

Using the GPIO interface on the Raspberry Pi

Syed Faisal Akber

2016-01-14



- 1 Introduction
- 2 P1 - GPIO Header
- 3 Simple I/O
- 4 Pulse-Width Modulation (PWM)
- 5 Serial Communications
- 6 Other Topics



Introduction

- Raspberry Pi has many interfaces
- Some we are already familiar with
- Today we'll look at P1 (GPIO Interface)
- Its basic usage and pointers to more detail

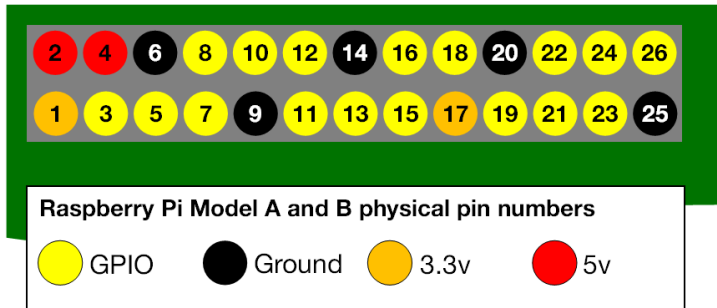


GPIO Pinouts I

- Each pin has a purpose
- Power and Ground
- I/O Pins
- PWM
- I^2C
- SPI
- No analog inputs
- No analog outputs*



GPIO Pinouts II



GPIO Pinouts III

RASPBERRY PI Revision 2 Pinout

<http://www.pinballsp.com>



UART-RTS

SPI

3V3

GPIO2
SDA

GPIO3
SCL

GPIO4

Ground

GPIO17

GPIO27

GPIO22

3V3

GPIO10
MOSI

GPIO9
MISO

GPIO11
CLK

Ground

Revision 2.0

5V **+5v**

5V

Ground **GND**

GPIO14
TXD
GPIO15
RXD **UART**

GPIO18 **PWM**

Ground

GPIO23

GPIO24

Ground

GPIO25

GPIO8
CE0
GPIO7
CE1 **SPI**

<https://www.facebook.com/pages/PinballSP/336137879799788>



GPIO Pinouts IV

3.3V	1		2	5V
GPIO 2 (I2C1_SDA)	3		4	5V
GPIO 3 (I2C1_SCL)	5		6	GND
GPIO 4 (GPCLK0)	7		8	GPIO 14 (UART_TXD)
GND	9		10	GPIO 15 (UART_RXD)
GPIO 17	11		12	GPIO 18
GPIO 22	13		14	GND
GPIO 27	15		16	GPIO 23
3.3V	17		18	GPIO 24
GPIO 10 (SPI_MOSI)	19		20	GND
GPIO 9 (SPI_MISO)	21		22	GPIO 25
GPIO 11 (SPI_SCLK)	23		24	GPIO 8 (SPI_CE0)
GND	25		26	GPIO 7 (SPI_CE1)
ID_SD	27		28	ID_SC
GPIO 5	29		30	GND
GPIO 6	31		32	GPIO 12
GPIO 13	33		34	GND
GPIO 19	35		36	GPIO 16
GPIO 26	37		37	GPIO 20
GND	39		40	GPIO 21

Key

- Power (5 Volts)
- Power (3.3 Volts)
- Ground
- General Inputs/Outputs
- I2C Interface
- SPI Interface
- UART Interface
- ID EEPROM Interface



Simple Output I

- Simple outputs
 - 3.3V if set to 1
 - 0.0V if set to 0
- Easiest to program
- Can use Scratch, Python or C
- Example



Simple Output II

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

ledpin = 18
GPIO.setup(ledpin, GPIO.OUT)

GPIO.output(ledpin, 1)
GPIO.cleanup()
```



Simple Input I

- $3.3V = 1$
- $0.0V = 0$
- Use with buttons and other digital devices
- Built-in pull up/down resistors
- Example code



Simple Input II

```
btnpin = 18
GPIO.setup(btnpin, GPIO.IN,
            pull_up_down=GPIO.PUD_DOWN)

while True:
    button = GPIO.input(btnpin)
    if button:
        print('Button pressed')
        time.sleep(0.2) # Debounce
GPIO.cleanup()
```

- Debounce switches either in hardware or software using loops

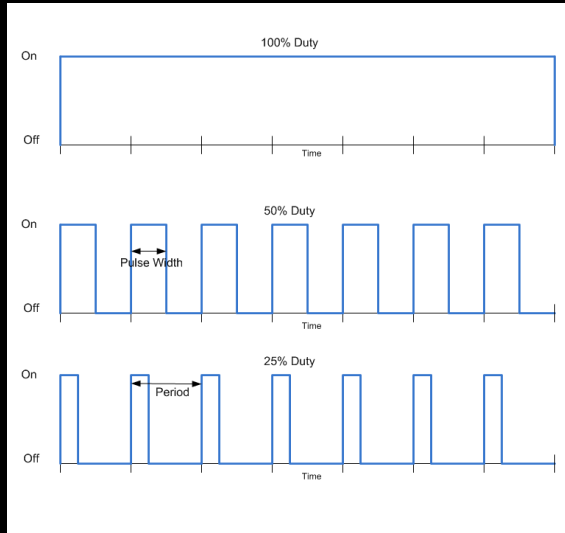


Pulse-Width Modulation (PWM) I

- PWM creates a pulse of varying widths



Pulse-Width Modulation (PWM) II



Pulse-Width Modulation (PWM) III

- Dim LED/Lights
- Control RGB LEDs
- Control motor speed
- Control servo motors
- Use as a timing signal
- Use to broadcast radio signals (PiFM)
- Example code



Pulse-Width Modulation (PWM) IV

```
ledpin = 18
freq = 50 # Hz
GPIO.setup(ledpin, GPIO.OUT)
pwm = GPIO.PWM(ledpin, freq)
pwm.start(25) # duty cycle
time.sleep(5)
pwm.ChangeDutyCycle(75)
time.sleep(5)
GPIO.cleanup()
```



Serial Communications I

- Standard via UART (Pins 8 and 10)
- By default, kernel messages and getty are running
- Update grub configuration and inittab to disable defaults
- To connect RS-232 devices use a MAX3232 to adjust voltages
- Other different interfaces exist
 - Serial Peripheral Interface (SPI)
 - Inter-Integrated Circuit (I^2C)
- Many peripherals use SPI or I^2C such as, LCD Screens and other devices



What to do when you're running out of pins!

- Multiplexing
- Specialized hardware drivers (i.e. LED Matrix drivers)
- Daisy-chaining
- Add-on boards (Guzunty, Gert, ...)



Questions?

