

Database IpGBT Configuration

Authors: Hayden Swanson and Naomi Gonzalez

Date: August 2025

The IpGBT needs specific registers set to particular values depending on what hardware it is interfacing with. This is what we call the configuration. There are about 500 registers on the IpGBT but not all are for writing (some are used as status registers for example).

Getting DB Ready for ETL

The DB accomodates BTL very well and has been designed to meet their needs. Now we are trying to also integrate our needs into the current MTD DAQ SW workflow. We would like to make some adjustments to the database. Here is what we propose:

Planned Solution

1. Define every writable register on the IpGBT in the database as a column
 - This will be done using SQLAlchemy models. (Maybe also use SQLModel instead? Pydantic + SQLAlchemy)
 - We seperate groups the registers into different tables the same way they group them in the manual for the IpGBT
 - <https://lpgbt.web.cern.ch/lpgbt/v1/registermap.html>
(<https://lpgbt.web.cern.ch/lpgbt/v1/registermap.html>)
2. We only write full register values to the database
 - Helper functions to write registers or groups of registers can be added as methods/functions to the SQLAlchemy models

- BTL db columns can be redefined as properties to the SQLAlchemy models to ensure their interface to the db is unchanged
 - Example: The BTL column `frequency_ps_all` can be mapped to a group of registers (which would be a group of columns)

3. Change everything from `cc` to board name in the database and sw

Work Finished

- register and values are not uniquely defined but are abstracted into groups like `frequency_ps_all`. The db configuration is obtained by requesting it from a server. We updated the server output to include this.
- Got rid of copy pasted code between ETL and BTL and moved the JSON output definition to a `to_dict` method in the models for ETL and BTL tables to share
 - We also made them share the same route and you just give what board (`cc` or readout board) and chip (`lpgbt`)
- We tested the code on BTL hardware, see full report here ()
 - For the future we will try to test on an individual spare `cc`
- We spun up a test instance of the db on the `etl@ceamtddaq01` to not interfere with BTL testing
- We outputted every register on the `lpgbt` after configuration with Tamalero
 - MTD vs Tamalero Registers
(https://docs.google.com/spreadsheets/d/1dzNn1ADX_3XBBtkMtJMBIpbZh5U2UM-WEs1sQf4FW3LQ/edit?gid=0#gid=0)

See here (<https://codimd.web.cern.ch/0eansdAyQy2iA7OkLIh5Hw>) for raw outputs from the terminal and report of testing on BTL hardware

Next Steps

- Finalize idea and garner approval from Mohamed
- Create Slides and present proposal to Ozgur