AN24-02 DemoRad
FMCW Basics

Inras GmbH
Altenbergerstraße 69
4040 Linz, Austria
Email: office@inras.at
Phone: +43 732 2468 6384

Linz, July 2017

1 FMCW Basics

In this application note, we shows how to perform FMCW measurements with the 24-GHz DemoRad radar system. First the RF transceivers are configured and in the next step the sampled IF signals are recorded.

The aim of the tutorial is to:

- configure FMCW measurements,
- access the sampled IF signals,
- understand data format of sampled IF signals.

The `Adf24Tx2Rx4` class design is used to configure the RF board. The implementation is inherited from the DemoRad class. Therefore it offers additional methods for configuring the FMCW part of the radar system. In Fig. 1, a block diagram of the RF part is shown. In the following paragraph the configuration of the transceiver chips ADF4159, ADF5901, and ADF5904 with the DemoRad is described in more detail. This application note highlights how to perform FMCW measurements without handling the configuration of all the registers of the transceiver chips.

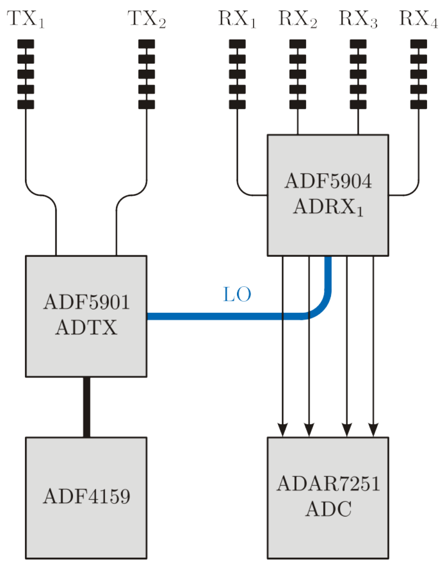

Figure 1: Block diagram of the DemoRad 24-GHz radar system.

In the first step the DemoRad must be powered on and connected to the PC. In addition, the `@Adf24Tx2Rx4`, `@DemoRad`, `@UsbAdi`, `@DevAdf5904`, `@DevAdf5901`, and `@DevAdf4159` classes as well as the `DemoRadUsb` mex file must be added to the Matlab search path. After generating a class object

```
Brd = AdfTx2Rx4();,
```

the frontend part and the sampling chain with the ADAR7251 can be configured. In this note we show how to perform FMCW measurements. The desired timing for the instantaneous transmit frequency $f_{\mathrm{t}}(t)$ is shown in Fig. 2. The current DSP software version `'1.0.0'` only supports a fixed timing for the chirp sequence. I.e. the chirp duration T_{Up} and the chirp repetition interval T_{p} are fixed to 280 μs and 284 μs, respectively. Only the start and the stop frequency of the FMCW waveform can be configured with the DSP software framework. The sampling chain is configured

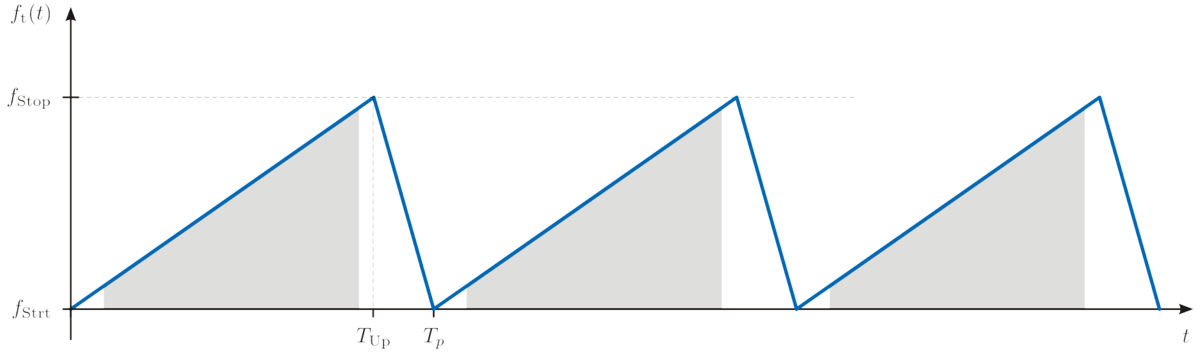

Figure 2: Measurement timing with instantaneous transmit frequency $f_{\mathrm{t}}(t)$.

to record the IF signals of all four receive channels with a fixed sampling rate of 1 MSPS during the gray colored part of the upchirp shown in Fig. 2. During this time 256 samples are recorded. The method

```
Brd.RfRxEna();
```

enables all IF channels of the receive chip (ADF5904). In the next step the transmit chip is enabled by

```
Brd.RfTxEna(1, 100);
```

where the first argument specifies the transmit antenna to be activated and the second argument specifies the output power. The argument for the transmit antenna is in the range from `0` to `2`, where `0` deactivates the transmit path. According to the datasheet a value in the range from 0 to 100 is allowed for the output power. Thereafter the timing of the transmit signal is specified with the following structure:

```
Cfg.fStrt = 24.0e9;
Cfg.fStop = 24.25e9;
Cfg.TRampUp = 280e-6;
Cfg.Tp = 284e-6;
Cfg.N = 256;
Cfg.NrFrms = 100;
```

The parameter `TRampUp` defines the duration of the upchirp and the chirp repetition interval is specified with the parameter `Tp`. This parameters are only specified for the calculation of the chirp rate. A change to the parameters has **no** effect on the programmed FMCW timing. The number of samples during a single upchirp is also fixed to 256 and a change to the parameter `N` has no effect on the sampling frequency in the current DSP software version. The DSP always collects 128 subsequent upchirps and with the `StrtIdx` and `StopIdx` the number of chirps send to the PC can be defined. To configure the synthesizer and start the FMCW timing

```
Brd.RfMeas('Adi',Cfg);
```

is called with the first argument `'Adi'` and the second argument being the previously defined configuration structure. After the configuration is programmed, the method

```
Data = Brd.BrdGetData();
```

can be used to read back the sampled IF signals. Every call of `Brd.BrdGetData()` returns the IF signals for all four channels and the number of chirps `Cfg.StopIdx - Cfg.StrtIdx` specified in the configuration structure. In Fig. 3 the sampled signals for a single chirp are plotted.

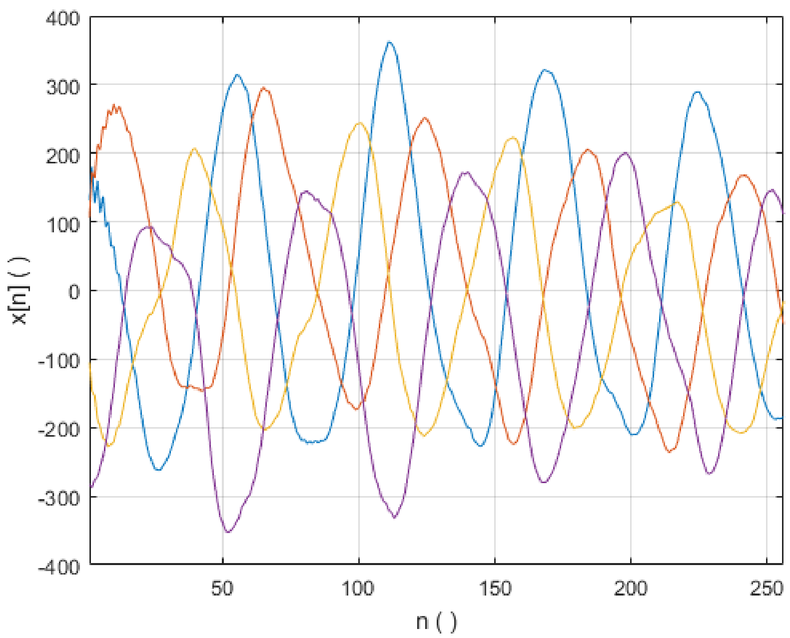

Figure 3: Sampled IF signals for a single upchirp and four data channels (plot(Data)).

Data Format of the IF Signal

The number of chirps returned depends on the selected `StrtIdx` and `StopIdx`. The difference of both indices `Cfg.StopIdx - Cfg.StrtIdx` is the number of chirps returned from the `BrdGetData()` method. In Fig. 4, the returned array for a single chirp is shown. The columns contain the sampled values for the four receive channels and the number of rows is the number of samples `N` for a single chirp.

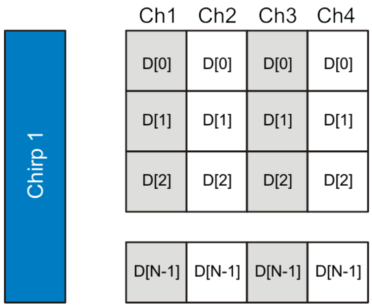

Figure 4: Returned array for a single chirp.

If two chirps are configured then the structure of the array is shown in Fig. 5. The columns correspond to the different receive channels and the signal of the second chirp is appended after the first one.

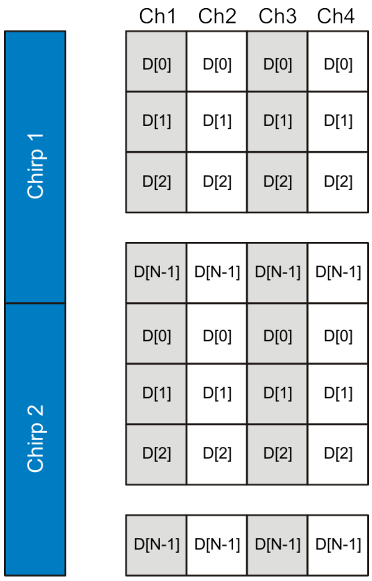

Figure 5: Returned array for two chirps.