

This document describes how the various microprocessors will communicate with the IHU.

IHU Command Interface

Description of interboard
communication protocol

Revision: [1.1.0]
Eric Curtland

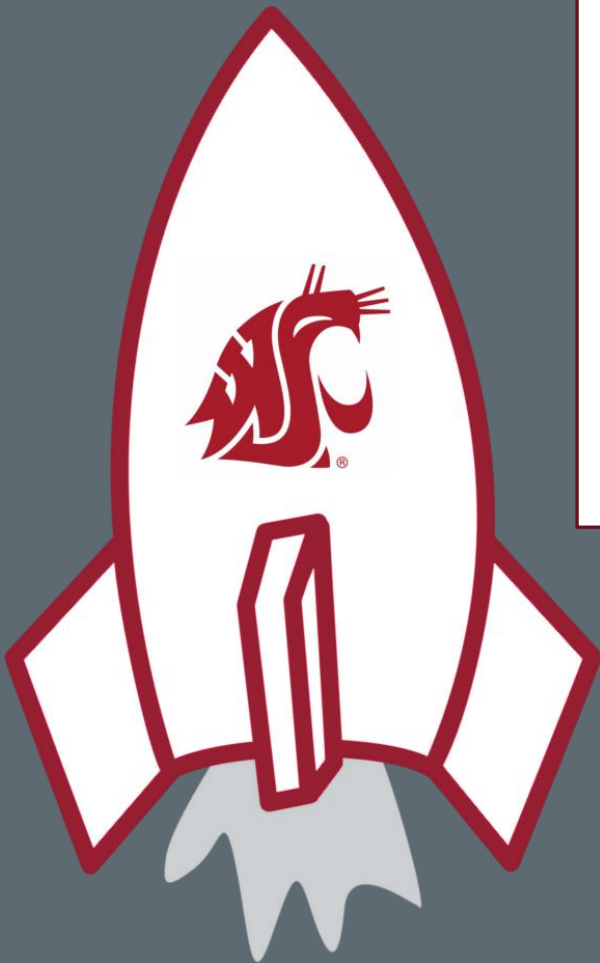


Table of Contents

1 Hardware Layer..... 2

2 Software Layer 2

2.1 Command Packet 2

2.1.1 CRC Check Sum 3

1 Hardware Layer

The hardware interface between the IHU and the sub-boards will use a standard implementation of I²C with a clock wire and data wire. See [Wikipedia's I2C article](#) for more information.

2 Software Layer

The software layer will consist of standard commands from the IHU to each board. The commands will all be 8 Bytes in length with the first byte corresponding to the command id and the last byte being for check sum. For commands that require more than 6 bytes to be transferred a unique command will be sent which will describe the length of the data packet.

2.1 Command Packet

The following serves as a visual guide to the commands described above.

Generic Command:

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	Command ID	Command data						Checksum

Command ID: Unique identifier for each command ranging from 0x01-0xFF

Command Data: Provides 6 bytes for data associated with the command.

Check Sum: A CRC algorithm will be used to ensure validity of sent command.

Long Command:

Byte Offset	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
0x00	0x00	Unsigned 16-bit Integer						Checksum

Command ID: The long message will use ID 0x00

Command Data: The length of the following message will be sent as an unsigned 16-bit integer

Check Sum: A CRC algorithm will be used to ensure validity of sent command.

2.1.1 CRC Check Sum

For error checking we will use an implementation of a CRC8 algorithm using 100101111 as the polynomial. For more documentation on the algorithm see [This Website](#). For use run the algorithm on the message to be sent and append the crc value to the end. When reading the message run the crc algorithm again and the crc value will be 0 if there is no error.