Driver Controls CAN Bus Communications Protocol Specification

19 March 2007

Brisbane, Australia
http://www.tritium.com.au

TABLE OF CONTENTS

1 INTRODUCTION

This document describes the protocol used to communicate over a CAN bus connection to and from the Tritium driver controls module.

2 CAN CONFIGURATION

2.1 HARDWARE

The CAN hardware interface used is compatible with the CAN 2.0B standard. The supported bit rates (bits per second) are 1 Mbps (default), 500 kbps, 250 kbps, 125 kbps, 100 kbps and 50 kbps.

2.2 SOFTWARE

The CAN protocol uses data frames for most communication. Remote frames are also enabled. The identifier field uses the standard frame definition length of 11 bits. All measurement data is transmitted using IEEE single-precision 32-bit format (IEEE 754) with most significant byte (MSB) sent first.

Bit Length	1	11	6	8 Bytes	16	2	7
	Start	Identifier	Control	Data Field	CRC	Ack	End

Figure 1. CAN data frame

2.2.1 Identifier

The identifier field has been split into two sections for Tritium devices. Bits 10-5 contain the device identifier and bits 4-0 contains the message identifier associated with that device, as shown in Figure 2. This means that there is a maximum of 64 Tritium devices that can be on the CAN bus at any one time. Each Tritium device can have 31 different types of messages, with the first message identifier being reserved as the device identification message.

10 … 5	4 … 0
DEVICE IDENTIFIER	MESSAGE IDENTIFIER

Figure 2. CAN device identifier address format

2.2.2 Data Field

The data field in all frames is fixed at 8 bytes (64 bits) and two IEEE 754 32-bit floating point variables as shown in Figure 3. The data field is sent and expected to be received least significant bite first. This allows a direct overlay of a float[2] array and char[8] array on a little endian processor, such as, an Intel x86 (or clone) or the TI MSP430 in the driver controls.

High float

```
 s eeeeeeee mmmmmmmmmmmmmmmmmmmmmmm
 ^ ^       ^ ^                     ^
63 62     55 54                    32
```

Low float

```
 s eeeeeeee mmmmmmmmmmmmmmmmmmmmmmm
 ^ ^       ^ ^                     ^
31 30     23 22                    0
```

Figure 3. Format of the data field in a data frame

2.2.3 Units

Please note that variables described in the following packets with units of percent "%" should be sent with a minimum floating-point value of 0.0 and a maximum value of 1.0. Do not send 100.0 as the maximum value.

2.2.4 Broadcast Frames

These are data frames broadcast from the driver controls to any listening motor controller. These commands contain desired set points for the control software to operate the controller. The commands are sent as required, however there is a maximum permissible delay between consecutive Motor Drive commands to prevent a timeout from occuring in the motor controller.

The driver controls also provides a convenient interface for a variety of switches and pushbuttons, that are not used directly by the motor controller. For example, another relay output module on the CAN bus may listen for the Switch Position frame and turn the indicator (blinker) lights on and off as appropriate.

2.2.5 Remote Frames

All frame types sent by the driver controls may also be requested on demand by using the CAN bus remote frame transmit (RTR) mechanism. This is accomplished by the remote device transmitting a frame containing the address (ID) of the frame request, with the RTR bit set. The driver controls will reply with the requested packet after a short delay. This mechanism is independent of the timer-based output of control frames from the driver controls.

3 BROADCAST MESSAGES

3.1 IDENTIFICATION

3.1.1 Identification Information

ID: Driver Controls Base Address + 0

Interval: 1 second

Variable	*Bits*	*Type*	*Description*
Serial Number	63 .. 32	Uint32	Device serial number, allocated at manufacture.
Tritium ID	31 .. 0	char[4]	"TRIb" stored as a string. msg[0] = 'T', msg[1] = 'R'...

The periodic broadcast of this message cannot be disabled. It is needed to help find the driver controls on the network if the base address is lost or mis-configured by the user.

3.2 DRIVE COMMANDS

3.2.1 Motor Drive Command

ID: Driver Controls Base Address + 1

Interval: 100 ms

Variable	*Bits*	*Units*	*Description*
Motor Current	63 .. 32	%	Desired motor current set point as a percentage of maximum current setting.
Motor Velocity	31 .. 0	m/s	Desired motor velocity set point in metres/second.

The WaveSculptor motor controller must receive a Motor Drive Command frame at least once every 250ms. If this does not occur, the controller will assume that communications have failed and will halt all motor control functions, placing the system into neutral.

3.2.2 Motor Power Command

ID: Driver Controls Base Address + 2

Interval: 100 ms or as needed

Variable	*Bits*	*Units*	*Description*
Bus Current	63 .. 32	%	Desired set point of current drawn from the bus by the controller as a percentage of absolute bus current limit.
Reserved	31 .. 0	–	–

3.2.3 Reset Command

ID: Driver Controls Base Address + 3

Interval: No fixed interval, not used during normal operation

Variable	*Bits*	*Units*	*Description*
Unused	63 .. 32	–	–
Unused	31 .. 0	–	–

Send a command from this address to reset the WaveSculptor.

3.3 SWITCH COMMANDS

3.3.1 Switch position / activity

ID: Driver Controls Base Address + 4

Interval: 100 ms

Variable	*Bits*	*Type*	*Description*
Switch Position	63 .. 32	Uint32	Current position of the switch inputs on the driver controls module DB37 connector: ***Bits*** — ***Parameter*** 31 .. 18 — Unused 17 — Right indicator output (90 per minute) 16 — Left indicator output (90 per minute) 15 — Unused 14 — Onboard (internal) debug button 13 — Encoder 1 pushbutton 12 — Encoder 2 pushbutton 11 — Right indicator switch 10 — Left indicator switch 9 — Hazards switch 8 — Horn switch 7 — Ignition – Accessory position 6 — Ignition – ON (Run) position 5 — Direction (0 = Forward, 1 = Reverse) 4 — Brake 1 switch 3 — Brake 2 switch 2 — Lights – Side / Running lights switch 1 — Lights – Low Beam switch 0 — Lights – High Beam switch
Switch Activity	31 .. 0	Uint32	Shows if the switch has changed state since the last time this CAN frame was sent. 1 = Switch has changed 0 = No change Bit positions are identical to the Switch Position bitfield shown above.

4 OPERATION

4.1 TORQUE CONTROL MODE

This control mode is analogous to the way a normal vehicle is operated, with the accelerator (gas) pedal controlling the motor's torque output. To drive in this mode, the driver controls should send a Motor Drive Command frame at least once every 200ms containing:

1) Desired motor current setting in percent, corresponding to pedal position
2) Maximum velocity (eg 100m/s)

With these settings, the motor controller will never be able to reach the setpoint velocity, so the operation of the vehicle will be limited by the motor current, which is proportional to torque.

4.2 SPEED CONTROL MODE

This control mode is analogous to cruise control in a normal vehicle. To drive in this mode, the driver controls should send a Motor Drive Command frame at least once every 200ms containing:

1) Maxiumum motor current (100%)
2) Desired vehicle velocity in metres per second

With these settings, the motor controller will use the maximum available current to reach the desired setpoint velocity.

The usual method of setting the target speed would be to monitor "Velocity Measurement" frames from the Wavesculptor, and when the driver wishes to set the target velocity (using some combination of input switches) then the driver controls should begin to transmit the last known velocity measurement as the target velocity.

Of course, any pre-programmed speed could also be used, however care must be taken when engaging speed control mode under these circumstances, as the vehicle will use the maxiumum allowable torque to reach the target velocity setpoint as fast as possible.

4.3 COMINBATION CONTROL MODES

Any combination of the above two modes can be implemented, as desired by the user. For example, in speed control mode, while setting the target vehicle velocity, a motor current setting of 50% could be sent, resulting in the motor controller only using half of it's torque capability to try and reach the velocity setpoint.

4.4 POWER CONTROL

The driver controls can also dynamically limit the maximum current that can be drawn by the Wavesculptor from the vehicle's power bus (battery). This may be useful in some applications to minimise high-current events when operating with a discharged main battery pack, or as part of an overall vehicle control strategy during racing.

To limit the power, the driver controls should send a Motor Power Command frame, containing the desired bus current as a percentage of maximum. This frame may be sent as often as desired, however please note that a Motor Drive Command frame must still be sent at least once every 200ms to prevent the Wavesculptor from shutting down.

5 REVISION RECORD

REV	*DATE*	*CHANGE*
1	19 March 2007	Document creation (JMK)