

The Caltech Gravity Wave Detector Master Allignment Control Program

Programmed by Brian Elliot Lemoff
as part of his 1987 SURF project

```

option base 1
defint a-z
dim shared hthresh!(2),hrthresh(4),green(2),glob(4),track(4),contr!(2)
dim shared lthresh!(2),lrthresh(4),gauto(2),rauto(4),mass$(4),bias(4,2)
dim shared intens(4),ip(2),g!(2),gout(4,2),rout(4,2),digit(4,2)
common shared see,con,il

* hthresh!(x) is the contrast on arm x where red/green switchover occurs
* lthresh!(x) is the contrast on arm x where green/red switchover occurs
* hrthresh(y) is diode intensity where local/global switchover occurs
* lrthresh(y) is diode intensity where global/local switchover occurs
* green(x) is the control flag for arm x: 1=green 0=red
* glob(y) is the red control flag for mass y: 1=global 0=local
* track(y) is the tracking flag for mass y: 1=tracking 0=no tracking
* gauto(x) is the red/green auto control flag: 1=auto on 0=auto off
* rauto(y) is the local/global auto control flag: 1=on 0=off
* con indicates whether or not the control loop is on at all
* bias(y,z) is the bias on mass y dimension z global UDT input
* contr!(x) is the contrast on arm x
* inten(y) is the intensity on global photodiode y
* il is the laser intensity as measured by the laser meter
* ip(x) is the light intensity on the arm x photodiode
* g!(x) is the contrast normalization factor: contr=(il - g*ip)/il

randomize timer
def fnarm(y) = int((y-1)/2)+1
def fnl$(x)
  if x=1 then fnl$="on " else fnl$="off"
end def
def fnc$(y)
  if green(fnarm(y))=1 then fnc$="green " : exit def
  if glob(y)=1 then fnc$="global":exit def
  fnc$="local "
end def
def fnr$(i)
  if green(i)=1 then fnr$="Resonating " else fnr$="Out of Lock"
end def
def fni(y,z) = 10^digit(y,z)
call default      'Get default values for all parameters
call main        'Begin main control loop
call finish      'Save new parameter values
end

sub default static

open "c:\default.dat" for input as #1
input #1, hthresh!(1),hthresh!(2),lthresh!(1),lthresh!(2)
input #1, gauto(1),gauto(2)
for i=1 to 4 : input #1, hrthresh(i),lrthresh(i),track(i),rauto(i)
glob(i)=1 : next i
green(1)=0 : green(2)=0 : for y=1 to 4:intens(y)=3000:next y
con = 0
close #1
g!(1) =.5 'For the sake of argument
g!(2) =.5 'For the sake of argument
see = 900

```

```
gout(y,z) = int(201*rnd)-100
next z
next y
```

```
mass$(1)="Huey" : mass$(2)="Huey Newie"
mass$(3)="Louie Newie" : mass$(4)="Louie"
end sub
```

```
sub finish static
open "c:\default.dat" for output as #1
print #1, hthresh!(1),hthresh!(2),lthresh!(1),lthresh!(2)
print #1, gauto(1),gauto(2)
for i=1 to 4 : print #1, hrthresh(i),lrthresh(i),track(i),rauto(i)
next i
close #1
end sub
```

```
sub title static
locate 1,18,0
print "Mass Orientation Control Program Version 1.0"
locate 4,20 : print "Type number of option:"
end sub
```

```
'*****
```

```
sub main static
```

```
Begin:  cls
        call title
        locate 6,25
        print "1. Begin control loop" : locate 8,25
        print "2. End control loop" : locate 10,25
        print "3. Manual control options" : locate 12,25
        print "4. Adjust parameters" : locate 14,25
        print "5. View status" : locate 16,25
        print "6. Normalize Contrast" : locate 18,25
        print "7. Return to system"
```

```
Kloop:  call corner : if con=1 then call update
        A$ = inkey$
        if A$="" then goto Kloop
        if A$="1" then con=1 : goto Kloop
        if A$="2" then con=0 : goto Kloop
        if A$="3" then call manual : goto Begin
        if A$="4" then call adjust : goto Begin
        if A$="5" then call status : goto Begin
        if A$="6" then call renorm : goto Begin
        if A$="7" then exit sub
        goto Kloop
```

```
end sub
```

```
sub update static          'The actual control program
```

```
if gauto(1)=1 then call gauto(1)
if gauto(2)=1 then call gauto(2)
for y = 1 to 4
  if green(fnarm(y))=0 and rauto(y)=1 then call rauto(y)
next y
end sub
```

```
sub manual static
```

```
Begin:  cls
```

```

print "Y-bias:";bias(y,2) : locate 15,c
print "Tracking: ";fnl$(track(y)) : locate 16,c
print "Red Intens:";intens(y) : locate 17,c
print "Low thresh:";lrthresh(y) : locate 18,c
print "High thresh:";hrthresh(y) : locate 19,c
print "Red Auto Switch:";fnl$(rauto(y))
next y
locate 25,5 : print "Press <RETURN> to return to main menu";
Klop3: if con=1 then call update
call contrast(1) : call contrast(2)
locate 6,19 : print using "###.##";contr!(1)
locate 6,59 : print using "###.##";contr!(2)

A$=inkey$
if A$=chr$(13) then see=0 : exit sub : else goto Klop3
end sub

sub renorm static 'Determines new normalization constant g!
cls
call title
locate 6,25
print "1. Normalize arm 1 contrast" : locate 8,25
print "2. Normalize arm 2 contrast" : locate 10,25
print "3. Return to previous menu" : locate 14,15
print "Do not type 1 or 2 unless that arm is out of lock!"
Klop4: call corner : if con=1 then call update
A$ = inkey$
if A$="" then goto Klop4
if A$="3" then exit sub
if A$<>"1" and A$<>"2" then goto Klop4
x=val(A$) : call getip(x) : call getil
g!(x) = il/ip(x)

beep
goto Klop4
end sub

sub bias(y,z,d) static 'Changes global servo bias
if bias(y,z)+d >2047 or bias(y,z)+d < -2047 then beep : exit sub
bias(y,z)=bias(y,z)+d
'output through DAC
end sub

sub contrast(x) static 'Determines arm x photodiode contrast
call getip(x) : call getil(x)
contr!(x) = ( il - g!(x)*ip(x) ) / il * 100
end sub

sub intensity(y) static 'Reads global photodiode intensity
'get intens(y) from ADC
intens(y)=3000 'For the sake of argument
end sub

sub rsw(y,f) static 'Switches local/global control on mass y
glob(y)=f
'output to switch through Digital I/O on DAC board
end sub

sub sw(x,f) static 'Switches red/green control for arm x
green(x)=f
'output to switch through Digital I/O on DAC board
end sub

sub gauto(x) static 'Auto allignment routine

```

```

    if z! >= lrthresh(y) and z! <= 4095 then hrthresh(y)=int(z!) else beep
    goto rplth
end if
if A$="7" then exit sub
goto klop22
end sub

sub flags static
cls
locate 1,18,0 : print "Mass Orientation Control Program Version 1.0"
flg: locate 5,23
print " Red/Green Automatic Switching:" : locate 6,25
print "0. First Arm - ";fml$(gauto(1)) : locate 7,25
print "1. Second Arm - ";fml$(gauto(2)) : locate 9,23
print "Local/Global Automatic Switching:" : locate 10,25
print "2. Huey - ";fml$(rauto(1)) : locate 11,25
print "3. Huey Newie - ";fml$(rauto(2)) : locate 12,25
print "4. Louie Newie - ";fml$(rauto(3)) : locate 13,25
print "5. Louie - ";fml$(rauto(4)) : locate 15,23
print "Global tracking of green:" : locate 16,25
print "6. Huey - ";fml$(track(1)) : locate 17,25
print "7. Huey Newie - ";fml$(track(2)) : locate 18,25
print "8. Louie Newie - ";fml$(track(3)) : locate 19,25
print "9. Louie - ";fml$(track(4)) : locate 22,2
print "Type number to toggle flag. Hit <RETURN> to return to previous menu"
klop23: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto klop23
if A$="0" then gauto(1)=1-gauto(1) : goto flg
if A$="1" then gauto(2)=1-gauto(2) : goto flg
if A$="2" then rauto(1)=1-rauto(1) : goto flg
if A$="3" then rauto(2)=1-rauto(2) : goto flg
if A$="4" then rauto(3)=1-rauto(3) : goto flg
if A$="5" then rauto(4)=1-rauto(4) : goto flg
if A$="6" then track(1)=1-track(1) : goto flg
if A$="7" then track(2)=1-track(2) : goto flg
if A$="8" then track(3)=1-track(3) : goto flg
if A$="9" then track(4)=1-track(4) : goto flg
if A$=chr$(13) then exit sub
goto klop23
end sub

sub status static "Displays status page
cls
see = 1
Beg3: call contrast(1) : call contrast(2)
for i=1 to 4: call intensity(i) : next i
locate 1,18,0 :print "Mass Orientation Control Program Version 1.0"
locate 3,28 : print "Control Loop: ";fml$(con)
for i=1 to 2 : c=10+40*(i-1)
locate 5,(c-3)
print "Arm";i;": ";fml$(i) : locate 6,c
print "Contrast:"; : print using "###.##";contr!(i); :print "%"
locate 7,c
print "Low Threshold:"; :print using "###.##";lthresh!(i); :print "%"
locate 8,c
print "High Threshold:"; :print using "###.##";hthresh!(i); :print "%"
locate 9,c
print "R/G Auto Switching: ";fml$(gauto(i))
next i
for y=1 to 4
c=20*(y-1)+1
locate 11,c
print mass$(y);": " : locate 12,c

```

```

    call sw(x,1) : beep
    if see=1 then
        locate 5,15+40*(x-1)
        print fnr$(x)
        for y=2*x-1 to 2*x
            locate 12,20*(y-1)+10
            print fnc$(y)
        next y
    end if
end if
exit sub
end if
if contr!(x) < lthresh!(x) then
    call sw(x,0) : beep
    if see=1 then
        locate 5,15+40*(x-1)
        print fnr$(x)
        for y=2*x-1 to 2*x
            locate 12,20*(y-1)+10
            print fnc$(y)
        next y
    end if
    exit sub
end if
for y = 2*x - 1 to 2*x
    if track(y) = 0 then goto elop
    for z = 1 to 2
        call ingreen(y,z) : call inred(y,z)
        if gout(y,z) = rout(y,z) then goto nz
        if gout(y,z) > rout(y,z) then call bias(y,z,1) else call bias(y,z,-1)
        if see=1 then
            locate 12+z,20*(y-1)+8
            print bias(y,z)
        end if
    next z
nz:
    next y
elop:
end sub

```

```

sub rauto(y) static 'Determines which red servo should be in control
call intensity(y)
if glob(y) = 1 then
    if intens(y) < lrthresh(y) then call rsw(y,0) : beep : exit sub
    exit sub
end if
if intens(y) > hrthresh(y) then call rsw(y,1) : beep : exit sub
end sub

```

```

sub corner static 'Updates status in upper left corner of screen
call contrast(1) : call contrast(2)
locate 3,1,0
print "Contrast:" : locate 4,3
print "Arm1:"; : print using "###.##";contr!(1); :print"%": locate 5,3
print "Arm2:"; : print using "###.##";contr!(2); :print"%": locate 6,1
print "Control:" : locate 7,3
print "H - ";fnc$(1) : locate 8,3
print "HN - ";fnc$(2) : locate 9,3
print "LN - ";fnc$(3) : locate 10,3
print "L - ";fnc$(4) : locate 11,1
print "Control loop: ";fnl$(con)
end sub

```

```

sub getip(x) static 'Reads arm x photodiode intensity
'Read ADC corresponding to arm x photodiode into ip(x)
ip(x)=0

```

```

if A$="1" then call rsw(y,0) : goto beg121
if A$="2" then call rsw(y,1) : goto beg121
if A$="3" then exit sub
goto Klop121
end sub

```

```

sub zero(y) static
cls
call title
locate 6,25
print "1. Increase ";mass$(y);"'s rotation bias" : locate 8,25
print "2. Decrease ";mass$(y);"'s rotation bias" : locate 10,25
print "3. Increase ";mass$(y);"'s tilt bias" : locate 12,25
print "4. Decrease ";mass$(y);"'s tilt bias" : locate 14,25
print "5. Change x-bias digit incremented" : locate 16,25
print "6. Change y-bias digit incremented": locate 18,25
print "7. Return to previous menu" : locate 22,15
print "x-bias:      y-bias:      contrast, arm";fnarm(y);":"
locate 23,26 - digit(y,1) : print "^"
locate 23,40 - digit(y,2) : print "^"

```

```

Klop122: call corner : if con=1 then call update
locate 22,22 : print using "#####"; bias(y,1)
locate 22,36 : print using "#####"; bias(y,2)
x=fnarm(y) : call contrast(x)
locate 22,61 : print using "###.##";contr!(x); : print "%"
A$=inkey$
if A$="" then goto Klop122
if A$="1" then call bias(y,1,fn1(y,1))
if A$="2" then call bias(y,1,-fn1(y,1))
if A$="3" then call bias(y,2,fn1(y,2))
if A$="4" then call bias(y,2,-fn1(y,2))
if A$="5" then
  digit(y,1) = (digit(y,1) + 1) mod 4
  locate 23,22 : print "      "
  locate 23,26 - digit(y,1) : print "^"
  end if
if A$="6" then
  digit(y,2) = (digit(y,2) + 1) mod 4
  locate 23,36 : print "      "
  locate 23,40 - digit(y,2) : print "^"
  end if
if A$="7" then exit sub
goto Klop122
end sub

```

```

sub adjust static
beg2:  cls
call title
locate 6,25
print "1. Adjust green contrast thresholds" : locate 8,25
print "2. Adjust global diode thresholds" : locate 10,25
print "3. Turn controls on or off" : locate 12,25
print "4. Return to main menu"

```

```

Klop2: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto Klop2
if A$="1" then call thresh : goto beg2
if A$="2" then call rswitch(3) : goto beg2
if A$="3" then call flags :goto beg2
if A$="4" then exit sub
goto Klop2
end sub

```

```

sub thresh static

```

```

call title
locate 6,25
print "1. Switch red/green control" : locate 8,25
print "2. Switch local/global control" : locate 10,25
print "3. Re-zero global photodiodes" : locate 12,25
print "4. Return to main menu"
<lop1: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto Klop1
if A$="1" then call switch : goto beg1
if A$="2" then call rswitch(1) : goto beg1
if A$="3" then call rswitch(2) : goto beg1
if A$="4" then exit sub
goto Klop1
end sub

sub switch static
cls
call title
locate 6,25
print "1. Switch arm 1 to Green" : locate 8,25
print "2. Switch arm 1 to Red" : locate 10,25
print "3. Switch arm 2 to Green" : locate 12,25
print "4. Switch arm 2 to Red" : locate 14,25
print "5. Return to previous menu"
<lop11: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto Klop11
if A$="1" then call sw(1,1) : goto Klop11
if A$="2" then call sw(1,0) : goto Klop11
if A$="3" then call sw(2,1) : goto Klop11
if A$="4" then call sw(2,0) : goto Klop11
if A$="5" then exit sub
goto Klop11
end sub

sub rswitch(i) static
<beg12: cls
call title
locate 6,25
print "1. Huey" : locate 8,25
print "2. Huey Newie" : locate 10,25
print "3. Louie Newie" : locate 12,25
print "4. Louie" : locate 14,25
print "5. Return to previous menu"
<lop12: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto Klop12
if A$="5" then exit sub
if asc(A$)<49 or asc(A$)>52 then goto Klop12
if i=1 then y=asc(A$)-48: call lorg(y)
if i=2 then y=asc(A$)-48: call zero(y)
if i=3 then y=asc(A$)-48: call rthresh(y)
goto beg12
end sub

sub lorg(y) static
<beg121: cls
call title
locate 6,25
print "1. Switch ";mass$(y);" to Local" : locate 8,25
print "2. Switch ";mass$(y);" to Global" : locate 10,25
print "3. Return to previous menu"
<lop121: call corner : if con=1 then call update

```

```

nlop: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto nlop
if ((asc(A$)>47 and asc(A$)<58) or A$=".") and len(B$)<5 then
  B$=B$+A$
  locate 23,39+len(B$) : print A$
  if len(B$)<5 then locate 23,40+len(B$) else locate 23,44
  goto nlop
end if
if A$=chr$(13) then
  if len(B$) = 0 or B$="." then z! = -1 else z!=val(B$)
  locate 23,25,0
  print " "
  exit sub
end if
if A$=chr$(8) and len(B$)<>0 then
  B$=right$(B$,len(B$)-1)
  locate 23,40+len(B$) : print " "
  if len(B$)<>0 then locate 23,39+len(B$) else locate 23,40
  goto nlop
end if
goto nlop
end sub

sub rthresh(y) static
cls
call title
locate 6,25
print "1. Increase ";mass$(y); " low threshold" : locate 8,25
print "2. Decrease ";mass$(y); " low threshold" : locate 10,25
print "3. Increase ";mass$(y); " high threshold" : locate 12,25
print "4. Decrease ";mass$(y); " high threshold" : locate 14,25
print "5. Input new ";mass$(y); " low threshold" : locate 16,25
print "6. Input new ";mass$(y); " high threshold" : locate 18,25
print "7. Return to previous menu": locate 22,10
print "Low Threshold:          High Threshold:          Intensity:"
rplth: locate 22,24 : print using "#####";lrthresh(y)
locate 22,47 : print using "#####";hrthresh(y)
klop22: call corner : if con=1 then call update
call intensity(y)
locate 22,65 : print using "#####";intens(y)
A$=inkey$
if A$="" then goto klop22
if A$="1" and lrthresh(y) < hrthresh(y) then
  lrthresh(y)=lrthresh(y)+1
  goto rplth
end if
if A$="2" and lrthresh(y) > 0 then
  lrthresh(y) = lrthresh(y) - 1
  goto rplth
end if
if A$="3" and hrthresh(y) < 4095 then
  hrthresh(y) = hrthresh(y) + 1
  goto rplth
end if
if A$="4" and lrthresh(y) < hrthresh(y) then
  hrthresh(y) = hrthresh(y) - 1
  goto rplth
end if
if A$="5" then
  call ninput(z!)
  if z!=-1 then goto klop22
  if z!<= hrthresh(y) then lrthresh(y)=int(z!) else beep
  goto rplth
end if

```



```

locate 6,25
print "1. Adjust arm1 threshold" : locate 8,25
print "2. Adjust arm2 threshold" : locate 10,25
print "3. Return to previous menu "
<lop21: call corner : if con=1 then call update
A$=inkey$
if A$="" then goto klop21
if A$="3" then exit sub
if A$="1" or A$="2" then x=asc(A$)-48 : call thresh1(x) : goto beg21
goto klop21
end sub

sub thresh1(x) static
cls
call title
locate 6,25
print "1. Increase arm";x;"low threshold" : locate 8,25
print "2. Decrease arm";x;"low threshold" : locate 10,25
print "3. Increase arm";x;"high threshold" : locate 12,25
print "4. Decrease arm";x;"high threshold" : locate 14,25
print "5. Input new arm";x;"low threshold" : locate 16,25
print "6. Input new arm";x;"high threshold" : locate 18,25
print "7. Return to previous menu": locate 22,10
print "Low Threshold:      High Threshold:      Contrast:      % "
plth: locate 22,24 : print using "##.##";lthresh!(x); :print "%"
locate 22,47 : print using "##.##";hthresh!(x); :print "%"
<lop211: call corner : if con=1 then call update
call contrast(x)
locate 22,64 : print using "##.##";contr!(x)
A$=inkey$
if A$="" then goto klop211
if A$="1" and lthresh!(x) <= hthresh!(x)-.1 then
  lthresh!(x)=lthresh!(x)+.1
  goto plth
end if
if A$="2" and lthresh!(x) >= .1 then
  lthresh!(x) = lthresh!(x) - .1
  goto plth
end if
if A$="3" and hthresh!(x) < 99.89 then
  hthresh!(x) = hthresh!(x) + .1
  goto plth
end if
if A$="4" and lthresh!(x) <= hthresh!(x)-.1 then
  hthresh!(x) = hthresh!(x) - .1
  goto plth
end if
if A$="5" then
  call ninput(z!)
  if z!=-1 then goto klop211
  if z!<= hthresh!(x) then lthresh!(x)=z! else beep
  goto plth
end if
if A$="6" then
  call ninput(z!)
  if z!=-1 then goto klop211
  if z! >= lthresh!(x) and z!< 100 then hthresh!(x)=z! else beep
  goto plth
end if
if A$="7" then exit sub
goto klop211
end sub

sub ninput(z!) static      'Input a number while maintaining mass control

```

```
call ingreen(y,z)

k! = ip(x) + (bias(y,z) - gout(y,z))^2
if abs(k!) < 32767 then ip(x) = k! else ip(x) = 32767

next z
next y
ip(x) = ip(x)/40
end sub

sub getil static      'Reads laser meter photodiode intensity
'Read ADC corresponding to laser meter into il(x)
il=500 ' For the sake of argument
end sub

sub inred(y,z) static 'Reads Global UDT output
'Read ADC corresponding to UDT output mass y dimension z into rout(y,z)
rout(y,z)=bias(y,z) 'For the sake of argument
end sub

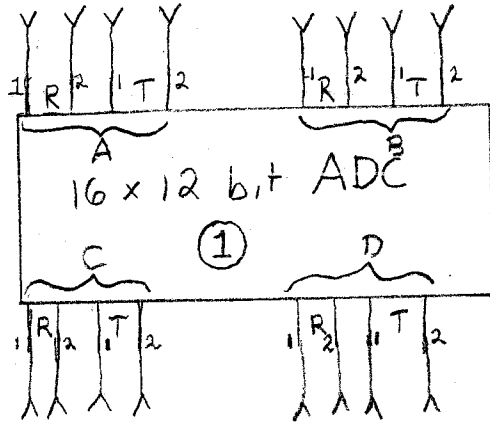
sub ingreen(y,z) static 'Reads Green servo output
'Read ADC corresponding to Green servo output mass y dimension z
gout(y,z)=gout(y,z)+int(rnd#3)-1 'For the sake of argument
end sub
```

Arms: 1, 2
 Masses: A, B, C, 1

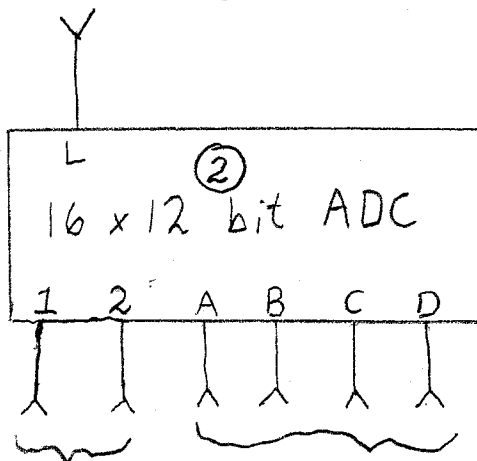
IBM PC/AT (clk

From Servo Outputs (Mass A, B, C, D)

1. Global UDT
2. Green System



From Laser Intensity Meter



From Arms 1, 2
 Photodiodes

From Masses A, B, C, D
 Global Photodiodes (Intensity)

While arm X is unc
 Read 2X and 2
 if Contr < LowThresh

While mass Y in ar
 Green control:
 Read 2X, 2L
 if Contr < HighThreshTh

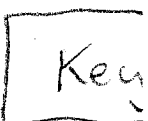
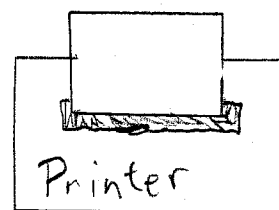
Let Z be either R
 or T

If $|YZ1| > |YZ$.
 Then Decr. 3Y
 if 3YZ is minim

If $|YZ1| < |YZ$:
 Then Incr. 3Y
 if 3YZ is max

Repeat $\forall Y, Z$ un

Optional: if Mass Y un
 if $2Y < RT$
 if Mass Y un
 if $2Y > RTh$



Dimensions: R, T (rot, tilt)

Lasers: R, G (red, green)

To Remote Bias Input of Global UDT Boxes. (Masses A, B, C)

Red control

then Clr 3PBXG

is under

Clr 3PBXR

rot
tilt

then set 3PAYZ

then set 3PAYZ

- Green

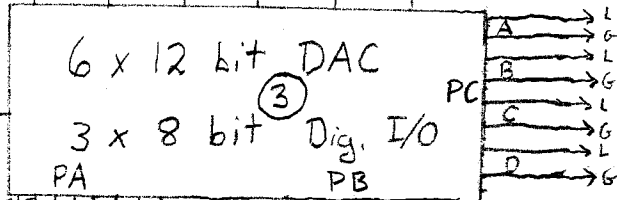
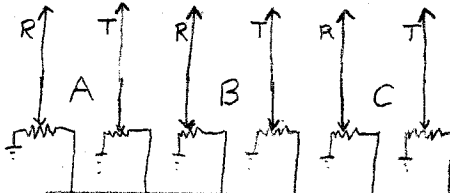
- Global:

then Clr 3PCYL

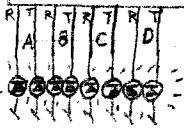
Local:

then Clr 3PCYG

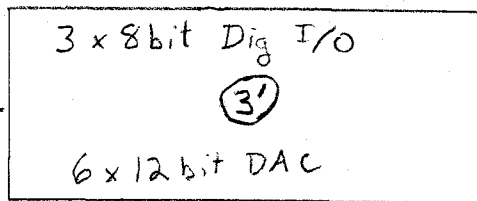
card



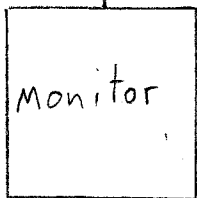
To Local/Global Switches



To Red/Green Switches



To Remote Bias UDT input (mass D)



CALIFORNIA INSTITUTE OF TECHNOLOGY GRAVITATIONAL PHYSICS		
BLSURF		
DRAWN BY	DATE	DRAWING NO. 1
CHECKED BY	SCALE	
APPROVED BY	W.O.	

Songs: 03

1

2

1. Error

~~B~~ B

B

2. Normalize Contrast

C16C16C16

C16C16C16

3. Bias out of range

~~BGG~~ BG

~~BGG~~ BG

4. Interferometer Locks

G8A8B8

C16F8F16F16G16D+8F8C8

5. Out of Lock

E8D8C8

F8C16C16D8C8P8E8F4

6. Switch to Local 4.5

C8D8C8

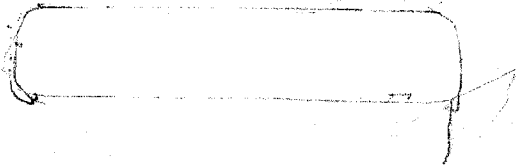
~~E8~~ E16D16C8C8C16D16E16F16G8G8G8E4

7. Switch to Global

F8E8F8

C16C8C16F5F5G5G5>C<P32A16F4

4.6 x 11



~~1. Begin Control Loop~~
~~2. End Control Loop~~
~~3. Manual Control Options~~
~~4. Adjust Parameters~~
~~5. View Status~~
~~6. Return to System~~

1. Begin Control Loop
2. End Control Loop
3. Manual Control Options
4. Adjust Parameters
5. View Status
6. Return to System

xbias: _ _ _ _ _ m

ybias: _ _ _ _ _

Contrast, Arm 1: _ _ _ _ _