Infotainment Business Unit
FreeRTOS PVT Application Example

1 Introduction

Purpose of this document is to provide a PVT example using FreeRTOS to help the first steps in application design.

2 Contents

2.1 Index

2.2 List of Tables

2.3 List of Figures

3 Document Management

3.1 Revision History

Rev	Date	Author	Notes
1.0	15/04/2016	S. Chambrillon	Creation
1.1	22/06/2016	S. Chambrillon	Update of template & compilation switch

Table 1 Revision history

3.2 Acronyms

Keyword	Definition
GNSS	Global Navigation Satellite System – It can include any combination of different satellite constellations like GPS, GLONASS, SBAS etc.
UART	Universal Asynchronous Receiver Transmitter

Table 2 Acronyms

3.3 Reference Documents

Reference	Title	Author	Version
	STA8088/90_Firmware_Configuration	Andrea Di Girolamo	1.1
	STA8088/STA8090 SDK Usage	Fulvio Boggia	1.8

Table 3 References

3.4 Contact info

Keyword	Definition
S. Chambrillon	samuel.chambrillon@st.com

Table 4 Contact name list

4 GNSS FreeRTOS Application Example

4.1 Functional behaviour

This example creates and launches a task which then runs in parallel of other GNSS tasks.

Aim of this task is to display the position, speed and UTC time once GNSS fix has been acquired.

Application messages are transmitted every 2 seconds using UART debug port and are included among other debug messages.

Here is an overview of what could be displayed in debug log when enabling this example:

```
…
[PVT_app]user position -> latitude: 050N24.098
[PVT_app]user position -> longitude: 008E04.27087
[PVT_app]user position -> height: 147.43
[PVT_app]user speed -> 72.540 km/h, 20.150 m/s
[PVT_app] week: 2783 UTC time:  5:22:31
…
```

4.2 Set-up and execution

Note: *Pre-requisite: user needs to have Eclipse already installed and to have opened the `gnssapp_demo_freertos_gae` project in it (refer to STA8088/STA8090 SDK Usage if needed).*

Main example files are located in `/apps/PVT_app` directory: `PVT_app.c` & `PVT_app.h`. And a little piece of code is also located in `sta8090/gpsapp/main_demo_fr.c`.

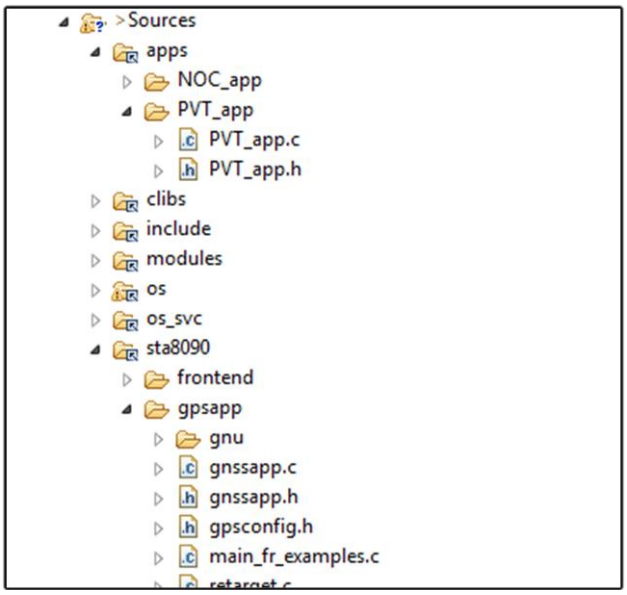

Figure 1: Screenshot of files tree in Eclipse project

Implementation in `PVT_app.c` mainly uses:

- GNSS libraries API
- FreeRTOS or gpOS API (user choice but set by default to gpOS API)
- In `main_demo_fr.c` file: `pvt_app_init()` is called in `main_idle_process()` function. This will launch the PVT example task creation.

Before starting building, make sure that `EXAMPLE_PVT` compilation flag is enabled in `build.mk`.

```
#
# C compiler extended options
#
EXT_CDEFS=EXAMPLE_PVT
```

To use FreeRTOS API instead of gpOS API, add `DEMO_USE_FREERTOS_API` compilation flag in `build.mk`.

```
# C compiler extended options
#
EXT_CDEFS=EXAMPLE_PVT DEMO_USE_FREERTOS_API
```

Now:

- Build the software by selecting the wished target in Eclipse (`SOC SQI(SQI)`)
- Ensure your UART debug port is correctly configured as indicated in 5.1 Enable debug log. Data are sent to Debug Port using `GPS_DEBUG_MSG()` macro.
- Flash generated binary located in `/bin` directory, using either Trace32 or XLoader.
- Then execute the software and look at debug log. User should observe same kind of display than in 4.1 chapter.

5 Miscellaneous functionalities

5.1 Enable debug log

Referring to STA8088/90_Firmware_Configuration.pdf document, two parameters are used to enable the debug port:

- ID 100: Debug port number (where value 0...2 correspond to debug UART port number)
- ID 103: GPS Debug Mode (for which the value must be set to 0 to enable the Debug Mode)

So, as an example, if user wants to enable Debug Port on UART0, it needs to create the following configuration file:

`DebugCfg.txt`

```
100 -> 0
103 -> 0
```

And apply it to its binary file using this command line (generated binary file is located in `/bin` directory):

```
FWConfig.exe –f <generated *.bin file> -c DebugCfg.txt –o
<output_image_file>
```

Where `output_image_file` = the binary image which includes the new configuration

Another method to update the firmware configuration is to use the method described in STA8088/STA8090 SDK usage: write the parameters values you need to change in a customized dedicated `fwcfg.txt` file so that they will be taken into account during post build.

Important notice: the example above is only given as an example and user must refer to STA8088/90_Firmware_Configuration.pdf document to be certain of ID numbers and command line parameters to use.

6 Disclaimer

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.