Flea-Scope Parts and Assembly and Test

# Parts

**U1** PIC32MK0512GPK064-I/MR has allowed substitutions (check availability and price for cheapest at [Microchip Direct](https://www.microchipdirect.com/product/PIC32MK0512GPK064-I/MR?productLoaded=true)):

* [PIC32MK0512GPK064-I/MR](https://www.microchipdirect.com/product/PIC32MK0512GPK064-I/MR?productLoaded=true) is first choice and usually cheapest at $7.20 in quantities of 1000’s
* [PIC32MK1024GPK064-I/MR](https://www.microchipdirect.com/product/PIC32MK1024GPK064-I/MR?productLoaded=true) is second choice at $7.40
* [PIC32MK0512MCM064-I/MR](https://www.microchipdirect.com/product/PIC32MK0512MCM064-I/MR?productLoaded=true) at $7.51
* [PIC32MK1024MCM064-I/MR](https://www.microchipdirect.com/product/PIC32MK1024MCM064-I/MR?productLoaded=true) at $7.70
* “-E/MR” suffix (extended temperature) adds $0.33 approximately and is also allowed
* (“/PT” suffix is for QFP instead of QFN and is **not** allowed!)

Do not substitute for Texas Instruments op amp **U3** OPA863SIDBVR.

Do not substitute for Wurth Elektronik LEDs **E1**, **E2**, and **E3**, even if other parts have identical forward voltage -- brightness has been calibrated to make all three LEDs equal; LEDs are non-linear, and it is actually possible to get a LED with a similar or even lower forward voltage that is less bright at the current I am driving it at (significantly below 20mA).

Resistors **R8**, **R9**, **R10**, and **R11** must be 1% tolerance, as in BoM.

Capacitors **C9** and **C10** must be 1% tolerance, as in BoM.

# Assembly

If highly compatible with your assembly process, a lead-free HASL board is our first choice because it will be slightly more convenient for customers to insert header pins, as there is slightly more restrictive clearance on our header pin holes. But if ENIG is more reliable in any way, we’d prefer reliability over customer convenience.

# Programming

The first step is to connect **both** the USB connector (for power) on the Flea-Scope board **and** the USB connector for the programmer (for programming) to the host computer (I use Windows 11 below). **Then** connect the programmer ICSP pins to the 6 board ICSP pins as shown below using a standard 0.1” (2.54 mm) header pin strip -- the holes in the flea-scope board are slightly “zigzag offset”, so they should hold the header pin strip snugly without need of bending or soldering.

Make sure pin #1 of the programmer is aligned with pin #1 (square pad) of the ICSP connector on the Flea-Scope board.

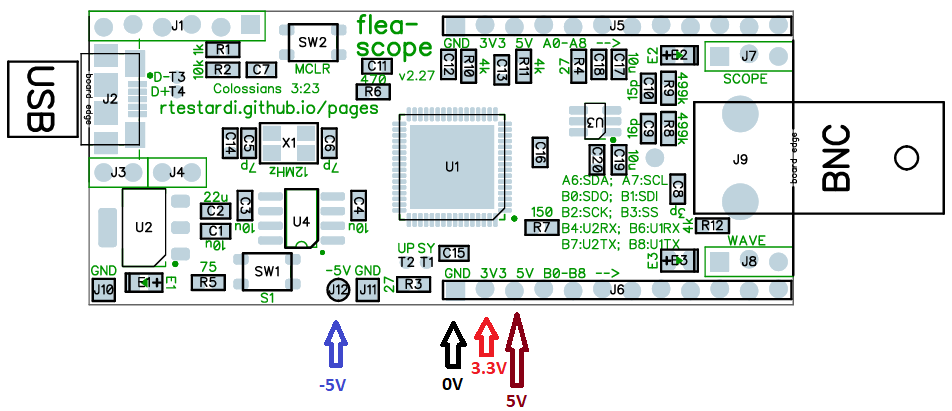
You can use any of these programmers:

* Snap (cheapest) -- <https://www.microchip.com/en-us/development-tool/PG164100>
* Pickit 3 -- available on ebay
* Pickit 4 -- <https://www.microchipdirect.com/dev-tools/PG164140?allDevTools=true>

I am using the “Snap” programmer in the pictures below, but the others are similar.



Once USB is connected to the Flea-Scope, you can also verify the Flea-Scope board has power by measuring these pins (5V and -5V are unregulated, so they are approximate -- anything within 0.25V is good enough):



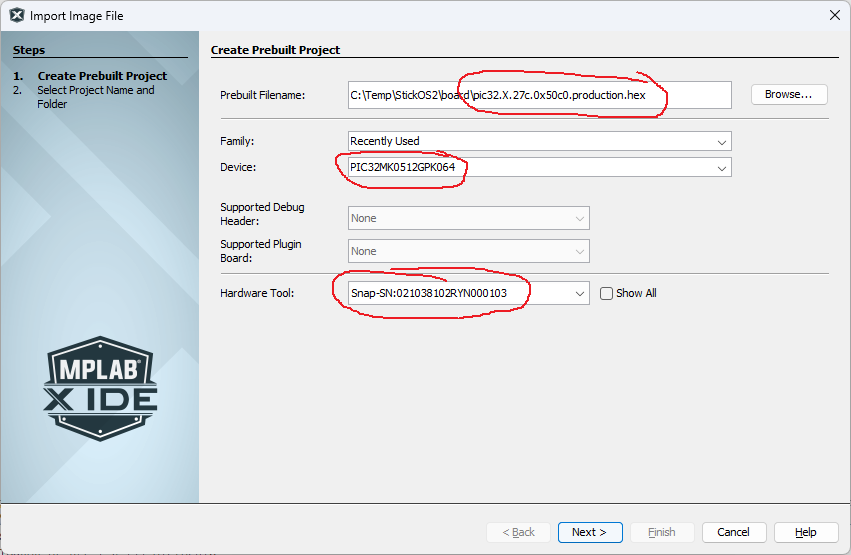
To program, I use MPLAB X v6.05 which is available here: [MPLAB® Ecosystem Downloads Archive | Microchip Technology](https://www.microchip.com/en-us/tools-resources/archives/mplab-ecosystem)

Search for “6.05” and download -- unfortunate3ly the newest version of MPLAB X (v6.15) has bugs and hangs often.

The next step is to start MPLAB X v6.05 and import the firmware hex (prebuilt) file:

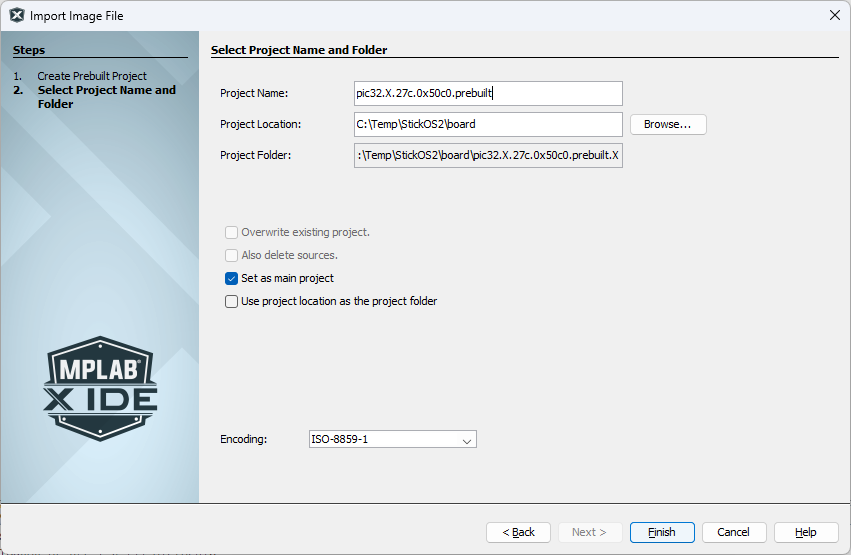


And then select your device and MCU and programmer as well:

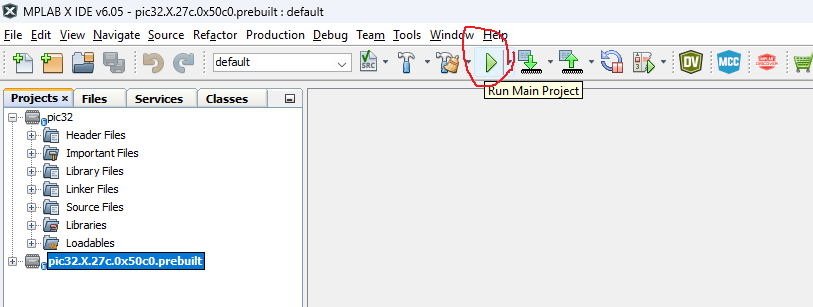


Note that if you are substituting a different MCU than the PIC32MK0512GPK064, select your MCU above. Likewise if you are using a different programmer.

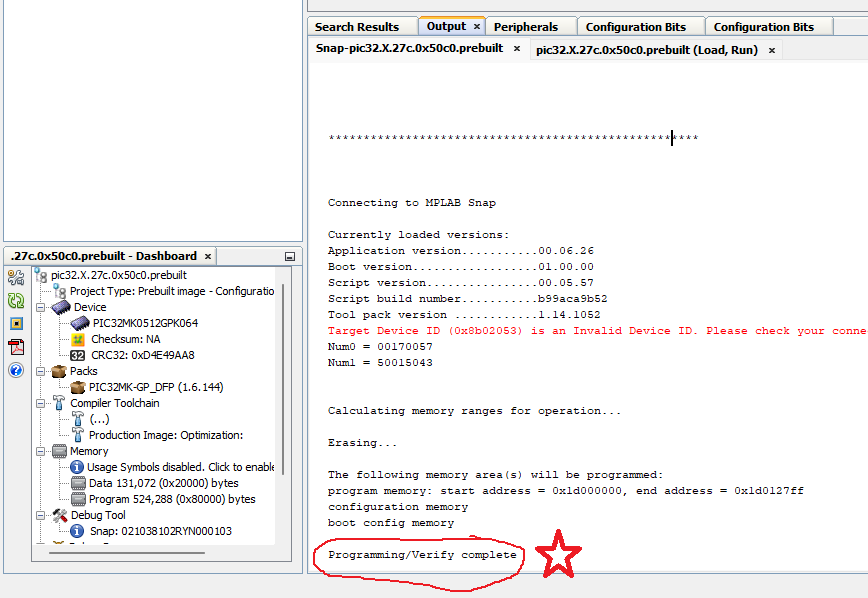
And then click “Finish” to save the project:



At this point you can use the “Run Main Project” button as shown below to program the MCU (you will only have one project -- I show two below):



Program the MCU and you should see something like this at the bottom of the screen (without the warning that I have the wrong device ID -- I program various MCUs from the same project):



Programming the MCU successfully tests the USB power and main regulator, as well as portions of the MCU and portions of the reset circuitry.

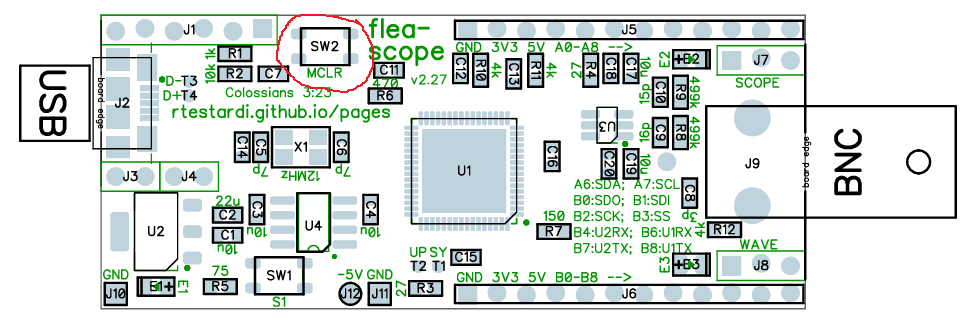
**At this point, your Flea-Scope should have both green and red LEDs lit up, and the blue LED should be blinking steady at 1 Hz:**



If the LEDs light up and flash, this tests large portions of the MCU, including flash self-programming (an error code flashed out the blue LED would be bad).

Finally, disconnect the programmer from the Flea-Scope (but leave its USB connector attached to the host computer).

**Next, press the reset button (labeled MCLR) and make sure the LEDs go out while the button is held** -- this tests the rest of the reset circuitry.



**Then release the reset button and the blue LED on the board should be blinking steady again at 1Hz, and red and green LEDs should be solid on.**

# Testing and Calibrating

Once the Flea-Scope is programmed and basically alive, it is time to test it more thoroughly as well as calibrate it.

First connect a 1x oscilloscope probe to the BNC connector of the Flea-Scope -- be sure the oscilloscope probe range switch (if present) is in the “1x” position:

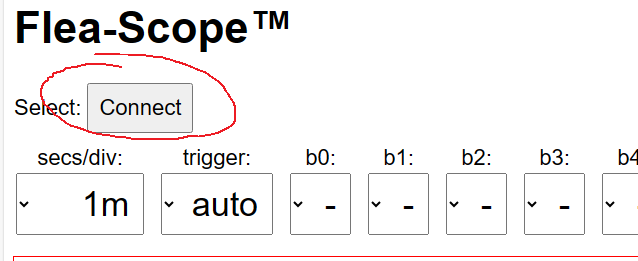
A close-up of a tool

Description automatically generated

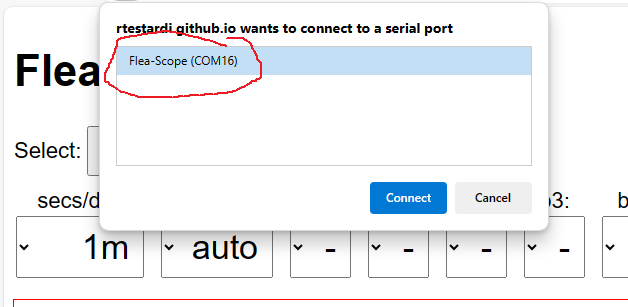
Then open the Flea-Scope User Interface web-page on the host computer:

<https://rtestardi.github.io/usbte/flea-scope.html>

And click the “Connect” button:



In the dialog box that pops up, select the Flea-Scope (you will likely have a different COM number than I show below) and then click “Connect” again:



**At this point you should be live accessing the Flea-Scope from the User Interface, and you should see it updating periodically!**

**The GUI should begin tracing, and all three LEDs should be blinking -- this tests USB data.**

The next step is to perform a probe calibration.

First we calibrate ground/zero by touching the oscilloscope probe tip to the ground/zero volt pin as shown below:

A close-up of a circuit board

Description automatically generated

Then press the “cal\_zero” button on the Flea-Scope User Interface:



Then we calibrate 3.3 volts by touching the oscilloscope probe tip to the 3.3 volt pin as shown below:

A close-up of a circuit board

Description automatically generated

Then press the “cal\_3v3” button on the Flea-Scope User Interface:

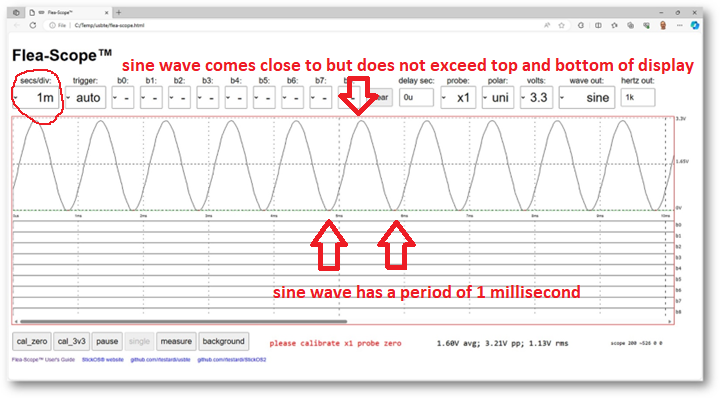


Finally, connect the oscilloscope probe to the wave output pin, and make sure you see a 1 kHz sine wave that approaches but does not actually touch the top and bottom of the display area:

A circuit board with a soldering tool

Description automatically generated

**The waveform on the Flea-Scope User Interface should look like this -- be sure it is full and unclipped:**

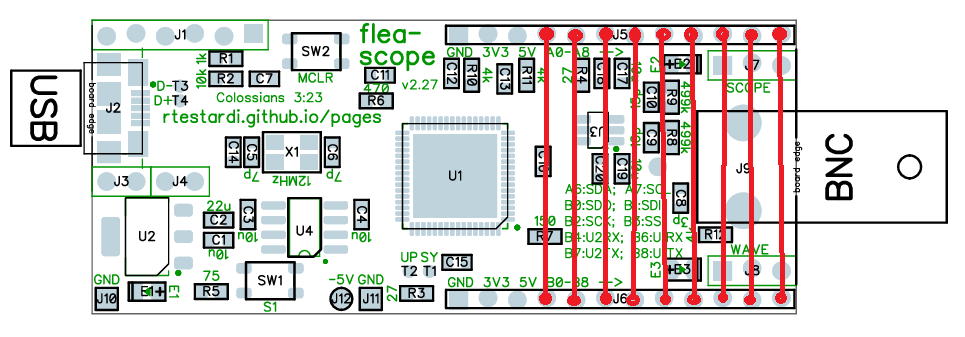


Note that if your oscilloscope probe range switch is incorrectly in the “10x” position, you will see a greatly attenuated waveform. If this happens, switch it to the “1x” position and perform all of the calibration steps again.

A screen shot of a graph

Description automatically generatedA proper waveform confirms the negative supply rail as well as the passive pullup/down circuitry, as well as the op amp as well as wave circuitry continuity.

For the final test, you need a small test harness that connects the “A” and “B” pins of the Flea-Scope together using spring/pogo pins like below (it is fine to also connect the first three power pins as well, like I did, but that is not needed):



A close-up of a circuit board

Description automatically generated

A green circuit board with many pins

Description automatically generated

**Put the board on the test bed (which simply connects all "A" pins to the corresponding "B" pins, switch the GUI to 5 us/division, and press the "test" button (labeled S1) -- you should see all nine digital signal lines (b0 thru b8) pulse in sequence as below** -- this tests the remaining MCU pins for shorts/opens.

(Notice there is a bit of noise induced on the floating analog input during this test, especially by "b8", but this is not a problem as the scope is careful to keep these signals drivel low during actual operation, forming a quasi-ground plane on the back-side of the board.)

A screenshot of a graph

Description automatically generated

# Troubleshooting

If your Flea-Scope is not working, the first things to check are the power supplies and clocks. Instructions for checking the power supplies are above.



To check the clocks, attach a working oscilloscope probe to the “UP and “SY” test points below and check the frequency:

* SY testpoint (system clock divided by 1024) should be 117.2 kHz
* UP testpoint (USB PLL divided by 1024) should be 46.9 kHz

Then check the USB testpoints:

* D- testpoint should be ~0V,
* D+ testpoint should be ~3V

Both of these are controlled by the MCU firmware, and allow the host computer to detect the Flea-Scope presence.