ME102B Final Report

Nitin Balasubramanian, Christopher Liang, Sohom Roy

Opportunity

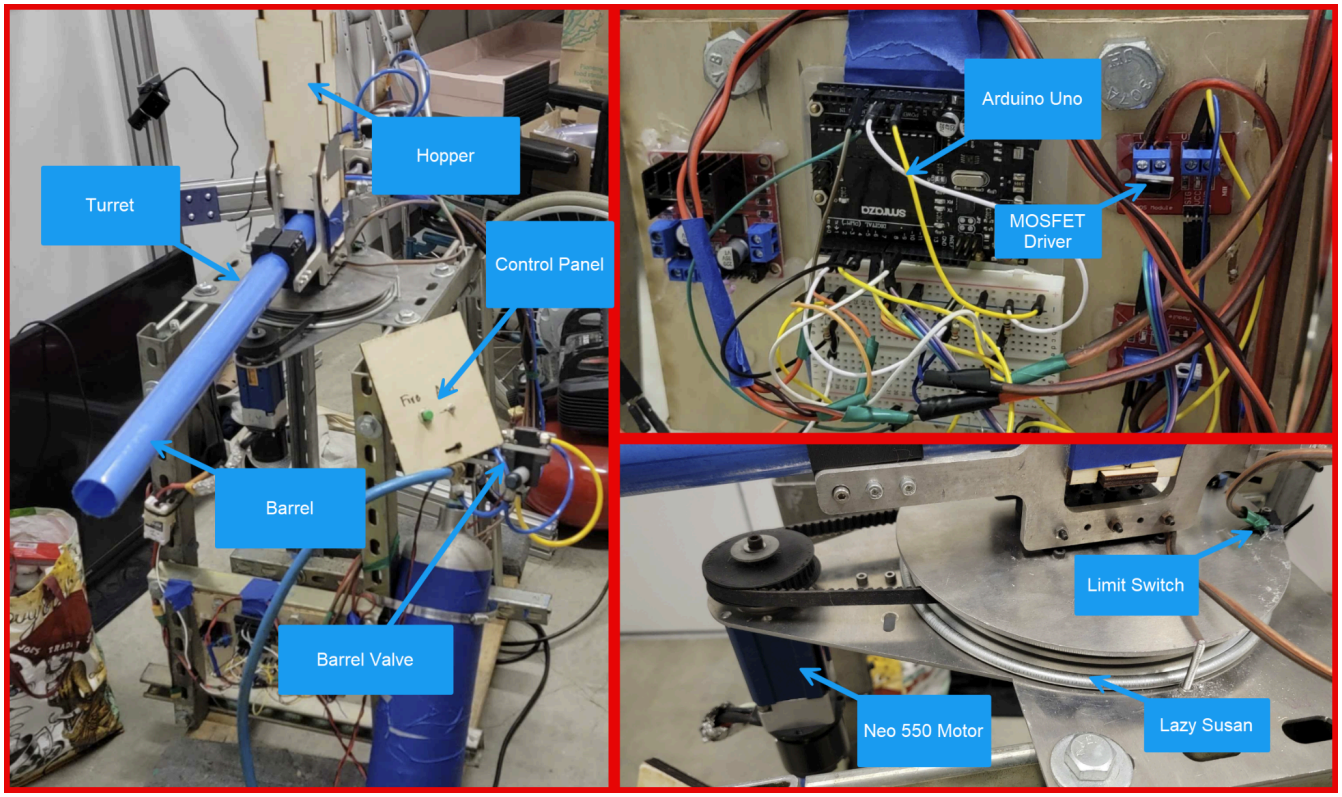
Our goal for this project is to upgrade our potato cannon, which was previously stationary with no range of motion, to track and shoot moving targets with ping pong balls for entertainment and athletic purposes. We see this project as a platform to demonstrate our extensive mechanical design experience involving pressurized systems and mechatronics while producing an exciting final result.

High Level Strategy - Realized vs. Desired

Our high level strategy involves mounting the launcher to a rotating turret that can sweep, aim, and launch in a wide field of view. This turret can be controlled by a motor and belt drive for variable turret speeds. To satisfy the project requirements, we will use a potentiometer to control the motion of the turret and a button to control the ping pong ball launcher.

In our initial desired functionality, we targeted the following capabilities: a traversal of 180 degrees in 1 second (or 30 RPM), a minimum range of 1 meter with 80% accuracy, and an increased hopper capacity of 12x ping pong balls compared to the previous 6-potato drum. Our realized product was able to meet our design requirements with a 30.05 RPM turret speed, an ability to consistently shoot a salvo of 12+ ping pong balls all into a 2x2 ft box at a 2 meter distance, and a 24 ball hopper capacity. The turret speed, however, was throttled down for safety reasons during the demo as well as some observed brownout behavior. We also fleshed out other relevant parameters, including a 150 degree range of motion and a firing rate of every 725 ms.

Labeled Photos



Function-Critical Decisions and Calculations

Turret Speed

For our turret speed, we had a NEO brushless motor and a 100:1 planetary gearbox on-hand for project use. To gear this down to a desired turret speed of 30 RPM, we used a 68:36 reduction (or 1.89:1).

$$30.05\ RPM\ =\ 5676\ RPM\ free\ speed\ *\ \frac{1}{100}\ planetary\ reduction\ *\ \frac{36}{68}\ belt\ reduction$$

Belt Tension

To determine the maximum belt tension and required preload, the following calculations were made using the maximum torque rating of the 100:1 planetary gearbox.

$$F_1\ =\ F_{1,i}\ -\ \frac{T}{D}\ =\ 0\ (no\ slack\ or\ tension)$$

$$F_{1,i}\ =\ F_{2,i}\ =\ 140\ lb\ in\ *\ \frac{\pi}{5\ mm\ pitch\ *\ 36\ teeth}\ *\ \frac{25.4\ mm}{1\ in}\ =\ 62.1\ lbs$$

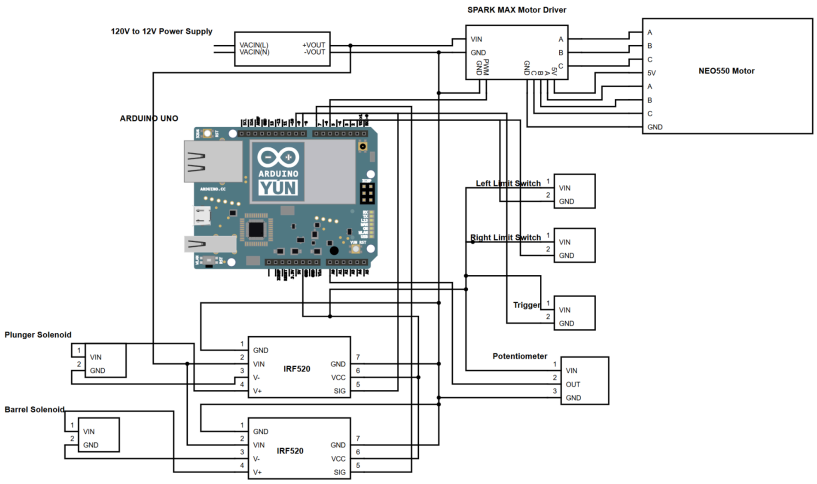
$$F_{preload}\ =F_{1,i}\ +\ F_{2,i}\ =\ 124.1\ lbs$$

It is important to note here that this is the maximum possible preload needed. Since our turret is expected to be free-spinning and unloaded aside from its own (fairly light) weight, it is unlikely the belt will actually see this much tension. Additionally, the breaking strength of our selected belt component is still far above this maximum expected belt tension–5mm HTD belts have a breaking strength of 316 N per 1 mm of width. For a 9 mm width belt, this would be 2844 N, or 639 lbs.

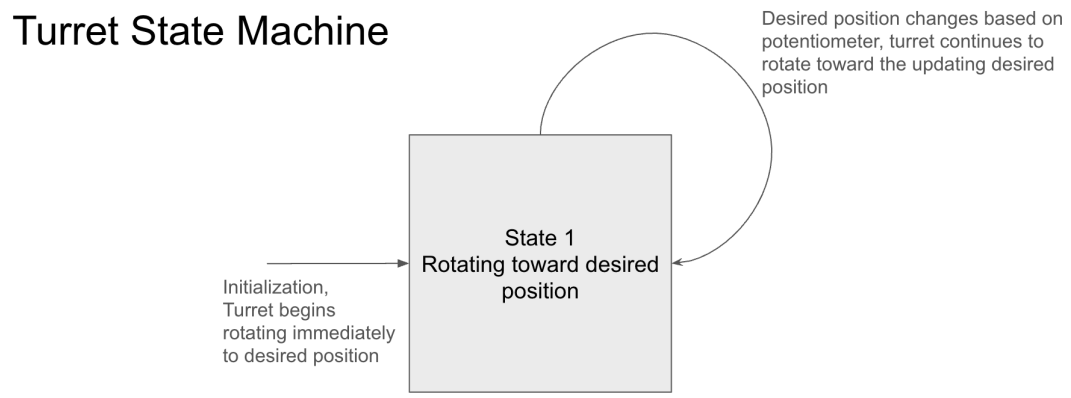
Lazy Susan Load

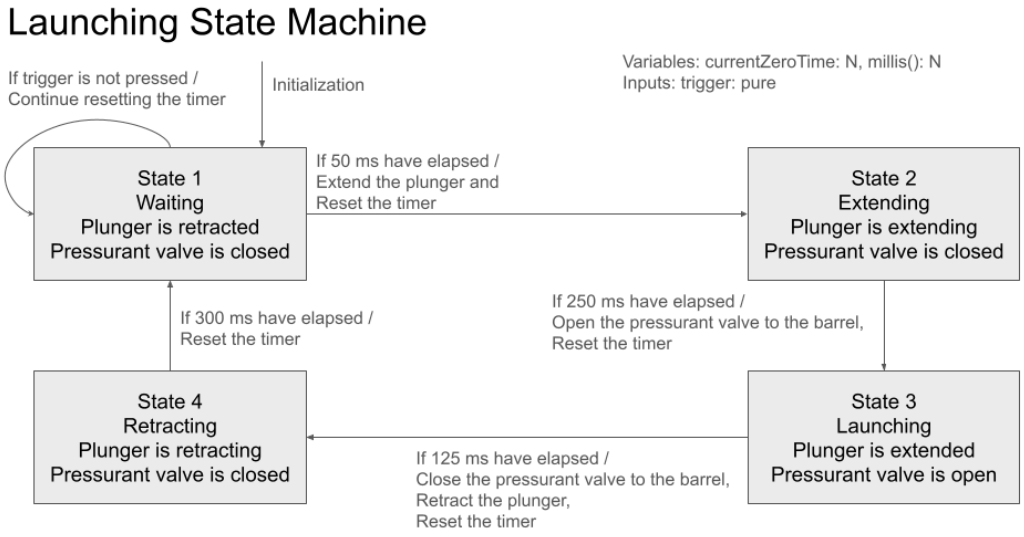
Per the product page, the maximum weight the lazy susan can support is 750 lbs, and our launcher assembly is well under 50 lbs (it is easy for a single person to pick up and handle). Since the weight is more than an order of magnitude less than the rated load, we are confident in assuming our selected lazy susan is sufficient for this application.

Circuit Diagram



State Transition Diagram





Reflection

Both our successes and shortcomings point toward the importance of getting physical hardware into our hands to test and tinker with as early as possible. This was really important for us when dialing in the performance of the ping pong ball launcher. For example, shimming the inside of the barrel to fit the ball more snugly could only have been prompted by hands-on, qualitative observation. We also felt the effects of not having parts on hand earlier in the semester. Getting around to ordering parts much later in the semester meant that we had to improvise less elegant workarounds when something wouldn't behave as expected.

Appendices

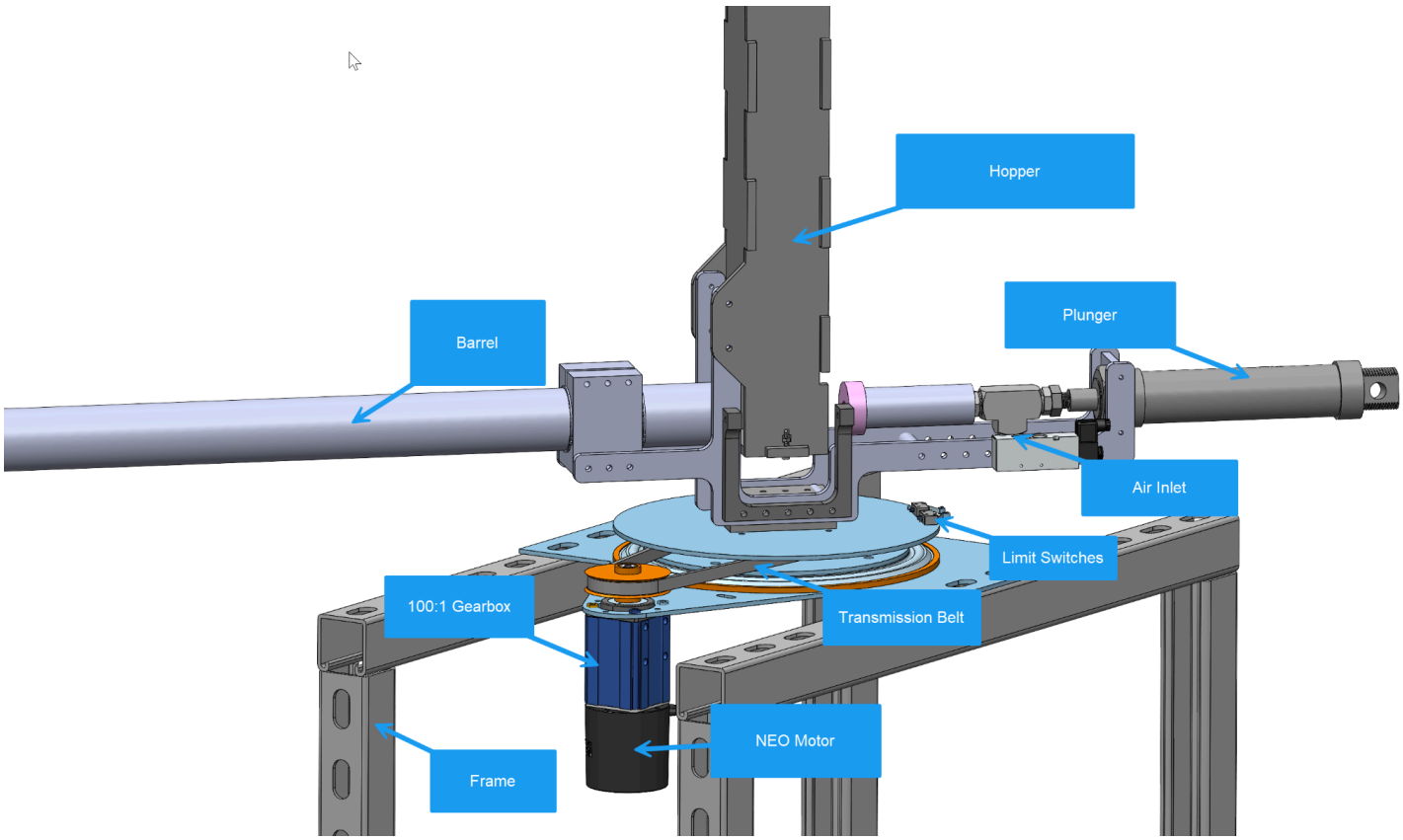
Bill of Materials

Sub Assembly	Component	Quantity	Cost	Link
Electronics	Arduino UNO	1	0	Personal Member Inventory
Electronics	IRF520 MOSFET	2	0	Previous Project
Electronics	SPARK MAX Motor Driver	1	0	Personal Member Inventory
Electronics	Limit Switch	2	0	Personal Member Inventory
Electronics	Power Supply	1	0	Personal Member Inventory
Electronics	Trigger Button	1	0	Personal Member Inventory
Electronics	Potentiometer	1	0	Personal Member Inventory
Ping Pong Ball Shooter	Pressurant Valve	1	0	Previous Project
Ping Pong Ball Shooter	Pressurant Tank	1	0	Previous Project
Ping Pong Ball Shooter	Pressurant Hoses	1	0	Previous Project
Ping Pong Ball Shooter	5 Way Solenoids	2	0	Previous Project
Ping Pong Ball Shooter	Pneumatic Hose	15 ft	0	Previous Project
Tripod	M4 Hardware	100+	0	Personal Member Inventory
Tripod	3ft Strut Channel	4	31.02	https://www.mcmaster.com/3310T57-3310T54/
Tripod	2ft Strut Channel	3	20.68	https://www.mcmaster.com/3310T57-3310T53/

Tripod	Pressurant Bottle Clamp	2	13.47	https://www.mcmaster.com/3166T52/
Turret	Lazy Susan	1	19.06	https://www.mcmaster.com/6031K21/
Turret	68t Pulley	1	0	Personal 3D-Printer
Turret	129t 5mm HTD Belt	1	25.32	https://www.mcmaster.com/7939K551
Turret	1/4" ID Bearing	3	20.37 (3x 6.79)	https://www.mcmaster.com/60355K151/
Turret	¼-20 Bolt	1	8.58 (25x pack)	https://www.mcmaster.com/92949A546/
Turret	¼-20 Washer	1	5.50 (100x pack)	https://www.mcmaster.com/92141A029/
Turret	¼-20 Locknut	1	4.49 (25x pack)	https://www.mcmaster.com/90630A110/
Turret	Shaft Spacer	1	0	Personal 3D-Printer
Turret	Base Plate	1	10	Jacobs Material + Waterjet
Turret	36t Pulley	1	0	Personal 3D-Printer
Turret	NEO Motor	1	48	https://www.revrobotics.com/rev-21-1650/
Turret	Planetary Gearbox	1	94	https://banebots.com/psh-455/
Turret	Turret Bottom Plate	1	5	Jacobs Material + Waterjet
Turret	Turret Top Plate	1	5	Jacobs Material + Waterjet
Turret	Mounting Block	1	0	Personal 3D-Printer
Turret	M4 x 0.7 Bolt	3	7.33 (10x pack)	https://www.mcmaster.com/97763A817/

Turret	M4 x 0.7 Washer	3	3.13 (100x pack)	https://www.mcmaster.com/98269A430/
Hopper	M3 x 0.5, 16mm Length Bolt	1	11.94 (100x pack)	https://www.mcmaster.com/91290A120/
Hopper	M3 x 0.5 Nut	1	2.81 (100x pack)	https://www.mcmaster.com/90591A250/
Hopper	Hopper Housing Plates	1	28.30	Jacobs Stock ¼ Plywood + Laser Cutter https://store.jacobshall.org/products/plywood-1-4-x-24-x-48

CAD



Code

```
#include <Button.h>
#include <Servo.h>

enum State {
  WAITING,
  EXTENDING,
  SHOOTING,
  RETRACTING,

  ROTATING
};


enum State shootingState = WAITING;
enum State turretState = ROTATING;

int barrel_limit_switch = 13;
unsigned long currentZeroTime;
int triggerPin = 9;

Servo motor;
int motorPin = 5;
int leftLimit = 2;
int rightLimit = 3;
int potPin = A0;

int barrelA = 8;
int plungerA = 7;

volatile bool leftLimitPressedFlag = false;
volatile bool rightLimitPressedFlag = false;

void setup() {
  motor.attach(motorPin);
  pinMode(plungerA, OUTPUT);
  pinMode(barrelA, OUTPUT);
  pinMode(barrel_limit_switch, INPUT);
  pinMode(potPin, INPUT);
  pinMode(leftLimit, INPUT);
  pinMode(rightLimit, INPUT);
```

```
  attachInterrupt(digitalPinToInterrupt(leftLimit), leftLimitPressed, CHANGE);
  attachInterrupt(digitalPinToInterrupt(rightLimit), rightLimitPressed, CHANGE);

  pinMode(triggerPin, INPUT);
  currentZeroTime = millis();
  Serial.begin(9600);

}
void loop() {
  switch (turretState) {
    case ROTATING:
      if (leftLimitPressedFlag) {
        //Serial.println("Left Limit");
        if (readPotentiometer() > 0) {
          turn_motor(readPotentiometer());
        }
        else {
          turn_motor(0);
        }
      }
      else if (rightLimitPressedFlag) {
        //Serial.println("Right Limit");
        if (readPotentiometer() < 0) {
          turn_motor(readPotentiometer());
        }
        else {
          turn_motor(0);
        }
      }
      else {
        turn_motor(readPotentiometer());
      }
      break;
  }
```

```
  switch (shootingState) {
    case WAITING:
      if (!trigger()) {
        currentZeroTime = millis();
      }
      if (millis() - currentZeroTime > 50) {
        shootingState = EXTENDING;
        extend_plunger();
        currentZeroTime = millis();
      }
    break;
    case EXTENDING:
      if (millis() - currentZeroTime > 250) {
        shootingState = SHOOTING;
        open_barrel();
        currentZeroTime = millis();
      }
    break;
    case SHOOTING:
      if (millis() - currentZeroTime > 125) {
        shootingState = RETRACTING;
        close_barrel();
        retract_plunger();
        currentZeroTime = millis();
      }
    break;
    case RETRACTING:
      if (millis() - currentZeroTime > 300) {
        shootingState = WAITING;
        currentZeroTime = millis();
      }
    break;
  }
}
```

```
void turn_motor(float speed) {
  //Serial.println(speed);
  // turn motor at speed
  if (speed < -100) {
    speed = -100;
  }
  if (speed > 100) {
    speed = 100;
  }
  int val = 90+(speed)/100.0*90.0+3;
  //Serial.println(val);
  motor.write(val);
}

int max_speed = 20;
float readPotentiometer() {
    int val = analogRead(potPin);
    return (val * max_speed/1024.0) - max_speed/2.0;
}

bool barrelLimitSwitch() {
  return !digitalRead(barrel_limit_switch);
}
bool trigger() {
  return digitalRead(triggerPin);
}
```

```
void leftLimitPressed() {
  leftLimitPressedFlag = digitalRead(leftLimit);
}

void rightLimitPressed() {
  rightLimitPressedFlag = digitalRead(rightLimit);
}


void extend_plunger() {
  digitalWrite(plungerA, HIGH);
  //Serial.println("Extending Plunger");
}
void retract_plunger() {
  digitalWrite(plungerA, LOW);
  //Serial.println("Retracting Plunger");
}
void open_barrel() {
  digitalWrite(barrelA, HIGH);
  //Serial.println("Opening Barrel");
}
void close_barrel() {
  digitalWrite(barrelA, LOW);
  //Serial.println("Closing Barrel");
}
```