**Solar Tracker**

import time  
from machine import Pin,PWM  
 #Looking at the panel with sensors facing you  
TL = Pin(3, Pin.IN) #LDR top left  
TR = Pin(4, Pin.IN) #LDR top right  
BL = Pin(5, Pin.IN) #LDR Bottom left  
BR = Pin(6, Pin.IN) #ldr Bottom right  
  
Xaxis = PWM(Pin(2)) # sets x and y axis servo pins as pwm signal  
Yaxis = PWM(Pin(17))  
  
Xaxis.freq(50) # sets x and y axis pwm output frequency  
Yaxis.freq(50)  
  
positionx = 1500000 # start each servo to their midpoint position in nanseconds as duty cycle  
positiony = 1500000  
  
while True: # loop section  
  
 if (TL.value() == 1 and TR.value() == 1): #if top left and top right LDR sensors reach their preset values  
 Yaxis.duty\_ns(positiony) # move y axis servo to saved variable  
 positiony = positiony + 5000 # add 5000 nanoseconds to duty cycle and change the variable value  
 time.sleep(0.01) # wait 10 milliseconds between each step  
  
 if (TL.value() == 1 and BL.value() == 1): # same sequences as before with different sensors sets  
 Xaxis.duty\_ns(positionx)  
 positionx = positionx - 5000 # negative indicates reverse direction  
 time.sleep(0.01)  
  
 if (BL.value() == 1 and BR.value() == 1):  
 Yaxis.duty\_ns(positiony)  
 positiony = positiony - 5000  
 time.sleep(0.01)  
  
 if (TR.value() == 1 and BR.value() == 1):  
 Xaxis.duty\_ns(positionx)  
 positionx = positionx + 5000  
 time.sleep(0.01)

**Main Components**

#from hx711\_gpio1 import HX711  
import time  
from machine import Pin,PWM  
  
# Constants for the stepper motor pins  
IN1 = Pin(18, Pin.OUT)  
IN2 = Pin(19, Pin.OUT)  
IN3 = Pin(20, Pin.OUT)  
IN4 = Pin(21, Pin.OUT)  
IRSensor = Pin(16, Pin.IN, Pin.PULL\_UP)  
Photosensor = machine.ADC(28)  
#data\_pin = Pin(14, Pin.IN, Pin.PULL\_DOWN)  
#clock\_pin = Pin(15, Pin.OUT)  
#hx711 = HX711(clock\_pin, data\_pin)  
#hx711.tare()  
lid = PWM(Pin(2))  
lid.freq(50)  
  
# Sequence for moving the stepper motor  
SEQUENCE = [[1,0,0,0], [0,1,0,0], [0,0,1,0], [0,0,0,1]]  
  
# Function to move the stepper motor  
def move\_stepper(direction, steps):  
 # Set the input pins  
 pins = [IN1, IN2, IN3, IN4]  
  
 # Set the direction of the sequence  
 if direction == 'forward':  
 sequence = SEQUENCE  
 elif direction == 'backward':  
 sequence = list(reversed(SEQUENCE))  
  
 # Loop through the specified number of steps  
 for i in range(steps):  
  
 # Set the input pins based on the current step  
 for j in range(len(pins)):  
 pins[j].value(sequence[i%4][j])  
  
 # Delay between steps  
 time.sleep(0.002)  
  
# Main loop  
while True:  
 print(IRSensor.value())  
 print("ADC: ", Photosensor.read\_u16())  
 #raw\_wt = hx711.read()  
 #sf = 340/350000  
 #weight = raw\_wt\*sf  
  
 if (IRSensor.value() == 0 and Photosensor.read\_u16() <= 800): # if bowl is empty (IR) and is daytime(photosensor)  
 move\_stepper('forward', 512)  
 time.sleep(2)  
  
 if (weight.value() <= 600 and Photosensor.read\_u16() <= 800): # if weight is in range and is daytime  
 for position in range(500000, 2500000, 5000): # open lid  
 lid.duty\_ns(position)  
 time.sleep(0.01)  
  
 else:  
 for position in range(2500000, 500000, -5000): # close lid  
 lid.duty\_ns(position)  
 time.sleep(0.01)

**Ultrasonic Sensor**

from machine import Pin  
import time  
trigger = Pin(2, Pin.OUT)  
echo = Pin(3, Pin.IN)  
def ultra():  
 trigger.low()  
 time.sleep\_us(2)  
 trigger.high()  
 time.sleep\_us(5)  
 trigger.low()  
 while echo.value() == 0:  
 signaloff = time.ticks\_us()  
 while echo.value() == 1:  
 signalon = time.ticks\_us()  
 timepassed = signalon - signaloff  
 distance = (timepassed \* 0.0343) / 2  
 print("The distance from object is ", distance, "cm")  
 if distance >60:  
 print("Storage Level Low")  
  
while True:  
 ultra()  
 time.sleep(0.1)