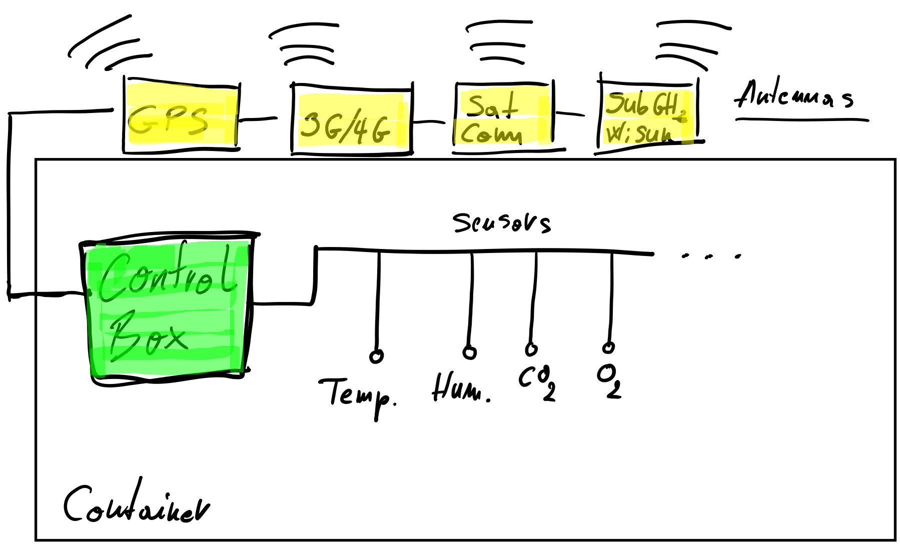
Container Monitor & Tracking Project

Introduction

This Document is a resume of what has been discussed till now. Here we are trying to describe every possible problem that exist already or can arise in the future and the respective solutions. The purpose behind is to have a common understanding of where the project is going more than have a final solution to the prototype. The main sections are *The Hardware* where it is discussed how the physical solution will be implemented and *The Firmware* where it is discussed the main block and logic of how the information will be transmitted and saved on the board itself. The last section is *The Plan* where we are trying to divide and estimate how much time this project will take.

The Hardware

The overview of how the product will be installed in the container. In the figure below is explained how the container will be equipped with the device from a schematic point of view. The antennas shall be positioned outside the container, optimally on top of it. Inside the container will be the main control box equipped with all the electronics. By different cables the sensors are going to be connected with the Control box.



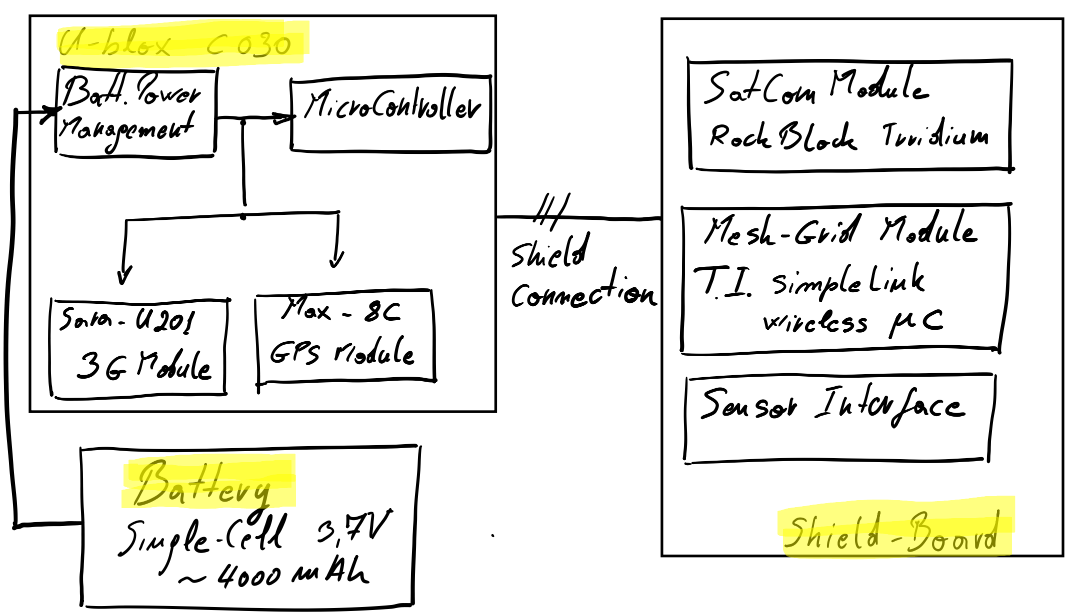
Control Box

The MVP requirements for the control box are:

- Minimal battery autonomy of 30 days.
- Maximum rate of transmission in 3G/4G mode of Once-per-Hour (min once every 6 hours)
- 3G transmission
- Satcom Transmission
- Sub-Ghz Mesh-grid communication via 6loWPAN network module
- GPS position
- Different sensors (Temperature, Humidity, CO_2, O_2, etc.)
- IP67 box

U-blox development board can fulfill only a part of the MVP requirements needed. One possible solution is the creation of a custom shield board (known as daughter board) which can be stacked on top of the C030. This solution is not the most elegant one, but should be the fastest way to have a finished prototype. In the figure below you can find a schematic explanation of the proposed solution. The shield will have 3 main blocks:

1. Satellite module for transmission of data directly to satellite. RockBlock Module based on Irridium is the most common module with full global coverage.
2. Texas Instruments has vast family of wireless microcontrollers specifically created for wireless communication, especially in the Sub-GHz bands using the 802.15.4/802.15.4g protocol.
3. Sensor Interface is the electronics that that translates the data from the sensors to the microcontroller and protects the last one from possible environment hazards (shortcut-protection, ESD protections, etc.).



To Be Defined

- Recharging Procedure.

  The battery is going to reside inside the plastic box which will be connected directly to the U-blox C030 board. Getting the battery out and replacing it would be the simplest solution but not the optimal one. Other possible options are: a) Recharging the battery via a small solar panel which arises other problems and complexities, but on the other hand, can be a definite solution. b) Exchanging the control box completely with another one fully charged. This solution is simpler from the hardware point of view, but it creates more routine management operations especially in coupling the Box ID with the Container ID from a worker in the field. c) Another possible solution is using standard size batteries (18650 is the most common). These batteries can be removed easily directly in the field.

- Sub-GHz Protocol.

  There are different protocols that have as their basis frequencies below 1 GHz and a structured layer with the purpose of having an IPv6 connection. Unfortunately till now I have not been able to find a transmission module with the Wi-Sun protocol embedded. Still there are different options on the table: a) wireless microcontroller family of Texas Instruments (CC13xx) have already an SDK with a full implementation of the Wi-Sun protocol, but without a defined protocol between the main microcontroller and the CC13xx. This adds a lot of work and is hard to say how much it is going to delay us. b) Using a module available in the market based in another type of protocol in the Sub-GHz band. Options may be Zigbee or Thread.

- The Interface

  One the problems that has not been discussed is whether the board shall have some kind of interface for parameter communication or other needs directly in the field. A possible option is having Bluetooth on board which can add to the complexity not only from the hardware and firmware side. An application should be created for communication directly to a smart phone which is not our field work area. Another group should work on this too.

The Firmware

In this section we are only showing what we think the basic structure of the Firmware should be. There are a lot of teams working together, so there should be a more detailed discussion and feedback on how this work will be divided and what has been done till now. If there has been already created a structure than parts of this section do not apply.

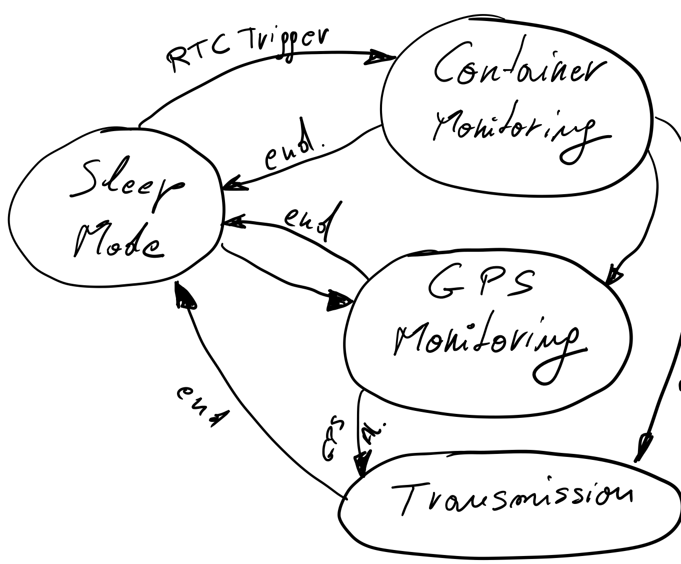
In our opinion the Firmware should have 4 different main states at the state machine:

1. Sleep Mode. Everything is off except the RTC which with a modifiable period it wakes up the board for the main purpose of measuring the sensors. Theoretically by putting everything in shutdown we should achieve current drainage that is in the range of leakage (in the order of microAmps). What we have achieved till now is only little below 1 mA. This kind of drainage tells us that only because of the sleep mode the board is going to need around 700 mAh of

battery capacity per month. Without calculating the energy needed when the board is actually running.

2. Container Monitoring. This should be a temporary state (only a couple of seconds max) where the board gets the information from the different sensors. This state should have the fastest period of sampling (a first thought ~5 min). This state will have a subset of states that depending on what the actual readings of the sensors are, it can trigger the state 4 for the transmission of the data. i.e. transmission of high temperature alarms, etc.
3. GPS Monitoring. This is the second temporary state with a slower period (a first thought ~ 20 min). In this state the board will try to get the coordinates from the GPS and log them on SD Card. This state will have its own sub states where depending on the actual data can trigger the transmission. For example, setting a geofencing which will transmit alarms depending on the position.
4. Transmission state. This is the state with the slowest period and the highest energy consumption. The battery levels will highly depend on this state and on how able are we to transmit as efficiently as possible. As talked the period of transmission can be from 1-6 hours. The sub state machine of this state is going to be the most complex where depending on what's available the board will decide to transmit via 3G Satellite or maybe to another board in its proximity that has one of these connections available.

In the figure below is shown only the big picture of these state machine and how it can work. Of course, there are a lot of details and edge cases depending on what the final requirements are.



Transmission state

This is the most complex state with 3 different types of communication:

1. Satellite communication
2. 3G cellular communication
3. Sub-GHz 6loWPAN mesh grid communication

In our opinion the transmission state should start by trying to transmit via the 3G module, if not possible then it should try the satellite module and only after the Sub-GHz wireless grid. This sentence is really simply put because there are a lot of edge cases and opportunities to optimize. For example, there is no need to try out the satellite communication if the GPS antenna could not get a connection. Meaning that a free view of the sky is not possible.

The work for the transmission state can be divided in 3 big parts:

1. Communication protocol for 3G cellular. To be decided what the JSON protocol is going to be for the transmission of sensor values, position values, battery levels, alarm states, etc. On top of it there has to be decided with what protocol is the JSON data going to be transmitted. Our proposition, since we have already worked with and we know that is relatively efficient, is: JSON over MQTT protocol to a broker using a secure layer such as TLS.
2. For the satellite communication, transmitting the data via JSON maybe is not the most efficient way since the prices per byte are much higher. JSON can be a starting ground but maybe as final result transmission directly in binary will be the solution. The same thing is worth doing with the secure protocol layers since they have a lot of headers and footers which can be optimized.
3. In the wireless grid most of the work will be focused on the application layer, since the protocol layers are already defined in the different standards. Choosing which device is going to transmit, and sending that data to all the other devices around by monitoring the battery is where the focus should be.

The Plan

In our view the work can be divided firstly in three big parts:

1. Pre-prototype
2. Full featured prototype
3. Production prototype

Pre-Prototype

In the pre-prototype only a subsection of the features will be available. All the features that are related with the custom-made shield will not be available. This is more related to the feasibility of the project and understanding that the project resided in solid ground. The features, from the hardware point of view, available will be:

- GPS positioning
- 3G communication
- Battery management
- Onboard Temperature monitoring

From the Firmware point of view:

- 3G communication protocol with the server
- Battery management and sleep-wake periods parameters that can be decided from the server side.
- Gathering of the GPS data for transmission

- Gathering of battery and onboard temperature values for transmission
- Data Logging via the SD Card

We estimate that the pre-prototype section can take around 300 hours.

Full Featured Prototype

By creating the custom made shield we will be able to have a prototype with all the features needed as explained in the control box section on the Hardware. The features added from the Hardware point of view are:

- External Sensors connected to the shield board via a interface still to be decided
- Satellite Communication residing in the shield board and external antenna
- Wireless Sub-GHz transceiver via a wireless microcontroller
- Creating the schematic and layout for the PCB of the shield board
- Producing the shield prototype board

These new hardware peripherals are going to add these sections from the firmware side:

- Satellite Communication protocol
- Creation of protocol between main microcontroller and wireless microcontroller
- Firmware of the wireless microcontroller
- Application layer for the data routing (between containers) in the mesh-grid network
- Sensor data gathering and saving

We estimate that the Full Featured Prototype will take about 700 hours.

Production Prototype

After a fully featured prototype has been tested, it will be much clear what features are a must and what are superfluous and don't add value to a final product. This is the right moment to start thinking about how the final product will be produced and what costs will it have. From the Hardware point of view it will be needed to fulfill this point:

- A new PCB layout design with all the modules needed from the u-blox module and the custom shield.
- Assembly of the first prototypes.
- Re-Evaluation of the BOM
- Enclosure optimization

The Firmware point of view:

- Firmware Robustness (watch-dog timers, debugging and optimizations)
- Driver updates as a consequence of possible hardware changes from the shield prototype
- Firmware Update over the Air (FOTA)

We estimate that the production prototype may take around 400 hours