

2021F-114-gpio NANO Board

Harry Li, Ph.D.

References on GPIO Interface

Pi and NANO are pin to pin compatible

[Jetson Nano GPIO - JetsonHacks](https://www.jetsonhacks.com)<https://www.jetsonhacks.com> > ... > [GPIO/I2C](#)

Jun 7, 2019 — As you may have heard, the GPIO pin layout on the Jetson Nano is compatible with the 40 pin layout of a Raspberry Pi (RPi).

```
/* blink.c
```

```
*
```

```
* Raspberry Pi GPIO example using sysfs interface.
```

```
* Guillermo A. Amaral B. <g@maral.me>
```

```
*
```

```
* This file blinks GPIO 4 (P1-07) while reading GPIO 24 (P1_18).
```

```
*/
```

```
//https://elinux.org/RPi_GPIO_Code_Samples#sysfs
```



[Main Page](#)

[Community portal](#)

[Current events](#)

[Recent changes](#)

[Help](#)

[Volunteering](#)

[Bug Tracker](#)

Where else to find us

[Twitter \(@elinux\)](#)

[#elinux on Libera.Chat](#)

[Facebook \(@elinux.org\)](#)

[Mailing Lists](#)

Page [Discussion](#)

RPi GPIO Code Samples

The [Raspberry Pi GPIOs](#) can be controlled using

Contents [t

1 [C](#)

1.1 [Direct register access](#)

1.2 [WiringPi](#)

1.3 [sysfs](#)

1.4 [bcm2835 library](#)

1.5 [pigpio](#)

1.6 [lgpio \(local /dev/gpiochip I/F\)](#)

1.7 [rgpio \(local & remote /dev/gpiochip I/F\)](#)



GPIO

<https://developer.nvidia.com/embedded/dlc/tegra-x1-technical-reference-manual>

Pads marked "CZ" can be configured to be 3.3V tolerant and driving; and pads marked "DD" can be 3.3V tolerant when in open-drain mode (only.)

Table 22: MPIO Pad Types

Chapter 9, pp. 277

Pad Type	I/O Rail Voltage (V)	Input Buffer	Output Buffer	I/O Voltage Tolerance	Nominal Pull Strength	"Slew Rate" Control
ST	1.8	Schmitt & CMOS	push-pull	VDDIO	100 kΩ	No
CZ	1.8, 3.3	Schmitt & CMOS	push-pull	VDDIO	18 kΩ (pull-up and pull-down)	2 bits, up & d
DD	1.8	Schmitt & CMOS	push-pull & open-drain	3.3V for open-drain, VDDIO otherwise	100 kΩ	No
LV_CZ	1.2, 1.8	Schmitt & CMOS	push-pull	VDDIO	15 kΩ	2 bits, up & d
ST_EMMC	1.2, 1.8	Schmitt & CMOS	push-pull	VDDIO		3 bits, down c
DP_AUX	1.8	Diff/Open Drain	push-pull & open-drain	3.3V	NA	NA

9.13.1 GPIO_CNF_0

Configuration register
Chapter 9, pp. 250

Designates whether each pin operates as a GPIO or as an SFIO programmed to GPIO mode at any stage.

Lock bits are used to control the access to the CNF and OE register can be programmed ONLY during Boot and get reset by chip res

This is an array of 4 identical register entries; the register fields b

GPIO Port A - D Configuration Registers

Offset: 0h..fh | Read/Write: R/W | Reset: 0b0000000000000000

NVIDIA Jetson Nano J41 Header Pinout

<https://www.jetsonhacks.com/nvidia-jetson-nano-j41-header-pinout/>

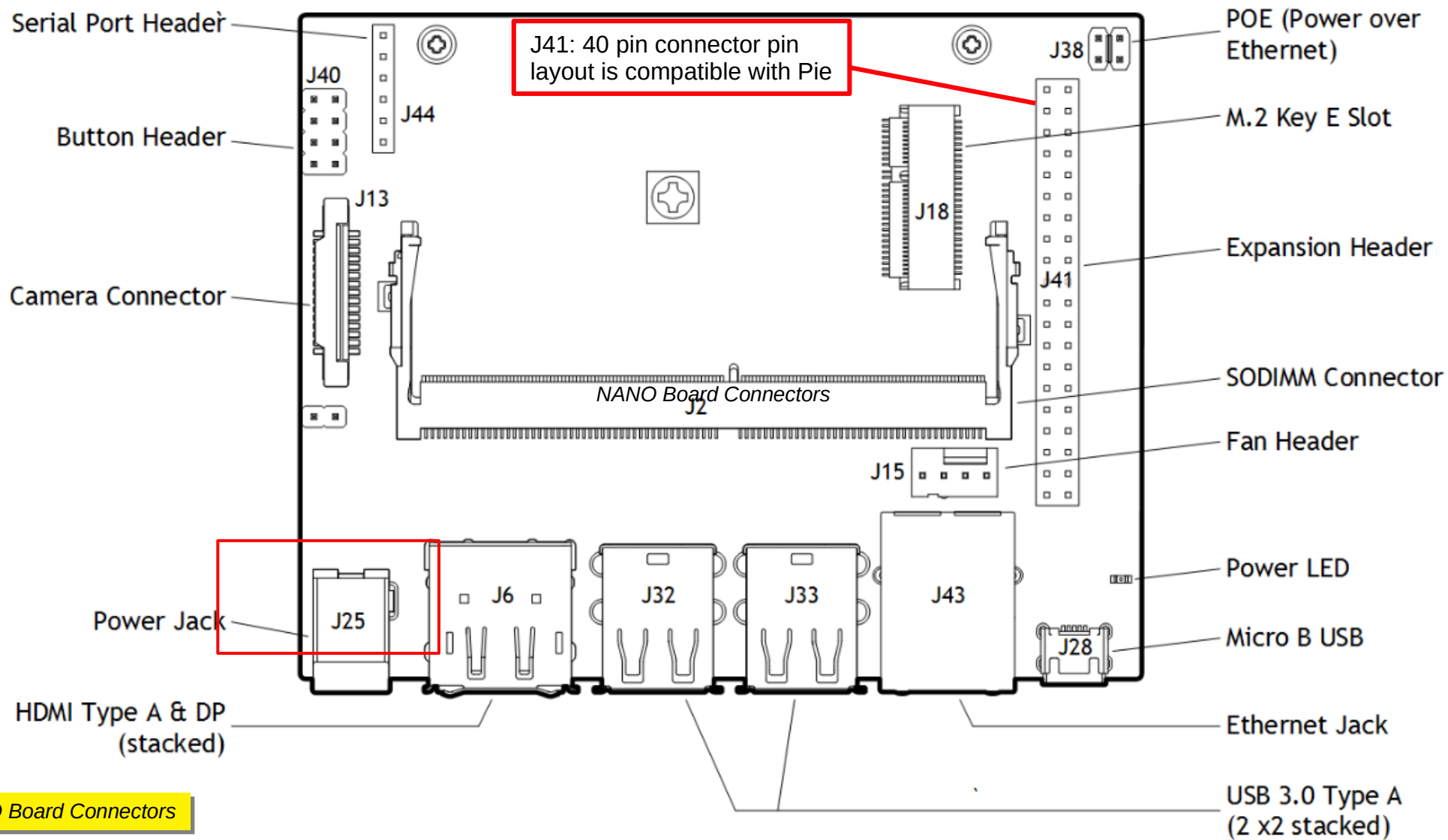
Note: I2C and UART pins are connected to hardware and should not be reassigned. By default, all other pins (except power) are assigned as GPIO. Pins labeled with other functions are recommended functions if using a different device tree.

1. take Pin 1 Vcc (3.3V) and Pin 39 GND to test out LED, make sure you can light up a LED with 220 Ohm resistor in series.

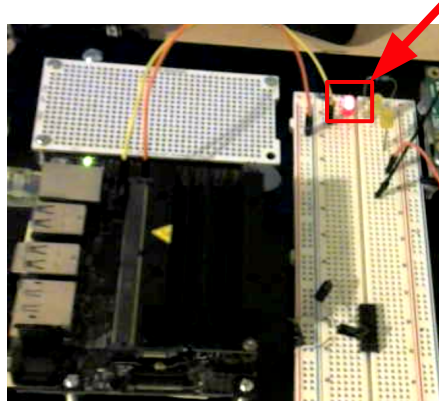
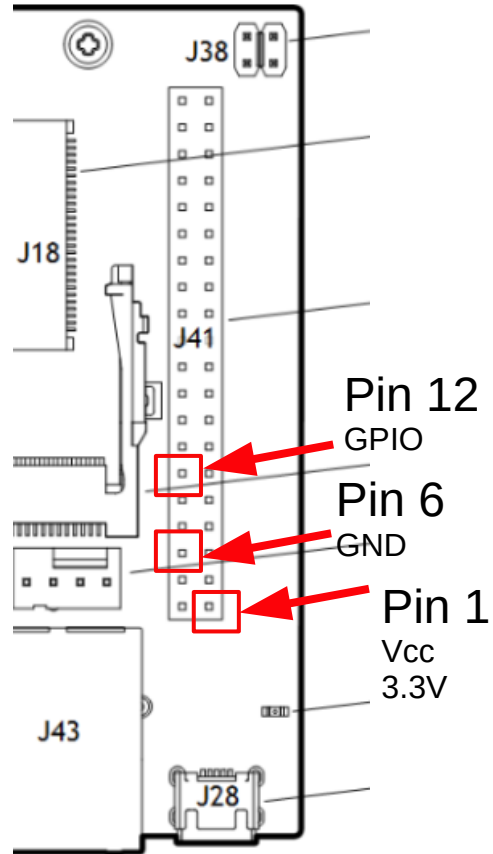
	GND	25	26	SPI_1_CS1	gpio20
	I2C_1_SDA I2C Bus 0	27	28	I2C_1_SCL I2C Bus 0	
gpio149	CAM_AF_EN	29	30	GND	
gpio200	GPIO_PZ0	31	32	LCD_BL_PWM	gpio168
gpio38	GPIO_PE6	33	34	GND	
gpio76	I2S_4_LRCK	35	36	UART_2_CTS	gpio51
gpio12	SPI_2_MOSI	37	38	I2S_4_SDIN	gpio77
	GND	39	40	I2S_4_SDOUT	gpio78

Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC Power	1	2	5.0 VDC Power	
	I2C_2_SDA I2C Bus 1	3	4	5.0 VDC Power	
	I2C_2_SCL I2C Bus 1	5	6	GND	
gpio216	AUDIO_MCLK	7	8	UART_2_TX /dev/ttyTHS1	
	GND	9	10	UART_2_RX /dev/ttyTHS1	
gpio50	UART_2_RTS	11	12	I2S_4_SCLK	gpio79
gpio14	SPI_2_SCK	13	14	GND	
gpio194	LCD_TE	15	16	SPI_2_CS1	gpio232
	3.3 VDC Power	17	18	SPI_2_CS0	gpio15
gpio16	SPI_1_MOSI	19	20	GND	
gpio17	SPI_1_MISO	21	22	SPI_2_MISO	gpio13
gpio18	SPI_1_SCK	23	24	SPI_1_CS0	gpio19
	GND	25	26	SPI_1_CS1	gpio20

Top View



Testing J41 40 Pin Connector with LED



We can control our LED from the command line. Here are some useful commands:

```
# Map GPIO Pin
# gpio79 is pin 12 on the Jetson Nano
$ echo 79 > /sys/class/gpio/export
# Set Direction
$ echo out > /sys/class/gpio/gpio79/direction
# Bit Bangin'!
$ echo 1 > /sys/class/gpio/gpio79/value
$ echo 0 > /sys/class/gpio/gpio79/value
# Unmap GPIO Pin
$ echo 79 > /sys/class/gpio/unexport
# Query Status
$ cat /sys/kernel/debug/gpio
```

In the above code, the 79 refers to a translation of the Linux sysfs GPIO named gpio79. If we look at the [Jetson Nano J41 Header Pinout](#), we can see that gpio79 is physically pin 12 of the header.

```
$echo 79 > /sys/class/gpio/export
```

```
$ echo out > /sys/class/gpio/gpio79/direction
```

```
$echo 1 > /sys/class/gpio/gpio79/value
```

```
$echo 0 > /sys/class/gpio/gpio79/value
```

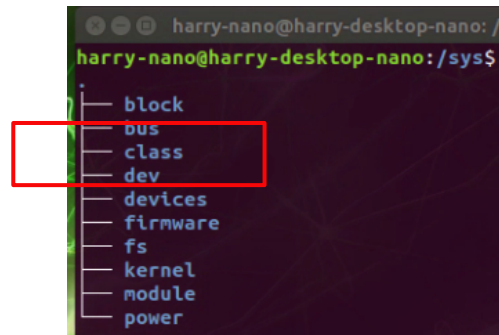
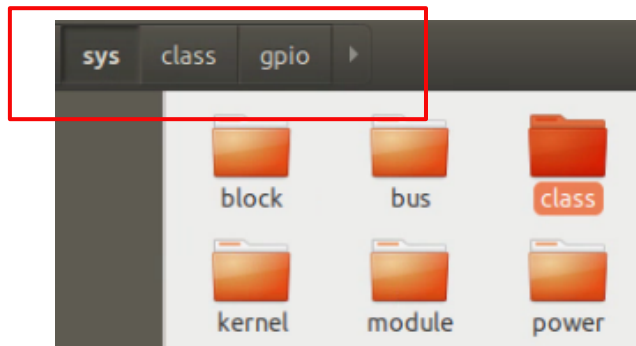
```
$echo 79 /sys/class/gpio/unexport
```

```
$cat /sys /kernel /debug/gpio
```

GPIO sysfs files from C/C++



Use the sysfs file nodes under `/sys/class/gpio/`. Reading and writing to these files is how languages like C/C++, Python, and other libraries implement device drivers.





Jetson NANO Adaptation and Bring Up

This is the most comprehensive reference source

https://docs.nvidia.com/jetson/141/index.html#page/Tegra%20Linux%20Driver%20Package%20Development%20Guide/adaptation_and_bringup_nano.html#wwplD0E0RR0HA

Nano Boards

Jetson Nano™ devices	Jetson Nano (P3448-0000) <i>Developer kit version</i>	Jetson Nano Developer Kit (P3450-0000) †; includes P3448-0000 module
	Jetson Nano (P3448-0002)	
	Jetson Nano 2GB (P3448-0003) <i>For experimental & educational use only</i>	Jetson Nano 2GB Developer Kit (P3541-0000, P3541-0001); includes P3448-0003 module

Board Configuration and Developer Kits

Jetson Nano

Jetson Nano 2GB

Board Naming

Placeholders in the Porting Instructions

Root Filesystem Configuration

Pinmux Changes

Updating the Bootloader Pinmux

Accessing GPIOs via “gpio” Device Labels

Exporting Pinmux for the Jetson Linux Kernel

Porting U-Boot

Porting the Linux Kernel

BSP (board support package)

<https://www.digikey.com/en/maker/projects/getting-started-with-the-nvidia-jetson-nano-part-1-setup/2f497bb88c6f4688b9774a81b80b8ec2>

