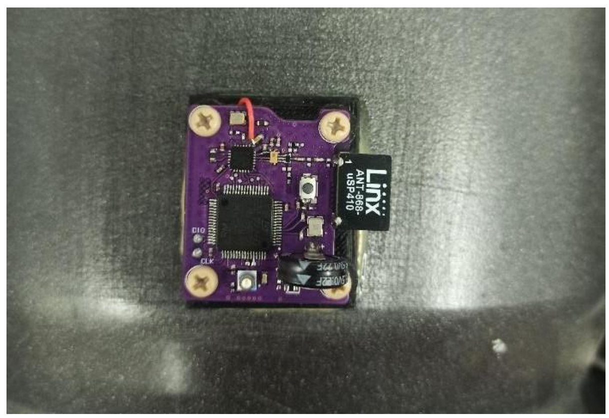
TPMS Report

Georgios Triantafyllou
Electronics Department
Prom Racing (FSAE Team)
NTU Athens

General concept / architecture

Considering the need of the team for a TPMS (Tire Pressure Monitoring System), I designed the PCBs, the code of the MCUs on the boards and a very basic mechanical mounting on the car's rims. 4 PCBs weighing 10.8g each (fully assembled) that have an incorporated pressure / temperature sensor are placed one in each rim, supplied by a coin cell battery and able to transmit wireless messages to the car's TPMS-receiver. In order to reduce the PCBs used, I have created the main telemetry system of the car that apart from transmitting messages of the status of the car to the receiving node, it also receives the packets from the 4 TPMS and the data are viewed live or logged. The TPMS can be configured remotely to send messages every 0.5s, 1s and 3s and have a minimum battery life of 1 year before cell replacement. TPMS rf frequency and transmission power can also be configured remotely.

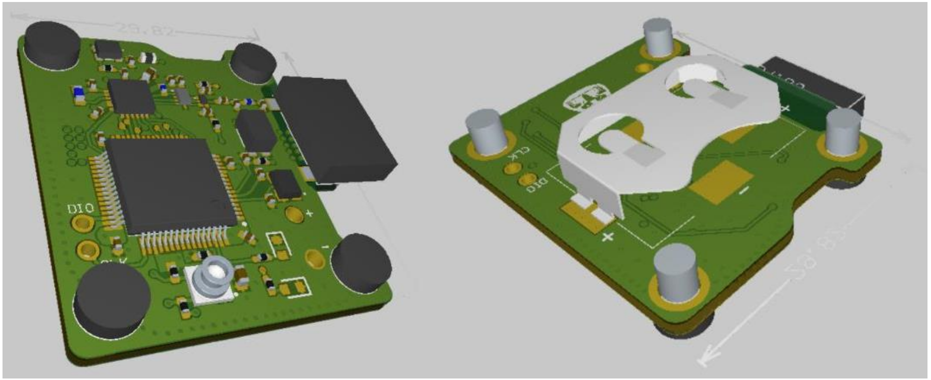
a) TPMS base glued on the car's rim



b) TPMS sensor screwed on the 3D-printed base

PCB description / code organization

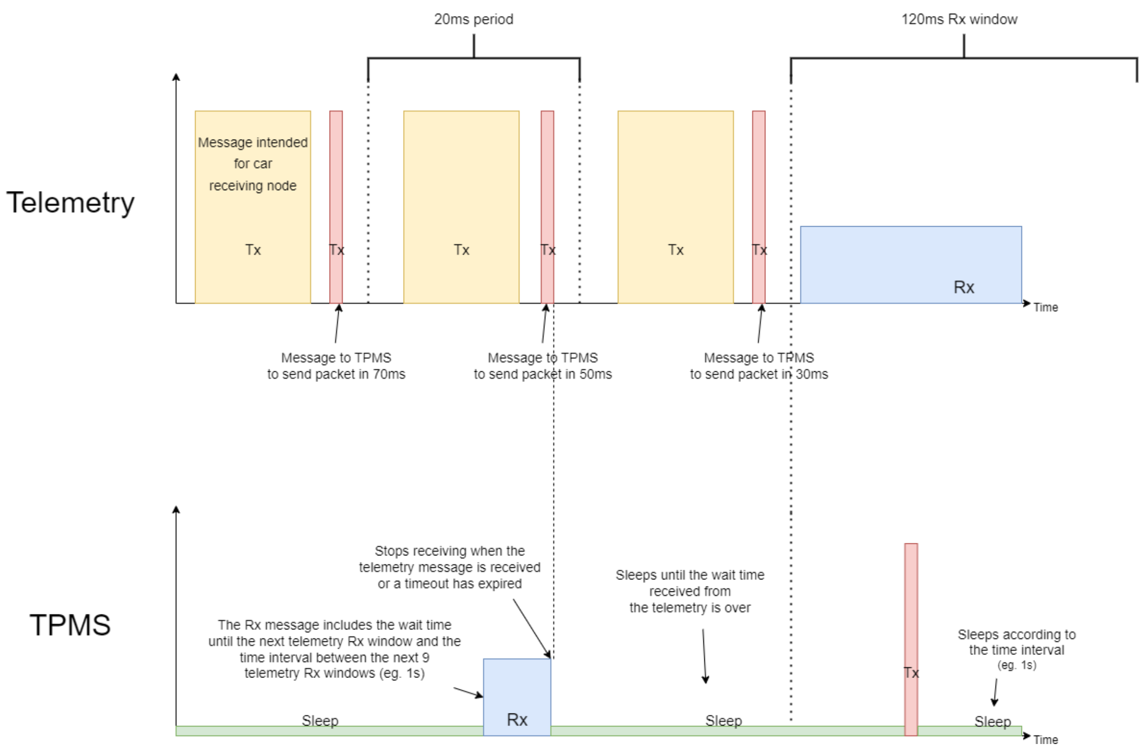
The PCB comprises of a 24-bit ADC digital temperature and pressure sensor interfaced via I2C from an STM32L4 MCU. The specific choice of MCU was made taking into account the extremely low energy consumption at sleep / run states and the availability from the distributors. The microcontroller uses an SX1262 IC to communicate wirelessly with the telemetry and transmit the messages utilizing (G)FSK modulation. Low frequency has been chosen in order to reduce packet loss while maintaining a low Tx power at 0dBm. The whole board is supplied directly from a CR2032W coin cell able to operate at up to 125°C, while all components are rated for at least 85°C (simulations showed that this is a safe range). The whole PCB is mounted using 4 M3 PEEK screws on a 3D printed base from ABS filament, glued with 3M on the rim. This ensures a safe mounting and an easy replacement of the battery or the board.
Pertaining to the code used, the TPMS alternates between 10s sleep and 100ms Rx state. At sleep the sensor is not supplied and the MCU / Radio transceiver are put to stop2 / sleep mode, maintaining their consumption to 20μA. This cycle is only interrupted if the telemetry board starts transmitting messages, so they are detected by the TPMS sensors during their Rx window. The ID of the telemetry message determines the action taken by the TPMS. Thus, the board can either transmit the pressure / temperature of the tire (as it will be explained) or change dynamically its transmitting power / RF frequency, enabling modifications without having access on the board.



Board’s front and rear view

Communication organization

When the telemetry board is supplied, it transmits 2 different types of messages with different sync words every 20ms. The first 50-bytes packet is intended for the receiving node while the second 5-bytes packet is intended for the TPMS. This message informs all the TPMS on when the next Rx window of the telemetry will be, as well as the time interval of consequent telemetry Rx windows (0.5s, 1s, 3s). When the Rx window time arrives, the telemetry dedicates 120ms for Rx and receives the messages from the TPMS. In order to avoid collisions and since every TPMS sensor is assigned a unique ID (0 to 3), the final wait time of a TPMS is: wait_time_from_telemetry_ms + ID * 20ms. After the Rx window, the telemetry continues transmitting the 2 types of packets again until the next Rx window and the TPMS transmit another 9 packets before listening for the telemetry again. These 9 packets are sent every 0.5s, 1s or 3s taking into account the time interval desired by the telemetry (05.s, 1s, 3s). Although this approach requires very thorough time calibration, it is energy efficient since the sensors receive only 1 message for every 10 they transmit. The first message wait time is dictated dynamically enabling automatic recalibration of the sensors and compensating for possible drifts, while the other 9 have a predefined delay from the previous one.



The image above describes the functionality of the TPMS when telemetry messages are detected. If the telemetry stops transmitting messages, the TPMS will enter the 10s sleep – 100ms Rx cycle