MPC1 MULTI PURPOSE CAN CONTROLLER - REFERENCE MANUAL

Baldur Gíslason

December 17, 2018

Contents

1 Introduction

MPC1 is a multi purpose CAN controller, capable of routing data across two incompatible CAN networks, intercepting and modifying selected data passing between two legs of an otherwise compatible network, reading speed or analog sensors and outputting the values onto a CAN network or controlling tachometers, speedometers, relays or PWM consumers based on data received via CAN or any combination of these functions all at once.
There are a number of user configurable logic functions as well as general purpose PID controllers, general purpose lookup functions and more.

2 Wiring

2.1 Pin-outs and description

2.1.1 Pin numbering

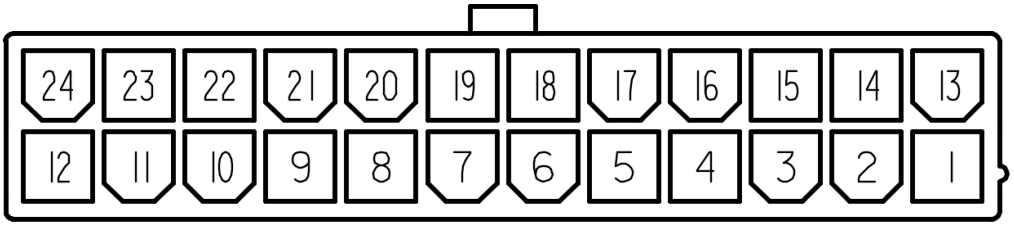

Figure 2.1: Connector A

2.1.2 Connector A pin-out

Pin	I/O	Function	Note
1	OUT	5V supply for sensors	200mA max
2	IN	Analog 0	51kΩ pull-up to 5V
3	IN	Analog 1	51kΩ pull-up to 5V
4	OUT	Ground return for sensors	
5	IN	Analog 4 0-5V	51kΩ pull-up to 5V
6	IN	Analog 5 0-5V	51kΩ pull-up to 5V
7	IO	CAN 1 high	120Ω termination on board
8	IO	CAN 1 low	120Ω termination on board
9	OUT	Output 1	Low-side switch, 5A max
10	OUT	Output 2	Low-side switch, 5A max
11	IN	Analog 7	51kΩ pull-up to 5V
12	IN	Power ground	
13	IN	Analog 6	51kΩ pull-up to 5V
14	IN	Analog 2	3kΩ pull-up
15	IN	Analog 3	3kΩ pull-up
16	IO	CAN 2 high	120Ω termination on board
17	IO	CAN 2 low	120Ω termination on board
18	IN	Digital in 1	6kΩ pull-up to 5V. 12V and AC tolerant (suitable for VR speed sensors)
19	IN	Digital in 2	6kΩ pull-up to 5V. 12V and AC tolerant (suitable for VR speed sensors)
20	IN	Digital in 3	6kΩ pull-up to 5V. 12V and AC tolerant (suitable for VR speed sensors)
21	OUT	Output 4	Low-side switch, 5A max
22	OUT	Output 3	Low-side switch, 5A max
23	IN	Digital in 4	6kΩ pull-up to 5V. 12V and AC tolerant (suitable for VR speed sensors)
24	IN	Switched +12V supply	

The combined current of all 4 low side switch outputs should not exceed 6A.

2.2 Usage

The PC interface application, BG Calibrator, can be downloaded from `https://controls.is/calibrator` and firmware updates for the controller may be found at `https://controls.is/firmware`
The remaining sections of this chapter describe different usage scenarios.

2.2.1 Filtered CAN routing

In this scenario, a previously whole CAN network can be broken into two segments and the MPC1 controls what data passes through and can edit the contents of selected packets. Typically used to recalibrate speedometers or change the source of the speed information that passes to certain devices on the CAN network.
Configure each of the CAN networks to the correct speed and set both of the CAN routing parameters to unfiltered initially. Verify that the CAN network is operational after power cycling the controller.
If selected packets that pass from CAN 1 to CAN 2 should be blocked or have their contents altered, set the CAN 1 to CAN 2 routing parameter to filtered, and vice versa for packets passing the other direction.
The CAN bus 1 ingress filter rules control routing of packets received on CAN 1 passing to CAN 2. This works sort of like a networking firewall. First populate the routing IDs that need special treatment, starting from the bottom row (numbered 0 in the routing IDs list). The first row that has an ID of 0 is treated as the default rule for any ID not listed.
Next set up the routing actions table. The routing actions table mirrors the IDs set up in the IDs table and offers a number of different actions. A pass action lets the packet pass unmodified over to the other CAN network. A drop action prevents the packet from crossing the networks. An edit action selects one of the edit rules that may be applied to the packet.
When the action table is parsed by the controller for each incoming packet it starts at the bottom and advances up until it has found a rule matching the ID of the incoming packet or an ID of zero.
The routing rules may be optional, by default the conditions table would have all the rules set to always on, but if a condition input (switch, logic condition or something else) is selected, an alternative action may be selected for that rule when the input is off.

The edit rules allow up to 4 parameters inside a packet passing through to be edited in place.
First select a data type for each of the parameter you would like to edit, the type names signify signed/unsigned, the number of bits to the integer and little or big endian. For example u16be would signify unsigned 16 bits big endian, s8 signified signed 8 bits (no endian for data types of 8 bits or smaller). Most automotive systems use big endian numbers.
Data length cut off should normally be set to 0 for each parameter, but it allows the editing of parameters that may not be a multiple of 8 bits in size. For example if a parameter has 15 bits of integer data with the

upper most bit used for other purposes, use the data length cut off to limit that parameter to 15 bits to avoid conflict with the most significant bit.
Next set the byte location (offset) of the item inside the CAN frame. A value from 0-7 unless the item is of type bit, then the offset can be from 0 to 63.
Replacement data source allows discarding the original data and replace it with data from another source. If this is not desired, select 'Keep original' in that table.
The data selected is then multiplied by the multiplier and the adder can add or subtract from it. The result is overflow protected.