General Purpose Input/Output (GPIO)

Jalal Kawash

Outline

1. The Raspberry Pi
2. Introduction to the RPi GPIO
3. Working with GPIO lines

Section 1

The Raspberry Pi

Section 1 Objective

At the end of this section you will

1. Be familiar with the Raspberry Pi


Ultra II
2.0GB

RASPBERRY PI MODEL B

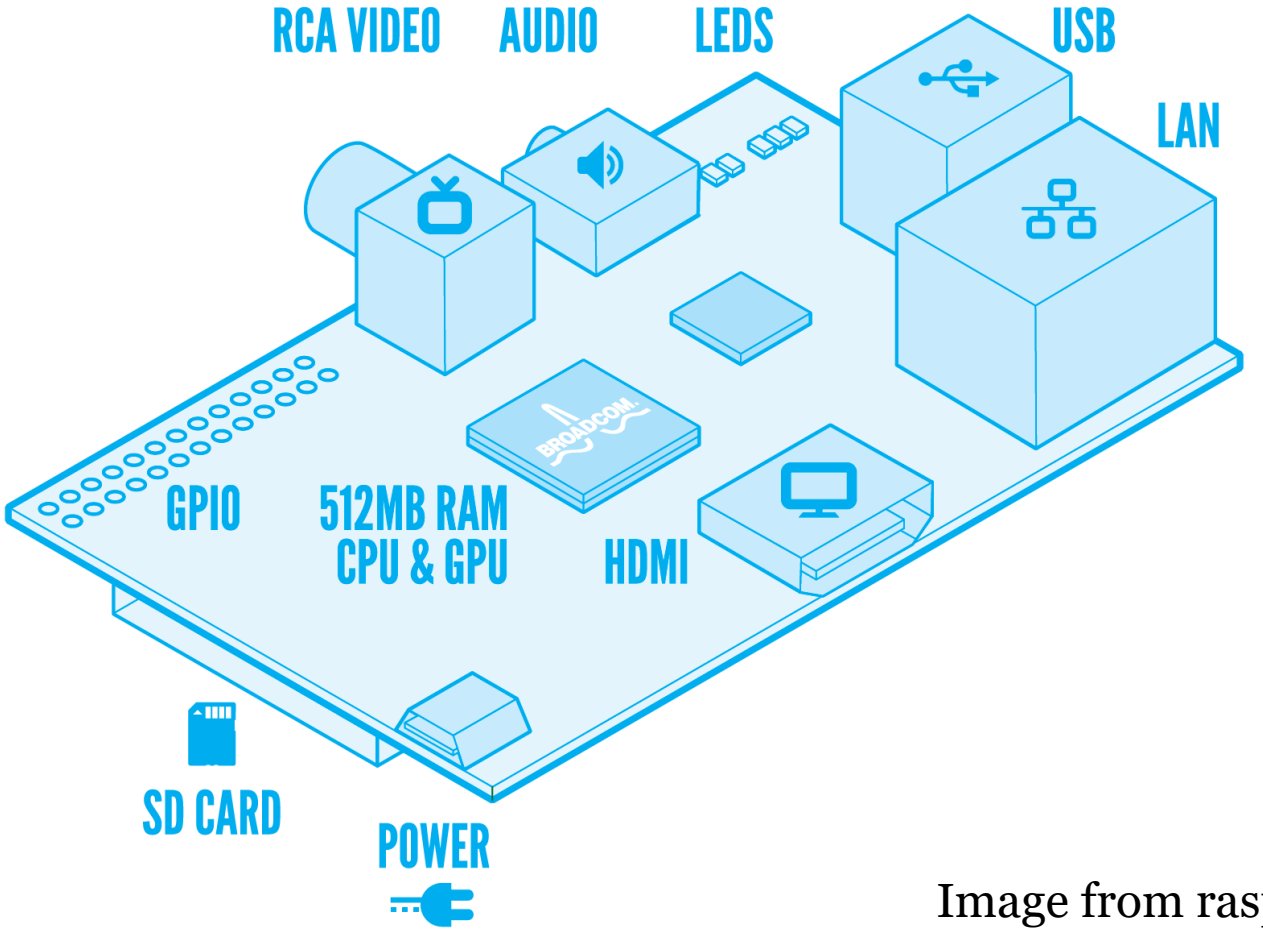
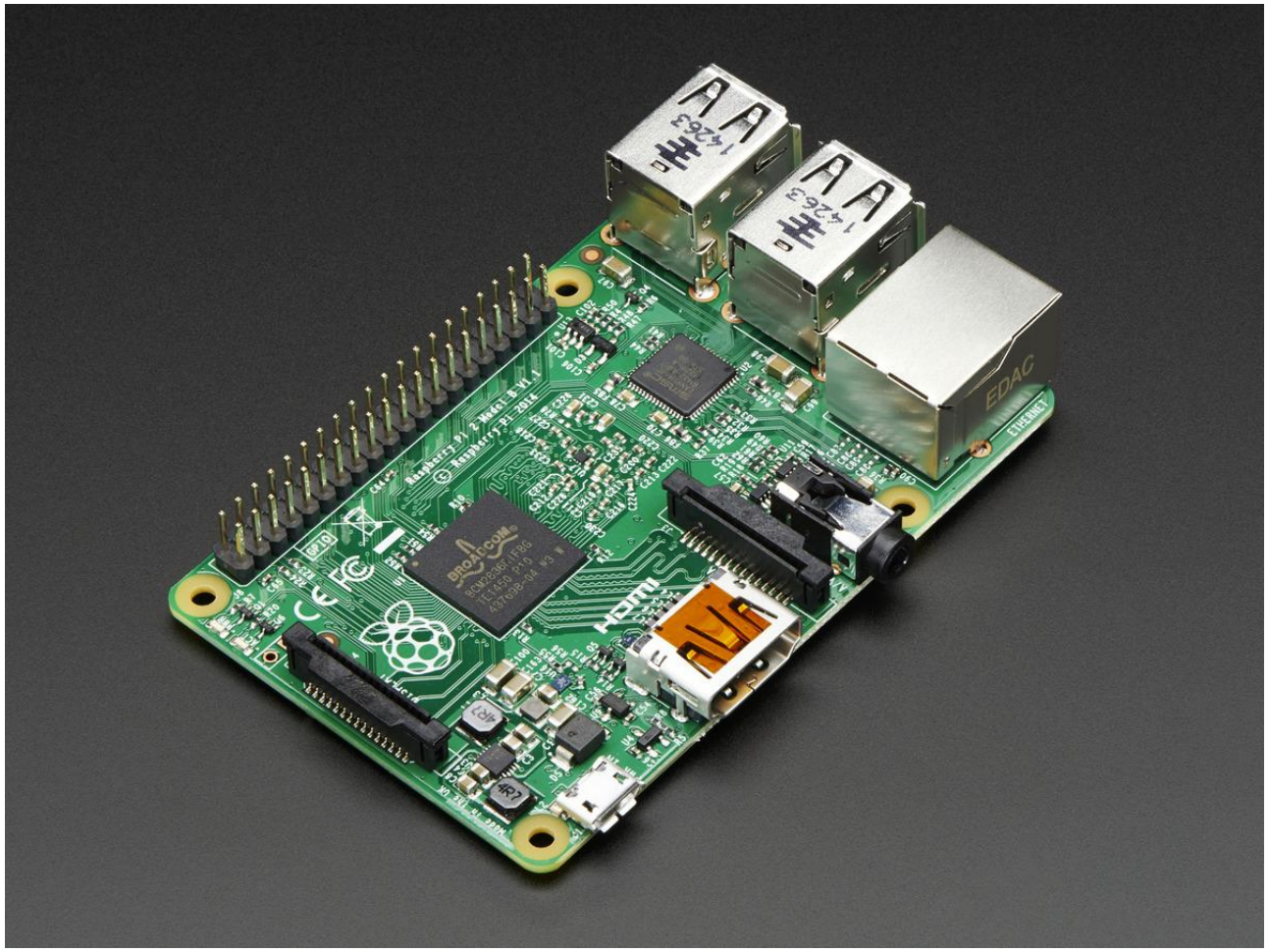


Image from raspberrypi.org

Technical Features	
Chip	Broadcom BCM2835 SoC full HD multimedia applications processor
CPU	700 MHz Low Power ARM1176JZ-F Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	512MB SDRAM
Ethernet	onboard 10/100 Ethernet RJ45 jack
USB 2.0	Dual USB Connector
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI
Onboard Storage	SD, MMC, SDIO card slot
Operating System	Linux
Dimensions	8.6cm x 5.4cm x 1.7cm

From element14.com

Raspberry Pi 2



Technical Specs

- 900 MHz **ARMv7 Quad Core Processor** (Cortex-A7)
- 1GB RAM
- **40 pin** extended GPIO
- 4 x USB 2 ports
- CSI camera port (Raspberry Pi camera)
- DSI display port (Raspberry Pi touch screen)

From element14.com

Section 2

Introduction to the Raspberry Pi GPIO

Section 2 Objectives

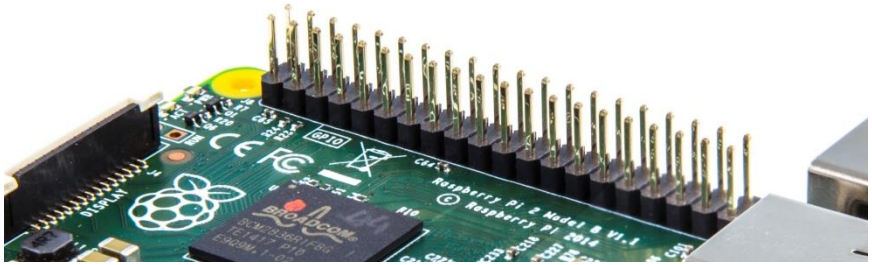
At the end of this section you will

1. Get a quick look at the RPi GPIO pins
2. Look at example uses of the GPIO

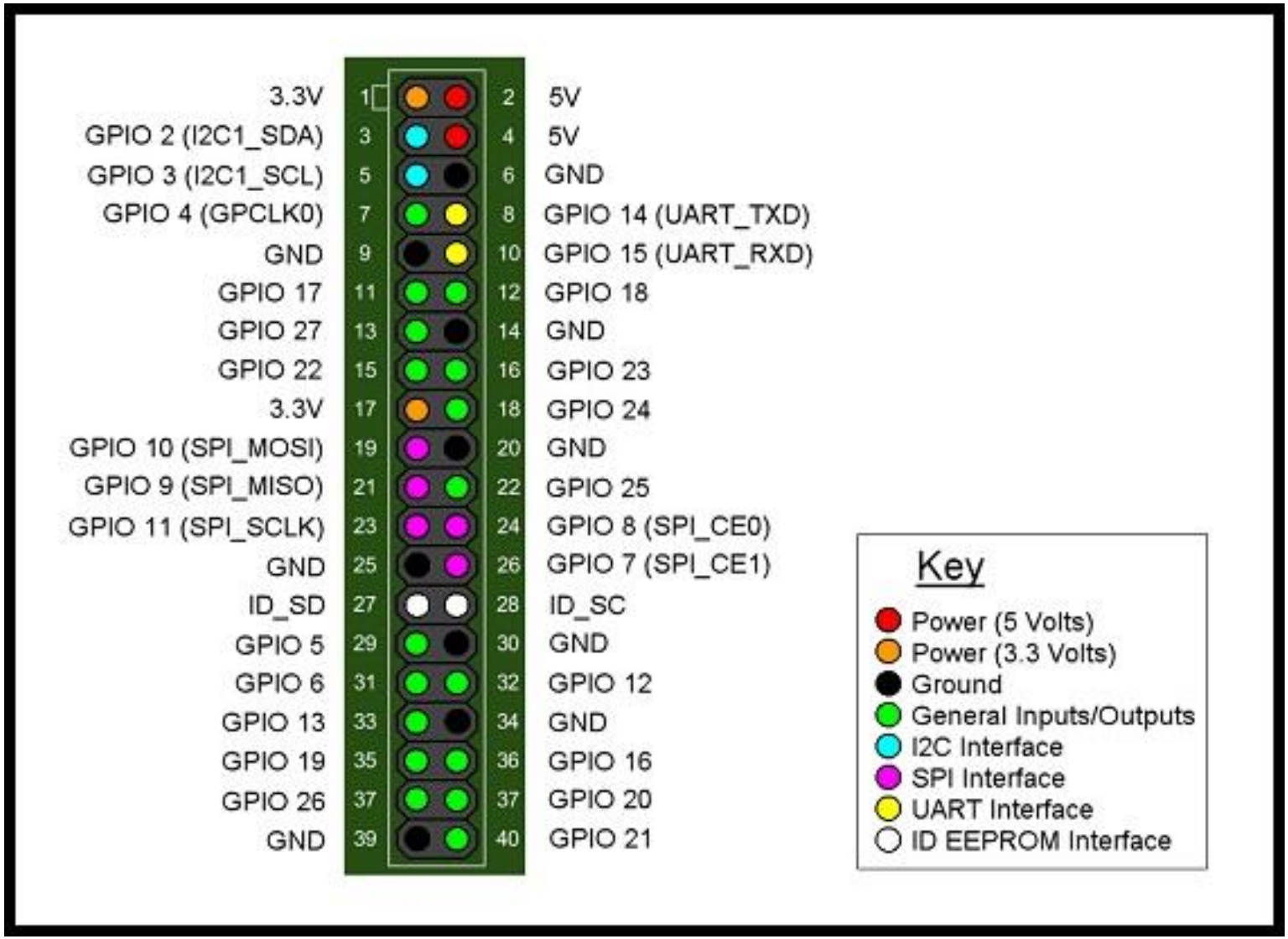
GPIO

- General Purpose Input/Output device
- Famous with microcontrollers
  - Provide extended functions without changing circuitry
- Has 54 lines
  - Only a subset is available for the RPi
    - 26 lines in Pi1
    - 40 lines in Pi2

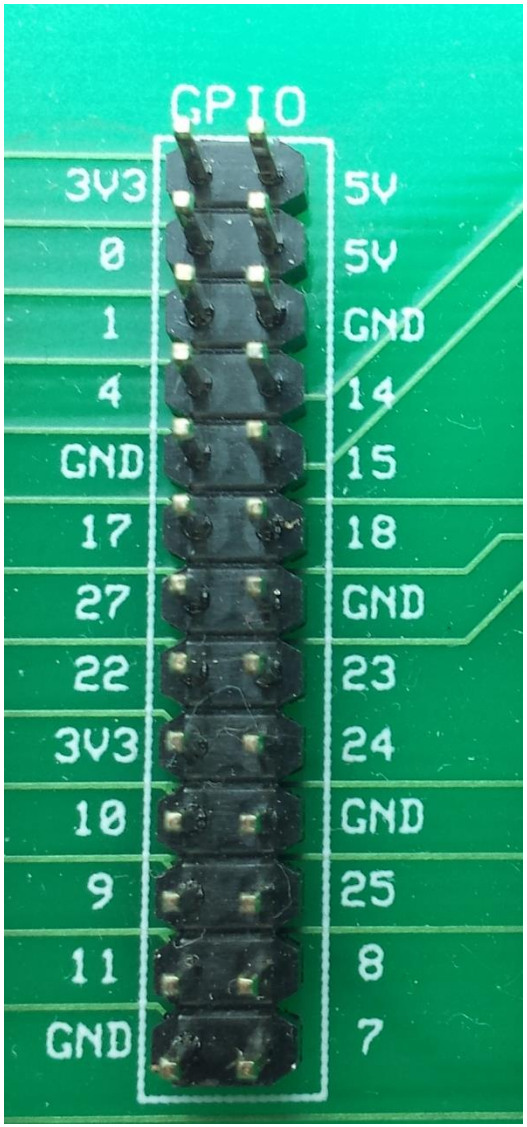
RPi GPIO

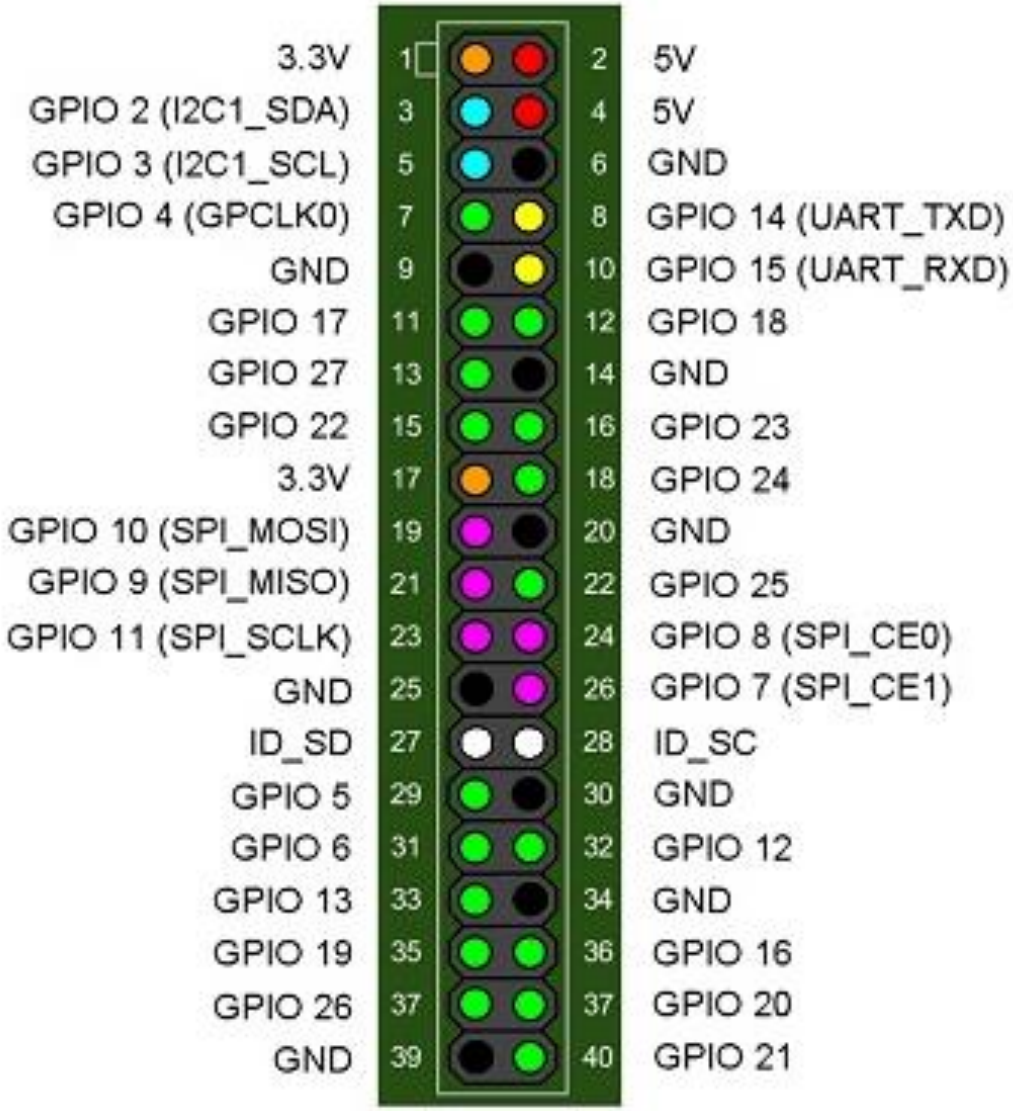


- Programmable
- 40 header pins on the RPi
- Provide UART and 5V power
- Provide others too e.g. CSI (camera serial interface) and DSI (Display serial interface)
- <u>Not plug and play</u>

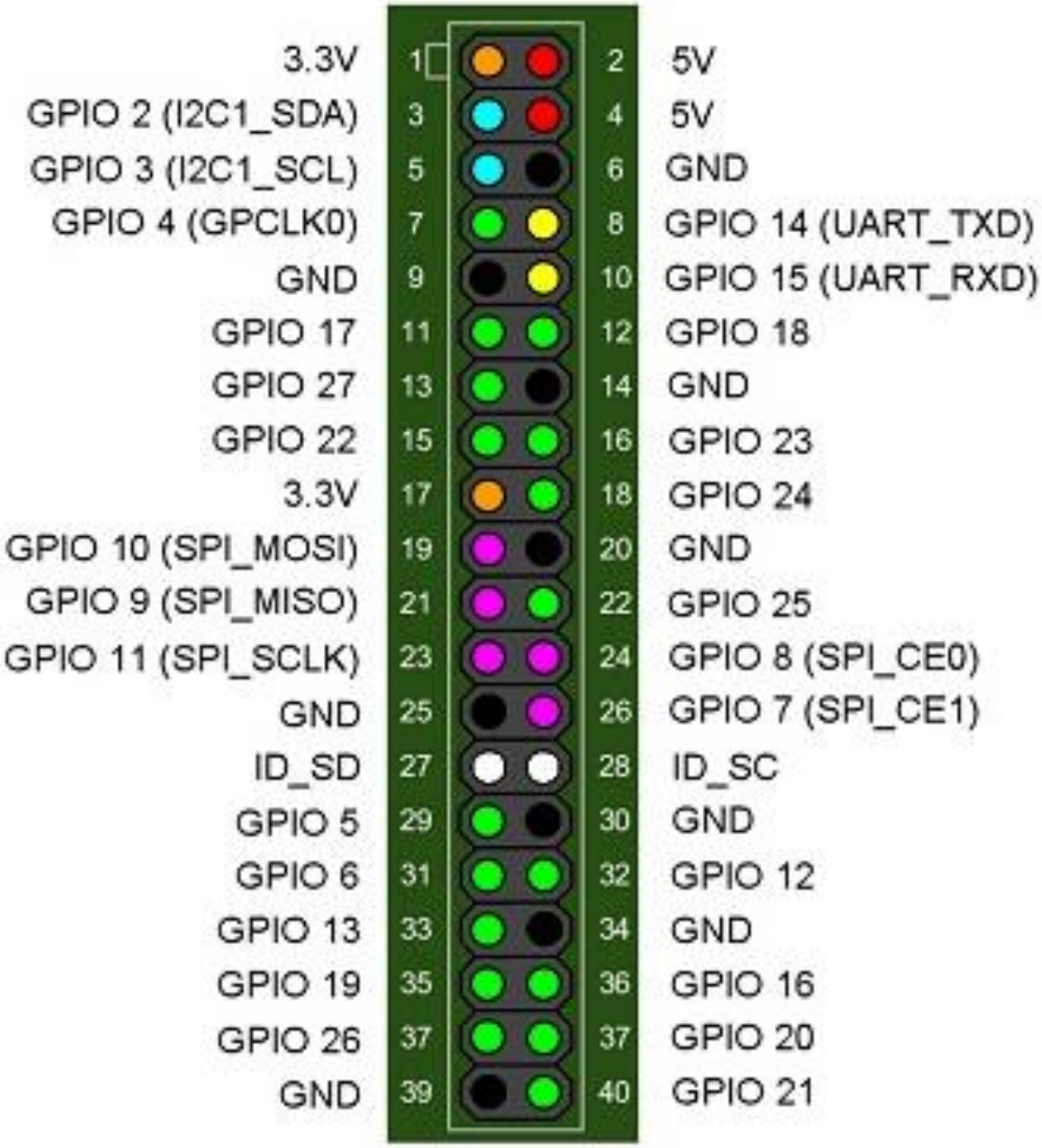

3.3V 1
2 5V
GPIO 2 (I2C1_SDA) 3
4 5V
GPIO 3 (I2C1_SCL) 5
6 GND
GPIO 4 (GPCLK0) 7
8 GPIO 14 (UART_TXD)
GND 9
10 GPIO 15 (UART_RXD)
GPIO 17 11
12 GPIO 18
GPIO 27 13
14 GND
GPIO 22 15
16 GPIO 23
3.3V 17
18 GPIO 24
GPIO 10 (SPI_MOSI) 19
20 GND
GPIO 9 (SPI_MISO) 21
22 GPIO 25
GPIO 11 (SPI_SCLK) 23
24 GPIO 8 (SPI_CE0)
GND 25
26 GPIO 7 (SPI_CE1)
ID_SD 27
28 ID_SC
GPIO 5 29
30 GND
GPIO 6 31
32 GPIO 12
GPIO 13 33
34 GND
GPIO 19 35
36 GPIO 16
GPIO 26 37
37 GPIO 20
GND 39
40 GPIO 21
Key
Power (5 Volts)
Power (3.3 Volts)
Ground
General Inputs/Outputs
I2C Interface
SPI Interface
UART Interface
ID EEPROM Interface

www.ebay.com


GPIO
3V3
0
1
4
GND
17
27
22
3V3
10
9
11
GND
5V
5V
GND
14
15
18
GND
23
24
GND
25
8
7


3.3V 1
GPIO 2 (I2C1_SDA) 3
GPIO 3 (I2C1_SCL) 5
GPIO 4 (GPCLK0) 7
GND 9
GPIO 17 11
GPIO 27 13
GPIO 22 15
3.3V 17
GPIO 10 (SPI_MOSI) 19
GPIO 9 (SPI_MISO) 21
GPIO 11 (SPI_SCLK) 23
GND 25
ID_SD 27
GPIO 5 29
GPIO 6 31
GPIO 13 33
GPIO 19 35
GPIO 26 37
GND 39
2 5V
4 5V
6 GND
8 GPIO 14 (UART_TXD)
10 GPIO 15 (UART_RXD)
12 GPIO 18
14 GND
16 GPIO 23
18 GPIO 24
20 GND
22 GPIO 25
24 GPIO 8 (SPI_CE0)
26 GPIO 7 (SPI_CE1)
28 ID_SC
30 GND
32 GPIO 12
34 GND
36 GPIO 16
37 GPIO 20
40 GPIO 21

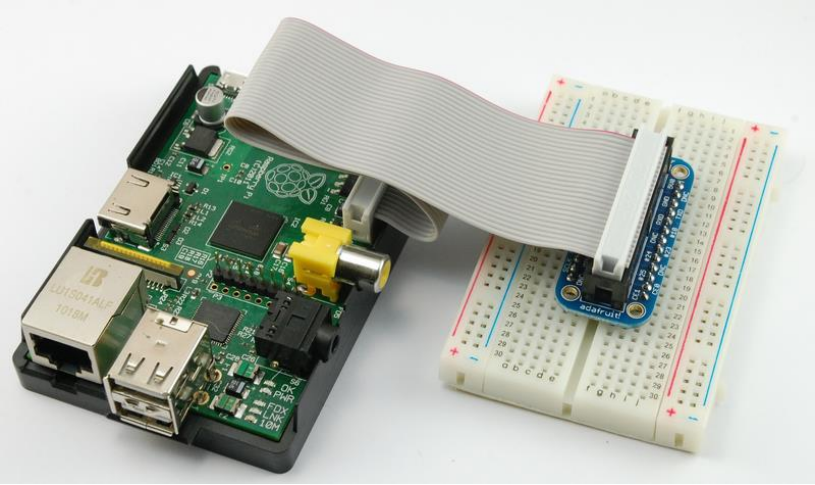
ebay.com

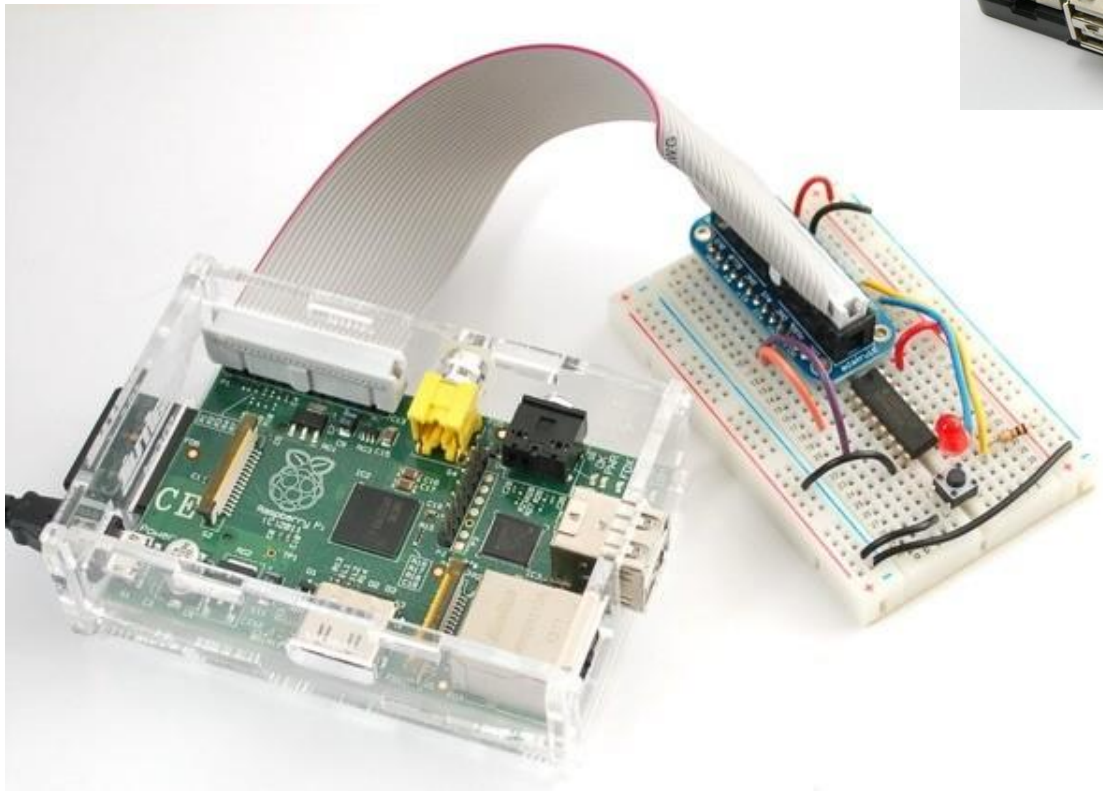

3.3V 1
2 5V
GPIO 2 (I2C1_SDA) 3
4 5V
GPIO 3 (I2C1_SCL) 5
6 GND
GPIO 4 (GPCLK0) 7
8 GPIO 14 (UART_TXD)
GND 9
10 GPIO 15 (UART_RXD)
GPIO 17 11
12 GPIO 18
GPIO 27 13
14 GND
GPIO 22 15
16 GPIO 23
3.3V 17
18 GPIO 24
GPIO 10 (SPI_MOSI) 19
20 GND
GPIO 9 (SPI_MISO) 21
22 GPIO 25
GPIO 11 (SPI_SCLK) 23
24 GPIO 8 (SPI_CE0)
GND 25
26 GPIO 7 (SPI_CE1)
ID_SD 27
28 ID_SC
GPIO 5 29
30 GND
GPIO 6 31
32 GPIO 12
GPIO 13 33
34 GND
GPIO 19 35
36 GPIO 16
GPIO 26 37
37 GPIO 20
GND 39
40 GPIO 21

Example Uses

Image from: www.raspberrypi.org

Example Uses



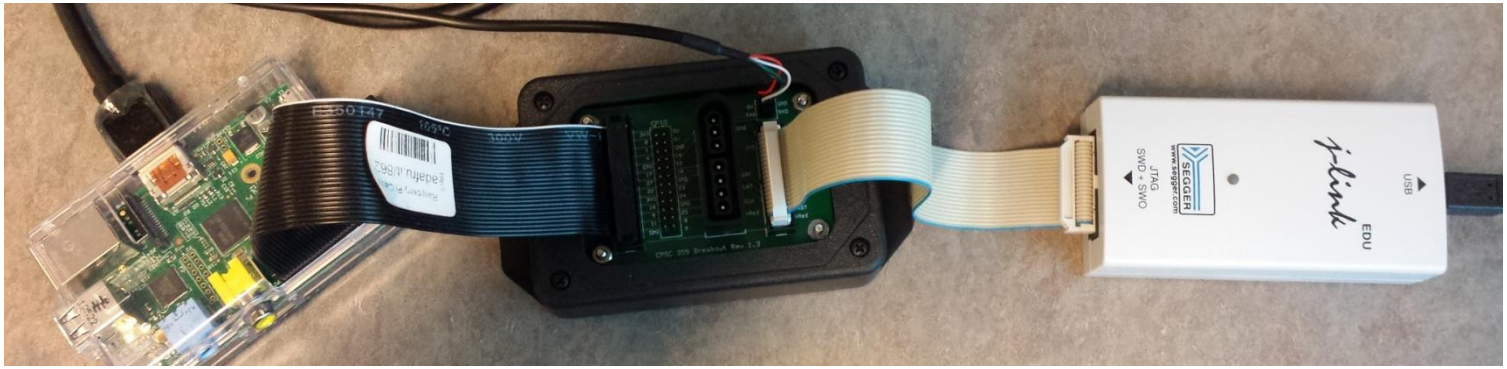


Images from: learn.adafruit.com & kiwi.psnc.pl

Example Uses

Image from: learn.adafruit.com

Example Uses



Example Uses

Section 3

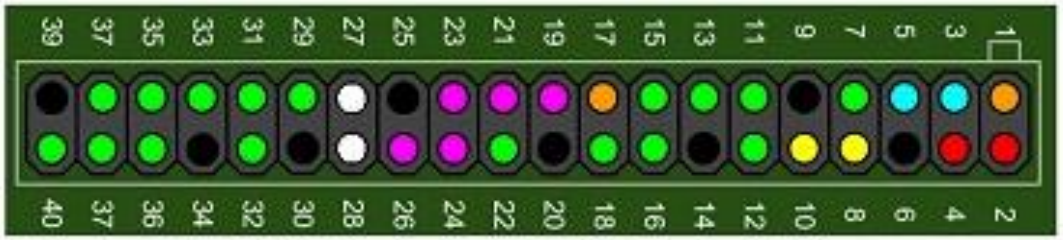
Working with GPIO Lines

Section 3 Objectives

At the end of this section you will

1. Have a closer look at GPIO registers
2. Know how to set functions to GPIO pins
3. Know how to use the pins for input and output

RPi GPIO pins

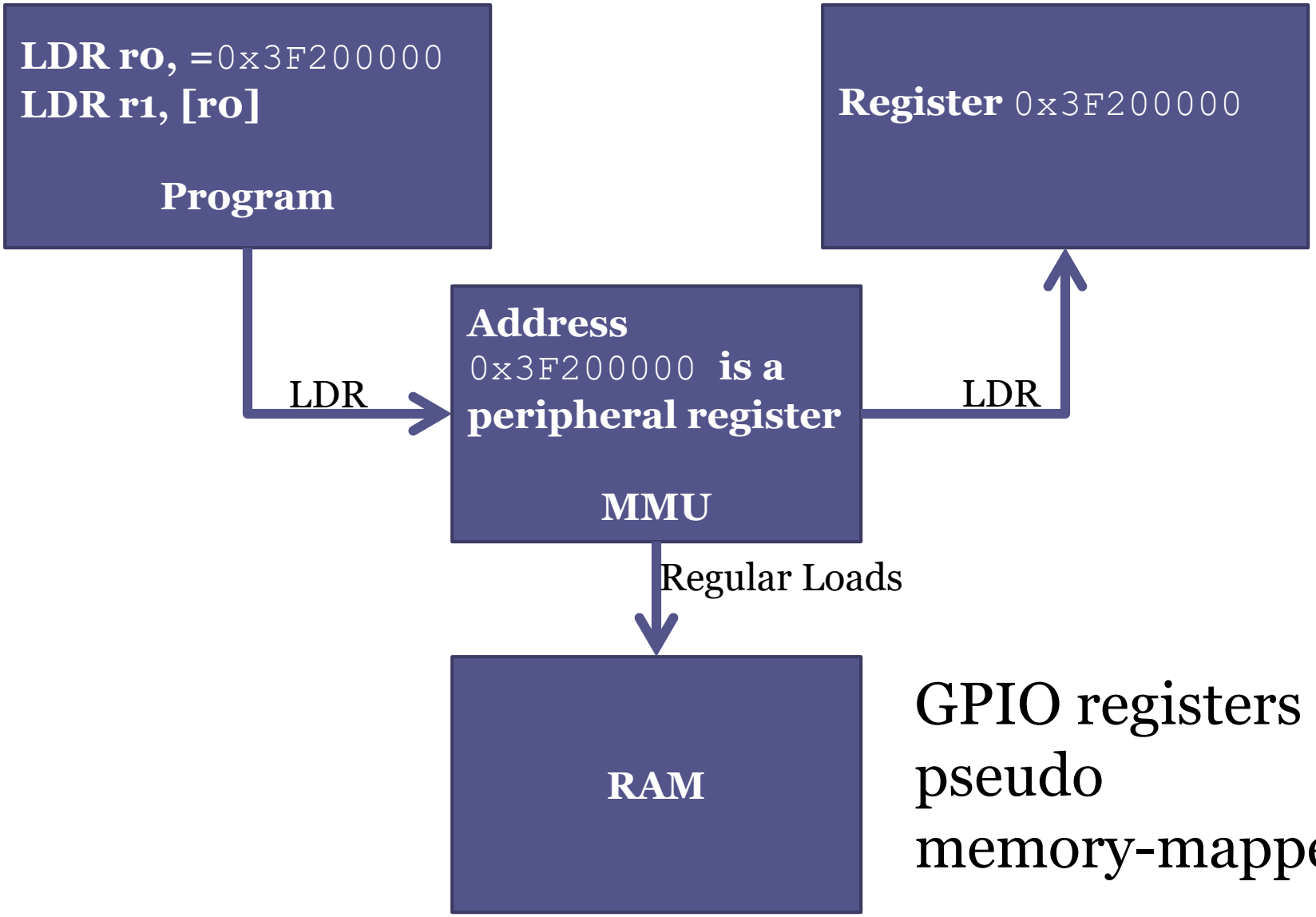


- The RPi GPIO has 16 GP pins
  - Green ones
- Since they are not plug-and-play, some work is need to initialize, write, and read
- Each of these pins can have up to 6 functions
  - For example function 1 is write & 0 is read
    - From core's point of view

Pseudo Memory Mapped Registers

- A memory-mapped register is dealt with as a memory address
  - It is accessed by LDR & STR
- A pseudo memory-mapped register is an actual peripheral register that is accessed as a memory location
  - MMU deals with these requests
  - MMU sends the request to the device rather than memory

Pseudo Memory Mapped Registers



GPIO registers are pseudo memory-mapped

GPIO Registers

Address	Field Name	Description	Size	Read/ Write
0x 7E20 0000	GPFSEL0	GPIO Function Select 0	32	R/W
0x 7E20 0004	GPFSEL1	GPIO Function Select 1	32	R/W
0x 7E20 0008	GPFSEL2	GPIO Function Select 2	32	R/W
0x 7E20 000C	GPFSEL3	GPIO Function Select 3	32	R/W
0x 7E20 0010	GPFSEL4	GPIO Function Select 4	32	R/W
0x 7E20 0014	GPFSEL5	GPIO Function Select 5	32	R/W
0x 7E20 0018	-	Reserved	-	-
0x 7E20 001C	GPSET0	GPIO Pin Output Set 0	32	W
0x 7E20 0020	GPSET1	GPIO Pin Output Set 1	32	W
0x 7E20 0024	-	Reserved	-	-
0x 7E20 0028	GPCLR0	GPIO Pin Output Clear 0	32	W
0x 7E20 002C	GPCLR1	GPIO Pin Output Clear 1	32	W
0x 7E20 0030	-	Reserved	-	-
0x 7E20 0034	GPLEV0	GPIO Pin Level 0	32	R
0x 7E20 0038	GPLEV1	GPIO Pin Level 1	32	R
0x 7E20 003C	-	Reserved	-	-

Function Select

Set (Write 1)

Clear (Write 0)

Level (Read)

More info refer to: BCM2835 ARM Peripherals (available on BB)

GPIO Registers

- There are more registers
- Refer to: BCM2835 ARM Peripherals (available on BB) for more info

- This document lists virtual addresses (in Linux) for registers
- Replace 7E in the virtual address by 3F to obtain the physical address

Function Select Registers

0x3F200014	GPFSEL5	For pins 50 to 53
0x3F200010	GPFSEL4	For pins 40 to 49
0x3F20000c	GPFSEL3	For pins 30 to 39
0x3F200008	GPFSEL2	For pins 20 to 29
0x3F200004	GPFSEL1	For pins 10 to 19
0x3F200000	GPFSEL0	For pins 0 to 9

Base GPIO address

Accessing GPIO Registers

- GPIO uses "pseudo" memory-mapped I/O
  - Accessed as if accessing memory locations

```
unsigned int *gpio = 0x3F200000;

#define    GPFSEL0          0
…
unsigned int v;
…
v = gpio[GPFSEL0];
gpio[GPFEL0] = v;
```

Pin Offset from GPIO Base Address

- A register corresponds to many GPIO pins
- For example

GPFSEL0

For pins 0 to 9

- Each pin (0 to 9) is controlled by 3 bits in GPFSEL0
- For example, bits 0-2 set the function for GPIO pin 0
- Each pin can be given a function (from up to 6 functions)
- In our case, it is mostly input or output functions

Inside GPFSEL0

Bit(s)	Field Name	Description	Type	Reset
31-30	---	Reserved	R	0
29-27	FSEL9	FSEL9 - Function Select 9 000 = GPIO Pin 9 is an input 001 = GPIO Pin 9 is an output 100 = GPIO Pin 9 takes alternate function 0 101 = GPIO Pin 9 takes alternate function 1 110 = GPIO Pin 9 takes alternate function 2 111 = GPIO Pin 9 takes alternate function 3 011 = GPIO Pin 9 takes alternate function 4 010 = GPIO Pin 9 takes alternate function 5	R/W	0
26-24	FSEL8	FSEL8 - Function Select 8	R/W	0
23-21	FSEL7	FSEL7 - Function Select 7	R/W	0
20-18	FSEL6	FSEL6 - Function Select 6	R/W	0
17-15	FSEL5	FSEL5 - Function Select 5	R/W	0
14-12	FSEL4	FSEL4 - Function Select 4	R/W	0
11-9	FSEL3	FSEL3 - Function Select 3	R/W	0
8-6	FSEL2	FSEL2 - Function Select 2	R/W	0
5-3	FSEL1	FSEL1 - Function Select 1	R/W	0
2-0	FSEL0	FSEL0 - Function Select 0	R/W	0

From: BCM2835 ARM Peripherals

Calculating the Pin Offset

- If pin# is single-digit
  - Offset = address for GPFSEL**0** (base GPIO address, 0x3F200000)
- If pin# > 9 (more than one digit)
  - Offset = base GPIO address + 4*(pin# div 10)
  - Address of GPFSEL1 to GPFSEL5
- Example (pins 10 to 19):
  - Offset = 0x3F200000 + 4*(1) = 0x3F200004
  - Which is address for GPFSEL**1**

Getting to Appropriate Pins

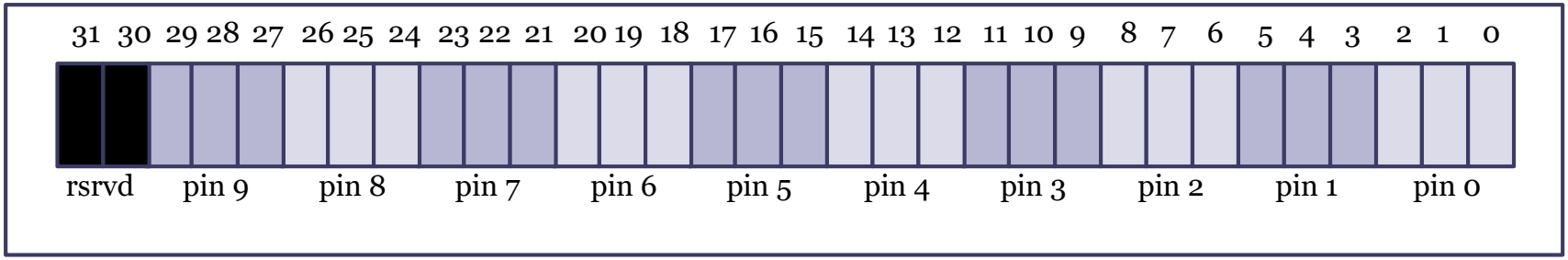
- Given some pin#
- Let GPFSEL{*n*} be the register that controls this pin#
  - *n* = pin# *div* 10
- Let *d* be the least significant digit of pin#
- The 3 bits of GPFSEL{*n*} that select pin#'s function start at ***d**3**

GPIOFSEL{n}

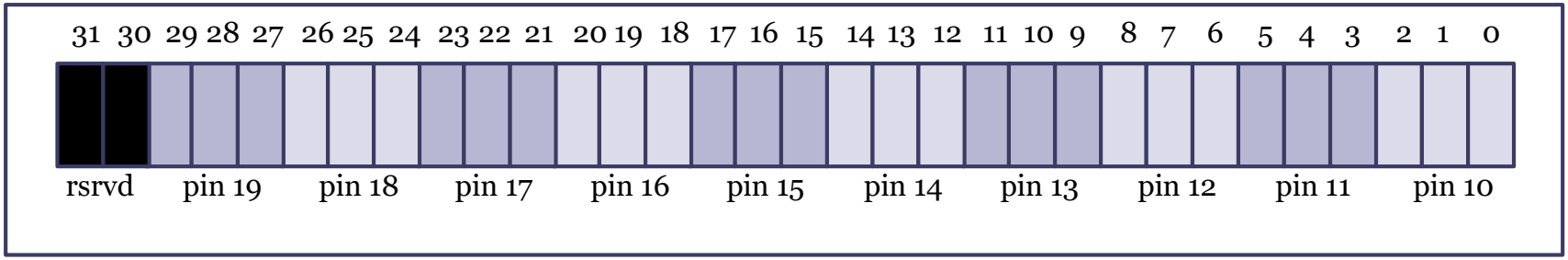
30-31	27-29	24-26	21-23	18-20	15-17	12-14	9-11	6-8	3-5	0-2
reserved	Pin {n}9	Pin {n}8	Pin {n}7	Pin {n}6	Pin {n}5	Pin {n}4	Pin {n}3	Pin {n}2	Pin {n}1	Pin {n}0

GPFSEL registers

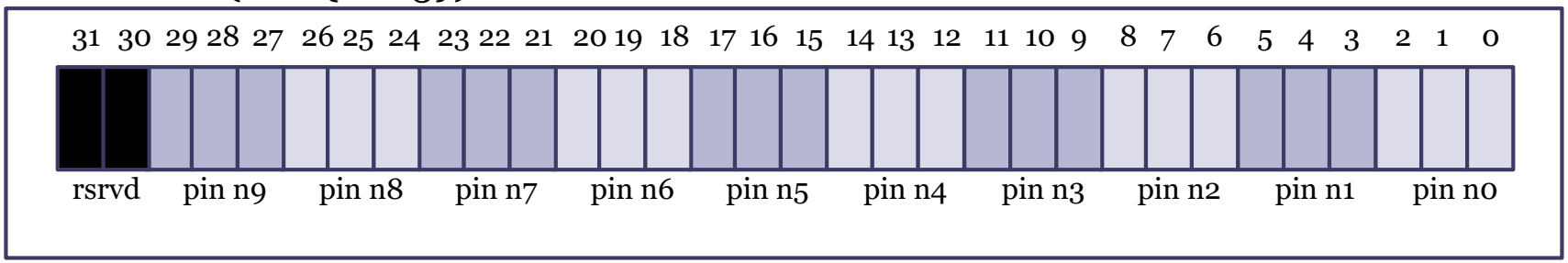
GPFSEL0



GPFSEL1



GPFSELn ($n \in \{0 .. 5\}$)



Example

- Pin 32
- Register GPFSEL3, offset 0x3F20000c
  - = 4*(32 div 10) + 0x3F200000
- Least significant bit of pin# is 2
- The three bits of GPFSEL3 that control pin 32 start at bit 3*2 = 6
- Hence, bits 6-8 define the function of pin 32

Setting the bits in GPFSEL{n}

- Clear the appropriate bits in GPFSEL{n}
- Set the appropriate bits in GPSFEL{n}

Resetting the bits for pin 3

Reset GPFSEL0[9..11] to 000

AND GPFSEL0 with 1..1 000 111 111 111

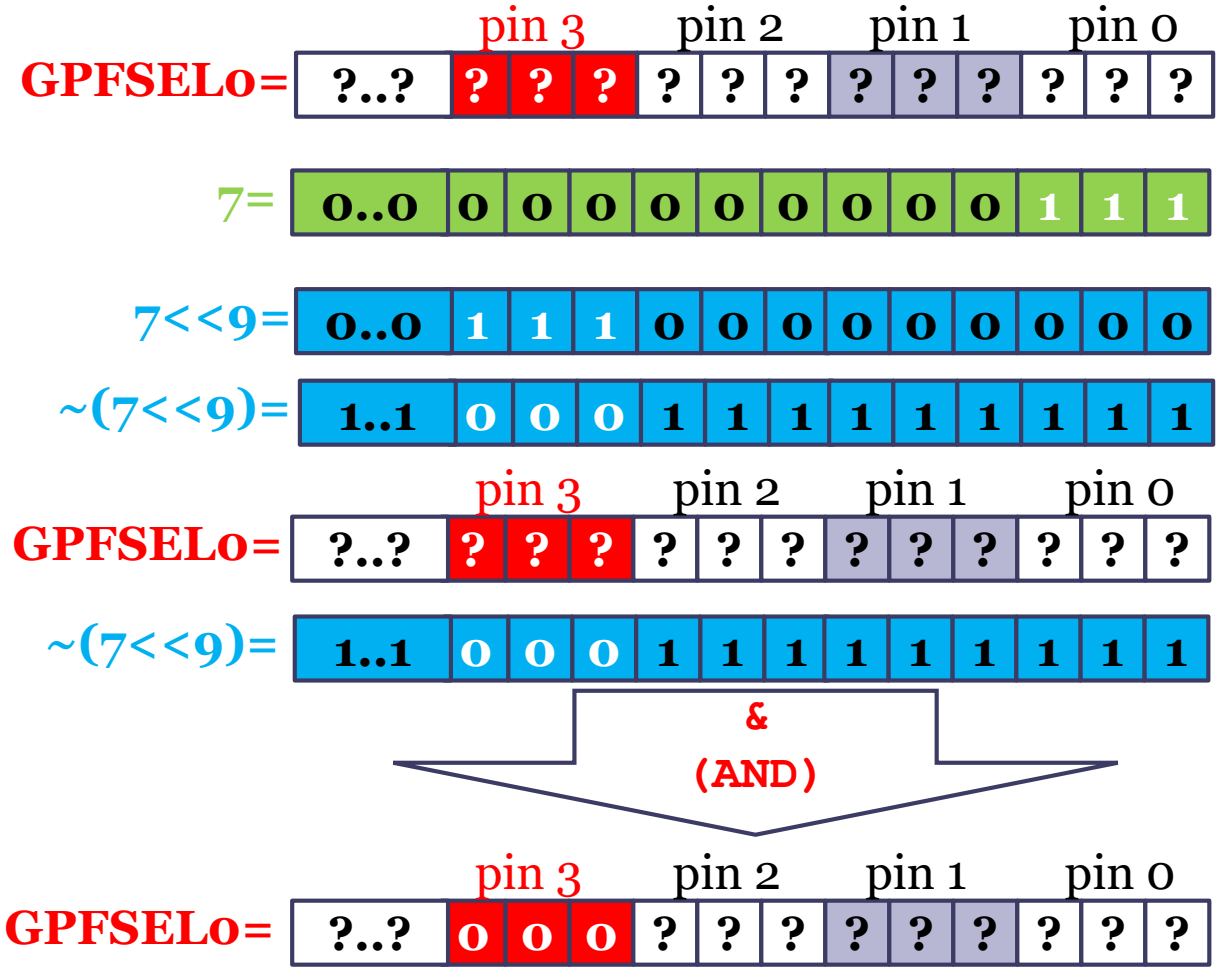
```
~(7<<(((p)%10)*3))
b0111 << (3%10)* 3
b0111 << 9
shift left 9 bits
0..0 111 000 000 000
~(0..0 111 000 000 000)
1..1 000 111 111 111
```

Resetting the bits for pin 3

```
#define GPIO_BASE   0x3F200000 // Address for GPIOFSEL0
unsigned *gpio = (unsigned*)GPIO_BASE;   // GPIO base

*(gpio+((p)/10)) &= ~(7<<(((p)%10)*3))
Bitwise AND [0x3F200000] with 1..1 000 111 111 111
```

Pin 3 is an input pin now

Clearing pin 3 bits (Input pin)



Resetting the bits for pin p

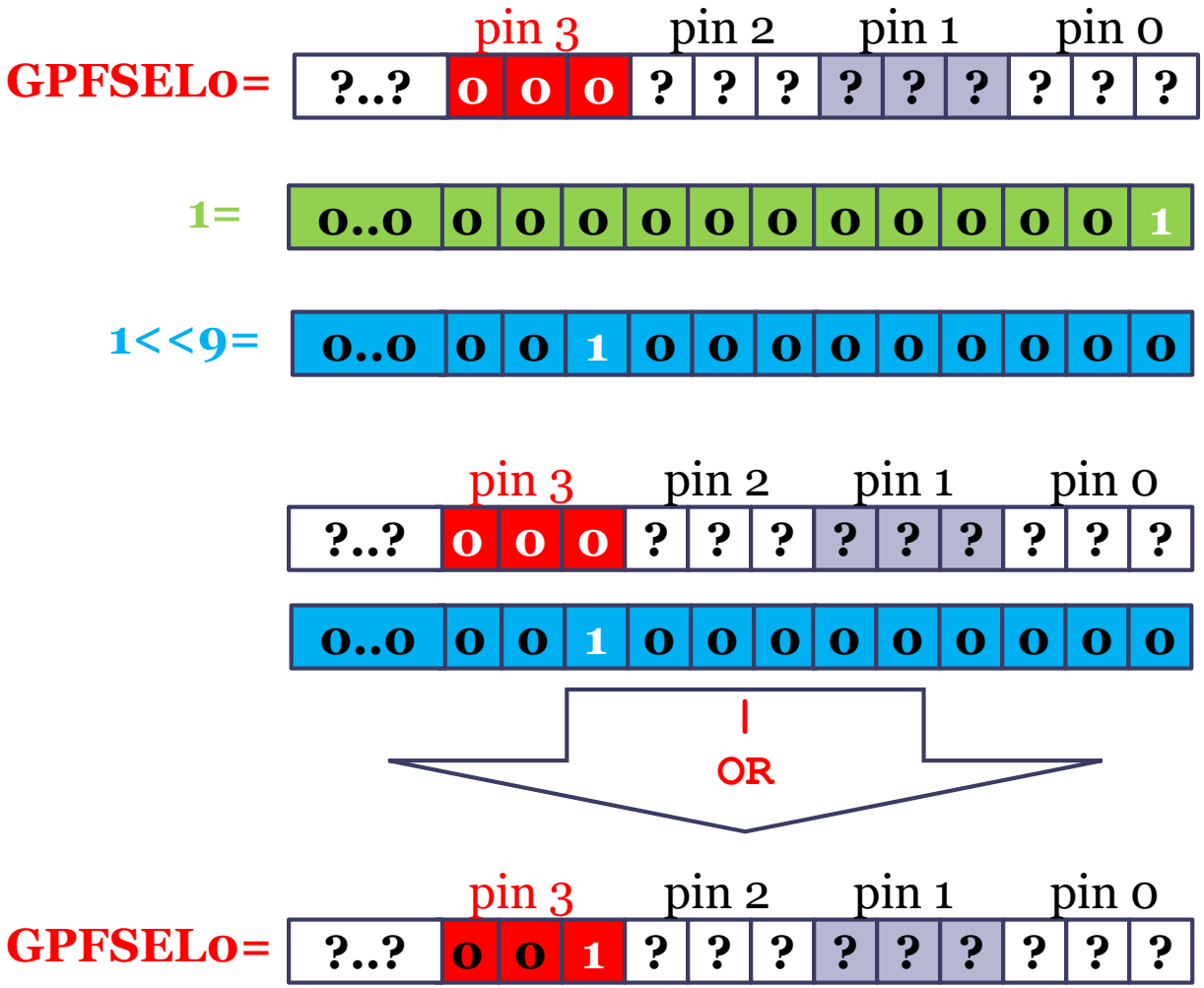
```
#define GPIO_BASE   0x3F200000 // Address for GPIOFSEL0
unsigned *gpio = (unsigned*)GPIO_BASE;   // GPIO base

#define INP_GPIO(p) *(gpio+((p)/10)) &= ~(7<<(((p)%10)*3))
```

Setting the bits for pin 3

```
Reset pin 3 (INP_GPIO(3))
*(gpio+((3)/10)) |= (1<<(((3)%10)*3))
(1<<(((3)%10)*3))
(1<<9)
 0..0 001 000 000 000
Bitwise OR [0x3F200000] with 0..0 001 000 000 000
```

Pin 3 is an output pin now

Example – Setting pin 3 bits



Setting the bits for pin p

```
#define GPIO_BASE   0x3F200000 // Address for GPIOFSEL0
unsigned *gpio = (unsigned*)GPIO_BASE;   // GPIO base

#define OUT_GPIO(p) *(gpio+((p)/10)) |= (1<<(((p)%10)*3))
```

Reading and Writing Pins

Performing I/O

- Once a pin function is set, then pin can be used for I/O
- To write to a pin, use GPSET{0,1} or GPCLR{0,1}
- To read from a pin, use GPLEV{0,1}

GPIO Pin Output Set Registers (GPSETn)

Synopsis The output set registers are used to set a GPIO pin. The SET{n} field defines the respective GPIO pin to set, writing a "0" to the field has no effect. If the GPIO pin is being used as in input (by default) then the value in the SET{n} field is ignored. However, if the pin is subsequently defined as an output then the bit will be set according to the last set/clear operation. Separating the set and clear functions removes the need for read-modify-write operations

Bit(s)	Field Name	Description	Type	Reset
31-0	SETn (n=0..31)	0 = No effect 1 = Set GPIO pin *n*	R/W	0

Table 6-8 – GPIO Output Set Register 0

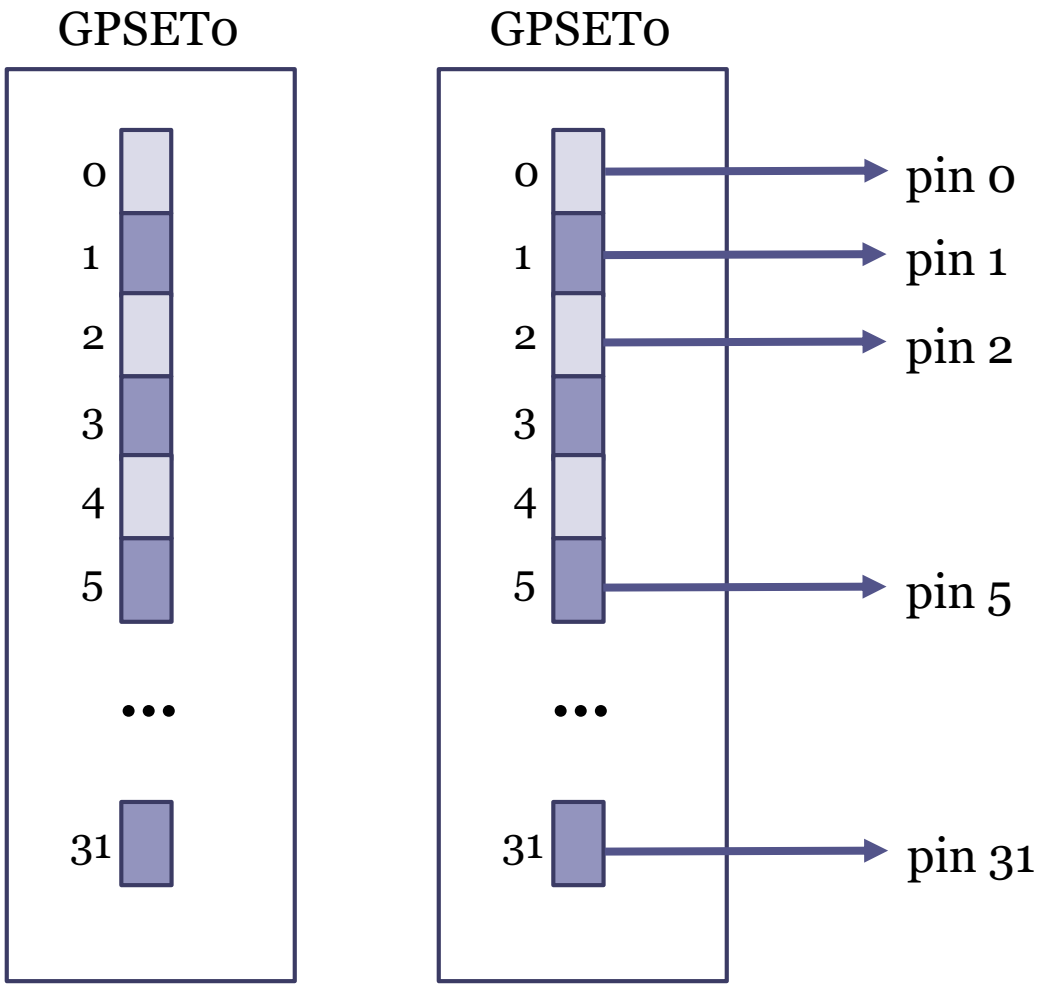
Dividing a pin# by 32 determines which register must be used

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	SETn (n=32..53)	0 = No effect 1 = Set GPIO pin *n*.	R/W	0

Table 6-9 – GPIO Output Set Register 1

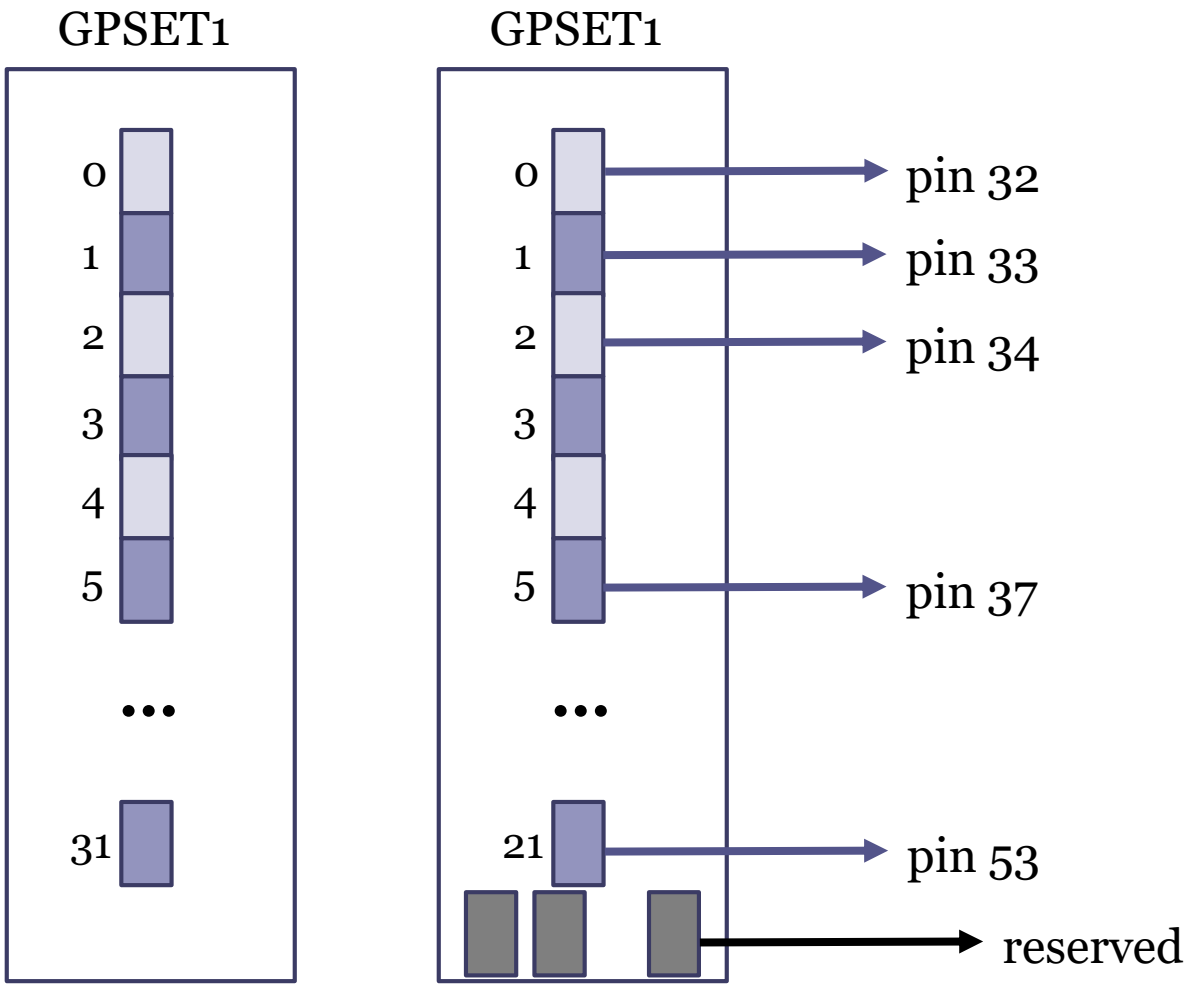
From: BCM2835 ARM Peripherals

GPSET0 register



GPSET1 register

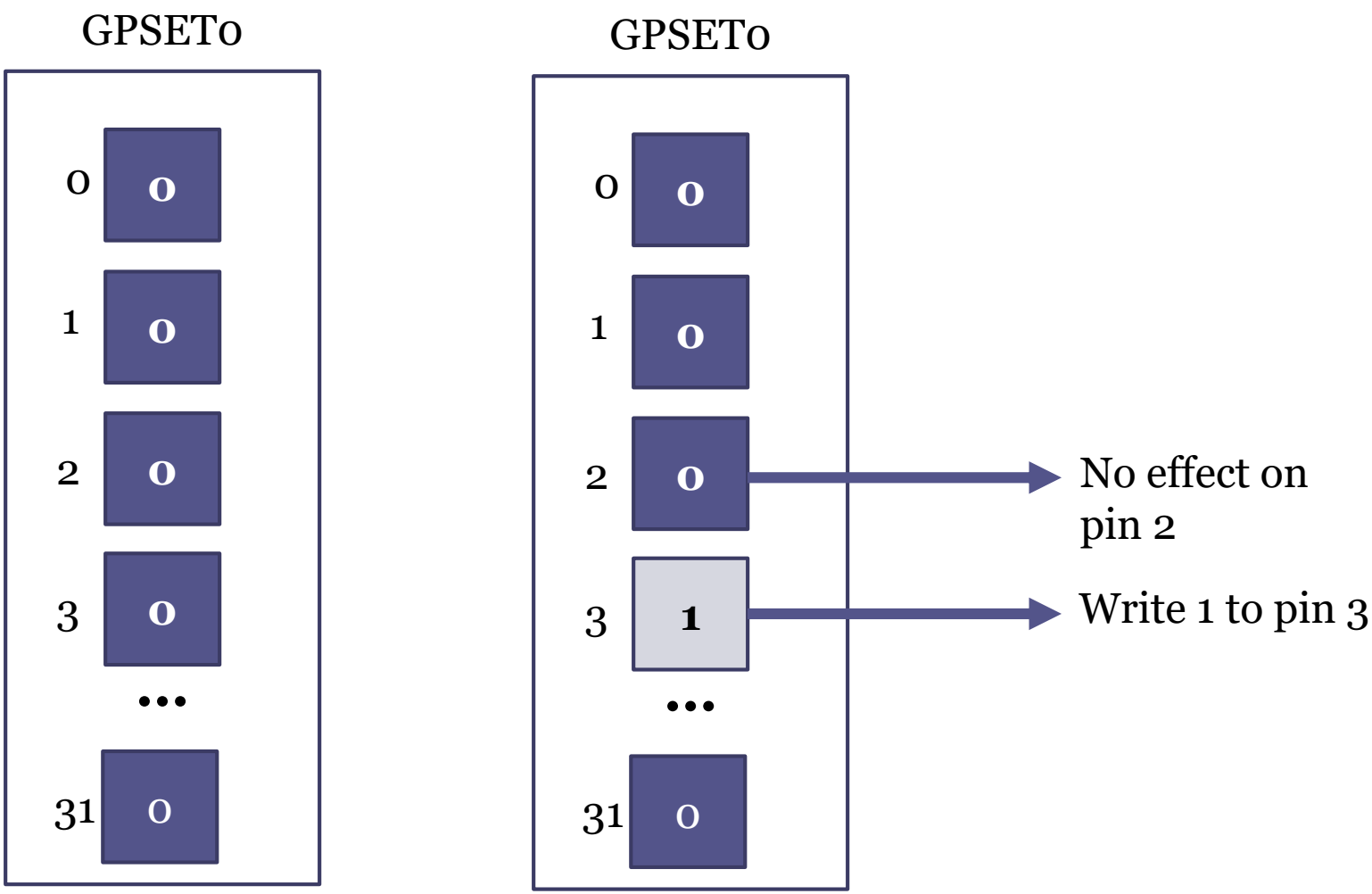
Not used on the Pi



GPSET0 Address

- Base GPIO Address : **`0x3F200000`**
- GPSET0 : Base + 28 = **`0x3F20001C`**
  - **`28 decimal is 0x1C`**

Setting (writing 1 to) a pin

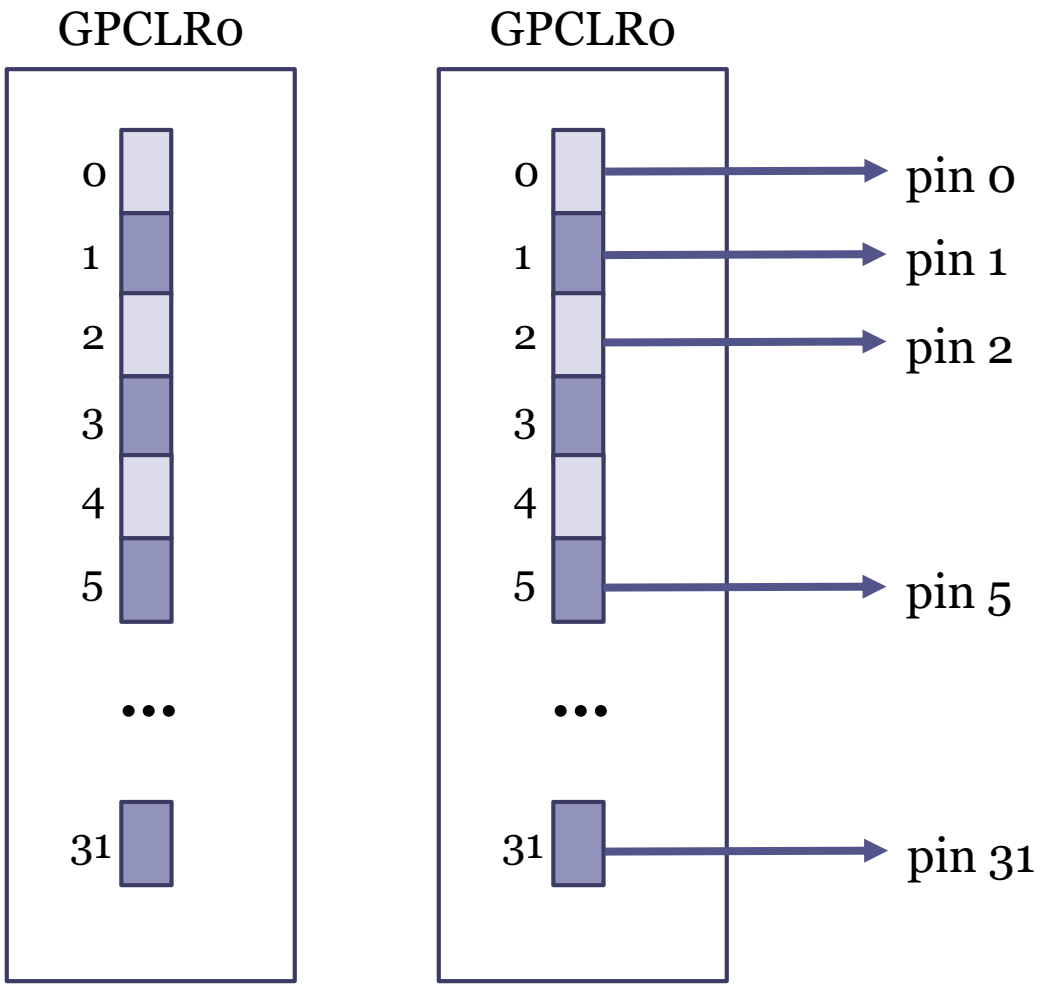


Setting (Writing to) a Pin

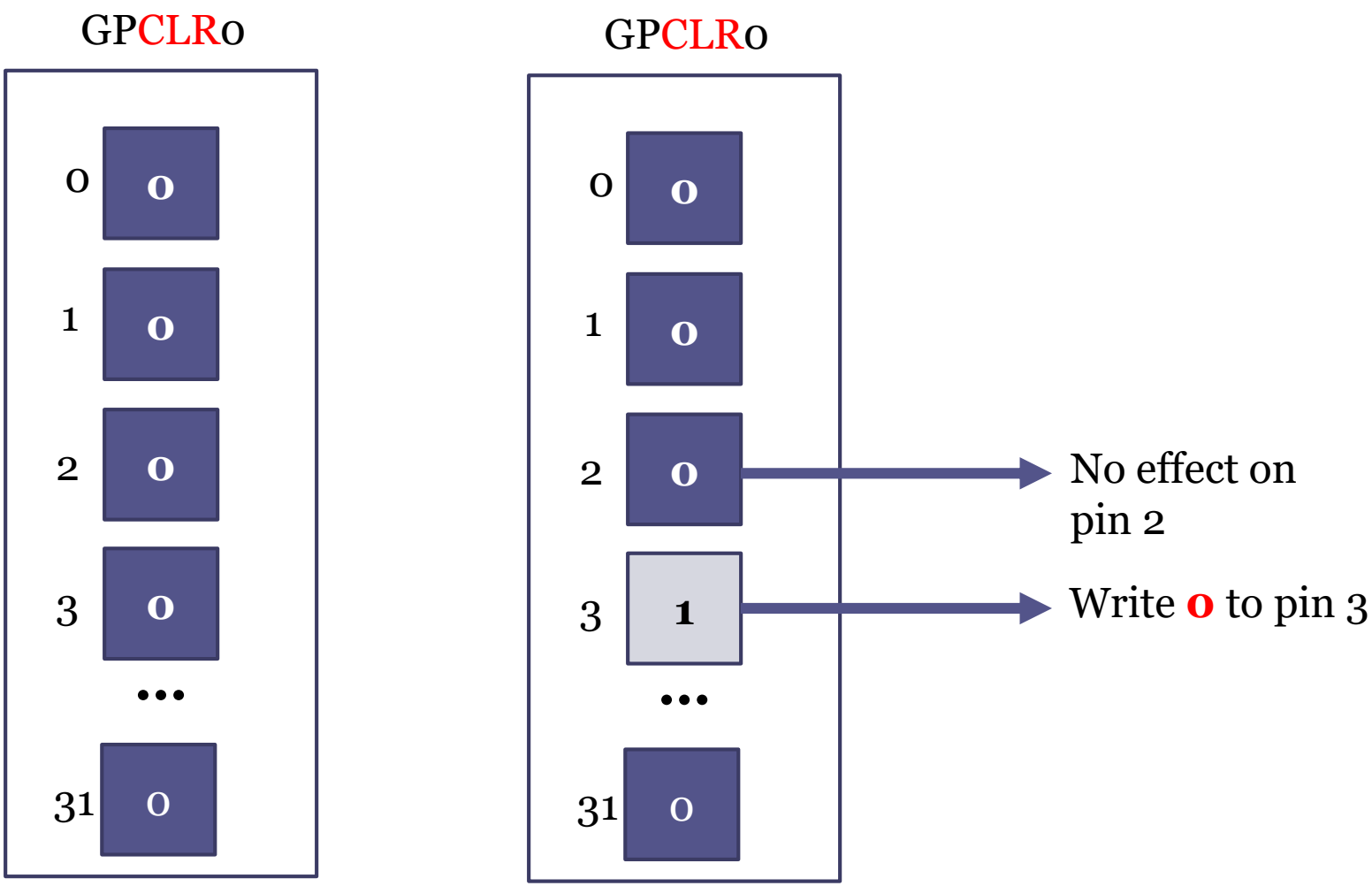
- If writing 1, then use GPSET{n}, n= 0,1
  - gpioAddr + 28 + {4*n}
    - 28 and 4*n are decimal values

- If writing 0, use GPCLR{n}
  - A write of 0 to GPSET{n} is ignored
  - gpioAddr + 40 + {4*n}
    - 40 and 4*n are decimal values

GPCLR0 register



Clearing (writing 0 to) a pin



Example

- To write 1 to (set) pin 3:

  GPSET[3] = 1

- To write 0 to (clear) pin 3:

  GPCLR[3] = 1

- GPSET[*i*] = 0 and GPCLR[*i*] =0 have no effect on pin

Writing to pins 0 – 31

```
unsigned int *gpio = 0x3F200000;

#define   GPSET0     7 \\ 28/4
#define   GPCLR0     10 \\ 40/4
...
if (writing 1)
     gpio[GPSET0] = 1 << pinNumber;
else
     gpio[GPCLR0] = 1 << pinNumber;
```

GPCLR1

- Used to clear the remaining pins
- Pins 32 to 53
- Bit 0 clears pin 32, …

- Not used on the Pi

GPIO Pin Output Clear Registers (GPCLRn)

Synopsis The output clear registers) are used to clear a GPIO pin. The CLR{n} field defines the respective GPIO pin to clear, writing a "0" to the field has no effect. If the GPIO pin is being used as in input (by default) then the value in the CLR{n} field is ignored. However, if the pin is subsequently defined as an output then the bit will be set according to the last set/clear operation. Separating the set and clear functions removes the need for read-modify-write operations.

Bit(s)	Field Name	Description	Type	Reset
31-0	CLRn (n=0..31)	0 = No effect 1 = Clear GPIO pin *n*	R/W	0

Table 6-10 – GPIO Output Clear Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	CLRn (n=32..53)	0 = No effect 1 = ~~Set~~ GPIO pin *n* **Clear**	R/W	0

Table 6-11 – GPIO Output Clear Register 1

From: BCM2835 ARM Peripherals

GPCLR0 Address

- Base GPIO Address : **`0x3F200000`**
- GPCLR0 : Base + 40 = **`0x3F200028`**

Reading a Pin

- Register GPLEVn is used to read a bit from a pin
- In the RPi, only GPLEV0 is used
  - Bit *n* corresponds to the value of pin *n*
- Address of GPLEV0 is 0x3F200000 + 52
  - 0x3F200034

Inside GPLEV{n}

GPIO Pin Level Registers (GPLEVn)
SYNOPSIS The pin level registers return the actual value of the pin. The LEV{n} field gives the value of the respective GPIO pin.

Bit(s)	Field Name	Description	Type	Reset
31-0	LEVn (n=0..31)	0 = GPIO pin *n* is low 0 = GPIO pin *n* is high	R/W	0

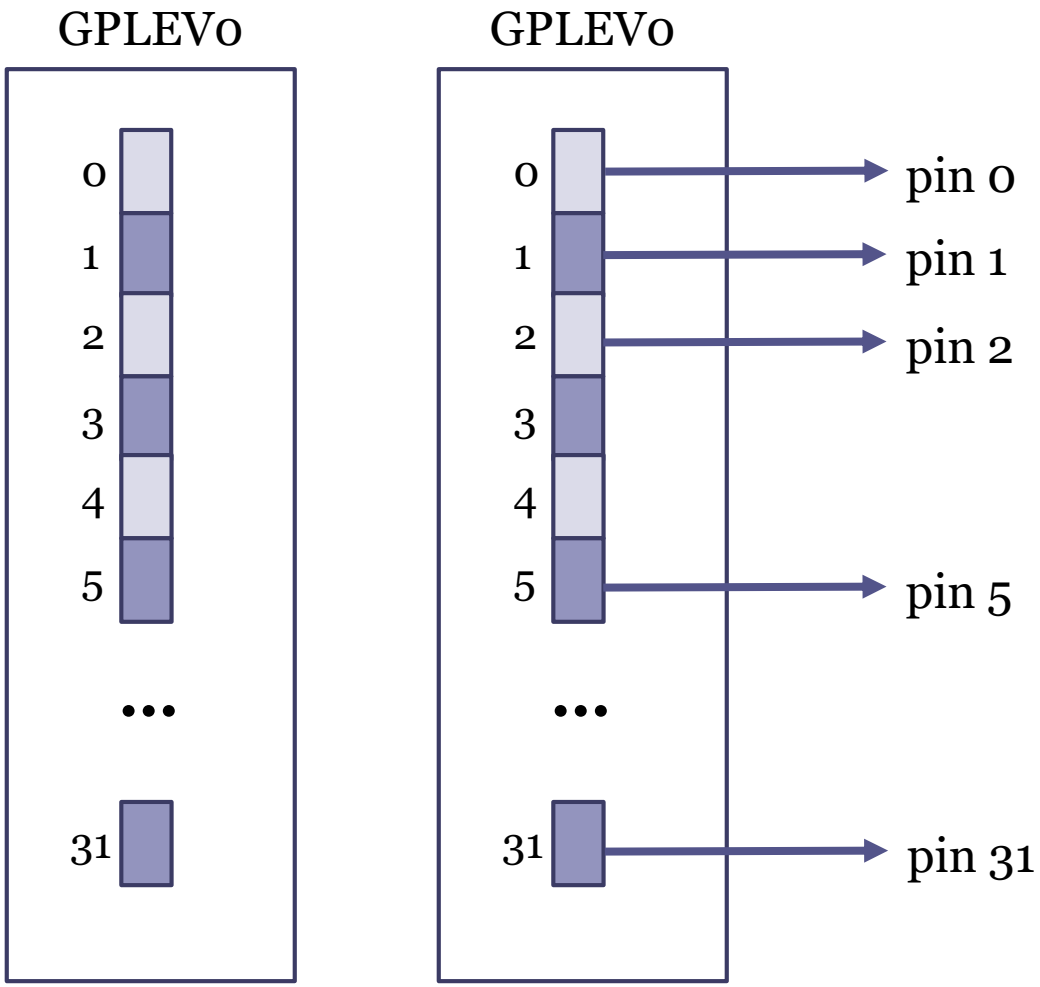
Table 6-12 – GPIO Level Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	LEVn (n=32..53)	0 = GPIO pin *n* is low 0 = GPIO pin *n* is high	R/W	0

Table 6-13 – GPIO Level Register 1

From: BCM2835 ARM Peripherals

GPLEV0 register



GPLEVn Addresses

- Base GPIO Address : **`0x3F200000`**
- GPLEV0 : Base + 52 = **`0x3F200034`**
- GPLEV1 : Base + 60 = **`0x3F20003C`**
  - Not used in the Pi

Reading from pins 0 – 31

```
unsigned int *gpio = 0x3F200000;

#define  GPLEV0    13
...
int v;

v = (gpio[GPLEV0] >> 3) & 1;
// v = pin 3 value
```

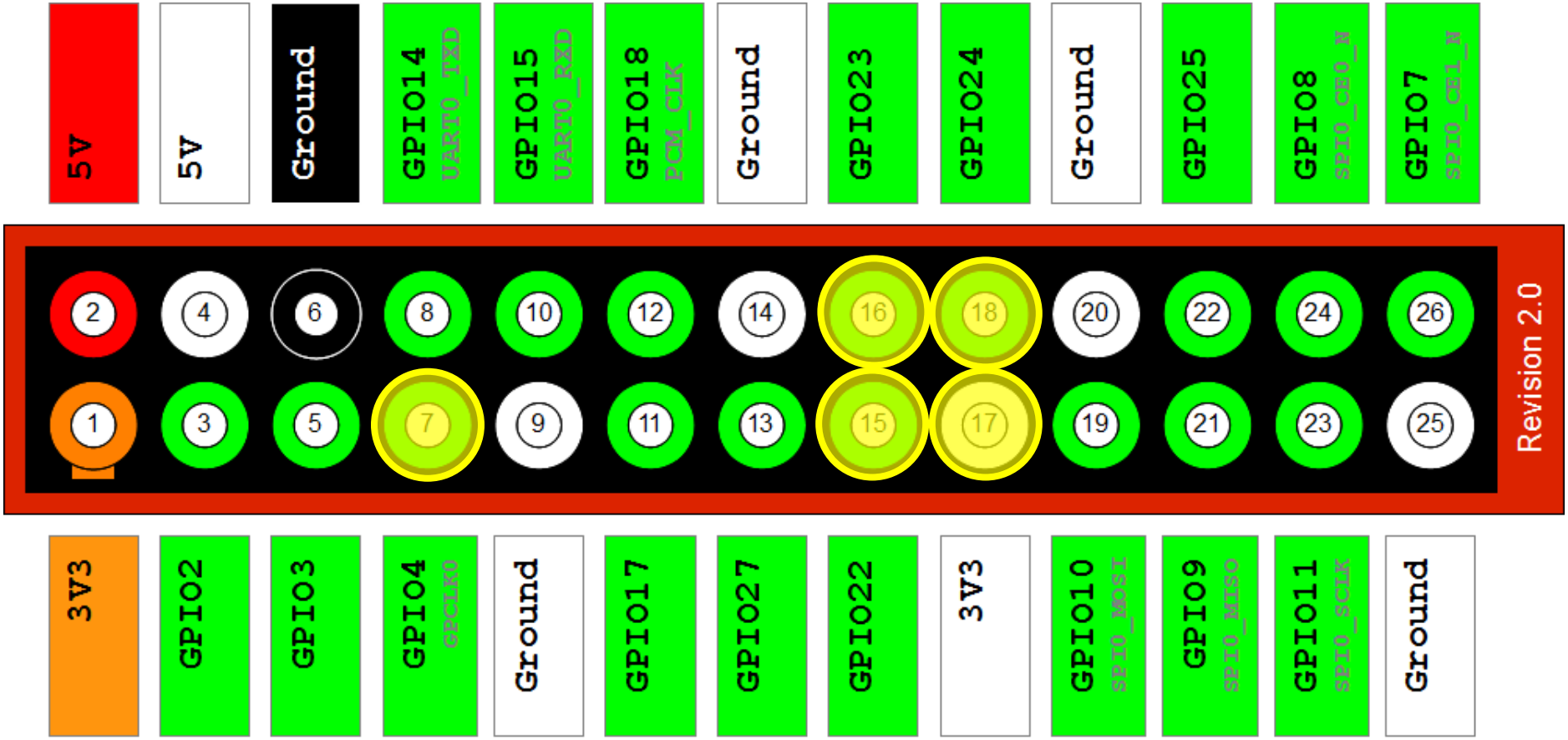
Other GPIO functions

- Clock signal
- UART
- JTAG
- Etc …

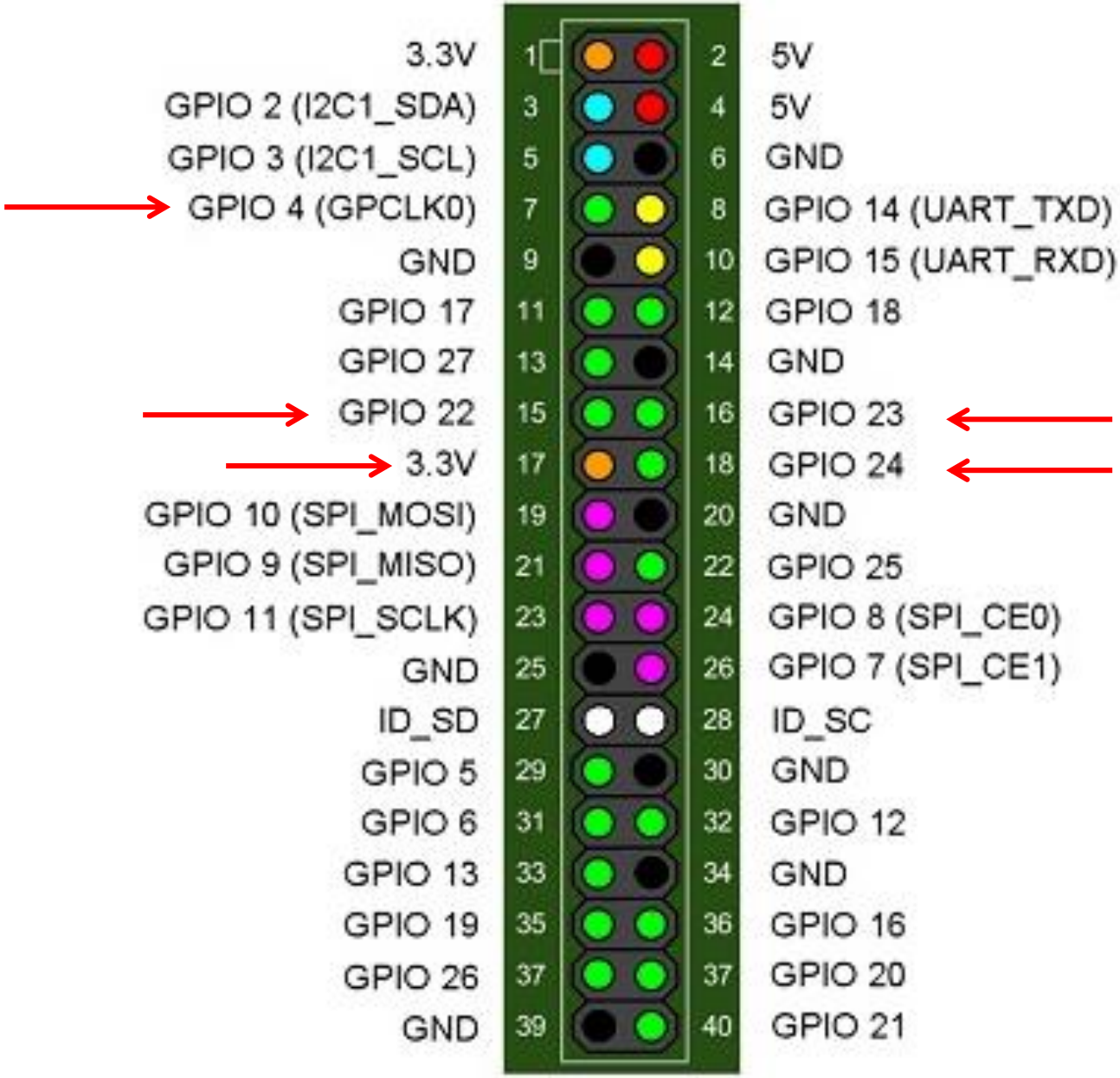
JTAG Interface

- Joint Test Action Group
  - IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture
- Originally used for IC debug ports
- Embedded system development relies on JTAG to perform debugging
- ARM11 has an extensive JTAG capability
- `EnableJTAG` routine in `jtag.s` (by Dmitry Zavyalov)

JTAG Interface Pins



www.raspberrypi-spy.co.uk


3.3V 1
2 5V
GPIO 2 (I2C1_SDA) 3
4 5V
GPIO 3 (I2C1_SCL) 5
6 GND
GPIO 4 (GPCLK0) 7
8 GPIO 14 (UART_TXD)
GND 9
10 GPIO 15 (UART_RXD)
GPIO 17 11
12 GPIO 18
GPIO 27 13
14 GND
GPIO 22 15
16 GPIO 23
3.3V 17
18 GPIO 24
GPIO 10 (SPI_MOSI) 19
20 GND
GPIO 9 (SPI_MISO) 21
22 GPIO 25
GPIO 11 (SPI_SCLK) 23
24 GPIO 8 (SPI_CE0)
GND 25
26 GPIO 7 (SPI_CE1)
ID_SD 27
28 ID_SC
GPIO 5 29
30 GND
GPIO 6 31
32 GPIO 12
GPIO 13 33
34 GND
GPIO 19 35
36 GPIO 16
GPIO 26 37
37 GPIO 20
GND 39
40 GPIO 21

ebay.com