The Internet of Things: Prototyping Boards & Languages

Course website: http://site.unibo.it/iot

Prof. Luciano Bononi
luciano.bononi@unibo.it

Prof. Marco Di Felice
marco.difelice3@unibo.it

Luca Sciullo
luca.sciullo@unibo.it

MASTER DEGREE IN COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF BOLOGNA, ITALY
Smart Things

- IoT world is made by **smart things**
- A smart thing is a physical object digitally augmented with one or more of the following:
  - (smart) **Sensors** (temperature, light, motion, and so on)
  - (smart) **Actuators** (displays, sound, motors, and so on)
  - **Computation** (can run programs and logic)
  - **Communication** interfaces (wired or wireless)
Smart Things
Smart Things

Architecture of a **Smart Object**
Smart Sensors

- A sensor is a device that is able to detect events or changes in its physical environment.
- A smart sensor is a device capable of measuring analog inputs from the physical environment and making them digital by using some built-in resources.

https://www.adafruit.com/
Smart Sensors

Sensors are getting cheaper and cheaper...


Sensors are getting smaller and smaller...

PROTOTYPING BOARDS AND LANGUAGES
L. BONONI, M. DI FELICE, L. SCIULLO, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF BOLOGNA, ITALY
Smart Sensors

UNDERWATER ACUSTIC SENSOR NETWORKS

UNDERGROUND SENSOR NETWORKS

WEARABLE SENSORS + NANOSENSORS

Source: http://www.ece.neu.edu/wineslab/
Btw, current smartphones embed a multitude of sensors that provide context-aware information about users.

- Iris (motion) sensor
- Pressure sensor
- Accelerometer
- Barometer
- Gyroscope
- Geomagnetic sensor
- Hall sensor
- HR sensor
- Proximity sensor
- RGB Light sensor
- ...

3D Indoor localization
Transportation mode detection
Human activity detection
Human presence detection
Health condition detection
Gesture tracking
...

MINING & LEARNING TECHNIQUES
Smart Sensors

확성기 are mostly kinds of transducers, i.e. they convert one form of energy (electrical or not) into another (electrical or not).

Negative Temperature Coefficient → Variable resistor that changes its resistance with change of the temperature: the resistance decreases with increase of the temperature.

Source: https://home.roboticlab.eu/
Smart Sensors

✧ **Sensors** are mostly kinds of **transducers**, i.e. they convert one form of energy (electrical or not) into another (electrical or not).

**EXAMPLE: ACCELEROMETER SENSOR**

✧ **Accelerometers** are electromechanical devices that sense either static or dynamic forces of acceleration.
✧ **Accelerometers** can measure acceleration on 1, 2 or 3 axes.
✧ **Acceleration** is derived by the displacement of an **internal spring** (Inertial Equation), often measured via capacitive plates.
Smart Actuators

- An actuator is a device that converts energy into motion.

- A **smart actuator** is a device capable of turning digital inputs into physical actions.

[Link to Adafruit](https://www.adafruit.com/)

[Link to UTT Education](http://www.utteducation.com)
Embedded Systems

- Integrated Microprocessor-based **hardware system** + **software/firmware** that is designed to perform a dedicated function/specific purpose.

Some examples:
- Industrial systems
- Home appliances
- Avionics
- Network devices (e.g. Routers, switches)
- Cache machines
- ...

Source: [1]

Embedded Systems

- **Embedded systems programming** often poses some unique challenges (not so common in general-purpose programming):
  - **Real-time vs non-real time systems**
    - Hard-real time vs soft real-time vs firm real-time
  - **Constrained resources** (CPU, power battery, ...)
  - **Limited memory** for data loading and data operations
  - **Limited memory** for storing the program source code
  - **Limited multitasking** (i.e. thread/process support)


WE WILL PROVIDE EVIDENCES OF THESE CHALLENGES THROUGH A DEMO
IoT Prototyping Boards

Hardware platforms that are most commonly used to build prototypes of IoT projects and IoT embedded systems.

- Validate the feasibility of a product
- Build a proof-of-concept
- Do-It-yourself (DIY) projects
- Pre-production phase of a new product
  - Choose hardware components
  - Deploy and test the software

IoT Prototyping boards can be often be classified according to their hardware/software characteristics into **two main families:**

- Microcontroller-based boards
- Single-board computers

https://mybroadband.co.za
Microcontroller Boards

It is a system on a chip (SoC) that contains processing cores, RAM and EPROM for the storage of custom programs that are executed on the microcontroller. It is a PCB with added circuitry.

- Available on the market since 1975
- Often composed by a single CPU + RAM memory + ROM memory (EPROM, EEPROM, ...)
- Limited processing power (clock speed <100 MHz)
- 8-bit/16-bit/32-bit architectures
- Single program or limited multitasking
- No operating system
Microcontroller Boards

Some examples (available in our lab):

- ARDUINO UNO
- NUCLEO STM32
- ARDUINO NANO IOT
- NODEMCU
- ESP32
- FISHINO UNO
- FISHINO SHARK
Microcontroller Boards

IoT Programming Languages for microcontroller-based boards

- **C/C++** and variants [C tutorial from W3C community](LINK)
- **LUA** [Tutorial: LINK]
- **Wiring** (Arduino, actually a dialect of C/C++)
  Tutorial: Sistemi Embedded Teoria e Pratica [LINK]
- **Micropython**
  Tutorial for different IoT boards [LINK]
- **Javascript** libraries [e.g. CYLON.JS, JOHNNY FIVE]
- ....
Single board Computers

- It is a **complete computer** on a single circuit board, including microprocessor(s), memory, input/output (I/O) and other features required by a functional computer.

- SBC computers typically provide a fan-less, low-power computing solution and a low profile architecture.

https://mybroadband.co.za
Single board Computers

Some examples (available in our lab):

- **RASPBERRY PI ZERO**
- **RASPBERRY PI 3**
- **PINE64**
- **BEAGLEBONE**
EXAMPLE OF MICRO-CONTROLLER BOARD: ARDUINO UNO
The Arduino Board

- Project started in 2003 as a program for students at the Interaction Design Institute (Ivrea, Italy).
- IDEA: Create simple, low cost instruments also for non-engineers, aimed to work on digital projects.
- Around 700,000 units shipped (in 2013).
- Partnership with ARM Holdings (2017).
The Arduino Board

Source: https://www.vellemanformakers.com/the-ultimate-arduino-guide/
### Using Arduino

**Installation & Troubleshooting**
- For problems with Arduino itself, NOT your project
- Last post: Today at 12:27 pm
  - Re: Arduino Mega not booting by husainsa12002

<table>
<thead>
<tr>
<th>Posts</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>88,276</td>
<td>20,625</td>
</tr>
</tbody>
</table>

**Introductory tutorials**
- Tutorials for new people on the forum.
- Last post: Jan 28, 2018, 07:44 pm
  - Re: State Machine and TI... by larry

<table>
<thead>
<tr>
<th>Posts</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>19</td>
</tr>
</tbody>
</table>

**Project Guidance**
- Advice on general approaches or feasibility
- Last post: Today at 01:25 pm
  - Re: DigitalRead multiple... by artistinfra

<table>
<thead>
<tr>
<th>Posts</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>447,172</td>
<td>60,817</td>
</tr>
</tbody>
</table>

**Programming Questions**
- Understanding the language, error messages, etc.
- Last post: Today at 01:30 pm
  - Re: Keyboard button hold... by jake44

<table>
<thead>
<tr>
<th>Posts</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>613,840</td>
<td>75,257</td>
</tr>
</tbody>
</table>
The Arduino Board

- Single-board microcontroller.
- Sets of digital and analog input/output (I/O) pins that can be interfaced to various expansion boards (shields) and other circuits

The Arduino Board

- **Arduino Analog Input**
  - *Resolution*: the number of different voltage levels (i.e., *states*) used to discretize an input signal.
  - Resolution values range from 256 states (8 bits) to 4,294,967,296 states (32 bits).
  - Arduino uses 1024 states (10 bits).
  - Smallest measurable voltage change is $5\text{V}/1024$ or $4.8\text{ mV}$
  - Maximum sample rate is 10,000 times a second.
The Arduino Board

- **Pulse Width Modulation (PWM)**
  - Digital pins can only directly supply 3V or 5V, but they can also pulse the output on and off really fast to produce the same effect of a different voltage supply.
  - The on-off pulsing is so fast that the connected output device “perceives” the result as a reduction in the voltage.

Image from *Theory and Practice of Tangible User Interfaces* at UC Berkley
The Arduino Board

**Fixed cycle length; constant number of cycles/sec**

Output voltage = \( \frac{\text{on\_time}}{\text{cycle\_time}} \times 5\text{V} \)

Image credit: Tod Kurt

PROTOTYPING BOARDS AND LANGUAGES

L. BONONI, M. DI FELICE, L. SCIULLO, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF BOLOGNA, ITALY
The Arduino Board

<table>
<thead>
<tr>
<th></th>
<th>Arduino Uno</th>
<th>Arduino Mega 2560</th>
<th>Arduino Due</th>
<th>Arduino Mini</th>
<th>Lily Pad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processor</strong></td>
<td>ATmega328</td>
<td>ATmega2560</td>
<td>AT91SAM3X8E</td>
<td>ATmega328</td>
<td>ATmega168V ATmega328V</td>
</tr>
<tr>
<td><strong>CPU speed [MHz]</strong></td>
<td>16</td>
<td>16</td>
<td>84</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td><strong>Analog IN/OUT</strong></td>
<td>6/0</td>
<td>16/0</td>
<td>12/2</td>
<td>8/0</td>
<td>6/0</td>
</tr>
<tr>
<td><strong>Digital IO/PWM</strong></td>
<td>14/6</td>
<td>54/15</td>
<td>54/12</td>
<td>14/6</td>
<td>14/6</td>
</tr>
<tr>
<td><strong>SRAM [KB]</strong></td>
<td>2</td>
<td>8</td>
<td>96</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td><strong>Flash [KB]</strong></td>
<td>32</td>
<td>256</td>
<td>512</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td><strong>UART</strong></td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Price</strong></td>
<td>~20€</td>
<td>~22€</td>
<td>~50€</td>
<td>~15€</td>
<td>~45€</td>
</tr>
</tbody>
</table>

PROTOTYPING BOARDS AND LANGUAGES
L. BONONI, M. DI FELICE, L. SCIULLO, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, UNIVERSITY OF BOLOGNA, ITALY
The Arduino Board

- **Sketches:**
  - `.ino` files, `.cpp` and `.h`
  - `setup()` → initialization of the initial values (pins, variables, etc.)
  - `loop()` → infinite repetition of the main code

- **Development cycle:**

  ![Diagram of the development cycle: Edit → Compile → Upload → Run]
The Arduino Board

```c
// The setup function runs once when you press reset or power the board
void setup() {
  ledPinMode(LEDBUILTIN, OUTPUT);  // set LED pin as output
}

// The loop function runs over and over again forever
void loop() {
  digitalWrite(LEDBUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                   // wait for a second
  digitalWrite(LEDBUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000);                   // wait for a second
}
```
#define LED_PIN 13

void setup() {
    // initialize digital pin LED.
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_PIN, HIGH);    // turn the LED on (HIGH voltage level)
    delay(1000);                    // wait for a second
    digitalWrite(LED_PIN, LOW);     // turn the LED off (voltage LOW)
    delay(1000);                    // wait for a second
}
The Arduino Board

- **boolean** (8 bit) - simple logical true/false.
- **byte** (8 bit) - unsigned number from 0-255.
- **char** (8 bit) - signed number from -128 to 127.
- **unsigned char** (8 bit) - same as ‘byte’; if this is what you’re after you should use ‘byte’ instead.
- **word** (16 bit) - unsigned number from 0-65535.
- **unsigned int** (16 bit) - the same as ‘word’. Use ‘word’ for clarity and brevity.
- **int** (16 bit) - signed number from -32768 to 32767.
- **unsigned long** (32 bit) - unsigned number from 0-4,294,967,295.
- **long** (32 bit) - signed number from -2,147,483,648 to 2,147,483,647.
- **float** (32 bit) - signed number from -3.4028235-E38 to 3.4028235-E38. Floating point is not native; the compiler has to jump through hoops to make it work. If you can avoid it, you should.
## The Arduino Board

### Digital I/O
- `digitalRead()`
- `digitalWrite()`
- `pinMode()`

### Analog I/O
- `analogRead()`
- `analogReference()`
- `analogWrite()`

### Communication
- Serial
- stream

### Time
- `delay()`
- `delayMicroseconds()`
- `micros()`
- `millis()`
The Arduino Board

- Audio
- Cloud
- Communications
- Robotics
- Sensors
- Storage and memory
- Etc..

EXAMPLE of AVAILABLE LIBRARIES

https://playground.arduino.cc/Main/LibraryList
Firmata

- Protocol for communication between software on a host (computer, smartphone, etc..) and microcontrollers
- Firmata is based on the MIDI message format in that commands bytes are 8 bits and data bytes are 7 bits

- Firmata libraries:
  - Python
  - Perl
  - Javascript
  - Java
  - ruby
### Firmata Message Type

<table>
<thead>
<tr>
<th>type</th>
<th>command</th>
<th>MIDI channel</th>
<th>first byte</th>
<th>second byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>analog I/O message</td>
<td>0xE0</td>
<td>pin #</td>
<td>LSB(bits 0-6)</td>
<td>MSB(bits 7-13)</td>
</tr>
<tr>
<td>digital I/O message</td>
<td>0x90</td>
<td>port</td>
<td>LSB(bits 0-6)</td>
<td>MSB(bits 7-13)</td>
</tr>
<tr>
<td>report analog pin</td>
<td>0xC0</td>
<td>pin #</td>
<td>disable/enable(0/1)</td>
<td>- n/a -</td>
</tr>
<tr>
<td>report digital port</td>
<td>0xD0</td>
<td>port</td>
<td>disable/enable(0/1)</td>
<td>- n/a -</td>
</tr>
<tr>
<td>start sysex</td>
<td>0xF0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>set pin mode(I/O)</td>
<td>0xF4</td>
<td>pin # (0-127)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>set digital pin value</td>
<td>0xF5</td>
<td>pin # (0-127)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sysex end</td>
<td>0xF7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protocol version</td>
<td>0xF9</td>
<td>major version</td>
<td>minor version</td>
<td></td>
</tr>
<tr>
<td>system reset</td>
<td>0xFF</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Firmata Message Type

- Firmata library written in Javascript

- Platform Support:
  - Arduino
  - Raspberry Pi
  - Intel platforms
  - Etc...

- APIs for using the following sensors/actuators:
  - Accelerometer, GPS, Joystick, Keypad, LCD, Led, Leds, Motor, Pin, Thermometer, etc..
EXAMPLE OF SINGLE-BOARD COMPUTER: RASPBERRY PI
The Raspberry Pi Board

- Developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries.

- “The Raspberry Pi Foundation is a UK-based charity that works to put the power of digital making into the hands of people all over the world, so they are capable of understanding and shaping our increasingly digital world, able to solve the problems that matter to them, and equipped for the jobs of the future.”

https://www.raspberrypi.org/about/
The Raspberry PI Board

- Foundation trustee Eben Upton assembled a group of teachers, academics and computer enthusiasts to devise a computer to inspire children.

- The computer is inspired by **Acorn's BBC Micro of 1981**. The Model A, Model B and Model B+ names are references to the original models of the British educational BBC Micro computer, developed by Acorn Computers.
The Raspberry PI Board

- Single-board computer with a Broadcom system on a chip (SoC) with an integrated ARM compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).
The Raspberry PI Board

- **GPIO**
  - 40 digital pins
  - PWM by software
  - UART
  - I2C
  - SPI
<table>
<thead>
<tr>
<th></th>
<th>PiA</th>
<th>PiB</th>
<th>Pi2B</th>
<th>Pi3B</th>
<th>PiZero</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU clock</strong></td>
<td>700 MHz</td>
<td>700 MHz</td>
<td>900 MHz</td>
<td>1.2 GHz</td>
<td>1 GHz</td>
</tr>
<tr>
<td><strong>Cores</strong></td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td><strong>Instruction set</strong></td>
<td>ARMv6</td>
<td>ARMv6</td>
<td>ARMv7-A</td>
<td>ARMv8-A</td>
<td>ARMv6</td>
</tr>
<tr>
<td><strong>SoC</strong></td>
<td>Broadcom</td>
<td>Broadcom</td>
<td>Broadcom</td>
<td>Broadcom</td>
<td>Broadcom</td>
</tr>
<tr>
<td></td>
<td>BCM2835</td>
<td>BCM2835</td>
<td>BCM2836</td>
<td>BCM2837</td>
<td>BCM2835</td>
</tr>
<tr>
<td><strong>USB ports</strong></td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>1 Micro</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>256 MB</td>
<td>512 MB</td>
<td>1 GB</td>
<td>1 GB</td>
<td>512 MB</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>SDCard</td>
<td>SDCard</td>
<td>Micro SD card</td>
<td>Micro SD card</td>
<td>Micro SD card</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>None</td>
<td>100 Mbit/s</td>
<td>100 Mbit/s</td>
<td>100 Mbit/s</td>
<td>None</td>
</tr>
<tr>
<td><strong>Wifi</strong></td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>802.11n</td>
<td>None</td>
</tr>
<tr>
<td><strong>Bluetooth</strong></td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>4.1</td>
<td>None</td>
</tr>
<tr>
<td><strong>GPIO</strong></td>
<td>26 pins</td>
<td>26 pins</td>
<td>40 pins</td>
<td>40 pins</td>
<td>40 pins</td>
</tr>
<tr>
<td><strong>Alimentation</strong></td>
<td>5V/1 Amp</td>
<td>5V/1 Amp</td>
<td>5V/2 Amp</td>
<td>5V/2.5 Amp</td>
<td>5V/2 Amp</td>
</tr>
<tr>
<td><strong>cost</strong></td>
<td>20$</td>
<td>35$</td>
<td>35$</td>
<td>35$</td>
<td>5$</td>
</tr>
</tbody>
</table>
The Raspberry PI Board

Software Distro

- **Linux Distros:**
  - Raspbian (Debian) and Ubuntu (at least ARMv7)
  - Archlinux, Fedora, Gentoo, etc..
  - Volumio, OpenElec
  - RISC OS

- **Others:**
  - Windows 10 IoT Core Edition (at least ARMv7)
  - Android
The Raspberry PI Board

- **Raspberry vs Arduino** Boards

**Advantages of Arduino:**
- Robustness (no OS on it)
- Low Power Consumption
- Price

**Advantages of Raspberry:**
- Powerfulness (multitasking)
- Networking on board
- All the OS functionalities
The Raspberry PI Board

Raspberry vs Arduino Boards

Choose Arduino if:
- You want to learn basics of electronics
- You have a simple project to deploy
- Your projects does not involve so much software

Choose Raspberry if:
- You have a complex project, with a lot of software involved
- You need OS functionalities
- Networking is involved
DEMO