Design Guide: TIDA-010022

People counting and tracking using mmWave radar sensor with sub-1 GHz reference design

Description

This design demonstrates the use of the IWR6843 device, a single-chip, mmWave radar sensor from TI with integrated DSP, for an indoor and outdoor people-counting application along with sub-1 GHz wireless communication. This reference design uses the MMWAVEICBOOST and IWR6843ISK evaluation modules (EVMs) together with the LAUNCHXL-CC1352R1 wireless MCU LaunchPad™. This solution is shown to localize people out to 6 m (short-range configuration) and 14 m (long-range configuration).

Resources

TIDA-010022	Design Folder
IWR6843	Product Folder
CC1352R	Product Folder
MWAVEICBOOST	Tool Folder
IWR6843ISK	Tool Folder
LAUNCHXL-CC1352R1	Tool Folder

Ask our TI E2E™ support experts

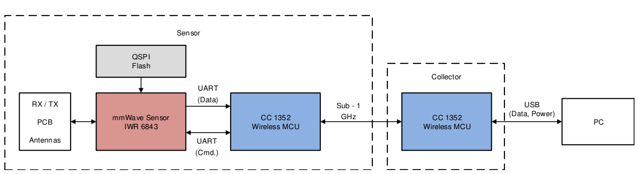
Features

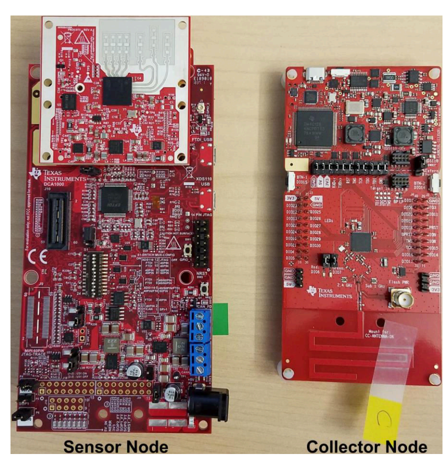
- Contains demonstration hardware and software to use the IWR6843 device and mmWave radar sensor for people counting and tracking
- Long-range, low power sub-1 GHz wireless connectivity using the CC1352 device
- IEEE 802.15.4e or IEEE 802.15.4g stack with collector and sensor application
- mmWave technology provides the range, velocity, and angle information – ideal for environmental effects
- Azimuth field of view of 120° across a distance of 6 m, that can extend to 14 m with different chip configuration
- Examples of implementation of static clutter and group- tracking algorithms

Applications

- Occupancy detection
- Motion detector
- Automated doors and gates
- IP network camera
- Smoke and heat detector
- Lighting sensors

Design Images





1 System Description

Industrial and building automation security systems can use radar to detect and track humans and other objects. In a security system, mmWave technology provides range, velocity, and angle information that is immune to environmental effects. The CC1352 wireless radio microcontroller provides extended long-range connection in spread-out areas to monitor and track activity with ultra-low power consumption.

Human monitoring has become an important area of exploration, due to its potential for understanding people's count, activities, intents, and health issues. The ability to continuously and consistently monitor human motion is an important function in numerous applications, including surveillance, control, and analysis. Accuracy and precision play an important role in these applications. While sensors such as passive infra-red (PIR) and time of flight (TOF) are in use, they suffer from limitations in accuracy, false alarms, and environmental changes such as darkness, brightness, and smoke.

Radars allow an accurate measurement of distances, relative velocities of people, and other objects. They are relatively immune to environmental conditions such as the effects of rain, dust, or smoke. Additionally, they can work in complete darkness or in bright daylight. They are therefore useful for building automation applications such as people counting, motion detection, IP network cameras, and safety guards.

Sub-1 GHz wireless radio microcontrollers are becoming a popular choice for many applications worldwide. These devices work on the ISM spectrum bands below 1 GHz, typically in the 769 MHz to 935 MHz, 315 MHz, and the 468 MHz frequency range, and with the emerging IoT market moving into industrial applications, sub-1 GHz wireless radio communication is becoming the standard for these applications due to three main reasons: range, low power consumption, and interference to avoid problems associated with high traffic bands.

This design guide addresses component selection, design theory, and the testing results of this TI Design system. The scope of this design guide gives system designers a head-start in integrating TI's mmWave sensor and SimpleLink™ multi-band wireless MCU.

The following subsections describe the various blocks within the TI Design system and what characteristics are most critical to best implement the corresponding function.

1.1 Key System Specifications

Table 1-1. Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Input power source	5 V	Section 2.2.2
Sensor type	mmWave Radar	Section 2.2.1
System power consumption	3.6 W	Section 3.2.2.2
Radar field of view	120° horizontal, 30° vertical	Section 3.2.2.1
Radar maximum range	6 m for short range configuration and 14 m for long range configuration	Section 3.2.2.1
Radar range resolution	4.9 cm for short range configuration and 12 cm for long range configuration	Section 3.2.2.1
Radar maximum velocity	18.64 km/h for short range configuration and 18.9 km/h for long range configuration	Section 3.2.2.1
Radar velocity resolution	0.297 km/h for both short and long range configuration	Section 3.2.2.1
Frame rate	Up to 20 frames per second	Section 2.2.4.3
Maximum number of objects	Up to 20 objects	Section 2.2.4.3
Radio transmitter frequency	915 MHz	Section 2.2.3
Radio transmit power	12 dBm	Section 2.2.3
Radio transmission range	70 m	Section 3.2.2.3
Network type	Star	Section 2.2.3

2 System Overview

2.1 Block Diagram

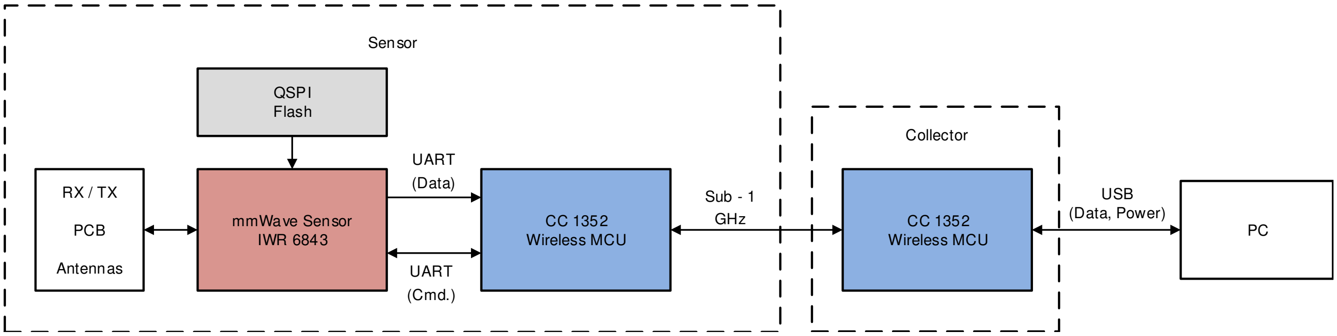


Figure 2-1. TIDA-010022 Block Diagram

2.2 Design Considerations

The people counting and tracking using mmWave radar sensor with Sub-1 GHz reference design counts the number of objects within its field of view and sends the data over the air to a collector node.

2.2.1 mmWave Sensor

2.2.1.1 mmWave Sensor People Counting

The implementation of the people-counting application example in the signal-processing chain consists of the following blocks, implemented as DSP code executing on the C674x core in the IWR6843. The tracking module runs on the ARM® Cortex®-R4F processor.

- Range Processing:
  - For each antenna, 1D windowing, and 1D fast Fourier transform (FFT)
  - Range processing is interleaved with the active chirp time of the frame
- Capon beam forming:
  - Static clutter removal
  - Co-variance matrix generation, inverse-angle spectrum generation, and integration is performed
  - Outputs range-angle heat map
- CFAR detection algorithm:
  - Two-pass, constant false-alarm rate
  - First pass cell averaging smallest of CFAR-CASO in the range domain, confirmed by second pass cell averaging smallest of CFAR-CASO in the angle domain, to find detection points.
- Doppler estimation:
  - For each detected [range, azimuth] pair from the detection module, estimate the Doppler by filtering the range bin using Capon beam-weights, and then run a peak search over the FFT of the filtered range bin.
- Tracking:
  - Perform target localization, and report the results.
  - Output of the tracker is a set of trackable objects with certain properties like position, velocity, physical dimensions, and point density

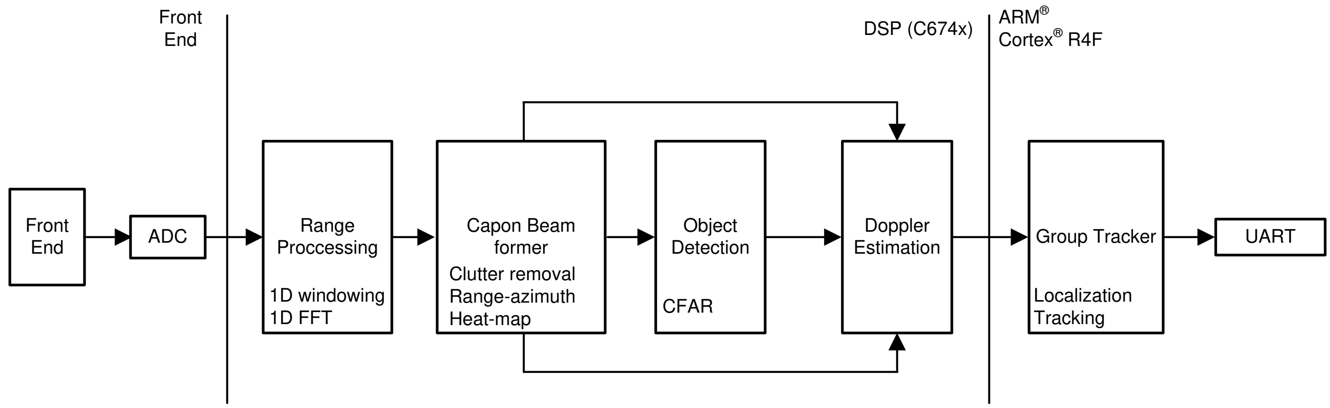


Figure 2-2. People-Counting Application Block Diagram

For more detailed information on the people-counting application, see *People Tracking and Counting Reference Design Using mmWave Radar Sensor*.

2.2.1.2 IWR6843 UART Communication

The CC1352 device configures the IWR6843 device via a command line interface (CLI) UART port. After the configuration is completed, the IWR6843 starts sending out radar data continuously back to CC1352 through a separate data UART port. Then, the CC1352 parses the data from the IWR6843 and packetizes and sends the data package over the air to the collector node constantly. Figure 2-3 is a block diagram of the IWR6843 UART communication.

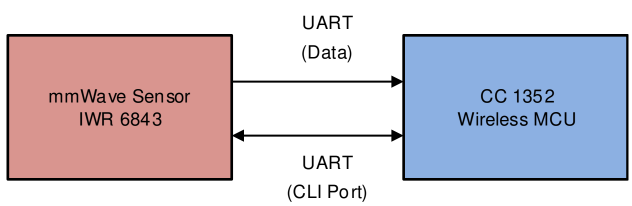


Figure 2-3. IWR6843 UART Communication

2.2.1.2.1 IWR6843 Frame Output

The IWR6843 outputs one frame every frame period. The frame has a fixed header, followed by a variable number of segments in tag, length, value (TLV) format. Each TLV has a fixed header, followed by a variable-size payload. Byte order is little endian. Figure 2-4 illustrates the IWR6843 output frame format.

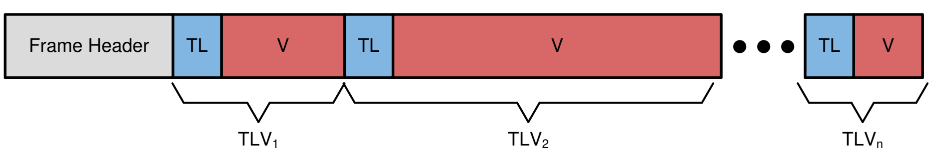


Figure 2-4. IWR6843 Output Frame Format

2.2.1.2.2 IWR6843 Frame Header

The frame header is a fixed size (52 bytes) and has the following structure (see *Figure 2-5* using MATLAB® notation, with name, type, and length in bytes). The header is designed to self-describe the content, and allow the user application to operate in a lossy environment.

```
frameHeaderStructType = struct(...
    'sync',             {'uint64', 8}, ... % Sync Pattern
    'version',          {'uint32', 4}, ... % mmWaveSDK version
    'platform',         {'uint32', 4}, ... % 0xA1642 or 0xA1443
    'timestamp',        {'uint32', 4}, ... % 600MHz free running clocks
    'packetLength',     {'uint32', 4}, ... % In bytes, including header
    'frameNumber',      {'uint32', 4}, ... % Starting from 1
    'subframeNumber',   {'uint32', 4}, ...
    'chirpMargin',      {'uint32', 4}, ... % Chirp Processing margin, in us
    'frameMargin',      {'uint32', 4}, ... % Frame Processing margin, in us
    'uartSentTime' ,    {'uint32', 4}, ... % Time spent to send data, in us
    'trackProcessTime', {'uint32', 4}, ... % Tracking Processing time, in us
    'numTLVs' ,         {'uint16', 2}, ... % Number of TLVs in thins frame
    'checksum',         {'uint16', 2});    % Header checksum
```

Figure 2-5. IWR6843 Frame Header

2.2.1.2.3 TLV Elements

Each TLV has a fixed header (8 bytes) followed by a TLV-specific payload (see Figure 2-6).

```
tlvHeaderStruct = struct(...
    'type',             {'uint32', 4}, ... % TLV object Type
    'length',           {'uint32', 4});    % TLV object Length, in bytes, including TLV header
```

Figure 2-6. TLV Header

Three TLVs are supported at this time, as follows:

- Point Cloud TLV
  - Type = POINT_CLOUD_2D
  - Length = sizeof (tlvHeaderStruct) + sizeof (pointStruct2D) × numberOfPoints

```
% Point Cloud TLV object consists of an array of points.
% Each point has a structure defined below
pointStruct2D = struct(...
    'range',            {'float', 4}, ... % Range, in m
    'azimuth',          {'float', 4}, ... % Angle, in rad
    'doppler',          {'float', 4}, ... % Doppler, in m/s
    'snr',              {'float', 4});    % SNR, ratio
```

Figure 2-7. Point Cloud TLV

- Target List TLV
  - Type = TARGET_LIST
  - – Length = sizeof (tlvHeaderStruct) + sizeof (targetStruct) × numberOfTargets

```
% Target List TLV object consists of an array of targets.
% Each target has a structure define below
targetStruct2D = struct(...
    'tid',              {'uint32', 4}, ... % Track ID
    'posX',             {'float', 4}, ... % Target position in X dimension, m
    'posY',             {'float', 4}, ... % Target position in Y dimension, m
    'velX',             {'float', 4}, ... % Target velocity in X dimension, m/s
    'velY',             {'float', 4}, ... % Target velocity in Y dimension, m/s
    'accX',             {'float', 4}, ... % Target acceleration in X dimension, m/s2
    'accY',             {'float', 4}, ... % Target acceleration in Y dimension, m/s
    'EC',               {'float', 9*4}, ... % Tracking error covariance matrix, [3x3], in
range/angle/doppler coordinates
    'G',                {'float', 4});    % Gating function gain
```

Figure 2-8. Target List TLV

Note

The target index TLV received in the N-th frame indices the point cloud in $(N–1)^{th}$ frame.

- Target Index TLV
  - Type = TARGET_INDEX
  - Length = sizeof (tlvHeaderStruct) + numberOfPoints
  - Payload is a byte array, each byte represents a tracking ID

Note

The track ID is a byte. Values 0 to 249 are supported. Values 250 to 255 are reserved.

2.2.2 Power Supply Design

To design a proper power supply for the system, consider the following specs:

1. The IWR6843 chip requires multiple voltage rails: 3.3 V, 1.8 V, 1.24 V, 1.0 V
2. The CC1352 chip requires a 3.3-V rail
3. The main power supply to the sensor node is 5 V

In this design, the LP8770 PMIC on the IWR6843ISK board is used to generate the voltage rails for the IWR6843. The IWR6843ISK receives 3.3 V from the MMWAVEICBOOST board.

The CC1352 device used in the sensor node is powered directly from the 3.3-V rail (BRD_3V3) generated by the LM53635-Q1 DC-DC converter on the MMWAVEICBOOST board. This 3.3 V rail provides power to various components (muxes, serial interfaces, debug, etc.) on the MMWAVEICBOOST board, including the IWR6843ISK board.

2.2.3 Wireless Network Design

This design leverages TI's SimpleLink TI-15.4 Stack to implement a star network. The TI-15.4 Stack implements the standard IEEE 802.15.4e, IEEE 802.15.4g specification and is designed to run on SimpleLink CC1352 ultra-low-power MCUs. The TI 15.4 Stack supports different modes including frequency hopping where network devices hop on different frequencies, beacon mode where the network coordinator transmits periodic beacons to indicate its presence, and non-beacon mode where the coordinator does not send out periodic beacons.

The network consists of a central device, or collector, and one or more sensor nodes. The collector performs the functions of a network coordinator (starts the network, allows devices to join the network) and receives data from the sensor nodes. The collector then sends periodic tracking request messages (to which it expects tracking response messages) to determine whether or not the sensor nodes are alive in the network.

Sensor nodes join the network started by the collector. They send sensor data reports to the collector and respond to the tracking messages sent by the collector. The sensor nodes also poll for messages from the collector allowing for two-way communication.

This TI Design together with a CC1352 LaunchPad implements the function of the sensor node and collector node. The TI-15.4 Stack is configured for operation in the 915-MHz frequency band (frequency hopping disabled), non-beacon mode enabled, and transmit power set to 12 dBm.

2.2.4 CC1352 Software Design

The collector and sensor software used in this TI design is a modified version of the collector and sensor example code provided with the TI 15.4 Stack. The TI 15.4 Stack and example codes run as state machines waiting for events. To make these example codes work, two modifications were necessary: A new event was added to the collector and sensor examples and added an additional task running in parallel in charge of triggering said event. The new event is triggered when new mmWave data is received from the mmWave. The following section provides is a description of the event and task.

2.2.4.1 Collector Node

Figure 2-9 illustrates the collector node software flowchart.

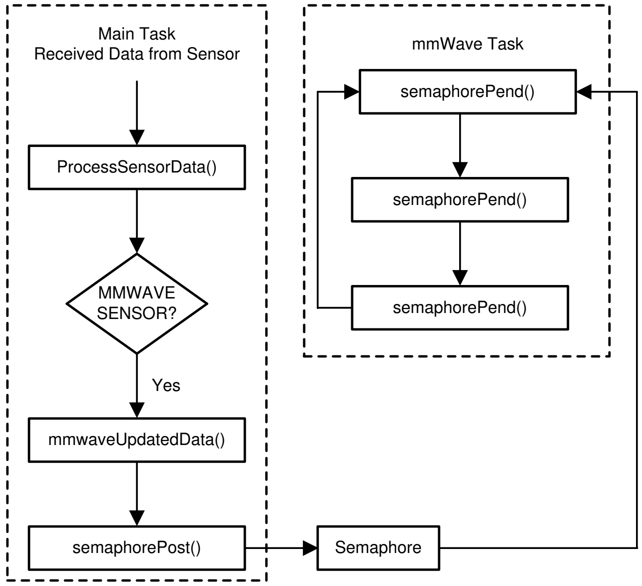


Figure 2-9. Collector Node Software Flowchart

2.2.4.2 Sensor Node

Figure 2-10 shows the sensor node software flowchart.

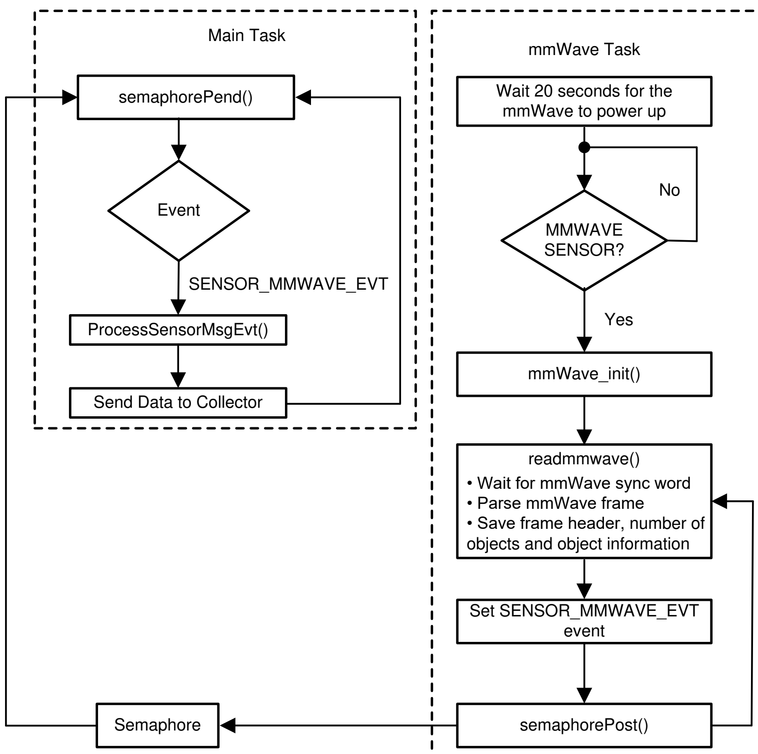


Figure 2-10. Sensor Node Software Flowchart

Each sensor packet includes 4 bytes of the frame number, 4 bytes of the number of objects detected, and 12 bytes of tracking information for each object.

The tracking information for each object is 12 bytes which include 4 bytes track ID, 4 bytes target position in the X dimension, and 4 bytes target position in the Y dimension.

```
/*!
 mmWave Object Field
 */
typedef struct _Smsgs_objtrackingfield_t
{
    uint32_t tid;
    uint32_t posx;
    uint32_t posy;
} Smsgs_objtrackingfield_t;
/*!
 mmWave Sensor Field
 */
typedef struct _Smsgs_mmwavesensorfield_t
{
    /*! mmWave frame number */
    uint32_t frame_number;
    /*! Number of objects detected */
    uint16_t num_objs;
    /*! Tracking information for each object */
    Smsgs_objtrackingfield_t objInfo[SMSGS_SENSOR_MMWAVE_MAX_OBJ];
} Smsgs_mmwaveSensorField_t;
```

2.2.4.3 mmWave Sensor Configuration Command List Modification

The short range configuration is used by default in this design. Also lowpower and pointCloudEn commands are changed compared to the default command list:

1. Changed "lowpower 0 1\r" to "lowpower 0 0\r" which disables the low power mode since ES1.0 does not use low power mode
2. Added additional command "pointCloudEn 0\r" to disable the point cloud to reduce the burden of the CC1352 device by providing a smaller frame for it to read

```
const char *CommandList[] = {"dfeDataOutputMode 1\r",
                    "channelCfg 15 5 0\r",
                    "adcCfg 2 1\r",
                    "adcbufCfg 0 1 1 1\r",
                    "profileCfg 0 60.6 30 10 62 0 0 53 1 128 2500 0 0 30\r",
                    "chirpCfg 0 0 0 0 0 0 0 1\r",
                    "chirpCfg 1 1 0 0 0 0 0 4\r",
                    "frameCfg 0 1 128 0 50 1 0\r",
                    "lowPower 0 1\r",
                    "guiMonitor 1 1 0 0\r",
                    "cfarCfg 6 4 4 4 4 16 16 4 4 50 62 0\r",
                    "doaCfg 600 1875 30 1 1 0\r",
                    "SceneryParam -6 6 0.5 6\r",
                    "GatingParam 4 3 2 0\r",
                    "StateParam 10 5 100 100 5\r",
                    "AllocationParam 250 250 0.25 10 1 2\r",
                    "AccelerationParam 1 1 1\r",
                    "PointCloudEn 0\r",
                    "trackingCfg 1 2 250 20 52 82 50 90\r",
                    "sensorStart\r"
};
```

For more detailed information on how to configure for long range mode, see *People Tracking and Counting Reference Design Using mmWave Radar Sensor*.

2.3 Highlighted Products

The TIDA-010022 design features the following devices and evaluation platforms (EVMs):

1. IWR6843: Single-chip 60- to 64-GHz mmWave sensor
2. CC1352: SimpleLink high-performance dual-band wireless MCU
3. IWR6843ISK and MMWAVEICBOOST: mmWave EVMs
4. LAUNCHXL-CC1352R1: SimpleLink™ Multi-Band CC1352R Wireless MCU LaunchPad™ Development Kit

For more information on these products, see their data sheet or product folders on www.ti.com.

2.3.1 IWR6843: Single-Chip 60- to 64-GHz mmWave Sensor

The IWR1642 device includes the entire Millimeter Wave blocks and analog baseband signal chain for two transmitters and four receivers, as well as a customer-programmable MCU and DSP. The IWR1642 handles all people counting and tracking functions.

In this reference design the IWR1642 device is paired with a CC1352R SimpleLink dual-band wirelss MCU to enable wireless communication of people counting and tracking data.

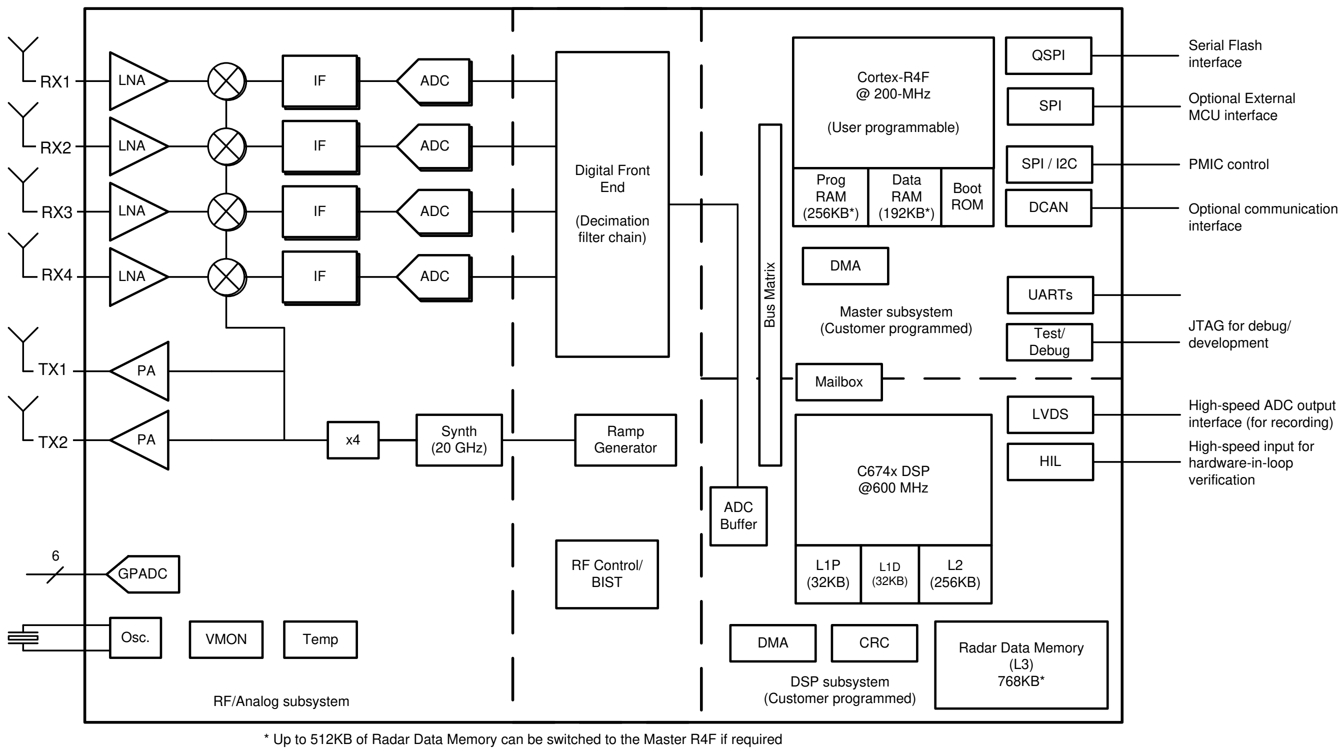


Figure 2-11. IWR6843 Block Diagram

2.3.2 CC1352R: SimpleLink™ High-Performance Dual-Band Wireless MCU

The CC1352R device is a multiprotocol Sub-1 and 2.4-GHz wireless MCU targeting Wireless M-Bus, IEEE 802.15.4g, IPv6-enabled smart objects (6LoWPAN), Thread, Zigbee ® , KNX RF, Wi-SUN® , Bluetooth® 5 Low Energy, and proprietary systems.

The CC1352R device is a member of the CC26xx and CC13xx family of cost-effective, ultra-low power, 2.4-GHz and Sub-1 GHz RF devices. Very low active RF and microcontroller (MCU) current, in addition to sub- μ A sleep current with up to 80KB of RAM retention, provide excellent battery lifetime and allow operation on small coin-cell batteries and in energy-harvesting applications.

The CC1352R device combines a flexible, very low-power RF transceiver with a powerful 48-MHz Arm® Cortex® -M4F CPU in a platform supporting multiple physical layers and RF standards. A dedicated Radio Controller (Arm® Cortex® -M0) handles low-level RF protocol commands that are stored in ROM or RAM, thus ensuring ultra-low power and great flexibility. The low power consumption of the CC1352R device does not come at the expense of RF performance; the CC1352R device has excellent sensitivity and robustness (selectivity and blocking) performance.

The CC1352R device is a highly integrated, true single-chip solution incorporating a complete RF system and an on-chip DC/DC converter.

The CC1352R device is part of the SimpleLink™ microcontroller (MCU) platform, which consists of Wi-Fi® , Bluetooth® Low Energy, Thread, Zigbee, Sub-1 GHz MCUs, and host MCUs, which all share a common, easy-to-use development environment with a single core software development kit (SDK) and rich tool set. A one-time integration of the SimpleLink platform enables you to add any combination of the portfolio 's devices into your design, allowing 100 percent code reuse when your design requirements change. For more information, visit ti.com/simplelink.

In this reference design, the CC1352R is used to connect to initialize the IWR1642, connect to a wireless sensor network, and transmit all people counting and tracking data to the network.

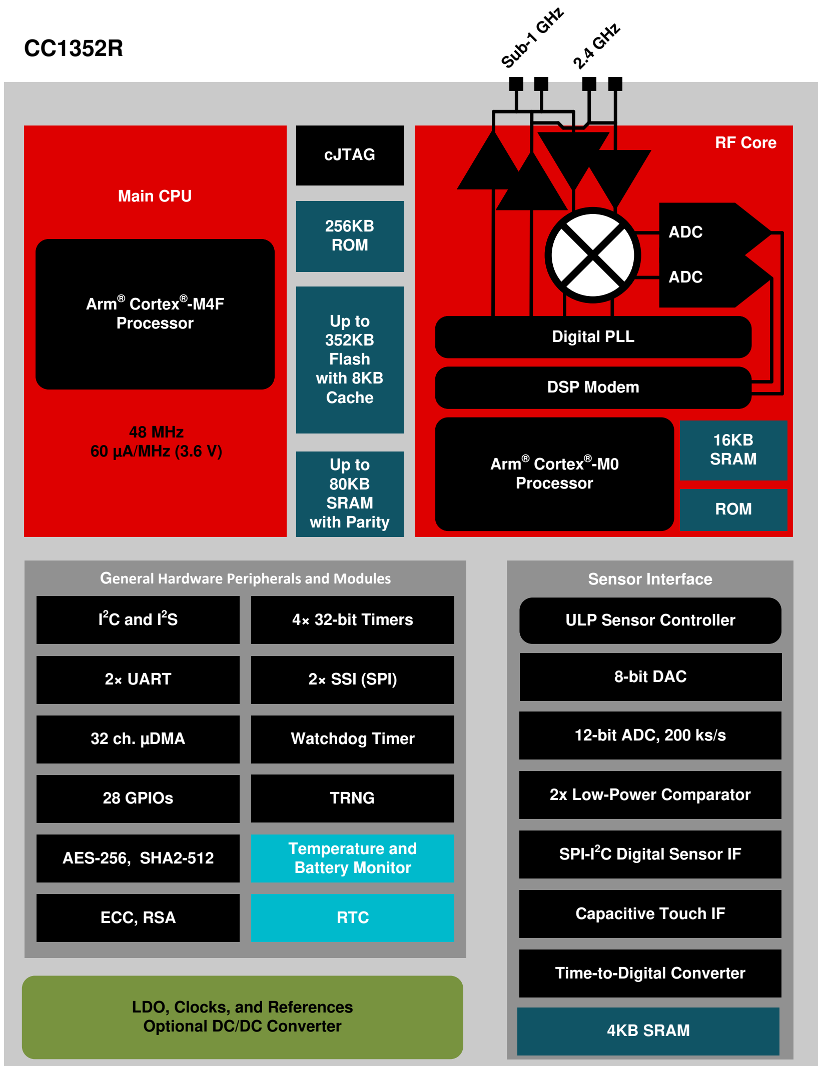


Figure 2-12. CC1352R Block Diagram

2.3.3 IWR6843ISK and MMWAVEICBOOST: mmWave EVMs

IWR6843ISK and MMWAVEICBOOST are part of the mmWave EVMs hardware platforms. The IWR6843 Industrial starter kit (IWR6843ISK) from Texas Instruments is an easy-to-use evaluation module for the IWR6843 mmwave sensing device and this board contains 60GHz mmwave Radar transceiver in which antennas are etched and act as Radar front-end board. The MMWAVEICBOOST is an add-on board used with TIs mmWave sensor used in all Starter Kits to provide more interfaces and PC connectivity to the mmWave sensors. The MMWAVEICBOOST board provides an interface for the mmWave Studio tool to configure the Radar device and capture the raw analog-to-digital converter (ADC) data using a capture board such as DCA1000 evaluation module (EVM). IWR6843ISK and MMWAVEICBOOST contains everything required to start developing software for on-chip C67x DSP core and low-power ARM R4F controllers. It provides interface to the MSP43xx boards through 40-pin LaunchPad™/ BoosterPack™ connectors.

The IWR6843ISK device includes the following features:

- 60-pin, high-density (HD) connector for raw analog-to-digital converter (ADC) data over LVDS and trace data capability

- Long range on-board antenna
- Current sensors for all rails
- On-board PMIC

The MMWAVEICBOOST device includes the following features:

- Hosts Starter Kit using two 60-pin high-density (HD) connector for the high-speed ADC data over CSI or LVDS and emulator signals
- FTDI-based JTAG emulation with serial port for programming flash on the Starter Kit
- XDS110-UART based QSPI flash programming
- 60-pin HD connector to interface with the DCA1000 EVM
- Two 20-pin LaunchPad connectors that leverage the ecosystem of the TI standard Launchpad and have all of the digital controls from the Radar chip
- Two onboard controller area network (CAN) transceivers
- On-board PMIC
- 60-pin MIPI HD connector for JTAG trace
- On-board FTDI chip to provide PC interface for serial peripheral interface (SPI), general-purpose input/output (GPIO) controls and universal asynchronous receiver/transmitter (UART) loggers
- On-board current sensors and temperature sensors

This reference design uses the IWR6843ISK and MMWAVEICBOOST to support all IWR6843 functions.

2.3.4 LAUNCHXL-CC1352R1: SimpleLink™ Multi-Band CC1352R Wireless MCU LaunchPad™ Development Kit

The CC1352R LaunchPad (LAUNCHXL-CC1352R1) is a microcontroller development kit for rapid prototyping featuring the CC1352R microcontroller.

- LaunchPad with Sub-1GHz radio for wireless applications with integrated PCB trace antenna
- Broad band antenna supports both 868 MHz ISM band for Europe and 915 MHz ISM band for US with a single board
- On-board emulator gets you started with instant code development in CCS Cloud
- Can be used with both LaunchPad kit and SmartRF™ Studio applications
- Access all I/O signals with the BoosterPack plug-in module connectors
- Compatible with LCD BoosterPack

This reference design uses two CC1352R LaunchPads to implement a sensor-collector network.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

The following sections describe the hardware and software required to get started with the people counting and tracking using the mmWave radar sensor with sub-1 GHz reference design.

3.1.1 Hardware Setup

The TIDA-010022 design requires the following hardware:

- One mmWave sensor carrier card platform (MMWAVEICBOOST)
- One IWR6843 intelligent mmWave sensor standard antenna plug-in module (IWR6843ISK)
- Two CC1352 LaunchPads (LAUNCHXL-CC1352R1)
- One 5 V, 2.5 A or greater DC power supply

Note

Before making any hardware modifications to the IWR6843 boards, flash them by following the steps in Section 3.1.2.

The CC1352 LaunchPad can attach to the MMWAVEICBOOST board after some hardware modifications. Table 3-1 lists the hardware modifications for the MMWAVEICBOOST board.

Table 3-1. MMWAVEICBOOST Hardware Modifications

MODIFICATION	REASON
Remove R91	Disconnects RADAR_HOSTINTR1 pin from LaunchPad
Add R81	Connects RADAR_MSS_LOGGER pin to LaunchPad
Add R122 and add jumper to J13 pin 2-3	Connects 3.3 V rail on MMWAVEICBOOST board to 3.3 V rail on CC1352 LaunchPad
Configure mux as follows: S1.1 OFF: SPI1 (Default) S1.1 OFF: SPI1 (Default) S1.2 ON : 40PIN S1.3 OFF: 40PIN/J16 S1.4 ON: FTDI/4PIN/J16 (Default) S1.5 OFF: 40/60PIN/FTDI S1.6 OFF: 40/60PIN (Default) S1.7 OFF: 40/60PIN S1.8 ON: 40PIN S1.9 OFF: 60PIN S1.10 ON: FTDI S1.11 ON: FTDI S1.12 ON: XDS110 (Default)	Enable 40-pin interface

The components that need to be removed from the MMWAVEICBOOST board are marked by the red squares in Figure 3-1. The components that need to be added are marked in blue squares. No modifications are needed on the IWR6843ISK board.

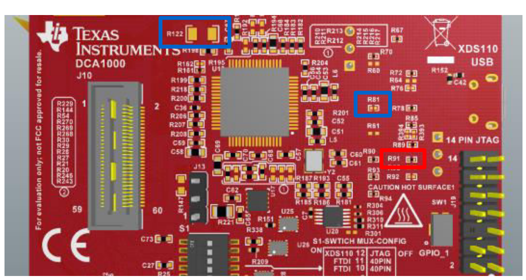


Figure 3-1. MMWAVEICBOOST Front PCB Layout View

Some modifications are needed on the CC1352 LaunchPad that will be used as the sensor node. Remove the following jumpers: "5V", "3V3", "RXD", and "TXD" and switch the XDS110 power jumper to "Extern" Power. Remove the SMA connector for the external antenna (J7 on the schematics) on the CC1352 LaunchPad for it to mate with the MMWAVEICBOOST board (see Figure 3-2).

No hardware modifications are needed on the CC1352 LaunchPad that will be used as the collector node.

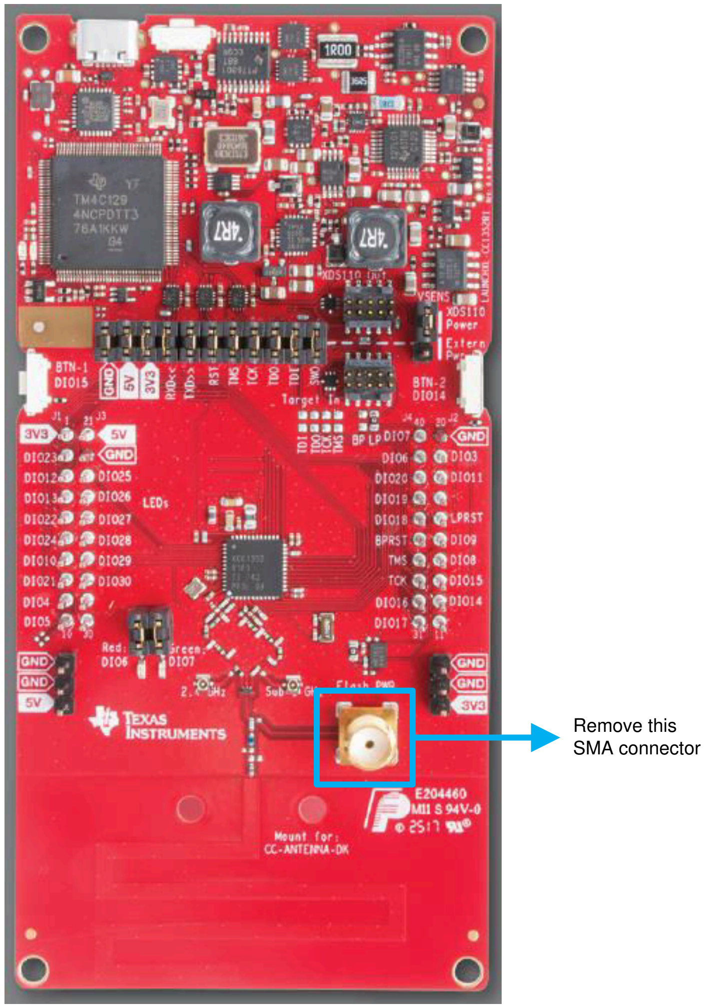


Figure 3-2. SMA Connector in CC1352 to be Removed (Sensor Board)

After all hardware modifications are completed, the IWR6843ISK board, MMWAVEICBOOST board, and one of the CC1352 LaunchPads can be assembled together to form the sensor node.

The sensor node (IWR6843 boards + CC1352) can be powered by connecting the 5-V power supply to the MMWAVEICEBOOST board and the collector node can be powered by connecting the USB power supply.

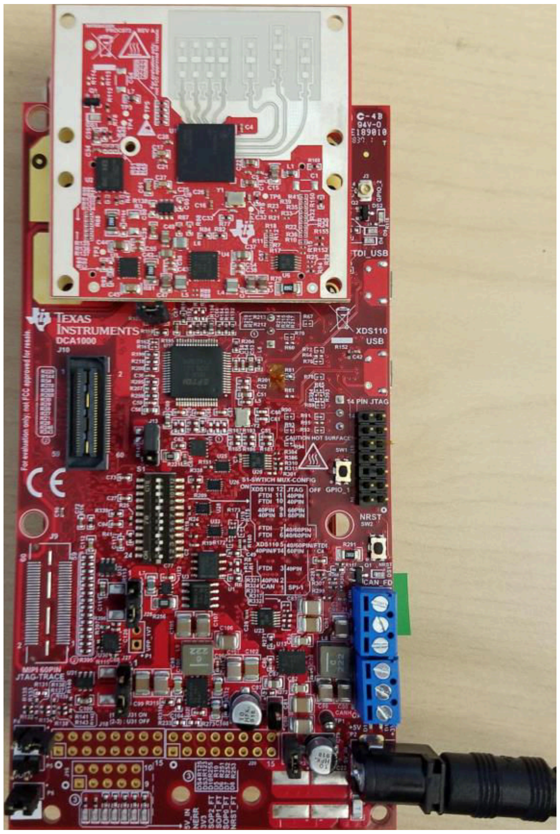


Figure 3-3. Sensor Node (IWR6843 Boards + CC1352) Power Supply

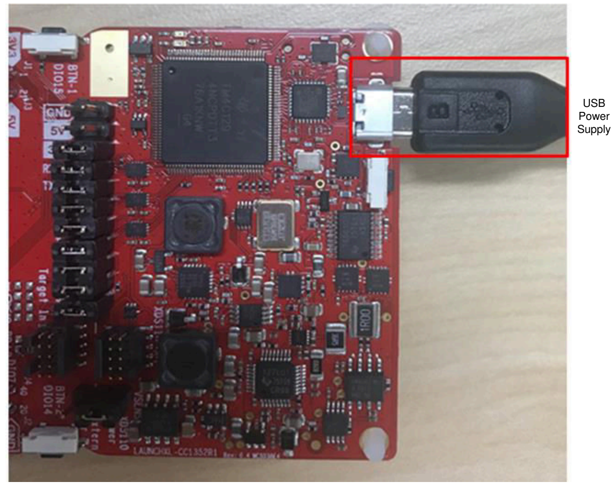


Figure 3-4. Collector Node (CC1352 Only) Power Supply

3.1.2 Software Setup

3.1.2.1 Loading the CC1352 Firmware

Follow these steps to load the firmware on the CC1352 LaunchPads:

1. Download UniFlash from http://www.ti.com/tool/UNIFLASH.
2. Open UniFlash and connect the sensor node and collector node to the PC.

> **Note**
>
> See Section 3.1.1 for details on setting up the IWR6843 boards and CC1352 LaunchPad.

3. Once the CC1352 is plugged into the PC, UniFlash will detect each individual device.

> **Note**
>
> Both the 5-V power supply and a USB cable are needed to program the sensor node, but only the USB cable is needed to program the collector node.

4. Highlight the device to be programmed and click the *Start* button.
5. In the *Program* section, under *Flash Image(s)*, click the *Browse* button.
   a. If loading firmware for the collector, navigate to the TIDA-010022 firmware folder and select the collector_cc1352r1lp.out file.
   b. If loading firmware for the sensor, navigate to the TIDA-010022 firmware folder and select the sensor_cc1352r1lp.out file.
6. After selecting the correct image file, click the *Load Image* button.
7. If the image was loaded successfully, the message "[SUCCESS] Program Load load completed successfully" is displayed.

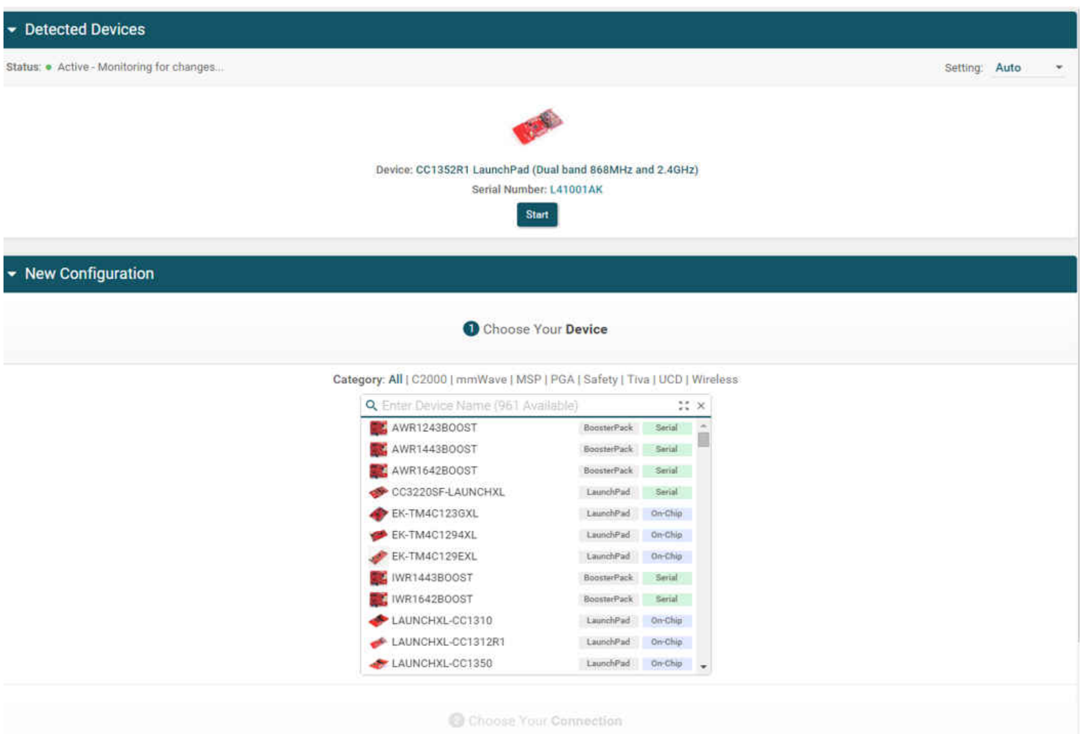


Figure 3-5. Detected Devices

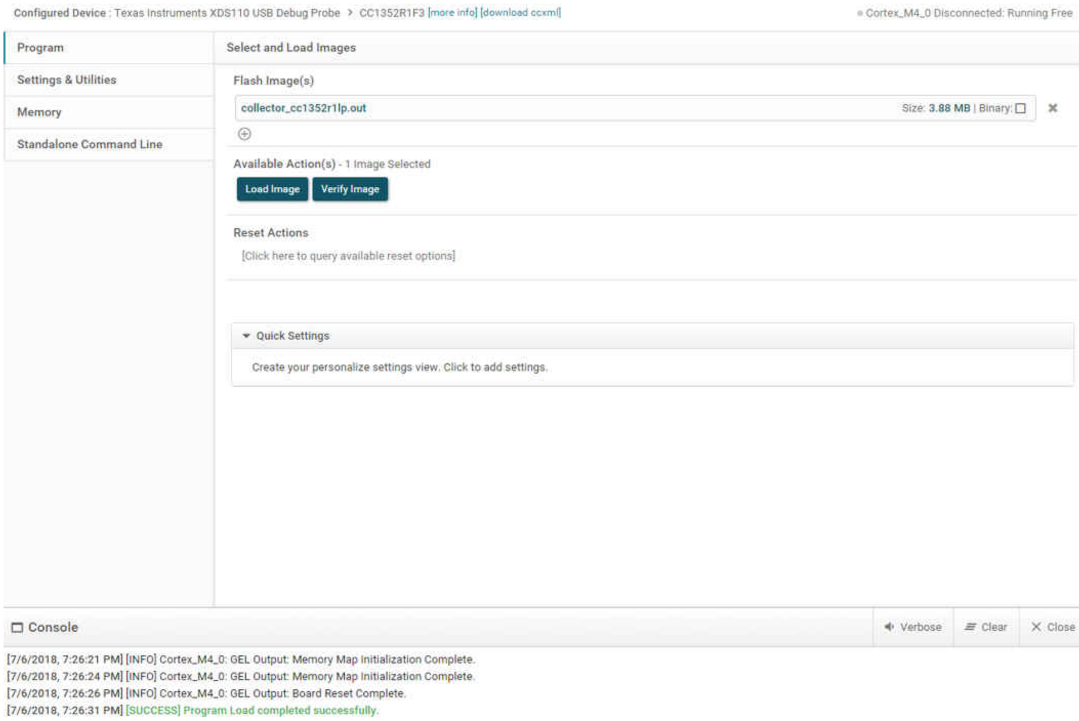


Figure 3-6. Loading Image Successfully

3.1.2.2 Loading the IWR6843 Firmware

This reference design was tested with and includes the binary image for the 68xx People Counting demo included in the mmWave Sensors Industrial Toolbox v. 3.2.0.

Follow these steps to load the IWR6843 firmware:

Note

Make sure the CC1352 LaunchPad is not attached to the MMWAVEICBOOST board.

1. Connect the IWR6843ISK and MMWAVEICBOOST boards together as shown in Figure 3-7.
2. Plug in micro-USB cable and 5-V power supply to mmWave boards as shown in Figure 3-7.

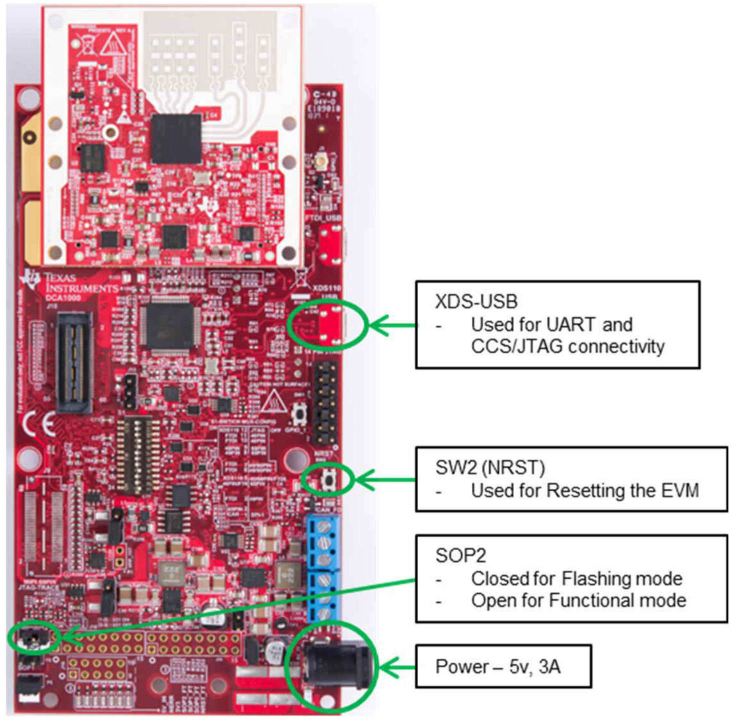


Figure 3-7. IWR6843ISK and MMWAVEICBOOST Boards

3. Connect the mini-USB cable to your PC and check the COM ports in Windows Device Manager. Figure 3-8 shows how the mmWave setup exports two virtual COM ports:
    a. XDS110 Class Application/User UART (COM_{UART}): Used for passing configuration data and firmware to the mmWave boards.
    b. XDS110 Class Auxiliary Data Port (COM_{AUX}): Used to send processed radar data output.

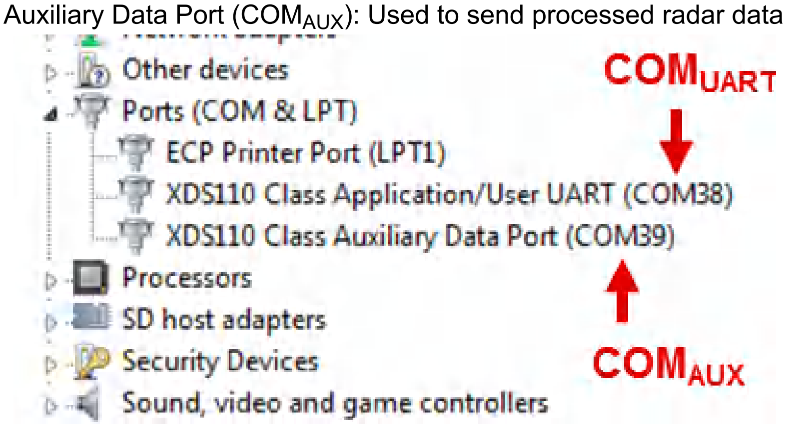


Figure 3-8. COM Port Identification

4. Put the MMWAVEICBOOST in flashing mode by connecting jumpers on SOP0 and SOP2 as shown in Figure 3-9. Then reset the MMWAVEICBOOST with SW2.

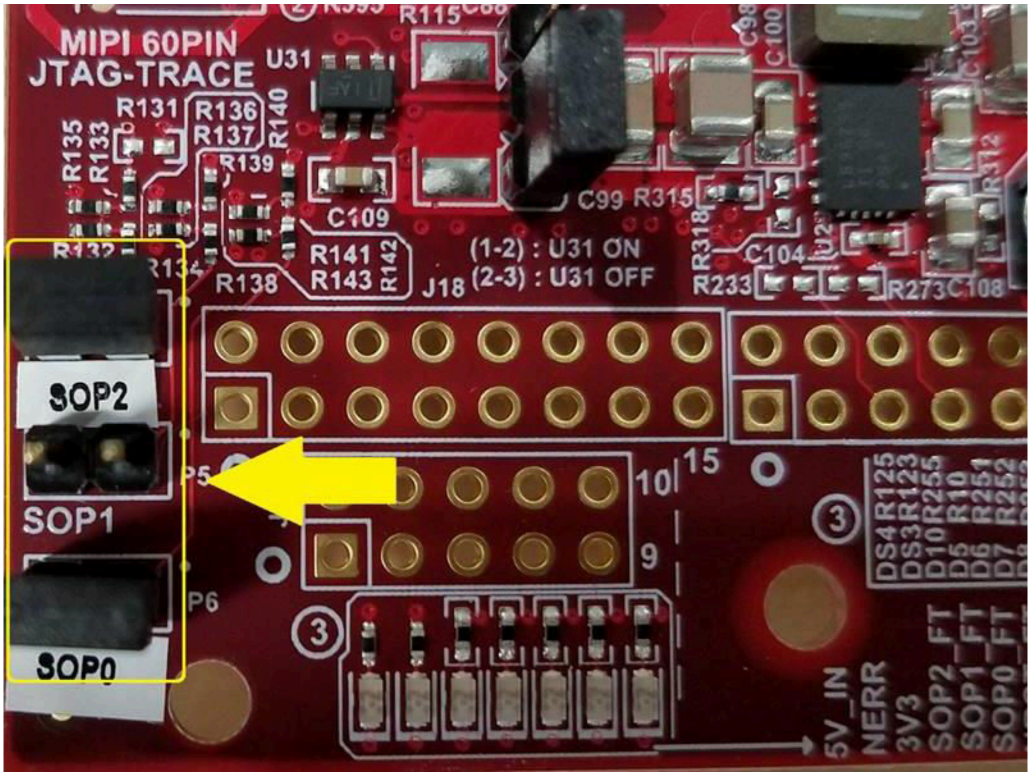


Figure 3-9. SOP0 and SOP2 Location

5. Download and open http://www.ti.com/tool/UNIFLASH.
6. In the *New Configuration* section, locate and select *IWR6843 ES1.0*.
7. Click *Start* to proceed.
8. Click the *Settings & Utilities* tab. Under setup, fill the COM Port text box with the Application/User UART COM port number (COM UART) noted earlier.
9. In the *Program* tab, browse and locate the PC_lab_xwr68xx.bin file from the TIDA-010022 firmware folder.

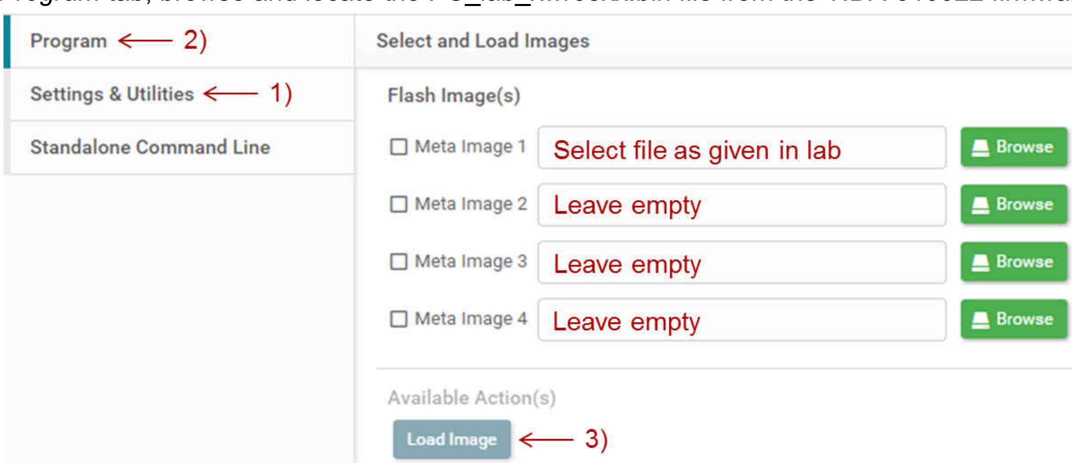


Figure 3-10. Loading Image

10. Reset the board with SW2 and click on *Load Image*.
    a. Successful Flash Procedure: The UniFlash console should indicate: [SUCCESS] Program Load completed successfully
11. Power off the board and remove only the SOP2 jumper.
12. Make sure SOP2 has been removed and that the IWR6843 boards are power cycled. This puts the boards back in functional mode.

3.1.2.3 Starting the Firmware

Follow these steps to start the firmware on the sensor and collector nodes. Refer to Section 3.1.1 for instructions on setting up the sensor node hardware.

1. Power on the collector node (CC1352 only) with the USB power supply first, and then wait for the red LED "Red: DIO6" to be on.
2. Power on the sensor node (IWR6843 boards + CC1352) with 5.0-V power supply
3. Reset the sensor node by pressing the SW2 (NRST) button on the MMWAVEICBOOST board (see Figure 3-11).
4. Press "BTN-2 DIO14" (on collector node) to permit nodes to join the network, the "Red: DIO6" LED should be blinking on the collector node to indicate the network is opened.
5. Once the sensor has joined the network, its red LED will turn on.
6. Press "BTN-2 DIO14" (on collector node) to stop permitting new nodes from joining the network.
7. Once the sensor node has joined a network and has waited 20 seconds for the IWR6843 device to power up, the CC1352 device will configure and start the mmWave.
8. On both sensor and collector nodes, the "Green: DIO7" LED will toggle on each successful transmission between the nodes.

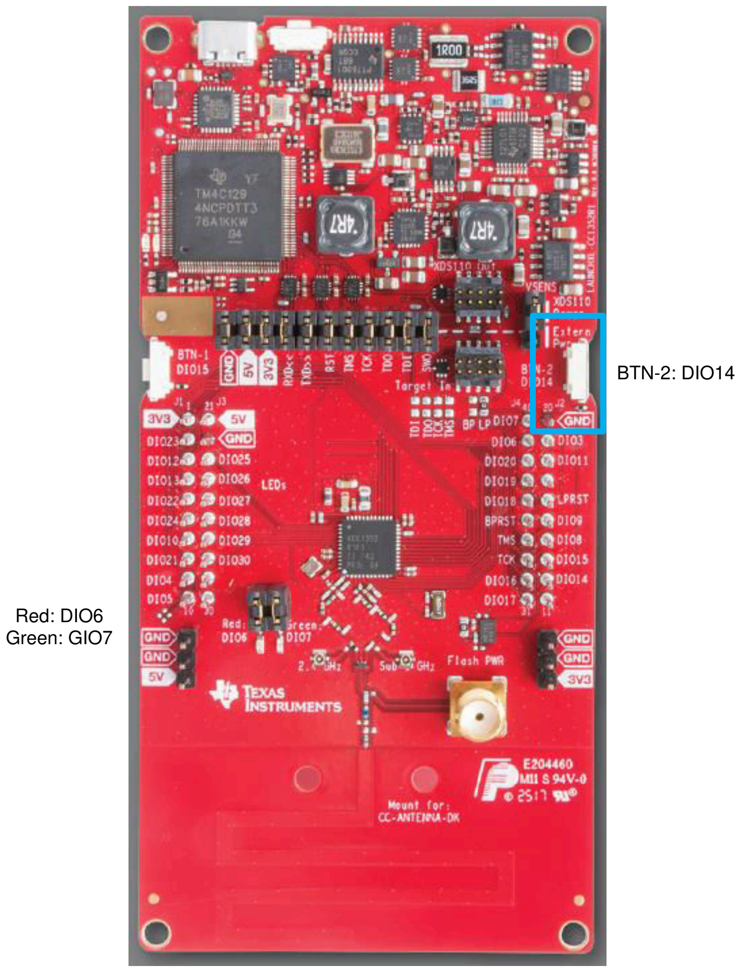


Figure 3-11. Location of LEDs and Button

3.1.2.4 Building the Firmware

The firmware for this TI design is based on the collector and sensor example code provided with the TI 15.4 stack. Patches are included with this TI design to apply the necessary changes to the example code. Follow these steps to apply the patches and build the example code.

1. Download Simplelink-cc13x2-sdk (version 2.10.00.48) from http://www.ti.com/tool/SIMPLELINK-CC13X2-SDK.
2. Project > Import CCS project.
3. Search in C:\ti\simplelink_cc13x2_sdk_2_10_00_48\examples\rtos\CC1352R1_LAUNCHXL\ti154stack directory.
4. Import two projects: collector_cc1352r1lp and sensor_cc1352r1lp.
5. Apply the patch for collector node:
   a. Right-click collector_cc1352r1lp and go to Team > Apply patch…
   b. Apply the collector patch (patch is found in *Software Files*)
   c. Right-click collector_cc1352r1lp and go to *Properties*
   d. In *Include Options*, add folder ${PROJECT_ROOT}/Application/collector_new
   e. In *Predefined Symbols*, add MMWAVE_SENSOR
6. Apply patch for sensor node:
   a. Right-click sensor_cc1352r1lp and go to Team > Apply patch…
   b. Apply the sensor patch (patch is found in *Software Files*)
   c. Right-click sensor_cc1352r1lp and go to *Properties*
   d. In *Include Options*, add folder ${PROJECT_ROOT}/Application/sensor_new
   e. In *Predefined Symbols*, add MMWAVE_SENSOR
7. Build the collector node code using Code Composer Studio:
   a. Select "collector_cc1352r1lp" and right click it, find *Build Project* and click on it to build the project
   b. "***** Build Finished ****" should display under the Code Composer Studio console when the project is built successfully.
8. Build the sensor node code using Code Composer Studio:
   a. Select "sensor_cc1352r1lp" and right click it, find "Build Project" and click on it to build the project
   b. "***** Build Finished ****" should display under the Code Composer Studio console when the project is built successfully.

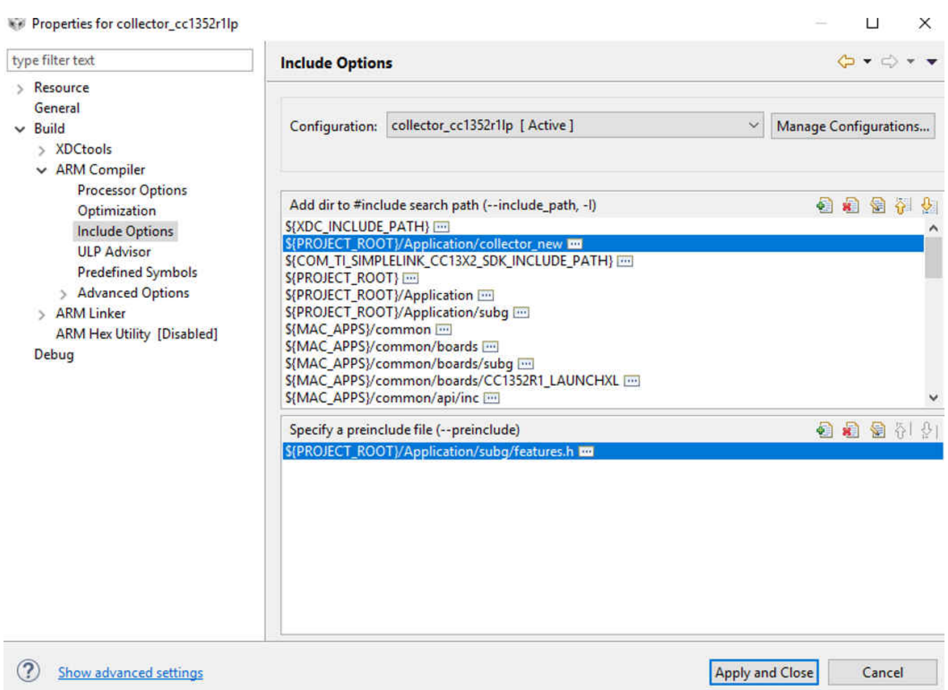


Figure 3-12. Adding Include Options for Collector Node

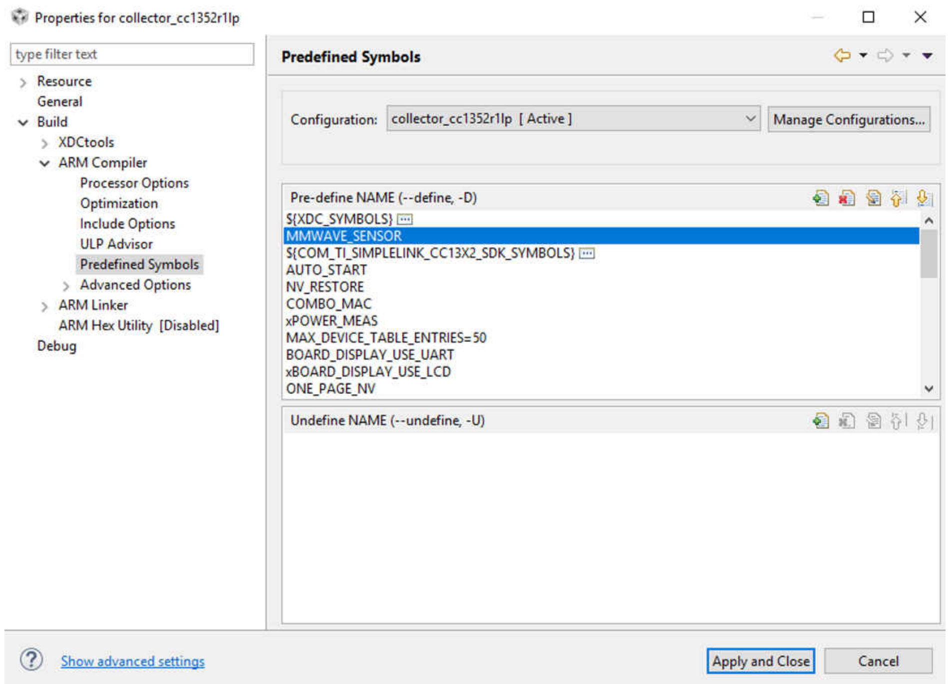


Figure 3-13. Adding Predefined Symbols for Collector Node

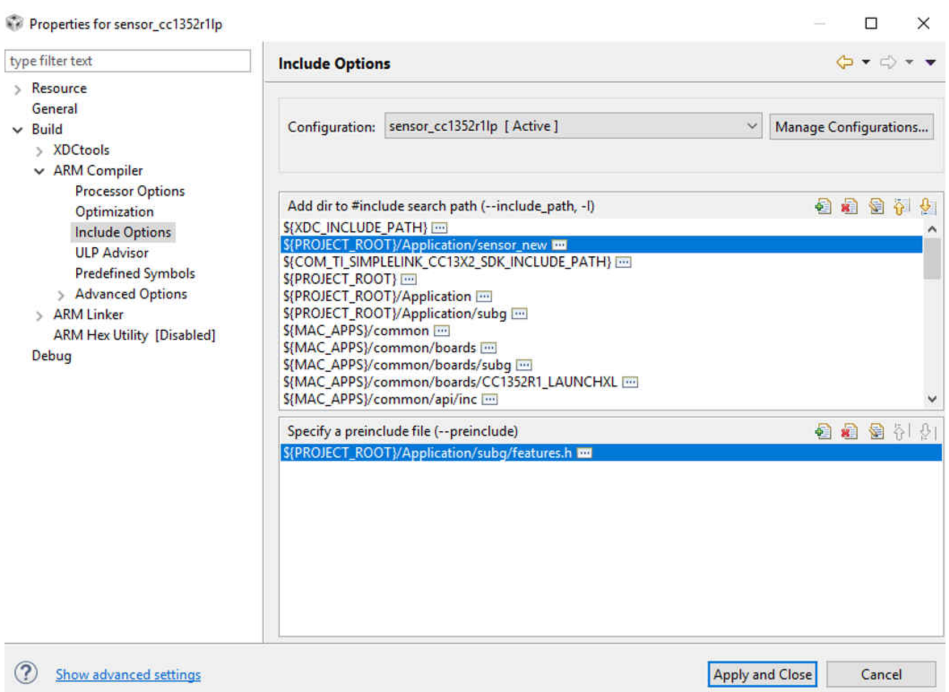


Figure 3-14. Adding Include Options for Sensor Node

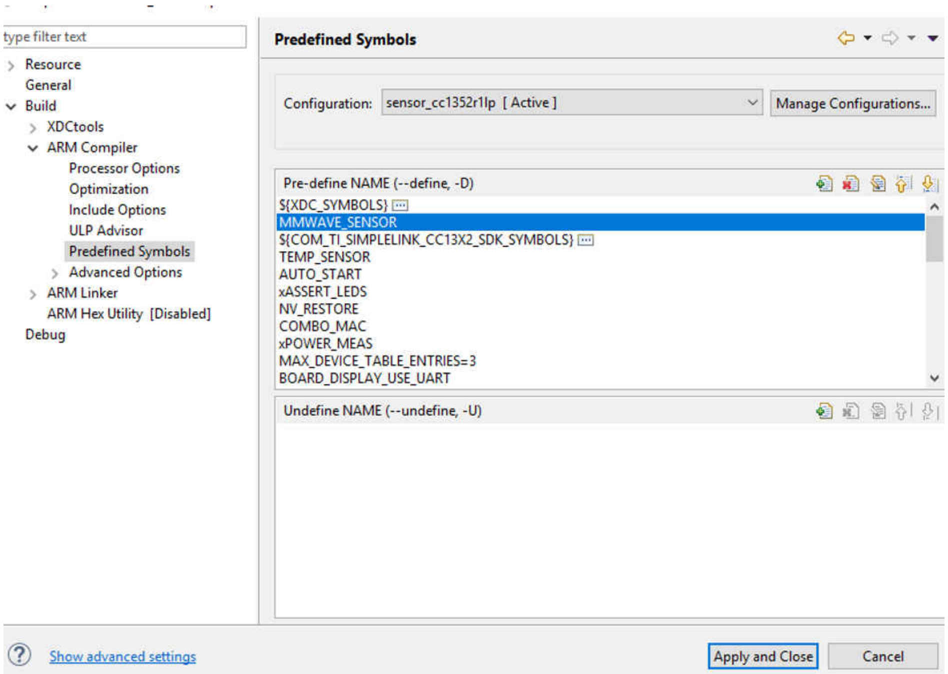


Figure 3-15. Adding Predefined Symbols for Sensor Node

3.1.2.5 Viewing Collector Data Output

When the collector node USB is connected to the PC, the device manager should recognize the following COM ports:

- XDS110 Class Application/User UART
- XDS110 Class Auxiliary Data Port

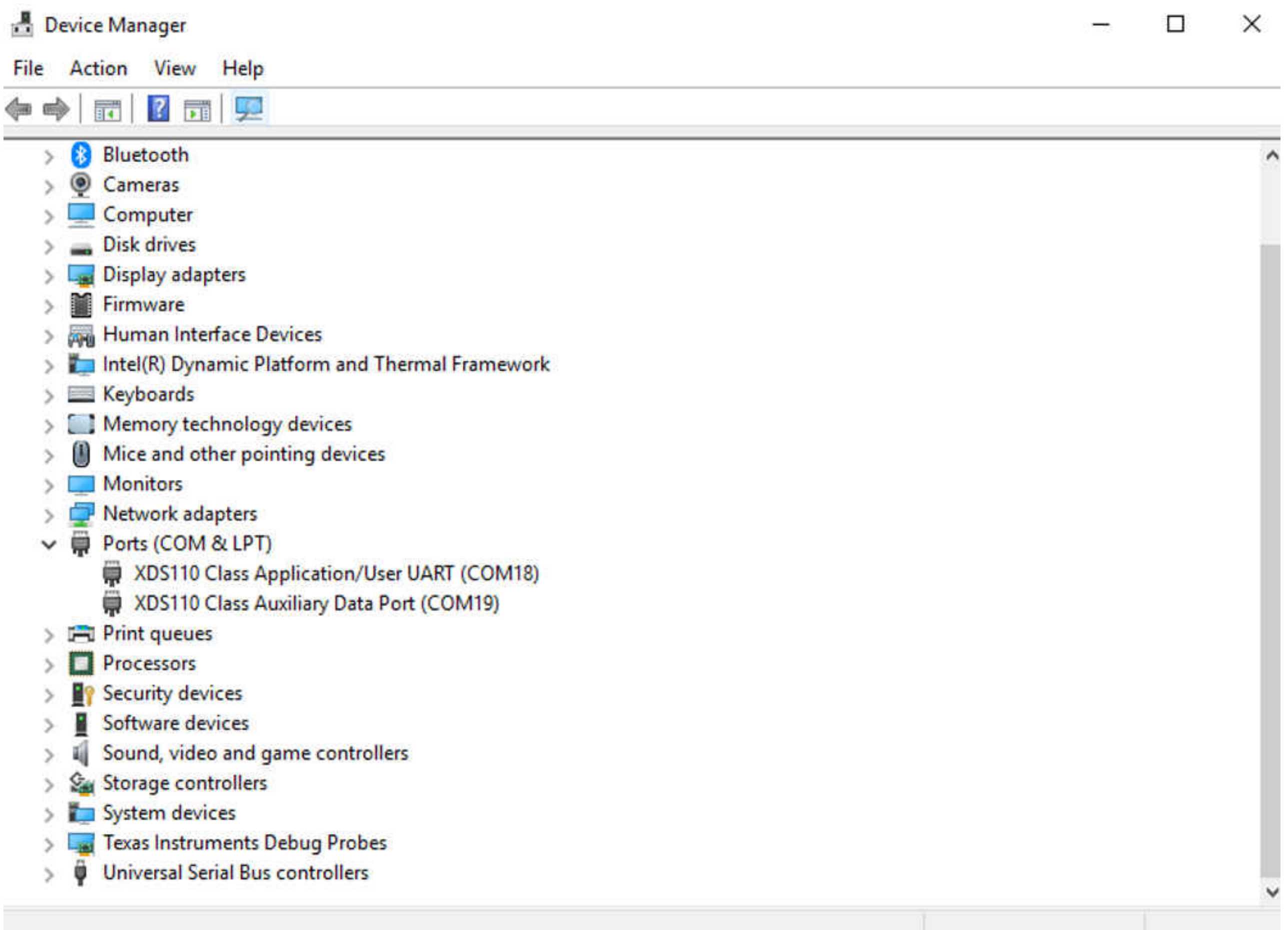


Figure 3-16. COM Ports

Launch TeraTerm, select "Serial" and select the XDS110 Class Application/User UART COM port from the drop-down menu and click the *OK* button. Verify that the COM port has the settings as Figure 3-17 shows.

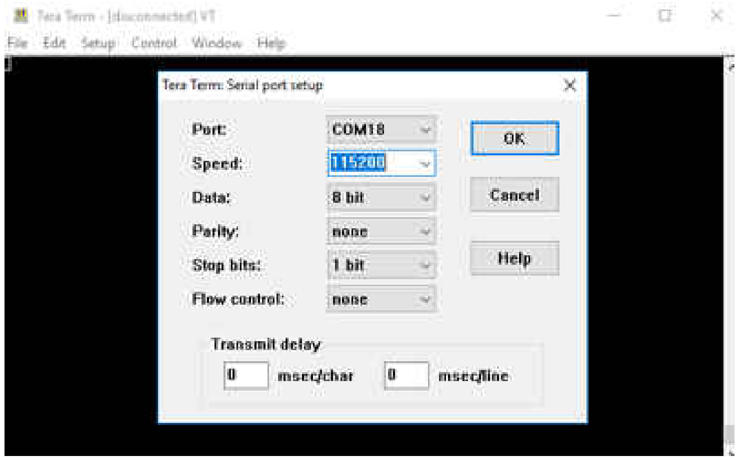


Figure 3-17. TeraTerm Setup

A TeraTerm macro is included to parse the output of the collector node. To launch the macro, select Control menu, then select Macro option. Navigate to the TIDA-010022 firmware folder and select mmWave_collector_macro.ttl.

Figure 3-18. Macro

3.1.2.6 Modifying the mmWave Configuration

The mmWave configuration can be modified by following these steps:

1. After applying the patch from building the firmware in Section 3.1.2.4, go to sensor_cc1352r1lp -> Application -> sensor_new ->mmwave_sensor.c
2. Inside mmwave_sensor.c, find "const char *CommandList[]".
3. Add, remove, or modify commands as needed and rebuild the sensor project.

```
const char *CommandList[] = {"dfeDataOutputMode 1\r",
                    "channelCfg 15 5 0\r",
                    "adcCfg 2 1\r",
                    "adcbufCfg 0 1 1 1\r",
                    "profileCfg 0 60.6 30 10 62 0 0 53 1 128 2500 0 0 30\r",
                    "chirpCfg 0 0 0 0 0 0 0 1\r",
                    "chirpCfg 1 1 0 0 0 0 0 4\r",
                    "frameCfg 0 1 128 0 50 1 0\r",
                    "lowPower 0 1\r",
                    "guiMonitor 1 1 0 0\r",
                    "cfarCfg 6 4 4 4 4 16 16 4 4 50 62 0\r",
                    "doaCfg 600 1875 30 1 1 0\r",
                    "SceneryParam -6 6 0.5 6\r",
                    "GatingParam 4 3 2 0\r",
                    "StateParam 10 5 100 100 5\r",
                    "AllocationParam 250 250 0.25 10 1 2\r",
                    "AccelerationParam 1 1 1\r",
                    "PointCloudEn 0\r",
                    "trackingCfg 1 2 250 20 52 82 50 90\r",
                    "sensorStart\r"
};
```

3.2 Testing and Results

3.2.1 Test Setup

3.2.1.1 People Counting Setup

For best results, the sensor node should be positioned high enough to be above the top of tracked objects and with a slight down tilt. The aim is to position the sensor node so that the antenna beam can encompass the area of interest. If the down tilt is too severe, noise from ground clutter increases and the effective sensing area decreases. If there is no down tilt, counting performance would be worse for cases in which one person is in line with and shielded by another person. Given the antenna radiation pattern of the sensor node, consideration should be taken to not mount the sensor node too close or oriented with the beam directed to the ceiling as this can increase the noise floor and result in less optimal performance.

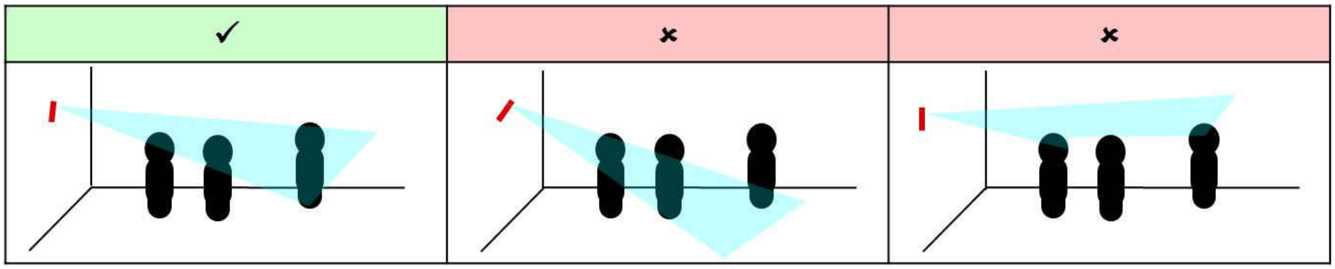


Figure 3-19. Sensor Node (IWR6843 + CC1352) Setup

Setup Requirements:

- Elevate Sensor Node 1.5–2.5 m high
- Down tilt: approximately 10 degrees

Setup using suggested tripod and smartphone clamp mount:

1. Screw on clamp mount to tripod
2. Clamp the sensor node across its width below the power barrel jack to attach sensor node.
3. Adjust the tripod head for approximately 10 degree downward tilt (Tip: Bubble or level smartphone applications can be used to measure down tilt)
4. Plug in the 5-V power supply to the sensor node
5. Extend the tripod so that the sensor node is elevated 1.5–2.5 m from the ground
6. Position the sensor node and tripod assembly in the desired location of room. The sensor node should be positioned so that the 120 degree FOV of the radar antenna encompasses the area of interest and points to the region in which people are expected to enter the space.

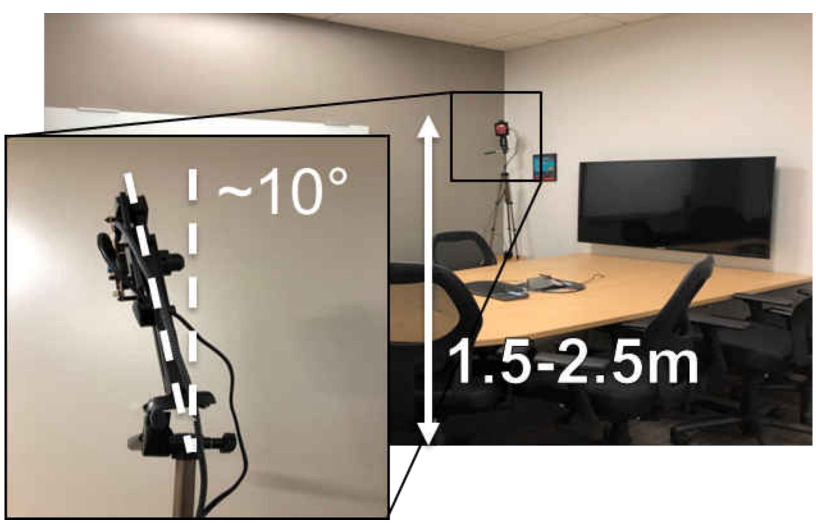


Figure 3-20. Setup with Clamp Mount

3.2.1.2 Power Consumption

The power consumption of the sensor (IWR6843 + CC1352) can be measured by connecting a bench power supply to the sensor.

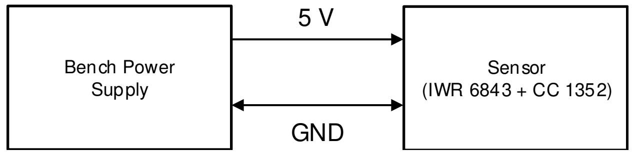


Figure 3-21. Power Consumption Measurement With Bench Power Supply Diagram

3.2.1.3 Wireless RF Range

The range of the wireless sub-1 GHz RF was measured using two CC1352 LaunchPads. One CC1352 LaunchPad was set up as a collector, and the other set up as the sensor. For this test, the sensor LaunchPad remained at a stationary location as the collector LaunchPad was moved away. While the collector LaunchPad was on the move, the collector was continuously receiving the data from the sensor to make sure there were radio packets constantly being transmitted. The distance at which packets were no longer received was then measured.

3.2.2 Test Results

3.2.2.1 People Counting Test Results

For more detailed information on the people-counting test results, see *People Tracking and Counting Reference Design Using mmWave Radar Sensor*.

3.2.2.2 Power Characterization

The results of the power measurements are given in Table 3-2. The current was measured at the input of the MMWAVEICBOOST board. The current consumption includes all devices on the IWR6843 and CC1352 boards.

Table 3-2. Power Characterization

SENSOR STATE	DESCRIPTION	MEASUREMENT
Reset	All boards in reset	5 V x 287 mA = 1.44 W
Idle	mmWave radar inactive, no wireless packets	5 V x 500 mA = 2.50 W
Active	mmWave radar actively tracking and wireless communication	5 V x 670 mA = 3.35 W

3.2.2.3 Wireless RF Range Results

The wireless RF range was measured to be approximately 70 m in a typical office environment with a direct line of sight.

There are ways to increase this distance even further. Using a whip antenna with gain instead of the passive PCB antenna could offer improvements in the wireless RF range. Another option would be to increase the transmit power of the CC1352 device at the expense of increased supply current during the radio transmission intervals.

4 Design Files

4.1 Schematics

To download the schematics for the board, see the design files at TIDA-010022.

4.2 Bill of Materials

To download the bill of materials for the board, see the design files at TIDA-010022.

4.3 PCB Layout Recommendations

To download the PCB layout recommendations for the board, see the design files at TIDA-010022.

4.4 Altium Project

To download the Altium Designer® project for the board, see the design files at TIDA-010022.

4.5 Gerber Files

To download the Gerber Files for the board, see the design files at TIDA-010022.

4.6 Assembly Drawings

To download the assembly drawings for the board, see the design files at TIDA-010022.

5 Software Files

To download the software files, see the design files at TIDA-010022.

6 Related Documentation

1. Texas Instruments, *IWR6843 single-chip 60- to 64-GHz mmWave sensor data sheet*
2. Texas Instruments, *CC1352R SimpleLink™ high-performance dual-band wireless MCU data sheet*
3. Texas Instruments, *LP87702-Q1 dual buck converter and 5-V boost with diagnostic functions data sheet*
4. Texas Instruments, *LM53625/35-Q1, 2.5-A or 3.5-A, 36-V synchronous, 2.1-MHz, step-down DC-DC converter data sheet*
5. Texas Instruments, *People tracking and counting reference design using mmWave radar sensor design guide*

7 Trademarks

LaunchPad™, and TI E2E™SimpleLink™ are trademarks of Texas Instruments.
BoosterPack™ are trademarks of TI.
MATLAB® is a registered trademark of The MathWorks, Inc..
Altium Designer® is a registered trademark of Altium LLC.
All trademarks are the property of their respective owners.

8 About the Author

LING LIN is an intern in the Building Automation System Engineering & Marketing (SEM) at Texas Instruments, where he is responsible for developing reference design solution for the connectivity between mmWave Radar Sensor and dual-band wireless communication. Ling just finished studying bachelor's degree at University of Florida, will continue graduate school at University of Texas, Austin.

GUSTAVO MARTINEZ is a senior systems architect at Texas Instruments where he is responsible for developing reference designs for industrial applications. Gustavo has ample experience developing system reference designs for the Smart Grid and home automation segments, which include high performance application processors, floating-point digital signal processors, and RF technology. Gustavo obtained his master of electrical engineering degree from the University of Houston and his bachelor of science in electrical engineering degree from the University of Texas at El Paso.

CHRISTINA L. LEE is a systems architect at Texas Instruments, where she is responsible for developing firmware for reference design solutions in the industrial segment. Christina has broad experience with applications processors, microcontrollers, and digital-signal processors with specialties in embedded firmware. Christina earned her bachelor of science (BS) in electrical and computer engineering from the University of Texas at Austin.

9 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision A (March 2019) to Revision B (April 2023)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1

Changes from Revision * (August 2018) to Revision A (March 2019)	Page
• Changed IWR1642 to IWR6843 throughout document	1
• Deleted IWR1642, LP87524B-Q1, TPS7A8101, and TPS7A88-Q1 from Resources	1
• Added IWR6843, MWAVEICBOOST, IWR6843ISK, LP87702-Q1, LM53635-Q1 to Resources	1
• Changed TIDA-010022 Block Diagram image	3
• Changed IWR6843 UART Communication image	4
• Changed the voltage rail requirements for the IWR6843 chip	6
• Deleted Sensor Packet image	8
• Added code	8
• Deleted mmWave Short Range Configuration Command List image	9
• Added code	9
• Deleted LP87524B-Q1, LP87524J-Q1 4-A + 2.5-A + Two 1.5-A Buck Converters With Integrated Switches, TPS7A8101 Low-Noise, Wide-Bandwidth, High PSRR, Low-Dropout 1-A Linear Regulator, and TPS7A88-Q1 Automotive, Dual, 1-A, Low-Noise (4 μV_{RMS}) LDO Voltage Regulator sections	9
• Added IWR6843ISK and MMWAVEICBOOST: mmWave EVMs section	11
• Added IWR6843ISK and MMWAVEICBOOST: mmWave EVMs section	11
• Added LAUNCHXL-CC1352R1: SimpleLink™ Multi-Band CC1352R Wireless MCU LaunchPad™ Development Kit section	12
• Added hardware requirement	13
• Changed title of IWR1642 BoosterPack Hardware Modifications table to MMWAVEICBOOST Hardware Modifications	13
• Changed information in MMWAVEICBOOST Hardware Modifications table	13
• Changed MMWAVEICBOOST Front PCB Layout View image	13
• Deleted Default Jumpers Setting, After Removing Four Jumpers, Default XDS110 Power Jumper Setting, and After Switching the XDS110 Power Jumper to "Extern" Power images	13
• Changed Sensor Node (IWR6843 Boards + CC1352) Power Supply image	13
• Added information to Loading the IWR6843 Firmware section	17
• Changed SOP0 and SOP2 Location image	17
• Added a step to start the firmware	20
• Added code	26
• Deleted mmWave Short Range Configuration Command List image	26
• Changed Power Consumption Measurement With Bench Power Supply Diagram image	28
• Changed information in Power Characterization section	28
• Changed Related Documentation	29

IMPORTANT NOTICE AND DISCLAIMER