BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIO ELECTRONICS

ÚSTAV RADIOELEKTRONIKY

DEVELOPMENT MODULE FOR RADAR SAFETY SENSOR IN SINGLE-TRACK VEHICLES

VÝVOJOVÝ MODUL PRO RADAROVÝ SENZOR BEZPEČNOSTI JEDNOSTOPÝCH VOZIDEL

SEMESTRAL THESIS
SEMESTRÁLNÍ PRÁCE

AUTHOR Martin Ťavoda
AUTOR PRÁCE

SUPERVISOR Ing. Marek Honek
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Semestral Thesis

Bachelor's study program **Electronics and Communication Technologies**

Department of Radio Electronics

Student: Martin Ťavoda ***ID:*** 240703

Year of study: 3 ***Academic year:*** 2023/24

TITLE OF THESIS:

Development Module for Radar Safety Sensor in Single-Track Vehicles

INSTRUCTION:

BP(K)C-SEP:

Familiarize yourself with the client's requirements for the development module and with the existing radar sensor. Create a hardware design for the module that will include the main MCU, radar sensor connectivity, IMU and wireless connectivity to connect to existing commercial cyclo-computers or smartwatches.

BP(K)C-BAP:

Learn about the MicroROS framework and ROS2 for peripheral control and connection to the test system. Make a prototype of the development module according to your design. Create a firmware design and implement it. Test the prototype module with a radar sensor, verify its functionality and evaluate the results.

RECOMMENDED LITERATURE:

[1] NGUYEN, Peter. MICRO-ROS FOR MOBILE ROBOTICS SYSTEMS. Sweden, 2014. Thesis for the Degree of Master of Science in Engineering. Mälardalen University School of Innovation Design and Engineering Västerås. Vedoucí práce Jonas Larsson.

[2] MEINEL, Holger H. Evolving Automotive Radar: from the very beginnings into the future. Daimler AG. Stuttgart, Germany, 2014.

Date of project specification: 22.9.2023 ***Deadline for submission:*** 2.1.2024

Supervisor: Ing. Marek Honek

doc. Ing. Lucie Hudcová, Ph.D.
Chair of study program board

ABSTRACT

The Radar Safety Sensor developed by ALPS ALPINE Co., Ltd uses a FMCW radar to track moving objects behind single-track vehicles. The current radar configuration and antenna allow only one-dimensional Field of view, which has significant object tracking deficiency when the vehicle makes a turn and the radar Field of view tilts. The assumed solution is to add an Inertial Measurement Unit data to the tracking algorithm. This unit should be implemented in a development module, which should also provide development interfaces, output peripherals, power supply, and mounting for the radar. This thesis discusses the hardware design of the development module.

KEYWORDS

embedded device, development module, inertial measurement unit, cyclo-safety equipment, ROS 2, micro-ROS, FMCW radar

ABSTRAKT

Radarový senzor bezpečnosti jednostopových vozidiel, vyvíjaný spoločnosťou ALPS ALPINE Co., Ltd. využíva radar s technológiou frekvenčne modulovanej postupnej vlny na sledovanie objektov, nachádzajúcich sa za jednostopovým vozidlom. Aktuálne nastavenie radaru a jeho anténa umožňujú iba jednorozmerné zorné pole, čo výrazne zhoršuje sledovanie objektov v zákrutách, keď sa radar nakláňa. Predpokladaným riešením je pridanie dát z inerciálnej meracej jednotky do sledovacieho algoritmu. Táto jednotka by mala byť implementovaná vo vývojom module, ktorý by mal poskytovať vývojové rozhranie, výstupné periférie, napájanie a upevnenie radaru. Obsahom semetrálnej práce je návrh hardvéru predmetného vývojového modulu.

KĽÚČOVÉ SLOVÁ

vstavaný systém, vývojový modul, inerciálna meracia jednotka, bezpečnostné cyklo doplnky, ROS 2, micro-ROS, FMCW radar

---

Typeset by the `thesis` package, version 4.09; https://latex.fekt.vut.cz/

ŤAVODA, Martin. *Development Module for Radar Safety Sensor in Single-Track Vehicles*. Semestral Project. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2024. Advised by Ing. Marek Honek

Author's Declaration

Author:	Martin Ťavoda
Author's ID:	240703
Paper type:	Semestral Project
Academic year:	2023/24
Topic:	Development Module for Radar Safety Sensor in Single-Track Vehicles

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank my technical supervisor, Ing. Juraj Giertl, and my formal advisor, Ing. Marek Honek, for their support, guidance, and patience during this project.

Contents

Introduction **11**

1 Commercial cyclo-safety equipment **12**

1.1 Ride Safety System RS 1000 . . . 12

1.2 Radar Tail Light GARMIN Varia . . . 13

1.3 Safety Radar Sensor . . . 13

2 Embedded Systems **15**

2.1 Inertial Measurement Unit . . . 15

2.1.1 Accelerometer . . . 15

2.1.2 Gyroscope . . . 16

2.1.3 Magnetometer . . . 17

3 Software in embedded systems **18**

3.1 Robot operating system . . . 18

3.1.1 Nodes . . . 18

3.1.2 Topics . . . 19

3.1.3 Services . . . 19

3.1.4 Actions . . . 20

3.1.5 Communication interfaces . . . 21

3.2 Micro-ROS . . . 23

3.3 Real-time operating system . . . 24

4 Schematic design **25**

4.1 ESP32 . . . 25

4.2 USB to UART bridge controller . . . 26

4.3 PHY . . . 27

4.4 IMU . . . 27

4.5 Power management . . . 28

4.5.1 DC/DC converters . . . 28

4.5.2 Power multiplexer . . . 29

4.5.3 Power monitor . . . 30

4.6 ANT+ device . . . 30

4.7 Other componets . . . 30

5 Hardware design **32**

Conclusion **33**

Bibliography **34**

Symbols and abbreviations **38**

List of appendices **40**

A Schematic diagram **41**

B PCB layout in 3D **49**

C PCB layout **51**

List of Figures

1.1 Garmin Varia RTL515 [4] 12
1.2 Difference between straight ride (a) and ride in turn (b) 13
2.1 Mechanical principle of accelerometer [11] 16
2.2 Mechanical principle of gyroscope [13] 16
3.1 Example of ROS 2 graph 19
3.2 Example of ROS 2 Service topology 20
3.3 Example of ROS 2 Action topology 20
3.4 Micro-ROS architecture [26] 23
4.1 Development module block diagram 26
4.2 Power management block diagram 28
5.1 RSS development module with attached Alps Generic Radar 5 32
B.1 PCB layout in 3D, top view 49
B.2 PCB layout in 3D, bottom view 50
C.1 PCB layout, top view 51
C.2 PCB layout, bottom view 52

Listings

3.1 Code example of messages declaration 21
3.2 Code example of services declaration 22
3.3 Code example of actions declaration 22

Introduction

During the development of firmware for a new product, there is usually a need for easy debugging and function testing. This can be done by a development module. It should provide all the much-needed interfaces to interact with the outside world, such as *Universal asynchronous receiver-transmitter* (UART) protocol, *General-purpose input/output* (GPIO) pins, a debugging interface, and function demonstration output elements.

In devices with complex software stacks, for example, radar systems, the first stage of software design is often done on an external powerful device such as a *Personal Computer* (PC) or a *Single Board Computer* (SBC). Unlike the embedded devices with a *Microcontroller Unit* (MCU), the computation units are not resource-constrained therefore the required development flexibility is available.

The radar board has only copper pads or brittle *Surface Mount Technology* (SMT) connectors, not designed to be often engaged. To easily connect external computation units and other components, the development module has to be equipped with easily accessible robust connectors.

This thesis deals with creating a specific development module for *Radar Safety Sensor* (RSS). The module will serve company ALPS ALPINE Co., Ltd. in the development of a commercial device similar to the already existing Ride Safety System RS 1000 or GARMIN Radar Tail Light. These devices are safety equipment for cyclists that offer information about vehicles approaching from behind.

In this case, the designed module will also serve as a demonstration module. To show the correct function of a radar data-based tracking algorithm, developers need a module with some output elements. *Light-Emitting Diodes* (LEDs), buzzer, or wirelessly connected commercial fitness equipment can provide the needed function demonstration.

During the development of the Alps Generic Radar 5 which is the core of the RSS, designers ran into a problem related to missing information about radar tilt in 3D space. This information can be obtained from an *Inertial Measurement Unit* (IMU). Consequently, the SBC can provide sensor fusion of the IMU and radar data, to improve the RSS algorithm accuracy.

To conclude, the development/demonstration module should process data from the IMU and send them to the SBC through a *Robot Operating system* (ROS) *Application Programming Interface* (API). As an output, it should collect result data from the ROS domain and trigger an alert in the form of lighting up onboard LEDs, and beeping a buzzer. Additionally, it should provide robust connectors, power management, ANT+ module for future development, and physical mount to the single-track vehicles, in this case, a bicycle.

1 Commercial cyclo-safety equipment

Since a cyclist riding on a road can hardly see what is behind him, cyclo-safety devices can be helpful tools to inform of a potential threat, for instance fast-approaching car or motorbike. They can provide live video or just simple visual and audio alerts. A blinking rear light or a loud beep is commonly used. The blinking high-brightness light is often triggered when the tailgating vehicle is too close to the bicycle or is approaching too quickly. [1]

1.1 Ride Safety System RS 1000

This cyclo-safety device developed by company ALPS ALPINE Co., Ltd. is a bicycle camera with an integrated hazard detection system based on *Artificial Intelligence* (AI). The wide-angle optics camera is mounted on the seat post or luggage of the bicycle, as illustrated in Figure 1.1. It provides HD image quality of the scene behind. The live video is displayed on the proprietary dash cam app running on a smartphone placed at the handlebars. It also offers a LED rear light. The advantages of this device are AI-based software warnings, which are adapted to the current traffic situation. In addition, it provides a multi-stage brake taillight that informs the vehicle behind about the cyclist's braking. [2]

The disadvantage is the need for the high processing power, because of the HD video data handling by AI-based software. This requirement makes the product very expensive and less affordable for common customers.

Fig. 1.1: Garmin Varia RTL515 [4]

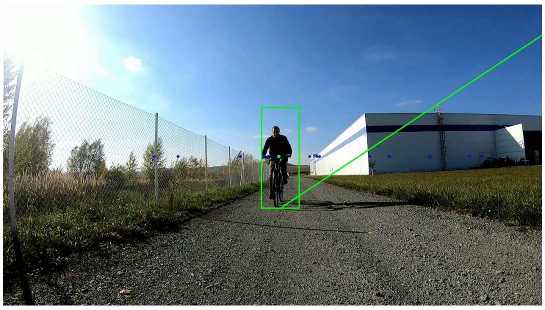
1.2 Radar Tail Light GARMIN Varia

Since the 1970s radar technology has been applied to vehicles as a means to reduce the accident rates on the streets. [3]

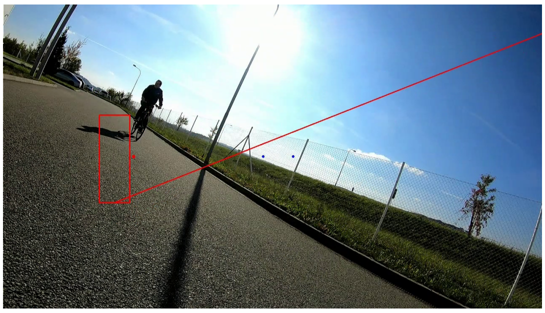
This Garmin Varia Radar Tail Light is similar to the previously discussed product, with the main difference being that this system is based on the radar sensor. [4] That means the cyclist has no real picture of the rear area but only information about the distance between him and the approaching vehicle. This information is displayed as a vertical progress bar on a smartphone app or a Garmin bike computer. [5] It also can be paired with other compatible Garmin devices such as smartwatches. This connection is provided by ANT+ or *Bluetooth Low Energy* (BLE) technology. [6]

1.3 Safety Radar Sensor

ALPS ALPINE Co., Ltd. is developing a new cyclo product using a proprietary radar system with a new tracking algorithm. However, they ran into a problem. In the default scenario, the radar is mounted under the bicycle seat and the bicycle is moving in a straight line (no tilt). The radar sees multiple points, where some of them are reflections from a static background (the blue dots in Figure 1.2a), and the rest are reflections from actual moving targets behind the bicycle.



(a) Correct tracking



(b) Tracking degradation

Fig. 1.2: Difference between straight ride (a) and ride in turn (b)

In this scenario, these two conditions are easily distinguishable in processed radar data output and the ego-motion vector can be calculated from these reflections of static background. Ego-motion vector describes the motion of an object relative to the rigid scene. [7] In this context it consists of two elements, direction (forward or backward), and speed (approaching or receding). In the current radar configuration, the radar *Field of view* (FOV) is only a horizontal line, there is no elevation data, and the ego-motion vector is only one-dimensional. The reflections from moving targets are clustered and classified, and a tracking algorithm is used. From these data, it is possible to estimate the moving target trajectory, and speed (approaching or receding), classify if it's a car or truck, and calculate the possibility of a potential collision.

However, if the bicycle tilts, for example, when turning, the radar FOV also tilts with the same angle and this condition cannot be detected from radar data alone. While the ego-motion vector is only a single dimensional and the height of the radar from the ground changes, the previously tracked object could be tracked incorrectly (see in Figure 1.2b) or can disappear altogether.

One possible solution to this problem seemed to be adjusting the tracking algorithm parameters to be more lenient, but this resulted in insufficient accuracy. The development team decided to try to add to the tracker algorithm some external information about radar tilt and position in space to enhance the tracking estimation. This information can be obtained from an IMU located in a development module.

2 Embedded Systems

Embedded systems are microprocessor- or microcontroller-based systems, which are designed for a dedicated function. They are usually implemented into a larger electrical or mechanical system and are often used in applications with size, weight, power, or cost constraints. Unlike general-purpose systems for multiple tasks, embedded systems are developed for a specific purpose. They are widely used in various areas, such as commercial, industrial, and military applications. Generally, an embedded system consists of hardware parts and application software. In more complex time-dependent applications its firmware includes an *Real-time Operating System* (RTOS). An embedded system contains a microprocessor or microcontroller. In general, a microprocessor is only a *Central Processing Unit* (CPU), hence other components (for example, memories, communication interfaces, peripherals) need to be added. On the other hand, a microcontroller consists of multiple parts which makes it a self-contained system. [8]

2.1 Inertial Measurement Unit

IMU is typically an electromechanical device that consists of an accelerometer, gyroscope, and magnetometer, which can measure linear acceleration, angular velocity, and heading to identify the orientation of an object. IMU is based on converting the detected inertia into a specific output electrical signal. For usage in miniature systems IMUs are designed as *Micro-electromechanical systems* (MEMS). [9]

2.1.1 Accelerometer

Accelerometers are motion sensors that measure linear acceleration, i.e. rate of change in velocity relative to its inertial frame of reference. Usually, they operate in 3 *Degrees of Freedom* (DoF), which means that they can measure linear acceleration in three perpendicular axes. They are often designed as a spring-mass-damper system, applying Newton's second law. The input accelerating force causes displacement of the proof mass, whereas it's held by mechanical springs. This displacement is sensed by a capacitive comb finger structure attached to the proof mass and fixed plates. The output analog voltage is therefore dependent on this capacitance. This architecture is shown in Figure 2.1. The mentioned mechanical design must be replicated three times, once for each axis of a 3 DoF accelerometer. [10]

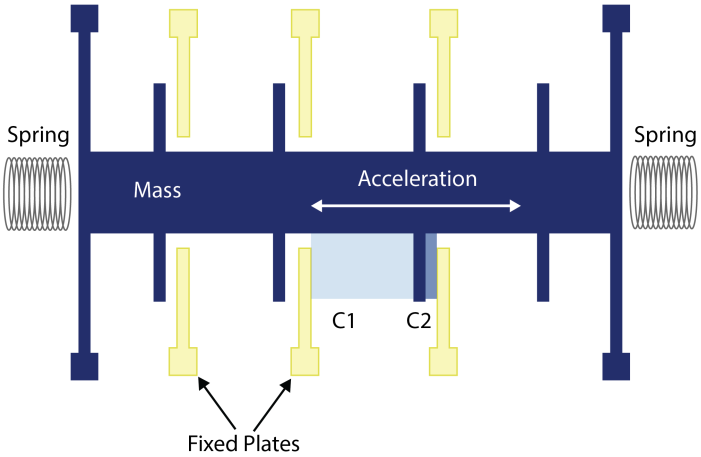


Fig. 2.1: Mechanical principle of accelerometer [11]

2.1.2 Gyroscope

The second part of common IMU is a gyroscope, whose mechanical principle is illustrated in Figure 2.2. It measures the angular (rotary) rate of a rigid element. Most of the MEMS gyroscopes are of *vibratory* kind, i.e. the applied angular velocity is detected by vibrating elements. The most common principle of measuring angular velocity is the method using *Coriolis effect*. This effect caused by *Coriolis force* is an apparent deflection of the trajectory of an object that moves within a rotating frame of reference. [12]

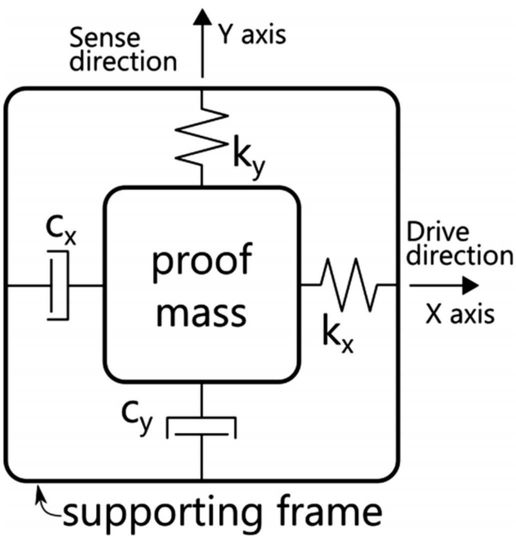


Fig. 2.2: Mechanical principle of gyroscope [13]

The basic structure of most vibratory gyroscopes consists of proof mass mounted by springs above a substrate. The proof mass can oscillate in two perpendicular directions. The *drive* (X) direction and the *sense* (Y) direction. The proof mass vibrates in the x direction with its natural frequency. When the angular force is applied (along the Z axis), Coriolis acceleration causes the oscillation in the sense direction. Therefore, the amplitude of the second oscillation provides information about the input angular velocity. The output signal of the sensor is obtained by demodulation of the secondary oscillation and can be sensed as an output voltage. Note that to achieve three DoF gyroscopes need to include three of these mechanical components, once for each axis. [9] [14] [15]

2.1.3 Magnetometer

Magnetometer is often not a part of the IMU, but rather an external component. It is used to measure the strength and direction of the Earth's magnetic field in order to determine orientation. [9] Various principles are used in the design of the magnetometers, such as magnetoresistive principle, Hall-effect sensors, Fluxgate sensors, and more. While Hall-effect magnetometers lead in high sensitivity and reliability, they face numerous challenges, including the material's incompatibility with standard manufacturing processes. Another modern alternative can be resonant Lorentz-force MEMS magnetometers, which measure the resonant frequency of a microstructure using, piezoresistive, capacitive, or optical techniques. [16]

3 Software in embedded systems

3.1 Robot operating system

Essentially, ROS is an open-source, meta-operating system that provides hardware abstraction, low-level device control, and message-passing between processes running either on one device or multiple devices. It also provides tools and libraries through which a developer can obtain, build, flash, and run code across multiple platforms. [17] ROS 2 contains two client libraries, *rclpy* and *rclcpp*. One is meant for Python and the other for C++ languages. They offer interfaces for users to access the functionalities of ROS 2 core, provided by the ROS client library *rcl*, written in C. Also, they allow users to wrap their code into the ROS 2 application. [18]

The message transfer in the system is based on a strongly typed, anonymous publish/subscribe mechanism. [19] Strongly typed language means that each type of data, in this case, the type of message, is predefined and each created message must correspond to one of these data types. [20] The mentioned publish/subscribe mechanism enables message delivery based on subscription. In other words, the message is addressed by a subject name rather than to a specific recipient. Recipients can subscribe to their preferred subjects using a variety of parameters and get these messages instantly without the need for constant polling. [21] The central component of any ROS 2 system is the ROS graph. It describes a network of nodes and connections between them. The following chapters will discuss the basic concepts of ROS 2, as they are described in the literature [19].

3.1.1 Nodes

A node is one member of the ROS 2 graph, which uses the client library to communicate with other nodes. Nodes can run as processes on one or multiple devices. In a ROS graph, nodes are the units of computation and each node should do only one logical thing. They can publish to a *topic* (a ROS communication channel) to deliver data to other nodes or subscribe to a topic to get data from other nodes. A node can be configured as *service client* to use another node for a specific computation. The node providing functionality for others is called *service server*. This principle can also be used for long-running computations, *actions*, where the nodes are named *action server* and *action client*. Nodes can be configured via *parameters*. Finally, a node can be a complex combination of subscribers, publishers, service clients or servers, and action clients or servers, all at once. From a programmer's point of view, these functions are objects of the ROS 2 client libraries. [18] [19]

3.1.2 Topics

A topic is an ROS 2 interface between nodes, that can be described as a virtual communication channel. A node that sends a message to a topic is called a *publisher*. On the other side, the node that obtains this message is called a *subscriber*. One topic can have multiple publishers and subscribers, as is shown in the Figure 3.1. Also, one node can consist of multiple publishers and subscribers who are performing different tasks. [18] [19]

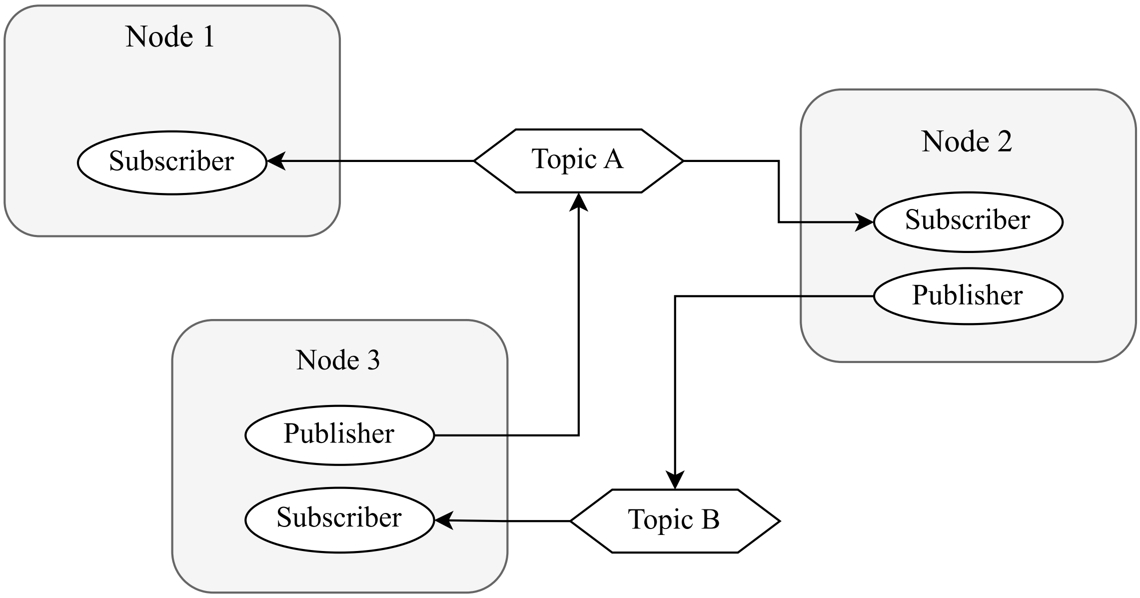


Fig. 3.1: Example of ROS 2 graph

3.1.3 Services

Service is the second of three communication interfaces in the ROS 2 graph. This method requires *service client* and *service server*. a node sends data to the *service server*, which provides some computation and then sends the answer back. The computation should be as short as possible because the requesting node is waiting for an answer. For complex time-consuming data processing, the *actions* should be used. Services have a service name, like topics (but in different namespaces). In ROS 2 environment there can be multiple nodes as service clients, but only one service server with a specific name. [18] [19]

In Figure 3.2 Node 1 is a server client, who sends a request to the service. Node 2 as a service server receives this request, does some computation, and replies with a response to the client.

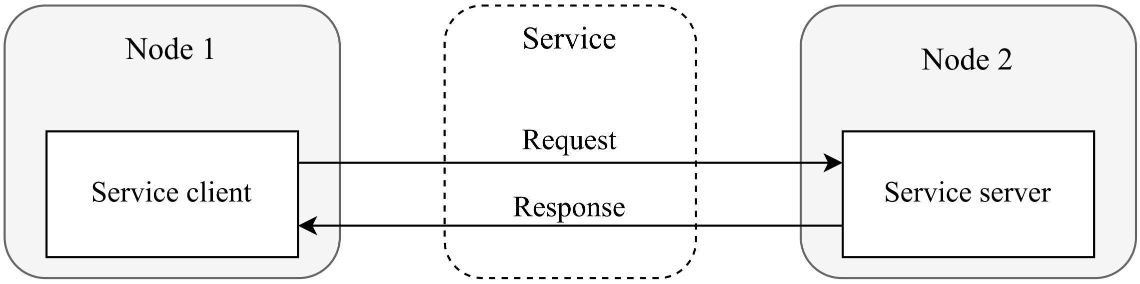


Fig. 3.2: Example of ROS 2 Service topology

3.1.4 Actions

This is the most complex communication interface. It conceptually includes topics and services. Action is a call for a long-running procedure with feedback and the ability to cancel the goal. For instance, the robot central unit calls an action to turn a camera, which can take a longer time. While the camera performs the turn it can send multiple feedback on the task progress to the central unit. [18] [19]

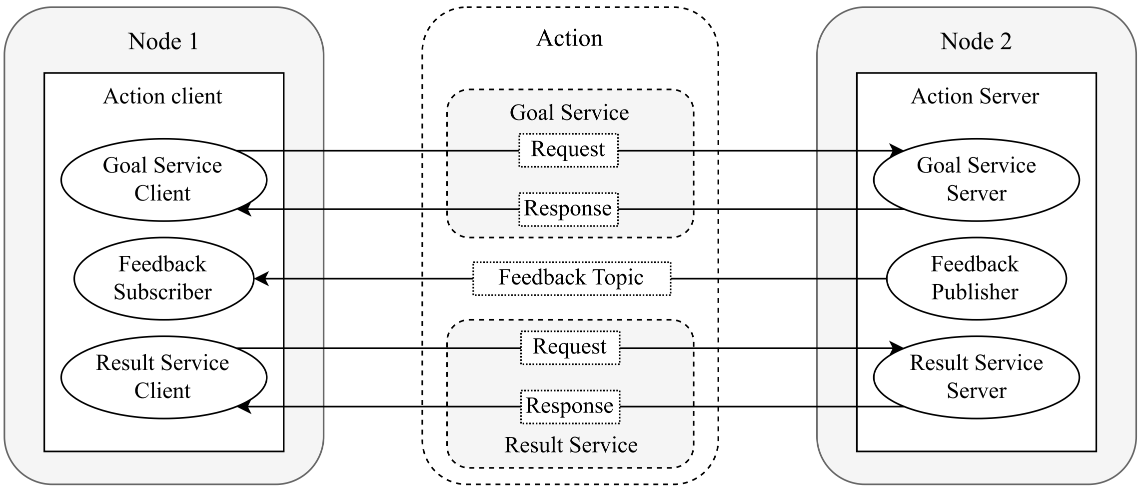


Fig. 3.3: Example of ROS 2 Action topology

Figure 3.3 shows the actions topology in ROS. This can be divided into three sections: goal service, feedback topic, and result service. The Node 1 is an *action client*. It sends a request for some required task to Node 2, an *action server*. This request is received by the goal service, which immediately replies with a goal response (an acknowledgment). Node 2 starts performing the task, during which it should regularly send feedback through a feedback topic. This topic can be subscribed not only by Node 1. After Node 2 successfully finishes the task, it responds to the result service client in Node 1. [22]

3.1.5 Communication interfaces

To use the mentioned topic, services, and actions the ROS 2 communication interfaces have to be defined. These interfaces are described in text files and coded in a simplified description language, the *Interface Definition Language* (IDL). This makes it possible for ROS tools to automatically generate source codes for the interfaces in different languages. There are three types of text files: `.msg` for topics, `.srv` for services, and `.action` for actions.

Messages

Topics use *messages* to define data types transferred through them. To customize these messages, the definition in the text file is written. On each line, there is a *field type* followed by a *field name*. The field can also be set with a default value. An example can be found in the Listing 3.1. This example message structure describes a topic, that has three fields: a 32-bit unsigned integer, a string, and a float with assigned value. Besides these, there can be used many other built-in or user-defined data types. Some of them are shown in Table 3.1, others can be found on the ROS 2 wiki. [19]

Type name	**C++**	**Python**	**DDS type**
bool	bool	builtins.bool	boolean
char	char	builtins.str*	char
float32	float	builtins.float*	float
uint16	uint16_t	builtins.int*	short
string	std::string	builtins.str	string

Table 3.1: Examples of ROS2 messages data types

There can also be defined arrays using the built-in types. Additionally, the IDL supports constant values, which cannot be changed programmatically later. Note the use of the assignment character.

```
% declaration example
int32 angle
string myName
float64 sample 58

% example of data arrays
int32[] unbounded_integer_array
int32[5] five_integers_array

string string_of_unbounded_size
```

```
string<=10 up_to_ten_characters_string

string[<=5] up_to_five_unbounded_strings
string<=10[<=5] up_to_five_strings_up_to_ten_characters_each

% example of constant values
int32 X=123
string EXAMPLE='Some default text.'
```

Listing 3.1: Code example of messages declaration

Services

Services are defined similarly but in the `.srv` file. The file needs to include a request message type and response message type separated by `---` sequence, as shown in Listing 3.2. There can be various data variables and constants. A service, that takes a string as a request and returns an integer as a response with some additional constants can be defined like in the example. ROS 2 also supports referring to a message from different packages.

```
string assignment
int8 ID=1
---
uint32 response
uint32 CODE=123456
another_pkg/YetAnotherMessage value
```

Listing 3.2: Code example of services declaration

Actions

As described before, actions consist of three data blocks. Each of them needs to be described in `.action` file separated by `---` sequence. The order must be maintained as follows. Firstly a request field, then a feedback field, and lastly a response field. There can be arbitrary numbers of each field or none. An action that receives a request for a turn of a camera as an integer, sends integer-type feedback and finally returns a response string can be defined like in the example (Listing 3.3).

```
int8 angle
---
uint32 progress
---
string result
```

Listing 3.3: Code example of actions declaration

3.2 Micro-ROS

While ROS 2 works effectively on personal computers, its' functionalities, packages, and libraries are not compatible with resource-constrained embedded devices. [23] For implementing the ROS 2 onto a MCU the micro-ROS can be used. It brings all ROS 2 major core concepts such as nodes, publish/subscribe mechanism, client/service topology, node graph, and lifecycle. The client API of micro-ROS is composed of the standard ROS 2 Client Support Library (rcl) and a set of convenient functions and extensions (rclc). This combination of libraries in micro-ROS is optimized for MCUs and after the initialization phase, it can be used without any dynamic memory allocations.

The connection of MCU and ROS 2 domain is provided by a micro-ROS agent. The agent has to run on a normal operating system, for example on a PC or SBC. This agent connects micro-ROS nodes on MCU seamlessly with the standard ROS 2 system. This allows accessing micro-ROS nodes with all known ROS 2 tools and APIs just as normal ROS 2 nodes. [24]

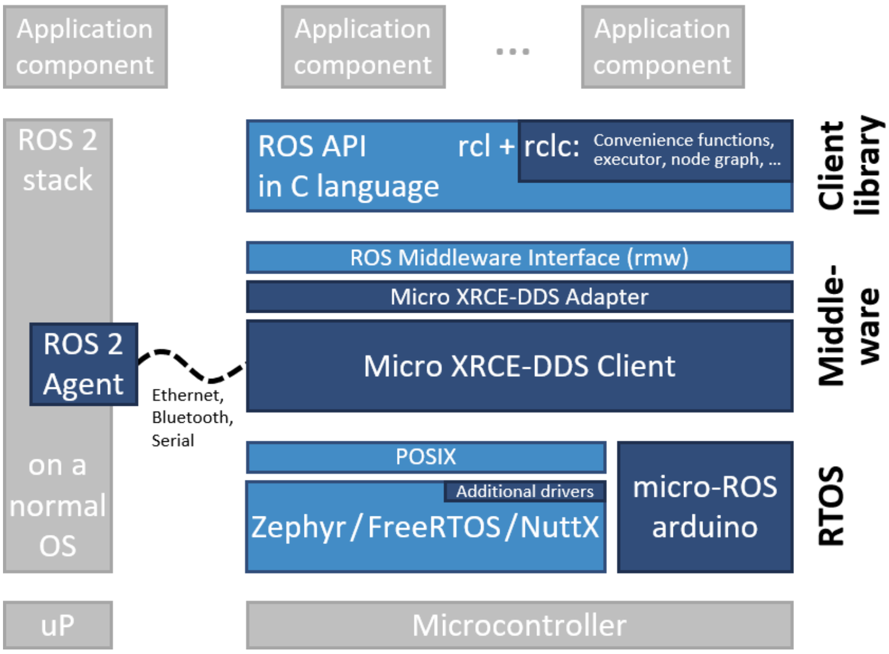


Fig. 3.4: Micro-ROS architecture [26]

Micro-ROS uses Micro XRCE-DDS by eProsima. DDS stands for *Data Distribution Service* and it is a fully distributed communication middleware, which provides low-latency messaging. [25] This micro-ROS's middleware meets all requirements for deeply embedded systems, which is what the abbreviation XRCE (Extremely Resource Constrained Environments) expresses. To avoid dynamic allocations at runtime the static memory pools were introduced for the integration with *ROS middleware interface* (rmw) in the micro-ROS stack. The middleware also provides built-in support for serial transports, *User datagram protocol* (UDP) over Ethernet, Wi-Fi, and more. [24]

The Figure 3.4 shows the layered architecture of micro-ROS. The ROS agent runs on host *Operating system* (OS) and communicates with Micro DDS middleware, which is managed by the RTOS on MCU. The Client library uses the rmw as an interface for the middleware. On the top can be multiple application components providing versatile tasks. Dark blue components are developed specifically for micro-ROS, whereas light blue components are taken from the standard ROS 2 stack.

3.3 Real-time operating system

Most operating systems allow multitasking which is possible thanks to the scheduler, a part of an OS. This scheduler decides which program to run when and therefore it creates an illusion of simultaneously running processes by rapidly switching between them. In RTOS, the scheduler is designed to provide a deterministic execution pattern. This is especially important in embedded systems because they often have real-time requirements. Real-time requirements mean that the embedded system must respond to a certain event within a strictly defined time – the *deadline*. In the FreeRTOS operating system, determinism is allowed by assigning a priority to each thread of execution, which is called a *task*. FreeRTOS is designed to be small enough to run on a MCU. [27]

4 Schematic design

The schematic design of the development module for the RSS consists of multiple blocks. The main part is the microcontroller ESP32 which collects the raw data from the IMU using the I^2C bus. These data are then processed and sent to the ROS domain via Ethernet. Ethernet connection is provided by *Media access control sublayer* (MAC) and *Physical layer* (PHY) interfaces. MAC is included in ESP32 and PHY is an external component. They communicate with each other by *Reduced Media-independent interface* (RMII). Ethernet was preferred over WiFi due to its lower latency.

On the SBC are running multiple processes, also ROS2 graph members, which merge the IMU and radar data and provide sensor fusion. The output of this process is sent back to the ESP32, whose task is to provide the alert by visual or sound peripherals. There is also a place for the ANT+ module, which should make a connection with external user devices; however, this feature is not yet supported. The design also contains the power management of four power lines and a *Universal Serial Bus* (USB) interface for programming and debugging the microcontroller. The block diagram in Figure 4.1 illustrates the hardware design. The radar is connected by spring-loaded pins and debug *Flexible flat cable* (FFC). Some parts of the design are shown in gray because the company requested them and they are not used in the first version of this project. All of the previously mentioned blocs are discussed in the next sections, referring to the schematic diagram shown in Appendix A.

4.1 ESP32

The Espressif ESP32 was selected as a main microcontroller because it has integrated WiFi, BLE, and MAC modules. Also, it has a great ratio between usage simplicity and computing power among available microcontrollers. The module used is an ESP32-WROVER-E-N16R8 with an integrated *Printed circuit board* (PCB) antenna. It comes with a 16 MB external *Serial Peripheral Interface* (SPI) flash and an additional 8 MB *Pseudo static RAM* (PSRAM). The core is the ESP32-D0WD-V3 chip. It has two available cores that can be individually controlled and the CPU clock is adjustable between 80 MHz to 240 MHz. This module includes many peripherals. This design uses GPIO, I^2C peripheral, two UART controllers, 12-bit *Analog-to-digital converter* (ADC), and the *Remote Control Transceiver* (RMT), among others. [28]

The schematic diagram in Appendix A shows also the additional components around ESP32. There are resistors for bootstrapping pins, the RC circuit on the EN

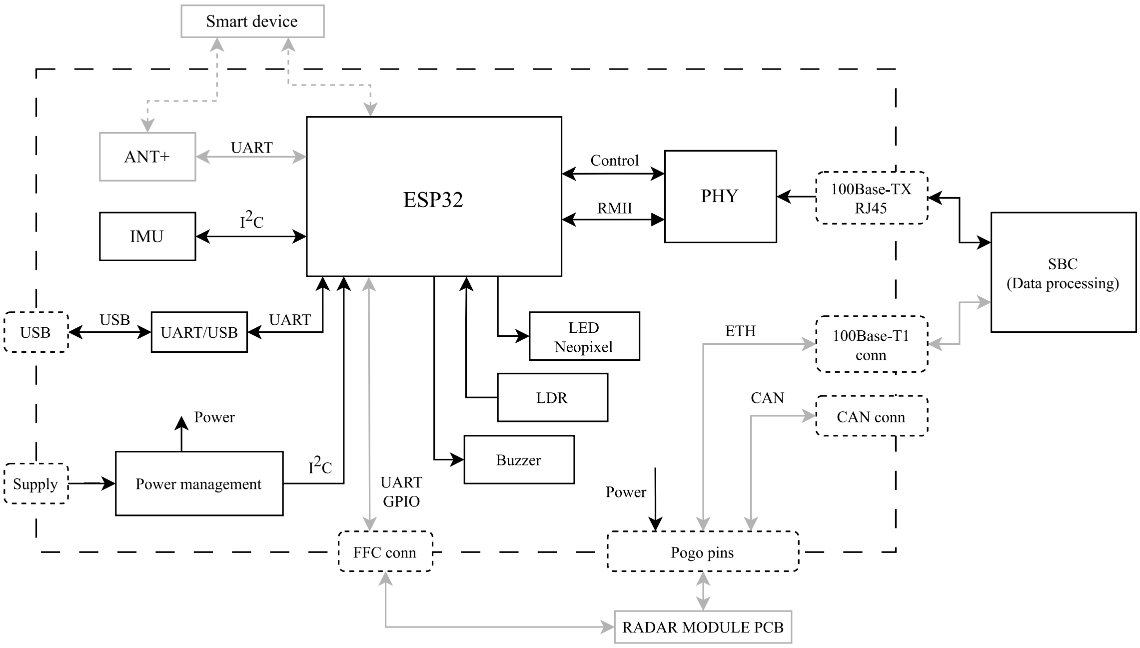


Fig. 4.1: Development module block diagram

pin enabling correct start-up, and some additional filtering and decoupling capacitors. The used I^2C bus is configured to work in Fast-mode, so the 4.7 kΩ resistors were chosen, according to [29]. During the boot from UART, there needs to be a low level on GPIO0. This is allowed by the TACT switch or for automatic setup during programming, the RTS pin of UART to USB converter is used. Also, the EN pin (equivalent to reset) is serviced manually by switch or automatically via the DTR pin.

4.2 USB to UART bridge controller

The Silicon Labs CP2102N is a modern and simple solution using minimum external components. It includes a USB 2.0 full-speed function controller, USB transceiver, oscillator, and can operate up to 3 Mbaud. In this use case, it is designed in a bus-powered configuration so the VBUS line is routed to the main voltage converters. The internal 3.3 V *Low Drop-out voltage* (LDO) regulator is powering only the internal circuitry of this chip. This chip offers four GPIO pins, of which two can be selected as LED indicators of ongoing communication. For USB *Electrostatic discharge* (ESD) protection, the recommended *Transient voltage suppression* (TVS) diode array is implemented. The ferrite bead for high-frequency noise suppression on the VBUS is used. The RTS and DTR pins are wired through transistors to the previously mentioned EN and GPIO0 pins of ESP32. [30] Additionally, the CC pins of the USB receptacle are pulled down by 5.1 kΩ resistors to ensure USB C power compatibility. According to the USB C application note [31], the host controller can supply current up to 3 A.

4.3 PHY

The Ethernet communication is provided by IC$^+$ Single Port 100Base-TX RMI-I/TP Fast Ethernet Transceiver IP101GR. Since it uses a reduced version of *Media-independent interface* (MII) and this requires only two data lines, the clock speed needs to be set up to 50 MHz. The clock signal is sourced from an external 25 MHz oscillator connected to the PHY chip. It has to be also synchronized with MAC by a specific pin. The oscillator circuit can be tuned by adjusting the capacitance or resistance on prepared footprints. For correct bus setup and chip configuration a few pull-up and pull-down resistors are used. Instead of using separate magnetics and LEDs, they are integrated into the RJ45 connector, which also provides Bob-Smith termination to reduce *Electromagnetic interference* (EMI). [32]

4.4 IMU

As an Inertial Measurement Unit is utilized the MPU-6050 by TDK InvenSense. This unit includes a gyroscope, accelerometer, and auxiliary I^{2}C for connecting an external magnetometer. Useful capabilities are low operating current, internal 1024 byte FIFO buffer, main I^{2}C and SPI interface, and integrated 9-Axis Motion-Fusion by the on-chip Digital Motion Processor. [33] This IMU is used very often in many applications and has good availability. Competitive Bosch BMI085 IMU had a bit better parameters but it was not available in store LSCS.

This application of the IMU in the development module is only for testing purposes, so there were no special requirements for accuracy and stability. After successful proof of concept, where the bicycle tilt data will be helpful in radar data processing, additional high-quality and more expensive IMU can be attached to the I^{2}C connector on board.

In the schematic diagram in Appendix A, MPU-6050 is surrounded by decoupling capacitors according to the datasheet. [33] In the power supply path the ferrite bead is included. The interrupt pin is routed to the GPIO pin of the MCU because the IMU can raise an interrupt on special events, such as new data availability or an accelerometer event interrupt.

4.5 Power management

The power management of this development board is a complex circuit. It consists of four power lines, three DC/DC converters, a two-channel power monitor, and a power multiplexer for USB or battery priority. These components are shown on the block diagram in Figure 4.2.

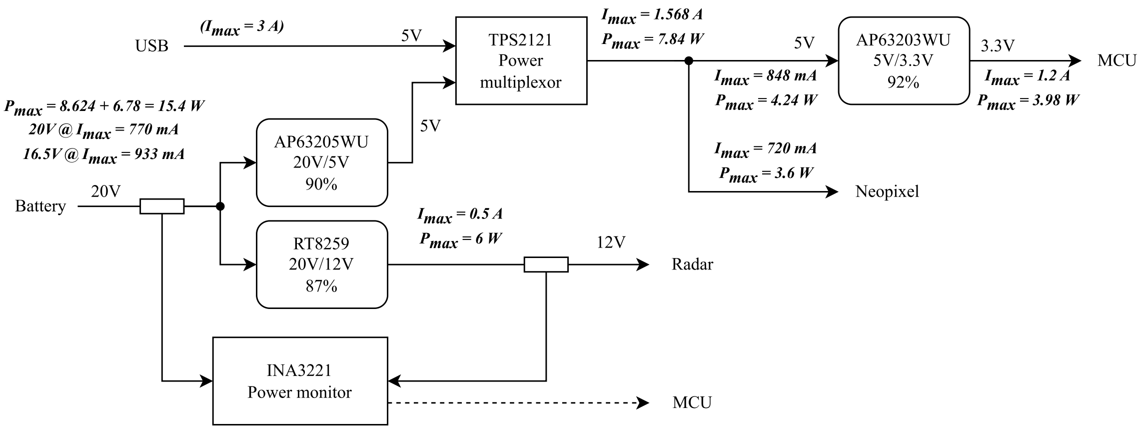


Fig. 4.2: Power management block diagram

4.5.1 DC/DC converters

RT8259

The input 16.5 V – 20 V power line is supplied from an external Li-Ion accumulator with a nominal voltage of 18 V. This line is routed to the DC/DC converter, via a polarity protection transistor and sense resistor for the power monitor. The used Richtek RT8259 is a switching DC/DC converter with an input voltage range of 4.5 V to 24 V, an adjustable output voltage level, and an output current of up to 1.2 A. In this use case, the output power line is set to $V_{OUT} = 12\,\mathrm{V}$ and it powers the radar with a maximum current of $I_{OUT} = 500\,\mathrm{mA}$. In these conditions, the efficiency is around $\eta = 92\%$, according to the datasheet. [34] Therefore, the input power P_{IN} needs to be higher by the losses in the converter, calculated in Eq. 4.1.

$$P_{IN} = V_{OUT} \cdot I_{OUT} + V_{OUT} \cdot I_{OUT} \cdot (1-\eta) = 0.5 \cdot 12 + 0.5 \cdot 12 \cdot (1-0.92) = 6.78\,\mathrm{W} \quad (4.1)$$

A strong point of this converter is its relatively high switching frequency of $f = 1.4\,\mathrm{MHz}$, which requires only a small output inductor. The output voltage is set by a voltage divider connected to the feedback pin. The first resistor in the divider was chosen as $R_1 = 51\,\mathrm{k\Omega}$ and the second one was calculated from Eq. 4.2.

$$R_2 = \frac{V_{FB} \cdot R_1}{V_{OUT} - V_{FB}} = \frac{0.8 \cdot 51 \times 10^3}{12 - 0.8} = 3.642\,\mathrm{k\Omega} \tag{4.2}$$

The feedback voltage $V_{FB} = 0.8\,\mathrm{V}$ is defined by internal voltage reference. The closest value for resistor R_2 in the E24 series is the resistance of $3.6\,\mathrm{k\Omega}$. The chosen resistors have a tolerance of 1%, which makes the inaccuracy in output voltage in the range of $< 11.9\,\mathrm{V}; 12.36\,\mathrm{V} >$. This inaccuracy is acceptable because the powered radar has a linear regulator with a wide voltage range at the input. Additionally, the enable pin of the Richtek converter is wired to the MCU, to be able to control the power to the radar module.

AP63205/03WU

For creating the 5 V and 3.3 V power rails in the design, the Diodes AP632XX series *Integrated circuits* (ICs), with fixed output voltages are used. The decisive advantages of these chips were a small compact package TSOT26, high switching frequency $f = 1.1\,\mathrm{MHz}$, 2 A continuous output current, and high efficiency. The requested input power was again calculated by the equation 4.1, but with efficiency 90% for the 5 V converter and 92% for the 3.3 V one, according to the datasheet. [35] These converters are connected in cascade. The 5 V rail powers the programmable Neopixel RGB LEDs and the 3.3 V rail supplies all digital circuity.

4.5.2 Power multiplexer

This development module is supplied from two sources. The 5 V USB connector and the 18 V external battery. To be able to control the priority of the power supply line and prevent the reverse current flow back to the sources, the Texas Instrument TPS2121 priority power multiplexer is used. It has the ability to automatically detect, select, and seamlessly transition between available inputs. It has low forward resistance $R_{ON} = 56\,\mathrm{m\Omega}$ due to an ideal diode implementation and high operating current, up to 4.5 A.

The utilized transition mode XCOMP provides an automatic switchover between the battery source (IN1) and the USB source (IN1) of the chip. The voltages on sense pins PR1 and CP2 are internally compared to the voltages on source pins. In this design, the battery source has higher priority, therefore the PR1 pin is directly connected to the IN1 power source. When the 5 V power rail sourced from the battery drops below 4.5 V, *Multiplexer* (MUX) switches over to the USB power source. Thus the USB power rail is sourced only when the USB is connected and the voltage on the 5 V battery rail is higher than 4.5 V. This threshold is set by the voltage divider connected to the CP2 sense pin. Additionally, the voltage divider is also prepaired for the PR1 pin however, is unused at this stage of the project. [36]

4.5.3 Power monitor

INA3221 by Texas Instruments is a Triple-channel High-side measure shunt and bus voltage monitor with I^2C bus. It also supports critical and warning alerts to detect out-of-range conditions for each channel. In this design, only two channels are used. The first one is measuring voltage and current on the 18 V battery power line and the second one is monitoring the 12 V power line for the radar. The current measurement is done by measuring the voltage on shunt resistors and the voltage measurement is provided by the negative side sense pin of each channel. As recommended in the datasheet [37], the sense signals are filtered by two series resistors and a parallel capacitor. Since the power monitor is sensitive to harmonics of the 500 kHz sampling frequency above 1 MHz and the DC/DC converters operate around this frequency, this filter is necessary. The power valid (PV) pin is routed to the MCU, for detection of exceeding the undervoltage threshold set by software.

4.6 ANT+ device

This RSS module should communicate with external devices not only through Bluetooth but also using the ANT+ technology by Garmin. This technology is supported by only a few modules. One of them is the D52 ANT *System on chip* (SoC) Module based on the nRF52832 SoC by Nordic Semiconductor. This chip offers 2.4 GHz transceiver, ARM® Cortex®-M4 32-bit processor, and many peripherals, such as UART, SPI, and I^2C. [38] However, implementing software for this module is complicated due to the complex library set and licensing difficulties. Therefore, only the schematic design and PCB layout of this SoC is provided in this phase of the project, and the chip itself is not populated.

The surrounded circuit provides UART connection with MCU, RC circuit to provide correct start-up, *Serial Wire Debug* (SWD) programming interface, and baud-rate setting pads. There are also exposed test points, to ensure future connection with additional components.

4.7 Other componets

The visual alert is provided by numerous programmable RGB LEDs by Neopixel. The chosen WS2812B mini is the smaller version of the widely used WS2812 LED.

It has a smaller package, but mainly it has a lower forward current of $I_f = 16\,\mathrm{mA}$. The LEDs are supplied by $5\,\mathrm{V}$ and controlled by the digital GPIO of the MCU.

The required brightness of LEDs can be set by the MCU, according to the photoresistor value. This component is the commonly used *Through-hole technology* (THT) photoresistor connected as a voltage divider to the ADC input of the MCU. Additionally, a resistor needs to be added parallelly to the photoresistor to match the voltage range of the ADC.

As an audible alert, a small magnetic buzzer is used. It operates at $5\,\mathrm{V}$, with SPL of $L_p = 80\,\mathrm{dB(A)}$ at $2.7\,\mathrm{kHz}$. The common flyback diode is connected antiparallelly to protect the bipolar transistor from peak voltage generated in the inductor.

Lastly, the radar connectors are designed. One is the connector with spring-loaded pins, which is used for power supplying the radar and interconnecting it with external components through ethernet and *Controller Area Network* (CAN) connectors. The second one is for future use and provides the UART and GPIO connection through FFC.

5 Hardware design

The Alps Generic Radar 5 is an 55 mm×55 mm PCB, with SMT componets on one side. On the other side, there is a radar antenna array. The PCB contains 10 THT pads for power supply and working signals and also a FFC debug connector.

The radar board is attached to the front side of the development module PCB. They are interconnected by the spring-loaded pogo pins. The visualization is shown in Figure 5.1.

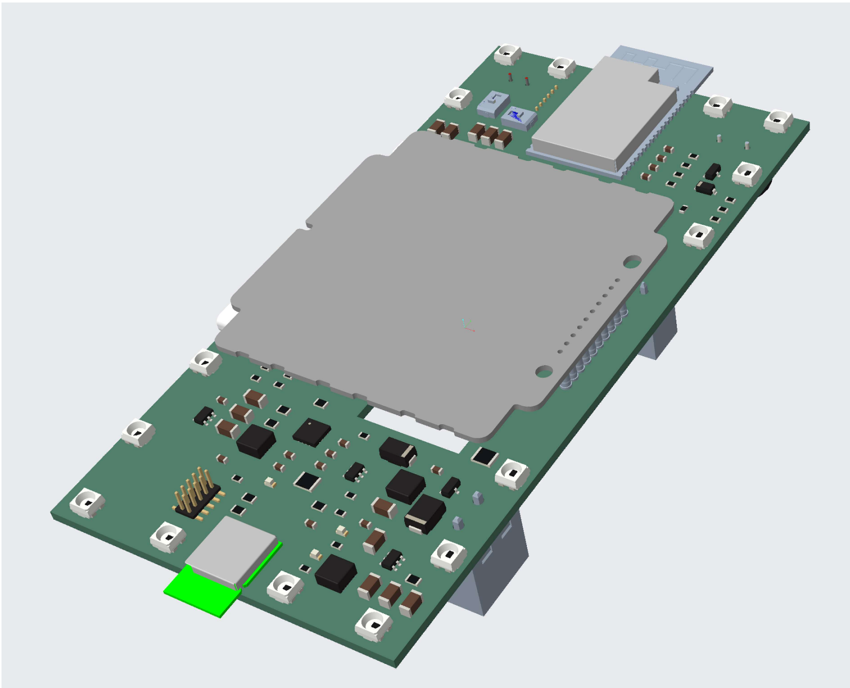


Fig. 5.1: RSS development module with attached Alps Generic Radar 5

There is an aperture in the module's PCB because the radar debug connector is tall and sufficient space must be provided for the counterpart connector. The module has 16 RGB LEDs distributed around the perimeter. They are pointing behind the cyclist together with the radar antenna. The radio SoCs are placed at the top and bottom edge of the module. The IMU is placed exactly in the center to ensure correct readings according to the radar beam. Additionally, all of the THT connectors, such as both ethernets, CAN, and power supply connectors are placed on the rear side of the module. In the appendix B and C is shown the mock-up design of the development board.

In the development stage of the product, the PCB assembly is placed on the rear side of the bicycle cargo rack, because of the presence of the SBC, battery, and other external components.

Conclusion

The work on this thesis was divided into several subsequent parts. The research about cyclo safety equipment and the ROS 2 domain was done at first. All basic API examples of ROS 2 were tried out and the micro-ROS stack was verified on ESP32 MCU. Also the basics about RTOS and IMU were researched. After these first steps, the development module architecture was designed and discussed with the company. The power requirements and component parameters calculations were performed. All components were selected based on datasheet parameters, recommendations, previous experiences, and availability. The schematic diagram was created considering all the potential requirements during future development. The hardware design of the module and connections to the radar and other peripherals were discussed with colleagues. Finally, the mock-up version of the hardware design and PCB layout was created.

The next step will be to route the PCB and therefore finish the hardware design. Then the board will be sent to the JLCPCB manufacturer to produce it and assemble it. During the manufacturing process, the software will be developed on a breadboard prototype. Afterward, the firmware will be flashed to the MCU on the finished PCB and all functions will be tested and debugged. Finally, the company will use the created module in the development of the tracking algorithm for the RSS.

Bibliography

[1] @Shane Miller - GPLama. *Garmin Varia Radar Review: RTL515 // RVR315 // Varia Mobile App*. Online, video. 2023-05-13 Available at: https://www.youtube.com/watch?v=UTLIeaRKN6c. [cit. 2023-12-13].

[2] ALPS ALPINE Co., Ltd. *Ride Safety System RS 1000*. Online. 2023. Available at: https://www.ride-safety.com/. [cit. 2023-12-13].

[3] MEINEL, Holger H. Evolving automotive radar 2014; From the very beginnings into the future. Online. In: *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*. IEEE, 2014, p. 3107. ISBN 978-8-8907-0184-9. Available at: https://doi.org/10.1109/EuCAP.2014.6902486. [cit. 2023-12-23].

[4] Garmin Ltd. *Varia RTL515 Radar Tail Light*. Online, Figure. In: garmin.com. 2023. Available at: https://www.garmin.com/en-US/p/698001. [cit. 2023-12-13].

[5] @Suntinel. *Radar na kole? - Garmin Varia RTL 515*. Online, video. 2023-04-22 Available at: https://www.youtube.com/watch?v=lglpwBA5wJo. [cit. 2023-12-13].

[6] Garmin Ltd. *VARIA REARVIEW RADAR RTL515/ RTL516 Owner's Manual*. Online. 2020. Available at: https://www8.garmin.com/manuals/webhelp/GUID-C41F445D-457F-447D-88C8-FE286BF157E9/EN-US/Varia_RTL515_516_OM_EN-US.pdf. [cit. 2023-12-13].

[7] KHAN, Naila Habib a ADNAN, Awais. Ego-motion estimation concepts, algorithms and challenges: an overview. Online. *Multimedia Tools and Applications*. 2017, vol. 76, issue. 15, p. 16584. ISSN 1380-7501. Available at: https://doi.org/10.1007/s11042-016-3939-4. [cit. 2023-12-26].

[8] SWARUP, Bhunia a TEHRANIPOOR, Mark. *Chapter 2 - A Quick Overview of Electronic Hardware*. Online. Hardware Security. Morgan Kaufmann, 2019. ISBN 978-0-12-812477-2. Available at: https://doi.org/https://doi.org/10.1016/B978-0-12-812477-2.00007-1. [cit. 2023-12-15].

[9] ADVANCED NAVIGATION. *Inertial Measurement Unit (IMU) – An Introduction*. Online. 2023. Available at: https://www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/ [cit. 2023-12-17].

[10] STHUTHI, A.; NITHYA, G.; NAVEEN KUMAR, K. a RAMYA, S.S. Design and simulation of dual-axis MEMS accelerometer. Online. *Materials Today:*

Proceedings. 2023, p. 1. ISSN 22147853. Available at: https://doi.org/10.1016/j.matpr.2023.05.569. [cit. 2023-12-16].

[11] Level Developments. *MEMS Accelerometer*. Online, Figure. Available at: https://www.leveldevelopments.com/wp/wp-content/uploads/MEMS-Accelerometer.png [cit. 2023-12-16].

[12] Britannica, The Editors of Encyclopaedia. Coriolis force. Online. *Encyclopaedia Britannica*. 11 Jul. 2023. ISSN 1085-9721. Available at: https://www.britannica.com/science/Coriolis-force. [cit. 2023-12-17].

[13] SPACIL, Tomas; RAJCHL, Matej; BASTL, Michal; NAJMAN, Jan a APPEL, Martin. Design of Deterministic Model for Compensation of Acceleration Sensitivity in MEMS Gyroscope. Online, Figure. In: SZEWCZYK, Roman; KREJSA, Jiří; NOWICKI, Michał a OSTASZEWSKA-LIŻEWSKA, Anna (ed.). *Mechatronics 2019: Recent Advances Towards Industry 4.0*. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, 2020, p. 287. ISBN 978-3-030-29992-7. Available at: https://doi.org/10.1007/978-3-030-29993-4_35. [cit. 2023-12-16].

[14] EGRETZBERGER, Markus; MAIR, Florian a KUGI, Andreas. Model-based control concepts for vibratory MEMS gyroscopes. Online. *Mechatronics*. 2012, vol. 22, issue 3, p. 241. ISSN 09574158. Available at: https://doi.org/10.1016/j.mechatronics.2011.06.003. [cit. 2023-12-16].

[15] PISHROBAT, M. Hoseini a KEIGHOBADI, J. Model Predictive Control of MEMS Vibratory Gyroscope. Online. *IFAC Proceedings Volumes*. 2014, vol. 47, issue 3, p. 7278. ISSN 14746670. Available at: https://doi.org/10.3182/20140824-6-ZA-1003.02322. [cit. 2023-12-16].

[16] MBAREK, Sofiane Ben; ALCHEIKH, Nouha; OUAKAD, Hassen M. a YOUNIS, Mohammad I. Highly sensitive low field Lorentz-force MEMS magnetometer. Online. *Scientific Reports*. 2021, vol. 11, issue 1, p. 1. ISSN 2045-2322. Available at: https://doi.org/10.1038/s41598-021-01171-z. [cit. 2023-12-17].

[17] Stanford Artificial Intelligence Laboratory et al. Introduction. Online. *Robotic Operating System*. 2018. Available at: https://wiki.ros.org/ROS/Introduction [cit. 2023-12-13].

[18] KORTELAINEN, Matti. *A short guide to ROS 2 Humble Hawksbill*. Online. Finland: School of Computing, University of Eastern Finland, Kuopio, Finland, 2023. Available at: https://erepo.uef.fi/bitstream/handle/123456789/29382/urn_nbn_fi_uef-20230186.pdf?sequence=1. [cit. 2023-12-17].

[19] MACENSKI, Steven; FOOTE, Tully; GERKEY, Brian; LALANCETTE, Chris a WOODALL, William. Robot Operating System 2: Design, architecture, and uses in the wild. Online. *Science Robotics*. 2022, vol. 7, p. 66. ISSN 2470-9476. Available at: https://doi.org/10.1126/scirobotics.abm6074. [cit. 2023-12-13].

[20] ZOLA Andrew. *Strongly typed programming language*. Online. Updated June 2022. TechTarget. Available at: https://www.techtarget.com/whatis/definition/strongly-typed. [cit. 2023-12-02].

[21] VO, Binh a BELLOVIN, Steven. Anonymous Publish-Subscribe Systems. Online. In: *International Conference on Security and Privacy in Communication Networks*. SpringerLink, 2015, p. 195-211. Available at: https://doi.org/10.1007/978-3-319-23829-6_15. [cit. 2023-12-17].

[22] @BotBuilder. *ROS2 Basics #7 - Understanding ROS2 Actions*. Online, video. 20200615 Available at: https://www.youtube.com/watch?v=zocGgf1RJRs. [cit. 2023-12-17].

[23] NGUYEN, Peter. *MICRO-ROS FOR MOBILE ROBOTICS SYSTEMS*. Master thesis. Västerås: Mälardalen University, School of Innovation, Design and Engineering., 2022. Available at: https://www.diva-portal.org/smash/get/diva2:1670378/FULLTEXT01.pdf. [cit. 2023-12-13].

[24] *Features and Architecture*. Online. ROS. 2023. Available at: https://micro.ros.org/docs/overview/features/. [cit. 2023-12-13].

[25] SAXENA, Shivam; FARAG, Hany E.Z. a EL-TAWEEL, Nader. A distributed communication framework for smart Grid control applications based on data distribution service. Online. *Electric Power Systems Research*. 2021, vol. 201, p. 2. ISSN 03787796. Available at: https://doi.org/10.1016/j.epsr.2021.107547. [cit. 2023-12-16].

[26] micro-ROS. *micro-ROS_architecture.png*. Online, Figure. 2021. In: micro.ros.org. Available at: https://micro.ros.org/img/micro-ROS_architecture.png. [cit. 2023-12-15].

[27] FreeRTOS™. *What is An RTOS?*. Online. Freertos. Available at: https://www.freertos.org/about-RTOS.html [cit. 2023-12-17].

[28] Espressif Systems (Shanghai) Co., Ltd. [online datasheet]. *IESP32 WROVER E, ESP32 WROVER IE*. 2023. [cit. 2023-12-17]. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf

[29] Texas Instruments Incorporated [online datasheet]. *I2C Bus Pullup Resistor Calculation*. 2015. [cit. 2023-12-17]. Available at: https://www.ti.com/lit/an/slva689/slva689.pdf?ts=1701864440162&ref_url=https%253A%252F%252Fwww.google.com%252F

[30] Silicon Laboratories Inc. [online datasheet]. *USBXpress™ Family CP2102N Data Sheet*. 2020. [cit. 2023-12-17]. Available at: https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf

[31] Microchip Technology Inc. [online datasheet]. *Introduction to USB Type-C™*. 2015. [cit. 2023-12-17]. Available at: https://ww1.microchip.com/downloads/en/appnotes/00001953a.pdf

[32] IC Plus Corp. [online datasheet]. *Single Port 10/100 MII/RMII/TP/Fiber Fast Ethernet Transceiver* . 2012. [cit. 2023-12-17]. Available at: https://datasheet.lcsc.com/szlcsc/IC-Plus-IP101GR_C79324.pdf

[33] TDK InvenSense Inc. [online datasheet]. *MPU-6000 and MPU-6050 Product Specification*. 2013. [cit. 2023-12-17]. Available at: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

[34] Richtek Technology Corp. [online datasheet]. *RT8259*. 2011. [cit. 2023-12-17]. Available at: https://www.richtek.com/assets/product_file/RT8259/DS8259-03.pdf

[35] Diodes Incorporated [online datasheet]. *AP63200-AP63201-AP63203-AP63205*. 2019. [cit. 2023-12-17]. Available at: https://www.diodes.com/assets/Datasheets/AP63200-AP63201-AP63203-AP63205.pdf

[36] Texas Instruments Inc. [online datasheet]. *TPS212x 2.8-V to 22-V Priority Power MUX with Seamless Switchover*. 2023. [cit. 2023-12-17]. Available at: https://www.ti.com/lit/ds/symlink/tps2120.pdf?ts=1702808818443&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS2120%252Fpart-details%252FTPS2120YFPR

[37] Texas Instruments Incorporated [online datasheet]. *INA3221 Triple-Channel, High-Side Measurement, Shunt and Bus Voltage Monitor with I2C- and SMBUS-Compatible Interface*. 2016. [cit. 2023-12-17]. Available at: https://www.ti.com/lit/ds/symlink/ina3221.pdf?ts=1702827590688&ref_url=https%253A%252F%252Fwww.google.com%252F

[38] Garmin Ltd. [online datasheet]. *D52 ANT SoC Module Series*. 2018. [cit. 2023-12-17]. Available at: https://www.thisisant.com/assets/resources/D00001687_D52_Module_Datasheet.v.2.3_(Garmin).pdf

Symbols and abbreviations

ADC	Analog-to-digital converter
AI	Artificial Intelligence
API	Application Programming Interface
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CPU	Central Processing Unit
DoF	Degrees of Freedom
EMI	Electromagnetic interference
ESD	Electrostatic discharge
FFC	Flexible flat cable
FMCW	Frequency-Modulated Continuous Wave
FOV	Field of view
GPIO	General-purpose input/output
HD	High Definition
IC	Integrated circuit
IDL	Interface Definition Language
IMU	Inertial Measurement Unit
LDO	Low Drop-out voltage
LED	Light-Emitting Diode
MAC	Media access control sublayer
MCU	Microcontroller Unit
MEMS	Micro-electromechanical systems
MII	Media-independent interface
MUX	Multiplexer

OS	Operating system
PC	Personal Computer
PCB	Printed circuit board
PHY	Physical layer
PMU	Power Management Unit
PSRAM	Pseudo static RAM
RAM	Random Access Memory
RMII	Reduced Media-independent interface
RMT	Remote Control Transceiver
rmw	ROS middleware interface
ROM	Read Only Memory
ROS	Robot Operating system
RSS	Radar Safety Sensor
RTOS	Real-time Operating System
SBC	Single Board Computer
SMT	Surface Mount Technology
SoC	System on chip
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
THT	Through-hole technology
TVS	Transient voltage suppression
UART	Universal asynchronous receiver-transmitter
UDP	User datagram protocol
USB	Universal Serial Bus

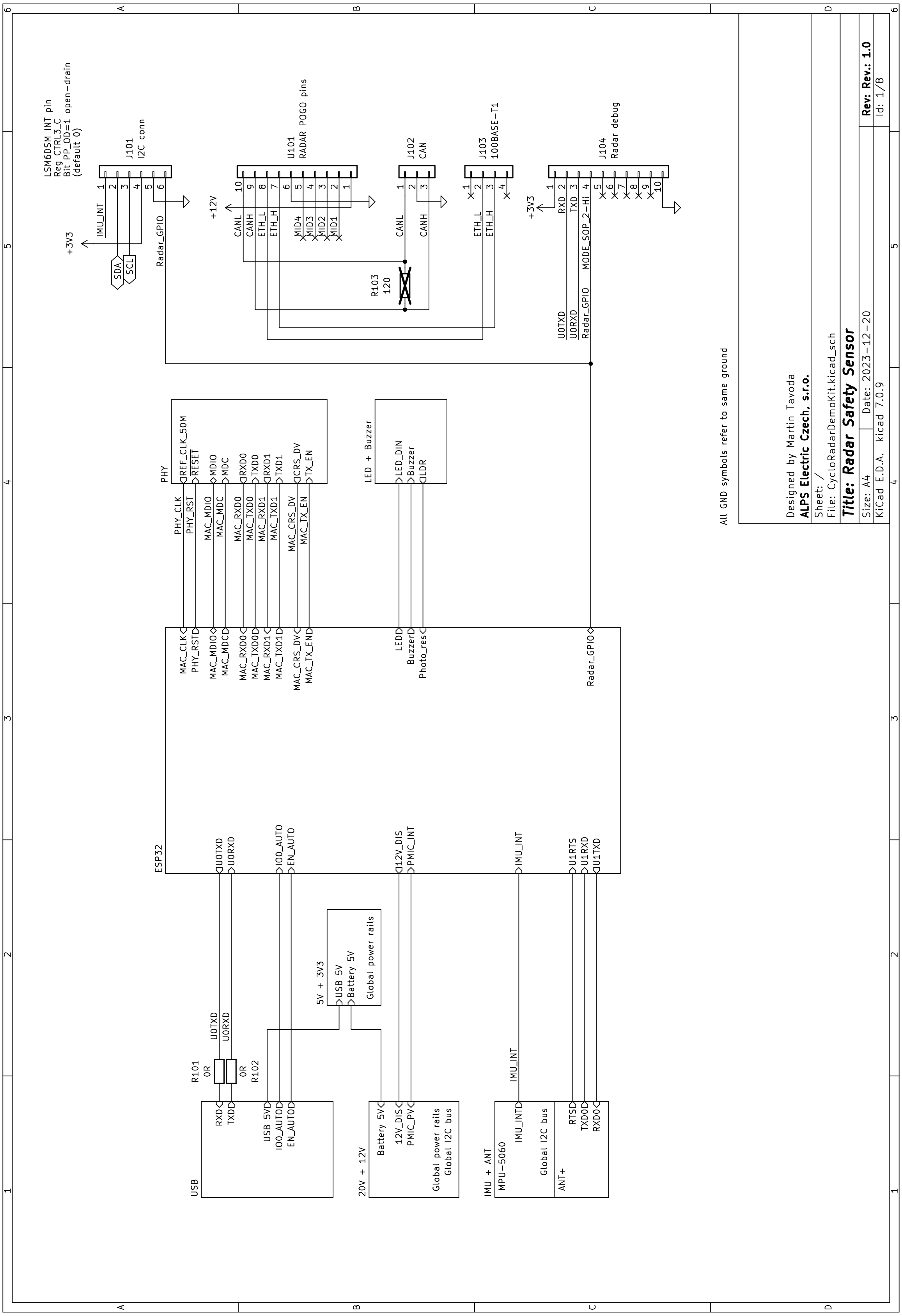
List of appendices

A Schematic diagram **41**

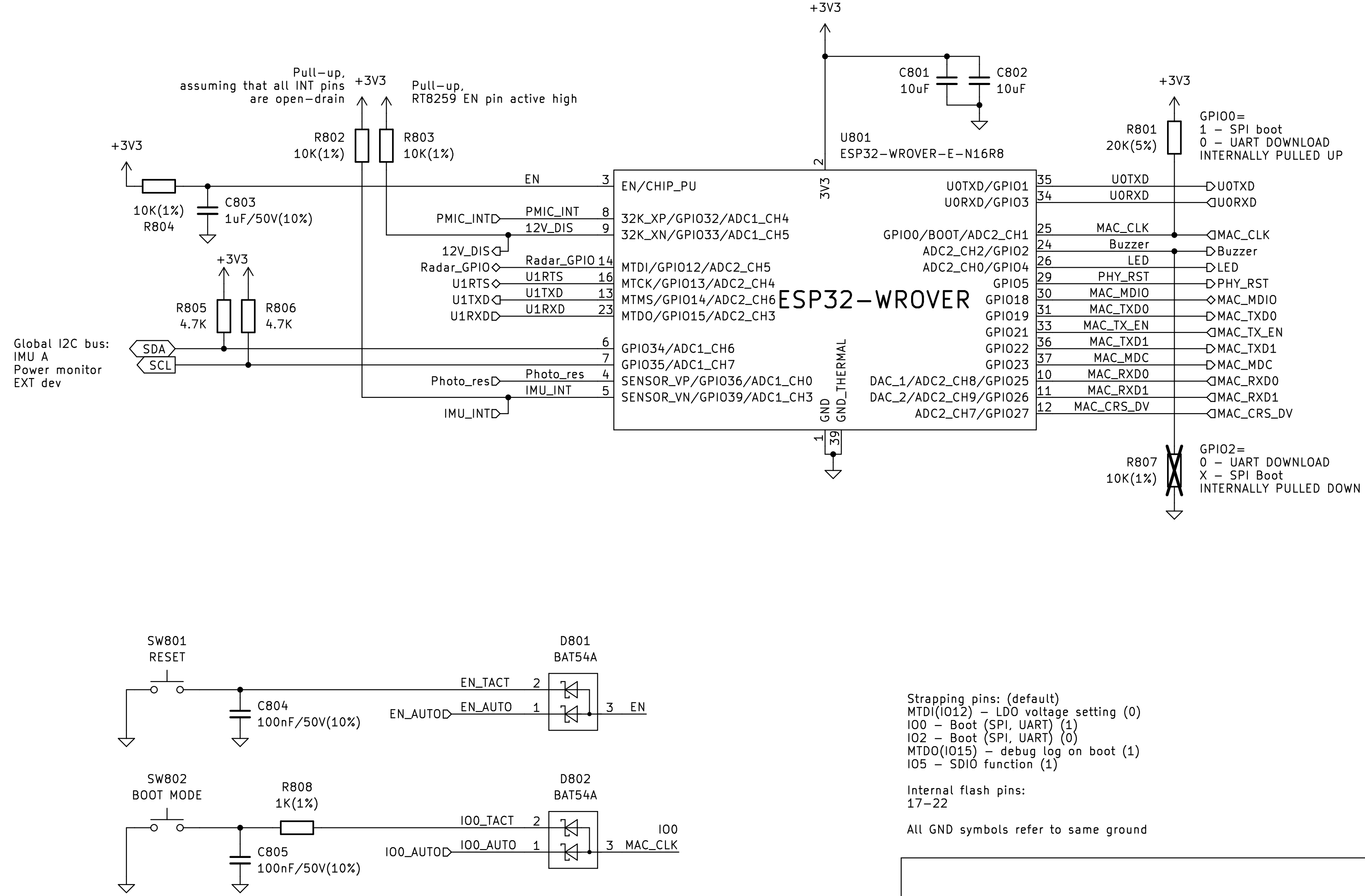
B PCB layout in 3D **49**

C PCB layout **51**

A Schematic diagram



Microcontroller



Strapping pins: (default)
MTDI(IO12) – LDO voltage setting (0)
IO0 – Boot (SPI, UART) (1)
IO2 – Boot (SPI, UART) (0)
MTDO(IO15) – debug log on boot (1)
IO5 – SDIO function (1)

Internal flash pins:
17–22

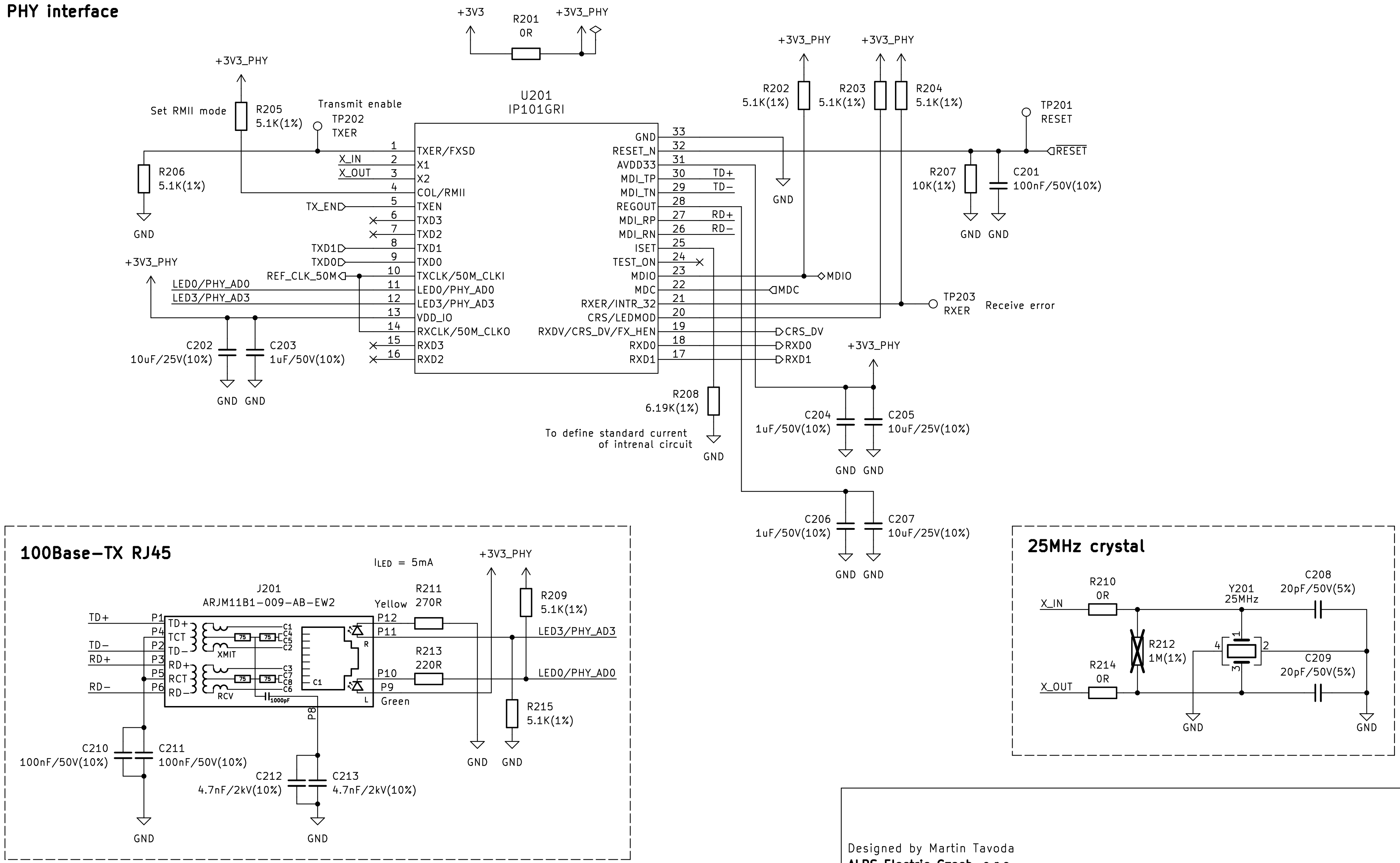
All GND symbols refer to same ground

Designed by Martin Tavoda
ALPS Electric Czech, s.r.o.
Sheet: /ESP32/
File: ESP32.kicad_sch

Title: Radar Safety Sensor

Size: A4 | Date: 2023–12–20 | **Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9 | Id: 2/8

PHY interface

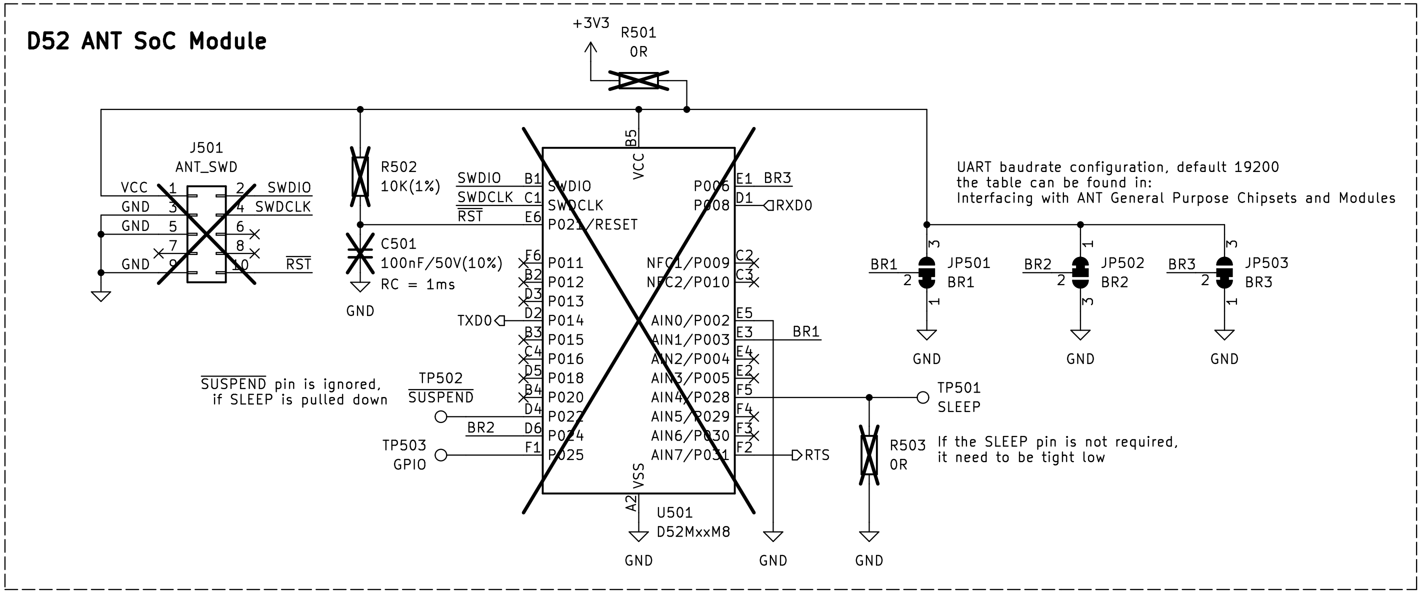


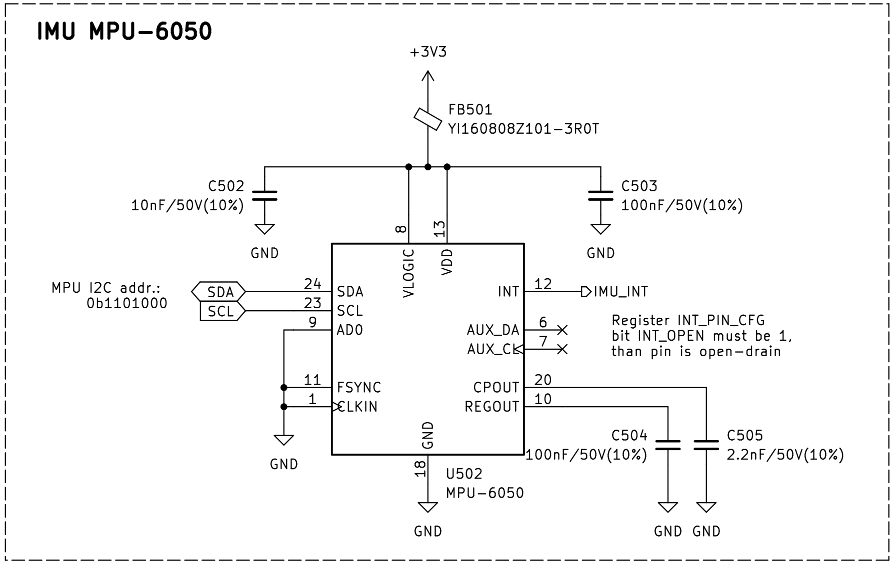
Designed by Martin Tavoda
ALPS Electric Czech, s.r.o.

Sheet: /PHY/
File: PHY.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023-12-20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 3/8





Designed by Martin Tavoda

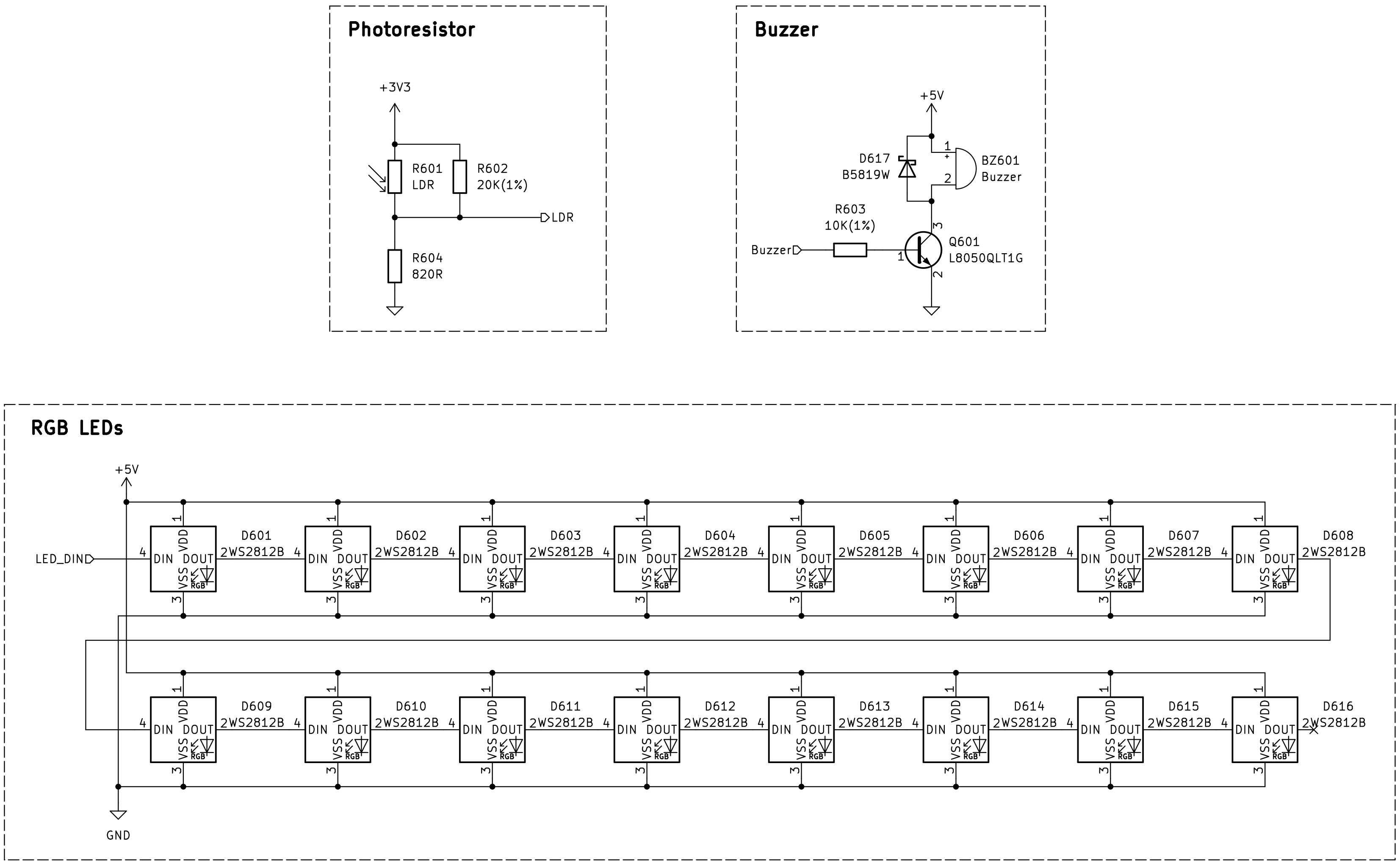
ALPS Electric Czech, s.r.o.

Sheet: /IMU + ANT/

File: IMU&ANT.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023-12-20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 4/8



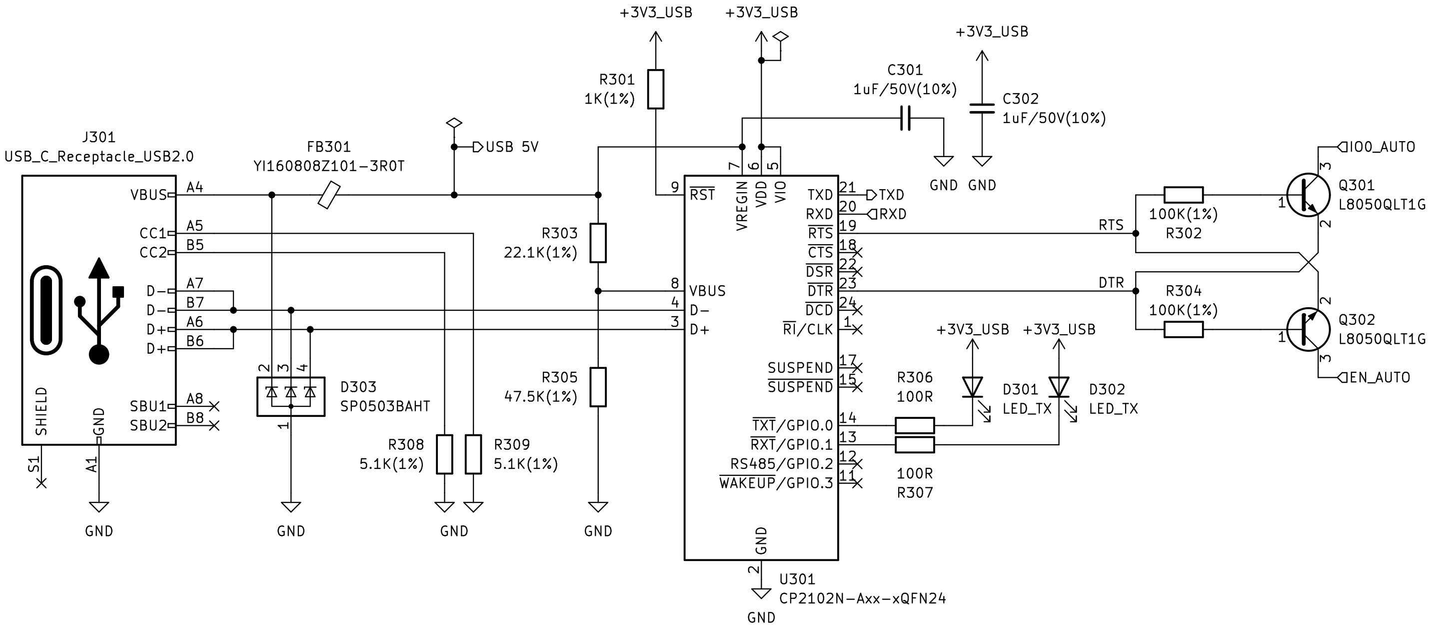
Designed by Martin Tavoda
ALPS Electric Czech, s.r.o.

Sheet: /LED + Buzzer/
File: LEDbuzzer.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023-12-20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 5/8

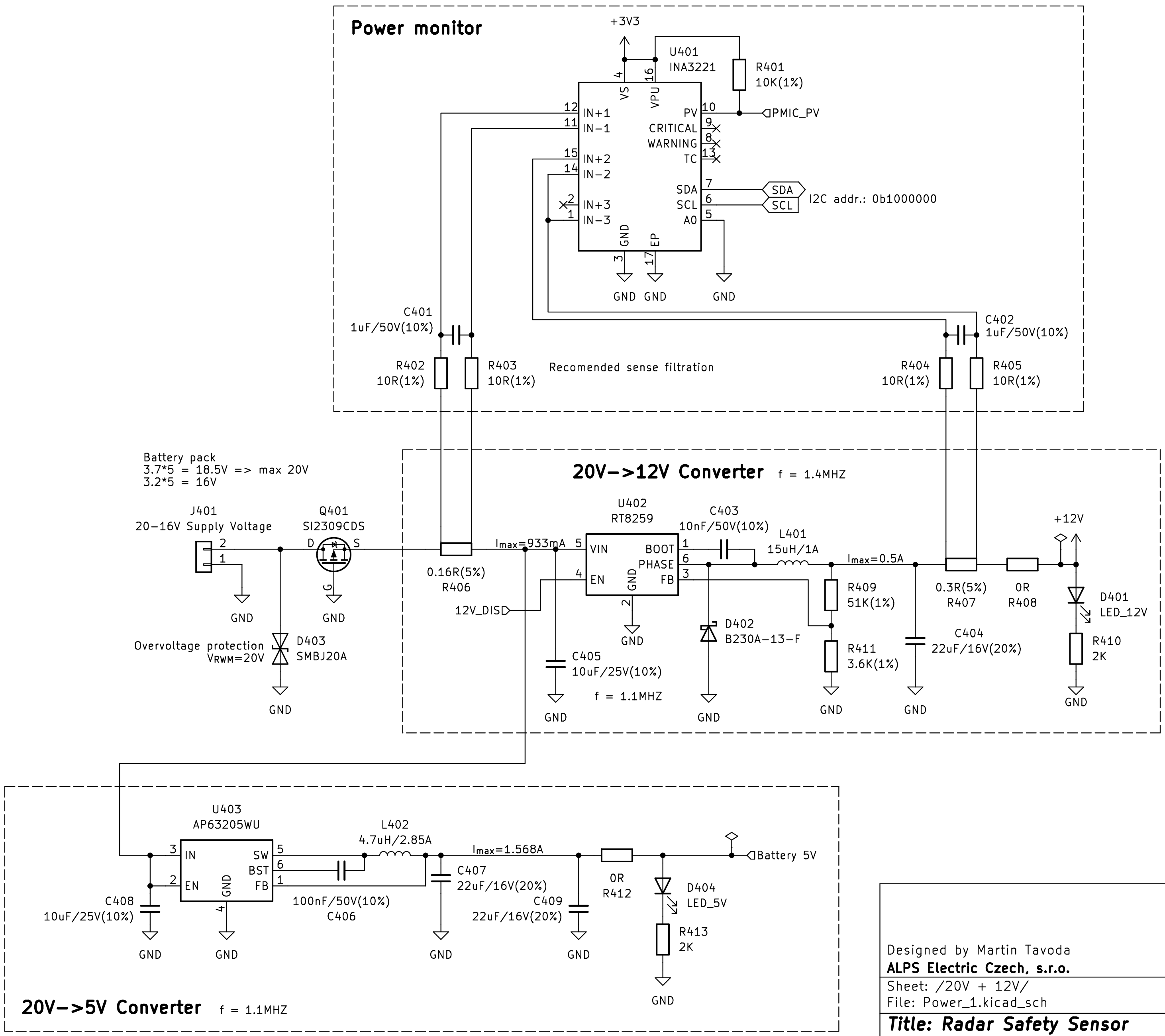
Notes:
+3V3_USB – local net only for CP2102
VBUS – 5V net from USB to power multiplexer



Designed by Martin Tavoda
ALPS Electric Czech, s.r.o.
Sheet: /USB/
File: USB.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023-12-20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 6/8



Designed by Martin Tavoda

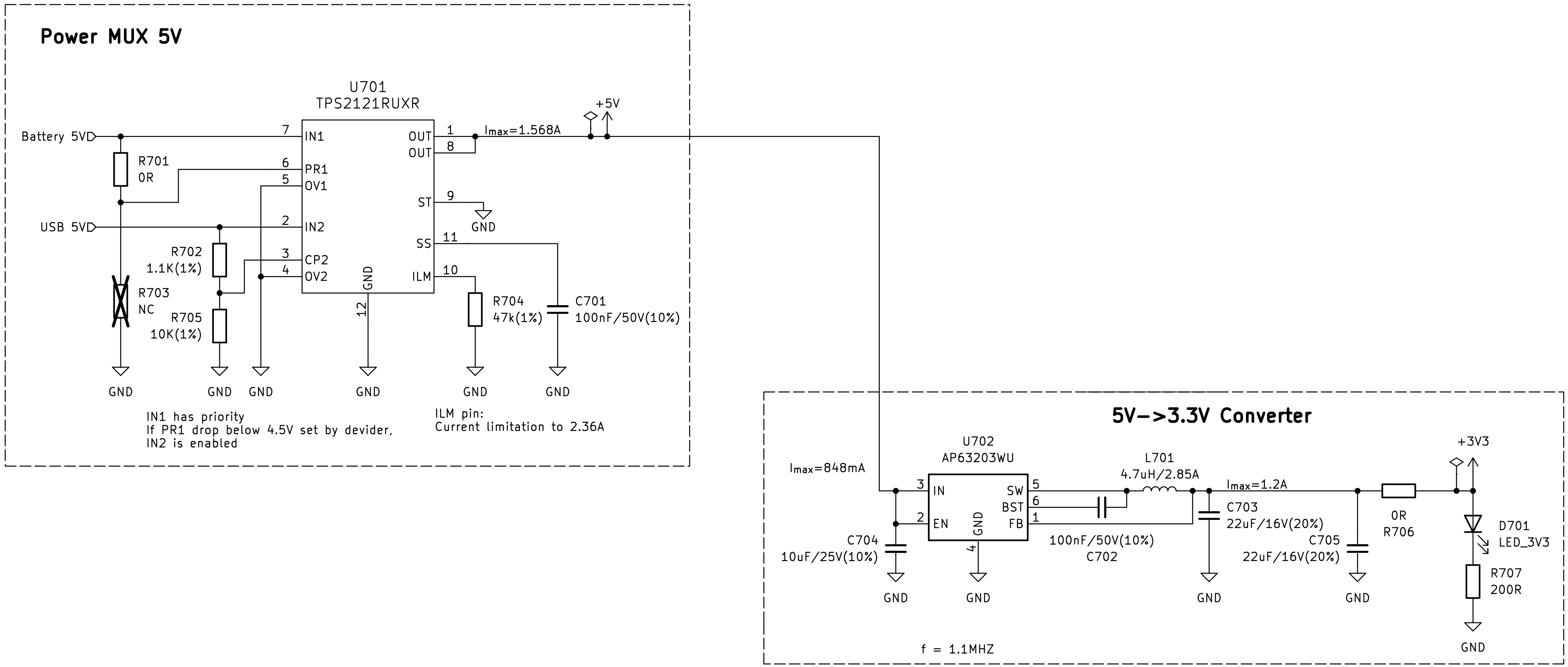
ALPS Electric Czech, s.r.o.

Sheet: /20V + 12V/

File: Power_1.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023−12−20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 7/8



Designed by Martin Tavoda
ALPS Electric Czech, s.r.o.

Sheet: /5V + 3V3/
File: Power_2.kicad_sch

Title: Radar Safety Sensor

Size: A4	Date: 2023-12-20	**Rev: Rev.: 1.0**
KiCad E.D.A. kicad 7.0.9		Id: 8/8

B PCB layout in 3D

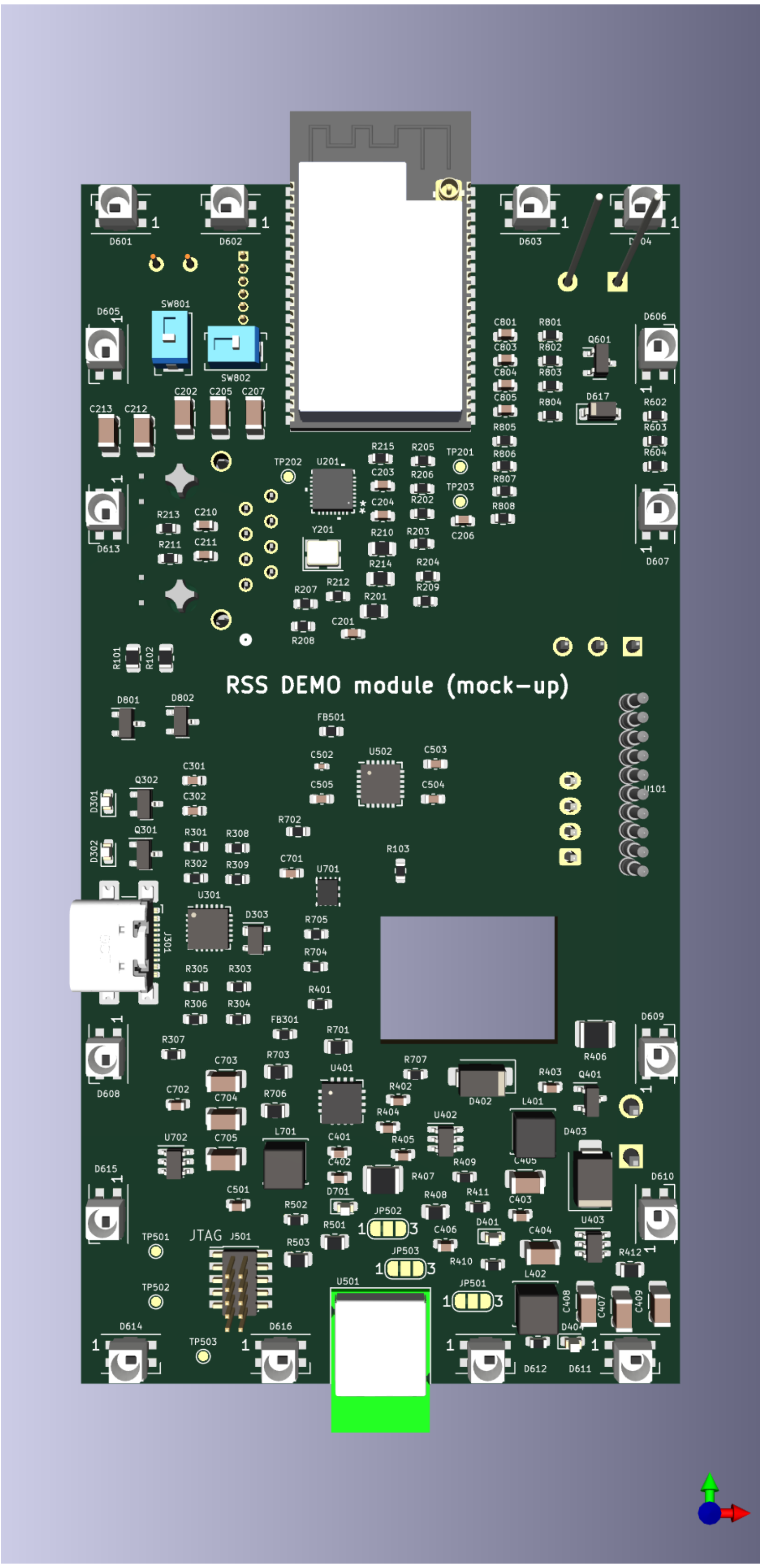


Fig. B.1: PCB layout in 3D, top view

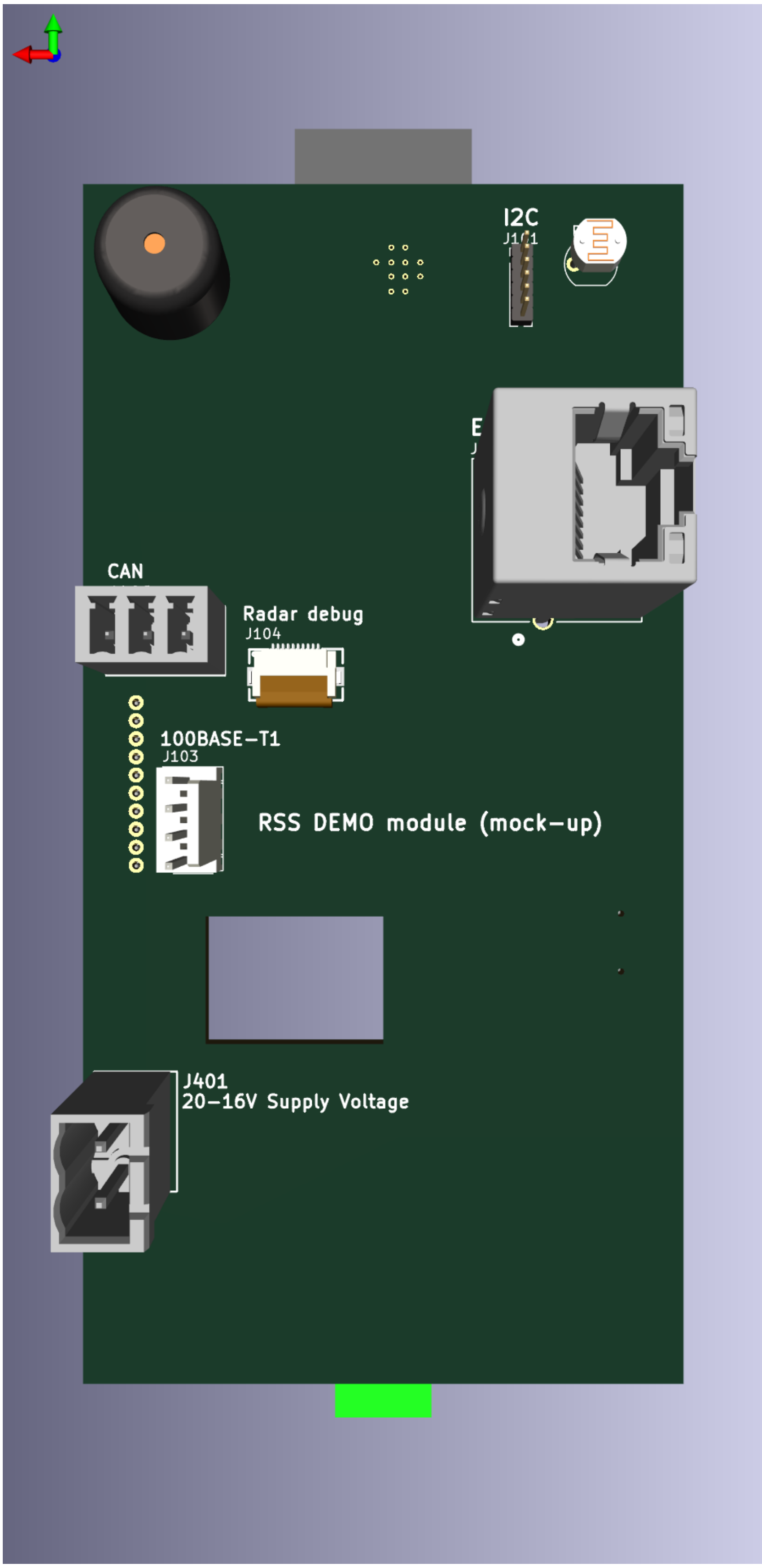


Fig. B.2: PCB layout in 3D, bottom view

C PCB layout

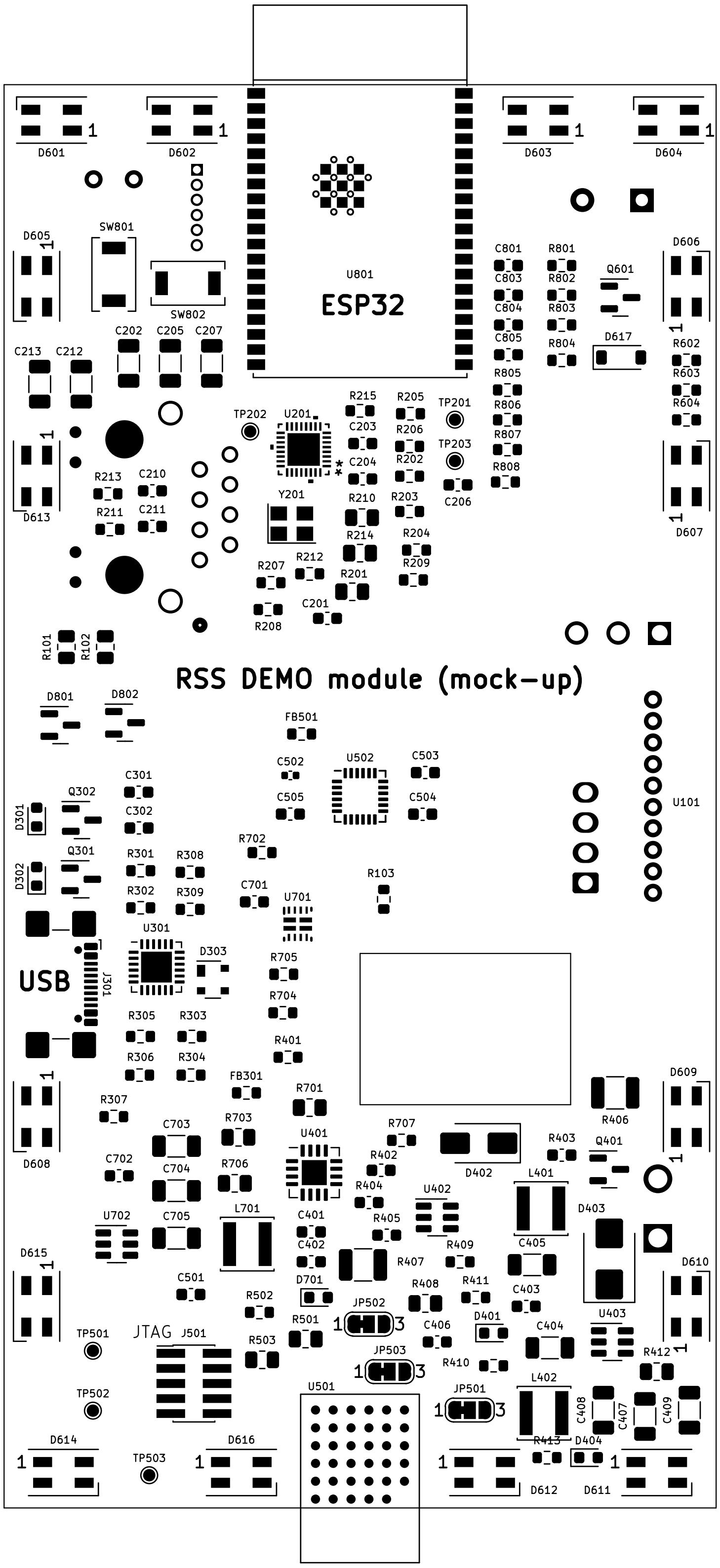


Fig. C.1: PCB layout, top view

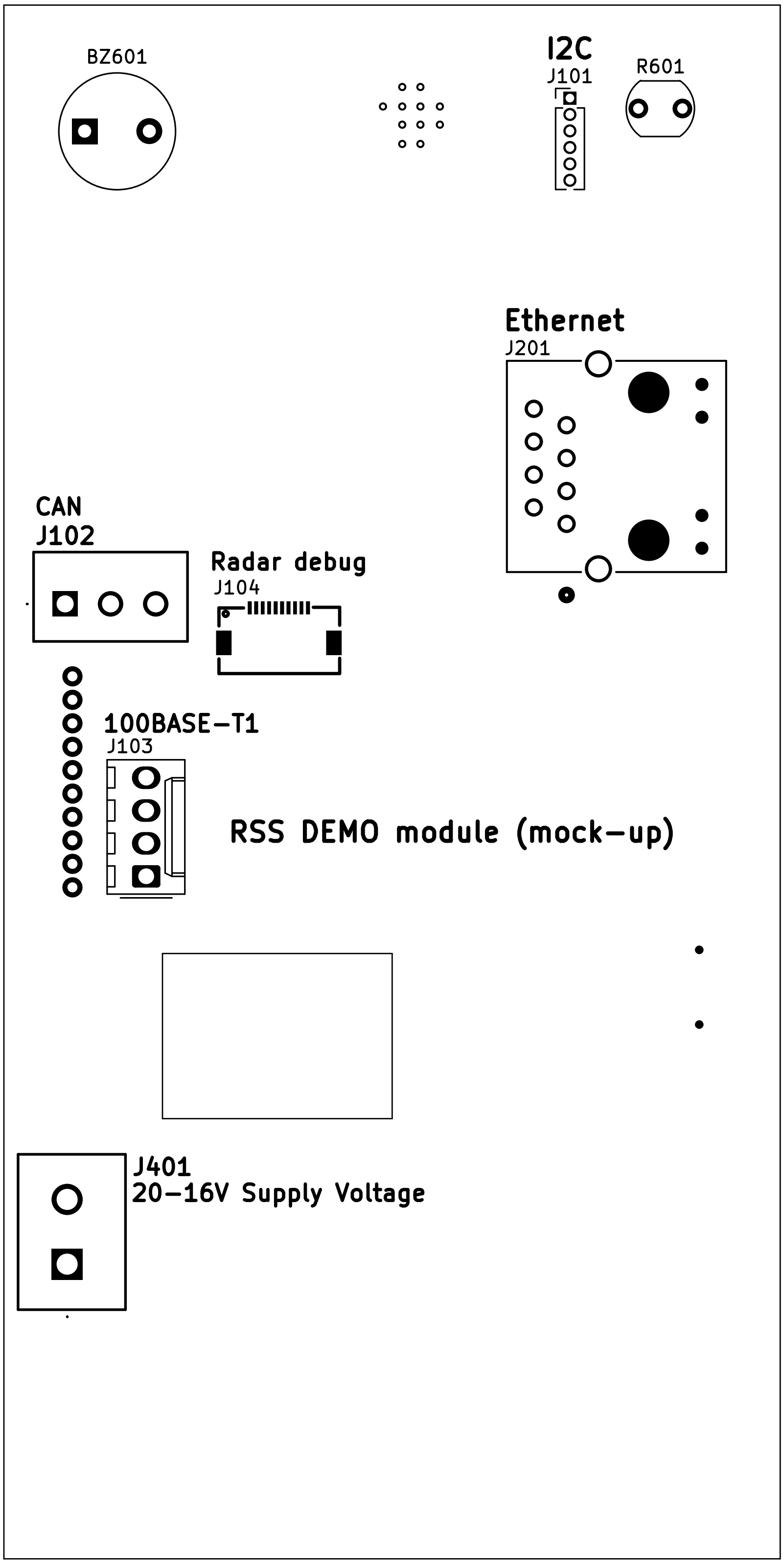


Fig. C.2: PCB layout, bottom view