

Document Number		CTN0038					
Title		Batch Fire Fuel Injection and Serial Communications Updates					
Revision Date		Prepared	Checked	Approved	Change History		
V1	23/07/2021	JG			Initial Release		
V2	15/10/2021	AW			update Bosch module version		

CONTENT

1	I INTRODUCTION	2
	1.1 Purpose	2
	1.2 Supported Packages	2
2	2 Batch Fire Fuel Injection	2
	2.1 Operation	3
3	3 Serial Communications	4
	3.1 Operation	4
4	1 Updating Firmware Project	5
	4.1 Custom Serial Interface	10
	4.2 Serial Group moved to be device specific	10
	4.3 Serial Interface class	10
	4.4 General Serial Port Use	11
	4.5 Configuration of the Serial Interface class for M130 or M122 use	12
	4.6 Unsupported Hardware	13

1 INTRODUCTION

1.1 Purpose

From M1 System Version 01.04.00.0091, two new pieces of functionality are included:

- 1. Batch Fire Fuel Injection.
- 2. Updated Serial Communications which allows M130 ECUs (from hardware revision Q and above) and M122 ECUs to transmit and receive serial communications.

1.2 Supported Packages

To enable these features the system version of the package must be V01.04.00.0091 or later. To check this:

- Open M1 Tune
- Open the desired package
- Select File > Override ECU System Version

Override ECU Sys	Override ECU System Version							
Specify the re	Specify the required ECU System Version for this Package							
Minimum Version 01.04.00.0091								
Required Version	Required Version Most Recent							
MoTeC strongly recommends setting the 'Required Version' to 'Most Recent' to ensure your ECU System is always up to date								

If your system version is less than V01.04.0091, please contact your package developer. Their details can be found in the summary section of the Firmware Help.

Developers who want to update their packages to take advantage of this new system can refer to the section <u>Updating Firmware Project</u> in this document.

2 Batch Fire Fuel Injection

Prior to M1 System Version 01.04.00.0091 batch fire ignition for a distributor setup was only supported on multi tooth reference modes with no camshaft synchronization. And fuel injection was only enabled once the crankshaft reference and camshaft synchronization patterns were both locked (i.e Engine Speed Reference States is Cycle Lock).

In this M1 System Version 01.04.00.0091 update, two new Engine Speed Reference States have been added; **Revolution Lock** and **Cylinder Lock**. This enables fuel injection and ignition when only the reference pattern is locked.

This allows for:

- faster starting with all Reference Modes
- starting the engine without a synchronization sensor on certain Reference Modes
- starting the engine with a faulty synchronization sensor on certain Reference Modes

Revolution Lock allows for wasted spark ignition and batch fire fuel injection. For this state, the reference pattern must only repeat once per crankshaft revolution (i.e. missing tooth reference modes). In this state

while the fuel injection is synchronously timed to the crankshaft, the timing could be offset by 360 degrees in a 4-stroke cycle engine.

Cylinder Lock allows for batch fire injection along with ignition via a distributor. For this state the reference pattern must only repeat once per cylinder (i.e multi tooth ref mode where the number of reference trigger teeth over a full engine cycle is the same as the number of cylinders) and all cylinders share the same ignition resource. While the fuel injection is synchronously timed to the crankshaft the timing could be offset by a multiple of the Cycle duration / The Number of Cylinders.

This batch fire feature should not be used on direct injected engines as the offset fuel timing will lead to irregular behaviour.

2.1 Operation

This function can be enabled or disabled by setting the Engine Speed Reference Start Mode to Batch Fire. Setting this mode to Sequential will only allow fuel injection and ignition when both the crankshaft reference pattern and camshaft synchronization pattern are locked (i.e Engine Speed Reference States is Cycle Lock).

When Engine Speed Reference Start Mode is set to Batch Fire and provided that the Engine Speed Reference State will go to Revolution Lock or Cylinder Lock once the reference pattern has been locked, enabling batch fire fuel injection and ignition. If the synchronisation pattern then locks to the reference pattern the Engine Speed Reference, based on the engine cycle as well as the crankshaft position.

The image below shows a four stroke 36-2 ref pattern that is Revolution Locked. The fuel is injected once per cycle (720 degrees) while the ignition outputs are fired every half cycle (360 degrees). While the ignition timing occurs at the user specified angle the Fuel Timing could be offset by half a cycle (360 degrees in this case).



3 Serial Communications

The hardware and firmware for the M150, M190, M14x, M18x ECUs supports serial communications. The M130 ECU hardware was updated to support serial communications at revision Q.

To check the ECU's hardware revision:

- Connect to the M130 using M1 Tune
- Select Tools > Show Information

General	
Hardware	M130 (Rev Q)
Serial#	5361
Flash Size	128 MB
Flash In Use	53%
RAM Size	1 MB
RAM In Use	72%
Flash Block Size	256 KB
Flash Blocks	512
Rash Blocks Free	238
Rash Blocks Reserved	32

All M122 ECU's support serial communications.

Changes made in system 01.04.00.0091 allow for serial communications to be added to the firmware for M130 (Rev Q+) and the M122 ECUs. This system change affects all hardware variants as it restructures how a serial port is used and configured.

3.1 Operation

The following resources can be used for serial communications. Pin numbering can be found in each specific hardware's documentation.

ECU	Transmit	Receive
M130 (rev Q+), M122	INJ_LS1, INJ_LS2	UDIG6, UDIG7
M150, M190, M14x, M18x	RS232_TX, LIN	RS232_RX, LIN

The M130 (rev Q+) and M122 serial ports are now multifunction pins. If these are selected to be a serial port pin then they're not available for use as a Udig or LS Out. The pins are removed from the resource list for other Udig or LS Out resources. The wiring list is also populated showing the pins as serial ports.

Note that the Serial Port settings and channels (i.e. Baud Rate, Diagnostic, etc.) now reside in the individual groups (i.e GPS, Bosch Wiper, etc.). The Baud Rate and Serial Port Resource need to be configured. Once configured the serial communications will operate as normal.

4 Updating Firmware Project

Follow these steps to update your M1 Build project so that Batch Fire Fuel Injection can be used in the output Package.

- 1. Open M1 Build
- 2. Select Help > Check for Updates

GPR (M150)		
File Build Window He	elp	
🗈 👍 🔒 👪 🖬	Check for Updates	
Tools & Help	Repair	🚻 GPR (M150) - GP Next - Ops 1
🏣 🕶 🔚 🔒 🕶 👎	Send Feedback	Settings 🔒 Modules 🤇
Name	Send Error Report	T
🗄 - 🔒 Anti Lag	Activate Features	Version 01
⊞ : 🔒 Bosch OE Integrat	About MoTeC M1 Build	Ma
	Licence agreement	
⊞ - 📑 Comms	Release Notes	
E · 📑 Cruise	Manuals >	T
🗈 🔒 Event 👘		Hardware: M15

- 3. Download and install the latest system and modules
- 4. Open the desired project
- 5. Select the Settings tab and in the System section, change Version to 01.04.00.0091 (or later).



6. Select the Modules tab

7. To update the **MoTeC Fuel** module (if in use), right-click on the module and select **Version**

🚰 Settings 🔒 Mo	dules 💊 Data Types 🚠 Objects 🛐	Schedule	Diagnostics	🐳 Security	■\$ I/O	DBC	
🗄 🔒 👌 😭	💱 🗙 🖙 💮 🛛 Filter :						
Name	^					Vernier	Duild
	- K					01.00	Build
Bosch UE Integ	ration					01.00	0029
I Motec ADR	Deserver					01.05	0001
Huth MaTaC Anti-Lac						01.05	0004
Huthe Motec And Lag	J					01.05	0007
H MoTeC Control						01.10	0002
H . A MoTeC Contant	Fan					01.02	0007
E S MoTeC Cruise						01.05	0024
MoTeC E8XX						01.02	0000
MoTeC Fr=			1			01.17	0028
🏠 MoTeC G 🖆	Version					01.07	0014
🗉 🚯 MoTeC lg 👡						01.10	0002
🗄 📸 MoTeC In 🔨	Remove		1			01.10	0025
🗄 📸 MoTeC Ki 👝	Move Up		1			01.00	0008
🗄 🚯 MoTeC L' 🚊	Move Down		1			01.09	0002
🗉 🔒 MoTeC La	Move bown		1			01.09	8000
🗄 🖓 MoTeC O	Select All	Ctrl+A	1			01.09	0008
⊞ · 🏠 MoTeC P	Invert Selection		1			01.04	0007
⊞ 🖓 MoTeC S	Class Calastian		1			01.09	0000
B MoTeC T	Clear Selection		1			01.06	0000
HoteC V	Columns		1			01.06	0001
H MoTeC W						01.04	0012
⊞	Expand					01.03	0016
	Expand all						
	Expand/Collapse all to same level						
	Collapse						
	Collapse all others						
			1				

8. Set the Version and Build Number to V1.18.0001 (or later).

'MoTeC Fuel' module properties	×
Select Version :	01.18 ~
Build Number :	
O Always use most recent build	
• Use specific build :	0000 ~
ОК	Cancel

- 9. Repeat steps 7 and 8 to update the following modules (if in use)
 - a. MoTeC GPS module to V1.07.0021 (or later).
 - b. Bosch OE Integration module to V1.00.0031 (or later).
 - c. MoTeC Input module to V1.10.0036 (or later). Note: depending on the previous version that was in use this may require additional changes to the MoTeC Input class instances that are used throughout the project. See MoTeC Input module help for more information on the changes between versions.
 - d. **MoTeC Comms** module to V1.10.0003 (or later).

10. Select the Objects tab

Select Engine Speed Reference Start Mode

Set the **Input Object** to either a **Parameter** (so that it can be set by the user in M1 Tune), or a **Constant** with the value Batch Fire. Refer to the enumeration help for details on each start mode.

🚰 Settings 🔒 Modules 💊 Data T	ypes 🚠 Objects 🛐 Schedule 🔤 Diag	nostics ᆛ Security 📧 I/O 🗉	Engine.Speed.	Reference.Start Mode	×
1 - 🗄 🔗 🍕 - 🕲 🖻	🕺 🗈 🖻 🗙 🏚 合 은 😫	😤 Filter:	T	Identification	
			News	Part Made	
Name	Class	Data Type	Inditie	Julian mode	•
	E8XX		Class	+E Value Input	
	Group				
🖃 🚼 Engine	Group		Include Conditions		S 🖬
🗸 🗸 State	Channel	Engine State Enumeration	Old Name		
🗠 🗸 Stall Time	Channel	Floating Point	Oid Name		
🗠 🔷 Crank Time	Channel	Floating Point	v	Input	
- ~ Trip Time	Channel	Floating Point	Input Object	🚥 🔻 Parameter	
🗷 🗄 Run Time	Group	Floating Point		Makin	
- 🚥 Run Threshold	Parameter	Floating Point		value	
Displacement	Parameter	Floating Point	Data Type	The Reference Start Mode	▼ 🕲 🔲
 Cylinders 	Parameter	Unsigned Integer		and Unitiese	
	Channel	Integer	Guanuty		
E Charge	Group		. .	Display	
⊞ Es Load	Group	Floating Point	Unit	none	- 🕄 🔲
E Es Speed	Group	Hoating Point			
Heterence	Reference		Format	Default	V III
Hesource	IO Resource Input			0	
A Engine Speed	Channel	Hoating Point		U	▼ ■
- A Instantaneous	Channel	Floating Point	. ▲	Validation	
Cycle Position	Channel	Floating Point	Type	None	V III
Task Dask	Channel	Ploating Point			
Tooth Flich	Channel	Posting Point	Minimum		V II
Tooth Index	Channel	Integer			
- Or State	Channel	Pefermana State			L
Diagonatic	Channel	Reference Diagnostic	T.	Attributes	
- Node	Value Input	Reference Mode	Taos	Engine, Setup, Input	- 🛞 🔒
-+El Start Mode	Value Input	Reference Start Mode	L	,= · · ·	
- + Tooth Count	Value Input	Unsigned Integer			
- + Cvlinders	Value Input	Unsigned Integer			
- + 🕅 Offset	Value Input	Floating Point			
→ Blank Ratio	Value Input	Floating Point			
+ Wide Pitch Thresh	. Value Input	Floating Point			
+⊠ Narrow Pitch Thre	Value Input	Floating Point			
- + Maximum	Value Input	Floating Point			
- + Test Speed	Value Input	Floating Point			
- 📆 Update	Scheduled Method				
- 📆 GetDiagnostic	Scheduled Method				

11. Select the Engine Calculation Scheduled Function and add the following highlighted code



12. Select the Warning Update Scheduled Function and add the following highlighted code

```
420 /*
421
    * Remains: Ref/Sync Diagnostics, GPS
422 */
423 else if (Mode.Engine Speed Reference Diagnostic eq Mode Enumeration.Enabled and
424
       Engine.Speed.Reference.Diagnostic.AsInteger() > 1)
425 {
426
        s = Source.Engine Speed Reference Diagnostic;
427
        m = Engine.Speed.Reference.Diagnostic.AsString();
428 }
429 else if (Mode.GPS Diagnostic eq Mode Enumeration.Enabled and
        Root.GPS.Diagnostic.AsInteger() < -1)
430
431 {
432
433
        s = Source.GPS Diagnostic;
        m = Root.GPS.Diagnostic.AsString();
434 }
435 else if (Mode.Engine Speed Reference State eq Mode Enumeration.Enabled and
436
        Engine.Speed.Reference.State neq Engine.Speed.Reference.State.Stall and
        Engine.Speed.Reference.State neq Engine.Speed.Reference.State.Cycle Lock and
Engine.Speed.Reference.State neq Engine.Speed.Reference.State.Revolution Lock and
437
438
      Engine.Speed.Reference.State neq Engine.Speed.Reference.State.Cylinder Lock)
439
440 {
441
        s = Source.Engine Speed Reference State;
442
        m = Engine.Speed.Reference.State.AsString();
443 ]
444 else if (Mode.Vehicle Acceleration Lateral Sensor Diagnostic eq Mode Enumeration.Enabled and
445
        Vehicle.Acceleration.Lateral.Sensor.Diagnostic.AsInteger() < 0)
446 {
<
                                                                                                                           3
📩 Warning.Update *
                                                                                                                         4 \triangleright \times
```

13. If Alternative Fuel is available

a. Select the **Alternative Fuel Calculation Scheduled Function** and remove the following highlighted code.



b. Select the **Engine Calculation Scheduled Function** and add the following highlighted code:



14. Delete any existing Serial Port classes (e.g. RS232, LIN)

	Settings 🔒 Modules 💊 Data Ty	/pes 🚠 Objects	3	Schedule Main Dia	agnostics 😲 Security	1/0	
	🔸 📰 🗞 🧏 🖌 👩 📃	🕺 🖻 🖻 🗙	(a]e	습 준 🚼	😪 Filter :		
Ful	Name	Class	^		Data Type		Quan
	LIN	Serial Port					
2	RS232	Serial Port					
R,	Boost Servo.Actuator	Servo Motor		Convert to			>
	Fuel.Volume.Cylinder 1	Single Injector Fu	v	C		out. V	
R.,	Fuel.Volume.Cylinder 2	Single Injector Fu	ð	Cut		Ctri+X	
R.,	Fuel.Volume.Cylinder 3	Single Injector Fu		Сору	(Ctrl+C	
R.,	Fuel.Volume.Cylinder 4	Single Injector Fu	12	Paste	(Ctrl+V	
24	Fuel.Volume.Cylinder 5	Single Injector Fu		Dacte Special			
R.,	Fuel.Volume.Cylinder 6	Single Injector Fu		Paste Special			
	Fuel.Volume.Cylinder 7	Single Injector Fu		Copy Settings	Ctrl+	Alt+C	
R.,	Fuel.Volume.Cylinder 8	Single Injector Fu		Dacte Settings	Ctrl+	Λl+⊥V	
	Fuel.Volume.Cylinder 9	Single Injector Fu		Paste Settings	Cult	AILTV	
	Fuel.Volume.Cylinder 10	Single Injector Fu	×	Delete		Delete	
	Fuel.Volume.Cylinder 11	Single Injector Fu	aïe	Rename		F2	
	Fuel.Volume.Cylinder 12	Single Injector Fu	afe	Kendine		12	

15. Select the Warning Update Scheduled Function and replace the following highlighted code:



with this highlighted code:



Note: If the above code does not exist a **Warning Mode RS232 Diagnostic** parameter and **Warning Source** enumerator will also need to be added.

- 16. To update any custom serial interfaces refer to the Custom Serial Interface section.
- 17. Validate and Build firmware

Once complete the package can then be used or uploaded to MoTeC online via the normal processes.

4.1 Custom Serial Interface

If the M1 Build project contains any other Custom Serial Interface, this section will provide additional information required to update this.

In system 91 the following library functions were removed from the SDK:

- Serial.PortDiagnostic()
- Serial.PortInit()
- RS232Comms.Diagnostic()
- RS232Comms.Init()

While the following hardware classes were added to the SDK:

- Hardware Digital Output Serial Port
- Hardware Digital Output LIN Bus
- Hardware Configuration Serial Interface

4.2 Serial Group moved to be device specific

Serial ports are typically dedicated to a single task. In the Build object tree, the serial port is now grouped with the subsystem that is using it (rather than placed in a separate RS232 group). The baud rate and diagnostics are configured in this subsystem. For example, the GPS baud rate is now set in the GPS group in the new module. The subsystem is now also included in the name, for example GPS Serial Port Baud, GPS Serial Port Diagnostic, etc.

For multiple serial devices in a project, multiple Serial Port classes can be added.

When the Port Resource is selected for one serial device, it blocks out the port for any other device like any other pin resource.

□·· [™] _E Some LIN Device	Group
- 🔂 Update	Scheduled Function
⊡ - E _B Serial	Group
·· 🗐 Port Resource	IO Resource Parameter
·· 💷 Port Baud	Parameter
🖻 🕞 Port	LIN Bus
	Serial Interface
□·· [™] _E Some RS232 Device	Group
- 🔂 Update	Scheduled Function
⊡ - E _B Serial	Group
·· 🕄 Port Resource	IO Resource Parameter
·· 💷 Port Baud	Parameter
🖃 🕞 Port	Serial Port
	Serial Interface

4.3 Serial Interface Class

The M130 ECU (from hardware revision Q+) and the M122 ECU have several options available for port configurations. This needs to be configured in the **Hardware Configuration Serial Interface** class.

The hardware has 2 identical ports with Rx and Tx that can be configured as RS232, Logic (RS232 TTL, inverted polarity) and LIN.

The resource can be set to Serial N (Rx and Tx pins used), Serial N Rx (Rx pin only) and Serial N Tx (Tx pin only) where 'N' is port number 1 or 2.

LIN always uses the both Rx and Tx pins, and requires some external circuitry or an external LIN driver chip to function correctly. Contact MoTeC for LIN on M130 requirements.

IMPORTANT: The Hardware Configuration Serial class should not be used in projects with M150, M190, M14x or M18x hardware as they have dedicated port pins.

4.4 General Serial Port Use

For an RS232 device, add the Serial Port class to the device group and for LIN add the LIN Bus class to the device group.

These only have Resource and Baud Rate value input objects to be configured. Channel feedback written by the class are Diagnostic and Port number.

Properties			Ļ
GPS.Serial.Port			∻
Ŧ	Identification		
Name	Port		
Class	▶ Serial Port		
Include Conditions		\otimes	
Old Name			
•	Hardware		
Resource Object	range varial GPS.Serial.Port Resource	- 🙁	
•	Input		
Baud Input Object	GPS.Serial.Port Baud	- 🙁	
V Defentitivelar	Value	- 0	
Default Value		- 0	
Port Display Minimum	Display	.1	
For Display Minimum			
Port Display Maximum		1	
T	Validation		
Port Validation Type	MinMax	~	-
Port Validation Minimum		-1	
Port Validation Maximum		1	
Baud Validation Type	MinMax	\sim	
Baud Validation Minimum		960	
Baud Validation Maximum		230400	
T	Attributes		
Tags	Vehicle	• 😣	
Resource Tags	Vehicle	• 🕲	
Diagnostic Default Log Rate	Default (Scheduled Rate)	• 😣	
Diagnostic Logging	Default (Log Rate)	- 😣	
Diagnostic Tags	🖄 Vehicle, Diagnostic	- 🙁	
Port Default Log Rate	Efault (Scheduled Rate)	• 🕲	
Port Diagnostic Logging	Default (Log Rate)	• 🕲	
Port Tags	` Vehicle, Advanced	• 😣	
Baud Tags	` Vehicle, Setup	• 😣	
Update Tags	🗎 Vehicle	• 🛞	
Update Allowed Events	🔛 On 10Hz	- ⊗	
Update Event	🖋 Events.On 10Hz	- 🙁	

Serial Port class properties example:

Typically the Resource object will be set to an external Resource Parameter placed at the top of the parent group so that objects beneath are hidden when the resource is Not in Use. The Baud input object is externally linked to a parameter.

Some restrictions now exist with the Baud parameter as the value input cannot be set to or linked to a channel. Only a parameter or constant is valid. This value is used during bootup, before any Build code is executed. This must be an Integer data type. The enumerated selection of Baud rates previously used is no longer supported. It is recommended that the Port Baud value input is linked to an external Parameter so that the help can list the common baud rates to enter. The Port Baud object within the class cannot have its help edited.

Name	Class
	Group
⊡ n E _B Serial	Group
··· 회 Port Resource	IO Resource Parameter
·· 🚥 Port Baud	Parameter
🖃 · 🛌 Port	Serial Port
	Serial Interface

Example object layout:

For M150, M190, M14x or M18x hardware the resource can be configured as a constant if only one serial device exists in the project. This resource must be a parameter if multiple serial devices are supported. The only selection is RS232 (port 0) for the Serial Port class, or LIN (port 1) for the LIN Bus class.

All other library functions for receiving or transmitting data are the same so changes to existing code in this area are not required.

The one optional exception is the Serial.Transmit() and Serial.Receive() functions. These take a port number argument and likely would have been hardcoded to 0 for RS232 and 1 for LIN in an existing project. That is still fine, but the preferred way is now to set the Port argument to the Serial Port/LIN Bus channel that refers to the Port number. This will always be 0 for RS232 and 1 for LIN, but if this same code was used for an M130 the port numbers can be different, so it's good practice to use the Port channel value here. In an M130 you could for example set a LIN device to Serial 1 (port 0 using Udig 6 and LS Out 1) and an RS232 device to Serial 2 (port 1 using Udig 7 and LS Out 2).

Example code:

Library.Serial.Transmit(h, This.Serial.Port, 4); // port based on the selected resource

4.5 Configuration of the Serial Interface class for M130 or M122 use

Typically this class would be grouped alongside the Serial Port/LIN Bus class and named Interface (as in the above example). For the particular application, the configuration objects should be known and so can be set as constants. This means the group will not appear in Tune.

Resource Object must be set as External and linked to the Serial Port/LIN Bus Resource object.

Mode selections are RS232 (RS232 levels), Logic (RS232 TTL levels, inverted polarity to the RS232 setting) and LIN.

Threshold allows the Udig pin voltage trigger level to be set. Typically set to 2 V for all applications.

4.6 Unsupported Hardware

The M130 prior to Rev Q hardware and all M170 ECUs do not support direct RS232 or LIN connection. A package built for M130 Rev Q hardware can still be loaded and used in an older M130, but the serial ports will not function. Likewise if this same code was built for an M170, it will compile but to prevent confusion, the serial port code should be removed from the Build project.

The Serial Port Diagnostic channel will report Hardware Error in this case and be seen as a fault in the Status view in Tune.

Properties		ąΧ	
Some RS232 Device.Se	erial.Interface	× ×	
T	Identification		
Name	Interface		
Class	To Serial Interface		
Include Conditions	8		
Old Name			
Resource Object	Some RS232 Device.Serial.Port.Resource 🔹 🛞		
▼ Input			
Mode Input Object	=X ▼ Constant		
Mode Value	RS232 ~		
Threshold Input Object	=X ▼ Constant		
Threshold Value	2.0 V 2.0 V		
Ŧ	Value		
Default Value	- · · · · · · · · · · · · · · · · · · ·		
▼ Display			
Threshold Display Unit	wolt [V]		
Threshold Display Format	Default ~		
Threshold Display DPS	1		
▼ Validation			
Threshold Validation Type	MinMax ~		
Threshold Validation Minimum	☑ 0.1 V 0.1 V		
Threshold Validation Maximum	☑ 10.0 V 10.0 V		
▼ Attributes			
Tags	- 😒		
Resource Tags	- 8		
Mode Tags	🗈 Setup, Output 💌 😵		
Threshold Tags	🚵 Setup, Output 🔻 😵		

Typical Serial Interface class properties for an RS232 device.