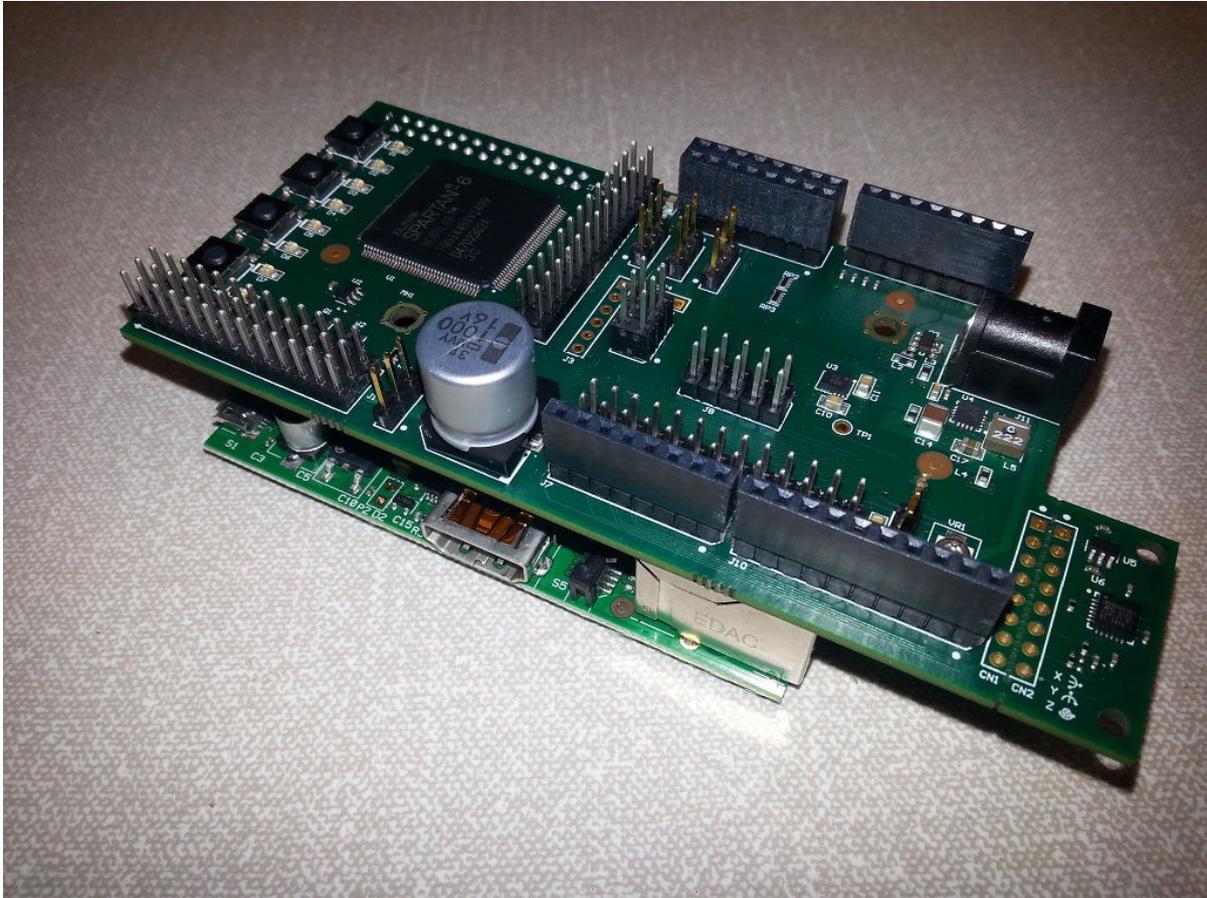


## Application Note AN-002

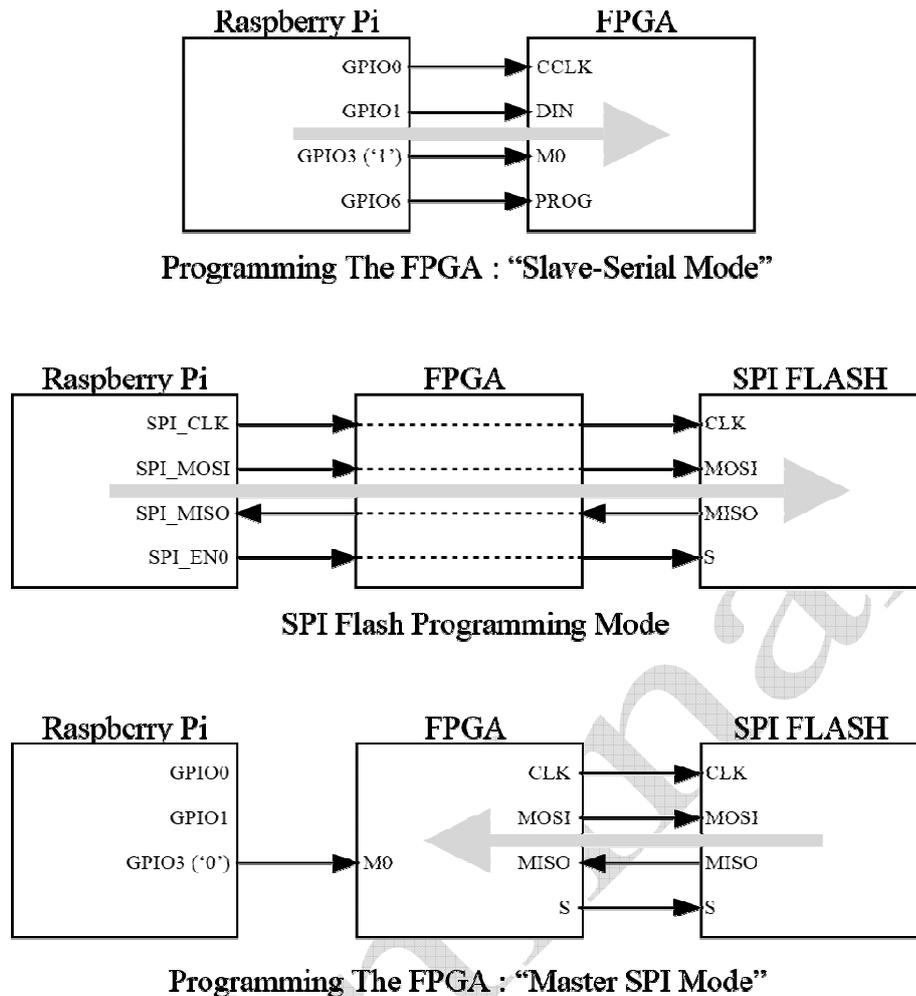
### Programming the FPGA on the Pixi<sup>2.0</sup>



### Summary

The Pixi<sup>2.0</sup> is designed to expand the general-purpose I/O capabilities of the Raspberry Pi and provide a low cost means of introducing the user to the world of digital electronics and FPGA technology as well as giving the 'Pi Enthusiast' a few more features to play with. The low product cost and feature-packed specification of the Pixi<sup>2.0</sup> makes it ideal for applications in computing, hobby-electronics, education, training and product development.

This application note explains how to use the Raspberry Pi to program the FPGA or the SPI flash on the Pixi<sup>2.0</sup>, either for upgrading the FPGA design or downloading your own FPGA designs. The FPGA configuration memory is volatile so it loses its configuration when the power is removed. The SPI flash is non-volatile and can retain the FPGAs configuration data to rapidly re-program the FPGA fully autonomously on power-up, even without the Raspberry Pi connected. Programming the FPGA or SPI flash is quick and simple using the PiXi-Tools software provided with the Pixi<sup>2.0</sup>.



**Figure 1 Configuring the FPGA using the Raspberry Pi**

Figure 1 describes some of the programming capabilities available to the user including two methods of configuring the FPGA and one method to store the configuration file in non-volatile SPI flash memory. The FPGA programming mode is controlled using the mode pins (M0 & M1) on the FPGA. Only M0 is made available to the user on the PiXi<sup>2.0</sup> since the other modes available by changing M1 are not supported on the PiXi<sup>2.0</sup>. M0 allows the user to select between 'Slave-Serial' mode and 'Master SPI' mode. In Slave-Serial mode the Raspberry Pi can program the FPGA directly using a serial interface that includes a clock 'CCLK', data 'DIN' and two programming controls 'PROGRAM' and 'INIT' (not shown), all driven from GPIO on the Raspberry Pi. In Master-SPI mode the FPGA reads the non-volatile configuration data from the SPI Flash on power-up. The SPI flash can be programmed using the Raspberry Pi but this is only possible by pre-loading the FPGA with a configuration that allows the Raspberry Pi's SPI interface to access and program the SPI flash using the FPGA in a SPI port 'forwarding' mode where the FPGA simply connects the SPI interface of the Raspberry Pi to the slave interface on the flash memory.

The FPGA and SPI flash can also be programmed using a Xilinx Impact™ software tools together with a Digilent USB JTAG cable or Xilinx programming hardware. The 6-pin header needed to connect the Digilent or Xilinx programming tools to the PiXi<sup>2.0</sup> is not fitted

by default or supplied with the PiXi<sup>2.0</sup> but may be purchased easily and soldered to the board with a reasonable amount of skill and care.

## Setting up the PiXi<sup>2.0</sup> for Programming on Power-Up

The PiXi<sup>2.0</sup> can automatically configure the FPGA from the SPI flash using the “Master SPI” mode on power-up if the "M0" pin of the FPGA is low ('0') at power-up. Link header JP2 can be used to enable this function. To select Master SPI mode JP2 should be configured to link pin 2 to pin 3.

The SPI Flash is only fitted to the PiXi<sup>2.0</sup> - STD so this method of programming does not work on the PiXi<sup>2.0</sup> - LITE. JP2 is not fitted on the LITE version.

## Setting up the PiXi<sup>2.0</sup> for Programming at Boot-Up

There may be some cases where it is desirable to have the Raspberry Pi configure the FPGA at boot-up using “Slave Serial” mode rather than using the SPI flash to configure the FPGA at power-up. In this case the "M0" pin on the FPGA must be held high during the power-up sequence. To select Slave Serial mode, JP2 should be configured with the link header completely removed.

This is the only mode available to program the FPGA on the PiXi<sup>2.0</sup> - LITE.

## Running the PiXi<sup>2.0</sup> Stand-Alone

The PiXi<sup>2.0</sup> - STD can be run stand-alone by using the SPI flash to program the FPGA on power-up.

There are no modifications needed to enable this, as long as there is an FPGA configuration successfully loaded into SPI flash and JP2 is configured for Master SPI linking pin 2 to pin 3, then the PiXi<sup>2.0</sup> - STD will automatically configure the FPGA at power-up, even if it isn't connected to the Raspberry Pi.

## Install PiXi-Tools

PiXi-Tools provides a convenient set of applications and libraries for accessing the PiXi functions and FPGA registers over SPI & I2C. It also includes software for programming the FPGA or SPI Flash. PiXi-Tools must be installed on the Raspberry Pi connected to the PiXi<sup>2.0</sup> before the FPGA can be updated. For more information on installing and using PiXi-Tools, please refer to application note AN-020.

## Download the latest FPGA Configuration File

The latest FPGA version can be downloaded from the Astro Designs website. Please go to [http://www.astro-designs.com/downloads\\_pixi.html](http://www.astro-designs.com/downloads_pixi.html) to download the latest and earlier versions of the FPGA configuration files that have been released.

## Programming the FPGA

The FPGA needs a binary data sequence ('bit-stream') or configuration file loaded into it before it can work. There are a number of ways that FPGAs can be configured but these are not all within the scope of this application note. This section will focus on how to program the FPGA using the Raspberry Pi and the PiXi<sup>2.0</sup> FPGA programming software included within PiXi-Tools. The PiXi-Tools software is supplied as both pre-compiled application and as raw source code, should the user wish to use the programming routines in their own code. PiXi-Tools also includes the default FPGA configuration file, the configuration file only needs to be downloaded if the version supplied with PiXi-Tools has been superseded with a new release or if the user is experimenting with their own design of FPGA.

### Check the FPGA Version

Before programming the FPGA, it's best to check the current version to determine if it needs updating or not. This also gives the user a means of verifying that it has been updated once the process is completed by comparing the version before and after the programming process. PiXi-Tools includes a function to check the current version. The following command can be run from the command line from any folder once PiXi-Tools has been installed.

```
pio fpga-version
```

The version number that this routine generates is a date-time code that identifies when the FPGA configuration file was generated. It consists of three 16-bit numbers that can be read from three register location within the FPGA and displayed in hexadecimal notation. PiXi-Tools does this for you. The 12 digit hexadecimal value describes a date-time code in the format:

```
0xhhmm, 0xssDD, 0xMMYY
```

Or, all-together in 48-bit hexadecimal format as returned by "pio fpga-version":

```
(0x)hhmmssDDMMYY
```

Where,

hh represents hour of the day,  
mm represents minutes,  
ss represents seconds,  
DD represents day of the month,  
MM represents month,  
YY represents the year,

It's important to note that the value must be displayed in hexadecimal notation in order to read the date-time code correctly.

The PiXi-Tools software can also display this in a more readable date / time format by using the following command:

### *pio fpga-build-time*

#### **Using the Raspberry Pi to Program the FPGA at Boot**

If the Raspberry Pi is being used to program the FPGA during the boot sequence, all that is needed is to replace the current FPGA configuration file on the Raspberry Pi with the latest version. This is typically the case for the 'Lite' model of the PiXi<sup>2.0</sup>.

To select this mode of programming, the programming mode jumper (JP2) must not be fitted.

Once the FPGA configuration file has been downloaded, copy the file to the following location:

```
/usr/share/pixi-tools/fpga/pixi_2vx
```

The file should be renamed to `pixi.bin`, replacing the existing file.

Simply rebooting the Raspberry Pi will result in the FPGA being re-configured with the new configuration towards the end of the Raspberry Pi's boot sequence.

Note: For this mode to work, PiXi-Tools must have installed the boot-time programming script.

#### **Using the Raspberry Pi to Program the FPGA at Any Time**

The PiXi-Tools software can be used to reprogram the FPGA instantly at any time if this is desired. This is particularly useful if the user is developing their own FPGA design and wishes to try several versions of the design as their FPGA design evolves.

To select this mode of programming, the programming mode jumper (JP2) must not be fitted.

First, copy the new configuration file to a convenient location such as `/home/pi`. Secondly, remove link header JP2 (if fitted, taking note of which two pins it bridges) Then, from the location containing the new configuration file, run the following command to use PiXi-Tools to write the configuration file direct to the FPGA.

```
pio load-fpga pixi.bin
```

This programs the FPGA immediately, if all is well, the DONE LED (D9) should go off for about two seconds before coming back on again indicating that the FPGA has been successfully been reconfigured. Note that in this case it's the FPGA which is being configured directly and as such the configuration will be lost if the power is removed. For permanent change to the configuration, either program the SPI flash (see below) or, if the configuration is loaded by the FPGA on boot, copy the FPGA configuration file to the location as described in the process above.

## Using the Raspberry Pi to Program the SPI Flash

The PiXi<sup>2.0</sup> - STD includes an SPI flash non-volatile memory that can be used to rapidly configure the FPGA on power-up, long before the Raspberry Pi has completed its boot sequence. The PiXi<sup>2.0</sup> - STD is supplied with this non-volatile pre-configured with an FPGA configuration. The Raspberry Pi can be used to replace the default FPGA configuration within the SPI flash using the PiXi-Tools software.

Note: Versions of the FPGA configuration with a build time / version earlier than 26<sup>th</sup> November, 2014 do not support programming the flash through the Raspberry Pi. If the FPGA configuration has a build time / version before the 26<sup>th</sup> of November, 2014, the recovery process described later in this section must be followed.

### Programming the PiXi<sup>2.0</sup> SPI flash (Method 1)

(Procedure required when running FPGA versions dated between 26<sup>th</sup> November 2014 to 16<sup>th</sup> February 2015)

- 1) Check JP2 (FPGA Programming Mode Jumper) is set correctly. It should be set to link pin2 to pin 3. Pin 1 is the pin marked with a dot on the circuit board, so the link should be positioned at the opposite end of the three pins of JP2 to where the dot is found on the circuit board.
- 2) Connect the PiXi<sup>2.0</sup> to the Raspberry Pi.
- 3) Power-up the Raspberry Pi. Note that it's advised to use a permanent power source and not a battery, just to ensure that the power cannot fail while programming the Flash.
- 4) Check that the PiXi-Tools software is up to date:

```
apt-get update pixi-tools
```

- 5) Install the gpio configuration script (download from [www.astro-designs.com](http://www.astro-designs.com)) to enable SPI flash programming on the PiXi<sup>2.0</sup>. It is recommended that this be placed in the /home/pi folder.
- 6) Install the new FPGA configuration file, renaming it to "pixi.bin" if necessary, onto the Raspberry Pi, placing it in the PiXi-Tools FPGA installation folder:

```
/usr/share/pixi-tools/fpga/pixi_2vx
```

- 7) Use PiXi-Tools to check the FPGA version:

```
pio fpga-version (or pio fpga-build-time)
```

If the FPGA version is dated before the 16<sup>th</sup> February, 2015 then continue with this process. If the FPGA version is dated on or after the 16<sup>th</sup> February 2015 then this process will not work, in which case one of the other two processes described in this section must be used.

- 8) Use the gpio configuration script to configure the PiXi<sup>2.0</sup> to direct SPI communication from the Raspberry Pi to the PiXi<sup>2.0</sup> SPI Flash:

```
/home/pi/pi-gpio-out.py 1 0 0
```

Note: If this is successful then calls to `pio fpga-version` should now fail.

9) Navigate to the PiXi-Tools FPGA installation folder:

```
cd /usr/share/pixi-tools/fpga/pixi_2vx
```

10) Run the `pio SPI flash` programming function to program the flash with the new configuration file:

```
pio flash-write 0 pixi.bin
```

11) Use the `gpio` configuration script to configure the PiXi2.0 to direct SPI communication from the Raspberry Pi to the PiXi2.0 SPI Register Map:

```
/home/pi/pi-gpio-out.py 0 0 1
```

12) Use PiXi-Tools to check the FPGA version:

```
pio fpga-version (or pio fpga-build-time)
```

Note: The version returned should be the same as recorded earlier in this process, the new version will only start working after power-cycling the PiXi<sup>2.0</sup>

13) Safely power-down the Raspberry Pi, turning the power off and back on again to load the new FPGA from flash.

14) Check the FPGA version, verify that the FPGA has been updated, and returns the correct date-code for the new FPGA:

```
pio fpga-version (or pio fpga-build-time)
```

If the date code of the FPGA matches the listed date-code for the FPGA file then the FPGA SPI flash has been successfully updated and the FPGA update process is complete.

### Programming the PiXi<sup>2.0</sup> SPI flash (Method 2)

(Procedure required when running FPGA versions dated on or after the 16<sup>th</sup> February 2015)

- 1) Check JP2 (FPGA Programming Mode Jumper) is set correctly. It should be set to link pin2 to pin 3. Pin 1 is the pin marked with a dot on the circuit board, so the link should be positioned at the opposite end of the three pins of JP2 to where the dot is found on the circuit board.
- 2) Connect the PiXi<sup>2.0</sup> to the Raspberry Pi.
- 3) Power-up the Raspberry Pi. Note that it's advised to use a permanent power source and not a battery, just to ensure that the power cannot fail while programming the Flash.
- 4) Check that the PiXi-Tools software is up to date:

*apt-get update pixi-tools*

- 5) Install the gpio configuration script (download from [www.astro-designs.com](http://www.astro-designs.com)) to enable SPI flash programming on the PiXi<sup>2.0</sup>. It is recommended that this be placed in the /home/pi folder.
- 6) Install the new FPGA configuration file, renaming it to "pixi.bin" if necessary, onto the Raspberry Pi, placing it in the PiXi-Tools FPGA installation folder:

*/usr/share/pixi-tools/fpga/pixi\_2vx*

- 7) Use PiXi-Tools to check the FPGA version:

*pio fpga-version (or pio fpga-build-time)*

If the FPGA version is dated before the 16<sup>th</sup> February, 2015 then continue with this process. If the FPGA version is dated on or after the 16<sup>th</sup> February 2015 then this process will not work, in which case one of the other two processes described in this section must be used.

- 8) Use the gpio configuration script to configure the PiXi<sup>2.0</sup> to direct SPI communication from the Raspberry Pi to the PiXi<sup>2.0</sup> SPI Flash:

*/home/pi/pi-gpio-out.py 1 0 0*

Note: If this is successful then calls to *pio fpga-version* should now fail.

- 9) Navigate to the PiXi-Tools FPGA installation folder:

*cd /usr/share/pixi-tools/fpga/pixi\_2vx*

- 10) Run the *pio* SPI flash programming function to program the flash with the new configuration file:

*pio flash-write 0 pixi.bin*

- 11) Use the gpio configuration script to configure the PiXi2.0 to direct SPI communication from the Raspberry Pi to the PiXi2.0 SPI Register Map:

*/home/pi/pi-gpio-out.py 0 0 1*

- 12) Use PiXi-Tools to check the FPGA version:

*pio fpga-version (or pio fpga-build-time)*

Note: The version returned should be the same as recorded earlier in this process, the new version will only start working after power-cycling the PiXi<sup>2.0</sup>

- 13) Safely power-down the Raspberry Pi, turning the power off and back on again to load the new FPGA from flash.

14) Check the FPGA version, verify that the FPGA has been updated, and returns the correct date-code for the new FPGA:

*pio fpga-version (or pio fpga-build-time)*

If the date code of the FPGA matches the listed date-code for the FPGA file then the FPGA SPI flash has been successfully updated and the FPGA update process is complete.

## Further Reading

If you want to learn more about the programming process, Xilinx application note XAPP502 describes the basics of using a micro computer such as the Raspberry Pi to program a Xilinx FPGA. The datasheet for the FPGA also contains more detailed information about the different programming modes.

PiXi-Tools is described in more detail in application note AN-020 "Installing PiXi-Tools on the Raspberry Pi".

The full register map for the PiXi can be found in application note AN-025 "PiXi SPI & I2C Register Map".

All of these documents and other documents are available for download from [www.astro-designs.com](http://www.astro-designs.com).

## Acknowledgements

“Raspberry Pi” is a trademark of the Raspberry Pi Foundation.

“Xilinx”, “Spartan”, “ISE”, “Impact” & “WebPack” are all registered trademarks of Xilinx.

Preliminary