M8T UoA CORS

User manual

Metadata

Author	Nathanael Esnault
Version	1.0.1
Verified	29/05/2018

Revisions

Date	Version
23/05/18	0.1
28/05/18	1.0.1

Modifications

Date	Author	Approved by	Short description
23/05/18	N.Esnault	N.Esnault	1st version

Contents

1. Glossary/Definition

- **RTCM**: Radio Technical Commission for Maritime Services
- **UoA**: University of Auckland
- **CORS**: Continuously Operating Reference Station
- **NMEA**: National Marine Electronics Association, define also the communication protocol (ascii)
- **UBX**: Ublox proprietary messages (binary)
- **RTKlib**: An Open Source Program Package for GNSS Positioning
- RPi/RPi0W: Raspberry pi, this project uses the version 0W (small form factor and built-in WiFi/Bluetooth)
- **Bootloader/BIOS**: Bootloader is a piece of code that runs before any operating system is running. Bootloader are used to boot other operating systems, usually each operating system has a set of bootloaders specific for it
- **M8T**: the GNSS receiver chosen and operating in this CORS. It is a receiver from the Swiss company U-blox. Note the "T". Its means this receiver has a relative better accuracy in timing so it is perfectly suitable for survey-in modes.

2. Quick data

For connecting from your computer to the CORS via Wi-Fi (wlan0)

- IP: 10.3.141.1

For connecting from your computer to the CORS via Ethernet cable (eth0) [you have to open the CORS and connect a USB to Ethernet to the RPi0W]

- IP: 192.168.137.137

If you wish to send commands or read what the station is doing:

- SSH default port : 22
- Username / password (raspberry pi Debian default): pi / raspberry

If you wish to read M8T data or modify its parameters:

- M8T data: port 54321
- No authentication

If you have an Xbee module and wish to monitor the communication between the CORS and the other GNSS receivers, you can use there parameters:

- Xbee S1
- Baud rate 57600
- Parity: None
- Flow control: none
- Bits: 8
- Stop bits: 1

If you are connected to the RPi0W (either by Ethernet cable or by Wi-Fi) and wish to modify the Wi-Fi of the raspberry pi:

Modifying Access Point data

- RaspAP
- admin / pa$$word

3. Normal operation

The following subchapter explain how to operate the CORS to start broadcasting GPS corrections data.

Note: in this chapter it is question of changing modes. Please take the time to understand that both the M8T and the RPi0W have 2 modes and have nothing in common.

1.1. Quick explanation

For the CORS to be able to send correction it need 2 pieces of information:

- the real-time data from the satellites as pseudo range, Doppler, and ephemeris (from the M8T)
- the actual position of the CORS

To get the second piece of information, the M8T must "survey" a static point by acquiring multiple GPS signal. Depending on the meteorological and topography (clear sky view and no multipath), this step can take from 10min to 5h.

1.2. The M8T modes

The M8T, as the RPi0W, has 2 modes. It automatically switches between survey-in to time mode[1]. There are 2 conditions (which can be altered) for the M8T to switch. In case these conditions are modified, it is necessary to restart from the survey-in mode.[2]

- 2 min have passed (minimum survey time)
- Minimum of 1m error RMS

On completion of these two criterions the M8T is ready to be used.

1.3. The RPi0W modes

In normal operation the RPi0W have 2 modes: the pass-through and the correction broadcast. Those 2 modes cannot co-exist and have to be made in that order.

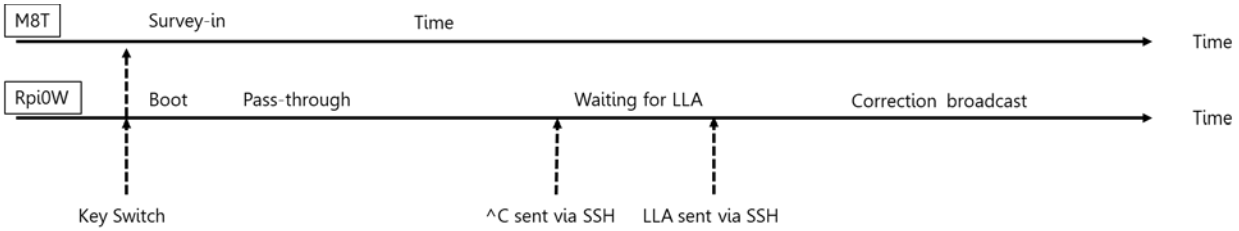


Figure 1: M8T and RPi0W modes

1.4. Software needed (all free and legal)

For operating the CORS, a SSH client software is needed to switch RPi0W mode. It exists multiple of SSH client software. Here we are going to use Putty (https://www.putty.org/). A

[1] Time mode is a GPS mode where the receiver knows so well its own true position that it can be used to calculate GPS signal errors. It is critical to wait for the M8T to be in time mode before the CORS starts calculating GPS corrections.

[2] This is usually done by saving the new parameters and restarting the CORS with the key.

second possible choice if you do not wish to use Putty is TeraTerm (https://ttssh2.osdn.jp/index.html.en).

Figure 2: Putty icon

For configuring the M8T, U-center (https://www.u-blox.com/en/product/u-center-windows) is needed. Though configuring the M8T is not necessary since the configuration already set up should be correct for all application.

Figure 3: U-center icon

1.5. Immobility of the CORS

The survey-in M8T mode starts when the system is powered on, there are no other required actions. Indeed, when you turn the key, all the CORS systems are powered on. Just after the power on, the RPi0W is booting up, but the M8T doesn't need to wait for the RPi0W to boot. The M8T automatically assume it is immobile and start tracking satellites.

Note: It is CRITICAL for the CORS to BE IMMOBILE from the moment the CORS is booted up to the moment you shut it down.

1.6. How to change RPi0W mode

1) Assure yourself that the station has enough power with the provided USB cable.
2) Position the CORS station to specific point that you choose. It should have a clear sky view and be in an open space.
3) Make sure the station is and will stay immobile. You can use the 4 holes in both flanges to fix it to the ground
4) Turn the key to power on the station

5) Prepare your computer with your preferred SSH client software. You can prepare the

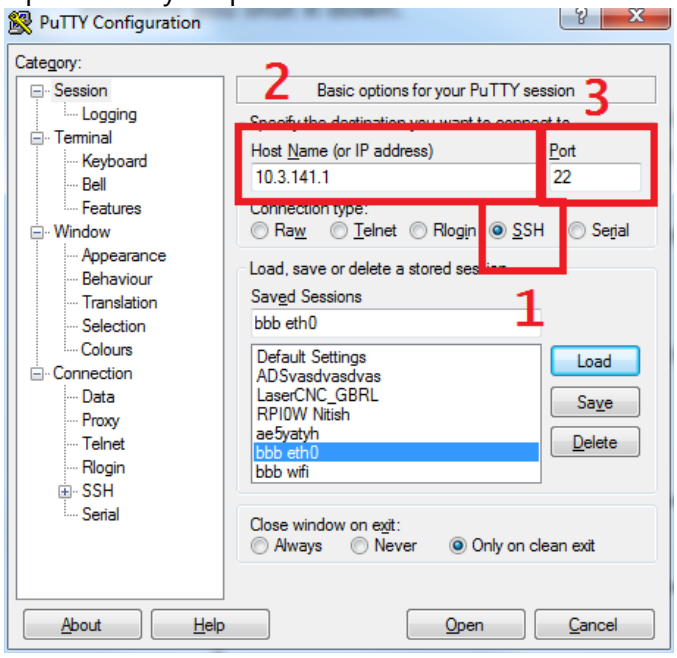


Figure 4: Preparing Putty

6) Wait for the Wi-Fi name M8T-CORS-RF115200 to appear in you available Wi-Fi names. Don't forget to refresh the list.
7) When the Wi-Fi is available, connect to it. It may take a bit of time for the RPi0W to accept your connection, don't worry it is normal.
8) When connected, you can initiate the SSH connection. If you are using Putty, you can press the "Open" located at the bottom right part of the window.

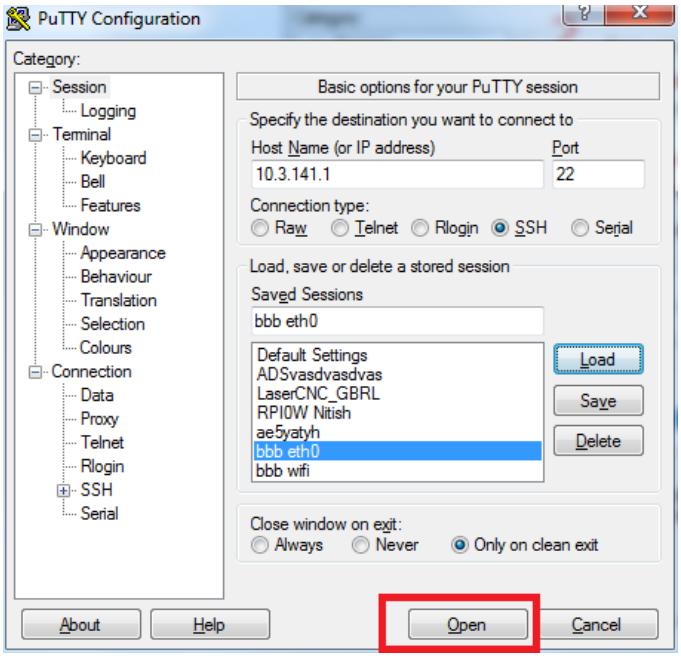
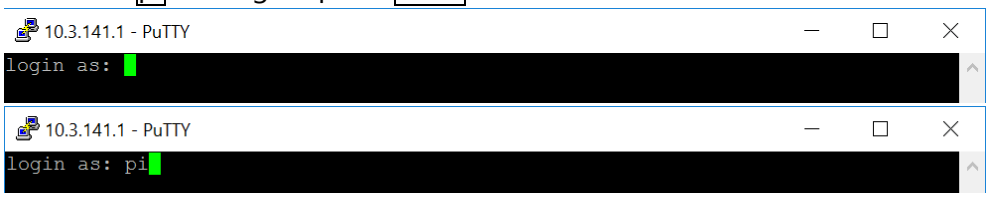


Figure 5: Initiate SSH communication

9) A window will appear and the RPI0W will ask you your credential.
   a. Enter first pi as "Login", press Enter.

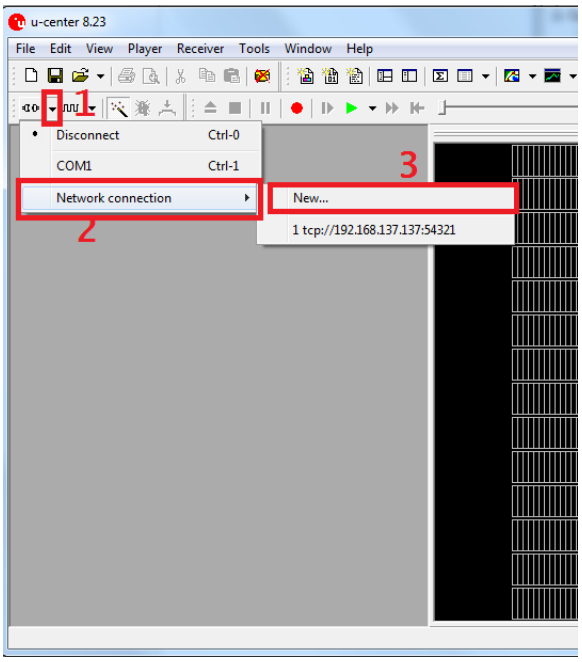


   b. Then it will ask for your password, which, for security reason, will not appear on the screen as you type it. Enter raspberry and press Enter.

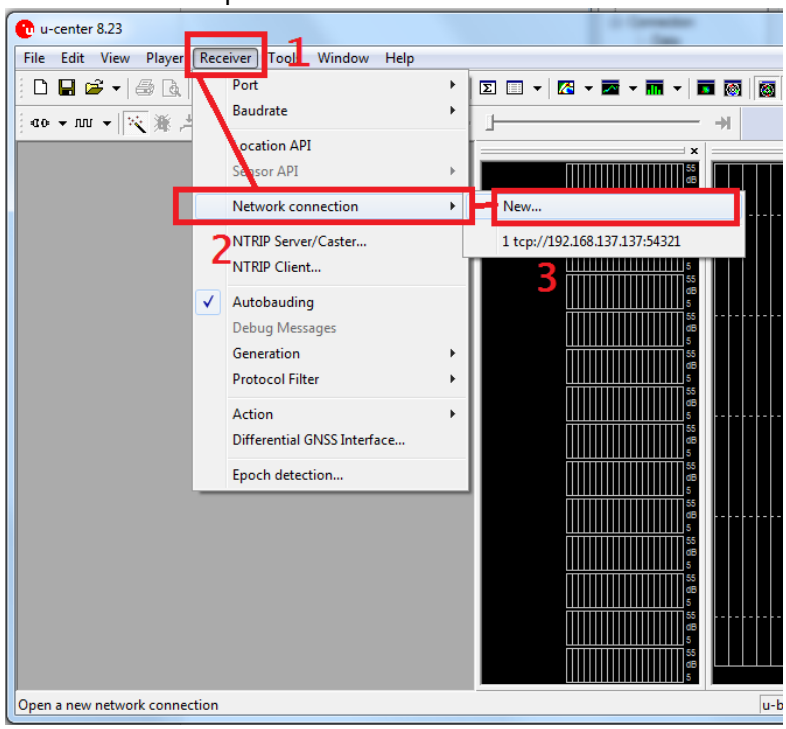
10) As you do this you will see the putty console becoming black. It is because the Rpi0W just stated the pass-through mode.
11) You can now monitor the M8T in U-center as a TCP/IP connection. The putty screen will remain in the state it is in right now until you decide to change the RPi0W mode.

The steps 12 to 14 are optional. Though they are highly recommended as to check the mode of the M8T.

12) Start U-center
13) Open a new TCP/IP connection. This step differ slightly depending if you have already done it before.
    a. If you have never done it before: You need to register a new connection as shown below.



Or you can do it with the top toolbar: Receiver → Network Connection → New.



Unconsiderate of what you have chosen previously, fill the window that poped up as below:

- Type: tcp
- IP: 10.3.141.1
- Port: 54321

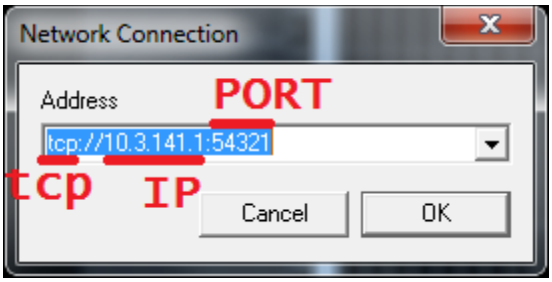


Figure 6: New network connection parameters in U-Center

If you have already registered this new network connection before, do as either one of the following screenshots:

Possibility a:

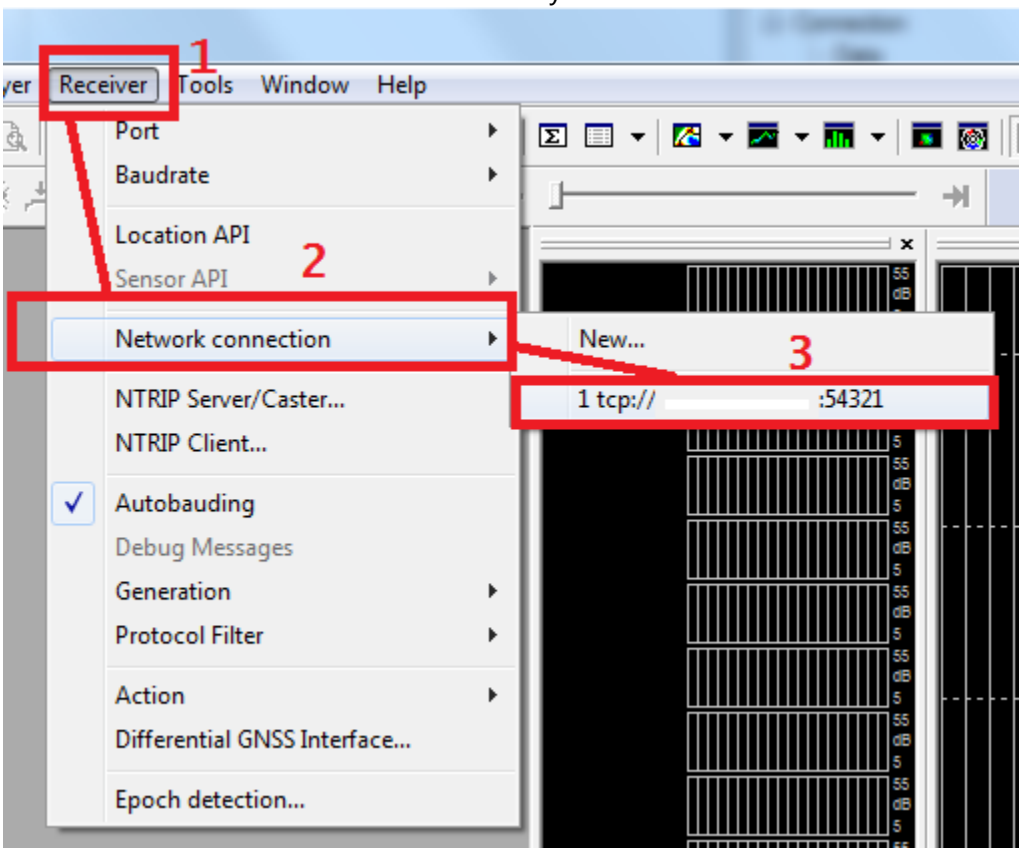


Figure 7: Starting an already registered network connection in U-Center via the top toolbar

Possibility b:

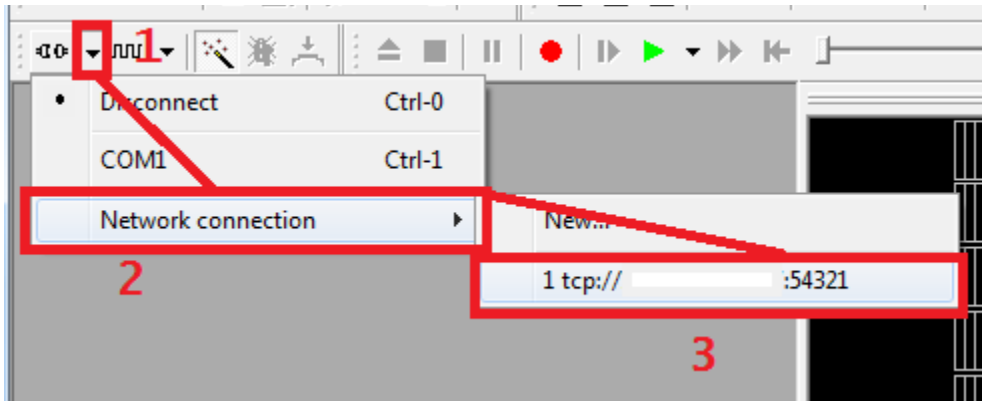


Figure 8: Starting an already registered network connection in U-Center via the quick connect button

Now you are connected to the M8T through the RPi0W. You can perform all the actions you normally do when using a direct connection to the M8T, the RPi0W will pass them through.

When in U-center, press F9 for message view. You can monitor the time mode of the M8T and enable and disable messages. If you disable the UBX messages, the CORS won't be able to generate the corrections. It is recommended to leave them enabled.

When the M8T has switched from survey-in to time mode and you want to start broadcasting the corrections perform the following steps:

14) Close U-center
15) Go back to putty console
16) Make sure you clicked in the console
17) Press Ctrl-c (^c).
18) The console should now start displaying some data about the messages sent. It is updated every 5s approximately. When you see this, it means correction data are being broadcasted via RF (Xbee S1 module) to all the listening Xbee S1 on the same channel (that include the rover).

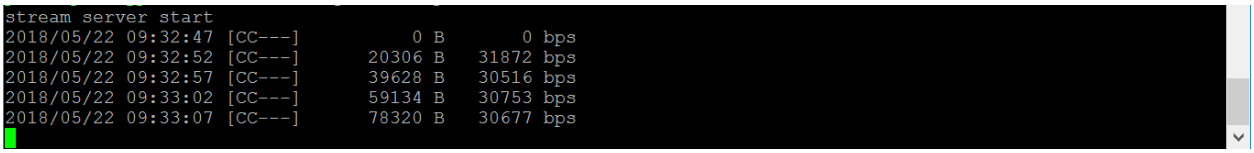


Figure 9: CORS console output when it starts to stream RTCM3 correction data via Xbee

19) That it for the correction. If you which to stop the system, either go to the Putty console, press ^c and then sudo shutdown now and turn the key. You can also turn the key directly without sending the previous command via Putty.

If you choose the last option, Putty will complain that it cannot reach the host anymore. This is not a problem, just acknowledge by pressing the OK button.

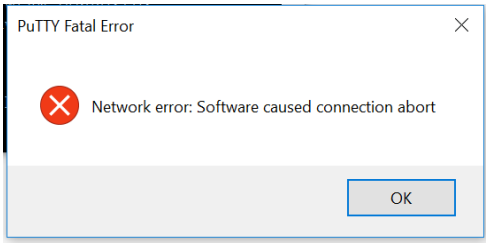


Figure 10: Putty network error

1.7. Monitoring the M8T to know when it is ready

It is possible to monitor if the M8T has switched mode by using U-center. The easiest way is to look on the right side of the U-center application and to look for a colored text next to the field "Fix mode". This field can only take a limited number of values:

- No FIX (red)
- 2D fix (blue or green)
- 3D fix (blue or green)
- 3DS + DR (strange orange)
- TIME (green)

The figure below is an example (in this case 3D):

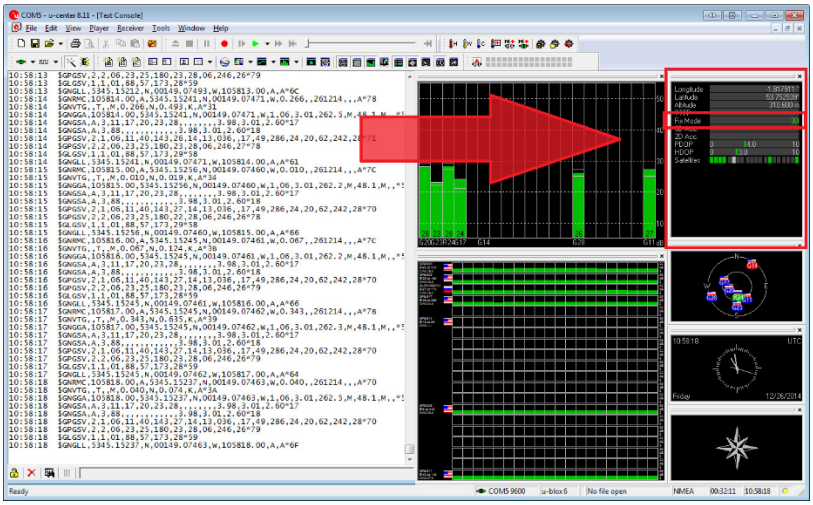


Figure 11: Fix mode in U-center

In survey-in mode, the fix status can be a variety of things but most likely will be "3D". In time mode, the fix status will switch to "TIME". So when you see "TIME" you know you can use the

The second way to monitor is to bring up the message view. Either press F9 or on the top toolbar: View->Message View.

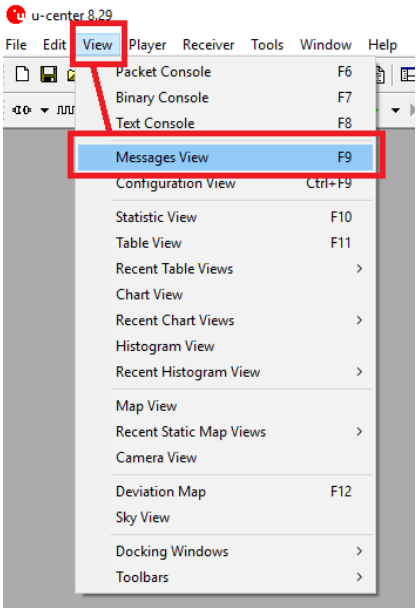


Figure 12: Enabling Message View via top toolbar

Then navigate to UBX-TIM-SVIN as shown in the picture below:

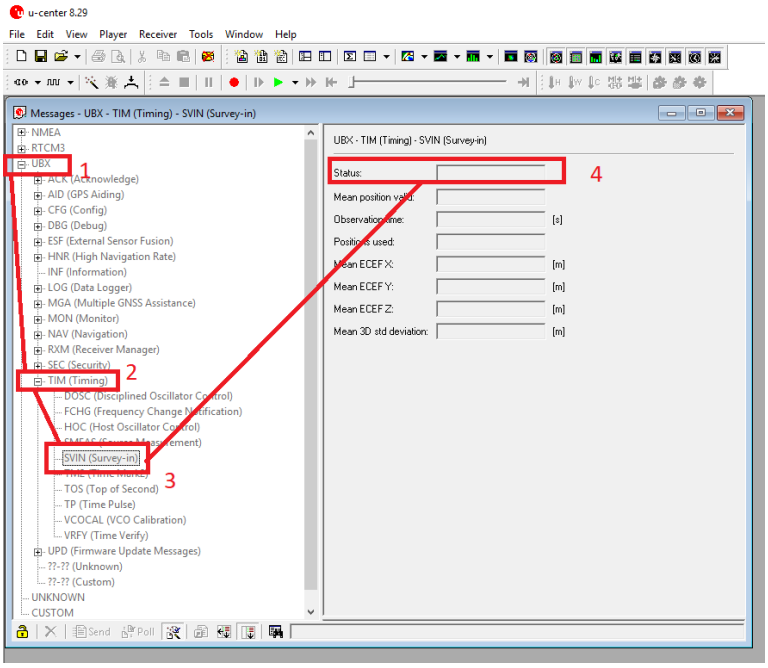


Figure 13: Checking Survey-in status of the M8T

The status can take 3 values:

-Not yet stated

-Running

-Successfully Completed

In order to know if the M8T is ready to be used, the SVIN status should be “Successfully Completed”

4. A quick peek inside

Coming soon

5. External links

- Author's GitHub page: https://github.com/Saultes45
- Author's GitHub Wiki: https://github.com/Saultes45/Bathymetry/wiki/MNU-code-and-package-used-from-the-community
- RTklib: http://www.rtklib.com/
- RaspAP: https://github.com/billz/raspap-webgui#manual-installation
- M8T product page: https://www.u-blox.com/en/product/neolea-m8t-series

6. RPi 0W data

Coming soon

- BerryBoot v2.0

7. Using the (Matlab) GUI

Coming soon

8. Commutating with the RPi0W (raspap-webgui)

Coming soon