RaceGrade Specification

Document Number	RG_SPEC-0024			
Title	8 Channel Thermocouple to CAN - TC8			
Can Speed: 1Mbps	**Base CAN ID:** 0x0F0h	**TC8 Mode:** 0 = E888	**Serial Number:**	**Checked By:**

Firmware Revision	Date	Prepared By	Change History
1.41	8/10/2015	Steven Bravek	Updated for manager
1.44	12/15/2015	Steven Bravek	Increased CAN message output rate to match E888
1.45	2/18/2016	Steven Bravek	Added user definable transmission rate in standard mode
1.46	5/16/2016	Steven Bravek	Fixed programming bug

A CAN based expansion module that allows for up to 8 thermocouple sensors to be connected. The module works with non-amplified K-type thermocouples only. Version 1.4 of TC8 is capable of being programed via CAN. This is usefully if you wish to have multiple TC8 on one CAN bus or are using a TC8 with a non Motec product. There are six different modes to choose from. The first four modes (0-3) mimic an E888. Modes four and five let you choose what CAN ID the messages are sent out on.

NOTE: When no sensor is connected to an input, the output reads between 1050 to 1250°C.

Part Number: RG TC8

Specifications:

Input Temp Range: 0 to 1000°C +/-4°C
Thermocouple Style: K-type only
Operating Voltage Range: 9 to 24 vDC
Operating Current Draw: < 0.1 Amps
Operating Temp Range: -40°C to 120°C
Dimensions: 3.5" x 2.26" x 1.0"
Weight: 173 grams
Current version: 1.46

Connection:

Mating Connector: ASL606-05SN
pin 1 – Ground
pin 2 – N/C
pin 3 – Power
pin 4 – CAN Lo
pin 5 – CAN Hi

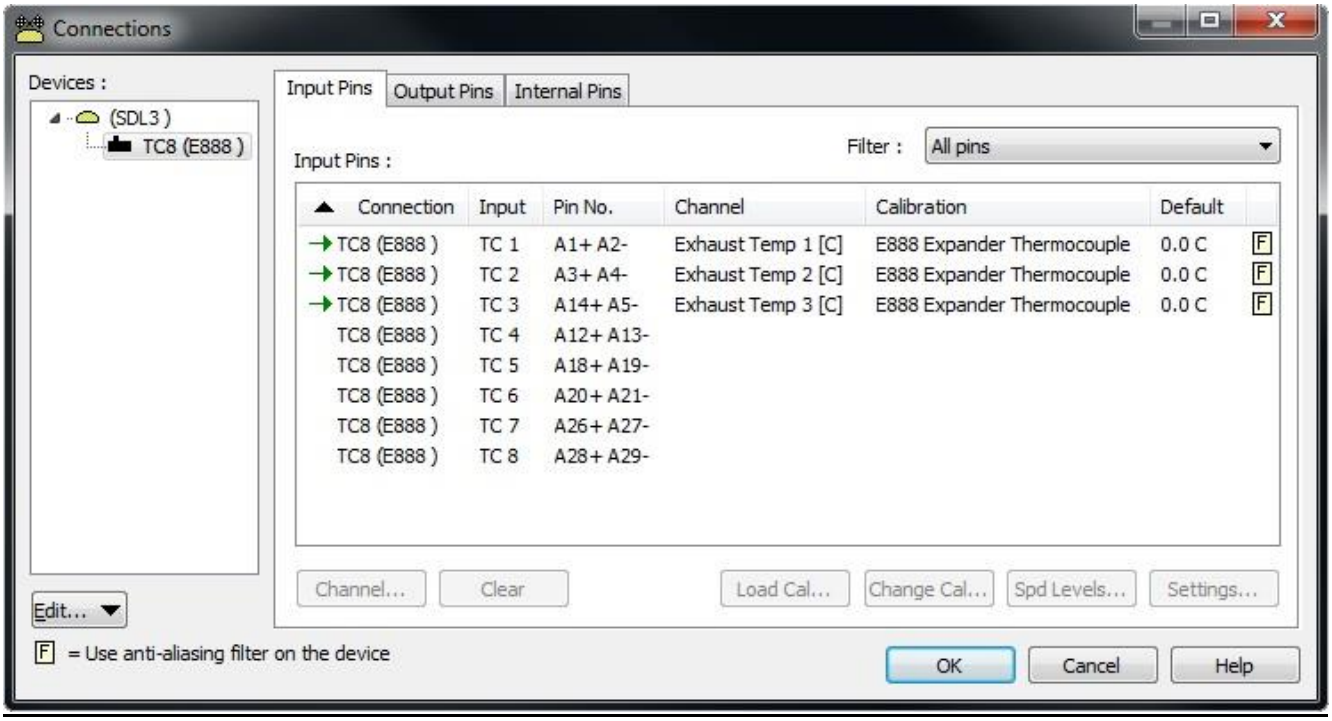
CAN Messaging:

CAN Bus speed: 1 Mbps, 500 Kbps, 250 Kbps, 125 Kbps
Setup is done similar to the MoTeC E888.
Channels are configured as E888 thermocouples.
Can be used on M84/Mx00/M1 ecus and all dashes.

Installation

Setting up the TC8 in Dash Manager is done similarly to an E888:

1. Under the "Connections" pulldown, select the "Devices" menu.
2. Right-Click the dash in the Devices list and click "Add".
3. Enter the following settings:
    a. Type: E888.
    b. Base Address: Depends on the TC8's configuration, see below. Default: 0x0F0.
    c. Name: User preference. Has no effect on the setup. Typical: "TC8".
    d. Can Bus: Select the bus the TC8 is wired into.
4. Select the newly added E888.
5. Right-Click each input, click "Select Channel".
6. Select a temperature channel with a resolution of 0.1 Celsius. Typical: "Exhaust Temp X"
7. Select the input, click "Load Calibration".
8. Select "E888 Expander Thermocouple".
9. Repeat for all inputs.
10. Setup Finished.



* After finishing setup, you connections screen should look like this.

Manager:

The management software uses a PCAN-USB by Peak Systems to communicate with the TC8 over CAN. The manager lets the user change CAN ID, CAN bus speed, and transmission mode. If you don't have a PCAN the follow programing message can be sent to change CAN ID, and transmission mode.

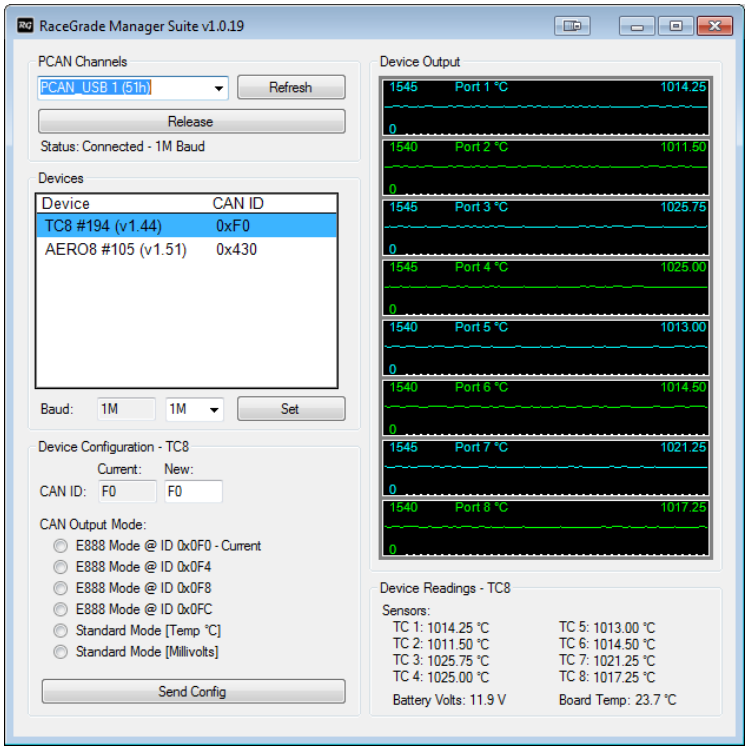
To program without the manager, send three CAN messages on ID 0x08. You only need to send a set of three CAN messages once to successfully program the TC8. The message structure is listed below.

The manager uses the following CAN ID's to communicate with the TC8: 0x006, 0x007, 0x008, 0x009, 0x014, and 0x016.

To download the latest software go to: http://www.racegrade.com/downloads.html

Message 1

```
BYTE(0)   = 0x01; //Compound Message Id
BYTE(1)   = 0x52; // R
BYTE(2)   = 0x47; // G
BYTE(3)   = 0x54; // T
BYTE(4)   = 0x43; // C
BYTE(5)   = 0x38; // 8
BYTE(6)   = 0x05; // End Message
BYTE(7)   = 0x00; // Unused
```



Message 2

```
BYTE(0)   = 0x02; //Compound Message Id
BYTE(1)   = 0x4D; // M
BYTE(2)   = 0x53; // S
BYTE(3)   = 0x47; // G
BYTE(4)   = 0x46; // F
BYTE(5)   = 0x54; // T
BYTE(6)   = 0x0X; // Message Format 0 - 5
BYTE(7)   = 0x00; // Unused
```

Message 3

```
BYTE(0)   = 0x03; //Compound Message Id
BYTE(1)   = 0x43; // C
BYTE(2)   = 0x41; // A
BYTE(3)   = 0x4E; // N
BYTE(4)   = 0x49; // I
BYTE(5)   = 0x44; // D
BYTE(6)   = 0x0X; // Top 3 Bits of Can Id 0 - 7
BYTE(7)   = 0xXX; // Lower 8 Bits of Can id
```

The following message formats are available for selection in Message 2, Byte 6:

- 0 = E888 0x0F0 (Standard)
- 1 = E888 0x0F4
- 2 = E888 0x0F8
- 3 = E888 0x0FC
- 4 = User Selectable CAN ID / Standard format / Output is Temperature in °C
- 5 = User Selectable CAN ID / Standard format / Output is Millivolts

The first four modes (0-3) will mimic the Motec E888 CAN Bus expander transmitted temperature. The E888 specific channels like cold junction reporting, firmware level, etc are not transmitted by the TC8 so these channels should be removed from any template used to receive the data. The only differences between the first four modes are the actual address in use.

The CAN ID of the TC8 is only user selectable when the message format is set to 4 or 5 (all other can IDs are fixed to E888 can IDs). Message 3 bytes 6 and 7 is where the CAN ID is programmed. Byte 6 contains the top 3 bits of the CAN ID and byte 7 will contain the lower 8 bits of the CAN ID. This completes an 11 bit standard CAN ID in the range of 0x000 to 0x7FF.

Mode 4 Use

Mode 4 will transmit each thermocouple amplifiers measured temperature value so no calibration is required in the receiving device.
The Temperature will be transmitted in Degrees C with 0.1 resolution. Range is 0 to 1000°C.

Mode 5 Use

Mode 5 will transmit each thermocouple amplifiers raw voltage value such that calibration may be done in the receiving device.
The voltage will be transmitted in millivolts. Range 0 to 5000 millivolts.

Mode 4 and 5 Format

Message 1

User selected CAN ID 0x000 – 0x7FF

```
BYTE(0)  = High byte channel 1
BYTE(1)  = Low  byte channel 1
BYTE(2)  = High byte channel 2
BYTE(3)  = Low  byte channel 2
BYTE(4)  = High byte channel 3
BYTE(5)  = Low  byte channel 3
BYTE(6)  = High byte channel 4
BYTE(7)  = Low  byte channel 4
```

Message 2

User selected CAN ID + 1

```
BYTE(0)  = High byte channel 5
BYTE(1)  = Low  byte channel 5
BYTE(2)  = High byte channel 6
BYTE(3)  = Low  byte channel 6
BYTE(4)  = High byte channel 7
BYTE(5)  = Low  byte channel 7
BYTE(6)  = High byte channel 8
BYTE(7)  = Low  byte channel 8
```

Message 3

User selected CAN ID + 2

```
BYTE(0)  = High byte serial number
BYTE(1)  = Low  byte serial number
BYTE(2)  = High byte firmware version
BYTE(3)  = Low  byte firmware version
BYTE(4)  = High byte battery volts
BYTE(5)  = Low  byte battery volts
BYTE(6)  = High byte board temp
BYTE(7)  = Low  byte board temp
```