

Chapter 4

Design Study

*Surf it, scroll it, pause it, click it,
cross it, crack it, switch - update it
name it, rate it, tune it, print it,
scan it, send it, fax - rename it
Touch it, bring it, pay it, watch it.*

Thomas & Guy-Manuel, Daft Punk

In this chapter the platform design is described. Design decisions, specifications, schemes and needed calculations will be exposed. Detailed characteristics of any chosen component, including microcontroller, radio interfaces, power-supply options will be provided.

4.1. Global Hardware Description

Attending to the already described requirements and the conclusions obtained from the FCD review, the following decisions have been taken before proceeding with the design process.

The whole system is thought as a teaser. Consisting on different fittable modules giving flexibility to the scheme. Main module, cNGD, is the cornerstone. cNGD supposes the main platform of the design, hosting the power supply system, possibilities to embed up to three RIs, a control core unit, and other interfacing possibilities studied further on. As part of this modular design, the interfaces cNGD hosts open possibilities to new extension board implementations through a pair of headers. These attachable boards, following the popular way these such of devices are called, will be referred as shields.

Offering the chance of embedding up to three RIs, the device could access three different frequency bands over the spectrum, fulfilling one of the desired requirements described in section 3.5. The chosen operation frequencies in our design are 434 MHz, 868 MHz, and 2.4 GHz, coupling the ISM bands in most part of Europe. These bands are preferred for WSN due to the data-rate, transmission power, and transmission range parameters they offer for a free cost. This configuration places the device on the bound of complexity and cost. At the same time, this feature is essential for a CWSN proper investigation, giving radio communication opportunities not provided by any other similar device so far. Moreover, a full exploitation of the firmware described in 3.3 can be done, since it offers an integrated MiWiTM stack for three RIs.

The three RI are based on the IEEE 802.15.4 standard for WPANs, commonly used in WSN. Specifically, the set of interfaces will operate under MiWiTM or MiWiTMP2P protocol. They both are proprietary protocols designed by Microchip Technology that uses small, low-power digital radios. They are open source option designed for low data transmission rates (up to 250Kbit/s) and short distance (up to 100 without obstacles), cost constrained networks. Main difference between MiWiTM and ZigBeeTM¹ is complexity. MiWiTM offers a much more simple operation, resulting in a lighter implementation. The size required for the MiWiTM stack is 3-10 KB on the ROM depending on the node role while ZigBeeTM takes 20-40 KB. MiWiTM is free of cost and it does not require for licenses acquisition as long as Microchip components are used, in fact Microchip requires the use of MiWiTM for its products. The global design becomes simpler and more homogeneous sharing a common communication protocol for all the RIs.

Since there were not found compatible and suitable size commercial options for 434 MHz band, a custom full implementation of this interface was required. This implementation is based on MRF49XA transceiver, which can operate also at 868 MHz. In order to provide homogeneity, and since it does not suppose a great extra effort, the new RI design will be used for both 434 and 868 MHz bands. These RI will be called μ Trans434/868 from now on. μ Trans434/868 full description is taken at section 4.4. Option at 2.4 GHz is a MRF24J40MA module.

For this work a pair of shields were designed. A simple serial communication complement that links one UART and a classic rs232 interface over the so-called rs232SHIELD. This interface would facilitate the software development since the firmware just allowed debugging through the UART at first and it is described in section 4.6. A battery charger Ni-Mh and Ni-Cd was also designed and implemented as a utility for portability options. The device was called as chargerSHIELD.

Therefore, the set of designs to carry out, showed at Figure 4.1, includes μ Trans 434 and 868, cNGD, rs232SHIELD and chargerSHIELD. All hardware design files can be freely accessed at a public GitHub² repository:
<https://github.com/agus-xyz/cNGD-hard>.

4.2. Tools

4.2.1. Altium Designer

For the whole designing task, a EDA (*Electronic Design Automation*)³ software will be use. Altium Designer, being one of the most important EDA softwares, is commonly used by researchers at LSI. All the cNGD design process is carried through Altium Designer on its version 10.

Altium Designer is software package for printed circuit board, FPGA and embedded software design, and associated library and release management automation. It is developed and marketed by Altium Limited of Australia.

Despite the amount of functions offered by Altium Designer the two main options needed for this project are schematic capture, for circuit edition, and PCB design, for

¹A very popular wireless IEEE 802.15.4 compatible protocol employed at WSNs.

²Web-based hosting service for software development projects that use the Git revision control system

³Electronic design automation is a category of software tools for designing electronic systems such as printed circuit boards and integrated circuits.

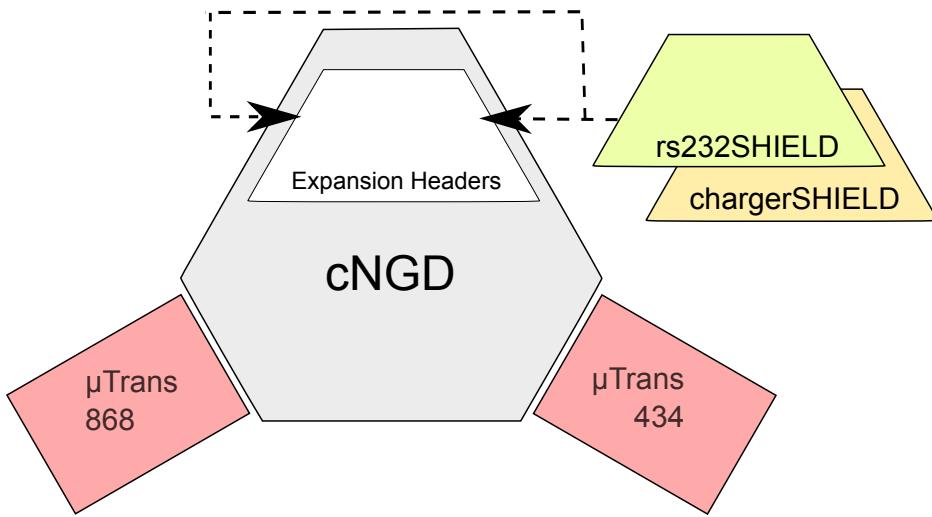


Figure 4.1: Global hardware modules of the platform to develop.



Figure 4.2: Altium Designer logo

subsequent layout deployment.

Options provided by the schematic capture module, required for the design task, includes:

- *Component library management.* Useful to create own components facing the logic design.
- *SPICE (Simulation Program with Integrated Circuits Emphasis) mixed-signal circuit simulation.* In order to simulate any functionality. Specially useful in analog modules design.
- *Netlist export.* This way, it is possible to export our designs easily to process under other softwares.
- *Reporting and BoM (Bill of Materials) facilities.* For simple management purposes.
- *Multi-channel, hierarchical schematics and design re-use.* Facilitating the design task.

On the other hand, required option from the PCB design module are:

- *Component footprint library management.* Equally useful to manage new components and embed them into our PCB designs.
- *Component placement.* In order to freely design our own layouts.

- *Manual trace routing, with support for multi-trace routing.* Enabling and facilitating the layout design.
- *Interactive 3D editing of the board and MCAD export to STEP.* To obtain more realistic views and complete idea of our implementations on advance to their manufacturing.
- *Manufacturing files generation with support for Gerber formats.* Essential for manufacturing process.

4.3. Components Library: cngd lib

For the accomplish the full design of the platform over Altium Designer, a component library with all the required but not found models was set all along the design process.

As said, the library includes models from all those components needed but not found as well as modifications of existing models in order to suit the design criteria. Every contained model embeds schematic, footprint and 3D model. 3D models are useful for 3D rendering and proper hardware fitting checking purposes. Electric models of components are not generally included.

Library is available at the mentioned repository hosting the hardware design files. All the content included into the library is ready to be used under Altium Designer. Any use of the library is free.



Figure 4.3: Footprints specially designed for the platform.

4.4. Transceivers - μ Trans 434/868

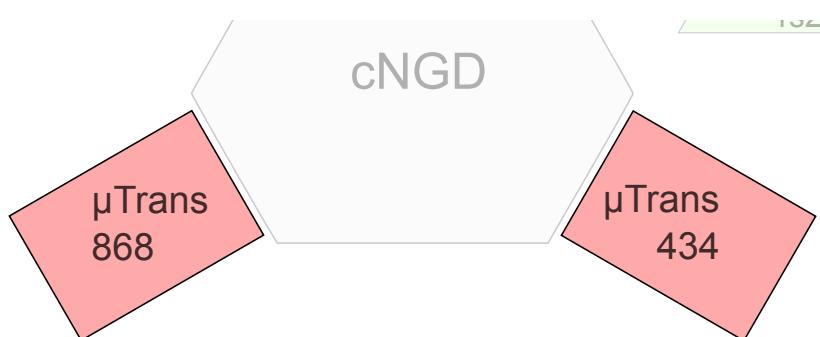


Figure 4.4: μ Trans detail over global hardware modules to develop.

The chosen firmware for our platform gives support for MRF49XA modules over 434 MHz and 868 MHz bands. Specifically, the modules used during the firmware development were the MRF49XA PICtailTM Daughter Board[97]. This board is a demonstration and development daughter board for the MRF49XA ISM Band Sub-GHz RF Transceiver. The board can plug into a fittable header, which makes it unsuitable for a reduced size and robust WSN design. It either accepts connection to external antennas, which is a desirable requirement.

Previous to the cNGD designing, which will embeds the μ Trans 434/868, the design of these RI was needed.

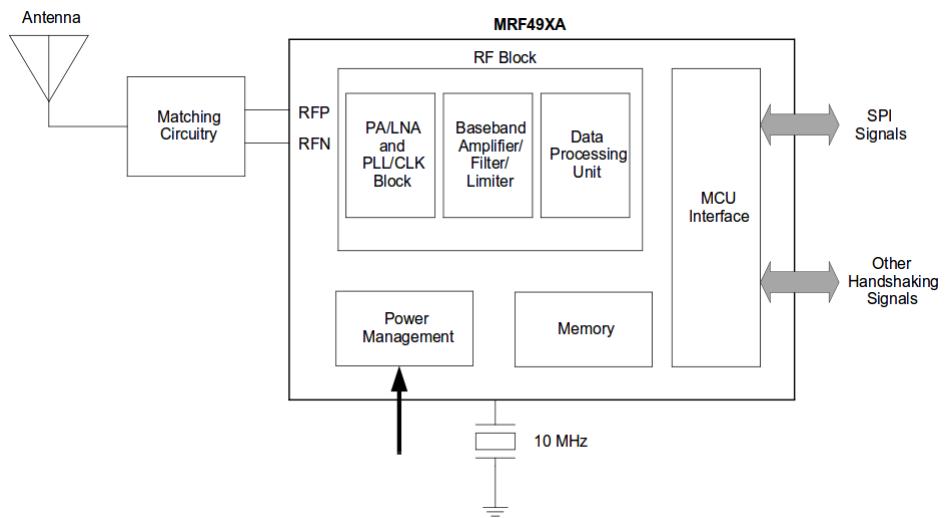


Figure 4.5: μ Trans 434/868 blocks diagram

4.4.1. Description

MRF49XA is a low-power fully integrated Sub-GHz RF transceiver. An ideal choice for low-cost, high-volume, low data rate (<256 Kbps), two-way, short range wireless applications. It can operate in the unlicensed 434, 868 and 915 MHz frequency bands. The transceiver is integrated with different sleep modes and an internal wake-up timer to reduce the overall current consumption. The device operates in the low-voltage range of 2.2V to 3.8V, and in Sleep mode, it operates at a very low-current state, typically 0.3 μ A.

Further details are given below:

- Modulation Technique: FSK (*Frequency-Shift Keying*) with FHSS (*Frequency Hopping Spread Spectrum*) capability.
- Differential RF Input/Output:
 - -110 dBm Typical sensitivity with 0 dBm maximum input level.
 - +7 dBm Typical transmit output power.
- Maximum supported data rates:
 - Digital mode 115.2 Kbps.

- Analog mode 256 Kbps.
- Current Consumption:
 - TX. 434MHz, I = 23 mA at 7 dBm. 868MHz, I = 24 mA at 5 dBm.
 - RX. 433MHz, I = 11 mA. 868MHz, I = 12 mA.
 - Idle. 1.2 mA (Max).
- Temperature range of operation from -40° to 85°.
- Other functionalities: DQI (*Data Quality Indicator*) data, RSSI data.

This transceiver is not IEEE 802.15.4 compatible since the number of channels it provides depends on the configured bit-rate, whereas the standard sets a fix number of channels. MRF49XA supports Microchip proprietary sub-GHz wireless protocols.

Due to the RSSI and DQI indicators, the transceiver is able to carry out ED (*Energy Detect*) scans over the supported frequencies to operate on the least-noisy channel. This sensing capability is crucial for CR tasks.

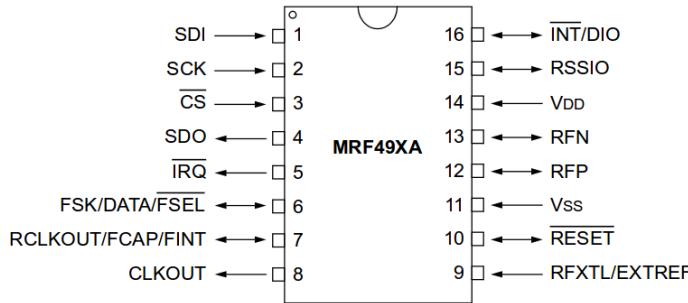


Figure 4.6: MRF49XA pin diagram

Figure 4.6 shows the MRF49XA pin diagram, a full description of each pin is given here:

- 1. *SDI*. Digital Input. Serial data input interface to MRF49XA (SPI input signal).
- 2. *SCK*. Digital Input. Serial clock interface (SPI clock).
- 3. *nCS*. Digital Input. Serial interface chip select (SPI chip/device select).
- 4. *SDO*. Digital Output. Serial data output interface from MRF49XA (SPI output signal).
- 5. *IRQ*. Digital Output. Interrupt Request Output: Receiver generates an active-low interrupt request for the microcontroller on the terminated events. Some of this events are:
 - Data-flow control registers notifications.
 - Negative pulse on interrupt input pin.
 - Wake-up timer time-out.
 - Supply voltage below the programmed value is detected.

- Power-on Reset.
- 6. *FSEL*. Digital Input/Output. FIFO (*First-in First-out*) Select: Selects the FIFO and the first bit appears on the next clock when reading the Receiver FIFO Read Register. The FSEL pin has an internal pull-up resistor. This pin must be “high” when the TX register is enabled. In order to achieve minimum current consumption, keep this pin “high” in Sleep mode. This pin could provide other functionalities not used for our design.
- 7. *FINT*. Digital Input/Output. FIFO Interrupt: When the FIFO enable bit of General Configuration Register is enabled, this pin acts as a FIFO full interrupt, indicating that the FIFO has been filled to its programmed limit. This pin could provide other functionalities not used for our design.
- 8. *CLKOUT*. Digital Output. Clock Output: The transceiver’s clock output can be used by the host microcontroller as a clock source.
- 9. *RFXTL*. Analog Input. RF Crystal: This pin is connected to a 10 MHz series crystal or to an external oscillator reference. The crystal is used as a reference for the PLL (*Phase-locked loop*) which generates the local oscillator frequency. It is possible to “pull” the crystal to the accurate frequency by changing the load capacitor value.
- 10. *nRESET*. Digital Input/Output. Active-low hardware pin. This pin has an open-drain Reset output with internal pull-up and input buffer.
- 11. *Vss*. Ground. Ground reference.
- 12. *RFP*. RF. Input/Output Differential RF input/output (+).
- 13. *RFN*. RF. Input/Output Differential RF input/output (-).
- 14. *VDD*. Power. RF power supply. Bypass with a capacitor close to the pin.
- 15. *RSSIO*. Analog Input/Output. Received Signal Strength. Indicator Output: The analog RSSI output is used to determine the signal strength. The response and settling time depends on the external filter capacitor.
- 16. *nINT*. Digital Input/Output. Interrupt: This pin can be configured as an active-low external interrupt to the device. If a logic '0' is applied to this pin, it causes the IRO pin to toggle, signaling an interrupt to the external microcontroller. The source of interrupt can be determined by reading the first four bits of STSREG. This pin can be used to wake-up the device from Sleep. This pin could provide other functionalities not used for our design.

An internal transmit/receive switch combines the transmitter and receiver circuits into differential RFP and RFN pins. These pins are connected to the impedance matching circuitry (balun) and to the external antenna connected to the device.

The quality of the data is checked or validated using the RSSI and DQI blocks built into the transceiver. Data is buffered in transmitter registers and receiver FIFOs. The Automatic Frequency Control feature allows the use of a low-accuracy and low-cost crystal. The transceiver is controlled via a 4-wire SPI, interrupt (INT and IRO), FSEL, FINT and RESET pins.

The complete design of the μ Trans 434/868 is based on the implementation guidelines provided by Microchip at MRF49XA datasheet. The only difference between 434 MHz and 868 MHz version of the μ Trans is the impedance matching circuitry facing the external antenna. The components composing it are different for each frequency band, values are specified at table A.1 placed at appendix A.

Current is bypassed through a capacitor and supplies the MRF49XA, a power indication LED (*Light-Emitting Diode*) is included in the design. A 10 MHz clock carries out the timing tasks. An impedance matching circuitry composed by inductors and capacitors interfaces the MRF49XA and the antenna. The RI hosts a 50Ω connector to which connect the corresponding antenna. 434 and 868 versions of the μ Trans are designed to attach 434 MHz and 868 MHz external monopole antennas respectively.

The designed module allows access to some of the MRF49XA pins, not all of them. The ports accessible at the RI are Interrupt (nINT), Interrupt Request Output, SPI ports, FIFO select, FIFO interrupt, and reset. These ports enable full functionality of the MRF49XA and rest of available pins are found dispensable.

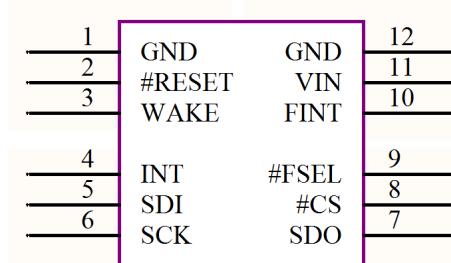


Figure 4.7: μ Trans 434/868 component model for use

4.4.2. Schematics

Schematic of μ Trans 434/868 can be found on Appendix A at Figure A.1. Table A.1 contains the different values for the balun circuit at 434 MHz and 868 MHz.

4.5. Main Board - cognitiveNextGenerationDevice

4.5.1. Description

As it was said, cNGD supposes the main part of the design, it hosts main capabilities for computation, power supply, and connectivity for the platform. It has been designed seeking simplicity and functionality.

The computation unit is located at the microcontroller, taking control over the rest of modules and running the software. This microcontroller can be programmed through its PGE port thanks to a PGE module that interfaces it.

As part of the main goal, the firmware reviewed in section 3.3 will be used. It supposes an approach to the requirements achievement since it gives support for three interfaces, it supposes a reduction on the computation load and therefore, on the needed resources and power consumption. Moreover it will facilitate the development, offering a stable firmware version to work over and avoiding software development

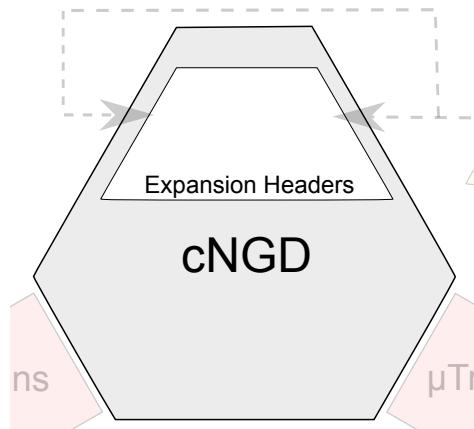


Figure 4.8: cNGD detail over global hardware modules to develop.

tasks. This fact will force the design to keep one architecture compatible with the firmware, hence a 32-bit Microchip PIC is required as microcontroller.

Communication and control over the RIIs is made through SPIs and a few more control signal described further on.

Power supply will be able to take place through three ways, a μ USB, a block terminal and a DC terminal. Two first options will take a 5V supply, whereas the third one will take 3.3V. This last option is thought to be supplied by a battery. μ USB option serves as serial communication chance, and since USB provides 5V power supply, this is employed. A second 5V supplying connector opens possibilities to not compulsory useUSB. A software-driven power supply system to the RIIs allows the application to control the power supply to these modules.

The device offers serial communication through the already named μ USB options. Moreover, it gives access to different peripherals through a pair of 20-pin headers. The list of approachable peripherals and pins covers battery connection (for charging purposes), GPIO (*general purpose input-output*s), MCLR (*master clear*) pin, external interruptions, analogue inputs, USB, Ethernet module, I2C bus, UARTs and SPI. These options give flexibility, modularity and versatility to the device. It provides possibilities for multiple kind of applications and open the design to new implementations and extensions for a particular applications. In addition, three LEDs and two push buttons give chance to the user to control the application if required.

4.5.2. Schematics

Schematics can be found all together on Appendix A for better convenience. Figure A.2 gathers all the modules into one global scheme. Figure A.3 shows the MCU connections schematic. Figures A.4, A.5, and A.6 focus on 434MHz, 868MHz and 2.4GHz RIIs, respectively. Figure A.7 gives the schematic of the power supply system. Expansion headers schematic is on Figure A.8, whereas Figure A.9, and A.10 illustrate the oscillator and programmer modules.

4.5.3. Microcontroller

Core unit is a PIC32MX675F256L[88], a mid-range 100-pin microcontroller from the mentioned 32-bit PIC family. Being the simplest of the options meeting memory

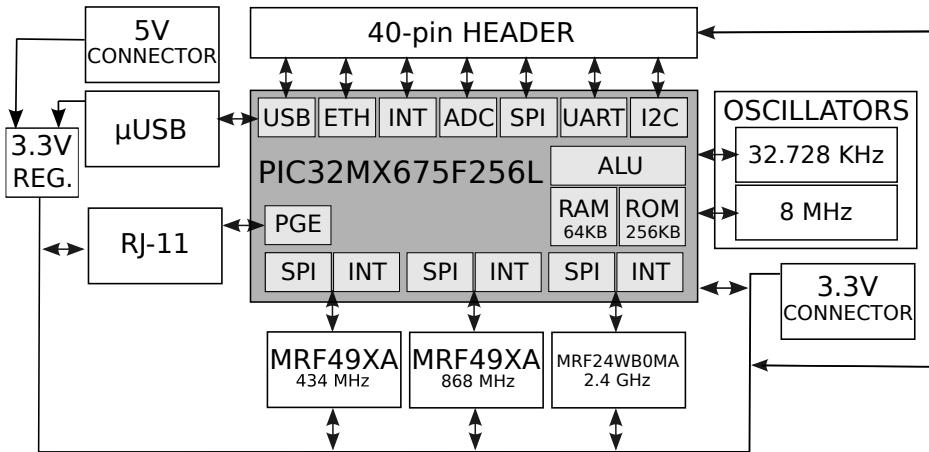


Figure 4.9: Architecture model of the cNGD.

performance and peripherals minimum requirements (Table 3.1). On the other hand, its 100 pins allow a better access of the peripherals provided. A larger memory version of this, PIC32MX675F512L, is pin compatible to the firstly mentioned. This fact gives the chance to swap among these two MCUs if larger memory was needed. Main technical features follow:

- Operating Voltage Range: 2.3V to 3.6V.
- 256/512 KB Flash (PIC32MX675F256L/PIC32MX675F512L). 64KB RAM.
- RUN, IDLE, and SLEEP modes. Multiple switchable clock modes for each power mode. Maximum speed is 80 MHz.
- Serial Communication Modules allow flexible UART/SPI/I2C configuration. 6 UARTs, 4 SPIs, and 5 I2Cs included.
- USB 2.0 On-The-Go peripheral with integrated PHY.
- 10/100 Ethernet MAC with MII (*Media Independent Interface*)/RMII (*Reduced Media Independent Interface*)⁴ interfaces.
- 2 wire programming and debugging interface.
- Five 16-bit Digital Timers embedded.
- 16 channel 10-bit ADC.

Figure 4.10 shows a block diagram of the PIC32 family. Distinct 32-bit buses for peripherals and system can be observed. Moreover, DMA (*Direct Memory Access*), RAM, and flash controllers, together with external modules take place on it.

The whole microcontroller can be divided into four blocks: core, memory, integration system and peripherals:

- The main core is based on a RISC MIPS M4K. This module takes control over the system. It controls the peripherals and execute the programmed code.

⁴Media Independent Interface and Reduced Media Independent Interface are standard interfaces used to connect a Fast Ethernet MAC-block to a PHY chip.

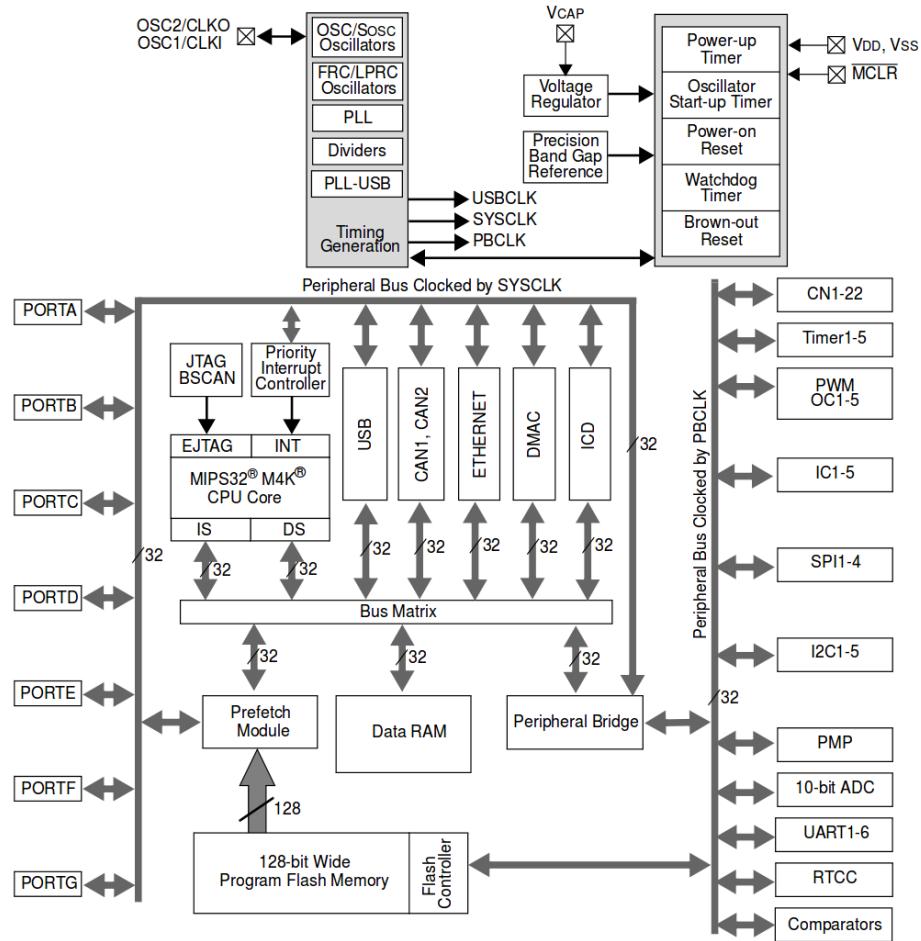


Figure 4.10: PIC32 family block diagram

- A further observation into the memory system allows differentiate among Flash, RAM, registers, and boot flash. The system load the data on a cache memory and prefetch the instructions. Figure 4.11 shows the block diagram of the prefetch module.
- Integration system is compound by the set of modules needed to make work the core and peripherals. These are interrupt control, oscillator system, reset, watchdog timer and power-up timer. For detailed explanation of each module consult the microcontroller manual [88].
- Peripherals are a set of independent modules oriented to perform specific tasks. Here it is described a brief description of the main peripherals and the use they receive in the system:
 - **SPI**. The SPI module is a synchronous serial interface that is useful for communicating with external peripherals and other microcontroller devices. These peripheral devices may be serial EEPROM (*Electrically Erasable Programmable Read-Only Memory*), shift registers, display drivers, etc. At cNGD, three SPIs are used to set communication with the RIIs. Another SPI is outputted to the expansion headers for general purposes.

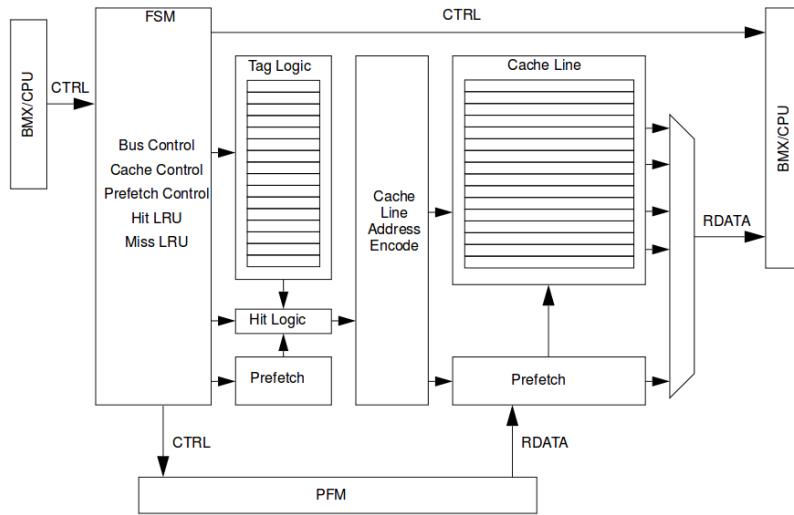


Figure 4.11: Prefetch cache module block diagram.

- **UART.** This peripheral is a serial I/O module. The UART is a full-duplex, asynchronous communication channel that communicates with peripheral devices and personal computers through protocols, such as RS-232, RS-485, LIN 1.2 and IrDA (*Infrared Data Association*).

Two UART modules are available at the expansion headers. RS232Shield described at section 4.6 use one of this modules to set an RS-232 serial interface.

- **USB OTG (On-The-Go).** The USB module contains analog and digital components to provide a USB 2.0 full-speed and low-speed embedded Host, full-speed Device or OTG implementation with a minimum of external components.

This USB peripheral is used to establish serial communication with other devices, cNGD it thought to include this serial communication way. Its ease of use, popularity, makes it a good choice for this purpose. Moreover, it is interfaced into the expansion headers for other future uses.

- **I^2C .** The I^2C module provides complete hardware support for both Slave and Multi-Master modes of the I^2C serial communication standard. This module is commonly used to establish communication among sensors and microcontroller. One I^2C can be accessed over the expansion headers.

- **TIMERS.** Four synchronous, one asynchronous, 16-bit timers that can operate as a free-running interval timer for various timing applications and counting external events.

This timers, properly configured, will drive in time the firmware operation. This includes, pretty much, all the tasks carried out by the node. Timers also must be available to drive in time the application if was required.

- **ADC.** It converts a continuous voltage quantity to a digital number that represents the quantity's amplitude.

Useful when dealing with sensors offering an analog output signal. The ADC included into the microcontroller can be accessed from the expansion headers.

- **ETHERNET**. The Ethernet controller is a bus master module that interfaces with an off-chip Physical Layer (PHY) to implement a complete Ethernet node in a system.

This module is accessible at the headers. It is essential to establish 802.11 communications, which is a desirable feature for WSN coordinator nodes or gateways.

- **GPIO**. General purpose I/O pins are the simplest of peripherals. They allow the PIC32 MCU to monitor and control other devices. Basically, almost every pin at the MCU can act as a GPIO if no other peripherals are active and multiplexed on such pin.

Several GPIOs can be accessed over the headers for general purposes.

- **RTCC** (Real-Time Clock/Calendar). It is intended for applications in which accurate time must be maintained for extended periods of time with minimal or no CPU intervention. Low-power optimization provides extended battery lifetime while keeping track of time.

This module is essential for accurate timing tasks, which are common in WSN for synchronous processes over the network. Synchronization helps saving energy since it allows better efficient communication and general operation.

4.5.3.1. Pin-out

Here is described the multiplexed functionalities available on each pin at the MCU. Functionalities emphasized in bold are functionalities available for use, not emphasized functionalities are not usable with the current design. In addition, module where the pin is used and linked signal are exposed. For further details about peripherals and pins description see the PIC32 family reference manual[88].

PIN	PERIPHERALS	MODULE	USE
1	AERXARR/RG15	-	-
2	VDD	Power Supply	
3	PMD5/ RE5	Header	p23
4	PMD6/ RE6	Header	p35
5	PMD7/ RE7	Header	p22
6	T2CK/ RC1	Header	p36
7	T3CK/RC2	-	-
8	T4CK/RC3	-	-
9	T5CK/ SDI1 /RC4	RI 434MHz	SPI SDO
10	ECOL/ SCK2/U6TX/U3nRTS/PMA5/CN8/RG6	Header	p37
11	ECRS/ SDA4/SDI2/U3RX/PMA4/CN9/RG7	Header	p21
12	ECRSDV/SCL4/SDO2/U3TX/PMA3/CN10/RG8	Header	p38
13	nMCLR	PGE/Pushbutton	Reset
14	ERECLK/nSS2A/U3nCTS/U6RX/PMA2/CN11/RG9	Header	p20
15	VSS	Ground	
16	VDD	Power Supply	
17	TMS/RA0	-	-
18	AERXD0/ INT1 /RE8	RI 434MHz	Interrup.
19	AERXD1/ INT2 / RE9	Header	p00
20	AN5/C1IN+/VBUSON/CN7/RB5	Header	p01
21	AN4/C1IN-/CN6/RB4	Header	p39
22	AN3/C2IN+/CN4/RB3	Header	p02
23	AN2/C2IN-/CN3/RB2	-	-

Follows on next page.

PIN	PERIPHERALS	MODULE	USE
24	PGEC1/AN1/CN2/RB1	PGE	PGEC
25	PGED1/AN0/CN1/RB0	PGE	PGED
26	PGEC2/AN6/OCFA/RB6	-	-
27	PGED2/AN7/RB7	RI 434MHz	Reset
28	VREF-/CVREF-/AERXD2/PMA7/RA9	Header	p18
29	VREF+/CVREF+/AERXD3/PMA6/RA10	Header	p19
30	AVDD	Power Supply	
31	AVSS	Ground	
32	AN8/C1OUT/RB8	RI 434MHz	Wake
33	AN9/C2OUT/RB9	RI 434MHz	Power Switch
34	AN10/CVREFOUT/PMA13/ RB10	RI 434MHz	Chip Select
35	AN11/ERXERR/AETXERR/PMA12/RB11	Header	p03
36	VSS	Ground	
37	VDD	Power Supply	
38	TCK/RA1	-	-
39	SCK4/U5TX/U2nRTS/RF13	RI 2.4GHz	SPI SCK
40	nSS34/U5RX/U2nCTS/ RF12	RI 868MHz	FIFO Int.
41	AN12/ERXD0/AEGRS/PMA11/RB12	Header	p04
42	AN13/ERXD1/AECOL/PMA10/RB13	Header	p05
43	AN14/ERXD2/AETXD3/PMALH/PMA1/RB14	Header	p17
44	AN15/ERXD3/OCFB/PMALL/PMA0/CN12/RB15	Header	p16
45	VSS	Power Supply	
46	VDD	Power Supply	
47	AETXD0/nSS3/U4RX/U1nCTS/CN20/ RD14	RI 868MHz	FIFO Sel.
48	AETXD1/ SCK3/U4TX/U1nRTS/CN21/RD15	RI 868MHz	SPI SCK
49	SDA5/ SDI4/U2RX/PMA9/CN17/RF4	RI 2.4GHz	SPI SDO
50	SCL5/ SDO4/U2TX/PMA8/CN18/RF5	RI 2.4GHz	SPI SDI
51	USBID/RF3	USB/Header	USB ID/p06
52	SDA3/ SDI3/U1RX/RF2	RI 868MHz	SPI SDO
53	SCL3/ SDO3/U1TX/RF8	RI 868MHz	SPI SDI
54	VBUS	USB/Header	VBUS/p07
55	VUSB	Power Supply + Cap	
56	D-/RG3	USB/Header	D-/p13
57	D+/RG2	USB/Header	D+/p12
58	SCL2/RA2	Header	p11
59	SDA2/RA3	Header	p10
60	TDI/RA4	LEDS	LED 3
61	TDO/RA5	LEDS	LED 2
62	VDD	Power Supply	
63	OSC1/CLK1/RC12	Oscillators	Prim. Osc. In.
64	OSC2/CLK0/RC15	Oscillators	Prim. Osc. Out.
65	VSS	Ground	
66	AETXCLK/SCL1/ INT3/RA14	RI 868MHz	Interrup.
67	AETXEN/SDA1/ INT4/RA15	RI 2.4GHz	Interrup.
68	RTCC/EMDIO/AEMDIO/IC1/RD8	Header	p09
69	nSS1/ IC2/RD9	Header	p32
70	SCK1/IC3/PMCS2/PMA15/RD10	RI 434MHz	SPI SCK
71	EMDC/AEMDC/IC4/PMCS1/PMA14/RD11	Header	p27
72	SDO1/OC1/INT0/RD0	RI 434MHz	SPI SDI
73	SOSCI/CN1/RC13	Oscillators	Sec. Osc. In.
74	SOSCO/T1CK/CN0/RC14	Oscillators	Sec. Osc. Out.
75	VSS	Ground	
76	OC2/RD1	RI 868MHz	Reset
77	OC3/RD2	RI 868MHz	Wake
78	OC4/RD3	RI 868MHz	Power Switch
79	ETXD2/IC5/PMD12/RD12	-	-
80	ETXD3/PMD13/CN19/RD13	-	-

Follows on next page.

PIN	PERIPHERALS	MODULE	USE
81	OC5/PMWR/CN13/ RD4	RI 868MHz	Chip Select
82	PMRD/ CN14/RD5	Push buttons	S2
83	ETXEN/PMD14/CN15/RD6	Header	p33
84	ETXCLK/PMD15/ CN16/RD7	Push buttons	S3
85	VCAP/VDDCORE	Capacitor	
86	VDD	Power Supply	
87	ETXD1/PMD11/RF0	Header	p26
88	ETXD0/PMD10/RF1	Header	p25
89	ETXERR/PMD9/RG1	-	-
90	PMD8/RG0	-	-
91	TRCLK/ RA6	LEDS	LED1
92	TRD3/RA7	-	-
93	PMD0/ RE0	RI 2.4GHz	Reset
94	PMD1/ RE1	RI 2.4GHz	Wake
95	TRD2/ RG14	RI 434MHz	FIFO Sel.
96	TRD1/RG12	-	-
97	TRD0/ RG13	RI 434MHz	FIFO Int.
98	PMD2/ RE2	RI 2.4GHz	Power Switch
99	PMD3/ RE3	RI 2.4GHz	Chip Select
100	PMD4/RE4	-	-

Table 4.1: PIC32675F256L pins use description.

4.5.4. Radio Interfaces

As it was mentioned, cNGD will be able to include up to three RIs, with three consequent frequency bands to access. This fact gives a flexibility over radio communications never offered by other WSN device. Frequencies to operate over are 434 MHz, 868 MHz, and 2.4 GHz, meeting desired features.

For communication on 434 MHz and 868 MHz, ad-hoc designed RIs described in Section 4.4 are used. These RIs are based on the MRFX49XA radio transceiver and a full description of them can be found at such section. These RIs share common features since the radio transceiver module is identical for both. This brings simplicity to the system. Interface between these RIs and MCU, illustrated at Figure 4.12, are the same.

Figure A.6 shows the scheme for the RI at 2.4 GHz. This is based on a MRF24J40MA[90]. This RI was already used at the FCD and it is supported by the firmware. It is a MiWiTM transceiver also compatible with the IEEE 802.15.4 standard, this homogeneity with the other two RIs supposes computation, cost, and power savings. It has access over the 15 5MHz-channels, the protocol enables and employs a QPSK (*Quadrature Phase-Shift Keying*) modulation. Figure 4.13 shows the MRF24J40MA block diagram. Further details about the MRF24J40MA RI follow:

- Operating Voltage: 2.4-3.6V (3.3V typical).
- Up to 120 meters range.
- Differential RF Input/Output:
 - -94 dBm Typical sensitivity with +5 dBm maximum input level.
 - +0 dBm Typical output power.
- Maximum supported data rates:

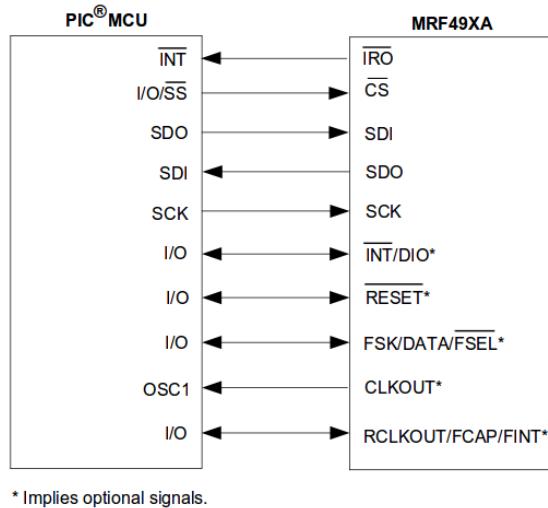


Figure 4.12: Microcontroller to MRF49XA interface

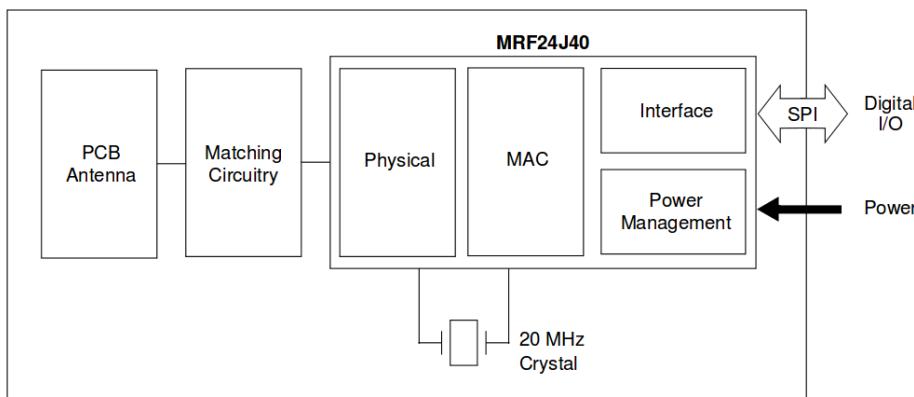


Figure 4.13: MRF24J40MA block diagram.

- 250 Kbps.
- Turbo mode enables up to 625 Kbps (not following the standard).
- Current Consumption:
 - TX. I = 23 mA (typical).
 - RX. 19 mA (typical).
 - Sleep. $2\mu\text{A}$ (typical).
- Integrated crystal, internal voltage regulator, matching circuitry, PCB antenna, and RSSI detector.

Interface between MRF24J40MA RI is driven by 7 signals, figure 4.14 illustrates it. Serial communication is carried through a SPI, containing 4 signals. Two interruption signals for both senses and a reset signal complete the whole set.

All the described RIs are based on transceivers supporting either MiWiTM Mesh or MiWiTM P2P protocols. Figure 4.15 shows how the protocol environment configuration.

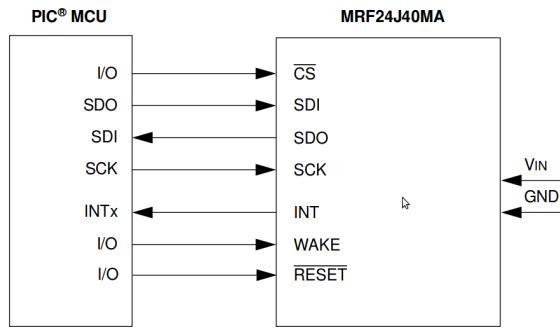
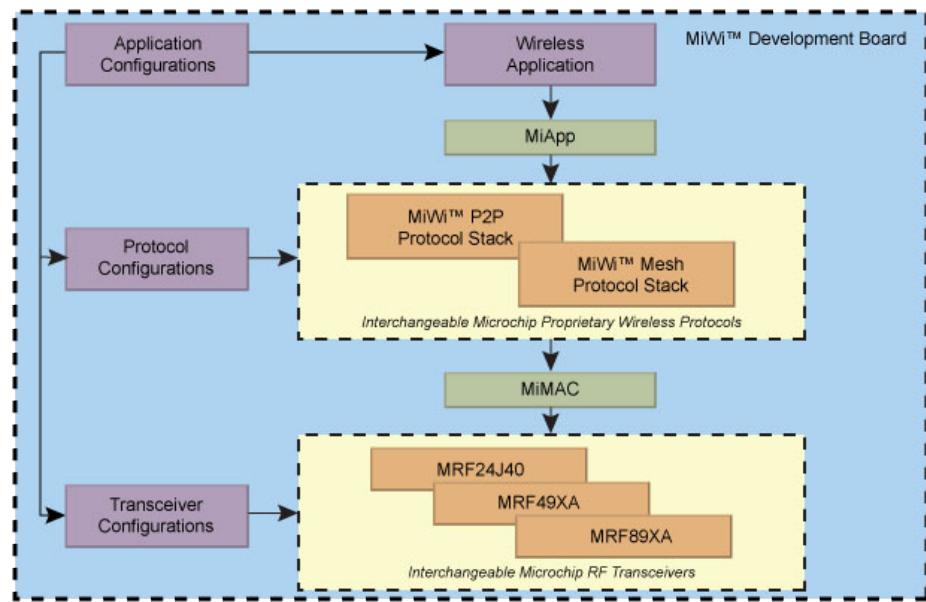


Figure 4.14: MRF24J40MA interface scheme

Figure 4.15: MiWiTM protocol configuration scheme.

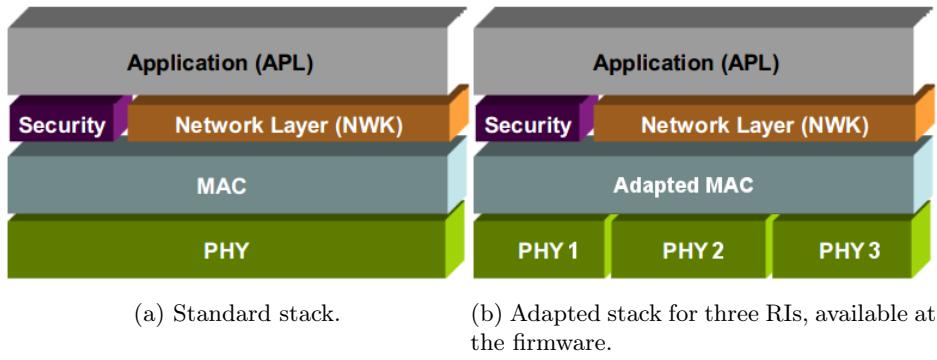
A network using the MiWiTM protocol[44] is capable of having a maximum of 1024 nodes on a network. Each coordinator is only capable of having 127 children, with a maximum of 8 coordinators in a network. Packets can travel a maximum of 4 hops in the network and 2 hops maximum from the PAN coordinator.

The MiWiTM P2P protocol[91] stack supports star and peer-to-peer wireless-network topologies, useful for simple, short-range, wireless node-to-node communication. Additionally, the stack provides sleeping-node, active-scan and energy-detect features while supporting the low-power requirements of battery-operated devices.

4.5.4.1. Software Controlled Power Supply

The designed system allows enabling/disabling the power supply to the RIs. This way, any RI can be completely switched off by the running application. Cutting down all possible power consumption that unused RIs could have by, for instance, running clocks.

The power control to the RIs is based on the ADG701[98]. This is a CMOS (*Com-*

Figure 4.16: MiWiTM Protocol.

*plementary metal oxide semiconductor) SPST (*Single pole, single throw*) switch. This switch is designed to provide low power dissipation, low on resistance and low leakage currents. It can operate from a single 1.8 V to 5.5 V supply, making it ideal for use in battery-powered instruments.*

Some of the features of this component are:

- 2 ohm (typical) on resistance.
- Fast switching times: tON = 18 ns, tOFF 12 ns.
- Typical power consumption <0.01 μ W.

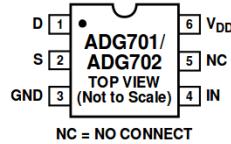


Figure 4.17: ADG701 pin diagram.

Pin 4 (IN) is a digital input where the control logic signal is placed. A logic '1' will make pins 1 (D - drain) and 2 (S - source) shorted, a logic '0' will open the switch. Pins 1 and 2 can be either inputs or outputs. Pin 5 (NC) must not be connected.

In case ADG701 switch was not included into the design for any reason, optional 0 Ω resistors must be used to enable power supply at the RIs. These optional resistors are R16, R17, and R18 for 434 MHz, 868 MHz, and 2.4 GHz RIs respectively.

4.5.5. Power Supply System

Power supply module is in charge of providing proper voltage and current to the cNGD. Chosen output voltage is 3.3V, being this a normal value for CMOS family chips. The power supply system must be able to support either wired-fixed or battery options. Figure A.7 shows a schematic of this module.

For external wired options, the input voltage is 5V. This value is widely used to supply digital circuits and it is easy to access to 5 V sources. Moreover, it couples with the USB voltage rate, being possible to supply the cNGD through a μ USB terminal

that this module includes. Apart from this USB terminal, a standard screw terminal block accepts 5V to supply the device. This 5V are lowered to 3.3V by a synchronous buck regulator. This buck regulator can be deactivated for power saving when unused just changing a jumper location.

For battery options, the module includes a DC connector. This connector accepts from 3.3V up to 3.6V. Higher voltages could damage the components since this voltage is directly injected into the circuitry to achieve higher efficiency. An included switch lets choosing the supply mode, either wired or portable.

A green LED indicates that power is applied into the circuit, on the other hand, a red LED indicates if power is supplied into the block terminal or the μ USB terminal. This way, is easily noticeable the presence of voltage and current on the cNGD and their origin input.

The employed synchronous buck regulator is a MCP1603[?] chip. This is a high-efficiency, fully-integrated 500 mA synchronous buck regulator with a 2.7V to 5.5V input voltage range. The MCP1603 is available with either an adjustable or fixed output voltage. The are several available fixed output voltage options, the chosen option here is 3.3V. When a fixed option is used, only three additional small external components are needed to form a complete solution, Figure 4.18.

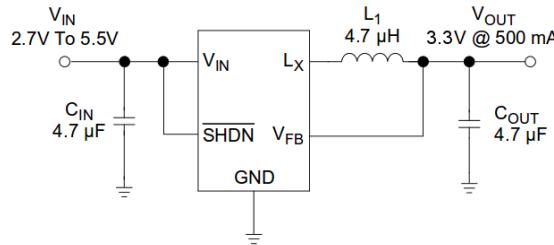


Figure 4.18: MCP1603 buck regulator configuration.

It was estimated that 500 mA was enough to supply the device regarding the described maximum consumptions. Table 4.2 gives 165 mA as maximum current consumption rate. Leaving buffer for current peaks at transmission and possible shields, a maximum current estimation of 500 mA seem reasonable and enough for the whole platform.

Module	Max. Power
PIC32MX675F256L (Run mode)	85 mA
μ Trans 434 (TX mode)	23 mA
μ Trans 868 (TX mode)	24 mA
MRF24j40MA (TX mode)	23 mA
LEDS	10 mA
TOTAL MAX CONSUMPTION	165mA

Table 4.2: Maximum estimated power consumption of the platform.

4.5.5.1. Battery

In order to give autonomy and portability to the device, a rechargeable battery system will be included. This system must keep down the global cost and size. At the same time it must offer autonomy to the device. Voltage offered by the battery system must be on the 3.3-3.6V range.

Firstly it is needed an estimation of platform global current consumption. This value will depends, relying strongly on the running application. However, we can assume some overestimated common values for our study. Considering two operation modes, Run and Sleep, and using values of current and operation times at Table 4.3, it results in an approximated global current consumption calculated as $C_{estimated}$.

Mode	Operation time percentage	Consumption
Run	10 %	120 mA
Sleep	90 %	100 μ A

Table 4.3: Estimated values for Run and Sleep operation modes.

$$C_{estimated} = 100 \times 10^{-3} \times 0,1 + 100 \times 10^{-6} \times 0,90 = 12,1 \text{ mA}$$

Table 4.4 shows a comparison between considered battery possibilities. Other technologies and sizes, not described at the table, have been directly discarded for showing unsuitable features. An example are Ni-Cd (*Nickel-Cadmium*) technology batteries, which may suffer “memory effect”⁵, or special size batteries⁶ whose price is extremely high. Possible options considered are some Ni-MH (*Nickel-Metal Hydride*)-based and Li-ion (*Lithium-Ion*) based batteries.

Technology	Cell Nominal Voltage	Min. Num. of cells	Use complexity	Size	Capacity (mHh)	Estimated autonomy (h.)	Aprox. Unit. Cost (€)
Ni-MH	1.2/1.25 V	3	Low	AAA	600 - 1250	50 - 100	1 - 2
				AA	1200 - 2900	100 - 240	1.5 - 3
Li-ion	3.6/3.7 V	1	Medium	AAA	500 - 600	40 - 50	3.5 - 4.5
				AA	900	75	4 - 6
				CR123A	700 - 2000	60 - 165	3 - 20
				CR2	600 - 800	50 - 65	3.5
				Other	up to 3000	up to 250	40 - 60

Table 4.4: Comparison among considered batteries.

Facing similar capacities, Li-ion options show a higher cost than Ni-MH even considering 3 Ni-MH cells are needed. In addition, Li-ion show difficulties for its use since Li-ion batteries must include circuitry to control discharge cycles and other security issues. Charging process is more complex as well.

The solution proposed consist of a battery consisting on 3 2900 mAh Ni-MH AA cells. This might provide up to 240 h. of autonomy considering our estimations. No more circuitry is needed for the battery except for a 3-AA battery-holder. The total cost for a complete battery pack is under 10 €.

⁵It is an effect observed in some kind of rechargeable batteries that causes them to hold less charge.

⁶For instance, 2/3 A, L-fat A, or CS sizes.

4.5.6. Timing

The microcontroller includes possibilities to properly place quartz crystals in parallel with internal amplifiers and a pair of suitable capacitors in such way they work as oscillators. The oscillator signal is then connected to the internal PLL to obtain the desired frequency. Figures 4.19 and 4.20 show the basic input scheme for primary and secondary oscillators respectively.

Primary oscillator is 8 MHz quartz crystal used as system reference clock. The respective PLL is configured to raise this frequency up to 80 MHz. It exists the possibility of replacing the functionality of this clock for a internal one, but stability and accuracy can be affected.

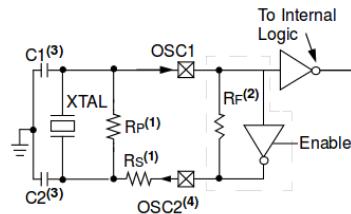


Figure 4.19: PIC32MX675F256L primary oscillator input.

In order to calculate the values for C_1 and C_2 capacitors, we will use a common formula used for crystal-based oscillator design.

$$C_L = \frac{C_1 C_2}{C_1 + C_2} + C_P + C_I$$

Where C_1 and C_2 have an equal value, C_L is the load capacitance required for the crystal, C_P is the circuit parasitic capacitance and C_I is the chip input capacitance. Typically $C_P + C_I$ is set to 5 pF. For the 8 MHz crystal, load capacitance specified by the manufacturer is 20 pF[99]. The resulting equation remains like follows:

$$20 \text{ pF} = \frac{C_1 C_2}{C_1 + C_2} + 5 \text{ pF}$$

Resolving the equation, it results that $C_1 = C_2 = 15 \text{ pF}$.

Second oscillator, a 32.728 KHz, supposes a low-power clock for real time clocking. It enables nodes synchronization, which is fundamental for power-saving modes.

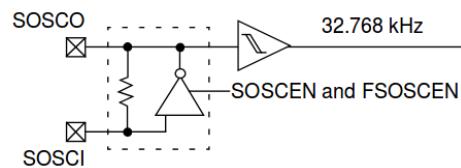


Figure 4.20: PIC32MX675F256L second oscillator input.

For this second clock, load capacitance comes to be 12.5 pF[100]. Hence the corresponding equation results as follows:

$$12.5 \text{ pF} = \frac{C_1 C_2}{C_1 + C_2} + 5 \text{ pF}$$

Resulting capacitors value are $C_1 = C_2 = 15 \text{ pF}$

4.5.7. Expansion Headers

Figure A.8 shows the schematic design for this module. Basically, two headers, containing 20 pins each, enable access to unused and useful peripherals the microcontroller includes. This offers possibilities to future implementations and expansions for the node by simply attaching a device through these headers and carrying through the corresponding firmware adaptations. Possible attachable implementations are 802.11 interfaces, data acquisition modules, external memories, extra RIIs, etc.

Here it is made a comprehensive description of the available peripherals and functionalities at the headers:

- *Power Supply*. Several pins offer power supply options:
 - VCC. Shorted to the cNGD supply system available at the platform, usually 3.3V. Pins 14, 24, and 28.
 - GND. Shorted to cNGD ground. Pins 15, 31, and 34.
 - BAT+. Directly linked to the positive pole of the battery, thought for charging purposes. Pin 29.
 - BAT-. Directly linked to the negative pole of the battery, thought for charging purposes. Pin 30.
- *nMCLR*. Allows shields to reset the platform. Pin 08.
- *SPI*. SPI 2 is available. SDI signal at pin 21, SDO at 38 and SCK at 37.
- *UART*. UARTS 3 and 6 are available.
 - UART 3. This UART supports hardware flow control. RX signal is at pin 21, TX at 38, nRTS at 37, and nCTS is at pin 20.
 - UART 6. Does not support hardware flow control. RX is at pin 20, TX at pin 37.
- *I²C*. Two I²C peripherals are accessible, I²C number 2 and 4.
 - I²C 2. SCL signal is at pin 38, and SDA signal is at pin 21.
 - I²C 4. SCL signal is at pin 11, and SDA signal is at pin 10.
- *USB*. Fully accessible. Notice the USB can not operate over the terminal at the cNGD and the headers at once. For further information consult the manual[88]. USBID signal is placed at pin 06, VBUSON at pin 07, D+ is available at pin 12, and D- is at 13.
- *Ethernet*. It is a bus master module that interfaces with an off-chip Physical layer to implement a complete Ethernet node. Headers signals provides an RMII mode interface. Signals placement follows:
 - EMDC. (Management Clock). Pin 27.
 - EMDIO. (Management I/O). Pin 09.
 - ETXEN. (Transmit Enable). Pin 33.

- ETXD0. (Transmit Data). Pin 25.
 - ETXD1. (Transmit Data). Pin 26.
 - EREFCLK. (Reference Clock). Pin 20.
 - ECRSDV. (Carrier Sense - Receive Data Valid). Pin 38.
 - ERXD0. (Receive Data). Pin 04.
 - ERXD1. (Receive Data). Pin 05.
 - ERXERR. (Receive Error). Pin 03.
- *ADC*. 9 out of 16 available ADC channels are accessible at the headers. In addition, VREF+ is at pin 19 and VREF- is at 18, these pins have optional functionalities for the ADC peripheral.
- Channel 05. Pin 01.
 - Channel 04. Pin 39.
 - Channel 03. Pin 02.
 - Channel 11. Pin 03.
 - Channel 12. Pin 04.
 - Channel 13. Pin 05.
 - Channel 14. Pin 17.
 - Channel 15. Pin 16.
- *GPIO*. A total of 30 GPIOs are available.
- Ports A2, A3, A9, and A10 pins 11, 10, 18, and 19.
 - Ports B3, B4, B5, B11, B12, B13, B14, B15 at pins 02, 39, 01, 03, 04, 05, 17, and 16.
 - Port C1 at pin 36.
 - Ports D6, D8, D9, and D11 at pins 33, 09, 32, and 27.
 - Ports E5, E6, E7, and E9 at pins 23, 35, 22, and 00.
 - Ports F0, F1, and F3 at pins 26, 25, and 51.
 - Ports G2, G3, G6, G7, G8, and G9 at pins 12, 13, 37, 21, 38, and 20.
- *Change Notification*. A total of change notification pins can be accessed. These are CN4 at pin 02, CN6 at pin 39, CN7 at pin 01, CN8 at pin 37, CN9 at pin 21, CN11 at pin 20, CN12 at pin 16, CN13 at pin 38, and CN15 at pin 33.
- *Input Capture*. Useful in applications requiring frequency and pulse measurement. Input Capture modules 1, 2 and 4 are placed at positions 09, 32 and 27.
- *Interruption*. External interruption 2 is available at pin 00.
- *Comparator*. A comparator is also available through C1IN+ at pin 01, and C1IN- at pin 39. CVREF+ and CVREF-, optional pins for the comparator, can be found at pins 18 and 19.

Notice that some of the peripherals described could be pin multiplexed and thereby, they can not be used simultaneously. For a header pin to pin description, refer to Table 4.5.

4.5.7.1. Pin-out

PIN	MCU PIN	1st TYPE	2nd TYPE	3rd TYPE	4th TYPE	5TH TYPE
00	19	INT2	RE9	-	-	-
01	20	VBUSON	AN5	C1IN+	CN7	RB5
02	22	AN3	CN4	RB3	-	-
03	35	AN11	ERXERR	RB11	-	-
04	41	AN12	ERXD0	RB12	-	-
05	42	AN13	ERXD1	RB13	-	-
06	51	USBID	RF3	-	-	-
07	54	VBUS	-	-	-	-
08	13	nMCLR	-	-	-	-
09	68	EMDIO	IC1	RD8	-	-
10	59	SDA2	RA3	-	-	-
11	58	SCL2	RA2	-	-	-
12	57	D+	RG2	-	-	-
13	56	D-	RG3	-	-	-
14	-	VCC	-	-	-	-
15	-	GND	-	-	-	-
16	44	AN15	CN12	RB15	-	-
17	43	AN14	RB14	-	-	-
18	28	VREF-	CVREF-	RA9	-	-
19	29	VREF+	CVREF+	RA10	-	-
20	14	REFCLK	U3nCTS	U6RX	CN11	RG9
21	11	SDA4	SDI2	U3RX	CN9	RG7
22	5	RE7	-	-	-	-
23	3	RE5	-	-	-	-
24	-	VCC	-	-	-	-
25	88	ETXD0	RF1	-	-	-
26	87	ETXD1	RF0	-	-	-
27	71	EMDC	IC4	RD11	-	-
28	-	VCC	-	-	-	-
29	-	BAT+	-	-	-	-
30	-	BAT-	-	-	-	-
31	-	GND	-	-	-	-
32	69	IC2	RD9	-	-	-
33	83	ETXEN	CN15	RD6	-	-
34	-	GND	-	-	-	-
35	4	RE6	-	-	-	-
36	6	RC1	-	-	-	-
37	10	SCK2	U6TX	U3nRTS	CN8	RG6
38	12	ECRSDV	SCL4	SDO2	U3TX	CN10/RG8
39	21	AN4	C1IN-	CN6	RB4	-

Table 4.5: Headers pins use description.

4.5.8. PGE

Figure A.10 contains the schematic for the programmer module, PGE. This module allows In-Circuit Serial Programming and debugging tasks over the microcontroller making use of the tools provided by Microchip. For this, PGE microcontroller ports at pins 24 and 25 interfaced through an RJ-11 connector together with power supply pins and reset signal. Figure 7.1 illustrates it.

4.5.9. Others

cNGD platform will include a push button to reset the MCU. It also includes three LEDs and two push buttons for general purposes. They allow the user, somewhat, interact with the running application if required.

- LED 1 represents the present signal at pin 91.
- LED 2 represents the present signal at pin 61.
- LED 3 represents the present signal at pin 60.
- Signal driven by push button 1 is obtained at pin 84.
- Signal driven by push button 2 is obtained at pin 82.

4.6. Serial Communication Board - rs232SHIELD

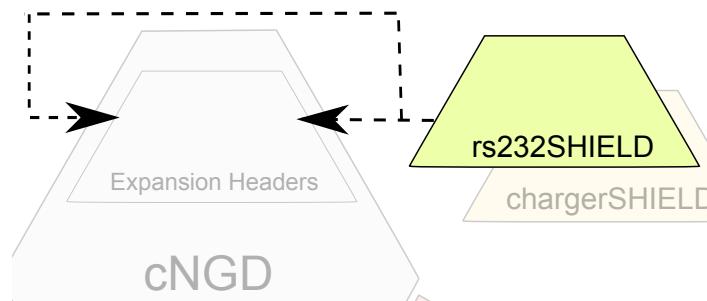


Figure 4.21: rs232SHIELD detail over global hardware modules to develop.

4.6.1. Description

This extension sets serial communication between the microcontroller and a PC over a RS-232 protocol. It supposes an alternative wired serial communication way for the included USB option. This technology has been progressively replaced by the already said USB for serial communication purposes, and it supposes an increase of the complexity, size, and consumption. That is why it was chosen not to embed this module over the cNGD, but rather create a expansion module.

This technology could be useful for old machines, low-resources researches, or for debugging purposes during the platform development until the USB communication is fully operative. This communication way has a simpler operation mode than USB.

Figure A.11 shows the schematic for this shield. UART 6 is used, accessible at the headers and not requiring hardware flow control. Since RS-232 standard operates on voltage values from -12V to 12V, a level conversion is needed. This conversion takes place at the MAX3232[101] transceiver. Figure 4.22 shows the MAX3232 scheme. TX pin of the UART is accessed at header pin 37, RX is at 20. Since the communication does not require for hardware flow control⁷, two pins are enough to set proper communication.

⁷Hardware flow control allows to control the serial data flow by using extra signals Apart from the data signals.

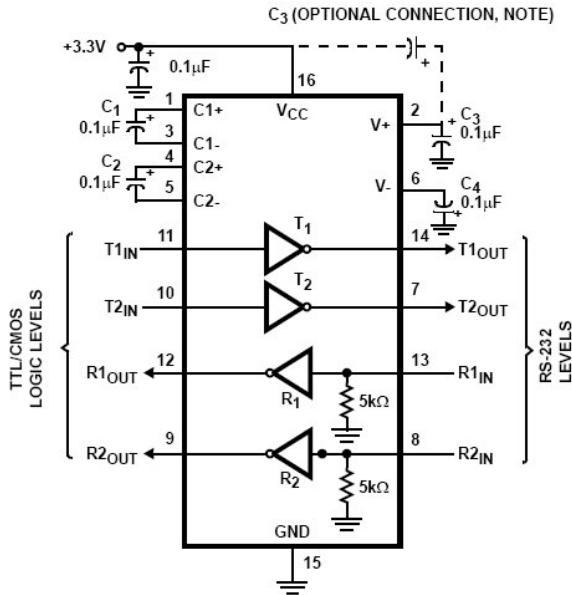


Figure 4.22: MAX3232 basic scheme.

4.6.2. Schematics

A schematic view of the rs232SHIELD is shown at Figure A.11 on Appendix A.

4.7. Batteries charger - chargerSHIELD

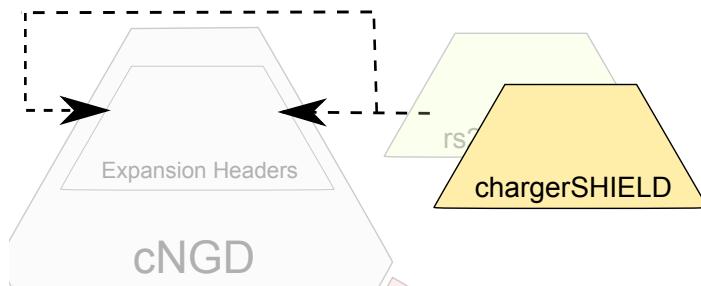


Figure 4.23: chargerSHIELD detail over global hardware modules to develop.

4.7.1. Description

Ni-MH batteries are amongst the hardest batteries to charge. Whereas with lithium ion and lead acid batteries you can control overcharge by just setting a maximum charge voltage, the nickel based batteries don't have a "float charge" voltage. So the charging is based on forcing current through the battery. What usually is called as linear charger.

Charge rate, usually denoted as C, signifies a charge or discharge rate equal to the capacity of a battery in one hour. For a 1 Ah battery, C is 1A. A charge rate of C/2 (0.5A), would need two hours, whereas a charge rate of 2C (2A), if supported, would

need 30 minutes to fully charge the battery. Ni-MH batteries are sensitive to damage on overcharge when the charge rate is over C/10.

Previous statements assumes that the battery has a 100 % coulometric charging efficiency⁸. The coulometric charging efficiency of Ni-MH batteries is typically 60 %. However, the faster you charge the worse this parameter gets.

Hence, for our 2.9 Ah batteries, considering a charge rate C/10. A charge current of 290 mA could be accepted, taking from 10 to 15 hours to fully charge the batteries. However, let set the charge current around 200 mA as precautionary measure. This way, quality of the batteries is preserved and the charger is suitable for less capacity batteries.

We also want our charger to automatically switch-off once the batteries are full. We will keep sensed the voltage at the batteries poles to detect the end-of-charge moment. As initial approximation we could establish the end-of-charge when voltage at the batteries is around 10 % higher than their nominal value, 3.6 V. The sensed voltage will be compared with a reference voltage at the input of an LM311[102] comparator. Reference voltage is obtained by a simple voltage divider.

The LM311 is a single high-speed voltage comparator. These devices are designed to operate from a wide range of power-supply voltages, including ± 15 V supplies for operational amplifiers and 5 V supplies for logic systems. The output levels are compatible with most TTL (*Transistor-Transistor Logic*) and MOS (*Metal-Oxide-Semiconductor*) circuits. These comparators are capable of driving lamps or relays and switching voltages up to 50 V at 50 mA. Figure 4.24 gives the recommended configuration provided by the manufacturer.

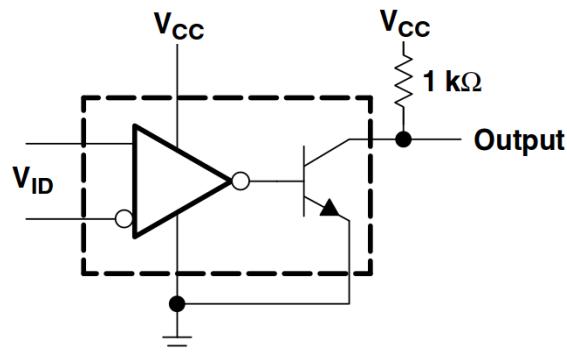
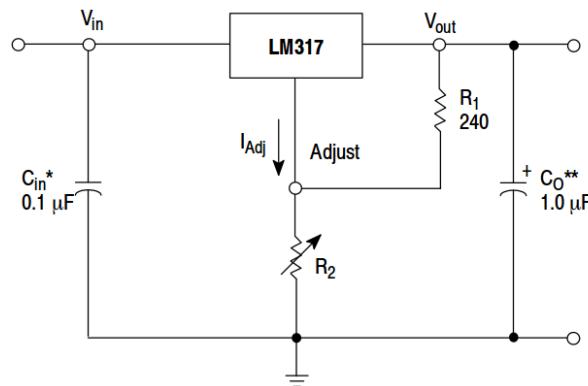


Figure 4.24: Collector output LM311 configuration.

To set a fixed current through the batteries, an LM317[103] voltage voltage regulator is used. This is an adjustable 3-terminal positive-voltage regulator designed to supply up to 1.5A of load current with an output voltage adjustable over a 1.2 V to 37 V range. It employs internal current limiting and thermal shutdown. Figure 4.25 illustrates the configuration required for the LM317. Formula below describes the relation among components and V_{out} .

$$V_{out} = 1,25V \left(1 + \frac{R_2}{R_1}\right) + I_{Adj}R_2$$

⁸Part of the electrical charge stored in a battery by charging that is recoverable during discharging. Usually expressed as a percentage.



* C_{in} is required if regulator is located an appreciable distance from power supply filter.
** C_o is not needed for stability, however, it does improve transient response.

Figure 4.25: Configuration scheme for the LM317 provided by the manufacturer.

This regulator is configured to an output of 1.25V by choosing an R_2 of 0Ω . Then a fix resistor between its output and ground will limit the output current and therefore, the input current. This fix current will flow through the batteries during the charge. In addition, the drain of an IRF5802[104] nMOSFET power transistor is placed at the regulator output. This means that the operation mode of the transistor affects the flowing current. If the transistor is in ohmic mode or saturation, current flows. The current does not flow if the transistor is cutoff.

IRF5802 presents a maximum drain-source on-resistance $R_{DS(on)}$ of 2.2Ω . Gate to source voltage V_{GS} can vary from -30 V to $+30$ V, and gate threshold voltage $V_{GS(th)}$ is around 4Ω . Maximum drain current I_D goes up to 600 mA.

The previously mentioned LM311 output controls the gate of the transistor. It switches the transistor operation modes, from ohmic to cutoff, when LM311 output is set to high or low, respectively. And the output of this operational amplifier, in turn, relies on the voltage sensed at the batteries.

A green LED driven by the comparator output, turns illuminated when this output drops enough to make current flows through the LED. This way, the LED indicates when the batteries voltage is over the reference voltage. A precautionary diode preserves the batteries from being reversely charged.

The whole system was simulated on Altium Designer and simulation outputs can be observed further on at this same section.

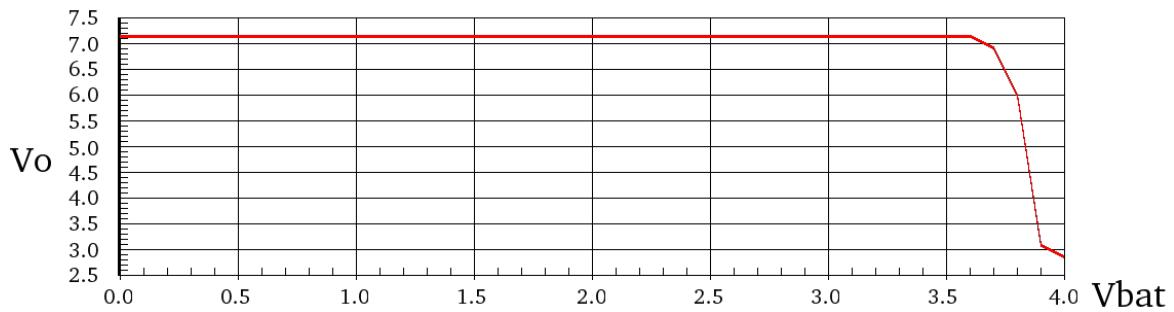
4.7.2. Schematics

A schematic of charger232SHIELD can be seen at Figure A.12 on Appendix A.

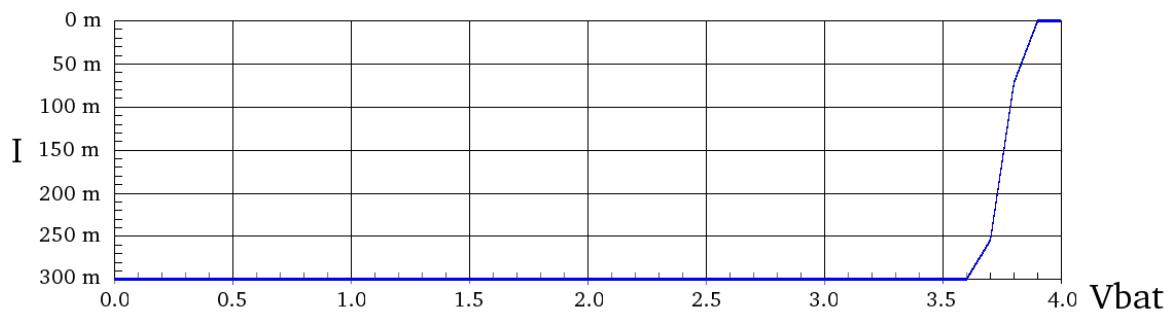
4.7.3. Simulation

Simulation at Altium Designer showed different behaviors facing different voltage at the batteries. It was needed to simulate models for LM311 operational amplifier, IRF5801 nMOSFET transistor. LM317AMDT was emulated composing simpler components. Figure 4.26a illustrates the output voltage at the LM311 comparator depending on the voltage at the battery. Figure 4.26b directly relates current flow over the battery with the voltage its presents. Graphics show the right behavior, changing

properly the desired voltage and current values at certain battery value.



(a) Output voltage at the comparator depending on the present voltage at the battery.



(b) Current flowing through the battery depending on its voltage.

Figure 4.26: chargerSHIELD simulation results.

Chapter 5

Implementation Study

Time of your life

Thomas & Guy-Manuel, Daft Punk

In this chapter, tasks related to the physical implementation of the platform are described. These tasks cover from the layout design, 3D modeling, and GERBER files generation, to the final components mounting on the PCB. Decisions about components or layout designs, and final results are exposed.

5.1. Introduction

Once the system is designed over a logic circuitry level, it is moment to build a PCB design responsive to the developed schematics. This process includes the PCB dimensions, components placement, tracks routing, silkscreens, etc. These tasks are achieved over the Altium Designer, same software described in Section 4.2.1.

All the implementations share some common features regarding their layout designs. On the one hand, the used components will keep, whenever possible¹, SMD packaging. This technology gives a more reduced size and supports lower power ratings. Precisely, the SMD package used for most of two-terminal components is the 0603 size².

All the created layouts are built over two-layer boards. Two layers are enough for our purposes since they allow to place the components in a sufficiently reduced area and power planes are not required. More layers, in addition, would increase significantly the board costs.

For the tracks routing process, some considerations are taken. Ground planes are placed to face noise reduction, this way a low impedance way is provided for the power signals. Moreover, parasitic EM emissions are reduced. Power tracks are width enough to support the supplied current flow.

Together with the layout deployment, Altium Designer offers the chance to embed 3D models of each component, building up eventually a whole 3D model of the complete board. This tool was employed, and 3D models were rendered, obtaining a preview of the modules on advance to their materialization.

¹Some components do not offer SMD (*Surface-Mount Devices*) packaging due to their electric properties.

²1.6 mm x 0.8 mm (0.063 in x 0.031 in). Typical power rating for resistors are 0.1 watt.

PCB manufacturer, PCBCART³ will produce the PCBs from the outputted fabrication files. Final PCB posses a 1.6 mm thickness, and copper layers are 35 μm thick.

Sequential mounting, and the existing firmware allow to carry through simple tests to check proper functionality and detect errors. Actually, several corrections and adjustments were made during the mounting task.

5.1.1. Tools

Small package components, like the employed here, requires for precision tools to be mounted. Proper tweezers are helpful dealing with tiny components. The soldering station used is a JBC AM 6000[105], offering several soldering tips on the 1-4 mm diameter and useful tools as hot tweezers or desoldering accessories. Microscope is very valuable for better precision, the one used is a Bausch & Lomb StereoZoom 4 Microscope.

Components use to include at their datasheets guidelines about their surrounding layout, soldering tips, or mounting processes. They are reviewed and taken into account for the process.

5.2. Transceivers - μ Trans 434/868



Figure 5.1: μ Trans 434/868 3D model.

³www.pcbcart.com

5.2.1. Description

As it was viewed, this module was developed because of the non-existence of reduced size RIs including the MRF49XA transceiver. Therefore, by definition, the module must posses a reduced size. Taking advantage of the included MRF24j40MA RI, it was decided to design a PCB module fitting the footprint of the named RI. This way a single footprint is common to all the included RIs. This footprint is defined in Figure 5.2.

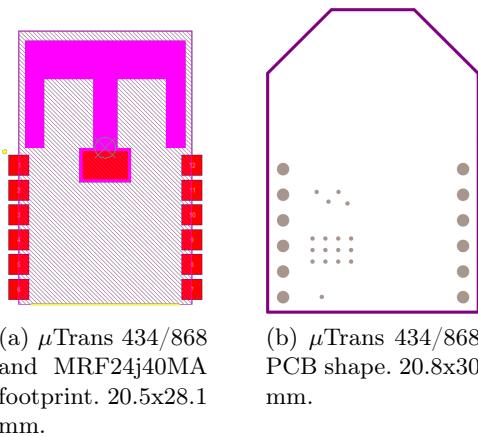


Figure 5.2: Real size footprint and shape for μ Trans module.

The PCB size must be suitable for this footprint, not exceeding its width. μ Trans PCB shape is defined as in Figure 5.2.

Module pads are based vias⁴, with a 1.2 mm diameter through hole. These holes are width enough to let the solder tip into them and solder the pads over the footprint. These pads constitute the electrical interface between the μ Trans and cNGD and suppose a low-cost and size solution.

MRF49XA datasheet provides hints for the inclusion of the transceiver into a PCB. It recommends to fill with ground vias the area underneath the transceiver. In addition, to have an easily accessible measure of the clock signal, the top layer posses a small pad that provides this signal. Minimal clearance among tracks is set to 0.2 mm and the standard track width is 0.2 mm. Power tracks width is set to 0.4 mm and ground planes are placed on top and bottom layers. Vias through holes minimal diameter are 0.4 mm width. Table B.1 at the Appendix B contains the needed components for the μ Trans 434/868 on both versions.

The μ Trans RI accepts external monopole antennas through a coaxial SMA (*Sub-Miniature version A*) female connector. Hence, antennas must posses an SMA male connector kind to be attached to the modules. Figre 5.3 illustrates the connectors kind.

5.2.2. Layout

Different layers of the layout can be observed at Appendix B. Figures B.1 and B.2 show the top and bottom copper and silkscreen layers respectively.

⁴A via is a electrical connection between layers in a physical electronic circuit that goes through the plane of one or more adjacent layers.



(a) μ Trans antenna receptacle, female SMA.
 (b) Suitable antenna connector, male SMA.

Figure 5.3: μ Trans antenna connectors.

5.2.3. Prototype mounting and testing

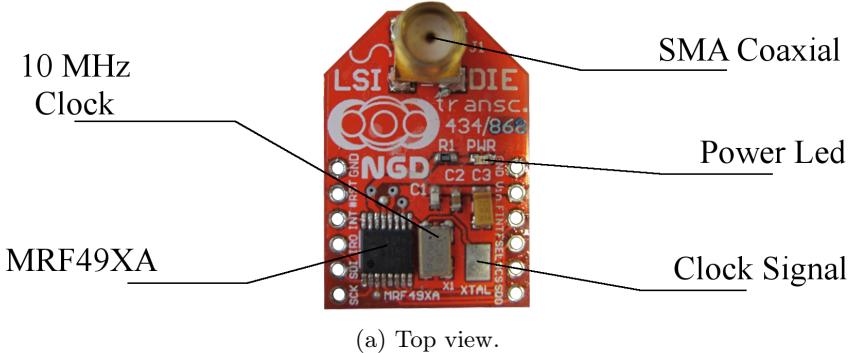
Basic tests to check proper functionality were made while prototyping the module.

Firstly, the modules were tested over the FCD platform. For this, the μ Trans pads were wired and connected to header prepared for the MRF49XA PICtail Plus Daughter Board. Over this platform and making use of a multi-probe oscilloscope, SPI and other interface signals were checked, and possible errors debugged.

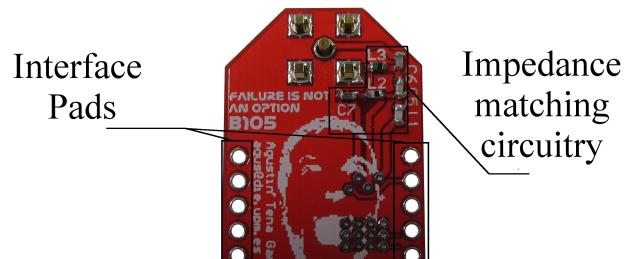
After this, and in collaboration with other MRF49XA modules, it was checked that the module properly carried out handshaking, network detection and joining, and data transmission and reception processes.

5.2.4. Final Result

Figure 5.4 shows a commented top and bottom photography of the μ Trans prototype.



(a) Top view.



(b) Bottom view.

Figure 5.4: Detailed view of the μ Trans module.

5.3. Main Board - cognitiveNextGenerationDevice

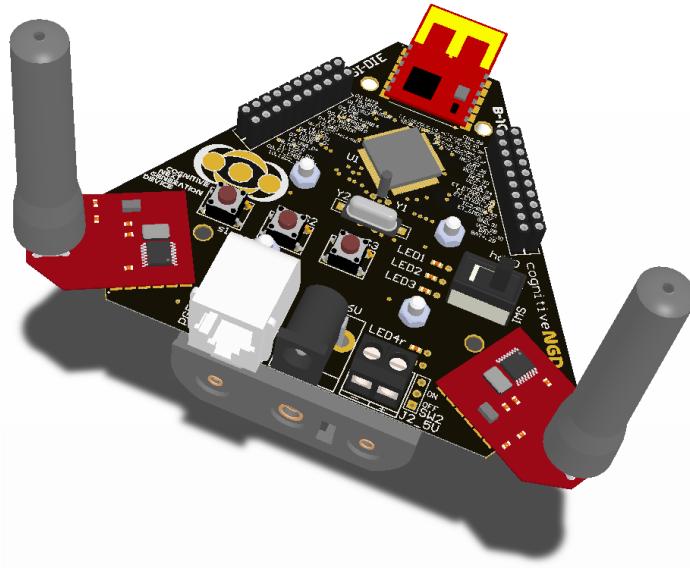


Figure 5.5: cNGD 3D model.

5.3.1. Description

The cNGD PCB has to implement the whole design described in Section 4.5 attending to expand functionality and usability, and reducing size and cost. Since this is key board of the whole system, most part of the weight of the operation requirements relies on it. It must mainly contain the three RIIs, give support to expansion modules, include the power supply and battery room, programming system, and microcontroller. In addition, it must provide an easy access to the control modules (LEDs and push buttons) and easy deployment. Regarding its test-bed character, the module must include some kind of anchors to fix it anywhere or encrust some feet. All these features must be achieved keeping size and cost limited, so the module become affordable and valuable. Figure 5.6 shows the PCB shape.

Power supply system, μ USB and PGE connector are gathered together as entire interfacing slot at the large edge 2 of the PCB. RIIs are thought to be placed at the three short edges. 434 MHz RI at edge 1, 868 MHZ RI at edge 2 and 2.4 GHz RI at edge 3. Figure 5.9a can give a better view of these ideas.

The board includes four holes for the battery to be attached. Figure 5.8 illustrates the way it works, four legs fix together the board and the batteries-holder.

Headers module has been placed relying on the large edges 1 and 3, Figure 5.8a shows it. Future expansion modules will be attached to this module through exactly equal headers on them, this way it will be possible to stack several expansion modules ones above others. The headers footprint to include in future expansion modules to fit the design is illustrated in Figure 5.8.

The minimum spacing among tracks is set to 0.1 mm around the MCU and 0.2 mm in the rest of the layout. Tracks widths vary from 0.2 mm up to 1.2 mm in power tracks. Vias through holes have a 0.4 mm width diameter. Table B.2 at Appendix B

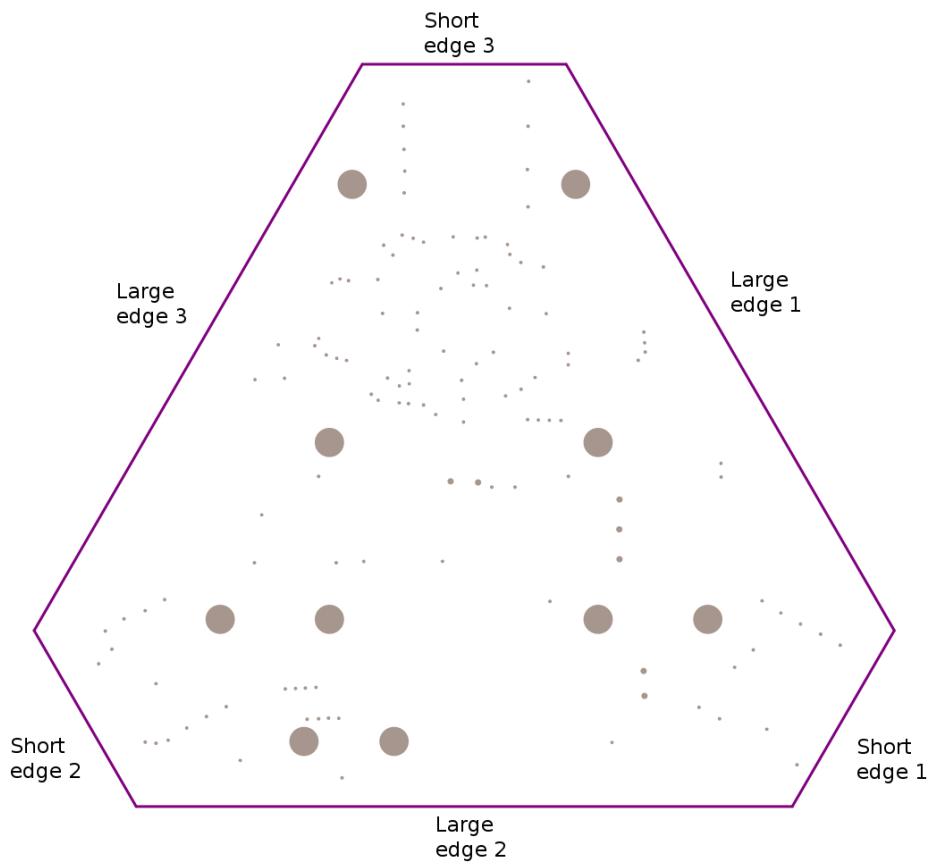


Figure 5.6: cNGD PCB real size shape. 97x84 mm

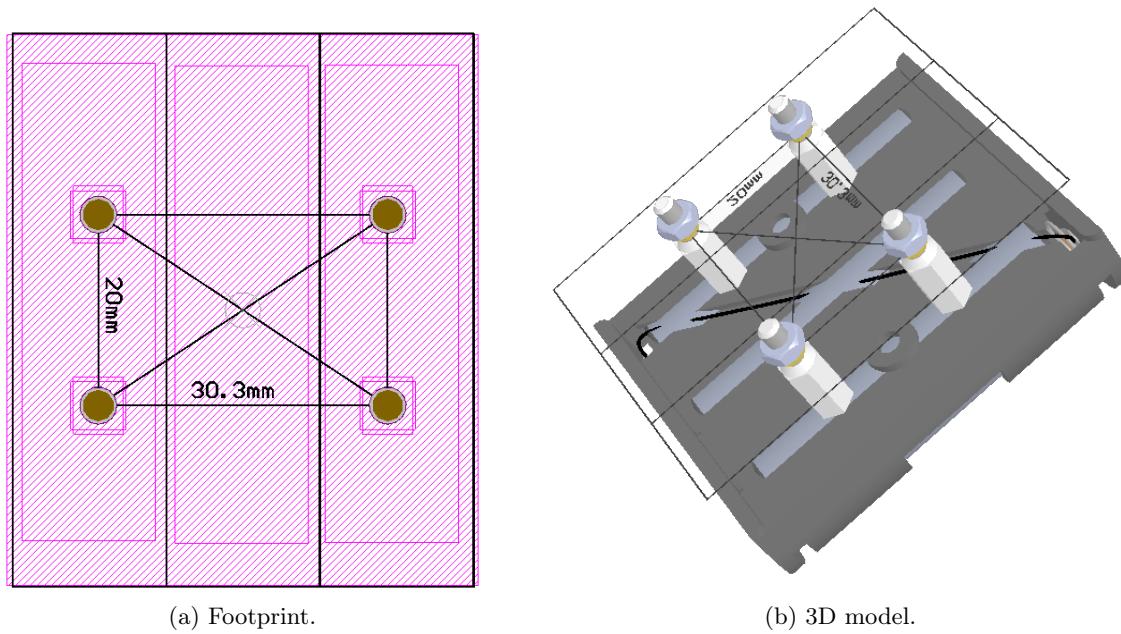


Figure 5.7: Battery anchorage system.

details the needed components for cNGD module.

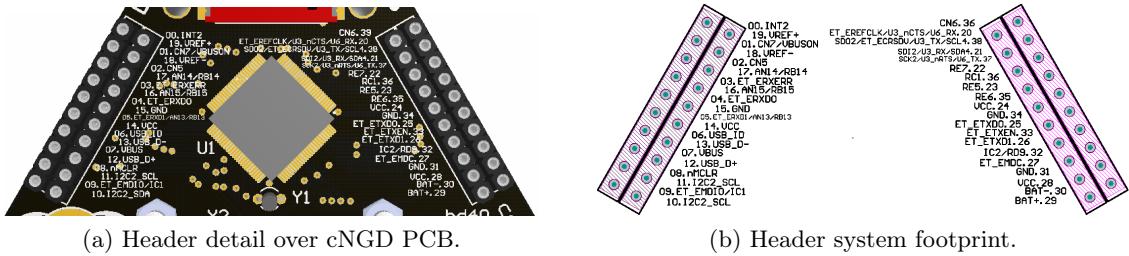


Figure 5.8: 40-pin Header system.

5.3.2. Layout

Different layers of the layout can be observed at Appendix B. Figures B.3 and B.4 show the top and bottom copper and silkscreens layers respectively.

5.3.3. Prototype mounting and testing

When prototyping, first part to test is the right operation of the MCU. After soldering R4, C2, the reset circuit, PGE module, and the MCU itself, the microcontroller is brought under its first programming. For a right operation, is needed a right software configuration of the MCU parameters, choosing the internal clock.

Once the right programming of the MCU is tested, the external clocks are mounted and tested by changing the software parameters. After this check, it is time for push buttons and LEDs, which are tested using test software described in Section ???. Rest of decoupling capacitors around the MCU can be soldered.

For these already described checkings, power supply from the ICD (*In-Circuit Debugger*) 3 is used. Next step is to mount the power supply system and, using external sources, try the right operation of the mounted modules. Since μ USB takes part of the power supply system, it was checked at this stage. A basic test for the USB was included at the firmware.

At this point, the need of debugging traces arises to keep on mounting the device. A serial communication interface is required, and since the used firmware already supports for RS-232 communication, the rs232SHIELD is mounted and used. To read more details about this mounting, go to Section 5.4. Headers must be soldered on the board to use this shield.

Last modules to try are the RIIs. One by one, they are added to the cNGD and checked. The chosen firmware offers several radio tests for general TX/RX, handshaking, channel switching. The RIIs can be tested in conjunction with the equivalent RIIs at the FCD. At this point, it is possible to check the proper operation for the software controlled switches that enable power to the RIIs. Basic tests have been specifically developed to check this function.

5.3.4. Final result

Figure 5.9 shows a commented top and bottom photography of the cNGD prototype.

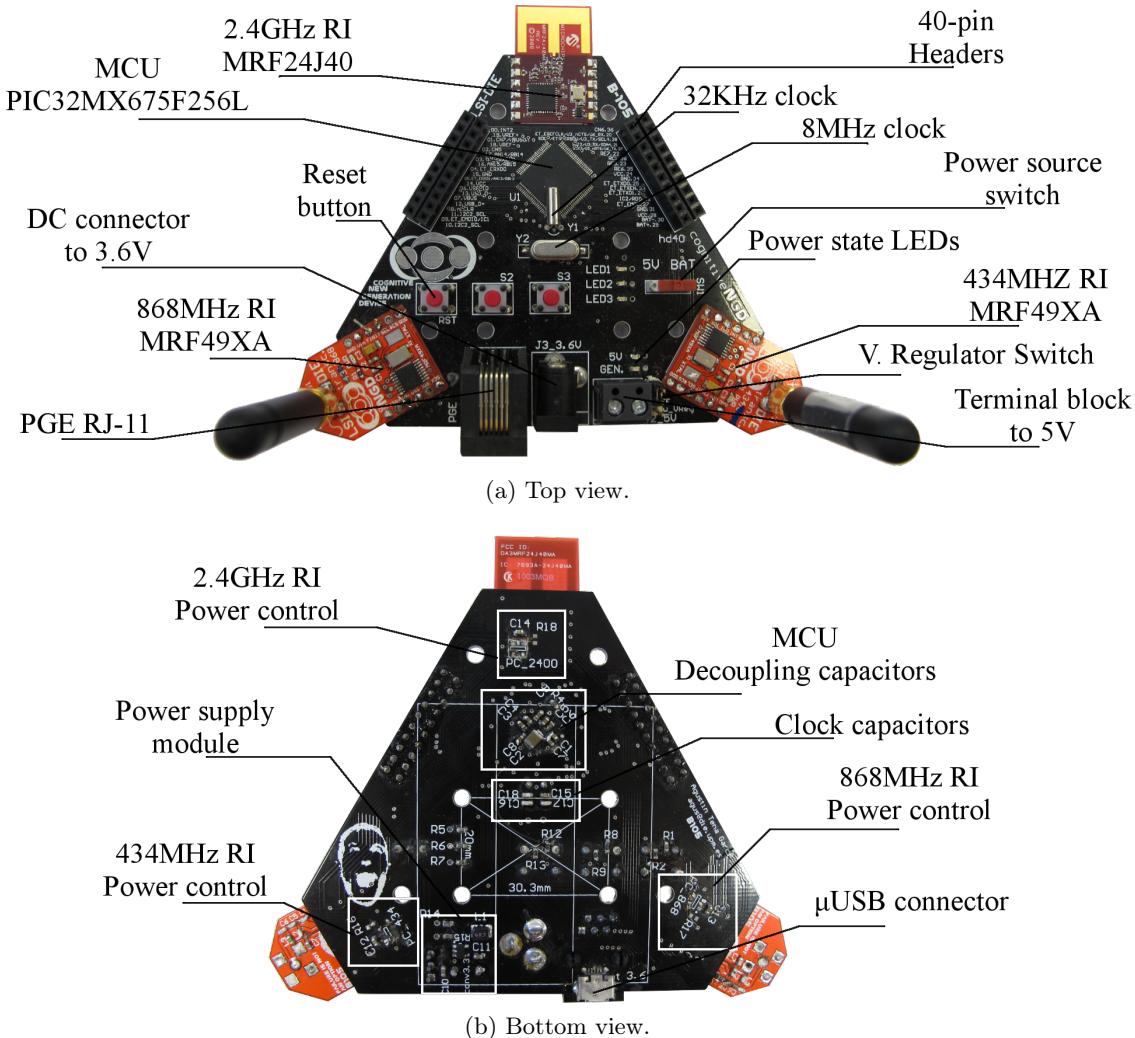


Figure 5.9: Detailed view of the cNGD module.

5.4. Serial Communication Board - rs232SHIELD

5.4.1. Description

This module is an expansion board or shield for the cNGD module. The cNGD sets the interface with the shield over the already described headers. The board shape posses a total size of 67.9x33 mm.

The RX signal is shorted to pin 21 and TX signal to pin 38.

In addition, the module contains a RS-232 female connector to be connected to the computer or other device. Minimal track spacing is set to 0.2 mm while the tracks width is from 0.2 mm to 1.2 mm on power tracks. Vias keep set to a 0.4 mm diameter. Table B.3 at Appendix B details the components needed for this module.

5.4.2. Layout

Different layers of the layout can be observed at Appendix B. Figures B.5 and B.6 show the top and bottom copper and silkscreen layers respectively.

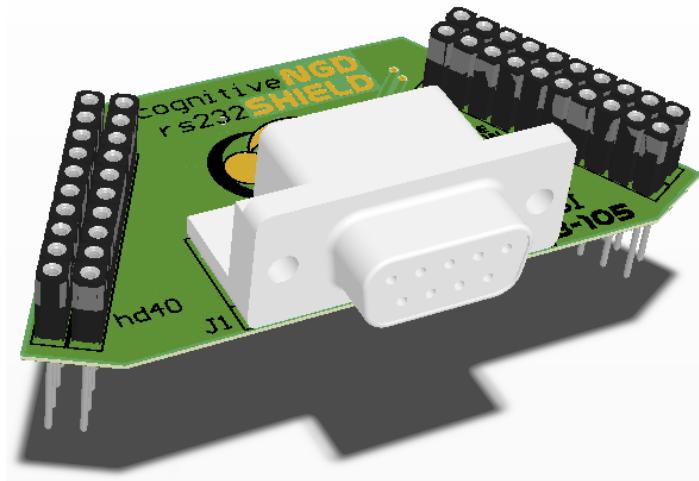


Figure 5.10: rs232SHIELD 3D model.

5.4.3. Prototype mounting and testing

This module offers simplicity and does not require much testing. Basically, the whole module is properly soldered and plugged into the cNGD headers. In addition, the connection headers enable stacking new shields over. Firmware must be configured to debug over the UART number 3.

Connecting the RS-232 connector to a computer and with a suitable software, such as Minicom⁵, and configuring properly the communication parameters, is enough to receive data from the running application.

Serial communication parameters can be changed at the firmware configuration. However, a standard configuration is as follows:

- Bit rate: 115200 bits/s
- Bits per framing: 8 bits
- Parity: No
- Stop bits: 1 bit

5.4.4. Final results

Figure 5.11 shows a commented top and bottom photography of the cNGD prototype.

5.5. Battery charger - chargerSHIELD

5.5.1. Description

This module is also a shield for the cNGD module. The cNGD sets the interface with the shield over the known expansion headers. This shield shares the same shape than rs232SHIELD.

⁵Minicom is a text-based modem control and terminal emulation program for Unix-like operating systems. A common use for Minicom is when setting up a remote serial console.

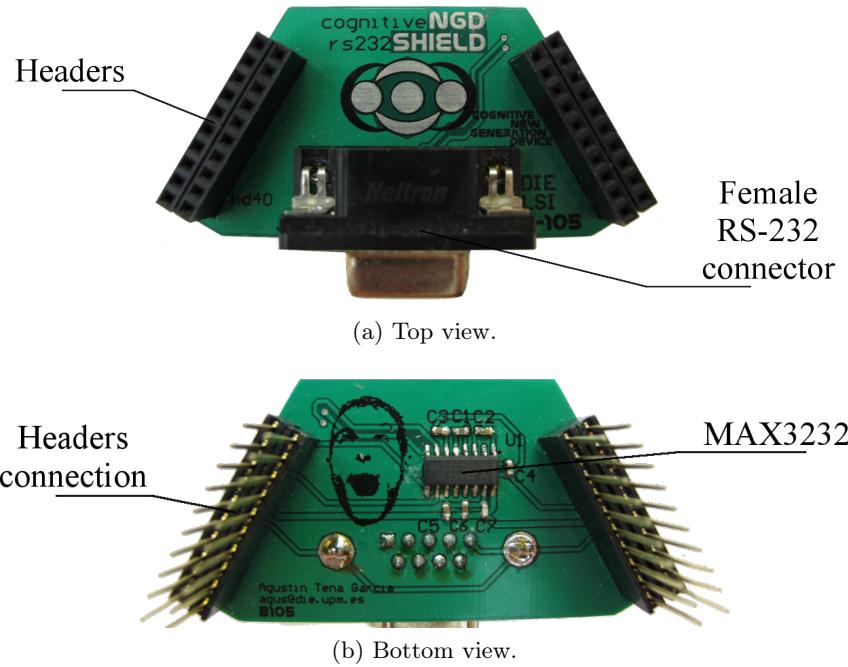


Figure 5.11: Detailed view of the rs232SHIELD module.

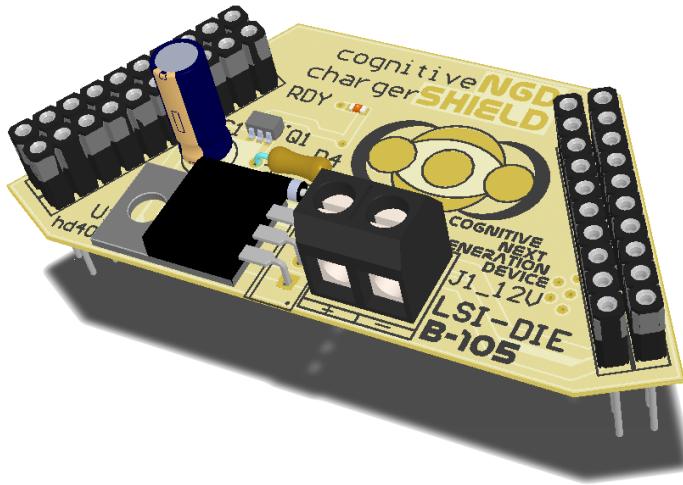


Figure 5.12: chargerSHIELD 3D model.

Battery positive pole is accessible at the header on pin 29, negative pole is at pin 30. These two pins are linked to the circuit. Apart from this interface, the charger encrusts a terminal block to connect a external 12V source. This source must be capable to provide up to 500 mA.

Current version of the charger does not include any heat dissipater. Tests showed that not heat dissipater was required for the current configuration. Any change over the resistors configuration must consider the overheating possibility on LM317AMDT component. Thus a heat dissipater could be needed.

Minimal track spacing is set to 0.2 mm while the tracks width is from 0.2 mm to 1.2 mm on power tracks. Vias posses a minimum through hole of 0.4 mm diameter. Table B.4 at Appendix B details the components needed for this module.

5.5.2. Layout

Different layers of the layout can be observed at Appendix B. Figures B.7 and B.8 show the copper and silkscreen layers top and bottom respectively.

5.5.3. Prototype mounting and testing

This module required for a lot of initial debugging, it was needed the use of a breadboard to mount parts of the module in parallel to compare behaviors. It is important to notice that some currents on the circuit are up to 250 mA, existing possibilities of malfunctions that make this even higher. It was needed a external power supply with output current indicator and limitation for it. Of course, this module was not initially tested over the cNGD, but emulating it just with batteries or a second voltage source.

First part of the chargerSHIELD to try is the LM311 amplifier and surrounding components, LED, R1, R2, and R3. These parts constitute the comparator and with this, the comparator operation intends to be tested. Once this is checked, rest of components are soldered.

At this point, a lot of tests were carried out using different voltages and number of batteries to calibrate the system. The goal was to adjust the current over the batteries to a proper level, measure charging times and the proper operation of the notification LED. R1, R2, R3, and R5 were finally set to their current value. Having this task achieved, external controlled source was swapped by a commercial 12V adapter and its proper operation was proved.

Once the chargerSHIELD showed a proper operation it was tested over the cNGD.

5.5.4. Final results

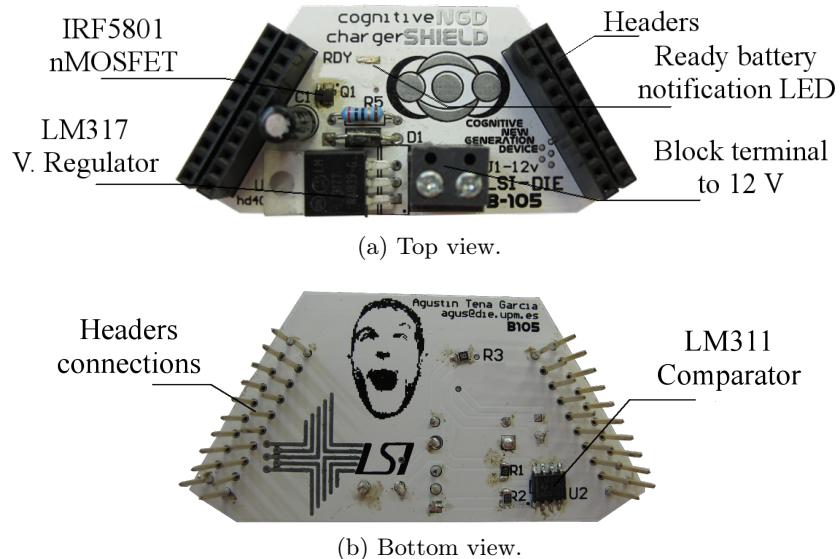


Figure 5.13: Detailed view of the chargerSHIELD module.

5.6. Conclusions

The whole implementation supposes a modular, versatile, functional, and flexible platform for CWSN development. cNGD module is the main piece, containing the essential functionalities for a WSN node. This main part is able to host up to three RIs, becoming the only WSN platform together with the FCD meeting this feature, and the only platform for WSNs capable to access three different frequency bands. In addition, the cNGD accepts attachable modules that brings up new functionalities without requiring a new complete design. An existing, ad-hoc developed, firmware has been used to drive the platform as a design requirement.

It was needed the development of ad-hoc RIs based on MRF49XA transceiver. These developed RIs, called μ Trans, operate over the 434 MHz and 868 MHz depending on the impedance matching circuitry and antenna they contain. They employ MiWi protocol and support sleep modes. The firmware employed is already prepared to fully drive these RIs. It even includes a single MiWi stack shared by the three possible RIs. μ Trans are designed to be soldered into the cNGD board and they accept external antennas over a coaxial connector.

Third RI, a MRF49J40MA, consist of a commercial option that operates over 2.4 GHz. It is also MiWi compliant, accepts sleep modes and it is supported by the named firmware. This RI is also attached to the cNGD by soldering and it contains a PCB antenna.

cNGD is controlled by a PIC32MX675F256L. A 32-bit, low-power, high performance, 100-pin microcontroller that posses a 64KB RAM memory and 256KB flash memory. A larger flash memory pin-compatible MCU exists in case of need. The MCU enables a clock speed up to 80MHz, including an external 8MHz clock and a low-power 32 KHz RTCC. The MCU supports sleep modes and it is widely featured.

Power supply encrusted into the cNGD has been designed to accept either 5V or 3.3V. 5V can be supplied over a μ USB connector, which also serves as serial interface, or a block terminal. This voltage is lowered down into the board voltage operation, 3.3V, by a buck regulator. 3.3V supply option is thought for portability chances, being possible to use an included 3.6V Ni-MH battery for the board. In addition, it is possible to control, from the MCU, the RIs power supply.

cNGD offer functionality expansions by use of a pair of 20-pin headers. These headers enable access to general use peripherals and unused pins at the MCU for future needed implementations. The board contains three LEDs and two push buttons apart from the reset push button. These components allow interacting with the application.

It has been also developed an attachable shied for the cNGD that enables serial communication over an RS-232 protocol accessing an UART available at the header. This serial interface has an easier internal operation than the USB option and it is useful for old machines or during the platform developing.

A charger shield was also developed in order to get the batteries easily charged through the headers. This charger automatically shut-down the charge current at the batteries when the full charge is detected. It also contains a LED that indicates when the battery is full. Current used to charge the battery is fixed and it is set to 200 mA. The charger is prepared to charge batteries from 1V to 5V.

Chapter 6

Testings and Results

*She's up all night to the sun
I'm up all night to get some
She's up all night for good fun
I'm up all night to get lucky*

Thomas & Guy-Manuel, Daft Punk

Some tests were applied to main platform modules for the purpose of proving their proper functionality. Main capabilities that make the platform valuable such as power consumption or spectrum sensing features have been tested. Descriptions of the carried out tests and corresponding results are exposed in this chapter.

6.1. Test Model

The purposes of testing some of the platform functions are to prove valuable capabilities and proper functionality of modules. Characterization is not expressly required and modules must respond to operation parameters provided by manufacturers. The already described used firmware includes several test-benches to apply. Most of the software tools employed to test the platform are extracted, at least partially, from them. For a further description on the provided tests design, consult ??.

Tests are simple and they check bounded functions related to the platform main capabilities. Sleeping modes operation and corresponding current consumption at different modes was tested. Microcontroller computing capabilities, and radio interfaces features, such as spectrum sensing, have been also measured. On the other hand, battery charger behavior was verified.

6.2. Energy management test

This test tries to bring the cNGD under different operation modes. On the one hand, different sleeping modes and energy options for RIs and MCU are achieved consecutively. On the other hand, consumed current by the platform is measured at each situation. This test was already included at the firmware and just few changes were needed to fully carry it out. These changes suppose the functions to control the power at the RIs and they are described at Section 7.2.3.

8 Different situations are configured. Following description covers the initial state and sequential happening events:

- A situation: Initial situation. The three RIs and the MCU are in run mode.
- B situation: 434 MHz RI goes to sleep mode.
- C situation: Power supply at 434 MHz RI is switch off.
- D situation: 868 MHz RI goes to sleep mode.
- E situation: Power supply at 868 MHz RI is switch off.
- F situation: 2.4 GHz RI goes to sleep mode.
- G situation: Power supply 2.4 GHz RI is switch off.
- H situation: MCU goes to sleep mode.

Consumption measures are reflected at Figure 6.1, where a graph shows how the current consumption goes down throughout the different set situations.

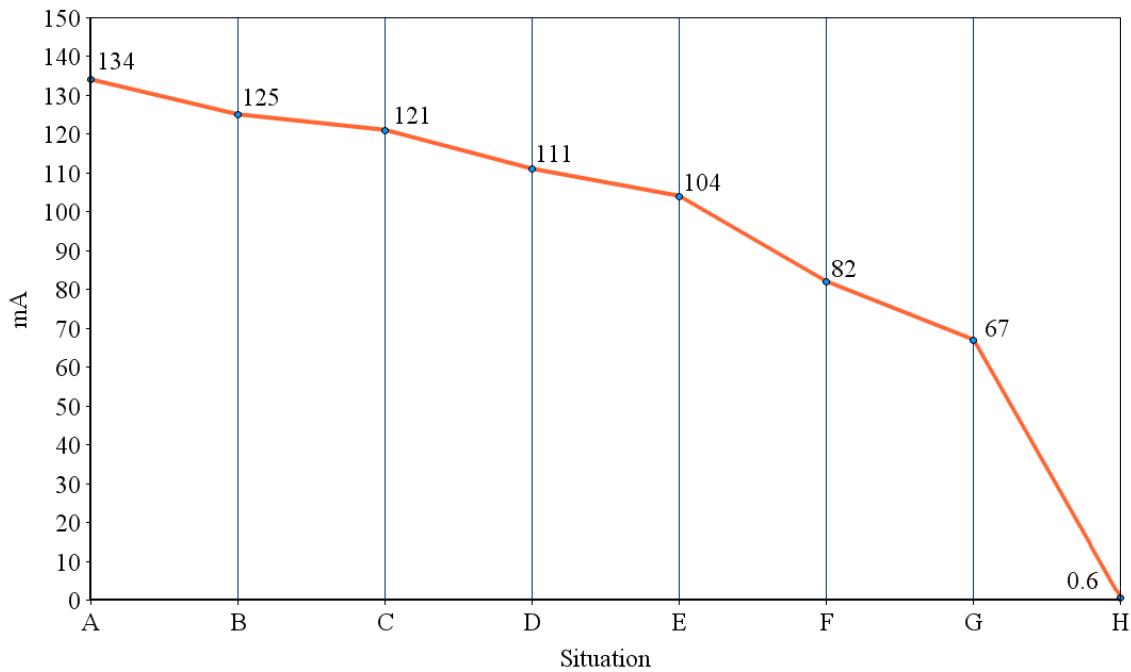


Figure 6.1: cNGD current consumption at different energy modes.

6.3. Computing test

This test intends to measure the computational cost for the management of the communication protocol stack. This could help to make a good software planning and avoid troubles interfering the application proper running. Moreover, it is important to adjust the management tasks period carefully for a good performance over communications.

For this, it will be determined how long takes the management task to be accomplished in a FFD (*Full Function Device*). Nominal clock frequency is set to 80 MHz and no network activity is considered. Management tasks are taken every 25 ms. This test is fully provided at the firmware.

Table 6.1 shows separately the measured time for RIs and different protocol versions, Mesh and P2P.

Management part	P2P Protocol	MiWi Protocol
Common management		
434 MHz RI management		
868 MHz RI management		
2.4 GHz RI management		
Total time		

Table 6.1: Time costs on the communication protocol management tasks.

6.4. Spectrum sensing test

This test tries to show the right spectrum sensing capability of the platform since it supposes a key point for its purposes. Using two cNGD prototypes and making use of the functions provided by the HAL, three simple scenarios show energy sensing features at the three different frequency bands. All the scenarios host a device continuously transmitting unicast packets at determined channels. This RF activity is detected by the platform at different spectrum energy scans.

First scenario is conducted over 434 MHz band. This band posses 2 available channels when using a bit-rate of 119,2 kbps. Transmitting device is making use of channel 1. Figure 6.2 shows the energy scan traces deployed by the firmware.

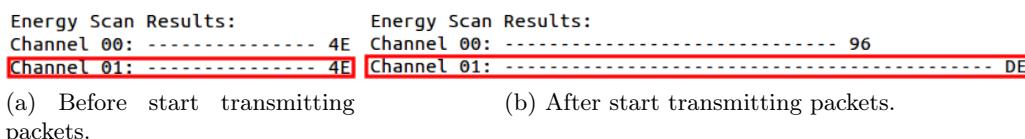


Figure 6.2: Spectrum sensing at 434 MHz.

Second scenario is conducted over 868 MHz band. This band posses 7 available channels when using a bit-rate of 119,2 kbps. Transmitting device is making use of channel 4. Figure 6.3 shows the energy scan traces deployed by the firmware.

Third scenario is conducted over 2.4 GHz band. This band posses 16 available channels. Transmitting device is making use of channel 4. Figure 6.4 shows the energy scan traces deployed by the firmware.

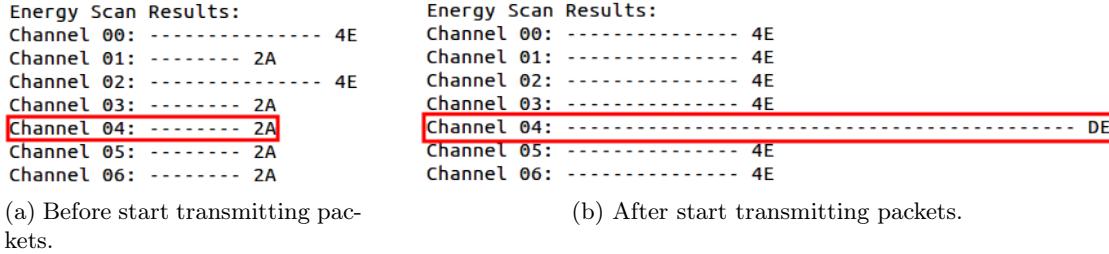


Figure 6.3: Spectrum sensing at 868 MHz.

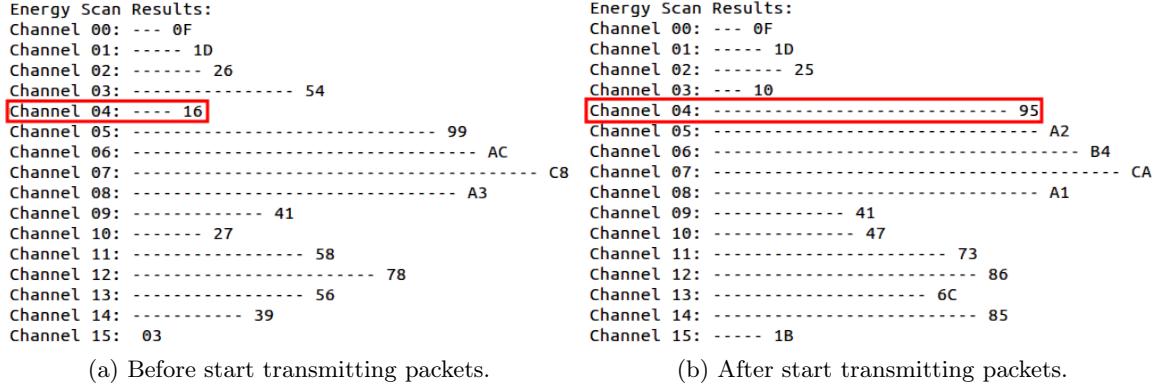


Figure 6.4: Spectrum sensing at 2.4 GHz.

6.5. Radio interfaces agility test

This test evaluates how long changes over the RIs take to be done. Parameters to change are TX power, operation channel and energy modes. This test was fully available at the firmware. Table 6.2 illustrates the results.

Operation	434 MHz RI	868 MHz RI	2.4 GHz RI
Switch transmission channel	30.45 ms	30.4 ms	0.151 ms
Change transmission power	55.8 μ s	55.8 μ s	70.1 μ s
Sleep and wake up	15.4 ms	15.4 ms	0.364 ms

Table 6.2: Time costs on the communication protocol management tasks.

6.6. Effective rate test

Effective rate provided by the MRF49XA and MRF24J40 transceivers driven by the firmware was already described in [8]. The obtained performance must match performance achieved at the cNGD. That is why this checking supposes a partial test compared to the full previous characterization. However, it tries new uncheckable aspects of the RIs management and proves functionality.

Most part of this test is obtained from the firmware, so the description is covered at its literature. Employed protocol at the test-bench, for simplicity, is MiWiTM P2P. The three RIs are checked and different maintenance task periods for the protocol stack are tried. In total, periods of 1 ms, 25 ms and 500 ms were tested, not obtaining

performance differences among them. To avoid redundancy, shown results correspond to 25 ms period.

Two TX modes are broadcast and unicast (long address). Main communication parameters set for the test follow:

- Payload size is set to 90 Bytes.
- TX and RX buffer is configured as 10 packets for each RI.
- Total number of sent packets is 250.
- VERIFY_TRANSMIT option is enabled for MRF24J40 module.
- ENABLE_SECURITY option is active.

Both versions, μ Trans module at 434 MHz and 868 MHz, posses similiar performance since the share the same transceiver. Tables 6.3 and 6.4 illustrate the results at two different bit-rates, 115200 and 38400 bps, for an μ Trans RI.

TX mode	Received Packets	Total bits	TX Time (s)	Effective rate (kpbs)
Broadcast	125 (50 %)	90000	2.88	31.250
Unicast	250 (100 %)	180000	7.34	24.511

Table 6.3: Effective rate, received packets comparison for 434/868 RI using P2P protocol at 115200 bps .

TX mode	Received Packets	Total bits	TX Time (s)	Effective rate (kpbs)
Broadcast	250 (100 %)	180000	6.22	28.938
Unicast	250 (100 %)	180000	8.7	20.669

Table 6.4: Effective rate, received packets comparison for 434/868 RI using P2P protocol at 38400 bps .

It is important to observe how there are not packet losses at unicast mode, and for 38400 bps there are not lost packets either when broadcasting. The reduced effective rate might be due to prototyping bad conditions, such as used corrections made with wires and rough connections that should be better at consecutive mountings. RF devices are very sensitive to these imperfections. On the other hand, TX power and antenna conditions must be review, since these suppose a change with respect to previous tests.

Table 6.5 illustrates the results for MRF24J40 transceiver as 2.4 GHz RI using a P2P protocol.

A worrying issue, already described in [8], is the packet loss even transmitting at unicast mode. The reason for this fact still remains unknown but there are some possible explanations. Most feasible option at the current moment is that ACK (*Acknowledgment*) packets are misinterpreted by the receiver and it considers a single ACK response for more than one packet. This is possible because ACK packets are not numerically sequenced at MiWi protocol.

In order to solve this issue, a work at the LSI presented in [9] show a flow control mechanism at MiWi MAC layer that apparently avoid packet loss at unicast mode.

TX mode	Received Packets	Total bits	TX Time (s)	Effective rate (kpbs)
Broadcast	130 (52 %)	93600	3.25	28.8
Unicast	140 (56 %)	100800	3.54	28.474

Table 6.5: Effective rate, received packets comparison for 2.4 GHZ RI using P2P protocol at 250 kbps.

6.7. chargerSHIELD performance test

As it was mentioned, small testings were made during the mounting process. Nevertheless, this test shows the chargerSHIELD performance test facing a battery full charge operation.

When testing the chargerSHIELD proper operation, few charging sequences were carried out. Taking measures every 2 hours of battery voltage conditions and the current flowing through it. Figure 6.5 shows the average results, battery voltages are view as blue bars referenced on the left side while current flow is expressed with an orange line respect to values at the right side.

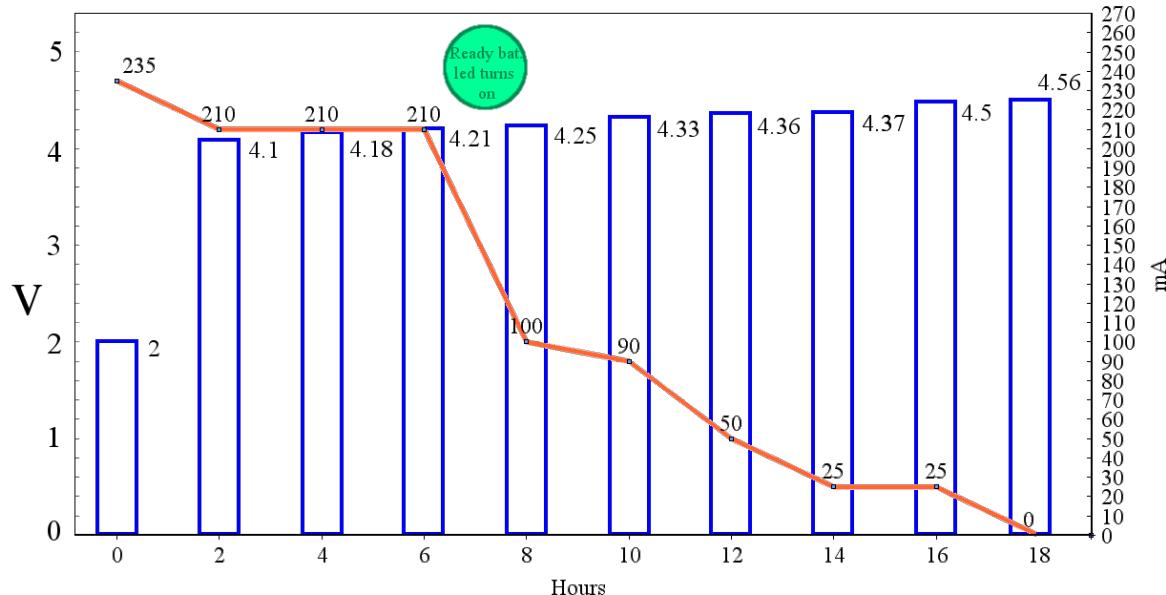


Figure 6.5: Batteries and parameters evolution during charging process at chargerSHIELD.

Charging time for the defined batteries can be estimated around 12-16 hours. The ready battery led notifies the battery is ready when approximately 60 % of the total charge is available, label at the figure shows when the led turns on. It is a normal behavior for Ni-MH batteries to supply higher voltages than their nominal value despite they are supposed to provide up to 3.6 V, specially during charging processes. The charger properly switches off the charging current when voltage at the battery reveals a complete charge.

Chapter 7

Software

*At last the long wait is over
The weight is off my shoulders
I'm taking all control, yeah*

Thomas & Guy-Manuel, Daft Punk

This chapter gives a vision about the main adaptations required for the firmware to be adapted to the platform, new software implementations and behavior of the final demo application.

7.1. Tools

7.1.1. MPLAB X

MPLAB X IDE (*Integrated development environment*) is a multi-platform software program to develop applications for Microchip microcontrollers and digital signal controllers. It is called an Integrated Development Environment because it provides a single integrated “environment” to develop code for embedded microcontrollers.

MPLAB X is based on the open source NetBeans IDE from Oracle. Some of its main features are:

- Supports Multiple Configurations within your projects.
- Support for multiple Debug Tools of the same type.
- Supports Live Parsing.
- Supports hyperlinks for fast navigation to declarations and includes.
- MPLAB X can Track Changes within your own system using local history.

7.1.2. Programmer: ICD 3

ICD 3 is a device belonging to Microchip products that allows to get the microcontroller programmed. In addition, it enables run-time debugging using the IDE. Up to 6 breakpoints can be enabled. This feature is quite valuable regarding the platform has a development character. Further information can be consulted at the manual[106].

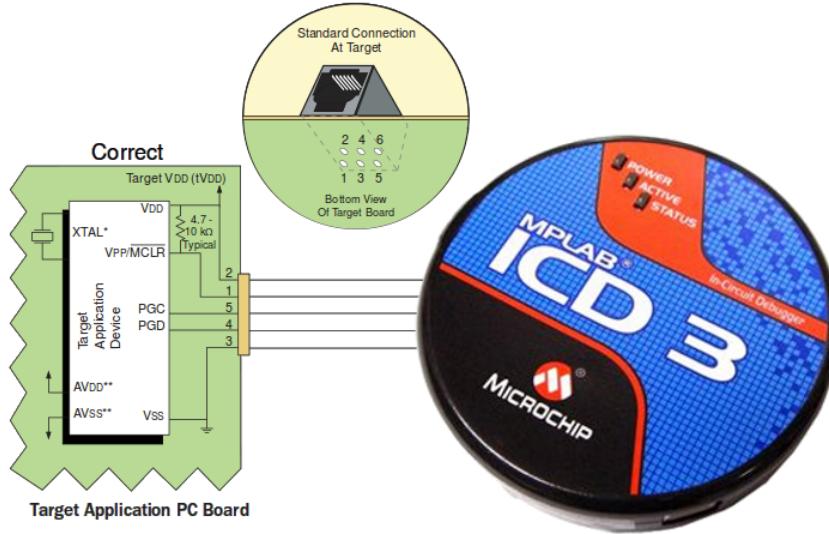


Figure 7.1: ICD3 programmer configuration.

7.2. Firmware and Hardware Abstraction Layer

When adapting the mentioned firmware?? to the developed hardware platform, it was needed to deal with minor changes resulting from its novelty and also because of required hardware adaptations.

Some minor bugs that impeded the right sleeping modes operation or the spectrum sensing were fixed. As well, due to lack of tests when using three RIs at once, some software configurations were not properly set. Current firmware version corrects these issues and it is fully operative over the hardware. For the right initialization of the MRF49XA transceivers, a 50 ms delay after setting the reset signal was included.

Other adaptations taken when developing software aim at establish better ease of use. An easier configuration scheme, USB tracing modes, RI power control functions, and new modules definition are the developed functionalities.

Current and older versions of the firmware, along with small software related tests, can be found at a public GitHub repository:

<https://github.com/agus-xyz/cognitiveNGD>. Figure 7.2 provides a QR (*Quick Response*) code to instantly access the repository.

7.2.1. Platform options

When configuring the firmware to operate over different hardware platforms such as the FCD expanded version, the cNGD, or a manually configurable platform, several changes at the configuration firmware files were needed.

With the changes that the current firmware implements, adapting the firmware to any of these platforms is done by changing a single option at *Include/HardwareConfig.h* file.

- *cNGD_PLATFORM*. Automatically sets a suitable configuration for the cNGD. *MIWI_0434_RI* is established over *MRF49XA_1* transceiver, *MIWI_0868_RI* is built over *MRF49XA_2*, and *MIWI_2400_RI* does it on *MRF24J40* transcie-



Figure 7.2: cNGD software repository.

ver. Remaining configurations to adequately take this options are automatically changed.

- *FCD_Exp_PLATFORM*. This possibility enables the configuration to employ the FCD adapted to the lately developed expansion board. It adapts the *MIWI_0434 RI* over *MRF49XA_1* transceiver, and *MIWI_2400 RI* at *MRF24J40*.
- *MANUAL_PLATFORM*. This option gives freedom to the user to full configure the employed hardware. Any of the possible MIWI and WIFI RIs can be chosen over different transceivers such as *MRF49XA_X*, *MRF89XA*, *MRF24J40*, or *MRF24WB0M*. Moreover, the used SPI for interfacing the RI is also configurable together with the external interruption line.

Basically, the old macros that used to be changed manually for each platform are now automatically configured when set on of the previous options. Main changes take place over RI configurations and pin-outs.

Other configurable option at this very same file is the possibility for debugging traces. This option is controlled enabling or disabling the *ENABLE_CONSOLE* option. Then, within each platform independent configuration, it is possible to select the module to output traces.

7.2.2. USB tracing

Regarding the wide acceptation that USB protocol has achieved and how common has this protocol become, it was interesting to offer possibility to an USB console. This way, it took advantage of the included μ USB connector. In this case, the implementation consisted on an adaptation of a Microchip USB-CDC (*Communications Device Class*) example. Stack used version is 2.9.

Adapted example offered possibilities for either polling or interruption methods, interruption management was chosen. A circular buffer would have been useful facing large volume of outputting data. So at the current state, the USB tracing supposes an ideal choice for controlled environments where tracing tasks are taken carefully and

localized. Otherwise data losses or even interferences with the communication protocol management might arise.

To configure the firmware for USB traces, firstly it is needed to set the *ENABLE_CONSOLE* option. Then, the console must be established as *DEBUG_USB*. Configuration file for USB stack is *include/USB/usb_config.h*

In case of need for further information about USB stack working, the framework [107] or stack [108] manual can be consulted. For information related to USB-CDC, manual [109] might be useful.

7.2.3. Radio Interfaces Power Control Functions

After including power control modules for each RI at the cNGD, it was needed some software adaptations to fully make them operate. For this, a pair of functions have been included at the HAL, *SwitchRION(radioInterface ri)* and *SwitchRIOff(radioInterface ri)*.

These functions set to high or low the signal that drives the power switch at each RI. Received parameter is the chosen RI. Return value might be *NO_ERROR* or an error code.

7.2.4. Headers and other new definitions

For an easy use and programming of pins at the headers and useful components as LEDs or buttons, some global definitions at the firmware provide access to them.

To access pins at the headers, applications can make use of the *HEADERS_XX* label definition and respective *HEADERS_XX_TRIS* to configure the sort of pin. *XX* parameter must be replaced with the desired pin number. For instance, *HEADERS_06_TRIS* must be set as *OUTPUT_PIN* in order to make it a digital output. To access signal at header pin number six, the pin definition would be *HEADERS_06*.

LEDs label definitions are *LED1*, *LED2*, and *LED3* respectively. These pins are configured as output when doing node initialization.

Push buttons 1 and 2 are defined as *BUTTON_1* and *BUTTON_2* respectively. The firmware includes as well a pair of masks *BUTTON_X_PORT_MASK* in case of these masks were required when reading whole port.

7.3. Demo Application Layer

Demo application layer implements a WSN application in which a transmitter and receiver module set communication and exhibit some platform communication capabilities.

Main goals for the software are to show some of the most valuable features such as spectrum sensing and RI agility on communications. Two devices take a role on the demo. One device will be the transmitter, configured at *include/MiApp.h* as *NODE_1*, and the other device will be the receiver, *NODE_2*. The application is prepared to output traces to a computer using the rs232SHIELD while running. Significant established parameters for the application are:

- P2P protocol is used for communications. The showed operation does not require routing modes offered by the MiWi protocol.

- Bit-rate for 434 MHz RI is 19.2 kbps, and 119.2 kbps for 868 MHz RI.
- Unicast communication mode.
- The two devices posses different addresses and a common PAN (*Personal Area Network*) id.
- Sleeping modes are disabled.

The normal operation of the application responds to a sequential set of steps. Below, are shown the different behaviors and tasks throughout the execution.

- RX device is switched on. It initializes the stacks, senses the spectrum and create PANs at the most suitable channel for each RI.
- TX device is switched on. It senses the spectrum, detects active PANs and joins them.
- TX starts to send packets at 434 MHz frequency band every 2 seconds.
- At this point, antennas at 434 MHz RIs are manually removed.
- After 5 failed sending attempts, the application makes a spectrum scan over 434 MHz and 868 MHz. It detects the most suitable channel over 868 MHz and request the RX to set communication over that channel.
- When RX device asserts, TX starts sending packets at 868 MHz band.
- At this point, antennas at 868 MHz RIs are manually removed.
- After 5 failed sending attempts, the application makes a spectrum scan over 434 MHz, 868 MHz and 2.4 GHz. It detects the most suitable channel over 2.4 GHz and request the RX to set communication over that channel.
- When RX device asserts, TX starts sending packets at 2.4 GHz band.
- After 5 successful packets, the communication ends.

The demo lets show how antennas affect the sensing capabilities of the platform and the different spectrum scan results all over the time. It shows how the platform is able to swap communications over 3 different frequency bands using 3 different RI. All the sent packets are proved to be received when showed at the traces. The application also proves the right operation of the rs232SHIELD.

A video showing the demo application working can be accessed at <http://tiny.cc/r5p14w>. For better comfort, Figure 7.3 provides access to the video.

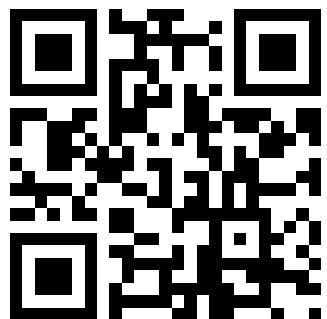


Figure 7.3: Demo application video.

Chapter 8

Conclusions

*Like the legend of the phoenix
All ends with beginnings
What keeps the planet spinning
The force from the beginning*

Thomas & Guy-Manuel, Daft Punk

This chapter covers a full review about the project. It is shown a general view of the implemented system together with the main taken decisions and carried out tasks. Most important conclusions are summarized and future lines are set.

The main goal of the project is to design and implement a node for the study of CWSNs. A demo application layer, also to be developed, must work integrated with an already implemented firmware. This employed firmware is able to give support to up to three transceivers sharing a single MiWiTM stack. It focus on abstracting the developer and the application from the cognitive model and the direct hardware management.

The platform was developed responding to the requirements and a final demo let prove the right operation of the whole system. The whole system respond to a modular design widely adaptable over the range of WSNs applications. Main module is called cNGD. This main module posses different functionalities that implements a node platform, such as power supply, power control over modules, battery, expansion headers, RIs, serial interface, and a control MCU unit. RIs on board of cNGD offer communication over 434, 868, and 2400 MHz. RIs used for 434 and 868 MHz are ad-hoc designed RIs due to the lack of suitable size modules. This ad-hoc designed RIs are called μ Trans 434/868 for 434/868 MHz respectively.

cNGD main module accepts expansion options over its headers, so-called shields, and for this project two shields were implemented. First shield gives chance to serial communication through an RS232 port. Second one is a suitable charger for the batteries. Shields allow developers to create new attachable functionalities and possibilities or give different capabilities to distinct reduced node groups of the total network. This helps to keep the downturn on complexity, consumption, size and cost but not on performance.

The sort of device shown at this project features some novel properties with respect to other CWSN devices, such as its capability to communicate over three different ISM RF bands. The hardware fits the conditions and requirements of WSN environments. Low power consumption, size and cost limitations are taken into account in order to

achieve real test-benching purposes, application development or even possible complete implementations.

In Section 1.2 takes place a fragmented set of subgoals is described, here they follow some conclusions obtained throughout the achievement of these subgoals:

- A review all over the recent implementations related to the CWSN was made at Chapter 3. The FCD architecture is quite close to the one pursued so is taken as guide for the designing process. Used technology is evaluated and weaknesses detected, also minimal requirements are stated. Many decisions, such as a reconfiguration of the RIs, the design of suitable size RIs or the need for expansion options, were taken in order to face the design.
- The whole design is covered in Chapter 4. A modular design builds up the whole system. A total of three modules were designed, a main module, an ad-hoc RI and two expansion shields that provide extra functionalities. Battery options were studied resulting in a final option of 3-cells Ni-MH battery. Components were chosen and required functionalities assigned to different modules.
- At Chapter 5 is covered the hardware implementation. The PCB layouts were deployed and prototypes mounted. Size achieved is fair for research purposes. Node functionalities were checked and design validation with consequent corrections, these task were carried out during mounting making use of small pieces of software.
- Developed hardware have been submitted to several tests to check right performance of main capabilities. Hence, MCU capabilities, RI features, sleeping modes and current consumption have been proven. For this, tests included at the software have been generally used. In some cases the suffered slight adaptations.
- Software developed for hardware checkings, firmware adaptations, and final demo application is described in Chapter 6. Firmware has been slightly modified to exploit the hardware features. Final demo application proves right operation of CR functions applied to WSN communications.

After carrying out all the exposed tasks, conclusions about viability, value and utility about the implemented device came out. Most important points are:

- Modularity achieved with the cNGD enables robust foundations to build over new designs easily. This makes wider the possibilities for test-bed platform applications and facilitates CWSN research.
- Flexibility is an important concept affecting both communications and applications. Three RIs make the cNGD the only WSN test-bed platform capable to access three different frequency bands. Applications might implement a great range of functionalities making use of the expansion slots that cNGD encrusts.
- Scalability is a fact since the used MiWiTM protocol provides support for variable network sizes up to 8000 nodes at its PRO version. Moreover, the employed protocol gives support for both P2P and mesh topologies.

- Firmware employed supposes an efficient way to deal with multiple interfaces. A common MiWiTM protocol stack is shared among RIs and this supposes computational and memory savings. The firmware also provides a HAL to deal easily with the hardware features.
- The MCU is a Microchip PIC32 due to firmware requirements. The model chosen supposes the less featured MCU fulfilling minimal peripherals needed and offering a wider pin-out to avoid pin-multiplexing incompatibilities.
- Power switches driven by the MCU control the power supply at the RIs. These have been shown as a really valuable function at energy saving modes. Sleeping modes are successfully implemented and allow a great autonomy using the provided battery. On the other hand, RIs have shown a very low power consumption behavior, what also helps to extend the autonomy.
- Size achieved at the platform, together with anchorage options reveals a great usability and provide easiness facing application developments.

8.1. Further Studies

Facing the future of the platform, it is important to pay attention to some points. These points have been considered as highly convenient in order to increase performance and obtaining a better utilization of the system.

- To exploit the real possibilities that cNGD platform offers, a real test-bed deployment is required. Real cognitive strategies and performance must be evaluated in realistic scenarios. An implementation consisting in a standard sized WSN using cNGD nodes would be the next step to recreate a realistic CWSN. A test-bed deployment brings some necessities that are exposed throughout these future lines.
- It is important, specially when facing a test-bed deployment, to establish an OTAP (*Over The Air Programming*) system. This would facilitate the task of getting all the test-bed nodes programmed. In addition, cNGD is hardware-capable to host an OTAP system. Just proper software is required.
- Since keeping an 802.11 interface is desirable in a WSN, a custom shield could give this possibility to the cNGD. Expansion options on board the cNGD gives chances for Ethernet connections. Moreover, facing the test-bed deployment, an 801.11 gateway becomes essential for some kind of IP access, control and storage.
- CRmodule integration at the current firmware version is essential for a test-bed implementation. This module is responsible to manage all the cognition related data-flow and control. It supposes an indispensable portion to make create a real CWSN.
- Some firmware modifications claims to be done. A firmware adaptation to notify low-battery state might be an easy, valuable and useful implementation. On the other hand, a complete function to change RI bit-rates while operating would a powerful tool for CWSN investigation and would provide extra flexibility. On the other hand, functions to search for PANs and establish connection to them

is still needed. USB tracing modes still need for a further work, a circular buffer is needed. Furthermore, energy thresholds and zero-levels still need adjustments when sensing the radio spectrum.

- Once the line of RI designing was started, there is room to keep improving the design. Giving to the current design the chance to swap its antenna impedance matching circuitry, would enable the transceiver to operate over both 434 and 868 MHz. Currently there are not commercial tuneable transceivers for WSN. It even might include a wake-on radio system like the one proposed in [110].
- Try new chip MRF24XA[111], launched during the development of this project, might bring a better performance on communications over 2.4 GHz. A corresponding review must be taken, specially when facing the packet losses fact at unicast mode.
- Studying different possible antennas and respective performance at μ Trans RI might provide perspectives to increase performance and therefore, improve communications.
- Many possible implementations are possible for the expansion options at the cNGD. Typical sensor modules on WSN applications could be embedded into a shield. Different gateways or communication options such as 3G, GSM (*Global System for Mobile*) or GPRS (*General Packet Radio Service*) could be of interest for certain applications. Even developing a generic empty board with soldering slots could be a cheap way to prototype shields.
- A review over the obtained performance and possibilities of less featured MCU, even changing the used architecture might set some guidances for future improvements on the platform, despite this supposes a firmware redesign.
- Optimization of the design. Current design is susceptible to suffer design optimizations. RJ-11 PGE programmer could be replaced by a simpler, smaller and cheaper option, or even be included at the header. Creating an unified clock signal for MCU and transceivers is possible and it would save energy on clock signals generation.
- A good complement for the OTAP system could be a wireless console system. This wireless console could provide wireless tracing of the platform or even debugging. This tool would be valuable facing application developments.
- It still exists possibilities for firmware improvements. An easier access to available peripherals at the headers might be carried out, or expand options through USB interface.

Chapter 9

Estimated Costs

*Human, human, human, human,
Human, human, human, human,
Human, human, human, human,
Human, human, human after all.*

Thomas & Guy-Manuel, Daft Punk

9.1. Physical implementation cost

Software	
Concept	Cost
MS Windows 7 Home	79.76 €
Altium Designer 10	195 €
Total	247.76 €

Hardware resources			
Concept	Cost	Amortization	Real cost
Personal computer	800.00 €	20 %	160.00 €
Laser printer	100.00 €	2 %	2.00 €
MPLAB ICD3 Programmer	200.00 €	2 %	4.00 €
Office consumables	50.00 €	100 %	50.00 €
Electronic working tools	50.00 €	100 %	50.00 €
Agilent E3644A Power supply	200.00 €	2 %	4.00 €
Tektronix TDS5054B Oscilloscope	10800.00 €	2 %	216.00 €
Bausch & Lomb StereoZoom 4 Binocular Microscope	324.00 €	2 %	6.48 €
JBC AM 6000 Soldering Station	2500.00 €	2 %	50.00 €
PCB manufacturing			
μTrans PCB x 20	213.31 €	100 %	213.31 €
cNGD PCB x 10	126.03 €	100 %	126.03 €
rs232SHIELD PCB x 5	95.04 €	100 %	95.04 €
chargerSHIELD PCB x 5	110.75 €	100 %	110.75 €
Electronic components	500 €	100 %	500 €
Total			1587.30 €

Labor costs			
Concept	Daily Cost	Days	Total cost
Full time engineer	113.32 €	220	24930.40 €
Social expenses (31.6 %)	35.80 €	220	7876.00 €
Total			32806.40 €

Physical implementation costs	
Software resources	247.76 €
Hardware resources	1587.30 €
Labor costs	32806.40 €
Printing and binding	195.00 €
Total	34836.43 €

9.2. Overhead and industrial profit

Overhead and industrial profit	
Physical implementation cost	34836.43 €
Overhead (15 %)	5225.47 €
Industrial profit (6 %)	2090.19 €
Contracted operation budget	42152.12 €

9.3. Fees for project management and writing

Fees for project management and writing	
Accumulated subtotal	42152.12 €
Writing fees (5.6 %)	2360.52 €
Management fees (5.6 %)	2360.52 €
Total	46873.15 €

9.4. Total budget

Total Budget	
Accumulated subtotal	46873.15 €
Value-added tax (21 %)	9843.36 €
Total	56716.52 €

The estimated cost for this project amounts to FIFTY-SIX THOUSAND SEVEN HUNDRED AND SIXTEEN POINT FIFTY-TWO euros.

Madrid, October 2013
Project Head Engineer

Signed: Agustin Tena Garcia
Telecommunication Engineer