CERN
CH-1211 Geneva 23
Switzerland

REV. 1.0 | VALIDITY Released

Date: August 30, 2024

Atlas ITk Common Electronics Qualification

Abstract

This report contains all the activities carried out at CERN as part of my Summer Studentship with the ATLAS team. Various tests have been conducted to validate the proper functioning of different BCM sensor components, including the LAPA and CALYPSO chips, as well as the bPol, optoboard, and Power Flex. Additionally, work was carried out to test and improve the EMRB, a board used for reading and storing data for the ATLAS ITk Environmental Temperature Monitoring system.

Valentin TIETZ

Supervised by: Carlos SOLANS, Ismet SIRAL

1 Introduction

I conducted several tests to validate the correct operation of the various BCM sensor components, including the LAPA and CALYPSO chips, as well as the bPol, optoboard, and Power Flex. I also worked on testing and improving the EMRB, a board used to read and store data for the ATLAS ITk Environmental Temperature Monitoring system. These activities provided me with the opportunity to learn how to test different parts of a setup step by step, improve my coding skills, and deepen my understanding of data analysis and statistics.

2 LAPA's H-bridges attenuation tests

2.1 Context

LAPA is a pseudo-LVDS driver designed in Tower 180 nm CMOS imaging technology for MALTA family of monolithic Pixel detectors. The LAPA test chip implements this driver, including several pads to change the settings of the driver. These tests follow on from the Single pad attenuation measurements carried out in part 3 of the report "LAPA configuration for BCM'" [1]. These tests consisted of testing a number of different configurations, each time placing a single jumper on the pins of the CTR TOP1 (input side) and CTR BTM1 (output side) headers, one at a time.

2.2 Tests setup

The test configuration is quite simple: a pulse generator sends a signal to the channel 5 input of a LAPA test chip mounted on its support. At the channel 5 output, a 100 ohm resistor is connected between the positive and negative of the LVDS [2]. Measurements are taken on an oscilloscope. A power supply is also used to power LAPA.

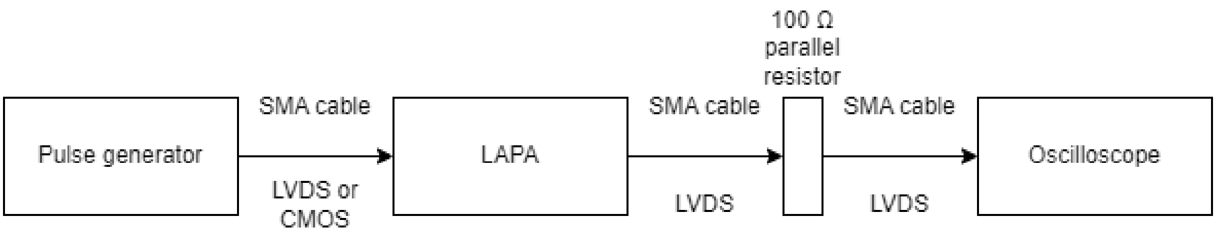

Figure 1: LAPA tests setup

2.3 Measurements

2.3.1 H-bridges attenuation measurement - LVDS mode

The tests are performed with 2 different configurations:

LVDS input / CMOS input

First, we test the LVDS mode, which is the board's default mode; there's nothing to set on the input.

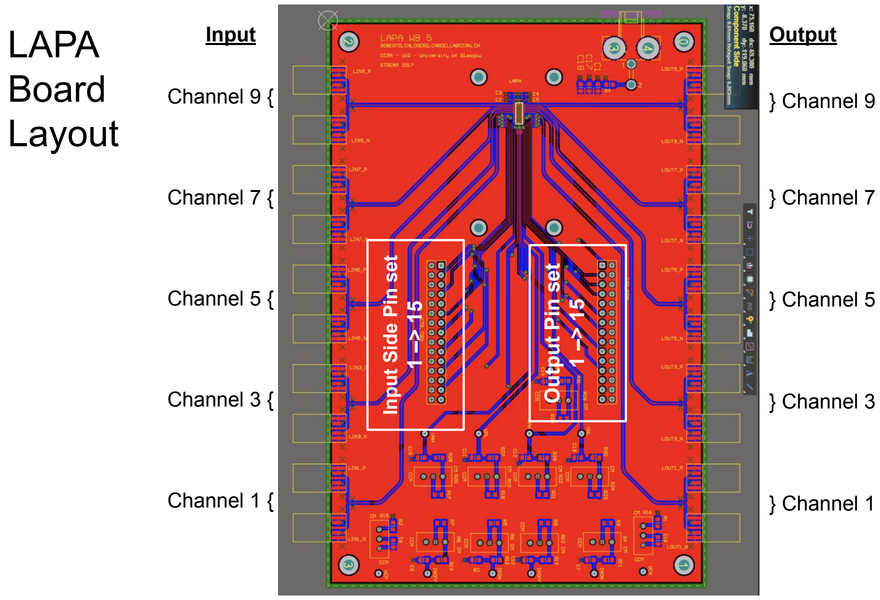

Figure 2: LAPA carrier board

The LVDS current at the output of channel 5 is then measured. An H-bridge is then deactivated by placing a jumper on the appropriate output pin (see figure 2). The measurement must be carried out for all configurations, from all disabled H-bridges to all enabled H-bridges. The output pin configuration vs the number of H-bridges disabled is shown in table 1

number of H-bridges disabled	0	1	2	3	4	5	6	7
Pin configuration [outputs 2-6]	00000	10000	11000	00001	10001	11001	11101	11111

Table 1: Output pins configuration vs number of H-bridges disabled (LVDS)

The input parameters used are listed in the table 2.

	LVDS	CMOS
frequency (kHz)	50	50
delay out1 (µs)	0	0
delay out2 (µs)	10	-
Width out1 (µs)	10	10
Width out2 (µs)	10	-
Amplitude (mV)	300	800

Table 2: Input settings

We obtain the results on figure 3. We take a margin of ± 5 percents for the current, due to the tolerance value of the RoHS 267-1733 100 ohms resistor used at the LAPA output to carry out the measurements.

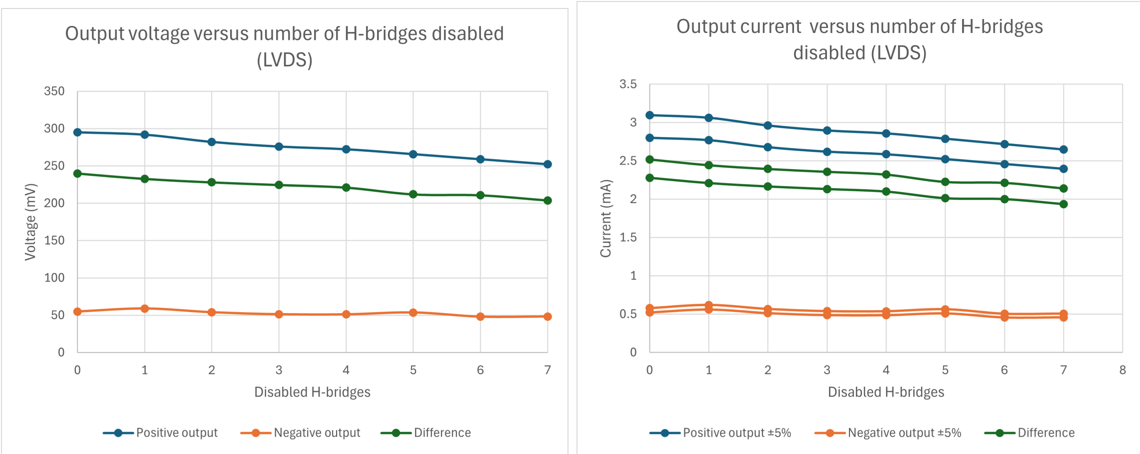

Figure 3: Voltage and current vs number of H-bridges disabled

Figure 3 shows the voltages measured by disabling the h-bridges.

2.4 H-bridges attenuation measurement - CMOS mode

Tests were also conducted in CMOS mode. To switch to CMOS mode, we're placing a jumper on the input pin 14 (see figure 2). The expected behavior of the CMOS mode hasn't been observed during these tests. Instead of getting an LVDS signal on the output, we're seeing a straight signal of approximately 290 mV for the positive pin and 60 mV for the negative one, independently of the inputs signal parameters. (see figure 4).

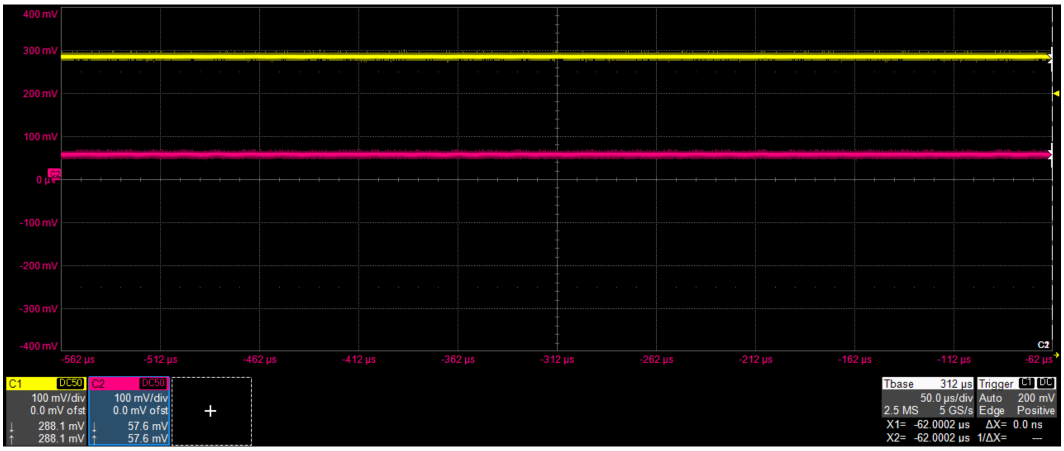

Figure 4: CMOS output signal channel 5

2.5 Conclusion

With an LVDS input, we can see that the power delivered is directly related to the number of H-bridges activated, which corresponds to the expected results. The relationship between both is proportional and linear. The CMOS mode was also tested, but did not behave as expected. The output doesn't seem to react to changes in input parameters, and always remains the same straight signal, with no variation in amplitude. As the board is not intended to be used with a CMOS-mode input, we'll stop testing here and continue later.

3 Optoboard and CALYPSO tests

3.1 Context

The CALYPSO chip is a front-end ASIC for the readout of the ATLAS ITk BCM' detector in the ATLAS experiment for the HL-LHC collider at CERN. The optoboard establishes communication between FELIX and CALYPSO, converting an electrical signal into an optical signal and vice versa.

3.2 Test setup

The test configuration is as follows: The optoboard communicates with FELIX, to which it is connected via optical fiber, and with CALYPSO [3] via an I2C connection. The optoboard is powered by 1.2V / 1.5A and 2.5V / 0.1A supplies, and CALYPSO by a 5V supply. (see figure 5)

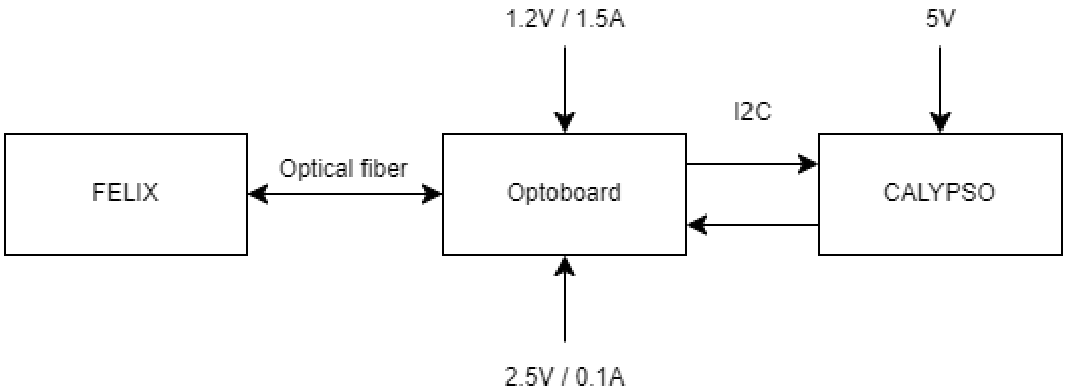

Figure 5: Optoboard / CALYPSO test setup

3.3 Results

For checking FELIX communication, we send a simple message (0x02) from FELIX, and check the optoboard behavior (figure 6).

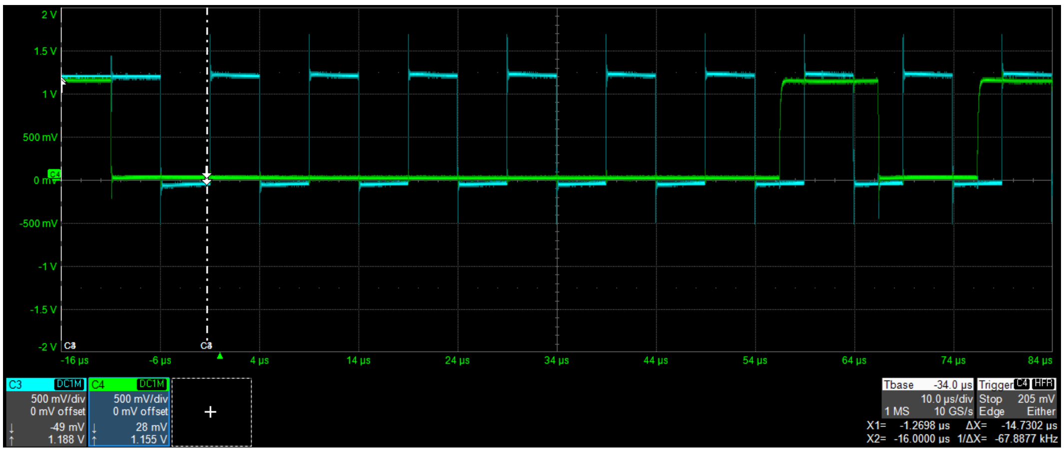

Figure 6: I2C signal received by the optoboard

In the figure above, you can see the I2C signal received by the optoboard (with CLK in blue and the sent message in green).

3.4 CALYPSO n°14 testing

The CALYPSO board used for the tests above was board n°2. We want to check that the board n°14, which is designed a bit differently, works properly. For this, we use a pulse generator to send a pulsed signal to CALYPSO, then we check the different board's outputs.

3.5 Output signals

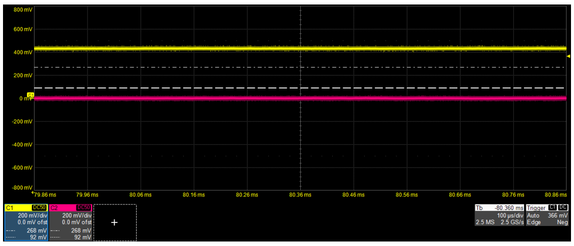

Figure 7: CALYPSO ch3 output

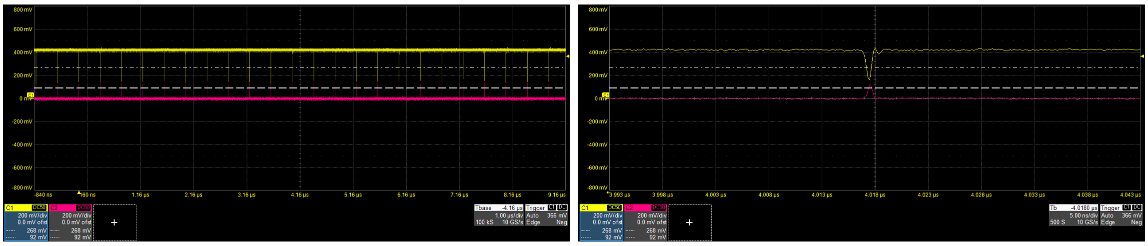

Figure 8: CALYPSO ch2 output

We can observe on figures 7 and 8 that the board don't behave as expected. The channel 2 provide a pulse, but we cannot change the width of the pulse, the amplitude isn't high enough and the signal should be square. And the channel 3 stay straight, no matter the settings.

4 bPOL / Power Flex testing

4.1 Context

The bPOL is a DC/DC converter 2.5V to 1.8V. It will be connected to the xPCB via the Power Flex, a flexible PCB, allowing to bend the connection tracks for the final product. The xPCB, the Power Flex and the bPOL are all parts of the BCM Prime [4]. These tests consists to check the correct behaviour of the Power Flex.

4.2 Test setup

The bPOL has been soldered on the Power Flex. Then, two banana plugs have been soldered on the GND and the input, and a 100 Ω resistor has been placed between the output and the ground (see figure 9 . Then the boards are supplied with 2.5 V and we check that the output value is correct. We then bend the Power Flex as it's supposed to be on final version of BCM (about 90°), and check the output again.

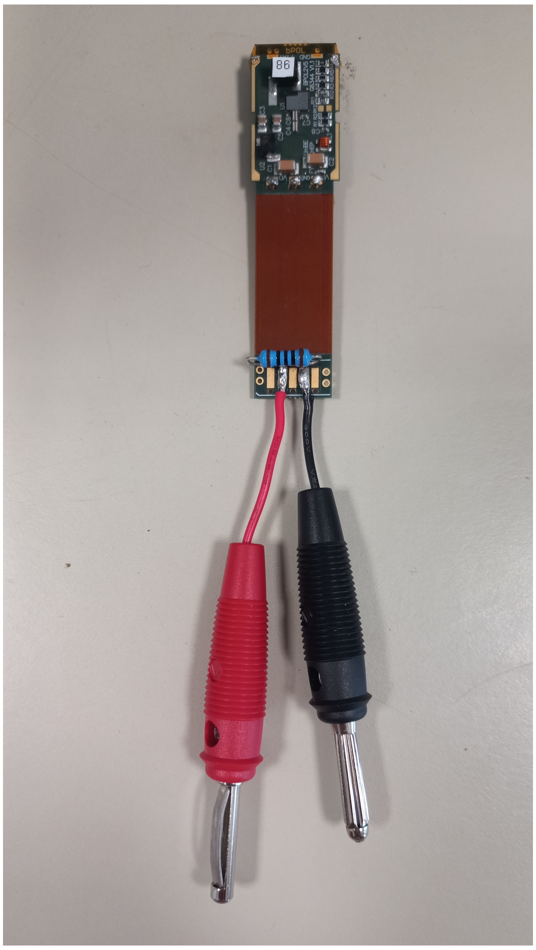

Figure 9: bPOL / Power Flex setup picture

4.3 Pins mapping

We can notice on figure 10 that the pins are inverted on the boards. We can still test the output and input behaviour, the pins will just be swapped for the tests. We decided to isolate the top pins (sense and GND) by wrapping them with Kapton tape to prevent create issues.

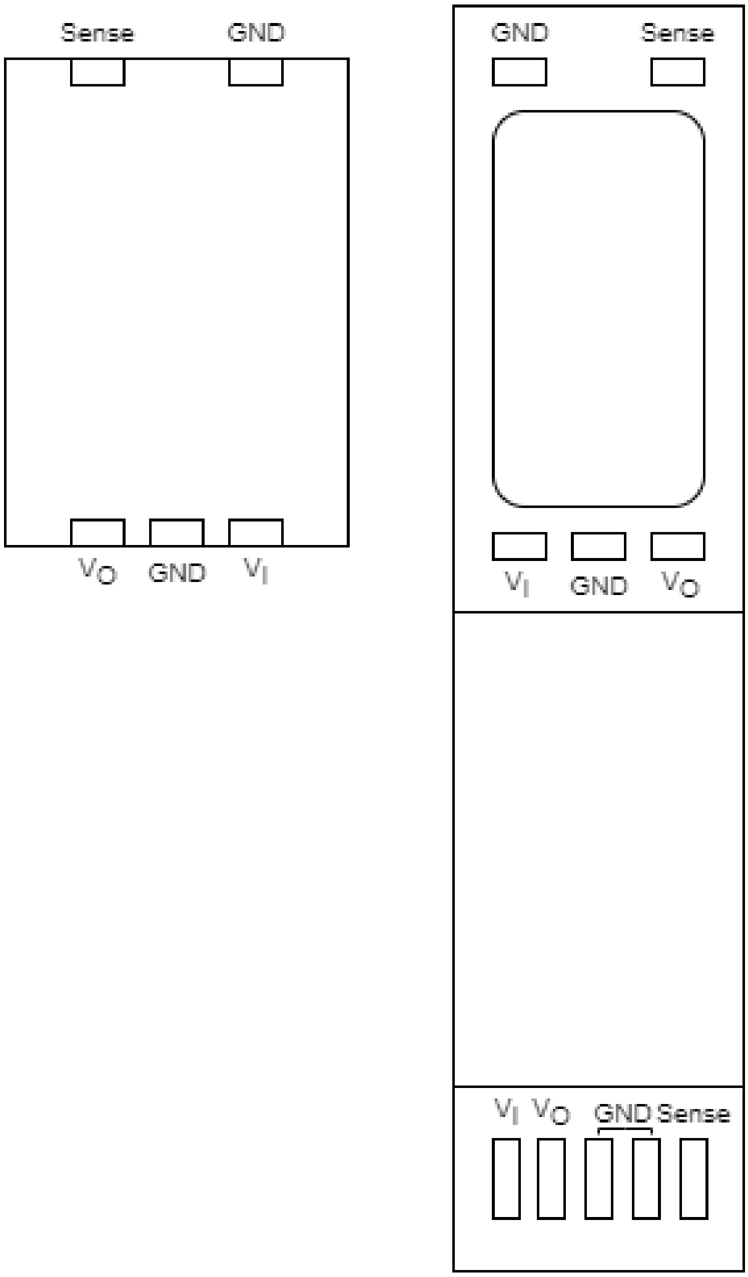

Figure 10: bPOL (left) and Power Flex (right) pins mapping

4.4 Results

The boards behave as expected. The output is about 1.78V, and the values remain the same when we bend the Power Flex. The tracks' resistance values also remain the same however we bend the board.

5 EMRB tests

5.1 Context

The Environmental Monitoring Readout Board (EMRB) is is designed to simplify PP2 Crate Backplane [5] data reading by replacing the ELMB card. This document describes the various test activities carried out in using the Environmental Monitoring Readout Board (EMRB). This new board is a simplified ethernet-based reader board for assembly sites, with an integrated OPC-UA server to replace the ELMB-CB. The board hosts a RaspberryPI and encapsulates all the following functionalities: ELMB-CB -> CAN-bus line -> USB converter -> PC server -> network. The RPi is capable of interfacing directly with the DCS system via the network (see figure 11).

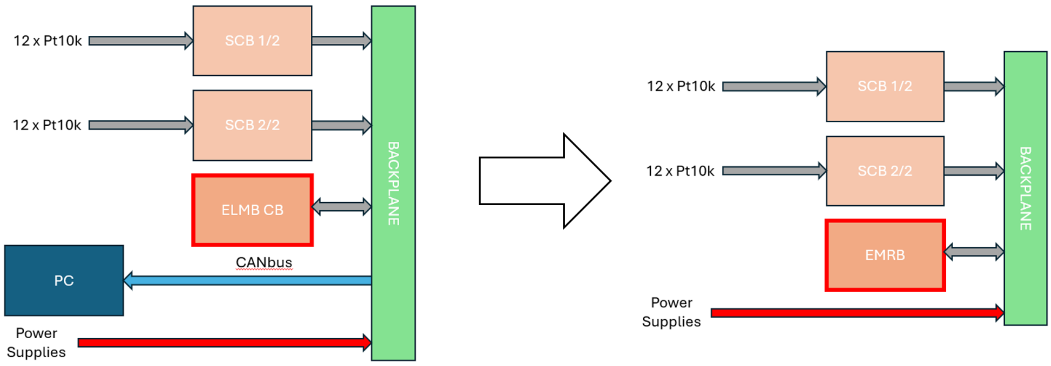

Figure 11: EMRB replacing ELMB for the PP2 Crate backplane reading

The EMRB is composed of multiplexers, DC/DC converters, a Raspberry Pi 4 model B and an ADC converter CS5523 - ASZ. The purpose of this converter is to convert the analog temperature value received from the multiplexers, and to convert it to a digital value which can be stored into the Raspberry. The point is that for now, there is no communication between the Raspberry and the CS5523. The Raspberry is emitting to the ADC converter, but this one doesn't respond. The purpose of these tests is to establish communication between both and check that the code is working as it should.

5.2 Functionality of the EMRB

1. Temperature value measured and transmitted to the input of one of the EMRB multiplexers.
2. The ADC converter reads the values from the multiplexers and converts the analog value into a digital value.
3. The converted value is transmitted by the A/D converter to the Raspberry PI / The Raspberry PI then runs a server that can directly interface with the DCS (see figure 12).

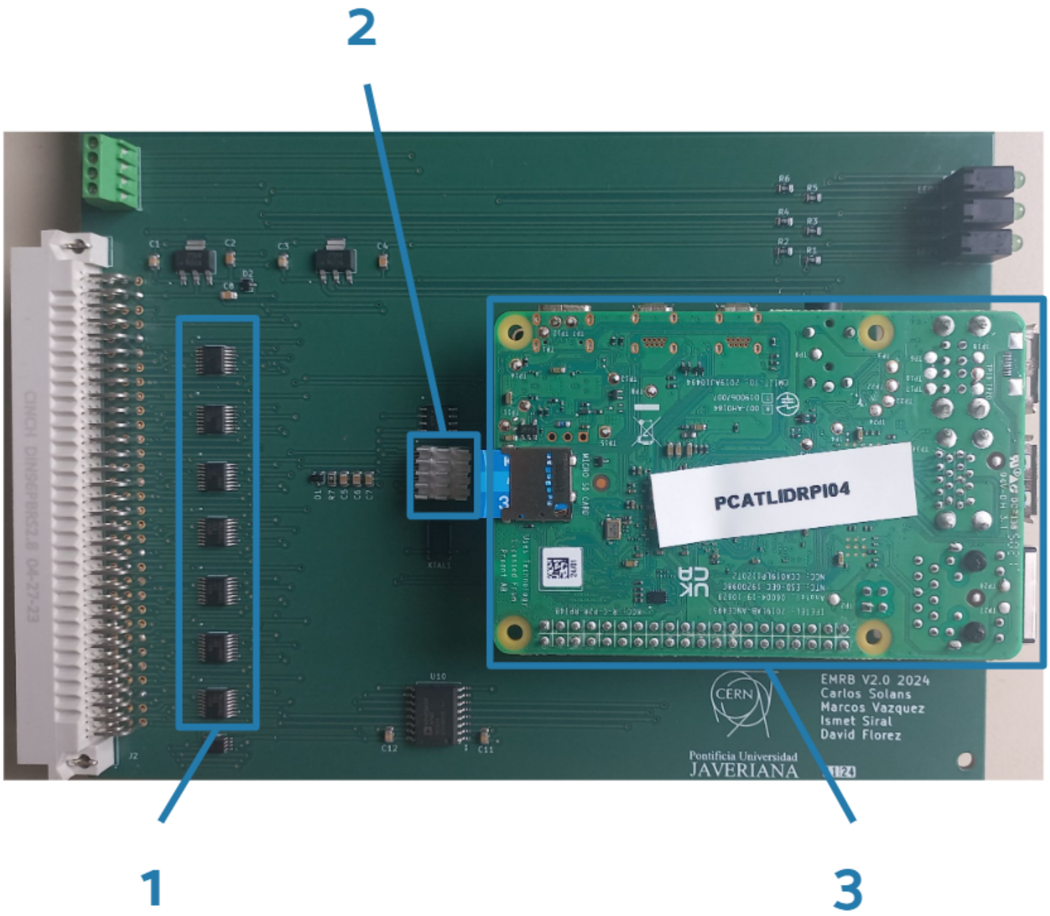

Figure 12: EMRB functioning description

5.3 Hardware checking

First, we checked the conductivity of each track between the Raspberry and the CS5523. Then, we checked that the 3.3V output from the Raspberry provides enough power to powered the CS5523 and the digital isolator ADUM3400ARWZ. We don't have information about the maximum power that the 3.3V output of the Raspberry Pi can provide, but estimates oscillate between 50 and 500 mA, representing 165 to 1650 mW., We can estimate the maximum consumption of the two chips to less than 85mW. Theoretically, the 3.3V output should provide enough power to powered both chips. Last, we try to connect our Raspberry to an other board, to check if the problem isn't coming from the board.

5.4 SPI communication

For testing SPI communication, we select a random pin, by choosing multiplexer and an ADC channel value (here multiplexer 0 and channel 3). Then we run a program we wrote for reading the pin value every second. (see figure 13).

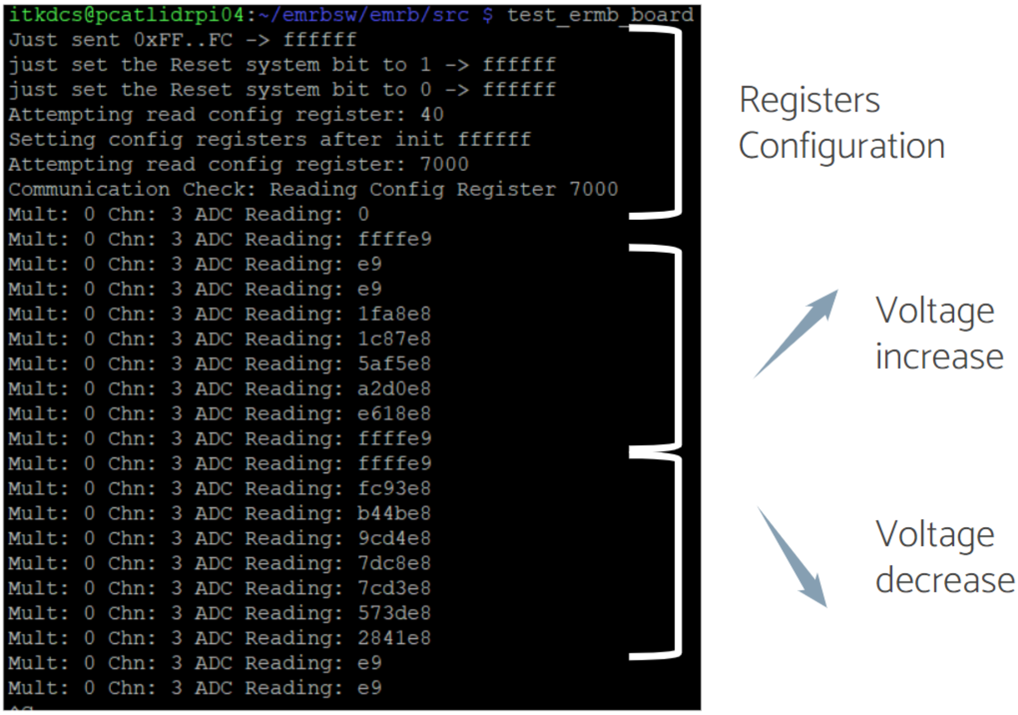

Figure 13: Communication test between Raspberry Pi and ADC converter

As you can see in the figure below, first we display the register configuration; then we increase the voltage value from 0V to Vmax (the displayed value is a hexadecimal value); finally, we decrease it back to 0V.

Next, we select several multiplexers and ADC channels and repeat the test, to ensure that the program works for each pin.

5.5 Input pins mapping

Two parameters are used to select a specific pin: the ADC channel and the multiplexer channel. Based on the EMRB diagram, we were able to determine the pin mapping (see figure 14).

Multiplexor	Mult_channel	SEL_4	SEL_3	SEL_2	SEL_1	ADC_channel
U1	SCB_A_1	0	0	0	0	00
	SCB_A_2	0	0	0	1	00
	SCB_A_3	0	0	1	0	00
	SCB_A_4	0	0	1	1	00
	SCB_B_1	0	0	0	0	10
	SCB_B_2	0	0	0	1	10
	SCB_B_3	0	0	1	0	10
	SCB_B_4	0	0	1	1	10
U2	SCB_A_5	0	0	0	0	01
	SCB_A_6	0	0	0	1	01
	SCB_A_7	0	0	1	0	01
	SCB_A_8	0	0	1	1	01
	SCB_B_5	0	0	0	0	11
	SCB_B_6	0	0	0	1	11
	SCB_B_7	0	0	1	0	11
	SCB_B_8	0	0	1	1	11
U3	SCB_A_9	0	1	0	0	00
	SCB_A_10	0	1	0	1	00
	SCB_A_11	0	1	1	0	00
	SCB_A_12	0	1	1	1	00
	SCB_B_9	0	1	0	0	10
	SCB_B_10	0	1	0	1	10
	SCB_B_11	0	1	1	0	10
	SCB_B_12	0	1	1	1	10
U4	SCB_A_13	0	1	0	0	01
	SCB_A_14	0	1	0	1	01
	SCB_A_15	0	1	1	0	01
	SCB_A_16	0	1	1	1	01
	SCB_B_13	0	1	0	0	11
	SCB_B_14	0	1	0	1	11
	SCB_B_15	0	1	1	0	11
	SCB_B_16	0	1	1	1	11
U5	SCB_A_17	1	0	0	0	00
	SCB_A_18	1	0	0	1	00
	SCB_A_19	1	0	1	0	00
	SCB_A_20	1	0	1	1	00
	SCB_B_17	1	0	0	0	10
	SCB_B_18	1	0	0	1	10
	SCB_B_19	1	0	1	0	10
	SCB_B_20	1	0	1	1	10
U6	SCB_A_21	1	0	0	0	01
	SCB_A_22	1	0	0	1	01
	SCB_A_23	1	0	1	0	01
	SCB_A_24	1	0	1	1	01
	SCB_B_21	1	0	0	0	11
	SCB_B_22	1	0	0	1	11
	SCB_B_23	1	0	1	0	11
	SCB_B_24	1	0	1	1	11
U7	SCB_A_25	1	1	0	0	00
	SCB_A_26	1	1	0	1	00
	SCB_B_25	1	1	0	0	10
	SCB_B_26	1	1	0	1	10

Figure 14: EMRB pins mapping

5.6 Linear conversion testing

To check that the ADC conversion is a linear function, we conducted the following test : we select different voltage values from 0V to Vmax, we read the hexadecimal values and convert them to decimal value,then we display the curve and the equation of the function.

Voltage (mV)	ADC values (hex)	ADC values (dec)
0	E9	233
250	1AE4D6	1762519
500	35C8F3	3524851
750	50AC8B	5287051
1000	6B8E5B	7048796
1250	8674D0	8811729
1500	A158DA	10574042
1750	BC3C4D	12336205
2000	D7200E	14098446
2250	F20138	15860024
2370	FE84E9	16680169
2380	FFFFE9	16777193

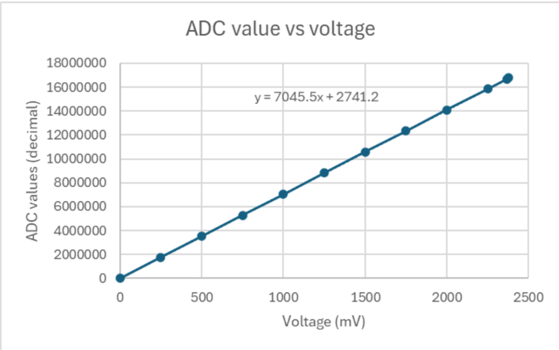

Figure 15: ADC value VS voltage

As you can see in figure 15, the ADC conversion is a linear function.

5.7 Conversion test

We've written functions to convert the hexadecimal values into voltage and temperature values and display them. We've also written a function to display the pin name.

```
SCB_B_16
Voltage = 1.25792 V
Temperature = 75.2193 °C
```

Figure 16: Pins readout new display

The figure 16 above shows the new display (here with the SCB_B_16 pin reading).

5.8 OPC-UA server creation

We've created an OPC-UA server with Quasar. This server is used to store all parameters and events, and to allow the user to access these values. The EMRBBoard class contains the variables waitTime and channelReading, and the channel class contains the variables channelNumber, channelName, datAndTime, ADCCalibrationFactor, enable, ADCValue, voltage and temperature.

When we run the program, the values are updated as we go along and stored in the server.

```
Chan: B_17, channelNumber: 43, dateAndTime: 1723455240, ADCCalibrationFactor: 1, enable: True
, ADCValue: 3134440.0, Voltage: 1.5424201488494873, Temperature: 88.16968032349126,
Chan: B_16, channelNumber: 42, dateAndTime: 1723455235, ADCCalibrationFactor: 1, enable: True
, ADCValue: 16777199.0, Voltage: -1.0, Temperature: -1.0,
```

Figure 17: Server reading

The figure 17 above shows an extract from the server display. As you can see, pin SCB_B_17 returns temperature and voltage values because we're supplying voltage on this pin, and pin SCB_B_16 returns minus one, which is the error value, because nothing is connected on this pin.

5.9 Backplane power supply

Due to the overheating of the voltage regulators, we decided to add heat sinks and a fan to dissipate the heat emitted. We also connected the positive and negative pins of the external power supply together. Next, we connected to the board multiple times, and ran the pin reading for several hours, to check the stability of the connection.

5.10 ADC calibration factor definition

The ADC calibration factor is used to convert the digital value into a voltage value. To define this factor, several digital values are collected for different voltage values for each of the pins. We can then plot the affine functions corresponding to the values collected (see chart 18).

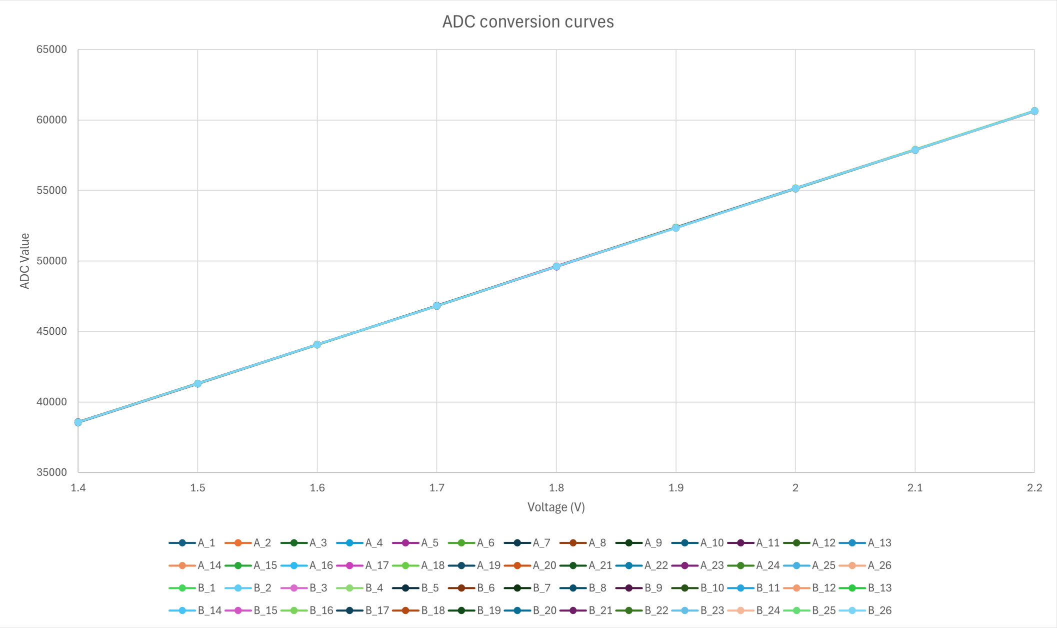

Figure 18: ADC conversion curve

We then calculated the slope and intercept for all the measurements. We can then plot the Gauss curve for each of these 2 values. (see charts 19).

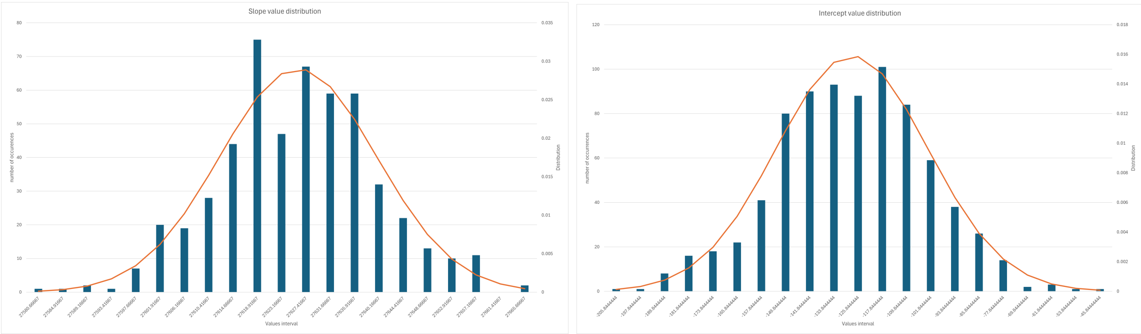

Figure 19: Slope value and intercept value distribution

We obtain for the slope a mean of 27626.1, a standard deviation of 13.7 and a standard error of 0.603, and for the intercept a mean of -127.9, a standard deviation of 25.1 and a standard error of 1.1.

5.11 Tests of other boards

Four other EMRB have been manufactured. After testing them, it appears that none of them behaves as expected. We read the SPI signal on the Raspberry, and for now, the results are as follows (in each figure, the CLK signal is represented in pink, the MOSI signal in blue and the MISO signal in yellow):

- The falling edges of the MISO signal from the EMRB2.0_2 board are not clear, but decrease little by little, which suggests an unexpected inductive phenomenon (see figure 20).

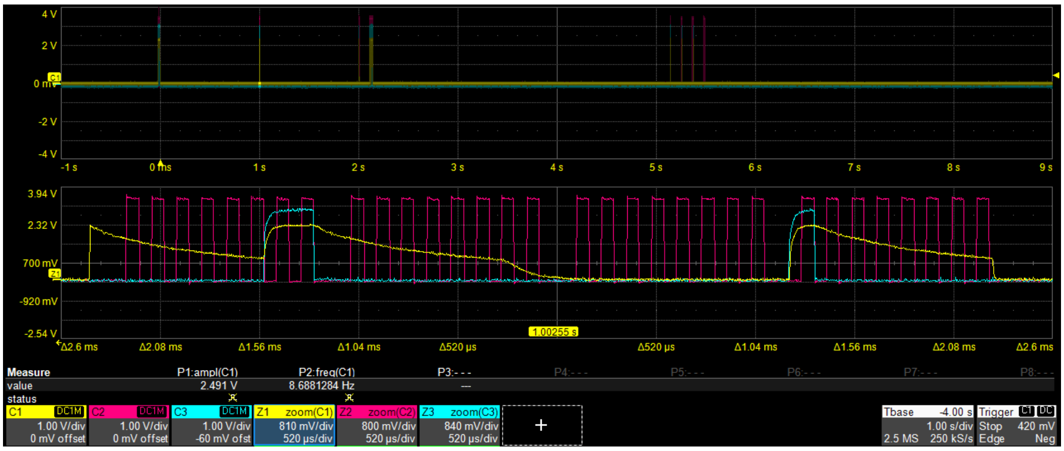

Figure 20: EMRB2.0_2 MISO, MOSI and CLK signals

- EMRB2.0_5 board never receives any MISO signal (see figure 21).
- The configuration signals of the EMRB2.0_3 and EMRB2.0_4 cards are the same as those of the card used throughout the tests, however, during reading, the signal read is always in the high state.

We guess that boards 2 and 5 could have been damaged during RPI connector changes. We need to focus on boards 3 and 4 for further investigation.

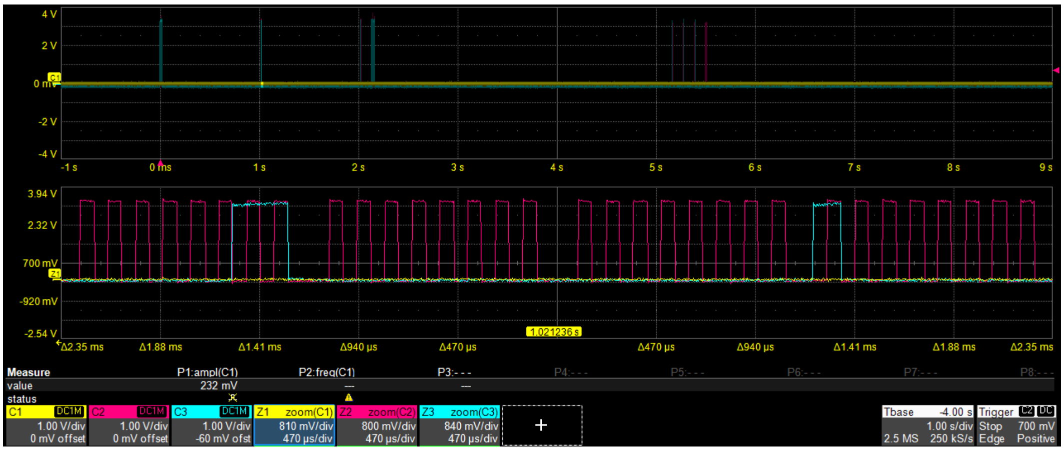

Figure 21: EMRB2.0_5 MISO, MOSI and CLK signals

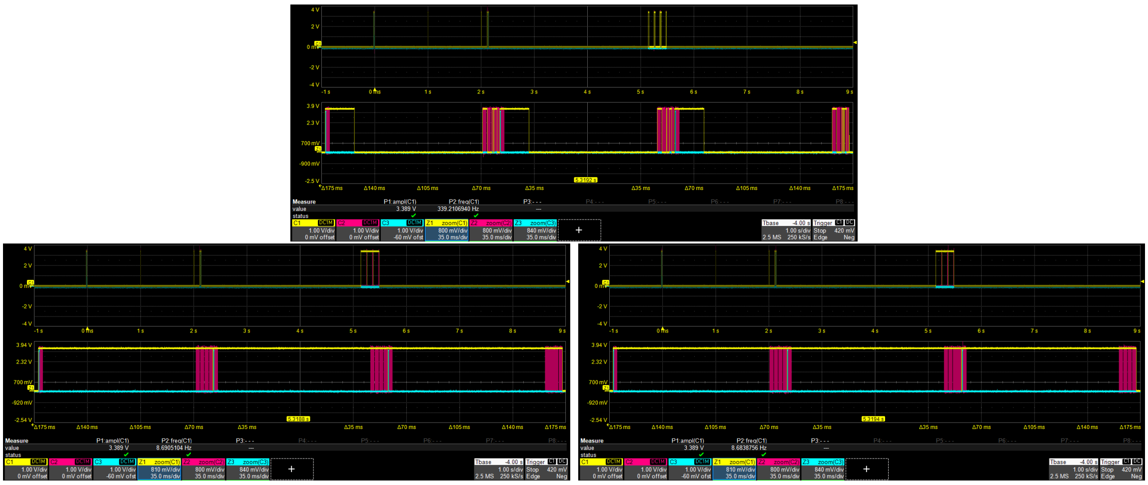

Figure 22: Signal of ADC conversion of EMRB2.0_3 and EMRB2.0_4 cards vs. initial signal

5.12 Conclusion

For the ERMB boards where SPI communication properly works, all EMRB functions have been tested and found to be working correctly. We created an OPC server to enable us to store and display all values as they should be. We can conclude that the EMRB is now ready to operate under real-world conditions. However, the other boards, designed in the same way, are not working at the moment. The next step will be to correct the ADC reading of the other boards, determine the temperature-dependent voltage calibration factor, and check whether the board behaves as it should when implemented in the system and reads the PP2 Crate backplane values.

References

[1] I. Siral *et al.*, “LAPA Configuration,” Tech. Rep. EDMS 3091836, CERN, 2024.

[2] IEEE, “Standard for Low-Voltage Differential Signals (LVDS) for Scalable Coherent Interface (SCI),” *STD 1596.3-1996*, pp. 1–34, 1996.

[3] R. Teuscher *et al.*, “Calypso specifications document,” Tech. Rep. EDMS 2369649, CERN, 2020.

[4] I. Asensi Tortajada *et al.*, “Development of the BCM prime system and readout for ATLAS,” *PoS*, vol. 448, no. 023, 2024.

[5] D.M. Flores Rubio *et al.*, “ITk DCS PP2 Crate,” Tech. Rep. EDMS 3118743, CERN, 2024.