Technic University of Madrid

Higher Technical School of Telecommunication Engineering

FINAL THESIS REPORT

M.S. Degree in Telecommunication Engineering

Development of a multiple RF interfaced platform for Cognitive Wireless Sensor Networks

Agustín Tena García

September 2013

Document formatted with TEXIS v.1.0+.

This document is prepared for duplex printing.

Final Thesis Report

Title:	Development of a multiple RF interfaced platform for Cognitive Wireless Sensor Networks
Author:	Agustín Tena García
Advisor:	Elena Romero Perales
Inspector:	Alvaro Araujo Pinto
Departament:	Electronic Engineering Department

Thesis Committee

President:	D. Álvaro de Guzman Fernández González
Member:	D. Alvaro Araujo Pinto
Secretary:	D. Pedro José Malagón Marzo
Assistant:	D. Miguel Ángel Sánchez García

SCORE:

It is certified that Agustín Tena García has successfully accomplished his Final Thesis Report, thus obtaining the M.S. Degree in Telecommunication Engineering at Higher Technical School of Telecommunication Engineering, Technical University of Madrid, Spain.

Madrid, a de de

Development of a multiple RF interfaced platform for Cognitive Wireless Sensor Networks

AUTHOR: Agustín Tena García

ADVISOR: Elena Romero Perales

INSPECTOR: Alvaro Araujo Pinto

Electronic Engineering Department
Higher Technical School of Telecommunication Engineering
Technic University of Madrid

September 2013

A mis abuelos

Acknowledgements

Abstract

Nowadays, WSNs (Wireless Sensor Networks) are subject to development constraints and difficulties, such as the increasing RF spectrum saturation. This hinders the deployment of Wireless Ad-hoc Sensor Networks, especially for critical and sensitive applications.

CNs (Cognitive Networks), leaning on a cooperative communication model, represent a new paradigm aimed at improving wireless communications. CWSNs (Cognitive Wireless Sensor Networks) compound cognitive properties into common WSNs, thereby developing new strategies to mitigate difficulties arising from the constraints these networks face regarding energy and resources.

It is important to investigate cognitive models to explore their benefit over our WAHSNs. However, few platforms allow their study due to their early research stage, and they still possess scarce or specific features. Investigations take place mainly over simulators, which provide partial and incomplete results.

This paper presents a versatile platform that brings together cognitive properties into WSNs. Hardware and software models are combined to create one instrument that is used to investigate CWSNs. The hardware fits WSN requirements in terms of size, cost, and energy. It allows communication over three different RF bands, and it thus the only cognitive platform for WSNs with this capability. In addition, its modular and scalable design is widely adaptable to almost any WAHSN application.

KEY WORDS: *cognitive networks*, *cognitive radio*, *wireless sensor network*, *platform*, *testbed*, *cognitive wireless sensor network*.

Index

List of Figures

List of Tables

Chapter 1

Testings and Results

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

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 valuable the platform 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.

1.1. Test Model

The purposes of testing functionalities of the platform are to prove valuable capabilities and proper functionality of modules. Characterization is not expressely 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, cosult **??**.

Tests are simple and check bounded functions. Aspects to check are related to sleeping modes operation and current consumption, microcontroller computing capabilities, spectrum sensing features and radio interfaces, or battery charger behaviour.

1.2. Energy management test

This test tries to bring the cNGD (*Cognitive Next Generation Device*) under different operation modes. On the one hand, different sleeping modes and energy options for RI (*Radio Interface*)s and MCU (*Microcontroller*) 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 2.2.4.

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

- 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 1.1, where a graph shows how the current consumption goes down throughout the different set situations.

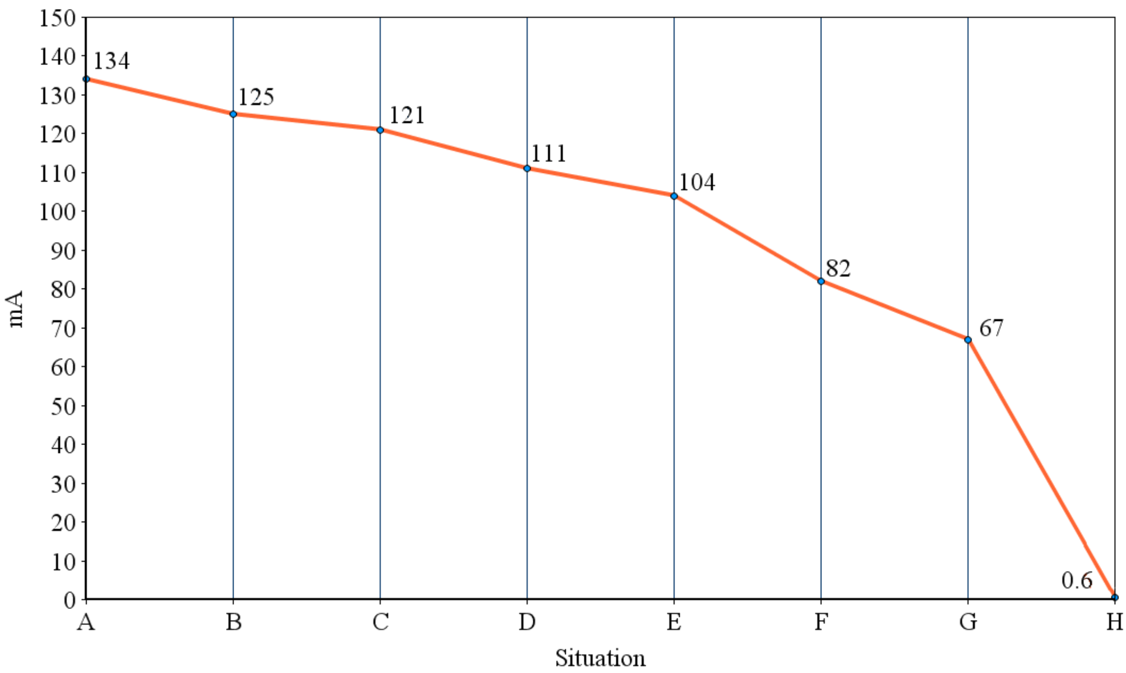


Figure 1.1: cNGD consumption at different energy modes.

1.3. 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 planification 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 1.1 shows separately the measured time for RIs and different protocol versions, Mesh and P2P.

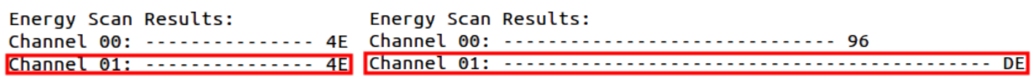
Management part	P2P Protocol	MiWi Protocol
Common management		
434 MHz RI management		
868 MHz RI management		
2.4 GHz RI management		
Total time		

Table 1.1: Time costs on the communication protocol management tasks.

1.4. 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 (*Hardware Abstraction Layer*), three simple scenarios show sensing features at the three different frequency bands. All the scenarios host a device transmitting continously unicast packets at determined channels. This RF (*Radio Frequency*) 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 bitrate of 119,2 kbps. Transmitting device is making use of channel 1. Figure 1.2 shows the energy scan traces deployed by the firmware.



(a) Before start transmitting packets.

(b) After start transmitting packets.

Figure 1.2: Spectrum sensgin at 434 MHz.

First scenario is conducted over 868 MHz band. This band posses 7 available channels when using a bitrate of 119,2 kbps. Transmitting device is making use of channel 4. Figure 1.3 shows the energy scan traces deployed by the firmware.

First scenario is conducted over 2.4 GHz band. This band posses 16 available channels. Transmitting device is making use of channel 4. Figure 1.4 shows the energy scan traces deployed by the firmware.

```
Energy Scan Results:
Channel 00: --------------- 4E
Channel 01: -------- 2A
Channel 02: --------------- 4E
Channel 03: -------- 2A
Channel 04: -------- 2A
Channel 05: -------- 2A
Channel 06: -------- 2A
```

(a) Before start transmitting packets.

```
Energy Scan Results:
Channel 00: --------------- 4E
Channel 01: --------------- 4E
Channel 02: --------------- 4E
Channel 03: --------------- 4E
Channel 04: ------------------------------------------- DE
Channel 05: --------------- 4E
Channel 06: --------------- 4E
```

(b) After start transmitting packets.

Figure 1.3: Spectrum sensgin at 868 MHz.

```
Energy Scan Results:
Channel 00: --- 0F
Channel 01: ----- 1D
Channel 02: ------- 26
Channel 03: ---------------- 54
Channel 04: ---- 16
Channel 05: ------------------------------ 99
Channel 06: ---------------------------------- AC
Channel 07: ---------------------------------------- C8
Channel 08: -------------------------------- A3
Channel 09: ------------- 41
Channel 10: ------- 27
Channel 11: ----------------- 58
Channel 12: ------------------------ 78
Channel 13: ----------------- 56
Channel 14: ----------- 39
Channel 15:  03
```

(a) Before start transmitting packets.

```
Energy Scan Results:
Channel 00: --- 0F
Channel 01: ----- 1D
Channel 02: ------- 25
Channel 03: --- 10
Channel 04: ----------------------------- 95
Channel 05: -------------------------------- A2
Channel 06: ------------------------------------ B4
Channel 07: ---------------------------------------- CA
Channel 08: -------------------------------- A1
Channel 09: ------------- 41
Channel 10: -------------- 47
Channel 11: ----------------------- 73
Channel 12: -------------------------- 86
Channel 13: --------------------- 6C
Channel 14: -------------------------- 85
Channel 15: ----- 1B
```

(b) After start transmitting packets.

Figure 1.4: Spectrum sensgin at 2.4 GHz.

1.5. Radio interfaces agility test

This test evaluates how long changes over the RIs take to be done. Parameters to change are TX (*transmission*) power, operation channel and energy modes. This test was fully available at the firmware. Table 1.2 ilustrates 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 μs	55.8 μs	70.1 μs
Sleep and wake up	15.4 ms	15.4 ms	0.364 ms

Table 1.2: Time costs on the communication protocol management tasks.

1.6. Effective rate test

Effective rate provided by the MRF49XA and MRF24J40 transceivers driven by the firmware was already described in [1]. 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 unchecked 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 MiWi^{TM} 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.

Tables 1.3 and 1.4 ilustrate the results at two different bit-rates for μTrans module at 434 MHz and 868 MHz. Both versions, sharing the same transceiver, posses similar performance. Used protocol is P2P.

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 1.3: 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 1.4: 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 raugh 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 suposse a change with respect to previous tests.

Table 1.5 ilustrates the results for MRF24J40 transceiver as 2.4 GHz RI using a P2P protocol.

A worrying issue, already described in [1], 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 due to the unsequenced character of ACK packets at MiWi protocol.

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 1.5: Effective rate, received packets comparison for 2.4 GHZ RI using P2P protocol.

In order to solve this issue, Jara, G. presents in [2] a flow control mechanism at MiWi MAC (*Medium Access Control*) layer that apparently avoid packet loss at unicast mode.

1.7. chargerSHIELD performance test

As it was mantioned, small checkings 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 1.5 shows the average results, battery voltages are view as blue bars referened on the left side while current flow is expressed with an orange line respect to values at the right side.

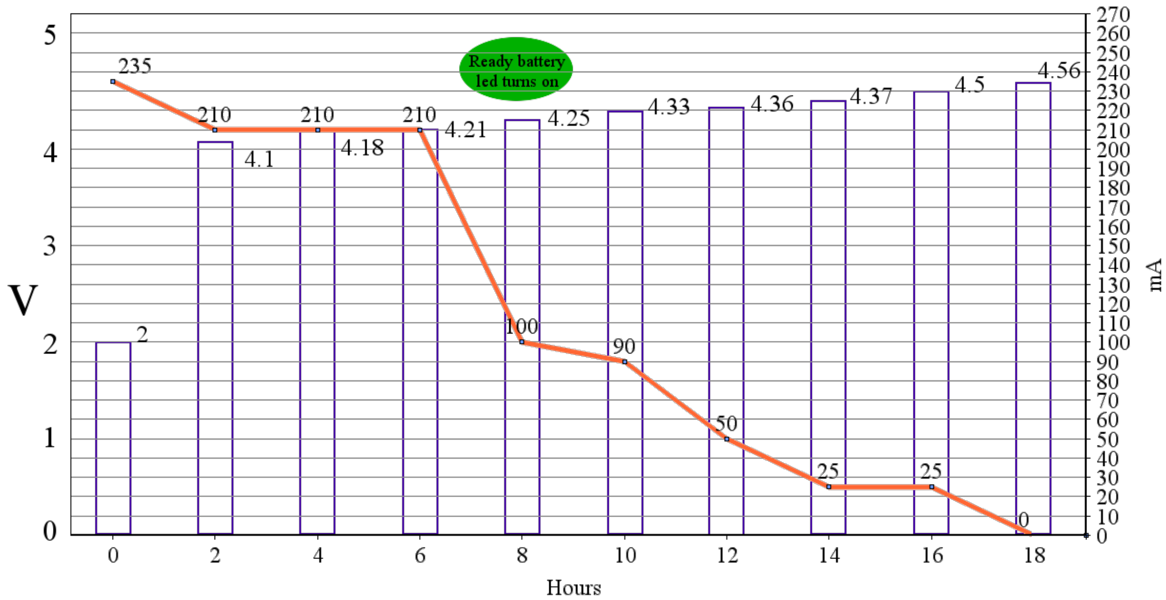


Figure 1.5: Batteries and parameters evolution during charging process at chargerSHIELD.

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. It might be extrange to observe so high voltage values at the batteries when they are supposed to provide 3.6 V, nevertheless, it is a normal behaviour. The charger properly switchs off the charging current when voltage at the battery reveals a complete charge.

Chapter 2

Software

Too long, can you feel it?
Too long, oh can you feel it ?

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 behaviour of the final demo application.

2.1. Tools

2.1.1. MPLAB X

MPLAB X IDE (*Integrated development environment*) is a multiplatform 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.

Figure 2.1: MPLAB X IDE logo.

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.

2.1.2. Programmer: ICD 3

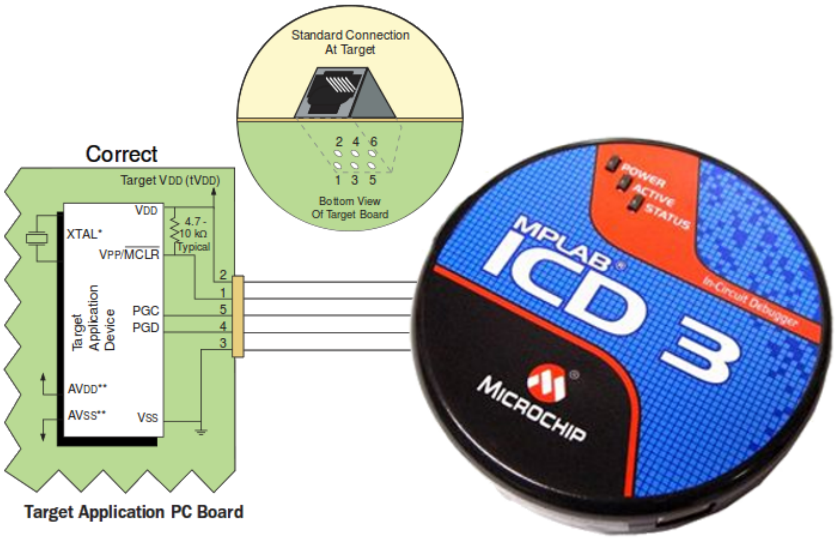


Figure 2.2: ICD3 programmer configuration.

ICD (*In-Circuit Debugger*) 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[3].

2.2. Firmware and Hardware Abstraction Layer

When adapting the 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 initiallization of the MRF49XA transceivers, a 50 ms delay after setting the reset signal was included.

Other adaptations taken when developing software aim at stablish better ease of use. An easier configuration scheme, USB (*Universal Serial Bus*) 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 in a public GitHub[1] repository: *https://github.com/agus-xyz/cognitiveNGD*. Figure 2.3 provides a QR (*Quick Response*) code to instantly access the repository.

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

Figure 2.3: cNGD software repository.

2.2.1. Platform options

When configuring the firmware to operate over different hardware platforms such as the FCD (*First Cognitive Device*) 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 stablished over *MRF49XA_1* transceiver, *MIWI_0868_RI* is built over *MRF49XA_2*, and *MIWI_2400_RI* does it on *MRF24J40* transceiver. 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 (*Serial Peripheral Interface*) 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 atomatically configured when set on of the previous options. Main changes take place over RI configurations and pinouts.

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.

2.2.2. 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 μUSB 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.

Adaptated example offered possibilities for either polling or interruption methods, interruption management was chosen. A circular buffer whould 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 migh arise.

To configure the firmware for USB traces, firstly it is needed to set the *ENABLE_ CONSOLE* option. Then, the console must be stablished 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 [4] or stack[5] manual can be consulted. For information related to USB-CDC, manual [6] might be useful.

2.2.3. 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 firmware 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.

2.2.4. Headers and other new definitions

For an easy use and programmation 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 *HEADER_ XX_ TRIS* to configure the sort of pin. *XX* parameter must be replaced with the desired pin number. For instance, *HEADER_ 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 *HEADER_ 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.

2.3. Demo Application Layer

Demo application layer was thought to be showed during these Master thesis presentation. It implements a WSN (*Wireless Sensor Network*) 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.
- Bitrate for 434 MHz RI is 19.2 kbps, and 119.2 kbps for 868 MHz RI.
- Unicast communication mode, to make sure of the right messages dispacthing.
- The two devices posses different addresses and a common PAN (*Personal Area Network*) id.
- Sleeping modes are disabled for proper PAN management.

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

- RX (*Reception*) device is shwitched on. It initalises 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. When
- 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 successfull 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 **********************. For better comfort, Figure 2.4 provides access to the video.

Figure 2.4: Demo application video.

Chapter 3

Conclusions

Television rules the nation.
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 CWSN (*Cognitive Wireless Sensor Network*)s. 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 MiWi^{TM}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 WAHSNs applications. Main module is called cNGD. This main module posses different functionalities that enhances 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 μTrans 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 downtrend 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.

After carrying out all the exposed tasks, conclusions about viability, valuability 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 platform applications and facilitates CWSN research.
- Flexibility is an important concept affecting both communications and applications. Three RIs make the cNGD the only WSN platform capable to access three different frequency bands. Applications might implement a great range of functionalities making use of the expansion slots that cNGD incrusts.
- Scalability is a fact since the used MiWi^{TM} 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 MiWi^{TM} 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 pinout 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 succesfully implemented and allow a great autonomy using the provided batery. On the other hand, RIs have shown a very low power consumption behaviour, 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.

3.1. 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 neccessities that are exposed throughout these future lines.
- It is important, specially when facing a test-bed deployment, to stablish 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 (*Internet Protocol*) access, control and storage.
- CRmodule (*Cognitive Radio Module*) 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 stablish connection to them is still needed. USB tracing modes still need for a further work.
- 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 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 [7].
- Try new chip MRF24XA[8], 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 μTrans 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 advantages and disadvantages of changing the used architecture might set some guidances for future improvements on the platform. Even designing a customized protocol from the scratch should be discussed.
- 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 complemen 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 4

Estimated Costs

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

Thomas & Guy-Manuel, Daft Punk

4.1. Physical implementation cost

Software

Concept	Cost
MS Windows 7 Home	79.76 €
Altium Deisgner 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 €

4.2. 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 €

4.3. 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 €

4.4. 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

Fdo.: Agustin Tena Garcia
Telecommunication Engineer

Part I

Appendixes

Appendix A

Schematics

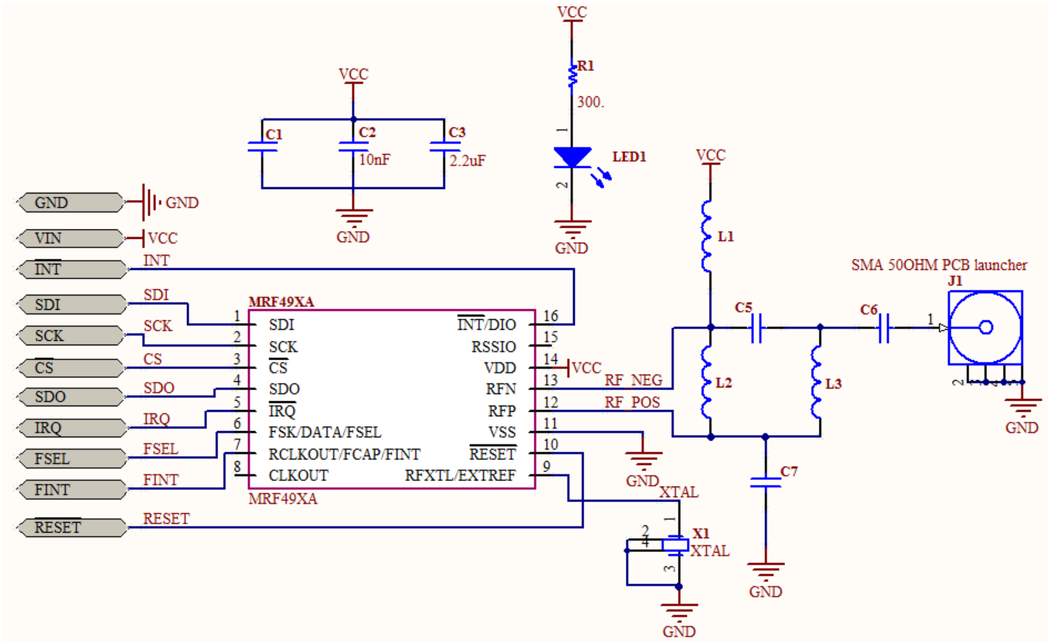


Figure A.1: μTrans 434/868 schematic

Freq	C1	L1	L2	L3	C5	C6	C7
434 MHz	220 pF	390 nH	33 nH	47 nH	2.7pF	68 pF	5.1 pF
868 MHz	47 pF	100 nH	8.2 nH	22 nH	1.2 pF	27 pF	2.7 pF

Table A.1: Values for balun components at different frequency bands.

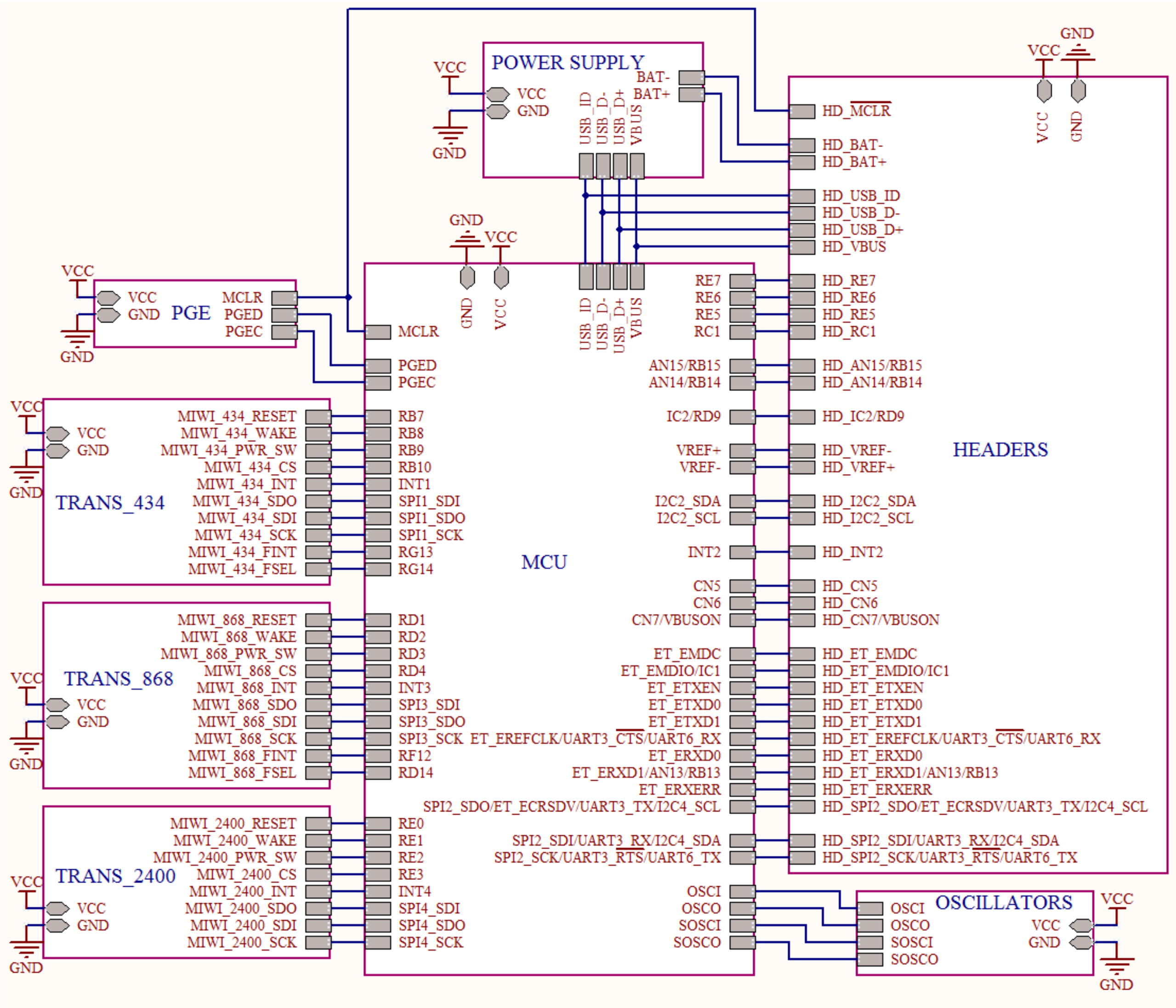


Figure A.2: cNGD general schematic

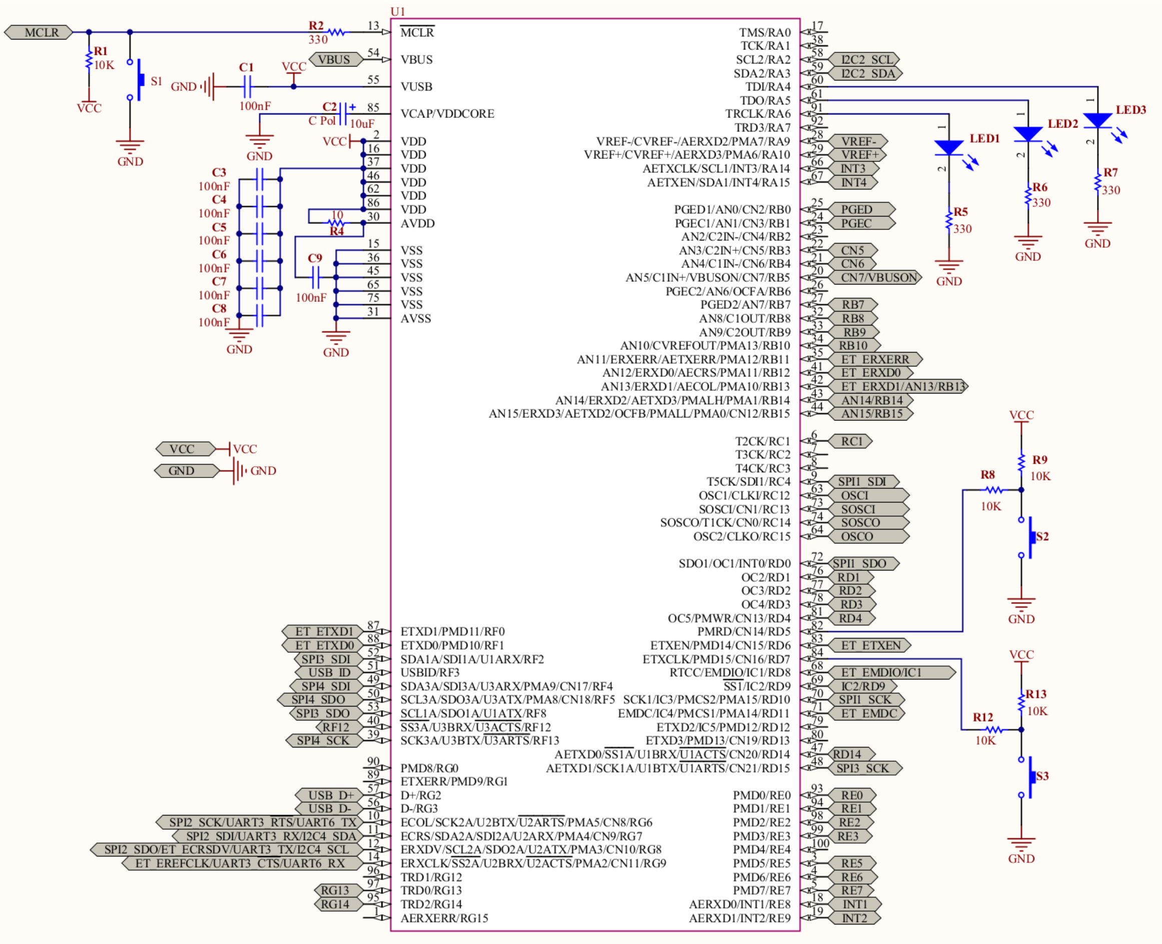


Figure A.3: Microcontroller specific schematic

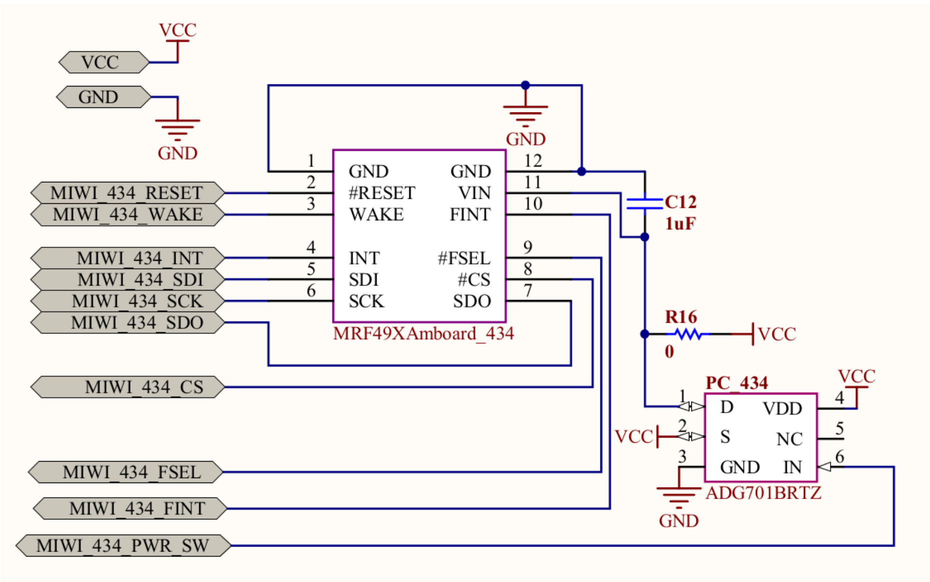


Figure A.4: 434 MHz Radio Interface systems schematic

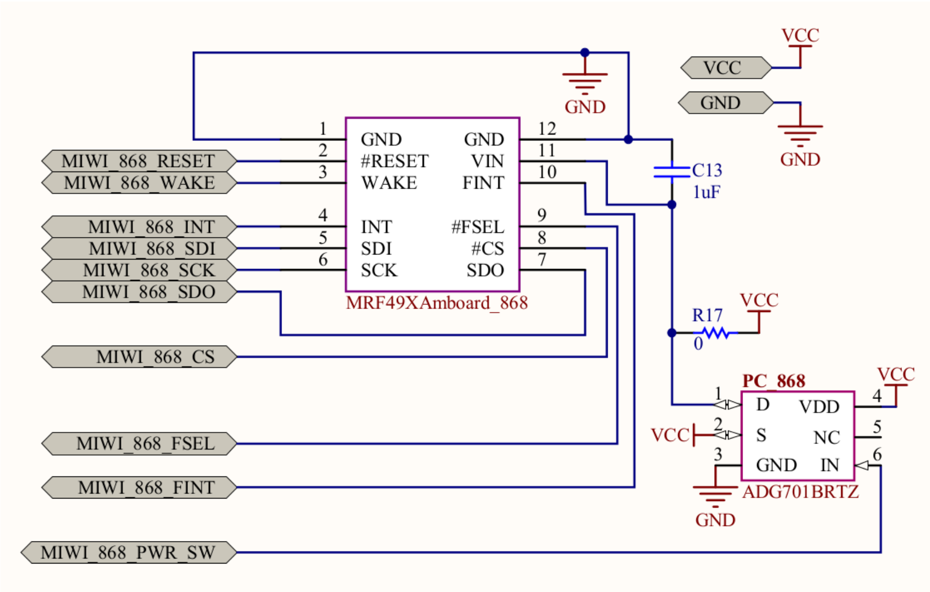


Figure A.5: 868 MHz Radio Interface systems schematic

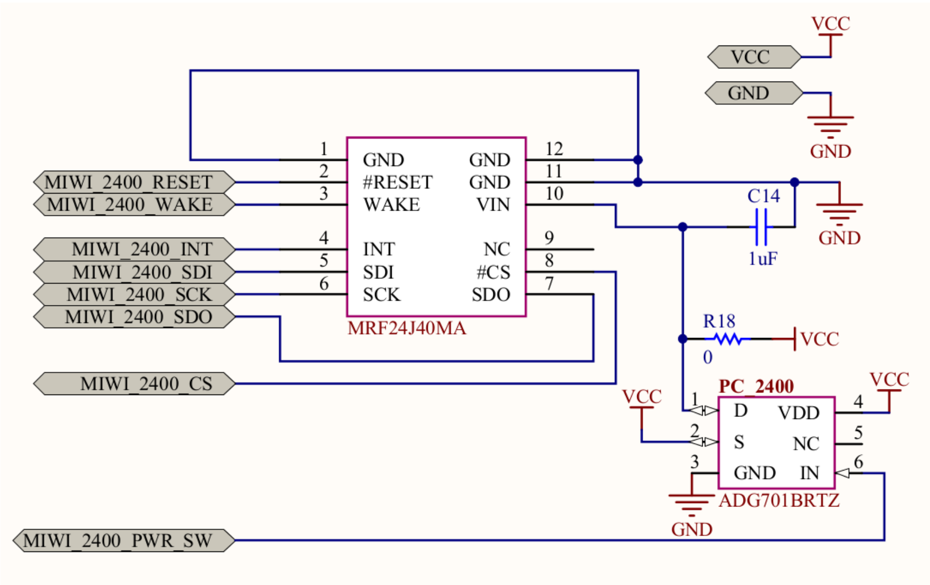


Figure A.6: 2.4 GHz Radio Interface systems schematic

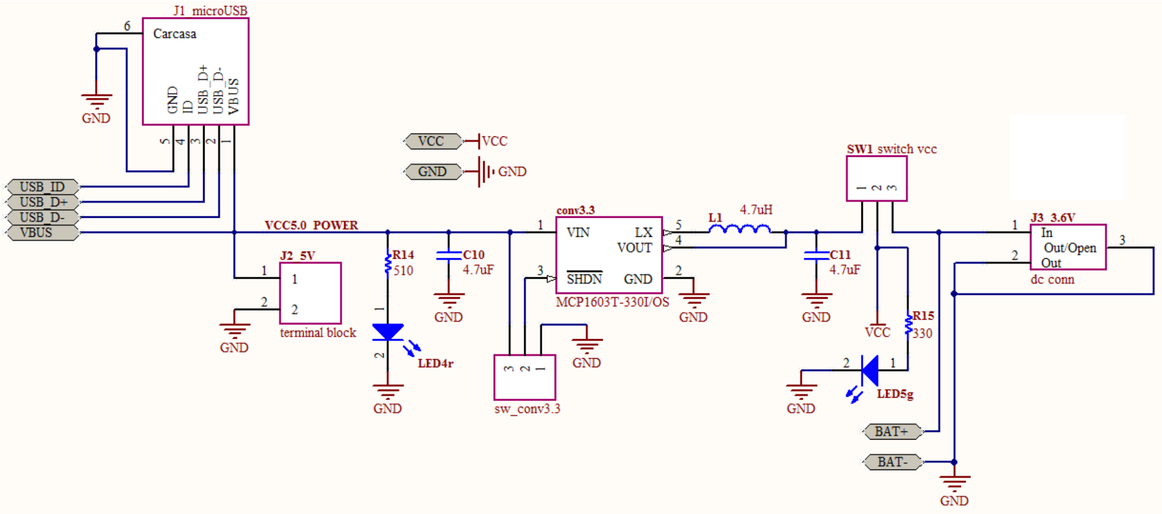


Figure A.7: Power supply system schematic

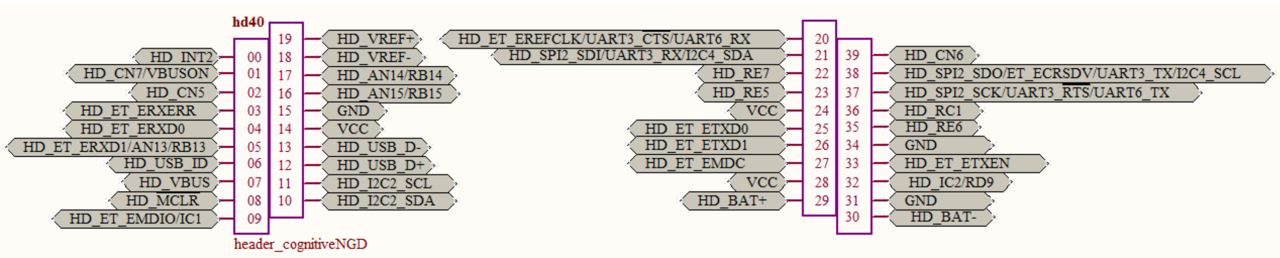


Figure A.8: Expansion header system schematic

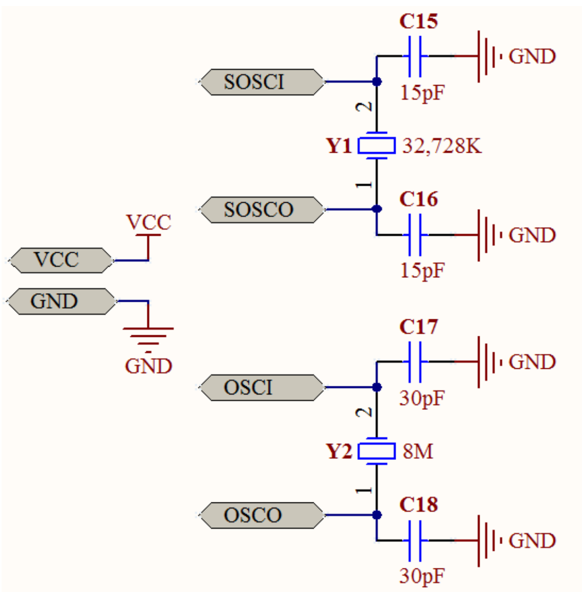


Figure A.9: Oscillator system schematic

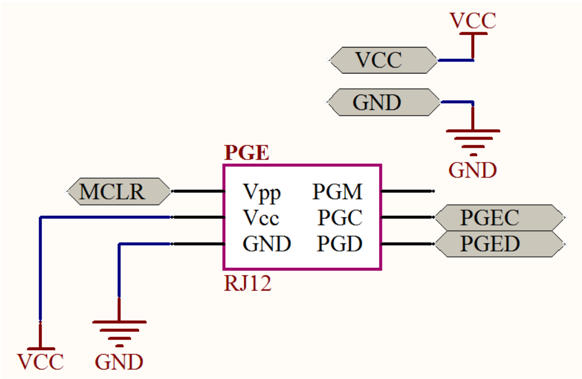


Figure A.10: PGE system schematic

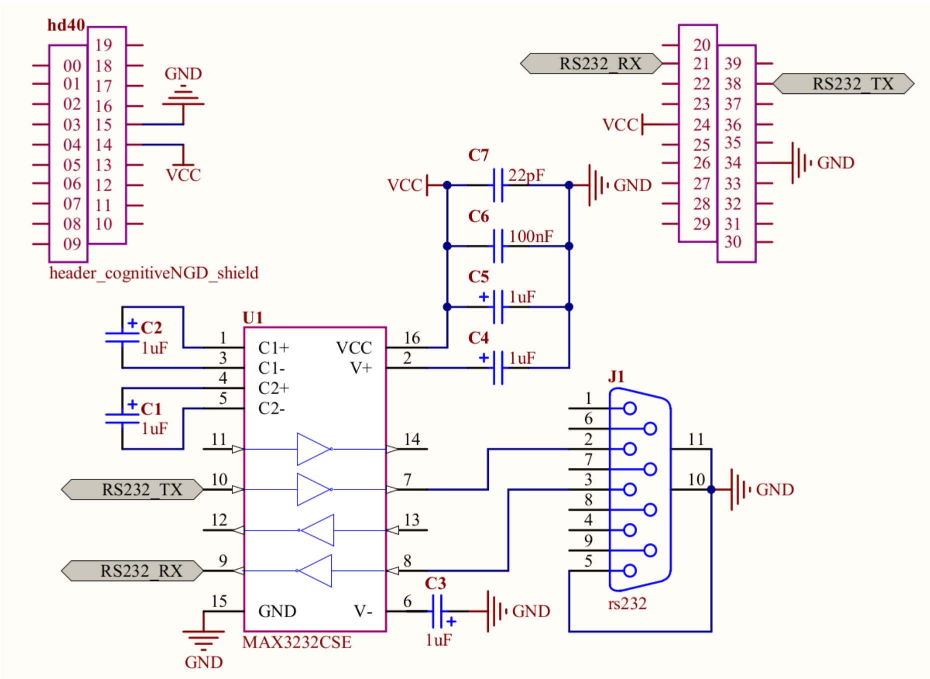


Figure A.11: rs232SHIELD schematic.

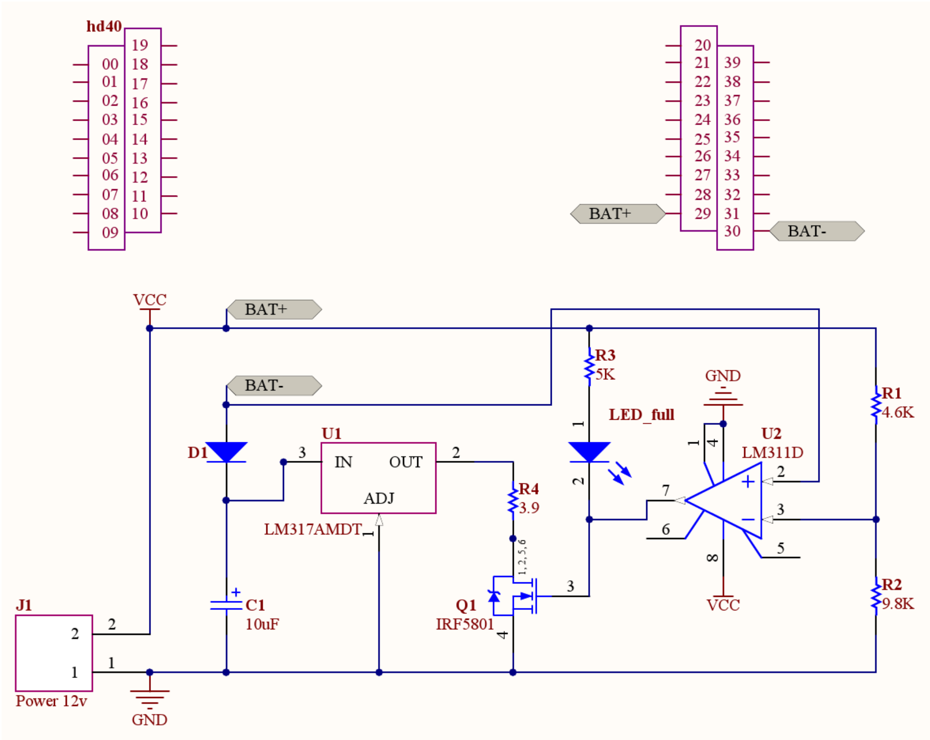


Figure A.12: chargerSHIELD schematic.

Appendix B

Layouts & components

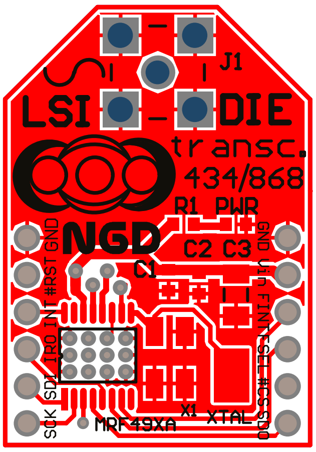


Figure B.1: μTrans Top layers.

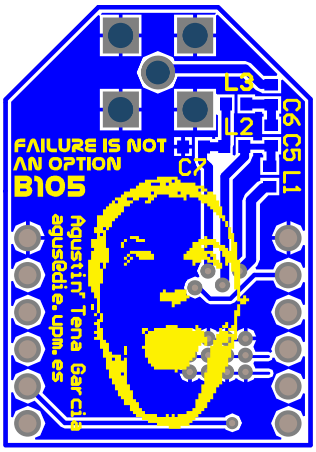


Figure B.2: μTrans Bottom layers.

Symbol	Nominal Value (434/868)	Description	Package Type	Quant.
C2	0.01μF	Ceramic Capacitor	0603	1
C3	2.2μF	Tantalum Capacitor	1206	1
X1	10MHZ	Quartz Crystal	5 x 3.2 mm SMD	1
MRF49XA		MRF49XA-I/ST - Transceiver, RF	16-TSSOP	1
J1		50 Ω Coaxial SMA Jack	Through Hole	1
D1		Light Emitting Diode	0603	1
R1	300 Ω	Resistor	0603	1
C1	220 pF/47 pF	Ceramic Capacitor	0603	1
C5	2,7 pF/1,2 pF	Ceramic Capacitor	0603	1
C6	68 pF/27 pF	Ceramic Capacitor	0603	1
C7	5.1 pF/2,7 pF	Ceramic Capacitor	0603	1
L1	390 nH/100 nH	Inductor	0603	1
L2	33 nH/8.2 nH	Inductor	0603	1
L3	47 nH/22N nH	Inductor	0603	1
J1	434MHz/868MHZ	Monopole Antenna	SMA(M)	1

Table B.1: Component description for μTrans 434/868.

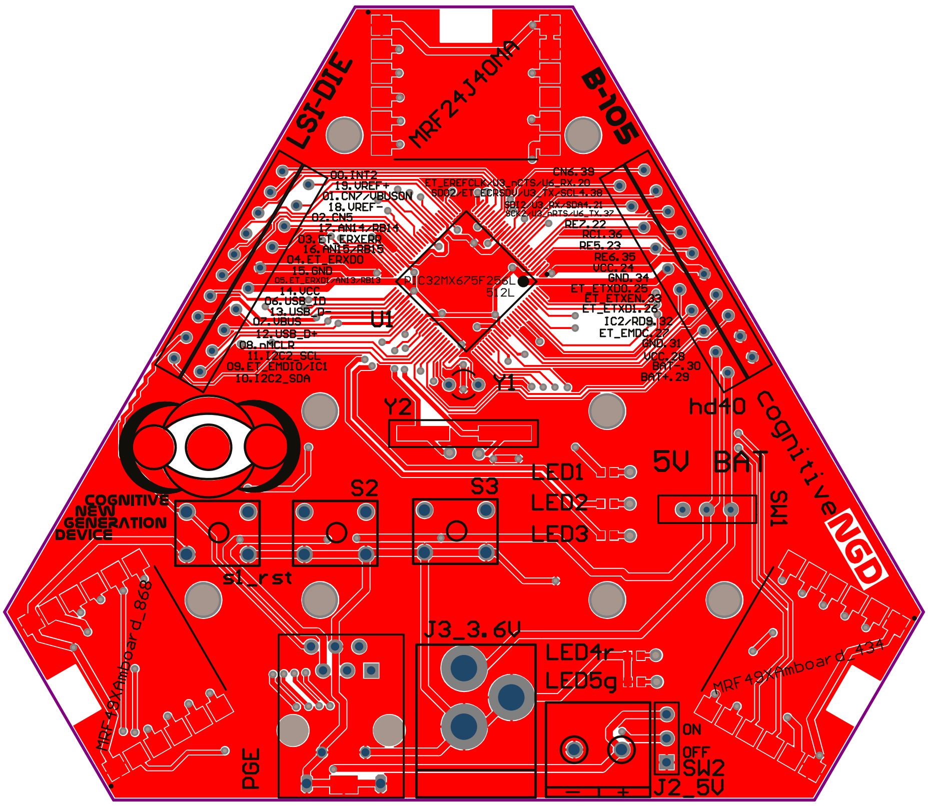


Figure B.3: cNGD Top layers.

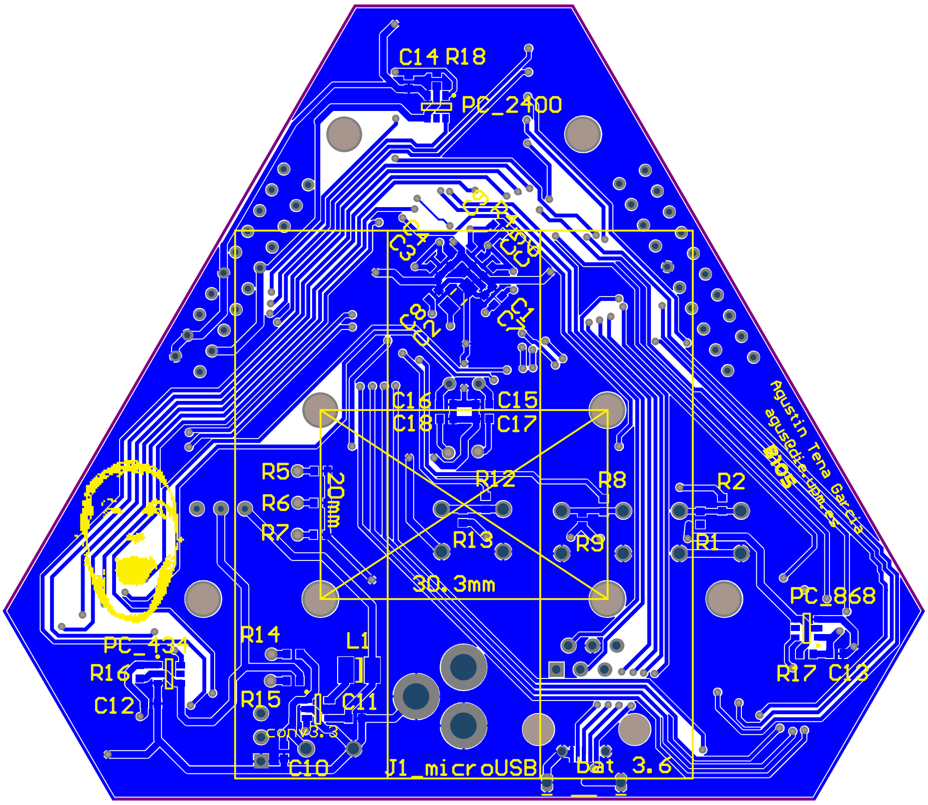


Figure B.4: cNGD Bottom layers.

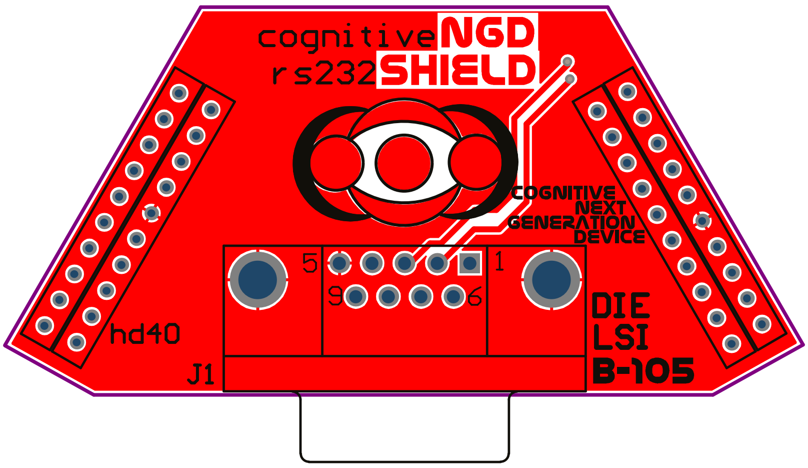


Figure B.5: rs232SHIELD Top layers.

Symbol	Nominal Value	Description	Package Type	Quant.
J1		MICRO USB A/B	SMD	1
C1,3-9	100 nF	Ceramic Capacitor	0603	8
C2	10 μF	Pol. Capacitor	1206	1
C10,11	4.7 μF	Ceramic Capacitor	0603	2
C12-14	1 μF	Ceramic Capacitor	0603	3
C15,16	15 pF	Ceramic Capacitor	0603	2
C17,18	30 pF	Ceramic Capacitor	0603	2
R1,8,9,12,13	10 KΩ	Resistor	0603	1
R4	10 Ω	Resistor	0603	1
R2,5-7,15	330 Ω	Resistor	0603	5
R15	510 Ω	Resistor	0603	1
Y1	32.768 KHz	12.5pF Load capacitance crystal	Through hole	1
Y2	8 MHz	20 pF Load capacitance crystal	SMD	1
PGE		RJ-11 Connector		1
LED1-5		LED	0603	5
U1		PIC32MX675F256L-80I/PT	TQFP	1
hd40		10-pin Stackable Header		4
L1	4.7mH	Inductance	1210	1
PC_434, 868, 2400		ADG701LBRTZ - SWITCH	SOT-23	3
conv3.3		MCP1603T-330I/OS	TSOT	1
J2		Terminal Block		1
J3		DC Female Connector	Through hole	1
S1-3		Tactile Switch STSP	60x60 mm	3
sw1		Slide Switch, SPDT	Through Hole	1
bat3.6	3-cells	Battery holder	Wired	1
		Legs	M3	4
		Screw	M3	4
		Nut	M3	4

Table B.2: Component description for cNGD.

Symbol	Nominal Value	Description	Package Type	Quant.
C1,2,3,4,5	1 μF	Pol. Capacitor	0603	5
C6	100 nF	Ceramic Capacitor	0603	1
C7	22 pF	Ceramic Capacitor	0603	1
J1		DB9 female header - right angle	PCB mount	1
U1		MAX3232CD - RS232 TXRX	SOIC	1
hd40		10-pin Stackable Header		4

Table B.3: Components description for rs232SHIELD.

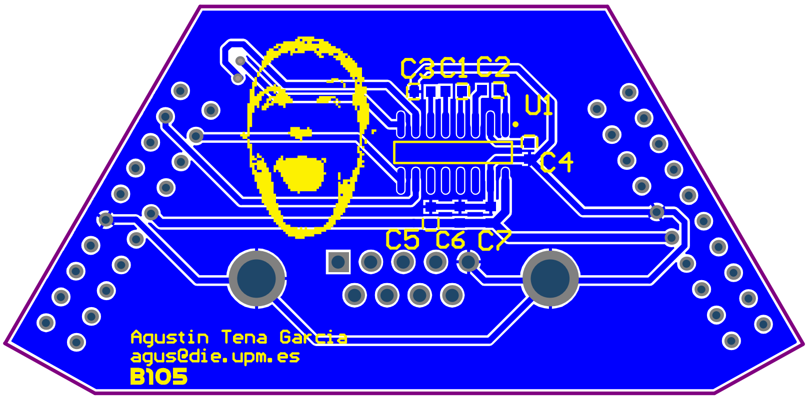


Figure B.6: rs232SHIELD Top layers.

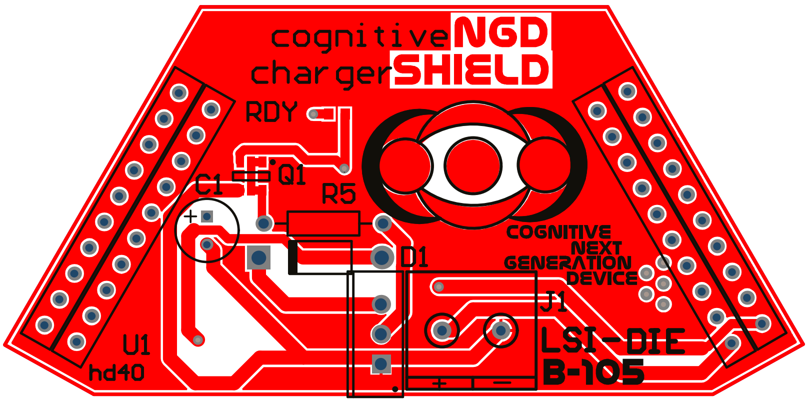


Figure B.7: chargerSHIELD Top layers.

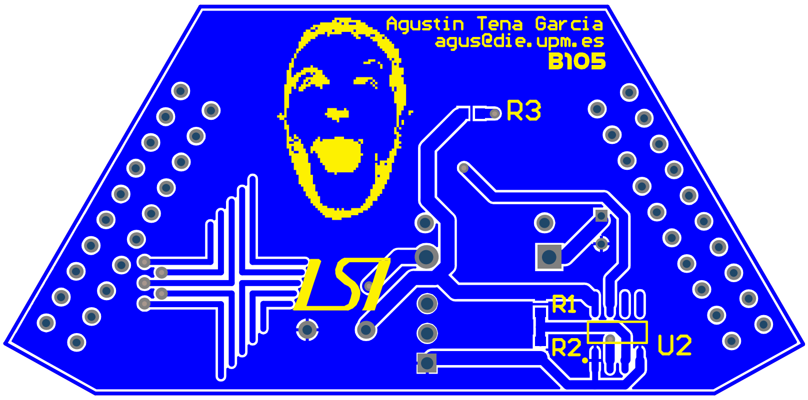


Figure B.8: chargerSHIELD Top layers.

Symbol	Nominal Value	Description	Package Type	Quant.
J1		Terminal Block		1
R3	5K	Resistor	0603	1
R4	3.9 Ω	1W Resistor	Axial	1
C1	10 μF	Terminal Block		1
U1		LM317AMDT - IC, V. REG.	TO-220	1
hd40		10-pin Stackable Header		4
Q1		MOSFET IRF5801	TSOP	1
R1	4.6 KΩ	Resistor	0603	1
R2	9.8 KΩ	Resistor	0603	1
U2		LM311D - IC, COMPARATOR	SOIC8	1
D1		DIODE, STANDARD, 0.5A	Axial Leaded	1

Table B.4: Components description for chargerSHIELD.

References

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

[1] Domingo, J., *Diseño, optimización y prueba un nodo para una red de sensores inalámbrica con capacidades cognitivas, PFC.* Laboratorio de Sistemas Integrados, ETSIT-UPM, Feb 2013.

[2] Jara, G., *Diseño e implementación de una arquitectura para la gestión de comunicaciones de una red de sensores inalámbricas cognitiva.* Laboratorio de Sistemas Integrados, ETSIT-UPM, Sep 2013.

[3] Microchip Technology Inc., *MPLAB ICD 3 In-Circuit Debugger User's Guide For MPLAB X IDE.*

[4] Microchip Technology Inc., *Microchip USB Device Firmware Framework User's guide.*

[5] Microchip Technology Inc., *AN1166 USB Generic Function on an Embedded Device.*

[6] Microchip Technology Inc., *AN1164 USB CDC Class on an Embedded Device.*

[7] Rodrí, R., *Diseño e implementación de un módulo de activación por radio frecuencia para redes de sensores inalámbricas, PFC.* Laboratorio de Sistemas Integrados, ETSIT-UPM, Dic 2001.

[8] Microchip Technology Inc., *MRF24XA Low-Power, 2.4 GHz ISM-Band IEEE 802.15.4^{TM} RF Transceiver.*

List of Acronyms

ACK .... *Acknowledgment*

CDC .... *Communications Device Class*

CMOS .... *Complementary metal oxide semiconductor*

cNGD .... *Cognitive Next Generation Device*

CRmodule ... *Cognitive Radio Module*

CWSN .... *Cognitive Wireless Sensor Network*

FCD .... *First Cognitive Device*

FFD .... *Full Function Device*

GPRS .... *General Packet Radio Service*

GSM .... *Global System for Mobile*

HAL .... *Hardware Abstraction Layer*

ICD .... *In-Circuit Debugger*

IDE .... *Integrated development environment*

IP .... *Internet Protocol*

MAC .... *Medium Access Control*

MCU .... *Microcontroller*

OTAP .... *Over The Air Programming*

PAN .... *Personal Area Network*

QR .... *Quick Response*

RF .... *Radio Frequency*

RI .... *Radio Interface*

RX .... *Reception*

SPI .... *Serial Peripheral Interface*

SPST .... *Single pole, single throw*

TX . . . . *transmission*

USB . . . . *Universal Serial Bus*

WSN . . . . *Wireless Sensor Network*