CSC270—Circuits & Systems

Week 10—Spring 2019

Dominique Thiébaut
dthiebaut@smith.edu
Finish Arduino Slides from Last Week…
Recap for I2C Communication

1. Get address of device to communicate with on I2C Bus

2. Get addresses of registers in device that contains the bytes (read or write)

3. Write **sketch** (Use the "Wire" library on Arduino, `#include <Wire.h>`)  

4. In loop() function:
   1. Initiate communication. Specify address of device to talk with
   2. Write or read to/from device
   3. Close communication
I2C Example

https://www.arduino.cc/en/Tutorial/MasterReader
Example 1
https://www.arduino.cc/en/Tutorial/MasterReader

On Mega 2560
- SDA = Pin 20
- SCL = Pin 21
Example 1

https://www.arduino.cc/en/Tutorial/MasterReader

```c
// Wire Master Reader
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Reads data from an I2C/TWI slave device
// Refer to the "Wire Slave Reader" example for use with this

// Created 29 March 2006
// This example code is in the public domain.

#include <Wire.h>

void setup() {
  Wire.begin();  // join i2c bus (address optional for master)
  Serial.begin(9600);  // start serial for output
}

void loop() {
  Wire.requestFrom(8, 6);  // request 6 bytes from slave device #8
  while (Wire.available()) { // slave may send less than requested
    char c = Wire.read(); // receive a byte as character
    Serial.print(c);  // print the character
  }
  // close communication
  delay(500);
}
Example 1

https://www.arduino.cc/en/Tutorial/MasterReader

```c
// Wire Slave Sender
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Sends data as an I2C/THI slave device
// Refer to the "Wire Master Reader" example for use with this

// Created 29 March 2006

// This example code is in the public domain.

#include <Wire.h>

void setup() {
  Wire.begin(8);    // join i2c bus with address #8
  Wire.onRequest(requestEvent); // register event
}

void loop() {
  delay(100);
}

// function that executes whenever data is requested by master
// this function is registered as an event, see setup()
void requestEvent() {
  Wire.write("hello "); // respond with message of 6 bytes
  // as expected by master
}
Modern Serial Interface: USB

- Started 1996 (USB 1.0)
- For data and power (5V)
- High Speed (1.5Mbits/s → 40GBits/s)
- Several Layers of Software
- Exchanges data in packets
- Uses some form of Acknowledge
Final Project

https://circuitdigest.com/microcontroller-projects/arduino-raspberry-pi-interfacing

D. Thiebaut, Computer Science, Smith College
Outline

• The Computer
  • History
  • Features
• Unboxing!
• Keyboard, Display, Mouse Connection
• Explore!
• Raspbian?
• How Fast?
• Internet Address
• Remote Connection
• Programming the GPIO
History

- First prototype: 2006
- Created by Eben Upton in the UK
- Revealed online in 2011
- 600,000+ views on YouTube in 2 days
- Upton, Director of studies at the University of Cambridge: "Designed to rekindle the curiosity about computing in a generation immersed in technology but indifferent to how it worked."

https://www.techrepublic.com/pictures/how-the-raspberry-pi-was-created-a-visual-history-of-the-35-board/
## Features

<table>
<thead>
<tr>
<th></th>
<th>Raspberry Pi 3 Model B</th>
<th>Raspberry Pi Zero</th>
<th>Raspberry Pi 2 Model B</th>
<th>Raspberry Pi Model B+</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction Date</strong></td>
<td>2/29/2016</td>
<td>11/25/2015</td>
<td>2/2/2015</td>
<td>7/14/2014</td>
</tr>
<tr>
<td><strong>SoC</strong></td>
<td>BCM2837</td>
<td>BCM2835</td>
<td>BCM2836</td>
<td>BCM2835</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>Quad Cortex A53 @ 1.2GHz</td>
<td>ARM11 @ 1GHz</td>
<td>Quad Cortex A7 @ 900MHz</td>
<td>ARM11 @ 700MHz</td>
</tr>
<tr>
<td><strong>Instruction set</strong></td>
<td>ARMv8-A</td>
<td>ARMv6</td>
<td>ARMv7-A</td>
<td>ARMv6</td>
</tr>
<tr>
<td><strong>GPU</strong></td>
<td>400MHz VideoCore IV</td>
<td>250MHz VideoCore IV</td>
<td>250MHz VideoCore IV</td>
<td>250MHz VideoCore IV</td>
</tr>
<tr>
<td><strong>RAM</strong></td>
<td>1GB SDRAM</td>
<td>512 MB SDRAM</td>
<td>1GB SDRAM</td>
<td>512MB SDRAM</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>micro-SD</td>
<td>micro-SD</td>
<td>micro-SD</td>
<td>micro-SD</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>10/100</td>
<td>none</td>
<td>10/100</td>
<td>10/100</td>
</tr>
<tr>
<td><strong>Wireless</strong></td>
<td>802.11n / Bluetooth 4.0</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td><strong>Video Output</strong></td>
<td>HDMI / Composite</td>
<td>HDMI / Composite</td>
<td>HDMI / Composite</td>
<td>HDMI / Composite</td>
</tr>
<tr>
<td><strong>Audio Output</strong></td>
<td>HDMI / Headphone</td>
<td>HDMI</td>
<td>HDMI / Headphone</td>
<td>HDMI / Headphone</td>
</tr>
<tr>
<td><strong>GPIO</strong></td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>$35</td>
<td>$5</td>
<td>$35</td>
<td>$35</td>
</tr>
</tbody>
</table>

[https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/](https://hackaday.com/2016/02/28/introducing-the-raspberry-pi-3/)
Features

- **Quad Core 1.2GHz** Broadcom BCM2837 64bit CPU
- **1GB RAM**
- **BCM43438 wireless LAN** and **Bluetooth Low Energy (BLE)** on board
- **100 Base Ethernet**
- **40-pin extended GPIO**
- **4 USB 2 ports**
- **4 Pole stereo output and composite video port**
- **Full size HDMI**
- **CSI camera** port for connecting a Raspberry Pi camera
- **DSI display port** for connecting a Raspberry Pi touchscreen display
- **Micro SD** port for loading your operating system and storing data
- Upgraded switched **Micro USB power** source up to 2.5 A

https://www.raspberry.pi.org/products/raspberry-pi-3-model-b/
ARM Processor

- **Quad Core** processor
- **1.2 GHz** speed
- **Floating-Point** Processor
- Supports *virtualization* in hardware
- **31** general purpose *registers*
- Hardware-assisted *cryptography*
RPi Block Diagram

http://doc.xdevs.com/doc/RPi/
Outline

- The Computer
  - History
  - Features
  - Unboxing!
  - Keyboard, Display, Mouse Connection
  - Explore!
  - Raspbian?
  - How Fast?
  - Internet Address
  - Remote Connection
  - Programming the GPIO
Unboxing
Putting Your Pi Together
1.3 BASIC SETUP
Fitting a Heat Sink
When the components in a computer system work hard, they generate heat. And above a certain level, this heat can reduce the lifespan of the components or even break them altogether. A “heat sink” is a carefully designed block of metal that takes the heat away from the electronic component and then passes it into the air surrounding the device.

There are two chips on a Raspberry Pi that can get very hot if the device is working hard: the central processing unit (1), and the chip that controls the Ethernet and USB ports (2).

Figure 2. The main heat-out points on a Pi (left); and installing a heat sink on the CPU (right)

To install a heat sink:
1. Unplug the Pi and leave it to cool before attempting to handle the device.
2. On the bottom of the heat sink, peel away the plastic backing that covers the adhesive.
3. Press the heat sink down firmly and directly onto the chip. Hold the pressure for a few seconds to allow the adhesive to work.
Putting Your Pi Together
If you buy heat sinks for your Pi, only use the thermal adhesive that they arrive with; never use any other type of adhesive or sticky plastic to install a heat sink on a Raspberry Pi. The adhesive must be a special compound so that it effectively transfers heat from the chip on the Pi to the metal of the heat sink.

**Connecting Power**

Power is fed to the Pi through the micro-USB socket or, if you have a suitable connector, through the general purpose input output pins. However, providing power through the GPIO header pins bypasses the on-board protection circuitry that is designed to prevent damage to the device. For this reason, it should only be attempted by people who are experienced in building electronic circuits.

**Connecting a Display**

HDMI offers a high-quality video and audio signal, and is the preferred way of connecting all models of Raspberry Pi to a modern television. To connect a high-definition television:

- Plug one end of an HDMI cable into the Raspberry Pi’s HDMI socket, and the other end into an HDMI input on your TV.

If your display does not support HDMI, you can use the composite video and audio outputs to the auxiliary A/V input of most other televisions. These connections are colored yellow, red, and white.

![Figure 3. Connecting a display to the Raspberry Pi](image)
Turning **ON** the Raspberry **Pi**

1. Display
2. Keyboard
3. Mouse
4. Charger
Turning off the Raspberry Pi

DO NOT
STOP THE PI
BY CUTTING
OFF POWER
Turning Off the Pi

Option 1

```
sudo halt -p
```

Option 2
2.4 RASPBIAN’S DESKTOP ENVIRONMENT

Raspbian’s desktop is similar to Microsoft Windows and many of the ways that you use it are the same. To interact with icons and buttons, click them with the mouse. If you right-click an icon, Raspbian displays a context menu. The options in a context menu only apply to the item that you clicked. The desktop takes up most of the screen. This is where your programs will appear when you start them. And if you save files to the `/home/pi/Desktop` directory then you can see links to these files appear on the desktop.

When Raspbian’s graphical desktop is running, you can access the raspi-config tool at any time by clicking LXTerminal on your desktop. Then type the following command and press Enter: `sudo raspi-config`

In the top left of the screen, you can find: the Raspbian menu, the Application Launcher, and the Wastebasket.
Time to Explore!
How many games? Many

Is there a Web browser? Yeet!

Can you write a "Hello World" program in C? Yes

What programming languages are supported? Perl, Python

Does it support C? C++? Yes

What productivity tools are included? Yes

What is inkspill.py? Yes

Can you write a "Hello World" program in Java? Yes

Can you write a "Hello World" program in Python? Yes
1: Stop Linux
2: Unplug
1: Start it again!
• The Computer
  • History
  • Features
• Unboxing!
• Keyboard, Display, Mouse Connection
• Explore!
  • Raspbian?
• How Fast?
• Internet Address
• Remote Connection
• Programming the GPIO
What's Raspbian?

- Operating System
- Free

- Rasp ← Raspberry Pi  
  bian ← Debian

- Based on Debian, Linux Kernel, optimized for RPi
- over 35,000 packages available for download (avail since 2012)
- Not affiliated with Raspberry Pi Foundation
- https://www.raspbian.org/
How Do We Get Raspbian?


2. **Unzip** the file downloaded (7-zip, Unarchiver, unzip)

2. **Write** disk image to microSD card (Win32 Disk Imager, dd)

3. **Put microSD in Raspberry Pi**, and go!

https://thepi.io/how-to-install-raspbian-on-the-raspberry-pi/
How Do We Get Raspbian?


2. **Unzip** the file downloaded (7-zip, Unarchiver, unzip)

3. **Write** disk image to microSD card (Win32 Disk Imager, dd)

4. **Put microSD in Raspberry Pi**, and go!

https://thepi.io/how-to-install-raspbian-on-the-raspberry-pi/
Good References

  
  • **apt-get**: Section 6.2
  
  • Configuring for a **different language**: Chapter 8
  
  • Managing **Users**: Section 8.4
  
  • Remote login: remote desktop (**vnc**), **ssh**: Chapter 9
  
  • **Web server**: Section 11.2
• *Lynda's Video Tutorials*: [https://www.linkedin.com/learning/](https://www.linkedin.com/learning/)
Continuing our Exploration...
How Fast?

```bash
pi@raspberrypi:~$ mkdir 270
pi@raspberrypi:~$ cd 270
pi@raspberrypi:~/270$ ls
pi@raspberrypi:~/270$ wget http://www.science.smith.edu/~dthiebaut/classes/112/queensdemo.c
Resolving www.science.smith.edu (www.science.smith.edu)... unresolved
Connecting to www.science.smith.edu (www.science.smith.edu):80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4307 (4.2K) [text/x-csrc]
Saving to: 'queensdemo.c'
queensdemo.c 100%[======================================] 4.21K --.-KB/s in 0.001s
2019-03-30 11:58:16 (6.35 MB/s) -- 'queensdemo.c' saved [4307/4307]

pi@raspberrypi:~/270$ ls
queensdemo.c
pi@raspberrypi:~/270$ gcc -o queensdemo queensdemo.c
pi@raspberrypi:~/270$ ./queensdemo 20
solution 0: 0 2 4 1 3 12 14 11 17 19 16 8 15 18 7 9 6 13 5 10
N=20 solution found!
20x20
pi@raspberrypi:~/270$ gcc -03 -o queensdemo queensdemo.c
pi@raspberrypi:~/270$ ./queensdemo 20
solution 0: 0 2 4 1 3 12 14 11 17 19 16 8 15 18 7 9 6 13 5 10
N=20 solution found!
20x20
pi@raspberrypi:~/270$ 
```
How Fast? Using wget

wget http://www.science.smith.edu/~dthiebaut/classes/112/queensdemo.c

or copy/paste from your browser
<table>
<thead>
<tr>
<th>NxN</th>
<th>Macbook Pro 2009</th>
<th>Macbook Pro 2014</th>
<th>MacPro 2009</th>
<th>MacPro 2014</th>
<th>Linux Mint Beowulf2</th>
<th>Laptop 1</th>
<th>Linux 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9x9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10x10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11x11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12x12</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13x13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14x14</td>
<td>26</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>17</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>15x15</td>
<td>17</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>16x16</td>
<td>37</td>
<td>14</td>
<td>27</td>
<td>18</td>
<td>46</td>
<td>1</td>
<td>33</td>
</tr>
<tr>
<td>17x17</td>
<td>28</td>
<td>10</td>
<td>19</td>
<td>13</td>
<td>31</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>18x18</td>
<td>87</td>
<td>34</td>
<td>67</td>
<td>41</td>
<td>104</td>
<td>5</td>
<td>60</td>
</tr>
<tr>
<td>19x19</td>
<td>19</td>
<td>7</td>
<td>13</td>
<td>9</td>
<td>22</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>20x20</td>
<td>144</td>
<td>64</td>
<td>119</td>
<td>70</td>
<td>152</td>
<td>24</td>
<td>99</td>
</tr>
<tr>
<td>21x21</td>
<td>36</td>
<td>13</td>
<td>26</td>
<td>17</td>
<td>50</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>22x22</td>
<td>750</td>
<td>311</td>
<td>662</td>
<td>384</td>
<td>610</td>
<td>221</td>
<td>463</td>
</tr>
<tr>
<td>23x23</td>
<td>81</td>
<td>31</td>
<td>63</td>
<td>39</td>
<td>101</td>
<td>3</td>
<td>58</td>
</tr>
<tr>
<td>24x24</td>
<td>242</td>
<td>101</td>
<td>199</td>
<td>119</td>
<td>226</td>
<td>56</td>
<td>157</td>
</tr>
<tr>
<td>25x25</td>
<td>92</td>
<td>37</td>
<td>72</td>
<td>44</td>
<td>111</td>
<td>7</td>
<td>64</td>
</tr>
<tr>
<td>26x26</td>
<td>248</td>
<td>102</td>
<td>201</td>
<td>121</td>
<td>227</td>
<td>58</td>
<td>161</td>
</tr>
<tr>
<td>27x27</td>
<td>284</td>
<td>118</td>
<td>232</td>
<td>142</td>
<td>257</td>
<td>69</td>
<td>184</td>
</tr>
<tr>
<td>28x28</td>
<td>1551</td>
<td>633</td>
<td>1250</td>
<td>734</td>
<td>1213</td>
<td>469</td>
<td>938</td>
</tr>
<tr>
<td>29x29</td>
<td>1518</td>
<td>356</td>
<td>687</td>
<td>413</td>
<td>684</td>
<td>244</td>
<td>522</td>
</tr>
<tr>
<td>30x30</td>
<td>33667</td>
<td>12586</td>
<td>23558</td>
<td>14158</td>
<td>22889</td>
<td>9041</td>
<td>17252</td>
</tr>
</tbody>
</table>
• The Computer
  • History
  • Features
• Unboxing!
• Keyboard, Display, Mouse Connection
• Explore!
• Raspbian?
• How Fast?
• Internet Address
• Remote Connection
• Programming the GPIO
RPi Internet Address

- *In terminal:* `/sbin/ifconfig`

- *dyndns address:* `270x.is-a-geek.net`

- *In terminal:* `sudo ddclient -force`
Remote Connection to the RPi

- **VNC Viewer**  (*virtual network computing*)
- **SSH/Putty**  (*secure shell/Putty*)

(Both will need Pulse Secure if connecting from off campus!)
Turn On VNC Server on RPi

- In Terminal:

    `sudo raspi-config`
VNC Server: 270a.is-a-geek.net
Name: RaspberryPi 270a

Enter a VNC Server address or search

Authentication

Enter the credentials expected by VNC Server running on the remote computer. Note these are not your RealVNC account credentials.

VNC Server: 270a.is-a-geek.net::5900 (TCP)
Username: pi
Password: ********

Signature: 23-54-4b-af-8b-8f-9c-58

OK Cancel
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Sat Mar 30 16:57:21 2019 from pi@raspberrypi:~ $
Try Both Connection Options!

https://eltechs.com/raspberry-pi-nas-guide/
<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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3.3V</td>
<td></td>
<td>5V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDA1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>15</td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td>GND</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>Clock</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>17</td>
<td>11</td>
<td>10</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>27</td>
<td>13</td>
<td>12</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>22</td>
<td>15</td>
<td>16</td>
<td>18</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>10</td>
<td>19</td>
<td>18</td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>9</td>
<td>21</td>
<td>16</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>11</td>
<td>23</td>
<td>18</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0</td>
<td>27</td>
<td>26</td>
<td>28</td>
<td>11</td>
</tr>
<tr>
<td>Clock</td>
<td>21</td>
<td>5</td>
<td>29</td>
<td>28</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>Clock</td>
<td>22</td>
<td>6</td>
<td>31</td>
<td>32</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>PWM</td>
<td>23</td>
<td>13</td>
<td>33</td>
<td>33</td>
<td>32</td>
<td>26</td>
</tr>
<tr>
<td>PWM</td>
<td>24</td>
<td>19</td>
<td>35</td>
<td>34</td>
<td>24</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>26</td>
<td>37</td>
<td>35</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GND</td>
<td></td>
<td>36</td>
<td>36</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>31</td>
<td></td>
<td>37</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td></td>
<td>39</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access
### 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></td>
<td></td>
<td>3.3V</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></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td></td>
<td>8</td>
<td>13</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>17</td>
<td>11</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>22</td>
<td>15</td>
<td></td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>10</td>
<td>19</td>
<td></td>
<td>18</td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>9</td>
<td>21</td>
<td></td>
<td>22</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>11</td>
<td>23</td>
<td></td>
<td>24</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></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></td>
<td>28</td>
<td>1</td>
<td>31</td>
</tr>
<tr>
<td>Clock</td>
<td>21</td>
<td>5</td>
<td>29</td>
<td></td>
<td>22</td>
<td>6</td>
<td>31</td>
</tr>
<tr>
<td>Clock</td>
<td>22</td>
<td>6</td>
<td>31</td>
<td></td>
<td>32</td>
<td>12</td>
<td>26</td>
</tr>
<tr>
<td>PWM</td>
<td>23</td>
<td>13</td>
<td>33</td>
<td></td>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWM</td>
<td>24</td>
<td>19</td>
<td>35</td>
<td></td>
<td>36</td>
<td>16</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>26</td>
<td>37</td>
<td></td>
<td>38</td>
<td>20</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
<td>21</td>
<td>29</td>
<td></td>
</tr>
</tbody>
</table>

[Source](https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access)
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 and is intended for use by experienced C/C++ programmers. It is not a newbie learning tool.

WiringPi is developed directly on a Raspberry Pi running 32-bit Raspbian. I do not support any other platform, cross compiling or operating systems. It has been ported to other platforms, other operating systems and some are cross compiling, however this author does not maintain those systems. If you are trying to use wiringPi on a platform other than the Raspberry Pi with Raspbian then you must contact the person who did the port and not me.
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/
// 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;
}
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) [source]
```

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**

<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></td>
<td></td>
<td>3.3V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDA1</td>
<td>8</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>5V</td>
<td></td>
</tr>
<tr>
<td>SCL1</td>
<td>9</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td>5V</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>3.3V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>17</td>
<td>10</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>27</td>
<td>12</td>
<td>18</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>11</td>
<td></td>
<td></td>
<td>PWM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>13</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>14</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>16</td>
<td>23</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>24</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>22</td>
<td>25</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>23</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28</td>
<td>1</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>32</td>
<td>12</td>
<td>26</td>
<td>PWM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GND</td>
<td>36</td>
<td>16</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>38</td>
<td>20</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
<td>21</td>
<td>29</td>
<td>Clock</td>
</tr>
</tbody>
</table>

[https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access](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)
You are Ready For Lab #9!