ACTUAL FILE STRUCTURE OF 619 is https://cernbox.cern.ch/index.php/s/UZvAnfAl3PQAgHo password is Cleanroom888 compare with the google sheets https://docs.google.com/spreadsheets/d/1Xcgwns5tEniBfvvNIwItc81DTGkS1qoR2xgMTo2ZgR4/edit#gid=1010834375

MAKE SURE YOU HAVE V6 NOT V5 OF THE REPOSITORY

Running Docker with the mounted shared directory

`docker run -it -v <data directory on host machine>:/home -v <output directory on host>:/output thorbenquast/lcd_hgc_ana_wf`

`docker run -it -v /home/ali/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES:/home -v /home/ali/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES/OUTPUT:/home/output thorbenquast/lcd_hgc_ana_wf`

`docker run -it -v /home/ali/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES:/home -v /home/ali/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES/OUTPUT_ALI:/home/OUTPUT_ALI thorbenquast/lcd_hgc_ana_wf` this command makes available HGCAL_ANALYSISWF_FILES directory, which has the github repo as well as the raw data and the output, available in the docker /home/ directory

with this, my local directory is available in home on the docker machine now edit the environment variables in setup.sh in /home/lcd_hgcal_analysisworkflows-v0.5/

its important to edit the

Editing DUT.csv and IVMeasurements.csv

ALL sensors should be stored in /home/lcd_hgcal_analysisworkflows-v0.5/databse/csv/DUT.csv, this is where each sensor is referenced by their ID, etc. `DUT.csv's` first column references sensors by `scratchpad_ID` whereas `IV_Measurements.csv` 's first column references sensosrs by `senor_ID and` second by `scratchpad_ID` .

you have to edit the raw data paths in `DUT.csv` and `IV_Measurements.csv` to that of your **docker paths**.

in IV_Measurements.csv the RawFilePaths end in "_IV.txt" `sudo sed -i 's:home\/input:home\/HGCSiliconRawData:g' IVMeasurements.csv` `sudo sed -i 's:home\/rawdata:home\/HGCSiliconRawData:g' IVMeasurements.csv` This replacement replaces the raw path that is referenced in IVMeasurements.csv, from the original /home/input to my actual directory structure in docker, which is /home/HGCSiliconRawData.

wen can leave hex_positions.. as "-"

setup.sh

`source setup.sh` has to be executed each time the docker image is initialized. Here is what my lchworflow/source.sh looks like:

export ROOT_SOURCE="/software/ROOT_6_06_06/bin/thisroot.sh";

export HEXPLOT_DIR="/software/hexplot/bin";

export WORKFLOW_DIR="/home/lcd_hgcal_analysisworkflows";

export DB_DIR="$WORKFLOW_DIR/database/csv";

export PYTHONPATH=$WORKFLOW_DIR:$PYTHONPATH;

export TMPFILES_DIR_IV="/home/OUTPUT_ALI/TMPFILES_DIR_IV"

export SUMMARY_DIR_IV="/home/OUTPUT_ALI/SUMMARY_DIR_IV"

source $WORKFLOW_DIR/workflow_IV/setup.sh;

export TMPFILES_DIR_CV="/home/OUTPUT_ALI/TMPFILES_DIR_CV"

export SUMMARY_DIR_CV="/home/OUTPUT_ALI/SUMMARY_DIR_CV"

source $WORKFLOW_DIR/workflow_CV/setup.sh;

Running the automatic analysis

Before running, **make sure your sensor is not only in IVMeasurements.csv but also is in DUT.csv, and also that their paths and the geo paths are correct**.

`AnalyseMeasurement_IV 2 N3308_5`

here N3308_5 is the sensor ID

Before deleting,

the output of my most recent run (N4790_19) is in ~/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES/OUTPUT_ALI/

Currently, the user needs to execute another command before deleting for the automatic update

This is so that we avoid linking the command AnalysicMeasurement to the execution of the python program. (we probably should clone fsudb in /software/ so that

the user doesn't see it)

`fsudb updateDB`

The grading info is in ~/Desktop/Research/HGCAL/DATABASE/HGCAL_ANALYSISWF_FILES/OUTPUT_ALI/TMPFILES_DIR_IV/-/grading/N4790_19

then you have to delete output if you want to run analysis again `DeleteAnalysisFiles_IV <MEASUREMENT ID>`

the "geo" files in DUT.csv is located in /software/hexplot/bin/geo/ .

It looks like the grading results are potentially stored, in `HGCAnalysisOutput/hgsensor_iv/grading/somesensor/grading_results.txt`

idea: presumably people are entering this file in each measurement in the DUT file by hand, but this too should be automated based on the scratchpad ID of the sensor, eg hex_positions_HPK_128ch_6inch.txt vs hex_positions_HPK_198ch_8inch_testcap.txt

E.g. N3308_5 is HPK_8in_192_N3308_5, meaning the entry for it in DUT is HPK_8in_192_N3308_5 and hence the geo file is /software/hexplot/bin/geo/hex_positions_HPK_198ch_8inch_edge_ring_testcap.txt

Also everything is permission blocked sudo. The Hexplot takes the longest to run.

Options for piping the results to our database:

```
1. work within the lcd_hgc_ana_wf code so that it opens the database and changes it (not preferred)
2. our code is initialized when "AnalyseMeasurement_IV 2 N3308_5" it imports the relevant package "from grading import IV_grading" o
```

Since the results are calculated in the grading code, we should access those variables - option 1: grepping from the latetex files of the results: this is in /lcd_hgcal_analysisworkflows/workflow_IV/tex option 2: actaully using the fuctions - this is preferred, but it will just take a bit more in the analysis.

```
option 3: very preferred: make a git branch (fork) that saves the summary stuff that we want into csv file that is easy to parse. Th

Things to do today:
* make sure my directory structure is like the standard one that Alex sent.
* learn about git branches and how I can create one, and how I can use that branch as default in 619. (I forked it at https://gitlab
* make the code for that branch that just just saves a csv file for each run of the summary results.
    The file that is readout is the *characterization.pdf file which is in workflow_IV/tex/main.tex
* work on my code to use that csv file to update the table(s) for IV.
```

Luigi

They use luigi as the wrapper for the commands, for example in /workflow_IV/setup.sh there is AnalyseMeasurement_IV() { MEASUREMENTID=$2 luigi --module workflow_IV.tasks.wrappers IV.AnalyseMeasurement --workers $1 --MeasurementID $MEASUREMENTID --local-scheduler;
} this links the wrappers, where there is a class for each command, eg class AnalyseMeasurement(MeasurementTask): task_namespace = 'IV' ... it has a requires and ourput (and run) functions