Process Monitoring using the Raspberry Pi

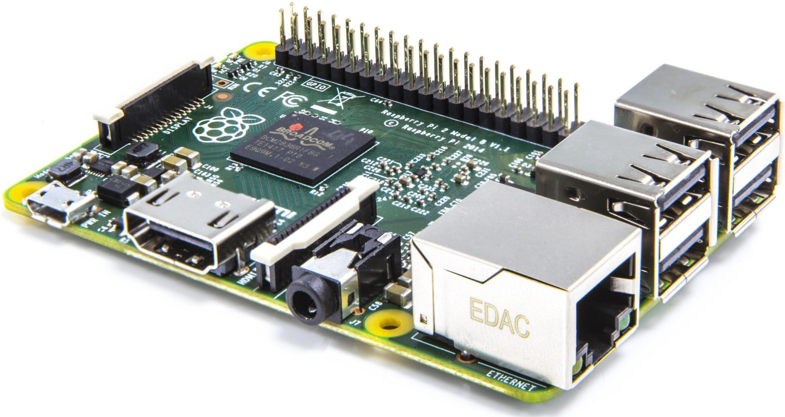
Ron Wright

University of Illinois at Urbana-Champaign

Raspberry Pi

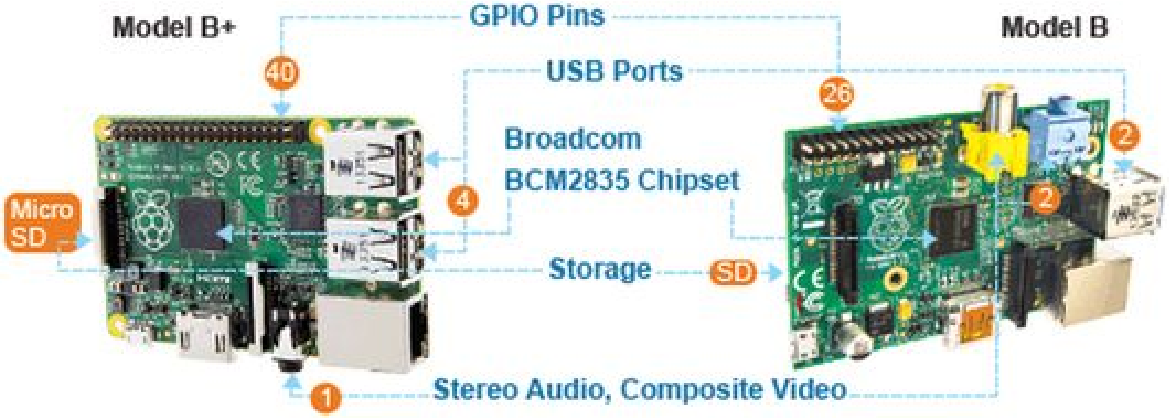
- A credit card-sized computer that can be used in numerous applications
- Typically used for embedded systems projects and educational purposes
- Applications:
  - Robotics
  - Home automation
  - Remote sensing
  - Music and multimedia

RaspberryPi



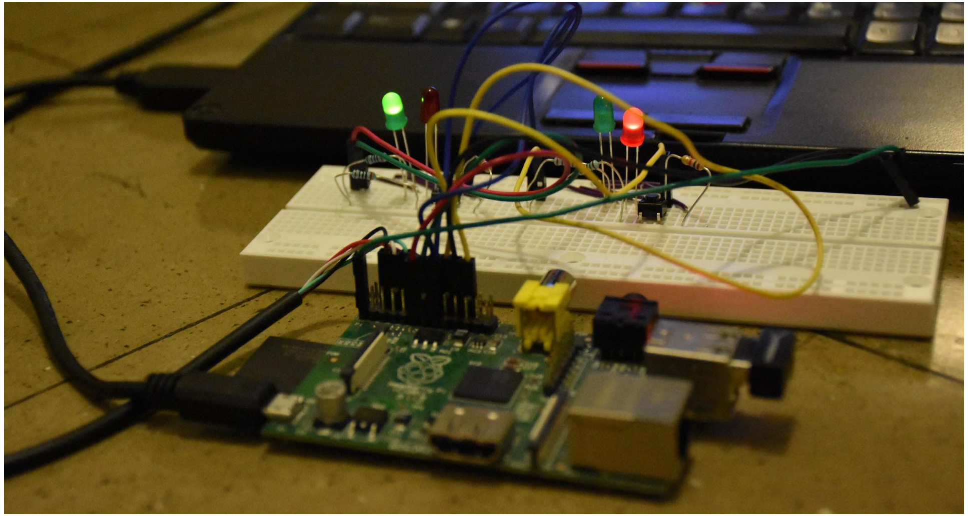
Most Common Versions

- Raspberry Pi 1 Model B
- Raspberry Pi 1 Model B+
- Raspberry Pi 2 Model B



Hardware and Software

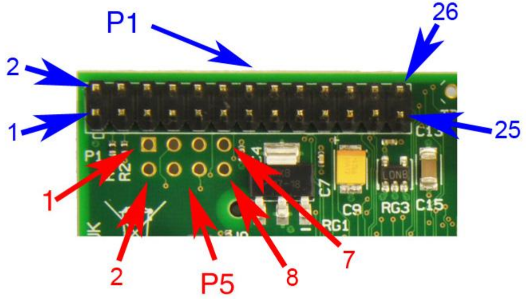
- Hardware is external to the Raspberry Pi and is connected to GPIO pins
- Software runs on the Raspberry Pi and can be implemented in any language of your choice
    - Python
    - C/C++
- Most common operating system: Raspbian GNU/Linux (a Debian derivative)



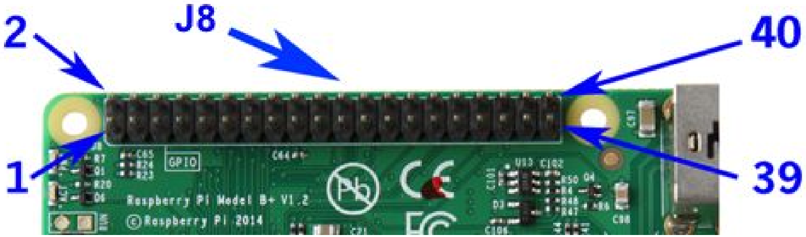
GPIO Pins

- Used to interface programs with external devices
- Read information from switches, buttons, sensors, or other devices
- Write information to LEDs, displays, controllers, or other devices
- Each GPIO pin is explicitly set to input or set to output in the program

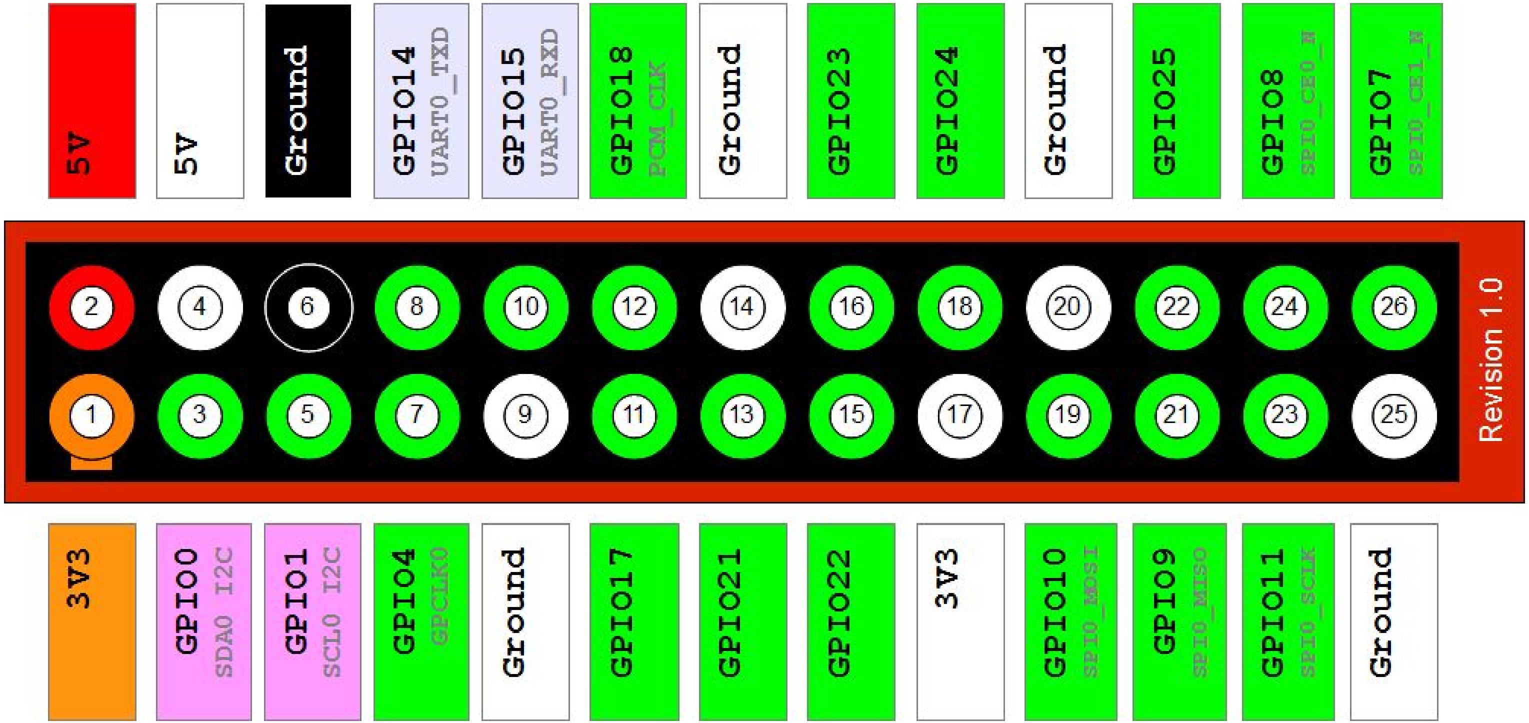
Raspberry Pi 1 Model B:



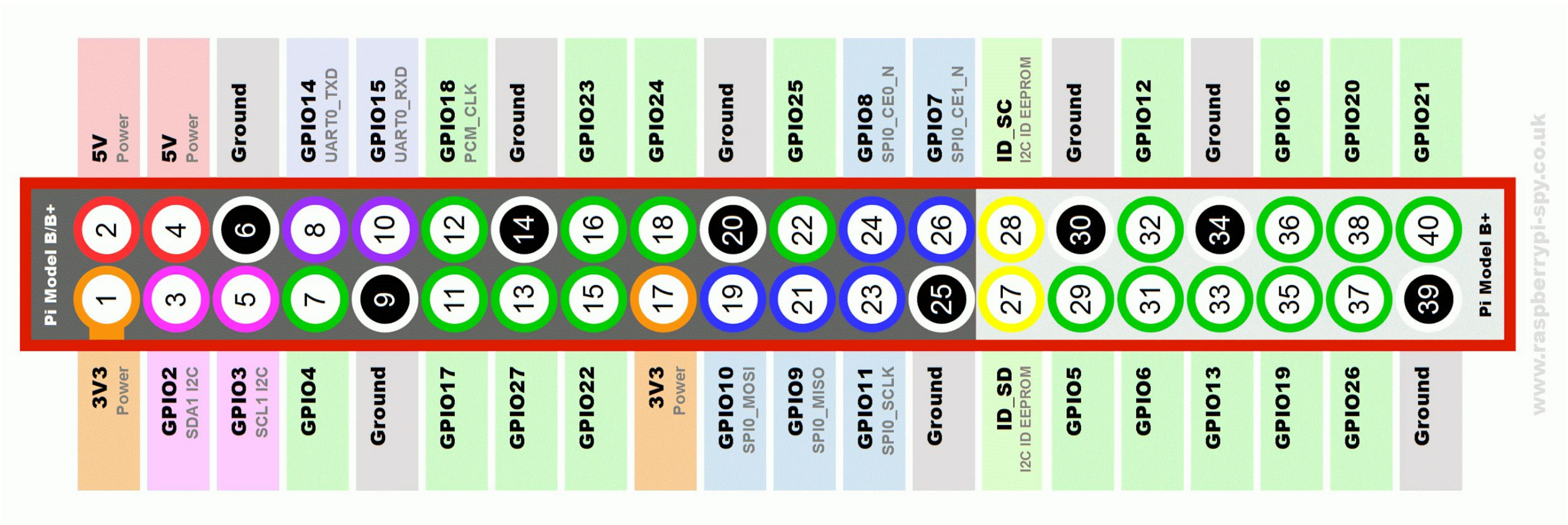
Raspberry Pi 1 Model B+/Raspberry Pi 2:



GPIO Pins for v1 Model B



GPIO Pins for v1 Model B+/v2 Model B



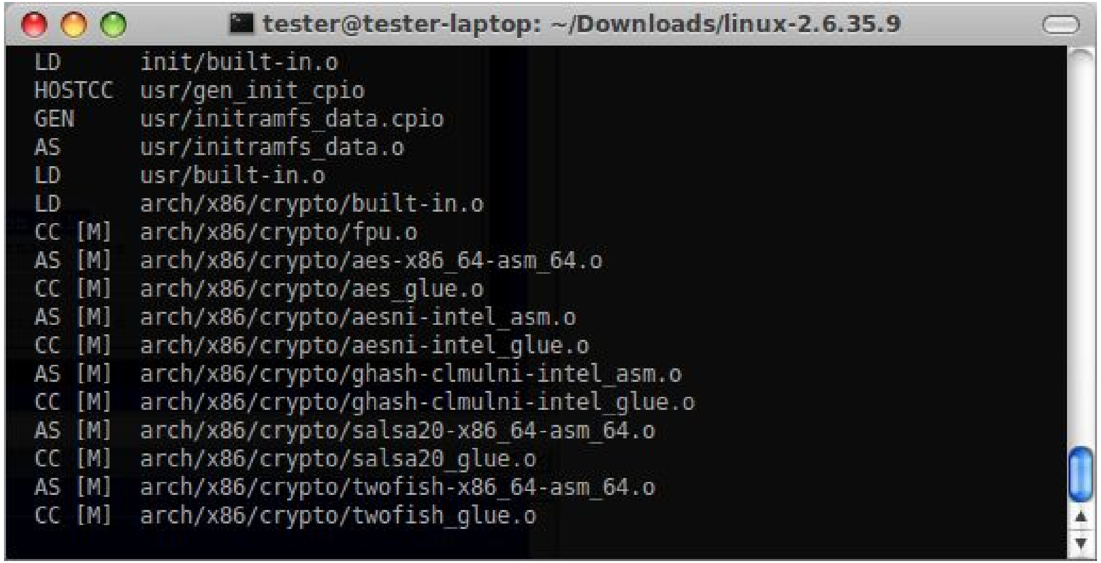
	5V Power	5V Power	Ground	GPIO14 UART0_TXD	GPIO15 UART0_RXD	GPIO18 PCM_CLK	Ground	GPIO23	GPIO24	Ground	GPIO25	GPIO8 SPI0_CE0_N	GPIO7 SPI0_CE1_N	ID_SC I2C ID EEPROM	Ground	GPIO12	Ground	GPIO16	GPIO20	GPIO21	
Pi Model B/B+	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	Pi Model B+
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39	
	3V3 Power	GPIO2 SDA1 I2C	GPIO3 SCL1 I2C	GPIO4	Ground	GPIO17	GPIO27	GPIO22	3V3 Power	GPIO10 SPI0_MOSI	GPIO9 SPI0_MISO	GPIO11 SPI0_SCLK	Ground	ID_SD I2C ID EEPROM	GPIO5	GPIO6	GPIO13	GPIO19	GPIO26	Ground	

Process Monitoring System

- A client/server system
  - The client is a Python wrapper that does the following:
    - Executes the given command
    - Sends the task’s total CPU usage to the server running on the Raspberry Pi
    - Sends the task’s status code to the server once it finishes
  - The server is a Python program that does the following:
    - Flashes a green LED when it sees that a program is running
    - Turns the green LED on without flashing when it sees that the program finished and has successfully executed
    - Turns the red LED on without flashing when it sees that the program finished but has unsuccessfully executed
    - Flash rate is directly proportional to the program’s CPU usage
    - Button resets whatever LED is solid

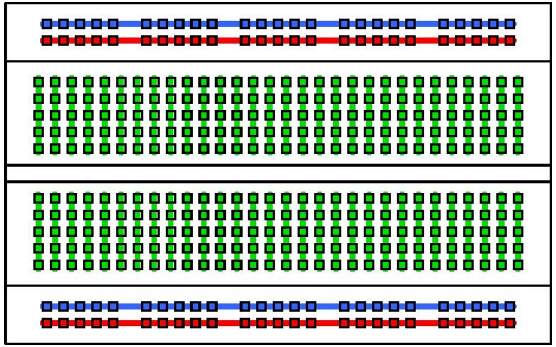
Process Monitoring System

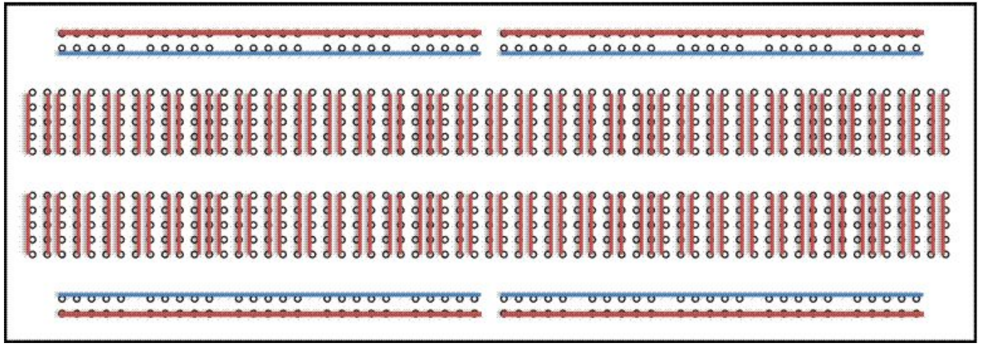
- Useful to check if a process finished simply by looking across the room at the LEDs
- Applications
  - Building a large program (such as the Linux kernel)
  - Running a large computation or long simulation



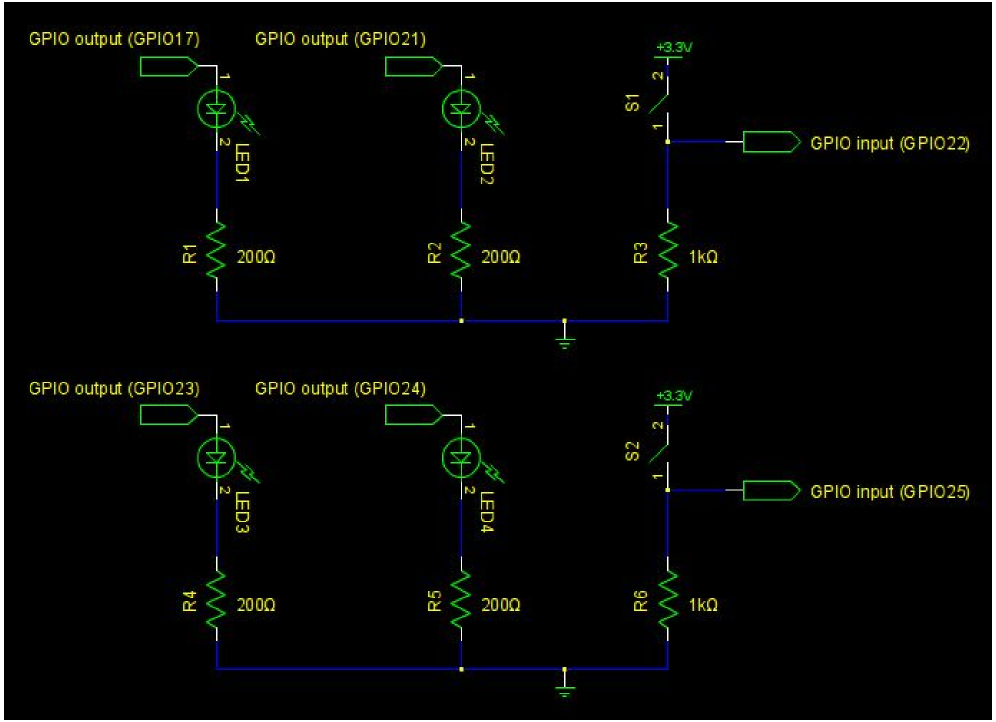
```
  LD      init/built-in.o
  HOSTCC  usr/gen_init_cpio
  GEN     usr/initramfs_data.cpio
  AS      usr/initramfs_data.o
  LD      usr/built-in.o
  LD      arch/x86/crypto/built-in.o
  CC [M]  arch/x86/crypto/fpu.o
  AS [M]  arch/x86/crypto/aes-x86_64-asm_64.o
  CC [M]  arch/x86/crypto/aes_glue.o
  AS [M]  arch/x86/crypto/aesni-intel_asm.o
  CC [M]  arch/x86/crypto/aesni-intel_glue.o
  AS [M]  arch/x86/crypto/ghash-clmulni-intel_asm.o
  CC [M]  arch/x86/crypto/ghash-clmulni-intel_glue.o
  AS [M]  arch/x86/crypto/salsa20-x86_64-asm_64.o
  CC [M]  arch/x86/crypto/salsa20_glue.o
  AS [M]  arch/x86/crypto/twofish-x86_64-asm_64.o
  CC [M]  arch/x86/crypto/twofish_glue.o
```

Common Breadboard Layouts

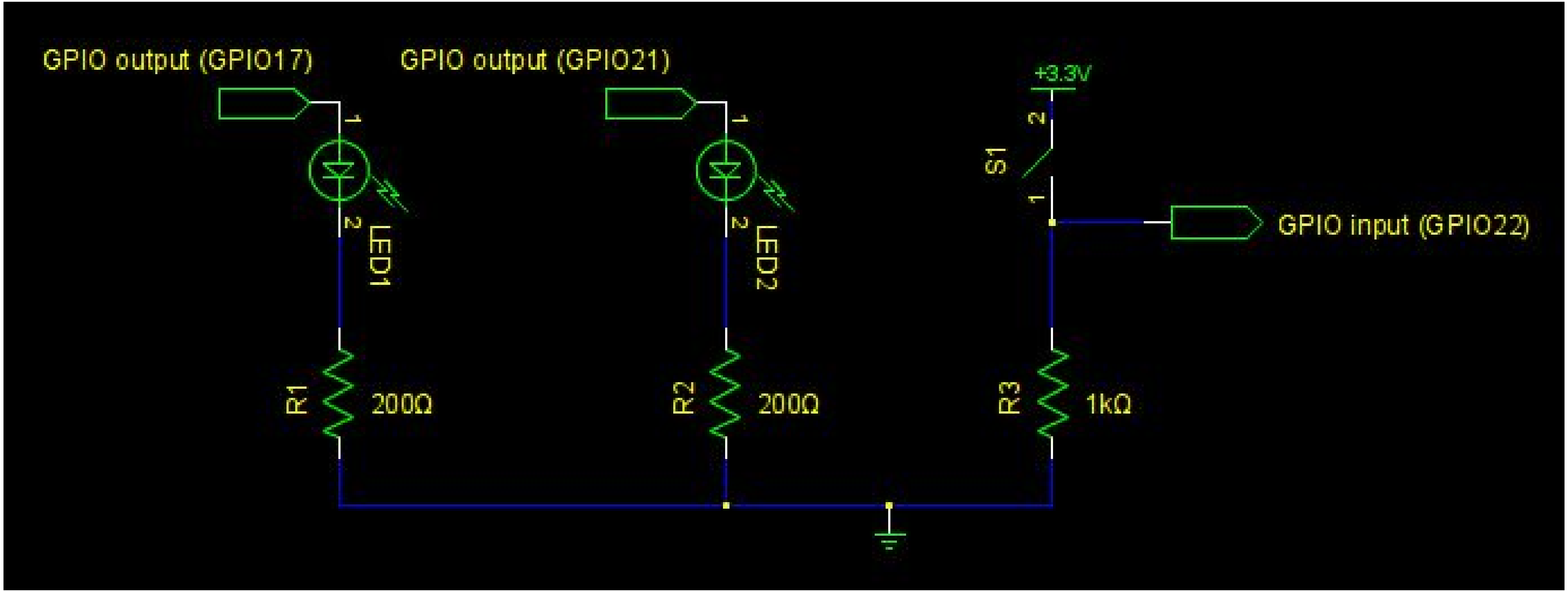




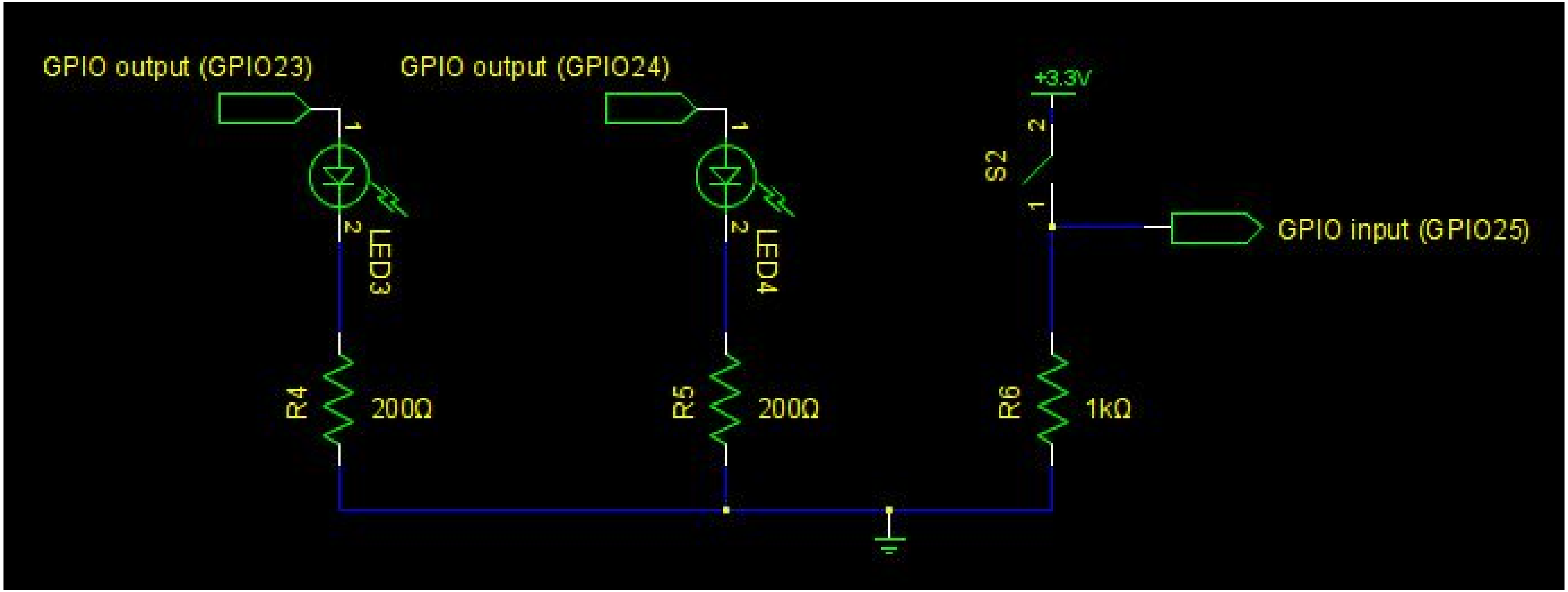
LED and Button Schematic



LED and Button Schematic (Top Half)



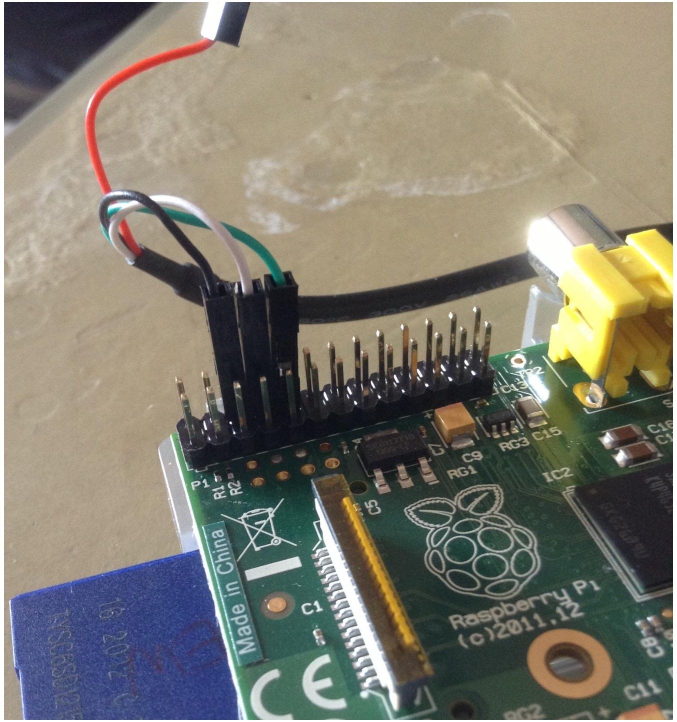
LED and Button Schematic (Bottom Half)



Accessing the Raspberry Pi

- Using a USB to serial cable

- Over network connection using SSH



Demo

Get the Code (and the Slides)!

The code and the slides for this workshop is hosted on GitHub.

http://github.com/logiconcepts819/program-status-server

Some useful notes:

- Code for the server (to be run on Raspberry Pi) can be found in “server” folder
- Code for the client (i.e., the wrapper to be run on a laptop or PC) can be found in “wrapper” folder

Thank you!