

# PiXi (2.0) Quick Start

## Setting up

1

Connect the PiXi Directly To the Raspberry Pi

2

Connect a Power Supply

Note - do not use the USB power Connector on the Raspberry Pi

6.5V - 15V  
DC Power (2.1mm)



Raspberry Pi  
Connecting Socket

GPIO  
3.3V I/O

Analogue /  
Arduino™ Shield Socket

RS232  
Serial

Configuration

Switches  
& LEDs

FPGA

Arduino™  
Shield Socket

GPIO2  
5V or 3.3V I/O

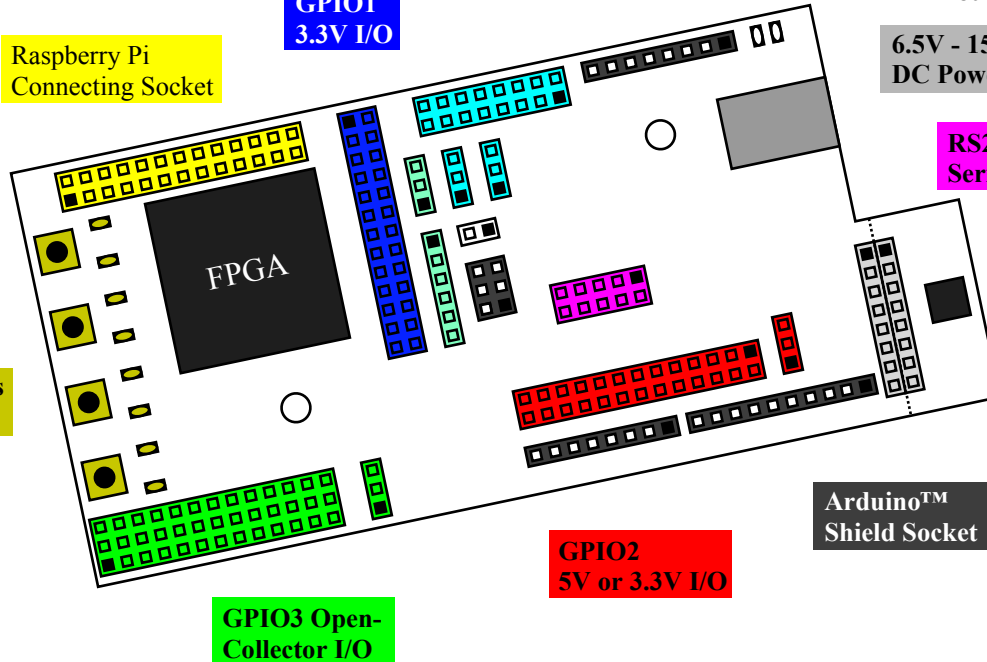
GPIO3 Open-  
Collector I/O

3

Install PiXi-Tools  
apt-get install pixi-tools  
(See over for more info)

4

Download Application Notes  
and User Guides from  
<http://www.astro-designs.com>



# PiXi (2.0) Quick Start

## Installing PiXi-Tools

### Manual Installation

- 1 Create a temporary installation folder:  

```
mkdir -p ~/pixi-setup  
cd ~/pixi-setup
```
- 2 Add the Astro Designs repository to apt-get:  

```
rm -f archive.key  
wget http://zaphod.astro-designs.net/raspbian/archive.key  
apt-key add archive.key  
rm -f astro-designs.list  
wget http://zaphod.astro-designs.net/raspbian/astro-designs.list  
cat astro-designs.list  
cp astro-designs.list /etc/apt/sources.list.d
```
- 3 Install (or update) PiXi-Tools:  

```
apt-get update  
apt-get install pixi-tools
```
- 4 Configure default FPGA image:  

```
mkdir /etc/pixi-tools  
ln -s /usr/share/pixi-tools/fpga/pixi_2vx/pixi.bin /etc/pixi-tools
```
- 5 Compile an example:  

```
pixi-cc /usr/share/doc/pixi-tools/examples/*.c -o pixi-example  
./pixi-example
```

### Automatic Installation

- 1 Download the automated installation script from [www.astro-designs.com/downloads/pixi/pixi-setup.sh](http://www.astro-designs.com/downloads/pixi/pixi-setup.sh) and ensure the script is installed into a temporary folder on the Raspberry Pi.

Convert the file to Unix/Linux  
`dos2unix pixi-setup.sh`

Ensure the script is executable:  
`chmod +x ./pixi-setup.sh`

Run the script:  
`./pixi-setup.sh`

Continue to step (4) of the Manual Installation

# PiXi (2.0) Quick Start

## A Few Quick Demonstrations

1

LEDs

Set LED mode:

```
pio spi-set 0x37 0 (Simple register mode)
```

Turn on LEDs

```
pio spi-set 0x36 0 (all off)
```

```
pio spi-set 0x36 1 (LED 1 slow flash)
```

```
pio spi-set 0x36 2 (LED 1 quick flash)
```

```
pio spi-set 0x36 3 (LED 1 on)
```

```
pio spi-set 0x36 4 (LED 2 slow flash)
```

```
pio spi-set 0x36 255 (all on)
```

2

Switches

Read switches:

```
pio spi-get 0x39
```

3

Temperature

Read temperature from MPU:

```
pio mpu-read-temp
```

4

ADC (Input range is 0V- 5V)

```
pio adc-read 0 (returns ADC channel 0 input)
```

```
pio adc-read 1 (returns ADC channel 1 input)
```

5

DAC (Output range defaults to 0V to 3.3V)

```
pio dac-write 0 0 (sets DAC channel 0 to 0V)
```

```
pio dac-write 0 3095 (sets DAC channel 0 to 3.3V)
```

6

GPIO1

```
pio spi-set 0x28 0x0000 (sets GPIO1[3:0] to inputs)
```

```
pio spi-set 0x29 0x0000 (sets GPIO1[7:4] to inputs)
```

...

```
pio spi-set 0x2D 0x0000 (sets GPIO1[23:20] to inputs)
```

```
pio spi-get 0x20 (returns GPIO1[7:0] status)
```

```
pio spi-get 0x21 (returns GPIO1[15:8] status)
```

```
pio spi-get 0x22 (returns GPIO1[23:16] status)
```

```
pio spi-set 0x28 0x1111 (sets GPIO1[3:0] to outputs)
```

```
pio spi-set 0x29 0x1111 (sets GPIO1[7:4] to outputs)
```

...

```
pio spi-set 0x2D 0x1111 (sets GPIO1[23:20] to outputs)
```

```
pio spi-set 0x20 [0-255] (sets GPIO1[7:0])
```

```
pio spi-set 0x21 [0-255] (sets GPIO1[15:8])
```

```
pio spi-set 0x22 [0-255] (sets GPIO1[23:16])
```

7

GPIO2

```
pio spi-set 0x2e 0x0000 (sets GPIO2[3:0] to outputs)
```

```
pio spi-set 0x2f 0x0000 (sets GPIO2[7:4] to outputs)
```

```
pio spi-set 0x23 [0-255] (sets GPIO2[7:0])
```

```
pio spi-set 0x2e [0-255] (sets GPIO2[3:0] to PWM)
```

```
pio spi-set 0x2f [0-255] (sets GPIO2[7:4] to PWM)
```

```
pio spi-set 0x40 512 (sets GPIO2[0] / PWM[0] to 50%)
```

```
pio spi-set 0x41 1023 (sets GPIO2[1] / PWM[1] to 100%)
```

# PiXi (2.0) Quick Start

## A Few Quick Demonstrations

8

### GPIO3

First - use JP6 to configure GPIO3 for 3.3V or 5V

```
pio spi-set 0x32 0x0000 (sets GPIO3[3:0] to outputs)
```

```
pio spi-set 0x33 0x0000 (sets GPIO3[7:4] to outputs)
```

```
pio spi-set 0x23 [0-255] (sets GPIO3[7:0])
```

```
pio spi-set 0x32 [0-255] (sets GPIO3[3:0] to PWM)
```

```
pio spi-set 0x33 [0-255] (sets GPIO3[7:4] to PWM)
```

```
pio spi-set 0x50 512 (sets GPIO3[0] / PWM[0] to 50%)
```

```
pio spi-set 0x51 1023 (sets GPIO3[1] / PWM[1] to 100%)
```

9

### Accelerometer...

Read accelerometer status:

```
pio read-mpu-acc-x
```

```
pio read-mpu-acc-y
```

```
pio read-mpu-acc-z
```

10

### Gyroscope...

Read gyroscope status:

```
pio read-mpu-gyro-x
```

```
pio read-mpu-gyro-y
```

```
pio read-mpu-gyro-z
```

11

### Magnetometer...

Read magnetometer status:

```
pio read-mpu-mag-x
```

```
pio read-mpu-mag-y
```

```
pio read-mpu-mag-z
```

12

### Application Notes

Download application notes for more information

[www.astro-designs.com/pixi-200\\_appnotes.html](http://www.astro-designs.com/pixi-200_appnotes.html)

AN-001: Beginners Guide to the PiXi (2.0)

AN-002: Programming the FPGA on the PiXi (2.0)

AN-003: FPGA Development on the PiXi (2.0)

AN-004: Connecting an LCD or Vacuum

Fluorescent Display to the PiXi (2.0)

AN-005: Using the GPIO on the PiXi (2.0)

AN-006: Using the Serial Interface on the PiXi (2.0)

AN-007: Using the Analogue Interface on the PiXi (2.0)

AN-008: Using the PiXi (2.0) in Education

AN-009: Using the Motion Sensors on the PiXi (2.0)

AN-011: Using the PiXi (2.0) to control a Hobby Servo

AN-012: Adding an external I2C device to the PiXi (2.0)

AN-018: Using the LEDs & Switches on the PiXi (2.0)

AN-021: Connecting an Arduino™ Shield to the PiXi(2.0)

AN-022: Connecting a Matrix Keypad to the PiXi (2.0)

We'll be adding more application notes soon...