Università
della
Svizzera
italiana

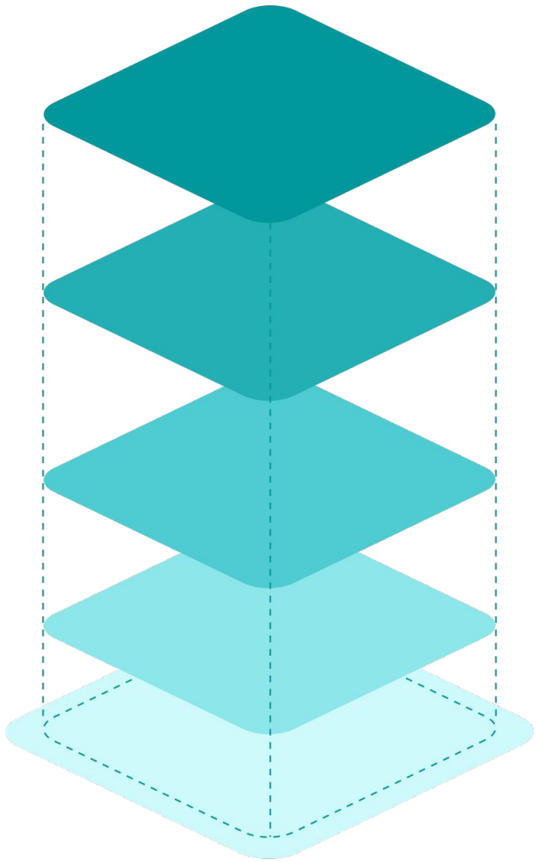
Facoltà
di
scienze
informatiche

Edge Computing in the IoT

Arduino Platform Hands-on

Alberto Ferrante
TA: Luca Butera

The Arduino abstraction



ARDUINO CODE

MBED APPLICATON CODE

MBED OS API

SECURITY APIS

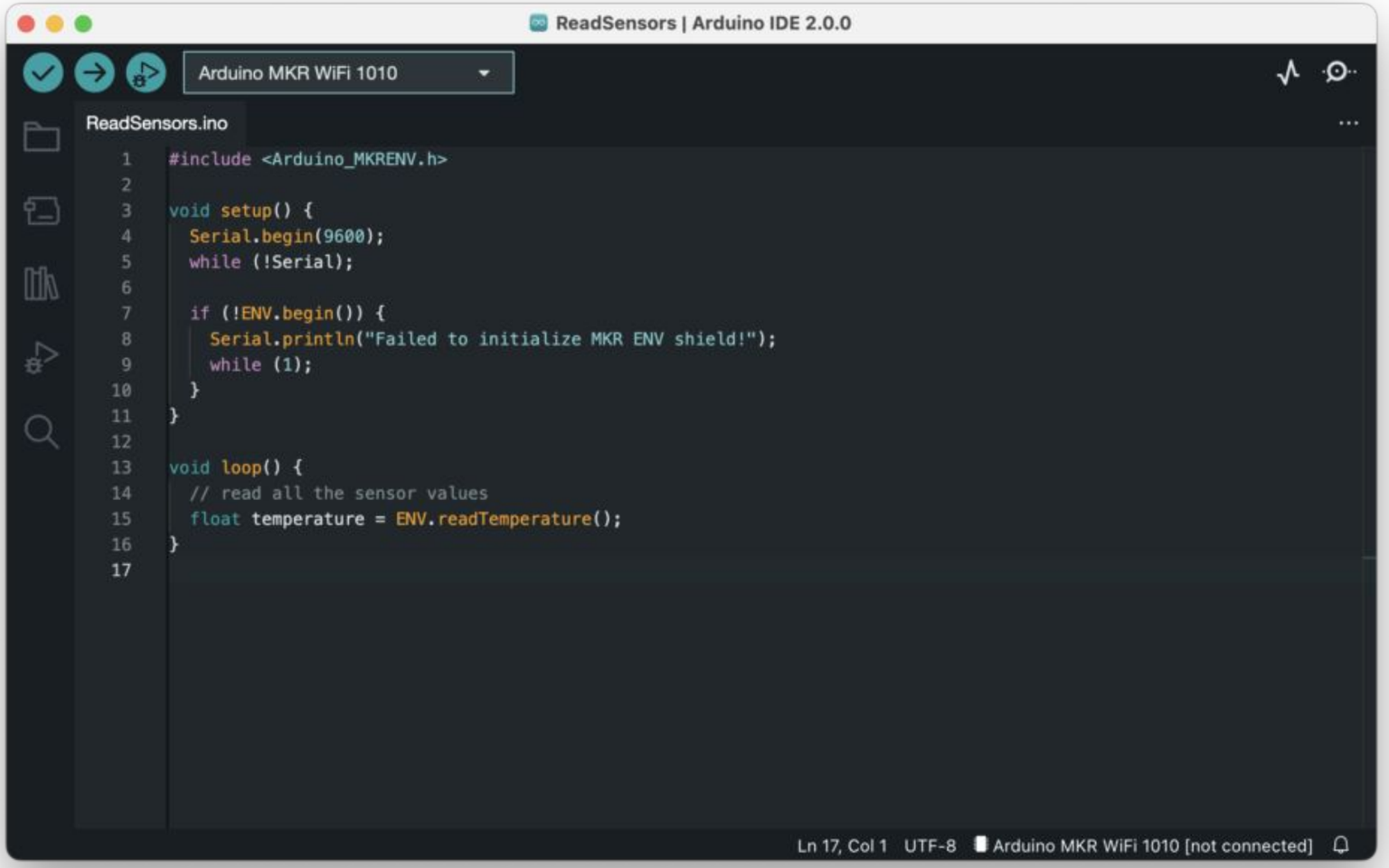
HARDWARE INTERFACES

We work mostly here

Arduino platform gives us this

Our boards work here

The Arduino IDE 2.0



IDE 2.0 available for download at: arduino.cc/en/software

Cloud IDE available at: cloud.arduino.cc/home/

Blink LED

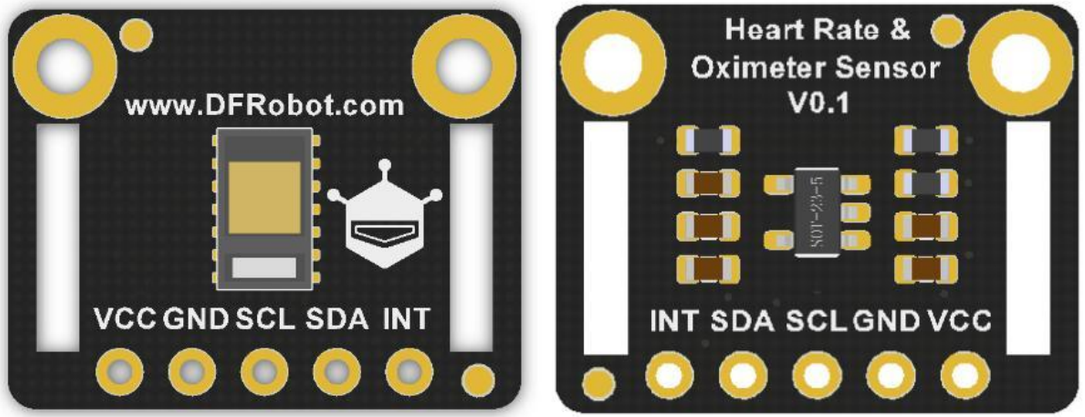
```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);
}

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

This example code can be found in the **Arduino IDE 2.0** under:

File -> Examples -> 01.Basics -> Blink

Pulse Oximeter Sensor

MAX30102

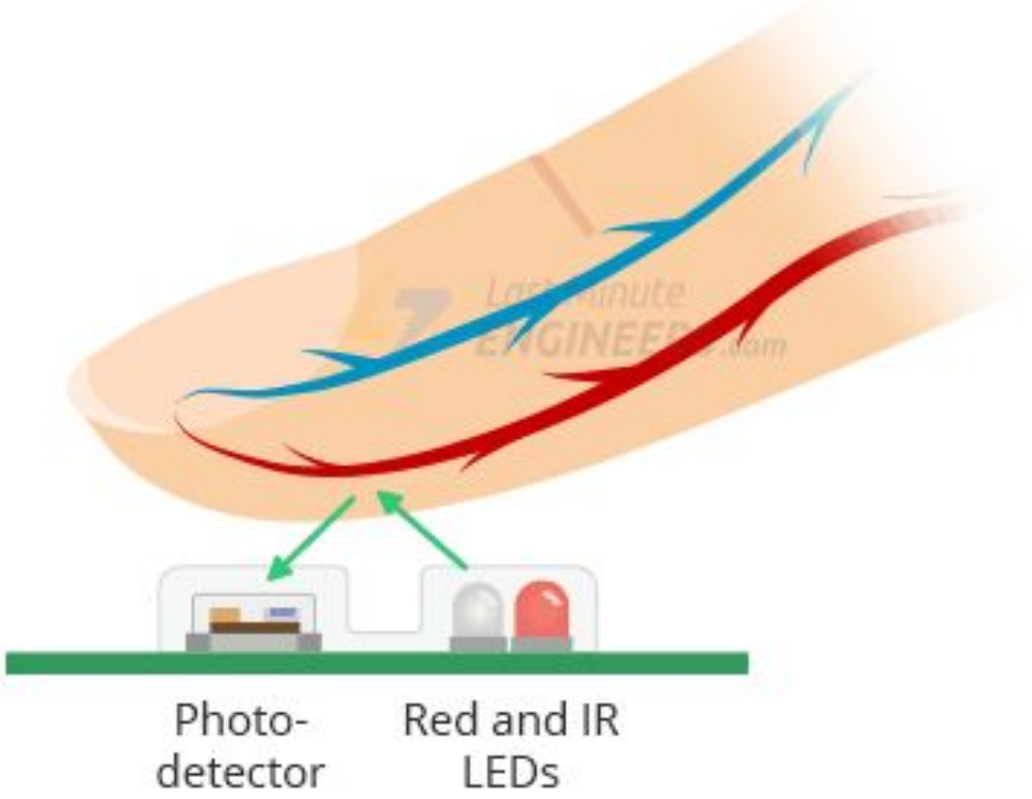


Sensor wiki:
https://wiki.dfrobot.com/Heart_Rate_and_Oximeter_Sensor_SKU_SEN0344

Sensor library:
https://github.com/DFRobot/DFRobot_MAX30102
or
https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library
(directly available in the IDE)

Communicates through I2C.

Working Principle

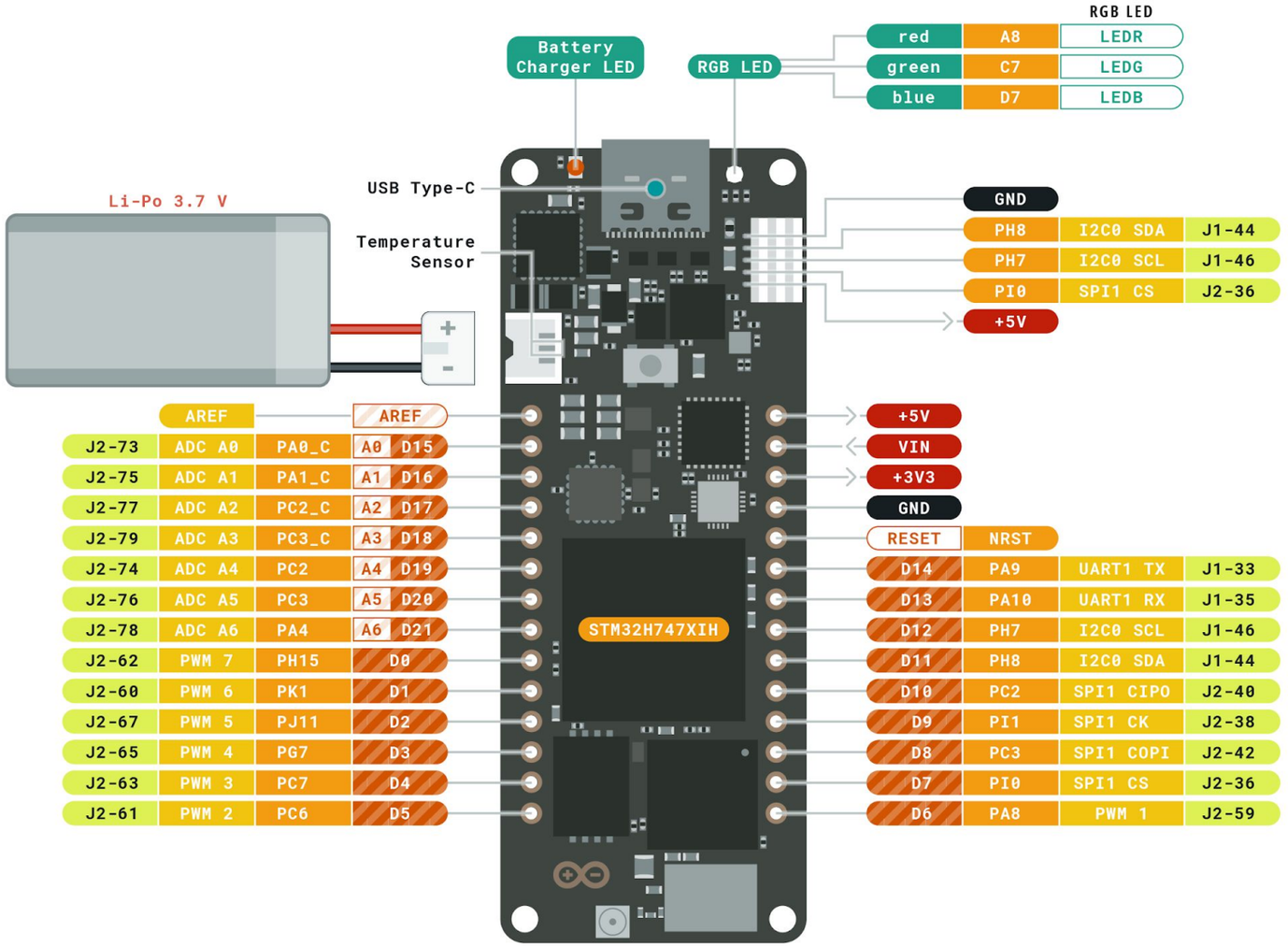


A RED and IR LEDs are shone through the skin and reflected onto a photo detector.

The amount of RED and IR light that is reflected depends on the bloodstream oxygen levels.

Wiring Diagram

Sensor	Arduino	Role
VCC	+5V/+3V3	Power
GND	GND	Ground
SCL	D12	I2C Clock Line
SDA	D11	I2C Data Line
INT	D13	Interrupt Pin



Debugging: code preparation

Add the following before your code and use the **Sketch -> Optimize for Debugging** option.

```
#include <ThreadDebug.h>
UsbDebugCommInterface debugComm(&SerialUSB);
ThreadDebug threadDebug(&debugComm, DEBUG_BREAK_IN_SETUP);

// Your code…
```

Execute **Sketch -> Export Compiled Binary**

Debugging: running the debugger

Things we need:

- gdb debugger executable → ~/Library/Arduino15/packages/arduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-gdb
- arduino board port → /dev/cu.usbmodem101
- compiled sketch file → ~/Documenti/Arduino/blink/Blink/build/arduino.mbed_nano.nano33ble/Blink.ino.elf

Populate the placeholders and run this command in your terminal.

```
<gdb> -ex "set pagination off" --baud <int> -ex "set target-charset ASCII" -ex "target remote <port>" <sketch>
```

```
~/Library/Arduino15/packages/arduino/tools/arm-none-eabi-gcc/7-2017q4/bin/arm-none-eabi-gdb -ex "set pagination off"
--baud 230400 -ex "set target-charset ASCII" -ex "target remote /dev/cu.usbmodem101"
~/Documenti/Arduino/blink/Blink/build/arduino.mbed_nano.nano33ble/Blink.ino.elf
```