TI mmWave Radar Set Up

1 Set Up

1.1 Github Download Instructions

This document is designed to provide you with essential information and resources for the course. In this section, you will find instructions on how to download course materials and resources from GitHub.

To get started with the course, you need to download the necessary course materials from our GitHub repository. There are multiple ways to download the repository.

1.1.1 Fork Via Terminal (Useful to Share Code within Groups)

1. Open up terminal (if on Linux or MacOS) or download Git Bash for Windows.
2. Go to the directory where you would like to clone the project repository to (use `cd folder-name`).
3. At the top of the GitHub repository select Fork, and select yourself as the owner. You can also change the name of your project if you would like (useful to differentiate between groups).
4. Click create fork.

More detailed instructions on how to fork and eventually how to clone from your forked repository are available here.

To clone this new forked repository into your computer:

1. Click on the green code button on your repository front page.
2. Copy the SSH link for Cloning.
3. In terminal or Git Bash, type `git clone git@github.com:<owner name>/comm-proj-radar.git`, making sure to do this in the directory you want to save the repository to.

1.1.2 Download Via Zip

1. Visit the link to your GitHub repository.
2. Click on the "Code" button, and select "Download ZIP" to download a zip archive of the repository.
3. Once the download is complete, extract the files to your local machine.

Now you have all the course materials ready for your study and projects.

1.2 Repository Organisation

This GitHub repository contains:

- The notebooks you will have to do as exercises.
- A notebook evaluation.
- A script to have live streaming of the radar (located at: `/streaming/realtime_streaming.py`).
- A Python library:
  - In folder `processing` you will find Python scripts used to process the data.
  - In folder `streaming` you will find Python scripts used to have a continuous stream of data coming from the radar and plot it with `realtime_streaming.py`.
  - In folder `scripts` you will find everything linked to the setup of the radar.
  - Folder `record` is the folder where you are expected to save samples captured with the radar.

1.3 TI Software Setup

mmWaveStudio is the software that will allow us to format and run the radar boards.

1.3.1 Requirements

- Windows machine with 2 USB ports and an ethernet port.
- XWR1X43BOOST board.
- DCA1000EVM.
- Two 5V power adapters.
- 2 USB to micro-USB cables.
- 1 Ethernet cable.

If you lack some of the ports, you will have to use one of the adapters that we will provide. If your machine is not running on Windows, you will have to use a laptop that will be at your disposition. This is because MMWave studio software is only supported on Windows.

1.3.2 Installing TI Software

Follow these steps to install the necessary TI software:

1. Install mmWave studio 2.1.1.0 (or whatever is newest and compatible with XWR1X43BOOST).
2. Install MATLAB runtime R2015aSP1 (8.5.1) in 32-bit: MATLAB-RUNTIME.
3. Install XDS Emulation Software (32-bit version): XDS.
4. Configure network settings to communicate with the DCA board by setting up the Ethernet network setting to (192.168.33.30 / 255.255.255.0).

5. Install Python via Anaconda.
6. Install additional required packages:
   - `pip install mat73`
   - `pip install panda3d`
   - `pip install pythonnet`
   - `conda install -c anaconda scipy`

2 Understanding Radar Data

The radar data is streamed via the Ethernet cable connected to the DCA1000. It saves the data in a binary format, which needs to be reformatted using `singlechip_raw_data_reader_example.py` to be usable.

3 Processing Captured Data

3.1 Data Visualization

Once reformatted, the data can be processed in various ways. The provided `visualize_data.ipynb` notebook shows an example pipeline for visualization.

4 Realtime Data Processing

To stream and process data in real time:

1. Run the radar in continuous capture mode.
2. Close mmWaveStudio and related tasks.
3. Run `realtime_streaming.py` to visualize data in real time.

This script reads from the Ethernet port, processes data, and updates plots dynamically.