CSC270

Circuits & Systems

CSC270 Unit #7

Introduction to the RPi's GPIO

Dominique Thiébaut

dthiebaut@smith.edu
Outline

- The Computer
  - History
  - Features
  - Unboxing!
  - Explore!
  - Raspbian?
  - How Fast?
  - Internet Address
  - Remote Connection
- Programming the GPIO
### GPIO Table

<table>
<thead>
<tr>
<th>Misc</th>
<th>C WiringPi</th>
<th>Python GPIOZero</th>
<th>Physical Pin #</th>
<th>Physical Pin #</th>
<th>Python GPIOZero</th>
<th>C WiringPi</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.3V</td>
<td></td>
<td>5V</td>
<td>5V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDA1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>17</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>11</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>14</td>
<td>8</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>17</td>
<td>11</td>
<td>2</td>
<td>12</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>22</td>
<td>15</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>10</td>
<td>19</td>
<td>22</td>
<td>25</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>9</td>
<td>21</td>
<td>23</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>11</td>
<td>23</td>
<td>26</td>
<td>7</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>21</td>
<td>5</td>
<td>29</td>
<td>22</td>
<td>12</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>22</td>
<td>6</td>
<td>31</td>
<td>32</td>
<td>16</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>23</td>
<td>13</td>
<td>33</td>
<td>36</td>
<td>19</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>24</td>
<td>19</td>
<td>35</td>
<td>38</td>
<td>20</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>26</td>
<td>37</td>
<td>40</td>
<td>21</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

[https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access](https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access)
### GPIO Pinout

<table>
<thead>
<tr>
<th>Misc</th>
<th>WiringPi</th>
<th>Python GPIOZero</th>
<th>Physical Pin #</th>
<th>Physical Pin #</th>
<th>Python GPIOZero</th>
<th>WiringPi</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
<td>5V</td>
<td>5V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDA1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>8</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>12</td>
<td>18</td>
<td>1</td>
<td>PWM</td>
</tr>
<tr>
<td>0</td>
<td>17</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>18</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>15</td>
<td>15</td>
<td>16</td>
<td>23</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>19</td>
<td>19</td>
<td>18</td>
<td>24</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>25</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>23</td>
<td>23</td>
<td>24</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>26</td>
<td>7</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0</td>
<td>27</td>
<td>27</td>
<td>28</td>
<td>1</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>21</td>
<td>5</td>
<td>29</td>
<td>28</td>
<td>1</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>22</td>
<td>6</td>
<td>31</td>
<td>32</td>
<td>12</td>
<td>26</td>
<td>PWM</td>
</tr>
<tr>
<td>PWM</td>
<td>23</td>
<td>13</td>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>24</td>
<td>19</td>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>37</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access](https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access)
**GPIO**

- **SDA1**: 8, 2, 3
- **SCL1**: 9, 3, 5
- **Clock**: 7, 4, 7

<table>
<thead>
<tr>
<th>Misc</th>
<th>WiringPi</th>
<th>Python GPIOZero</th>
<th>Physical Pin #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5V</td>
</tr>
</tbody>
</table>

- **Clock**: 0, 17, 11
- **PWM**: 12, 10, 19
- **GND**: 3, 22, 15
- **Clock**: 4, 18, 24
- **GND**: 5, 12, 18
- **Clock**: 6, 14, 15
- **GND**: 7, 16, 23
- **Clock**: 8, 10, 19
- **GND**: 9, 12, 25
- **Clock**: 11, 13, 23
- **GND**: 14, 15, 26
- **Clock**: 27, 21, 29
- **GND**: 28, 22, 31
- **Clock**: 5, 21, 29
- **GND**: 6, 22, 32
- **Clock**: 13, 23, 33
- **GND**: 14, 24, 36
- **Clock**: 19, 25, 35
- **GND**: 26, 27, 38
- **Clock**: 26, 28, 37
- **GND**: 29, 30, 40

**Source**

https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access
• **40 pins**

• All (except power & ground) can be programmed as **digital input or output**

• **5V & 3.3V available.**

• Output high is 3.3 V (max), max current per pin 16mA

• **PWM** (software: all pins, hardware: 12, 13, 18, 19)

• SPI0 & SPI1: serial communication

• i2c (GPIO Pin 2 & 3)

• **Serial** (TxD: 14 & RxD: 15)

Two Terminal/Bash Commands

- `pinout`  
  provided by `gpiozero` python library

- `gpio readall`  
  provided by `wiringPi` C library
I/O Programming in C: 
WiringPi Library

http://wiringpi.com/
Wiring Pi

GPIO Interface library for the Raspberry Pi

About

WiringPi is a PIN based GPIO access library written in C for the BCM2835, BCM2836 and BCM2837 SoC devices used in all Raspberry Pi. versions. It’s released under the GNU LGPLv3 license and is usable from C, C++ and RTB (BASIC) as well as many other languages with suitable wrappers (See below) It’s designed to be familiar to people who have used the Arduino “wiring” system\(^1\) and is intended for use by experienced C/C++ programmers. It is not a newbie learning tool.

Recent Posts

- wiringPi – deprecated...
- wiringPi updated to 2.52 for the Raspberry Pi 4B
- wiringPi updated to 2.46 for the new Pi v3+
- wiringPi updated to 2.36
- wiringPi update to 2.29
Wiring Pi

GPIO Interface library for the Raspberry Pi

The I2C, SPI and UART interfaces can also be used as general purpose I/O pins when not being used in their bus modes, giving a grand total of $8 + 2 + 5 + 2 = 17$ I/O pins on the P1 connector (plus 4 more on the P5 connector on a Revision 2 Pi) and 28 I/O pins on the B+ and version 2 and 3 boards (Although 2 are reserved for the HAT I2C interface, but can be used as normal GPIOs if not using a HAT board).

**WiringPi** includes a command-line utility `gpio` which can be used to program and setup the GPIO pins. You can use this to read and write the pins and even use it to control them from shell scripts.
WiringPi
Core Functions

- void pinMode (int pin, int mode) ;
- void pullUpDnControl (int pin, int pud) ;
- void digitalWrite (int pin, int value) ;
- void pwmWrite (int pin, int value) ;
- int digitalRead (int pin) ;
- analogRead (int pin) ;
- analogWrite (int pin, int value) ;

http://wiringpi.com/reference/core-functions/
Setup For My RPi

Green LED (green wire) —> Physical Pin 33
Red LED (red wire) —> Physical Pin 11
Switch (orange wire) —> Physical Pin 12
// blink.c
// D. Thiebaut
// Copied from the WiringPi library site.
// A blink program that activates an LED connecting to
// WiringPi Pin 0, which is GPIO 17, which is Physical Pin 11.

// use the WiringPi library
#include <wiringPi.h>

int main (void) {

    // setup the RPi, and define WiringPin 0 as output
    wiringPiSetup ();
    pinMode (0, OUTPUT);

    // loop forever
    for (;;) {
        // turn LED ON
        digitalWrite (0, HIGH);
        // wait 1 sec
        delay (1000);
        // turn LED OFF
        digitalWrite (0, LOW);
        // wait 1 sec
        delay (1000);
    }

    // we'll never reach that point...
    return 0;
}
// blink.c
// D. Thiebaut
// Copied from the WiringPi library site.
// A blink program that activates an LED connecting to
// WiringPi Pin 0, which is GPIO 17, which is Physical Pin 11.

// use the WiringPi library
#include <wiringPi.h>

int main (void) {

    // setup the RPi, and define WiringPin 0 as output
    wiringPiSetup () ;
    pinMode (0, OUTPUT) ;

    // loop forever
    for (;;) {
        // turn LED ON
        digitalWrite( 0, HIGH ) ;
        // wait 1 sec
        delay( 1000 ) ;
        // turn LED OFF
        digitalWrite( 0, LOW ) ;
        // wait 1 sec
        delay( 1000 ) ;
    }

    // we'll never reach that point...
    return 0;
}
// blink.c
// D. Thiebaut
// Copied from the WiringPi library site.
// A blink program that activates an LED connecting to
// WiringPi Pin 0, which is GPIO 17, which is Physical Pin 11.

// use the WiringPi library
#include <wiringPi.h>

int main (void) {

    // setup the RPi, and define WiringPin 0 as output
    wiringPiSetup ();
    pinMode (0, OUTPUT);

    // loop forever
    for (;;) {
        // turn LED ON
        digitalWrite (0, HIGH);
        // wait 1 sec
        delay( 1000 );
        // turn LED OFF
        digitalWrite (0, LOW);
        // wait 1 sec
        delay( 1000 );
    }

    // we'll never reach that point...
    return 0;
}
Video Demo

```c
#include <wiringPi.h>

int main (void) {
    // setup the RPi, and define WiringPi 0 as output
    wiringPiSetup ();
    pinMode (0, OUTPUT);

    // loop forever
    for (;;) {
        // turn LED ON
        digitalWrite( 0, HIGH );
        // wait 1 sec
        delay( 1000 );
        // turn LED OFF
        digitalWrite( 0, LOW );
        // wait 1 sec
        delay( 1000 );
    }

    // we'll never reach that point...
    return 0;
}
```
// button.c
// D. Thiebaut
// Button is on WiringPi Pin 1
// LED is on wiringPi Pin 0
//
// Compile and run as follows:
//
// gcc -o button button.c -lwiringPi
// ./button
//
#include <wiringPi.h>

const int LED = 0;
const int BUTTON = 1;

int main (void) {

// setup
wiringPiSetup ();

pinMode(LED, OUTPUT);
pinMode(BUTTON, INPUT);
pullUpDnControl( BUTTON, PUD_UP);

// infinite loop
for (;;) {
    int val = digitalRead( BUTTON );
    digitalWrite( LED, val );
}

// we will never reach this place...
return 0;
}
// Button is on WiringPi Pin 1
// LED is on wiringPi Pin 0
//
// Compile and run as follows:
//
// gcc -o button button.c -lwiringPi
// ./button
//
#include <wiringPi.h>

const int LED = 0;
const int BUTTON = 1;

int main (void) {

    // setup
    wiringPiSetup ();

    pinMode(LED, OUTPUT);
    pinMode(BUTTON, INPUT);
    pullUpDnControl( BUTTON, PUD_UP);

    // infinite loop
    for (;;) {
        int val = digitalRead(BUTTON);
        digitalWrite(LED, val);
    }

    // we will never reach this place...
    return 0;
}
I/O Programming in Python: GPIOzero

14. API - Output Devices

These output device component interfaces have been provided for simple use of everyday components. Components must be wired up correctly before use in code.

**Note**

All GPIO pin numbers use Broadcom (BCM) numbering by default. See the Pin Numbering section for more information.

14.1. Regular Classes

The following classes are intended for general use with the devices they represent. All classes in this section are concrete (not abstract).

14.1.1. LED

```python
class gpiozero.LED(pin, *, active_high=True, initial_value=False, pin_factory=None)
```

Extends `DigitalOutputDevice` and represents a light emitting diode (LED).

Connect the cathode (short leg, flat side) of the LED to a ground pin; connect the anode (longer leg) to a limiting resistor; connect the other side of the limiting resistor to a GPIO pin (the limiting resistor can be placed either side of the LED).

## GPIO

![Raspberry Pi GPIO Board](https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access)

<table>
<thead>
<tr>
<th>Misc</th>
<th>WiringPi</th>
<th>Python</th>
<th>Physical Pin #</th>
<th>Physical Pin #</th>
<th>Python</th>
<th>WiringPi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.3V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDA1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>14</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>Clock</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access
# blink.py
# D. Thiebaut
# Taken from https://gpiozero.readthedocs.io/en/stable/
# activates an LED connected to GPIO Pin 17, which is Physical Pin 11.
#
from gpiozero import LED
from time import sleep

# SETUP
# define a LED object connected to GPIO Pin 17
led = LED(17)

# LOOP
while True:
    led.on()
    print( "LED ON" )
    sleep(1)
    led.off()
    print( "LED OFF" )
    sleep(1)
# blink.py
# Simple python program to make the LED connected to GPIO Pin 17, which is physical Pin 11.
# Taken from
#
from gpiozero import LED
from time import sleep

# define an LED object connected to Pin 17
led = LED(17)

def main():
    # forever...
    while True:
        led.on()
        print( "LED ON" )
        sleep(1)

        led.off()
        print( "LED OFF" )
        sleep(1)

main()
Push Button in Python

```python
# button.py
# D. Thiebaut
# Taken from https://gpiozero.readthedocs.io/en/stable/
# Activates an LED connected to GPIO Pin 17, which is Physical Pin 11,
# when a switch connected to GPIO Pin 18, which is Physical Pin 12,
# is activated. The switch connects to Physical Pin 12 and GND.
# Run in a terminal window:
#
# python button.py

from gpiozero import LED, Button
from signal import pause

# setup the I/O pins. Set the input for the button with
# a pull-up resistor.
led = LED( 17 )
button = Button( 18, pull_up = True )

# loop forever
while( True ):
    if button.is_pressed:
        led.on()
    else:
        led.off()

pause()
```
# when a switch connected to GPIO Pin 18, which is Physical Pin 12,
# is activated. The switch connects to Physical Pin 12 and GND.
# Run in a terminal window:
#
# python button.py

from gpiozero import LED, Button
from signal import pause

# setup the I/O pins. Set the input for the button with
# a pull-up resistor.
led = LED( 17 )
button = Button( 18, pull_up = True )

# loop forever
while( True ):
    if button.is_pressed:
        led.on()
    else:
        led.off()

pause()
You are Ready For Unit-7 Lab!