Cyber-Physical Systems

Input and Output

ICEN 553/453– Fall 2018

Prof. Dola Saha

Connecting the Analog and Digital Worlds

- Cyber
  - Digital
  - Discrete in Time
  - Sequential
- Physical
  - Continuum
  - Continuous in time
  - Concurrent

Practical Issues

- Analog vs. digital
- Wired vs. wireless
- Serial vs. parallel
- Sampled or event triggered
- Bit rates
- Access control, security, authentication
- Physical connectors
- Electrical requirements (voltages and currents)

GPIO Block Diagram

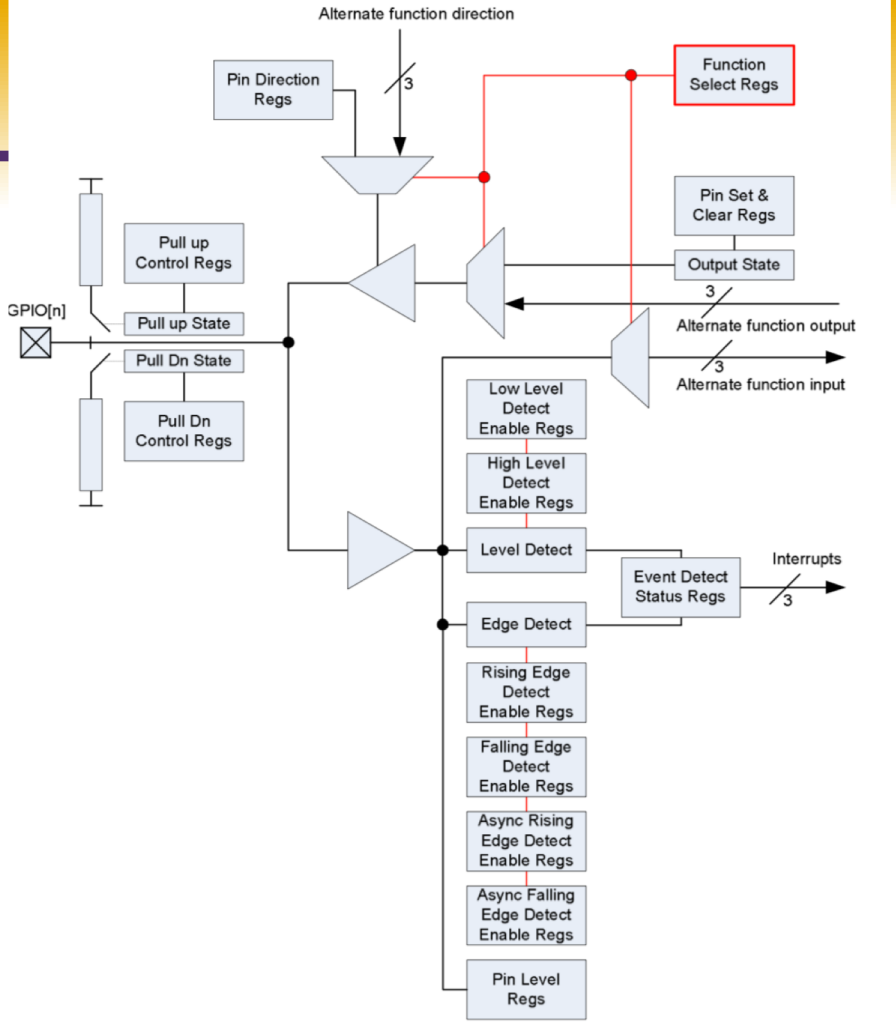
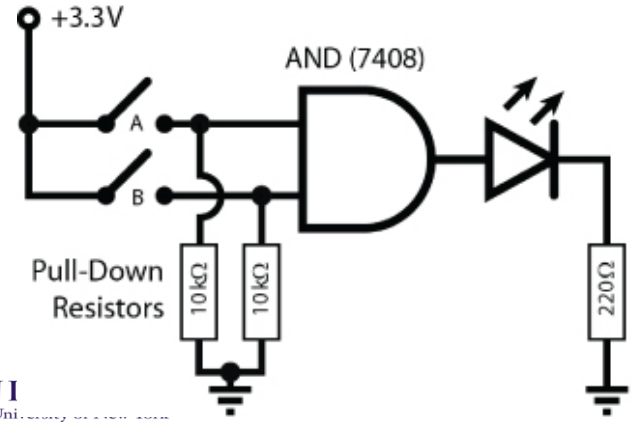
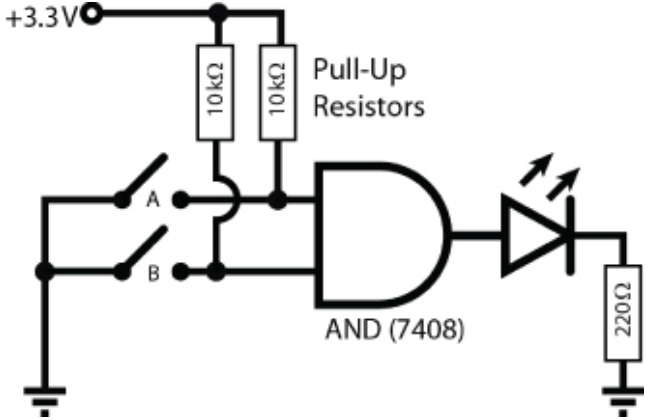


Figure 6-1 GPIO Block Diagram

Pull-down and Pull-up Resistors

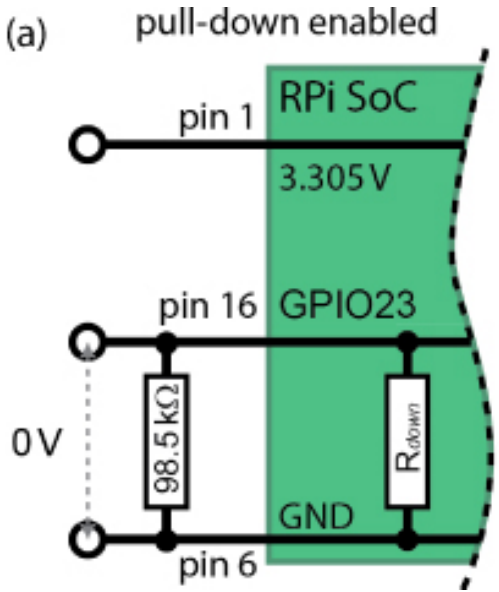
- Used to ensure that the switches do not create floating inputs
- Pull-down resistors:
  - used to guarantee that the inputs to the gate are low when the switches are open
- Pull-up resistors:
  - used to guarantee that the inputs are high when the switches are open.

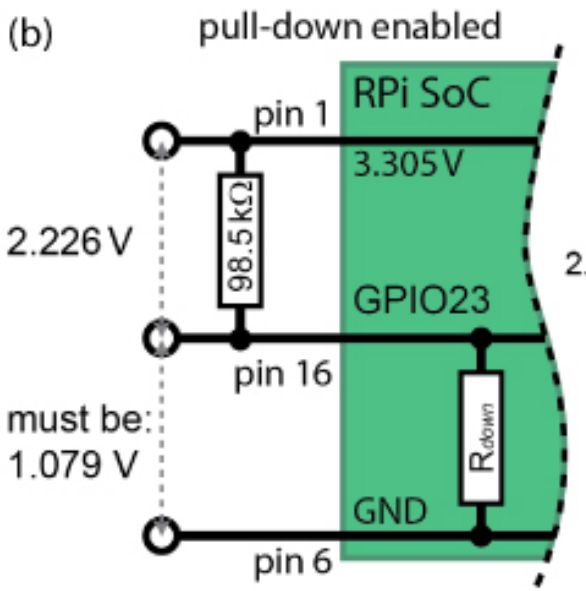




Calculate Internal Resistor Value

- Voltage Divider





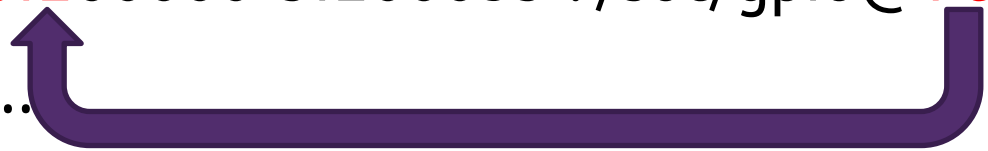
voltage divider:

$$2.226\,V = 3.305\,V \times \frac{98.5\,k\Omega}{98.5\,k\Omega + R_{down}}$$

$$R_{down} = 47.75\,k\Omega$$

Internal pull-up/pull-down Resistors

- Can be configured using memory based GPIO control
- **cat /proc/iomem**
- 00000000-3b3fffff : System RAM
- ....
- 3f200000-3f2000b3 : /soc/gpio@7e200000
- ...



Address Mapped

Use /dev/mem directly

- **wget** http://www.lartmaker.nl/lartware/port/devmem2.c
- **gcc devmem2.c -o devmem2**
- **./devmem2**

Usage: ./devmem2 { address } [type [data]]

address : memory address to act upon

type : access operation type : [b]yte, [h]alfword, [w]ord

data : data to be written

GPIO Pull-up/down Register Control

- The **GPIO Pull-up/down Register** controls the actuation of the internal pull-up/down control line to ALL the GPIO pins. This register must be used in conjunction with the **2 GPPUDCLKn registers**.
- Note that it is **not possible to read back** the current Pull-up/down settings and so it is the users' responsibility to 'remember' which pull-up/downs are active. The reason for this is that GPIO pull-ups are maintained even in power-down mode when the core is off, when all register contents is lost.

BCM 2837 Manual

- Table 6-1

0x 7E20 0094	GPPUD	GPIO Pin Pull-up/down Enable	32	R/W
0x 7E20 0098	GPPUDCLK0	GPIO Pin Pull-up/down Enable Clock 0	32	R/W
0x 7E20 009C	GPPUDCLK1	GPIO Pin Pull-up/down Enable Clock 1	32	R/W

- Table 6-28

31-2	---	Unused	R	0
1-0	PUD	PUD - GPIO Pin Pull-up/down 00 = Off – disable pull-up/down 01 = Enable Pull Down control 10 = Enable Pull Up control 11 = Reserved *Use in conjunction with GPPUDCLK0/1/2	R/W	0

BCM 2837 Manual

Bit(s)	Field Name	Description	Type	Reset
(31-0)	PUDCLKn (n=0..31)	0 = No Effect 1 = Assert Clock on line *(n)* *Must be used in conjunction with GPPUD	R/W	0

Table 6-29 – GPIO Pull-up/down Clock Register 0

Bit(s)	Field Name	Description	Type	Reset
31-22	-	Reserved	R	0
21-0	PUDCLKn (n=32..53)	0 = No Effect 1 = Assert Clock on line *(n)* *Must be used in conjunction with GPPUD	R/W	0

Table 6-30 – GPIO Pull-up/down Clock Register 1

Control Pull-up/down

The GPIO Pull-up/down Clock Registers control the actuation of internal pull-downs on the respective GPIO pins. These registers must be used in conjunction with the GPPUD register to effect GPIO Pull-up/down changes. The following sequence of events is required:

1. Write to GPPUD to set the required control signal (i.e. Pull-up or Pull-Down or neither to remove the current Pull-up/down)
2. Wait 150 cycles – this provides the required set-up time for the control signal
3. Write to GPPUDCLK0/1 to clock the control signal into the GPIO pads you wish to modify – NOTE only the pads which receive a clock will be modified, all others will retain their previous state.
4. Wait 150 cycles – this provides the required hold time for the control signal
5. Write to GPPUD to remove the control signal
6. Write to GPPUDCLK0/1 to remove the clock

Pull Down Resistor is enabled

- Set bit 4 on the GPPUDCLK0 register, clear the GPPUD register, and then remove the clock control signal from GPPUDCLK0
  - GPIO4 is bit 4, which is 10000 in binary ($0x10_{16}$)
- Get the Value in GPIO 4
  - sudo su
  - cd /sys/class/gpio/
  - echo 4 > export
  - cd gpio4
  - cat value

Pull Down Resistor is enabled

- GPPUD Enable Pull-down
  - sudo /home/dsaha/myCode/devmem2 0x3F200094 w 0x01
- GPPUDCLK0 enable GPIO 4
  - sudo /home/dsaha/myCode/devmem2 0x3F200098 w 0x10
- GPPUD Disable Pull-down
  - sudo /home/dsaha/myCode/devmem2 0x3F200094 w 0x00
- GPPUDCLK0 disable Clk signal
  - sudo /home/dsaha/myCode/devmem2 0x3F200098 w 0x00
- cat value
  - 0

Pull up Configuration

- GPPUD Enable Pull-up
  - sudo /home/dsaha/myCode/devmem2 0x3F200094 w 0x02
- GPPUDCLK0 enable GPIO 4
  - sudo /home/dsaha/myCode/devmem2 0x3F200098 w 0x10
- GPPUD Disable Pull-up
  - sudo /home/dsaha/myCode/devmem2 0x3F200094 w 0x00
- GPPUDCLK0 disable Clk signal
  - sudo /home/dsaha/myCode/devmem2 0x3F200098 w 0x00
- cat value
  - 1

Before Next Class

- Install WiringPi
  - http://wiringpi.com