FINAL PROJECT REPORT
ME 102B Group 4: DrinkBot
Members: Dixun Cui, Andrew Chen, Mayank Gupta, Ocean Zhou

Opportunity this device represents

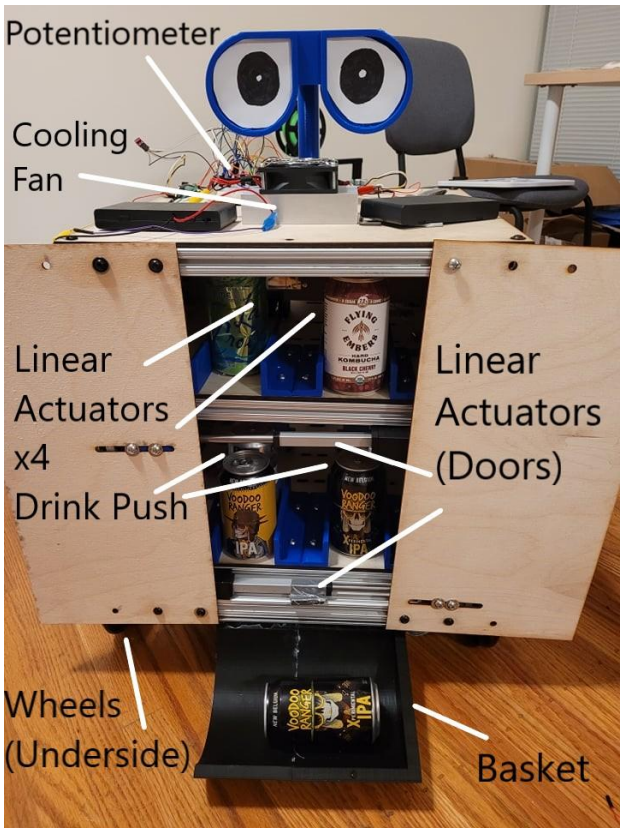
As hard working students of Mechanical Engineering at UC Berkeley, we often find ourselves tired and in dire need of a beverage or three. Unfortunately, current options require us to walk all the way to the fridge to acquire our desired beverage, and market research shows this is simply too much walking for tired and thirsty students. We realized there was a clear opportunity to make beverage accruement simpler and less strenuous.

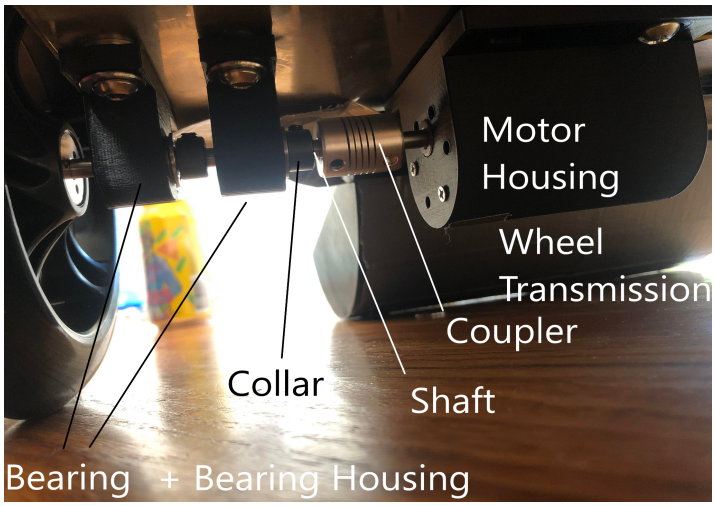
Device's High Level Strategy

The *initial high level strategy* was too maximalist and time consuming for the time limit of one semester. In our original strategy, after the robot's single door opened, a motor-controlled gripper arm would grab the drink and hand it to the client. A wireless remote would control the robot's driving. Furthermore, the original design had a collision sensor in front which would warn if the robot was too close to an obstacle. We had also considered making the robot be able to take pre-programmed routes to destinations.

Our *updated high level strategy* instead uses 2 doors that are pushed open by linear actuators. Linear actuators sitting inside the robot will push drinks out one at a time which a bucket will catch the drink as it falls out from the robot. The robot is driven by a wireless infrared (IR) remote which communicates with a corresponding sensor. The remote is also used to command a drink to be pushed out. The speed was not a concern as we were focused on getting it to drive. Our resultant speed was close to 0.5 m/s which satisfied our initial requirements. Furthermore, the drivetrain system was able to support the load of the frame.

Labeled Photo of Final Device

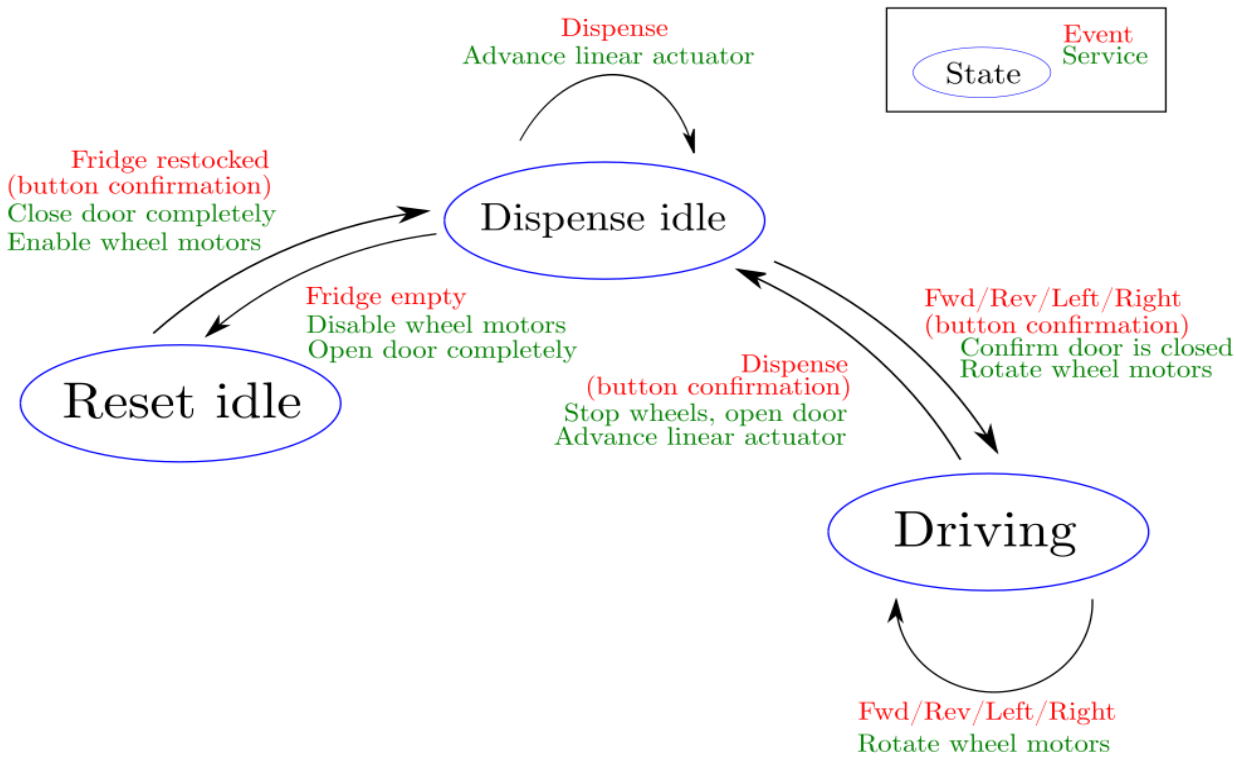




Function Critical Decisions you made

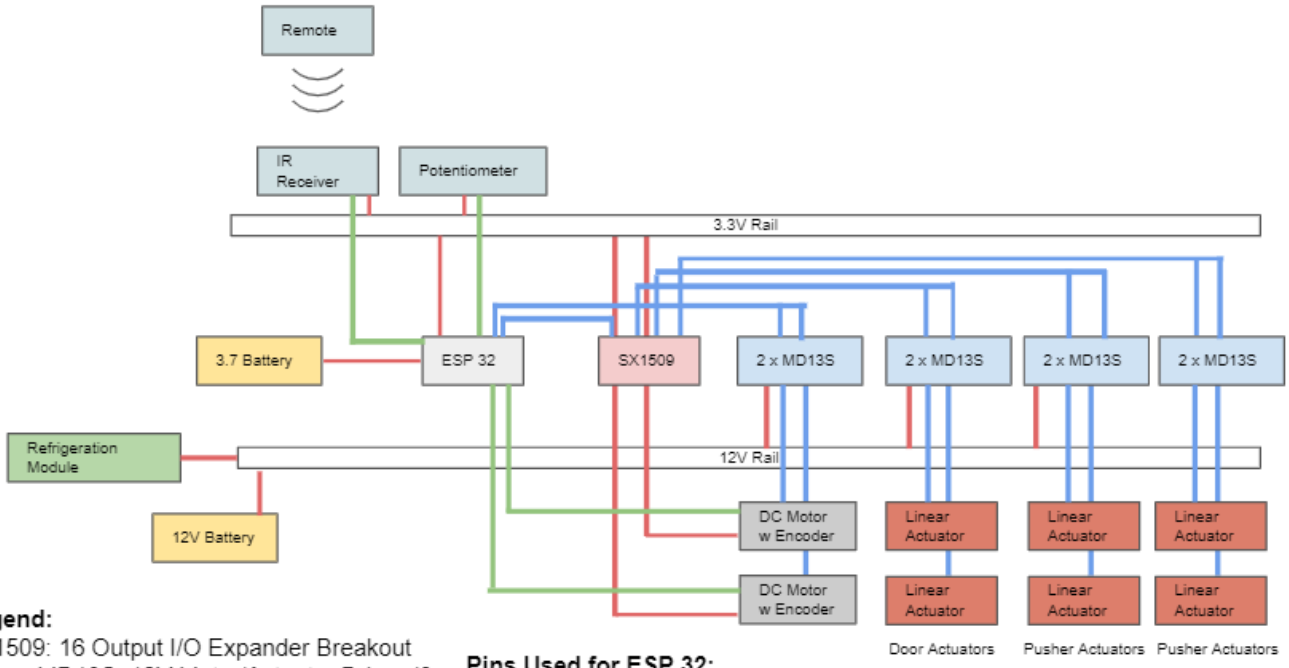
1. We first calculated the estimated weight of the device. The density of wood, PLA (for 3D) and aluminum are ρ_w = 1500kg/m^3, ρ_{PLA} = 1250kg/m^3 and ρ_{al} = 2710kg/m^3. The wood plates have weight: $\rho_w(17\times17\times0.25in^3)(1.64\times10^{-5}m^3/in^3)(9.8m/s^2)$ + $3\rho_w(17\times17\times0.125in^3)(1.64\times10^{-5}m^3/in^3)(9.8m/s^2)$ = 43.3N. The aluminum base plate has weight: $\rho_{al}(17\times17\times0.25in^3)(1.64\times10^{-5}m^3/in^3)(9.8m/s^2)$ = 31.44N. The 2 drink racks plus the sodas have weight: $2\rho_{wd}(15\times15\times0.25in^3)(9.8m/s^2)$ + 8*3.35N = 48.3N. The frame weighs 8(17in)(0.0424lb/in) + 8(15in)(0.0424lb/in) = 10.8544lb. The 6 linear actuators weigh 3 lbs in total. The electronics attached will weigh at most 4 lbs. Converting to pounds, in total this means the robot will weigh (not including the motors and drivetrain system), Mg = 36.56 lbs. Upon testing, we found that the drivetrain and baseplate were able to handle this weight sufficiently.
2. We also calculated the minimum force needed from a linear actuator to push 2 drinks. The average weight of a can of soda is 0.75 lb. The weight of 2 soda cans is then 1.5 lb which equals 6.7 N, meaning $F_g = F_N$ = 6.7N. Online sources say the coefficient of static and kinetic friction of aluminum on wood is μ_s = 0.20,μ_k = 0.15. Thus, the max static friction is $F_{s,max} = \mu_s F_N$ = 1.34N = 0.3lbs. The kinetic friction is $F_{k,max} = \mu_k F_N$ = 1.01N = 0.23lbs. The linear actuator in our BOM can provide forces up to 4.5 lbs so it is able to push two sodas easily.
3. We also calculated the necessary torque that needed to be delivered so the robot could drive. For a wheel to be able to drive and not slip, Torque $\tau = \mu_s$Mgr, M is the robot mass, and r is the wheel radius. The robot has a mass of 16.7 kg. The wheels have radius r = 72mm = 7.2cm. Thus in units of kg-cm, each motor much exert a torque of τ= 0.2(16.7 kg)(7.2cm) = 24.048 kg-cm. The motors we bought had a stall torque of 49 kg cm and were thereby able to drive the robot easily.

State Transition Diagram (same as before)



Updated Circuit Diagram

The updated circuit diagram is shown on the next page. The key difference is: The drivers for the linear actuators and motors are no longer the TB6612 dual motor drivers but instead are the Cytron MD13S motor drivers which each individually drive one motor/actuator. Other than this all else is the same.



Legend:
SX1509: 16 Output I/O Expander Breakout
Cytron MD13S: 12V Motor/Actuator Driver (8 in total, one for each motor/actuator)
Red Wires: Power
Green Wires: Input Signal
Blue Wires: Output Signal
Everything is grounded, but wires are not shown to reduce clutter

Pins Used for ESP 32:
GND: will be connected to ground rail
BAT: will be connected to 3.7V Battery
3.3V: will supply 3.3V rail
I2C: connected to breakout board
A2,A3 used for receiver, potentiometer
27 ,33, 15, 32 used for motor encoders
12, 14, A0, A1 connected to the driver for motor outputs

ESP 32 Protection:
The higher 12V power rail will not be connected to the ESP. For motors, the ESP and also the expander board will only be connected to the drivers, which will supply the 12V power.

Short Reflection

Working as roommates made the entire project logistically much less challenging, as we had a single place to store parts, could easily integrate multiple systems as needed and bounce ideas off each other easily, and we would highly recommend this where feasible. We also ensured our project's complexity stemmed from assessable requirements, rather than including ideas and mechanisms beyond the scope of class learnings. In saying that, what we wish we had done differently is descoping the project earlier and adding complexity after proving initial capabilities, as spent time on earlier project requirements devising a project that was unfeasible with the time we had, and found ourselves being more successful, making the system more complex after iterating through an original, base, design.

Appendix

The BOM will be shown on the next page.

Images of CAD

The CAD remains the same as the previous assignment P3. To show the inside, the doors have been hidden.

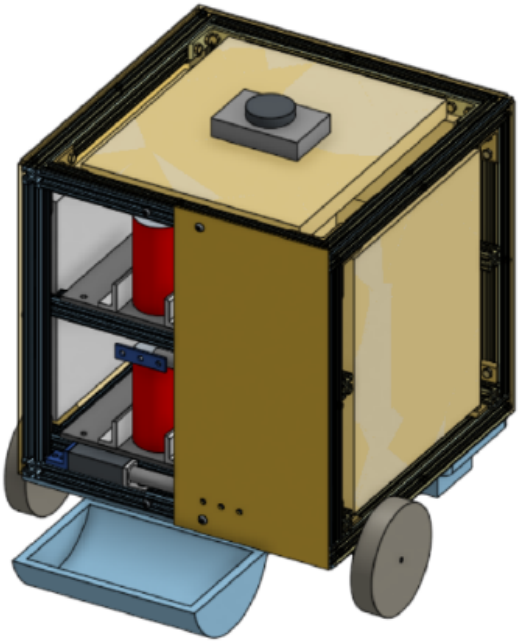


Fig. 1: Isometric View

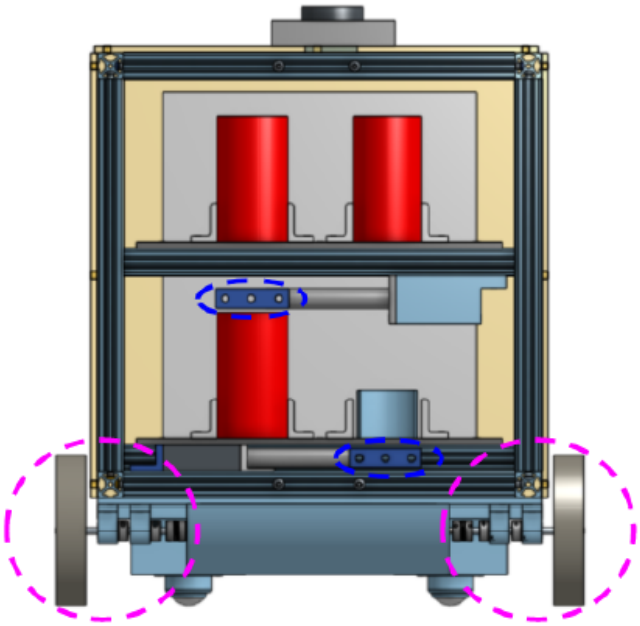
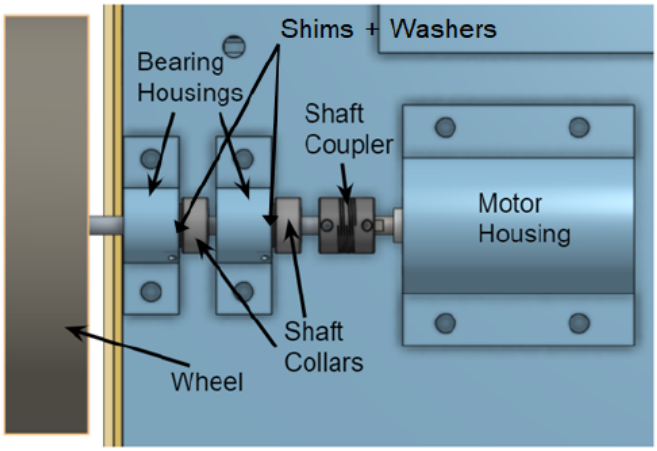
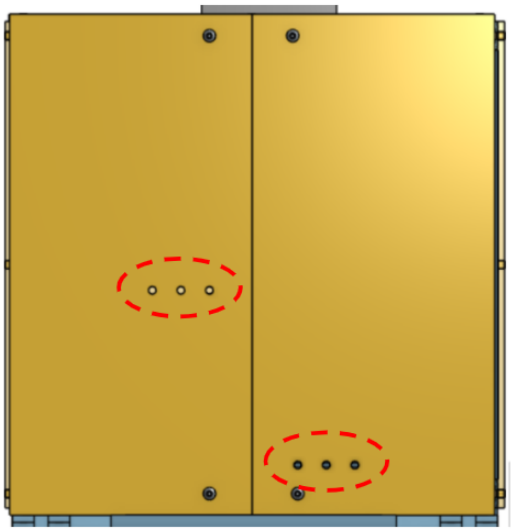


Fig. 2: Front View

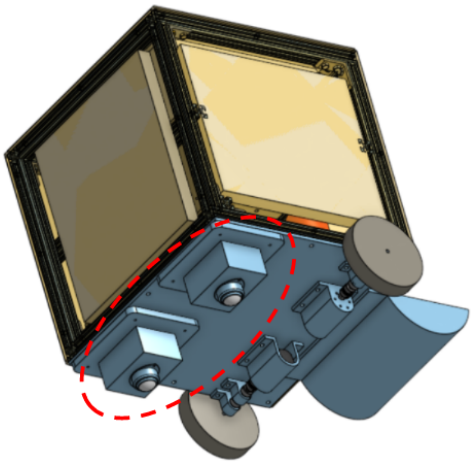
Zooming in on the magenta circled regions, the transmission system is below:



The front view with the doors not hidden looks as so. The doors are opened by the blue couplers attached to linear actuators.



The bottom view, with the ball casters circled



Project Name's Purchase Portfolio								Total (Projected):	$ 596.88
Item Name	**Description**	**Purchase Justification**	**Serial Number / SKU**	**Price (ea.)**	**Quantity**	**Vendor**	**Link to Item**	**Notes**	**Subtotal**
Project Name's Purchase Portfolio								**Total (Projected):**	**$ 596.88**
Item Name	**Description**	**Purchase Justification**	**Serial Number / SKU**	**Price (ea.)**	**Quantity**	**Vendor**	**Link to Item**	**Notes**	**Subtotal**
[Example] 1/2" Ball Bearings	1/2" ID, .75" OD unshielded ball bearings	Used to constrain shaft rotation for main driveshaft	60355K505	$ 6.27	1	McMaster Carr	https://www.mc	May need to replace with shielded bearings down the line if debris accumulates...	$ 6.27
Linear Actuator	Mini Electric Linear Actuator Stroke 4"–Force 4.5 lbs–12V | High-Speed 1.97"/sec–Weight 0.25KG Ideal for Intelligent Range Hood, Fan Blades, Cabinets, Window Opener, Robotics, Home Automation	4 Used to push drinks from along drink rail and into the drink bucket so user can receive drinks. Other 2 used to actuate opening and closing of door	L11010101010-1	$ 29.99	1	Amazon	https://www.an		$ 29.99
Styrofoam Boards	1 Inch Thick Foam Board Sheets, 17x11 Polystyrene Rectangles for DIY Crafts, Art Supplies, Sculpture (6 Pack)	Needed for insulating the fridge as it is being cooled	B07C969WR8	$ 17.99	1	Amazon	https://www.an		$ 17.99
Flanged Bearing	uxcell F696ZZ Flange Ball Bearing 6x15x5mm Shielded Chrome Bearings 5pcs	Needed to support motor shaft when driving	a19040300ux0199	$ 7.49	1	Amazon	https://www.an		$ 7.49
M3/M4 Bolts	180pcs Screws Assortment Sets, M3, M4 Screw & Nut, Gasket Copper Pillar, DIY Electronics Mechanical Assemble Repair Hardware Fasterner Kit, Flat Hex Washer Head Cap Socket Bolt	Used for Smaller Assemblies, such as the motor housing	31161500	$ 4.99	1	Amazon	https://www.an		
Ball Casters	4-Pack Antrader Two-Hole Flange Mounted 1-inch Nylon Ball Transfer Unit Bearing Conveyor Roller Casters, 66lbs Load Capacity	Provide passive wheel stability to system (attached to the base plate)	AZ18110704	$ 13.99	1	Amazon	https://www.an		$ 13.99
12V Power Supply	CO-RODE 8 x AA 12V Battery Holder Case Box Wired ON/Off Switch w Cover Pack of 2	Needed to power our entire system	B00VE7HBMS	$ 8.00	1	Amazon	https://www.an		$ 8.00
Refrigeration Module	DIY Refrigeration Semiconductor, DC12V Thermoelectric Cooling System, Conduction Module, Radiator, Fan, XH-X200	Need to cool drinks within the housing, to ensure fresh for consumption	XH-X200	$ 36.79	1	Amazon	https://www.an		$ 36.79
Flexible Shaft Coupler	Sydien 2pcs 6 to 6mm CNC Shaft Coupling Flexible Coupler Motor Connector,D19xL25mm	Needed to convert motor torque to drive wheel torque	B083LFR5N6	$ 11.19	1	Amazon	https://www.an		$ 11.19
SparkFun 16 Output I/O Expander Breakout - SX1509	Expansion Board for ESP 32, for More Pins	Give Enough Pins for All Sensors + Actuators	BOB-13601	$ 6.50	2	Sparkfun	https://www.sp		$ 13.00
IR (Infrared) Receiver Sensor	IR Signal Receiver	Receive Signals to Remote Control Robot	PID: 157	$ 1.95	3	Adafruit	https://www.ad		$ 5.85
Mini Remote Control	IR Remote	Send Signals to Control Robot	PID: 389	$ 4.95	1	Adafruit	https://www.ad		$ 4.95
TB6612 Motor Driver	12V Motor Driver, Dual Channel	Used to Drive/Power Motors and Actuators	713	$ 3.33	3	Pololu	https://www.po		$ 9.99

Project Name's Purchase Portfolio								Total (Projected):	$ 596.88
Item Name	**Description**	**Purchase Justification**	**Serial Number / SKU**	**Price (ea.)**	**Quantity**	**Vendor**	**Link to Item**	**Notes**	**Subtotal**
Scooter/Skate Wheel 144 mm	Rubber Scooter Wheel	Wheels for Bot	3281	$ 7.95	2	Pololu	https://www.po		$ 15.90
Aluminum Scooter V	Adapter from Wheel to Shaft	Needed to Connect from 6mm shaf	2674	$ 4.95	2	Pololu	https://www.po		$ 9.90
30:1 Metal Gearmot	DC Motor with 6mm shaft	Used to Drive Robot	4752	$ 39.95	2	Pololu	https://www.po		$ 79.90
150:1 Metal Gearm	DC Motor with 6 mm shaft	Driving Full Weight Robot	2828	$ 39.95	2	Pololu	https://www.po		$ 79.90
M3x 25 mm Bolt	M3x 25 mm Socket Head Bolts	For Wheel Adatper	9180490	$ 5.19	1	Amazon	https://www.an		$ 5.19
Linear Actuator	Mini Electric Linear Actuator Stroke 4"–Force 4.5 lbs–12V | High-Speed 1.97"/sec–Weight 0.25KG Ideal for Intelligent Range Hood, Fan Blades, Cabinets, Window Opener, Robotics, Home Automation	4 Used to push drinks from along drink rail and into the drink bucket so user can receive drinks. Other 2 used to actuate opening and closing of door	L11010101010-1	$ 29.99	5	Amazon	https://www.an		$ 149.95
6 mm x 200 mm Shaft	Eowpower 4Pcs Stainless Steel 6mm x 200mm Round Rod Turning Lathe Bars Tool	Shaft for Drive System	30102405	$ 9.69	1	Amazon	https://www.an		$ 9.69
1/4-20 Kit	VIGRUE 300pcs 304 Stainless Steel 1/4-20 UNC Hex Button Head Cap Screw Bolts Flat Washers Nuts Assortment Kit Machine Screws Set, 8 Sizes (Length from 3/8" to 2"), Upgraded	For Assembling Frame	31160000	$ 24.99	1	Amazon	https://www.an		$ 24.99
Alligator Clip Jumpters	UCTRONICS Breadboard Alligator Clip Jumpers - Gator to Male and Female Jumper Wires Test Lead 2x10pcs 8 inch for Power Supply, LED Strips, Multimeters, Arduino Lilypad and Raspberry Pi	For Connecting to Frayed Ends of Actuators	41110000	$ 7.99	1	Amazon	https://www.an		$ 7.99
1/4-20 Rivet Nuts	Rivet Nut, LOKMAN 100 Pieces 1/4-20UNC Carbon Steel Flat Head Rivnut Threaded Insert Nut,Knurled Body (1/4"-20)	For Attaching Drive System to Frame	39121400	$ 10.99	1	Amazon	https://www.an		$ 10.99
6mm Bore Drill Stop	uxcell 6mm Bore Drill Stop, Drill Bit Depth Holder, Set Screw Style, Black Oxide Carbon Steel, Pack of 2	For use as shaft collars for bearings	a18103100ux0510	$ 5.99	2	Amazon	https://www.an		$ 11.98
1/8" Aluminum Plate	.125 1/8" Mill Finish Aluminum Sheet Plate	Mounting plate for drive system	303279339244	$ 25.00	1	Ebay	.125 1/8" Mill Finish Aluminum Sheet Plate 5052 18" x 18"		$ 25.00

Screenshots of Entire Code Showing Event Driven Programming

The code is shown below in sequential order of left to right and then top to bottom.

DrinkBot §

```
#include <IRremote.h>
#include <ESP32Encoder.h>
#define outputLED 33
#define outputLED2 15
#define POT 25

#include <Wire.h> // Include the I2C library (required)
#include <SparkFunSX1509.h> // Include SX1509 library

const byte SX1509_ADDRESS = 0x3E;
SX1509 io; // Create an SX1509 object
// Setup variables -----------------------------------------

//Top row of drinks
const int DIR1 = 0;
const int PWM1 = 1;
const int DIR2 = 2;
const int PWM2 = 3;

//Bottom row of drinks
const int DIR3 = 13;
const int PWM3 = 12;
const int DIR4 = 15;
const int PWM4 = 14;

//Doors
const int DIRL = 6;
const int PWML = 7;
const int DIRR = 9;
const int PWMR = 8;

unsigned long time1;
boolean newData = false;
int routine = 0;
```

```
//Drinks Dispensed Tracking
int dispensed1 = 0;
int dispensed2 = 0;
int dispensed3 = 0;
int dispensed4 = 0;

//For debugging
int pwm_val, pwm_2, pwm_3, pwm_4 = 0;
int dir_val, dir_2, dir_3, dir_4 = 0;


int disp_length = 4000; //ms
int full_length = 9000; //ms

//motor 1 pins
#define BIN_1 34
#define BIN_2 39
#define DIR_PIN 32
#define PWM_PIN 14

//motor 2 pins
#define BIN2_1 15
#define BIN2_2 33
#define DIR2_PIN 12
#define PWM2_PIN 27

ESP32Encoder encoder;
ESP32Encoder encoder2;
```

```
//Set Closed-loop Control Parameters and Other Settings
double Kp = 20;
double Ki = 1;
int D = 0;
unsigned long currentTime, previousTime;
double elapsedTime;
double error;
double lastError;
double input, input2, omegaDes, D1, D2;
double cumError;
const int MAX_PWM_VOLTAGE = 255;
const int NOM_PWM_VOLTAGE = 225;
int RECV_PIN = 21;
volatile int VALUE = LOW;
int command = 0;
int startMillis;
int currentMillis;
int case = 1;

IRrecv irrecv(RECV_PIN);


decode_results results;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.println("Enabling IRin");
  irrecv.enableIRIn();
  Serial.println("Enabled IRin");
  pinMode(outputLED, OUTPUT);
  pinMode(outputLED2, OUTPUT);
  startMillis = millis();
  omegaDes = 0;
  pinMode(POT, INPUT);
```

```
pinMode(POT, INPUT);
pinMode(DIR_PIN, OUTPUT);
pinMode(PWM_PIN, OUTPUT);
pinMode(DIR2_PIN, OUTPUT);
pinMode(PWM2_PIN, OUTPUT);
ESP32Encoder::useInternalWeakPullResistors = UP; // Enable the weak pull up resistors
encoder.attachHalfQuad(BIN_1, BIN_2); // Attache pins for use as encoder pins
encoder.setCount(0);  // set starting count value after attaching
encoder2.attachHalfQuad(BIN2_1, BIN2_2); // Attache pins for use as encoder pins
encoder2.setCount(0);  // set starting count value after attaching

ledcSetup(0, 5000, 8);
ledcAttachPin(PWM_PIN, 0);

//channel 1 is for motor 2
ledcSetup(1, 5000, 8);
ledcAttachPin(PWM2_PIN, 1);

Wire.begin();
  if (io.begin(SX1509_ADDRESS) == false){
    Serial.println("Failed to communicate. Check wiring and address of SX1509.");
    while (1)
      ; // If we fail to communicate, loop forever.
      // Troubleshooting: try SCL on GPIO22 and SDA on GPIO 21
  }
io.pinMode(DIR1, OUTPUT);
io.pinMode(PWM1, OUTPUT);
io.pinMode(DIR2, OUTPUT);
io.pinMode(PWM2, OUTPUT);
io.pinMode(DIR3, OUTPUT);
io.pinMode(PWM3, OUTPUT);
io.pinMode(DIR4, OUTPUT);
io.pinMode(PWM4, OUTPUT);
```

```
oid loop() {
 //Pull PWM (arming pins) low
 io.digitalWrite(PWM1, LOW);
 io.digitalWrite(PWM2, LOW);
 io.digitalWrite(PWM3, LOW);
 io.digitalWrite(PWM4, LOW);
 io.digitalWrite(PWMR, LOW);
 io.digitalWrite(PWML, LOW);
 // Probe for Receiver
 if (irrecv.decode(&results)) {
   if (results.value == 16621663 || results.value == 16584943 || results.value == 16601263 || results.value == 16625743)
     command = results.value;
     state = 1;
   } else if (results.value == 16582903 || results.value == 16615543 || results.value == 16599223 || results.value ==
   16591063 || results.value == 16589023 || results.value ==16617583) {
      command = results.value;
      state = 1;
   } else if (results.value == 4294967295) {
     if (command == 16621663 || command == 16584943 || command == 16601263 || command == 16625743) {

     } else {
       command = 0;
       state = 2;
     }
   } else {
     command = 0;
     stopmotor();
     state = 3;
   }
   irrecv.resume();
   startMillis = millis();
 } else {
   currentMillis = millis();
   if ((currentMillis - startMillis)>250) {
```

```
// State Machine Implementation
switch (state) {
  case 1: // driving
  if (command == 16621663) {
  Serial.println("Go forward");
  forward();
} else if (command == 16584943) {
  Serial.println("Go left");
  left();
} else if (command == 16601263) {
  Serial.println("Go right");
  right();
} else if (command == 16625743) {
  Serial.println("Go back");
  back();
} else if (command == 16582903) {
  if (dispensed1<2) {
    Serial.println("Dispense1");
    dispense1();
    dispensed1+=1;
    }
  }
 break;
 case 2: //dispense idle
 if (command == 16582903) {
  if (dispensed1<2) {
    Serial.println("Dispense1");
    dispense1();
    dispensed1+=1;
  }
} else if (command == 16615543) {
  if (dispensed2<2) {
    Serial.println("Dispense2");
    dispense2();
    dispensed2+=1;
  }
} else if (command == 16599223) {
  if (dispensed3<2) {
    Serial.println("Dispense3");
    dispense3();
    dispensed3+=1;
  }

} else if (command == 16591063) {
  if (dispensed4<2) {
    Serial.println("Dispense4");
    dispense4();
    dispensed4+=1;
  }
 state = 1;
 break;
 case 3: //Reset Idle
if (command == 16589023) {
  Serial.println("Reload");
  reload();
} else if (command == 16617583) {
  Serial.println("Reload close");
  reloadclose();
}
state = 2;
break;
}
```

```
// Helper Functions for Operations
void forward(){
  D = map(analogRead(POT), 0, 4095, 0, NOM_PWM_VOLTAGE);
  currentTime = millis();
  elapsedTime = currentTime - previousTime;
  omegaDes = 0.27*D;
  if (elapsedTime > 10) {
    input = encoder.getCount( );
    input2 = encoder2.getCount( );
    encoder.clearCount( );
    encoder2.clearCount( );
    previousTime = currentTime;
  }
  D1 = D + Kp*(omegaDes - input);
  if (D1 > MAX_PWM_VOLTAGE) {
    D1 = MAX_PWM_VOLTAGE;
  }
  D2 = D + Kp*(omegaDes - input);
  if (D2 > MAX_PWM_VOLTAGE) {
    D2 = MAX_PWM_VOLTAGE*0.95;
  }
  ledcWrite(0, D1);
  ledcWrite(1, D2);
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR2_PIN, HIGH);
}

void left(){
  D1 = 150;
  D2 = 150;
  ledcWrite(0, D1);
  ledcWrite(1, D2);
  digitalWrite(DIR_PIN, LOW);
  digitalWrite(DIR2_PIN, LOW);
```

```
void right(){
  D1 =150;
  D2 = 150;
  ledcWrite(0, D1);
  ledcWrite(1, D2);
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR2_PIN, HIGH);
}

void back(){
  D = map(analogRead(POT), 0, 4095, 0, NOM_PWM_VOLTAGE);
  currentTime = millis();
  elapsedTime = currentTime - previousTime;
  omegaDes = 0.27*D;
  if (elapsedTime > 10) {
    input = encoder.getCount( );
    input2 = encoder2.getCount( );
    encoder.clearCount( );
    encoder2.clearCount( );
    previousTime = currentTime;
  }
  D1 = D + Kp*(omegaDes - input);
  if (D1 > MAX_PWM_VOLTAGE) {
    D1 = MAX_PWM_VOLTAGE;
  }
  D2 = D + Kp*(omegaDes - input);
  if (D2 > MAX_PWM_VOLTAGE) {
    D2 = MAX_PWM_VOLTAGE;
  }
  ledcWrite(0, D1);
  ledcWrite(1, D2);
  digitalWrite(DIR_PIN, HIGH);
  digitalWrite(DIR2_PIN, LOW);
}
```

```
  void stopmotor(){
    ledcWrite(0, 0);
    ledcWrite(1, 0);
  }

  void open_both(){
    time1 = millis();
    io.digitalWrite(PWMR, HIGH);
    io.digitalWrite(DIRR, LOW);
    io.digitalWrite(PWML, HIGH);
    io.digitalWrite(DIRL, LOW);
    while(millis() < time1 + full_length) {
    Serial.println("Opening both doors");
    }
    io.digitalWrite(PWMR, LOW);
    io.digitalWrite(PWML, HIGH);
  }
  void close_both(){
    time1 = millis();
    io.digitalWrite(PWMR, HIGH);
    io.digitalWrite(DIRR, HIGH);
    io.digitalWrite(PWML, HIGH);
    io.digitalWrite(DIRL, HIGH);
    while(millis() < time1 + full_length) {
    Serial.println("Closing both doors");
    }
    io.digitalWrite(PWMR, LOW);
    io.digitalWrite(PWML, HIGH);
  }
  void retract_all(){
    time1 = millis();
    io.digitalWrite(PWM1, HIGH);
    io.digitalWrite(PWM2, HIGH);
    io.digitalWrite(DIR1, HIGH);
```

```
void dispense4() {
  open_both();
  time1 = millis();
  io.digitalWrite(PWM4, HIGH);
  io.digitalWrite(DIR4, LOW);
  while (millis() < time1 + disp_length) {
    Serial.println("Extending 4 for 4 sec");
  }
  io.digitalWrite(PWM4, LOW);
  close_both();
}

void reload() {
  open_both();
  retract_all();
  dispensed1 = 0;
  dispensed2 = 0;
  dispensed3 = 0;
  dispensed4 = 0;
}

void reloadclose() {
  close_both();
}
```