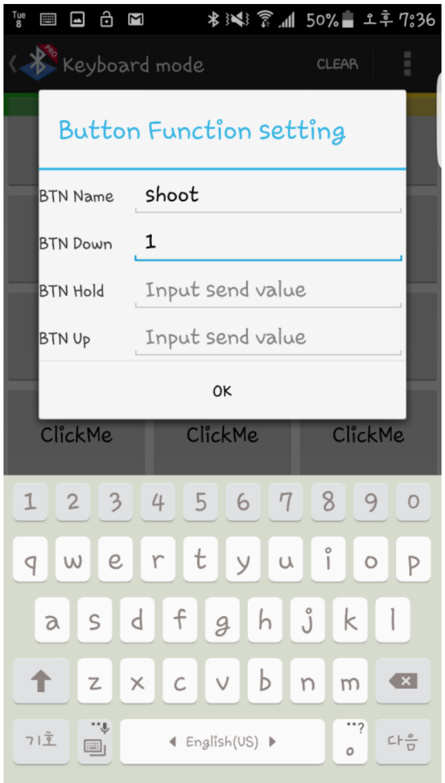
Txd: 0B Rxd: 0B Running: 7s

Received data area (Click show:Sent data area)

```
Long-press the trigger frequency: 500ms;
Command line last chars is: Char('\r\n');
Waiting to receive…
```

Figure 23: Interface Offering 3 modes. Right: Keyboard mode

Desired data to be sent can be modified in the Application as shown in Figure 24 below.



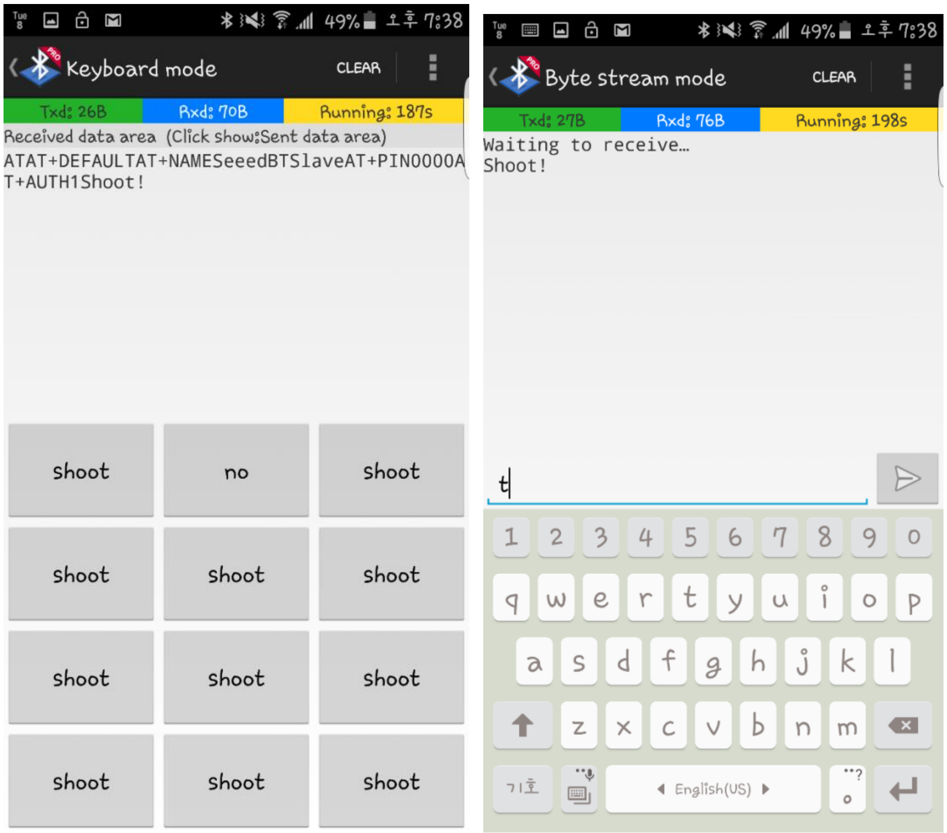
Keyboard mode

Button Function setting

BTN Name: shoot
BTN Down: 1
BTN Hold: Input send value
BTN Up: Input send value

Figure 24: Keyboard Mode Button Modification

Now, when any button is clicked or touched our Android device sends data to Bluetooth Shield and controls Arduino wirelessly. To test whether it is working, test case was created; as the Android device sends character, ‘t’ as data, Arduino will send a message, “Shoot!”, to our Android device as shown in Figure 25 below.



```
Received data area  (Click show:Sent data area)
ATAT+DEFAULTAT+NAMESeeedBTSlaveAT+PIN0000A
T+AUTH1Shoot!
```

```
Waiting to receive…
Shoot!
```

Figure 25: Testing Received Data by Keyboard Mode. Right: Testing With Byte Stream mode

3.6 Webcam

Webcam is a video camera that inputs to a computer, streaming image in real time to computer network. It will be powered by USB cable from personal computer, interfacing with Roborealm visual software application. We will discuss later in the document about Roborealm software application. However, the motion tracking system wouldn’t be able to work without a webcam. Consideration must also be given to the sensor/resolution of the webcam where the better webcam can detect the color scale in greater distance to track the motion. Webcam is simply connected to the personal computer and the rest of the setting is made by Roborealm application. The following table and figure display the features of the webcam from Farassoo.

Farassoo Webcam FC-1400
Sensor/Resolution: CMOS with 300k pixels (640x480)
Gross Weight: 105g
Dimensions: 63.5 (w) x 87 (h) x 68 (d) mm
Precise Focus Tuning
30 FPS Video Image at VGA
Rotator View Angle
16-bit Color Display

Table 8: Features of Farassoo Webcam FC-1400

Figure 26: Connect Webcam to Personal Computer

Farassoo webcam attaches in front of the platform. If SM is a neck of the SmartPitcher, the webcam is an eye to identify objects ahead. The following figures are the graphical communication of the design building application areas that are discussed as 2D geometry in mechanical drawing.

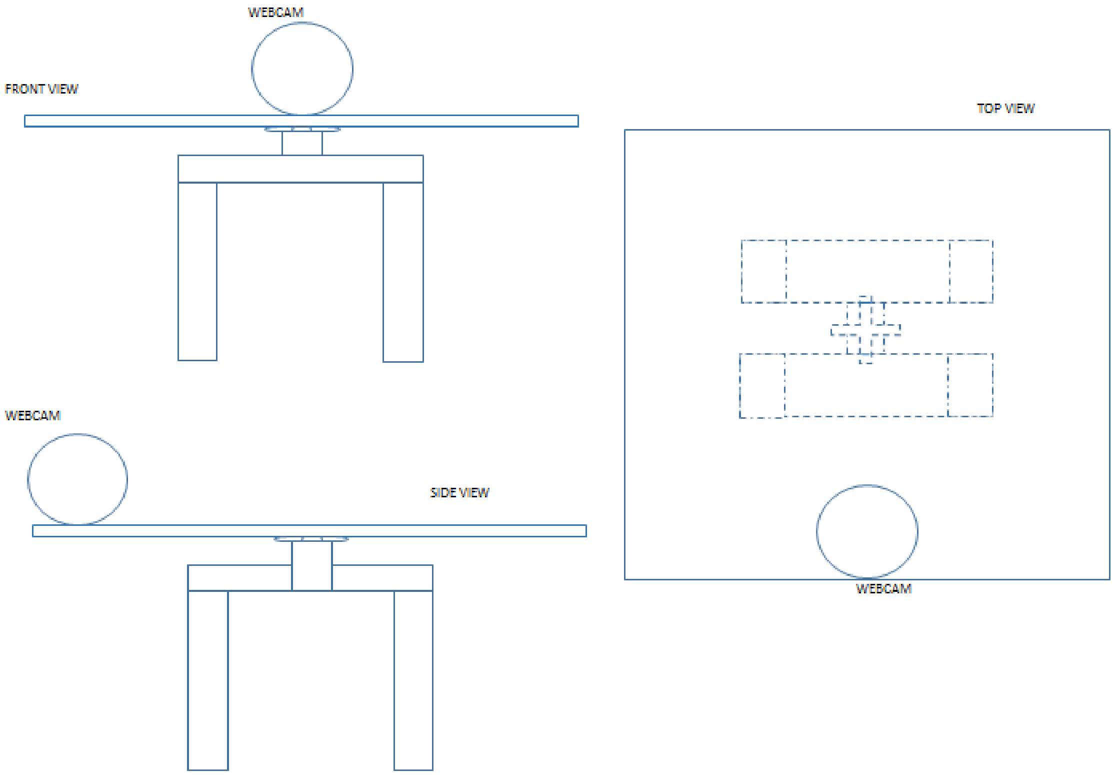


Figure 27: 2D Graphical Communication of Design of Webcam on Platform

3.7 Gear Mechanism

To design our ball loading mechanism, we use a set of rack and pinion gears in order to convert rotation into linear motion. A rack and pinion gears system is composed of two gears. The standard round gear is the pinion gear, and the straight gear is the rack. The rack is teethed which meshes with the teeth of the pinion gear.

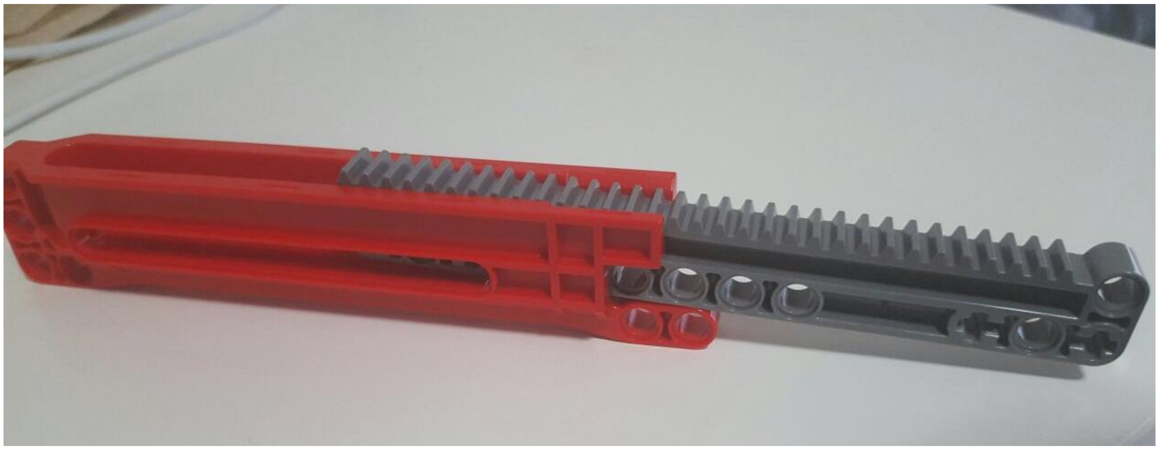


Figure 28: Rack and pinion gears (side view)

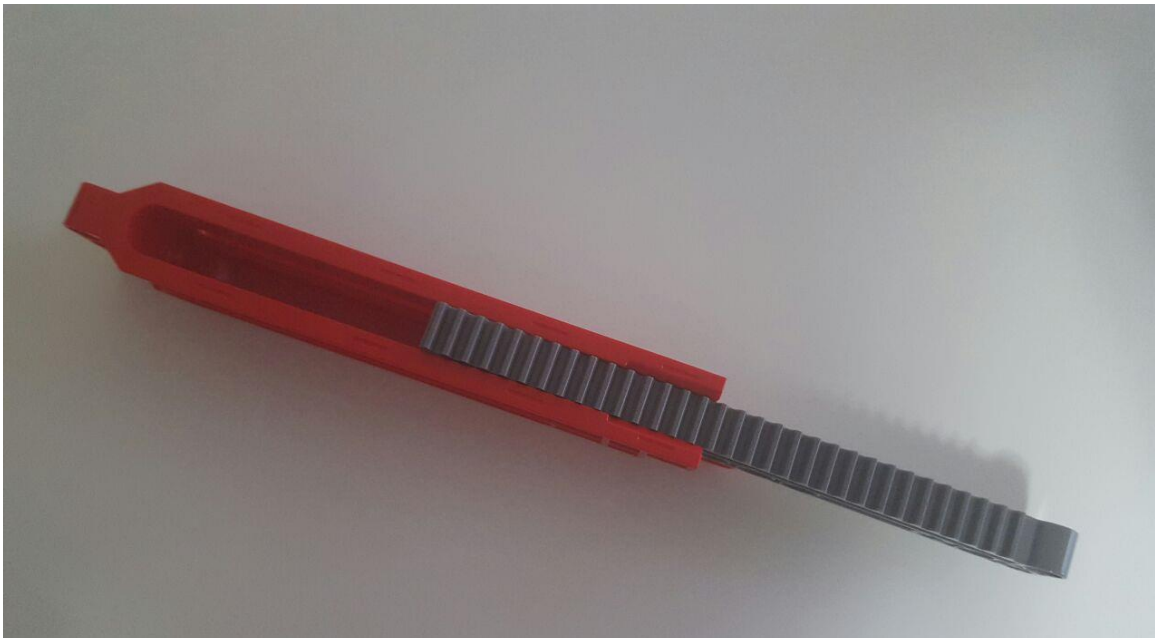


Figure 29: Rack and pinion gears (up view)

Rack and pinion gears deliver a less mechanical benefit than other known mechanisms that we could use, but they were chosen because they offer a better feedback and steering sensation. The speed with which the rack moves as the pinion turns is determined by the diameter of the gear and the number of teeth on the rack does not affect to the velocity ratio. For a rack and pinion to mesh together properly, the pitch of rack and pinion must be equal, i.e., $p_{pinion} = p_{rack}$. The gear ratio equation is given by

$$\frac{\omega_{pinion}}{V_{rack}} = \frac{1}{r_{pinion}} \tag{1}$$

The linear speed of the rack is basically a function of the pinion pitch radius and the angular velocity of the pinion.

Figure 30: Gear Rack

Figure 31: Linear Gear

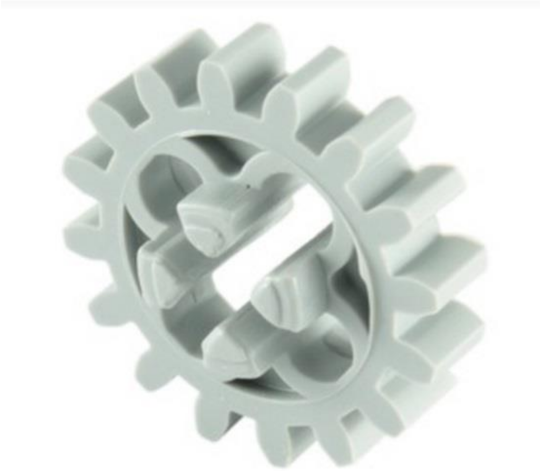


Figure 32: Round Gear

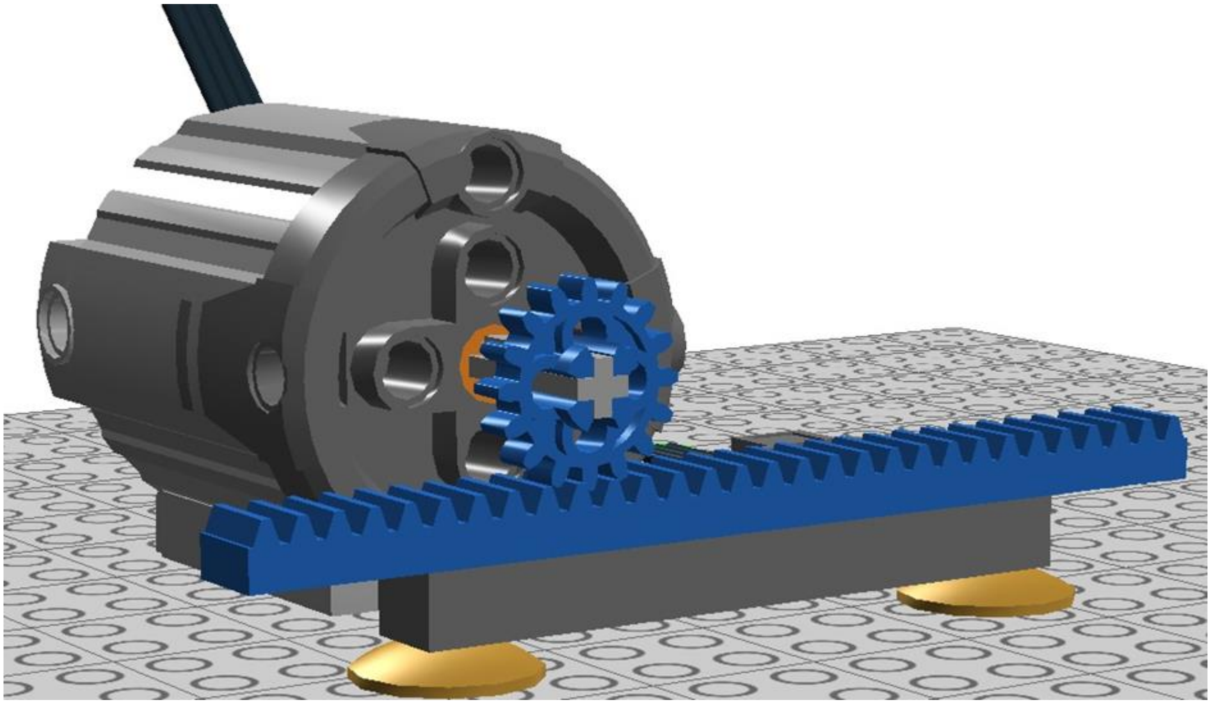


Figure 33: Rack and pinion gears connected to motor

Once a “shoot” signal is received through our Bluetooth shield, the motor turns the rounds which engages the tooth on the linear gear and pushes the linear gear forward. The linear gear hits the ball that is in the front of ball cartage and pushes it into the spinning wheels.

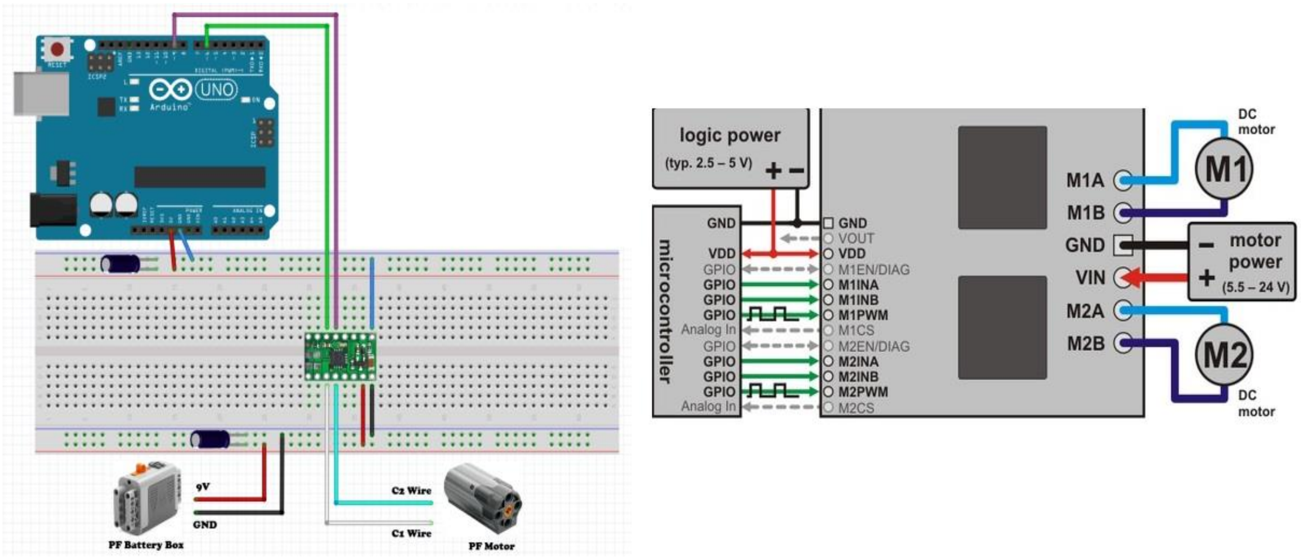


Figure 34: Circuit Schematic and breadboard connection

The battery box powers the system. A green light indicates that the power is on. The orange control switch can be pushed in either side driving a motor in either direction. In the middle position power is turned off. The plug is a feeding plug just like the usual AC plugs are. The battery box needs to be filled with 6 AA-size batteries

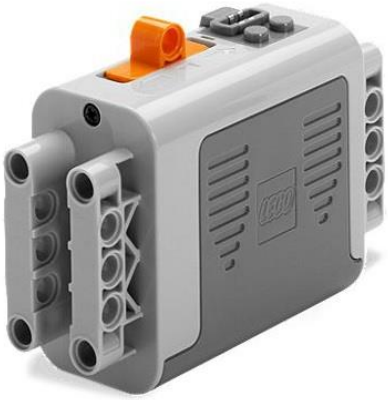


Figure 35: LEGO Battery

The motors are internally highly geared down, turning slowly and delivering high torque. The XL motor is studless only and its diameter is 5L wide. To control a motor, large amount of power is needed, more than a microcontroller can handle. We use special motor driver chips to handle the motor current that can be controlled by a signal from the Arduino. There are numerous motor control shields available for the Arduino, but our company preferred using external drivers, since they can connect to any microcontroller out there.

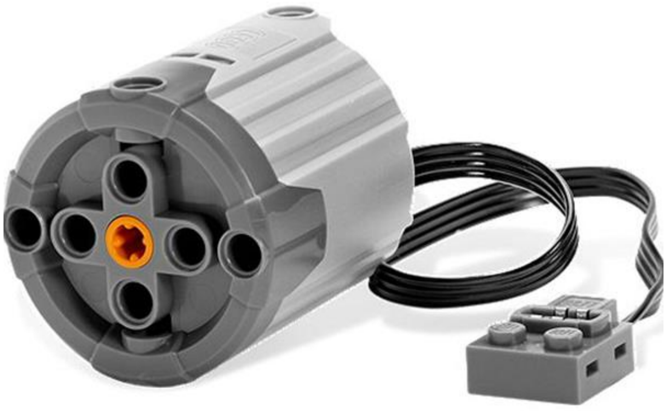


Figure 36: LEGO motor

4. Firmware Design

This section lists the programming codes which will be applicable to the hardware components of the SmartPitcher.

4.1 Microcontroller

As mentioned above in section 3.3, we are using two boards to fit all the components due to the lack of available pins. So the first microcontroller is used to develop sensor, LCD, and motor driver, and the second microcontroller is connected with the Bluetooth shield, and the servo motor which is used to rotate the whole platform. Fortunately, both of the microcontrollers are equipped with full compatibility for Arduino IDE software. The software provides a GCC C/C++ compiler, development platform, along with many built-in libraries, which are designed specifically for microcontrollers that handle communication at the register level.

4.1.1 Sensor + LCD

The ultrasonic sensor and LCD are equipped with pre-existing libraries in the IDE. The standard pin connection to the microcontroller is previously described in Section 3.1. In the IDE all the pin connections follow the typical C variable convention and declaration. The table below summarizes all the used libraries,

Pin Connection	Pin Declaration	Libraries
Pin 11 to ECHO	#define ECHO_PIN 11 // Defines Pin 11	#include <wire.h>
Pin 12 to TRIG	#define TRIGGER_PIN 12 // Defines Pin 12	#include <LiquidCrystal_I2C.h> // display library
A5 to SCL	#define MAX_DISTANCE 500 // Defines maximum distance	#include <NewPing.h> // Sensor library
A4 to SDA		

Table 9: Pin connection convention in Arduino IDE

The following function is called from the <NewPing.h> library to setup the pins that are declared and measure the distance.

```
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
unsigned int uS = sonar.ping(); // send pings in ms to measure distance
```

The following functions are called from the <LiquidCrystal_I2C.h> library to display the variable 'int uS' in cm on the LCD display.

```
lcd.setCursor(4,0);             // Set Cursor
lcd.print("Rangefinder");         // text

lcd.setCursor(2,2);             // move Cursor
lcd.print("Distance: ");        // Print "Distance"
lcd.setCursor(12,2);             // move Cursor
lcd.print(uS / US_ROUNDTRIP_CM); // Convert ping time to distance in cm and print
result (0 = outside set distance range)
lcd.setCursor(15,2);             // move Cursor
lcd.print("cm");              // print Something
```

4.1.2 Servo Motor

Arduino IDE provides a library that allows the microprocessor to control servo motors (SM). This library supports up to 12 motors at once on most of the Arduino board. Moreover, Arduino Mega model can hold up to 48 without interfering with PWM functionality. The standard pin connection to the microprocessor is previously described in section 3.2.2. The following IDE lines of code are essential and significant in motion tracking system to control the SM.

IDE Code	Description
#include <Servo.h>	Library that allows the board to control SMs
myservo.attach(9)	Attach the servo variable to a pin
Serial.begin(9600)	Open the serial port with a 9600 baud rate
Serial.available()	Get the data when it becomes available at the serial port
incomingData[i] = Serial.read()	Assign the input value to the incoming data buffer
Distance = atoi(incomingData)	Converting ASCII to integer

Table 10: IDE Code description for Motion Tracking System

In order to transfer information through processor and Roborealm visual software application, the communication channel must be declared to match the identical baud rate. In the loop, the processor waits for data to become available through the serial port, receiving and storing the in-coming data in the buffer.

Just to explain briefly about the Roborealm to understand how the motion tracking system works with SM, Roborealm is able to snip the size of the assigned object by filters. It supports the XY values for the center of gravity (COG) box size; thus, we know how far the object is positioned away from the origin. If the distance is bigger than negative 15 then add 1 to the current angle value and update the servo motor. Vice versa, if the distance is bigger than positive 15 then subtract to the current angle value and update the servo motor in order for the object is placed in the range of (-15,+15) which is nearly in the center of the origin. The detailed features of Roborealm are explained in Section 5 Software Design. Lastly, constrain the value of the current angle to within 0 to 180 degrees to update the COG box size value according to the XY coordinates.

4.1.3 Firmware (Bluetooth Shield)

In this section, our test case for the Bluetooth Shield will be explained. On the phone it will receive “Shoot!” as message whenever keypad is touched. Keypads were modified to send ‘1’ only as data.

UART Pin Number	Pin Declaration	Library
7	#define RxD 7	SoftwareSerial.h
6	#define TxD 6	SoftwareSerial.h

Table 11: Pin Declaration

UART pins are set with jumpers on Bluetooth Shield.

Following lines are used for Arduino and Bluetooth Shield to receive, read and transmit data.

```
if(blueToothSerial.available())
    {//check if there's any data sent from the remote bluetooth shield
      recvChar = blueToothSerial.read(); //data is stored
      Serial.print(recvChar); //send the data to Arduino

      if(recvChar == '1')//check if the data received is 1
      {
        blueToothSerial.print("Shoot!!"); //this will display the message

         delay(2000);}} }
```

In this section of if statement, motor control will be added. As the Bluetooth Shield receives the data ‘1’, Bluetooth Shield will send data to Arduino to start the motor. The motor will rotate twice, pushing a tennis ball using linear gear, delay, and then move the linear gear back to original position.

Also, following function, ‘setupBlueToothConnection(),’ is used to initialize our Bluetooth Shield.

```
void setupBlueToothConnection(){
        blueToothSerial.begin(9600);  //sets the data rate in bits per second for Bluetooth Shield
        blueToothSerial.flush();}
```

4.2 Motor Shield

The manual of Pololu Dual VNH5019 Motor Driver Shield provides the pre-existing Arduino libraries which can be revised and utilized for a smooth manipulation of up to two motors during the integration of the microcontroller and the motor shield [2, 3]. Like the functionality of Pololu Dual VNH5019 Motor Driver Shield itself, the Arduino library is also designed for the manipulation of only DC motors.

Our company is intending to create the firmware codes where the rotational speed of the pitching machine wheels and DC motors alters once the user is located either below or above certain amount of distance range from the SmartPitcher. For instance, the revolution speed of the motors is 300 rpm when the distance is measured as "≤ 3 meter", and 450 rpm if the machine and the user are separated "> 3 meter". Therefore, the types of programming codes that are going to be applied in Arduino IDE 1.6.7 will be the communication to DC motors, setting the motor speed limits, and controlling the motor speed based on the distance value collected from the ultrasonic range finder.

The following "void" functions provided below describes the codes that relates to the revolution speed of 2 DC motors. While the first/second "void" function declares the speed limit of the first/second DC motor with the constant variable RPM, declared by the user, the third "void" function sets the speed of both DC motors.

```
void DualVNH5019MotorShield::setM1Speed(int speed)

{

  if (speed > RPM)  // Max

    speed = RPM;

}
```

```
void DualVNH5019MotorShield::setM2Speed(int speed)

{

   if (speed > RPM)

       speed = RPM;

}

void DualVNH5019MotorShield::setSpeeds(int m1Speed, int m2Speed)

{

  setM1Speed(m1Speed);

  setM2Speed(m2Speed);

}
```

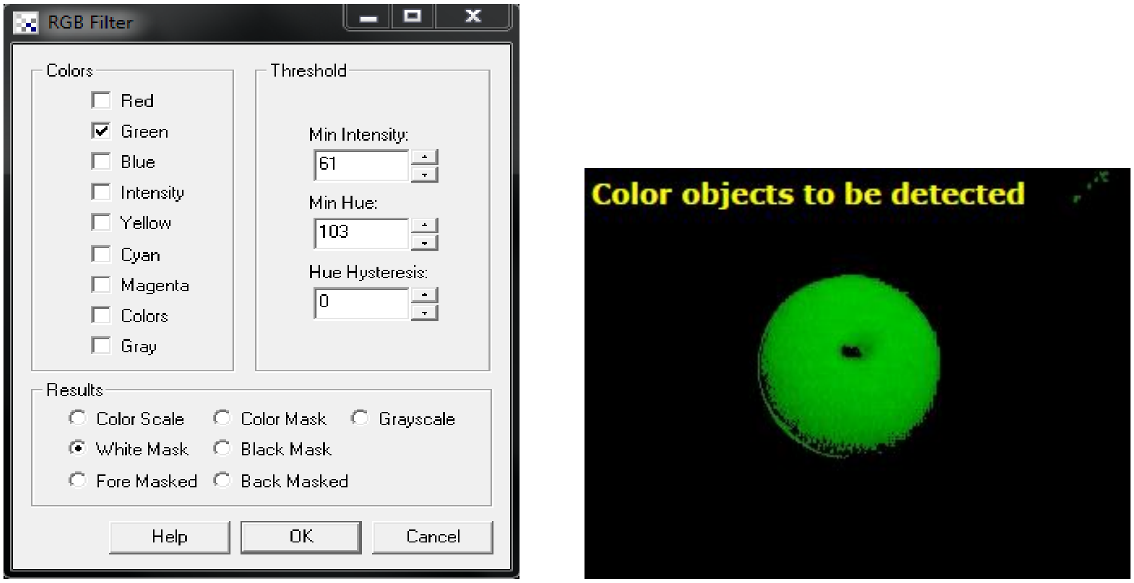
5. Software Design

The horizontal motion tracking system by SmartPitcher operates by using Arduino IDE Codes that interfaces with Roborealm (RR), a software application for use in machine vision. Moreover, RR data travels by the serial communication module to Arduino Uno (AU). RR supports various filters and features, enhancing image pixels or developing characteristics of image to see the environment, process acquired images, analysis of what need to be done and send the signals to the external hardware controller.

5.1 Roborealm Color Code Identification with Arduino IDE

Roborealm (RR) is a software application for use in machine vision, computer vision, image analysis, and robotic vision system used for color object tracking method. RR compiles image processing functions that can interface with a webcam. It supports hundreds of image processing modules with real-time parameter changes. Applying eight different filters and image processing modules can detect the object with a clear vision through webcam Frassoo FC-1400. The athlete wears one color jersey vest in order to diminish all pixels that are not of the selected colors.

RGB Filter uses RGB values to focus the point towards the primary RGB colors. We set the colors to Green and threshold values depend on the surrounding environment. Consideration can be given to the light environment and reflection environment to enhance the normalization of the pixel. Threshold selects the minimum intensity of the pixel value that can detect the color and remove the lower pixel value. The minimum hue threshold also removes the colors that are not enough blue and red. Hue Hysteresis delivers blobs to grow into surrounding area. The selected values and setting to detect the green ball for testing is shown in the figure.



RGB Filter

Colors

□ Red
☑ Green
□ Blue
□ Intensity
□ Yellow
□ Cyan
□ Magenta
□ Colors
□ Gray

Threshold

Min Intensity: 61
Min Hue: 103
Hue Hysteresis: 0

Results

○ Color Scale ○ Color Mask ○ Grayscale
● White Mask ○ Black Mask
○ Fore Masked ○ Back Masked

Figure 37: RGB Filter pointing the primary RGB color

Blob Size Filter is used to remove other objects below certain size. If there are more than two objects detected by RGB Filter, interfering to detect the athlete, the Blob Size Filter chooses to remove all but the largest blobs. The figure below displays the feature of Blob Size Filter to eliminate the same colored but smaller objects, entering value to count the objects in the order of the largest to smallest.

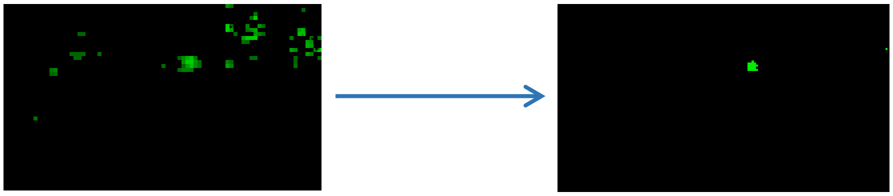
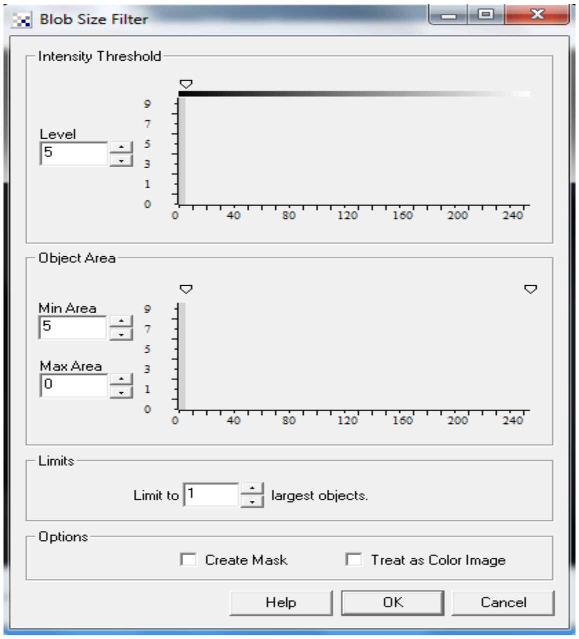


Figure 38: Before and After Effect of Blob Size Filter



Blob Size Filter

Intensity Threshold

Level: 5

Object Area

Min Area: 5

Max Area: 0

Limits

Limit to 1 largest objects.

Options

□ Create Mask

□ Treat as Color Image

Figure 39: Blob Size Filter to limit the number of objects

Next we consider the module called, Center of Gravity (COG), where the mass statistic calculates the position of the image lies. Show COG as a crosshair marker in the center of the object. Connect COG of Center of Screen to deliver variable information of COG X-Y coordinate. As mentioned previously from Servo IDE code, the information data about X-Y coordinates values travel through the serial communication with 9600 baud rate and 8 bits of data bits, sending data through microprocessor to Servo motor to rotate appropriate amount of angle. The figure below present what feature does COG can do after the RGB Filter.

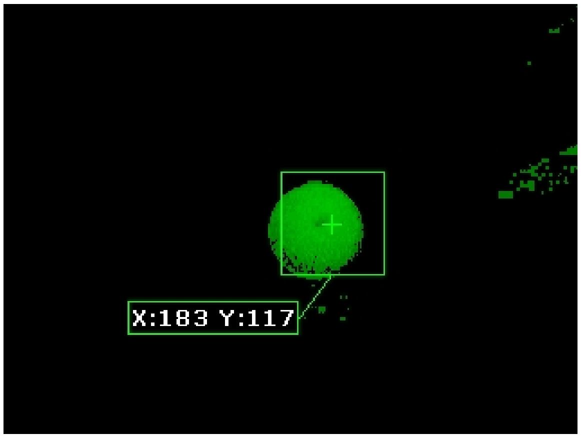
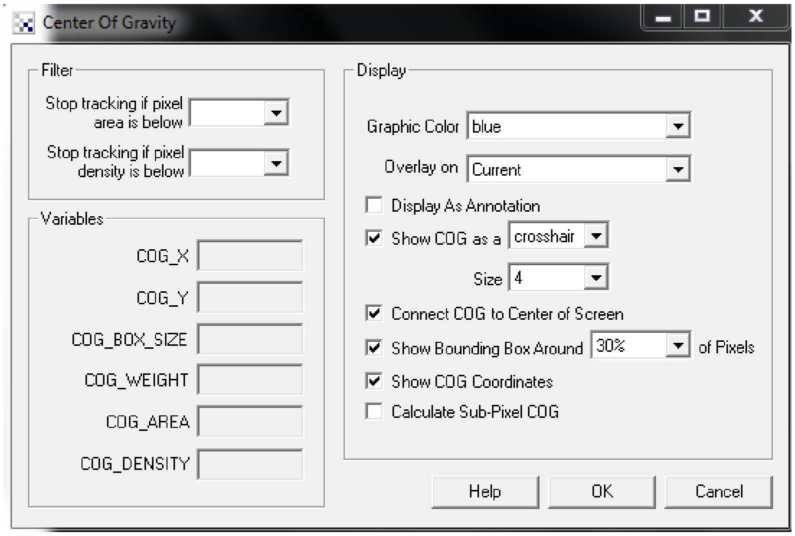


Figure 40: Center of Gravity with display modules and XY-coordinate values



Center Of Gravity

Filter

Stop tracking if pixel area is below

Stop tracking if pixel density is below

Variables

COG_X

COG_Y

COG_BOX_SIZE

COG_WEIGHT

COG_AREA

COG_DENSITY

Display

Graphic Color: blue

Overlay on: Current

☐ Display As Annotation

☑ Show COG as a crosshair

Size: 4

☑ Connect COG to Center of Screen

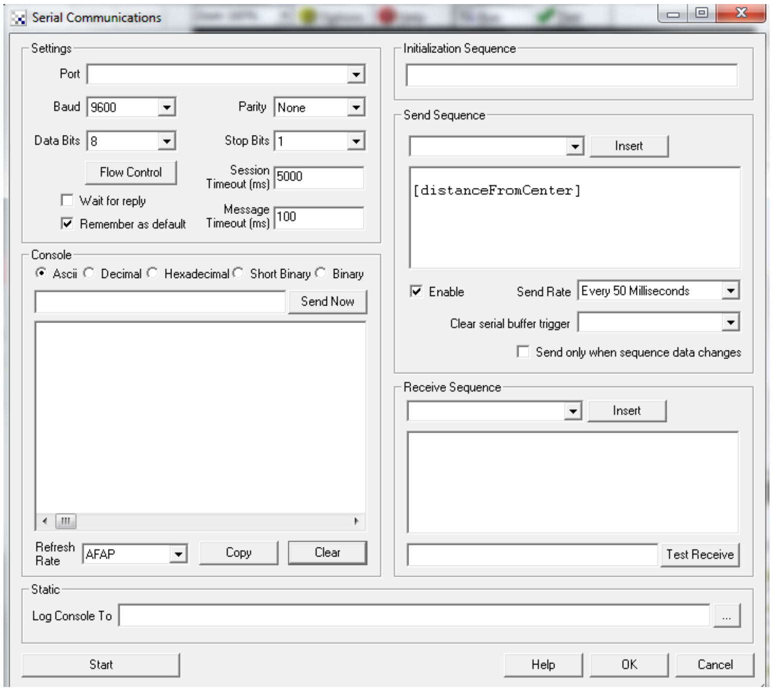
☑ Show Bounding Box Around 30% of Pixels

☑ Show COG Coordinates

☐ Calculate Sub-Pixel COG

Figure 41: Center of Gravity functional features

The visual basic script, VBScript Program module, provides a way to create custom scripts that can be used to process image statistics and map then toward servo motor values. It helps when the module intends to use a quick performance in custom operations without the needs of implantation of a Plugin or use the API which requires external tools. Serial Communication module can set the port, baud, and data bits to interface with the microprocessor. Ultimately leads to control the servo motor. SmartPitcher's tracking system can operation in horizontal rotation; thus, the distance from center only matters in the horizontal perspective, not in vertical perspective. Serial communication sends sequence of the distance from center in 50 millisecond sending rate. The figure below displays the features of Serial module.



Serial Communications

Settings

Port

Baud 9600

Parity None

Data Bits 8

Stop Bits 1

Flow Control

Session Timeout (ms) 5000

Message Timeout (ms) 100

□ Wait for reply

☑ Remember as default

Console

● Ascii ○ Decimal ○ Hexadecimal ○ Short Binary ○ Binary

Send Now

Refresh Rate AFAP

Copy Clear

Initialization Sequence

Send Sequence

Insert

[distanceFromCenter]

☑ Enable

Send Rate Every 50 Milliseconds

Clear serial buffer trigger

□ Send only when sequence data changes

Receive Sequence

Insert

Test Receive

Static

Log Console To ...

Start Help OK Cancel

Figure 42: Serial Communication module options

6. Test Plans

In order to verify that all the functions within SmartPitcher are operating properly and precisely, various types of testing within software, mechanical, and final prototype sections are going to be performed. The following table, Table 12, indicates the detailed schedule of our company's testing plan.

Table 12: Timeline Schedule throughout the Term

MONTHS	JANUARY			FEBRUARY				MARCH					APRIL	
WEEKS	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	5th	1st	2nd
PHASE														
Research	→	→	→	→	→	→	→	→	→	→				
Design			→	→	→									
Development					→	→	→	→						
Integration								→	→	→	→			
Testing											→	→	→	→
Parts order/Purchase			→	→			→	→						
Electronics/Hardware					→	→	→	→		→	→	→	→	
Mechanical	→	→	→	→	→	→	→	→	→	→	→	→	→	→
Software				→	→	→	→	→	→	→	→	→	→	
DELIVERABLE														
Functional Spec						◆								
Design Overview								◆						
Written Progress											◆			
Test Plans											◆			
Presentation														◆

6.1 SOFTWARE TESTING

Software Testing will be performed from the beginning of development to the end of integration phase. During the development phase, three types of software testing will be conducted, since there exists three essential parts, motor speed control, motion tracking system, and loading mechanism, to create the SmartPitcher.

In order to manipulate the motor speed of the pitching machine, the most recent version of Arduino IDE (1.6.7) and the applications of C/C++ codes will be utilized [19]. Software testing of the codes written in Arduino IDE plays an important role in steps where the calculations of the distance between the user and the device are taking place, and the display of the calculated value correctly appears within Arduino LCD module.

The significance in software testing also takes place while creating the motion tracking system of the SmartPitcher with the computer/robotic vision analysis software, Roborealm [20]. The software possesses API feature that facilitates the image processing/analysis procedure and provides Windows-based GUI which can be experimented with wide variety of modules, including VBScript and Python [14]. This software and possibly Python codes will be used and tested to identify whether the machine can detect the precise movement of athlete and then rotates to the correct direction. Moreover, the codes which execute color identification feature will be included to detect the motions of athlete and his/her uniform for efficient analysis.

Lastly, the software testing with the use of pre-existing smartphone application, Bluetooth spp pro that includes the Bluetooth triggering feature will be conducted to operate the successful ball loading mechanism. The application is utilized to trigger the ball to be fired once the athlete transmits, and the machine receives the Bluetooth signal. Moreover, it is also going to be tested to verify whether the ball is being shot accurately and immediately after receiving signal response from the user.

6.2 MECHANICAL TESTING

Mechanical Testing will be performed simultaneously with software testing and will take place from the beginning of development to the end of integration phase. SmartPitcher utilizes wide variety of mechanical components, such as sensors, motor driver, microcontroller, webcam, motors, and linear gear. During the development phase, all the mechanical components will be integrated while conducting the software testing. Then, the testing and verifications of whether the mechanical components of SmartPitcher are functioning properly or not will be performed until the end of integration phase. The following subcategories display the components that will be used and tested in motor speed control, motion tracking system, and loading mechanism section.

6.2.1 MOTOR SPEED CONTROL SYSTEM

Motor speed control system involves range finder, microcontroller, motor driver, two DC motors, and two spinning wheels of the SmartPitcher. The following functionalities from these components shall be tested once they are connected to the Arduino software, and the programming codes within Arduino IDE are executed.

- Range finder sensor measures and identifies the distance of either a dummy objector an individual that is placed up to the distance between 3 and 10 metre.
- Two DC motors, with 12 V power supply, are connected to the Pololu VNH5019 Motor Driver Shield in order to trigger two pitching machine wheels to spin.
- Depending on the distance range between the user and the SmartPitcher, Arduino microcontroller and motor driver shield either restricts or manipulates the rotation speed of spinning wheels, with the applications and executions of C/C++ codes from Arduino IDE.
- The first row of 20X2 Arduino LCD modules displays correct displacement (m) value between the SmartPitcher and the athlete while the second row shows accurate revolution speed of two spinning wheels (rpm).

6.2.2 MOTION TRACKING SYSTEM

The following features from camera and Servo motor will be tested while they are being operated and integrated with Roborealm.

- Camera detects the movement of athlete by receiving the information from Roborealm's real-time tracking system and the color code identification feature.
- After camera tracks the athlete's uniform color, servomotor facilities the rotation of the SmartPitcher up to 180°range and will point to the user at a predetermined speed.

6.2.3 BALL LOADING MECHANISM

One of the most special features of the SmartPitcher is the automatic ball loading mechanism which loads and fires the ball from the machine without the need of manpower. Once the user provides Bluetooth signal using the Arduino Bluetooth Control app, Bluetooth spp pro, the Bluetooth shield attached above the Arduino microcontroller will trigger LEGO motor to rotate the linear gear which loads and fires the balls from the SmartPitcher. In order to build a flawless ball loading mechanism, following features from Bluetooth shield, linear gear, and LEGO stepper motor need to be tested.

- Bluetooth shield requests the microcontroller to load the ball after receiving the signal from the Arduino Bluetooth Control application.
- LEGO Stepper motor, added to the microcontroller, triggers a linear gear to perform the rotation for loading and shooting the balls to the user's location when the orders from the microcontroller are received.

6.3 FINAL PROTOTYPE TESTING

Final prototype testing of the SmartPitcher is planning to be performed during the final integration phase, which is taking place after the successful collaborations in each of software and mechanical components within motor speed control, motion tracking system, and loading mechanism parts.

After combining these three sections, it is expected to proceed to the final prototype testing phase in order to ensure that all of them work simultaneously and verify other testing issues, related to entire systems, batteries, and stabilities.

The following steps below indicate the order of entire system testing procedure:

1. Stay at most 10 meters of range after the initiation of the SmartPitcher.
2. Move around within 180° of range to identify whether the machine rotates to the correct direction at a predetermined speed.
3. Send the signal to the pitching machine by tapping the smartphone application attached on the forearm.
4. The ball is fired from the SmartPitcher to the user.
5. Repeat the above steps for approximately an hour.

The following tests based on batteries and stabilities will also be performed:

- Since the motor parts of the SmartPitcher are powered by power supplies, our product is expected to run approximately minimum an hour. This test will be conducted by operating the machine continuously for an hour during indoor/outdoor sports' practice.
- Stability testing is categorized by balance and temperature. Since, the stable balance of the SmartPitcher is essential while the two wheels are spinning. We can test the balance by constantly changing the position of the user, and check to see if there are certain positions that cause much stress on the machine. Because we are using power supplies to operate the motors and the microcontroller, we have to test if the temperature will cause overheating during the operation. This test will be conducted by continuously monitoring the rise of temperature and comparing the corresponding value with the mechanical/hardware components' specification limits.
- Different values of voltage ranging from 1V to 12V, is going to be applied to the motors to obtain the optimum power supply values.

7. Conclusion

The in-depth analysis and details arranged within this document specifically and concisely provide the design specifications of the SmartPitcher. As mentioned within the content of this document, there exist three important mechanisms within the SmartPitcher: Motion speed control, motion tracking mechanism, and Bluetooth automated ball loading system. This indicates that the intention of our company is conducting a thorough design analysis and various types of testing in each mechanism for the successful development of the SmartPitcher prototype.

All or most of the following types of design analysis and test plans will be conducted in each mechanism of the SmartPitcher.

- Hardware Design of each component describes how all components are planning to be connected and integrated each other.
- Firmware Design demonstrates the programming codes which are going to be applied into the hardware components.
- Software Design describes the requirements in user settings and inputs during the use of Roborealm which provides the accurate color code identification feature.
- Software Testing is going to be conducted to gain the precise distance measurement, accurate color filtering, and stable Bluetooth connection.
- Mechanical Testing will be performed to check the performance of pitching machine wheels, rotational platform, and automatic ball shooting system.

After software and mechanical testing is completed in each mechanism, final prototype testing, where the last check-up of entire systems including balance/temperature stabilities and batteries, is going to take place.

Furthermore, our company is going to utilize this design specification document as a guide until the development of final prototype product is achieved. Furthermore, we will comply with all the requirements, provided within the functional specification documents, while working on the project design and test plans.

Even though the proof-of-concept and final prototype of SmartPitcher are currently under development, the expected completion date of the pitching machine is planning to be April 11th, 2016.

Reference

[1] Pololu. “*Pololu Dual VNH5019 Motor Driver Shield User`s Guide?*”,Pololu Corporation. [PDF]. Available: https://www.pololu.com/docs/pdf/0J49/dual_vnh5019_motor_driver_shield.pdf. [Accessed: Mar. 7, 2016].

[2] GitHUB. “*Arduino library for the Pololu Dual VNH5019 Motor Driver Shield*”, github.com. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed: Mar. 5, 2016].

[3] ARDUINO & GENUINO Forum. “*VNH5019 Motor Driver – Arduino Mega 2560 – Help Needed*”, forum.arduino.cc. [Online]. Available: http://forum.arduino.cc/index.php?topic=132552.0. [Accessed: Mar. 5, 2016].

[4] StackExchange. “*Torque kgcm (what is kgcm)?*”, electronics.stackexchange.com. [Online]. Available: http://electronics.stackexchange.com/questions/27931/torque-kgcm-what-is-kgcm. [Accessed: Mar. 4, 2016].

[5] StackExchange. “*How are current and voltage related to torque and speed of a brushless motor?*”, electronics.stackexchange.com. [Online]. Available: http://electronics.stackexchange.com/questions/39387/how-are-current-and-voltage-related-to-torque-and-speed-of-a-brushless-motor. [Accessed: Mar. 4, 2016].

[6] StackExchange. “*Is It Safe to Apply 24V Across a 12V motor?*”,electronics.stackexchange.com. [Online]. Available: http://electronics.stackexchange.com/questions/107881/is-it-safe-to-apply-24v-across-a-12v-motor. [Accessed: Mar. 4, 2016].

[7] Society of Robots – Robot Forum. “*12V motor 9.6V battery*”, societyofrobots.com. [Online]. Available: http://www.societyofrobots.com/robotforum/index.php?topic=14232.0. [Accessed: Mar. 6, 2016].

[8] StackExchange. “*How many volts will it take to run a 12 volt motor?*”, electronics.stackexchange.com. [Online]. Available: http://electronics.stackexchange.com/questions/178613/how-many-volts-will-it-take-to-run-a-12-volt-motor. [Accessed: Mar. 6, 2016].

[9] ARDUINO & GENUINO. “*Compare board specs*”, arduino.cc. [Online]. Available: https://www.arduino.cc/en/Products/Compare. [Accessed: Mar. 2, 2016].

[10] ARDUINO & GENUINO. “*Arduino Uno & Genuino UNO Overview*”, arduino.cc. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno. [Accessed: Mar. 4, 2016].

[11] J. Henrik (n.d). “*Arduino ultrasonic range finder, with I^2C lcd display*”. instructables [Online]. Available: http://www.instructables.com/id/Arduino-ultrasonic-range-finder-with-lcd-display/. [Accessed: Mar. 5, 2016].

[12] Brick Owl. “*LEGO Gear with 16 Teeth Unreinforced (4019)*”, brickowl.com. [Online]. Available: http://www.brickowl.com/catalog/lego-gear-with-16-teeth-unreinforced-4019. [Accessed: Mar. 2, 2016].

[13] Brick Owl. “*LEGO Gear Rack 14 x 2m with groove (18942)*”, brickowl.com. [Online]. Available: http://www.brickowl.com/catalog/lego-gear-rack-14-x-2m-with-groove-18942. [Accessed: Mar. 2, 2016].

[14] Brick Owl. “*LEGO Housing 2 x 15 x 3m F*/gear Rack (18940)”, brickowl.com. [Online]. Available: http://www.brickowl.com/catalog/lego-housing-2-x-15-x-3m-f-gear-rack-18940. [Accessed: Mar. 2, 2016].

[15] LEGO Shop. “*LEGO Power Functions XL-Motor*”, shop.lego.com. [Online]. Available: http://shop.lego.com/en-CA/LEGO-Power-Functions-XL-Motor-8882. [Accessed: Mar. 3, 2016].

[16] LEGO Shop. “*LEGO Power Functions Battery Box*”, shop.lego.com. [Online]. Available: http://shop.lego.com/en-CA/LEGO-Power-Functions-Battery-Box-8881. [Accessed: Mar. 3, 2016].

[17] SEEED WIKI. “*Bluetooth Shield V2.0*”, seeedstudio.com. [Online]. Available: http://www.seeedstudio.com/wiki/Bluetooth_Shield_V2.0. [Accessed: Feb. 28, 2016].

[18] LEGO. “*MOTOR SPECIFICATION*”, shop.lego.com. [PDF]. Available: http://shop.lego.com/en-CA/LEGO-Power-Functions-Battery-Box-8881. [Accessed: Mar. 1, 2016].

[19] ARDUINO & GENUINO. “*What is Arduino?*”, arduino.cc. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction. [Accessed: Feb. 12, 2016].

[20] Roborealm. “*Extending/Interfacing/Programming Roborealm*”, roborealm.com. [Online]. Available: http://www.roborealm.com/help/Integration.php. [Accessed: Feb. 13, 2016].

[21] WIKIPEDIA (2016, Feb. 19). “*GNU Compiler Collection*”, en.wikipedia.org. [Online]. Available: https://en.wikipedia.org/wiki/GNU_Compiler_Collection. [Accessed: Mar. 9, 2016].