SmartSonic Presence Detection User Guide

Chirp Microsystems
2560 Ninth Street, Ste 220, Berkeley, CA 94710
U.S.A
+1(510) 640–8155
www.chirpmicro.com

Document Number: AN-000227
Revision: 1.0
Release Date: 7/8/2020

1 INTRODUCTION

This example project shows how to build and run an ultrasonic presence detection application using the Chirp SonicLib sensor API.

This is a companion document to AN-000214 Presence Detection, and AN-000226 CH201 Ultrasonic Presence Detection Reference Design User Guide. Application note AN-000214 focuses on the test results of the embedded static target rejection algorithm and the presence detection algorithm used in the CH201 presence detection evaluation module. Application note AN-000226 is a user guide for the CH201 ultrasonic presence detection reference design module. The Presence Detection example is a simple C application that demonstrates the use of motion detection to determine the presence of people and objects. The application uses the Chirp SonicLib API functions to initialize, configure, and operate one or more ultrasonic sensors. It uses special sensor firmware along with API functions which specifically support presence detection.

The application discovers what sensors are connected to the board, programs and configures them, and then triggers and displays ultrasonic measurement results through a serial connection. Each measurement result includes both a range to the detected object and a confidence level, based on the presence detection algorithm.

In this example, the application is built using Atmel Studio 7 for the Chirp SmartSonic evaluation board, which features an Atmel SAMG55 microcontroller. The SmartSonic board uses a sensor daughterboard that supports up to four ultrasonic sensors. The Presence Detection application can detect and run with either a single sensor or multiple sensors connected to the board, however the most common configuration is to use a single sensor.

The **source/application/smartsonic-presence-example/main.c** file is the main file in the application. It contains extensive comments explaining how the SonicLib interfaces are used. The **source/application/smartsonic-presence-example/app_config.h** file contains the setting for the application's overall timing.

1.1 REQUIRED EQUIPMENT

- SmartSonic evaluation board
- Chirp sensor daughter board
- One Micro-USB cable
- Internet connection (if downloading and installing files)

1.2 REQUIRED SOFTWARE PACKAGES

- **package-smartsonic-presence-example-v*X_X_X*.zip** (actual file name will include version number), includes:
    - Presence Detection application source files
    - Chirp SonicLib sensor API and driver files. The Presence Detection application requires SonicLib v2.0.5 or later.
    - Sensor firmware image files
    - Board support package files for Chirp SmartSonic board
    - Atmel Studio 7 project files to build the application
- Atmel Studio 7 integrated development environment – download from Microchip.com
- Terminal emulator of your choice (for example PuTTY or TeraTerm)

TABLE OF CONTENTS

LIST OF FIGURES

2 INSTALLATION / PREPARATION

- Download and install Atmel Studio 7 IDE.
- Download and install (unzip) the SmartSonic_Presence application to a project directory of your choice.
- Install terminal emulator.
- Connect the Chirp sensor daughterboard to the SmartSonic board. Be careful to align the white arrows.
- *Optional:* Using flat flex cables, attach additional off-board sensor(s) to the connectors on the daughterboard. If an offboard sensor is connected to the Sensor 0 connector (J6), you must set the switch on the side of the daughterboard to use the off-board sensor as Sensor 0 instead of the sensor on the daughterboard.
- Connect the SmartSonic board to a Windows PC with the USB cable. Configure the jumpers on the board as shown in the following photo with J1 in EDBG USB mode (short 3-4)

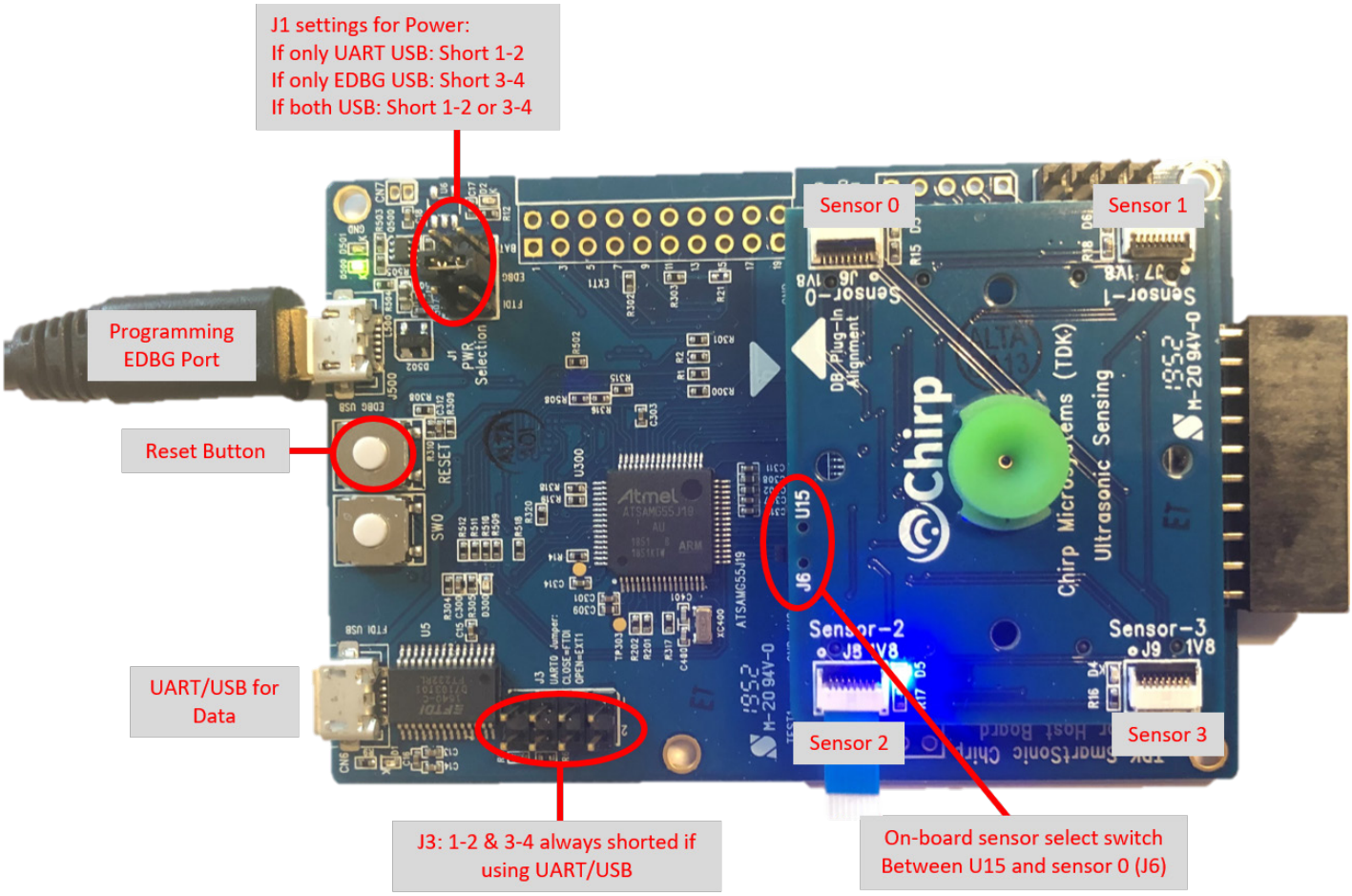

Figure 1. SmartSonic with CH201 Daughter Board

- Open Windows Device Manager, open the Ports (COM & LPT) list, and identify the COM port number assigned to the SmartSonic board. There will be one port associated with the SmartSonic board called“EDBG”. (A second COM port will also be assigned but is not used in this application).
    - The EDBG port is used to connect to the on-board debugger and for programming the board, as well as to display output from the program when it runs.

3 BUILDING THE PRESENCE DETECTION APPLICATION

- Open Atmel Studio 7
- Open the Presence Detection project:
    - Open **File** menu
    - Select **File > Open > Project/Solution…**
    - Select the **atmelstudio/smartsonic-presence-example/smartsonic-presence-example.atsln** file in the project directory.
    - Click **Open**. The program should locate the project files and display the name of the project.
- The Presence Detection project files are organized in three top-level sub-directories under **source**:
    - **algo**– contains **src** and **inc** sub-directories with the motion detection algorithm code and definitions
    - **application** – contains src and inc directories with the Presence Detection application
        - The **application/smartsonic-presence-example/main.c** file contains the entry point for the application along with various routines that demonstrate how to read and manage the Chirp sensor(s). See the comments in that file for detailed information about the operation of the application.
        - The **application/smartsonic-presence-example/inc/app_config.h** file specifies the overall measurement interval. The default is 100 ms (10 Hz sample rate).
    - **board** – contains support files for the Chirp SmartSonic board.
        - The main board support package routines, as defined in **drivers/chirpmicro/inc/chirp_bsp.h**, are in the **board/HAL/src/chbsp_chirp_samg55.c** file.
        - The **board/config/chirp_board_config.h** file is required by SonicLib. This file contains definitions for the number of possible devices and I^2C buses on the board and is used for static allocation of arrays.
    - **drivers/chirpmicro** – contains the SonicLib API and driver files, sensor firmware modules, and other distribution files from Chirp.
        - The **drivers/chirpmicro/inc** directory contains header files that must be included when building applications with SonicLib. In particular, the **drivers/chirpmicro/inc/soniclib.h** file contains the key definitions for the SonicLib API.
- Build the project:
    - Select **Build > Build Solution**

      The project should build successfully. The default build configuration is “Debug” so the build output files will be placed in the **Debug** sub-directory.

4 PROGRAMMING THE SMARTSONIC BOARD

- Setup SmartSonic board Jumper J1 as in Figure 1. If only one cable is used (connected to the EDBG port), pins 3 & 4 should be shorted.
- Connect the SmartSonic board to a Windows PC with EDBG USB cable.
- In Atmel Studio 7 select **Tools > Device Programming**.
- The Device Programming screen will appear:

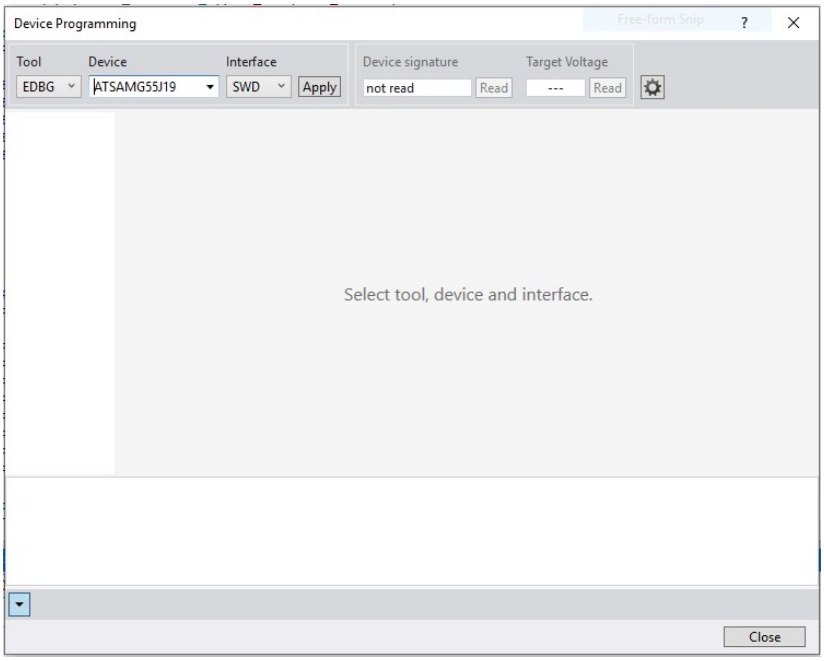

Figure 2. Device Programming Screen

- Verify that the tool is **EDBG**, device is **ATSAMG55J19**, and interface is **SWD**.
- Select **Apply**.
    - **Note**: Atmel Studio 7 may require you to update the EDBG debug interface firmware on the SmartSonic board before continuing. Follow the on-screen instructions to update the EDBG firmware.
- The Device Programming screen will prompt to set the programming clock frequency. Leave the clock frequency unchanged (use the default).
- Select **Read** near the **Device signature** field. The device signature bytes should be read and should not generate any error messages. The Device programming menu should look as follows:

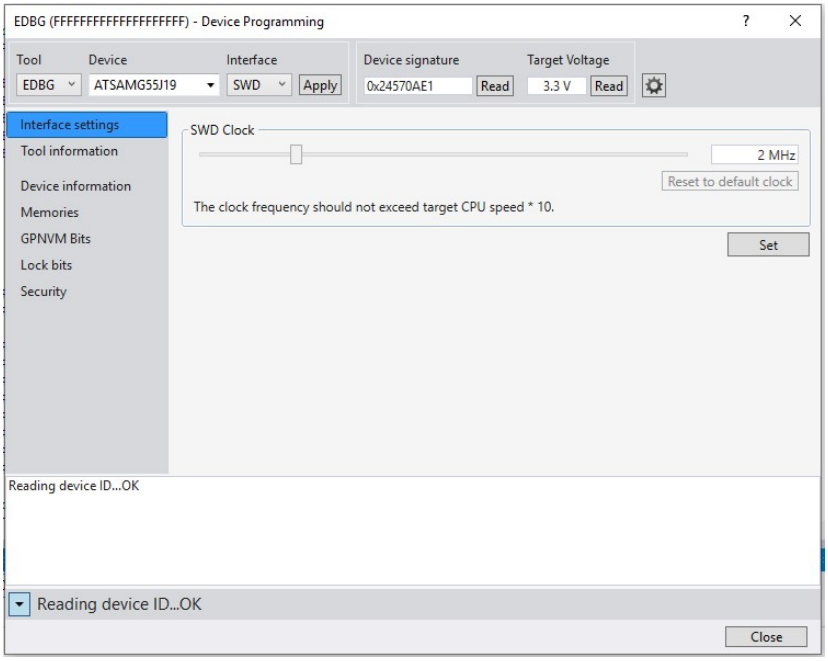

Figure 3. Device Signature and Target Voltage

- Select **Memories** on the Device Programming menu.
- The Device Programming menu will prompt for the name of the file to program:

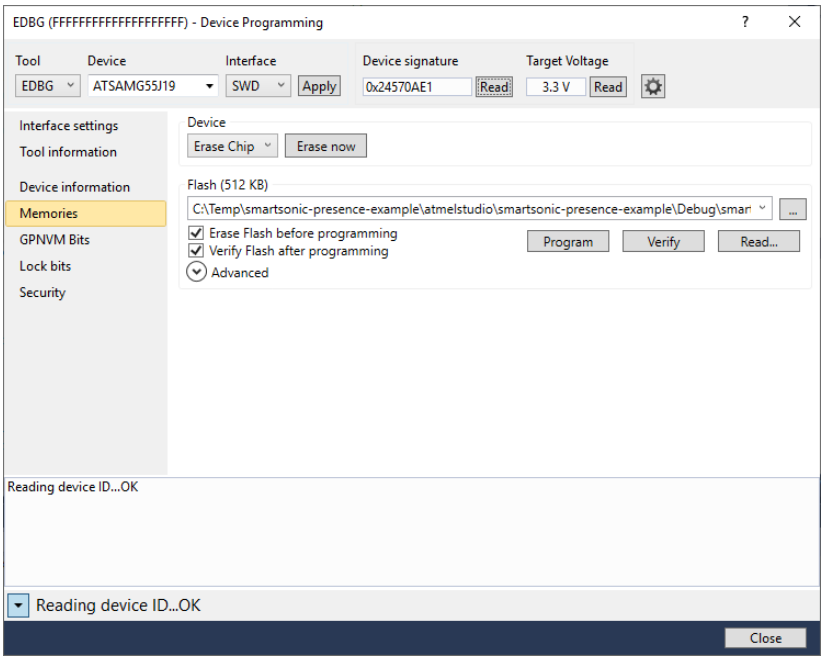

Figure 4. Programming Hex File

- From the Device Programming screen scroll to the project's Debug directory and select the s**martsonic-presence-example.hex** file.
    - **Note**: Alternately, you may use the **smartsonic-presence-example.elf** file, which contains symbolic debug information. (Both files are generated when you build the application. If you are simply running the Presence Detection application, and do not need to use the Atmel Studio 7 debugging features, it does not matter which file you use.)
- Select **Program**. Your SmartSonic board is successfully programmed when the Device Programming screen displays the "OK" messages shown below on the bottom left:

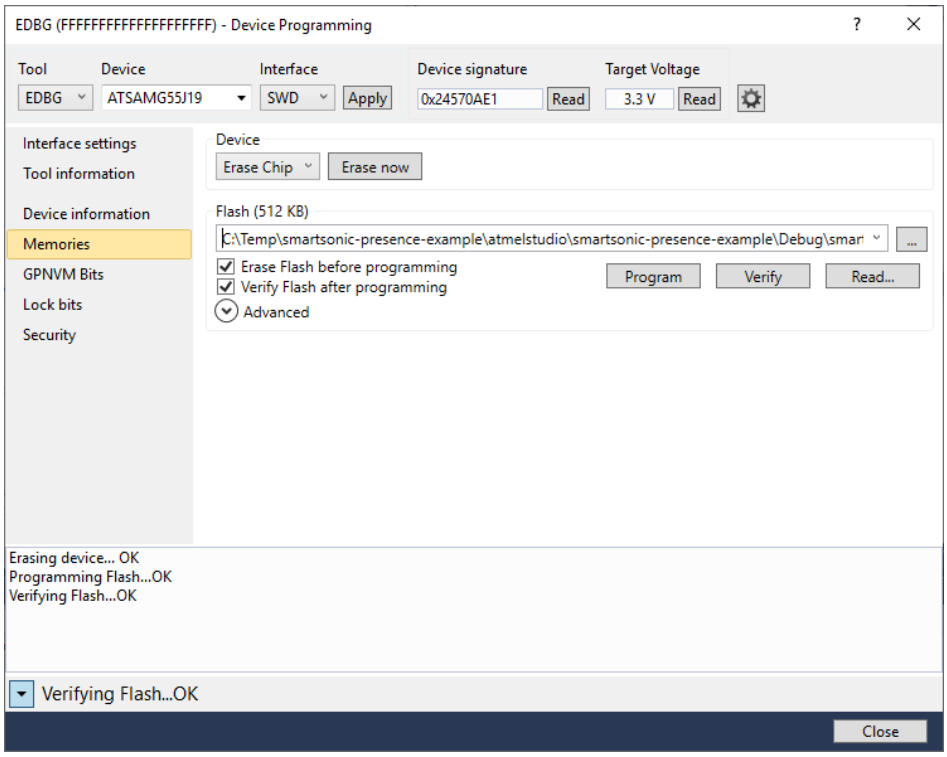

Figure 5. Successful Programming

5 *RUNNING THE PRESENCE DETECTION APPLICATION*

- Start the terminal emulator program and open/configure the COM port assigned to the SmartSonic board "EDBG":
    - **1000000 baud**
    - **8 bits data, no parity, 1 stop bit**
    - **New-line sequence = Line Feed only (no carriage return)**
        - PuTTY: Open **Terminal** configuration. Select **"Implicit CR in every LF"**.
        - TeraTerm: Open **Setup > Terminal**. Under **"New-line"** set **"Receive"** to **"LF"**.

- Reset the SmartSonic board using the board's reset button (next to the Programming EDBG connector).
- Status messages from the application will appear on the terminal output, followed by summary data from the sensor initialization (device frequency, etc.) and configuration (maximum range, etc.).
- Measurement data from each sensor device will then be output in a continuous loop. During each measurement cycle, a line of output is generated for each sensor. Each line of output includes:
    - Internal timestamp value
    - Sensor number
    - Boolean indicator of whether object presence was detected (0 = no presence, 1 = presence)
    - Range measurement, expressed in millimeters
    - Confidence level (expressed as a percentage). In the presence algorithm the detection of motion is based on a criterion passing a threshold. The confidence is derived from the following criterion value (the criterion value is defined in the algorithm and is not settable in the end application):
        - If the criterion < threshold, confidence is 0 (or very low)
        - If the criterion ≥ 10 x threshold, confidence is maximum (100%)
        - In between the confidence is linearized.
- If more than one sensor is used, an additional line of output is generated during each measurement cycle, with a merged result based on combined data from all sensors.

- Because the Presence Detection algorithm detects changes in the ultrasound environment, it is normal for the first few measurement cycles to report no presence detected. Figure 6 shows the typical application output during initialization, including the first measurement results.

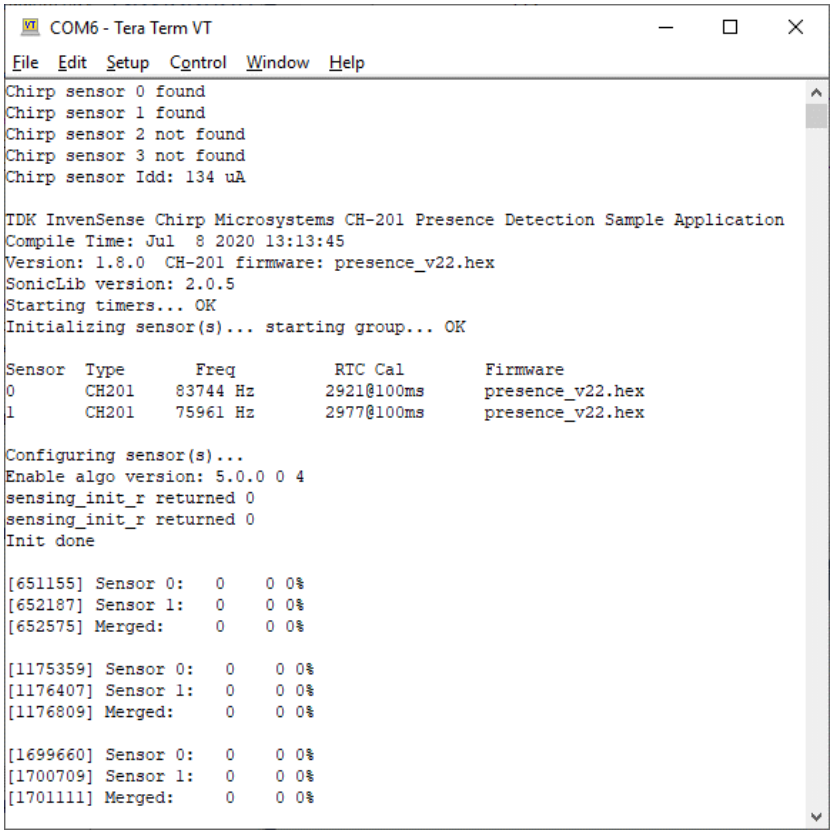

Figure 6. Typical Output During Initialization - 2 Sensors

After a baseline set of measurements has been made, the presence of objects can be detected. Figure 7 shows typical application output during operation, with two sensors. The “Merged” output lines indicate the overall Presence Detection result, based on data from both sensors.

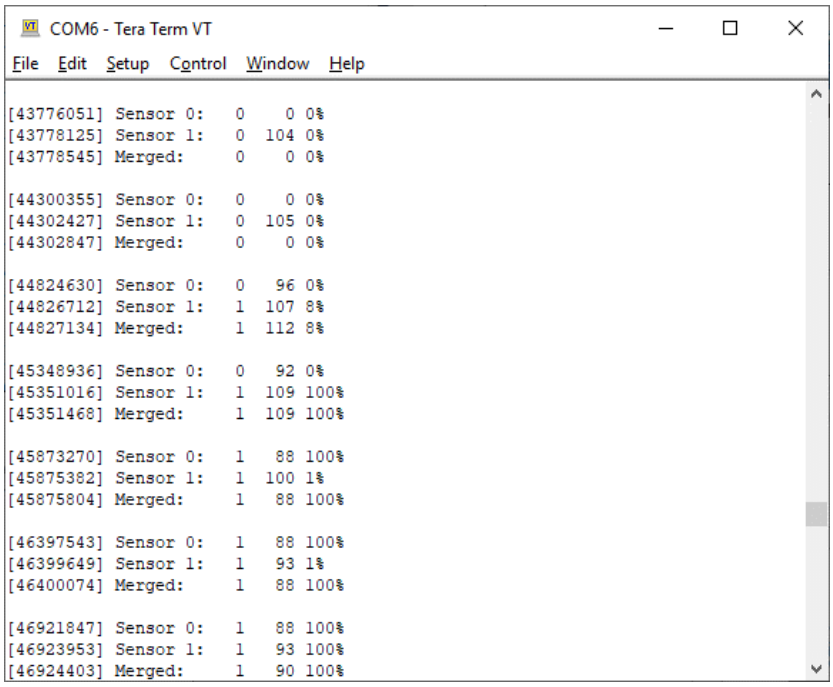

Figure 7. Typical Output During Operation - 2 Sensors

6 CHANGING THE PRESENCE DETECTION APPLICATION SETTINGS

The **app_config.h** header file contains symbolic definitions for parameters that affect the application execution. You may change the following definition to adjust the application's timing.

6.1 MEASUREMENT_INTERVAL_MS

Specifies how often a new measurement cycle will be initiated, in milliseconds. Default = 100 ms (10Hz sample rate).

This value is used in *main()* as a parameter to the *chbsp_periodic_timer_init()* function. Once set, the timer will generate an interrupt every MEASUREMENT_INTERVAL_MS milliseconds, which will cause the *periodic_timer_callback()* routine in **main.c** to execute. That routine then triggers a new measurement cycle by calling *ch_group_trigger()*.

7 REFERENCE DOCUMENTS

- AN-000214 – Presence Detection
- AN-000226 – CH201 Ultrasonic Presence Detection Reference Design User Guide

8 REVISION HISTORY

REVISION DATE	REVISION	DESCRIPTION
7/08/2020	1.0	Initial version.

TDK Chirp Microsystems