TI Designs: TIDA-010022
People Counting and Tracking Using mmWave Radar Sensor With Sub-1 GHz Reference Design

TEXAS INSTRUMENTS

Description

This TI design demonstrates the use of the IWR1642 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 IWR1642BOOST evaluation module (EVM) and 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
IWR1642	Product Folder
IWR1642BOOST	Tool Folder
CC1352	Product Folder
LAUNCHXL-CC1352R1	Tool Folder
LP87524B-Q1	Product Folder
TPS7A8101	Product Folder
TPS7A88-Q1	Product Folder

TI E2E™ Community

ASK Our E2E™ Experts

Features

- Contains demonstration hardware and software to use the IWR1642 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

Sensor Node   Collector Node
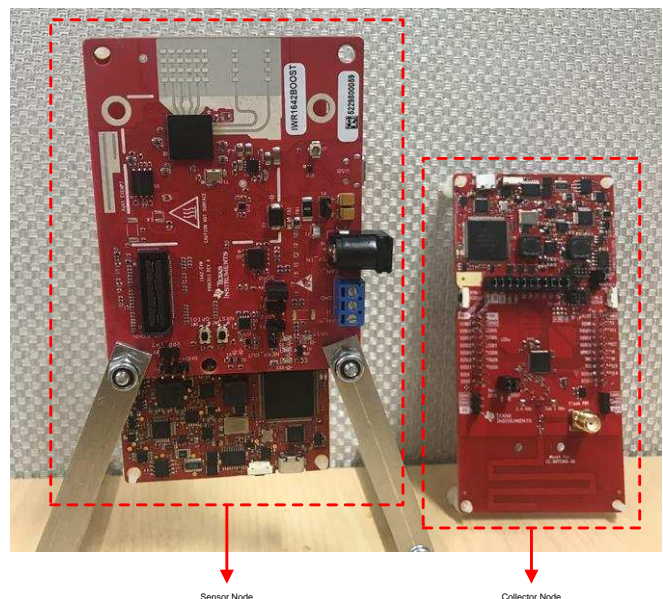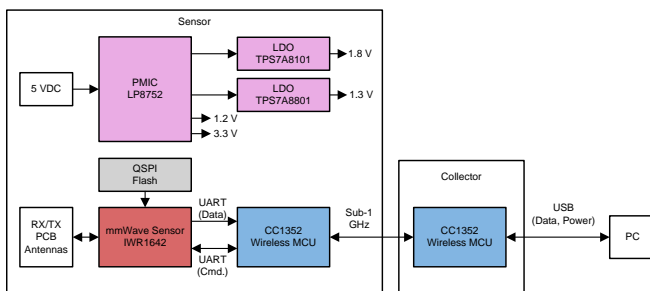
An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

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 extend long-long 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. 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
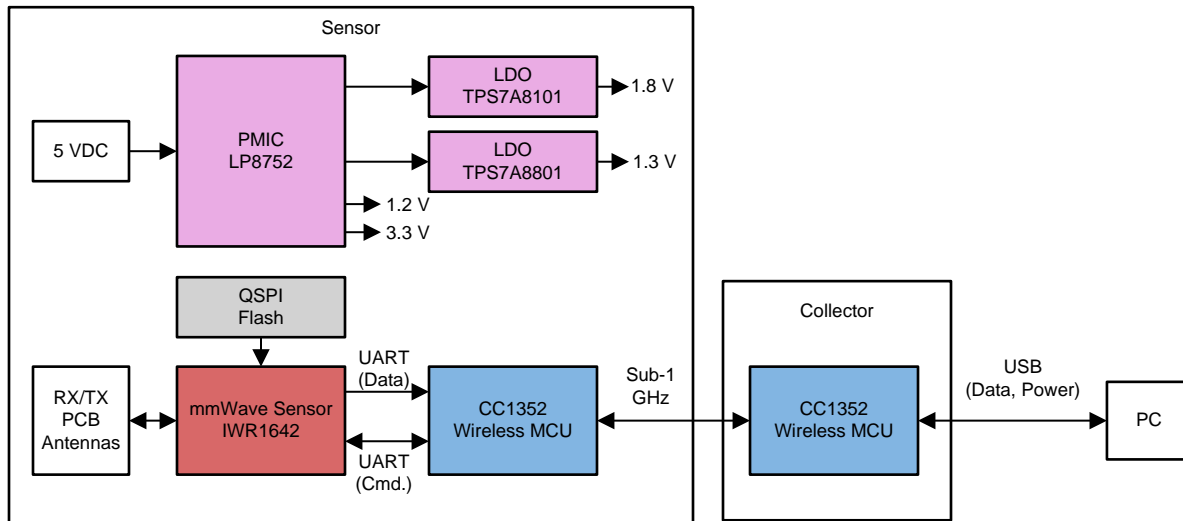
2.1 Block Diagram

Figure 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 IWR1642. 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
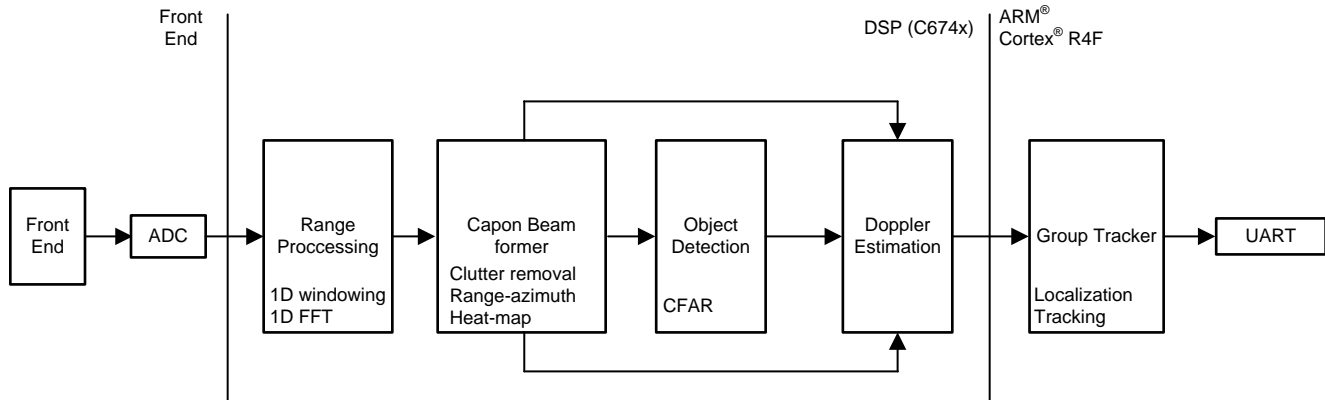

Figure 2. People-Counting Application Block Diagram

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

2.2.1.2 IWR1642 UART Communication

The CC1352 device configures the IWR1642 device via a command line interface (CLI) UART port. After the configuration is completed, the IWR1642 starts sending out radar data continuously back to CC1352 through a separate data UART port. Then, the CC1352 parses the data from the IWR1642 and packetizes and sends the data package over the air to the collector node constantly. Figure 3 is a block diagram of the IWR1642 UART communication.
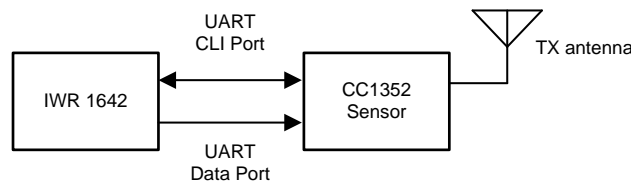
Figure 3. IWR1642 UART Communication

2.2.1.2.1 IWR1642 Frame Output

The IWR1642 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 4 illustrates the IWR1642 output frame format.

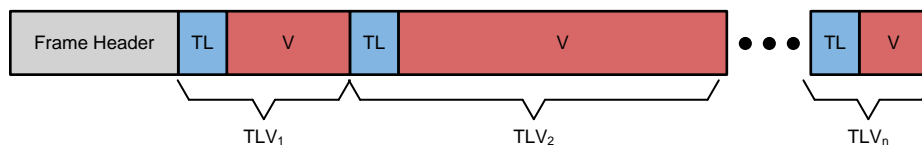**Figure 4. IWR1642 Output Frame Format**

2.2.1.2.2 IWR1642 Frame Header

The frame header is a fixed size (52 bytes) and has the following structure (see *Figure 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 5. IWR1642 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 6).

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

Figure 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 7. Point Cloud TLV

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

*People Counting and Tracking Using mmWave Radar Sensor With Sub-1
GHz Reference Design*

```
% 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 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 IWR1642 chip requires multiple voltage rails: 1.8 V, 1.3 V, 1.2 V, and 3.3 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 TI design, a PMIC is used to generate 2.3 V, 1.8 V, 1.2 V and 3.3 V from the 5-V input rail. Two LDOs are used to generate 1.8 V from the 2.3-V PMIC rail and 1.3 V from the 1.8-V PMIC rail.

The CC1352 device used in the sensor node is powered directly from the 3.3-V rail generated by Buck 0 of the PMIC. This buck has a current rating of 1.5 A. The CC1352 data sheet specs a current consumption of 24.3 mA at 14 dBm and 3.6 V and the IWR1642 data sheet specs a current consumption of 50 mA for the 3.3-V rail. Therefore, the 3.3-V rail from the PMIC can support the both the IWR1642 and CC1352 devices.

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*
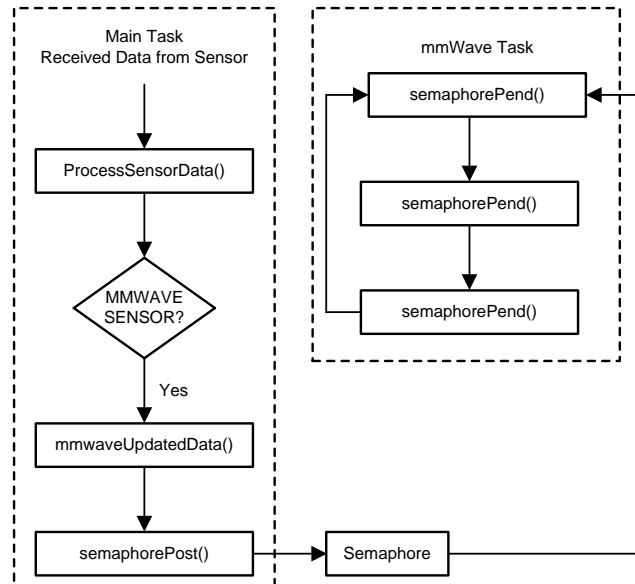
Figure 9 illustrates the collector node software flowchart.

Figure 9. Collector Node Software Flowchart

2.2.4.2 *Sensor Node*

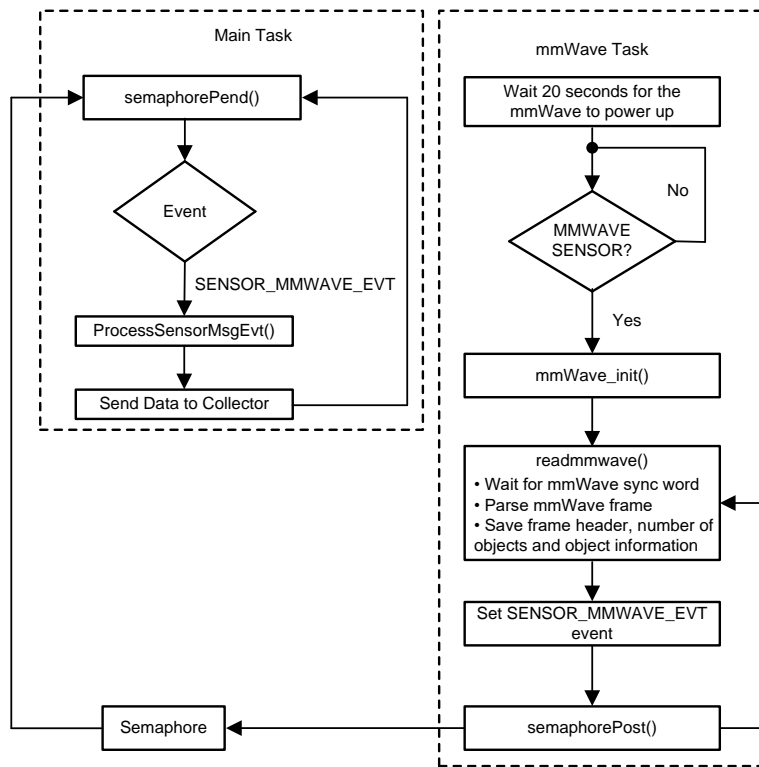Figure 10 shows the sensor node software flowchart.

Figure 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;
```

Figure 11. Sensor Packet

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 3 0\r",
                "adcCfg 2 1\r",
                "adcbufCfg 0 1 1 1\r",
                "profileCfg 0 77 30 7 62 0 0 60 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 2\r",
                "frameCfg 0 1 128 0 50 1 0\r",
                "lowPower 0 0\r",
                "guiMonitor 1 1 0 0\r",
                "cfarCfg 6 4 4 0 0 16 16 4 4 50 62 0\r",
                "doaCfg 600 1875 30 1\r",
                "SceneryParam -6 6 0.05 4\r",
                "GatingParam 4 3 2 0\r",
                "StateParam 10 5 10 100 5\r",
                "AllocationParam 250 0.01 5 1 2\r",
                "VariationParam 0.289 0.289 1.0\r",
                "PointCloudEn 0\r",
                "trackingCfg 1 2 250 20 200 50 90\r",
                "sensorStart\r"
};
```

Figure 12. mmWave Short Range Configuration Command List

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 TI Design features the following devices:

1. IWR1642: Single-chip 76- to 81-GHz mmWave sensor
2. CC1352: SimpleLink multi-band wireless MCU LaunchPad development kit
3. LP87524B and LP87524BJ-Q1: 4-A + 2.5-A + two 1.5-A buck converters with integrated switches
4. TPS7A8101: Low-noise, wide-bandwidth, high PSRR, low-dropout 1-A linear regulator
5. TPS7A88-Q1: Automotive, dual, 1-A, low-noise (4 μV_{RMS}) LDO voltage regulator

For more information on the device, see its datasheet or product folders on www.ti.com.

2.3.1 IWR1642: Single-Chip 76- to 81-GHz mmWave Sensor

The IWR1642 device is an integrated single-chip mmWave sensor based on FMCW radar technology capable of operation in the 76- to 81-GHz band with up to 4-GHz continuous chirp. The device is built with TI's low-power 45-nm RFCMOS process, and this solution enables unprecedented levels of integration in an extremely small form factor. The IWR1642 is an ideal solution for low-power, self-monitored, ultra-accurate radar systems in industrial applications such as building automation, factory automation, drones, material handling, traffic monitoring, and surveillance.

The IWR1642 device is a self-contained, single-chip solution that simplifies the implementation of mmWave sensors in the band of 76 to 81 GHz. The IWR1642 includes a monolithic implementation of a 2TX, 4RX system with built-in PLL and A2D converters. The IWR1642 also integrates a DSP subsystem, which contains TI's high-performance C674x DSP for the radar signal processing. The device includes an ARM R4F-based processor subsystem, which is responsible for front-end configuration, control, and calibration. Simple programming model changes can enable a wide variety of sensor implementation with the possibility of dynamic reconfiguration for implementing a multi-mode sensor. Additionally, the device is provided as a complete platform solution including reference hardware design, software drivers, sample configurations, API guide, training, and user documentation.

Figure 13 shows the IWR1642 block diagram.
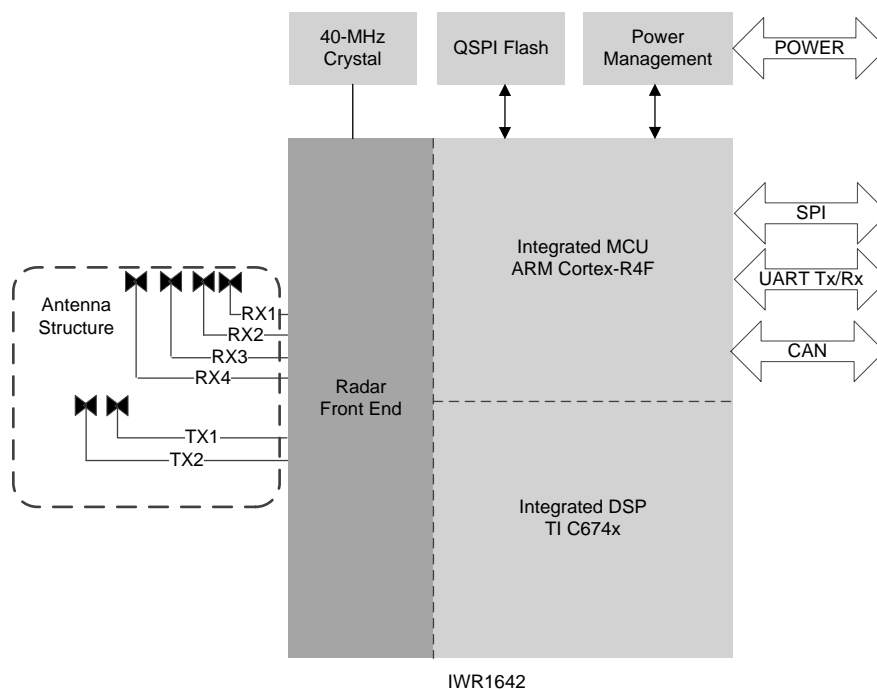
Figure 13. IWR1642 Block Diagram

2.3.2 CC1352: SimpleLink™ Multi-Band Wireless MCU LaunchPad™ Development Kit

The CC1352 device is part of the SimpleLink microcontroller (MCU) platform which consists of Wi-Fi®, Bluetooth® low energy (BLE), sub-1 GHz, 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. This LaunchPad kit supports programming and debugging from Code Composer Studio™ and IAR Embedded Workbench® integrated development environments (IDEs).

The CC1352 device includes the following features:

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

Figure 14 shows the CC1352 block diagram.
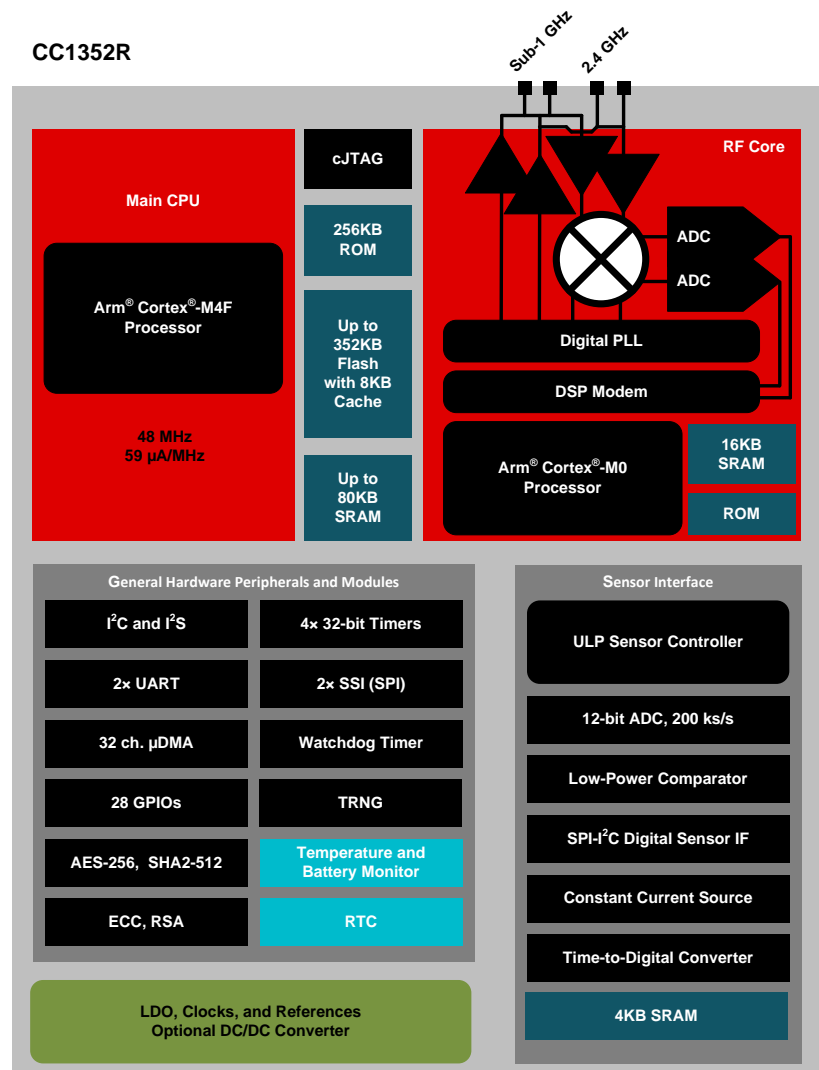
Figure 14. CC1352 Block Diagram

2.3.3 LP87524B-Q1, LP87524J-Q1 4-A + 2.5-A + Two 1.5-A Buck Converters With Integrated Switches

The LP87524B-Q1 and LP87524J-Q1 include four step-down DC-DC converter cores configured for four single-phase outputs. The cores are designed for flexibility; most of the functions are programmable, thus giving a possibility to optimize the regulator operation for each application.

The LP87524B-Q1 and LP87524J-Q1 have the following features:

- DVS support
- Automatic-mode control based on the loading (PFM or PWM mode)
- Forced-PWM mode operation
- Optional external clock input to minimize crosstalk
- Optional spread spectrum technique to reduce EMI
- Phase control for optimized EMI
- Synchronous rectification
- Current mode loop with PI compensator
- Soft start
- Power Good flag with maskable interrupt
- Power Good signal (PGOOD) with selectable sources
- Average output current sensing (for PFM entry and load current measurement)

Figure 15 illustrates the LP87524B-Q1 and LP87524J-Q1 buck converter block diagram.
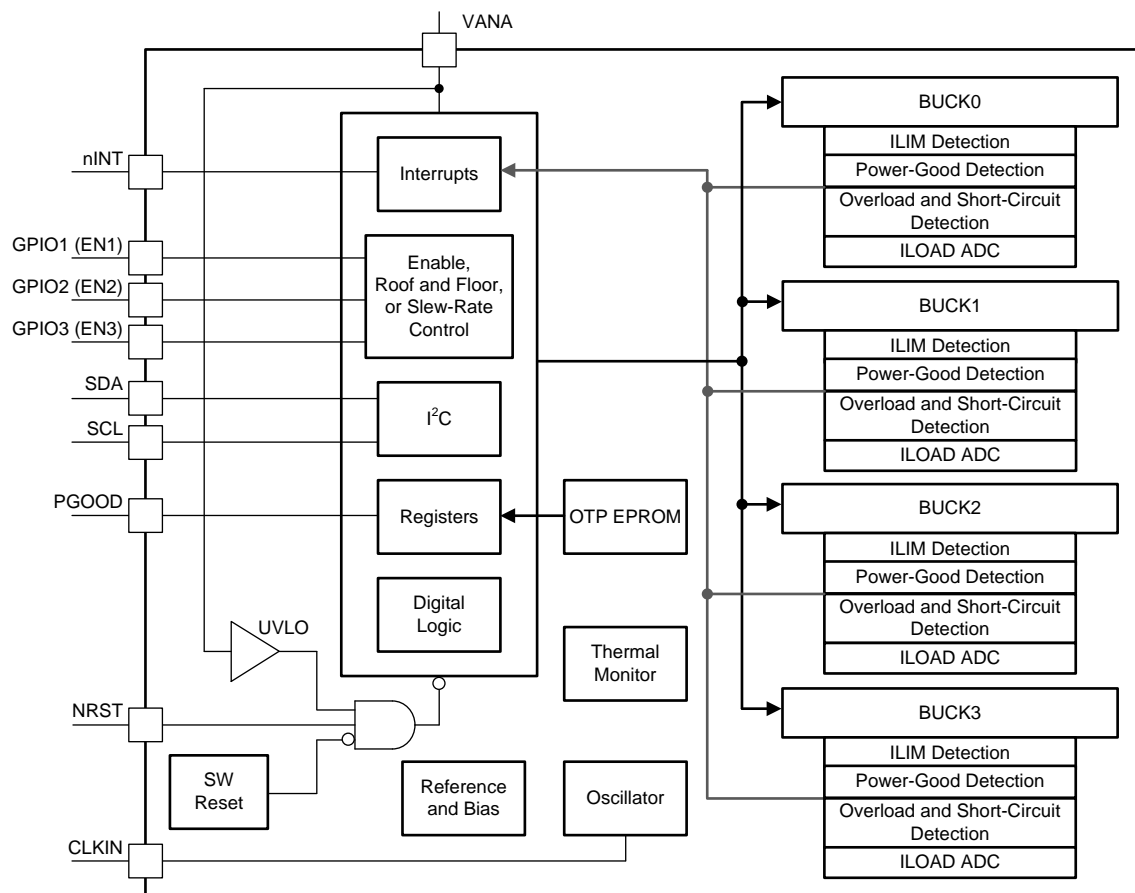
Figure 15. LP87524B-Q1 and LP87524J-Q1 Buck Converter Block Diagram

2.3.4 TPS7A8101 Low-Noise, Wide-Bandwidth, High PSRR, Low-Dropout 1-A Linear Regulator

The TPS7A8101 device belongs to a family of new-generation LDO regulators that use innovative circuitry to achieve wide bandwidth and high loop gain, resulting in extremely high PSRR (over a 1-MHz range) even with very low headroom (VI – VO). A noise-reduction capacitor (C(NR)) at the NR pin and a bypass capacitor (C(BYPASS)) decrease noise generated by the bandgap reference to improve PSRR, while a quick-start circuit fast-charges the noise-reduction capacitor. This family of regulators offers sub-bandgap output voltages, current limit, and thermal protection, and is fully specified from –40°C to 125°C.

Figure 16 illustrates the TPS7A8101 LDO regulator block diagram.
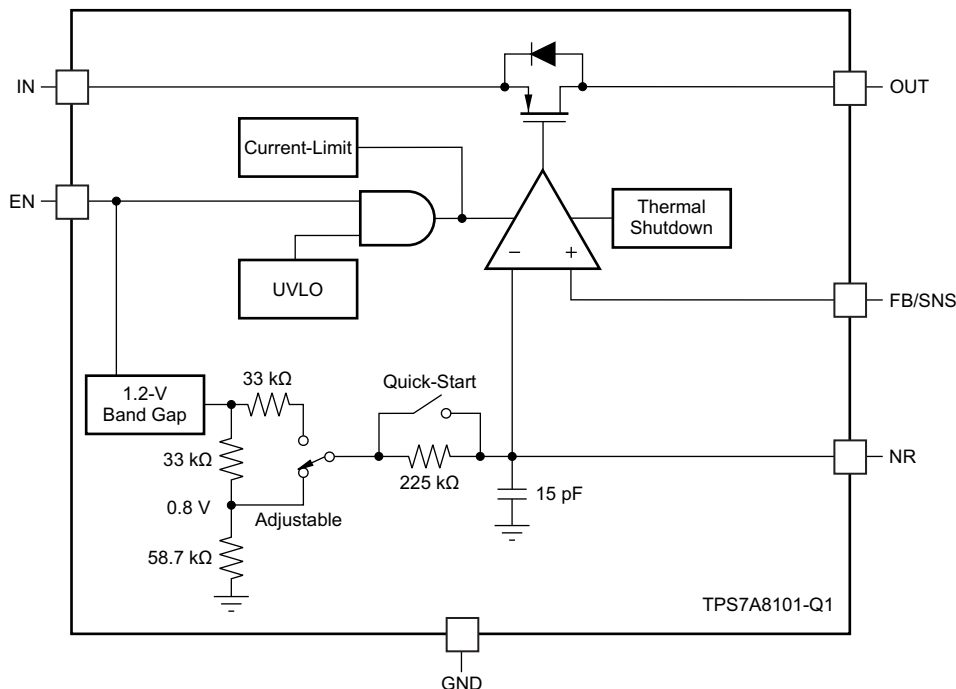
Figure 16. TPS7A8101 LDO Regulator Block Diagram

2.3.5 TPS7A88-Q1 Automotive, Dual, 1-A, Low-Noise (4 µV$_{RMS}$) LDO Voltage Regulator

The TPS7A88-Q1 is a dual, low-noise (4 µV$_{RMS}$), low-dropout (LDO) voltage regulator capable of sourcing 1 A per channel with 250 mV of maximum dropout.

The TPS7A88-Q1 provides the flexibility of two independent LDOs and approximately 50% smaller solution size than two single-channel LDOs. Each output is adjustable with external resistors from 0.8 V to 5.15 V. The TPS7A88-Q1 wide input-voltage range supports operation as low as 1.4 V and up to 6.5 V.

With 1% output voltage accuracy (over line, load, and temperature) and soft-start capabilities to reduce inrush current, the TPS7A88-Q1 is ideal for powering sensitive analog low-voltage devices (such as voltage-controlled oscillators [VCOs], analog-to-digital converters [ADCs], digital-to-analog converters [DACs], high-end processors, and field-programmable gate arrays [FPGAs]).

The TPS7A88-Q1 is designed to power noise-sensitive components such as those found in RF, radar communications, and telematic applications. The low 4-µV$_{RMS}$ output noise and wideband PSRR (40 dB at 1 MHz) minimizes phase noise and clock jitter. These features maximize performance of clocking devices, ADCs, and DACs. The TPS7A88Q1 features wettable flanks for simple optical inspection
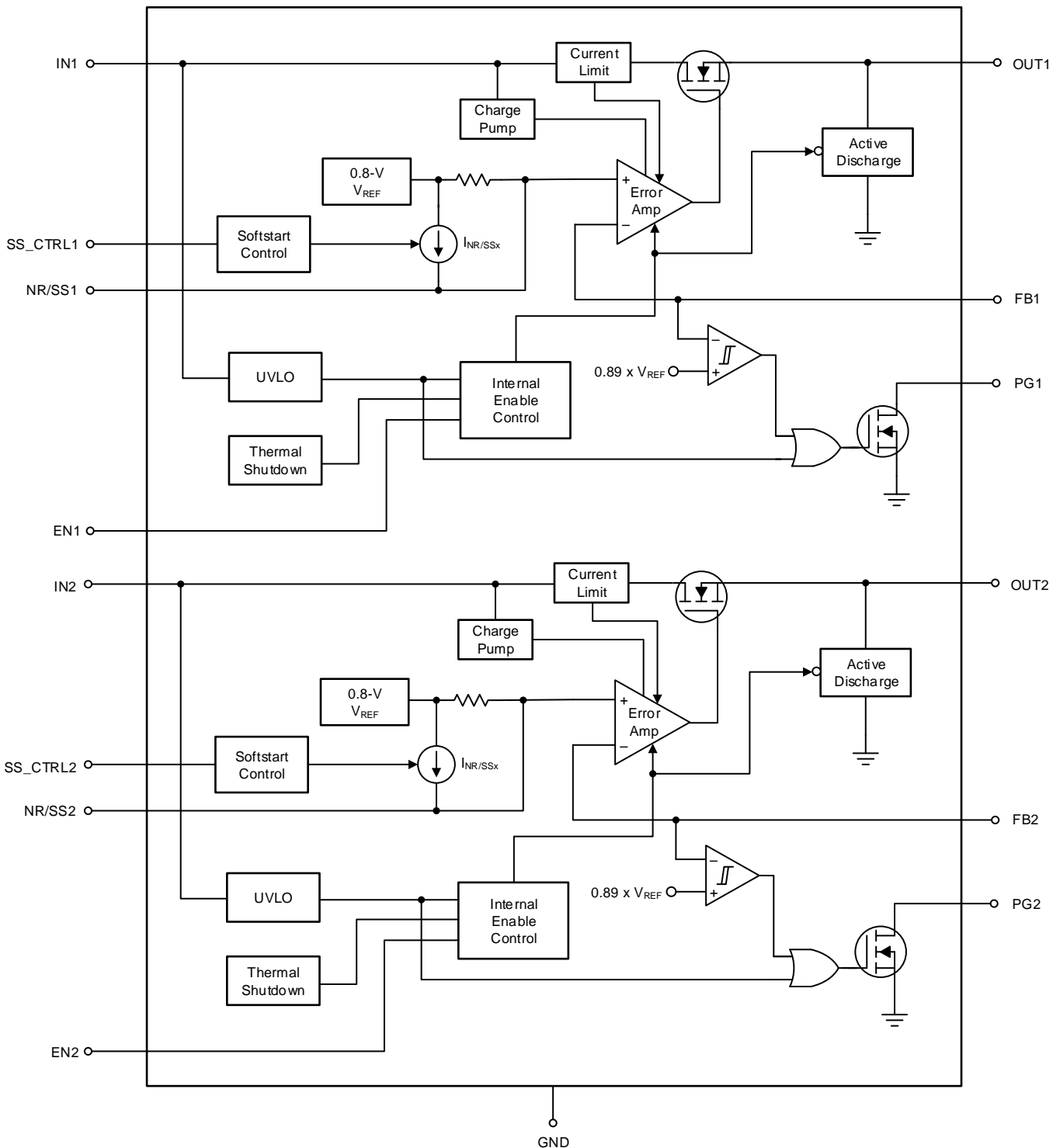
Figure 17. TPS7A88-Q1 LDO Regulator Block Diagram

Copyright © 2018, Texas Instruments Incorporated

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 TI Design requires the following hardware:
- One IWR1642 mmWave sensor BoosterPack (IWR1642BOOST)
- Two CC1352 LaunchPads (LAUNCHXL-CC1352R1)
- One 5 V, 2.5 A or greater DC power supply

NOTE: Before making any hardware modifications to the IWR1642 BoosterPack, flash the board with the steps in Section 3.1.2.

The IWR1642 BoosterPack can attach directly to the CC1352 LaunchPad after some hardware modifications. Table 2 lists the hardware modifications for the IWR1642 BoosterPack.

Table 2. IWR1642 BoosterPack™ Hardware Modifications

MODIFICATION	REASON
Remove R136	Disconnects AR_MSS_LOGGER pin from JTAG debugger
Remove R128	Disconnects AR_RS232TX pin from JTAG debugger
Remove R129	Disconnects AR_RS232RX pin from JTAG debugger
Remove R106	Disconnects AR_ANATEST4 pin from LaunchPad
Remove R107	Disconnects PGOOD pin from LaunchPad
Remove R102	Disconnects PMIC_EN1 pin from LaunchPad
Remove R76	Disconnects AR_SPICLK1 pin from LaunchPad
Remove R19	Disconnects AR_NERR_OUT pin from LaunchPad
Remove R60	Disconnects AR_DP7 pin from LaunchPad
Remove R4	Disconnects AR_MOSI1 pin from LaunchPad
Remove R6	Disconnects AR_MISO1 pin from LaunchPad
Remove R10	Disconnects AR_HOSTINTR1 pin from LaunchPad
Remove R26	Disconnects AR_MSS_LOGGER pin from LaunchPad
Remove R1	Disconnects AR_CS1 pin from LaunchPad
Remove R24	Disconnects AR_NERRIN pin from LaunchPad
Add R169	Connects AR_RS232TX pin to LaunchPad
Add R166	Connects AR_RS232RX pin to LaunchPad
Add R26	Connects AR_MSS_LOGGER pin to LaunchPad
Add R130	Connects +3.3-V supply to LaunchPad

The components that need to be removed from the original mmWave board are marked by the red squares in Figure 18 and Figure 19. The components that need to be added are marked in blue squares.
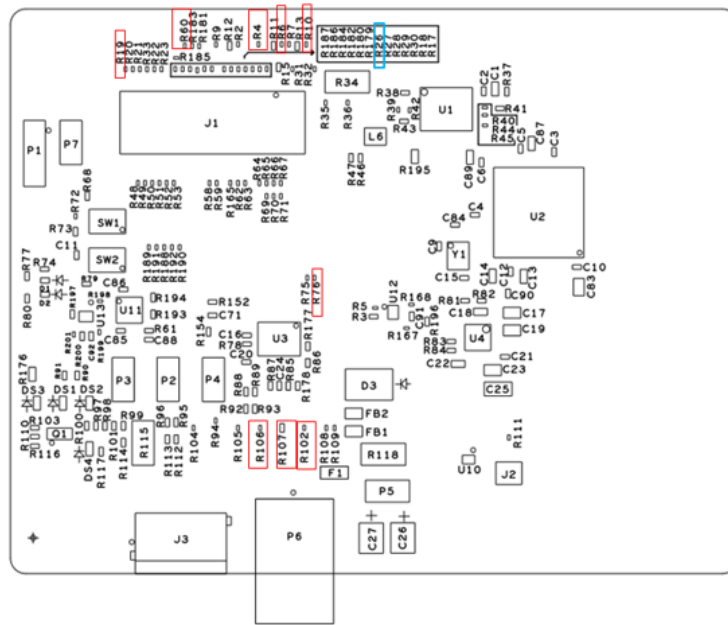
Figure 18. mmWave Radar Front PCB Layout View
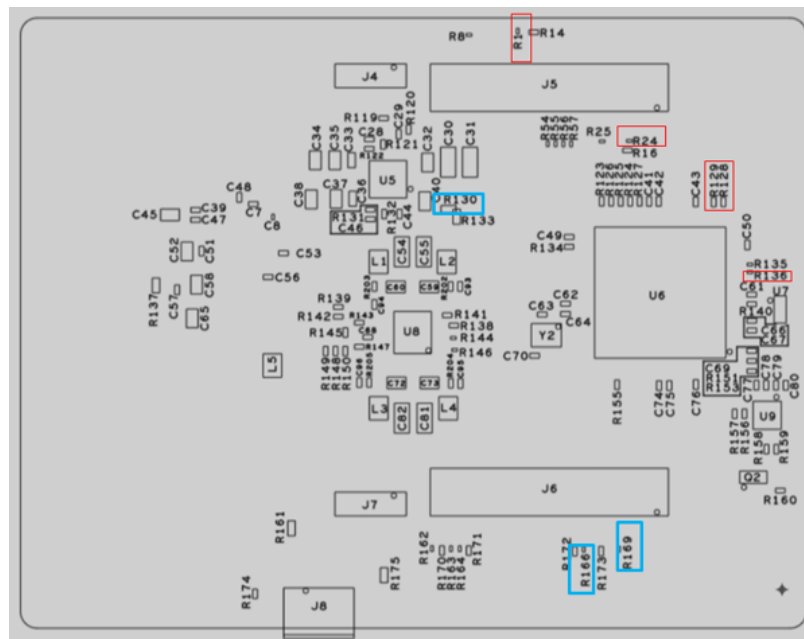
Figure 19. mmWave Radar Back PCB Layout View

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 IWR1642 BoosterPack (see Figure 24).

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

Figure 20. Default Jumpers Setting

Figure 21. After Removing Four Jumpers

Figure 22. Default XDS110 Power Jumper Setting
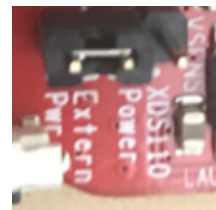
Figure 23. After Switching the XDS110 Power Jumper to "Extern" Power
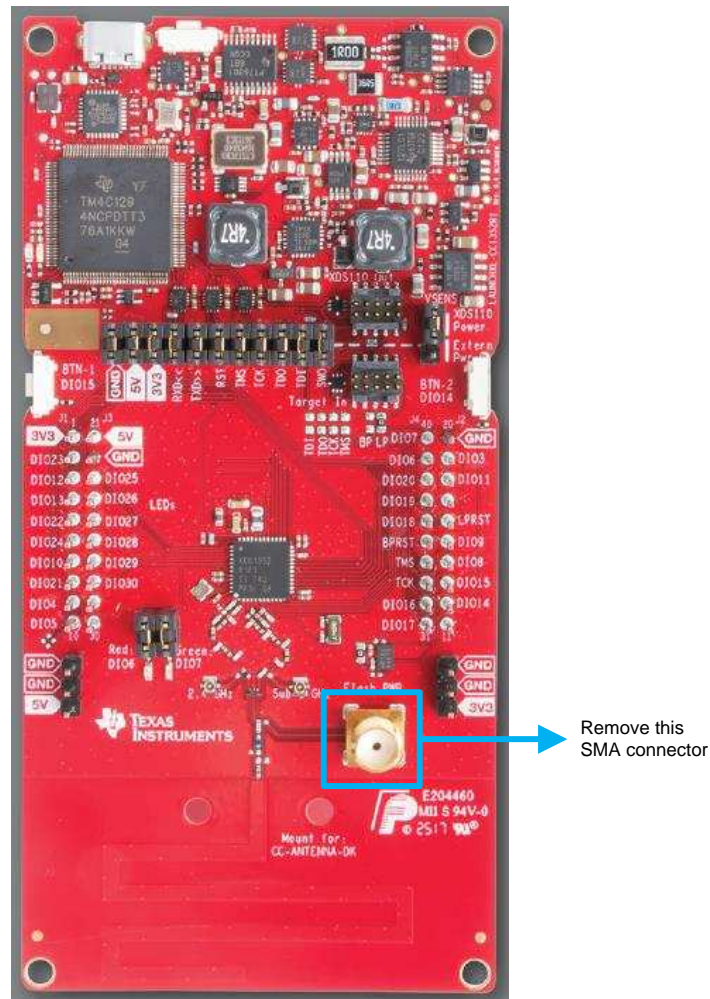
Figure 24. SMA Connector in CC1352 to be Removed

After all hardware modifications are completed, the IWR1642 BoosterPack and CC1352 LaunchPads can be mated.

The sensor node (IWR1642 + CC1352) can be powered by connecting the 5-V power supply to the IWR1642 BoosterPack and the collector node can be powered by connecting the USB power supply.
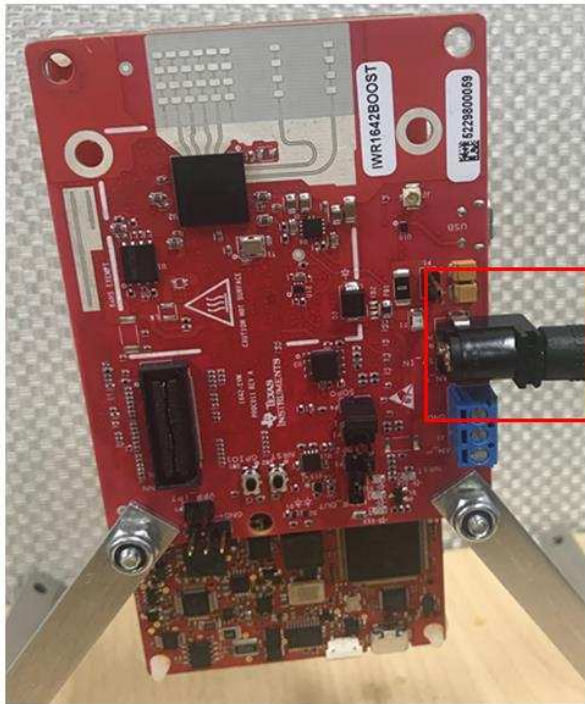
Figure 25. Sensor Node (IWR1642 + CC1352) Power Supply
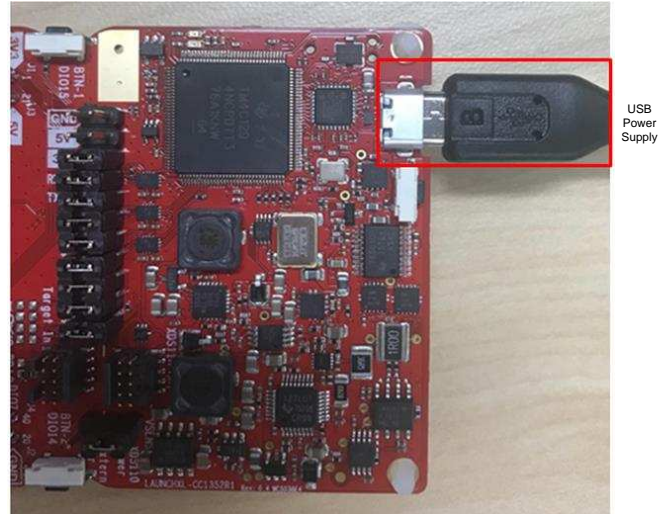
Figure 26. Collector Node (CC1352 Only) Power Supply

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 IWR1642 BoosterPack 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.
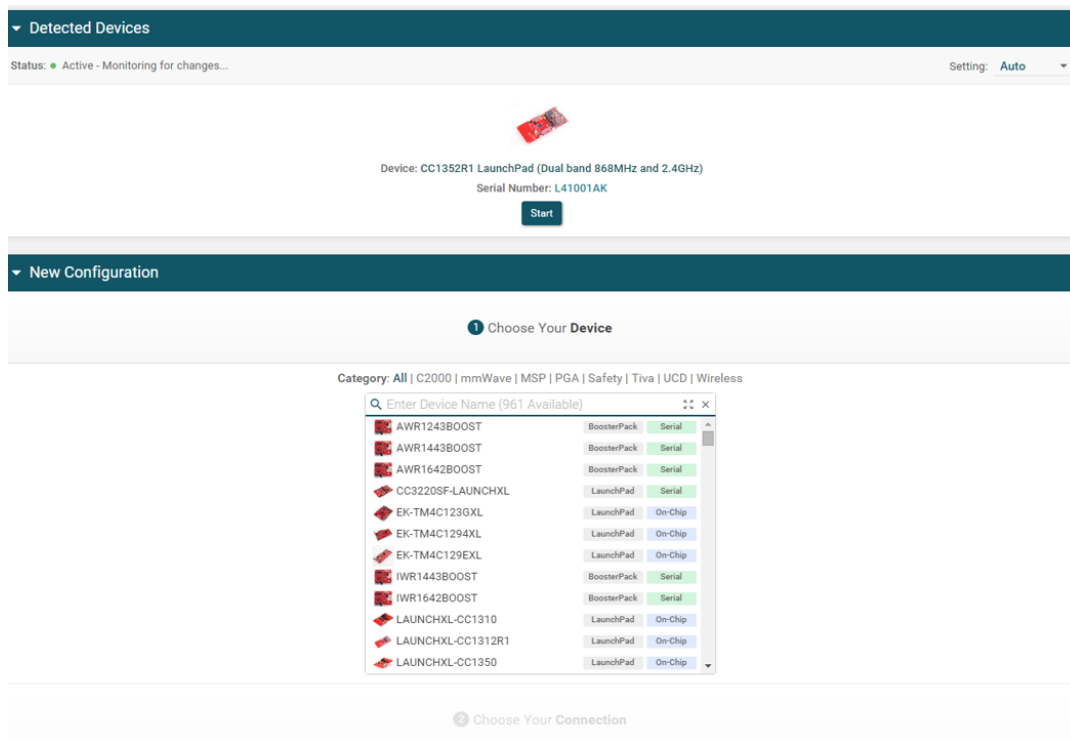
Figure 27. Detected Devices

Copyright © 2018, Texas Instruments Incorporated
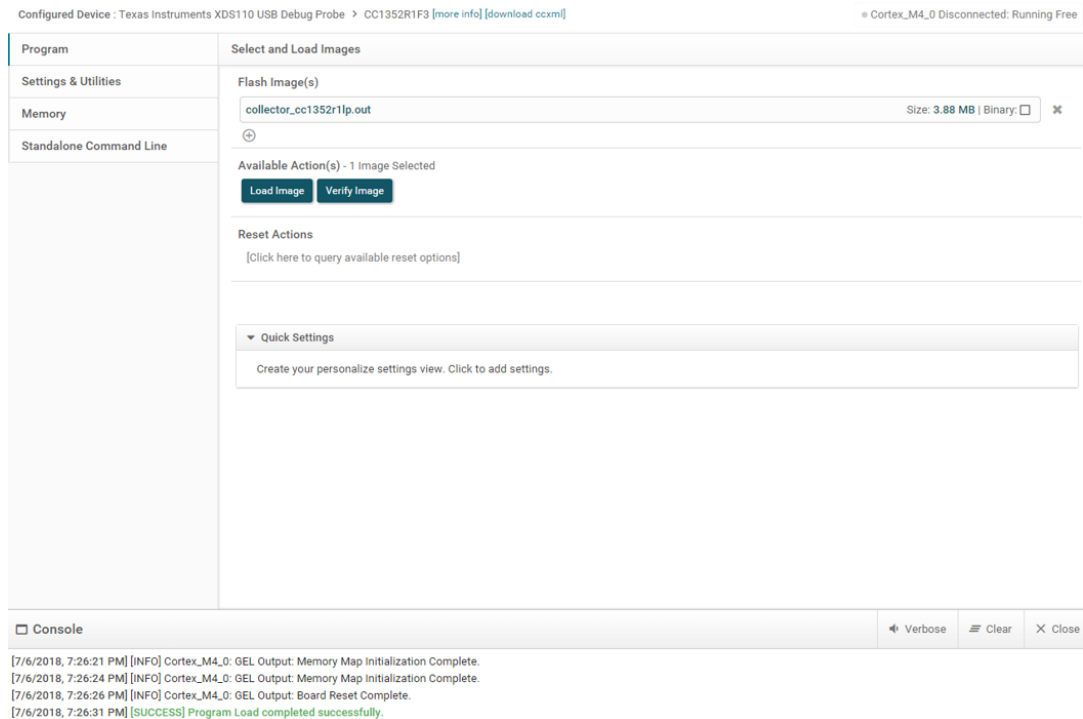
Figure 28. Loading Image Successfully

3.1.2.2   Loading the IWR1642 Firmware

Follow these steps to load the firmware on the IWR1642 BoosterPack:

> **NOTE:**   Make sure the AR_RS232TX and AR_RS232RX pins are connected to the debugger and that the CC1352 LaunchPad is not attached to the IWR1642 BoosterPack.

1. Power on the BoosterPack using a 5-V power supply.
2. Connect the BoosterPack to your PC and check the COM ports in Windows Device Manager. Figure 29 shows how the BoosterPack exports two virtual COM ports:
   a. XDS110 Class Application/User UART (COM$_{UART}$): Used for passing configuration data and firmware to the BoosterPack.
   b. XDS110 Class Auxiliary Data Port (COM$_{AUX}$): Used to send processed radar data output.
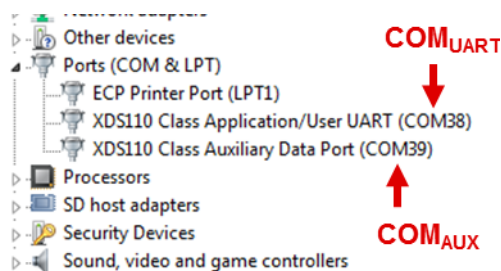
Figure 29. COM Port Identification

3. Put the BoosterPack in flashing mode by connecting jumpers on SOP0 and SOP2 as shown in Figure 30. Then power cycle the BoosterPack with SW2.
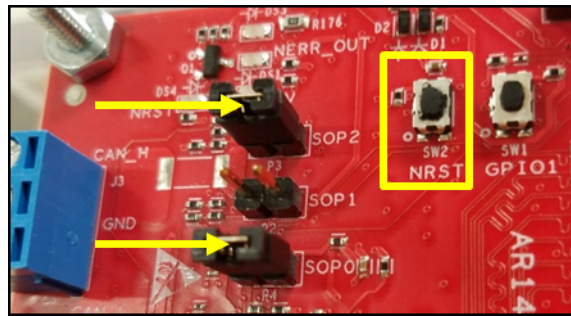
Figure 30. SOP0 and SOP2 Location

4. Download and open http://www.ti.com/tool/UNIFLASH.

5. In the *New Configuration* section, locate and select *IWR1642*.

6. Click *Start* to proceed.

7. 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.

8. In the *Program* tab, browse and locate the images (.bin file).

    a. Navigate to the TIDA-010022 firmware folder and select the xwr16xx_pcount_lab_es1.bin or the xwr16xx_pcount_lab_es2.bin file, depending on the board being used.

    b. ES2.0 IWR1642 BoosterPacks have an ES2.0 sticker. If the BoosterPack does not have a sticker it is an ES1.0 board.
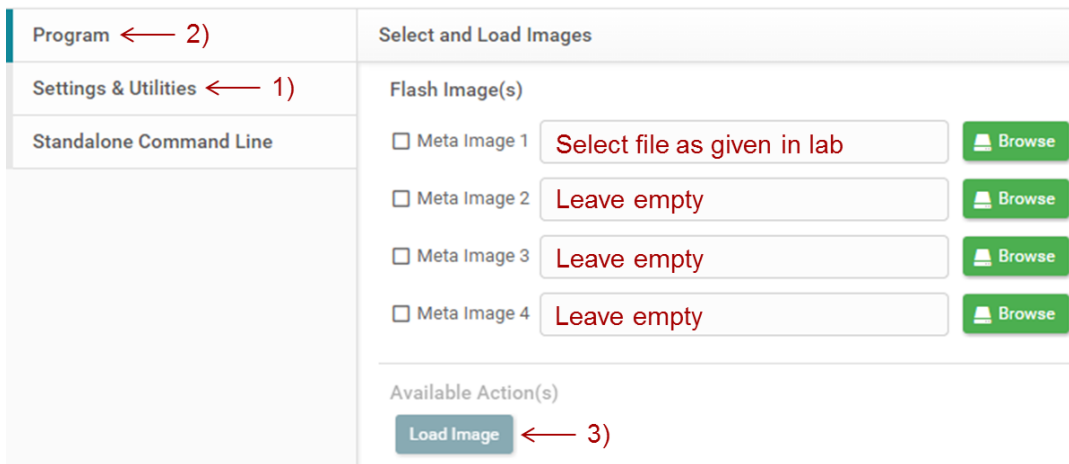
Figure 31. Loading Image

9. Power cycle the device and click on *Load Image*.

    a. Successful Flash Procedure: The UniFlash console should indicate: [SUCCESS] Program Load completed successfully

10. Power off the board and remove only the SOP2 jumper.

11. Make sure SOP2 has been removed and that the BoosterPack power cycled. This puts the board 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 (IWR1642 + CC1352) with 5.0-V power supply

3. 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.

4. Once the sensor has joined the network, its red LED will turn on.

5. Press "BTN-2 DIO14" (on collector node) to stop permitting new nodes from joining the network.

6. Once the sensor node has joined a network and has waited 20 seconds for the IWR1642 device to power up, the CC1352 device will configure and start the mmWave.

7. On both sensor and collector nodes, the "Green: DIO7" LED will toggle on each successful transmission between the nodes.
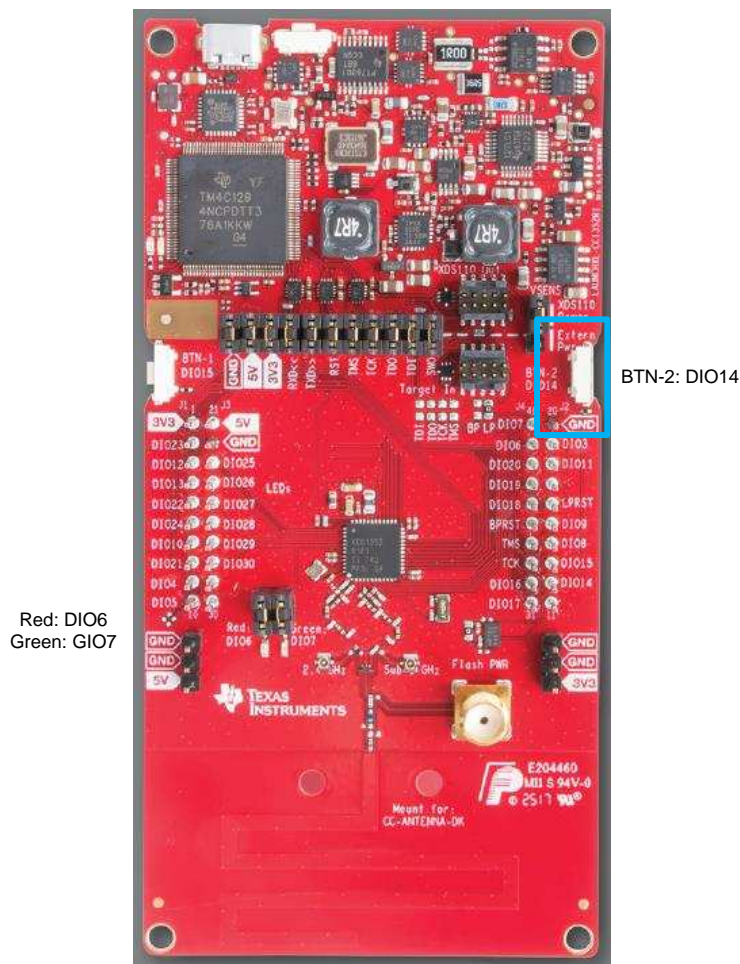
Figure 32. Location of LEDs and Button

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.
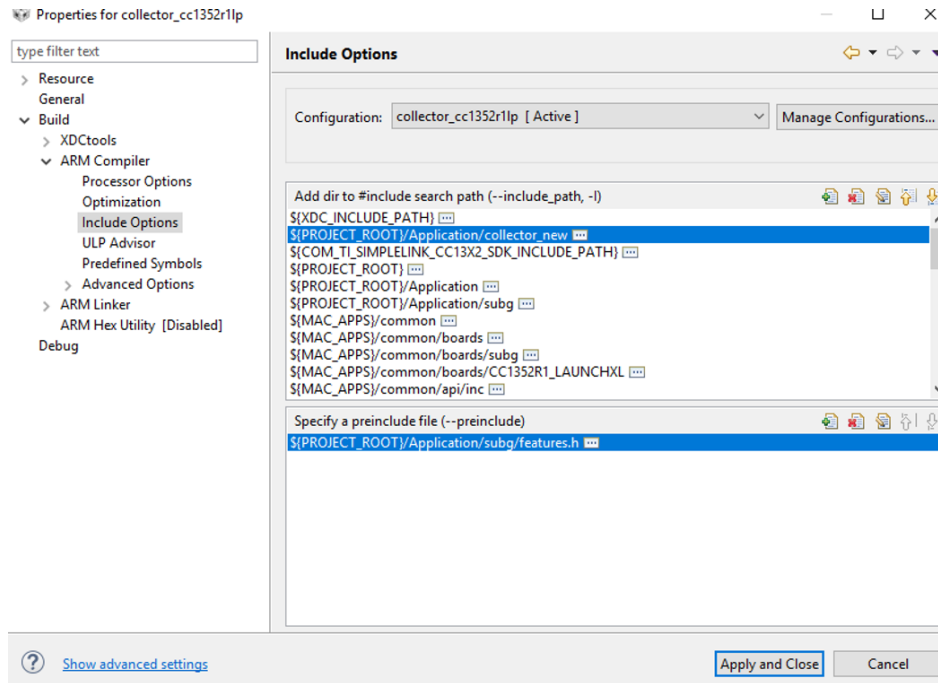
Figure 33. Adding Include Options for Collector Node
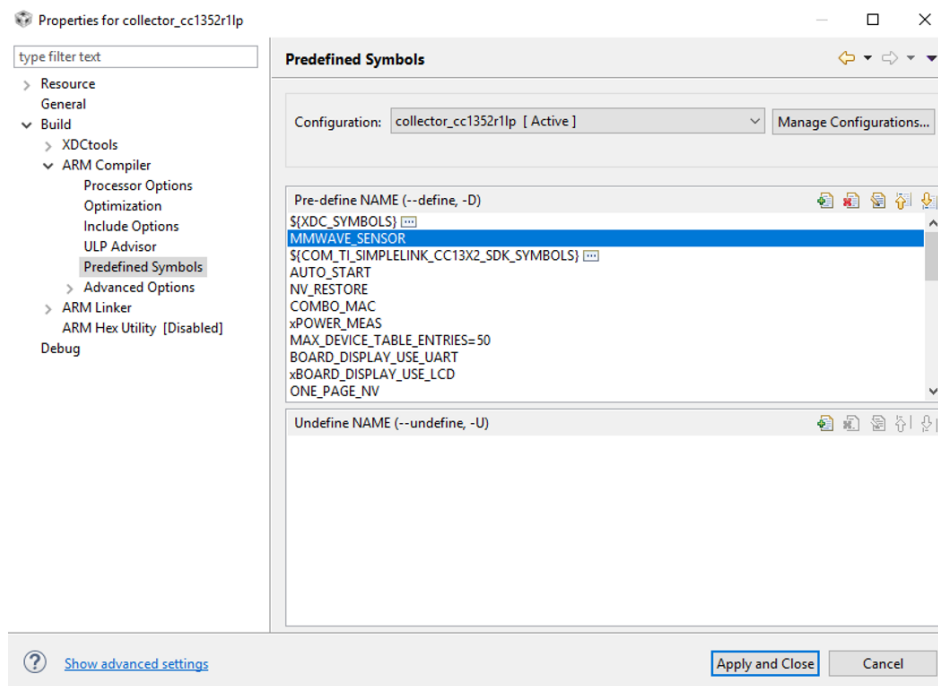
Figure 34. Adding Predefined Symbols for Collector Node

TIDUED6–August 2018
Submit Documentation Feedback
People Counting and Tracking Using mmWave Radar Sensor With Sub-1 GHz Reference Design
25

Copyright © 2018, Texas Instruments Incorporated
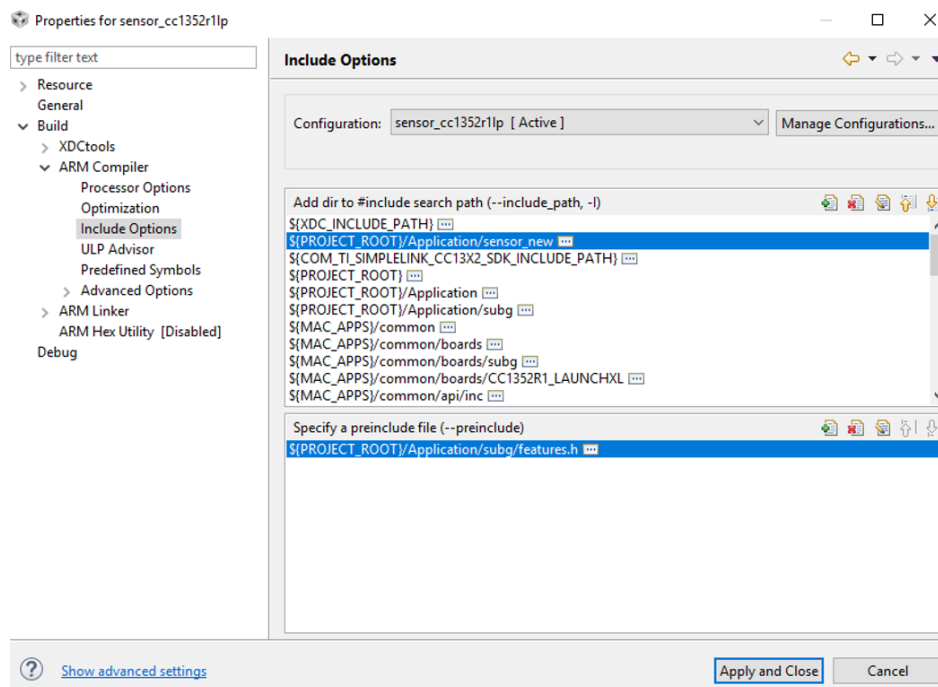
Figure 35. Adding Include Options for Sensor Node
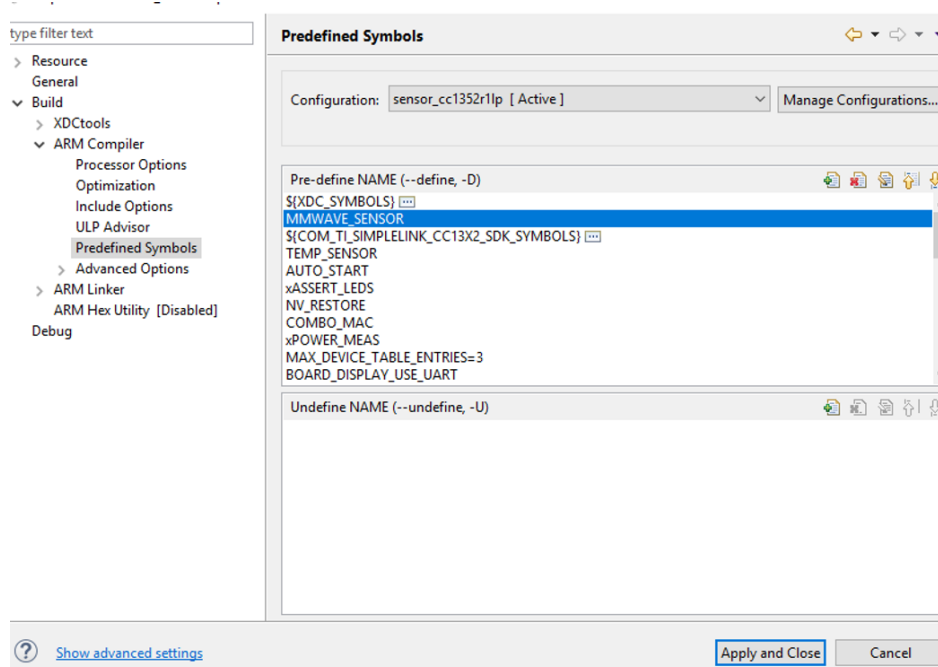
Figure 36. Adding Predefined Symbols for Sensor Node

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
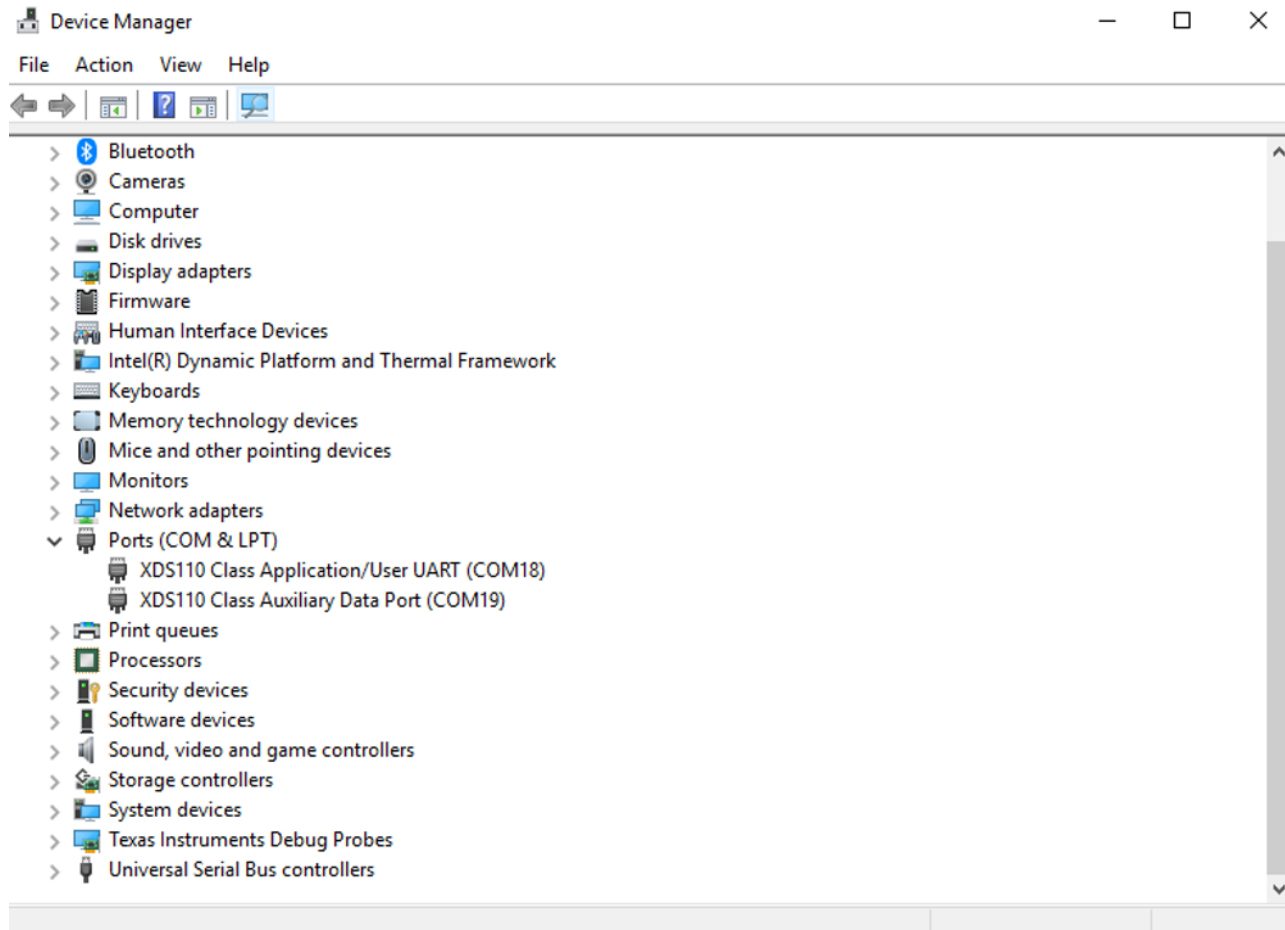
Figure 37. COM Ports

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 38 shows.
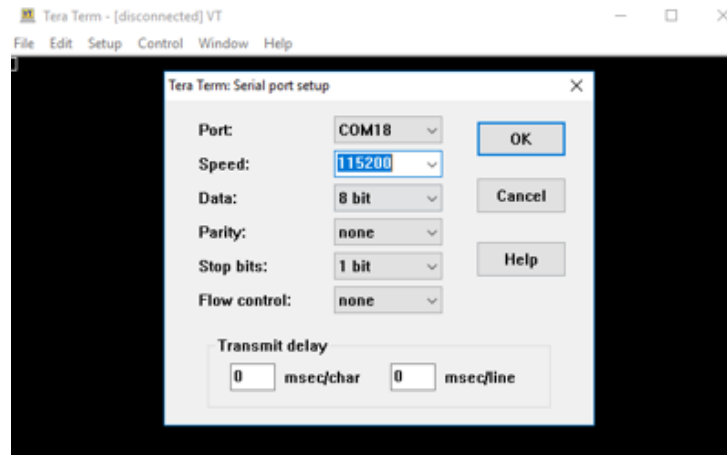
Figure 38. TeraTerm Setup

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.
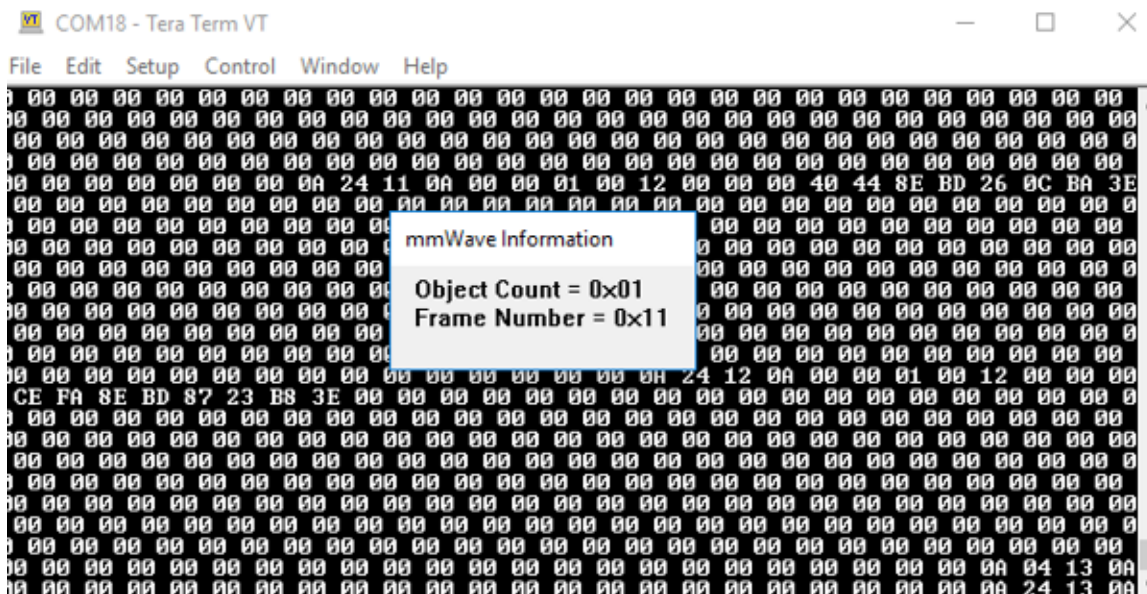
Figure 39. Macro

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.

TIDUED6–August 2018
Submit Documentation Feedback

```
const char *CommandList[] = {"dfeDataOutputMode 1\r",
                             "channelCfg 15 3 0\r",
                             "adcCfg 2 1\r",
                             "adcbufCfg 0 1 1 1\r",
                             "profileCfg 0 77 30 7 62 0 0 60 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 2\r",
                             "frameCfg 0 1 128 0 50 1 0\r",
                             "lowPower 0 0\r",
                             "guiMonitor 1 1 0 0\r",
                             "cfarCfg 6 4 4 0 0 16 16 4 4 50 62 0\r",
                             "doaCfg 600 1875 30 1\r",
                             "SceneryParam -6 6 0.05 4\r",
                             "GatingParam 4 3 2 0\r",
                             "StateParam 10 5 10 100 5\r",
                             "AllocationParam 250 0.01 5 1 2\r",
                             "VariationParam 0.289 0.289 1.0\r",
                             "PointCloudEn 0\r",
                             "trackingCfg 1 2 250 20 200 50 90\r",
                             "sensorStart\r"
};
```

Figure 40. mmWave Short Range Configuration Command List

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.
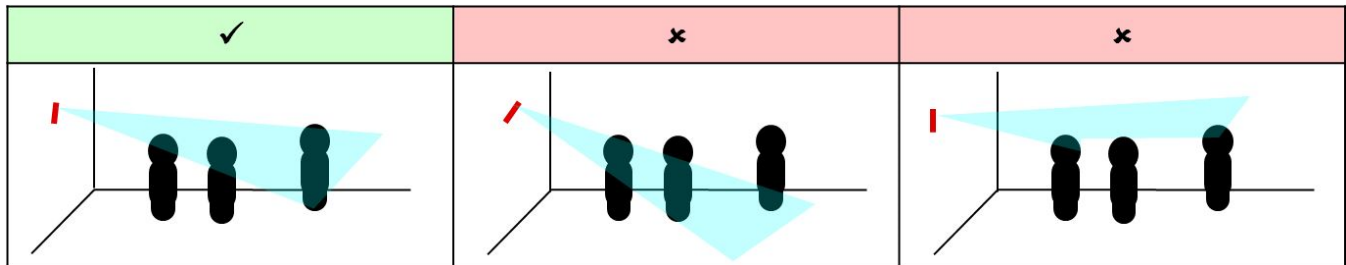
Figure 41. Sensor Node (IWR1642 + CC1352) Setup

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.
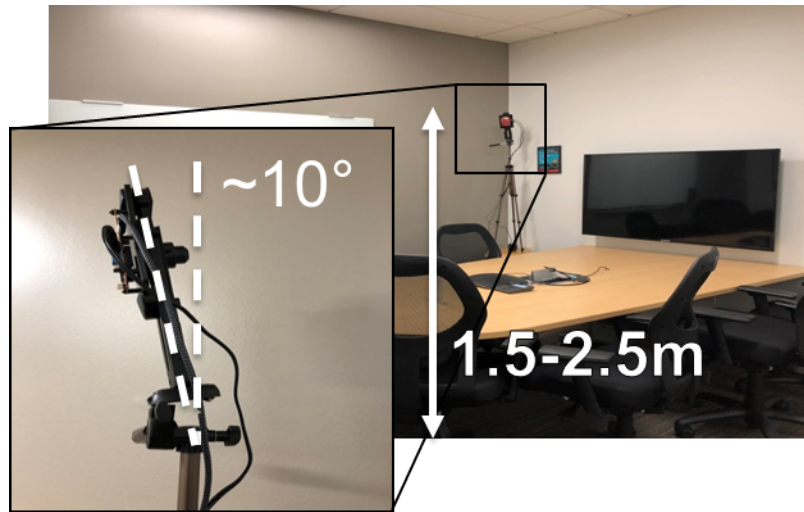
Figure 42. Setup with Clamp Mount

3.2.1.2 Power Consumption

The power consumption of the sensor (IWR1642 + CC1352) can be measured by connecting a bench power supply to the sensor.
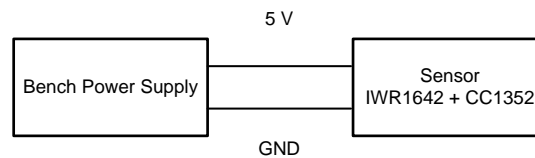
Figure 43. Power Consumption Measurement with Bench Power Supply Diagram

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 system power consumption for the sensor node is 3.6 W.

Power = input voltage × input current = 5 V × 0.72 A = 3.6 W
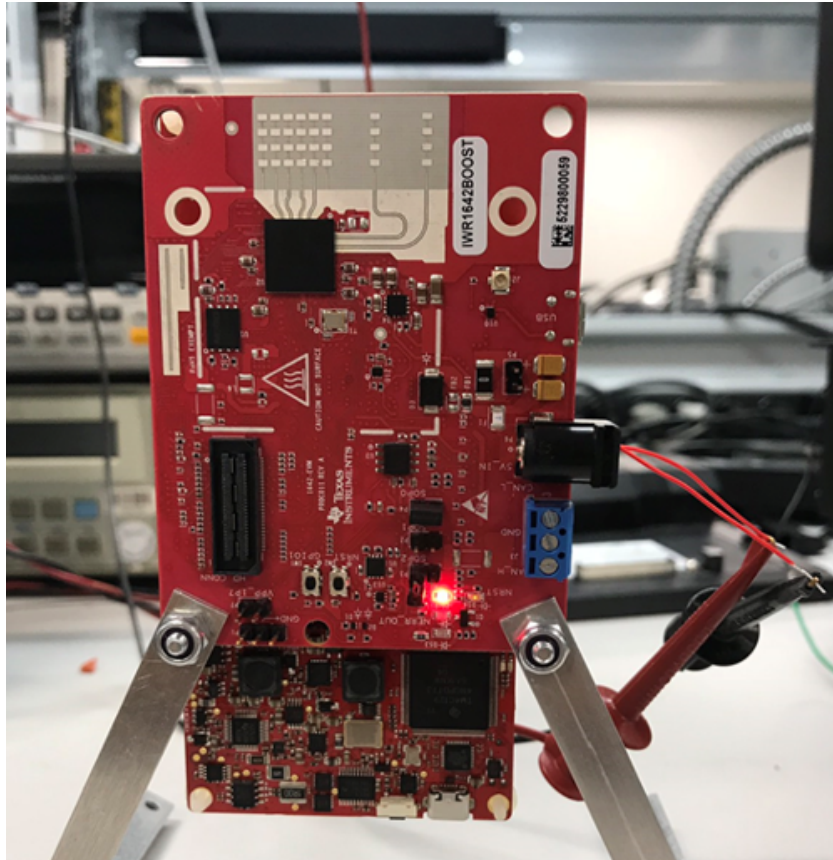
Figure 44. Measuring System Power Consumption in Sensor Node

Figure 45. Bench Power Supply for Measuring System Power Consumption

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 Documents

1. Texas Instruments, *IWR1642 Single-Chip 76- to 81-GHz mmWave Sensor Data Sheet*
2. Texas Instruments, *LP87524B/J-Q1 4-A + 2.5-A + Two 1.5-A Buck Converters With Integrated Switches Data Sheet*
3. Texas Instruments, *CC1352R SimpleLink™ High-Performance Dual-Band Wireless MCU Data Sheet*
4. Texas Instruments, *TPS7A8101 Low-Noise, Wide-Bandwidth, High PSRR, Low-Dropout 1-A Linear Regulator Data Sheet*
5. Texas Instruments, *TPS7A88-Q1 Automotive, Dual, 1-A, Low-Noise (4 μV_{RMS}) LDO Voltage Regulator*
6. Texas Instruments, *People Tracking and Counting Reference Design Using mmWave Radar Sensor Design Guide*

7 Trademarks

LaunchPad, E2E, SimpleLink, Code Composer Studio, BoosterPack are trademarks of Texas Instruments.
Altium Designer is a registered trademark of Altium LLC.
Bluetooth is a registered trademark of Bluetooth SIG, Inc..
IAR Embedded Workbench is a registered trademark of IAR Systems AB.
MATLAB is a registered trademark of The MathWorks, Inc..
Wi-Fi is a registered trademark of Wi-Fi Alliance.

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.

Submit Documentation Feedback