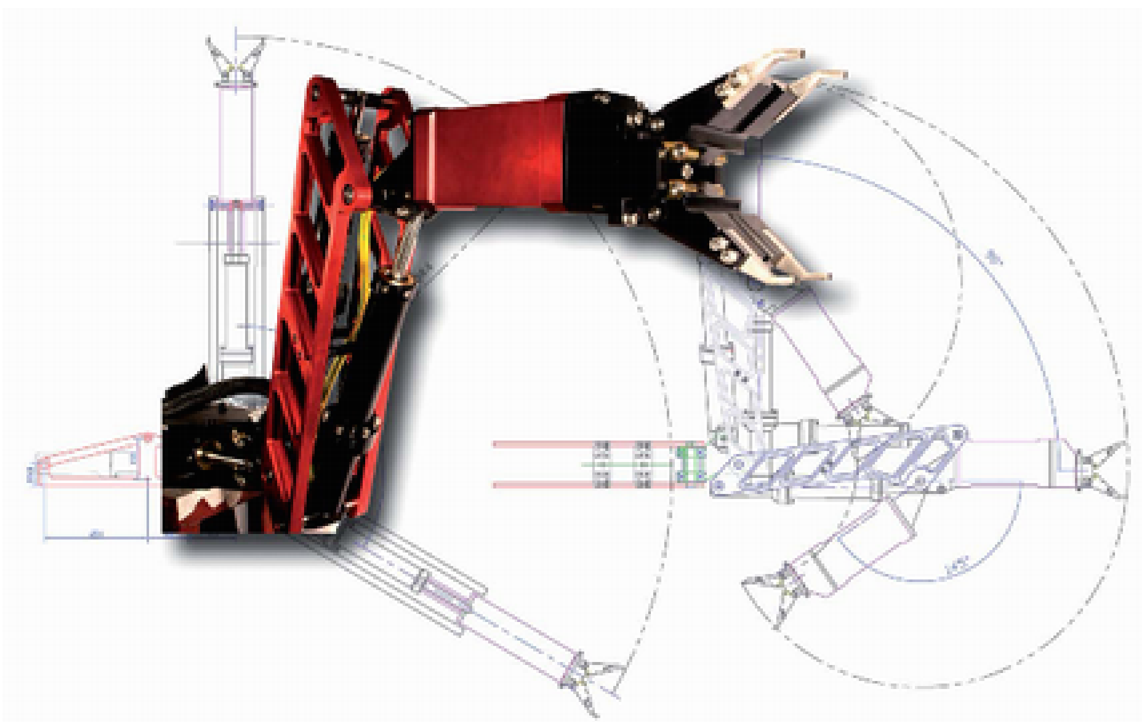
# CSIP 5E MK2 CONTROL SYSTEM

**Description: Communications and Advanced Details**

**Document Revision 1.0 Beta**

**Preliminary: For Internal Use Only**



**Author: David Scott Frame**

**Date: 17th May 17, 2010**

# 1 Abbreviations/definitions

CHK: Checksum
EOM: End of message
SOM: Start of message
COMS: Communications
CAN: Controller Area Network
ECAN: Enhanced CAN (See microchip manual dspic33f ECAN)
PWM: pulse width modulation
PID: Proportional integral derivative
P: Proportional (denoted with Position or Speed)
I: integral (denoted with Position or Speed)
D: derivative (denoted with Position or Speed)
Brownout: drop in users power supply voltage due to overload
USB: universal serial bus
LSB: Least significant bit
MSB: Most significant bit
5E: 5 Function electronic Arm
MK2: CSIP Digital control system that supersedes MK1 which has no position feedback.
Serial Port: Refers to a Serial Port or 485 port

# 2 Table of contents

# 3 Communication details

The communications are via Serial Port and CAN bus. Serial Port is provided to connect to a PC, CAN bus is responsible for communications between the motor drivers and Serial Port link *(Figure 1: Communications paths*). This document will focus solely at Serial Port level.

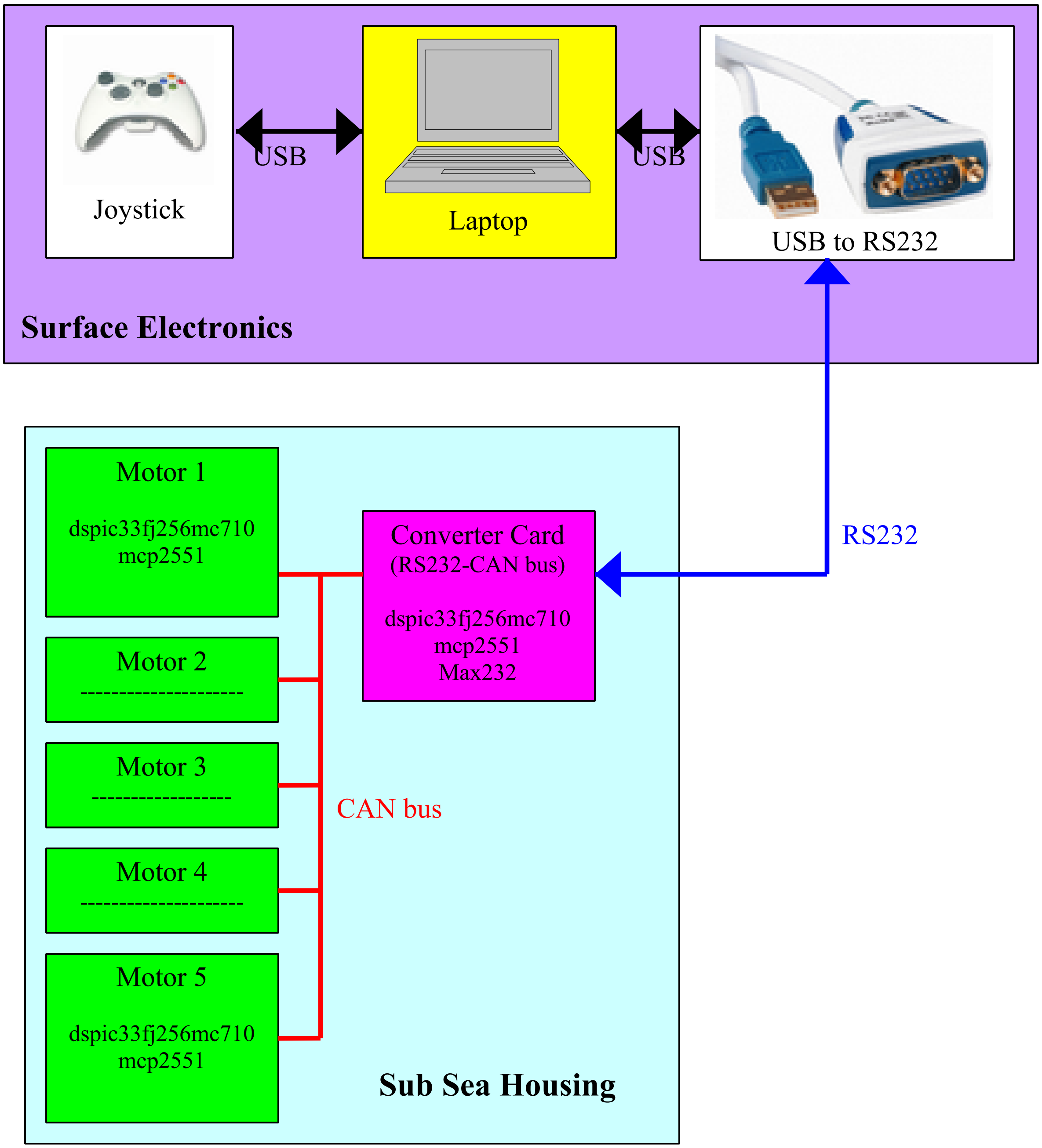



**Figure 1: Communications paths**

### 3.1 Data Types

Bellow *(Figure 2: Data Variables)* is a full list of all data variables available along with the direction they are sent. Thought-out this document explanation will be given on how to align these variables into packages and how to convert them.

| **Name** | **Direction** | **Quantity** | **size** | **Range** |
|---|---|---|---|---|
| Demand Type | To arm motor | 1 per motor | 8 bits | 0 = stop<br><br>1 = Voltage demand clockwise<br><br>2= Voltage demand Anti clockwise<br><br>3 = Speed demand clockwise<br><br>4= Speed demand Anti clockwise<br><br>5=Position |
| Demand | To arm motor | 1 per motor | Up too 16 bits | Voltage Demand Type = 0- 65535<br><br>Speed Demand Type = 0- 4095<br><br>Position Demand Type = 0- 65535 |
| Speed Limit | To arm motor | 1 per motor | 12 bits | 0- 4095 |
| Current Limit | To arm motor | 1 per motor | 12 bits | 0- 4095 |
| Position | To PC | 1 per motor | 16 bits | 0- 65535 |
| Speed | To PC | 1 per motor | 12 bits | 0- 4095 |
| Current | To PC | 1 per motor | 12 bits | 0- 4095 |
| Temp | To PC | 1 per motor | 8 bits | 0-255 |
| P Position | To arm motor | 1 per motor | 8 bits | 0-255 |
| I Position | To arm motor | 1 per motor | 8 bits | 0-255 |

| | | | | |
|---|---|---|---|---|
| D Position | To arm motor | 1 per motor | 8 bits | 0-255 |
| P Speed | To arm motor | 1 per motor | 8 bits | 0-255 |
| 16 bit alignment<br>I Speed | To arm motor | 1 per motor | 8 bits | 0-255 |
| D Speed | To arm motor | 1 per motor | 8 bits | 0-255 |
| Master Temperature | To PC | 1 only | 8 bits | 0-255 |
| Master Voltage | To PC | 1 only | 8 bits | 0-255 |
| Master Current | To PC | 1 only | 8 bits | 0-255 |

**Figure 2: Data Variables**

## *3.2 Alignment*

This section details the alignment of the 16, 12 and 8 bit data variables.

### 3.2.1 16 bit alignment

This will consist of two bytes

| First transmitted byte | | | | | | | | Second Transmitted byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB | MSB | | | | | | | LSB |
| D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### 3.2.2 12 bit alignment

This will consist of two bytes with the 4 MSB not used as shown bellow.

| First transmitted byte | | | | | | | | Second Transmitted byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB | MSB | | | | | | | LSB |
| Must be zero | Must be zero | Must be zero | Must be zero | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

### 3.2.3 8 bit alignment

This will consist of one byte

| Transmitted byte | | | | | | | |
|---|---|---|---|---|---|---|---|
| MSB | | | | | | | LSB |
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |

## *3.3 Conversions*

This Section shows how to convert the communications values to more meaningful figures.

### 3.3.1 Master and Motor Temperature

This is the 8 bit temperature sensor located on the heat sinks.

The 8 bit temperature values for both the motor and the master can be approximated to °C using the following equation.

Temprature °C = ((8BitTemprature / 255) * 3.3) / 0.0066101694915254237288135593220339

### 3.3.2 Master Voltage

This is the resistor diode network connected to the power supply. Note this value is for guidance only and subject to error under load due to nonlinear effects of diodes in the system. It is solely used to indicate brownout of the users power supply.

The 8 bit Voltage value can be approximated to Volts using the following equation.

Volts V = (8BitVoltage / 255 * 3.3) / (6800 / 111500)

### 3.3.3 Master Current

This is the value from the LEM Hall Effect sensor on the master card.

The 8 bit Current value can be approximated to Amps using the following equation.

Current Amps =(((8BitCurrent / 511 * 3.3) / (39 / 59)) / 0.625) * 6 - 0.2

### 3.3.4 Motor Current

This 12bit DAC value is from the current sense resistor on the motor driver.

## *3.4 Demand Type and Demand*

Demands must be constantly transmitted to the motors. If any motor fails to receive demand within 500ms then the motor will enter the emergency stop condition.

The motor can be demanded to move or stop in multiple ways.

| Demand Type | Description | Demand value |
|---|---|---|
| 0 | Stop Demand | Not Defined |
| 1 | Voltage Demand Clockwise | 16 bit PWM value |
| 2 | Voltage Demand Anticlockwise | 16 bit PWM value |
| 3 | Speed Demand Clockwise | 0-4095 RPM |
| 4 | Speed Demand Anticlockwise | 0-4095 RPM |
| 5 | Position Demand | 16 bit position demand |

### 3.4.1 Stop/hold Demand (Demand Type = 0)

When this demand is found the motor is put in the stop condition, this is for future development. Note that the motors windings are shorted when the motors fall bellow a preset PWM value.

### 3.4.2 Voltage Demand clockwise (Demand Type = 1)

This is a 16 bit value which sets the motors drive PWM voltage between 0-100%, the motor is spun in the clockwise direction.

### 3.4.3 Voltage Demand anticlockwise (Demand Type = 2)

This is a 16 bit value which sets the motors drive PWM voltage between 0-100%, the motor is spun in the anticlockwise direction.

### 3.4.4 Speed Demand clockwise (Demand Type = 3)

This value is 12 bit 0-4095 and sets the motor speed in RPM in the clockwise direction. Note this is a closed loop PID control and can be tuned using the PID message.

### 3.4.5 Speed Demand anticlockwise (Demand Type = 4)

This value is 12 bit 0-4095 and sets the motor speed in RPM in the anticlockwise direction. Note this is a closed loop PID control and can be tuned using the PID message.

### 3.4.6 Position Demand (Demand Type = 5)

This is a 16 bit position which the motor will attempt to drive to via the quickest route.
Note this is a closed loop PID control and can be tuned using the PID message.

**WARNING, DEMANDS OUTSIDE THE STROKE OF THE ACTUATOR WILL PUT HIGH FORCE ON THE ENDSTOPS CAUSING DAMAGE**.

# 4 Serial Port Overview

The master Converter is responsible for communication via SERIAL PORT. Unless otherwise stated the Serial Port is programmed for the following

9600baud
8 Data bits
No parity
1 stop bit

Communications start at the surface where a package is constructed containing control data for each motor *(Figure 3: Communications path)*. This message when accepted by the manipulator will trigger the response containing all known data from the manipulator. A timer is used to refresh the manipulators data. Note that this timer must be faster than the safety stop timer but slower than the time taken during a send receive cycle.

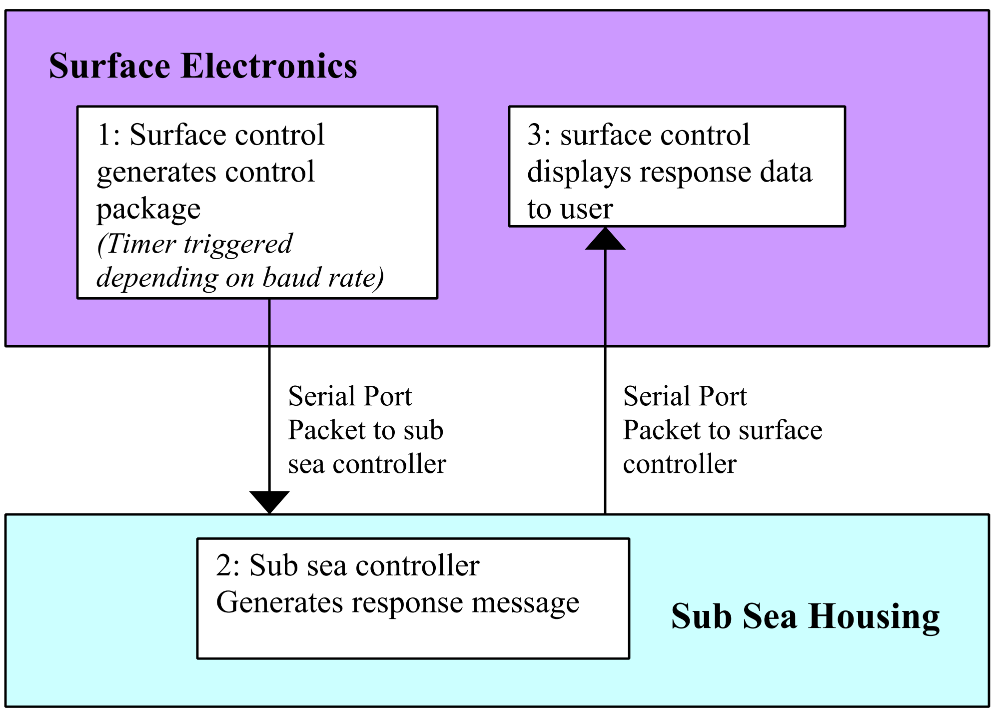



**Figure 3: Communications path**

# 5 Serial Port Package Detail

The Serial Port packages are always 51 bytes long; along with the required data they also contain a basic checksum and start/ending message markers to denote valid package *( Figure 4: Basic structure of Serial Port package)*. The message is always in a standard format due to the way the sub sea converter talks the CAN bus.

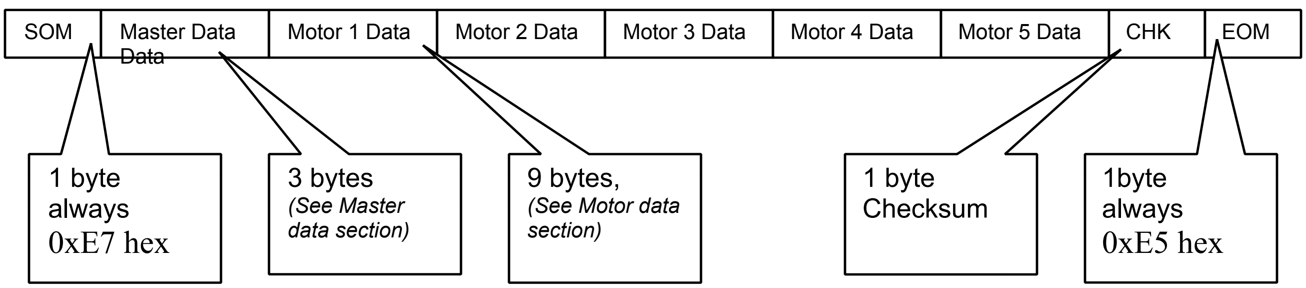



**Figure 4: Basic structure of Serial Port package**

## *5.1 SOM and EOM*

The SOM and EOM are the start of message and end of message bytes. Along with the checksum they allow the controllers to synchronize the message.
SOM = 0xE7 hex
EOM = 0xE5 hex

## *5.2 Checksum (CHK)*

Check sum is the mathematical addition of all bytes in the SOM, Master data and Motor data blocks. Note that it is an 8 bit value and will rollover, for example 0xFF+0xF0 = EF. **See appendix for an example calculation**.

## *5.3 Master data*

The master data block is 3 bytes (*Figure 5: Master Data).* In the demand (PC to Manipulator) these values are left blank and reserved for future use as master limits. In response (Manipulator to PC) these values are the sensor values

| Master Data | | |
|---|---|---|
| First Transmitted Byte 0 | Byte 1 | Last Transmitted Byte 2 |
| Master temperature | Master voltage | Master current |

**Figure 5: Master Data**

*For information on converting byte values to real units see the conversions section of this document*

## *5.4 Motor data*

9 bytes of motor data are inserted into the Serial Port package; there are 3 possible different messages Demand, PID and Sensors *(Figure 6-8)*.

Demand and PID data is sent from the master to the motor drivers. Sensor messages are transmitted to the Master. **Note that the Demand message must be refreshed every 500ms or the motor will stop as a safety feature.**

| Motor Demand message (generated by PC) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| First Transmitted Byte 0 | Byte 1 | Byte 2 | Byte | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Last Transmitted Byte 8 |
| 0x00 hex Constant | Demand type | Demand (alignment depends on demand) | | Speed Limit (see 12 bit alignment) | | Current Limit (see 12 bit alignment) | | Reserved 0x00 Default |

**Figure 6: Demand Message**

| Motor PID message (generated by PC) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| First Transmitted Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Last Transmitted Byte 8 |
| 0x01 hex Constant | P Position | I Position | D Position | P Speed | I Speed | D Speed | Reserved 0x00 hex Default | Reserved 0x00 hex Default |

**Figure 7: PID message**

| Motor Sensors message (generated by manipulator) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| First Transmitted Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Last Transmitted Byte 8 |
| 0x01 hex Constant | Position | | Speed (see 12 bit alignment) | | Current (see 12 bit alignment) | | Temperature | Reserved 0x00 Default |

**Figure 8: Sensors Message**

# 6 Appendix

## *6.1 Example of Serial Port message packet transmitted from PC to manipulator*

This is an example message packet transmitted from the PC to the manipulator. Where

- Motor 1 is sent a voltage demand,
- Motor 2 is sent a Speed demand,
- Motor 3 is sent a position demand,
- Motor 4 and motor 5 are sent PID demands.

Note after this message is sent and accepted the manipulator will respond with all known senor values.

| **Index** | **Example Value (hex)** | **Description** | **Checksum calculation example** |
|---|---|---|---|
| 0<br>(first transmitted) | 0xE7 | Start of message 0xE7 | E7 |
| 1 | 0x00 Reserved for future use | Master Temperature | E7+0= E7 |
| 2 | 0x00 Reserved for future use | Master Voltage | E7+0= E7 |
| 3 | 0x00 Reserved for future use | Master Current | E7+0= E7 |
| 4 | 0x00 | Motor 1 demand message prefix | E7+0= E7 |
| 5 | 0x01 | Motor 1 Voltage demand clockwise | E7+1= E8 |
| 6 | 0xFF | Motor1 Demand MSB<br>100% PWM | E7+FF= E6<br>(note roll over) |
| 7 | 0xFF | Motor 1 Demand LSB 100% PWM | E6+FF= E5 |
| 8 | 0x0F | Motor 1 Speed MSB Limit = 4095 RPM | E5+0F=F4 |
| 9 | 0xFF | Motor 1 Speed LSB Limit = 4095 RPM | F4+FF=F3 |
| 10 | 0x0F | Motor 1 Current MSB Limit = full current | F3+0F= 02 |
| 11 | 0xFF | Motor 1 Current | 02+FF = 01 |

| | | | |
|---|---|---|---|
| | | Limit LSB = full current | |
| 12 | 0x00 Reserved for future use | Motor 1 Data 7 | 01+00= 01 |
| 13 | 0x00 | Motor 2 demand message prefix | 01+00= 01 |
| 14 | 0x03 | Motor 2 Speed demand clockwise | 01 + 03 = 04 |
| 15 | 0x03 | Motor 2 Demand MSB 1000RPM | 04 + 03 = 07 |
| 16 | 0xE8 | Motor 2 Demand LSB 1000RPM | 07 + E8 = EF |
| 17 | 0x0F | Motor 2 Speed MSB Limit = 4095 RPM | EF + 0F = FE |
| 18 | 0xFF | Motor 2 Speed LSB Limit = 4095 RPM | FE + FF = FD |
| 19 | 0x0F | Motor 2 Current MSB Limit = full current | FD + 0F = 0C |
| 20 | 0xFF | Motor 2 Current LSB Limit = full current | 0C + FF = 0B |
| 21 | 0x00 Reserved for future use | Motor2 Data 7 | 0B + 0 = 0B |
| 22 | 0x00 | Motor 3 demand message prefix | 0B + 0 = 0B |
| 23 | 0x05 | Motor 3 Position demand | 0B + 05 = 10 |
| 24 | 0x1F | Motor 3 Demand MSB Pos = 8177 | 10 + 1F = 2F |
| 25 | 0xF1 | Motor 3 Demand LSB Pos = 8177 | 2F + F1 = 20 |
| 26 | 0x0F | Motor 3 Speed MSB Limit = 4095 RPM | 20 + 0F = 2F |
| 27 | 0xFF | Motor 3 Speed LSB Limit = 4095 RPM | 2F + FF = 2E |
| 28 | 0x0F | Motor 3 Current MSB Limit = full current | 2E + 0F = 3D |

| | | | |
|---|---|---|---|
| 29 | 0xFF | Motor 3 Current LSB Limit = full current | 3D + FF = 3C |
| 30 | 0x00 Reserved for future use | Motor 3 Data 7 | 3C + 00 = 3C |
| 31 | 0x01 | Motor 4 PID Message prefix | 3C + 01 = 3D |
| 32 | 0xFF | Motor 4 P Position | 3D + FF = 3C |
| 33 | 0x0F | Motor 4 I Position | 3C + 0F = 4B |
| 34 | 0xF0 | Motor 4 D Position | 4B + F0 = 3B |
| 35 | 0xFF | Motor 4 P Speed | 3B + FF = 3A |
| 36 | 0x01 | Motor 4 I Speed | 3A + 01 = 3B |
| 37 | 0x77 | Motor 4 D Speed | 3B + 77 = B2 |
| 38 | 0x00 Reserved for future use | Motor3 Data 6 | B2 + 00 = B2 |
| 39 | 0x00 Reserved for future use | Motor3 Data 7 | B2 + 00 = B2 |
| 40 | 0x01 | Motor 5 PID Message prefix | B2 + 01 = B3 |
| 41 | 0xFF | Motor 5 P Position | B2 + FF = B1 |
| 42 | 0x0F | Motor 5 I Position | B1 + 0F = C0 |
| 43 | 0xF0 | Motor 5 D Position | C0 + F0 = B0 |
| 44 | 0xFF | Motor 5 P Speed | B0 + FF = AF |
| 45 | 0x01 | Motor 5 I Speed | AF + 01 = B0 |
| 46 | 0x77 | Motor 5 D Speed | B0 + 77 = 27 |
| 47 | 0x00 Reserved for future use | Motor5 Data 6 | 27 + 0 = 27 |
| 48 | 0x00 Reserved for future use | Motor5 Data 7 | 27 + 0 = 27 |
| 49 | 0x27 | Checksum | =27 |
| 50 | 0xE5 | EOM 0xE5 | |

## *6.2 Example of Serial port message transmitted from Manipulator*

Bellow is an example of a message packet transmitted from the manipulator. This message contains all known sensor values. It is sent directly after the manipulator receives a message block from the surface control.

| **Index** | **Example Value (hex)** | **Description** | **Checksum calculation example** |
|---|---|---|---|
| 0<br>(first transmitted) | 0xE7 | Start of message 0xE7 | E7 |
| 1 | 0x14 | Master Temperature<br>= 39.1°C | E7+0x14 = 0xFB |
| 2 | 0x76 | Master Voltage =25V *(see conversions section)* | FB + 0x76 = 0x71 |
| 3 | 0x0D | Master Current =1A<br>*(see conversions section)* | 0x71+0x0D= 0x7E |
| 4 | 0x01 | Motor 1 Sensors prefix | 7E + 0x01 =7F |
| 5 | 0x0F | Motor 1 position MSB<br>=4095 | 7F + 0x0F = 0x8E |
| 6 | 0xFF | Motor1 position LSB<br>=4095 | 8E + 0xFF = 0x8D |
| 7 | 0x03 | Motor 1 speed MSB<br>=1023 RPM | 8D + 0x03 = 0x90 |
| 8 | 0xFF | Motor 1 speed LSB<br>= 1023 RPM | 0x90+0xFF= 0x8F |
| 9 | 0x00 | Motor 1 Current MSB<br>= 0 | 0x8F+0x00=0x8F |
| 10 | 0x00 | Motor 1 Current LSB =0 | 0x8F+0c00=0x8F |
| 11 | 0x14 | Motor 1 temperature<br>= 39.1 °C | 0x8F + 0x14 =0xA3 |
| 12 | 0x00 Reserved for future use | Motor 1 Data 7 | 0xA3 + 0x00 = 0xA3 |

| | | | |
|---|---|---|---|
| 13 | 0x01 | Motor 2 Sensors prefix | 0xA3+0x01=0xA4 |
| )14 | 0x03 | Motor 2 position MSB<br>= 1023 | 0xA4 + 0x03 = 0xA7 |
| 15 | 0xFF | Motor 2 position LSB<br>= 1023 | 0xA7+0xFF=0xA6 |
| 16 | 0x0F | Motor 2 speed MSB<br>=4095 RPM | 0xA6+0x0F= 0xB5 |
| 17 | 0xFF | Motor 2 speed LSB<br>=4095 RPM | 0xB5 + 0xFF =0xB4 |
| 18 | 0x00 | Motor 2 Current MSB<br>=0 | 0xB4 + 0x00= 0xB4 |
| 19 | 0x00 | Motor 2 Current LSB<br>=0 | 0xB4+0x00=0xB4 |
| 20 | 0x14 | Motor 2 temperature<br>= 39.1 °C | 0xB4+0x14=0xC8 |
| 21 | 0x00 Reserved for future use | Motor2 Data 7 | 0xC8+0x000xC8 |
| 22 | 0x01 | Motor 3 Sensors prefix | 0xC8+0x01 = 0xC9 |
| 23 | 0x0F | Motor 3 position MSB<br>=4095 | 0xC9+0x0F = 0xD8 |
| 24 | 0xFF | Motor 3 position LSB<br>=4095 | 0xD8+0xFF=0xD7 |
| 25 | 0x03 | Motor 3 speed MSB<br>=1023 RPM | 0xD7+0x03=0xDA |
| 26 | 0xFF | Motor 3 speed LSB<br>= 1023 RPM | 0xDA+0x0FF=0xD9 |
| 27 | 0x00 | Motor 3 Current MSB<br>= 0 | 0xD9+0x00 = 0xD9 |
| 28 | 0x00 | Motor 3 Current LSB<br>=0 | 0xD9+0x00 = 0xD9 |
| 29 | 0x14 | Motor 3 | 0xD9+0x14= 0xED |

| | | | |
|---|---|---|---|
| | | temperature<br>= 39.1 °C | |
| 30 | 0x00 Reserved for future use | Motor 3 Data 7 | ED+0x00= 0xED |
| 31 | 0x01 | Motor 4 Sensors prefix | 0xED+0x01 = 0xEE |
| 32 | 0x03 | Motor 4 position MSB<br>= 1023 | 0xEE+0x03= 0xF1 |
| 33 | 0xFF | Motor 4 position LSB<br>= 1023 | 0xF1 + 0xFF= 0xF0 |
| 34 | 0x0F | Motor 4 speed MSB<br>=4095 RPM | 0xF0+0x0F=0xFF |
| 35 | 0xFF | Motor 4 speed LSB<br>=4095 RPM | 0xFF+0xFF=0xFE |
| 36 | 0x00 | Motor 4 Current MSB<br>=0 | 0xFE+0x00=0xFE |
| 37 | 0x00 | Motor 4 Current LSB<br>=0 | 0xFE+0x00=0xFE |
| 38 | 0x14 | Motor 4 temperature<br>= 39.1 °C | 0xFE+0x14= 0x12 |
| 39 | 0x00 Reserved for future use | Motor4 Data 7 | 0x12+0x00=0x12 |
| 40 | 0x01 | Motor 5 Sensors prefix | 0x12+0x01 = 0x13 |
| 41 | 0xFF | Motor 5 position MSB<br>=65280 | 0x13+0x0FF=0x12 |
| 42 | 0x00 | Motor 5 position LSB<br>=65280 | 0x12+0x00=0x12 |
| 43 | 0x03 | Motor 5 speed MSB<br>=1023 RPM | 0x12+0x03=0x15 |
| 44 | 0xFF | Motor 5 speed LSB<br>= 1023 RPM | 0x15+0xFF=0x14 |
| 45 | 0x00 | Motor 5 Current MSB<br>= 0 | 0x14+0x00=0x14 |

| | | | |
|---|---|---|---|
| 46 | 0x00 | Motor 5 Current LSB =0 | 0x14+0x00=0x14 |
| 47 | 0x14 | Motor 5 temperature = 39.1 °C | 0x14+0x14=0x28 |
| 48 | 0x00 Reserved for future use | Motor5 Data 7 | 0x28+0x00=0x28 |
| 49 | 0x28 | Checksum | 0x28 |
| 50 | 0xE5 | EOM 0xE5 | |