Flea-Scope Parts, Assembly, and Test

Contents

Parts

U1 PIC32MK0512GPK064-I/MR has allowed substitutions (check availability and price for cheapest at Microchip Direct):

- PIC32MK0512GPK064-I/MR is first choice and usually cheapest at $7.20 in quantities of 1000's
- PIC32MK1024GPK064-I/MR is second choice at $7.40
- PIC32MK0512MCM064-I/MR at $7.51
- PIC32MK1024MCM064-I/MR 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. (A more expensive part OPA810IDBVT may be used in the future if I add real AC coupling and front-end amplification ranges.)

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.

Circuit Board

The latest circuit board layout has been optimized for ENIG surface finish. The pin strip through holes (not populated during assembly) are staggered slightly to grab the pin strips, to allow end users to conveniently insert pin strips without necessarily even needing to solder them. The staggering assumes the thinner ENIG surface finish. (So please do not use the thicker HASL surface finish, as the pin strip through holes will then be staggered too much.)

Panel boards as needed but please depanel prior to soldering the **single through-hole component** (the BNC connector) and programming and test.

Assembly

See the assembly PDF and pickplace CSV files for SMT details.

Please also solder the **single through-hole component** (the BNC connector).

Programming

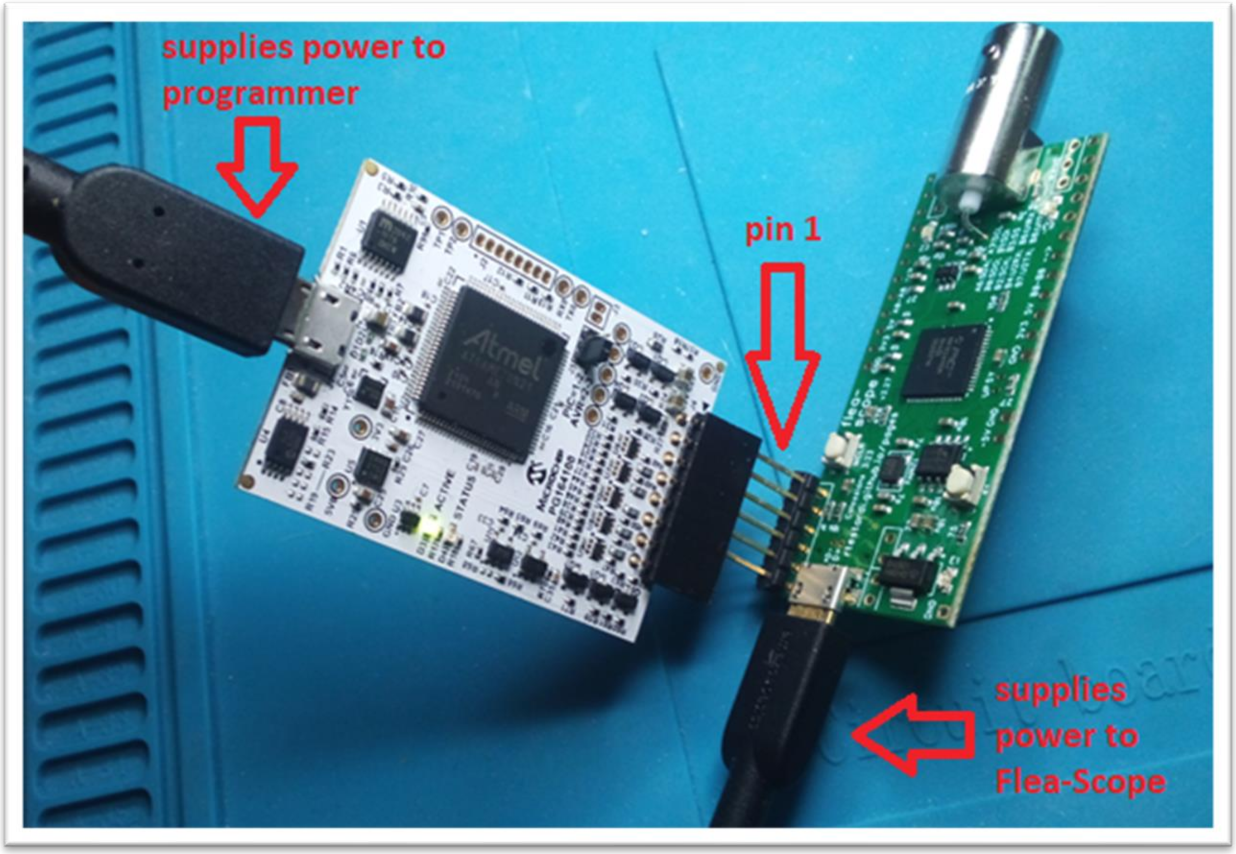
After assembly and depaneling, the first step is to connect **both** the USB connector (for power) on the Flea-Scope board **and** the USB connector for the programmer (for programming) to the host computer (I use Windows 11 below). **Then** connect the programmer ICSP pins to the 6 board ICSP pins as 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.

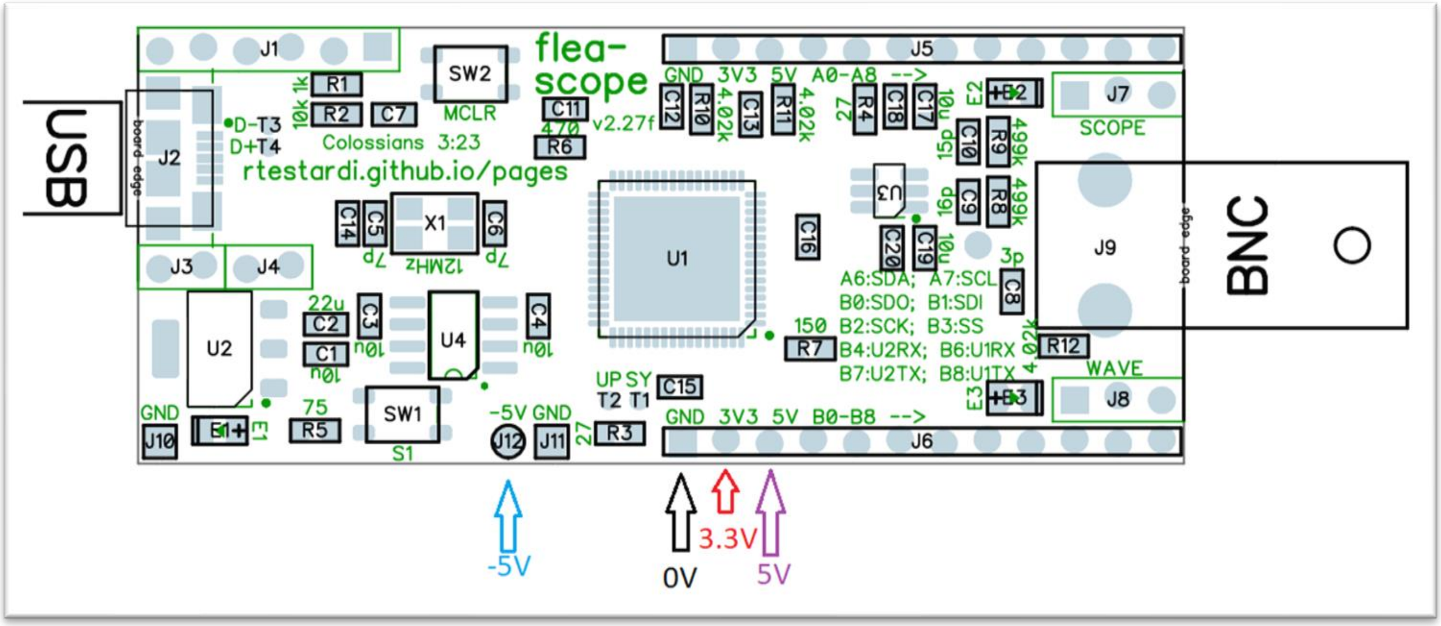
You can use any of these programmers:

- 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.



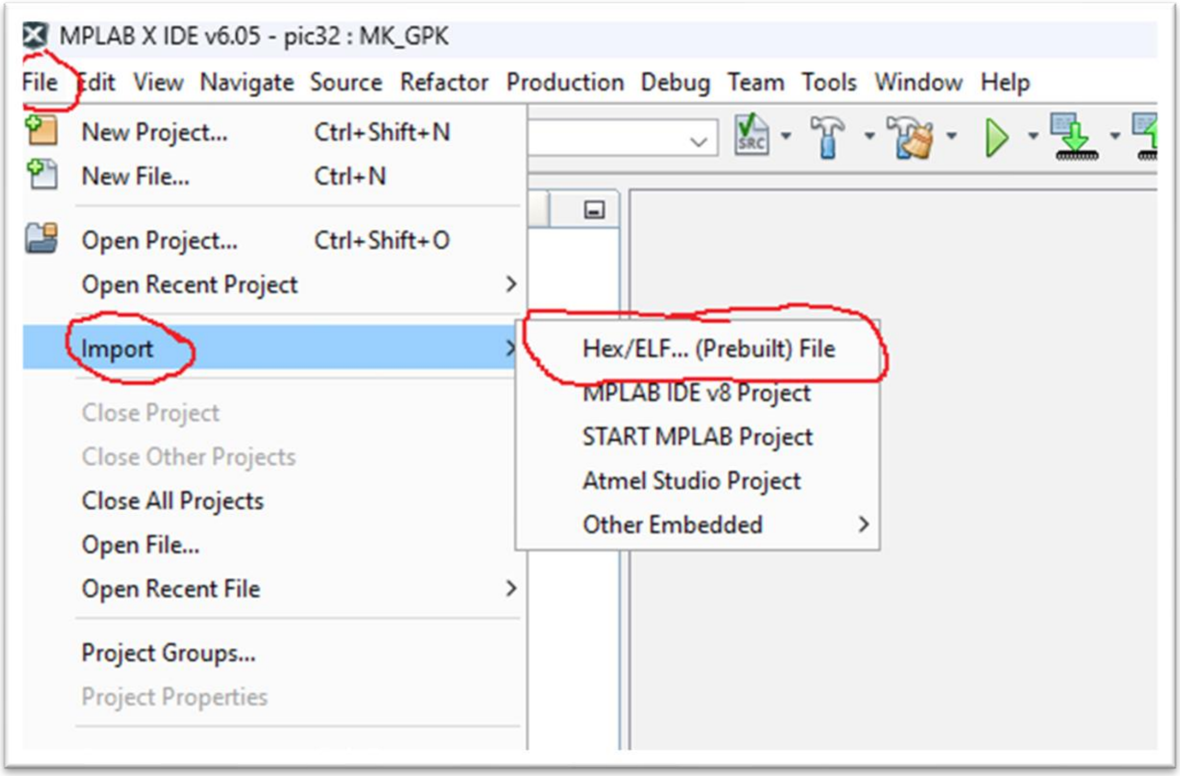
0. Once USB is connected to the Flea-Scope, you can also verify the Flea-Scope board has power by measuring these pins (5V and -5V are unregulated, so they are approximate -- anything within 0.25V is good enough):



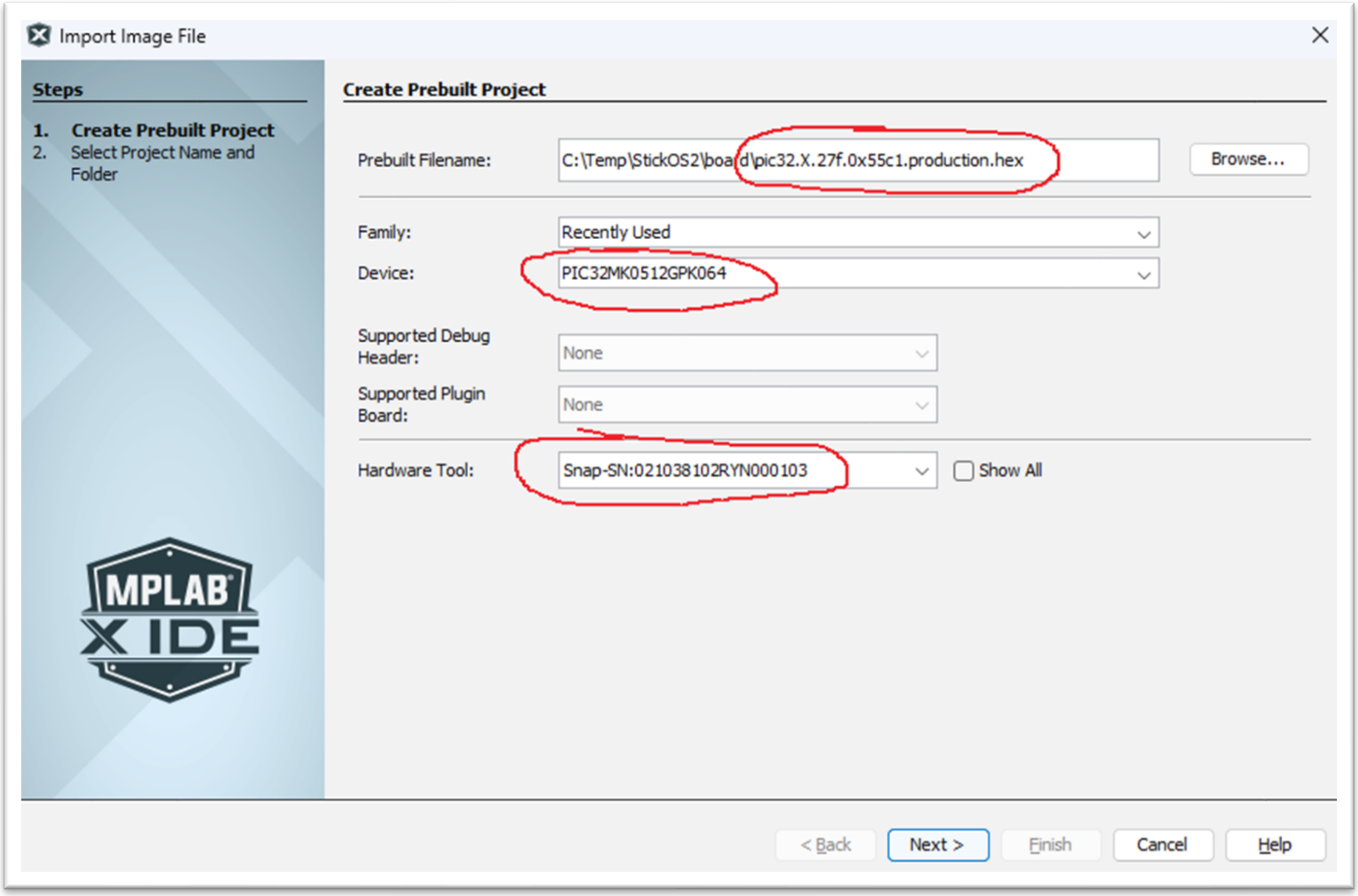
To program, I use MPLAB X IDE v6.20 which is available here: [MPLAB® X IDE | Microchip Technology]

(Pictures below are from v6.05, but the process is the same. Note the pictures may also show older firmware.)

The next step is to start MPLAB X IDE v6.20 and import the firmware hex (prebuilt) file:



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



Create Prebuilt Project

Prebuilt Filename: C:\Temp\StickOS2\board\pic32.X.27f.0x55c1.production.hex

Family: Recently Used

Device: PIC32MK0512GPK064

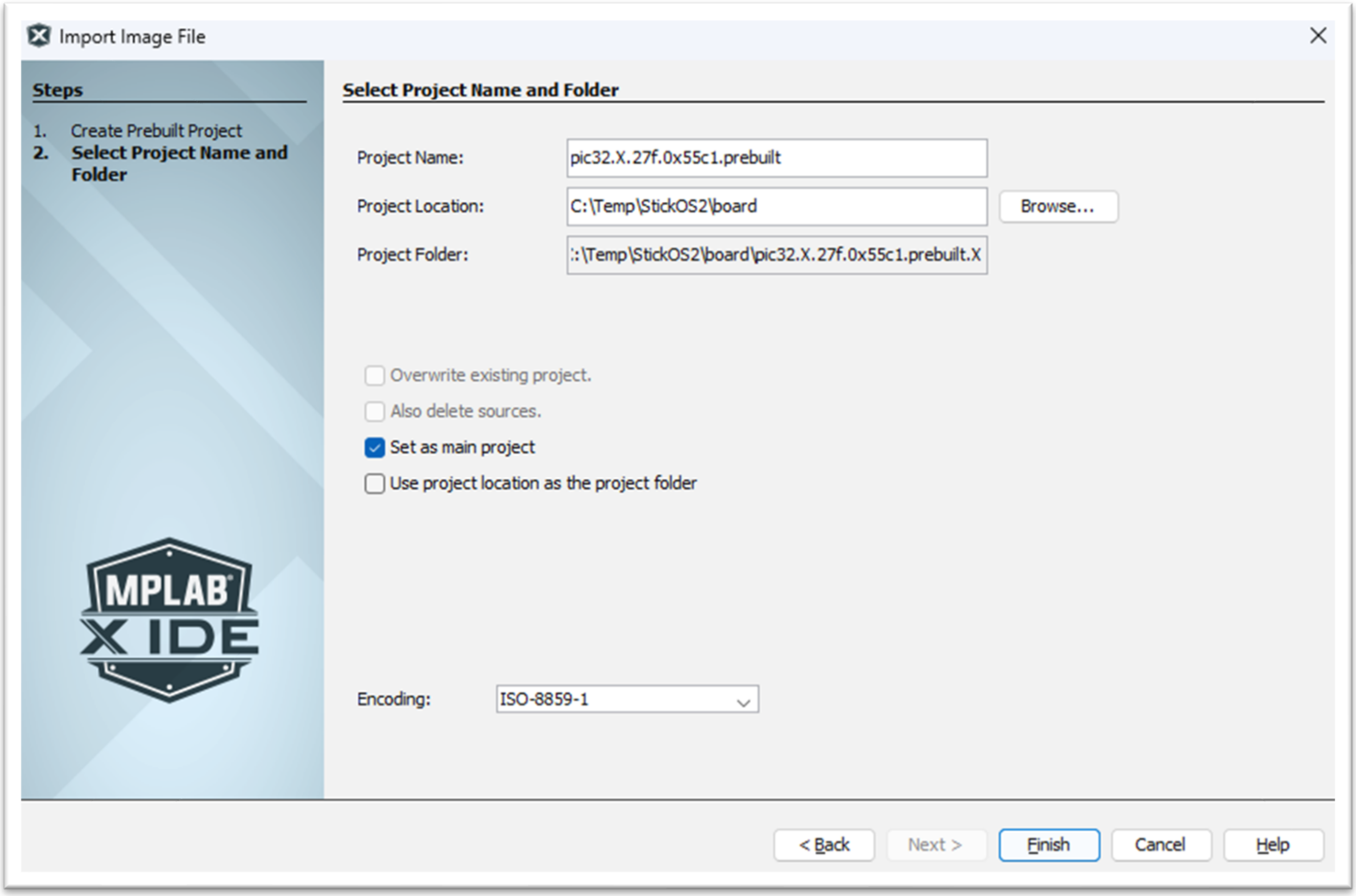
Supported Debug Header: None

Supported Plugin Board: None

Hardware Tool: Snap-SN:021038102RYN000103 □ Show All

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:



Select Project Name and Folder

Project Name: pic32.X.27f.0x55c1.prebuilt

Project Location: C:\Temp\StickOS2\board Browse...

Project Folder: :\Temp\StickOS2\board\pic32.X.27f.0x55c1.prebuilt.X

□ Overwrite existing project.

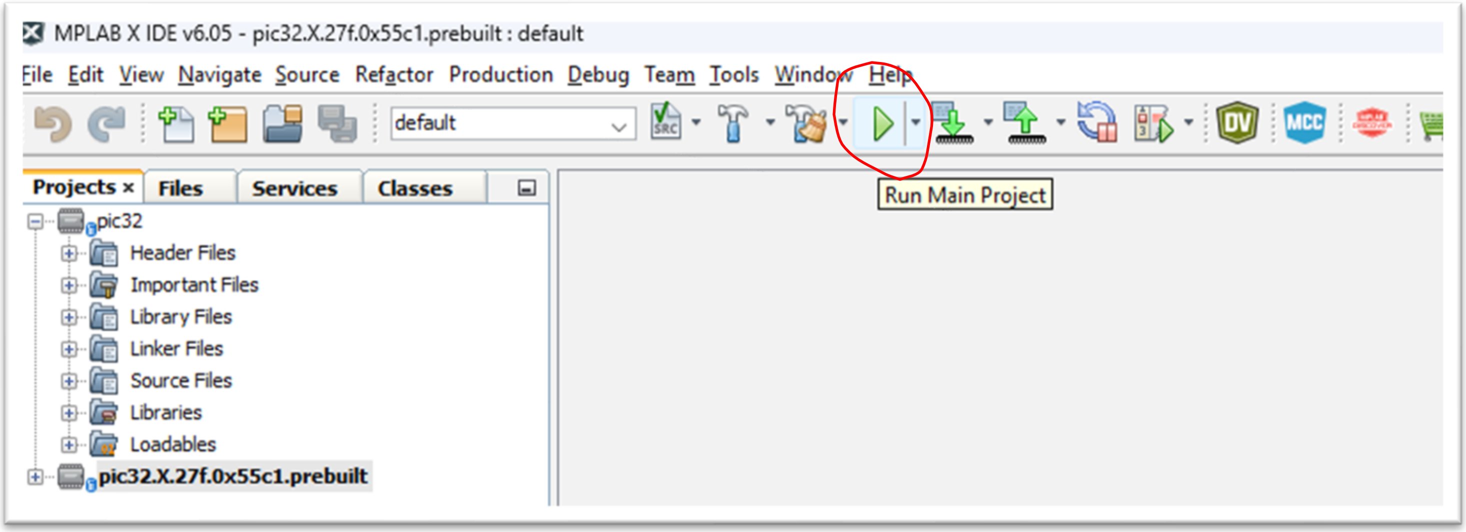
□ Also delete sources.

☑ Set as main project

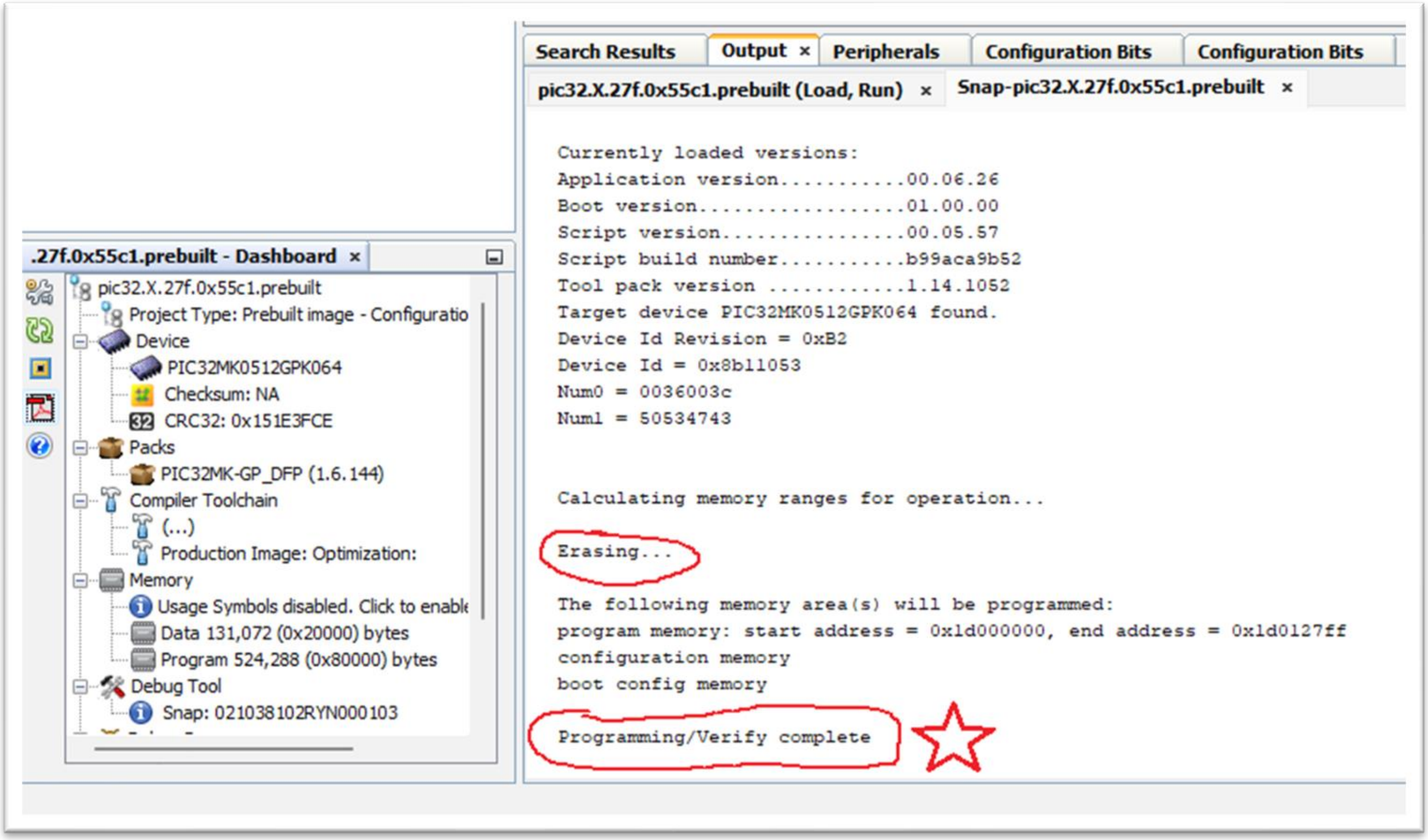
□ Use project location as the project folder

Encoding: ISO-8859-1

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


MPLAB X IDE v6.05 - pic32.X.27f.0x55c1.prebuilt : default
File Edit View Navigate Source Refactor Production Debug Team Tools Window Help
default
DV
MCC
Run Main Project
Projects × Files Services Classes
pic32
Header Files
Important Files
Library Files
Linker Files
Source Files
Libraries
Loadables
pic32.X.27f.0x55c1.prebuilt

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):

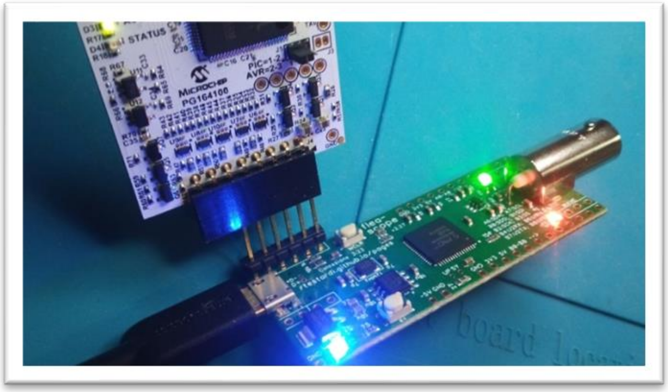

Search Results
Output
Peripherals
Configuration Bits
Configuration Bits
pic32.X.27f.0x55c1.prebuilt (Load, Run)
Snap-pic32.X.27f.0x55c1.prebuilt
.27f.0x55c1.prebuilt - Dashboard

- pic32.X.27f.0x55c1.prebuilt
  - Project Type: Prebuilt image - Configuratio
  - Device
    - PIC32MK0512GPK064
    - Checksum: NA
    - CRC32: 0x151E3FCE
  - Packs
    - PIC32MK-GP_DFP (1.6.144)
  - Compiler Toolchain
    - (...)
    - Production Image: Optimization:
  - Memory
    - Usage Symbols disabled. Click to enable
    - Data 131,072 (0x20000) bytes
    - Program 524,288 (0x80000) bytes
  - Debug Tool
    - Snap: 021038102RYN000103

```
Currently loaded versions:
Application version...........00.06.26
Boot version..................01.00.00
Script version................00.05.57
Script build number...........b99aca9b52
Tool pack version ............1.14.1052
Target device PIC32MK0512GPK064 found.
Device Id Revision = 0xB2
Device Id = 0x8b11053
Num0 = 0036003c
Num1 = 50534743


Calculating memory ranges for operation...

Erasing...

The following memory area(s) will be programmed:
program memory: start address = 0x1d000000, end address = 0x1d0127ff
configuration memory
boot config memory

Programming/Verify complete
```

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

1. 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:

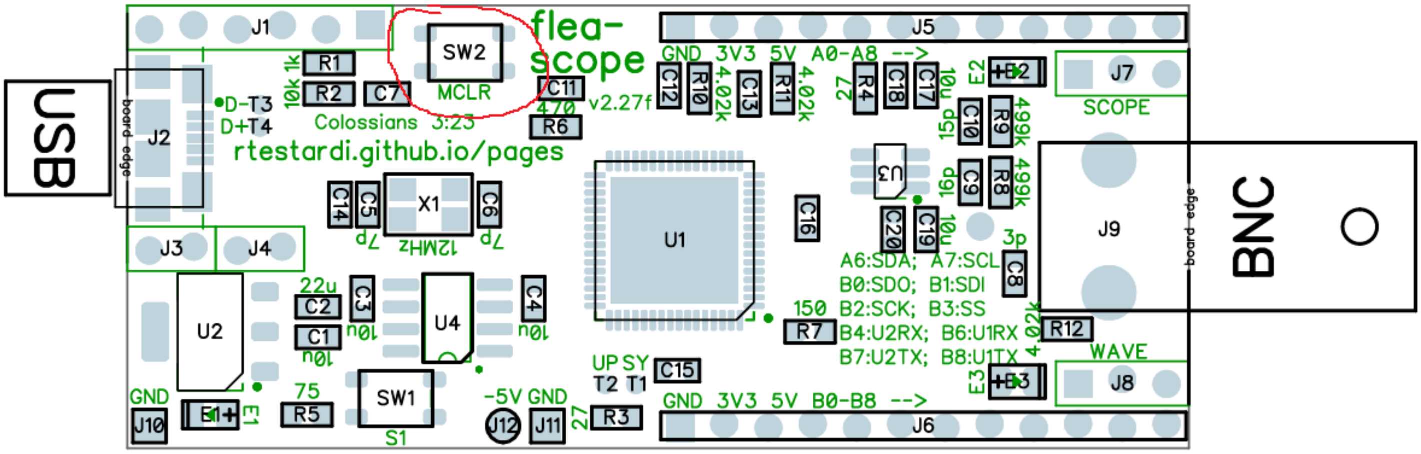


If the LEDs light up and flash, this tests large portions of the MCU, including flash self-programming (an error code flashed out the blue LED would be bad).

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

Testing and Calibrating

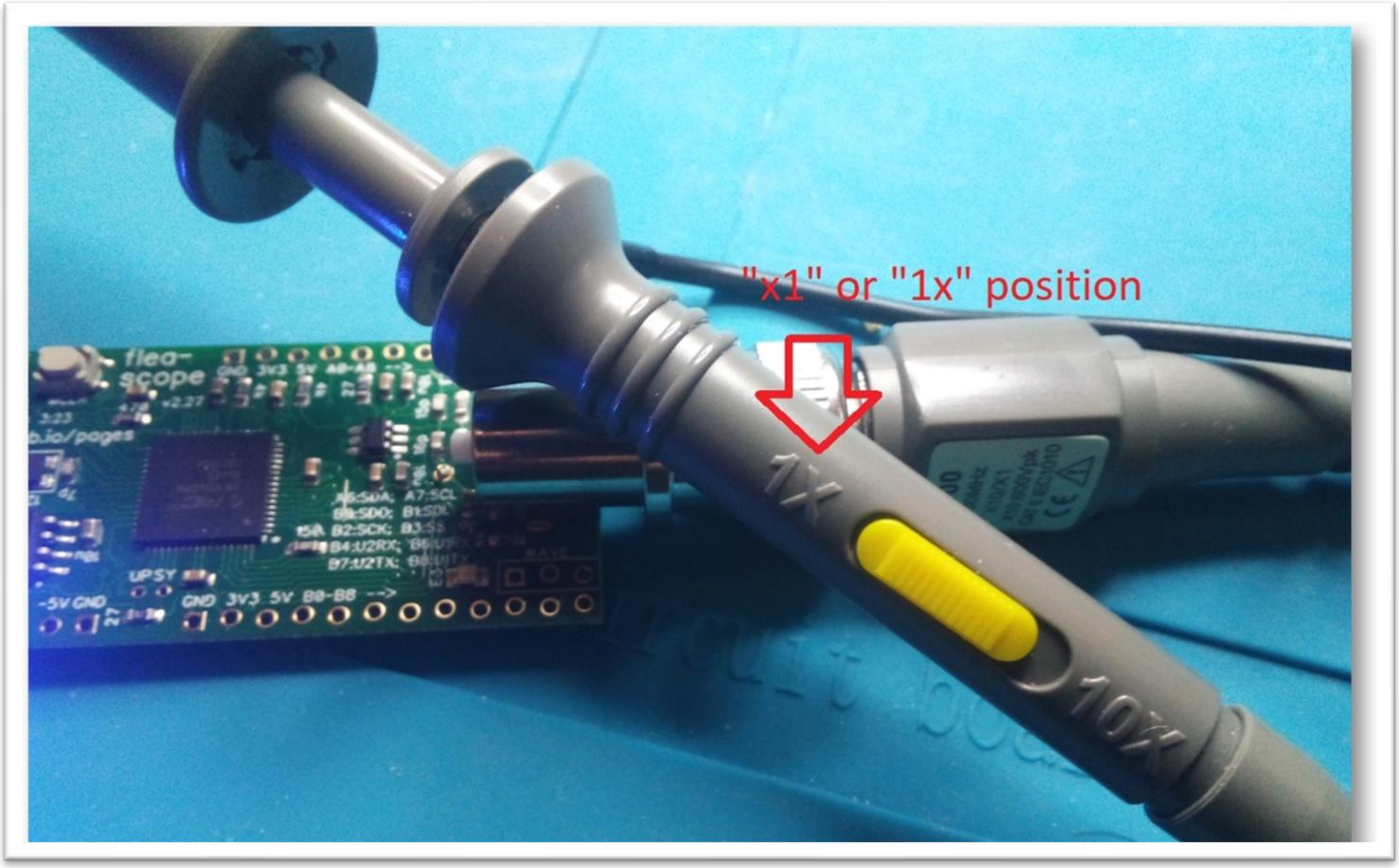
2. Next, press the reset button (labeled MCLR) and make sure the LEDs go out while the button is held -- this tests the rest of the reset circuitry, including the button.



3. 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.

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

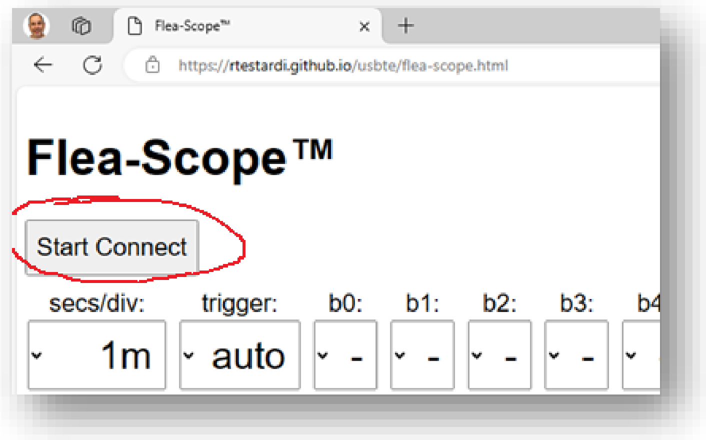
First connect a 1x oscilloscope probe to the BNC connector of the Flea-Scope -- be sure the oscilloscope probe range switch (if present) is in the “1x” position:



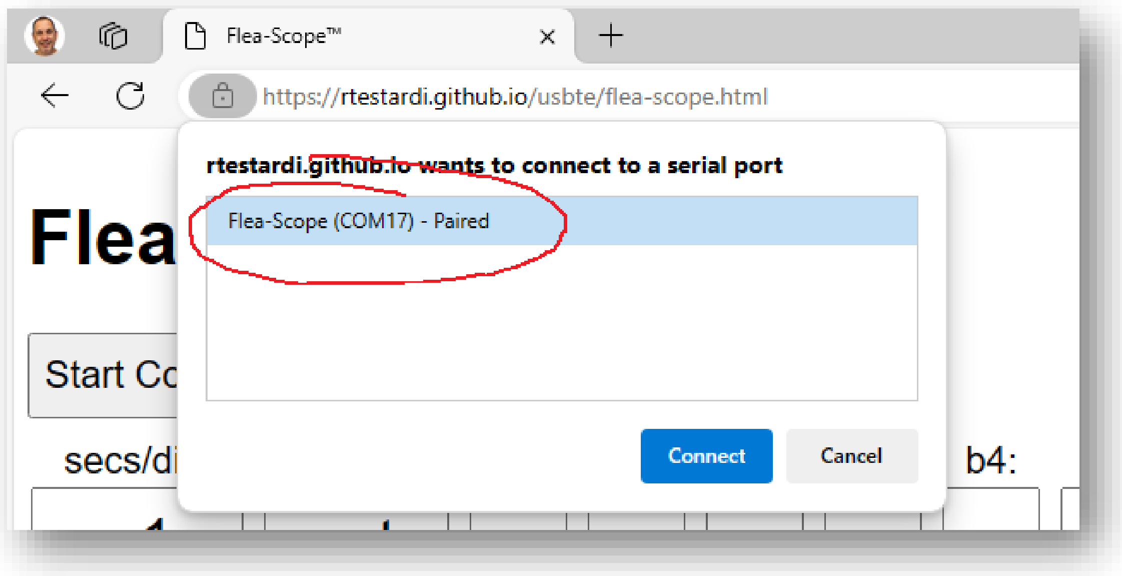
Then open the Flea-Scope User Interface web-page on the host computer:

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

And click the “Start Connect” button:



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

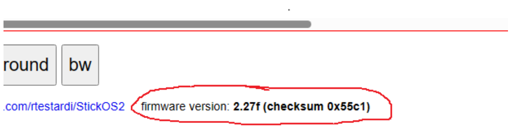


rtestardi.github.io wants to connect to a serial port

Flea-Scope (COM17) - Paired

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

5. Make sure the Flea-Scope firmware is the proper version “2.27g (checksum 0x7c8e)” at the bottom of the screen (note pictures below may show older versions):

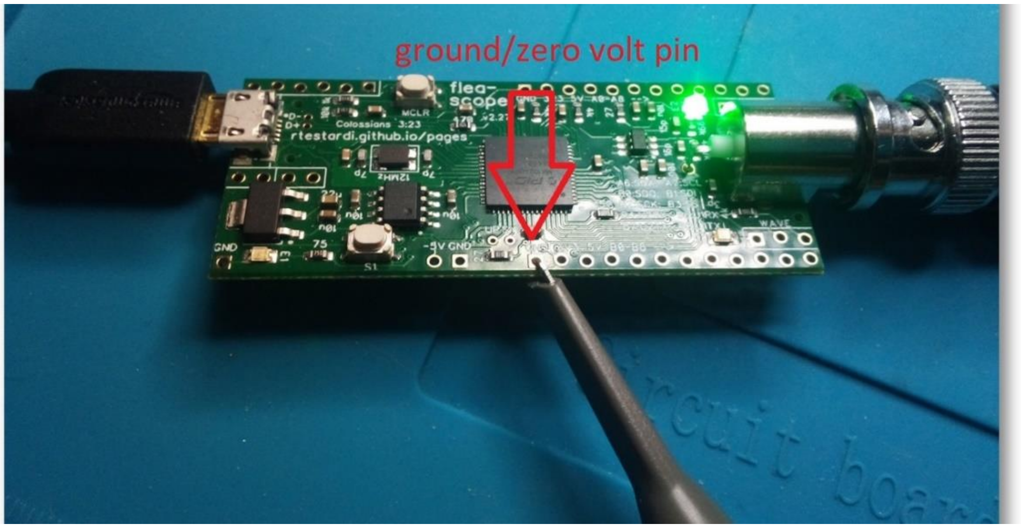


.com/rtestardi/StickOS2 firmware version: **2.27f (checksum 0x55c1)**

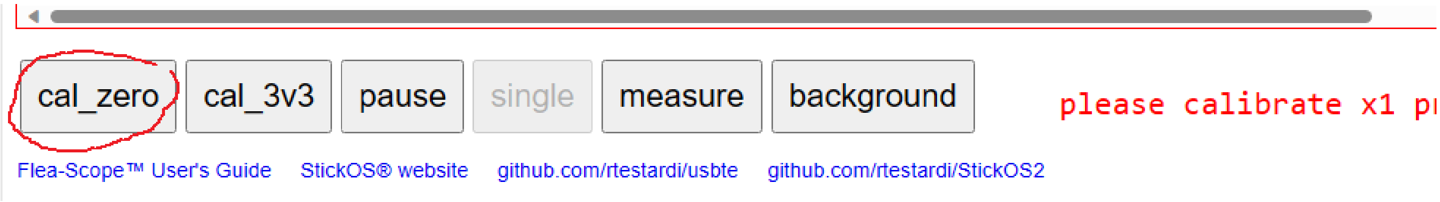
6. The GUI should begin tracing, and all three LEDs should be blinking -- this tests USB data.

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:



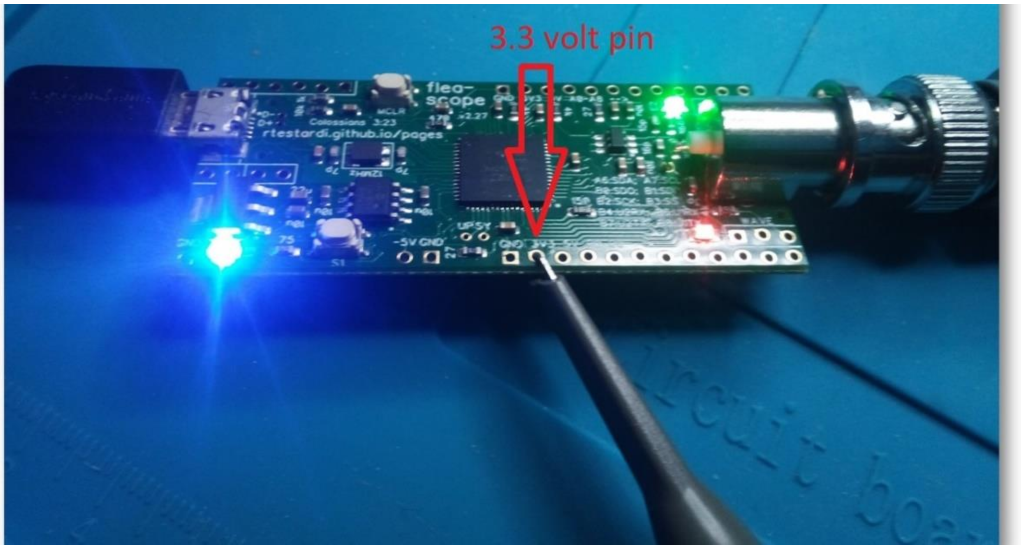
Then press the “cal_zero” button on the Flea-Scope User Interface:



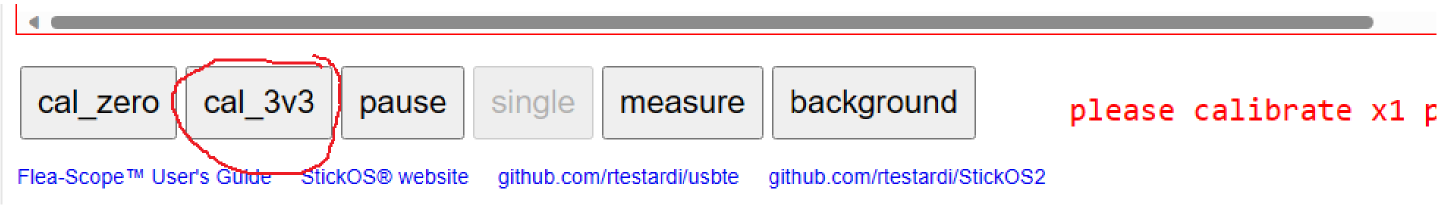
please calibrate x1 pı

7. The cal_zero_x1 value should be displayed within the range of 7 to 23.

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



Then press the “cal_3v3” button on the Flea-Scope User Interface:

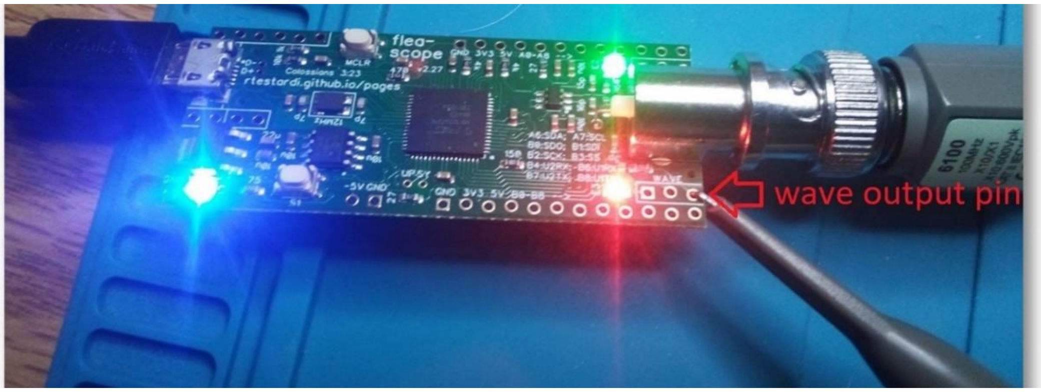


please calibrate x1 p

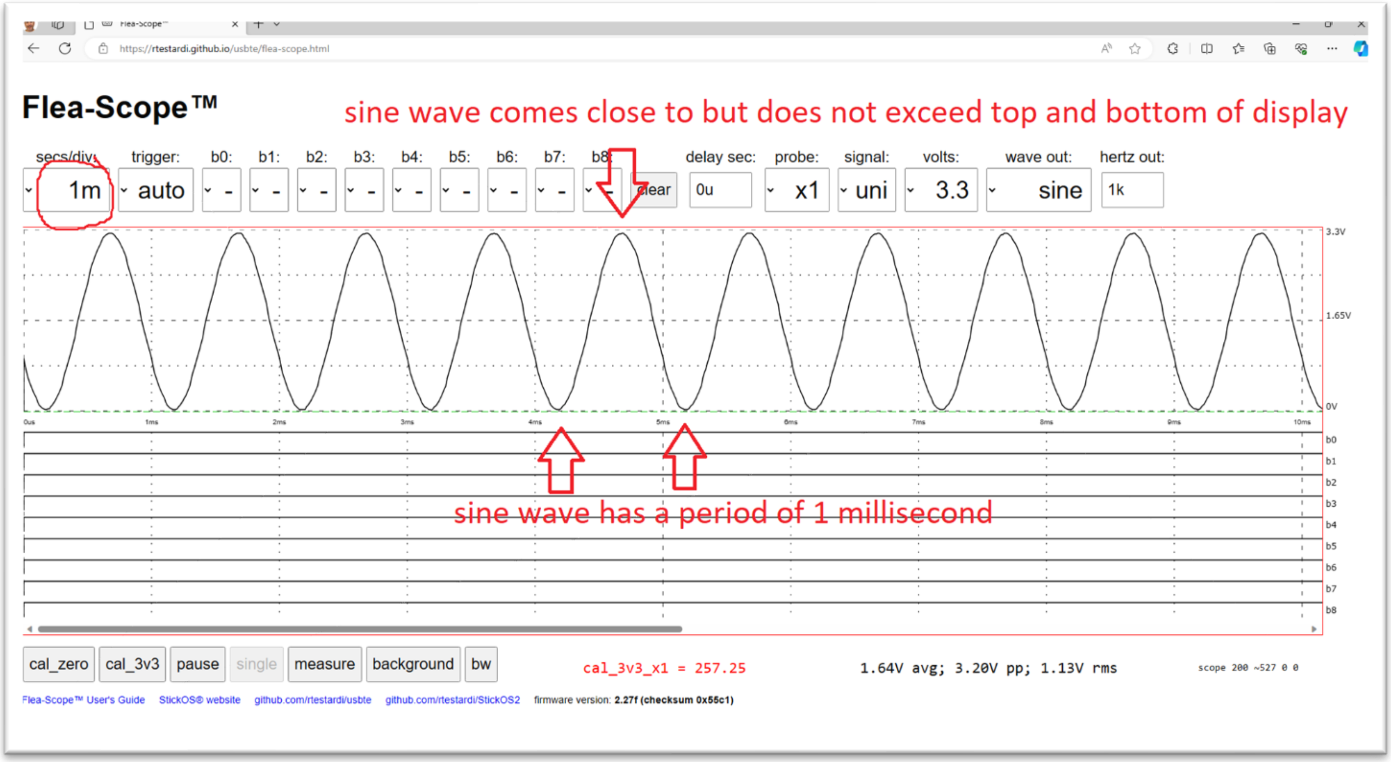
8. The cal_3v3_x1 value should be displayed within the range of 245 to 275.

The presence of proper calibration values tests the negative supply rail as well as the passive pullup/down circuitry, as well as the op amp. To force calibration even if the values are out of range, press the "measure" button first.

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:



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


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

Flea-Scope™

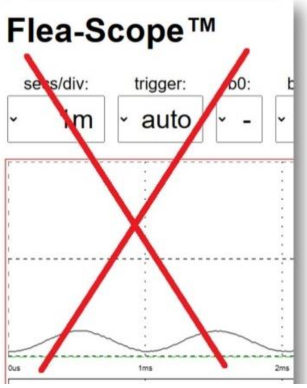
sine wave comes close to but does not exceed top and bottom of display

secs/div: 1m trigger: auto b0: - b1: - b2: - b3: - b4: - b5: - b6: - b7: - b8: - delay sec: 0u probe: x1 signal: uni volts: 3.3 wave out: sine hertz out: 1k

sine wave has a period of 1 millisecond

```
cal_3v3_x1 = 257.25          1.64V avg; 3.20V pp; 1.13V rms          scope 200 ~527 0 0
```

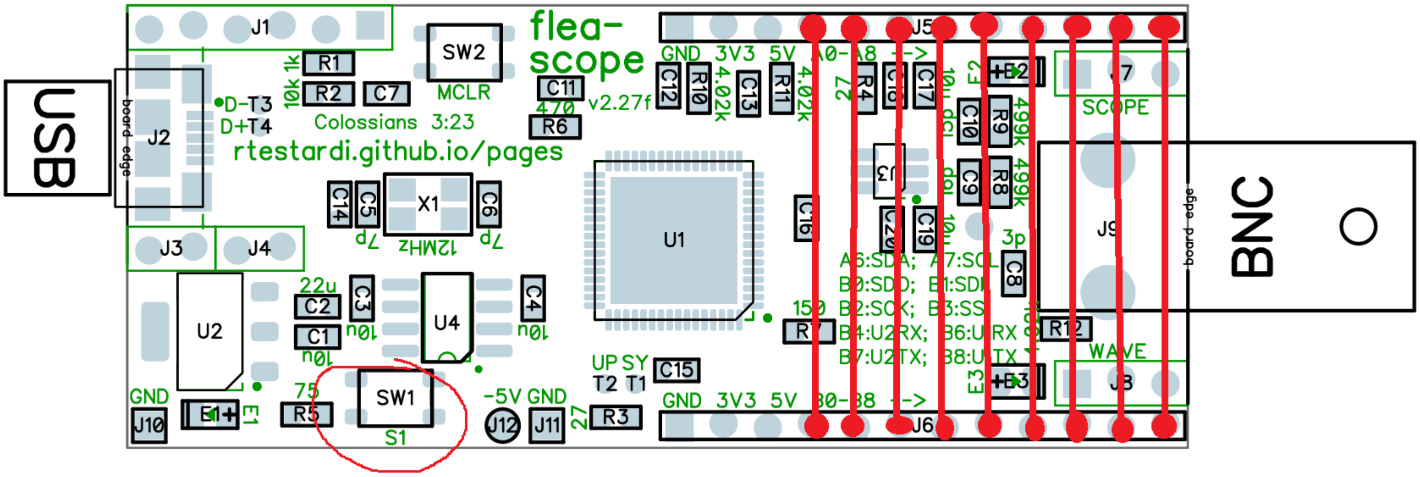
Flea-Scope™ User's Guide   StickOS® website   github.com/rtestardi/usbte   github.com/rtestardi/StickOS2   firmware version: **2.27f (checksum 0x55c1)**

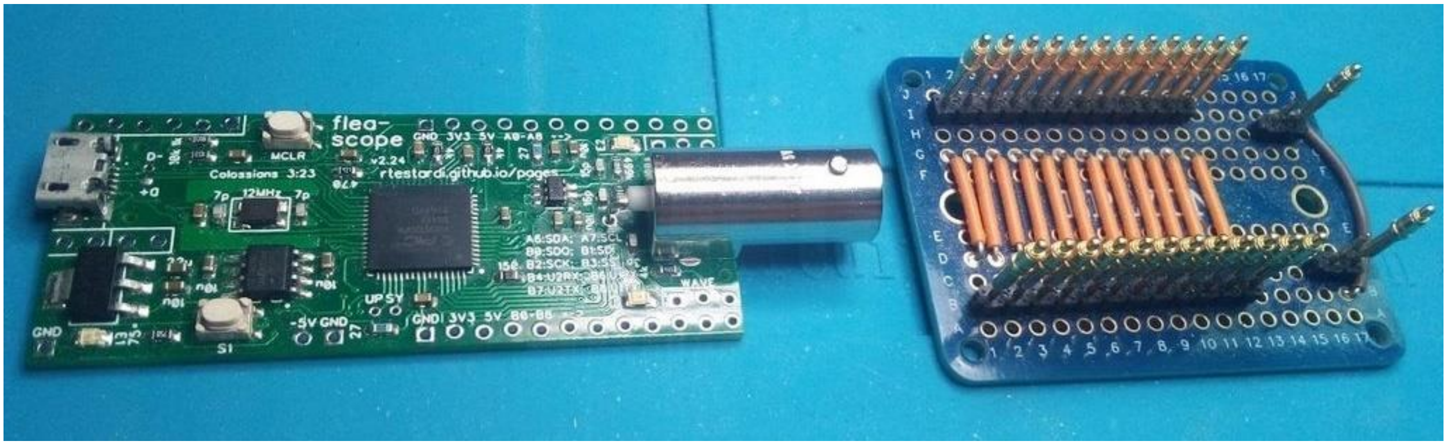


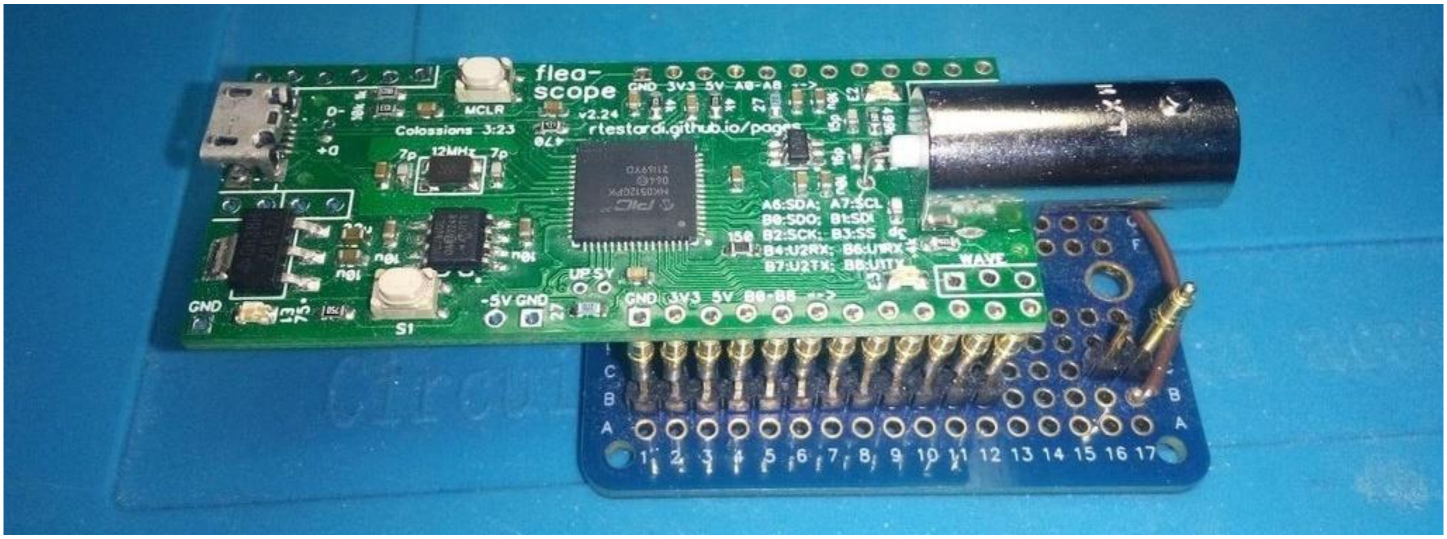
Note that if your oscilloscope probe range switch is incorrectly in the "10x" position, you will see a greatly attenuated waveform. If this happens, switch it to the "1x" position and perform all of the calibration steps again.

A 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):

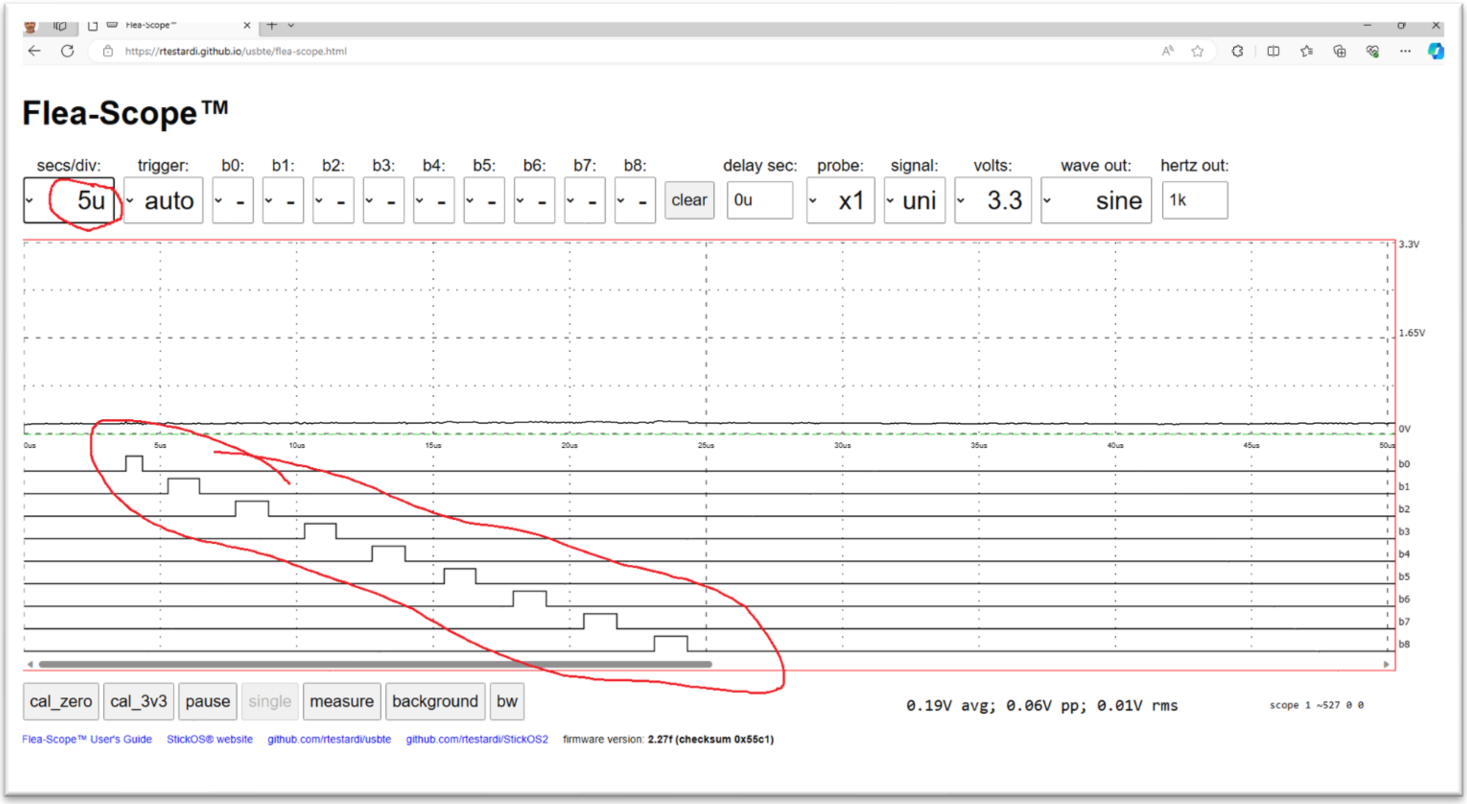






10. Put the board on the test bed (which simply connects all "A" pins to the corresponding "B" pins, switch the GUI to 5 us/division, and press the "test" button (labeled S1) -- you should see all nine digital signal lines (b0 thru b8) pulse in sequence without any overlap (!) as below -- this tests the remaining MCU pins for shorts/opens and the S1 button.

(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.)



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

Flea-Scope™

secs/div: 5u; trigger: auto; b0: -; b1: -; b2: -; b3: -; b4: -; b5: -; b6: -; b7: -; b8: -; clear; delay sec: 0u; probe: x1; signal: uni; volts: 3.3; wave out: sine; hertz out: 1k

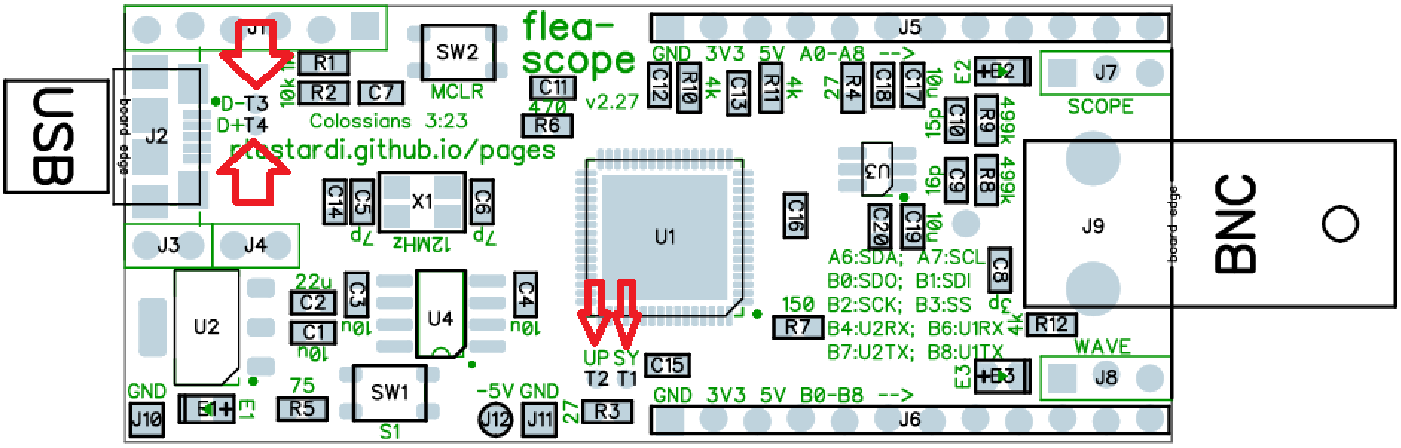
cal_zero | cal_3v3 | pause | single | measure | background | bw

```
0.19V avg; 0.06V pp; 0.01V rms          scope 1 ~527 0 0
```

Flea-Scope™ User's Guide   StickOS® website   github.com/rtestardi/usbte   github.com/rtestardi/StickOS2   firmware version: **2.27f (checksum 0x55c1)**

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 (these are controlled by firmware to allow the host computer to detect the Flea-Scope):

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

Packaging

Please individually static bag each tested board, and then bulk pack them securely for shipping. Thank you!

Test Checklist

	indicate check mark for successful test									
board number	1	2	3	4	5	6	7	8	9	10
0. 3.3 volt, 5 volt, and -5 volt power supplies are good										
1. programming successful, blue LED is flashing, red/green LEDs are on										
2.pressing reset (MCLR) button clears all LEDs										
3. LEDs light again when reset button (MCLR) is released										
4. Web-page user interface connects successfully to Flea-Scope!										
5. red, green, and blue LEDs are all flashing as user interface traces										
6. confirm firmware version is “2.27g (checksum 0x7c8e)”										
7. x1 zero/ground calibration succeeds with value in range 7 to 23 *										
8. x1 3.3 volt calibration succeeds with value in range 245 to 275 *										
9. the 1 kHz test waveform in web-page user interface looks good and is nearly full height and unclipped!										
10. the testbed pulses when pressing S1 button at 5us/div are shown in sequence without any overlap!										

* If any boards fail calibration, please mark them for special attention by me -- they may still be fine, but I may need to adjust the calibration limits in the web-page software.

	indicate check mark for successful test									
board number	1	2	3	4	5	6	7	8	9	10
0. 3.3 volt, 5 volt, and -5 volt power supplies are good										
1. programming successful, blue LED is flashing, red/green LEDs are on										
2.pressing reset (MCLR) button clears all LEDs										
3. LEDs light again when reset button (MCLR) is released										
4. Web-page user interface connects successfully to Flea-Scope!										
5. red, green, and blue LEDs are all flashing as user interface traces										
6. confirm firmware version is “2.27g (checksum 0x7c8e)”										
7. x1 zero/ground calibration succeeds with value in range 7 to 23 *										
8. x1 3.3 volt calibration succeeds with value in range 245 to 275 *										
9. the 1 kHz test waveform in web-page user interface looks good and is nearly full height and unclipped!										
10. the testbed pulses when pressing S1 button at 5us/div are shown in sequence without any overlap!										

	indicate check mark for successful test									
board number	1	2	3	4	5	6	7	8	9	10
0. 3.3 volt, 5 volt, and -5 volt power supplies are good										
1. programming successful, blue LED is flashing, red/green LEDs are on										
2.pressing reset (MCLR) button clears all LEDs										
3. LEDs light again when reset button (MCLR) is released										
4. Web-page user interface connects successfully to Flea-Scope!										
5. red, green, and blue LEDs are all flashing as user interface traces										
6. confirm firmware version is “2.27g (checksum 0x7c8e)”										
7. x1 zero/ground calibration succeeds with value in range 7 to 23 *										
8. x1 3.3 volt calibration succeeds with value in range 245 to 275 *										
9. the 1 kHz test waveform in web-page user interface looks good and is nearly full height and unclipped!										
10. the testbed pulses when pressing S1 button at 5us/div are shown in sequence without any overlap!										