

AI on NVIDIA Jetson Nano (Day 6)

Outline

- Face Mask Detection Project (cont'd)
- Jetson Nano GPIO

Prerequisites

- Jetson Nano Developer Kit
- Computer with Internet Access and SD card port
- microSD Memory Card (32GB UHS-I minimum)
- Compatible 5V 4A Power Supply with 2.1mm DC barrel connector
- 2-pin jumper
- USB cable (Micro-B to Type-A)
- Logitech C270 Webcam (Optional)
- LED & Resistor (220 Ohm)

Face Mask Detection Project

- Face Detection
- Mask Detection
- Trigger the LED is on when the face mask is detected.
- Download from Venus-Solutions GitHub account

```
$ git clone https://github.com/Venus-Solutions/nvidia-jetson-hands-on-training
```

Jetson Nano GPIO

- Jetson TX1, TX2, AGX Xavier, and Nano development boards contain a 40 pin GPIO header, similar to the 40 pin header in the Raspberry Pi.
- These GPIOs can be controlled for digital input and output using the Python library provided in the Jetson GPIO library package.
- The library has the same API as the RPi.GPIO library for Raspberry Pi in order to provide an easy way to move applications running on the Raspberry Pi to the Jetson board.

Jetson Nano Pinout

Jetson Nano Dev-Board Expansion Header

Alt Function	Linux(BCM)	Board Label		Board Label	Linux(BCM)	Alt Function
DAP4_DOUT	78(21)	D21	40 39	GND		
DAP4_DIN	77(20)	D20	38 37	D26	12(26)	SPI2_MOSI
UART2_CTS	51(16)	D16	36 35	D19	76(19)	DAP4_FS
		GND	34 33	D13	38(13)	GPIO_PE6
LCD_BL_PWM	168(12)	D12	32 31	D6	200(6)	GPIO_PZ0
		GND	30 29	D5	149(5)	CAM_AF_EN
		D1/ID_SC	28 27	D0/ID_SD		
SPI1_CS1	20(7)	D7	26 25	GND		
SPI1_CS0	19(8)	D8	24 23	D11	18(11)	SPI1_SCK
SPI2_MISO	13(25)	D25	22 21	D9	17(9)	SPI1_MISO
		GND	20 19	D10	16(10)	SPI1_MOSI
SPI2_CS0	15(24)	D24	18 17	3.3V		
SPI2_CS1	232(23)	D23	16 15	D22	194(22)	LCD_TE
		GND	14 13	D27	14(27)	SPI2_SCK
DAP4_SCLK	79(18)	D18	12 11	D17	50(17)	UART2_RTS
		RXD/D15	10 9	GND		
		TXD/D14	8 7	D4	216(4)	AUDIO_MCLK
		GND	6 5	SCL/D3		
		5V	4 3	SDA/D2		
		5V	2 1	3.3V		

Jetson Nano vs. Raspberry Pi

Alt Function	Linux(BCM)	Board Label	Board Label	Linux(BCM)	Alt Function
DAP4_DOUT	78(21)	D21	40 39	GND	
DAP4_DIN	77(20)	D20	38 37	D26	12(26) SPI2_MOSI
UART2_CTS	51(16)	D16	36 35	D19	76(19) DAP4_FS
		GND	34 33	D13	38(13) GPIO_PE6
LCD_BL_PWM	168(12)	D12	32 31	D6	200(6) GPIO_PZ0
		GND	30 29	D5	149(5) CAM_AF_EN
		D1/ID_SC	28 27	D0/ID_SD	
SPI1_CS1	20(7)	D7	26 25	GND	
SPI1_CS0	19(8)	D8	24 23	D11	18(11) SPI1_SCK
SPI2_MISO	13(25)	D25	22 21	D9	17(9) SPI1_MISO
		GND	20 19	D10	16(10) SPI1_MOSI
SPI2_CS0	15(24)	D24	18 17	3.3V	
SPI2_CS1	232(23)	D23	16 15	D22	194(22) LCD_TE
		GND	14 13	D27	14(27) SPI2_SCK
DAP4_SCLK	79(18)	D18	12 11	D17	50(17) UART2_RTS
		RXD/D15	10 9	GND	
		TXD/D14	8 7	D4	216(4) AUDIO_MCLK
		GND	6 5	SCL/D3	
		5V	4 3	SDA/D2	
		5V	2 1	3.3V	

Alternate Function				Alternate Function
	3.3V PWR	1		2 5V PWR
I2C1 SDA	GPIO 2	3		4 5V PWR
I2C1 SCL	GPIO 3	5		6 GND
	GPIO 4	7		8 UART0 TX
	GND	9		10 UART0 RX
	GPIO 17	11		12 GPIO 18
	GPIO 27	13		14 GND
	GPIO 22	15		16 GPIO 23
	3.3V PWR	17		18 GPIO 24
SPI0 MOSI	GPIO 10	19		20 GND
SPI0 MISO	GPIO 9	21		22 GPIO 25
SPI0 SCLK	GPIO 11	23		24 GPIO 8
	GND	25		26 GPIO 7
	Reserved	27		28 Reserved
	GPIO 5	29		30 GND
	GPIO 6	31		32 GPIO 12
	GPIO 13	33		34 GND
SPI1 MISO	GPIO 19	35		36 GPIO 16
	GPIO 26	37		38 GPIO 20
	GND	39		40 GPIO 21
				SPI0 CS0
				SPI0 CS1
				SPI1 CS0
				SPI1 MOSI
				SPI1 SCLK

Jetson Nano GPIO Library

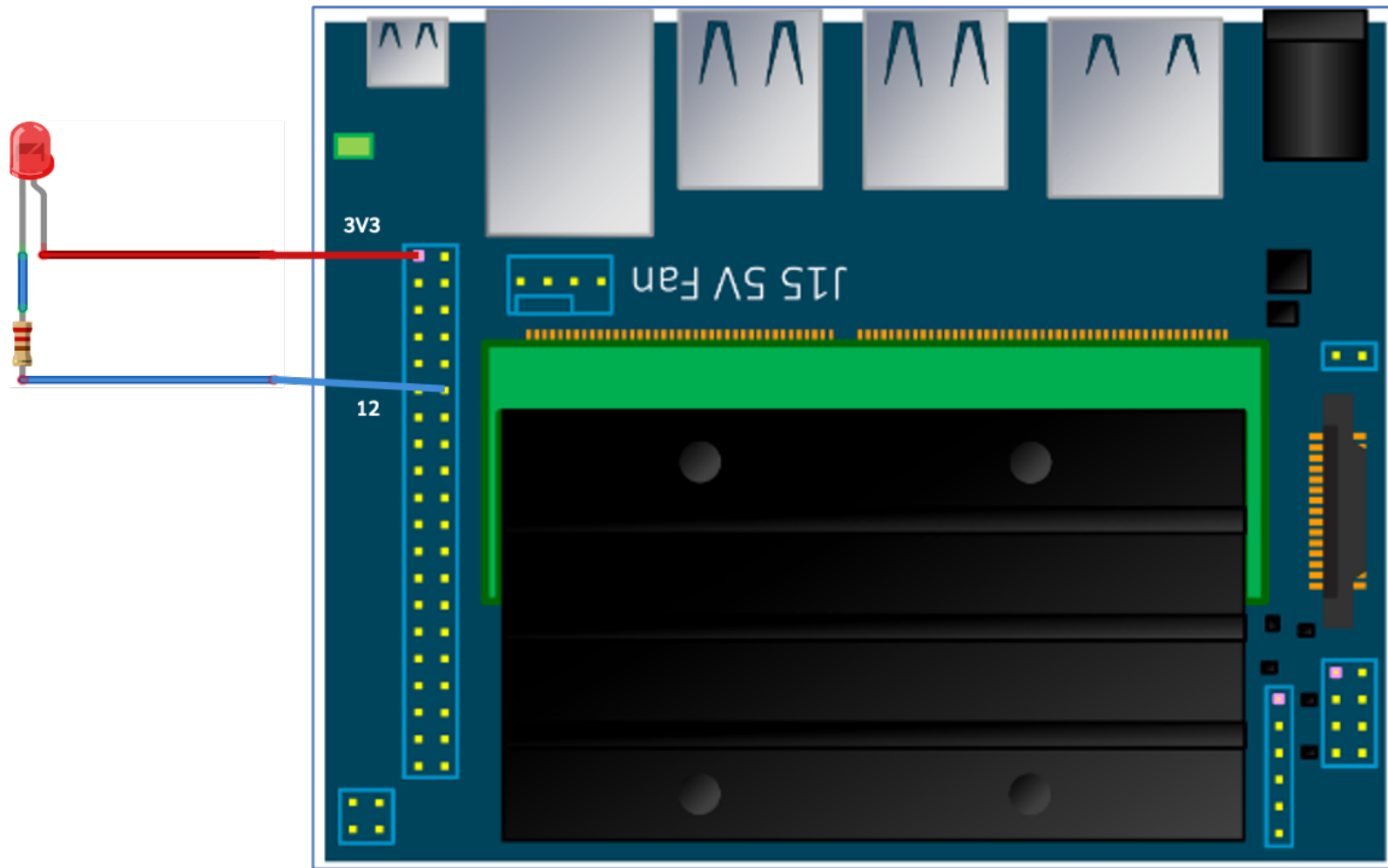
- Installation

```
$ sudo pip3 install Jetson.GPIO
```

- Setting User Permissions

```
$ sudo groupadd -f -r gpio  
$ sudo usermod -a -G gpio YOUR_USER_NAME
```


Control an LED with Jetson Nano



Control an LED with Jetson Nano (cont'd)

```
import Jetson.GPIO as GPIO
import time

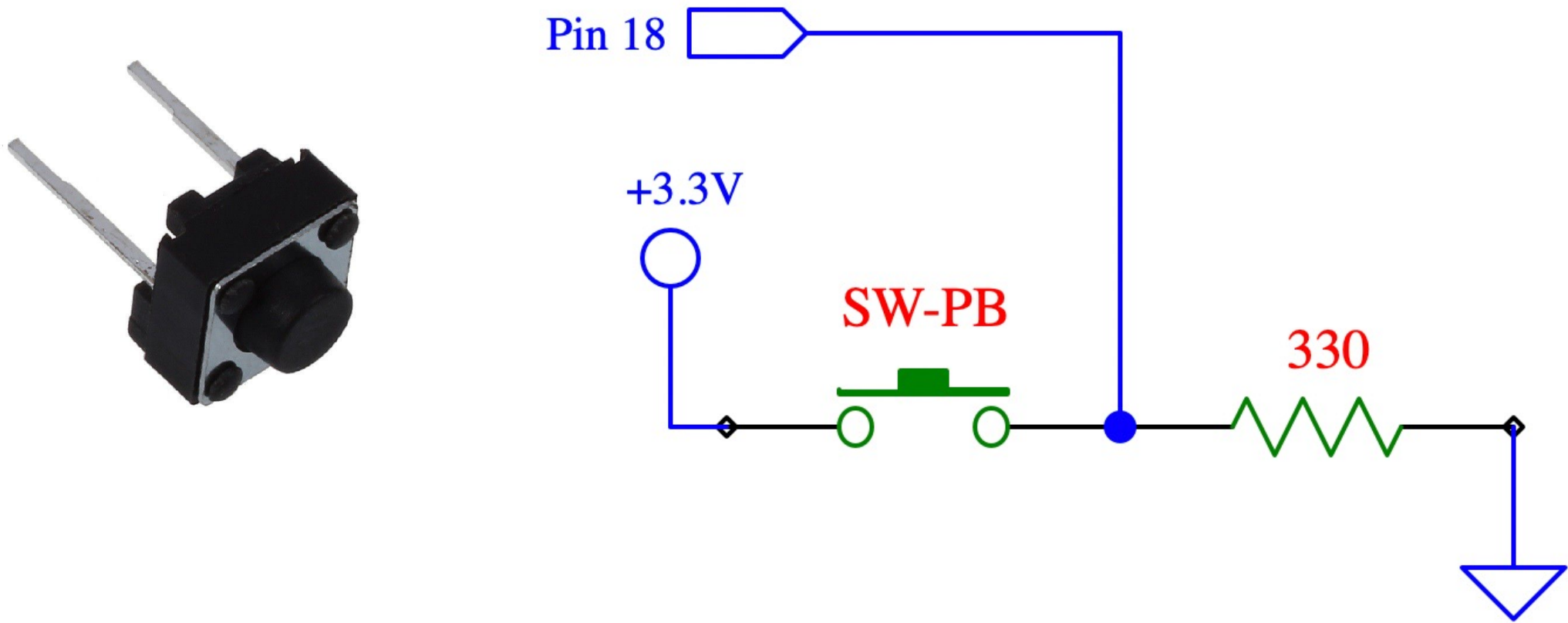
led_pin = 11

def main():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(led_pin, GPIO.OUT, initial=GPIO.LOW)

    try:
        while True:
            GPIO.output(led_pin, GPIO.HIGH)
            time.sleep(1)
            GPIO.output(led_pin, GPIO.LOW)
            time.sleep(1)
    finally:
        GPIO.cleanup()

if __name__ == '__main__':
    main()
```

Control a push button switch with Jetson Nano



Control a push button switch with Jetson Nano (cont'd)

```
import Jetson.GPIO as GPIO
import time

switch_pin = 12

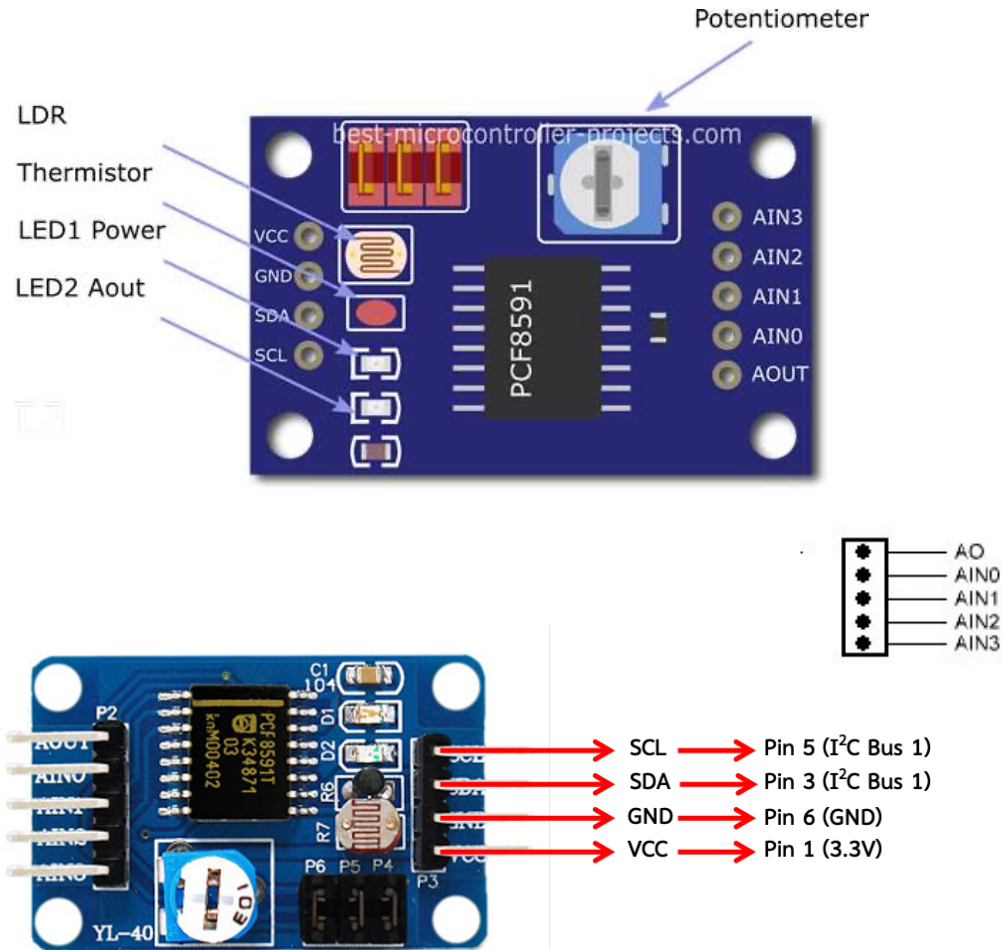
def main():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(switch_pin, GPIO.IN)

    try:
        while True:
            if GPIO.input(switch_pin) == GPIO.HIGH:
                print("Button was pushed.")

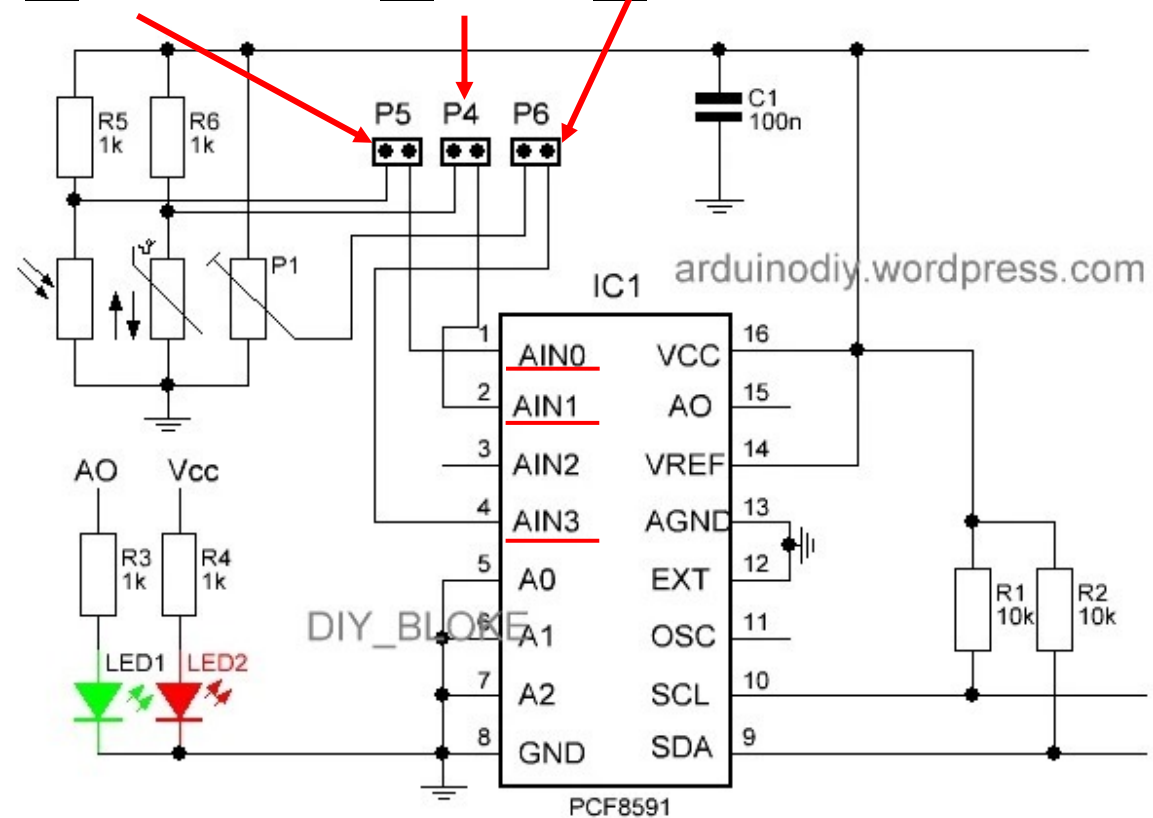
                time.sleep(1)
    finally:
        GPIO.cleanup()

if __name__ == '__main__':
    main()
```

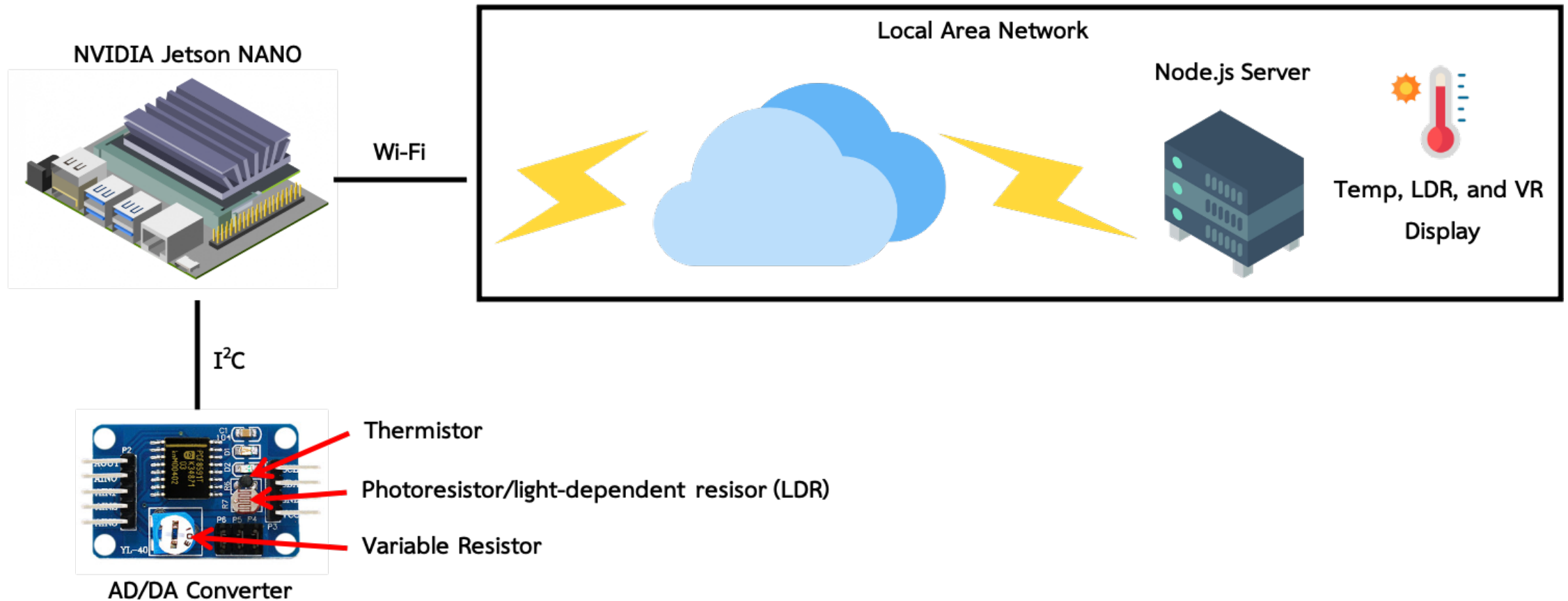
Control a YL—40 module with Jetson Nano



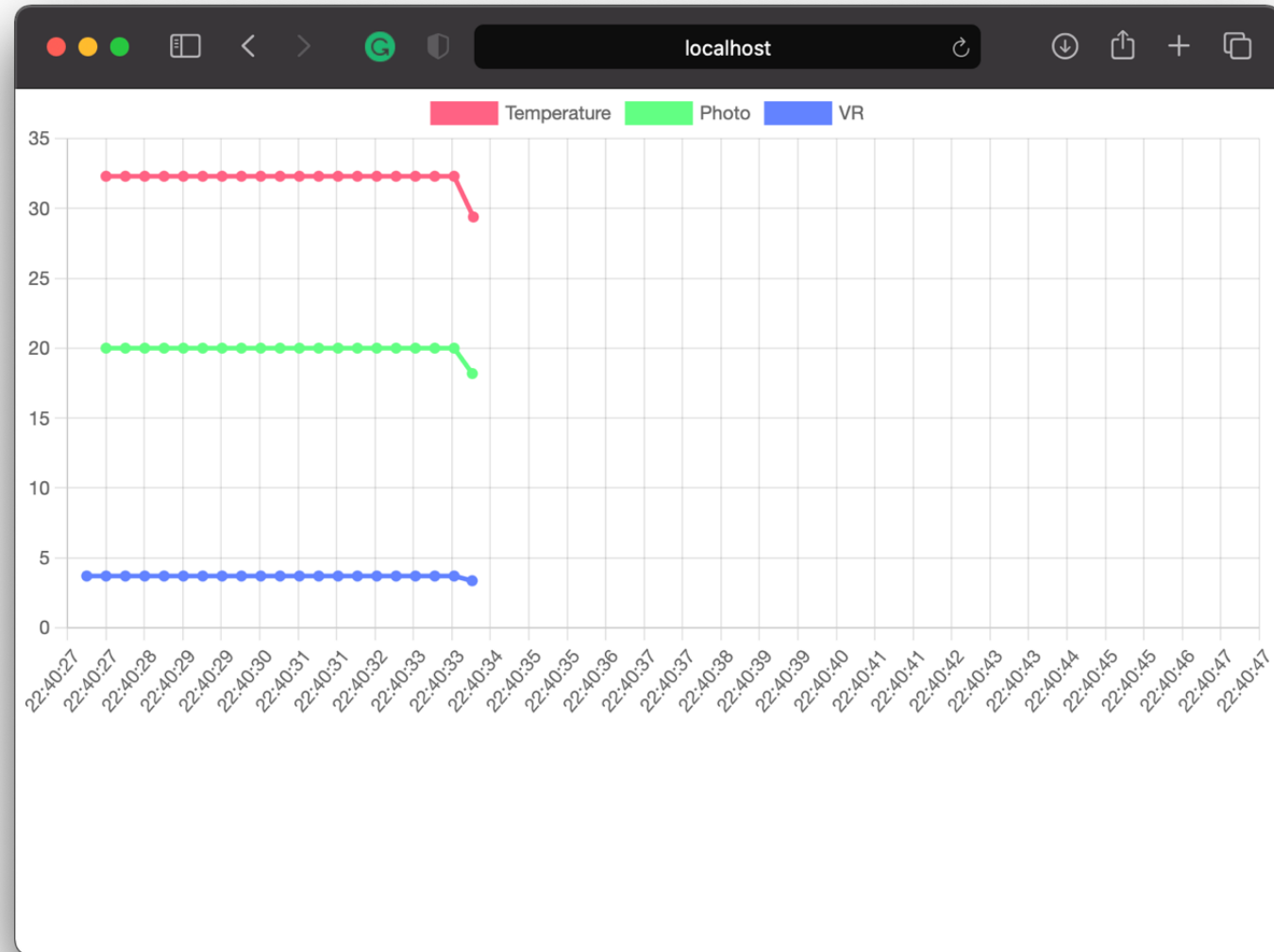
P5 = Thermistor, P4 = LDR, P6 = Potentionmeter



Control a YL—40 module with Jetson Nano (cont'd)



Control a YL—40 module with Jetson Nano (cont'd)



Control a YL—40 module with Jetson Nano (cont'd)

- Installation

```
$ sudo apt-get install python3-smbus  
$ sudo pip3 install python-socketio
```

- Read i2c Address (Bus 1 – Pin 3, 5)

```
$ i2cdetect -r -y 1
```


Control a YL—40 module with Jetson Nano (cont'd)

Jetson Nano Dev-Board Expansion Header

I2c Bus 0
27 (SDA), 28 (SCL)

Alt Function	Linux(BCM)	Board Label	Board Label	Linux(BCM)	Alt Function
DAP4_DOUT	78(21)	D21	40 39	GND	
DAP4_DIN	77(20)	D20	38 37	D26	12(26) SPI2_MOSI
UART2_CTS	51(16)	D16	36 35	D19	76(19) DAP4_FS
		GND	34 33	D13	38(13) GPIO_PE6
LCD_BL_PWM	168(12)	D12	32 31	D6	200(6) GPIO_PZ0
		GND	30 29	D5	149(5) CAM_AF_EN
		D1/ID_SC	28 27	D0/ID_SD	
SPI1_CS1	20(7)	D7	26 25	GND	
SPI1_CS0	19(8)	D8	24 23	D11	18(11) SPI1_SCK
SPI2_MISO	13(25)	D25	22 21	D9	17(9) SPI1_MISO
		GND	20 19	D10	16(10) SPI1_MOSI
SPI2_CS0	15(24)	D24	18 17	3.3V	
SPI2_CS1	232(23)	D23	16 15	D22	194(22) LCD_TE
		GND	14 13	D27	14(27) SPI2_SCK
DAP4_SCLK	79(18)	D18	12 11	D17	50(17) UART2_RTS
		RXD/D15	10 9	GND	
		TXD/D14	8 7	D4	216(4) AUDIO_MCLK
		GND	6 5	SCL/D3	
		5V	4 3	SDA/D2	
		5V	2 1	3.3V	

I2c Bus 1
3 (SDA), 5 (SCL)

Control a YL—40 module with Jetson Nano (cont'd)

- Web App

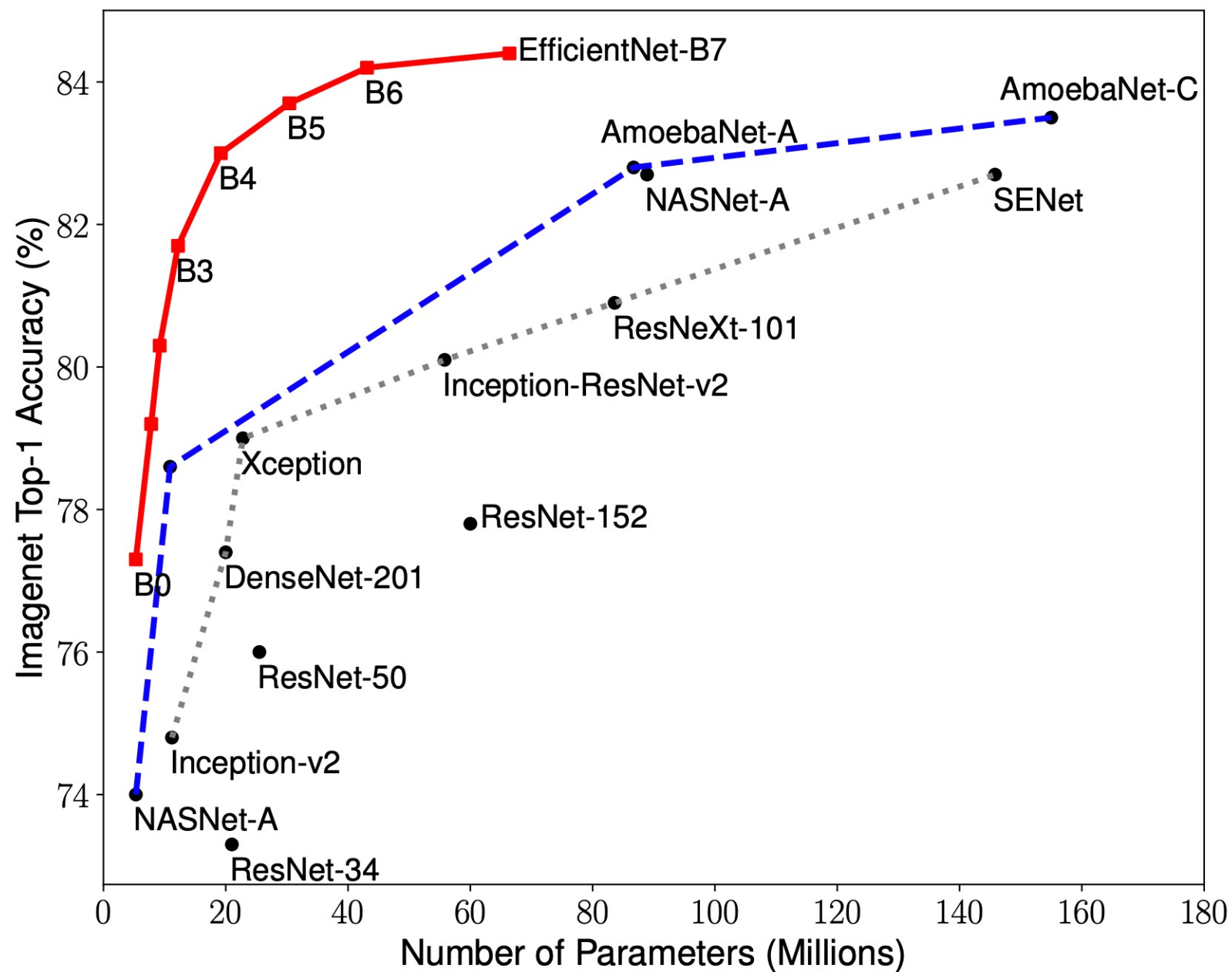
```
$ npm init  
$ npm install  
$ npm install express --save  
$ npm install socket.io
```

- Run web app

```
$ node app.js
```

- Go to link: <http://localhost:8000>

Extra



Paper: <https://arxiv.org/pdf/1905.11946v5.pdf>

Extra

Model	2D-CNN	3D-CNN
	Params	Params
VGG-16	134.7 M	179.1 M
ResNet-18	11.4 M	33.3 M
ResNet-34	21.5 M	63.6 M
ResNet-50	23.9 M	46.4 M
ResNet-101	42.8 M	85.5 M
ResNet-152	58.5 M	117.6 M
DenseNet-121	7.2 M	11.4 M
DenseNet-169	12.8 M	18.8 M

References

- Jetson GPIO – Linux for Tegra
 - <https://github.com/NVIDIA/jetson-gpio>
- PCF8591 Datasheet
 - <https://www.nxp.com/docs/en/data-sheet/PCF8591.pdf>
- PCF8591 ADC (YL-40 Module)
 - https://www.elektronika.ftn.uns.ac.rs/racunarska-elektronika/wp-content/uploads/sites/21/2018/03/YL_40_PCF8591.pdf
- Jetson Nano Using i2c
 - <https://www.jetsonhacks.com/2019/07/22/jetson-nano-using-i2c/>
- Socket.IO Server
 - <https://python-socketio.readthedocs.io/en/latest/server.html>