



# VICOR

Probe Interface Hardware Checker

Team Members: Boston Le (ELE), Giuliano Biondi (CPE)

Technical Directors: Al Binder, Nathan Shake, Daniel Hartnett



## PROJECT MOTIVATION

The purpose of a hardware checker is that it immediately tells the test engineer that either a passive component value is incorrect, a component doesn't have a good connection, or a relay is backwards or non-functioning. Having a working hardware checker available gives the ability to quickly debug the hardware and find out the problem. Without one it'd be unspecified if the hardware is dejected, which wouldn't inspire confidence in the board. This ensures that before VICOR sends out the board overseas, the hardware sent is precise and done properly, guaranteeing customer satisfaction. Having the hardware checker at our disposal saves money on company travel, where traditionally it would require an engineer to fly out to the CM to solve the issues. This would allow for the company to save time that would've been lost from having to mail hardware that could potentially run into shipping delay, and money that it would cost to ship out.

## KEY ACCOMPLISHMENTS

**P012/P014 Probe Card:** There were many technical responsibilities our team had for VICOR's hardware checkers. To begin, for the first couple weeks, our responsibilities included reviewing code for the P012, and the P014. We received code along with the schematics that matched between the two. Our task was to review the code, see what on the schematics the code was testing, and highlight it. This ensured that we could read code and actually see what it was doing and how it was able to test a specific path. We then were given a P012 bump and trim schematic, along with the code only for the trim file. Using what we had, we had to relate the two bump and trim schematics and build a P012 bump code that tests everything in the bump schematic.

**P052 Probe Card:** The P052 hardware checker was completed by team member Giuliano Biondi. This newly updated schematic of the P052 probe card involved a new timing page, along with a brand-new main page, since the newer P052 differed in pins. In this hardware checker code, it consisted of code written for each APU, every path that could be tested that had a resource, resistor tests, and capacitor tests. The second page of the schematic involved the timing page. To test this page, I had to switch relays and follow the paths from the main page onto the timing page.

**Eagle Test Program:** After putting time into understanding the eagle test program, our team is now able to understand how to compile, test, and work on the code inside the eagle test program. This is beneficial because in eagle, there are many different buttons that assist you in writing and editing code. Programs such as raide and other tools proved to be helpful in debugging code. You are able to compile, run the code, and actually see where you get errors, and go to that line and step through each line to see where there is an error in the code, which can save a lot of time for projects.

**P069 Probe Card:** The P069 hardchecker was completed by the team member Boston Le. The P069 was similar to the first probe card received; the P014. However, there were minor differences that made a tremendous impact with pins and apu's being in different locations or having mat's that will provide different readings. Like most other hardware checkers this had a schematic that had to be contact traced following different paths to see what was occurring. Simultaneously these paths would allow the designer to construct code following the paths and in this scenario from left to right. The code is created to verify the pins by forcing current/voltage to either measure voltage/current. If the range falls within the proper limits, then the code works correctly. With relays being closed and opened to the APU's you can expect 0 uA if the relay does not open to the APU, however if it opens to the APU without any other resources such as resistors you can expect the certain amount of voltage/current forced to make it to the pin.

**P040 Probe Card:** The P040 hardware checker was completed by the team member Boston Le. In this spring semester a new assignment to create a hardware checker was tasked. This hardchecker is the P040's and it was an all-new checker with new schematics. The main page, timing, and driver page were all fresh and it would be my first time viewing it. In the beginning we verify a path has been tested by highlighting contact traces to see which pins went to certain APU's or resources.

**100 Pin Needle Checker:** The 100 Pin Needle Checker was completed by the team member Giuliano Biondi. After the completion of the P052 Hardware Checker, the next task was to create a program that tests if each pin on a needle work. To do this, a program was created by testing the resistivity of the pin being tested and its surrounding pins while there was an open connection, and again while there was a short connection. After that was created, two new sets of tests needed to be created, testing the capacitance of the pin being tested and its neighboring pins again, while having an open connection and a short connection. After all these tests, we can conclude if the pins are correctly working or not.

## ANTICIPATED BEST OUTCOME

The best anticipated outcome of the project by April 15th, 2022 would be that the designers create as many hardware checkers as possible, and with that, gain a good understanding of how the code works. The hardware checkers will be able to test the probe cards to confirm the hardware's functionality. These hardware checkers will be fully functional for every probe card created under the same specifications as the code written.

## PROJECT OUTCOME

The Anticipated Best Outcome was achieved.

## FIGURES

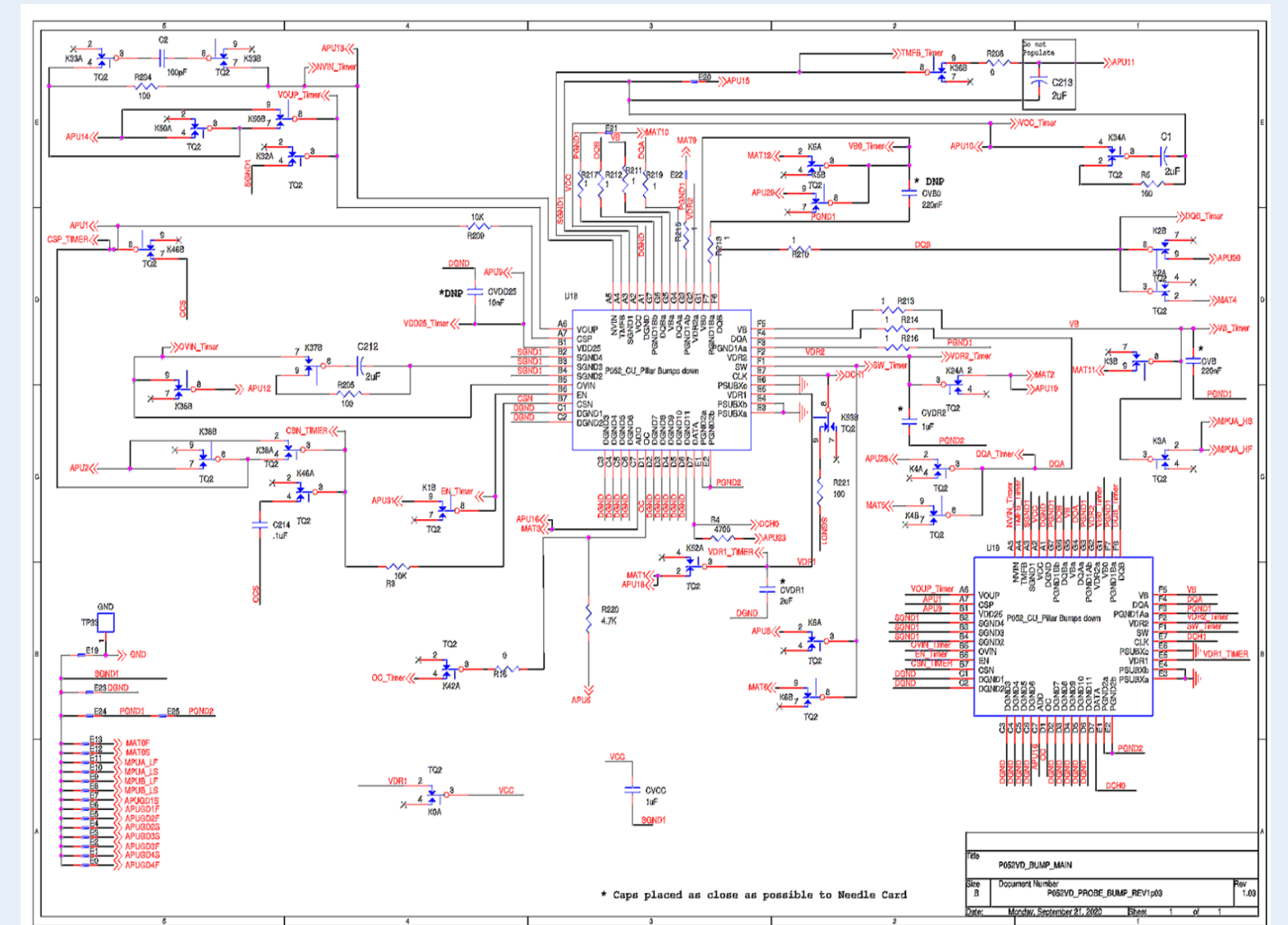


Figure 1: P052 Main Page

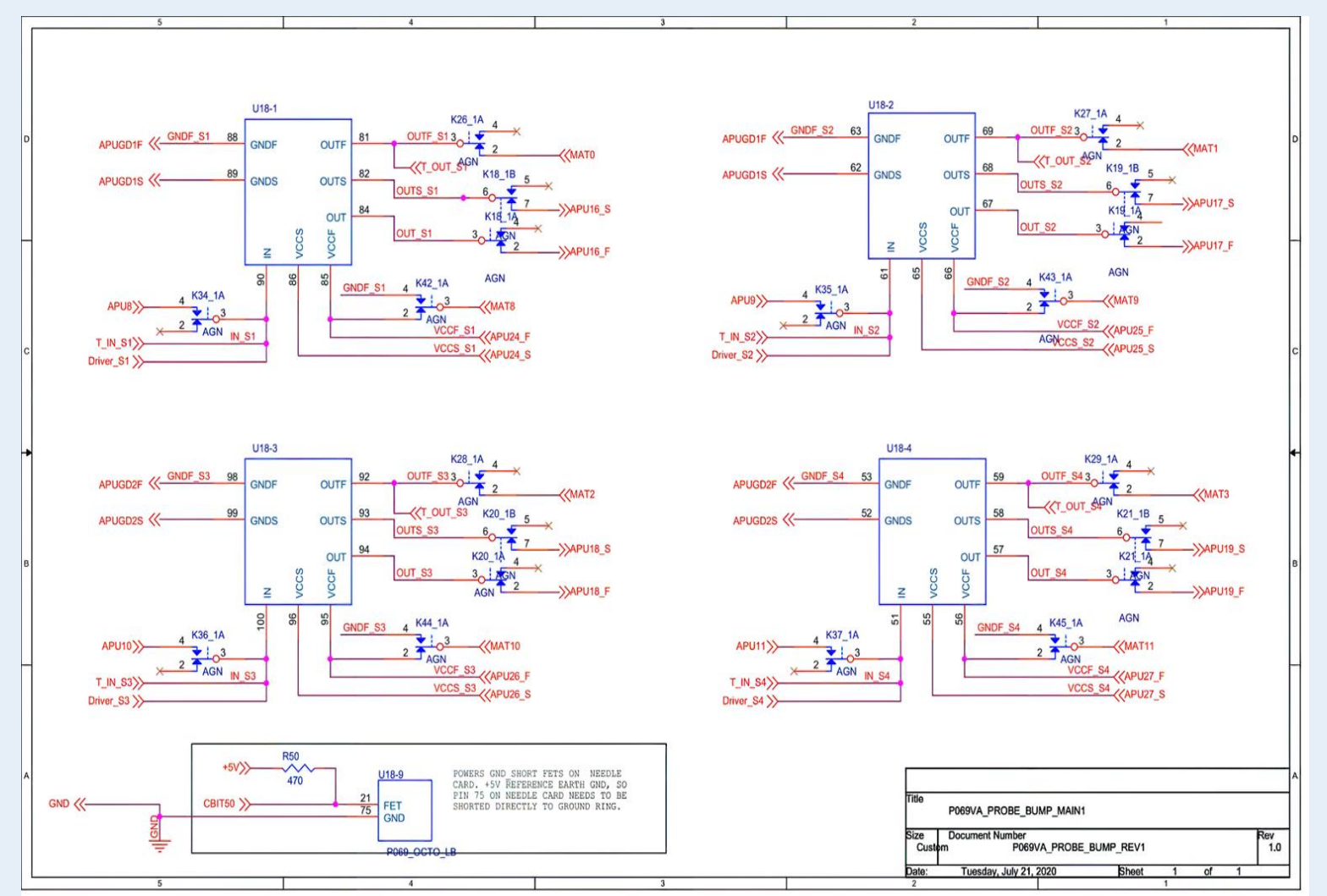


Figure 2: P069 Main Page

```
// PIN 71
apuset( PIN_71 , APU_FV, 1.0, APU_I0V, APU_I0A, PIN_TO_VI ); //turn PIN71 ON
lwait( 1000 );

//Pin 71 to 70
apuset( PIN_70 , APU_FV, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait( 1000 );
pin71_70_o=apumi( PIN_71, 10, 10 );
pin71_70_c*=le3;

apuset( PIN_70 , APU_VIOFF, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait(1000);

//Pin 71 to 71
pin71_71_o=apumi( PIN_71, 10, 10 );
pin71_71_c*=le3;

//Pin 71 to 72
apuset( PIN_72 , APU_FV, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait( 1000 );
pin71_72_o=apumi( PIN_71, 10, 10 );
pin71_72_c*=e3;

apuset( PIN_72 , APU_VIOFF, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait(1000);

//Pin 71 to 73
apuset( PIN_73 , APU_FV, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait( 1000 );
pin71_73_o=apumi( PIN_71, 10, 10 );
pin71_73_c*=le3;

apuset( PIN_73 , APU_VIOFF, 0.0, APU_I0V, APU_I0A, PIN_TO_VI );
lwait(1000);
```

Figure 3: Needle Checker Resistance Test for pin 71