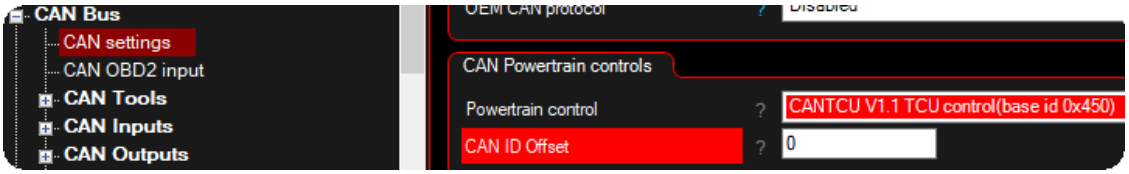
CANTCU - MaxxECU integration v1.1

- **CAN Speed** is configurable
- CAN2.0B, Standard 11bit identifiers
- All 16 bit values are **Little Endian**

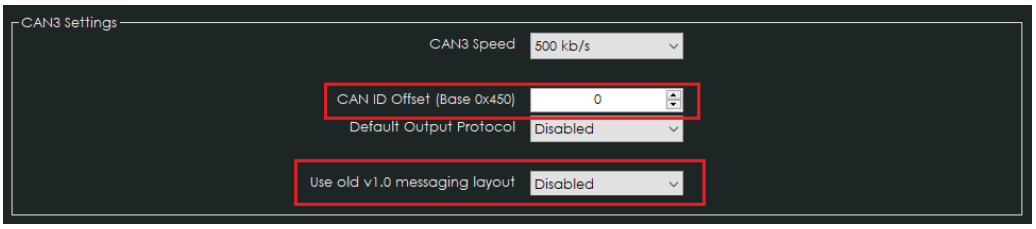
MaxxECU Configuration

- **CANTCU V1.1** needs to be selected in the **CAN Powertrain controls**.



CANTCU Configuration

- Match the **CAN ID Offset**s in both CANTCU and MaxxECU configurations



Available Realtime-values in CANTCU (sent from MaxxECU)

- *Engine RPM*
- *TPS Value*
- *Engine MAP*
- *Wheel Speeds*
- *Brake Switch*
- *Coolant Temperature*
- *Engine Oil Tempetature*

Available Realtime-values in MaxxECU (sent from CANTCU)

- *VSS Gear*
- *Gearbox Mode*
- *Gearbox Oil Temp*
- *Gearbox Calc Engine Torque*
- *Gearbox Engine Target Torque*
- *Shiftcut 0/1*
- *Blip 0/1*
- *Clutch Slip %*

Extra values sent on the CAN-bus from CANTCU to aid configuration:

Rate	CAN ID	Type	Offset (bits)	Realtime variable	Unit	Factor
50 Hz	0x450	Bit	0	TCU Cut 0/1		1
50 Hz	0x450	Bit	1	TCU Blip 0/1		1
50 Hz	0x456	UINT16	0	TCU RPM Target	RPM	1
50 Hz	0x456	UINT8	16	TCU Cut %	%	1
50 Hz	0x456	UINT8	24	TCU Blip %	%	1
50 Hz	0x456	INT16	32	TCU Delta TQ	Nm	1
50 Hz	0x456	UINT16	48	TCU Delta RPM	RPM	1
10 Hz	0x457	UINT8	0	TCU Auto/Manual		1
10 Hz	0x457	UINT8	8	TCU DL/DriveMode		1
10 Hz	0x457		16-63	RESERVED		

CAN-ID 0x456h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 7	TCU Delta RPM							
Byte 6								
Byte 5	TCU Delta TQ							
Byte 4								
Byte 3	TCU Blip %							
Byte 2	TCU Cut %							
Byte 1	TCU RPM Target							
Byte 0								

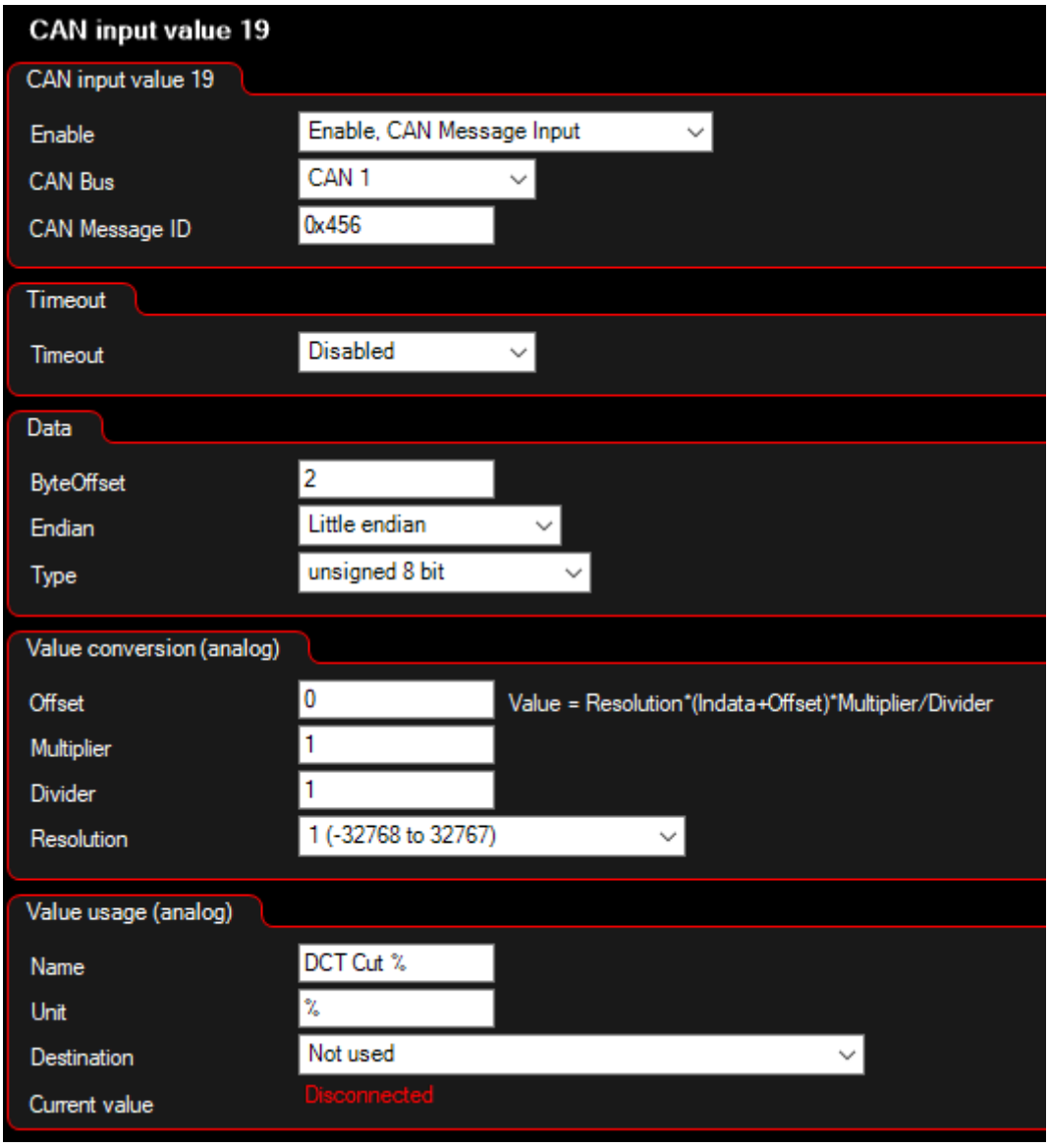
CAN-ID 0x457h

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 7	-	-	-	-	-	-	-	-
Byte 6	-	-	-	-	-	-	-	-
Byte 5	-	-	-	-	-	-	-	-
Byte 4	-	-	-	-	-	-	-	-
Byte 3	-	-	-	-	-	-	-	-
Byte 2	-	-	-	-	-	-	-	-
Byte 1	TCU DL/DriveMode							
Byte 0	TCU Auto/Manual							

Configuring Extra Values (CAN Inputs)

Extra values sent from CANTCU can be imported into MaxxECU by using the **CAN Inputs**. All the parameters needed for successful configuration of the extra values can be found in the tables in this document.

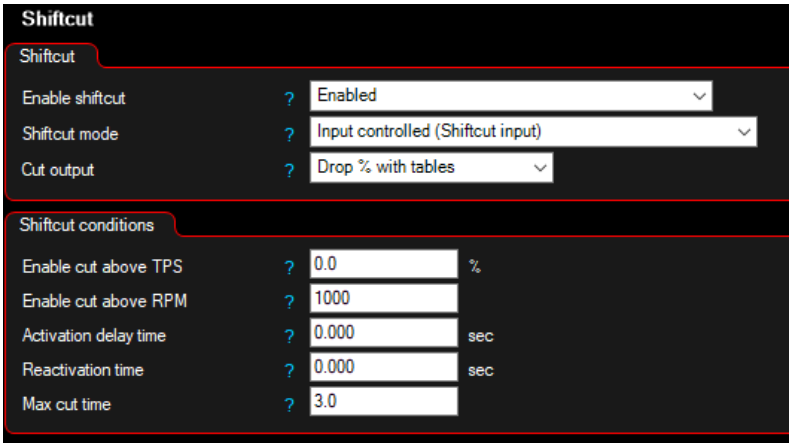
An example of configuring the extra value **Cut %** on a **CAN Input**:



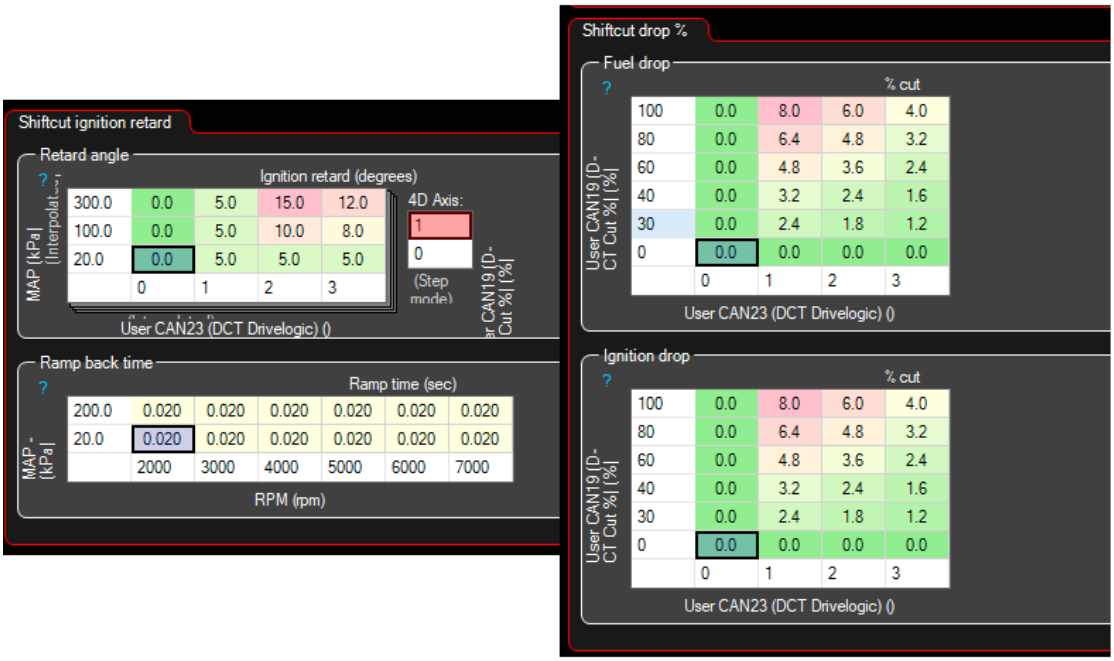
Tuning - Shiftcut

There are several ways of achieving a proper shiftcut. By default, CANTCU sends a shiftcut request (0/1) to MaxxECU **Shiftcut Input**. This request can be combined with the various extra values available like **Cut %**, **Delta TQ**, **Delta RPM**, **Drive Mode** etc. to get a nice and versatile cut that works optimally in every situation.

- MaxxECU **Shiftcut Mode** must be configured as "*Input Controlled*" to let CANTCU decide the duration of the cuts.
- Cut should be enabled also without throttle input and without any delays



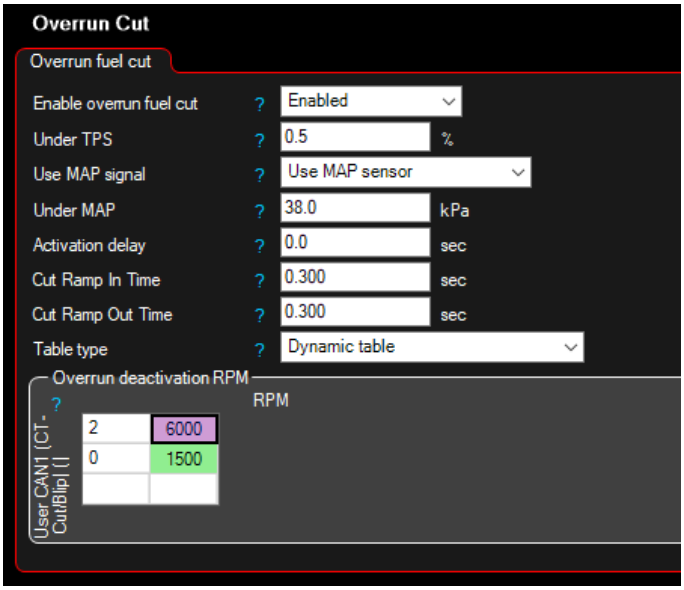
Example configuration using F-Series DCT with cuts based on **Cut %** and **DriveLogic Level** from **CAN Inputs**



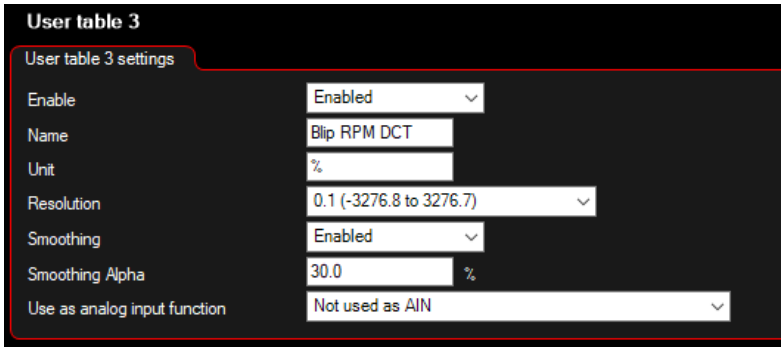
Tuning - Blips

There are many strategies available for blipping the throttle. The best way to achieve sophisticated and optimally targeted blips is to continuously vary the throttle position throughout the blip process. This functionality can be achieved in MaxxECU by using a custom blip process instead of the default built-in function (Motorsport->Throttle Blip). For the custom blip process, a **User Table** and some extra values from CANTCU assigned to **CAN inputs** are needed.

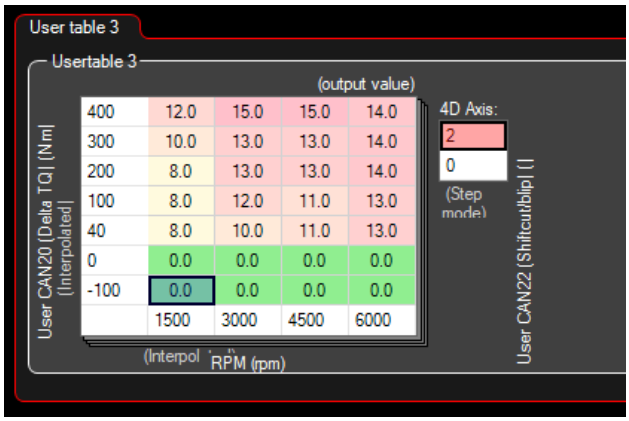
To be able to do blips without the MaxxECU default blip function, the **Overrun Cut** (if enabled) needs to be configured to deactivate overrun fuel cut during CANTCU blip requests. This can be done by adding a table to the function, and defining the **Overrun deactivation RPM** to be higher than the maximum blip RPM (e.g. 6000RPM) when CANTCU blip request (CAN input value=2) is active. The other parameters like **Activation delay** and **Cut Ramp In/Out Times** may need some tweaking, but a configuration similar to the one below should provide a good starting point.



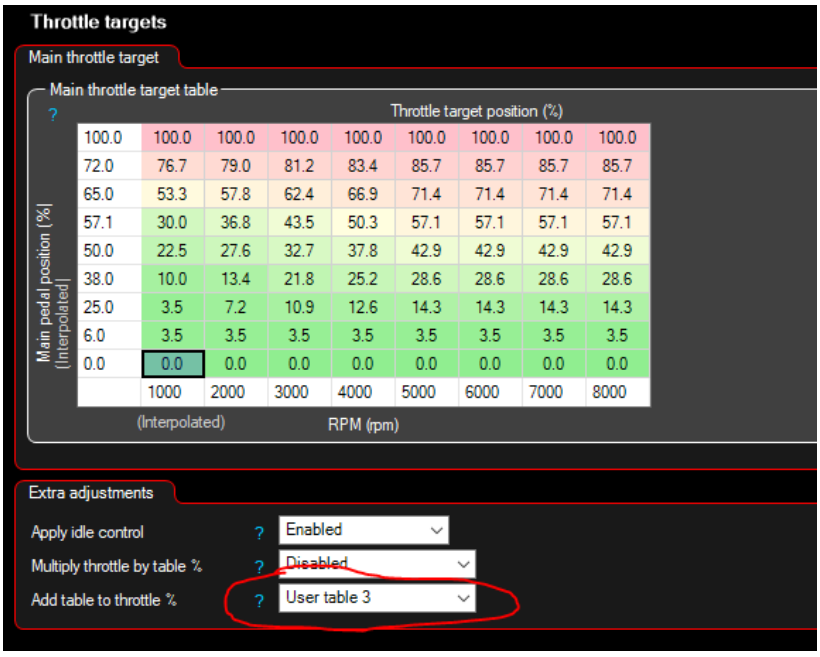
A free **User Table** needs to be enabled. The **User Table** settings can be configured freely. Using smoothing can help calm down throttle control oscillations if needed.



The **User Table** output is a percentage added to throttle position depending on various user-definable inputs. **Delta TQ** or **Delta RPM** combined with **Engine RPM** have been proven to work nicely, but basically any combination can be used, thanks to the freely configurable axes.



To activate the blip function, the **User Table** created earlier is added to **Main Throttle Target** (E-Throttle -> Throttle Targets).



NOTE!

All tuning should always be done by a professional in safe environment (track/dyno)

Before activating the blip function, it's recommended to test downshifting and verify (realtime or logging) that the user table is behaving correctly during the blip. Starting values for tuning the blip should be low and gradually increased to avoid overrevving and undesired behavior/acceleration during the shift.