Politecnico di Torino

Master of Science in Electronic Engineering

Electronic systems engineering

FLYING MOUSE:

A flying Bluetooth mouse, based on LSM6DS0 MEMS inertial module: 3D gyroscope and 3D accelerometer

Author:

Edoardo Degrassi

Professor:

Eros Pasero

June 20th, 2016.

Contents

Figures index

Tables index

Chapter 1: Introduction

Flying mouse: a flying Bluetooth mouse based on a MEMS 3D accelerometer and 3D gyroscope.

Thanks to the Bluetooth wireless capability the *Flying mouse* is very flexible and versatile; in fact, unlike most other wireless device technology, Bluetooth technology is built into most of today's laptops and desktops. For this reason the product can be connected to a lot of computers as well as on smartphones and tablets. The *Flying mouse* can free the users' desktop and laptop bags of tangled wires, without the needing of an external USB dongle. The key novelty is the fact a table where to place the mouse is no more needed, allowing to use it on the sofa, on a plane or simply from a far distance. The key limitation is the fact the Bluetooth module cannot enter into a suitable sleep mode when it is connected and it leads to a great power consumption limiting the battery duration, but on the other hand eliminates the mess for the user to lost the connection while using the mouse. The user can keep in his hand the device as a remote control device.

Project specifications:

- Motion detection using MEMS sensors (accelerometer, gyroscope, magnetometer);
- Click detection using switches and/or buttons;
- Wireless communication between the device and the PC/smartphone/tablet using a HID-enabled Bluetooth module class 2 (since most of the PCs are compatible with class 2 only at the moment);
- Battery power supplied;
- Sleep modes to increase battery life;
- Rechargeable battery circuit;
- On/Off switch;
- LED to indicate the battery recharging and the power-on of the system with a fast blink;

- As small as possible PCB dimensions (keeping into account the battery size and the handling capability).

Design flow:

- Focus on the main features of the project, its feasibility and state-of-the-art of the available components;
- Title the project with a meaningful and simple to remember name;
- Project proposal, listing the required and the desired specifications;
- Components search and purchase of development and evaluation boards;
- Prototyping and firmware development using a STM32Nucleo development board with STM32L476RG (ultra-low-power) MCU, a STmicroelectronics X-NUCLEO-IKS01A1 MEMS expansion board, Microchip RN42 Bluetooth 2.0 module and using Mbed as development environment;
- Design of circuit schematic using Cadence ORCAD Capture;
- Netlist creation, PCB layout and Gerber files creation using Cadence Allegro;
- Video making of the circuit design using the Cadence softwares, to be published on the Artedas S.r.l Youtube channel;
- Report writing.

Chapter 2:
System-level description

Block diagram

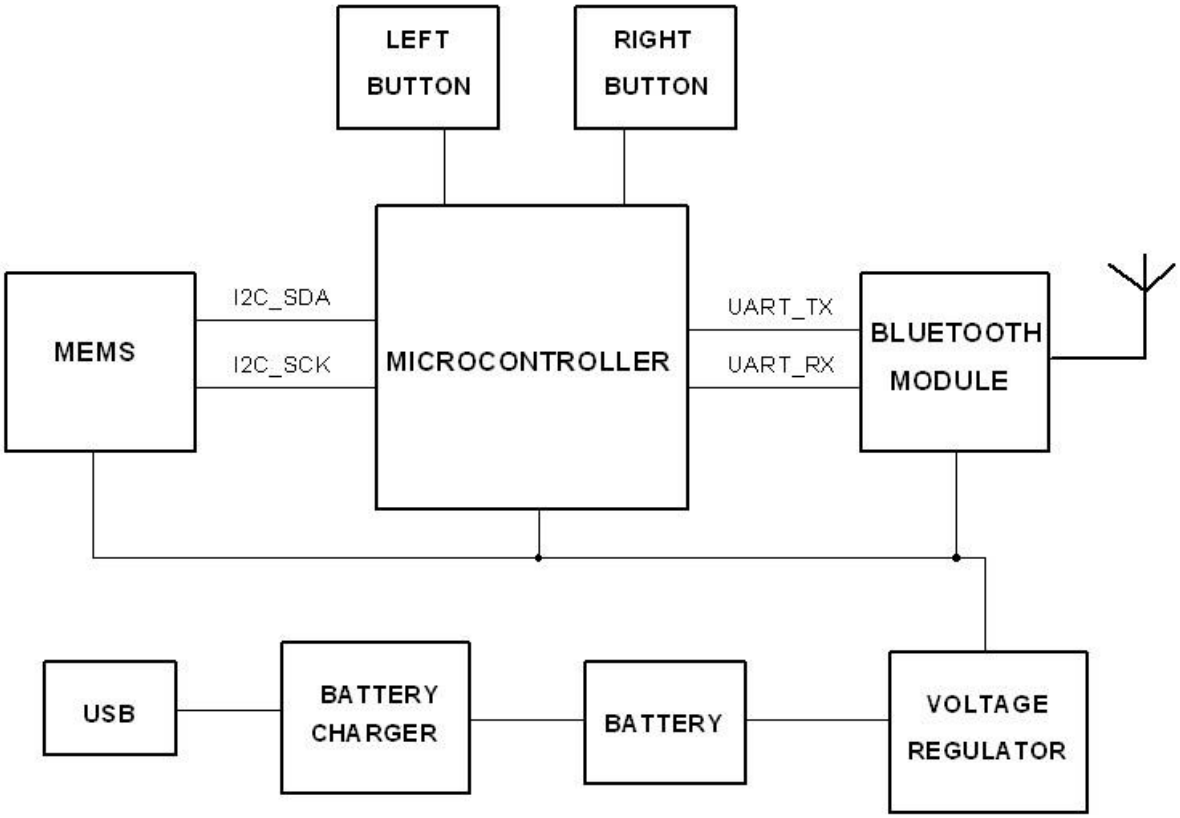


Figure 1: Block diagram of the entire circuit.

System structure

All the system parts are battery powered from a 3.7 V Li-ion rechargeable battery; a buck/boost voltage regulator (MAX1759) is needed to supply the system with a stable voltage equal to 3.3 V. It is good since the μC, the Bluetooth module and the MEMS should be supplied at this voltage level. The battery can be recharged through a micro-USB connector and a battery-charger IC (MAX1551). The *Flying mouse* works in the following way: the user moves it and the LSM6DS0 MEMS IC uses its accelerometer and gyroscope to detect its movements; the accelerometer and gyroscope sampled values are stored in the MEMS internal registers and are transmitted using the I2C

serial protocol to the STM32L476RG microcontroller. The LSM6DS0 is a slave in the I2C bus; the I2C_SDA line is bidirectional and can be used to write data from the master μC to the MEMS or to read the MEMS registers. The μC usually is in sleep mode; it wakes up at regular time periods, acquires the MEMS values using the I2C and checks if the buttons have been pressed, elaborates them (see *Firmware development* section for further details) and transmits the computed values to the RN42-HID Bluetooth module using the UART serial interface; no control features (CTS and RTS) are used but only TX and RX signals. Then the μC re-enters in sleep mode. The Bluetooth module, that has an embedded antenna, transmits the received data to the connected PC/smartphone/tablet. A feature to save power has been implemented: if the MEMS values are stable for more than 5 seconds the μC enters in deep sleep mode to save power and it wakes up only when a button is pressed. The microcontroller acts as a master in both the serial communication protocols, it requests the data from the MEMS driving the clock of the I2C interface and sends data to the Bluetooth module at a baud rate equal to 115200 kbps. Furthermore, between the battery and the voltage regulator there is a switch that allows the user to switch off the mouse, disconnecting the battery from the other parts of the circuit.

Chapter 3: Components

In this chapter the selected components are described; the components features are explained as well as the reason why they have been chosen. However, although there are a lot of similar products available on the market, the following criteria has been used: the suggestion from the Professor Pasero was to use a STMmicroelectronics STM32 Nucleo-64 evaluation board (see Prototype and Verification chapter) , and so the choice of the microcontroller was restricted to the available μC sold with the Nucleo boards. Another suggestion was to use the X-NUCLEO-IKS01A1, a MEMS inertial and environmental sensor evaluation board from STmicroelectronics (from this one derived the choice of the inertial MEMS sensor LSM6DS0).

STM32L476RG microcontroller

The STM32L476RG is a 32-bit microcontroller from STmicroelectronics, based on an Ultra-low-power ARM Cortex-M4 32-bit RISC core operating at a frequency of up to 80 MHz. It is the most important component in the system since it performs all the computations and the signal acquirements. It has been chosen among all the others STmicroelectronics microcontrollers because it embeds the most performant ARM core of the Ultra-low-power series, ensuring high performances at a very low power consumption. Anyway, as can be seen below, it is a very powerful μC, with a lot of different characteristics, that are not well exploited in this project; the choice was done in order to have a powerful project core, that allows to develop some additional features in the future (see *Market analysis and future developments* for further details).

Figure 2: Panorama of ARM cortex cores, courtesy of STmicroelectronics.

Main features

Some of the many features of this microcontroller are the following:

- 1.71 V to 3.6 V power supply
- 128 Kbyte of SRAM and 1 Mbyte of Flash memory, with several protection mechanisms for embedded Flash memory and SRAM: readout protection, write protection, proprietary code readout protection and Firewall
- Three fast 12-bit ADCs (5 Msps) with 16 channels, two comparators, two operational amplifiers, two 12-bit DAC channels, an internal voltage reference buffer, a low-power RTC, two general-purpose 32-bit timer, two 16-bit PWM timers dedicated to motor control, seven general-purpose 16-bit timers, and two 16-bit low-power timers
- Core: ARM® 32-bit Cortex-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS/1.25DMIPS/MHz (Dhrystone 2.1), and DSP instructions
- 12 capacitive sensing channels
- Many communication interfaces: three I2Cs, three SPIs, three USARTs, two UARTs and one Low-Power UART, two SAIs (Serial Audio Interfaces), one

CAN, one USB OTG full-speed and one SWPMI (Single Wire Protocol Master Interface)

- Operates in the -40 to +85 °C (+105 °C junction), -40 to +105 °C (+125 °C junction) and -40 to +125 °C (+130 °C junction) temperature ranges from a 1.71 to 3.6 V power supply
- True random number generator
- Two analog comparators and two operational amplifiers
- Output current sunk and sourced by any I/O and control pin equal to 20 mA
- Total output current sunk and sourced by sum of all I/Os and control pins equal to 100 mA.

Clock sources

- **System clock sources**: four different clock sources can be used to drive the master clock SYSCLK:
  - 4-48 MHz high-speed external crystal or ceramic resonator (HSE), that can supply a PLL. The HSE can also be configured in bypass mode for an external clock.
  - 16 MHz high-speed internal RC oscillator (HSI16), trimmable by software, that can supply a PLL.
  - Multispeed internal RC oscillator (MSI), trimmable by software, able to generate 12 frequencies from 100 kHz to 48 MHz. When a 32.768 kHz clock source is available in the system (LSE), the MSI frequency can be automatically trimmed by hardware to reach better than ± 0.25% accuracy. In this mode the MSI can feed the USB device, saving the need of an external high-speed crystal (HSE). The MSI can supply a PLL.
  - System PLL which can be fed by HSE, HSI16 or MSI, with a maximum frequency at 80 MHz.
- **Peripheral clock sources**: Several peripherals (USB, SDMMC, RNG, SAI, USARTs, I2Cs, LPTimers, ADC, SWPMI) have their own independent clock whatever the system clock. Three PLLs, each having three independent outputs

allowing the highest flexibility, can generate independent clocks for the ADC, the USB/SDMMC/RNG and the two SAIs.

- **Startup clock**: after reset, the microcontroller restarts by default with an internal 4 MHz clock (MSI). The prescaler ratio and clock source can be changed by the application program as soon as the code execution starts.

Power modes

The STM32L476RG microcontroller is built around a Cortex®-M4 with FPU and DSP instruction set. The high processing performance in Run mode (expressed in DMIPS/MHz) is achieved thanks to the use of a Cortex-M4 core associated with the interfaces of its memories. In order to allow full performance operation up to 80 MHz, the STM32L4 embeds the ART Accelerator™ which masks the Flash memory access wait state, and allows to achieve 1.25 DMIPS/MHz whatever the system clock frequency. The high energy efficiency, expressed as mA/DMIPS, is obtained by adapting dynamically the internal supply voltage to the operating frequency. This method is called "*undervolting*".

STM32L476RG microcontroller offers two dynamically selectable voltages and frequency ranges: Range 1 for System frequency up to 80 MHz and Range 2 for System frequency up to 26 MHz with improved efficiency (up to 15% higher than Range 1). A dedicated Low-power run mode (LPRun) allows to execute at up to 2 MHz. This is achieved by supplying the logic with the internal low-power regulator.

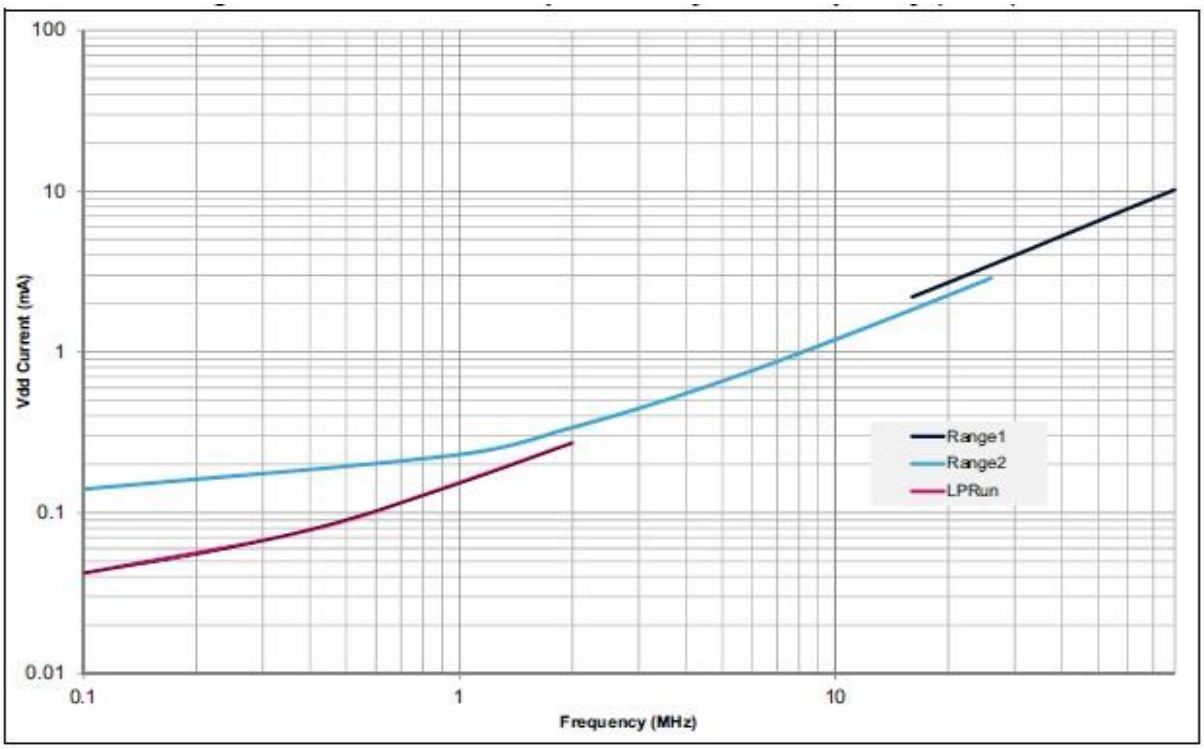


Figure 3: Current consumption vs. system frequency (25°C), courtesy of STmicroelectronics.

Many different power modes are supported, seven of them are low-power. In the following table have been reported only the power modes used in the project (See datasheet of STM32L476xx family for further details) :

MODE	CPU	FLASH	SRAM	CLOCKS	WAKEUP SOURCES	CONSUMPTION	WAKEUP TIME
Run	Yes	On	On	Any	N/A	110 µA/MHz	N/A
Sleep	No	On	On	Any	Any interrupt or event	36 µA/MHz	6 cycles
Stop 1	No	Off	On	LSE LSI	Reset pin, all I/Os, BOR, RTC, Comparators, USARTs, I2Cs and others[1]	6.6 µA w/o RTC 6.9 µA w RTC	4 µs in SRAM 6 µs in Flash

Table 1: STM32l476RG power modes exploited in the project.

[1] See datasheet of STM32L476xx family for further details.

The Standby and the Shutdown modes have not been used in order to not lose the SRAMs content.

Package

The STM32L476RG is sold in a LPQF64 package, that means Low-Profile Quad Flat package with 64 pins and its dimensions are 10x10x1.4 mm and has gull-wing SMT leads.

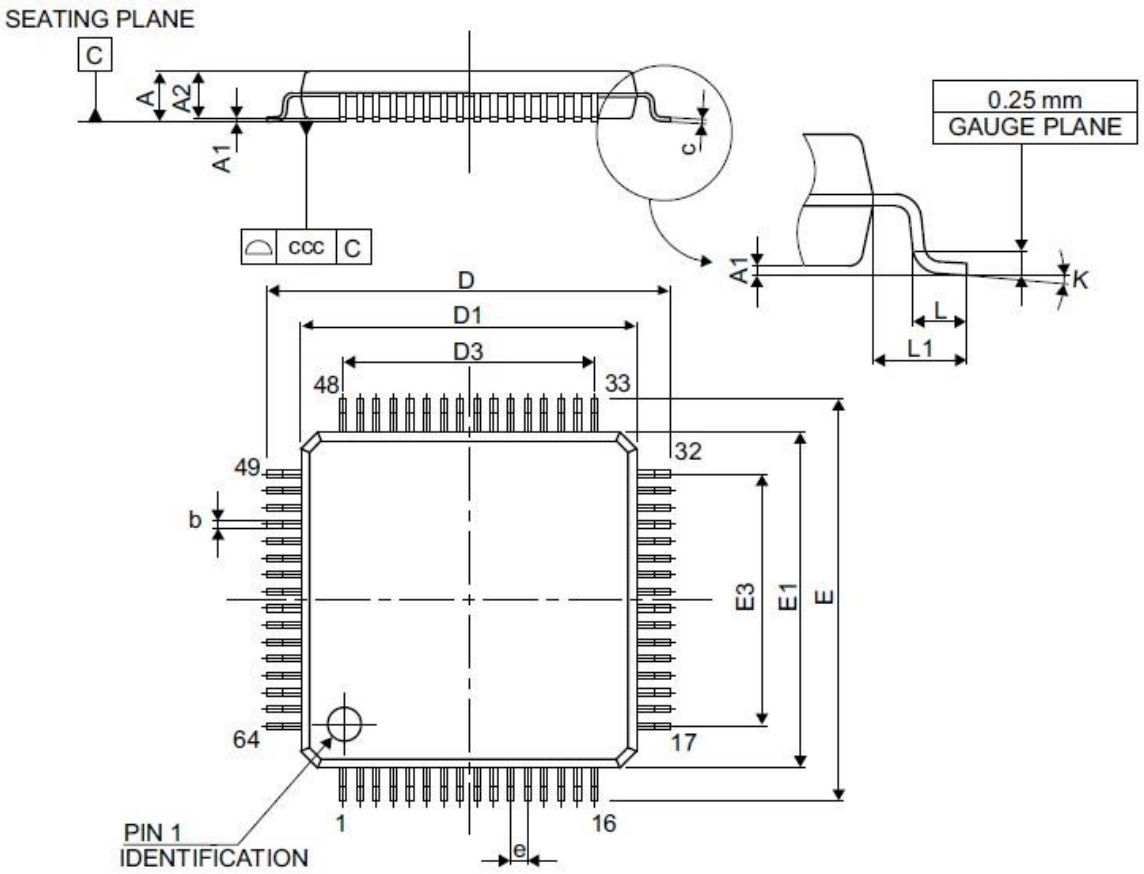


Figure 4: LPQF64 low-profile quad flat package outline.

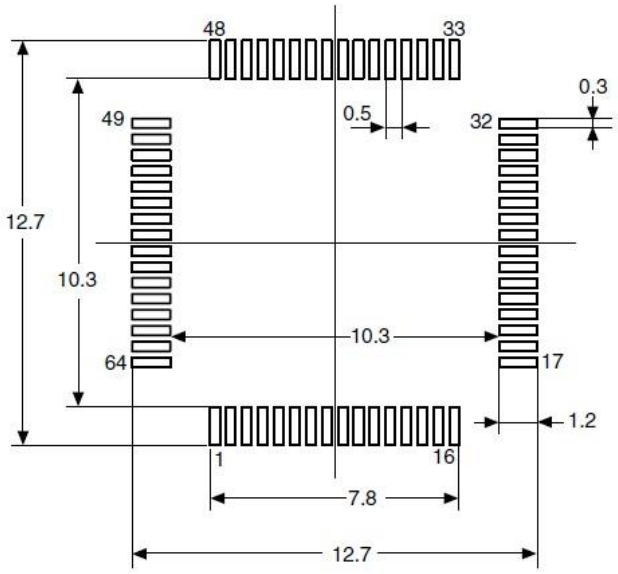


Figure 5: STM32L476RG recommended footprint (dimensions are expressed in millimiters).

LSM6DS0 MEMS inertial module

The LSM6DS0 is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope and has a full-scale acceleration range of ±2/±4/±8/±16 *g* and an angular rate range of ±245/±500/±2000 dps. Since this module integrates complementary types of sensors it is more compact, robust and easy-to-assemble compared to discrete MEMS products; it is also the reason why it has been chosen. On the other hand the reasons why only one MEMS has been used (e.g. a magnetometer wasn't included in the project) are described in the *Software flow* section. The LSM6DS0 is the worship of this project since it allows to keepsa reduce size of the entire design, without giving up anything in terms of measurement precision.

Main features

- Analog supply voltage: 1.71 V to 3.6 V
- "Always on" eco power mode down to 1.8 mA
- 3 independent acceleration channels and 3 angular rate channels ±2/±4/±8/±16 *g* full scale and ±245/±500/±2000 dps full scale
- SPI/I2C serial interface
- Embedded temperature sensor

- Temperature range: -40 to +85 °C
- Embedded FIFO

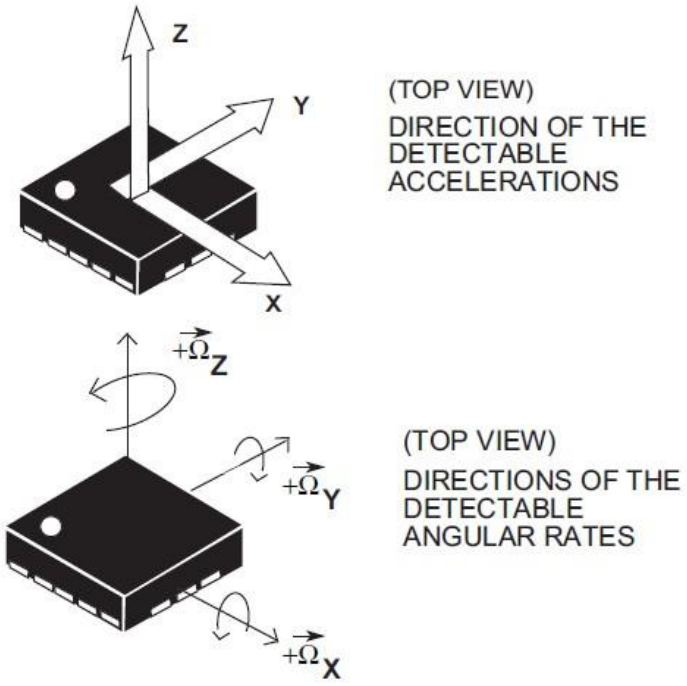


Figure 6: Panorama of the 3D accelerometer and 3D gyroscope capabilities.

I^2C interface

The I2C interface has been chosen instead of the SPI one because it was the interface used on the X-NUCLEO-IKS01A1 sensor evaluation board, used during prototyping. The LSM6DS0 I2C is a bus slave. The I2C is employed to write the data to the registers whose content can also be read back. There are two signals associated with the I2C bus: the serial clock line (SCL) and the Serial Data line (SDA). The latter is a bidirectional line used for sending and receiving the data to/from the interface. Both the lines must be connected to VDD through an external pull-up resistor. When the bus is free, both the lines are high.

The transaction on the bus is started through a START (ST) signal. A START condition is defined as a high-to-low transition on the data line while the SCL line is held high. After this has been transmitted by the master, the bus is considered busy. The next byte of data transmitted after the start condition contains the address of the slave in the first 7 bits and the eighth bit tells whether the master is receiving data from the slave or transmitting data to the slave. When an address is sent, each device in the system compares the first seven bits after a start condition with its address. If

they match, the device considers itself addressed by the master. The Slave ADdress (SAD) associated to the LSM6DS0 is 110101xb. The SDO/SA0 pin can be used to modify the less significant bit of the device address. If the SDO/SA0 pin is connected to a voltage supply, LSb is '1' (address 1101011b), else if the SDO/SA0 pin is connected to ground, the LSb value is '0' (address 1101010b). This solution permits to connect and address two different inertial modules to the same I2C bus. Following the structure implemented in the evaluation board (since the firmware was working on it), the LSb has been set to '1'.

Data transfer with acknowledge is mandatory. The transmitter must release the SDA line during the acknowledge pulse. The receiver must then pull the data line low so that it remains stable low during the high period of the acknowledge clock pulse. A receiver which has been addressed is obliged to generate an acknowledge after each byte of data received. Each data transfer contains 8 bits. The number of bytes transferred per transfer is unlimited. Data is transferred with the Most Significant bit (MSb) first. If a receiver can't receive another complete byte of data until it has performed some other function, it can hold the clock line, SCL low to force the transmitter into a wait state. Data transfer only continues when the receiver is ready for another byte and releases the data line.

Package

The LSM6DS0 is available in a LGA-16L package, that means plastic land grid array package with 16 pins. Its dimensions are 3x3x0.86 millimeters.

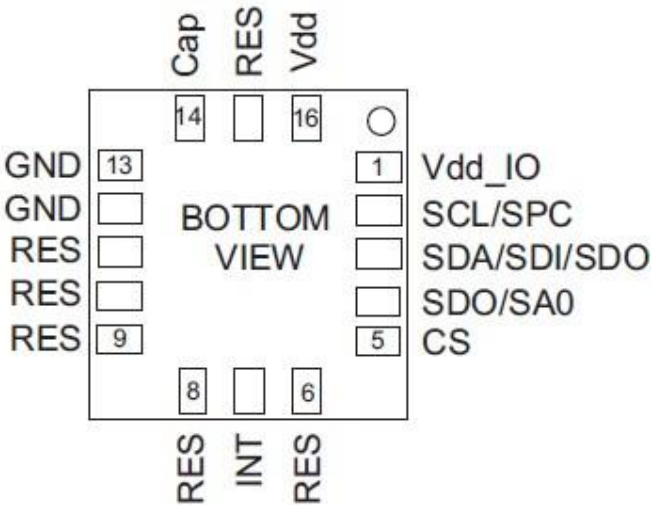


Figure 7: LSM6DS0 pin description.

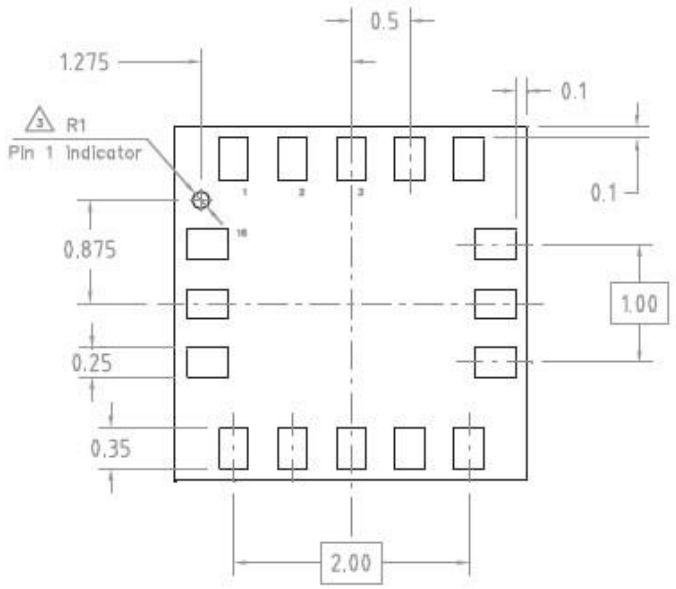


Figure 8: LGA 3x3x0.86 16L package outline and dimensions.

RN42-HID Bluetooth module

The Microchip RN42 module is a small form factor, low-power, class 2 Bluetooth module with an high-performance, PCB trace antenna. This module supports a variety of Bluetooth profiles, including Human Interface Device (HID), Serial Port Profile (SPP), DUN, HCI, and iAP for use with iPad, iPod and iPhone devices. The majority of personal computer nowadays have Bluetooth class 2 compatibility only and for this reason a class 2 module had been preferred over a class 4 BLE module; on the other hand RN42 is quite expensive compared to the other class 2 Bluetooth modules on the market (the most famous are the very cheap HC-05 and the HC-06 modules). It has been chosen among all the others because it supports the HID profile (see *HID* section below); this choice allowed to do a fast prototyping (we can describe it as a plug-and-play module) and to provide compatibility with all the devices that support a mouse (PCs, smartphones, tablets) without the need of a target driver installed in the host.

Main features

- version 2.1 + Enhanced Data Rate (EDR) supported
- ASCII command interface over UART
- Embedded Bluetooth stack profiles include: GAP, SDP, RFCOMM, L2CAP protocols, with SPP, HID, and DUN profile support (does not require any host stack)

- Temperature Range: -40 to +85 °C
- Supply Voltage (DC): 3.3V
- Compliance (RN42): European R&TTE Directive Assessed Radio Module
- Integrated crystal, internal voltage regulator, matching circuitry, power amplifier, low noise, memory amplifier and PCB trace antenna
- Up to 10 meters range.

Operating modes

The RN42 module is managed through ASCII commands via the UART and/or PIO signals. A microcontroller unit (MCU) or host processor sends commands to configure module features, read status, and manage Bluetooth data connections; during the project life time the STM32 MCU does this job, while during the project development all the module's settings had been made using a host PC. The Bluetooth module operates in two modes: data mode (default) and command mode. While in data mode, the module operates as a data pipe. When the module receives data, it strips the Bluetooth headers and trailers and passes the user data to the UART port. When data is written to the UART port, the module constructs the Bluetooth packet and sends it out over the Bluetooth wireless connection. Thus, the entire process of sending/receiving data to the host is transparent to the end microprocessor.

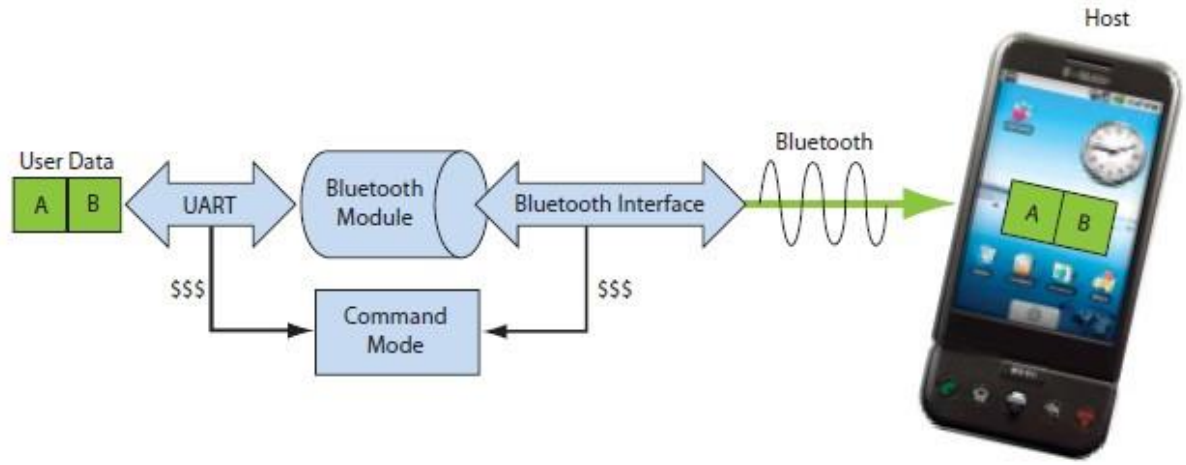


Figure 9: data mode acts as a transparent pipe; using three $ symbols the command mode is entered.

The command mode has been used initially to set the module's name, the transmission power (lowered with respect to the default one in order to consume less power) and to set the module as a mouse device that uses the HID profile; furthermore the pin code

of the module has been disabled in order to make the *flying mouse* connectable to every master device without should uncomfortably insert the pin code at every connection. The baud rate had been left as the default value, equal to 115200 kbps, as well as the UART , set to 8 bits of data packet, no parity bit and 1 stop bit. No flow control had been used.

A connection is established in the following way:

- *Discovery*: In the discovery phase, the Bluetooth module broadcasts its name, profile support, and MAC address. It is ready for other devices to pair with it.
- *Pairing*: During pairing, the Bluetooth module and the Bluetooth master validate the pin code. If the pin code validates successfully, they exchange security keys and a channel hopping pseudo-random sequence. Successful pairing results in the module and master establishing link keys.
- *Connecting*: Before connecting, the Bluetooth devices must have paired successfully. The master initiates a connection, the master and slave validate the link keys, and a Bluetooth link is established.

HID: Human Interface Device

The important characteristic of the RN42 module is its firmware, that is the version 6.11, that includes the HID profile. The HID standard was adopted primarily to enable innovation in PC input devices and to simplify the process of installing such devices. HID-defined devices deliver self-describing packages that may contain any number of data types and formats. A single HID driver on a computer parses data and enables dynamic association of data I/O with application functionality, which has enabled rapid innovation and development of new human-interface devices. HID communications happen between a device and a host, where the latter is usually a PC or a smartphone. Devices define their data packets and then present a HID descriptor to the host. The HID descriptor is a hard coded array of bytes that describe the device's data packets. This includes: how many packets the device supports, the size of the packets, and the purpose of each byte and bit in the packet. This standard was born for the USB protocol, but today there is also the Bluetooth-HID standard.

The RN42 module is recognized from the host as a mouse, thanks to the configuration settings (that should be done only once). To send data packets from the device to the host the Raw Report mode of the RN42 had been used; the Raw Report template is reported below. The start byte is stripped and the following bytes are sent without interpretation; HID reports are sent one report at a time. The format of the data bytes depends on the descriptor type.

START BYTE	LENGTH BYTE	DESCRIPTOR BYTE	DATA BYTES

Figure 10: Raw Report consists of a start byte, length byte, descriptor type (which defines the type of HID device), and data.

The Start byte was set to 0xFD, that denotes that the packet is a Raw Report, the Length byte was set to 0x05 since there are five bytes after it: the descriptor byte and four data byte. The Descriptor byte was set to 0x02, that is the mouse descriptor in the HID profile. Finally four data bytes had been used, one for the right and left buttons, one for the horizontal movement and one for the vertical movement. The last byte had been implemented for a future development of the wheel scroll (not implemented in this project to maintain the design as small as possible).

Power modes

The main issue of this module is its power modes; in fact when it is active it consumes much more power than a class 4 BLE module, but we need a class 2 Bluetooth module for laptops compatibility. On the other hand the RN42 does not have a good sleep mode: the goal of the project was to obtain a fluid hand movement tracking and so the Sniff mode was not used. In fact in Sniff mode, the radio wakes up at specific intervals and sleeps in very low power mode (around 2 mA) the rest of the time; in the project the sensors' data are sampled and transmitted every 5 milliseconds and so the module would take more time to enter and exit from Sniff mode than the time it would remain in and this fact can lead to an higher power consumption than the normal mode. And so the only remaining way to save power was to set the module to the Deep Sleep mode; the problem is that it can enter the Deep Sleep mode only when it is disconnected and it is very uncomfortable from the user point of view. So the choice was to left the module always connected. In table 2 are listed the power consumptions for the various operating modes.

Operating mode	Typical power consumption	Units
Radio On (Discovery)	40	mA
Connected Idle (No Sniff)	25	mA
Connected Idle (Sniff 100 ms)	12	mA
Connected with data transfer	40 ÷ 50	mA
Deep Sleep Idle Mode	26	μA

Table 2: RN42 power consumption in the different operating modes.

Package

The RN42 module is sold in a SMD package of size 13.4 x 25.8 x 2.4 mm.

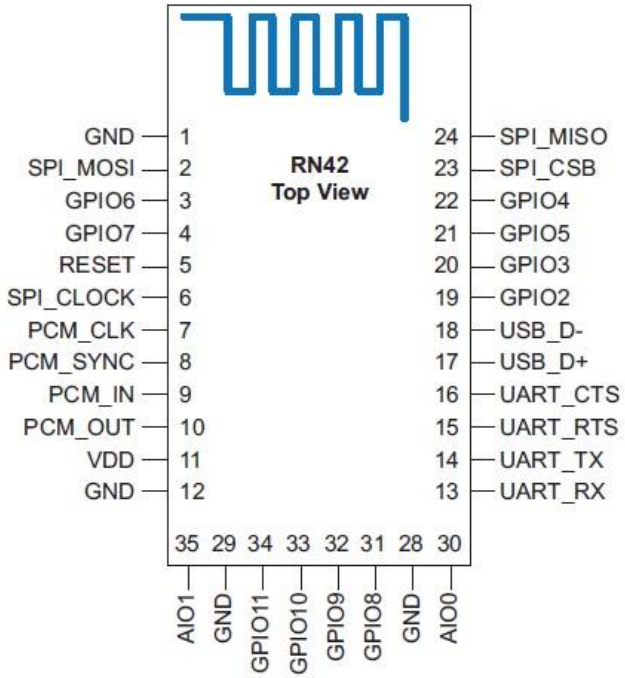


Figure 11: RN42 pin description.

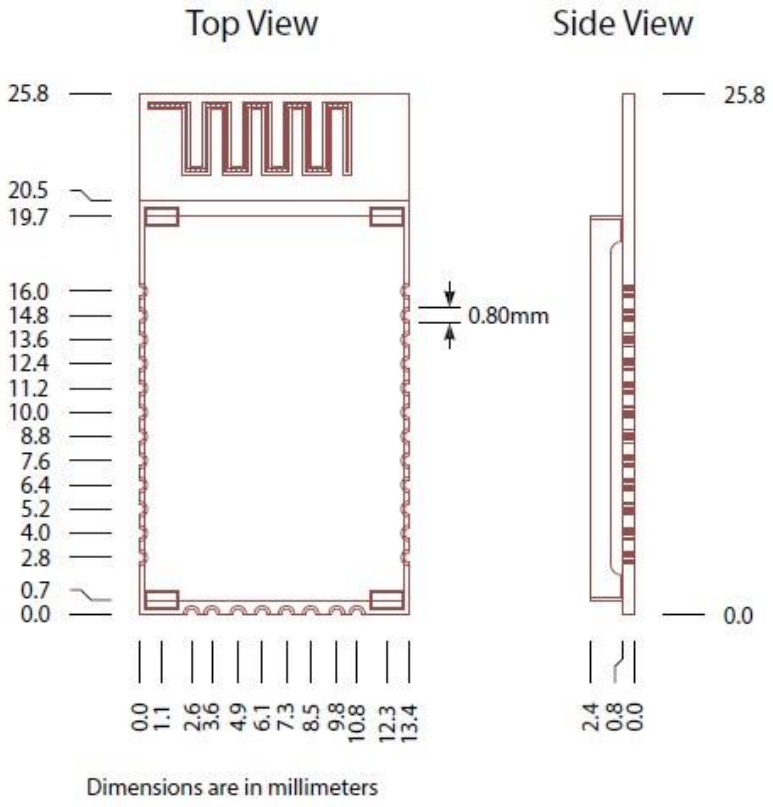


Figure 12: RN42 recommended footprint and dimensions.

MAX1759 Buck/boost voltage regulator

The MAX1759 is a buck/boost regulating charge pump that generates a regulated output voltage from a single lithium-ion (Li+) cell, or two or three NiMH or alkaline cells; the buck/boost charge-pump architecture allows the input voltage to be higher or lower than the regulated output voltage. The MAX1759 is guaranteed to deliver a regulated 3.3V at 100mA continuous, from a +2.5V input. Peaks up to 200mA are acceptable as long as the current is ≤ 100mA (RMS).

Following the design guidelines illustrated in the datasheet the FB pin had been connected to GND, in order to obtain a regulated 3.3V output; furthermore, to achieve the minimum output voltage ripple, ceramic capacitors with the highest suggested value had been chosen.

Main features

- Regulated Output Voltage (Fixed 3.3V or Adjustable 2.5V to 5.5V)
- 100mA Guaranteed Output Current
- +1.6V to +5.5V Input Voltage Range
- Low 50 μA Quiescent Supply Current
- 1 μA Shutdown Mode
- Load Disconnected from Input in Shutdown
- Short-Circuit Protection and Thermal Shutdown
- Temperature Range: -40 to +85 °C

Package

The MAX1759 is available in a small 10 pins μMAX package and its dimensions are 3x3x1.1 millimeters (E1 is equal to 3 mm, while E is equal to 5 mm).

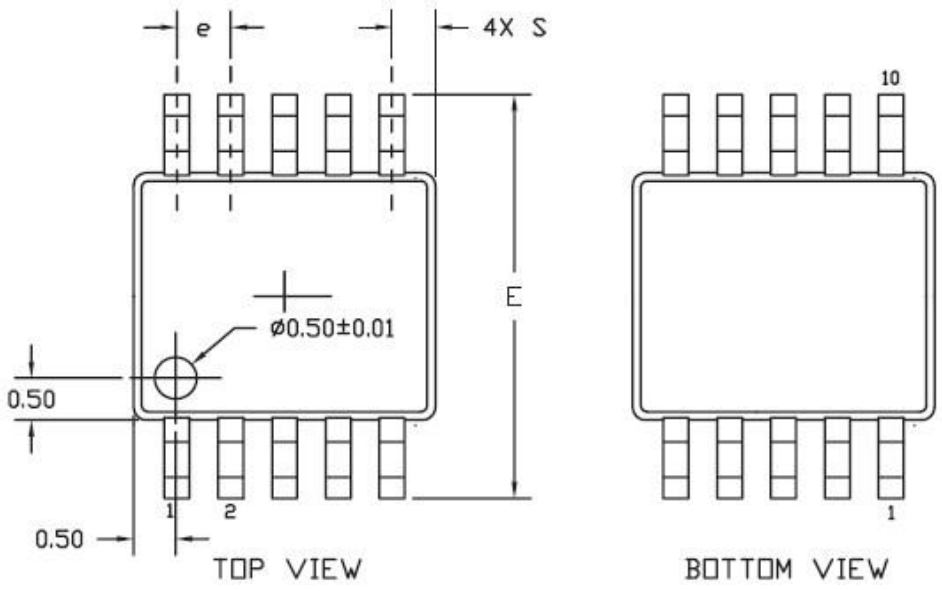


Figure 13: Top and bottom view of the µMAX package.

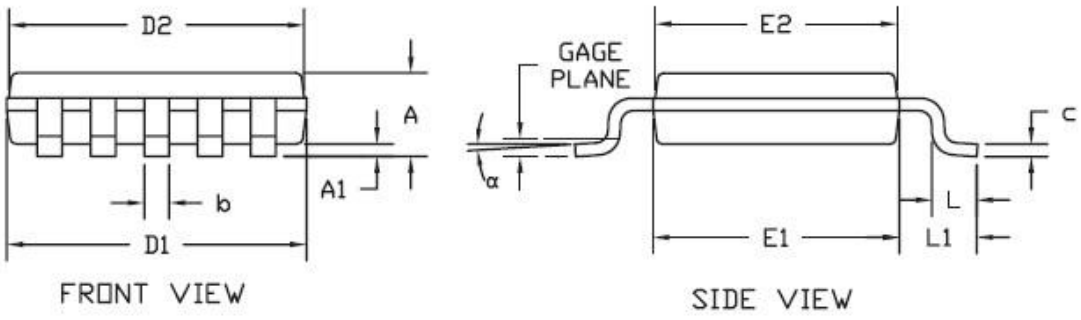


Figure 14: Front and side view of the µMAX package.

MAX1551 Battery charger

The MAX1551 can charge a single-cell Li+ battery from both USB and AC adapter sources; with USB connected, but without DC power, the charge current is set to 100mA. This device operates with no external FETs or diodes, and accepts operating input voltages up to 7V. When input power is removed, battery leakage current is less than 5µA, no input-blocking diodes are required to prevent battery drain and when the MAX1551 thermal limit is reached, the charger does not shut down, but simply reduces charging current. Finally The MAX1551 features a pre-charge current to protect deeply discharged cells. If VBAT is less than 3V, the device enters pre-charge mode where charging current is limited to 40mA.

Main features

- Charge from USB or AC Adapter
- On-Chip Thermal Limiting
- Charge Status Indicator
- Temperature Range: -40 to +85 °C

Package

The MAX1551 is available in a 5 pins TSOP23 package, that means Thin Small Outline package and its dimensions are 1.6x3.0x1.1 mm.

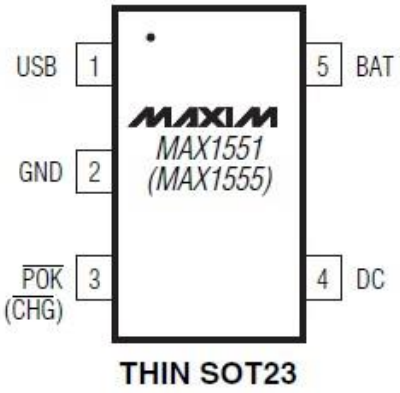


Figure 15: MAX1551 pin description.

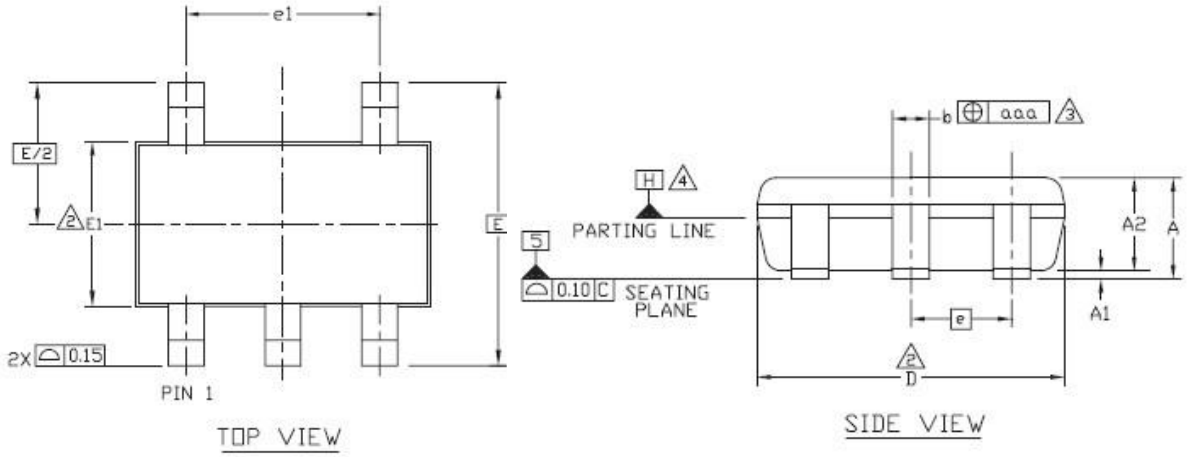


Figure 16: MAX1551 TSOP23 package top and bottom view.

Chapter 4:
Design and firmware choices

At the end of the components selection, the firmware was developed. After set the basic configuration of the Bluetooth module, a complimentary sensor fusion algorithm had been used and calibrated in order to track the user hand motion; finally the press detection of the left and right buttons had been implemented in firmware.

Sensor fusion algorithm

For this kind of applications a sensor fusion algorithm usually is implemented; in fact three rotational movements can be measured: the yaw, which is the Z-axis rotation, the pitch, which is Y-axis rotation and the roll, which is X-axis rotation. A typical inertial measurement system that implements a sensor fusion algorithm is made with an accelerometer, a gyroscope, and a magnetometer. Anyway a good algorithm that allows to use only an accelerometer and a gyroscope had been found and it allows to save money and space on the final printed circuit board. The characteristic of the used sensors are:

- *Accelerometer*: an accelerometer measures proper acceleration, which is the acceleration it experiences relative to freefall and is the acceleration felt by people and objects. An accelerometer at rest relative to the Earth's surface will indicate approximately 1 g upwards, because any point on the Earth's surface is accelerating upwards relative to the local inertial frame (the frame of a freely falling object near the surface). The angle between an axis of the device and the horizontal plane can be also computed: it is called *Tilt angle* and small angle approximation and the horizontal axis to save processor time and coding complexity. So an accelerometer provides a reference vector, gravity, that does not drift (long-term), but has poor transient response.

- *Gyroscope*: A gyroscope measures angular velocity, or how fast an object is spinning about an axis and angles can be computed by numerical integration. Gyroscopes are excellent for computing orientation (short-term), can give fast transient response and are not affected by gravity, so they can complement accelerometers. However they integrate effects over time of a slowly varying offset and noise; this drift must be eliminated, and it requires an external reference vector that does not drift.

So gyroscopes tend to drift as time increases, imitating low-pass filter characteristics. Accelerometers tend to be inaccurate at short time intervals, imitating a high-pass filter. Individually, the two sensors are too inaccurate to determine angles, but together, they can provide accurate angles of rotation, without the need of an additional magnetometer.

Since a mouse needs only two directions, the yaw and the pitch angles had been measured; to determine the yaw only the gyroscope had been used, since the accelerometer is not sensible to the rotational movements around the Z-axis. However in this way the accelerometer cannot take care of the gyroscope drift; for this reason a rounding factor had been used, in order to neglect the least significant bits of the gyroscope value, that were involved in the drift. On the other hand to compute the pitch angle an algorithm proposed by Shane Colton, a professor of the MIT (Massachusetts Institute of Technology) had been implemented; this algorithm uses both the accelerometer and the gyroscope. So to compute the pitch, the first step is to implement a Low-Pass Filter on the accelerometer sampled value; it means let pass through only long-term changes, filtering out short-term fluctuations. One way to do this is to force the changes to build up little by little in subsequent times through the program loop, with constants K_1 and K_2 to set the desired time constant of the filter :

$$angle = K_1 \cdot angle + K_2 \cdot acc\_value$$

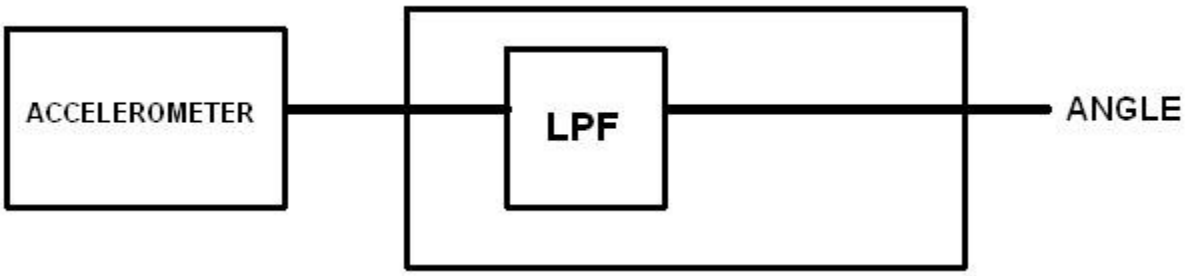


Figure 17: LPF to filter out the high frequency noise of the accelerometer.

The second step is to implement a numerical integrator on the gyroscope sampled value, in fact in order to compute the actual angle (Δt is the time interval between samples) the old position should be taken and added to the change in position:

$$angle = angle + gyro\_value \cdot \Delta t$$

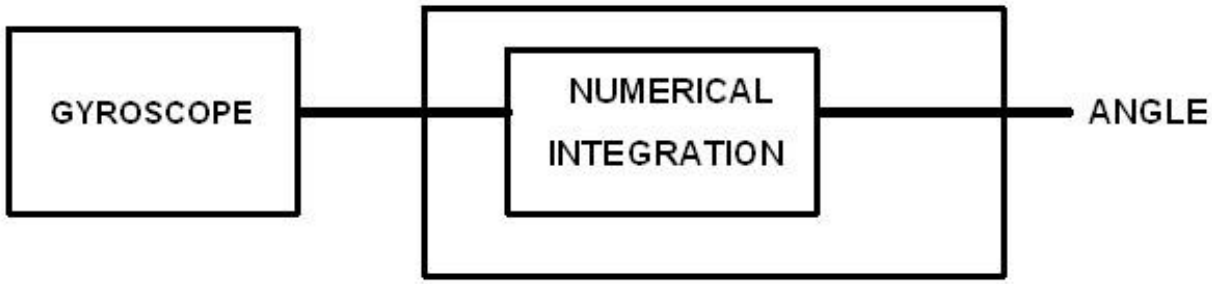


Figure 18: Numerical integrator to compute the actual angle.

The last step, that implements the effective complementary sensor fusion algorithm proposed by Colton is to add a High-Pass Filter on the gyroscope output value (to set the circuit response more than to filter) and to add the gyroscope values with the accelerometer ones. The constant K_{LPF} and K_{HPF} are greater than zero and their sum is equal to 1.

$$angle = K_{HPF} \cdot (angle + gyro\_value \cdot \Delta t) + K_{LPF} \cdot acc\_value$$

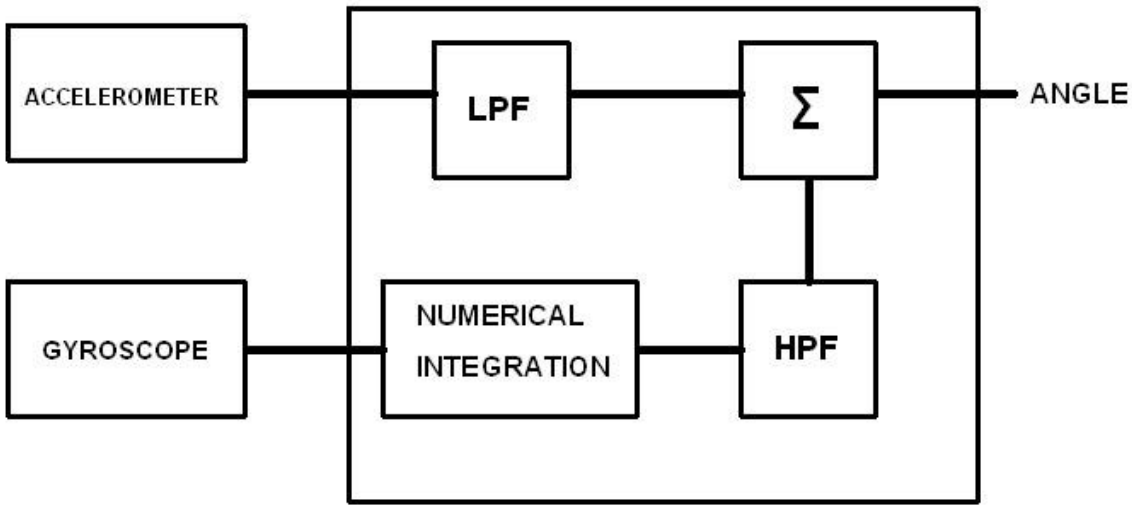


Figure 19: Sensor fusion algorithm used to compute the pitch angle.

Offset computation and elimination

When the motion detection algorithm was implemented and tested, the following goal was to make the pointer fixed and stable on the screen while the mouse is not moving. To obtain this, the offset of the accelerometer and the gyroscope had to be computed in order to be eliminated. So the microcontroller was programmed such that it reads the sensors every 5 milliseconds and then it adds these values to the previous sampled

values; this procedure was done in order to perform an average every 100 thousand samples.

```
While(1) {
if(timer_irq_triggered) {
    accelerometer->Get_X_Axes(axes);
    actmpoffset[0] += axes[0];
    actmpoffset[1] += axes[1];
    actmpoffset[2] += axes[2];

    gyroscope->Get_G_Axes(axes);
    gyrtmpoffset[0] += axes[0];
    gyrtmpoffset[1] += axes[1];
    gyrtmpoffset[2] += axes[2];

    j++;
   if(j== 100000){
  offsetacc[0]= actmpoffset[0]/j;
  offsetacc[1]= actmpoffset[1]/j;
  offsetacc[2]= actmpoffset[2]/j;

  offsetgyr[0]= gyrtmpoffset[0]/j;
  offsetgyr[1]= gyrtmpoffset[1]/j;
  offsetgyr[2]= gyrtmpoffset[2]/j;

pc.printf("LSM6DS0 [acc/mg] offset: %6ld, %6ld, %6ld\r\n", offsetacc[0], offsetacc[1], offsetacc[2]);
pc.printf("LSM6DS0 [gyro/mdps] offset: %6ld, %6ld, %6ld\r\n", offsetgyr[0], offsetgyr[1], offsetgyr[2]);
        j=0;
     actmpoffset[0] =0;
     actmpoffset[1] =0;
     actmpoffset[2] =0;
     gyrtmpoffset[0] =0;
```

```
    gyrtmpoffset[1] =0;
    gyrtmpoffset[2] =0;
  } } }
```

Table 3: Pseudo-code used for the sensors offset computation.

This procedure was iterated 16 times (there were $16 \cdot 100000 = 1600000$ sampled values for each axis of each sensor) and the average of the 16 averages was made (reported in table 4).

```
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     97,  -2425,   1197
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     89,  -2446,   1208
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     67,  -2433,   1212
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     40,  -2425,   1200
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:     37,  -2410,   1189
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     38,  -2422,   1203
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:     58,  -2464,   1228
LSM6DSO [acc/mg] offset:    -15,     14,    974
LSM6DSO [gyro/mdps] offset:     54,  -2461,   1224
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:     32,  -2450,   1207
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:    -17,  -2445,   1190
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:    -10,  -2480,   1197
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:    -16,  -2469,   1194
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:    -19,  -2455,   1182
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:      4,  -2473,   1205
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:      7,  -2490,   1227
LSM6DSO [acc/mg] offset:    -15,     15,    974
LSM6DSO [gyro/mdps] offset:     11,  -2486,   1238
```

Figure 20: The sixteen averages for each axis of both the accelerometer and the gyroscope.

Sensor	**X-axis**	**Y-axis**	**Z-axis**
Accelerometer	-15	15	974
Gyroscope	30	-2452	1206

Table 4: Average of the averages of the sensors offset.

As can be seen in Table 4 the offset along the three axis of the accelerometer are very small, since it is -15 and 15 and about 7 for the Z-axis (981 - 974). The gyroscope instead has higher offset values for the Y-axis and the Z-axis. Since in the project only the accelerometer Y-axis value and the gyroscope X-axis (yaw) and Z-axis (pitch) are used, only the significant offset value had been considered while the others had been neglected. Precisely only the Z-axis gyroscope offset value, equal to 1206 was considered.

By subtracting this value from the pitch value the pointer of the mouse is able to remain stable and to not move if the mouse is not moving. The implemented formulas are the following:

$$pitch_{angle} = K_{HPF} \cdot \left(pitch_{angle} + gyro_{X_{axis}} \cdot \Delta t\right) + K_{LPF} \cdot acc\_Y_{axis}$$

$$yaw_{angle} = -\left(gyro\_Z_{axis} - gyro\_Z_{offset}\right)$$

Then both the angles have been multiplicated for a sensitivity constant, in order to obtain the desired direct link between the hand movement and the speed of the pointer on the host screen.

$$pitch_{angle} = S_{pitch} \cdot pitch_{angle}$$

$$yaw_{angle} = S_{yaw} \cdot yaw_{angle}$$

The effects of temperature on the sensors and the relative drift have been neglected because the mouse is designed to work always inside a house, so at room temperature (25 °C) or slightly higher or lower and so the relative drift in the mouse due to temperature would be very small.

Design choices

Bluetooth module, battery charger and oscillator

As said before the RN42 has been left always connected when the system is on; on the other hand, since a mouse should work quite near the target host device, the transmit power has been lowered from 4 dBm to -8 dBm in order to save power. It has been tested and it works from about 8 meters away. Anyway, since the power consumption of the RN42 when it is connected and it's not trasmitting is equal to 25 mA, a Li-ion rechargeable battery is included in the design and so also a battery charger IC,

precisely the MAX1551. This choice has been made since some Bluetooth modules have auto-reconnection problems when they are switched on and off many times; so for the user point of view manually reconnecting the mouse to the host device is very boring. So the user owns a device that is always connected, at the price of a smaller time period between two consecutive battery charges. A micro-USB connector has been chosen, in order to allow the device to be charged with almost every commercial smartphone cable.

Finally a 8 MHz ceramic SMD crystal has been chosen, the value was the same of the one implemented in the Nucleo board used in the prototype.

Switches, LEDs and SWD programmer

A led has been added and it blinks only at the power-on of the circuit, to show that the system is working; it is driven by a PWM for 60 ms, with a duty cycle equal to 30%. Another led has been used, to indicate the charging of the battery. It is connected between the USB charger and the MAX1551 charger IC POK pin; the POK pin is active low and so when the battery is charging the led is on and about 5 mA flows in it.

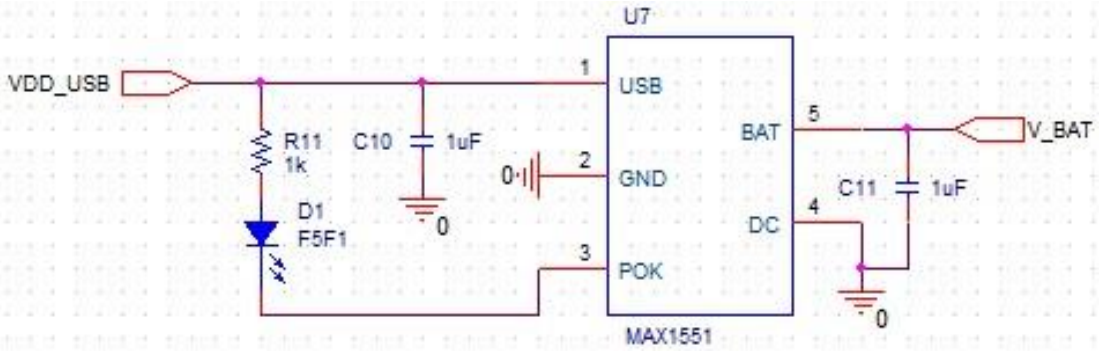


Figure 21: Schematic of the MAX1551 charger IC with a LED.

A slide switch has been implemented between the battery and the voltage regulator that supplies the entire system; in this way the user can disconnect the system from the power supply, but this choice allows the recharging of the battery also with the system switched-off.

Finally a SWD programmer has been implemented; it allows to program the μC, and so also the Bluetooth module, on the field, using the upper part of the STmicroelectronics Nucleo board (used for prototyping), connecting it to the *Flying mouse* using jumpers. It could be useful to calibrate the mouse response in the final board, that since it is smaller than the prototype would move faster in the user's hand and it allows to not spend a lot of money buying an additional programmer.

Software flow

At the power-on the dedicated led blinks for about 60 ms; then there is an initialization phase, where the id of the sensors are read, the various ISR (Interrupt Service Routine) are linked to the correspondent event (e.g. timer period expiring or external interrupt on the falling edge of a pin, when a button is pressed). Then the internal pull-up resistors of the pins where the left and right buttons are linked are disabled and a periodic timer is set, in order to wake-up the processor every 5 ms. At this moment an infinite loop is entered and it acts in the following way: the microcontroller goes in sleep mode. Then, every time a button is pressed, the correspondent isr set a flag to 1; the flag remains to 1 until a button is released, than another ISR set the flag to 0. It allows to move the mouse with the buttons pressed. Furthermore a check is done that allows to only one button to be active at the same time.

When the μC wakes-up due to expiration of the timer period, it acquires the accelerometer ad gyroscope values, it checks if the buttons have been pressed or released while it was sleeping and executes the proper function, sending different ASCII values to the Bluetooth module, that then sends the correspondent Raw Report HID packets to the host. Then the timer flag is reset and the μC returns in sleep mode. So the μC does a timer-driven periodic polling check and update.

An additional functionality has been added; if the mouse is not moved for a long enough time period, the μC enters into deep sleep mode. So, if the horizontal movement is very near to zero a counter is increased (the vertical movement has not been considered, since it depends also on the tilt angle of the accelerometer and so if the mouse is left in a non-horizontal position it tracks a vertical movement). If this counter reaches a value equal to 600 the μC enters into the deep sleep mode, so if the mouse is left quiet for a little bit more than 3 seconds. Every time the mouse is moved horizontally the counter is reset to zero. From the deep sleep mode the microcontroller can be waken-up only by an external interrupt and so by the pressure of the left or right button.

Chapter 5:
Prototype and verification

As suggested by Professor Pasero a set of stackable STM development boards have been used. Precisely a STM32 Nucleo-64 board and a X-NUCLEO-IKS01A1 MEMS inertial and environmental sensor evaluation board have been used to fast prototype the project; on the other hand, for the already discussed reasons, instead of the STmicroelectronics Bluetooth class 4 evaluation board a breakout board of the RN42 module from Sparkfun had been used: the BlueSMiRF HID module. The firmware has been developed using *ARM mbed Developer Site*, an online free Integrated Development Environment, that was also suggested by the Professor; it allowed a fast prototyping, but it's not so good from the optimization point of view, since it's not possible to change some functions and to really know what happens at low level.

ST-NUCLEO-L476RG

This Nucleo board has the STM32L476RGT6 microcontroller in LQFP64 package embedded, the Arduino connectivity support and ST Morpho headers make it easy to expand the functionality of the Nucleo platform with a wide choice of specialized shields. The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

Main features:

- Two types of extension resources:
  - Arduino Uno Revision 3 connectivity
  - STMicroelectronics Morpho extension pin headers for full access to all STM32 I/Os
- On-board ST-LINK/V2-1 debugger/programmer with SWD connector
  - Selection-mode switch to use the kit as a standalone ST-LINK/V2-1
- Flexible board power supply
  - USB VBUS or external source (3.3 V, 5 V, 7 - 12 V)
  - Power management access point
- User LED (LD2)
- Two push buttons: USER and RESET

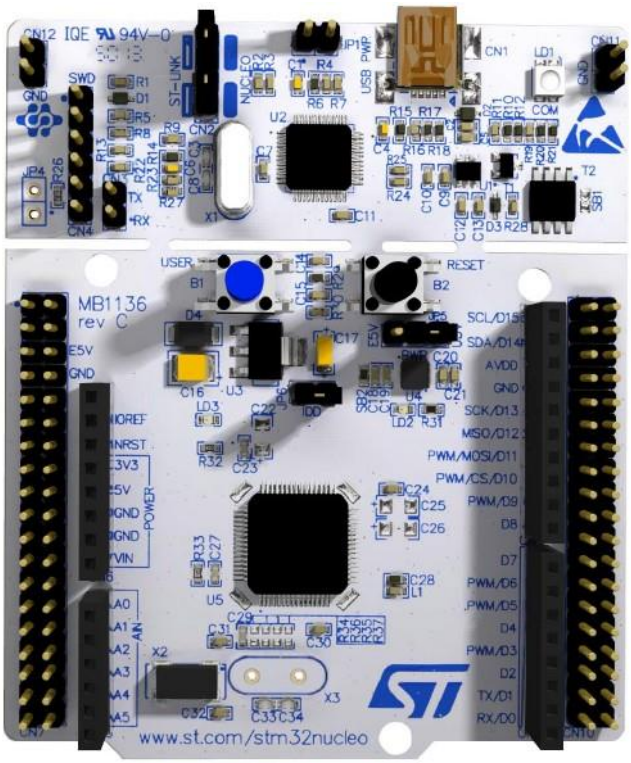


Figure 22: STM32 Nucleo-64 board.

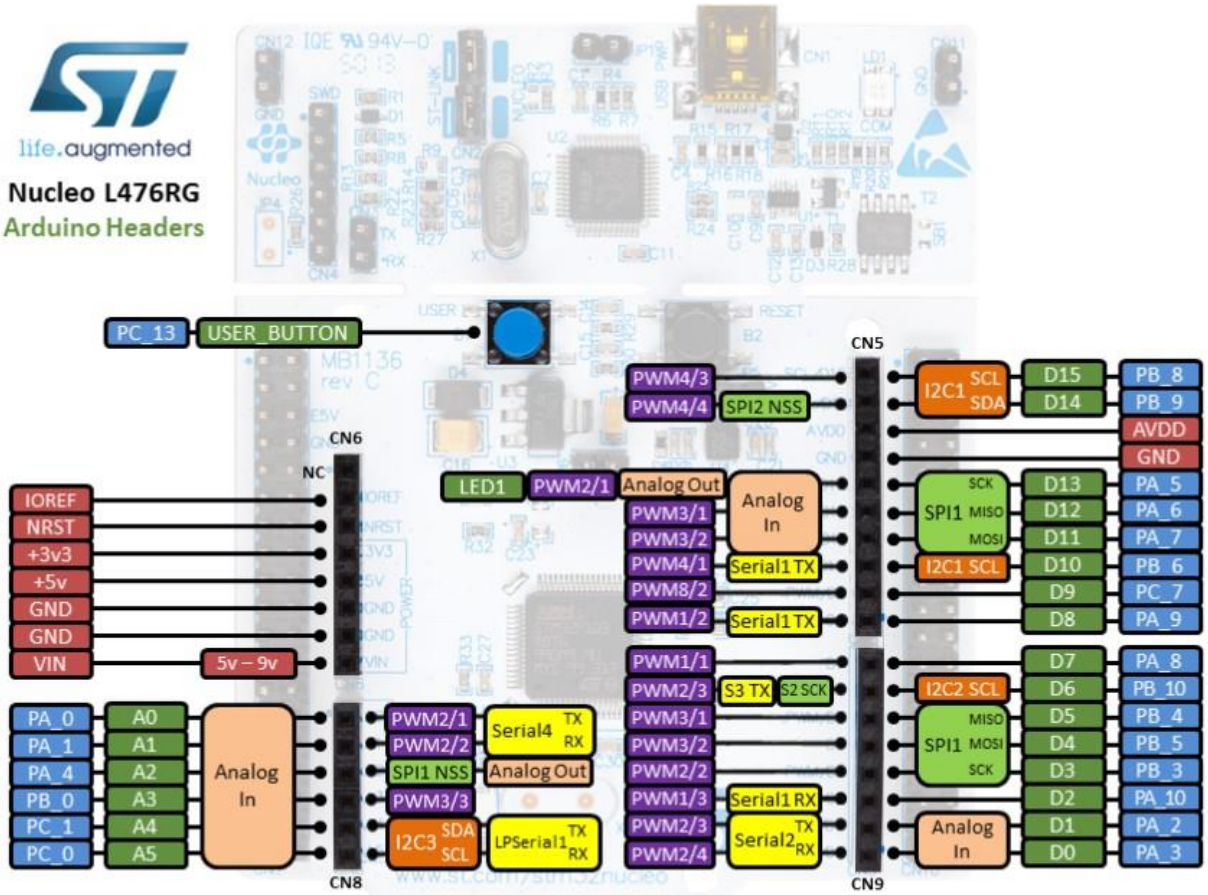


Figure 23: Arduino-compatible headers.

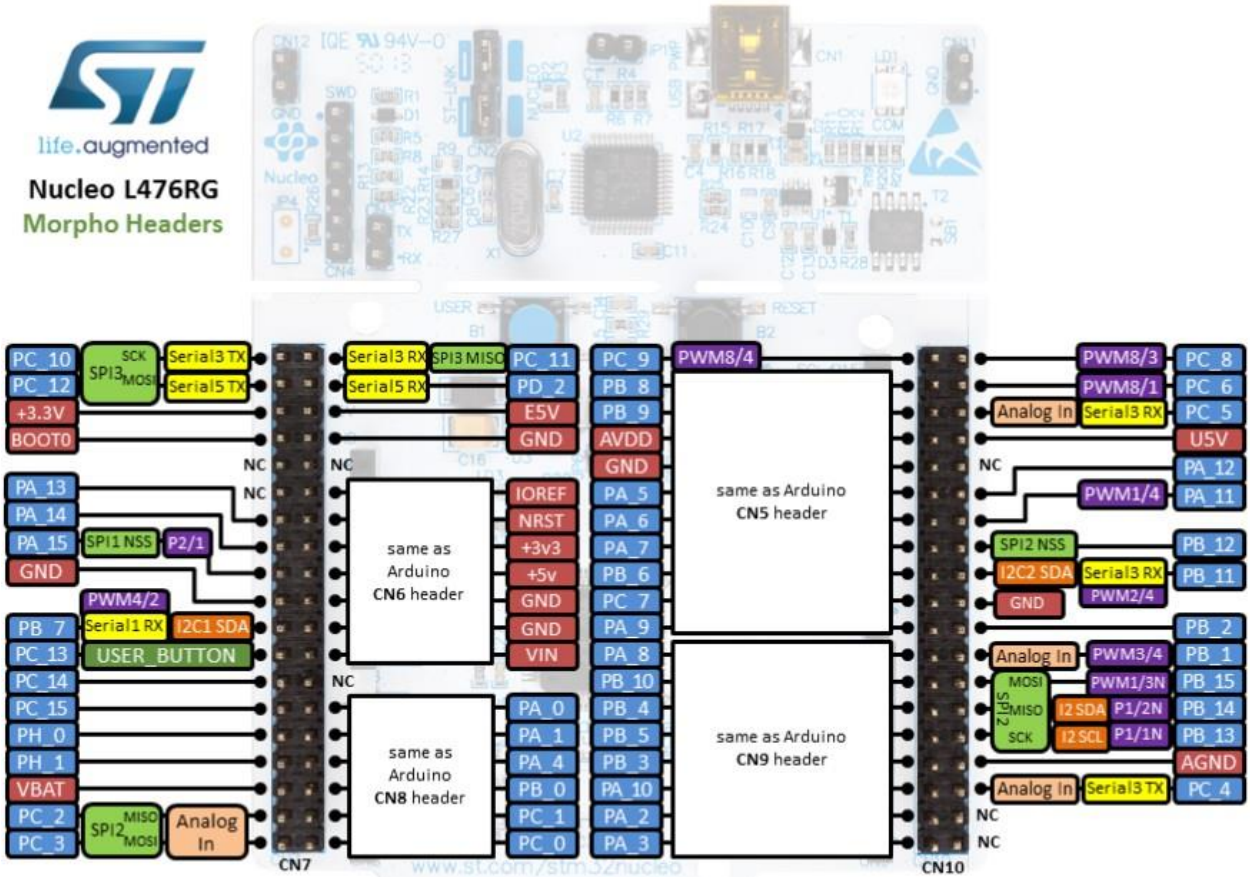


Figure 24: Morpho headers.

X-NUCLEO-IKS01A1

The X-NUCLEO-IKS01A1 MEMS inertial and environmental sensor evaluation board allows to test the functions of the STMicroelectronics motion MEMS. The devices used on the board are:

- LSM6DS0: MEMS 3D accelerometer (±2/4/8g) + 3D gyroscope (± 245/500/2000dps)
- LIS3MDL: MEMS 3D magnetometer (±4/8/12/16 gauss)
- LPS25HB: MEMS pressure sensor, 260-1260 hPa absolute digital output barometer
- HTS221: Capacitive digital relative humidity and temperature sensor

The X-NUCLEO-IKS01A1 interfaces with the STM32 MCU via the I^2C pin. The sensors are connected on a single bus via I²C and each device has a separate power supply to allow measurement of the power consumption of each single sensor. The board also has a DIL24 socket to connect and test additional sensors.

Figure 25: X-NUCLEO-IKS01A1 board.

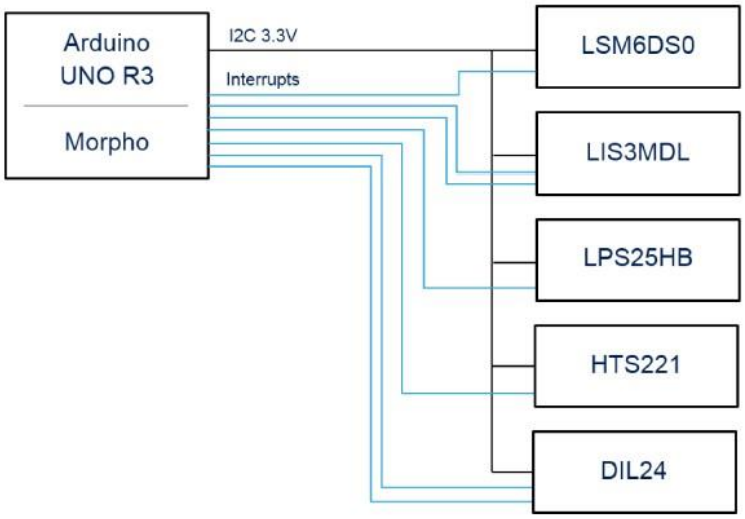


Figure 26: X-NUCLEO-IKS01A1 block diagram.

BlueSMiRF HID module

It is a breakout board from Sparkfun; it has two LEDs that shows the status of the Bluetooth connection, a voltage regulator and ease the connection between a microcontroller and this module thanks to a pin header line (on which a male pin header has been soldered).

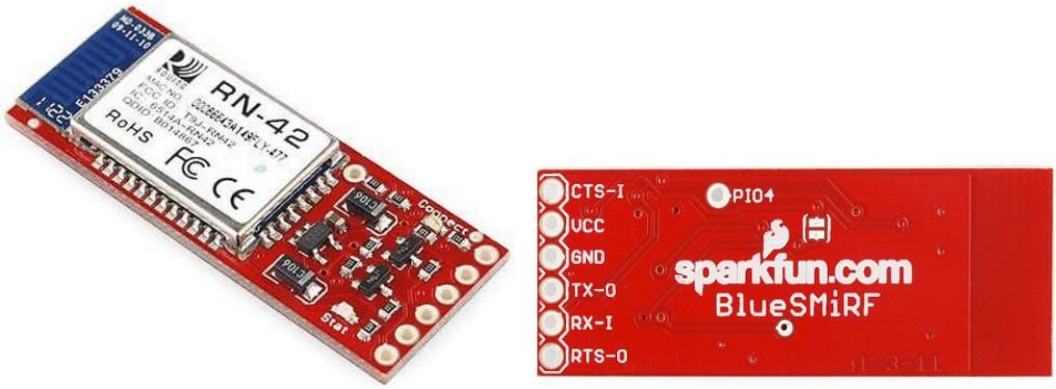


Figure 27: BlueSMiRF HID breakout board from Sparkfun, on the right there is the bottom view.

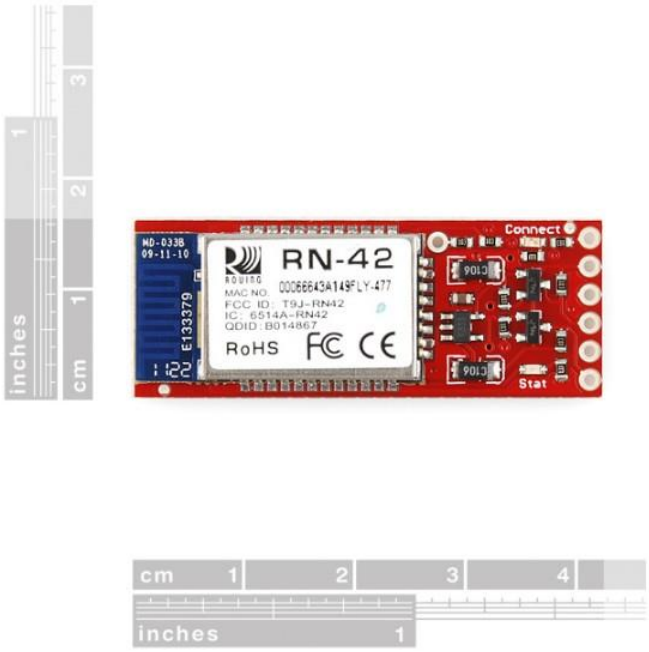


Figure 28: BlueSMiRF HID breakout board top view and dimensions.

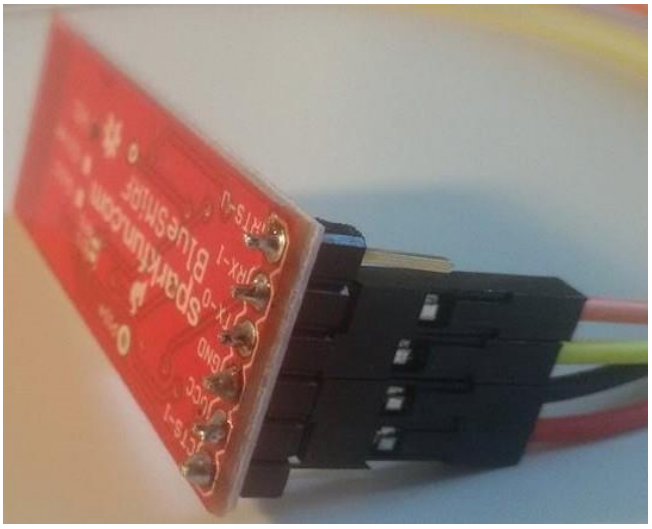


Figure 29: Breakout board with the soldered pin header.

Tactile switches

Two tactile switches have been used together with two 4.7 kΩ pull-up resistors; they have been soldered together with some jumpers over a stripboard.

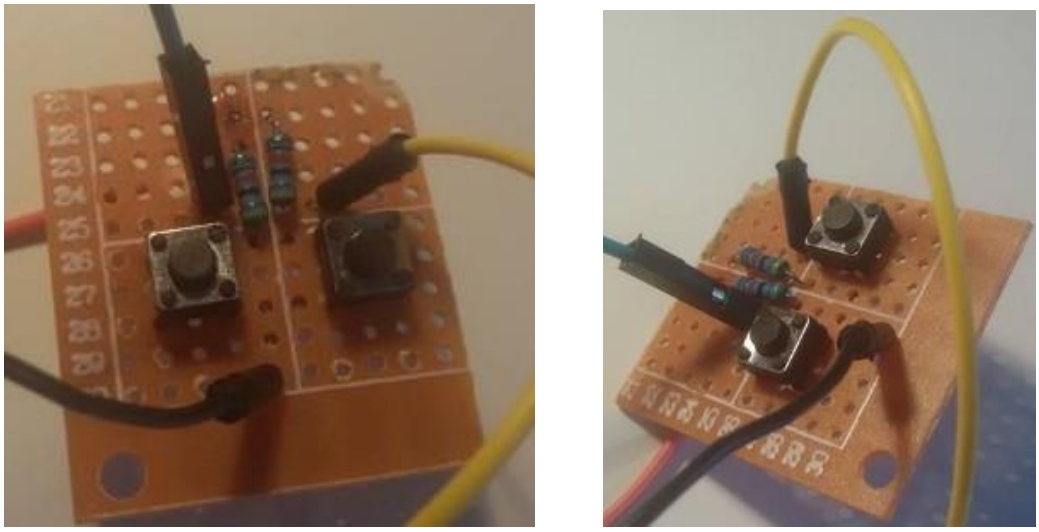


Figure 30: Soldered tactile switches and pull-up resistors.

Verification

The prototype has been tested using as a host both a Samsung Galaxy Ace II smartphone and a Samsung NP-RC530 laptop and it works until a distance of about 8 meters inside a house. From the figure below can be seen as the smartphone recognizes the *Flying mouse* as a mouse device and that is able to connect to it.

Figure 31:Screenshot of the Samsung Galaxy Ace II.

As can be seen by the following figures, the prototype is not very handful actually, but it needs only to demonstrate that it can work properly; the final board will be much smaller.

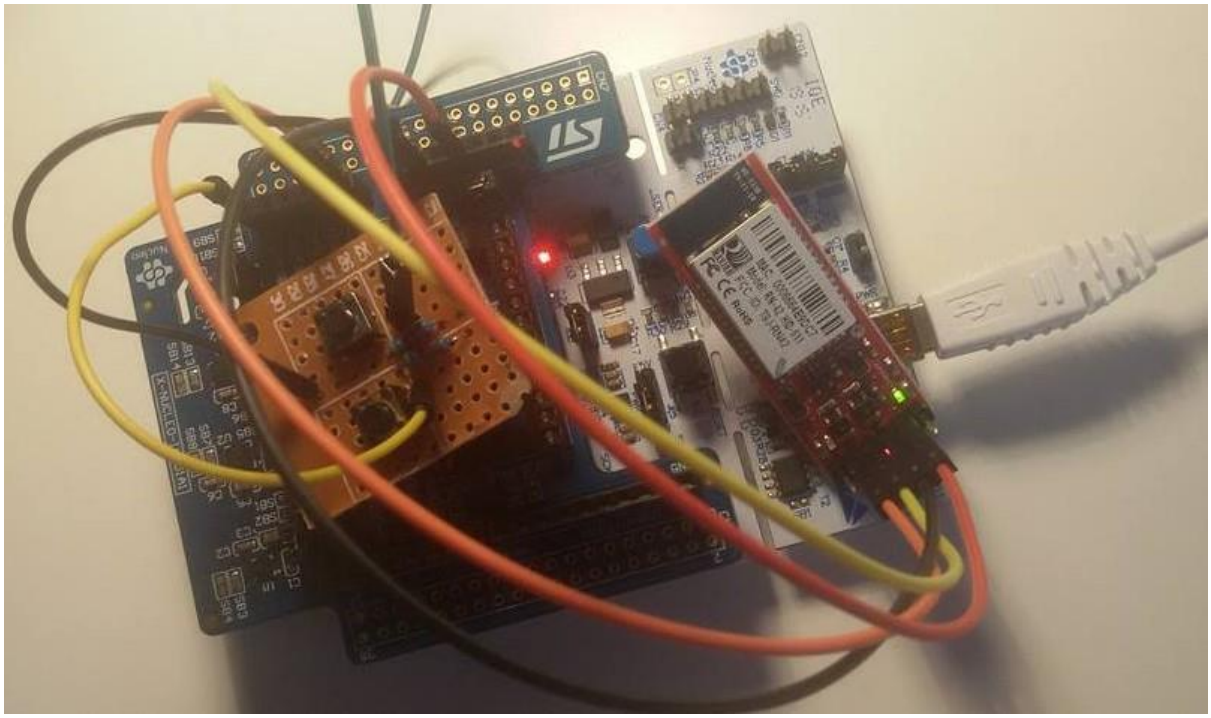


Figure 32: Lateral view of the prototype.

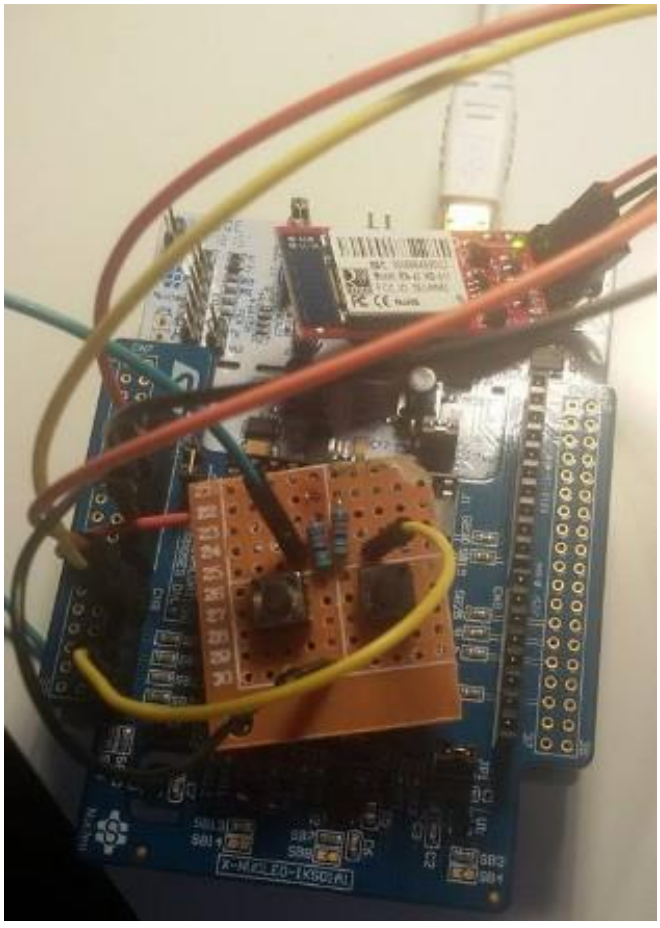
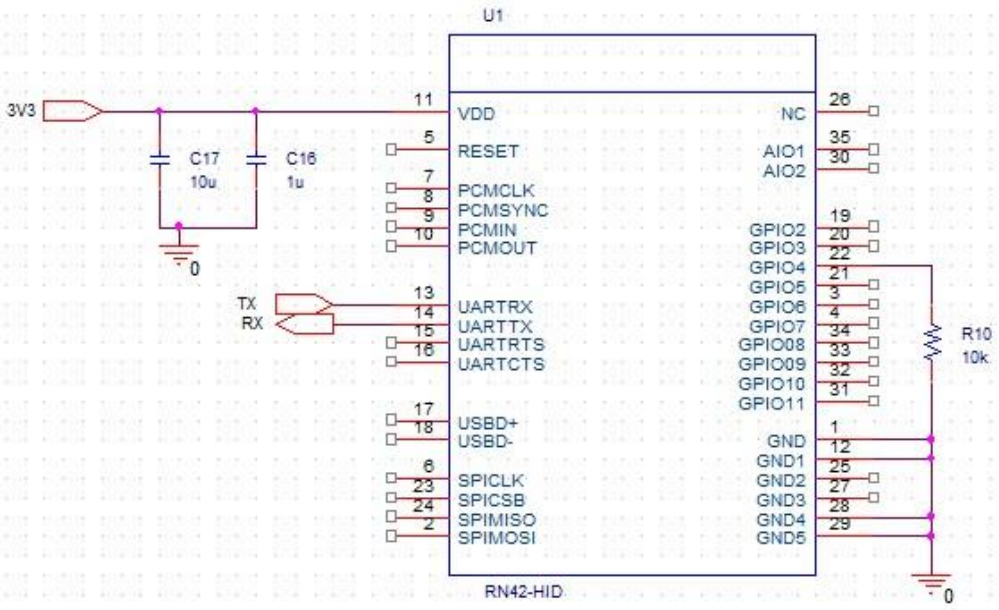


Figure 33: User point of view of the prototype

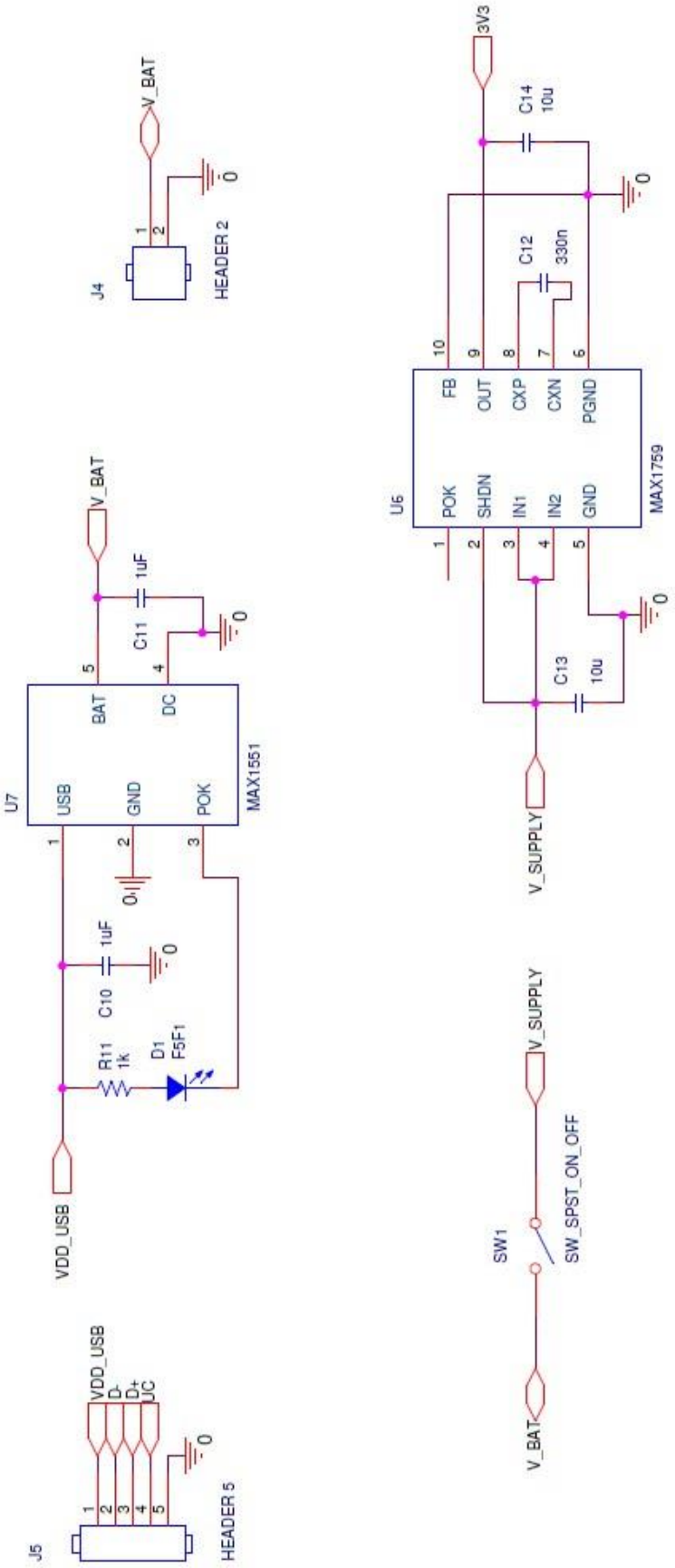
Chapter 6:
Schematic diagrams

The schematic diagrams have been developed setting one Cadence OrCAD Capture CIS page for each different topic of the design. In order to choose the pins and some connections, the Nucleo board connections have been followed, since it works in the prototype; furthermore some capacitors and resistor values have been chosen following the suggestions written in the datasheet of the various components. The schematic diagrams are very important since they are the main reference document of the design.

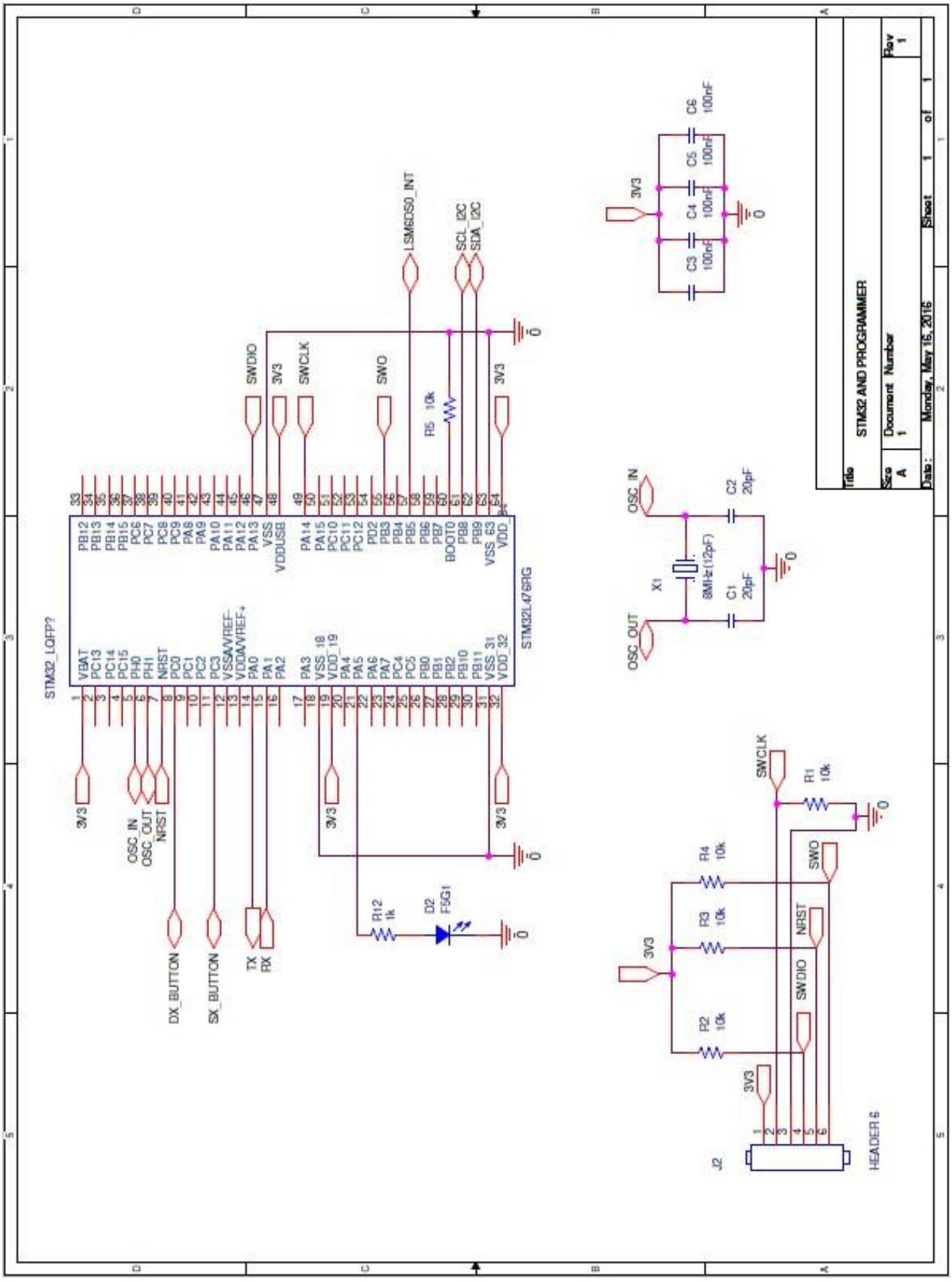
Bluetooth module



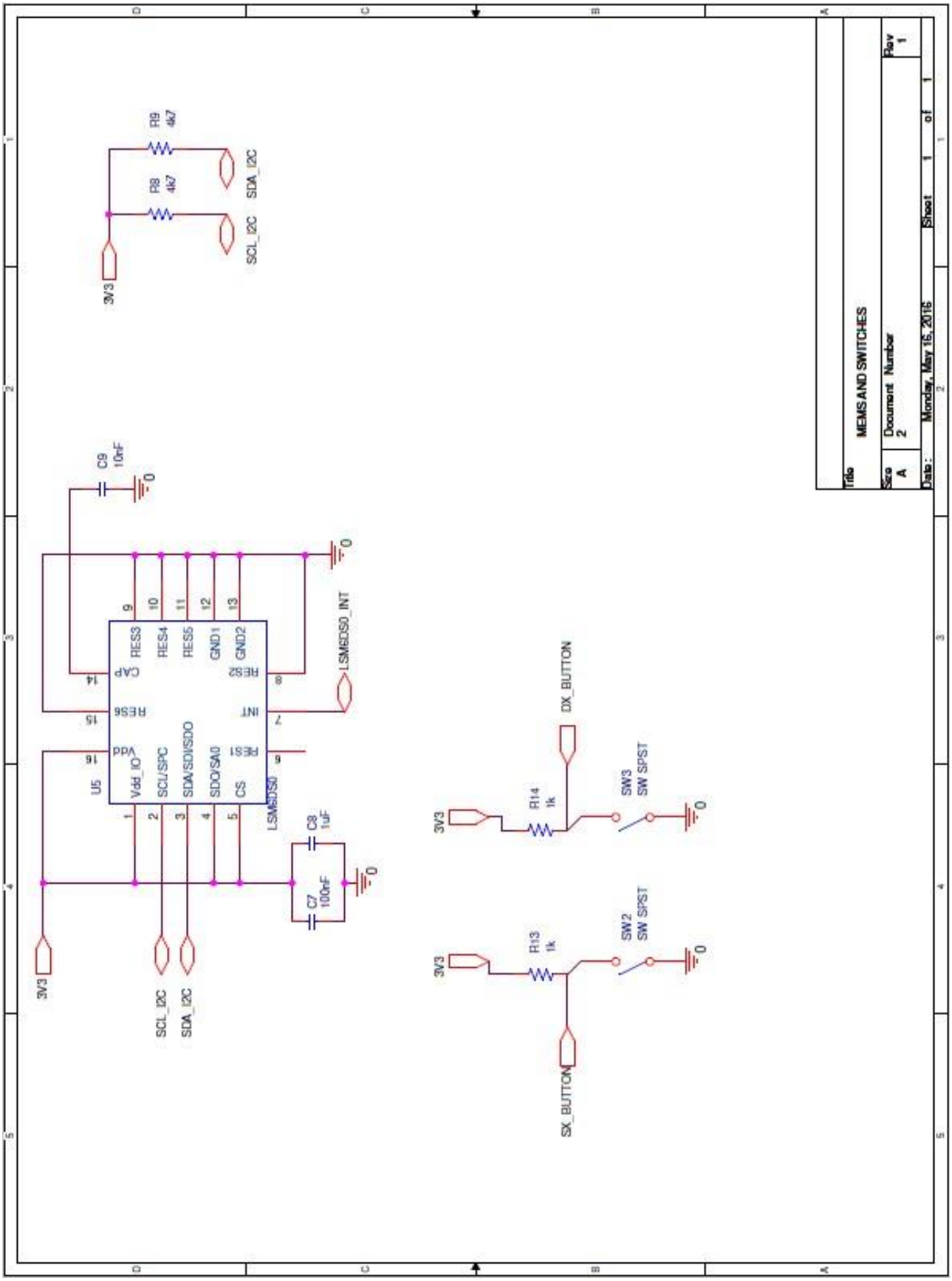
Power supply and on/off switch



STM32L476RG and SWD programmer



MEMS and buttons



Chapter 7:
BOM: Bill Of Materials

The Bill Of Material has been initially computed with the automatic tool offered within Cadence OrCAD Capture CIS and then integrated with models and prices from Digikey. Only one distributor has been considered in order to reduce the order cost doing a single order. Anyway the unit price is reported for a low quantity order; the price can be reduced a lot by ordering components in stock and by changing distributor for the most expensive components. Only the Lithium Polymer rechargeable battery has not been found in the Digikey website, so it is referred to a possible online order on Amazon.it. The SMD package is reported in Imperial code.

Item	Quantity	Reference	Part value	Package reference	Manufacturer	Unit price [€]	Subtotal [€]
1	2	C1,C2	20pF	SMD 0603	Samsung Electro-Mechanics America, Inc.	0.09	0.18
2	5	C3,C4,C5,C6, C7	100nF	SMD 0603	Murata Electronics North America	0.10	0.5
3	3	C10,C11,C16	1uF	SMD 0603	Murata Electronics North America	0.13	0.39
4	1	C9	10nF	SMD 0603	TDK Corporation	0.21	0.21
5	1	C12	330nF	SMD 0603	TDK Corporation	0.16	0.16
6	4	C8,C13,C14, C17	10uF	SMD 0603	Murata Electronics North America	0.34	1.36
7	2	D1,D2	Red LED	SMD 0603	Panasonic Electronic Components	0.44	0.88
8	1	J2	HEADER 6	Male pin header 6	Amphenol FCI	0.24	0.24
9	1	J4	HEADER 2	B2B-PH-K-S	JST Sales America Inc.	0.15	0.15
10	1	J5	HEADER 5	0473461001	Molex, LLC	0.77	0.77
11	6	R1,R2,R3,R4, R5,R10	10k	SMD 0603	Rohm Semiconductor	0.09	0.54
12	2	R8,R9	4k7	SMD 0603	Yageo	0.09	0.18

13	4	R11,R12,R13, R14	1k	SMD 0603	Yageo	0.09	0.36
14	1	STM32_LQFP	STM32L476RG	64-LQFP	STMicroelectronics	9.44	9.44
15	1	SW1	SW_SPST_ON _OFF	AS11AH	NKK Switches	3.60	3.6
16	2	SW2,SW3	SW SPST	1825910-6	TE Connectivity Alcoswitch Switches	0.10	0.2
17	1	U1	RN42-HID	RN42HID-I/RM	Microchip Technology	13.,49	13.49
18	1	U5	LSM6DS0	LSM6DS0T 16LGA	STMicroelectronics	4.45	4.45
19	1	U6	MAX1759	10-uMAX	Maxim Integrated	6.33	6.33
20	1	U7	MAX1551	TSOT-23-5	Maxim Integrated	1.,79	1.79
21	1	X1	8MHz(12pF)	4-SMD	Abracon LLC	0.58	0.58
22	1	BAT	3.7V 500mAh Li-Po	Battery with JST connector	Syma	4.79	4.79
TOT	43	ND	ND	ND	ND	ND	50.59 €

Chapter 8:
PCB layout

To develop the Printed Circuit Board layout and extract the final Gerber files, Cadence Allegro PCB Editor has been used extracting the netlist from Cadence OrCAD tool. The footprints and the solderpads (the solderpads are called padstacks in Cadence software) of many components have been manually drawn, checking their correct sizes.

To decide the width of the traces, the spacing between them and the Vias drill size I referred to the OSH Park website; the choice of this manufacturer was made since on www.oshpark.com is possible to order PCBs, in particular 2 layer boards cost $5 per square inch (with 3 copies of the board included in that price), so it's a very good price for an initial test board. It offers also a board thickness of 0.8 mm with copper foils on both sides, for a total copper thickness equal to 70 μm (2oz: ounces per square foot); the dielectric is FR4 (Fire Retardant Epoxy number 4). The implemented signal traces are 8 mils wide, while the power and ground lines are 12 mils wide; the standard value of the Cadence software was equal to 5 mils. The minimum spacing between all traces instead is equal to 8 mils. The Vias (called also plated through holes) have the drill equal to 13 mils. The traces could be done also thinner, but it was not necessary and so a cost-effective solution has been adopted.

To choose the board size, the initial reference was the size of a mini-mouse (I took as reference the SKINTEK SK-TM032-B model), that was 50x80 millimetres. From this starting point I reduced the size until reach 41.2 x 65.1 millimetres. I chose to not do it even smaller in order to avoid manageability problems. To reduce the board dimensions I also chose a 0603 SMD package (Imperial code) for resistors and capacitors.

The PCB is a two layers board, on the top layer there are the components and the traces, while on the bottom layer there are only few signal traces and the ground plane. The layout is not mirrored, but the manufacturer design guidelines should be checked carefully to know if the layout should be given to the manufacturer in the correct position or mirrored.

The decoupling capacitors of the microcontroller, MEMS, voltage regulator, battery charger and Bluetooth module have been placed as near as possible to the correspondent power supply pins. The ground plane, since it is on the bottom layer, is linked to the various pins through vias. Under the PCB trace antenna of the RN42

module is not present the ground plane, as suggested by the Bluetooth module datasheet. The trough-hole pins of the buttons have been implemented with a larger annular ring, in order to ease a future manual soldering, reducing the risk of create some short-circuits between the pin and the ground plane.

Finally, the micro-USB connector has been placed slightly outside the board limit, in order to ease the introduction of the male USB connector inside it, considering a future device package.

Board View

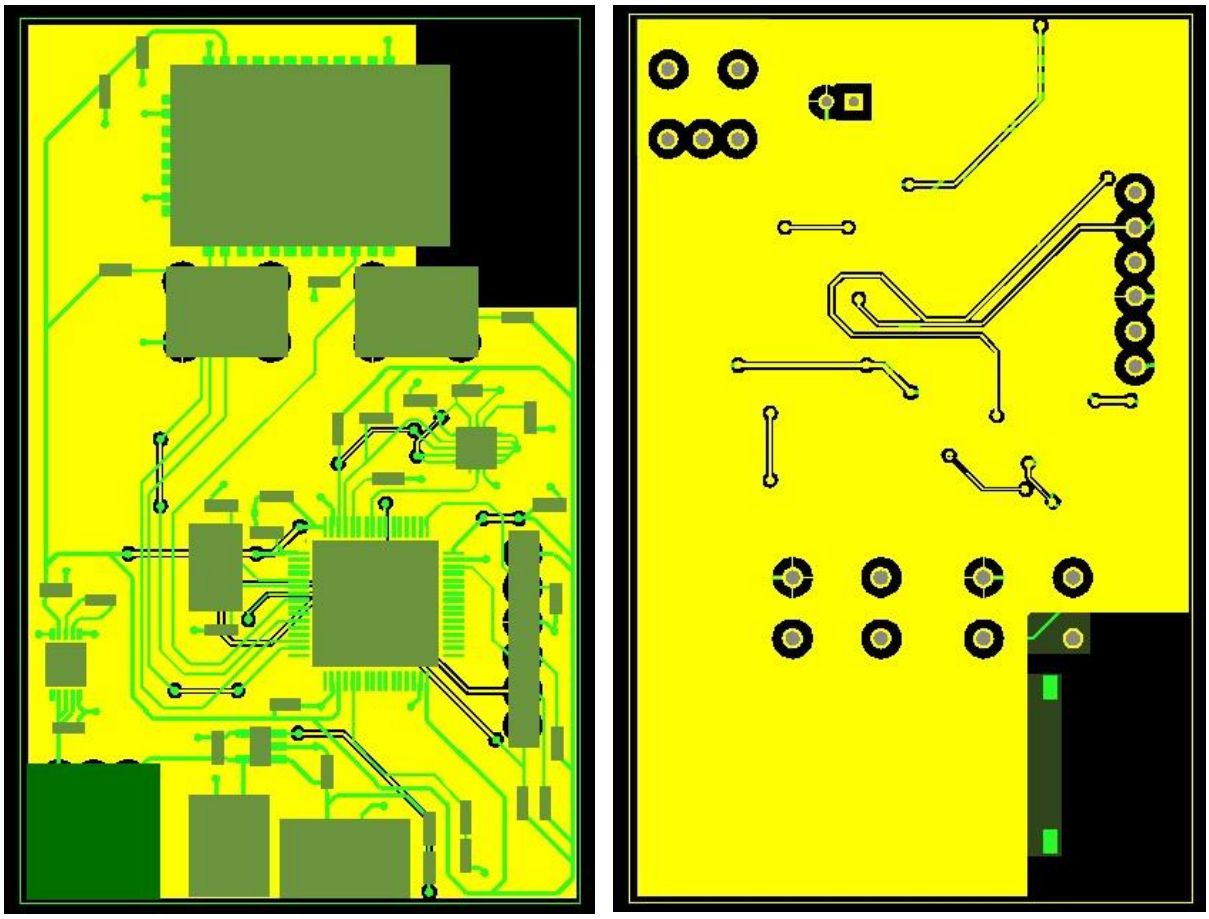


Figure 34: Top view of the PCB on the left, bottom view on the right.

In the 3D view of the board the height of the components have been adjusted, but is not set exactly, especially for the buttons and switches, which model is simply a cube.

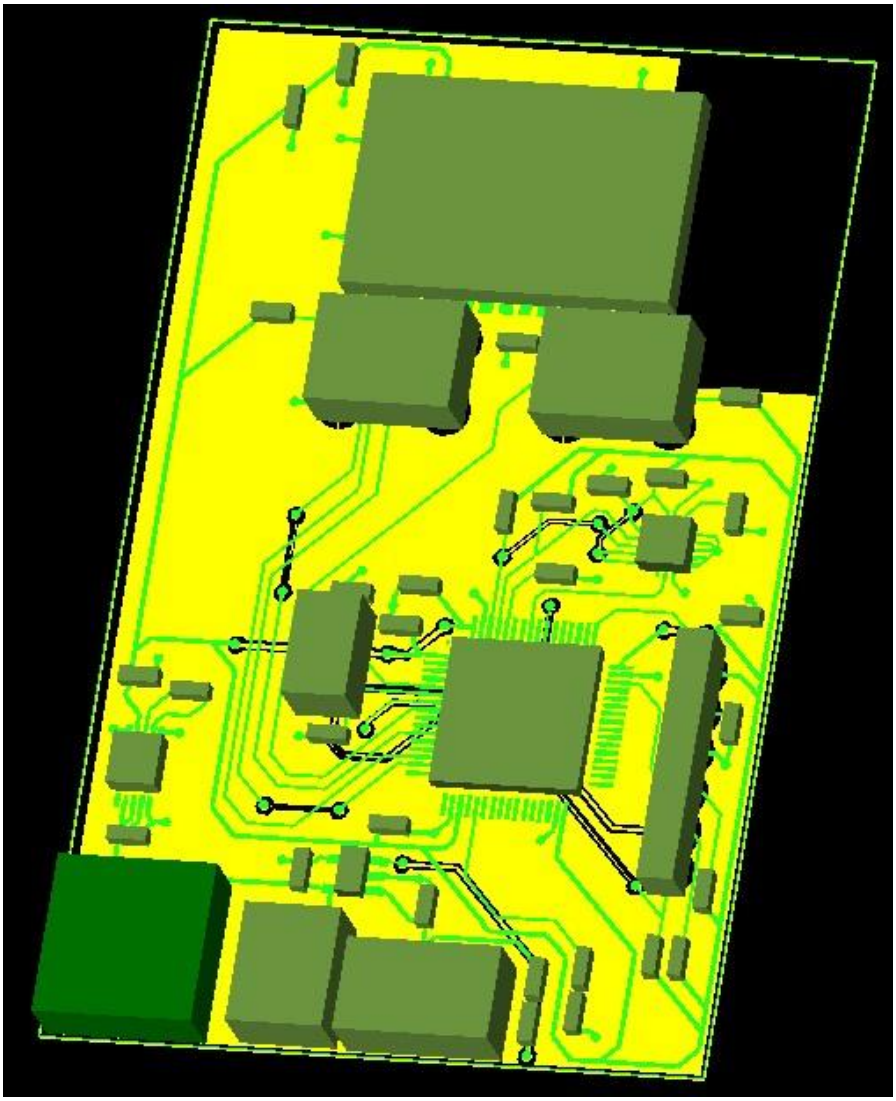


Figure 35: 3D view of the PCB.

Board layout

Finally, the last step was to extract the Gerber files of the design. So the artworks of the top and the bottom layers have been created, in order to allows the photolithography and then the etching processes, to remove material from the core of the board, leaving only the desired conductive paths. Then the soldermasks of the top and the bottom layer have been created (the soldermask is the (usually) green material on board, that is a layer of polymer that protects the copper traces against oxidation, leaving openings for the pads to which components will be soldered; it can be also of another colour); then there would be an exposed conductor plating to protect the exposed pads from oxidation.

Also the silkscreen (present only on the top layer) has been created in order to add documentation to the board and ease the placement and soldering of the components.

Finally a NCD (Numerically Controlled Drill file), called also Excellon file, has been created, which reports information about the drill sizes and locations; it comes together with a NC legend. These artworks are reported below:

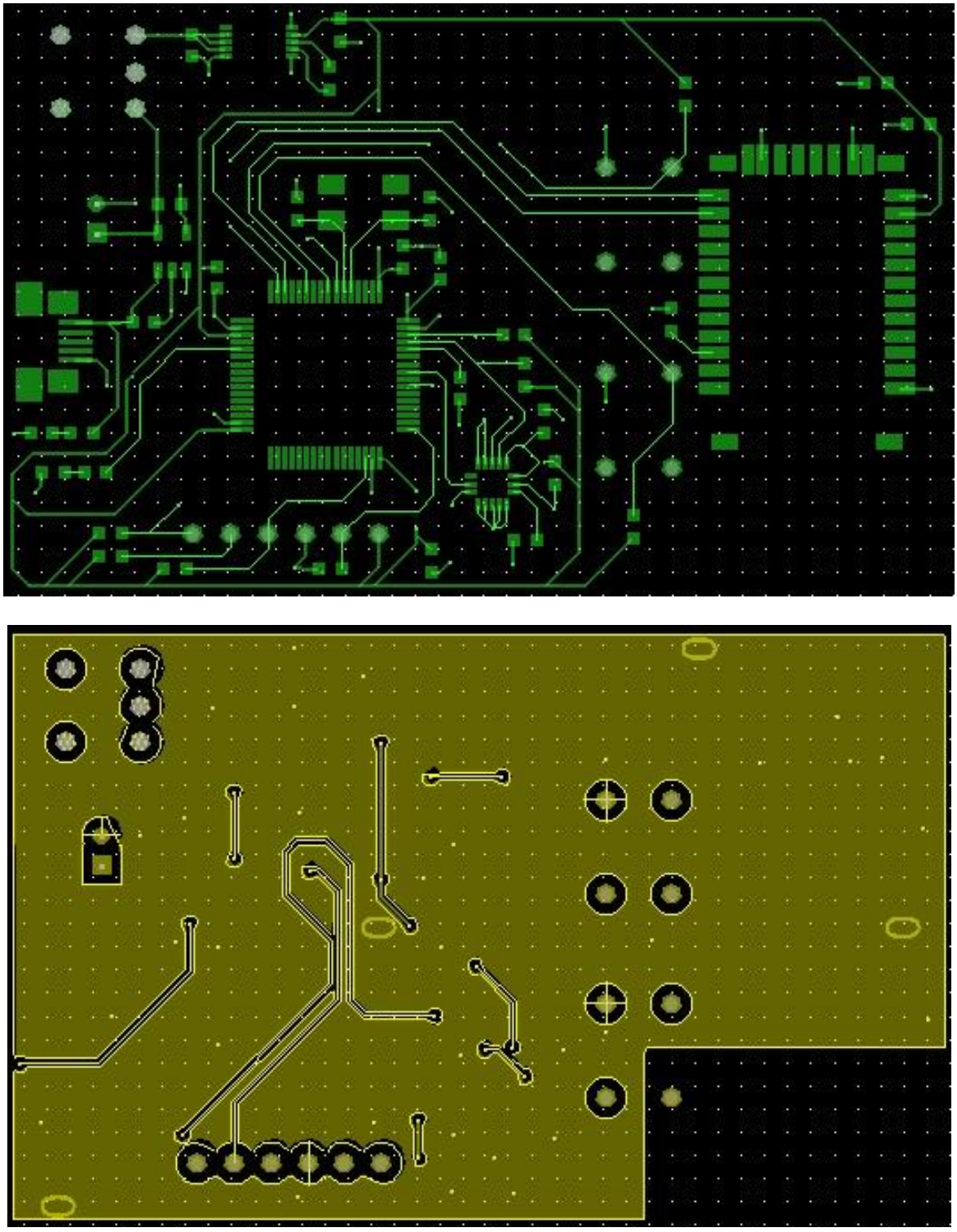


Figure 36: Top and bottom artworks on the upper part of the figure and on the lower part respectively.

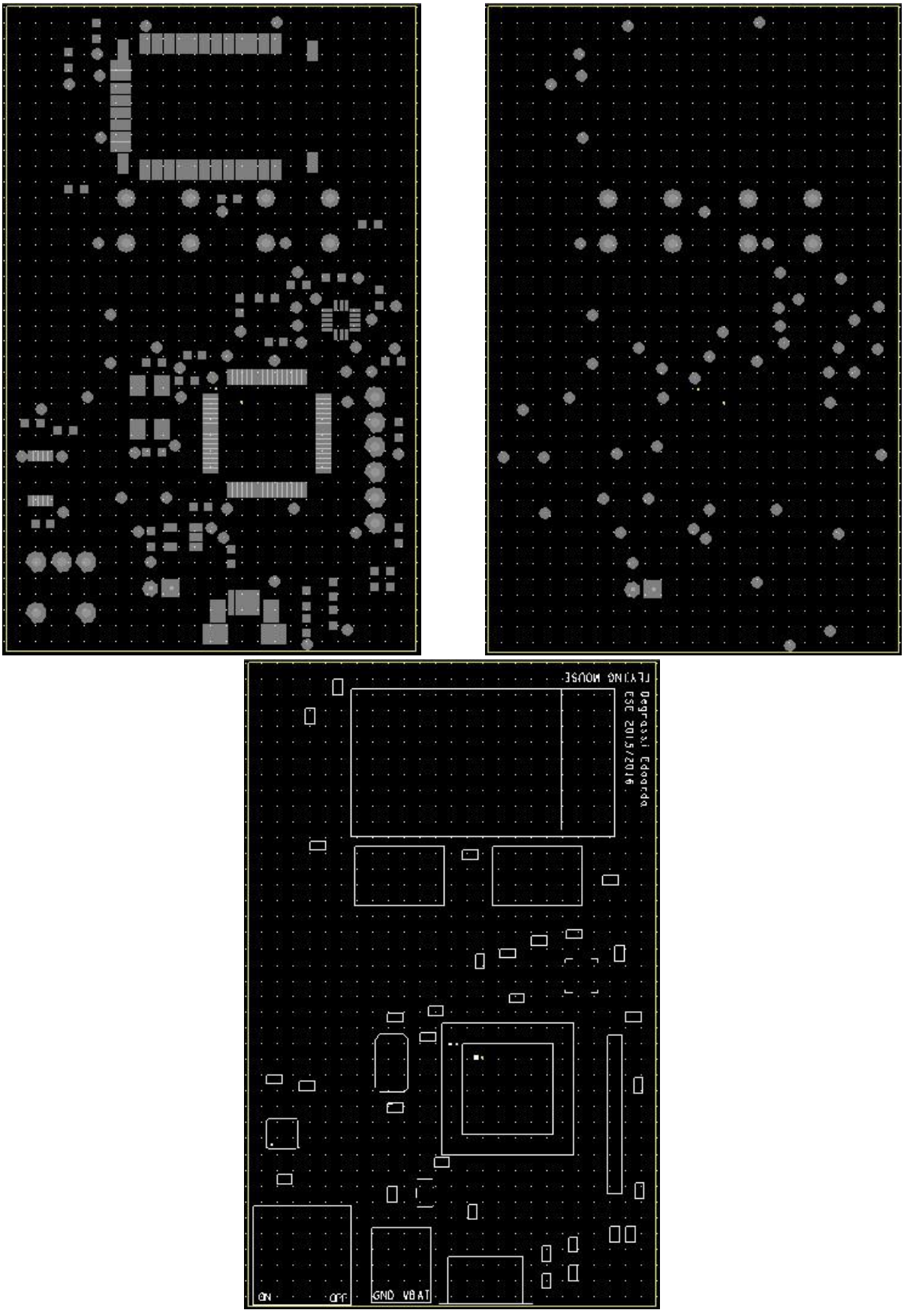


Figure 37: Top and bottom soldermask on the left and right respectively. Silkscreen on the bottom.

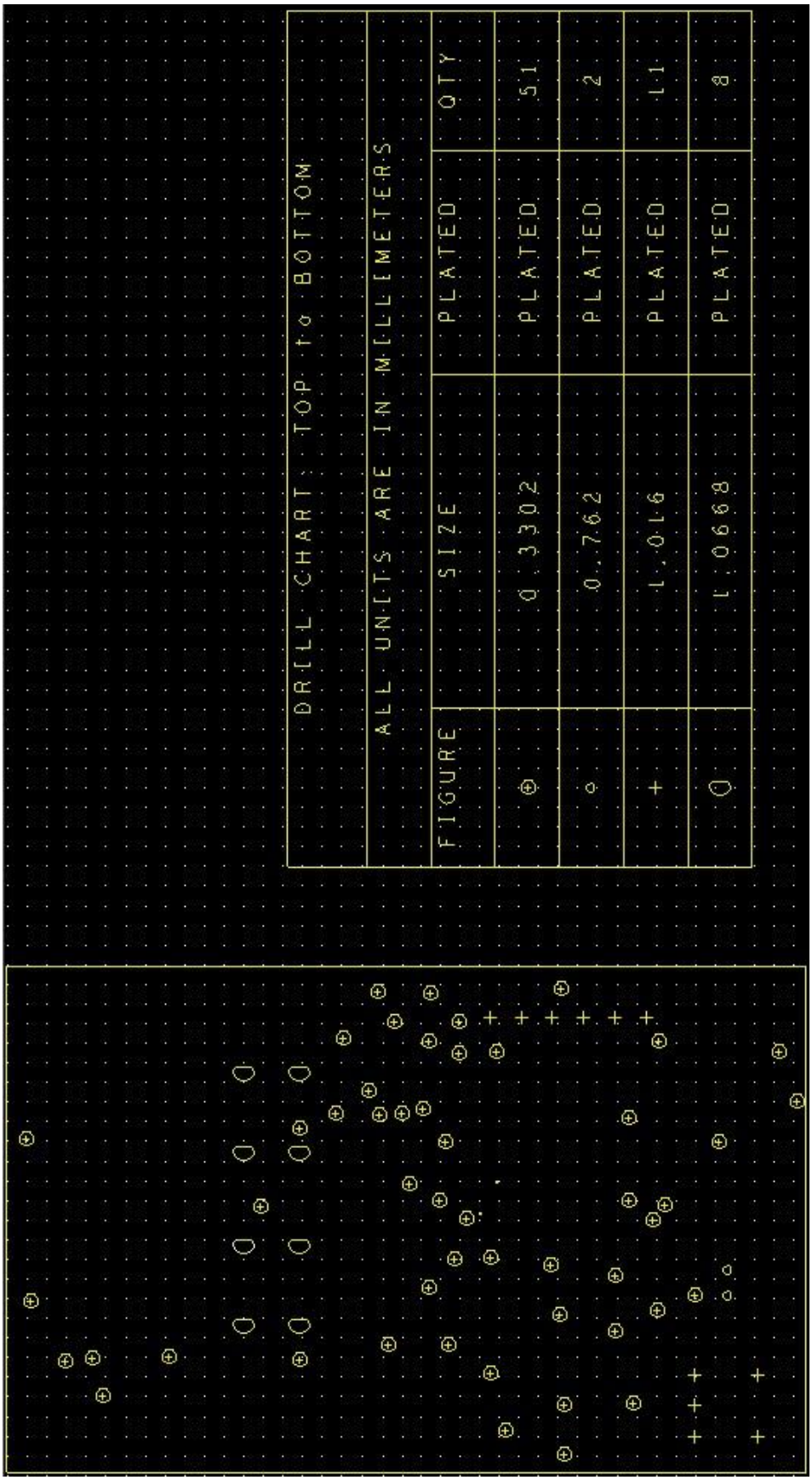


Figure 38: Drill artwork and drill legend.

Chapter 9:
Market analysis and future developments

The device is a wearable Bluetooth mouse and it can offer a more enthusiastic game experience to videogame players. The Flying mice does not need an external USB dongle. This is due to the Bluetooth wireless capability; in fact, unlike most other wireless device technology, Bluetooth technology is built into most of today's laptops and desktops and all smartphones support it. So it could be used on eache Bluetooth class 2.0 enabled laptops, smartphones and tablets, allowing to watch video on each of them from a couch, interacting with the devices from distance. There isn't the need of a particular installed driver on the host thanks to the HID profile of the Bluetooth module. Furthermore, the Flying mice can be used to avoid the carpal tunnel syndrome, since people should no more place its hand on the table, but has only to move it in the air.

There are no products like this one available in the market, unless very expensive ones. So, since it has a product cost in the worst-case equal to 50 euro, it could become very cheaper and so be very competitive.

Anyway, actually the Flying mouse is still considered a prototype, to be optimized and improved. For this reason there are many possible future developments; the device is very malleable and it can be transformed for example into a finger mouse, in order to free the users' desktop and laptop bags of tangled wires; actually already exist a product like this and it's called Mycestro, but it costs around 145 €, shipping it to Italy and 149 $ on Amazon.com. So the *Flying mice* could be a great cost-effective market competitor of it. Furthermore, if it would be needed for wearable problems, a flexible PCB (with a thickness equal to 0.1 mm) could be implemented; this could lead also to a glove implementation or any other type of wearable motion tracking device. So it could be used to track movements of patients doing rehabilitation movements, to recognize and interpret sign language, to offer a more enthusiastic game experience to videogame players, and so on and so forth. The design and the prototype are ready to be improved changing only the board layout, because the powerful microcontroller allows huge computations and the single IC MEMS containing both the accelerometer and the gyroscope allows to keep a reduced size.

Appendix A: User manual

In this short User manual only the Android smartphones and the Windows PCs have been considered, since they are the most diffused operating systems.

Using the *Flying mouse* with an Android smartphone

How to establish a connection

- Switch-on the *Flying mouse*
- Go to *System setting* → *Connections* → enable the *Bluetooth* connectivity
- Click on the *Bluetooth* button → *Scan* to search new devices
- When the *Flying mouse* will be displayed on the screen click on it to pair and connect your smartphone with it
- If the connection does not happen for a while, or the mouse does not automatically reconnect with the smartphone after a disconnection repeat the previous steps

How to set the pointer speed

You can set the pointer speed directly on your smartphone; to do so:

- Go to *System settings* → *Controls* → *Language and input* → *Mouse/trackpad*
- Click on *Pointer speed* and set the speed as You prefer

Using the *Flying mouse* with a Windows 10 laptop

Establishing a connection

- Switch-on the Flying mouse
- Go to Control panel → Hardware and sound → Add device

- When the *Flying mouse* will be displayed on the screen click on it to pair and connect your laptop with it
- If You are asked to choose the type of connection, choose the connection without any code
- If the connection does not happen for a while, or the mouse does not automatically reconnect with the smartphone after a disconnection repeat the previous steps

How to set the pointer speed

- Go to *Settings* → *Devices* → *Mouse and touchpad* → *Additional mouse options*
- Select *Pointer options* → *Movements*
- Set the speed as You prefer

Appendix B: Bibliography

- Electronic Design Documentation, R. B. Darling, University of Washington
- 120 Watt Stereo Power Amplifier using the Leach Low-TIM3A Design, R. B. Darling, University of Washington
- Electronic Systems Engineering slides, E. Pasero, Politecnico di Torino
- STM32L476xx family datasheet
- AN4621: Application note, STM32L4 ultra-low-power features overview
- Reference manual: STM32L4x6 advanced ARM®-based 32-bit MCUs
- Microchip RN42/RN42N datasheet
- Microchip Roving Networks Bluetooth Data Module Command Reference & Advanced Information User's Guide
- LSM6DS0, iNEMO inertial module:3D accelerometer and 3D gyroscope datasheet
- MAXIM1551/1555 datasheet
- MAXIM1759 datasheet
- Sensor fusion for motion processing and visualization, A. Baharev, TAMOP Budapest
- The Balance Filter: A Simple Solution for Integrating Accelerometer and Gyroscope Measurements for a Balancing Platform, S. Colton, MIT
- Wireless Hand Sensor: Rotational Tracking to Control a Computer Mouse, M. Sarker, J. Muhlestein, A. Bilbaeno, Oregon State University
- www.wikipedia.com
- https://developer.mbed.org/
- https://developer.mbed.org/platforms/ST-Nucleo-L476RG/
- STM32 Nucleo-64 boards User manual UM1724
- Getting started with motion MEMS and environmental sensor expansion board for STM32 Nucleo: User manual UM1820
- Cadence OrCAD Flow Tutorial
- Complete PCB Design using OrCAD Capture and Layout, Kraig Mitzner, Elsevier
- http://community.cadence.com/
- FlowCAD Webinar OrCAD/Allegro PCB Editor Tipps und Tricks
- https://www.youtube.com/user/ArtedasItaliaSrl/
- https://www.youtube.com/user/parsysEDA