Overview

This lab demonstrates the use of TI mmWave sensors for gesture recognition applications. The range, velocity, and angle data from mmWave sensors can enable the detection and classification of several natural gestures. The example provided in this demo can recoginize 9 distinct hand gestures: Left swipe, Right swipe, Up swipe, Down swipe, Clockwise twirl, Counterclockwise twirl, On gesture, Off gesture, and Shine gesture. This demo is compatible with xWR6843ISK-ODS or xWR6843AOP EVMs.

NOTE: This demo is compatible with both the xWR6443 and xWR6843, as it only uses the on-chip Hardware FFT acclerator (HWA) and does not utilize the on-chip c674x DSP. While this demo is intended for the xWR6443 device, a xWR6843 device can be used for emulation.

Requirements

Hardware Requirements

Item	Details
Device	xWR6843ISK-ODS ES2.0 Antenna Module (https://www.ti.com/tool/IWR6843ISK-ODS) or xWR6843AOP ES2.0 Antenna Module (https://www.ti.com/tool/IWR6843AOPEVM)
MMWAVEICBOOST Carrier Board	OPTIONAL: MMWAVEICBOOST Carrier Board (http://www.ti.com/tool/MMWAVEICBOOST) for CCS based development and debugging
	Note: The rest of this document will refer to the above board combination as **EVM** .
Computer	PC with Windows 10. If a laptop is used, please use the 'High Performance' power plan in Windows.
Micro USB Cable	
Power Supply	5V, >3.0A with 2.1-mm barrel jack (center positive). The power supply can be wall adapter style or a battery pack with a USB to barrel jack cable.

Note: Both AWR6843ISK-ODS and IWR6843ISK-ODS are supported and can be used interchangeably. Please consult the respective datasheets for details on the differences between the devices.

Software Requirements

Tool	Version	Download Link
TI mmWave SDK	3.5.0.x	TI mmWave SDK 3.5.0.x (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_05_00_04/index_FDS.html) and all the related tools are required to be installed as specified in the mmWave SDK release notes (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_05_00_04/exports/mmwave_sdk_release_notes.pdf)
Applicable mmWave Toolbox	Latest	Download and install the toolbox. Look in the upper right corner for the download icon or hover next to the name of the toolbox to bring up the download menu.
Uniflash	Latest	Uniflash tool is used for flashing TI mmWave Radar devices. Download offline tool (http://www.ti.com/tool/UNIFLASH) or use the Cloud version (https://dev.ti.com/uniflash/#!/)

Getting familiar with the device

> **Run Out of Box Demo**
>
> Before continuing with this lab, users should first run the out of box demo for the EVM. This will enable users to gain familiarity with the sensor's capabilities as well as the various tools used across all labs in the mmWave Toolbox.

Quickstart

1. Configure the EVM for Flashing Mode

- Follow the instructions for Hardware Setup of Flashing Mode (mmwave_industrial_toolbox\docs\hardware_guides\evm_setup_operational_modes.html)

2. Flash the EVM using Uniflash

Flash the binary listed below using UniFlash. Follow the instructions for using UniFlash (mmwave_industrial_toolbox\docs\software_guides\using_uniflash_with_mmwave.html)

BIN Image Name	Location
9_gesture_6443.bin	`<MMWAVE_TOOLBOX_INSTALL_DIR>\labs\gesture_recognition\9_gesture\prebuilt_binaries\9_gesture_6443.bin`

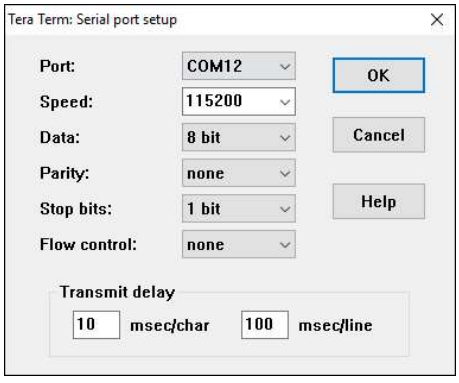
3. Configure the EVM for Functional Mode

- Follow the instructions for Hardware Setup of of Functional Mode (mmwave_industrial_toolbox\docs\hardware_guides\evm_setup_operational_modes.html)

4. Run the Lab

1. GUI Setup

- Open a TeraTerm Instance for the User UART port as described below:
  - For Radar EVM only
    - USER UART = Silicon Labs Dual CP2105 USB to UART Bridge: Enhanced COM Port
  - For MMWAVEICBOOST + Radar EVM
    - USER UART = XDS110 Class Application/User UART
- **TeraTerm (User UART):** Go to **Setup** → **Serial Port** dialog and enter the COM Port number for the User UART Port and select the baud rate and other settings as shown below and press OK.



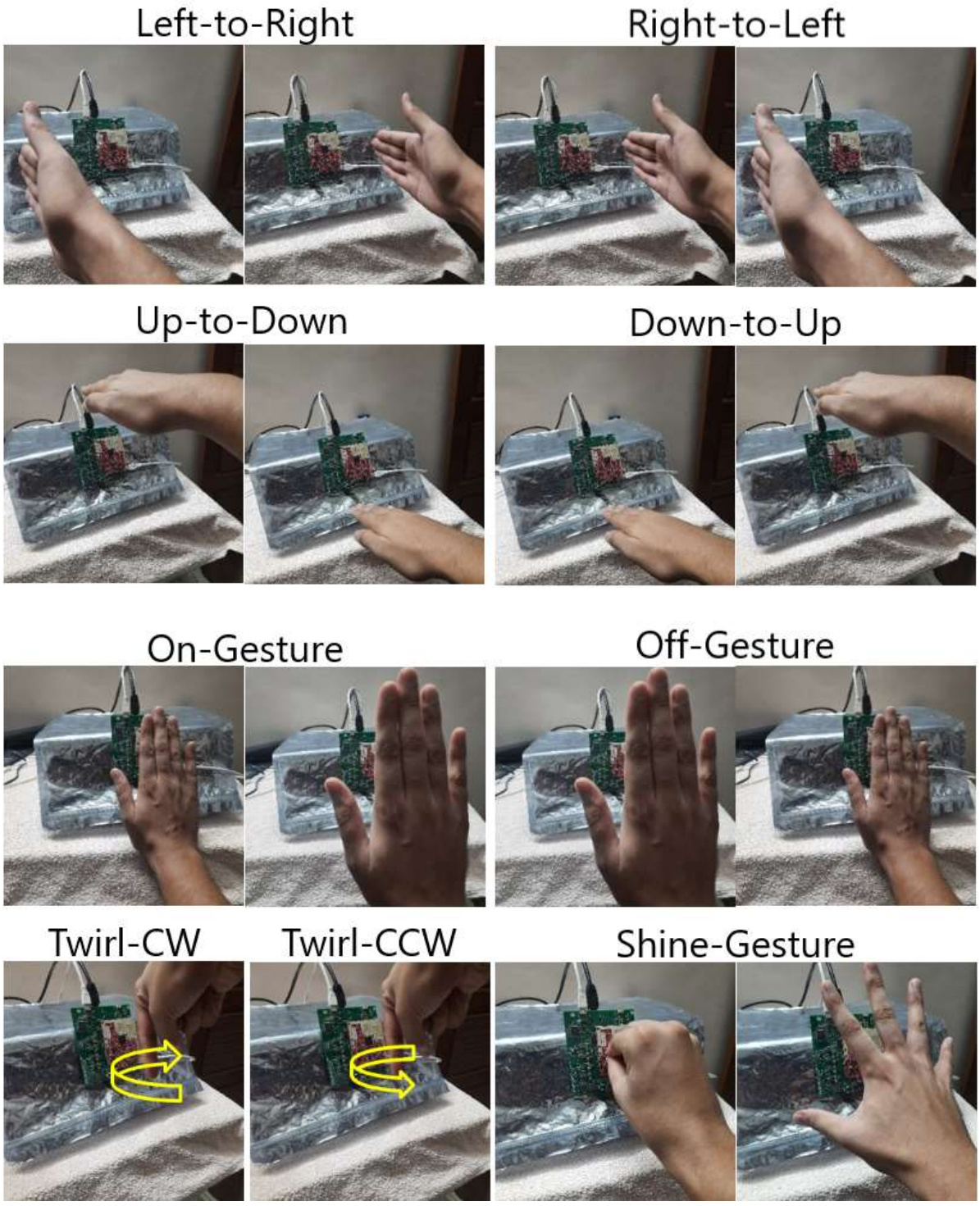
- Press Enter on the Control (User) UART terminal. You should see the **mmWave:/>** prompt which indicates that the demo started correctly. Please note that the gesture demo firmware auto-configures the sensor with pre-programmed chirp configuration at startup.
- The detected gesture output will be displayed on the User UART terminal. This demo also outputs the extracted features used for the nerual network inference as well as the raw neural network output probabilities for each gesture in a TLV structure to the Data UART port.

Expected Format of Data Being Received		
Header		
8 Bytes	-	Magic Word
4 Bytes	-	SDK version
4 Bytes	-	Total packet length
4 Bytes	-	Platform
4 Bytes	-	Frame no.
4 Bytes	-	Time in CPU cycles (R4F)
4 Bytes	-	numDetected Objects
4 Bytes	-	no. of TLV
4 Bytes	-	sub-frame no.
Gesture Features TLV		
4 Bytes	-	TLV Length
4 Bytes	-	TLV Type
4 Bytes	-	Weighted Doppler
4 Bytes	-	Weighted Positve Doppler
4 Bytes	-	Weighted Negative Doppler
4 Bytes	-	Weighted Range
4 Bytes	-	Number of Points > threshold
4 Bytes	-	Weighted Azimuth Mean
4 Bytes	-	Weighted Elevation Mean
4 Bytes	-	Azimuth Doppler Correlation
4 Bytes	-	Weighted Azimuth Dispersion
4 Bytes	-	Weighted Elevation Dispersion
ANN Output Probabilities TLV		
4 Bytes	-	TLV Length
4 Bytes	-	TLV Type
4 Bytes	-	Probability of No Gesture
4 Bytes	-	Probability of Gesture 1 (L2R)
4 Bytes	-	Probability of Gesture 2 (R2L)
4 Bytes	-	Probability of Gesture 3 (U2D)
4 Bytes	-	Probability of Gesture 4 (D2U)
4 Bytes	-	Probability of Gesture 5 (CW)
4 Bytes	-	Probability of Gesture 6 (CCW)
4 Bytes	-	Probability of Gesture 7 (OFF)
4 Bytes	-	Probability of Gesture 8 (ON)
4 Bytes	-	Probability of Gesture 9 (SHINE)

2. Running the Demo

This lab demonstrates the following features.

- Detection/Classification of nine gestures (within a range of about 0.3m): **Right to Left Swipe**, **Left to Right Swipe**, **Up to Down Swipe**, **Down to Up Swipe**, **Clockwise Twirl**, **Counterclockwise Twirl**, **On Gesture**, **Off Gesture**, **Shine Gesture**.



- Perform the above gestures with your hand in front of sensor. The User UART terminal should show the corresponding output.

  This concludes the Quickstart Section

Developer's Guide

1. Software Requirements

Tool	Version	Download Link
Demo source code	Latest	Available on on my secure software
TI mmWave SDK	3.5.x.x	TI mmWave SDK (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_05_00_04/index_FDS.html) and all the related tools are required to be installed as specified in the mmWave SDK release notes (http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/03_05_00_04/exports/mmwave_sdk_release_notes.pdf)
Code Composer Studio	Latest	Code Composer Studio (https://www.ti.com/tool/CCSTUDIO)

2. Import Lab Project

To import the source code into your CCS workspace, a CCS project is provided in the lab at the path given below.

- Start CCS and configure workspace as desired.
- Import the project specified below to CCS. `<DEMO_INSTALL_DIR>\target_code\9_gesture_6443.projectspec`
- Verify that the import occurred without error by checking that `9_gesture_6443` shows up in the project explorer

> **Error during Import to IDE**
>
> If an error occurs, check that the software dependencies listed above have been installed. Errors will occur if necessary files are not installed in the correct location for importing.

3. Build the Lab

- Select the **9_gesture_6443** in **Project Explorer** so that it is highlighted. Right click on the project and select **Rebuild Project**. The project will then build.
- On successful build, the following should appear:
  - In `<PROJECT_WORKSPACE_DIR>\9_gesture_6443 → Debug`
    - **9_gesture_6443.xer4f** (this is the Cortex R4F binary used for CCS debug mode)
    - **9_gesture_6443.bin** (this is the flashable binary used for deployment mode)

Selecting Rebuild instead of Build ensures that the project is always re-compiled. This is especially important in case the previous build failed with errors.

> **Build Fails with Errors**
>
> If the build fails with errors, please ensure that all the software requirements are installed as listed above and in the mmWave SDK release notes.

> **Note**
>
> As mentioned in the Quickstart section, pre-built binary files, both debug and deployment binaries are provided in the pre-compiled directory of the lab.

4. Execute the Lab

There are two ways to execute the compiled code on the EVM:

- **Deployment mode**: the EVM boots from flash and starts running the bin image at power-up
  - Using Uniflash, flash the **9_gesture_6443.bin** found at `<PROJECT_WORKSPACE_DIR>\9_gesture_6443\Debug`
  - The same procedure for flashing can be used as detailed in the Flash the EVM section.
- **Debug mode**: This mode is used for downloading and running the executable from CCS. This mode enables JTAG connection with CCS while the lab is running and is useful during development and debugging.
  - Follow the CCS Debug Mode Guide (mmwave_industrial_toolbox/docs/software_guides/using_ccs_debug.html)

Need More Help?

- Search for your issue or post a new question on the mmWave E2E forums (https://e2e.ti.com/support/sensor/mmwave_sensors/f/1023)
- See the SDK for more documentation on various algorithms used in this demo. Start at `<mmWave_sdk_install_dir>/docs/mmwave_sdk_module_documentation.html`