Infineon Mobile Robot (IMR) system overview

Using IMR demo boards

About this document

Scope and purpose

This document provides a comprehensive functional description and a guide for the CAN bus usage for Infineon Mobile Robot (IMR). Additionally, this document provides the message structure and functionality of IMR.

Intended audience

This document is intended for design engineers, technicians, and developers of electronic systems.

Table of contents

About this document **1**
Table of contents **2**
1 Introduction **3**
1.1 Mobile robot general description 3
1.2 Infineon Mobile Robot (IMR) 3
1.3 Structural boards 4
1.4 Functional blocks 5
1.4.1 Battery management system (BMS) 5
1.4.2 Power distribution unit (PDU) 5
1.4.3 Main control 6
1.4.4 Motor control 6
1.4.5 LED boards 6
1.4.6 Navigation stack (Optional) 7
1.4.7 Ultra-Wideband (UWB) localization (Optional) 7
1.5 Getting started 7
2 Specifications **8**
3 Power distribution and wiring **11**
4 Software **12**
5 CAN bus **13**
5.1 CAN bus related hardware 14
5.2 CAN bus related software 15
5.3 Device Configurator settings 16
5.4 CAN message description by boards 16
5.5 Can messages reserved for future modules 18
5.6 Troubleshooting 20
Revision history **21**
Disclaimer **22**

1 Introduction

1.1 Mobile robot general description

Mobile robots are now common in applications from logistics and warehouse centers and production sites to hospitals, restaurants, and schools, and as last-mile package delivery vehicles. On a high level, mobile robots are of two main types:

- **Automated guided vehicles (AGV):** "Fixed" vehicles that follow predefined paths using lasers, barcodes, radio waves, vision sensors, or magnetic tape for navigation
- **Autonomous mobile robots (AMR):** "Not fixed" and do not need external paths as these use autonomous mapping, localization, navigation, and obstacle avoidance by using sensors

Usually, robots are battery-powered where the voltage level depends on the size and weight characteristics.

1.2 Infineon Mobile Robot (IMR)

The board described in this document is primarily targeted to be used in combination with the Infineon Mobile Robot. IMR is a comprehensive robotic platform for use with a wide variety of boards, such as sensors, motor controls, wireless communication, and battery management system (BMS).

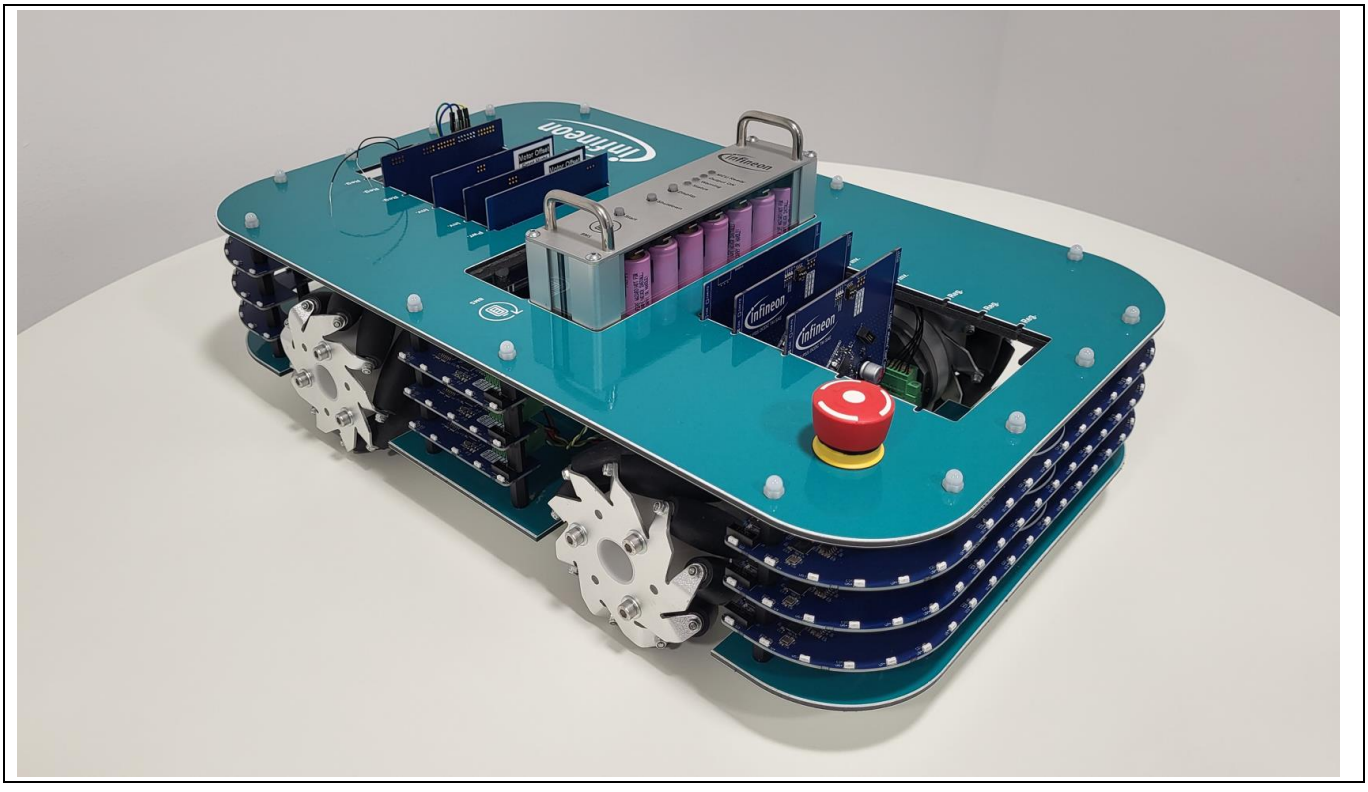


Figure 1 Infineon Mobile Robot (IMR): Isometric view

The IMR is intended to provide a demonstration platform for autonomous service robot functionalities by using Infineon components. This introduces the main structure, functional blocks, and CAN bus as central communication bus. Figure 2 shows the block diagram of the IMR.

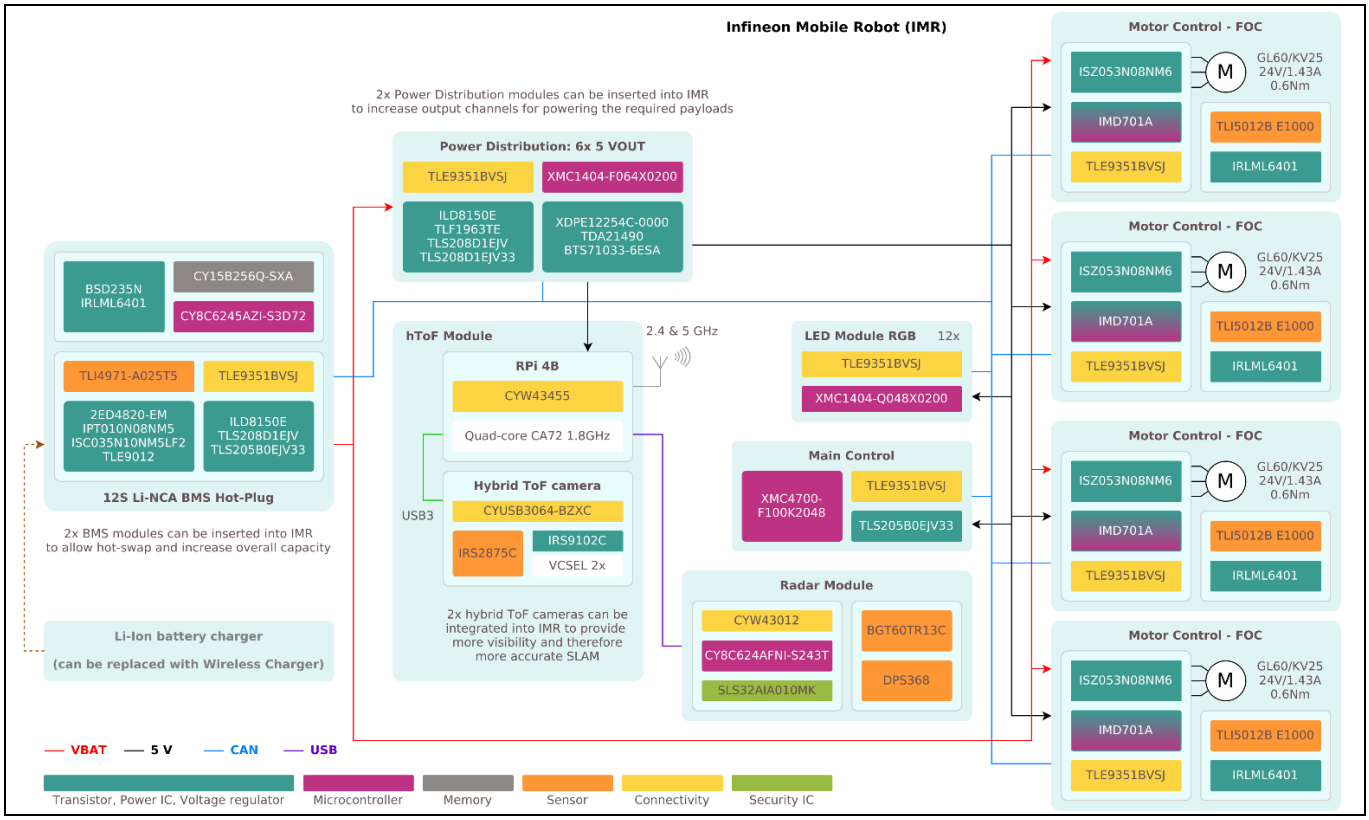


Figure 2 IMR block diagram

1.3 Structural boards

The individual modules and functional blocks are connected to the rest of the robot by structural boards. These structural boards serve the following purposes:

- Routing of power
- Routing of communication channels
- Mechanical integrity

Figure 3 shows the robot without submodules and top-plate, revealing the structural boards. For more information about the structural boards, see the Structural boards user guide, including the pin map of the card slots.

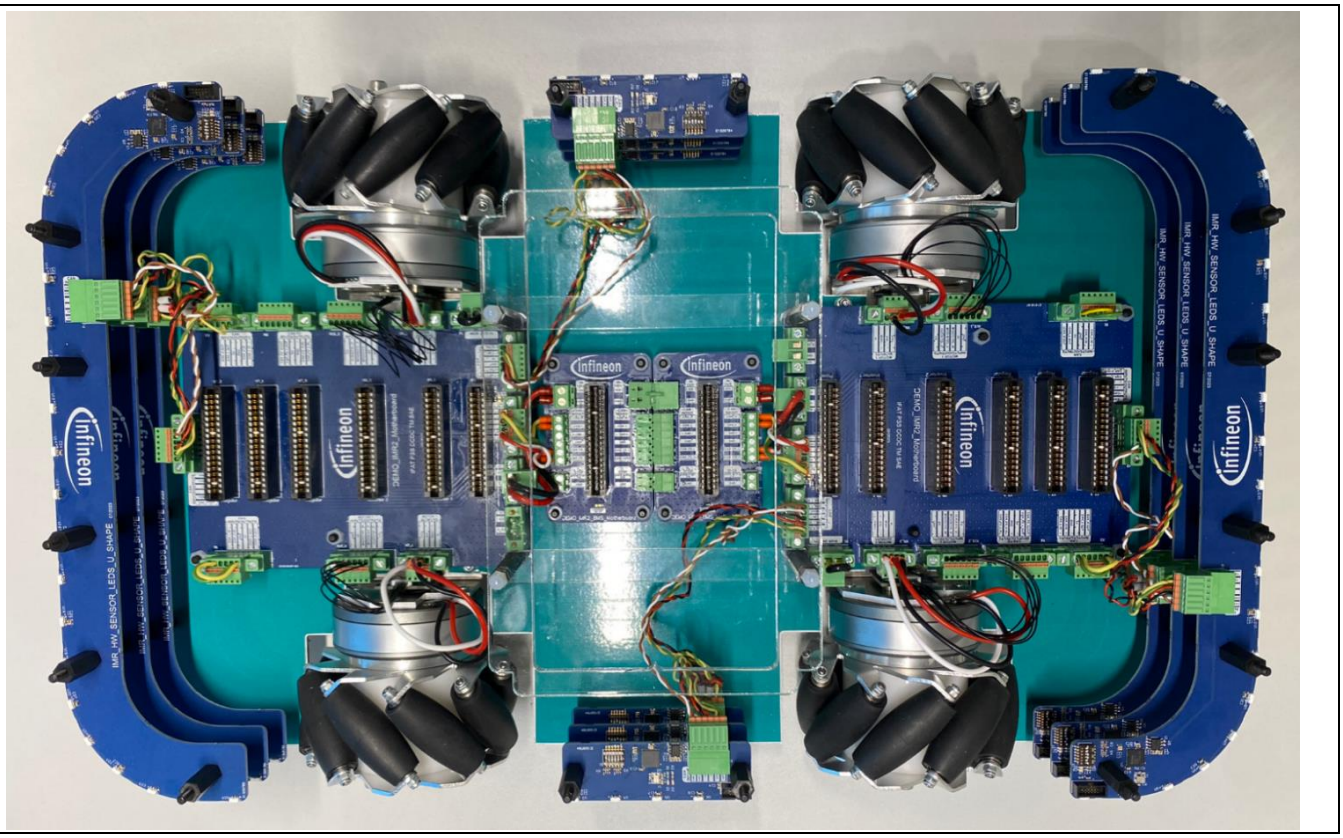


Figure 3 Structural boards of the robot without submodules and top-plate

1.4 Functional blocks

1.4.1 Battery management system (BMS)

The BMS is the primary source of power for the robot. Up to two BMS can be placed in the center of the robot, their nominal voltage is 48 V. Each BMS consists of two parts: BMS control and BMS power. To know more about IMR BMS, see IMR battery management control and IMR battery management system power user guide.

1.4.2 Power distribution unit (PDU)

The PDU converts the battery voltage to 5 V and supplies the submodules. Each side of the robot (front/back) has its own PDU. Most IMRs are assembled with a preliminary power unit that features a simple 48 V/5 V DC-DC converter, see Figure 4, where all 5 V output channels are connected to the single output of the converter.

An advanced PDU-based on multiphase buck converter is available, offering switchable outputs along with the lower output voltage and current ripples, higher efficiency and better transient performance are desired. This PDU is currently on-request but the PCB design files and its user guide are publicly available.

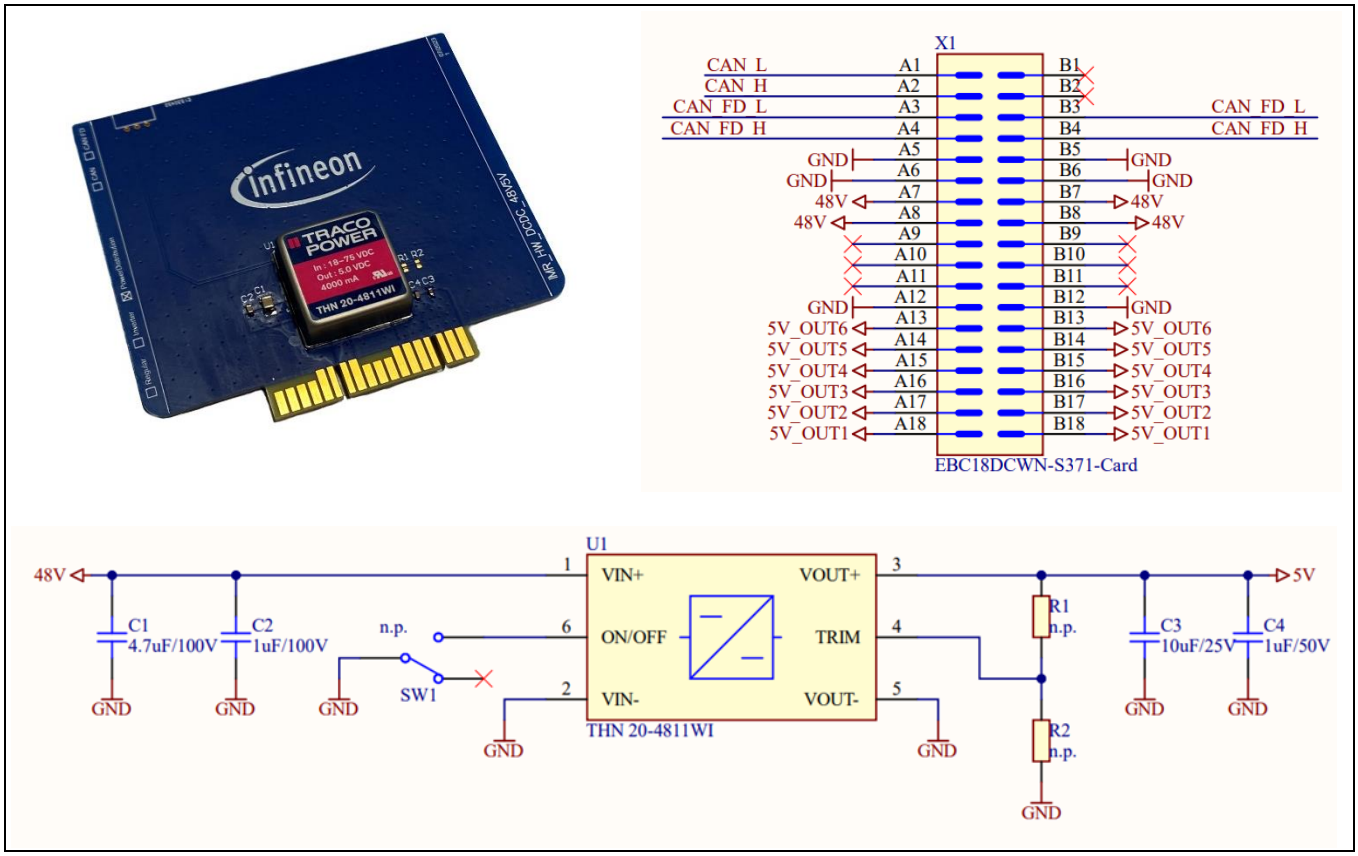


Figure 4 Preliminary power distribution unit

1.4.3 Main control

The main control is based on a XMC4700 MCU and it is the central control hub for the robot. Most communication goes to and from the main control via CAN bus. Each IMR is equipped with one main control board. To know more about this board, see IMR main control user guide.

1.4.4 Motor control

Two generations of motor control solutions for the IMR are available:

- Single motor control based on IMD701, see IMR motor control user guide
- Dual motor control based on PSOC™ Control C3P, see REF_48V_2X1KW_ASFOC dual-motor drive for robotics applications user guide

Both solutions allow for precise individual control of each motor.

1.4.5 LED boards

The LED boards come in two variants (short and long) and are placed along the perimeter of the robot. They are controlled via the CAN bus and serve as a feedback channel to the user. Details are described in the IMR LED functionalities user guide in addition to their PCB design files for the short board and the long board.

1.4.6 Navigation stack (Optional)

The navigation system is based on hybrid time-of-flight (hToF) cameras and features a Jetson Orin NX as the processing unit. It enables the robot to map its surroundings, navigate along a path and avoid obstacles. For details, see the Navigation Stack User Guide (available in the future release).

1.4.7 Ultra-Wideband (UWB) localization (Optional)

A localization system based on UWB is available for the IMR. For details, see the UWB User Guide (available in the future release).

1.5 Getting started

For first-time users of a fully assembled IMR, a User Guide including unboxing instructions is available. The Infineon Mobile Robot assembly information user guide describes the full step-by-step assembly, from the bill of materials to flashing software onto the individual boards.

2 Specifications

The specifications and general information on the IMR are shown in Table 1.

Table 1 IMR specifications

General information	
Type	Autonomous Mobile Robot (AMR)/Remote controlled vehicle
Designated use	Tradeshow demonstration and testbed for new subsystems
Electronics board design	Modular insertion into card-edge connector
Dimension	
Length x Width	500 x 300 mm
Height	116.4 mm (from ground to top plate) 142.1 mm (from ground to emergency stop) 204.3 mm (from ground to battery module handle)
Mass	5 kg (without battery module) 6.2 kg (with 1x battery module) – default setup 7.4 kg (with 2x battery modules)
Clearance from ground	25.6 mm
Driving	
Number of driving wheel	4
Wheel type	Mecanum wheel
Wheel diameter and thickness	100 mm and 50 mm
Maximum speed	1 m/sec (software limit in motor control) 3.2 m/sec (motor limit) at 7.5 kg total mass
Acceleration	≤ 2 m/sec2 (motor limit) at 7.5 kg total mass
Maximum inclination	0° (test environment), 3° (theoretical, motor limit)
Payload	
Theoretical payload	Up to 1.3 kg (motor limit and for IMR with 1x battery module) *Note: IMR structure is not actually designed to carry payload*
Payload placement	Spread uniformly around the perimeter of the top surface *Note: IMR top plate design is not actually designed to carry payload, additional structure needs to be added to allow secure payload placement*
Subsystem	
Main control	Based on XMC4700 Arm® Cortex®-M4 Available interfaces (not in the card-edge bus): SPI, UART, I2C, ADC, PWM, GPIO, SBUS

Specifications

Motor control	Based on IMD701A Arm® Cortex®-M0 and 80 V OptiMOS™ 6 FOC algorithm with position sensor feedback from TLI5012B E1000
Battery Management (BMS)	Based on PSOC™ 62 dual-core Arm® Cortex®-M4 and Arm® Cortex®-M0+ 12 cells in series managed by TLE9012DQU Charging and discharging circuit using 2ED4820-EM and OptiMOS™ 5 Current sensing with TLI4971 25 A Non-volatile memory 256 kbit F-RAM CY15B256Q-SXA
Power distributor	Based on voltage converter THN 20-4811WI Input: 18 V – 75 V Output: 5 V and up to 4 A
Computing module (optional)	Optional: NVIDIA Jetson Orin NX 8 GB Voltage converter TEN60-4812N (Input 36 V – 75 V, Output 12 V and up to 5 A)
Navigation sensor (optional)	
3D camera	Two pieces of hybrid Time-of-Flight (hToF) camera: front and left side of IMR Within a camera: Spot illumination for mapping and localization: 100° x 21° field of view, 0.1 m – 8 m range, ~ 600 spots Flood illumination for obstacle avoidance: 100° x 45° field of view, 0.05 m – 2 m range, 20000 spots
Inertial Measurement Unit (IMU)	3-axis gyroscope and 3-axis accelerometer based on LSM6DS3 Output data rate: 104 Hz Gyroscope: ± 125 deg/sec full scale range Accelerometer: ± 4 g full scale range
Power	
Battery type	18650 form factor Li-ion (e.g., LNO Samsung INR18650-35E)
Battery voltage	43.2 V nominal (30 V – 50.4 V) – 12 cells in series
Battery capacity	146.88 Wh (3.4 Ah at 43.2 V) per battery module *Note: Currently only one battery module can be active at a time*
Hot-swap	Enabled
Charging	
Input voltage	90 ~ 264 VAC 47 ~ 63Hz
Charging voltage	50.4 V
Charging current	Up to 1.7 A
Safety	
Features	Overcurrent protection: Current limit at +6 A and -2.5 A (triggers BMS inactive) Emergency stop: Send signal to BMS to enter SAFE_STATE (triggers BMS inactive)
Connectivity	

Specifications

Communication bus	CAN based on TLE9351BVSJ 1 Mbps is implemented Optionally, 5 Mbps CAN FD (CAN protocol needs to be adjusted for this)
Remote control	2.4 GHz wireless and SBUS protocol for remote controlled vehicle 2.4 GHz wireless nano receiver via USB (on NVIDIA Jetson Orin NX) for AMR
Wi-Fi (optional)	Triple-band 2.4 GHz, 5 GHz, 6 GHz Wi-Fi module based on CYW55573 Dual-band 2.4 GHz (disabled) and 5 GHz router Protocol: Wi-Fi 6 IEEE 802.11ax/ac/n/a
Environment	
Usage	For indoor only
Ingress Protection (IP) class	None
Floor conditions	No dirt, no water, no oil
Temperature & humidity	20°C to 30°C non-condensing humidity
User station (optional)	
Laptop	16 GB RAM, 512 GB SSD, Wi-Fi 6, touchscreen, Ubuntu OS with ROS Noetic

3 Power distribution and wiring

The IMR has two separate power rails, see Figure 5.The Vbat rail distributes the battery voltage (nominal 48 V) to the motors, each of the regular slots and the PDUs. The 5 V rail supplies regular slots as well as the angle sensor boards in the motors and LED boards.

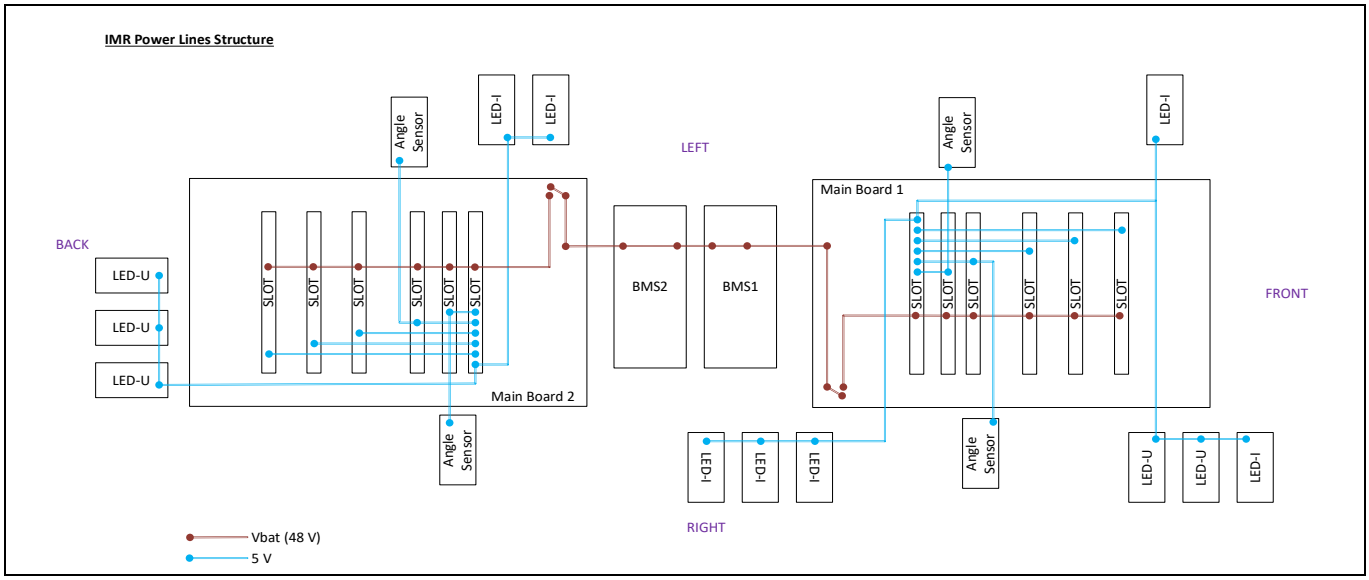


Figure 5 Power distribution wiring

4 Software

The overall software landing page of IMR and other boards related to robotics application can be accessed here: Software collection for Infineon Robotics Development Platform. Specifically, for the IMR subsystems, the following repositories are available to enable the first try of the hardware for evaluation purpose:

- IMR battery management system based on PSOC™ 62: IMR Software for PSOC™ 6 Battery Management
- IMR main / robot control based on XMC™ 4700: IMR Software for XMC4700 Main / Robot Control
- IMR motor control based on MOTIX™ IMD701A: IMR Software for IMD701A Motor Control
- IMR LED control based on XMC™ 1404: IMR Software for XMC1404 LED Control

5 CAN bus

The CAN bus serves as the internal communication bus for the robot. With the exception of high bandwidth applications (e.g., camera data transfer), all data transfer between the robot's subsystems uses the CAN bus. Table 2 shows the basic parameters of the bus.

Table 2 CAN bus parameters

Parameter	Value
Mode	Standard (non FD)
ID format	Standard (11 bit)
Bitrate	1 Mbit/s

The 11-bit identifier indicates the message type. A lower identifier also grants a message a higher priority, as defined by the CAN bus arbitration. Within the space of the available 11 bits, reserved ID ranges were defined, see Table 3.

Note that these ranges are only partially occupied by the current implementation. If new components are added to the system, you should choose a non-occupied range with appropriate priority. Figure 6 shows the entire 11-bit ID space with the reserved portions marked in magenta.

Table 3 Reserved ID ranges

Group	Range (in hex)
Emergency commands	80 – 87
BMS State	100 – 10F
BMS commands	180 – 1BF
PD state / output current	280 – 297
PD commands	300 – 307
Speed command	380 – 387
Motor encoder reading	400 – 407
Velocity command	41C – 41F
Odometry estimate	440 – 443
IMU data	460 – 463
UWB	470 – 473
LED	480 – 49F
Calibration	540 – 547
Board information	600 – 615

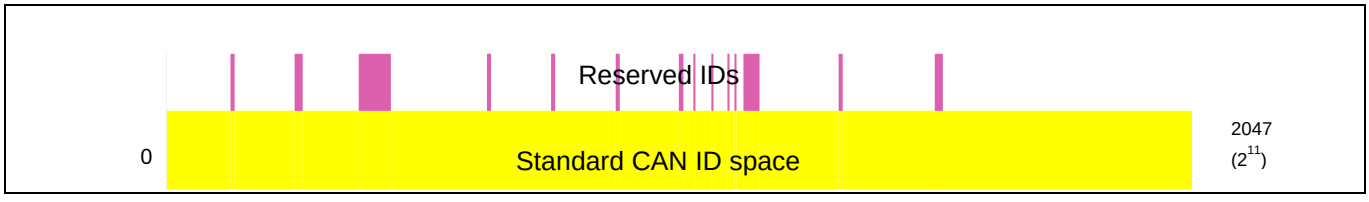


Figure 6 Visualization of reserved CAN IDs (magenta)

5.1 CAN bus related hardware

Figure 7 shows the electric routing of the CAN bus on the IMR. Sections implemented as PCB traces are shown in orange, cables and connectors are shown in blue. Note that differential CAN high and CAN low lines are not shown separately. At each end of the bus, a 120 Ω terminal resistor is placed between CAN high and CAN low on the main boards. The connected subsystems do not carry terminal resistors. Not shown is the separate CAN FD bus that runs largely in parallel to the CAN bus, but it is not utilized for the current implementation.

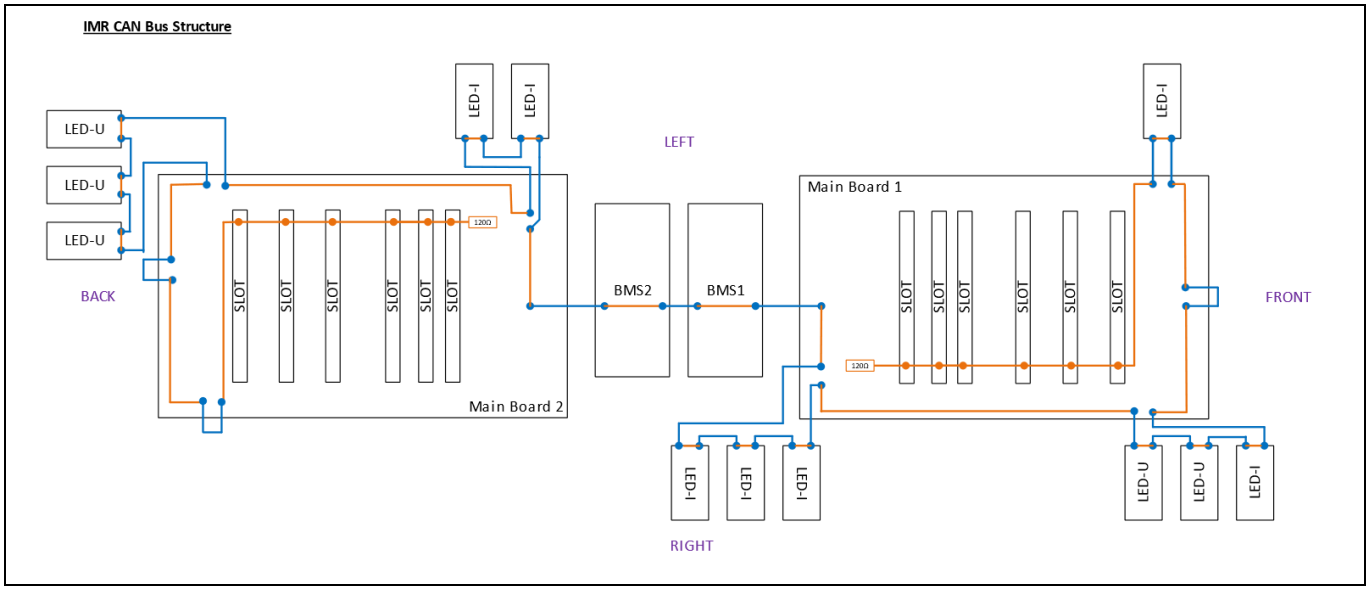


Figure 7 CAN bus routing

Figure 8 shows the schematic for the CAN transceiver. This circuit (or very similar) can be found on all components connected to the CAN bus. In the case of the Jetson, it is part of the carrier board. In most cases, a TLE9351BVSJ is used. This translates the unidirectional TX and RX UART signals from the microcontroller into the differential signal for the CAN bus. The supply voltage (VCC) of this transceiver is 5 V, but the V_IO pin can be either 5 V or 3.3 V, depending on the requirements of the connected microcontroller.

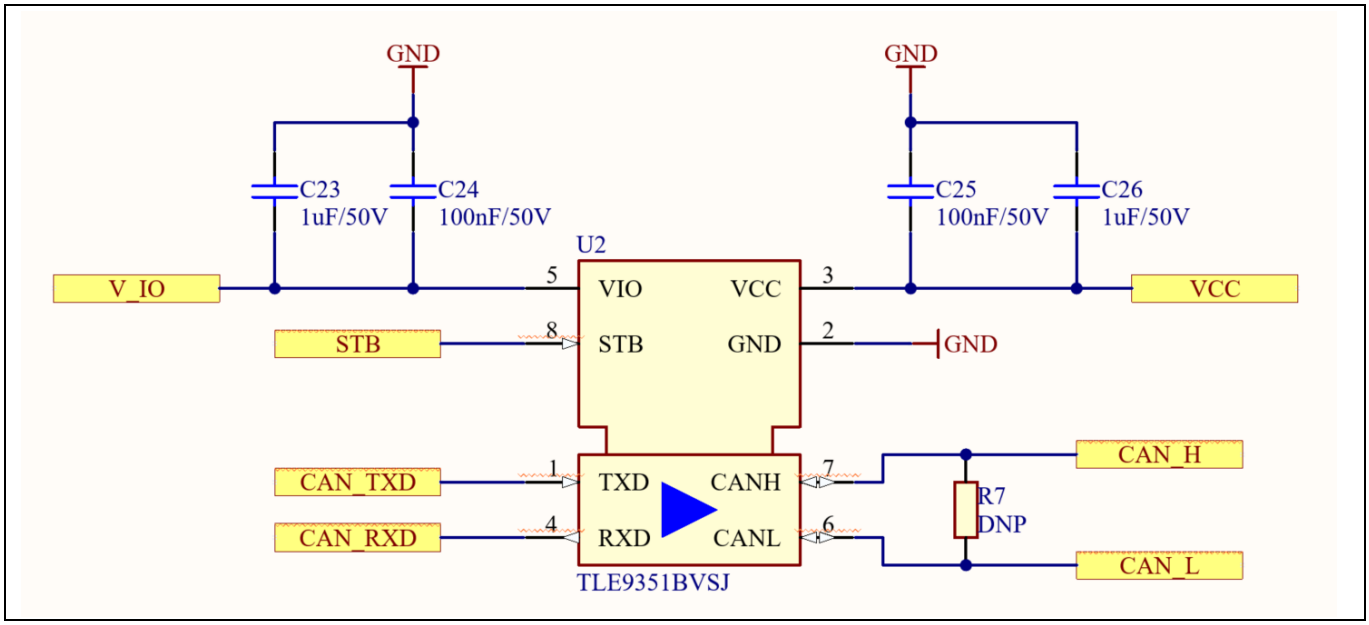


Figure 8 CAN transceiver circuit

5.2 CAN bus related software

Every type of board that connects to the CAN bus has its individual set of functions related to its interaction with the CAN bus. An effort has been made to structure these sections of code to be structured in a similar manner, while still considering the individual function and underlying controller architecture. Generally, the CAN implementation consists of several files:

- IMR_CAN_GLOBAL.h

  This file contains several typedefs and enums:

  - IMR_CAN_MESSAGE_STRUCT
  - IMR_CAN_MESSAGE_IDS
  - CAN_STATUS_t
  - IMR_CAN_SENSOR_LED_COMMANDS

  The universally available “CAN_TX_Request(…)” is also declared here.

  This file is intended to be kept identical between all participants of the CAN bus. If new functionality requires the definition of additional message IDs, these changes should be propagated not only to the participants that will send or receive using these message IDs, but all boards, in order to ensure consistency.

- IMR_CAN.h

  This file contains the board specific definitions/declarations. In the case of XMC boards, the logical message objects (LMOs) and their corresponding interrupt handlers are defined here.

- IMR_CAN.c

  This file contains the board specific CAN implementation including message/interrupt handlers and initialization methods.

- For XMC based components: IMR_CAN_XMC_shared.c
  Contains implementation for 2 methods, “XMC_CAN_MO_Busy(…)” and “CAN_TX_Request(…)”

In order to add a new message, its corresponding ID must be added to IMR_CAN_MESSAGE_IDS in IMR_CAN_GLOBAL.h. Now, the new message can be transmitted using CAN_TX_REQUEST(…) and message handling has to be added in all receiving components. The message handling methods will generally be located in IMR_CAN.c.

5.3 Device Configurator settings

1. In **ModusToolbox™** > **Quick Panel** > **BSP Configurators** > **Device Configurator**, the settings for underlying libraries are made
2. The CAN driver can be activated in **Peripherals** > **Communication** > **Controller Area Network**
   Make sure the **Nominal Bitrate** is correct (1000 kbps) and the **Nominal Synchronization Jump Width** is set to the maximum possible value

The latter ensures that even if the participants have some tolerance in their timing, messages can still be received successfully. Filter elements for filtering CAN IDs can be set here, the details differ between PSOC™ and XMC™ microcontrollers.

5.4 CAN message description by boards

While messages are broadcast onto the CAN-bus and, in theory can be received by any component, most messages have a very specific use between just between two components. Some components, such as motor control and LED boards can exist multiple times in the IMR. In this case, the boards are identified by their dip switch setting (see User Guides of the specific boards). Therefore, every board receives unique message IDs. The message ID fully describes the type of the message, allowing the specific recipient to recognize it while keeping the sent data free of metadata.

Some messages are sent periodically to provide constant status information, other messages are sent only on demand. A graph showcasing the most important interactions between different components in the IMR is shown in Figure 9.

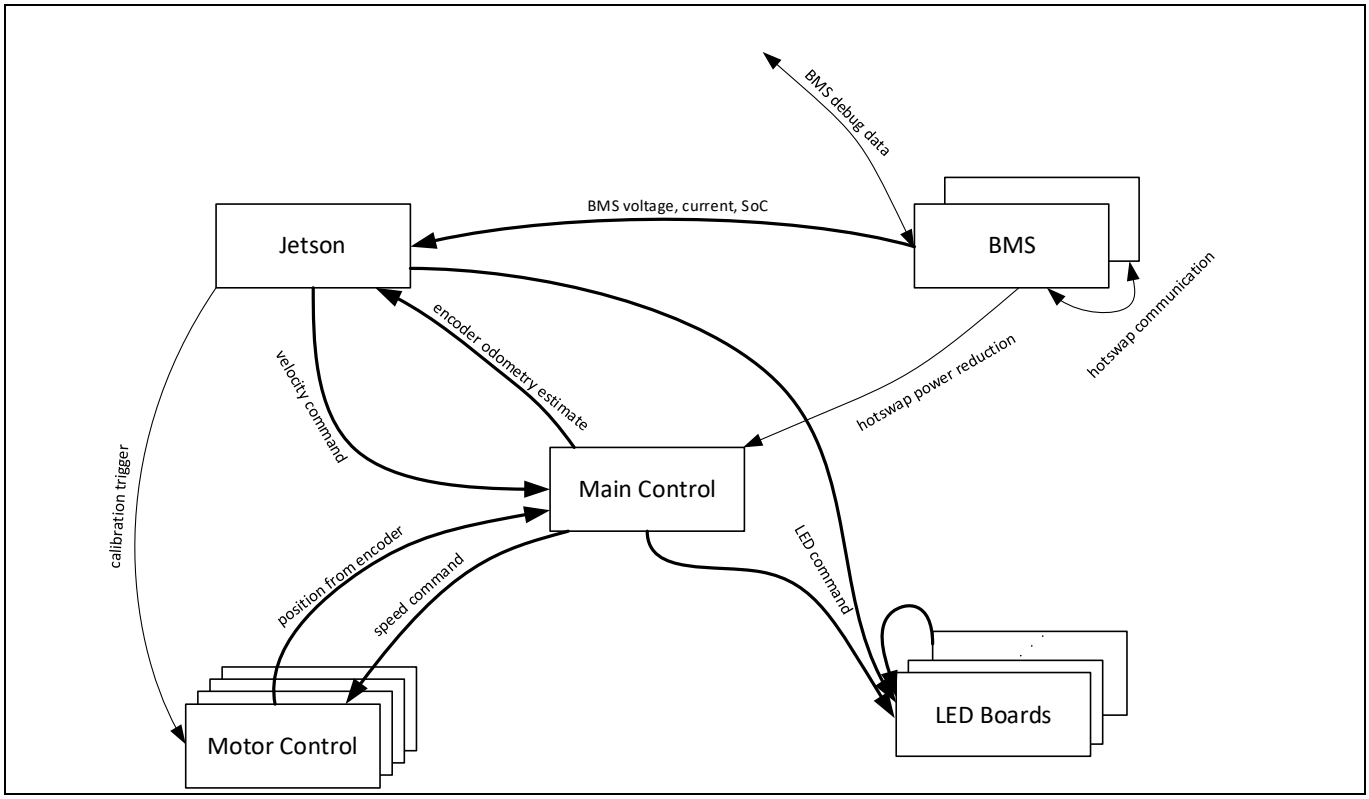


Figure 9 Flowchart for CAN messages; bold arrows: messages sent with fixed rates during operation; thin arrows: messages triggered by user or special event

The individual CAN messages with their IDs are listed in the respective board User Guides, see Section 1.4. A DBC (CAN Database) file *IMR CAN.dbc* is available to support the automated interpretation of messages on the bus.

5.5 Can messages reserved for future modules

Table 4 contains message definitions that are not yet part of the code base but will likely become so in the future. The IDs should not be used for any other purposes.

Table 4 CAN Messages reserved for future modules

		Byte #									
Command	**ID**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Description**	**Rate**
Emergency_Stop	0x80	–	–	–	–	–	–	–	–	Set motor speed to 0 and shut off BMS	On demand
PD_CH1TO3_ OUTPUT_CURRENT (front|back)	0x280 |0x281	Output Current CH1 U, mA		Output Current CH2 U, mA		Output Current CH3 U, mA		–		Output current of each channel from 1 to 3	1 Hz
PD_CH4TO6_ OUTPUT_CURRENT (front|back)	0x290 |0x291	Output Current CH4 U, mA		Output Current CH5 U, mA		Output Current CH6 U, mA		–		Output current of each channel from 1 to 3	1 Hz
PD_CHANNEL_SET (front|back)	0x300 |0x301	CH1	CH2	CH3	CH4	CH5	CH6	–	–	Disable (=0) / Enable (=1) each channel	On demand
IMU_Acc	0x460	Acc_X S, 0.00012207, g (earth's accel.)		Acc_Y S, 0.00012207, g		Acc_Z S, 0.0001220, g		Range U, 1, -		Acceleration Data sent by IMU Acc Factor: Range (=4) / (2^15)	100 Hz
IMU_Gyro	0x461	Vel_X S, 0.0038147, deg/s		Vel_Y S, 0.0038147, deg/s		Vel_Z S, 0.0038147, deg/s		Range S, 1, -		Gyro Data sent by IMU Vel factor: Range (=125) / (2^15)	100 Hz
IMU_Temp	0x462	Temp S, 0.0625 Offset: 25, °C		–	–	–	–	–	–	Temperature Data sent by IMU Temp Factor: (1/16)	100 Hz

		Byte #									
Command	**ID**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Description**	**Rate**
UWB_Position	0x470	Pos_x		Pos_y		Pos_z		–	–	Position according to Ultra-Wideband Localization System	–
Calibration_Req_IMU	0x541	–		–	–	–	–	–	–	Triggers IMU offset calibration, make sure robot is not moving	On demand

5.6 Troubleshooting

Table 5 contains common failures related to the CAN bus.

Table 5 Troubleshooting

Apparent fault	Probable cause	Recommended action
No communication between front/back of the robot, e.g. front wheels do not turn with main control in the back	CAN high/low lines defect. Likely to occur across a plug	Check the resistance along the CAN lines. In case of high resistance (>10 Ω) or no connection, search for missing contact, especially in the cables leading to the LED boards
CAN messages are received by some participants, but not others	Timing tolerance prevents reception and acknowledgement	Increase the synchronization jump width in the device configurator of the board not receiving the messages
The BMS shuts down immediately after startup	A CAN probe or adapter acknowledges the first CAN messages of the BMS, which is not expected by the BMS	Remove/power down the CAN probe or adapter during the startup of the BMS, or switch it to listening mode if available

Revision history

Document revision	Date	Description of changes
V 1.0	2025-07-30	Initial release

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

PSOC™, formerly known as PSoC™, is a trademark of Infineon Technologies. Any references to PSoC™ in this document or others shall be deemed to refer to PSOC™.

Edition 2025-07-30

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

AN151158

Important notice

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind (including without limitation warranties of non-infringement of intellectual property rights of any third party) with respect to any and all information given in this application note.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.