

Darwin Tech BLE 5.x Dongle Technical Reference



Table of Contents

1. Revision History	3
3. Software Compatibility	4
4. Hardware Description	4
5. Custom software.....	4
6. Bootloader	4
7. CP2102 Connections	4
A. Power Amplifier Power Supply Connection.....	4
B. LED control.....	4
C. USB Device Descriptor	5
8. Hardware Abstraction Layer Header	6
9. USB Device Descriptor	7



1. Revision History

Revision	Author	Comments
A	Skip Hansen (08/24/2020)	Initial Draft



3. Software Compatibility

The Darwin Tech BLE 5.x Dongle is intended to be used with host side applications written to use a Silicon Labs compatible Network Co-Processor (NCP). In this mode the communication between the host side application and dongle is defined by the Silicon Labs Proprietary Protocol called BGAPI.

The dongle is **NOT** compatible with the Bluetooth SIG's Host Controller Interface (HCI) control protocol.

The Darwin Tech BLE 5.x Dongle appears to the host side application software as a Virtual COM Port (VCP) device. Application software running on the PC accesses the dongle as it would access a standard hardware COM port.

NOTE: The Silicon Labs USB to UART Bridge Virtual COM Port (VCP) drivers may be needed for some operating systems. See <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers> for more details.

4. Hardware Description

The Darwin Tech BLE 5.x Dongle in a simple two chip design consisting of an EFR32BG13P632F512GM32-Blue Gecko Bluetooth® Low Energy SoC and a CP2102N USBXpress USB-to-UART bridge controller.

The device is powered entirely from the USB port.

Transmit and receive data LEDs provide visual indication of serial data flowing between the host and the dongle. The activity shown by these LEDs are not directly related to data flowing on Bluetooth.

5. Custom software

This section provides information needed for developers wanting to create custom firmware to run in the Darwin Tech BLE 5.x Dongle. This information is **NOT NEEDED** for the normal use case where the stock factory firmware is used along with a host side application program.

NB: Loading custom firmware voids the factory warrantee.

6. Bootloader

The factory firmware includes the standard BGAPI gecko bootloader which can be used to load custom firmware into the Dongle. It is **strongly suggested** that Silicon Labs development boards be used for initial development and testing of custom code to prevent dongle “bricking” due to programming errors.

7. CP2102 Connections

The RTS output from the EFR32 on PC10 is connected to the CTS input of a CP2102. The CTS input to the EFR32 on PC11 is connected to the RTS output of a CP2102. With these connections the “location” of PC10 should be programmed to #10 and the “location” of PC11 should be programmed to #12.

A. Power Amplifier Power Supply Connection

The internal DC-DC 1.8 volt regulator powers the Bluetooth power amplifier.

B. LED control

The LEDs are controlled by the CP2102 USB-to-UART bridge controller; they cannot be controlled by the Blue Gecko SoC.

C. USB Device Descriptor

The USB device descriptor is controlled by the CP2102 USB-to-UART bridge controller. Refer to the USBXpress™ Family CP2102N Data Sheet for details.



8. Hardware Abstraction Layer Header

The following is hal-config-board.h used by the factory firmware:

```

/*****
 * (C) Copyright 2019 Darwin Tech, LLC, http://www.darwintechologiesllc.com
 *****/
 * This file is licensed under the Darwin Tech Embedded Software License Agreement.
 * See the file "Darwin Tech - Embedded Software License Agreement.pdf" for
 * details. Read the terms of that agreement carefully.
 *
 * Using or distributing any product utilizing this software for any purpose
 * constitutes acceptance of the terms of that agreement.
 *****/
#ifndef HAL_CONFIG_BOARD_H
#define HAL_CONFIG_BOARD_H

#include "em_device.h"
#include "hal-config-types.h"

// [CMU]
#define BSP_CLK_HFXO_PRESENT (1)
#define BSP_CLK_HFXO_FREQ (3840000)
#define BSP_CLK_HFXO_INIT CMU_HFXOINIT_DEFAULT
#define BSP_CLK_HFXO_CTUNE (346)
#define BSP_CLK_HFXO_CTUNE_TOKEN (0)

#define BSP_CLK_LFXO_PRESENT (1)
#define BSP_CLK_LFXO_FREQ (32768)
#define BSP_CLK_LFXO_INIT CMU_LFXOINIT_DEFAULT
// [CMU]$

// [DCDC]
#define BSP_DCDC_PRESENT (1)
#define BSP_DCDC_INIT EMU_DCDCINIT_DEFAULT
// [DCDC]$

// [UARTNCP]

// PC11 is USART0 CTS hardware flow control input
#define BSP_UARTNCP_USART_PORT (HAL_SERIAL_PORT_USART0)
#define BSP_UARTNCP_CTS_PIN (11)
#define BSP_UARTNCP_CTS_PORT (gpioPortC)
#define BSP_UARTNCP_CTS_LOC (12)

#define BSP_UARTNCP_RX_PIN (1)
#define BSP_UARTNCP_RX_PORT (gpioPortA)
#define BSP_UARTNCP_RX_LOC (0)

#define BSP_UARTNCP_TX_PIN (0)
#define BSP_UARTNCP_TX_PORT (gpioPortA)
#define BSP_UARTNCP_TX_LOC (0)

// PC10 is USART0 RTS hardware flow control output
#define BSP_UARTNCP_RTS_PIN (10)
#define BSP_UARTNCP_RTS_PORT (gpioPortC)
#define BSP_UARTNCP_RTS_LOC (10)
// [UARTNCP]$

// [USART0]
#define BSP_UARTNCP_USART_PORT (HAL_SERIAL_PORT_USART0)
#define BSP_USART0_CTS_PIN (11)
#define BSP_USART0_CTS_PORT (gpioPortC)
#define BSP_USART0_CTS_LOC (10)

#define BSP_USART0_RTS_PIN (10)
#define BSP_USART0_RTS_PORT (gpioPortC)
#define BSP_USART0_RTS_LOC (10)

#define BSP_USART0_RX_PIN (1)
#define BSP_USART0_RX_PORT (gpioPortA)
#define BSP_USART0_RX_LOC (0)

#define BSP_USART0_TX_PIN (0)
#define BSP_USART0_TX_PORT (gpioPortA)
#define BSP_USART0_TX_LOC (0)
// [USART0]$
// Note: These might be necessary for the selected mcu or board as well.
#define HAL_PA_ENABLE (1)
// #define FEATURE_PA_HIGH_POWER

#endif /* HAL_CONFIG_BOARD_H */

```

9. USB Device Descriptor

Device Descriptor:

bLength	18
bDescriptorType	1
bcdUSB	2.00
bDeviceClass	0 (Defined at Interface level)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
idVendor	0x10c4
idProduct	0xea60
bcdDevice	1.00
iManufacturer	1 Darwin Tech LLC
iProduct	2 CP2102N USB to UART Bridge Controller
iSerial	3 0001
bNumConfigurations	1

Configuration Descriptor:

bLength	9
bDescriptorType	2
wTotalLength	32
bNumInterfaces	1
bConfigurationValue	1
iConfiguration	0
bmAttributes	0x80

(Bus Powered)

MaxPower	100mA
----------	-------

Interface Descriptor:

bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	2
bInterfaceClass	255 Vendor Specific Class
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	0

Endpoint Descriptor:

bLength	7
bDescriptorType	5
bEndpointAddress	0x02 EP 2 OUT
bmAttributes	2
Transfer Type	Bulk
Synch Type	None
Usage Type	Data
wMaxPacketSize	0x0040 1x 64 bytes
bInterval	0

Endpoint Descriptor:

bLength	7
bDescriptorType	5
bEndpointAddress	0x82 EP 2 IN
bmAttributes	2
Transfer Type	Bulk
Synch Type	None
Usage Type	Data
wMaxPacketSize	0x0040 1x 64 bytes
bInterval	0

Device Status: 0x0000