Internet of Things
Introduction to hardware

Pierre Rust

[1]MINES Saint-Étienne, CNRS
Lab Hubert Curien UMR 5516

[2]Orange Labs

Hardware in IoT

IoT is about connected objects, aka ***Things***

Things = Hardware + Software

- Software is well taught today, and relatively well known by most students and engineers
- Hardware is often "mysterious"

"There is a reason they call it hardware — it is hard,"
Tony Fadell, father of the iPod

Hardware in IoT

Things have some very important differences with generic personal computers:

- considered as devices / appliances, not PC
- have a specific purpose: generally meant to do one single task
- used in a huge number of different use-cases
- linked to their environment: sensors and actuators
- connected to various kind of networks
- often constrained devices

Hardware in IoT

Two big families:

- Microprocessor based devices: CPU
- Microcontroller based devices: MCU

CPU based devices

Microprocessor-based devices are basically specialized computers.

- microprocessor = general purpose applications
- use same kind of CPU architecture than your PC or your smartphone:
  - most of the time ARM: smartphones, gadgets, etcs.
  - Intel x86 (generally used for desktop, laptops, servers)
  - MIPS: gateways, network appliances
- run a full Operating System
  - generally a Linux derivative: android, openWTR, Brillo, Ubuntu IoT, etc.
- a PC with no keyboard, mouse nor screen !

CPU based devices

Advantages

- "lots" of processing power
- "lots"" of memory: several MB to several GB of RAM
- a full Operating System (generally Linux)
- can run several programs simultaneously
- easy Internet connectivity
- relatively easy to program: languages, tools, etc.

CPU based devices

Limitations

- expensive
- high power consumption
- not real time
- limited connectivity to hardware and sensors
  - much better than a traditional PC, but still not at the level of a MCU
  - some kind interfaces are more difficult to handle as the OS is not real time

CPU based devices

Some example of devices using a CPU:

Prototyping boards

There's a huge variety of prototyping board today, for makers, hobbyist, ... and students:

- The most successful: Raspberry Pi
- many other now :
  - Intel: Edison & Galileo
  - Beaglebone
  - UDooo
  - Cubieboard
  - etc.

Microcontrollers

"A microcontroller (or MCU, short for microcontroller unit) is a small computer (SoC) on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals." (Wikipedia)

MCU based devices

Advantages

- cheap !
- (very) low power consumption
- real time
- easily connectable to peripherals
  - Very good at controlling hardware: sensors, etc.
  - Digital and analog GPIO, SPI, I2C, UART, etc. PWM
- no OS: run a single program, very predictable (good thing!)

MCU based devices

Limitations

- limited processing power: a few MHz
- limited memory:
  - ram : 2Ko (arduino) to 128 KB (SMT32 for example)
  - flash : a few KB to 1 MB
- more difficult to program
  - No OS: single loop program, interrupts, etc.
  - specific tools
- limited connectivity :
  - Internet (through ethernet / wifi) is not appropriate
  - more suited to low power communication : Zigbee, Thread, 6LowPan, LoRa, SigFox, Bluetooth, etc.

MCU based devices

Some example of devices using a MCU:

Interfacing with peripherals

Peripheral

- any device not being part of the board / CPU /MCU we are considering, and attached to it
- sensors: pressure, temperature, light level, heat, push button, etc.
- actuators: relay, motor, servomotor, etc.
- other devices: display, RFId tag reader, etc.

Applies to both CPU and MCU:

- CPU for embedded application have GPIOs
- but MCU are still much better at this !

Interfacing with peripherals: Pins

Basic peripherals are connected to PINs on the prototyping board.

Type of PINs:

- GPIO: General Purpose Digital Input Output
- PWM : Pulse Width modulation
- ADC / DAC : Analog - Digital conversion

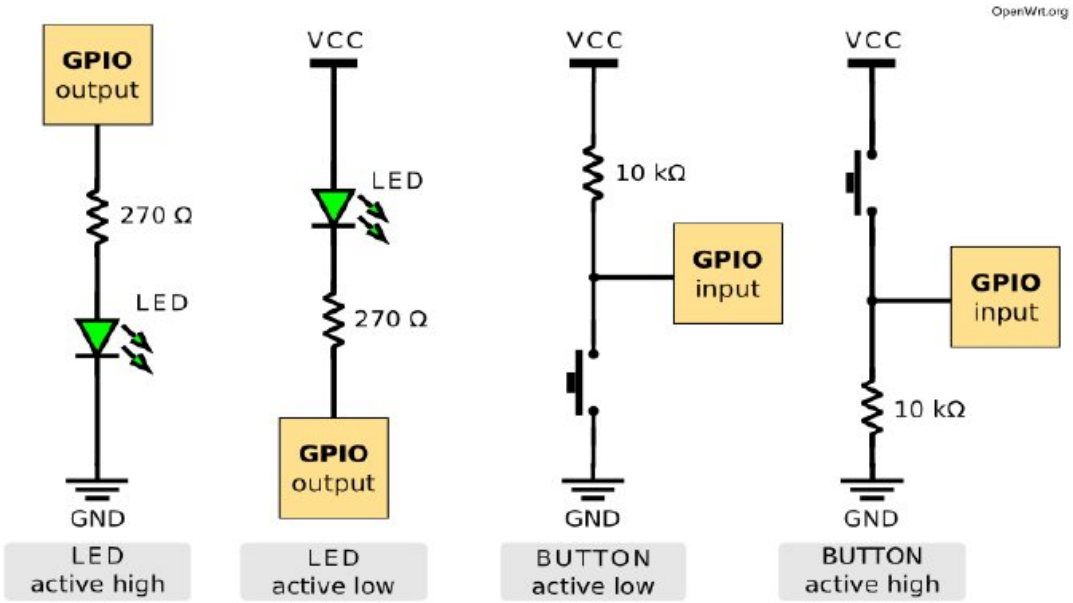
More advanced peripherals use specific interfaces:

- SPI
- I2C
- CAN
- 1-Wire

GPIO

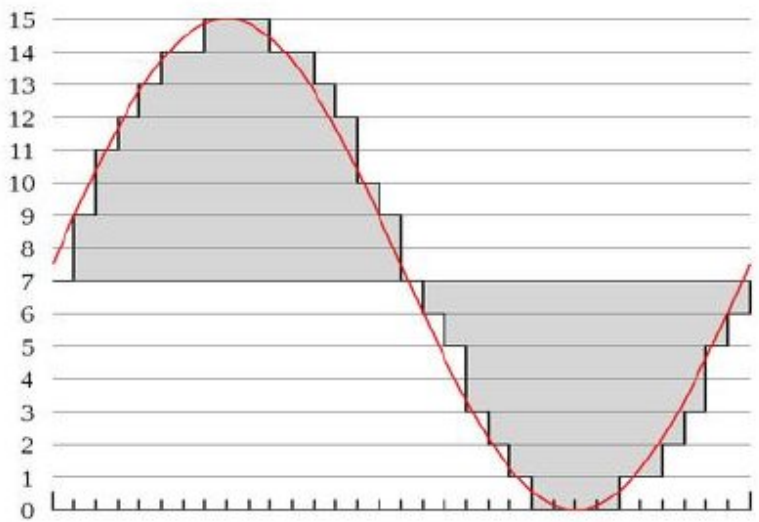
General Purpose Digital Input Output

- Input / Output
- Analog / Digital



Analog to Digital Conversion

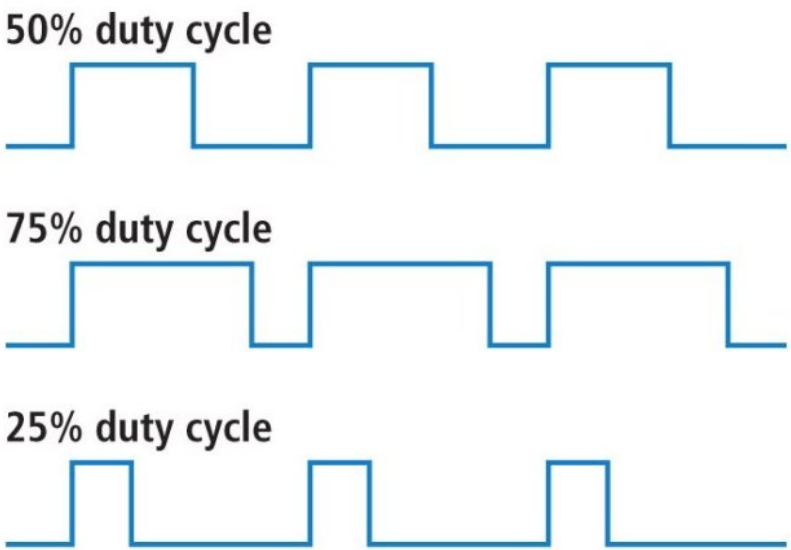
- ADC: Analog to Digital Conversion, converts an analog voltage to a digital value
- DAC: Digital to Analog Converter (less common): converts a digital value to an analog voltage



see `https://learn.sparkfun.com/tutorials/analog-to-digital-conversion`

Pulse Width Modulation

- The output oscillates between 1 and 0
- Creates analog voltage from a digital output by sending a series of pulses
- Used for servo-motors, leds (dimmer), etc.
- Can be emulated with software, but difficult and not stable enough for some use



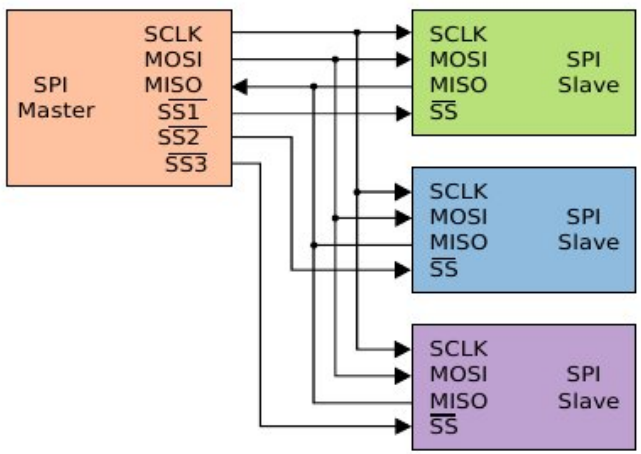
50%, 75%, and 25% Duty Cycle Examples

see `https://learn.sparkfun.com/tutorials/pulse-width-modulation`

Serial bus: SPI

SPI: Serial Peripheral Interface

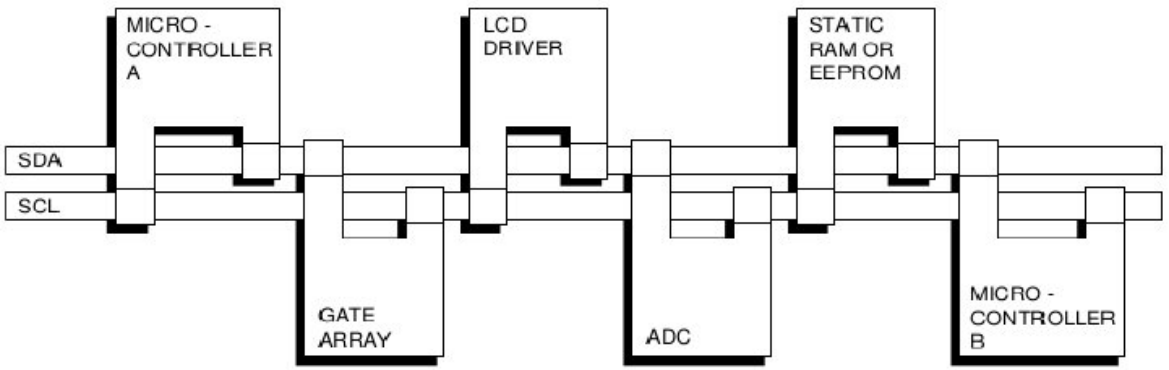
- four-wire synchronous serial bus
- master/slave
- short distance
- for sensors, LCDs, card readers, etc.
- the peripherals contains a microcontroller !



Serial bus: I2C

I2C: Inter-Integrated Circuit

- two-wire synchronous serial bus
- multi-master
- short distance
- same applications than for SPI



Questions ?