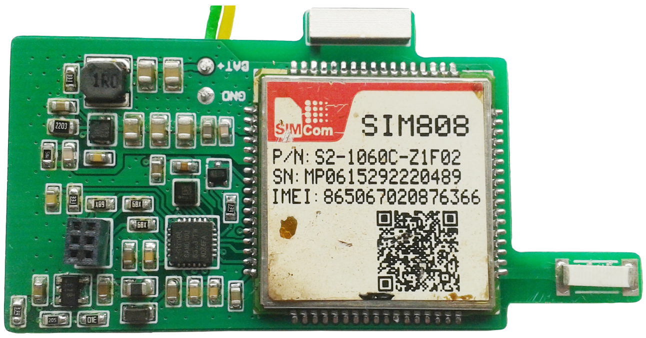
EITF40 Digital and Analogue Projects - GNSS Tracker 2.4

Magnus Wasting

26 February 2018



Abstract

In this report a mobile global navigation satellite system with SMS and alarm functionality is constructed. The design was constructed on a 32x62mm printed circuit board and was built into a bicycle light as an "anti-bicycle-theft". With a 1100mAh lithium polymer battery and a sleep interval of 30 minuets the battery life can reach 60 days.

Contents

1 Introduction

This project is done under the EITF40 Digital and Analogue Project course at Lund technical university. The project consist of the latest version of the GNSS Tracker which is built upon a previous version of the same project idea. The idea is to build a small and mobile GNSS tracker which you can communicate to by sending and receiving SMS. There are many applications were a mobile global navigation satellite system (GNSS) can be of service, some of these applications can be in the form of a anti-theft protection by finding your belongings. In this case the GNSS Tracker will be implemented as a ”anti-bike-theft” by building it inside of a bicycle light. The tracker was manufactured on a printed circuit board (PCB).

2 Block Diagram

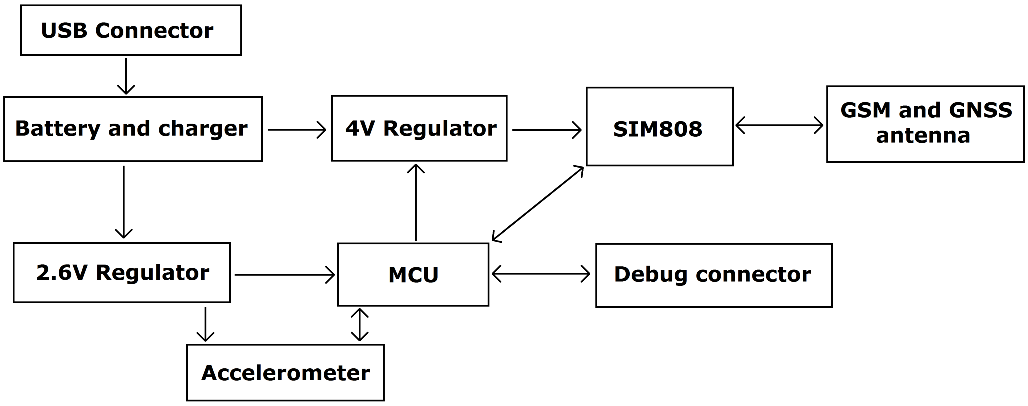


Figure 1: A simple block diagram representation of the system

3 Method of operating

The microcontroller (MCU) spends most of the time in a sleep state while the rest of the system are completely shutdown. This is to minimize energy consumption, thus maximizing the battery life. The MCU has an integrated real time counter (RTC) which counts up in seconds until it reaches a value set by the user, this triggers an interrupt which wakes the MCU from the sleep state to fully running. The MCU starts by configuring system clocks, peripheral settings and reading data from the integrated electrically erasable programmable read-only memory (EEPROM). The EEPROM contains user, debug and error information. After this, the MCU reads the battery voltage with the integrated analog to digital converter (ADC), if the voltage is high enough (3V) the MCU starts the 4V regulator by setting it's enable pin high. To ensure a stable 4V rail the MCU waits 100ms. When this is done, the MCU toggles SIM808's PWRKEY for 1s which enables the SIM808. The MCU waits until SIM808's STATUS pin is high, this indicates that SIM808 is fully running and it's serial port is active. The rest of the communication between SIM808 and the MCU is now done by sending and receiving AT-Commands. Several AT-commands are then performed to set baud rate, call settings and SMS text format. At last the MCU waits until SIM808 has establish an connection to the mobile base station.

When the setup is finally done the MCU waits (max 30s) for a ”new SMS to read at index x” command from SIM808. If there's a new SMS then the MCU asks for the content of SMS x and saves the phone number of the sender. To be sure that the SMS contains a valid command from the user a start and end flag was introduced, if one of these is not found the MCU sends a ”delete SMS at index x” command to SIM808. If it's a valid command the MCU starts executing the command. The MCU can handle a maximum of 30 commands at the same time by the use of a first in first out (FIFO) queue.

An alarm feature can be enabled with a SMS command from the user. By doing this the MCU configures another interrupt that can wake the MCU from the sleep state. The interrupt is created by the accelerometer which triggers on sudden movements.

Another feature is that the tracker will warn the user if the battery voltage is to low and needs to be recharge, this will only execute if the owner has been set.

4 User Interface

All commands must follow the <start flag><command><variable if needed><end flag> pattern, in this case the start flag is <-> and the end flag is <;>

- **–hi;** Used to see if the tracker is alive, basic parameters status will also be given as battery voltage, sleep time and how much money that is left.
- **–gps;** Used to get the location of the tracker, the location will come in the form of a Google maps link.
- **–alarm;** Used to activate the alarm function, no reply will be given. This command will only work if the tracker has an owner.
- **–owner;** Used to set the owner of the tracker, the tracker will save the phone number of the sender.
- **–sleep<variable>;** Used to set the sleep time in minutes, no reply will be given.
- **–money<variable>;** Used to set how much money is left on the prepay mobile simcard, no answer will be given.
- **–cost<variable>;** Used to set the cost of sending one SMS, no reply will be given.
- **–help;** Used to get a list of all possible commands.
- **–debug;** Used only for debug purpose, will reply with all parameters contained in the EEPROM.

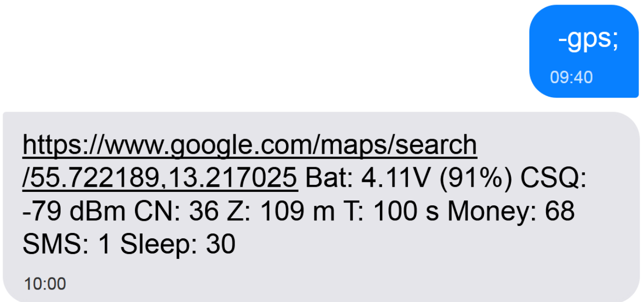


Figure 2: Sending the -gps; command and the corresponding response from the GNSS Tracker

The response from a -gps; command (figure 2) includes the location within a Google maps link, this is followed by the battery voltage, signal strength to the base station, signal quality for the GNSS receiver, meters over sea level, time it took to get a location, how much money is left, how many SMS that's been sent and sleep time.

5 Hardware And Schematics

As the project is battery driven, low energy consumption for all active components are needed. Board space is scarce which highlights the need for small components.

5.1 MCU

The microcontroller is implemented with a SAMD10 from Microchip. SAMD10 is a low power MCU with a 48MHz CORTEX-M0+ CPU from ARM with 16kB of flash memory and 4kB of SRAM. Other necessary peripherals includes UART, SPI, ADC, EEPROM and a RTC timer. All this within a small QFN-24 package of 4x4mm. As there is no need for high speed, the main clock will run at a low 1MHz to save energy.

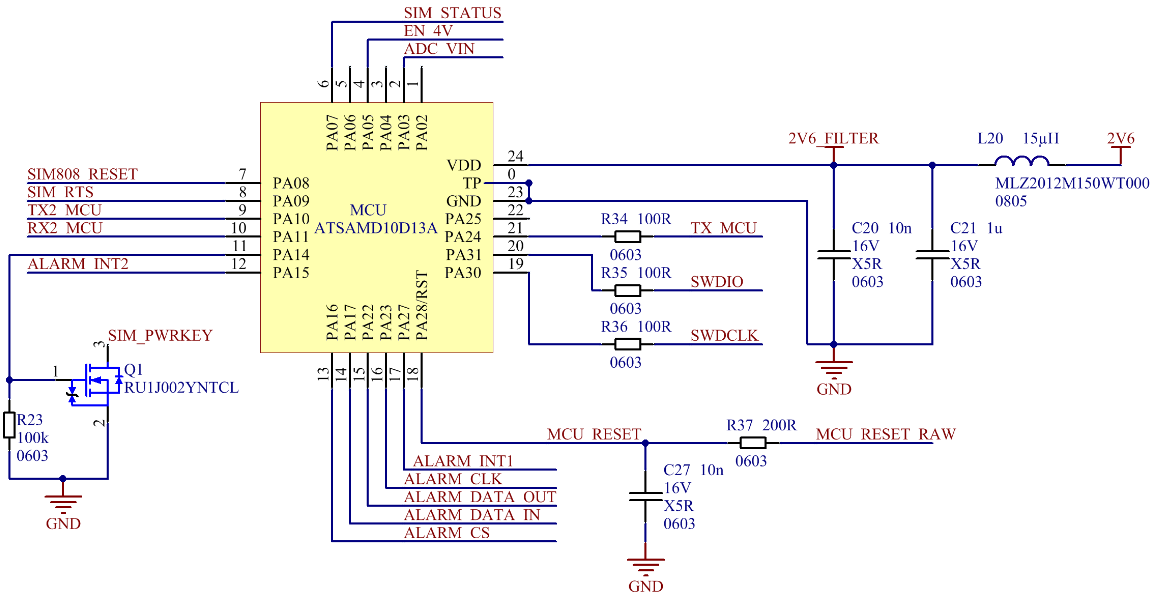


Figure 3: Schematic for the MCU

Beside the standard use of decoupling capacitors (C20 and C21 in fig. 3) the MCU has also an inductive chock, this is to minimize noise when reading the battery voltage with the integrated ADC. A simple RC filter is included on the reset pin as good practise. The transistor Q1 is needed to pull down the PWRKEY on SIM808, the PWRKEY is pulled-up internally to SIM808 power pin VBAT (max 4.2V), if one should connect PWRKEY directly to the MCU it would exceed the MCU maximum rating of 3.6V and permanently damage the chip.

5.1.1 Battery Voltage Sense

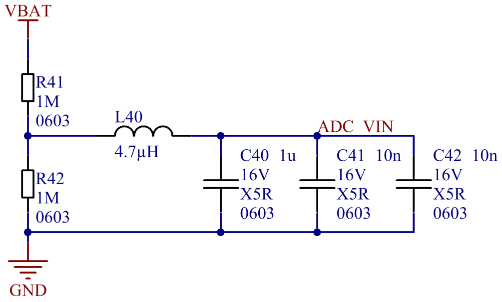


Figure 4: Schematic for the voltage division and filter

The battery voltage will vary from 4.2 to 2.8V, which is too high for the integrated ADC. R41 and R42 in figure 4 divides the voltage by 2 to a span of 2.1 to 1.4V. This is followed by a LC filter to filter noise and keep a stable DC level.

5.2 SIM808

SIM808 is an easy solution to get GSM and GNSS technologies without worrying about creating your own RF solution, all you need to add is the two antennas and a stable 4V 2A rail. The module is from SIMCOM and is a complete Quad-Band GSM/GPRS module which combines GNSS technology for satellite navigation. Communication is done with UART by sending and receiving AT-Commands. SIM808 comes in a non-standard package, size is 24x24mm.

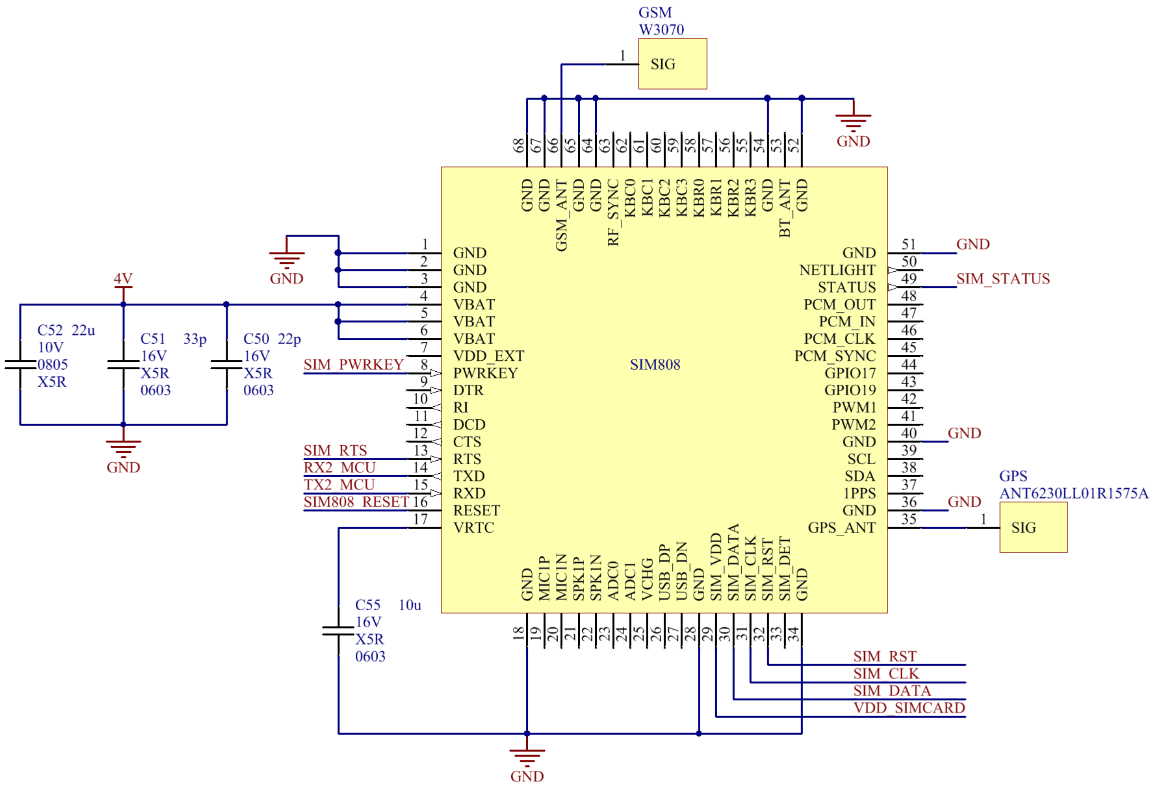


Figure 5: Schematic for SIM808 and the two antennas

As seen in figure 5 only 4 bypass capacitors and the two antennas are needed.

5.2.1 GSM and GNSS Antenna

Both antennas needs to be small as the GNSS Tracker will be built into a small enclosure. Due to this the antennas needs to be of ceramic type. For GSM the W3070 from Pulse was chosen, W3070 is a ceramic dual band monopole antenna specifically made for the 900 and 1800MHz band. For the GNSS antenna ANT6230LL01R1575A from YAGEO was chosen, what type of antenna is not specified by the manufacture, in any case the antenna is specifically made for the 1.575GHz band. Both antennas are omnidirectional which suites the project well.

5.2.2 Simcard

A micro simcard connector from molex (part number 786463001) was used. Beside the standard bypass capacitor ESD protection was also implemented. ESDA6V1BC6 is made by STMicroelectronics and comes in a SOT23-6 package.

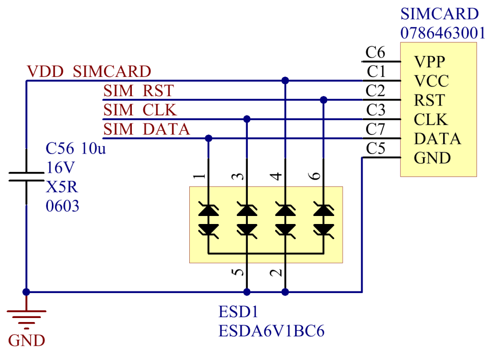


Figure 6: Schematic for the micro simcard connector

5.3 Accelerometer

The accelerometer is implemented with a LIS2DH from STMicroelectronics. LIS2DH is a ultra-low power MEMS (micro electro mechanical system) accelerometer that only draws 2uA while being active, which is suitable for the project as it will always be on. Communication with the MCU is done by 4-wire SPI. LIS2DH comes in an very small package, 2x2mm LGA-14 with a pitch of only 0.35mm.

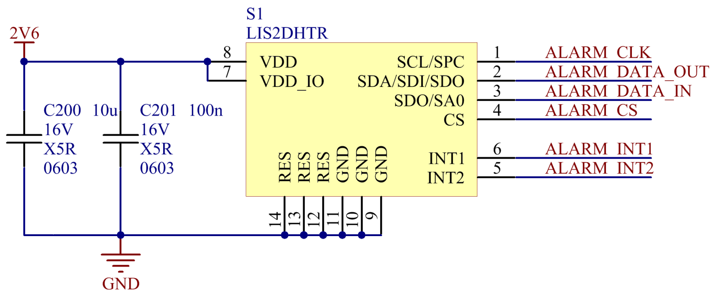


Figure 7: Schematic for the accelerometer

5.4 4V Regulator

The 4V regulator is realized with TPS61230A from Texas instrument, which is a fully integrated synchronous boost converter. A synchronous converter was chosen as it's able to give zero voltage at the output compared to a non-synchronous boost converter which outputs a minimum voltage of $V_{in} - V_{diode}$. This is in turn to minimize the power consumption when the 4V regulator is off (leakage current into SIM808). TPS61230A comes in a non-standard 2x2mm VQFN package.

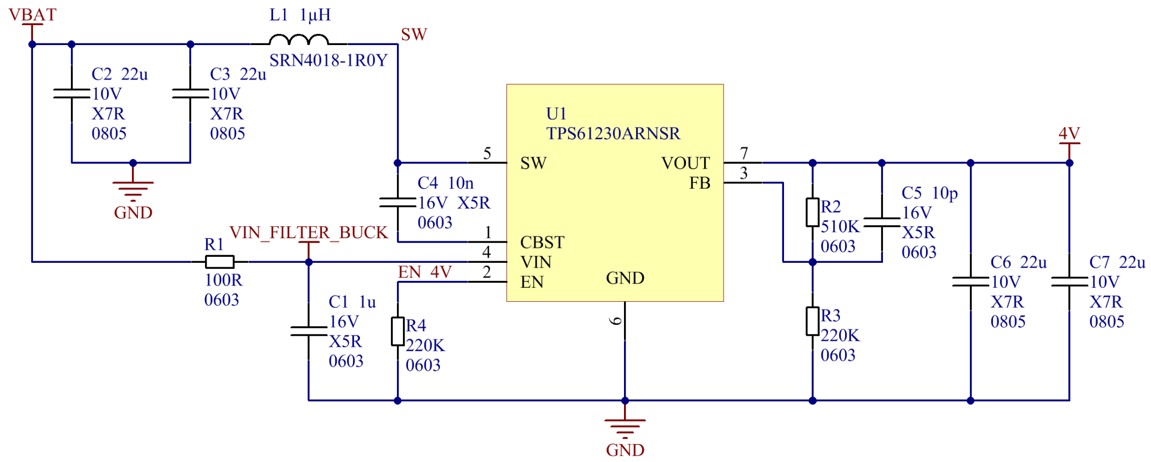


Figure 8: Schematic for the 4V regulator

TPS61230A has a internal under-voltage lockout (UVLO) at 2.5V, this caused problems in a previous version. When SIM808 is transmitting it draws 2A in short pulses, this in turn causes a voltage drop seen at VBAT, in some cases the voltage dropped below 2.5V which activates the UVLO and the regulator shuts down. A simple RC filter (R1 and C1 in figure 8) solves this problem.

5.5 2.6V Regulator

Since the 2.6V regulator is always on the supply current needs to be very low. TPS78326 is a fixed 2.6V linear regulator from Texas instrument and has a supply current of 0.5uA. TPS78326 comes in a SOT23-5 package.

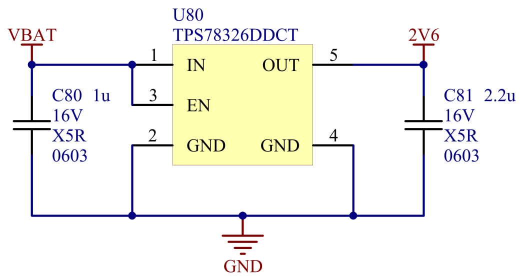


Figure 9: Schematic for the 2.6V regulator

5.6 Enclosure

The enclosure is a bicycle light, seen in figure 10, it's driven by a 1100mA lithium polymer battery. The enclosure also includes the battery charging circuit and a micro-USB connector for easy charging. This is also why the there's no charger circuit and USB connector on the main PCB as it would be redundant.

Figure 10: A bicycle light is used as the enclosure

6 PCB

The PCB has two layers and a size of 62x32mm, it was designed in Altium Designer 16. Manufacturing was made by JLCPCB.com located in China. Standard design rules was applied when designing the PCB as bypass capacitors was placed closed to their respective power pin, current loops was considered if high frequency content was available, the ground plane on the bottom layer was kept as intact as possible.

6.1 Top Layer

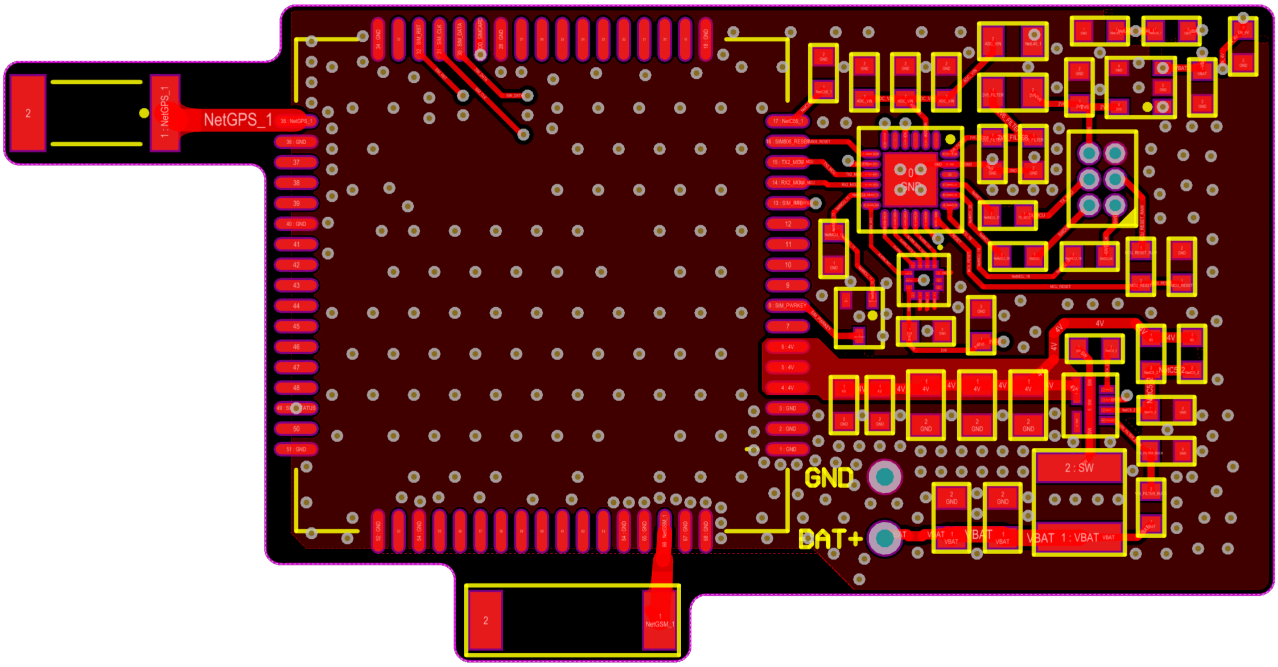


Figure 11: Top layer depicted in Altium Designer

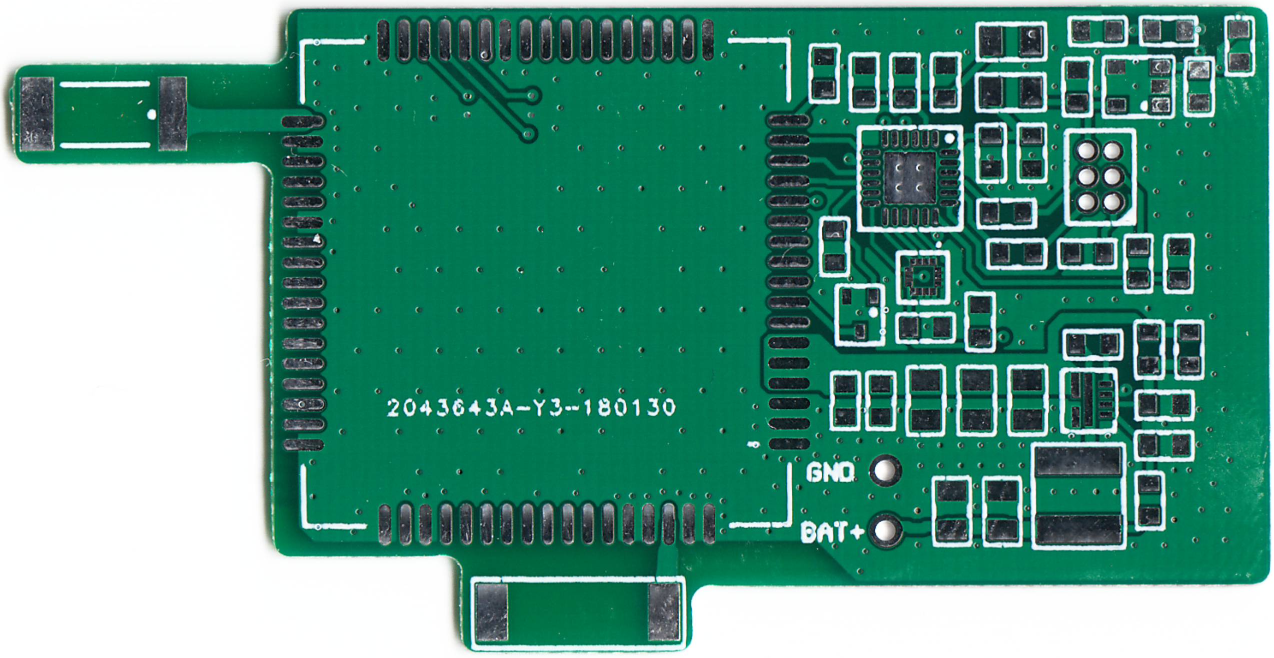


Figure 12: Manufactured PCB, top layer

6.2 Bottom Layer

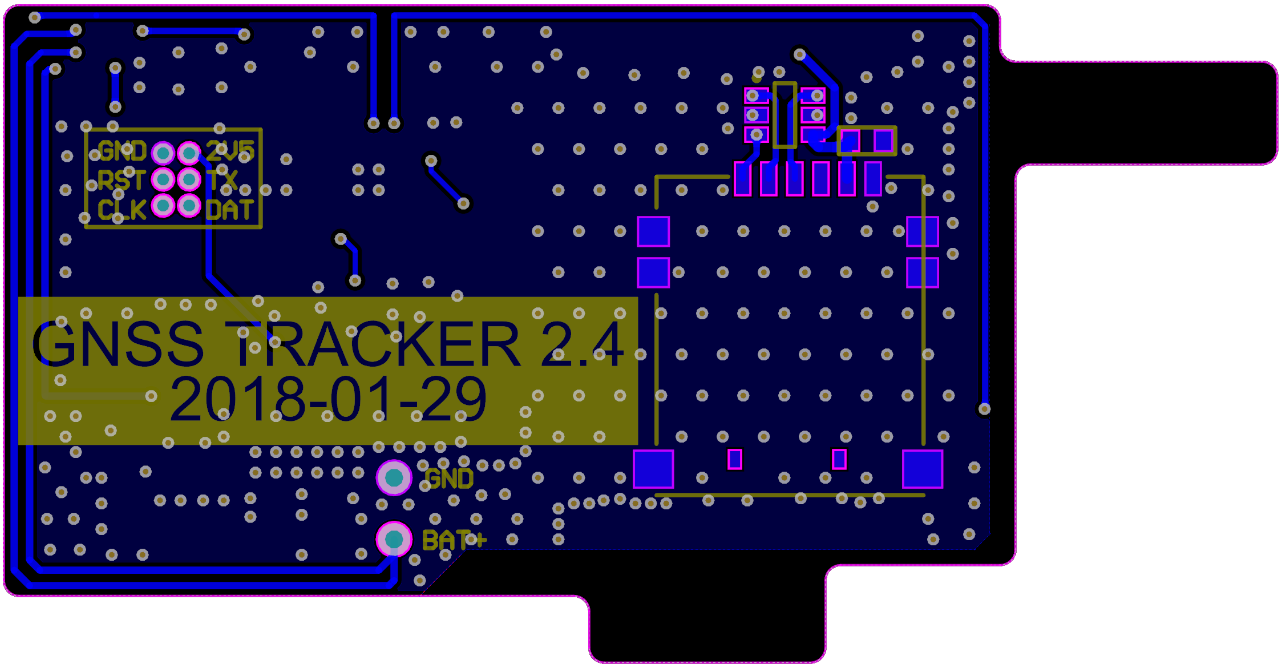


Figure 13: Bottom layer depicted in Altium Designer

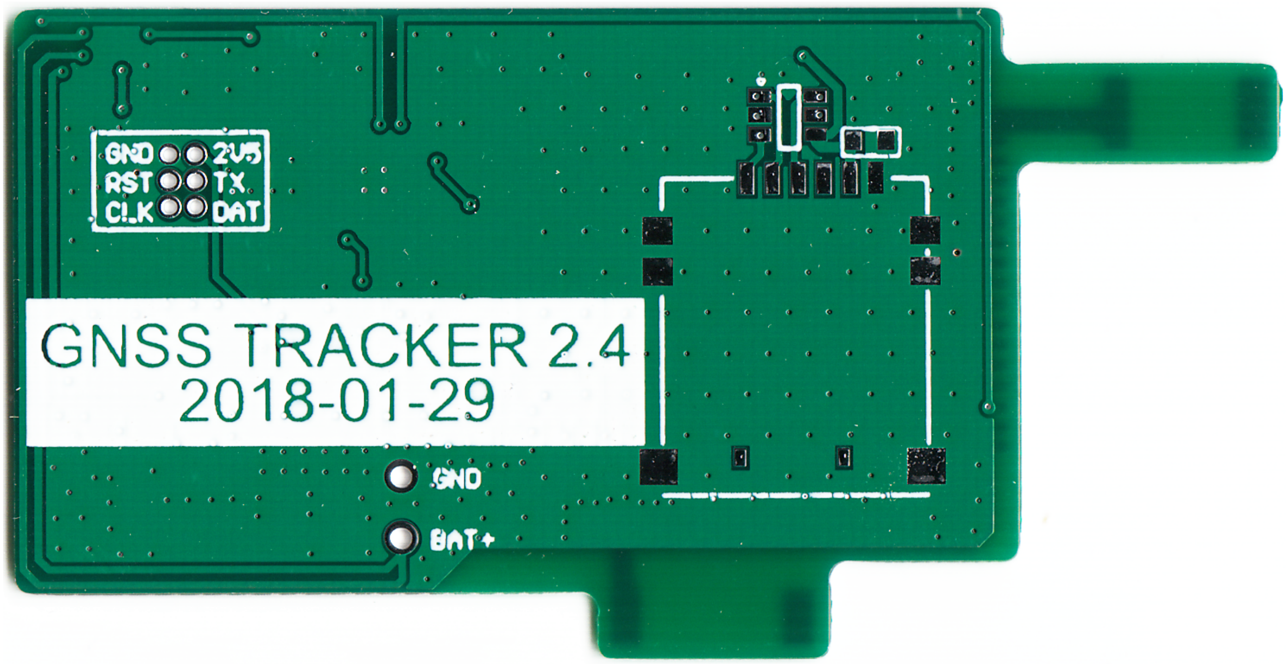


Figure 14: Manufactured PCB, bottom layer

6.3 Complete PCB With Components

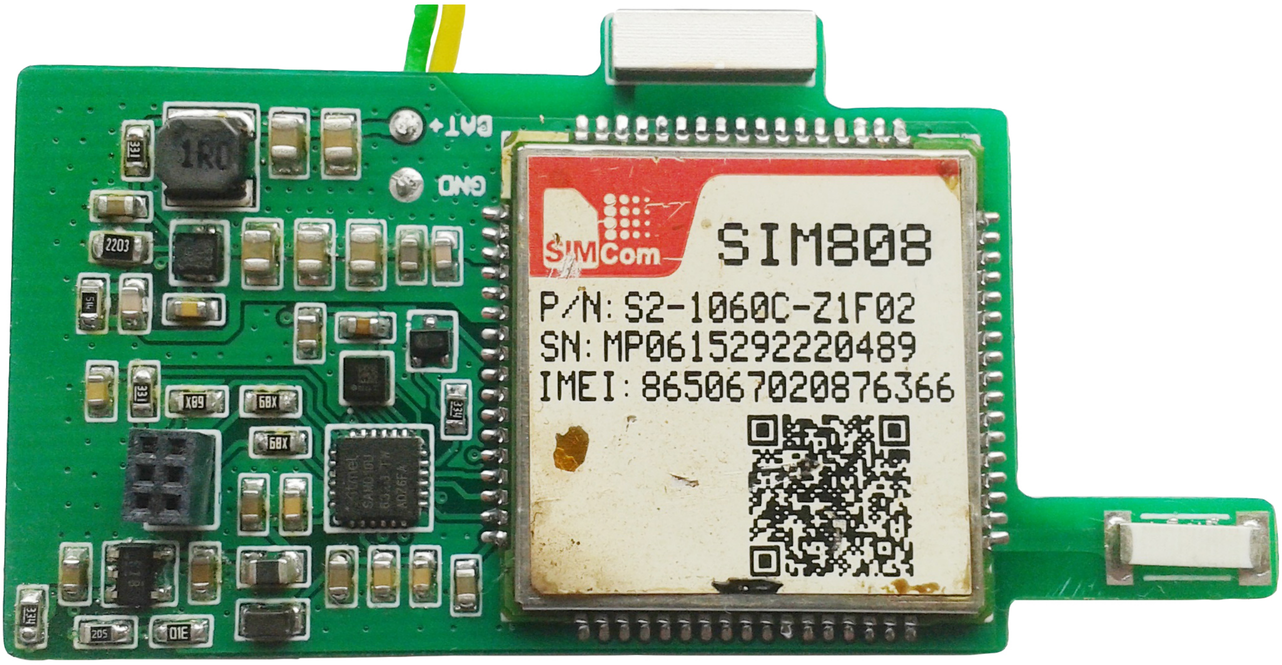


Figure 15: Complete PCB with components, top layer

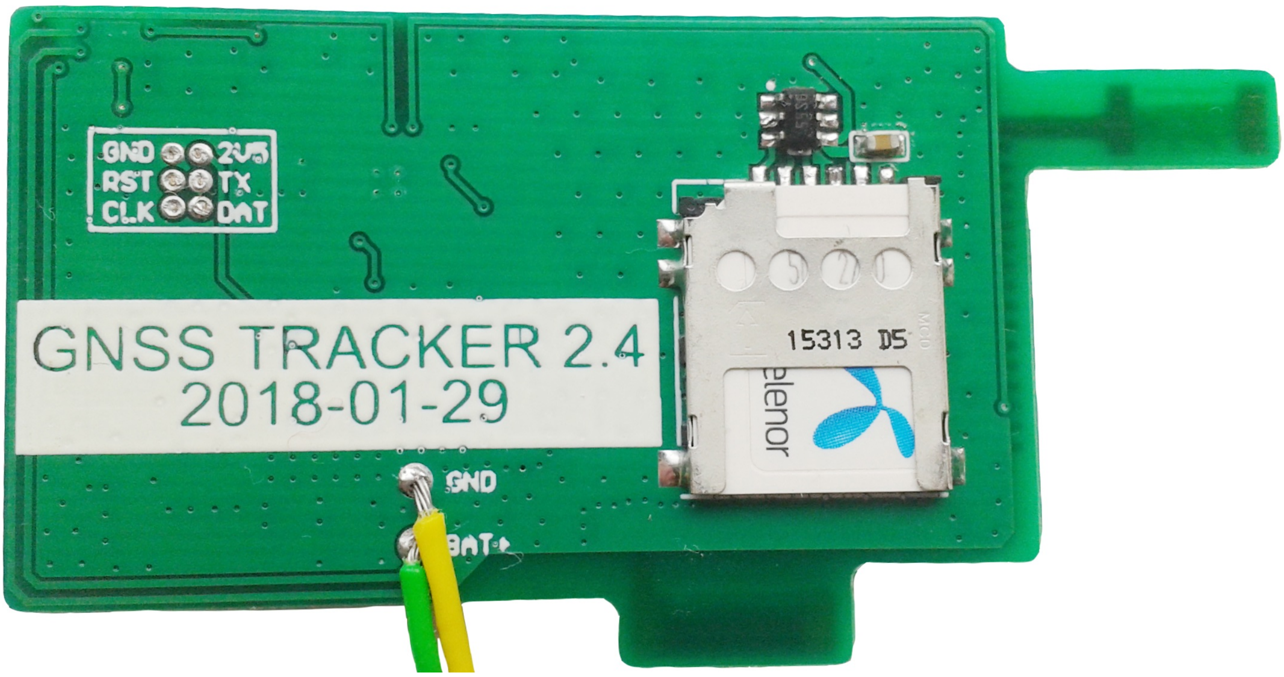


Figure 16: Complete PCB with components, bottom layer

7 Software

Software development was done in Atmel Studio 7 and programming was done with Atmel-ICE. The final size of the programming code ended at 16108 bytes which only leaves 272 bytes left. In other words, further code optimization is needed if more user commands are to be implemented. The source code can be found here.

8 Results And Measurements

8.1 Battery Life

As the current consumption varies greatly when the system are fully active, from 20mA in idle mode to pulses of 2A when transmitting. Because of this it's hard to get a estimate of the average current consumption for one cycle without using high-end measurement equipment. A simple method was used to get a rough estimate of the average current consumption, this was done by logging the number of cycles from a full to dead battery, seen in figure 17. 2878 cycles was counted, with a battery capacity of 1100mA and a typical cycle time of 42s the average current consumption is 32.8mA which is greatly higher than 4uA which is the current consumption in sleep mode.

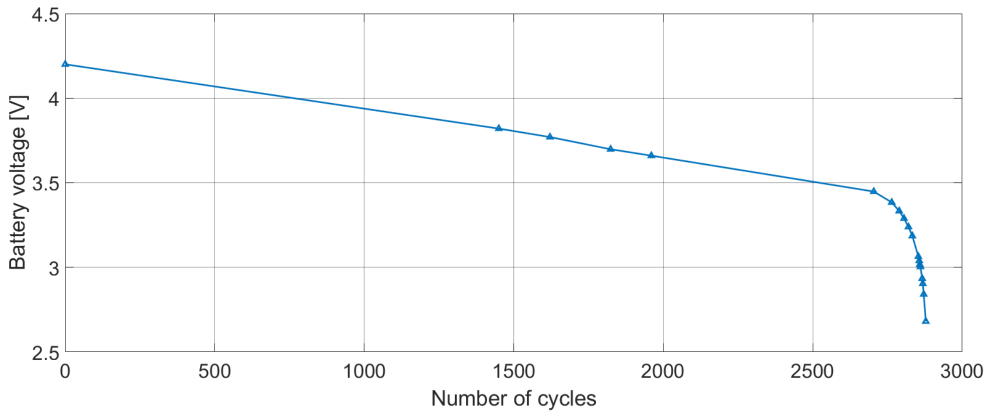


Figure 17: Complete PCB with components, bottom layer

A more important concept can be concluded from figure 17. By ignoring the sleep current completely the battery life is roughly 2878 multiplied by the sleep time variable. The default sleep time of 30 minutes gives a battery life of 60 days, this is of course the best-case scenario as zero commands was issued by the user.

8.2 RF performance

No RF measurements was performed on the two antennas, luckily the SIM808 can give some insight. Typical signal strength to the base station was -70 to -80 dBm. For the GNSS receiver the carrier to noise density ratio (C/N_0) was 30 to 40dBHz.

9 Discussion

After a lot of work particular on the software development the project ended successfully. The only drawback is the limited programmable space of 16kB, with 32kB more commands and other functions could be implemented. The battery life can easily reach more than 30 days with a sleep time of 30 minuets.

The implementation of the accelerometer went without problems, considering the 0.35mm pitch hand soldering was thought to be difficult but was successfully soldered at first try.

This project shows the importance of never giving up on your idea. As the version number tells you, multiple versions have been made. Version 2.4 is the eighth version from the first prototype. The first three versions was exclusively made for prototyping, the fourth and fifth versions had problems with an unstable 4V rail that drives the SIM808. The sixth version worked fine beside bad GNSS performance (typical $(C/N_0) < 25$) which resulted in long location time, the seventh version fixed the problem by moving the GNSS antenna 1cm from the SIM808. The eighth version includes the new alarm feature and multiple software improvements and bug fixes. A ninth version is considered, which would include switching SIM808 to SIM868 as it has 50% smaller footprint and switching the MCU to SAMD20 with 32kB of programmable memory.