MUX64 Software Development - Raw ADC value

Author: Naomi Gonzalez and Hayden Swanson
Date: August 2025

Introduction

The MUX64 is an analog multiplexer with 64 input channels, one output, and 6 select input signals. This chip is used to monitor temperatures/voltages on the service hybrid (readout board) and connected modules/ETROCs. Its select channels are set by using GPIO pins on the Master lpGBT and its output is read through an ADC input of the Master lpGBT. Since MTD DAQ already has code to control the lpGBT we inherited this and made a new mux64 controller class to control the mux64 on MTD DAQ SW.

Tests

Tamalero + KCU105 Reference

In order compare outputs read with the Serenity and MTD DAQ software first we extracted the correct readout values using the existing (tested and verified) Tamalero and KCU105 test-stand setup.

We used the `module_test_sw` docker image, and ran `python -i test_tamalero.py --power_up --configuration modulev0b`. Afterwards then running the read all channels command of the mux64 using the interactive terminal, which obtained the following outputs:

- Readout Board with no module connected

```
>>> rb.MUX64.read_channels()
```

Channel	Pin	Reading (raw)	Reading (calib)	Voltage (direct)	Voltage (conv)	Comment
A0	0	809	827.932	0.809318	8.8916	LV RB
A1	1	515	530.973	0.519035	1.03807	lpGBT ADC5 (RT2), R2 values is approximate
mod0_a5	3	24	32.3243	0.0306069	0.0315976	VREF on module 1
mod0_a6	4	24	32.3243	0.0315976	0.0315976	VTEMP on module 1
mod1_a5	7	24	32.3243	0.0315976	0.0315976	VREF on module 2
mod1_a6	8	24	32.3243	0.0315976	0.0315976	VTEMP on module 2
mod2_a5	11	24	31.3108	0.0315976	0.0315976	VREF on module 3
mod2_a6	12	24	32.3243	0.0315976	0.0325883	VTEMP on module 3

- Readout Board with module 25 connected

```
RB configured successfully. Rhett is happy
>>> rb.MUX64.read_channels()
```

Channel	Pin	Reading (raw)	Reading (calib)	Voltage (direct)	Voltage (conv)	Comment
A0	0	806	564.419	0.807337	8.86981	LV RB
A1	1	543	559.351	0.546776	1.08959	lpGBT ADC5 (RT2), R2 values is approximate
mod0_a5	3	997	1018.47	0.996565	0.995575	VREF on module 1
mod0_a6	4	569	583.676	0.571544	0.571544	VTEMP on module 1
mod1_a5	7	24	32.3243	0.0306069	0.0315976	VREF on module 2
mod1_a6	8	24	32.3243	0.0315976	0.0315976	VTEMP on module 2
mod2_a5	11	25	32.3243	0.0325883	0.0315976	VREF on module 3
mod2_a6	12	25	32.3243	0.0306069	0.0315976	VTEMP on module 3

Initial Serenity Tests Summary

Results

Ran test based on the following code developed: [mux64 controller]

- Relied on the existing DB to run the lpgbt configuration

After debugging and running the test script we see the following outputs (whose values does not match tamalero)

```
[cmx@serenity-2368-15 lpbt_setup_mtd_daq]$ python -m
src.mtddaqsw.apps.S_mux64_test
api_url=http://localhost:8001/etl_chip_config/Master LPGBT
Success: Received configuration for Master LPGBT in Readout Board
Trying MUXCNT4, current_dir=1, gpio_port=1
Trying MUXCNT6, current_dir=1, gpio_port=2
Trying MUXCNT3, current_dir=1, gpio_port=3
Trying MUXCNT1, current_dir=1, gpio_port=4
Trying MUXCNT2, current_dir=1, gpio_port=5
Trying RESET1, current_dir=1, gpio_port=7
Trying LD_RSTN, current_dir=1, gpio_port=9
Trying RESET2, current_dir=1, gpio_port=10
Trying LD_DIS, current_dir=1, gpio_port=13
Trying LED_RHETT, current_dir=1, gpio_port=15
Trying MUXCNT5, current_dir=1, gpio_port=0
--->LPGBT Setting gpio direction to 1010011010111111
Trying MUXCNT4, current_dir=1, gpio_port=1
Trying MUXCNT6, current_dir=1, gpio_port=2
Trying MUXCNT3, current_dir=1, gpio_port=3
Trying MUXCNT1, current_dir=1, gpio_port=4
Trying MUXCNT2, current_dir=1, gpio_port=5
Trying RESET1, current_dir=1, gpio_port=7
Trying LD_RSTN, current_dir=1, gpio_port=9
```

```
Trying RESET2, current_dir=1, gpio_port=10
Trying LD_DIS, current_dir=1, gpio_port=13
Trying LED_RHETT, current_dir=1, gpio_port=15
Trying MUXCNT5, current_dir=1, gpio_port=0
api_url=http://localhost:8001/etl_chip_config/MUX64
Success: Received configuration for MUX64 in Readout Board
LV_RB - adc_port:0
Raw Voltage: 1023       Convreted Voltage:93.0
RT2 - adc_port:1
Raw Voltage: 1023       Convreted Voltage:511.5
slot0_a4 - adc_port:2
Raw Voltage: 712        Convreted Voltage:256.0
slot0_a5 - adc_port:3
Raw Voltage: 256        Convreted Voltage:256.0
slot0_a6 - adc_port:4
Raw Voltage: 712        Convreted Voltage:256.0
slot0_a7 - adc_port:5
Raw Voltage: 256        Convreted Voltage:256.0
slot1_a4 - adc_port:6
Raw Voltage: 256        Convreted Voltage:256.0
slot1_a5 - adc_port:7
Raw Voltage: 256        Convreted Voltage:256.0
slot1_a6 - adc_port:8
Raw Voltage: 720        Convreted Voltage:256.0
slot1_a7 - adc_port:9
Raw Voltage: 256        Convreted Voltage:256.0
slot2_a4 - adc_port:10
Raw Voltage: 256        Convreted Voltage:256.0
slot2_a5 - adc_port:11
Raw Voltage: 256        Convreted Voltage:256.0
slot2_a6 - adc_port:12
Raw Voltage: 256        Convreted Voltage:256.0
slot2_a7 - adc_port:13
Raw Voltage: 256        Convreted Voltage:256.0
```

> **Note:** For more information on other commands and changes attempted consult: mux 64 outputs

Conclusions

Since the raw ADC value is not matching it seems like the adcs might not be getting configured correctly on the lpGBT. We realized that some lpGBT registers corresponding to the ADCs are not set/defined in the DB. This needs to be updated and looked into, especially since the lpGBT uses a 10 bit adc the fact that we are reading all '1's for the LV_RB (9V+) and other floating signals having a high raw adc values suggests incorrect configuration settings

In Tamalero gain is calculated as a float to help convert the raw adc value to a translated voltage. In MTD DAQ gain is an integer written into an lpGBT register, this needs to be further investigated to clarify what setting Tamalero is using for this value.

Intermediate Steps

Verified against Tamalero software and realized the gain value MTD sets is equal to 0 on the ETL lpGBTs.

We moved away from relying on the DB to write the lpGBT configuration since the realization that not all lpgbt registers were included in the DB. To continue testing we dumped all lpGBT registers values using the KCU105 and Tamalero software and then created a CSV that held all register values and names.

Created temporary functions that read the csv file and wrote the full configuration to the lpgbt and verified the register values matched the expected Tamalero values.

Created a function that initialized the adc exactly as Tamalero does.

We wanted to test which of these changes resulted in achieving the correct raw adc readout for the mux64 output.

Final Serenity Tests Summary

Results

Ran test based on the following code developed: new mux64 controller

After debugging and running the test script we see the following outputs (whose values raw adc values are similar to tamalero)

```
[cmx@serenity-2368-15 mux64]$ python -m src.mtddaqsw.apps.S_mux64
Fetching config from: http://localhost:8001/etl_chip_config/Master LPGBT
Success: Received configuration for Master LPGBT
Fetching config from: http://localhost:8001/etl_chip_config/MUX64
Success: Received configuration for MUX64
RESERVED1
```

```
RESERVED2
reg=DACCONFIGH | 207, 0xcf
reg=CURDACVALUE | 28, 0x1c
reg=CURDACCHN | 1, 0x1
reg=VREFCNTR | 128, 0x80
reg=VREFTUNE | 99, 0x63
REG NOT SAME [VREFCNTR]: LPGBT=0x80, TAMALERO=0x1
RESERVED1
RESERVED2
REG NOT SAME [I2CM1CTRL]: LPGBT=0x0, TAMALERO=0xa
REG NOT SAME [I2CM2CTRL]: LPGBT=0x0, TAMALERO=0x6
reg=DACCONFIGH | 207, 0xcf
reg=CURDACVALUE | 28, 0x1c
reg=CURDACCHN | 1, 0x1
reg=VREFCNTR | 128, 0x80
reg=VREFTUNE | 99, 0x63
read_adc 1 15 0
read_adc 1 15 0
LV_RB - adc_port:0
Raw Voltage: 810    Convreted Voltage:0.07198080511863503
read_adc 1 15 0
read_adc 1 15 0
RT2 - adc_port:1
Raw Voltage: 501    Convreted Voltage:0.24535679374389052
read_adc 1 15 0
read_adc 1 15 0
slot0_a4 - adc_port:2
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot0_a5 - adc_port:3
Raw Voltage: 25 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot0_a6 - adc_port:4
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot0_a7 - adc_port:5
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
```

```
read_adc 1 15 0
read_adc 1 15 0
slot1_a4 - adc_port:6
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot1_a5 - adc_port:7
Raw Voltage: 24 Convreted Voltage:0.024437927663734114
read_adc 1 15 0
read_adc 1 15 0
slot1_a6 - adc_port:8
Raw Voltage: 24 Convreted Voltage:0.024437927663734114
read_adc 1 15 0
read_adc 1 15 0
slot1_a7 - adc_port:9
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot2_a4 - adc_port:10
Raw Voltage: 24 Convreted Voltage:0.024437927663734114
read_adc 1 15 0
read_adc 1 15 0
slot2_a5 - adc_port:11
Raw Voltage: 24 Convreted Voltage:0.024437927663734114
read_adc 1 15 0
read_adc 1 15 0
slot2_a6 - adc_port:12
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
read_adc 1 15 0
read_adc 1 15 0
slot2_a7 - adc_port:13
Raw Voltage: 24 Convreted Voltage:0.02346041055718475
```

Conclusions

The raw ADC output now match Tamalero given a resonable tolerance.

- **LV_RB**:
  - Tamalero value: 809
  - MTD value: 810

- **ADC_5**:
  - Tamalero value: 515
  - MTD value: 501
- **Unconnected modules**:
  - Tamalero value: 24
  - MTD value: 24

We now need to debug the translation of this value to a voltage. It turns out only the full writing of the lpGBT configuration from the csv is needed and the extra adc_init function is not needed as its values get set when writing the whole csv configuration.