

## Programming Pi2Go in Python

The best way to program the Pi2Go is by using the python library module included on the SD card: `ttspi2go.py`

There are functions in the library module that handle everything including turning the motors at varying speeds in either direction; setting the colours of the LEDs, reading the InfraRed sensors and determining the distance to objects using the ultrasonic sensor.

For a complete list of the functions available, read the first few lines of the library file in any text editor. They are also copied at the end of this document.

If you want to access the features on the Pi2Go using your own code, then you will need to know the details of all the GPIO pins used by the Pi2Go. When discussing the GPIO pins we can either use physical pin numbers on the J5 connector on the Pi, or we can refer to the GPIO number given by Broadcom. For various reasons we use the physical pin numbers within the library, but the following table lists both. If using the module `RPi.GPIO` to access the GPIO pins, then use:

`GPIO.setmode (GPIO.BOARD)` for **physical pin numbers** or  
`GPIO.setmode (GPIO.BCM)` for **Broadcom signal numbers**

| Function               | Physical Pin | BCM Number | Comments   |
|------------------------|--------------|------------|--|
| Left Motor             | 24, 26       | 8, 7       | Set one HIGH and other LOW to move                         |
| Right Motor            | 19, 21       | 10, 9      | Set one HIGH and other LOW to move                         |
| Front Left IR Sensor   | 11           | 17         | Is LOW when there is a reflection                          |
| Front Right IR Sensor  | 7            | 4          | Is LOW when there is a reflection                          |
| Front Centre IR Sensor | 13           | 27         | Is LOW when there is a reflection                          |
| Left Line Sensor       | 12           | 18         | Is HIGH above non-reflecting Line                          |
| Right Line Sensor      | 15           | 22         | Is HIGH above non-reflecting Line                          |
| Ultrasonic Sensor      | 8            | 14         | Set to Output to send pulse and Input to receive the reply |
| Switch                 | 16           | 23         | LOW when switch is pressed                                 |

NB.

- RGB LEDs are driven by a PCA9685 chip on the I<sup>2</sup>C bus
- Light sensors are read by a PCF8591 ADC chip on the I<sup>2</sup>C bus

### General Functions

- `init()`. Initialises GPIO pins, switches motors and LEDs Off, etc
- `cleanup()`. Sets all motors and LEDs off and sets GPIO to standard values
- `version()`. Returns 3 for Pi2Go. Invalid until after `init()` has been called

### Motor Functions

- `stop()`: Stops both motors
- `forward(speed)`: Sets both motors to move forward at speed.  $0 \leq \text{speed} \leq 100$
- `reverse(speed)`: Sets both motors to reverse at speed.  $0 \leq \text{speed} \leq 100$
- `spinLeft(speed)`: Sets motors to turn opposite directions at speed.  $0 \leq \text{speed} \leq 100$
- `spinRight(speed)`: Sets motors to turn opposite directions at speed.  $0 \leq \text{speed} \leq 100$
- `turnForward(leftSpeed, rightSpeed)`: Moves forwards in an arc by setting different speeds.  $0 \leq \text{leftSpeed}, \text{rightSpeed} \leq 100$
- `turnreverse(leftSpeed, rightSpeed)`: Moves backwards in an arc by setting different speeds.  $0 \leq \text{leftSpeed}, \text{rightSpeed} \leq 100$
- `go(leftSpeed, rightSpeed)`: controls motors in both directions independently using different positive/negative speeds.  $-100 \leq \text{leftSpeed}, \text{rightSpeed} \leq 100$
- `go(speed)`: controls motors in both directions together with positive/negative speed parameter.  $-100 \leq \text{speed} \leq 100$

### RGB LED Functions

- `setLED(LED, Red, Green, Blue)`: Sets the LED specified to required RGB value.  $0 \leq \text{LED} \leq 4$ ;  $0 \leq \text{R}, \text{G}, \text{B} \leq 4095$
- `setAllLEDs(Red, Green, Blue)`: Sets all LEDs to required RGB.  $0 \leq \text{R}, \text{G}, \text{B} \leq 4095$

### IR Sensor Functions

- `irLeft()`: Returns state of Left IR Obstacle sensor
- `irRight()`: Returns state of Right IR Obstacle sensor
- `irCentre()`: Returns state of Centre IR Obstacle sensor
- `irAll()`: Returns true if any of the Obstacle sensors are triggered
- `irLeftLine()`: Returns state of Left IR Line sensor
- `irRightLine()`: Returns state of Right IR Line sensor

### UltraSonic Functions

- `getDistance()`. Returns the distance in cm to the nearest reflecting object. 0 == no object

### **Light Sensor Functions**

- `getLight(Sensor)`. Returns the value 0..1023 for the selected sensor,  $0 \leq \text{Sensor} \leq 3$  (anticlockwise from Front-Right)
- `getLightFL()`. Returns the value 0..1023 for Front-Left light sensor
- `getLightFR()`. Returns the value 0..1023 for Front-Right light sensor
- `getLightBL()`. Returns the value 0..1023 for Back-Left light sensor
- `getLightBR()`. Returns the value 0..1023 for Back-Right light sensor

### **Switch Functions**

- `getSwitch()`. Not Active. Returns the value of the tact switch: True==pressed