CH-101 Example Driver Hands On

Chirp Microsystems
2560 Ninth Street, Ste 200, Berkeley, CA 94710 U.S.A
+1(510) 640–8155
www.chirpmicro.com

Document Number: AN-000154
Revision: 1.0
Release Date: 08/02/2019

CH-101 Example Driver Hands On

INTRODUCTION

This exercise demonstrates how to build and run a simple ultrasonic sensing application using the Chirp CH-101 example driver. The application runs on the SmartSonic evaluation board, which uses an Atmel SAMG55 microcontroller. The application can run with either a single sensor or multiple sensors connected to the board.

REQUIRED EQUIPMENT

- SmartSonic evaluation board
- CH-101 sensor daughter board
- Two Micro-USB cables
- Internet connection (if downloading and installing files)

REQUIRED SOFTWARE PACKAGES

- Chirp CH-101 driver example – Atmel Studio 7 project files
- SmartSonic_ExampleDriver.zip
- Atmel Studio 7
- Terminal emulator of your choice (for example PuTTY or TeraTerm)

CH-101 Example Driver Hands On

TABLE OF CONTENTS

INTRODUCTION 1
REQUIRED SOFTWARE PACKAGES 2
1 INSTALLATION / PREPARATION 4
2 BUILDING THE EXAMPLE APPLICATION AND DRIVER 5
3 PROGRAMMING THE SMARTSONIC BOARD 6
4 RUNNING THE EXAMPLE APPLICATION 9
5 USING ALTERNATE SENSOR FIRMWARE IMAGES 10
6 REVISION HISTORY 11

LIST OF FIGURES

Figure 1. SmartSonic with CH-101 Daughter Card 4
Figure 2. Device Programming Screen 6
Figure 3. Programming Clock Frequency 7
Figure 4. Device Signature and Target Voltage 7
Figure 5. Programming Hex File 8
Figure 6. Successful Programming 8
Figure 7. Example Application Output 9

1 INSTALLATION / PREPARATION

- Download and install Atmel Studio 7 IDE.
- Download and install the CH-101 driver example.
- Install terminal emulator.
- Connect the CH-101 sensor to the DVB board with flex cable(s).
- Connect the SmartSonic board to a Windows PC with the two USB cables.

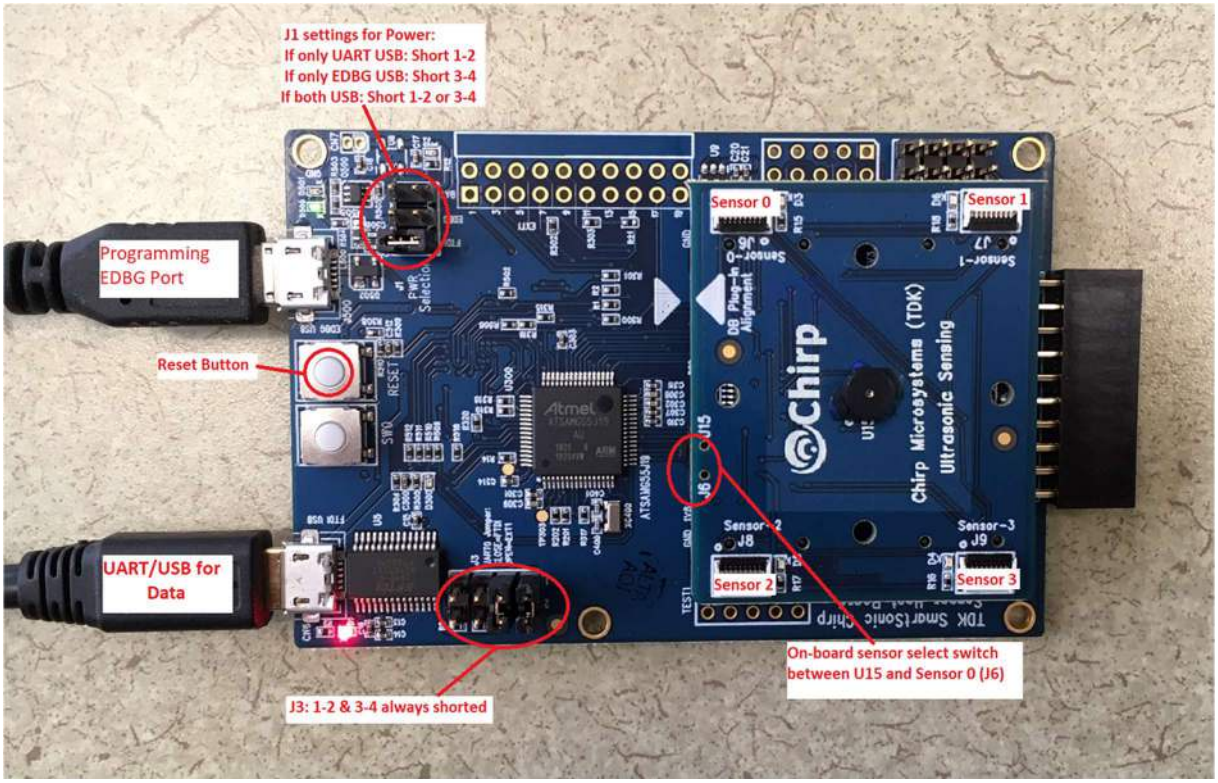

Figure 1. SmartSonic with CH-101 Daughter Card

- Open Windows Device Manager look in the Ports (COM & LPT) list and identify the COM port numbers assigned to the SmartSonic board. There will be two ports associated with the SmartSonic board: EDGB and USB Serial Port. You will need to specify the USB serial port number when using a terminal emulator to display output.

2 BUILDING THE EXAMPLE APPLICATION AND DRIVER

- Open Atmel Studio 7
- Open the driver example project:
  - Open **File** menu
  - If you are using a new System Workbench installation:
    - Select **File > Open > Project/Solution…** > Select the **SmartSonic_ExampleDriver.atsln** file in the project directory.
    - Click **Open**. The program should locate the project files and display the name of the project.
- Build the project:
  - Select **Build > Rebuild Solution**

  The project should build successfully (with some warnings). The default build configuration is "Debug" so the build output files will be placed in the Debug sub-directory.

  - Warning: **Do NOT** select "**Clean Solution**" or otherwise clean the build output files. Doing so will remove some build definition files, and the next build(s) will fail. If you have this problem, run "**Build Project**" several (3 or 4) times until the build succeeds – the repeated builds will ultimately reconstruct the missing files.

3 PROGRAMMING THE SMARTSONIC BOARD

- Connect the SmartSonic board to a Windows PC with the two USB cables
- Select **Tools > Device Programming**.
- The Device Programming screen will appear:

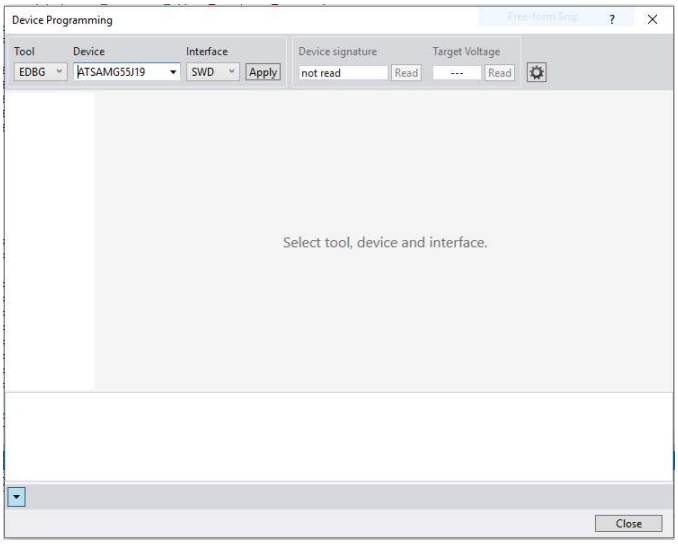

Figure 2. Device Programming Screen

- 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:

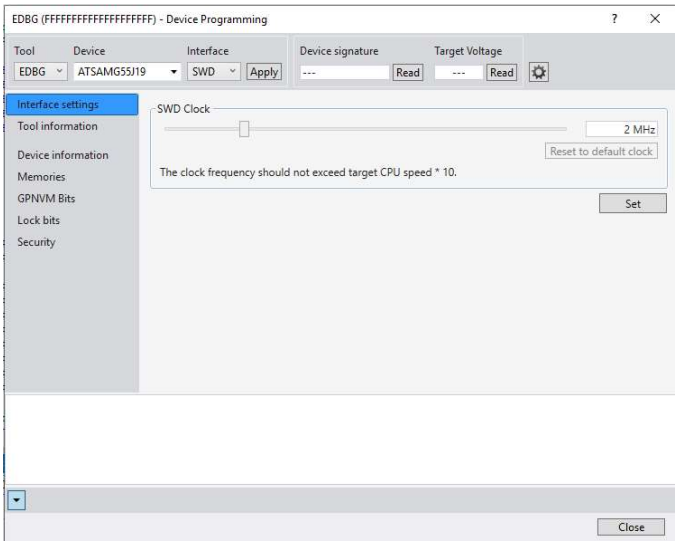

Figure 3. Programming Clock Frequency

- Leave the clock frequency at the default and select **Set**.
- Select **Read** near the **Device signature** field.
- The Device programming menu should look as follows:

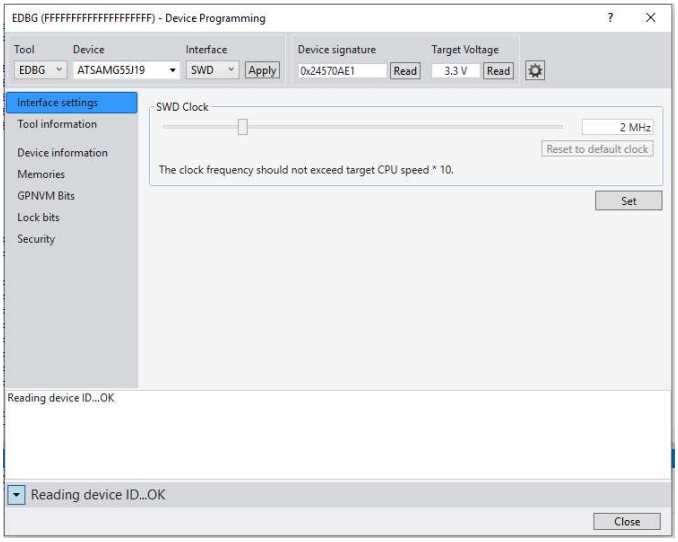

Figure 4. Device Signature and Target Voltage

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

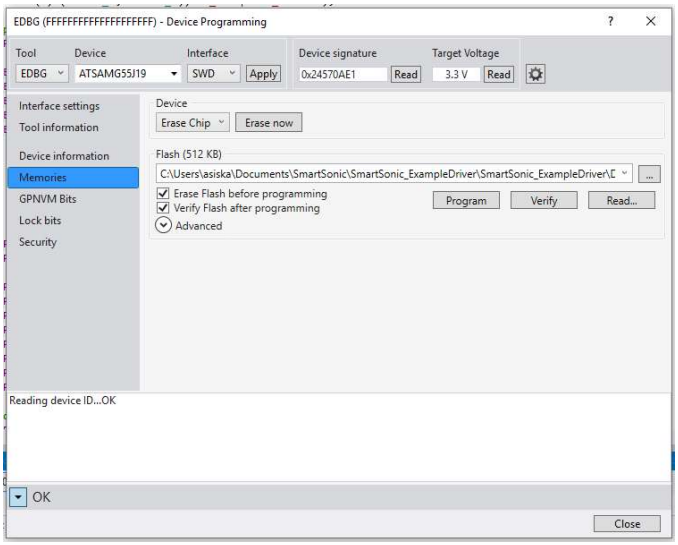

Figure 5. Programming Hex File

- From the Device Programming screen scroll to the project's debug directory and select the **SmartSonic_ExampleDriver.hex** file.
- Select **Program**. Your SmartSonic board is successfully programmed when the Device Programming screen displays the following on the bottom left:

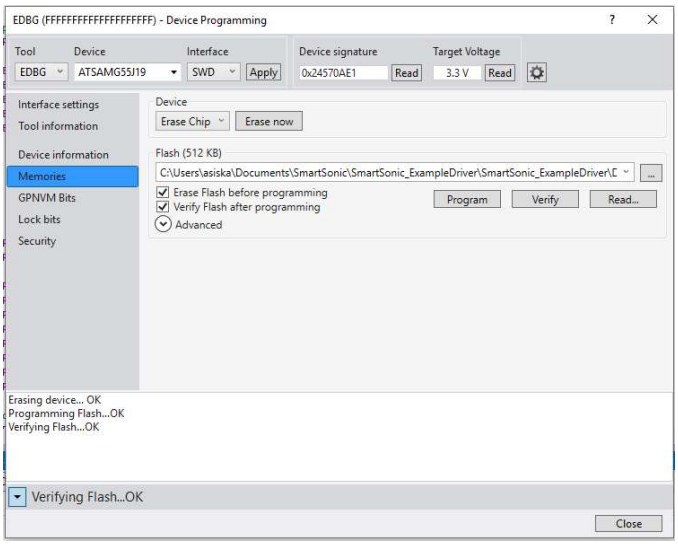

Figure 6. Successful Programming

4 RUNNING THE EXAMPLE APPLICATION

- Start the terminal emulator program and open/configure the COM port assigned to the SmartSonic board:
  - **921600 baud** (non-standard but should work with most terminal emulators)
  - **8 bits data, no parity, 1 stop bit**
  - **New-line sequence = Line Feed only (no carriage return)**
- Reset the SmartSonic board using the board's reset button (next to the Programming EDGB connector).
- Status messages from the application will appear on the terminal output, followed by summary data from the sensor initialization (device frequency, etc.).
- Range measurement data from the sensor device(s) will be output in a continuous loop. Note that distance measurements are expressed in **millimeters**.

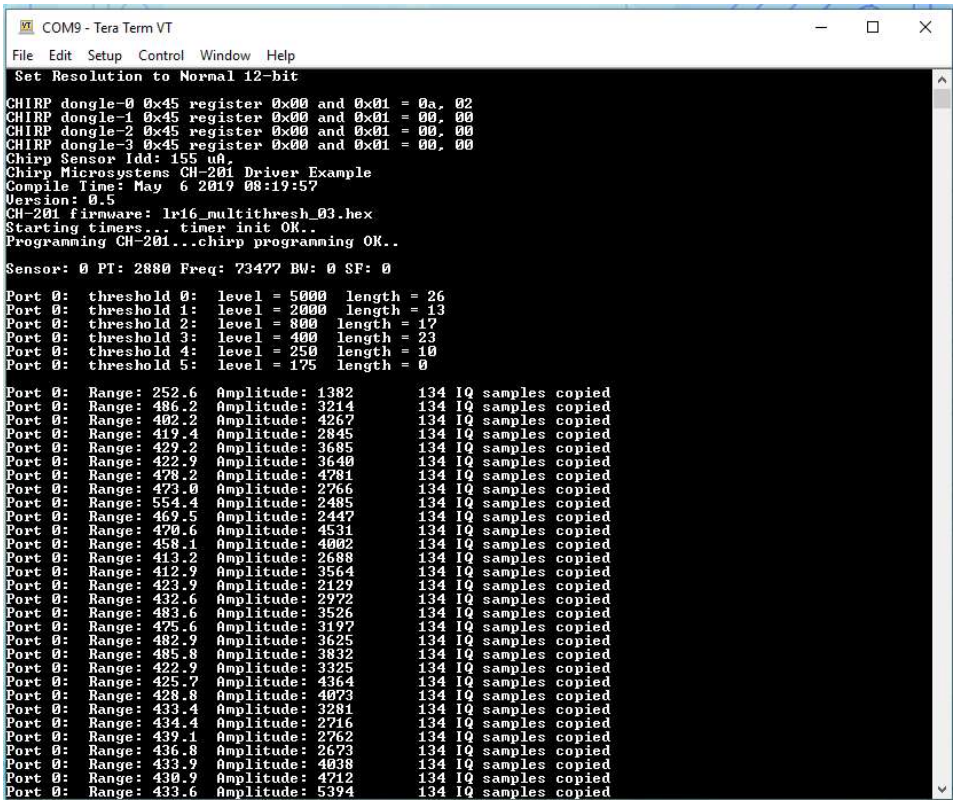

Figure 7. Example Application Output

5 USING ALTERNATE SENSOR FIRMWARE IMAGES

By default, the example application uses standard Chirp "GPR" (General Purpose Rangefinder) sensor firmware. This firmware provides good performance over various distances under most conditions. Normally, the standard firmware is recommended, and no changes are required.

For special applications using CH-101 sensors, two options are available. Either or both may be selected by defining symbols in the **inc/main.h** header file.

To use a special firmware version that is optimized for short-range performance, define **USE_SHORT_RANGE** in **main.h**. This special firmware provides more measurement resolution at close distances. However, the maximum range for the sensor is reduced significantly (by a factor of 4).

To use a special firmware version that uses less power when the device is idle, define **USE_LOW_POWER** in **main.h** (at or near line 107). This version has a significant disadvantage, however – the performance of the CH-101 sensor is negatively affected by sunlight. Therefore, it is not recommended for most applications.

6 REVISION HISTORY

Revision Date	Revision	Description
08/02/2019	1.0	Initial Release

TDK Chirp Microsystems