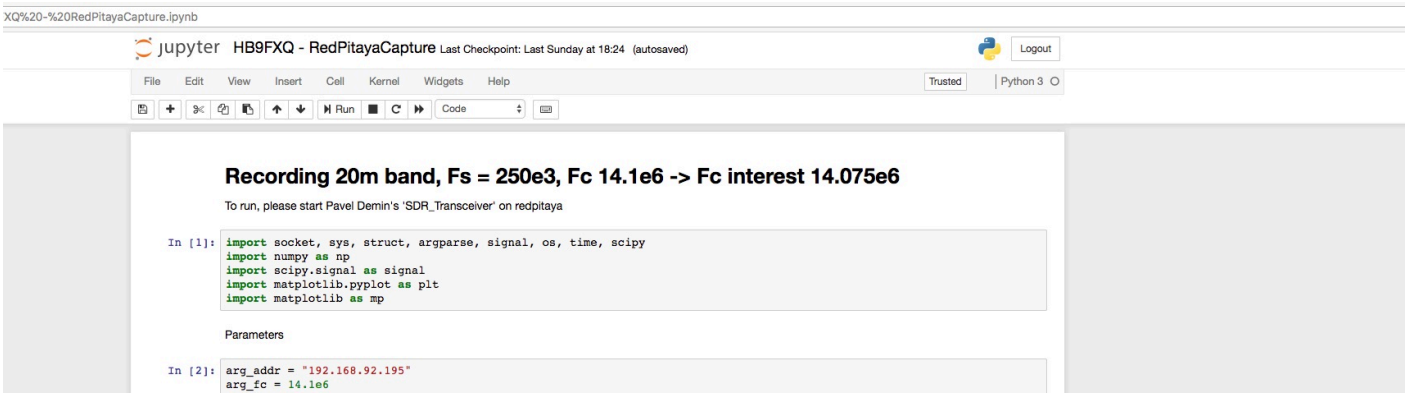
Using the RedPitaya in a Jupyther Notebook

03.03.2018 10:20



Using Jupyter notebooks for rapid prototyping, short proof of concepts and documentation is fun. Now I wanted to introduce some signal aquisition using the RedPitaya (+ http://pavel-demin.github.io/red-pitaya-notes/sdr-transceiver/).

Pretty simple task. Two years ago I've contributed a short python script to fetch samples directly from the trx project. Pretty simple to include it directly into a notebook:

```
fs_to_indexfs = {20000:0, 50000:1, 100000:2, 250000:3, 500000:4, 1250000:5}

class Transceiver(object):
def __init__(self, ipaddr):
self.p_ip_address = ipaddr
self.control_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
self.data_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

def shutdown(self):
print("RX Shutdown")
self.data_socket.close()
self.control_socket.close()

def start_rx(self, freq, fs_index, corr, n_bytes, iq_filename):
# Setup control socket
self.control_socket.connect((self.p_ip_address, 1001))
self.control_socket.send(struct.pack('>> rx force shutdown\n")
self.shutdown()
f_iq_raw.close()
```

The acquisition looks like:

```
trx = Transceiver(arg_addr)
start = time.time()
activeReceiver = trx;
```

```
trx.start_rx(arg_fc, fs_to_indexfs[arg_fs], 0,arg_bytes_to_fetch, arg_raw_iq_file)

end = time.time() - start
print(time.strftime("%H:%M:%S", time.gmtime(end)))

samples = np.fromfile(arg_raw_iq_file, dtype=np.complex64)
x1 = np.array(samples).astype("complex")
print(len(x1))
```

That’s it. Now there is some numpy / matplotlib fun ahead. Let’s quickly render e spectrogram: