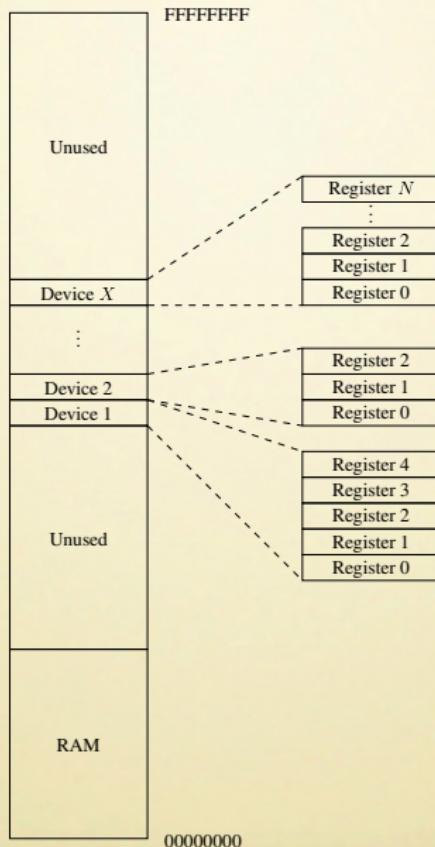


Modern Assembly Language Programming  
with the  
ARM processor  
Chapter 11: Devices



- 1 Introduction
- 2 Accessing Devices
- 3 General Purpose Input/Output Device (GPIO)
- 4 Real GPIO Devices

# Memory-Mapped Devices



# Raw Access to Device Registers

## NAME

`mmap, munmap` - map or unmap files or devices into memory

## SYNOPSIS

```
#include <sys/mman.h>

void *mmap(void *addr, size_t length, int prot, int flags,
           int fd, off_t offset);
int munmap(void *addr, size_t length);
```

See NOTES for information on feature test macro requirements.

## DESCRIPTION

`mmap()` creates a new mapping in the virtual address space of the calling process. The starting address for the new mapping is specified in `addr`. The `length` argument specifies the length of the mapping.

If `addr` is `NULL`, then the kernel chooses the address at which to create the mapping; this is the most portable method of creating a new mapping. If `addr` is not `NULL`, then the kernel takes it as a hint about where to place the mapping; on Linux, the mapping will be created at a nearby page boundary. The address of the new mapping is returned as the result of the call.

## Mapping a Device in C

The program must have permissions to read and write /dev/mem.

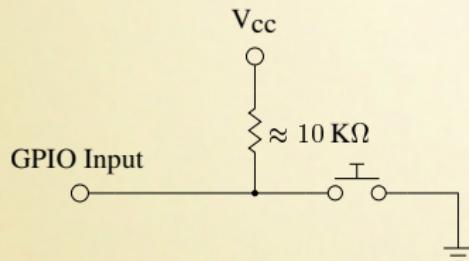
```
1 #include <sys/mman.h>
2 #include <fcntl.h>
3 :
4 void *devaddress = HARDWARE_ADDRESS;
5 int memfd;
6 :
7 memfd = open("/dev/mem", O_RDWR);
8 if(memfd<0) exit(1);
9 devaddress = mmap(devaddress, 4096, PROT_READ|PROT_WRITE,
10                   MAP_SHARED, memfd, 0);
11 /* devaddress now points to the mapped device registers */
12 :
13 /* Finished accessing the device registers . Clean up. */
14 munmap(devaddress, 4096);
15 close(memfd);
16 :
```

## General Purpose Input/Output Device (GPIO)

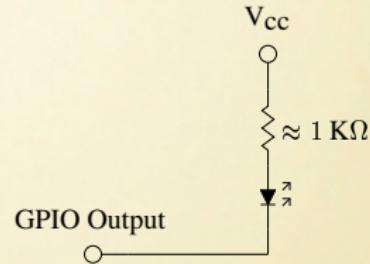
For most GPIO devices:

- individual pins or groups of pins can be configured,
- pins can be configured to be input or output,
- pins can be disabled so that they are neither input nor output,
- input values can be read by the CPU (typically high=1, low=0),
- output values can be read or written by the CPU, and
- input pins can be configured to generate interrupt requests.

## Input and Output



GPIO pin being used as input to read the state of a push-button switch.



GPIO pin being used as output to drive an LED.

## Raspberry Pi GPIO

The Raspberry Pi is based on the Broadcom BCM2835.

- The GPIO device controls 54 pins.
- Pins are named GPIO $x$ , where  $x$  is a number between 0 and 53.
- Each pin can be used
  - for general purpose input,
  - for general purpose output, or
  - for one of six pre-defined alternate functions.

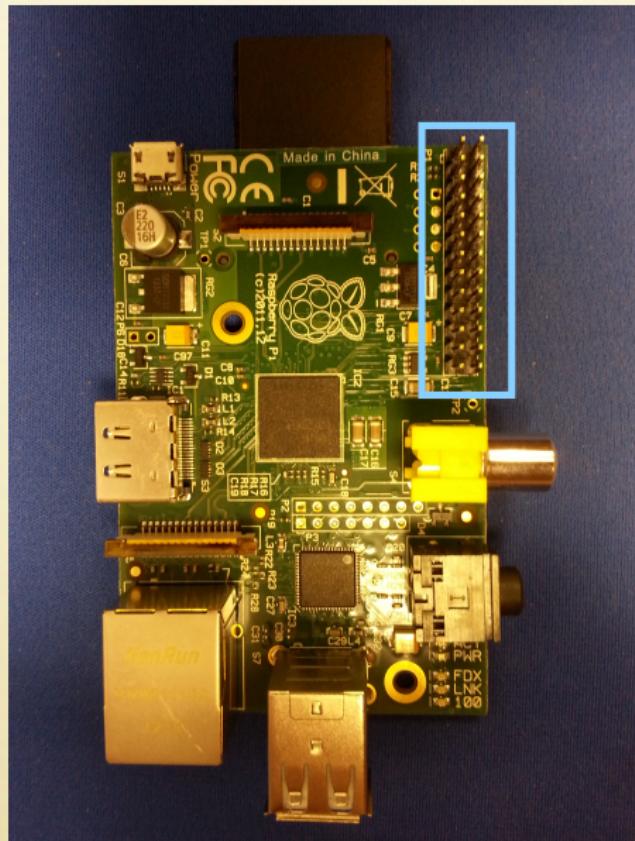
## Raspberry Pi GPIO

For example, GPIO4 can be used

- for general purpose I/O,
- to send the signal generated by General Purpose Clock 0 to external devices,
- to send bit one of the Secondary Address Bus to external devices, or
- to receive JTAG data for programming the firmware of the device.

The last 8 GPIO pins, GPIO46–GPIO53 have no alternate functions, and are used only for GPIO.

# Raspberry Pi GPIO



# Raspberry Pi GPIO

3V Power	(1) (2)	5V Power
GPIO 2	(3) (4)	5V
GPIO 3	(5) (6)	Ground
GPIO 4 (GPCLK0)	(7) (8)	GPIO 14 (TXD0) Universal Asynchronous Receiver/Transmitter
Ground	(9) (10)	GPIO 15 (RXD0) (UART0)
GPIO 17	(11) (12)	GPIO 18 (PWM0)
GPIO 27	(13) (14)	Ground
GPIO 22	(15) (16)	GPIO 23
3V Power	(17) (18)	GPIO 24
Serial Peripheral Interface (SPI0)	GPIO 10 (19) (20)	Ground
	GPIO 9 (21) (22)	GPIO 25
	GPIO 11 (23) (24)	GPIO 8
Ground	(25) (26)	GPIO 7

## pcDuino GPIO

The pcDuino is based on the AllWinner A10/A20.

- The GPIO device controls 175 pins.
- Pins are arranged in seven ports.
- Each of the seven ports is identified by a letter between “A” and “I”.
- Some ports have as many as 28 physical pins, while others have as few as six.
- Regardless of the number of pins, the port registers are all laid out the same way on every port.
- Each pin can be used
  - for general purpose input,
  - for general purpose output, or
  - for one of six pre-defined alternate functions.

## pcDuino GPIO

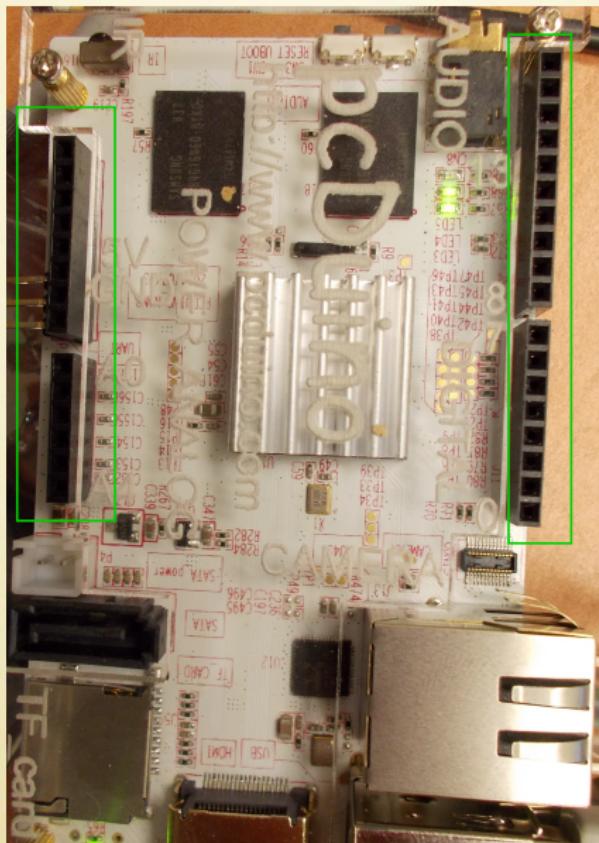
For example, Port B, Pin 2 can be used

- for general purpose I/O, or
- to send the output of Pulse Width Modulator 0 to an external device.

Port I, Pin 19 can be used

- for general purpose I/O,
- to receive RS232 serial data, or
- to route an interrupt signal from an external device to the interrupt controller.

# pcDuino GPIO



# pcDuino GPIO

