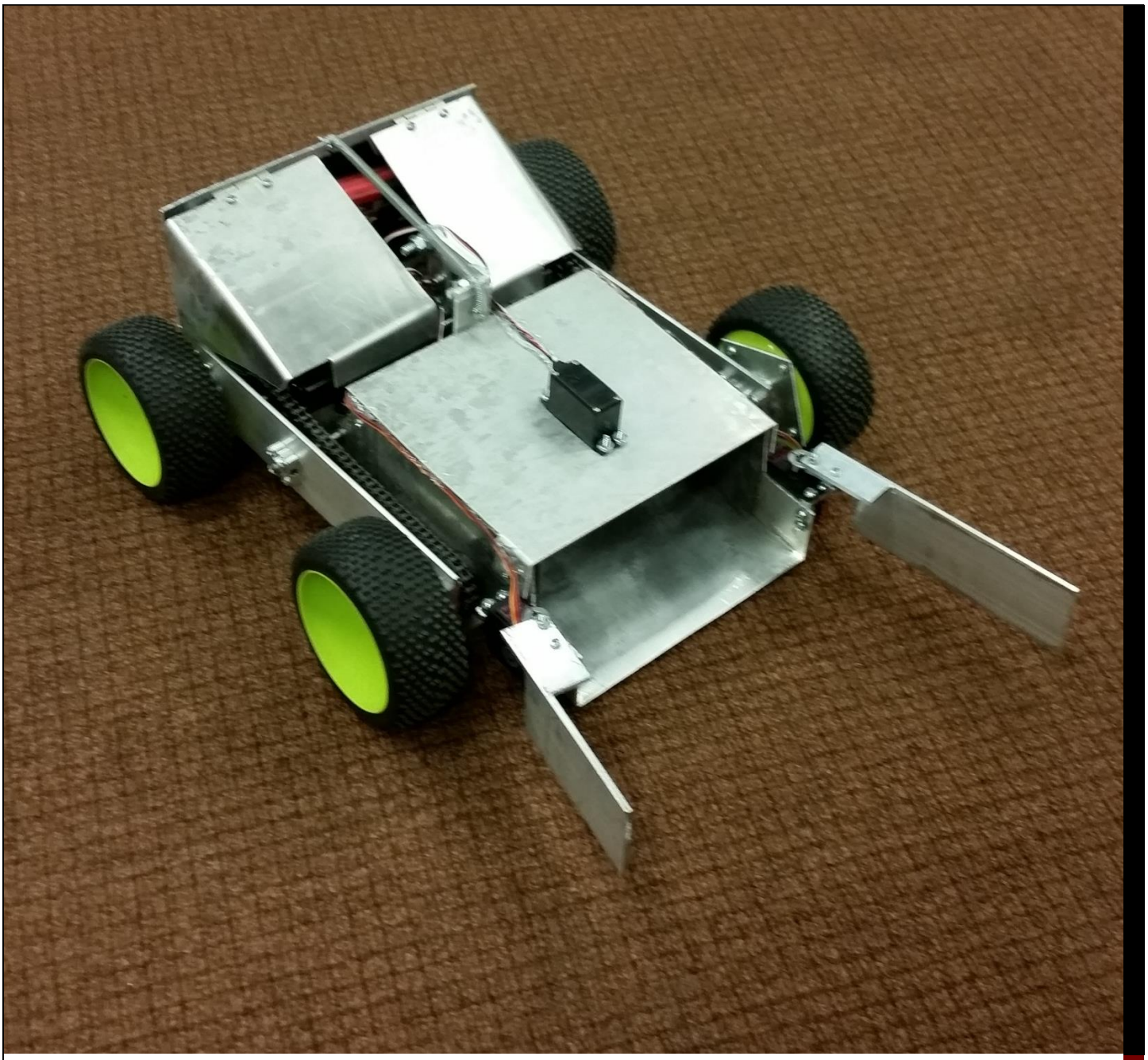


DUSTY 2.0

SIU ROBOTICS

Technical Report - ATMAE Robotics Competition – 2016

Lincoln Kinely, Sarah Handrock, Paul Schumacher, Nicholas Sjoberg, Tyler Winkleman

DUSTY 2.0

TABLE OF CONTENTS

DESIGN METHODOLOGY

FIGURE 1 – ADJUSTABLE PAN

THE OBSTACLE COURSE

The obstacle course part of the competition was the most challenging part of the competition last year and it is the part that weighed most heavily on the design. The key to having an effective robot for the burlap is having a low center of gravity and the ability to keep traction on the burlap. In testing, the best grip was obtained with knobby tires (Figure 2). Dusty 2.0 employs four-wheel drive, so if even one tire holds traction, the robot can continue to function on the burlap.

In order to keep a low profile, the design has its roots in last year's design. The highest point on the robot is 6 1/2 inches high, while most of the weight sits around 1-3 inches from the ground. This low profile allows the robot to remain stable on the climb and decline of the burlap (Figure 3). We also worked to minimize any protruding parts or screws from the robot. This prevents it from getting caught on the burlap. Additionally, the pan on the front of the robot is able to be lifted so it will not cause any friction as it climbs (Figure 1).

FIGURE 2 – TIRE TREAD

The teeter totter is a mildly challenging obstacle. With the current tires, the robot has not had any problem

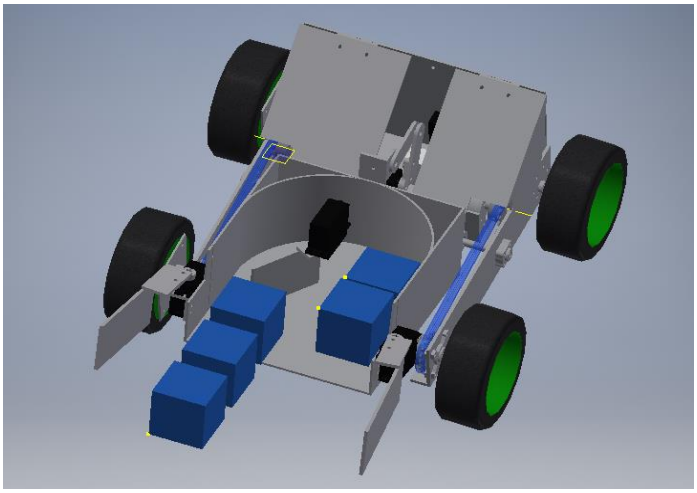
FIGURE 3 – BURLAP CLIMB

driving up and down the obstacle. One small concern is that the robot is wider than its predecessor. This allows for a larger pan. The robot will fit within the teeter totter, though with smaller margins this year.

FIGURE 4 – LINE FOLLOWING SYSTEM

For the autonomous portion of the competition, we have elected to go with a continuity test to follow the line. The test is made up of a 3d printed part that has 3 wires running through and under it that are connected to our Arduino (Figure 4). The Arduino sends a signal through the middle wire. When the robot crosses another wire it will short over the left or right wire. This will tell the Arduino to turn left or right. For collecting the blocks, we will be leaving our bumpers open and they will collect the blocks as we follow the line. Once collected, all 5 blocks will be secured in the pan with a lid and the bumpers closed.

FIGURE 5 – BLOCK UNLOADING SYSTEM



For the autonomous unloading portion, we will be using a continuous rotation servo. The pan will have a curved piece of aluminum that will fit the shape of the pan (Figure 5). This gives the back of the pan a curved shape as well as covering up screws that could catch on our blocks. The servo and middle bumper are attached to the pan lid on the top. When activated the servo will proceed to spin and unload the blocks one by one as we slowly back up. The pan is small enough that it will only allow 1 block to exit at a time as they cannot sit side by side.

Once unloaded, we will use the front of our robot to push the blocks together to spell out ATMAE. We decided to use this instead of adding an extra mechanism as a sorting mechanism would likely impair us during other parts of the competition. We did accelerate our build schedule to allow 2 weeks of time to practice the sorting.

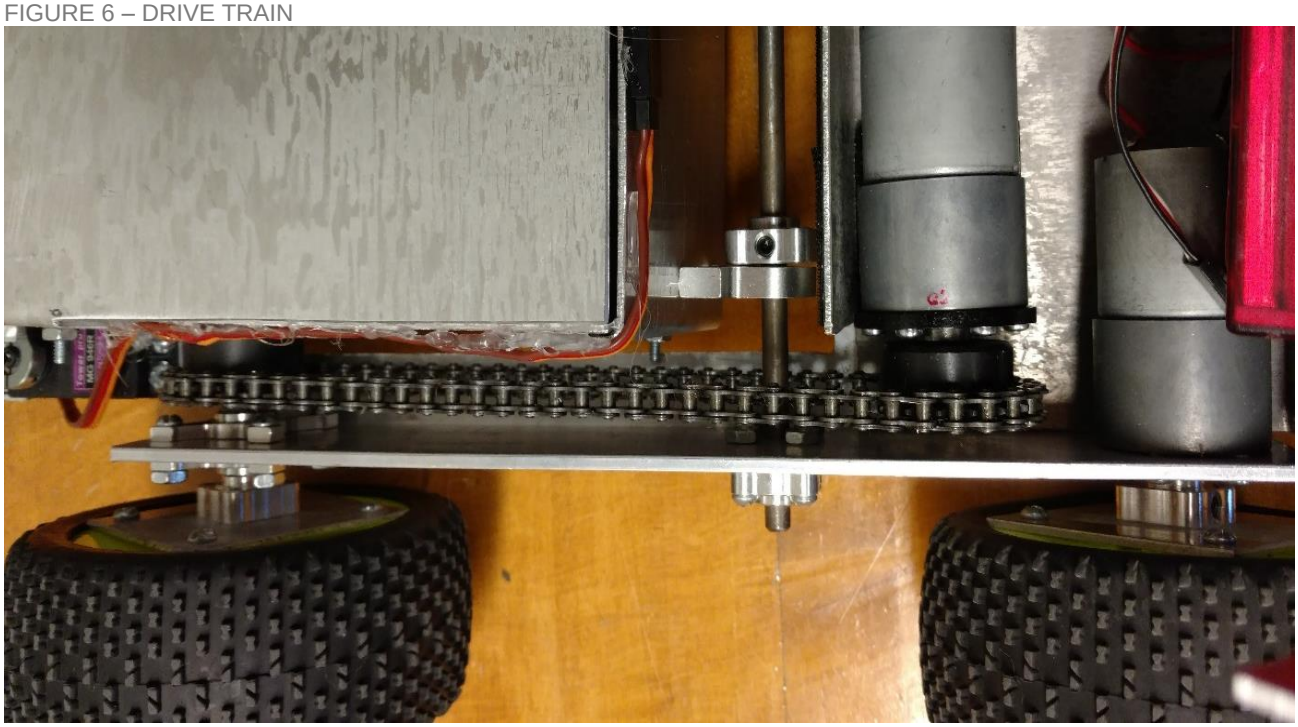
THE DRIVE TRAIN

For our drive system we used 4 12V 170 oz-in 200rpm metal gear motors. The rear wheels are directly driven onto the wheels, while the front wheels are driven off of a chain and sprocket system (Figure 6). The reason for this type of setup is due to the size limitations. We needed four-wheel drive to get up the burlap, in fact when we tried getting up without four-wheel drive, we barely got half way up. Our loading methodology consists of a pan that runs along the ground. If we had the motors directly driven onto the front wheel we would have cut down the size of the pan by at least 2 inches. The gears do add an extra layer of complexity, but the space saving is worth it. It also allows us to condense our wiring to our chassis area. The only wires that needs ran through the robot are the bumper servos in the front as well as the line following underneath.

The wheels we are using are typically used for off road RC buggies. This type of tread is critical for our success on the burlap. We needed them to have a large amount of traction. To attach the wheel, we elected to have a metal plate to mount to the rim. This rim has 4 holes that attach on the outer area of the rim. The main reason for this is because we have had trouble in the past with wheels stripping. With the wheels being plastic stripping was a big concern. The metal plates were then attached to a set screw hub that efficiently attached to the wheels. The set screw hub then was either attached to the motor shaft or sprocket shaft with a set screw.

With the front wheel attachment, we used 2 bearings to hold a shaft that was connected to both the sprocket and the set screw hub. Originally we only had one, but to lessen the load on the bearings we have one on each side of the aluminum side plates.

FIGURE 6 – DRIVE TRAIN



RELAY RACE

For the relay race, our robot will be completing the task with a pan that runs against the ground as well as 2 bumpers to assist in loading the blocks. The pan consists of .08 in aluminum that measures 7X8X3. The pan size was created by taking the dimensions of the largest puzzle piece and adding a little extra wiggle room (Figure 7). The pan also rotates on an axle. This is accomplished by attaching bearings to the back side of the pan, which in turn, rotate on a 1/4in steel rod. This is raised and lowered by a servo that is hooked up with a system of linkages and arms. Due to the weight of the pan, we have a spring that helps the servo lift the pan.

When we are loading the puzzle pieces, the pan will be in the lowest position. We also left some slack in the pan servo's mount so the pan is allowed to buck up if it gets caught on the carpet. Being flush against the carpet is critical for the success of this robot. Since we are relying on simply the bumpers flipping them in, we must make sure we minimize the chance that the puzzle pieces getting caught.

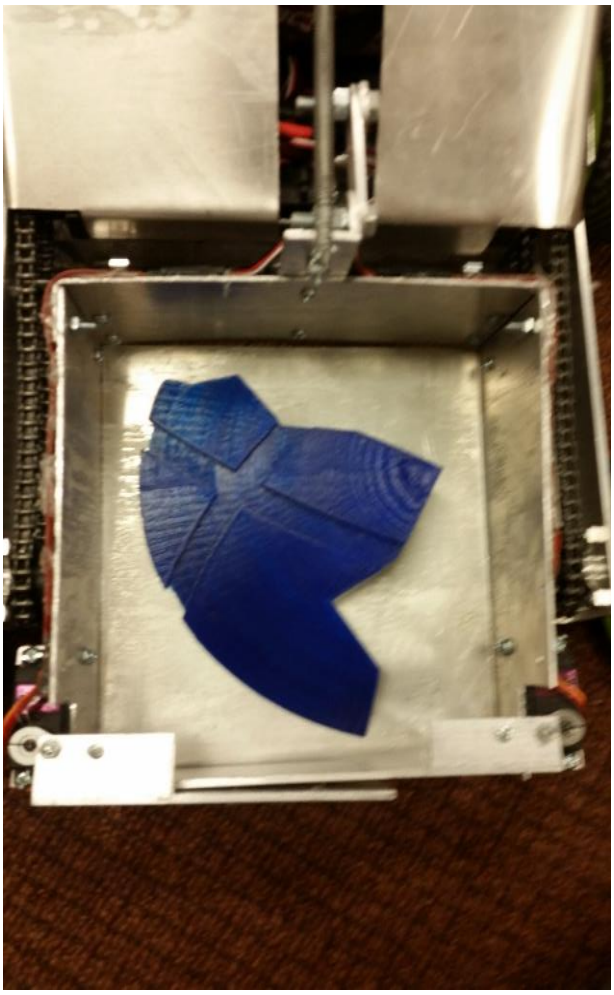


FIGURE 7 – PAN SIZE

The bumpers are made of 2in angle aluminum. This allows for a solid mechanism that has a large surface area and easy attachment. We have the bumpers having different lengths. One arm is 4 in long, this one is mainly used for staging the block. The other one, 7 in long, is used for completely loading the block into the pan.

Once the block is loaded, it can never touch the ground. To keep this from happening, we will raise our pan up, making sure the entire pan is off the floor.

For the unloading portion, we will rely on the law of inertia to unload the puzzle pieces. We will be coming into the area at a decently high speed. If we open the bumpers and lower the pan, and slam into reverse at the very end; the puzzle piece will easily slide into our unloading zone. This will require us to get a lot of practice in perfecting this technique, however, we allotted 2 weeks in our project schedule for strictly practice and last minute modifications.

CONSTRUCTION

Our Robot was made almost entirely out of aluminum. Aluminum is reasonably strong and "is a softer metal and is therefore easier to work with" (Materials for Robot Building: An Introduction, n.d.). Aluminum is also more abundant than steel, as we already had a surplus at our shop, with any extra readily available at Lowes. This paired with its low weight made it the clear choice for our build.

Since we focused on our cad very early in the design process, when it came to building, we were able to print off templates for our components to mark cut lines and hole locations (Figures 9, 12 and 13). Often in the past we had trouble with parts lining up. This eliminated the variability of our hole locations. It was also much simpler and quicker than measuring out all of our holes.

Our metal bender was a key tool in the construction of the robot (Figure 13). "Metal bending will produce essentially no change in the thickness of the sheet metal while creating a desired geometric form" (Sheet Metal Bending, n.d.).The metal bender was used in the making of our pan, chassis, pan lid, and covers. We were able to cut out using brackets and screws for many of our attachment points. We also have a much stronger pan. Instead of having a few contact points, our entire edge acts as a connection point to the rest of the robot. This cut down the number of protruding screws as well as the number of parts.

The electrical components are held on with a high strength Velcro (Figure 15). Often mounting electrical components can be tricky as they don't have good or strong holes for mounting or were not manufacture with the intention of being mounted to metal. We decided to use Velcro to stick each component on. It also doubles as an insulator with reduces the chance for it to short across the metal chassis. It also allows for our wiring and soldier to come easier as we can remove the components, soldier then, and return them without wasting time messing around with screws. It also makes it super convenient for replacing components if one burns out or needs rewired.

Tapping was a vital manufacturing process for this year's robot (Figure 10). The best example is our servo mounts. We had 4 tapped holes on each servo mount, 2 for attachment and 2 for holding on the servo. These tapped holes eliminated the use of nuts on our compact design. It may not seem like a big deal but for where the servos are located an extra quarter of an inch could make the difference, especially for loading and unloading blocks. We also tapped holes on our motor mounts. The motor mounts are in a very tight area of the robot. Getting nuts to these locations is nearly impossible. Having tapped holes on the mount allowed us the option to make a quick motor change if need be, without tearing the robot apart to get at a few nuts.

FIGURE 8 – PAN SIZE

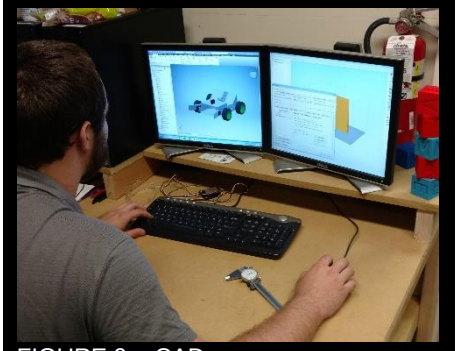


FIGURE 9 – CAD

FIGURE 10 – TAPPING

FIGURE 11 – METAL BENDING

For the servos we decided to go with all metal servos and servo horns. Although “metal horns are the heaviest and having the highest wear rate of all servo horn types, they offer incredible strength” (RC Basics, n.d.). Often servos come with cheap plastic servo horns that are easy to strip. We decided to pay a little extra and get some high quality horns. They even included set screws for the teeth and threaded holes for attachment to our bumpers.

We only used 3d printing for one component on our robot. Some teams will say using 3d printing is innovative, but the truth is that 3d printing cannot compete with “traditional manufacturing processes and is not yet suitable for mass production” (White, 2015). It is great for prototyping and custom parts, but as for reliability and strength, metals will always have the upper hand. The one part we used it for was our line following plate. This allowed us to put custom holes into a triangular piece that will be super glued to the bottom of our pan. This piece also makes the wires that run to our continuity test hit the ground flat instead of at an angle.

One method we found very useful is the use of standard parts. If you look closely, a lot of the components on the robot come from the same type of material. For example, the side plate, and all three of the bumpers are made from 2in angle aluminum. Also the chassis, covers, pan, and pan cover are all made of .08in thick sheet aluminum. The use of standard parts made it very simple for identifying correct materials and we also only had to buy one big sheet rather than several small ones. The use of standardization was probably best used for our screws. We only had 4 sizes of screws, with the majority being #6 and #8. We did have a few uncommon parts, but that was mainly because they were purchased components. The standard screws make assembly quite easy as instead of finding which type of screw goes where, you can have only one or two bins to choose from. Something as small as reducing the types of screws can save a lot of time and frustration in the long run.

Another innovative feature for our robot is the use of a spring for our pan lifting servo (Figure 14). This spring sits in tension with our pan, holding it at around the halfway point. This drastically reduces the wear and tear on our servo. Instead of lifting the solid metal pan alone, the spring takes much of the work away from it. Instead the servo is mainly just pushing the spring in and out. We have also seen it have a much quicker response time than without the spring.

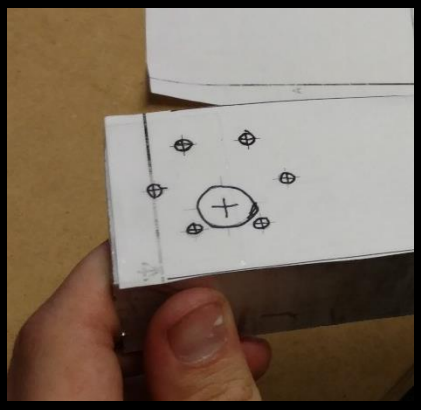


FIGURE 12 – TEMPLATE

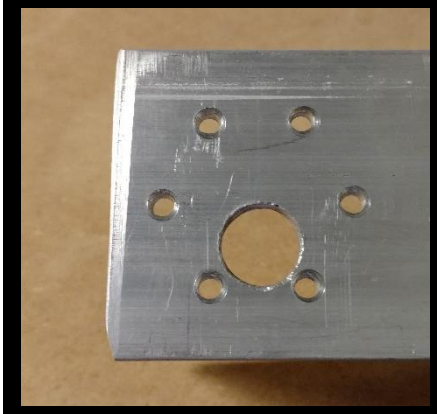
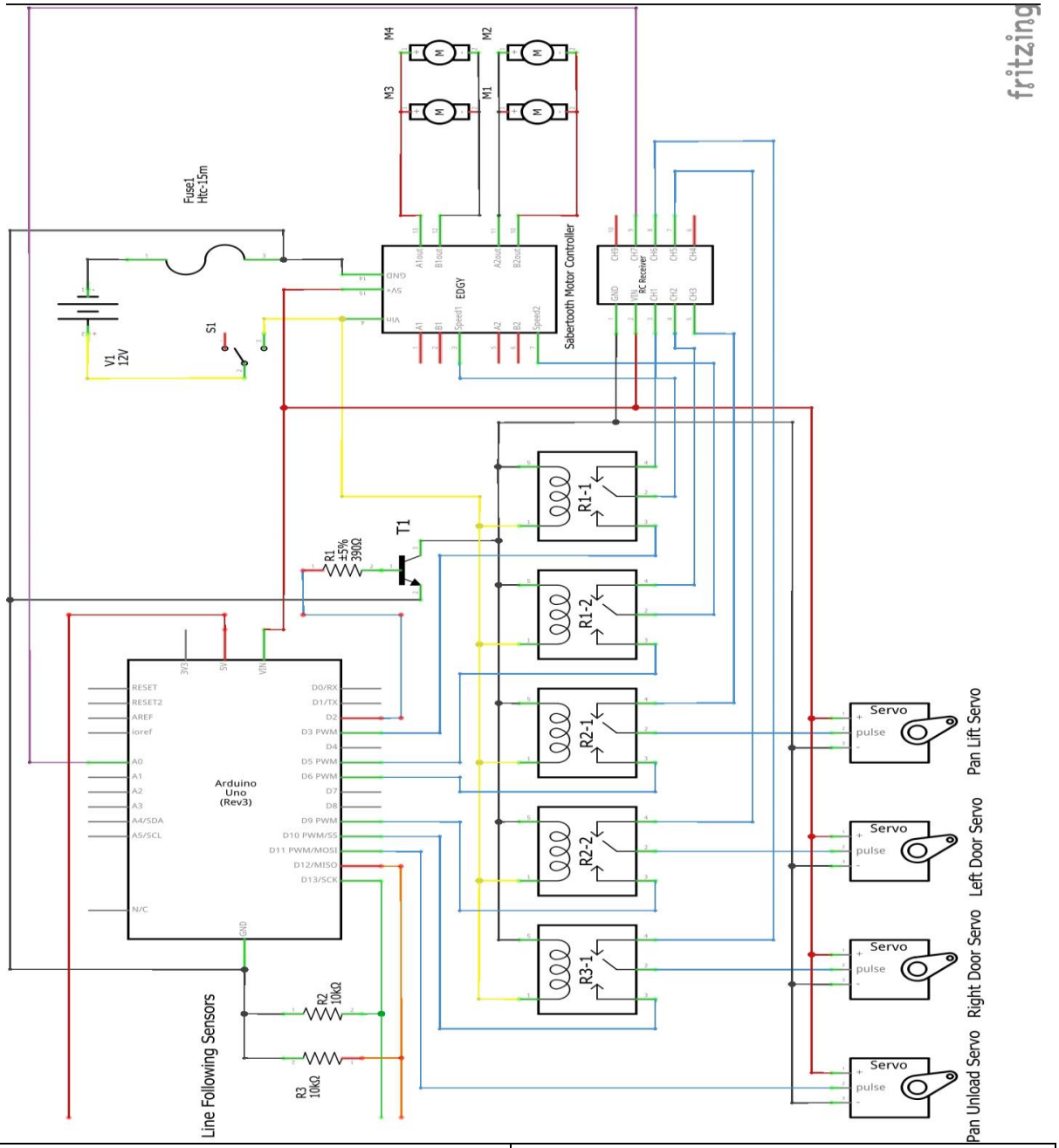


FIGURE 13 – RESULTING HOLES

FIGURE 14 – SPRING

FIGURE 15 – VELCRO FASTENING

ELECTRICAL SCHEMATIC



Color	Meaning
	+12v With Respect to Negative
	+5v With Respect to Negative
	PWM Signal
	Negative
	Autonomous Trigger Signal
	Left Autonomous Sensor
	Right Autonomous Sensor

WIRING

Last year our control methodology was very complex and relied on 3 wireless connections and a common WIFI network. This year we decided to simplify the controls to improve reliability. The brain of Dusty 2.0 is a standard Arduino Uno. The Arduino is constantly watching the input value from channel 7 of the RC Receiver. When the input is in the lowest range, the Arduino enters the line following autonomous mode. If the input changes to the middle range, the Arduino switches back to normal operation. When the input is in the high range the Arduino switches to the autonomous mode that handles unloading the ATMAE blocks.

The heart of Dusty 2.0 is a Sabertooth Dual 12A 6v-24v RC Motor Driver. The Sabertooth was chosen because it is very small, has a good set of features, and can handle 12 amps per channel. The Sabertooth is in a tank steering mode where one channel input is drive forward and backward while the other channel is steering input. The Sabertooth handles all of the channel mixing on its own with no need to change advanced settings on the radio controller. There are four motors total connected to the motor controller. They are divided up into left and right side. Both motors from each side are wired in parallel with each other so that they are supplied the same voltage. One set of motors is wired with the polarity reversed because the motors are physically facing the opposite direction.

The power source for Dusty 2.0 is a Venom 3.3Ah 35C 3 cell lithium polymer battery. The battery is rated at 11.1v, however the fully charged voltage is 12.6v. The battery is rated at 35C, so the rated continuous current is 3.3Ah*35C = 115.5 amps with a peak current output of 165 amps. This value is confirmed in the specifications listed at the top of the battery. An important note about this battery is that it is a hard case LIPO battery, meaning that there is a hard plastic case around the lithium cells. This makes it much harder to puncture a cell causing a lithium fire. The power from the battery goes through a switch and then goes to the input of the Sabertooth motor controller. There is also a fuse on the negative side. From there 12v is distributed to the relays and negative goes to almost all of the electronic components.

Initially our idea was to feed the signals from the RC receiver through the Arduino to the servos and motor controller. However, there was a lot of lag and other issues with that solution. So we switched to a mechanical switching solution. When the Arduino triggers either autonomous mode, it triggers T1. T1 is a standard transistor which is used to go from the low voltage and current triggering that the Arduino is capable of to higher voltage and current triggering. When the transistor is triggered, it triggers all of the relays. Each relay mechanically switches the signal connection to the servos and motor controller from the RC receiver to the Arduino. This gives us the added advantage that there is absolutely no way the operator can do anything in autonomous mode except for exit it.

Last year we tried a variety of sensors in several different configurations in order to autonomously follow the line. We found that those sensors worked great on a solid background, but not on the seemingly random patterns of the carpet that the competition was held on. We decided to simplify it as much as we could. There are three metal contacts on the bottom of the pan on Dusty 2.0. The center contact is connected directly to the 5v output of the Arduino. The left and right contacts are connected to digital input pins on the Arduino with a 10k pull down resistor to ground. The pull down

resistor sets the input on the digital pin to 0 when there is not a 5v (HIGH) input. When the left or right contact touches the tape it will trigger 1 HIGH and the autonomous code will know to change directions.

Dusty 2.0 uses a Flysky FS-TH9X 2.4ghz radio system that is commonly used in hobby grade RC aircraft. Having a 2.4ghz radio has the advantage of channel hopping. Meaning that if the radio detects another radio trying to use the same channel of the 2.4ghz band it will automatically switch to another one that is not in use. The radio and motor controller both have failsafe settings enabled so that if the connection becomes disconnected the robot will stop all operation, even in autonomous mode.

PROGRAMMING

EXPLANATION

Our robot uses a hybrid control method. It is passively controlled by an RF remote control and receiver. When the robot needs to enter an autonomous mode, a potentiometer is set to either a high-end value or a low-end value, instead of being mid-range. The RF receiver then relays the value to an analog input pin on the Arduino Uno. Based on whether the signal is high or low, the Arduino triggers a line-following function or a block-unloading function.

The line-following function uses two pins to read digital values to the Arduino. A relay pin is set to be HIGH at the start of line-following. When a pin on the left or right is in contact with the metal tape at the same time as the relay pin, the Arduino reads the left or right pin as HIGH and adjusts the drivetrain appropriately. If the left pin reads HIGH, the robot steers left; if the right pin reads HIGH, the robot steers right. If neither pin is touching, the robot simply drives forward at a constant rate. Every so often, the robot stops to cycle the doors and clear any blocks that are in the way. The program continues to listen for the value of the potentiometer to ensure that the robot can easily and reliably exit autonomous mode and return to manual control.

The unloading function opens the doors of the robot, drops the pan of the robot, and finally cycles a continuous rotation servo until all blocks have been successfully unloaded. At any time, the operator may exit this autonomous unloading function by returning the potentiometer on the RF remote to a mid-range position.

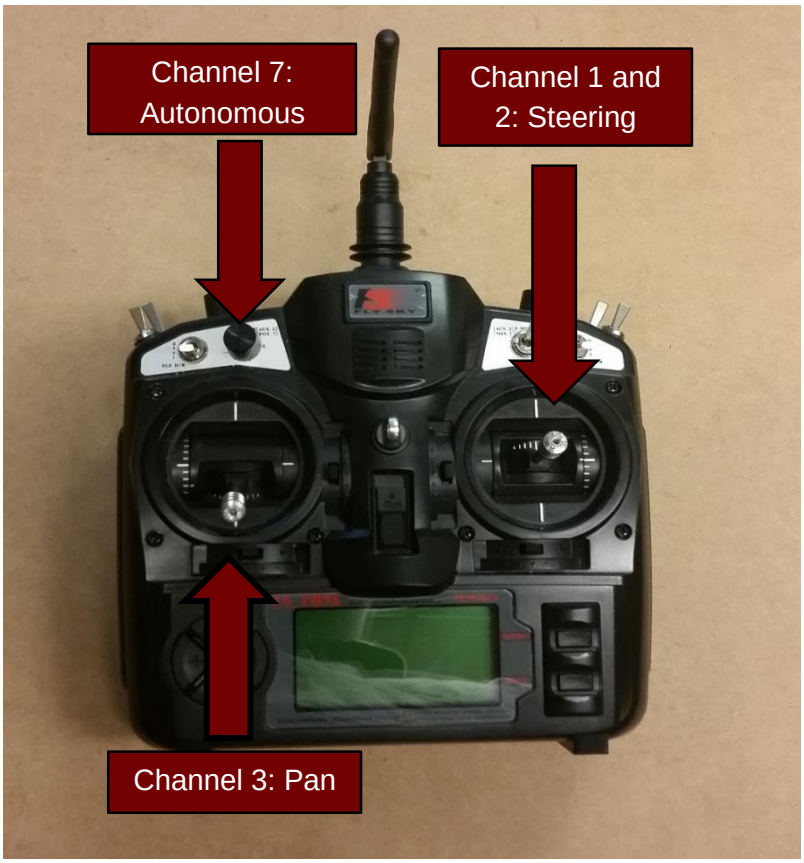
CODE:

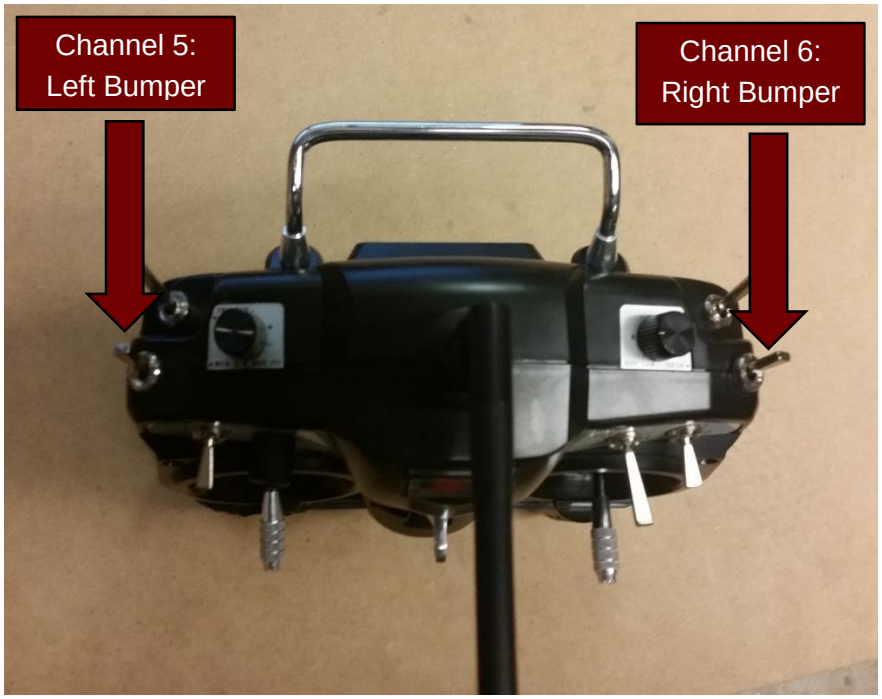
```
#include <Servo.h>


//-servos
Servo turning;
Servo drive;
Servo doorLeft;
Servo doorRight;
Servo dustPan;
Servo unload;
//end servos

//-pins
const int PIN_TURNING = 5;
const int PIN_DRIVE = 3;
const int PIN_AUTO = A0;
const int PIN_RELAY = 2;
const int PIN_RIGHT = 13;
const int PIN_LEFT = 12;
const int PIN_DOOR_L = 9;
const int PIN_DOOR_R = 10;
const int PIN_PAN = 6;
const int PIN_UNLOAD = 11;
//end pins

//-constants
const int DRIVE_LEFT = 1575;
const int NO_STEER = 1500;
const int DRIVE_RIGHT = 1425;

const int DRIVE_FWD = 1425;
const int DRIVE_BCK = 1650;
const int DRIVE_STOP = 1500;

const int DOOR_L_OPEN = 1200; //needs testing
const int DOOR_R_OPEN = 0; //needs testing
const int DOOR_L_CLOSED = 0; //needs testing
const int DOOR_R_CLOSED = 1200; //needs testing

const int PAN_DOWN = 0; //needs testing
const int PAN_UP = 0; //needs testing

const int THRESH_LINE = 1200; //needs testing
const int THRESH_UNLOAD = 200; //needs testing

const int UNLOAD_SPEED = 120; //needs testing
//end constants

//-variables
int rightState = 0, leftState = 0;
int autoVal = 0, turningVal = 0, driveVal = 0;
```

```
//end variables

//-methods
void openDoors() {
  analogWrite(PIN_DOOR_R, DOOR_R_OPEN);
  delay(100);
  analogWrite(PIN_DOOR_L, DOOR_L_OPEN);
}

void closeDoors() {
  analogWrite(PIN_DOOR_L, DOOR_L_CLOSED);
  delay(100);
  analogWrite(PIN_DOOR_R, DOOR_R_CLOSED);
}

void dropPan() {
  dustPan.write(PAN_DOWN);
}

void liftPan() {
  dustPan.write(PAN_UP);
}
//end methods

void setup() {
        pinMode(PIN_AUTO, INPUT);
        pinMode(PIN_RELAY, OUTPUT);

        pinMode(PIN_RIGHT, INPUT);
        pinMode(PIN_LEFT, INPUT);

        turning.attach(PIN_TURNING);
        drive.attach(PIN_DRIVE);
        doorLeft.attach(PIN_DOOR_L);
        doorRight.attach(PIN_DOOR_R);
        unload.attach(PIN_UNLOAD);
        dustPan.attach(PIN_PAN);

        Serial.begin(9600);  // begin serial communication
}

void loop() {
        autoVal = pulseIn(PIN_AUTO, HIGH);  // read autoVal from pin

        if (autoVal > THRESH_LINE) {  // line-following autonomous
   openDoors();

   dropPan();

   int count = 0;
   while (autoVal > THRESH_LINE) {  //-follow line
     if (count == 100) {
       closeDoors();
       delay(100);
```

```
      openDoors();
      count = 0;
    }

    autoVal = pulseIn(PIN_AUTO, HIGH);  // read autoVal from pin

                digitalWrite(PIN_RELAY, HIGH);
                rightState = digitalRead(PIN_RIGHT);
                leftState = digitalRead(PIN_LEFT);

                //Serial.print("Right Pin State: ");
                //Serial.print(rightState);
                //Serial.print(" Left Pin State: ");
                //Serial.println(leftState);

                if(rightState == HIGH & leftState == LOW) {
                        turning.write(DRIVE_RIGHT);
                        drive.write(DRIVE_STOP);
                }

                if(rightState == LOW & leftState == HIGH) {
                        turning.write(DRIVE_LEFT);
                        drive.write(DRIVE_STOP);
                }

                if(rightState == LOW & leftState == LOW) {
                        drive.write(DRIVE_FWD);
                        turning.write(NO_STEER);
                }

   count++;
  } //end follow line
        } else if (autoVal < THRESH_UNLOAD) {  // unloading autonomous
          openDoors();

  dropPan();

  while (autoVal < THRESH_UNLOAD) {
    autoVal = pulseIn(PIN_AUTO, HIGH);  // read autoVal from pin

    //-spin servo
    unload.write(UNLOAD_SPEED);
    //end spin servo
  }

  unload.write(0);
        } else  {
                digitalWrite(PIN_RELAY, LOW);
        }
}
```

CONTROLER LAYOUT





SIZING SPECIFICATIONS

SCREWS (TAPPED)

Component	Screw Size	# of Screws Used
Motor Face	M3 6mm	24
Parallel Bearing	#6 ½ in	4
Servo Horns	M3 6mm	3
Servo	M3 3mm	4
Servo Mount	#8-32 3/8 in	20
Set Screw Hub	#10-32 Set Screw	8
Set Screw Hub	#6-32 1/2 in	24
Motor Mounts	#8-32 3/8 in	8
18 Tooth ANSI #25 Sprocket	#8-32 Set Screw	8
Pan Lid	#8-32 3/8 in	4

SCREWS (NOT TAPPED)

Component	Screw Size	# of Screws Used
Servo Horns	#4-40 ½ in	4
Chassis	#6-32 ½ in	4
Lift Mechanism	1/4in 1 in bolts	2
Lift Mechanism	#6-32 ½ in	3
Perpendicular Bearings	#6-32 ½ in	8
Hinge	#6-32 ½ in	8
Spring	#6-32 ½ in	1
Chassis	#6-32 ½ in	2
Wheel Mount	#6-32 ½ in	16
Pan Lid	#6-32 ½ in	2

ALAN WRENCH SIZING CHART

Component	Alan Wrench Size
Servo Horn	1.5mm
Set Screw Hub	3/32 in
18 Tooth ANSI #25 Sprocket	2 mm

BILL OF MATERIALS

Component	Quantity	Price/piece
Chassis	1	15.78
Left 2in Angle	1	7.28
Right 2in Angle	1	7.28
Pan	1	14.58
Pan Rod	1	2.25
Front Wheel Shafts	2	0.42
Servo Mounts	4	.20
Left Bumper	1	4.58
Right Bumper	1	4.58
Middle Bumper	1	4.58
Left Cover	1	6.98
Right Cover	1	8.57
Wheel Mounts	4	4.26
Lift Mechanism 1	1	.87
Lift Mechanism 2	1	.72
Lift Mechanism 3	1	1.02
Pan Lid	1	10.58
Motor Mounts	2	3.35
Parallel Bearings	2	5.99
Perpendicular Bearings	4	5.99
Set Screw Hubs	8	4.99
ANSI #25 18 Tooth Sprockets	4	4.81
2ft ANSI #25 Chain	1	2.78
ANSI Master Link	2	.42
RC Off Road Buggy Tires	4	4.82
Metal Servo horn	4	4.06
MG946R Metal Gear 13KG Torque Digital Servo	3	11.80
Continuous Rotation Servo	1	12.57
12V 170 oz-in Metal Gearmotor 200 RPM	4	24.95
Relay	3	5.25

Sabertooth Dual 12A 6V-24V R/c Regenerative Motor Driver	1	64.99
Arduino Uno	1	24.95
Adafruit Proto Shield for Arduino Kit	1	19.39
Assorted Wires	61	0.05
FlySky FS-i6-M2 2.4Ghz 6-Channel Transmitter/Receiver	1	52.99
M3 6mm Screws	27	.04
M3 3mm Screws	4	.04
#4-40 ½ in Screws	4	.05
#6 ½ in Screws	72	0.08
#8-32 3/8 in Screws	28	0.09
#8-32 Set Screw	8	0.09
#10-32 Set Screw	8	0.10
¼ in Bolt	2	.25
Total		**$589.9**

BIBLIOGRAPHY

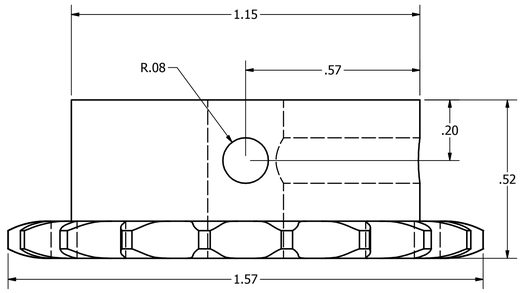
Materials for Robot Building: An Introduction. (n.d.). Retrieved from Robotics Universe: http://www.robotoid.com/howto/materials-for-robot-building-an-introduction.html

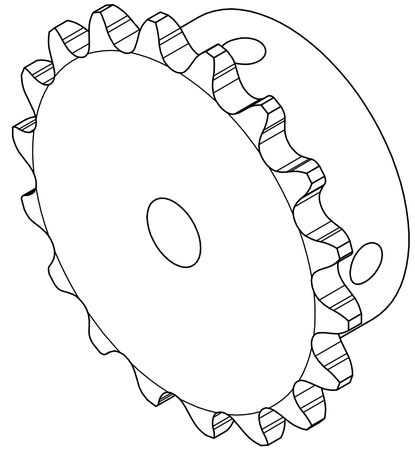
RC Basics. (n.d.). Retrieved from SuperDroid Robots: http://www.superdroidrobots.com/shop/custom.aspx/remote-control-rc-support/40/

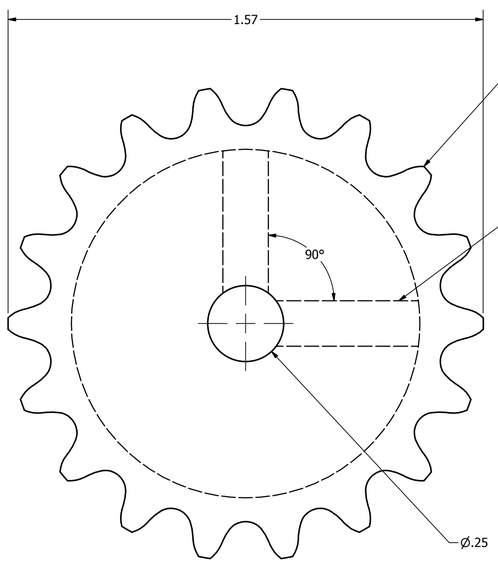
Sheet Metal Bending. (n.d.). Retrieved from The Library of Manufacturing: http://thelibraryofmanufacturing.com/sheetmetal_bending.html

White, G. (2015, March 27). *Indusrty Analysis: The Pros and Cons of 3d Printing*. Retrieved from Manufacturing Global: http://www.manufacturingglobal.com/technology/408/INDUSTRY-ANALYSIS:-The-pros-and-cons-of-3d-printing

TECHNICAL DRAWINGS

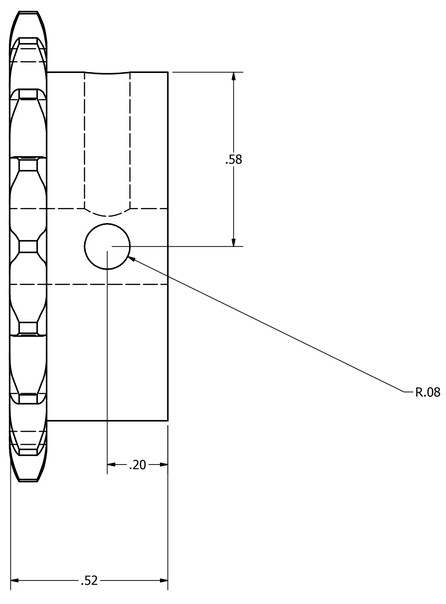




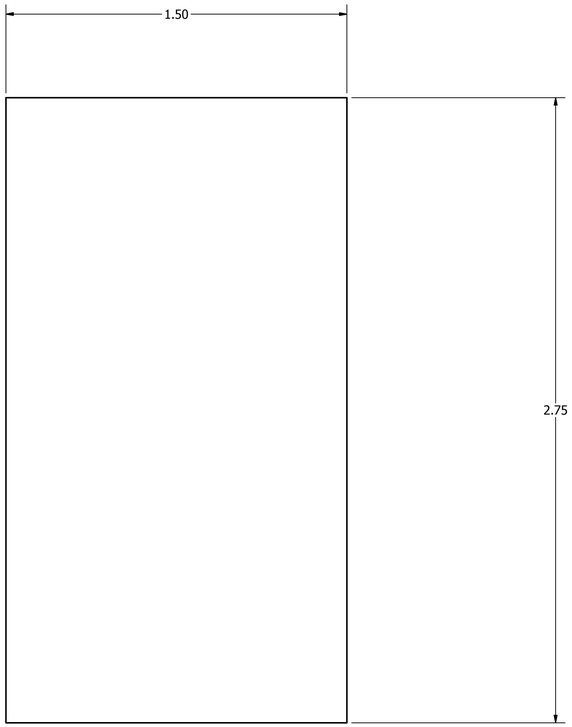


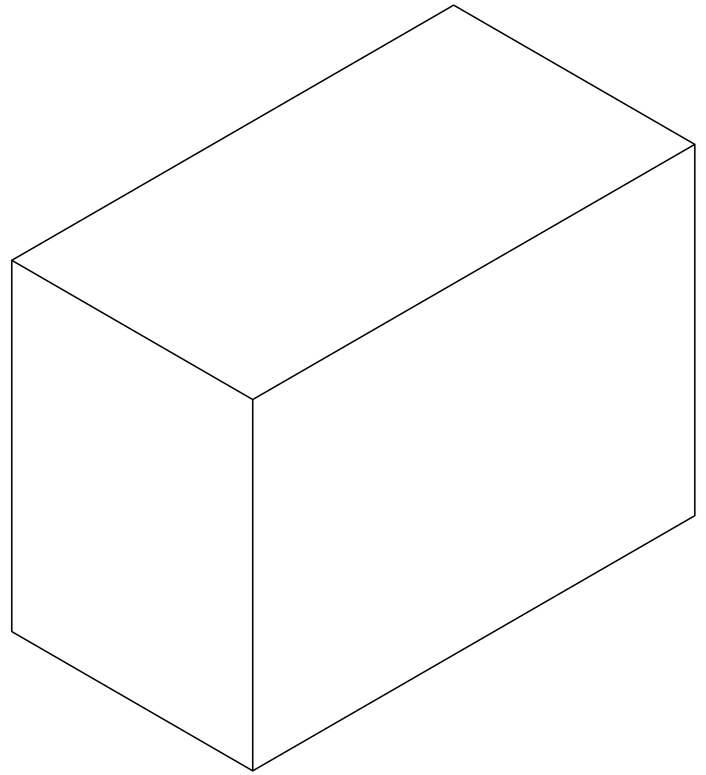
ANSI #25 sprocket pattern

Threaded for #8-32 Set Screw

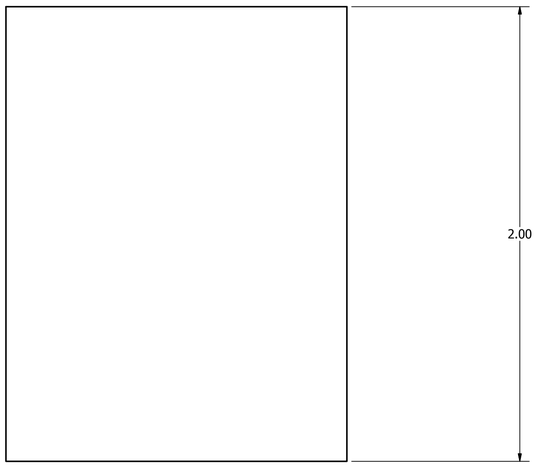


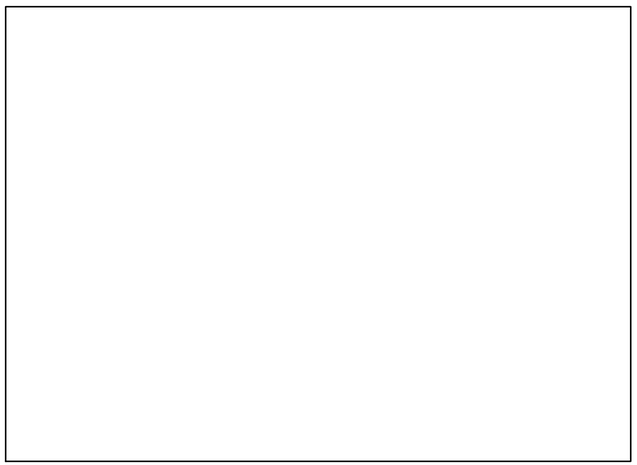
DRAWN siu853102150	10/1/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO ANSI #25 18 Tooth Sprocket REV
		SCALE 4 : 1	SHEET 1 OF 1



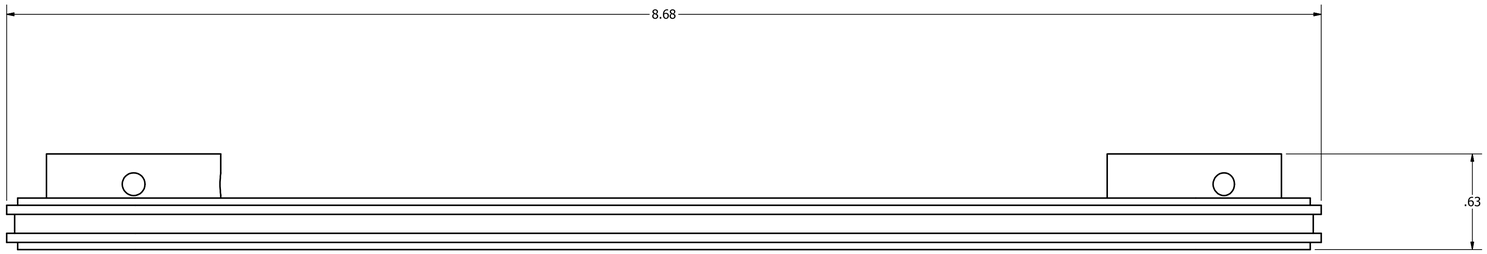


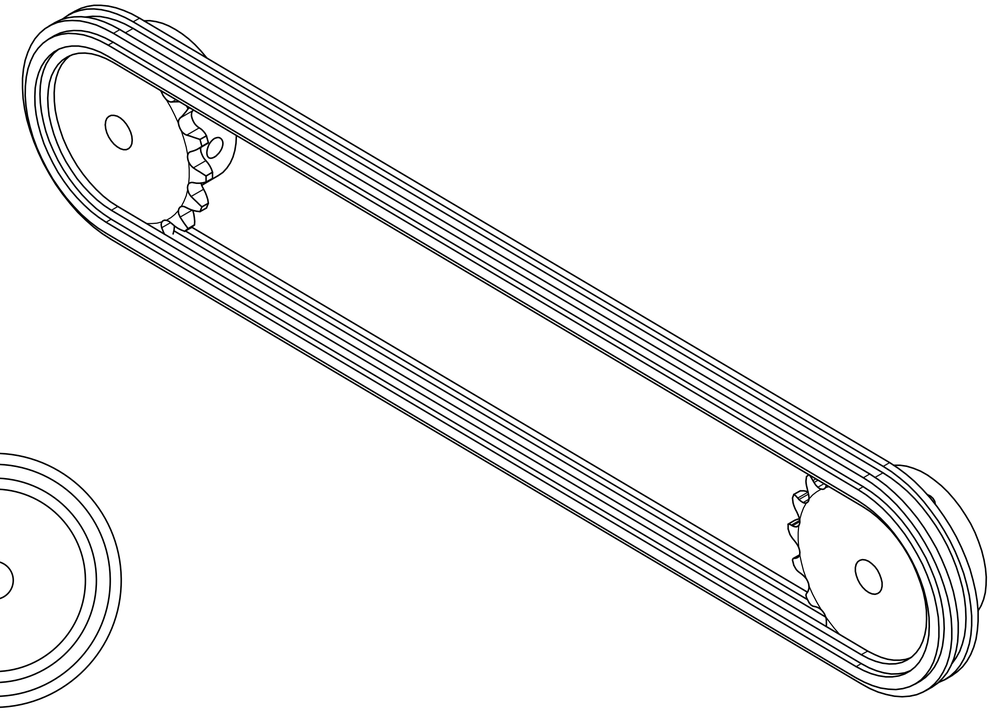
Arduino and Shield will be attached using Velcro

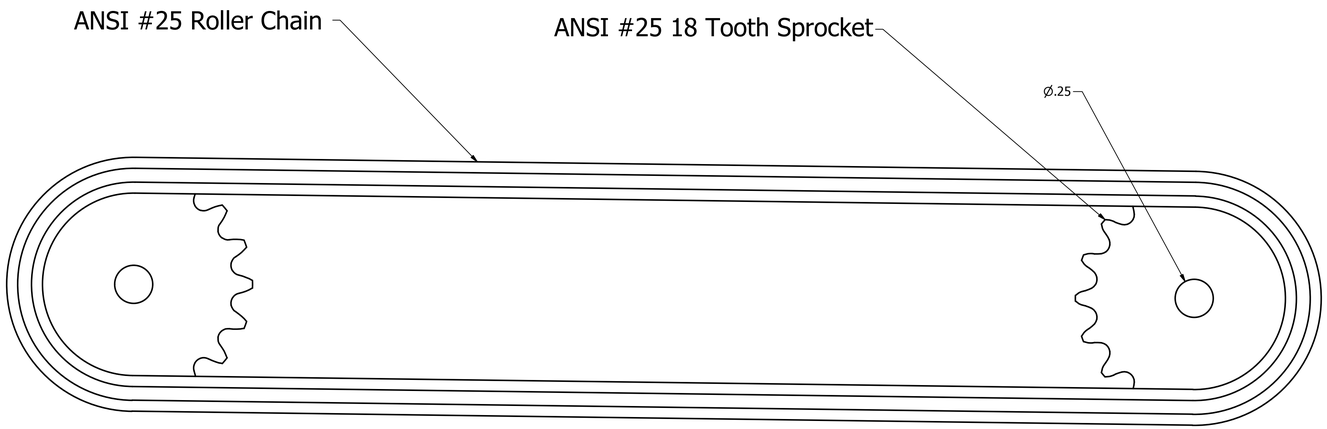




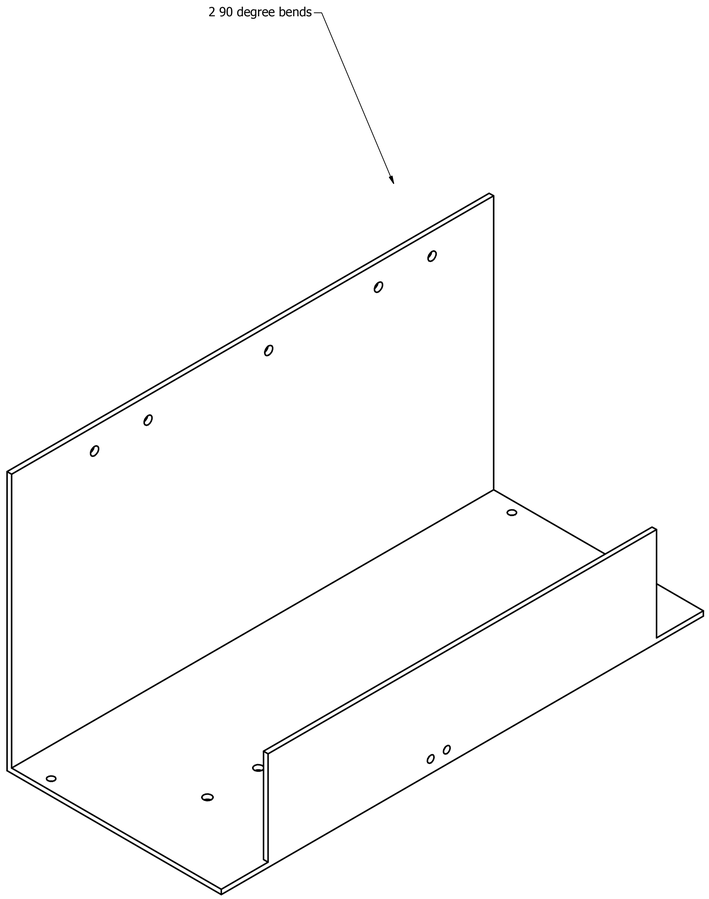
DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Arduino and Shield / REV
		SCALE 3 : 1	SHEET 1 OF 1

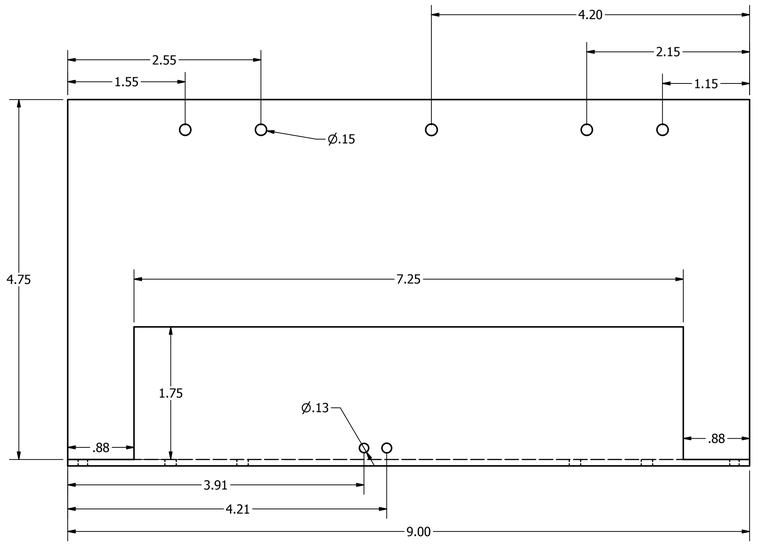


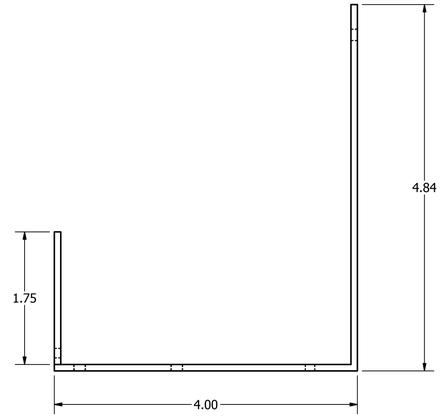


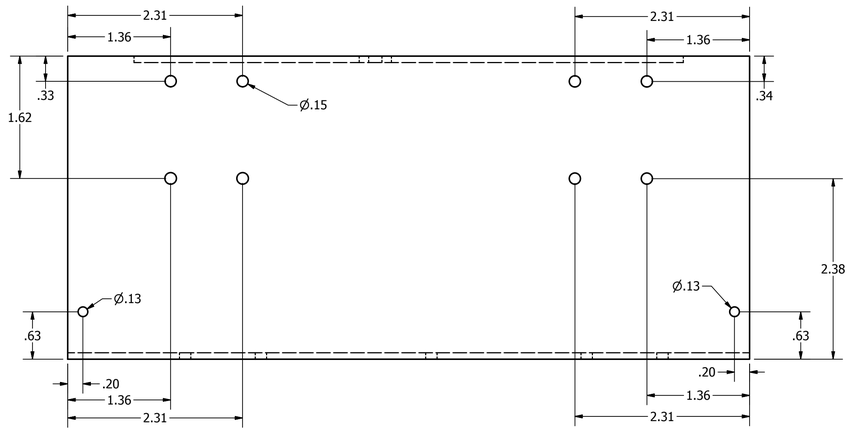


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Chain Assembly REV
		SCALE 2 : 1	SHEET 1 OF 1

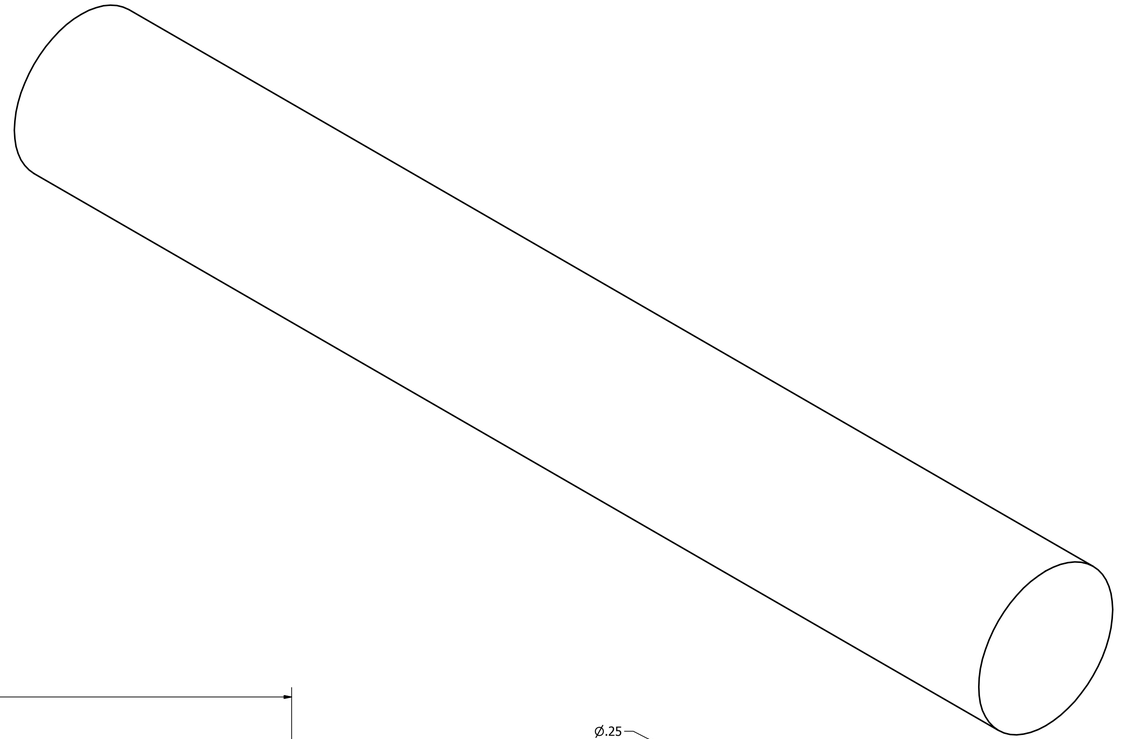


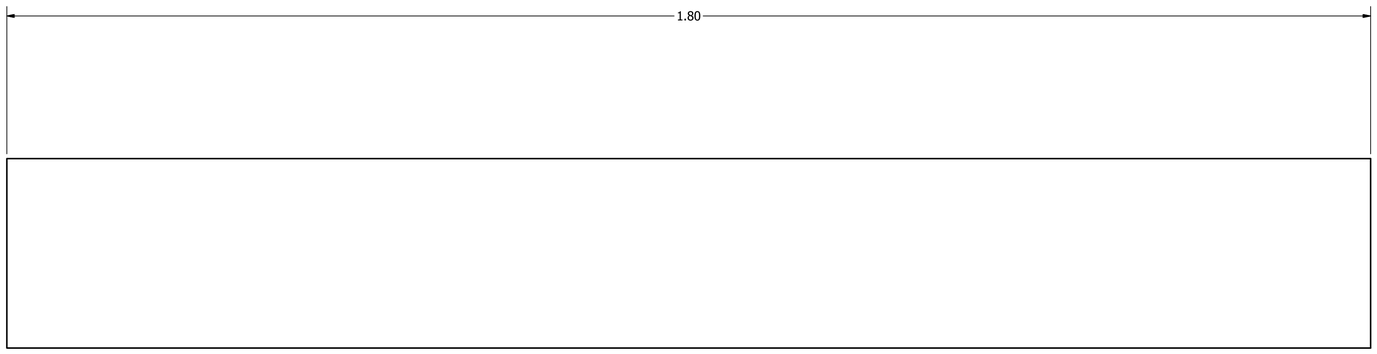


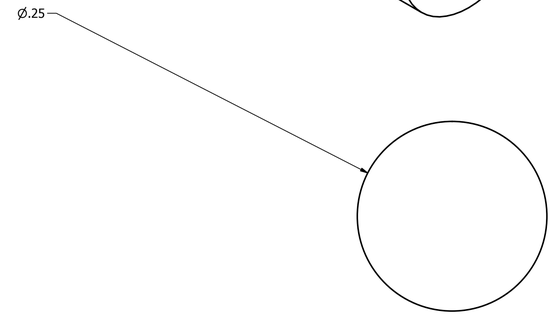




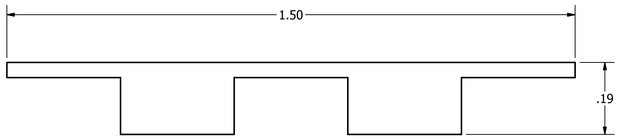
DRAWN siu853102150	9/30/2016	
CHECKED		TITLE
QA		
MFG		
APPROVED		SIZE D / DWG NO Chasis / REV
		SCALE 1 : 1 / SHEET 1 OF 1

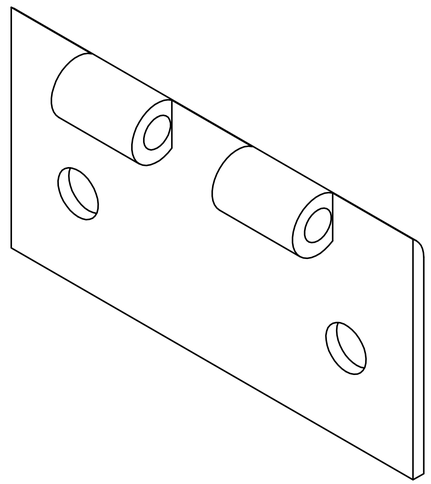


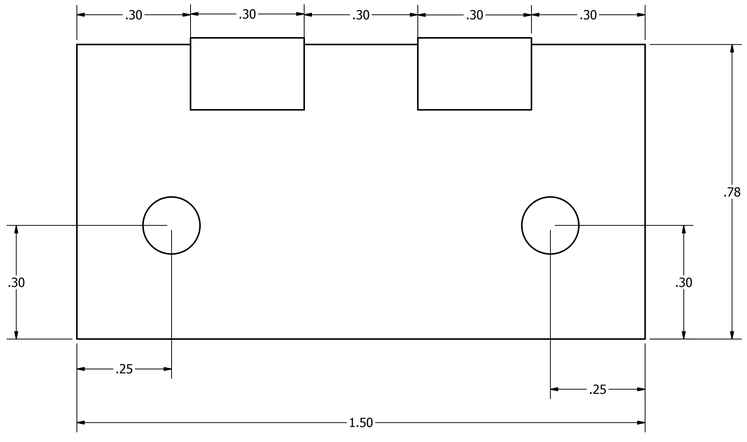


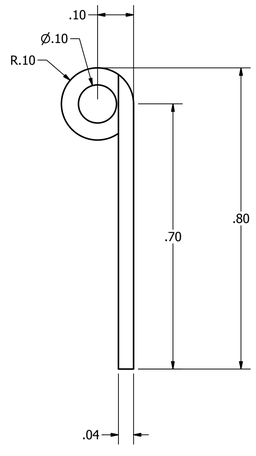


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Front Wheel Shafts
		SCALE 10 : 1	SHEET 1 OF 1

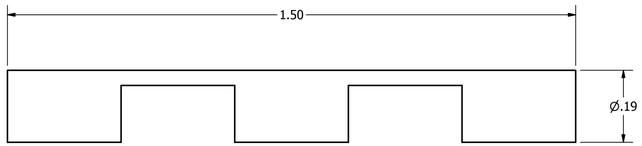


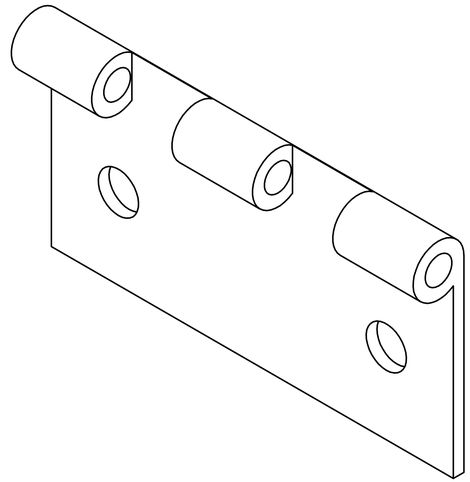


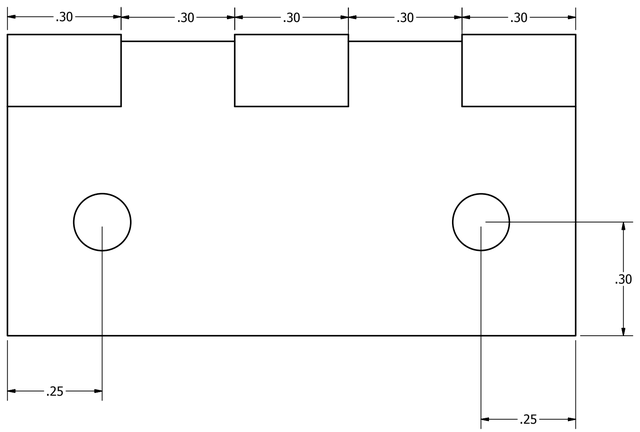


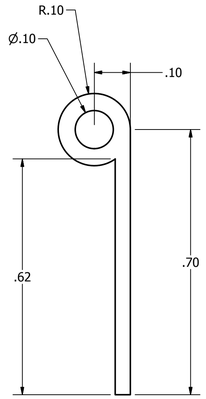


DRAWN siu853102150	10/6/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Hinge part 1 / REV
		SCALE 5 : 1	SHEET 1 OF 1

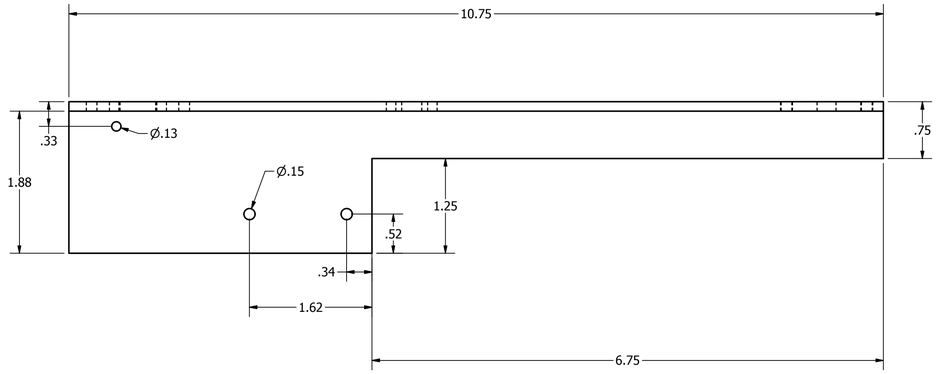


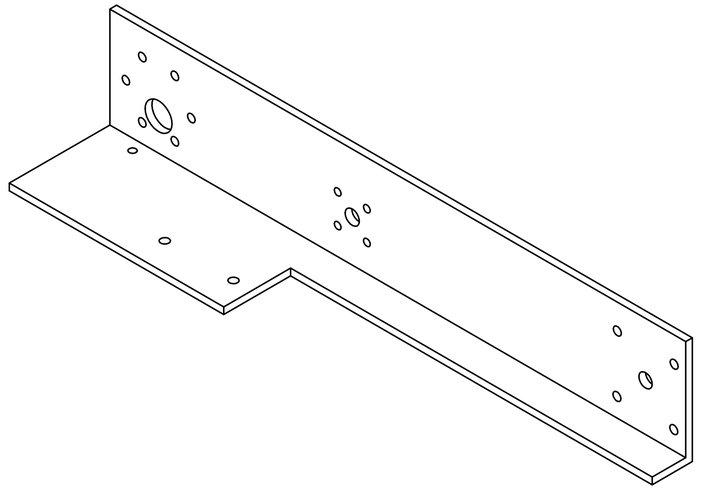


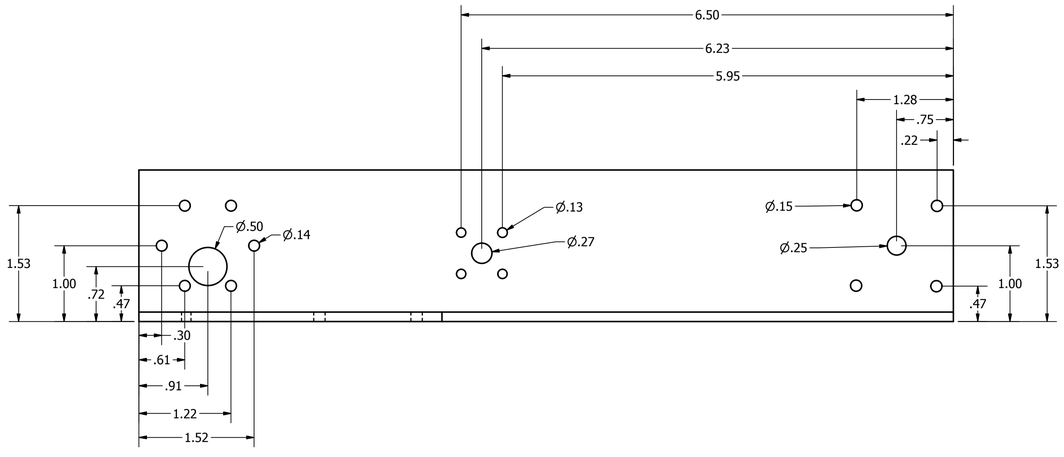


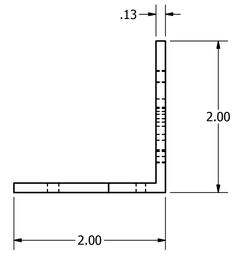


DRAWN siu853102150	10/6/2016			
CHECKED		TITLE		
QA				
MFG				
APPROVED				
		SIZE D	DWG NO Hinge part 2	REV
		SCALE 5 : 1	SHEET 1 OF 1	

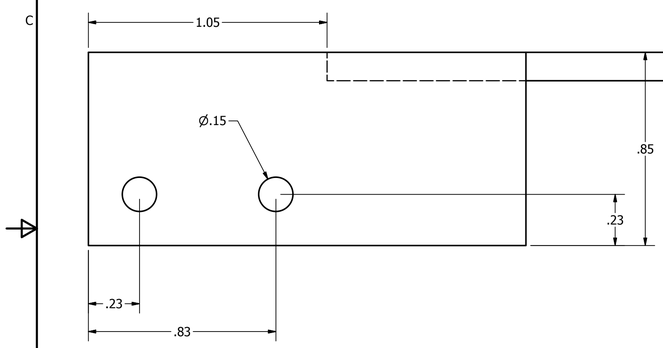


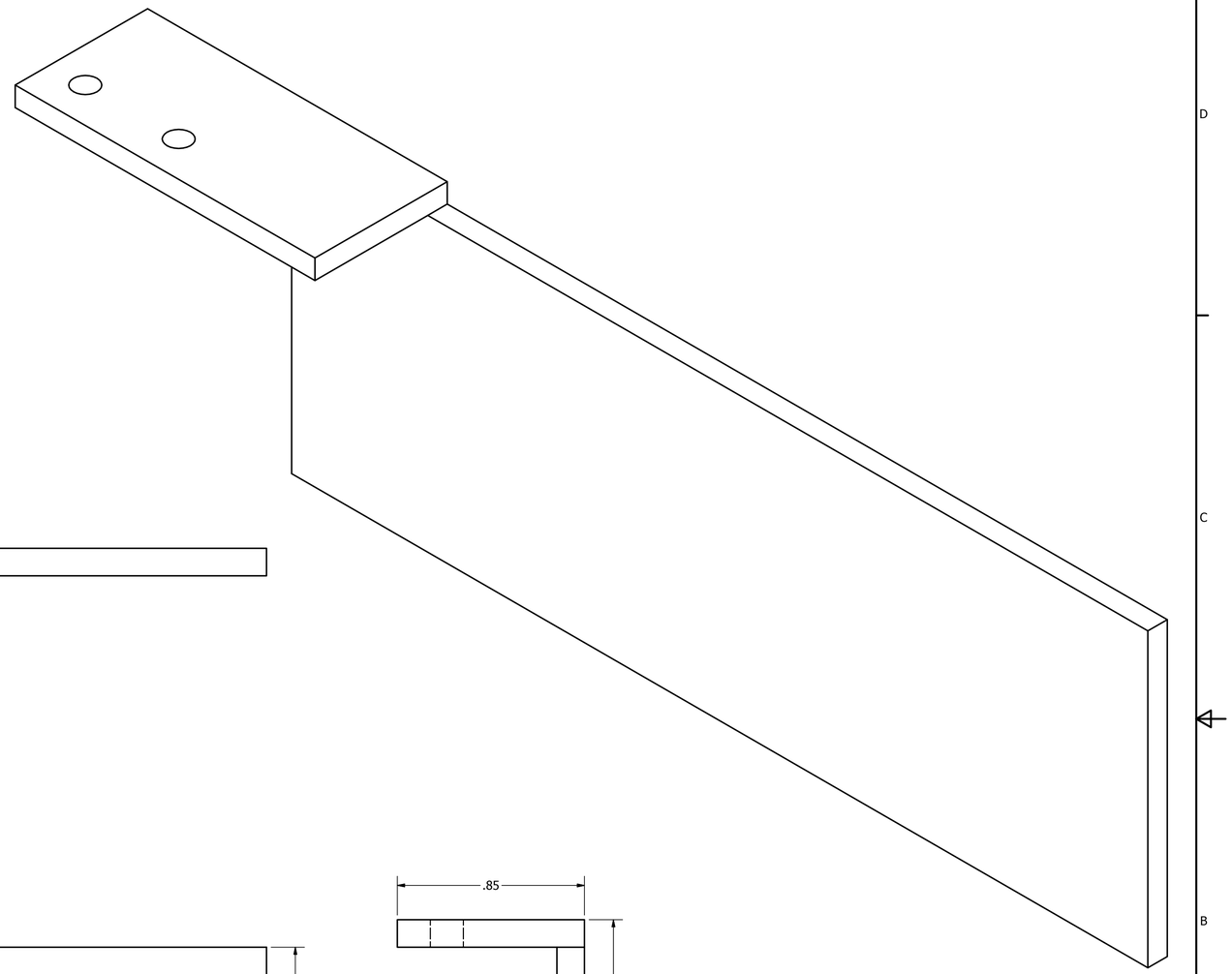


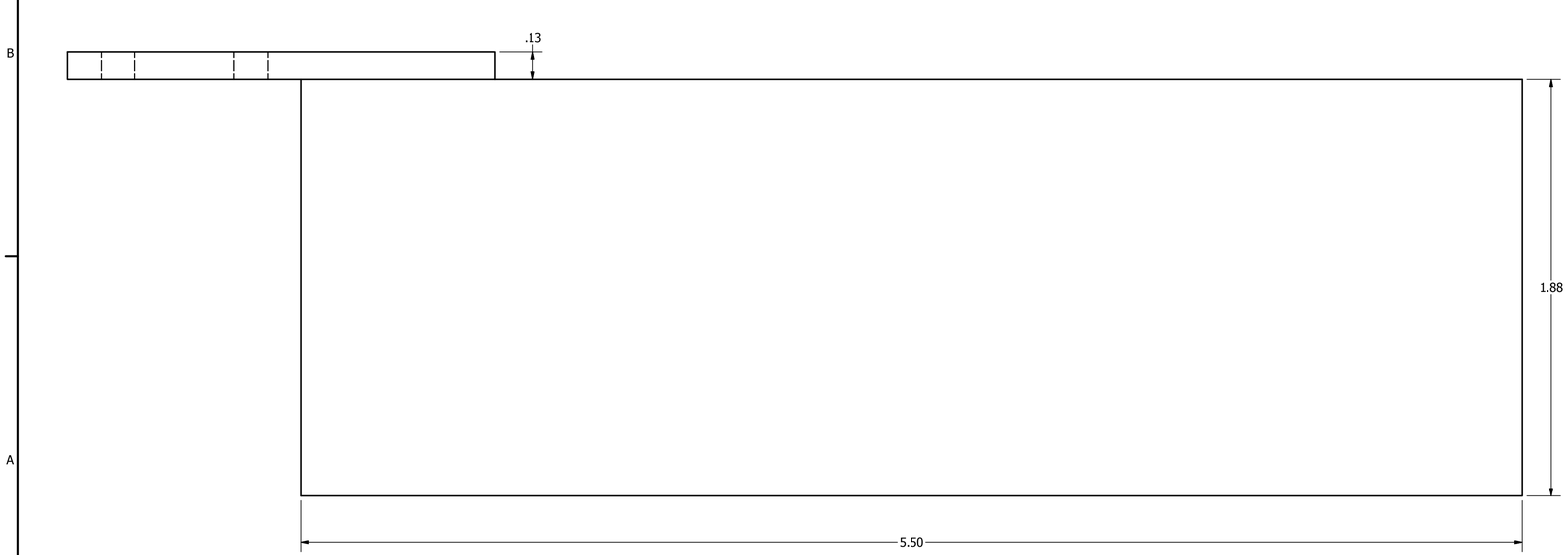


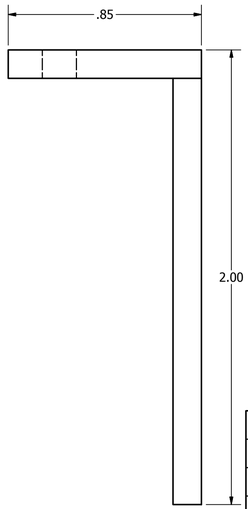


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Left 2in Angle / REV
		SCALE 1 : 1	SHEET 1 OF 1

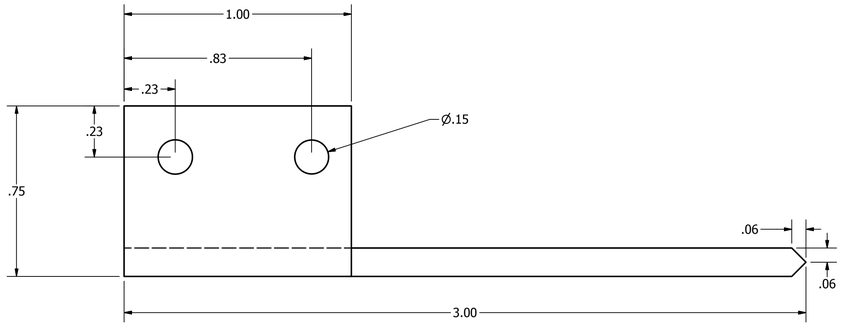


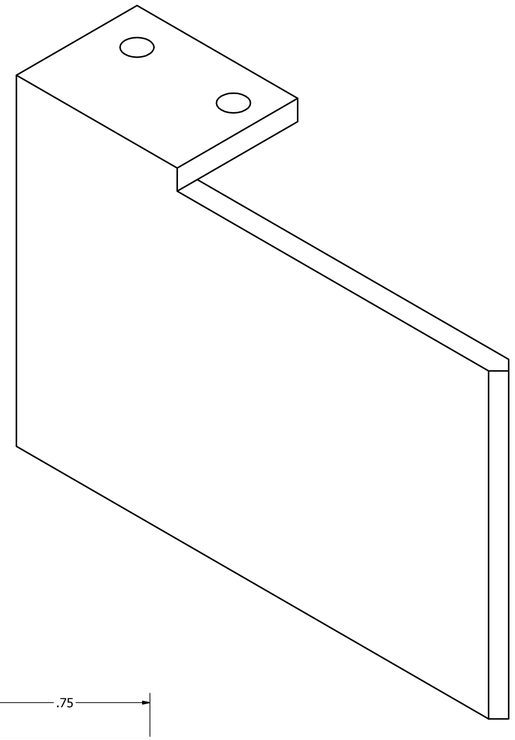


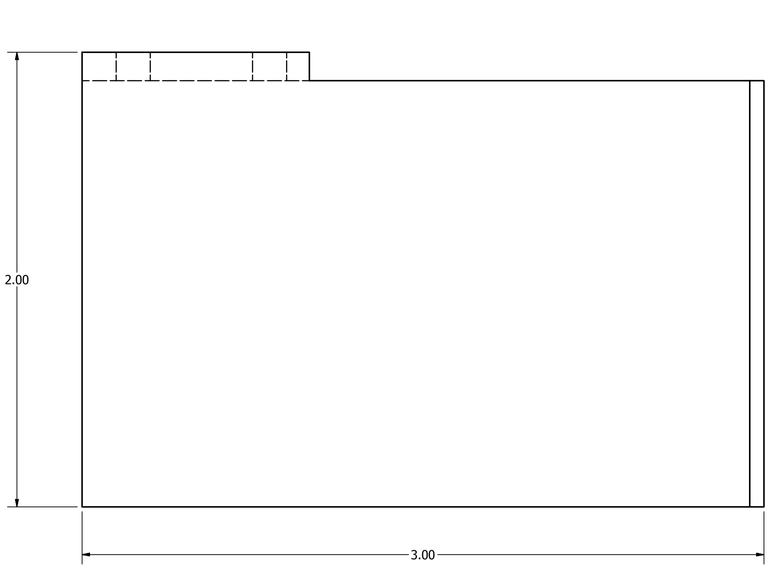


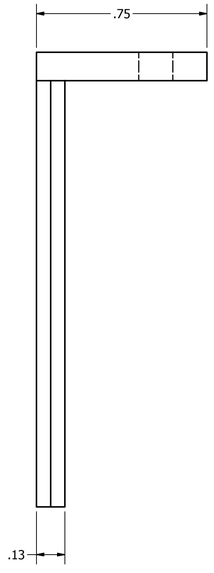


DRAWN siu853102150	10/5/2016	
CHECKED		TITLE
QA		
MFG		
APPROVED		
		SIZE D · DWG NO Left Bumper · REV
		SCALE 3 : 1 · SHEET 1 OF 1

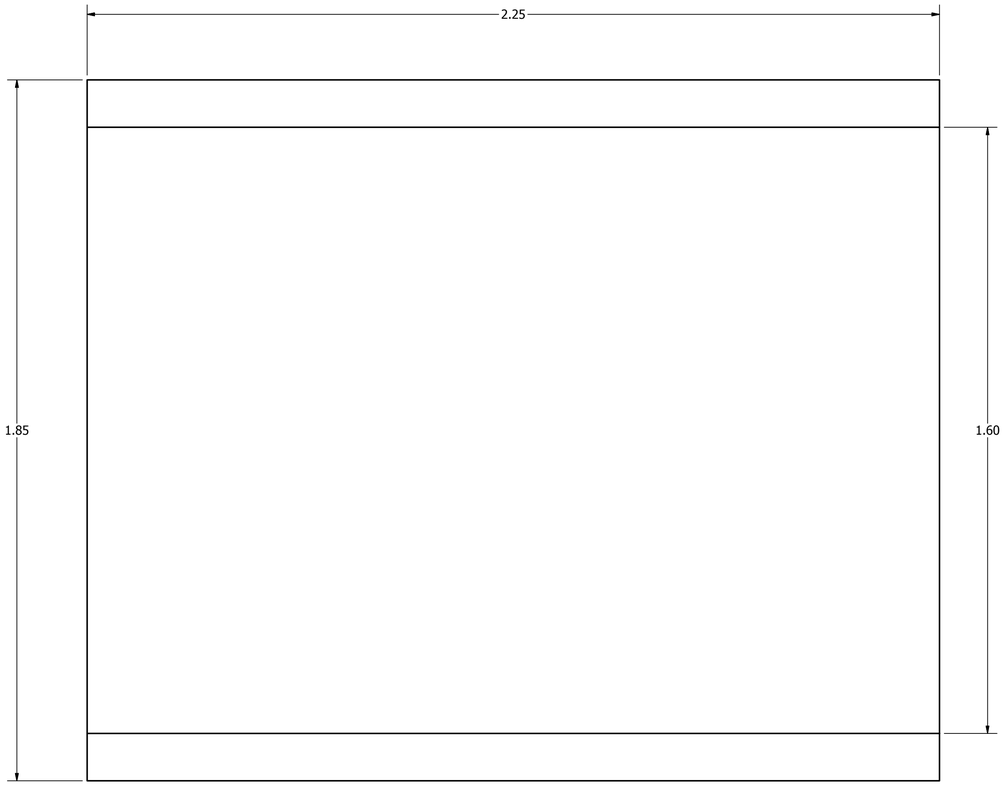


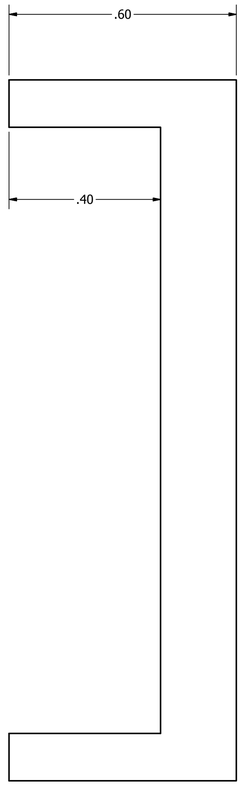


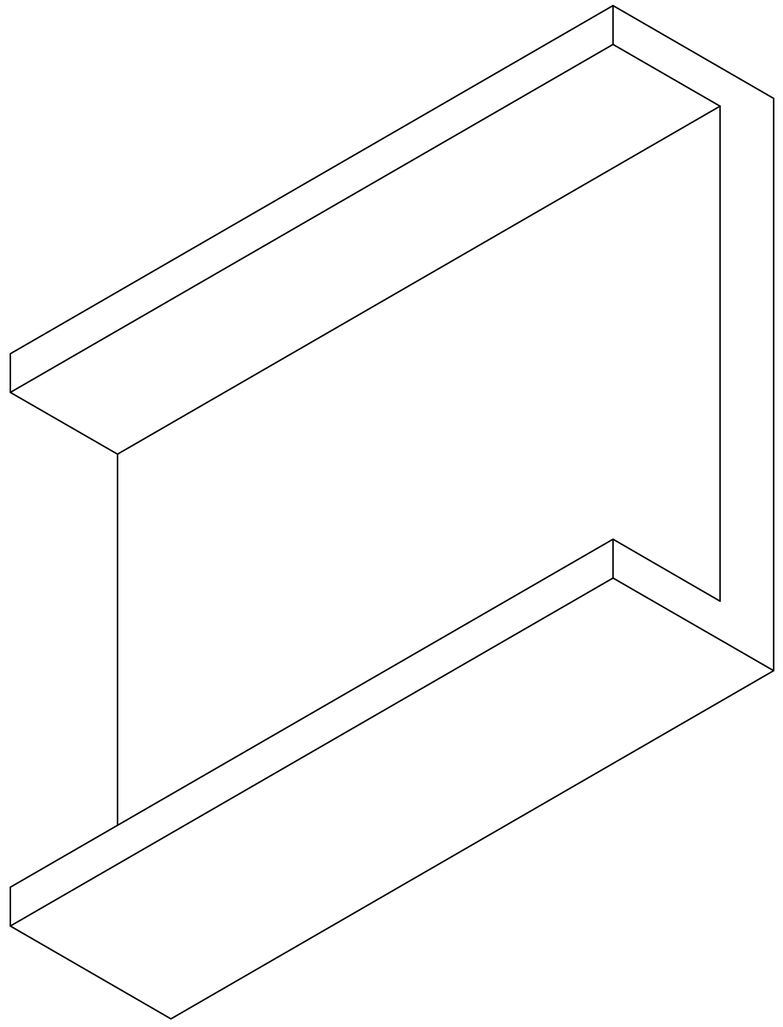


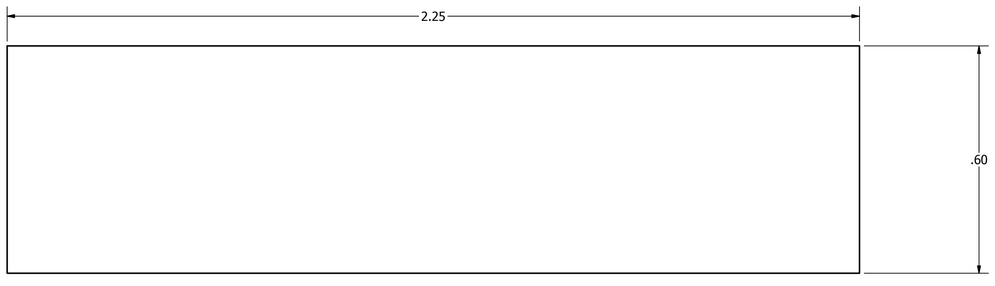


DRAWN siu853102150	10/5/2016	
CHECKED		TITLE
QA		
MFG		
APPROVED		SIZE D / DWG NO Middle Bumper / REV
		SCALE 3 : 1 / SHEET 1 OF 1



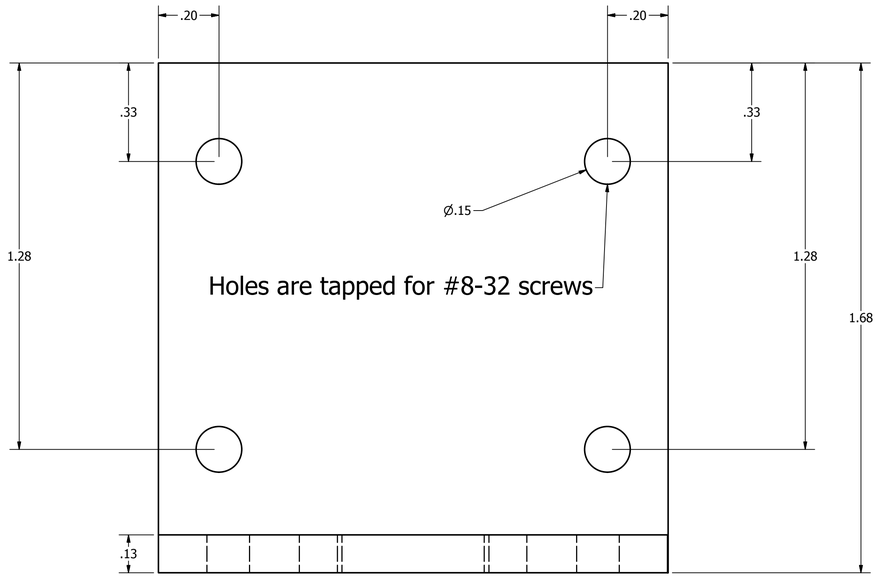


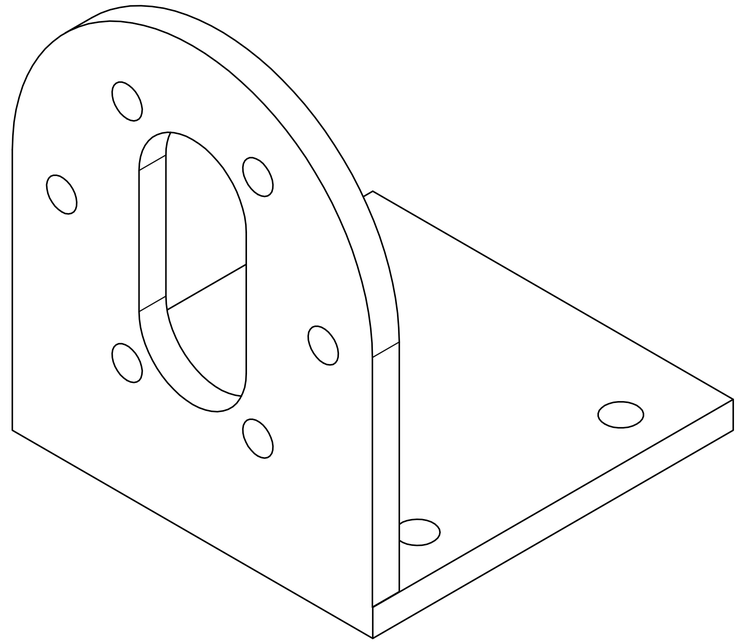


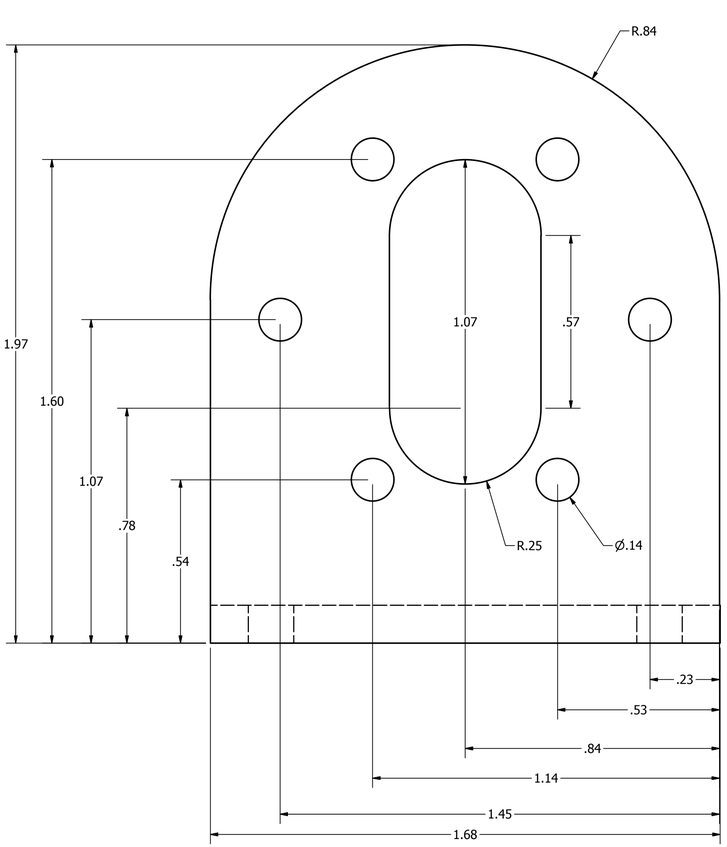


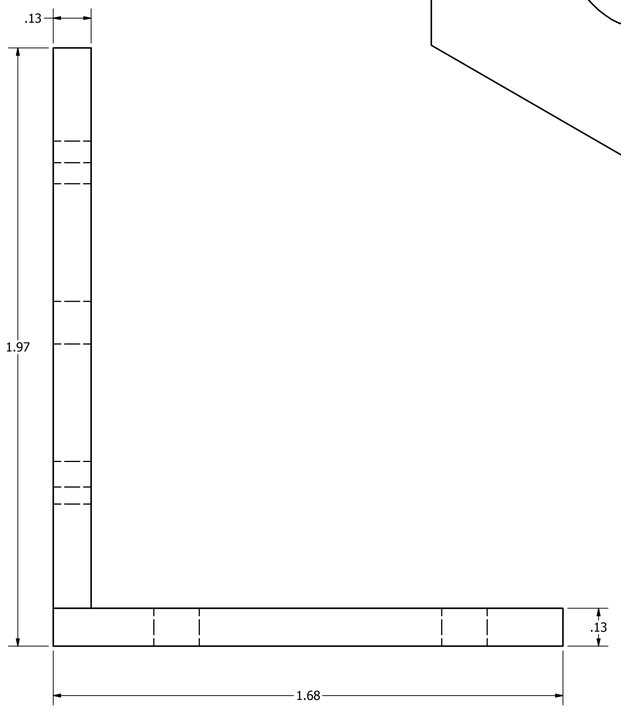
Motor Controller will be attached with Velcro

DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Motor Controller REV
		SCALE 5 : 1	SHEET 1 OF 1



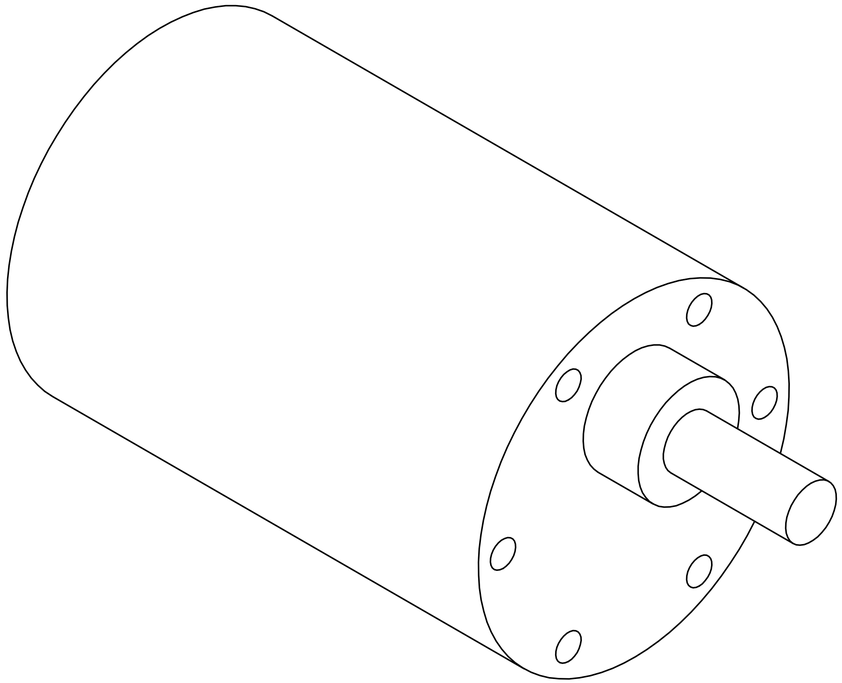


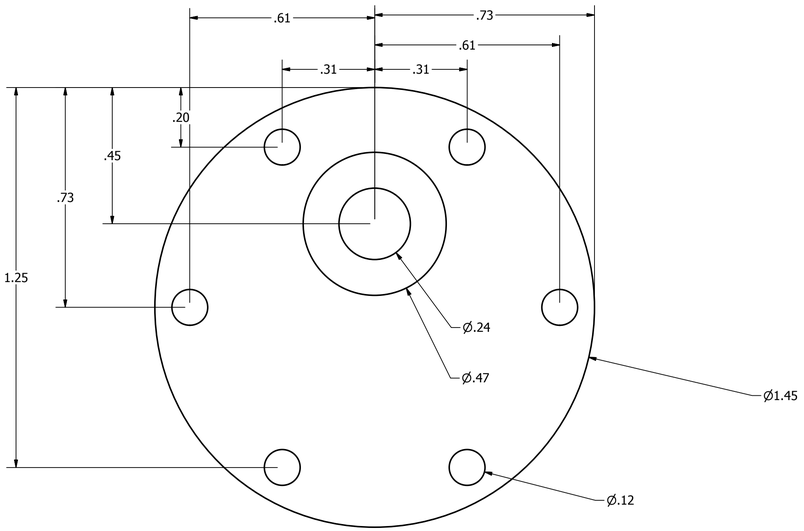


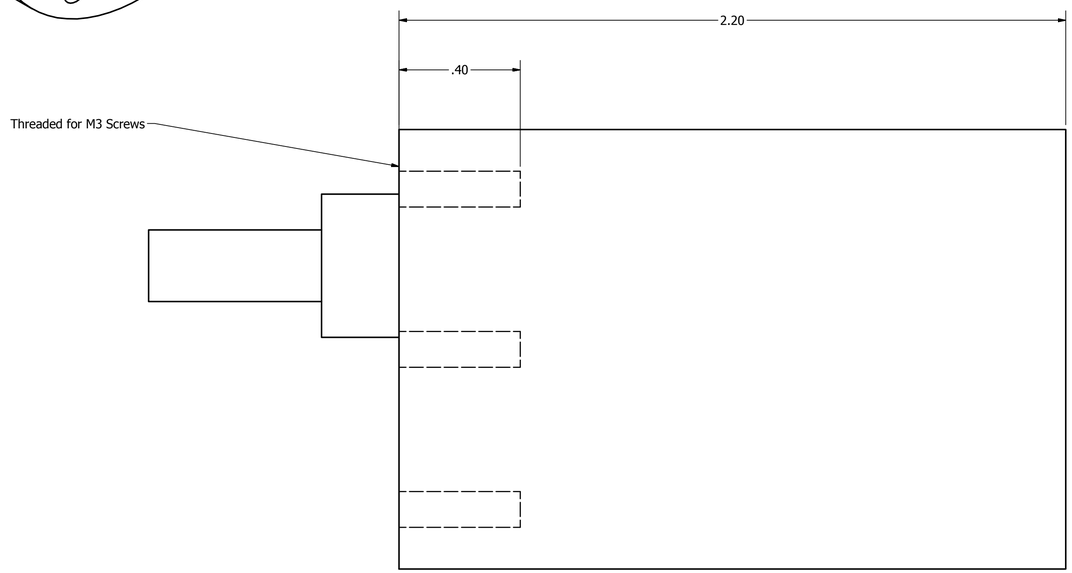


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED			
		SIZE D	DWG NO Motor Mounts
		SCALE 4 : 1	SHEET 1 OF 1

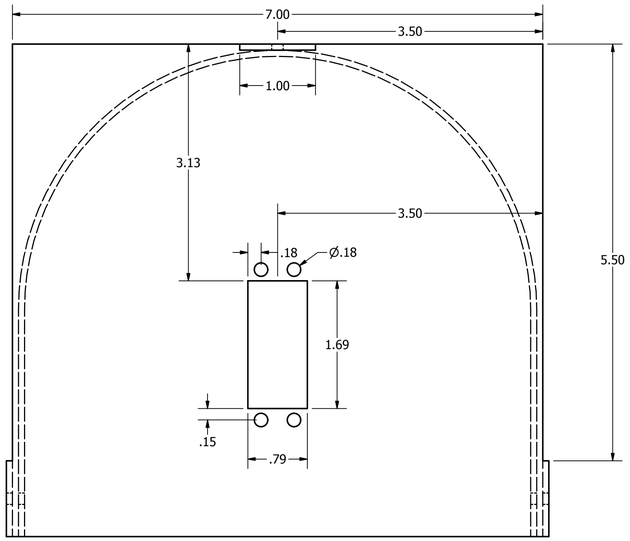
50:1 Gearmotor, 200 RPM, 170 oz-in torque

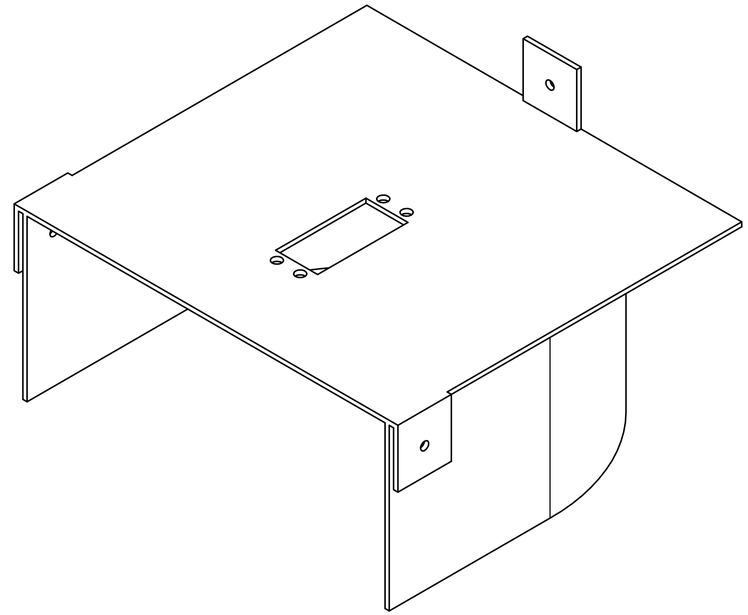


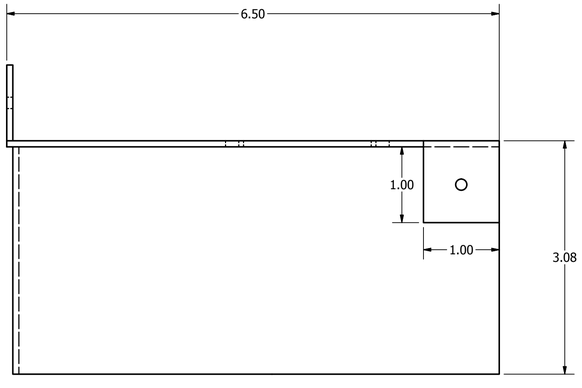


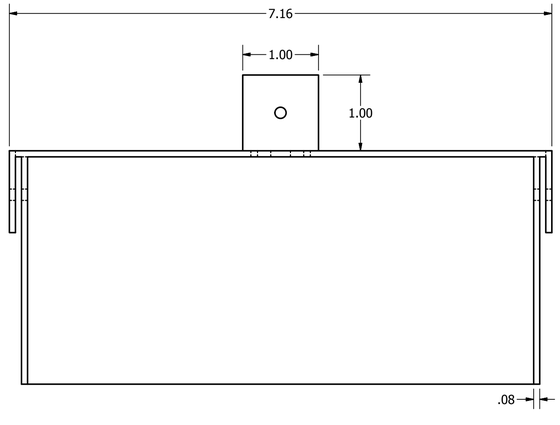


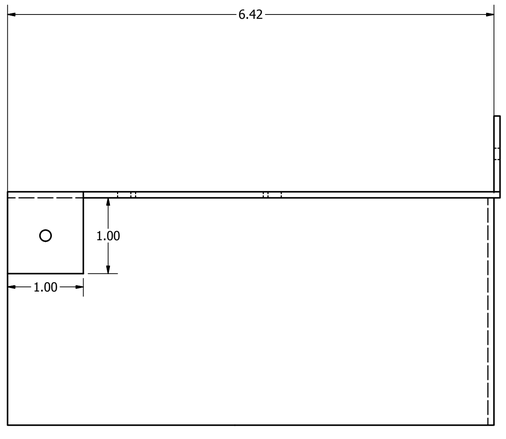
DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED			
		SIZE D	DWG NO Motor
		SCALE 4 : 1	SHEET 1 OF 1



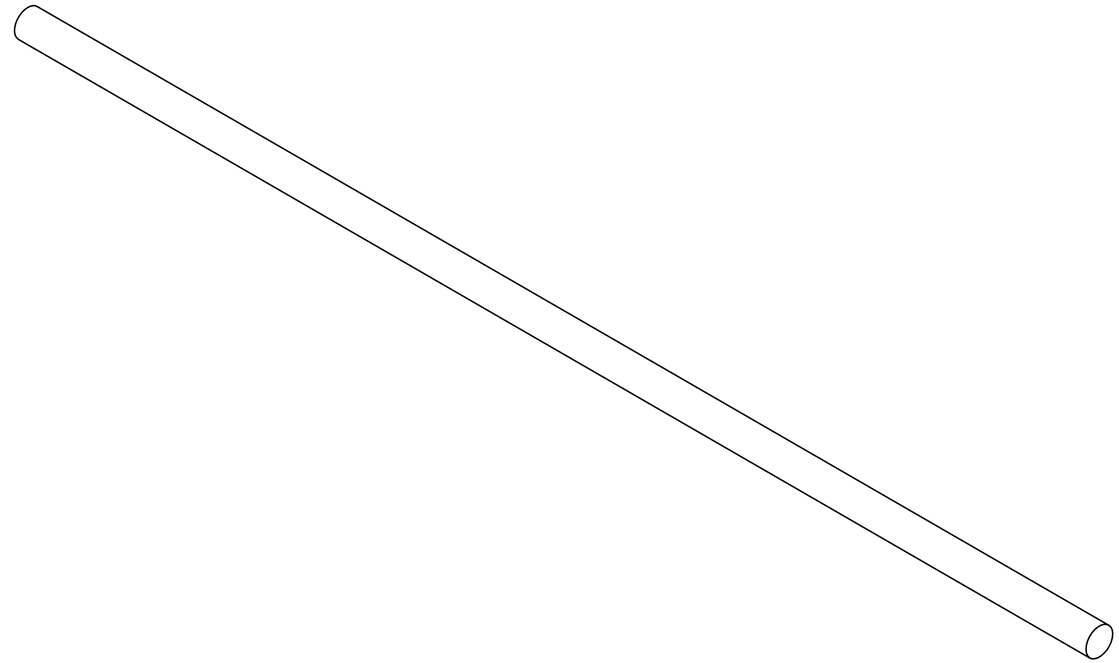


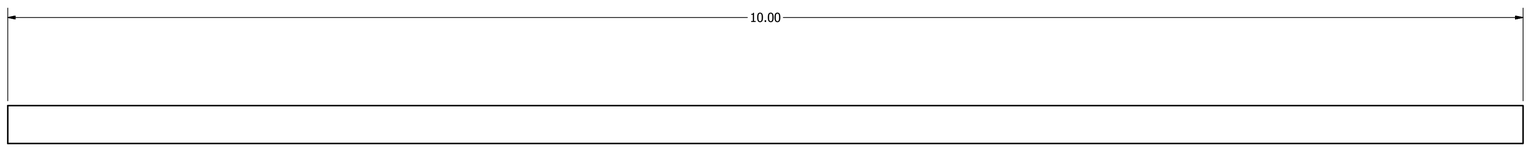


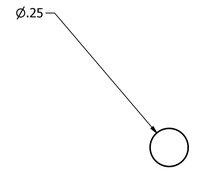




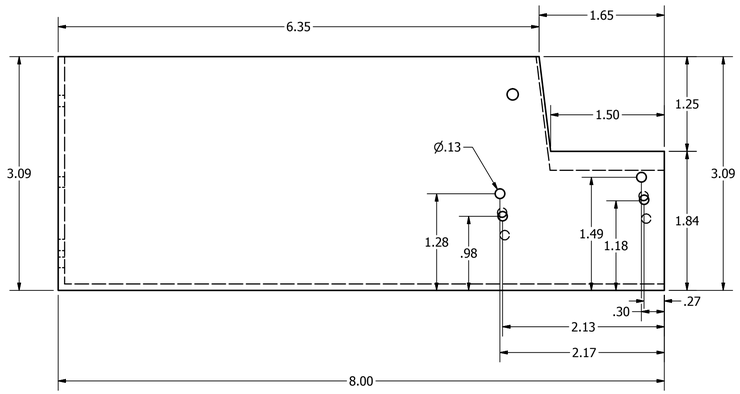
DRAWN siu853102150	10/5/2016	
CHECKED		TITLE
QA		
MFG		
APPROVED		SIZE D / DWG NO Pan Lid / REV
		SCALE 1 : 1 / SHEET 1 OF 1

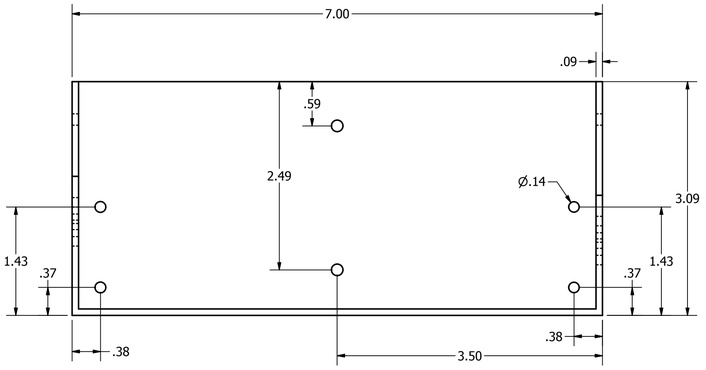


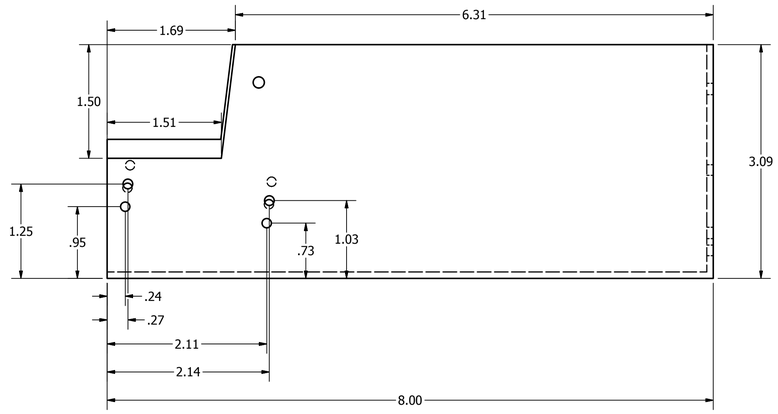


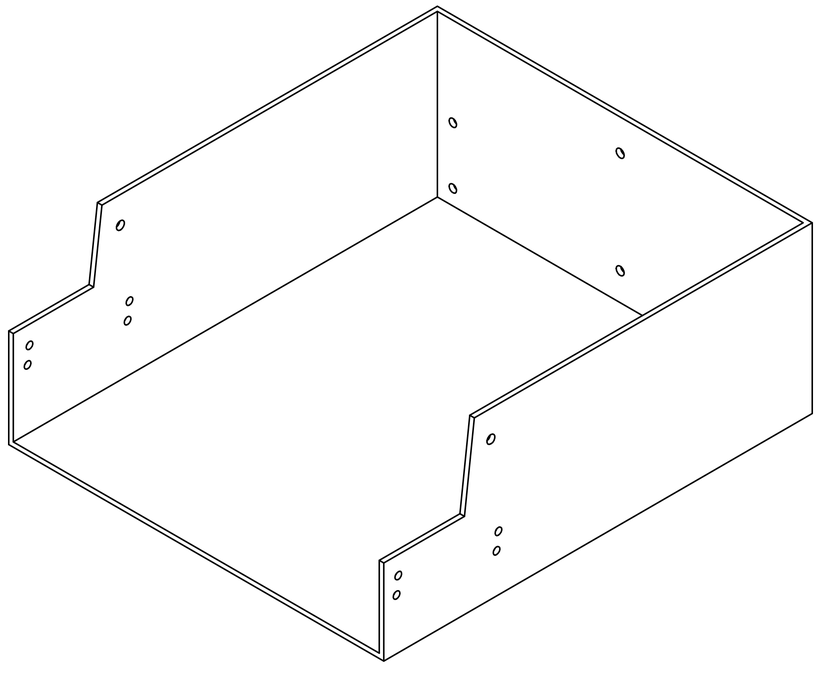


DRAWN siu853102150	9/30/2016			
CHECKED		TITLE		
QA				
MFG				
APPROVED		SIZE D	DWG NO Pan Rod	REV
		SCALE 2 : 1	SHEET 1 OF 1	

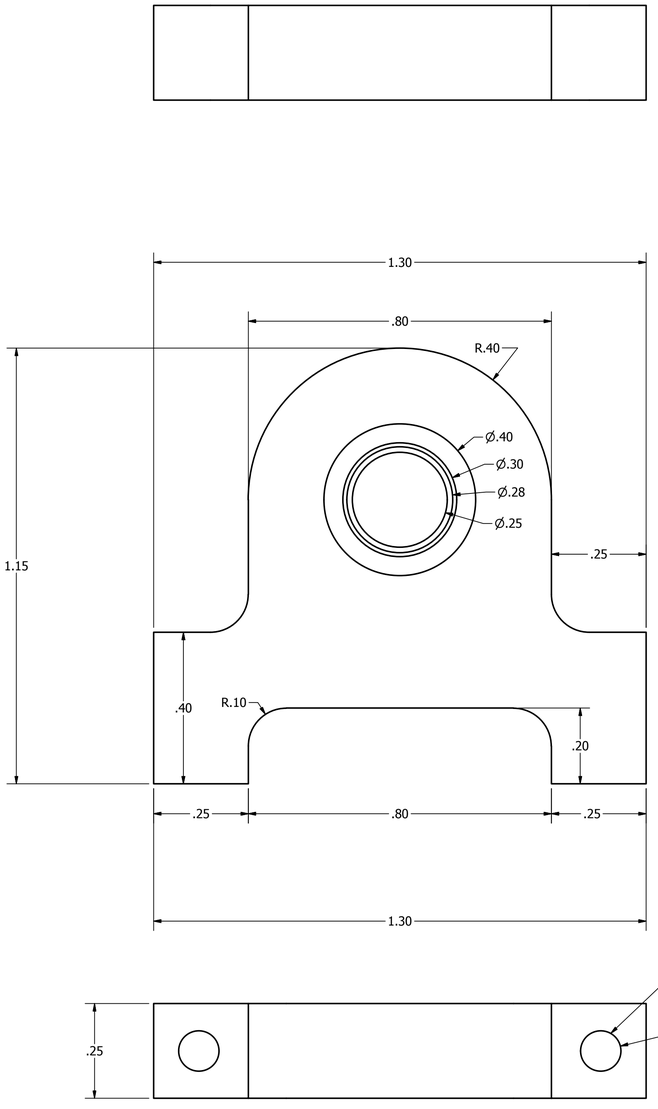


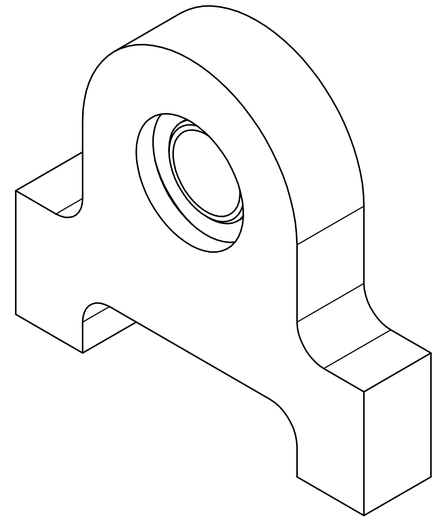




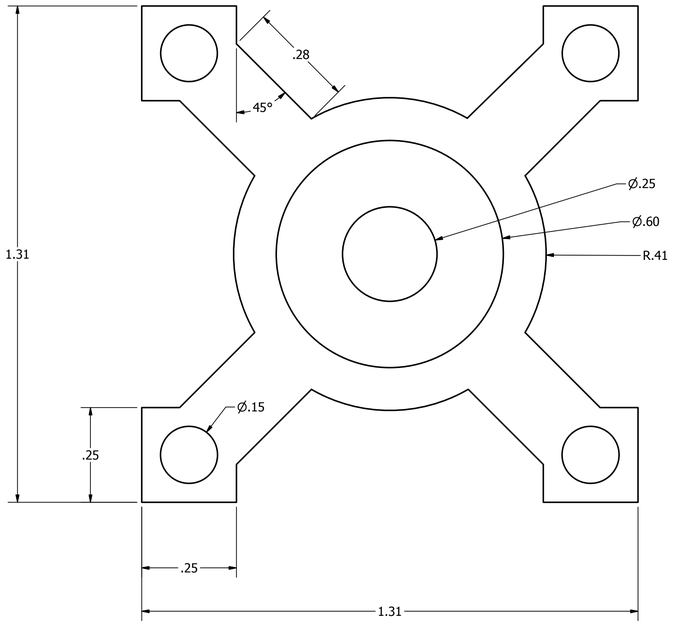


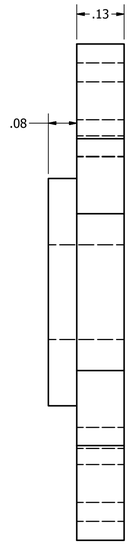
DRAWN siu853102150	9/30/2016		
CHECKED			
QA		TITLE	
MFG			
APPROVED			
		SIZE D	DWG NO Pan / REV
		SCALE 1 : 1	SHEET 1 OF 1

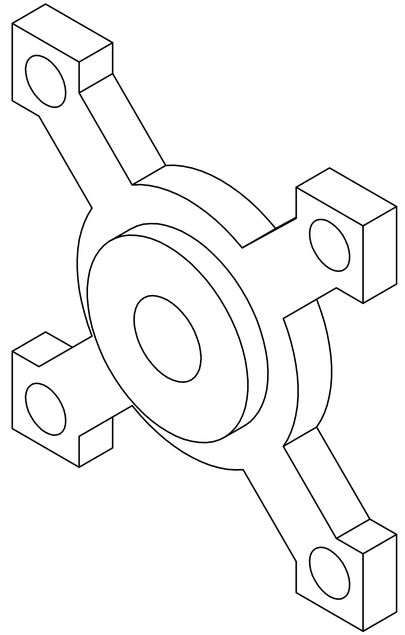




DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED			
		SIZE D	DWG NO Parallel Bearing / REV
		SCALE 5 : 1	SHEET 1 OF 1

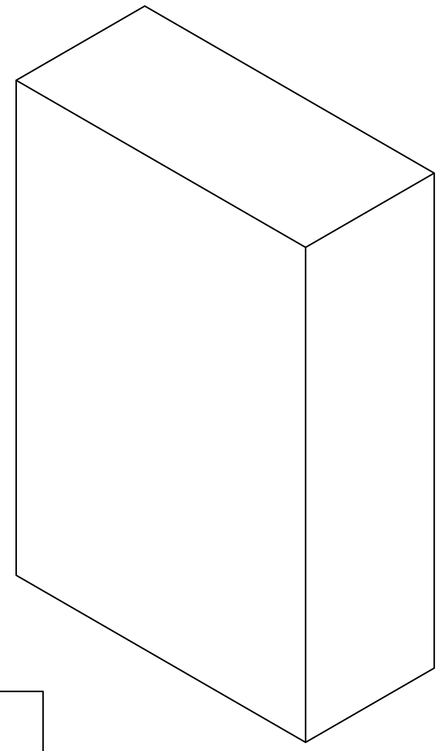


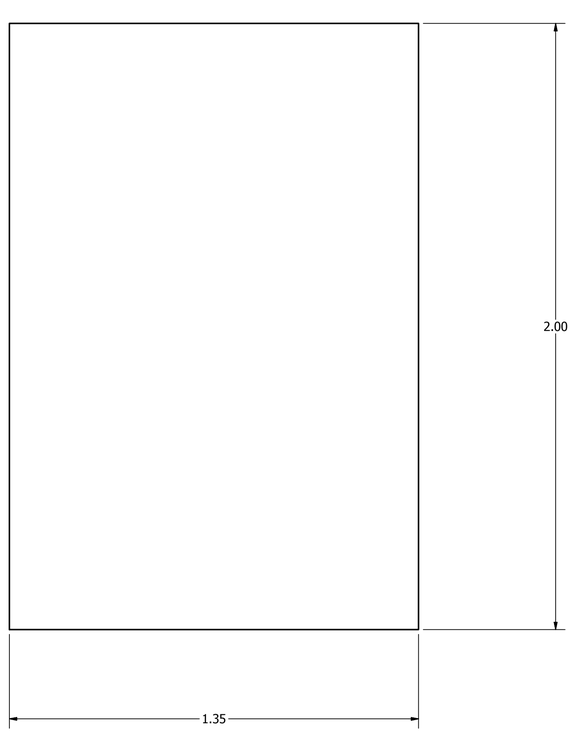


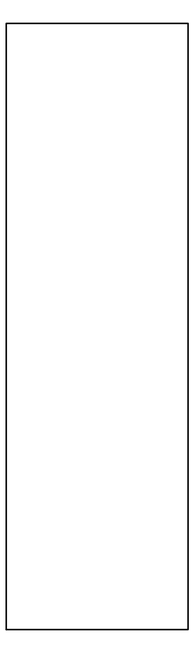


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Perpendicular Bearing / REV
		SCALE 5 : 1	SHEET 1 OF 1

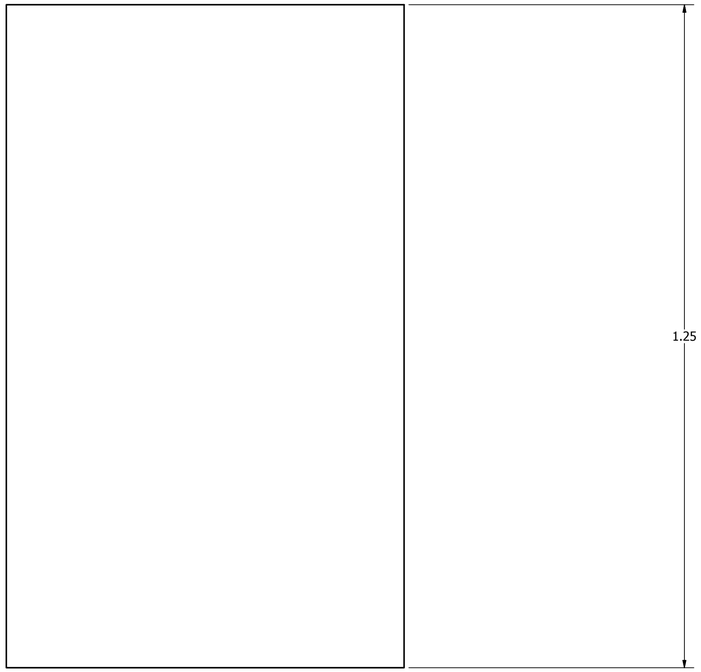
Radio Reciever will be attached by Velcro



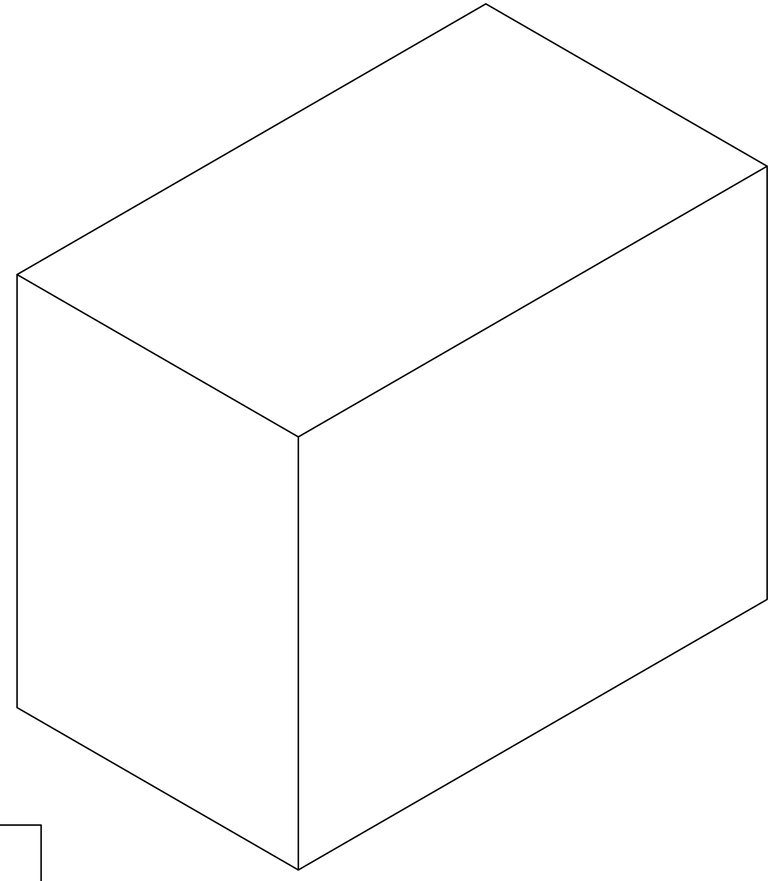


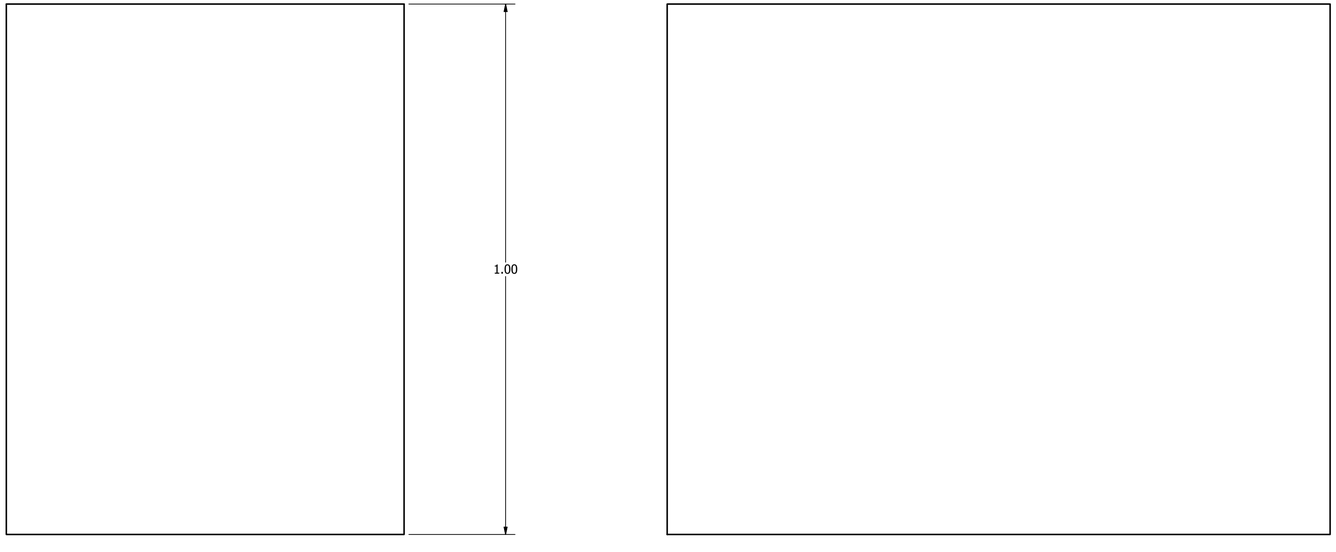


DRAWN siu853102150	9/30/2016			
CHECKED		TITLE		
QA				
MFG				
APPROVED		SIZE D	DWG NO Radio Reciever	REV
		SCALE 4 : 1	SHEET 1 OF 1	

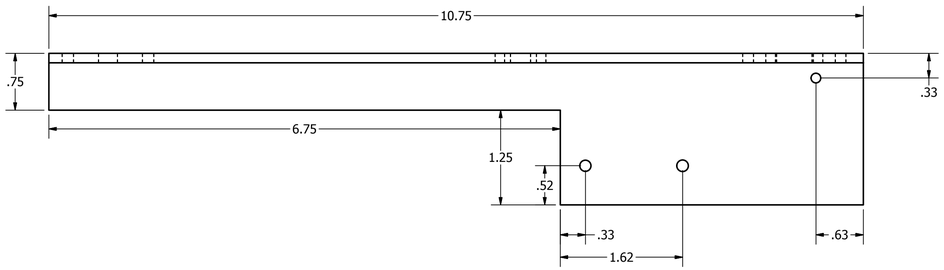


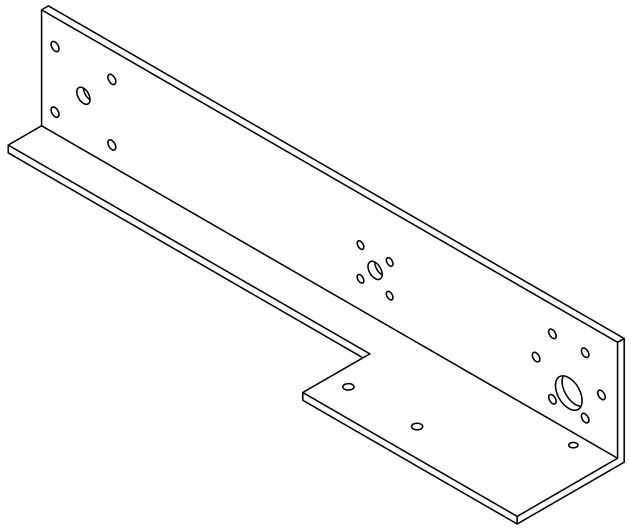
Relays will be attached with Velcro

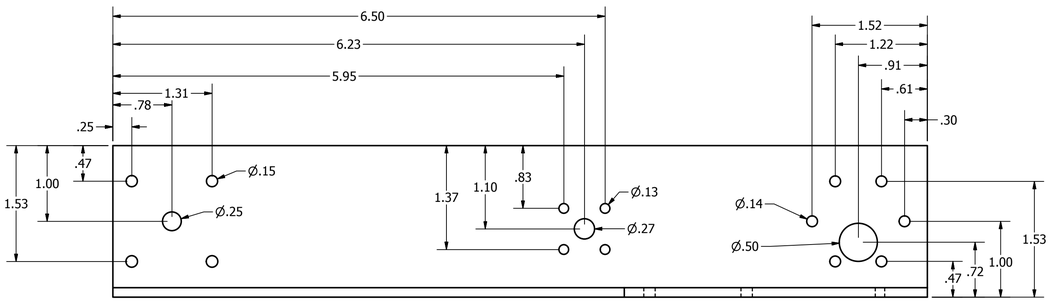


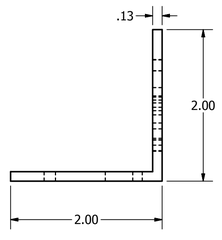


DRAWN siu853102150	9/30/2016		
CHECKED			
QA		TITLE	
MFG			
APPROVED		SIZE D	DWG NO Relay / REV
		SCALE 7 :1	SHEET 1 OF 1

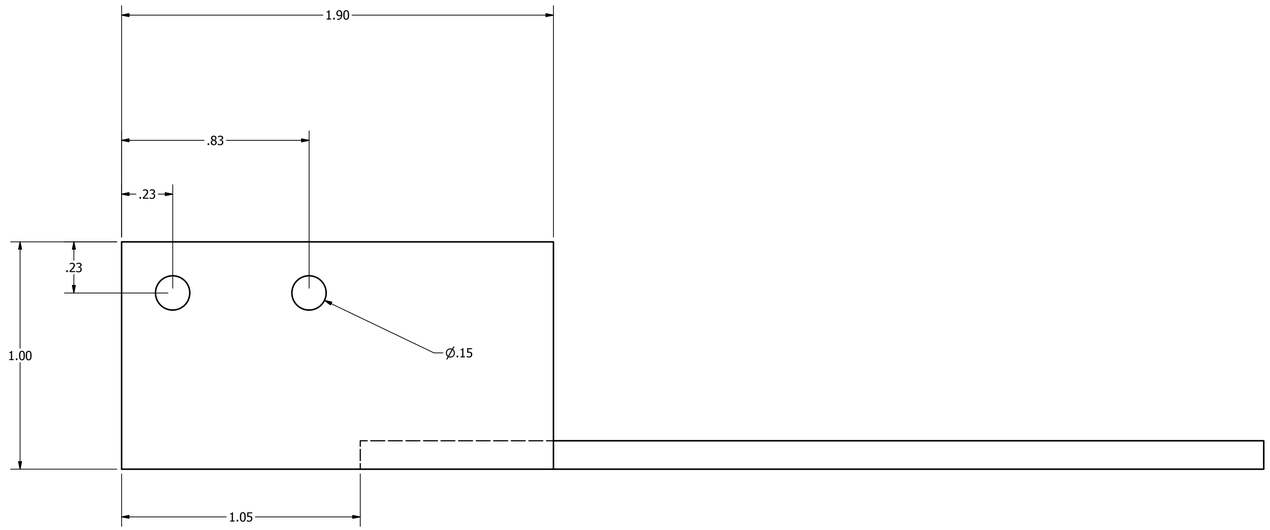


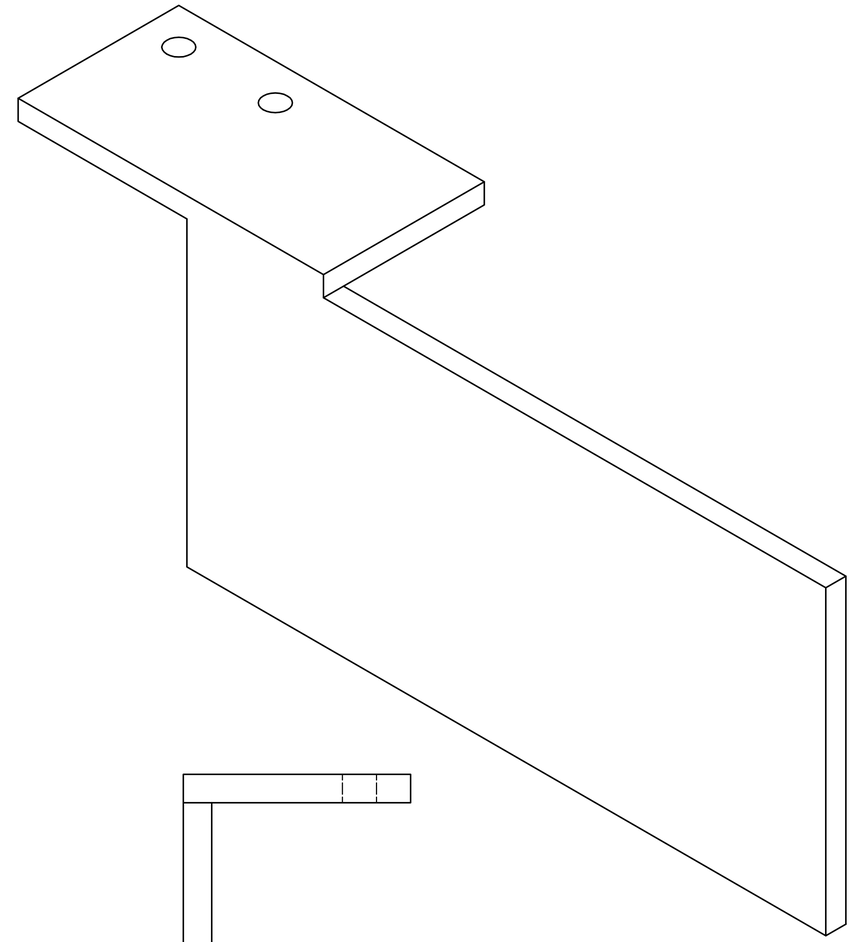


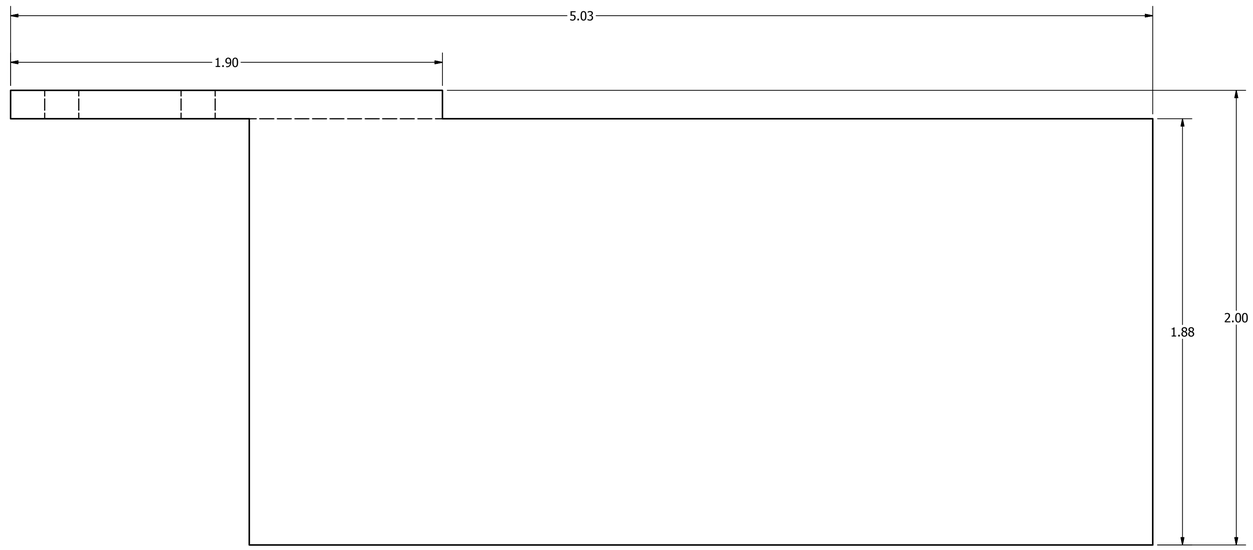


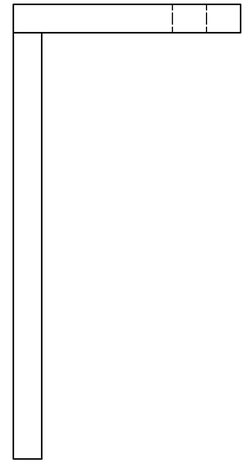


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Right 2in Angle / REV
		SCALE 1 : 1	SHEET 1 OF 1

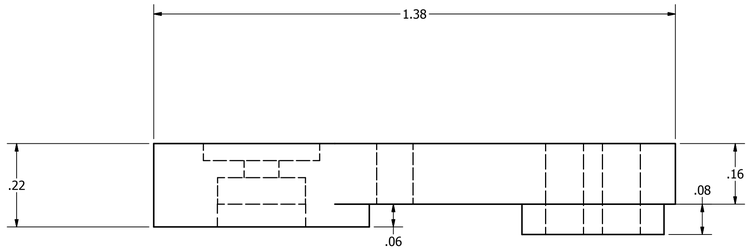


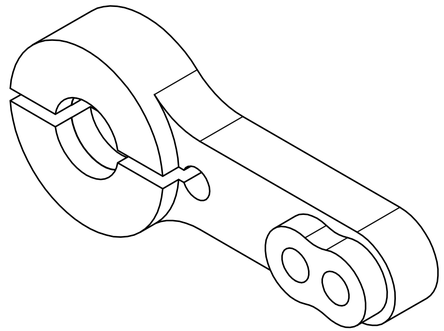


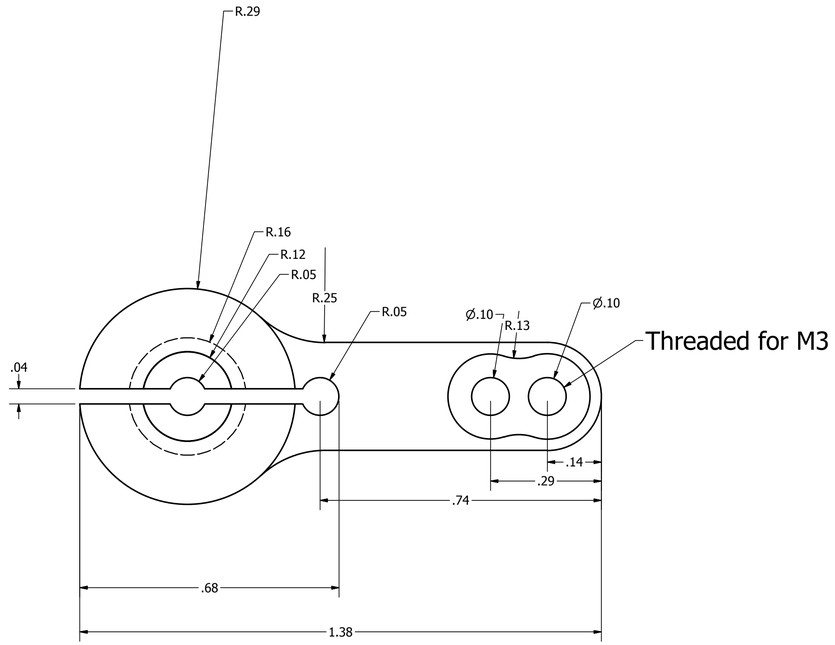


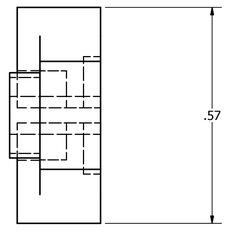


DRAWN siu853102150	10/5/2016			
CHECKED		TITLE		
QA				
MFG				
APPROVED		SIZE D	DWG NO Right Bumper	REV
		SCALE 3 : 1	SHEET 1 OF 1	

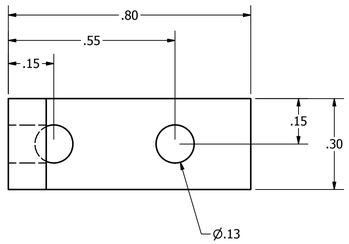


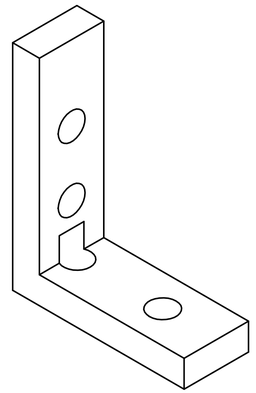




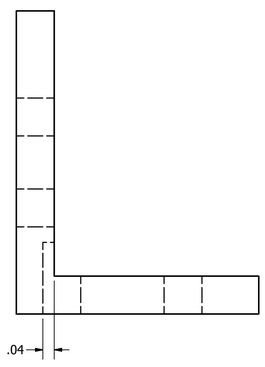


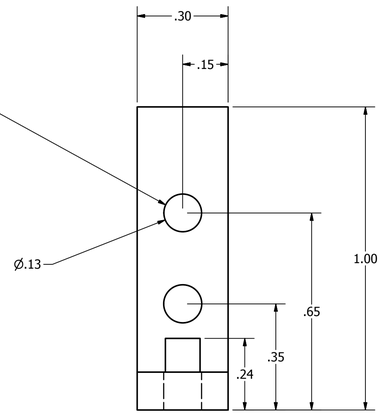
DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Servo Horn / REV
		SCALE 5 : 1	SHEET 1 OF 1



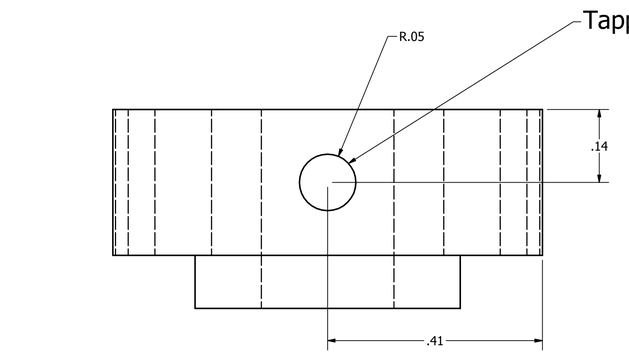


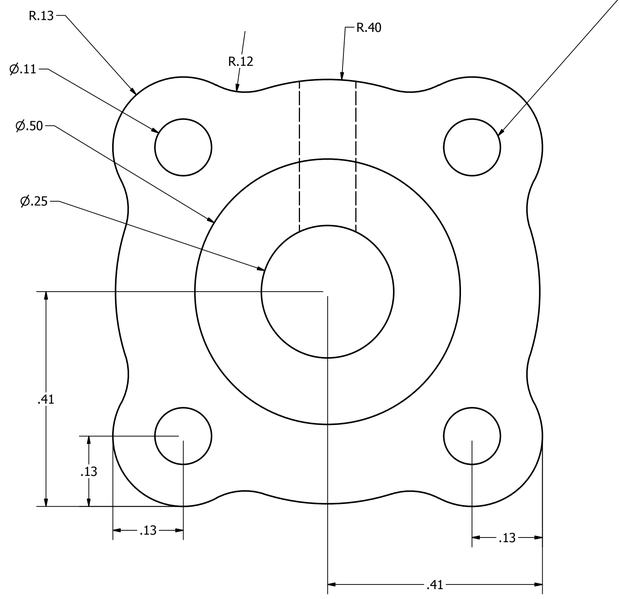
All Holes Taped for #8-3/8 Bolts

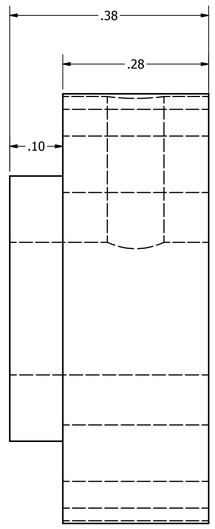


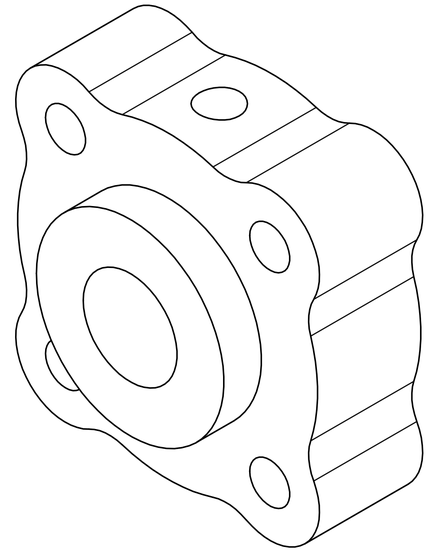


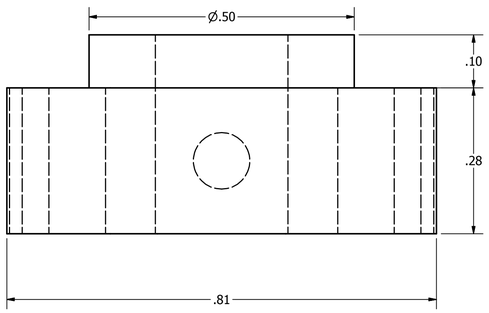
DRAWN siu853102150	9/30/2016			
CHECKED		TITLE		
QA				
MFG				
APPROVED		SIZE D	DWG NO Servo Mount	REV
		SCALE 4 : 1	SHEET 1 OF 1	



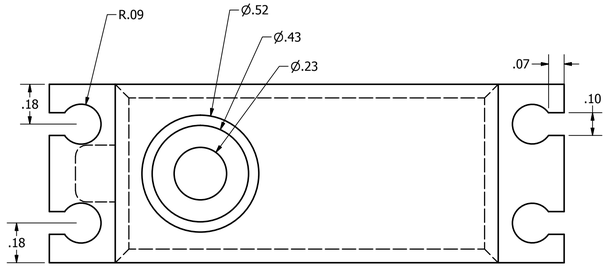


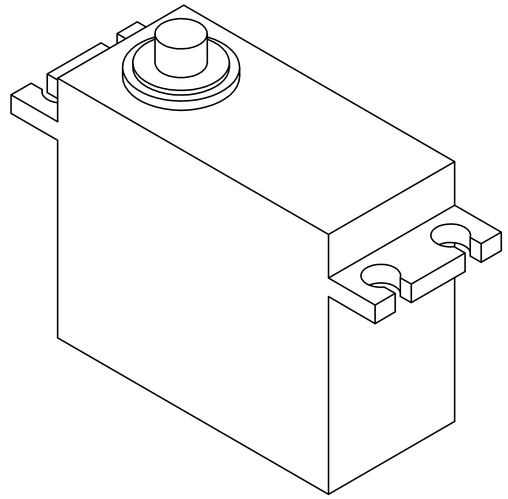


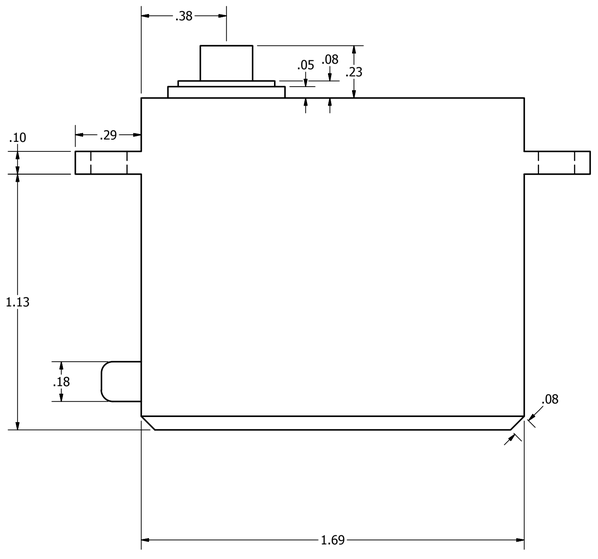


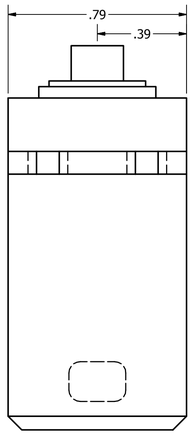


DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Set Screw Hub / REV
		SCALE 7: 1	SHEET 1 OF 1

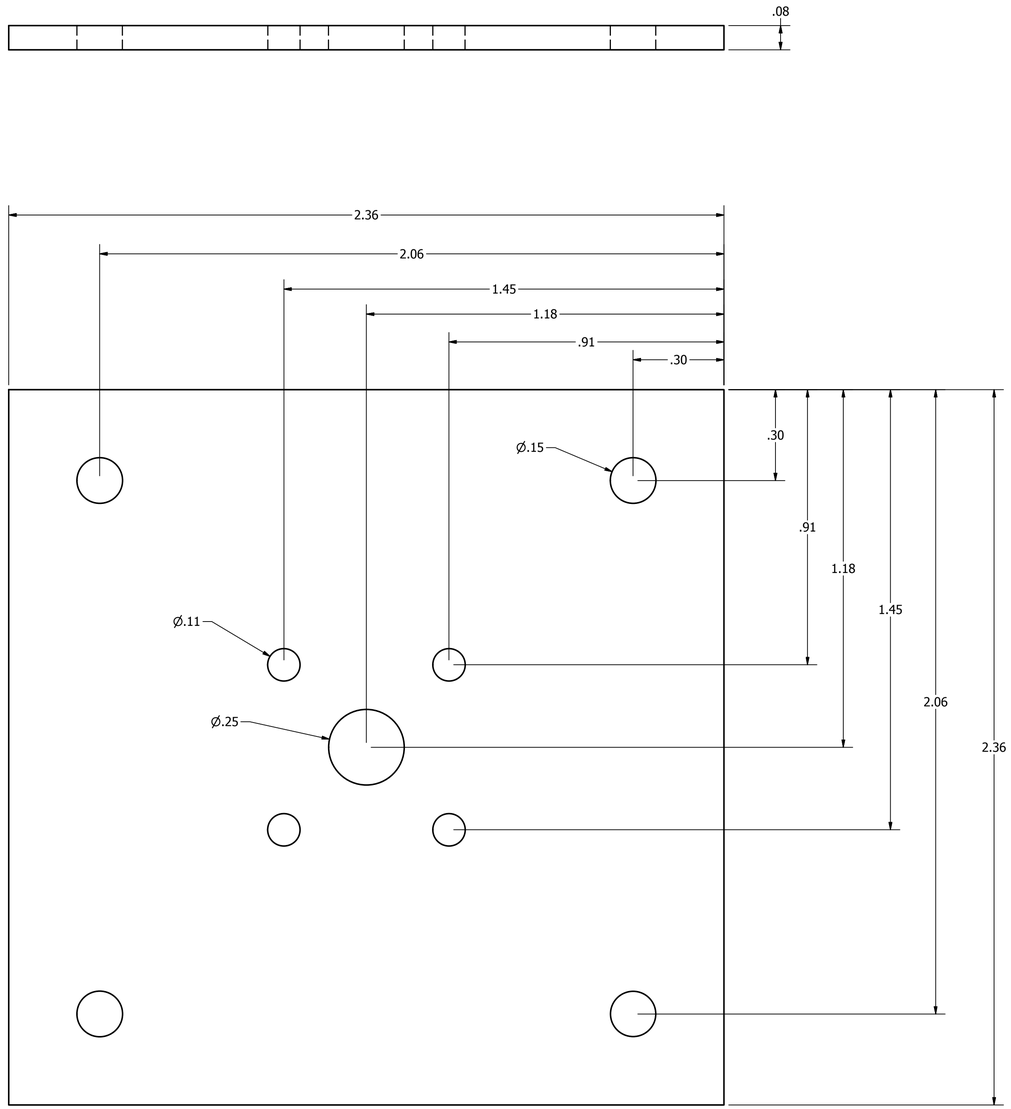


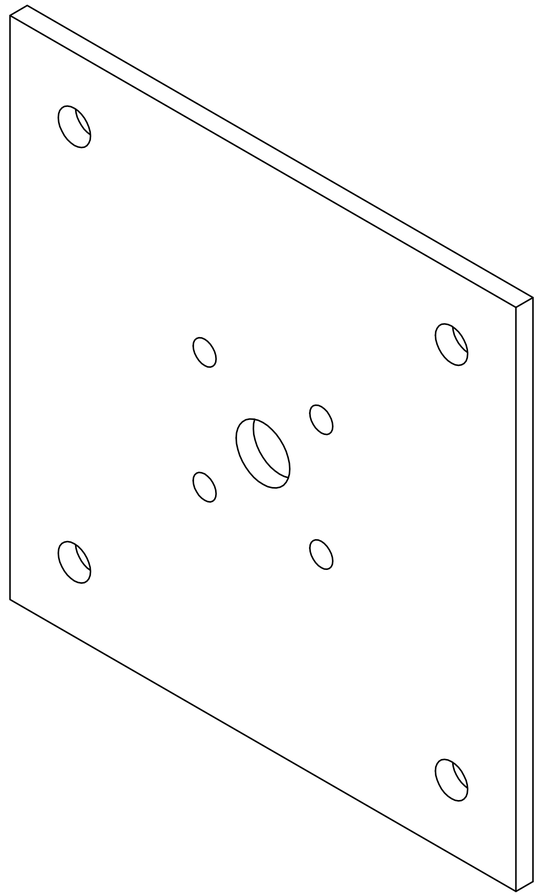




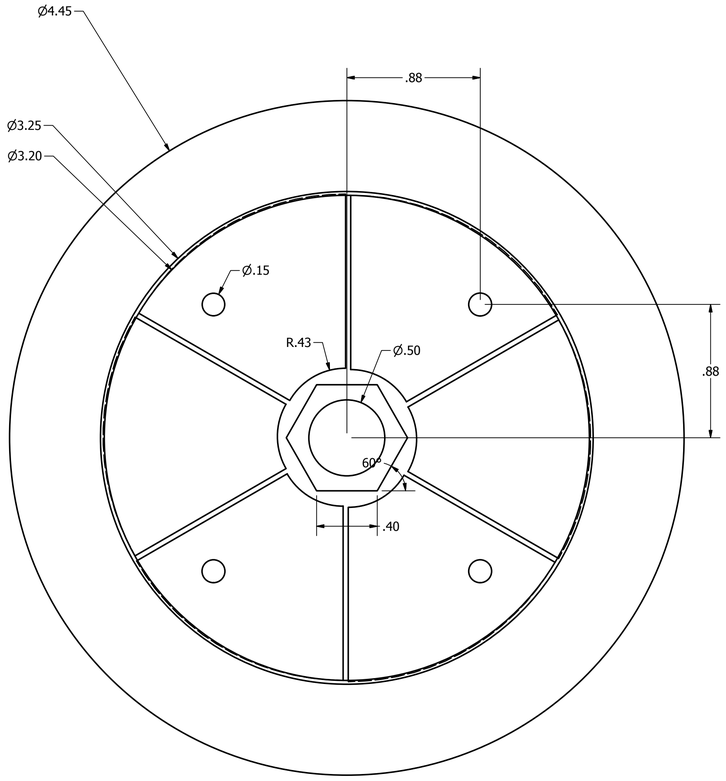


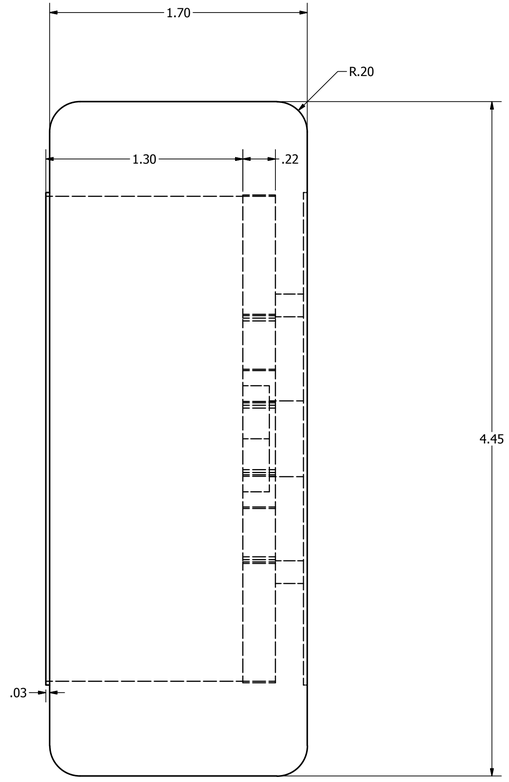
DRAWN siu853102150	10/5/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Tower Pro MG996R Servo
		SCALE 3 : 1	SHEET 1 OF 1

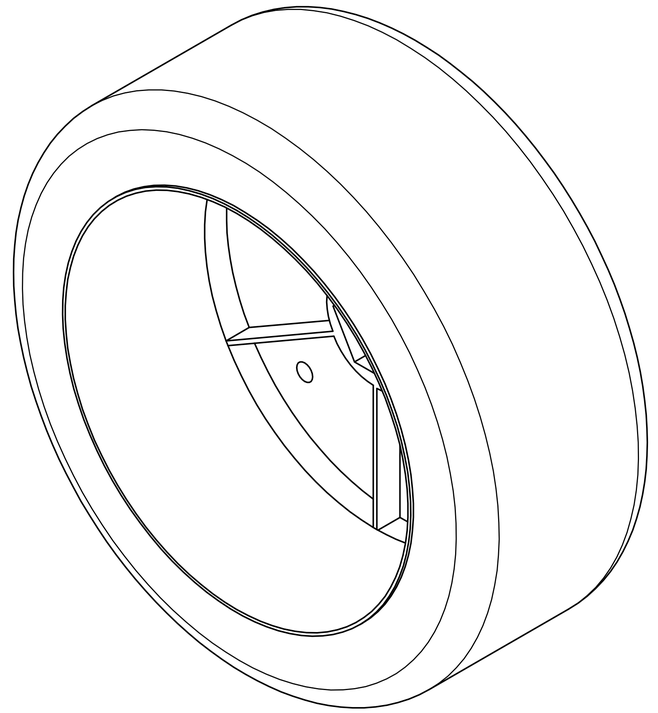




DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Wheel Mount / REV
		SCALE 4 : 1	SHEET 1 OF 1







DRAWN siu853102150	9/30/2016		
CHECKED		TITLE	
QA			
MFG			
APPROVED		SIZE D	DWG NO Wheel / REV
		SCALE 2 : 1	SHEET 1 OF 1