V1.0.0 - 2025

CAN Bus Motor Control User Manual

This document contains the protocol details required to control the ARM-1412IPPR36 actuator via CAN bus.

1. ID Architecture and Communication

The system uses the **Standard 11-bit CAN ID** structure. The ID is divided into two main parts:

- **Communication Speed:** 1 Mbit/s (1000 kbps)
- **Access Modes:** Indicated in the table as **R** (Read - from device) and **W** (Write - to device).
- **Device ID (Bits 6 - 10):** Specifies the address of the target device (5 bits). The default Device ID is 0x01.
- **Data ID (Bits 0 - 5):** Specifies the identity of the command or data being sent (6 bits).

Example Calculation: CAN ID for Device ID 0x01 and Data ID 0x15 (Target Angle): (0x01 << 6) | 0x15 = 0x40 | 0x15 = 0x55

---

2. CAN Frame Structure (C Example)

On the software side, CAN packets are managed using the following struct structure:

```
typedef struct CanBusPack {
  union {
    uint16_t value = 0;
    struct {
      uint16_t dataID : 6;
```

```
        uint16_t deviceID : 5;
        uint16_t : 5;
      }part;
    }id;
    uint8_t dataLen = 0;
    uint8_t data[8] = {0};
    uint8_t can_RTR = 0 ; // Data or Remote Mode
}_CanBusPack;
```

---

3. Joint ID List

The following table contains all primary IDs processed by the Joint class and provided via telemetry.

ID	Name	Data Type	R/W	Description
0x00	SETTINGSBYBROADCAST	uint32[2]	W	Broadcast setting commands (ID Change, EEPROM Erase).
0x01	BC_CONNECTION	uint8	R/W	Connection control and Emergency Stop.
0x03	UNIT_VERSION_SW	float	R	Software version info.
0x04	UNIT_VERSION_HW	float	R	Hardware version info.
0x05	UNIT_RESET	remote	R/W	Restarts the device / Returns the reset count.
0x06	UNIT_RUNTIME_ERROR	uint32	R	System errors (RTErrorCode) bitmask.
0x08	TELEMETRY_REQUEST	uint8	W	Veri request for a specific Data ID.
0x09	UNIT_DRIVER_VERSION	uint8[2]	R	Driver Maj/Min version info.
0x0A	MCU_TEMP	uint8	R	Processor temperature info (°C).
0x0B	UNIT_ECHO	any	R/W	Returns the sent packet exactly as received (For testing purposes).
0x0C	UNIT_ERROR_CLEAR	remote	W	Clears current errors and starts the system.
0x12	M12_ID_CURRENT	float	R	Current instantaneous current (Iq).
0x13	M12_ID_CURRENT_LIMIT	float	R	Maximum current limit (maxIq).
0x14	M12_ID_REFERENCE_ANGLE	uint8	R/W	Reference (Home) setting operation control.
0x15	M12_ID_TARGET_ANGLE	float	W	Target Angle (**Degrees**). **Note:** Device automatically returns 0x18.
0x16	M12_ID_MIN_ANGLE	float	R	Minimum angle limit (**Degrees**).
0x17	M12_ID_MAX_ANGLE	float	R	Maximum angle limit (**Degrees**).
0x18	M12_ID_POSITION_ANGLE	float	R	Current internal angle (**Degrees**).
0x19	M12_ID_POS_SENS_RES	float	R	Position sensor resolution (PPR).

0x1E	M12_ID_MTR_TEMP	uint8	R	Motor temperature info (°C).

---

4. Advanced Command Descriptions

4.1. SETTINGSBYBROADCAST (0x00) - System Settings

This ID is a broadcast message that reaches all devices. Packet content consists of 2 uint32 (8 bytes total) data words.

- **Changing Node ID:**
  - Data[0-3] (Command): 0x41 (SBB_SET_ID)
  - Data[4-7] (Value): New Node ID to be assigned.
  - *After the operation, the device writes to EEPROM, returns echo, and resets.*
- **EEPROM Erase:**
  - Data[0-3] (Command): 0x42 (SBB_CLEAR_EEPROM)
  - Data[4-7] (Value): 0
  - *After the operation, all parameters are erased, an echo is returned, and the device resets.*

4.2. M12_ID_REFERENCE_ANGLE (0x14) - Reference (Home) Setting

Used to determine the zero point (Home) of the device.

- **Step 1: Setting Temporary Reference (Mode 1)**
  - Send Data[0] = 1.
  - Device accepts current sensor angle as temporary reference and **Resets**.
- **Step 2: Saving Permanent Reference (Mode 2)**
  - After physically moving the motor to the home point, send Data[0] = 2.
  - Device permanently saves the current angle to EEPROM and **Resets**.
- *In response, the device returns the current refSettedState information.*

---

5. Telemetry Request

Data ID 0x08 is used to pull instantaneous data from the device.

- **Request Format:**
  - **CAN ID:** (DeviceID << 6) | 0x08
  - **Data[0]:** The **Data ID** of the requested data (e.g., 0x12 for current).
- **Response Format:**
  - The device returns a packet containing the data related to the ID specified in Data[0].

---

6. Runtime Errors

The device broadcasts a 32-bit error bitmask with Data ID 0x06. Each bit represents a different system error.

Bit	Hex Value	Error Name	Description
0	0x00000001	ERR_EEPROM_RW	EEPROM read/write/communication error.
1	0x00000002	ERR_MCU_INTTEMP	Microcontroller internal temperature error.
2	0x00000004	ERR_MAIN_LOOP	Main software loop error.
3	0x00000008	ERR_MOTOR_CURRENT	Motor current error.
4	0x00000010	ERR_MOTOR_MOVE	Motor movement error.
5	0x00000020	ERR_MOTOR_HS_SOCKET	Motor hall sensor socket error.
6	0x00000040	ERR_MOTOR_HS_STATE	Motor hall sensor state error.
7	0x00000080	ERR_MOTOR_NONTYPE	Motor type error.
8	0x00000100	ERR_MOTOR_POS_CHANGED	Motor fast side (AS5304) position sensor value is not changing or lagging behind the target angle error.
9	0x00000200	ERR_MOTOR_OUTER_POS_DIF	Slow side (AS5047P) sensor large difference between two readings.
10	0x00000400	ERR_MOTOR_INNER_POS_DIF	Fast side (AS5304) sensor large difference between two readings.
11	0x00000800	ERR_MOTOR_POS_LIMIT	Motor position is outside the limit angle.
12	0x00001000	ERR_MOTOR_REF_SETTED	Motor reference mode notification. Motor continues to operate. However, motor considers the point where it was first turned on as the reference.

13	0x00002000	Reserved	
14	0x00004000	ERR_CANBUS_TIMEOUT	CAN bus message transmission timeout exceeded.
15	0x00008000	ERR_MOTOR_ENC_PARITY	Slow side (AS5047P) parity bit error.
16	0x00010000	ERR_MOTOR_ENC_INVCOMM	Slow side (AS5047P) invalid command error.
17	0x00020000	ERR_MOTOR_ENC_FRERR	Slow side (AS5047P) frame error.
18	0x00040000	ERR_MOTOR_ENC_COF	Slow side (AS5047P) CORDIC overflow error.
19	0x00080000	ERR_MOTOR_ENC_MAGL	Slow side (AS5047P) magnet too far error.
20	0x00100000	ERR_MOTOR_ENC_MAGH	Slow side (AS5047P) magnet too close error.
21	`0x00200000"	ERR_MOTOR_ENC_LF	Slow side (AS5047P) offset adjustment error.
22	`0x00400000"	ERR_VBUS	Motor supply voltage error.
23	`0x00800000"	ERR_GD_INIT	Motor driver communication error.
24	`0x01000000"	ERR_GD_FAULT	Motor driver is in "fault" state.
25	`0x02000000"	ERR_SINE_LOOKUP	Motor signal sine table error.
26	`0x04000000"	ERR_MOTOR_TEMP	Motor temperature error.
27	`0x08000000"	ERR_MOTOR_CS_CALIB	Motor phase currents calibration error.
28	`0x10000000"	ERR_MOTOR_ENC_MAGH_AO	Fast side (AS5304) magnet too close error.
29	`0x20000000"	ERR_MOTOR_ENC_MAGL_AO	Fast side (AS5304) magnet too far error.
30	`0x40000000"	ERR_ENC_SPI	Slow side (AS5047P) communication error.
31	`0x80000000"	ERR_MOTOR_ENC_PARITY_R	Slow side (AS5047P) parity bit read error.

---

7. Data Types and Units

- **Unit:** All angle values (Target/Position/Limits) are in **Degrees**.
- **float:** 4-byte data in IEEE 754 standard (Little-endian).
- **uint32:** 4-byte integer.
- **remote:** Data-less RTR packet or empty data packet.

---

8. Example Scenario

Sending a 45.0 degree angle to the motor with Device ID 1:

1. **CAN ID:** (1 << 6) | 0x15 = 0x55
2. **Data (45.0 float):** 00 00 34 42
3. **Sent Packet:** ID: 0x55, DLC: 4, Data: 00 00 34 42
4. **Automatic Response:** Device returns an ID: 0x18, DLC: 4, Data: [Float Current Angle] packet.
5. **Velocity and Acceleration Control:** There is no direct velocity or acceleration command on the device. For smooth movement, velocity and acceleration profiles are provided by sending timed (**trajectory tracking**) **Position Control** messages (0x15) from the main controller.
