

## AT32WB415 BLE MIDI入门指南

## 前言

音乐数字接口(Musical Instrument Digital Interface, 以下简称 MIDI)是一个工业标准的电子通讯协议, 本身不传输声音数据, 只传输相关的控制讯号。过去可以使用 USB 等接口传输信息给计算机等设备进行声音的组合, 现在在蓝芽通讯上, 我们也可以使用通用的规范来进行传输。本应用笔记将会介绍如何建立一个 MIDI 的控制器, 发送控制讯号给手机进行播放。

支持型号列表:

支持型号	AT32WB415
------	-----------

## 目录

<b>1</b>	<b>BLE MIDI Profile .....</b>	<b>5</b>
1.1	MIDI Packet.....	5
1.2	联机间隔.....	5
1.3	BLE MIDI 的服务与特征.....	5
1.4	数据传输.....	6
<b>2</b>	<b>MIDI 控制器.....</b>	<b>7</b>
2.1	功能简介.....	7
2.2	资源准备.....	7
2.3	软件设计.....	7
2.4	实验效果.....	7
<b>3</b>	<b>文档版本历史 .....</b>	<b>8</b>

## 表目录

表 1. Note Hex Value .....	5
表 2. BLE MIDI Service and Characteristic.....	5
表 3. BLE MIDI 特征属性.....	6
表 4. 文档版本历史 .....	8

## 图目录

图 1. MIDI Packet.....	5
图 2. A Simple MIDI System.....	6

# 1 BLE MIDI Profile

BLE MIDI Profile 的详细内容，可以上 MIDI Association 的官方网站去下载[规格书](#)，这里简单将各个必要条件，归纳成各小节简单描述。

## 1.1 MIDI Packet

透过 BLE 传输的 MIDI 封包格式如图 1

图 1. MIDI Packet

<b>Header</b>	<b>Timestamp</b>	<b>Status</b>	<b>Note</b>	<b>Velocity</b>
Byte[0]	Byte[1]	Byte[2]	Byte[3]	Byte[4]

其中 Header 的 bit-7 是起始位强制为 1, bit-6 为保留位强制为 0; Timestamp 的 bit-7 为起始位强制为 1。也就是说 Header 的 bit-0 到 bit-5 为 13 位时间戳的高六位, Timestamp 的 bit-0 到 bit-6 为 13 位时间戳的低七位。

Status, Note 及 Velocity 在 MIDI 的[规格书](#)有定义, 这边简单列出部份 Note 对应到的音调, 如表 1。

表 1. Note Hex Value

Note	C	C#	D	D#	E	F	F#	G	G#	A
Hex Value	0x3C	0x3D	0x3E	0x3F	0x40	0x41	0x42	0x43	0x44	0x45

## 1.2 联机间隔

BLE MIDI 设备必须请求 15 毫秒或更短的联机间隔, 在大多数的 MIDI 应用中, 首选是较短的联机间隔; 实际在联机时, 支持以中心(Central)和外围(Peripheral)两者之中联机间隔较短者。另外, 若是与 Apple 的装置联机, 这个间隔时间建议为 11.25 毫秒到 15 毫秒之间。

## 1.3 BLE MIDI 的服务与特征

BLE MIDI Profile 并未向 Bluetooth SIG 注册一个 short UUID, 所以它的服务及特征为完整的 128-bit, 具体的数值如表 2。

表 2. BLE MIDI Service and Characteristic

Name	UUID
MIDI Service	03B80E5A-EDE8-4B33-A751-6CE34EC4C700
MIDI Data I/O Characteristic	7772E5DB-3868-4112-A1A9-F2669D106BF3

特征的属性如表 3, 这边要注意到权限的部份必须完全按照规格去开启, 否则联机无法建立。

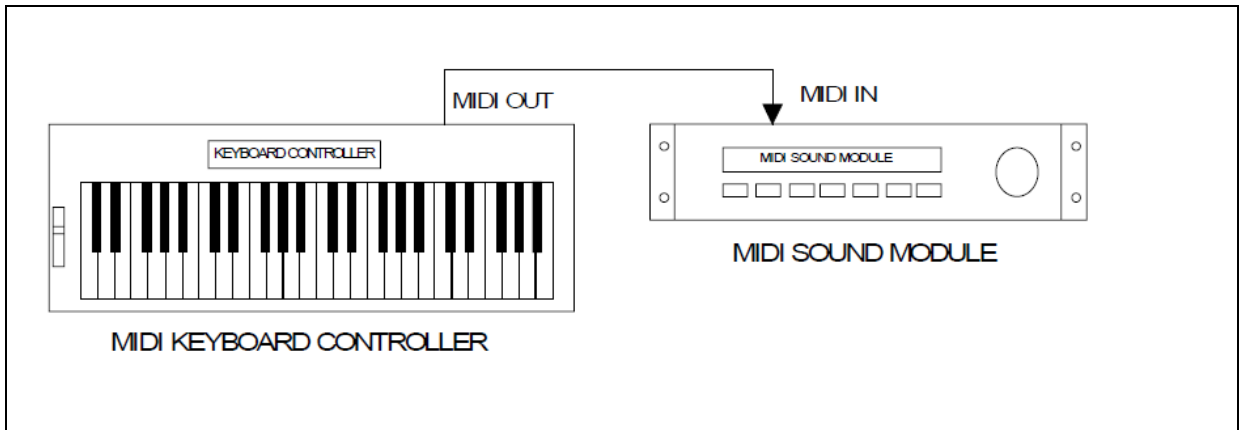
表 3. BLE MIDI 特征属性

Name	Permission	Properties	Comment
MIDI Data I/O	Read/Write/Notify	Pairing Required	读、写和通知都推荐加密。 读响应不带负载；写的操作不需要回复

## 1.4 数据传输

在本应用笔记中，AT32WB415 是作为一个 MIDI 控制器与手机联机，示意图如图 2。

图 2. A Simple MIDI System



WB415 就如同 MIDI 键盘一般，发送 MIDI 数据给手机端进行译码，从而播放出对应的音调甚至旋律。

## 2 MIDI 控制器

### 2.1 功能简介

本应用笔记实作 MIDI 控制器的部份，透过按下发展板上的 USER KEY，将一串 MIDI 数据传给对接的装置，装置收到后便会播放出声音。

### 2.2 资源准备

1) 硬件环境:

对应产品型号的 AT-START BOARD

2) 软件环境

SourceCode\at32wb415\_ble\_midi\_ble\_v2.0.0\projects\ble\_app\_gatt\_midi

SourceCode\at32wb415\_ble\_midi\_mcu\_v2.0.0\utilities\ble\_midi\_mcu\mdk\_v5

### 2.3 软件设计

1) 配置流程

- 定义传输的数据格式为完整的 MIDI Message
- 定义出要传输的旋律(Melody)
- 配置 USER Key，使其按下后将 MIDI Message 传给蓝芽端
- 蓝芽端的代码无须修改，仅将 MCU 的数据透传给对接装置

2) 代码介绍

- midiCommand 函数代码描述

```
static void midiCommand(uint8_t status, uint8_t note, uint8_t velocity)
{
    /* | header | timestamp | status | note | velocity | */
    // MIDI data goes in the last three bytes of the midiMessage array:
    midiMessage[MIDI_STATUS_BYTE] = status;
    midiMessage[MIDI_NOTE_BYTE] = note;
    midiMessage[MIDI_VELOCITY_BYTE] = velocity;
}
```

- MIDI Message 和 Melody 代码描述

```
/* | header | timestamp | status | note | velocity | */
static uint8_t midiMessage[] = {0x80, 0x80, 0x00, 0x00, 0x00};
/* Note value */
static uint8_t midiMelody[] = {64, 66, 71, 73, 74, 66, 64, 73, 71, 66, 74, 73};
```

### 2.4 实验效果

- 每按压一次 USER KEY 就发送一次旋律

### 3 文档版本历史

表 4. 文档版本历史

日期	版本	变更
2022.12.19	2.0.0	最初版本



## 重要通知 - 请仔细阅读

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途(及其依据任何司法管辖区的法律的对应情况)，或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：(A) 对安全性有特别要求的应用，如：生命支持、主动植入设备或对产品功能安全有要求的系统；(B) 航空应用；(C) 汽车应用或汽车环境；(D) 航天应用或航天环境，且/或(E) 武器。因雅特力产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险由购买者单独承担，并且独力负责在此类相关使用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技有限公司 保留所有权利