TileLib Quick Start Guide

Dialog DA14585

1. Introduction

This guide explains how to enable Tile functionality in your BLE product using Dialog DA14585.

A software or firmware engineer should be able to enable TileLib by following this guide. A recommended first step is to launch Tile Demo application on a Dialog Development Kit and this is explained in this guide.

Integrating TileLib requires having access to TileLib binary and API header files. This is explained in this guide too.

Here is to contact Tile firmware team: firmware_support@tile.com
Here is to contact Tile business development team: bizdev@thetileapp.com

2. Tile Demo App

Tile Demo Application uses a Keil uVision project.

It runs on:
- DA14585 Development Kit Pro: Recommended.
- DA14585 Development Kit Basic: OK but no button.

It has been tested on Dialog SDK version 6.0.10.511

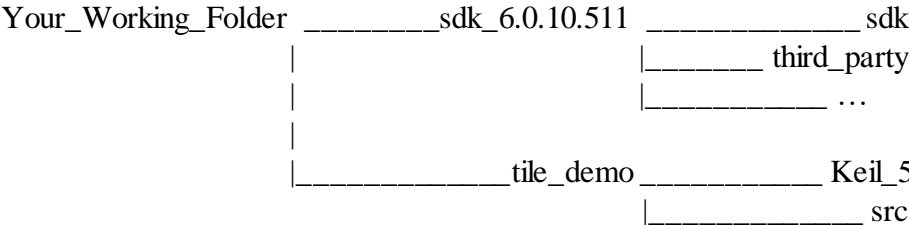The Tile Demo version is given in the file tile_demo\src\tile_shim\Tile_version.h

Building the Tile Demo application source files

Download Tile Demo Application source files:

These can be downloaded:
- From Dialog here:https://support.dialog-semiconductor.com/
- From Tile: email firmware_support@tile.com or bizdev@thetileapp.com

Make sure you have Dialog SDK 6.0.10 available at the same folder as tile_demo as follows:

```
Your_Working_Folder _____sdk_6.0.10.511 _____ sdk
                    |                              |_____ third_party
                    |                              |_____ …
                    |
                    |_____tile_demo _____ Keil_5
                                                  |_____ src
```

You can now open Tile Demo project at: tile_demo/Keil_5/ tile_demo.uvprojx

Get Tile Node credentials

You need valid Tile Node credentials [interim_tile_id, interim_tile_key] in order to access Tile Network. Without valid Tile Node credentials, your prototype will not interoperate with the Tile Network or Tile Mobile Application.
Follow this link to get Tile Node credentials for your prototype:
https://www.thetileapp.com/en-us/platform-business-solutions
Then fill interim_tile_id and interim_tile_key with the numbers provided to you by Tile at the top of tile_demo\src\tile_shimuser_tile_api_if.c

Build the application:

Build.

For information how to set up the Dialog Development kit PRO see the manual in the following link: https://support.dialog-semiconductor.com/resource/um-b-049-da14585-da14586-getting-started-guide-pro-development-kit-html

Development Kit configuration for Tile Demo application

Configure the board for SPI using the jumpers

The jumper configuration is described on the board.

Plug a Piezo or Buzzer for audio.

In order to hear the audio, you will need to plug a buzzer to the board. It goes between P_0_7 and GND.

You can use the following part: PKM13EPYH4002-B0

here is the Digikey part number 490-4697-ND

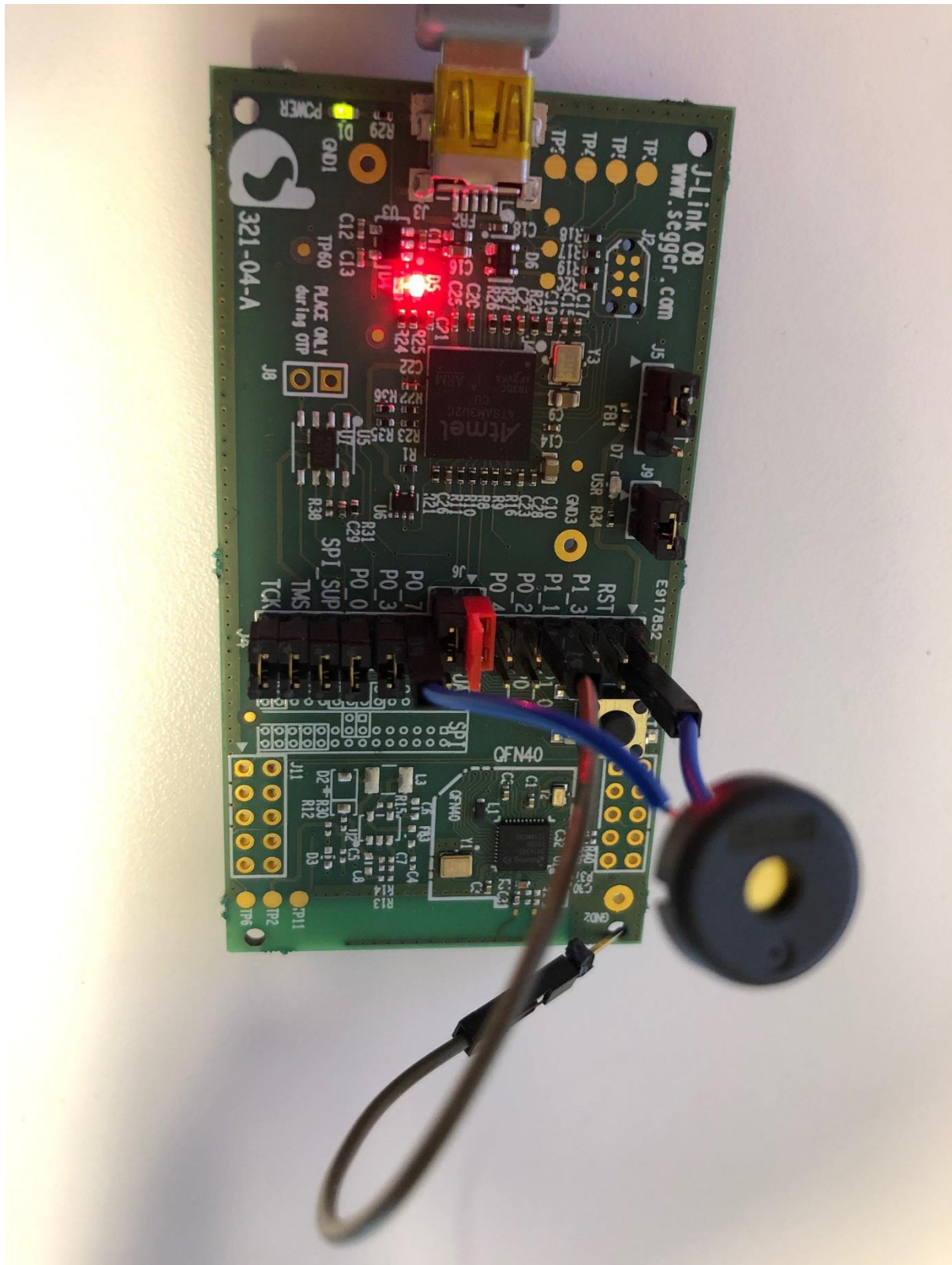### *Make a Button (Basic Kit only).*

On the Pro Kit, the Tile Button is mapped to SW3.

On the Basic Kit, you can plug a wire jumper to P1_1 and make contact with ground to trigger a button press.

Picture of a Configured DA14585 Development Kit Pro:

Picture of a Configured DA14585 Development Kit Basic:

Loading the Tile Demo application

The Tile Demo Application can be loaded using either SmartSnippets (usually for storing into flash) or using Keil (usually for debugging).

Using SmartSnippets Toolbox:
Open the SPI Flash Programmer. Keep all the defaults settings.
Click Connect, then Erase.
Select the File to Download: tile_demo\Keil_5\out_585\tile_demo_585.hex
Click Burn.
Select Yes to make the flash memory bootable.

After that step, if you press Reset on the board, the Tile application should start. If you have a Piezo or Buzzer, you should hear the Tile Wakeup Song.

Also, at this point you should be able to run the full Tile Basic Tests.

3. Tile Basic Tests

Download the Tile App (Apple or Android).

Activate the Tile Node from the Mobile App.

Ring the Tile Node from the Mobile App.

Check whether the Proximity Meter varies with distance / isolation.

Double click the Tile button to ring the phone.

4. Integrating TileLib in your Application

This guide assumes you start from an application running on 585 using a Keil project. Most of the code needed to integrate TileLib is provided in Tile Demo application.

Add Tile source files to your project:

Add to your project all the source files located in the following folders (you may copy the folders first):
- tile_common
- tile_lib
- tile_shim

Add Tile Include files path to your project:

As follows:
.\..\src\tile_common;.\..\src\tile_lib;.\..\src\tile_lib\include

Insert Calls to Tile in your application

You can compare the config and platform folders from the Tile Demo application with the ones from the Dialog ble_app_barebone ble example target app.

Here is a list of the differences for Tile support in the config folder:
- da1458x_config_advanced.h
  - #define CFG_USE_CHACHA20_RAND
    We recommend CHACHA20 for a better security.
  - #define CFG_BLE_METRICS
    This is useful to our distance estimation algorithm.
  - #define CFG_RET_DATA_SIZE (3072)
    Enabling Tile requires a little bit of memory.
- user_callback_config.h
  - #include "user_tile_demo.h"
  - #include "user_tile_api_if.h"
  - .app_on_generate_static_random_addr =
    user_tile_app_generate_static_random_addr,

This is only needed if you want to generate random static mac addresses that you would like to save in flash and reuse later. Tile would still work properly if the address is public.

- user_config.h
    - #define USER_CFG_ADDRESS_MODE   APP_CFG_ADDR_STATIC
      This is only needed if the MAC address used is random static.
    - #define USER_DEVICE_NAME   "DLG-Tile"
      You can change the Device Name here.
- user_modules_config.h
    - #define EXCLUDE_DLG_MSG   (0)
      Do not exclude Dialog messages
- user_periph_setup.h
  Configure here your GPIOs for Tile features.
- user_profiles_config.h
    - #define CFG_PRF_TILE
      Enable Tile Service

Here is a list of the differences for Tile support in the platform folder:
- user_periph_setup.c
    - #include "user_tile_api_if.h"
    - void GPIO_reservations(void)
      Add reservation for Tile features.
    - void set_pad_functions(void)
      Configure GPIOs for Tile features.
    - void periph_init(void)
      Add tile_button_enable();

The other files in the platform folder have been copied from the sdk and patched. You need to remove the old sdk files from your project and add the patched ones. You can see the differences by comparing the files.

Here are the Tile related items that need to be added to your main application (search for "tile" in user_tile_demo.c):
- Set Tile Advertising Data.
- Notify Tile of a connection.
- Notify Tile of a disconnection.