Playing with Fusion

BDC-10001 CIM Motor/Controller – CAN User’s Guide

## Introduction

The BDC-10001 combines a highly efficient motor controller, speed sensor, and a CIM motor into a single, convenient package. It may be controller through a servo style (1 to 2ms pulse) PWM interface or through a 1M bit CAN (Controller Area Network) interface.

All messages used extended (29-bit) CAN identifiers.

Multi-byte data fields are little endian (least significant byte first). The number 0x1234 in hexadecimal is encoded in two bytes. The first byte is 0x34 and the second byte is 0x12. All signed numbers are represented in two’s complement.

Many messages include a ‘Device ID’ as the least significant byte of the CAN message identifier. This device ID is programmed into each BDC-10001 by the end user and is used to identify individual devices on a CAN bus. All sensors ship with a default device ID of zero.

## Glossary

|  |  |
| --- | --- |
| CAN | Controller Area Network |
| DLC | Data Length Code; The number of data bytes present in CAN message (0-8) |

## Control Modes

### Proportional (Coast & Braking)

### Speed Control

|  |  |  |
| --- | --- | --- |
| Constant | CAN Value | Real World value |
| Kf | 47 | 0.184 |
| B | 0 | 0 |
| Kp | 200 | 0.195 |
| Ki | 10 | 0.010 |
| Max V | 5500 | 5500 RPM |
| Max A | 250 | 20,000 RPM/sec |
| Max J | 50 | 31,250 RPM/sec2 |

### Torque Control

|  |  |  |
| --- | --- | --- |
| Constant | CAN Value | Real World value |
| Kf | 0 | 0 |
| B | 0 | 0 |
| Kp | 200 | 0.195 |
| Ki | 10 | 0.010 |
| Max V | N/A | N/A |
| Max A | N/A | N/A |
| Max J | N/A | N/A |

### Position Control

|  |  |  |
| --- | --- | --- |
| Constant | CAN Value | Real World value |
| Kf | 47 | 0.184 |
| B | 0 | 0 |
| Kp | 1536 | 1.5 |
| Ki | 0 | 0 |
| Max V | 5500 | 5500 RPM |
| Max A | 200 | 20,000 RPM/sec |
| Max J | N/A | N/A |

### Motion Profile

## Transmitted messages

### Heartbeat

Heartbeat message is transmitted once a second and provides the sensor firmware version and hardware serial number

|  |  |
| --- | --- |
| **Message ID:** | 0x020508ss (ss = device ID) |
| **DLC:** | 8 |
| **Byte 0:** | Reserved, always 0 |
| **Bytes 1-3:** | Hardware serial number |
| **Bytes 4-5:** | Manufacturer part number, always 0x110 |
| **Bytes 6-7:** | Firmware version |

### Control State

TODO

|  |  |
| --- | --- |
| **Message ID:** | 0x020504ss (ss = device ID) |
| **DLC:** | 8 |
| **Bytes 0-1:** | Signed (2’s complement) motor duty cycle in percent. To convert to engineering units, multiply the CAN value by (100%/1023). The sign of the duty cycle defines which direction torque will be applied. Positive numbers will cause an unloaded motor to rotate clock-wise when looking at the motor shaft. |
| **Bytes 2-3:** | Signed sum of the proportional and integral terms of the internal PI controller. To convert to engineering units, multiply the CAN value by (100%/1023). |
| **Bytes 4-5:** | Signed input to the internal PI controller. The units and scaling of this field vary with the current control mode and are the same as the Command signal in the Motor Command CAN message. |
| **Bytes 6-7:** | Absolute value of the motor electrical current in milliamps. |

### Speed and Position

TODO

|  |  |
| --- | --- |
| **Message ID:** | 0x020506ss (ss = device ID) |
| **DLC:** | 8 |
| **Bytes 0-3:** | Signed motor angle in degrees |
| **Bytes 4-5:** | Signed motor velocity in RPM |

### Voltage and Health

TODO

|  |  |
| --- | --- |
| **Message ID:** | 0x020505ss (ss = device ID) |
| **DLC:** | 8 |
| **Bytes 0-1:** | Analog voltage at auxiliary input on encoder PCB in millivolts |
| **Bytes 2-3:** | Supply voltage in millivolts |
| **Byte 4:** | Motor absolute position encoder state |
|  | **Bits 0-6:** Encoder gain. Should be between 20 and 44.  **Bit 7:** Encoder locked and able to measure motor position |
| **Bytes 6-7:** | Motor back plate temperature in degrees Celsius time 10. To convert to engineering units, multiply the CAN value by 0.10. |

### Motion Profile State

Only transmitted while executing a motion profile

|  |  |
| --- | --- |
| **Message ID:** | 0x020507ss (ss = device ID) |
| **DLC:** | 8 |
| **Bytes 0-2:** | Signed motor angle command in degrees |
| **Bytes 3-4:** | Signed motor speed command in RPM |
| **Bytes 5-6:** | Current motion profile point. Only 12 bits wide. 4 least significant bits of byte 6 are used |
| **Bytes 6-7:** | Number of free motion profile points available. Only 12 bits wide. 4 most significant bits of byte 6 are the least significant bits of the final value. |

## Received Messages

### Motor Command

TODO

|  |  |  |
| --- | --- | --- |
| **Message ID:** | 0x020501ss (ss = device ID) | |
| **DLC:** | 8 | |
| **Byte 0:** | Bits 0-1: Heartbeat. This field must change at least once every 200 milliseconds.  Bit 2: Reset motor position. Set to 1 to reset reported motor angle to 0.  Bits 4-7: Commanded mode | |
|  | Value | Description |
| 0 | Motor off |
| 1 | Proportional (Coast). Command signal specifies signed motor duty cycle as a value between -1023 and 1023. A command of 0 causes the motor to free wheel |
| 2 | Proportional (Brake). Command signal specifies signed motor duty cycle as a value between -1023 and 1023. A command of 0 causes the motor to brake |
|  | 3 | Torque Control (Current control). Command signal specifies the signed target motor current in milliamps. |
|  | 4 | Speed Control. Command signal specifies signed motor speed target in RPM. |
|  | 5 | Position Control. Command signal specifies the signed target motor angle (45 degrees per bit) |
|  | 6 | Execute Motion Profile. Command signal is ignored |
|  | 7 | Follow the Leader. Command specifies the motor ID of the lead motor (the motor to follow) |
|  | 8 | Voltage Control (Voltage compensated proportional mode). Command signal specifies the signed motor voltage command in millivolts. |
|  |  |  |
| **Bytes 1-2:** | Motor Command, see byte 0 for a description of how Command is used. | |
| **Bytes 3-4:** | Feed forward gain (kf). Ratio of 1/256 per bit. Multiply real world value by 1/256 to calculate CAN value. 12-bit value. The 4 least significant bits of byte 4 are used | |
| **Byte 4-5:** | Feed forward offset (b). Ratio of 100%/1024 per bit. Multiply real world value by 100%/1024 to calculate CAN value. 12-bit value. The 4 most significant bits of byte 4 are used | |
| **Bytes 6-7:** | Maximum commanded motor velocity (unsigned or absolute value) when in Speed or Position control Modes. Unused in other control modes | |

### PI Gain Configuration

TODO

|  |  |
| --- | --- |
| **Message ID:** | 0x020502ss (ss = device ID) |
| **DLC:** | 8 |
| **Byte 0:** | Maximum motor acceleration (unsigned). 100 RPM per second per bit. Valid in Position and Speed control modes |
| **Byte 1:** | Maximum motor jerk (unsigned). 625 RPM per second squared per bit. Only valid in Speed control mode |
| **Bytes 2-3:** | Proportional gain (kp). 100%/1024 per bit. Only 12 bits wide. 4 least significant bits of byte 3 are used |
| **Bytes 3-4:** | Integral gain (ki) 100%/1024 per bit. Only 12 bits wide. 4 most significant bits of byte 3 are the least significant bits of the final value. |
| **Bytes 5-6:** | Signed minimum PI controller output value. 100%/1024 per bit. Only 12 bits wide. 4 least significant bits of byte 6 are used |
| **Bytes 6-7:** | Signed maximum PI controller output value. Only 12 bits wide. 4 most significant bits of byte 6 are the least significant bits of the final value. |

### Motion Profile Point

TODO

|  |  |  |
| --- | --- | --- |
| **Message ID:** | 0x020503ss (ss = device ID) | |
| **DLC:** | 8 | |
| **Bytes 0-1:** | Delta time since last point in milliseconds | |
| **Bytes 2-3:** | Commanded velocity in RPM | |
| **Bytes 4-6:** | Commanded motor angle in degrees (360 degrees per revolution) | |
| **Byte 7:** | Point type | |
|  | Value | Description |
| 0 | First point in a new motion profile |
|  | 1 | Append point to end of motion profile point list |
|  | 2 | Final point in motion profile. Motor will break and hold position upon reaching this point. |

### Device Configuration

The Device Configuration is used to set the device id (least significant byte of CAN ID). This message is unique in that the message ID is fixed. Since the device ID is not part of the CAN ID, the particular device is specified by the serial number and manufacturer part number (available from the Heartbeat message) in bytes 1-5.

|  |  |
| --- | --- |
| **Message ID:** | 0x1F0503FF |
| **DLC:** | 7 |
| **Byte 0:** | Command byte, must be 0x0C |
| **Bytes 1-3:** | Hardware serial number |
| **Bytes 4-5:** | Manufacturer part number, always 291 |
| **Byte 6:** | New device ID |

### Device Identification

The Device Identification messages flashes the sensor LED for five seconds to identify a physical sensor. Since the device ID is not part of the CAN ID, the particular device is specified by the serial number and manufacturer part number (available from the Heartbeat message) in bytes 1-5.

|  |  |
| --- | --- |
| **Message ID:** | 0x1F0503FF |
| **DLC:** | 6 |
| **Byte 0:** | Command byte, must be 0x0D |
| **Bytes 1-3:** | Hardware serial number |
| **Bytes 4-5:** | Manufacturer part number, always 291 |