Computational models for embedded systems

Assignment 3: **Embedded Hardware Building Blocks.**

Embedded Board

Associate Professor Andreea Vescan

Babeș-Bolyai University

Cluj-Napoca

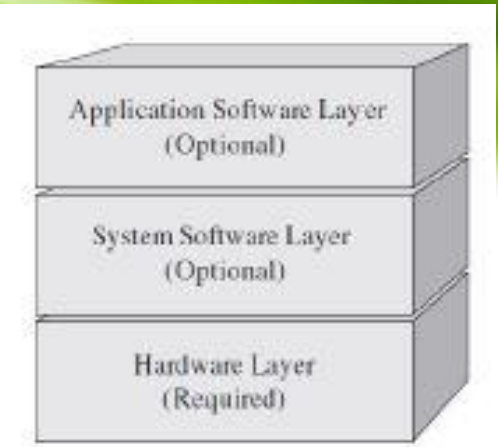
2025-2026

Outline

- Embedded Hardware Building Blocks and the Embedded Board
- CPUs vs. MCUs vs. Embedded Systems
- Electric Circuits
- Basic Hardware Materials

- Nucleo STM32F401 – board
- Nucleo projects

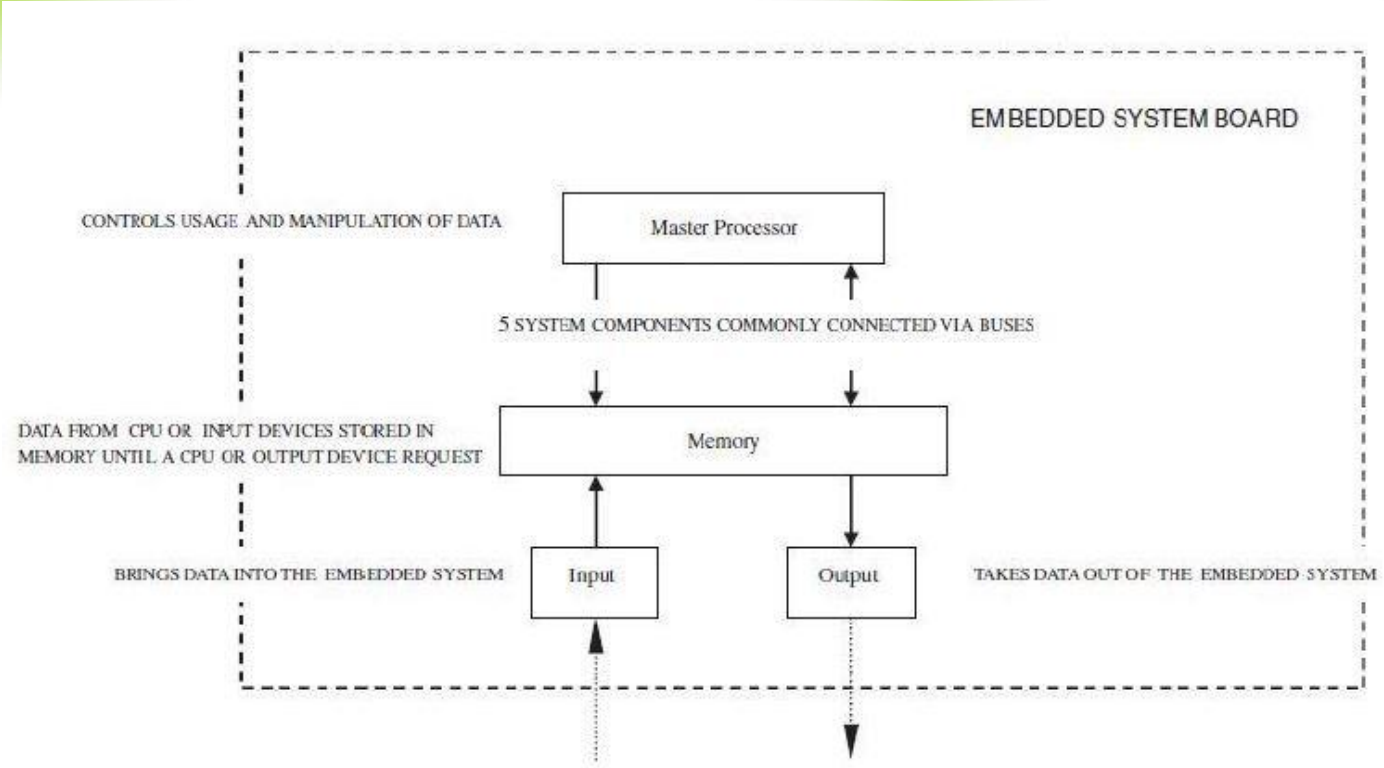
Embedded Hardware Building Blocks



- Embedded System Model
  [Noergaard2005]

- PCB – Printed Circuit Board or PW – Printed Wiring board
  - Thin sheets of fiberglass
  - Electrical part is printed in cooper – electrical signals is carried between the connected components
- Major hardware components of most boards
  - Central Process Unit
  - Memory
  - Input Device (s)
  - Output Device (s)
  - Data Pathway(s)/Bus(es)

Embedded Hardware Building Blocks



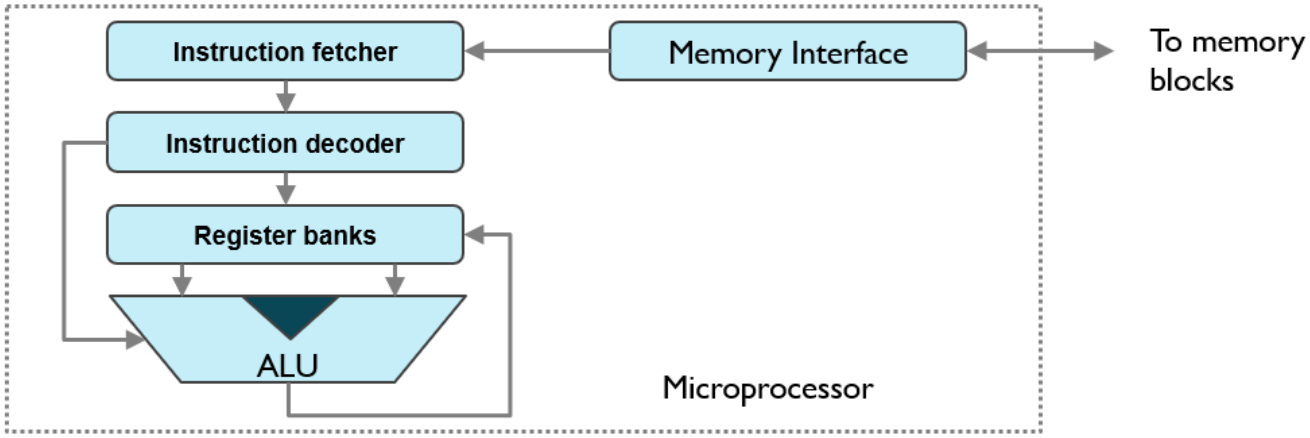
- Embedded System board organization [Noergaard2005]

Embedded Microcomputer System

- Embedded microcomputer system
  - Embedded = hidden inside so one can’t see it.
  - Micro = small.
  - Computer = contains a processor, memory and a means to exchange data with the external world.
  - System = multiple components interfaced together for a common purpose; have structure, behavior and interconnectivity operating in a framework bound by rules and regulations.

CPUs vs. MCUs vs. Embedded Systems [ARM-RESD]

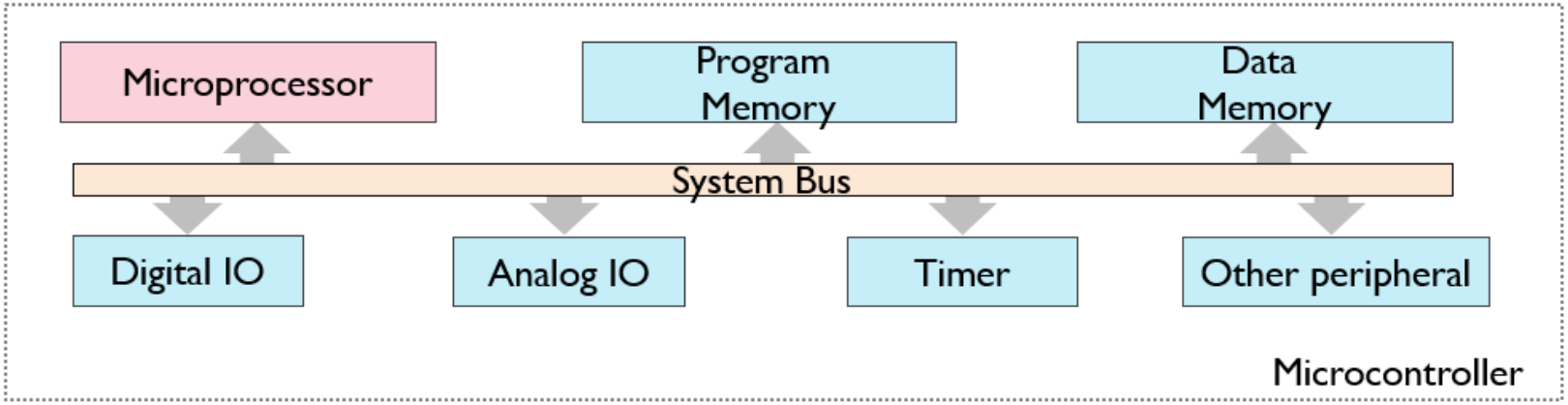
- Microprocessor (CPU)
  - Defined typically as a single processor core that supports at least instruction fetching, decoding, and executing
  - Normally can be used for general purpose computing, but needs to be supported with memories and Input/Outputs(IOs)



ARM

CPUs vs. MCUs vs. Embedded Systems [ARM-RESD]

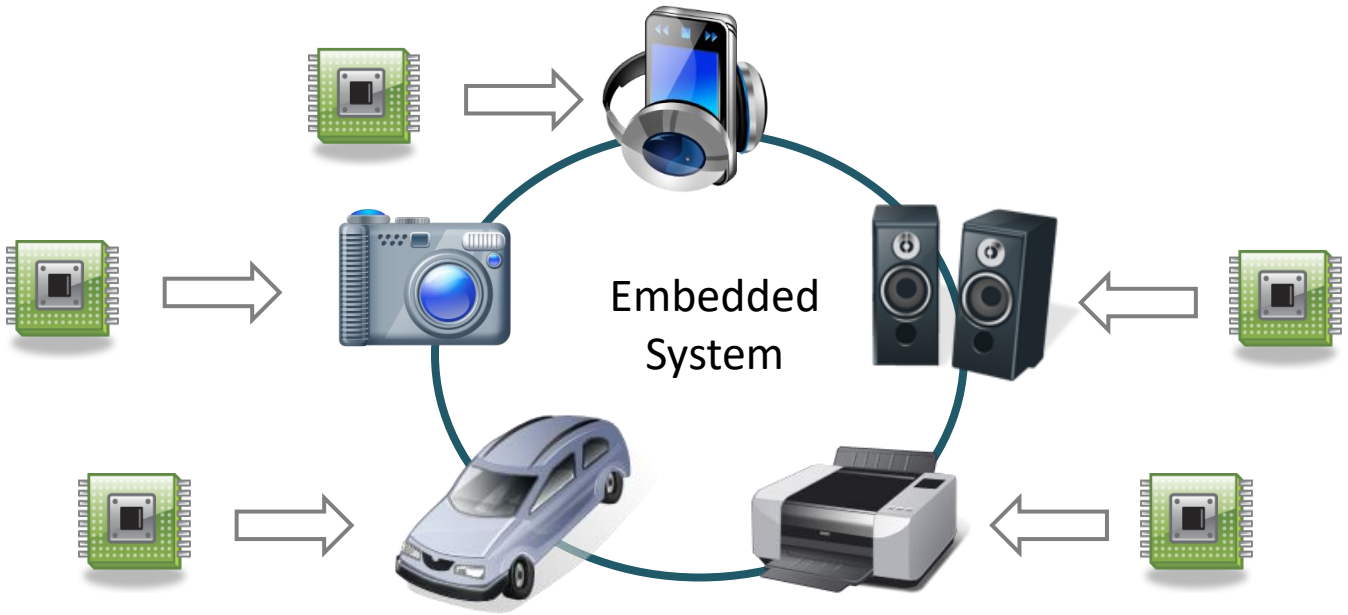
- Microcontroller (MCU)
  - Typically has a single processor core
  - Has memory blocks, Digital IOs, Analog IOs, and other basic peripherals
  - Typically used for basic control purpose, such as embedded applications



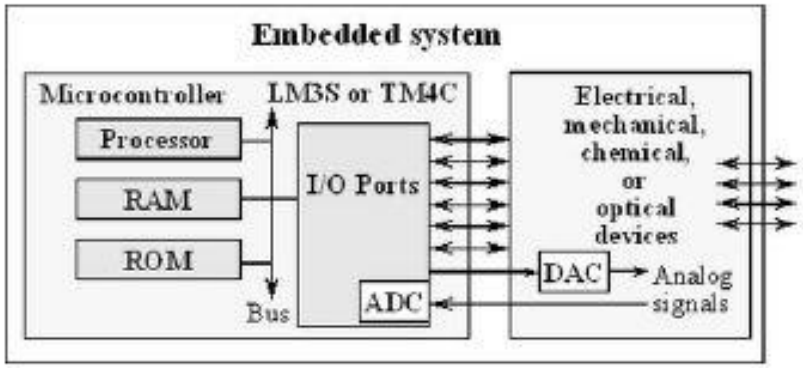
ARM

CPUs vs. MCUs vs. Embedded Systems [ARM-RESD]

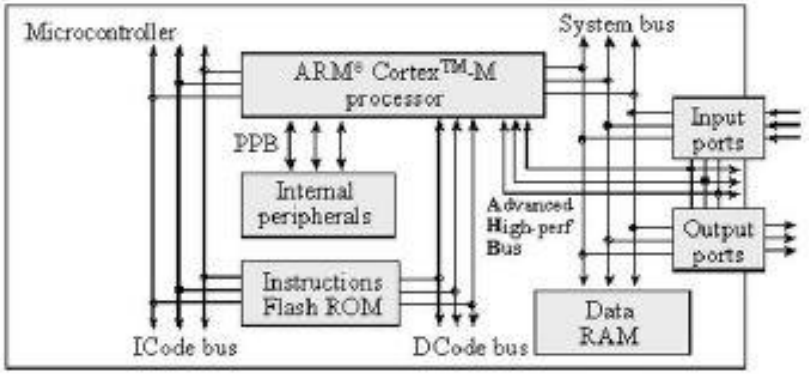
- Embedded System
  - Typically implemented using MCUs
  - Often integrated into a larger mechanical or electrical system
  - Usually has real-time constraints



Non Neumann vs. Harvard Architecture



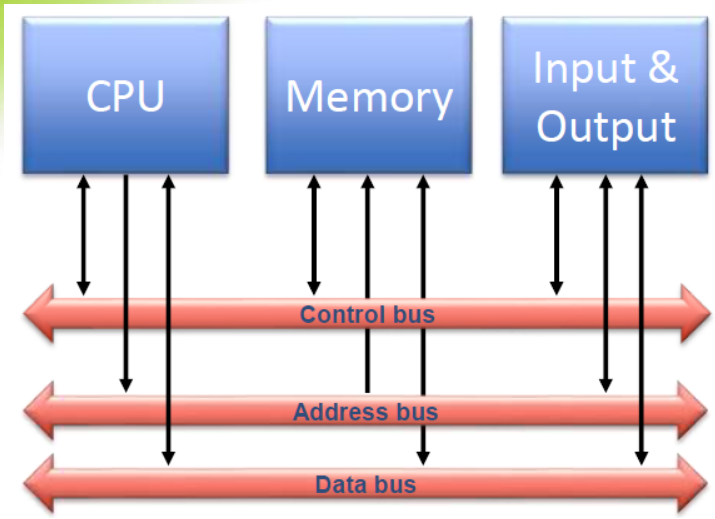
VON NEUMANN ARCHITECTURE

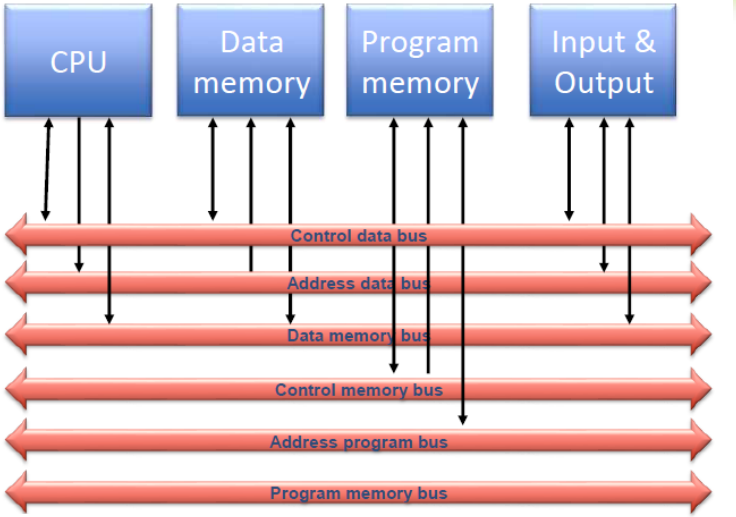


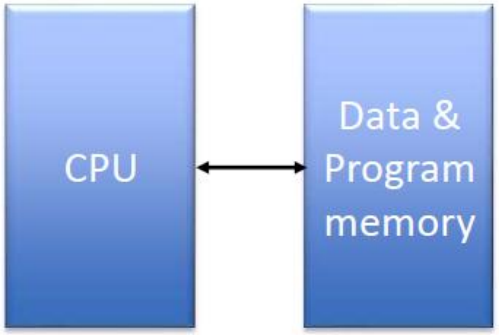
HARVARD ARCHITECTURE

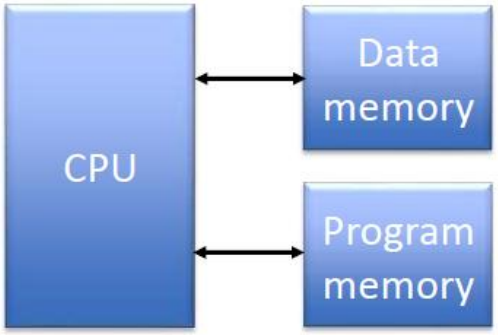
- Harvard architecture – has separate data and instruction buses.
  - Instructions are fetched from flash ROM using ICode bus.
  - Data are exchanged with memory and I/O via the System bus interface.

Non Neumann vs. Harvard Architecture







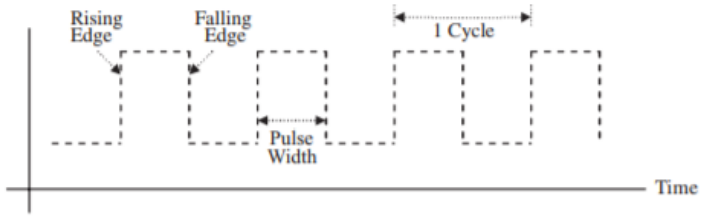


\+ Simple HW -

\- Execution time +

CPU and System (master) Clock [Noergaard2005]

- CPU: ALU (Arithmetic Logic Unit) +registers + CU (control unit) + internal CPU
- A processor's execution is ultimately synchronized by an external system or master clock, located on the board.
- The master clock is an oscillator along with a few other components, such as a crystal. It produces a fixed frequency sequence of regular on/off pulse signals (square waves) buses.



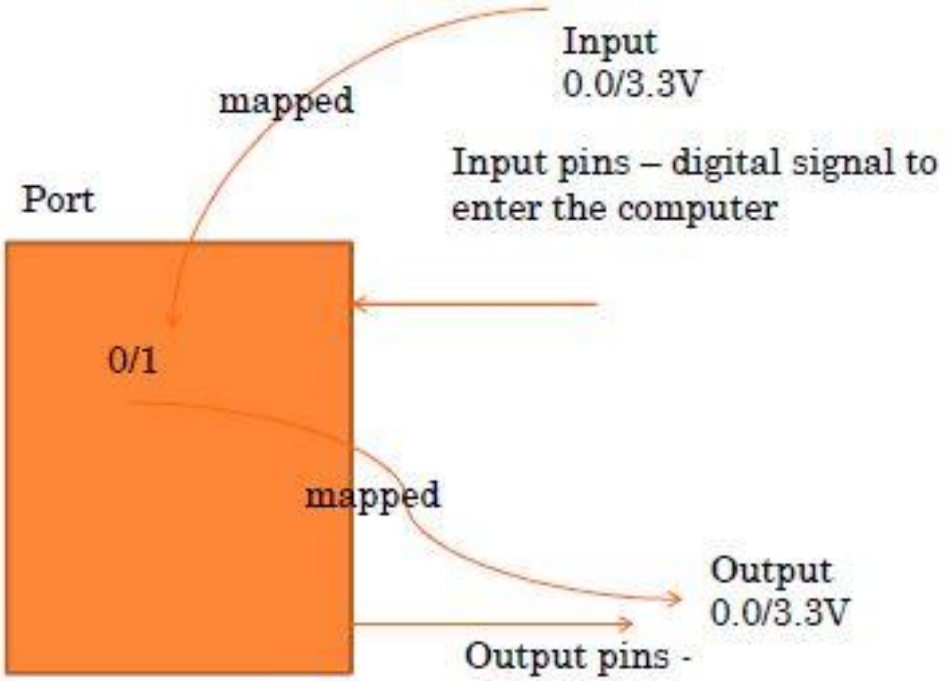
- The CU, along with several other components on an embedded board, depends on this master clock to function.
- Components are driven by either the actual level of the signal (a "0" or a "1"), the rising edge of a signal (the transition from "0" to "1"), and/or the falling edge of the signal (the transition from "1" to "0")

I/O Ports or I/O Devices or Interfaces

- **I/O Ports**
  - Input port – is hardware on the microcontroller that allow information about the external world to be entered into the computer.
  - Output port – is hardware on the microcontroller that sends information out to the external world.
- **Interface** – the collection of the I/O port, external electronics, physical devices, and the software, which combine to allow the computer to communicate with the external world.
  - Example
    - input interface = switch - the operator toggles the switch and the software can recognize the switch position.
    - output interface = light-emitting diode (LED) – the software can turn the light on and off and the operator can see whether or not the light is shining.
- **Inputs/outputs** – digital or analog.
- **Classification** of I/O interfaces
  - **Parallel** = binary data are available simultaneously on a group of line;
  - **Serial** = binary data are available one bit at a time on a single line;
  - **Analog** = data are encoded as an electrical voltage, current, or power;
  - **Time** = data are encoded as a period, frequency, pulse width, or phase shift.

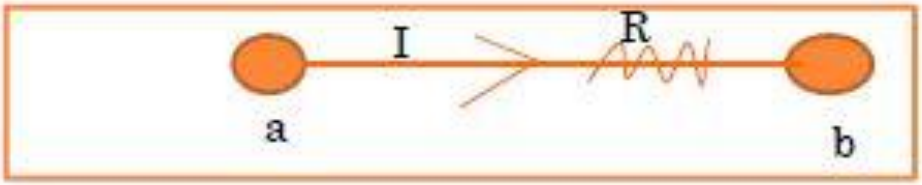
Microcontroller Ports

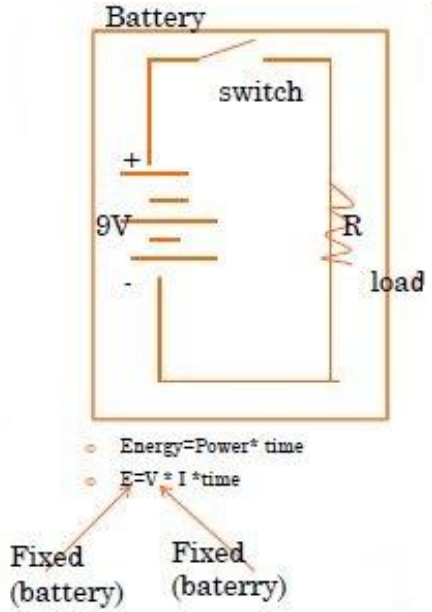
- Parallel
  - How it works?
  - Parallel = binary data are available simultaneously on a group of line.



Electric Circuits

- Current -I – flow of charge -> Amperes
- Voltage – V – potential difference→ Joules/coulomb=Volts
- Amount of energy needed to move 1 unit of charge (=1coulomb) from a to b





- Ohms law – give a relationship between V and I

  V=I*R

- The current flow through a conductor (that has a resistor)
- Power = V * I(watts)=Joules/sec

(the power dissipated through this resistance)

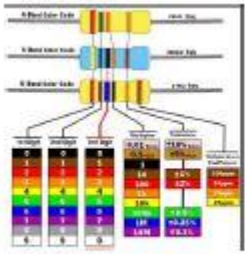
Powering the hardware: Some embedded boards plug into power supplies. (3.3V or 5V or 12V)

Basic Hardware Materials

- Categories – the ability of the materials to conduct electric current.
- **Conductors**
  - materials that have fewer impediments to an electric current.
  - have 3 or fewer valence electrons.
- **Insulators** (non-metals, air, paper, oil, plastic, glass, rubber)
  - impede an electric current.
  - have 5 or more valence electrons.
- **Semiconductors**
  - have 4 valence electrons
  - Materials whose base elements have a conductive nature that can be altered by introducing other elements into their structure.
  - have the ability to behave both as a conductor and as an insulator.
  - Impurities
  - Donor impurities → create a surplus of electrons ←→N-type conductor
  - Acceptors impurities → produce a shortage of electrons ←→P-type conductor

Passive Components on Boards (and in Chips)
Resistors, Capacitors, Inductors

- **Resistors**
  - devices made up of conductive materials that have their conductivity altered in some fashion to allow for an increase in resistance.
  - provide the inherent function, to create a resistive force in a circuit.
  - are a means (Alternating Current or Direct Current circuit) to control the current or voltage by providing some amount of resistance to the current or voltage that flows across them.
- Type
  - **Fixed resistors** – are manufactured to have only one resistance value.
  - **Variable resistors** –vary their resistance on-the-fly.
    - manually –potentiometers;
    - by changes in light –photosensitive resistor;
    - by changes in temperature – thermally sensitive/thermistor.

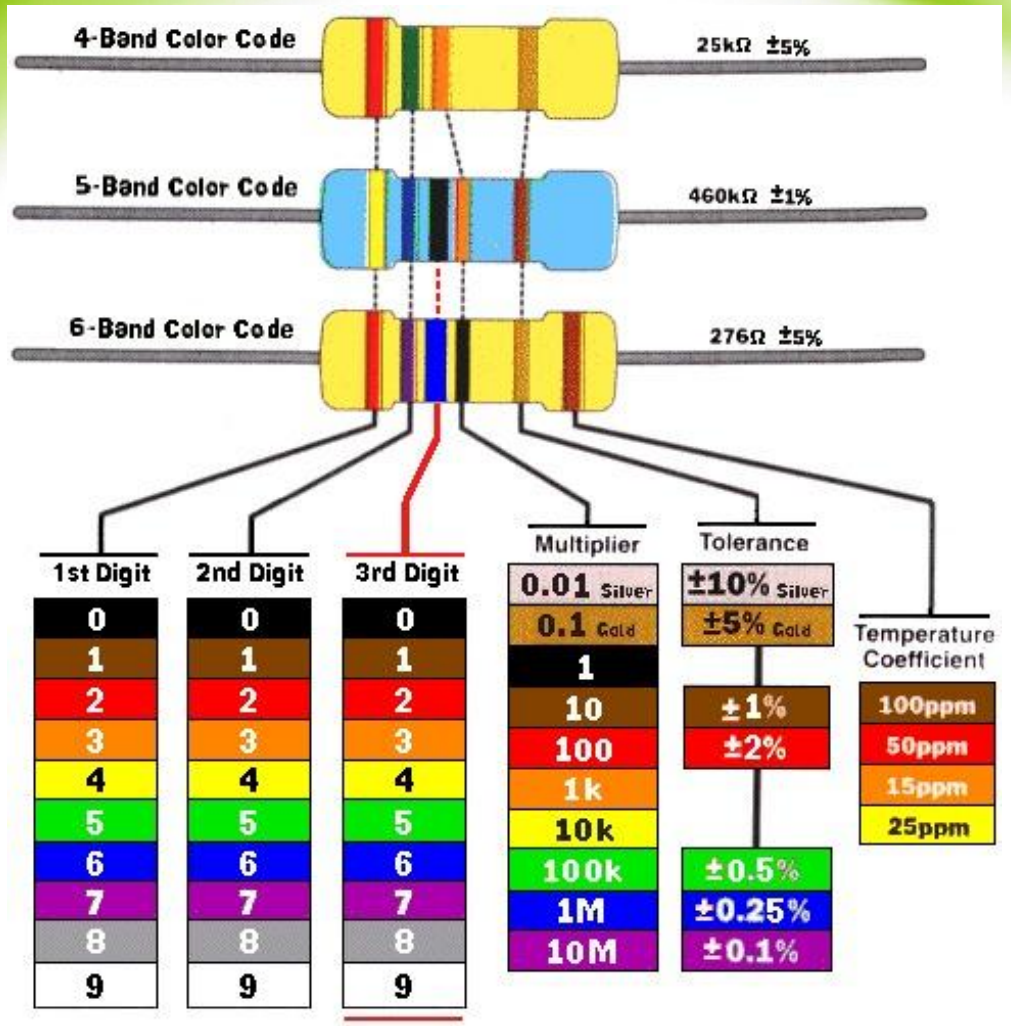


Passive Components on Boards (and in Chips)

- **Capacitors**
  - store energy in *electric* fields.
- impede the flow of energy (commonly used in AC circuits) and gives this same energy back to the circuit in its original form (electrically).
- Type
  - Variable, ceramic, electrolytic, etc.
  - Adjusted on-the-fly or not

- **Inductors**
  - store energy in *magnetic* fields.

Resistor – color guide



- The reading direction might not always be clear. Sometimes the increased space between band 3 and 4 give away the reading direction. Also, the first band is usually the closest to a lead. A gold or silver band (the tolerance) is always the last band.

- https://www.allaboutcircuits.com/tools/resistor-color-code-calculator/

What is ARM Architecture [ARM-RESD]

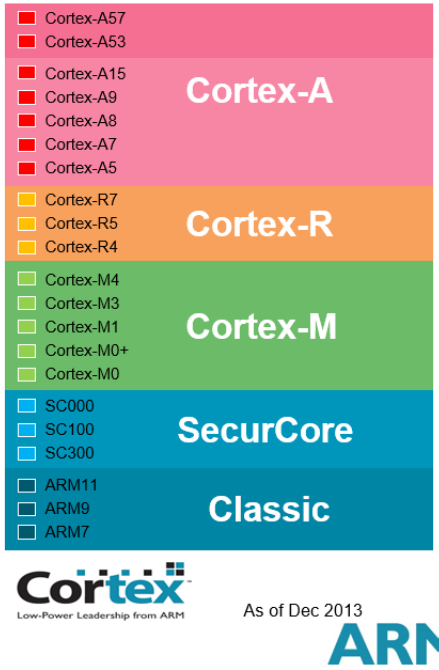
- ARM architecture is a family of RISC-based processor architectures

  RISC = Reduced Instruction Set Computer

  - Well-known for its power efficiency;
  - Hence widely used in mobile devices, such as smartphones and tablets
  - Designed and licensed to a wide eco-system by ARM
- ARM Holdings
  - The company designs ARM-based processors;
  - Does not manufacture, but licenses designs to semiconductor partners who add their own Intellectual Property (IP) on top of ARM's IP, fabricate and sell to customers;
  - Also offer other IP apart from processors, such as physical IPs, interconnect IPs, graphics cores, and development tools.

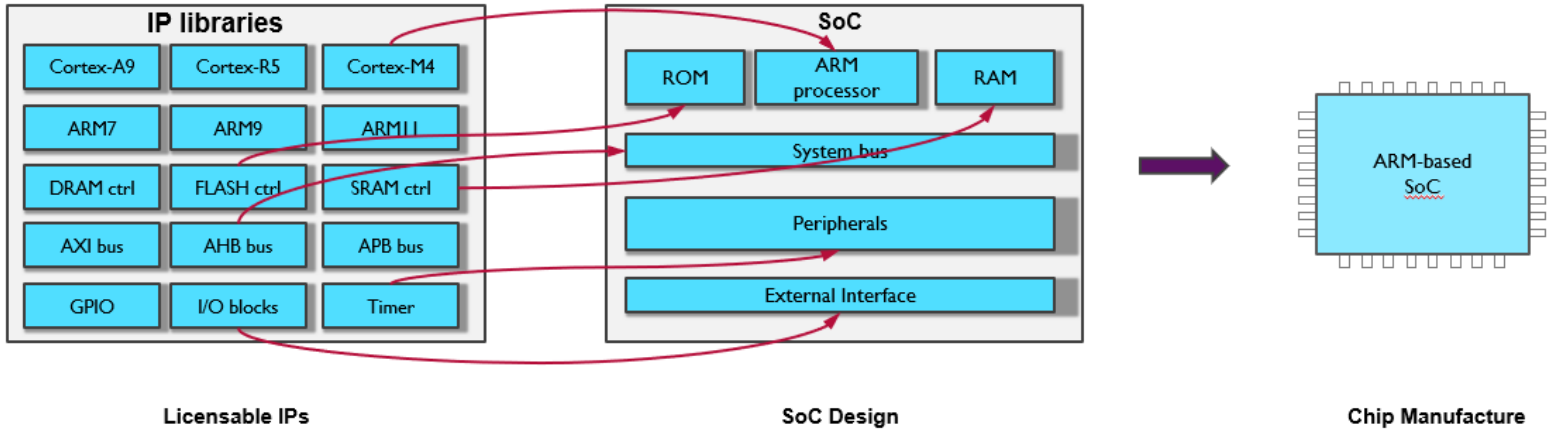
ARM Processor Families [ARM-RESD]

- Cortex-A series (Application)
  - High performance processors capable of full Operating System (OS) support;
  - Applications include smartphones, digital TV, smart books, home gateways etc.
- Cortex-R series (Real-time)
  - High performance for real-time applications;
  - High reliability
  - Applications include automotive braking system, powertrains etc.
- Cortex-M series (Microcontroller)
  - Cost-sensitive solutions for deterministic microcontroller applications;
  - Applications include microcontrollers, mixed signal devices, smart sensors, automotive body electronics and airbags;
- SecurCore series
  - High security applications.
- Previous classic processors
  - Include ARM7, ARM9, ARM11 families



Design an ARM-based SoC [ARM-RESD]

- Select a set of IP cores from ARM and/or other third-party IP vendors
- Integrate IP cores into a single chip design
- Give design to semiconductor foundries for chip fabrication



ARM

ARM Cortex-M Series

- Cortex-M series: Cortex-M0, M0+, M1, M3, M4.
- Energy-efficiency
  - Lower energy cost, longer battery life
- Smaller code
  - Lower silicon costs
- Ease of use
  - Faster software development and reuse
- Embedded applications
  - Smart metering, human interface devices, automotive and industrial control systems, white goods, consumer products and medical instrumentation

ARM Processors vs. ARM Architectures

- ARM architecture
  - Describes the details of instruction set, programmer's model, exception model, and memory map
  - Documented in the Architecture Reference Manual
- ARM processor
  - Developed using one of the ARM architectures
  - More implementation details, such as timing information
  - Documented in processor's Technical Reference Manual

NUCLEO-F401RE board

- **Core Architecture:** ARM
- **Core Sub-Architecture:** Cortex-M4

Installation instructions

- https://os.mbed.com/studio/

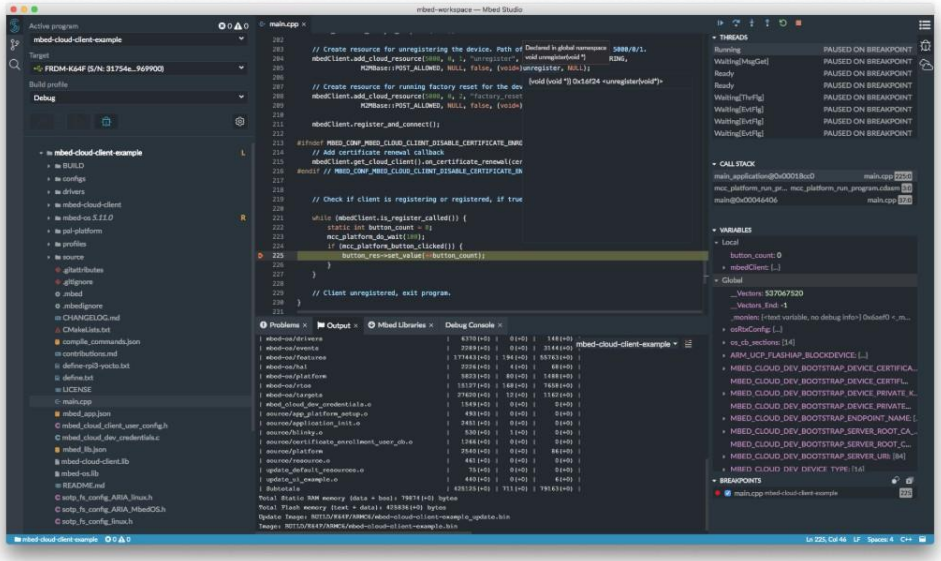
Mbed Studio

The desktop IDE for Mbed

Mbed Studio is a free IDE for Mbed OS application and library development, including all the dependencies and tools you need in a single package so that you can create, compile and debug your Mbed programs on the desktop.

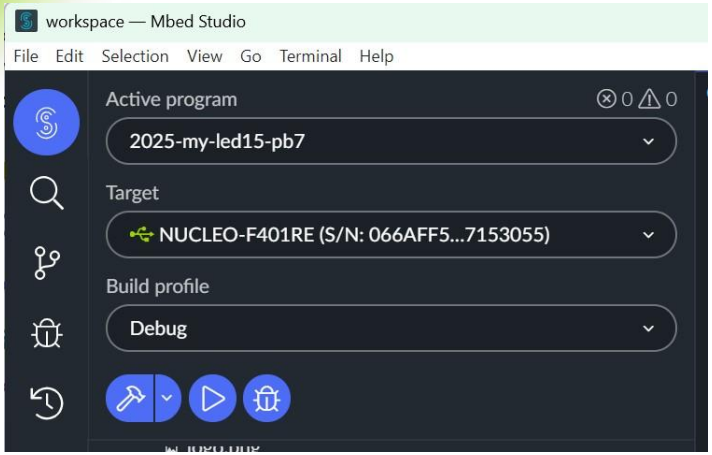
Download for Windows | Download for Mac | Download for Linux

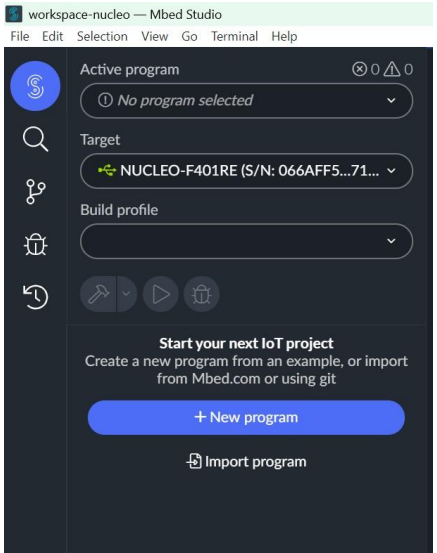
Documentation »

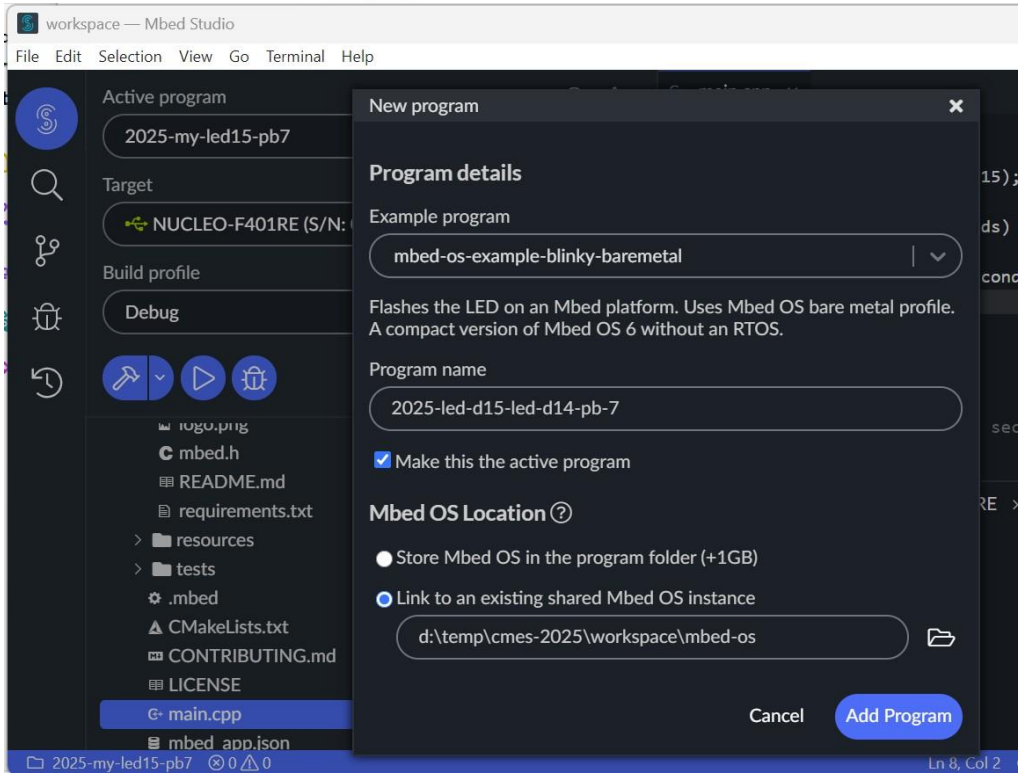


Installation instructions

- https://os.mbed.com/studio/

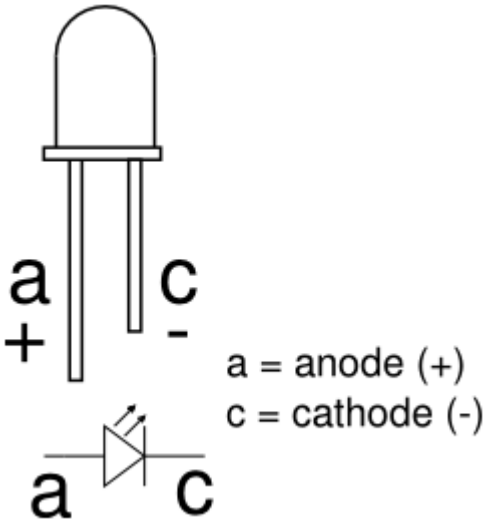


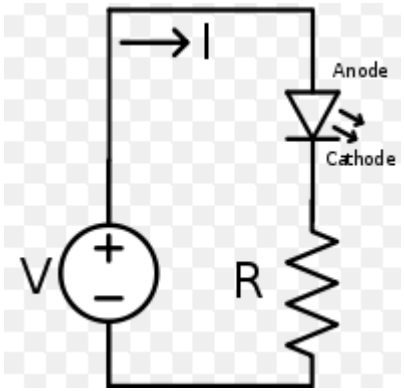




LED

- LED – light emitting diode
- https://www.evilmadscientist.com/2012/resistors-for-leds/





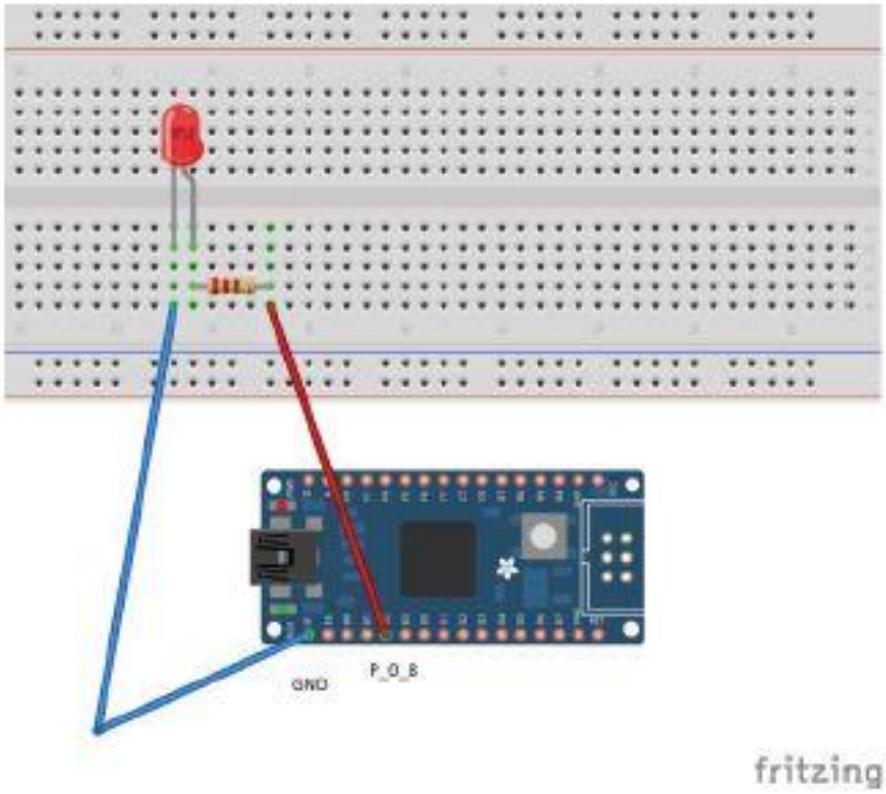
NUCLEO-F401RE projects

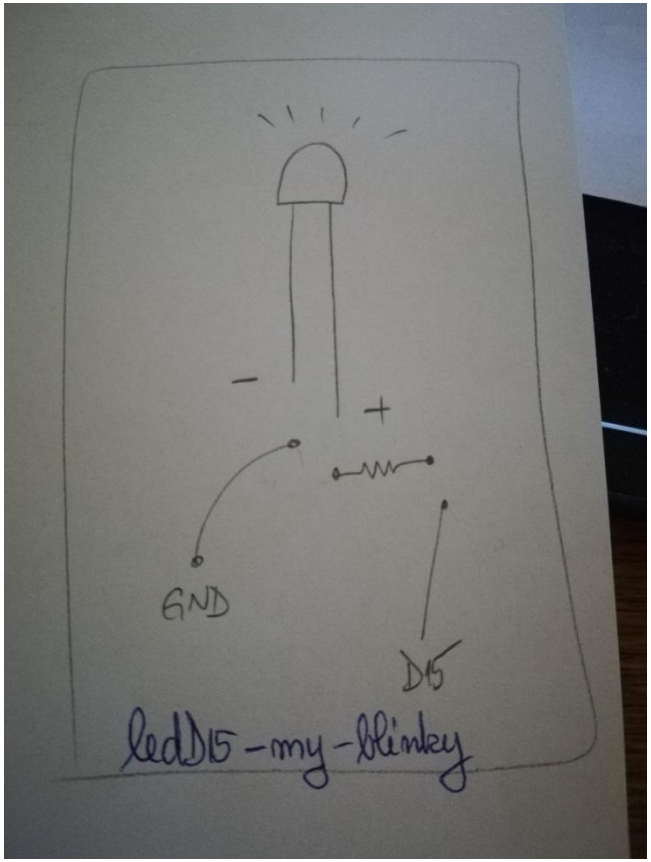
my_blinky_ledD15

my_blinky_ledD15_ledD14_pbD7_uvision5_nucleo_f401re
my_blinky_ledD15_pbD7_uvision5_nucleo_f401re
my_blinky_ledD15_uvision5_nucleo_f401re

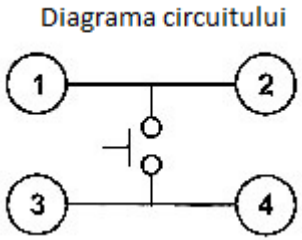
https://os.mbed.com/studio/

Electronic Circuit





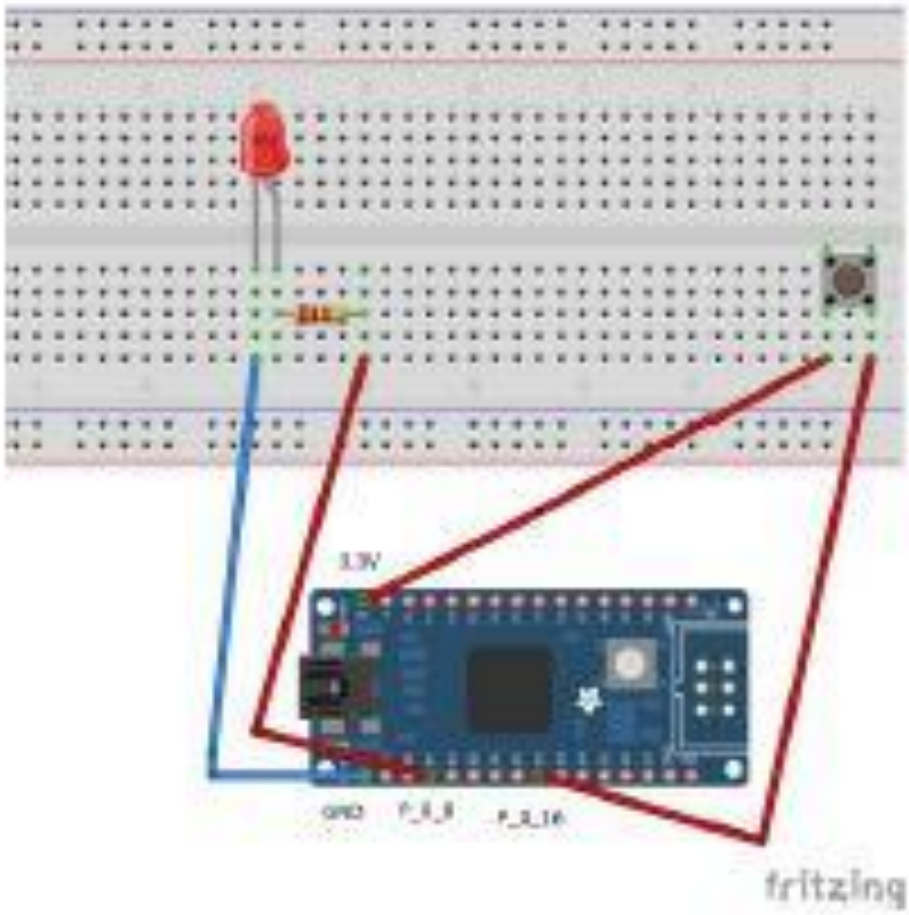
Push button

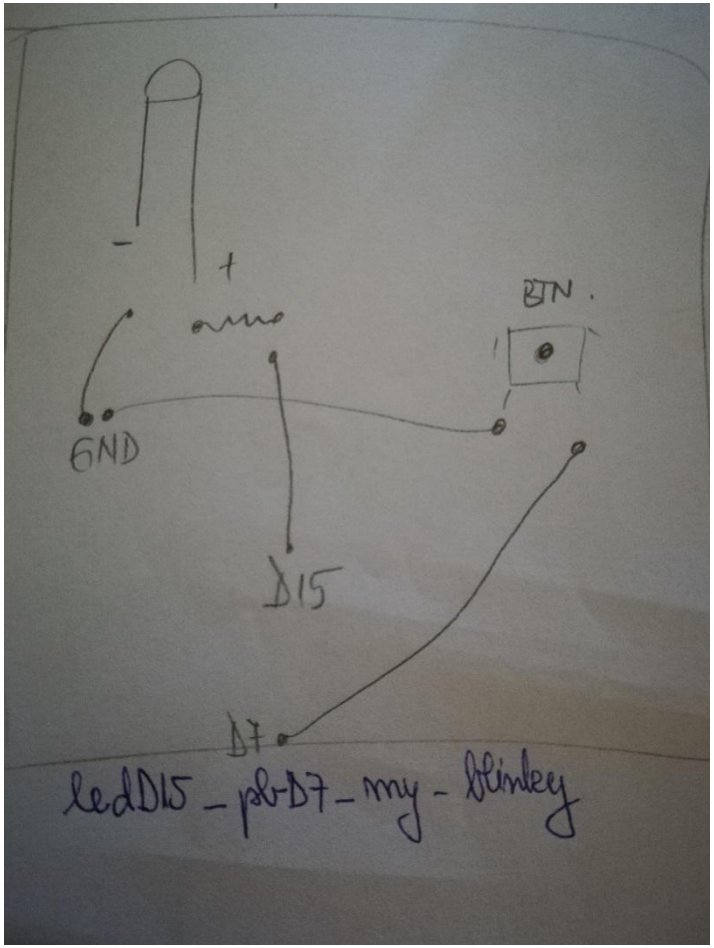


my_blinky_ledD15_ledD14_pbD7_uvision5_nucleo_f401re
my_blinky_ledD15_pbD7_uvision5_nucleo_f401re
my_blinky_ledD15_uvision5_nucleo_f401re

NUCLEO-F401RE projects
my_blinky_ledD15_pbD7

https://os.mbed.com/studio/



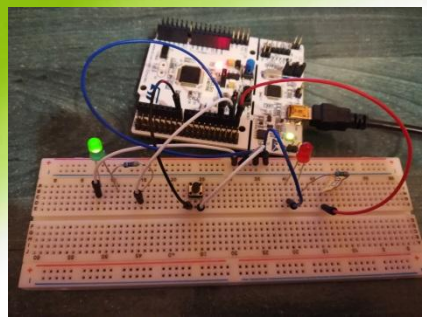


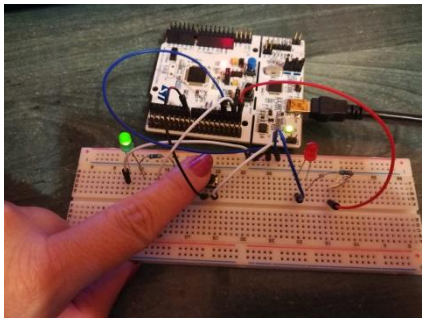
NUCLEO-F401RE =

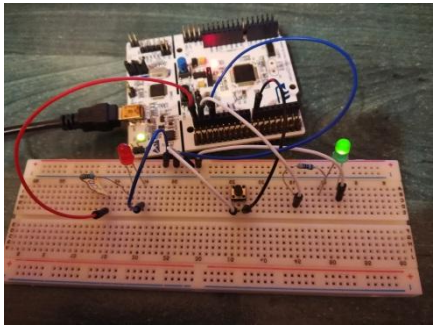
- my_blinky_ledD15_ledD14_pbD7_uvision5_nucleo_f401re
- my_blinky_ledD15_pbD7_uvision5_nucleo_f401re
- my_blinky_ledD15_uvision5_nucleo_f401re

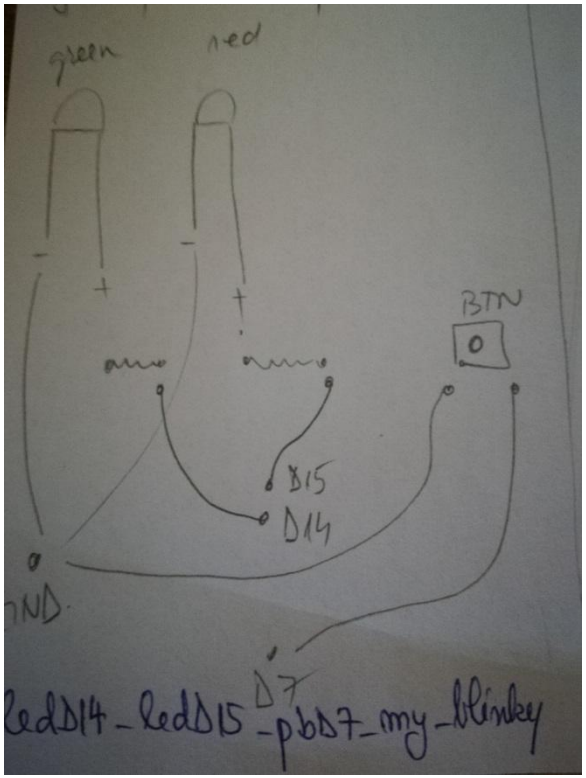
my_blinky_ledD15_ledD14_pbD7

https://os.mbed.com/studio/



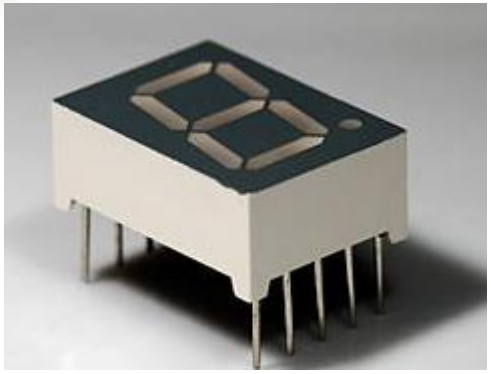


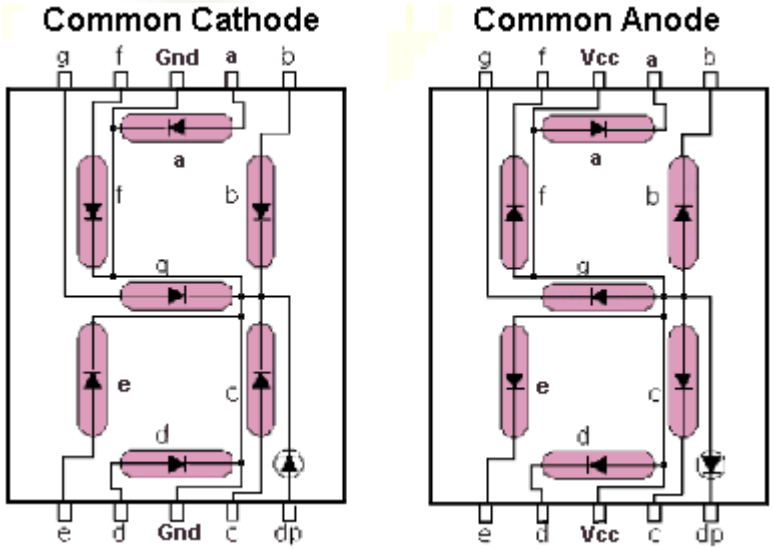


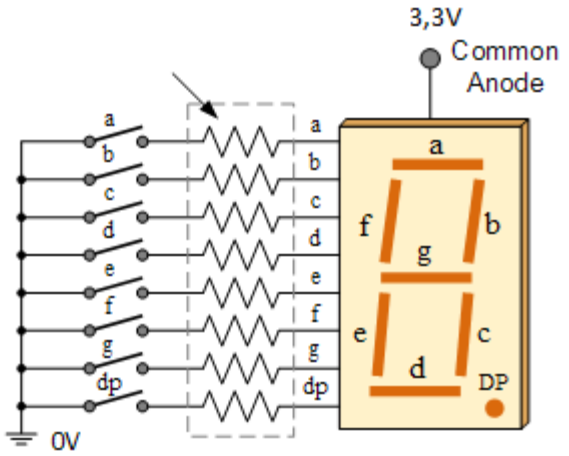


NUCLEO-F401RE projects

7 segments







- Write number “2”.
- Bonus: Activity in class: 25 XP

NUCLEO-F401RE

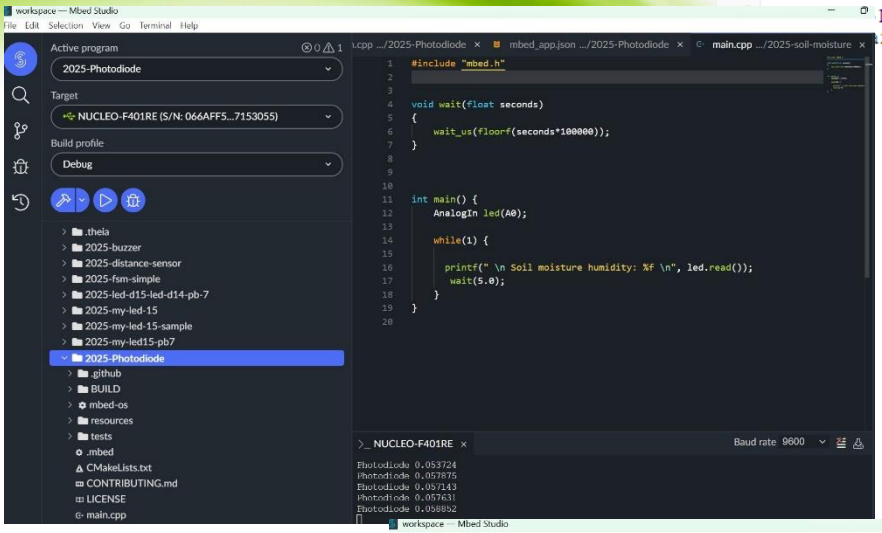
Sensors

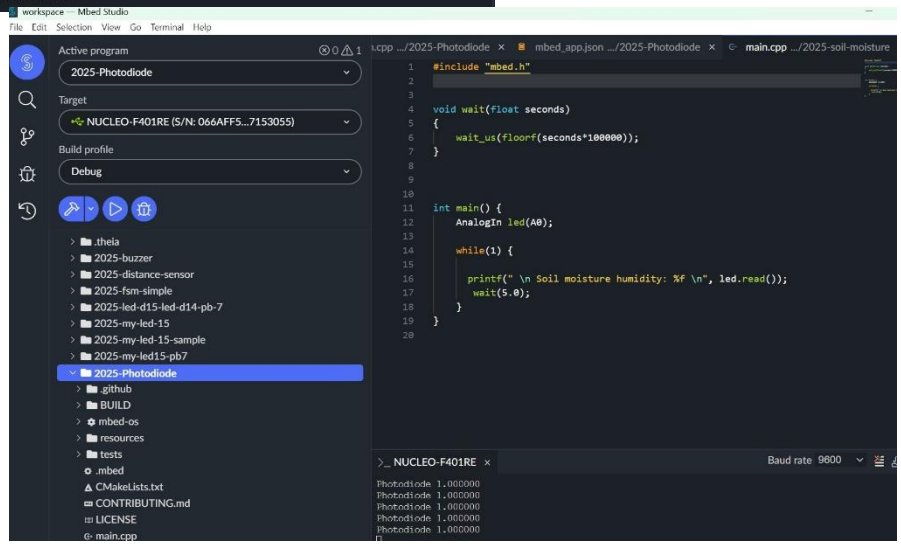
- Sensor_Photodiode_uvision5_nucleo_f401re
- Sensor_Buzzer_uvision5_nucleo_f401re
- Sensor_LCD_Text_uvision5_nucleo_f401re
- Sensor_Soil_Moisture_uvision5_nucleo_f401re
- Sensor_Distance_HCSR04_uvision5_nucleo_f401re
- Sensor_pir_motion_hc-sr501_uvision5_nucleo_f401re
- Sensor_Temperature_Humidity_Air_uvision5_nucleo_f401re

Photodiode

main.cpp

```
#include "mbed.h"
Serial pc(USBTX, USBRX);
int main() {
    AnalogIn led(A0);
    pc.baud(115200);

    while(1) {
        printf(" Photodiode: %f ", led.read(),"\n");
        wait(1.0);
```





- https://www.optimusdigital.ro/ro/senzori-senzori-optici/5116-modul-cu-fotodioda.html

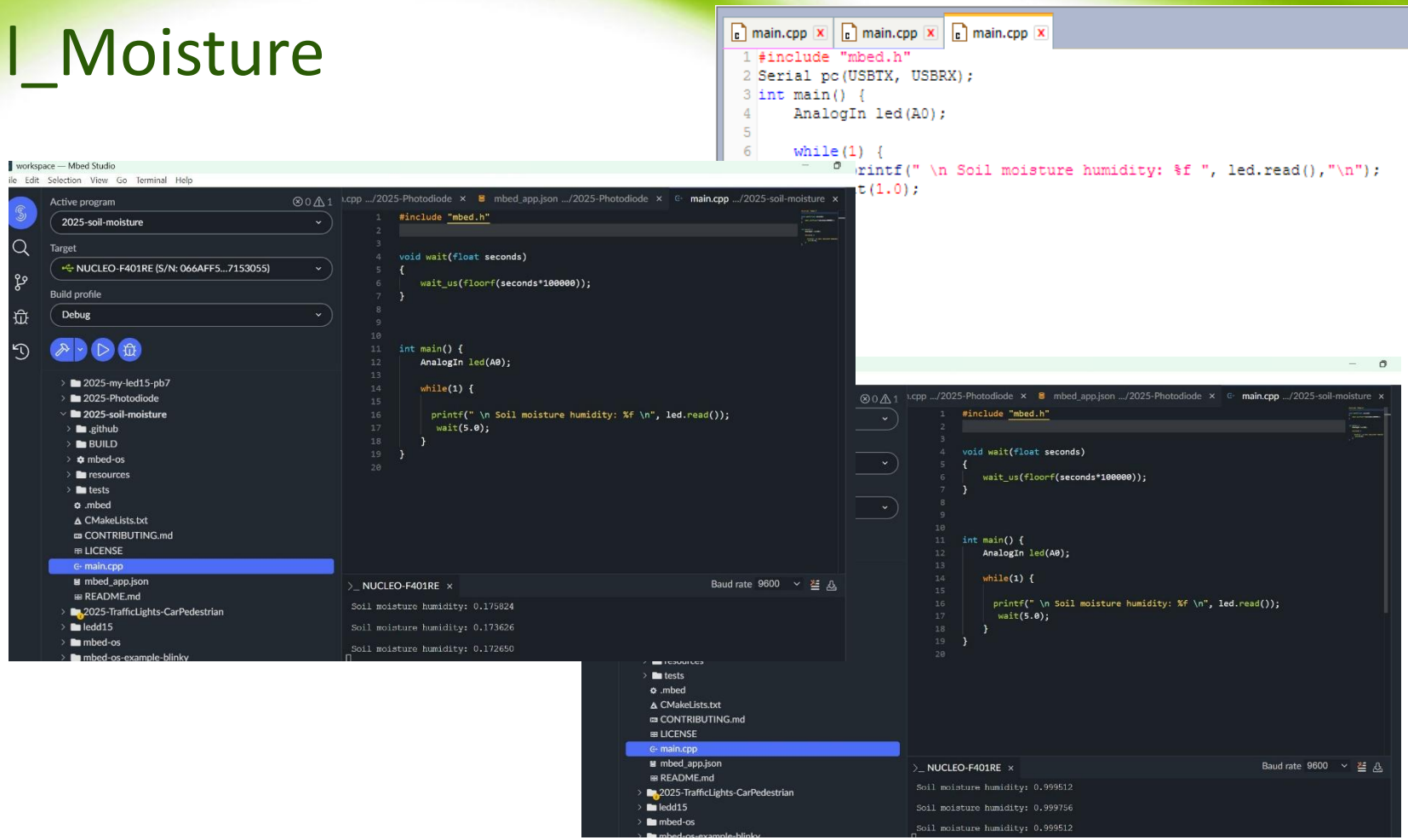
Buzzer

main.cpp main.cpp

```
1
2  /** class to make sound with a buzzer, based on a PwmOut
3   *   The class use a timeout to switch off the sound  - it is not blocking while making noise
4   *
5   * Example:
6   * @code
7   * // Beep with 1Khz for 0.5 seconds
8  */
9   #include "mbed.h"
10  #include "buzzer.h"
11
12  Beep buzzer(D3);//(PTC2);
13
14  int main() {
15
16    buzzer.beep(1000,0.5);
17    wait(2.0);
18    buzzer.beep(1000,0.5);
19    wait(2.0);
20    buzzer.beep(1000,0.5);
21  }
22
```

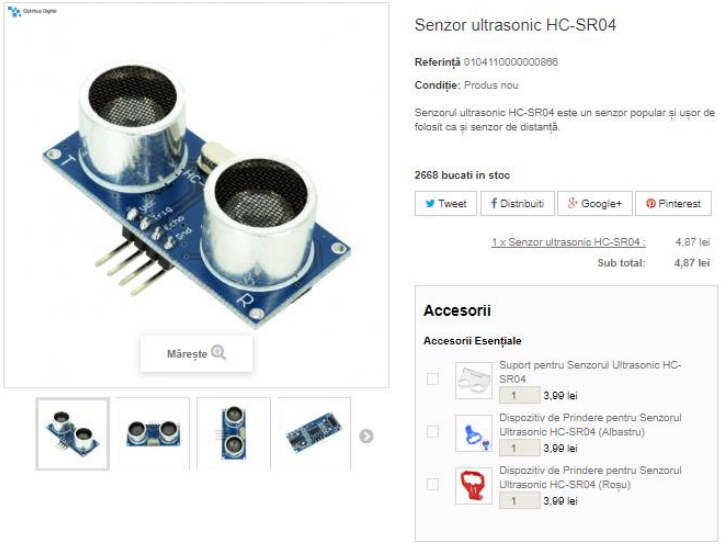
- https://www.optimusdigital.ro/ro/audio-buzzere/635-buzzer-activ-de-3-v.html?search_query=buzzer&results=48

Soil_Moisture



- https://www.optimusdigital.ro/ro/senzori-senzori-de-umiditate/73-senzor-de-umiditate-a-solului.html?search_query=senzor+de+umiditate+a+solului&results=2

Distance Sensor - Pir-hc-sr501

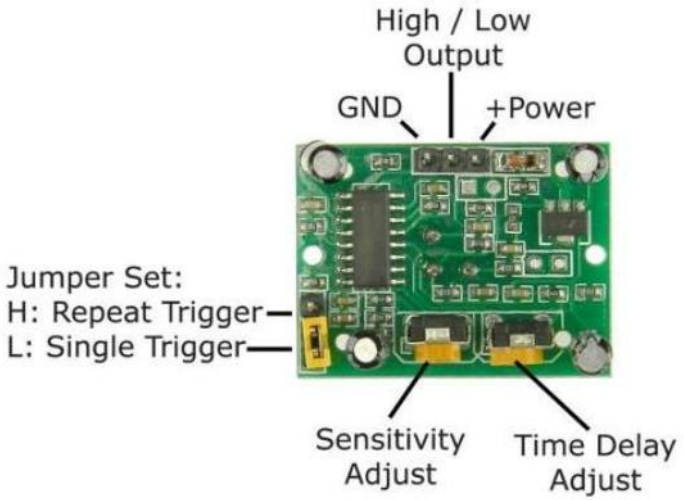


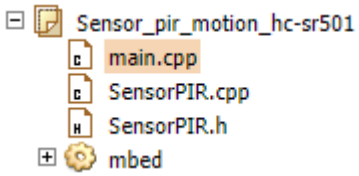
main.cpp | main.cpp | main.cpp | main.cpp | main.cpp | SensorPIR.cpp

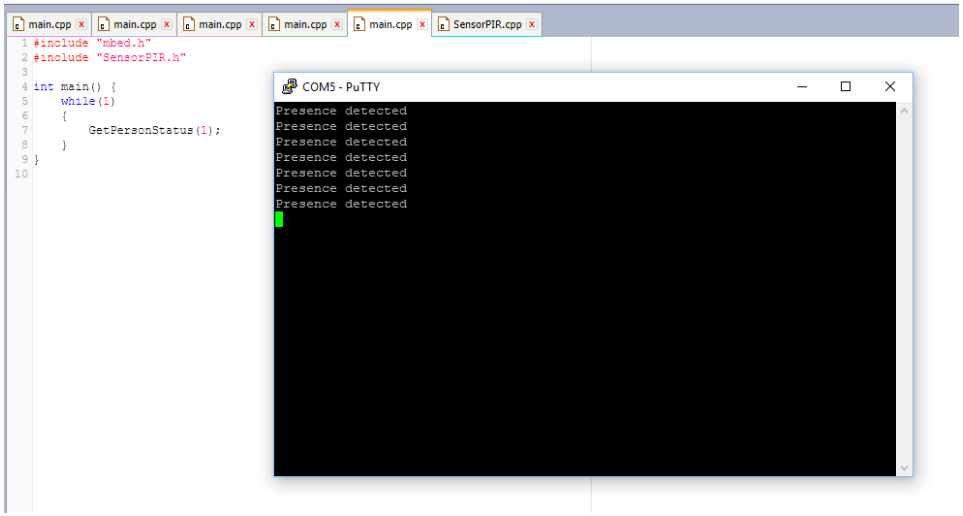
```
//*********************************************************
 // Here is a sample code usage
//*********************************************************
#include "hcsr04.h"
                  //triger, echo
HCSR04  usensor(D15,D14);
//VCC to 5V

Serial pc(USBTX, USBRX);
int main()
{
    unsigned char count=0;
    while(1) {
        usensor.start();
        wait_ms(500);
        int dist=usensor.get_dist_cm();
        pc.printf(" cm:%d ",dist);
        count++;
    }
}
```

- https://www.optimusdigital.ro/ro/senzori-senzori-ultrasonici/9-senzor-ultrasonic-hc-sr04-.html?search_query=ultrasonic+hr+sr04&results=25

Motion Sensor - Pir-hc-sr501







- https://www.optimusdigital.ro/ro/senzori-senzori-pir/106-modul-senzor-pir-hc-sr501.html?search_query=senzor+pir+hc+sr501&results=2

Temperature-Air

Modul Senzor de Temperatură DHT11 cu LED

Referință 0104110000036803

Condiție: Produs nou

Modul Senzor de Temperatură DHT11 cu LED

315 bucati in stoc

Tweet | Distribuiti | Google+ | Pinterest

Scrie o recenzie

Imprimă

```
#include "DHT11.h"

PinName myledD15(D15);
// Humidity sensor
DHT11 d(myledD15);

Serial pc(USBTX, USBRX);

int main() {
    int state;

    while(1) {

        state = d.readData();

        if (state != DHT11::OK) {

           printf("\n Error: %d", state);
        } else {

            printf("\n T: %dC, H: %d%%", d.readTemperature(), d.readHumidity());
        }

         //pc.printf("\n T: %dC, H: %d%%", d.readTemperature(), d.readHumidity());


        wait(2.0);
```

Speed: 9600

```
PinName myledD15(D15);
// Humidity sensor
DHT11 d(myledD15);

Serial pc(USBTX, USBRX);

int main() {
    int state;

    while(1) {

        state = d.readData();

        if (state != DHT11::OK
           printf("\n Error: %
        } else {
            printf("\n T: %dC,
        }
         //pc.printf("\n T: %d

        wait(2.0);
```

COM5 - PuTTY

```
T: 29C, H: 43%
T: 29C, H: 44%
T: 29C, H: 44%
T: 29C, H: 44%
T: 28C, H: 43%
T: 2
8C, H: 43%
T: 28C, H: 43%
T: 28C, H: 43%
T: 28C, H: 43%
T: 28C, H: 43%
T: 28C, H
: 43%
T: 28C, H: 44%
T: 28C, H: 44%
T: 28C, H: 45%
T: 28C, H: 56%
T: 28C, H: 65%
T: 28C, H: 72%
T: 28C, H: 77%
T: 29C, H: 82%
T: 29C, H: 86%
```

- https://www.optimusdigital.ro/ro/senzori-senzori-de-temperatura/4762-modul-senzor-de-temperatura-dht11-cu-led.html?search_query=dht11&results=19

References & Sources

[Noergaard2005] *Tammy Noergaard,* Embedded Systems Architecture, A Comprehensive Guide for Engineers and Programmers, Elsevier, 2005

[fritzing] http://fritzing.org/download/

[ARM-RESD] **ARM-based Rapid Embedded Systems Design Education Kit**
https://developer.arm.com/academia/arm-university-program/for-educators/embedded-system-design/course-details

[ARM] Architecture Reference Manual,
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.subset.architecture.reference/index.html

[ARM_M0]Cortex™-M0 Technical Reference Manual,
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0432c/index.html

[ARM_devices] Cortex-M4 Devices Generic User Guide,
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/index.html