

Type ZF/ZY

Bluetooth® smart BLE modules

This document covers basic questions about Murata's Bluetooth® Low Energy Type ZF and ZY modules.

Q1. What is the difference between Type ZY and ZF?

The Type ZF module is supplied without an antenna and is smaller in size. It requires an external antenna; however, it is still FCC certified.

Q2. What is the module part number to use when ordering engineering samples?

The sample part number is LBCA2HNZYZ-TEMP for ZY.

Q3. What is the module part number to use when ordering for production run?

The module part number is LBCA2HNZYZ-711 for ZY and LBCA2BZZFZ-710 for ZF.

Q4. What is the module part number for the design kit?

The module design kit part number is LBCA2HNZYZ-TEMP-DK.

Below is a design kit with a similar module called Type ZF. For the Type ZY, you will get a daughterboard with Type ZY mounted as shown in figure 2.

All contents of the design kit is shown in figure 1.

Keil™ is the IDE tool that runs on Windows XP or Windows 7. It is available as a free version. Instructions for its use are available through [my Murata](#).

Hardware contents

- Motherboard
- Type ZF daughterboard
- J-Link debugging adapter

Software contents

- Software on download-link
- Keil™ tools, with DA14580 peripherals
- Sample application/ Profile source code
- OTP programming utility
- Power consumption profiling tool
- Documentation



Figure 1. Type ZF Design Kit contents



Figure 2. Daughterboard with Type ZY mounted

Q5. Can a daughterboard with Type ZY be purchased separately?

Yes; the part number is LBCA2HNZYZ-TEMP-D-MU. However, it will only work with the motherboard shown in figure 1.

Q6. Do I need to add a fast clock to the module?

No; the module includes a 16MHz fast clock.

Q7. What is the host interface?

Type ZY has a UART connection available for interfacing to the host MCU when design is implemented to use the host MCU. The UART interface can be either 4-wire UART or 2-wire UART.

For a 4-wire UART, the table below is the required connection. Note that the interface name is the viewpoint from the module. For example, module pin 1 is input for UART and output from the host MCU.

Module pin	Terminal Name	Description
12	PO_2	CTS
11	PO_3	RTS
24	PO_4	UART_TX
1	PO_5	UART_RX

For a 2-wire UART with software flow control, you can disable the hardware flow control by calling the below API. Pin 12 or 11 will not be required in this case

UART Flow off

Disables the UART RTS signal (active low)

Function Name	void uart_flow_off(void)
Function Description	Disables the UART RTS signal (active low)
Parameters	None
Return values	None
Notes	The RTS pad, if configured, will be driven low (active)

Q8. Does the module support iBeacon? Can these modules be pre-programmed with iBeacon support?

Yes, the module supports iBeacon. The modules cannot be pre-programmed; download the source code of the iBeacon application and program it into the module. Customer needs to set up the “Device Name” on each product; therefore, pre-programming is not feasible.

Q9. Does the user need to be an MFi licensee to evaluate iBeacon?

Yes, the customer needs to have an MFi license to get the iBeacon sample code.

Q10. What is the current consumption when in iBeacon mode?

The current consumption in iBeacon mode 4.8mA Tx, 5.1mA Rx.

Q11. Is there a CPU inside the module and what is its processing power?

There is an ARM Cortex-M0 inside the module running with a 16MHz clock. In addition, there is a dedicated Link Layer processor and AES-128 bit encryption processor.

Q12. What is the memory size inside the module and how is it allocated?

DA14580 inside the module includes 32KB OTP, 42KB System SRAM, 84KB ROM, and 8KB Retention RAM.

84KB ROM contains Bluetooth® Smart Protocol stack as well as the boot code sequence.

32KB OTP is a one-time programmable memory array, used to store the application code as well as Bluetooth® Smart Profiles. It also contains the system configuration and calibration data.

42KB System SRAM is primarily used for mirroring the program code from the OTP when the system wakes/powers up. It also serves as DATA RAM for intermediate variables and various data that the protocol requires. Optionally, it can be used as extra memory space for the BLE TX and RX data structures.

8KB retention RAM has 4 special low leakage SRAM cells used to store various data of the Bluetooth® Smart protocol as well as the system’s global variables and processor stack when the system goes into deep sleep mode. Storage of this data ensures secure and quick configuration of the BLE core after the system wakes up. Every cell can be powered on or off according to the application needs for retention area when in deep sleep mode.

Q13. How is memory used during development?

During the development stage, the application code is developed using the ARM Cortex M0 software environment. The compiled code is then downloaded into the System RAM or any Retention RAMs by means of SWD (JTAG) or any serial interface (e.g. UART). Address 0x00 is remapped to the physical memory that contains the code and the CPU is configured to reset and execute the code from the remapped device. This mode enables application development, debugging, and on-the-fly testing. Once the application code is ready and verified, the code can be burned into the OTP. When the system boots/wakes up, the DMA of the OTP controller will automatically copy the program code from the OTP into the system RAM.

Q14. What are SWDIO and SWCLK pins?

These pins are not for OTP writing but for JLink adapter and for firmware development. Module pin 4 is the clock signal and pin 5 is the data input/output.

Q15. Can the module run standalone without the host MCU?

Yes; the software can be configured such that both the application and the BLE profile can run inside the module and will not need an external MCU.

Another option is to run the application code in the host MCU to control the BLE module through a serial interface. Both options are shown in figure 3.

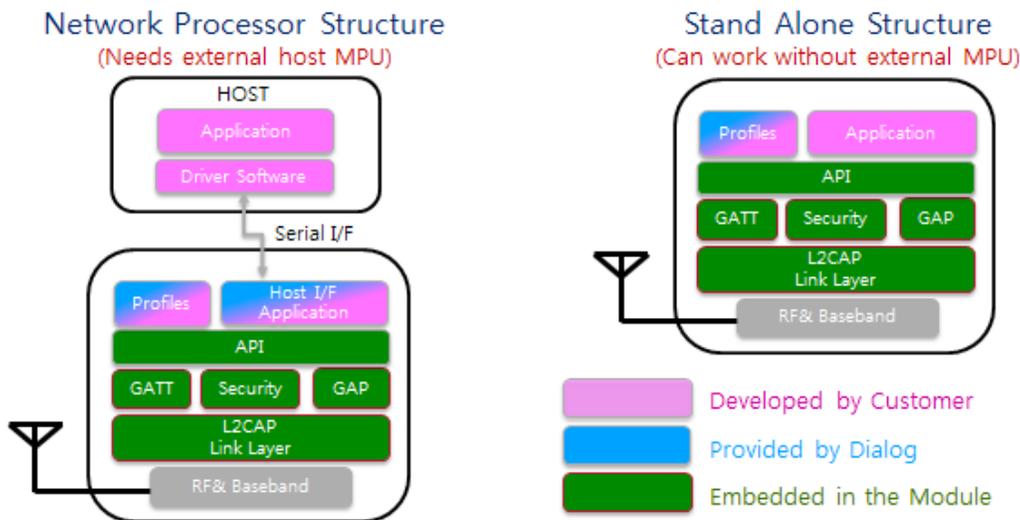


Figure 3. Software structure options.

Q16. How much code space will the BLE profile use?

As an example, Bluetooth® Low Energy Glucose Profile occupies only 6KB, which can be programmed into the OTP. Another example is the heart rate monitor, which consumes 4.6KB out of 32KB OTP to burn the application code and heart rate monitor profile. This heart rate monitor solution will use 22KB out of 50KB RAM for the application code, heart rate monitor profile, and runtime memory consumption.

For a large BLE solution like a keyboard, 37KB out of 50KB RAM is required for keyboard application code, HOGP profile, and runtime memory consumption. 18.3KB out of 32KB OTP space is required to burn the application code and HOGP profile.

Q17. What BLE profile is available?

Bluetooth® standard has list of BLE profiles that are available. The module will support the BLE Profiles including but not limited to the list below.

Q18. What is the expected battery life of this module?

- FindMe
- Proximity
- Time
- Health Thermometer
- Heart Rate
- Blood Pressure
- Glucose
- Battery Profile/Service
- Alert Notification
- Phone alert status
- Running Speed and Cadence
- Cycling Speed and Cadence
- HID over GATT
- Scan Parameters

The average power consumption varies according to use cases. The table below is an estimate of CR2032 battery life depending on connection interval and data rate.

Note: Maximum of 20 bytes can be sent in one connection event.

Connection Interval	Data Rate	Average Current	Battery Life (CR2032: 225mAh)
1000 ms	160 bps	8.2 μ A	3.2 years
100 ms	1600 bps	69.6 μ A	4.5 months
10 ms	16 kbps	0.68 mA	2 weeks

Q19. What is the transmit power and range of the device?

The output transmit power at the antenna is -1dBm and open air range is ~70 meters.

Q20. Is an external 32.768KHz clock required?

The purpose of the 32.768KHz clock is to provide timing accuracy when the BLE comes out of sleep mode whether it is from sleep, deep sleep, or extended sleep mode. For sleep mode exceeding two seconds, this clock is required.

Although the external clock can be eliminated, it is recommended to have the external clock; a communication link between two BLE devices is impacted by the timing of the active mode turn on from the sleep modes.

Another option is to use a 32.768KHz clock from an external MCU, but peak-to-peak voltage may not exceed 1.2V. This clock can be either square wave or sine wave. Voltage attenuation can be achieved by placing a capacitor between the source and the module if needed. Also, a 32.768KHz signal input to XTAL32P should be turned off or its peak-to-peak voltage should be less than 400mV when VBAT is turned off.

Q21. What tools are available for software development and debugging?

The ARM Cortex M0 can work with existing ARM tool chains. IDE tools for software development and debugger such as Keil™ uVision and IAR can be used. If the code size is smaller than 32KB, which is equivalent to OTP size of DA14580, most of these tools are free to use.

The reference software projects in the SDK provided are developed using Keil™ uVision.

Additionally, latest SDK (3.0.4 onwards) supports the GCC compiler.

Q22. What is the regulatory certification for the module?

Type ZY has FCC, IC, TELEC certification and CE compliance.

FCC ID: VPYLBZY

IC: 772C-VPZY

TELEC: 001-P00500

CE: ETSI EN300328 v1.8.1 conducted test report available and used for the customer's CE declaration

This certification can be applied to Type ZF/ZW; see my Murata.

Q23. Are there any impedance controls required for the BLE_ANT_IN and BLE_ANT_OUT nets?

The trace needs to be 50 Ω ; see my Murata for layout recommendations.

Q24. How do I get SDK?

Go to the [Dialog support site](#) and register. Go to Products -> DA14580 -> Software & Tools.

Q25. How do I get detailed software information?

Go to the [Dialog Support site](#). Go to Products -> DA14580 -> Documents

Q26. How do I get started with the software?

Please refer to "[First Step Guide](#)" at my Murata.

Download SDK, SmartSnippets, and Connection Manager software from the [Dialog Support site](#). (for source code and test application).

Download the Keil tool from the [Keil website](#) (for software development). Download the [Starter's Guide](#).

Download [Segger J-Link tool](#) (for debugging); choose the J-Link Software and documentation pack for Windows.

Q27. What's the difference in content between my Murata and the Dialog Support Site?

my Murata has documents related to the Murata module, such as the module datasheet, EVK schematic, EVK user guide, Application note for Murata module/EVK.

my Murata does not have API documents, functional specification, software development kit (SDK), evaluation tool, IC spec, etc. If you need such content, please go to the Dialog support site.

Technical support

Go to <https://my.murata.com/en/home>. You will need your serial number for registration.

Q28. Q. What support does Murata provide for modules?

In most cases support is provided through the [Murata website](#). Inquiries can be made through the web page or through your sales representative. Click [here](#) and use the Contact Us form.

Murata provides technical support including technical documentation through the [my Murata technical support](#) web portal. You will need to obtain the Murata design kit and use the serial number on the mother board to register at my Murata. Register first to access [my Murata](#), then request approval to access your product section. Once approved, you will have access to software as well as technical documentation to start development into your product. Be sure to read and agree to the SLA before getting the software.