BMduino-UNO Development Board

BM53A367A User Guide

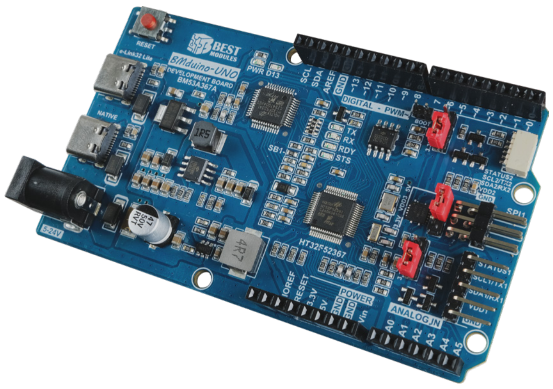
Revision: V1.20   Date: June 13, 2024

www.bestmodulescorp.com

Contents

Introduction

The BMduino-UNO BM53A367A is a development board implemented using a Holtek 32-bit MCU, the HT32F52367, which is especially designed to be pin-compatible with the Arduino UNO R3 development board. It can support both Arduino IDE and Keil IDE development platforms to assist beginners to learn programming more easily. The BM53A367A uses the HT32F52367, which is based on a 3.3V series Arm® Cortex®-M0+ core, as the master MCU. It also supports various power supply modes and has additional common communication interfaces such as I^{2}C, SPI, UART and USB.

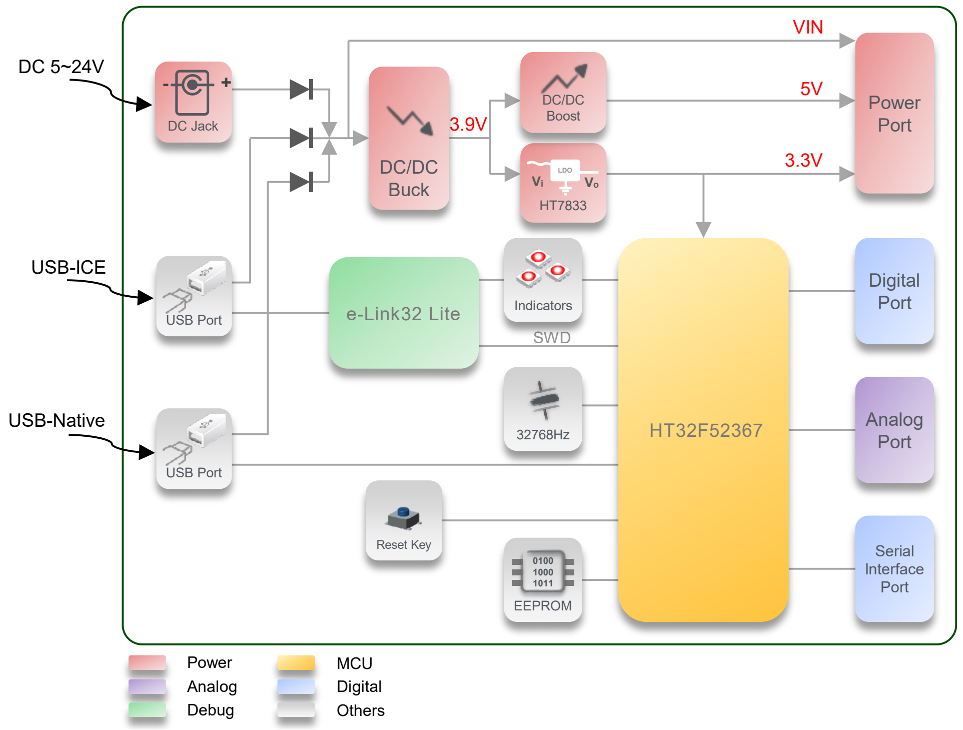


Features

- Master MCU: HT32F52367 – 64-pin LQFP
  - Cortex®-M0+, 60MHz
  - Flash Memory: 256KB
  - SRAM: 32KB
- 31 digital I/O pins, 17 of which can be used as PWM outputs
- 12-bit resolution A/D converter with 7 analog inputs
- 14 external interrupts
- EEPROM: 4KB
- Communication interfaces: UART, SPI, I^{2}C, USB
- Power supply inputs: USB interface×2, DC interface, external Vin
  - e-Link32 Lite USB: Type-C USB interface
  - Native USB: Type-C USB interface, supports BC 1.2 and QC 2.0, the preset boost for the quick charging is 12V
  - DC interface: Circular connector to interface to an external power supply (e.g. transformer), voltage range: DC 5~24V
  - External Vin: Vin pin, voltage range: DC 5~24V
- Debug tool: e-Link32 Lite (ICE), implements program debugging using the e-Link32 Lite circuit on the development board
- Programing modes:
  - In-Circuit Programing – ICP, implements programing using the e-Link32 Lite circuit on the development board
  - In Application Programming – IAP, implements programing using a COM port

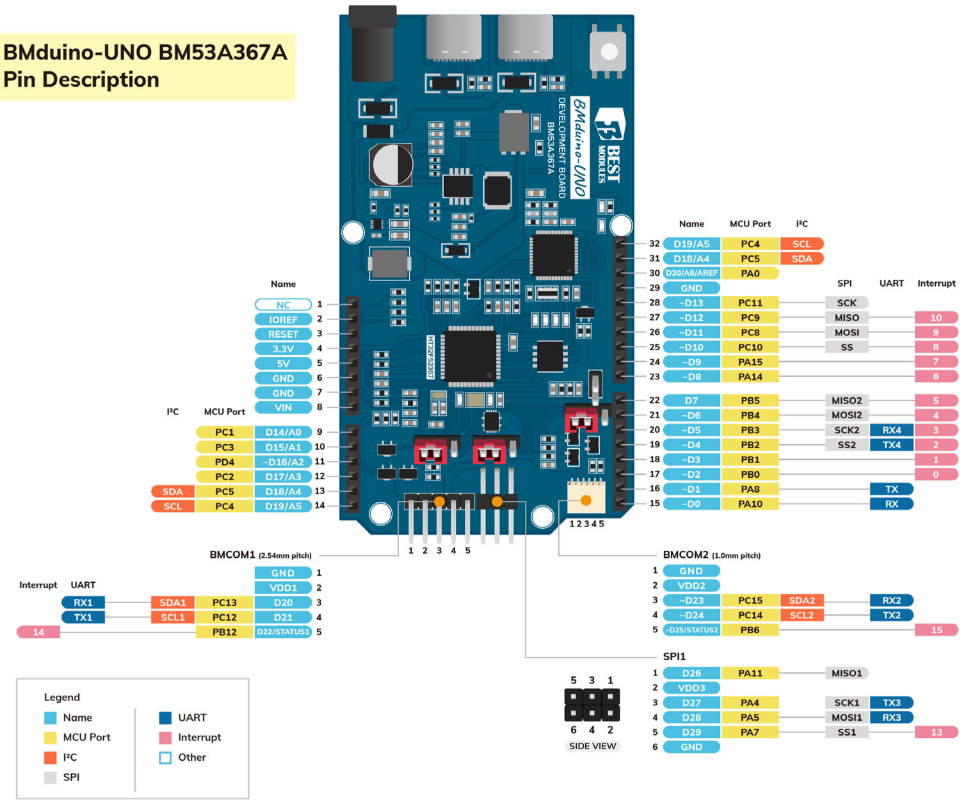
- Development environment: supports Arduino IDE and Keil IDE development platforms
- Interface resources: BMCOM1 (lead pitch: 2.54mm, optional 3.3V/5V), BMCOM2 (lead pitch: 1.0mm, 3.3V/5V optional), SPI1 (lead pitch: 2.54mm, optional 3.3V/5V)
- Development board size: 53.4mm×93.221mm×14.2mm, pin-compatible with the Arduino UNO R3 development board
- Development board weight: 28.1g (Net)

Block Diagram



Pin Description

BMduino-UNO BM53A367A Pin Description

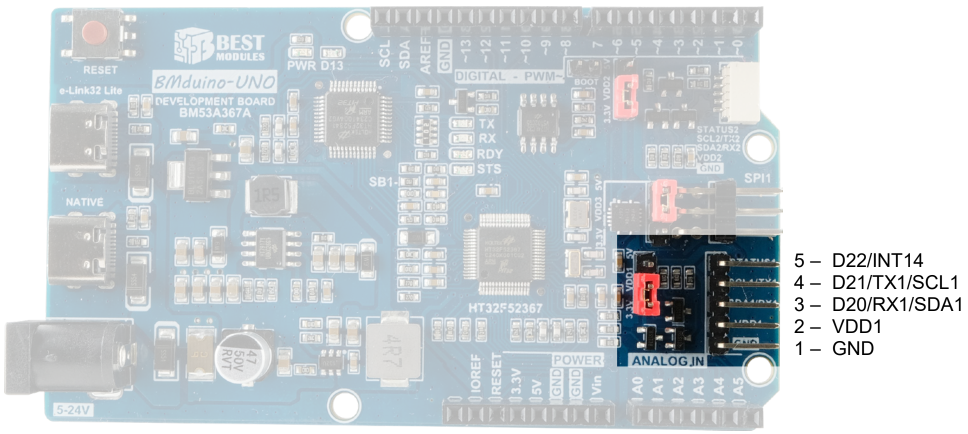


Pin No.	Function	Description
1	NC	Not connected
2	IOREF	I/O logic reference voltage, +3.3V
3	RESET	Reset pin
4	+3V3	+3.3V power supply output
5	+5V	+5V power supply output
6	GND	Power ground
7	GND	Power ground
8	Vin	Power supply input 5~24V
9	A0/D14	Analog input pin A0 / Digital pin D14
10	A1/D15	Analog input pin A1 / Digital pin D15
11	A2/~D16	Analog input pin A2 / Digital pin D16 with PWM function
12	A3/D17	Analog input pin A3 / Digital pin D17
13	A4/D18/SDA	Analog input pin A4 / Digital pin D18 / I²C0 interface SDA pin
14	A5/D19/SCL	Analog input pin A5 / Digital pin D19 / I²C0 interface SCL pin
15	~D0/RX	Digital pin D0 with PWM function / UART0 receive pin
16	~D1/TX	Digital pin D1 with PWM function / UART0 transmit pin
17	~D2/INT0	Digital pin D2 with PWM function / External interrupt pin INT0
18	~D3/INT1	Digital pin D3 with PWM function / External interrupt pin INT1
19	~D4/TX4/SS2/INT2	Digital pin D4 with PWM function / UART4 transmit pin / SPI2 interface SS2 pin / External interrupt pin INT2
20	~D5/RX4/SCK2/ INT3	Digital pin D5 with PWM function / UART4 receive pin / SPI2 interface SCK2 pin / External interrupt pin INT3

Pin No.	Function	Description
21	~D6/MOSI2/INT4	Digital pin D6 with PWM function / SPI2 interface MOSI2 pin / External interrupt pin INT4
22	D7/MISO2/INT5	Digital pin D7 / SPI2 interface MISO2 pin / External interrupt pin INT5
23	~D8/INT6	Digital pin D8 with PWM function / External interrupt pin INT6
24	~D9/INT7	Digital pin D9 with PWM function / External interrupt pin INT7
25	~D10/SS/INT8	Digital pin D10 with PWM function / SPI0 interface SS pin / External interrupt pin INT8
26	~D11/MOSI/INT9	Digital pin D11 with PWM function / SPI0 interface MOSI pin / External interrupt pin INT9
27	~D12/MISO/INT10	Digital pin D12 with PWM function / SPI0 interface MISO pin / External interrupt pin INT10
28	~D13/SCK	Digital pin D13 with PWM function / SPI0 interface SCK pin
29	GND	Power ground
30	AREF/A6/D30	Analog reference voltage / Analog input pin A6 / Digital pin D30
31	A4/D18/SDA	Analog input pin A4 / Digital pin D18 / I²C0 interface SDA pin
32	A5/D19/SCL	Analog input pin A5 / Digital pin D19 / I²C0 interface SCL pin

Pin 13 is pin-shared with Pin 31 and Pin 14 is pin-shared with Pin 32.

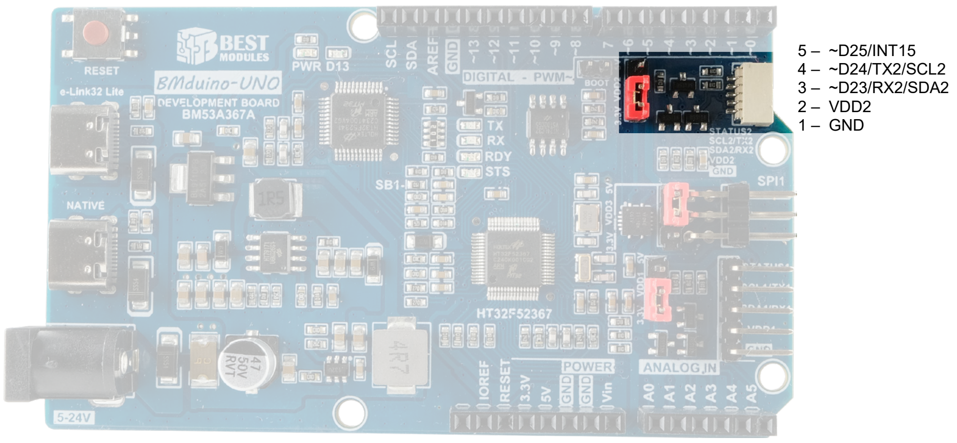
BMCOM1: this can be used as an I²C or UART interface (I²C1, Serial1)



Pin No.	Function	Description
1	GND	Power ground
2	VDD1	3.3V or 5V power supply output, determined by the adjacent jumper
3	D20/RX1/SDA1	Digital pin D20 / UART1 receive pin / I²C1 interface SDA1 pin
4	D21/TX1/SCL1	Digital pin D21 / UART1 transmit pin / I²C1 interface SCL1 pin
5	D22/INT14	Digital pin D22 / External interrupt pin INT14

The voltage level of Pin 3, Pin 4 and Pin 5 can be set to either 3.3V or 5V by the adjacent jumper.

BMCOM2: this can be used as an I²C or UART interface (I²C2, Serial2)

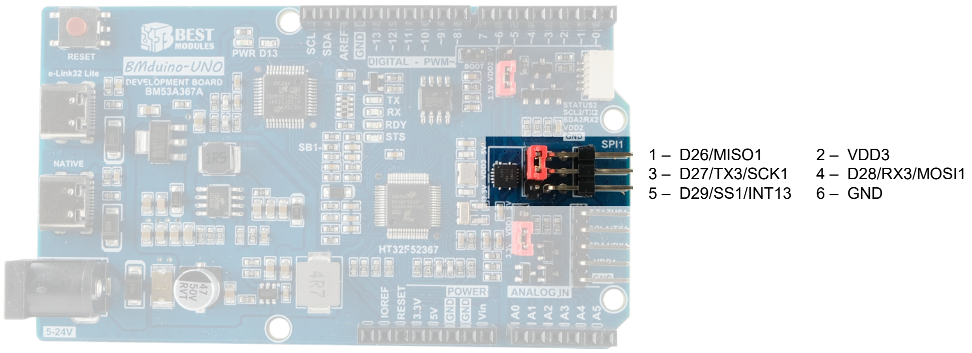


Pin No.	Function	Description
1	GND	Power ground
2	VDD2	3.3V or 5V power supply output, determined by the adjacent jumper
3	~D23/RX2/SDA2	Digital pin D23 with PWM function / UART2 receive pin / I²C2 interface SDA2 pin
4	~D24/TX2/SCL2	Digital pin D24 with PWM function / UART2 transmit pin / I²C2 interface SCL2 pin
5	~D25/INT15	Digital pin D25 with PWM function / External interrupt pin INT15

The voltage level of Pin 3, Pin 4 and Pin 5 can be set to either 3.3V or 5V by the adjacent jumper.

Note: I²C1 and I²C2 share the same physical I²C interface in the MCU. Therefore BMCOM1 I²C and BMCOM2 I²C cannot be used at the same time.

SPI1: it can be used as an SPI or UART interface (SPI1, Serial3)



Pin No.	Function	Description
1	D26/MISO1	Digital pin D26 / SPI1 interface MISO1 pin
2	VDD3	3.3V or 5V power supply output, determined by the adjacent jumper
3	D27/TX3/SCK1	Digital pin D27 / UART3 transmit pin / SPI1 interface SCK1 pin
4	D28/RX3/MOSI1	Digital pin D28 / UART3 receive pin / SPI1 interface MOSI1 pin
5	D29/SS1/INT13	Digital pin D29 / SPI1 interface SS1 pin / External interrupt pin INT13
6	GND	Power ground

The voltage level of Pin 1, Pin 3, Pin 4 and Pin 5 can be set to either 3.3V or 5V by the adjacent jumper.

Technical Specifications

Absolute Maximum Ratings

- Supply power for the MCU using the e-Link32 Lite USB interface or the Native USB interface.

Note: The USB power supply cannot be less than 4.5V.

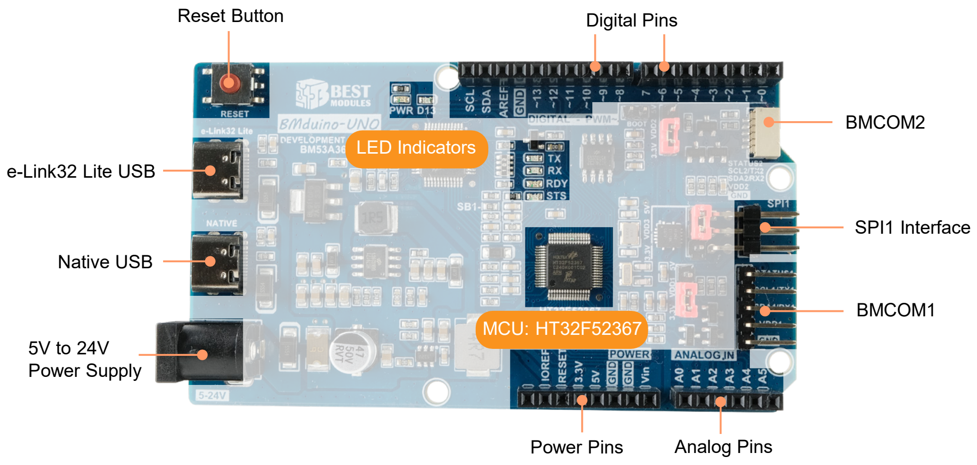
- A voltage of 5~24V is input through the DC Jack, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.
- A voltage of 5~24V is input through Vin, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{inMAX}	Input Voltage from Vin Pin	24	—	26	V
V_{USBMAX}	Input Voltage from USB Connector	4.5	5.5	—	V

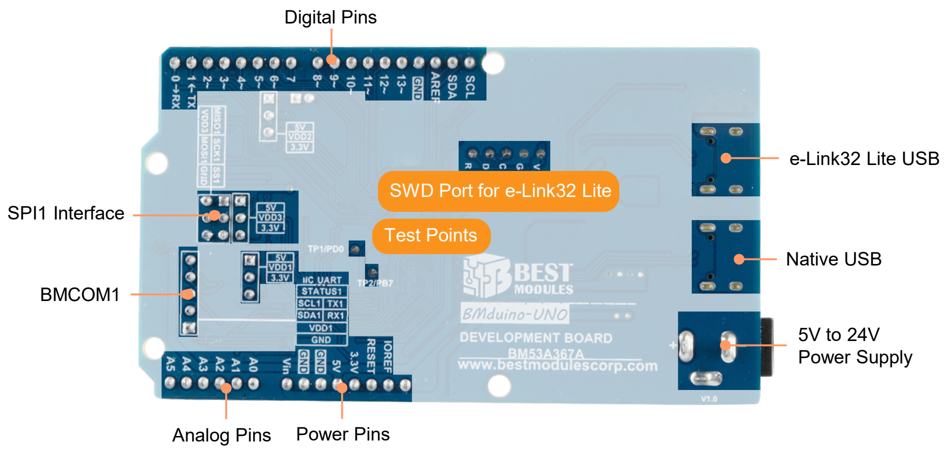
Recommended Operation Conditions

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{in}	Input Voltage	—	5	—	24	V
I_{OUT}	5V Output Current	V_{in}>11V	—	—	1.3	A
		V_{in}=4.6V	—	—	800	mA
	3.3V Output Current	—	—	—	500	mA
	I/O Output Current	—	—	—	16	mA

Hardware Overview



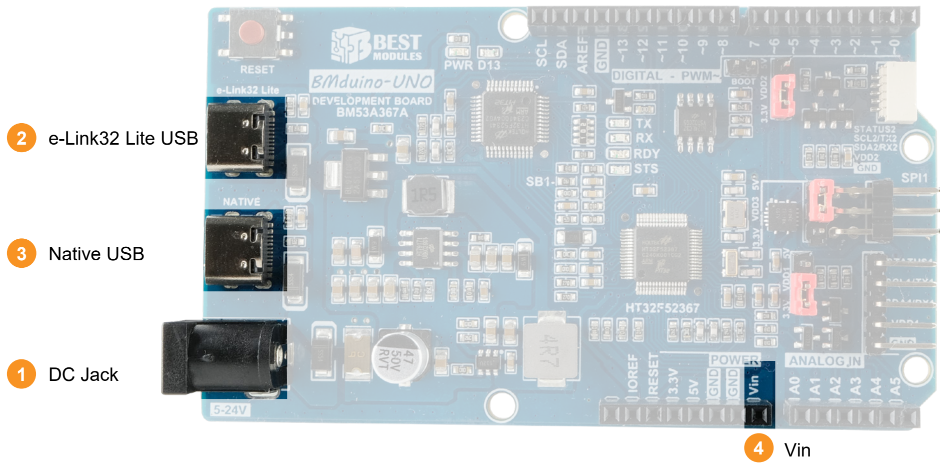
PCBA Front View



PCBA Back View

Power Supply

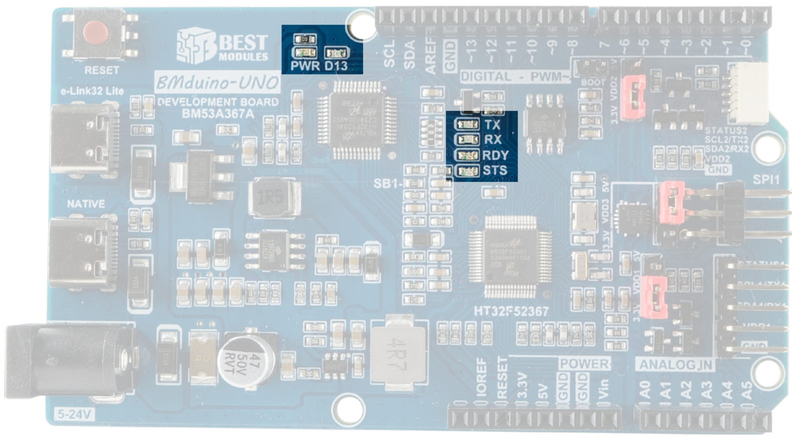
- A voltage of 5V is input through the e-Link32 Lite USB interface, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.
- A voltage of 5V or 12V (boosted to 12V when connected to a quick charger for QC 2.0) is input through the Native USB interface, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.
- A voltage of 5~24V is input through the DC Jack, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.
- A voltage of 5~24V is input through the Vin, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.



Power Interface Diagram

Note: When the DC Jack, Native USB or e-Link32 Lite USB is used together with Vin at the same time, the Vin input voltage must be greater than or equal to the DC Jack, Native USB or e-Link32 Lite USB in order to prevent reverse voltage problems.

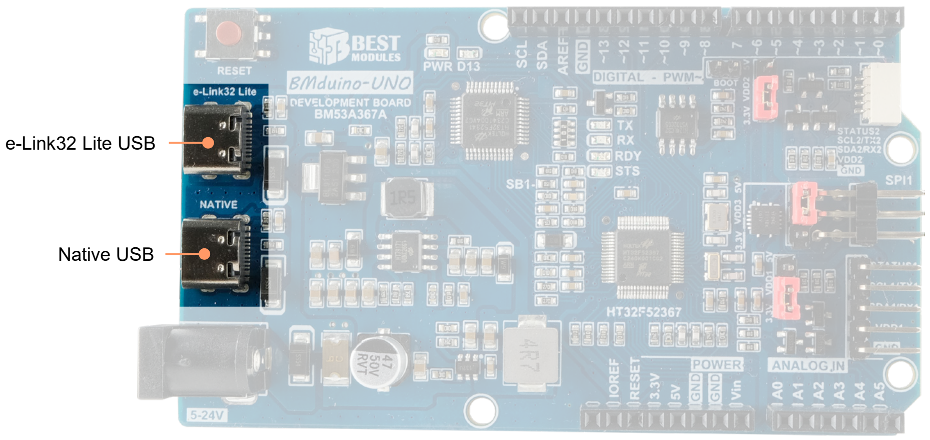
LED Indicators



LED Configuration Diagram

- PWR: Power LED.
- D13: Connected to Digital pin 13, used for example programs to observe program status.
- TX and RX: Indicate the UART TX or RX bus transmission status, flashing during data transmission.
- RDY: Indicates the e-Link32 Lite USB connection status and turns on the LED when connection with the computer has completed.
- STS: Indicates the e-Link32 Lite programming status and blinks during the programing process.

USB Interface Circuit



The BMduino-UNO BM53A36A development board has two USB ports, e-Link32 Lite and NATIVE.

- e-Link32 Lite main functions:
  1. When the Arduino IDE or Keil IDE is used for program development, connect this USB port to the computer's USB port to implements the master MCU programming.
  2. Supports the Virtual COM Port, VCP, to connect to the master MCU TX and RX pins.
  3. Provides the program debugging function under the Keil IDE development environment, such as for breakpoint setting.

The BMduino-UNO BM53A367A is preset to use the e-Link32 Lite circuit for programming, without requiring the TX and RX pins.

- NATIVE main functions:
  1. Power supply interface
     When the BMduino-UNO BM53A36A development board is powered by the QC 2.0 charger through the Native USB, the preset boost for the Native USB interface is 12V.
  2. USB communication
     The USB interface is preset to VCP function, which can implement communication using the SerialUSB of the Arduino API. When the BMduino-UNO BM53A36A development board is used to develop a product with a USB function such as a mouse or keyboard, this USB port is the USB port of the product. If it is required to use a keyboard or mouse, the header file such as #include <Mouse.h> must be added to Sketch.

Obj	SerialUSB (USB CDC)	Keyboard (USB HID)	Mouse (USB HID)
Pins	Native USB Port only	Native USB Port only	Native USB Port only

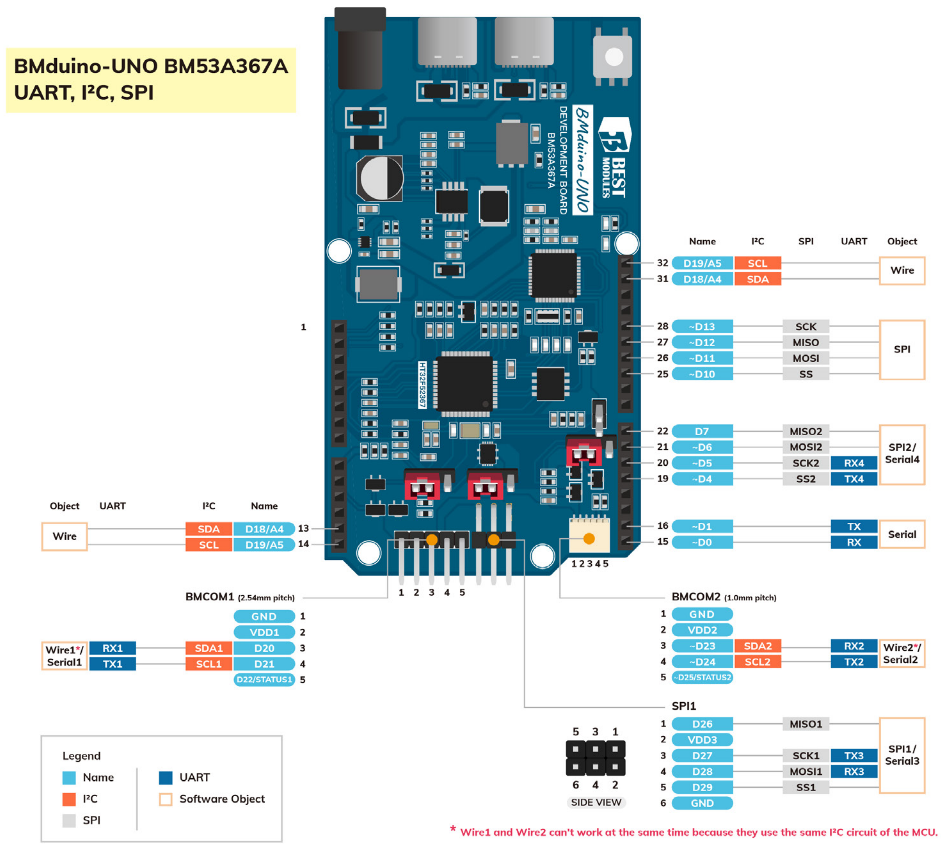
For example, when the Bmduino-UNO is connected to a PC using the Native USB port as a USB keyboard or mouse, declare the Keyboard.h and Mouse.h files and then use the keyboard and mouse objects in the program.

Example program:

```
#include "Mouse.h"
#include "Keyboard.h"

// set pin numbers for the five buttons:
const int upButton = 2;
const int downButton = 3;
const int leftButton = 4;
const int rightButton = 5;
const int mouseButton = 6;

int range = 5;            // output range of X or Y movement; affects
                          // movement speed
int responseDelay = 20;   // response delay of the mouse, in ms

void setup() {
  // initialise the buttons' inputs:
  pinMode(upButton, INPUT);
  pinMode(downButton, INPUT);
  pinMode(leftButton, INPUT);
  pinMode(rightButton, INPUT);
  pinMode(mouseButton, INPUT);

  // initialise mouse control:
  Mouse.begin();
  // initialise keyboard control:
  Keyboard.begin();
  // initialise SerialUSB
  SerialUSB.begin(9600);
}

void loop() {
  // read the buttons:
  int upState = digitalRead(upButton);
  int downState = digitalRead(downButton);
  int rightState = digitalRead(rightButton);
  int leftState = digitalRead(leftButton);
  int clickState = digitalRead(mouseButton);

  // calculate the movement distance based on the button states:
  int xDistance = (leftState - rightState) * range;
  int yDistance = (upState - downState) * range;

  // if X or Y is non-zero, move:
  if ((xDistance != 0) || (yDistance != 0)) {
  // Move the mouse cursor.
    Mouse.move(xDistance, yDistance, 0);

    // The keyboard outputs the coordinates of the mouse.
    Keyboard.print("Move: ");
    Keyboard.print(xDistance);
    Keyboard.print(',');
    Keyboard.println(yDistance);
```

```
      Keyboard.flush();

      // The SerialUSB outputs the coordinates of the mouse.
      SerialUSB.print("Move: ");
      SerialUSB.print(xDistance);
      SerialUSB.print(',');

      SerialUSB.println(yDistance);
      SerialUSB.flush();
    }

    // if the mouse button is pressed:
    if (clickState == HIGH) {
      // if the mouse is not pressed, press it:
      if (!Mouse.isPressed(MOUSE_LEFT)) {
        Mouse.press(MOUSE_LEFT);
        // If Keyboard println is uncommented, Mouse cursor
        // changes will cause the string to be overwritten.
        // Keyboard.println("Press MOUSE_LEFT");
        // Keyboard.flush();
        SerialUSB.println("Press MOUSE_LEFT");
        SerialUSB.flush();
      }
    }
    // else the mouse button is not pressed:
    else {
      // if the mouse is pressed, release it:
      if (Mouse.isPressed(MOUSE_LEFT)) {
        Mouse.release(MOUSE_LEFT);
        // If Keyboard println is uncommented, Mouse cursor
        // changes will cause the string to be overwritten.
        // Keyboard.println("Release MOUSE_LEFT");
        // Keyboard.flush();
        SerialUSB.println("Release MOUSE_LEFT");
        SerialUSB.flush();
      }
    }

    // a delay so the mouse doesn't move too fast:
    delay(responseDelay);
  }
```

UART, I²C, SPI



* Wire1 and Wire2 can't work at the same time because they use the same I²C circuit of the MCU.

- Supports five UART interfaces, whose object names are Serial, Serial1 to Serial4.

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

For example, to select the BMCOM1 UART interface, use the Serial1 object in the program.

Example program:

```
void setup() {
   // initialise serial communication:
   Serial1.begin(9600);
}

void loop() {
   // send the value of analog input 0:
   Serial1.println(analogRead(A0));
   // wait a bit for the analog-to-digital converter to stabilise after
   //the last read:
   delay(20);
}
```

When the Arduino Serial Library is used for the BMduino-UNO BM53A367A development board, the number of data bits can be from 7 to 9, while for the Arduino IDE the preset number is 5~8.

- Supports three I²C interfaces, whose object names are Wire, Wire1 and Wire2. Here Wire1 and Wire2 share the same physical I²C. Only one is permitted to implement communication at any single time.

Obj	Wire	Wire1	Wire2
Pins	SDA / SCL or A4 / A5 (shared I/O)	SDA1 (D20) / SCL1 (D21) (BMCOM1)	SDA2 (D23) / SCL2 (D24) (BMCOM2)

Note: I²C1 and I²C2 share the same physical I²C in the MCU. Therefore, the corresponding Wire1 and Wire2 are not able to operate at the same time. Only one of either Wire1 or Wire2 can be used. If both Wire1 and Wire2 exist in a program, refer to the following operations:

1. Use Wire1.begin() or Wire2.begin() for initialisation.
2. To switch the current Wire1 or Wire2, run Wire1.end() or Wire2.end() to finish the current Wire1 or Wire2.
3. Run Wire1.begin() or Wire2.begin() to initialise another Wire1 or Wire2.

If both Wire1 and Wire2 have been initialised, the one that is initialised first is the one that will be effective.

For example, to select the BMCOM1 I²C interface, declare Wire.h and use the Wire1 object in the program.

Example program:

```
#include <Wire.h>

void setup()
{
  Wire1.begin();        // join I²C bus (address optional for master)
  Serial.begin(9600);   // start serial for output
}

void loop()
{
  Wire1.requestFrom(2, 6);     // request 6 bytes from slave device #2

  while(Wire1.available())     // slave may send less than requested
  {
    char c = Wire1.read();     // receive a byte as character
    Serial.print(c);           // print the character
  }

  delay(500);
}
```

- Supports three SPI interfaces.

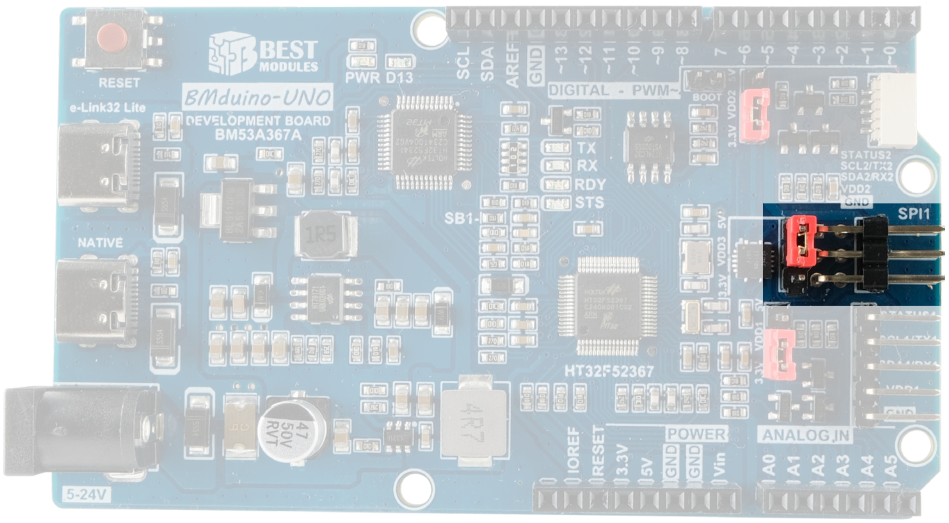
Obj	SPI	SPI1	SPI2
Pins	SS (D10) / MOSI (D11) / MISO (D12) / SCK (D13)	SS1 (D29) / MOSI1 (D28) / MISO1 (D27) / SCK1 (D26) (SPI1)	SS2 (D4) / MOSI2 (D6) / MISO2 (D7) / SCK2 (D5)

For example, to select the SPI1 interface with a 6-pin header, declare SPI.h and use the SPI1 object in the program.

Example program:

```
// include the SPI library:
#include <SPI.h>

// set pin 10 as the slave select for the digital port:
const int slaveSelectPin = 10;

void setup() {
  // set the slaveSelectPin as an output:
  pinMode(slaveSelectPin, OUTPUT);
  // initialise SPI1:
  SPI1.begin();
}

void loop() {
  // go through the six channels of the digital port:
  for (int channel = 0; channel < 6; channel++) {
    // change the resistance on this channel from min to max:
    for (int level = 0; level < 255; level++) {
      digitalPotWrite(channel, level);
      delay(10);
    }
    // wait a second at the top:
    delay(100);
    // change the resistance on this channel from max to min:
    for (int level = 0; level < 255; level++) {
      digitalPotWrite(channel, 255 - level);
      delay(10);
    }
  }
}

void digitalPotWrite(int address, int value) {
  // take the SS pin low to select the chip:
  digitalWrite(slaveSelectPin, LOW);
  // send in the address and value via SPI1:
  SPI1.transfer(address);
  SPI1.transfer(value);
  // take the SS pin high to de-select the chip:
  digitalWrite(slaveSelectPin, HIGH);
}
```

6-pin header on development board (3-pin × 2 rows, 90 degrees):

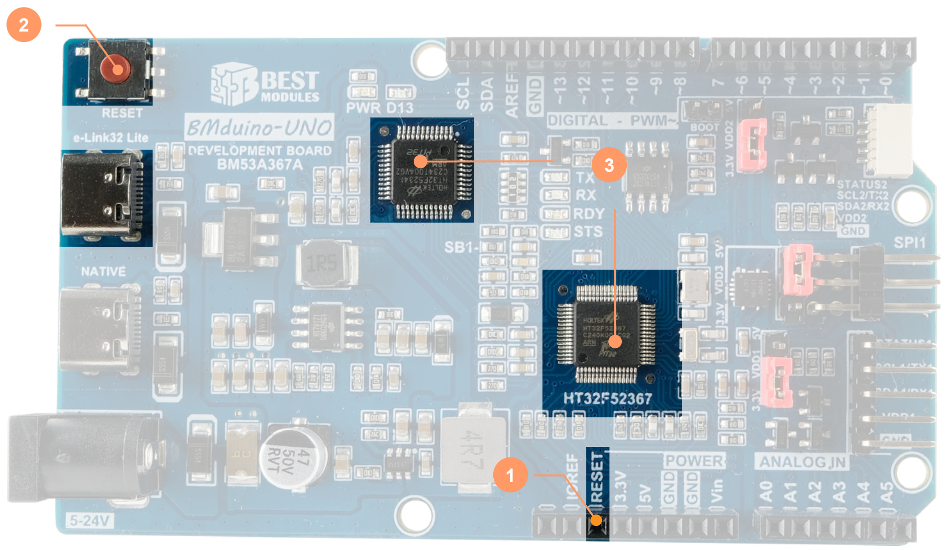


In the Arduino UNO R3 development board, this 6-pin header is called ICSP and its pins are pin-shared with Pin 11, Pin 12 and Pin 13 of the development board.

In the BMduino-UNO BM53A367A development board, this 6-pin header is used for SPI1 and does not share pins with Pin 11, Pin 12 and Pin13 of the development board. It is an independent SPI interface.

The Holtek 32-bit MCU uses a Cortex®-M0+ architecture and provides an SWD interface for programming instead of using the SPI interface. The SWD description can be obtained from the following link: https://developer.arm.com/documentation/100956/0529/Arm-DSTREAM-Target-Interface-Connections/Signal-descriptions/Serial-Wire-Debug?lang=en

Reset Circuit



- Set Pin 3 (RESET) to a low level for 1ms to trigger a system reset.
- Use the RESET button to reset the MCU.

- Connect the RESET circuit to the e-Link32 Lite (VCP) DTR to reset the MCU by enabling a COM port.

BM53A367A vs Arduino UNO R3

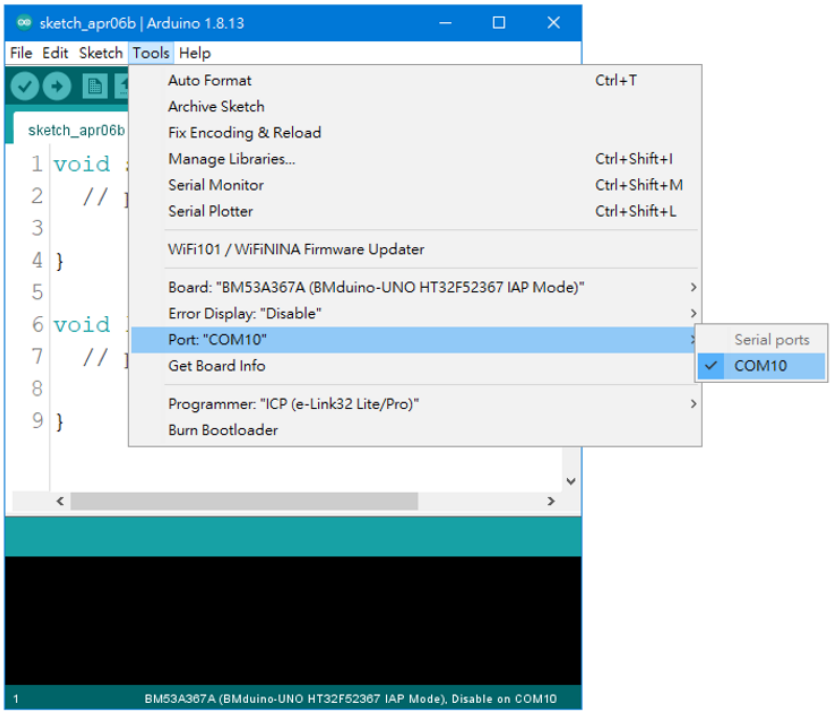
Board	BMduino-UNO BM53A367A	Arduino UNO R3
Core	Cortex M0+, 60MHz	AVR 8-bit, 16MHz
Flash / EEPROM / SRAM	256KB / 4KB / 32KB	32KB / 1KB / 2KB
Operating Voltage	3.3V	5V
Programming Mode	ICP/IAP (Bootloader)	IAP (Bootloader)
Programming Interface	SWD (Target Board Menu) / UART (Port Menu)	UART (Port Menu)
Development Environment	Arduino, Keil	Arduino
I/O Drive Current	16mA	20mA

Programming

The BM53A367A programming methods in each IDE are summarised as follows.

1. Arduino:

- ICP mode (default mode): Click on the "Upload" icon below the "Sketch" menu, and the Sketch will compile and execute programming. The programmed results can be observed in the bottom status window. For more details, refer to the Arduino official website: https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics.
- IAP mode: This method is the same as the ICP mode. However it is necessary to select the COM Port corresponding to the board using "Tools → Port" (same method as Arduino UNO R3) before "Upload", as shown in the following diagram.


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_apr06b
void
//
}
void
//
}
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"
Error Display: "Disable"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
Serial ports
COM10
1
BM53A367A (BMduino-UNO HT32F52367 IAP Mode), Disable on COM10

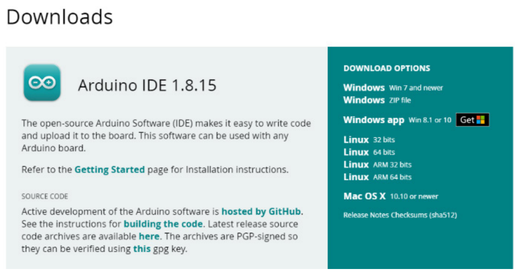
COM Port Menu

2. Keil: Click the "Build" icon from the IDE menu for program compilation, after the completion click the "Download" icon to execute programming. The compiled or programmed results can be observed in the lower window. For more details, refer to the Keil official website: https://www2.keil.com/mdk5/learn.

Arduino IDE Software

Arduino IDE Download and Installation

Open the Arduino official website (http://www.arduino.cc/en/Main/Software) to download the corresponding Arduino IDE software according to the computer operation system and execute the installation.


Downloads

Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the Getting Started page for Installation instructions.

SOURCE CODE

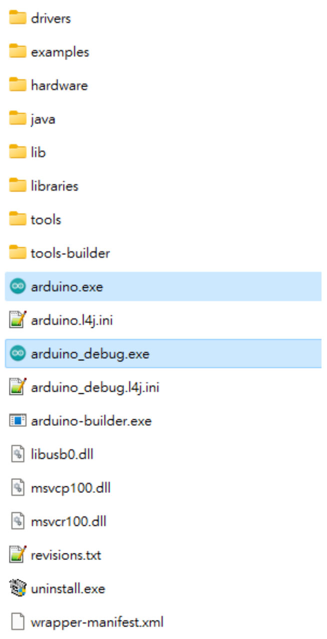
Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer
Release Notes Checksums (sha512)

Arduino IDE Software Download

After the installation, two execution files named Arduino.exe and Arduino_debug.exe in the folder can be found, both of which can open the program development environment. The difference is that Arduino_debug.exe can open the debugging window. When the program is executed or compiled, corresponding information will be displayed on the debug window, so that users can locate where the problem is.

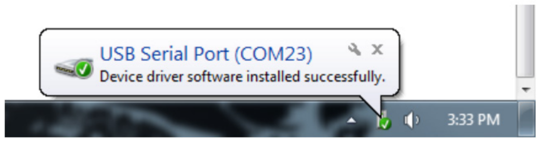

drivers
examples
hardware
java
lib
libraries
tools
tools-builder
arduino.exe
arduino.l4j.ini
arduino_debug.exe
arduino_debug.l4j.ini
arduino-builder.exe
libusb0.dll
msvcp100.dll
msvcr100.dll
revisions.txt
uninstall.exe
wrapper-manifest.xml

Arduino IDE Folder

Drivers and Other Downloads

USB Driver Download

1. In Windows 10, when the development board is connected to the computer, the computer will automatically install the VCP driver. After the driver has been automatically downloaded successfully, a message will be displayed.



USB Serial Port (COM23)
Device driver software installed successfully.

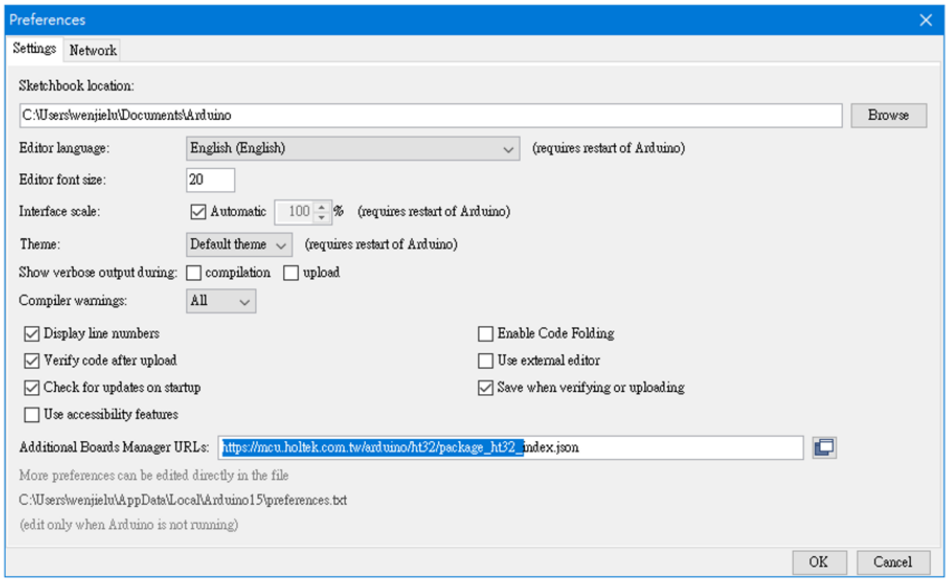
Mount VCP

2. In Windows XP and Windows 7, the VCP driver should be downloaded manually. Refer to the "Keil IDE Software" section for more details.

IDE Setup

Install Holtek HT32 Boards

1. Click "File → Preferences", and select the "Settings" tab. In the "Additional Boards Manager URLs" type https://mcu.holtek.com.tw/arduino/ht32/package_ht32_index.json. After this, click on OK.



Preferences

Settings | Network

Sketchbook location: C:\Users\wenjielu\Documents\Arduino [Browse]

Editor language: English (English) (requires restart of Arduino)

Editor font size: 20

Interface scale: [✓] Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: [ ] compilation [ ] upload

Compiler warnings: All

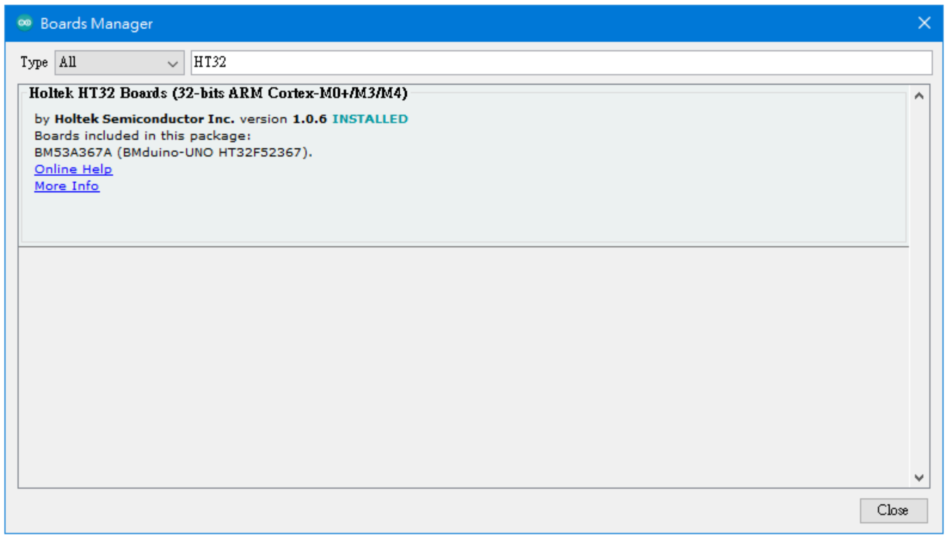
[✓] Display line numbers
[✓] Verify code after upload
[✓] Check for updates on startup
[ ] Use accessibility features
[ ] Enable Code Folding
[ ] Use external editor
[✓] Save when verifying or uploading

Additional Boards Manager URLs: https://mcu.holtek.com.tw/arduino/ht32/package_ht32_index.json

More preferences can be edited directly in the file
C:\Users\wenjielu\AppData\Local\Arduino15\preferences.txt
(edit only when Arduino is not running)

Type Json Path

2. Click “Tools → Board: “Arduino UNO” → Board Manager” to enter the Boards Manager window and search for “HT32”. Locate the Holtek HT32 Boards installation window. Select the latest version and click “Install”. After the installation has completed, click on “Close”.

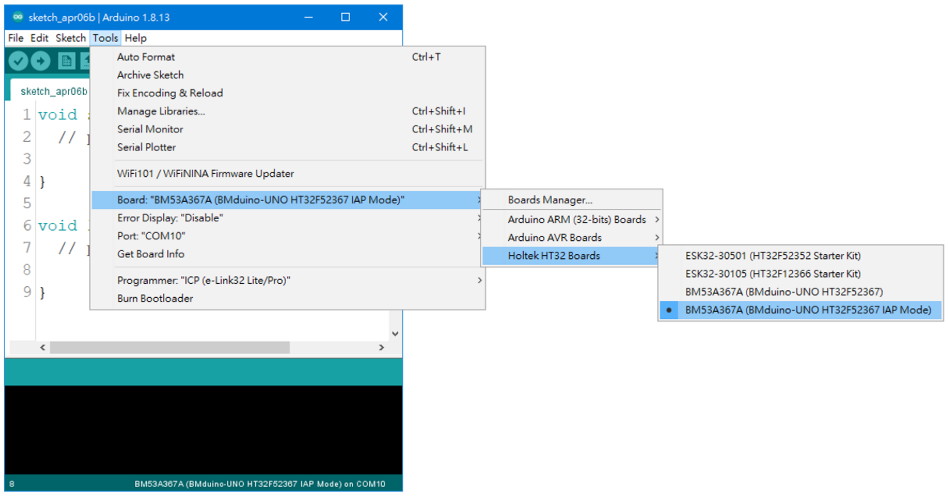


Holtek HT32 Boards (32-bits ARM Cortex-M0+/M3/M4)
by **Holtek Semiconductor Inc.** version **1.0.6** **INSTALLED**
Boards included in this package:
BM53A367A (BMduino-UNO HT32F52367).
Online Help
More Info

Install Holtek Library

ICP Mode Settings – recommended

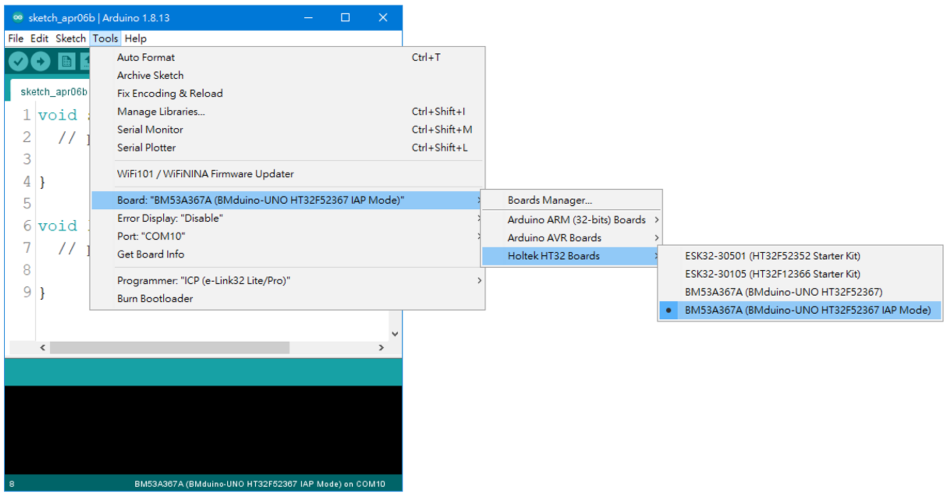
Click “Tools → Board → Holtek HT32 Boards → BM53A367A (BMduino-UNO HT32F52367)” to complete the initialisation.


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_apr06b
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"
Error Display: "Disable"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
Boards Manager...
Arduino ARM (32-bits) Boards
Arduino AVR Boards
Holtek HT32 Boards
ESK32-30501 (HT32F52352 Starter Kit)
ESK32-30105 (HT32F12366 Starter Kit)
BM53A367A (BMduino-UNO HT32F52367)
BM53A367A (BMduino-UNO HT32F52367 IAP Mode)
BM53A367A (BMduino-UNO HT32F52367 IAP Mode) on COM10

Select BM53A367A

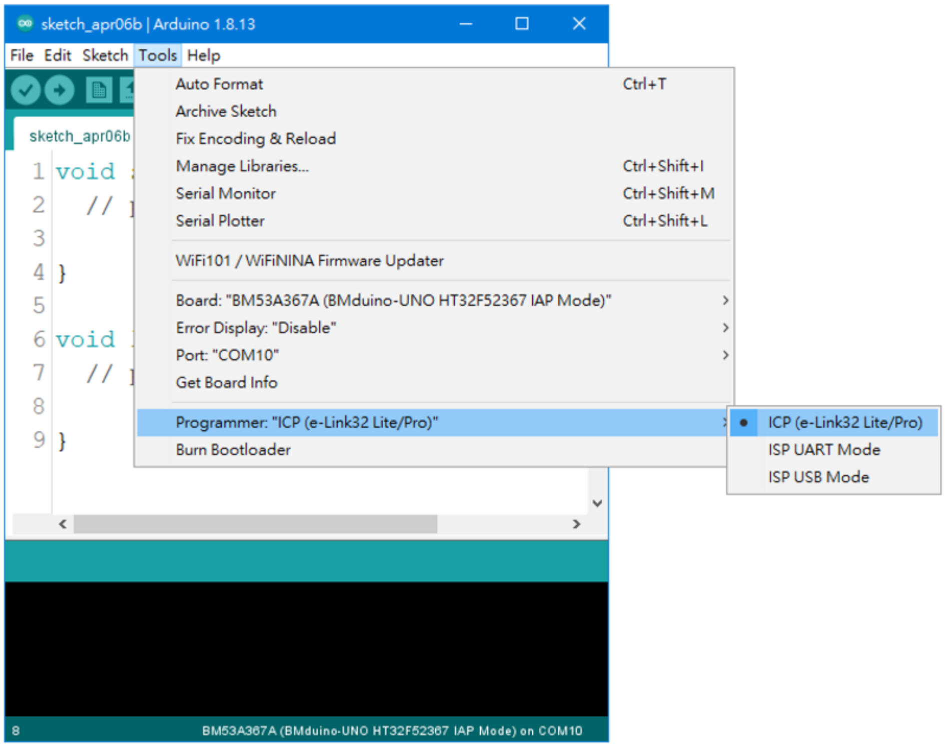
IAP Mode Settings

Setp 1. Click "Tools → Board → Holtek HT32 Boards → BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"
Error Display: "Disable"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
Boards Manager...
Arduino ARM (32-bits) Boards
Arduino AVR Boards
Holtek HT32 Boards
ESK32-30501 (HT32F52352 Starter Kit)
ESK32-30105 (HT32F12366 Starter Kit)
BM53A367A (BMduino-UNO HT32F52367)
BM53A367A (BMduino-UNO HT32F52367 IAP Mode)
BM53A367A (BMduino-UNO HT32F52367 IAP Mode) on COM10

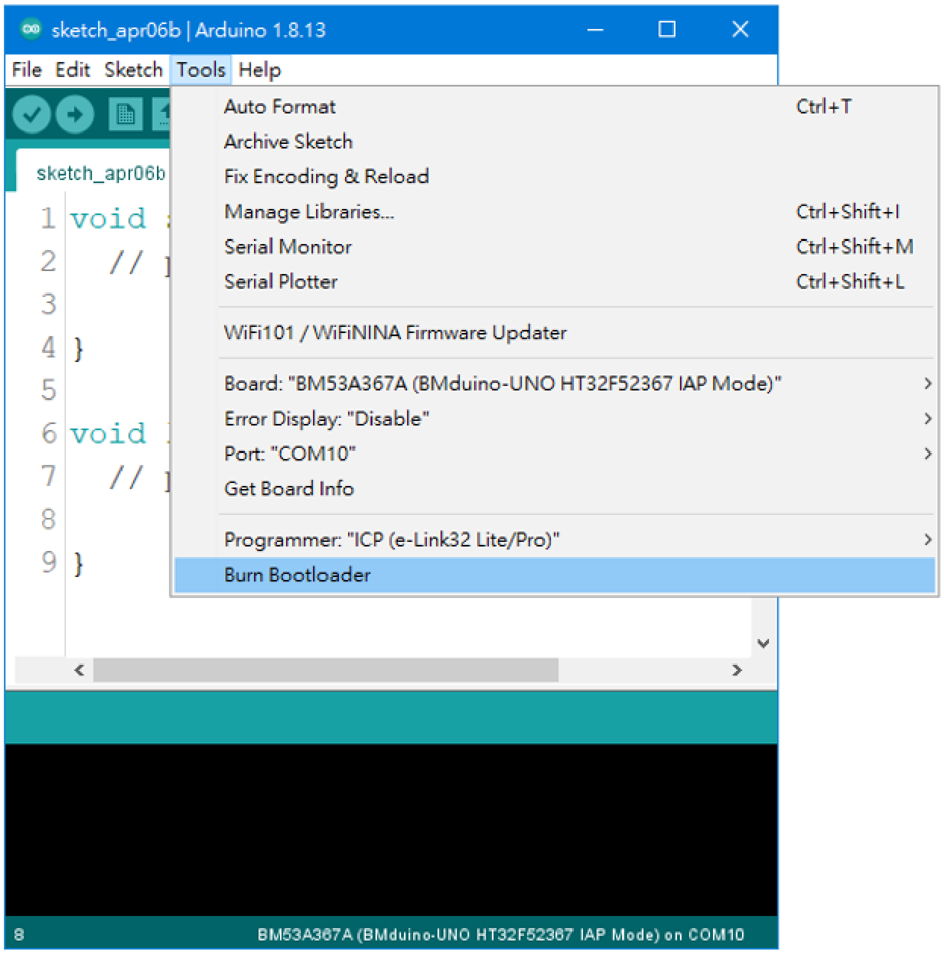
Select BM53A367A IAP Mode

Setp 2. The BM53A367A Bootloader will not have been programmed before delivery. Therefore, it is necessary to program the Bootloader first. To determine which method is used to program the Bootloader, select "Tools → Programmer → ICP (e-Link32 Lite/Pro)".


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"
Error Display: "Disable"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
ICP (e-Link32 Lite/Pro)
ISP UART Mode
ISP USB Mode
BM53A367A (BMduino-UNO HT32F52367 IAP Mode) on COM10

Select ICP (e-Link32 Lite/Pro)

Setp 3. To implement Bootloader programming, click "Tools → Burn Bootloader".


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367 IAP Mode)"
Error Display: "Disable"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
BM53A367A (BMduino-UNO HT32F52367 IAP Mode) on COM10

Select Burn Bootloader

Setp 4. After completion of the Burn Bootloader, the D13 LED will flash to indicate that the Bootloader has been successfully programmed and that the IAP mode initialisation has been completed.

Arduino Library Reference

As the hardware resources between the BMduino-UNO BM53A367A and the Arduino UNO R3 are different, some Library parameters or execution methods will be different when using the Arduino IDE integrated libraries, as shown in the following table:

#	Library	BMduino-UNO BM53A367A	Arduino UNO R3
1	Serial	Data bits: 7~9 bits Receive buffer: 255 bytes	Data bits: 5~8 bits Receive buffer: 64 bytes
2	analogReference()	Internal Vref: 1.215V or 2V or 2.5V or 2.7V	Internal Vref: 1.1V
3	SPI	setClockDivider(4) → SPI SCK=15MHz	setClockDivider(4) → SPI SCK=4MHz
4	tone()	Minimum frequency: 1Hz	Minimum frequency: 31Hz
5	analogWrite()	PWM frequency: 1000Hz Pins: D0~D13	PWM frequency: 490/980Hz Pins: D3, D5, D6, D9, D10, D11

#	Library	BMduino-UNO BM53A367A	Arduino UNO R3
6	SoftwareSerial	TX: 230400bps, RX: 115200bps Response delay time: >120μs	TX/RX: 57600bps Response delay time: >15μs
7	Servo	Disables analogWrite() on D23 (BMCOM2)	Disables analogWrite() on D9, D10
8	MsTimer2	Integrated Library, renamed MsTimer to prevent duplicate naming	Third-party Library
9	attachInterrupt()	D2~D12, D22, D25, D29	D2, D3

1. Serial.begin()

   Syntax: Serail.begin(speed, config)

   Valid values for config are as follows:

BMduino-UNO BM53A367A	Arduino UNO R3
SERIAL_7N1 SERIAL_8N1 (default) SERIAL_9N1 SERIAL_7N2 SERIAL_8N2 SERIAL_9N2 SERIAL_7E1: even parity SERIAL_8E1 SERIAL_9E1 SERIAL_7E2 SERIAL_8E2 SERIAL_9E2 SERIAL_7O1: odd parity SERIAL_8O1 SERIAL_9O1 SERIAL_7O2 SERIAL_8O2 SERIAL_9O2	SERIAL_5N1 SERIAL_6N1 SERIAL_7N1 SERIAL_8N1 (default) SERIAL_5N2 SERIAL_6N2 SERIAL_7N2 SERIAL_8N2 SERIAL_5E1: even parity SERIAL_6E1 SERIAL_7E1 SERIAL_8E1 SERIAL_5E2 SERIAL_6E2 SERIAL_7E2 SERIAL_8E2 SERIAL_5O1: odd parity SERIAL_6O1 SERIAL_7O1 SERIAL_8O1 SERIAL_5O2 SERIAL_6O2 SERIAL_7O2 SERIAL_8O2

2. analogReference()

   Syntax: analogReference(type)

   Valid values for type are as follows:

BMduino-UNO BM53A367A	Arduino UNO R3
DEFAULT: the default analog reference of 3.3V INTERNAL1V215: integrated 1.215V reference INTERNAL2V0: integrated 2V reference INTERNAL2V5: integrated 2.5V reference INTERNAL2V7: integrated 2.7V reference EXTERNAL: the voltage applied to the AREF pin (0 to3.3V only) is used as the reference	DEFAULT: the default analog reference of 5V INTERNAL: integrated 1.1V reference EXTERNAL: the voltage applied to the AREF pin (0 to 5V only) is used as the reference

3. SPI.setClockDivider()

   Syntax: SPI.setClockDivider(divider)

   The setClockDivider is the API for frequency division. Due to different operating frequencies, the SCK output frequency is different even if the divider parameter is the same. The BM53A367A has an operating frequency of 60MHz while the Arduino UNO R3 has 16MHz. For example:

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

4. tone()

   Syntax: tone(pin, frequency)

   tone(pin, frequency, duration)

   The BM53A367A has a minimum output frequency of 1Hz while the Arduino UNO R3 has 31Hz.

5. analogWrite()

   Syntax: analogWrite(pin, value)

   The BM53A367A has a PWM cycle of 1000Hz while the Arduino UNO R3 has 490Hz/980Hz.

6. SoftwareSerial

   BM53A367A: TX supports up to 230400bps and RX supports up to 115200bps

   Arduino UNO R3: TX/RX supports up to 57600bps

7. Servo

   BM53A367A: the D23 analogWrite() PWM output function is disabled. D23 belongs to BMCOM2.

   Arduino UNO R3: the D9 and D10 analogWrite() PWM output functions are disabled.

8. MsTimer2

   BM53A367A: The Library is integrated, whose name has been changed to MsTimer. Related examples can be found in "File → Example → MsTimer".

   Arduino UNO R3: The Library provided by the third party should be downloaded through the Library Manager.

9. attachInterrupt()

   Syntax: attachInterrupt(interruptNum, FuncPtr callback, mode)

   BM53A367A: D2~D12, D22, D25, D29 can be used as external interrupts.

   Arduino UNO R3: D2 and D3 can be used as external interrupts.

Examples

Hardware Preparation

It is necessary to prepare the development board, Type-C USB cable and computer. Connect the e-link32 Lite of the development board to the computer using a USB cable. At this point, the PWR LED will be illuminated. After the e-link32 Lite completes its enumeration, the RDY LED will be illuminated. The hardware preparation has now completed.

Code

Execute the Blink example, the details of which can obtained from the following link. The D13 LED will flash once per second after the example programming has completed.

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

Keil IDE Software

Keil IDE Download and Installation

Open the Keil official website (https://www.keil.com/demo/eval/arm.htm) to download the MDK-ARM and execute the installation. For details, refer to the link below:

https://www.holtek.com/documents/10179/6393504/HT32_Keil-QuickStartv110.pdf

IDE Setup

1. Download the HT32 development resources: Click the link below to download the latest HT32F5 Series (Cortex®-M0+), which contains all the resources required for HT32 development. Extract the files from the download after completion.

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

2. Install HT32 Packs and execute "\HT32_M0p_vxxxxxxxx\Tools\Holtek.HT32_DFP.xx.xx.xx.pack".
2. Install the VCP driver and execute "\HT32_M0p_vxxxxxxxxx\Tools\HT32_VCP_Driver_vxxxx.exe".
4. Extract the HT32 FW Lib into the path "\Firmware_Library\HT32_STD_5xxxx_FWLib_Vx.x.x_xxxx.zip".

Run Example

1. Open the Keil project in the HT32 FW Lib. Path: "\Firmware_Library\HT32_STD_5xxxx_FWLib_Vx.x.x_xxxx.zip\project_template\IP\Example\MDK_ARMv5\Project_53a367a.uvprojx".
2. Compile and program using the Keil IDE. Refer to the "Programming" section for details.
3. Press the RESET button and observe if the D13 LED flashes 5 times quickly which indicates test completion.

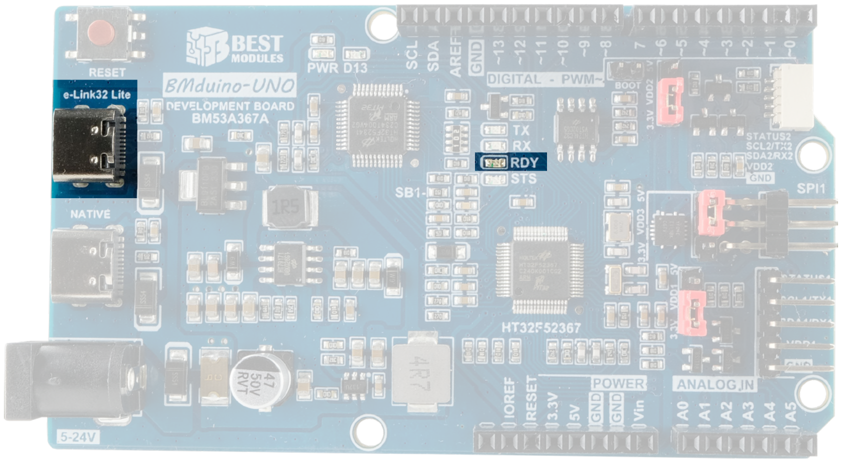
   Supplement: This example also illustrates the Serial (115200, 8, N, 1) function, enables the COM Port and observes the prompts using the terminal software such as Tera Term. For details refer to the official website: https://ttssh2.osdn.jp/index.html.en.

Troubleshooting

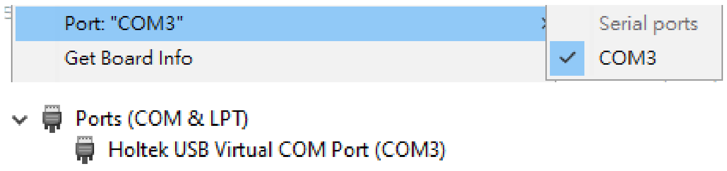
Serial Ports not Appearing in Port Menu

This indicates that the e-Link32 Lite VCP connection has failed. Execute the following steps to debug:

1. Check whether the e-Link Lite USB is connected to the PC and whether the RDY LED is illuminated.



2. If the RDY LED has not illuminated, remove the e-Link Lite USB and insert it into another USB port on the PC.
3. Restart the PC if the RDY LED has still not illuminated after 10 seconds. If the RDY LED has illuminated, the BMduino COM port can be found in the Arduino COM Port menu.

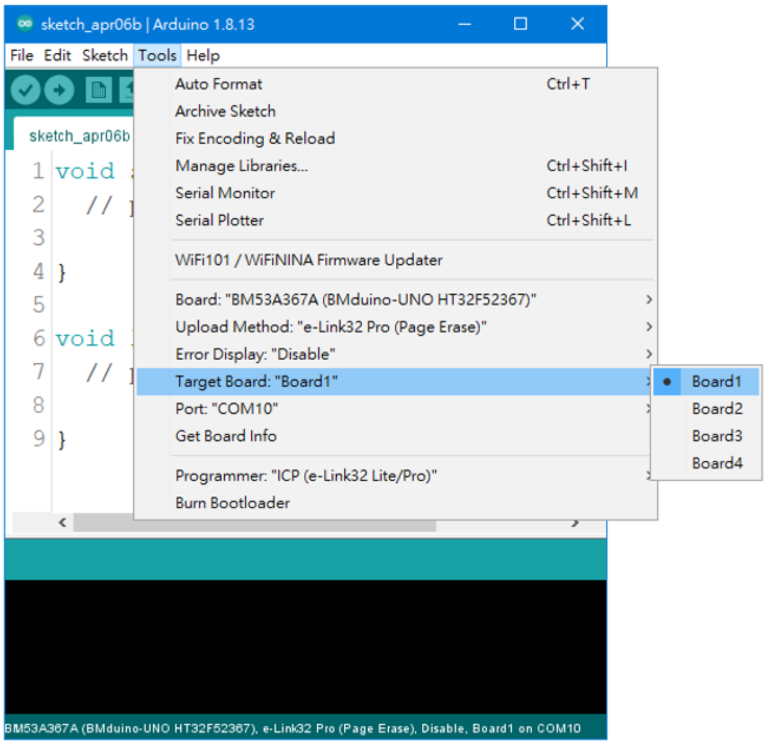


4. If the RDY LED has illuminated, but the BMduino COM Port does not appear in the Arduino COM Port menu, install the VCP driver. This is only available for Windows computers.

   Refer to the IDE Setup step 3 in the Keil IDE Software section to install the VCP driver.

Program Upload Fails or Freezes

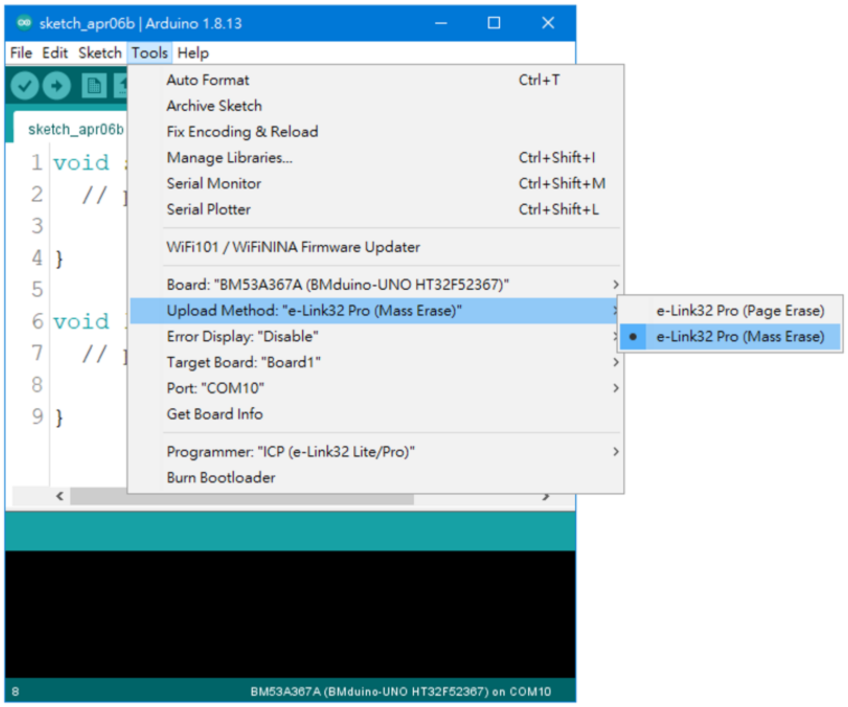
1. The error message "**This computer can't enumerate any e-Link32 Pro/Lite. Please make sure this computer has indeed connected to e-Link32 Pro/Lite.**" indicates that the connection of e-Link32 Lite CMSIS-DAP (programming interface) has failed. Check whether the e-Link Lite USB is connected to the PC and whether the RDY LED is illuminated. If the RDY LED is not illuminated, remove the e-Link Lite USB and insert it into another USB port on the PC. Restart the PC if the RDY LED has still not illuminated after 10 seconds.
2. The error message "**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.**" indicates that the specified target board has not been found. Click "Tools → Target Board: → Board1" and then execute "Upload".


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367)"
Upload Method: "e-Link32 Pro (Page Erase)"
Error Display: "Disable"
Target Board: "Board1"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
● Board1
Board2
Board3
Board4

```
BM53A367A (BMduino-UNO HT32F52367), e-Link32 Pro (Page Erase), Disable, Board1 on COM10
```

Select Board1

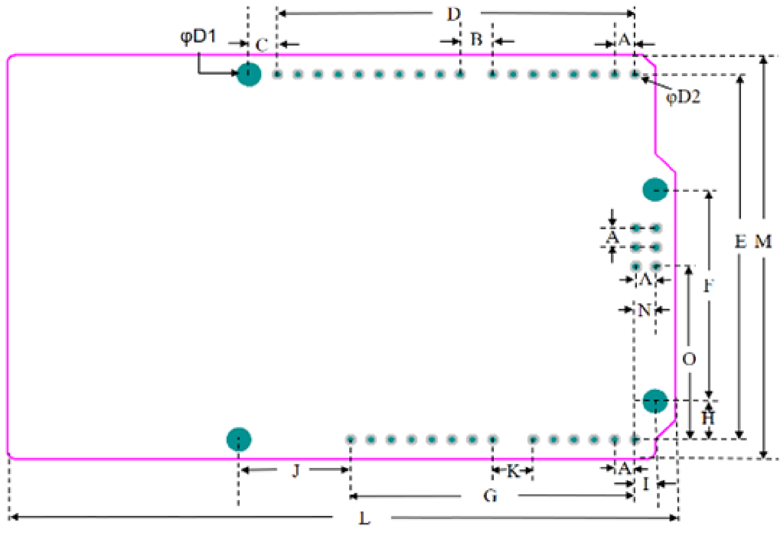
3. If the Upload failure information is not in any of the above described situations, execute the Mass Erase command to erase the MCU and then execute programming again. Click "Tools → Upload Method: → e-Link32 Pro (Mass Erase)" and then execute "Upload".


sketch_apr06b | Arduino 1.8.13
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "BM53A367A (BMduino-UNO HT32F52367)"
Upload Method: "e-Link32 Pro (Mass Erase)"
Error Display: "Disable"
Target Board: "Board1"
Port: "COM10"
Get Board Info
Programmer: "ICP (e-Link32 Lite/Pro)"
Burn Bootloader
e-Link32 Pro (Page Erase)
● e-Link32 Pro (Mass Erase)

```
8                    BM53A367A (BMduino-UNO HT32F52367) on COM10
```

Mass Erase

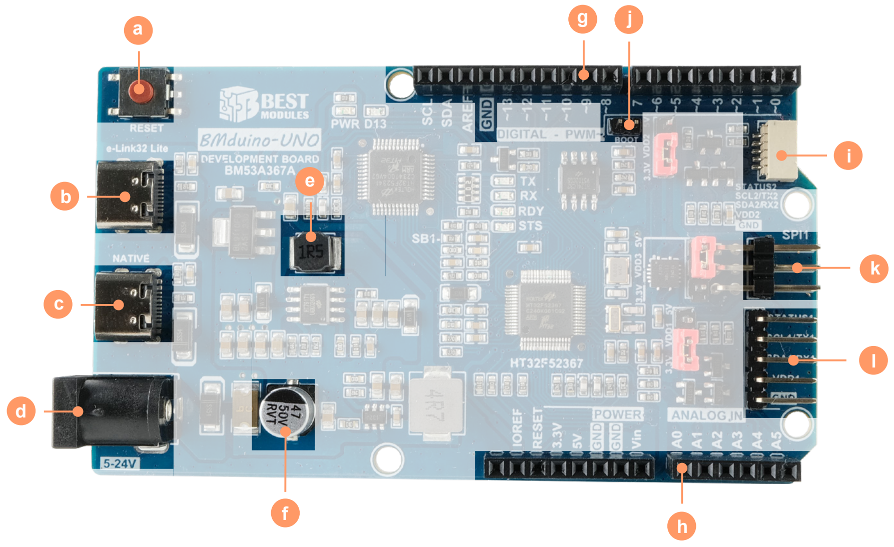
Dimensions



Dimension Information

Unit / No.	mm	inch
A	2.54	0.1
B	4.064	0.16
C	3.556	0.14
D	44.704	1.76
E	48.26	1.9
F	27.94	1.1
G	36.56	1.4
H	5.08	0.2
I	2.54	0.1
J	13.97	0.55
K	5.08	0.2
L	93.221	3.67
M	53.35	2.1
N	2.667	0.105
O	22.86	0.9
D1	3.2512	0.128

Dimension List



Component Size – Height Information

No. \ Size	Length		Width		Height	
	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
c	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
l	8.5	0.334	12.776	0.503	4	0.157

Default height: 11mm / 0.433 inch

Component Size – Height List