CE-Adapter Board Functionality Quality Assurance

Krishna Lakkaraju[a], Andrew Arbuckle[b], Pamela Marr-Laundrie[b], and Brian Rebel[a]

[a]Department of Physics, University of Wisconsin-Madison, WI, USA
[b]Physical Sciences Laboratory, 3725 Schneider Dr, Stoughton, WI, USA

* *E-Mail: Klakkaraju@wisc.edu*

Abstract

The Cold Electronics (CE)-Adapter board is a critical component of the Anode Plane Assembly (APA). These boards are designed to pass ionization signals form the APA to the cold electronics. At the Physical Sciences Laboratory (PSL) of the University of Wisconsin-Madison, blank CE-Adapter boards are populated with their respective connectors. A dedicated CE-Adapter board test ensures the proper functionality of these boards.

Contents

1. Introduction

Each board's front face is fitted with two 96-pin male connectors, inserted through the front and soldered from the back, as seen in Figure 1. The back face houses two 72-pin connectors with right-angle pins, inserted through the back and soldered from the front ,as seen in Figure 2. These connectors facilitate signal transmission, with the 72-pin connectors receive signals from the CR board and the 96-pin connectors route signals to the CE box.

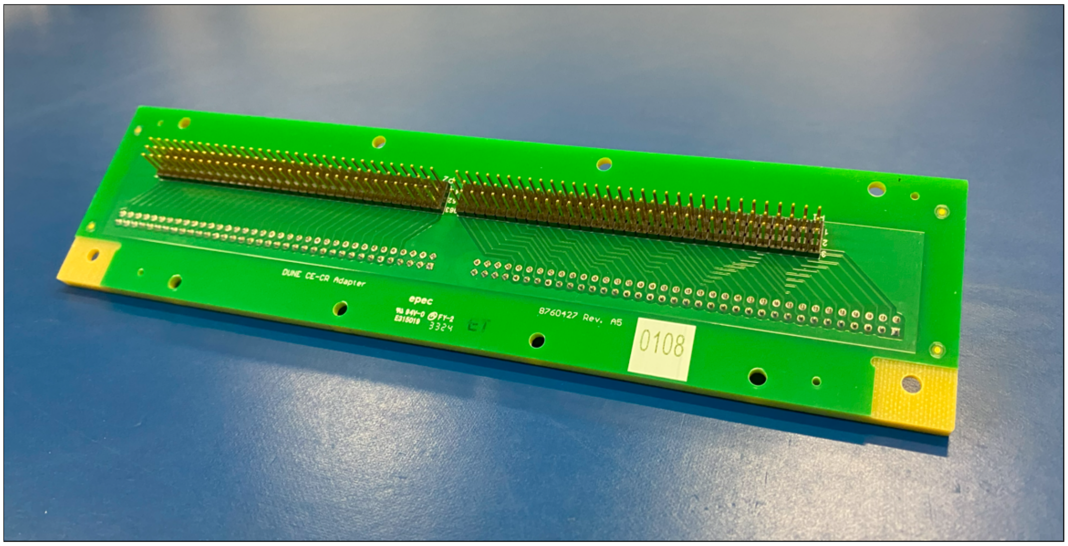

Figure 1. The front face of a populated CE-Adapter board.

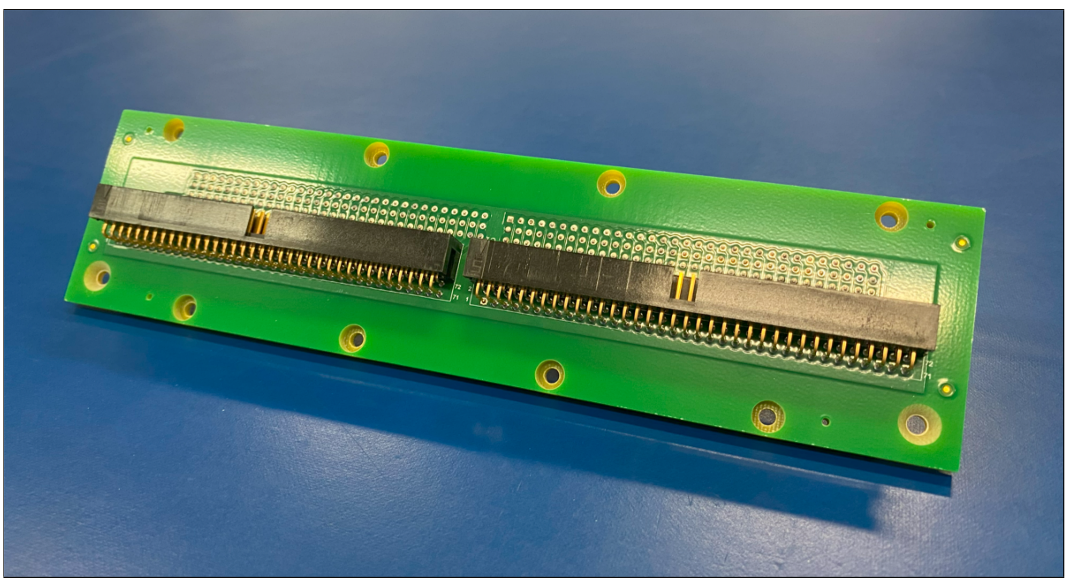

Figure 2. The back face of a populated CE-Adapter board.

2. Test Stand Hardware & Design

2.1. Components

The CE-Adapter board test bench incorporates various components to perform an isolation and continuity tests, as explained in §2.2 & 2.3. Two Arduino MEGA devices serve as the primary processing units. Upon initiation of the software by a technician on a laptop, the Arduinos execute onboard scripts to conduct the tests. The Arduino MEGA is particularly suited for this application due to its ability to generate and detect five-volt signals. Given the 336 pins to be tested distributed across the four connectors of the adapter board and the Arduino's limit of 54 pins, the CD74HC4067 Digi-Key multiplexer breakout boards, as shown in Figure 3, are used to expand the available output signals. The multiplexers function as data selectors, utilizing four-bit signals to switch between sixteen output channels. Each board houses an enable (EN) pin as a global on/off switch for all of the output pins on the breakout board. Each multiplexer also has a signal (SIG) pin which either drives $5V$ or is set to receive signal from the pins on the CE-Adapter board. These capabilities allows the Arduino to probe each pin on the connectors effectively. The multiplexer outputs are connected to the pins on the adapter board connectors through ribbon cables.

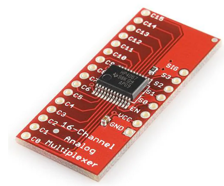

Figure 3. CD74HC4067 Digi-Key multiplexer breakout board (see [1] for specifications).

2.2. Isolation Test

During fabrication, a selective soldering machine is used to secure the connectors to the adapter board efficiently. While this method is more efficient and reliable than manual soldering, there remains a small risk of solder balls forming under or around the solder pads, potentially causing unintended electrical connections between adjacent pins. To mitigate this issue, an isolation test is conducted on the. In this test, the Arduino probes a pin on a given connector and then checks adjacent pins for any detected signal. If a probed pin and one of its adjacent pins exhibit continuity, the two pins are identified as electrically connected. The Arduino software iterates through all possible adjacent pin-pairs.

The middle row of the 96-pin connector is interconnected to the same ground for redundancy. In addition, the first eight pins of the J1 72-pin connector are grounded together, as shown in table 1. Thus, when one of these grounded pins is probed, we should see signal on the surrounding grounded pins. The Arduino software accounts for this by considering electrical continuity between two pins as expected behavior.

2.2.1. The Voltage Decay Problem

Each multiplexer will be connected to at least one pair of adjacent pins, requiring it to alternate between driving a $5V$ signal and switching to "input" mode to check for a signal on the adjacent pin. Since a single multiplexer cannot simultaneously drive and read signals from two pins, adjacent or otherwise, there will be a brief interval during which the signal is not driven while adjacent checks are performed. The voltage decay during this interval can be modeled by the equation:

$$V(t) = V_0 e^{-t/\tau} \tag{1}$$

where τ is the RC time constant determined by the impedance (R) and the parasitic capacitance (C). We find that $R \approx 10M\Omega$ as the effective resistance due to the small leakage currents and the high input impedance of the Arduino pins. While the multiplexer's resistance is only $45 - 70\Omega$ when actively driving a signal during a test [3], when the Arduino stops driving signal to begin the check it is limited by minuscule leakage currents (nA range) and the Arduino's high input impedance. Contrastingly, the total capacitance of the circuit associated with each pin is influenced by the amount of surrounding conductive material. Beginning with the multiplexer's $10pF$ capacitance in the "on-path" [3], we chose a conservative total estimate $C_{total} \approx 40pF$ after considering the nearby wiring and components that might contribute to the parasitic capacitance. So we approximate $\tau \approx 400\mu s$ Since an Arduino will typically read a signal as "HIGH" if it is at least $2.5V$, equation (1) tells us $.28ms$ is the maximum time that it should take to conduct a check of adjacent pins before the voltage decay affects the test reading. Since the 'DigitalRead()' Arduino function takes anywhere from $5 - 6\mu s$ we approximate the time for each adjacent pin check to be $10\mu s$ to account for any extra software delays. Thus, the voltage decay is unlikely to pose an issue as the pin check time is much shorter than the allotted time by (1). To further mitigate any possible voltage decay, the isolation test software will incorporate interspersed iterative pulses rather than testing all adjacent pins sequentially. After each adjacent pin check, the Arduino will drive a pulse just long enough for the voltage to stabilize. By introducing intermittent pulses, the effects of voltage decay on the tests are effectively negated.

2.3. Mapping Test

To verify the connections between the 72-pin and 96-pin signals, the CE-Adapter board test bench employs a mapping test powered by the Arduino software. During this process, the Arduino drives a 5-volt signal to a designated indexed pin and checks for the signal on the corresponding pin of the paired connector described in Table 1 & 2. The software systematically cycles through all pin mappings, ensuring the integrity of the connections. For grounded pins on the 96-pin connector, no signal should be detected on the corresponding pins of the 72-pin connector during probing, verifying proper grounding and signal isolation. This test serves to identify boards with cut traces or irregular electrical mappings, adding a layer of redundancy to the mapping verification process. Additionally, if two pins are incorrectly mapped, the resulting failure in electrical isolation will be detected during testing. Voltage decay is not a concern in the mapping test, as the 5-volt signal can be continuously driven through one pin while the corresponding pin is simultaneously probed by a separate multiplexer.

72-Pin	96-Pin	Layer
1	GND	N/A
2	GND	N/A
3	GND	N/A
4	GND	N/A
5	GND	N/A
6	GND	N/A
7	GND	N/A
8	GND	N/A
9	3	X26
10	1	X25
11	6	X28
12	4	X27
13	9	X30
14	7	X29
15	12	X32
16	10	X31
17	15	X34
18	13	X33
19	18	X36
20	16	X35
21	21	V22
22	19	V21
23	24	V24
24	22	V23
25	27	V26
26	25	V25
27	30	V28
28	28	V27
29	33	V30
30	31	V29
31	36	U22
32	34	U21
33	39	U24
34	37	U23
35	42	U26
36	40	U25
37	45	U28
38	43	U27
39	48	U30
40	46	U29
41	51	X38
42	49	X37
43	54	X40
44	52	X39
45	57	X42
46	55	X41
47	60	X44
48	58	X43
49	63	X46
50	61	X45
51	66	X48
52	64	X47
53	69	V32
54	67	V31
55	72	V34
56	70	V33
57	75	V36
58	73	V35
59	78	V38
60	76	V37
61	81	V40
62	79	V39
63	84	U32
64	82	U31
65	87	U34
66	85	U33
67	90	U36
68	88	U35
69	93	U38
70	91	U37
71	96	U40
72	94	U39

Table 1. Formatting of the **J1** 72-pin to a 96-pin signal, as described in [2].

72-Pin	96-Pin	Layer	72-Pin	96-Pin	Layer	72-Pin	96-Pin	Layer
1	3	X2	25	39	U4	49	75	V16
2	1	X1	26	37	U3	50	73	V15
3	6	X4	27	42	U6	51	78	V18
4	4	X3	28	40	U5	52	76	V17
5	9	X6	29	45	U8	53	81	V20
6	7	X5	30	43	U7	54	79	V19
7	12	X8	31	48	U10	55	84	U12
8	10	X7	32	46	U9	56	82	U11
9	15	X10	33	51	X14	57	87	U14
10	13	X9	34	49	X13	58	85	U13
11	18	X12	35	54	X16	59	90	U16
12	16	X11	36	52	X15	60	88	U15
13	21	V2	37	57	X18	61	93	U18
14	19	V1	38	55	X17	62	91	U17
15	24	V4	39	60	X20	63	96	U20
16	22	V3	40	58	X19	64	94	U19
17	27	V6	41	63	X22	65	GND	N/A
18	25	V5	42	61	X21	66	GND	N/A
19	30	V8	43	66	X24	67	GND	N/A
20	28	V7	44	64	X23	68	GND	N/A
21	33	V10	45	69	V12	69	GND	N/A
22	31	V9	46	67	V11	70	GND	N/A
23	36	U2	47	72	V14	71	GND	N/A
24	34	U1	48	70	V13	72	GND	N/A

Table 2. Formatting of the **J2** 72-pin to a 96-pin signal, as described in [2].

2.4. Electrical Diagrams

It is essential to understand how electrical connections are established between the components on the test stand to ensure that the software accurately identifies failure modes when they occur. This understanding also facilitates repeatable construction of the test stand. For instance, the connectors on the ribbon cable, which interface with each test board, will require periodic replacement due to the high insertion force and the gradual deformation of the springs in the female receptacles.

Figures 4 and 5 provide electrical diagrams to guide any repairs or replacements. As indicated, each multiplexer has the same set of nodes: S0-S3 for input signal, SIG for switching modes between sending and receiving, EN as an on/off switch to the output nodes, and C0-C15 for output. Each SIG and EN pin must be connected to its own unique pin on the Arduino so that the software can individually communicate with each multiplexer. This allows many multiplexers to be active at the same time for parallel processing. Since a multiplexer will only access one pin at a time, the signal pins used for a specific connector, as indicated at the top and bottom of each drawing, may be connected to the same Arduino pins.

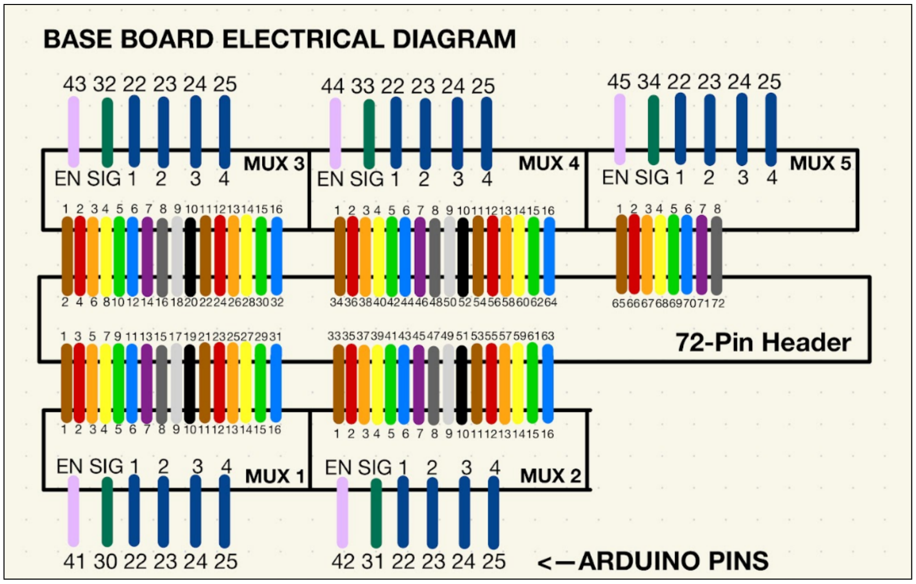

Figure 4. Electrical diagram for each of the 72-pin connectors on the baseboard of the test stand and how they interface with the Arduino pins.

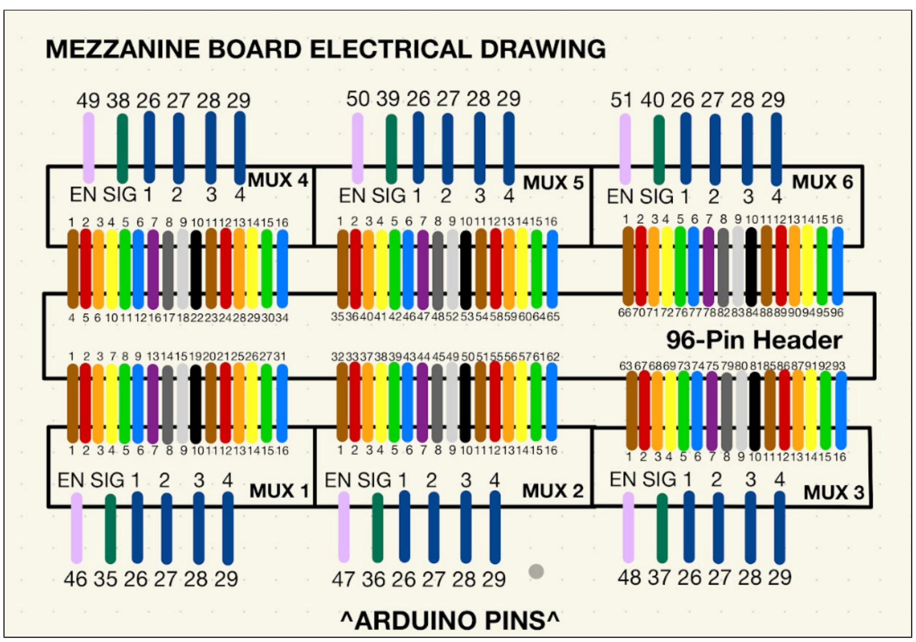

Figure 5. Electrical diagram for each of the 96-pin connectors on the baseboard of the test stand and how they interface with the Arduino pins.

3. Analysis Software

The analysis software for the test stand is initialized through the technician's GUI and executed by two Arduinos on the bench. The GUI software, written in Python (§ 6), incorporates visualization packages that simplify the execution of Arduino scripts. The code executed by the Arduinos is optimized for runtime efficiency to minimize inspection duration and errors caused by voltage decay, as discussed in §2.2.1. When the inspection begins, both Arduinos run the same script simultaneously, using two distinct pin maps. These pin maps define the relationship between the expected signals for a given pin and any corresponding pin on either connector, enabling their use in both the isolation and mapping tests. Additionally, the software utilizes an adjacent pin map that specifies which adjacent pins share the same multiplexer. The software uses these pin maps to activate the appropriate multiplexer for driving a signal and to enable corresponding multiplexers to check for signals on either adjacent or mapped pins. It cycles through all pins on each connector before concluding the test. If a failure is detected, the corresponding Arduino outputs an error message to its "Serial Output", which the GUI captures and records in the test log.

4. User Guide

The purpose of this section is to provide step-by-step instructions for technicians analyzing a populated CE-Adapter board using the test bench.

1. To begin, power on and sign into the laptop connected to the test bench and the Arduino COM ports.
2. Launch the software located at C:\Users******\Desktop\CE BOARD TEST GUI.
3. Next, verify the output directory for log files, which determines where test data will be saved.
   (a) To change the directory, click "Change Directory," navigate to the desired folder, and select "Select Folder."
   (b) Once the log directory is configured, enter the serial number of the current test board into the text box. The serial number must consist of a series of integers (e.g., "125"). If the input is invalid, an error message will be displayed.
4. After entering a valid serial number, press "Run Test" to initiate the analysis.

The test should take less than 20 seconds to complete. Upon test completion, the GUI will display a green box indicating a pass or a red box indicating a failure. For a pass, an entry will be added to the log file, including the serial number, time of the test, and a message stating that the board "passed" the tests. For a failure, an entry will include the serial number, time of the test, the type of failure (e.g., isolation or mapping), and the specific pins involved. Example log entries are as follows:

- Isolation failure: "Serial Number: 1234, Time of Test: 11/27/24 at 10:30 AM, Failed, Isolation, Unexpected continuity between Pin X and Pin Y on the 72-pin connector."
- Mapping failure: "Serial Number: 1234, Time of Test: 11/27/24 at 10:30 AM, Failed, Mapping, Unexpected mapped signal from (72) pin X to (96) pin Y, expected signal map: (72) pin X to (96) pin Z."
- Pass: "Serial Number: 1234, Time of Test: 11/27/24 at 10:30 AM, Passed Isolation and Mapping Test"

The log file is updated for each test conducted on a specific day. For instance, if boards 1-70 are tested on the first day, the log file will be titled "CE BOARD TEST 11_27_24 #1-70" with 70 entries. If more boards are tested on subsequent days, new log files will be created for each respective day. Once all tests are completed, the Arduino COM ports can be safely disconnected.

5. Acknowledgments

All development of the CE-Adapter Board test bench is made possible by the staff at the Physical Sciences Laboratory. Their continued support and guidance allowed this work to be conducted.

6. Appendix

1. Click here to view a list of all of the necessary materials and a construction procedure of the test stand. This describes the fabrication process and how each component is prepared.
2. A link to the GitHub repository with all of the related software can be found here. In this repository are the scripts for both the test GUI and the Arduino code.(Can replace with a non-personal git account)

References

[1] SparkFun Electronics. Bob-09056 breakout board. Available online: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/1918/BOB-09056_Web.pdf. (cit. on p. 3.)

[2] European Organization for Nuclear Research (CERN). Document no. 8760427. Available in EDMS. (cit. on pp. 5 and 6.)

[3] Texas Instruments. Cd74hc4051 multiplexer datasheet, 2024. Available online: https://www.ti.com/lit/ds/symlink/cd74hc4051.pdf. (cit. on p. 4.)