```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: AutomaticObjectFinderUsingLeftPoint
%   Source Files: rotateBotLeftPoint, driveDirection, fsrBumpSensor,
%   straightScan, findNewObject
%   Description: This script scans and locates four objects with a robot
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set up boards
a = arduino('COM3','uno'); % Bottom Boaar
b = arduino('COM4','uno'); % Top Boaardcclea

% Set pins
% Board A (bottom)
triggerPin='D5'; % THIS PIN IS USED CURRENTLY
echoPin='D2';
FSRPin1='A0';
FSRPin2='A1';
headLight='D10';
ultrasonicObj=ultrasonic(a,triggerPin,echoPin);

% Board B (top)
LED1='D6';
LED2='D5';
LED3='D4';
LED4='D2';
sPin='D10';
tailLights='D9';

% On Off
On=1;
Off=0;

% Set speeds
halfSpeed=2.65; % speed in volts
backUpTime=.25; % time in seconds

% Number of Objects
numOfObjects=3;

% SCAN ANGLES
scanAngle=0:15:180; % Scan Angles
scanAngleFlip=flip(scanAngle); % Flips scan angle for logical viewing
scanAngleRad=(scanAngle/180)*pi; % Radians for graph
scanAngleRad=flip(scanAngleRad); % Flipped Radians
scanAngVol=scanAngle/180; % Voltages for servo

% Line Up Bot with left most object
```

```
LeftPointDist=rotateBotLeftPoint(a,b,sPin,ultrasonicObj,scanAngVol,scanAngleFlip, ↙
scanAngleRad);

% turn on headlights
writeDigitalPin(a,headLight,On)

% Forward Drive (Slow Mode)
direction='f';
speed=halfSpeed;
driveDirection(a,b,direction,speed)

% Checks for bump
bumpSensor=fsrBumpSensor(a,FSRPin1,FSRPin2);
while bumpSensor==0
    %Read bump sensor
    bumpSensor=fsrBumpSensor(a,FSRPin1,FSRPin2);
end

% Turn off headlights
writeDigitalPin(a,headLight,Off)

% Stop robot movement
direction='s';
speed=Off;
driveDirection(a,b,direction,speed)

% Turn ON A LED
writeDigitalPin(b,LED1,On)

% Reverse robot with tail lights for safety
writeDigitalPin(b,tailLights,On)

% Reverse Drive (Slow Mode)
direction='b';
speed=halfSpeed;
driveDirection(a,b,direction,speed)
pause(backUpTime) % Move back without scanning to elimnate errors

% Scan Object while backing up
Distance=straightScan(b,sPin,ultrasonicObj);

% backsup till desired distances
while Distance<LeftPointDist
    Distance=straightScan(b,sPin,ultrasonicObj);
end

% Turn off tail lights
writeDigitalPin(b,tailLights,Off)
```

```
% Stop robot movement
direction='s';
speed=Off;
driveDirection(a,b,direction,speed)


% Find next three objects
for i=1:numOfObjects % loops 3 times

    % Locate and line up with next object
    nextLeftPointDist=findNewObject(a,b,sPin,ultrasonicObj,scanAngVol,scanAngleRad,
scanAngleFlip);

    % Turn on headlights
    writeDigitalPin(a,headLight,On)

    % Forward Drive (Slow Mode)
    direction='f';
    speed=halfSpeed;
    driveDirection(a,b,direction,speed)

    % Checks for bump
    bumpSensor=fsrBumpSensor(a,FSRPin1,FSRPin2);
    while bumpSensor==0
        %Read bump sensor
        bumpSensor=fsrBumpSensor(a,FSRPin1,FSRPin2);
    end

    % Turn off headlights
    writeDigitalPin(a,headLight,Off)

    % Stop robot movement
    direction='s';
    speed=Off;
    driveDirection(a,b,direction,speed)

    % determine which LED to turn on
    if i==1
        % Turn ON A LED
        writeDigitalPin(b,LED2,On)
        %input('Object 1 idetified')
    elseif i==2
        writeDigitalPin(b,LED3,On)
    else
        writeDigitalPin(b,LED4,On)
    end

    % Turn on tail lights for safety
    writeDigitalPin(b,tailLights,On)
```

```
    % Reverse Drive (Slow Mode)
    direction='b';
    speed=halfSpeed;
    driveDirection(a,b,direction,speed)
    pause(backUpTime)% ALLOWS IT TO MOVE BACK WHILE NOT SCANNING

    % Scan Object while moving Backwards
    Distance=straightScan(b,sPin,ultrasonicObj);
    % backsup till desired distances
    while Distance<nextLeftPointDist
        Distance=straightScan(b,sPin,ultrasonicObj);
    end

    % Turn off tail lights
    writeDigitalPin(b,tailLights,Off)

    % Stop robot movement
    direction='s';
    speed=Off;
    driveDirection(a,b,direction,speed)

end
```

```
function FirstLeftPointDist=rotateBotLeftPoint(a,b,sPin,ultrasonicObj,scanAngVol, ↙
scanAngleFlip,scanAngleRad)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: rotateBotLeftPoint
%   Source Files: radarScanToInf, straightScan
%   Description: This function scans objects and lines robot up with the
%   object
%   Input: a- arduino bottom, b - arduino top, ultrasonicObj - ultrasonic
%   object, scanAngVol - vector of voltages for servo, scanAngleFlip -
%   vector of angles in classic direction, scanAngleRad - vector of scan
%   angles in radians
%   Output: FirstLeftPointDist - First left point distance
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Turn Speed
turnSpeed=5;

% Error +- in inches for loctating objects
error=1.5;

% Middle Point Voltage
MidPoint= 0.5;

% Off
Off=0;

% Number of points scanned
numOfElementsInVec=length(scanAngVol);

% Scan for objects
[scanDistance]=radarScanToInf(b,sPin,ultrasonicObj,scanAngVol);  % Call scan function

% Find first point that is not infinity and store it it as i
i=1;
while scanDistance(i)==inf && (i+1)<numOfElementsInVec
    i=i+1;
end


% Checks if first left point is all the way right
if  i==13
    FirstLeftPointDist=inf;
else
    % adds one to i so left most point is more centered on object
    FirstLeftPointDist=scanDistance(i+1);
end
```

```
% check if the point is inf
if FirstLeftPointDist==inf
    % Rescan by rerunning this function
    FirstLeftPointDist=rotateBotLeftPoint(a,b,sPin,ultrasonicObj,scanAngVol,↙
scanAngleRad,scanAngleFlip);

else

    % Flips angle for reading and understanding
    objAngDeg=scanAngleFlip(i+1);

    % Plots the objects
    figure
    polarplot(scanAngleRad,scanDistance,'o')
    title('Radar Scan')

    % Board B
    s=servo(b,sPin);
    writePosition(s,MidPoint);
    clear s


    % Determine direction bot should turn
    if  objAngDeg==90
        % Drive staright

    elseif objAngDeg>90
        % rotate bot counterclockwise
        currentDis=straightScan(b,sPin,ultrasonicObj);

        % Set direction to ccw for when its greater then 90
        direction='ccw';
        speed=turnSpeed;
        driveDirection(a,b,direction,speed)

        % Turns till it reads distance within the given error
        while currentDis > (FirstLeftPointDist+error) || currentDis <↙
(FirstLeftPointDist-error)% ERROR SHOULD INCREASE FOR FUTHER CLOSE POINT
            currentDis=straightScan(b,sPin,ultrasonicObj);
        end

        % stopping bot
        direction='s'; %Stop
        speed=Off;
        driveDirection(a,b,direction,speed)

    else
        % objAngDeg<90 so turn clockwise
```

```
        currentDis=straightScan(b,sPin,ultrasonicObj);

        % Set direction to cw for when its less then 90
            direction='cw';
            speed=turnSpeed;
            driveDirection(a,b,direction,speed)

        % Turns till it reads distance within the given error
        while currentDis > (FirstLeftPointDist+error) || currentDis <↙
(FirstLeftPointDist-error)
            currentDis=straightScan(b,sPin,ultrasonicObj);
        end

        % Stopping bot
        direction='s'; %Stop
        speed=Off;
        driveDirection(a,b,direction,speed)
    end

end
```

```
function driveDirection(a,b,direction,Speed)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: driveDirection
%   Source Files: none
%   Description: This function will take an inputted dricetion and run 4
%   motors to complete the stated direction
%   Input: a-ardunio bottom, b- arduino top, direction- string for
%   direction, speed- in volts
%   Output: none
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Set motor Pins
AIN1 = 'D13'; %direction one
AIN2 = 'D12'; %direction two
PWMA = 'D11'; % b( Back right ) / a( Front Right )
BIN1 = 'D8'; %direction one
BIN2 = 'D7'; %direction two
PWMB = 'D3'; % b( Back Left ) / a( Front Left )

% On Off
ON = 1;
OFF = 0;

switch direction

    case 'cw'

        % Turning code Clockwise

        writeDigitalPin(a,AIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,AIN2 ,OFF) %set AIN2 to 0V (LOW)
        writeDigitalPin(a,BIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,BIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,AIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,AIN2 ,OFF) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,BIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,BIN2 ,ON) %set AIN2 to 0V (LOW)

        % Turn on Motors
        writePWMVoltage(b,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMA,Speed);%write a PWM voltage to the speed pin


    case 'ccw'
```

```
        % Turning code CounterClockwise

        writeDigitalPin(a,AIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,AIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(a,BIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,BIN2 ,OFF) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,AIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,AIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,BIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,BIN2 ,OFF) %set AIN2 to 0V (LOW)

        % Turn on Motors
        writePWMVoltage(b,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMA,Speed);%write a PWM voltage to the speed pin



    case 'f'

        % Forward

        writeDigitalPin(b,AIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,AIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(a,AIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,AIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,BIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,BIN2 ,ON) %set AIN2 to 0V (LOW)
        writeDigitalPin(a,BIN1, OFF) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,BIN2 ,ON) %set AIN2 to 0V (LOW)

        % Turn on Motors
        writePWMVoltage(b,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMA,Speed);%write a PWM voltage to the speed pin


    case 'b'

        % Backwards

        writeDigitalPin(b,AIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,AIN2 ,OFF) %set AIN2 to 0V (LOW)
        writeDigitalPin(a,AIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,AIN2 ,OFF) %set AIN2 to 0V (LOW)
        writeDigitalPin(b,BIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(b,BIN2 ,OFF) %set AIN2 to 0V (LOW)
```

```
        writeDigitalPin(a,BIN1, ON) %set AIN1 to 5V (HIGH)
        writeDigitalPin(a,BIN2 ,OFF) %set AIN2 to 0V (LOW)

        % Turn on Motors
        writePWMVoltage(b,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMA,Speed);%write a PWM voltage to the speed pin


    case 's'

        % Stop

        Speed=0;
        writePWMVoltage(a,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(a,PWMB,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMA,Speed);%write a PWM voltage to the speed pin
        writePWMVoltage(b,PWMB,Speed);%write a PWM voltage to the speed pin


end
```

```
function Bumped = fsrBumpSensor(a,FSRPin1,FSRPin2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: fsrBumpSensor
%   Source Files: none
%   Description: This function checks if the bump sensor is activated
%   Input: a- arduino bottom, FSRPin1 - One fsr Sensor, FSRPin2 - Second
%   fsr sensor
%   Output: Bumped - Returns value if bumped
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Min required to be activated for bump sensor
% ( Must be greater then force applied from accelerating)
grams1=1.4; % Volts
grams2=1.1; % Volts

% Read voltages
Vol1=readVoltage(a,FSRPin1); % Call voltage
Vol2=readVoltage(a,FSRPin2); % Call voltage

% pause time
dt=.01;

% Either voltage is greater then accepted value then produces value of 1
if Vol1>grams1 || Vol2>grams2
    pause(dt)

    % Requires 2 consecutive readings to elimnate error
    Vol1=readVoltage(a,FSRPin1); % Call voltage
    Vol2=readVoltage(a,FSRPin2); % Call voltage

    % Returns value that its bumped
    if Vol1>grams1 || Vol2>grams2

        Bumped=1;

    else
        Bumped=0;
    end
else
    Bumped=0;
end
```

```
function Distance=straightScan(b,sPin,ultrasonicObj)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: straightScan
%   Source Files: none
%   Description: This function sets the ultrasonic sensor straight and
%   rescans objects
%   Input:  b - arduino top, sPin - servo Pin, ultrasonicObj - ultrasonic
%   object
%   Output: Distance - distance in inches
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Board B - Declare servo
s=servo(b,sPin);

% MidPoint, Voltage for servo at 90 degrees straight for robot
MidPoint=.5;

% Time for servo to settle
dt=0.05;

% Sets servo to middle
straight=MidPoint;
writePosition(s,straight);
pause(dt)

%scan Distance
distanceMeter=readDistance(ultrasonicObj);
Distance=distanceMeter*39.3700787; % Converts to inches

% clears servo
clear s
```

```
function FirstLeftPointDist=findNewObject(a,b,sPin,ultrasonicObj,scanAngVol,↙
scanAngleRad,scanAngleFlip)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: findNewObject
%   Source Files: radarScanToInf, findNewObject, straightScan
%   Description: This function scans objects after one has been detected
%   and lines robot up with the object
%   Input: a- arduino bottom, b - arduino top, ultrasonicObj - ultrasonic
%   object, scanAngVol - vector of voltages for servo, scanAngleFlip -
%   vector of angles in classic direction, scanAngleRad - vector of scan
%   angles in radians
%   Output: FirstLeftPointDist - First left point distance
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Error +- in inches for object scanning
error=.7;

% calculates voltage for each scan angle
turnSpeed=5;

% Scans for objects and returns vector of distances
[scanDistance]=radarScanToInf(b,sPin,ultrasonicObj,scanAngVol);

% Elimnates distances between 180-60 to prevent rescanning objects
scanDistance(1:9)=inf;

% Plot
polarplot(scanAngleRad,scanDistance,'o')
title('Radar Scan')

% Find first point that is not infinity and store it it as i
i=1;
while scanDistance(i)==inf && ( i+1)<arrayLen
    i=i+1;
end


% Checks if first left point is all the way right
if  i==13
    FirstLeftPointDist=inf;
else
    % adds one to i so left most point is more centered on object
    FirstLeftPointDist=scanDistance(i+1);
end

% check if the point is inf
if FirstLeftPointDist==inf
```

```
    % Rescan by rerunning this function
    FirstLeftPointDist=findNewObject(a,b,sPin,ultrasonicObj,scanAngVol,scanAngleRad, ↙
scanAngleFlip);

else

    % Flips angle for reading and understanding
    objAngDeg=scanAngleFlip(i+1);

    % Plots the objects
    figure
    polarplot(scanAngleRad,scanDistance,'o')
    title('Radar Scan')

    % Board B
    s=servo(b,sPin);
    writePosition(s,MidPoint);
    clear s


    % Determine direction bot should turn
    if  objAngDeg==90
        % Drive staright

    elseif objAngDeg>90
        % rotate bot counterclockwise
        currentDis=straightScan(b,sPin,ultrasonicObj);

        % Set direction to ccw for when its greater then 90
        direction='ccw';
        speed=turnSpeed;
        driveDirection(a,b,direction,speed)

        % Turns till it reads distance within the given error
        while currentDis > (FirstLeftPointDist+error) || currentDis < ↙
(FirstLeftPointDist-error)% ERROR SHOULD INCREASE FOR FUTHER CLOSE POINT
            currentDis=straightScan(b,sPin,ultrasonicObj);
        end

        % stopping bot
        direction='s'; %Stop
        speed=Off;
        driveDirection(a,b,direction,speed)

    else
        % objAngDeg<90 so turn clockwise
        currentDis=straightScan(b,sPin,ultrasonicObj);

        % Set direction to cw for when its less then 90
```

```
        direction='cw';
        speed=turnSpeed;
        driveDirection(a,b,direction,speed)

        % Turns till it reads distance within the given error
        while currentDis > (FirstLeftPointDist+error) || currentDis <↙
(FirstLeftPointDist-error)
            currentDis=straightScan(b,sPin,ultrasonicObj);
        end

        % Stopping bot
        direction='s'; %Stop
        speed=Off;
        driveDirection(a,b,direction,speed)
    end

end
```

```
function [scanDistance]=radarScanToInf(b,sPin,ultrasonicObj,scanAngVol)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: radarScanToInf
%   Source Files:None
%   Description: This function will take a vector of scan angles and move a
%   ultrasonic sensor to the angle and take distance data
%   Input: b-ardunio top, sPin-servo pin, ultrasonicObj- Pin for ultrasonic
%   sensor, scanAngVol- vector of voltages for scan angles
%   Output: scanDistance- vector of distances at different angles
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pause time between readings
time=0.25;

% Any objects detected farther then this will be written as inf
OutsideRange=40;

% Board B declares servo
s=servo(b,sPin);

% allocates memory
arrayLen=length(scanAngVol);
scanDistance=ones(arrayLen,1);

% Scans every 15 degrees and elimnates any distances greater then outside
% range
for i = 1:arrayLen
    writePosition(s,scanAngVol(i));
    pause(time) % settle time for servo
    distanceMeter=readDistance(ultrasonicObj);
    scanDistance(i)=distanceMeter*39.3700787; % Converts to inches
    % checks range
    if scanDistance(i)>OutsideRange
        scanDistance(i)=inf;
    end
end

% Clears servo object
clear s
```

```
function [scanDistance]=radarScan(b,sPin,ultrasonicObj,scanAngVol)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: radarScan
%   Source Files:None
%   Description: This function will take a vector of scan angles and move a
%   ultrasonic sensor to the angle and take distance data
%   Input: b-arduino top, sPin-servo Pin, ultrasonicObj- Pin for ultrasonic
%   sensor, scanAngVol- vector of scan Angles
%   vector of angles to be scanned
%   Output: scanDistance- vector of distances at different angles
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pause time
time=0.25;

% Board B intializing servo
s=servo(b,sPin);

% allocate memory
arrayLen=length(scanAngVol);
scanDistance=ones(arrayLen,1);

% Distance vectors
for i = 1:arrayLen
    writePosition(s,scanAngVol(i));
    pause(time)
    scanDistance(i)=scanDistFun(ultrasonicObj);
     % Converts to inches
end

% clear servo
clear s
```

```
function DistInch=scanDistFun(ultrasonicObj)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Main File: radarScan
%   Source Files:None
%   Description: This function will take a vector of scan angles and move a
%   ultrasonic sensor to the angle and take distance data
%   Input: ultrasonicObj- ultrasonic object for scanning
%   Output: DistInch- Distance scanned in inches
%   Author: Connor Furlong, Lizzy Kuhn, Colin Todd
%   Date: 5/22/2023
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Range for noise
OutsideRange=40;

% read distances
distanceMeter=readDistance(ultrasonicObj);
DistInch=distanceMeter*39.3700787; % Converts to inches

% Check if within range
if DistInch>OutsideRange
    DistInch=inf;
end
end
```