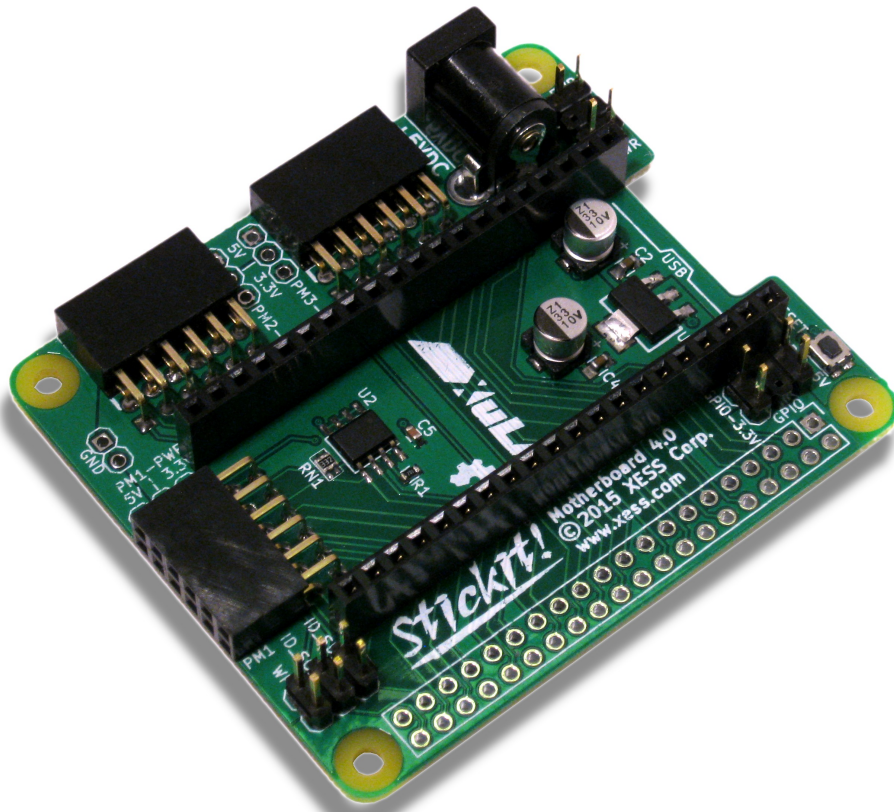


StickIt! Board Manual

*How to install and use
your new StickIt! Board*



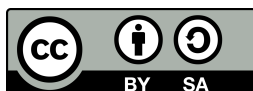
XESS is disclosing this Document and Intellectual Property (hereinafter “the Design”) to you for use in the development of designs to operate on, or interface with XESS hardware devices. XESS expressly disclaims any liability arising out of the application or use of the Design. XESS reserves the right to make changes, at any time, to the Design as deemed desirable in the sole discretion of XESS. XESS assumes no obligation to correct any errors contained herein or to advise you of any correction if such be made. XESS will not assume any liability for the accuracy or correctness of any engineering or technical support or assistance provided to you in connection with the Design.

THE DESIGN IS PROVIDED “AS IS” WITH ALL FAULTS, AND THE ENTIRE RISK AS TO ITS FUNCTION AND IMPLEMENTATION IS WITH YOU. YOU ACKNOWLEDGE AND AGREE THAT YOU HAVE NOT RELIED ON ANY ORAL OR WRITTEN INFORMATION OR ADVICE, WHETHER GIVEN BY XESS, OR ITS AGENTS OR EMPLOYEES. XESS MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DESIGN, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

IN NO EVENT WILL XESS BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOST DATA AND LOST PROFITS, ARISING FROM OR RELATING TO YOUR USE OF THE DESIGN, EVEN IF YOU HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE TOTAL CUMULATIVE LIABILITY OF XESS IN CONNECTION WITH YOUR USE OF THE DESIGN, WHETHER IN CONTRACT OR TORT OR OTHERWISE, WILL IN NO EVENT EXCEED THE AMOUNT OF FEES PAID BY YOU TO XESS HEREUNDER FOR USE OF THE DESIGN. YOU ACKNOWLEDGE THAT THE FEES, IF ANY, REFLECT THE ALLOCATION OF RISK SET FORTH IN THIS AGREEMENT AND THAT XESS WOULD NOT MAKE AVAILABLE THE DESIGN TO YOU WITHOUT THESE LIMITATIONS OF LIABILITY.

The Design is not designed or intended for use in the development of on-line control equipment in hazardous environments requiring fail-safe controls, such as in the operation of nuclear facilities, aircraft navigation or communications systems, air traffic control, life support, or weapons systems (“High-Risk Applications”). XESS specifically disclaims any express or implied warranties of fitness for such High-Risk Applications. You represent that use of the Design in such High-Risk Applications is fully at your risk.

©2012-2015 XESS, Inc. XESS, the XESS logo, and other designated brands included herein are trademarks of XESS Corporation. PMOD® is a trademark of Diligent Inc. All other trademarks are the property of their respective owners.



This document is licensed under the Attribution-ShareAlike 3.0 Unported license, available at <http://creativecommons.org/licenses/by-sa/3.0/>.

Document Revision History

Date	Version	Revision
01/15/2012	1.0	Initial release for StickIt! Board V1.0.
09/23/2012	2.0	Revised for StickIt! Board V2.0.
12/11/2012	3.0	Revised for StickIt! Board V3.0.
06/04/2015	4.0	Revised for StickIt! Board V4.0.

0 Contents

1 Preliminaries	1
Getting Help!	1
Take Notice!	1
Packing List	2
2 Setup	3
Inserting a XuLA Board into Your StickIt! Board	3
Applying Power to Your StickIt! Board	4
Applying Power Through the USB Port	4
Applying Power Through the Power Jack	5
Applying Power to Stand-Alone Applications	6
Applying Power from a Raspberry Pi	7
3 Connections	8
XuLA Board Socket	8
Power Circuitry	10
PMOD Sockets	11
Raspberry Pi GPIO Connector	14
HAT EEPROM	16
4 Using Modules	17
Using PMODs	17
The Hard Way	17
The Easy Way	19
Using the Raspberry Pi	20
The Hard Way	20
The Easy Way	21
Configuring the Raspberry Pi GPIO	21
Using Generic Circuitry	22
The Hard Way	22
The Easy Way	22
A I/O Locations	23
B Schematic	24

1 Preliminaries

Here's some helpful information before getting started.

Getting Help!

Here are some places to get help if you encounter problems:

- If you can't get the StickIt! Board to work, send an e-mail message describing your problem to help@xess.com.
- Or submit a problem report at www.xess.com/interact/contact/.
- Our [web site](#) also has
 - [example designs](#),
 - [application notes](#), and
 - [tutorials](#).

Take Notice!

- The XuLA Board is not 5V-tolerant. **Do not connect 5V logic signals to the PMOD sockets of the StickIt! Board.**
- **Only power the StickIt! Board with a regulated 5 VDC, center-positive power supply.**

Packing List

Here is what you should have received in your package:

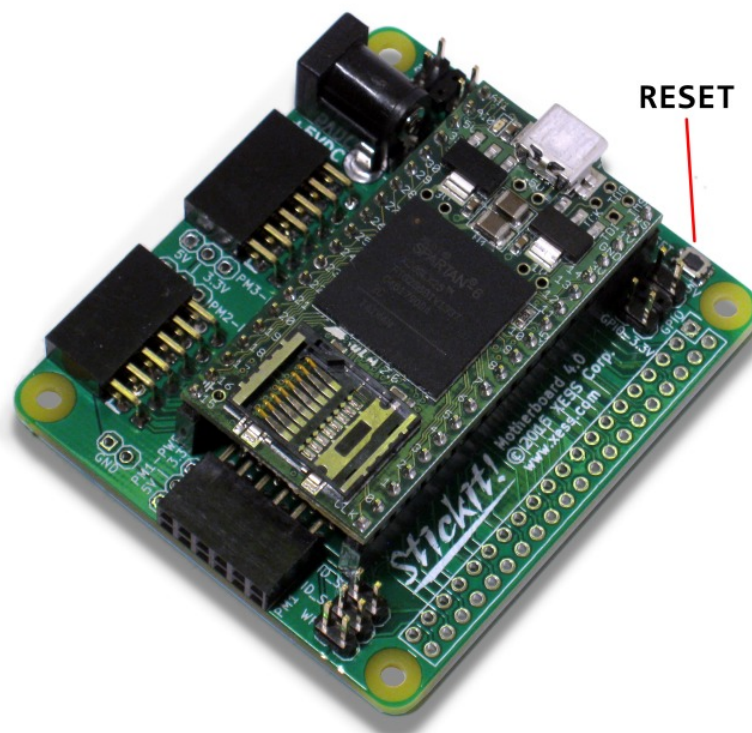
- a StickIt! Board.
- a 20×2 socket.
- a 20×2 header.

2 Setup

The StickIt! Board accepts a XuLA Board and provides it with connections to separate StickIt! modules such as LED displays, buttons, audio I/O, etc.

Inserting a XuLA Board into Your StickIt! Board

You insert a XuLA Board into the central row of sockets on the StickIt! Board as shown below. Make sure the XuLA Board prototyping header pins line-up with the sockets and that the USB connector is oriented towards the StickIt! Board RESET button.



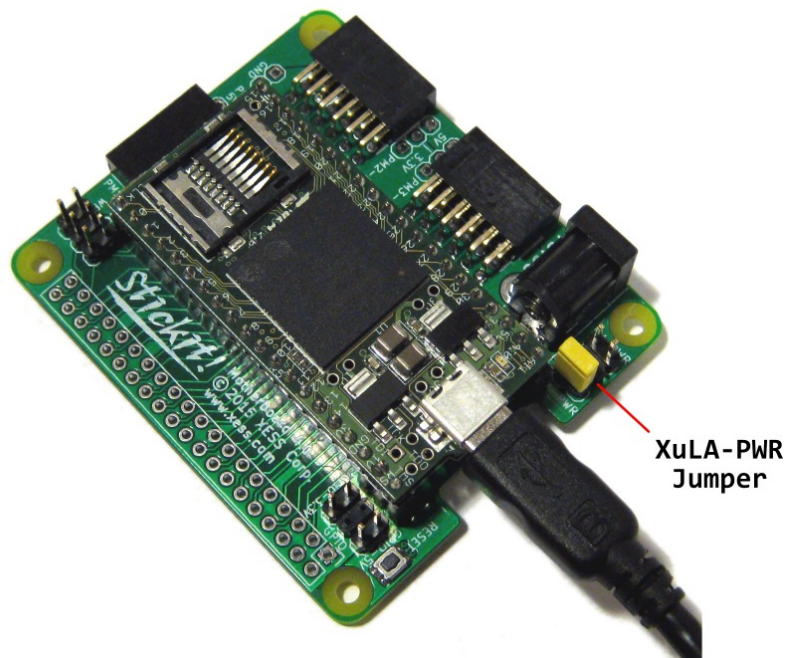
To insure a stable connection with the StickIt! Board, the XuLA Board should have 0.025" square posts soldered into its prototyping header. **The 0.019" round posts used for mounting a XuLA Board into a solderless breadboard should not be used or else intermittent connections will occur!**

Applying Power to Your StickIt! Board

There are several ways to power the combination of your XuLA Board and StickIt! Board. Each of these will be discussed below.

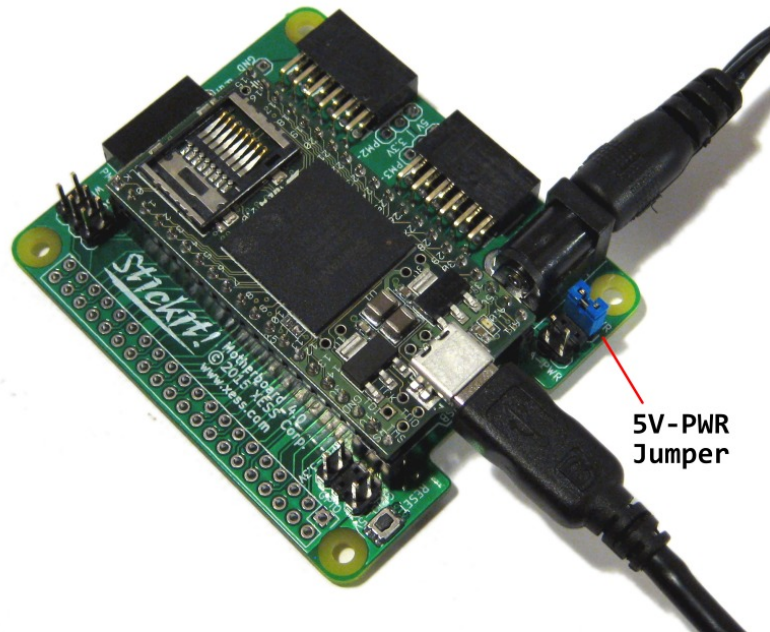
Applying Power Through the USB Port

Connecting the XuLA Board to a USB port provides it with a 5V supply capable of delivering up to 500 mA of current. The 5V supply can also power the StickIt! Board and any attached StickIt! modules by placing a shunt on the XULA-PWR jumper as shown below.



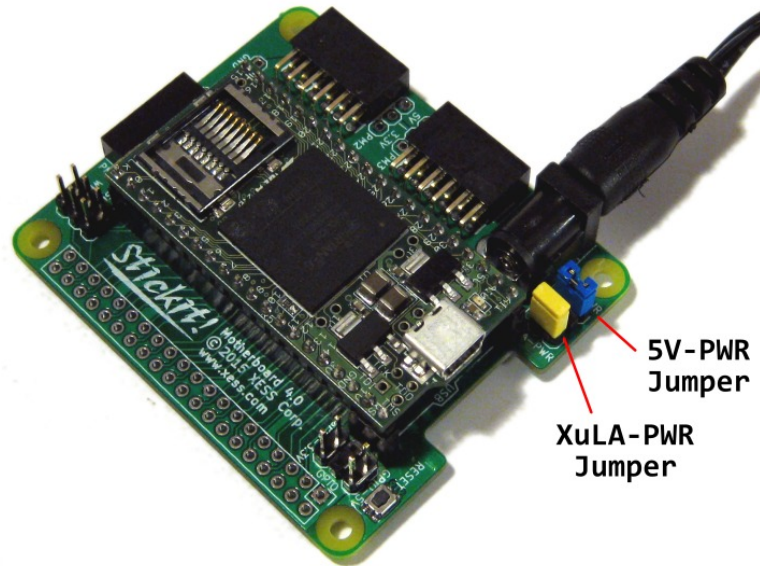
Applying Power Through the Power Jack

For applications that require more power than the USB port can provide, you can attach a regulated, 5 VDC center-positive power adapter to the 5VDC jack on the StickIt! Board. In this case, the shunt should be removed from the XULA-PWR jumper and placed on the 5V-PWR jumper as shown below. In this configuration, the USB port powers the XuLA Board while the adapter powers the StickIt! Board and any attached StickIt! modules.



Applying Power to Stand-Alone Applications

For applications that operate stand-alone with no connection to a USB port, you can attach a regulated, 5 VDC center-positive power adapter to the 5VDC jack on the StickIt! Board. In addition, shunts should be placed on both the XULA-PWR and 5V-PWR jumpers as shown below. This will transfer power from the adapter through the StickIt! Board to the XuLA Board.

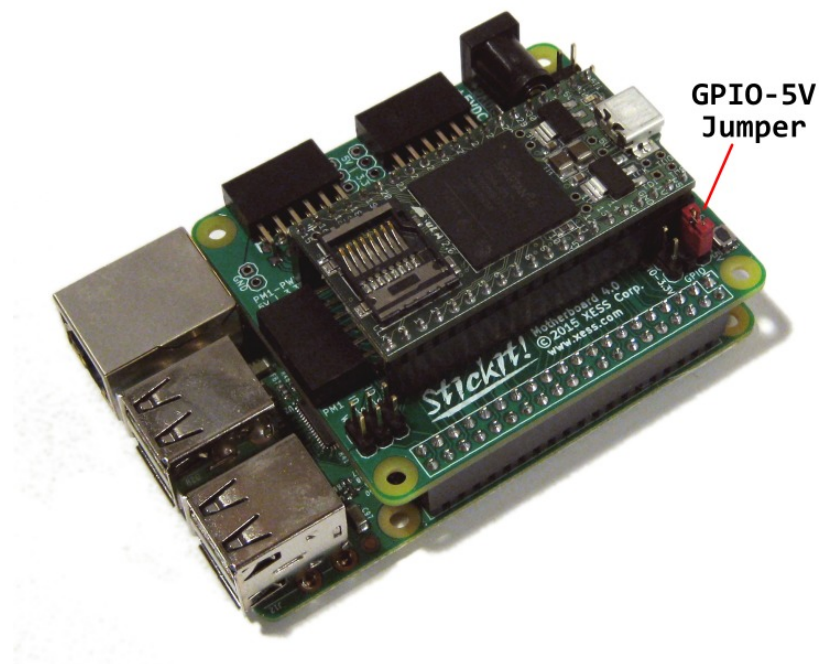


A USB cable should not be attached to the XuLA Board in this configuration or damage may result from the direct connection of the power adapter to the 5V supply pin of the USB port!

In order to prevent inadvertent damage, you can remove the shunt on 5V jumper of the XuLA Board to disconnect the USB 5V supply as shown below. Note that doing this will prevent you from powering the XuLA Board through the USB port until the 5V shunt is restored. **By default, the 5V jumper is closed by a wire trace on the underside of the XuLA Board! You will need to cut this trace to open the jumper.**

Applying Power from a Raspberry Pi

The StickIt! Board can get power through the GPIO connector of a Raspberry Pi by placing a shunt on the GPIO-5V jumper. This connects the 5V supply from the Raspberry Pi to the same power rail as the 5V power jack. Because of this, **a power adapter should not be connected to the 5V jack in this configuration or damage may result from the direct connection of the power adapter to the 5V supply of the Raspberry Pi!** A power adapter *can* be used to power the StickIt! Board when it is attached to a Raspberry Pi only if the shunt is removed from the GPIO-5V jumper. In addition, **a shunt should *never* be placed on the GPIO-3.3V jumper when the StickIt! Board is attached to the Raspberry Pi.**

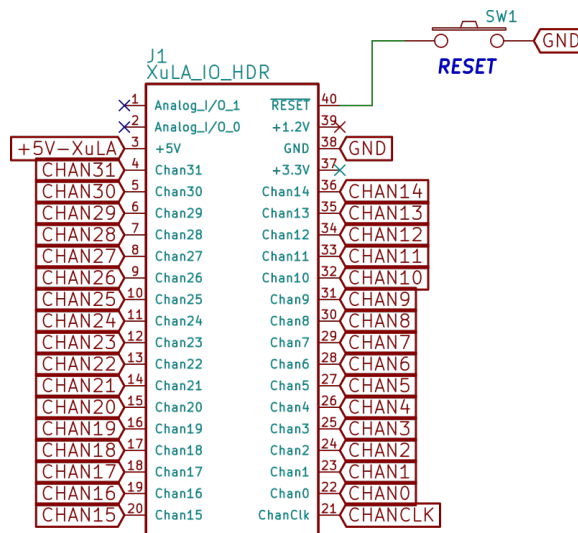


3 Connections

This chapter describes the various sections of the StickIt! Board and shows how the XuLA Board I/O connects to them. In addition to the partial schematics that follow, you can find a complete schematic at the end of this manual.

XuLA Board Socket

A XuLA Board connects to the StickIt! Board through the J1 socket.



Please note the following:

- The XuLA Board 5V pin is connected to the +5V-XULA signal on the StickIt! Board. This allows the XuLA Board to send power to or receive power from the StickIt! Board. This is the only voltage supply shared by both the XuLA Board and the StickIt! Board.

- The XuLA Board 3.3V and 1.2V pins do not connect to anything else on the StickIt! Board.
- The XuLA Board ground is connected to the StickIt! Board ground.
- The reset pin of the XuLA Board connects to the to the RESET button on the StickIt! Board. Pushing this button resets the XuLA Board.
- The analog I/O pins of the XuLA Board do not connect to anything else on the StickIt! Board.

Here are the FPGA pins connected to each channel for both the XuLA and XuLA2 Boards:

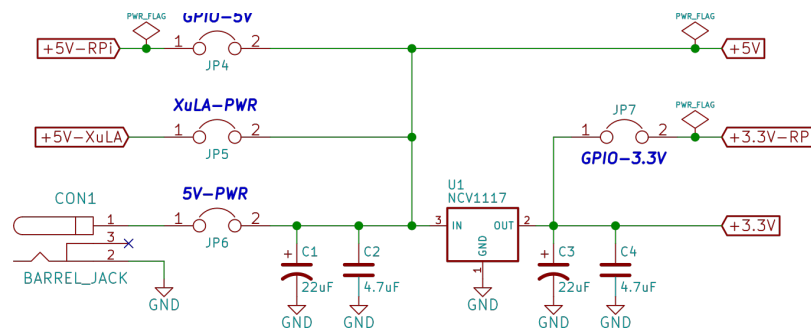
XuLA	XuLA2	J1	Pin#	J1	XuLA2	XuLA
AN1	AN1		1	21	RESET	RESET
AN0	AN0		2	22		+1.2V
+5V	+5V	+5V	3	23	GND	GND
P88	A2	CH31	4	24		+3.3V
P89	B2	CH30	5	25	CH14	B15
P93	B1	CH29	6	26	CH13	B16
P94	C1	CH28	7	27	CH12	C15
P97	E2	CH27	8	28	CH11	C16
P3	E1	CH26	9	29	CH10	F16
P4	F2	CH25	10	30	CH9	F15
P7	F1	CH24	11	31	CH8	J14
P12	H2	CH23	12	32	CH7	J16
P13	H1	CH22	13	33	CH6	K16
P19	J4	CH21	14	34	CH5	K15
P20	K3	CH20	15	35	CH4	M16
P21	M1	CH19	16	36	CH3	M15
P32	M2	CH18	17	37	CH2	R16
P33	R1	CH17	18	38	CH1	R15
P34	R2	CH16	19	39	CH0	R7
P35	T4	CH15	20	40	CHCLK	T7

Power Circuitry

The StickIt! Board accepts +5V from either

- a XuLA Board (when there is a shunt on the XULA-PWR jumper),
- an external power adapter (when there is a shunt on the 5V-PWR jumper), or
- a Raspberry Pi (when there is a shunt on the GPIO-5V jumper).

The StickIt! Board voltage regulator creates the +3.3V supply.

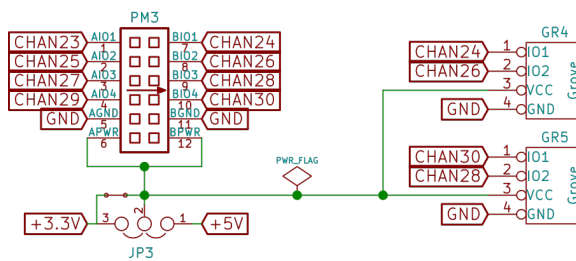
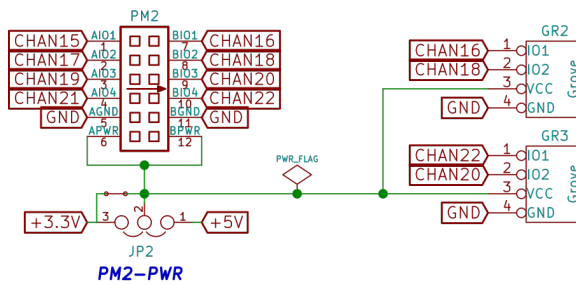
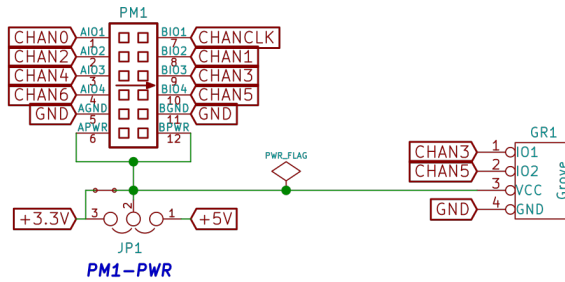


Be aware that placing shunts on both the XULA-PWR and 5V-PWR jumpers will connect the external power adapter to the XuLA Board power supply. **This will cause damage** unless 1) the USB cable is removed from the XuLA Board, or 2) the 5V jumper on the XuLA Board is open. (Note that in its factory-original configuration, the 5V jumper on the XuLA Board is shorted by a wiring trace on the bottom of the PCB which must be cut to disconnect the 5V supply from the USB port.)

Also be aware that placing shunts on both the XuLA-PWR and GPIO-5V jumpers when the StickIt! Board is attached to a Raspberry Pi will connect the external power adapter to the Raspberry Pi 5V supply. **This will cause damage!** There must *never* be a shunt on both the XuLA-PWR and GPIO-5V jumpers when the StickIt! Board is attached to a Raspberry Pi.

PMOD Sockets

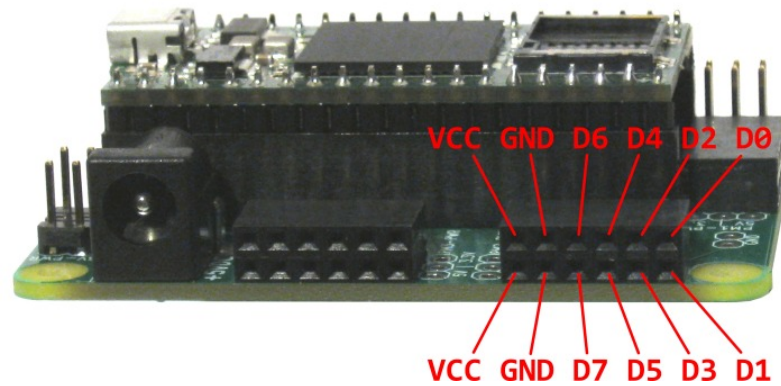
There are three sockets (PM1–PM3) for connecting external PMOD modules to the StickIt! Board. All of these sockets accept either four-bit or eight-bit PMOD modules. (There are also five sockets for connecting Grove boards, but these are covered by the PMOD sockets and can't be used. Just ignore them. Really, act like they're not even there.)



A total of 24 channels connect from the XuLA Board to the PMOD sockets, so you can use any combination of modules that require 24 total I/Os or less. The channel connections to each of the PMOD sockets are shown in the following table:

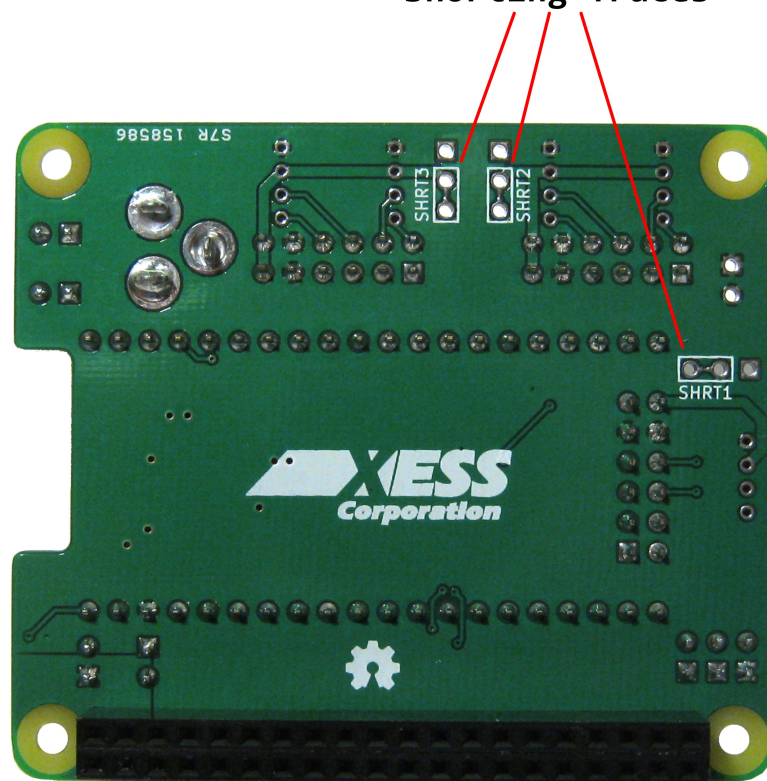
Chan	PMOD I/O	Pin#		PMOD I/O	Chan
PM1					
CH0	D0	1	7	D1	CHCLK
CH2	D2	2	8	D3	CH1
CH4	D4	3	9	D5	CH3
CH6	D6	4	10	D7	CH5
	GND	5	11	GND	
	VCC	6	12	VCC	
PM2					
CH15	D0	1	7	D1	CH16
CH17	D2	2	8	D3	CH18
CH19	D4	3	9	D5	CH20
CH21	D6	4	10	D7	CH22
	GND	5	11	GND	
	VCC	6	12	VCC	
PM3					
CH23	D0	1	7	D1	CH24
CH25	D2	2	8	D3	CH26
CH27	D4	3	9	D5	CH28
CH29	D6	4	10	D7	CH30
	GND	5	11	GND	
	VCC	6	12	VCC	

The physical arrangement of the PMOD I/O signals is shown below.



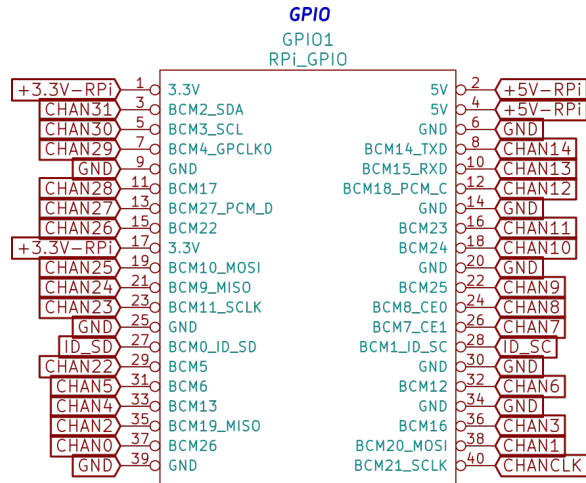
Each PMOD socket has an associated jumper to connect either the 3.3V or 5V power supply to the attached module. In their factory-original configuration, each of these jumpers is set to 3.3V by a shorting trace on the bottom of the StickIt! Board PCB. You must cut this trace and install a jumper and shunt if you want to use a 5V module.

Shorting Traces



Raspberry Pi GPIO Connector

You can solder a 20×2 socket to the GPIO port of the StickIt! Board so it can be connected to a Raspberry Pi. The GPIO port provides 26 general-purpose I/O pins (the rest are for power, ground, and access to the serial EEPROM on the StickIt! Board.)



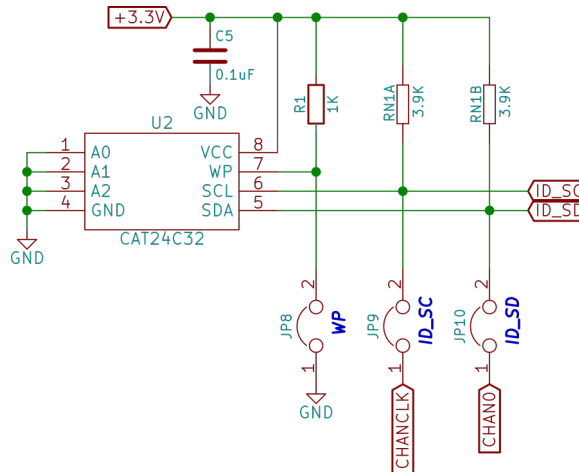
The channel connections to the GPIO are shown in the following table:

Chan	RPi I/O	Pin#		RPi I/O	Chan
	+3.3V	1	2	+5V	
CH31	BCM2_SDA	3	4	+5V	
CH30	BCM2_SCL	5	6	GND	
CH29	BCM4_GPCLK0	7	8	BCM14_TXD	CH14
	GND	9	10	BCM15_RXD	CH13
CH28	BCM17	11	12	BCM18_PCM_C	CH12
CH27	BCM27_PCM_D	13	14	GND	
CH26	BCM22	15	16	BCM23	CH11
	+3.3V	17	18	BCM24	CH10
CH25	BCM10_MOSI	19	20	GND	
CH24	BCM9_MISO	21	22	BCM25	CH9
CH23	BCM11_SCLK	23	24	BCM8_CE0	CH8
	GND	25	26	BCM7_CE1	CH7
	BCM0_ID_SD	27	28	BCM1_ID_SC	
CH22	BCM5	29	30	GND	
CH5	BCM6	31	32	BCM12	CH6
CH4	BCM13	33	34	GND	
CH2	BCM19_MISO	35	36	BCM16	CH3
CH0	BCM26	37	38	BCM20_MOSI	CH1
	GND	39	40	BCM21_SCLK	CHCLK

You can also use the GPIO port to connect the StickIt! Board to other pieces of external circuitry. When doing this, you can place shunts on the GPIO-5V and GPIO-3.3V jumpers to provide power to the external circuitry if needed. However, **never place a shunt on the GPIO-3.3V jumper if the external circuitry is generating its own 3.3V supply voltage!** The StickIt! Board is able to supply 3.3V to other circuitry, but it cannot accept it.

HAT EEPROM

In accordance with the [Raspberry Pi HAT \(Hardware Attached on Top\) requirements](#), the StickIt! Board has an EEPROM that stores information about the device and how it connects to the Raspberry Pi.



During normal operation, the EEPROM contents are read by the Raspberry Pi so that it can see what GPIO pins are used by the XuLA Board inserted in the StickIt! Board. In order to program the EEPROM with that information, you must place a shunt on the WP jumper to disable the write-protect function. The Raspberry Pi can then load the EEPROM through the ID_SC and ID_SD pins of the GPIO header. (Read more about this process on page [21](#).)

The EEPROM can also be loaded by the XuLA Board (if programmed appropriately) by placing shunts on the ID_SC and ID_SD jumpers as well as the WP jumper.

4

Using Modules

The StickIt! Board serves as a means of connecting a XuLA Board to various pieces of electronics, be they PMODs, a Raspberry Pi, or just some generic circuitry. The details of how this is done are presented below.

Using PMODs

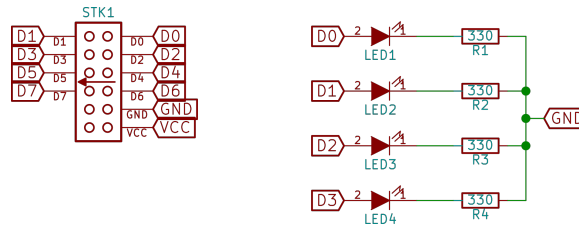
To use the functions of a particular PMOD, you have to determine which I/O signals of the module are connected to which pins of the FPGA. This is complicated by the fact the module could be plugged into any of the PMOD sockets. As with many things in life, there's a hard way to figure this out, and an easy way.

The Hard Way

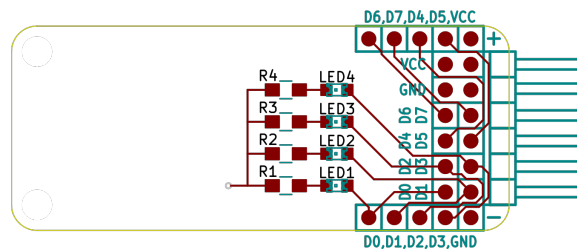
You can manually trace the connection of a PMOD's I/O signals to the FPGA pins using the following procedure:

- Select a PMOD socket to attach the module to.
- Use the [table](#) on page 12 to determine which module I/O signal terminates on each channel of the PMOD socket.
- Find which FPGA pin of the XuLA Board connects to each channel using the [table](#) on page 9.
- Make a UCF file associating each FPGA pin with each I/O of the module.
- Include the UCF file in your Xilinx ISE FPGA project.

As an example, consider a simple PMOD with four LEDs, each connected to one of the I/O signals like so:



The PCB wiring associated with this module looks like so:



From this, you can determine the following connections of the LEDs to the pins of the PMOD socket:

LED1	D0
LED2	D1
LED3	D2
LED4	D3

If this module is attached to PMOD socket PM3 on the StickIt! Board, then the channel connections are:

LED1	CH23	# PM3	D0
LED2	CH24	# PM3	D1
LED3	CH25	# PM3	D2
LED4	CH26	# PM3	D3

Now, assuming a XuLA2 Board is plugged into the StickIt! Board, the mapping of the channels to the FPGA pins is:

CH23	H2	# PM3	D0
CH24	F1	# PM3	D1
CH25	F2	# PM3	D2
CH26	E1	# PM3	D3

From this, you can store the following pin assignments in a UCF file:

```
NET LED1 LOC = H2; # LED1 -> PM3-D0 -> CH23 -> H2
NET LED2 LOC = F1; # LED2 -> PM3-D1 -> CH24 -> F1
NET LED3 LOC = F2; # LED3 -> PM3-D2 -> CH25 -> F2
NET LED4 LOC = E1; # LED4 -> PM3-D3 -> CH26 -> E1
```

Then include this UCF file in your ISE project.

The Easy Way

Tracing the paths a module's I/O signals take through a particular PMOD socket to a channel which then connects to an FPGA pin is a tedious, error-prone process. The `xsconnect` Python package (<https://pypi.python.org/pypi/xsconnect>) provides two scripts to make the process easier.

`xsconn` is the command-line script for generating pin assignments:

```
usage: xsconn.py [-h] [-v] [-p [PERIPHERALBOARD]] [-m [MOTHERBOARD]]
               [-d [DAUGHTERBOARD]] [-n [PORTNAME]] [-l]
```

optional arguments:

```
-h, --help            show this help message and exit
-v, --version        show program's version number and exit
-p [PERIPHERALBOARD], --peripheralboard [PERIPHERALBOARD]
-m [MOTHERBOARD], --motherboard [MOTHERBOARD]
-d [DAUGHTERBOARD], --daughterboard [DAUGHTERBOARD]
-n [PORTNAME], --portname [PORTNAME]
-l, --list
```

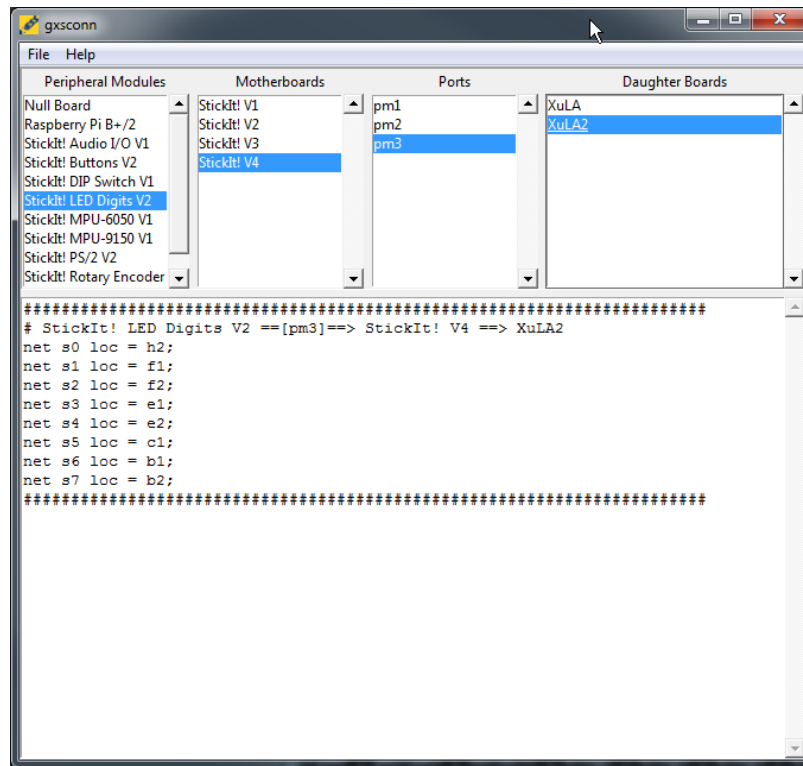
For example, if a StickIt! LEDDigits peripheral board is connected to the PM3 port of the StickIt! Board which in turn holds a XuLA2 FPGA board, then the command:

```
xsconn -p leddigits -m stickit4 -n pm3 -d xula2
```

will generate the output:

```
#####
# StickIt! LED Digits V2 ==[pm3]==> StickIt! V4 ==> XuLA2
net s0 loc = h2;
net s1 loc = f1;
net s2 loc = f2;
net s3 loc = e1;
net s4 loc = e2;
net s5 loc = c1;
net s6 loc = b1;
net s7 loc = b2;
#####
```

The `gxscnn` script does the same thing as `xscnn`, but with a GUI:



The output from `xscnn` and `gxscnn` is formatted for use as a UCF file in your ISE project.

Using the Raspberry Pi

Just as with the PMODs, connecting the StickIt! Board to a Raspberry Pi requires you to determine the connections of the I/O signals to the pins of the FPGA. And, once again, there's a hard way and an easy way to do that.

The Hard Way

Manually tracing the connections of the Raspberry Pi I/O signals to the pins of the FPGA is done as follows:

- Use the [table](#) on page 15 to determine which Raspberry Pi I/O signal terminates on each channel of the XuLA Board socket.
- Find which FPGA pin of the XuLA Board connects to each channel using the [table](#) on page 9.
- Make a UCF file associating each FPGA pin with each Raspberry Pi I/O.
- Include the UCF file in your Xilinx ISE FPGA project.

The Easy Way

Use `xscconn` or `gxscconn` just like with the PMODs. Set the Raspberry Pi as the peripheral board and select GPIO as the port and you'll get a complete list of the connections to the FPGA.

Configuring the Raspberry Pi GPIO

Most HATs perform a specific function that requires communicating with the Raspberry Pi through a fixed set of GPIO pins. The Raspberry Pi reads the list of pins from the EEPROM on the HAT and configures its I/O pins appropriately.

But the XuLA Board and StickIt! Board do not perform one fixed function: they can be programmed for many types of applications. Therefore, the list of GPIO pins used by a particular application has to be written into the EEPROM on the StickIt! Board. This is done as follows:

1. Create your FPGA application for the XuLA Board along with the required pin assignments for communicating through the GPIO connector.
2. Attach the StickIt! Board to the GPIO connector and place a shunt on the WP jumper to enable writing to the EEPROM. (A XuLA Board can be inserted into the J1 socket at this point, but it should not have a bitstream loaded into its flash memory that might cause the FPGA to drive the pins of the GPIO connector.)
3. Download the code from <https://github.com/raspberrypi/hats/tree/master/eeepromutils> onto your Raspberry Pi.
4. Edit the `eeeprom_settings.txt` file to reflect how the GPIO pins are used by your application.
5. Compile the `eeepmake.c` program and then run it to create a binary data file that stores your GPIO pin settings:

```
eeepmake eeeprom_settings.txt eeeprom_settings.bin
```

6. Execute the `eeepflash.sh` shell program to write the binary data file to the EEPROM:

```
eeepflash -w -f=eeeprom_settings.bin -t=-24c32
```

7. Remove the shunt from the WP jumper and reboot the Raspberry Pi.

After the above procedure is completed, you can load the XuLA Board with the bitstream for the application. You can change the application as long as you don't change how it drives the GPIO pins. (For example, don't take a GPIO pin that was previously driving the FPGA and reverse it so that the FPGA now drives the Raspberry Pi.) If you do change how the GPIO pins are used, you will need to update the StickIt! Board EEPROM again to reflect those changes before changing the FPGA bitstream.

Using Generic Circuitry

Instead of an Raspberry Pi, you can connect generic digital circuitry to the GPIO port. **The circuitry must interface using 3.3V logic levels. The XuLA Board will not tolerate 5V logic levels.**

The Hard Way

Use the following procedure to determine the connections of the I/O signals from your circuitry to the pins of the FPGA:

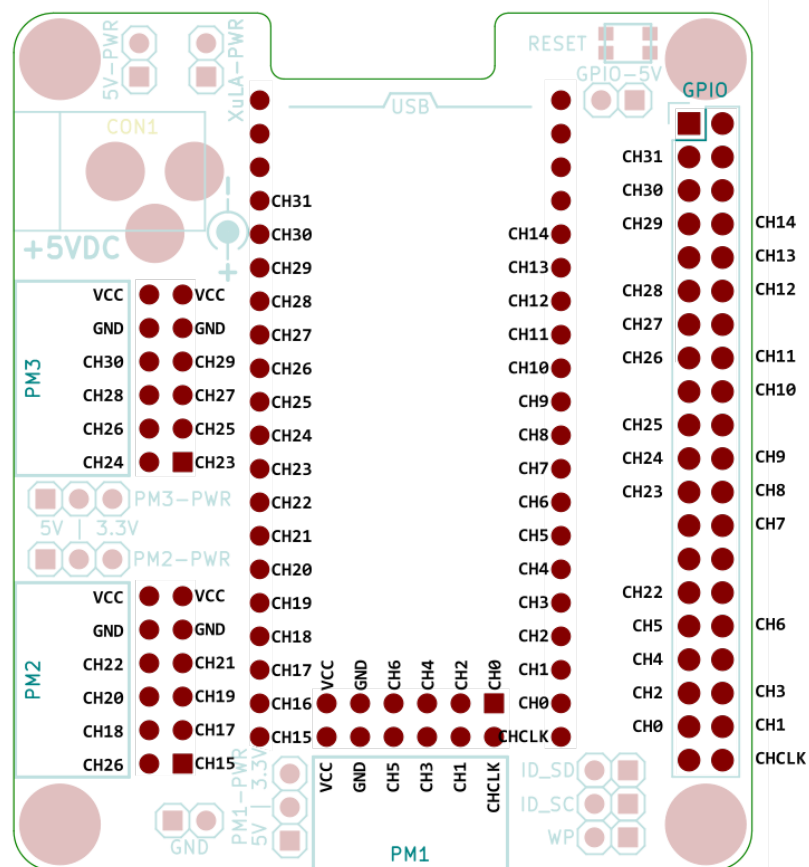
- Attach the I/O signals from the generic circuitry to pins 1–40 of the GPIO connector. There are 26 usable I/O pins; the rest are mostly power and ground pins that you can use to provide power to the external circuitry if desired.
- Use the [table](#) on page 15 to determine which GPIO pin connects to each channel of the XuLA Board socket.
- Find which FPGA pin of the XuLA Board connects to each channel using the [table](#) on page 9.
- Make a UCF file associating each FPGA pin with an I/O signal.
- Include the UCF file in your Xilinx ISE FPGA project.

The Easy Way

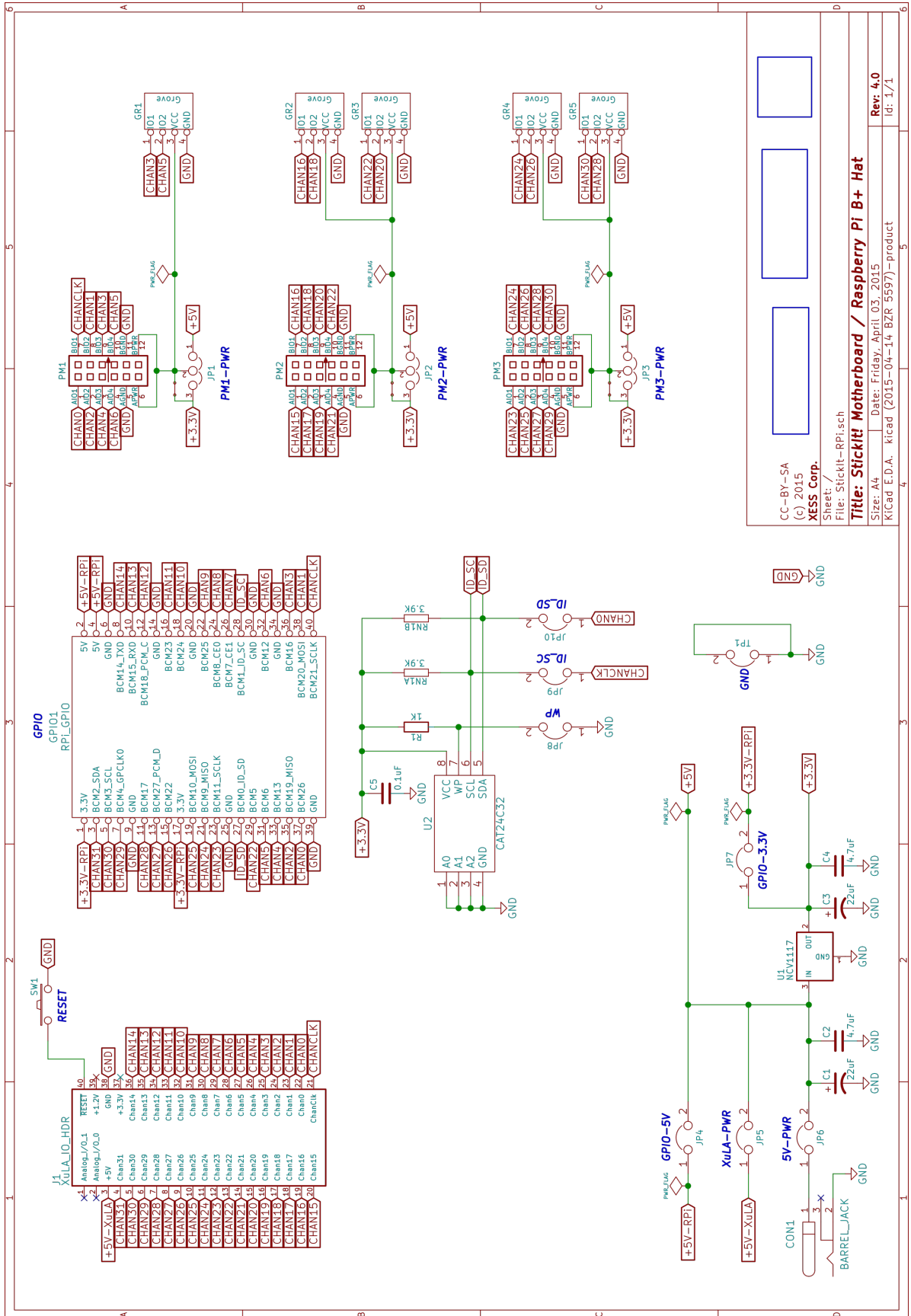
For generic circuitry, `xscconn` or `gxscconn` can still make the pin-tracing process a little easier. Just set the peripheral board to be Generic and select GPIO as the port. Then you'll get a complete list of connections between GPIO pins 1–40 and the FPGA. You will have to manually substitute your I/O signal names for the pin numbers in the list to create the final UCF file for your ISE project.

A I/O Locations

The connections of the XuLA Board I/O channels to the PMOD and Raspberry Pi sockets of the StickIt! Board are shown below.



B***Schematic***



CC-BY-SA
(c) 2015
XESS Corp.

Sheet: /
File: StickIt-RPi.sch

Title: StickIt! Motherboard / Raspberry Pi B+ Hat

Size: A4 Date: Friday, April 03, 2015
KICad E.D.A. kicad (2015-04-14 BZR-5597)-product

Rev: 4.0
Id: 1/1