Infineon Mobile Robot (IMR) motor control

Using DEMO_IMR_MTRCTRL_V1 and DEMO_ANGLE_SENS_V1

About this document

Scope and purpose

This document describes the functions and usage of the DEMO_IMR_MTRCTRL_V1 board together with the DEMO_ANGLE_SENS_V1 board. DEMO_IMR_MTRCTRL_V1 board can drive a brushless DC (BLDC) motor using Hall or encoder-based sensors and completely sensorless. This proposed design is intended to be used for mobile robotic applications like automated guided vehicles (AGV) or autonomous mobile robots (AMR) in the low speed regime focusing on a modular design approach.

The selected communication interface is CAN, where speed and direction commands are received. The board utilizes the MOTIX™ IMD701A fully-programmable motor controller, combining microcontroller and gate driver IC in combination with OptiMOS™ 6 ISZ053N08NM6 power MOSFETs. The IMD701A integrates a fully-programmable XMC1404 Arm® Cortex®-M0 microcontroller from Infineon XMC1400 family with 6EDL7141, a 60 V three-phase smart gate driver with integrated power supply. Integrated devices allows an ultra-compact design for drives applications up to 60 V including not only the microcontroller and a flexible three-phase gate driver, but also the complete power supply required in the system (synchronous buck converter and LDO), three current sense amplifiers (CSAs), protections, and a set of configurations to adjust to specific needs. Additionally, the used angle sensor solution is described.

Intended audience

The document is addressed to design engineers, technicians, and developers in the field of robotics and battery-powered motor drives who strive for highly integrated and efficient solutions.

Infineon components featured

- OptiMOS™ 6 ISZ053N08NM6 80 V/5.3 mΩ PG-TSDSON-8 FL
- MOTIX™ IMD701A-Q064X128AA fully-programmable motor controller combining microcontroller and gate driver IC
- TLE9351VSJ high-speed CAN FD transceiver
- BAS52-02V 45 V Silicon Schottky diode with low forward voltage at 200 mA
- XENSIV™ TLI5012B E1000 angle sensor with Incremental Interface (IIF)

V 2.0
2025-04-24

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**

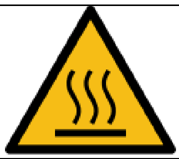
	**Warning:** *The DC link potential of this board is up to 1000 VDC. When measuring voltage waveforms by oscilloscope, high-voltage differential probes must be used. Failure to do so may result in personal injury or death.*
	**Warning**: *The evaluation or reference board contains DC bus capacitors, 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 have become a firmly established part of day-to-day operations. It is used not only in logistic and warehouse centers, production sites but also in hospitals, restaurants, schools, or as last-mile delivery vehicles of packages and goods.

Many types of mobile robots exist, but on a high-level, the two main types are:

- **Automated guided vehicles (AGVs):** AGVs are fixed, they follow predefined paths using lasers, barcodes, radio waves, vision sensors, or magnetic tape for navigation
- **Autonomous mobile robots (AMRs):** AMRs are not fixed and do not need external paths as they utilize autonomous mapping, localization, navigation, and obstacle avoidance by using sensors

Both types have in common that the operating speed of the robot is low and therefore have similar requirements in terms of motor control and drive. Usually, robots are battery-powered whereas 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). The IMR is a comprehensive robotic platform intended to be used with a wide variety of boards (sensors, motor controls, wireless communication, battery management, etc.).

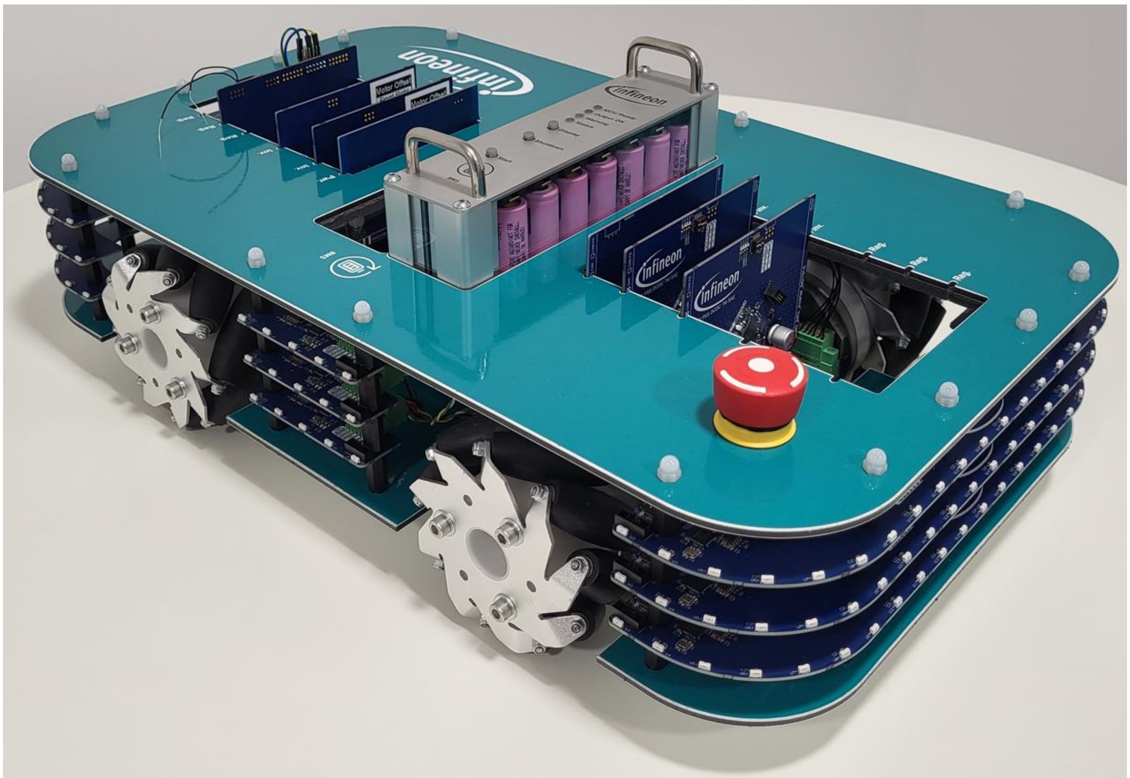


Figure 1 Isometric view of the IMR

The overall target of the IMR is to provide a demonstration platform for autonomous service robot functionalities applying Infineon components. This document describes the DEMO_IMR_MTRCTRL_V1 board to directly drive the BLDC motor in the IMR and DEMO_ANGLE_SENS_V1 board to provide the angular position of the rotor in the motor in such a way to provide a more accurate motor control especially for a very low speed.

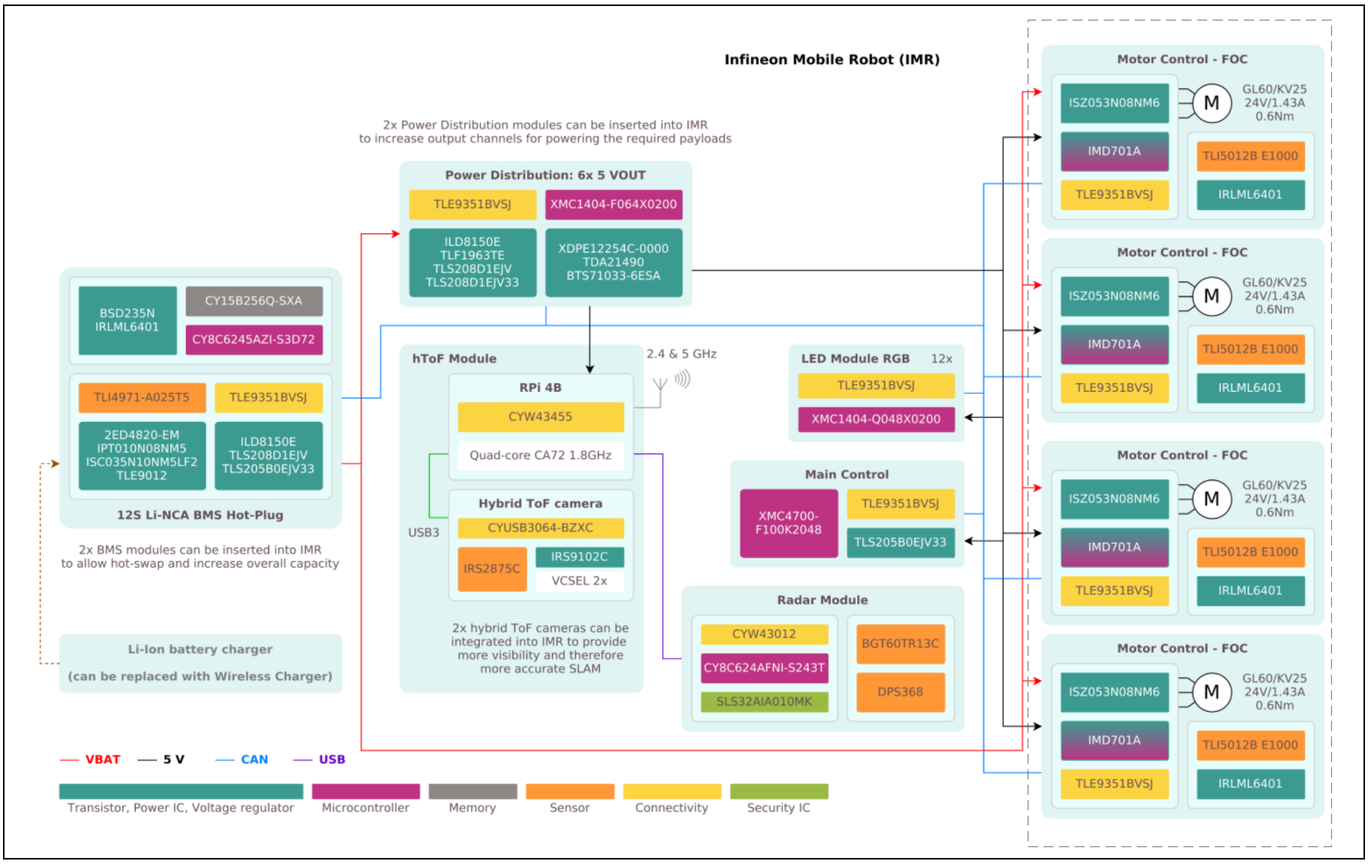


Figure 2 **IMR overview**

The board described in this document is motor control–FOC (highlighted) as shown in Figure 2. Four motor control boards are installed in IMR to drive the four wheels of IMR.

1.3 DEMO_IMR_MTRCTRL_V1

This document provides a detailed description of the DEMO_IMR_MTRCTRL_V1 board for mobile robot applications. The board is designed to drive either one BLDC or stepper motor, with multiple angle sensor options including encoder and Hall-based. See the simplified system block diagram in Figure 4. The intended voltage supply is battery-powered using a 12-cell Lithium-ion configuration but operation with a reduced count is also possible. However, the power level will be reduced.

The board can be interfaced via CAN and receive the speed information. Latter can be used as a reference input for the speed loop control. Several boards can be used on the same bus with identical software as the CAN address can be changed via a dual in-line package (DIP) switch.

Several different BLDC and stepper motors can be used with this board. In this document, the GL60 out-running gimbal motor is used as an example. The GL60 motor consists of 14 pole pairs, making low-speed rotation without a gear box. In addition to a BLDC motor, an external J-Link debugger is needed to flash the software (SW) into the microcontroller.

The form-factor of the board is selected due to the modular concept used in the IMR. The card edge connector is used to interface with the motherboard of the IMR.

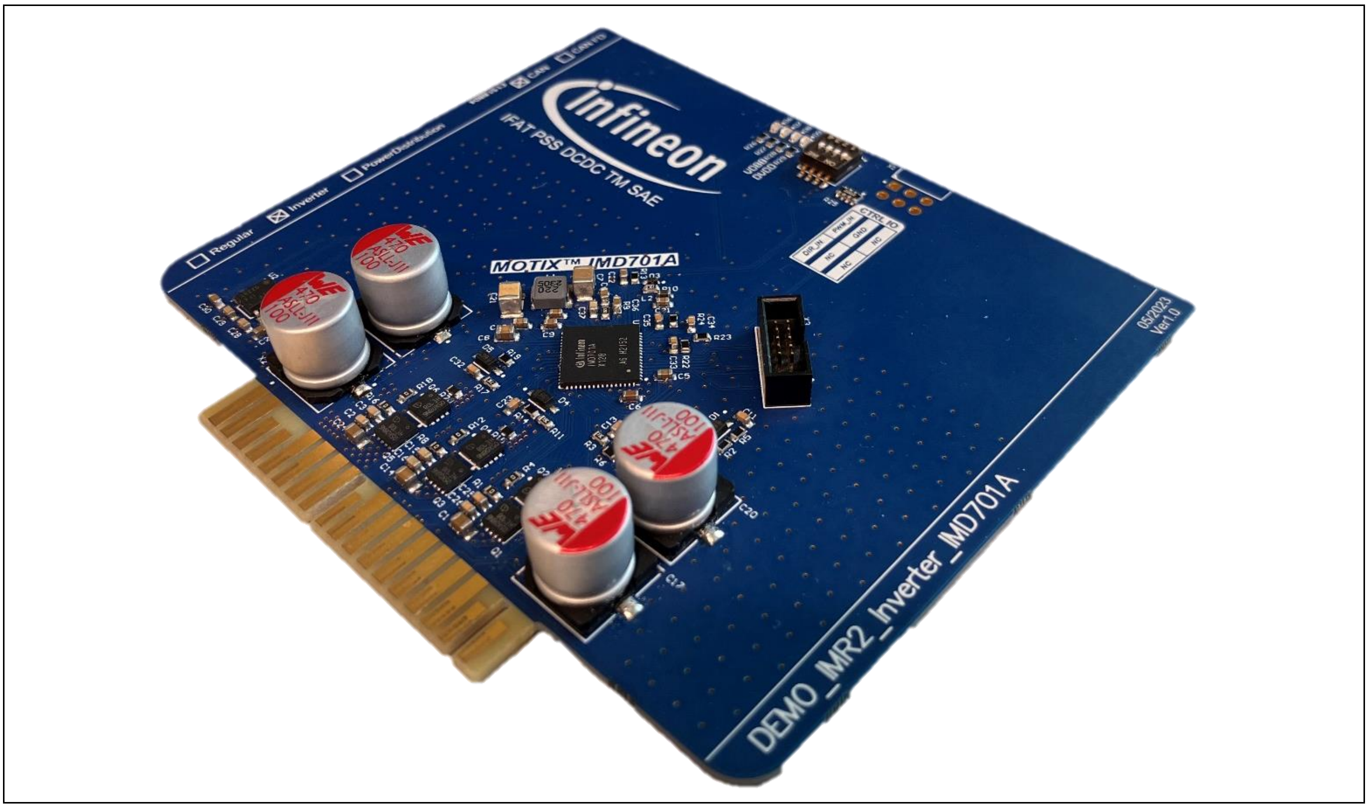


Figure 3 DEMO_IMR_MTRCTRL_V1 board

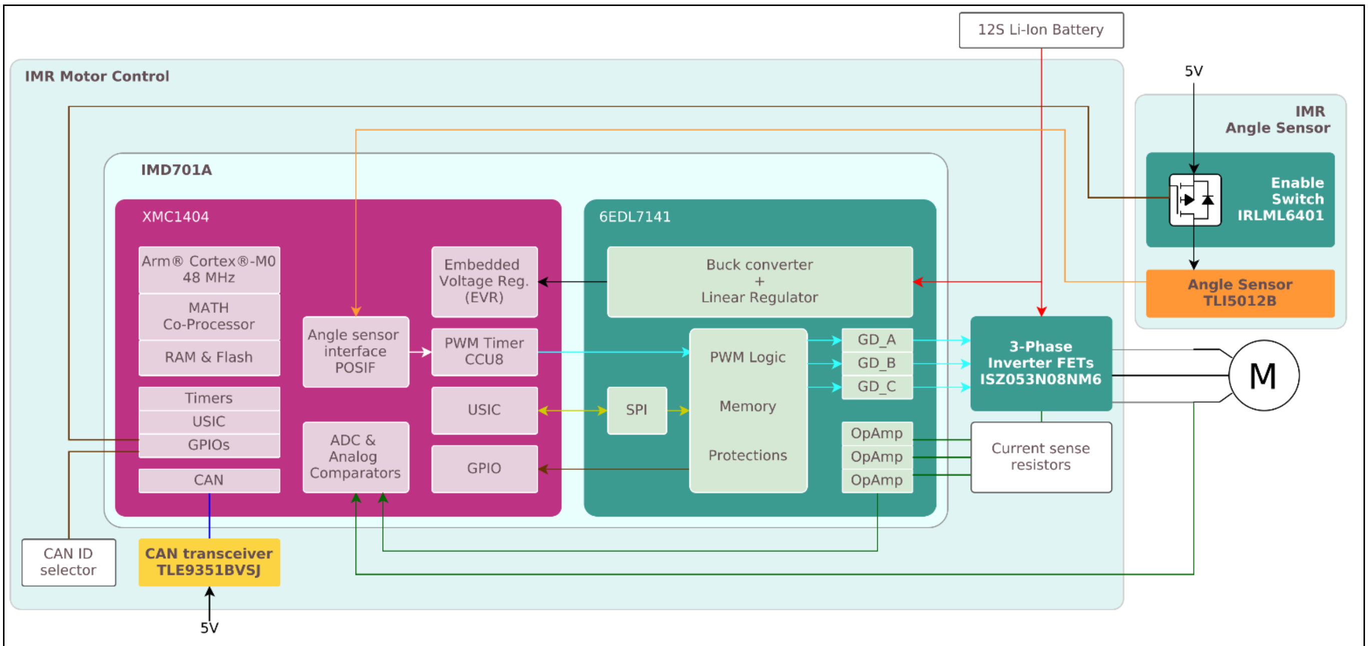


Figure 4 DEMO_IMR_MTRCTRL_V1 block diagram

2 Specifications

Input and output at normal operation

- DC input voltage 18 to 60 V, nominal 48 V
- Maximum input current 10 A
- Output voltage three-phase field-oriented control (FOC)
- Maximum output current per phase 5 A_{RMS}
- Maximum output continuous power 150 W

Control scheme

- Sensored/sensorless FOC with speed control (the document focus is encoder-based)
- Switching frequency 20 kHz
- Three current shunts

Protection features

- Overcurrent protections (OCP)
  - DVDD linear regulator
  - Buck converter
  - Motor leg shunt OCP
- Undervoltage lockout (UVLO) protection
  - Gate driver supply voltage both high side and low side drivers
  - Supply voltage PVDD
  - DVDD linear regulator output voltage
  - Buck converter output voltage
- DVDD linear regulator overvoltage lockout (OVLO) protection
- Overtemperature shutdown (OTS) and warning (OTW)

Maximum component temperature

In an ambient temperature of 30°C, the maximum allowed component temperatures are as follows:

- Resistors less than 100°C
- Ceramic capacitors, film capacitors, and electrolytic capacitors less than 100°C
- MOSFET transistors and diodes less than 100°C
- ICs less than 100°C

Dimensions of the evaluation board

Width 100 mm, length 88.09 mm, and height 12.25 mm.

Note: *To operate the board, configure the software correctly for the specific motor being driven. This requires motor parameters such as phase-winding inductance and resistance to be entered into depending on either the motor control GUI or the software before flashing it to the target board.*

Attention: ***The board must be tested only by qualified engineers and technicians.***

3 DEMO_IMR_MTRCTRL_V1 Schematics

The main component on the board is the IMD701A-Q064X128-AA, see Figure 5, which is a highly integrated chip consisting of a:

- Fully-programmable drives optimized Arm® Cortex®-M0 microcontroller (XMC1404 at 48 MHz main clock) with additional MATH co-processor (96 MHz)
- Three-phase smart gate driver: 1.5 A sink/ 1.5 A source peak gate driver currents
- Three current sense amplifiers with integrated gain and offset generation
- Integrated synchronous buck converter controller and LDO for complete BLDC system supply

This allows for a compact design with highly reduced BoM and complexity. There is only a minimum of external components necessary which include several buffer capacitors as well as one external inductor for the internal synchronous buck converter.

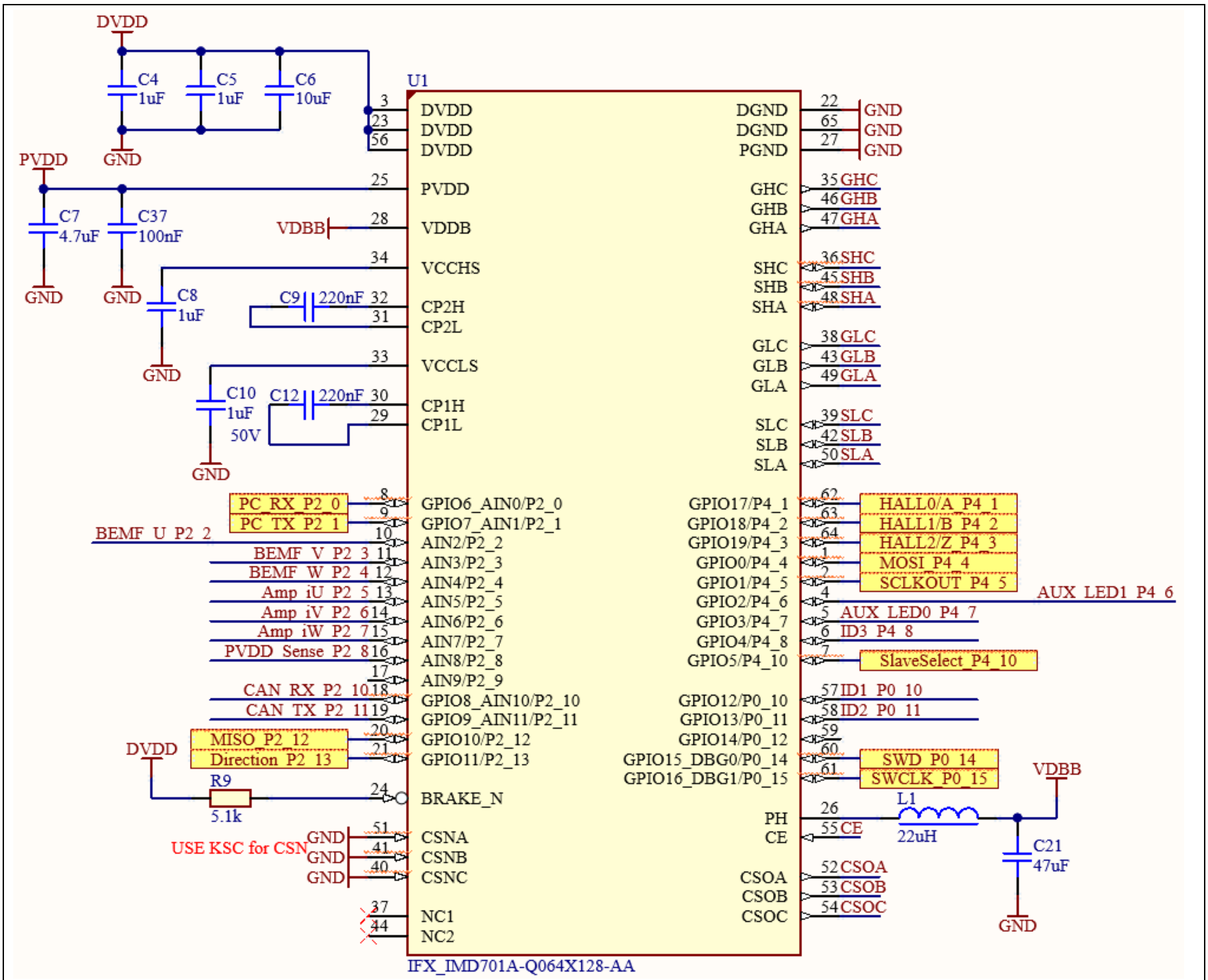


Figure 5 Schematic around the IMD701A motor controller

The board can be controlled via a CAN interface. The CAN transceiver used is the TLE9351VSJ, which is a high-speed CAN transceiver used for automotive and industrial applications. It can be supplied on in- and output with different voltage levels. This allows the IMD701A motor controller to be used directly with TLE9351VSJ CAN transceiver without any level shifter. As the board is used with other boards on the same CAN bus, there is the option to change the CAN address via the mounted DIP switch (S1). This allows the same software to be flashed onto multiple boards situated on the same bus. Note that in the software only ID1 and ID2 are in use.

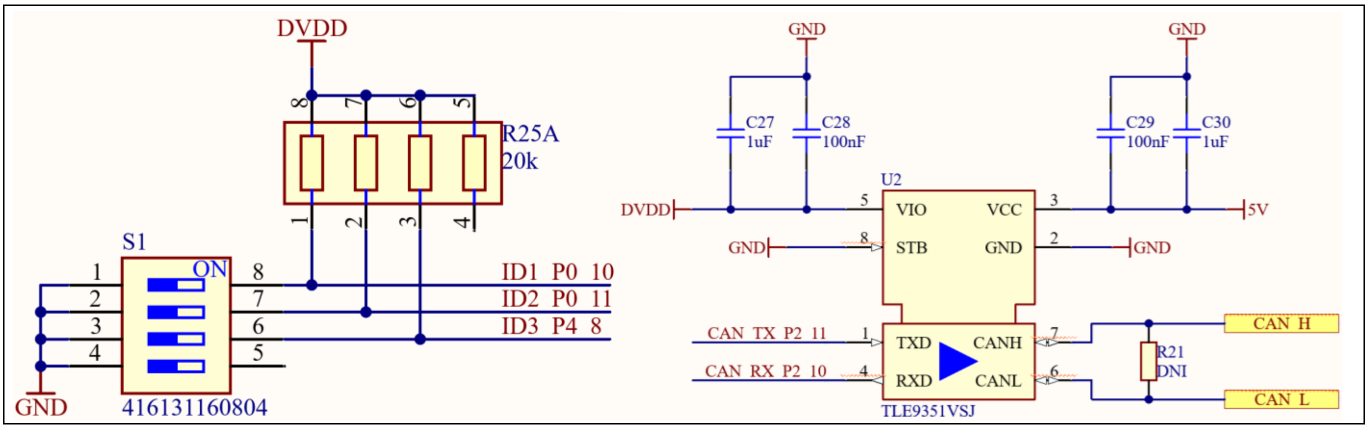


Figure 6 DIP switch for CAN address setting and TLE9351VSJ CAN transceiver

The three-phase inverter itself consists out of six OptiMOS™ 6 ISZ053N08NM6 80 V/5.3 mΩ switches in a compact PG-TSDSON-8 FL package. Each phase output is connected to the card edge connector to be fed to the motor as well as to a voltage divider if back EMF information is required. Furthermore, each half-bridge possesses their own buffer capacitors as close as possible to the switches. All gates are connected via 0 Ω resistors to the IMD701A which are only used for debugging and measurement purposes. The gate charge current itself can be programmed onto the IMD701A making the use of external gate resistors abundant.

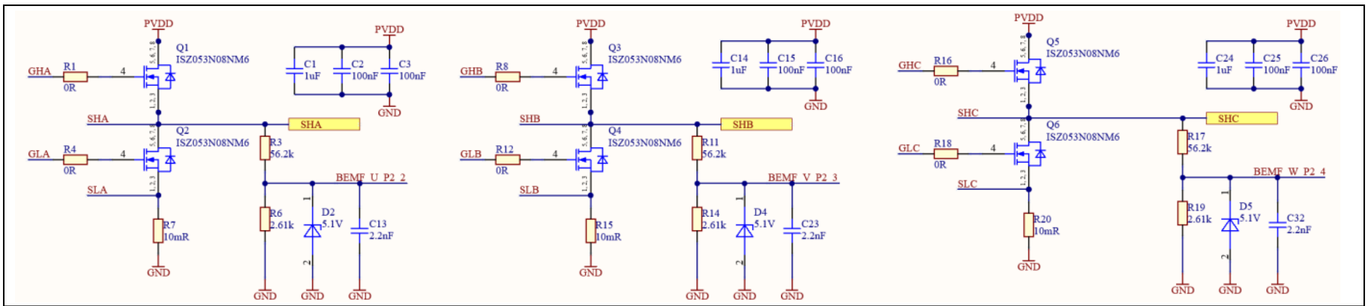


Figure 7 Three-phase inverter utilizing the OptiMOS™ 6 ISZ053N08NM6 80 V/5.3 mΩ PG-TSDSON-8 FL package

The connection to the board can happen via two interfaces, the 10-pin debugging interface for programming and the card edge header. Latter is used for supplying the board with the battery voltage (nominal 48 V) and carrying the phase output voltages, angle sensor pins, and CAN interface. This header layout is standardized so that the board can be interchanged if another inverter solution is chosen.

The card edge header on the PCB should be designed based on the receptacle connector EBC18DCWN-S371 from Sullins Connector Solutions.

Table 2 Pin out of inverter 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	Reserved for CAN_FD_L	B3	Reserved for CAN_FD_L
A4	Reserved for CAN_FD_H	B4	Reserved for CAN_FD_H
A5	GND	B5	GND
A6	GND	B6	GND
A7	Battery voltage	B7	Battery voltage
A8	Battery voltage	B8	Battery voltage
A9	Reserved for Stepper B-	B9	Reserved for Stepper B-
A10	W/Stepper B+	B10	W/Stepper B+
A11	V/Stepper A-	B11	V/Stepper A-
A12	U/Stepper A+	B12	U/Stepper A+
A13	A/TTL U	B13	B/TTL V
A14	C(Z)/TTL W	B14	MOSI
A15	MISO	B15	CS
A16	CLK	B16	No connection
A17	Reserved for GND	B17	Reserved for GND
A18	Reserved for 5 V	B18	Reserved for 5 V

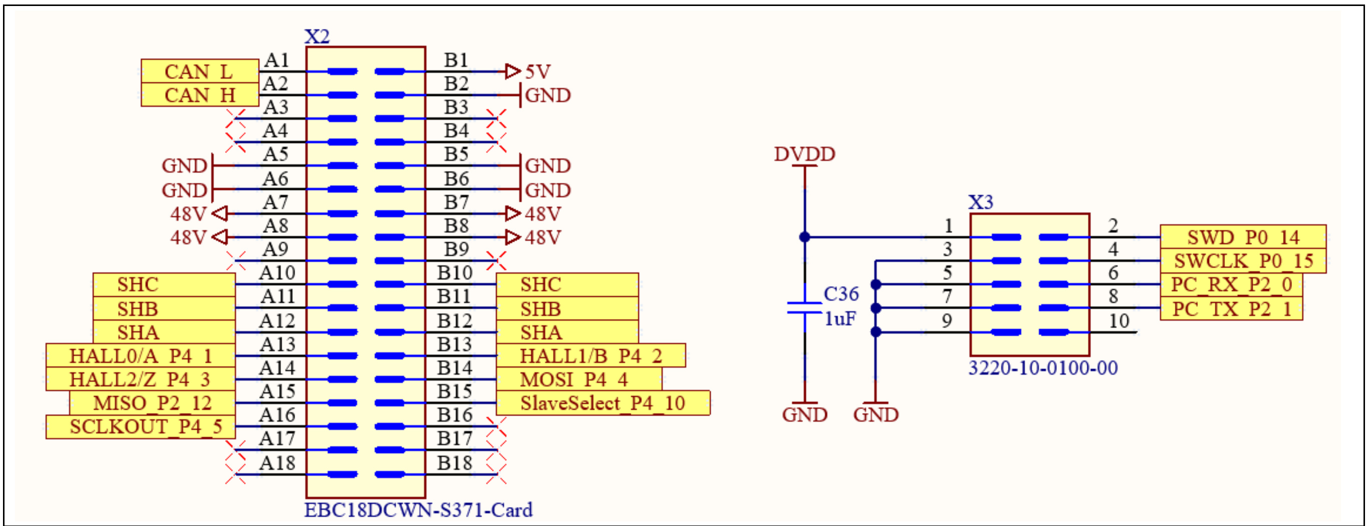


Figure 8 Card edge header and 10-pin debug connector

The IMD701A motor controller has three internal amplifiers for phase current sensing. The gain can be adjusted in the software by following values: 4, 8, 12, 16, 20, 24, 32, and 64 and an external filter network is populated on the board. For more details, see the IMD70xA datasheet [1]. Furthermore, the input voltage to the three-phase inverter is measured and can be used to tune the modulation index in the Space Vector Pulse Width Modulation (SVPWM).

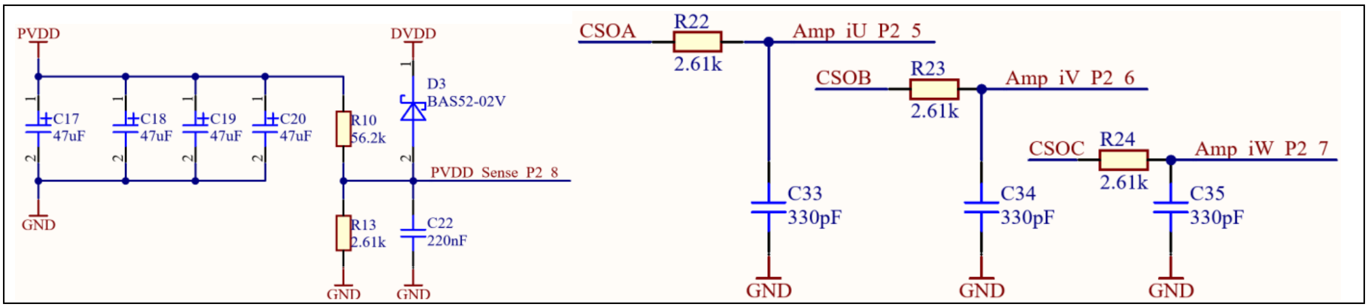


Figure 9 PVDD and phase current sensing filters

Furthermore, on the board several signal LEDs are populated where D8 signals for the presence of the gate drive voltage and D9 the voltage of the internal voltage regulator. D6 and D7 are controlled via the software, where D6 indicates a received CAN message and D7 has no function until now.

Additionally, on board, the chip enable of the IMD701A needs to be set to high as soon as the battery voltage is supplied.

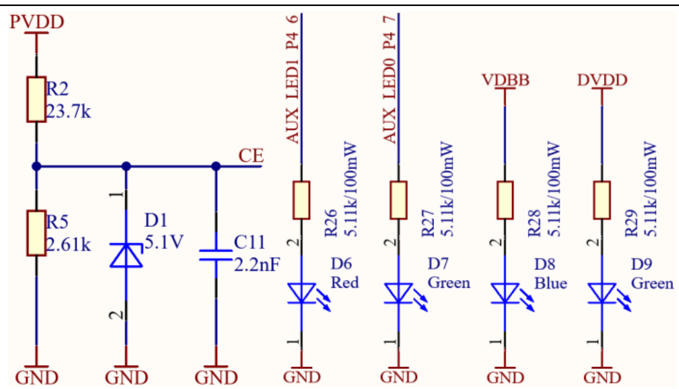


Figure 10 Chip enable and signal LEDs

Figure 11 shows all areas with their respective functions on the board. The board itself is a four-layer board with a size of 100 mm times 88.09 mm. The shape and size of the board is standardized so that it can be easily replaced by a board with the same functionality. This enables a wide testing and showcase environment to compare and demonstrate various solution options.

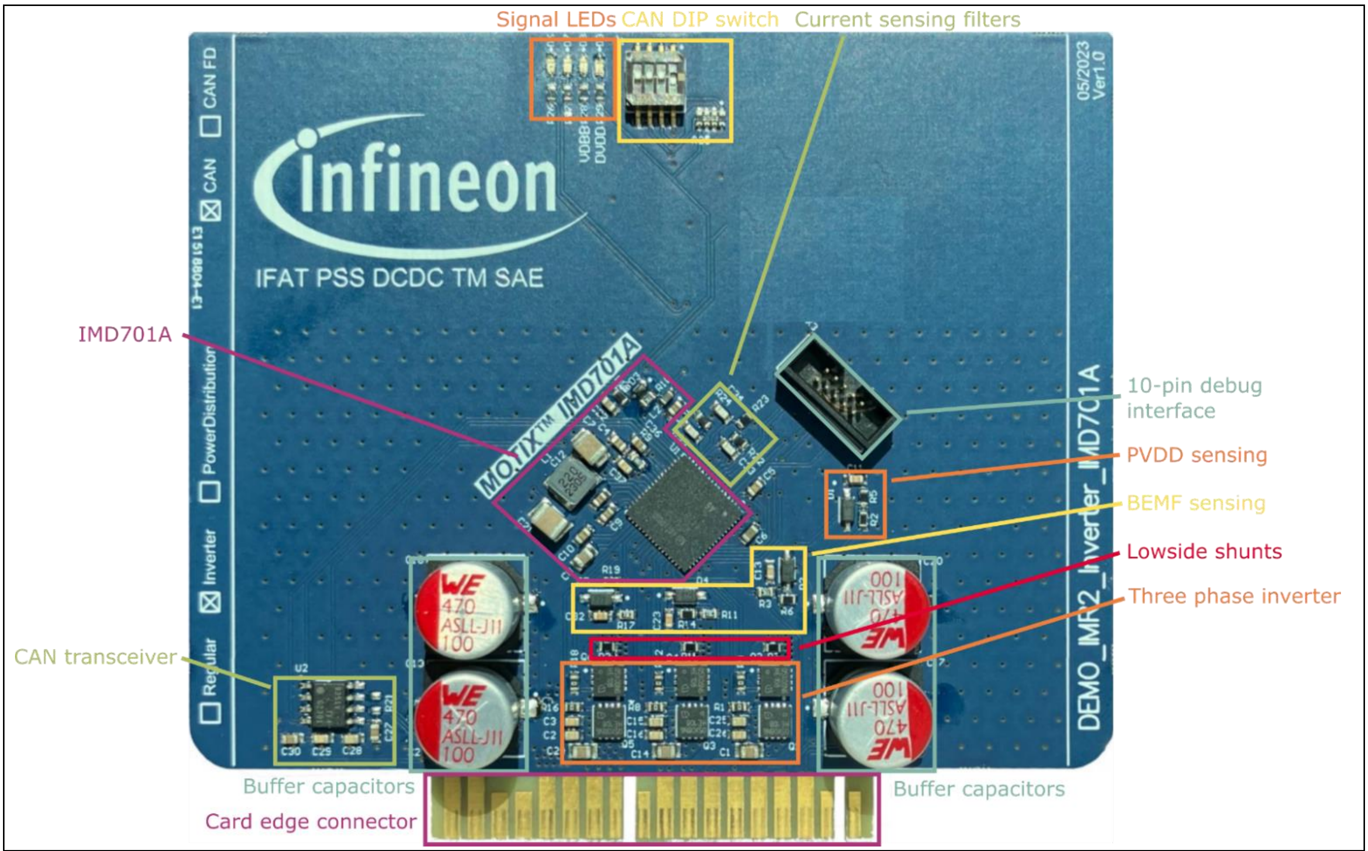


Figure 11 DEMO_IMR_MTRCTRL_V1 with all function blocks

4 DEMO_IMR_MTRCTRL_V1 PCB layout

The DEMO_IMR_MTRCTRL_V1 board utilizes a four-layer PCB with 1 oz. Copper on the top and bottom layers and 1 oz. Copper on the internal layers. Components are mounted on the top and bottom sides. The width is 100 mm, and the length is 88.09 mm.

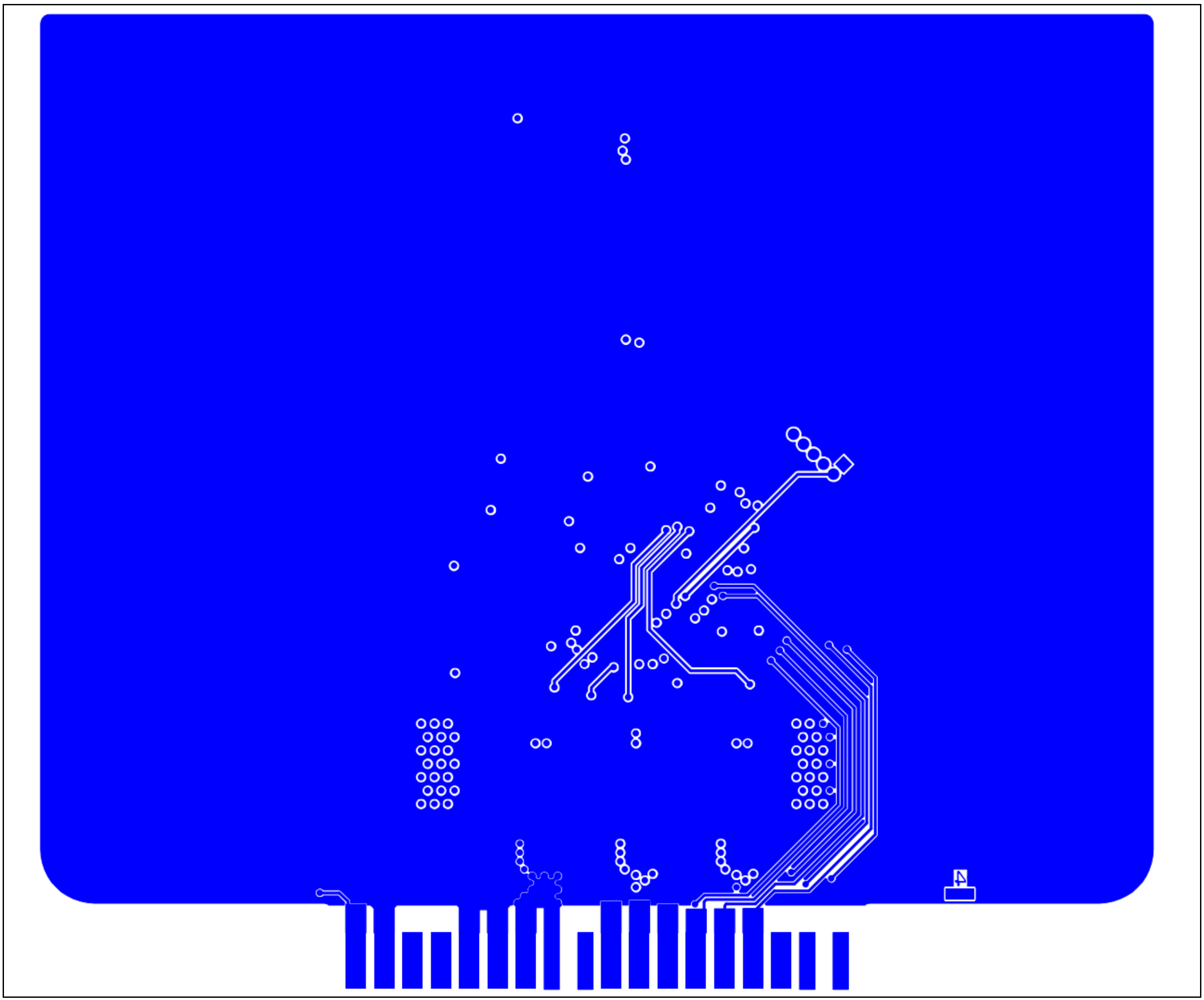


Figure 12 **DEMO_IMR_MTRCTRL_V1 PCB bottom layer**

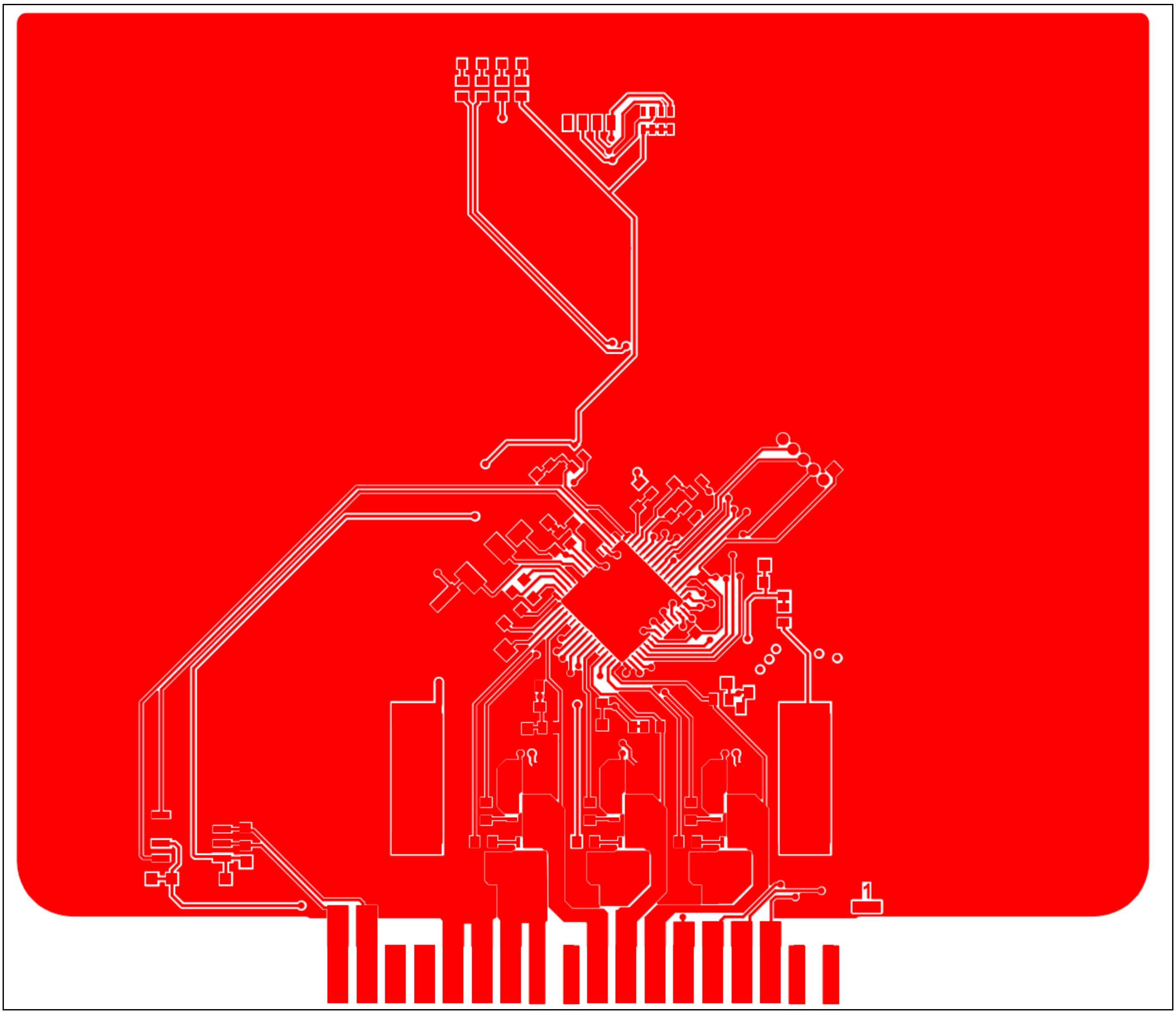


Figure 13 DEMO_IMR_MTRCTRL_V1 PCB top layer

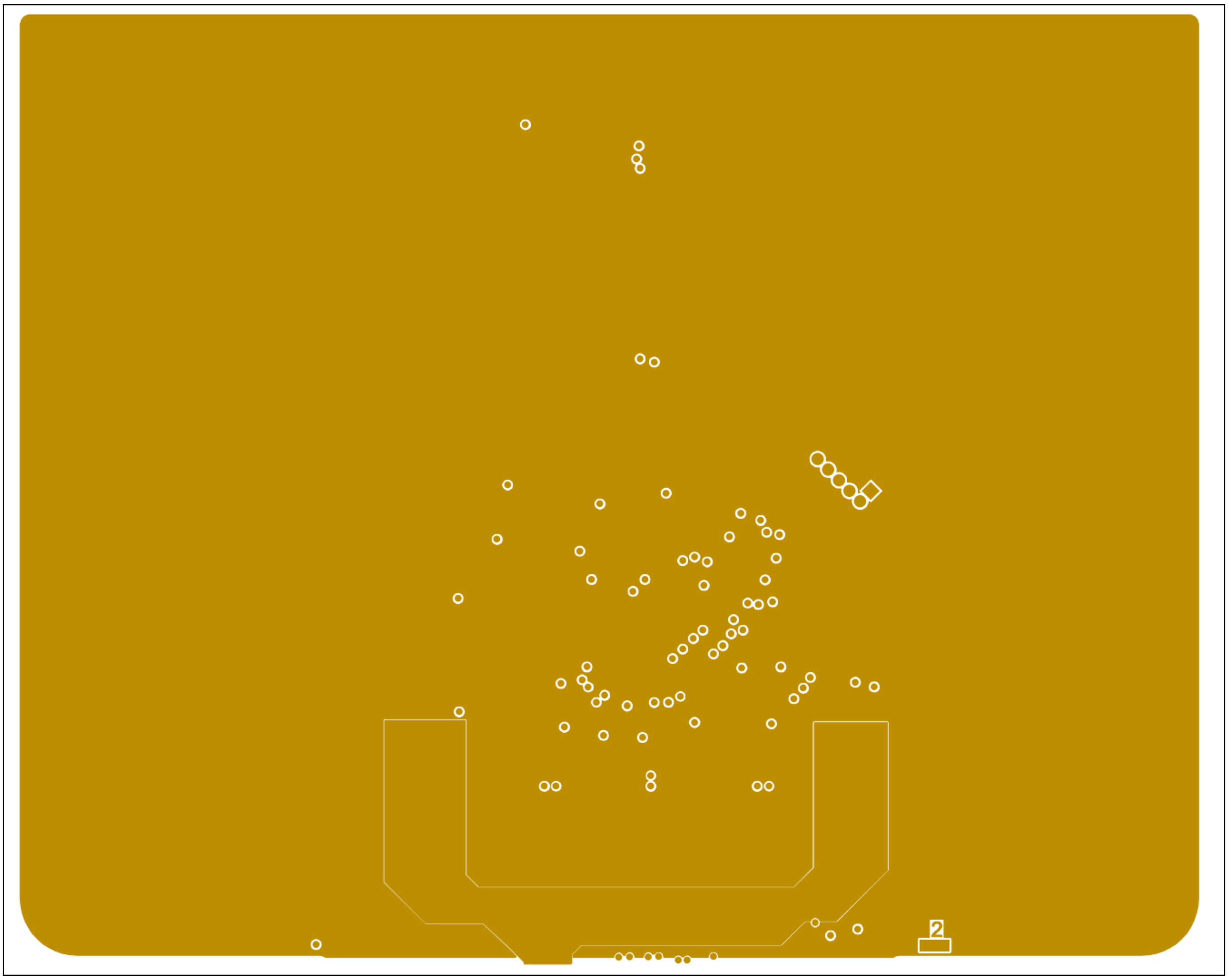


Figure 14 DEMO_IMR_MTRCTRL_V1 PCB mid layer 1

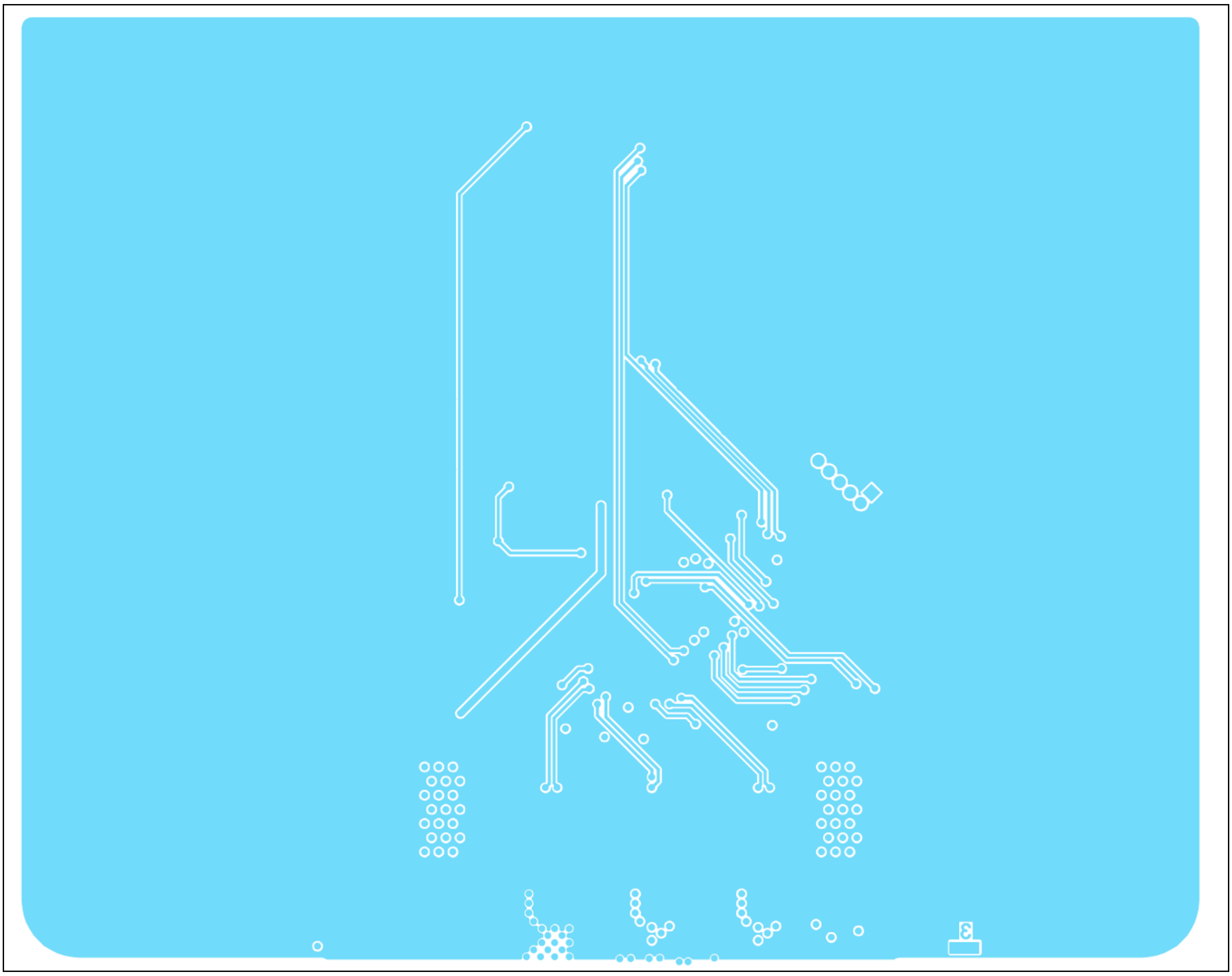


Figure 15 DEMO_IMR_MTRCTRL_V1 PCB mid layer 2

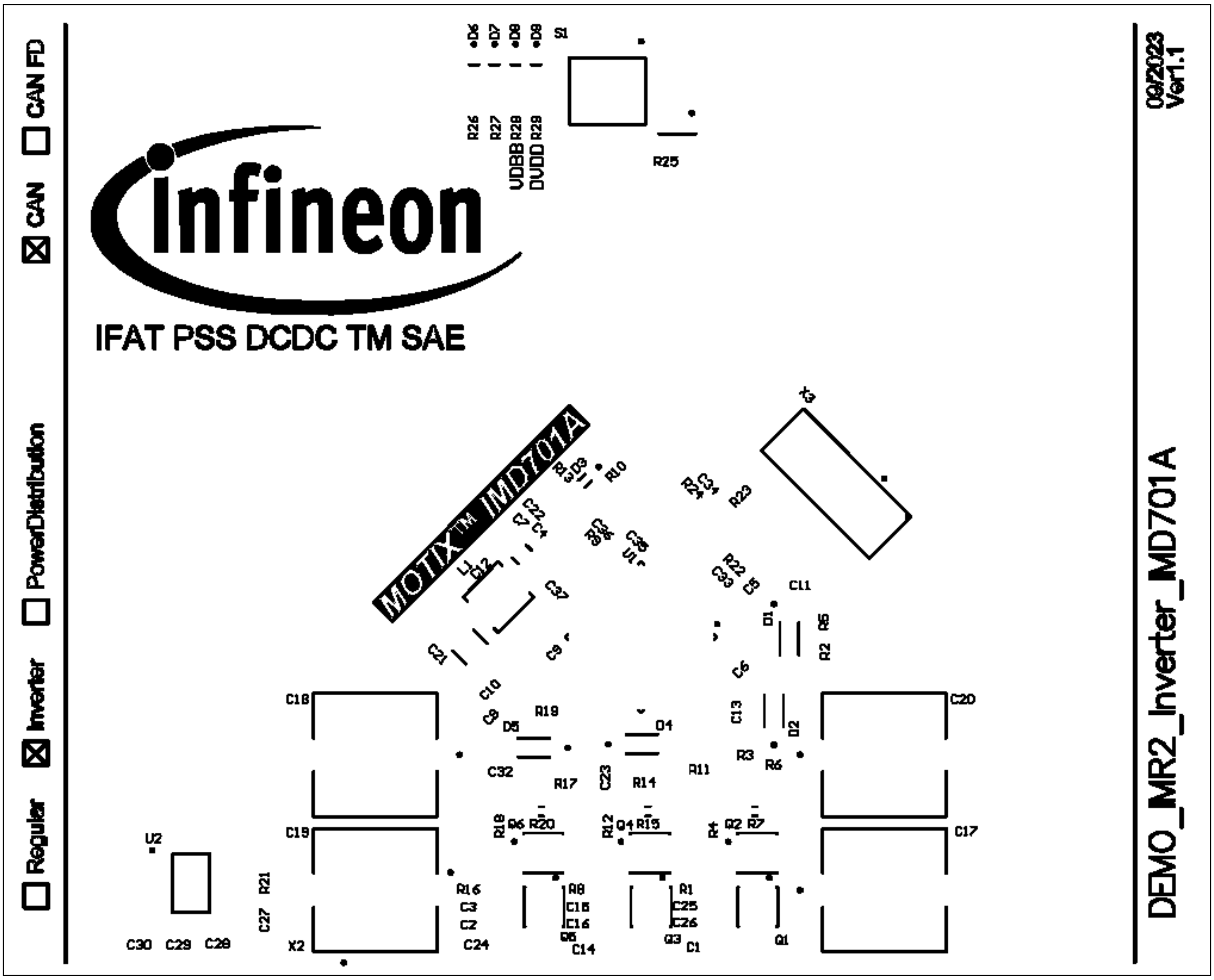


Figure 16 **DEMO_IMR_MTRCTRL_V1 PCB top silkscreen layer**

The PCB layout is optimized to minimize parasitics. This is done by keeping the loops carrying the switching currents as small as possible. The top layer and the first inner layer are used for the 48 V supply to the inverter. The remaining area of the first inner layer is completely occupied by ground potential. The second inner layer and bottom layer are used for signal routing. Special care has been laid on the parallel routing of the current measurement traces which are shielded by the ground layers. The components are all placed on the top layer to allow for easier measurement access due to the evaluation nature of the board.

5 Motor

As the motor control board is used in an AGV/AMR application, the requirements on the drive train are usually focused on a low form-factor with low weight and low rotational speeds. Generally, the low rotational speeds are reached via a high ratio gear box, however, this adds additional weight, size, and cost to the robot. Therefore, for this application, a motor with a high pole pair count is chosen that is usually used in gimbal systems specifically design for cogging free motion at very low speeds. The motor in use is the GL60. It is an out-running motor with 14 pole-pairs and a rated voltage of 24 V. However, the manufacturer advised that an operation at 48 V is possible. The diameter of the motor is 69 mm with a thickness of 22.3 mm. The phase-to-phase inductance is at 2.72 mH and the phase-to-phase resistance at 5.5 Ω.

Figure 17 GL60 motor, an out-running PMSM used for gimbal systems

The motor offers three solder pads for the phase connection. The mounting can be done with screws and cables can be fed through the motor if needed. In this case, the center is occupied due to the usage of a magnet for a rotary sensor.

A cross-section through an exemplary buildup of the motor wheel system is shown in Figure 18. On the motor itself, the mounting bracket is screwed on the stator and the coupling to the wheel is screwed on the rotor. The coupling is then connected with long screws to the mecanum wheel. This wheel type is chosen to ease robot control as instantaneous movement in all directions is needed. Furthermore, a smooth rolling of the wheels is highly beneficial for the use with time of flight (ToF) or radar sensors, therefore, Mecanum wheels are chosen over omni wheels. The complete motor assembly is 100 mm x 100 mm x 125 mm (L × W × H) in size.

Inside the coupling, a magnet is glued that allows the rotary sensor PCB on top to sense the mechanical rotation angle of the motor itself. As the magnetization orientation of the magnet differs depending on the actual rotation of the magnet to the motor when it is glued, it is required to calibrate the motor-offset position in the software. The calibration also needs to be done in case the coupling is demounted from the motor and mounted again.

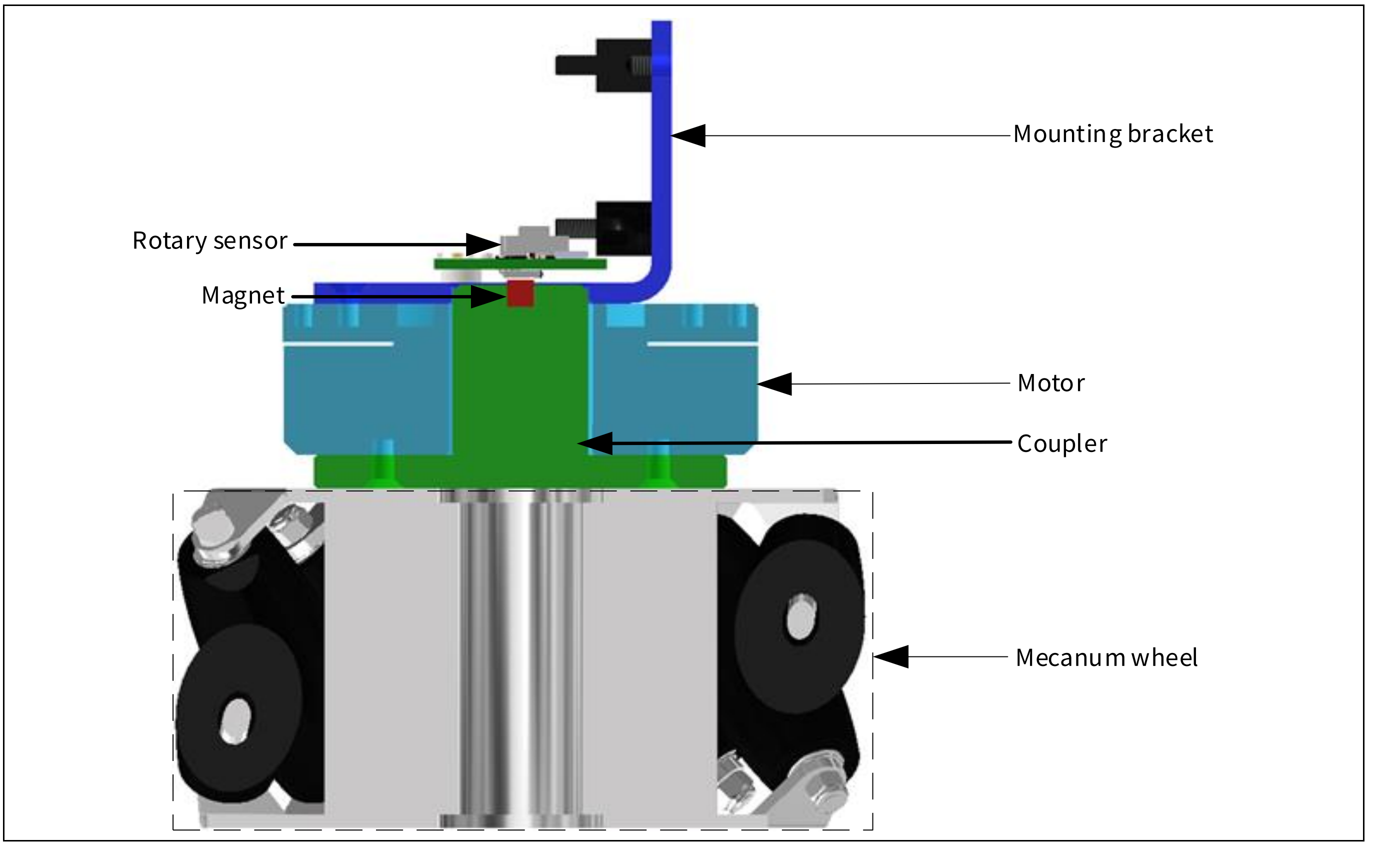


Figure 18 Cross-section through a wheel as used in the IMR demonstrator

6 DEMO_IMR_ANGLE_SENS_V1 board

As the applications require low speeds and torque from standstill, a sensorless approach would lead to poor performance, requiring the rotational angle information of each motor. Therefore, an angle sensor PCB is designed with the TLI5012B E1000.

The TLI5012B E1000 is a 360° angle sensor that detects the orientation of a magnetic field. This is achieved by measuring sine and cosine angle components with monolithic integrated giant magneto resistance (iGMR) elements. These raw signals (sine and cosine) are digitally processed internally to calculate the angle orientation of the magnetic field (magnet). The TLI5012B E1000 is a pre-calibrated sensor. The calibration parameters are stored in laser fuses.

At startup, the values of the fuses are written into flip-flops, where these values can be changed by the application-specific parameters. Further, the precision of the angle measurement over a wide temperature range and a long lifetime are improved with the internal autocalibration algorithm. Data communications are accomplished with a bidirectional synchronous serial communication (SSC) that is SPI-compatible. The sensor configuration is stored in registers, which are accessible by the SSC interface. Additionally, four other interfaces are available with the TLI5012B sensor: pulse-width modulation (PWM) protocol, short-PWM-code (SPC) protocol, Hall switch mode (HSM), and incremental interface (IIF). These interfaces can be used in-parallel with SSC or alone. Pre-configured sensor derivates with different interface settings are available. The one used in IMR is the IIF-type: E1000.

6.1 Incremental interface (IIF)

The incremental interface (IIF) emulates the operation of an optical quadrature encoder with a 50% duty cycle. It transmits a square pulse per angle step, where the width of the steps can be configured from 9-bit (512 steps per full rotation) to 12-bit (4096 steps per full rotation) within the register MOD_4 (IFAB_RES)1). The rotation direction is given either by the phase shift between the two channels IFA and IFB (A/B mode) or by the level of the IFB channel (Step/Direction mode). The incremental interface can be configured for A/B mode or Step/Direction mode in the MOD_1 (IIF_MOD) register. Using the IIF requires an up/down counter on the microcontroller, which counts the pulses, keeps track of the absolute position. The counter can be synchronized periodically by using the SSC interface in parallel. The angle value (AVAL register) read out by the SSC interface can be compared to the stored counter value. In case of a non-synchronization, the microcontroller adds the difference to the actual counter value to synchronize the TLI5012B sensor with the microcontroller.

The TLI5012B-E1000 sensor is preconfigured for IIF and fast angle update period (42.7 µs). It is most suitable for BLDC motor commutation.

- Autocalibration mode 1 enabled
- Prediction enabled
- Hysteresis is set to 0.703
- 12-bit mode, one count per 0.088° angle step
- IIF A/B mode

7 DEMO_IMR_ANGLE_SENS_V1 schematic and PCB

There are two interface options available on the PCB; IIF and SPI, where the latter is not used in this application. When using only IIF, the initial angle detection must be obtained right after startup of the sensor, since the IIF transmits a number of pulses, which correspond to the actual absolute angle value, only right after startup. The index signal that indicates zero crossing is available on the IFC pin.

Therefore, for the microcontroller to get information about the absolute position, the angle sensor must be started only after the position interface (POSIF) of the microcontroller is already active. This is achieved by Q1 that can be enabled by the motor control board and the sensor can be powered up and down at will.

The PCB has two elongated holes which allow for flexible positioning to be used also in other applications. However, the downside of having the elongated mounting holes is the center of the sensor may have little misalignment with center of the magnet. This assembly misalignment can be corrected via the motor control software calibration, which is explained in Section 10.4.3.

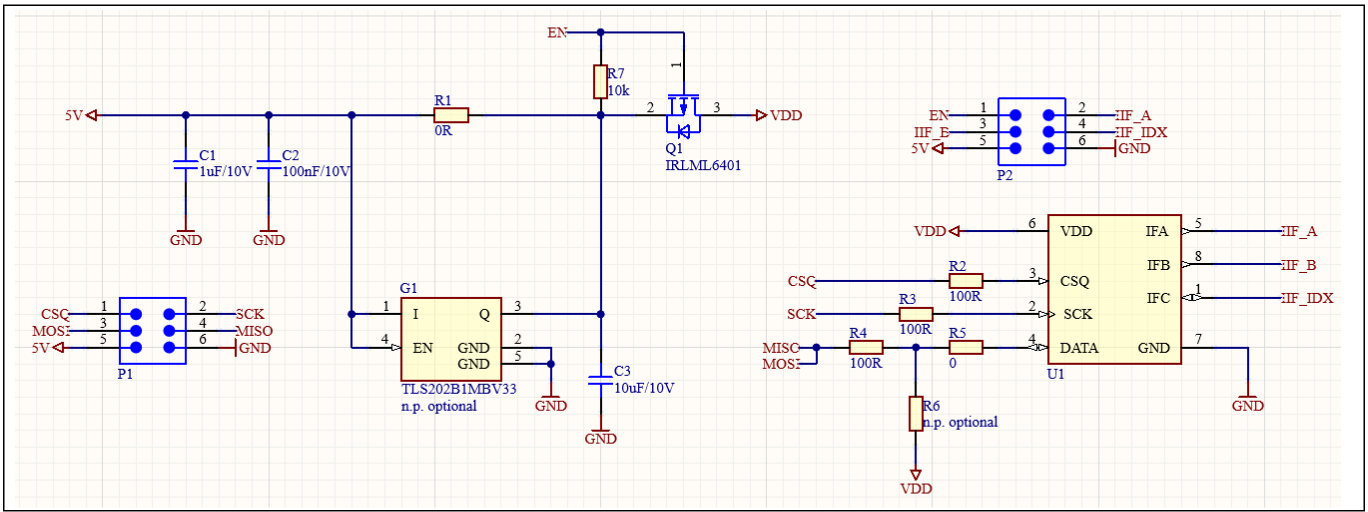


Figure 19 Schematic of the DEMO_IMR_ANGLE_SENSE_V1 board

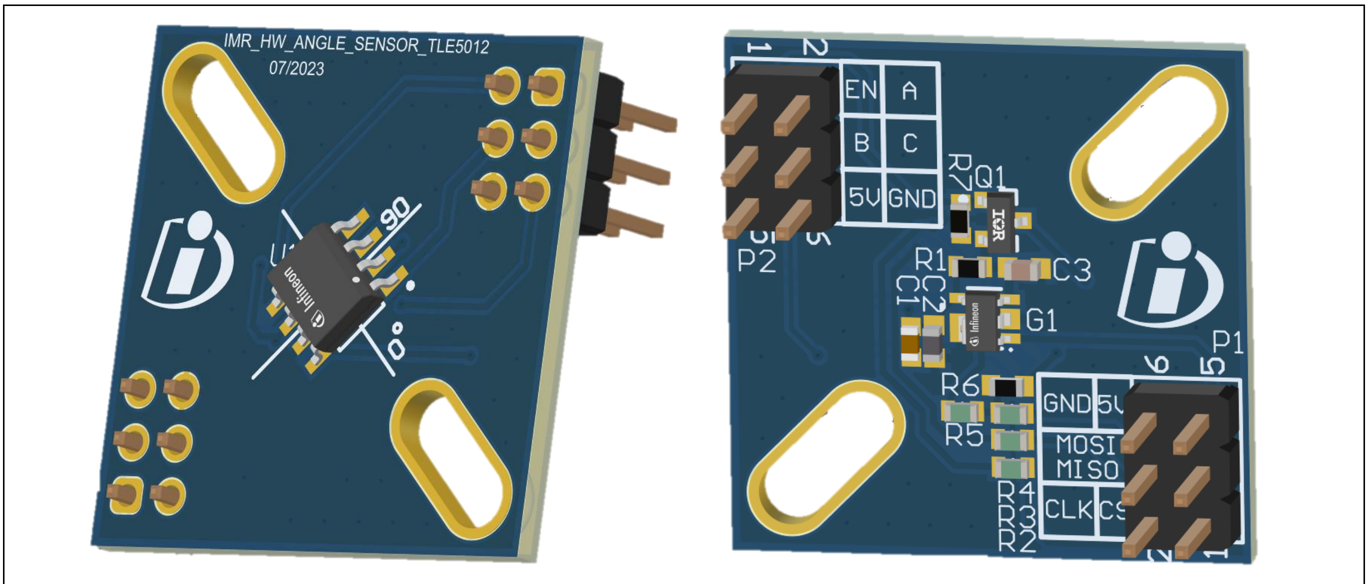


Figure 20 3D view of the DEMO_IMR_ANGLE_SENSE_V1 board

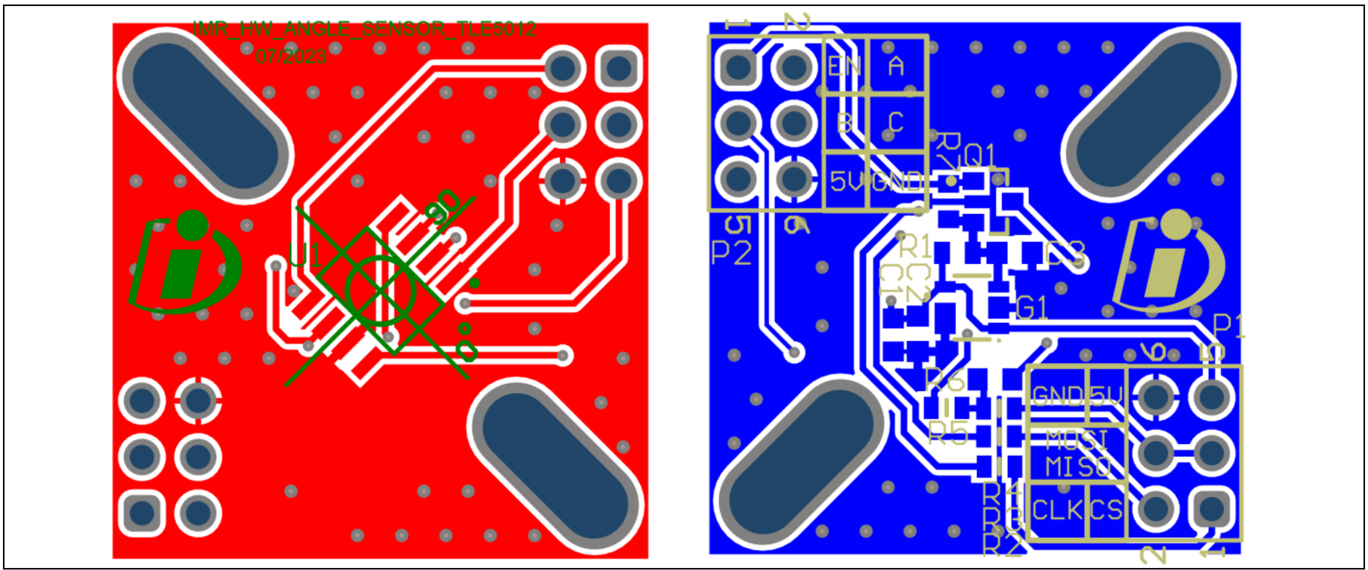


Figure 21 Top – top overlay layer and bottom – bottom overlay layer

8 Bill of materials (BOM)

8.1 DEMO_IMR_MTRCTRL_V1 BOM

Table 3 DEMO_IMR_MTRCTRL_V1 BOM

Reference	Qty	Type	Value/rating	Manufacturer	Part number
C1, C8, C14, C24	4	Capacitor	1 uF, 100 V, 0805, X7S	Murata	GRJ21BC72A105KE11L
C2, C3, C15, C16, C25, C26, C37	7	Capacitor	100 nF, 100 V, 0603, X7R	Würth Elektronik	885012206120
C4, C5, C27, C30, C36	5	Capacitor	1 uF, 25V, 0603, X7R	Würth Elektronik	885012206076
C6	1	Capacitor	10 uF, 16V, 0603, X5R	Murata	GRM188R61C106KAAL
C7	1	Capacitor	4.7 uF, 100 V, 1206, X7S	Murata	GRM31CC72A475KE11L
C9, C12, C22	3	Capacitor	220 nF, 25 V, 0603, X7S	Würth Elektronik	885012206073
C10	1	Capacitor	1 uF, 50 V, 0603, X5R	Murata	GRM188R61H105MAAL
C11, C13, C23, C32	4	Capacitor	2.2 nF, 25 V, 0603, X7R	Murata	GRM188R71E222JA01
C17, C18, C19, C20	4	Capacitor	47 uF, 100 V, Aluminium electrolytic capacitor	Würth Elektronik	865060857005
C21	1	Capacitor	47 uF, 16 V, 1210, X5R	Murata	GRM32ER61C476KE15
C28, C29	2	Capacitor	100 nF, 25 V, 0603, X7R	Würth Elektronik	885012206071
C33, C34, C35	3	Capacitor	330 pF, 50 V, 0603, X7R	Würth Elektronik	885012206080
D1, D2, D4, D5	4	Diode	Zener, 5.1 V, 500 mW, SOD-123	Onsemi	MMSZ4689T1G
D3	1	Diode	Schottky, 45 V, 200 mA, SC79	Infineon	BAS52-02V
D6	1	Diode	LED, Red	Würth Elektronik	150060RS55040
D7, D9	2	Diode	LED, Green	Würth Elektronik	150060VS75000
D8	1	Diode	LED, Blue	Würth Elektronik	150060BS75000
L1	1	Inductor	22 uH, 1 A	Würth Elektronik	74437324220
Q1, Q2, Q3, Q4, Q5, Q6	6	MOSFET	N-channel, 80 V, 90 A, 5.3 mΩ, PG-TSDSON-8 FL package	Infineon	ISZ053N08NM6
R1, R4, R8, R12, R16, R18	6	Resistor	0 R, Jumper, 0603	Yageo	RC0603JR-070RL

Reference	Qty	Type	Value/rating	Manufacturer	Part number
R2	1	Resistor	23.7 kΩ, 0.1 W, 1%, 0603	Vishay	CRCW060323K7FK
R3, R10, R11, R17	4	Resistor	56.2 kΩ, 0.1 W, 1%, 0603	Vishay	CRCW060356K2FK
R5, R6, R13, R14, R19, R22, R23, R24	8	Resistor	2.61 kΩ, 0.1 W, 1%, 0603	Vishay	CRCW06032K61FK
R7, R15, R20	3	Resistor	10 mΩ, 0.33 W, 1%, 0603	Würth Elektronik	580060716002
R9	1	Resistor	5.1 kΩ, 0.1 W, 1%, 0603	Vishay	CRCW06035K10FK
R21	1	Resistor	120 Ω, 0.1 W, 1%, 0603	Yageo	RC0603FR-07120RL
R25	1	Resistor	20 kΩ, Array, 0.063 W, 1%, 1206	Bourns	CAT16A-2002F4LF
R26, R27, R28, R29	4	Resistor	5.11 kΩ, 0.1 W, 1%, 0603	Vishay	CRCW06035K11FK
S1	1	Switch	DIP switch, 25 mA, 24 V	Würth Elektronik	416131160804
U1	1	IC	Gate driver, three phase, motor drive, PGVQFN-64-8 package	Infineon	IMD701A-Q064X128-AA
U2	1	IC	High speed CAN transceiver	Infineon	TLE9351VSJ
X2	1	Header	Card edge header	N/A	N/A
X3	1	Connector	Male box header, 5 x 2, through-hole, 1.27 mm pitch	Würth Elektronik	62701021621

8.2 DEMO_IMR_ANGLE_SENS_V1 BOM

Table 4 DEMO_IMR_ANGLE_SENS_V1 BOM

Reference	Qty	Type	Value/rating	Manufacturer	Part number
C1	1	Capacitor	1 uF, 10 V, 0603, X7S	Würth Elektronik	885012206026
C2	1	Capacitor	100 nF, 10 V, 0603, X7R	Würth Elektronik	885012206020
C3	1	Capacitor	10 uF, 10 V, 0603, X7R	Murata	GRM188R61A106ME69D
G1	1	IC	Voltage regulator 3.3 V	Infineon	TLS202B1MBV33
P1, P2	2	Connector	Header vertical 3 x 2	Würth Elektronik	61300621121
Q1	1	PMOS	-12 V P-channel power MOSFET	Infineon	IRLML6401
R1, R5	2	Resistor	0R, 0603	Bourns	CR0603AJ/-000EAS
R2, R3, R4	3	Resistor	100R, 0603	Yageo	RC0603FR-07100RL
R6	1	Resistor	n.p., optional	–	–
R7	1	Resistor	10k R, 0603	Yageo	RC0603JR-0710KL
U1	1	IC	Magnetic angle sensor	Infineon	TLI5012B E1000

9 Motor control software

9.1 General software concept

The IMR motor control software is available at IMR Software for IMD701A Motor Control.

The software stack for motor control in the IMR is based on a *fixed-point implementation* for XMC1400 microcontroller series. The fixed-point implementation is optimized with hardware-accelerated coordinate rotation digital computer (CORDIC) implementation for motor control applications. Based on this fixed-point data format, the next subsections describe in detail the unit and data format handling.

9.1.1 System-base units

In this chapter, the concept of defining variable system base units to make the algorithms scalable to many applications is explained. First, the Q-format concept is explained, which is used throughout the software to represent a fraction number using integer numbers. Subsequently, the system-base unit concept is described, which explains how configurable scaling factors are defined to achieve consistency for data representation of all physical quantities used in this software. Then the configuration concept is described in detail, from which the minimum and maximum representable values for each type of data are defined. At the end the standard units of measurement used for defining the system base units are listed.

9.1.1.1 Q-format concept

The fractional numbers representing physical quantities with a unit of measurement, as well as fractional numbers representing quantities without unit (e.g. a ratio of two values with the same base unit) can be represented with integers. Because the ARM® Cortex®-M0 Core does not have a hardware floating point unit and for performance reasons (calculation using float data type would be very slow), the fixed-point calculation using integer data types is necessary.

The Q-number format is used for representing rational numbers using integer data types. The Q-number format is a binary fixed point number format where the number of fractional bits and optionally the number of integer bits are specified.

The Q-format abbreviation used in the software is Qx, where x represents number of fractional bits. The integer bits are not mentioned, because they depend on the length of the integer data type. This number has a range between minimum $-(2^{(n-x-1)})$ and maximum $-(2^{(n-x-1)})$ - (2^{-x}) , where n is the number of bits of a signed integer including the bit of sign and 2^{-x} is the resolution.

For example:

- A Q15 number stored on a 16-bit integer has 0 integer bits, 15 fractional bits, and one bit of sign. It can store fractional numbers in the range between -1.0 and 0.999969482421875, with a resolution of 2^{-15}
- A Q15 number stored on a 32-bit integer has 16 integer bits, 15 fractional bits, and one bit of sign. It can store fractional numbers in the range between -65536.0 and 65535.999969482421875, with a resolution of 2^{-15}
- A Q14 number stored on a 16-bit integer has 1 integer bit, 14 fractional bits, and one bit of sign. It can store fractional numbers in the range between -2.0 and 1.99993896484375, with a resolution of 2^{-14}
- A Q31 number stored on a 16-bit integer has 0 integer bits, 15 fractional bits, and one bit of sign. It can store fractional numbers in the range between-0.0000152587890625 and 0.000015258323401212692260742 1875, with a resolution of 2^{-31}

The Q15 format is the most used Q-format for data representation of physical quantities the IMR software. To represent physical quantities having a unit and a specific operating brange using the Q15 number format and to do this with the optimal resolution that can be achieved on the available number of fractional bits, an additional operation is necessary: Q-format data scaling.

9.1.1.2 System-base unit concept

A system-base unit is a measurement unit defined for a physical quantity in a specific system. A system in this context is defined as an electromechanical application containing a motor, an electrical control unit based on the motor control software. When using Q15 format, a system base unit is the maximum representable value of a physical quantity represented by using standard units of measurement in the software.

A system base unit is defined as quantity that includes both the value and the standard unit of measurement. The representation of physical quantities in a system such as a motor controller, consisting of inputs, outputs, and parameters can be 'normalized' by dividing each input and parameter by its corresponding chosen or calculated system base units, thus making all signals, operations inside the system as well as the outputs 'unitless'. If the system base units are chosen/calculated correctly, then it is possible to represent all signals using Q15 number stored on a 16-bit integer, having them all in the range between -1 to 1. This makes it sufficient to use the Q15 format as a unique standard data type. The physical value of a signal is obtained by multiplying the Q15 number with the system base unit assigned to that signal.

When using a Q15 format, the resolution of the system base unit equals 2^{-15} *system_base_unit*. The conversion between physical quantity and a 16-bit integer value representing a Q15 number format that can be seen in the debugger is described as *phys_value* = *16_bit_number* · 2^{-15} · *system_base_unit*. For example, we have a physical value representing a voltage that has the system base unit 24 V, and the decimal integer value 16931 on the variable in the debugger: the physical value is calculated based on the given formula as 12.4 V.

9.1.1.3 Configuration

The configuration of the system base units, which are used by several software components of the underlying motor control sotware are listed in this chapter. The system base values can be found in the `user_defines.h` file.

Table 5 System base values in the motor control software.

Parameter name	Variable name	Default value	Description
Base voltage	`V_BASE`	52.65 [V]	System-base unit for scaling the data signals of voltage type
Base current	`I_BASE`	10 [A]	System-base unit for scaling the data signals of current type
Base speed	`RPM_BASE`	1000 [rpm]	System-base unit for scaling the data signals of speed type
Base impedance	`Z_BASE`	`V_BASE/I_BASE` [ohm]	System-base unit for scaling the data signals of impedance type

Parameter name	Variable name	Default value	Description
Base angle	`THETA_BASE`	3.1415 [rad]	System-base unit for scaling the data signals of angle type
Base for angular velocity	`OMEGA_ME_BASE`	104.7198 [rad/s]	-

10 Motor control solution

10.1 Operation Overview

The software is structured into different routines where each has a specific functionality. First of all, there is an initialization routine which initialized hardware-based components, such as timer interfaces, ADC as well as the POSIF interface for position sensing. Furthermore, the initialization also initializes the control framework, i.e. PI-controller parameters, etc. After this routine is finalized, the motor control is executed as an interrupt-based scheme with empty main() function. There is one main interrupt with two sub-routines defined.

The main interrupt routine of the motor control is divided into a fast loop and a slow loop. The fast loop is synchronized with the PWM generation and operates at the switching frequency. Its main operations are angle retrievement, phase current retrievement, and Clarke/Park transformation, as well as executing the current controllers and updating the SV-PWM compare values.

The slow loop runs at factor 10 slower and computes the averaged speed as well as executes the speed controller. Once the speed controller is done, the current reference for the faster current controller is also updated.

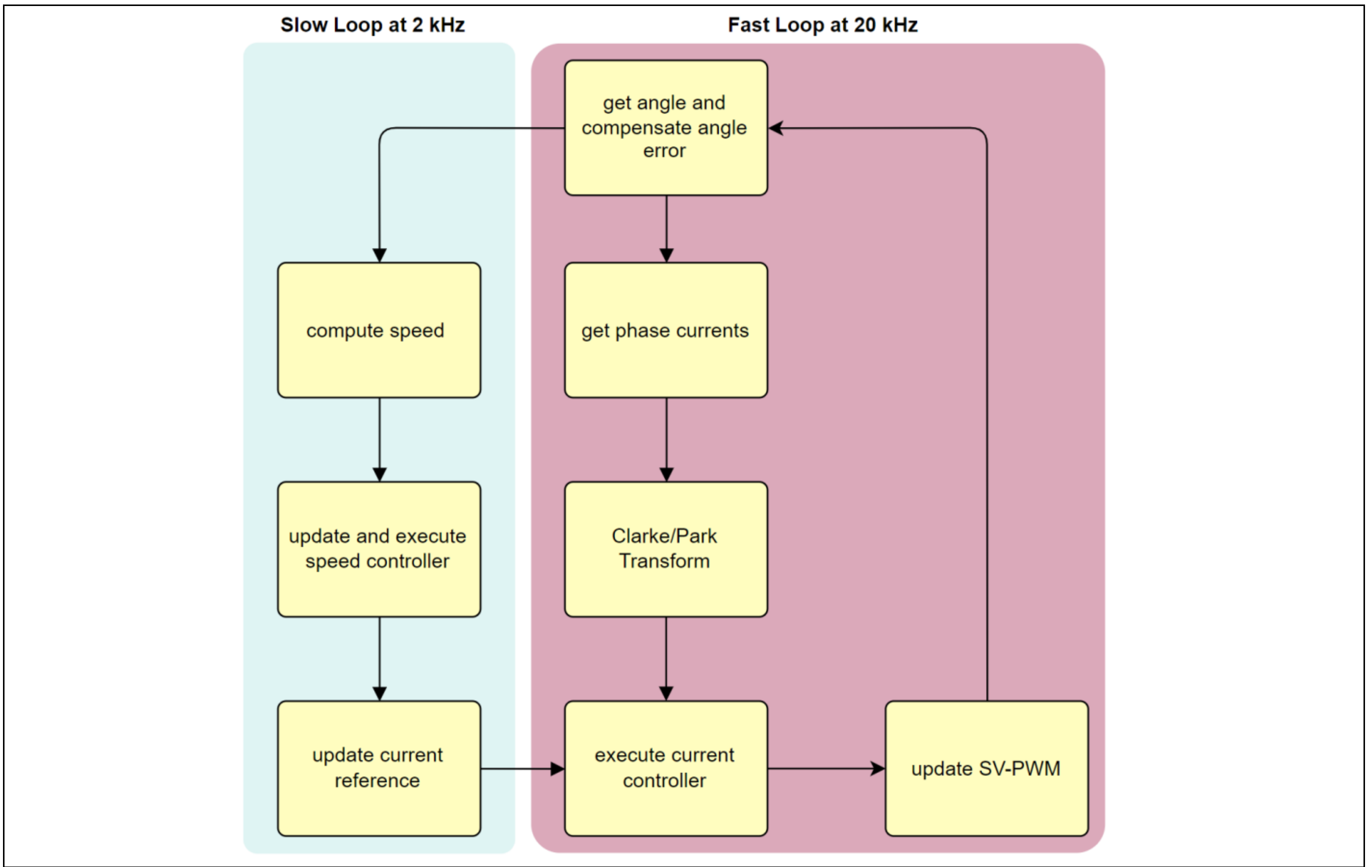


Figure 22 Overview of the motor control solution

10.2 Modulation strategy

The modulation strategy used in the software is the so-called space-vector modulation (SVM) technique. In the SVM method, the voltage reference is created using the two adjacent voltage vectors. To obtain the voltage vectors, Consider the simple representation of the B6 inverter bridge, shown in Figure 23.

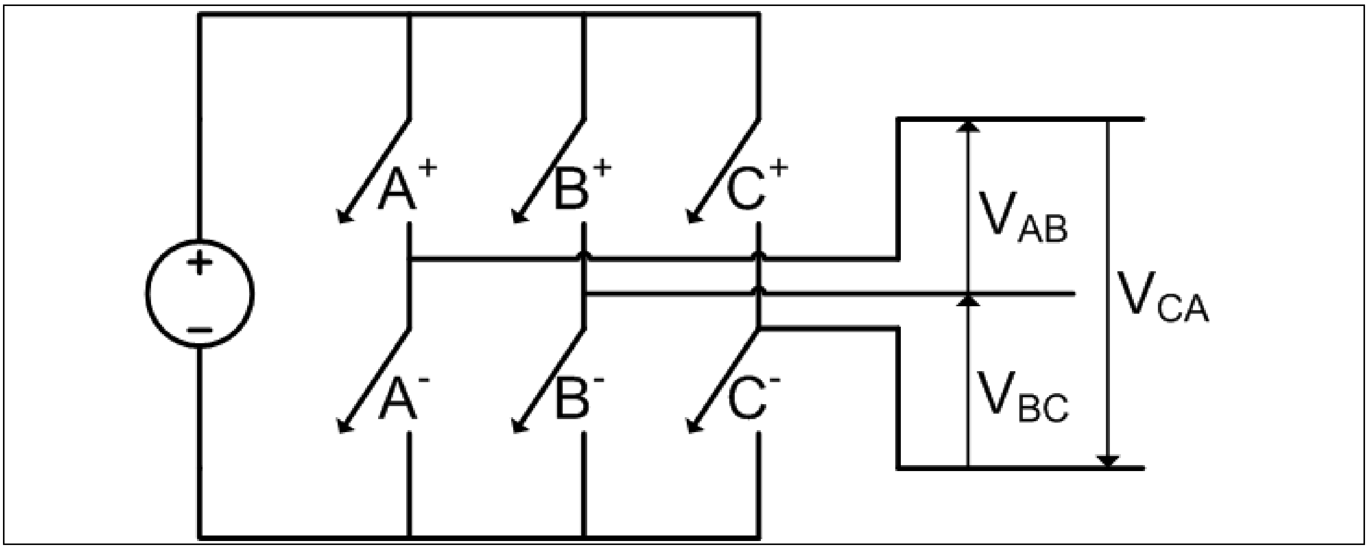


Figure 23 Simplified representation of B6 bridge

Since the two switches per leg must not be active at the same time, one can represent the phase leg as a binary number, which maps the switching states into a truth table. Considering all six switches one can create the following truth table Table 6.

Table 6 Truth table of SVM technique in B6 inverter bridge configuration

Vector	A^+	B^+	C^+	A^-	B^-	C^-	V_{AB}	V_{BC}	V_{CA}	
V_0 = {000}	OFF	OFF	OFF	ON	ON	ON	0	0	0	zero vector
V_1 = {100}	ON	OFF	OFF	OFF	ON	ON	$+V_{dc}$	0	$-V_{dc}$	active vector
V_2 = {110}	ON	ON	OFF	OFF	OFF	ON	0	$+V_{dc}$	$-V_{dc}$	active vector
V_3 = {010}	OFF	ON	OFF	ON	OFF	ON	$-V_{dc}$	$+V_{dc}$	0	active vector
V_4 = {011}	OFF	ON	ON	ON	OFF	OFF	$-V_{dc}$	0	$+V_{dc}$	active vector
V_5 = {001}	OFF	OFF	ON	ON	ON	OFF	0	$-V_{dc}$	$+V_{dc}$	active vector
V_6 = {101}	ON	OFF	ON	OFF	ON	OFF	$+V_{dc}$	$-V_{dc}$	0	active vector
V_7 = {111}	ON	ON	ON	OFF	OFF	OFF	0	0	0	zero vector

Making use of these states allows to define so-called sectors where a so-called reference vector is rotating based on the motor's position/speed and the computed magnitude.

Before diving into the exact computation, let us define the most important parameters for SVM. The switching period time is defined as Ts, the time in which voltage vector x is active is defined as Tx (see Table 6), the times where none of the voltage vectors are active (i.e. the time related to zero vectors V_0 and V_7) is T_0 and T_7, respectively. Consider the reference vector in the first sector, as shown in Figure 24. In this sector, two main voltage vectors are used, namely V_1(100) and V_2(110). To generate the reference voltage, V_1 and V_2 for a time T_1 and T_2 are applied respectively within a switching period T_s. For the rest of the period T_s, the zero vectors V_0(100) and V_7(111) are applied for a time T_0 and T_7, which act as scaling of the vector magnitude as expressed by V_{ref} in Equation 1. Therefore, Equation 1 obtains:

Equation 1

$$V_{ref} = \frac{T_0V_0 + T_1V_1 + T_2V_2 + T_7V_7}{T_s}$$

Figure 25 shows the two different switching schemes for each leg of the inverter based on the timings acquired for each voltage vector. There are a few different switching schemes that can be used to create the vectors determined by the SVM method. However, 7-segment and 5-segment switching methods as shown in the Figure 25 are the most popular (T0 = T7). The software in the IMR is using the 7-segment modulation technqiue.

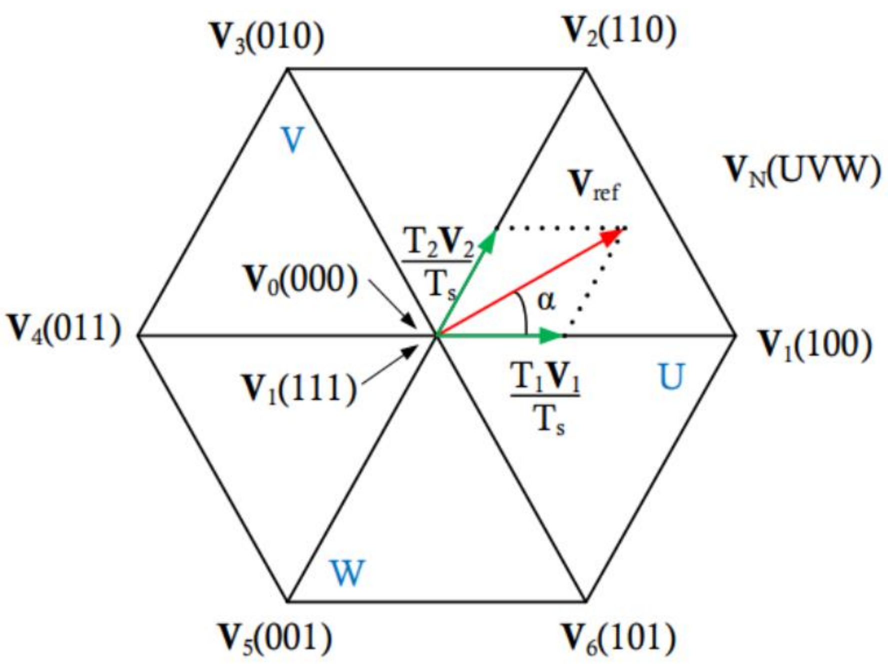


Figure 24 **Overview of the space-vector modulation technique.**

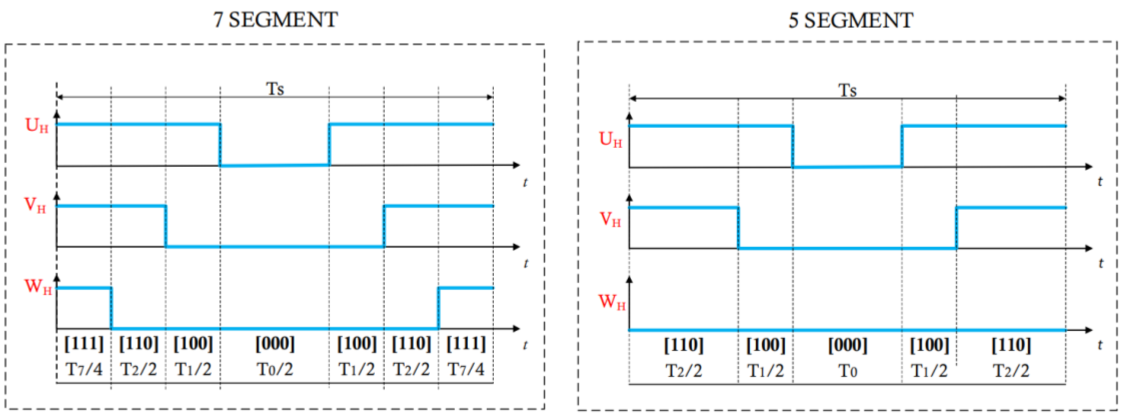


Figure 25 **7-segment and 5-segment modulation pattern**

The difference between the two switching methods is that in 7-segment, both V0 and V7 times are used to create zero voltage while in 5-segment only V0 is used to create zero voltage. Note that the 5-segment switching method creates less switching compared to the 7-segment that reduces the switching losses. However, the current ripple in this switching method is higher compared to 7-segment.

10.3 Current sensing

Current sensing is enabled using low-side shunt resistors (also known as three-shunt current sensing), R7, R15, and R20 as shown in Figure 28. In three-shunt current sensing, the ADC conversion trigger is set at half of the PWM cycle where all the low-side switches are on, and the high-side switch is off.

10.3.1 Current transformations

One important sub-block of modern motor control algorithms is the current transformation. This sub-block is needed to transform the three-phase current information into a reference frame that represents the currents in a simple 2D vector-space, aligned with the rotor angle to apply similar control strategies as known from DC machines. Therefore, two famous mathematical operations, the so-called Clarke and Park transforms are introduced in this subsection. s

The αβ- or Clarke-transformation describes a transformation of the three-phase system to a two-phase system with a rectangular coordinate system that is fixed to the stator and, hence, a stationary coordinate system. Its abscissa is the real alpha (α) axis and is aligned with the first inductance of the machine (U in UVW-described machine phases). The ordinate is the imaginary beta (β) axis. Hence, the αβ-coordinate system is a Cartesian coordinate system for complex numbers with both, α and β part being real numbers, see Figure 26.

The dq- or Park-transformation is similar to and extends the αβ-transformation as it transforms the three-phase system to a two-phase coordinate system, too. The main difference is the coordinate system as it is not stationary and fixed to the first inductance but rotates with electrical frequency of the rotor. This results in a transformation of the rotating field in two constant values, the d (direct) and q (quadrature) part of the complex number $\underline{x} = x_d + j \cdot x_q$. Due to this transformation, control is only applied to the d (direct) and q (quadrature) part of the current, hence, two scalar values that compute a current vector. This simplifies the overall control strategy and allows simple implementation using PI controllers.

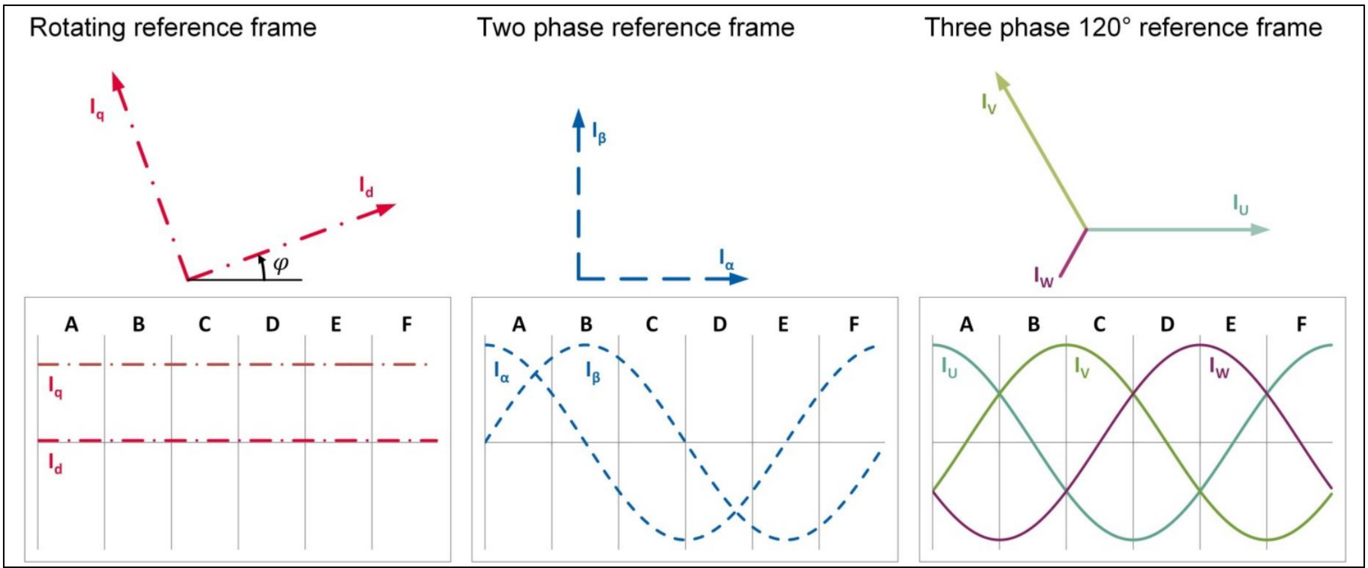


Figure 26 Clarke and Park transformations

10.4 Speed and position sensing

10.4.1 Incremental interface (IIF)

The incremental interface (IIF) emulates the operation of an optical quadrature encoder with a 50% duty cycle. It transmits a square pulse per angle step, where the width of the steps can be configured from 9-bit (512 steps per full rotation) to 12-bit (4096 steps per full rotation) within the register MOD_4 (IFAB_RES)1). The rotation direction is given either by the phase shift between the two channels IFA and IFB (A/B mode) or by the level of the IFB channel (Step/Direction mode).

The incremental interface can be configured for A/B mode or Step/Direction mode in the MOD_1 (IIF_MOD) register. Using the IIF requires an up/down counter on the microcontroller, which counts the pulses, keeps track of the absolute position. The counter can be synchronized periodically by using the synchronous serial controller (SSC) interface in parallel. The angle value (AVAL register) read-out by the SSC interface can be compared to the stored counter value. In case of a non-synchronization, the microcontroller adds the difference to the actual counter value to synchronize the TLI5012B sensor with the microcontroller.

The TLI5012B-E1000 sensor is preconfigured for IIF and fast angle update period (42.7 µs). It is most suitable for BLDC motor commutation.

- Autocalibration mode 1 enabled
- Prediction enabled
- Hysteresis is set to 0.703
- 12-bit mode, one count per 0.088° angle step
- IIF A/B mode

A feature of the TLI5012B is the possibility to sense absolute position. Therefore, the initial position of the sensor must be extracted during the initialization. This can be done either by using the SPI interface or by enabling the sensor. Enabling the sensor is triggered throughout the microcontroller by pulling the EN pin of the sensor to LOW. After boot-up, the sensor is pulsing the initial position through the IIF interface, and the number of pulses is used to compute the initial position.

10.4.2 POSIF interface

A POSIF is a universal position interface unit. When used in conjunction with the CAPCOM units, CCU4, and CCU8, POSIF offers powerful solutions for motion control systems that use various position sensors or rotary encoders in the feedback loop. This enables the building of both simple and complex control feedback loops for industrial and automotive motor applications, targeting high-performance motion, and position monitoring.

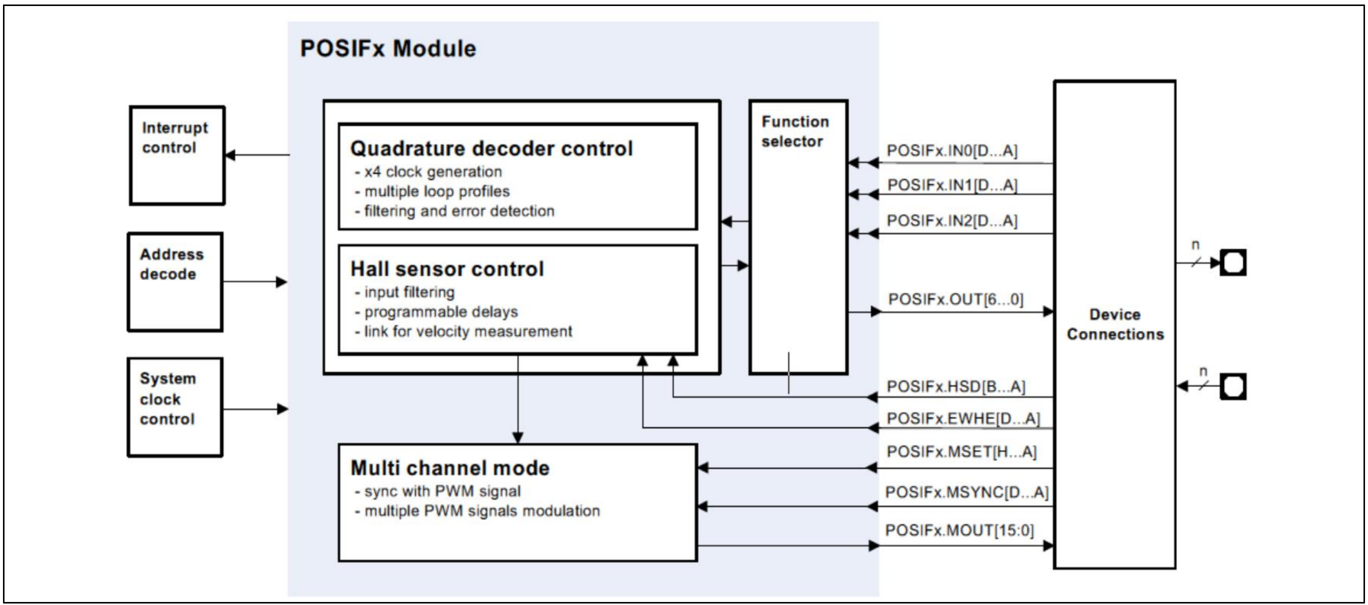


Figure 27 General overview of the POSIF interface

10.4.2.1 Quadrature decoder mode

The quadrature decoder unit is used for position control linked with a rotary incremental encoder. It has interfaces for position measurement, motor revolution, and velocity measurement. It provides an easy plug-in for rotary encoders:

- With or without index/top marker signal
- Gear-slip or shaft winding compensation
- Separate outputs for position, velocity and revolution control, matching different system requirements
- Extended profile for position tracking, with revolution measurement, and multiple position triggers for each revolution
- Support for high dynamic speed changes due to tick-to-tick and tick-to-sync capturing method

Inside the quadrature decoder mode, there are two different sub-sets available:

- Standard quadrature mode. – The standard mode is used when the external rotary encoder provides two-phase signals and an index/marker signal that is generated once per shaft revolution
- Direction count mode. – The direction count mode is used when the external encoder only provides a clock and a direction signal

Each quadrature clock cycle is a 4-state, 4 clocks, and pulse train that is generated by two phase-shifted signals from a sensor-pair, (A) and (B). Since the state order determines which sensor is in the sense position of the leading phase, each state transition is unique for the motion direction; clockwise or counterclockwise.

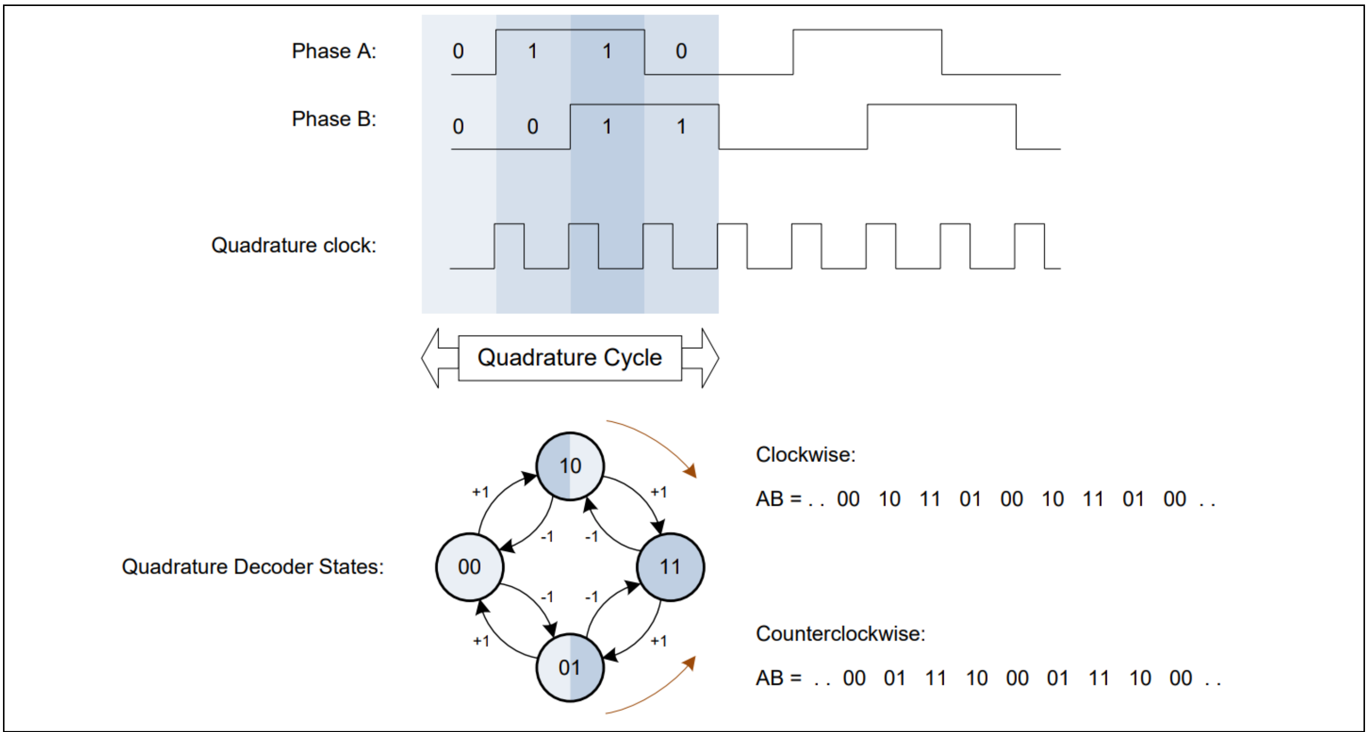


Figure 28 Quadrature encoder signals and states.

Position encoders that work in standard quadrature mode provide two phase signals (phase A and phase B), and encoders with or without an index/top marker signal.

- The inputs should be mapped to:
  - POSIFx.IN0[D...A]
  - POSIFx.IN1[D...A]
  - POSIFx.IN2[D...A]
- The output pins are POSIFx.OUT0 to POSIFx.OUT5 for quadrature clock, direction, period clock, index occurrence, index (revolution), and synchronous start.
  - The quadrature clock gives position up/down count information in conjunction with the direction signal
  - The period clock contains sequences of (AB) or (BA) pulse-pairs for velocity detection
  - The index occurrence pin asserts Z-mark/index (on QDC.ICM conditions), whilst the Index revolution asserts only once per revolution
  - Synchronous start output, POSIFx.OUT5, is linked with the module run bit can be used together with the CAPCOM units for a complete synchronous start, if those CAPCOM units have been preset for external events control (external start and extended start for start for example, or flush/start)

10.4.3 Encoder autocalibration

An important aspect of robust and high-performance motor control lies in the quality of the sensor feedback for position and speed measurement. The implemented software solution relies on encoder-based feedback and therefore, effects such as offsets as well as non-linearities must be avoided to ensure high-performance control. The sensor readings are very sensitive to any misalignment between the sensor IC and the permanent magnet mounted on the rotating shaft. Since mechanical assembly always comes with tolerances, an autocalibration feature is implemented to ensure the best performance during operation as well as robustness against mechanical misalignment, electrical offsets, etc. The auto-calibration feature can be triggered externall via CAN or directly in the software by setting `encoder_calib_flag = 1`.

The next sections describe the autocalibration procedure in detail. A flowchart representation is shown in Figure 29. For a well-aligned mechanical system, the autocalibration is not necessary and the autocalibration code can be left commented.

10.4.3.1 Pre-positioning

Pre-positioning is used to lock the rotor and perform d-axis alignment. The pre-position serves as the zero position for both electrical as well as mechanical angle. To achieve pre-positioning, the motor is modulated with zero angle and approx. 10% nominal current. Consequently, the rotor will align with the nearest magnetic north pole to align with the stator magnetic field. The encoder readings at locked position are considered as DC offsets and are stored for further calibration. The pre-position is a reference position for both mechanical as well as electrical angle.

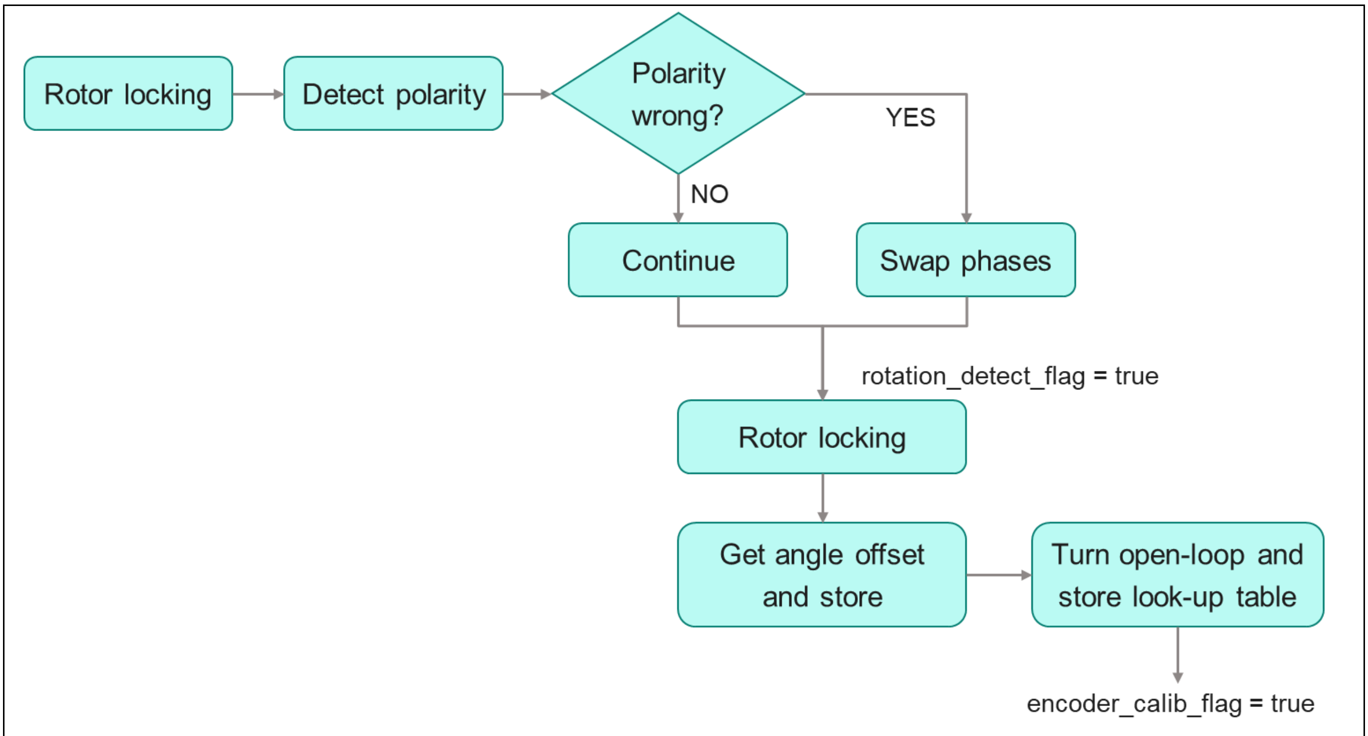


Figure 29 Flowchart of the encoder autocalibration routine.

10.4.3.2 Polarity detection

The first step of the autocalibration is polarity detection, i.e. making sure that the sign of both the electrical as well as the sensed mechanical angle match. Therefore, the motor is pre-positioned and is rotated in open-loop stepwise in positive electrical direction. The mechanical sensor readings are recorded and the difference between successive steps is computed. If the difference is positive, the mechanical angle is increasing and therefore matches the sign of the electrical angle. However, if the difference is negative, the mechanical angle is decreasing. To match the sign, phase swapping is necessary. This is achieved by swapping the respective CRS registers of two phases of the CCU8 timer.

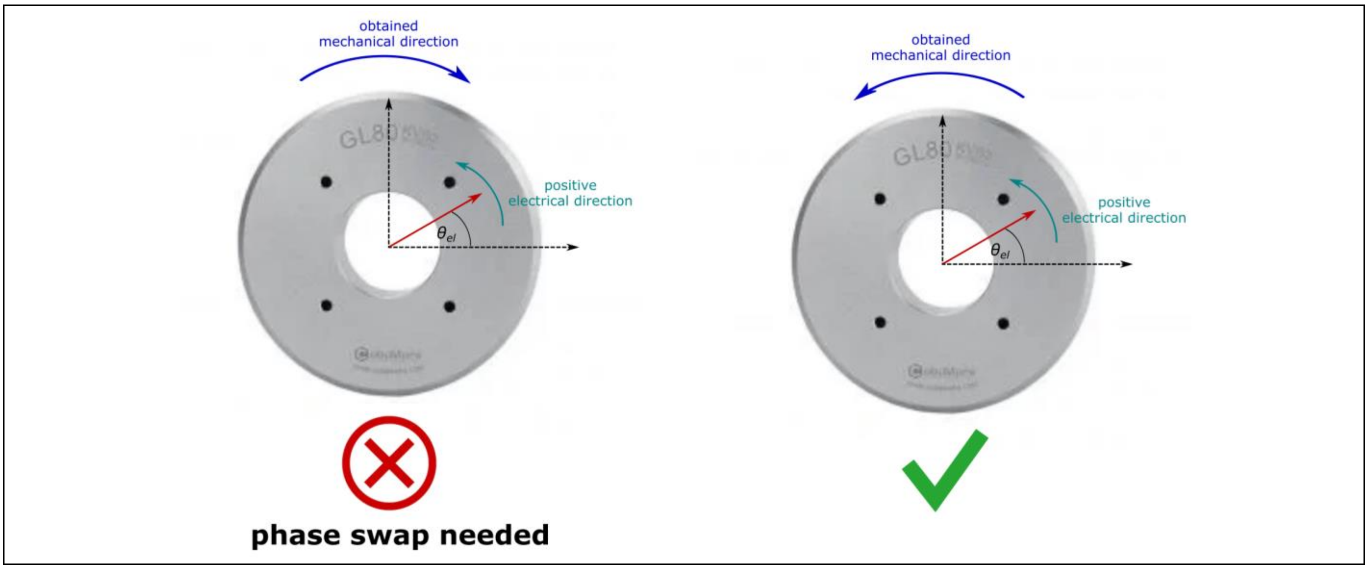


Figure 30 Illustration of the phase swapping.

10.4.3.3 Misalignment calibration

To further increase the robustness of the control, encoder eccentricity calibration is performed. Eccentricity occurs due to misalignment of the sensor position with respect to the magnet mounted on the shaft. The effect is obtained as a nonlinear angle response over a full turn. Since the motor is a high pole-pair type, eccentricity becomes even more severe as small mechanical mismatch may lead to large electrical errors, which drastically reduces control performance. Figure 31 illustrates sensor mismatch and the effect of eccentricity on the angle signal. Clearly, nonlinear behavior is obtained that gives wrong readings and decreases performance. To further highlight the impact, consider that the angle error due to eccentricity is 5° mechanical. Given a motor with 14 pole-pairs, this would result in an electrical angle error of 21.42° which is clearly not feasible.

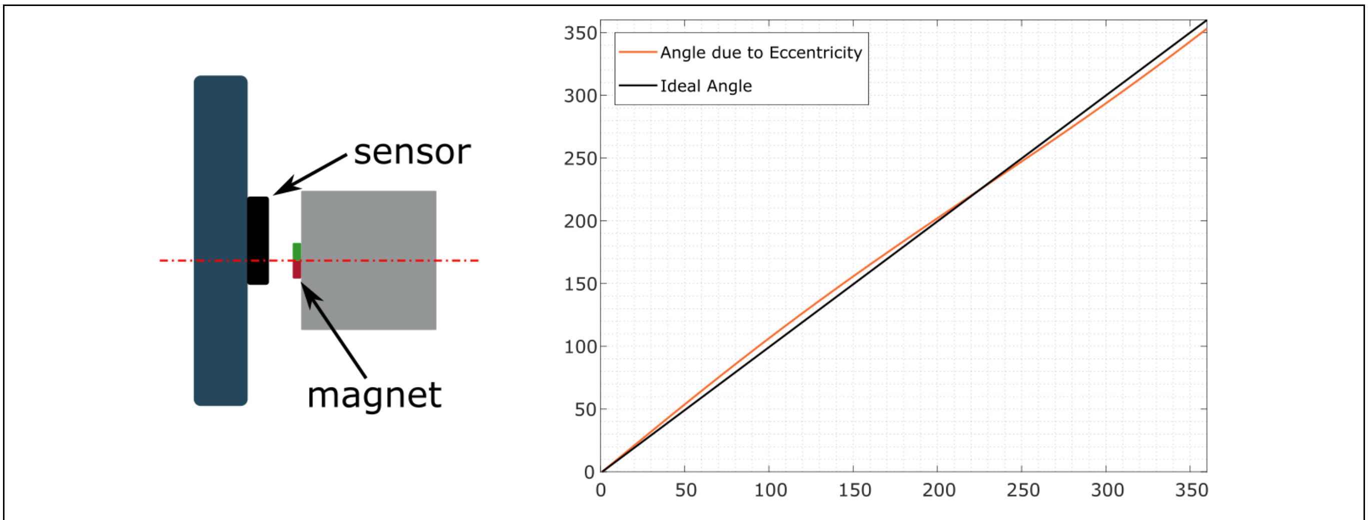


Figure 31 Illustration of eccentricity.

The calibration routine for eccentricity is based on a look-up table approach. After pre-positioning, the rotor is turned one full rotation in open-loop control. The sensor readings during this rotation are stored in a look-up table on the MCUs flash memory. Afterards, an ideal angle look-up table is computed based on a linear curve ranging from 0 to 360°, which represents the angle under ideal conditions. When a new angle reading is obtained during control, the angle error between the look-up table values from the calibration and the ideal look-up table is computed. The obtained error is subtracted from the sensor reading and hence, the reading is calibrated towards the ideal angle. Given this calibration method, the control performance is getting independent from sensor mounting position. This method does not use any averaging, since sensor noise was not obtained as a critical issue. However, to further improve the performance, also averaging can be supplemented.

10.5 Control strategy

The control strategy of the software solution is based on classic field-oriented-control (FOC) using a cascaded PI structure for speed and current control including important mathematical operations as well as sensor feedback for speed and position acquisiton of the rotor. The speed controller output computes the reference current for the q-axis current whereas the d-axis is controlled towards zero current (i.e. no flux-weakening). Current measurements are acquired from low-side shunt resistors, which are amplified and converted to digital representations through analog-to-digital conversion. A general overview block diagram is shown below in Figure 32. Light green blocks are parts implemented in software whereas blue blocks represent hardware parts.

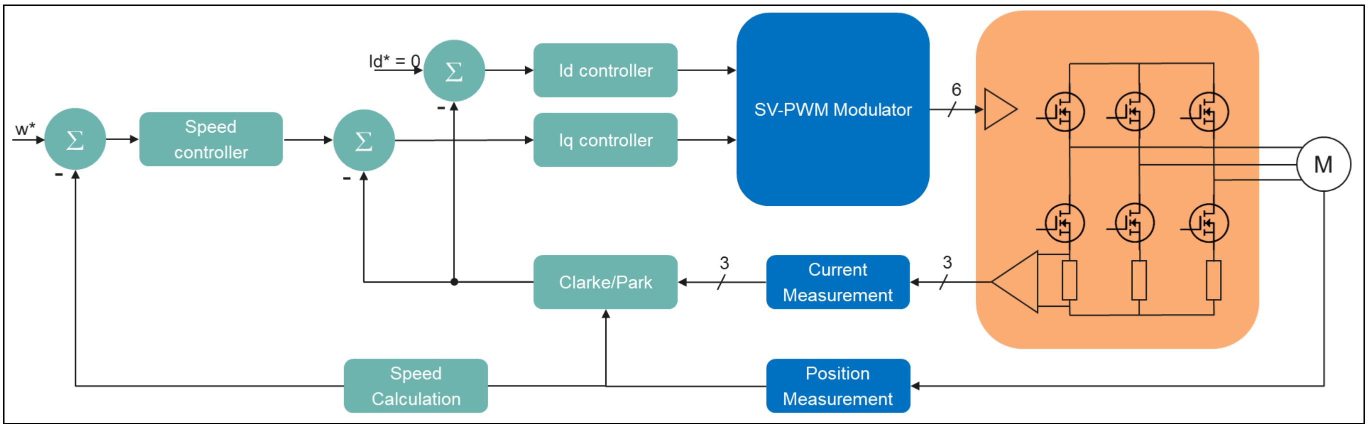


Figure 32 Standard FOC topology used in the IMR motor control

10.5.1 PI current controller

Current control is enabled by using conventional Proportional – Integral (PI) controllers to control the d-axis and q-axis currents. The software is using PI controllers in standard parallel path topology with anti-windup and output clamping. The update frequency of the PI current controller is set to the interrupt rate of the CCU8 timer, synchronized with the PWM generation and the bandwidth is set to 250 Hz. Since flux-weaking operation is not needed for the IMR, the d-axis controller is set to standard Id = 0 control.

The PI controllers for current control can be tuned based on the parameters of the motor. Considering the motor as a linear system using RL equivalent circuit, the transfer function is given by

$$G(s) = \frac{I(s)}{U(s)} = \frac{1}{R(1 + s\,T_{Mot})(1 + sT_{Dt})}$$

Equation 2

where,

- U(s) is the input voltage, and I(s) is the output current
- T_{Mot} is the time constant of the motor
- T_{Dt} is the delay and dead-time due to discrete controller and inverter

Based on this representation of the system, the controller parameters can be calculated as follows:

- $K_p = \frac{L}{T_c}$
- $K_i = \frac{R\,T_c}{T_s}$

Where Ts is the sampling time and Tc is the inverse of the bandwidth. To tune the 'aggressiveness'of the controller, adjustment factors in the range from 0 to 1 are implemented for fine-tuning of the motor.

- $K_p = k_{wp}\frac{L}{T_c}$
- $K_i = k_{wi}\frac{R\,T_c}{T_s}$

10.5.2 PI speed controller

Speed control is enabled by using a conventional PI controller in standard parallel path topology with anti-windup and output clamping. The update rate of the PI speed controller is set to 2 kHz, and the bandwidth is set to 25 Hz, i.e. factor 10 smaller as the current controller that is usually used in motor drive applications. The output of the speed controller directly gives the reference q-current for the underlying current controller. Note that from experimental investigations, the gain of the I-part of the controller was set to zero and only the P-part reacts to speed changes. If some more aggressive control is necessary, the I-part can be included, however this should be treated with caution as the mobile platform reacts very sensitive to overshoot and might fall into an oscillation state. Therefore, it is recommended to neglect the I-part for speed control and rely on P-control with the effect of having a remaining control error. However, the residual speed error is minor and does not affect overall system performance dramatically.

10.5.3 Important parameters

Most important parameters and variables regarding the control strategy are listed and explained below. All variables can be found in `PI.h` header file. For different setups, the parameters must be changed according to the user's needs.

10.5.3.1 Speed control

- `#define Fs_SPD_CTRL`

  – Defines the speed controller update rate in Hz
  Standard value is set to 2000 Hz

- `#define Fc_SPD_CTRL`

  – Defines the bandwidth of the speed controller in Hz
  Standard value is set to 25 Hz

- `#define RPM_MAX`

  – Defines the max. allowed speed of the motor in RPM
  Standard value is set to 300 RPM

- `#define OUT_SPD_CTRL_MAX`

– Defines the max. output value of the speed controller in A
Standard value is set to 6 A.

- `#define OUT_SPD_CTRL_MIN`

– Defines the max. output value of the speed controller in A
Standard value is set to -6 A

- `#define Kp_SPD_CTRL`

– Defines the Kp gain value of the speed controller in A/rad/s.
Standard value is set to 0.1 A/rad/s

- `#define Ki_SPD_CTRL`

– Defines the Ki gain value of the speed controller in A/rad/s
Standard value is set to 1.0*Ts_SPD_CTRL A/rad/s

10.5.3.2 Current control

- `#define Fc_CRNT_CTRL`
  - Defines the current controller bandwidth in Hz
    Standard value is set to 250 Hz

- `#define Tc_CRNT_CTRL`
  - Defines the update cycle time of the current controller in seconds
    Standard value is set to 1/20000 seconds

- `#define SVM_VREF_MAX`
  - Defines the max. SVM value per unit
    Standard value is set to 1

- `#define OUT_CRNT_CTRL_MAX`
  - Defines the max. output value of the current controller in pu
    Standard value is set to 1

- `#define OUT_CRNT_CTRL_MIN`
  - Defines the max. output value of the current controller in pu
    Standard value is set to 1

- `#define KpD`
  - Defines the Kp gain value of the d-current controller
- `#define KpQ`
  - Defines the Kp gain value of the q-current controller

- `#define KiD`
  - Defines the Ki gain value of the d-current controller

- `#define KiQ`
  - Defines the Ki gain value of the q-current controller

11 Software configuration

Below are listed the most important configuration parameters for the motor control software solution. They can be found in `user_defines.h` and shall be adapted for a new motor-inverter configuration.

11.1 Motor configuration

- `#define VDC_NOM`

– Defines the nominal voltage of the motor in V
Standard value is set to 24 V

- `#define RPM_RATED`

– Defines the rated speed of the motor in RPM
Standard value is set to 350 rpm

- `#define POLE_PAIR`

– Defines the number of pole pairs of the motor used to calculate electrical RPM of the motor. Standard value is set to 14

- `#define RS_MTR_LL`

– Defines the stator line-to-line resistance of the motor in Ω
Standard value is set to 5.5 Ω

- `#define LS_MTR_LL`

– Defines the stator line-to-line inductance of the motor in H
Standard value is set to 0.00272 H

11.2 Board configuration

- `#define ADC_RESOLUTION`

– Defines the resolution of the ADC in bit
Standard value is set to 12 bit

- `#define V_ADC_REF`

– Defines the reference voltage of the ADV in V
Standard value is set to 5 V

- `#define OP_AMP_IN_MAX`

– Defines the max. input base voltage of the current sense amplifier in V
Standard value is set to 0.3 V

- `#define OP_AMP_GAIN`

– Defines the gain of the current sense amplifier
  Standard value is set to 64

- `#define R_UPPER_VDC_DIV`

– Defines the resistance of the upper leg resistor of the voltage divider in Ω
  Standard value is set to 56.2 kΩ

- `#define R_LOWER_VDC_DIV`

– Defines the resistance of the lower leg resistor of the voltage divider in Ω
  Standard value is set to 2.61 kΩ

- `#define R_SHUNT`

– Defines the low-side shunt resistance value in Ω
  Standard value is set to 10 mΩ

11.3 Important functions

The main functions for initialization are `CrntFbk_FixedPoint_Init()` that initialized the current offset readings. `CAN_Initialize()` initialized the CAN communication of the board. `ControlLoop_FixedPointInit()` initialized the PI controller gains of both speed and current controllers. For each peripheral, individual initalization functions are used and executed at program start-up.

The main control execution is implemented inside `ControlLoop_FixedPoint()` that is executed every 50 µs under the function `Timer_Control_Loop()` and executes:

1. Current reading,
2. Position reading,
3. Speed computation,
4. Current transformation, and
5. PI controller execution as well as updating the space-vector modulator.

12 CAN communication

The board receives CAN messages setting the desired motor speed and returns the data from the angular sensor. Table 7 shows the CAN messages. Calibration can be triggered through CAN by a single-shot message. The speed is received and returned as a signed 16-bit integer, scaled to a maximum theoretical motor speed of 60 rad/s (573 rpm). This arbitrary choice is derived from a (arbitrary) maximum robot speed $V_{max,Robot}$ of 3 m/s, and considering the wheel radius r_{wheel} of 5 cm. The position is an unsigned 16-bit integer scaled to a maximum of 2π (full rotation). If no SPEED_COMMAND message is received for more than 125 ms, the speed is set to zero as a safety measure. For every physical signal contained in the data portion, the decoding description is given in italics:

- U/S: Unsigned/signed
- Factor: Factor with which the value in the CAN data must be multiplied to arrive at the physical value.
- Unit: The unit of the signal.

All signals use Big-endianness (Motorola convention) for their byte order.

The DIP switch setting on the board is used to set the position of the attached motor. The settings {0000, 0001, 0010, 0011} correspond to {front left (FL), front right (FR), back left (BL), back right (BR)}. From this setting, each board determines an individual CAN ID for SPEED_COMMAND and ENCODER_DATA.

Table 7 Motor control CAN messages

Command	ID	Byte # 0	1	2	3	4	5	6	7	Description	Rate
Speed_Command (FL|FR|BL|BR)	0x380 |0x381 |0x382 |0x383	Speed *S, 1.8315e-03, rad/s*		Position* *U, 9.5871e-05, rad*						Speed command to specified motor. Speed Factor: $\frac{3}{2^{15} \cdot 0.05} = \frac{V_{max,Robot}}{2^{15} \cdot r_{wheel}}$	10 Hz
Encoder_Data (FL|FR|BL|BR)	0x400 |0x401 |0x402 |0x403	Speed *S, 1.8315e-03, rad/s*		Position *U, 9.5871e-05, rad*						Data sent by specified motor. Speed Factor: see above Position Factor: $\frac{2\pi}{2^{16}}$	100 Hz
Calibration_Req_All_Motors**	0x540									Triggers offset calibration on all motors, make sure robot is off the ground	On demand

* Not currently implemented

** Only possible if the auto-calibration function in the software is enabled

References

[1] Infineon Technologies AG: *MOTIX™ IMD701A datasheet*; Available online

[2] Infineon Technologies AG: *OptiMOS™ 6 ISZ053N08NM6 datasheet*; Available online

[3] Infineon Technologies AG: *XMC1404 Arm® Cortex®-M0 datasheet*; Available online

[4] Infineon Technologies AG: *TLE9351VSJ datasheet*; Available online

[5] Infineon Technologies AG: XENSIV™ TLI5012B E1000 *datasheet*; Available online

[6] *GL60 24V Gimbal System Motor-KV55*; Available online

[7] Infineon Technologies AG: *AP0803620 - XC866 Optimized Space Vector Modulation and Over-modulation with the XC866*; Available online

Revision history

Document revision	Date	Description of changes
V 1.0	2024-04-08	Initial release
V 2.0	2025-04-24	Updated software documentation and CAN communication

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

Edition 2025-04-24

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

UG091243

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.