# Interface Control Document

Galvanic Vestibular Stimulation Collision Avoidance System

| | | |
|---|---|---|
| Initial Release | P. Quevedo | Jan 24 2010 |
| | | |

# Table of Contents

# 1. Introduction

The Galvanic Vestibular Stimulation based Collision Avoidance System communicates with other sensors in the system using the CAN 2.0A/B protocol [1] as defined by BOSCH and implemented on a PIC18F microcontroller [2]. Any sensor can be added to the network and communicate with the master control unit to provide either distances to objects, navigation information or trajectory measurements. The master control unit is equipped with onboard sensors to determine subject movement and heading however if more precise measurements are required they can be sent in from an external Trajectory Measurement Unit (TMU) over the CAN bus.

This document will outline the necessary information to add a new node to the network and communicate with the master control unit (MCU).

# 2. Number Scaling

Q scaling is used to represent floating point values over the CAN Bus. It is denoted with the syntax “Qi.f “ where “i” indicates the number of bits used to represent the integer portion of the number and “f” the number of bits used to represent the fractional portion. The sum of “i” + “f” indicates the number of bits in total that are used to represent the floating point number.
When looking just at the integer portion of the number, the LSB has value $2^0$ and the MSB $2^{(i-1)}$. Looking at the fractional portion, the LSB has value 2-$^{f}$ and the MSB $2^{-1}$. The resolution of the Qi.f format is given by $2^{-f}$. As an example consider the format Q5.3, an 8 bit number.

$$\bar{2}^4\ \bar{2}^3\ \bar{2}^2\ \bar{2}^1\ \bar{2}^0\ \bar{2}^{-1}\ \bar{2}^{-2}\ \bar{2}^{-3}$$

The number 25.125 can be represented as a Q5.3 number having a hex value of 0xCA (0d202). The following C macros can be used to convert to and from a Qi.f number.

```
#define    SCALE(x,f)       x*(1<<f)
#define    UNSCALE(x,f)     ((float)x)/(1<<f)
```

“x” is the number to be scaled and f is the number of bits used to represent the fractional portion.

# 3. Configuration

## 3.1. Frame Format

All messages will be sent using a standard “Base Frame Format” - CAN identifier fields set to 11 bits in length. Remote Transmission Requests are not supported and the RTR bit should not be set in any messages.

### 3.2. Baud Rate

The baud rate used by the MCU is set to 125 kpbs. Any sensor being added to the node must be set to this baud rate in order to communicate successfully. Parameters used to generate this on a Microchip PIC18 processor from a 40MHz clock (10MIPS) with a clock prescaler of 10 are as follows:

i. Synchronized Jump Width (SJW) 2 Tq[1]
ii. Synchronization Segment 1 Tq
iii. Phase Segment 1 7 Tq
iv. Phase Segment 2 7 Tq
v. Propagation Segment 1 Tq[2]

## 4. Communication

### 4.1. Identifier Frame

Any message sent over the CAN bus will have one of 5 identifier frame values.

| | | | Identifier Frame Values |
|---|---|---|---|
| **Value** | **Destination** | **Source** | **Message Type** |
| 0x001 | MCU | Sensor Node | All |
| 0x002 | MCU | Navigation Node | All |
| 0x004 | MCU | IMU Node | All |
| 0x008 | Global | MCU | Node Requests |
| 0x010 | Global | MCU | Navigation Output |

### 4.2. Obtain ID Number

Before any data can be sent to the MCU the node being added to the network must obtain a node ID number from the MCU. When sending any data to the MCU this node ID must be included in order for the MCU to successfully process the data. The node will send a message with the following information:

| | | | Request for Node ID Message<br>Message Frequency: Once |
|---|---|---|---|
| Identifier Frame | 0x001 (Sensor Nodes)<br>0x002 (Navigation Nodes)<br>0x004 (IMU Nodes) | | |
| Data Frame [0] | Node ID = 0xFF | uint8 | Request for a Node ID. |
| Data Frame [1] | ID Tag – High Byte | uint16 | Any unique random number |
| Data Frame [2] | ID Tag - Low Byte | | |
| Data Frame [3-7] | Reserved | | |

Upon receiving a "Request for Node ID Message", the MCU will respond with a message as follows:

[1] Tq = Time Quanta
[2] Propagation Segment calculated for 10m wire @ 5.5nSec/meter

| Response for Node ID Message<br>Message Frequency: Once | | | |
|---|---|---|---|
| Identifier Frame | 0x008 (Global – Node Requests) | | |
| Data Frame [0] | Node ID | uint8 | Unique ID issued by MCU |
| Data Frame [1] | ID Tag – High Byte | uint16 | Received ID Tag |
| Data Frame [2] | ID Tag - Low Byte | | |
| Data Frame [3-7] | Reserved | | |

The ID tag received by the MCU in the "Request for Node ID Message" will be sent back in the "Response for Node ID Message". Any node awaiting a Node ID must ensure that the Received ID Tag matches the ID tag that was sent in the request.
Once a Node ID has been received from the MCU, the ID Tag is no longer required.

## 4.3. Sensor Data Message

This message contains information about any objects detected by the sensor node.

| Sensor Data Message<br>Message Frequency: 1-30Hz | | | |
|---|---|---|---|
| Identifier Frame | 0x001(Sensor Node) | | |
| Data Frame [0] | Node ID | uint8 | Unique ID issued by MCU |
| Data Frame [1] | Angle Start - High Byte | int16 | Range: -180 - 180 deg |
| Data Frame [2] | Angle Start - Low Byte | | |
| Data Frame [3] | Angle Width – High Byte | uint16 | Range: 1 - 359 |
| Data Frame [4] | Angle Width – Low Byte | | |
| Data Frame [5] | Distance – High Byte | uint16 | Scaled Q9.7 |
| Data Frame [6] | Distance – Low Byte | | |
| Data Frame [7] | Reserved | | |

The angular spread that the sensor covers is defined by two integer values. Angle Start defines what angle the sensor starts it coverage relative to the front of the person. From a top down perspective of a person, 0 defines the forward looking position, to the left is negative, to the right is positive and +/- 180 is directly behind.
Angle width defines how many degrees the sensor area covers starting from Angle Start and incrementing in a clockwise fashion (as viewed from a top down perspective). The value must be between 1 and 359.

## 4.4. TMU Data Message

Not currently supported.

## 4.5. Navigation Data Message

Not currently supported.

## 4.6. MCU Navigation Output

The MCU will broadcast the current heading and number of steps taken since the device was activated over the CAN bus at a frequency of 10Hz.

| MCU Navigation Output<br>Message Frequency: 10Hz | | | |
|---|---|---|---|
| Identifier Frame | 0x010(Global – Navigation Output) | | |
| Data Frame [0] | Heading – High Byte | uint16 | Range: 0-359<br>N=0;W=90 |
| Data Frame [1] | Heading – Low Byte | | |
| Data Frame [2] | Pedometer Count – High Byte | uint16 | Range: 0-65535 |
| Data Frame [3] | Pedometer Count – Low Byte | | |
| Data Frame[4-7] | Reserved | | |

## 5. References

[1] "CAN Specification Version 2.0", Robert Bosch GmbH, 1991.

[2] "PIC18F2480/2580/4480/4580 Data Sheet", Microchip Technology Inc., 2007