# Infineon Mobile Robot (IMR) main control

## User guide for DEMO_IMR_MAINCTRL_V1

## About this document

### Scope and purpose

The purpose of this document is to provide a comprehensive functional description and user guide for the DEMO_IMR_MAINCTRL_V1 demonstration board. Both hardware and firmware are shown and explained in detail with additional flowcharts where necessary. Furthermore, a basic quick-start guide is provided for the intended use as the Infineon Mobile Robot (IMR) main processor.

### Intended audience

The intended audience for this document is design engineers, technicians, and developers of electronic systems.

### Infineon components featured

- TLS205B0EJV33 low-noise 3.3 V fixed linear voltage regulator
- TLE9351BVSJ High-speed automotive CAN transceiver (CAN and CAN FD)
- XMC4700-F100K2048 AA 32-bit microcontroller with Arm® Cortex® M4 (XMC™)

## Important notice

**“Evaluation Boards and Reference Boards” shall mean products embedded on a printed circuit board (PCB) for demonstration and/or evaluation purposes, which include, without limitation, demonstration, reference and evaluation boards, kits and design (collectively referred to as “Reference Board”).**

**Environmental conditions have been considered in the design of the Evaluation Boards and Reference Boards provided by Infineon Technologies. The design of the Evaluation Boards and Reference Boards has been tested by Infineon Technologies only as described in this document. The design is not qualified in terms of safety requirements, manufacturing, and operation over the entire operating temperature range or lifetime.**

**The Evaluation Boards and Reference Boards provided by Infineon Technologies are subject to functional testing only under typical load conditions. Evaluation Boards and Reference Boards are not subject to the same procedures as regular products regarding returned material analysis (RMA), process change notification (PCN) and product discontinuation (PD).**

**Evaluation Boards and Reference Boards are not commercialized products, and are solely intended for evaluation and testing purposes. In particular, they shall not be used for reliability testing or production. The Evaluation Boards and Reference Boards may therefore not comply with CE or similar standards (including but not limited to the EMC Directive 2004/EC/108 and the EMC Act) and may not fulfill other requirements of the country in which they are operated by the customer. The customer shall ensure that all Evaluation Boards and Reference Boards will be handled in a way, which is compliant with the relevant requirements and standards of the country in which they are operated.**

**The Evaluation Boards and Reference Boards as well as the information provided in this document are addressed only to qualified and skilled technical staff, for laboratory usage, and shall be used and managed according to the terms and conditions set forth in this document and in other related documentation supplied with the respective Evaluation Board or Reference Board.**

**It is the responsibility of the customer’s technical departments to evaluate the suitability of the Evaluation Boards and Reference Boards for the intended application, and to evaluate the completeness and correctness of the information provided in this document with respect to such application.**

**The customer is obliged to ensure that the use of the Evaluation Boards and Reference Boards does not cause any harm to persons or third-party property.**
**The Evaluation Boards and Reference Boards and any information in this document is provided "as is" and Infineon Technologies disclaims any warranties, express or implied, including but not limited to warranties of noninfringement of third-party rights and implied warranties of fitness for any purpose, or for merchantability.**

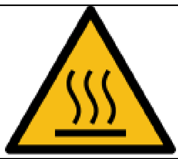
**Infineon Technologies shall not be responsible for any damages resulting from the use of the Evaluation Boards and Reference Boards and/or from any information provided in this document. The customer is obliged to defend, indemnify, and hold Infineon Technologies harmless from and against any claims or damages arising out of or resulting from any use thereof.**

**Infineon Technologies reserves the right to modify this document and/or any information provided herein at any time without further notice.**

# Safety precautions

*Note:* *Please note the following warnings regarding the hazards associated with development systems.*

**Table 1 Safety precautions**

| | |
|---|---|
|  | **Warning:** *The DC link potential of this board is up to 1000 VDC. When measuring voltage waveforms by oscilloscope, high-voltage differential probes must be used. Failure to do so may result in personal injury or death.* |
|  | **Warning**: *The evaluation or reference board contains DC bus capacitors, which take time to discharge after removal of the main supply. Before working on the drive system, wait 5 minutes for capacitors to discharge to safe voltage levels. Failure to do so may result in personal injury or death. Darkened display LEDs are not an indication that capacitors have discharged to safe voltage levels.* |
|  | **Warning:** *The evaluation or reference board is connected to the grid input during testing. Hence, high-voltage differential probes must be used when measuring voltage waveforms by an oscilloscope. Failure to do so may result in personal injury or death. Darkened display LEDs are not an indication that capacitors have discharged to safe voltage levels.* |
|  | **Warning:** *Remove or disconnect power from the drive before you disconnect or reconnect wires, or perform maintenance work. Wait five minutes after removing power to discharge the bus capacitors. Do not attempt to service the drive until the bus capacitors have discharged to zero. Failure to do so may result in personal injury or death.* |
|  | **Caution:** *The heat sink and device surfaces of the evaluation or reference board may become hot during testing. Hence, necessary precautions are required while handling the board. Failure to comply may cause injury.* |
|  | **Caution:** *Only personnel familiar with the drive, power electronics and associated machinery should plan, install, commission and subsequently service the system. Failure to comply may result in personal injury and/or equipment damage.* |
|  | **Caution:** *The evaluation or reference board contains parts and assemblies sensitive to electrostatic discharge (ESD). Electrostatic control precautions are required when installing, testing, servicing or repairing the assembly. Component damage may result if ESD control procedures are not followed. If you are not familiar with electrostatic control procedures, refer to the applicable ESD protection handbooks and guidelines.* |
|  | **Caution:** *A drive that is incorrectly applied or installed can lead to component damage or reduction in product lifetime. Wiring or application errors such as undersizing the motor, supplying an incorrect or inadequate AC supply, or excessive ambient temperatures may result in system malfunction.* |
|  | **Caution:** *The evaluation or reference board is shipped with packing materials that need to be removed prior to installation. Failure to remove all packing materials that are unnecessary for system installation may result in overheating or abnormal operating conditions.* |

# Table of contents

# 1 Introduction

## 1.1 Mobile robot general description

Mobile robots are now common in applications from logistics, warehouse centers, production sites, hospitals, restaurants, 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 tapes for navigation.
- **Autonomous mobile robots (AMR):** Not "fixed", and do not need external paths as these use autonomous mapping, localization, navigating, and obstacle avoidance by using sensors.

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

## 1.2 IMR description

The board described in this document is primarily targeted to be used in combination with the Infineon Mobile Robot (IMR). The IMR is a comprehensive robotic platform intended to be used with a wide variety of different boards (sensors, motor control, wireless communication, battery management, etc.).

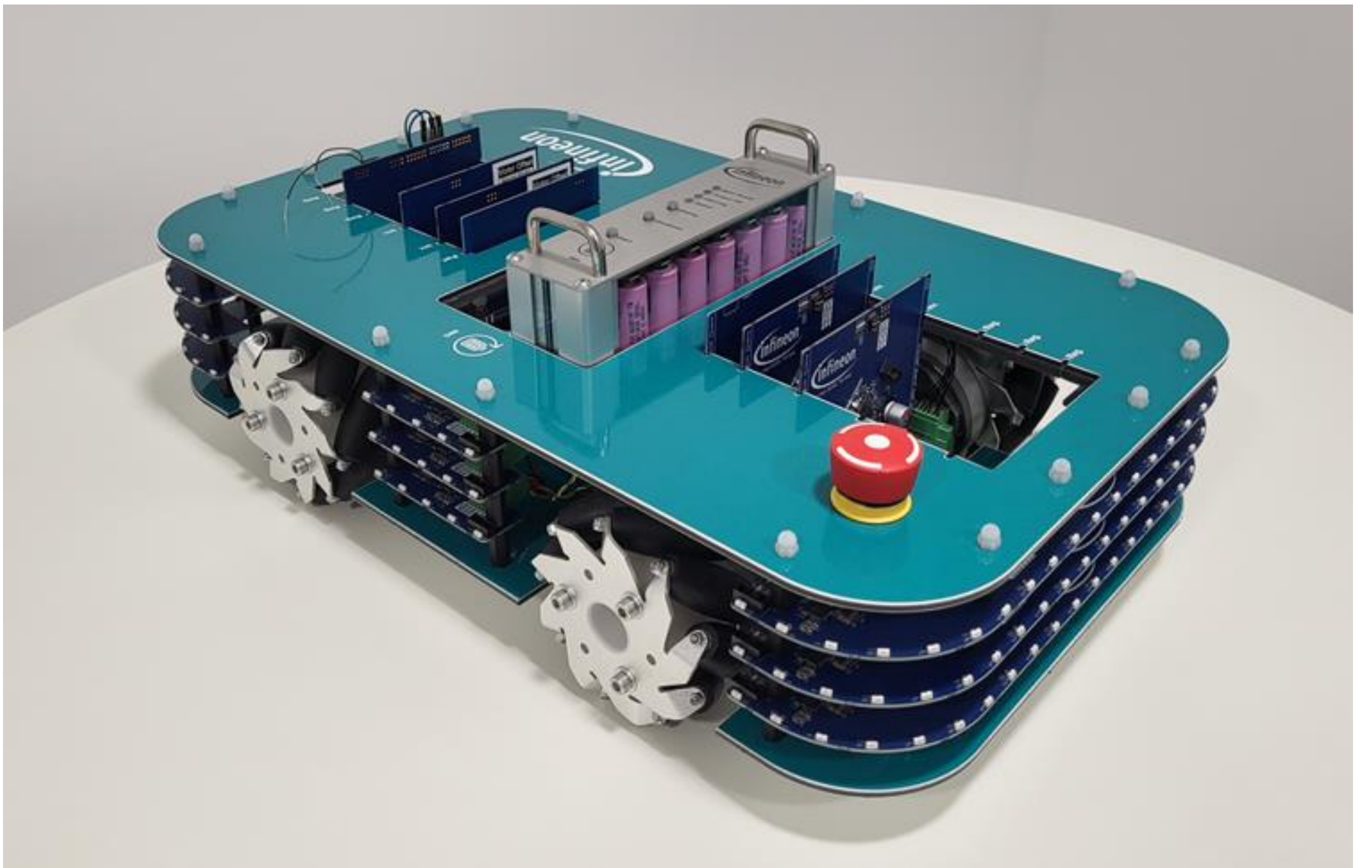


**Figure 1 Isometric view of the IMR**

The overall target of the IMR is to provide a demonstration platform for autonomous service robot functionalities using Infineon components. In the initial version of the system, the target is to use a commercially available remote control in combination with the DEMO_IMR_MAINCTRL_V1 board to control the IMR's initial features.

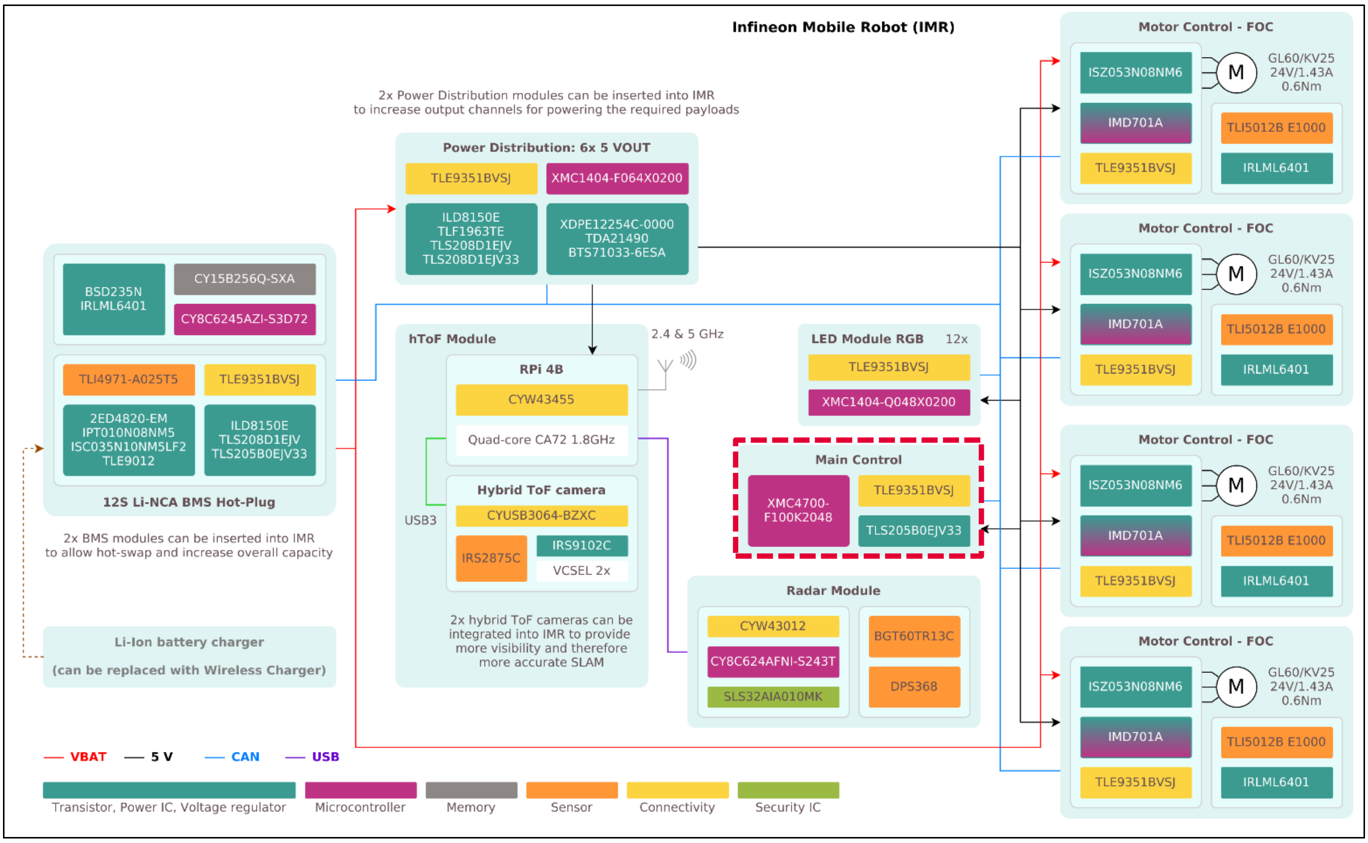



**Figure 2 Overview of the IMR with the board described in this document is highlighted in red**

## 1.3 Demo board description

The DEMO_IMR_MAINCTRL_V1 board is a modular robot control solution, specifically designed to work with the IMR card edge connector interface. The board features an XMC4700 series Arm® Cortex® M4 microcontroller performing all necessary calculations regarding robot movements. The board can be supplied either via the external 5 V power connector or via the 5 V line provided by the edge card connector. Furthermore, this voltage is used by the TLS205B0EJV33 linear regulator to generate the necessary 3.3 V supply for the XMC4700.

After installing the board vertically into one of IMR's regular slots, all the necessary communication is handled via the provided serial communication CAN interface. The board utilizes the TLE9351BVSJ high-speed automotive CAN transceiver, which is capable of supporting up to 5 Mbit/sec data rate. Specific to IMR, the standard data rate of up to 1 Mbit/sec is adopted. The board's unique CAN identifier can be set using the on-board DIP switch.

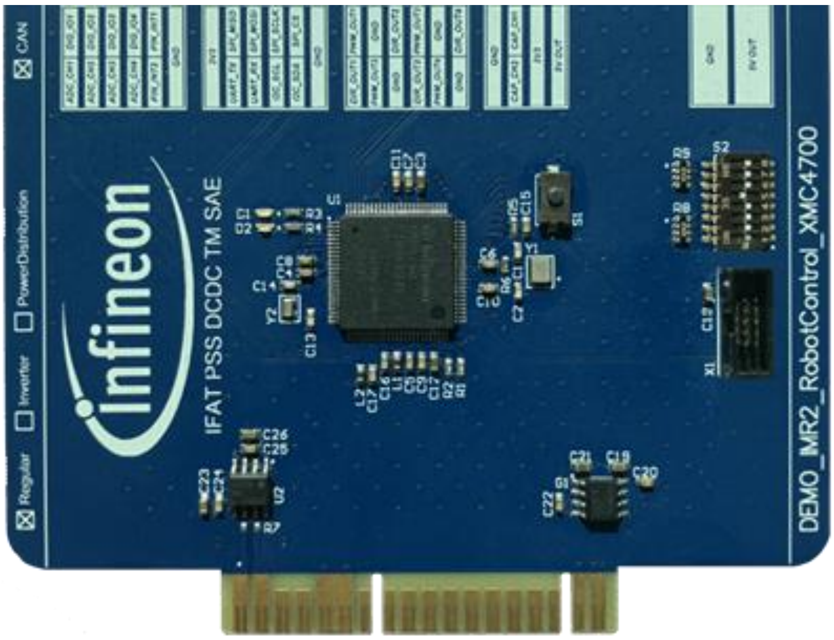



**Figure 3 DEMO_IMR_MAINCTRL_V1 board top view (width × height: 100 mm × 88 mm)**

## 1.4 Overview of the connectors

To support a wide variety of different peripherals in combination with the DEMO_IMR_MAINCTRL_V1 board, several connectors are placed on the board. To utilize most of the available pins on the XMC4700, the board provides: four digital inputs/outputs, four analog inputs, two ERU digital inputs, four digital PWM outputs, four digital outputs and two PWM capture inputs. For serial communication, the connectors feature UART, I2C, and SPI.

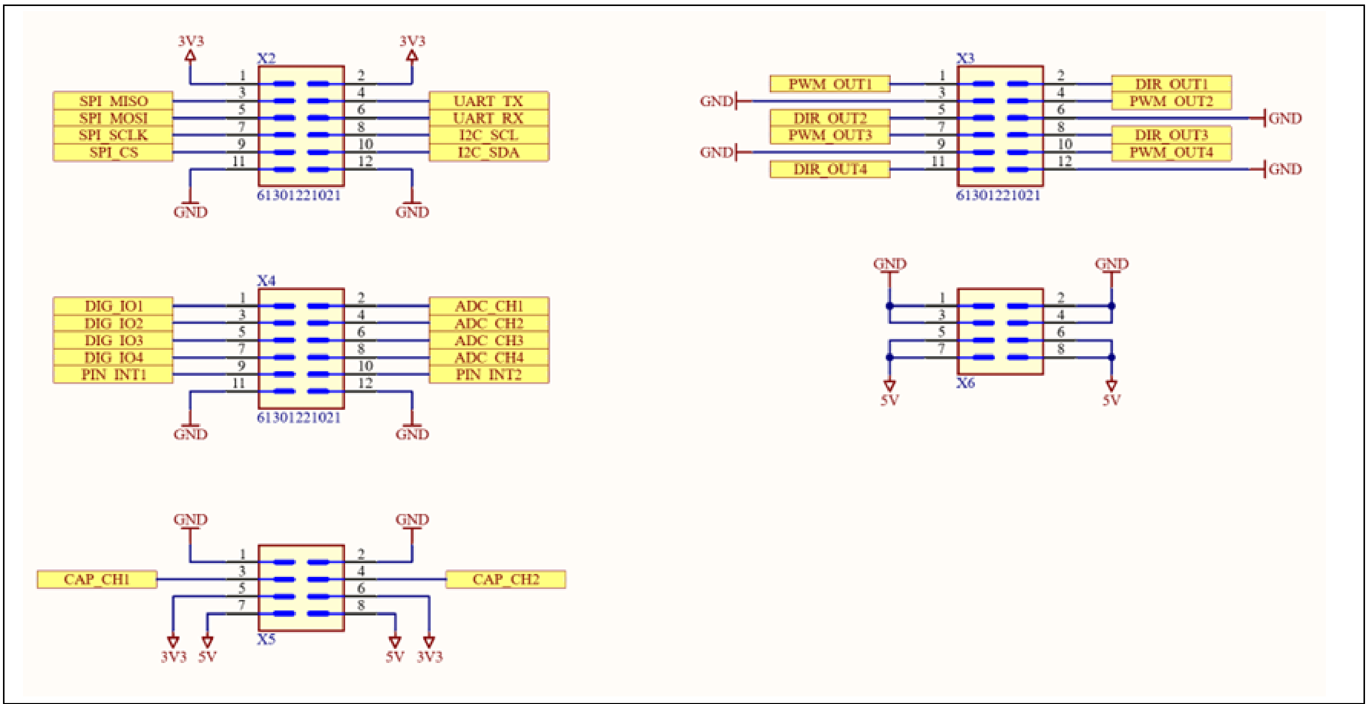



**Figure 4 Schematic overview of peripheral connectors: COM_CommunicationPort.SchDoc**

The following table shows the connection between connector X4 and XMC4700 featuring most of the dedicated analog and digital inputs/outputs.

**Table 1 Connector X4 and connection to XMC4700 – Analog/digital inputs/outputs**

| Function | XMC™ pin | Function | Description |
|---|---|---|---|
| ADC_CH1 | P14.0 | Analog input | ADC input channel 1 |
| ADC_CH2 | P14.1 | Analog input | ADC input channel 2 |
| ADC_CH3 | P14.2 | Analog input | ADC input channel 3 |
| ADC_CH4 | P14.3 | Analog input | ADC input channel 4 |
| DIG_IO1 | P3.0 | Digital input/output | Universal digital IO |
| DIG_IO2 | P3.1 | Digital input/output | Universal digital IO |
| DIG_IO3 | P3.2 | Digital input/output | Universal digital IO |
| DIG_IO4 | P0.9 | Digital input/output | Universal digital IO |
| PIN_INT1 | P2.0 | Digital input/ERU | ERU capable digital IO |
| PIN_INT2 | P2.7 | Digital input/ERU | ERU capable digital IO |
| GND | – | Signal ground | 0 V |

The following table shows the connection between connector X2 and XMC4700 for all necessary serial communication protocols featuring full-duplex UART, I2C, and four-wire SPI.

**Table 2 Connector X2 and connection to XMC4700 – USIC serial inputs/outputs**

| Function | XMC™ pin | Function | Description |
|---|---|---|---|
| 3V3 | – | Signal power | 3.3 V DC regulated |
| UART_TX | P5.1 | Serial output | UART transmit line |
| UART_RX | P5.0 | Serial input | UART receive line |
| I2C_SCL | P2.4 | Serial I2C clock | I2C clock line |
| I2C_SDA | P2.5 | Serial I2C data | I2C clock data |
| SPI_MISO | P3.4 | Serial SPI MISO | SPI master in; slave out |
| SPI_MOSI | P3.5 | Serial SPI MOSI | SPI master out; slave in |
| SPI_SCLK | P3.6 | Serial SPI SCLK | SPI serial clock |
| SPI_CS | P4.1 | Serial SPI CS | SPI chip select |
| GND | – | Signal ground | 0 V |

The connector dedicated to motor drive is described in the following table. Should the connected hardware not be able to receive motor drive commands via the CAN interface, up to four individual PWM signals can be used to interface the motor drives. Each of these PWM signals is also accompanied by an additional direction pin (digital output).

**Table 3 Connector X3 and connection to XMC4700 – Motor drive inputs/outputs**

| Function | XMC™ pin | Function | Description |
|---|---|---|---|
| PWM_OUT1 | P1.0 | Digital output/PWM | Motor drive 1 PWM output |
| DIR_OUT1 | P0.3 | Digital output | Motor drive 1 direction output |
| GND | – | Signal ground | 0 V |
| PWM_OUT2 | P1.1 | Digital output/PWM | Motor drive 2 PWM output |
| DIR_OUT2 | P0.4 | Digital output | Motor drive 2 direction output |
| GND | – | Signal ground | 0 V |
| PWM_OUT3 | P1.2 | Digital output/PWM | Motor drive 3 PWM output |
| DIR_OUT3 | P0.5 | Digital output | Motor drive 3 direction output |
| GND | – | Signal ground | 0 V |
| PWM_OUT4 | P1.3 | Digital output/PWM | Motor drive 4 PWM output |
| DIR_OUT4 | P0.6 | Digital output | Motor drive 4 direction output |
| GND | – | Signal ground | 0 V |

The following table shows the connection between connector X5 and XMC4700 featuring two PWM capture inputs. These inputs can be used to interface most PPM remote controls if necessary. Additionally, both 3.3 V and 5 V are provided to be used by an external remote control.

**Table 4 Connector X5 and connection to XMC4700 – Remote connector**

| Function | XMC™ pin | Function | Description |
|---|---|---|---|
| GND | – | Signal ground | 0 V |
| CAP_CH1 | P2.2 | Digital input/output | PWM capture input channel 1 |
| CAP_CH2 | P2.3 | Digital input/output | PWM capture input channel 2 |
| 3V3 | – | Signal power | 3.3 V DC regulated output |
| 5V | – | Signal power | 5 V DC regulated output |

**Table 5 Connector X6 – 5 V power connector**

| Function | XMC™ pin | Function | Description |
|---|---|---|---|
| GND | – | Signal ground | 0 V |
| 5V | – | Signal power | 5 V DC regulated output |

### 1.4.1 Edge card header

In addition to the connectors mentioned before, the main interface to the IMR is realized as an edge card header on the bottom of the board, as shown below. The double-sided header has 18 pins on each side for a total of 36 pins.

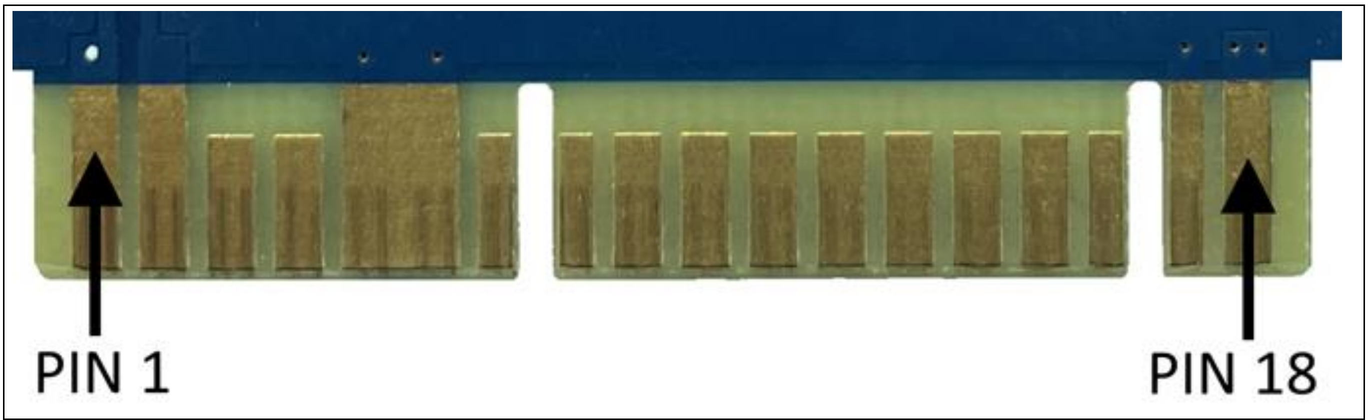



**Figure 5 Edge card header with 36 pins (18 pins symmetrical, top and bottom); designed to work with EBC18DCWN-S371 board-to-board receptacle connector**

The selected counterpart for the edge card header is the EBC18DCWN-S371 board-to-board receptacle connector from Sullins Connectors. Since all communication is done via the CAN interface (CAN_H and CAN_L), these are the only interface pins needed in addition to power connections, as can be seen in the tables below.

**Table 6 Pin out of main control board slot on IMR motherboard with thick row lines marking polarizing key positions**

| PIN | Function | PIN | Function |
|---|---|---|---|
| A1 | CAN_L | B1 | 5 V |
| A2 | CAN_H | B2 | GND |
| A3 | – | B3 | – |
| A4 | – | B4 | – |
| A5 | GND | B5 | GND |
| A6 | GND | B6 | GND |
| A7 | – | B7 | – |
| A8 | – | B8 | – |
| A9 | – | B9 | – |
| A10 | – | B10 | – |
| A11 | – | B11 | – |
| A12 | – | B12 | – |
| A13 | – | B13 | – |
| A14 | – | B14 | – |
| A15 | – | B15 | – |
| A16 | – | B16 | – |
| A17 | GND | B17 | GND |
| A18 | 5 V | B18 | 5 V |

# 2 Hardware

This chapter focuses on describing the most important functional parts of the board hardware in more detail. Additionally, the full schematics can be found in the appendix including microcontroller schematics and an overall overview.

## 2.1 3.3 V DC regulator circuitry

To provide the necessary 3.3 V power supply for the XMC4700 microcontroller, the TLS205B0EJ V33 LDO is chosen. The TLS205B0 is a micropower, low noise, low dropout voltage regulator. The device is capable of supplying an output current of 500 mA with a dropout voltage of 320 mV. Designed for use in battery-powered systems, the low quiescent current of 30 µA makes it an ideal choice [1]. Furthermore, it supports a wide input voltage range of up to 20 V, while keeping the number of necessary auxiliary components low.

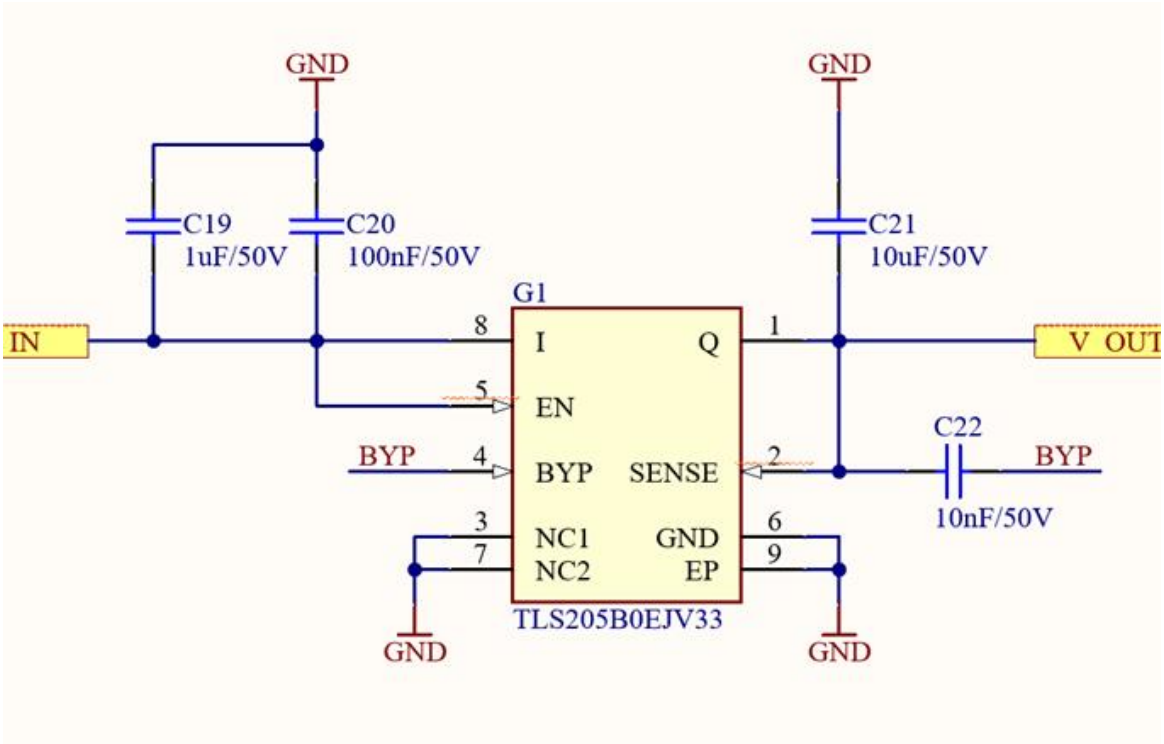



**Figure 6 TLS205B0EJV33 LDO circuit for a fixed 3.3 V output**

## 2.2 CAN serial communication interface

All necessary communication from and to the board is realized via either CAN or CAN FD bus communication. For this purpose, the TLE9351BVSJ high-speed CAN FD transceiver was chosen as it is a high-speed CAN transceiver, used in HS CAN systems for automotive applications as well as for industrial applications [2].

The VIO feature of the TLE9351BVSJ allows the user to set individual voltage levels for the CAN bus and the interface to the microcontroller. In this case, the VCC is supplied by 5 V via the edge card connector while the VIO is supplied with 3.3 V to be able to interface the transceiver directly with the microcontroller. Additionally, the STB pin is routed to the microcontroller for the possibility to send the transceiver into standby-mode.

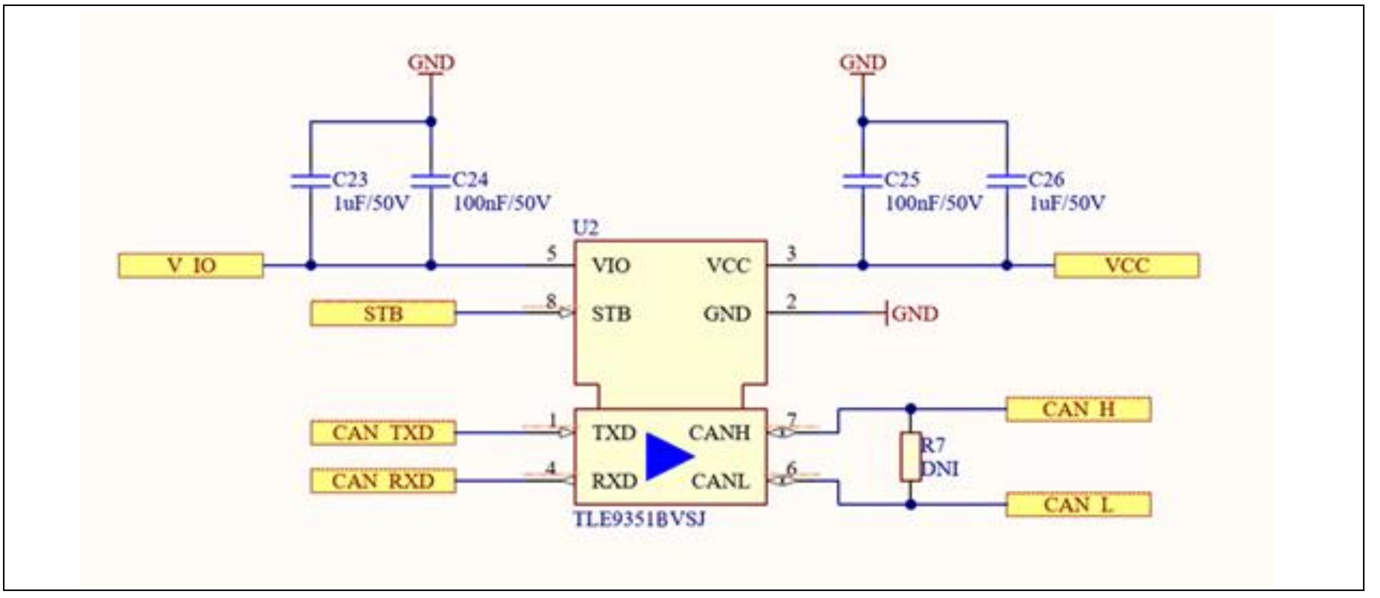



**Figure 7** TLE9351BVSJ CAN transceiver including the VIO feature for a separate IO supply level

For the unique identification of each board interfaced via CAN bus, the following DIP switch is implemented. Each pin is pulled-up via a 20 kΩ resistor and therefore provides an inverted logic that must be considered when reading the pin via the XMC4700 microcontroller.

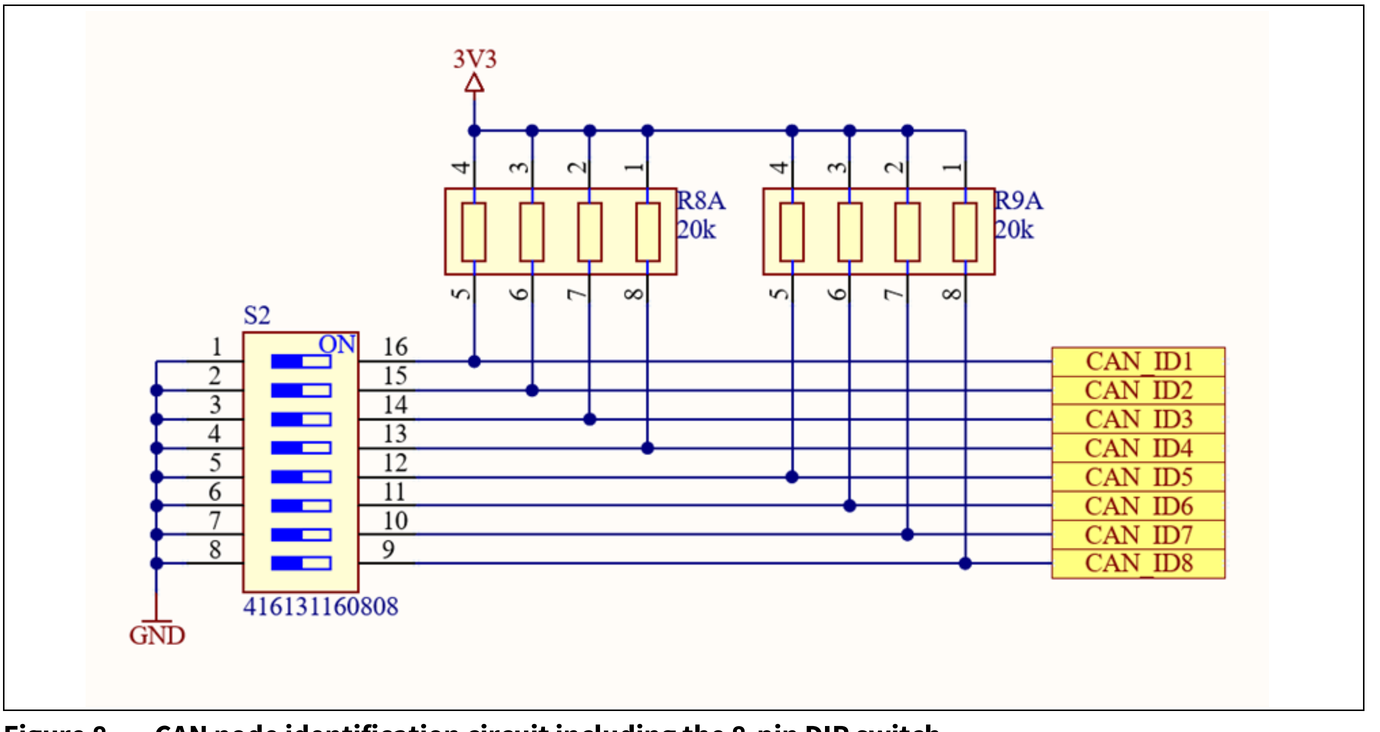



**Figure 8** CAN node identification circuit including the 8-pin DIP switch

Each individual switch is connected to an individual CAN_ID pin of the XMC4700 to individually set up to 255 devices on the same CAN/CAN FD bus. Furthermore, the following table shows each DIP switch pin and the corresponding XMC pin for reference, as well as the decimal value assigned to the pin. With the pull-up resistor to 3.3 V, the pins follow an inverted logic: the OFF state on the DIP switch represents a logic high on the XMC pin and vice-versa.

**Table 7 DIP Switch S2 – CAN node identifier logic and pin assignment**

| CAN ID Bit | ID10 | ID9 | ID8 | ID7 | ID6 | ID5 | ID4 | ID3 | ID2 | ID1 | ID0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DIP Switch | N/A | N/A | N/A | Pin 8 (MSB) | 7 | 6 | 5 | 4 | 3 | 2 | Pin 1 (LSB) |
| XMC™ pin | | | | P1.4 | P1.5 | P1.10 | P1.11 | P1.12 | P1.13 | P1.14 | P1.15 |
| RobotControl CAN ID | 0 | 0 | 0 | 0 | 0 | 0 | 1 | RobotCtrl[3:0] | | | |

# 3 Firmware

After giving a short introduction to XMC™ microcontrollers, this chapter focuses on describing the functional firmware routines in more detail. Additionally, the full firmware can be found in the appendix.

## 3.1 XMC™ microcontroller

The XMC™ microcontroller family, based on the Arm® Cortex® M cores, is suitable for real-time critical applications where an industry-standard core is needed. It is dedicated to applications in the segments of power conversion, factory and building automation, transportation, as well as home appliances. The XMC4000 series features the Arm® Cortex® M4 core in combination with exceedingly versatile peripherals for most applications [3].

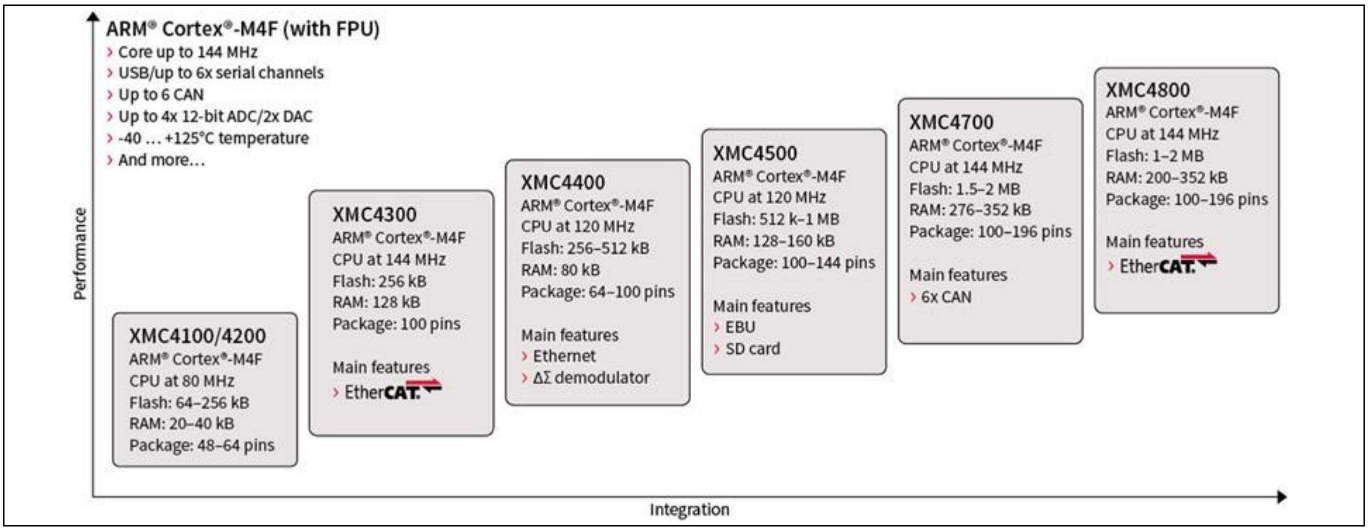



**Figure 9 XMC4000 series sub-groups [3]**

The microcontroller for this specific application must perform at a reasonable clock frequency, provide multiple analog and digital channels, and have the capability to support multiple communication interfaces (I2C, SPI, UART, CAN etc.). The 100-pin LQFP100 packaged XMC4700-F100K2048 supports a 144 MHz clock frequency, 2048 kB of flash memory, 352 kB of SRAM and enough I/O pins to support the required functions [4].

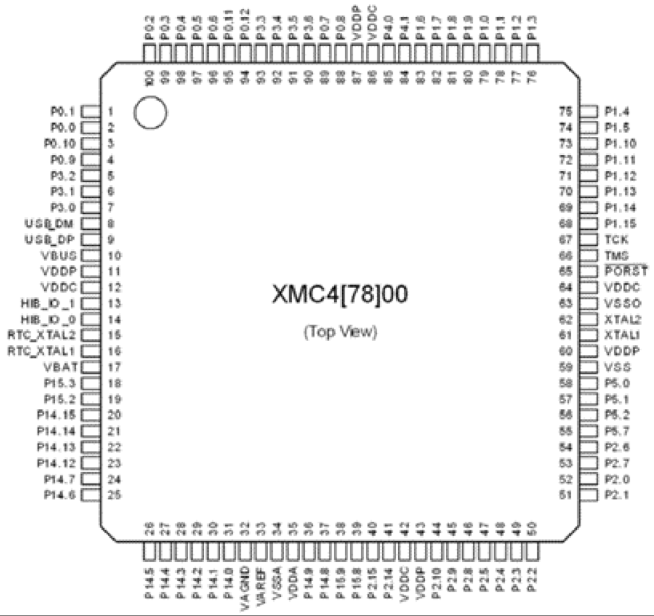



**Figure 10 XMC4700/4800 LQFP100 pin configuration (top view) [4]**

## 3.2 Firmware description and flowcharts

The firmware main loop starts by initializing the peripherals with the settings from the DAVE™ apps, which can be found inside the project repository. After initialization, the boards' unique CAN node identification is set using the 8-pin DIP switch. Afterwards, the firmware enters the program's while-loop to continuously check for new SBUS messages that can be parsed into any commands for the IMR.

Whenever a complete SBUS message has been received, the remote-control channel data can be processed using the "`Process_RC_Inputs`" routine, followed by resetting the timeout variable that is cyclically set in the 100 ms routine. More information regarding the SBUS protocol can be found, e.g., on the Bolderflight GitHub [5]. With the processed data, the "Switch_SA" position is checked to decide if a new lighting effect needs to be sent to the LED boards. Next, the "Switch_SD" position is checked to decide if the robot is "Armed" or not. If the switch is in its lowest position, the remote-control inputs are processed using the "`Process_MecanumWheels`" routine, otherwise all motor outputs are set to zero. Further details on switch positions can be found in the "Quick Start Guide" section about the remote-control implementation.

After processing the appropriate mecanum wheel commands, they are then sent to each individual motor (Front Left – **FL**, Front Right – **FR**, …) using the "`CAN_TX_Request`" routine. If needed, the lighting CAN command is also sent to the respective LED boards.

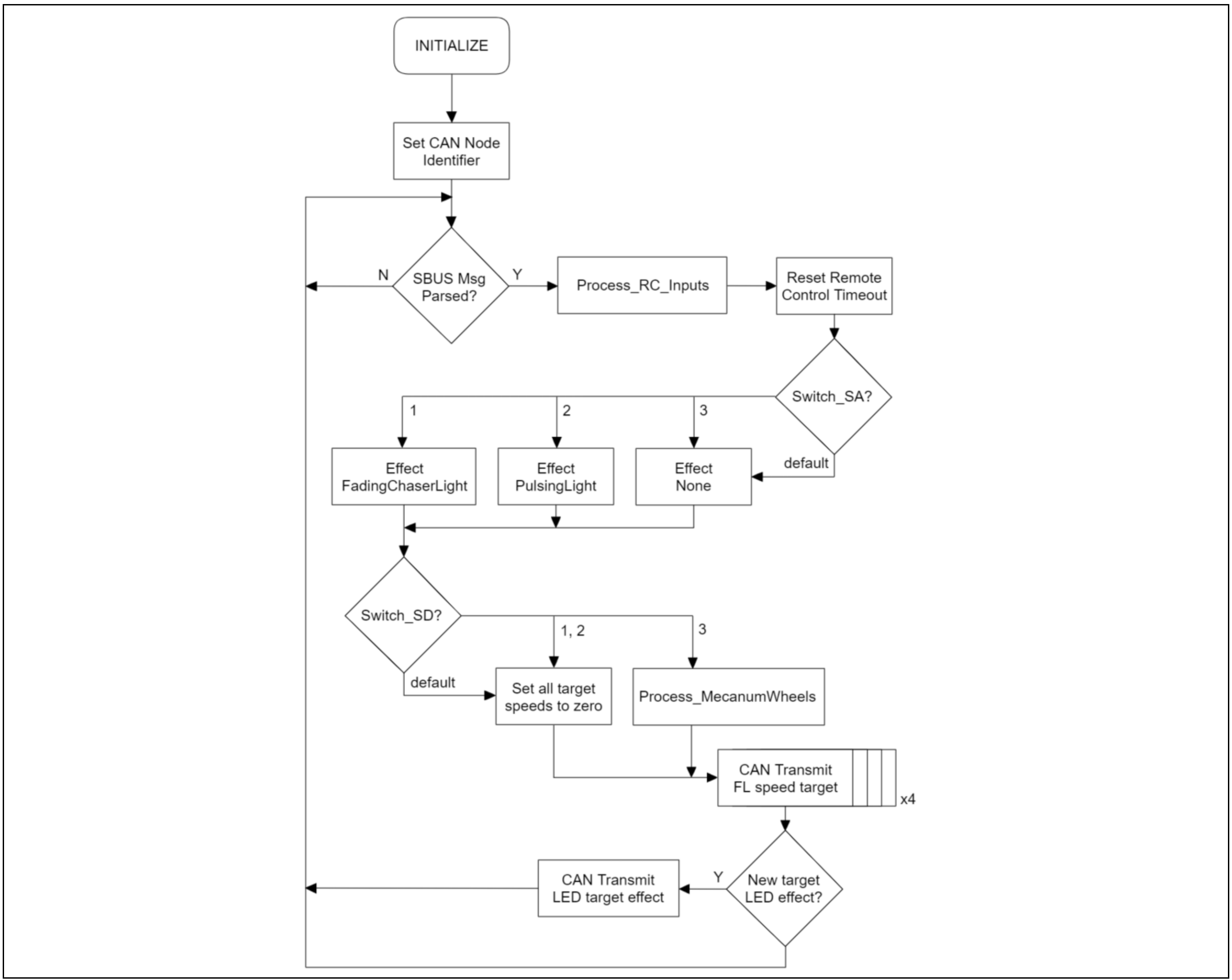



**Figure 11 Main program loop**

To prepare for the event of a remote-control failure or disconnect, a periodic 100 ms routine is implemented. The routine checks if the timeout variable is still active since the last time it has been set inside the same routine. This means, if the routine is called two consecutive times without the timeout variable being reset in the main routine, a timeout has been detected and a zero command is sent to each of the motor drives.

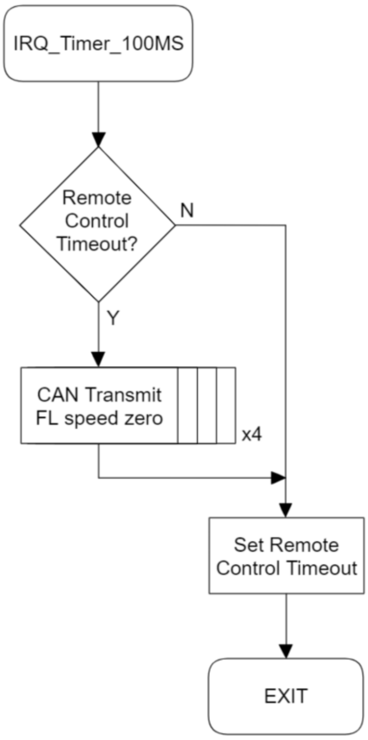



**Figure 12 Interrupt service routine for detecting timeouts periodically (100 ms)**

The following diagram shows the "`Process_RC_Inputs`" routine used to process the remote-control channel data and normalize it for further calculations. The stick inputs are normalized using the values for their center value, the maximum deviation from the center and the normalized target maximum value for further processing. Additionally, an adjustable square dead-zone is implemented to reduce unnecessary and unwanted twitching of the motors. The dead-zone assists the user by allowing a certain threshold when moving the control sticks back to the center position. This routine is performed for each individual stick input:

- Left Stick: Up, Down, Left, and Right
- Right Stick: Up, Down, Left, and Right

Subsequently, a similar process is used on each of the switch channels. Depending on the remote-control setup, each channel must be checked if it fits between the selected channel range. Depending on the selected switch, either two or three switch positions can then be identified. In the current version, lateral movements (left, right, forward, backward, and diagonal) are performed using the right control stick while rotational movements can be performed by moving the left stick left and right.

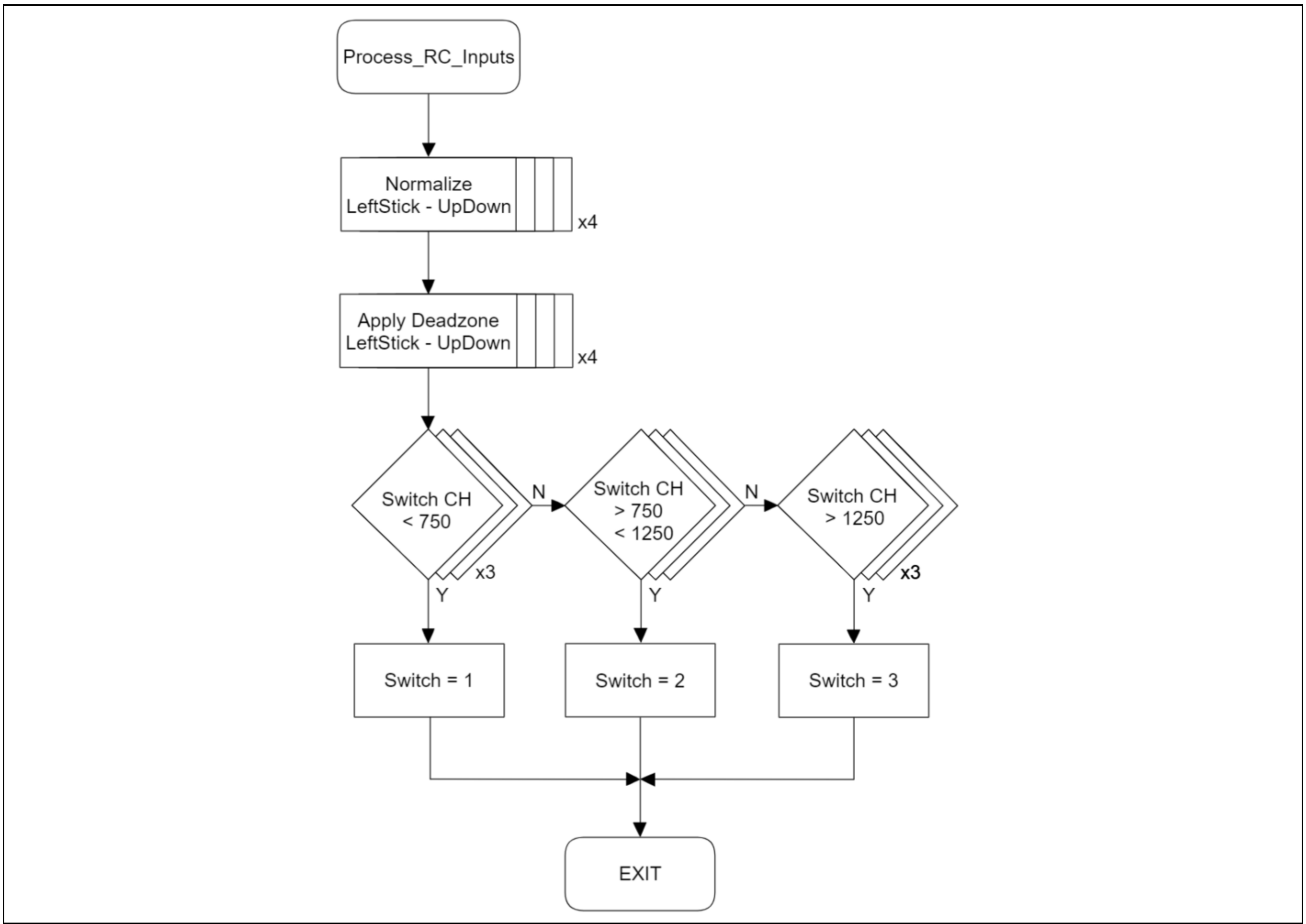



**Figure 13** **Routine for processing received channel information from remote control**

After scaling and normalizing all required remote-control inputs, further processing needs to be done to define the motor control signals. Each vehicle direction needs specific signals to each of the motor drives to perform either lateral (**a**, **b,** or **c**) or rotational (**d**, **e,** and **f**) movements, also simultaneously. The diagram below shows the possible movements and each respective motor direction to perform the said movement [6].

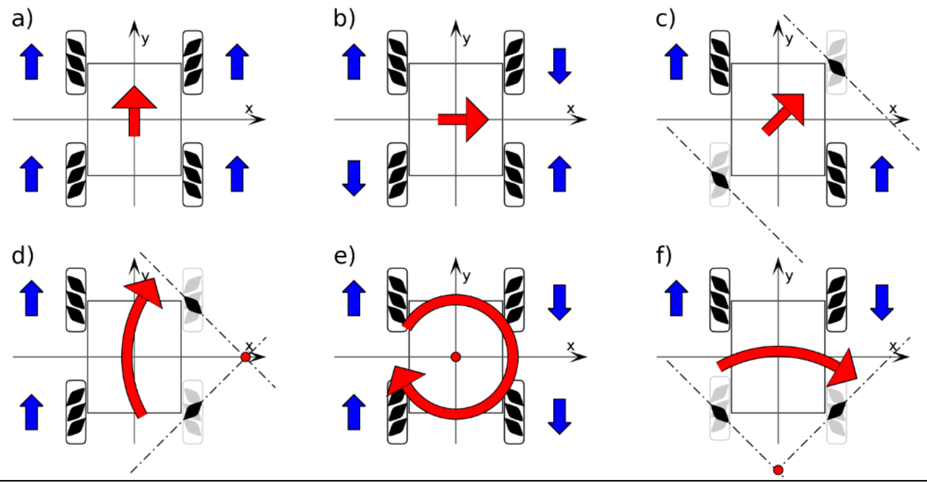



**Figure 14 Specific motor actuation for movement in any direction (blue: wheel drive direction; red: vehicle moving direction) [6]**

The following diagram shows the "`Process_MecanumWheels`" routine that computes the motor control signals for each of the four motor drives on the IMR. Additionally, each of the calculated outputs is again limited between a maximum and minimum value (default: **-100** to **100**; adjustable) as a safety precaution. Additional details about the mecanum wheel calculation performed can be found in the firmware appendix.

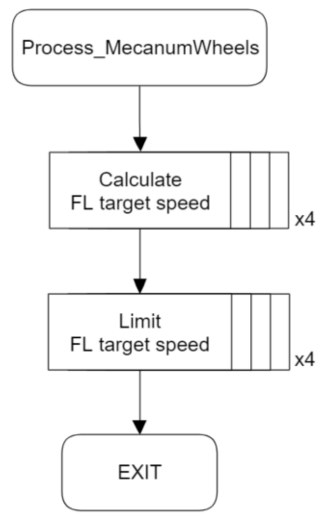



**Figure 15 Routine to calculate target mecanum wheel speeds from remote control inputs**

The following diagram shows the routine used for all CAN transmissions coming from the board. The routine first updates the target CAN ID followed by the data length and finally updates the data buffer for transmission. After initializing the transmission, the routine constantly checks the status of the transmission. Should the transmission not yet be finished, a specific timeout period can be set to wait for the transmission to go through. After the said period, the transmission is canceled, and the program proceeds as normal. This timeout is necessary due to the chance of a problem with CAN lines, which would otherwise stall the program until resolved.

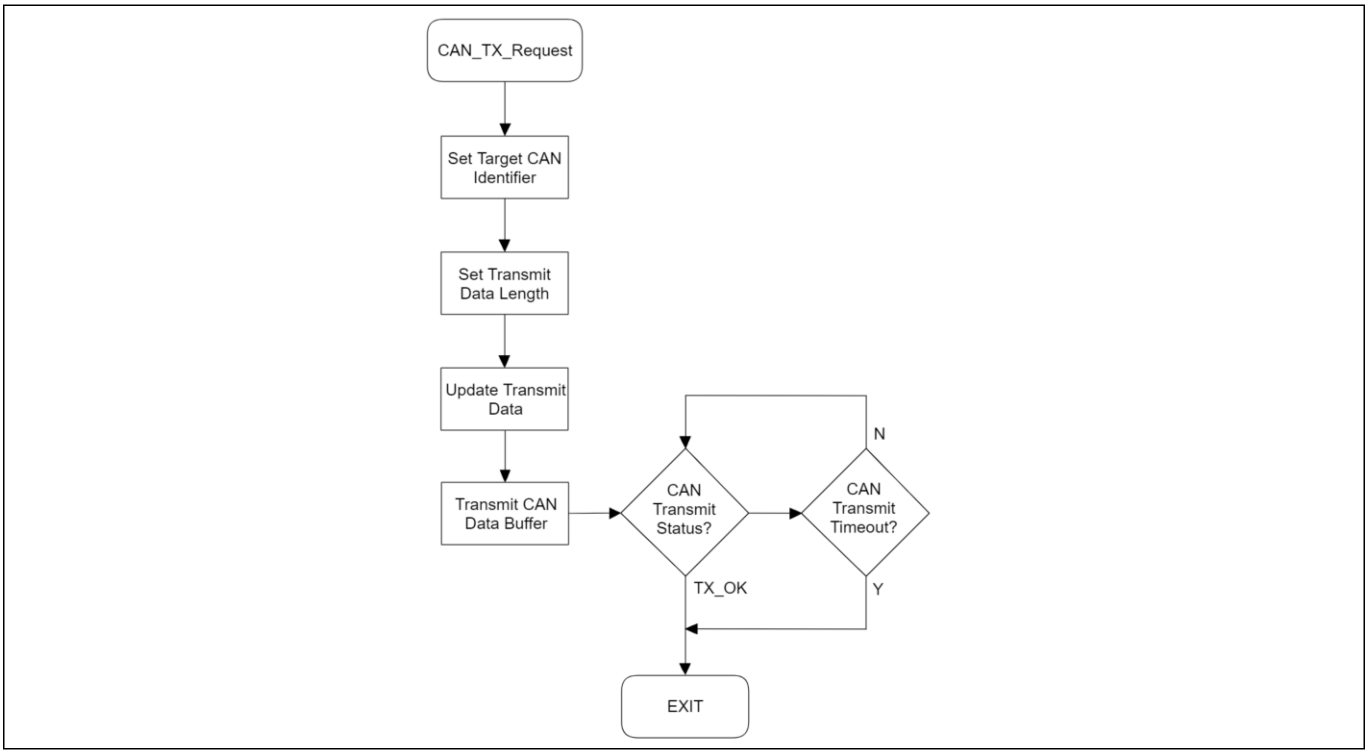



**Figure 16 CAN message transmission request routine**

### 3.2.1 CAN message structure

This chapter describes the currently available CAN messages to each specific recipient in more detail. Each message follows a similar structure following the standardized CAN protocol, which can be seen in the following figure. In general, each message consists of a unique identifier describing the recipient (11 bits) followed by the data bytes (0 – 64 bits) containing the relevant information. Additionally, several bits are included to allow a successful operation: Start of frame, stuff bits, Data Length Code (DLC), checksums, etc. [6].

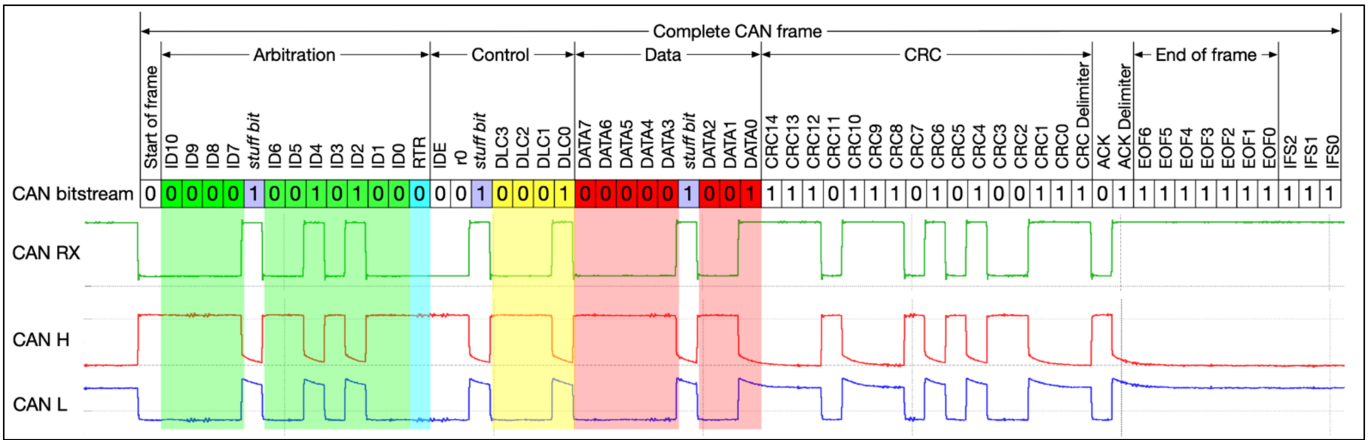



**Figure 17 A complete CAN bus frame, including stuff bits, a correct CRC, and inter-frame spacing [7]**

In the case of the CAN identifier (Arbitration), only eight bits are used in the IMR to specify the type and specific number of the board (**ID0** to **ID7**). The remaining three bits (**ID8** to **ID10**) remain on their original level (logic low). For the data frames **Byte 1** always corresponds to the byte that is transmitted/received first, followed by the other bytes from the same CAN transmission (Byte 2, …).

To distinguish between different hardware that can be found on the IMR, each type of board is classified via a specific CAN ID range and therefore reserving a certain address region. Depending on the type of board, specific CAN ID bits can be set differently to differentiate boards from the same type (See Table 8). This means, only marked bit-fields are taken from DIP-switch settings, others are defined in the associated firmware due to the board type. The table below shows the currently available CAN node masks and is subject to change/adaption with upcoming hardware releases. Additionally, these ID values can be found in use in the firmware appendix.

**Table 8 Specific CAN identifier generation**

<table>
<tr><th>CAN ID Bit</th><th>ID10</th><th>ID9</th><th>ID8</th><th>ID7</th><th>ID6</th><th>ID5</th><th>ID4</th><th>ID3</th><th>ID2</th><th>ID1</th><th>ID0</th><th>CAN-ID</th></tr>
<tr><td>Robot control</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td colspan="4">RobotCtrl[3:0]</td><td>0x10 | RobotCtrl</td></tr>
<tr><td>Motor control</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td colspan="2">INV[1:0]</td><td>0x60 | INV</td></tr>
<tr><td>Sensor LED</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td colspan="4">LED[3:0]</td><td>0xC0 | LED</td></tr>
<tr><td>BMS</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>BMS[0]</td><td>0x3C | BMS</td></tr>
<tr><td>Power board</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>PWR[0]</td><td>0xA8 | PWR</td></tr>
</table>

A total of eight data bytes can be transmitted in a single CAN message. The overall CAN communication setup used on the IMR consists of a single byte containing the target command, followed by another byte containing the CAN message sender ID used for identifying where a specific CAN message came from, in the system. This results in a maximum of six data bytes remaining for additional data if necessary. A detailed overview of the message setup can be seen in the table below.

**Table 9 IMR CAN message data structure overview**

<table>
<tr><th></th><th>Byte 1</th><th>Byte 2</th><th>Byte 3</th><th>Byte 4</th><th>Byte 5</th><th>Byte 6</th><th>Byte 7</th><th>Byte 8</th></tr>
<tr><td>CAN Message</td><td>Command</td><td>Sender ID</td><td colspan="6">Data[6:0]</td></tr>
</table>

Furthermore, each board has its own specific commands that can be used to transmit and receive CAN messages from specific IMR boards. The specific CAN messages are described in the following sections.

#### 3.2.1.1 Robot control CAN messages

This section describes the currently available commands to transmit CAN messages to the DEMO_IMR_MAINCTRL_V1 board. The command available allows the motor control boards on the IMR to transmit their current motor position via the encoder value to the robot control board. The encoder value itself is stored inside an unsigned 16-bit integer and must therefore be split into an upper and lower byte for transmission.

**Table 10 IMR control CAN message command overview**

| | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Description |
|---|---|---|---|---|---|---|---|---|---|
| EncoderValue Command | 0x0E | Sender ID | Encoder_Value | | N/A | N/A | N/A | N/A | Transmit encoder values from motor control to robot control |

#### 3.2.1.2 Motor control CAN messages

This section covers the different commands for the CAN communication with the IMR motor control boards. Each specific motor control board is able to receive a motor speed target value as a 2-byte data value. The calculated target motor speed from -100 to 100 is stored inside a signed 16-bit integer and must therefore be split into an upper and lower byte for transmission. The table below shows all available messages that can currently be received and interpreted by the motor control boards.

**Table 11 IMR control CAN message command overview**

| | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Description |
|---|---|---|---|---|---|---|---|---|---|
| Motorspeed Command | 0x1D | Sender ID | Target_Speed | | N/A | N/A | N/A | N/A | Transmit target speed to motor control boards |

### 3.2.1.3 Sensor LED CAN messages

Similar to the previous interface, the sensor LED boards can be interfaced using the same CAN message protocol. Currently, there are three different lighting modes that can be activated via the commands listed below. Only the first data byte (Command-Byte) currently sets the target lighting effect, the remaining data bytes are reserved for future purposes.

**Table 12 IMR sensor LED CAN message command overview**

| | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Description |
|---|---|---|---|---|---|---|---|---|---|
| Effect_None Command | 0x4A | Sender ID | 0x00 | | N/A | N/A | N/A | N/A | Transmit command to stop lighting effects |
| ChaserLight Command | 0x4B | Sender ID | 0x00 | | N/A | N/A | N/A | N/A | Transmit ChaserLight to sensor LED boards |
| PulsingLight Command | 0x4C | Sender ID | 0x00 | | N/A | N/A | N/A | N/A | Transmit pulsing light to sensor LED boards |

Each command only needs to be sent to one of the Sensor LED boards in each row, since communication is done between all sensor LED boards automatically to match the same lighting pattern.

### 3.2.1.4 BMS CAN messages

This section describes the possible CAN messages to the IMR BMS boards available in the system. Currently reserved for future development.

**Table 13 IMR BMS CAN message command overview**

| | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Description |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | 0xB0 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xB1 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xB2 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xB3 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |

#### 3.2.1.5 Power board CAN messages

This section describes the possible CAN messages to the IMR power boards available in the system. Currently reserved for future development.

**Table 14 IMR Power Board CAN message command overview**

| | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 | Description |
|---|---|---|---|---|---|---|---|---|---|
| Reserved | 0xF0 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xF1 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xF2 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |
| Reserved | 0xF3 | Sender ID | N/A | N/A | N/A | N/A | N/A | N/A | Reserved |

# 4 Quick start guide

This section is intended to be a reference to get started with using the described board as a robot control module for IMR. Further chapters include the necessary steps to program and debug the hardware, set up the hardware for CAN communication, and to interface a commercial remote control for the use with the IMR.

## 4.1 Software deployment and debugging

The firmware controlling this demo board was developed using the DAVE™ IDE, which can be downloaded free of charge from the Infineon website. Programming and debugging are carried out via the XMC™ Link isolated debug probe, shown below. The DEMO_IMR_MAINCTRL_V1 board is connected to the debug probe using the smaller 1.27 mm pitch ribbon cable and the 10-pin box header X1 (the larger ribbon cable will not be used).

*Note: An additional external power supply is required when using an isolated debugging probe.*

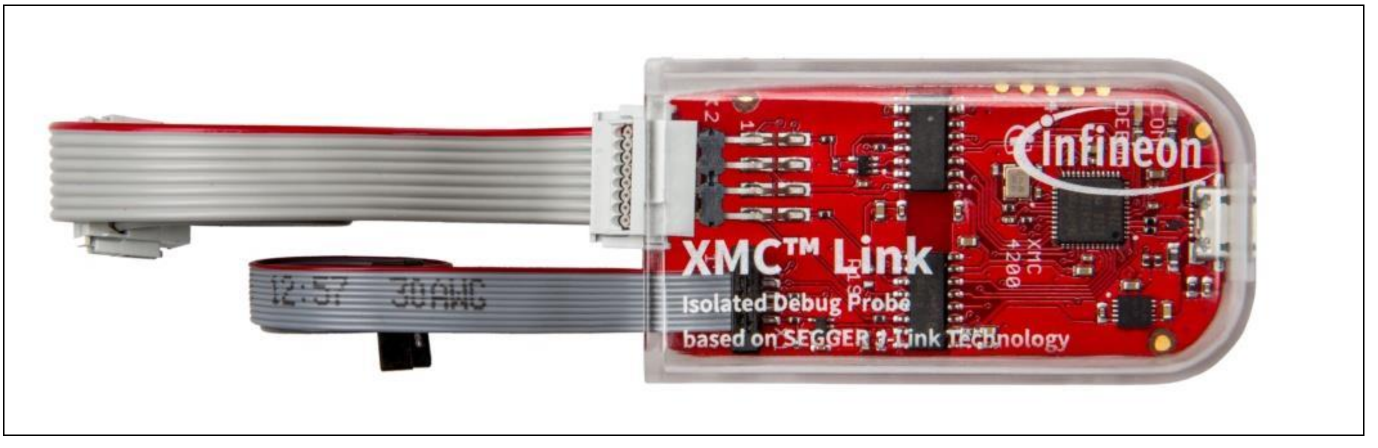



**Figure 18 XMC™ Link isolated debugger probe**

A project was created within DAVE™ IDE containing the device definition, settings, and source files required to compile and build the executable code, which can be downloaded into the flash program memory of the XMC™ controller. To debug and apply the firmware to the board, the newest version of the DAVE™ IDE is necessary and can be downloaded directly from the Infineon website. Inside the DAVE™ IDE the firmware must be imported via **File > Import > Infineon > DAVE Project**.

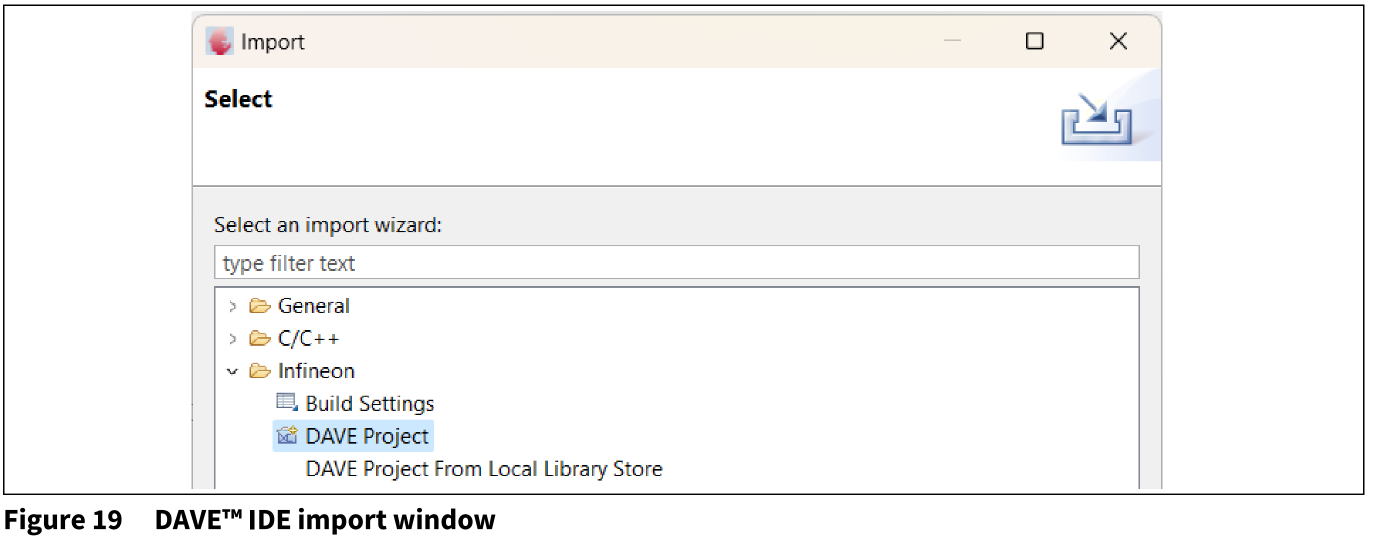



**Figure 19 DAVE™ IDE import window**

After importing the project into the DAVE™ IDE, its contents can be found in the "C/C++ Projects" tab as shown below. The "main.c" file contains the main body of the source code, "board_pin_label.bpl" contains necessary naming for easier pin assignments, and the SBUS folder contains the library for the serial remote-control interface.

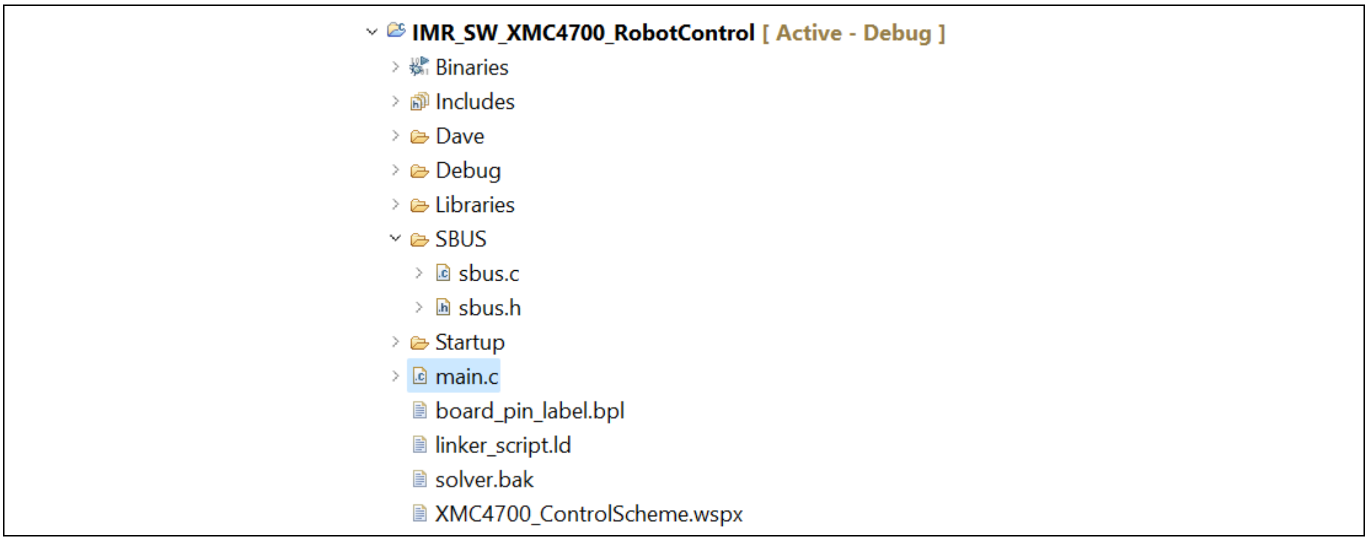



**Figure 20 IMR_SW_XMC4700_RobotControl – DAVE IDE project structure**

The apps are listed in the app dependency tree window in the DAVE™ CE screen and displayed graphically in the app dependency window. Double-clicking on any app opens a menu allowing the programmer to configure the app. The manual pin allocator is used to select which I/O pins are mapped to which of the app inputs and outputs. When the configuration is complete, the corresponding ".c" and ".h" source code files are generated by clicking the "Generate Code" button normally located on command bar, which is located below the menu bar at the top of the DAVE™ CE screen shown below. For more complex functionality it is necessary to obtain the relevant functions from the DAVE™ library.

To apply the imported firmware to the board, the firmware must be built using the "Build Active Project" button and then programmed on the board using the "Debug" button. The DAVE™ IDE will then bring the user into the debug screen, where the program can be launched.


**Figure 21 DAVE™ IDE main commands**

## 4.2 In-circuit debugging with Micriμm® μC/Probe™ for XMC™

For further debugging, the μC/Probe™ XMC™ developed by Micriμm® is available (see https://infineoncommunity.com/uc-probe-xmc-software-download_id712). The μC/Probe™ can be used to track selected variable values without the need to pause the debugging session. For this specific project, a pre-configured μC/Probe™ GUI has been realized and can be found inside the project files with the ".wspx" extension. The panel may be configured by the developer to display any variables of interest, which can be represented numerically or in a graphical form such as a linear meter, dial, or cylindrical bar indicator. Numeric values can be scaled to display a percentage or shown in their original form.

The panel shows each individual motor outputs and all remote-control switch readings. Each motor output ranges from -100 to +100 and can be monitored while actively debugging. Furthermore, the switches indicate the current status for lighting effects and arming of the motor outputs.

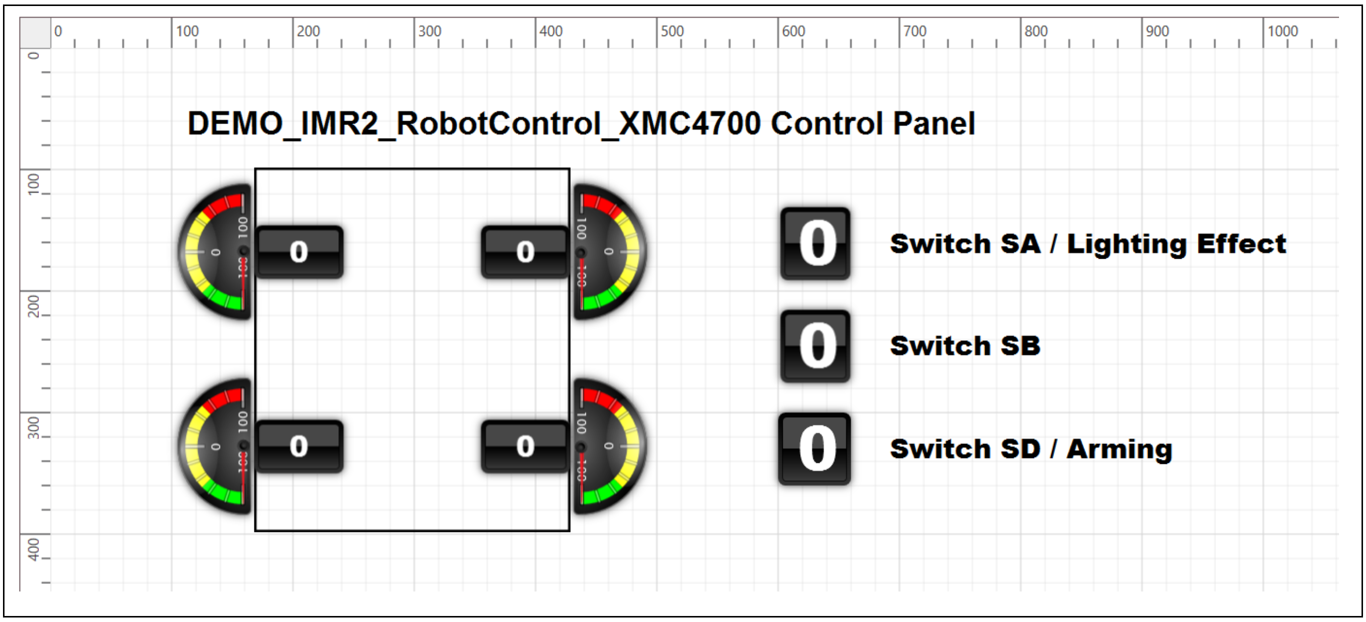



**Figure 22** **Micrium® µC probe monitoring panel**

## 4.3 CAN node identification

After flashing the board with the firmware, the board's CAN node unique identification needs to be set. This can be done by using the 8-pin DIP switch S2, as described in the hardware section.

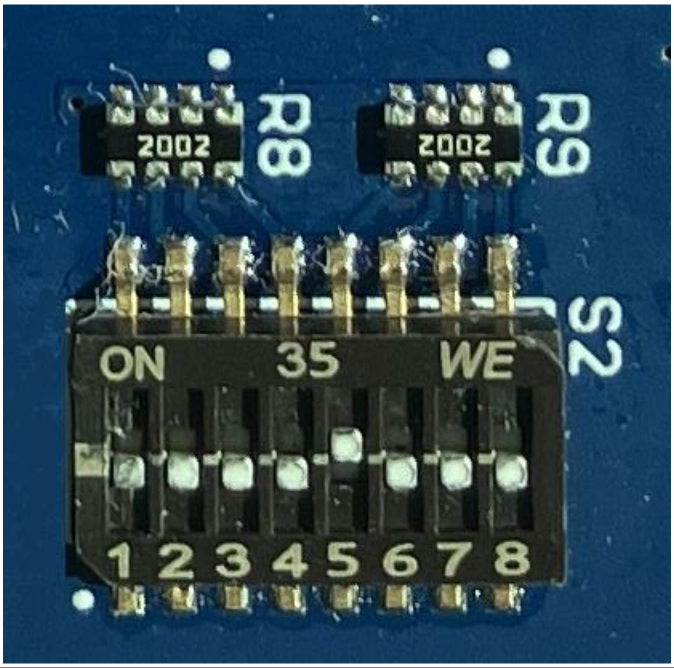



**Figure 23** **DIP switch to set the boards unique CAN node ID**

**Example:** This example describes the result of the following switch position on the CAN node identifier module.

**Table 15 DIP switch S2 – CAN node identifier example 1**

| DIP switch | Pin 8 – MSB | Pin 7 | Pin 6 | Pin 5 | Pin 4 | Pin 3 | Pin 2 | Pin 1 - LSB |
|---|---|---|---|---|---|---|---|---|
| Switch position | – | – | – | – | ON | OFF | OFF | OFF |
| RobotCtrl[3:0] | – | – | – | – | 1 | 0 | 0 | 0 |

In this example DIP switch configuration, the expected CAN node identifier for the DEMO_IMR_MAINCTRL_V1 board would be **24 (DEC)** or **0x18 (HEX)**. This is the result of the board's CAN mask (see Table 8) and the selected switches as seen above. This identifier can be used by other boards present to reach the DEMO_IMR_MAINCTRL_V1 board with CAN messages.

## 4.4 Remote control implementation

The DEMO_IMR_MAINCTRL_V1 board features multiple interfaces to implement an external remote control with either SBUS, IBUS, or PPM. Depending on the selected protocol and the chosen remote-control manufacturer, a different interface might be needed. In this specific case, the FrSky XM+ 16-channel receiver is used in combination with the FrSky Taranis X9D Plus remote control. Due to FrSky being the chosen remote-control manufacturer, the protocol being implemented is the SBUS interface.

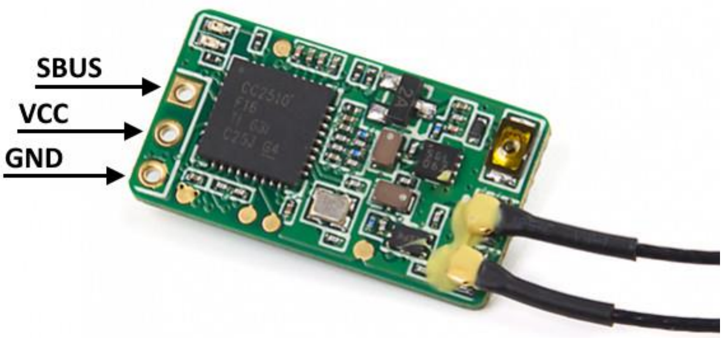



**Figure 24 FrSky XM+ overview with available connectors: SBUS, VCC, and GND**

The SBUS interface is typically only available as an inverted signal that needs to be inverted manually outside of the FrSky receiver hardware. To be able to interface the FrSky XM+ receiver with the XMC4700 micro-controller, we need a non-inverted signal, which can be accessed on the receiver hardware shown in the figure below. The originally inverted SBUS signal is cut-off from the external connector (Pin 1) while the non-inverted signal is extracted and connected to the connector in its place.

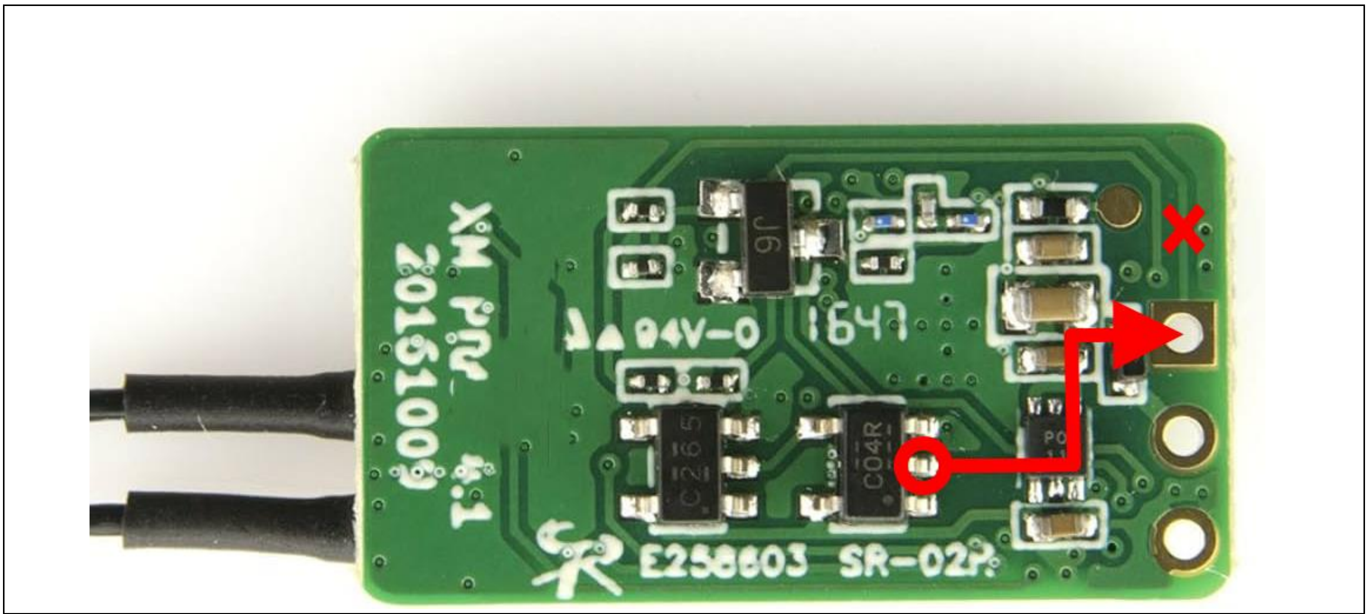



**Figure 25 FrSky XM+ modifications to access the non-inverted SBUS signal**

After applying the necessary modifications to the FrSky XM+ receiver, it can be connected to the DEMO_IMR_MAINCTRL_V1 board. To remove the need of a level shifter, the receiver is powered using the 3.3 V regulated DC output. Furthermore, the SBUS line is connected to the boards receiving UART line (UART_TX) to be able to properly parse the incoming serial communication.

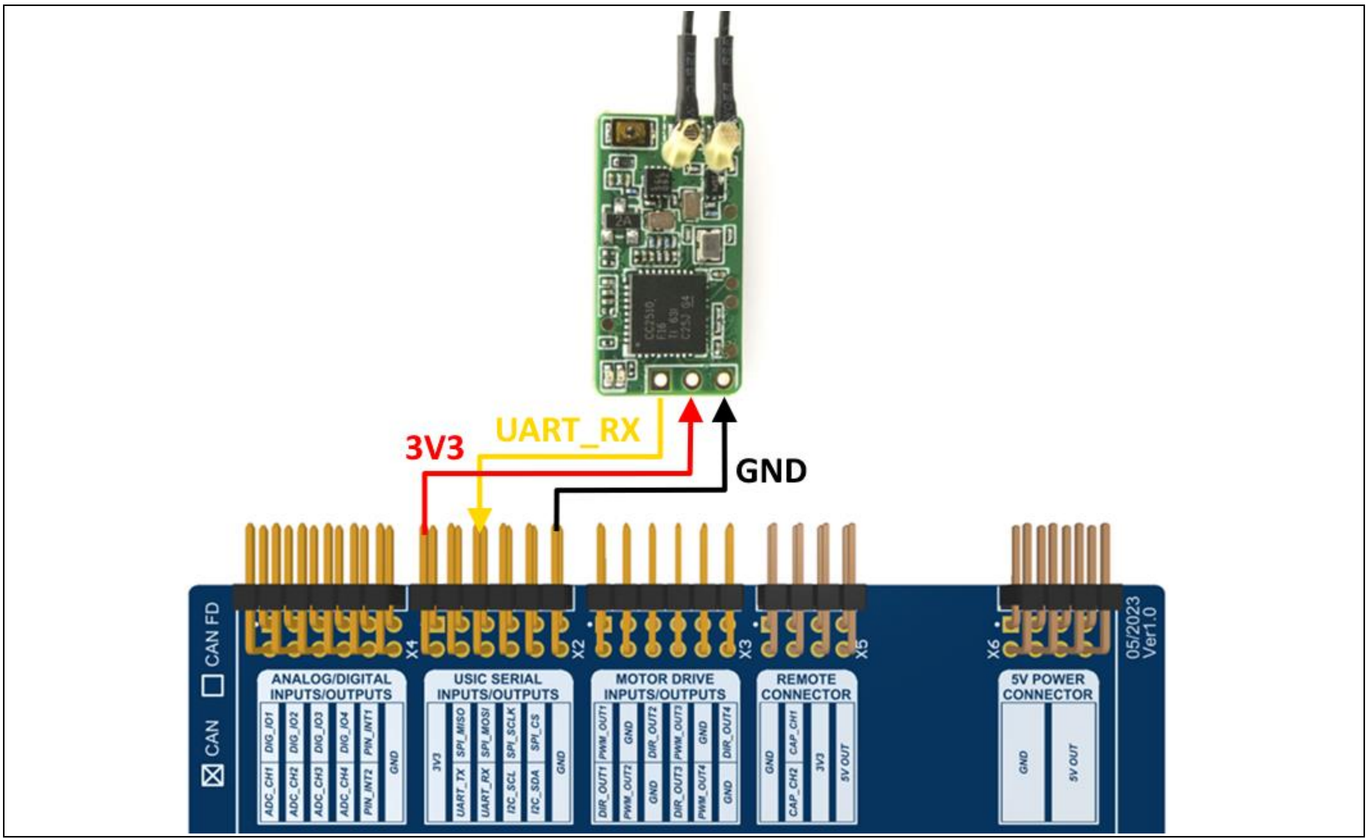



**Figure 26 Wiring diagram to connect the FrSky XM+ with the DEMO_IMR_MAINCTRL_V1 board**

With the wiring completed, the receiver can then be connected to the remote control following the standard RC binding procedure (see Instruction Manual for FrSky XM+). Note that the remote-control mode must be set to D16 with the channel range from CH1 to 16. The selected hardware used to work with the provided firmware is the X9D+ digital telemetry radio system from Taranis, as shown below.


**Figure 27 Taranis X9D+ digital telemetry radio system – remote control**

# 5 Bill of Materials (BOM)

| Designator | Manufacturer | Part number | Quantity | Value/rating |
|---|---|---|---|---|
| C1, C2 | Würth Elektronik | 885012006093 | 2 | 8.2pF/100V/0603/10% |
| C3, C12, C21 | Würth Elektronik | 885012106031 | 3 | 10µF/25V/0603/20% |
| C4, C5, C6, C7, C8, C9, C10, C11, C15, C16, C17, C20, C24, C25 | Würth Elektronik | 885012206071 | 14 | 100nF/25V/0603/10% |
| C13, C14 | Würth Elektronik | 885012006050 | 2 | 6.8pF/50V/0603/10% |
| C18, C19, C23, C26 | Würth Elektronik | 885012206076 | 4 | 1uF/25V/0603/10% |
| C22 | Würth Elektronik | 885012206065 | 1 | 10nF/25V/0603/10% |
| D1 | Würth Elektronik | 150060BS75000 | 1 | Blue/470nm |
| D2 | Würth Elektronik | 150060GS75000 | 1 | Green/525nm |
| G1 | Infineon Technologies | TLS205B0EJV33 | 1 | LDO 3.3V/500mA |
| L1, L2 | Würth Elektronik | 742792651 | 2 | 600R@100MHz 1000mA |
| R1, R2, R5 | Yageo | RC0603FR-074K7L | 3 | 4k7/100mW/0603/1% |
| R3, R4 | Yageo | AC0603JR-071KL | 2 | 1k/100mW/0603/5% |
| R6 | Vishay | CRCW0603510RFK | 1 | 510R/100mW/0603/1% |
| R7 | Yageo | RC0603FR-07120RL | 1 | 120R/100mW/0603/1% |
| R8, R9 | Bourns | CAT16A-2002F4LF | 2 | 20k/63mW/1206/1% |
| S1 | Würth Elektronik | 434121025816 | 1 | WS-TASV Tact Switch |
| S2 | Würth Elektronik | 416131160808 | 1 | WS-DISV 8p DIP Switch |
| U1 | Infineon Technologies | XMC4700-F100K2048 AA | 1 | 32-bit ARM Cortex®-M4 |
| U2 | Infineon Technologies | TLE9351BVSJ | 1 | CAN FD Transceiver |
| X1 | Würth Elektronik | 62701021621 | 1 | WR-BHD 10p 1.27mm |
| X2, X3, X4 | Würth Elektronik | 61301221021 | 3 | WR-PHD 12p 2.54mm |
| X5, X6 | Würth Elektronik | 61300821021 | 2 | WR-PHD 8p 2.54mm |
| X7 | N/A | N/A | 1 | Card edge header |
| Y1 | Würth Elektronik | 830108206909 | 1 | 12MHz/8pF/10ppm |
| Y2 | Würth Elektronik | 830105946101 | 1 | 32.768kHz/7pF/20ppm |

# 6 PCB layout

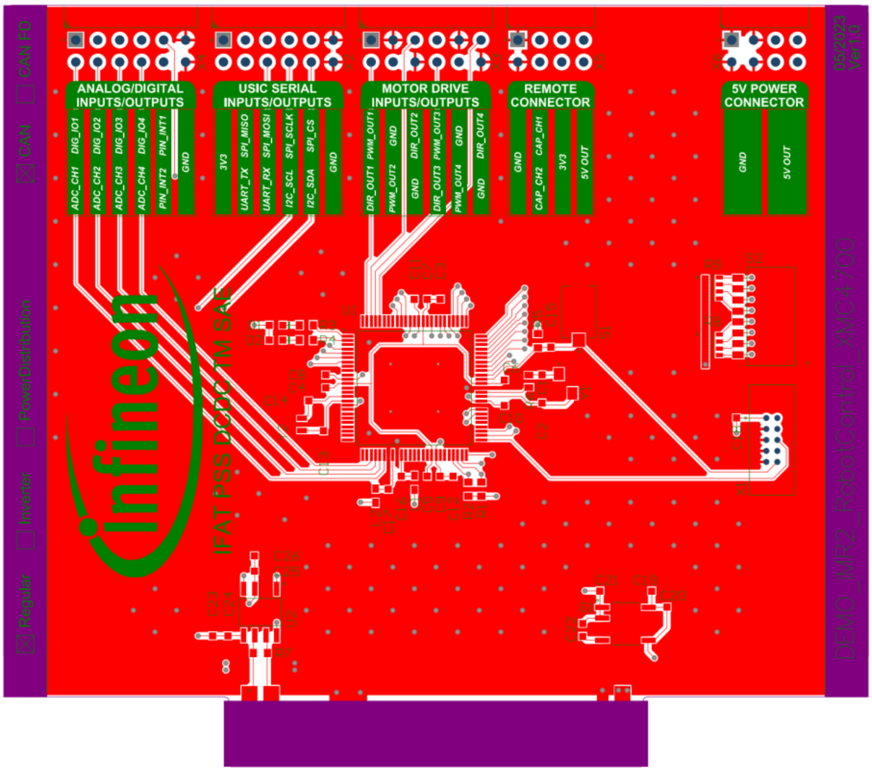



**Figure 28 DEMO_IMR_MAINCTRL_V1 board top side**

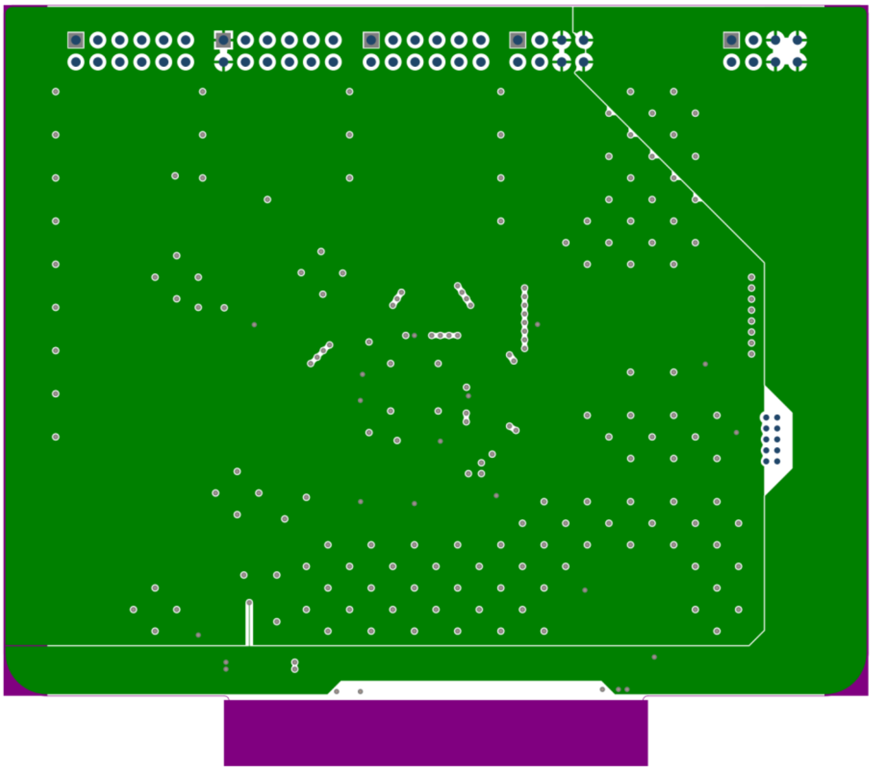


**Figure 29 DEMO_IMR_MAINCTRL_V1 board first inner layer**

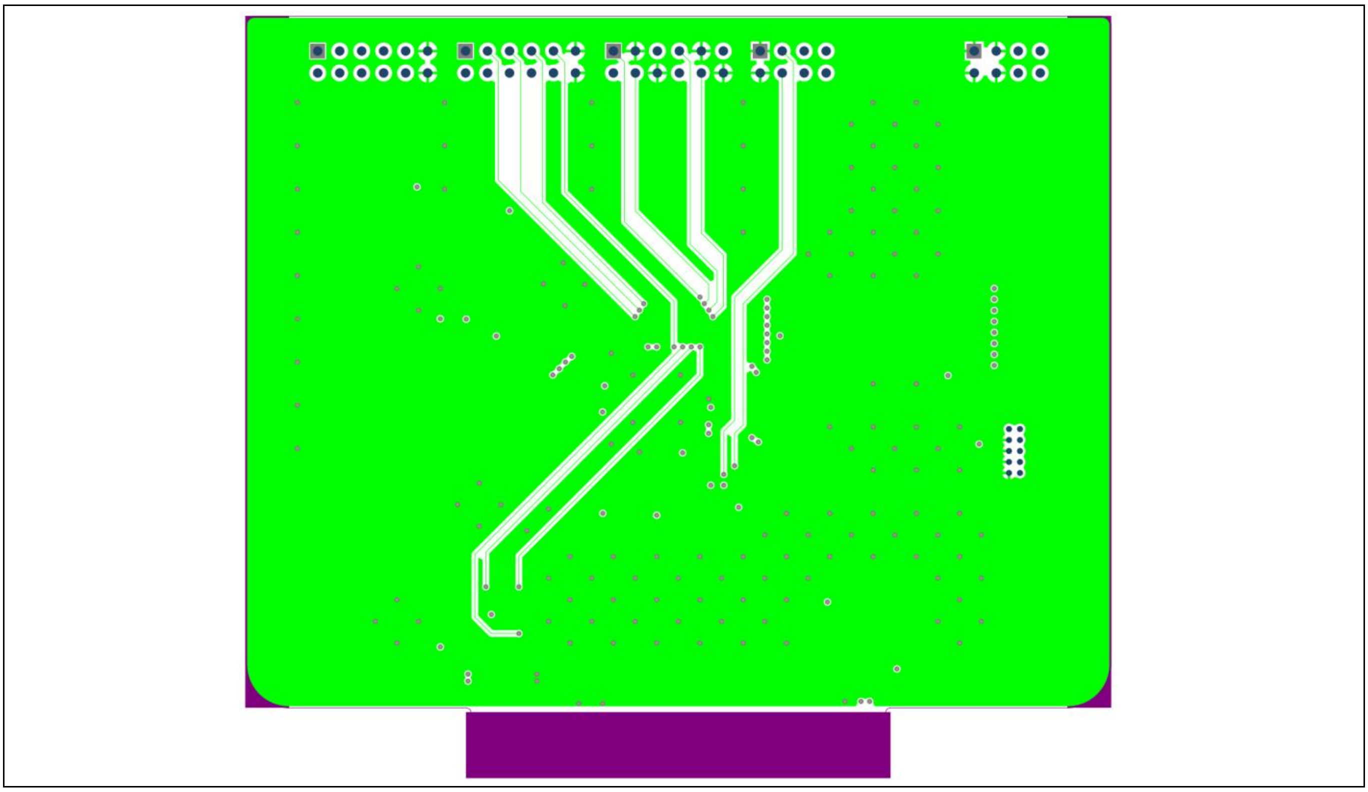


**Figure 30** **DEMO_IMR_MAINCTRL_V1 board second inner layer**

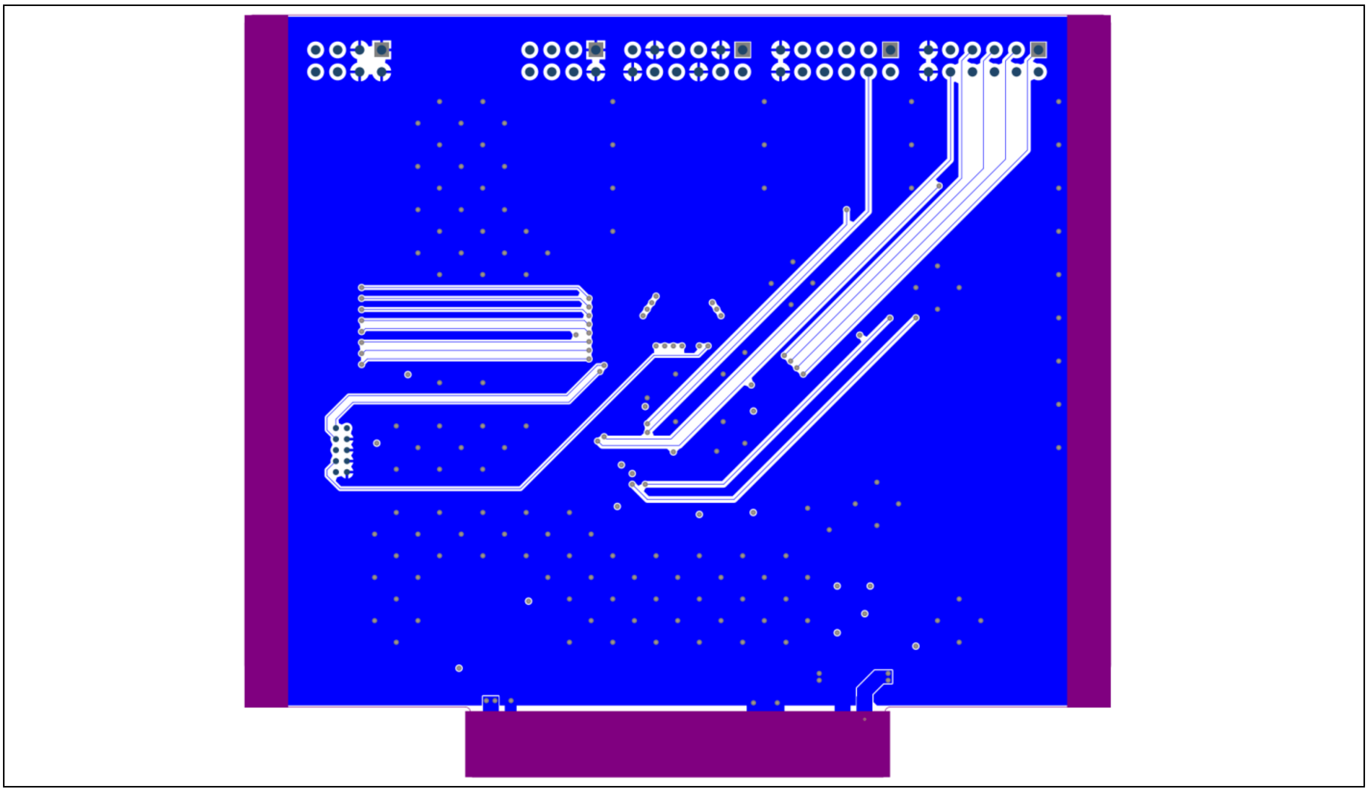


**Figure 31** **DEMO_IMR_MAINCTRL_V1 board bottom side**

# Appendices

## A Schematics

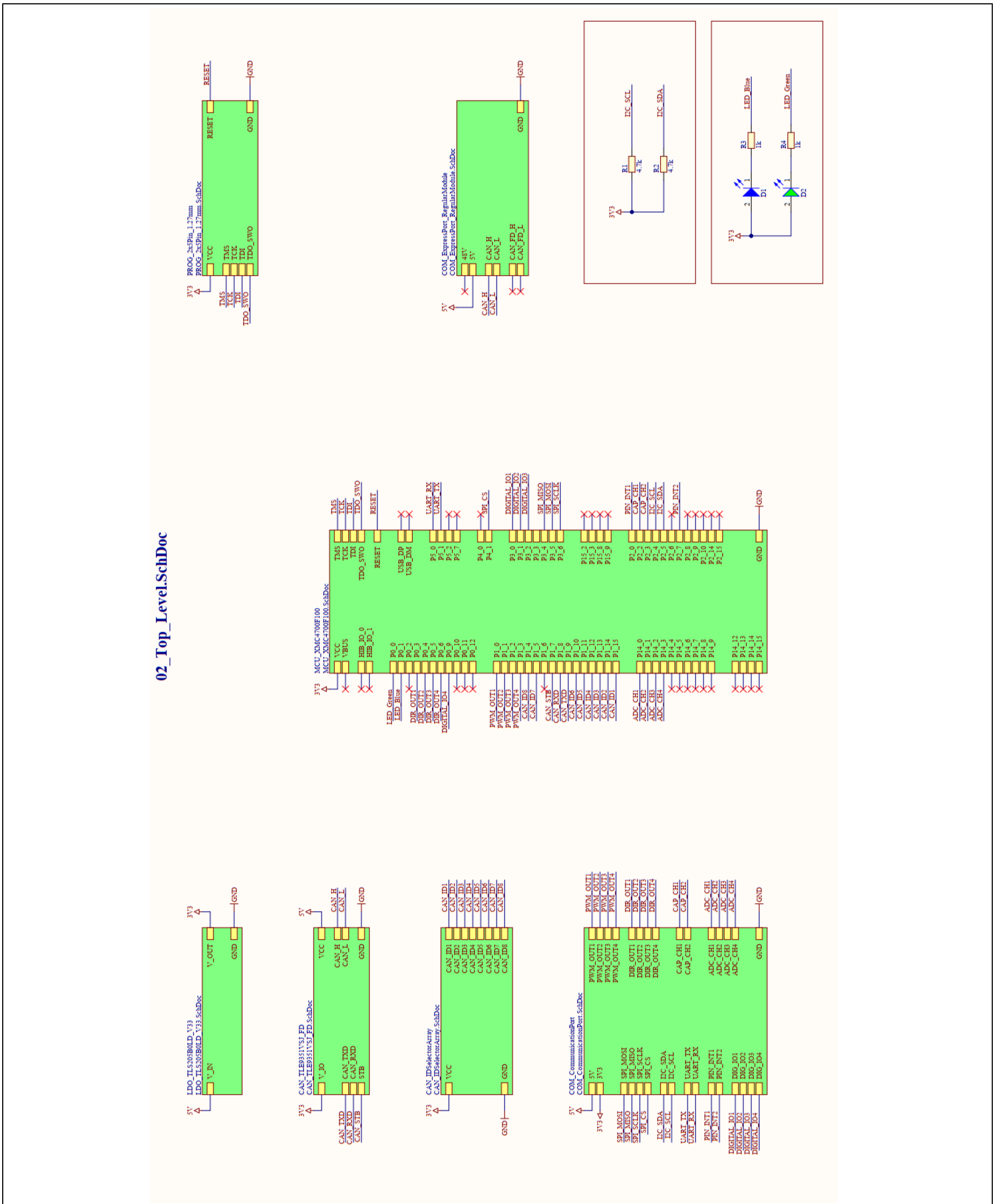


**Figure 32 DEMO_IMR_MAINCTRL_V1 – 02_Top_Level.SchDoc**

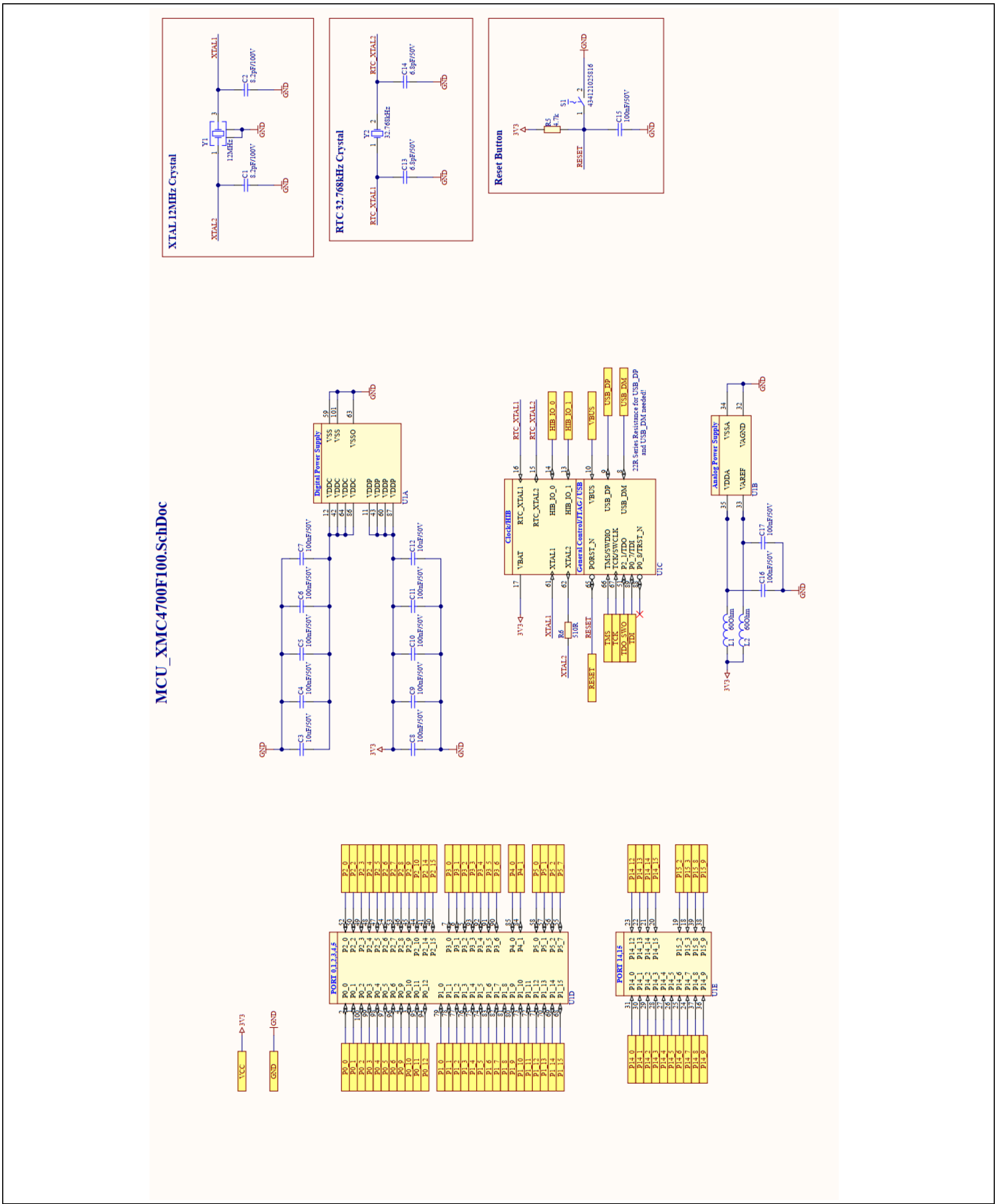



**Figure 33** **DEMO_IMR_MAINCTRL_V1 – MCU_XMC4700F100.SchDoc**

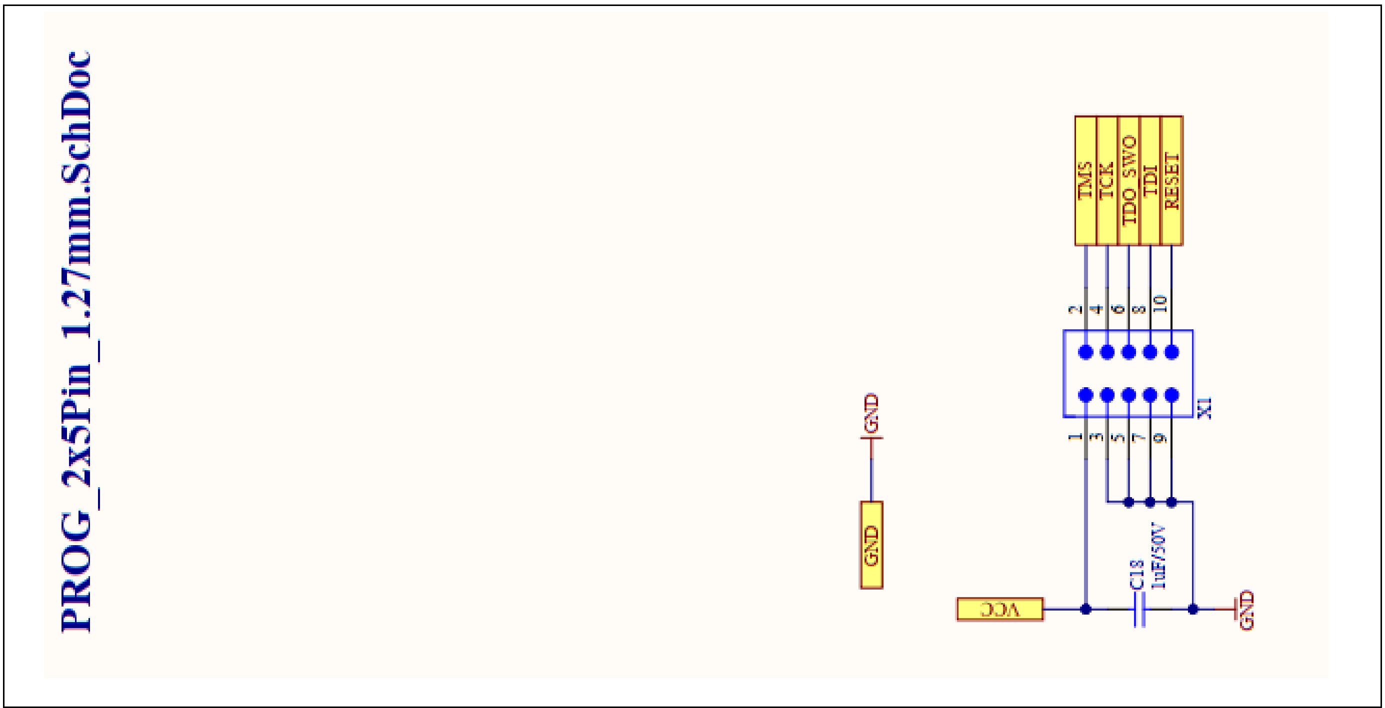



**Figure 34 DEMO_IMR_MAINCTRL_V1 – PROG_2x5Pin_1.27mm.SchDoc**

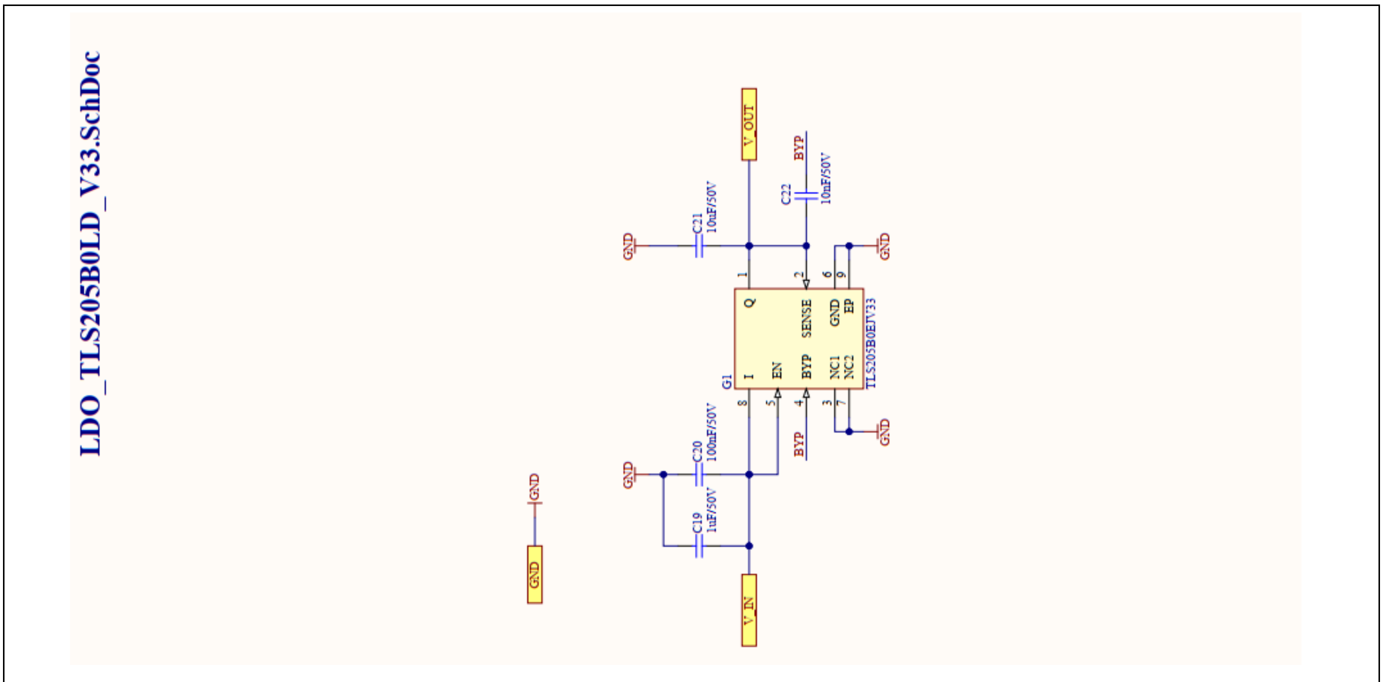



**Figure 35 DEMO_IMR_MAINCTRL_V1 – LDO_TLS205B0LD_V33.SchDoc**

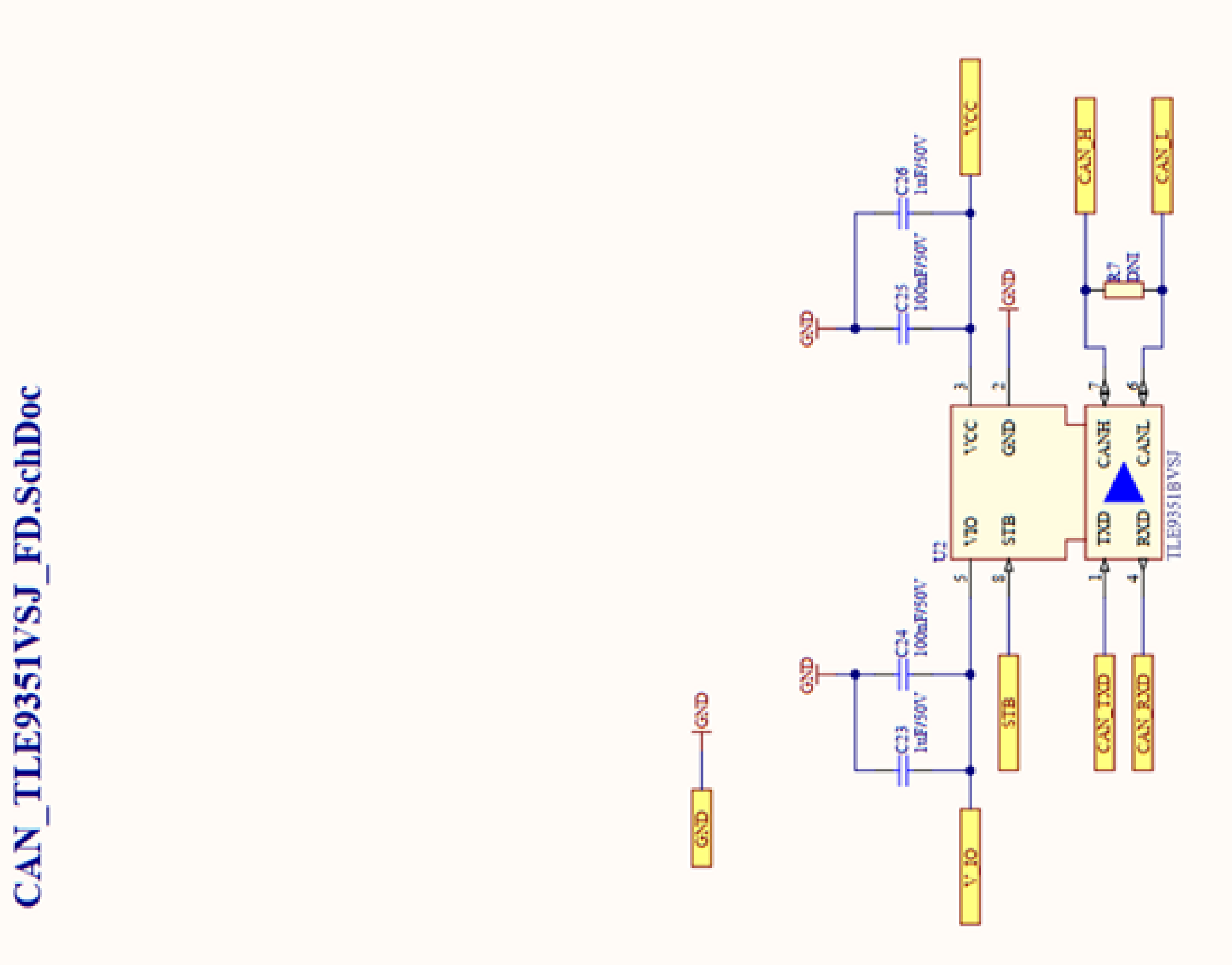



**Figure 36** DEMO_IMR_MAINCTRL_V1 – CAN_TLE9351VSJ_FD.SchDoc

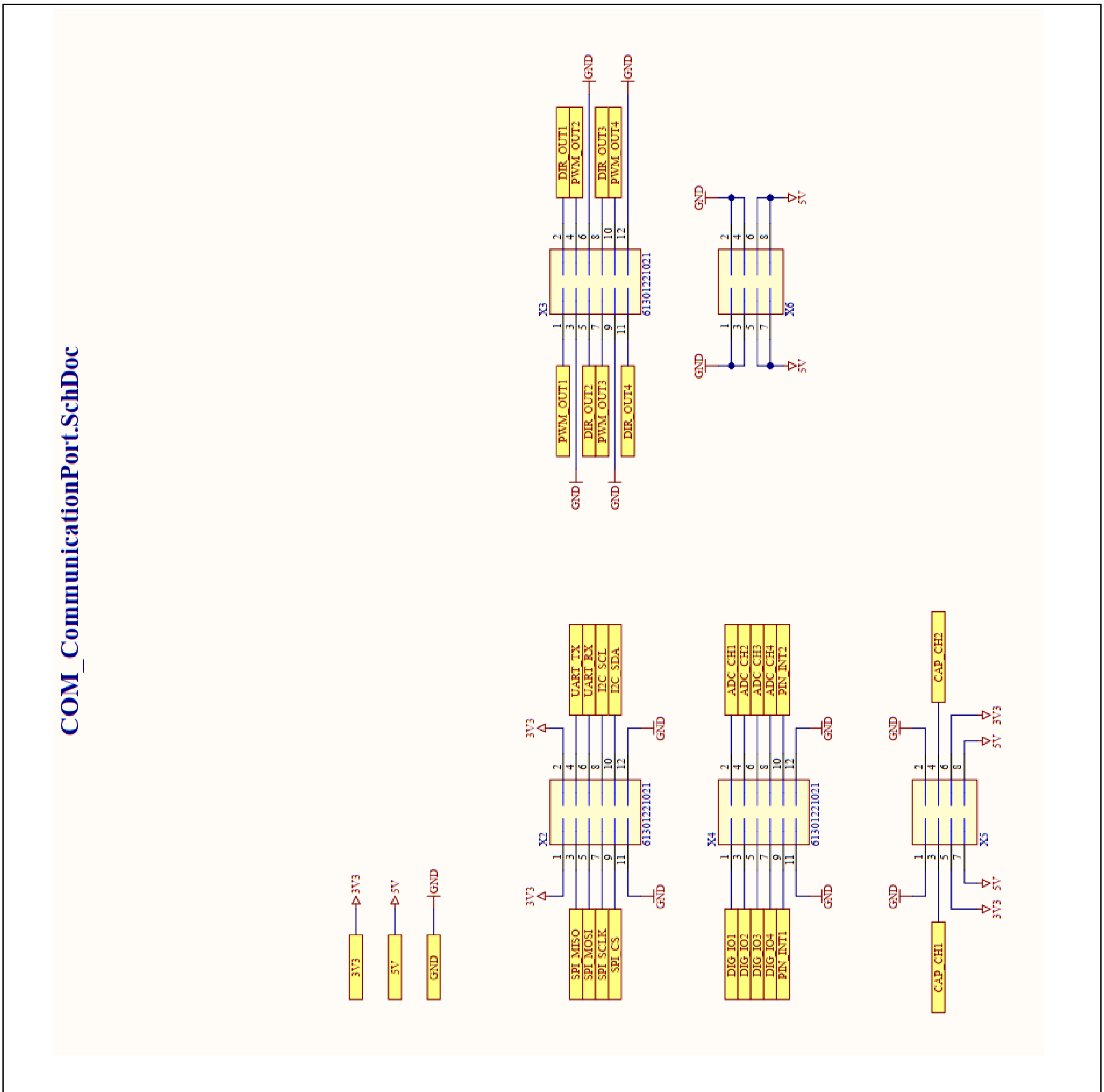


Figure 37 DEMO_IMR_MAINCTRL_V1 – COM_CommunicationPort.SchDoc

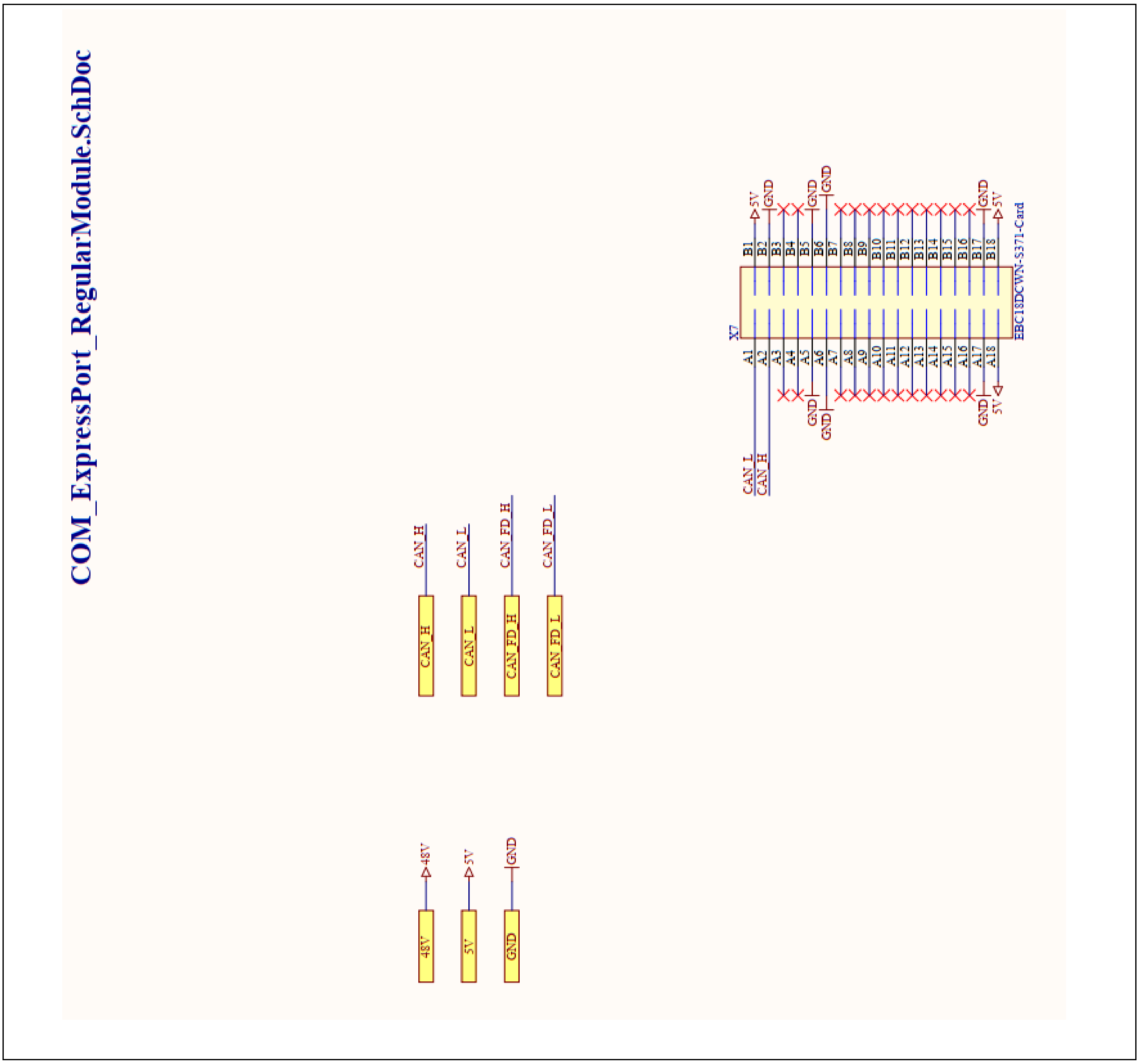



**Figure 38** DEMO_IMR_MAINCTRL_V1 – COM_ExpressPort_RegularModule.SchDoc

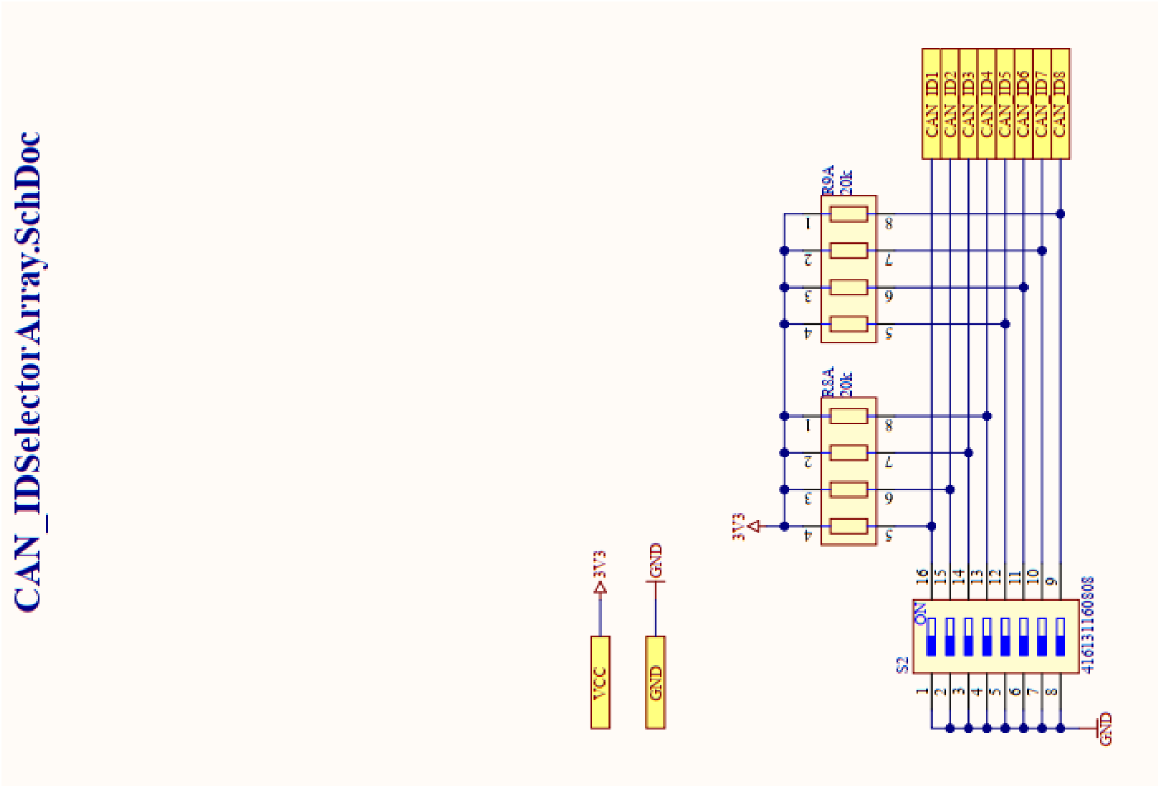


Figure 39 DEMO_IMR_MAINCTRL_V1 – CAN_IDSelectorArray.SchDoc

## References

[1] Infineon Technologies AG. Datasheet (2015-01-15): TLS205B0 Data Sheet. V1.2 www.infineon.com

[2] Infineon Technologies AG. Datasheet Z8F65123481 (2023-08-11): TLE9351BVSJ Data Sheet. V1.0 www.infineon.com

[3] Infineon Technologies AG. XMC4000 Family Overview www.infineon.com

[4] Infineon Technologies AG. Datasheet (2023-04): XMC4700/XMC4800 Microcontroller Series for Industrial Applications XMC4000 Family Data Sheet. V1.2 www.infineon.com

[5] Bolderflight repository, ‘SBUS’, GitHub, 29 September 2022, Online [accessed 22 November 2023]

[6] Wikipedia contributors, 'Mecanum wheel', Wikipedia, The Free Encyclopedia, 24 July 2023, 12:15 UTC, Online [accessed 2 October 2023]

[7] Canis Automotive Labs, ‘The canframe.py tool’, 25 October 2020, Online [accessed 22 November 2023]

## Revision history

| Document revision | Date | Description of changes |
|---|---|---|
| 1.0 | 2024-04-08 | Initial version |

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

**Edition 2024-04-08**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**



**Do you have a question about this document?**

**Email:** erratum@infineon.com

**Document reference**

**UG090902**

**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.