mmwave Documentation

Release 0.1.0

Aaron Finney

Sep 02, 2019

Contents

1 Overview **3**

2 Installation **5**

3 Usage **7**
3.1 Capture Object Properties and Methods 8
3.2 Included FileWriter Output Sink Properties and Methods 8

A python module for capturing and processing streamed ethernet data from the Texas Instruments DCA1000EVM Radar Data Capture Card.

CHAPTER 1

Overview

Currently, the available functionality includes:

- listen for streamed capture data on UDP port 4098 *(Capture module)*
- save the streamed data to an output file *(FileWriter output sink)*

CHAPTER 2

Installation

The current version is pre-alpha, and has not been packaged/deployed. To install it:

1. clone this repo
2. cd to the mmwave project root directory
3. run ‘python setup.py install’ to install the module

CHAPTER 3

Usage

To use the module, create an instances of the Capture object class and an instance of one or more output sink objects. Configuration options can be set either using object properties or passing them in the object initialization:

```
# import the Capture module and the default file writer output sink
from mmwave import Capture, FileWriter

# create new instance of Capture object
listener = Capture()

# create new instance of FireWrite output sink
# you can set config options by passing them as arguments
filesink = FileWriter(output_filename = "/root/adc_raw_seq_out.bin")

# you can also set config options as properties
filesink.output_format = "RAW_SEQ"  # set the output format to RAW_SEQ
filesink.output_overwrite = False

# add the output sink to the Capture object instance (returns True if successful)
if not listener.add_sink(filesink):
    # put in logic to handle failure adding sink here
    pass

# start listening for packets on UDP port 4098
# by default, it listens forever for the first packet, and
# stops listening if no packet has been sent for 1 second
listener.start()

# view stats for this capture
print(listener.stats)
```

The listener will wait ad infinitum for the first packet, unless program execution is stopped. Once the first packet has been received, the listener will time out and clean up 1s after the last packet is received.

3.1 Capture Object Properties and Methods

Properties

- bind_address (string, rw) - the address to bind the listener too *(default: '0.0.0.0')*
- source_format (string, rw) - the source format for the incoming stream, can be one of - RAW *(default)* - DATA_SEPARATED_MODE *(not yet implemented)*
- message_window_size (integer, rw) - the number of messages in the streaming window, valid range is 1-128 *(default is 16)*
- output_sinks (list of objects, ro) - returns all added output sinks
- stats (dictionary, ro) - returns execution stats - messages - the total number of messages processed - bytes - the total number of payload bytes received (actual) - out_of_order - count of out of order messages - missing - count of missing messages (gaps in sequence numbers)

Methods

- add_sink(<output-sink-instance>) - send an output stream to the specified output sink instance
- start() - start listening

3.2 Included FileWriter Output Sink Properties and Methods

Properties

- name (string, rw) - a name tag to uniquely identify this output sink *(default: 'FileWriter')*
- output_filename (string, rw) - the destination path/filename for the output file *(default: outfile.bin)*
- output_format (string, rw) - the output format to send to the sink, can be one of - RAW_NO_SEQ - output file only contains binary capture payload *(default)* - RAW_SEQ - output file contains binary capture payload prepended with message sequence number, message payload size, and total capture size - DATA_SEPARATED_NO_SEQ *(not yet implemented)* - DATA_SEPARATED_SEQ *(not yet implemented)*