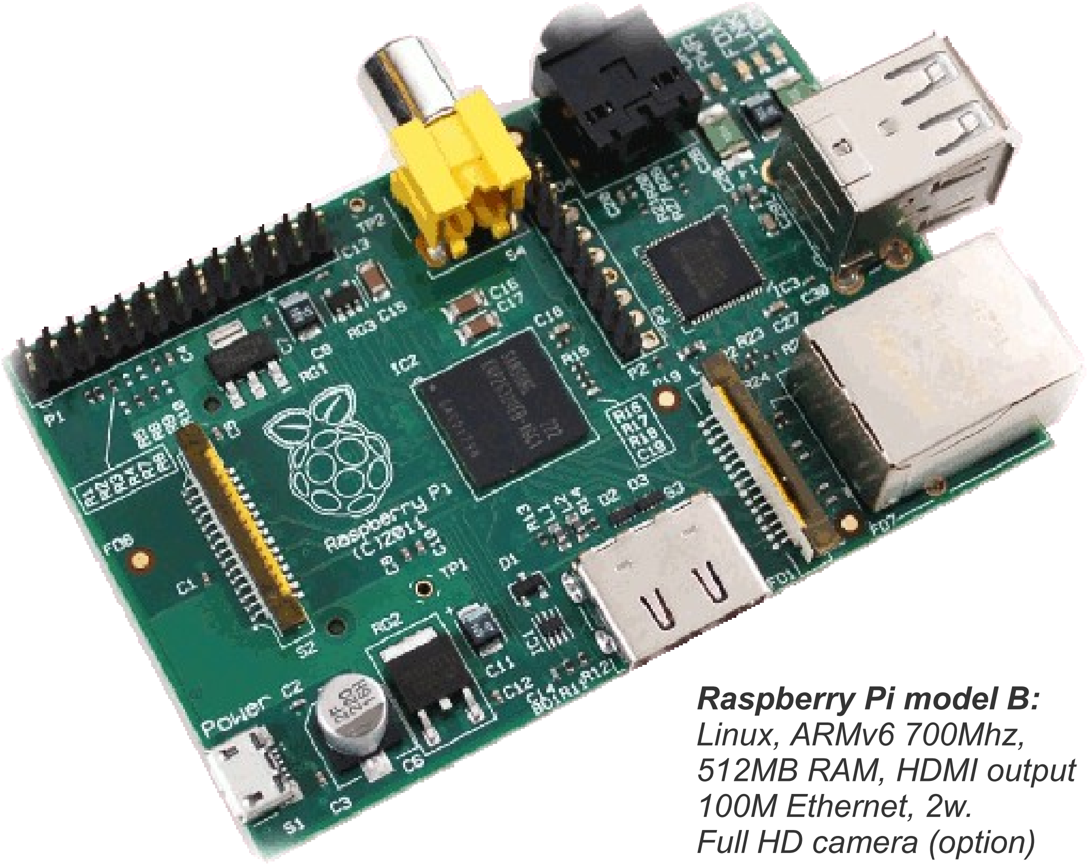
LPro G2EBI: UE3.3

IT & Networks

http://camsi.ups-tlse.fr

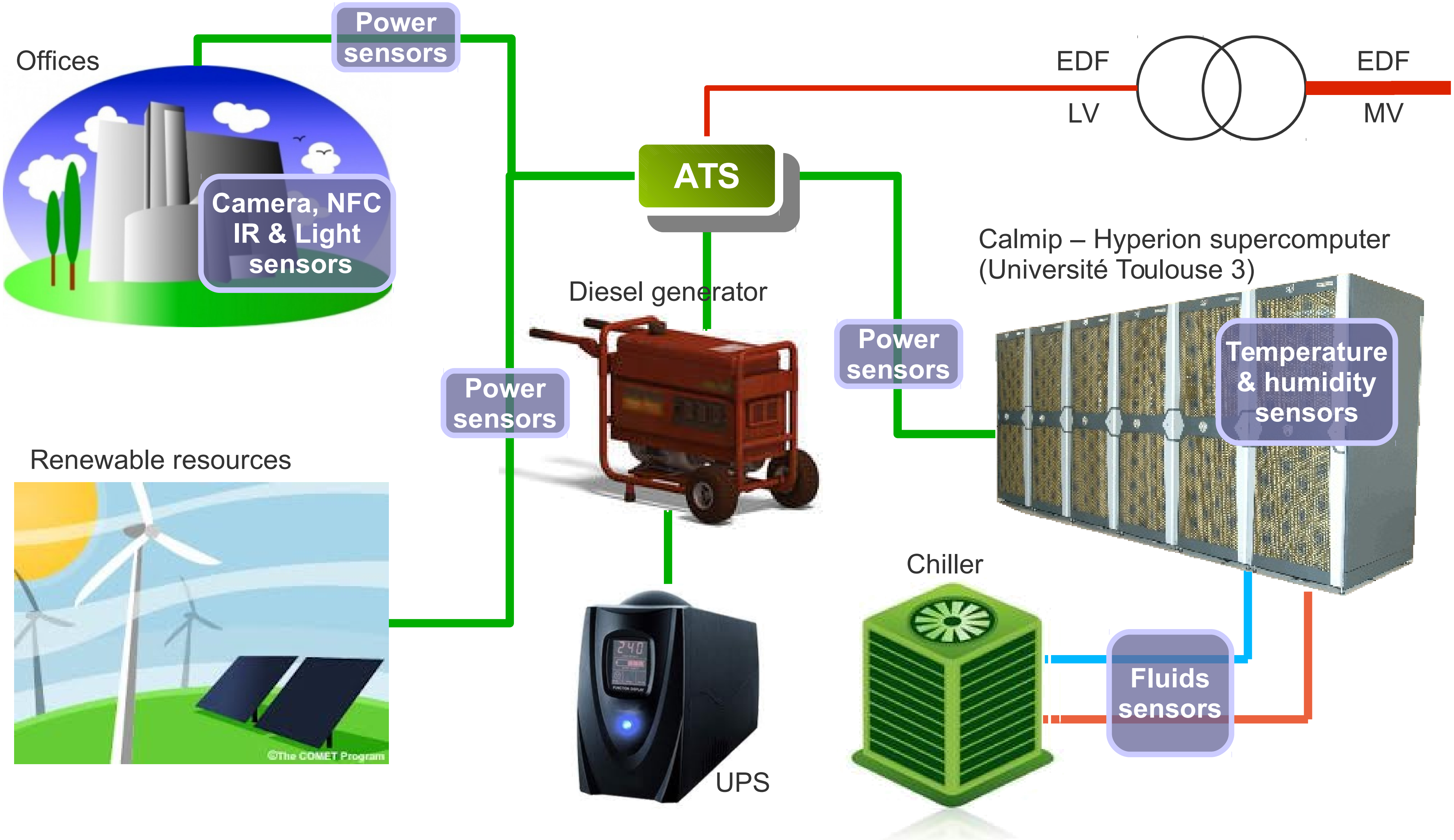
Thiebolt François | IRIT

thiebolt@irit.fr



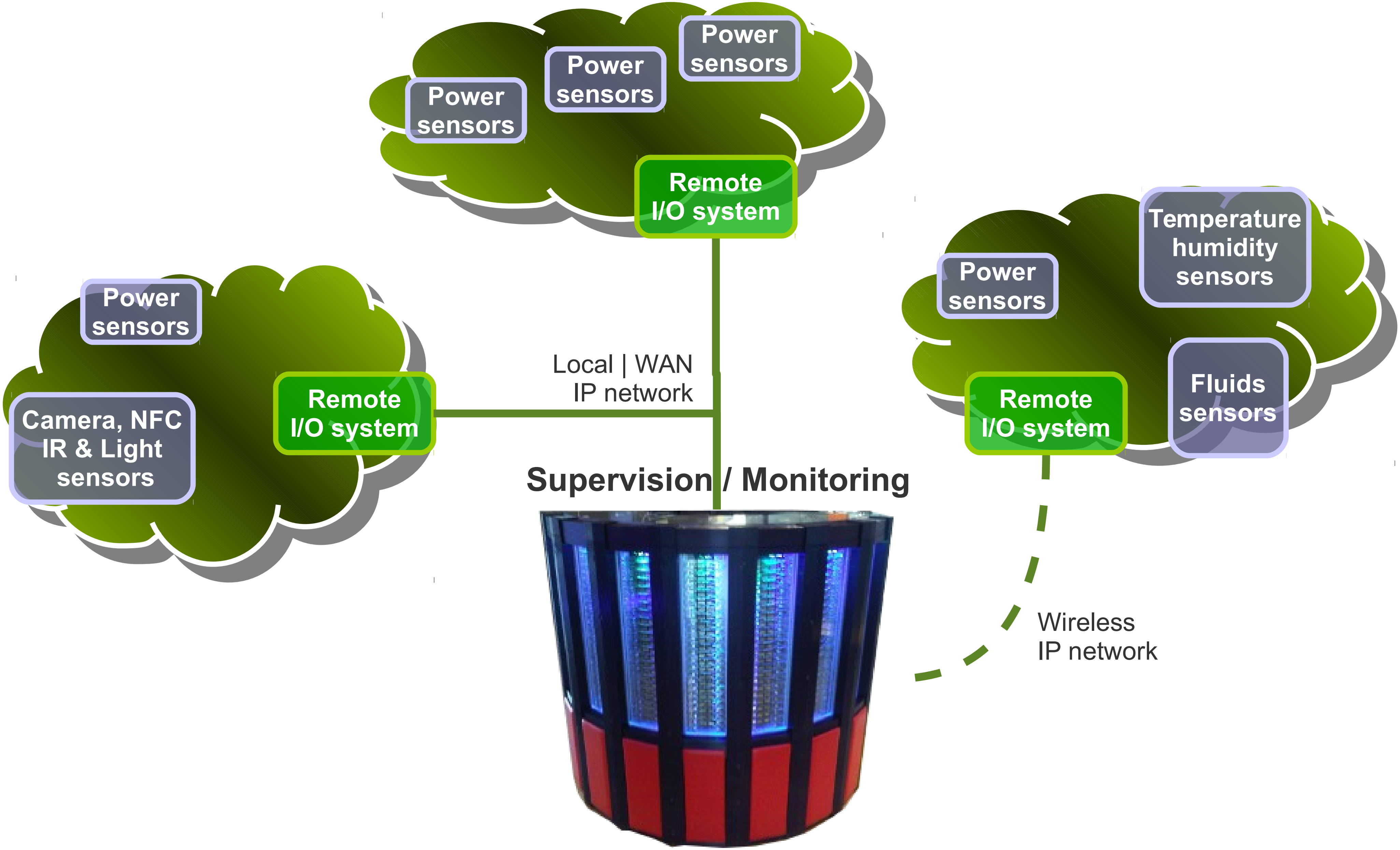
Raspberry Pi model B:
Linux, ARMv6 700Mhz,
512MB RAM, HDMI output
100M Ethernet, 2w.
Full HD camera (option)

Overview

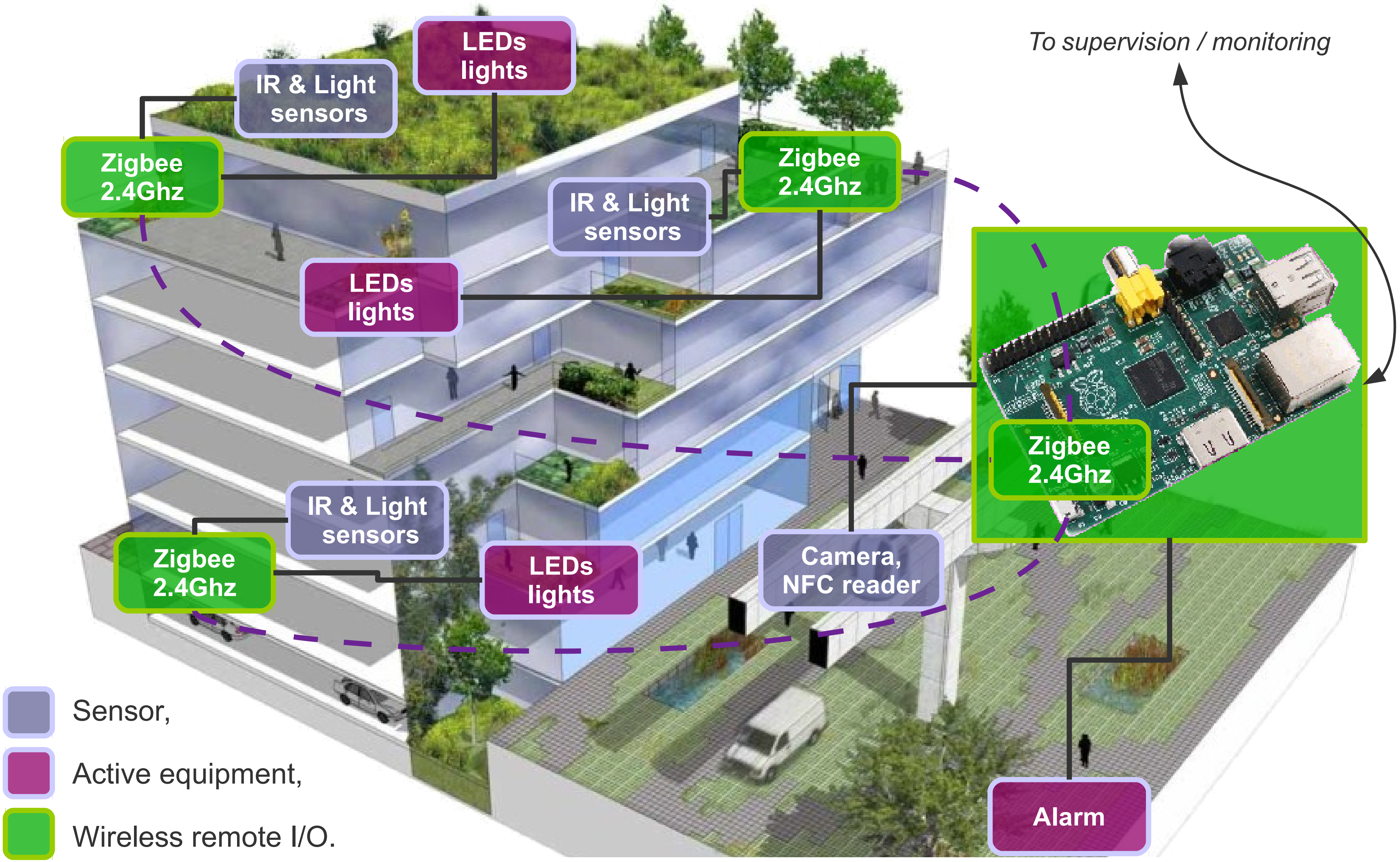


A microgrid is a localized grouping of electricity generation, energy storage, and loads that normally operates connected to a traditional centralized grid (macrogrid). This single point of common coupling with the macrogrid can be disconnected. The microgrid can then function autonomously. [Wikipedia]

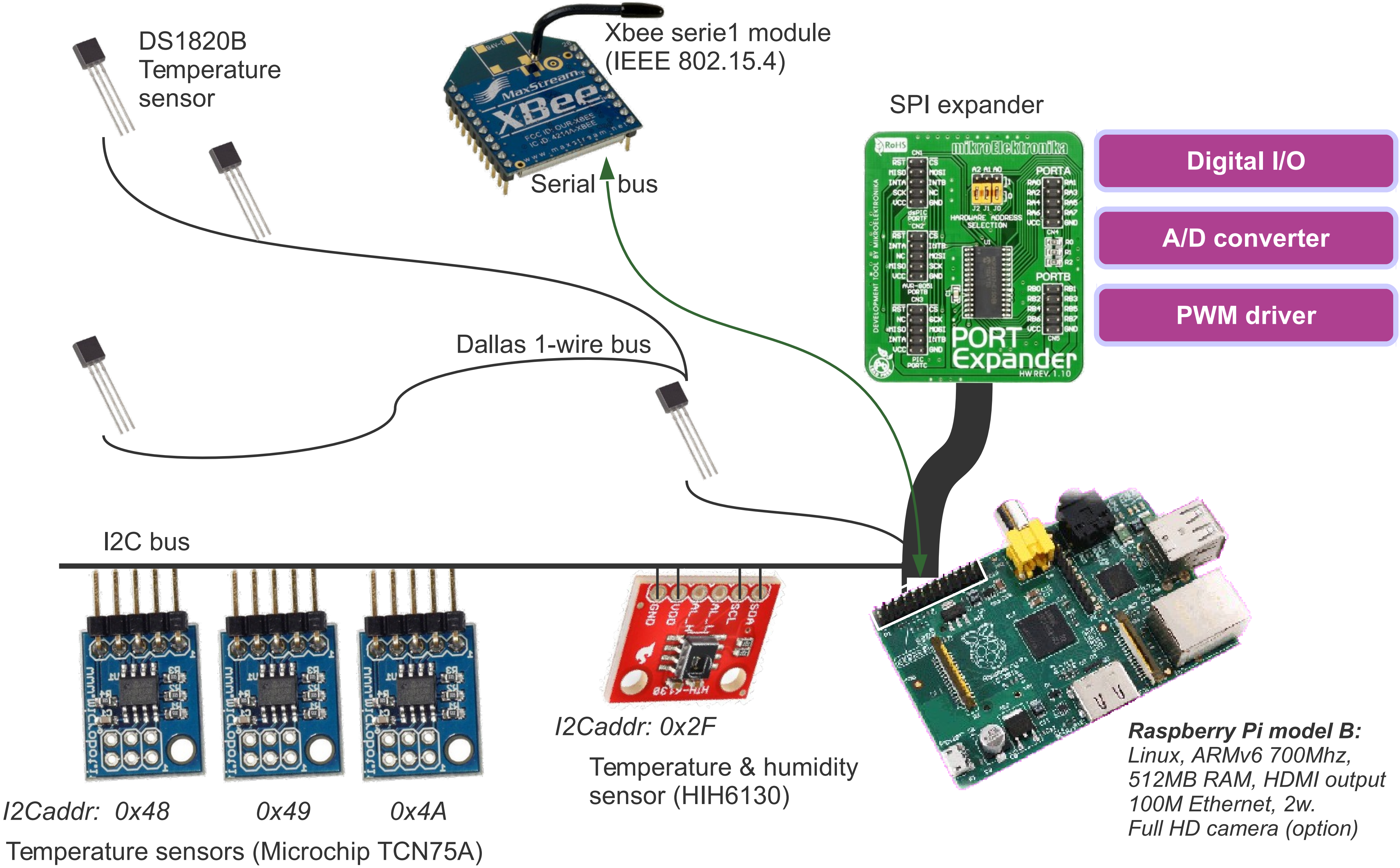
Overview


Power sensors
Power sensors
Power sensors
Remote I/O system
Power sensors
Camera, NFC IR & Light sensors
Remote I/O system
Local | WAN IP network
Power sensors
Temperature humidity sensors
Fluids sensors
Remote I/O system
Supervision / Monitoring
Wireless IP network

Overview


LEDs lights
IR & Light sensors
Zigbee 2.4Ghz
Zigbee 2.4Ghz
IR & Light sensors
LEDs lights
To supervision / monitoring
Zigbee 2.4Ghz
IR & Light sensors
Zigbee 2.4Ghz
LEDs lights
Camera, NFC reader
Alarm
🟪 Sensor,
🟪 Active equipment,
🟩 Wireless remote I/O.

Overview



DS1820B Temperature sensor

Xbee serie1 module (IEEE 802.15.4)

Serial bus

SPI expander

Digital I/O

A/D converter

PWM driver

Dallas 1-wire bus

I2C bus

I2Caddr: 0x2F

Temperature & humidity sensor (HIH6130)

Raspberry Pi model B:
Linux, ARMv6 700Mhz,
512MB RAM, HDMI output
100M Ethernet, 2w.
Full HD camera (option)

I2Caddr: 0x48 *0x49* *0x4A*

Temperature sensors (Microchip TCN75A)

Plan

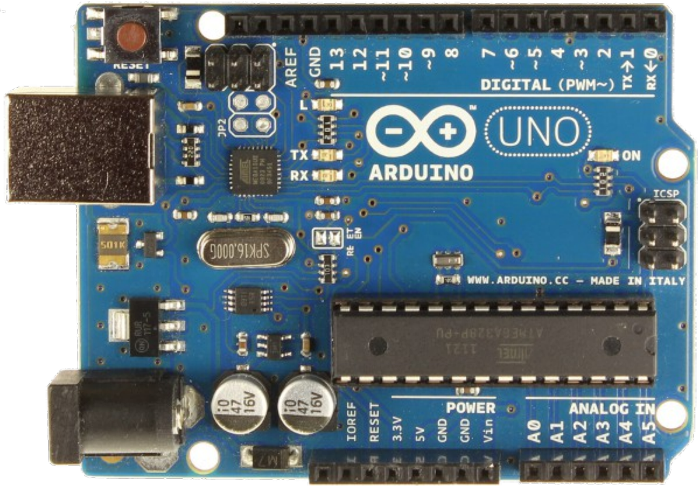
Part I - Principles

- Embedded Systems | Bare & OS-powered boards,
- Chip-level communications | I2C, SPI, Dallas 1-wire,
- Introduction to Python,
- Introduction to Raspberry Pi,
- Python @ Raspberry Pi | application to the I2C bus.

Embedded systems

- Bare boards

Arduino → Atmel AVR chips + simple & efficient libraries + CrossAVR IDE,
PIC → famous µcontrolers from Microchip,
ARM boards from Atmel, Embedded Artist etc etc etc
…………



http://www.arduino.cc

+

Very (very) cheap (may requires only main chip + some passives),
Simple and efficient libraries to use,
Full control over the execution runtime,
Low cost development boards.

−

Usually requires JTAG or some special link to update code,
Difficult to add, for example, an SSH server,
Mainly binary code (e.g. Python interpreter is very unlikely).

Embedded systems

- Smart Boards (OS-powered embedded systems):

Raspberry Pi → ARMv6, Linux powered (Android soon)
Beaglebone → ARMv7 (Cortex A8), Linux & Android powered
… and some boards running Windows Embedded version

+

Low-cost solution (starting from 2012 for Rpi & Beagleboard),

Fully featured systems with virtually everything (apache server, java and python interpreters, dhcp server, SSH server …)

Full range of powerful libraries and bindings like I2C/SMBus for python,

Easy to update either your code or the system wherever you are.

−

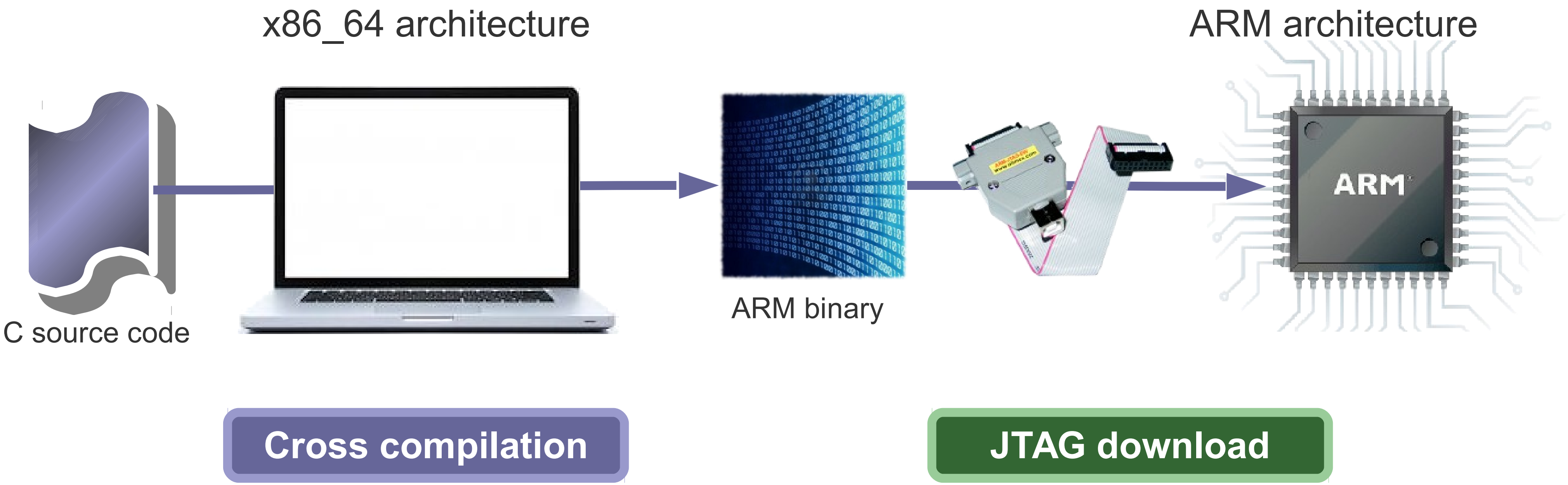
High complexity boards along with ASIC → impossible to build your own board unless you're a professional,

Except with real-time kernel, execution runtime will experience variability.

Cross Compilation

- Generate code for bare boards: cross compilation tool-chain

Build binaries for an ARM architecture from a x86_64 computer.



Cross Compilation

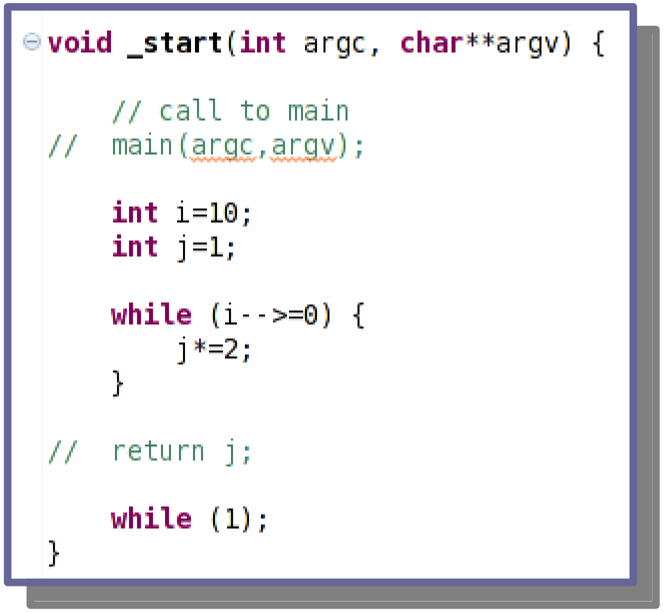
- Example: generate binaries for ARM

Note: we do not use any of the default C libraries nor the low-level initialisation assembly code.

To download code through JTAG, generated binary may need to be converted to either DWARF, S3 or ELF format:

```
#> arm-eabi-objdump <options> <file>
```

Test.c



```
void _start(int argc, char**argv) {

     // call to main
//   main(argc,argv);

     int i=10;
     int j=1;

     while (i-->=0) {
         j*=2;
     }

//   return j;

     while (1);
}
```

```
08:50:52 **** Build of configuration Debug for project simpleTest ****
make all
Building file: ../test.c
Invoking: Cross GCC Compiler
arm-eabi-gcc -O0 -g3 -Wall -c -fmessage-length=0 -MMD -MP -MF"test.d" -MT"test.d" -o "test.o" "../test.c"
Finished building: ../test.c

Building target: simpleTest
Invoking: Cross GCC Linker
arm-eabi-gcc -nostartfiles -nodefaultlibs -nostdlib -o "simpleTest"  ./test.o
Finished building target: simpleTest


08:50:54 Build Finished (took 1s.374ms)
```

Smart boards

- Running code on a smart board

Launch Python code through a SSH connexion to a distant board.

```
thiebolt@metis[~] ssh root@raspicam
Linux raspicam1 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct  8 23:30:36 2013 from frontal.amilab.irit.fr
root@raspicam1[~] l
total 76K
8.0K -rwxr-xr-x  1 root root 4.5K Jun 21 15:57 Adafruit_I2C.py
4.0K -rw-r--r--  1 root root 2.2K Sep 29 23:24 enable_second_I2C.py
4.0K -rwxr-xr-x  1 root root 2.1K Oct  3 14:39 recs_temp_sensors.py
 20K -rwxr-xr-x  1 root root  18K Oct  9 00:38 temp_zabbix_RECS.py
root@raspicam1[~] ./recs_temp_sensors.py
I2C: Wrote 0x20 to register 0x01
I2C: Device 0x4D returned the following from reg 0x00
[21, 128]
Current temperature is 21.50°c ... conversion took 3ms
root@raspicam1[~]
```

```
root@raspicam1[~] cat /proc/cpuinfo
Processor       : ARMv6-compatible processor
rev 7 (v6l)
BogoMIPS        : 697.95
Features        : swp half thumb fastmult vfp
edsp java tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xb76
CPU revision    : 7

Hardware        : BCM2708
Revision        : 000e
Serial          : 000000006b04abe9
```

Plan

Part I - Principles

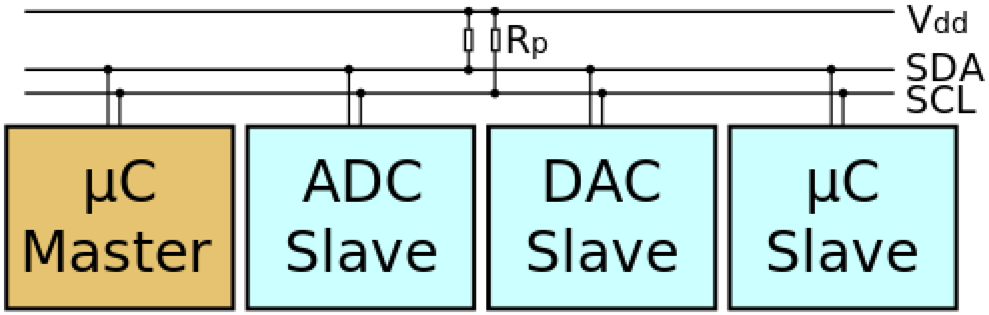
- Embedded Systems | Bare & OS-powered boards,
- Chip-level communications | I2C, SPI, Dallas 1-wire,
- Introduction to Python,
- Introduction to Raspberry Pi,
- Python @ Raspberry Pi | application to the I2C bus.

I2C bus

- Inter-Integrated Circuit, a chip to chip communication bus ...
  - Designed by Phillips in the 80's | first version 100KHz in 1982,
  - Master / Slaves relationship,
  - Up to 128 slaves,
  - Serial bus, 2 wires: SDA & SCL with pull-up resistors,
  - Half duplex,
  - VCC ranges from 2.7v to 5.0v (RPi is only 3.3v!),
  - 400 kbit/s Fast mode (1992), 1 Mbit/s Fast mode plus or Fm+, and 3.4 Mbit/s High Speed mode,
  - Partially user-configurable devices address,
  - 400 pf max. capacitance → with cat. 5E cable 17pf/m ==> up to 7 meters.

http://www.atmicroprog.com/cours/I2C/i2c.php

I2C bus



Typical I2C devices interconnect.

- SMBus: a lightweight I2C compatible protocol

• SMBus definied by intel in 1995 is derived from I2C.

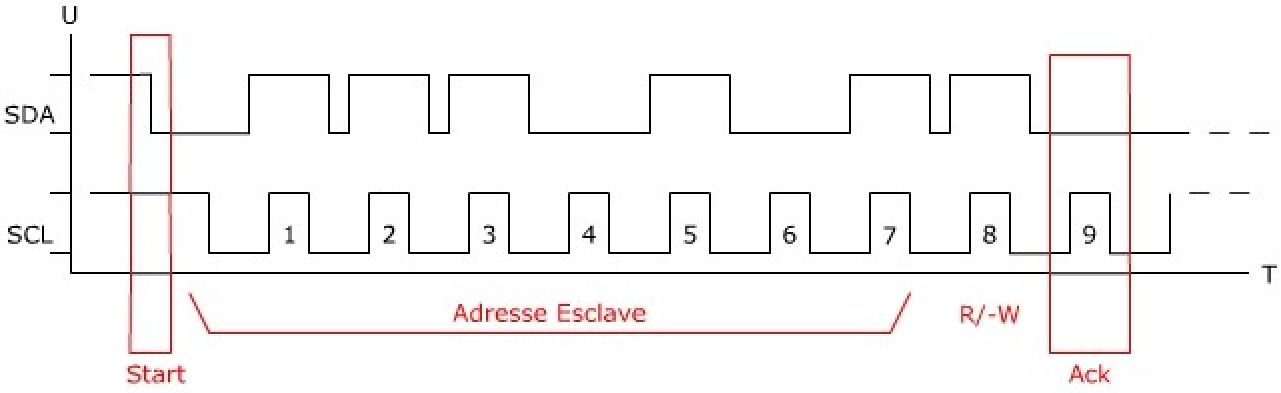
• SMBus is a single-ended simple two-wire bus for the purpose of lightweight communication.

• SMBus' clock frequency range is 10 kHz to 100 kHz. Voltage levels and timings are I2C compatible hence I2C and SMBus devices are often successfully mixed on the same bus.

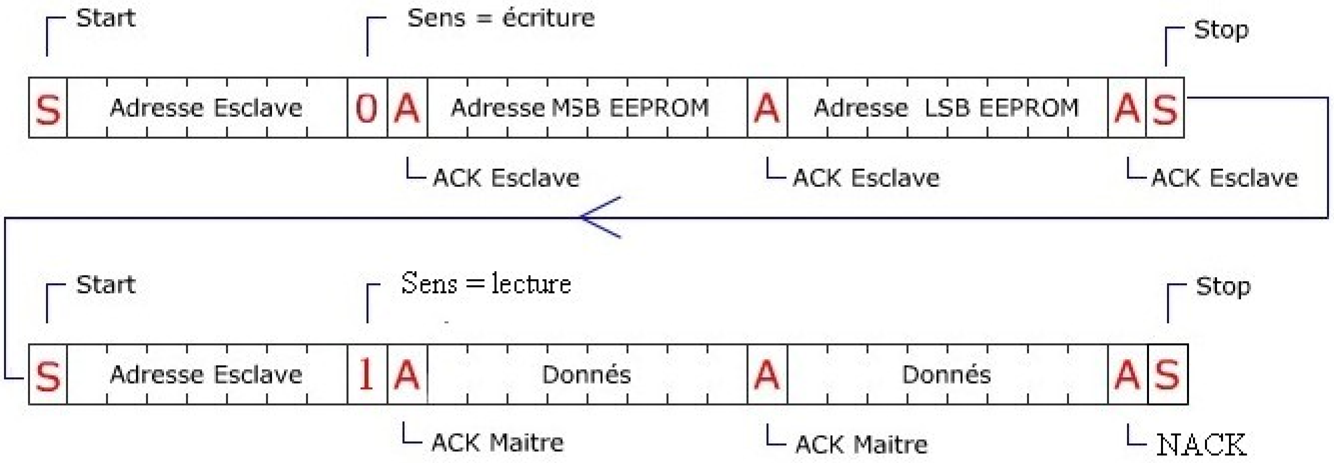
• Most of the time, micro-controllers feature an SMBus with extended capabilities that looks like an I2Cbus. It is commonly named I2C/SMBus.

I2C bus

- I2C device addr = 7 bits + 1 RW bit
  RW bit: '0' means write and '1' read.



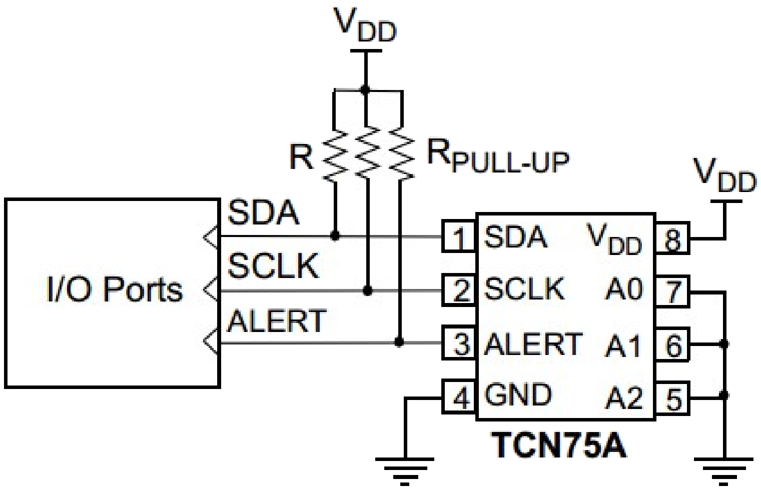
- Protocol: Indirect access to registers within device, e.g write to an eeprom.



I2C bus

- I2C devices usually features 1 to 3 bits (sometimes 4 bits) for user-configurable device address.
- Others address bits are factory-programmed.

Example: TCN75A, an I2C temperature sensor



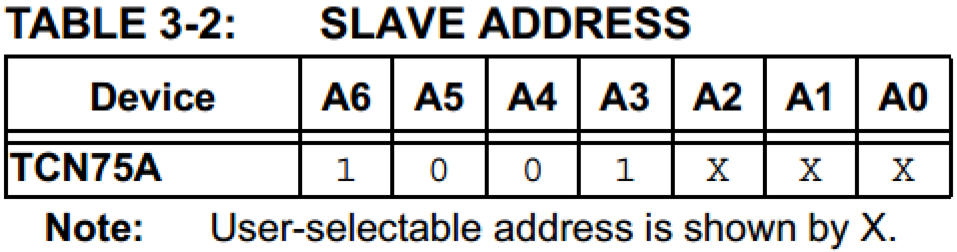


TABLE 3-2: SLAVE ADDRESS

Device	A6	A5	A4	A3	A2	A1	A0
TCN75A	1	0	0	1	X	X	X

Note: User-selectable address is shown by X.

- Up to 8 devices,
- I2C Address range from 0x48 to 0x4F.

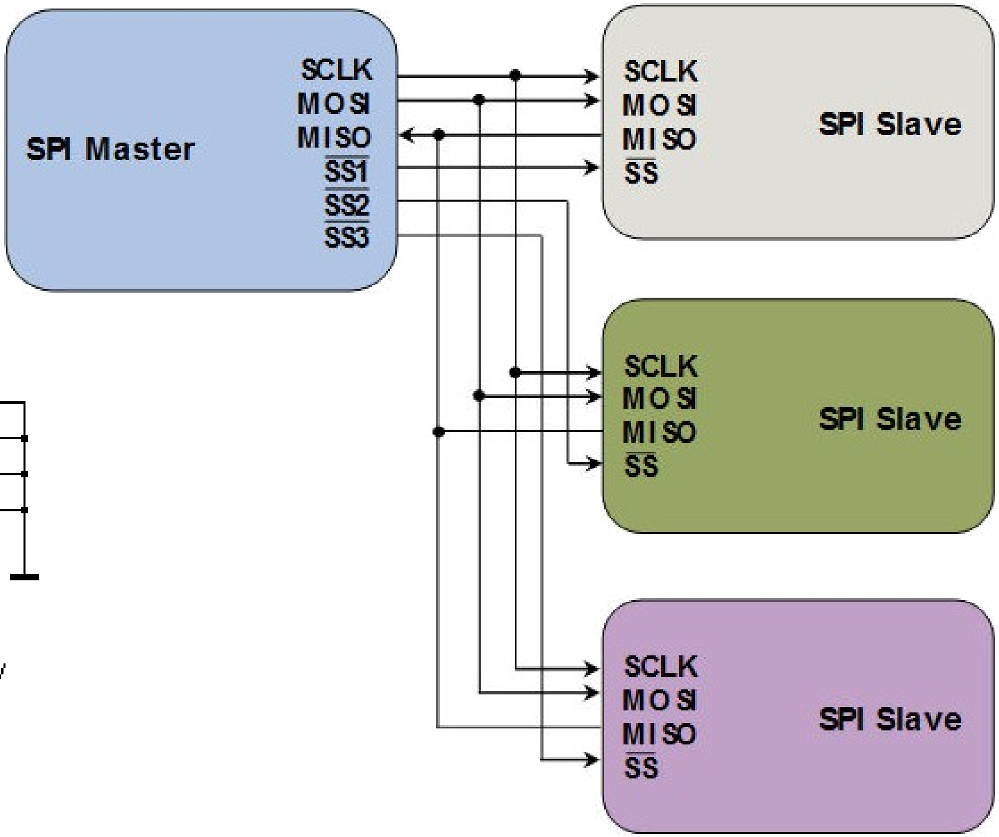
SPI bus

- Serial Peripheral Interface
  - Like I2C, designed by the beginning of the 80's but for higher throughput,
  - Master / Slaves relationship,
  - Serial bus with 3 wires: MISO, MOSI & SCLK,
  - Full duplex,
  - VCC ranges from 2.7v to 5.0v (RPi is only 3.3v!),
  - Intended to PCB (short wires),
  - On Raspberry Pi, SPI speed = APB* core clock (250 Mhz) / 2 to 32768,
  - Chip Select pins (CS) are generated by the Master to select one device at time,
  - On Raspberry Pi, 2 CS signals available.

**ARM Peripheral Bus*

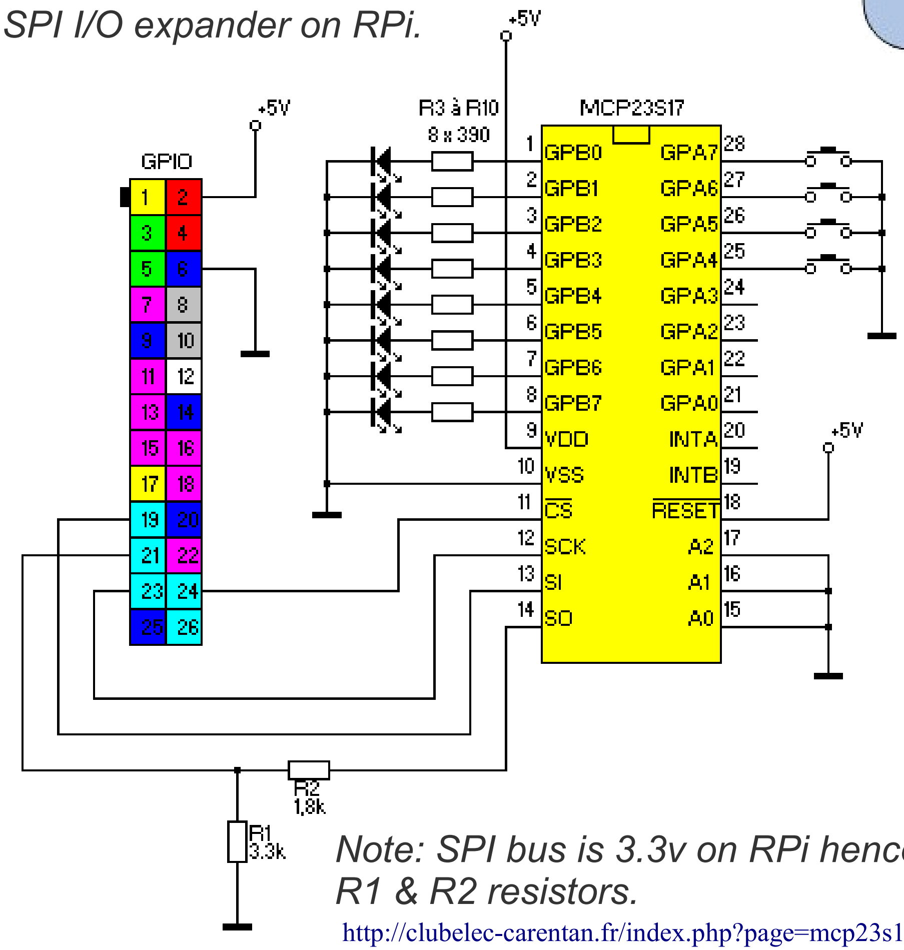
http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/

SPI bus



Typical SPI devices interconnect.

SPI I/O expander on RPi.



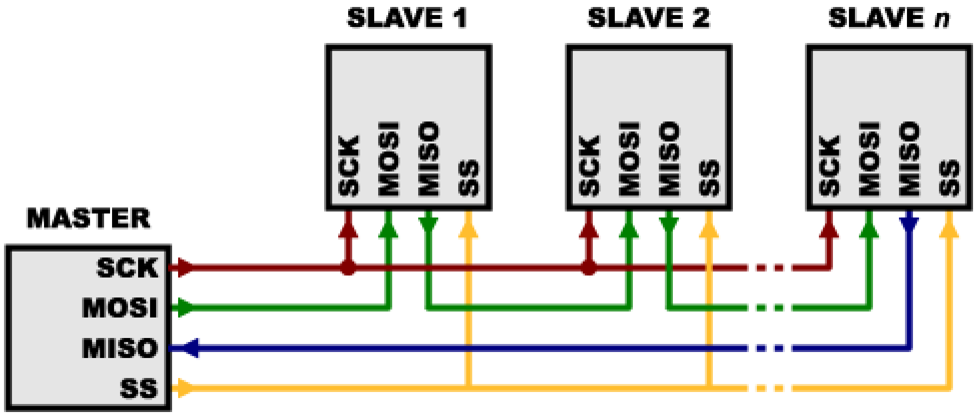
Note: SPI bus is 3.3v on RPi hence the R1 & R2 resistors.

http://clubelec-carentan.fr/index.php?page=mcp23s17

SPI bus

- Each SPI device requires a dedicated CS line,
- Things become tricky when having a large number of SPI devices …

SPI daisy-chain mode



- Only one CS line for multiple slaves,
- Rely on slave's internal shift registers which propagates MOSI → MISO as long as CS remains active,
- On rising-edge of CS, each slave executes command in its input buffer. This way, all slaves may execute a different command.

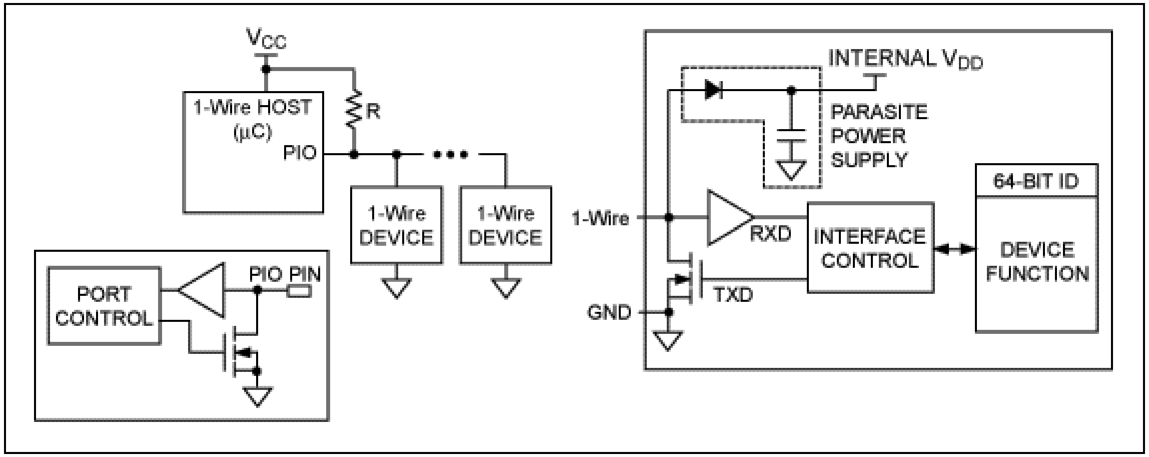
Infineon: SPI interface used in daisy-chain
https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/slave-select-ss

Dallas 1-wire bus

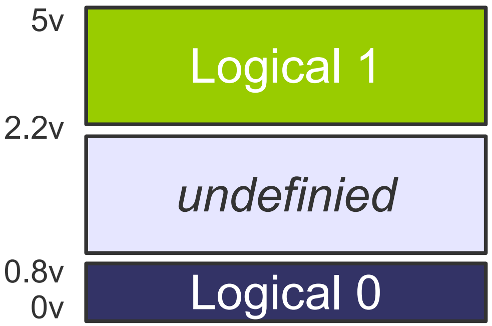
- Dallas 1-wire bus
  - Designed by Dallas Semiconductor (now Maxim),
  - Lower data rates than I2C but longer range,
  - Typically used to communicate with low cost devices like temperature sensors,
  - 1-wire devices include an 800 pF capacitor to store charge, and power the device during periods when the data line is active (parasitic mode),
  - Pull-up resistor on data line to power devices. Devices and Master exhibit an open drain,
  - VCC ranges from 2.7v to 5.25v,
  - 15.4 kbps (standard) to 125 kbps (overdrive),
  - Slave device has a unique, unalterable, factory-programmed, 64-bit ID,
  - Use twisted cable,
  - Up to 200m with regular pull-up resistor and up to 500m with active termination.

http://www.maximintegrated.com/app-notes/index.mvp/id/148
http://www.maximintegrated.com/app-notes/index.mvp/id/1796

Dallas 1-wire bus

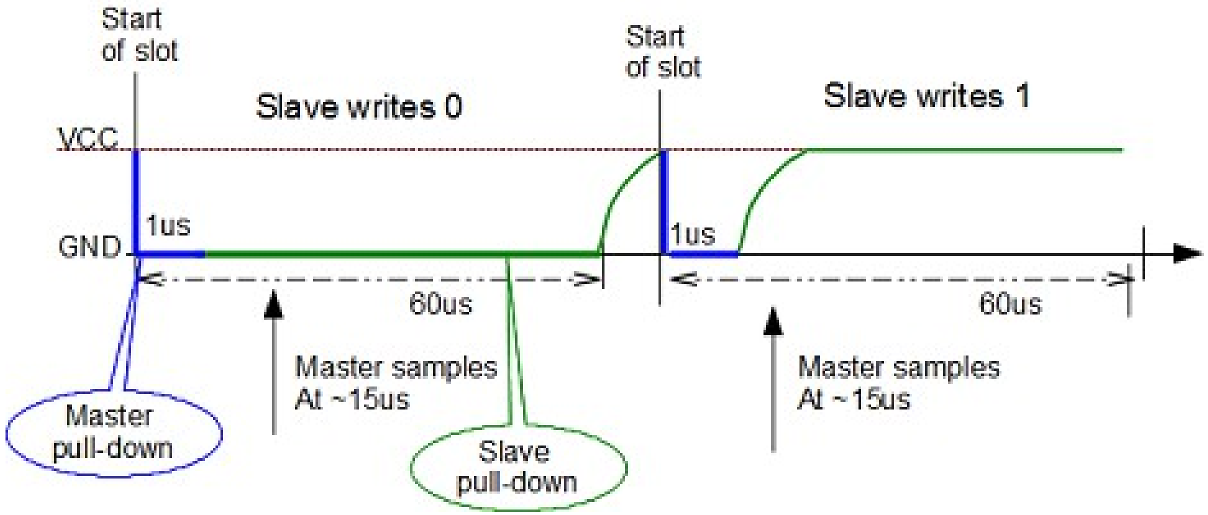


1-wire typical interconnect & slave internals.

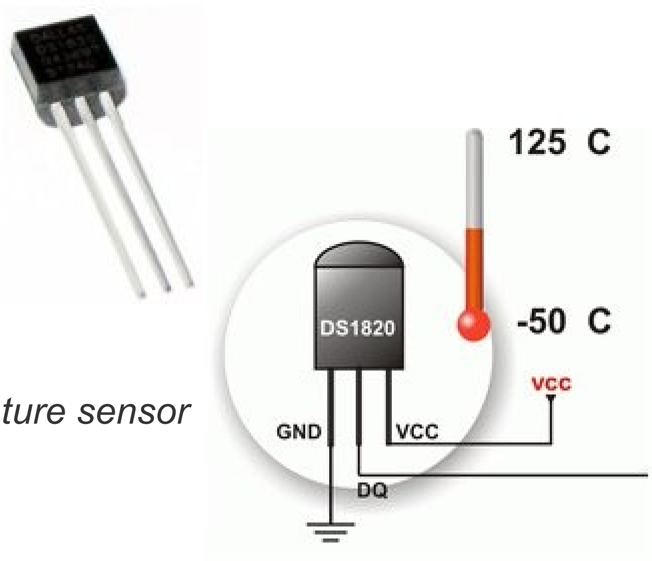


1-wire logical levels.

- Concept of time slot: 60µs (standard) & 8µs (overdrive),
- 0 and 1 are encoded within timeslot.



Dallas 1-wire bus



1-wire temperature sensor Dallas 1820B.

iButton is 1-wire powered.

- 1-Wire on Raspberry Pi

  There's no 1-Wire master device on RPi. Instead one GPIO pin + pull-up resistor is used to drive the line. The **w1-gpio** kernel module implement the 1-Wire protocol (Bit banging).

http://blog.gegg.us/2013/03/4-different-methods-of-1-wire-access-on-raspberry-pi/

Chip-level buses: synthesis

- Comparison table

Name	Wires, Duplex	CLK / Speed	Length	Notes
I2C/SMBus	Two wires, Half duplex	400 Khz 1 Mhz	Up to 7 meters	One master, up to 128 slaves, pull-up resistors
SPI	Three wires, Full duplex	Chip dependant, 10 Mhz std.	PCB intended	One master
Dallas 1-Wire	One wire, Half duplex	15.4 kbps (std) 125 kbps	May reach 500-700m with ad-hoc termination	One master, fixed devices addr.
CAN	Two wires (differential), Half duplex	1 Mbits/s	40m (1 Mbits/s), 500m (125 kbits/s) 6km (10 kbits/s)	Multi-master, CSMA/CD, 120 ohm termination

Note: Of course, all these bus require an additional wire for GND ;)

Plan

Part I - Principles

- Embedded Systems | Bare & OS-powered boards,
- Chip-level communications | I2C, SPI, Dallas 1-wire,
- Introduction to Python,
- Introduction to Raspberry Pi,
- Python @ Raspberry Pi | application to the I2C bus.

Python

- Introduction to the Python language
  - Open-source general-purpose language,
  - Object-oriented, procedural, functional,
  - Interpreted language (i.e not for bare boards),
  - Widely available for almost all platforms (Mac, Windows, Linux etc...),
  - Available python interactive interpreter,
  - Python 2.7

Type in a shell ...

```
ssh -p 2220 lg2ebi@camsi.ups-tlse.fr
Passwd: <secret>
lg2ebi@camsi[~] python
Python 2.6.6 (r266:84292, Feb 21 2013, 19:26:11)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)] on linux2
Type "help", "copyright", "credits" or "license" for more
information.
>>>
```

Python

Python scripts

To execute python code directly from files:

1. create a script file named `<xxx>.py`,
2. add it execution capability bit.

```
touch script.py
chmod a+x ./script.py
```

A Python script ought to contain these first two lines

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#
<code>
```

Alternatively, first line may be `#!/usr/bin/python`

… let's switch to Python tutorial
(Python_tutorial.pdf)

Python exercises

- Some simple problems to be solved with Python ...

Bubble sort

```
raw_data = [ 52, 17, 23, 5, 19, 4 ]
output_list = [ 4, 5, 17, 19, 23, 52 ]
```

Sum of the multiples of 3 or 5

```
If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9.
The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.
```

Fibonacci number

```
Each element is the sum of the previous two. List starts with 0 and 1:
Fibonacci = [ 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 89, 144, ... ]

Display Fibonacci number up to 2000.
```

Mathematical problems to solve with a computer http://projecteuler.com

[Advanced] Python

- Timer in Python | Introduction to multi-threading

… or the art of running simultaneous instances of a same portion of code

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
#

import threading

def do_every (interval, worker_func, iterations = 0):
    if iterations != 1:
        threading.Timer (
                        interval,
                        do_every, [interval, worker_func, 0 if iterations == 0 else iterations-1]
                        ).start ();

    worker_func ();

def print_hw ():
    print "hello world";

def print_so ():
    print "stackoverflow"


# call print_so every second, 5 times total
do_every (1, print_so, 5);

# call print_hw two times per second, forever
do_every (0.5, print_hw);
```

Plan

Part I - Principles

- Embedded Systems | Bare & OS-powered boards,
- Chip-level communications | I2C, SPI, Dallas 1-wire,
- Introduction to Python,
- Introduction to Raspberry Pi,
- Python @ Raspberry Pi | application to the I2C bus.

Raspberry Pi

- Raspberry Pi … a revolution ?
  - Fully-featured embedded system with Linux* (Android on way),
  - Powerful ARMv6 @ 700Mhz,
  - GPU | H264 encode/decode → Full HD grade display and camera,
  - Type B: Ethernet 100Mbps, 512MB ram,
  - Plenty of available I/O, expansion shields** (e.g PiFace),
  - Promoted by a non-profit foundation,
  - 2W and very cheap!,
  - Huge community!
- but ...
  - Not totally open-hardware nor open-source,
  - blob driven GPU (Binary Large OBject),
  - No PXE boot.

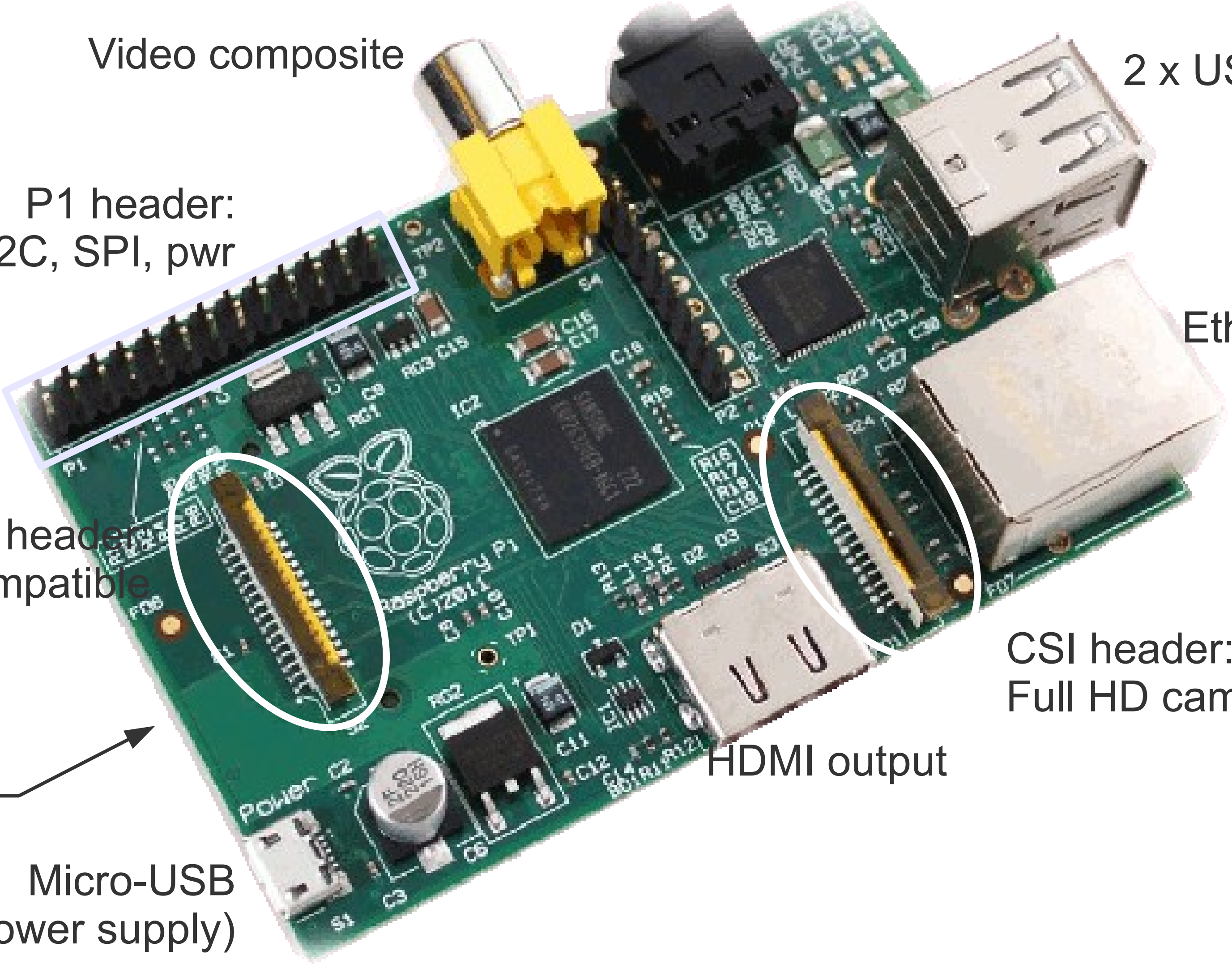
** there exists several Linux distributions for this board like Raspbian, Pidora, Arch Linux … and NOOBS which let you deciding what system to install (keyboard and HDMI display required).*

Unless otherwise specified, we'll use the Raspbian distribution (Linux Debian based on).
http://www.raspberrypi.org/downloads

*** arduino terminology*

Raspberry Pi

- I/O interfaces



Sound

Video composite

2 x USB2.0

P1 header:
Serial, GPIO, I2C, SPI, pwr

Ethernet 100Mbps

DSI header:
Display Port compatible

CSI header:
Full HD camera (option)

SD Card
slot below

HDMI output

Micro-USB
(power supply)

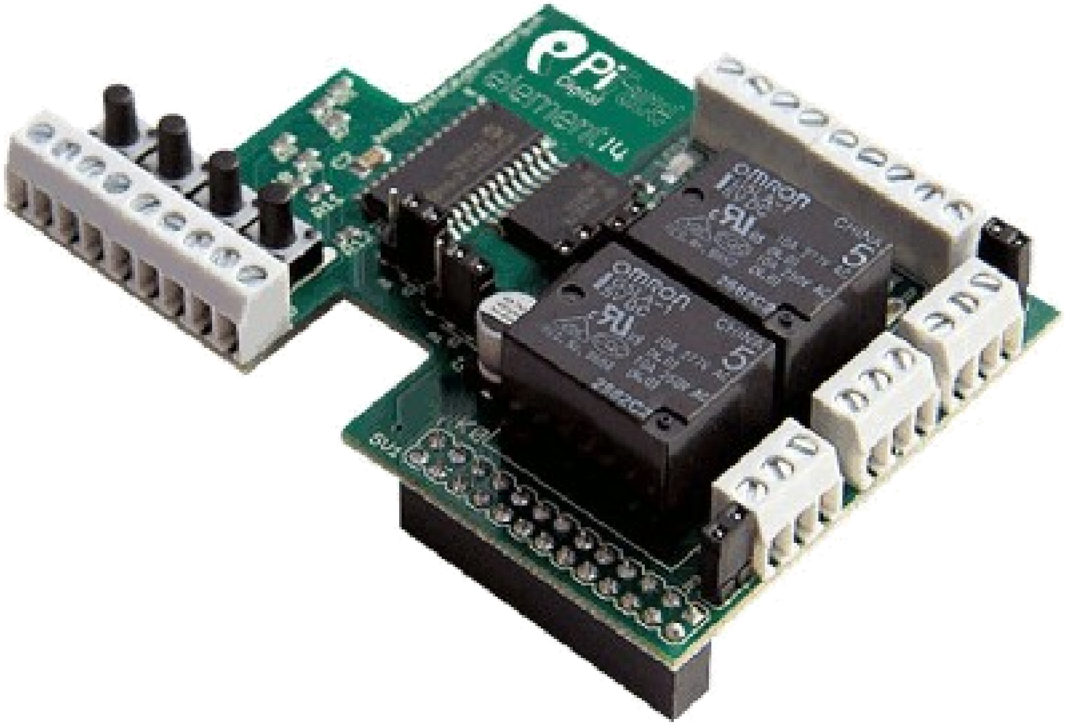
Raspberry Pi

- P1 header details

3.3V	1	2	5V
I2C0 SDA	3	4	DNC
I2C0 SCL	5	6	GROUND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21	13	14	DNC
GPIO 22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Note: DNC stands for GND

- PiFace expansion board



This board provides 4 digital input coupled with on-board switches along with 8 digital output coupled with on-board leds. Two of the eight outputs are also tied with 5v relay coils to drive 230v loads.

Raspberry Pi

- SSH connexion to a Raspberry Pi board

```
<etudiant> ssh -p 2220 lg2ebi@camsi.ups-tlse.fr
Passwd: <secret>
lg2ebi@camsi[~] ssh root@raspicam
Linux raspicam1 3.6.11+ #538 PREEMPT Fri Aug 30 20:42:08 BST 2013 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct  8 23:30:36 2013 from frontal.amilab.irit.fr
root@raspicam1[~] l
total 76K
8.0K -rwxr-xr-x  1 root root 4.5K Jun 21 15:57 Adafruit_I2C.py
4.0K -rw-r--r--  1 root root 2.2K Sep 29 23:24 enable_second_I2C.py
4.0K -rwxr-xr-x  1 root root 2.1K Oct  3 14:39 recs_temp_sensors.py
 20K -rwxr-xr-x  1 root root  18K Oct  9 00:38 temp_zabbix_RECS.py
root@raspicam1[~] ./recs_temp_sensors.py
I2C: Wrote 0x20 to register 0x01
I2C: Device 0x4D returned the following from reg 0x00
[21, 128]
Current temperature is 21.50°c ... conversion took 3ms
root@raspicam1[~]
root@raspicam1[~] cat /proc/cpuinfo
Processor       : ARMv6-compatible processor rev 7 (v6l)
BogoMIPS        : 697.95
Features        : swp half thumb fastmult vfp edsp java tls
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xb76
CPU revision    : 7

Hardware        : BCM2708
Revision        : 000e
Serial          : 000000006b04abe9
```

Raspberry Pi

- Board configuration | the `raspi-config` command

  - Enable / disable camera support,
  - Set hostname,
  - Password change,
  - Boot options,
  - Extend filesystem up to the whole SDcard,
  - Enable / disable SSH server [default on],
  - Overclocking,
  - CPU | GPU memory split,
  - ...

```
root@raspicam1[~] cat /proc/meminfo
MemTotal:          383712 kB
MemFree:           196820 kB
Buffers:            48028 kB
Cached:            111360 kB
SwapCached:             0 kB
Active:            121756 kB
Inactive:           47196 kB
Active(anon):        9592 kB
Inactive(anon):       164 kB
Active(file):      112164 kB
Inactive(file):     47032 kB
Unevictable:            0 kB
Mlocked:                0 kB
SwapTotal:         102396 kB
SwapFree:          102396 kB
Dirty:                  8 kB
Writeback:              0 kB
AnonPages:           9576 kB
Mapped:              7344 kB
Shmem:                196 kB
Slab:               10592 kB
SReclaimable:        7560 kB
SUnreclaim:          3032 kB
KernelStack:         1016 kB
PageTables:           580 kB
NFS_Unstable:           0 kB
Bounce:                 0 kB
WritebackTmp:           0 kB
CommitLimit:       294252 kB
Committed_AS:       79780 kB
VmallocTotal:      630784 kB
VmallocUsed:          780 kB
VmallocChunk:      416228 kB
```

Note: the `raspi-config` command comes from the Linux Raspbian distribution.