FRUIT PICKING ROBOT FINAL DESIGN DOCCUMENTATION

Group 7

Submitted: Feb 25, 2020

Prof. A. Samani

Alexander Cleator
Ziqin Shang
Qi Qi
Mingyang Xu

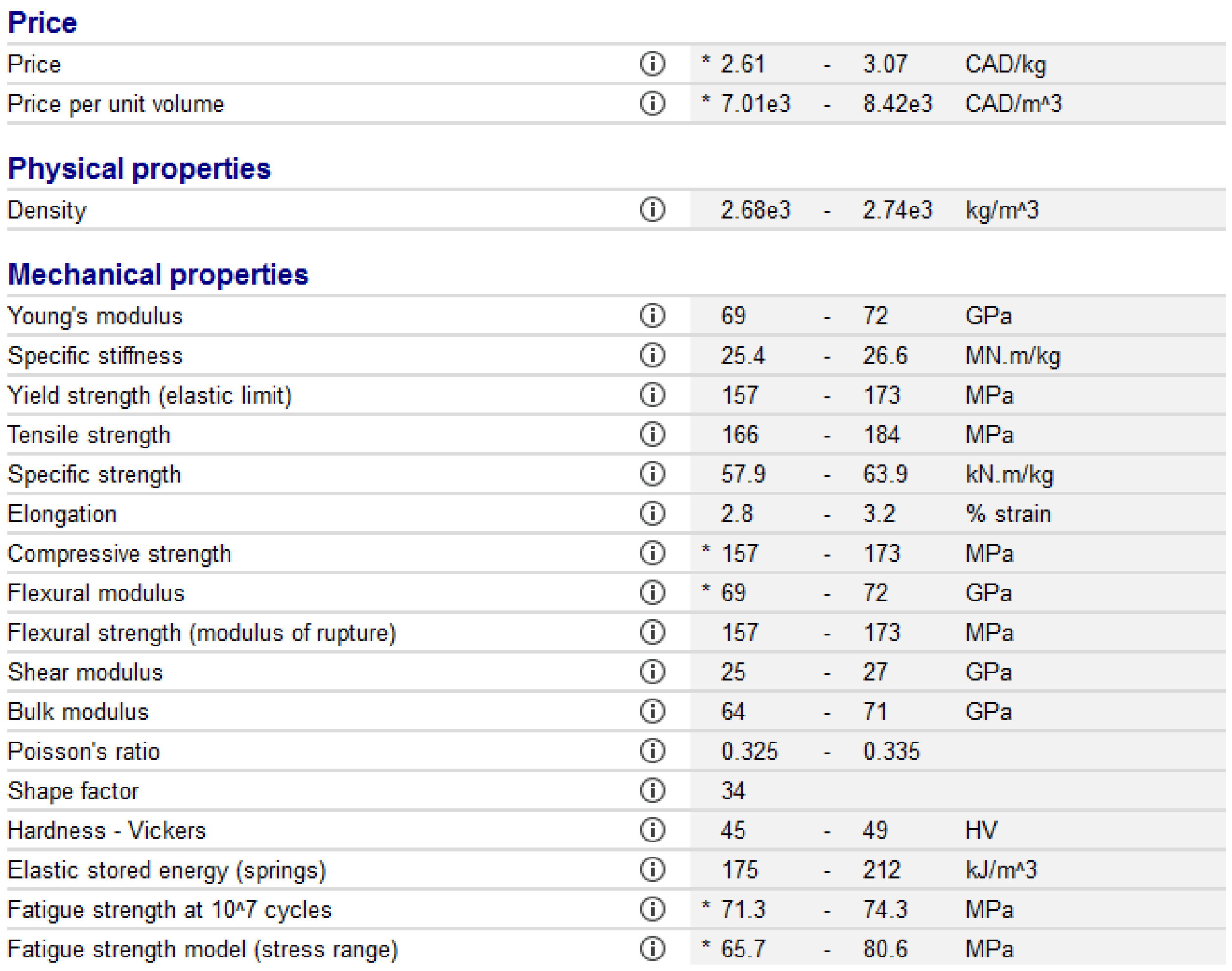
Table of Contents

Part1: Design Analysis

Frame Analysis

The frame is made from extruded 1060 aluminum alloy beams with following material properties found on CES 2019:

Table 1: 1060 Aluminum Properties



Price					
Price	ⓘ	* 2.61	-	3.07	CAD/kg
Price per unit volume	ⓘ	* 7.01e3	-	8.42e3	CAD/m^3
Physical properties					
Density	ⓘ	2.68e3	-	2.74e3	kg/m^3
Mechanical properties					
Young's modulus	ⓘ	69	-	72	GPa
Specific stiffness	ⓘ	25.4	-	26.6	MN.m/kg
Yield strength (elastic limit)	ⓘ	157	-	173	MPa
Tensile strength	ⓘ	166	-	184	MPa
Specific strength	ⓘ	57.9	-	63.9	kN.m/kg
Elongation	ⓘ	2.8	-	3.2	% strain
Compressive strength	ⓘ	* 157	-	173	MPa
Flexural modulus	ⓘ	* 69	-	72	GPa
Flexural strength (modulus of rupture)	ⓘ	157	-	173	MPa
Shear modulus	ⓘ	25	-	27	GPa
Bulk modulus	ⓘ	64	-	71	GPa
Poisson's ratio	ⓘ	0.325	-	0.335	
Shape factor	ⓘ	34			
Hardness - Vickers	ⓘ	45	-	49	HV
Elastic stored energy (springs)	ⓘ	175	-	212	kJ/m^3
Fatigue strength at 10^7 cycles	ⓘ	* 71.3	-	74.3	MPa
Fatigue strength model (stress range)	ⓘ	* 65.7	-	80.6	MPa

1060 Aluminum is a strong, lightweight material that suitable for various of applications, in this application, the strength of aluminum fulfilled the design requirement of supporting approximately 5kg of fruit and the 10kg mass of the robot. An FEA analysis was conducted with a loading scenario of 100N distributed force on the base plate, a fine mesh was used in this analysis. The results of the FEA analysis are shown in the following figures.

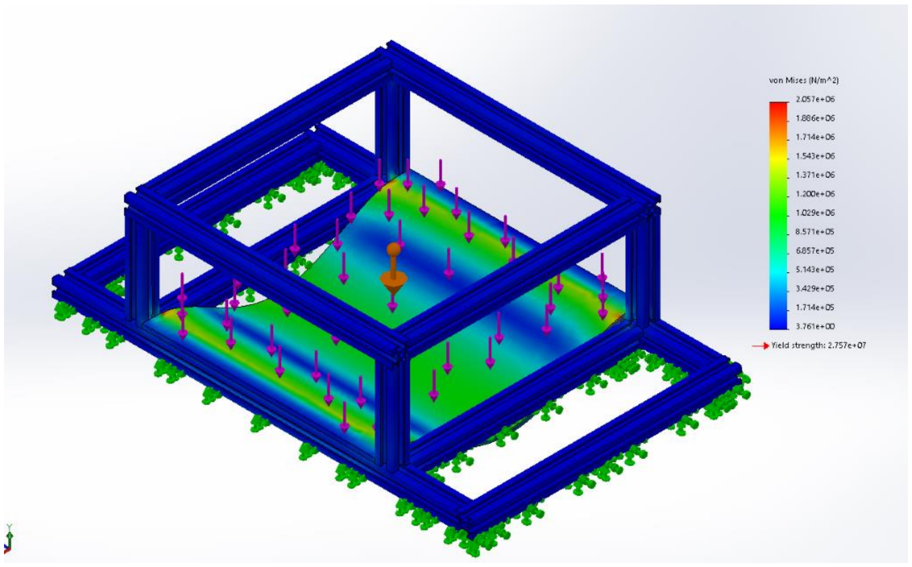


Figure 1: Von Mises Stress

It was found that the maximum Von Mises stress occurs on the contact edge of the plate and the frame, which is 2.057 MPa, the yield strength of 1060 aluminum is approximately 160 MPa, indicating no risk of failure by yielding.

The resultant displacement from the same analysis is shown below; it was found that the maximum displacement of 0.03mm occurred along the center line of the plate.

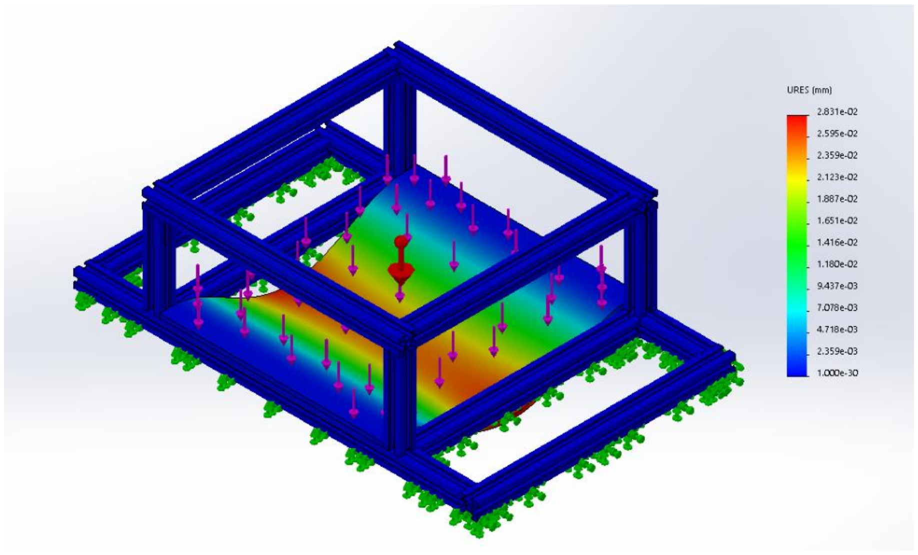


Figure 2: Deflection

Motor analysis

By estimating the mass of the robot to be 10kg, a normal force of 24.5N can be applied to each of the four wheels. Assuming the frictional coefficient between the wheels and the ground is 0.6, a frictional force of 14.7N must be overcome. Based on Newton's second law, a net force of 1N must be applied to allow the 10kg device to accelerate at a rate of 0.1 $\frac{m}{s^2}$.

Disregarding the moment of inertia of the wheel (as it was determined to be inconsequential), the required torque can be calculated as follows:

$$\tau = F\ x\ r = (15.7N)\ x\ (0.09m) = 1.41\ Nm$$

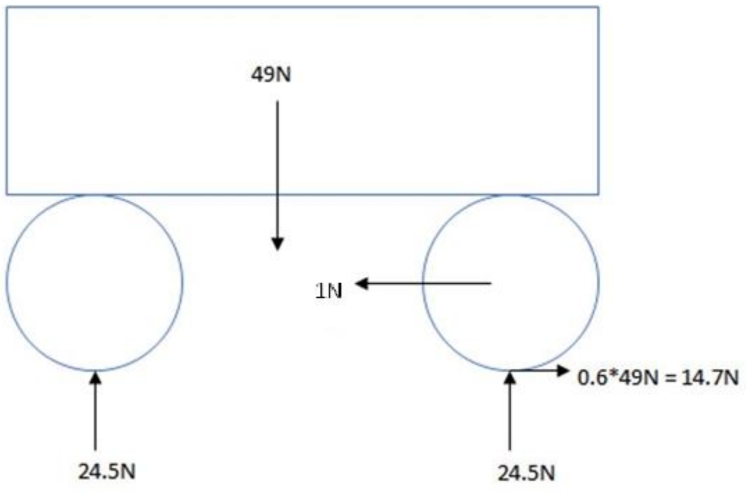


Figure 3: Chassis Static Force Analysis

Equating the applied force and friction, the required torque for continuous velocity can be determined as follows.

$$\tau_{steady\ state} = F_{friction} \times r_{wheel} = 14.7N \times 0.09m = 1.32Nm$$

By estimating the operational speed of $0.5\frac{m}{s}$, the required rotational speed can be calculated using the diameter of the wheel and speed of the robot:

$$circ. = \pi d = 0.18m\ x\ \pi = 0.565m$$

$$f = \frac{v}{circ}. = \frac{\frac{0.5m}{s}}{0.565m} = 0.884\ Hz$$

$$f_{rpm} = 60f = 60(0.884) = 53\ rpm$$

Considering the limited budget, a brushed DC motor is the optimal solution.

In this application, a gearbox is required because of the high speed, low torque direct output from the motor.

Based on the specification that the selected motor must have a stall torque under 1.4 Nm and a rated torque ideally as close as possible to 1.3Nm to maximize mechanical efficiency, a gearbox motor with the following properties was selected.

Table 2: DC Motor Properties

Voltage	No-load speed	Rated speed	Rated torque	No-load current	Rated current	Stall current
24V DC	325±30 RPM	230 ± 20 RPM	1.324N.m	0.3A	2.3A	7A

The motor has an integrated 1:27 gear reducer, which amplifies the rated torque from 50mNm to 1.324Nm that will fulfill the design requirement. A CNC machined steel shaft mount with an integrated bearing was also purchased to mount the motor on the chassis. A 180mm diameter wheel with aluminum wheel hub and rubber tire will be attached to the motor assembly.

The motor can be controlled using pulse width modulation, after a series of tests, a 15VDC supply, operating at a 30% duty cycle could achieve the specified rate of 0.5 $\frac{m}{s}$, a peak speed of 2.7$\frac{m}{s}$ was observed during testing, which would be implemented if not for safety considerations. For this device, rear differential steering was used due to the potential for zero-turn-radius steering. After completing the concept generation and selection phases of the design development, construction began on both a physical prototype and a Solidworks assembly to model the completed concept.

Navigational System analysis

The navigational system uses an 8-bit hall-effect based magnetic line tracker and a 1.5-inch magnetic strip. The magnetic tape is to be adhered to ground and line tracking sensor will detect the horizontal relative location of the tape. By inputting the location information into a microprocessor and using if-then logic, the speed of the two motor can be effectively controlled.

The magnetic sensor communicates serially based on MODBUS-RTU communication protocol or as well as outputting the states of each of the hall-effect sensors NPN open-drain configurations. After having difficulty establishing communication with the sensors MODBUS, tracking was achieved by polling each of the sensors independently. Using the internal pull-up resistors in the microprocessor, this information can be read directly through corresponding digital pins. After processing, the microprocessor outputs a 5V PWM signal, which is amplified to a nominal voltage using an integrated motor driver.

After a series of tests with different duty cycles and nominal voltages, following test results were measured.

Table 3:Driving Speed Testing

Nominal Voltage	PWM ratio	Speed
10V	30%	0.3m/s
10V	80%	0.85m/s
15V	30%	0.5m/s
15V	60%	1m/s
24V	30%	0.8m/s
24V	100%	2.7m/s

Based on the test results, it was found that the robot can travel at a maximum speed of 2.7$\frac{m}{s}$. By varying the nominal voltage and duty cycle, the speed of robot can vary from 0.3 $\frac{m}{s}$ to 2.7 $\frac{m}{s}$. The device was unable to drive if the nominal voltage applied was less than 10V irrespective of duty cycle, voltages above 24V were not tested to avoid potential damage to the motors being tested.

5 Degree of Freedom Manipulator

The manipulator operates using open loop controllers for each of the joints, which were coded by the manufacturer. There are unfortunately no feedback sensors integrated into the manipulator as shown in the manipulator's owner manual and communication protocol. This resulted in observable steady-state error in the z-direction when operating autonomously. After initial testing each joint was tested to move from -90^{o} to 90^{o} in 1 second with zero observable overshoot. From the appearance of the motion, the manipulator appears to use a cubic polynomial segmentation algorithm, however this is speculation.

The manipulator has a reachable workspace consisting of a dome with a radius of 36cm centered at a height of 9cm above the platform it is attached to.

Forward Kinematics

To model the manipulator, both forward and inverse kinematic models were generated. The forward kinematics were determined based on the following DH parameters, from which a series of transformation matrices were calculated for each of the joint locations. The theta values used were adjusted such that the commanded angles inherently programmed into the controller matched the measured joint angles observed. Based on the following parameters an accurate model of the manipulator kinematics was generated using MATLAB.

Table 4: DH Parameters

I	α_{i-1}	a_{i-1}	d_i	Θ_i
1	0	0	0.09	$\Theta_1 - \pi/2$
2	0	0	0	$\Theta_2 - \pi/2$
3	$-\pi/2$	0.11	0	Θ_3
4	0	0.09	0	Θ_4
5	0	0.16	0	0

Inverse Kinematics

The inverse kinematics of the manipulator were determined using an iterative algorithm called FABRIK, Forward and Backwards Reaching Inverse Kinematics developed at the Department of Engineering at Cambridge University, Cambridge, UK. The algorithm is best demonstrated in the following Figure 4 . This method is far simpler than others used as it does not require matrix manipulation, instead using simple geometry. This algorithm operates by positioning the end-effector at the desired positionFigure 4 (b) , drawing a line between that position and the next joint n-1, positioning joint n-1 a distance equal to the length of the link away from joint n along that line Figure 4 (c), and repeating for the remaining jointsFigure 4 (d). Then, the base joint is placed at its proper location Figure 4 (e). The same process repeats propagating outwards from the base towards the end effector. This cycle repeats until the base is located at its appropriate location and the end-effector is close enough to its desired position to be considered acceptable Figure 4 (f). Testing in MATLAB resulted in the end-effector being located within 0.1mm from the desired position, although higher accuracy would be possible with more iterations.

This algorithm works very well in solving planar manipulators; however, a slight modification was required to allow the manipulator to accurately model the manipulator used which is a planar manipulator with a rotating base. Simply using the coordinate frame of the base rather than the universal frame in this analysis resolved this issue. The code implementing this process, is also shown in the following section, with the joint numbering reversed to correspond with the indexing used by the manufacturers' controller.

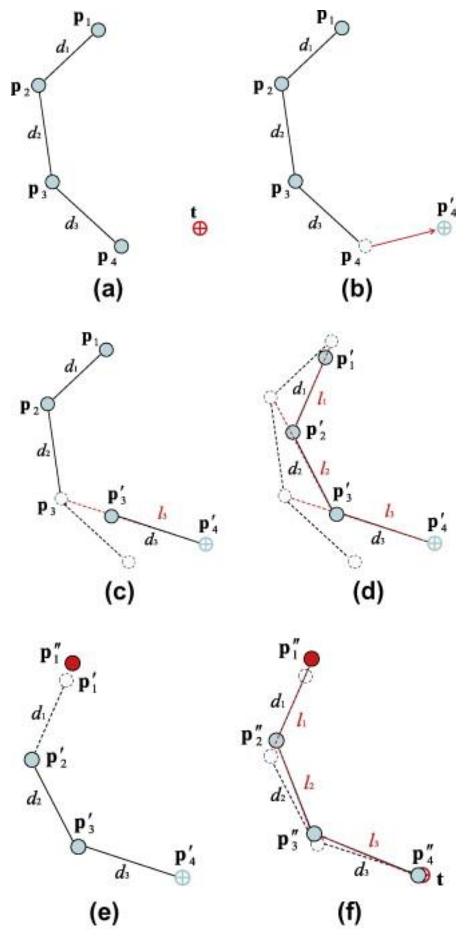


Figure 4: FABRIK Illustrated

The resolution of the manipulator is less precise than can allow for the 0.1mm accuracy determined by the FABRIK algorithm. Each joint has a resolution of 0.18^{o}, which, during testing corresponded to an accuracy of ±0.5 cm in the x and y directions. This is also due to inherent error in the controller provided, which does not appear to account for the weight of the manipulator based on the observed steady state error in the z direction.

End-effector Position Testing

The purpose of this test was to verify that the position of the end-effector commanded to the manipulator matched with that being observed through measurement. This testing was to ensure that the manipulator with the attached cutting head would be able to move to the desired location to effectively cut the tomato from the vine.

During testing, the servo in joint 4 stopped working. It would not move under any applied load, only operating under no-load conditions. As a result, the accuracy was greatly impacted as the manipulator being used effectively had three degrees of freedom. This failure also prevented the manipulator from operating within a 18cm dome centered on the base of the manipulator. The test results of commanding a series of positions are as follows.

Table 5: Left/Right Sweep

Commanded Position			Actual Position		
25	20	20	25	20	5
20	20	20	20	20	15
25	20	20	25	20	20
10	20	20	10	20	20
-10	20	20	-10	20	20
-20	20	20	-20	20	15
-25	20	20	-25	20	5

Table 6:Forward/Back Sweep

Commanded Position			Actual Position		
0	0	45	0	5	40
0	10	30	0	12	25
0	15	30	0	15	25
0	20	30	0	20	25
0	25	30	0	25	30
0	30	30	0	30	20
0	35	15	0	35	10

The x/y coordinates commanded were consistently accurate, however the measured z position was consistently less than the commanded position. This is likely a control issue, not accounting for the weight of the manipulator as the servos are not at their mechanical limits for any of the commanded positions. Furthermore, the manipulator has no integrated feedback devices, resulting in open loop control. This Prevents any sort of effective control without making significant modifications to the construction of the manipulator.

Manipulator Power Requirements

Table 7: Observed Power Requirements

Required Voltage	7.5V
Peak Current (during acceleration)	2.0A
Steady State Current (extended reach)	0.35A
Stead State Current (upright rest position)	0.2A

Based on the average AA battery, the battery pack of 5 AAs in series has a total 7.5V and a total capacity of approximately 2500 mAh. Assuming an average current of 0.3A, the battery pack should last approximately 8 hours before needing to be replaced or recharged. This battery life could be extended by placing multiple battery packs in parallel, or by using higher capacity lithium-ion batteries. However, for the purposes of a prototype, this was not attempted. For the final design, a single battery is to be used, along with a voltage regulator as having multiple batteries of different voltages is both redundant and poor design.

Tomato Recognition

In this project, machine vision is used to recognize tomatoes from complex backgrounds and return cartesian space information to robotics arm kinematics software for further processing. In completed design, the tomato recognition software would be fully integrated with ultrasonic and robotics arm kinematics to decide the movement of robotics arm.

The workflow of object recognition in this application is represented below:

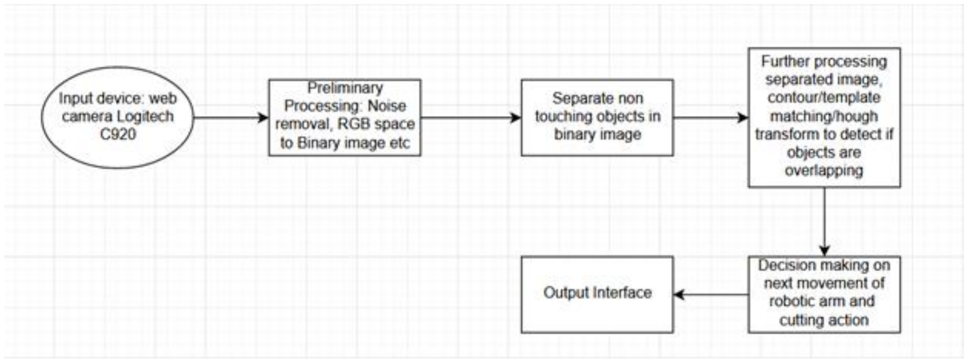

Input device: web camera Logitech C920
Preliminary Processing: Noise removal, RGB space to Binary image etc
Separate non touching objects in binary image
Further processing separated image, contour/template matching/hough transform to detect if objects are overlapping
Decision making on next movement of robotic arm and cutting action
Output Interface

Figure 5: Segmentation Algorithm

Because the algorithm is processing live video, the processing time for each frame must be less than the capture frame rate. The decision-making process also needs to be aware of the response time of the robotic arm and cutting action. The code implemented was developed using Matlab's advanced image processing toolbox and fast development nature. In production, the code will be ported to OpenCV. Current code written on Matlab attempts to utilize vectorization, memory pre-allocation and parallel processing to increase speed.

Input and Connection

Using Matlab Image Acquisition toolbox and standard 'winvideo' hardware adapter. Color space is set to RGB, resolution is set to 640x480 to reduce processing time. The capture framerate is set to 10 fps as it is fast enough for this application.

First, the RGB channels are combined into grayscale. Since the grape tomatoes are saturated red with a mix of green, brown (red + green) and blue-sky background, we decided to use a linear algorithm that promotes red and demotes green and blue to create the grayscale image. Grayscale=3*red channel-2*(blue channel+ green channel).

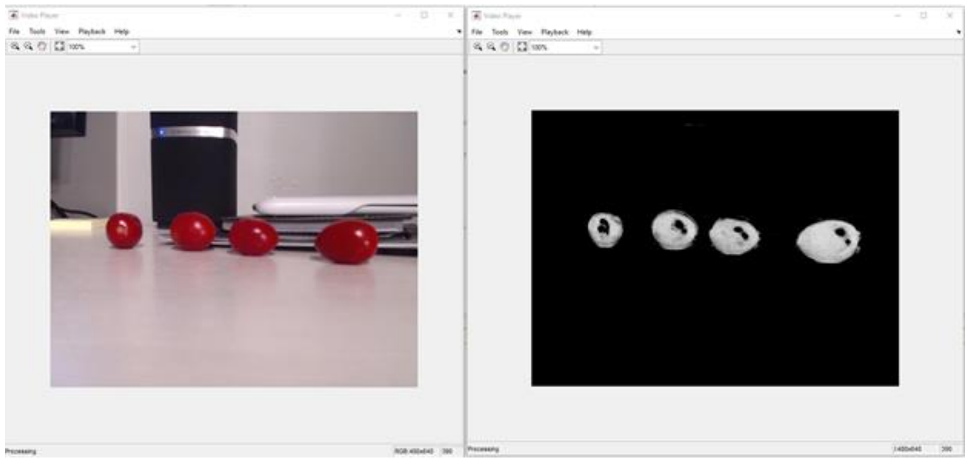


Figure 6: Initial Grayscale Results

Notice on the image, the holes inside the tomatoes are from overexposure areas due to the point light in the test environment. Filling the contour is required in later processing. To get a cleaner image and fill the hole, morphological reconstruction is used on the image to fill the small gaps and holes.

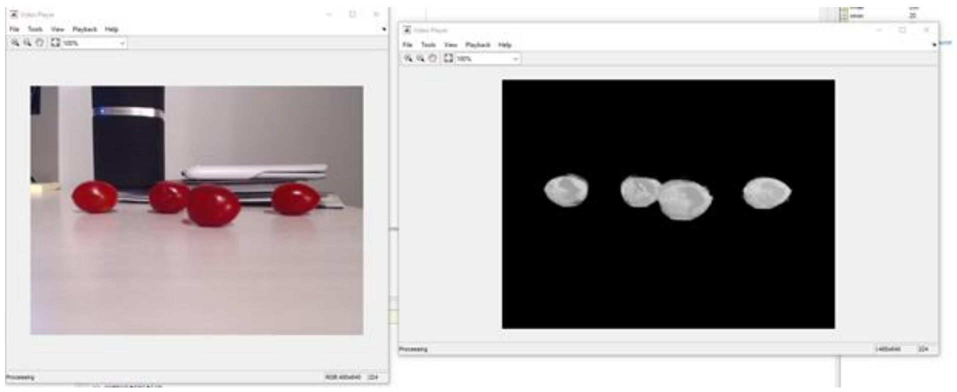


Figure 7: Morphologically Reconstructed Segmented Tomatoes

Each tomato is separated based on its position relative to the surrounding area. The top location of the tomato is added to location of the end-effector and then returned as the output of the function.

Emergency Stop Voice Recognition

As this robot is considerably dangerous, safety is an important consideration throughout the devices' development. When the user is close to the robot and they feel they are in danger of the robot's movement, for example he falls by accident in front of the robot or the robot is out of control. In these scenarios, the user can yell "Stop!" out loud. The microphone on the robot is constantly listening to the surrounding voice, it will capture the signal and compare the signal to the trained neural network based on TensorFlow and act accordingly. The voice recognition system consists of two parts: a preliminary band-pass filter and a trained neural network model.

Band-pass filter

3 different "stop" voice samples by different people are recorded and analyzed.

Their frequency and spectrographs are represented below:

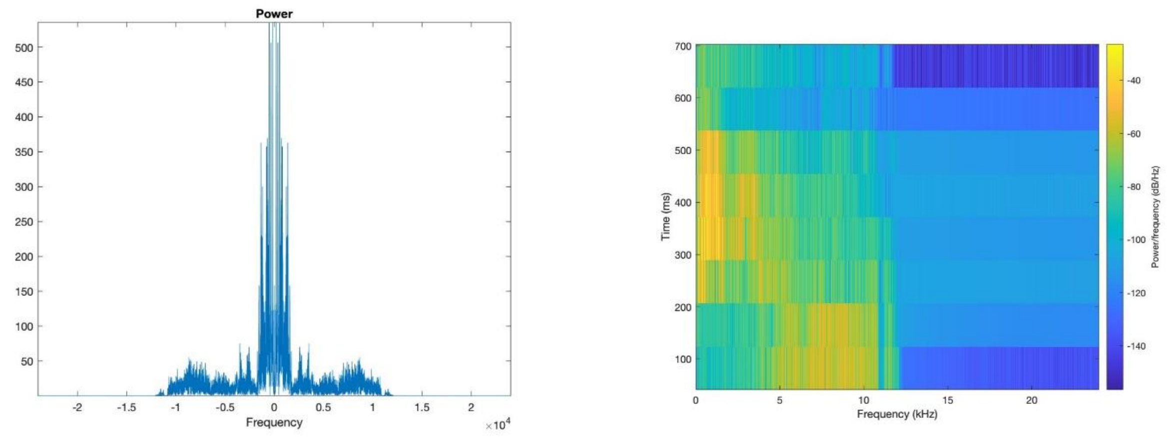


Figure 8: Test 1: Frequency and Spectrograph

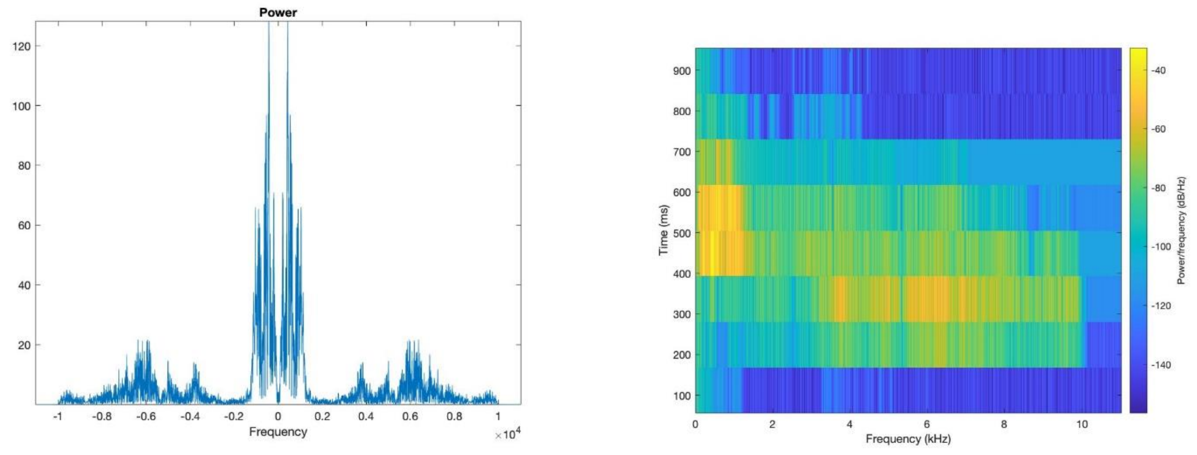


Figure 9:Test 2: Frequency and Spectrograph

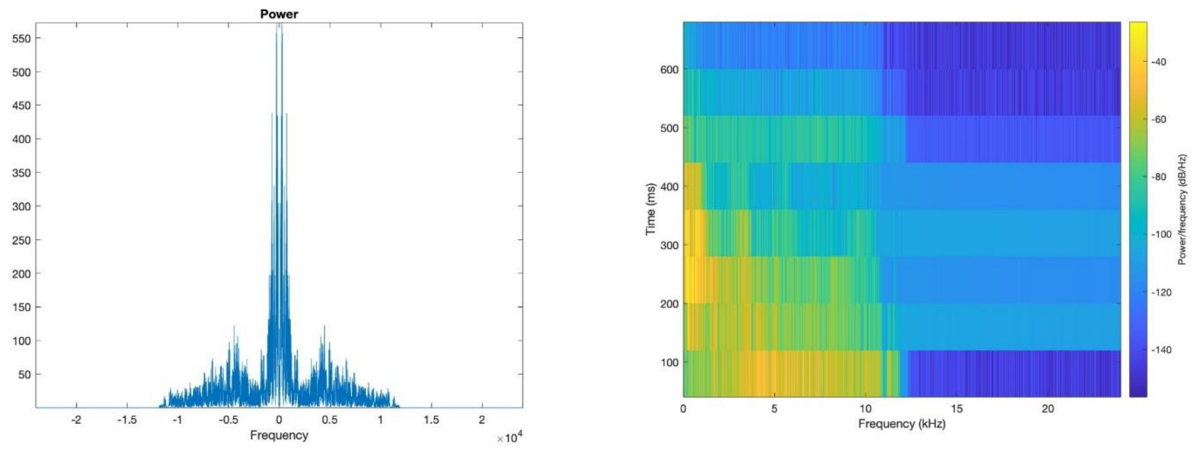


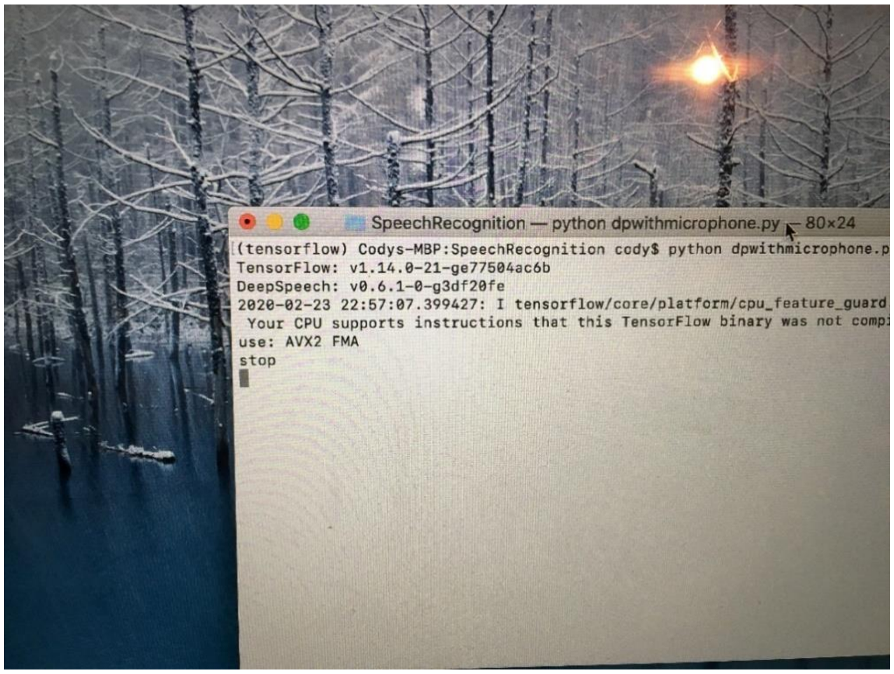
Figure 10: Test 3: Frequency and Spectrograph

From the graph, most power of the “s” sound ranges between 5kHz to 8kHz, “top” sound ranges below 400Hz and above 3kHz.

Based on the characteristic of “stop” sound, a Hamming windows FIR 400Hz-8kHz passband filter is implemented to reduce noise.

Neural network model

The neural network model is based on a Deepspeech model retrained with a focus on simple word commands such as “stop”. The databases used include: LibriSpeech, Voice Commands Dataset,and my owned recorded voices.



```
(tensorflow) Codys-MBP:SpeechRecognition cody$ python dpwithmicrophone.p
TensorFlow: v1.14.0-21-ge77504ac6b
DeepSpeech: v0.6.1-0-g3df20fe
2020-02-23 22:57:07.399427: I tensorflow/core/platform/cpu_feature_guard
 Your CPU supports instructions that this TensorFlow binary was not comp
use: AVX2 FMA
stop
```

Part 2: Detailed Design Documentation

Manipulator Code

The following are a series of scripts that are command the robot to move to a specified location, these scripts are triggered by the image processing code determining the location of a tomato. The manipulator moves to that location, closes the gripper (cutting the vine), reopens it, then returns to its original position until the script is called again.

Serial_Comm.m

This script initializes the device, determines the required joint angles to move the end-effector to a desired location in cartesian coordinates, then executes those same joint movements.

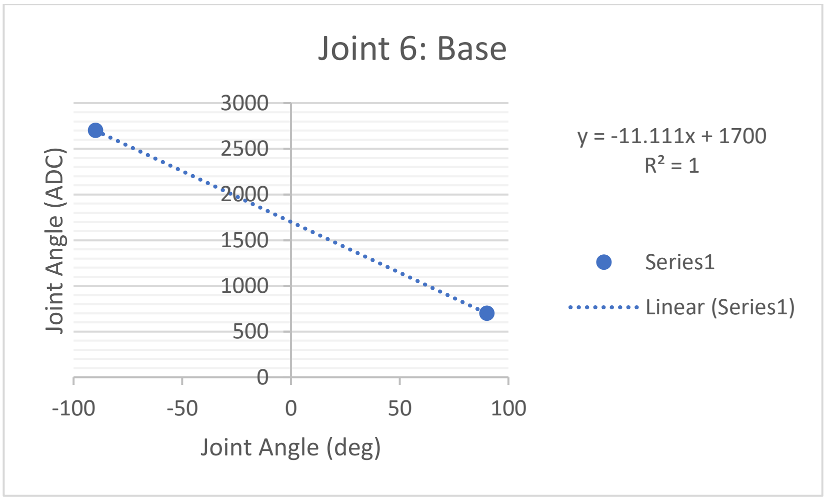
```
clc
clear
pause on
%Reset the device before initialzing communications
instrreset
s = serial('COM5');
s.baud = 9600;
fopen(s);
s;

%Determine the required joint angles and write them to the manipulator
%sequentially
resetServo(s)

Joint = invKinFcn(tomX;tomY;tomZ])
MoveAllServo(s,1500,joint(1),joint(2), joint(3), joint(4),1500,1500)
Pause(1.5);
MoveAllServo(s,1500,joint(1),joint(2), joint(3), joint(4),1500,2000)
Pause(1.5);
MoveAllServo(s,1500,joint(1),joint(2), joint(3), joint(4),1500,1500)

resetServo(s)
```

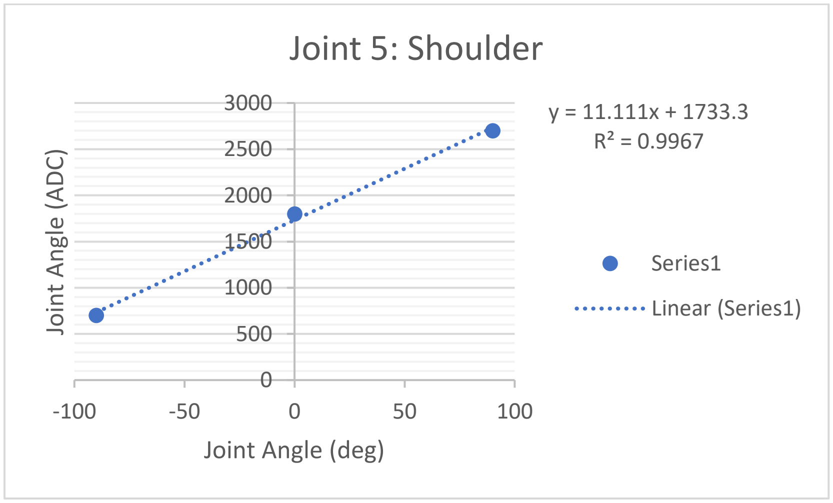
MoveServo.m

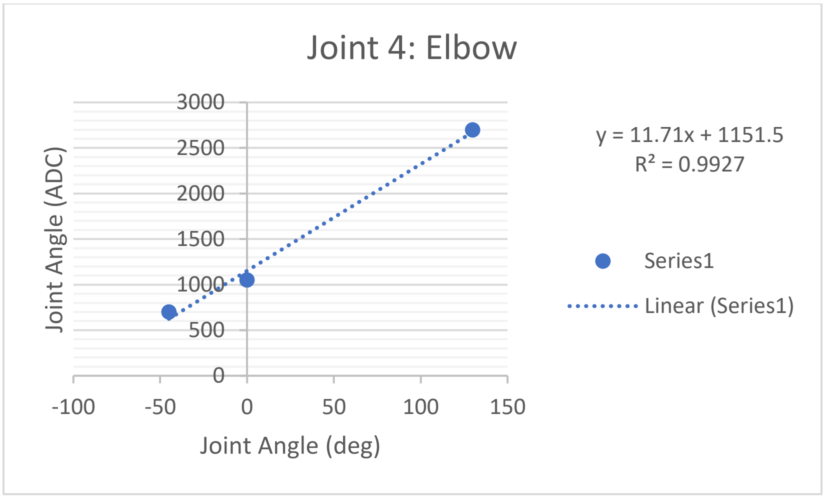
This script moves a single servo to a specified position using serial communication. It first converts the desired position from degrees to a position value that can be interpreted by the manipulator before separating bother the position and time values into 16-bit unsigned integers and writing those values to the manipulator.

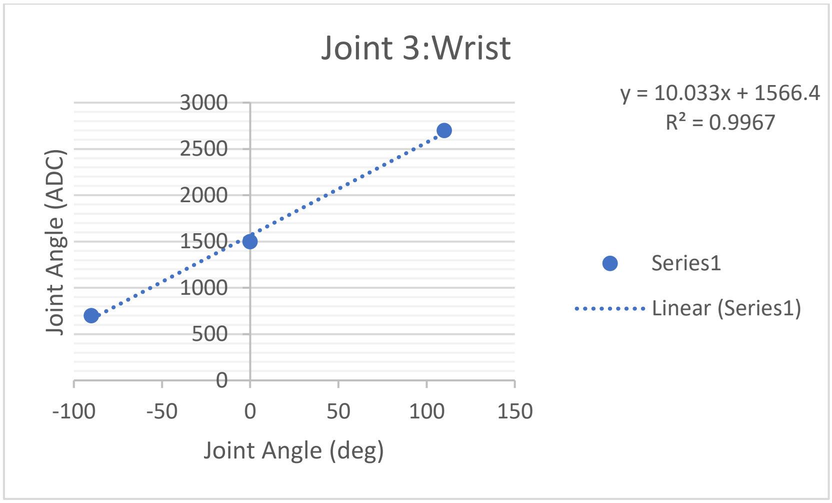
```
function [] = moveServo(device,ServoID,Time,Position)

%   Commands a servo to move to a specified position in a specified time
%Linearize the position in degrees to an ADC value to be sent to the
%manipulator
if ServoID == 6
   outPosition = -11.111*Position + 1700;

end
if ServoID == 5
    outPosition = 11.111*Position + 1733.3;
end
if ServoID == 4
     outPosition = 11.71*Position + 1151.5;
end
if ServoID == 3
    outPosition = 10.033*Position + 1566.4;
end
if ServoID == 2 || ServoID == 1
    outPosition = Position;
end

buf(1) = uint8(hex2dec('55'));
buf(2) = uint8(hex2dec('55'));
%length
buf(3) =  uint8(hex2dec('08'));
%command
buf(4) = uint8(hex2dec('03'));
%# of servos to be controlled
buf(5) = uint8(1);
% t
buf(6) = uint8(Time & 255);
buf(7) = uint8(floor(Time/16/16));
%sevo ID
buf(8) = uint8(ServoID);
%Servo position
buf(9) = uint8(outPosition & 255);
buf(10) = uint8(floor(outPosition/16/16));

%Write buffer to the manipulator
fwrite(device,buf)
```

Linearization is achieved by measuring the joint angle of each of the servos in degrees and plotting that against the digital value commanded to achieve that position. A linear regression was calculated, and the corresponding equations were used to control the manipulator.

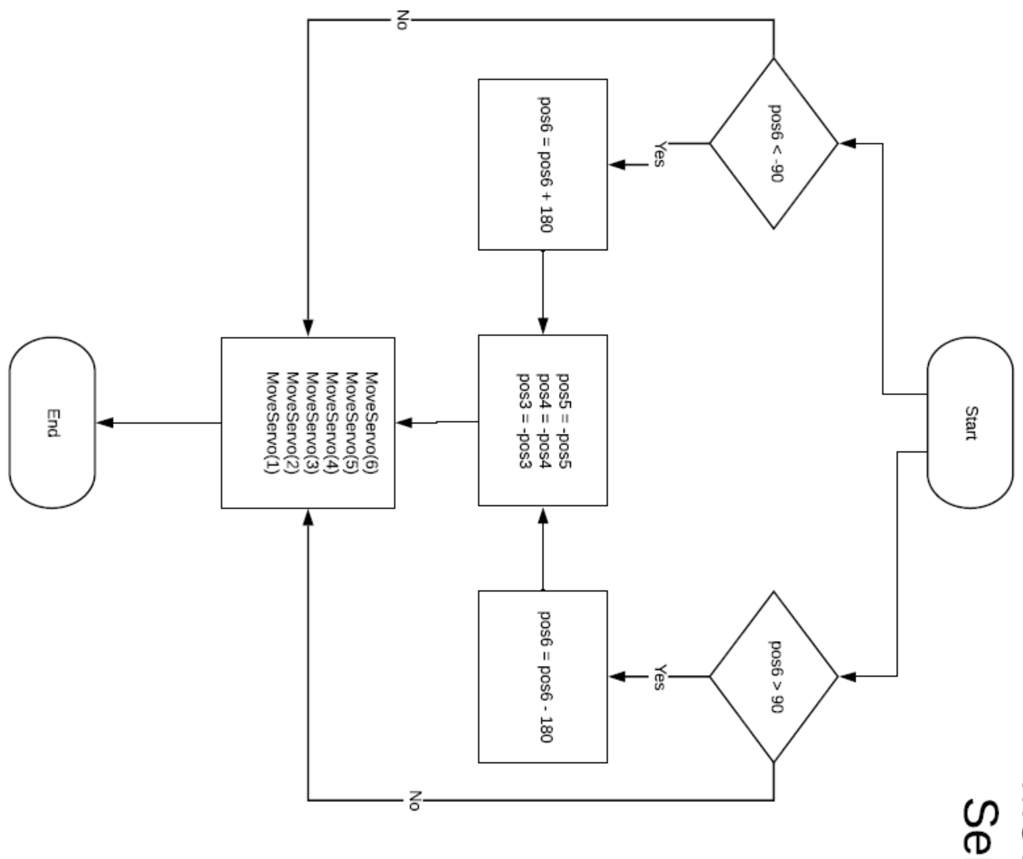




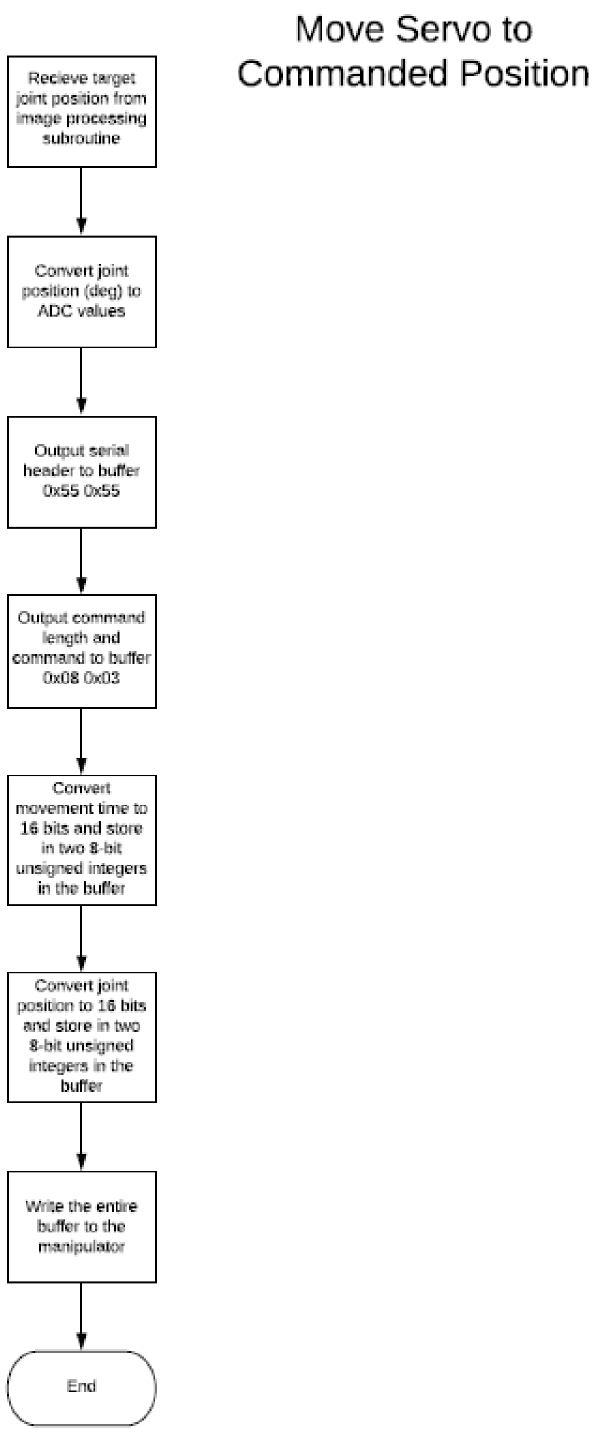




Joint 6		**Joint 5**		**Joint 4**		**Joint 3**	
Deg	ADC	**Deg**	ADC	**Deg**	ADC	**Deg**	ADC
90	700	**-90**	700	**-45**	700	**-90**	700
-90	2700	**0**	1800	**0**	1050	**0**	1500
		90	2700	**130**	2700	**110**	2700



Move All Servos


Move Servo to Commanded Position
Recieve target joint position from image processing subroutine
Convert joint position (deg) to ADC values
Output serial header to buffer 0x55 0x55
Output command length and command to buffer 0x08 0x03
Convert movement time to 16 bits and store in two 8-bit unsigned integers in the buffer
Convert joint position to 16 bits and store in two 8-bit unsigned integers in the buffer
Write the entire buffer to the manipulator
End

InvKinFcn.m

This function is a modified implementation of the FABRIK algorithm (Forward and Backwards Reaching Inverse Kinematics). This technique was published in a 2011 issue of the Graphical Models journal and was developed by Andreas Aristidou and Joan Lasenby at the Cambridge University Department of Engineering. This algorithm was modified to be used for the planar manipulator implemented by the fruit picking robot by rotating the coordinate system attached at the base of the robot to match the universal coordinate system.

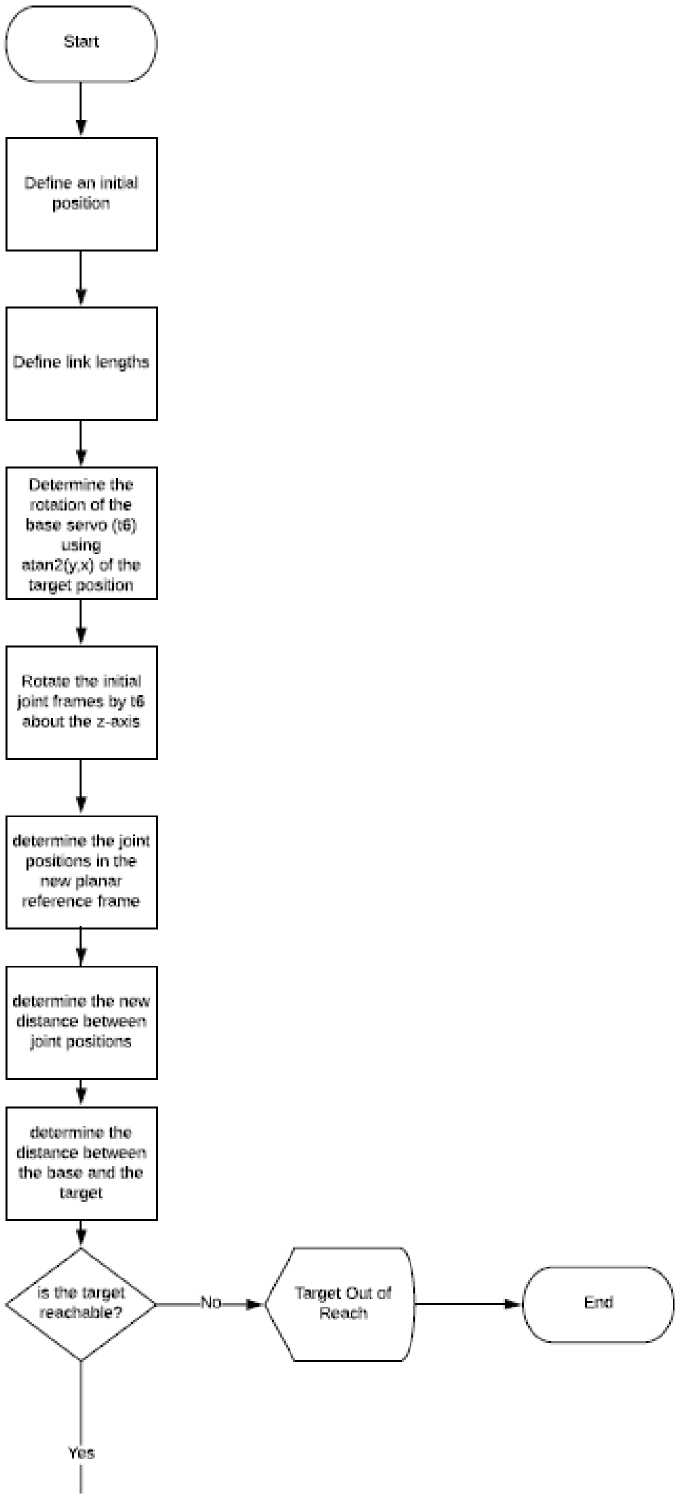
```
function [ theta ] = invKinFcn( t )
%Inverse Kinematics function using FABRIK: Position control only
p0 = [0;0;0.09];
p1 = [0;0;0.09];
p2 = [0;0;0.09];
p3 = [0;0;0.29];
p4 = [0;0;0.29];
p5 = [0;0;0.29];
p6 = [0;0;0.45];
p = [p0,p1,p2,p3,p4,p5,p6];

%define link lengths, d2, d5 defined as non-zero to avoid signular division
%by zero error
d1 = 0.09;
d2 = 0.0001;
d3 = 0.29;
d4 = 0.0001;
d5 = 0.0001;
d6 = 0.16;

%rotate frame by t1 to create planar problem
t6 = atan2(t(2),t(1));
T01= [cos(t6), -sin(t6),0,0; sin(t6),cos(t6),0,0;0,0,1,0; 0,0,0,1];

%define target and rotate intitial joint positions to new plane
target = T01*[t;1];
p1 = T01*[p1;1];
p2 = T01*[p2;1];
p3 = T01*[p3;1];
p4 = T01*[p4;1];
p5 = T01*[p5;1];
p6 = T01*[p6;1];

%simplify position vectors to x/y for 2d plotting
target = target(2:3,1);
p1 = p1(2:3,1);
p2 = p2(2:3,1);
p3 = p3(2:3,1);
```

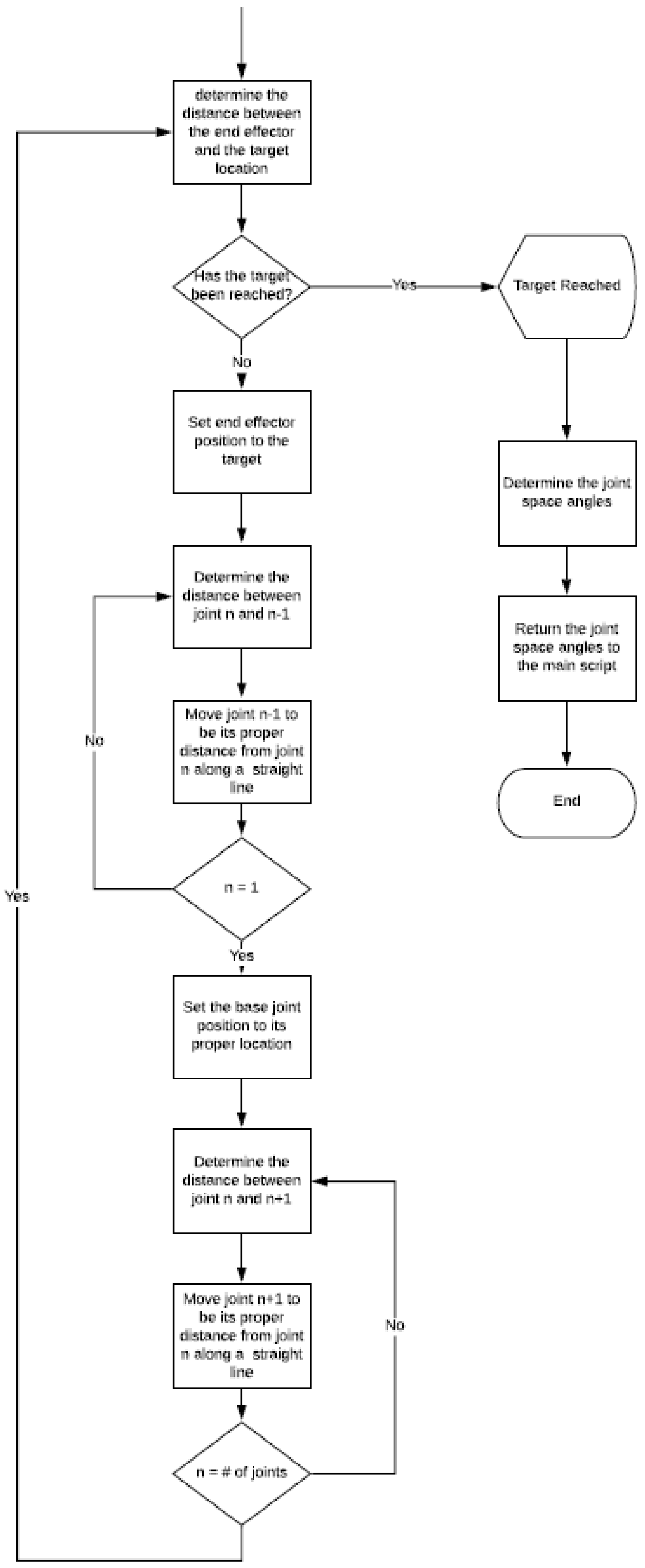
```
p4 = p4(2:3,1);
p5 = p5(2:3,1);
p6 = p6(2:3,1);

p = [[0;0.09],p1,p2,p3,p4,p5,p6];


%distance between joints
d = zeros(2:6);
dMag = zeros(6,1);
for i = 1:6
    d(:,i) = p(:,i+1) - p(:,i);
    dMag(i) = sqrt(d(1,i)^2 + d(2,i)^2);

    if dMag(i) == 0
        dMag(i) = 0.0001;
    end
end
dTarget = sqrt(target(1)^2 + target(2)^2);

%Determine if the target is withing reach
if dTarget > (d1 + d2 + d3 + d4 +d5 +d6)
    fprintf('%s\n','Target out of reach')
else
    fprintf('%s\n','Target in reach')

diffMag = 1;

while diffMag > 0.0001
     b = p(:,1);
      p(:,7) = target;

   for i = 6:-1:1
       %fprintf('%s\n','forward reach')
        r(:,i) = p(:,i+1) - p(:,i);
        rMag(i) = sqrt(r(1,i)^2 + r(2,i)^2);

            if rMag(i) == 0
               rMag(i) = 0.0001;
            end
        lambda(i) = dMag(i)/rMag(i);
        p(:,i) = (1 - lambda(i))*p(:,i+1) + lambda(i)*p(:,i);
    end

p(:,1) = b;

for i = 1:6
     %fprintf('%s\n','backwards reach')
    r(:,i) = p(:,i+1) - p(:,i);
    rMag(i) = sqrt(r(1,i)^2 + r(2,i)^2);

if rMag(i) == 0
   rMag(i) = 0.0001;
end
```

```
lambda(i) = dMag(i)/rMag(i);

p(:,i+1) = (1 - lambda(i))*p(:,i) + lambda(i)*p(:,i+1);
end

diff = p(:,7) - target;
diffMag = sqrt(diff(1)^2 + diff(2)^2);
end
end

zMatrix = zeros(3,7);
zMatrix(2:3,:) = p;

R01 = T01(1:3,1:3);

jointPos = zeros(3,7);
for i = 1:7
    jointPos(:,i) = R01'*zMatrix(:,i);
end

r = zeros(2,6);
rMag = zeros(1,6);
for i = 1:6;
    r(:,i) = p(:,i+1) - p(:,i);
    rMag(i) = sqrt(r(1,i)^2 + r(2,i)^2);

if rMag(i) == 0;
   rMag(i) = 0.0001;
end
end

%determine required joint angles using atan2
p;
pGood(:,1) = p(:,1);
pGood(:,2) = p(:,4);
pGood(:,3) = p(:,5);
pGood(:,4) = p(:,7);

dx(1) = pGood(1,2) - pGood(1,1);
dx(2) = pGood(1,3) - pGood(1,2);
dx(3) = pGood(1,4) - pGood(1,3);

dy(1) = pGood(2,2) - pGood(2,1);
dy(2) = pGood(2,3) - pGood(2,2);
dy(3) = pGood(2,4) - pGood(2,3);

t5 = atan2(dx(1),dy(1));
t4 = 0;
t3 = atan2(dx(3),dy(3)) - t4 - t5;

%output resultant angles, made negative to counteract opposite direction of
%frame of reference Z vector
theta = [rad2deg(t6),rad2deg(t5),rad2deg(t4),rad2deg(t3)];
end
```

FABRIK Inverse Kinematics Algorithm
Designed by Andreas Aristidou, Joan Lasenby. Published in Graphical Models (2011)

Modified by Alexander Cleator


Start
Define an initial position
Define link lengths
Determine the rotation of the base servo (t6) using atan2(y,x) of the target position
Rotate the initial joint frames by t6 about the z-axis
determine the joint positions in the new planar reference frame
determine the new distance between joint positions
determine the distance between the base and the target
is the target reachable?
No
Target Out of Reach
End
Yes


determine the distance between the end effector and the target location
Has the target been reached?
Yes
Target Reached
No
Set end effector position to the target
Determine the joint space angles
Determine the distance between joint n and n-1
Return the joint space angles to the main script
Move joint n-1 to be its proper distance from joint n along a straight line
No
End
n = 1
Yes
Yes
Set the base joint position to its proper location
Determine the distance between joint n and n+1
Move joint n+1 to be its proper distance from joint n along a straight line
No
n = # of joints

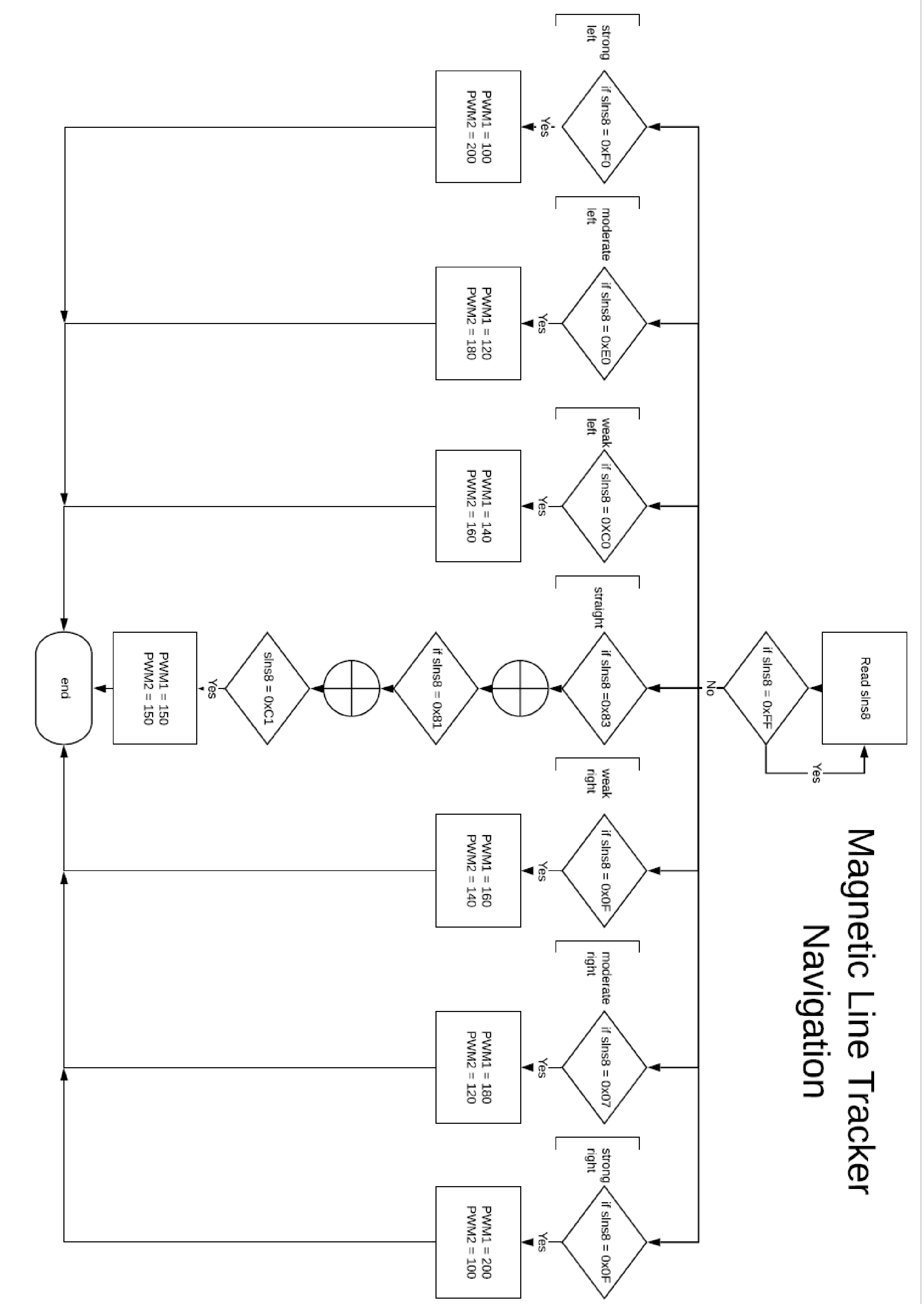
Chassis Navigation Code

The following code implements differential steering navigation based on the output of the line tracker sensor attached to the front of the device. If the device begins to turn to the left, the left wheel drives faster to ensure proper tracking, the same for the left wheel. The entire process is shown in the following flowchart.

```
int PWM1 = 10;
int INA1 = A2;
int INB1 = A4;
int PWM2 = 11;
int INA2 = A3;
int INB2 = A5;
int s1 = 0;//initial values for each pin on sensor
int s2 = 0;
int s3 = 0;
int s4 = 0;
int s5 = 0;
int s6 = 0;
int s7 = 0;
int s8 = 0;
int stopPin = 12;
void setup()
{
  Serial.begin(9600);
  pinMode(PWM1, OUTPUT);
  pinMode(INA1, OUTPUT);
  pinMode(INB1, OUTPUT);
  pinMode(PWM2, OUTPUT);
  pinMode(INA2, OUTPUT);
  pinMode(INB2, OUTPUT);
 pinMode(stopPin, INPUT);
  for (int i = 2; i <= 9; i++) pinMode(i, INPUT_PULLUP); // set pin 2-9 to input with pull up
}
void loop()
{
  s1 = digitalRead(2);
  s2 = digitalRead(3);
  s3 = digitalRead(4);
  s4 = digitalRead(5);
  s5 = digitalRead(6);
  s6 = digitalRead(7);
  s7 = digitalRead(8);
  s8 = digitalRead(9);
 stopFlag = digitalRead(stopPin);
  digitalWrite(INA1,LOW); //motor 1 and 2 turning forward
  digitalWrite(INA2,LOW); //motor 1: left motor
  digitalWrite(INB1,HIGH);   //motor 2: right motor
  digitalWrite(INB2,HIGH);
  if(s1==1 && s2==0 && s3==0 && s4==0 && s5==0 && s6==0 && s7==1 && s8==1){
    displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
    Serial.print("Going Straight");
    analogWrite(PWM1,75);
    analogWrite(PWM2,75);
  }
    if(s1==1 && s2==0 && s3==0 && s4==0 && s5==0 && s6==0 && s7==0 && s8==1){
    displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
    Serial.print("Going Straight");
    analogWrite(PWM1,75);
    analogWrite(PWM2,75);
```

```
}
 if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0 && s6==0 && s7==0 && s8==1){
 displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
 Serial.print("Going Straight");
 analogWrite(PWM1,75);
 analogWrite(PWM2,75);
}
 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==0 && s6==0 && s7==1 && s8==1){
 displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
 Serial.print("Going Straight");
 analogWrite(PWM1,75);
 analogWrite(PWM2,75);
}
 if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0 && s6==0 && s7==0 && s8==0){
 displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
 Serial.print("Turing Left slightly");
 analogWrite(PWM2,70);
 analogWrite(PWM1,80);
}
 if(s1==1 && s2==1 && s3==1 && s4==0 && s5==0 && s6==0 && s7==0 && s8==0){
 displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
 Serial.print("Turing Left moderately");
 analogWrite(PWM2,60);
 analogWrite(PWM1,90);
}
 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0 && s6==0 && s7==0 && s8==0){
 displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
 Serial.print("Turing Left intensely");
 analogWrite(PWM2,50);
 analogWrite(PWM1,100);
 }
  if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0 && s6==0 && s7==1 && s8==1){
  displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
  Serial.print("Turing Right slightly");
  analogWrite(PWM2,80);
  analogWrite(PWM1,70);
 }
  if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0 && s6==1 && s7==1 && s8==1){
  displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
  Serial.print("Turing Right moderately");
  analogWrite(PWM2,90);
  analogWrite(PWM1,60);
 }
  if(s1==0 && s2==0 && s3==0 && s4==0 && s5==1 && s6==1 && s7==1 && s8==1){
  displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
  Serial.print("Turing Right intensely");
  analogWrite(PWM2,100);
  analogWrite(PWM1,50);
 }
  if((s1==1 && s2==1 && s3==1 && s4==1 && s5==1 && s6==1 && s7==1 && s8==1)||stopFlag == 1){
  displaysensorvalue(s1,s2,s3,s4,s5,s6,s7,s8);
  Serial.print("Vehicle Stopped,no signal detected");
  digitalWrite(INA1,LOW);
  digitalWrite(INB1,LOW);
  digitalWrite(INA2,LOW);
  digitalWrite(INB2,LOW);
 }
 delay(100);

}
```

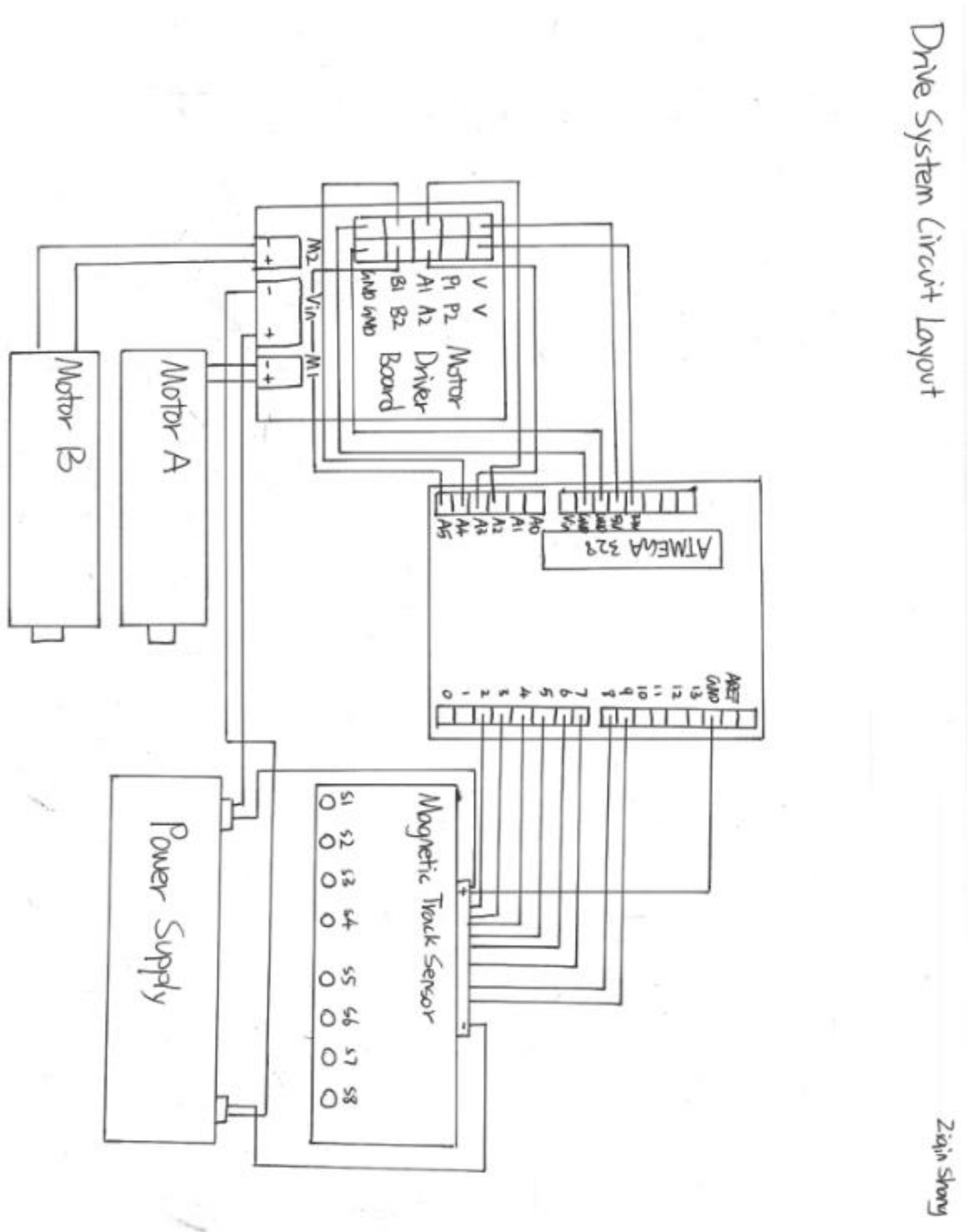
```
void displaysensorvalue(int s1,int s2,int s3,int s4,int s5,int s6,int s7,int s8){
 Serial.print(s1);
 Serial.print(s2);
 Serial.print(s3);
 Serial.print(s4);
 Serial.print(s5);
 Serial.print(s6);
 Serial.print(s7);
 Serial.print(s8);
 Serial.print("\n");
}
```


Magnetic Line Tracker Navigation
Read slns8
if slns8 = 0xFF
Yes
No
strong left
if slns8 = 0xF0
Yes
PWM1 = 100
PWM2 = 200
moderate left
if slns8 = 0xE0
Yes
PWM1 = 120
PWM2 = 180
weak left
if slns8 = 0XC0
Yes
PWM1 = 140
PWM2 = 160
straight
if slns8 =0x83
if slns8 = 0x81
slns8 = 0xC1
Yes
PWM1 = 150
PWM2 = 150
end
weak right
if slns8 = 0x0F
Yes
PWM1 = 160
PWM2 = 140
moderate right
if slns8 = 0x07
Yes
PWM1 = 180
PWM2 = 120
strong right
if slns8 = 0x0F
Yes
PWM1 = 200
PWM2 = 100

Chassis and Manipulator Bill of Materials

ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	2020 alu section 450mm		2
2	2020 alu section 260mm		5
3	2020 alu section362.78mm		3
4	2020 alu section 100mm		4
5	2020 alu section 95mm		4
6	2020 alu section 69mm		4
7	JQR03210-40ALCPU612		2
8	180mm wheel		2
9	DCmotor		2
10	motorholder		2
11	wheelshaftconnector		2
12	motorshaft		2
13	magneticsensorholder		1
14	container wall		3
15	container base plate		1
16	manipulatorbase		1
17	gripper base plate		1
18	spacer		3
19	drive gear		2
20	gripper		4
21	link		4
22	gripper mount		2
23	LFD-O6 Servo		3
24	servo mount		2
25	bearing		1
26	large U bracket		3
27	small U bracket		1
28	LDX-218 Digital Servo		3
29	base LDX-218 Servo Mount		1
30	upper base plate		1
31	base ring		3
32	large bearing		1
33	lower base plate		1
34	9mm column		4
35	15mm column		4
36	40 mm column		4
37	container wall with hole		1

Navigation System Wiring Diagram



Motor Driver Schematic

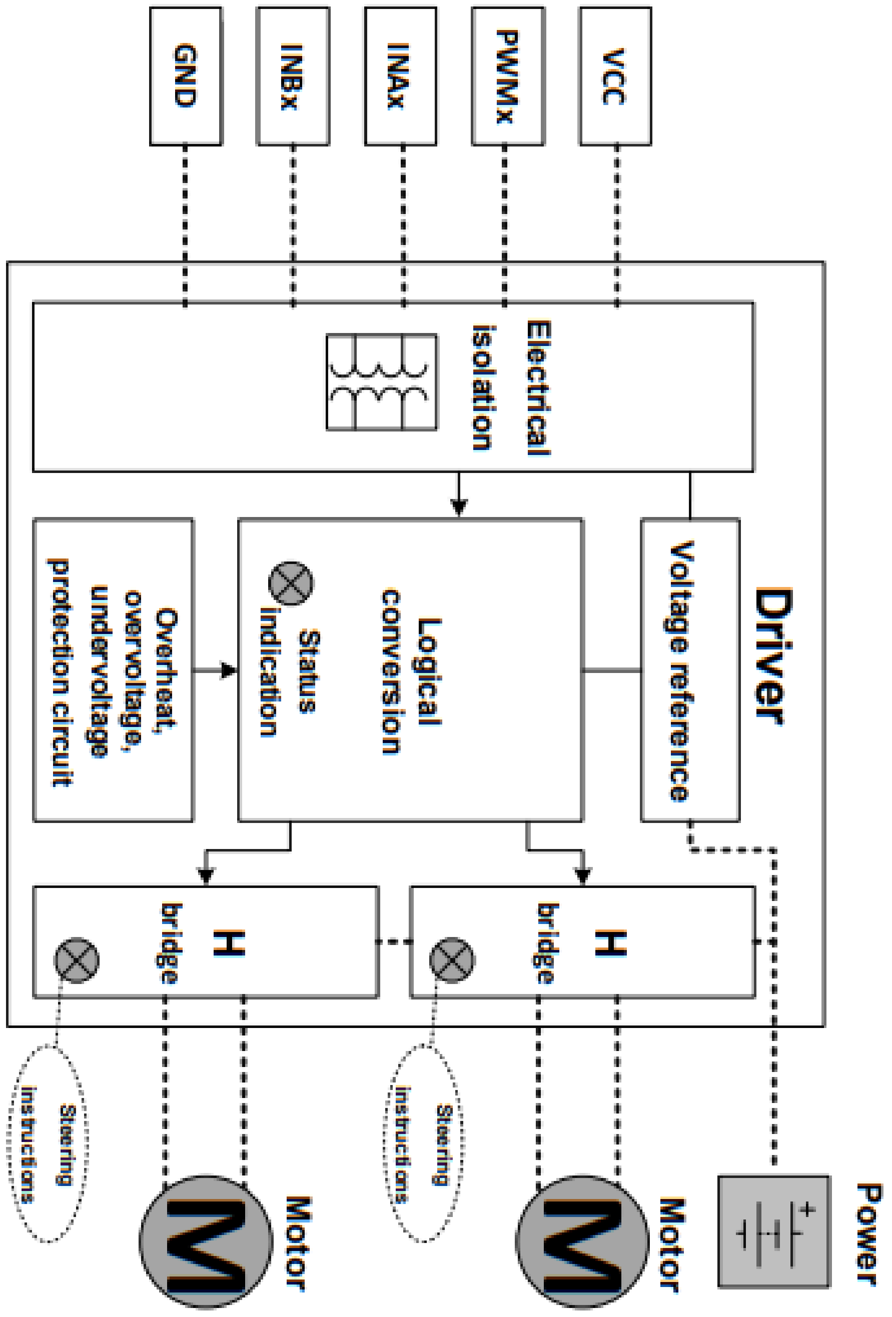


Image Processing Code

```
%connecting hardware
vidinput=videoinput('winvideo',1,'MJPG_640x480');
%Configure the triiger mode to be manual, greadtly reduce the triiger time
%and overhead
triggerconfig(vidinput, 'manual');
%framerate
vidsrc=getselectedsource(vidinput);
vidsrc.FrameRate='10.0000';
%Video player object to display frame
disporiginal=vision.VideoPlayer;
disppros=vision.VideoPlayer;

%start video
start(vidinput);

tic

%variables delcared here for image processing
threshold=30;
%hough trasnfrom parametes
rmin=20; rmax=200;

%try to do vectorization to increase speed
for i=1:500
   im=getsnapshot(vidinput);
   disporiginal.step(im);

   %split into rgb channels
```

```
    redchannel=squeeze(im(:,:,1));
    greenchannel=squeeze(im(:,:,2));
    bluechannel=squeeze(im(:,:,3));

    %create binary image, mathematic operation 3R-G-B
    impros=3*redchannel-2*(greenchannel+bluechannel);

    %Median filter , default is 3x3 matrix
    impros=medfilt2(impros,[2 2]);

    %thresholding
    impros=logical(impros-threshold);

    %remove pixel groups that have less than 200 pixels in a group
    impros=bwareaopen(impros,200);

    %Currently don't need to close the boundary, it also results
    %over-emerging of different obejcts
    impros=imfill(impros,'holes');

    %impros=bwconvhull(impros,'objects');
    impros=logical(impros-threshold);
    %define circle size to be 40:120
    %[houghtranscenter,houghtransradi]=imfindcircles(impros,[rmin
rmax],'ObjectPolarity','bright','Sensitivity',0.9);
    %houghtranscenter=round(houghtranscenter);

    disppros.step(impros);
end
```

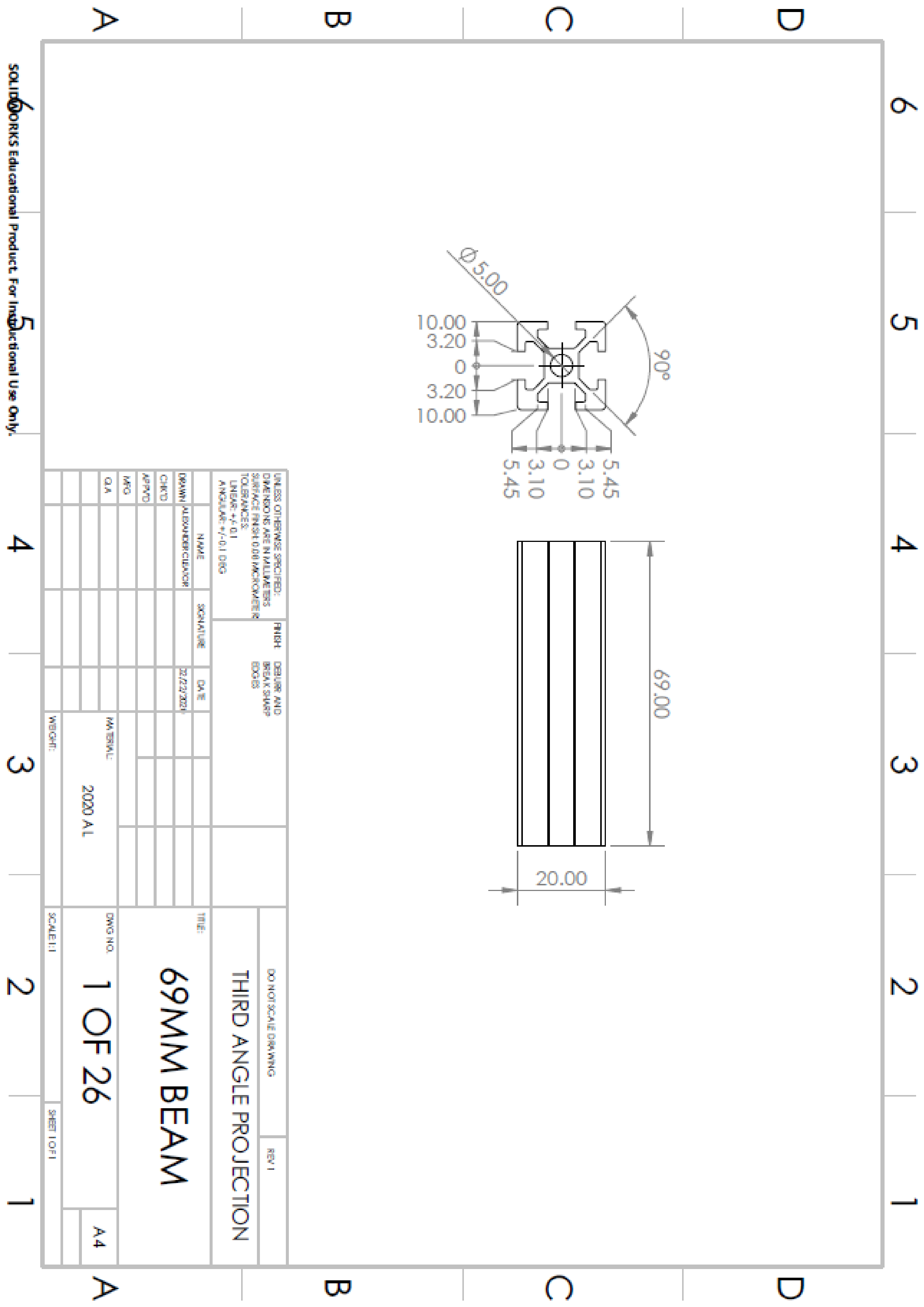
```
%%%end task
timedur=toc;
eachframe=timedur/500;



stop(vidinput);
delete(vidinput);
clear vidinput;
```

Emergency Stop Voice Recognition

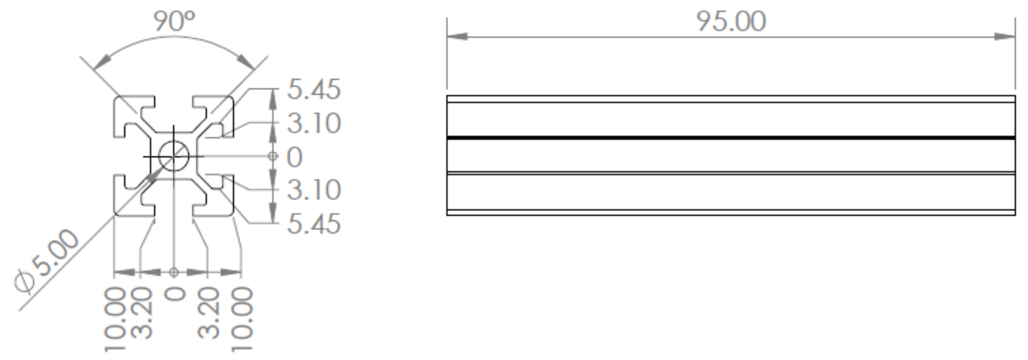
```
from deepspeech import Model
import numpy as np
import speech_recognition as sr

beam_width = 500
micfs=16000

models_folder = 'models/'
model_name = models_folder+"output_graph.pbmm"

ds = Model(model_name,beam_width)

mic = sr.Microphone(sample_rate=micfs)
r = sr.Recognizer()
with mic as source:r.adjust_for_ambient_noise(source)

while True:
    with mic as source:
        audio=r.listen(source,phrase_time_limit=1.3)
        audio = np.frombuffer(audio.frame_data, np.int16)
        print(ds.stt(audio))
```

Mechanical Drawings

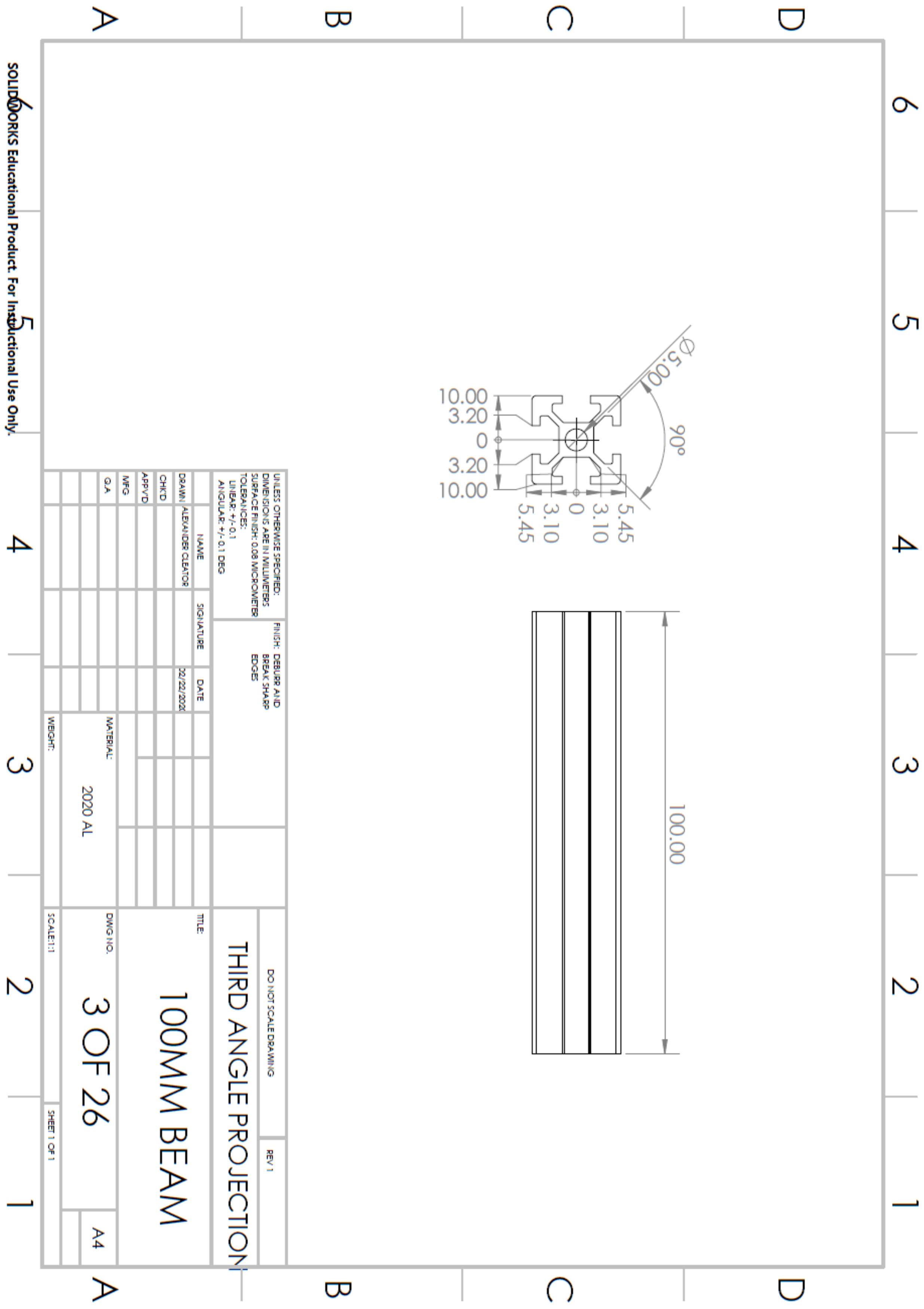
The following are the mechanical drawings necessary to fabricate the device. Components such as screws, nuts, washers etc were not drawn as they are standard components that would not need to be fabricated by a manufacturer.



A
B
C
D
6
5
4
3
2
1
SOLIDWORKS Educational Product. For Instructional Use Only.
Ø5.00
90°
10.00
3.20
0
3.20
10.00
5.45
3.10
0
3.10
5.45
69.00
20.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH 0.08 MICROMETERS
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/-0.1 DEG
FINISH:
DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEAVER
22/2/2021
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
2020 A L
WEIGHT:
DO NOT SCALE DRAWING
REV 1
TITLE:
THIRD ANGLE PROJECTION
69MM BEAM
DWG NO.
1 OF 26
A4
SCALE:1:1
SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: 0.08 MICROMETER TOLERANCES: LINEAR: +/- 0.1 ANGULAR: +/- 0.1 DEG		FINISH: DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REV 1
			THIRD ANGLE PROJECTION	
	NAME	SIGNATURE	DATE	TITLE: 95MM BEAM
DRAWN	ALEXANDER CLEATOR		02/22/2020	
CHK'D				
APPV'D				
MFG				
Q.A			MATERIAL: 2020 AL	DWG NO. 2 OF 26	A4
			WEIGHT:	SCALE:1:1	SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/-0.1
ANGULAR: +/- 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		2/22/202
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 2020 AL

WEIGHT:

TITLE: THIRD ANGLE PROJECTION

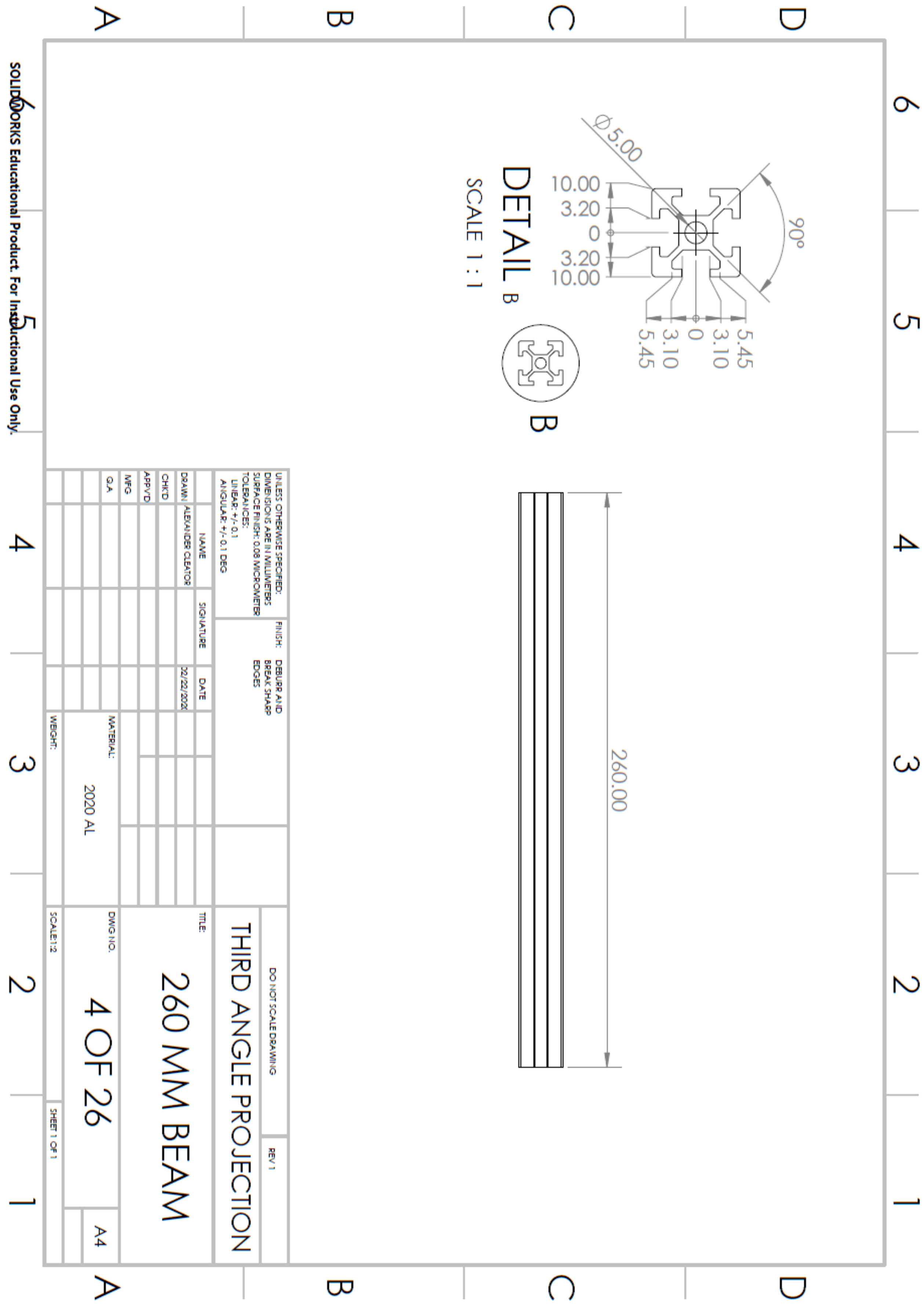
100MM BEAM

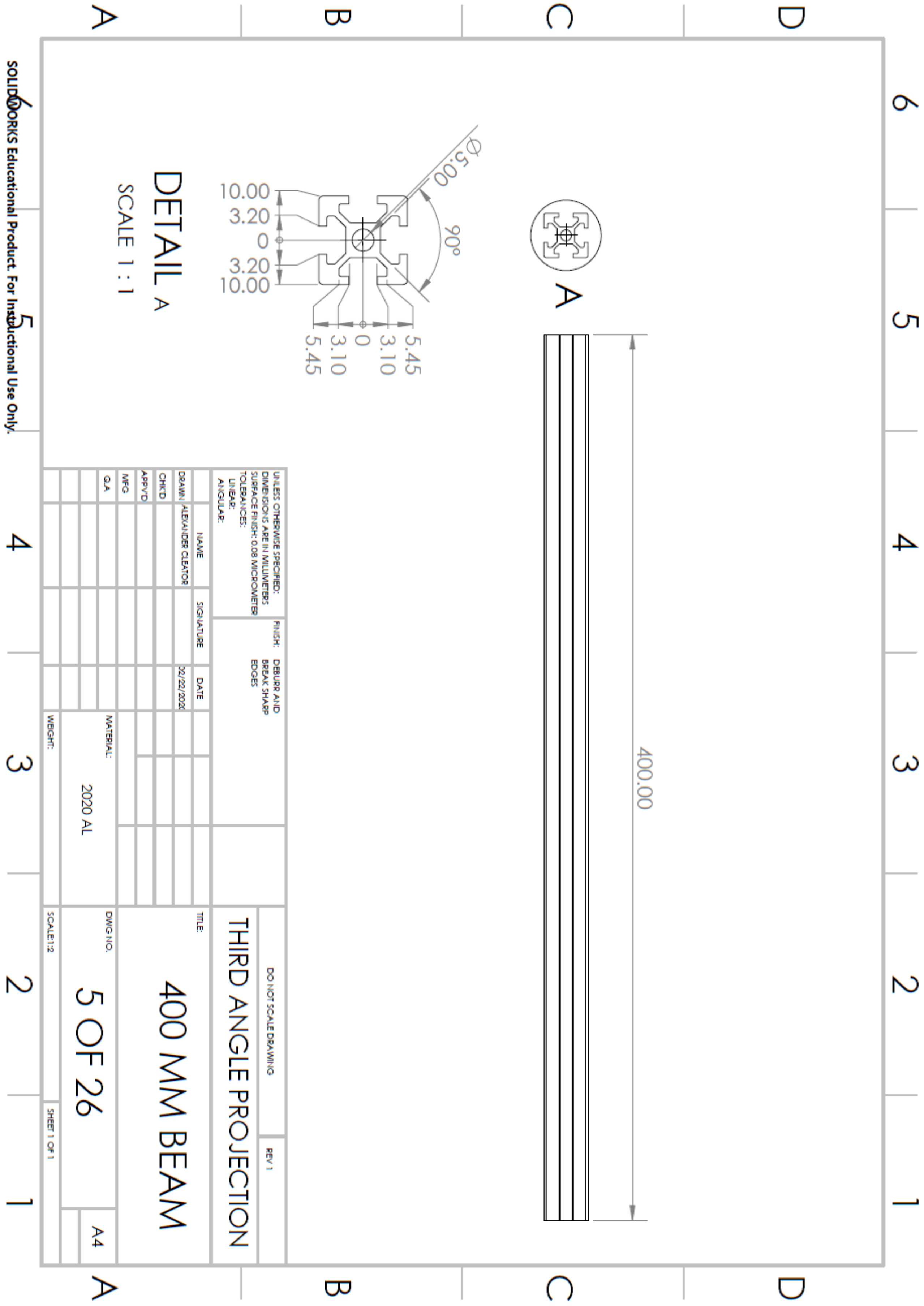
DWG NO. 3 OF 26

A4

SCALE:1:1

SHEET 1 OF 1


Ø5.00
90°
10.00
3.20
0
3.20
10.00
5.45
3.10
0
3.10
5.45
DETAIL B
SCALE 1 : 1
B
260.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 02/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
2020 AL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
TITLE:
THIRD ANGLE PROJECTION
260 MM BEAM
DWG NO.
4 OF 26
A4
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR:
ANGULAR:

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

THIRD ANGLE PROJECTION

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/202
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 2020 AL

WEIGHT:

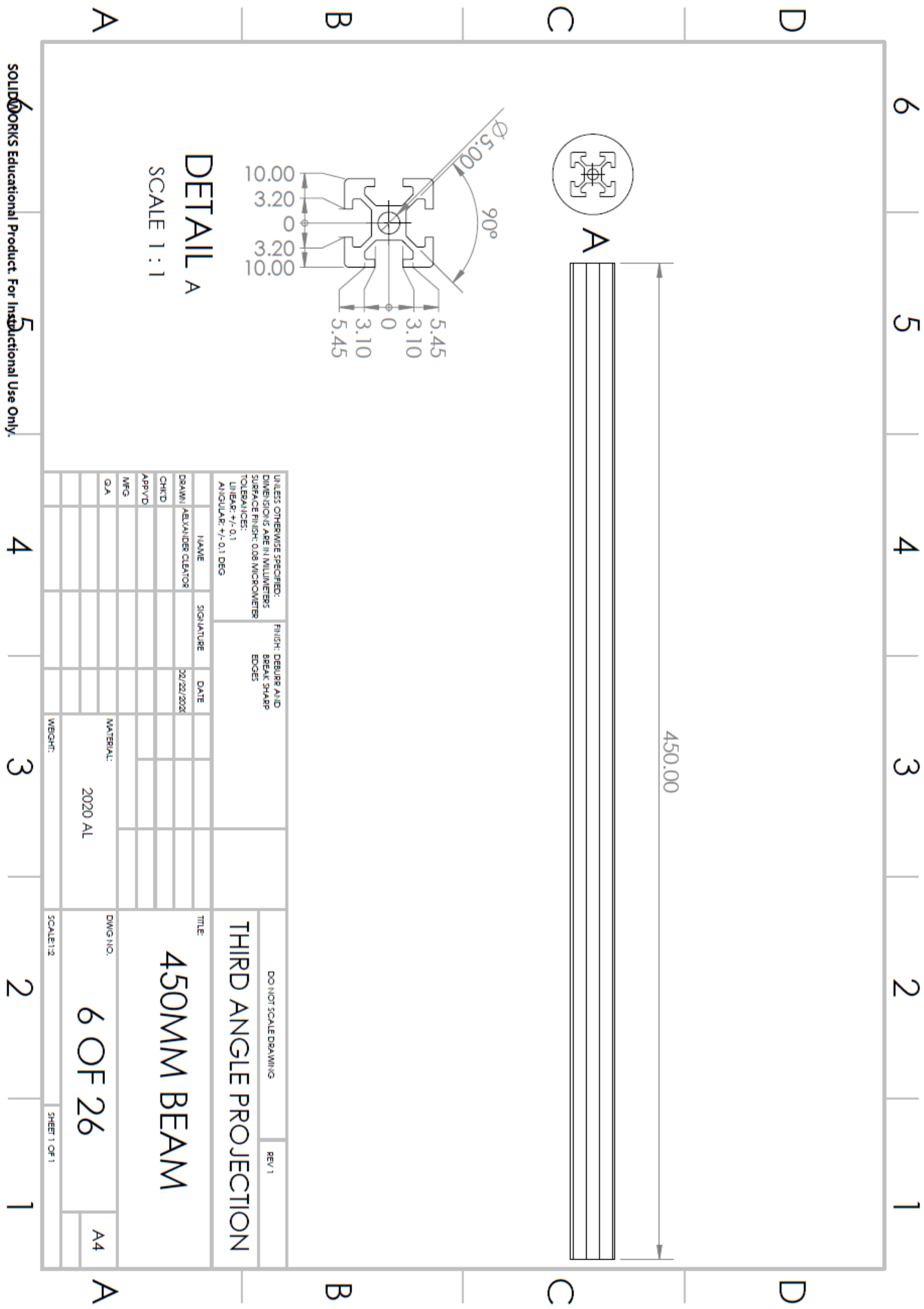
TITLE: 400 MM BEAM

DWG NO. 5 OF 26

A4

SCALE:1:2

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

	NAME	SIGNATURE	DATE
DRAWN	AELXANDER CLEATOR		22/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 2020 AL

WEIGHT:

THIRD ANGLE PROJECTION

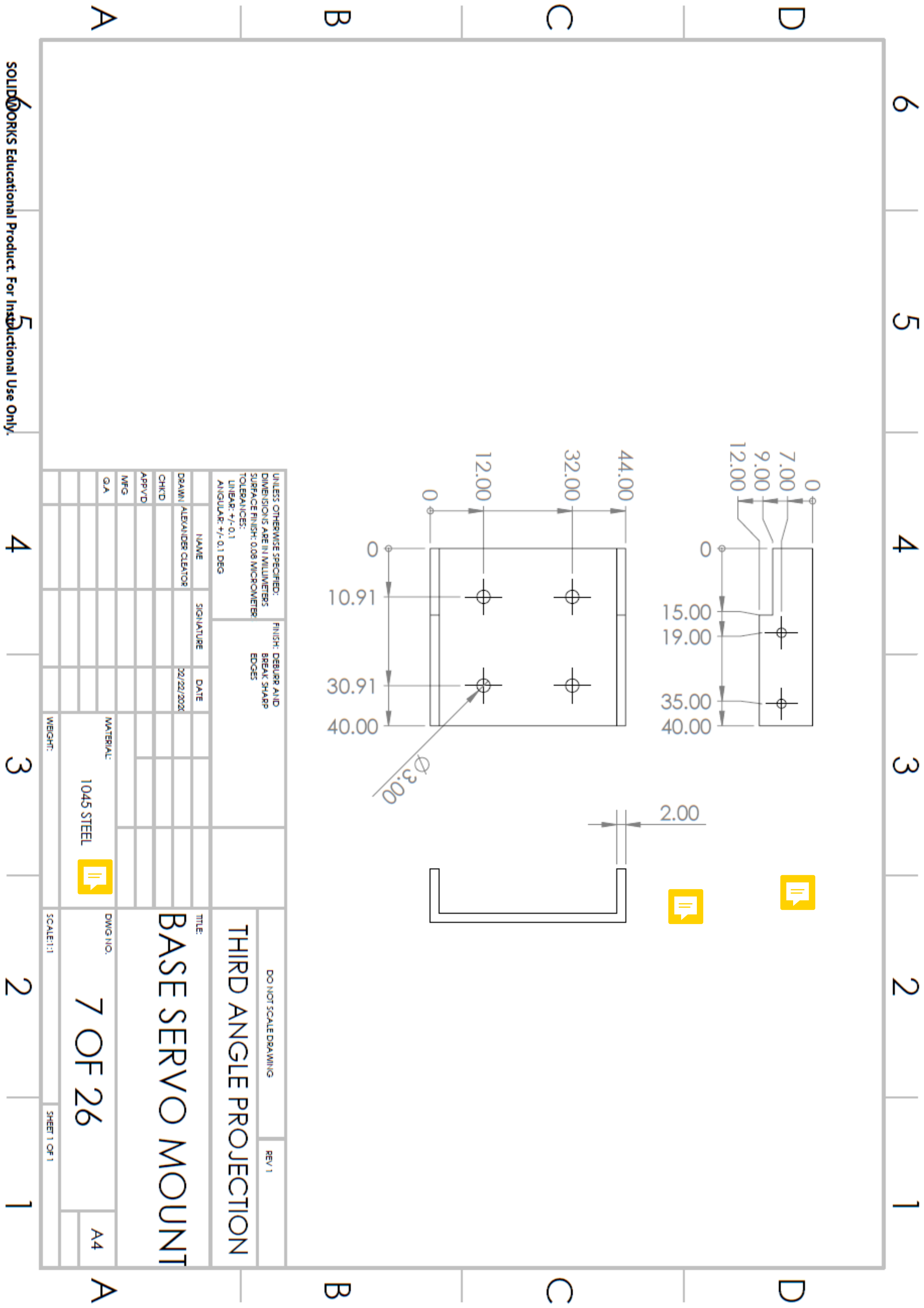
TITLE: 450MM BEAM

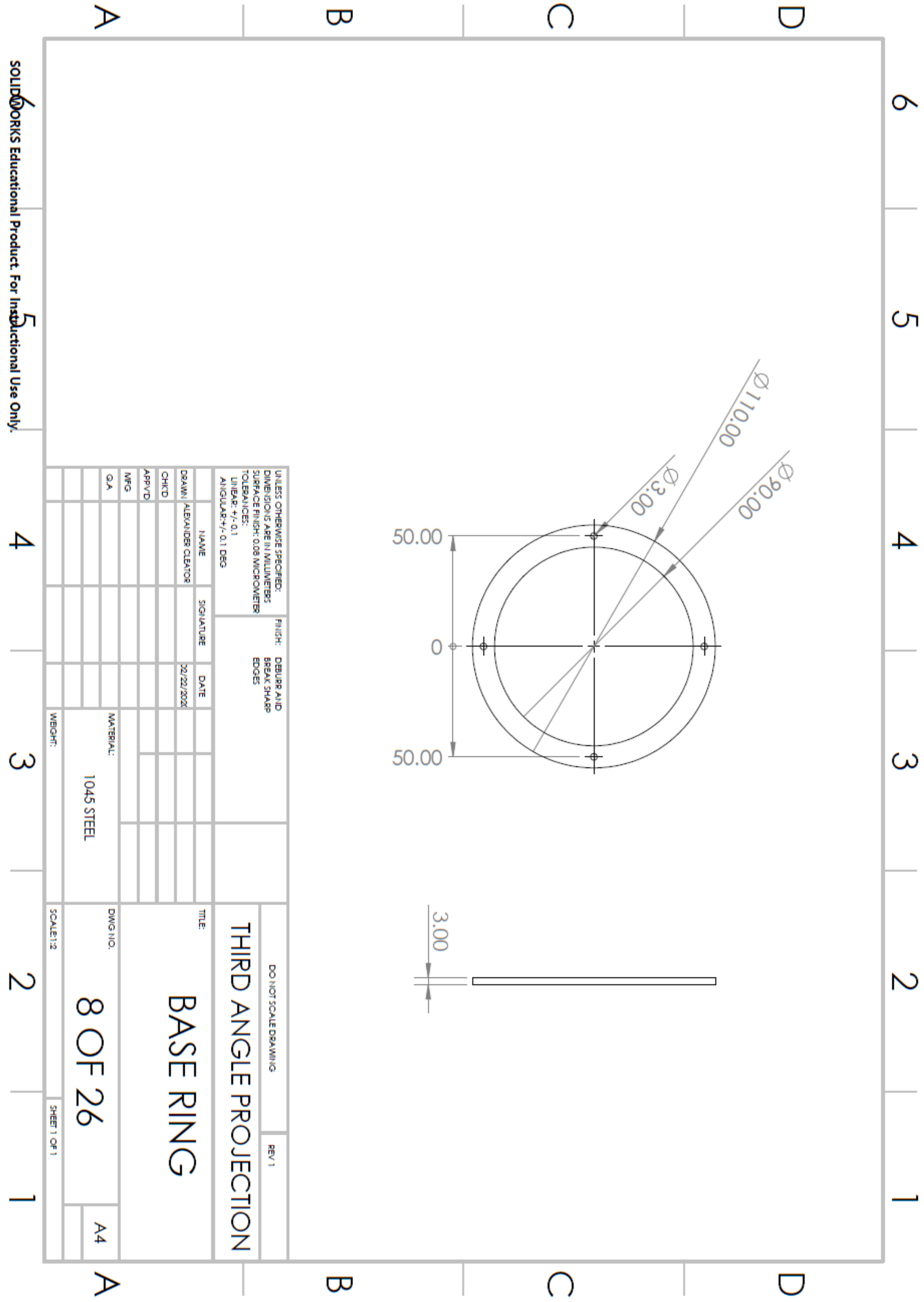
DWG NO. 6 OF 26

A4

SCALE:1:2

SHEET 1 OF 1


SOLIDWORKS Educational Product. For Instructional Use Only.
0
7.00
9.00
12.00
15.00
19.00
35.00
40.00
12.00
32.00
44.00
10.91
30.91
40.00
⌀3.00
2.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/-0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REV 1
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 20/02/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 STEEL
WEIGHT:
TITLE:
THIRD ANGLE PROJECTION
BASE SERVO MOUNT
DWG NO.
7 OF 26
A4
SCALE:1:1
SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/-0.1
ANGULAR:+/-0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 1045 STEEL

WEIGHT:

THIRD ANGLE PROJECTION

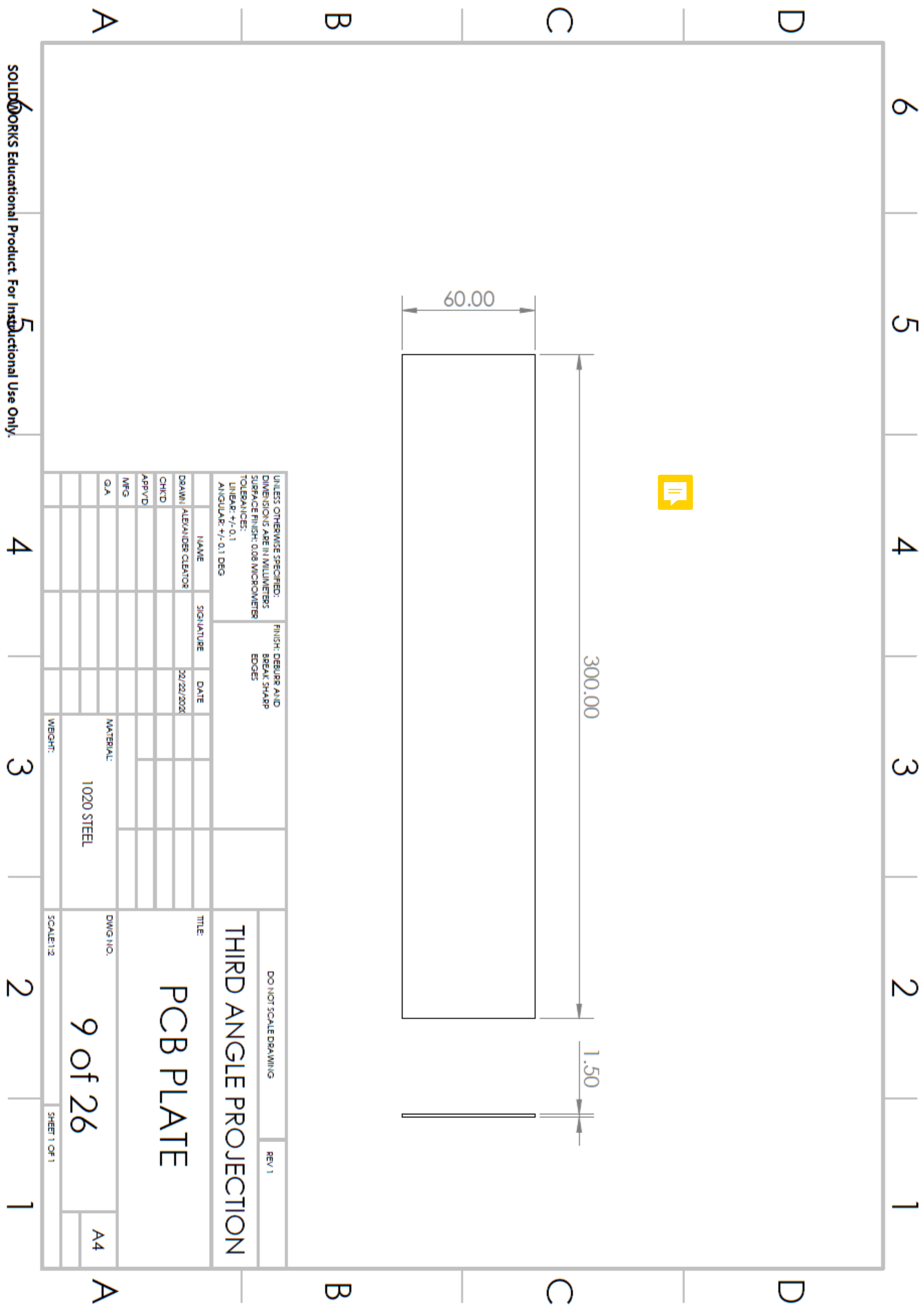
TITLE: BASE RING

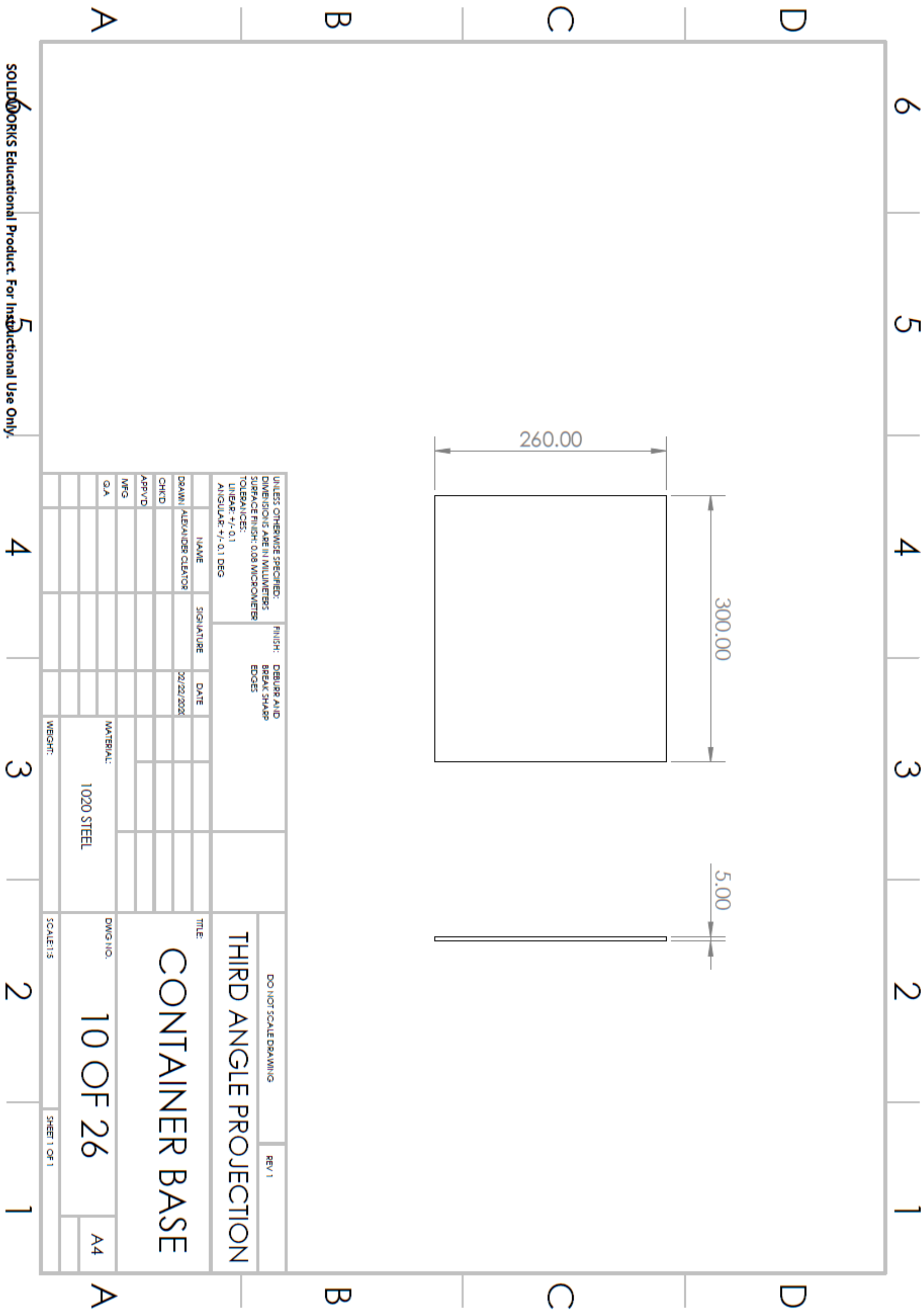
DWG NO. 8 OF 26

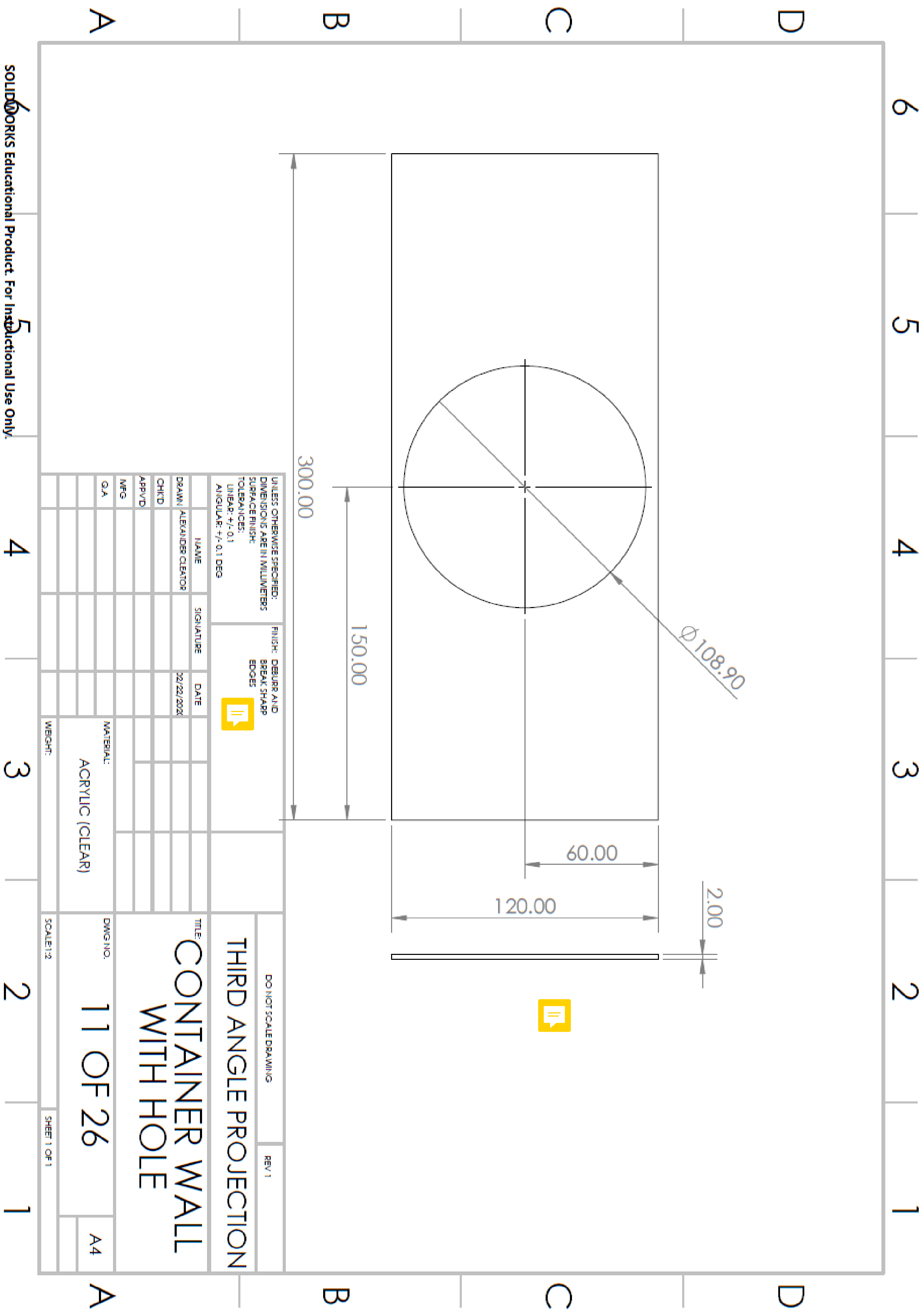
A4

SCALE:1:2

SHEET 1 OF 1


60.00
300.00
1.50
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/-0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
22/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1020 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
PCB PLATE
DWG NO.
9 of 26
A4
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


260.00
300.00
5.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
22/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1020 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
CONTAINER BASE
DWG NO.
10 OF 26
A4
SCALE:1:5
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



SOLIDWORKS Educational Product. For Instructional Use Only.

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/202
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: ACRYLIC (CLEAR)

WEIGHT:

THIRD ANGLE PROJECTION

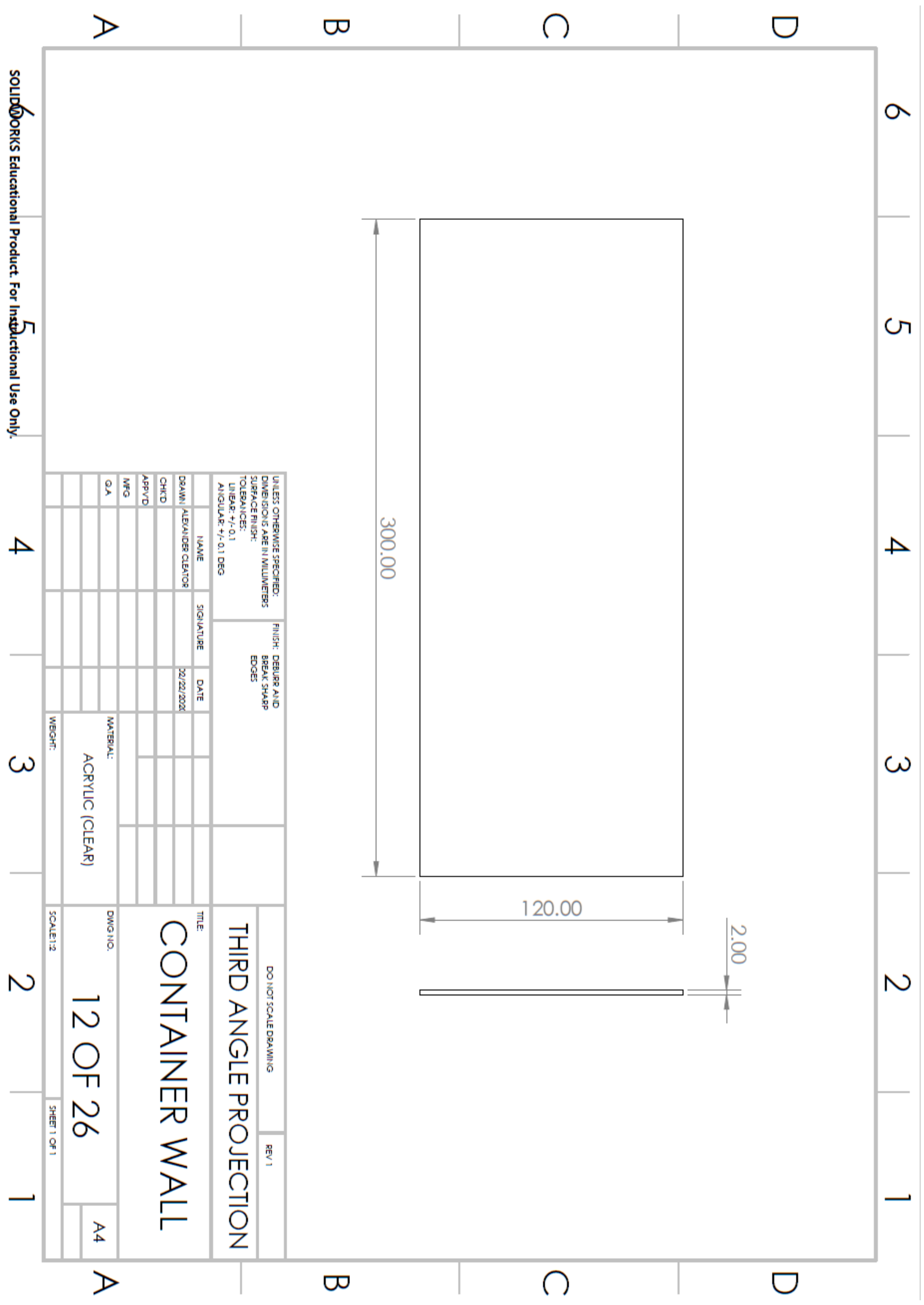
TITLE: CONTAINER WALL WITH HOLE

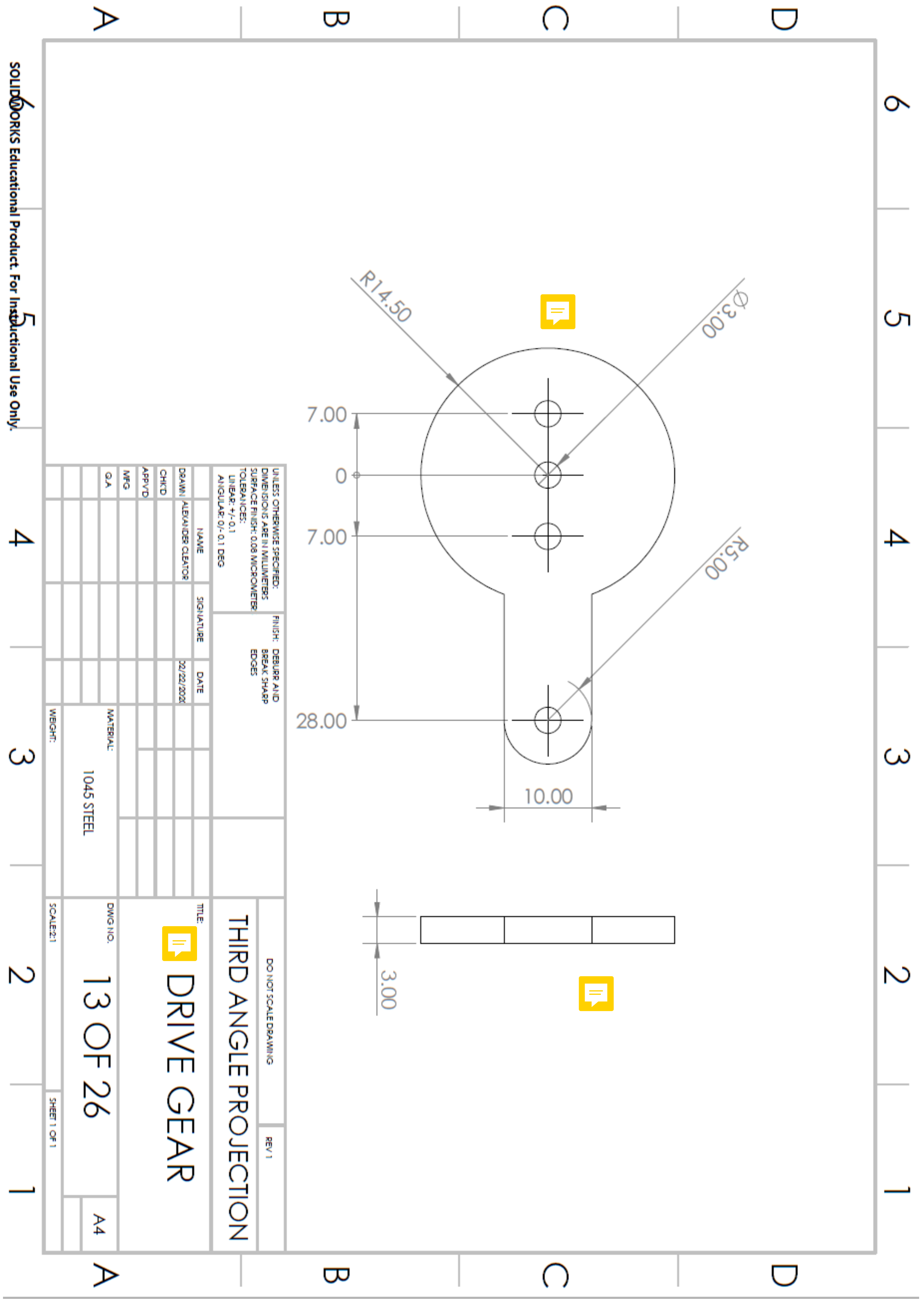
DWG NO. 11 OF 26

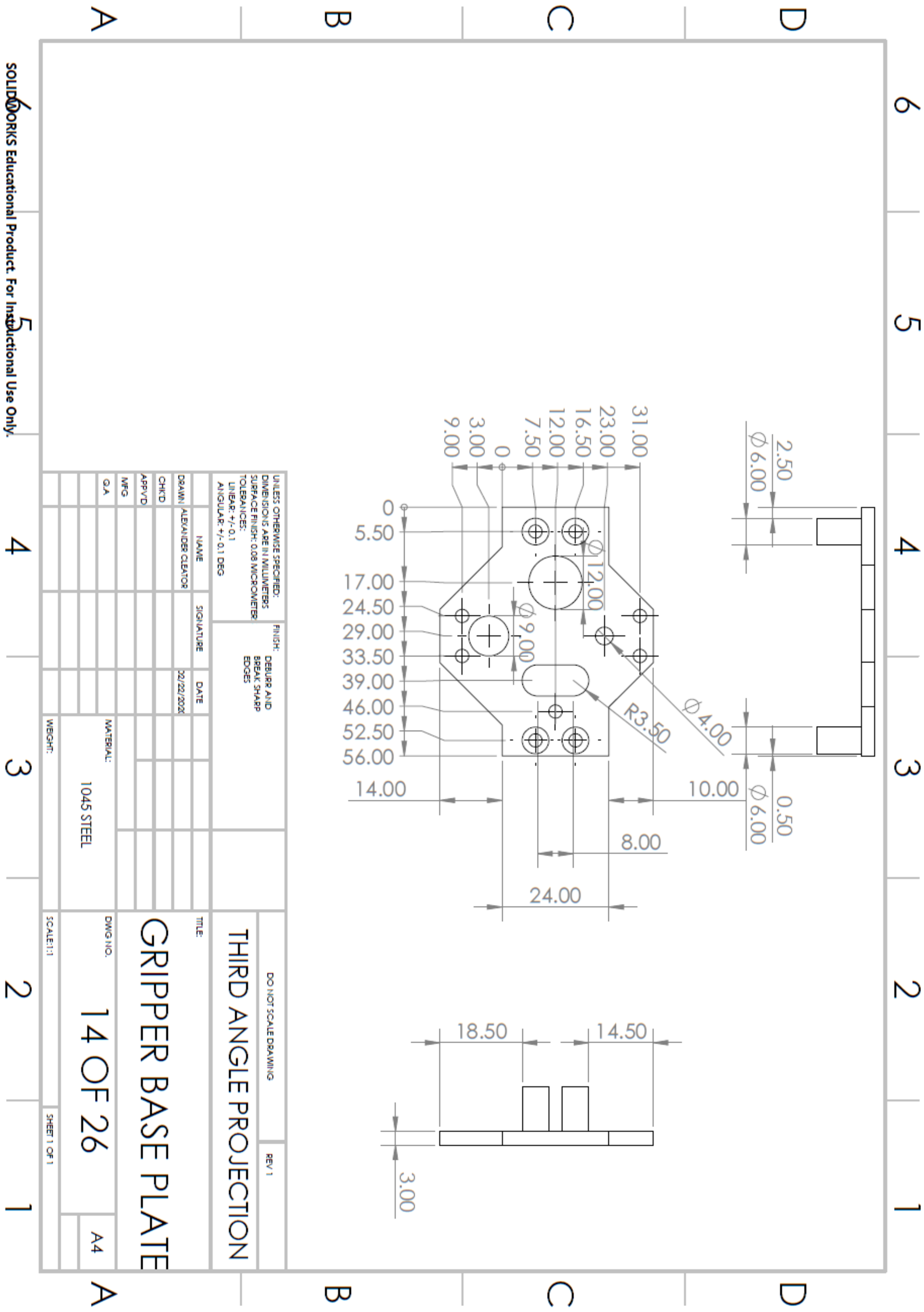
A4

SCALE:1:2

SHEET 1 OF 1


300.00
120.00
2.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR: +/-0.1
ANGULAR: +/-0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 22/22/202
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
ACRYLIC (CLEAR)
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
CONTAINER WALL
DWG NO.
12 OF 26
A4
SCALE:1:2
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


R14.50
Ø3.00
R5.00
7.00
0
7.00
28.00
10.00
3.00
A
B
C
D
1
2
3
4
5
6
SOLIDWORKS Educational Product. For Instructional Use Only.
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/-0.1
ANGULAR: 0/-0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
2/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
DRIVE GEAR
DWG NO.
13 OF 26
A4
SCALE:2:1
SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 1045 STEEL

WEIGHT:

DO NOT SCALE DRAWING

REV 1

THIRD ANGLE PROJECTION

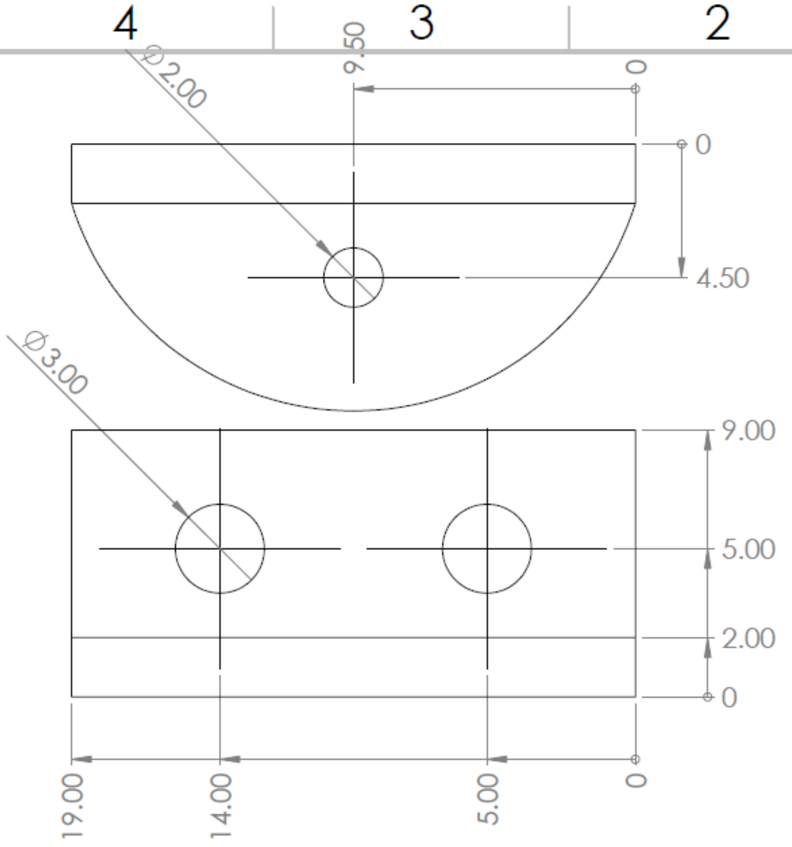
TITLE: GRIPPER BASE PLATE

DWG NO. 14 OF 26

A4

SCALE:1:1

SHEET 1 OF 1



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

THIRD ANGLE PROJECTION

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: GRIPPER MOUNT

MATERIAL: 1045 STEEL

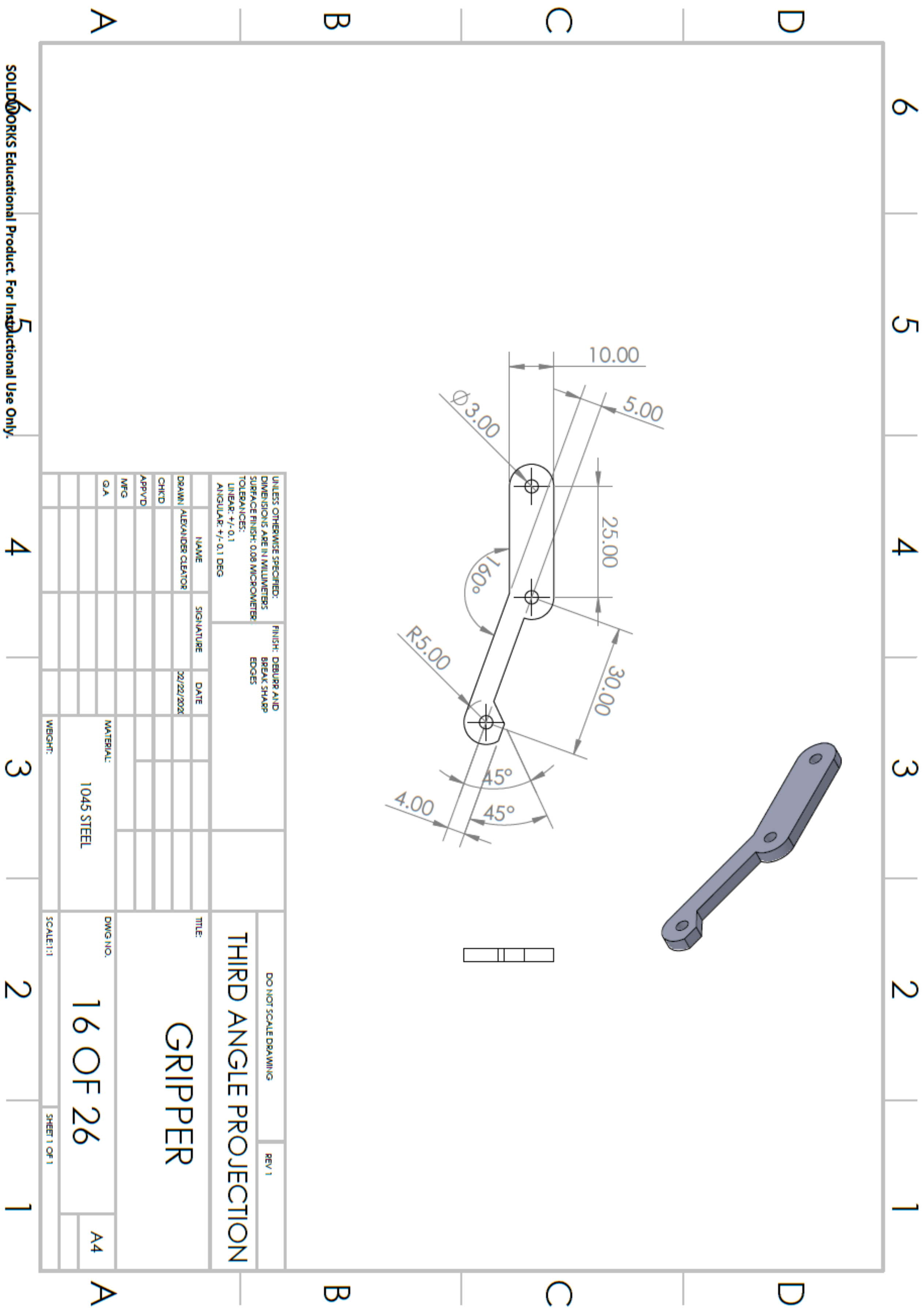
DWG NO. 15 OF 26

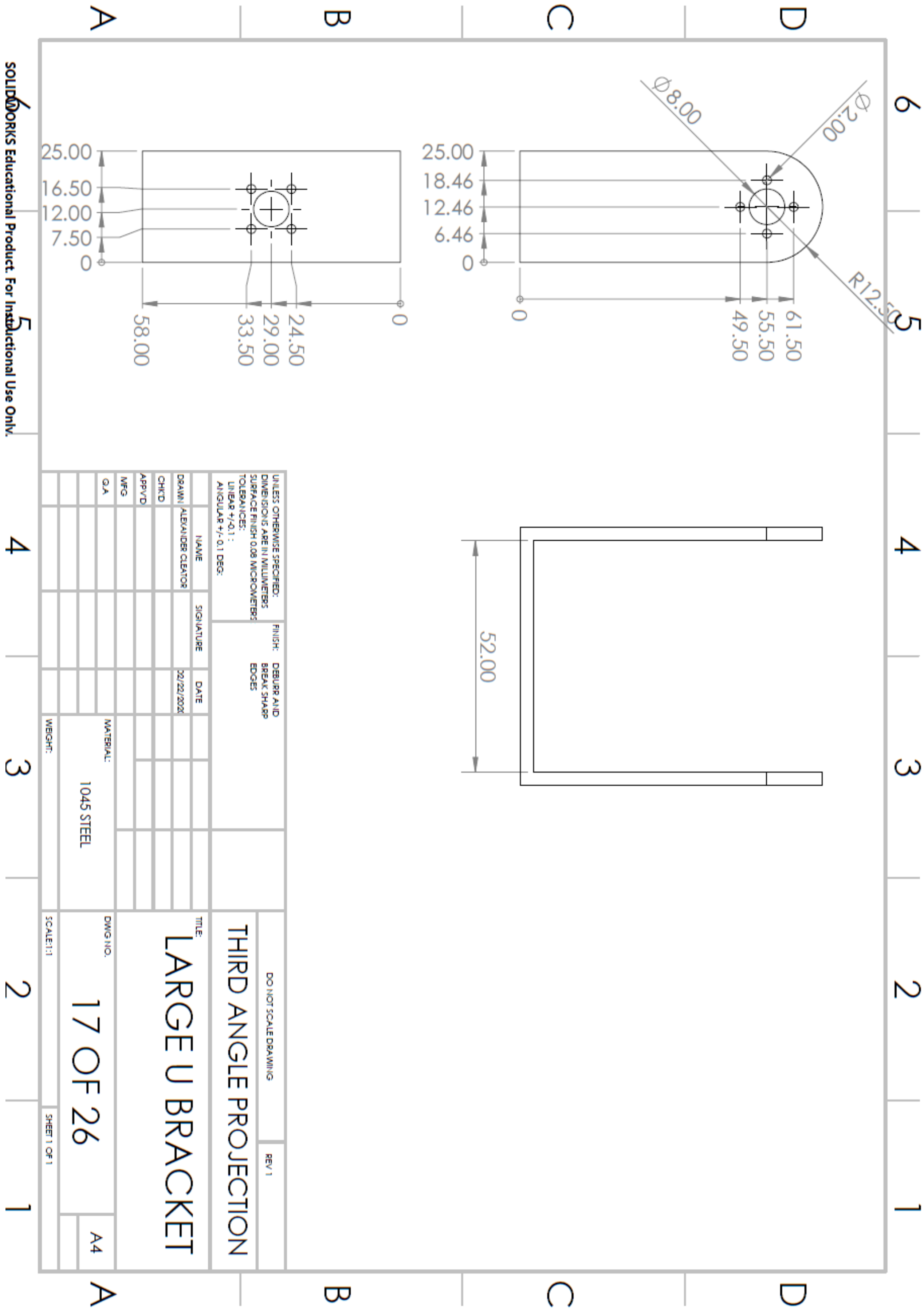
A4

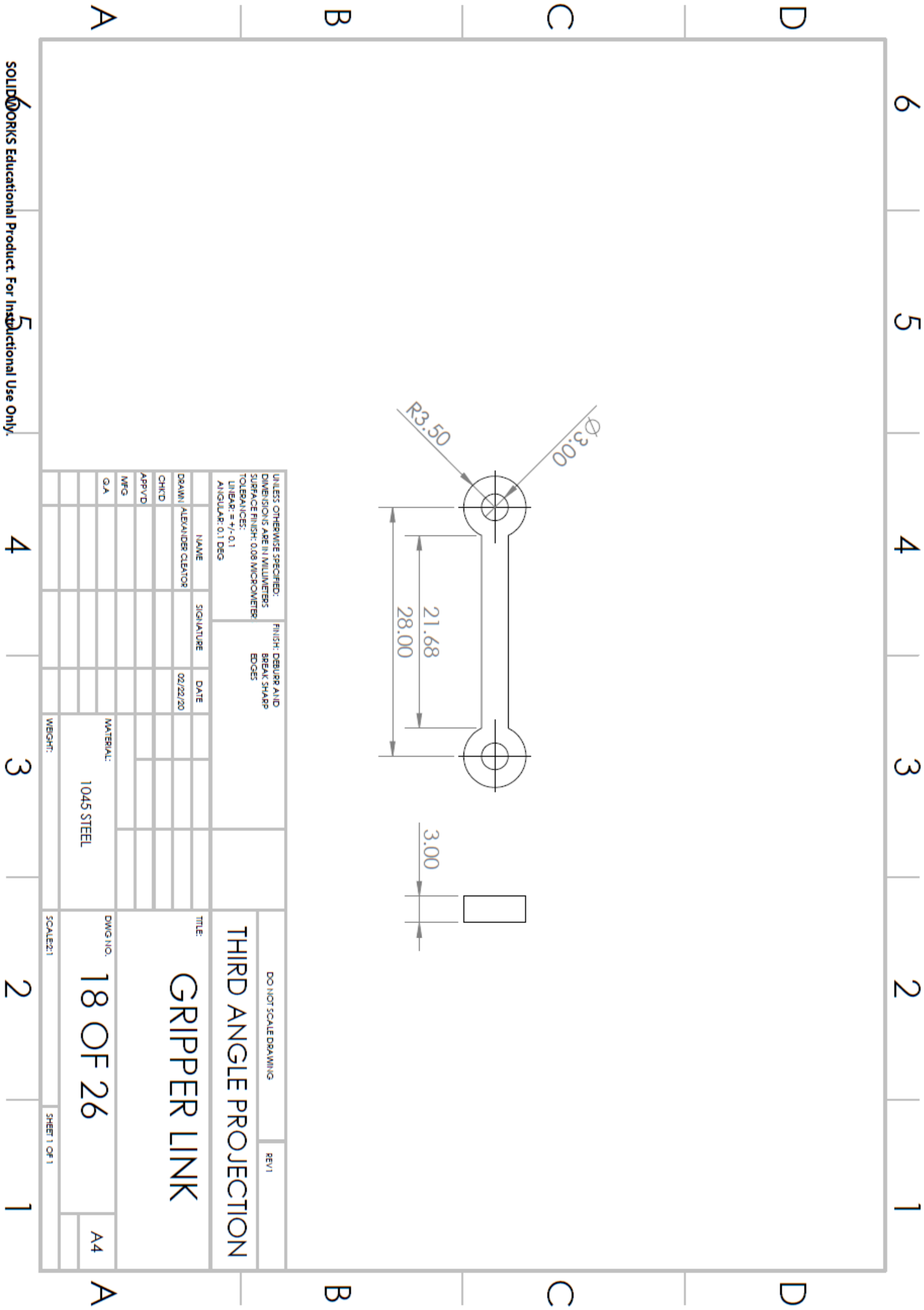
WEIGHT:

SCALE:5:1

SHEET 1 OF 1


10.00
5.00
Ø3.00
25.00
160°
R5.00
30.00
45°
45°
4.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REV 1
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
22/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 STEEL
WEIGHT:
TITLE:
THIRD ANGLE PROJECTION
GRIPPER
DWG NO.
16 OF 26
A4
SCALE:1:1
SHEET 1 OF 1


SOLIDWORKS Educational Product. For Instructional Use Only.
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH 0.08 MICROMETERS
TOLERANCES:
LINEAR +/-0.1 :
ANGULAR +/-0.1 DEG:
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 20/02/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL: 1045 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE: LARGE U BRACKET
DWG NO. 17 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
Ø8.00
Ø2.00
R12.50
25.00
18.46
12.46
6.46
0
61.50
55.50
49.50
16.50
12.00
7.50
24.50
29.00
33.50
58.00
52.00



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: = +/- 0.1
ANGULAR: 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/20
CHK'D			
APPV'D			
MFG			
Q.A			

MATERIAL: 1045 STEEL

WEIGHT:

DO NOT SCALE DRAWING

REV1

THIRD ANGLE PROJECTION

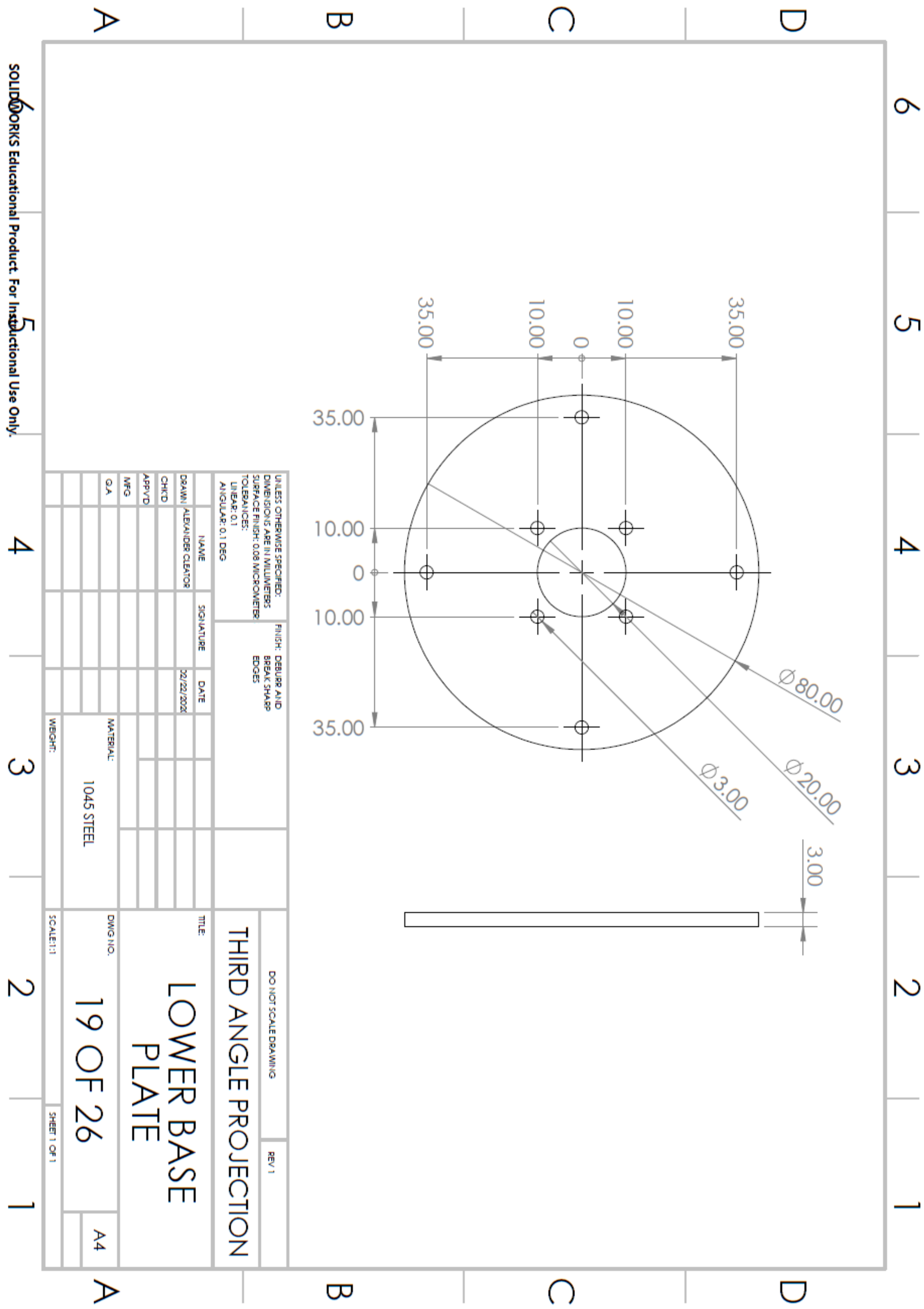
TITLE: GRIPPER LINK

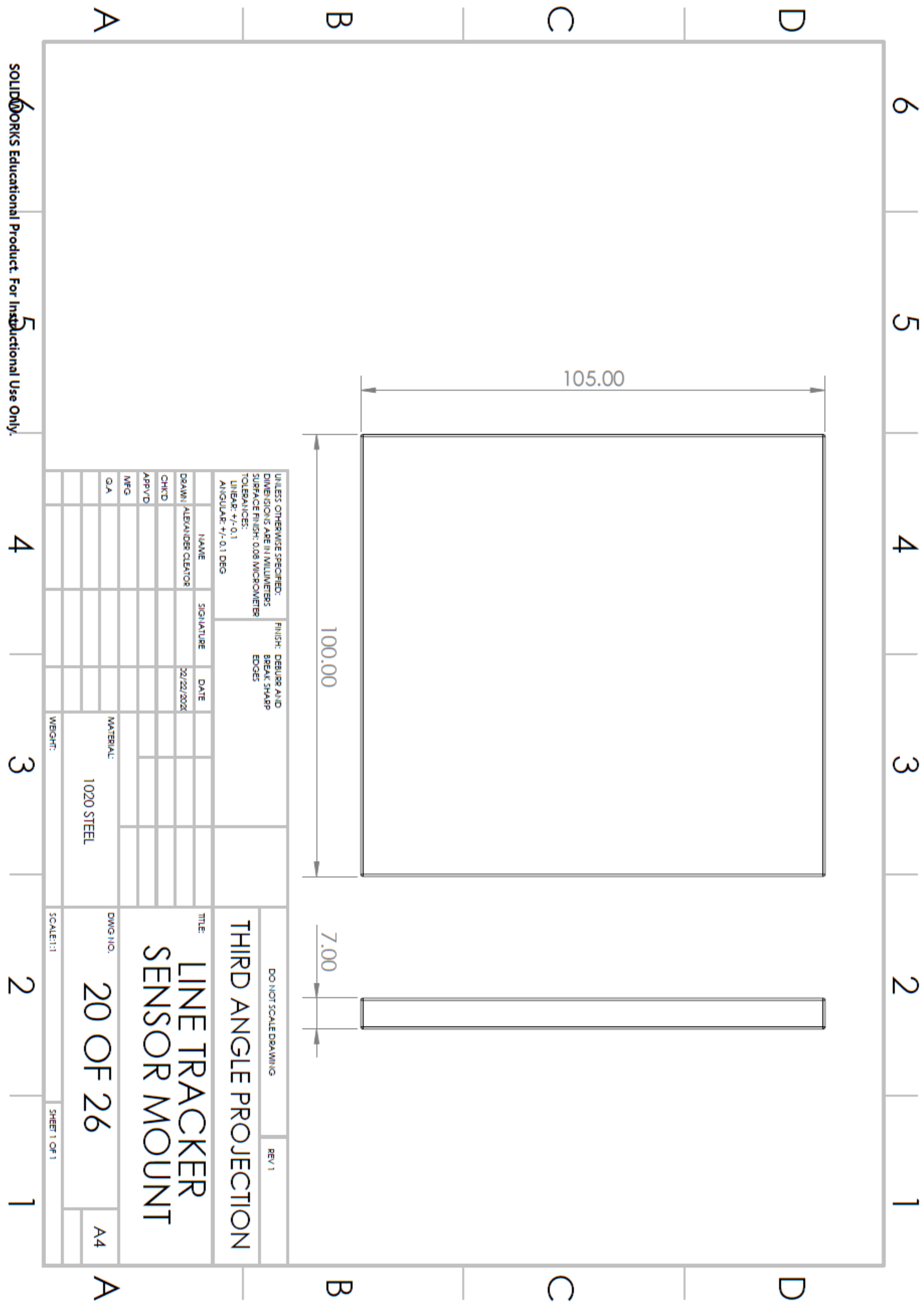
DWG NO. 18 OF 26

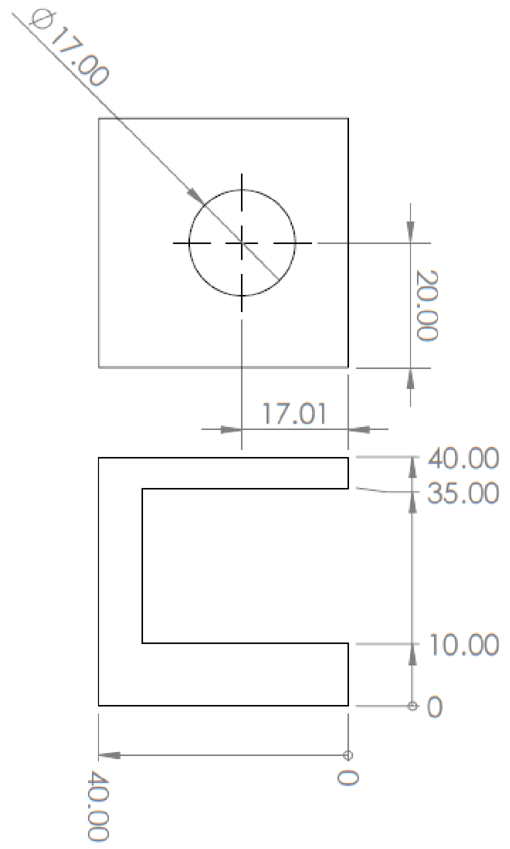
A4

SCALE:2:1

SHEET 1 OF 1


35.00
10.00
0
10.00
35.00
35.00
10.00
0
10.00
35.00
Ø80.00
Ø3.00
Ø20.00
3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: 0.1
ANGULAR: 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
22/22/202
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
LOWER BASE PLATE
DWG NO.
19 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


105.00
100.00
7.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REV 1
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 02/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1020 STEEL
WEIGHT:
THIRD ANGLE PROJECTION
TITLE:
LINE TRACKER SENSOR MOUNT
DWG NO.
20 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: 0.08 MICROMETER TOLERANCES: LINEAR: +/- 0.1 ANGULAR: +/- 0.1 DEG		FINISH: DEBURR AND BREAK SHARP EDGES	
	NAME	SIGNATURE	DATE
DRAWN	ALEXANDER CLEATOR		02/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			
MATERIAL:	1020 STEEL		
WEIGHT:			

THIRD ANGLE PROJECTION

DO NOT SCALE DRAWING

REV 1

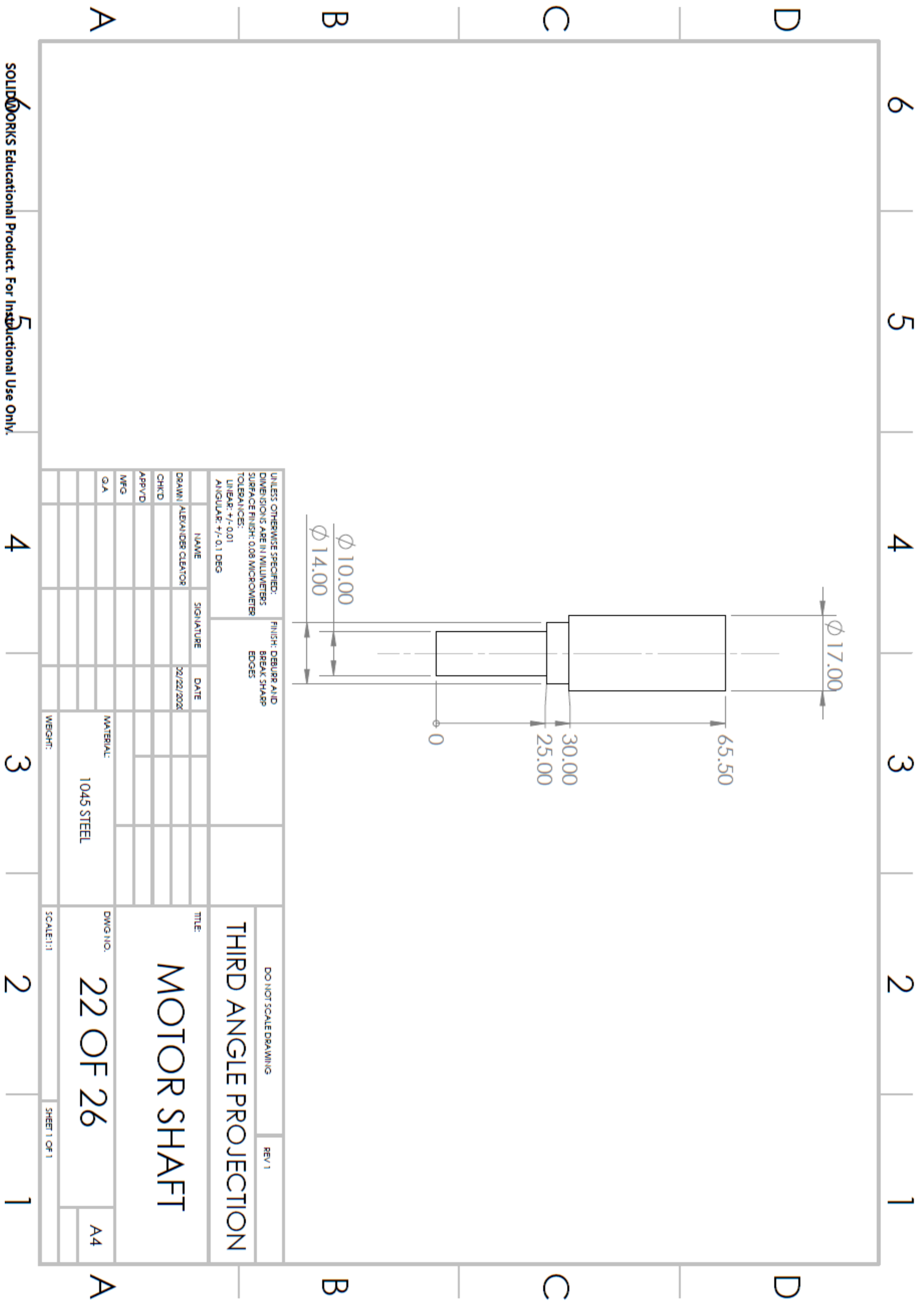
TITLE: MOTOR MOUNT

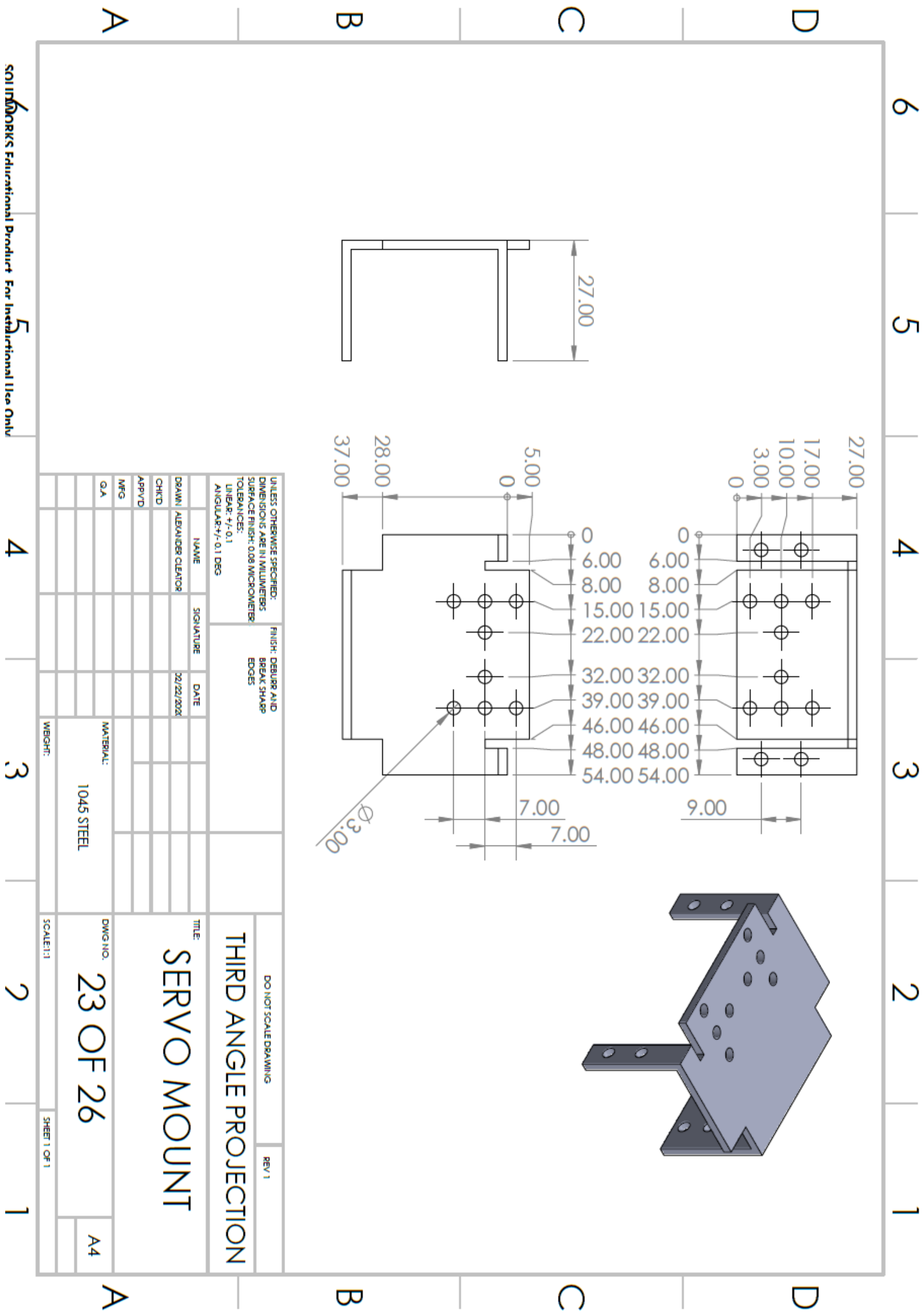
DWG NO. 21 OF 26

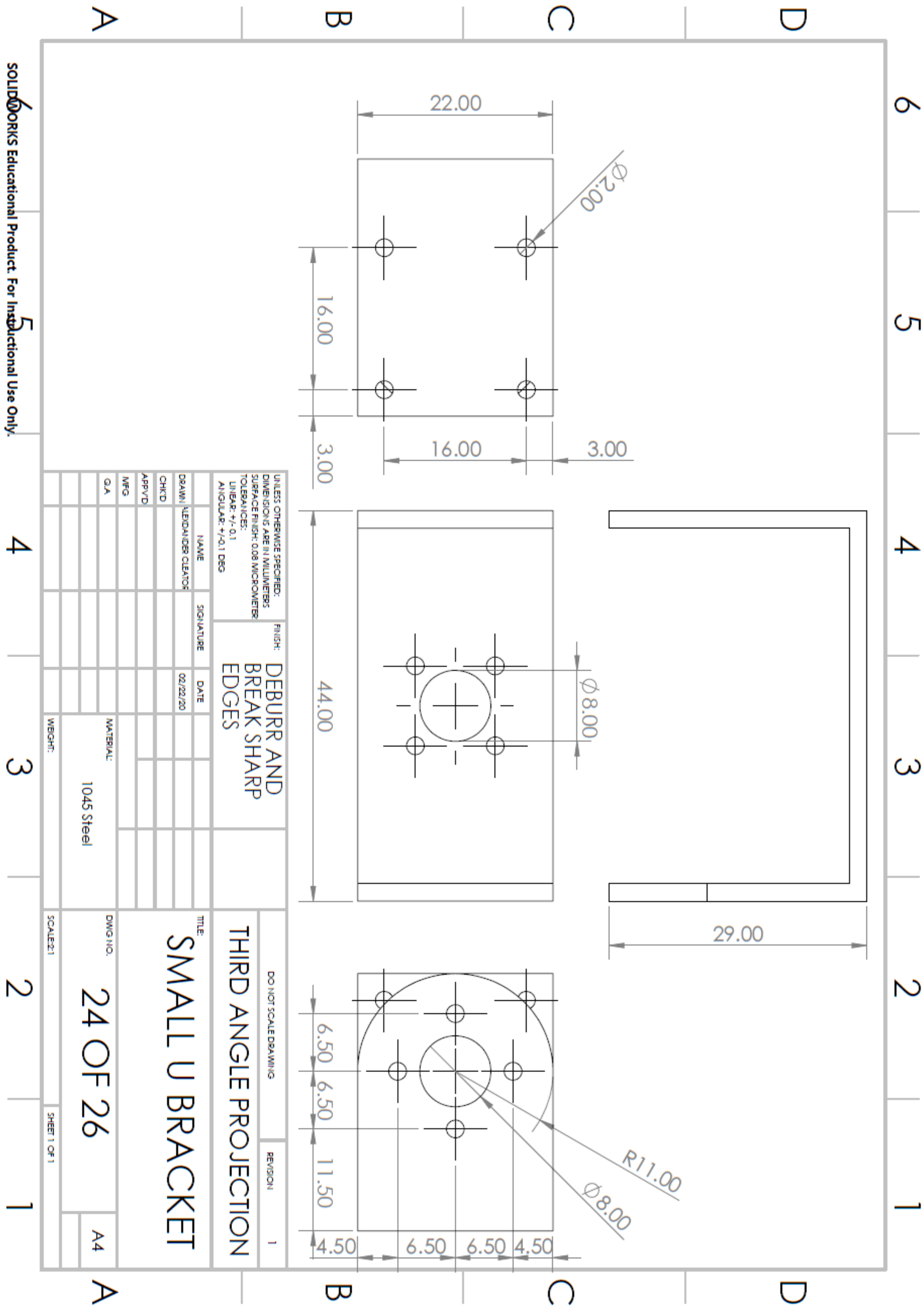
A4

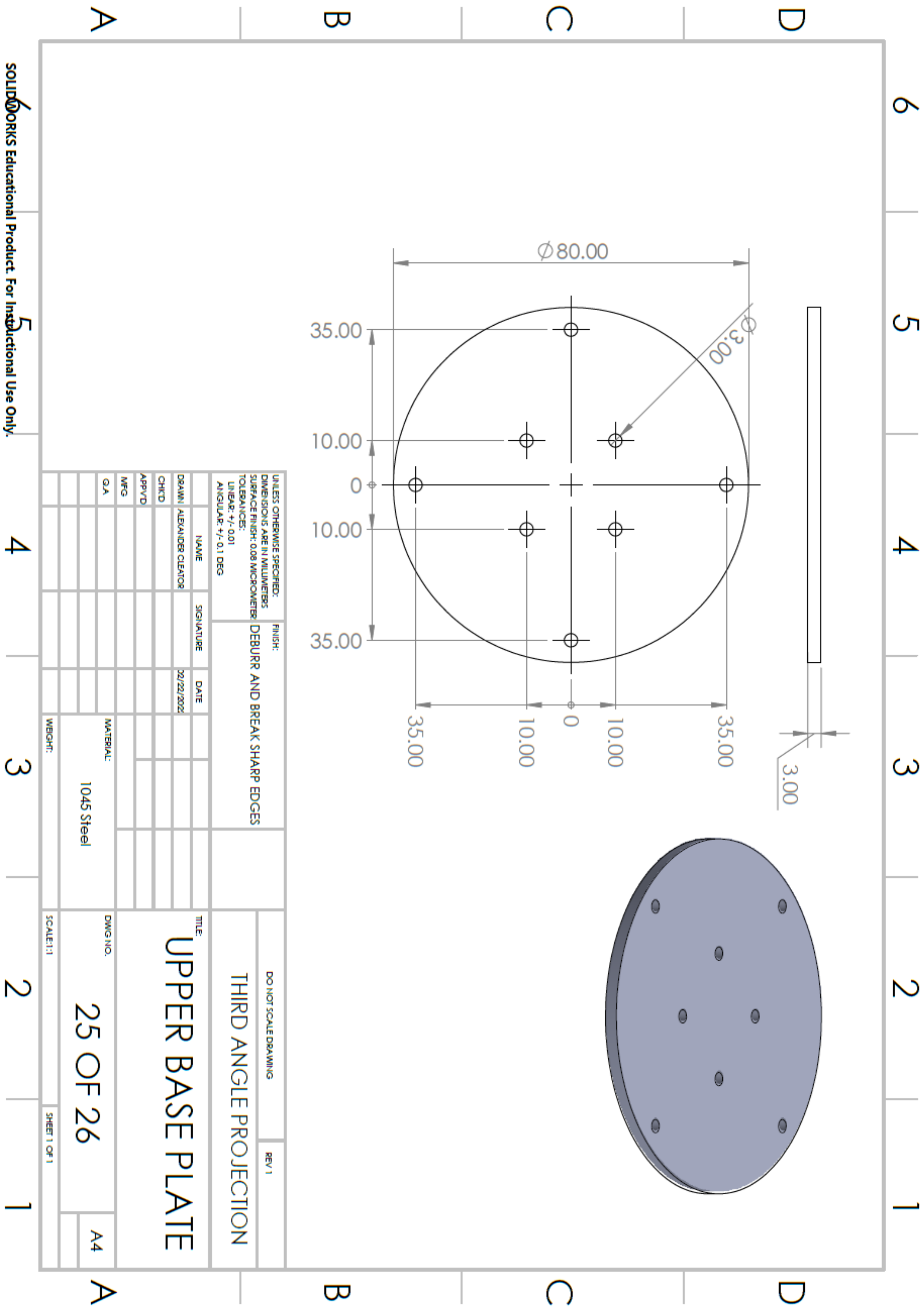
SCALE:1:1

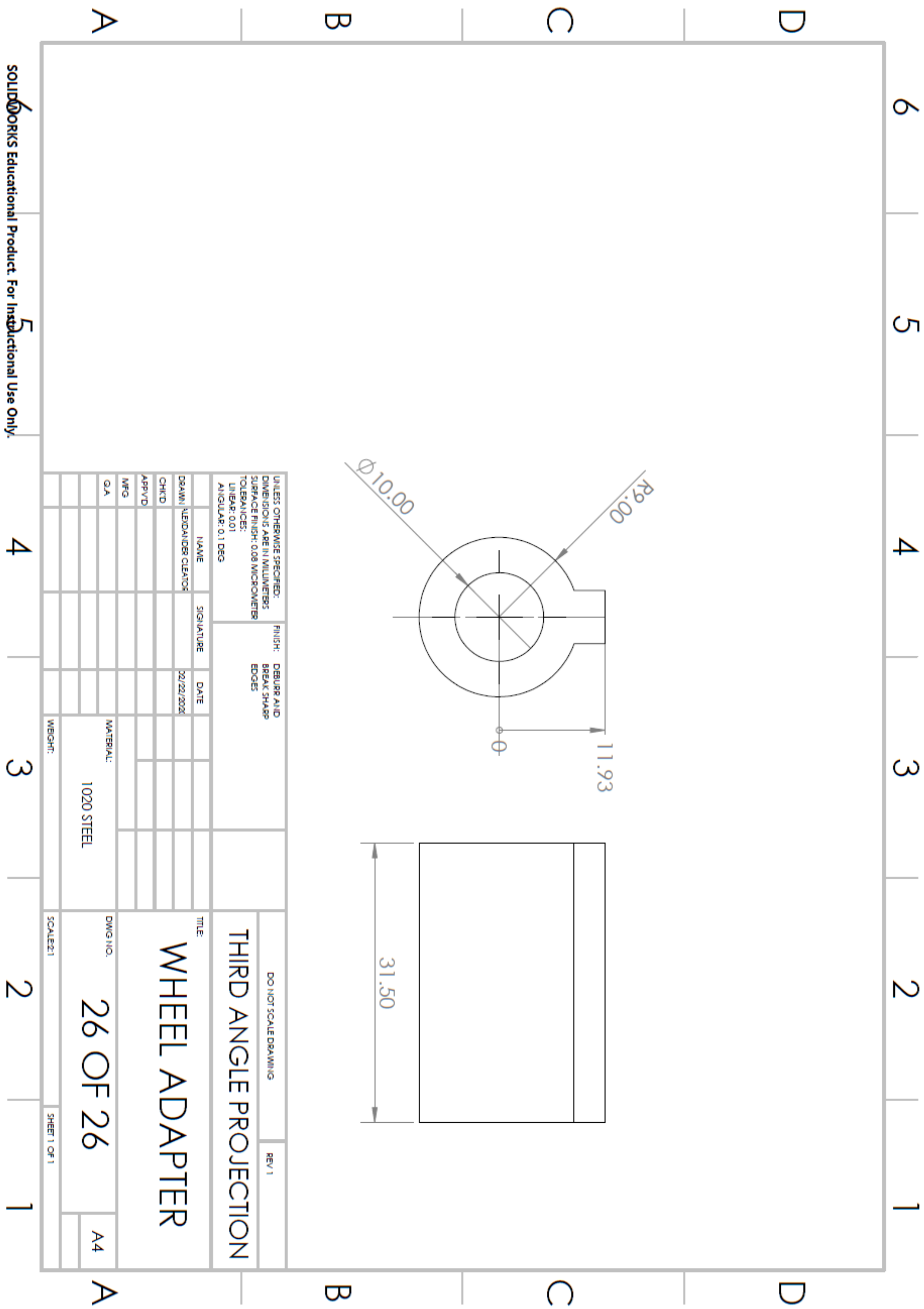
SHEET 1 OF 1


Ø17.00
Ø10.00
Ø14.00
0
25.00
30.00
65.50
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: 0.08 MICROMETER TOLERANCES: LINEAR: +/- 0.01 ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REV 1
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 02/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL: 1045 STEEL
WEIGHT:
TITLE: THIRD ANGLE PROJECTION
MOTOR SHAFT
DWG NO. 22 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


27.00
37.00
28.00
5.00
0
27.00
17.00
10.00
3.00
0
0 6.00 8.00 15.00 22.00 32.00 39.00 46.00 48.00 54.00
0 6.00 8.00 15.00 22.00 32.00 39.00 46.00 48.00 54.00
7.00
7.00
9.00
Ø3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETERS
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/- 0.1 DEG
FINISH: DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN ALEXANDER CLEATOR 02/22/2020
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 STEEL
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
SERVO MOUNT
DWG NO.
23 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only


22.00
Ø2.00
16.00
3.00
16.00
3.00
44.00
Ø8.00
29.00
6.50
6.50
11.50
4.50
6.50
6.50
4.50
R11.00
Ø8.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.1
ANGULAR: +/-0.1 DEG
FINISH:
DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXDANDER CLEATOR
02/22/20
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 Steel
WEIGHT:
DO NOT SCALE DRAWING
REVISION
1
THIRD ANGLE PROJECTION
TITLE:
SMALL U BRACKET
DWG NO.
24 OF 26
A4
SCALE:2:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.


A
B
C
D
1
2
3
4
5
6
Ø80.00
Ø3.00
35.00
10.00
0
10.00
35.00
3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: +/- 0.01
ANGULAR: +/- 0.1 DEG
FINISH:
DEBURR AND BREAK SHARP EDGES
NAME
SIGNATURE
DATE
DRAWN
ALEXANDER CLEATOR
22/02/2022
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
1045 Steel
WEIGHT:
DO NOT SCALE DRAWING
REV 1
THIRD ANGLE PROJECTION
TITLE:
UPPER BASE PLATE
DWG NO.
25 OF 26
A4
SCALE:1:1
SHEET 1 OF 1
SOLIDWORKS Educational Product. For Instructional Use Only.



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH: 0.08 MICROMETER
TOLERANCES:
LINEAR: 0.01
ANGULAR: 0.1 DEG

FINISH: DEBURR AND BREAK SHARP EDGES

DO NOT SCALE DRAWING

REV 1

THIRD ANGLE PROJECTION

	NAME	SIGNATURE	DATE
DRAWN	ALEXDANDER CLEATOR		02/22/2020
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: WHEEL ADAPTER

MATERIAL: 1020 STEEL

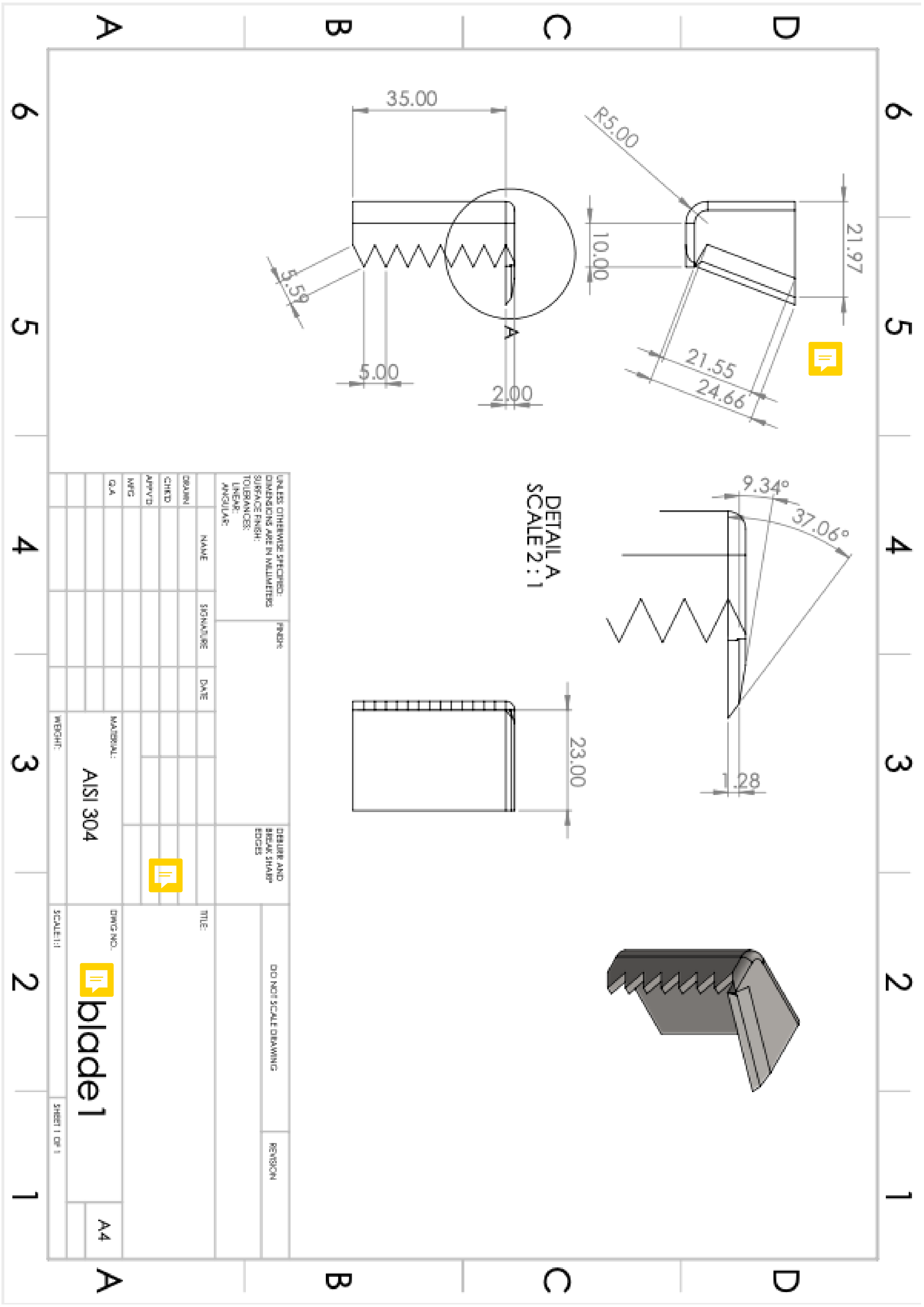
DWG NO. 26 OF 26

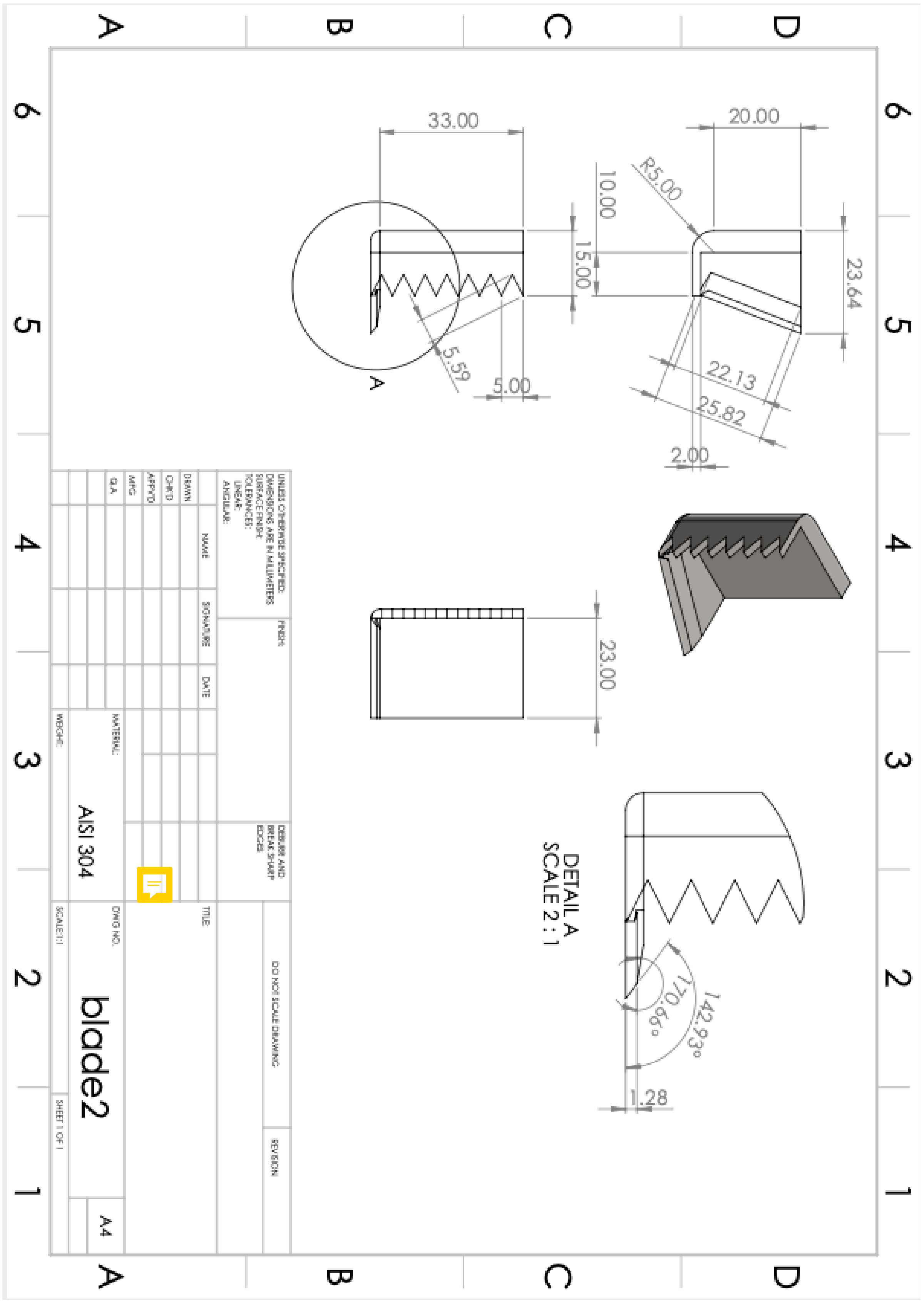
A4

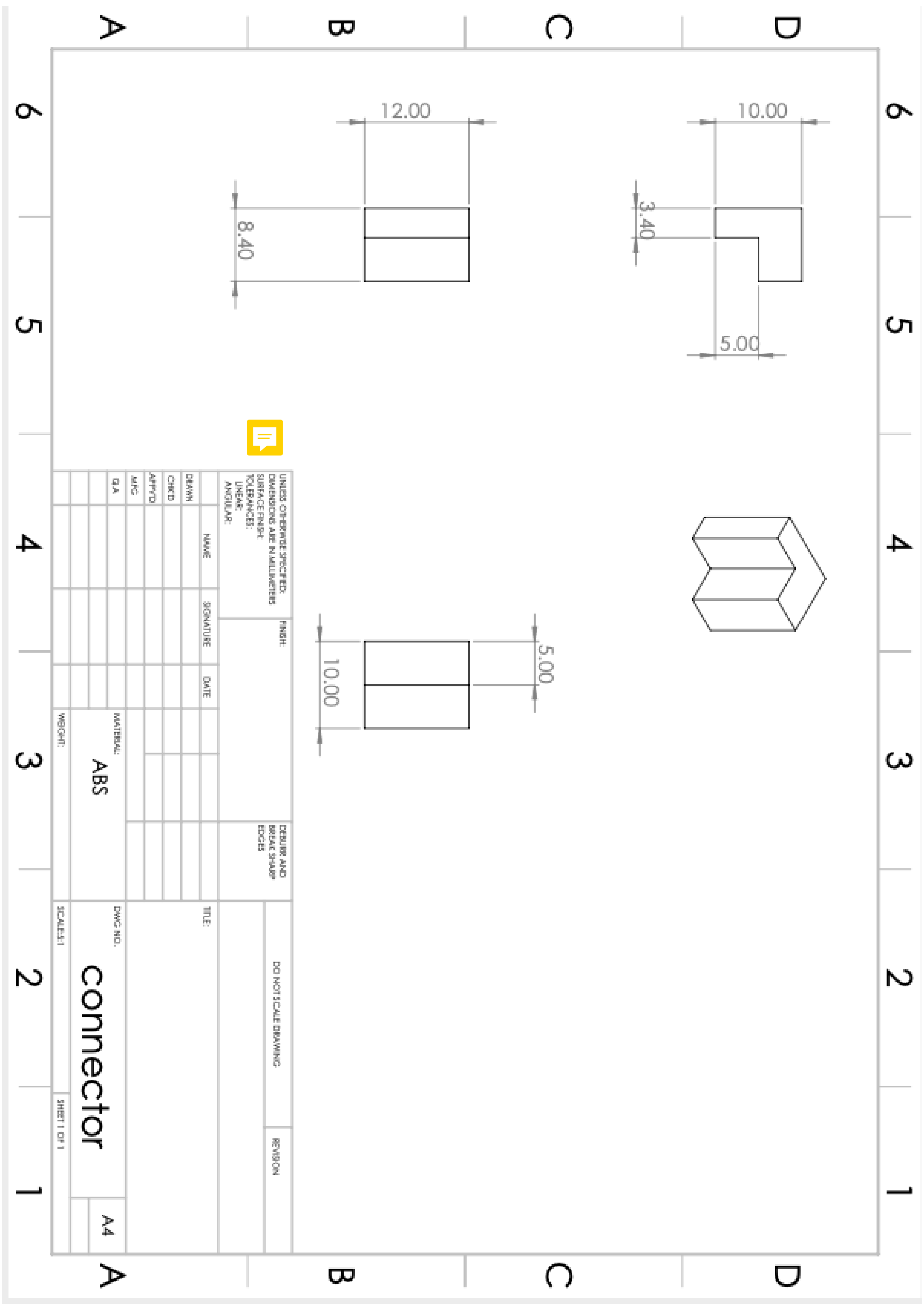
WEIGHT:

SCALE:2:1

SHEET 1 OF 1


35.00
R5.00
21.97
10.00
5.59
5.00
2.00
21.55
24.66
A
DETAIL A
SCALE 2 : 1
9.34°
37.06°
1.28
23.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
AISI 304
DWG NO.
blade1
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1


33.00
20.00
R5.00
10.00
15.00
23.64
5.59
5.00
22.13
25.82
2.00
A
23.00
DETAIL A
SCALE 2 : 1
170.66°
142.93°
1.28
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR:
ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
TITLE:
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
AISI 304
DWG NO.
blade2
A4
WEIGHT:
SCALE:1:1
SHEET 1 OF 1


12.00
8.40
10.00
3.40
5.00
10.00
5.00
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:
FINISH:
DEBURR AND BREAK SHARP EDGES
DO NOT SCALE DRAWING
REVISION
NAME
SIGNATURE
DATE
DRAWN
CHK'D
APPV'D
MFG
Q.A
MATERIAL:
ABS
WEIGHT:
TITLE:
DWG NO.
connector
A4
SCALE:5:1
SHEET 1 OF 1

Part 3: Prototype Structure, Fabrication Plan and Cost

Prototype Functionality

The prototype constructed is as fully functional model of the finished design. It is a fully autonomous device capable of following a magnetic strip, while picking cherry tomatoes. Each subsystem - autonomous navigation, the robotic manipulator, image processing, and end-effector - were developed relatively independently and integrated into a single system.

The navigation system integrated in the prototype uses an 8-bit magnetic line tracker that responds based on the relative position of the tracker with the magnetic strip that it is following. Based on the data provided by the sensor, differential steering is used to control the device accordingly. During testing, this system proved to be very effective at the prescribed speed of 0.5m/s and was capable of tracking curves with a radius of as low as 1m.

The manipulator was less effective as after purchase and implementation it was discovered that it is only capable of open loop control, and that it was not possible to request the position of the integrated servos to allow for closed loop control. As a result, steady state error was observed in the z-direction as a result of the manipulators included controller not accounting for its own weight, rectifying this issue was not possible with the included controller as the error was non-linear and could not be calibrated out of the system.

Due to inefficient scheduling near the beginning of the course, the revised design of the end-effector was not implanted as was originally intended, rather a modified version of the manufacturers' gripper was used to cut the stem of the tomato vine. Unfortunately, the open loop control, and lack of feedback from the manipulator did not allow for successful implementation of the cutting mechanism except for in a limited workspace where the moment generated by the weight of the arm is minimal.

The image processing however did work very well and was able to effectively return cartesian coordinates for the centroid of the tomato, from which the stem could be cut above the tomato, checking that no other tomatoes are being cut accidentally. During testing, image processing software was shown to be able to identify tomatoes effectively, even if they were in front of each other, were moving slightly, or were in front of a complex background. Some accuracy was lost if the tomatoes were blocked by leaves or other foliage, as that influenced where the software identified to centroid to be.

Manipulator Assembly Instructions

1. Attach one base LDX-218 servo mount to an upper base plate using four 10mm M3 screws and four M3 nuts.
2. Attach the metal servo plate on a separate LDX-218 Servo to the lower base plate. Insert the servo plate into the hole in the base plate and attach using four 10 mm 3M screws and four M3 nuts.
3. Position one large bearing between the upper base plate the lower base plate ensuring the M3 screw holes are aligned.
4. Connect the bearing, upper, and lower base plates using 15mm M3 screws and M3 nuts, tighten to apply equal circumferential pressure to the bearing.
5. Align two base rings with the completed assembly such that the screw holes on the outer rings align radially with the base plates. The outer rings should be covering half of the bearing both above and below.
6. Connect the outer rings together using 12mm M3 screws and 15mm threader columns. Tighten evenly around the circumference of the rings, to apply consistent pressure to the bearing.

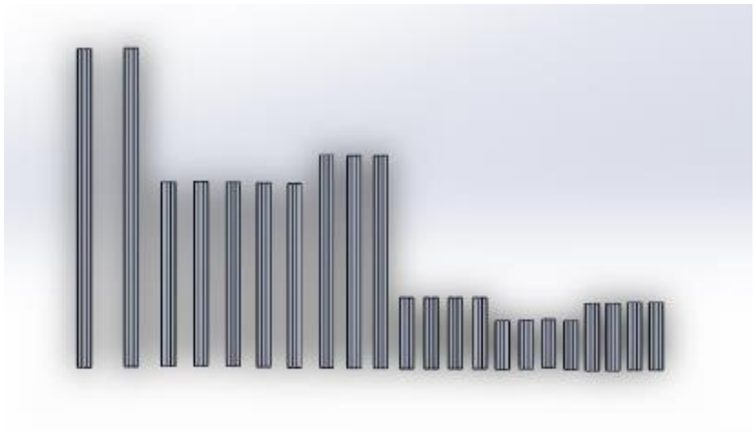
7. Position a base ring with the screw threaded columns at the base of the assembly, connect using 40mm columns.
8. Attach one LDX-218 Servo to the LDX-218 servo mount, aligning the driving plate with the indentation on the mount, and attaching using three 5mm M2 screws.
9. Attach two large U brackets to each other using 10mm M3 Screws, aligning the center holes on the base of each with each other.
10. Attach one end of the U bracket assembly to the assembled base and attach the other to the driving plate of another LDX-218 servo.
11. Attach a small U Bracket to another large U bracket using 5mm 2M screws and 2M nuts similarly to the large U brackets previously assembled.
12. Attach the small U bracket to the LDX-218 servo using 5mm 2M screws.
13. Attach two LDF-06 servo mounts together rotated 90^o relative to each other using 10mm 3M screws and nuts.
14. To one side of the large U bracket in the previous assembly, attach the servo mount assembly and small bearing such that the bearing is in the center hole of the large bracket.
15. Attach one LFD-06 servo to the servo mount using 10mm 3M screws and 3M nuts.
16. Attach the servo driving plate to the other side of the large bracket using 5mm 2M screws.
17. Attach a second LFD-06 servo to the available servo mount using 10mm 3M screws and 3M nuts.
18. Attach the gripper assembly to the driving plate of this servo.

Gripper Assembly Instructions

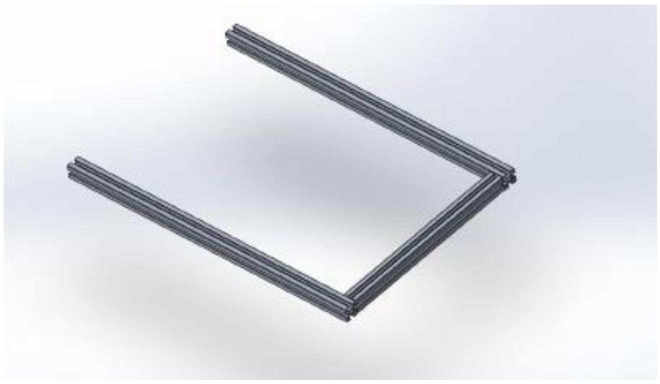
1. Attach two servo mounting clips to opposite sides of the base plate using 10mm 3M screws.
2. Attach one driving gear and a spacer to the base plate using a 15mm 3M screw.
3. Attach one LDF-06 servo to the four mounting posts on the base plate. Attach a second driving gear to the driving plate on the servo. The two driving gears should be level with each other.
4. Attach two grippers to the holes at the end of each other driving gears. Connect the driving gear, and two grippers with a 15mm M3 Screw and M3 nut. Mirror this connection on the other side of the mechanism.
5. Attach one link above the base plate, and a spacer and link below using a 15mm M3 screw. The link above the base plate should be level with the driving gears.
6. Attach the opposite ends of the links to the gripper. One link above the two gripper pieces, and one link below them. Connect the four components together using a 15mm M3 screw and M3 nut.
7. Repeat steps 5 and 6 on the opposite side of the mechanism.
8. Attach a single utility knife blade to the end of each of the grippers. Move the grippers manually to ensure the blades pass each other closely.

Chassis Assembly Instructions

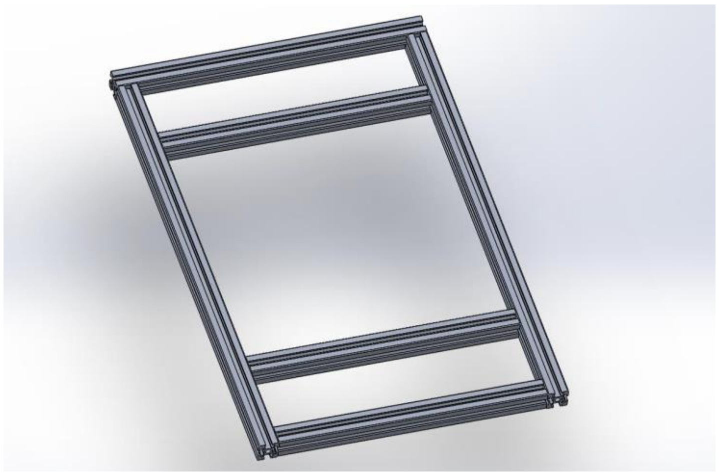
1. Cut a series of 2020 extruded aluminum beams to the following lengths:
    a. 450 mm (x2)
    b. 300 mm (x2)
    c. 260mm (x5)
    d. 100mm (x4)
    e. 95mm (x4)
    f. 70mm (x4)



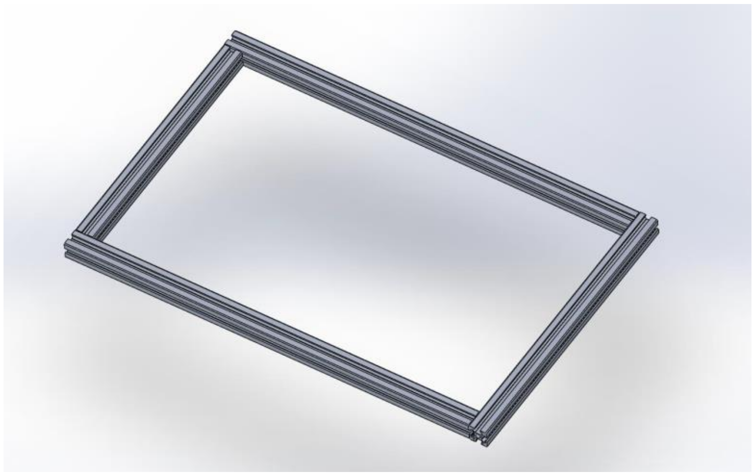
2. Connect two 450mm beams with a 260mm beam and fasten using 12mm M5 screws and corner brackets



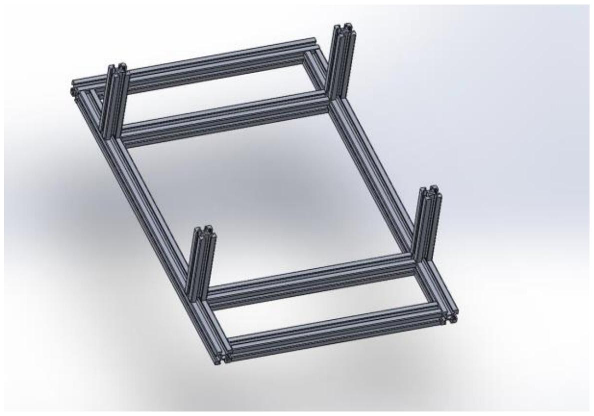
3. Insert two 260mm beams centered about the middle of the 450mm beam. Leave a space of 260mm between them.



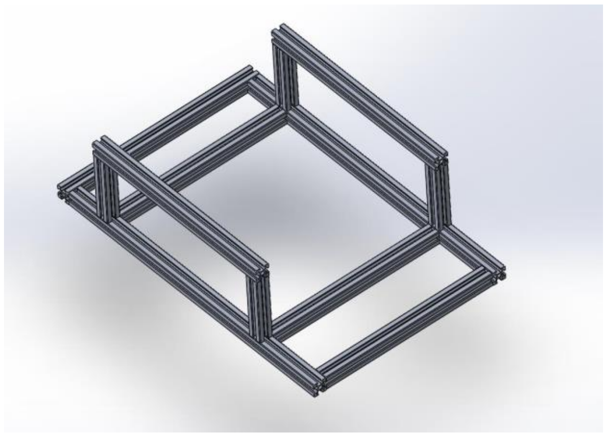
4. Close the opposite end of the frame using a 300mm beam, allowing this beam to overhang the ends of the 450mm beams.



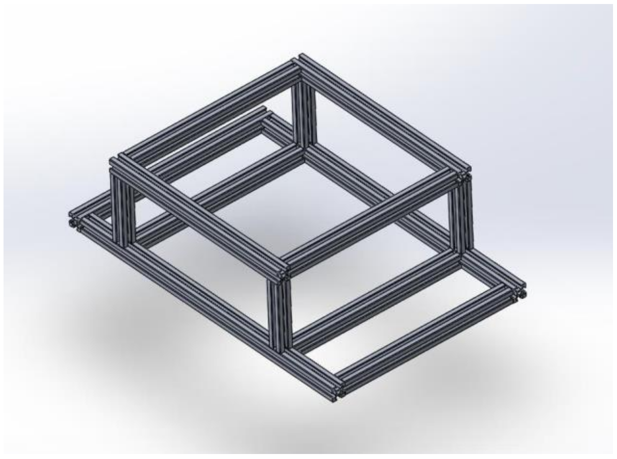
5. Install four 100mm beams at the corners of the inner 260mm square frame.



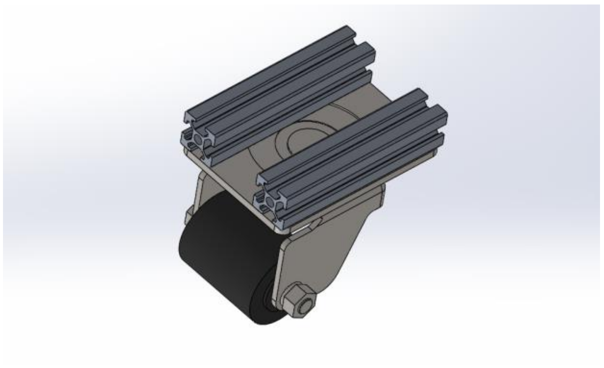
6. Install two 300mm beams on top of the 100mm beams, parallel to the 450mm beams already installed.



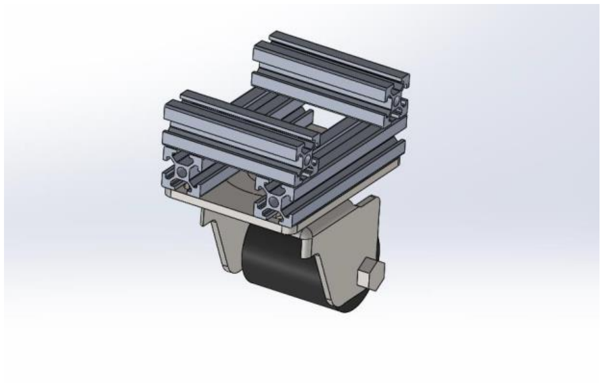
7. Connect two 260mm beams perpendicular to the 300mm beams, this completes a 300mm square platform above the base of the frame.



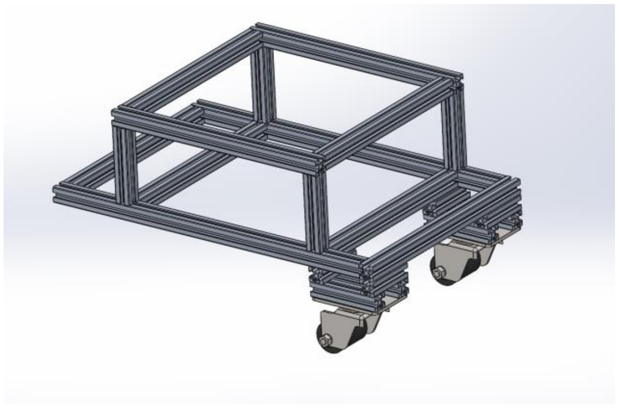
8. Attach two 95mm beams to the flat plate of each caster.



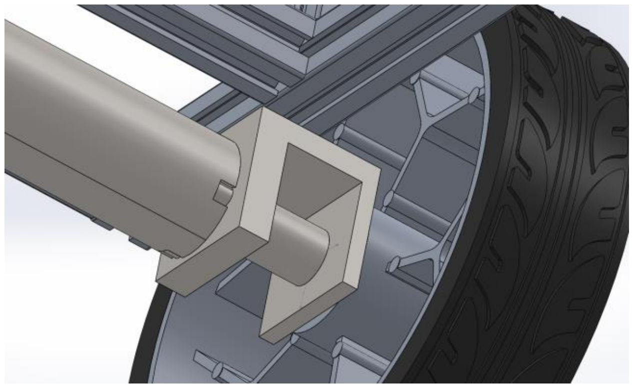
9. Attach two 70mm beams perpendicular to the 95mm beams to each caster.



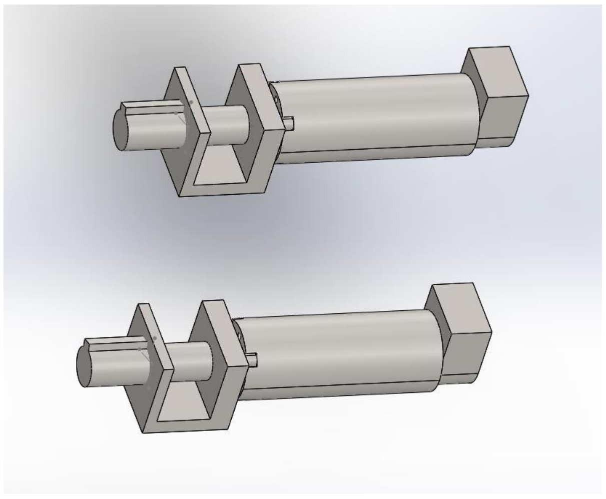
10. Attach the caster assemblies to one end of the frame.



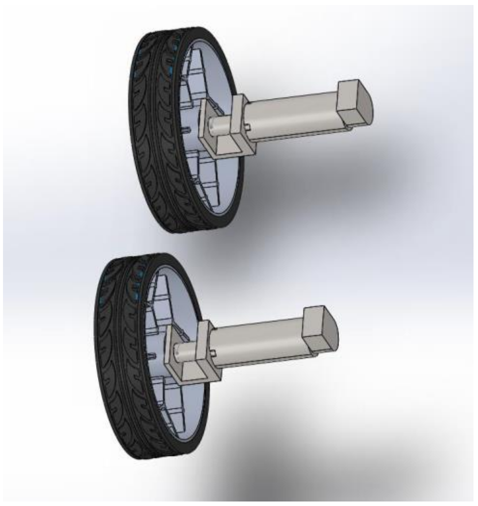
11. Attach the motor mounts to the frame using machine screws. Position the motor mount the corner of the upper square frame.



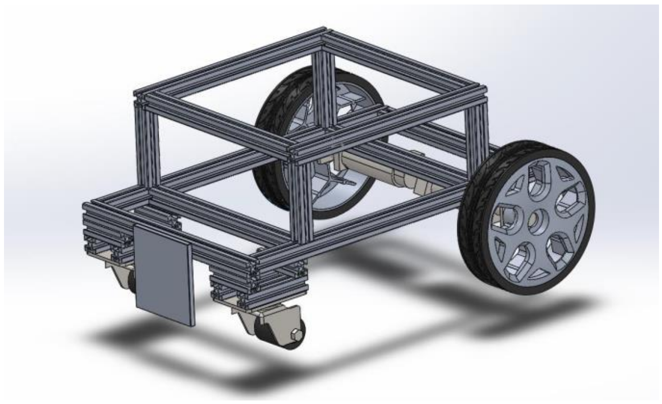
12. Attach two motor mounts to each of the 24V DC motors.



13. Press fit the wheels to the motor shaft, aligning the key and keyway on each.

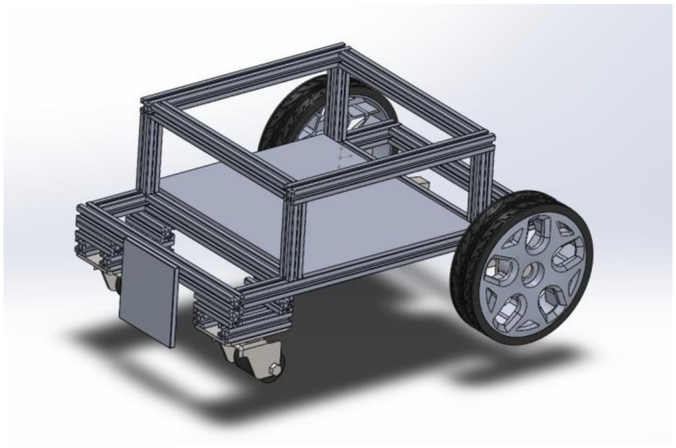


14. Position the magnetic line tracker sensor mount in the center of the front of the frame.

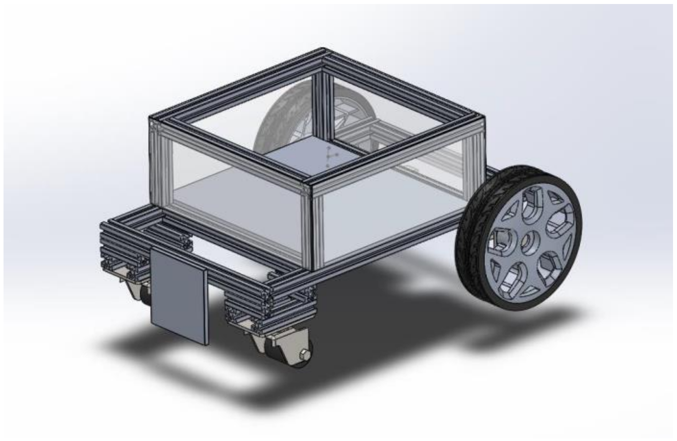


15. Attach the magnetic line tracking sensor to its mount using the appropriate holes.

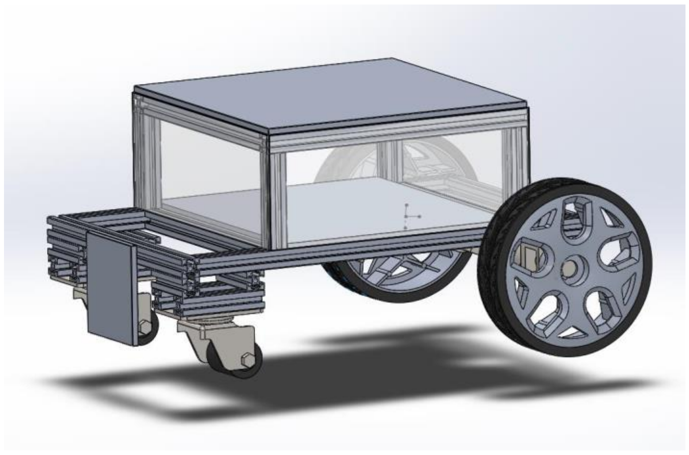
16. Install the container base plate inside the square frame.



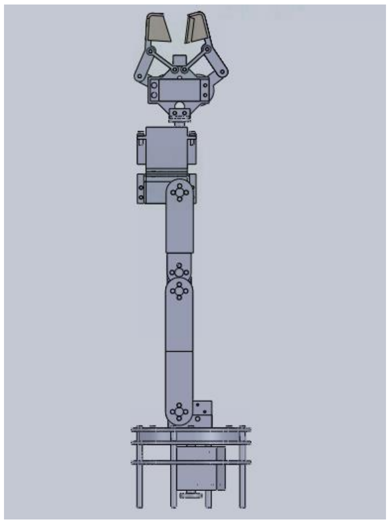
17. Install the acrylic side panels to the sides of the square frame. Ensuring the panel with the hole is installed on one of the sides.

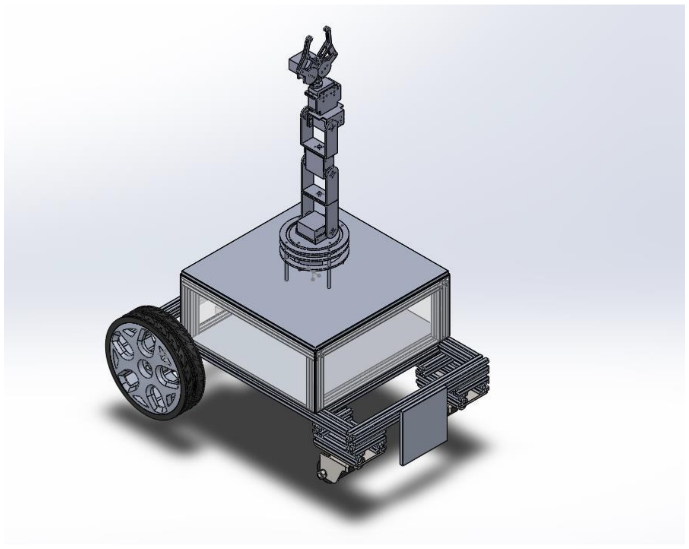


18. Attach the upper container plate using machine screws to the top of the square frame.



19. Attach the manipulator to the top of the container plate using locking nuts.





20. Mount the motor driver, microcontroller and battery above the 24V motors, and wire according to the appropriate diagram.
21. Connect the flexible collection tube to the end effector collar using the previously attached L bracket. Connect the opposite end of the tube to the hole in the side of the collection bin.

Alternative Fabrication Options

In developing the prototype aluminum components were primarily used due to their light weight and ease of availability. Furthermore, aluminum can easily be machined using the available equipment in the Student Machine Shop. Steel was considered, however using a steel chassis would have greatly increased its weight, which would negatively impact battery life.

The cost of the container could greatly be reduced by using a prefabricated container rather than assembled panels. However, this predominantly only affected the prototype as the final design (being mass produced) would be much cheaper due to the economies of scale in mass producing simple plastic panels. Furthermore, in the final design acrylic is specified as the material to be used for the panels, although polyethylene, polycarbonate or other plastics would function similarly. Acrylic was chosen due to its properties as being very hard and tough, which allowed for vigorous testing and abuse during the prototyping phase.

Material Costs

The following table outlines a comprehensive, itemized invoice of the material costs associated with the construction of the prototype and ultimately the final design of the fruit picking robot.

Item	Quantity	Unit Cost	Total Cost
Drive wheel	2	$18.63	$37.26
Driven wheel	2	$12.61	$25.22
8-Bit magnetic line tracking sensor	1	$55.51	$55.51
BTN7971 motor driver	1	$24.09	$24.09
MD36 35W 24V DC Motor with optical encoder	2	$73.92	$147.84
Ultrasonic sensor	1	$1.32	$1.32
Ultrasonic sensor mount	1	$0.56	$0.56
24V to 5V voltage converter	1	$3.20	$3.20
M5 washer	50	$0.02	$1.00
M5-12 screw	10	$0.08	$0.80

M5-14 screw	10	$0.09	$0.90
M5 – 16 screw	10	$0.09	$0.90
M5 nylon lock nut	30	$0.04	$1.20
Aluminum beam corner bracket	40	$0.53	$21.20
80cm 2020 aluminum beam	10	$1.67	$16.70
5-DOF Manipulator and associated hardware	1	$165.29	$165.29
Acrylic Panels	4	$1.10	$4.40
100 mm flexible hose	1	$2.50	$2.50
		Total	$509.89

Potential Cost Savings

Item	Cost	Savings
Reed sensor SPST	$18.00	$37.51
24V 5300RPM DC motor	$25.84	$96.15
Injection molded wheels	$10	$24.25
Total Savings		$156.11
Potential cost		$353.78

As can be seen, the project did exceed the budget by $210, this could be addressed by using single hall-effect sensors instead of the single integrated sensor, using a lower-powered motor instead of the 24V motor specified and operating at a lower efficiency (sacrificing battery life), and at a higher average voltage. Savings also could have been had on the driving wheels. The wheel specified are cast aluminum, plastic wheels could have been used at the expense of durability. The reasons these changes were not implemented were to both improve the quality of the finished design, ease maintenance, and allow for easier testing and prototyping.

Fabricated Prototype

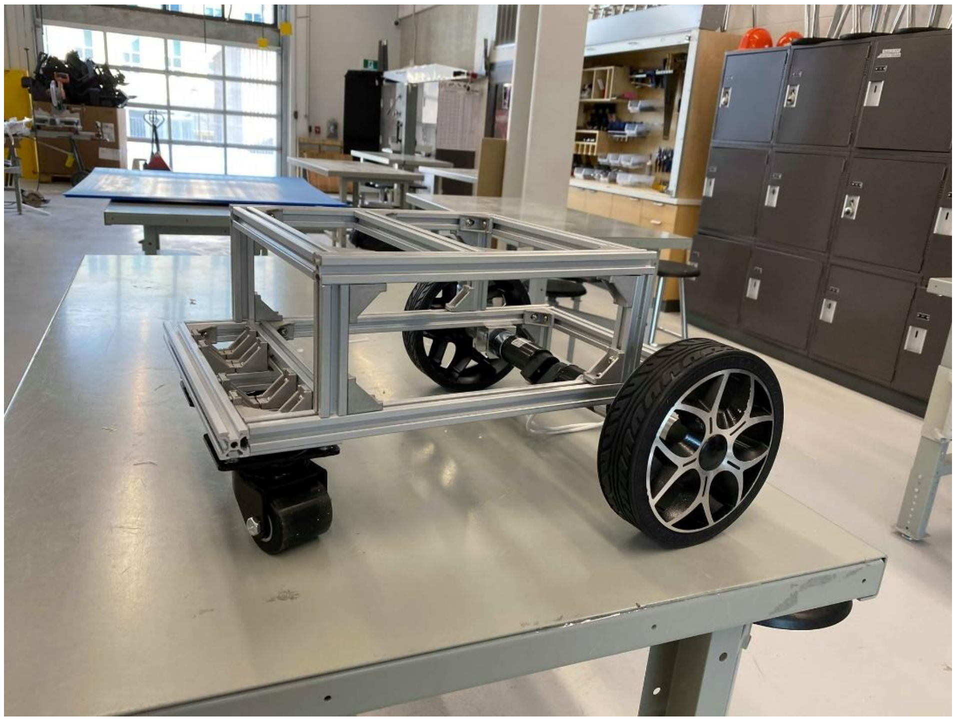


Figure 11. Fabricated Chassis

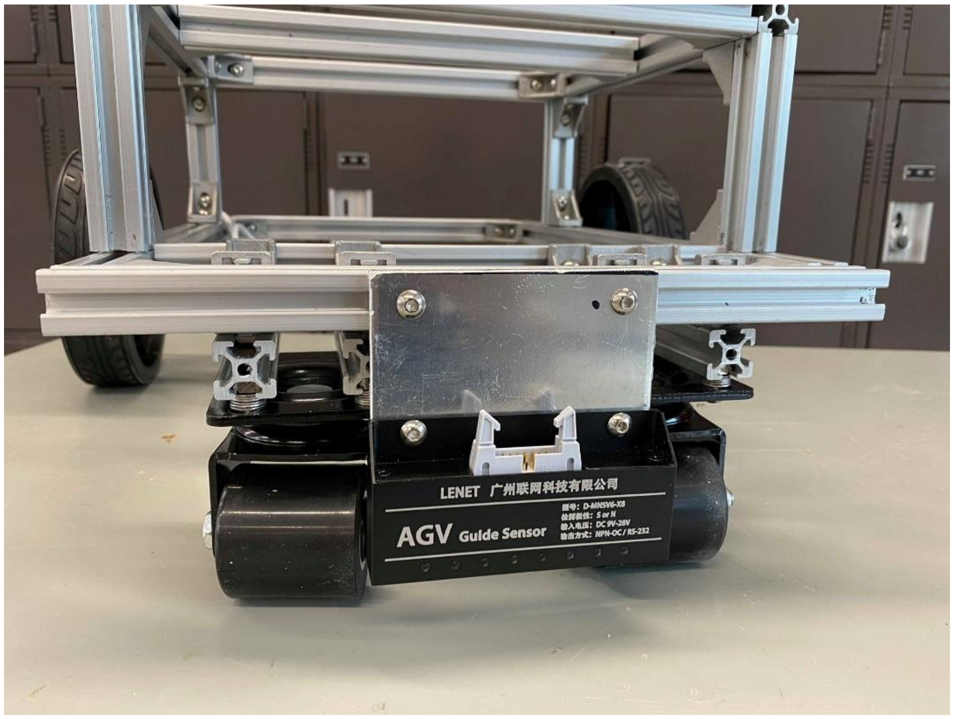


Figure 12. Enlarged view of the magnetic Sensor

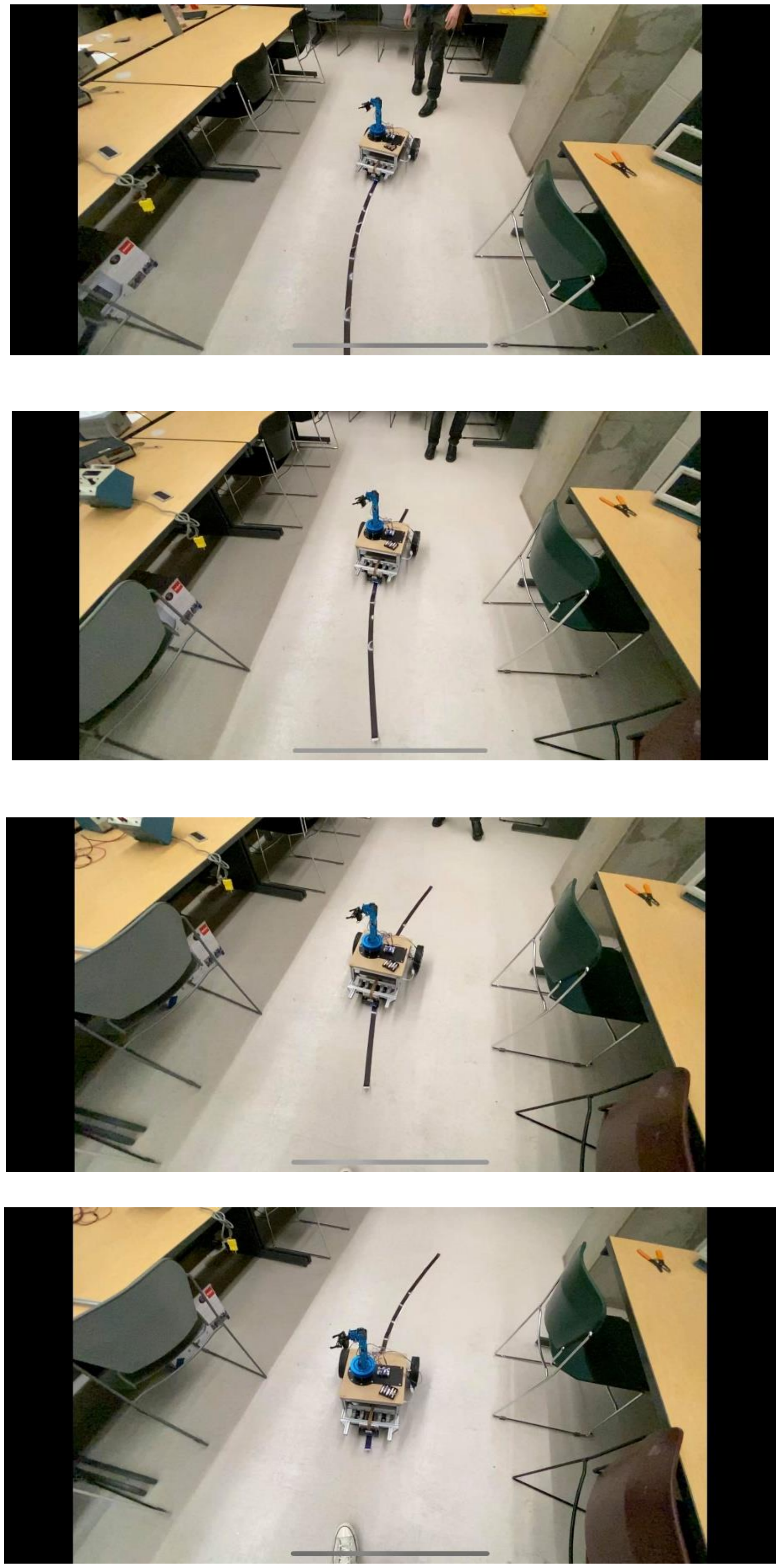


Figure 13 Demonstration of the robot following a magnetic track