Remote mmWave Data Capture System Software Developer Guide

Revision 1.00
<13-Dec-2021>

Revision History

Version	Date	Author	Description
1.00	14-Dec-2021	Michael Dittman	Initial draft

Document Description

This document describes the calling conventions of the BeagleBone API developed for Texas Instruments for the DCA1000EVM.

Contents

1 Introduction

This document outlines the calling conventions for the BeagleBone API developed for the DCA1000EVM. The BeagleBone API was developed to collect analog RADAR data from an antenna connected to the DCA1000EVM. The DCA1000EVM board sends the analog data to an SBC connected to the DCA1000EVM via an ethernet connection. The BeagleBone API was designed with the intention to collect analog radar data and video data simultaneously with an SBC, but could also be used on any Linux or Windows machine with proper configuration. This document outlines key functionality of the API and how the user can utilize the API.

1.1 Terms / Acronyms / Abbreviations

API	Application Programming Interface
CLI	Command Line Interface
COM	Communication Port
DCS	Data Capture System
EVM	Evaluation Module
FPGA	Field Programmable Gate Array
HMI	Human Machine Interface
JSON	JavaScript Object Notation
PIP	Pip Installs Python (Python Package Manager)
RADAR	Radio Detection and Ranging
SBC	Single Board Computer
SD	Secure Digital
UART	Universal Asynchronous Receiver-Transmitter

Table 1: Terms and Abbreviations

1.2 Audience

Anyone interested in utilizing the BeagleBone API for a DCA1000EVM project.

2 Required Libraries

The BeagleBone API is written in the Python language. As such, much of the functionality comes from pre-existing libraries that can be installed via PIP or another package manager.

2.1 Python Libraries

This section outlines the packages(libraries) required for the BeagleBone API.

Package Name	Install Command with PIP
serial	pip install pyserial
numpy	pip install numpy
cv2	pip install cv2
os	Pre-existing in python
time	Pre-existing in python
math	Pre-existing in python
sys	Pre-existing in python

Table 2: Required Packages for BeagleBone API

3 Specific Functionality

This section contains an enumeration of the key functionality the DCS provides through the BeagleBone API.

3.1 System Functionality

The System Functionality section gives a concise language description of the functionality the DCS provides through the BeagleBone API. Also listed is what function can be used to provide the described service. RADAR acquisition is accomplished by utilizing the DCA1000EVM CLI.

3.1.1 Connect the SBC or other machine to COM ports.

a) connect_com_ports(uartCom, dataCom)

b) uartCom – The UART port to connect to. Labeled as “XDS110 Class Application/User UART”.

c) dataCom – The data port to connect to.

d) Both uartCom and dataCom should be passed as filenames on Linux (ex. tty/ACM0) or numbers on windows (ex. 1)

3.1.2 Send configuration file (chirp file) to the antenna over COM ports.

a) send_cfg(cfg, uartCom)

b) cfg – The configuration file to send over the UART port.

c) uartCom – The UART port to send configuration file to. Labeled as “XDS110 Class Application/User UART”.

d) ‘cfg’ should be sent as a python file object. This can be retrieved using the get_cfg(path) function with the parameter ‘path’ representing the file location.

3.1.3 Configure FPGA with JSON file.

a) configure_fpga(json_file_path)

b) json_file_path – The path of the JSON file to configure the FPGA with.

3.1.4 Configure the record delay between packets sent by the DCA1000EVM.

a) record_delay(json_file_path)

b) json_file_path – The path of the JSON file to configure the record delay.

3.1 System Functionality Cont.…

3.1.5 Start RADAR acquisition process.

a) start_record(json_file_path)

b) json_file_path – The path of the JSON file to configure the start record process.

3.1.6 Stop RADAR acquisition process.

a) stop_record(json_file_path)

b) json_file_path – The path of the JSON file to configure the stop record process.

3.1.7 Start a video record process.

a) start_video_acqusition()

3.1.8 Configure the FPGA, record delay, start RADAR record, and video record simultaneously.

a) radar_record_start(json_file_path)

b) json_file_path – The path of the JSON file to configure the various parameters.

4 References

[1] "PIP Documentation." *Installation - Pip Documentation v21.3.1*, https://pip.pypa.io/en/stable/installation/.

[2] "MMWAVE-SDK." MMWAVE-SDK Software Development Kit (SDK) | TI.com