**DAQCS MSP430-Teensy SPI Interface Control Document**

This document lays out the messaging interface between the MSP430 and Teensy microcontrollers located on the Data Acquisition (DAQCS) subsystem of the HABIP. The interface is over the SPI bus, in which the MSP430 is the master device, and the Teensy is the slave. The SPI buffer shall be set up such that it is 8 bits long, allowing for three different transmissions per message. This document will have two sections, a series of messages that go from the MSP430 to the Teensy, and a series of messages that go from the Teensy to the MSP430. The general formatting of the messages is as follows:

* Each message shall be 24 bits long. The first 16 bits shall be used as information and the last 8 bits will be used as a cyclical redundancy check (CRC) to verify the validity of the message.
* The CRC is calculated using polynomial long division. The dividend of the division is the most significant 16 bits of the message. The divisor of the division is the following polynomial: 0xB1.This can be expanded as x8 + x5 + x4 + 1.This is the CRC-8 Dallas/Maxim polynomial. The remainder of the division is the CRC value that shall be included in the message. If the CRC check fails either the state of the SPI bus, or the entire SPI bus shall be reset.
* Each message shall include a four bit header representing the message number associated with it. This allows for 16 unique messages. If necessary for growth the message numbers can be reused (ie. there can be a message 0x0 going from the MSP430 to the Teensy, as well as a message 0x0 going from the Teensy to the MSP430) which would allow for 32 unique messages. Unless it is necessary in future growth, message numbers will only be used once.
* Bit definitions are defined in the series of tables below.

**Message from DACQS MSP430 to DACQS Teensy**

**MSP430 Resend Data - Message 0x00:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | Not Used | Not Used | Not Used | Not Used |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Not Used | Not Used | Reset Bus | Prev. Msg. | Msg. Num | Msg. Num | Msg. Num | Msg. Num |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Number** | **Definition** |
| 23:20 | "0000" Header to Identify Message |
| 19:14 | Not Used |
| 13 | Reset Bus  0: No Action  1: Reset SPI Bus |
| 12 | Resend Previous Message:  0: See Bits 3:0  1: Resend Previous Message |
| 11:8 | Message Number(1):  Message Number to be resent |
| 7:0 | CRC(3) |
| (1) This message shall be used if the data received is corrupted. If the message number is corrupted bit number 4 can be set in order to resend whatever the previous message was.  (2) This message number shall only be valid if bit 4 is set to 0.  (3) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1). | |

**MSP430 Instruction - Message 0x02:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 1 | 0 | Seg. Select | Seg. Select | Seg. Select | Seg. Select |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Motor | Motor | Motor | Battery | Temp. | IMU | IMU | IMU |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Number** | **Definition** |
| 23:20 | "0010" Header to Identify Message |
| 19:16 | Segment Select (1):  0000 - Pressure Data  0001 - IMU Data - Bits 10:8  0010 - Temperature Data - Bit 11  0011 - Battery Data - Bit 12  0100 - Motor Data - Bit 15:13  0101 - All Gyroscope Data  0110 - All Acceleration Data  0111 - All IMU Data  1000 - Both Temperature Sensors  1001 - Main Battery Data  1010 - All Motor Data  1011 - All on Board Data  1100 - Not Used  1101 - Not Used  1110 - Not Used  1111 - Not Used |
| 15:13 | Motor Data:  000 - Motor Enable  001 - Motor Direction  010 - Motor Battery Voltage  011 - Motor Current Draw  100 - Motor Speed  101 - Not Used  110 - Not Used  111 - Not Used |
| 12 | Battery Data:  0 - Supply Voltage  1 - Supply Current |
| 11 | Temperature Data:  0 - On Board  1 - Off Board |
| 10:8 | IMU Data:  000 - X Gyroscope  001 - X Acceleration  010 - Y Gyroscope  011 - Y Acceleration  100 - Z Gyroscope  101 - Z Acceleration  110 - Not Used  111 - Not Used |
| 7:0 | CRC(2) |
| **Notes:**  (1) The Segment Select Bits shall be used to identify which piece of data is being requested. If '0000' is received the temperature data is requested and 7:0 are ignored. If 0001 through 0100 are requested the corresponding set of bits will be looked at to further identify which sensor is being requested. Segments 0101 through 1011 are used to request multiple pieces of data. If multiple pieces of data are requested bits 7:0 are ignored. For multiple sensors a Message "0x03" (and Message "0x04" if necessary) are sent for each individual piece of data. Segments 1100 through 1111 are not used.  (2) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1). | |

**Teensy Response - Message 0x05**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 1 | 0 | 1 | Not Used | Not Used | Not Used | Not Used |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Not Used | Not Used | Not Used | Not Used | Not Used | Not Used | Not Used | Not Used |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Number** | **Definition** |
| 23:20 | "1001" Header to Identify Message |
| 19:8 | Not Used |
| 7:0 | CRC(2) |
| (1) This message shall be used to open up the bus in order for the Teensy to respond to the MSP430 with its data. This message shall either be sent once or twice, depending on if a message 0x04 is expected as well.  (2) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1). | |

**Messages from DACQS Teensy to DACQS MSP430**

**Teensy Resend Data - Message 0x01:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 1 | Not Used | Not Used | Not Used | Not Used |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Not Used | Not Used | Reset Bus | Prev. Msg. | Msg. Num | Msg. Num | Msg. Num | Msg. Num |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Number** | **Definition** |
| 23:20 | "0001" Header to Identify Message |
| 19:14 | Not Used |
| 13 | Reset Bus  0: No Action  1: Reset Bus |
| 12 | Resend Previous Message:  0: See Bits 3:0  1: Resend Previous Message |
| 11:8 | Message Number(1):  Message Number to be resent |
| 7:0 | CRC(3) |
| (1) This message shall be used if the data received is corrupted. If the message number is corrupted bit number 4 can be set in order to resend whatever the previous message was.  (2) This message number shall only be valid if bit 4 is set to 0.  (3) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1). | |

**Teensy Instruction Acknowledge - Message 0x03:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 0 | 1 | 1 | Seg. Select | Seg. Select | Seg. Select | Msg. 0x04 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Seg. Data | Seg. Data | Seg. Data | Binary Dat | Data | Data | Data | Data |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Number** | **Definition** |
| 23:20 | "0011" Header to Identify Message |
| 19:17 | Segment Select (1):  000 - Pressure Data  001 - IMU Data  010 - Temperature Data  011 - Battery Data  100 - Motor Data  101 - Not Used  110 - Not Used  111 - Not Used |
| 16 | Message 0x4 expected(3)  0 - Message 0x4 is expected  1 - Message 0x4 is not expected |
| 15:13 | Segment Data(2) |
| 12 | Binary Result(7) |
| 11:8 | Data(4,5) |
| 7:0 | CRC(6) |
| **Notes:**  (1) The Segment Select Bits shall be used to identify which group of sensors to look at. Segments 0101-1111 are currently not being used.  (2) Segment Data shall be used to say which individual sensor is being recorded. This data should be taken from the Message 0x2 table. If the data does not require three bits (ie. Battery is only one bit) the most significant bits shall be set to '0'.  (3) If the returned data is binary it shall be included in this message, if it is a data string it shall be included in message 4. This bit says if a message 0x4 will be sent as well.  (4) If bit 16 is a 1 that means that the data coming back is binary. If this is the case the result will be stored in bit 12.  (5) Bits 11:8 are the most significant nibble of the data. The data is stored in two's compliment format.  (6) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1).  (7) If the result is binary it shall be stored in this bit. This is used for Motor Enable and Motor Direction. This bit is only valid if bit 16 is a 1. | |

**Teensy Data - Message 0x04**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| 0 | 1 | 0 | 0 | Data | Data | Data | Data |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Data | Data | Data | Data | Data | Data | Data | Data |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CRC | CRC | CRC | CRC | CRC | CRC | CRC | CRC |

|  |  |
| --- | --- |
| **Bit Numbers** | **Description** |
| 23:20 | "0100" Header Used to Identify Message |
| 19:8 | Data(1) |
| 7:0 | CRC(2) |
| **Notes:**  (1) See "Data from Teensy Expanded" to see information. The most significant nibble is taken from Message 3. This allows for 16 bits of data to come back. The data is stored in two's compliment format.  (2) The polynomial used to calculate the CRC shall be x8 + x5 + x4 + 1 (0xB1). | |

**Data from Teensy Expanded:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sensor** | **Number of Bits** | **Format** | **LSB** |
| Board Temperature(1) | 16 | Sign: 0 = +, 1 = -  XXX.XXX ◦C | TBD |
| Board Pressure(1) | 16 | XXXX.XXX mBar | TBD |
| Off Board Temp.(1) | 16 | Sign: 0 = +, 1 = -  XXX.XXX ◦C | TBD |
| Battery Supply Voltage | 16 | XX.XXX V | TBD |
| Battery Supply Current | 16 | XXX.XXX mA | TBD |
| Gyroscope X | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s | TBD |
| Acceleration X | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s^2 | TBD |
| Gyroscope Y | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s | TBD |
| Acceleration Y | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s^2 | TBD |
| Gyroscope Z | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s | TBD |
| Acceleration Z | 16 | Sign: 0 = +, 1 = -  XXXX.XX deg/s^2 | TBD |
| Motor Speed | 16 | Sign: 0 = +, 1 = -  XXXX.XX RPM | TBD |
| Motor Current | 16 | TBD | TBD |
| Motor Voltage | 16 | XX.XXX V | TBD |
| Motor Direction | 1 | 1 = CW, 0 = CCW | N/A |
| Motor Enable | 1 | 1 = On, 0 = Off | N/A |
| (1) This data needs to be multiplied by 2^4 when it's received. | | | |