MicroIAS (Integrated Alarm System)

Sensor Fusion Based Alarm System

About this document

Scope and purpose

The document serves as a guide for using the Infineon Integrated Alarm System (IAS) in customer end applications for development or deployment. Infineon's MicroIAS (an embedded form-factor board) has been used as a reference hardware platform in order to qualify the IAS software algoritms. From an IAS software algorithm usage standpoint, this document should also serve as a user guide for other platforms such as XMC. Hardware details including set-up and API documentation for the XMC platform are part of XMC IAS Package.

Intended audience

The intended audience are customers who would use Infineon's IAS Solution.

Production Release Version

Table of contents

1 What is the MicroIAS?

Infineon's MicroIAS is a small form factor reference board (Figure 1) running Integrated Alarm System (IAS) software algorithms. It employs a high accuracy digital barometric pressure sensor (DPS310), a MEMS digital microphone (IM69D130) with high SNR and a powerful small form-factor ARM M4F core microcontroller (MCU) from Quicklogic that processes sensor data from the microphone (listens to glass break) and the pressure sensor (monitors changes in pressure after a glass break happens) and runs sensor fusion algorithms to trigger an alarm.

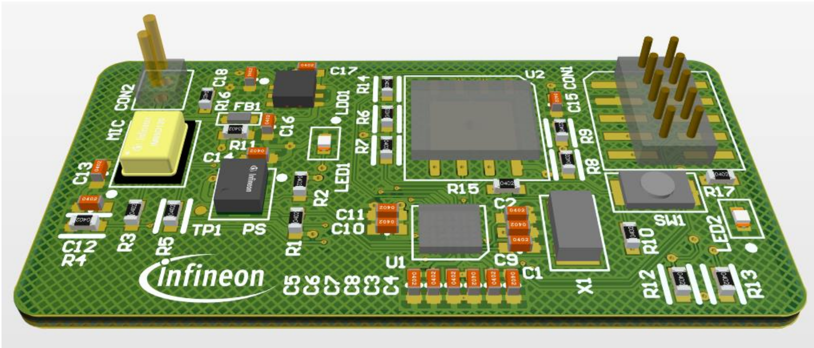

Figure 1 MicroIAS Board – 3D Render

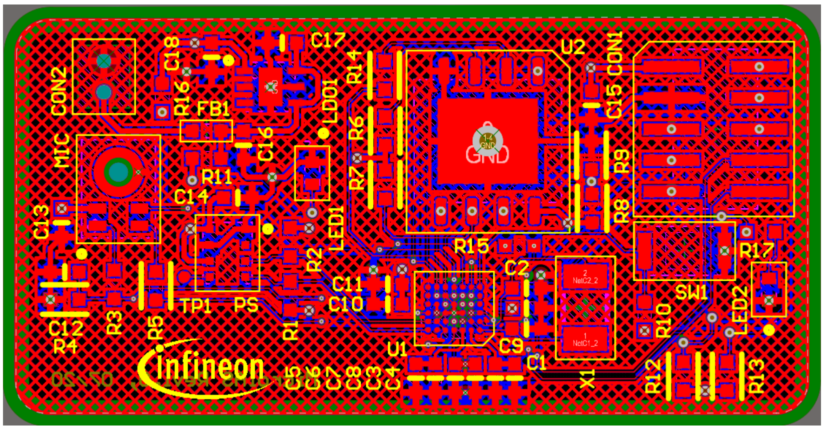

Figure 2 MicroIAS Board – Top View

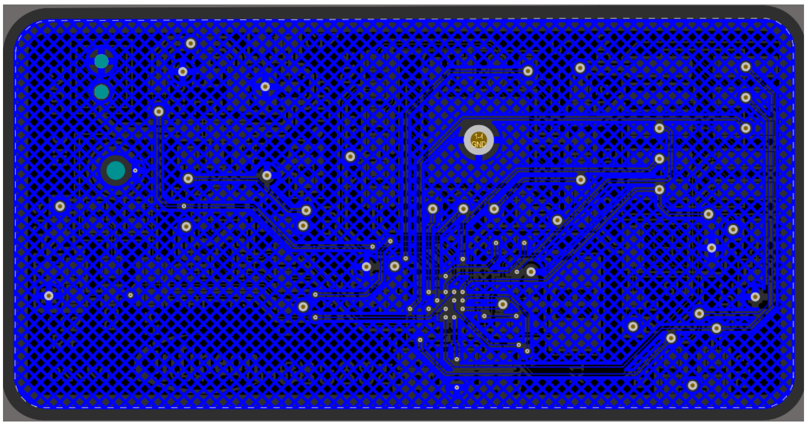

Figure 3 MicroIAS Board – Bottom View

2 MicroIAS System

2.1 Key Features and Benefits

Table 1 Key features of the MicroIAS

Feature	Benefit
Up to 8m (26 feet) coverage	Can monitor living spaces/room sizes of up to of 50m² (~ 540sqft)
Battery operated	Runs up to 5+ months from a single 1,600mAh CR123A* Li cell
High-performance compact solution	1.8V system operation on a 32mm x 16mm PCB with ARM-core M4F MCU and on-board Flash supports external bootloading mechanism
Utilizes digital microphone IM69D130 and pressure sensor DPS310	Fusion based 2-sensor monitoring eliminate false alarms
Flexible interface capabilities	Small form-factor allows developers to tap into JTAG/SWD ports and UART interface via convenient 2x5 micro connector
Sensitivity Modes for glassbreak events and intruder detection	High, Low (Section 3.2) **

**refer to section 2.8 for a more detailed discussion on power consumption*
*** refer to API Documentation "**ifx_alarm_set_configuration**"*

2.2 Mounting Recommendation

Please refer to the *Micro IAS Mounting Recommendation* documents for further details.

2.3 Overview

The MicroIAS board allows a software developer access to all essential interfaces such as UART, SWD (CON1 connector for development and debug located on the upper right corner of the board), power supply, a manual reset (SW1) and it provides visual aids (LED1 and LED2) for the user to identify power applied (LED = green) and alarm triggered (LED = red).

An external power supply with a 3.6V to 6V range must be applied to GND and VCC on CON2 (upper left corner of the board) to produce the board's needed power supply. LED1 (green), lights up when the board has received the proper supply voltage. On the MicroIAS, the voltage is routed through a small LDO to generate the 1.8V needed to power sensors (MIC and PS) as well as SPI Flash and MCU (Figure 4). Please refer to the *Hardware Reference Design* document for details on schematic, BOM and gerber files.

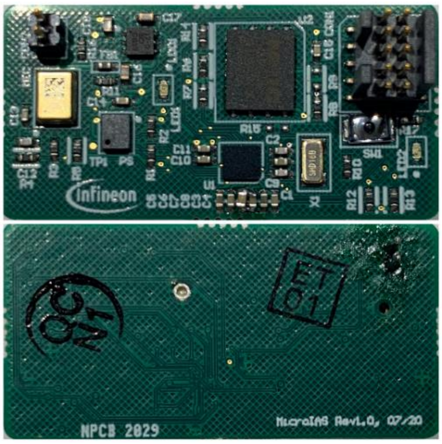

Figure 4 MicroIAS – Assembled Module, Top/Bottom View

2.4 Developer's Connectivity Guide

Refer to the following section and figures for details on how to connect the MicroIAS for testing, debugging and re-programming purposes. Power to the MicroIAS is applied through CON2 in the upper left corner of the board (Figure 5). A matching connector cable is included for this purpose.

Power is also routed to the IDC connector CON1 in the upper right corner of the MicroIAS board (Figure 5). This carries importance, if the MicroIAS is used in combination with the breakout board that is also included in the evaluation kit's contents.

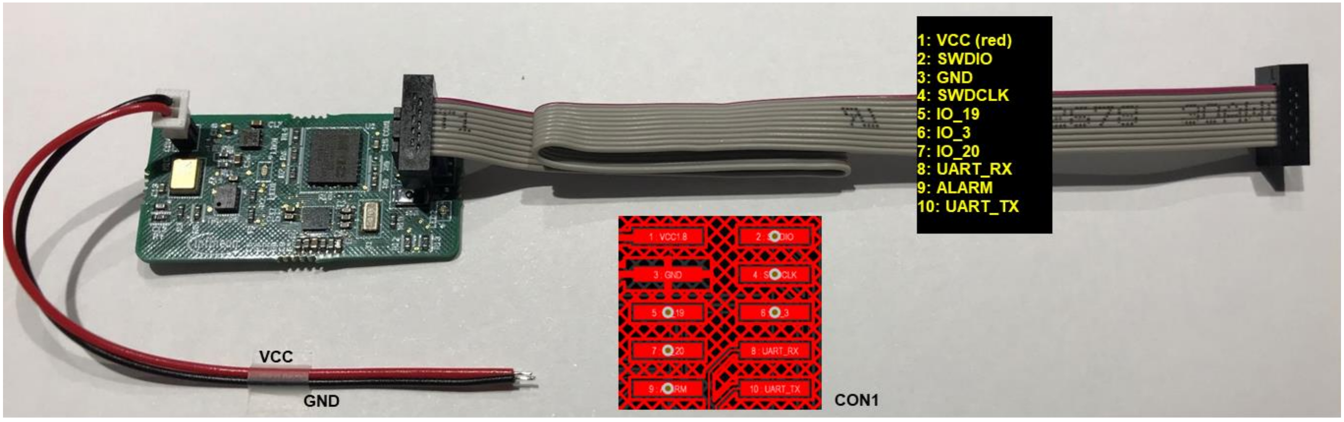

Figure 5 MicroIAS – Power & Communications Interfaces

For programming purposes or GUI use (UART communications), the kit offers a breakout board that allows the MicroIAS to communicate either with a Segger JTAG programmer or a USB-to-UART bridge. For re-programming via JTAG, the breakout board offers jumpers and connectors, which must be shunted accordingly to enter the proper programming mode. See Figure 6a for more details.

The left side of the breakout board connects to the MicroIAS (IAS_CON), while the right side of the board connects to the JTAG side (JTAG_CON), if re-rogramming the MicroIAS (e.g. for GUI use) is desired. CON3 in the center of the board serves as "shunting hub" between the two sides of the board. How to program/re-program the MicroIAS will described later in this section.

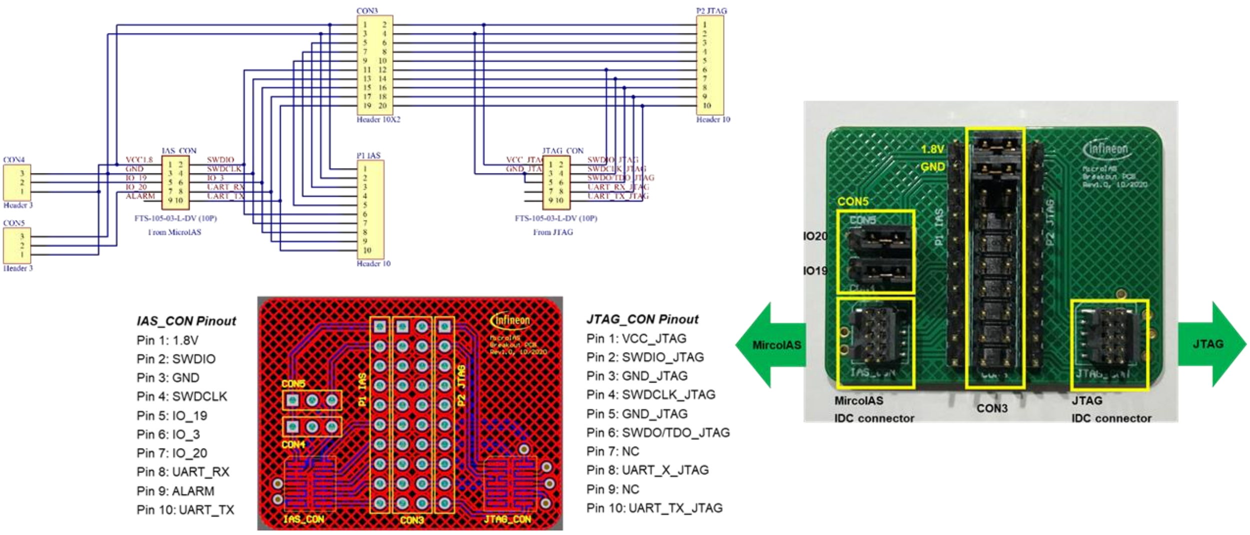

Figure 6a: MircoIAS Breakout Board

2.5 Default Operation

Note that the MicroIAS works out of the box and doesn't need to be programmed. All it requires is a power supply. It has been programmed for use with the on-board red LED, which flashes once for an intrusion and twice for a glass break detection event.

2.6 Re-Programming the MicroIAS

Re-programming is ony required, if developers would like to implement their own algorithms, desire a factory reset or would like to use the MicroIAS with a GUI (instead of LED-only) or other UART communictions (e.g. TeraTerm). For that purpose, new firmware will need to be flashed onto the MCU (see section 4.3). By default the breakout board has shunts installed across pins 1-2 (1.8V) and 3-4 (GND) on CON3 as well as pins 2-3 (GND) on CON4 (IO_19) and CON5 (IO_20).

Ensure that your system is properly setup for programming and UART communications. Figures 6b and 6c depict the breakout board setup for both options respectively.

- For re-programming the MicroIAS, use one of the 2 IDC cables to connect the IAS_CON (breakout board) to CON1 (MircoIAS) and the 2nd IDC cable to connect JTAG_CON (breakout board) to the IDC connector on the J-LINK Adapter CortexM board plugged into the JTAG programming tool from Segger (www.segger.com).
- Although all IDC connectors on breakout and MicroIAS boards provide proper directionality (they can only be plugged in one direction), make sure that the IDC cable pin 1 indicators (red wire of the ribbon cable) line up as shown in Figure 6b.
- For programming, shunts across CON4 and CON5 must be installed across pins 1-2, so move them from their original shipping position (2-3) to 1-2.
- Shunts across pins 1-2 and 3-4 of CON3 ensure power and ground connections between both systems are established. Ensure you make the SWD interface connections between MicroIAS and JTAG by installing shunts across pins 11-12 (SWDIO) and 13-14 (SWDCLK) on CON3.
- Once all the needed connections and shunt installations have been completed, follow the step-by-step instructions for re-programming in section 4.3.

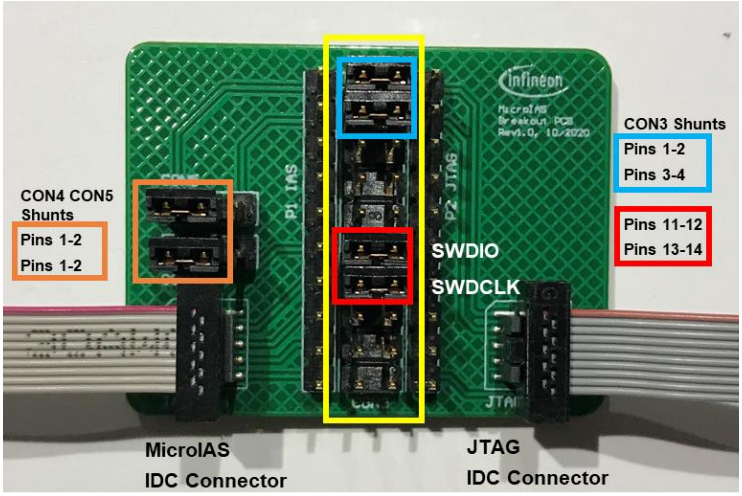

Figure 6b: Breakout Board Setup for Programming

- After having followed the programming steps for e.g. flashing the proper FW for UART communication (use /w TeraTerm or Infineon's MicroIAS GUI), you may set up your system for UART communications.
- Figure 6c depicts how to setup/modify the shunts and connections for proper UART communications.
- Remove shunts from pin 11-12 (SWDIO) and 13-14 (SWDCLK) of CON3. They are no longer needed.
- Keep shunts across 1-2 (VCC) and 3-4 (GND) in place.
- Ensure that after programming and resetting the MicroIAS system and before using it with UART communications, CON4 and CON5 have their shunts installed across 2-3 (GND).

2.7 Operating the MicroIAS via UART Interface

For GUI and terminal communications, the MicroIAS uses UART transmit and receive pins. Ensure connections between your UART bridge (see section 4.1, step 7 for recommendation of a suitable USB-to-UART bridge) and the MicroIAS are made with female-to-female jumper cables.

- UART_RX and UART_TX can be accessed on header P1_IAS (pins 9 and 10 respectively) or on CON3 (pins 17 and 19 respectively).
- Note, headers P1_IAS (MicroIAS) and P2_JTAG (JTAG side of the board) are connected to CON3 on their respective sides of the board, to allow for the convenience of an additional probing pin, if needed.

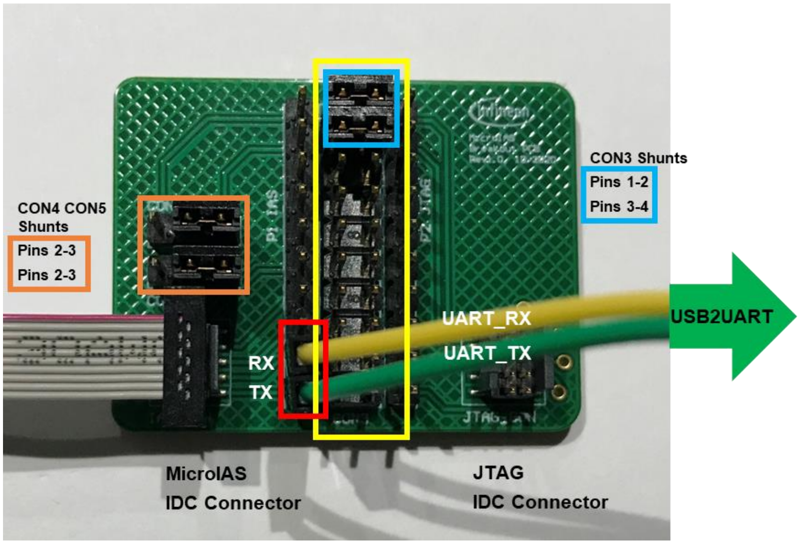

Figure 6c: Breakout Board Setup for UART Communications

Figure 6d details the full setup for GUI or terminal use via UART communications. Infineon's USB-to-UART Reference Design kit serves as power and communications platform between MicroIAS and GUI. Refer to the *MicroIAS (Integrated Alarm System) – GUI for Sensor Fusion Based Alarm System* app note for more details on SW installation and setup.

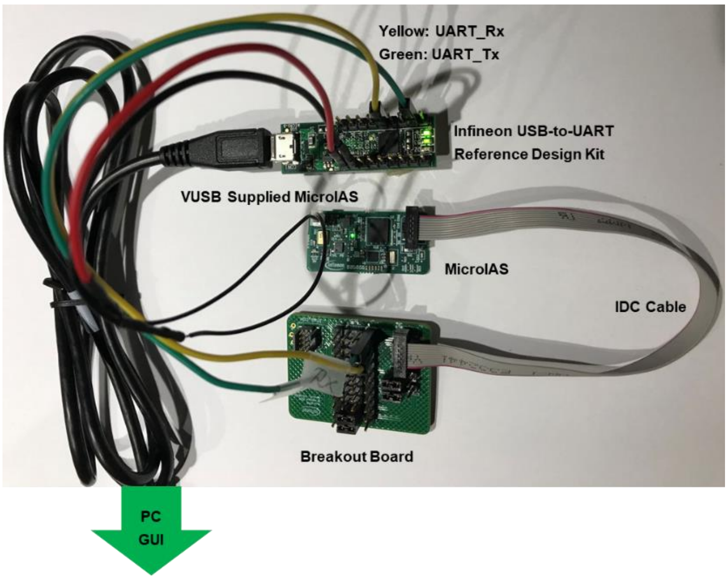

Figure 6d: Full Setup Configuration for UART Communications

2.8 Power Consumption

The MicroIAS is a reference platform built for small spaces and it can support the need for optimized battery powered operation, if a few HW and SW changes are to be taken into account. These changes would require the proper selection of a highly efficient power management chip (replace current 5V-to-1.8V LDO with a >85% efficiency DC/DC converter for higher efficiency) and the code optimization for event detection and classification that keeps the system optimized for lower-power. Contact Infineon Technologies for additional details regarding a battery operated version of the MicroIAS.

3 Detailed Description & Setup

Infineon's IAS is a stand alone system that can be mounted to the ceiling, to a wall or be posititioned within the room (e.g. table, etc). Its range of coverage reaches from 0m to 8m (0 feet to 26 feet) and is meant to be used in regular living spaces/rooms up to a size of $50m^2$ (< 540sqft).

Note: Please refer to *Micro IAS Mounting Recommendation* document for the recommended settings.

Ensure that windows are closed for the alarm system to operate at its peak performance.

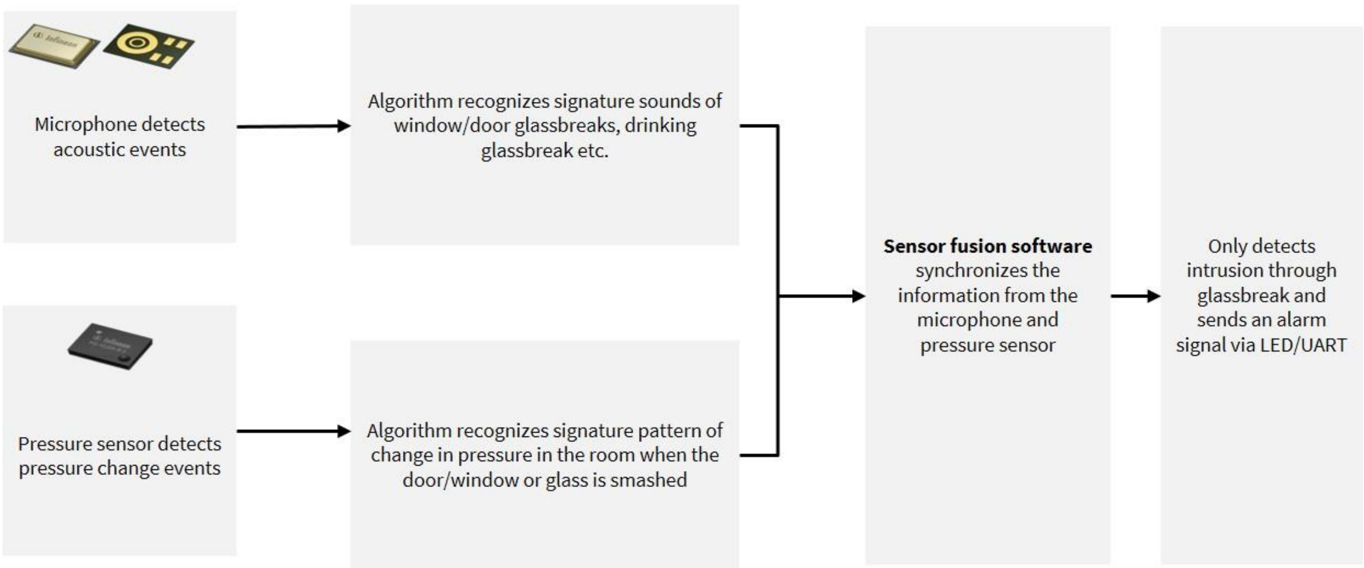

Figure 6 Buiding Blocks of the IAS Software

The MicroIAS engineering solution is a sensor fusion-based glassbreak detector and incorporates an algorithm for pressure event detection.

The IAS functionality is qualified on both QuickLogic (MicroIAS) and XMC reference platforms using Infineon sensors (digital barometric Pressure sensor DPS310 and MEMS digital microphone IM69D130). The focus has been to qualify the IAS software algorithms and not the reference platform hardware. All known limitations for the IAS Software have been documented in the *Release Notes*. A detailed test report for each platform can be provided on request.

3.1 Detected event types

The IAS enables the detection of two types of events:

a) **Glassbreak** – When a window or door glass is broken, the algorithm extracts the intensities in the frequency spectrum, specifically at low and high frequencies and also captures the signature pattern of the change in pressure. The sensor fusion software synchronizes the output and indicates a glassbreak alarm.

b) **Intruder** – When all windows and doors are closed, there is stable pressure inside the room. Opening or closing a window or door triggers a pressure change within the room. Each pressure event has a signature pattern. Therefore, opening or closing a door or window triggers an intruder event alarm.

The IAS software supports the working of multiple modes:

- **Glassbreak:** detects an actual glassbreak event e.g. shattering of a window and the pressure pattern of change of pressure in the room. It is indicated via the on-board Alarm LED (LED2, red) through two blinks and via UART through *"GLASSBREAK EVENT"*.

  Note: With the current algorithm, a glassbreak simulation and a simultaneous pressure change event is also categorized under **Glassbreak**. Hence, it is indicated via the on-board Alarm LED (LED2, red) through two blinks and via UART through *"GLASSBREAK EVENT"*.

- **Intruder**: detects the pressure pattern of a pressure change in a room such as opening and closing of a door. It is indicated via the on-board Alarm LED (LED2, red) through a single blink and via UART through *"INTRUDER EVENT"*.
- **Audio event:** indicates the capture of high and low frequency sounds (essentially a glassbreak sound). It is indicated via UART through *"AUDIO EVENT"*.
- **High frequency audio:** indicates a high frequency audio such as loud claps. It is indicated via UART through *"HIGH FREQUENCY AUDIO"*.
- **Low frequency audio**: indicates a low frequency audio such as tapping on the table. It is indicated via UART through *"LOW FREQUENCY AUDIO"*.

3.2 Sensitivity setting for Glassbreak detection

The system supports the following sensitivity settings for detecting Glassbreak events:

- **High sensitivity:** optimized for optimal glassbreak detection; assumes that no home owner/occupant is present
- **Medium sensitivity:** optimized to work reliably if people are active at home
- **Low sensitivity**: optimized to work reliably if people are active at home and won't be confused by disturbers (e.g. loud music)

The system also supports the pressure sensitivity settings associated with the detection of an Intruder event:

- **High sensitivity:** lowers the threshold for the peak detection in an intruder event.
- **Medium sensitivity:** a threshold for peak detection that lies between the high and low sensitivity.
- **Low sensitivity**: increases the threshold for peak detection in an intruder event.

There are 2 ways to configure the sensitivity settings:

- **Via the Source Code:** The user has to first set up the IDE (see Section 4.1). Then the the settings can be directly modified in the source code to accomodate the user's/application's requirements. Therefore, open the file **ifx_sensor_task.c** in the Eclipse Project. In this file search for the function:

  ```
  ifx_alarm_set_configuration (alarm, "sensitivity", "1.0");
  ```

  Replace the "1.0" value (high sensitivity) with "0.0" for low sensitivity or "0.5" for medium sensitivity. This configures the Glassbreak sensitivity.

The pressure sensitivity can also be configured similarly by replacing "sensitivity" with "pressure_sensitivity" and changing "1.0" (high sensitivity) to either "0.0" (low sensitivity) or "0.5" (medium sensitivity).

```
ifx_alarm_set_configuration (alarm, "pressure_sensitivity", "1.0");
```

Compile the project and flash it onto the MicroIAS board to make the new settings effective (for flashing details, see Section 4.3).

- **Via the Debug Mode:** The debug mode allows a user to configure the sensitivity settings during runtime via certain commands. Please refer to the Debug Mode section in the API Documentation for further details. Note that the settings will be reverted back to its original when the MicroIAS is powered off and powered on.

4 Project setup and flashing

4.1 Installation of IDE and necessary tools

This document provides instructions for setting up the Eclipse-based project of the IAS on a Windows platform and would be suitable for those developers who wish to flash the project onto the MicroIAS board and run it as is or build features on top of the existing application. The user needs to follow the procedures mentioned in sections 4.1 and 4.2 only once to setup the project.

1. Ensure that Java SE DK and Java SE RE are installed as Eclipse IDE requires JAVA
   https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
   https://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html
2. Download the Eclipse installer from the following link and install Eclipse IDE for C/C++ Developers
   https://www.eclipse.org/downloads/download.php?file=/oomph/epp/oxygen/R2/eclipse-inst-win64.exe
3. Download **GNU MCU Eclipse Build Tools zip (Win 64)** from https://github.com/gnu-mcu-eclipse/windows-build-tools/releases and extract the zip file.
4. Download **ARM GCC Toolchain (Win 32 zip)** from https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads and extract the zip file.
5. Download the **J-Link Software and Documentation pack for Windows** from https://www.segger.com/downloads/flasher/ and install it.
6. Download and install a terminal emulator program. The recommended software is Tera Term https://ttssh2.osdn.jp/index.html.en.
7. A USB-to-UART module is required to visualize the output via UART. The recommended module can be obtained from Infineon. The module is called CYUSBS232 USB-UART LP Reference Design Kit and its landing page/link can be found below: https://www.cypress.com/documentation/development-kitsboards/cyusbs232-usb-uart-lp-reference-design-kit
   The CYUSBS232 USB-UART LP Referece Design Kit should not require any additional SW and for most laptops/PCs, the board is automatically detected and set up by the Windows system; however, if problems develop (e.g. driver cannot be located, etc) on the user system, follow the reference kit quick start guide to resolve them.
8. The Quick Logic Flashing GUI utilizes Python 3.4.0 for flashing the binary, so download and install Python v3.4.0 (Windows x86-64 MSI installer) from https://www.python.org/downloads/release/python-340/and Pip from https://pip.pypa.io/en/stable/installing/. Add the path to the folder where Python is installed and the Scirpts folder inside Python to the System Environment Variables:
   a. Goto ***Control Panel>>System and Security>>System***
   b. Goto ***Advanced System Settings>>Environment Variables***
   c. Double click on **Path** and add the paths by clicking on **New** as shown in Figure 7.

   Then install PySerial Module from https://pypi.org/project/pyserial/.

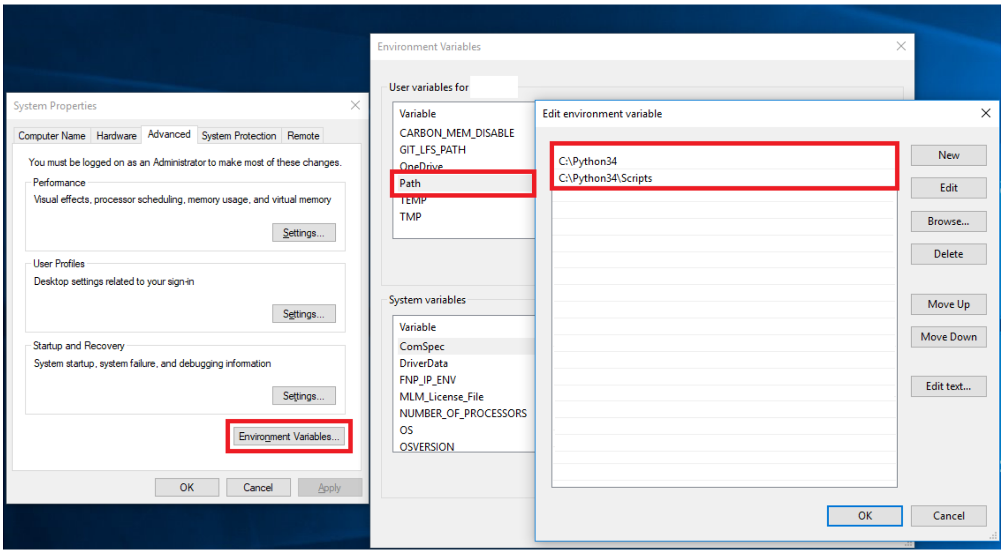

Figure 7 Setting Python path in Environment Variables

4.2 Importing the project into IDE and setting up toolchains

1. Open Eclipse, goto ***Help >> Eclipse >> Market place*** and search for ***GNU MCU Eclipse Plugin.***
2. Install v4.5.1 (Figure 8).

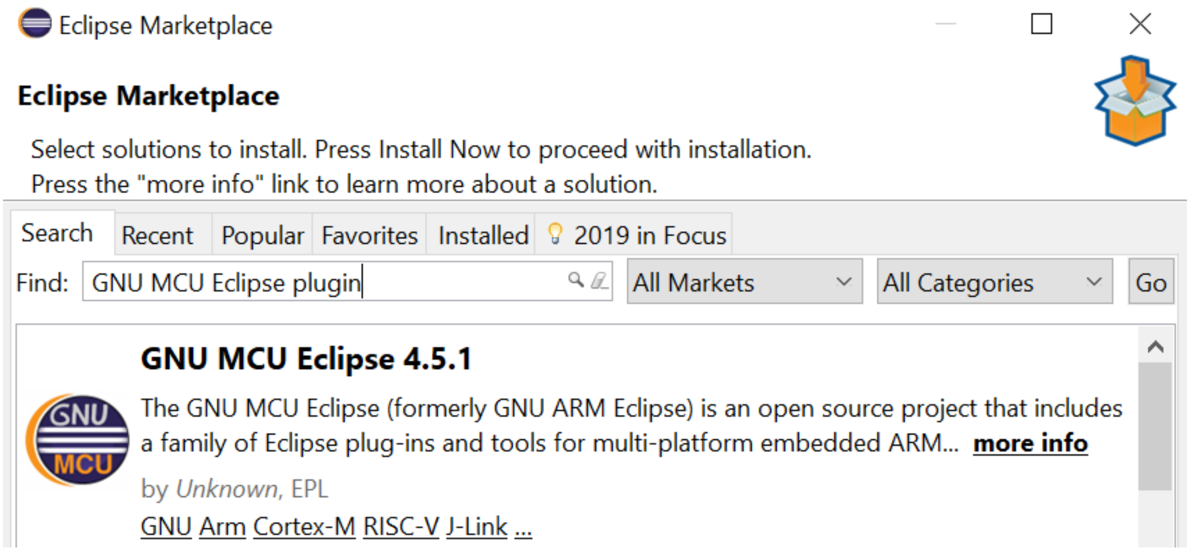

Figure 8 Installing GNU MCU Eclipse Plugin

3. Extract the MicroIAS firmware project (e.g. **Ifx_Alarm_QL_1.0.0_20190524_02ae295c.zip**) from the zip file.
4. In Eclipse, click on ***File >> Import >> Existing projects into Workspace*** and select the directory QL_M4 in the extracted folder.
5. Once the project is imported, right click on the project name from the workspace and select **Properties**.

6. Goto the **MCU** section and and in the **ARM Toolchains Path**, browse to the bin folder of extracted ARM GCC Toolchain (Figure 9) and click on Apply.

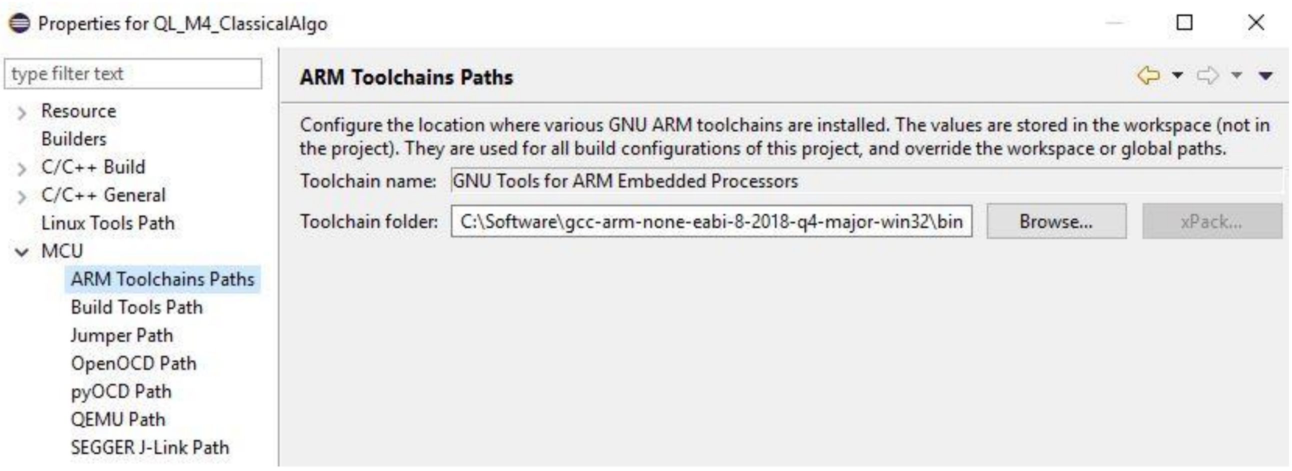

Figure 9 **ARM GCC Toolchains Path**

7. In the **Build Tools Path**, browse to the bin folder of the extracted GNU MCU Build tools (Figure 10) and Apply changes.

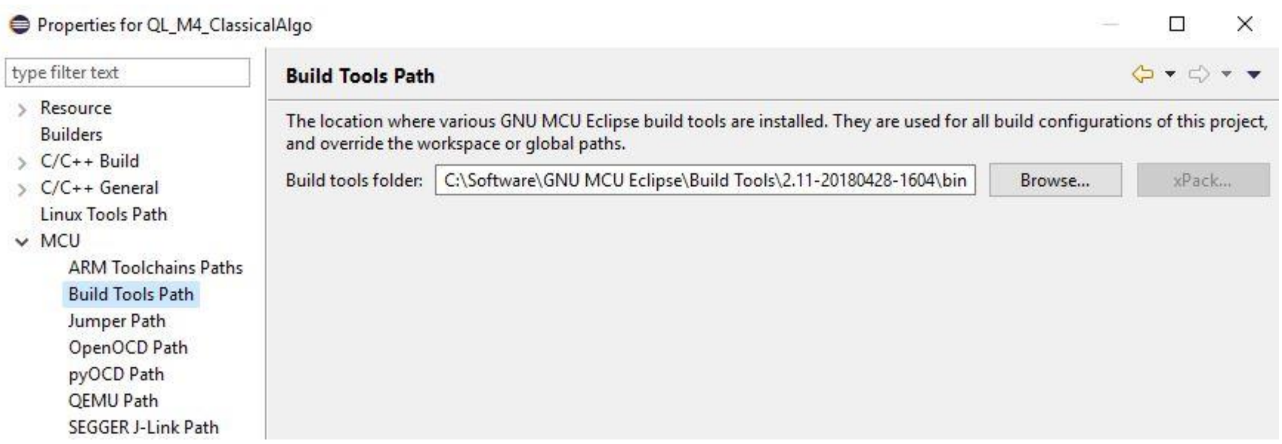

Figure 10 **Build Tools Path**

8. Next, in the **SEGGER J-Link path**, browse to the folder where **JLinkGDBServerCL.exe** is located (Figure 11) and click on Apply.

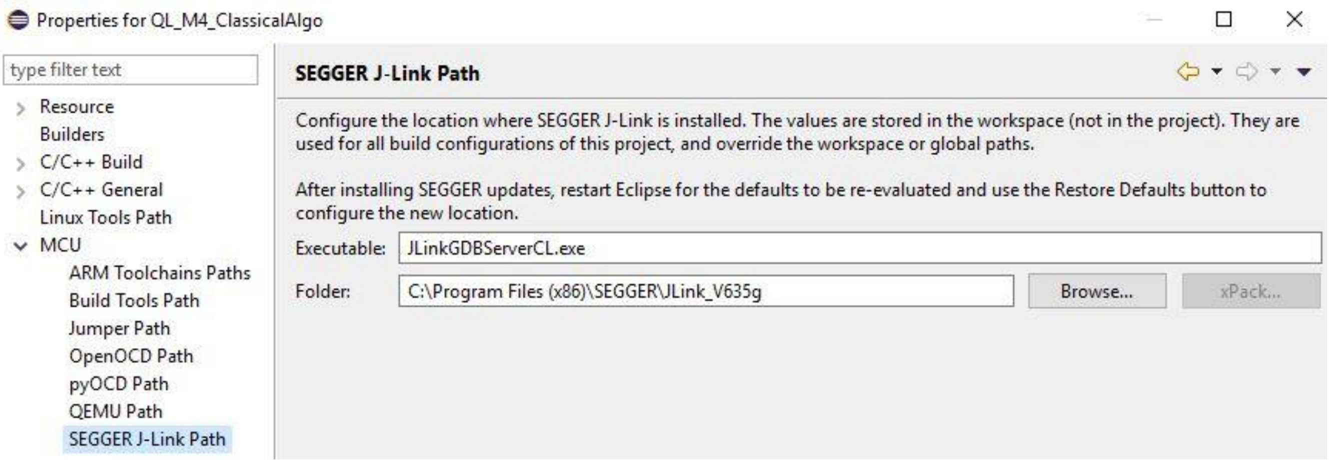

Figure 11 SEGGER J-Link Path

9. Next Open ***C/C++ Build*** **>>** ***Environment*** and Edit the **PATH.** Add Paths for the bin folder of ARM Toolchains, GNU MCU Build tools (Figure 12) and click on Apply.

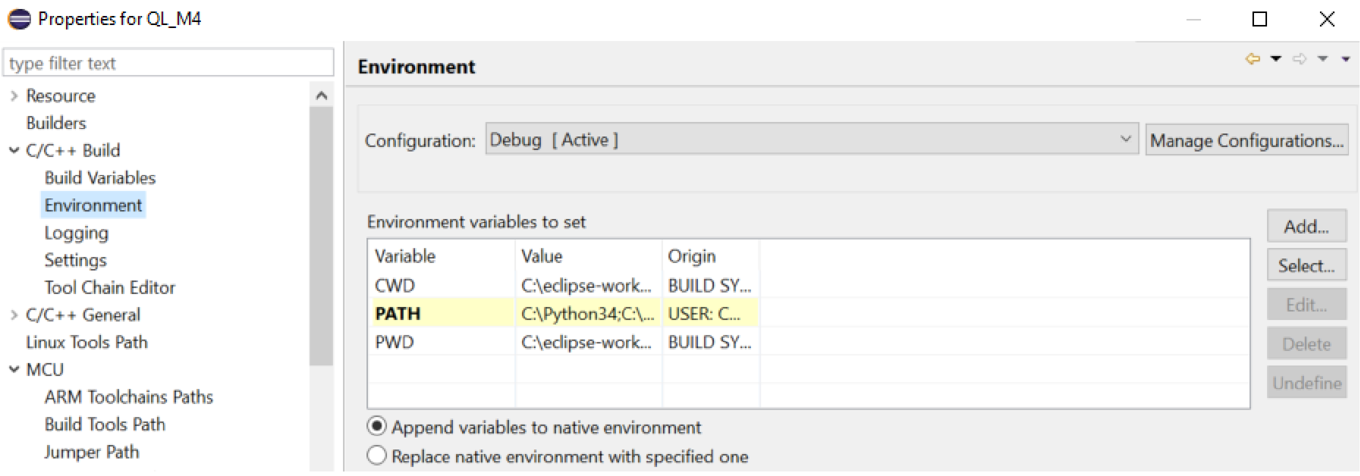

Figure 12 PATH Variables

10. Click on ***C/C++ Build*** **>>** ***Settings*** **>>** ***Toolchains*** and make sure the **Toolchain Path** and **Build Tool Path** from Section 4.2 are included in the path (Figure 13).

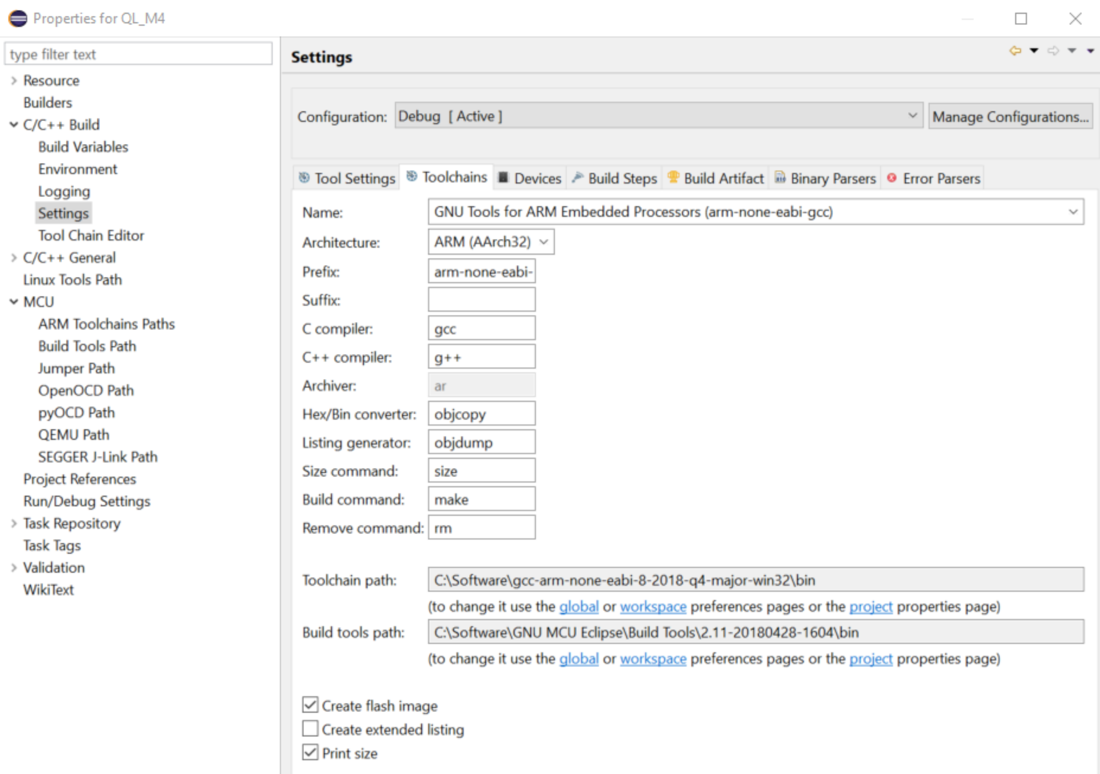

Figure 13 Toolchains and Build Tools Path

4.3 Flashing the project onto the MicroIAS

The binary file of the project can be flashed onto the board using J-Link as well as Quick Logic's Flashing GUI. With the former, booting from flash is not supported and would have to be carried out each time the board is turned off and turned back on, but with the latter, booting from flash is supported.

4.31 Flashing the binary using J-Link

1. Connect the J-Link adapter to the form factor board (**Error! Reference source not found.**). Ensure t hat IO19 and IO20 pins on the J-Link Connection are connected to VCC (see section 2.4 for instructions on hardware setup via the breakout board).
2. Plug in the power supply for the board using the micro USB cable.
3. Click on the reset pin of the MicroIAS board as show in Figure 17.
4. Next, build the project: right click on the project name and select ***Build Project***.
5. Click on the ***Debug*** icon and select ***Debug Configurations***. Double click on the ***GDB SEGGER J-Link Debugging*** and then ensure the ***Debugger*** tabs has the following settings (Figure 14). Apply the changes.
6. Click on the Debug icon to start the flashing.
7. Next, make the UART connections with the MicroIAS.
8. Open Tera Term and configure the settings as depicted in Figure 15.

9. Once Tera Term has the settings configured, click on the Resume icon on Eclipse (alternatively F8 can also be pressed).
10. The events get printed via UART on the Tera Term screen as shown in Figure 16. For an Intruder event, the LED blinks once and for a Glassbreak event, it blinks twice.

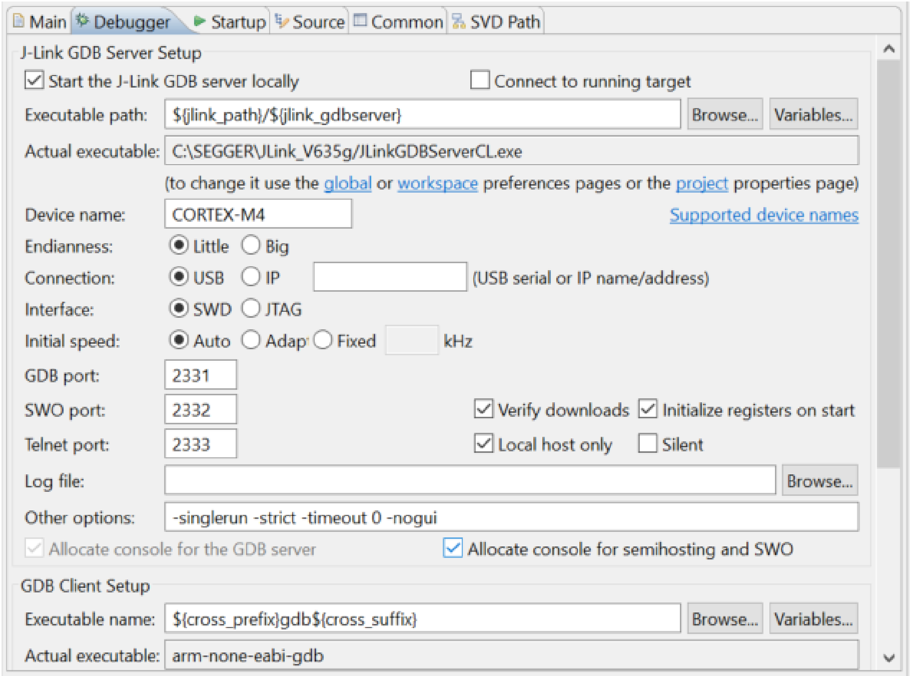

Figure 14 J-Link Debugger Configurations

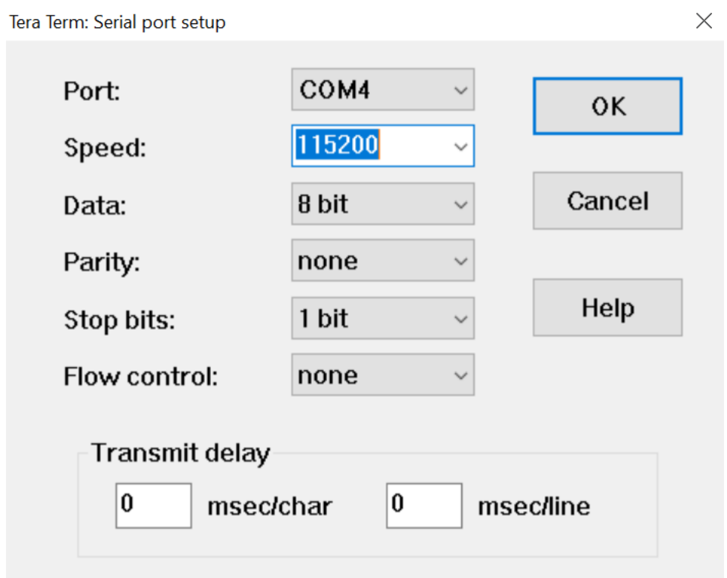

Figure 15 Tera Term Serial Port Setup

Figure 16 Output via UART

Note: Please refer to the API Documentation for further information on the Debug Mode

4.32 Flashing the binary using Quick Logic's Flashing GUI

1. Connect the J-Link adapter to the MicroIAS via the included breakout board (**Error! Reference source not found.**a and 6b).
2. Ensure that IO19 and IO20 pins on the J-Link connection are connected to VCC (see section 2.4 for instructions on hardware setup via the breakout board).
3. Plug in the power supply for the board using the micro USB cable.
4. Click on the reset pin of the MicroIAS board as show in Figure 17.

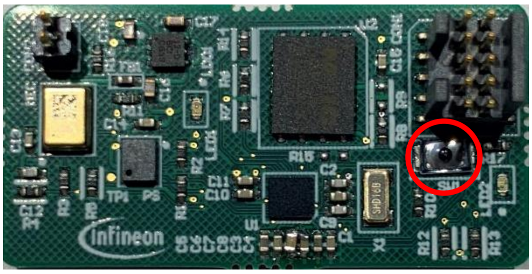

Figure 17 Reset pin of the MicroIAS

5. Open ***QLFlashImage.exe*** and select the PIT file (***PIT.xml,*** describes the files that would be utilized for flashing), Bootloader binary (***EOSS3B_BL.bin***) file and the project binary file (generated and located in Debug folder in the Eclipse Project) as shown in Figure 18. Next, click on Select All and click on Start to start the Flashing.
6. If the flashing is successful, a success message is displayed on the GUI as shown in 0.
7. Disconnect the power, J-Link connection from the MicroIAS board and connect the IO19 and IO20 pins from VCC to GND.
8. Apply power to the board and click on the reset button. After a few seconds, the Alarm LED on the board will blink and TeraTerm will display an output as shown in Figure 16. The MicroIAS is ready for testing.

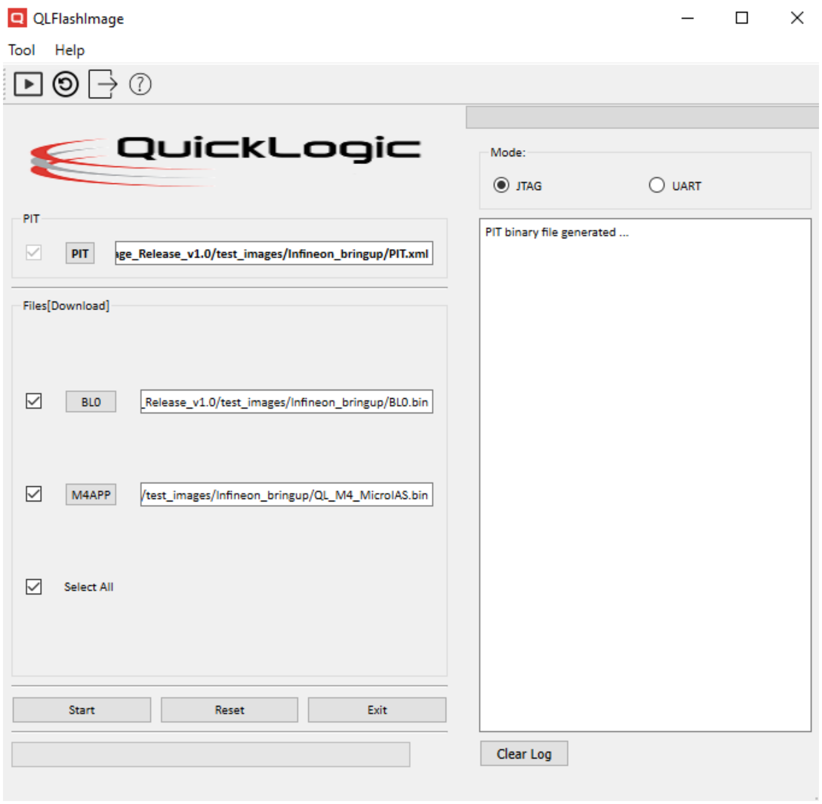

Figure 18 Quick Logic Flashing GUI

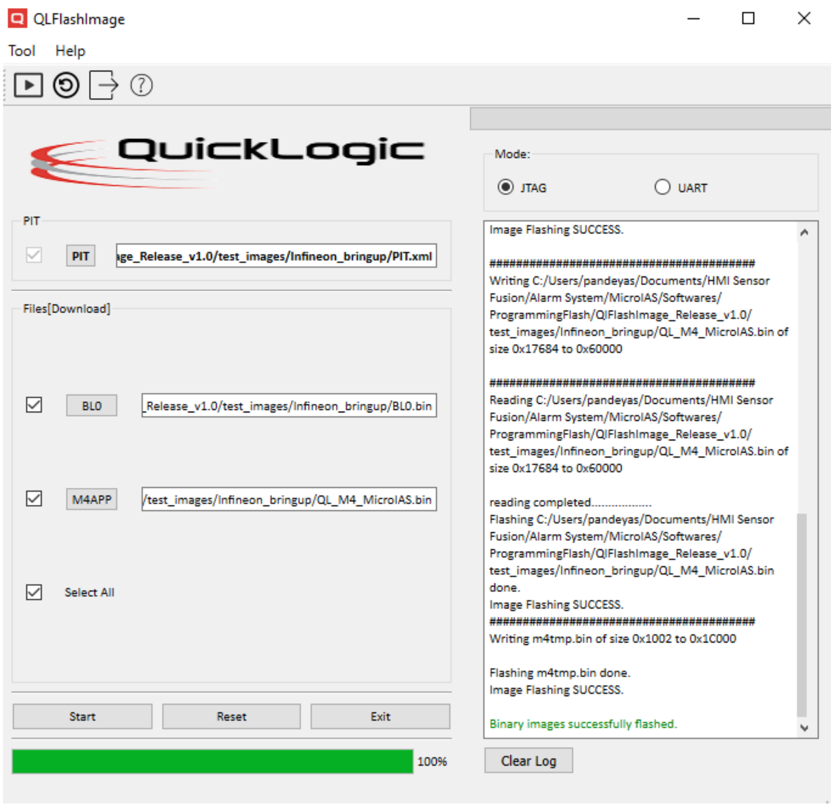

Figure 19 Flashing successful

5 Revision history

Document version	Date of release	Description of changes
0.1	04/12/2019	Intitial Version
1.0	05/27/2019	Engineering Samples Release Version
1.1	10/10/2019	Production Release Version
1.2	01/14/2020	Figures and cross references updated, corrected typos, Sensitivity section updated
1.3	03/03/2020	Corrected connector pin labeling errors in Figure 5
1.4	09/15/2020	Updated pictures and drawings to reflect updates to layout (added CON2 and reassigned pins on CON1)
1.5	12/02/2020	Updated section 2.4 by adding detailed connectivity descriptions for re-programming and UART communications via the included breakout board and IDC cables; page 4-6 have been added for this purpose)

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition yyyy-mm-dd

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference
AppNote Number