Talaria TWO™(INP2045)

Low Power Multi-Protocol Wireless Platform SoC

IEEE 802.11 b/g/n, BLE 5.0

IOT Google Cloud

Contents

1 Figures

2   Introduction

The application discussed in this document provides a brief on using Talaria TWO board and the SDK with Google Cloud (GCP) IoT Core.

More information on the GCP IoT developer guide can be found at:

https://cloud.google.com/iot/docs/how-tos/getting-started

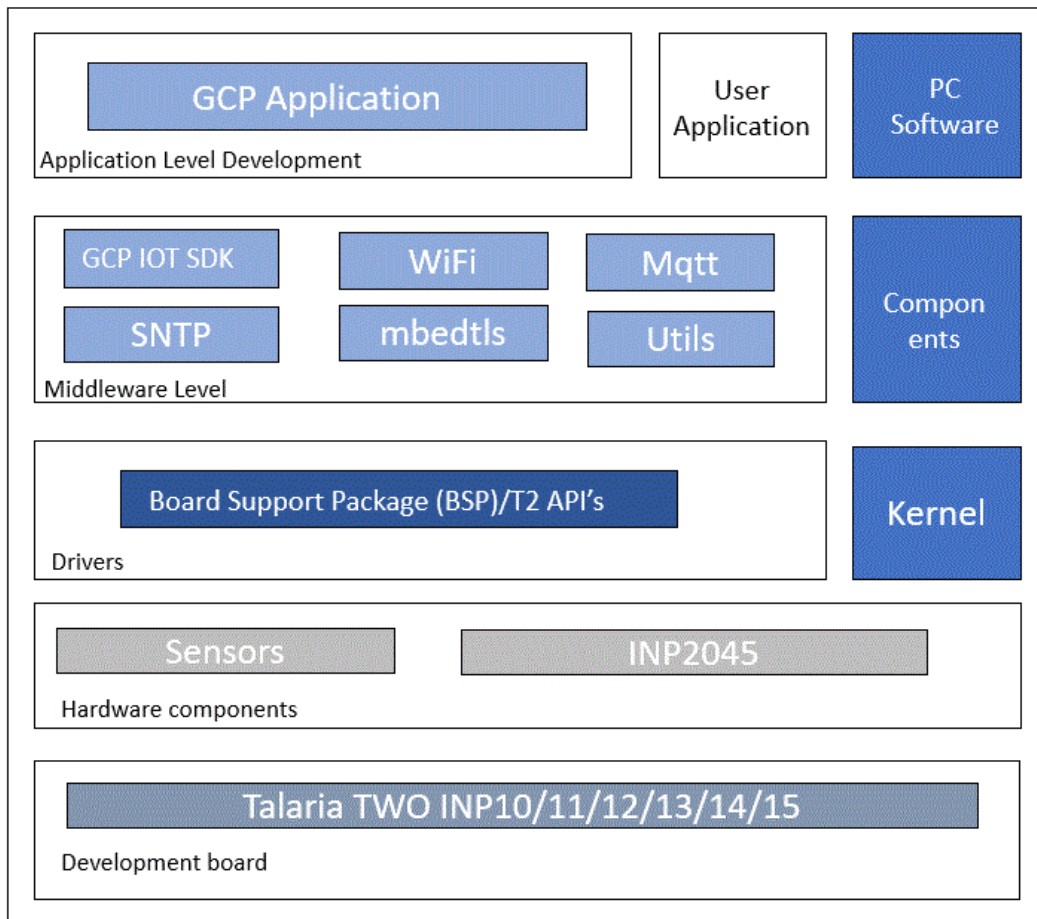## 3   Application Architecture overview

Figure 1: Application architecture

4   GCP IoT Core SDK Embedded C on Talaria TWO Platform

Version v1.0.3 of iot-device-sdk-embedded-C is ported on Talaria TWO platform. The port along with the sample application, is available as a public Github repository here:

```
https://github.com/Sapthas007/talaria_two_gcp
```

More information on iot-device-sdk-embedded-C release tag v1.03 is available here:

```
https://github.com/GoogleCloudPlatform/iot-device-sdk-embedded-c
```

API Documentation and other details specific to GCP IoT Device SDK Embedded C release tag V1.0.3 is available here:

```
https://github.com/GoogleCloudPlatform/iot-device-sdk-embedded-
c/blob/master/doc/user_guide.md
```

Follow the instructions in the subsequent sections to clone, patch and compile the repository with Talaria TWO SDK.

4.1 Cloning the talaria_**two_gcp** repository

Create a new folder in any place and clone the "`talaria_two_gcp`" repo using below command

```
git clone --recursive https://github.com/Sapthas007/talaria_two_gcp.git
```

Note: This repository uses Git Submodules for its dependencies. The option --recursive is required to clone the Git Submodule repository, further required by talaria_two_gcp repository.

4.2 Compiling the **GCP IoT SDK Embedded C** and the Sample App

1. After the clone is complete as described in the previous step, run the python script using the following command python3 gcp_auto.py.
2. It will apply the patch and generate the elf files in the output directory.

4.3 Folder structure of the talaria_two_gcp repository

The repo `talaria_two_gcp` has the below directories/files:

- directory iot-device-sdk-embedded-c-master- Contains the Google cloud IoT Device SDK Embedded-C Release.
- directory patches - Contains patch file iot_gcp.patch for google IoT Device for Talaria TWO compatibility.
- directory files- Its 'Platform Adaptation Layer' contains Talaria TWO Platform-specific porting needed to adapt to Google IoT SDK.

5   Overview of Sample Applications

Sample Applications ported to the Talaria TWO Platform can be found in the path /examples/iot_core_mqtt_client. A brief overview of these apps is provided in this section.

Follow Application Note provided with the Talaria TWO SDK at the path <sdk_path>/apps/iot_gcp for further details on programming certs, keys and executable binaries on Talaria TWO based EVB-A boards and running the Sample Applications / verifying the expected outputs using the Debug Console and GCP Web Console.

5.1   iot_core_mqtt_client

This example takes the parameters from the T2 boot arguments and establishes a connection to the Google CLoud IoT Platform. Then it subscribes and publishes to the topics provided as bootArgs subscribe_topic and publish_topic.

If all the certs/keys are correct, in the T2 Console you should see alternate QoS0 and QoS1 messages being published to 'publish_topic' by the application in a loop.

GCP IoT Console->Test page can be used to subscribe to '/devices/atest-dev/commands/#' (or T2's 'publish_topic' provided as the bootArg to App) to observe the messages published by the App.

GCP IoT Console->Test page can be used to publish the message to '/devices/atest-dev/state'(or T2's 'subscribe_topic' provided as the bootArg to App), and T2 App will receive the messages and they will be seen on T2 Console.

JSON formatted text as shown below should be used for publishing to T2.

```
{
 "Hello From Innophase T2"
 }
```

The application takes in the ssid, passphrase, iotc_project_id, iotc_publish_topic, iotc_publish_message, and iotc_device_path as must provide bootArgs.

Certs and keys are stored in rootFS and read from app specific paths defined in the sample code.

6 Google Cloud Set-up

1. Create a Google cloud IoT account An GCP account is needed to run the sample applications. GCP accounts include three months of Free Tier Access.
   More information on: https://console.cloud.google.com/billing

2. Create and register device/thing Device/thing must be registered onto the GCP IoT registry. Use the   following link to AWS IoT user guide to download the necessary certificates and private key:
   https://services.google.com/fh/files/misc/iot_partner_quickstart1.0.pdf

3. Save Certificate and Private Key onto the device.

   There are three certificates that will be downloaded from GCP for the created Thing. Out of which Public Key should add to google cloud platform device authentication.
   The private key needs to be saved over Talaria TWO modules file system to create a JSON token to connect the sever. To save the private key, store it into a folder named as 'data' and use the Download tool filesystem write.

7 Programming VM-based applications

7.1 Programming Talaria TWO board with certificates

The default path for GCP should be: /data/

7.2 Show File System Contents

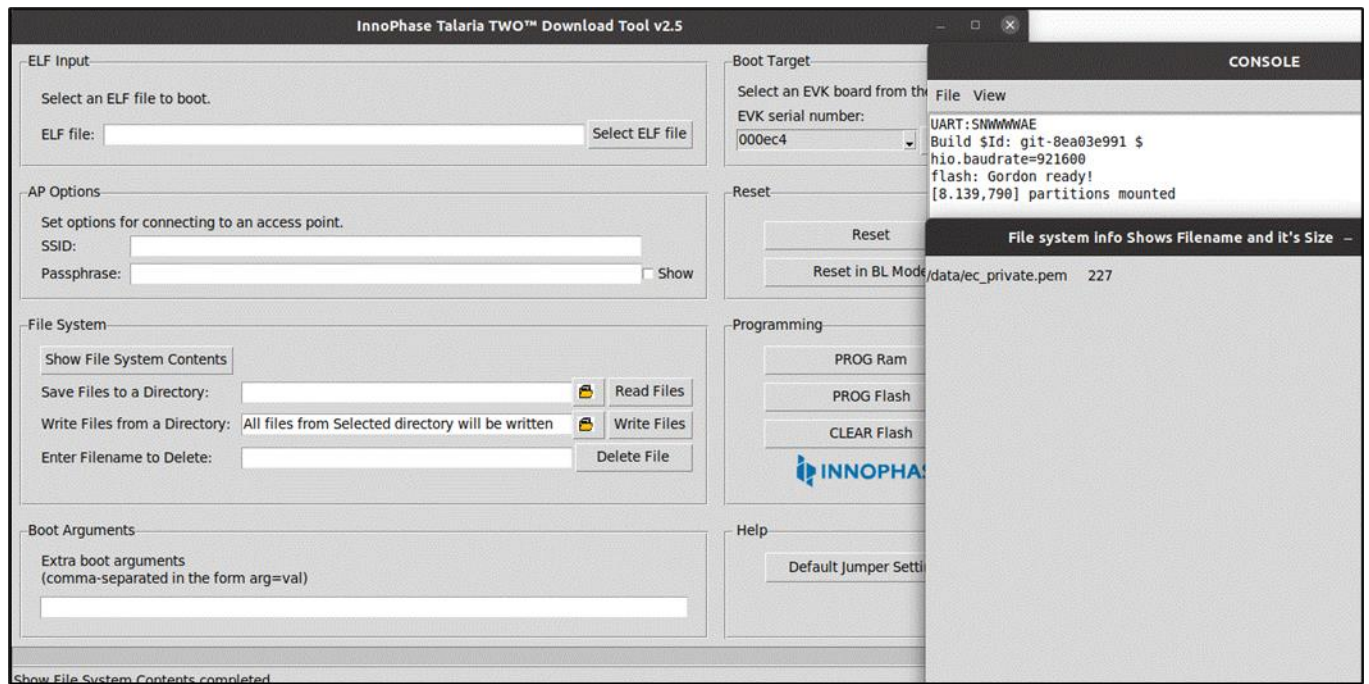Click on Show File System Contents to see the current available files in the file system.

Figure 2: Show File System contents

7.3   Write Files

To write files into Talaria TWO, user must create a folder with the name root and place all certificates either directly into the data or they can create multiple subfolders (for example: /data/iot_gcp) and place the certificates inside the sub-directory and update the path as per the file system in the .c file.

The default path is /data/ app. If user writes into data/iot_gcp/cert_names then the path should be updated in the .c file accordingly. Any number of files/folders inside root will be written.
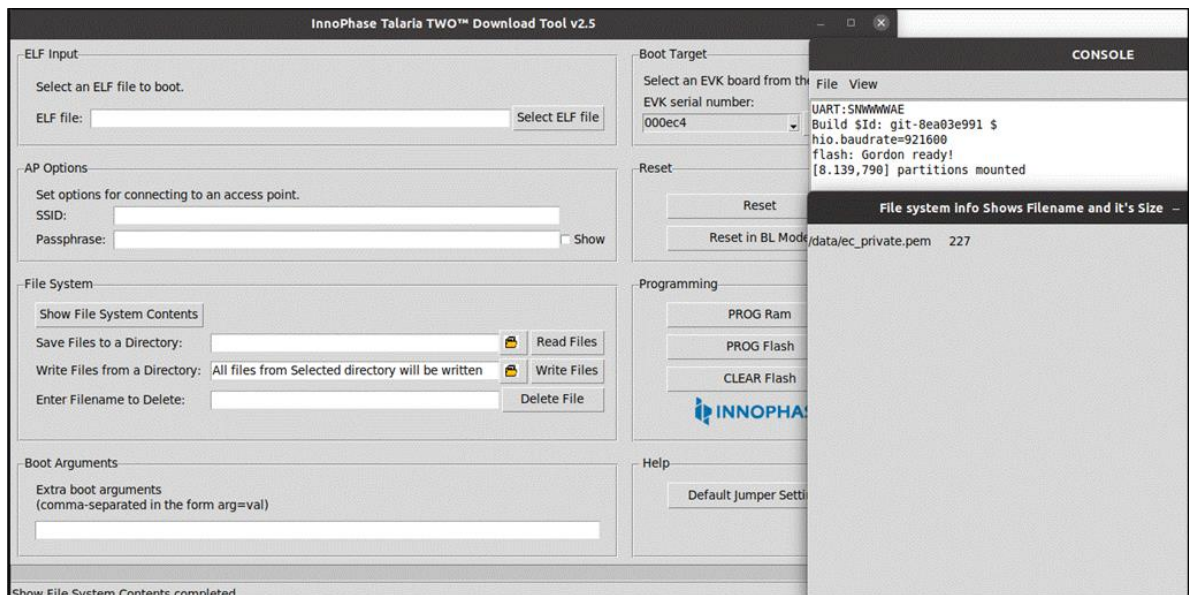

Figure 3: Write Files

7.4   Programming Talaria TWO board with ELF

Program the ELFs (sdk_x.y\apps\iot_gcp\bin) onto Talaria TWO using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO SDK.
2. In the GUI window:
    a. Boot Target: Select the appropriate EVK from the drop-down
    b. ELF Input: Load the appropriate ELF by clicking on Select ELF File.
    c. AP Options: Provide the SSID and Passphrase under AP Options to connect to an Access Point.
    d. Boot arguments: Pass the following boot arguments:

    ```
    iotc_project_id = xxxxxxxx, publish_topic=xxxxxxx,

    iotc_publish_message=xxxxx, iotc_device_path =xxxxx,
    ```

    Note: Replace the xxxxxx with the appropriate details.

    e. Ensure correct boot parameters are supplied to your Wi-Fi network and the information from the device/registary created previously on GCP.
    f. Programming: Prog RAM or Prog Flash as per requirement.
    For more details on using the Download tool, refer to the document: UG_Download_Tool.pdf (path: sdk_x.y/pc_tools/Download_Tool/doc).

7.5   MQTT Publish and Subscribe

This section describes about the Talaria TWO MQTT publish and subscribe client.

Note: All GCP IoT Console screenshots might not look exactly as shown in the figures but might be a variation of the same. This is in-line with the ever-evolving console and its layouts.

6.5.1 MQTT Subscribe

From iot_core select registrar and select the device.

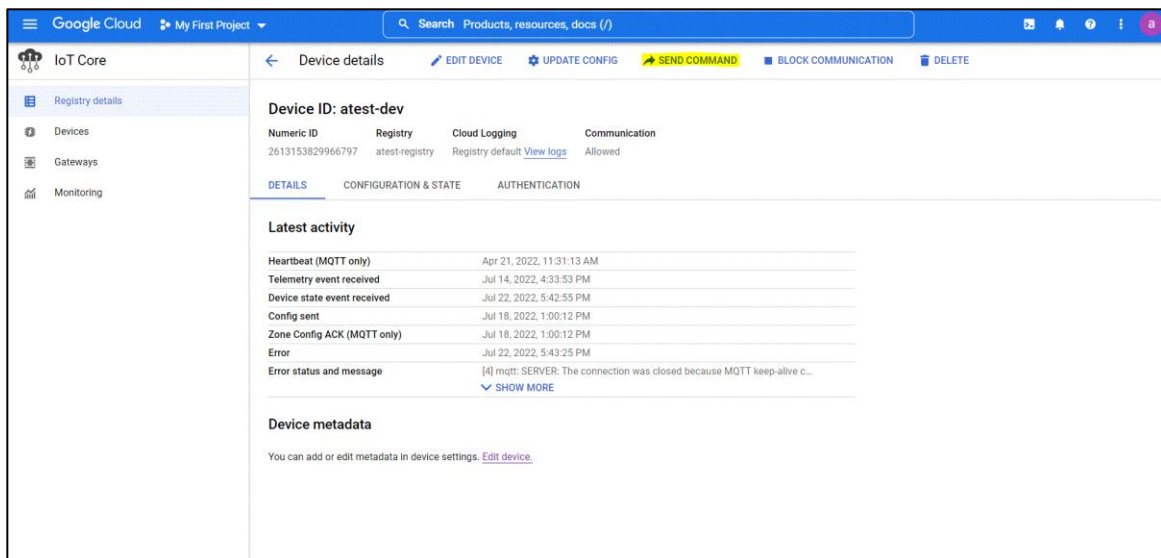In the IoT Core, go to **Registry** ->Devices and select your device.

Figure 4: MQTT subscribe

Then select the Send Command option and add the text in the command data box and send it.

We can turn on and off the LED in the T2 EVB board using the subscribe method. If we send {"LED": ON} then the LED present in the EVB will start to glow. The J3 jumper should be connected to the EVB. To turn off the LED, send the {"LED": OFF} command.
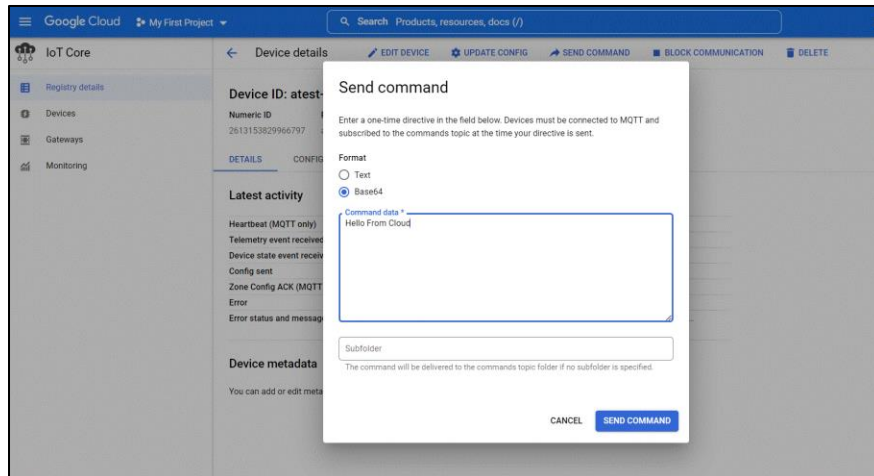
Figure 5: MQTT send command

6.5.2 MQTT Publish

In publish message also select the device from the IoT Core field. it will show the latest activities.
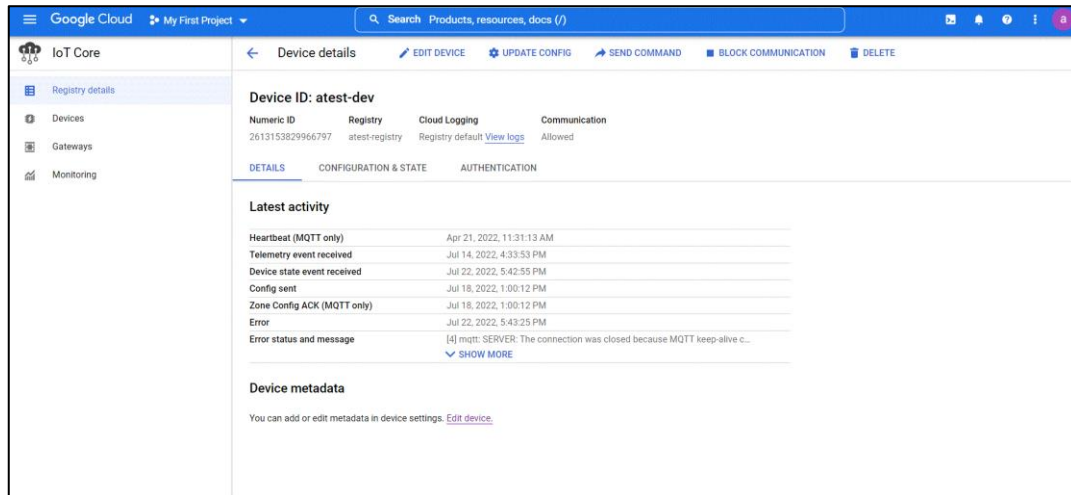
To view the details, click on CONFIGURATION & STATE.

Figure 6: MQTT Publish

The configuration &state will show the publish messages with history.
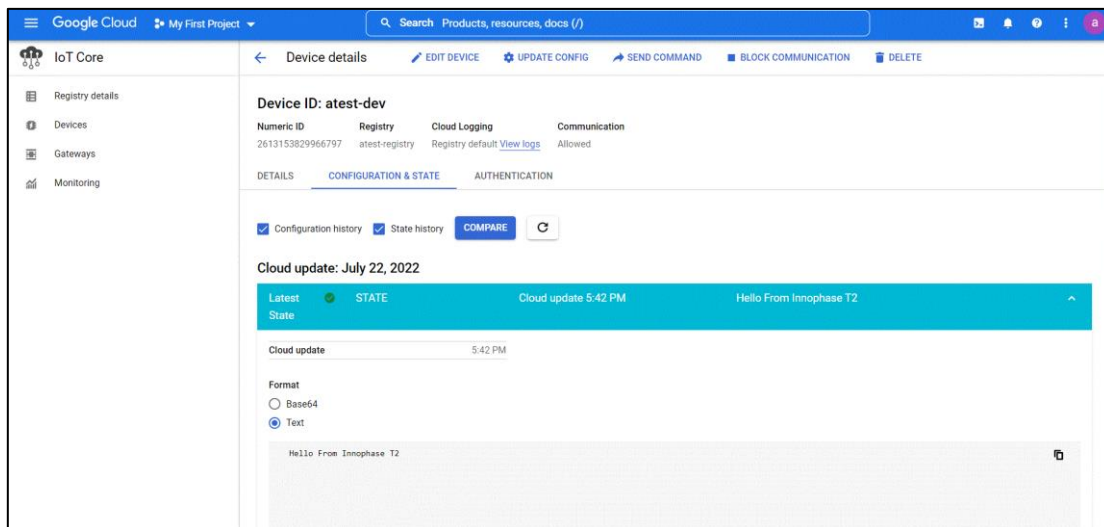

Figure 7: Published message

6.2 Expected Output

`iot_core_mqtt_client.elf` is created when compiling the code which provides the following console output   when programmed to Talaria TWO.

```
UART:SNWWWWAE
Build $Id: git-8ea03e991 $
hio.baudrate=921600
flash: Gordon ready!
Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7
ROM yoda-h0-rom-16-0-gd5a8e586
FLASH:PNWWWWWAEBuild $Id: git-8ea03e991 $
ssid=InnoPhase_AE passphrase=Inno@1234 iotc_project_id=root-range-346206
iotc_publish_topic=/devices/atest-dev/state iotc_publish_message=Hello From Innophase
T2 iotc_device_path=projects/root-range-346206/locations/us-central1/registries/atest-
registry/devices/atest-dev
addr e0:69:3a:00:0e:c4
network profile created for ssid: InnoPhase_AE
[0.991,844] CONNECT:98:da:c4:73:b7:76 Channel:11 rssi:-40 dBm
[1.171,948] MYIP 192.168.0.116
[1.172,228] IPv6 [fe80::e269:3aff:fe00:ec4]-link
connected with AP
SNTP: using server: pool.ntp.org
sntp_process: Wed Jul 27 07:51:42 2022
 utc:1658928102:1658928098804592:3195414
 connected to mqtt.2030.ltsapis.goog:8883
 publishing msg "Hello From Innophase T2" to topic: "/devices/atest-dev/state"
 Subscription Topic: /devices/atest-dev/commands/#
 Message Payload:
 publishing msg "Hello From Innophase T2" to topic: "/devices/atest-dev/state"
 publishing msg "Hello From Innophase T2" to topic: "/devices/atest-dev/state"
 Subscription Topic: /devices/atest-dev/commands
```

8 Support

1. Sales Support: Contact an InnoPhase sales representative via email – sales@innophaseinc.com
2. Technical Support:
   a. Visit: https://innophaseinc.com/contact/
   b. Also Visit: https://innophaseinc.com/talaria-two-modules
   c. Contact: support@innophaseinc.com

InnoPhase is working diligently to provide outstanding support to all customers.

8 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, InnoPhase Incorporated does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and assumes no liability associated with the use of such information. InnoPhase Incorporated takes no responsibility for the content in this document if provided by an information source outside of InnoPhase Incorporated.

InnoPhase Incorporated disclaims liability for any indirect, incidental, punitive, special or consequential damages associated with the use of this document, applications and any products associated with information in this document, whether or not such damages are based on tort (including negligence), warranty, including warranty of merchantability, warranty of fitness for a particular purpose, breach of contract or any other legal theory. Further, InnoPhase Incorporated accepts no liability and makes no warranty, express or implied, for any assistance given with respect to any applications described herein or customer product design, or the application or use by any customer's third-party customer(s).

Notwithstanding any damages that a customer might incur for any reason whatsoever, InnoPhase Incorporated' aggregate and cumulative liability for the products described herein shall be limited in accordance with the Terms and Conditions of identified in the commercial sale documentation for such InnoPhase Incorporated products.

Right to make changes — InnoPhase Incorporated reserves the right to make changes to information published in this document, including, without limitation, changes to any specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — InnoPhase Incorporated products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an InnoPhase Incorporated product can reasonably be expected to result in personal injury, death or severe property or environmental damage. InnoPhase Incorporated and its suppliers accept no liability for inclusion and/or use of InnoPhase Incorporated products in such equipment or applications and such inclusion and/or use is at the customer's own risk.

All trademarks, trade names and registered trademarks mentioned in this document are property of their respective owners and are hereby acknowledged.