

Emitting Hellschreiber from a Raspberry Pi GPIO: combining gr-hellschreiber with gr-rpitx

GNU Radio running on an embedded board as emitter

Jean-Michel Friedt¹, Thomas Lavarenne²

¹FEMTO-ST Time & Frequency department, Besançon, France

¹Lycée Jean Rostand, Villepinte, France

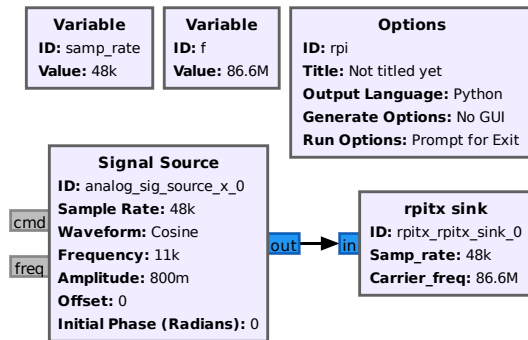
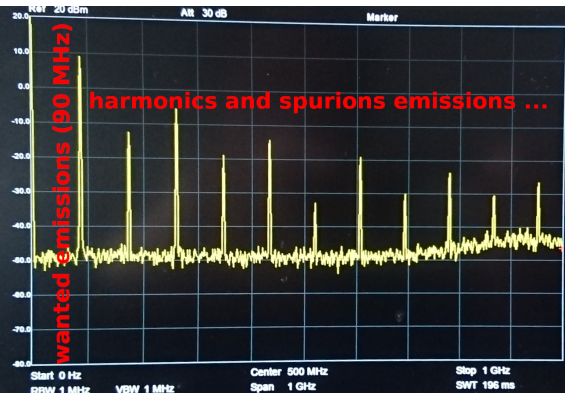
jmfriedt@femto-st.fr

SDR DEVR	FOSDEM	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	FOSDEM	SDR DEVR
SDR DEVR	FOSDEM	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	FOSDEM	SDR DEVR
SDR DEVR	FOSDEM	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	SDR DEVR	FOSDEM	SDR DEVR

slides and references at jmfriedt.free.fr

gr-rpitx sink block

- ▶ `gr-rpitx`¹: connect any (10-250 kHz bandwidth limited by DMA rate) GNU Radio processing sequence to a transmitting sink
- ▶ be aware of the many harmonics and spurious: bandpass filtering is mandatory
- ▶ **do not** emit over the air unless you know what you are doing

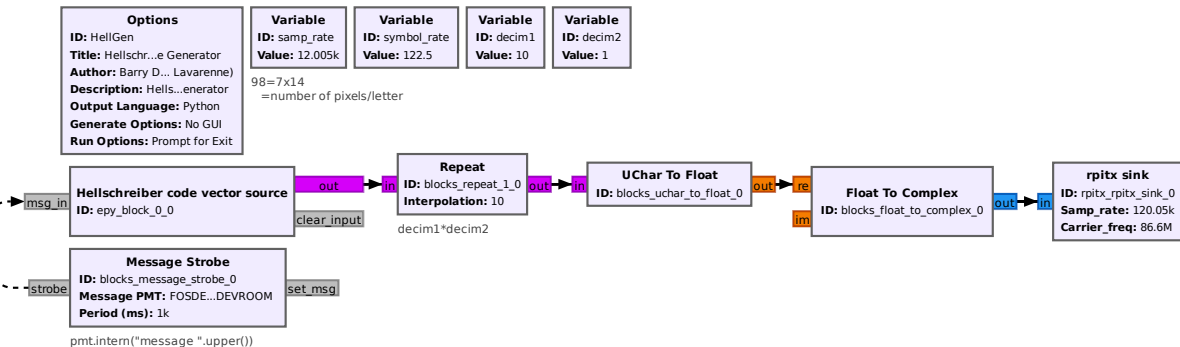


¹<https://github.com/jmfriedt/gr-rpitx>

gr-hellschreiber² source and sink blocks

gr-hellschreiber: low bandwidth Fax-like graphical communication mode

- ▶ emitter block (to rpitx, running on the embedded board)
- ▶ receiver block (from RTL-SDR dongle, running on the embedded board)
- ▶ decoder block (ZMQ stream to binary picture): display \Rightarrow running on laptop



²<https://github.com/tlavarenne/gr-hellschreiber>

Hellschreiber fonts and communication protocol



a

^a<https://www.nonstopsystems.com/radio/hellschreiber-fonts.htm>

a. Definition

<https://www.qsl.net/z11bpu/DOCS/Hellspec.pdf>

Feld-Hell is an on-off keyed uncoded text transmission system. Individual text characters are defined by dots in a dot matrix, and the matrix is transmitted in a fixed raster-scanned sequence. The transmission of dots is nominally timed at 122.5 baud, but no synchronising system is used. The receiving system operates quasi-synchronously, ie. is adjusted to run at about the same speed, and small errors in speed are compensated for by displaying the received image twice, so that phase errors have limited effect. The system is thus immune to the sync-related errors of many text transmission systems

Feld-Hell is relatively immune to noise interference, does not require complex transmission equipment, and performs well with low power.

b. Coding

Feld-Hell is uncoded, but sent as a raster scan of text characters. At the receiving end the received data, with errors, is displayed for visual interpretation. Since no coding is used, errors introduce noise, and thus effect the readability of the text, but cannot change the characters from one to another, as happens in coded systems, such as radio teletype.

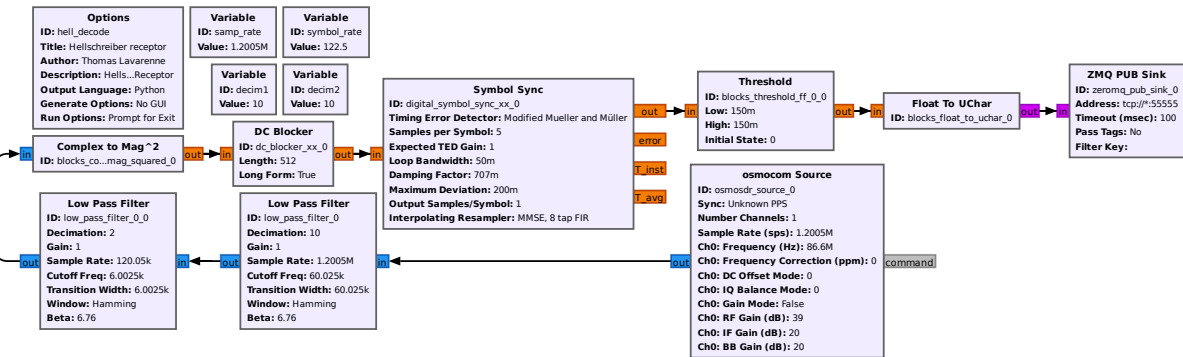
a

^a<https://www.qsl.net/z11bpu/DOCS/Hellspec.pdf>

gr-hellschreiber³ source and sink blocks

gr-hellschreiber: low bandwidth Fax-like graphical communication mode

- ▶ emitter block (to rpitx, running on the embedded board)
- ▶ receiver block (from RTL-SDR dongle, running on the embedded board)
- ▶ decoder block (ZMQ stream to binary picture): display \Rightarrow running on laptop



³<https://github.com/tlavarenne/gr-hellschreiber>

GNU Radio 3.9 Buildroot cross compilation

- ▶ Cross-compile ⁴ ⁵ with GNU Radio 3.9 on Buildroot : make menuconfig and select GNU Radio 3.9
- ▶ Connect blocks with GNU Radio Companion on the PC, export resulting Python script to RPi and execute on embedded board

```
buildroot-2021.11 RP14 GNURadio39/ config - Buildroot 2021.11 Configuration
external options
External options
Arrow keys navigate the menu. <Enter> selects submenus --> (or empty submenus ---).
Highlighted letters are hotkeys. Pressing <Y> selects a feature, while <N> excludes a feature.
Press <Esc>/<Mac> to exit, <?> for Help, </> for Search. Legend: [ ] feature is selected [ ]
feature is excluded
(*) XTRX Packages ***
[ ] libim7052a
[ ] libim7050
[ ] libstrxll
[ ] libstrxll
[ ] libstrxdep
*** XTRX Packages end ***
[ ] pluto default system_top.bit
[ ] poll_sigs
*** gr-iiio needs gnradio w/ python support ***
[*] vxll
[*] librpitx
[ ] ppls
[ ] r8_lib
[*] gnradio
[*] gr-audio
-- blocks support
[ ] criport support
[ ] gr-dtv support
[ ] gr-fec support
[*] python support
[*] gr-utils support
[*] gr-zeromq support
-- gr-analog support
[ ] gr-channels support
[*] gr-digital support
-- gr-ft support
-- gr-filter support
[ ] gr-trellis support
[*] gr-uhd support
-- volk
(*)
[Select] < Exit > < Help > < Save > < Load >
```



⁴G. Goavec-Merou & J.-M Friedt, *Never compile on the target ! GNU Radio on embedded systems using Buildroot*, FOSDEM (2021) at

https://archive.fosdem.org/2021/schedule/event/fsr_gnu_radio_on_embedded_using_buildroot/

⁵https://github.com/oscimp/oscimp_br2_external

Conclusion

- ▶ functional port of GNU Radio to Buildroot for developing on embedded boards
- ▶ transition from GNU Radio 3.8 to 3.9 eased by OOT Porting Guide ⁶
- ▶ Raspberry Pi (4)+RTL-SDR as affordable hardware to become familiar with narrow bandwidth (<200 kHz) communication protocols implemented as SDR

⁶https://wiki.gnuradio.org/index.php/GNU_Radio_3.9_OOT_Module_Porting_Guide