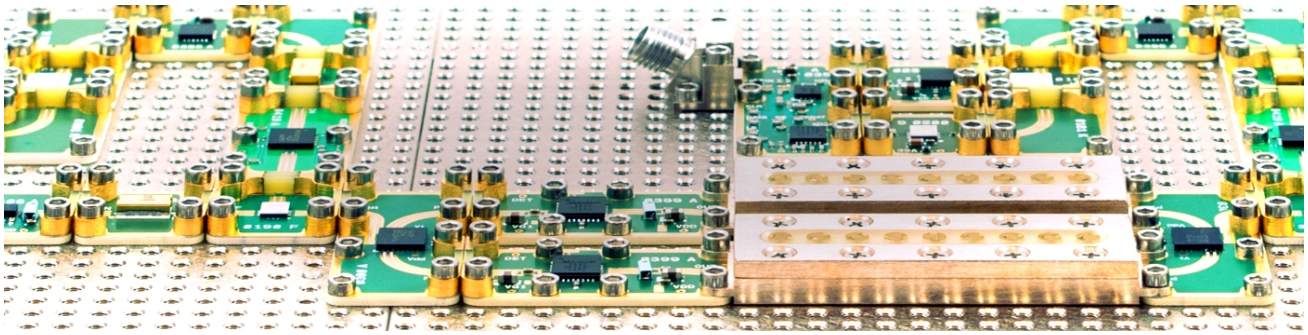
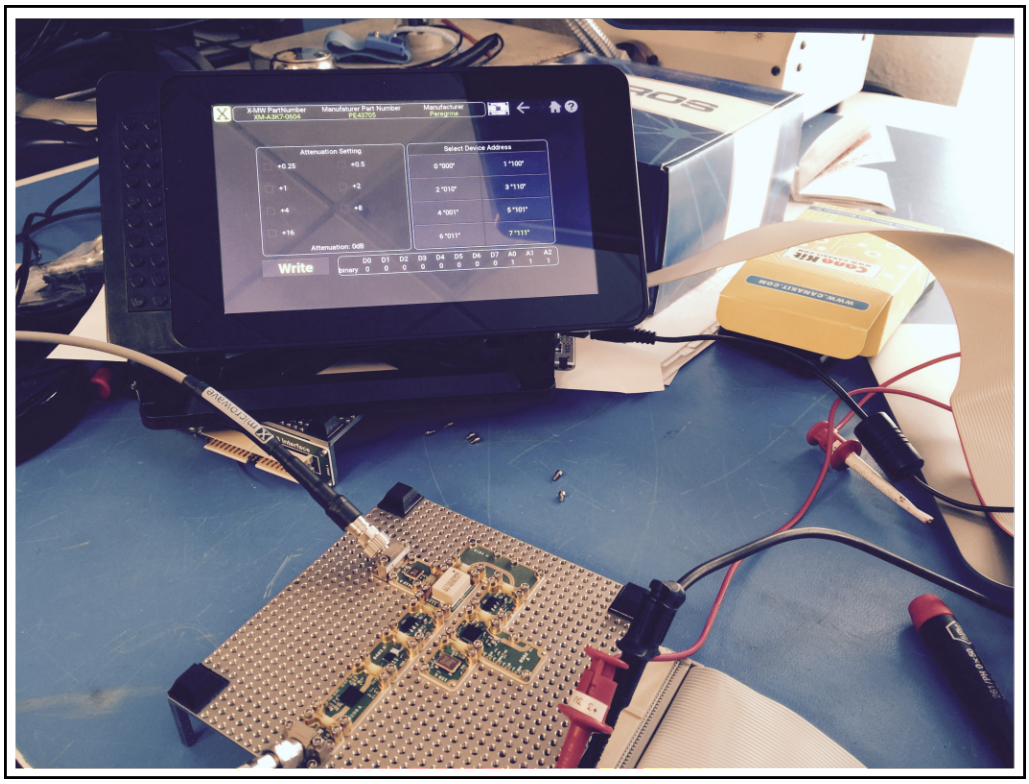
Getting Started: X-MWcontroller



1. X-MWsystem Control with the X-MWcontroller

1.1 Overview

The X-Microwave X-MWcontroller provides a fully standalone, easy-to use, plug-and-play interface for controlling all programmable X-MWblocks on the X-Microwave format using a RaspberryPi computer and touchscreen interface.



Background

Part manufacturer's typically offer USB dongles with customized software to control their devices using serial and parallel interfaces. The number of USB dongles and control software packages grows when multiple devices from multiple manufacturers are used. This can be cumbersome to manage and make it difficult to pick up where you left off the previous day.

X-Microwave addresses this challenge with the X-MWcontroller. This RaspberryPi based controller provides a standalone interface for configuring and controlling any manufacturer's

device available on an X-MWblock. It provides a consistent interface across all manufacturers and devices. All settings can be configured, saved, loaded, and exported allowing you to streamline your design flow.

Control Basics

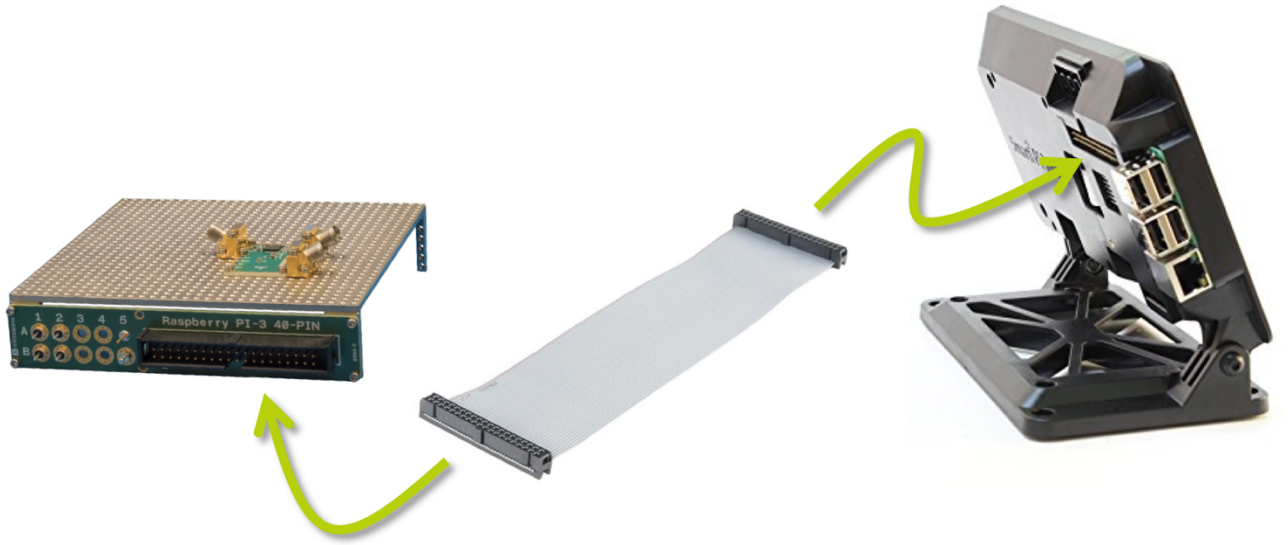
Programmable RF and microwave components are often controlled using a digital interface with either parallel or serial control.

Parallel

Parallel control refers to direct control of a device using individual digital control lines. The primary advantage of parallel control is that it is fast and relatively simple to use, but is limited to simple on-off signaling. An example of a parallel control device is a SPST (single pole single throw) switch where 0V is the default state and 3.3V activates the switch.

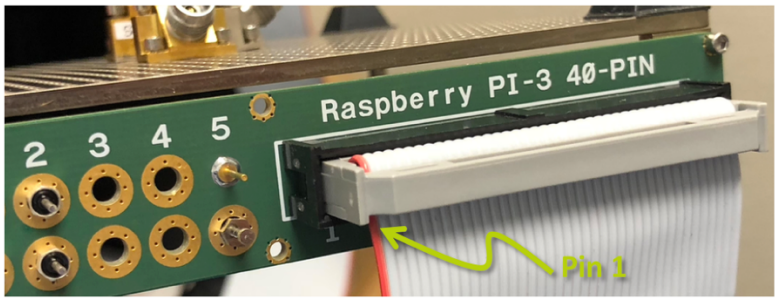
Serial

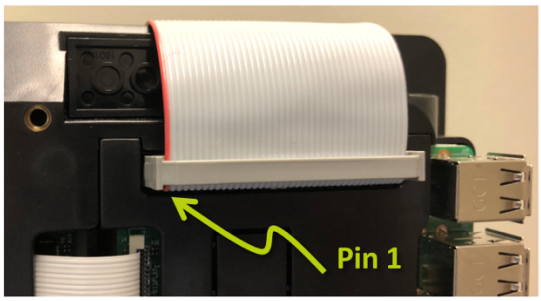
Serial (SPI) control refers to a digital communication method where data is written to a device using a wire based digital protocol. The data is written to memory registers in the device allowing multiple parameters to be set using a 3-wire connection (data, clock, enable). An example of a serial control device is a programmable PLL that can be configured through register settings for control and operation of the device including start-up behavior, phase adjustment, output power level, and more.



Basic Setup

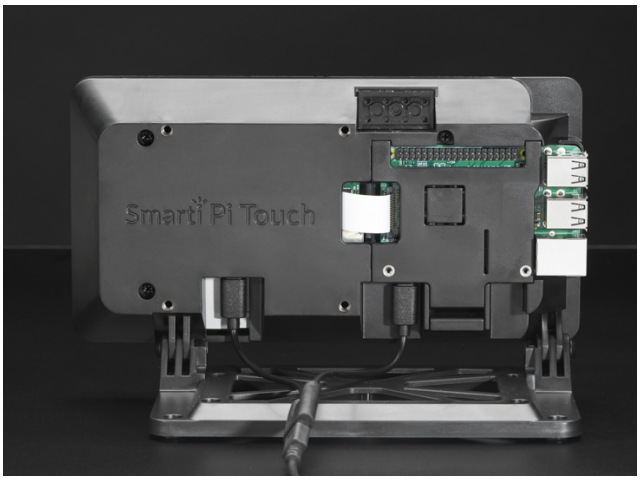
The connection between the prototyping plate and X-MWcontroller is achieved through a common 40 pin ribbon cable. Proper cable orientation is critical. The red line on the cable should match up with PIN 1 by convention. PIN 1 is located on the left side on both the RaspberryPi and the X-Microwave prototyping plate when you are facing the connector.





Raspberry Pi Touch Platform

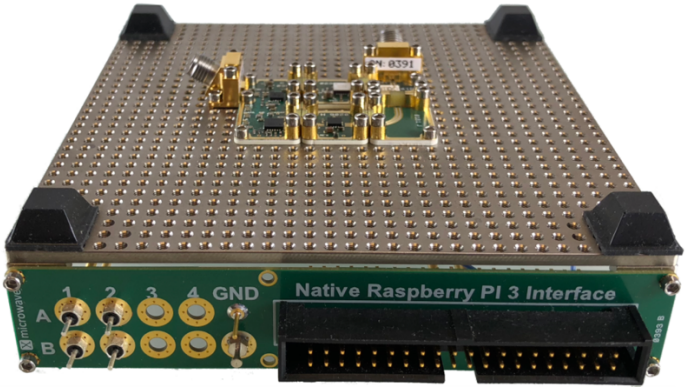
The X-Microwave RaspberryPi Touch kit includes the RaspberryPi 3, touch screen, plastic case, US power adapter, USB power splitter adapter, ribbon cable, and pre-loaded memory card with OS and X-Microwave software. This is everything required to connect the RaspberryPi to your X-Microwave prototyping plate.



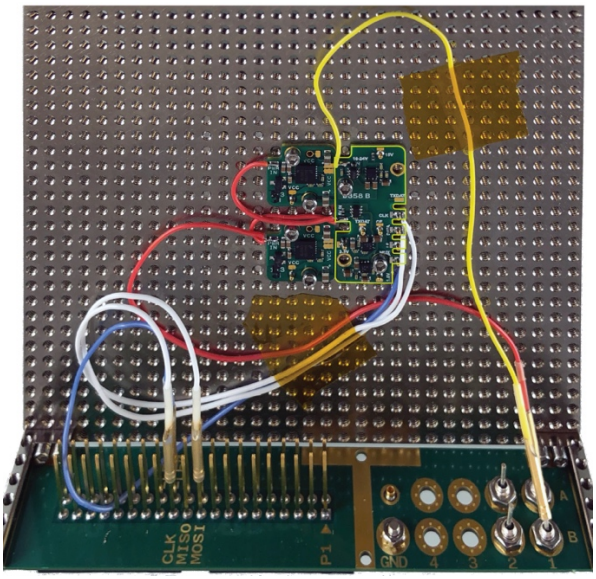
Making the Connections

First you'll need to solder jumper wires to the bias and control boards and attach them to the back side of the interface board. Feedtru capacitors are typically used for power and the 40 pin connector for digital control and data. From the front side power can be attached with clip-on leads from a power supply. A post is provided for ground (which is a common ground to the entire prototyping plate surface.) Connection of digital I/O connections is unique the communication style and is covered in the next section. Once wiring is complete, the prototyping plate can be connected to the RaspberryPi using the standard 40 pin ribbon cable.

For your convenience, X-Microwave offers several colors of jumper wire cut on one end and socketed on the other for soldering to the bias controller and plug-on attachment to the interface board.



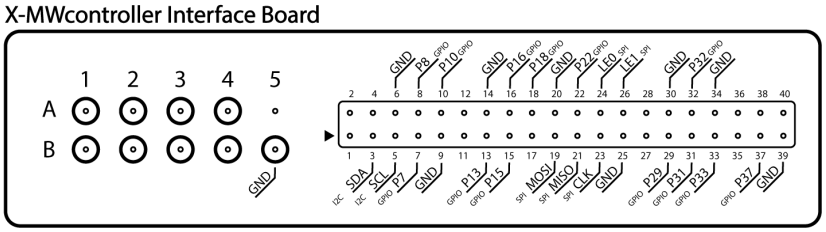
Top Side of Plate

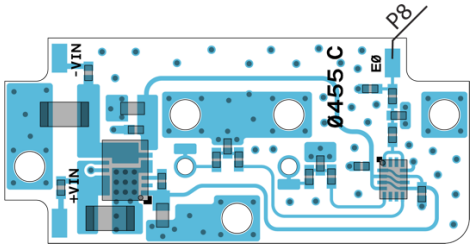


Bottom Side of Plate

Parallel Interface Basics

The parallel digital pins are labeled on the back side of the interface board with the annotation P followed by the pin number (with the exception of P1 which is labeled for reference only on some PCBs.) In the example image control board XM-A6K1-0409D, PCB #455, which is used to control the multiple PE42524 switches on the XM-A3N9-0409D RF block, PCB #220. The switches are controlled with a single digital line to the enable pin E0. On the bottom side of the prototyping plate solder a wire lead to E0 on the bias controller. Solder wires for +VIN and -VIN and connect each to a feedthrough capacitor. Connect the ribbon cable and power everything up. Now you're ready to control the part with the RaspberryPi software.

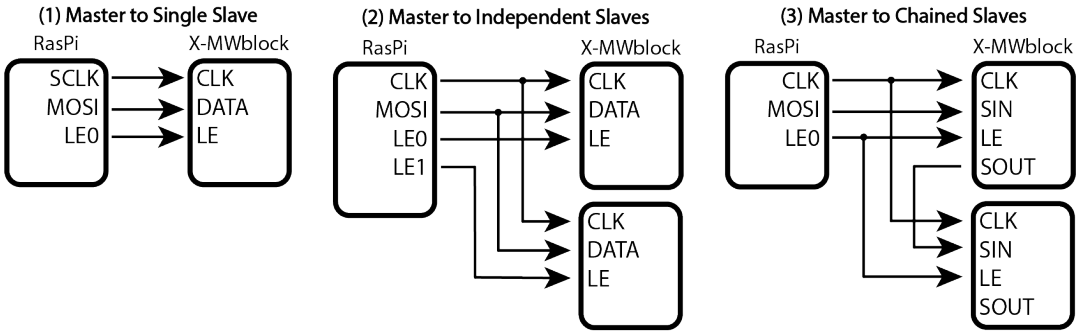




Serial Interface (SPI) Basics

Serial Peripheral Interface (SPI) bus is a 4 wire multi-node, bi-directional communication interface. The RaspberryPi is the master communicating with multiple X-MWblock slave devices. Multiple configurations are supported allowing a single master to control multiple devices with minimal wiring.

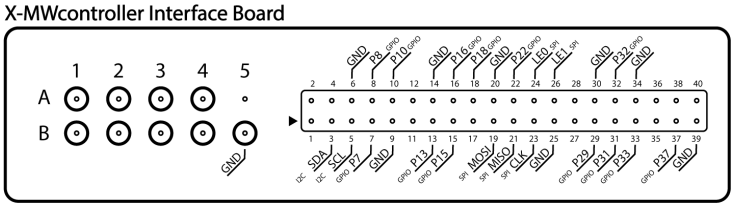
Most applications can be achieved with only 3 wires: clock (CLK), master out - slave in (MOSI), and line enable (LE). Three common configurations outlined below: Configuration (1) supports a single slave. Configuration (2) supports two slaves. Configuration (3) supports up to 7 slave devices but may not be supported by all X-MWblocks.

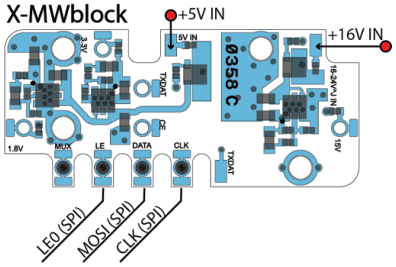


Making the Connections

The prototyping plate interface board connects to the RaspberryPi through a standard 40 pin ribbon cable. Feedtru capacitor holes are typically used for power with a clip-on terminal for ground. The SPI pins are labeled on the back side of the interface board. In the example image control board XM-A746-0409D, (board #358) which is used to provide bias and control to PLL devices such as the ADF4169, shown to the top and to the left. Note that the photo to the left also includes two 0404 voltage regulators (VREG) in addition to bias control board #358.

A RED wire supplies +5 volts to the first VREG board and short RED jumper wires are used to supply +5 V to the second VREG and #358 . +16 Volts is supplied through the yellow wire to board #385. WHITE wires connect MOSI on the RasPi interface to DATA and connect the two CLK pins. The BLUE wire connects the two LE pins. Yellow tape is used to secure the wires and reduce strain on the solder joints. Connect the ribbon cable and power everything up. Now you're ready to control the part with the RaspberryPi software.





Important Things to Remember

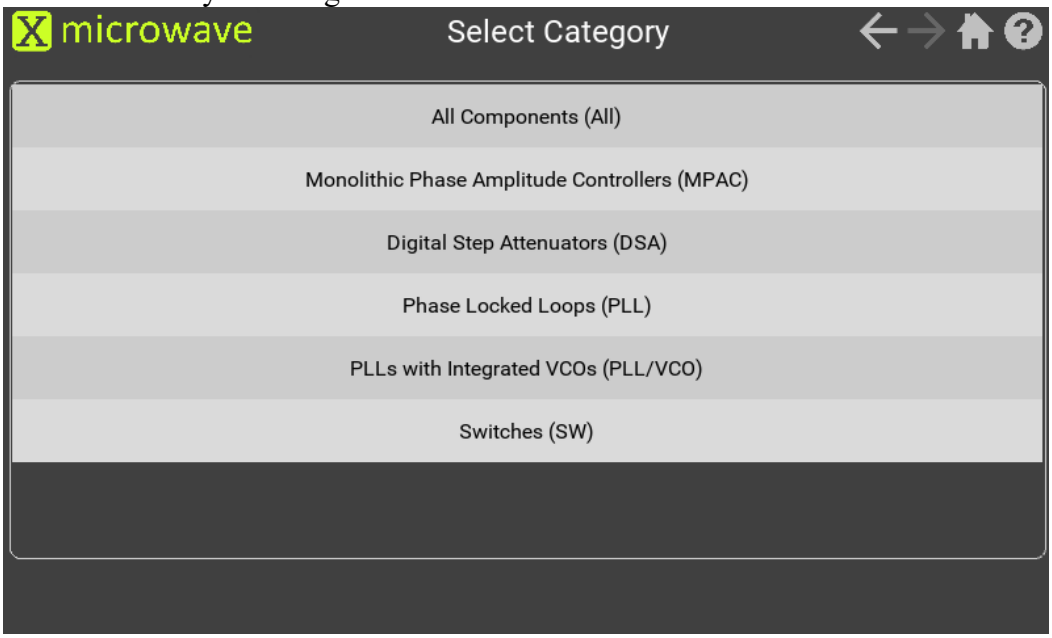
Make sure the ribbon cable is plugged in correctly. A stripe on the cable typically indicates pin 1. The settings in the part are likely to be lost each time you cycle the power so you will need to re-write each time.

The RaspberryPi is a computer, running an OS. Graceful shutdown will reduce your chances of corrupting the system.

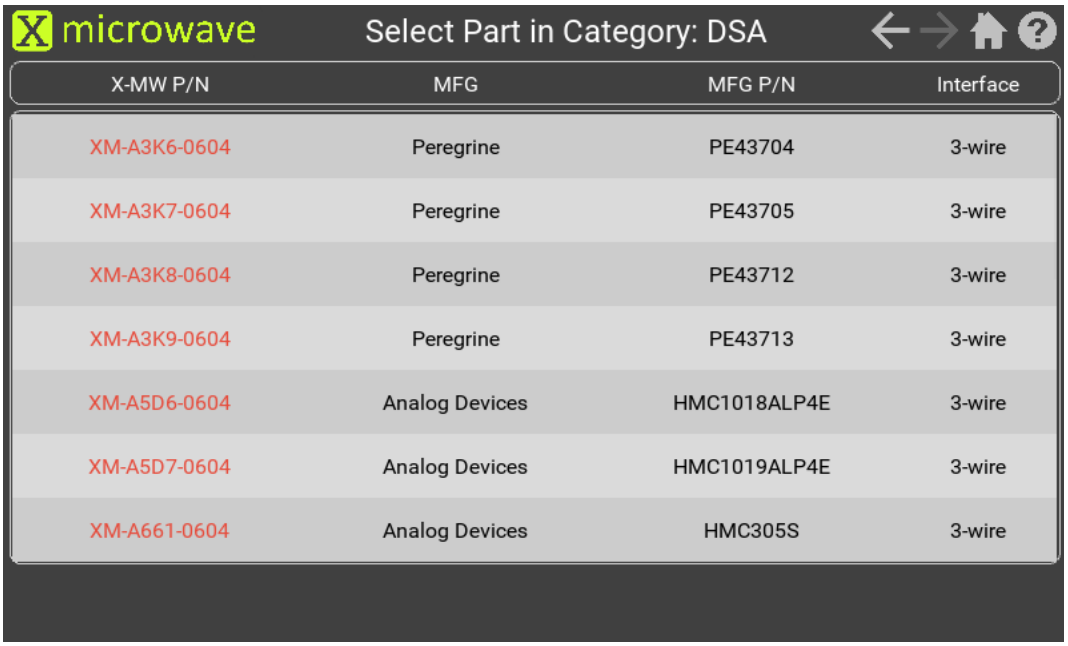
2. Using the RaspberryPi Software

Start by powering up the RaspberryPi by connecting the power cable. After boot-up you'll see the following Home screen. Once configured, this screen will display a list of all the parts in your system.

Add parts to the list by selecting ‘Add Part’.

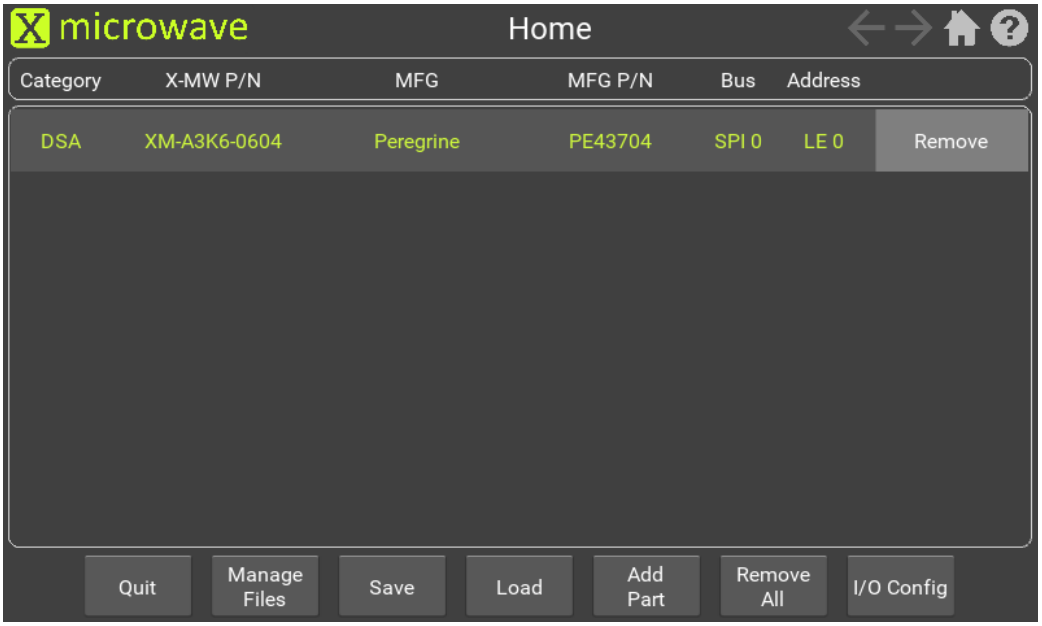


Select the ‘Category’ for the part to ‘Add’. Select ‘DSA’ in this example.

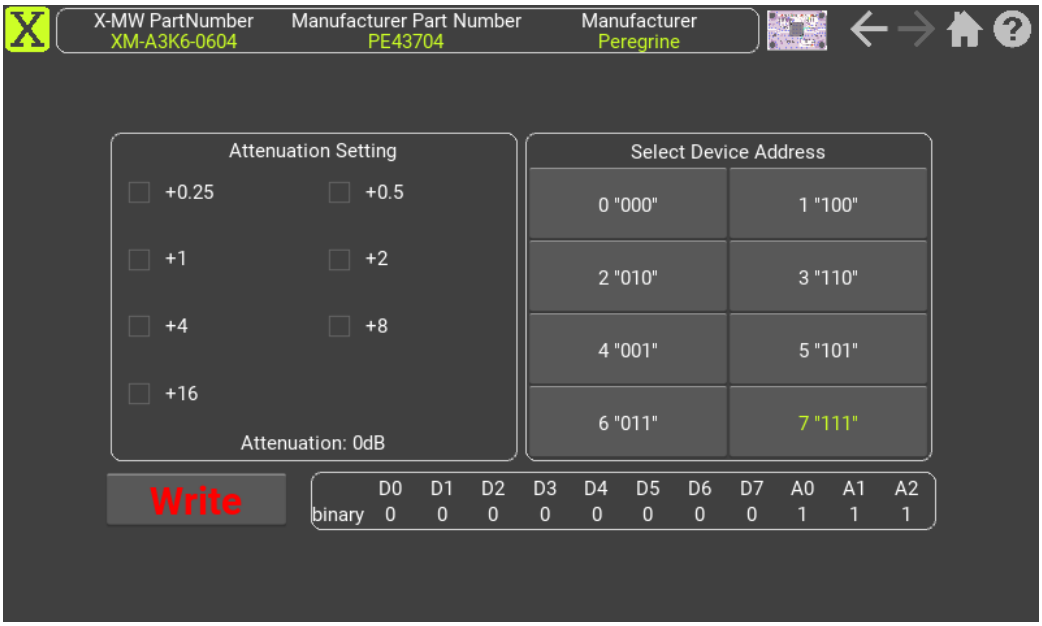


Note: ***3-wire*** indicates serial (SPI) device. ***parallel*** indicates direct digital IO control.

Choose the Peregrine PE43704 Digital Step Attenuator. Swipe down if you don't see it in the list. Selecting it will add it to the Home screen.



Select the part to configure it.



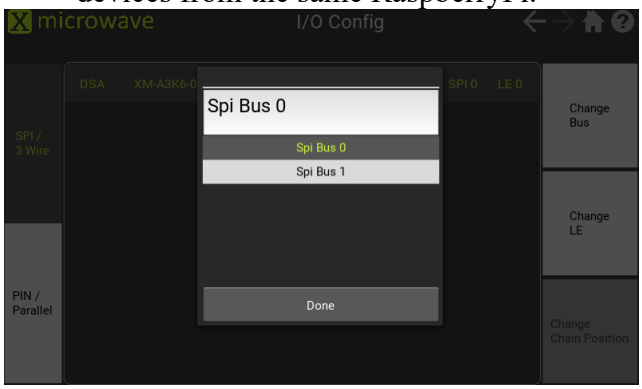
From here you can configure the part, in this case modifying the attenuation level.
Note that the register map is shown at the bottom of the screen. The register map is only sent to the part when the 'Write' button is selected by the user.

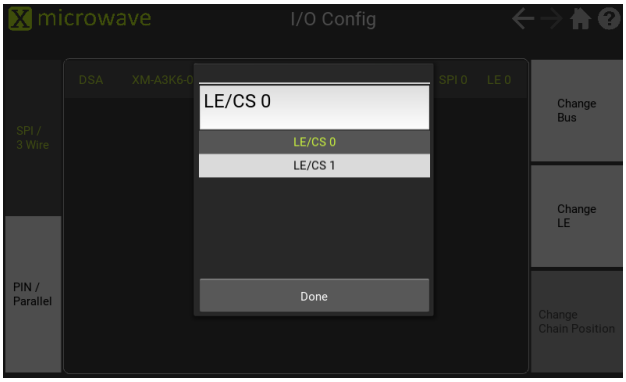
3. Part I/O Pin Configuration

Start by returning to the home menu.

Select ***I/O Config*** on the home screen and choose ***SPI / 3Wire*** on the left-hand side.

Change the Bus using ***Change Bus*** and line enable ***Change LE*** if needed to support multiple devices from the same RaspberryPi.





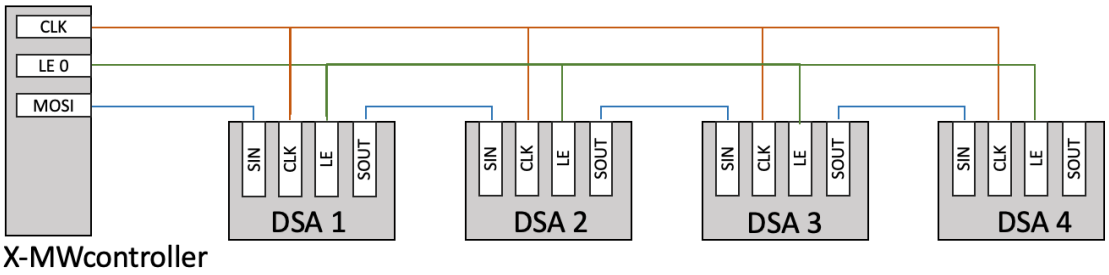
4. Appendix

4.1 Appendix I: Programming Multiple Digital Step Attenuators with a Single SPI Bus

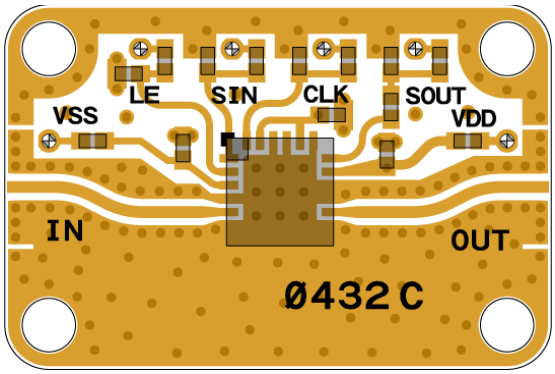
Digital step attenuators (DSA) often employ one of two approaches to allow multiple devices to share a single SPI Bus. The first method is by chaining multiple devices together in series and the second is through hard-wired addressing of each device.

Chained Devices

When chaining multiple devices in series, CLK and LE are shared among all of the devices in the chain. The first device in the chain receives SIN (Serial In) from the master controller while the each following DSA receives its SIN data from the SOUT (Serial Out) pin from the device before it. The chain of DSA form a shift register. As data is clocked into the first device its 'old' data is written to the next device in the sequence. In order to program all 4 devices in the figure below, DSA 1 is programmed 4 consecutive times. The data intended for the last-device in the chain is written first. After the 4th write, each DSA will have the intended value.



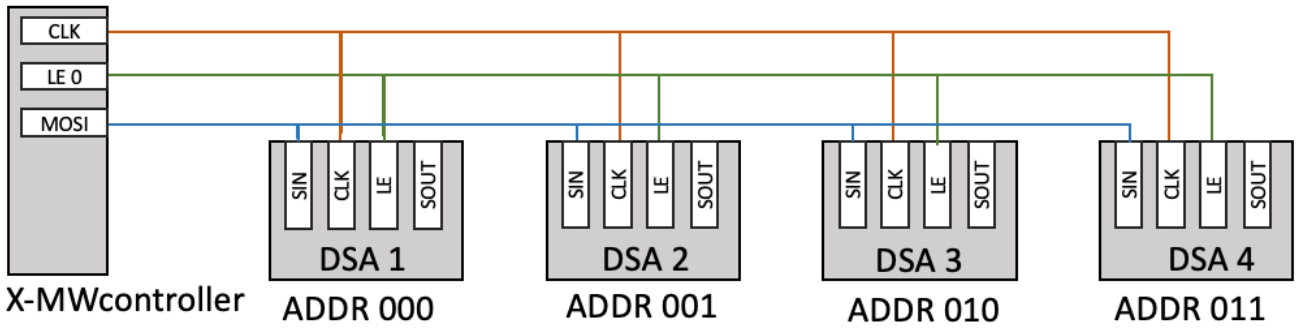
Visually, a DSA board that supports this type of programming looks like PCB 701. Notice that SIN and SOUT are each labeled.



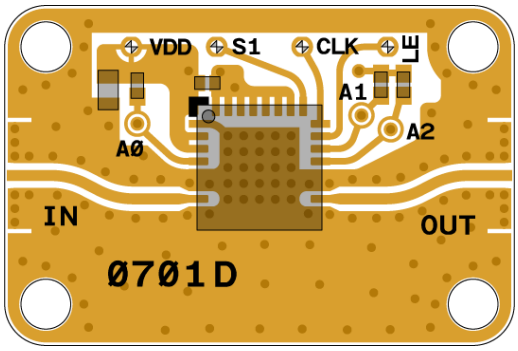
Uniquely Addressable Devices

In some cases, devices are individually addressable. In many cases these are labeled as A0, A1, and A2. Three address bits allows for 8 unique addresses 000 to 111. The serial data packet structure includes

both the address bits and data bits. While all DSAs are always listening, the value is only changed when the address bits match that of the corresponding DSA.



Visually, an addressable DSA X-MWblock often looks like PCB 432. Note that A0, A1, and A2 each include a pull-up resistor making the default address 111. Placing a 'solder dot' on the circle bridges the trace to the center ground VIA changing the value for that address bit to 0.



Visit XMicrowave.com

Learn more at our web site: xmicrowave.com.

