



# Daniel Rebollo Lorenzo

*Madrid, Spain*

**Embedded Systems and electronics devices prototype & development & analysis & manufacture.**

Electronics, hardware, software & firmware, IoT, FPGA engineer. IEEE member.

## Contact

**LinkedIn profile:**

<https://es.linkedin.com/in/danirebollo>

**Webpage:**

<http://danirebollo.es>

**GitHub:**

<https://github.com/danirebollo/>

**email:** dani@danirebollo.es

Embedded Systems and electronics devices development, improvements, analysis, modification & fabrication.

Proficient in electronics research, PCB design, C++ applied to microcontrollers, FPGA low level configuration (VHDL), project management and electronics manufacture.

Selling products like EGPU (formerly LCDBPV2) to clients like TESLA, MIT, Sandia Lab and other universities.

## Skills

- Electronic standalone & embedded systems development & prototype
- Design and make up PCB for fabrication, Gerber generation.
- Altium Designer design and Vault server for inventory and version control of components and boards.
- C & Python & Java programming
- Web pages (JSP & PHP & JS & MySQL databases)
- Microcontrollers: ESP32, PIC, MSP430, Stellaris, Tiva, CC3200, STM32 & AVR.
- Robotics: Bare Metal / ROS2
- Programmable logic: Discrete & FPGA & CPLD (Xilinx & Altera) with VHDL & schematics.
- Microcontroller & GPU design with VHDL
- Power electronics (linear & switch power supplies, other...)
- Communication protocols: I2C, SPI, EHA, parallel, LVDS, HDMI, VGA, USB
- Design & repair audio power amplifiers, foot pedal effects and instruments (guitar and bass).

## Laboratory

- Prototype and test in Madrid lab.
- More than 10000 component references, both mechanical and electronic chips.
- Oscilloscopes
- Logic analyzers
- Spectrum analyzers
- Power supplies
- Bluetooth analyzer
- Zigbee analyzer
- RF 433/868 analyzer
- PCB reflow oven
- Pick & place machine
- Solder stations
- PCB fabrication (Photoresist, etching, drilling, solder)
- Mechanical fabrication: FDM 3D printers, SLA 3D printers, silicon molds and polyurethane parts

Mass fabrication in Shenzhen: Agreements with different factories: batteries, electronics, mechanical, molds, assembly.

# Index

Transcript of records 3

Relevant resume 4

Projects 5

Other 15

### **Development of electronic products (bachelor degree) (2011) [Salesianos Atocha, Madrid]**

- Analog electronics
- Digital electronics (Embedded, standalone and microprocessors (PIC) )
- Power electronics
- Programming (C)
- Quality control
- Maintenance of industrial, telecommunication and embedded electronics systems
- Development and construction of electronic prototypes
- Project development
- Job training and guidance
- Orientation for small and medium-sized enterprises
- On the Job Training: Honeywell internship (Madrid-barajas airport and TorreEspacio control center)

### **Electronics degree (2015) [Technical University of Madrid (UPM)]**

- Professional Communication
- Electronics I & II
- Introduction to Telecommunications
- Programming I (C) & II (JAVA)
- Circuit analysis I & II
- Search techniques and information systems
- Telecommunication networks and services
- Microprocessors (ASM, ARM)
- Science Technology and Society
- Analog electronics
- Advanced Applications Programming (JSP)

### **Freelance: Hardware, firmware & software engineer (2015- )**

Prototype & final products develop.

Embedded systems, consumer electronics and laboratory tools.

### **Further training**

- EDX verified student into electronics embedded (UTAustinX: UT.6.02x Embedded Systems - Shape the World).
- Analog Devices seminars
- Altium Designer seminars
- Agilent test & measure seminars
- Multicore recodification video DSP systems seminar
- IEEE national congress 2014: power electronics, batteries and radio-controlled systems.
- Cross Border Mentorship Week – Hardware & TravelTech; at DCU Ryan Academy for entrepreneurs (Dublin, Ireland). 2016
- TechHub at Campus Madrid by Google. 2015, 2016
- Intel ISDF 2016
- Microsoft Tech Summit Madrid 2016
- Google Campus meetup Madrid 2016
- Electronica Munich 2016
- AWS summit 2018
- Python coursera course 2020
- IA coursera course 2022

## Relevant resume

### Programming languages

- PHP
- Java, JSP
- JS
- C, C++
- Python

### Databases

- MySQL / MariaDB
- InfluxDB

### Digital logic

- VHDL
- Schematics design and RTL comprehension.

### Microcontrollers

- ESP32 / ESP8266
- Stellaris & Tiva (ARM) / Tiva connected (Ethernet)
- CC3200 (WiFi ARM)
- MSP430
- NXP Xpresso LPC1114 (ARM)
- PIC16 / PIC32
- STM32
- Nordic NRF

### Digital buses

- I2C / SPI
- CAN
- RS232
- UART
- EHA
- SPDIF / I2S
- LVDS / HDMI / VGA
- USB

### Sensors

- PIR motion
- IMU / accelerometer / compass / magnetometer
- GPS
- RFID
- LIDAR
- Motor encoders
- Temperature / humidity / pressure

### RF communications

- WiFi
- Zigbee
- SubGHZ
- LoRa
- ISM RF / SDR

### Robotic

- ROS2
- baremetal (C++ MCU/FPGA and sensors)

### Serverless

- Azure / AWS / GCloud / Lambda – Functions / IoT Hub

### FPGA & CPLD

- Altera MAX II / 10 / Cyclone IV
- Xilinx XC9536 / Spartan 3 / Spartan 6 / Zynq

### DSP

- TMS320C5505
- ADAU1701

### Design tools

- PCB: Altium Designer (schematics, PCB, FPGA, gerber generation, assembly (Altium Nexus))
- OrCAD, Multisim, PSpice, Eagle Kickad
- Solidworks (mechanical)

### Integrated development environment (IDE)

- Eclipse
- Code Composer Studio
- VSCode / Platformio / ESP-IDF / Arduino
- Xilinx ISE
- Altera Quartus
- Altium Designer (VHDL)

### Operating systems

- Windows
- Debian / Ubuntu / FreeBSD / CentOS

### Analytics

- Grafana

### Web & CMS

- Wordpress
- Joomla

### Project management

- Trello
- Notion
- Jira
- Dolibarr

### Other

- Proxmox / LXC / VM
- Docker
- GIT
- Cloudflare DNS / Workers / Tunnel
- FDM / SLA 3D printer
- Advanced knowledge of screen / LCD protocols

### Other skills

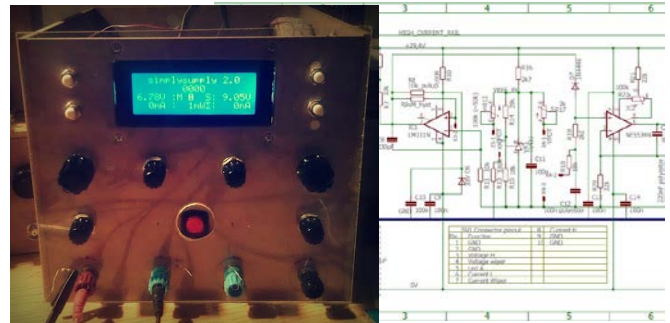
- Hardware design (analog, digital, power electronics (linear & switch power supplies, other...))
- Project management
- PCB manufacture
- Product manufacture (electronics, plastics, wires, programming, assembly...)
- 3D print (design and print)
- Machining (Wood / Metal)
- ABS mould design & parts production

## Projects

### SimplySupply (2012)

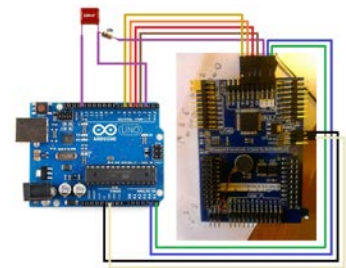
Dual trackable and isolated linear power supply with all digital control system (ADC, LCD, relays, temperature sensors...).

Starting from a commercial toroidal transformer, for this project it was necessary to add 4 more windings to isolate all stages.



### LCD Boosterpack V1 (2013)

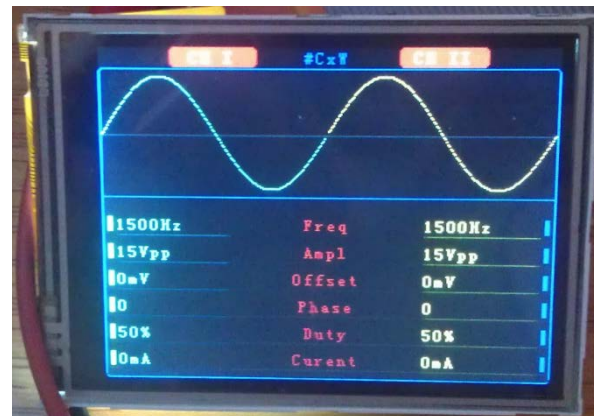
Parallel LCD (SSD1289 chip) with graphics C++ libraries developed for Stellaris/Tiva Texas Instruments microcontroller.



### Complex Waveform Generator (2013)

ComplexWaveformGenerator is a Project developed as a continuation and upgrade of a classical analog function generator, improving some features due to the use of digital logic. The main component of the project is the Xilinx Spartan 6 FPGA. This project was submitted to the 2013 Texas Instruments Analog design Contest.

- Power supply using devices chosen by TI's WEBENCH, optimizing BOM, Price and efficiency.
- Friendly user interface using a 3.2 inches touch LCD, switches, rotary encoders and RGB leds.
- Signal synthesizer of both classical signals such as sine or square waves, and arbitrary user-created signals.
- Signal post processors with digital controlled analog devices, assuring low noise levels and a high (up to 40dB) voltage gain.



### LCD Boosterpack V2 (2014)

This project is the second part of a successful project with some important improvements. The main goal of LCDBPV2 is to be an easy and cheap way to integrate a touch LCD into any project. To achieve this target, one CPLD fully programmed in VHDL and some software tips like LRE compression and variable length protocol was used (developed by myself: "4 wire EHA protocol").

LCDBPV2 has libraries for Arduino, Tiva Launchpad, CC3200, Tiva connected, STM32, NXP Xpresso, PIC (CCS) and MSP430, but the "boosterpack" word means that this project was developed for Texas Instruments platform and is totally compatible with his great Graphics Library. First version was developed using Xilinx XC9572, later Altera Max II EPM1270, and continued with Xilinx XC3S200AN.

This project, designed, assembled and manufactured by myself is sold and supported in my web: [lcdbpv2.danirebollo.es](http://lcdbpv2.danirebollo.es).



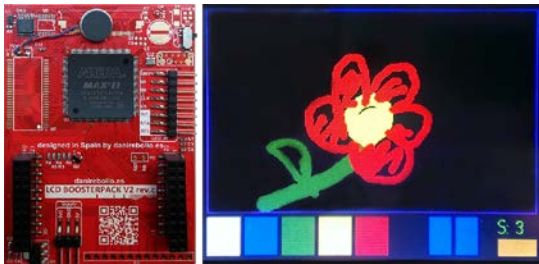
## Home Automation System (2014)



It was submitted to the 2014 Texas Instruments Analog design Contest. The Home Automation System (H.A.S.) is meant to offer a huge variety of home services to the user through a friendly and easy to use GUI. The HAS mixes a powerful and flexible hardware, to adapt to almost any need, with an optimized software, creating a fully integrated system. To create this smart home environment we've integrated almost all the TI development systems to control the sensors and actuators that provides a suitable consequence for every need.

The main goals of our project are to create a cheap and easy to use Internet of Things home. Home Automation System is formed by one Sitara ARM server (Beaglebone Black) and some nodes around it collecting data (CC3200 & MSP430).

## LCD Boosterpack V3 (2015)

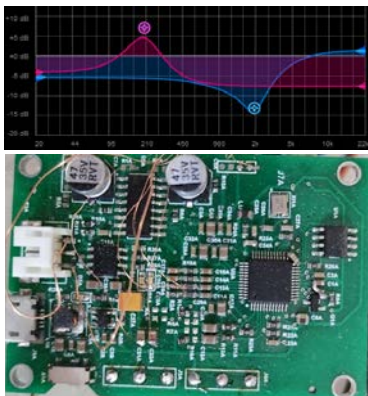


Some of the main improvements was:

- Fully featured GPU with SRAM framebuffer
- Haptics support
- Improved touch control and automatic calibration of touch data
- EHA2 and I2C control.

<https://danirebollo.es/index.php/lcdbpv2>

## Total Audio System (2015)

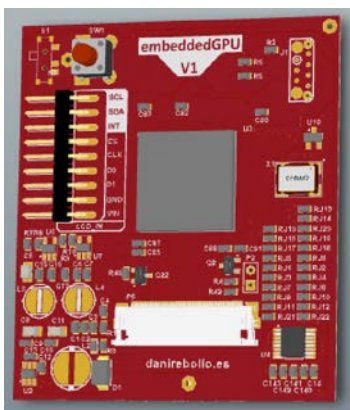


Audio interface with multiple in/out channels witch can apply effects (equalization, compression, delay...) with a javascript & html web page.

ADAU1701 based DSP board with external I2S ADC, DACS and I2S multitrack recording capability.

The system is controlled by a CC3200 WiFi MCU (ADAU1701 and DACS through I2C) with a basic configuration but other can be set with Analog Device's SigmaStudio.

## eGPU, formerly LCDBP (2016)



Latest upgrade of the LCDBP/eGPU series which is pleased to have had customers like Tesla Motors, Sandia laboratories, BQ, MIT, University of Switzerland, Berlin, Singapore, India and other persons around the world.

VHDL redesign, starring GPU, touch calibration, JPEG decoder, I2C & EHA control...

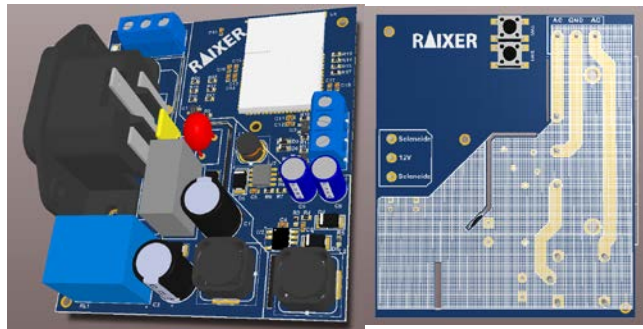
Current features:

- Touch & backlight & vibrator I2C driver (VHDL modules)
- EHA decoder for graphics (VHDL module)
- FIFO on both I2C and EHA sides
- Low BOM count (FPGA, touch ADC, boost converter for backlight, buck converter for power supply)
- HDMI output
- VHDL 32 bits soft-microprocessor based on ZPU.
- JPEG decoder



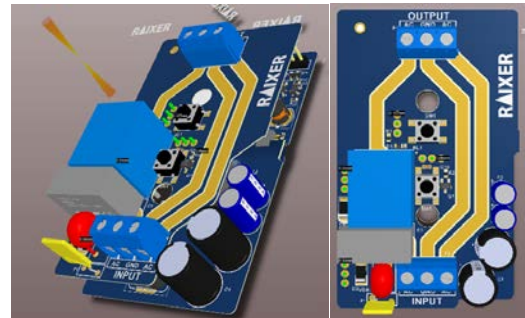
## WiFi electric plug “Raixer” (Freelance. 2016. Client: Raixer)

IoT smart plug with Particle Proton P1 boards. In this project I was the hardware engineer in charge of the sensors and power supply (switch, non isolated).



## Wifi Electric plug “Raixer Mini” (Freelance. 2016. Client: Raixer)

IoT smartplug with Particle Proton P1 boards. In this project I was the hardware engineer in charge of the sensors and power supply (switch, non isolated). Mini version.



## Preventive detection of catenary breakage (Freelance. 2016. ADIF outsource)

The goal of this project is to allow users to be able to detect catenary's train breakage before it takes place. To do this inductive sensors are used. These inductive sensors can measure the catenary's copper section in a slow trip done by an overnight test train. Several sensors in the array compensate for the error thrown by oil and other mechanical particles.

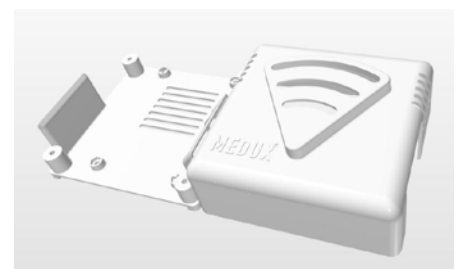
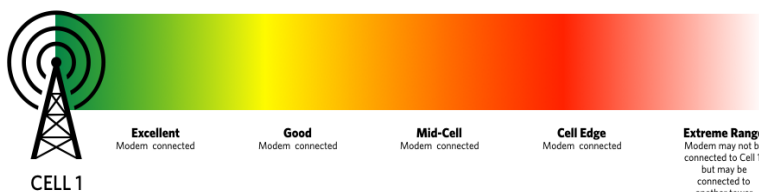
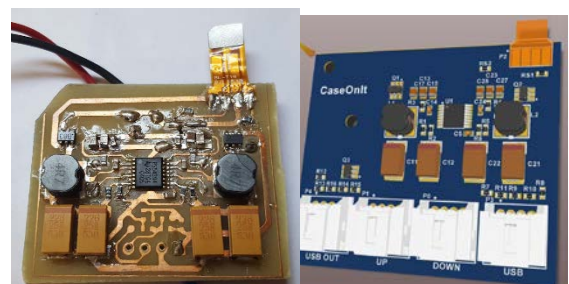


## Embedded power supply & mechanical design (Freelance. 2016. Client: Medux)

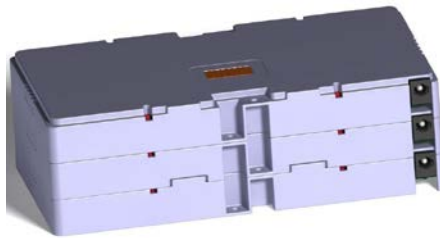
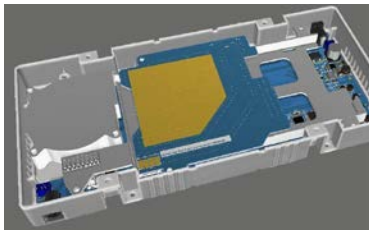
IoT board based on a Smartphone with WiFi and other sensors. Development of the power supply (switch, isolated) for GSM/CDMA 4G device with battery with battery emulator circuit.

Altium Designer – Catia – Solidworks interaction.

The second stage was to design an Uninterruptible Power Supply (UPS) for the previous board with a supercapacitor and LiPo battery.



## FPGA stackable control board (Freelance. 2017. Client: Medux)



This project includes multiple interconnected boards, a fan, a SIM card extension with FPC cable, Xilinx spartan 3 FPGA and Power supply for it and for the embedded device that needs to control.

## License plate reader (Freelance. 2016. InPark)



Through this system it is intended to reemplace the traditional keys of a shared garaje (public or neighborhood garage) with a Smartphone app or NFC tag using a M2M device based on Atheros AR9331 Wifi/Ethernet microcontroller.

Case design, Atheros AR9331 and peripherals, Uninterruptible Power Supply (UPS) and NFC transceiver.

## License plate reader & automatic barrier opening (Freelance. 2017)

Using MJPEG camera and embedded ARM system (Raspberry pi 2 industrial) with openalpr (recompiled post process and filter stage to optimize coincidences) the system reads car's license plates and uploads this data to server, creating API to interact with.

Also is done:

Frontend to see occurrences and analytics

MQTT to connect embedded devices for debug purposes

System reads license plate, compares with white list, if not compares with server.

## Electric motorbike controller (Freelance. 2017)



The main part of this project is the controller of the triphasic motor and battery. Once this part is working, user data is transmitted with Bluetooth to the mechanical workshop or insurance company so they can check metrics and do forensic analysis.



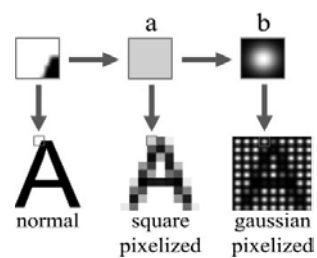


## TV QOS Real-Time analysis (using FPGA) (Freelance. 2017-2018. Client: Medux)

Definition and direction of the project in its early years. Design mechanical parts and manufacture plastic boxes.

The project analyzes the HDMI decoder's signal of platforms like Netflix or Amazon Prime in real time to detect the quality of the signal and report it to a backend.

For this, a Xilinx Zynq FPGA and an HDMI deserializer chip were used. The FPGA performed algorithms that had been previously tested and designed in Python. The system used the ARM cores built into the FPGA to provide connectivity and report metrics to the backend.



## OpenLock (danirebollo.es 2017-2018)

Electronic lock controllable by Zigbee / Bluetooth / SubGhz / GSM . This lock is adaptable to practically any lock (European, American, other non-standard models) due to its mechanical design.

If GSM or WiFi is used, the lock itself connects with the backend. In the opposite case, a gateway is required to provide connectivity or the use of a mobile application that connects by bluetooth and provides connectivity to the lock.

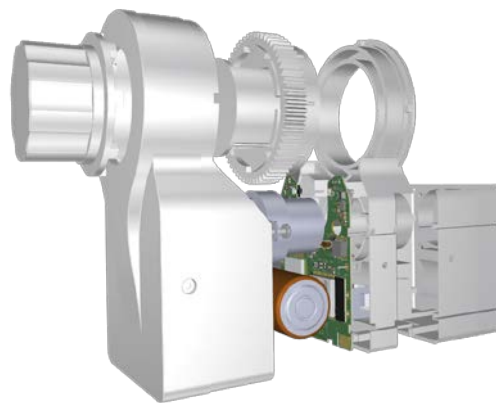
Communication with the backend is done using MQTT directly or Azure IoT hub, AWS, GCloud, which also implement part of the MQTT stack.

Battery lifetime varies between 3 months and a year using ER26500 batteries with special chemistry (GSM, WiFi). Several years using bluetooth, subGhz or Zigbee communication and gateway that provides connectivity.

Hardware designed with Altium Designer.

Mechanics designed with Solidworks.

<https://danirebollo.es/openlock/>



## Bike rental lock (Freelance. 2018-2019. Client: Movus)

Commercial bicycle lock mechanic with customized electronics.

Hardware: PCB developed using Altium Designer. Main microcontroller: ESP32. Communications and GPS positioning module: SIM868. Backup battery ER26500 + Supercap. Charging from the main bike battery.

Frontend: Visualization of geolocation, opening, alarms.

Backend: Azure IoT hub (MQTT), Functions (javascript)

Analytics: Grafana

Hardware designed with Altium Designer.

Mechanics designed with Solidworks.



## Intercom & door opener & alarm (GSM, WiFi, Bluetooth, zigbee, energy harvesting) (Freelance. 2018-2020. Client: Ringo)



The project consists of several stages:

### -Hardware:

Several microcontrollers are used in this development:

--ESP32 with modified partition map to use 16mb Flash ROM. Dual core, bluetooth, WiFi. FreeRTOS disabling automatic WatchDog control. Individual core's control. Downclock to 80MHz, heavy use of interrupts, lightsleep and deepsleep to obtain a consumption of 20mAh (ESP32 only).

--SIM800: GSM 2G connectivity. Modified firmware to use Azure IoT hub stack without host microcontroller intervention.

--CC2530: Zigbee 3.0. using ZNP

In addition, other important points are:

--Call detection and opening of analog intercoms.

--Decoding and sending of messages on IDLE high, IDLE low digital intercoms.

- GSM communication

Harvesting of available data lines (the device works without power)

Customized 8000mAh battery.

### -Firmware:

Using Platform.io. ESP32 with modified partition map to use 16mb Flash ROM. FreeRTOS disabling automatic WatchDog control. Individual core's control. Downclock to 80MHz, intensive use of interrupts, lightsleep and deepsleep to obtain a consumption of 20mAh. Communications libraries with different intercom decoders, sensors, locks. OTA updates via GSM

### -Backend:

Azure IoT hub, Azure functions (javascript).

-Frontend application:

Frontend that allows you to use the device from a web: configure and monitor it.

-Frontend analytics:

Frontend 1 extracting telemetry from mysql database and influxdb designed in PHP.

Frontend 2 using Grafana. Optimizing data ingestion from the backend.

### -Mechanics:

Box design with Solidworks. Integration with Altium Designer.

Prototypes made with a resin SLA 3D printer, optimized for mass production of ABS injection.

Prototyped and tested in my own laboratory (Madrid), mass manufacture in Shenzhen, in battery, electronics, mechanical, plastic and assembly factories



### Electronic & mechanical consultant for automotive IoT (Freelance. 2021. Client: Looper)

Review fabrication files and suggest improvements to the current design.



### PCB designer for wireless charger (Freelance. 2021. Client: La siesta technologies)

PCB design for wireless charger with already designed enclosure.



### Hardware consultant (Freelance. 2021. Client: Bestseller)

Hardware consultant. Wireless video transmission.



## Zephyr OS firmware programmer (Freelance. 2021. Client: Watchman Door)

Firmware programming for Nordic ARM microcontroller using Zephyr OS.  
Smart lock product.

```

1  //
2  // "name", "name"
3  // "name", "Microsoft Corporation"
4  //
5  import {CancellationToken} from "vs/base/common/cancellation";
6  import {CancellationError} from "vs/base/common/errors";
7  import {IHost} from "vs/base/common/host";
8  import {IHostInitiationService} from "vs/base/services";
9  //
10 //
11 // "returns the last element of an array."
12 //
13 // "Returns if which element from the end (default is 1)
14 //
15 export function lastOf<T>(array: T[], startIndex: number = 0): T {
16     return array[array.length - 1 - startIndex];
17 }
18 //
19 //
20 export function last2Of<T>(array: T[], index: T): T {
21     return array[array.length - 1 - index];
22 }
23 //
24 //
25 // "name", "name"
26 // "name", "Microsoft Corporation"
27 //
28 // "name", "name"
29 // "name", "name"
30 // "name", "name"
31 // "name", "name"
32 // "name", "name"
33 // "name", "name"
34 // "name", "name"
35 // "name", "name"
36 // "name", "name"
37 // "name", "name"
38 // "name", "name"
39 // "name", "name"
40 // "name", "name"
41 // "name", "name"
42 // "name", "name"
43 // "name", "name"
44 // "name", "name"
45 // "name", "name"
46 // "name", "name"
47 // "name", "name"
48 // "name", "name"
49 // "name", "name"
50 // "name", "name"
51 // "name", "name"
52 // "name", "name"
53 // "name", "name"
54 // "name", "name"
55 // "name", "name"
56 // "name", "name"
57 // "name", "name"
58 // "name", "name"
59 // "name", "name"
60 // "name", "name"
61 // "name", "name"
62 // "name", "name"
63 // "name", "name"
64 // "name", "name"
65 // "name", "name"
66 // "name", "name"
67 // "name", "name"
68 // "name", "name"
69 // "name", "name"
70 // "name", "name"
71 // "name", "name"
72 // "name", "name"
73 // "name", "name"
74 // "name", "name"
75 // "name", "name"
76 // "name", "name"
77 // "name", "name"
78 // "name", "name"
79 // "name", "name"
80 // "name", "name"
81 // "name", "name"
82 // "name", "name"
83 // "name", "name"
84 // "name", "name"
85 // "name", "name"
86 // "name", "name"
87 // "name", "name"
88 // "name", "name"
89 // "name", "name"
90 // "name", "name"
91 // "name", "name"
92 // "name", "name"
93 // "name", "name"
94 // "name", "name"
95 // "name", "name"
96 // "name", "name"
97 // "name", "name"
98 // "name", "name"
99 // "name", "name"
100 // "name", "name"

```

**IoT medical wearable consultant (Freelance. 2021-2022. Client: loon technologies)**

Supervise electronic designer's work. Brainstorming. Review the whole project, from idea to mechanical, server and electronic parts.

Highlights: Bluetooth low energy, battery use, biometric sensors, mechanical design.



**Employee trainer / teacher (Freelance. 2021-2022. Client: Zooo)**

Electronic trainer. From basic electronics to repair oriented. Side projects.



## IoT Electronic designer (Freelance. 2021-2022. Client: Last Basic)



Design IoT electronic product for Last Basic company. ESP32 microcontroller. I2C weight sensor, I2C multibus, WiFi control.

Redesign using hardware good design practices. Prototype.

## Electronic consultant. CC3220 firmware & project management ideas (Freelance. 2021-2022. Client: Electro Drives)



Electronic consultancy based on C3220 WiFi microcontroller using Code Composer Studio. Contribute with some project management ideas. GIT version control train.

## Arcade machine restoration (Freelance. 2022. Client: Zooo)



Upgrade & restoration of "Tarot machine" adding LCD screen, laser printer and WiFi control. Linux Raspberry Pi based.

## GSM Firmware for AWS IoT (Freelance. 2022. Client: inbiot monitoring)

```
1 import { CancellationToken } from 'vs/base/common/cancellation';
2 import { CancellationError } from 'vs/base/common/errors';
3 import { Emitter } from 'vs/base/common/sequence';
4 import { findFirstNonWhitespaceCharacter } from './arraysFind';
5
6 /**
7  * Returns the last element of an array.
8  * @param array The array.
9  * @param n which element from the end (default is zero).
10 */
11 export function tailOf(array: ArraylikeT, n: number = 0): T {
12     return array[array.length - (1 + n)];
13 }
14
15 export function tailOf(arr: T[]): T[] {
16     if (arr.length === 0) {
17         throw new Error('Invalid tail call');
18     }
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Develop ESP32 firmware connecting mcu with AWS IoT Core cloud, using SIM800 GSM modem through AT commands, and custom SIM800 firmware with MQTT stack.



## Current meter for home automation (Freelance. 2023)

ESP32 based.  
ESPhome compatible.  
Home Assistant front and backend.



## IoT devices. GPS, NBIoT. Hardware Engineer (2022-2023. Bizintek)

Design IoT devices (ESP32 with sensor, tracking, LTE, NBIoT, MQTT, GPS).  
Repair broken designs. Look for failures in hardware and software and fix them in the laboratory.

Analyze accelerometer patterns with FFT and python scripts to infer movements.

PCB design and manufacture files: Altium Designer.

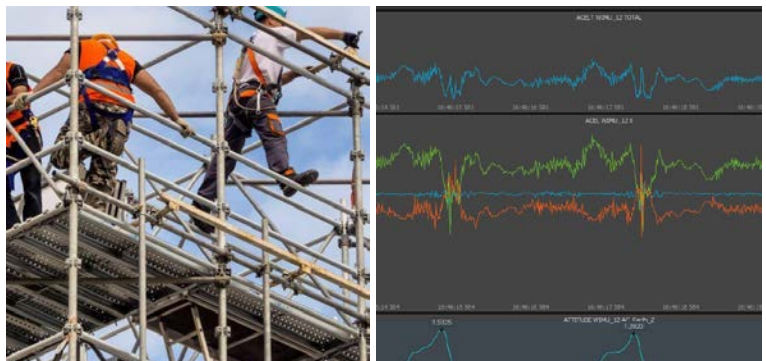
Firmware programming using Platformio and GIT in VSCode. Project validation.



## Solar panel installation tool project



## Worker electronic fall detection project





## ROS2 Robot. R&D & Hardware Engineer (Freelance. 2023. Client: Ikea)

Develop a robot capable of reading RFID tags, locate into a map and send the processed map to the backend.

To do this the robot is using a Raspberry Pi 4 with ROS2, RFID reader, LIDAR, IMU and Create2 chassis. All the projects are using docker.

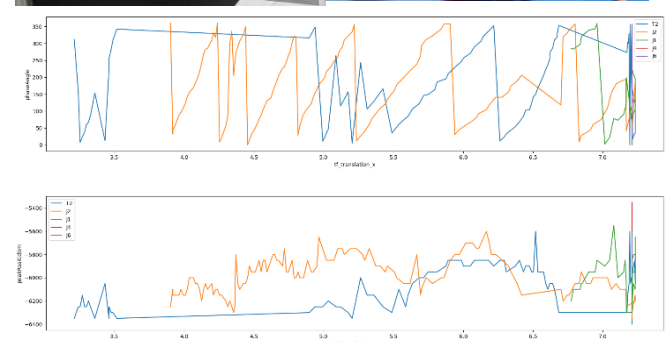
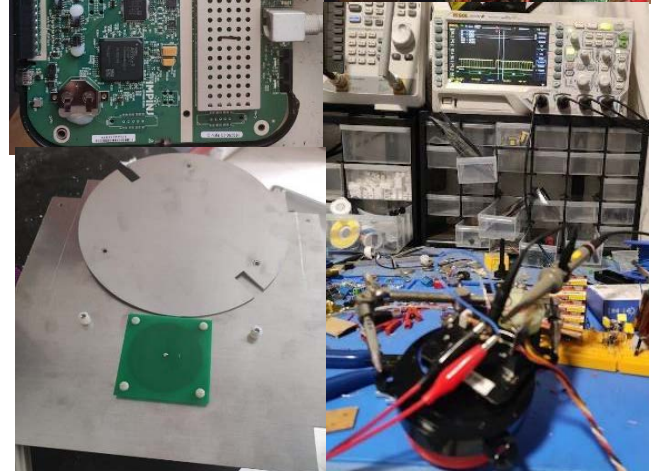
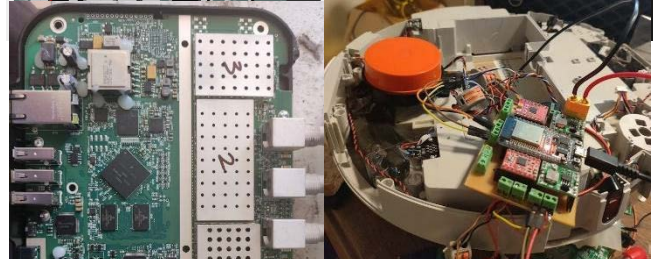
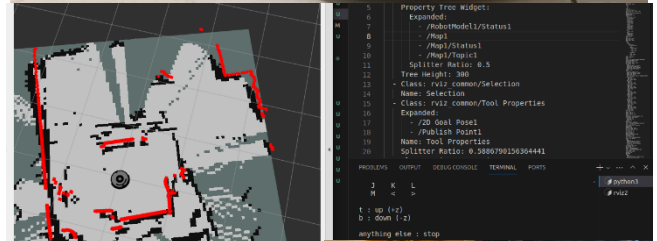
To achieve this goal using docker and specific parts there is an extense hardware and software modification in this project.

### Tech & hardware stack:

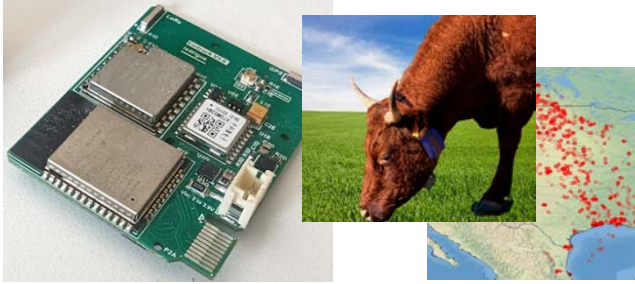
- Linux, bash, python
- Docker, portainer
- MQTT
- MySQL / MariaDB
- ROS2, SLAM, odometry, ROS2 MQTT gateway
- BLE PS4 gamepad control
- ESP32
- Sensors: LIDAR, IMU, motor wheel encoders, RFID tags & reader (IMPINJ & ali...)
- Power management: gas gauge, buck supply, boost supply, 100Ah @ 40V li-ion battery & charger, POE+
- Wood work, foam, spray paint
- RFID tag vectors trilateration algorithm

### Tests

- RFID tags benchmark
- RFID readers benchmark
- RFID antennas benchmark
- RFID trilateration tests, RSSI / phase benchmark
- Mechanical design: solidworks. Parts: Wood, screws, foam, 3d print
- Motor damper springs tests to re-tune the damper effect given the increased weight. Increase downforce in 2 wheel drive chassis.
- Motor tests: 2WD vs 4WD vs mecanum
- Image processing: Real-Time RLE compression and transmission through MQTT.
- Polar transform of images with given angle



## GPS & LoRa Cow tracker. R&D & Hardware Engineer (Freelance. 2024)



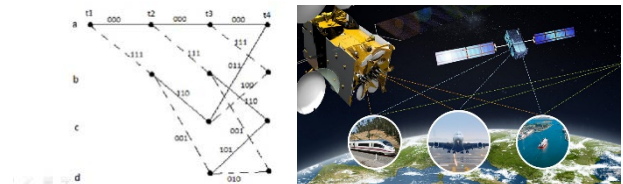
Develop a LoRa tracker with GPS and ultra low power consumption of 4uA sleeping. Based on ESP32 and ESP32 ULP. 2 years battery life.

Also develop a LoRa gateway with MQTT connection to the backend.

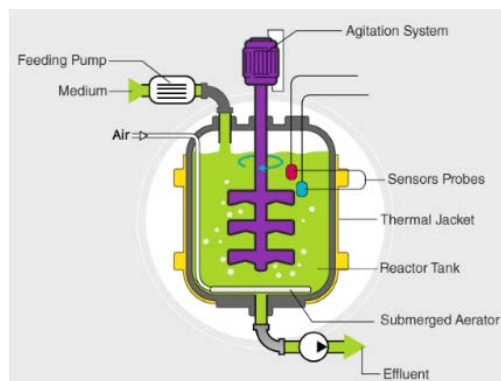
PCBA Fabrication, molds, enclosure IP65.

## FPGA VHDL module (Indra - Freelance. 2024)

Design a VHDL convolutional encoder module for an improvement of space EGNOS Augmentation system project of the European Space Agency.



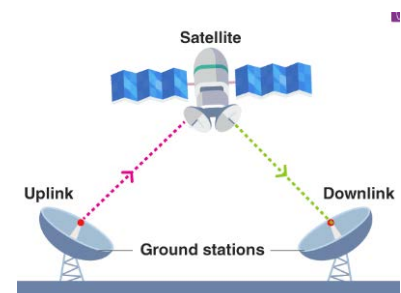
## IoT controlled fermenter for obtaining proteins (Noxon / Moa biotech - Freelance. 2024)



Design an IoT product to control and sensor all the required parts and also make a web frontend to expose an easy-to-use user interface.

## Testing & design proof of concept (Hydra Space - Freelance. 2024)

Prototype (hardware & firmware) and analysis to validate data fallback between NB-IoT and Satellital communications.



## Hangar for autonomous drones (Cepssa – Moebe - Freelance. 2024)

Design of the infrastructure necessary to allow autonomous drones to carry out inspections in refineries.



For this, we integrate different sensors:

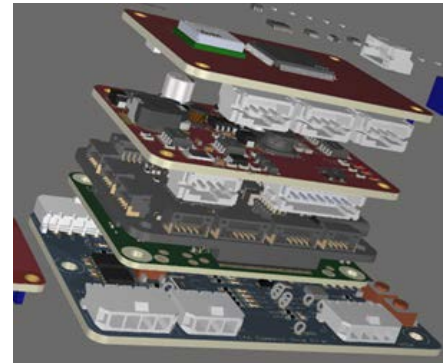
- MQTT
- Internet fallback
- ADS-B to detect nearby aircraft
- Presence and people sensors
- Meteorological sensors
- Integration of mechanical elements (gates, elevators)
- UPS & charge

## Control system for weapon sight (DUMA engineering - Freelance. 2024)

Design of different PCBs in a mechanical system, integrating different sensors such as accelerometer, gyroscope, laser meter, GPS...

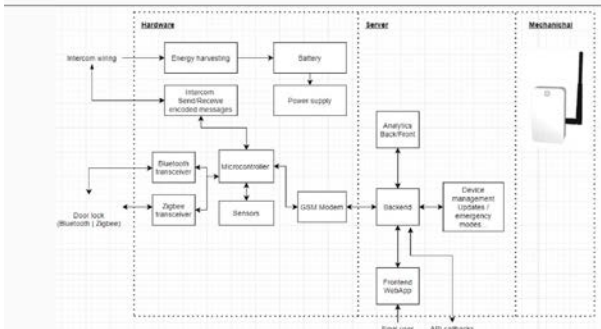
In addition to battery charger and other protections.

Design & prototype

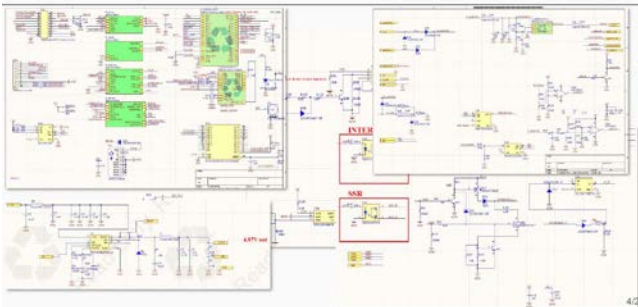




## 1 · Project scope



## 2.2 · Embedded device: Schematics



## 2.3 · Embedded device: Firmware diagram & terminal (C++)



## 5 · Mechanical design

SLA 3D printed enclosures  
Design optimized for mould



## 4.2 · Analytics: final version: Grafana



## 7.2 · Frontend: Install app for non technical people. Web App

This image shows the frontend of the web app, designed for non-technical users. It includes a "Pruebas gateway e instalación" section with steps for selecting the type of telephone, pinning and cloning the gateway, and installing the model. The interface is user-friendly and includes clear instructions and buttons for navigation.

## 9 · Design custom factory tools

Electronics factory needs to assembly the whole parts and also program it. To program device in a quick way without increase cost of the product a card-Edge connector is used. Also a software "modem flash mode" was made to allow modem flash from main microcontroller instead using secondary UART port.



## 8.1 · Documentation: Wiki

