UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Embedded System

A.Y. 2025/2026

RELATORI
Elisa Riceputi

SEDE
University of Bergamo, Dalmine

DATA
16/03/2026

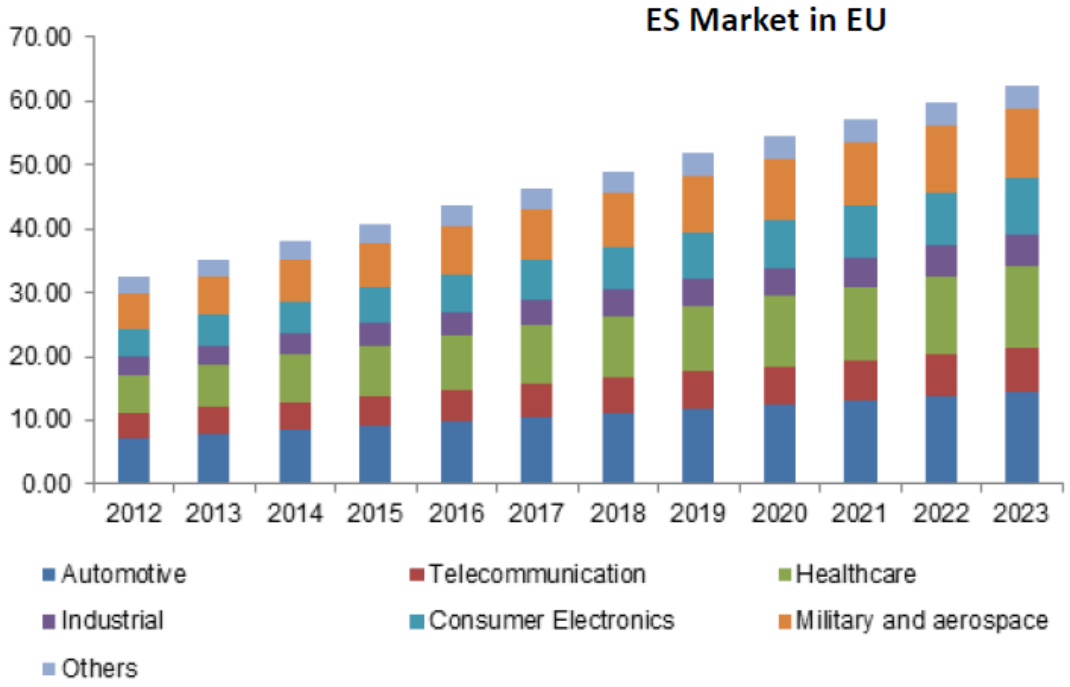
What is an “embedded system”?

- A special-purpose computer system designed to perform **one or a few dedicated functions**, often with real-time computing constraints
  - In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming
- It is usually embedded as part of a complete device including hardware and mechanical parts

- An embedded system:
  - Takes advantage of application characteristics to **optimize the design**
  - **Responds, monitors** and **controls** the external environment using sensors and actuators

Distribution of embedded systems

- Embedded systems control many of the common devices in use today



- ~80 million PCs vs ~3 billion embedded CPUs annually!

Some examples

- ATM and POS
- Mobile phones
- Printers and scanners
- Ultrasound scanners, medical scanners for magnetic resonance, ecc…
- Microwave ovens, washing machines, ecc.
- Digital scale

Wearable sensor systems

Such technology progress has inspired and pushed forward the development of **embedded wearable sensor systems**

Applications

- **Physiological parameters** monitoring
  (HR, blood pressure, body temperature, etc.)
- **Environmental parameters** monitoring
  (humidity, temperature, pressure, etc.)

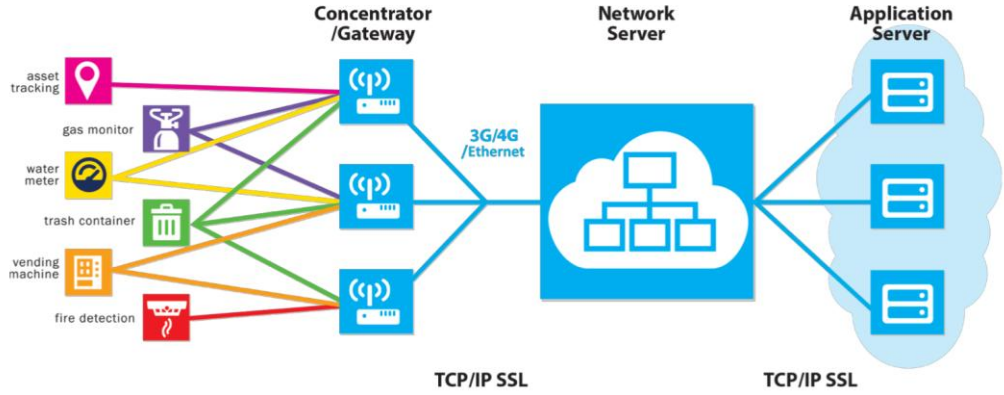
Field of application

- Medical
- Sport
- Fitness
- Wellness

Internet of Things (IoT)

Portable and **fully autonomous systems** can be **interconnected** to exchange information: they contributed to the birth of the so-called **Internet of Things (IoT)**

1. Physical devices interconnected one another which collect and exchange data
2. Embedded sensors detect information about the surrounding
3. Information is exchanged through wired/radio communication with the Internet
4. Services on the cloud provide key parameters from the massive amount of info collected



As the complexity of these systems increases, particular care must be taken at the *design stage* in order to optimally exploit the **number of functionalities**, the **processing power**, the **communication layer** and the **data collected**

1 trillion new IoT devices expected to be produced by **2035**!

System blocks

Specifications: engineer side

- **Inputs:** *"What data will be collected, what are the controls?"*
- **Sample rate:** *"What interval does it need to collect data?"*
- **Outputs:** *"What will be transmitted, displayed or controlled?"*

- **Processing:** *"What processing needs to be done on the data?"*
- **Storage:** *"What data needs to be stored? For how long?"*

- **Communication:** *"What data needs to be sent and received? How?"*

- **Power:** *"How long does it need to last between charges?"*

- **Form factor:** *"How will the device fit into the physical world?"*
- **Cost:** *"How much will the users be willing to pay?"*

Specifications: customer side

- **Inputs:** “Accelerations and angular rates”
- **Sample rate:** “1000 of samples per second (= 1 kHz)”
- **Outputs:** “Motion trajectory”

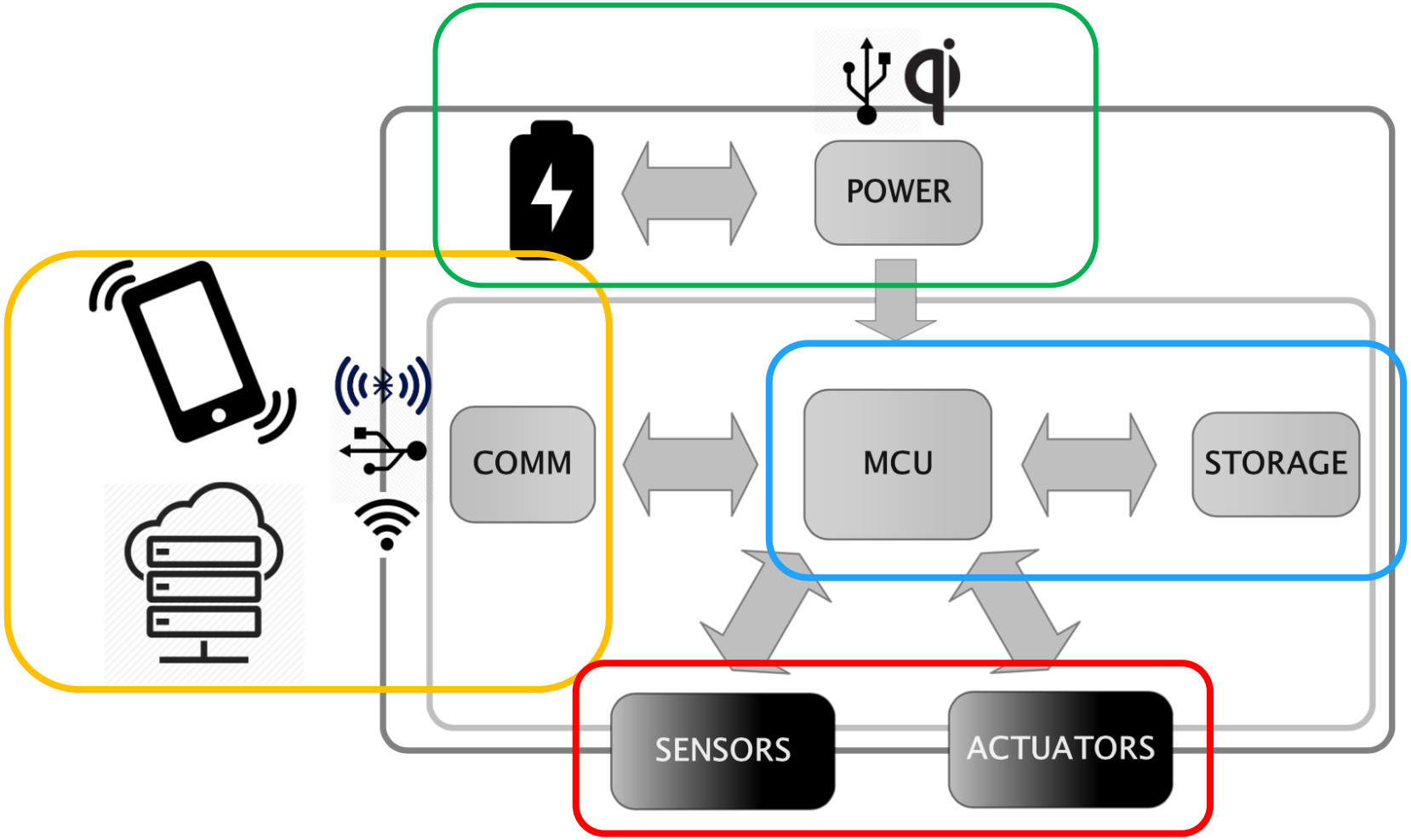
- **Processing:** “It depends: are we computing locally or in the cloud?”
- **Storage:** “It depends: are we computing locally or in the cloud?”

- **Communication:** “Processed data transmitted **wirelessly** at 250 Hz”

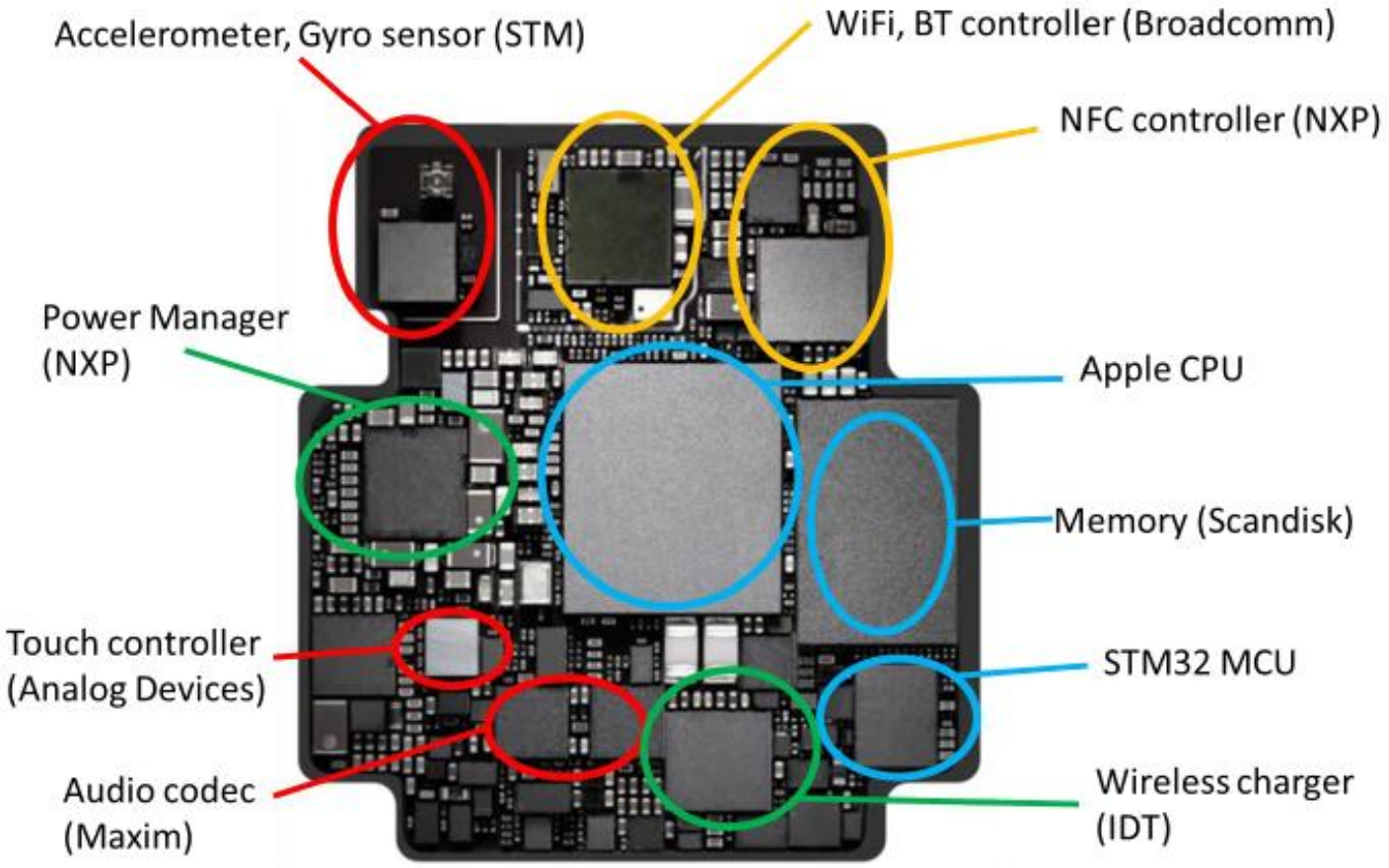
- **Power:** “At least 24 hours”

- **Form factor:** “It should be small enough to be mounted on a wrist”
- **Cost:** “150 €”

Wearable sensor system: Block diagram



Example: Apple watch

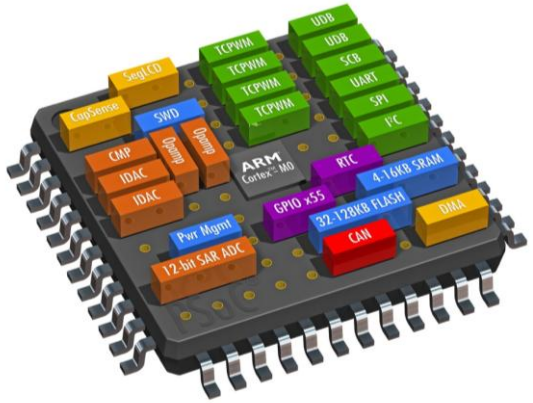


Wearable device challenges

- Typical embedded system challenges:
  - **Small size, low weight**
    - Handheld electronics
    - Transport cost
  - **Low power**
    - Battery powered for 8+ hours
    - Small size means small thermal dissipation capabilities
  - **Safety critical operation**
    - Embedded systems work in danger-prone environments (Industrial, Biomedical,...)
  - **Cost sensitivity**

Microcontroller Unit (MCU)

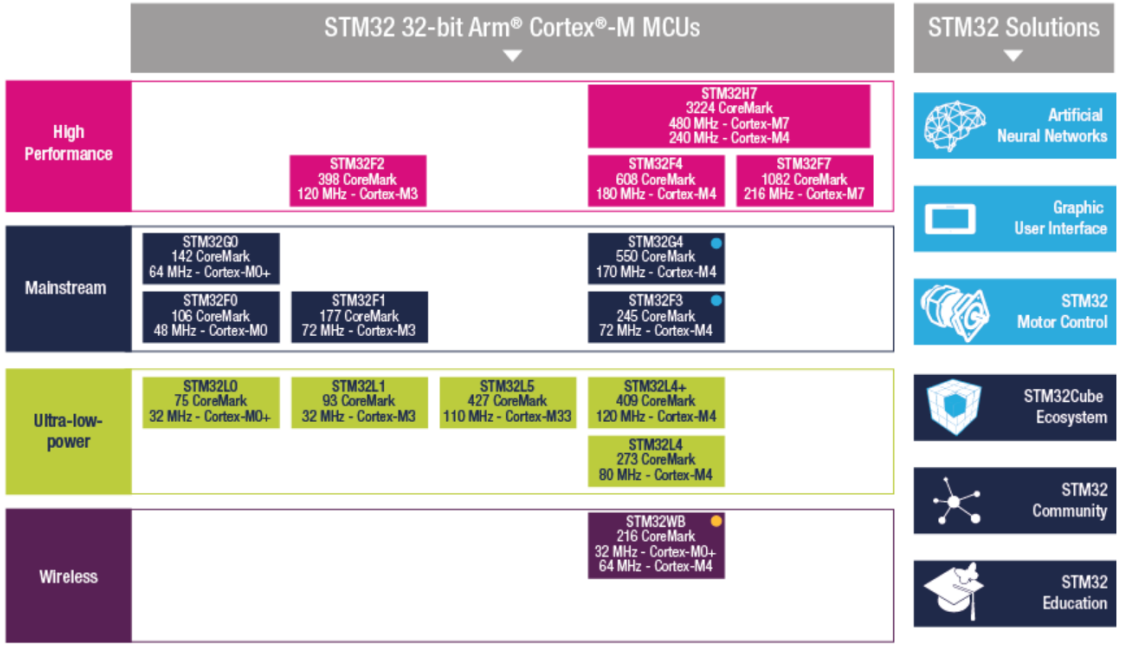
- System operation is managed by the MCU core block
- An **MCU** is an embedded electronic computing device that includes **several components on a single chip**
  - Microprocessor
  - RAM and flash memory
  - Input/output ports
  - ADCs, timers
  - Interfaces with other blocks



- High performance architectures integrate:
  - *Digital Signal Processor* (**DSP**), to measure, filter or compress continuous real-world analog signals
  - *Floating Point Unit* (**FPU**), a coprocessor specialized in carrying out operations on floating point numbers

ARM architecture

- Among the families of processor architectures available, Advanced Risc Machine (**ARM**) architecture is the most used
- The **32-bit ARM Cortex-M** architecture is optimized for embedded processing and microcontroller applications, coming with several core implementations designed for different requirements:



Main features of STM32 class MCUs

	Cortex	Clock [MHz]	Stop [μA]	Stand-by [μA]	Flash [kB]	Ram [kB]	FPU	Performance	Cost
STM32L0	M0	32	≤ 1	≤ 1	192	20	no	low	low
STM32L1	M3	32	1 - 1.5	1 - 1.5	256	16	no	low	medium
STM32L4	M4	80	1 - 2	$\leq$0.5	1024	160	yes	high	high
STM32F0	M0	48	2 - 3	1 - 2	256	32	no	low	low
STM32F1	M3	72	20 - 30	1 - 3	1024	96	no	medium	medium
STM32F2	M3	120	350	2-3	1024	128	no	medium	medium
STM32F4	M4	180	40 - 300	2 - 5	2048	384	yes	high	medium
STM32F7	M7	216	150 - 400	2 - 5	2048	256	yes	high	high

Low-power applications

- Long-term monitoring (e.g. environmental monitoring), simple processing techniques

High performance applications

- Short-term monitoring (e.g. motion tracking), complex analysis

Sensors alternatives

- There are often different sensor implementation to measure a given quantity
- It is best to carry out these choices during the specifications drafting, and leave specific sensors (aka vendor) choices for this phase
- Broadly speaking, it is best to take into account:
  - **Measurement range:** the measured quantity must fall inside the measurement range, possibly with a bit to spare a “guard”
  - **Accuracy:** the accuracy of the sensor must be suitable for the application. There is a thing as too high of an accuracy, as it can weigh on the system
  - **ODR:** frequency of measurement is crucial to gathering the correct data

Memory alternatives

- The capability of retaining in a non-volatile way large amounts of data makes the system as most autonomous as possible
- The term **memory** refers to an electronic device for data storage

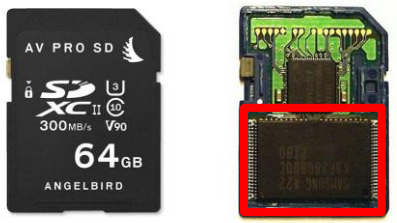
- Allowed operations
  - *Writing*: information storage operation
  - *Reading*: restoration operation of stored information

- Different storage approaches depending on:
  - The kind of data to be collected
  - The sampling frequency

On-board storage types

- **MCU-level storage**
  - Within the MCU flash memory
- **Dedicated blocks**
  - NAND flash memory
  - NOR flash memory
- **Secure-Digital (SD) cards**
  - Employment within wearable systems must be carefully considered due to:
    - Significantly higher cost and area occupancy
    - The need of a dedicated parallel interface, not always integrated within MCUs

Type	Area [mm2]	Capacity [Gb]	Speed [MB/s]	Supply voltage [V]	Cost
SD	165	≤16	2 - 25	3.3	medium
SDHC	165	32 - 16k	2 - 600	1.8 - 3.3	high
Serial NAND	60	≤1	6 - 13	1.8 - 3.3	high
Parallel NAND	60	1-16	20 - 40	1.8 - 3.3	low
Serial NOR	50	≤1	16	1.8 - 3.3	low



Flash memory

Popular communication methods

USB (Universal Serial Bus)

- Need of a wired connection
- Significantly important for charging purposes
- Less and less used for data exchange (due to the need for a wired connection)

Bluetooth v4.1 and Bluetooth Low Energy (BLE)

- Becoming a common solution for Body Area Networks implementation
- Relatively high throughput (≈20 kbps)
- Extremely low power consumption, down to few μA

Wi-Fi

- Connection to the Internet is typically implemented on devices where the power consumption is not a major concern, such as devices powered through the electrical grid

Wireless communications for IoT

- IoT applications have specific requirements
  - Long range (up to 1 km data transmission)
  - Low data rate (pieces of information to be transmitted)
  - Low energy consumption (→ battery saving)
  - Cost effectiveness

- The widely used **short-range radio technologies** (e.g. Bluetooth) are not suitable for scenarios that require long range transmission
- Solutions based on **cellular communications** (e.g. 2G, 3G, and 4G) can provide larger coverage, but they consume excessive device energy

- Therefore, IoT applications' requirements have driven the emergence of a new wireless communication technology: **Low Power Wide Area Network (LPWAN)**

BAN/WAN
(Wide Area Networks)

LPWAN
(Low Power WAN)

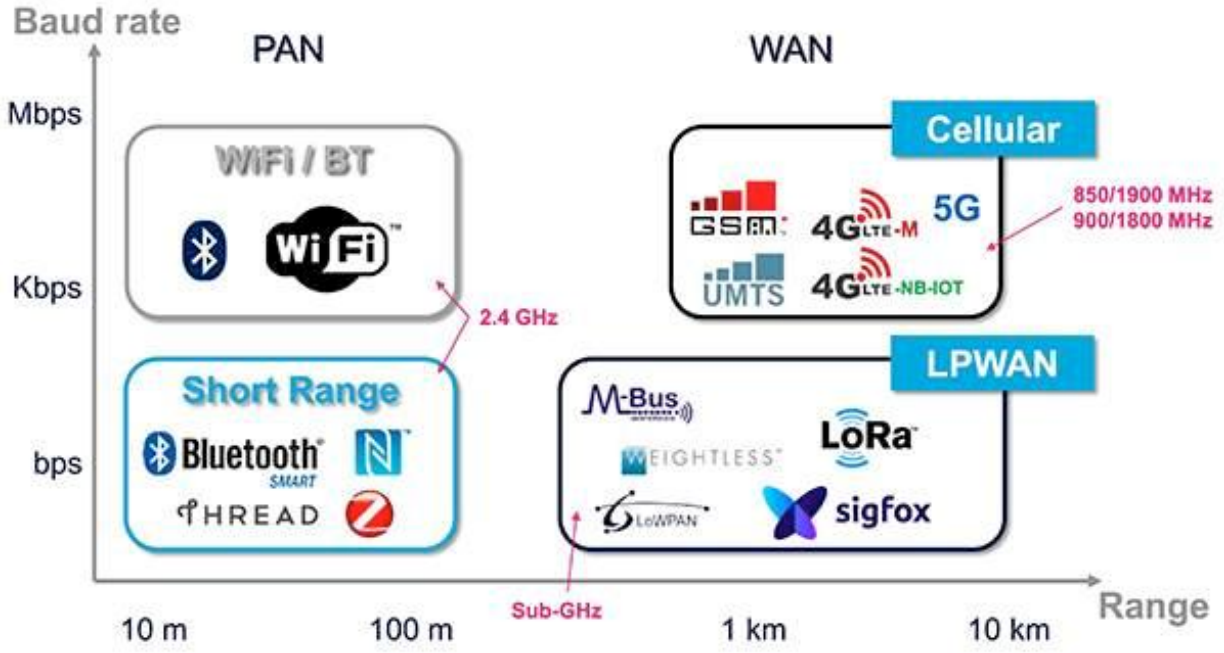
Wireless-network comparison

Body Area Network
- Short/medium range
- Good battery life
- Low number of nodes

Wireless LAN
- Short range
- Low battery life

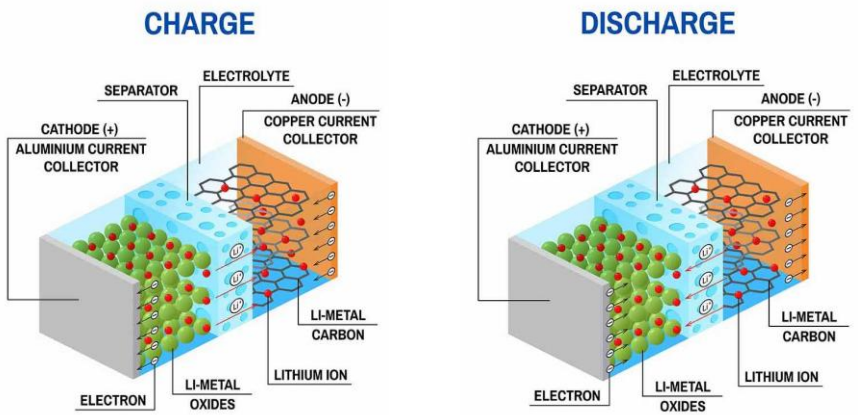
Cellular
- Low battery life
- High cost
- Mobile Network Operator-controlled

LPWAN
- Longest range
- Low data rate
- Low power consumption

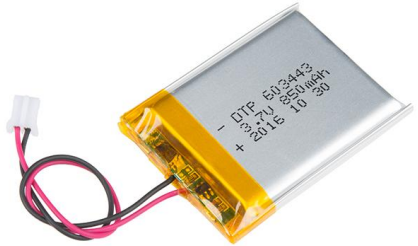


Power supply

- Supplied by a **rechargeable lithium-ion battery**
  - **Positive**: lithium-nickel-manganese-cobalt oxide
  - **Negative:** graphite
- Main features:
  - High density
  - Low discharge memory
  - Low self-discharge current

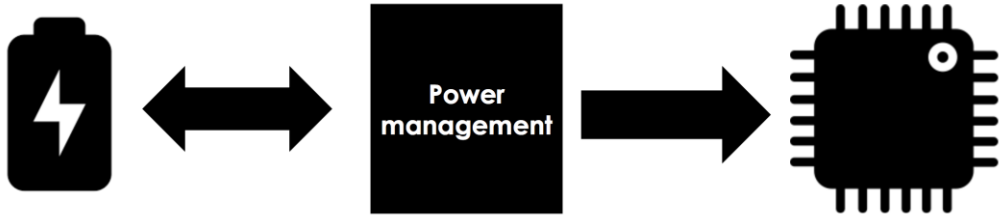


- Example
  - Nominal voltage: 3.7 V
  - Capacity: 800-3000 mAh
  - Maximum voltage during recharge: 4.2 V
  - Voltage when discharged: 3.2 V



Power management

- Typically, the building blocks of an embedded system do not support a *too high* supply voltage (e.g. 3.3 V provided by batteries)
- A dedicated section between the battery and the overall system is needed, in order to:
  - Provide a **scaled and regulated supply voltage** compatible with electrical specifications and other building blocks
  - **Optimize** the power consumption of the system



- These supplies are normally generated by DC-DC converters
- Most of the ICs used for IoT applications work with supply voltages as low as 1.8 V

Battery recharging

- For the system to fully recover the state of charge stored within the battery, a **periodic recharge** is needed

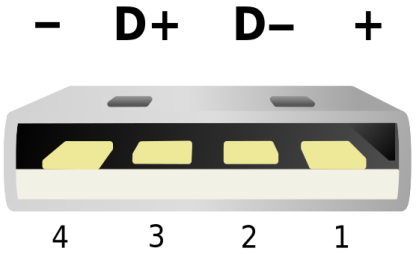
- This is obtained with specific ICs, named **battery chargers**, powered by a constant voltage source provided externally

- Three different recharging solutions
  - Wired recharging
  - Wireless recharging
  - Energy harvesting

Battery recharging solutions (1/3)

Wired

- Wired solution is commonly adopted, either with standard plugs (e.g. **USB cable**) and receptacles or with dedicated adapters specifically designed to match physical requirements of the system
- The same support can also be **used to exchange data** with the external world, if supported by the MCU



Battery recharging solutions (2/3)

Wireless

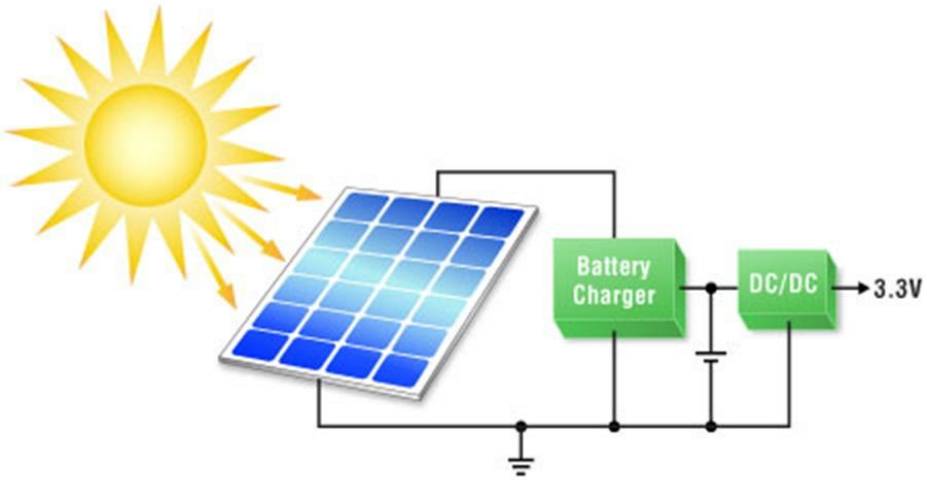
- If no electrical connection can be physically established or where the user experience demands it, wireless power transfer techniques can be employed to deliver power to the battery charger

- On the embedded side, **a receiving coil** acting as antenna **is needed to receive the power**, along with a dedicated circuit, implemented either with discrete components or with specific ICs

Battery recharging solutions (3/3)

Energy harvesting

- The battery can be assisted by energy harvesting solutions when:
  - The power consumption of the system is sufficiently low
  - Common recharging techniques are not easily viable
  - The budget cost allows it
- Solutions based on thermoelectric generators, photovoltaic cells or piezoelectric modules
- Low power budget (~ 100 μA)

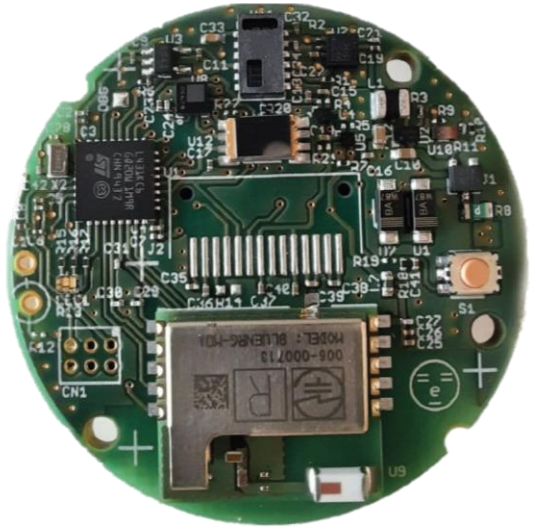


Development phase – Outline

- **Goal:** develop hardware and firmware to create system functionalities
  - **PCB development**
    - Schematic drawing
    - Layout drawing
  - **Firmware development**
    - Hardware and firmware link
    - Finite State Machine definition and implementation
    - Event-driven firmware development

PCB introduction (1/2)

- **P**rinted **C**ircuit **B**oards are, as the name suggest, **circuits printed onto a mechanical support**
- A PCB is essentially a collection of metal layers (interconnections) laminated between a non-conductive support (usually fiberglass weave)
- It is manufactured starting from the inner layers etching away unwanted metal areas
- More layers are then added (2 at a time, top and bottom) to this ”sandwich” and the process is repeated until the PCB is complete

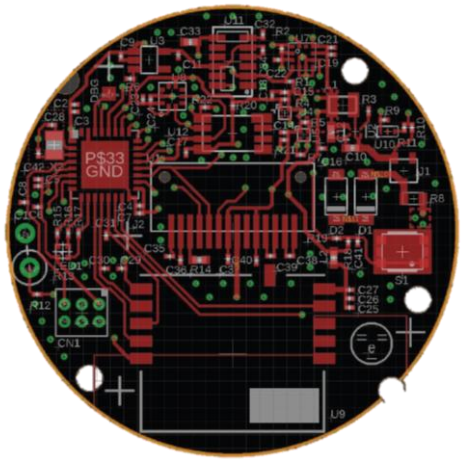


PCB introduction (2/2)

- A PCB can be manufactured with 3 elements:
  - Stackup

#	Name	Type	Thickness	#	Thru 1:6	Buried 2:5	µVia 1:2	µVia 6:5
	TopOverlay	Overlay						
	TopSolder	Solder Mask	0.025mm					
1	L1	Signal	0.036mm	1				
	Dielectric1	Prepreg	0.071mm					
2	L2	Signal	0.036mm	2				
	Dielectric 4	Prepreg	0.071mm					
3	L3	Plane	0.018mm	3				
	Dielectric2	Core	0.229mm					
4	L6	Plane	0.018mm	4				
	Dielectric3	Prepreg	0.071mm					
5	L7	Signal	0.036mm	5				
	Dielectric 5	Prepreg	0.071mm					
6	L8	Signal	0.036mm	6				
	BottomSolder	Solder Mask	0.025mm					
	BottomOverlay	Overlay						

  - Gerber

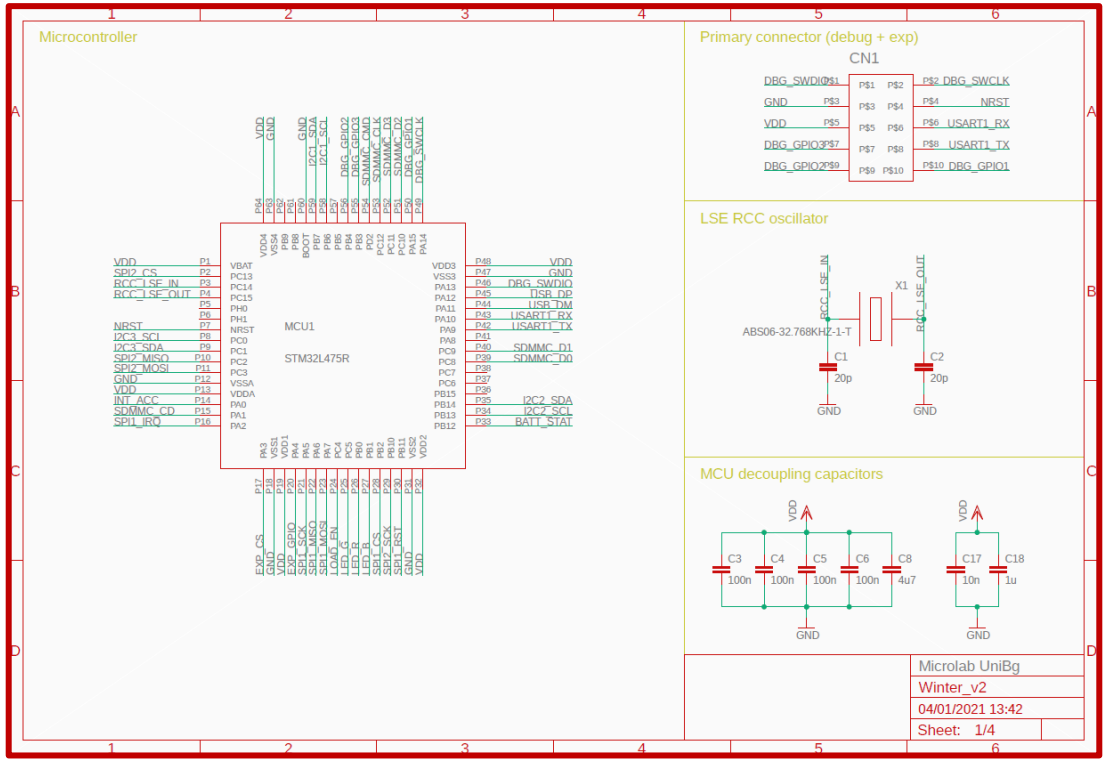


  - Bill of material (BOM)

DEVICE	Q.TY	MANUFACTURER	DESCRIPTION	VALUE	RATING	PACKAGE	PARTS
BLM15FD121SN1D	4	MURATA	SMD ferrite bead 120 Ω (L x W x H) 1.1 x 0.5 x 0.5 mm			1040	L10, L4, L5, L7
NC7SZ32L6X	1	FAIRCHILD SEMICONDUCTOR	TinyLogic UHS Two-Input OR Gate			MICROPACK6L	U13
0908-2-15-20-75-14-11-0	6	MILLMAX	Foot Sensor Wall, Spring loaded pin AU over NI 0.295in				TP1, TP2, TP3, TP4, TP9, TP10, TP11, TP12
90505-A0	4	preci-dip	Hip Sensor Wall, spring loaded pogo pin without a tail				TP5, TP6, TP7, TP8
ERJ-2GE0R00X	2	PANASONIC	Thick Film Resistors - SMD 0402 Zero ohms 5% Tol AEC-Q200	0R0		R0402	R6, L6
ERJ-2RKF10R0X	6	Panasonic	SMD 0402 10.0ohms 1%	10		R0402	R24, R27, R28, R30, R40, R41
MC0402X104M100CT	11	MULTICOMP	SMD Multilayer Ceramic Capacitor, 0402 [1005 Metric], 0.1 µF, 10 V, ± 20%	100n	10V	C0402	C3, C6, C8, C14, C15, C18, C25, C36, C76, C23, C27
GRM1555C1H101JA01D	0	Murata	SMD/SMT 0402 100pF 50volts C0G 5%	100p	50V	C0402	
105443-1101	1	Molex	Conn Micro USB Type B RCP 5 POS 0.65mm Solder RA SMD 5 Terminal 1 Port Embossed T/R				CN2
C1005X5R0J106M050BC	2	TDK	Multilayer Ceramic Capacitors MLCC - SMD/SMT 0402 6.3volts 10uF X5R 20%	10u		C0402	C39,C40
ERJ-2GE/154X	2	Panasonic	SMD 0402 150Kohms 5%	150k		R0402	R13, R21
GRM155R71C154KA12D	0	MURATA	Multilayer Ceramic Capacitor, GRM Series, 0.15 F, 10%, X7R, 16 V, 0402	150n		C0402	C80
GRM1555C1H150JA01D	4	MURATA	Ceramic capacitor SMD 0402 15 pF 50 V 5 %	15p		C0402	C3, C31, C37, C26
ERJ-2RKF1004X	2	PANASONIC	Cermet resistor 1 MΩ SMD 0402 0.1 W 1 % 100 ppm/°C	1M		R0402	R16, R25
GRM155R71H102KA01D	1	Murata	Capacitor 1nF	1n		C0402	C13
GRM155R60J105ME19D	10	Murata	Multilayer Ceramic Capacitors MLCC - SMD/SMT 0402 1uF 6.3volts X5R 20%	1u	6.3V	C0402	C2, C4, C5, C7, C16, C20, C21, C22, C28, C32
GRM155R60J225ME95D	1	Murata	Cap Ceramic 2.2uF 6.3V X5R 20% SMD 0402 85°C Paper T/R	2.2u	6.3V	C0402	C37, C12
GRM155R60J225ME15D							
7447927222C	1	Wurth Electronics	Fixed Inductors WE-PMCI 0806 Power 2.2uH 1.6A 112mOhms	2.2u		C0805	L1
LPS4018-222MR	1	COILCRAFT	Power Inductor, shielded, 20/30% tol, SMT, RoHS	2.2u			L3
C1608X5R1A226M080AC	2	TDK	C Series 0603 22 uF 10 V ±20 % Tolerance X5R SMT Multilayer Ceramic Capacitor	22u		C0603	C24, C57
RC0402JR-0727ORL ERJ-2RKF2700X RK73H1ETTP2700F	2	YAGEO	Surface Mount Thick Film Resistor, RC Series, 270 ohm, 63 mW, - 5%, 50 V, 0402 [1005 Metric]	270		R0402	R26, R43
ERJ-2GEJ273X	1	PANASONIC	Cermet resistor 27 kΩ SMD 0402 0.1 W 5 % 200 ppm/°C Panasonic	27k		R0402	R3
CRCW04022K00JNEDC	1	VISHAY	SMD 1/16watt 2Kohms 5%	2k		R0402	R12

PCB schematic (1/2)

- The **PCB schematic** is an **abstracted view of the interconnections** in the system
- All selected components are placed as abstract blocks and connected properly in this view
- Some connections can be modified iteratively if a different choice eases the layout process

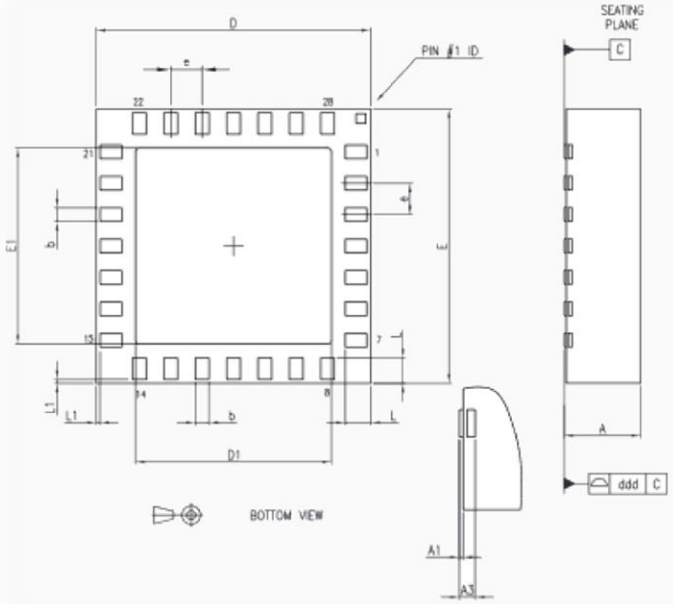


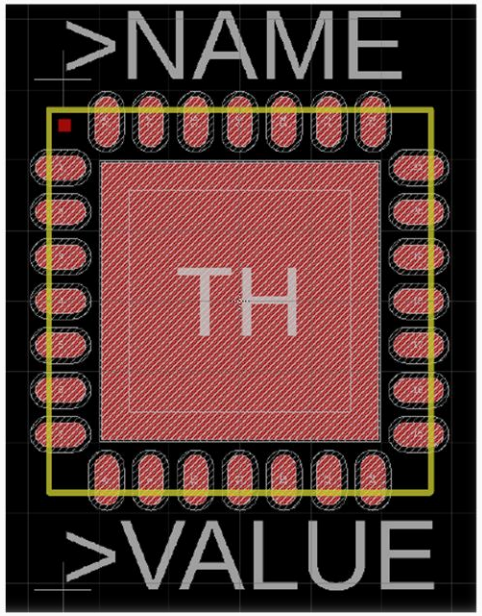
PCB schematic (2/2)

- Once the schematic is drawn, an **E**lectrical **R**ule **C**heck is performed, to lower the possibility of errors in the design
- Some examples of errors reported by the ERC:
  - floating input pins;
  - nets with multiple names;
  - power pins unconnected;
  - nets with only one pin;
  - …

Footprints

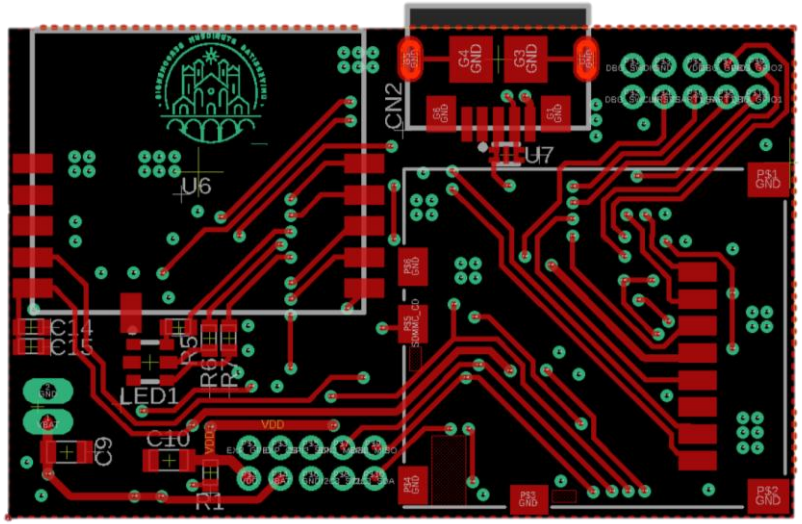
- Adding a component to a schematic, it must be linked to a symbol and to a footprint
- Although the former is useful to have drawn nicely, the latter is of the utmost importance
- A footprint represents the interconnection between a PCB and a IC and must be realized to specification





PCB layout (1/2)

- The PCB layout is the **implementation of the connections** drawn in the schematic
- All selected components are placed and represented as they actually will be on the final PCB
- All proportions in the layout are what will be present on the manufactured PCB
- The layout is actually realized by **tracing** (drawing) the **interconnections**, guided by the layout tool
- The tool uses information from the schematic to guide the process



PCB layout (2/2)

- Once the layout is drawn, a **D**esign **R**ule **C**heck is performed, ensure that the design can be fabricated
- Some examples of errors reported by the DRC:
  - holes too smalls;
  - routes too small;
  - gaps between routes too small;
  - components too close to each other or to the PCB border;
  - ...

Placement

- Component placement refers to the process of arranging the components onto the board

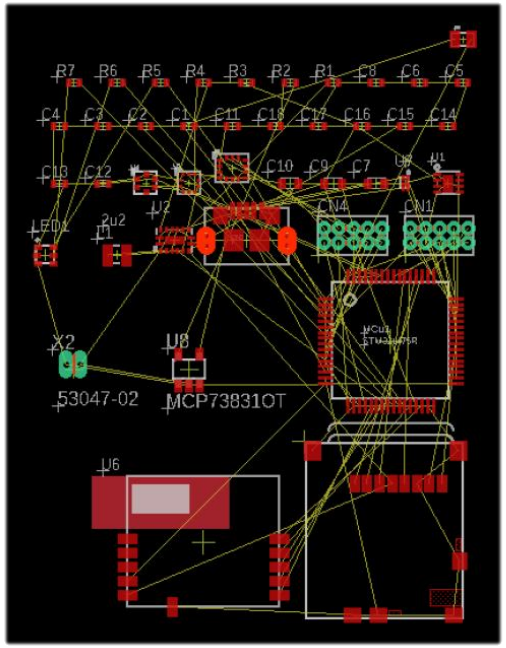
- It is done before starting to trace the routes with the goal of:

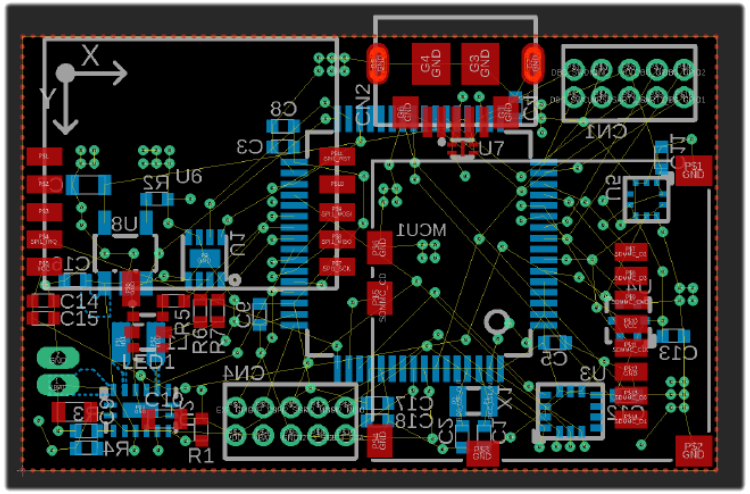
  - **verifying that all components fit** nicely inside the defined board shape/dimensions

  - creating (physical) **separation between analog and digital** components

  - finding out best position and orientation to **minimize trace length and complexity**

  - **identifying pin alternatives** that can lead to better layouts





Routing

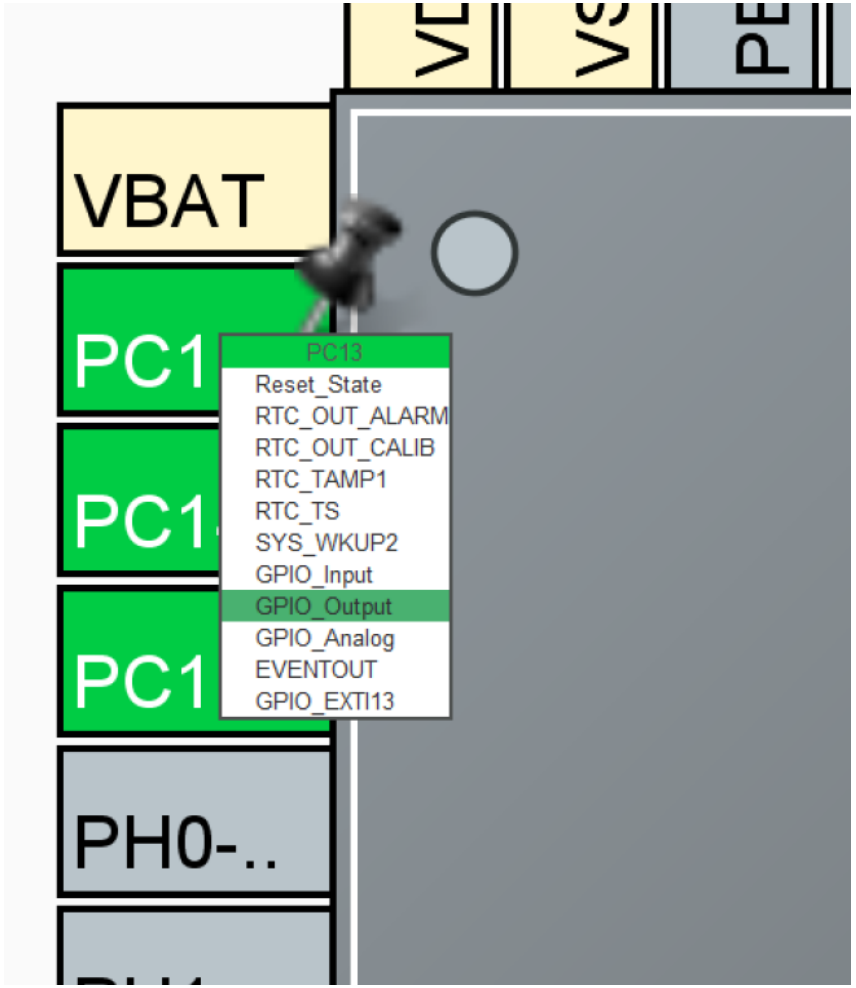
- Routing is a process that can be easy to approach, but difficult to master
- Some guidelines to keep in mind when routing:
  - keep traces short & direct
  - always have a ground plane
  - keep the ground plane as uninterrupted as possible and route high frequency traces ”on top” of the ground plane
  - try to avoid 90° turns on traces
  - size traces according to current; rule of thumb 10 mils $\cong 0.3\ \mathrm{A}$

Hardware and firmware link

- It is necessary to **define all the interfaces between** the already designed **PCB and the firmware** as a first step of its development
- The microcontroller is the bridge between the hardware and the firmware, it must be configured
- Configuration can happen by simpling writing code that configures the microcontroller on startup, OR by using tools to generate this configuration code
- This process will define:
  - Pin configuration
  - Internal peripherals settings
  - Clock tree

Pin configuration

- Microcontroller pin functions must be defined
- Using a tool to do so simply means assigning a function to each pin
- Remember that communication peripherals require more than one pin!



GPIOs (1/2)

- Most of the microcontroller pins can be set as **G**eneral **P**urpose **I**nput **O**utput (GPIO), which are a part of the microcontroller interface toward the extern (PCB)

- They must be set to a specific function:

  - input
  - output
  - Interrupt

- The internal circuitry of the GPIO can also be set:

  - pull-up/pull-down
  - open-drain/push-pull

GPIOs (2/2)

- Function-based GPIO classification:

Output

Digital output; can be either high (1) or low (0)
Cannot source much current (15 mA to 20 mA max)

Input

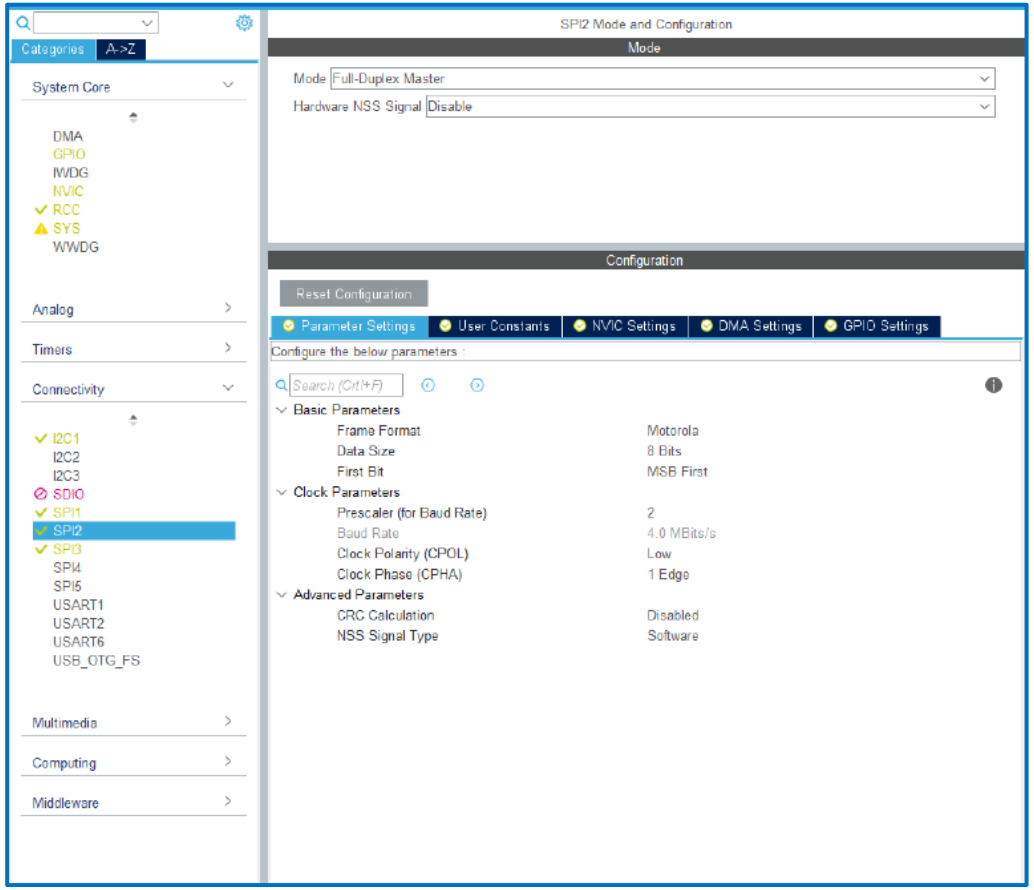
Digital input; can read either high (1) or low (0)
High input impedance

Interrupt

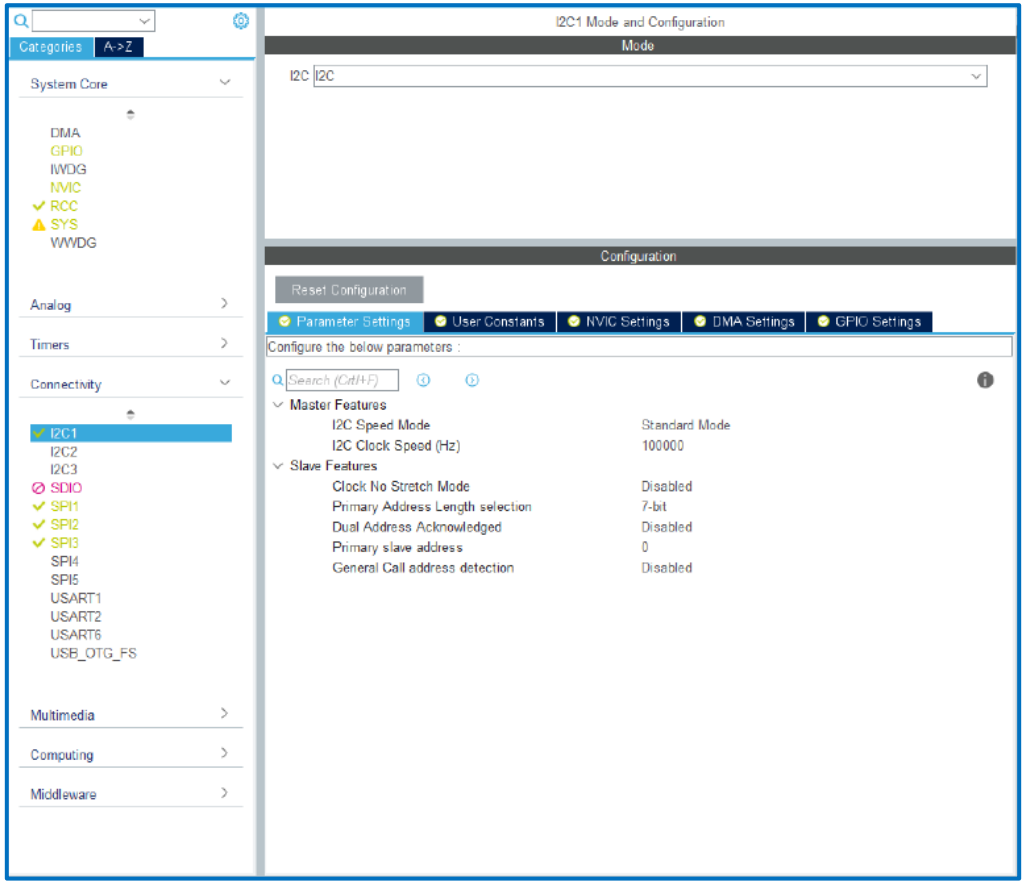
Digital input, but edge triggered (configurable)
An edge on this pin creates a block of current code execution to allow a specific response

Peripheral configuration

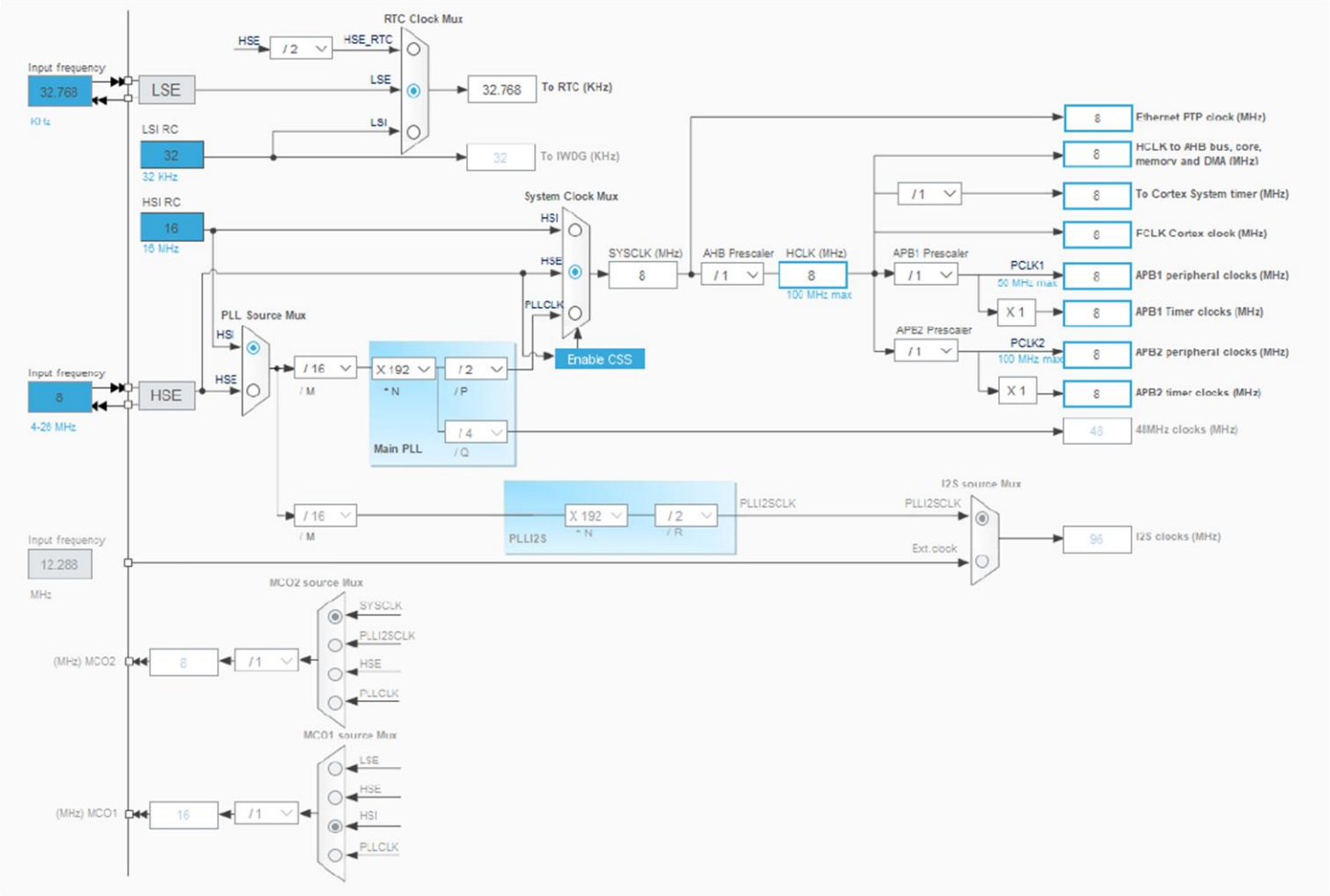
SPI Configuration



I2C Configuration



Clock tree


RTC Clock Mux
HSE
/ 2
HSE_RTC
LSE
LSI
Input frequency
32.768
KHz
LSE
32.768
To RTC (KHz)
LSI RC
32
32 KHz
32
To IWDG (KHz)
HSI RC
16
16 MHz
System Clock Mux
HSI
HSE
PLLCLK
SYSCLK (MHz)
8
AHB Prescaler
/ 1
HCLK (MHz)
8
100 MHz max
APB1 Prescaler
/ 1
PCLK1
50 MHz max
X 1
APB2 Prescaler
/ 1
PCLK2
100 MHz max
X 1
8
Ethernet PTP clock (MHz)
8
HCLK to AHB bus, core, memory and DMA (MHz)
8
To Cortex System timer (MHz)
8
FCLK Cortex clock (MHz)
8
APB1 peripheral clocks (MHz)
8
APB1 Timer clocks (MHz)
8
APB2 peripheral clocks (MHz)
8
APB2 timer clocks (MHz)
48
48MHz clocks (MHz)
Enable CSS
PLL Source Mux
HSI
HSE
/ 16
/ M
X 192
* N
/ 2
/ P
/ 4
/ Q
Main PLL
Input frequency
8
4-26 MHz
HSE
/ 16
/ M
PLLI2S
X 192
* N
/ 2
/ R
PLLI2SCLK
I2S source Mux
PLLI2SCLK
Ext.clock
96
I2S clocks (MHz)
Input frequency
12.288
MHz
MCO2 source Mux
SYSCLK
PLLI2SCLK
HSE
PLLCLK
(MHz) MCO2
8
/ 1
MCO1 source Mux
LSE
HSE
HSI
PLLCLK
(MHz) MCO1
16
/ 1

Clock tree glossary

- **LSE L**ow **S**peed **E**xternal: microcontroller input for a low speed (32 kHz) external oscillator
- **HSE H**igh **S**peed **E**xternal: microcontroller input for a high speed (4 MHz to 26 MHz) external oscillator
- **LSI L**ow **S**peed **I**nternal: internal RC oscillator for a 32 kHz clock
- **HSI H**igh **S**peed **I**nternal: internal RC oscillator for a 16 MHz clock
- **PLL P**hase **L**ocked **L**oop: versatile circuit, used in this case as a frequency multiplier

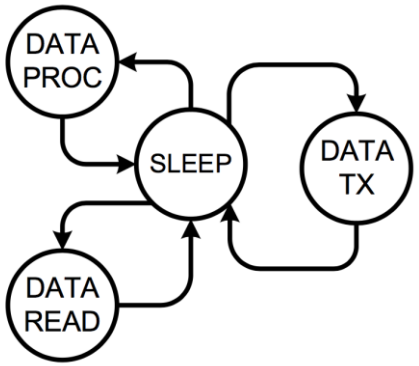
Finite State Machine

- An abstract machine that can be in exactly one of a finite number of states at any given time
- It can change from one state to another in response to some external inputs and/or a condition is satisfied

- E.g.: from the sleep state, the MCU moves into the *data collection* state, the *data processing* state and the *data transmission* state

Data processing
- Sensor fusion algorithms
- Real-time features analysis
- Trigger to actuators

Data readout
- Interrupt driven
- Analog/digital interface



Data transmission
- Network connectivity
- Stream vs burst

Sleep
- Stand-by mode (1 uA or lower)
- Stop mode (1 uA - 100 uA)