Qualcomm Bluetooth Low Energy SDK
Qualcomm® Bluetooth® Low Energy

Terminology clarification

In this document you will notice a number of references are made to Qualcomm Bluetooth Low Energy SDK. While this is the official name of the SDK moving forward, within the actual SDK you will see it currently referred to as CSR µEnergy. In addition, many of the directories and folders may have the CSR µEnergy SDK naming convention. Please be assured these are the same thing.

Thank you for your understanding while we take the necessary steps to phase out the use of CSR µEnergy across our product lines.
Qualcomm® Bluetooth® Low Energy

The μEnergy SDK is the IDE at the core of the toolchain used to develop for, program and debug the CSR101x family of silicon.

Fully documented API plus code examples

The SDK contains a number of example applications, both to show example peripheral usage, and fully pre-qualified example implementations of Bluetooth Smart profiles.
CSR102x SDK: Core and Packages

Segment Specific Packages

Available from https://www.csrsupport.com/uEnergy/Software
CSR102x SDK: Core and Packages

Segment Specific Packages

The core package is mandatory and contains all the basic elements to permit the programming and debug of applications on CSR102x.

Additional packages can be downloaded which provide additional example profile applications and associated tools.

To access example code for a Heart Rate profile compatible with the Qualcomm development kits would require

1. Core Package
2. Health and Fitness Package
CSR102x SDK Included in the SDK:
Peripheral example applications

In addition to example profiles, the core SDK also includes example code for utilising on board peripherals

• Edge Capture
• Infra-red (remote control) driver
• Keyboard scanner
• Programmable Digital I/O
• Pulse Width Modulation
• Quadrature Decoder
• UART I/O
CSR102x SDK Launch SDK

Start the SDK from the desktop icon or the Start menu

Accepting default options during installation (recommended) will result in the SDK being added to the Start Menu, and a shortcut placed on the desktop as shown below.
CSR102x SDK Launch SDK
The SDK presents the support page when it opens
CSR102x SDK Apps

Opening an Application

All applications can be accessed via the Project / Open Workspace menu option.

Each project contains an application note which details the architecture, functionality and configuration options for each application.

This is a PDF inside the project directory itself, with the project source code.
CSR102x SDK - Application entry points:

**Application Entry Points**

**Application Initialisation - AppInit**
Executed at Power on reset / wakeup from sleep.
Initialisation of PIO states, internal and external peripherals and timers

**System Event Handler - AppProcessSystemEvent**
Executed whenever a system event is raised
Battery low, PIO change, wake pin toggled etc (full details available in the SDK help)

**Peripheral Event Handler - AppProcessEvent**
Allows handling of messages from system peripherals
I2C, UART, Keyscanner etc (full details available in the SDK help)
CSR102x SDK Hands-on: creating a minimal application

Output appears in the terminal window

Create a new “minimal” Hello World application using “New Project”

A terminal emulator compatible with UART must be utilised for this. Settings: 2400 baud, 8N1
CSR102x SDK - Application entry point example:

“Hello World” - AppInit()

Application Initialisation entry point:

Initialises UART pins
Sets Baud Rate
Establishes Serial Debug Interface
Outputs “Hello World” over the serial port

This application does not have requirement to handle any system, peripheral or connection manager events

```c
void AppInit(sleep_state last_sleep_state)
{
    /* UART Configuration structure */
    uart_pio_pins_t uart;

    /* Initialise UART Configuration structure */
    uart.rx = UARTPIO_RX;
    uart.tx = UARTPIO_TX;
    uart.rts = UARTPIO_RTS;
    uart.cts = UARTPIO_CTS;

    /* Initialise Default UART communications */
    DebugInit(1, UART_RATE_2K4, 0, &uart);

    /* Print Welcome Message */
    DebugWriteString("Hello, world\r\n");
}
```
CSR102x SDK Advantages of SDK-3.x applications

Uniform applications
• Connection Manager & NVM Manager provide uniform application interface
• All applications will follow the same design approach

Re-usable code
• Connection Manager, NVM Manager and Common Services shared by all apps
• Easy maintainability of the code

Easy to customize
• Services can be added/removed easily with minimal changes

Supports CSR101x and CSR102x
• Connection Manager and NVM Manager are hardware independent
• Same application compiled for CSR101x and CSR102x using respective SDK
CSR102x SDK Connection Manager
A uniform interface over LS, GAP, GATT and SM APIs

Communication between the applications and CM happens through requests and events

Handler registration
• Application handler
• Client service handler
• Server service handler

Events routing
• Generic events (connection, bonding/encryption) are broadcasted to all the handlers (clients, servers and main application)
• GATT access events sent to respected handler (client or server service handler)
• Some events are sent only to the application such as radio event etc..
CSR102x SDK Connection Manager - Contd.

SDK Settings of GAP and GATT roles:
• Peripheral, Central, Client and Server

Application allocates memory for:
• Connections, bonding, and Client/Server services Information

Connections differentiated by device id (assigned after connection)
• Device id to be used as a connection handle
• Shields the low level connection handle from accidental corruption

Public interface
• cm_api.h contains the public APIs
• cm_types.h contains the structure types

API documentation(doxygen) available as part of the SDK-3.x
SDK Help and Documentation

Help Documents

Clicking on the help tab accesses

SDK Support Documentation
Application Notes
User guides
Package Documentation

Firmware Library
API functions specific to CSR102x family

Common Library
API functions common to single mode SoC product range