

# **IQRF Wireless Upload**

**The Lite version**

**Application Note**

**AN009**



Simple way to smarter wireless solutions

IQRF TR modules with operating system v2.11 or higher can be uploaded even in a wireless way (RF PGM). It is advantageous especially for TRs which can not be unplugged from application boards due to housings, soldering etc.

The programmed TR is not inserted in the programmer during the RF upload but it is connected to the programmer in a wireless way by an auxiliary TR module. This application note describes the Lite RF PGM version. Advanced versions with extended features and configuration will be available later.

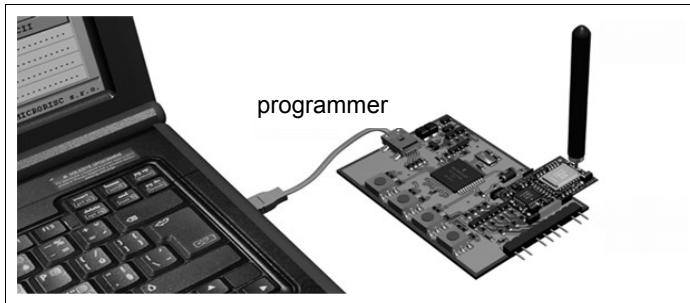


Fig. 1: Standard (wired) upload in a programmer

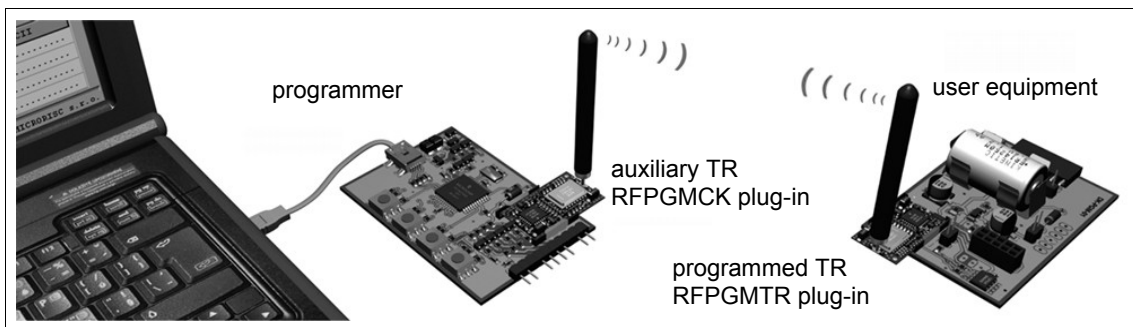


Fig. 2: Wireless upload (RF PGM)

Both TR modules have to have the respective plug-ins (SW options extending the OS) uploaded in advance:

- Plug-in for the auxiliary TR module (RFPGMCK): must be uploaded by the user
- Plug-in for programmed TR module: RFPGMTR
  - For OS v2.11 build 889 and higher: The Lite version is already uploaded.
  - For OS v2.11 build lower than 889:
    - The PATCH-211-RFPGMTR patch plug-in must be uploaded first by the user.
    - Then upload the RFPGMTR plug-in. Avoid uploading both these plug-ins at the same time (even if your programmer supports this).

During wireless upload the programmed TR module is not in the standard programming mode but in the RF PGM one. The standard mode is emulated by the auxiliary TR module in the programmer. From the IQRF IDE point of view all is the same as for standard programming except of the following:

- When using the Lite plug-in version the IDE does not differentiate normal programming mode from RF PGM at all.
- Programmed TR module is switched to the RF PGM mode (under conditions described below) but neither by the **Enter PROG Mode** button nor by the **F5** key (using these would unintentionally switch the auxiliary TR module to standard programming mode) but by the **Upload** button only.
- The *Module Info* window and the *Reset TR Module* button in IQRF IDE does not relate to the programmed TR module.
- It is possible to program more TR modules simultaneously.

There are the following plug-in functions available in the user application (see the Appendix):

- `runRFPGM()` – switch TR to the RF PGM mode
- `enableRFPGM()` – request to enable switching TR to the RF PGM mode after reset
- `disableRFPGM()` – request to disable switching TR to the RF PGM mode after reset

To utilize these functions in user program it is needed to make the `RFPGMTR-LITE.h` header file accessible to the compiler (by the `#include` directive with the path to the file).

`enableRFPGM()` is intended to configure OS so that TR module is switched to RF PGM mode whenever after following resets. `disableRFPGM()` is intended to configure OS so that the TR module is switched to run mode whenever after following resets. Thus, TR module can be set to RF PGM by reset (if enabled) or by the `runRFPGM()` function (unconditionally). Stand-alone TR modules are delivered with this option disabled.

RF PGM mode can be terminated without RF uploading by the low level voltage applied to the C5 pin for at least ~0.7 s (e.g. by the pushbutton on the development kit used). Then:

- If RF PGM was invoked by reset based on the `enableRFPGM()` function the original application is run from beginning.
- If RF PGM was invoked by the `runRFPGM()` function the application continues program executing.

**Caution:** Low level voltage applied to the C5 pin for at least ~0.7 s during RF PGM will abort the upload.

## LED indication

Operation	Auxiliary TR		Programmed TR	
	Green	Red	Green	Red
<b>RF PGM mode</b>	short flash in 1 s period	–	continuous	short flashes in 2 s period
<b>RF PGM mode finished</b>	–	–	–	single flash for 2 s
<b>Uploading</b>	short flash in 1 s period	fast blinking	continuous	fast flashing

## Common practice for RF PGM

Typical requirement used to be a different behavior during development (when modified program is uploaded repeatedly) and after finishing the development (when following-up reprogramming is not desired at all or is required for service purposes on special request only).

The `enableRFPGM()` is intended typically for development and especially for applications with TR modules inaccessible for wired upload. E.g. an error in user program under development can cause that the `runRFPGM()` function can not be accessed any more. Due to the fact that this TR can not be reprogrammed externally the only way how to force programming mode is to switch to RF PGM automatically after reset. Thus, main purposes of `enableRFPGM()` are:

- To enable entering RF PGM while routine development in a fast and reliable way.
- A loop-hole against user errors during development.

But it is recommended to keep a way of how to invoke RF PGM mode even in completed and debugged designs on request (at least to allow services and future upgrades). That is why `disableRFPGM()` should not be used unless there is a user's defined way of how to invoke back the RF PGM mode via `runRFPGM()` or `enableRFPGM()`. It is up to the user to define the entry point into RF PGM, e.g. using a jumper, pushbutton or a special RF packet. If being omitted there is no any other way of getting back into RF PGM unless removing the module from a device and placing it into the programmer.

Thus, RF upload is fully under the user's control. Invoking the RF PGM mode, its termination, behavior after reset – everything can be adapted according to the user's needs.

**Caution:** The `enableRFPGM()` and the `disableRFPGM()` always modificate the MCU Flash memory. Thus, number of calling these functions is limited due to the Flash endurance (e.g. ~ 10 000 cycles min. for TR-52B). The user should avoid situations when these functions are called too frequently (in repeated cycles, typically after resets e.g. due to the watchdog). It is not a topic for applications like well-programmed remote controllers RC-04 where frequent repetitions are not possible in normal operation and the device is reset after battery change only. See the Example below, points 6 and 11.

**Example:** Wireless upload procedure can be tested using the demo program `RF-PGM-TST`:

1. **Preparatory phase:** Upload the `RFPGMCK` plug-in to the auxiliary TR module (not in a wireless way).
2. For programmed TR with OS v2.11 build 889 or higher go to step 4.
3. Insert the programmed TR to the programmer first and upload (not in a wireless way):
  - The `PATCH-211-RFPGMTR` patch plug-in first.
  - The `RFPGMTR` plug-in then.
4. Insert the programmed TR to the programmer and upload (not in a wireless way) the `RF-PGM-TST` demo program. Then:
  - The application just starts up which is indicated by red LED flashing.
  - But thanks to the `enableRFPGM()` function used in the demo program the programmed TR is configured so that it will be switched to RF PGM after reset from this time.
5. **Development phase:** Arrange both TR modules according to Fig. 2 for RF PGM upload.
6. The `enableRFPGM()` function should be removed now from the program.
7. For test purpose change LED flashing from the red LED to the green one in user program and compile this.
8. Set the programmed TR module in RF PGM mode (e.g. after the TR module reset) and invoke upload in the IDE environment by the *Upload* button (but neither by *F5* nor *Enter programming mode*). After RF PGM finish the red LED flashes for 2 s and then the application starts up automatically. (RF PGM mode can be terminated without RF uploading by the pushbutton on the kit by pressing for ~0.7 s.)
9. Repeat steps 7 and 8 until the application is debugged. Respective LED flashing indicates success.
10. **Final upload:** Upload final application **with** `disableRFPGM()`. After finishing the RF PGM mode is invoked for the last time. Press the pushbutton for more than 0.7 s to terminate this. The application starts to run, executes the `disableRFPGM()` function and automatical entering RF PGM after following resets is disabled.
11. RF PGM is accessible via the pushbutton only since now.
12. Upload final application **without** `disableRFPGM()`.
13. Reset the TR module, leave the RF PGM by the pushbutton for ~0.7 s and the application is ready to be released. Following resets will immediately enter the run mode.
14. **RF PGM upgrades** stay accessible via `runRFPGM()` by the pushbutton later on.

**Note:** Removals of `enableRFPGM()` and `disableRFPGM()` in steps 6 and 11 is due to Flash endurance only.

## Features

- Fixed channel (default one, e.i. 52 for 868 MHz or 104 for 916 MHz bands) and fixed RF bit rate (19.2 kb/s)
- "Non-networking" communication, without addressing and broadcast
- `RFPGMCK`:
  - Not joined to the TR module ID
  - Reprogrammable
- `RFPGMTR`:
  - No configuration available

**Appendix – Plug-in functions**
**enableRFPGM**

<b>Function</b>	Request to configure OS for switching to RF PGM mode after TR module reset
<b>Purpose</b>	Enable switching to RF PGM mode after reset
<b>Syntax</b>	<code>void enableRFPGM()</code>
<b>Parameters</b>	–
<b>Return value</b>	–
<b>Output values</b>	OS configured
<b>Preconditions</b>	–
<b>Remarks</b>	This function must be executed first to modify OS and just the following reset will switch to RF PGM.
<b>Side effects</b>	MCU Flash memory is modified even if RF PGM has already been enabled). Thus, number of calling this function is limited due to the Flash endurance (e.g. ~ 10 000 cycles min. for TR-52B).
<b>See also</b>	<code>disableRFPGM</code> , <code>runRFPGM</code>
<b>Example</b>	See <code>disableRFPGM</code>

**disableRFPGM**

<b>Function</b>	Request to configure OS for not switching to RF PGM mode after TR module reset
<b>Purpose</b>	Disable switching to RF PGM mode after reset
<b>Syntax</b>	<code>void disableRFPGM()</code>
<b>Parameters</b>	–
<b>Return value</b>	–
<b>Output values</b>	OS configured
<b>Preconditions</b>	–
<b>Remarks</b>	This function must be executed first to modify OS and just the following reset will not switch to RF PGM.
<b>Side effects</b>	MCU Flash memory is modified even if RF PGM has already been disabled. Thus, number of calling this function is limited due to the Flash endurance (e.g. ~ 10 000 cycles min. for TR-52B).
<b>See also</b>	<code>enableRFPGM</code>
<b>Example</b>	<pre>enableRFPGM();           // During development // disableRFPGM();      // For final application</pre>

### runRFPGM

<b>Function</b>	Switch to RF PGM mode
<b>Purpose</b>	One-shot immediate switching to RF PGM mode
<b>Syntax</b>	<code>void runRFPGM()</code>
<b>Parameters</b>	–
<b>Return value</b>	–
<b>Output values</b>	RF PGM mode initiated
<b>Preconditions</b>	The Lite RF PGM is not configurable that is why parameters like RF channel, bit rate and filtering are fixed to OS default values. If the application uses different ones, they must be set to default before using <code>runRFPGM</code> and then (just for the case of RF PGM refusal by the C5 pin) restored to desired values.
<b>Remarks</b>	<ul style="list-style-type: none"> <li>• RF PGM mode can be refused by low level on the C5 pin for at least ~0.7 s. Then the application continues without reset.</li> <li>• After successful RF PGM finishing the application is reset.</li> </ul>
<b>Side effects</b>	–
<b>See also</b>	<code>enableRFPGM</code>
<b>Example 1</b>	<code>if (jumperSet) runRFPGM(); // Enter RF PGM mode on special request</code>
<b>Example 2</b>	<pre> setFilteringOff();           // If filtering is used setRFchannel(52);           // If a different channel is used                              // default (52 for 868 MHz or 104 for 916 MHz)  runRFPGM();  setFilteringOn();           // Restoring (for continuing without RF PGM only) setRFchannel(my_channel);   //      "-"</pre>
<b>Example 3</b>	<pre> setFilteringOff(); setRFchannel(52); runRFPGM(); reset();                     // Reset (for continuing without RF PGM only)                              // Restoring is not necessary here</pre>

---

## Sales and Service

---

MICRORISC s.r.o., Delnicka 222, 506 01 Jicin, Czech Republic, EU  
Tel: +420 493 538 125, Fax: +420 493 538 126, [www.microrisc.com](http://www.microrisc.com)

### Partners and distribution

Please visit [www.iqrf.org/partners](http://www.iqrf.org/partners)

---

### Quality management

*ISO 9001 : 2000 certified*

### Trademarks

*The IQRF name and logo are registered trademarks of MICRORISC s.r.o.  
PIC, SPI, Microchip, MPLAB and all other trademarks mentioned herein are property of their respective owners.*

### Legal

*All information contained in this publication is intended through suggestion only and may be superseded by updates without prior notice. No representation or warranty is given and no liability is assumed by MICRORISC s.r.o. with respect to the accuracy or use of such information.*

*Without written permission it is not allowed to copy or reproduce this information, even partially.*

*No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.*

*The IQRF products utilize several patents (CZ, EU, US)*

---

**On-line support: [www.iq-esupport.com](http://www.iq-esupport.com)**

---



Simple way to smarter wireless solutions