



HGCROC input signal polarity swap

C. Krug and M. Bregant

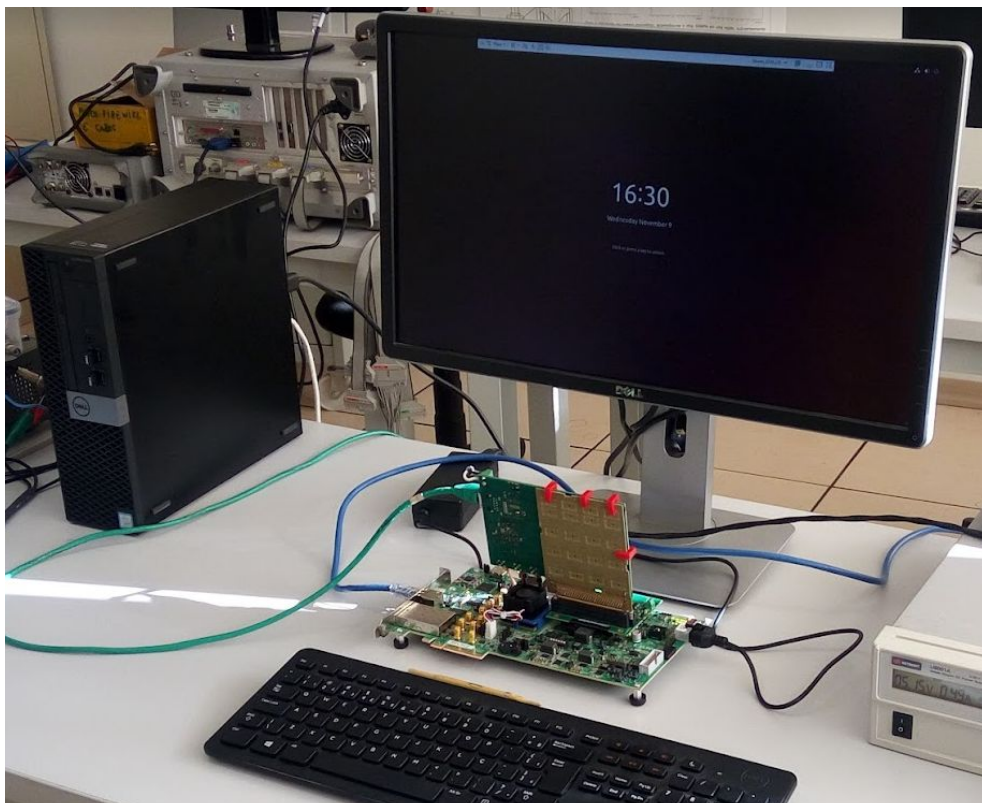
August 24, 2023

Motivation

Enable FoCal-E test beam run in October with n-type pad sensors (from India)

Issue

Proper signal acquisition with HGCROC from n-type sensors is still lacking



Test bench in Sao Paulo
(remotely accessible)

- HGCROC (v1):

```
>>single_hgcroc IDCODE 0x5110fb06 Single pad v1
```

- KCU105
- Injector board v1

3.2 "Channel-wise" I2C parameters

Register # 0			
Bit	Name	Default	Description
0	Inputdac<0>	"0"	Leakage current Input-DAC Value
1	Inputdac<1>	"0"	
2	Inputdac<2>	"0"	
3	Inputdac<3>	"0"	
4	Inputdac<4>	"0"	
5	Inputdacpol	"0"	Leakage current Input-DAC polarity

HGCROC2 / 2A DATASHEET

49

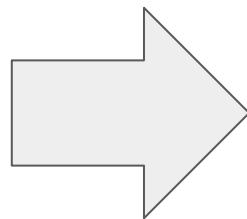
3.3 "Global analog" I2C parameters

Register # 8			
Bit	Name	Default	Description
0	Rc<1>	"1"	Register # 14
1	Rc<0>	"0"	
2	Sel_input_toa	"0"	
3	En_hyst_tot	"1"	
4	Cf_comp<0>	"0"	
5	Cf_comp<1>	"1"	
6	Neg	"1"	"1" = negative input polarity (Default)
7	Pol_trig_toa	"1"	Polarity of the TOA discri output

Register # 14			
Bit	Name	Default	Description
0	Pol_adc	"1"	ADC input swap

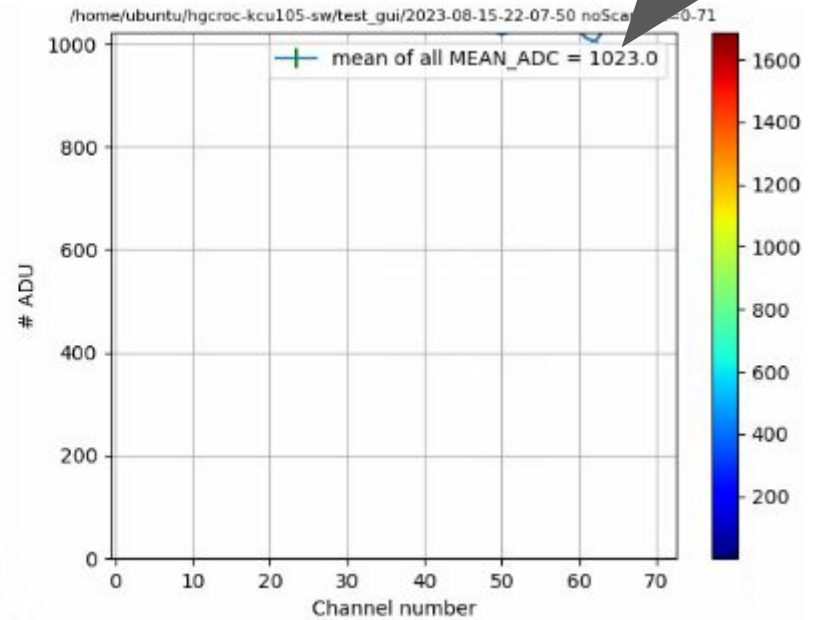
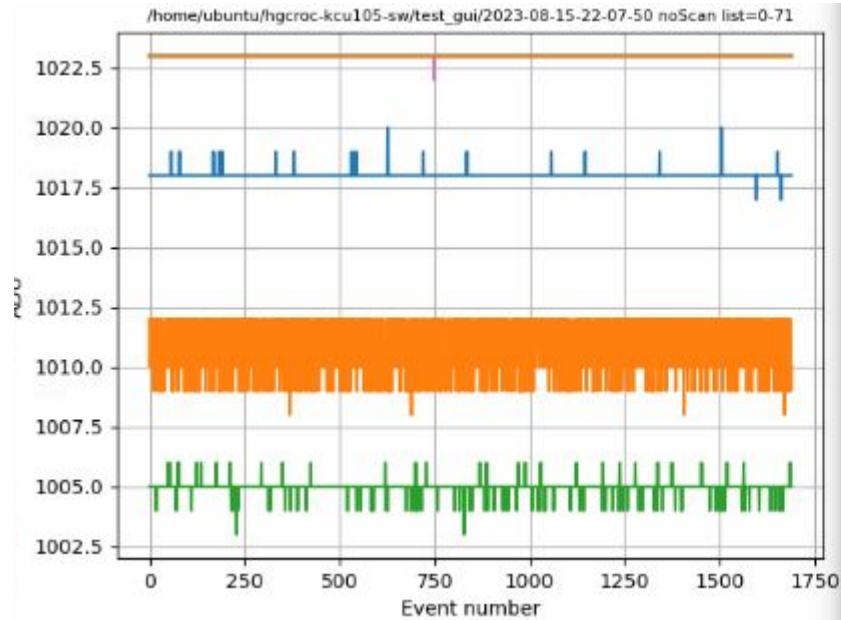
55

```
Tkcu105_API.cpp <Select Symbol>
1027
1028 //to be moved in mainwidget
1029 setSignalPolarity(!isPos>false);
1030
1031 //default setting
1032 uint8_t channel[72];
1033 uint8_t calib[2];
1034 uint8_t CM[4];
1035
1036 //uint16_t my_dac_val=100;
1037
1038 for (int i=0;i<72;i++)
1039     channel[i]=0;
1040 for (int i=0;i<2;i++)
1041     calib[i]=0;
1042 for (int i=0;i<4;i++)
1043     CM[i]=0;
1044 setChannelLeakageCompensation(!isPos>false,channel,calib,CM);
1045 //setChannelLeakageCompensation(!isPos>true,channel,calib,CM);
```



```
Tkcu105_API.cpp <Select Symbol>
1027
1028 //to be moved in mainwidget
1029 setSignalPolarity(!isPos>true);
1030
1031 //default setting
1032 uint8_t channel[72];
1033 uint8_t calib[2];
1034 uint8_t CM[4];
1035
1036 //uint16_t my_dac_val=100;
1037
1038 for (int i=0;i<72;i++)
1039     channel[i]=0;
1040 for (int i=0;i<2;i++)
1041     calib[i]=0;
1042 for (int i=0;i<4;i++)
1043     CM[i]=0;
1044 //setChannelLeakageCompensation(!isPos>false,channel,calib,CM);
1045 setChannelLeakageCompensation(!isPos>true,channel,calib,CM);
```

ADC readings, zero input



```
ckrug@charrua:~/Documents/jobs/ALICE/hgcroc-kcu105-sw$ grep --exclude=*.ipynb -r
n -e "isPos"
kcu105_API/Tkcu105_API.cpp:1029:setSignalPolarity(/*isPos*/false);
kcu105_API/Tkcu105_API.cpp:1044:setChannelLeakageCompensation(/*isPos*/false,cha
nnel,calib,CM);
kcu105_API/Tkcu105_API.cpp:1045://setChannelLeakageCompensation(/*isPos*/true,ch
annel,calib,CM);
hgcroc-api/Thgcroc_I2C_API.cpp:214:void Thgcroc_I2C_API::setSignalPolarity(bool
isPos)
hgcroc-api/Thgcroc_I2C_API.cpp:216:hgcroc_i2c_indirect_setField(/*sub_block_add*
/globalAnalogAdd(0),/*reg*/8,/*bitOffset*/6,/*fieldWidth*/1,(isPos)?0:1,"setSign
alPolarity");
hgcroc-api/Thgcroc_I2C_API.cpp:217:hgcroc_i2c_indirect_setField(/*sub_block_add*
/globalAnalogAdd(1),/*reg*/8,/*bitOffset*/6,/*fieldWidth*/1,(isPos)?0:1,"setSign
alPolarity");
hgcroc-api/Thgcroc_I2C_API.cpp:220:void Thgcroc_I2C_API::setChannelLeakageCompen
sation(bool isPos,uint8_t channel[72],uint8_t calib[2], uint8_t CM[4])
hgcroc-api/Thgcroc_I2C_API.cpp:223:      hgcroc_i2c_indirect_write(/*sub_block_ad
d*/channelAdd(i),/*reg*/0,channel[i] | ((isPos)?0:(1<<5)),"setChannelLeakageComp
ensation"); //TODO: check isPos enable
hgcroc-api/Thgcroc_I2C_API.cpp:226:      hgcroc_i2c_indirect_write(/*sub_block_ad
d*/calibAdd(i),/*reg*/0,calib[i] | ((isPos)?0:(1<<5)),"setChannelLeakageCompensa
tion"); //TODO: check isPos enable
hgcroc-api/Thgcroc_I2C_API.cpp:229:      hgcroc_i2c_indirect_write(/*sub_block_ad
d*/CMAdd(i),/*reg*/0,CM[i] | ((isPos)?0:(1<<5)),"setChannelLeakageCompensation"
); //TODO: check isPos enable
hgcroc-api/Thgcroc_I2C_API.hpp:24:      void setSignalPolarity(bool isPos);
hgcroc-api/Thgcroc_I2C_API.hpp:25:      void setChannelLeakageCompensation(bool is
Pos,uint8_t channel[72],uint8_t calib[2], uint8_t CM[4]);
```

Only reg 0, bit 5 and
reg 8, bit 6 covered

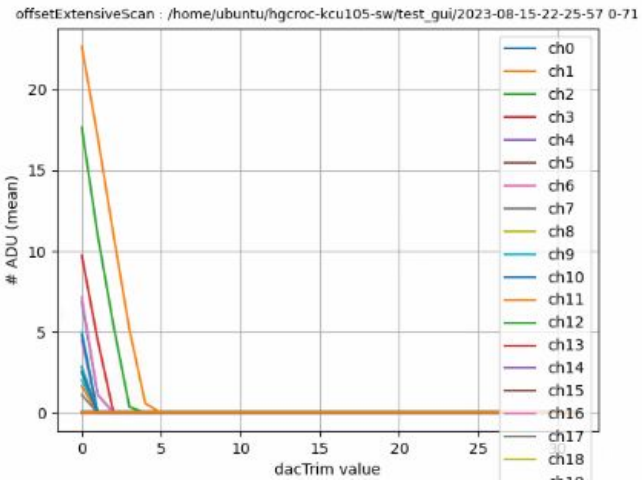
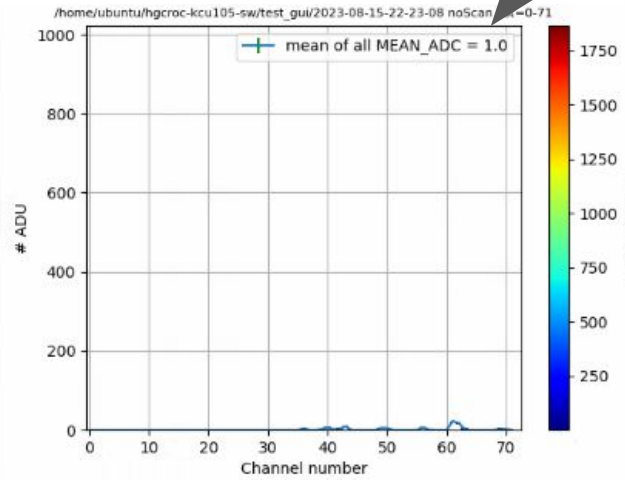
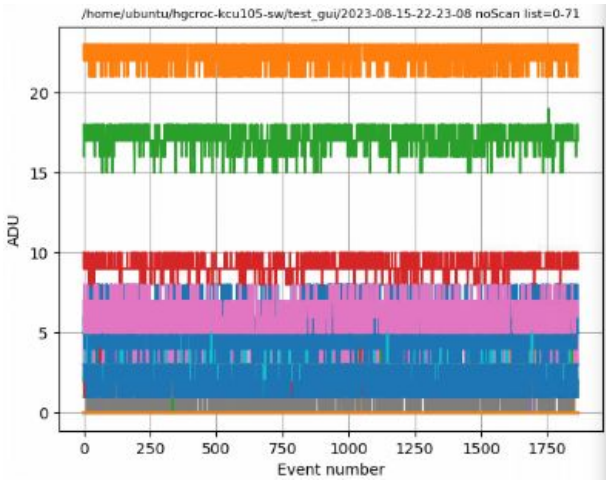
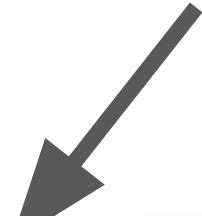
Added to code

Reg 8, bit 7

Reg 14, bit 1

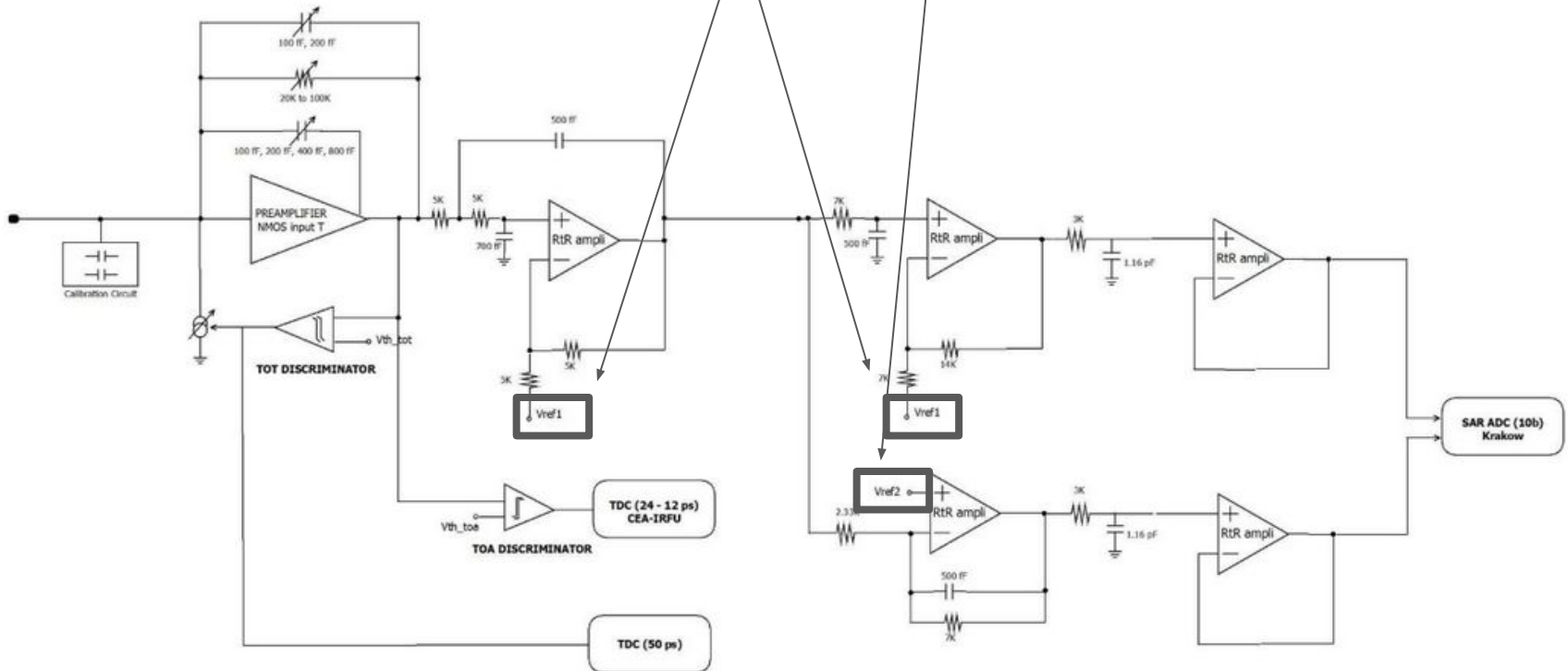
```
Tkcu105_API.cpp* <Select Symbol>
1071 hgcroc_i2c_indirect_setField(/*sub_block_add*/globalAnalogAdd(a),/*reg*/8,/*bitOffset*/7,/*fieldWidth*/1,0,"initHgcroc: pol_trig_toa"); //pol_trig_toa
1072 hgcroc_i2c_indirect_setField(/*sub_block_add*/globalAnalogAdd(a),/*reg*/14,/*bitOffset*/1,/*fieldWidth*/1,0,"initHgcroc: pol_adc"); //pol_adc
```

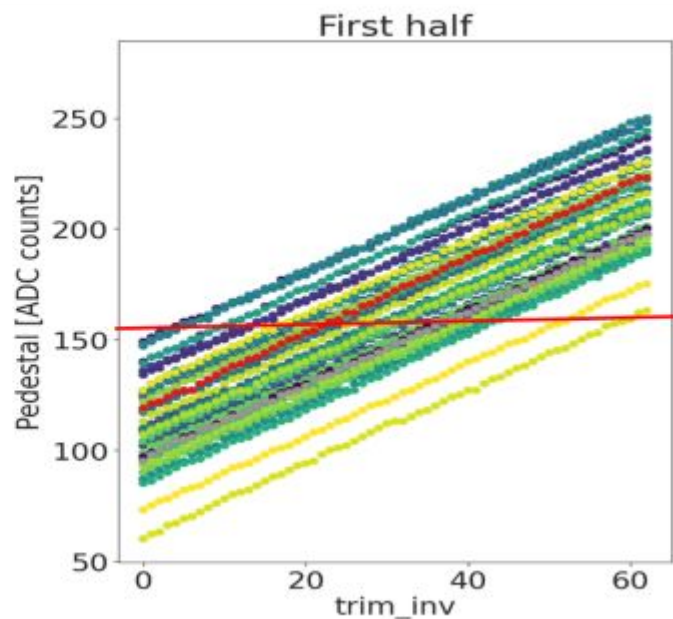

ADC readings, zero input



1.1.1. Pedestal trimming

The default value of V_{ref_inv} and V_{ref_noinv} (224 and 160) don't seem to be the best ones as a starting point. I believe that 260 and 100 seems to be better to avoid a saturation of one shaper branch.





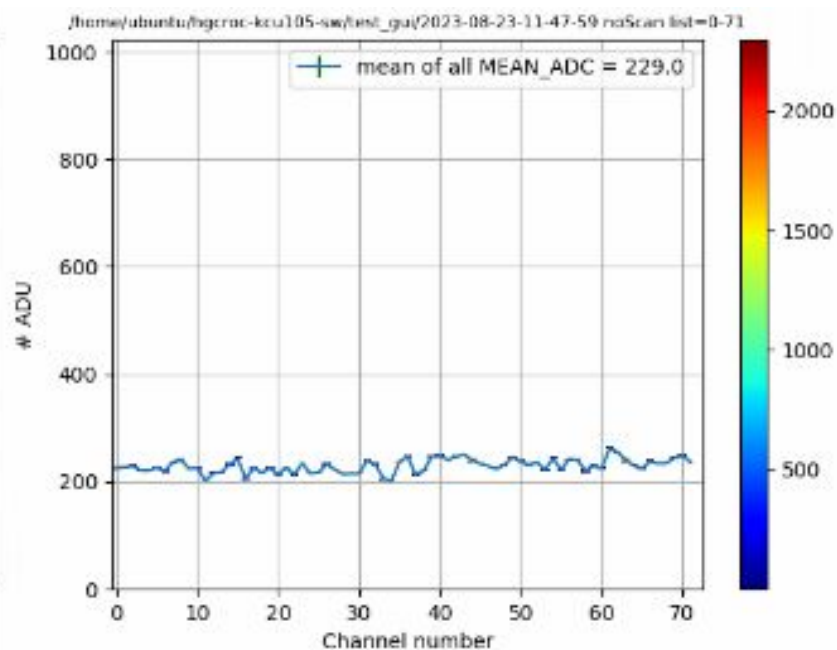
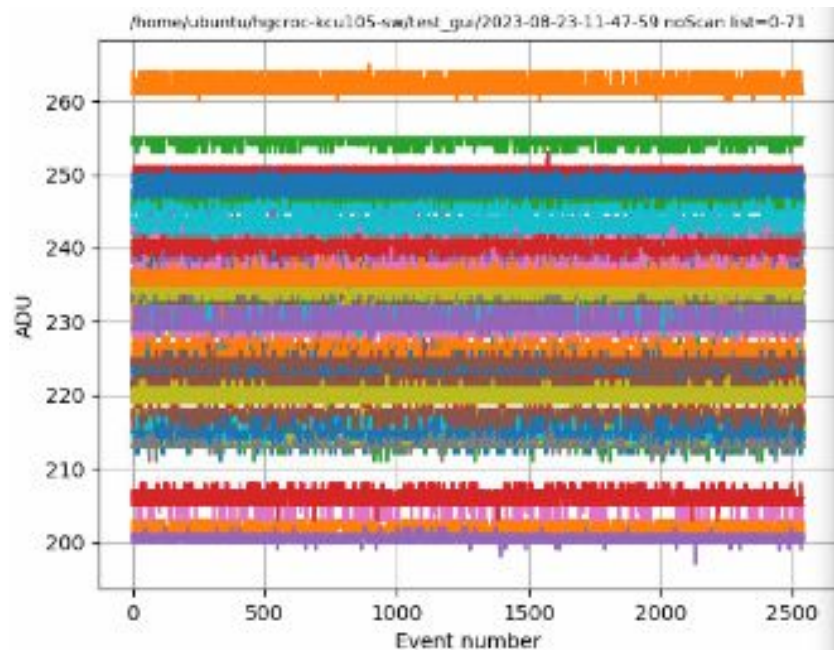
```

TsingleHgcrocConfigurator.cpp
/home/ubuntu/hgcroc-kcu105-sw/test_gui/2023-08-21-05-56-59 0-71 Offse...
849 {
850   noInv[0]=110;
851   noInv[1]=110;
852   inv[0]=350;
853   inv[1]=350;
854 }
855 ///////////////
856 void TsingleH
857 {
858   int fileInvEx
859   int fileNoInv
860   for (int i=0;
861   {
862     noInv[i]=
863     inv[i]=0;
864   }
865
866   fileInvExist=
867   fileNoInvExis
868
869   ostream
870   oss << asicDi
871   if (fileInvEx
872   {
873     oss << "
874     setDefaul
875   }
876
877   if (noInv[0]<
878   {
879     oss << "
880     setDefaul
  
```

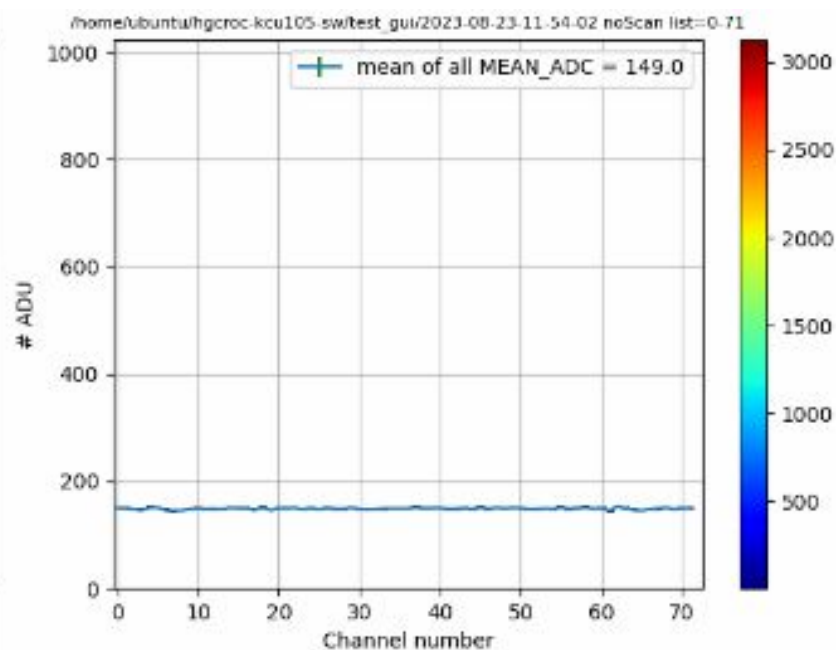
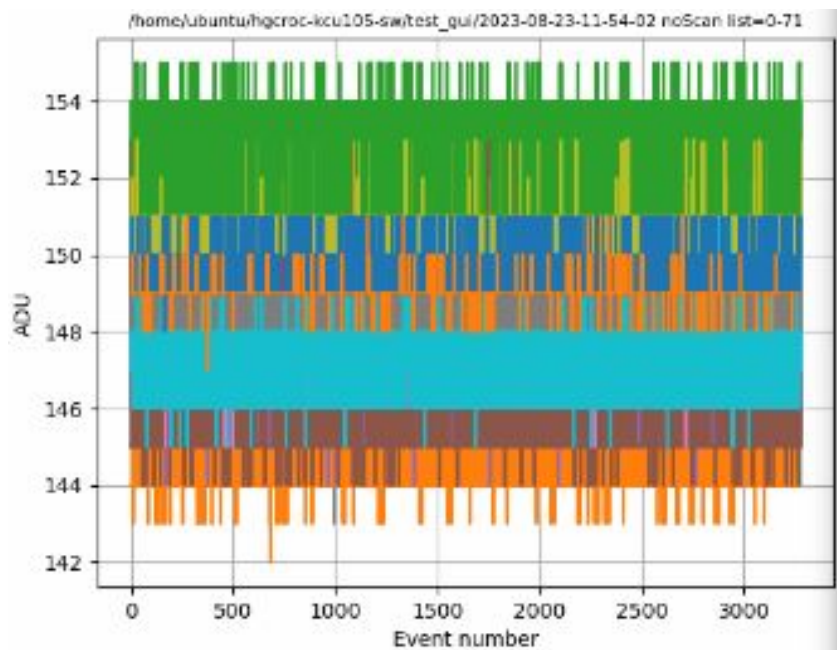
offsetExtensiveScan

This graph shows the mean number of ADU versus the dacTrim value for an offsetExtensiveScan across channels ch0 to ch18. The x-axis, 'dacTrim value', ranges from 0 to 25. The y-axis, '# ADU (mean)', ranges from 50 to 250. All channels exhibit a downward trend. A thick black horizontal line is drawn at approximately 140 ADU.

Before trimming



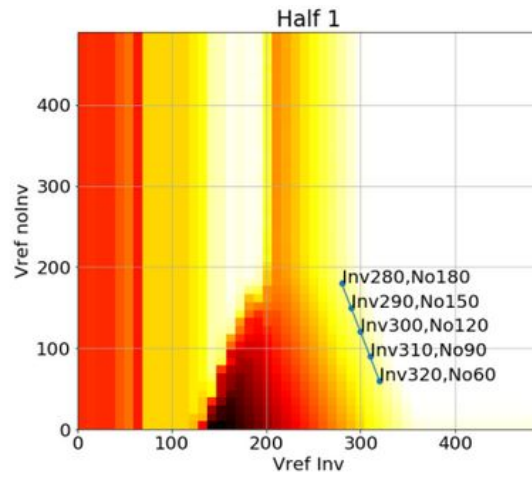
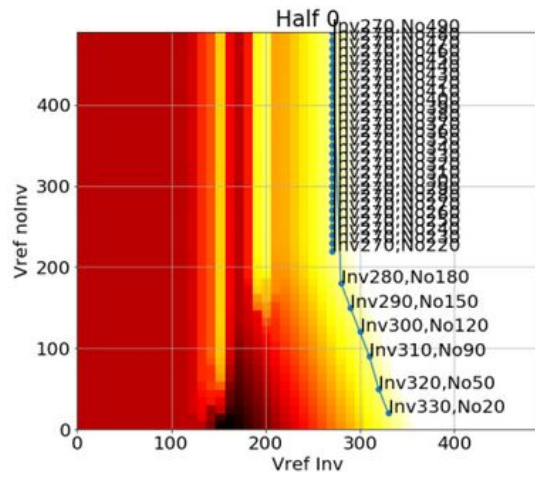
After trimming



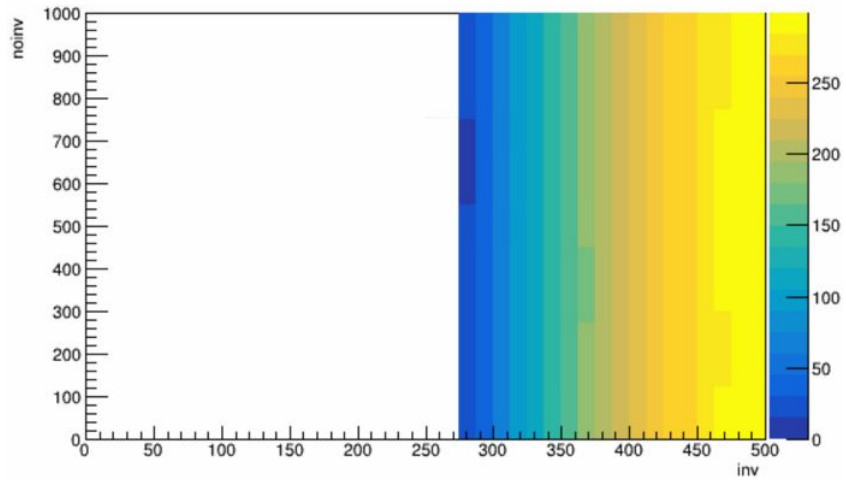
```
TsingleHgcrocConfigurator.cpp* [X] [A] TsingleHgcrocConfigurator::
688 //          correction_val=(max_offset-offsetNull[36*s+i])/fact[36*s+i];
689 |          correction_val=(146-offsetNull[36*s+i])/fact[36*s+i]; // ck
```

Pedestal trimming accomplished!

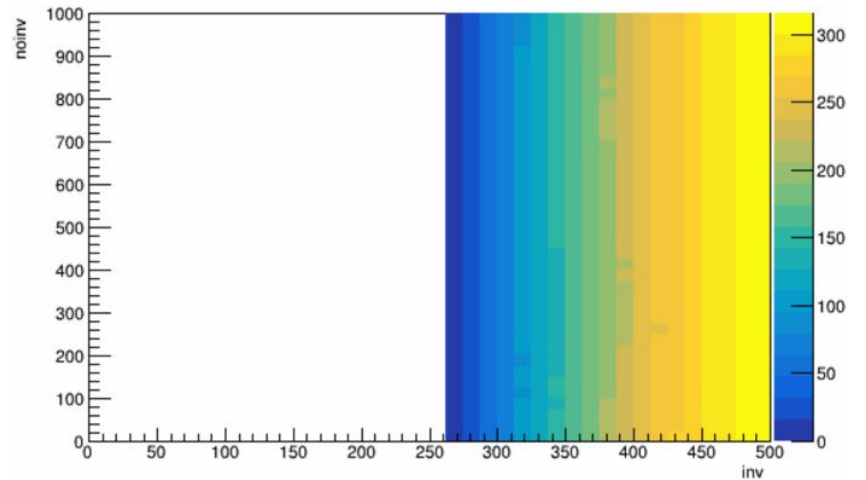
Now pedestal setting...



DynRange2D_L



DynRange2D_R



Summary

- Polarity swap accomplished (apparently...)
- Pedestal trimming “fully” understood
- Pedestal setting in progress
 - Check role of noinv setting
- Next step: delay scan
 - Must find a suitable signal source