TEA-GSE Documentation and Procedure

Friday 18^{th}, December, 2017

Micah Johnson

Introduction

This outlines the procedure for using the TEA GSE (Transfer ESIS Alignment Ground Station Equipment), as well as contains all the electrical and mechanical parts, custom part drawings, electrical schematics, codes, troubleshooting, and potential improvements. The instrument should be capable of position measurements of the ESIS diffraction gratings to a precision of 14 micron in three channels.

When run, TEA will move towards and away from a grating, measuring its position. The positions will be saved in a test report to be compared to when the alignment is being transferred.

Procedure

Startup

1. Turn all lasers and computer on
2. Once the computer is warmed up, open Pycharm.
3. Plug the USB for the Agilent into the computer, and turn it on.
4. Plug the USB for the K-cube into the computer, and turn it on.
5. Turn the voltage source on, and set it to 5 Volts.
6. Place thermal isolating box (like a thick box of tin-foil) over the stages and gratings, and tape it down to thermally isolate the system.
7. Wait approximately 3 hours for the lasers to warm up and for everything to come to thermal equilibrium.

Running

1. If the photodiodes are going to be needed, and you are not just doing a thermal measurement, turn the switches on the electronics box to on.

2. Edit the configurations for `vis_euv_trans.py` in Pycharm to [start position (mm), stop position (mm), number of positions to measure, number of times to loop through the positions, grating type (VIS or EUV), grating number, comment]. (Note the comment will not be saved).

3. Run the program. (This will take several hours).

4. Allow it to run until a pdf test report is created.

5. It may prompt you to enter a new configuration. If so, input a new start position, end position, number of steps, and number of measurements per step.

Parts

The following parts are all the parts used in the buildup of the TEA GSE that were not custom designed. Custom parts can be found the the Mechanical section of this document.

Common Name	Part Number
Voltmeter	Agilent 34970A
Voltage Source	Agilent E3631A
Battery	Any 9 Volt battery
Laser	Thorlabs CPS635S
Laser Power Supply	Thorlabs LDS5
Lens	Thorlabs LA1576-A
Beam splitter	Thorlabs BS007
Retaining Right	Thorlabs SM9RR
Photodiode	Digikey Everlight 1080-1141-ND
Motor	Thorlabs MT1-Z8
Motor Base Plate	Thorlabs MT401
Motor Driver	Thorlabs KPS101
Manuals Stage	Thorlabs PT1

Software

Overview

The software for finding the position of the gratings runs Python and Matlab. Matlab can be opened by directing to the path `/usr/local/MATLAB/R2017a/bin`, and running "./matlab". Pycharm can be opened by directing to the `/Downloads/pycharm-community-2017.1.3/bin`, and running "./pycharm.sh". Matlab does not need to be open to run the software, but is recommended if small changes are needed.

When the software runs, Python (`vis_euv_trans.py`) collects voltage measurements from the Agilent and moves the stepper motor. The voltage measurements from the thermistors are converted into temperature using the SteinhartHart equation (`temp_conversion.py`). The thermistor temperatures are then sent to Matlab where they will be plotted (tmp.m). Voltages from the photodiode and positions of the stepper motor are recorded and sent to Matlab. Matlab (`TEA_GSE_Final_Align.m`) will plot the voltages, output new start/stop/stepsize for the motor, and test report.

There is also software from taking a series of temperature measurements without also driving the stepper motor (`long_temp.py`). It also uses the SteinhartHart equation (`temp_conversion.py`). Matlab then plots the temperature with time (`temp_long.m`). The configuration for the program is only the number of seconds that you want the temperature measurements taken over.

All software can be pulled from Gitlab at `https://titan.ssel.montana.edu/gitlab/MOSES-ESIS/ESIS/Optics/Transfer_ESIS_Alignment_GSE.git`. It also documents the installation proceedure.

Matlab

`TEA_GSE_Final_Align`

```
1
2
3 %OVERVIEW: This program takes the data from the agilent
      for three
4 %photodiode channels and outputs an adaptive grid which
      the stepper moter
5 %and agelant will test over to get the precision of the
      peak intensity to
```

```
%be within a certain tolerance. A test report will be generated after
%sufficient certanty in measurements is made.


% the following tripple commented lines are to be used for testing purposes
% by creating a simulated data matrix

%%% clear all
%%% close all
%%%
%%% stpsz=.002;
%%% iteration=1;
%%% dth=2.5:stpsz:5.5;
%%% vth1=-(dth-3.2).^2+dth+5;
%%% vth2=-(dth-3.1).^2+dth+4.5;
%%% vth3=-(dth-3.35).^2+dth+4.3;
%%%
%%% looops=10;
%%% n=0;
%%% dthex=[];
%%% vthex1=[];
%%% vthex2=[];
%%% vthex3=[];
%%%
%%% while n<looops
%%%     n=n+1;
%%%     dthex=[dthex dth];
%%%     vthex1=[vthex1 vth1];
%%%     vthex2=[vthex2 vth2];
%%%     vthex3=[vthex3 vth3];
%%% end
%%%
%%% dnos=((rand(1,(length(dth)*looops))-.5)/5000)+1;
%%% vnos1=((rand(1,(length(dth)*looops))-.5)/20)+1;
%%% vnos2=((rand(1,(length(dth)*looops))-.5)/20)+1;
%%% vnos3=((rand(1,(length(dth)*looops))-.5)/20)+1;
%%% vthnos1=vnos1.*vthex1;
%%% vthnos2=vnos2.*vthex2;
%%% vthnos3=vnos3.*vthex3;
```

```
45 %%% dthnos=dnos.*dthex;
46 %%%
47 %%% data=[dthnos;vthnos1;vthnos2;vthnos3]';
48 %%% csv_dir='/home/krg/Desktop/tester/iteration_3/
       data.csv';
49 %%% grating=3;
50 %%% type='VIS';
51
52
53
54 %input and outputs arguments
55 function [cont_loop, strt, stp, stept, mest,iteration]
       = TEA_GSE_Final_Align(csv_dir,stpsz,iteration,
       grating,type)
56
57 %polynomial fits are normally not ideal
58 warning('off','all');
59
60 %data will be interpreted from its csv format
61 iteration=iteration+1;
62 data=csvread(csv_dir);
63 iteration=iteration+1;
64
65 %stage location at measurement
66 d=data(:,1);
67 v1=data(:,2);
68 v2=data(:,3);
69 v3=data(:,4);
70
71 strt=min(d);
72 stp=max(d);
73 stps=round((stp-strt)/(stpsz))+1;
74 lps=round(length(d)/(stps-1));
75
76 %loop prep
77 zerlps=zeros(stps,lps);
78 ds=zerlps;
79 vs1=zerlps;
80 vs2=zerlps;
81 vs3=zerlps;
82 %takes the distance vs voltage data array and creates a
```

```
    matrix of simaler
%distance measurements to be averaged together later,
    and matracies of
%voltage measurements made at identical position to be
    bootstrapped later
for i=1:lps
    ds(:,i)=d(((i-1)*stps+1):(i*stps));
    vs1(:,i)=v1(((i-1)*stps+1):(i*stps));
    vs2(:,i)=v2(((i-1)*stps+1):(i*stps));
    vs3(:,i)=v3(((i-1)*stps+1):(i*stps));
end

[h,~]=size(vs1);

%loop prep
zerstp=zeros(1,stps);
dav=zerstp;
vav1=zerstp;
vav2=zerstp;
vav3=zerstp;
%finds the average voltage for a given distance
for n=1:stps
    dav(n)=mean(ds(n,:));
    vav1(n)=mean(vs1(n,:));
    vav2(n)=mean(vs2(n,:));
    vav3(n)=mean(vs3(n,:));
end

%does a 4th order polynomial fit to all data
pd=min(d):.0001:max(d);

p1=polyfit(d,v1,4);
p2=polyfit(d,v2,4);
p3=polyfit(d,v3,4);

pruf1=polyval(p1,pd);
pruf2=polyval(p2,pd);
pruf3=polyval(p3,pd);

%finds over which data range to base fit
off1=1.5;
```

```
121 off2=1.75;
122 off3=2;
123
124 %does a polynomial fit over relavent data. The higher
       the iteration number,
125 %the wider the range over which is fit
126 if iteration==1
127     pd1=(dav(vav1==max(vav1))-off1):.0001:(dav(vav1==
          max(vav1))+off1);
128     pd2=(dav(vav2==max(vav2))-off1):.0001:(dav(vav2==
          max(vav2))+off1);
129     pd3=(dav(vav3==max(vav3))-off1):.0001:(dav(vav3==
          max(vav3))+off1);
130 end
131 if iteration==2
132     pd1=(dav(vav1==max(vav1))-off2):.0001:(dav(vav1==
          max(vav1))+off2);
133     pd2=(dav(vav2==max(vav2))-off2):.0001:(dav(vav2==
          max(vav2))+off2);
134     pd3=(dav(vav3==max(vav3))-off2):.0001:(dav(vav3==
          max(vav3))+off2);
135 end
136 if iteration==3
137     pd1=(dav(vav1==max(vav1))-off3):.0001:(dav(vav1==
          max(vav1))+off3);
138     pd2=(dav(vav2==max(vav2))-off3):.0001:(dav(vav2==
          max(vav2))+off3);
139     pd3=(dav(vav3==max(vav3))-off3):.0001:(dav(vav3==
          max(vav3))+off3);
140 end
141
142 drel1=dav(dav>=min(pd1) & dav<=max(pd1));
143 drel2=dav(dav>=min(pd2) & dav<=max(pd2));
144 drel3=dav(dav>=min(pd3) & dav<=max(pd3));
145 vrel1=vav1(dav>=min(pd1) & dav<=max(pd1));
146 vrel2=vav2(dav>=min(pd2) & dav<=max(pd2));
147 vrel3=vav3(dav>=min(pd3) & dav<=max(pd3));
148
149 pav1=polyfit(drel1,vrel1,4);
150 pav2=polyfit(drel2,vrel2,4);
151 pav3=polyfit(drel3,vrel3,4);
```

```
152
153 pavd1=polyval(pav1,pd1);
154 pavd2=polyval(pav2,pd2);
155 pavd3=polyval(pav3,pd3);
156
157 %plots all data with fits
158 all_fig_combo=figure(1);
159 hold on
160 p1=scatter(d,v1,'r','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.05);
161 p2=scatter(d,v2,'b','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.05);
162 p3=scatter(d,v3,'g','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.05);
163 p4=scatter(dav,vav1,'m','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.1);
164 p5=scatter(dav,vav2,'c','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.1);
165 p6=scatter(dav,vav3,'k','MarkerFaceAlpha',.5,'
      MarkerEdgeAlpha',0.1);
166 p7=plot(pd,pruf1,'r');
167 p8=plot(pd,pruf2,'b');
168 p9=plot(pd,pruf3,'g');
169 p10=plot(pd1,pavd1,'m');
170 p11=plot(pd2,pavd2,'c');
171 p12=plot(pd3,pavd3,'k');
172 legend([p7 p8 p9 p10 p11 p12],{'All Data 1','All Data 2
      ','All Data 3','Averaged Data 1','Averaged Data 2','
      Averaged Data 3'})
173 xlabel('Distance (mm)')
174 ylabel('Voltage (V)')
175 title('Data and Averaged Data with Fits')
176
177 %saves as png. pdf would be preferable, but file is too
       large it crashes
178 %computer
179 print(all_fig_combo,strrep(csv_dir,'data.csv','
      all_fig_combo'),'-dpng')
180
181 %finds peak of data
182 dmv1=dav(vav1==max(vav1));
```

```
183 dmv2=dav(vav2==max(vav2));
184 dmv3=dav(vav3==max(vav3));
185
186 %adaptive grid for 2nd run through
187 if iteration==1
188     strt=min([dmv1 dmv2 dmv3])-off1;
189     stp=max([dmv1 dmv2 dmv3])+off1;
190     mest=20;
191     stept=.002;
192     cont_loop=0;
193 end
194
195 %adaptive grid for 3rd run through. stops if 4th run
      through. outputs pdf
196 %test report.
197 if iteration==2 ||iteration==3
198     %loop prep
199     times=1000;
200     tim=0;
201     mxxs1=zeros(1,times);
202     mxxs2=zeros(1,times);
203     mxxs3=zeros(1,times);
204     vbot1=zeros(1,lps);
205     vbot2=zeros(1,lps);
206     vbot3=zeros(1,lps);
207
208     while(tim<times)
209
210         %display progress through loop
211         if rem(10*tim,times)==0
212             sprintf('%g%%',round(100*tim/times))
213         end
214         tim=tim+1;
215
216         %samples with replacement from measurements at
              every distance
217         for i=1:h
218             vbot1(i)=mean(datasample(vs1(i,:),lps));
219             vbot2(i)=mean(datasample(vs2(i,:),lps));
220             vbot3(i)=mean(datasample(vs3(i,:),lps));
221         end
```

```
222
223         %does polynomaial fit to boostrapped data
224         pbot1=polyfit(dav,vbot1,4);
225         pbot2=polyfit(dav,vbot2,4);
226         pbot3=polyfit(dav,vbot3,4);
227         ft1=polyval(pbot1,pd1);
228         ft2=polyval(pbot2,pd2);
229         ft3=polyval(pbot3,pd3);
230         mx1=pd1(ft1==max(ft1));
231         mx2=pd2(ft2==max(ft2));
232         mx3=pd1(ft3==max(ft3));
233         %find max for bootstrapped data
234         mxxs1(tim)=mx1(1);
235         mxxs2(tim)=mx2(1);
236         mxxs3(tim)=mx3(1);
237     end
238
239     %creates a confidence interval
240     mxxs1=sort(mxxs1);
241     mxxs2=sort(mxxs2);
242     mxxs3=sort(mxxs3);
243
244     num_pt=round(times*.98/2,0);
245
246     lwbd1=mxxs1(times/2-num_pt);
247     lwbd2=mxxs2(times/2-num_pt);
248     lwbd3=mxxs3(times/2-num_pt);
249     hibd1=mxxs1(times/2+num_pt);
250     hibd2=mxxs2(times/2+num_pt);
251     hibd3=mxxs3(times/2+num_pt);
252
253     %plots histograms with 98% confidence intervals
254     hist1=figure();
255     hold on
256     hist(mxxs1,times/10)
257     line([lwbd1 lwbd1],ylim,'Color','r');
258     line([hibd1 hibd1],ylim,'Color','r');
259     hist2=figure();
260     hold on
261     hist(mxxs2,times/10)
262     line([lwbd2 lwbd2],ylim,'Color','r');
```

```
263     line([hibd2 hibd2],ylim,'Color','r');
264     hist3=figure();
265     hold on
266     hist(mxxs3,times/10)
267     line([lwbd3 lwbd3],ylim,'Color','r');
268     line([hibd3 hibd3],ylim,'Color','r');
269
270     %convert to micron
271     rng1=(hibd1-lwbd1)*1000;
272     rng2=(hibd2-lwbd2)*1000;
273     rng3=(hibd3-lwbd3)*1000;
274
275     %largest confidence interval
276     hi_rng=max([rng1 rng2 rng3]);
277
278     %change according to resolution buget. adaptive
            grid for 3rd run
279     %through.
280     if hi_rng<=15.3
281         pf='Passed';
282         strt=min([dmv1 dmv2 dmv3])-off2;
283         stp=max([dmv1 dmv2 dmv3])+off2;
284         mest=30;
285         stept=.002;
286         cont_loop=0;
287     else
288         pf='Failed';
289         strt=0;
290         stp=0;
291         mest=0;
292         stept=0;
293         cont_loop=2;
294     end
295
296     %saves histograms. pdf would be better
297     print(hist1,strrep(csv_dir,'data.csv','Histogram 1'
            ),'-dpng')
298     print(hist2,strrep(csv_dir,'data.csv','Histogram 2'
            ),'-dpng')
299     print(hist3,strrep(csv_dir,'data.csv','Histogram 3'
            ),'-dpng')
```

```
300
301     %begin formatting for pdf output
302     %all fig combo file location
303     alfg=strrep(csv_dir,'data.csv','all_fig_combo.png')
            ;
304     %variables from script to be used in test report
305     variableList={type num2str(grating) num2str(mode(
            mxxs1)) num2str(rng1) num2str(mode(mxxs2))
            num2str(rng2) num2str(mode(mxxs3)) num2str(rng3)
             num2str(iteration) num2str(lps) num2str(stps)
            alfg pf};
306
307     %tex lines with script inputs
308     line0 = cellstr(['{\textbf{Test ',variableList{13},
            '} \par}']);
309     line1 = cellstr(['{\scshape\large Grating Type:',
            variableList{1},' \par}']);
310     line2 = cellstr(['{\scshape\large Grating Number:'
            variableList{2} ' \par}']);
311     line3 = cellstr(['\indent...for channel 1 lies at '
             variableList{3} ' mm to a precision of '
            variableList{4} ' microns.\\']);
312     line4 = cellstr(['\indent...for channel 2 lies at '
             variableList{5} ' mm to a precision of '
            variableList{6} ' microns.\\']);
313     line5 = cellstr(['\indent...for channel 3 lies at '
             variableList{7} ' mm to a precision of '
            variableList{8} ' microns.\\']);
314     line6 = cellstr(['\indent...after ' variableList{9}
             ' intensity scans.\\']);
315     line7 = cellstr(['\indent...with ' variableList{10}
             ' measurements taken at each distance.\\']);
316     line8 = cellstr(['\indent...with data from '
            variableList{11} ' distances.\\']);
317     line9 = cellstr(['\includegraphics[]{' variableList
            {12} '}\\']);
318
319     %entire tex file
320     str = ['\documentclass[12pt,a4paper]{article}'
321            '\usepackage{datetime}'
322            '\usepackage{ragged2e}'
```

```
323             '\usepackage{graphicx}'
324             '\usepackage{subcaption}'
325             '\usepackage{mwe}'
326             '\usepackage{float}'
327             '\usepackage{grffile}'
328             '\pagenumbering{gobble}'
329             ''
330             '\begin{document}'
331             ''
332             '\begin{center}'
333             '{\scshape\LARGE VIS-EUV Test Report \par}'
334             '{\scshape\Large \today, \currenttime \par}'
335             '\bigskip'
336             line1
337             line2
338             '\bigskip'
339             line0
340             '\end{center}'
341             ''
342             '\noindent We are 98\% confident that the
                  peak intensity...\\'
343             line3
344             line4
345             line5
346             ''
347             '\noindent The final model was created...\\'
348             line6
349             line7
350             line8
351             ''
352             '\begin{figure}[H]'
353             '\centering'
354             line9
355             '\end{figure}'
356             ''
357             '\end{document}'];
358
359     %file path of tex document
360     pth=csv_dir(1:end-20);
361     pthfol=strcat(pth,'reports');
362     pth=strcat(pth,'reports/Test_Report.tex');
```

```
363
364     %prepare file for tex report
365     formatSpec = '%s\r\n'; %specify the data type
            being output through fprintf
366     [nRows,~] = size(str); %tell matlab the size of
            the file
367     mkdir(pthfol);
368     fileID = fopen(pth,'wt');   %open the blank report
            file
369
370     %writes tex file line by line
371     for row = 1:nRows
372         fprintf(fileID,formatSpec,str{row,:});      %
                print the data onto the file
373     end
374
375     fclose(fileID); %close the finished file
376     cd (pth(1:end-15));
377
378     %compile and open latex pdf document
379     system('pdflatex Test_Report.tex > /dev/null 2>&1')
            ;
380     system('xdg-open Test_Report.pdf');
381
382 end
```

tmp

```
1
2
3  function tmp(csv_dir)
4
5  dat=csvread(csv_dir);
6
7  sec=dat(:,16);
8
9  temp=figure();
10 hold on
11 for i=[1:4 6:10]
12     t=dat(:,(i+5));
13     plot(sec,t)
```

```
end

legend('1','2','3','4','6','7','8','9','10')
xlabel('Time (seconds)')
ylabel('Temperature (F)')
title('Temperature in Time')

print(temp,strrep(csv_dir,'data.csv','Temperature'),'-dpdf');
```

temp_long

```


function temp_long(csv_dir)

dat=csvread(csv_dir);

sec=dat(:,12);

temp=figure();
hold on
for i=[1:4 6:10]
    t=dat(:,(i+1));
    plot(sec,t)
end

legend('1','2','3','4','6','7','8','9','10')
xlabel('Time (seconds)')
ylabel('Temperature (F)')
title('Temperature in Time')

print(temp,strrep(csv_dir,'data.csv','Temperature'),'-dpdf');
```

Python

vis_euv_trans.py

```
# Python 3 Program for moving ThorLabs Z812B motorized stage and taking  intensity measurements with Agilent 34970A
```

```
2  # Calling sequence: python3 vis_euv_trans.py [start
       position] [end position] [number of steps] [number
       of measurements per step]
3
4  import thorpy
5  import thorpy.message
6  import thorpy.comm.discovery as comm
7  import thorpy.stages
8  import agilent
9  from temp_conversion import steinhart
10 import os
11 import csv
12 import time
13 from time import sleep
14 import sys
15 import matlab.engine
16 eng = matlab.engine.start_matlab()
17 eng.cd(r'../Matlab')
18
19 # Check if the program received the correct number of
       arguments
20 if len(sys.argv) != 8:
21     sys.exit("Incorrect number of arguments! \n Calling
           sequence: python3 vis_euv_trans.py [start
          position] [end position] [number of steps] [
          number of measurements per step] [VIS | EUV] [
          Grating #] [Comment] \n" + str(sys.argv))
22
23 # Set up measurement variables
24 start_pos = float(sys.argv[1])    # Position where the
       sequence starts, measured from the home position
25 end_pos = float(sys.argv[2])     # Position where the
       sequence ends, measured from the home position
26 num_steps = int(sys.argv[3])    # Number of steps to
       take with the motor between start and end position
27 num_meas = int(sys.argv[4])       # Number of intensity
       measurements to take per step
28
29 # Type of grating to analyze for this measurement (
       Visible or EUV)
30 grating_type = str(sys.argv[5])
```

```
31 if(grating_type != "VIS" and grating_type != "EUV"):
32     sys.exit("Incorrect grating type argument, use 'VIS
       ' or 'EUV'")
33
34 # Serial number of the grating under test
35 grating_num = int(sys.argv[6])
36 if(grating_num < 1 or grating_num > 6):
37     sys.exit("Incorrect grating number specified. Use a
           value between 1 and 6")
38
39 # Comment associated with this test run
40 grating_comment = str(sys.argv[7])
41
42 # prepare the directory structure
43 grating_dir = "/home/krg/ESIS_VisEUV_transfer_sw/Output
    /" + grating_type + "/" + "grating_" + str(
    grating_num)
44 if not os.path.exists(grating_dir):
45     os.makedirs(grating_dir)
46
47 timestr = time.strftime("%Y%m%d-%H%M%S")
48 exp_dir = grating_dir + "/" + timestr
49 os.makedirs(exp_dir)
50
51 report_dir = exp_dir + "/reports"
52 os.makedirs(report_dir)
53 iter_dir = exp_dir + "/iterations"
54 os.makedirs(iter_dir)
55
56
57
58 cont_loop = 0       # 0 = for continuing adptive mesh
    refinement, 1 = for manual mesh refinement, 2 =
    bootstrapping converged, exit program.
59 iteration = 0
60
61 # Open the Z812B motorized stage
62 motor = None
63 x = comm.discover_stages()
64 for i in x:
65     motor = i
```

```
66
67 if motor == None:
68     sys.exit("Z812 stage not connected")
69
70 # Open the Agilent 34970A for voltage measurements
71 voltmet = agilent.init_serial()
72
73 # Prepare the motor for a sequence
74 motor.acceleration = 0.5
75 motor.max_velocity = 2.3    # Set max velocity
      parameter on Z812 (mm/s)
76 motor.print_state()
77
78 motor.home()
79
80 while(True):
81
82     # Check the return value of Matlab script to
         determine what to do next
83     if(cont_loop == 1):
84         [start_pos, end_pos, num_steps, num_meas] =
            input("Adaptive grid selection failed,
            Please enter the new gridsize...")
85     elif(cont_loop ==2):
86         break
87
88     dx = (end_pos - start_pos) / num_steps  # Amount to
         move for each step
89
90     # Check that the arguments are physically
         attainable
91     if start_pos == end_pos:
92         sys.exit("Nothing to do. Start position equal
            to end position")
93
94     if min(start_pos, end_pos) < motor.
         home_offset_distance:
95         sys.exit("Minimum value too close to end of
            motor")
96
97     if max(start_pos,end_pos) > (12.0 - motor.
```

```
        home_offset_distance):
98          sys.exit("Maximum value too close to end of
                motor")
99
100     if num_meas <= 0:
101         sys.exit("Number of measurements must be larger
                than zero")
102
103     if num_steps <= 0:
104         sys.exit("Number of steps must be larger than
                zero")
105
106     # Prepare directory for this iteration
107     cur_dir = iter_dir + "/iteration_" + str(iteration)
108     os.makedirs(cur_dir)
109
110     # Open CSV file for writing output
111     csv_fn = cur_dir + "/data.csv"
112     csvfile =  open(csv_fn, 'w', newline='')
113     csv_writer = csv.writer(csvfile)
114
115
116     # Loop through the sequence
117     for num in range(0,num_steps+1):
118
119         # Make sure the motor isn't going to be damaged
                by the move
120         if(motor.status_motor_current_limit_reached or
                motor.status_motion_error or motor.
                status_forward_hardware_limit_switch_active
                or motor.
                status_reverse_hardware_limit_switch_active)
                :
121             sys.exit("Z812B encountered an error")
122
123         # Move the motor to the new position and wait
                for completion
124         motor.position = start_pos + num * dx
125         sleep(0.2)
126         while (motor.status_in_motion_forward or motor.
                status_in_motion_reverse or motor.
```

```
            status_in_motion_jogging_forward or motor.
            status_in_motion_jogging_reverse or motor.
            status_in_motion_homing):
127             sleep(0.1)
128             #motor.print_state()
129
130         # Take the intesity measurements
131         data = []
132         data.append(str(motor.position))
133         for j in range(0, num_meas):        # Take
                intensity measurements
134             raw = agilent.measure(voltmet, 101, 101)
135             rawlist = raw.split(",")
136             for k in range(0, len(rawlist)):
137                 if k < 3:   # photodiode measurement
138                     data.append(rawlist[k])
139                 else:
140                     data.append(steinhart(rawlist[k]))
141
142         csv_writer.writerow(data)   # write next row of
                data to file
143         csvfile.flush()
144
145
146
147     csvfile.close() # Close CSV file
148
149     [cont_loop, start_pos, end_pos, num_steps, num_meas
        , iteration] = eng.complete_align_single(csv_fn,
         iteration,grating_num,grating_type, nargout=6)
150     cont_loop = int(cont_loop)
151     num_steps = int(num_steps)
152     num_meas = int(num_meas)
153     iteration= int(iteration)
154     print(cont_loop, start_pos, end_pos, num_steps,
        num_meas,iteration)
155
156
157
158
159 input("Press Enter to quit...")
```

temp_conversion

```
import math

def steinhart(V):
    Rb = 4990
    C1 = 1.285e-3
    C2 = 2.362e-4
    C3 = 9.285e-8
    Vin = 12.

    if V <= 0:
        V = 1e-6

    # R = -Rb * V / (V-Vin)
    R = Rb * (Vin/V - 1)
    return 1 / (C1 + C2 * math.log(R) + C3 * math.pow(
        math.log(R), 3)) - 273.15
```

long_temp

```


import thorpy
import thorpy.message
import thorpy.comm.discovery as comm
import thorpy.stages
import agilent
from temp_conversion import steinhart
import os
import csv
import time
from time import sleep
import sys
import matlab.engine
eng = matlab.engine.start_matlab()
eng.cd(r'../Matlab')


duration = float(sys.argv[1])

```

```
21 temp_dir = "/home/krg/Transfer_ESIS_Alignment_GSE/
       control_software/Output/Temperature"
22
23 if not os.path.exists(temp_dir):
24     os.makedirs(temp_dir)
25
26 timestr = time.strftime("%Y%m%d-%H%M%S")
27 exp_dir = temp_dir + "/" + timestr
28 os.makedirs(exp_dir)
29
30 # Open the Agilent 34970A for voltage measurements
31 voltmet = agilent.init_serial()
32
33 csv_fn = exp_dir + "/data.csv"
34 csvfile = open(csv_fn, 'w', newline='')
35 csv_writer = csv.writer(csvfile)
36
37 start_time=time.time()
38
39 dif = 0
40
41 while dif < duration:
42     data = []
43     raw = agilent.measure(voltmet,104,114)
44     rawlist = raw.split(",")
45     for k in range(0,(len(rawlist)+1)):
46         if k == 0:  # photodiode & Vin measurement
47             data.append(rawlist[k])
48         elif k == 11:
49             dif = time.time()-start_time
50             data.append(dif)
51         else:
52             data.append(steinhart(float(rawlist[k]),
                   float(rawlist[0])))
53
54     csv_writer.writerow(data)  # write next row of data
           to file
55     csvfile.flush()
56
57 eng.temp_long(csv_fn, nargout=0)
58
```

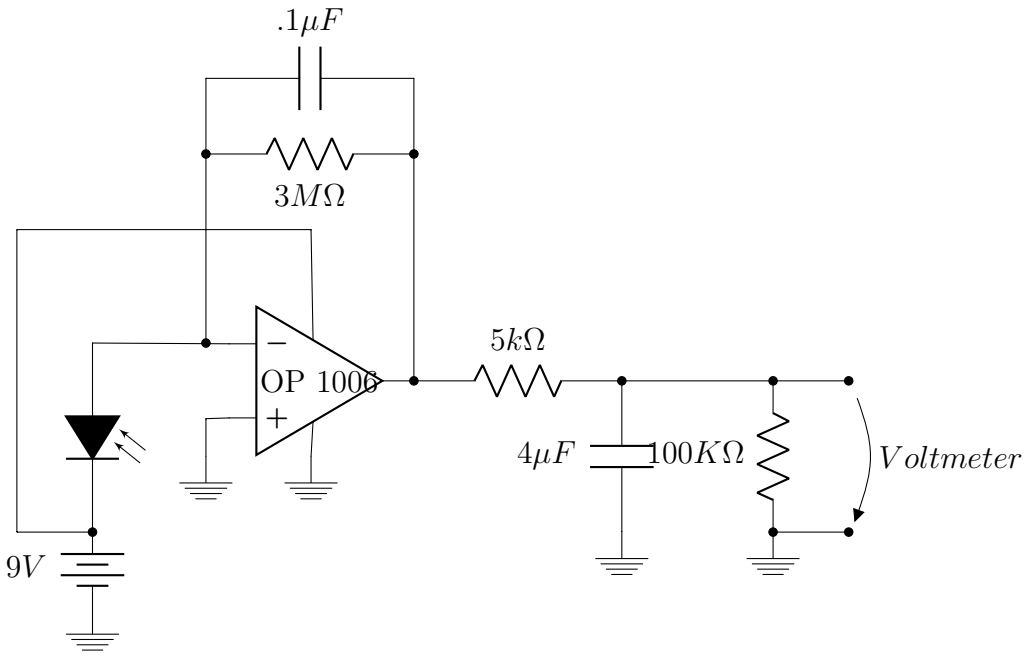
```
59 input("Press Enter to quit ...")
```

Electrical

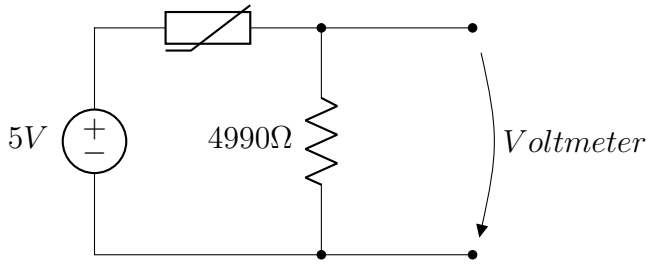
Overview

There are two electronic systems used in measurements for the TEA GSE. One is a simple voltage divider with a thermistor and resistor. This is used to measure the temperature. There are 10 repetitions of the temperature circuit so the temperature of 10 locations can be measured simultaneously. One of the channels does not work, and is labeled as such. The other circuit is meant for measuring the photodiodes. The photodiode circuit converts the current measurement of the photodiode into voltage, and filters out the high frequency noise. There are 3 repetitions of the photodiode circuit for each of the three channels. All outgoing wires are connected by D-subs.

Photodiode



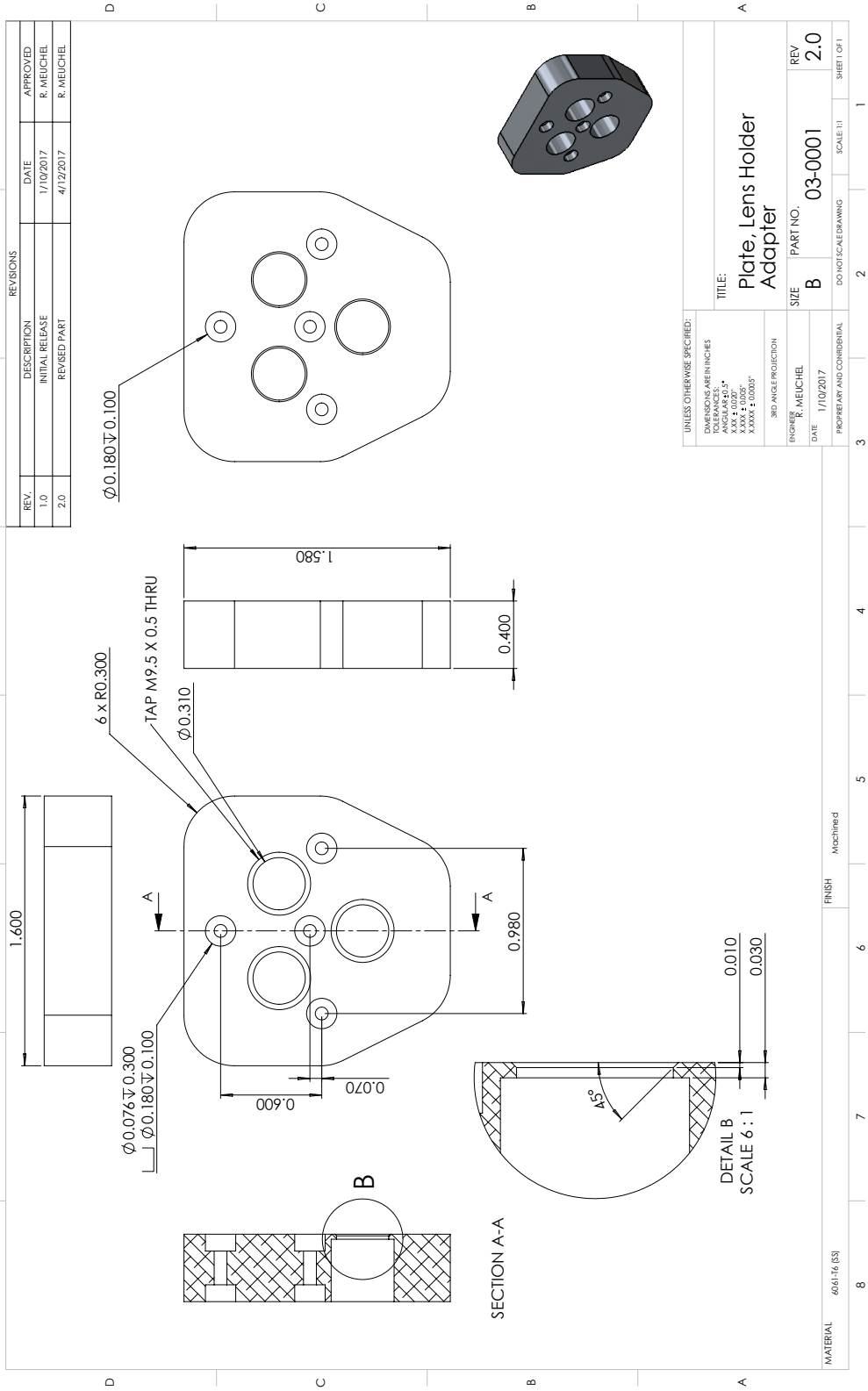
Temperature



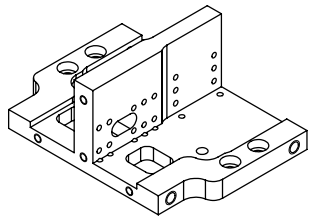
Mechanical

The parts designed and built for the grating alignment can be divided up into "TEA GSE", "External", and "Electrical". The TEA GSE includes the parts used to actually make the grating measurements. It can be attached to another kind of buildup to make measurements of another kind of mirror or grating, attached in any other way. The external buildup is used to hold the TEA, stages, and gratings. It is specific to the ESIS gratings held by the tuffet. The Electrical buildup holds all the electronics, apart from the voltmeter and voltage supply. The housing for the measurement electronics is also included.

TEA GSE


REVISIONS
REV. DESCRIPTION DATE APPROVED
1.0 INITIAL RELEASE 1/10/2017 R. MEUCHEL
2.0 REVISED PART 4/12/2017 R. MEUCHEL
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR: ±0.5°
X.XX ± 0.020"
X.XXX ± 0.005"
X.XXXX ± 0.0005"
3RD ANGLE PROJECTION
ENGINEER R. MEUCHEL
DATE 1/10/2017
PROPRIETARY AND CONFIDENTIAL
TITLE:
Plate, Lens Holder Adapter
SIZE B
PART NO. 03-0001
REV 2.0
DO NOT SCALE DRAWING
SCALE: 1:1
SHEET 1 OF 1
MATERIAL 6061-T6 (SS)
FINISH Machined
⌀0.180 ⌄ 0.100
1.580
0.400
6 x R0.300
TAP M9.5 X 0.5 THRU
⌀0.310
1.600
A
A
0.980
⌀0.076 ⌄ 0.300
⌊ ⌀0.180 ⌄ 0.100
0.600
0.070
0.010
0.030
45°
B
SECTION A-A
DETAIL B
SCALE 6 : 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	1/11/2016	R. MEUCHEL
1.1	CHANGE SLOT TO SQUARE CUTOUT	6/28/2017	R. MEUCHEL



MATERIAL	6061-T6 (SS)	FINISH	Machined

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ±0.5°
X.XX ± 0.020"
X.XXX ± 0.005"
X.XXXX ± 0.0005"

3RD ANGLE PROJECTION

ENGINEER: R. MEUCHEL

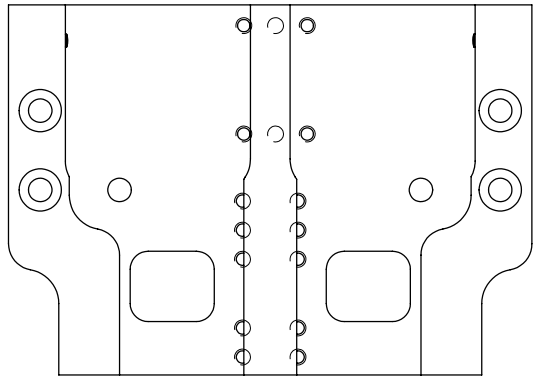
DATE: 1/11/2016

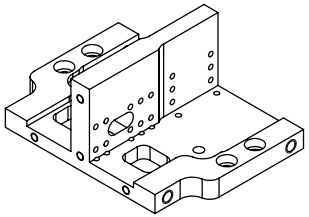
PROPRIETARY AND CONFIDENTIAL

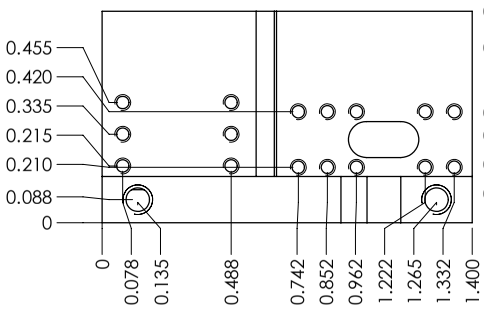
TITLE: Beam

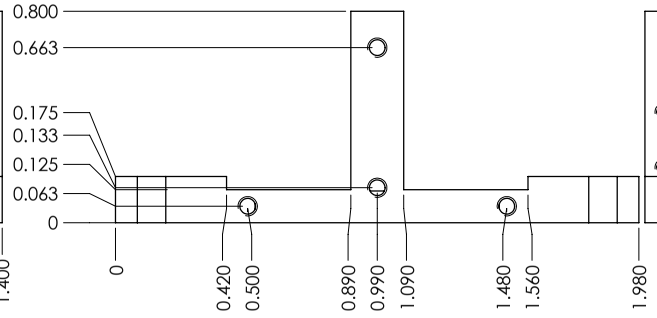
SIZE	PART NO.	REV
B	03-0002	1.1

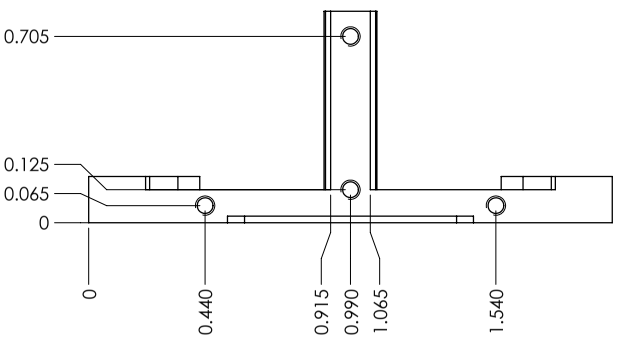
DO NOT SCALE DRAWING | SCALE: 2:1 | SHEET 1 OF 2

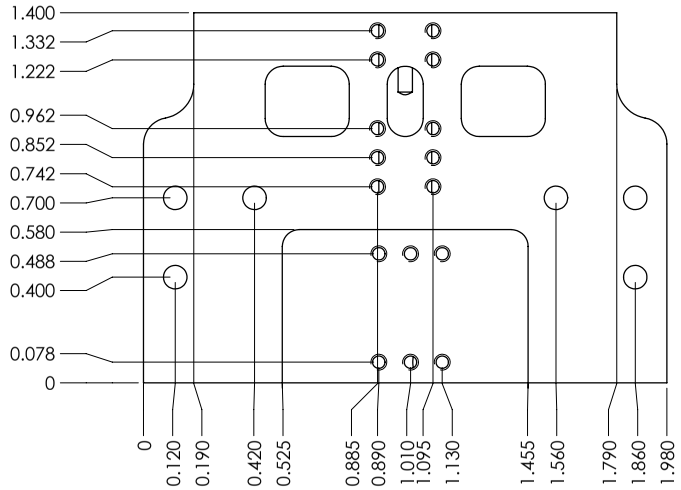








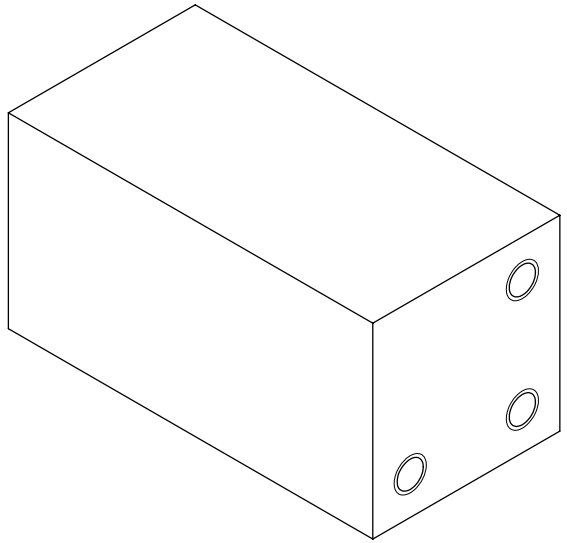


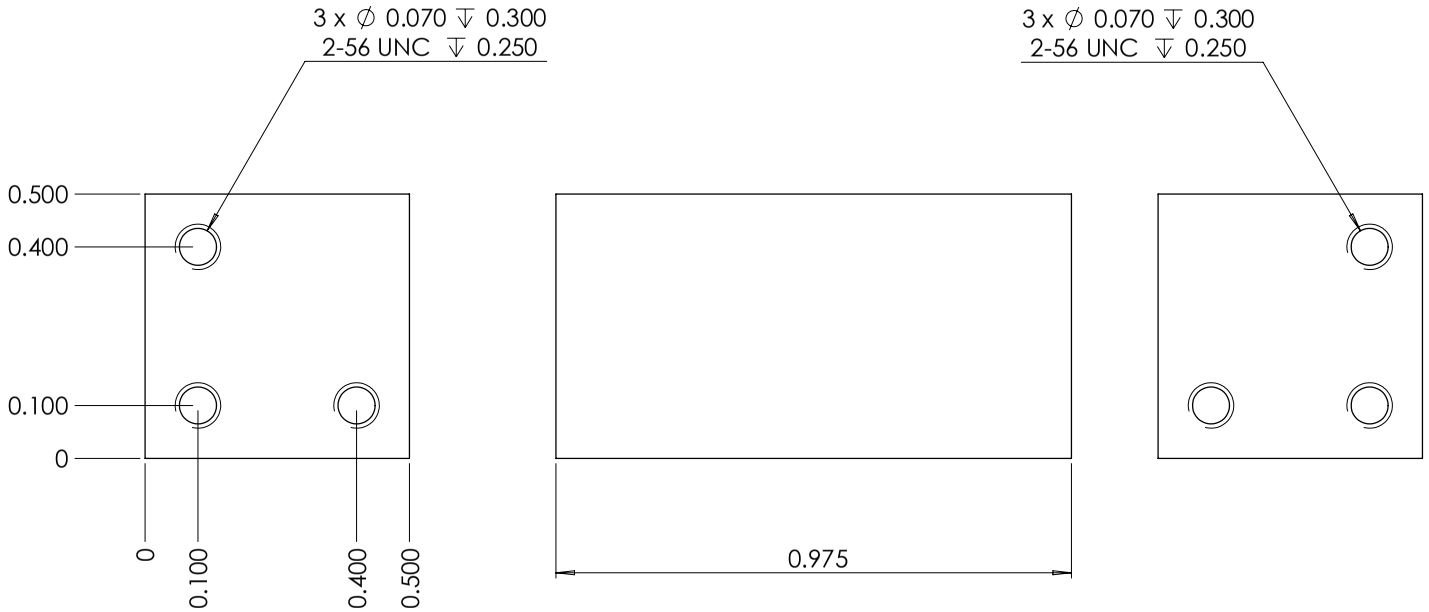


UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005" 3RD ANGLE PROJECTION	TITLE: Beam		
ENGINEER R. MEUCHEL	SIZE B	PART NO. 03-0002	REV 1.1
DATE 1/11/2016			
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE: 2:1	SHEET 2 OF 2

MATERIAL 6061-T6 (SS)	FINISH Machined

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL DESIGN	1/12/2017	R. MEUCHEL

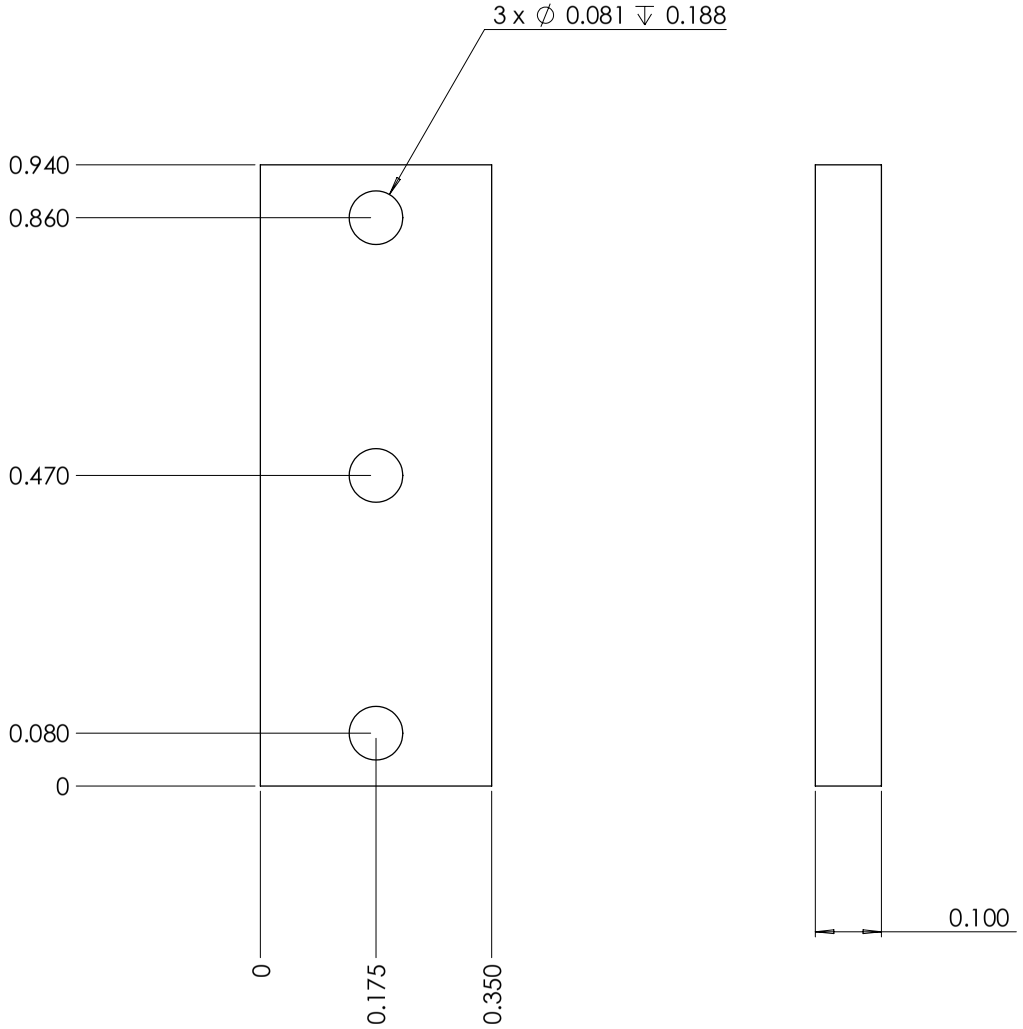


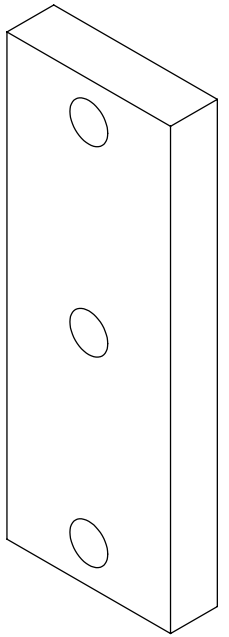


MATERIAL	6061-T6 (SS)	FINISH	Machined

UNLESS OTHERWISE SPECIFIED:				
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005"		TITLE: Post		
3RD ANGLE PROJECTION				
ENGINEER R. MEUCHEL		SIZE B	PART NO. 03-0003	REV 1.0
DATE 1/12/2017				
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE: 2:1	SHEET 1 OF 1	

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	1/13/2017	R. MEUCHEL





MATERIAL: Delrin 2700 NC010, Low Viscosity Acetal Copolymer (SS)

FINISH:

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ±0.5°
X.XX ± 0.020"
X.XXX ± 0.005"
X.XXXX ± 0.0005"

3RD ANGLE PROJECTION

ENGINEER: R. MEUCHEL

DATE: 1/13/2017

PROPRIETARY AND CONFIDENTIAL

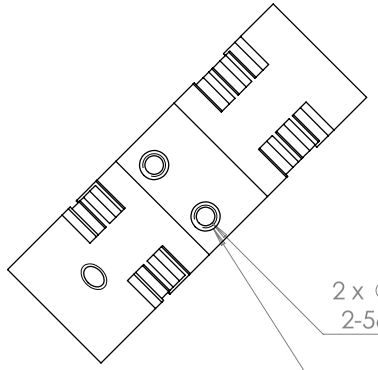
TITLE:

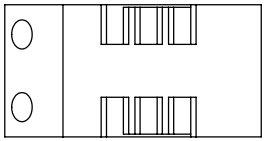
Cable Stress Relief Plate

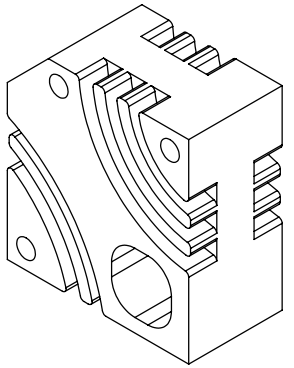
SIZE	PART NO.	REV
B	03-0004	1.0

DO NOT SCALE DRAWING | SCALE: 5:1 | SHEET 1 OF 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	3/28/2017	R. MEUCHEL



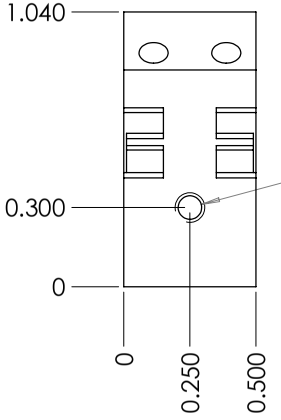




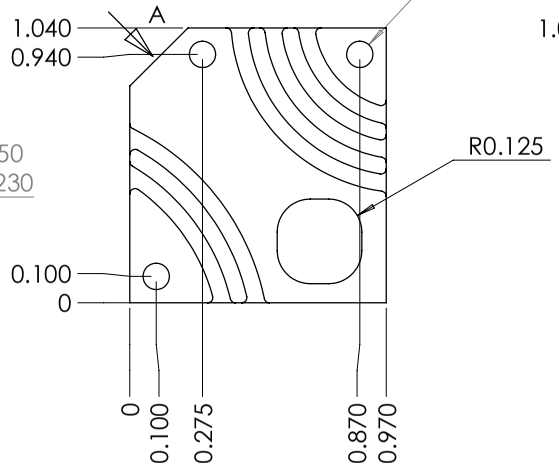
2 x ∅ 0.070 ↧ 0.850
2-56 UNC ↧ 0.850

2 x ∅ 0.110 ↧ 0.400

VIEW A

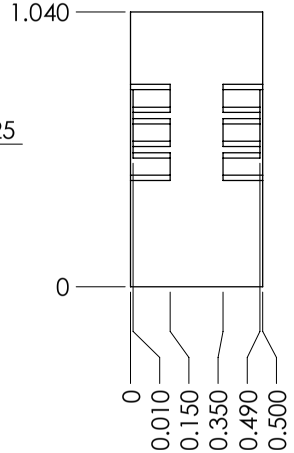


∅ 0.089 ↧ 0.250
4-40 UNC ↧ 0.230



3 x ∅ 0.0995 THRU ALL

R0.125

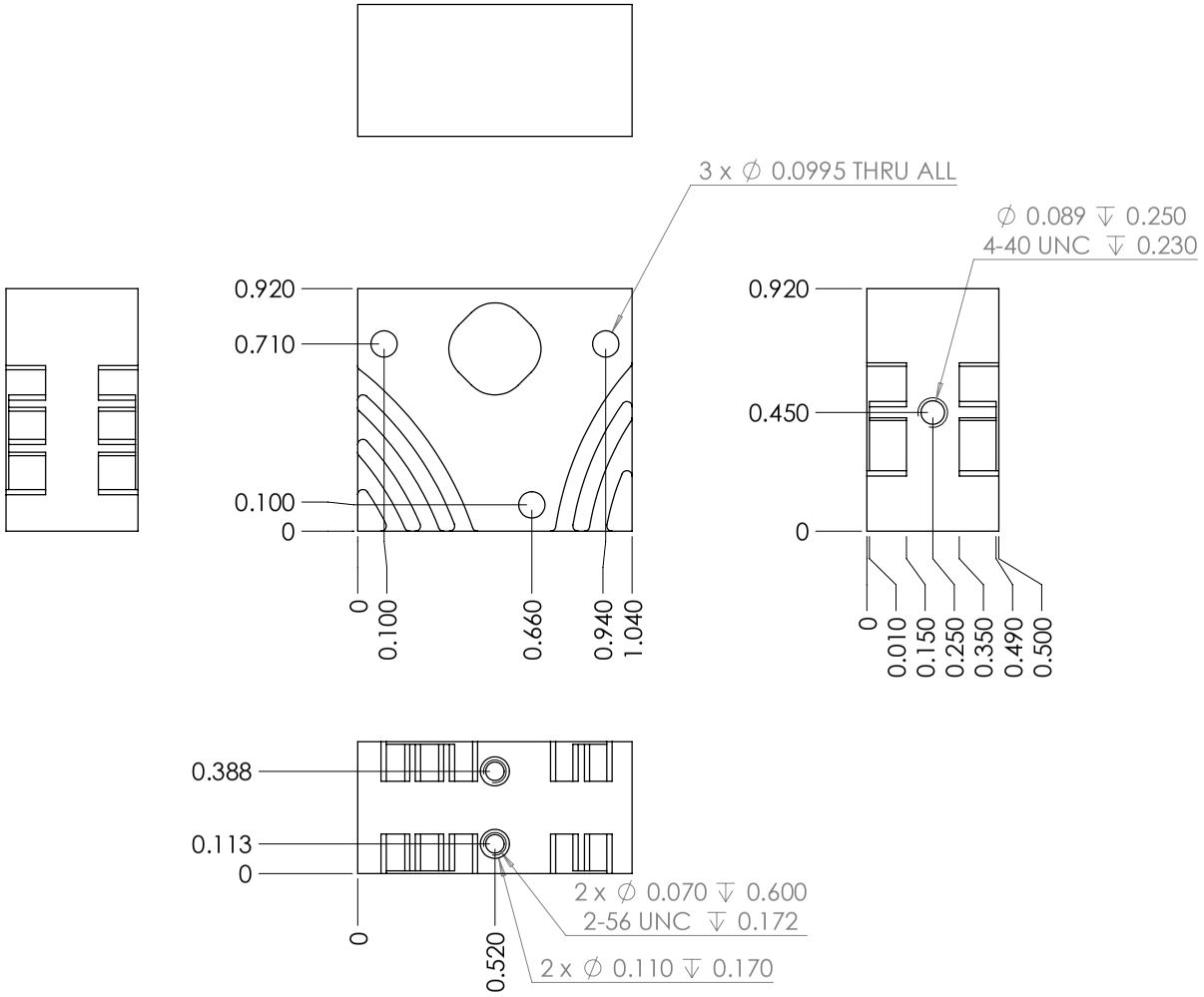


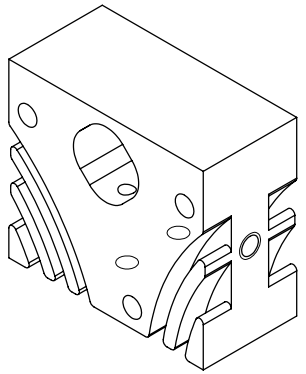
QUANTITY: 2

MATERIAL	6061-T6 (SS)
FINISH	De-Burr

UNLESS OTHERWISE SPECIFIED:	
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005"	TITLE: Left_Right Laser Holder
3RD ANGLE PROJECTION	
ENGINEER R. MEUCHEL	SIZE B / PART NO. 03-0024 / REV 1.0
DATE 3/28/2017	
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING / SCALE: 2:1 / SHEET 1 OF 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	3/28/2017	R. MEUCHEL





MATERIAL	6061-T6 (SS)	FINISH	De Burr

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ±0.5°
X.XX ± 0.020"
X.XXX ± 0.005"
X.XXXX ± 0.0005"

3RD ANGLE PROJECTION

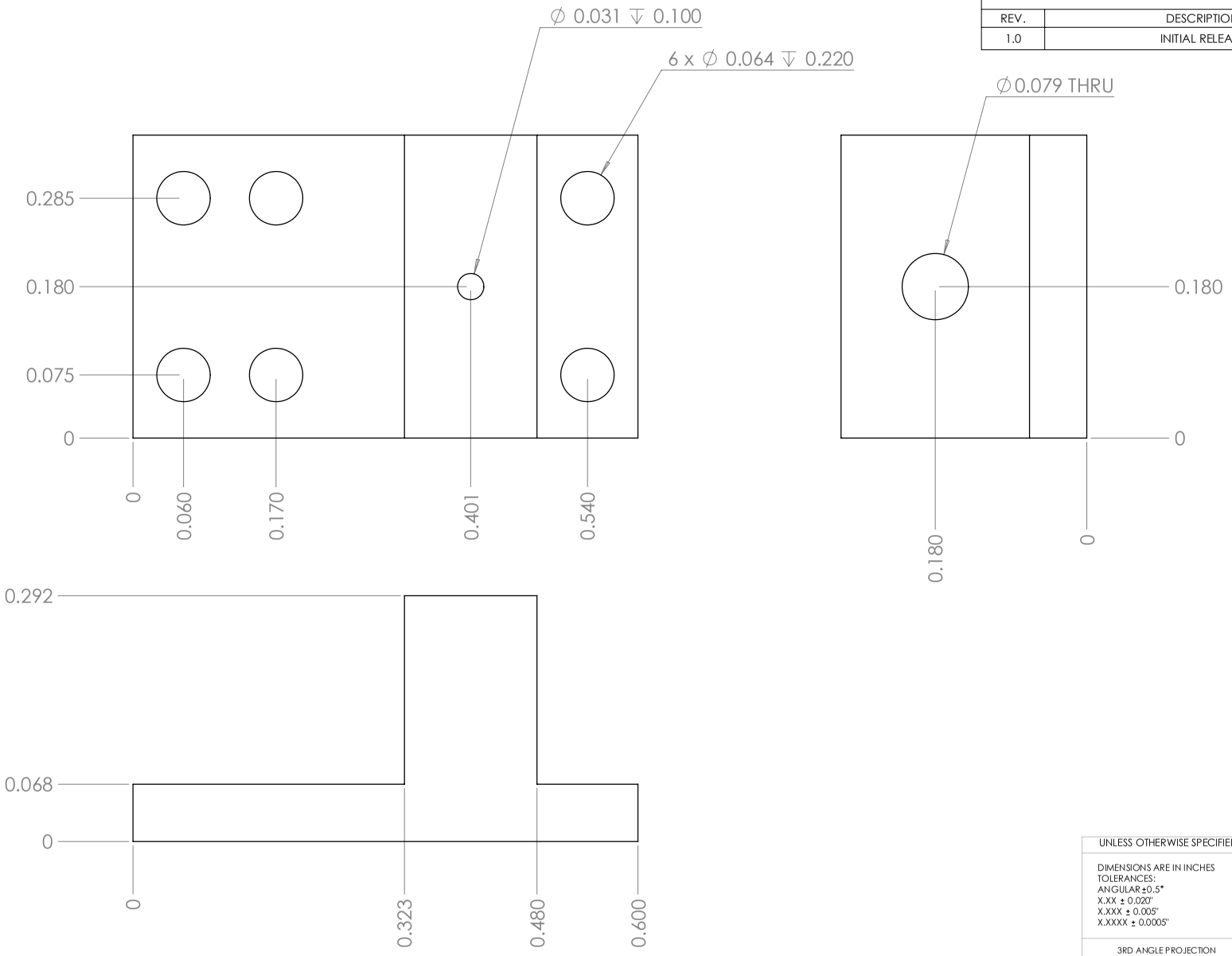
TITLE:

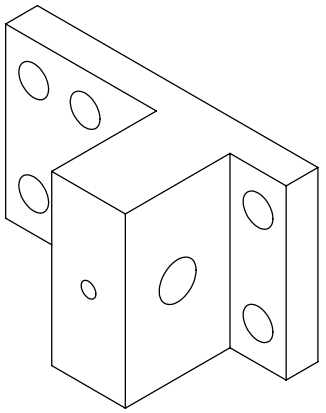
Bottom Laser Holder

ENGINEER	SIZE	PART NO.	REV
R. MEUCHEL	B	03-0025	1.0
DATE 3/28/2017			

PROPRIETARY AND CONFIDENTIAL | DO NOT SCALE DRAWING | SCALE: 2:1 | SHEET 1 OF 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	1/1/2017	





MATERIAL	6061-T6 (SS)
FINISH	De Burr

UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ±0.5°
X.XX ± 0.020"
X.XXX ± 0.005"
X.XXXX ± 0.0005"

3RD ANGLE PROJECTION

ENGINEER F. Last

DATE 1/1/2017

PROPRIETARY AND CONFIDENTIAL

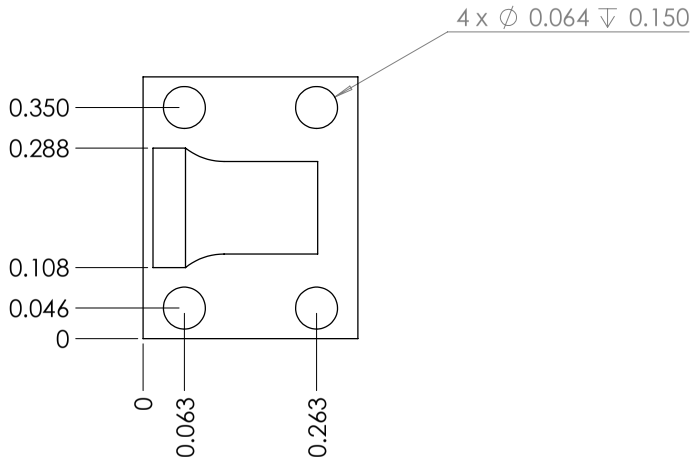
TITLE:

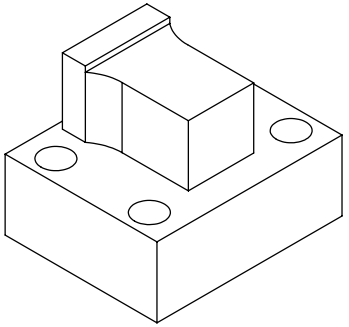
Grin Lens Holder_Side 1

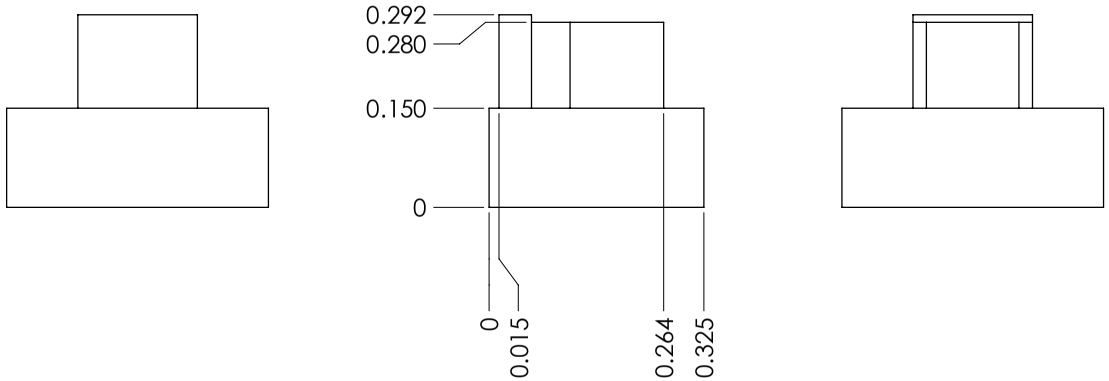
SIZE	PART NO.	REV
B	03-0026	1.0

DO NOT SCALE DRAWING | SCALE: 5:1 | SHEET 1 OF 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	3/28/2017	R. MEUCHEL





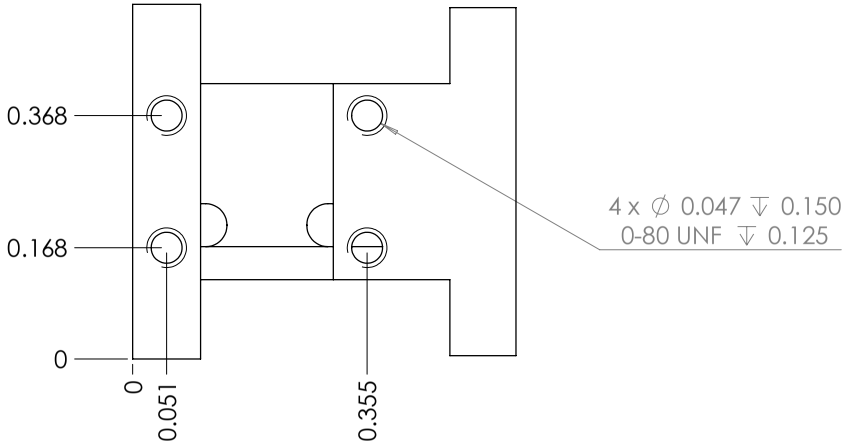


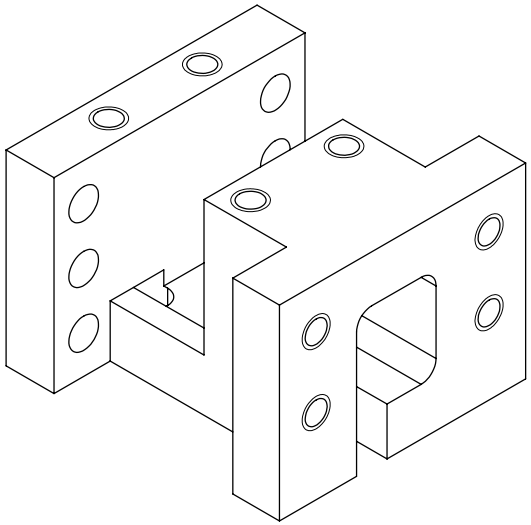
QUANTITY: 3

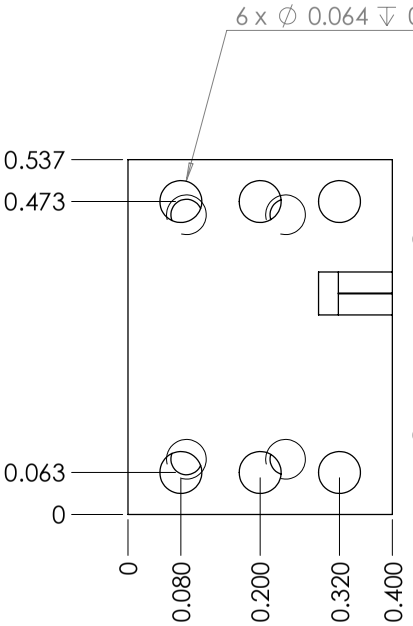
MATERIAL	6061-T6 (SS)
FINISH	De Burr

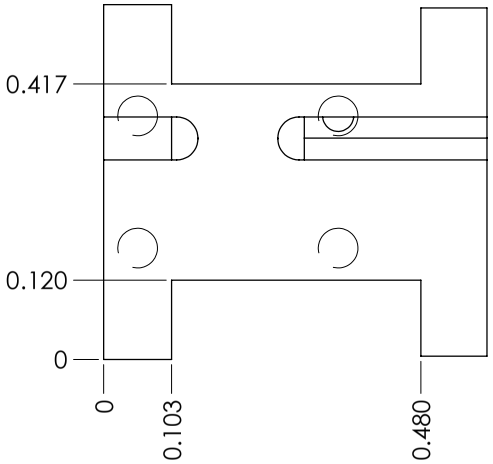
UNLESS OTHERWISE SPECIFIED:			
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005"	TITLE: 5mm Lens Holder Top		
3RD ANGLE PROJECTION			
ENGINEER R. MEUCHEL	SIZE B	PART NO. 03-0028	REV 1.0
DATE 3/28/2017			
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE: 5:1	SHEET 1 OF 1

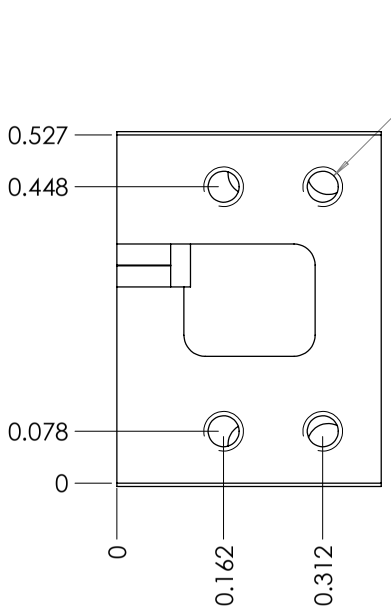
REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	3/28/2017	R. MEUCHEL







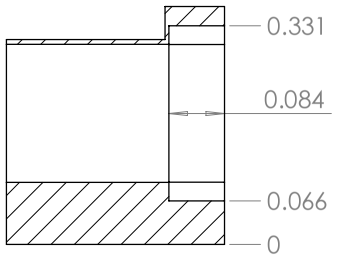




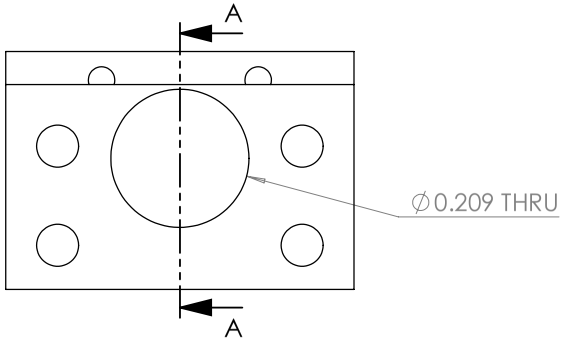
QUANTITY: 3

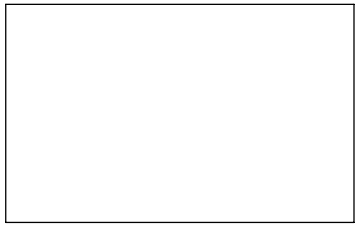
MATERIAL: 6061-T6 (SS)	FINISH: De Burr		
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005" 3RD ANGLE PROJECTION	TITLE: 5mm Lens Holder Bottom		
ENGINEER: R. MEUCHEL DATE: 3/28/2017	SIZE: B	PART NO.: 03-0029	REV: 1.0
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE: 5:1	SHEET 1 OF 1

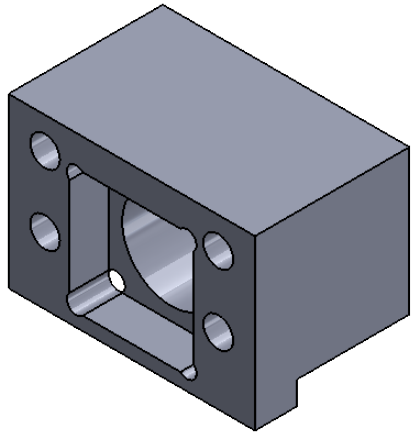
REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	1/1/2017	R. MEUCHEL

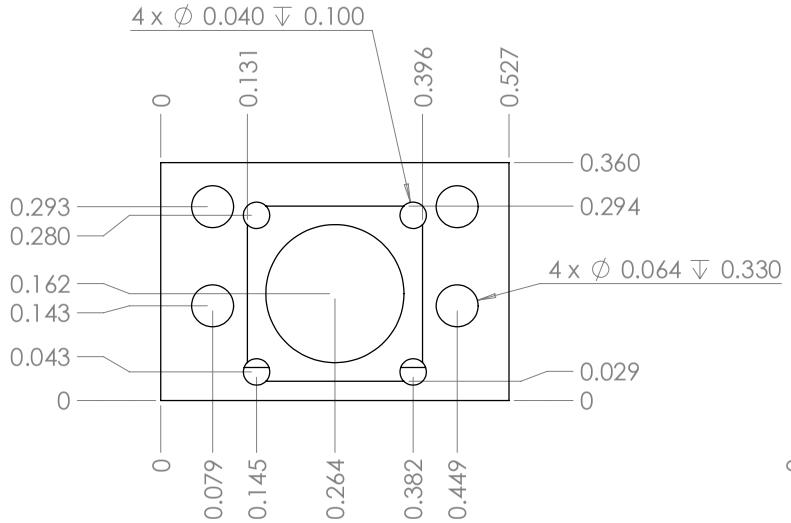


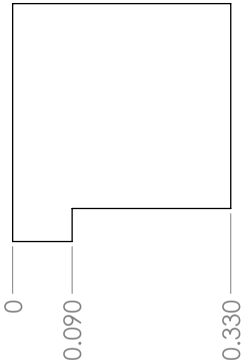
SECTION A-A









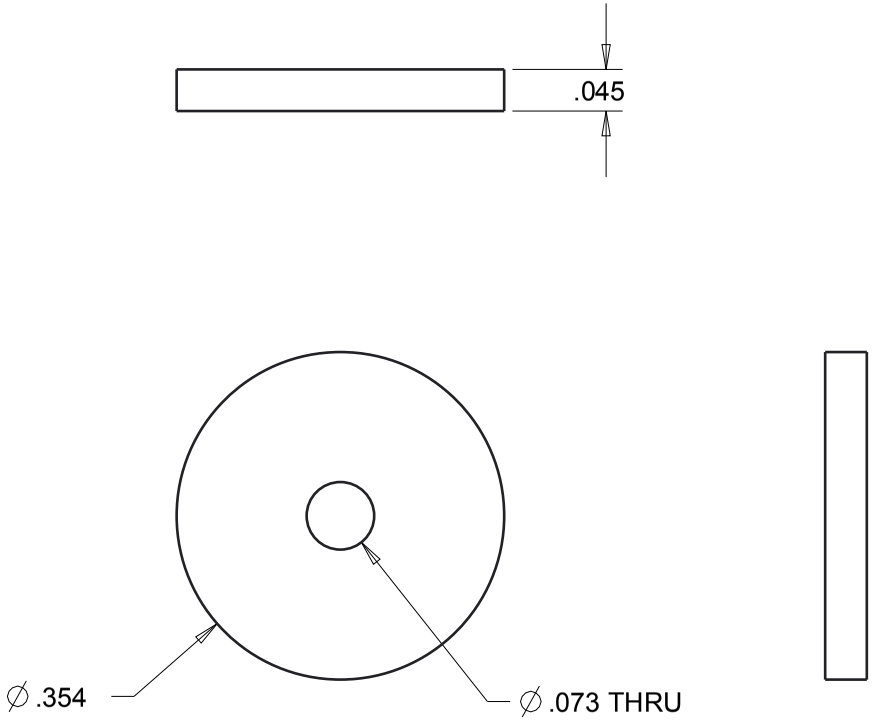


UNLESS OTHERWISE SPECIFIED:	
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ±0.5° X.XX ± 0.020" X.XXX ± 0.005" X.XXXX ± 0.0005"	TITLE: Spacer_Glass Holder
3RD ANGLE PROJECTION	
ENGINEER R. MEUCHEL	SIZE B / PART NO. 03-0043 / REV 1.0
DATE 12/5/2017	
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING / SCALE: 5:1 / SHEET 1 OF 1

MATERIAL Material <not specified>

FINISH

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	Original Rlease	8/8/17	M. Johnson
1.1			
1.2			



UNLESS OTHERWISE SPECIFIED:

DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0.5°
X.XX ± 0.020
X.XXX ± 0.005
X.XXXX ± 0.0005

3RD ANGLE PROJECTION

TITLE: Third_Apet

ENGINEER: M. Johnson

DATE: 8/8/17

SIZE: B

PART NO.

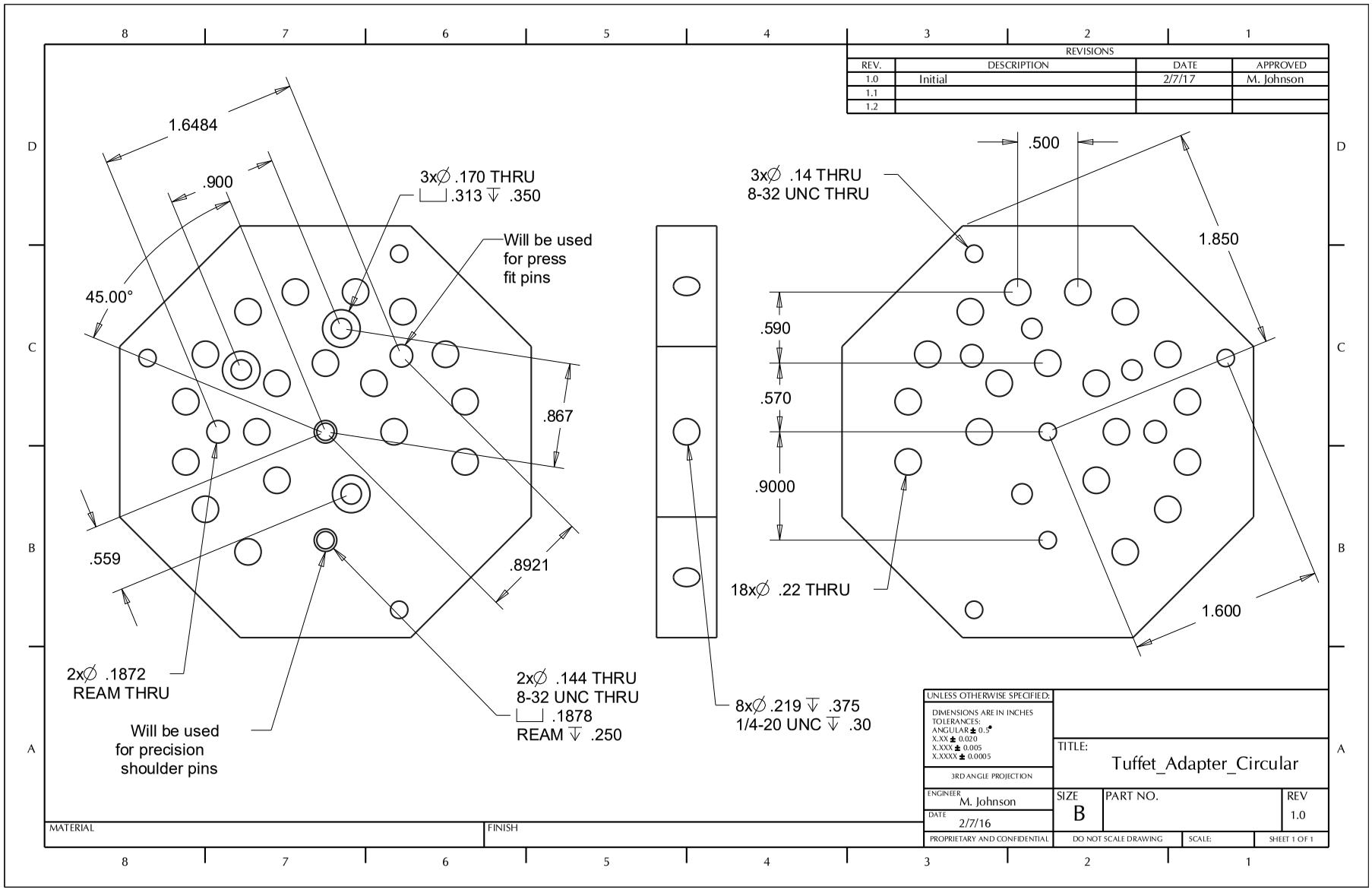
REV: 1.0

MATERIAL: Aluminum 5056

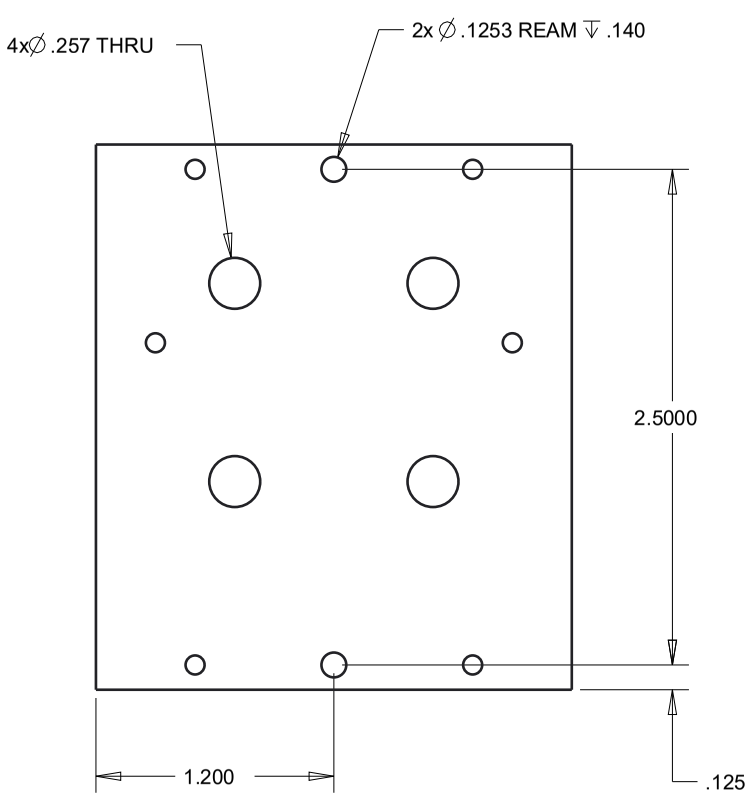
FINISH: Black Anodize

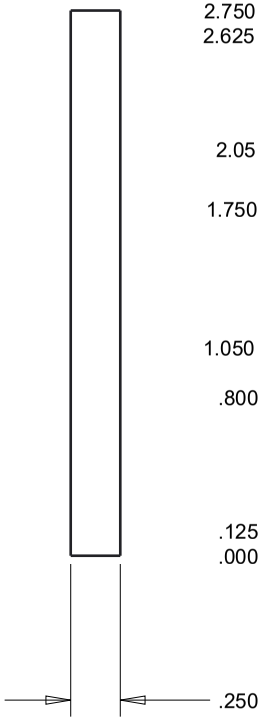
PROPRIETARY AND CONFIDENTIAL | DO NOT SCALE DRAWING | SCALE: 7.0 | SHEET 1 OF 1

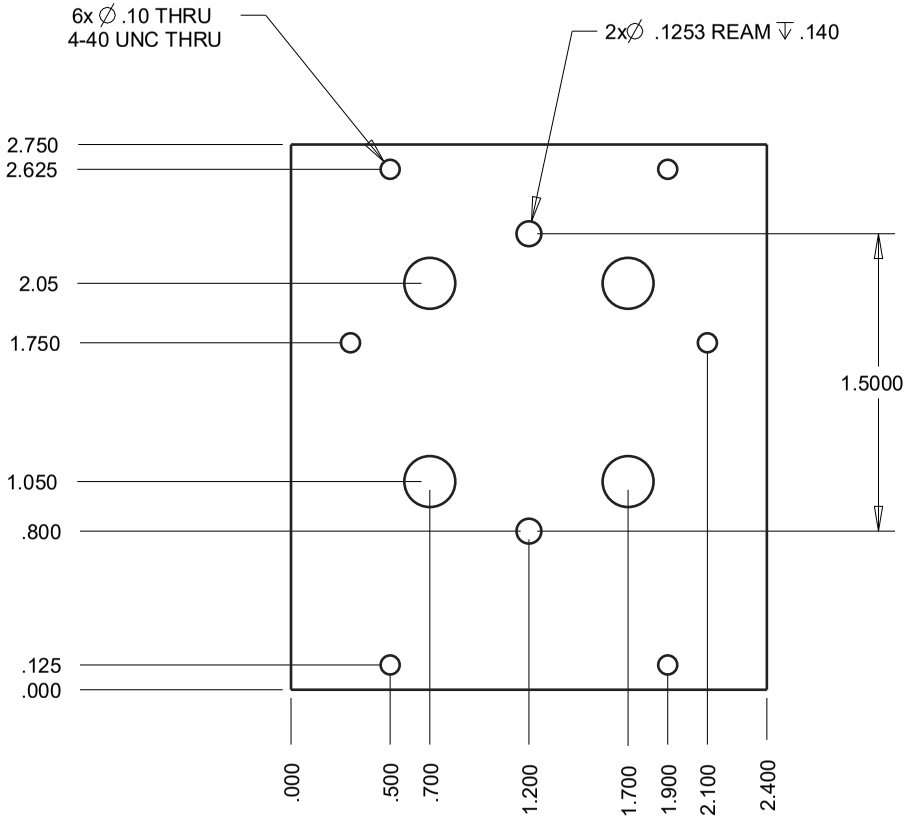
External Buildup



REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	INITIAL RELEASE	2/2/17	M. Johnson
1.1			
1.2			





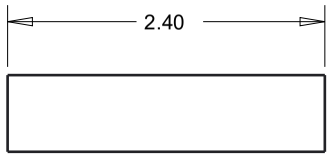


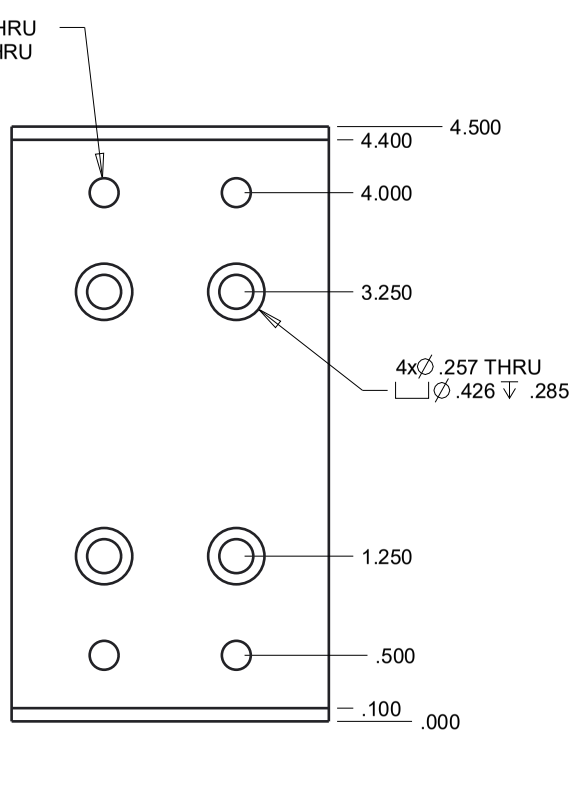
SCALE 1.500

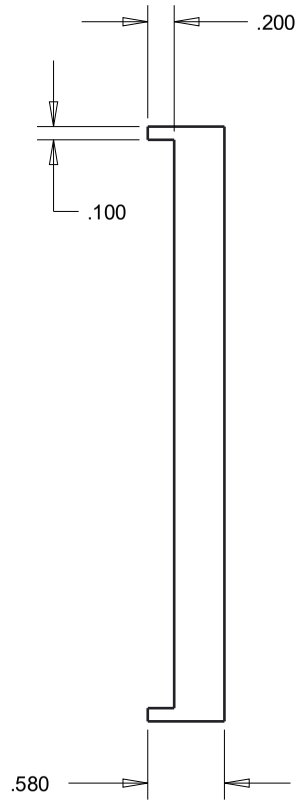
UNLESS OTHERWISE SPECIFIED:	
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ± 0.5° X.XX ± 0.020 X.XXX ± 0.005 X.XXXX ± 0.0005	TITLE: Top_Mot_Stage_Adp
3RD ANGLE PROJECTION	
ENGINEER M. Johnson	SIZE B / PART NO. / REV 1.0
DATE 2/2/17	
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING / SCALE: / SHEET 1 OF 1

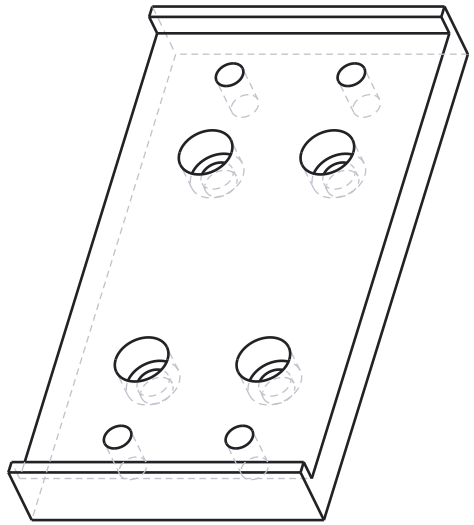
MATERIAL	FINISH

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	Initial	2/2/17	M. Johnson
1.1			
1.2			





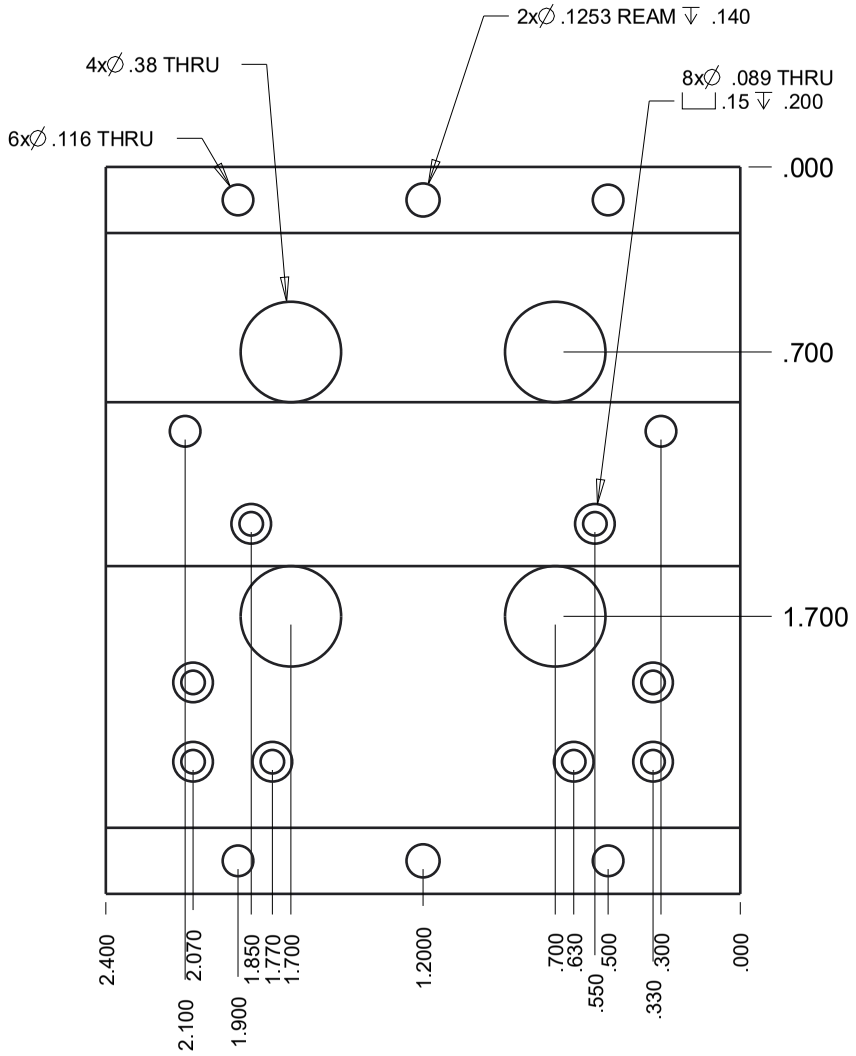


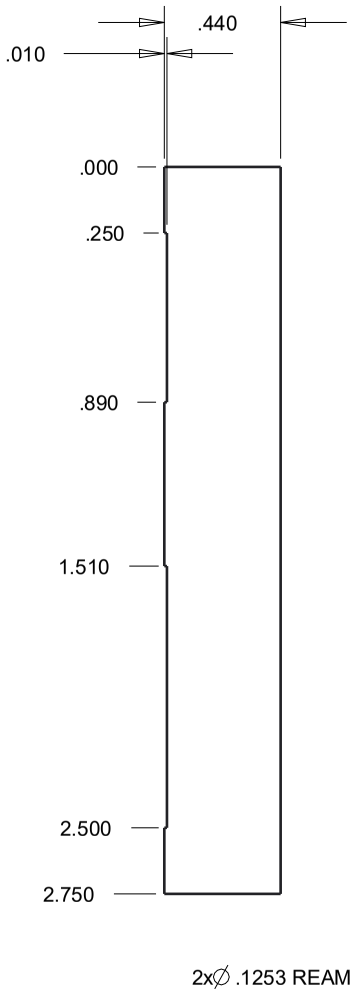


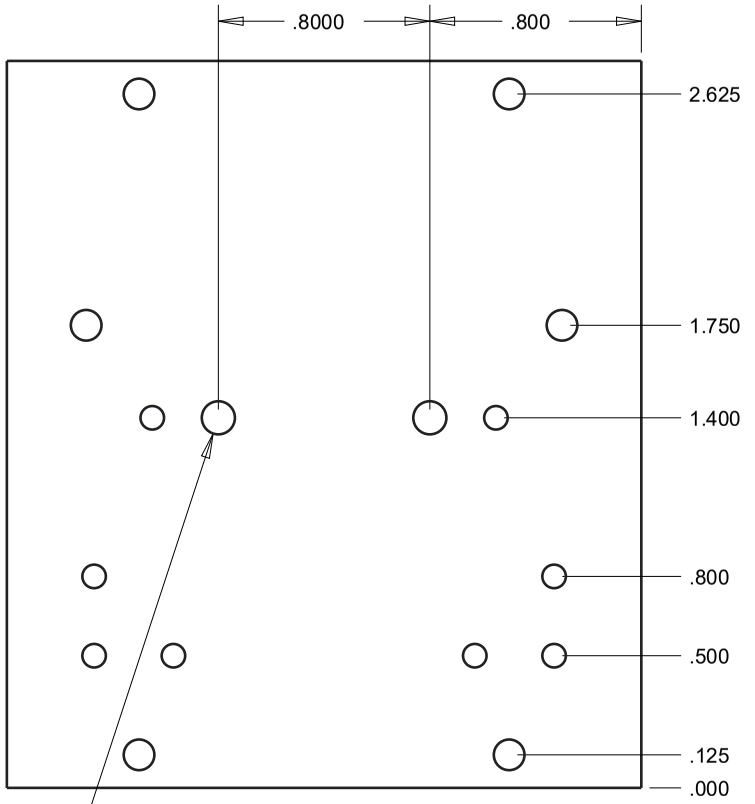
UNLESS OTHERWISE SPECIFIED:			
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ± 0.5° X.XX ± 0.020 X.XXX ± 0.005 X.XXXX ± 0.0005		TITLE: Mot_Man_Stage_Adp	
3RD ANGLE PROJECTION			
ENGINEER M. Johnson	SIZE B	PART NO.	REV
DATE 2/2/17			
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE:	SHEET 1 OF 1

MATERIAL	FINISH

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	Initial	2/2/17	M. Johnson
1.1			
1.2			







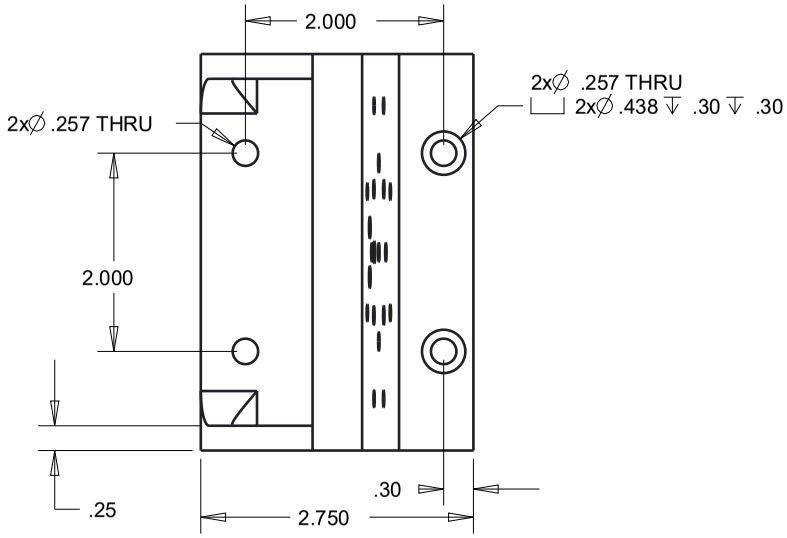
Dimentionas to 4 decimal places
are for slip pinned interfaces

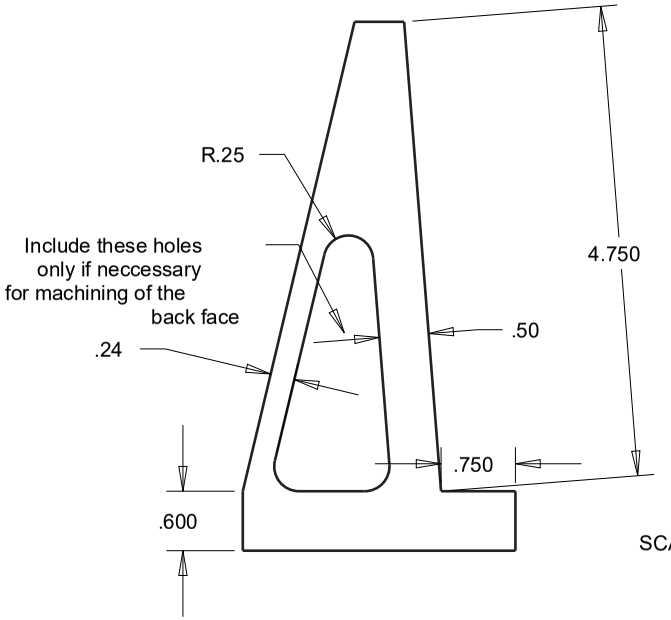
SCALE 2.000

MATERIAL	
FINISH	

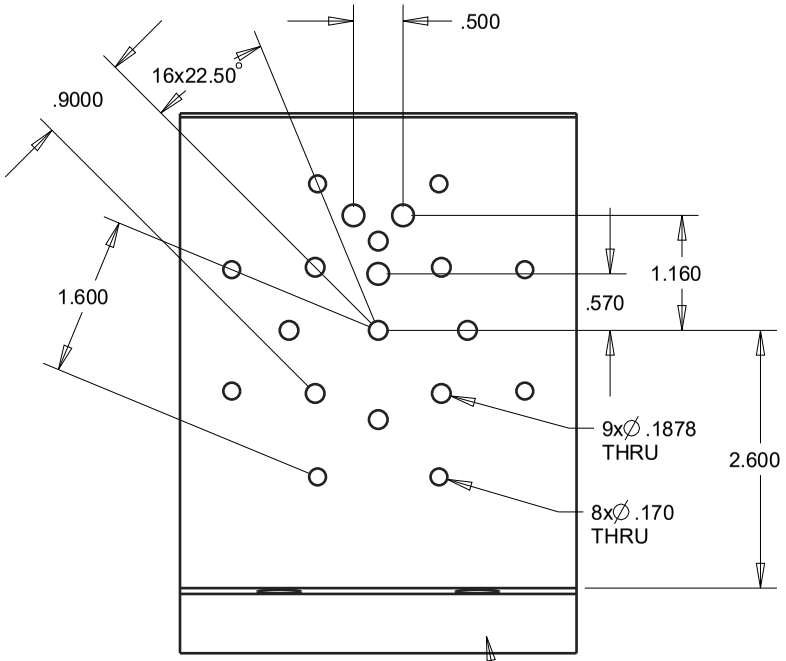
UNLESS OTHERWISE SPECIFIED:			
DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ± 0.5° X.XX ± 0.020 X.XXX ± 0.005 X.XXXX ± 0.0005		TITLE: Microscope_Holder	
3RD ANGLE PROJECTION			
ENGINEER M. Johnson	SIZE B	PART NO.	REV 1.0
DATE 2/2/17			
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE:	SHEET 1 OF 1

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	Initial	2/6/17	M. Johnson
1.1			
1.2			





SCALE 0.750



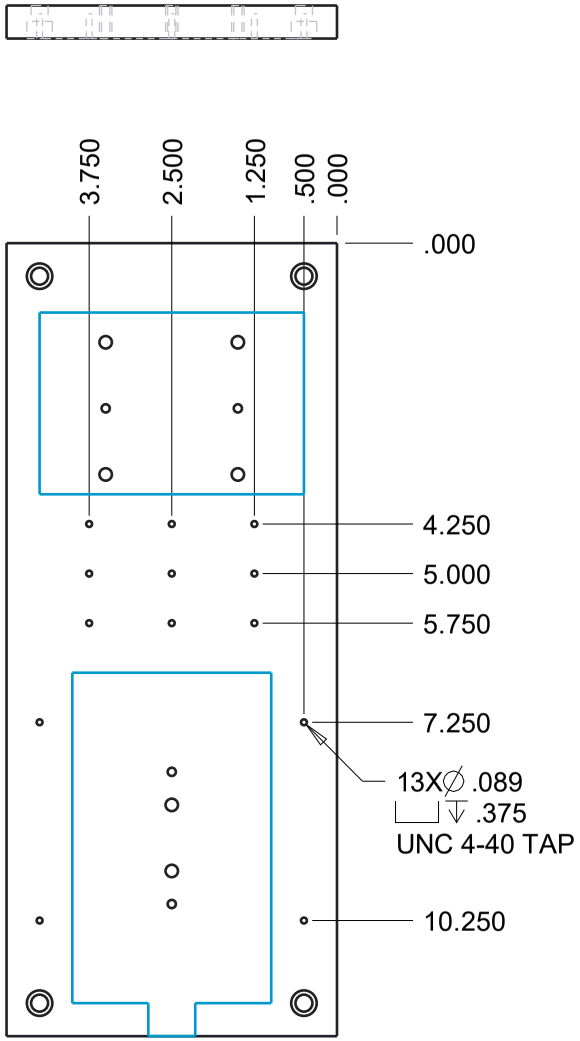
SCALE 0.750

This view is rotated so the angled face is parallel to drawing surface

UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN INCHES TOLERANCES: ANGULAR ± 0.5° X.XX ± 0.020 X.XXX ± 0.005 X.XXXX ± 0.0005		TITLE: Conf_Grat_Surf2	
3RD ANGLE PROJECTION			
ENGINEER M. Johnson	SIZE B	PART NO.	REV 1.0
DATE 2/6/17			
PROPRIETARY AND CONFIDENTIAL	DO NOT SCALE DRAWING	SCALE:	SHEET 1 OF 1

MATERIAL	FINISH

Original Release 8/10/17 M. Johnson



SCALE 0.500

Tab_Adap

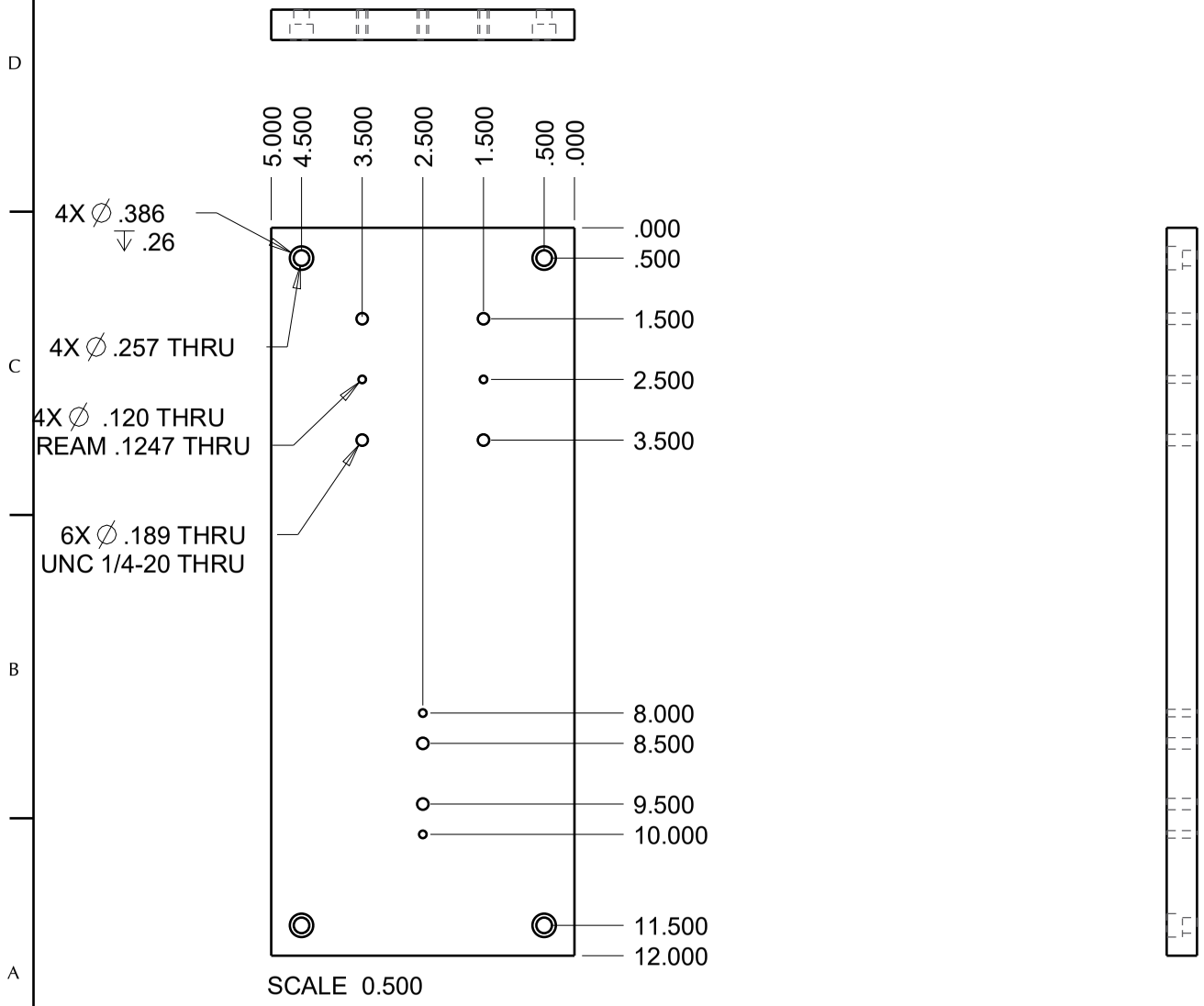
M. Johnson
8/10/17

1.0

Aluminum 5056

Deburr

REVISIONS			
REV.	DESCRIPTION	DATE	APPROVED
1.0	Original Release	8/10/17	M. Johnson
1.1			
1.2			



UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0.5°
X.XX ± 0.020
X.XXX ± 0.005
X.XXXX ± 0.0005
3RD ANGLE PROJECTION

TITLE: Tab_Adap

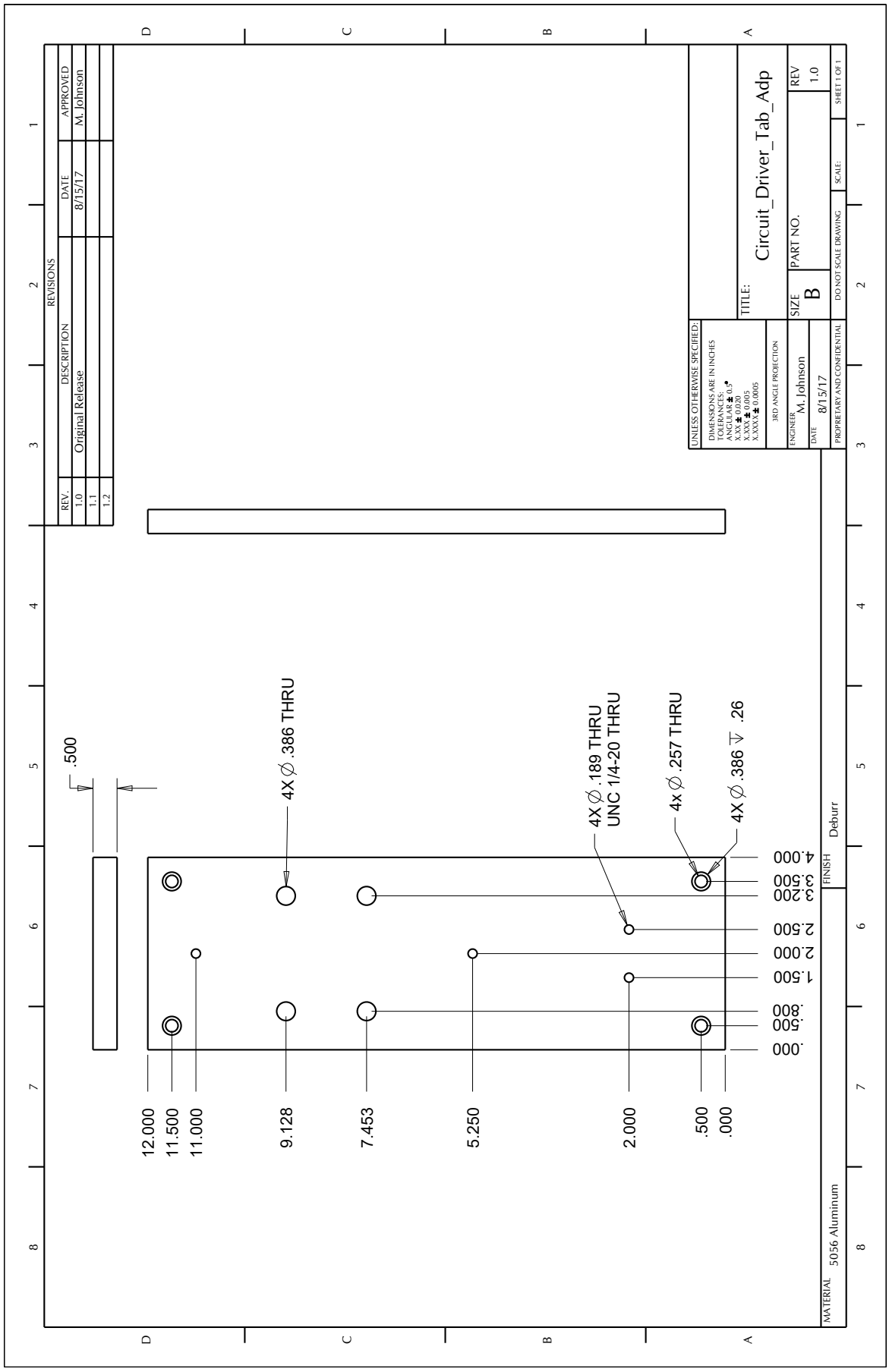
ENGINEER: M. Johnson
DATE: 8/10/17

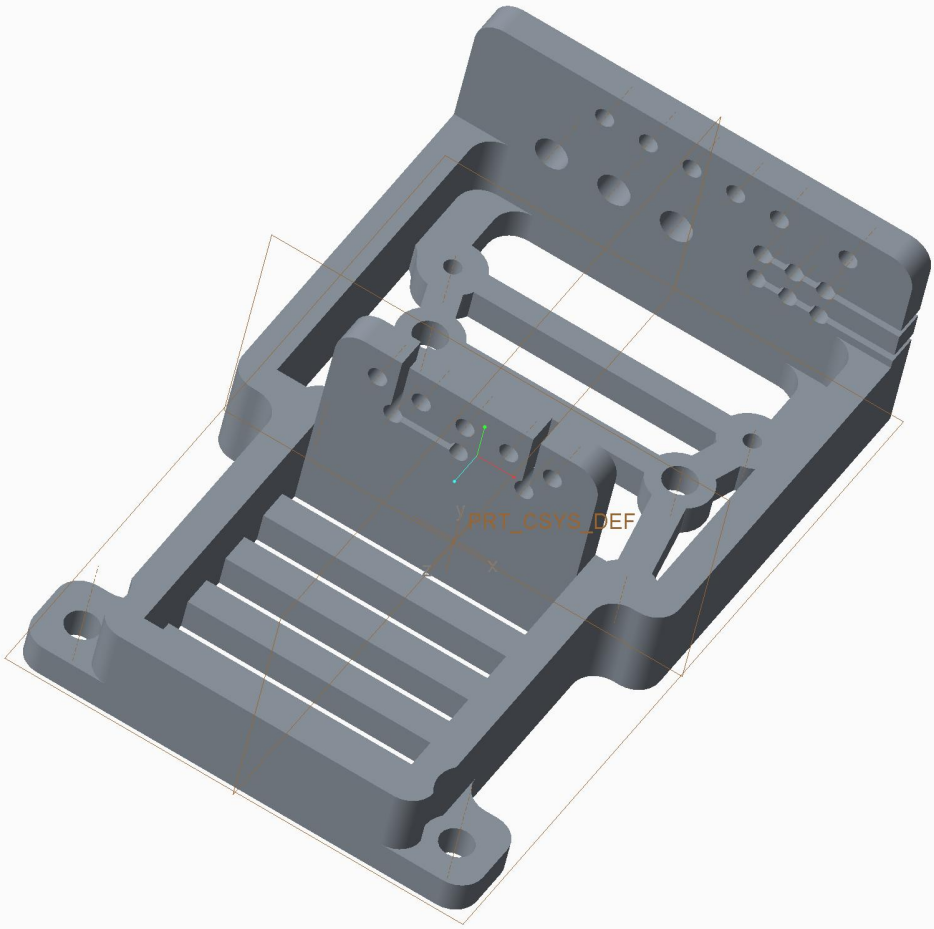
SIZE: B | PART NO. | REV: 1.0

MATERIAL: Aluminum 5056 | FINISH: Deburr

PROPRIETARY AND CONFIDENTIAL | DO NOT SCALE DRAWING | SCALE: | SHEET 1 OF 1

Electrical


REVISIONS
REV: DESCRIPTION DATE APPROVED
1.0 Original Release 8/15/17 M. Johnson
1.1
1.2
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
ANGULAR ± 0.5°
X.XX ± 0.020
X.XXX ± 0.005
X.XXXX ± 0.0005
3RD ANGLE PROJECTION
ENGINEER M. Johnson
DATE 8/15/17
TITLE: Circuit_Driver_Tab_Adp
SIZE B
PART NO.
REV 1.0
DO NOT SCALE DRAWING
SCALE:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
MATERIAL 5056 Aluminum
FINISH Deburr
.500
4X ∅ .386 THRU
4X ∅ .189 THRU
UNC 1/4-20 THRU
4x ∅ .257 THRU
4X ∅ .386 ↧ .26
4.000
3.500
3.200
2.500
2.000
1.500
.800
.500
.000
12.000
11.500
11.000
9.128
7.453
5.250
2.000
.500
.000


PRT_CSYS_DEF

Examples

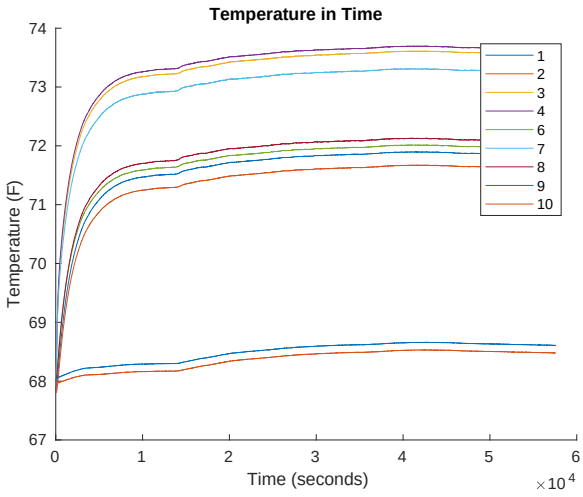


Figure 1: The temperature of each of the thermistor channels over 16 hours. The Lasers were turned on in the first 10 seconds of plotting. TEA was in a box of aluminum foil taped to the table throughout the duration of the test.

VIS-EUV Test Report

Wednesday 30th August, 2017, 14:32

Grating Type:VIS
Grating Number:1

We are 98% confident that the peak intensity...
...for channel 1 lies at 4.2401 mm to a precision of 952.5 microns with a target of 15.3 micron.
The final model was created...
...after 4 intensity scans.
...with 5 measurements taken at each distance.
...with data from 3465 distances.

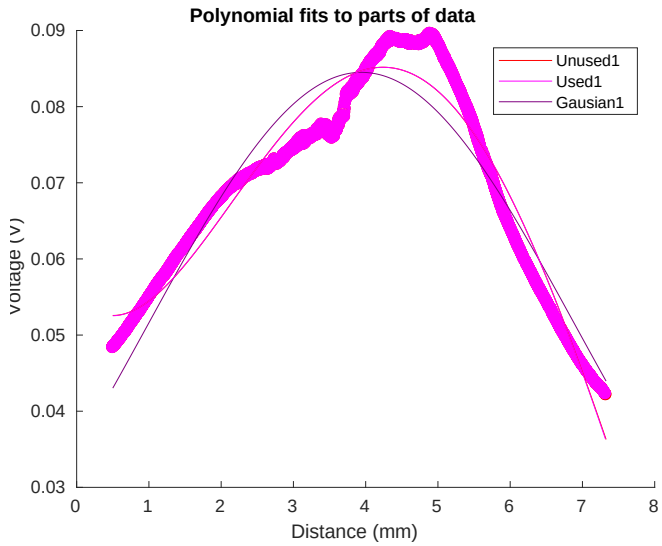


Figure 2: An example test report for a single channel being read. A test report for all three channels will contain the same information.

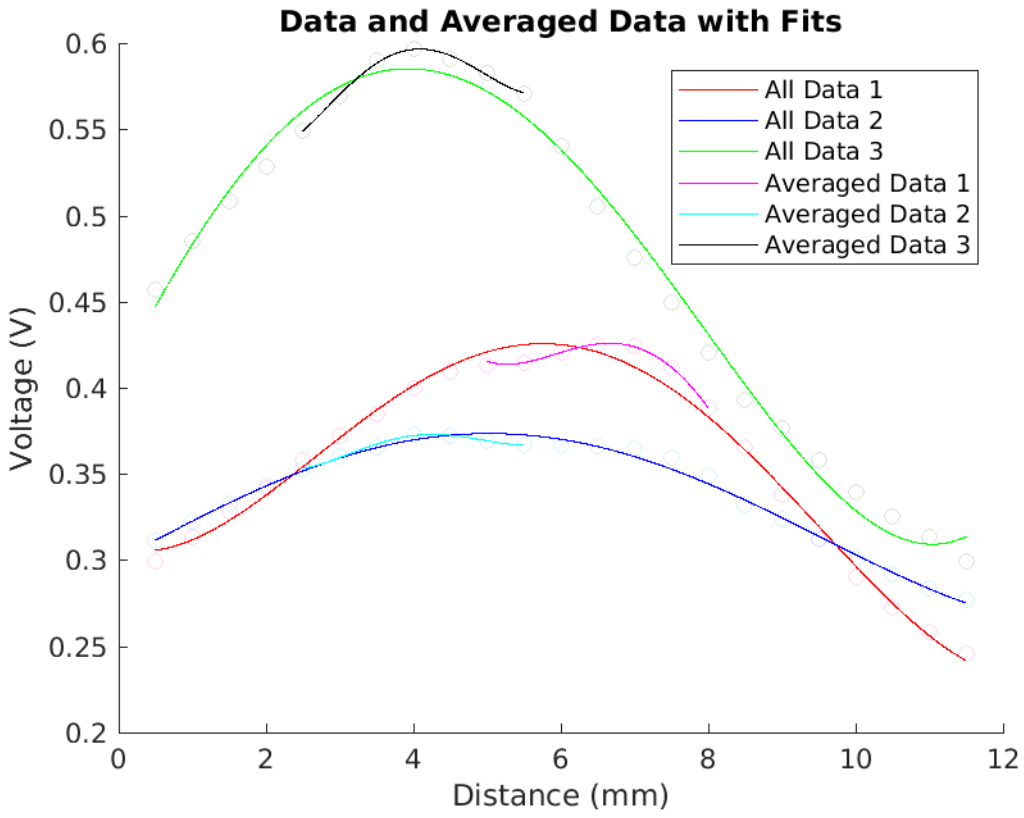


Figure 3: This is the type of plot that should be created when a test is run. Work will continue to sharpen the intensity peaks.

Troubleshooting

1. Q — The programs are throwing unexpected errors, and the K-cube has stopped responding

   A — Turn the Agilent and the K-cube off and unplug them from the computer. Make sure the computer is turned on, and logged in. First plug the Agilent USB in. Turn the Agilent on. Wait a few seconds and plug the K-cube into the computer. Turn the K-cube on. This should reset the way the computer reads from the USB's

2. Q — The power button of the K-cube has stopped working.

   A — Unplug the K-cube, turn off the cube, plug it in, and turn it on. It tends to do then when the USB order has been messed up, so see the above answer.

3. Q — The program has stopped running after the first iteration

   A — It is likely that one of the photodiode channel is is not receiving enough light, and checking the graph should reveal that the curvature of one of channels is not very strong. This will mess with the adaptive grid algorithm. Provide more light to the channel in question, and the algorithm should be able to work properly.

Improvements

- Place an opto-isolator between the beam splitter and laser to prevent feedback into the laser to change intensity and laser modes.
- Connect an oscilloscope in parallel with a laser to check the stability of the laser is the system.
- Use a block machined to a greater precision than of the stepper motor to zero the motor before every test.
- Make all parts of titanium to prevent thermal fluctuations from influencing buildup.
- Test different diffusing glass between the beam splitter and photodiode to increase sensitivity.

- Replace amplifying resistor in photodiode circuit to increase gain and voltage output.
- Wire directly into Agilent card.
- Stop down aperture between the beam splitter and laser so the entire lens can be used.
- Create a system for organizing the thermistor wires.

Images