

BMduino-UNO 开发板

BM53A367A 使用手册

版本: V1.30 日期: 2024-08-22

www.bestmodulescorp.com



目录

简介	3
特性	3
方框图	4
引脚说明	5
技术规格	8
极限参数	8
建议工作条件	9
硬件概述	9
电源	
LED 指示灯	
USB 接口电路	11
UART, I ² C, SPI	14
复位电路	17
BM53A367A vs Arduino UNO R3	
烧录方法	
烧录方法 Arduino IDE 软件	18 19
烧录方法	18 19 19
烧录方法 Arduino IDE 软件 Arduino IDE 下载及安装 驱动及其他下载	18 19
烧录方法	18 19 20 20
烧录方法	18 19 20 20 23
烧录方法	18 19 20 20 23 26
烧录方法	18 19 20202326 26
烧录方法	
 烧录方法 Arduino IDE 软件 Arduino IDE 下载及安装 驱动及其他下载	
 烧录方法 Arduino IDE 软件 Arduino IDE 下载及安装 驱动及其他下载 IDE 设置 Arduino Library 参考 范例 Keil IDE 软件 Keil IDE 下载及安装 IDE 设置	18 19 202023262626262626
 烧录方法	
 烧录方法 Arduino IDE 软件 Arduino IDE 下载及安装	18 19 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 26 26 26 26 26 26 26 26 26 26 26 26 26 27
 烧录方法 Arduino IDE 软件 Arduino IDE 下载及安装 驱动及其他下载 IDE 设置 Arduino Library 参考 范例 Keil IDE 软件 Keil IDE 下载及安装 IDE 设置 运行范例 问题排查 申口未出现在端口菜单中	



简介

BMduino-UNO BM53A367A 是 Holtek 32-bit MCU HT32F52367 的开发板,特别 设计成与 Arduino UNO R3 开发板引脚兼容,可支持 Arduino IDE 开发平台和 Keil IDE 开发平台,帮助初学者更容易地学习程序设计。BM53A367A 采用 3.3V 系列 Arm[®] Cortex[®]-M0+ 内核的 HT32F52367 为主控 MCU,支持多种供电方式, 具备额外的 I²C、SPI、UART、USB 等常用通信接口。



■ 特性

- 主控 MCU: HT32F52367 (64-pin LQFP)
 - ♦ Cortex[®]-M0+, 60MHz
 - ◆ Flash 存储器: 256KB
 - ♦ SRAM: 32KB
- 31 个数字 I/O 引脚(其中 17 个可作为 PWM 输出)
- •7个模拟输入, A/D 转换器分辨率为 12-bit
- •14个外部中断
- EEPROM: 4KB
- 通信接口: UART, SPI, I²C, USB
- 电源输入: USB 接口 ×2, DC 接口,外部 Vin
 - ◆ e-Link32 Lite USB: Type-C USB 接口
 - ◆ Native USB: Type-C USB 接口,支持 BC 1.2 与快充 QC 2.0,快充预设升 压为 12V
 - ◆ DC 接口:圆形接口,连接外部电源(例如变压器),电压范围 DC 5~24V
 - ◆ 外部 Vin: Vin 引脚, 电压范围 DC 5~24V
- 除错工具: e-Link32 Lite (ICE),通过开发板上的 e-Link32 Lite 电路进行程序 除错
- 烧录方式:
 - ◆ ICP (In-Circuit Programing),通过开发板上的 e-Link32 Lite 电路进行烧录
 - ◆ IAP (In Application Programming),通过 COM 端口进行烧录
- 开发环境: 支持 Arduino IDE、Keil IDE 两种开发平台



- 接口资源: BMCOM1 (脚距 2.54mm, 3.3V/5V 可选), BMCOM2 (脚距 1.0mm, 3.3V/5V 可选), SPI1 (脚距 2.54mm, 3.3V/5V 可选)
- 开发板尺寸: 53.4mm×93.221mm×14.2mm, 与 Arduino UNO R3 开发板引脚相 容
- 开发板重量: 28.1g(净重)





引脚说明



引脚编号	功能	描述		
1	NC	未连接		
2	IOREF	/O 逻辑参考电压,+3.3V		
3	RESET	复位引脚		
4	+3V3	+3.3V 电源输出		
5	+5V	+5V 电源输出		
6	GND	电源地		
7	GND	电源地		
8	Vin	电源输入 5~24V		
9	A0/D14	模拟输入引脚 A0 / 数字引脚 D14		
10	A1/D15	模拟输入引脚 A1 / 数字引脚 D15		
11	A2/~D16	模拟输入引脚 A2 / 数字引脚 D16 带 PWM 功能		
12	A3/D17	模拟输入引脚 A3 / 数字引脚 D17		
13	A4/D18/SDA	模拟输入引脚 A4 / 数字引脚 D18 / I ² C0 接口的 SDA 引脚		
14	A5/D19/SCL	模拟输入引脚 A5 / 数字引脚 D19 / I ² C0 接口的 SCL 引脚		
15	$\sim D0/RX$	数字引脚 D0 带 PWM 功能 / UART0 接收引脚		
16	~D1/TX	数字引脚 D1 带 PWM 功能 / UART0 发送引脚		
17	~D2/INT0	数字引脚 D2 带 PWM 功能 / 外部中断 INT0		
18	~D3/INT1	数字引脚 D3 带 PWM 功能 / 外部中断 INT1		



引脚编号	功能	描述			
19	~D4/TX4/SS2/ INT2	数字引脚 D4 带 PWM 功能 / UART4 发送引脚 / SPI2 接口 的 SS2 引脚 / 外部中断 INT2			
20	~D5/RX4/ SCK2/INT3	数字引脚 D5 带 PWM 功能 / UART4 接收引脚 / SPI2 接口 的 SCK2 引脚 / 外部中断 INT3			
21	~D6/MOSI2/ INT4	数字引脚 D6 带 PWM 功能 / SPI2 接口的 MOSI2 引脚 / 外部中断 INT4			
22	D7/MISO2/ INT5	数字引脚 D7 / SPI2 接口的 MISO2 引脚 / 外部中断 INT5			
23	~D8/INT6	数字引脚 D8 带 PWM 功能 / 外部中断 INT6			
24	~D9/INT7	数字引脚 D9 带 PWM 功能 / 外部中断 INT7			
25	~D10/SS/ INT8	数字引脚 D10 带 PWM 功能 / SPI0 接口的 SS 引脚 / 外部 中断 INT8			
26	~D11/MOSI/ INT9	数字引脚 D11 带 PWM 功能 / SPI0 接口的 MOSI 引脚 / 外部中断 INT9			
27	~D12/MISO/ INT10	数字引脚 D12 带 PWM 功能 / SPI0 接口的 MISO 引脚 / 外部中断 INT10			
28	~D13/SCK	数字引脚 D13 带 PWM 功能 / SPI0 接口的 SCK 引脚			
29	GND	电源地			
30	AREF/A6/D30	模拟参考电压 / 模拟输入引脚 A6 / 数字引脚 D30			
31	A4/D18/SDA	模拟输入引脚 A4 / 数字引脚 D18 / I ² C0 接口的 SDA 引脚			
32	A5/D19/SCL	模拟输入引脚 A5 / 数字引脚 D19 / I ² C0 接口的 SCL 引脚			

Pin 13 与 Pin 31 共用引脚, Pin 14 与 Pin 32 共用引脚。

BMCOM1:可作为 I²C 或 UART 接口 (I²C1、Serial1)



5 – D22/INT14 4 – D21/TX1/SCL1 3 – D20/RX1/SDA1 2 – VDD1 1 – GND



引脚编号	功能	描述		
1	GND	电源地		
2	VDD1	3.3V 或 5V 电源输出,由相邻 Jumper 设定		
3	D20/RX1/SDA1	数字引脚 D20 / UART1 接收引脚 / I ² C1 接口的 SDA1 引脚		
4	D21/TX1/SCL1	数字引脚 D21 / UART1 发送引脚 / I ² C1 接口的 SCL1 引脚		
5	D22/INT14	数字引脚 D22 / 外部中断 INT14		

Pin 3、Pin 4、Pin 5 的电压电平 (3.3V 或 5V) 由相邻 Jumper 设定。

BMCOM2:可作为 I²C 或 UART 接口 (I²C2、Serial2)



引脚编号	功能	描述		
1	GND	电源地		
2	VDD2	3.3V 或 5V 电源输出,由相邻 Jumper 设定		
3	~D23/RX2/SDA2	数字引脚 D23 带 PWM 功能 / UART2 接收引脚 / I ² C2 接口的 SDA2 引脚		
4	~D24/TX2/SCL2	数字引脚 D24 带 PWM 功能 / UART2 发送引脚 / I ² C2 接口的 SCL2 引脚		
5	~D25/INT15	数字引脚 D25 带 PWM 功能 / 外部中断 INT15		

Pin 3、Pin 4、Pin 5 的电压电平 (3.3V 或 5V) 由相邻 Jumper 设定。

注意: I²C1 与 I²C2 共用 MCU 上 的 同 一 个 实 体 I²C, BMCOM1 的 I²C 与 BMCOM2 的 I²C 无法同时使用。





SPI1:	可作为 SPI 或 UART 接口 (SPI1	Serial3)	
-------	---------------------	------	----------	--

引脚编号	功能	描述		
1	D26/MISO1	数字引脚 D26 / SPI1 接口的 MISO1 引脚		
2	VDD3	3.3V 或 5V 电源输出,由相邻 Jumper 设定		
3	D27/TX3/SCK1	数字引脚 D27 / UART3 发送引脚 / SPI1 接口的 SCK1 引脚		
4	D28/RX3/MOSI1	数字引脚 D28 / UART3 接收引脚 / SPI1 接口的 MOSI 引脚		
5	D29/SS1/INT13	数字引脚 D29 / SPI1 接口的 SS1 引脚 / 外部中断 INT13		
6	GND	电源地		

Pin 1、Pin 3、Pin 4、Pin 5 的电压电平 (3.3V 或 5V) 由相邻 Jumper 设定。

技术规格

极限参数

• 通过 e-Link32 Lite USB 接口 / Native USB 接口给 MCU 供电。

注意: USB 供电时电压不可小于 4.5V。

- 通过 DC Jack 输入 5~24V,由降压电路降压为 5V 及 3.3V 输出并给 MCU 供电。
- •通过 Vin 输入 5~24V 电压,由降压电路降压为 5V 及 3.3V 输出并给 MCU 供电。

符号	参数	最小	典型	最大	单位
VinMAX	Vin 引脚输入电压	24		26	V
VUSBMAX	USB 连接器输入电压	4.5	5.5		V



建议工作条件

符号	参数	测试条件	最小	典型	最大	单位
Vin	输入电压		5		24	V
	5W 检山山运	Vin>11V			1.3	А
T	5V 1扣出电沉	Vin=4.6V			800	mA
LOUT	3.3V 输出电流				500	mA
	I/O 输出电流				16	mA



PCBA 反面图



电源

- 通过 e-Link32 Lite USB 接口输入 5V, 由降压电路降压为 5V 及 3.3V 输出并 给 MCU 供电。
- 通过 Native USB 接口输入 5V 或 12V (支持 QC 2.0 接快充会升压到 12V), 由降压电路降压为 5V 及 3.3V 输出并给 MCU 供电。
- 通过 DC Jack 输入 5~24V,由降压电路降压为 5V 及 3.3V 输出并给 MCU 供电。
- 通过 Vin 输入 5~24V 电压,由降压电路降压为 5V 及 3.3V 输出并给 MCU 供电。



电源接口图

注意: 当 DC Jack / Native USB / e-Link32 Lite USB 与 Vin 同时供电时,为了防止电压反灌问题, Vin 输入电压必须大于或等于 DC Jack / Native USB / e-Link32 Lite USB。

LED 指示灯



LED 配置图

- PWR: 电源 LED。
- D13: 与数字引脚 13 相连,用于范例程序,观察程序状态用。



- TX 和 RX:指示 UART 的 TX / RX 总线传输状态,数据传输时闪烁。
- RDY: 指示 e-link32 Lite 的 USB 连线状态,当与电脑完成连线时点亮 LED。
- STS: 指示 e-link32 Lite 的烧录状态,当烧录程序时会闪烁。

USB 接口电路



BMduino-UNO BM53A36A 开发板上有 2 个 USB 接口,分别是 e-Link32 Lite 及 NATIVE。

- e-Link32 Lite: e-Link32 Lite 主要提供几个功能:
 - 1. Arduino IDE 或 Keil IDE 做程序开发时,此 USB 接口连接电脑实现对主控 MCU 烧录。
 - 2. 支持 VCP (Virtual COM Port) 连接到主控 MCU 的 TX 和 RX。
 - 3. 在 Keil IDE 的开发环境下提供程序除错的功能 (例如,设置断点)。

BMduino-UNO BM53A367A 预设是通过 e-Link32 Lite 电路做烧录,烧录时不 会使用到 TX 及 RX 引脚。

- Native USB: Native USB 主要提供几个功能:
 - 1. 供电接口

当 QC 2.0 的充电器通过 Native USB 给 BMduino-UNO BM53A36A 开发板 供电时, Native USB 接口预设升压为 12V。

2. USB 通信

USB 接口预设功能为 VCP,通过 Arduino API 的 SerialUSB 进行通信。使用 BMduino-UNO BM53A36A 开发板来开发具有 USB 的产品时(例如鼠标 或键盘),此 USB 接口即是产品的 USB 接口。若要使用 Keyboard / Mouse 必须将头文件加入 Sketch 中才可使用,例如 #include <Mouse.h>。

Obj	SerialUSB	Keyboard	Mouse
	(USB CDC)	(USB HID)	(USB HID)
引脚	仅 Native USB 接口	仅 Native USB 接口	仅 Native USB 接口

例如, BMduino-UNO 使用 Native USB 接口连接 PC 作为 USB 键盘或鼠标时, 只需先声明 Keyboard.h 及 Mouse.h 并在程序中使用 Keyboard 及 Mouse 物件。



范例程序:

```
#include "Mouse.h"
#include "Keyboard.h"
// 为五个按钮设置引脚编号:
const int upButton = 2;
const int downButton = 3;
const int leftButton = 4;
const int rightButton = 5;
const int mouseButton = 6;
                       // X 或 Y 移动的输出范围;影响移动速度
int range = 5;
int responseDelay = 20; // 鼠标应答延时时间,单位为ms
void setup() {
  // 按钮输入初始化:
  pinMode(upButton, INPUT);
 pinMode(downButton, INPUT);
  pinMode(leftButton, INPUT);
  pinMode(rightButton, INPUT);
 pinMode(mouseButton, INPUT);
  // 鼠标控制初始化:
 Mouse.begin();
 // 键盘控制初始化:
  Keyboard.begin();
  // SerialUSB 初始化
  SerialUSB.begin(9600);
}
void loop() {
 // 读取按钮状态:
 int upState = digitalRead(upButton);
 int downState = digitalRead(downButton);
  int rightState = digitalRead(rightButton);
  int leftState = digitalRead(leftButton);
  int clickState = digitalRead(mouseButton);
  // 根据按钮状态计算移动距离:
  int xDistance = (leftState - rightState) * range;
  int yDistance = (upState - downState) * range;
  // 如果 X 或 Y 非零,则移动:
  if ((xDistance != 0) || (yDistance != 0)) {
  // 移动鼠标光标
   Mouse.move(xDistance, yDistance, 0);
    // 键盘输出鼠标的坐标
    Keyboard.print("Move: ");
    Keyboard.print(xDistance);
    Keyboard.print(',');
    Keyboard.println(yDistance);
    Keyboard.flush();
```



```
// SerialUSB 输出鼠标的坐标
   SerialUSB.print("Move: ");
   SerialUSB.print(xDistance);
   SerialUSB.print(',');
   SerialUSB.println(yDistance);
   SerialUSB.flush();
 }
 // 如果鼠标按钮按下:
 if (clickState == HIGH) {
   // 如果鼠标未按下,则按下:
   if (!Mouse.isPressed(MOUSE LEFT)) {
     Mouse.press(MOUSE LEFT);
     // 如果键盘 println 未注释,则鼠标光标改变会导致字符串被覆盖
     // Keyboard.println("Press MOUSE LEFT");
     // Keyboard.flush();
     SerialUSB.println("Press MOUSE LEFT");
     SerialUSB.flush();
   }
 }
 // 否则鼠标按钮未按下:
 else {
   // 如果按下鼠标,则释放鼠标:
   if (Mouse.isPressed(MOUSE LEFT)) {
     Mouse.release(MOUSE LEFT);
     // 如果键盘 println 未注释,则鼠标光标改变会导致字符串被覆盖
     // Keyboard.println("Release MOUSE LEFT");
     // Keyboard.flush();
     SerialUSB.println("Release MOUSE LEFT");
     SerialUSB.flush();
   }
 }
 // 延时使鼠标不会移动太快:
 delay(responseDelay);
}
```

UART, I²C, SPI



• 支持 5 组 UART,物件名称分别为 Serial、Serial1~Serial4。

Obj	Serial	Serial1	Serial2	Serial3	Serial4
引脚	RX (D0) / TX (D1)	RX1 (D20) / TX1 (D21) (BMCOM1)	RX2 (D23) / TX2 (D24) (BMCOM2)	RX3 (D28) / TX3 (D27) (SPI1)	RX4 (D5) / TX4 (D4)

例如,要使用 BMCOM1 的 UART 接口时,只需在程序中使用 Serial1 物件。 范例程序:

```
void setup() {
    // 串行通信初始化:
    Serial1.begin(9600);
}
void loop() {
    // 发送模拟输入 0 的值:
    Serial1.println(analogRead(A0));
    // 等待一段时间使 A/D 转换器在最后一次读取后稳定下来:
    delay(20);
}
```

在 BMduino-UNO BM53A367A 开发板使用 Arduino Serial Library 时,数据位可以是 7~9 位,而在 Arduino IDE 预设则是 5~8 位。



• 支持 3 组 I²C,物件名称 Wire 其中 Wire1 与 Wire2 共用实体 I²C,同时只能有一组通信。

Obj	Wire	Wire1	Wire2
引助	SDA/SCL或A4/A5	SDA1 (D20) / SCL1 (D21)	SDA2 (D23) / SCL2 (D24)
ין גען	(共用 I/O)	(BMCOM1)	(BMCOM2)

- 注意: I²C1 与 I²C2 共用 MCU 上的同一个实体 I²C, 链接库 Wire1 与 Wire2 无法同时工作,只能选择使用其一 (Wire1 或 Wire2)。如果程序中同时 存在 Wire1 及 Wire2,请参考下方做法:
 - 1. 通过 Wire1.begin() 或 Wire2.begin() 进行初始化。
 - 2. 需要切换当前 Wire1 或 Wire2, 请通过 Wire1.end() 或 Wire2.end() 关 闭当前的 Wire1 或 Wire2。
 - 3. 再通过 Wire1.begin() 或 Wire2.begin() 初始化另外一个 Wire1 或 Wire2。
 - 若 Wire1 与 Wire2 都进行初始化,以先初始化的 Wire1 或 Wire2 为有效。

例如,要使用 BMCOM1 的 I²C 接口时,只需先声明 Wire.h 并在程序中使用 Wire1 物件。

范例程序:

```
#include <Wire.h>
void setup()
{
               // 加入 I<sup>2</sup>C 总线 ( 主机地址可选 )
 Wirel.begin();
 Serial.begin(9600); // 启动串行输出
}
void loop()
{
                        // 请求从机设备 2 发送 6 字节
 Wire1.requestFrom(2, 6);
                        // 从机发送的字节可能小于请求数量
 while(Wire1.available())
 {
                        // 接收1字节的字符
   char c = Wire1.read();
   Serial.print(c);
                         // 打印该字符
  }
 delay(500);
}
```

• 支持3组 SPI。

Obj	SPI	SPI1	SPI2
引脚	SS (D10) / MOSI (D11) / MISO (D12) / SCK (D13)	SS1 (D29) / MOSI1 (D28) / MISO1 (D27) / SCK1 (D26) (SPI1)	SS2 (D4) / MOSI2 (D6) / MISO2 (D7) / SCK2 (D5)

例如,要使用 6-pin 排针的 SPI1 接口时,只需先声明 SPI.h 并在程序中使用 SPI1 物件。



范例程序:

```
// 包含 SPI library:
#include <SPI.h>
// 设置 pin 10 为数字端口的从机选择引脚:
const int slaveSelectPin = 10;
void setup() {
 // 设置 slaveSelectPin 为输出:
 pinMode(slaveSelectPin, OUTPUT);
  // SPI1 初始化:
 SPI1.begin();
}
void loop() {
 // 通过数字端口的 6 个通道:
  for (int channel = 0; channel < 6; channel++) {</pre>
    // 将通道上的电阻从最小值改到最大值:
    for (int level = 0; level < 255; level++) {</pre>
     digitalPotWrite(channel, level);
     delay(10);
    }
    // 在最大值时等待1秒:
    delay(100);
   // 将通道上的电阻从最大值改到最小值:
    for (int level = 0; level < 255; level++) {
     digitalPotWrite(channel, 255 - level);
     delay(10);
    }
 }
}
void digitalPotWrite(int address, int value) {
 // SS 引脚为低时选择芯片:
  digitalWrite(slaveSelectPin, LOW);
 // 通过 SPI1 发送地址和值:
 SPI1.transfer(address);
 SPI1.transfer(value);
 // SS 引脚为高时取消选择芯片:
  digitalWrite(slaveSelectPin, HIGH);
```





开发板上的 6-pin 排针 (3-pin × 2 排, 90 度):

在 Arduino UNO R3 开发板,此 6-pin 排针称为 ICSP,引脚与开发板的 Pin 11、Pin 12、Pin 13 共用。

在 BMduino-UNO BM53A367A 开发板,此 6-pin 排针为 SPI1,并没有与开发 板的 Pin 11、Pin 12、Pin 13 共用引脚,是一个可单独使用的 SPI 接口。 Holtek 32-bit MCU 为 Cortex[®]-M0+ 架构,使用 SWD 接口烧录,不是使用 SPI 接口烧录。SWD 说明链接: <u>https://developer.arm.com/documentation/100956/0529/</u> <u>Arm-DSTREAM-Target-Interface-Connections/Signal-descriptions/Serial-Wire-</u>

Debug?lang=en

复位电路



- 可通过将 Pin 3 (RESET) 设置为低电平 1ms 触发系统复位。
- 可通过 RESET 按钮进行 MCU 复位。



• RESET 电路连接 e-Link32 Lite (VCP) 的 DTR,可通过开启 COM 端口进行 MCU 复位。

BM53A367A vs Arduino UNO R3

Board	BMduino-UNO BM53A367A	Arduino UNO R3			
内核	Cortex [®] -M0+, 60MHz	AVR 8-bit, 16MHz			
Flash / EEPROM / SRAM	256KB / 4KB / 32KB	32KB / 1KB / 2KB			
工作电压	3.3V	5V			
烧录模式	ICP/IAP(引导程序)	ICSP/IAP(引导程 序)			
烧录接口	SWD(目标板菜单)/UART(端口 菜单)	UART (端口菜单)			
开发环境	Arduino, Keil	Arduino			
I/O 驱动电流	16mA	20mA			



BM53A367A 在各 IDE 烧录方法整理如下。

1. Arduino:

- ICP 模式(预设模式):点选"项目(Sketch)"下方选单的"Upload"
 Sketch 将进行程序编译与烧录。通过下方状态视窗可观察烧录结果。详细 请 参考 Arduino 官 方 网 站: <u>https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics</u>。
- IAP 模式:与 ICP 模式方法相同,但进行"Upload"前需通过"工具→端口"选择板子对应的 COM 端口(与 Arduino UNO R3 做法相同),如下图。





 Keil: 点选 IDE 上方表单的 "Build" 进行程序编译,完成后点选表单的 "Download" 增进行程序烧录。通过下方 "Build Output" 视窗可观察编译 / 烧录结果。详细请参考 Keil 官方网站: <u>https://www2.keil.com/mdk5/learn</u>。

Arduino IDE 软件

Arduino IDE 下载及安装

打开 Arduino 官方网站 (<u>http://www.arduino.cc/en/Main/Software</u>) 下载对应电脑操 作系统的 Arduino IDE 软件并且执行安装。

Downloads



Arduino IDE 软件下载

安装完成后可以在文件夹中找到 Arduino.exe & Arduino_debug.exe 两个执行文件,两者皆可以开启程序开发环境,不一样的是 Arduino_debug.exe 可以开启除错视窗,当程序执行或编译时都会将信息显示在除错视窗上,方便用户查看哪个环节发生问题。





驱动及其他下载

USB 驱动下载

1. 在电脑 Win10 系统下,开发板与电脑相连时,电脑会自动挂载 VCP 驱动。 当驱动自动下载成功,可看到提示信息。



2. 在 XP 和 Win7 系统下, 需手动下载 VCP 驱动。详细做法请参考"Keil IDE 软件"章节。

IDE 设置

安装 Holtek HT32 Boards

 点选"文件→首选项",选择"设置"页签,在"附加开发板管理器网址" 内输入"<u>https://mcu.holtek.com.tw/arduino/ht32/package_ht32_index.json</u>",完成后点选"好"。

首选项		×				
设置 网络						
项目文件夹位置						
C:\Users\wenjielu\Documents\Arduino	浏览	ĩ				
编辑器语言 简体中文 (Chinese (China)) ~	(需要重启 Arduino)					
编辑器字体大小 20						
界面缩放: 🔽 自动调整 100 🔷 % (需要重启 Arduino)						
Theme: Default theme ~ (需要重启 Arduino)						
显示详细输出: 🗌 编译 🔲 上传						
编译器警告: 全部 🗸						
☑ 显示行号	□ 启用代码折叠					
☑ 上传后验证代码	□ 使用外部编辑器					
☑ 启动时检查更新	☑ 当验证或上传时保存					
Use accessibility features						
附加开发板管理器网址:https://mcu.hollek.com.tw/arduino/ht32/psokage_ht32_index.json						
在首选项中还有更多选项可以直接编辑						
C-\Users\wenjielu\AppData\Local\Arduino15\preferences.txt						
(只能在 Arduino 未运行时进行编辑)						
	好取	肖				

输入 Json 路径



 点选"工具→开发板: "Arduino UNO"→开发板管理器"。完成后跳出"开 发板管理器"视窗,搜寻"HT32"发现 Holtek HT32 Boards 安装视窗,请选 最新版本点选"安装"并等待安装完成。完成后点选"关闭"。

∞ 开发板管理器	×
类型 全部 ✓ HT32	
Holtek HT32 Boards (32-bits ARM Cortex-M0+/M3/M4)	^
by Holtek Semiconductor Inc. 版本 1.0.6 INSTALLED 这个组织会的开发框:	
BM53A367A (BMduino-UNO HT32F52367). Online Help	
More Info	
送择版本 🗸 安装 更新 删	除
	~
	关闭

安装 Holtek Library

ICP 模式设置(建议)

点选"工具→开发板→Holtek HT32 Boards→BM53A367A (BMduino-UNO HT32F52367)"完成初始化设定。



选择 BM53A367A



IAP 模式设置

步骤 1. 点选"工具 → 开发板 → Holtek HT32 Boards → BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"

🥺 sketch_apr06b	Arduino 1.8.13 — 🗆 🗙				
文件 編辑 项目 工	員 帮助		-		
	自动格式化	Ctrl+T			
	项目存储				
sketch_apr06b	修正编码并重新加载				
1 void	管理庫	Ctrl+Shift+I			
2 11	串口监视器	Ctrl+Shift+M			
2 11	串口绘图器	Ctrl+Shift+L			
3	MERION (MERININA FILMERIA LICATION				
4 }	Winton / Winter inter opdater				
5	开发板: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"	3	开发板管理器		
6 woid	Error Display: "Disable"	3	Arduino ARM (32-bits) Boards	>	
7 ((端口: "COM10"	3	Arduino AVR Boards	>	
/ //	取得开发板信息		Holtek HT32 Boards	2	ESK32-30501 (HT32F52352 Starter Kit)
8	编码码·"ICD (a-Link32 Lite/Dro)"				ESK32-30105 (HT32F12366 Starter Kit)
9}					BM53A367A (BMduino-UNO HT32F52367)
	2004 ST 6 18/17]	•	BM53A367A (BMduino-UNO HT32F52367 IAP Mode)
	~				
<	>				
1	BM53A387A (BMduino-UNO HT32F52387 IAP Mode) \pm COM10				

选择 BM53A367A IAP 模式

步骤 2. BM53A367A 出厂未烧录引导程序 (Bootloader),因此需先进行引导程序 的烧录。首先决定烧录引导程序的方法,点选"工具→编程器→ICP (e-Link32 Lite/Pro)"。



选择 ICP (e-Link32 Lite/Pro)

步骤 3. 进行引导程序烧录,点选"工具→烧录引导程序"。

🥯 sketch_apr06b A	rduino 1.8.13 — 🗆 🗙	
文件编辑项目 工具	帮助	
sketch_apr06k	自动格式化 项目存档 修正编码并重新加载	Ctrl+T
1 void 2 //	管理库 串口监视器 串口绘图器	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L
4	WiFi101 / WiFiNINA Firmware Updater	
5 6 void 7 //	开发板: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)" Error Display: "Disable" 靖口: "COM10" 取得开发板信息	:
8 9 }	编程器: "ICP (e-Link32 Lite/Pro)" 焼录引导程序	:
<	×	
1	BM53A387A (BMduino-UNO HT32F52387 IAP Mode)	

选择烧录引导程序

Arduino Library 参考

由于 BMduino-UNO BM53A367A 与 Arduino UNO R3 两者的硬件资源不同,因此在使用 Arduino IDE 内建的 Library 时,有一部分 Library 的参数或执行方式 会有不同,表列如下:

步骤 4. 完成引导程序烧录后可发现 D13 LED 开始连线闪烁,代表完成引导程序的烧录,完成 IAP 模式的初始化。



#	Library	BMduino-UNO BM53A367A	Arduino UNO R3		
1	Serial	数据位: 7~9位 接收缓存器: 255字节	数据位: 5~8 位 接收缓存器: 64 字节		
2	analogReference()	内部参考电压: 1.215V、2V、2.5V 或 2.7V	内部参考电压: 1.1V		
3	SPI	setClockDivider(4) → SPI SCK=15MHz	setClockDivider(4) → SPI SCK=4MHz		
4	tone()	最低频率 1Hz	最低频率 31Hz		
5	analogWrite()	PWM 频率: 1000Hz 引脚: D0~D13	PWM 频率: 490/980Hz 引脚: D3、D5、D6、 D9、D10、D11		
6	SoftwareSerial	TX: 230400bps, RX: 115200bps 应答延时时间: >120µs	TX/RX: 57600bps 应答延时时间: >15µs		
7	Servo	除能 analogWrite() on D23 (BMCOM2)	除能 analogWrite() on D9, D10		
8	MsTimer2	内建 Library,为了防止名称重复改 名为 MsTimer	第三方 Library		
9	attachInterrupt()	D2~D12、D22、D25、D29	D2, D3		

1. Serial.begin()

Syntax: Serail.begin(speed, config) 其中 config 的有效值为:

BMduino-UNO BM53A367A	Arduino UNO R3
SERIAL_7N1	SERIAL_5N1
SERIAL_8N1(默认)	SERIAL_6N1
SERIAL_9N1	SERIAL_7N1
SERIAL_7N2	SERIAL_8N1(默认)
SERIAL_8N2	SERIAL_5N2
SERIAL_9N2	SERIAL_6N2
SERIAL_7E1: 偶校验	SERIAL_7N2
SERIAL_8E1	SERIAL_8N2
SERIAL_9E1	SERIAL_5E1: 偶校验
SERIAL_7E2	SERIAL_6E1
SERIAL_8E2	SERIAL_7E1
SERIAL_9E2	SERIAL_8E1
SERIAL_701: 奇校验	SERIAL_5E2
SERIAL_801	SERIAL_6E2
SERIAL_901	SERIAL_7E2
SERIAL_702	SERIAL_8E2
SERIAL_802	SERIAL_501: 奇校验
SERIAL_902	SERIAL_601
	SERIAL_701
	SERIAL_801
	SERIAL_502
	SERIAL_602
	SERIAL_702
	SERIAL_802



2. <u>analogReference()</u>

Syntax: analogReference(type)

type 的有效值为:

BMduino-UNO BM53A367A	Arduino UNO R3
DEFAULT:默认 3.3V 模拟参考电压	DEFAULT: 默认 5V 模拟参考电压
INTERNAL1V215: 内部 1.215V 参考电压	INTERNAL: 内部 1.1V 参考电压
INTERNAL2V0: 内部 2V 参考电压	EXTERNAL: 施加在 AREF 引脚的
INTERNAL2V5:内部 2.5V 参考电压	电压 (仅 0~5V) 被用作参考电压
INTERNAL2V7:内部 2.7V 参考电压	
EXTERNAL: 施加在AREF引脚的电压(仅	
0~3.3V) 被用作参考电压	

3. <u>SPI.setClockDivider()</u>

Syntax: SPI.setClockDivider(divider)

setClockDivider 为除频的 API。因为工作频率不同,导致参数 divider 相同但 SCK 输出的频率不同, BM53A367A 工作频率 60MHz; Arduino UNO R3 工 作频率 16MHz。举例如下:

BMduino-UNO BM53A367A	Arduino UNO R3
SPI.setClockDivider(4) \rightarrow SCK =	$SPI.setClockDivider(4) \rightarrow SCK =$
60MHz / 4 = 15MHz	16MHz / 4 = 4MHz

4. tone()

Syntax: tone(pin, frequency)

tone(pin, frequency, duration)

BM53A367A 最低输出频率为 1Hz; Arduino UNO R3 最低输出频率 31Hz。

5. analogWrite()

Syntax: analogWrite(pin, value) BM53A367A 的 PWM 周期 1000Hz; Arduino UNO R3 的 PWM 周期 490Hz/980Hz。

6. SoftwareSerial

BM53A367A: TX 最高支持 230400bps, RX 最高 115200bps Arduino UNO R3: TX/RX 最高支持 57600bps

7. <u>Servo</u>

BM53A367A: D23 的 analogWrite() 输出 PWM 的功能被禁止, D23 属于 BMCOM2。

Arduino UNO R3: D9, D10 的 analogWrite() 输出 PWM 的功能被禁止。

8. <u>MsTimer2</u>

BM53A367A: 内建此 Library 名称改为 MsTimer, 可在文件 \rightarrow 示例 \rightarrow MsTimer 内找到相关范例。

Arduino UNO R3: 第三方提供 Library 需通过 Library Manager 下载。

9. attachInterrupt()

Syntax: attachInterrupt(interruptNum, FuncPtr callback, mode) BM53A367A: D2~D12、D22、D25、D29 可作为外部中断使用。 Arduino UNO R3: D2、D3 作为外部中断使用。



范例

硬件准备

需准备开发板、Type-C USB 传输线、电脑。将开发板的 e-link32 Lite 通过 USB 线与电脑连接,此时 PWR LED 被点亮,还需等 e-link32 Lite 被列举完成,列举 完成 RDY LED 会被点亮此时硬件准备完成。

范例代码

请执行 Blink 范例,详细请参考下方链接。完成范例烧录后可发现 D13 LED 每 秒翻转一次。

https://docs.arduino.cc/built-in-examples/basics/Blink

Keil IDE 软件

Keil IDE 下载及安装

打开 Keil 官方网站 (<u>https://www.keil.com/demo/eval/arm.htm</u>) 下载 MDK-ARM 并 且执行安装。详细安装步骤请参考下方链接:

https://www.holtek.com.cn/documents/10179/6393521/sim_HT32_Keil-QuickStartv110.pdf

IDE 设置

1. 下载 HT32 开发资源:通过下方链接下载最新 HT32F5 Series (Cortex[®]-M0+)。 内包含了 HT32 开发所需的全部资源,下载完成后请解压缩。

https://mcu.holtek.com.cn/ht32/resource/

- 2. 安装 HT32 Packs, 执行"\HT32_M0p_vxxxxxx\Tools\Holtek.HT32_DFP.xx.xx. xx.pack"。
- 3. 安装 VCP 驱动,执行"\HT32_M0p_vxxxxxx\Tools\HT32_VCP_Driver_vxxx.exe"。
- 4. 将 HT32 FW Lib 解压缩,路径 "\Firmware_Library\HT32_STD_5xxxx_FWLib_ Vx.x.x_xxxx.zip"。

运行范例

- 1. 开启 HT32 FW Lib 内的 Keil 工程, 路径 "\Firmware_Library\HT32_STD_5xxxx_ FWLib_Vx.x.x_xxxx.zip\project_template\IP\Example\MDK_ARMv5\ Project_53a367a.uvprojx"。
- 2. 通过 Keil IDE 进行编译与烧录,详细请参考"烧录方法"章节。
- 3. 按压 RESET 按钮,观察 D13 LED 快速闪烁 5 次,完成测试。
 - 补充:此范例也展示 Serial (115200, 8, N, 1)的功能,请通过终端机软件(例如 Tera Term,详细内容请参考官方网站 <u>https://ttssh2.osdn.jp/index.html.en</u>), 开启 COM 端口观察提示信息。



问题排查

串口未出现在端口菜单中

代表 e-Link32 Lite 的 VCP 连线失败,请依照下方步骤除错: 1. 请确认 e-Link Lite 的 USB 是否连接 PC,以及 RDY LED 是否点亮。



- 2. 若 RDY LED 未点亮请重新插拔 e-Link Lite 的 USB 并选择 PC 上其他 USB 接口连接。
- 3.10 秒后若 RDY LED 未被点亮请重新启动 PC。若是 RDY 点亮就可在 Arduino COM 端口菜单内发现 BMduino 的 COM 端口。



✓ 賞 端口 (COM 和 LPT) 員 USB 序列裝置 (COM3)

4. RDY LED 已点亮但 Arduino COM 端口菜单未显示 BMduino 的 COM 端口, 请安装 VCP 驱动程序(仅限 Windows 电脑)。

参考 Keil IDE 软件里 IDE 设置的第三点安装 VCP 驱动。



范例上传失败或冻结

- 1. 错误信息 "This computer can't enumerate any e-Link32 Pro/Lite. Please make sure this computer has indeed connected to e-Link32 Pro/Lite."代表 e-Link32 Lite 的 CMSIS-DAP (烧录接口)连线失败,请确认 e-Link Lite 的 USB 是否连 接 PC 以及 RDY LED 是否点亮,若 RDY LED 未点亮请重新插拔 e-Link Lite 的 USB 并选择 PC 上其他 USB 接口连接,10 秒后若 RDY LED 未被点亮请 重新启动 PC。
- 2. 错误信息 "The corresponding e-Link32 Pro/Lite can't be found by the target ID/SN, which can be in INI file or specified by users."代表未找到你指定的目标板,请点选"工具→Target Board:→Board1",再进行"Upload"。

🥺 sk	ketch	_apr06	56 A	rduino 1.8.13 — 🗆	×			
文件	编辑	项目	工具	帮助				
	Ð			自动格式化	Ctrl+T			
				项目存档				
ske	etch_a	apr06k		修正编码并重新加载				
1	vo	id		管理库	Ctrl+Shi	ft+I		
2		11		串口监视器	Ctrl+Shi	ft+M		
2		//		串口绘图器	Ctrl+Shi	ft+L		
4	3			WiFi101 / WiFiNINA Firmware Updater				
5	1			开发板: "BM53A367A (BMduino-UNO HT32F52367))"	>		
2				Upload Method: "e-Link32 Pro (Mass Erase)"		>		
6	vo	ıd		Error Display: "Disable"		>		
7		//		Target Board: "Board1"		2	•	
8				端口: "COM10"		;		
9	}			取得开发板信息				
_	<i>'</i>			编程器: "ICP (e-Link32 Lite/Pro)"		2		1
				烧录引导程序				
							1	
					~			
	<				>			

选择 Board1



 若上传失败信息并非上述的状况,请通过"Mass Erase"将 MCU 清空后再 烧录程序。具体做法请点选"工具 → Upload Method: → e-Link32 Pro (Mass Erase)",再进行"Upload"。



Mass Erase



■ 尺寸规格



尺寸	信息
----	----

单位 编号	mm	inch	
А	2.54	0.1	
В	4.064	0.16	
С	3.556	0.14	
D	44.704	1.76	
Е	48.26	1.9	
F	27.94	1.1	
G	36.56	1.4	
Н	5.08	0.2	
Ι	2.54 0.1		
J	13.97	0.55	
K	5.08	0.2	
L	93.221	3.67	
М	53.35	2.1	
N	N 2.667 0.105		
0	O 22.86 0.9		
D1	3.2512	0.128	

尺寸列表





元件尺寸 – 高度信息

尺寸	ĸ		宽		高	
编号	mm	inch	mm	inch	mm	inch
a	6.2	0.236	6.2	0.236	2	0.078
b	8	0.314	5.15	0.203	3	0.118
с	8	0.314	5.15	0.203	3	0.118
d	14.2	0.559	9	0.354	11	0.433
e	5	0.197	5	0.197	4	0.157
f	6.3	0.248	6.3	0.248	7.7	0.303
g	48	1.89	2.5	0.098	8	0.315
h	38	1.496	2.5	0.098	8	0.315
i	4.53	0.178	7.5	0.295	3.1	0.122
j	4	0.155	2	0.80	4	0.155
k	13.7	0.540	7.4	0.292	4	0.157
1	8.5	0.334	12.776	0.503	4	0.157

默认高度: 11mm / 0.433 inch

元件尺寸 – 高度列表



Copyright[©] 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版时倍创已针对所载信息为合理注意,但不保证信息准确无误。文中提到的信息仅是提供 作为参考,且可能被更新取代。倍创不担保任何明示、默示或法定的,包括但不限于适合商品化、 令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。倍创就文中提到 的信息及该信息之应用,不承担任何法律责任。此外,倍创并不推荐将倍创的产品使用在会由于故 障或其他原因而可能会对人身安全造成危害的地方。倍创特此声明,不授权将产品使用于救生、维 生或安全关键零部件。在救生/维生或安全应用中使用倍创产品的风险完全由买方承担,如因该等 使用导致倍创遭受损害、索赔、诉讼或产生费用,买方同意出面进行辩护、赔偿并使倍创免受损害。 倍创(及其授权方,如适用)拥有本文件所提供信息(包括但不限于内容、数据、示例、材料、图形、 商标)的知识产权,且该信息受著作权法和其他知识产权法的保护。倍创在此并未明示或暗示授予 任何知识产权。倍创拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息,请与我 们联系。