Watt & Well
CONVERTING POWER INTO CONFIDENCE

## NAR MOTION SERIES

# High-Temp – Medium Power
# Nar Motion M

HIGH TEMPERATURE | OIL & GAS APPLICATION

## OVERVIEW

Optimize your system by selecting our bestseller, Nar Motion M, that includes FOC and HV capabilities. It is part of the Nar Motion series.

Nar Motion is our innovative high-temperature motor controller series, providing optimal operation **and high-reliability in harsh environment**. Typical applications are Downhole Tools for Oil&Gas and Geothermal markets. Nar Motion M is the **cost-effective version for Medium Power applications.**

**Standard** version includes pre-programmed firmware with more than hundred of parameters to tune the controller.

**-MBD** version includes firmware as a provided Simulink Model. The control loop is easy to customize via a Simulink Model and Embedded Coder code generation.

## FEATURES

- High Temperature: up to 175 ºC (347 ºF)
- High Vibration: up to 20 G RMS
- Typical Power: 200 W to 3 kW
- Bus Voltage: up to 600 V
- Phase Current: up to 6 A RMS
- Field Oriented Control for optimum system efficiency with PMSM motors
- Sensorless, Hall effect or Resolver
- Four-quadrant operation with independent speed, power and torque limits for easy motor paralleling
- Full digital control and monitoring via CAN 2.0B bus
- Advanced system protection and monitoring
- Highly Configurable through an extensive Graphical User interface with logging capability

Lx W x H (mm)
292x40x24

## DIMENSIONS

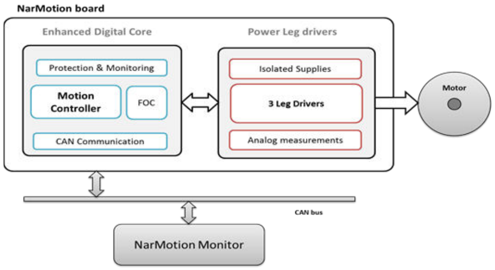
| Units | **LxWxH** |
|---|---|
| **mm** | 292 x 40 x 24 |
| **inches** | 11.49 x 1.57 x 0.94 |

## VERSIONS

| Part-number | **Rotor position sensor** |
|---|---|
| **NM-40-600-6-RESO** | Resolver sensor |
| **NM-40-600-6-HALL** | Hall Effect sensor |
| **NM-40-600-6-SLES** | Sensorless |

-**MBD** version includes a customizable firmware as a Simulink Model.

## BLOCK DIAGRAM



## APPLICATIONS

- Drilling
- Wireline

## CUSTOMIZATION OPTIONS

- On demand

**WARNING**

This equipment operates at voltages and currents that can result in electrical shock, fire hazard and/or personal injury if not properly handled or applied. Equipment must be used with necessary caution and appropriate safeguards employed to avoid personal injury or property damage.

This board must be used only by qualified engineers and technicians' familiar with risks associated with handling high voltage electrical and mechanical components, systems and subsystems.

**TABLE OF CONTENTS**

# 1. Absolute maximum ratings

| Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|
| HV Input Voltage (motor driver supply) | $V_{HV}-V_{HV\_RTN}$ | 0 | 700 | V |
| HV Input Voltage common mode | $V_{HV\_RTN}-V_{chassis}$ | -2 | +2 | V |
| HV Input Current | | | 6 | A |
| HV Input Power | | | 3000 | W |
| Phase Current | | | 12 | A |
| LV Input Voltage (control electronics supply) | $V_{LV}-V_{LV\_RTN}$ | 0 | 30 | V |
| LV Input Voltage common mode | $V_{LV\_RTN}-V_{chassis}$ | 0 | 0 | V |
| Operating Temperature[1] | | 5 | 185 | °C |
| Long term storage Temperature | | 5 | 85 | °C |
| Temperature change rate | | | 4 | °C/min |

**Table 1: Absolute maximum ratings**

[1] Temperature provided is the chassis temperature.

## 2. Electrical Characteristics

All specifications are given for the full temperature range unless otherwise noted

| Parameter | Condition | Value | | | Units |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| **General** | | | | | |
| Motor Type | Three phase synchronous motor | | | | |
| Driver Efficiency | | - | 97 | - | % |
| Phase continuous output current | T = 175 °C | - | - | 6 | Arms |
| Phase instantaneous output current | T = 175 °C | - | - | 10 | Apeak |
| Nominal speed Range | With 4-pole-pair motor<br>With 2-pole-pair motor | -7 500<br>-15 000 | - | 7 500<br>15 000 | rpm |
| Switching frequency | | - | 10 | - | kHz |
| Frequency control accuracy | | - | - | 1 | % |
| Thermal Shutdown[2] | | Software Programmable | | | |
| Resolver sensor excitation | No load | 5.7 | 6 | 6.3 | $V_{RMS}$ |
| Resolver sensor excitation current | | - | 50 | 75 | mA |
| Resolver sensor excitation frequency | | 9.9 | 10 | 10.1 | kHz |
| Resolver sensor input | | 0 | | 3.3 | V |
| Resolver input impedance | | 2.27 | 2.3 | 2.33 | kΩ |
| CAN baudrate | | 1000 | | | kbauds |
| CAN differential output (dominant) | $CAN_H - CAN_L$ | 3.3V compatible | | | |
| CAN common mode range | $(CAN_H + CAN_L)/2 - V_{LV\_RTN}$ | -2 | - | +3 | V |
| CAN differential output (dominant) | CAN transceiver powered with 3.3V | 1.1 | - | - | V |
| CAN differential input voltage | (compatible with external transceivers from 3.3 to 5V) | -5 | - | +5 | V |
| **HV input** | | | | | |
| Input voltage | | - | 600 | 625 | V |
| Input current | | - | - | 5.0 | $A_{RMS}$ |

[2] Optional Thermal shutdown requires internal or external temperature sensors and is software programmable. External sensor (e.g. motor temperature) is not included

| | | | | | |
|---|---|---|---|---|---|
| Under Voltage Shutdown (factory programmable) | | 200 | - | - | V |
| Over Voltage Shutdown (factory programmable) | | - | - | 650 | V |
| **LV input** | | | | | |
| Input voltage | | 20 | 24 | 28 | V |
| Input current | T = 150 °C – input voltage = 24V | - | 0.25 | 0.3 | A |
| Under Voltage Shutdown | T = 25 °C<br>T = 175 °C | 19.5<br>19 | 20 | 20.5<br>21 | V |
| Over Voltage Shutdown | T = 25 °C<br>T = 175 °C | 29.5<br>29 | 30 | 30.5<br>31 | V |
| **Input / Output** | | | | | |
| Hall effect supply output | Max 20mA | 4.8 | 5 | 5.2 | V |
| Hall effect senses input | Open drain sensors | 0 | | 3.3 | V |
| External temperature measurement | | | | | |
| Pressure voltage supply output (Pout_P - Pout_N) | Max 5 mA | 4.8 | 5 | 5.2 | V |
| Pressure sensor input common mode | | 0 | | 3.3 | V |
| Pressure sensor input differential mode | | 0 | | 3.3 | V |
| ASIO | Typical impedance 20kΩ | 0 | - | 6.6 | V |
| DSIO | Input mode | 0 | | 3.3 | V |
| DSIO Low-level input voltage ($V_{IL}$) | Input mode | 0 | | 0.55 | V |
| DSIO High-level input voltage ($V_{IH}$) | Input mode | 1.9 | | 3.3 | V |
| DSIO | Output mode | 0 | | 3.3 | V |
| DSIO | Output mode | 0 | | 10 | mA |

**Table 2: Electrical characteristics**

# 3. Theory of operation

The Nar Motion-M module board consists of a power stage with three phase legs and a control stage with the following blocks:

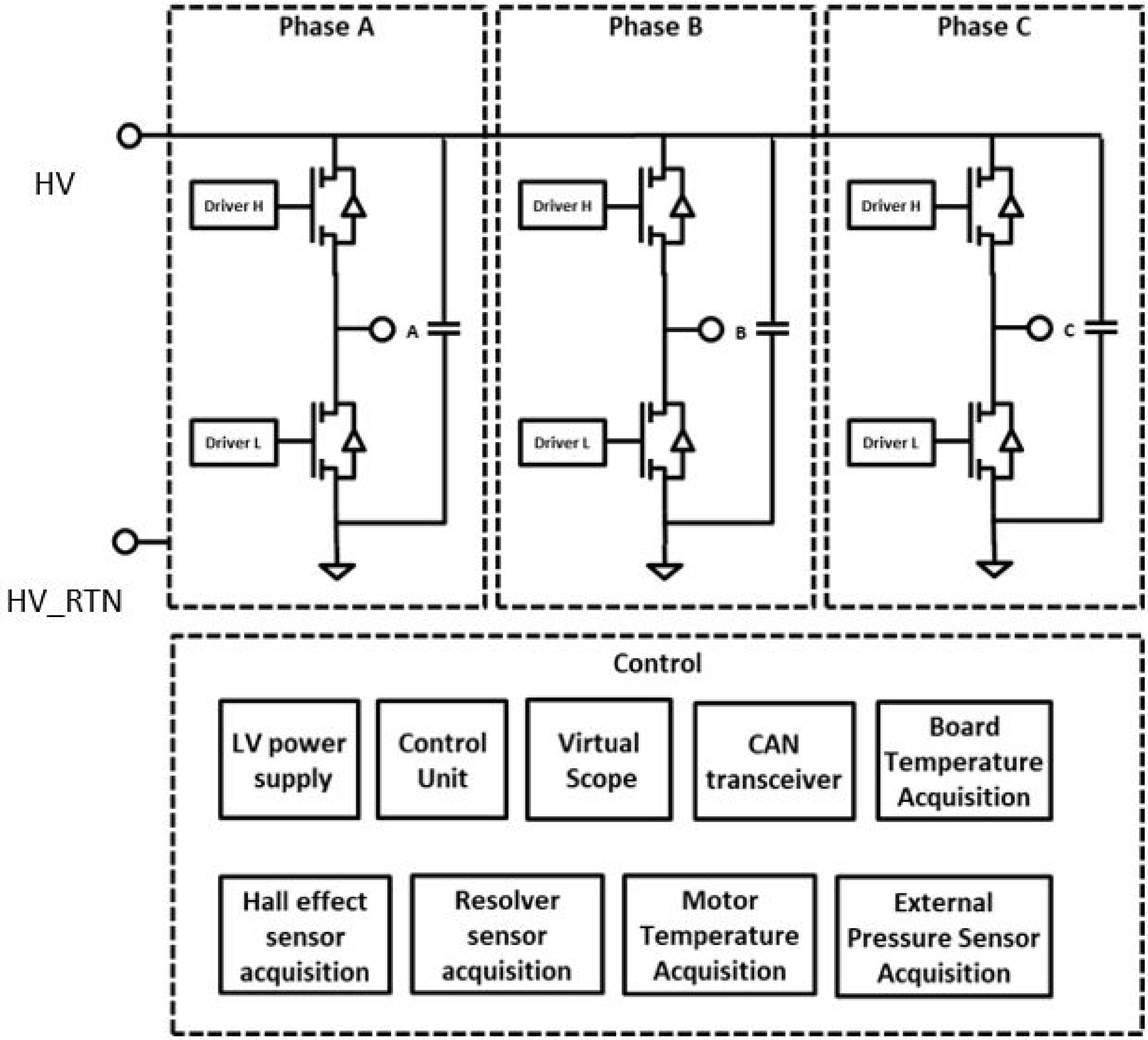

Phase A
Phase B
Phase C
HV
HV_RTN
Driver H
Driver L
A
B
C
Control
LV power supply
Control Unit
Virtual Scope
CAN transceiver
Board Temperature Acquisition
Hall effect sensor acquisition
Resolver sensor acquisition
Motor Temperature Acquisition
External Pressure Sensor Acquisition

**Figure 1: Nar Motion-M block diagram**

## 3.1. CAN Transceiver

Non-isolated CAN transceiver to communicate through a high speed CAN. The driver itself is powered at 3.3V (CMOS levels) but it accepts inputs from CAN transceiver powered at 5V (TTL or CMOS levels). CAN bus is expected to be terminated at both ends with a resistance matching the intrinsic impedance of the twisted pair used (typically 120 Ω). By default, Nar Motion-M has a 120 Ω resistor internally connected.

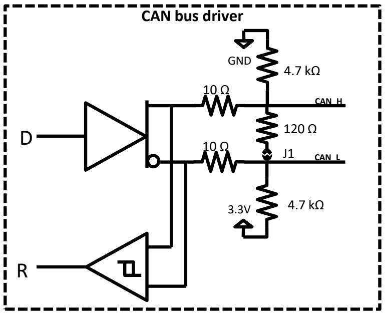


**Figure 2: CAN transceiver simplified diagram**

## 3.2. Resolver Acquisition

The resolver provides the rotor angle position. The rotating speed is derived from this position. This bloc includes the excitation to the resolver sensor and the acquisition chain for the sinus and cosinus outpus of the sensor

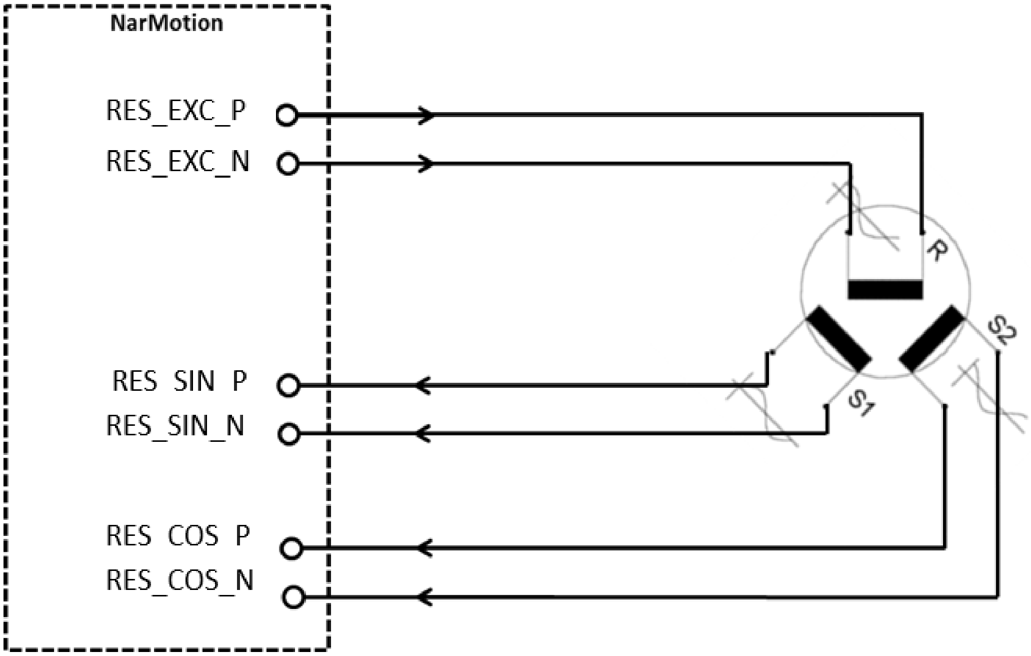


**Figure 3: Simplified resolver sensor intergace diagram**

## 3.3. Hall Sensor Acquisition

Nar Motion M product provides power to the hall sensors (5V, 20mA) and acquires each sensor state. This block is compatible with Open Drain/Open Collector output. Typical pull-up resistor used is 10kΩ. A 10nF capacitor is also included. For noise immunity, proper wiring and shielding is strongly advised for best results.

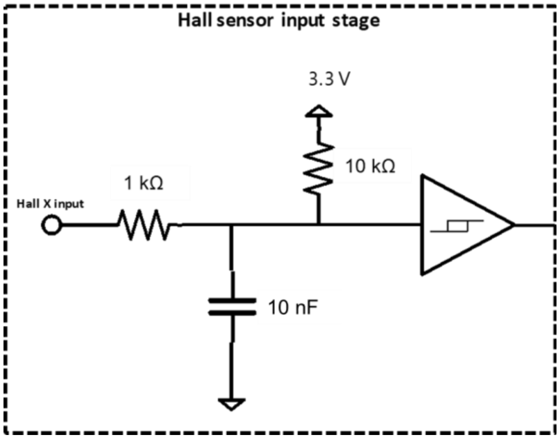


**Figure 4: Hall Effect sensor interface simplified diagram**

## 3.4. Digital Signal Input / Output (DSIO0 and DSIO1)

Two digital signal input/output are accessible from P2 connector and configurable through the DSP. Both digital input/output are bufferized.

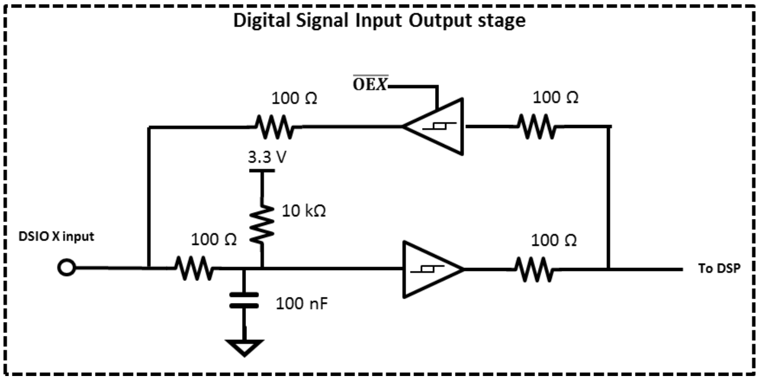


**Figure 5: DSIO sensor interface simplified diagram**

## 3.5. Ambient Temperature Sensor

A temperature sensor is located on the power board.

## 3.6. External Temperature Sensor

An acquisition circuit is located on the board to measure precisely a typical 1 kΩ sensor through Wheatstone bridge circuit.
This circuit is designed to measure the temperature with a PT1000 sensor.

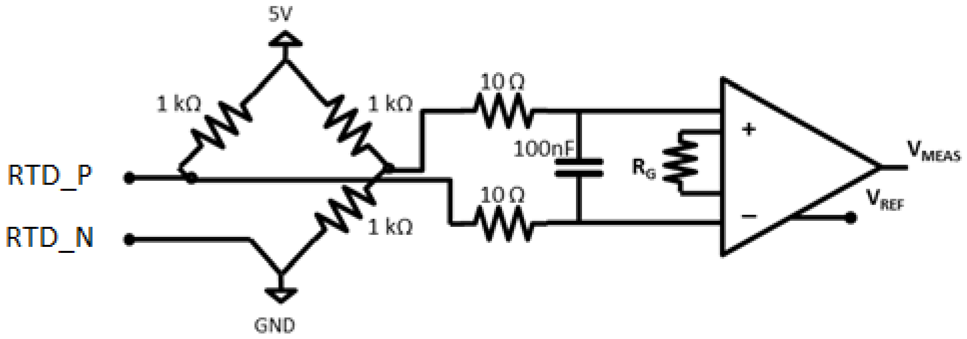


**Figure 6: Acquisition circuit for External Temperature Sensor**

## 3.7. External Pressure Sensor

One acquisition circuit is located on the board to precisely measure a strain gauge sensor, with a minimum sensor impedance of 1 kOhm.
Nar Motion M product provides a power supply to the pressure sensor (5V, 0.8mA).

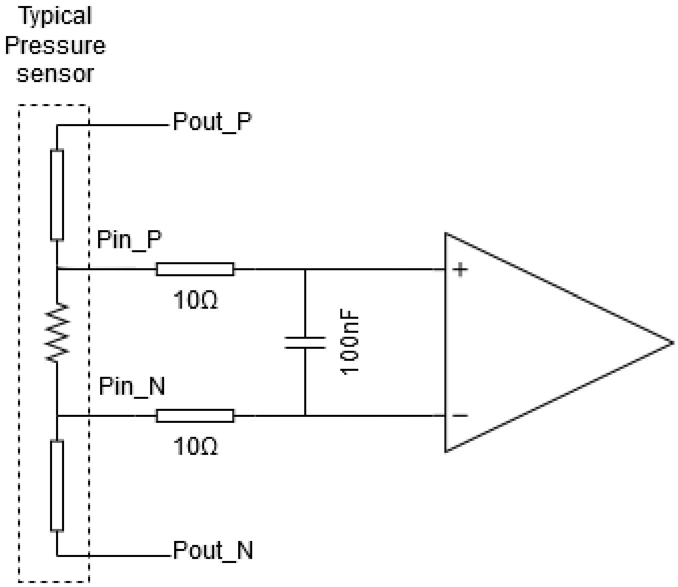


**Figure 7: Acquisition circuit for External Pressure Sensor**

# 4. Functional overview

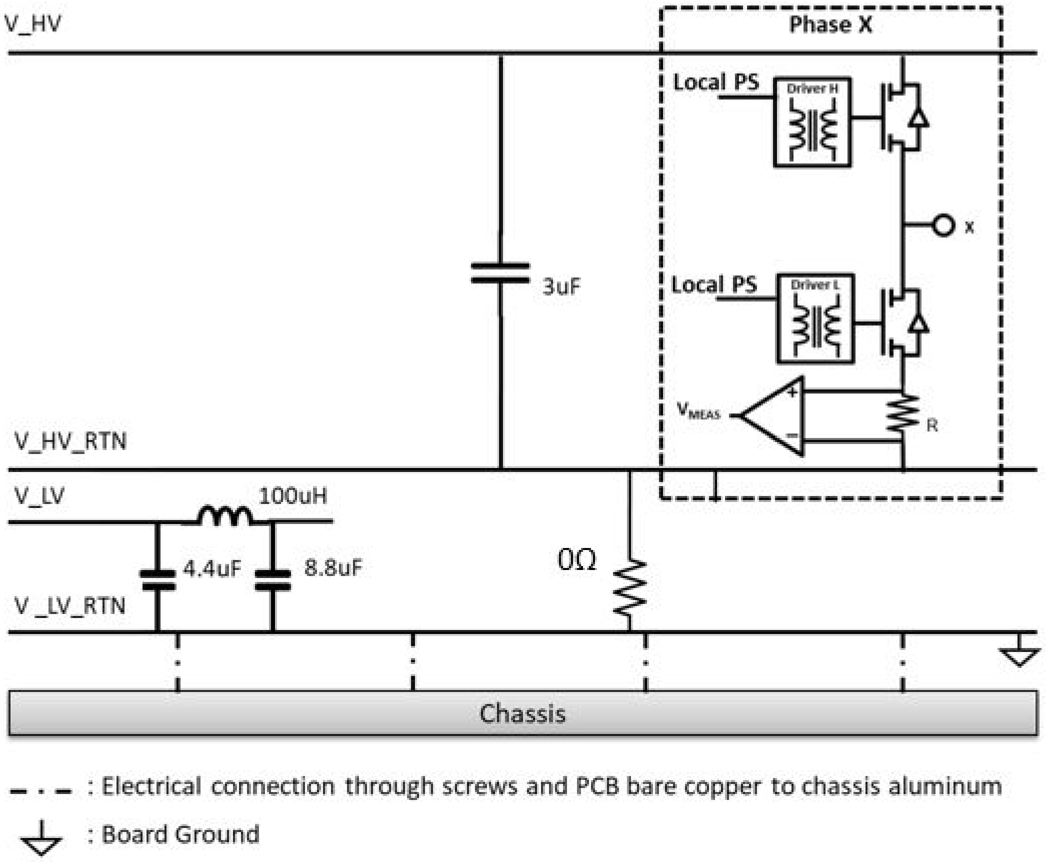
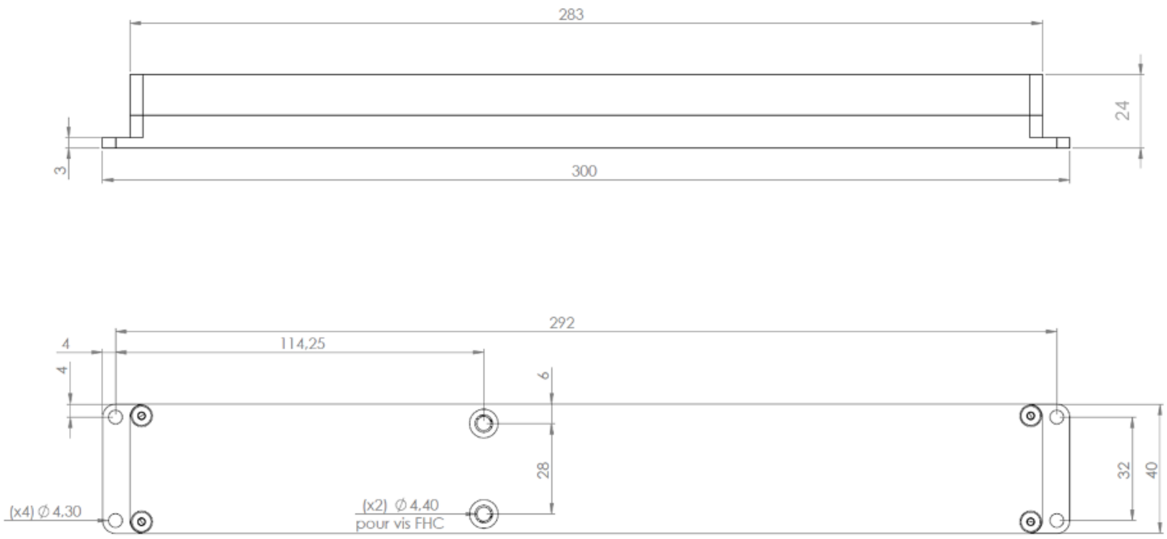


**Figure 8: Internal design of Nar Motion-M**

# 5. Mechanical specifications



All dimensions in mm
General tolerances: ISO 2768 mK

# 6. Wiring connections

## 6.1. Low power signals

The connectors described in this section are used for low voltage power interface.

- A 9 pin connector (MDAL09SCBPPT1 from AXON CABLE) is used to interface the system through Low Voltage Input, CAN Communication and a few GPIO and analog signals, on the left side of the PCB.
  Its recommended matting micro-sub D connector is MDAL09PCWxxxxx from AXON CABLE.
- A 21 pin connector(MDAL21SCBPPT1 from AXON CABLE) is used to interface the motor sensors,on the right side of the PCB.
  Its recommended matting micro-sub D connector is MDAL21PCWxxxxx from AXON CABLE.

Pin-outs are shown in Table 3 and Table 4.

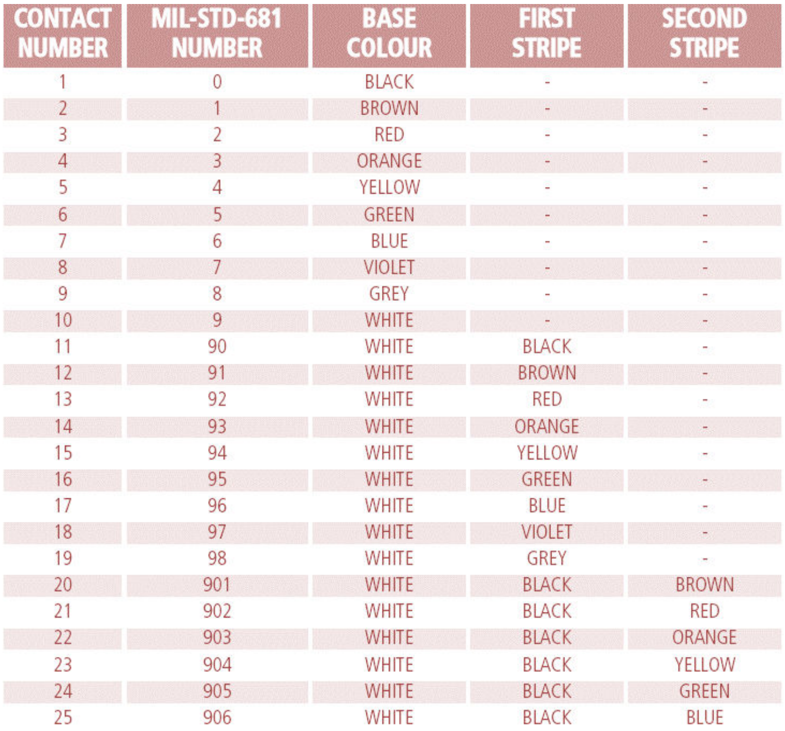
| Pin | Signal Name | Details | Harness - MIL-STD-681 Wire Color |
|---|---|---|---|
| 1 | RES_EXC_Shield | Resolver excitation shield | Black |
| 2 | RES_SIN_N | Resolver input sinus - | Brown |
| 3 | RES_SIN_P | Resolver input sinus + | Red |
| 4 | RES_COS_Shield | Resolver cosinus shield | Orange |
| 5 | POUT_P | 5V power supply for pressure sensor | Yellow |
| 6 | PIN_P | Pressure sensor positive input | Green |
| 7 | RTD_shield | RTD and/or pressure sensor and/or hall shield (internally connected to ground) | Blue |
| 8 | PIN_N | Pressure sensor negative input | Purple |
| 9 | Hall_GND | 5V return to ground for hall sensors | Grey |
| 10 | Hall_C | Hall effect input C | White |
| 11 | Hall_A | Hall effect input A | White – Black |
| 12 | RES_EXC_N | Resolver excitation - | White – Brown |
| 13 | RES_EXC_P | Resolver excitation + | White – Red |
| 14 | RES_SIN_Shield | Resolver sinus shield | White – Orange |
| 15 | RES_COS_N | Resolver cosinus - | White – Yellow |
| 16 | RES_COS_P | Resolver Cosinus + | White – Green |
| 17 | POUT_N | 5V return to ground for pressure sensor | White – Blue |
| 18 | RTD_N | External temperature sensor signal - | White – Purple |
| 19 | RTD_P | External temperature sensor signal + | White – Grey |
| 20 | Hall_B | Hall effect input B | White – Black – Brown |
| 21 | Hall 5V_Out | 5V power supply for Hall Sensors | White – Black – Red |

**Table 3: P1 pinout (motor interface)**

| Pin | Signal Name | Details | Harness - MIL-STD-681 Wire Color |
|---|---|---|---|
| 1 | CAN_L | CAN bus low | Black |
| 2 | ASI0 | Analog Input 0 | Brown |
| 3 | ASI0_RTN | Analog Input RTN (GND) | Red |
| 4 | DSIO0 | Configurable Digital Signal Input/Output 0 | Orange |
| 5 | LV_RTN | LV supply – (GND) | Yellow |
| 6 | CAN_H | CAN bus high | Green |
| 7 | CAN_Shield | CAN bus shield. This pin is internally connected to GND. | Blue |
| 8 | DSIO1 | Configurable Digital Signal Input/Output 1 | Purple |
| 9 | LV_P | LV supply + | Grey |

**Table 4: P2 pinout (external interface)**

Depending on the connector used for P1, it may use a color code following MIL-STD-681 specification. Table 5: MIL-STD-681 Wire Color Code is included for reference purposes.



| CONTACT NUMBER | MIL-STD-681 NUMBER | BASE COLOUR | FIRST STRIPE | SECOND STRIPE |
|---|---|---|---|---|
| 1 | 0 | BLACK | - | - |
| 2 | 1 | BROWN | - | - |
| 3 | 2 | RED | - | - |
| 4 | 3 | ORANGE | - | - |
| 5 | 4 | YELLOW | - | - |
| 6 | 5 | GREEN | - | - |
| 7 | 6 | BLUE | - | - |
| 8 | 7 | VIOLET | - | - |
| 9 | 8 | GREY | - | - |
| 10 | 9 | WHITE | - | - |
| 11 | 90 | WHITE | BLACK | - |
| 12 | 91 | WHITE | BROWN | - |
| 13 | 92 | WHITE | RED | - |
| 14 | 93 | WHITE | ORANGE | - |
| 15 | 94 | WHITE | YELLOW | - |
| 16 | 95 | WHITE | GREEN | - |
| 17 | 96 | WHITE | BLUE | - |
| 18 | 97 | WHITE | VIOLET | - |
| 19 | 98 | WHITE | GREY | - |
| 20 | 901 | WHITE | BLACK | BROWN |
| 21 | 902 | WHITE | BLACK | RED |
| 22 | 903 | WHITE | BLACK | ORANGE |
| 23 | 904 | WHITE | BLACK | YELLOW |
| 24 | 905 | WHITE | BLACK | GREEN |
| 25 | 906 | WHITE | BLACK | BLUE |

**Table 5: MIL-STD-681 Wire Color Code**

### a) Recommended connection for CAN bus

CAN bus signals can be connected to a 9 pin Sub-D socket connector (also known as female DB9) for easy interface with commercial CAN transceivers (such as NI CAN 8473 which features a DB9 header connector). The pin-out of such connector is as follow :

| Signal Name | Pin in P2 | Pin in DB9 (female) |
|---|---|---|
| **CAN_H** | 6 | 7 |
| **CAN_L** | 1 | 2 |
| **CAN_Ref** | 7 | 3 |
| **CAN_Shield** | 7 | 5 |

**Table 6: CAN pinout on external connector**

**CAN bus shield**

Although ISO-11898-2 does not specify the wires type or the need for a shield, a shielded cable is recommended for electronically harsh environments. It is recommended to ground the shield at a single point on one of the dedicated CAN_Shield pins of the Nar Motion boards to avoid ground loops.

## 6.2. High power signals

### a) Power signals colors

The 5 high power signals (DC link and Phases) are accessible using flying wires. (Length 350mm)

| Signal Name | Side | Color | AWG |
|---|---|---|---|
| HV_IN | Left | Red | 18 |
| HV_RTN | Left | Black | 18 |
| PH_A | Right | Red | 18 |
| PH_B | Right | White | 18 |
| PH_C | Right | Black | 18 |

**Table 7: High power signal wires**

### b) Grounding recommendation

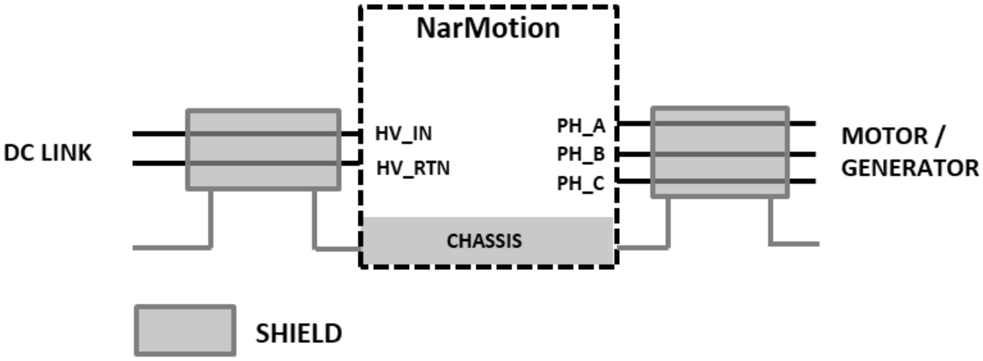


**Figure 9: Grounding recommendation**

# 7. Firmware specifications

## 7.1. CAN protocol

The motor driver can be operated by a CAN bus communication.

Then from the motor point of view, in addition to the power supply and its load, the speed, torque and power consumption of the motor can also be limited by the Nar Motion CAN protocol.

The Nar Motion CAN protocol is a simple protocol designed for **real time efficiency** (low CPU overhead) and **simplicity** (no communication stack required). Nar Motion protocol is fully documented and the protocol can be reused on other projects without licensing restrictions.

When using this protocol, Nar Motion boards includes severals functionalities:

- High level monitoring of motor control driver
- Real time scope functionalities available

## 7.2. Digital bus configuration

Each motor controller features a CAN transceiver to communicate with a bus master. In this Master/Slave scheme, each motor controller is considered as a slave, while the master can be either a Graphical User Interface in a PC or another electronic board.

CAN bus is configured to respect the standard 2.0B (29-bit message identifier) at a default baudrate of **1000 kbps** with sampling point at **80**%. For best result, it is advised that all CAN nodes use the same sampling point configuration. Herunder an example of CAN configuration with a CPU clock of 90MHz:

- Clock for CAN peripheral : 15MHz (90MHz/(2*(BRPREG + 1)) with BRPREG = 2 in the Nar Motion controller)
- Time before sample ($TSEG_{reg}1$ + 1) = 11
- Time after sample ($TSEG_{reg}2$ + 1) = 3

## 7.3. Non-volatile memory

Nar Motion board enables to store parameters configuration (ie calibration, motor settings) in non volatile memory, for easy deployment.

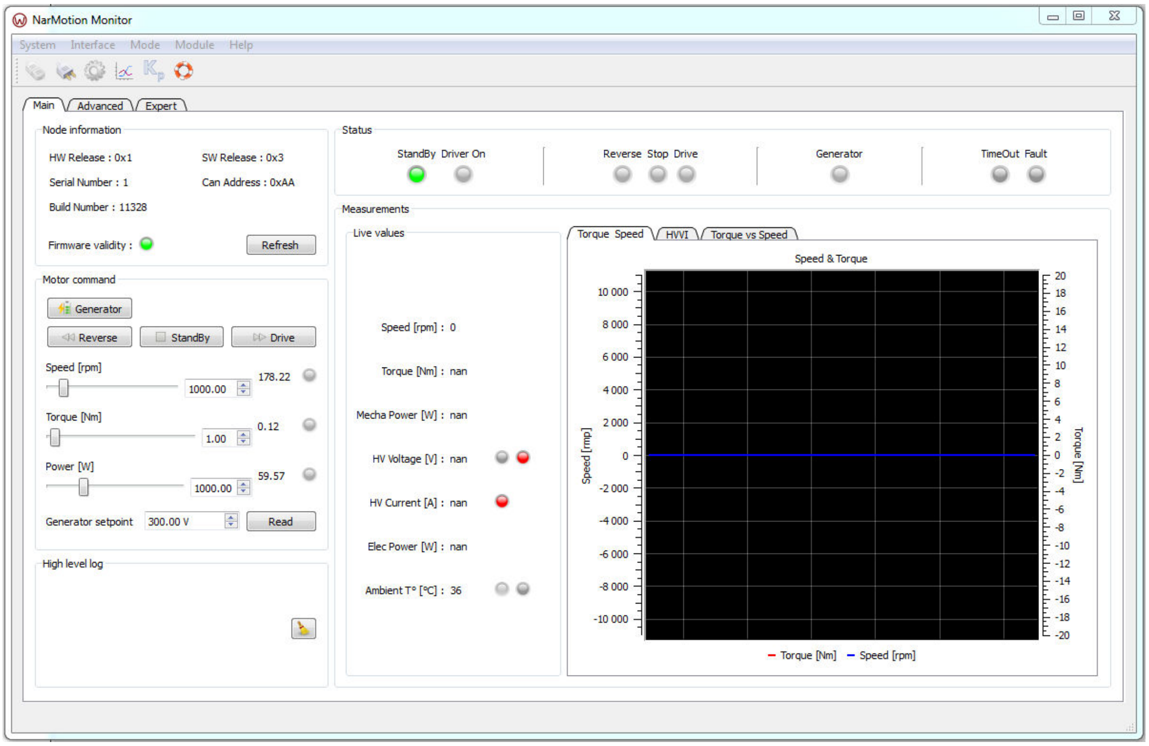
## 7.4. CAN Programming Capability

The firmware embeds a bootloader to be able to program the board on the fly.

## 7.5. Graphical User Interface

The product is provided with a graphical user interface that contains a large variety of functionality:

- Basic Operation : enable start /stop the drive and to set the speed, torque and power setpoint, and the get main measurements from the Nar Motion board.
- Virtual Scope : enable to trigger and download measurements (phase currents, position, …) and internal signals (id, iq, …) at the control rate (10kHz).
- Parameters update : enable to tune the whole board parameters, including controllers loops constants, limitations, calibrations, ect.
- CAN Programming

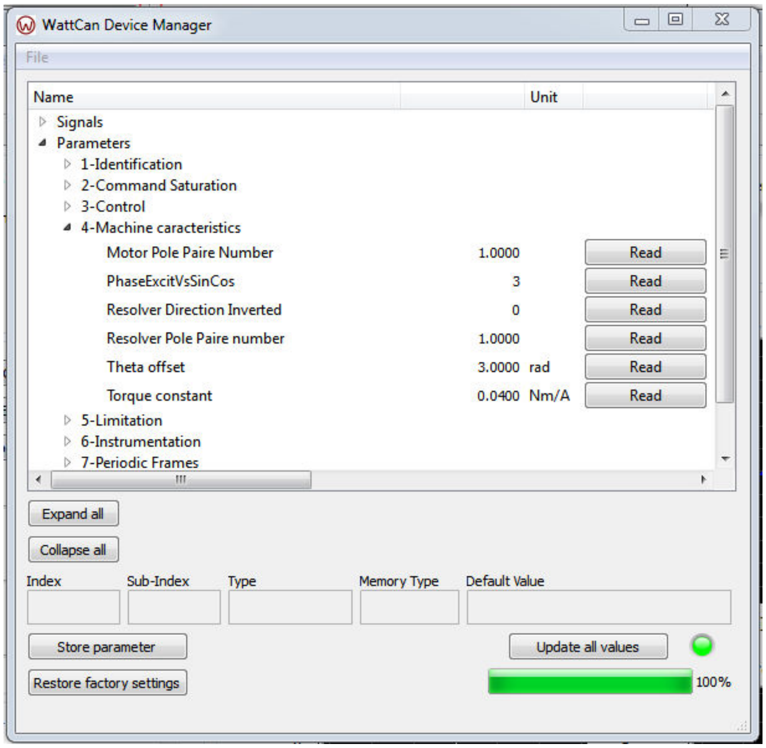


HW Release : 0x1
SW Release : 0x3
Serial Number : 1
Can Address : 0xAA
Build Number : 11328

Speed [rpm] 1000.00 178.22
Torque [Nm] 1.00 0.12
Power [W] 1000.00 59.57
Generator setpoint 300.00 V

Speed [rpm] : 0
Torque [Nm] : nan
Mecha Power [W] : nan
HV Voltage [V] : nan
HV Current [A] : nan
Elec Power [W] : nan
Ambient T° [°C] : 36

**Figure 10: GUI Main panel**

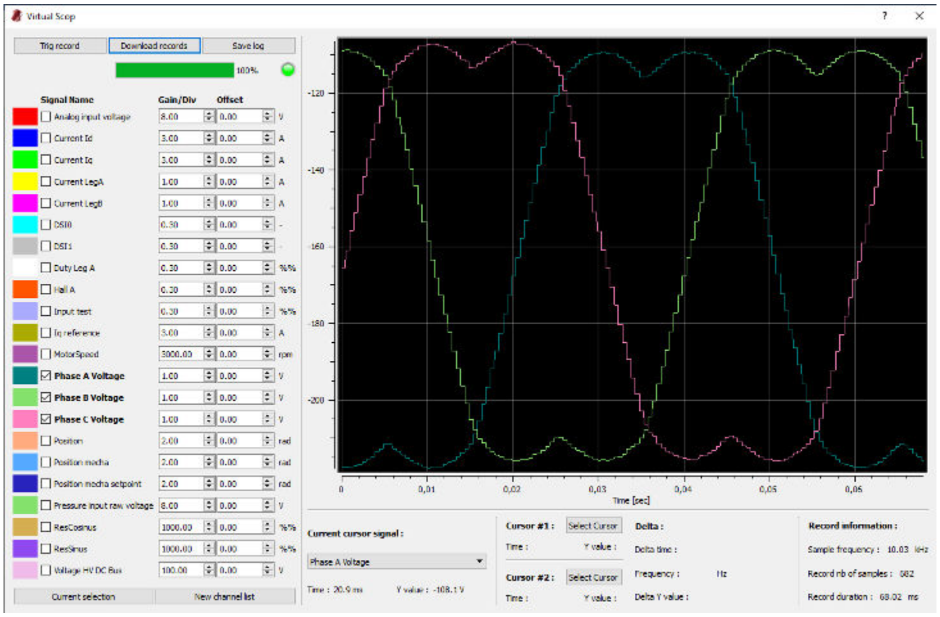


- Signals
- Parameters
  - 1-Identification
  - 2-Command Saturation
  - 3-Control
  - 4-Machine caracteristics
  - 5-Limitation
  - 6-Instrumentation
  - 7-Periodic Frames

| Name | | Unit | |
|---|---|---|---|
| Motor Pole Paire Number | 1.0000 | | Read |
| PhaseExcitVsSinCos | 3 | | Read |
| Resolver Direction Inverted | 0 | | Read |
| Resolver Pole Paire number | 1.0000 | | Read |
| Theta offset | 3.0000 | rad | Read |
| Torque constant | 0.0400 | Nm/A | Read |

| Index | Sub-Index | Type | Memory Type | Default Value |
|---|---|---|---|---|
| | | | | |

**Figure 11: Parameters panel**



| | Signal Name | Gain/Div | Offset | |
|---|---|---|---|---|
| 🟥 | ☐ Analog input voltage | 8.00 | 0.00 | V |
| 🟦 | ☐ Current Id | 3.00 | 0.00 | A |
| 🟩 | ☐ Current Iq | 3.00 | 0.00 | A |
| 🟨 | ☐ Current LegA | 1.00 | 0.00 | A |
| 🟪 | ☐ Current LegB | 1.00 | 0.00 | A |
| 🟦 | ☐ DSI0 | 0.30 | 0.00 | - |
| ⬜ | ☐ DSI1 | 0.30 | 0.00 | - |
| ⬜ | ☐ Duty Leg A | 0.30 | 0.00 | %% |
| 🟧 | ☐ Hall A | 0.30 | 0.00 | %% |
| 🟪 | ☐ Input test | 0.30 | 0.00 | %% |
| 🟨 | ☐ Iq reference | 3.00 | 0.00 | A |
| 🟪 | ☐ MotorSpeed | 3000.00 | 0.00 | rpm |
| 🟦 | ☑ **Phase A Voltage** | 1.00 | 0.00 | V |
| 🟩 | ☑ **Phase B Voltage** | 1.00 | 0.00 | V |
| 🟥 | ☑ **Phase C Voltage** | 1.00 | 0.00 | V |
| 🟧 | ☐ Position | 2.00 | 0.00 | rad |
| 🟦 | ☐ Position mecha | 2.00 | 0.00 | rad |
| 🟦 | ☐ Position mecha setpoint | 2.00 | 0.00 | rad |
| 🟩 | ☐ Pressure input raw voltage | 8.00 | 0.00 | V |
| 🟨 | ☐ ResCosinus | 1000.00 | 0.00 | %% |
| 🟪 | ☐ ResSinus | 1000.00 | 0.00 | %% |
| 🟪 | ☐ Voltage HV DC Bus | 100.00 | 0.00 | V |

**Current cursor signal :**
Phase A Voltage
Time : 20.9 ms Y value : -108.1 V

**Cursor #1 :**
Time : Y value :

**Cursor #2 :**
Time : Y value :

**Delta :**
Delta time :
Frequency : Hz
Delta Y value :

**Record information :**
Sample frequency : 10.03 kHz
Record nb of samples : 682
Record duration : 68.02 ms

**Figure 12: Virtual scope panel**

# 8. Simulink Model Based Design Overview.

This Simulink model is accessible and customizable only for products with -**MBD** option.

## 8.1. Control Strategy

### a) FOC

The torque loop is based on Filed Oriented Control (FOC) and is shared between inverter and rectifier modes. It contains :

- **Clark transformation:** to transform currents from three phase abc frame to two phase αβ frame
- **Park transformation:** to transform signals from rotational αβ frame to stationary dq frame
- **Currents control loops:** Anti-windup PID controllers for Id and Iq currents.
- **Space vector modulation:** to calculate control duty cycles from alpha/beta voltages

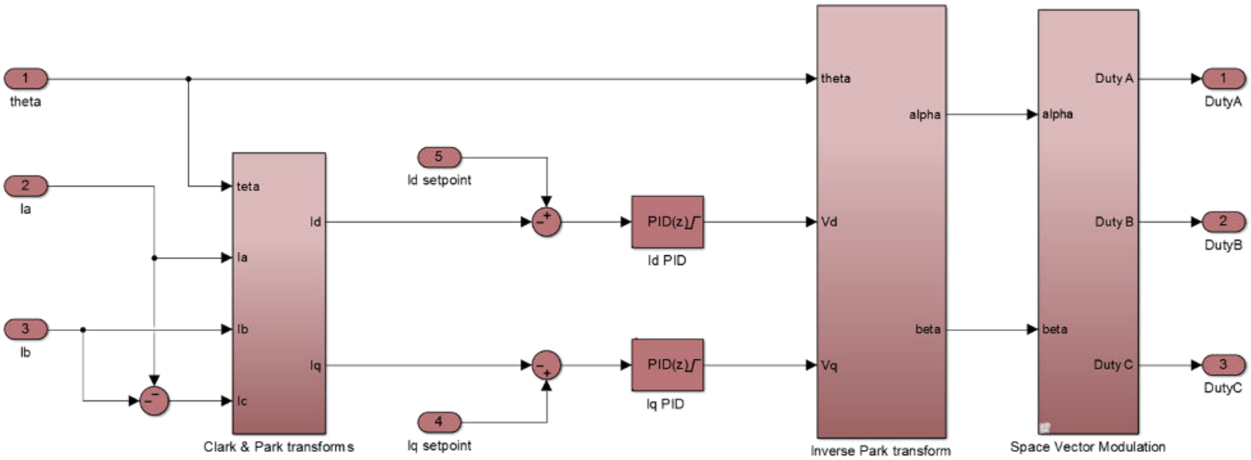


**Figure 13 Field oriented control scheme**

### b) Inverter

This structure is composed of a speed loop that feeds the torque for the Field Oriented Control loops.
The power can be limited with torque limitation through speed measurements.
The speed loop is running at 1.6 kHz (Medium frequency).
The torque loop is running at 10kHz (Fast frequency).

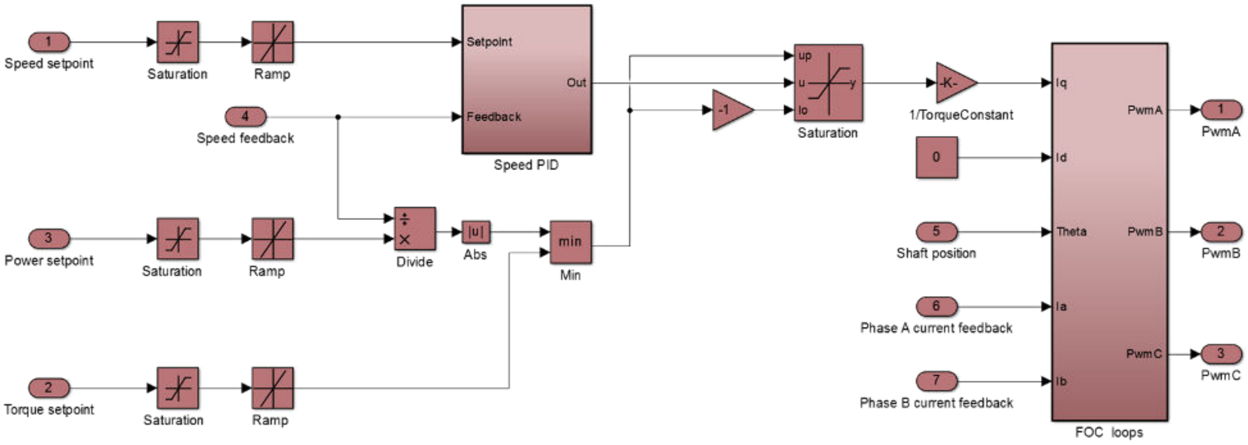


**Figure 14 Inverter control structure**

## 8.2. Simulink models presentation

### 8.2.1. Control Block Model

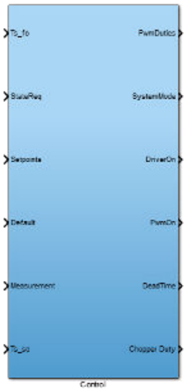


**Figure 15 Field oriented control scheme**

The control block contains all the control loops for simulation model and code generation.

### 8.2.2. Simulation model

For this product, there are 2 simulation models. One model for the inverter and one model for the rectifier.

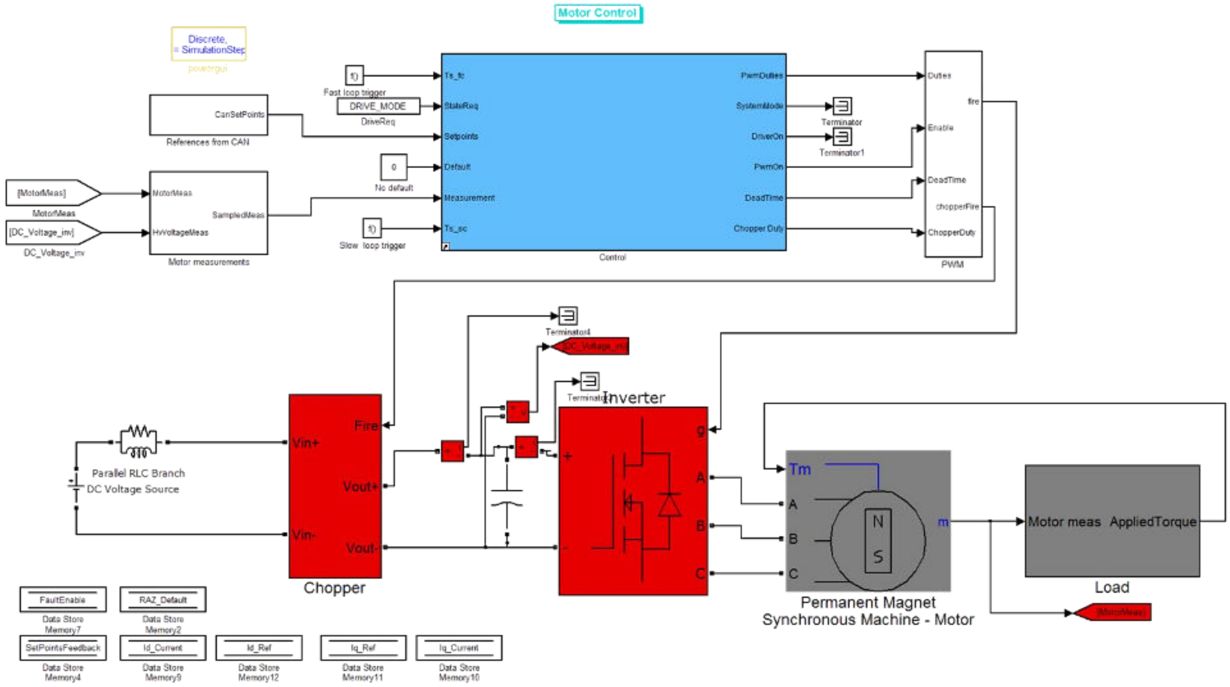


**Figure 16 Inverter's model simulation**

### 8.2.3. Code generation model

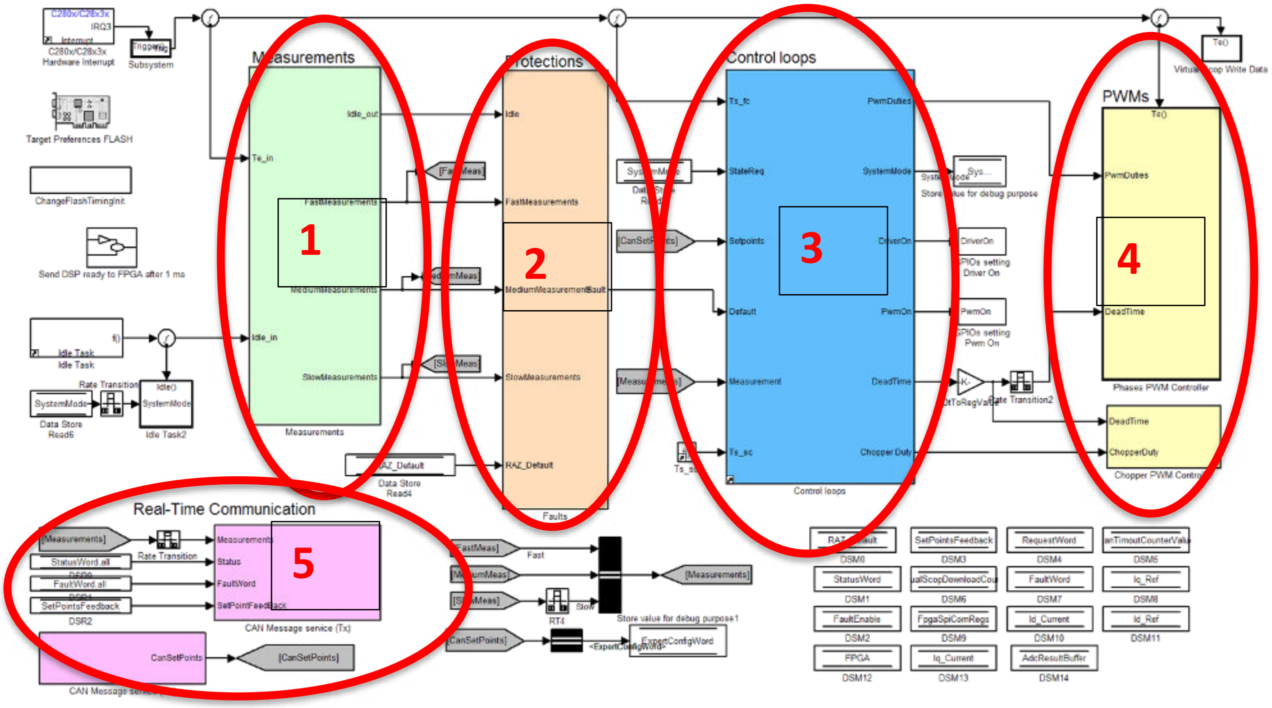


**Figure 17 Model for code generation**

1. **Measurements block** : This block handles the calibration corrections such as the current/voltage gain/offset, converts and adapts the analog inputs into digital signals. It delivers output information under 3 sample frequencies :
    a. Fast : 10kHz for the phases current/position
    b. Medium : 1,67 kHz for the voltages, speed, torque
    c. Slow : Idle task like for example temperatures
2. **Protection block** : Its purpose is to compare the measurements received to the predefined thresholds to trig protections if over/under voltages/currents or thermal shutdown occur. Each of these protections can be enabled or disabled through the CAN. This block can treat the information for the 3 sample times.
3. **Control block** : The control block used here is a library, it is also instantiated in the simulation model (Rectifier/Inverter). It contains the state machine of the system and speed, voltage, current control loops. The main parameters can be enable/disabled by CAN. This block takes the parameters of 2 sample times : Fast and Medium.
4. **PWM driver block** : Generates 10kHz PWM (Pulse Width Modulation) and is complementary with dead time.
5. **Real time communication block** : This final block enables the user to set parameters such as main parameters or setpoints, and get these main parameters or even the measurements. The baud rate has been fixed to 1000 kbps and the extended CAN frame format 2.0B (29 bits).

# 9. Safety instruction

## 9.1. Caution

The following safety instruction must be observed during all phases of operation, service and repair of this equipment. Failure to comply with the safety precautions or warnings in this documentation violates safety standards of design, manufacture and intended use of this equipment and may impair the built-in protections within. WATT & WELL shall not be liable for users to comply with these requirements.

## 9.2. Input rating

Do not use power supplies which exceeds the input voltage rating of this instrument. The electrical rating of this instrument is given into the chapter 2 of this document.

## 9.3. Live circuits

Operating personnel are not allowed to open the case of this equipment.
Internal adjustment or component replacement is not allowed by non WATT & WELL qualified personnel. Never replace components with cable connected to this instrument. To avoid injuries, always disconnect power and remove external voltage sources before touching components.

# 10. Installation

Do not use or install product in case of visible physical damage.

## 10.1. Mechanical installation

Refer to chapter 5 for the dimensions of the product.

## 10.2. Electrical installation

Never invert polarity of the connectors. Never force to place a connector. Use only approved manufacturer parts for electrical or mechanical connection.
It is strongly recommended to fix the cables to avoid any stress on connection. All high-power connectors must be screwed to avoid any disconnection.
Be careful if other devices are connected, risk of electrical charge transfer.

Wait two minutes before touching the device after complete suppression of input voltage. Check for lack of voltage, on all access, with the correct equipment.

### 10.3. Disposal

Do not dispose of electronic tools tighter with household waste material. In accordance with WEEE European Directive (2012/19/UE), Electric material that have reach the end of their life must be collected separately and return to an environmentally compatible recycling facility. Please contact WATT & WELL for any questions about WEEE.

## 11. Ordering information

| PART NUMBER | ROTOR POSITION SENSOR |
|---|---|
| **NM-40-600-6-RESO** | Resolver sensor |
| **NM-40-600-6-HALL** | Hall Effect sensor |
| **NM-40-600-6-SLES** | Sensorless |

-**MBD** version includes a customizable firmware as a Simulink Model.

contact@wattandwell.com

Engineering Center: 129 avenue de Paris - Massy (91300) France
Production Facilities: 121 rue Louis Lumière - Pertuis (84120) France
USA Subsidiary: One Riverway, Suite 1700, 770 South Post Oak Lane - Houston TX (77056) USA

# Annexe A - Nar Motion CAN protocol

## 1. Message Identifier format

Each CAN 2.0B frame has a 29 bit identifier and up to 64 bits of data. Nar Motion CAN protocol uses this message ID for node addressing and message description (opcodes):

| **Bit** | **28-27** | **26-19** | **18-11** | **10-9** | **8 -0** |
|---|---|---|---|---|---|
| **Field** | Priority Bits (2bits) | SRC Address (8bits) | DST Address (8bits) | MSG Type (2bits) | Opcode (9bits) |

- Priority bits: priority level of the message. Priority bits can be used to optimize message delivery but should be ignored by each node when decoding the message.
  - 00 : highest priority
  - 01: high priority
  - 10: low priority
  - 11 : lowest priority
- SRC Address: source address.
- DST Address: destination address.
- MSG type: 3 types of message are defined :
  - Type 0 or « Slave Tx »: Message sent from slave to master
  - Type 1 or « Data Set »: Message sent from master to the slave to set a value on the slave memory.
  - Type 2 or « Data Request » Message sent from master to slave to request data from slave. Each data request will trigger Type 0 (Slave Tx) from the slave.
- OpCode: Identifier of message used to describe signification of message.

By default, the setting is the following :

| | **Address** | **Priority** |
|---|---|---|
| **Master (Monitor)** | 0xAA | 1 |
| **Slave (Nar Motion)** | 0x55 | 2 |

**Table 8: Default CAN settings**

## 2. CAN messages dictionnary

With the 29 bits CAN ID defined above, each node can build a message dictionary using the 9-bits Opcode.

| OpCode Name | OpCode value | Msg type | Format | Word 0 | Word 1 | Word 2 | Word 3 | Comment |
|---|---|---|---|---|---|---|---|---|
| **BUILD_INFO** | **0** | **0** | Uint16/ Uin32 / Uint16 | BuildNb | SrcModified | BuildDate | | Answer |
| **BUILD_INFO** | **0** | **2** | Uint16 | (0x0000) | (0x0000) | (0x0000) | (0x0000) | Request |
| **DEVICE_INFO** | **1** | **0** | Uint16 | HwRelease | SwRelease | DeviceNumber | CanAddress | Answer |
| **DEVICE_INFO** | **1** | **2** | Uint16 | (0x0000) | (0x0000) | (0x0000) | (0x0000) | Request |
| **REQUEST_WORD** | **2** | **1** | Bit-Wise | RequestWord | | (0x0000) | (0x0000) | State request |
| **STATUS_WORD** | **3** | **0** | Bit-Wise | StatusWord | | (0x0000) | (0x0000) | State feedback - Sent periodically by the slave |
| **FAULT_WORD** | **4** | **0** | Bit-Wise | FaultWord | | (0x0000) | (0x0000) | Sent periodically by the slave |
| **CRITICAL_FAULT** | **5** | **0** | Bit-Wise | Critical FaultWord | | ExtendedFaultCode | (0x0000) | Sent asynchronously when system is shut down (on rising edge of a critical fault) |
| **EMERGENCY_SHUTDOWN** | **100** | **0** | Uint16 | Ack(0x0001) | (0x0000) | (0x0000) | (0x0000) | |
| **REBOOT** | **101** | **0** | Uint16 | Ack(0x0001) | (0x0000) | (0x0000) | (0x0000) | |
| **MOTOR_CMD** | **102** | **0** | Q3.12 | SpeedCmd | TorqueCmd | PowerCmd | (0x0000) | Setpoint feedback |
| **MOTOR_CMD** | **102** | **1** | Q3.12 | SpeedCmd | TorqueCmd | PowerCmd | (0x0000) | Setpoint write |
| **DIAG_CPU_LOAD_FAST_ISR** | **114** | **0** | Q3.12 | MaxCpuLoadFastIsr | (0x0000) | (0x0000) | PeriodFastIsr | Sent periodically by the slave |
| **DIAG_CPU_LOAD_SLOW_ISR** | **115** | **0** | Q3.12 | MaxCpuLoadSlowIsr | (0x0000) | (0x0000) | PeriodSlowIsr | Sent periodically by the slave |
| **RESET_CPU_DIAG** | **116** | **1** | Q3.12 | Reset CPU diag Req (0x0000) | (0x0000) | (0x0000) | (0x0000) | Word 0 must be 0x0000 to unlock reset |
| **MEASUREMENT_0** | **200** | **0** | Q3.12 | CurrentPhaseA | CurrentPhaseB | CurrentPhaseC | Pressure | Sent periodically by the slave |
| **MEASUREMENT_1** | **201** | **0** | Q3.12 | SpeedMeas | TorqueMeas | PowerMecha | PowerElec | Sent periodically by the slave |
| **MEASUREMENT_2** | **202** | **0** | Q3.12 | HV_Voltage | HV_Current | LV_Voltage | LV_Current | Sent periodically by the slave |
| **MEASUREMENT_3** | **203** | **0** | Q3.12 | Vref 5V LOC | (0x0000) | (0x0000) | (0x0000) | Sent periodically by the slave |
| **MEASUREMENT_4** | **204** | **0** | Q3.12 | Temp_Ambient | Temp_Motor | (0x0000) | (0x0000) | Sent periodically by the slave |
| **MEASUREMENT_5** | **205** | **0** | Q3.12 | Vref 15V LEG | Vref 3.3V | Vref 1.8V | (0x0000) | Sent periodically by the slave |
| **MEASUREMENT_7** | **207** | **0** | Q3.12 | IqRef | IqFdb | IdRef | IdFdb | Sent periodically by the slave |
| **MEASUREMENT_8** | **208** | **0** | Q3.12 | Position | (0x0000) | (0x0000) | (0x0000) | Sent periodically by the slave |

**CAN data description** details data contained in those message frames and, where applicable, gives the normalization value (for details on how to convert this fixed point values to physical unit see section Conversion factor and units).

| Frame | Data | Format | Normalization | Description |
|---|---|---|---|---|
| **BUILD_INFO** | BuildNb | Uint16 | - | Version control number |
| **BUILD_INFO** | SrcModified | Uint16 | - | Version control check result (0-> OK, 1-> fail) |
| **BUILD_INFO** | BuildDate | Uint32 | - | Build date (second since 1900) |
| **BUILD_INFO** | SwRelease | Uint16 | - | Software Release version |
| **BUILD_INFO** | HwRelease | Uint16 | - | Hardware Release version |
| **BUILD_INFO** | DeviceNumber | Uint16 | - | Number of the device |
| **BUILD_INFO** | CanAddress | Uint16 | - | CAN address of device |
| **REQUEST_WORD** | RequestWord | Uint32 bit wise | - | System Control Request Word |
| **STATUS_WORD** | StatusWord | Uint32 bit wise | - | System Control Status Word |
| **FAULT_WORD** | FaultWord | Uint32 bit wise | - | Fault word |
| **CRITICAL_FAULT** | CriticalFaultWord | Uint32 bit wise | - | Critical fault word |
| **MOTOR_CMD** | SpeedCmd | Q3.12 | SPEED_TORQUE_RPMM | Speed setpoint |
| **MOTOR_CMD** | TorqueCmd | Q3.12 | TORQUE_BASE_NMM | Torque setpoint |
| **MOTOR_CMD** | PowerCmd | Q3.12 | PMECA_BASE_W | Power setpoint |
| **DIAG_CPU_LOAD_FAST_ISR** | MaxCpuLoadFastIsr | Q3.12 | PERCENT_BASE | CPU load of fast task in % |
| **DIAG_CPU_LOAD_FAST_ISR** | PeriodFastIsr | Q3.12 | FAST_PERIOD_BASE_US | Fast task period |
| **DIAG_CPU_LOAD_SLOW_ISR** | MaxCpuLoadSlowIsr | Q3.12 | PERCENT_BASE | CPU load of slow task in % |
| **DIAG_CPU_LOAD_SLOW_ISR** | PeriodSlowIsr | Q3.12 | MEDIUM_PERIOD_BASE_MS | Fast task period |
| **MEASUREMENT_0** | CurrentPhaseA | Q3.12 | HC_CURRENT_BASE_A | Phase A current |
| **MEASUREMENT_0** | CurrentPhaseB | Q3.12 | HC_CURRENT_BASE_A | Phase B current |
| **MEASUREMENT_0** | CurrentPhaseC | Q3.12 | HC_CURRENT_BASE_A | Phase C current |
| **MEASUREMENT_0** | Pressure | Q3.12 | PRESSURE_BASE_PSI | Pressure |
| **MEASUREMENT_1** | SpeedMeas | Q3.12 | SPEED_TORQUE_RPMM | Shaft speed |
| **MEASUREMENT_1** | TorqueMeas | Q3.12 | TORQUE_BASE_NMM | Shaft torque |
| **MEASUREMENT_1** | PowerMecha | Q3.12 | PMECHA_BASE_W | Motor mechanical power |
| **MEASUREMENT_1** | PowerElec | Q3.12 | PELEC_BASE_W | Inverter electrical power |
| **MEASUREMENT_2** | HV_Voltage | Q3.12 | HV_VOLTAGE_BASE_V | Inverter input HV voltage |
| **MEASUREMENT_2** | HV_Current | Q3.12 | HC_CURRENT_BASE_A | Inverter input HV current |
| **MEASUREMENT_2** | LV_Voltage | Q3.12 | LV_VOLTAGE_BASE_V | Low voltage power supply current |
| **MEASUREMENT_2** | LV_Current | Q3.12 | LC_CURRENT_BASE_A | Low voltage power supply voltage |
| **MEASUREMENT_3** | Vref 5V LOC | Q3.12 | LV_VOLTAGE_BASE_V | Local 5V bus voltage |

| | | | | |
|---|---|---|---|---|
| **MEASUREMENT_4** | Temp_Ambient | Q3.12 | TEMPERATURE_BASE_DEGCEL | Inverter board temperature |
| **MEASUREMENT_4** | Temp_Motor | Q3.12 | TEMPERATURE_BASE_DEGCEL | Motor temperature |
| **MEASUREMENT_5** | Vref 15V LEG | Q3.12 | LV_VOLTAGE_BASE_V | Local 15V bus voltage |
| **MEASUREMENT_5** | Vref 3.3V | Q3.12 | LV_VOLTAGE_BASE_V | Local 3.3V bus voltage |
| **MEASUREMENT_5** | Vref 1.8V | Q3.12 | LV_VOLTAGE_BASE_V | Local 1.8V bus voltage |
| **MEASUREMENT_7** | IqRef | Q3.12 | HC_CURRENT_BASE_A | Quadrature current reference |
| **MEASUREMENT_7** | IqFdb | Q3.12 | HC_CURRENT_BASE_A | Quadrature current feedback |
| **MEASUREMENT_7** | IdRef | Q3.12 | HC_CURRENT_BASE_A | Direct current reference |
| **MEASUREMENT_7** | IdFdb | Q3.12 | HC_CURRENT_BASE_A | Direct current feedback |
| **MEASUREMENT_8** | Position | Q3.12 | ANGLE_BASE_DEG | Electrical position |

**Table 9: CAN data description**

## 2.1. Request word, status word and fault word messages

Three words are defined bit wise:

- *Request Word*: used to configure the state of the system. Table 10: Request Word/Status Word Bit definition details these words.
- *Status word*: acknowledges that status in the system. It has a similar definition to the request word but is read only. This word is sent by controller periodically.
- *Fault Word*: word containing flag errors. This word is sent by controller synchronously. Table 11: Fault word bit wise definition details these words.

| Bit | Request Word | Status Word | Description |
|---|---|---|---|
| **0:3** | System Mode | | Set/acknowledge the system state<br>0 - System Off : PWM off (power switches in high impedance)<br>1 - System is initializing leg supply<br>2 - Driver are On ( Pwm are OFF)<br>3 - Reserved<br>4 - Drive : positive speed<br>5 - Reverse: negative speed<br>6 - Reserved<br>7 – Fault Mode: PWM off because of a critical fault. System status should be set to system off to clear the fault<br>8 - Generator (rectifier mode) |
| **4** | Reserved | | |
| **5** | Reserved | | |
| **6** | Reserved | SpeedLimitation | Motor operates at speed limit |
| **7** | Reserved | TorqueLimitation | Motor operates at torque limit |
| **8** | Reserved | PowerLimitation | Motor operates at power limit |
| **9-32** | Reserved | | |

**Table 10: Request Word/Status Word Bit definition**

| Bit | Description | Default value |
|---|---|---|
| **0** | Over Voltage Fault on Bus Lv Voltage meas. | 28 |
| **1** | Under Voltage Fault on Bus Lv Voltage meas. | 20 |
| **2** | Warning Temp Fault on Temperature Ambient meas. | 150 |
| **3** | ShutdownTemp Fault on Temperature Ambient meas. | 175 |
| **4** | OverVoltage Fault on HV Voltage meas. | 500 |
| **5** | UnderVoltage Fault on HV Voltage meas. | 200 |
| **6** | OverCurrent Fault on HV Current meas. | 2 |
| **7** | Over current on HV Current (HW detection)" | High level – fast trip |
| **8** | OverCurrent Fault on PhaseA Current meas. | 10 |
| **9** | OverCurrent Fault on PhaseB Current meas. | 10 |
| **10** | OverCurrent Fault on PhaseC Current meas. | 10 |
| **11** | OverCurrent Fault on Chopper Current meas. | N/A |
| **12** | Warning Temp Fault on Temperature Motor meas. | 150 |
| **13** | ShutdownTemp Fault on Temperature Motor meas. | 175 |
| **14** | Error PhaseA | N/A |
| **15** | Error PhaseB | N/A |
| **16** | Error PhaseC | N/A |
| **17** | Error Chopper | N/A |
| **18** | Error HV Voltage | N/A |
| **19** | Error HV Current | N/A |
| **20** | Unused | |
| **21** | Unused | |
| **22** | Unused | |
| **23** | Unused | |
| **24** | Unused | |
| **25** | Unused | |
| **26** | Wrong mapping on External memory used for non volatile parameters | N/A |
| **27** | Emergency Shutdown | N/A |
| **28** | External Shutdown | N/A |
| **29** | Keep Alive Can Message Timeout | N/A |
| **30** | External memory used for non volatile parameters is absent | N/A |
| **31** | SW version on External memory used for non volatile parameters and on program not match | N/A |

**Table 11: Fault word bit wise definition**

## 2.2. Can error code

Not implemented in this version.

## 2.3. Critical Fault frame

When critical fault occurred, a critical fault message is sent asynchronously. It supplies the last fault word to indicate which critical fault puts the system in “Fault Mode”.
This frame include the fault word (state of each fault flag) at the instant of the fault trip.
The extended fault code is not implemented in this version.

# 3. Data normalization

Data words (commands, parameters, measurement) that represent a physical value (voltage, current, time….) are normalized by a "Base Value". These normalized data (also known as *per unit* values) are sent in fixed point format “Q3.12”.

The standard notation QX.Y uses X bits for the integer part (before the point) and Y bit for the decimal part (after the point) and one for the sign. This way, Q3.12 uses 3 bits for the integer part and 12 bits for the decimal part which gives a range between -8 and 7.9997 with a resolution of 2.44e-04.

| # | Name | PU base | Resolution [units] | Min Units | Max Units | Units |
|---|---|---|---|---|---|---|
| **0** | VERY_SLOW_PERIOD_BASE_SEC | 2048 | 0.499992 | -16384 | 16383.5 | sec |
| **1** | SLOW_PERIOD_BASE_MS | 500 | 0.122068 | -4000 | 3999.87793 | ms |
| **2** | MEDIUM_PERIOD_BASE_MS | 10 | 0.002441 | -80 | 79.99755859 | ms |
| **3** | FAST_PERIOD_BASE_US | 20 | 0.004883 | -160 | 159.9951172 | µs |
| **4** | PELEC_BASE_W | 4000 | 0.976548 | -32000 | 31999.02344 | W |
| **5** | HC_CURRENT_BASE_A | 10 | 0.002441 | -80 | 79.99755859 | A |
| **6** | LC_CURRENT_BASE_A | 1 | 0.000244 | -8 | 7.999755859 | A |
| **7** | HV_VOLTAGE_BASE_V | 400 | 0.097655 | -3200 | 3199.902344 | V |
| **8** | LV_VOLTAGE_BASE_V | 24 | 0.005859 | -192 | 191.9941406 | V |
| **9** | TEMPERATURE_BASE_DEGCEL | 150 | 0.036621 | -1200 | 1199.963379 | °C |
| **10** | ANGLE_BASE_RAD | 6.28318531 | 0.001534 | -50.2654825 | 50.26394848 | rad |
| **11** | ANGLE_BASE_DEG | 360 | 0.087889 | -2880 | 2879.912109 | deg |
| **12** | PERCENT_BASE | 100 | 0.024414 | -800 | 799.9755859 | % |
| **13** | ONE_BASE | 1 | 0.000244 | -8 | 7.999755859 | - |
| **14** | HUNDRED_BASE | 100 | 0.024414 | -800 | 799.9755859 | - |
| **15** | RESISTOR_BASE_OHMS | 1000 | 0.244137 | -8000 | 7999.755859 | Ω |
| **16** | FREQUENCY_BASE_HZ | 50 | 0.012207 | -400 | 399.987793 | Hz |
| **17** | PRESSURE_BASE_PSI | 5000 | 1.220684 | -40000 | 39998.7793 | PSI |
| **18** | PMECHA_BASE_W | 4000 | 0.976548 | -32000 | 31999.02344 | Watt |
| **19** | SPEED_BASE_RPMM | 10000 | 2.441369 | -80000 | 79997.55859 | Rpmm |
| **20** | TORQUE_BASE_NMM | 10 | 0.002441 | -80 | 79.99755859 | Nmm |

**Table 12: Base normalization Table**

Example of conversion given a raw value of HV measurements of 5000 read from the CAN bus.

First, it should be converted to a floating point value. To convert 5000 in Q3.12 notation to a floating just divide by 4096 (or perform a 12 bit shift): 5000 / 4096 = 1.22. This indicates that measurement is 22% above its nominal value. To get the measurement in volts, the PU base of 400V must be applied: 1.22*400 = 488V.

## 4. Endianness

Endianness and data format within a CAN frame is application dependent (not part of the CAN specification). Nar Motion protocol uses Big Endian (also known as Network Order)[3]. The resulting byte order for Nar Motion communications is the following:

| Bits | 0 7 | 8 15 | 16 23 | 24 31 | 32 39 | 40 47 | 48 55 | 56 63 |
|---|---|---|---|---|---|---|---|---|
| Bytes | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| | **MSB** | **LSB** | | | | | | |
| 16-bits words: | High word | | Low word | | High word | | Low word | |
| 32-bits words | 32-bit word 0 | | | | 32-bit word 1 | | | |

For example, the 64-bit hexadecimal number « 0x0001 0203 0405 0607 » will be send in CAN bus as follows

| **Bytes** | **16-bits words** | **32-bits words** |
|---|---|---|
| Byte 0: 0x00 | High word: 0x0001 | 32-bit word: 0x00010203 |
| Byte 1: 0x01 | | |
| Byte 2: 0x02 | Low word: 0x0203 | |
| Byte 3: 0x03 | | |
| Byte 4: 0x04 | High word: 0x0405 | 32-bit word: 0x04050607 |
| Byte 5: 0x05 | | |
| Byte 6: 0x06 | Low word: 0x0607 | |
| Byte 7: 0x07 | | |

[3] Computers are usually based on the x86 architecture which is little endian for 16 and 32 bits words. 64-bit numbers (not used on Nar Motion communications) in 32-bits architectures are compiler dependent.

# 5. CAN frames example

## 5.1. Data set message

Here is an example on how to send a MOTOR_CMD frame (see CAN messages dictionary p.27) to NarMotion, in order to set a motor speed of 1000 rpmm, 1 Nmm Torque and 1000 W power :

- Generate the ID corresponding to the command :
  - Define the priority of the frame, in this example : High Priority (0b01)
  - Set the source Address : 0x55 (command is sent by master)
  - Set the destination Address : 0xAA (command is sent to slave)
  - Set the message type : 0b01 (« Data Set » message)
  - Set the Opcode corresponding to the command : 102

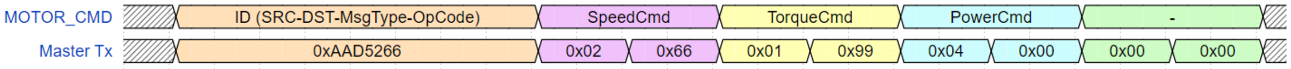
| **Bit** | **28-27** | **26-19** | **18-11** | **10-9** | **8 -0** |
|---|---|---|---|---|---|
| **Field** | Priority<br>0b01 | SRC Address<br>0b0101 0101 (=0x55) | DST Address (8bits)<br>0b1010 1010 (=0xAA) | MSG Type<br>0b01 | Opcode (9bits)<br>0b0 0110 0110 (=102) |

The corresponding ID is : 0b1010 1010 1101 0101 0010 0110 0110 = 0xAAD5266

- Convert the setpoints to fixed-point values, following Table 12 :
  - $\text{PowerCmd} = 4096 * \frac{setpoint}{PU} = 4096 * \frac{1000}{4000} = 1024 = 0x0400$
  - $\text{TorqueCmd} = 4096 * \frac{setpoint}{PU} = 4096 * \frac{1}{10} = 409 = 0x0199$
  - $\text{SpeedlCmd} = 4096 * \frac{setpoint}{PU} = 4096 * \frac{1500}{10000} = 614 = 0x0266$

Therefore, the following CAN frame must be sent :

- ID : 0xAAD5266
- Data : 0x18 0x00 0x01 0x99 0x04 0x00 0x00 0x00



## 5.2. Request/Answer messages

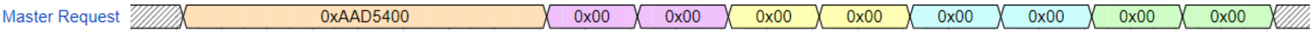
This example shows how to send BUILD_INFO request to NarMotion and how to read its answer.

- Generate the ID corresponding to the command :
  - Define the priority of the frame, in this example : High Priority (0b01)
  - Set the source Address : 0x55 (command is sent by master)
  - Set the destination Address : 0xAA (command is sent to slave)
  - Set the message type : 0b10 (« Data Request » message)
  - Set the Opcode corresponding to the command : 0

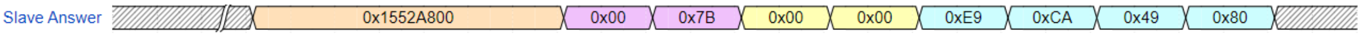
| **Bit** | **28-27** | **26-19** | **18-11** | **10-9** | **8 -0** |
|---|---|---|---|---|---|
| **Field** | Priority<br>0b01 | SRC Address<br>0b0101 0101 (=0x55) | DST Address (8bits)<br>0b1010 1010 (=0xAA) | MSG Type<br>0b10 | Opcode (9bits)<br>0b0 0000 0000 |

The corresponding ID is : 0b0 1010 1010 1101 0101 0100 0000 0000 = 0xAAD5400.
For this request, all data bytes must be 0, therefore, the following CAN frame must be sent :

- ID : 0xAAD5400
- Data : 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00



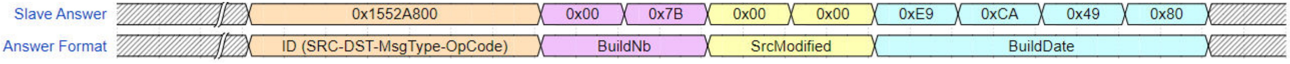
After receiving this message, NarMotion should answer with the following frame :



The identifier (0x1552A800 = 0b10 1010 1010 0101 0101 00 0000000) gives the following informations :

- Low priority message (0b10)
- Source address : slave (0b1010 1010)
- Destination address : master (0b0101 0101)
- MsgType : « Slave Tx » (0b00)
- OpCode : 0

According to the CAN messages dictionary, it corresponds to a BUILD_INFO answer.



The informations given by this frame are :

- BuildNb : 0x007B = 123
- SrcModified : 0x0000 = 0
- BuildDate : 0xE9CA4980 = 3 922 348 416 (seconds since 1900)

The build currently in use is the build N°123, which is a valid version created on 17 April 2024, 13:13:36 UTC.