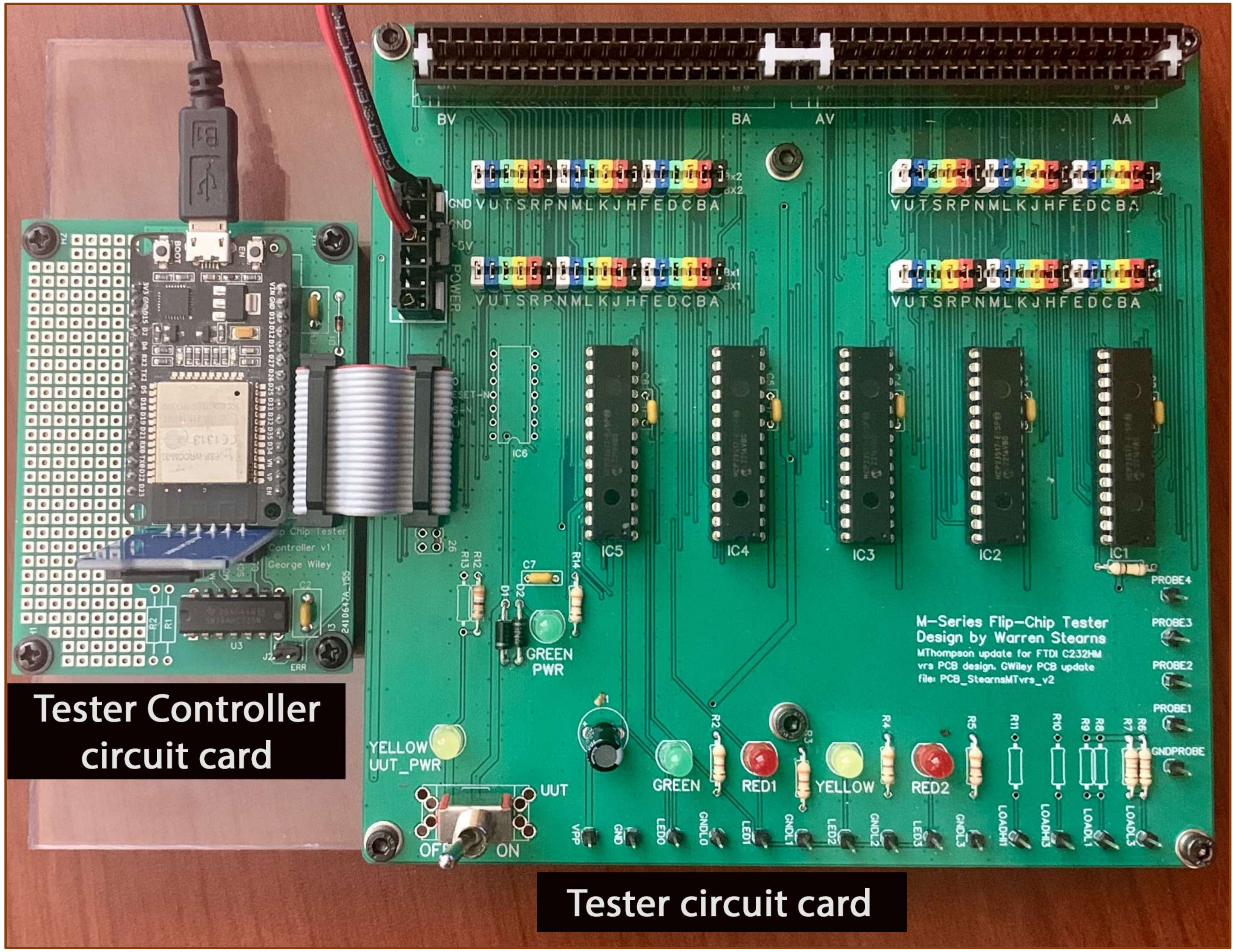
Flip Chip Tester System Build Information

- The following pages describe how to build and set-up the Flip Chip Tester System.

- The Flip Chip Tester System is comprised of a Tester Controller circuit card and a Tester circuit card. Both circuit cards are connected via a 20-pin flat cable. The Tester Controller has a microSD adapter so test programs can be retrieved from a microSD card.



Background

The Flip-Chip Tester System is a tool to perform verification testing and debugging of most M-series and some G-series Flip Chip modules. The system is comprised of a Tester board and a Tester Controller board connected together via a 20-pin flat cable. A photo of the entire system is on the previous page. The Tester Controller has a microSD adapter so test programs can be retrieved locally from a microSD card.

An accessory called the "IC Test Jig" plugs into the UUT socket of the tester which enables functional testing of digital ICs.

This Tester board is based on a tester designed by Michael Thompson with PCB by Vince Slyngstad. Refer to the StearnsMTvrs files in this folder: https://svn.so-much-stuff.com/svn/trunk/Eagle/projects/Stearns%20Tester/

The original Flip-Chip Tester was designed by Warren Stearns. There was a version of this original tester that connected to a PC through a parallel printer port, and I believe another version controlled by an MC6809 microprocessor. A more detailed and complete history of the tester evolution is here: https://so-much-stuff.com/pdp8/repair/fc-tester.php

It is important to mention that a significant library of test vector files has already been created by the original developers of the tester to test "M" and "G" series Flip Chip modules. These are mainly the single and double width modules used in the PDP-8/L and PDP-8/I. This library of existing tests is a tremendous benefit to the PDP-8 enthusiast community. https://svn.so-much-stuff.com/svn/trunk/pdp8/warren/16bit/tests/

There are also some PDP-8/L module test files in the repository: "Flip_Chip_test_procedures_and_vectors" https://github.com/G-Wiley/Flip_Chip_test_procedures_and_vectors

The Tester circuit card is comprised of a set of five MCP23S17 16-bit GPIO Expander chips and a double-wide Flip Chip connector. The configurability of the GPIO Expander chips makes it possible for each Flip Chip module pin, the Unit Under Test (UUT) to be configured individually as an output or input.

Each MCP23S17 16-bit GPIO Expander has an SPI port which is connected to the tester controller CPU. The SPI bus MISO and MOSI signals of all 5 GPIO expander chips are tied to common MISO and MOSI signals and each GPIO expander has its own chip select signal and device address.

The tester controller consists of an ESP32 module and a microSD card adapter module. The microSD card adapter is on the same SPI bus as the GPIO Expanders on the Tester circuit card, but the microSD card adapter has a unique chip select signal so the CPU can address only one device at a time. A prototype version of the Tester Controller is shown below.

The MISO signal from the microSD adapter is connected through an SN74AHC125 tri-state buffer enabled by the microSD chip select so that the global SPI MISO signal is driven only when the microSD is accessed. This makes it possible for the microSD card adapter and MCP23S17 16-bit GPIO Expanders to share the same SPI.

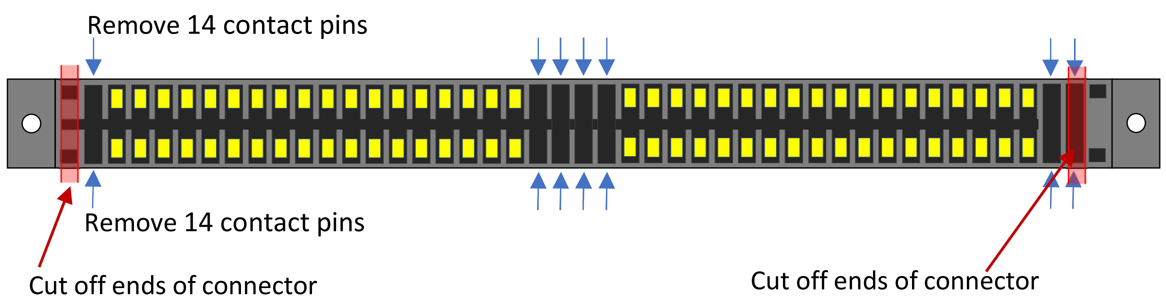
The original Stearns tester software has been ported to run on the ESP32 and compiled in the Arduino development environment. Modifications have been made to add the ability to access the microSD adapter and to access the GPIO Expanders one at a time over the common MISO and MOSI signals. The printer port I/O handlers have been removed and replaced with handlers to communicate with the MCP23S17 GPIO Expanders over the SPI.

Instructions to modify the edge connector #1

The connector that holds the UUT is purchased as a single 2 x 43 pin connector. It needs to be modified so it can accept either a single or dual Flip Chip module and have that module be properly aligned with the connector contact pins. Also, the connector, as purchased, is too wide to be installed in the Tester PCB and would interfere with two of the mounting screws on the Tester circuit card.

The following instructions describe how to remove unused contact pins, cut the connector to the proper width, and insert shims into the unused contact pin locations so the UUT will be properly aligned when it is inserted into the socket.

2 x 43 pin connector modifications



Note: It is possible to avoid cutting the ends of the connector. However, the extra width will interfere with two of the standoff mounting holes on the Tester PCB.

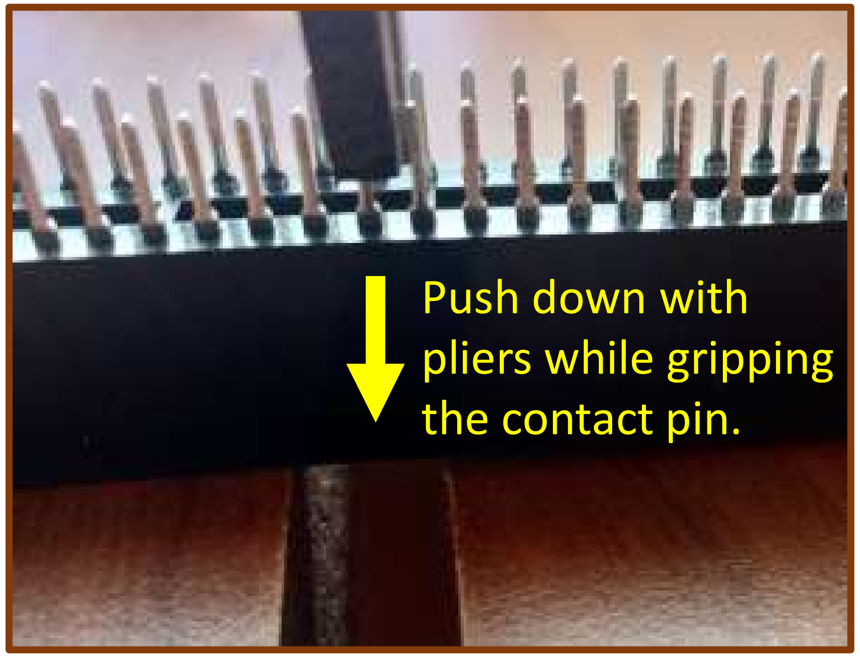
Instructions to modify the edge connector #2

Remove the extra connector contact pins

Remove the extra connector contact pins before cutting. The contact pins to be removed are indicated by the blue arrows on the previous page. Removal of pins is necessary so the saw blade doesn't bind on the metal contact pins, and so the shims will fit in the empty contact positions.

An easy way to remove the pins is to place the connector slot-side down, pin-side up, on a flat surface that has an opening that allows the pins to be easily pushed through. The photo below shows an example of needle nose pliers gripping a contact pin which is pressed downward.

The example below shows two sides of a desk supporting the connector. A simple alternative would be to drill a large hole in a block of wood to support the connector while pushing the contact pin downward but to allow the removed pin to slip through the opening.

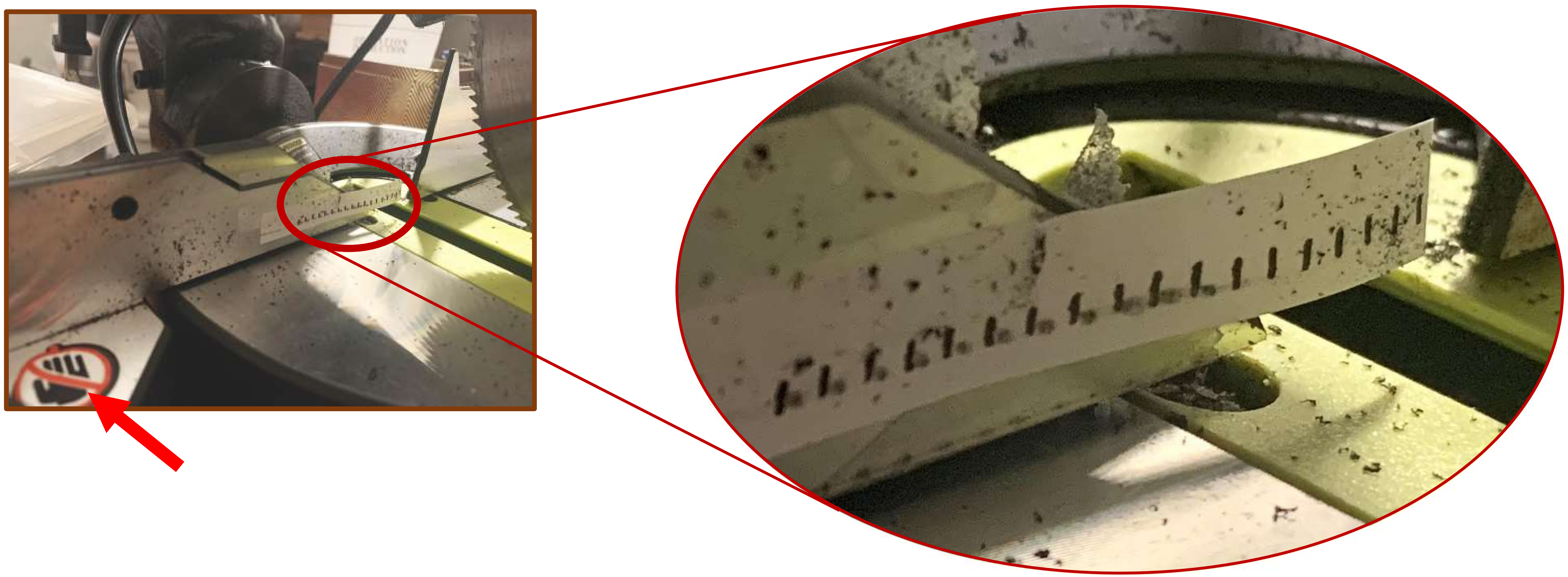


Instructions to modify the edge connector #3

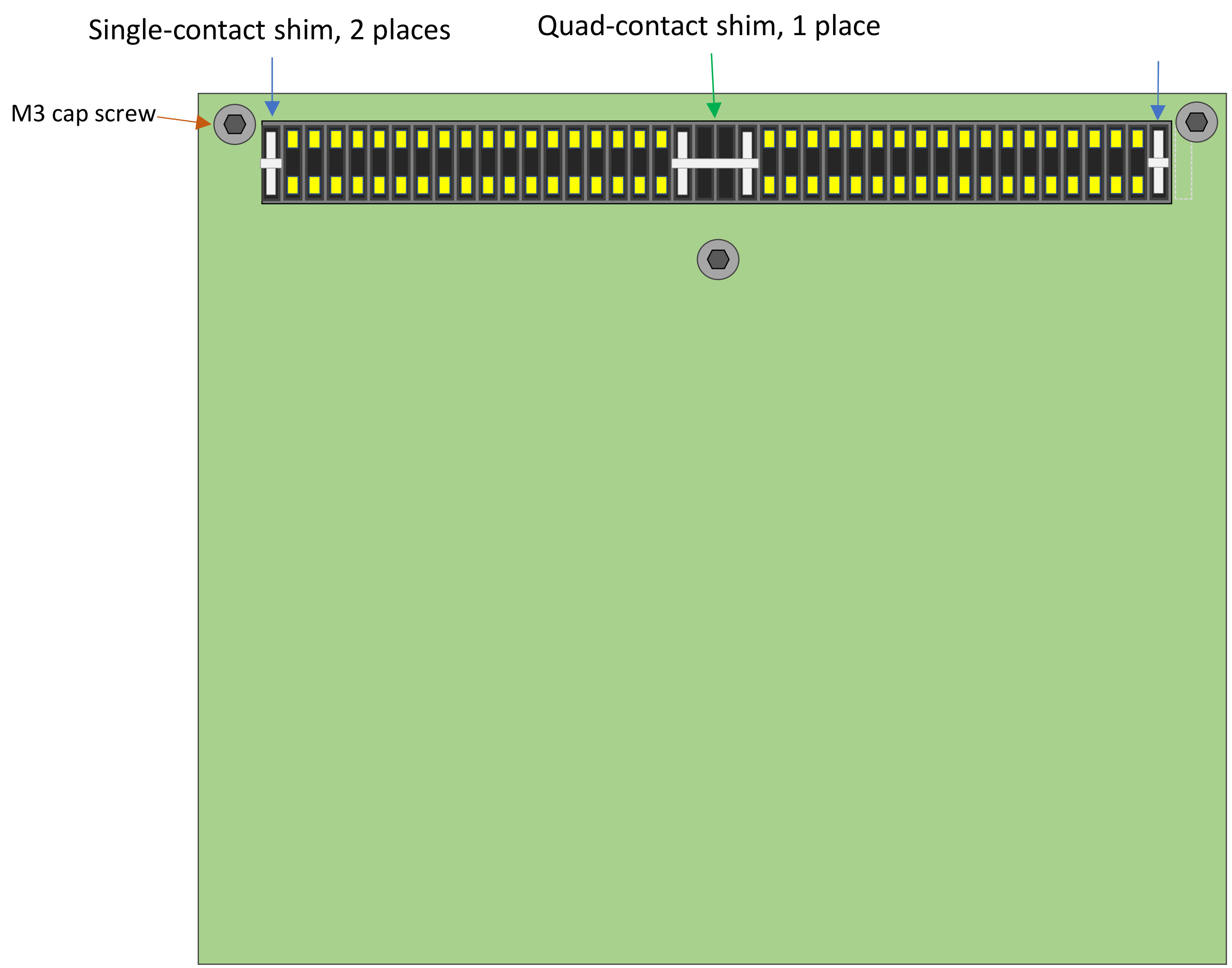
Cutting the edge connectors

I used a small 7 ¼ inch power miter saw with a 140-tooth blade. Make the cuts at the locations indicated by the red bands on the first page.

The photo below shows a guide taped to the saw that is used to help align the connector during the cutting process. Notice the icon in the lower left warning to not put your hand there, so a clamp is necessary to secure the connector while cutting. Small marks from the connector pins are visible on the paper guide. I didn't have the marks perfectly aligned when I attached the paper guide and adjusted the position of the connector early in the cutting process. The connector pins made the small dots to the right of the larger marks. The connector was cut lying flat with the open slot toward the front and the pins pressed against the guide. <u>When cutting the ends, take care to leave the full thickness of the connector pin separation wall at the outside of the connector so the single-width shims on the outside spaces will fit tightly.</u>



Connector Shims #1

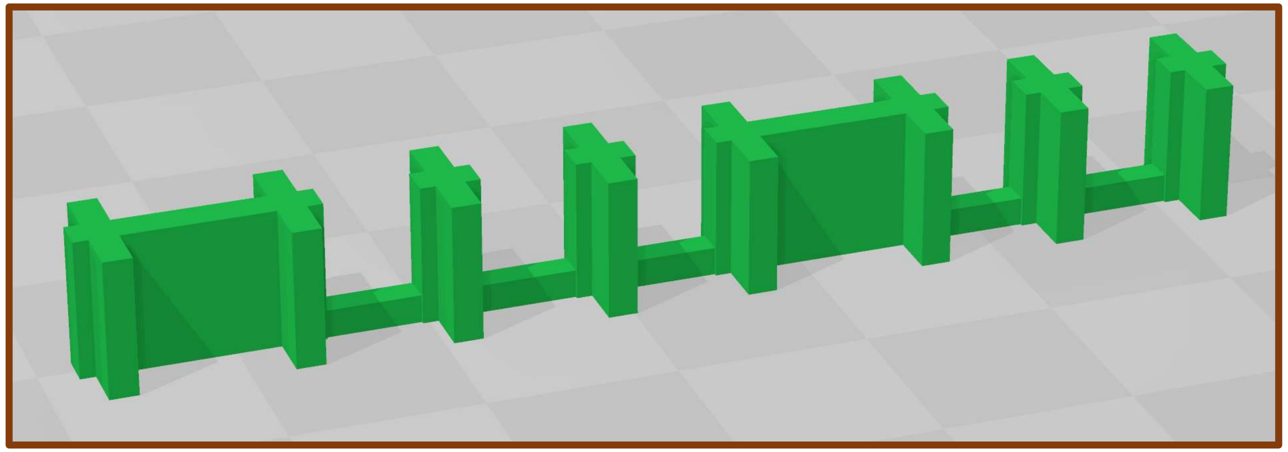


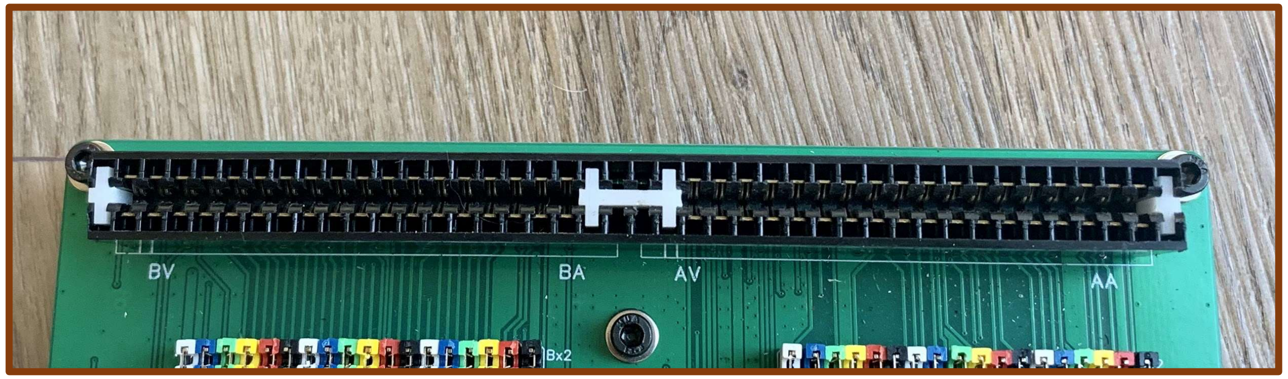
Connector Shims #2

The connector shims are 3D printed in a small rack that contains twice the number of shims required for a Flip Chip Tester. It's nice to have extras in case these small pieces are lost or damaged. There are 2 quad-contact shims and 4 single-contact shims. An entire set of shims are printed together because it is easier to handle a single 3D print job and it reduces the printing cost. The shims in the image below were printed at JLCPCB in LEDO 6060 Resin. The file to be printed is "Flip_Chip_Tester_shims_v00.stl", an STL format file in millimeter units. (At JLCPCB, simply upload the file, choose "LEDO 6060 Resin", and order the part.)

A 1.5 mm disposable rib connects the entire set of shims. They can be cut apart easily using a set of diagonal cutters that is flat on one side. When separating a single shim, place the flat side of the diagonal cutter next to the part you are removing from the rib.

The two single-contact shims might fit rather loosely in the connector if the outside spacer walls were cut too thin. If so, they should be glued so they remain in place. A tiny sub-drop of epoxy should hold the shim in place. Be very careful not to get any glue on the connector contacts.





Ordering the PCBs and Shims

PCBs:
The Tester and Tester Controller were designed to be fabricated with the most basic PCB process. Both boards can be fabricated from standard 1.6 mm thick FR4 material or a different thickness can be used without any adverse effects. The Lead Free HASL-RoHS process is recommended. There are two sets of Gerber files:

- “Gerber_PCB_StearnsMTvrs_v2.zip” for the Tester board.
- “Gerber_PCB_FlipChip Tester Controller ESP32 v1.zip” for the Tester Controller board.

For example: At JLCPCB, simply upload each zip file, choose “LeadFree HASL-RoHS” process, and proceed to checkout. You may want to order the boards separately to avoid confusion.

Note:

- Tester board “Gerber_PCB_StearnsMTvrs_v2.zip” requires some modifications but is available now.
  The rework instructions are in: “Rework Instructions for the Module Tester PCB v2.pdf”
- Tester board “Gerber_PCB_StearnsMTvrs_v3.zip” has the bugs fixed but has not yet been released (as of 8-Oct-2022).

Shims:
The shims were designed to be fabricated using a low-cost 3D-printing process. As mentioned previously, there is a single STL file for printing two sets of shims for the tester:

- “Flip_Chip_Tester_shims_v00.stl”

For example: At JLCPCB, simply navigate to the 3D printing area, upload the STL file, choose “LEDO 6060 Resin”, and order the part.

Mounting Components on the PCBs

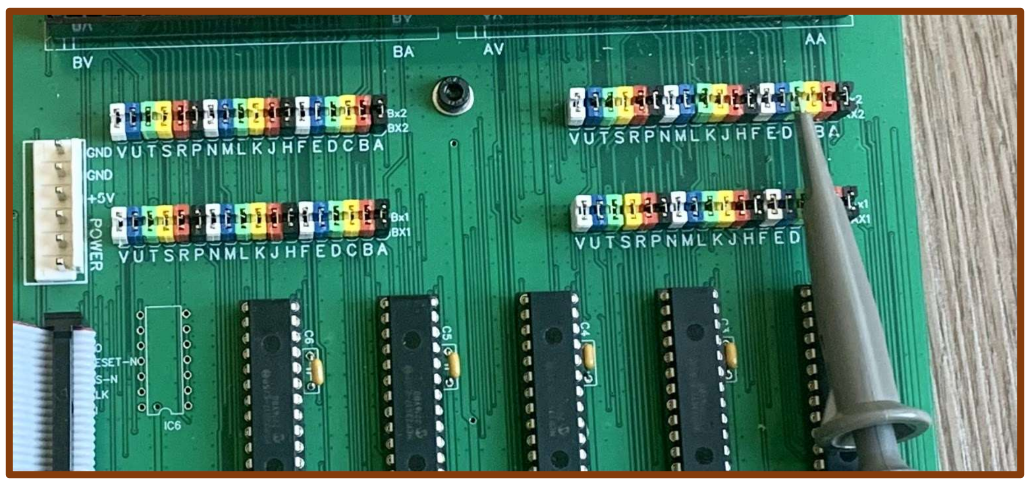
Tester PCB

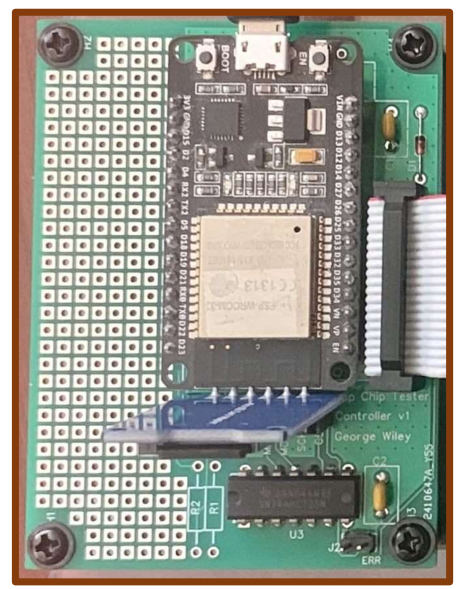
There's not much unusual here. The components are installed as indicated by the BOM spreadsheet: "BOM_PCB_StearnsMTvrs_v2_2022-0927.xls"
The Tester BOM is the "Tester Circuit Card" worksheet.
Some components are not installed as noted in the BOM. R6 through R11 & R13 are DNI.
The PCB is designed with pads for a 26-pin header at location PC1, but a 20-pin header should be installed in that location instead. The <u>20-pin header should be installed at pins 1 through 20</u> with nothing installed in pins 21 through 26.

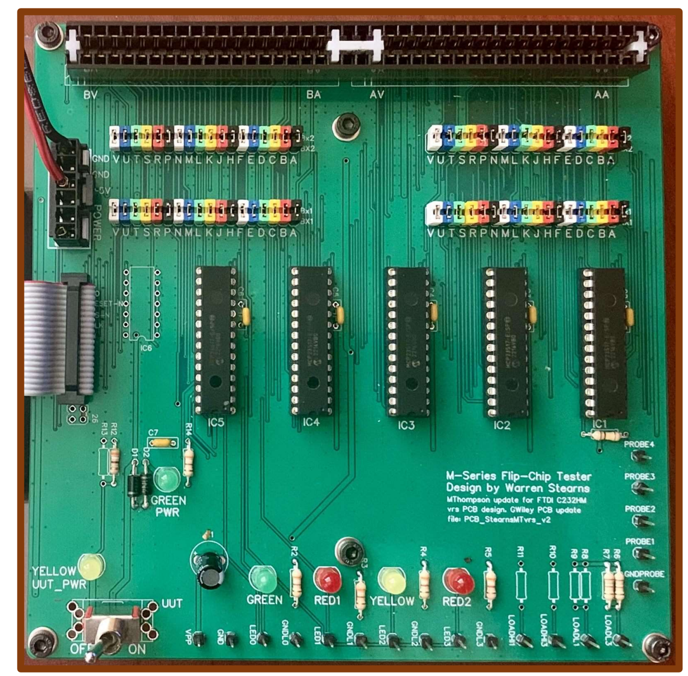
Install 72 jumper plugs onto the four 36-pin headers. It's convenient to use multiple-colored jumpers so it's easier to see which pin label on the board silkscreen aligns with the metallic contact at the top of each jumper plug. This is useful when probing signals at the top of the jumper plugs while running scope loop tests. Board edge connector signals can more easily be probed at the tops of the jumper plugs compared to counting gold fingers on the front and back of the UUT. Also, a scope probe with the witch's hat attachment can be clipped to the tops of the jumper plugs.

Tester Controller PCB

The components are installed as indicated on the BOM spreadsheet: "BOM_PCB_StearnsMTvrs_v2_2022-0927.xls"
The Tester Controller is the "Tester Controller ESP32" worksheet.
Some components are not installed as noted in the BOM. R1 & R2 are DNI.

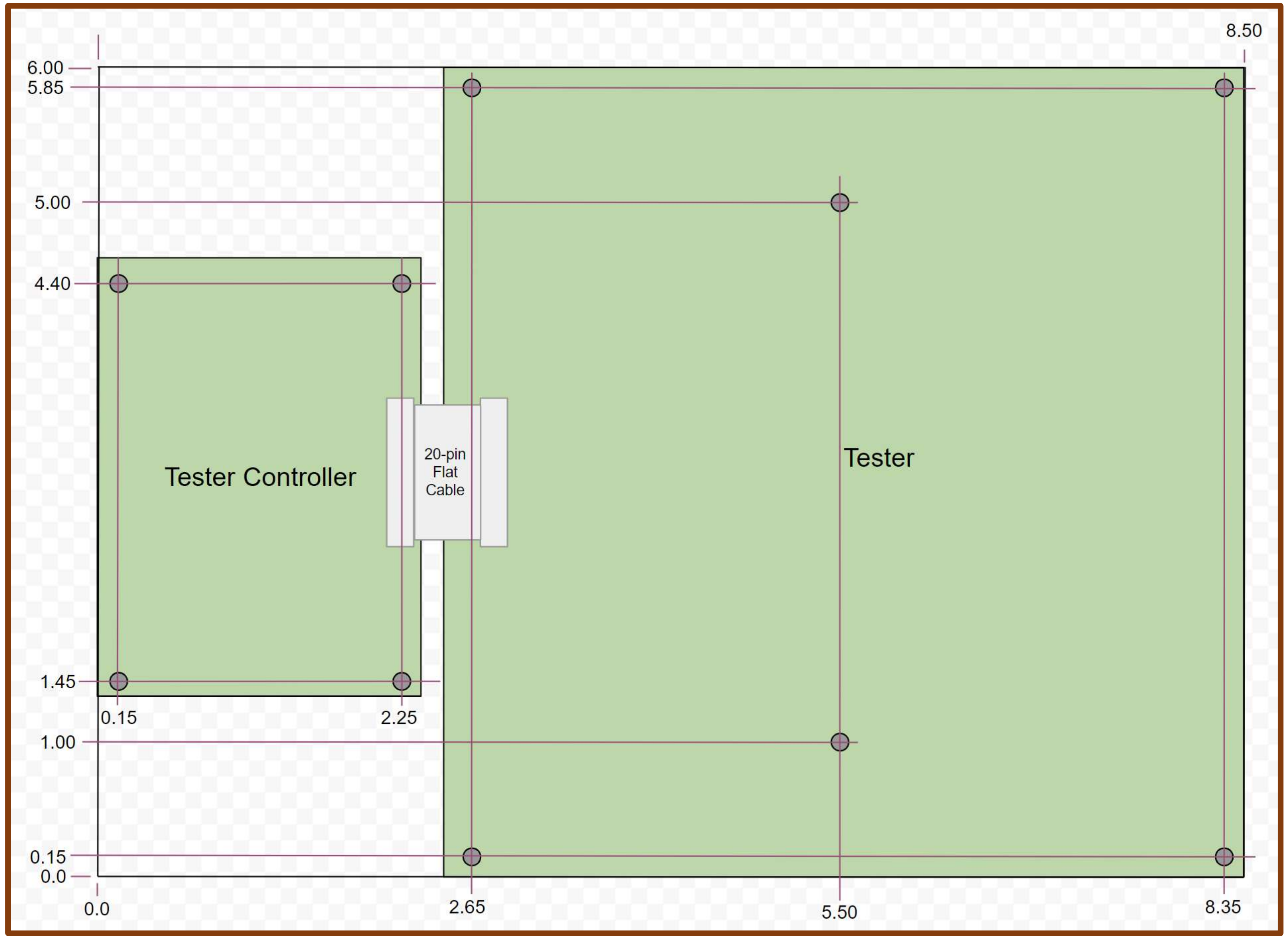






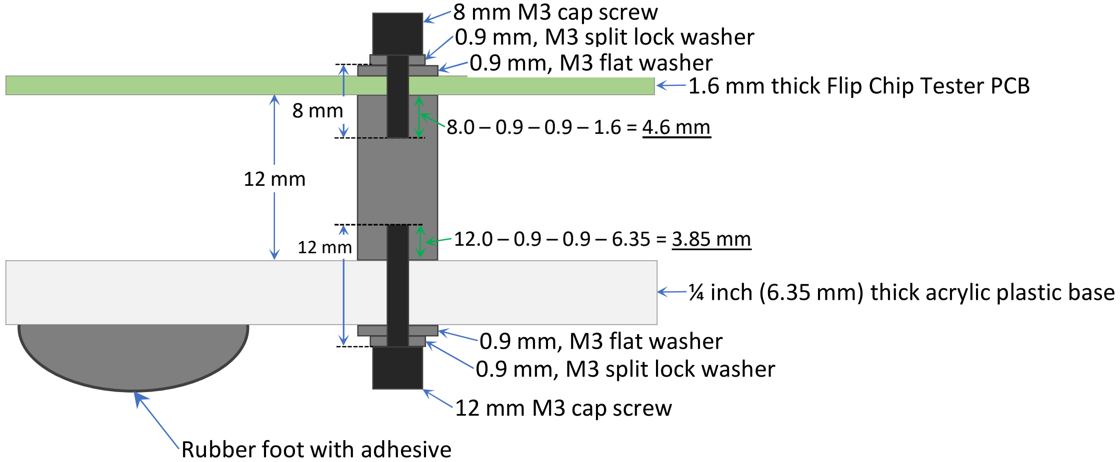
Tester and Tester Controller Mounting on ¼” Acrylic Base #1

The drawing to the right shows the locations of the mounting holes for the Tester Controller and Tester circuit boards. The hole size should be about 1/8” (3.2 mm to 3.4 mm) for proper clearance for an M3 screw. The screw locations shown will result in perfect alignment of the 20-pin headers on both boards.



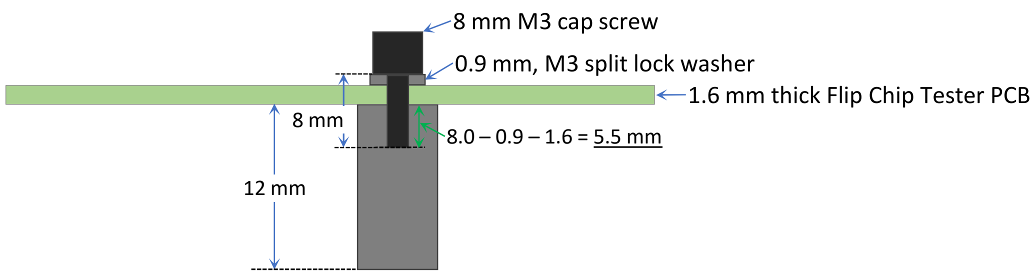
Tester and Tester Controller Mounting on ¼" Acrylic Base #2

Use M3 hardware to mount the Tester and Tester Controller to an acrylic base. M3 cap screws have a small diameter head so they fit well in the mounting holes at the corners of the Tester board. A screw with a large diameter head can overlap the edges of the board in the corner hole locations. Tall adhesive rubber feet prevent the screws on the bottom of the acrylic sheet from scratching the desktop. Take care to ensure the 20-pin headers on the Tester Controller and Tester board are perfectly aligned so the flat cable can connect straight across between the two boards without any bending of folding. This will also allow the flat cable to be as short as possible.



Mounting without Acrylic Base

Although the thick acrylic base provides the best support for both insertion and removal of boards, it is also possible to use the Tester with the nylon standoffs as supporting feet. Mounting to an acrylic base provides better support when removing Flip Chip modules. Also, mounting to a base provides better support for the two boards which are connected via a 20-pin flat cable.



Setting up the Tester System for use

Final Assembly

Mount the two boards to an acrylic base using 12 mm nylon standoffs and M3 hardware as previously described. Connect a 20-pin flat cable between the boards. Connect a micro-USB cable to the connector on the ESP32 module and connect the other end to the computer where the Arduino development environment is installed.

Copy the test files with ".TST" extension to a microSD card.

Download the test files from:

- https://svn.so-much-stuff.com/svn/trunk/pdp8/warren/16bit/tests/
- https://github.com/G-Wiley/Flip_Chip_test_procedures_and_vectors

Save the test files to the microSD card. You can save them to the root level because no other files need to be saved on the microSD card.

The Arduino Development Environment for the Tester

Setting up the Arduino environment

The following are helpful tutorials to get your Arduino dev tools set up properly for the ESP32:

- https://randomnerdtutorials.com/getting-started-with-esp32/
- https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/
- ESP32: Guide for MicroSD Card Module using Arduino IDE: https://RandomNerdTutorials.com/esp32-microsd-card-arduino/
- The Arduino SPI library: https://docs.arduino.cc/learn/communication/spi

I highly recommend getting the DOIT ESP32 DEVKIT V1 module so it matches the tutorial. This is the module specified in the Tester Controller parts list.

Download the Tester Controller Arduino code from github to the host PC. The ".ino" files need to be in a folder of the same name, but the folder name is without the ".ino" file extension. This is a requirement for the Arduino tools to work properly. For example: copy the file ESP32_blink_v00.ino to a folder named ESP32_blink_v00

The following two files are useful to test that the Arduino environment is set up properly and to test the hardware:

- ESP32_blink_v00.ino – a very simple program that blinks the LED on the ESP32 module. This will also work with the ESP32 not inserted in the Tester Controller board; just the ESP32 board sitting on a table and connected to the USB cable will work. This program is good to verify that the Arduino development environment is set up properly. It does not require the SD library or SPI.
- ESP32_blinkSPI_v03.ino – blinks the LEDs on the Tester board out of phase with the ESP32 module blue LED blinking. It does require SPI to communicate with the MCP23S17 chips but does not use the microSD file system.

The Flip Chip Tester software is in one file:

- DOIT flipchip_tester_ESP32_v01.ino – the Flip Chip Tester operating software. This requires use of SPI and the microSD file system. At the time you are reading this, the current version might be later than v02, so modify the instructions accordingly.

High-Level Instructions

The following assumes that the reader has some familiarity with the Arduino development environment. If needed, refer to the tutorial links mentioned previously.

1. Set up the Arduino environment for the DOIT ESP32 DEVKIT V1 module: https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/
2. It might be also necessary to install the SD library: https://github.com/espressif/arduino-esp32/tree/master/libraries/SD
3. Download the code from github: flipchip_tester_esp32_v01.ino and save it on your computer in a folder with the same name, flipchip_tester_esp32_v01
4. In Arduino, open flipchip_tester_esp32_v01.ino.
   Set the serial port: Tools → Port and select the proper COM port.
   Set the board type: Tools → Board and select DOIT ESP32 DEVKIT V1
   Compile and upload to the ESP32: Sketch → Upload
5. If there are problems, the following can be used to isolate faults:
   a. A simple program to blink the LEDs on the Tester circuit card and the on-board ESP32 blue LED out-of-phase with each other is in github: ESP32_blinkSPI_v03.ino.
   b. The most simple blinker… If there are more basic problems then try this simple program to blink just the on-board LED on the ESP32 module. It's in github: ESP32_blink_v00.ino This simple blinker needs only the DOIT ESP32 DEVKIT V1 module connected to the host computer via USB. It is powered through USB. No external power is required. The ESP32 module does not need to be attached to the Tester Controller circuit card.
   c. If there are problems with the microSD, either with the software build or functionality, then try to run the simple Rui Santos example code at the bottom of this page: https://randomnerdtutorials.com/esp32-microsd-card-arduino/
   The 20-pin flat cable to the Tester board can be removed if you like while running this example code so that only the microSD adapter is connected to the SPI signals. The serial baud rate for this example code is 115200. Tools → Serial Monitor is helpful.
6. The Tester Controller is configured for console I/O through the USB serial port running at 921,600 baud. A simple terminal emulator such as Putty is useful. If a different baud rate is preferred then search for this line of code: `Serial.begin(921600);` and change the baud rate parameter.
7. It's helpful to power the tester with a lab power supply adjusted to output 5.0 volts so the current consumption can be observed while running tests.

Using the Tester

After loading flipchip_tester_ESP32_v01.ino into the ESP32 module using the Arduino dev tools, run a terminal emulator program such as Putty. The Flip Chip Tester software initializes the serial port to run at 921,600, so set the baud rate in Putty to 921,600 bps. If necessary, the baud rate can be modified by changing the value in the Serial.begin(921600) function call. The Tester splash screen will probably display immediately after starting Putty. If not, then press the “EN” button on the ESP32 module (next to the USB connector), not the “BOOT” button. The “EN” button will reset the ESP32.

Most of the original Stearns Tester commands are still supported. Some functions are not available if they no longer make sense, such as: the “Exit” command from the main menu and diagnostic test commands LPT functions (that test the PC parallel port) and SPI functions.

Each Flip Chip module has one or more associated test files with file extension “TST”. Save the TST files on a microSD card and then plug that microSD into the socket on the Tester Controller. It’s probably easier to save the TST files to the root level of the microSD card. The original Flip Chip TST files are available here:
https://svn.so-much-stuff.com/svn/trunk/pdp8/warren/16bit/tests/
Some additional scope loop tests for PDP-8/L modules are here: https://github.com/G-Wiley/Flip_Chip_test_procedures_and_vectors
Test Data Sheets that describe how to run each test are available in this same github repository. These Test Data Sheets are needed to properly run the scope loop tests.

If it’s necessary to add TST files to the microSD card then the Tester Controller must be reset after the microSD is re-inserted into the socket on the tester.

Simple tester use instructions:

1. From the main menu, “1” (read test file), then type in the test file name, such as M113.TST followed by the “Enter” key. Filenames are not case-sensitive.
2. With the UUT power switch on the Tester board in the “OFF” position (handle to the left), insert the module you want to test into the UUT socket. Single modules go in the right side and double modules use both sides. The A and B pin labels are indicated on the Tester board silkscreen next to the UUT socket.
3. Turn on the UUT power switch on the Tester board (handle to the right). If the switch remains off then the tester software will complain in the next step.
4. To run the test, “4” (run test).
5. Nine different testing options are now available. The most useful are “O” and “S”. “O” runs a test once and specific errors are displayed. “S” is a scope loop test mode where the test is looped at high speed and only a pass/fail status is displayed for each loop through the test vectors. Some Flip Chip tests can only be run as scope loop tests where the pass-fail criteria is determined by examining a waveform on the scope.
6. For scope tests & debugging:
   Error Pulse: When using the scope loop test mode (“S” command), the ERR testpoint on the Tester Controller board will produce a narrow positive pulse every cycle where an error occurs. The pulse will appear at the end of the specific vector cycle so look at the logic state of pins just prior to this pulse to observe the conditions that caused the failure in that clock cycle.
   Start of Test Pulse: The LED3 testpoint on the Tester PCB produces a positive pulse at the beginning of the cycle of tests. It is useful to use this signal as a scope trigger for scope loop tests. The time of this LED3 pulse can be adjusted using the Tester main menu command “2”, set test delay. The default delay is zero.