Autonomous Cat Toy

Alan Monge, Austin Kim, Danny Kwon, Marcos Marmolejo

Project Opportunity:

Our envisioned project opportunity was to use a robotic arm to perform some simple task for the user and make their life easier, and to incorporate elements from both lab and lecture. Our autonomous cat toy accomplishes this by providing entertainment for a cat when the owner either does not have enough time to play with their pet or lacks the physical capabilities of being able to play with their pet.

Integrated Physical Device:

Note: Servo #2 was only intended to hold the robot shoulder in place.

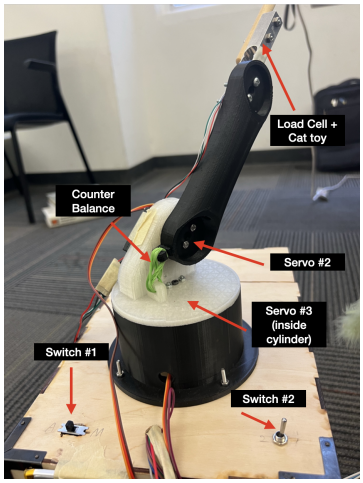


Figure 1

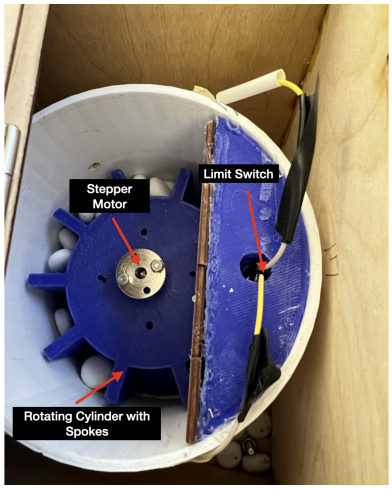


Figure 2

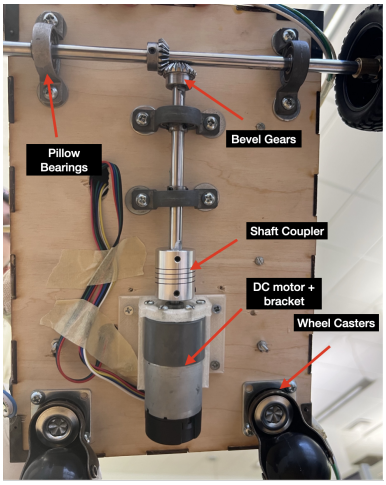


Figure 3

High-Level Strategy:

Our cat toy consists of two subsystems. One subsystem is concerned with the motion of the cat toy and consists of a servo motor driving the robotic arm and the DC motor used to drive the wheels on the bottom of the cat toy back and forth. The second subsystem involves the dispensing of treats and uses a load cell to detect when the cat has successfully swiped at the cat toy.

Our first subsystem can be broken down into three main states, idling, where neither the DC motor nor the servo motor are being driven and the toy is stationary. The second state is manual mode where the toy takes inputs from an IR remote and sensor on our breadboard to toggle the servo to perform a sweeping motion, drive the DC motor forward or backwards, both with the use of PI control, and stop all of the previous actions. The third state is automatic mode where we have two presets, one for driving our DC motor forward for 1 second, and back for 1 second repeatedly, and another preset for having the servo motor perform a sweeping motion. While our toy is in any of the three states we are taking readings from the load cell to determine if any treats should be dispensed or not.

Our second subsystem consists of a load cell to detect force from a cat, a stepper motor used to drive a cylindrical piece with spokes on it that is used to dispense treats and a limit switch to detect when the stepper motor has rotated a certain amount and thus dispensed a certain amount of treats. After our load cell threshold is reached we set a delay for five seconds to stop taking readings from the load cell in order for the toy to run smoother and not dispense too much food.

Our end product functionality differs quite drastically from our initial vision for our project as originally we had planned on creating a robotic arm periscope that can be controlled by an app and

provide precise movements however, now we have a product that is erratic and entertaining enough, which we defined as having the system move at over $1.0m/s$ and have an arm sweeping on top for a cat that consistently dispenses 3-4 treats when a cat tugs on the toy at the end of the arm.

Functional Critical Decisions and Calculations:

1.3mm washers were used to properly align the shafts to the DC motor to prevent any damages to the whole system. Loads on bearings were also calculated, only considering rolling resistance and μ = .02:

$$F = \mu mg = .02(3kg)(9.81m/s^2) = .5886N$$
$$F_{wheel} = F/4 = .147N$$
$$\Sigma M = 0 = (-.025m)z \times (-F_{wheel})x + (.05m)z \times (F_B) + (-.075m)z \times (-F_{wheel})x$$
$$= .01482Nmy + .05F_Bx - .05F_By$$
$$F_Bx = 0$$
$$F_By = .2965N$$
$$F_Ay = -F_By = -.2965N$$
$$F_Ax = F_Bx = 0$$

Repeating these calculations for the other bearings reveals that the maximum radial load we will experience is less than 1lbf which is far under the radial load capacity of bearings which is 300lbs.

We wanted robot to move at $1.0m/s$ because it is slightly slower than average walking speed of a human. Since we chose a driving wheel whose diameter is 76mm, RPM of the wheel shaft should be approximately 250RPM (1). We searched for a motor with reducer first, and would use a gear transmission later for insufficient gear ratio if necessary. Sum of estimated weight of the robot from CAD and weights of electronical parts was approximately $3kg$. assuming there is no slip between the wheel and ground, required torque should be $5.7kg \cdot cm$ (2).

In consideration of these two design factors, we could find a motor with enough torque and desired RPM. According to a catalog of the motor we found, its speed is 251RPM and stall torque is $18kg \cdot cm$ (safety factor ≈ 3) at rated voltage.

$$(76 \times 10^{-3}m)(x\,RPM)(\tfrac{\pi}{60}) = 1.0m/s,\ x \approx 250(RPM) \quad \cdots \quad (1)$$

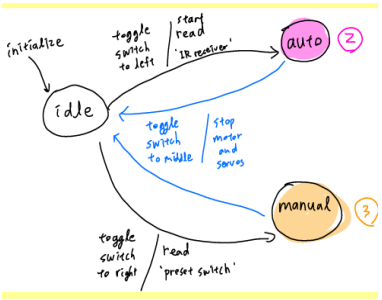
$$\tau_{wheel} = \tau_{motor}\ ,\ \mu = 0.5$$

$$(\tau_{wheel}kg \cdot m) \div (0.5 \cdot 76 \cdot 10^{-3}m) \cdot (9.81\tfrac{Nm}{kgm}) \geq (3kg)(9.81m/s^2)(0.5),\ \tau_{motor} \approx 5.7kg \cdot cm \quad \cdots \quad (2)$$

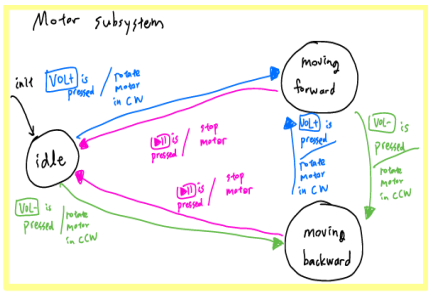
One stepper motor was used to rotate the wheel part inside of the dispenser.(figure 2). We lifted the wheel part a little up to eliminate the contacts between the wheel and the bottom part of the dispenser as a method for overload protection. The total weight of the wheel part is calculated to be 197.5596 grams. (The total volume of the wheel part is 158.04777 cm^3 and the density of PLA is 1.25 $\frac{g}{cm^3}$). We then assumed the wheel part as a round plate. (The inertia of a round plate is $I = m * \frac{r^2}{2}$). The radius of the wheel part was estimated as 42 mm. Therefore, the inertia of the wheel part was calculated to be 1.7424* $kg * cm^2$. The stepper motor has 4.28 $kg * cm$ of torque at low speeds. According to the torque equation $\tau = I * a$, The angular acceleration should be lower than 2.45 $\frac{radian}{s^2}$ to satisfy the torque limit.

State Transition Diagram:

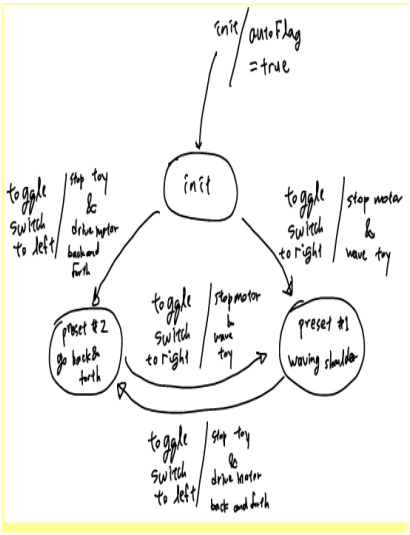
Main

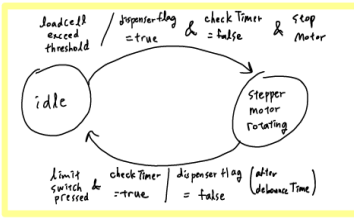


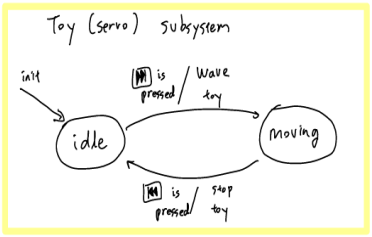
Manual



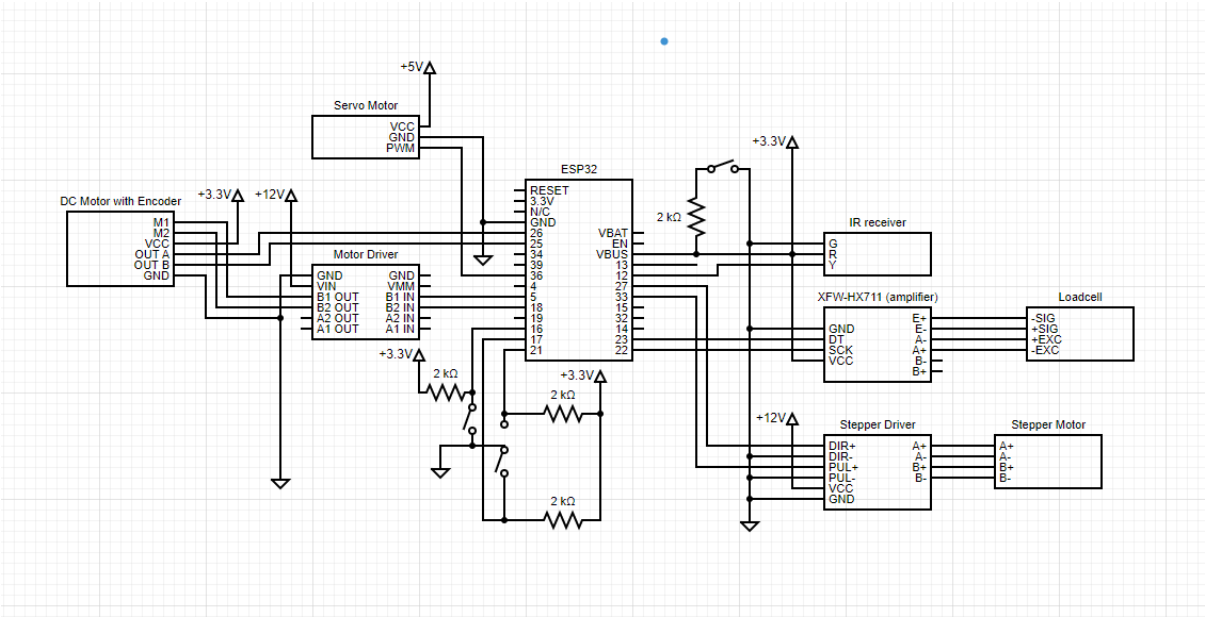
Auto







Circuit Diagram:



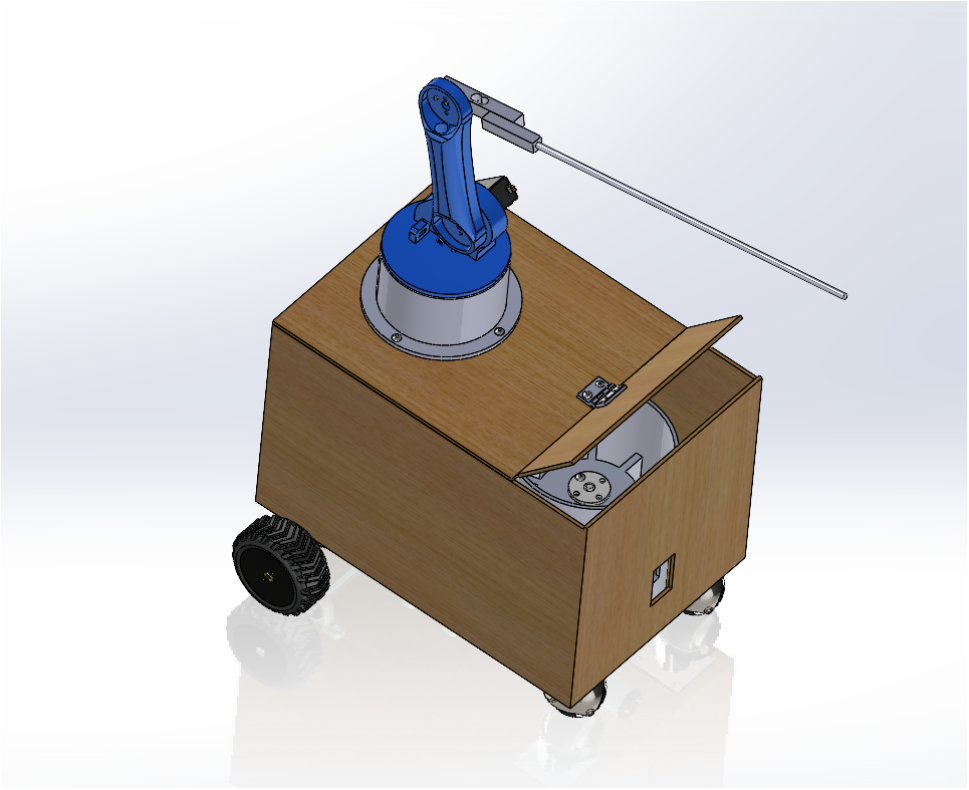
Reflection

Team meetings and communication are incredibly valuable and critical to the success of your project. One thing we implemented that worked well was having a team meeting every week of the semester but one thing that could have made this smoother was making an agenda for each meeting of things that needed to get done so group members could easily pick a task to do and have a better sense of direction. Making sure everyone is on the same page and is invested is crucial to having a smooth project experience. Finally, the last thing we believe we could have done better is to be more ambitious. This class teaches a lot of things and is a great opportunity to showcase all of the years of classes we have taken and design and build something impressive.

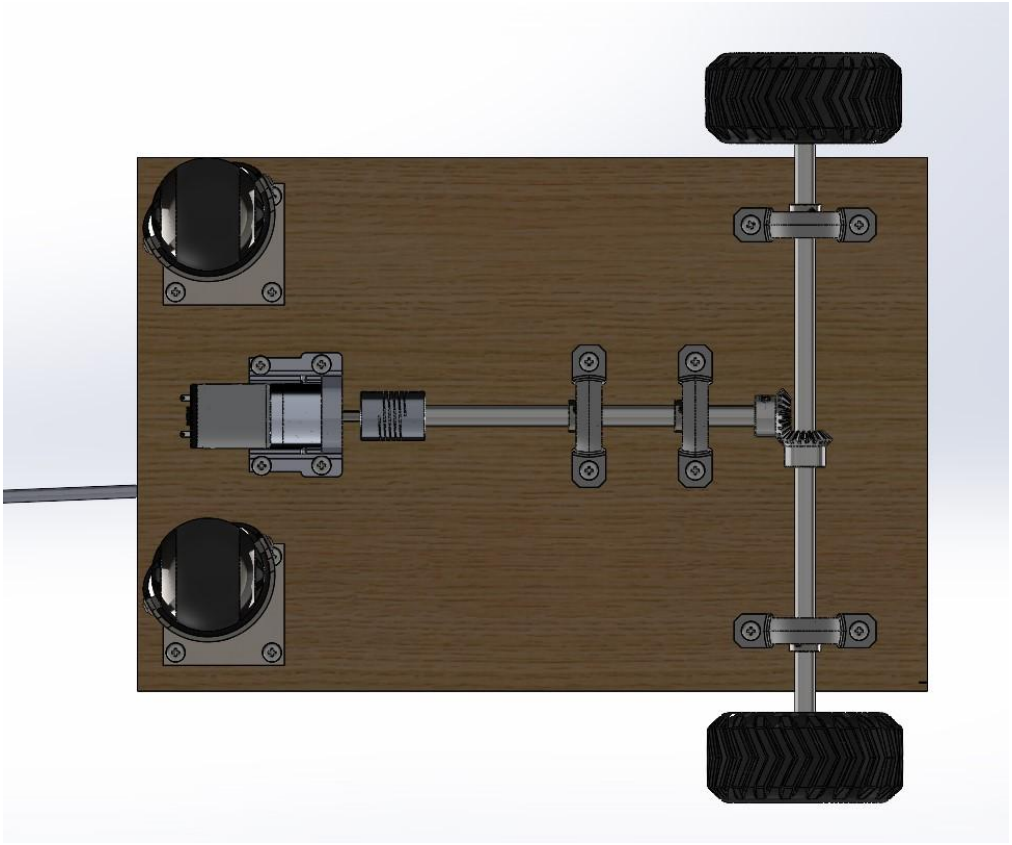
Appendix

CAD Designs:

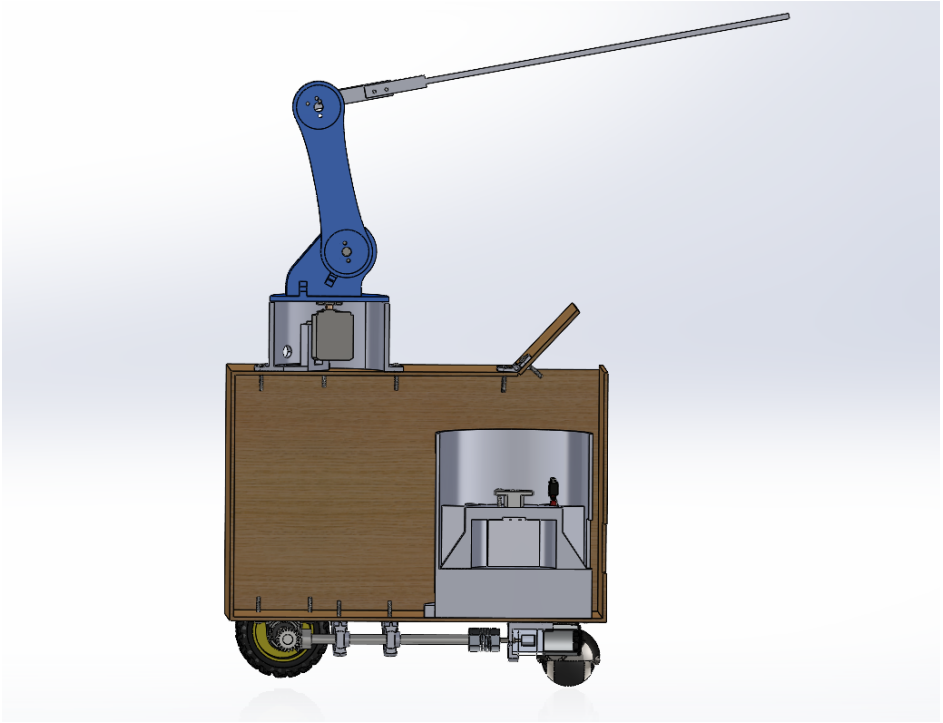
Front view



Isometric view



Bottom view showing the transmission parts



The section view

Bill of Materials

Part(s)	Quantity	Cost ($)	Source Link
Plywood - 1/8" x 18" x 30"	3	$6.66	https://store.jacobshall.org/products/plywood-1-8-x-18-x-30
Stainless Steel Folding Hinge 2 inch Door and Window Hinge is assigned 120 Screws	1	$11.97	https://www.amazon.com/Liberty-20Pcs-Stainless-Folding-Window-assigned/dp/B088KBZ9C6/ref=sr_1_6?crid=245YPOWGRC9C5&keywords=hinges&qid=1670742368&sprefix=hinge%2Caps%2C147&sr=8-6
uxcell 33x33x33mm Metal Box Corner Protectors Edge Guard Black 4pcs	1 (4pcs)	$9.49	https://www.amazon.com/dp/B09MYCYRNT/ref=sspa_dk_detail_0?psc=1&pd_rd_i=B09MYCYRNT&pd_rd_w=fRjhT&content-id=amzn1.sym.4d0fffec-3aba-4480-8fad-c6bd8f7f6b41&pf_rd_p=4d0fffec-3aba-4480-8fad-c6bd8f7f6b41&pf_rd_r=CYQ35J0X7VNWXM794AA5&pd_rd_wg=Ur23Z&pd_rd_r=5e31ccbd-e698-466b-bf86-ff6b30b09366&s=hi&sp_csd=d2lkZ2V0TmFtZT1zcF9kZXRhaWxfdGhlbWF0aWM&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExT1VMV1pTNENDMkNTJmVuY3J5cHRlZElkPUEwOTM3ODUwMlJRVVU2OFVXNlRaRiZlbmNyeXB0ZWRBZElkPUEwNzI2NDE2MlRIOU1IWTVaWFI5UCZ3aWRnZXROYW1lPXNwX2RldGFpbF90aGVtYXRpYyZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=
Robot arm, joints and base (PLA)	1	$6.50	Jacobs Self Printing Service
4 Pack Metal Gear Torque Digital Servo with Arm Horn, MELIFE Robot Servo Motor for FUTABA Hitec Sanwa GWS JR RC	1 (4pack)	$17.99	https://www.amazon.com/MG996R-Torque-Digital-MELIFE-Helicopter/dp/B09BQP2F6M/ref=sr_1_1?crid=UT14NJ9MGV7U&keywords=4+pack+metal+gear+torque+digital+servo+with+arm+horn&qid=1665450637&sprefix=4+pack+metal+gear+torque+digital+servo+with+arm+horn%2Caps%2C134&sr=8-1

Helicopter Car Boat Robot			
Load Cell	1	$0	-
Load Cell Amplifier	1	$0	-
Cat Toy	1	$6.99	Target
AllinRC 12mm Hex RC Tires & Wheel Rims Foam Inserts Height 2.99 inch / 76mm	2(4 pack)	$16.17	https://www.amazon.com/AllinRC-Height-Rubber-Tires-Inserts/dp/B09JSWVQJ2/ref=sr_1_3?crid=3B2DUPDW1WEEM&keywords=AllinRC+12mm+Hex+RC+Tires&qid=1665451863&sprefix=allinrc+12mm+hex+rc+tires%2Caps%2C121&sr=8-3
Motor Bracket (PLA)	1	$3	Jacobs Self Printing Service
Dispenser Wheel	1	$4.24	Jacobs Self Printing Service
Disperser	1	$7	Jacobs Self Printing Service
Twotrees Stepper Motor Nema 17 Motor High Torque 1.5A (17HS4401) 42N.cm (60oz.in) 1.8 Degree 38MM 4-Lead with 1m Cable and Connector for 3D Printer	1	$10.99	https://www.amazon.com/Twotrees-Stepper-17HS4401-Connector-Printer/dp/B07THK76QQ/ref=sr_1_3?crid=4B8DYGUQ4VO7&keywords=twotrees+nema+17+stepper+motor+high+torque+1.5A&qid=1670739268&s=industrial&sprefix=twotrees+nema+17+stepper+motor+high+torque+1.5%2Cindustrial%2C127&sr=1-3
EASON Stepper Motor Driver TB6600 4A DC9-42V 32 Subdivision for Nema 17, 23 Series Stepper Motor CNC Controller Single Axis Hybrid MicroStepping	1	$18.69	https://www.amazon.com/Stepper-DC9-42V-Subdivision-Controller-MicroStepping/dp/B08SG7L54W/ref=sr_1_1_sspa?crid=3NZY7GS2IVPA4&keywords=twotrees+nema+17+stepper+motor&qid=1670739236&s=industrial&sprefix=twotrees+%2Cindustrial%2C130&sr=1-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEyQ1dRNTFCV08yWDcmZW5jcnlwdGVkSWQ9QTA3NDQyNzczSDlES1RRVldNWk1RJmVuY3J5cHRlZEFkSWQ9QTAyODc0MzUzTkQ1UE80QjZYRjJDJndpZGdldE5hbWU9c3BfYXRmJmFjdGlvbj1jbGlja1JlZGlyZWN0JmRvTm90TG9nQ2xpY2s9dHJ1ZQ==
MUZHI SPDT 1NO	1 (12 pcs)	$5.99	https://www.amazon.com/MXRS-Hinge-Momentary-

1NC Hinge Lever Momentary Push Button Micro Limit Switch AC 5A 125V 250V 3 Pins 12 Pcs			Button-Switch/dp/B07MW2RPJY/ref=sr_1_4?keywords=limit+switch&qid=1670739808&sr=8-4
2PCS 8mmX250mm Linear Motion Rod Shaft Guide Diameter 8mm for 3D Printer and DIY Craft Tool	1 (2 pack)	$11.05	**https://www.amazon.com/8mmX350mm-Linear-Motion-Shaft-Diameter/dp/B08HYKCJD5/ref=sr_1_6?crid=29L9QI5BFWJ71&keywords=8mm%2Bd%2Bprofile%2Bshaft&qid=1665707556&sprefix=8mm%2Bd%2Bprofile%2Bshaft%2Caps%2C145&sr=8-6&th=1**
2PCS 5mm to 10mm Flexible Couplings 25mm Length 19mm Diameter Shaft Couplings for 3D Printer and CNC Machine	1	$8.99	https://www.amazon.com/Flexible-Couplings-Diameter-Printer-Machine/dp/B09DSQCD3V/ref=sr_1_2_sspa?crid=BMJE68PE389Y&keywords=8mm+shaft+coupler&qid=1670742459&sprefix=8mm+shaft+couple%2Caps%2C132&sr=8-2-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExWFlSSFVPRzVUMDFVJmVuY3J5cHRlZElkPUExMDM3MjUxMTlMWVZERkJIT05WWCZlbmNyeXB0ZWRBZElkPUEwNDE3MTgwM0NXTUMwVkRZTkEyWiZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=
Hobbypark 8mm Thickness Brass 12mm Hex Wheel Hubs Adapters for Traxxas TRX-4 Axial SCX10 RC4WD D90 RC 1/10 Crawler (8mm)	1	$8.98	https://www.amazon.com/Hobbypark-Adapters-Traxxas-Redcat-Crawler/dp/B07TXYQDCC/ref=sr_1_6?crid=31MEYHZCD67UI&keywords=8mm+wheel+hub&qid=1670738480&sprefix=8mm+wheel+hub%2Caps%2C134&sr=8-6
PGFUN 1:1 Right Angle Bevel Gearbox 1 Module 20 Teeth 90° Angle Drive Steering Gear Device Simple Mechanical DIY Module with 8mm Shaft	**1**	$36.99	https://www.amazon.com/PGFUN-Gearbox-Module-Steering-Mechanical/dp/B09V5SCSMP/ref=sr_1_1_sspa?crid=2RDJO5SWC4W12&keywords=8mm+bevel+gear&qid=1670738673&sprefix=8mm+bevel+gear%2Caps%2C119&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzN1BLMTk0WTJVV1ZDJmVuY3J5cHRlZElkPUEwMDMwNzk0MzJESEVIVVJZMlJNNiZlbmNyeXB0ZWRBZElkPUEwOTE0MTY5MUY0Tk0wNk9NMzdNOSZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=
DC Motor Gearmotor 251 RPM Incremental 12VDC	1	$29.00	https://www.digikey.com/en/products/detail/dfrobot/FIT0186/6588528?utm_adgroup=Motors%20-%20AC%2C%20DC&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_Product_Motors%2C%20Solenoids%2C%20Driver%20Boards%2FModules_NE

			W&utm_term=&utm_content=Motors%20-%20AC%2C%20DC&gclid=Cj0KCQiAnNacBhDvARIsABnDa68kU96Q-LMax6Qpjsm28lwPEfyvKMI_2tvm1Q8G-sHqymf0NKi0X2YaApcsEALw_wcB
2 Inch Ball Caster Wheels for Furniture, Antique Furniture Casters Brass, Vintage Caster Wheels Heavy Duty, Swivel Plate Ball Casters Set of 4, Furniture Wheels for Table Dresser Cabinet	1	$14.99	https://www.amazon.com/JGJC-Furniture-Antique-Cabinet-Mounting/dp/B083R71RWB/ref=sr_1_4?crid=WWZ4HGV7ZQ2K&keywords=ball+casters&qid=1670738935&s=industrial&sprefix=ball+casters%2Cindustrial%2C130&sr=1-4
Sydien 4Pcs 8mm Inner Bore Ball Mounted Pillow Block Flange Micro Vertical Bearing	1	$8.99	https://www.amazon.com/Sydien-Pillow-Block-Flange-Bearing/dp/B07C5MTB5X/ref=sr_1_2?crid=1C4WPF7CMY7HA&keywords=sydien+4+pcs+8mm+inner+bore+pillow+bearings&qid=1670739085&s=industrial&sprefix=sydien+4+pcs+8mm+inner+bore+pillo+wbearing%2Cindustrial%2C123&sr=1-2
Trivd Frol 750PCS #2-56#4-40#6-32#8-32 Phillips Pan Head Screws Bolt Nut Flat Washers 304 Stainless Steel Machine Screws Nuts Washers Assortment Kit (#2-56#4-40#6-32#8-32)	1	$20.99	https://www.amazon.com/Trivd-Frol-Phillips-Stainless-Assortment/dp/B0B9JSJK8L/ref=sr_1_1_sspa?crid=3I3084KFG22C9&keywords=m3+machine+screws+and+nuts&qid=1670741011&sprefix=m3+machine+screws+and+nut%2Caps%2C142&sr=8-1-spons&psc=1&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUExV1RMVkpZVFhJTVQ2JmVuY3J5cHRlZElkPUEwODc0ODYxMkpCS0xOUFBLMTNGRSZlbmNyeXB0ZWRBZElkPUEwNjU1Nzc3MTlMOVVUVzJQVkY2NCZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU=
M3 Nylon lock nut	2	$0.69	Ace
Chanzon 120pcs 10cm 20cm 30cm Long Header Jumper Wire Dupont Cable Line Connector Assorted Kit Set (Male Female M-M M-F F-F) Solderless Multicolor for	1	$9.99	https://www.amazon.com/Connector-Solderless-Multicolor-Electronic-Breadboard/dp/B09FPGT7JT/ref=sr_1_3?crid=VIH078EIDFY2&keywords=jumper+cables+electronics&qid=1670741290&sprefix=jumper+cables+elec%2Caps%2C137&sr=8-3

Arduino Raspberry pi Electronic Breadboard PCB			
KOOBOOK 2Sets HX1838 NEC Infrared IR Remote Control Sensor Module IR Receiver Module DIY Kit for arduino	1	$8.93	https://www.amazon.com/KOOBOOK-Infrared-Wireless-Control-Receiver/dp/B07S67SFSF/ref=asc_df_B07S67SFSF/?tag=hyprod-20&linkCode=df0&hvadid=563688069599&hvpos=&hvnetw=g&hvrand=13909991124313571729&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9032074&hvtargid=pla-1460859225564&psc=1
Resistor Kit	1	$0	-
ESP32	1	$0	-
Li Ion Polymer Battery	1	$0	-
Universal AC Adapter 15V 1A Power Supply 15W AC110V to DC15V Power Driver 15V 100mA~1000mA All Compatible with 8 DC Plug Tips adapters Charger Transformer AC-DC Adapter (DC15V~1A)	1	$9.58	https://www.amazon.com/Universal-100mA-1000mA-Compatible-adapters-Transformer/dp/B08T6275KH/ref=asc_df_B08T6275KH/?tag=hyprod-20&linkCode=df0&hvadid=507647815824&hvpos=&hvnetw=g&hvrand=12989832345561435399&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9032074&hvtargid=pla-1331706870139&psc=1
5V Power Supply	1	$0	from the labkit
Power Supply adapter	2	$0	from the labkit
Toggle switches	2	$8.41	https://www.digikey.com/en/products/detail/tubedepot/108-1MS1T2B3M1QE-EVX/10487757?utm_adgroup=Tubedepot&utm_source=google&utm_medium=cpc&utm_campaign=Shopping_DK%2BSupplier_Tier%201%20-%20Block%201&utm_term=&utm_content=Tubedepot&gclid=Cj0KCQiAnNacBhDvARIsABnDa6_efpMa3jNBNeSeSJL0mxzldjuT36AZGs0Vz2qbba02yEYmFgx03uYaAgkrEALw_wcB
DRV8835 Dual	1	$9.95	https://www.pololu.com/product/2135

Motor Driver Carrier			
Slide Switch SPDT Panel Mount	1	$5.14	https://www.digikey.com/en/products/detail/nkk-switches/MS13AFG01/1052492
TUOFENG 22 awg Wire Solid Core Hookup Wires-6 Different Colored Jumper Wire 30ft or 9m Each, 22 Gauge Tinned Copper Wire PVC (OD: 1.55mm) Hook up Wire Kit	1	$5.59	https://www.amazon.com/TUOFENG-Hookup-Wires-6-Different-Colored/dp/B07TX6BX47/ref=asc_df_B07TX6BX47/?tag=hyprod-20&linkCode=df0&hvadid=366430786295&hvpos=&hvnetw=g&hvrand=5885886009363271769&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9032082&hvtargid=pla-793870383734&psc=1&tag=&ref=&adgrpid=75985294733&hvpone=&hvptwo=&hvadid=366430786295&hvpos=&hvnetw=g&hvrand=5885886009363271769&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9032082&hvtargid=pla-793870383734

Code:

```
/*
 * UC Berkeley ME 102B: Mechatronics Fall 2022
 * Prof. Hannah Stuart
 * Team 17: A.D.A.M.S
 * Mobile Cat Toy Feeder Robot Project
 */

//----------------- libraries --------------------//
#include <ESP32Servo.h>
#include "IRremote.h"
#include "HX711.h"
#include <ESP32Encoder.h>
//----------------- flags --------------------//
// check timer interrupt
volatile bool interruptCounter = true;
volatile bool autoForward = true;
volatile bool autoFlag = false;
volatile bool dirElbow = true;
volatile bool dirShoulder = true;
volatile bool started = false;
volatile bool toggle = false;
volatile bool dispense = false;
volatile bool debounce = false;
volatile bool check = true;
// counts the number of triggering of the alarm
int totalInterrupts;
//hw_timer_t * timer = NULL;
//portMUX_TYPE timerMux0 = portMUX_INITIALIZER_UNLOCKED;

hw_timer_t * autoTimer = NULL;
portMUX_TYPE timerMux1 = portMUX_INITIALIZER_UNLOCKED;

hw_timer_t * debounceTimer = NULL;
portMUX_TYPE timerMux2 = portMUX_INITIALIZER_UNLOCKED;

hw_timer_t * checkTimer = NULL;
portMUX_TYPE timerMux3 = portMUX_INITIALIZER_UNLOCKED;
//----------------- global variables --------------------//
Servo joint1;
Servo joint2;


int shoulder_pos = 90;
int elbow_pos = 0;

const int freq = 5000;
const int ledChannel_1 = 1;
const int ledChannel_2 = 2;
const int resolution = 12;
int autoPWM = 4095;
float loadCellReading;
const float pthresh = 22000.0;
const float nthresh = -22000.0;
int omegaSpeed = 0;
int omegaDes = 13;
int omegaMax = 18;
int D = 0;
int i_integral = 0;
int Kp = 50;
int Ki = 100;
int IMax = 50;
//----------------- pins --------------------//
const byte modeSwitchA = 16;
const byte modeSwitchB = 19;
const byte presetSwtich = 17;
const byte JOINT_1 = 12;
const byte JOINT_2 = 25;
const byte irReceivePin = 27;
const byte BIN_1 = 4;
const byte BIN_2 = 5;
const byte hx711_data_pin = 22;
const byte hx711_clock_pin = 14;
const byte dirPin = 15;
const byte stepPin = 33;
const byte limit_switch = 23;
//----------------- forward declaration --------------------//
void IRAM_ATTR onTime();
void IRAM_ATTR autoTime();
void stopShoulder();
void waveShoulder();
//void stopElbow();
void waveElbow();
void motorBackward();
void motorForward();
void stopMotor();
byte presetSwtichRead();
byte remoteControllerChecker();
byte modeSwitchRead();
void stepOnce();
void checkLoadCell();
void switchTriggered();
void dispenseFood();
void debounceSwitch();
void checkFlag();
//Q2HX711 hx711(hx711_data_pin, hx711_clock_pin);
//----------------- Main functions --------------------//
HX711 scale;

void setup() {
  Serial.begin(115200);
  scale.begin(hx711_data_pin, hx711_clock_pin);
  scale.tare();
      // switches for mode
  //attachInterrupt(hx711_data_pin, RISING,checkLoadCell);
  pinMode(modeSwitchA, INPUT);
  pinMode(modeSwitchB, INPUT);
  pinMode(BIN_1,OUTPUT);
  pinMode(BIN_2,OUTPUT);
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin,OUTPUT);
      // swtich for preset
  // LOW = preset #1 | HIGH = preset #2
  pinMode(presetSwtich, INPUT);

      // IR remote contoller inputs and initialization
  IrReceiver.begin(irReceivePin);


  attachInterrupt(limit_switch, switchTriggered, FALLING);
    // initialization of Servo
  joint1.attach(JOINT_1, 500, 2400);
  joint2.attach(JOINT_2, 500 ,2400);
  joint1.write(90);
  ledcSetup(ledChannel_1, freq, resolution);
  ledcSetup(ledChannel_2, freq, resolution);
  ledcAttachPin(BIN_1, ledChannel_1);
  ledcAttachPin(BIN_2, ledChannel_2);

  ESP32Encoder::useInternalWeakPullResistors = UP;
  encoder.attachHalfQuad(26, 25);
  encoder.setCount(0);
```

```
  autoTimer = timerBegin(2, 80, true);
  timerAttachInterrupt(autoTimer, &autoTime, true);
  timerAlarmWrite(autoTimer, 1000000, true);

  debounceTimer = timerBegin(3, 80, true);
  timerAttachInterrupt(debounceTimer, &debounceSwitch, true);
  timerAlarmWrite(debounceTimer, 250000, true);

  checkTimer = timerBegin(0,80,true);
  timerAttachInterrupt(checkTimer, &checkFlag, true);
  timerAlarmWrite(checkTimer, 4000000, true);

    // Enables timer
  //timerAlarmEnable(timer);
  timerAlarmEnable(autoTimer);
  timerAlarmEnable(debounceTimer);
  timerAlarmEnable(checkTimer);
}

/****************************** Loop ***********************
void loop()
{
  checkLoadCell();
  if(dispense){
    dispenseFood();
    scale.tare();
  }

  switch (modeSwitchRead()) {
    /* ---------- MAIN STATE 1: idle ---------- */
    case 1: // idle
    stopMotor();
    //stopElbow();
    toggle = false;
    Serial.println(" Idling ");
    break;

    /* ---------- MAIN STATE 2: manual ---------- */
    case 2:
    // event checker for the arm and wheels
    // changes two states: motorState, wavingArmState

            // this event checker function
            // returns signal from IR remote controller
            Serial.println(" Manual Mode ");
    switch (remoteControllerChecker()) {
      /* ----- subsystem for motor ----- */
      case 64: // idle, '▶||' button
      stopMotor();
      break;

      case 70: // go forward, 'VOL+' button
      motorForward();
      break;

      case 21: // go backward, 'VOL-' button
      motorBackward();
      break;

      /* ----- subsystem for robotic arms ----- */
      case 67: // start waving arm, '▶▶|' button
      toggle = true;
      break;

      case 68:
      toggle = false;
      break;
    }
    break; // break of main switch case 2

    /* ---------- MAIN STATE 3: auto ---------- */
    case 3:

      /* ---------- subsystem for robotic arms ---------
    switch(presetSwtichRead()) {
      case 1:
      Serial.println(" Auto Mode ");
      Serial.println(" Preset 1 ");
      stopMotor();
      toggle = true;
      break;

      case 2:
      Serial.println(" Auto Mode ");
      Serial.println(" Preset 2");
      if(autoForward) {
        motorForward();
      }
      else {
        motorBackward();
      }
      toggle = false;
      break;
    } // end of preset subsystem
    break;
  } // end of main state system
if(toggle){
  waveShoulder();
}
else{
  stopShoulder();
}
}

/************************ Event Checker Functions ********
byte modeSwitchRead() {
  // LOW & LOW = idle mode
  // HIGH & LOW = manual mode
  // LOW & HIGH = preset mode

  if (digitalRead(modeSwitchA) == LOW) {
    return 2;
  }
  else if (digitalRead(modeSwitchB) == LOW) {
    if(!autoFlag){
      autoFlag = true;
    }
    return 3;
  }
  else {
    return 1;
  }
}

byte remoteControllerChecker() {
  // recieve code from IR remote controller
  if (IrReceiver.decode()) {
    byte irData = (byte)(IrReceiver.decodedIRData.command);
    IrReceiver.resume();
    return irData;
  }
  return 0;
```

```
byte presetSwtichRead() {
  if (digitalRead(presetSwtich) == HIGH) {
    return 1;
  }
  else {
    return 2;
  }
}

/*************************** Service Functions *****************
// ====== Motor functions
void stopMotor()
{
  digitalWrite(BIN_1, LOW);
  digitalWrite(BIN_2,LOW);
  Serial.println("Stop Motor");
}

void PID(){
  omegaSpeed = encoder.getCount();
  encoder.clearCount();
  i_integral = (omegaDes - omegaSpeed);
  if (i_integral > IMax) i_integral = IMax;
      else if (i_integral < 0) i_integral = 0;

      D = Kp * (omegaDes - omegaSpeed) + (i_integral) * (Ki / 100);

      if (D > MAX_PWM_VOLTAGE) {
          D = MAX_PWM_VOLTAGE;
      }
      else if (D < -MAX_PWM_VOLTAGE) {
          D = -MAX_PWM_VOLTAGE;
      }
}
void motorForward()
{
  PID();
  if(modeSwitchRead() == 3) {
    digitalWrite(BIN_1, D);
    digitalWrite(BIN_2,LOW);
  }
  else {
    digitalWrite(BIN_1, D);
    digitalWrite(BIN_2,LOW);
  }
  Serial.println("Motor Forward");
}
void motorBackward()
{
  PID();
  if(modeSwitchRead() == 3) {
    digitalWrite(BIN_1, LOW);
    digitalWrite(BIN_2,D);
  }
  else {
    digitalWrite(BIN_1,LOW);
    digitalWrite(BIN_2,D);
  }
  Serial.println("Motor Backward");
}
      // ====== Servo functions
void waveShoulder() {
  if(!toggle) return;

  if (interruptCounter) {
    Serial.println("shoulder angle: ");
    if (dirShoulder) {
      shoulder_pos += 5;
      joint1.write(shoulder_pos);
      Serial.println(shoulder_pos);
      if (shoulder_pos >= 180)
      dirShoulder = false;
    }
    else {
      shoulder_pos -= 5;
      joint1.write(shoulder_pos);
      Serial.println(shoulder_pos);
      if (shoulder_pos <= 0)
      dirShoulder = true;
    }
  }

}

void stopShoulder() {
  joint1.write(shoulder_pos);
  //Serial.println(" Shoulder Stopped" );
}

void dispenseFood(){
  Serial.println("Dispensing ");
 digitalWrite(dirPin, HIGH);
  // Spin the stepper motor 1 revolution slowly:
  stopMotor();
  while(dispense){
    // These four lines result in 1 step:
    //Serial.println(" Dispensing ");
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2500);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2500);
  }
}

void checkLoadCell() {
  Serial.print(check);
  Serial.println(scale.get_units());

  if(check){
    if(scale.get_units() > pthresh) {
          check = false;
          dispense = true;
          debounce = true;
          timerStart(debounceTimer);
          timerStart(checkTimer);
          started = true;
    }
    else if(scale.get_units() < nthresh){
      check = false;
      dispense = true;
      debounce = true;
      timerStart(debounceTimer);
      timerStart(checkTimer);
    }
  }
 }
void moveRandom(){
  if(check){
    joint1.write((int)random(180));
  }
}
```

```
400       // ====== Timer functions
401   //void IRAM_ATTR onTime() {
402     //portENTER_CRITICAL_ISR(&timerMux0);
403     //interruptCounter = true; // the function to be called when timer interrupt is triggered
404     //portEXIT_CRITICAL_ISR(&timerMux0);
405   //}
406
407   void IRAM_ATTR autoTime() {
408     portENTER_CRITICAL_ISR(&timerMux1);
409     autoFlag = false;
410     if(autoForward) {
411       autoForward = false;
412     }
413     else {
414       autoForward = true;
415     }
416     portEXIT_CRITICAL_ISR(&timerMux1);
417   }
418
419   void IRAM_ATTR switchTriggered() {
420     if(!debounce){
421       dispense = false;
422       timerRestart(debounceTimer);
423       timerStop(debounceTimer);
424     }
425   }
426
427   void IRAM_ATTR debounceSwitch(){
428     debounce = false;
429   }
430
431   void IRAM_ATTR checkFlag(){
432     check = true;
433     timerRestart(checkTimer);
434     timerStop(checkTimer);
435
436   }
```