RPi General Purpose Input/Output Reference

The Raspberry Pi 1 Model B is a small computer with an ARMv6 processor (ARM11 family). As mentioned in the introduction section, the SoC is a Broadcom BCM2835[1] and it consists of CPU (ARM1176JZFs), *Graphical Processing Unit* (GPU), *Digital Signal Processor* (DSP), SDRAM (512 MB shared with GPU), a single USB port direct from the BCM2835 and a 10/100 Ethernet connector. Figure 1 illustrates these components. The ARM11176JFZs is based on version 6 of the ARM architecture, the GPU is a low-power mobile multimedia processor architecture and the DSP is a specialised microprocessor with an architecture optimised for digital signal processing operations.There are two models of Raspberry Pi (model A and model B), which have different specifications.

The Raspberry Pi B model's SoC has a total of 54 GPIO pins, however only the first 26 are easily accessible. Each pin can be individually enabled or disabled, and configured as an input or output. GPIO inputs are readable (high = 1, low = 0) and GPIO outputs are both readable and writable. Each GPIO pin can be accessed and controlled by modifying control bits in specific memory locations. The following table illustrates the mapping between the GPIO pins and the physical memory address for the relevant pins[2].

Physical address	byte 4	byte 3	byte 2	byte 1	GPIO pins
`0x2020 0008`	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx	20-29
`0x2020 0004`	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx	10-19
`0x2020 0000`	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx	0-9

Each physical address contains three control bits per pin, leaving the last two bits (31-30) reserved. For example, at physical address `0x2020 0008`, you can access the following pins:

31	30	29 28 27	26 25 24	23 22 21	20 19 18	17 16 15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
x	x	29	28	27	26	25	24	23	22	21	20

For example pin 20 has 3 control bits (at positions 0-2). The control bits set the functionality of the pin, writing bit pattern `000` sets the pin as an input pin, writing pattern `001` makes it an output pin. Therefore to set pin 20 as an output, we can write the bit pattern 001 into bits 0-2[3].

Once a pin has been set up as an output, you then have to write a bit to a different memory address to turn the output pin on or off. There are two 4-byte memory addresses that control the output state of the first 32 GPIO pins. The 32-bits starting at `0x2020 0028` control clearing the pins, and the 32 bits starting at `0x2020 001C` control setting (turning on) the pins.

You will always need to clear a pin, by writing to the *clear* area, before you can set it. If you were to write a 1 into bit 20 at address `0x2020 0028` you would clear pin 20. To subsequently turn pin 20 on, you would write a 1 into bit 20 at address `0x2020 001C`. Note that writing a 0 bit has no effect, so it is safe to write `0x0010 0000` into the addresses to only change the state of pin 20. You can assume that pins are wired to LEDs, so any pin can turn a LED on when the pin is on and turn a LED off when the pin is *clear*. Note also that the GPIO memory addresses (`0x2020 008`, `0x2020 0004`, `0x2020 0000`) can contain any value, as any pin can have a high or low input applied. Therefore, you should assume that reads to GPIO memory addresses return the value of the address that was read. i.e. a read of memory address `0x2020 0004` will return `0x2020 0004`.

[1] On CATe for this course you will find a document called `SoC_ARM_peripherals.pdf` where you can find all the hardware information about the SoC

[2] Note that the physical address "`0x2020 ####`" is equivalent to the bus address "`0x7E## ####`" in the hardware information PDF

[3] However think carefully about how to do this, if you were to write 0x00000000 to the four byte address 0x2020008, you will set *all* the above pins as inputs

<50mA	3.3V	1	2	5V	
GPIO 02	SDA 0/1	3	4	NC	
GPIO 03	SCL 0/1	5	6	GROUND	
GPIO 04		7	8	UART TXD	GPIO 14
	GROUND	9	10	UART RXD	GPIO 15
GPIO 17		11	12		GPIO 18
GPIO 27		13	14	GROUND	
GPIO 22		15	16		GPIO 23
	3.3V	17	18		GPIO 24
GPIO 10	SP10(MOSI)	19	20	GROUND	
GPIO 9	SP10(MISO)	21	22		GPIO 25
GPIO 11	SP10(SCKL)	23	24		SP10(CE0)
	GROUND	25	26		SP10(CE1)

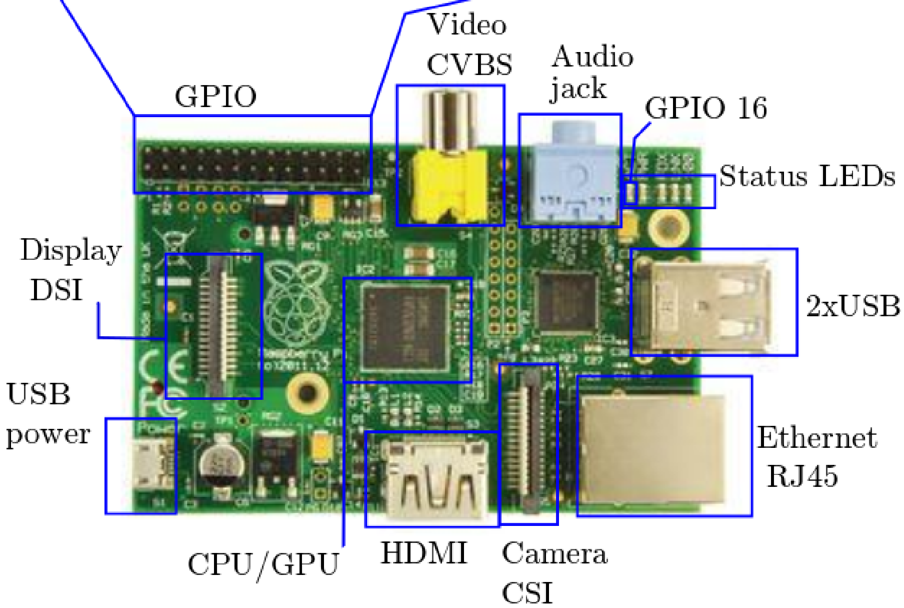

Figure 1: Raspberry PiB model (Revision 2) with GPIO pin-outs.