

EFI TECHNOLOGY

PGQ16 / PGV18

Start & Stop (RISS 2.0 / RISS 2.3)

For ASW-NEXT

Short User Manual



October 2023

Important notes:

If engine starts but its speed is limited to 2000 rpm - possibly ECU is VIRGIN.

When ECU is in **VIRGIN** state, engine starts with any transponder of correct type but engine speed is limited to 2000 rpm. Engine does not start without transponder or with transponder of incorrect type.

Please check the number of stored keys by pressing button [Read Identification](#). ECU is **VIRGIN** if KEYS PROGR.: [NO KEYS PROGRAMMED]

Learn keys using [key programming procedure for VIRGIN ECU](#) or write at least 2 keys [manually](#).

PGQ16 uses transponder key of new type.

PGQ16 / PGV18 uses Piaggio key with PCF7936 inside it. **Key of old type with header BE FA (7D 5F) will not work here!**

Below in this user manual you will find information how to program PCF7936 or T5 to make it compatible with PGQ16 / PGV18.

Now software of ASW-NEXT supports:

- *Key data extraction from ECU without erasing existing key values*
- *Read / Write ignition advance and fuel tables for chip tuning*
- *IMMO-OFF / IMMO-ON*

From V1.21 ASW-NEXT software will output troubleshooting information in case of error message **Set REPROG...FAILED** during write key or program maps.

This error takes place when engine is running at the moment of attempt to set REPROG mode or when supply voltage is lower than 10V or higher than 14.4V

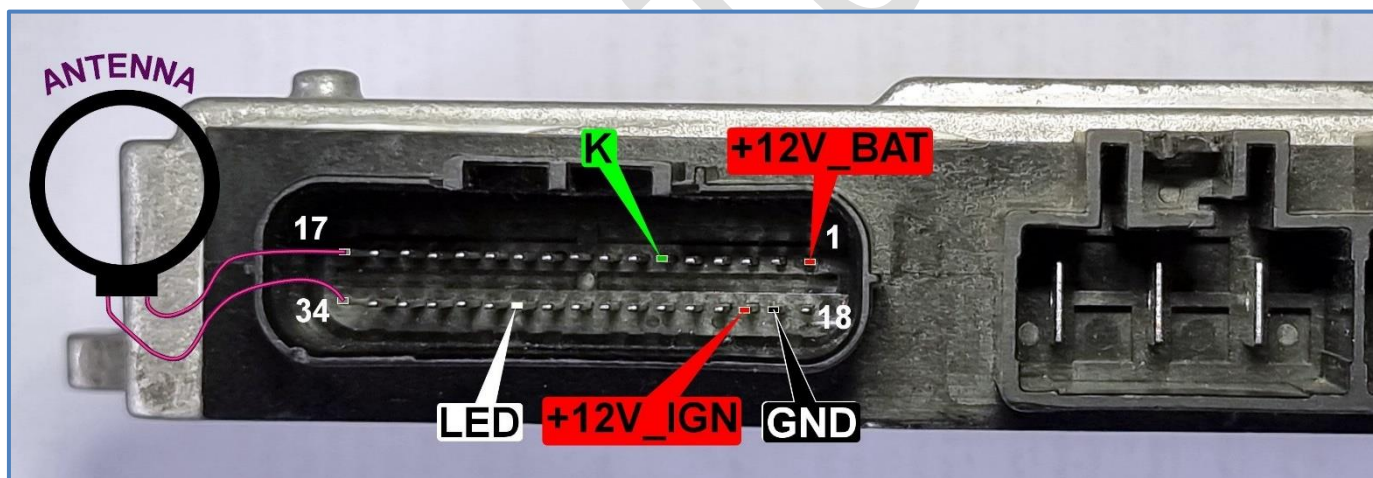
You can suspect internal hardware fault of EFI TECHNOLOGY control unit if ignition voltage displayed in troubleshooting information significantly differ from the real voltage of power supply. Minor difference (+/-0.6V) is allowed.

1. How PGQ16 / PGV18 looks like?



2. How to connect on-the-bench?

PGQ16 and PGV18 have same basic pinout



Antenna and LED are optional components. They should be connected only to test key validity on-the bench.

Place antenna on the ignition lock. Transponder may not be read correctly without ignition lock inside the antenna coil!

NOTE: It is not necessary to connect antenna and LED for key programming or reset-to-VIRGIN procedures to perform.

3. How to connect in the scooter?

- ✓ Connect **RED** and **BLACK** crocodile clips of ASW-NEXT interface to “+” and “-” of scooter battery
- ✓ Connect **GREEN** wire of ASW-NEXT interface (“K”) to the diagnostic connector. Prepare yourself a piece of wire to make a bridge between green wire of ASW-NEXT and the diagnostic connector.



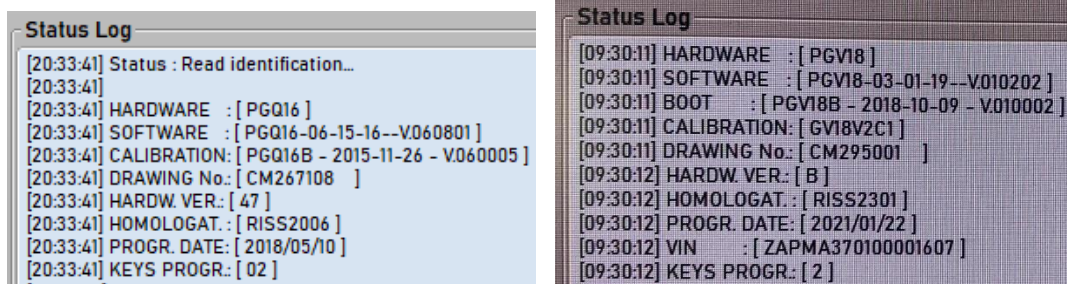
When scooter is equipped with 3-pin connector



When scooter is equipped with EURO-5 connector

4. Software functions

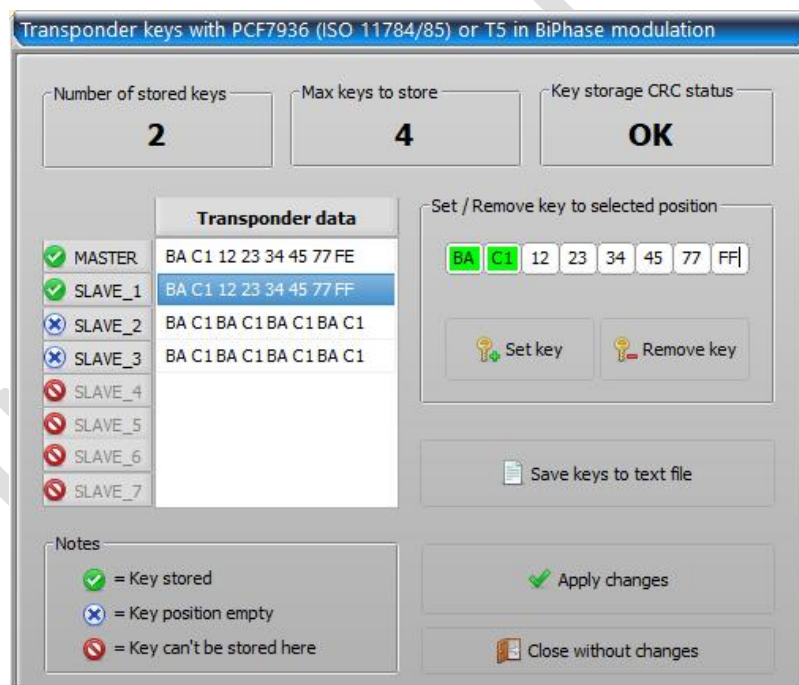
- **Read Identification**



- **Read / Write keys**

To read existing or to store new desired values of transponders use „**Read/Set Keys**“. Values up to 4 keys can be read or set during this procedure. Make a key using [instructions](#) described below and a suitable transponder programmer.

- **Important!** When modifying key storage memory please fill in at least two key positions, **MASTER** and **SLAVE_1** as PGQ16 / PGV18 will treat itself as not completely programmed with only one key stored. To avoid this you must to store one more key, even if single key is needed. Just enter any valid value for a second key position, different from the value of **MASTER** key.



- „**Save keys to text file**“ - text file, containing report with information for keymaking to **PCF7936** and **T5** will be created. File name is built from date of creation. **You can find key values anytime in the text report file.**
- „**Apply changes**“ - to write transponder data to **PGQ16 / PGV18** by K-Line.

- **Reset to VIRGIN**

Erases all keys. Control unit is ready to self-learn new transponder values after Reset-to-VIRGIN.

To reset PGQ16 / PGV18 to VIRGIN:

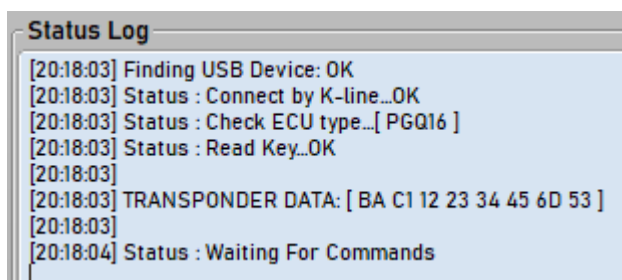
Press button „**Reset to VIRGIN**“

or

Fill all key positions with **BA C1 BA C1 BA C1 BA C1** during procedure „**Write keys**“.

- **Read key in ignition lock**

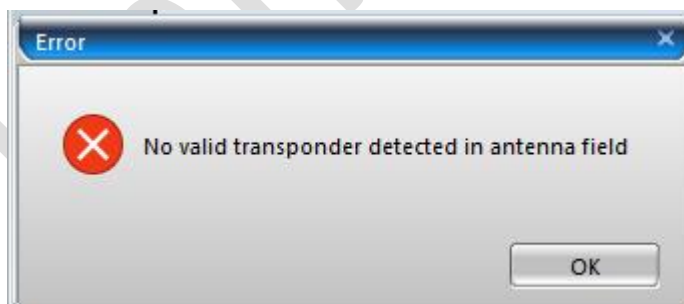
PGQ16 / PGV18 can be used as a transponder reader. Turn on ignition with a key you want to read, after press button „**Read key in ignition lock**“:



NOTE 1: Only key of the new type, based on **PCF7936** or on **T5**, set to **Biphase** encoding, can be read here.

NOTE2: Key is read at the moment when ignition is turned on, not when „**Read key in ignition**“ is pressed.

Error message will take place in case of wrong transponder in antenna field, no transponder or wrong antenna connected to PGQ16 / PGV18:



5. Read data from genuine key using HITAG2 programmer

PGQ16 / PGV18 are designed to accept transponder key that operates in **Biphase** encoding only and it has header „**BA C1**“. Genuine Piaggio key of valid type has **PCF7936** inside. Regular transponder of TYPE-11 in Manchester encoding with header „**BE FA**“ is not accepted by PGQ16 / PGV18 .

To extract transponder data from genuine Piaggio key you need a **HITAG2** transponder programmer **with fast initialization capability** to be able to read **PCF7936** that was previously set to **read-only mode**. Follow these simple steps to read genuine key, based on **PCF7936**:

- 1) Select manually „**HITAG2**“ transponder type in the selection menu of your transponder programmer. **Do not let transponder programmer to detect transponder type automatically!**
- 2) Select „**PCF7936**“ from the list of supported HITAG2 transponders
- 3) Make sure the Password / Encryption Key field is filled with factory default „**4D 49 4B 52 4F 4E**“
- 4) Select „**PASSWORD**“ mode if your programmer has an option to choose between PASSWORD and CRYPTO modes (some HITAG2 programmers do auto-detect encryption mode).
- 5) Read all transponder pages (all at once or one-by-one). Pages **User0-User3** contain information that you need. Transponder data in the example below is **BA C1 12 23 34 45 6D 53**. Transponder data is repeated twice.

The screenshot shows the HITAG2 programmer interface with the following data:

Field	Value
Select Type TP	PCF7936
ID	65 8E 90 10
ISKL	4D 49 4B 52
ISKH	00 18 4F 4E
TMFC	00 AA 48 54
User0	BA C1 12 23
User1	34 45 6D 53
User2	BA C1 12 23
User3	34 45 6D 53
RSKL	
RSKH	
RCF	
Sync	
Encryption Key	4D 49 4B 52 4F 4E
Status	Encoding: Manchester Access: OK! Mode: Password

NOTE: Fail to read data from transponder may mean that your HITAG2 programmer does not support **fast initialization**. Repeat steps 1 to 5 using different HITAG2 programmer.

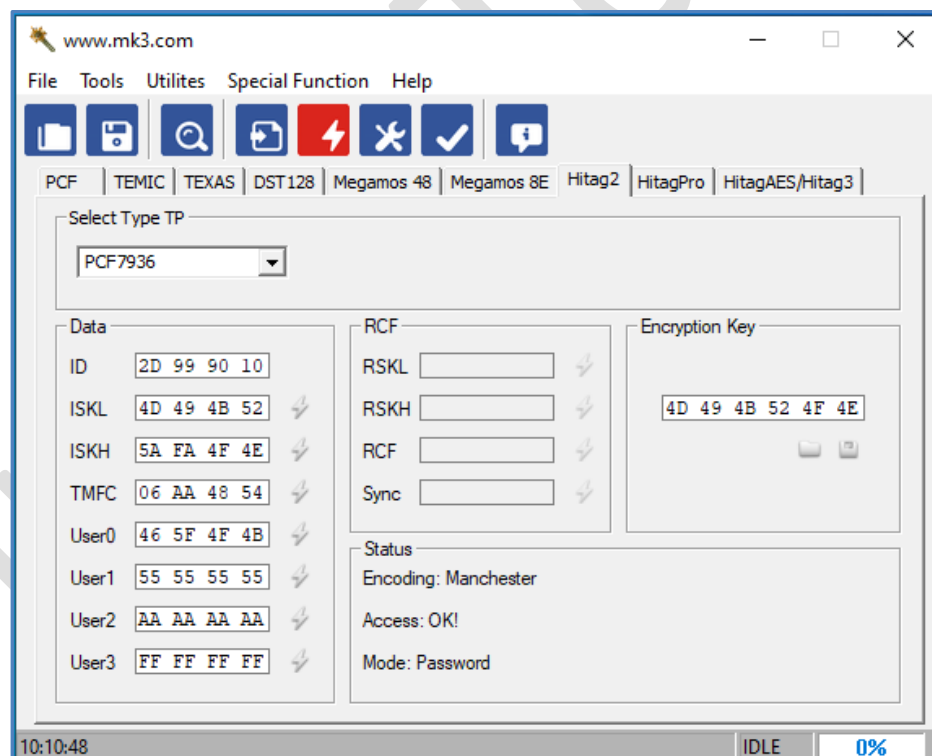
6. Make a key for PGQ16 / PGV18 into transponder PCF7936

NOTE: Use PCF7936 transponder only! HITAG2 emulating transponders (TPX3/TPX4/CN2 or similar) are not accepted here as most of them do not support Read Only Modes!

Take blank **PCF7936** and follow these steps:

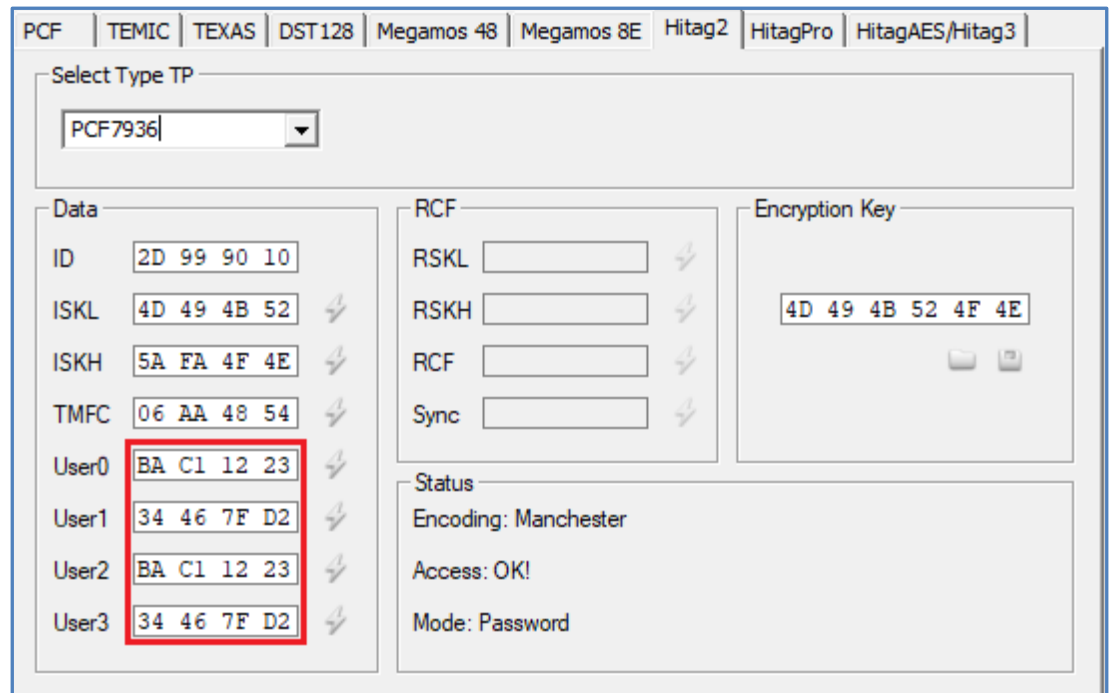
- 1) Select „**HITAG2**“ transponder type in the selection menu of your transponder programmer or let your transponder programmer to detect transponder type automatically.
- 2) Select „**PCF7936**“ from the list of supported HITAG2 transponders
- 3) Make sure the Password / Encryption Key field is filled with factory default „**4D 49 4B 52 4F 4E**“
- 4) Select „**PASSWORD**“ mode if your programmer has an option to choose between PASSWORD and CRYPTO modes (some HITAG2 programmers do auto-detect encryption mode).
- 5) Read all transponder pages (all at once or one-by-one).

Result should look like this:

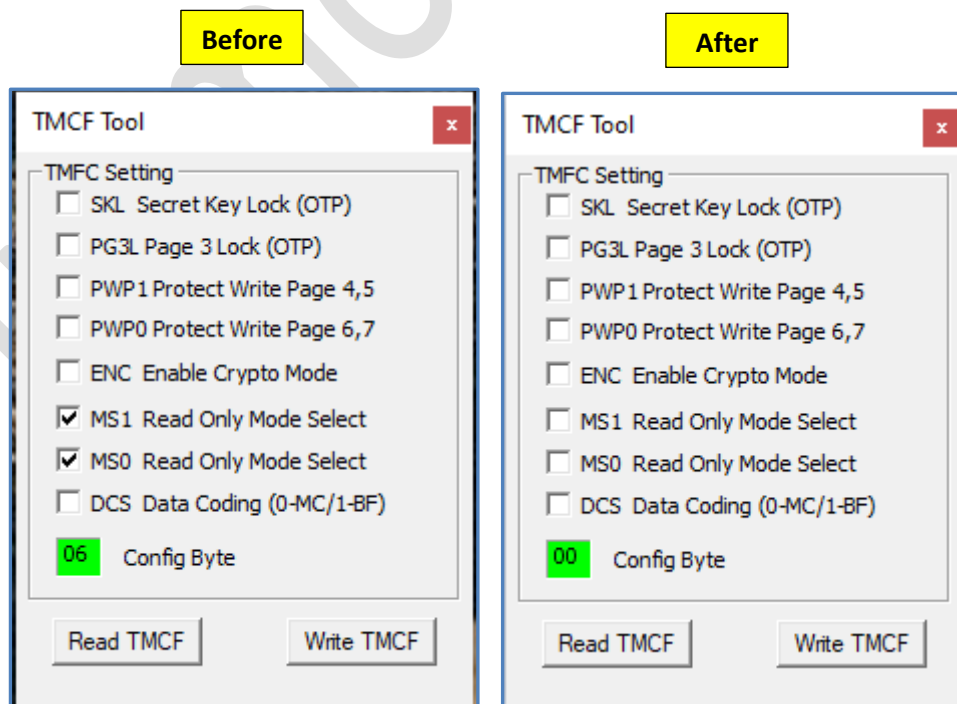


- 6) For example, you need to make a key with value **BA C1 12 23 34 46 7F D2**.
Fill fields **User0** and **User2** with **BA C1 12 23**
Fill fields **User1** and **User3** with **34 46 7F D2**
Make sure User0 = User2 and User1 = User3 !
Make sure the first two bytes of User0 and User2 are equal to BA C1 !

Result should look like this:



- 7) Write data to transponder. You may write only pages **User0-User3** one by one or write all transponder pages at once.
- 8) Go to transponder configuration menu in Hitag2 programmer and **clear all bits**. It may look different for various Hitag2 programmers but main idea is the same. Example illustrates configuration menu in MK3, please find out by yourself how it looks in your particular Hitag2 programmer.



- 9) Write configuration **00** to transponder. From this point your **PCF7936**, configured in such way, will be recognized by universal transponder programmer not as a **PCF7936**, but as **TEMIC11**.

NOTE: Proceed with transponder configuration change to Read Only Mode only after user pages *User0-User3* have been programmed with key data.

In other words, program *User0-User3* right before setting transponder configuration to "00".

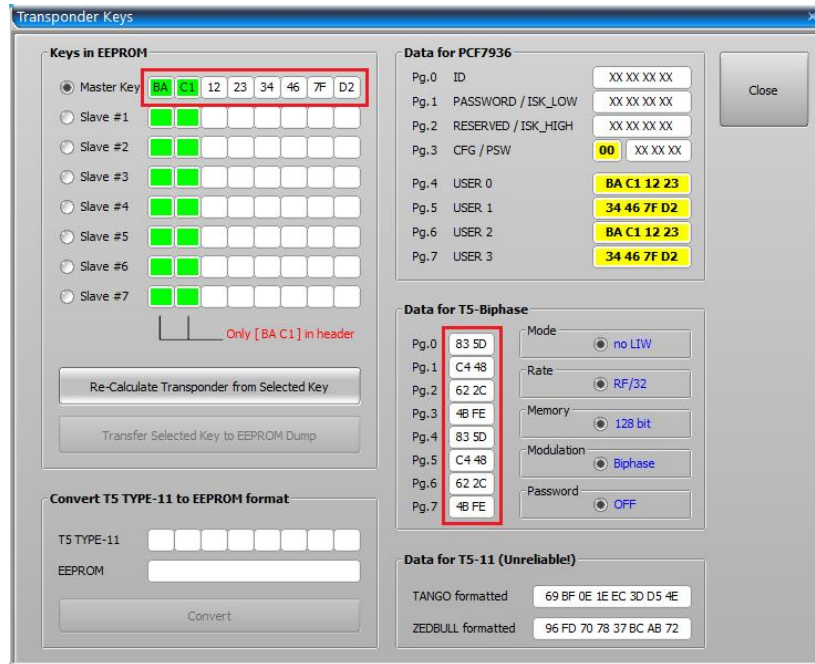
8. Make a key for PGQ16 / PGV18 to transponder T5

PGQ16 can decode transponder data encoded in **BiPhase only**. The fact that regular **T5-11** is encoded in **Manchester** does not allow us to use a basic **T5** programmer. Solution is to program data into **T5** in extended configuration mode where **BiPhase** encoding can be selected. Most of modern transponder programmers have extended configuration for **T5**.

- 1) Open report file that was created during procedure „**Write Keys**“. Here you will find information how to fill data pages 0-7 of transponder **T5** for every key that was programmed to **PGQ16**.

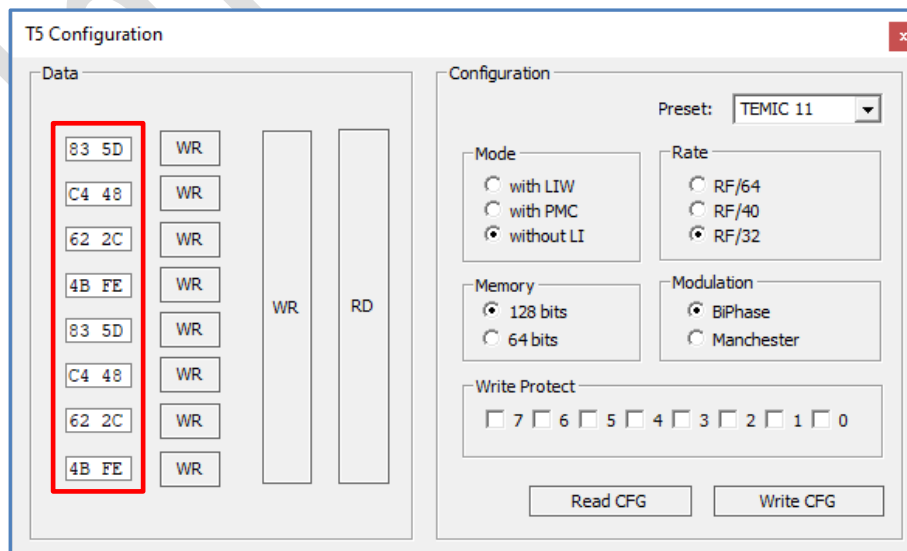
```
-----  
Data for making a transponder  
-----  
[*] MASTER : BA C1 12 23 34 46 7F D2  
  
>> Data for PCF7936 <<  
USER0: BA C1 12 23  
USER1: 34 46 7F D2  
USER2: BA C1 12 23  
USER3: 34 46 7F D2  
  
>> Data for T5 in BiPhase modulation <<  
Pg.0: 83 5D  
Pg.1: C4 48  
Pg.2: 62 2C  
Pg.3: 4B FE  
Pg.4: 83 5D  
Pg.5: C4 48  
Pg.6: 62 2C  
Pg.7: 4B FE  
-----
```

You may also use dialog **Transponders** from **Magneti Marelli MIU G4 (M4G1)** in the software of ASW-NEXT to generate data for **T5**:



- Go to **T5** configuration in your transponder programmer. Please note, that some basic transponder programmers does not allow to configure **T5** at advanced level. Such programmer won't work here – find yourself another programmer.
- Fill all data fields of T5 with data from report file or data, generated by transponder dialog of software for **MIU G4**. Set configuration as in the picture below:

MODE = [Without LIW]
RATE = [RF/32]
MEMORY = [128 bits]
MODULATION = [BiPhase]
WRITE PROTECT = [All off]



- 4) Write all pages and configuration to **T5**. From this point your **T5**, configured in such way, will be recognized by universal transponder programmer as **T5-11**.

9. How to learn keys to VIRGIN PGQ16 / PGV18

According to transponder making instruction above make yourself two valid transponders (**PCF7936** or **T5**) or take two ready suitable keys. **Both transponders must be programmed with different values.**

- Turn on ignition with first key for ~1.5 sec – LED goes on. Turn off ignition while LED is still on. This key will be stored as a **MASTER**
- Turn on ignition with second key for ~1.5 sec – LED goes on. Turn off ignition while LED is still on. This key will be stored as a **SERVICE**
- Turn on ignition with first key again for ~1.5 sec. to close programming procedure. Turn off ignition while LED is still on.

NOTE: Do not exceed time of 1.5 seconds when changing key in the ignition lock between every step. Turn ignition off while LED is still on!

10. Read / Write MAPS

Software of ASW-NEXT is now able to read and reprogram ignition advance and fuel tables. You may find this useful to improve engine performance or to remove speed limiter.

NOTE: Software of ASW-NEXT by itself does not modify map files to remove any limitations. Please use appropriate software to edit map files, e. g. WinOLS.

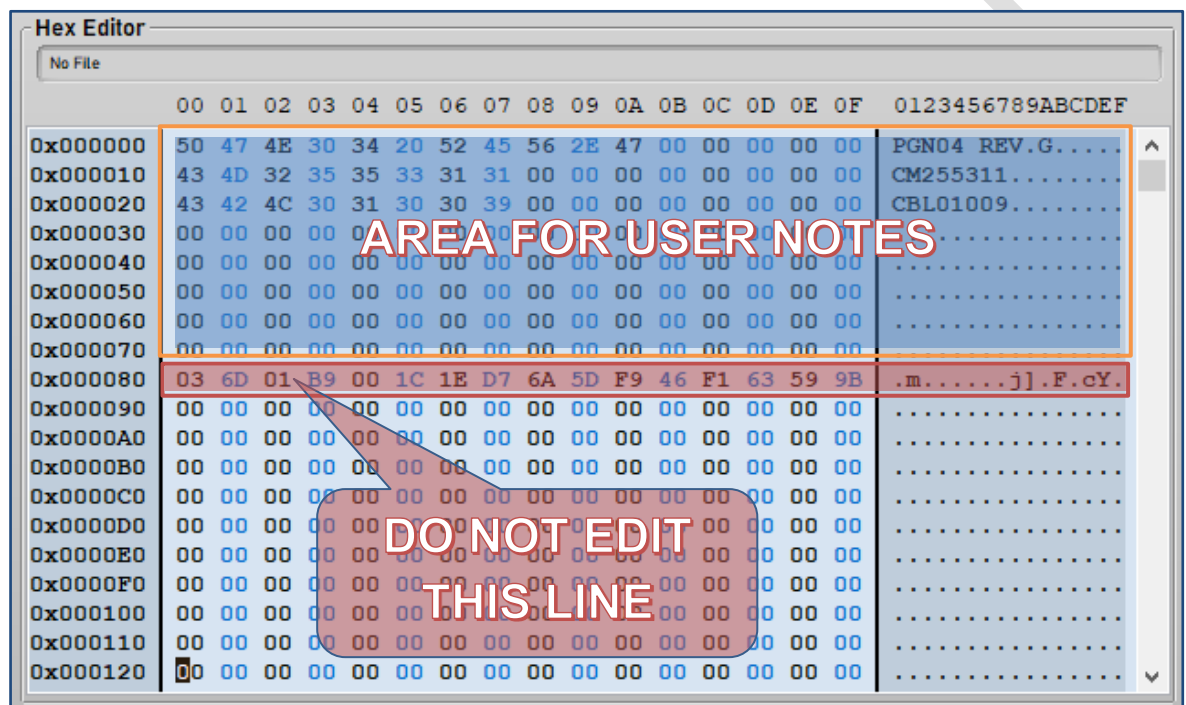
NOTE: Software of ASW-NEXT will recalculate CRCs in map file, modified by third-party software before to program it to engine control unit.

11. MAP file

MAP is saved to binary file. File size for PGQ16 / PGV18 MAP file is 96kb. Ignition advance and fuel tables in binary MAP file are located at their physical addresses in the memory space of STM8.

Upper 8 lines (128 bytes) can be modified by the user without any limitations. This area is reserved to store various text notes about MAP file.

NOTE: Do not edit data at 0080-008F. MAP file will be treated as corrupted once data at line 0080 was modified.



12. IMMO-OFF / IMMO-ON

This function configures immobilizer feature presence. Works by K-Line.

