

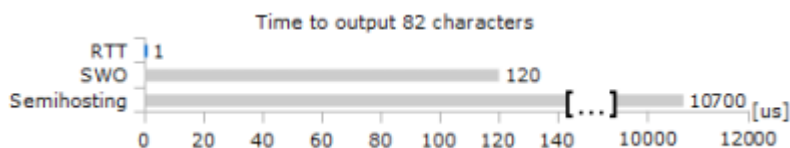
Segger Tools require a Segger probe to be used. For non-commercial use Segger offers the low priced „J-Link Edu“ probe

Segger supports the Cortex M7 SWV interface used on OVI40 UI and offers plug-ins for MCU eclipse.

For Debug output Segger offers two options in MCU eclipse:

- TRACESWO output as defined by CORTEX M7. This requires a dedicated pin, currently used for M3 pushbutton switch
- Segger Real Time Tracing (RTT) with outputs debug messages in high speed via SWV debug lines (CLK and SWData). No need for an additional line to printout debug messages

The speed advantage of RTT - according to Segger data - is impressive:



RTT speed comparison, from Segger web site [here](#)

## Install J-Link Software

### Debian SW Install

- Download .deb J-Link packages from [http://www.segger.com/download\\_jlink.html](http://www.segger.com/download_jlink.html)
- Download J-link .deb file, open terminal and go to download folder with terminal

```
$ sudo dpkg -i nameofdebpackage.deb
```

### Windows SW Install

ToDo

## Install & Configure J-Link in GNU MCU Eclipse

See [here](#)

### Install Packs

For installing CMSIS packs see [here](#)

- Install the GNU MCU Eclipse Packs
- Download CMSIS packs from Keil, see
- Install local cop of STM32F7 and STM32H7

## J-Link and Eclipse

- Describe Eclipse settings (MCU, CMSIS, probe settings, ...) to use J-link probe on Eclipse with uHSDR

## Segger Real Time Terminal RTT

- See [here](#)
- See

AN08005

(although this is for Cortex-A / Cortex-R and not the Cortex-M used by OVI40 it still gives some insights)

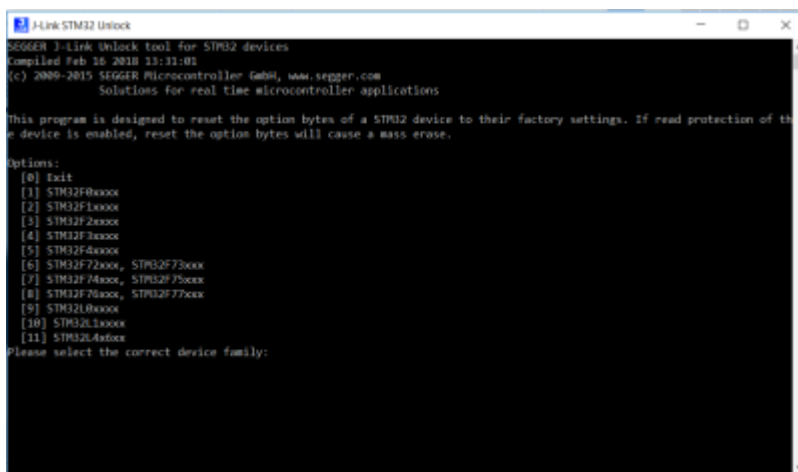
- [Try RTT on UHSDR](#) - is it really 100 times faster than SWO print and 10.000 times faster than semi-hosting? Segger says [here](#) that Cortex-M supports background memory access and that RTT will have no impact on Cortex-M execution speed

### RTT target code

The RTT target code is shipped as part of the J-Link Software and Documentation Pack which can be downloaded [here](#). The RTT sources can be found in the J-Link software package under: Samples/RTT

## J-Link support for Option bytes and Lock / Unlock

Segger provides a „STM32 unlock“ utility as part of the standard J-Link SW pack. The tool can be started in Windows using the „start“ menu and resets all values to factory default:



```
J-Link STM32 Unlock
SEGGER J-Link Unlock tool for STM32 devices
Compiled Feb 16 2018 13:11:01
(c) 2009-2015 SEGGER Microcontroller GmbH, www.segger.com
Solutions for real time microcontroller applications

This program is designed to reset the option bytes of a STM32 device to their factory settings. If read protection of the device is enabled, reset the option bytes will cause a mass erase.

Options:
[0] Exit
[1] STM32F0xxxx
[2] STM32F1xxxx
[3] STM32F2xxxx
[4] STM32F3xxxx
[5] STM32F4xxxx
[6] STM32F72xxxx, STM32F73xxxx
[7] STM32F74xxxx, STM32F75xxxx
[8] STM32F76xxxx, STM32F77xxxx
[9] STM32L0xxxx
[10] STM32L1xxxx
[11] STM32L4xxxx

Please select the correct device family:
```

Segger J-Link STM32 unlock utility

Further reading:

- Check Option Byte support on J-Link, see [here](#) and [here](#) for a starting point.
- See [here](#) and [here](#)
- See 3.12.2 J-Link STM32 Unlock (Command line tool) in [J-Link User Guide](#)

- J-Link adapter seems to support option byte programming via J-link utility and/or J-Flash utility. This is indicated in previous versions of the J-Link or J-Flash user guides, but not in the current ones. See for example section 7.5.1.2 [here](#)
  - ToDo: Verify and try out

## ToDo

- Checkout [Segger STM32 support Wiki](#) in general

## Further Reading

- J-Link User Guide
- J-Flash User Guide

From:

<https://amateurfunk-sulingen.de/wiki/> - Afu - Wiki des DARC OV Sulingen I40

Permanent link:

[https://amateurfunk-sulingen.de/wiki/doku.php?id=en:uhsdr\\_dev:segger\\_j\\_link&rev=1519481739](https://amateurfunk-sulingen.de/wiki/doku.php?id=en:uhsdr_dev:segger_j_link&rev=1519481739)

Last update: **24.02.2018 14:15**

