

# Jinga-Hi

---

from the FieldTrip wiki (<http://www.ru.nl/donders/fieldtrip/>)

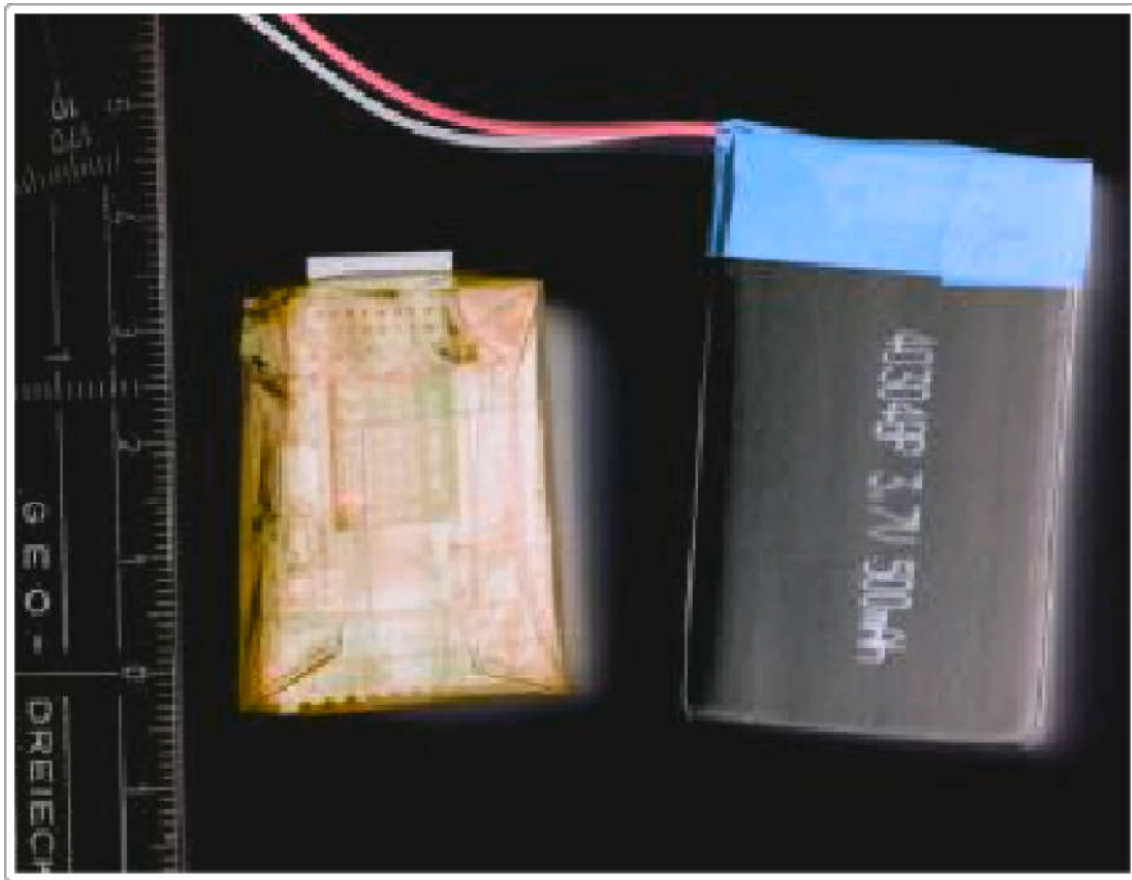
- MATLAB-based interface
- Standalone interface with jaga2ft

This page describes the interface between the Jinga-Hi JAGA16 device and the FieldTrip buffer.



The Jinga-Hi JAGA16 is a miniaturised stand-alone device includes amplifier, digitizer, signal processor and transmitter. The device has 16 channels. Although the JAGA16 has been primarily designed for electrophysiological recordings (spikes and LFPs) in small animals, its specifications also make it very applicable to wireless EEG-BCI systems.

The packaged version that is depicted above includes a micro-USB interface at the side that can be used to power the device, although for normal operation it should be battery powered. Below is the device (wrapped in protective foil) besides a 500mAh battery.



Jinga-Hi releases MATLAB (mex-based) and Python code to interface with the device. Since the device dumps the data in a well-defined format over a UDP connection, it is not hard to implement your own software to access the data.

We provide two interfaces between the JAGA16 device and the FieldTrip buffer: one implemented in native MATLAB and one implemented as stand-alone application in C.

The data files created by Jinga-hi software can also be read using `ft_read_header` and `ft_read_data` for offline analysis.

## MATLAB-based interface

---

The JAGA16 device streams the data over UDP network connection. The `ft_realtime_jaga16proxy` function (part of the `realtime` module in FieldTrip) sets up a UDP server that listens to port 55000 and writes all data it receives to a FieldTrip buffer. The FieldTrip buffer is a multi-threaded and network transparent buffer that allows data to be streamed to it, while at the same time allowing another MATLAB session on the same or another computer to read data from the buffer for analysis.

Subsequently in another MATLAB session you can read from the FieldTrip buffer using the `ft_read_header`, `ft_read_data` and `ft_read_event` functions by specifying 'buffer://hostname:port' as the filename to the reading functions, e.g.

```
hdr = ft_read_header('buffer://hostname:port');
dat = ft_read_data('buffer://hostname:port', 'begsample', 1, 'endsample', hdr.Fs);
```

The UDP network interface in MATLAB is implemented in the freely available TCP/UDP/IP toolbox. You should download this toolbox and add it to your MATLAB path if you want to use the `ft_realtime_jaga16proxy` function.

The MATLAB implementation is mainly for educational and testing purposes. For proper real-time analyses we recommend you to use the standalone interface, which is faster and requires less system resources.

## Standalone interface with `jaga2ft`

---

Instead of `ft_realtime_jaga16proxy` and MATLAB, you can use `jaga2ft` to transport data from the UDP network connection to a FieldTrip buffer. `jaga2ft` is written in C and takes 2 optional command line arguments

---

```
jaga2ft [ftHostname] [ftPort]
```

For example, if you want to stream the data to a remote buffer on mentat205:1972, you would type

```
jaga2ft mentat205 1972
```

---

For spawning a local FieldTrip buffer within **jaga2ft** at port 1234, you would use a dash (-) instead of the second hostname and write

```
jaga2ft - 1234
```

---

Leaving out all arguments spawns a local buffer on the default port 1972:

```
jaga2ft
```

---

## Compilation

On the command line, change to the `realtime/src/acquisition/jaga` directory and type `make`. Note that you might need to compile the buffer library first.