BMduino-Nano Development Board

BM18B367A
User Guide

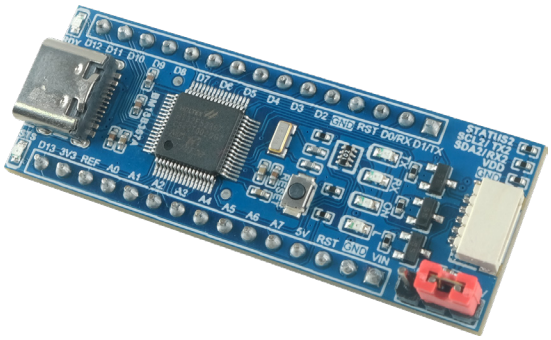
Revision: V1.10 Date: November 22, 2024

www.bestmodulescorp.com

Contents

Introduction

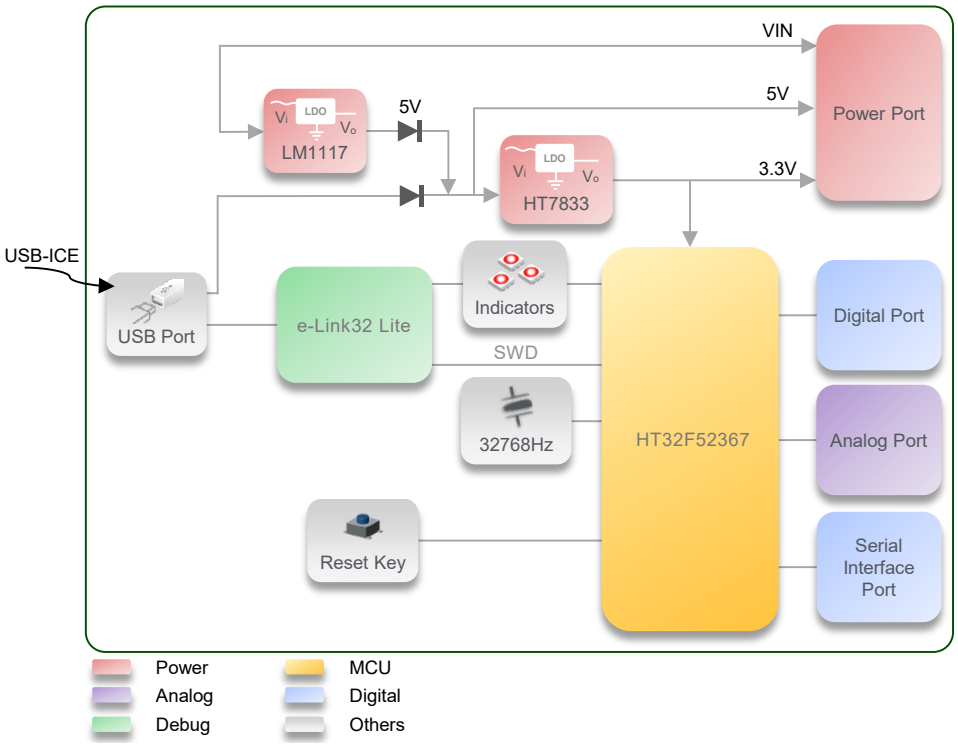
The BMduino-Nano BM18B367A is a development board implemented using a Holtek 32-bit MCU, the HT32F52367, which is especially designed to be pin-compatible with the Arduino Nano development board. It can support both Arduino IDE and Keil IDE development platforms to assist beginners to learn programming more easily. The BM18B367A uses the HT32F52367, which is based on a 3.3V series Arm® Cortex® -M0+ core, as the master MCU. It also has additional common communication interfaces such as I²C, SPI and UART.



Features

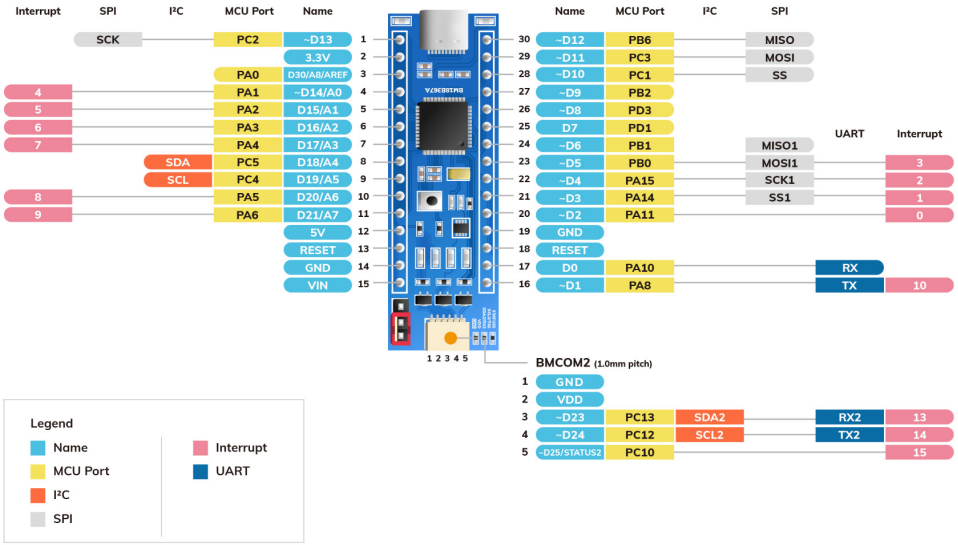
- Master MCU: HT32F52367 - 64-pin LQFP
  - ♦ Cortex®-M0+, 60MHz
  - ♦ Flash Memory: 256KB
  - ♦ SRAM: 32KB
- 26 digital I/O pins, 16 of which can be used as PWM outputs
- 12-bit resolution A/D converter with 9 analog inputs
- 14 external interrupts
- Communication interfaces: UART, SPI, I²C
- Power supply inputs: USB interface, external Vin
  - ♦ e-Link32 Lite USB: Type-C USB interface
  - ♦ External Vin: Vin pin, voltage range: DC 7V~12V
- 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: BMCOM2 (lead pitch: 1.0mm, optional 3.3V/5V)
- Development board size: 50mm×18mm×18.4mm, pin-compatible with the Arduino Nano development board
- Development board weight: 7g (Net)

Block Diagram



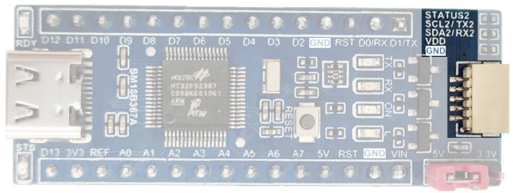
Pin Description

BMduino-Nano BM18B367A Pin Description



Pin No.	Function	Description
1	~D13/SCK	Digital pin D13 with PWM function / SPI0 interface SCK pin
2	+3V3	+3.3V power supply output
3	AREF/A8/D30	Analog reference voltage / Analog input pin A8 / Digital pin D30
4	A0/~D14/INT4	Analog input pin A0 / Digital pin D14 with PWM function / External interrupt pin INT4
5	A1/D15/INT5	Analog input pin A1 / Digital pin D15 / External interrupt pin INT5
6	A2/D16/INT6	Analog input pin A2 / Digital pin D16 / External interrupt pin INT6
7	A3/D17/INT7	Analog input pin A3 / Digital pin D17 / External interrupt pin INT7
8	A4/D18/SDA	Analog input pin A4 / Digital pin D18 / I²C0 interface SDA pin
9	A5/D19/SCL	Analog input pin A5 / Digital pin D19 / I²C0 interface SCL pin
10	A6/D20/INT8	Analog input pin A6 / Digital pin D20 / External interrupt pin INT8
11	A7/D21/INT9	Analog input pin A7 / Digital pin D21 / External interrupt pin INT9
12	+5V	+5V power supply output
13	RESET	Reset pin
14	GND	Power ground
15	Vin	Power supply input 7V~12V
16	~D1/TX/INT10	Digital pin D1 with PWM function / UART0 transmit pin / External interrupt pin INT10
17	D0/RX	Digital pin D0 / UART0 receive pin
18	RESET	Reset pin
19	GND	Power ground
20	~D2/INT0	Digital pin D2 with PWM function / External interrupt pin INT0
21	~D3/SS1/INT1	Digital pin D3 with PWM function / SPI1 interface SS pin / External interrupt pin INT1
22	~D4/SCK1/INT2	Digital pin D4 with PWM function / SPI1 interface SCK pin / External interrupt pin INT2
23	~D5/MOSI1/INT3	Digital pin D5 with PWM function / SPI1 interface MOSI pin/ External interrupt pin INT3
24	~D6/MISO1	Digital pin D6 with PWM function / SPI1 interface MISO pin
25	D7	Digital pin D7
26	~D8	Digital pin D8 with PWM function
27	~D9	Digital pin D9 with PWM function
28	~D10/SS	Digital pin D10 with PWM function / SPI0 interface SS pin
29	~D11/MOSI	Digital pin D11 with PWM function / SPI0 interface MOSI pin
30	~D12/MISO	Digital pin D12 with PWM function / SPI0 interface MISO pin

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



Pin No.	Function	Description
1	GND	Power ground
2	VDD	3.3V or 5V power supply output, determined by the adjacent jumper
3	~D23/RX2/SDA2/INT13	Digital pin D23 with PWM function / UART2 receive pin / I²C2 interface SDA2 pin / External interrupt pin INT13
4	~D24/TX2/SCL2/INT14	Digital pin D24 with PWM function / UART2 transmit pin / I²C2 interface SCL2 pin / External interrupt pin INT14
5	~D25/STATUS2/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.

Technical Specifications

Absolute Maximum Ratings

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

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

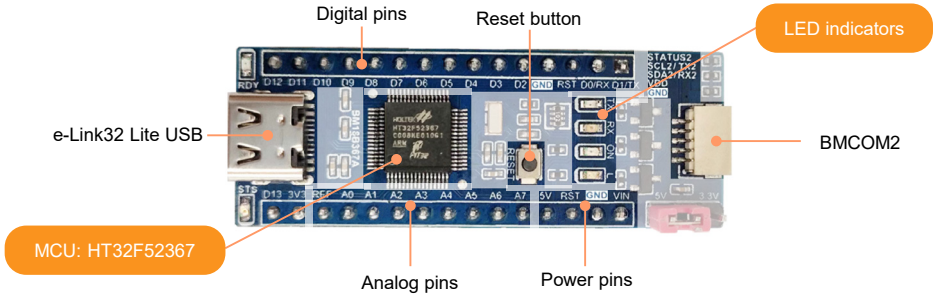
- A voltage of 7V~12V is input through Vin, which is reduced to 5V and 3.3V using a buck regulator circuit to provide the MCU power supply.

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{inMAX}	Maximum Input Voltage from Vin Pin	—	—	15	V
V_{USBMAX}	Maximum Input Voltage from USB Connector	—	5.5	—	V

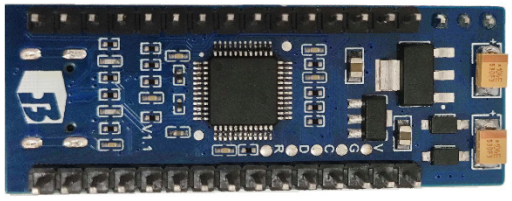
Recommended Operation Conditions

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
V_{in}	Input Voltage	—	7	—	12	V
I_{OUT}	5V Output Current	V_{in}=10V	—	—	900	mA
	3.3V Output Current	—	—	—	500	mA
	I/O Output Current	—	—	—	16	mA

Hardware Overview



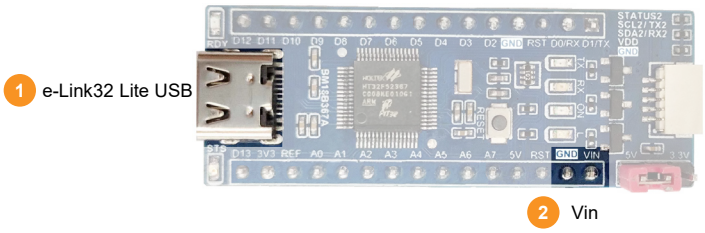
PCBA Front View



PCBA Back View

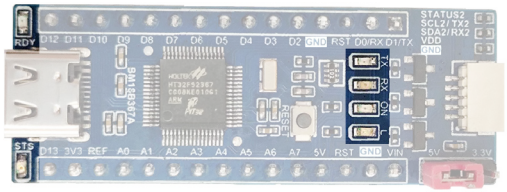
Power Supply

- 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 7V~12V 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

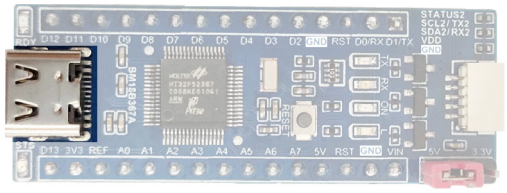
LED Indicators



LED Configuration Diagram

- ON: Power LED.
- L: 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.

e-Link32 Lite

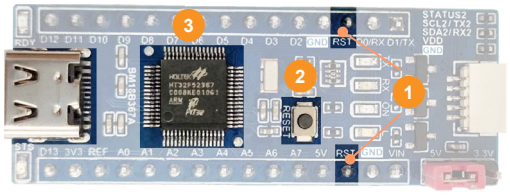


e-Link32 Lite main functions:

- Implements the master MCU programming.
- Supports the Virtual COM Port, VCP, to connect to the master MCU TX and RX pins.
- Provides the program debugging function under the Keil IDE development environment, such as for breakpoint setting.

BMduino-Nano BM18B367A is preset to use the e-Link32 Lite circuit for programming, without requiring the TX and RX pins.

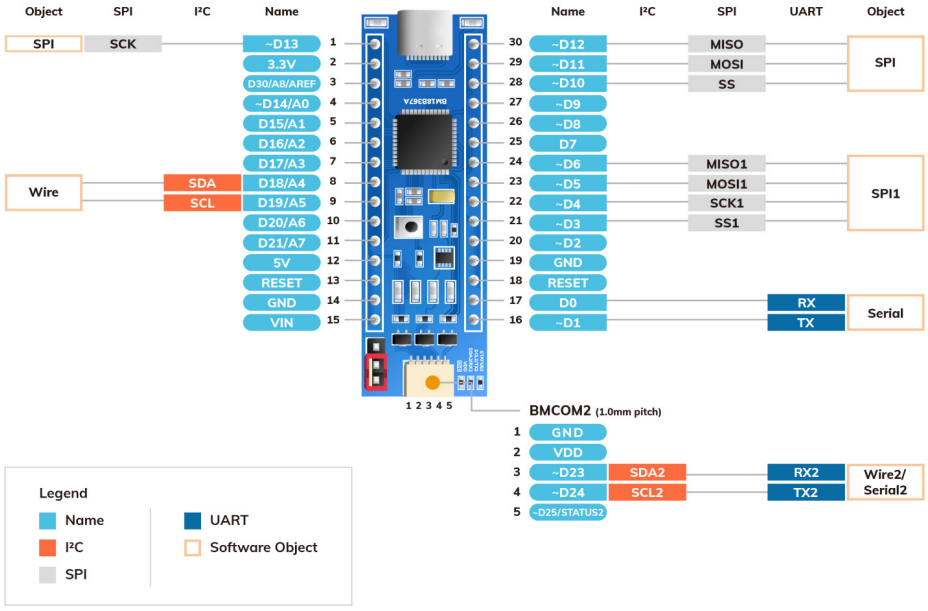
Reset Circuit



- Set the RST pin 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.

UART, I²C, SPI

BMduino-Nano BM18B367A
UART, I²C, SPI



- Supports two UART interfaces, whose object names are Serial and Serial2.

Obj	Serial	Serial2
Pins	D0 (RX) / D1 (TX)	RX2 / TX2 (BMCOM2)

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

Example program:

```
void setup() {
  // initialise serial communication:
  Serial.begin(9600);
}
void loop() {
  // send the value of analog input 0:
  Serial.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-Nano BM18B367A 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 two I²C interfaces, whose object names are Wire and Wire2.

Obj	Wire	Wire2
Pins	A4(SDA) / A5(SCL)	SDA2 / SCL2 (BMCOM2)

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

Example program:

```
#include <Wire.h>
void setup()
{
  Wire.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(Wire.available())  // slave may send less than requested
  {
    char c = Wire.read();  // receive a byte as character
    Serial.print(c);       // print the character
  }
  delay(500);
}
```

- Supports two SPI interfaces.

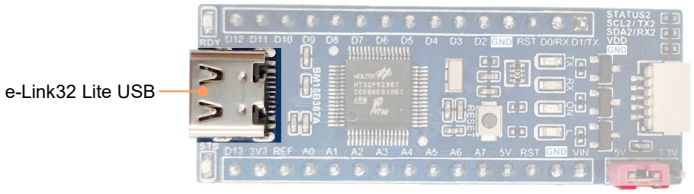
Obj	SPI	SPI1
Pins	D10 (SS) / D11 (MOSI) / D12 (MISO) / D13 (SCK)	D3 (SS1) / D5 (MOSI1) / D6 (MISO1) / D4 (SCK1)

For example, to select the SPI interface, declare SPI.h and use the SPI 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 SPI:
  SPI.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 SPI:
  SPI.transfer(address);
  SPI.transfer(value);
  // take the SS pin high to de-select the chip:
  digitalWrite(slaveSelectPin, HIGH);
}
```

USB Interface Circuit



BMduino-Nano BM18B367A development board has a USB port.

- e-Link32 Lite : used for programming and VCP, connect it to the computer to program and generate a COM port. When the Arduino IDE or Keil IDE is used for program development, connect this USB port to the computer's USB port.

BM18B367A vs Nano

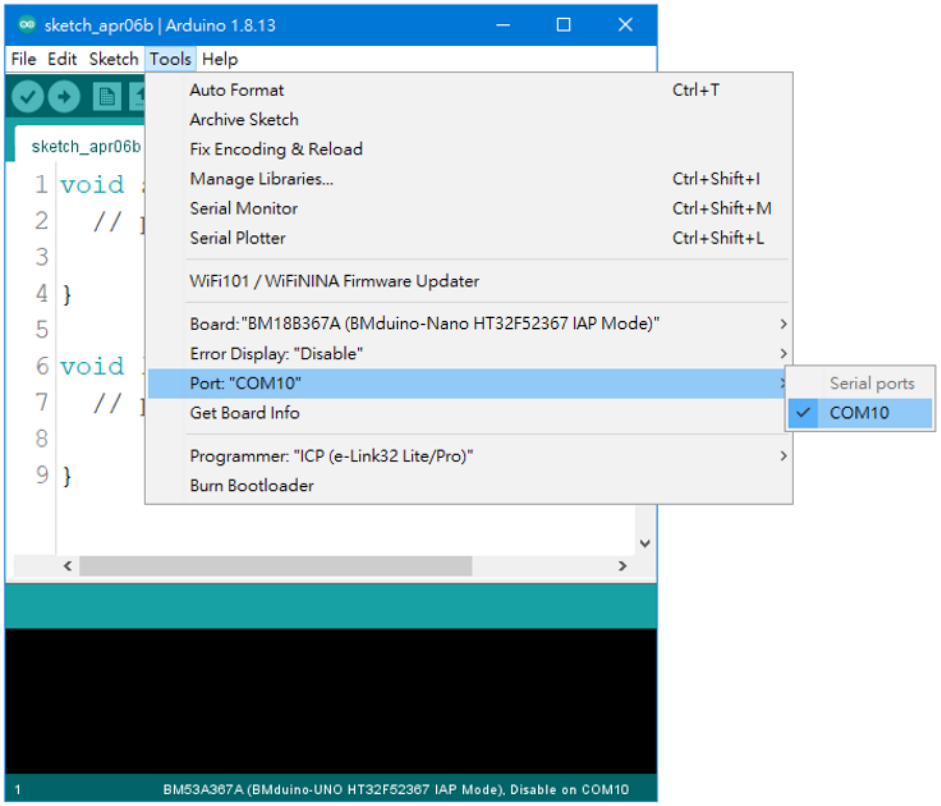
Board	BMduino-Nano BM18B367A	Arduino Nano
Core	Cortex®-M0+, 60MHz	AVR 8-bit, 16MHz
Flash / EEPROM / SRAM	256KB / - / 32KB	32KB / 1KB / 2KB
Operating Voltage	3.3V	5V
Programming Mode	ICP/IAP (Bootloader)	IAP (Bootloader)
Programming Interface	SWD (Target Board Menu) / UART (Port Menu)	UART (Port Menu)
Development Environment	Arduino, Keil	Arduino
I/O Drive Current	16mA	20mA

Programming

BM18B367A 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”before “Upload”, as shown in the following diagram.



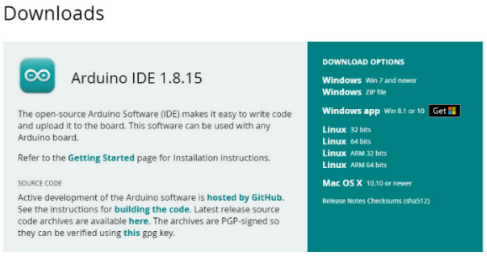
COM Port Menu

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 “Build Output” window . For more details, refer to the Keil official website: https://www2.keil.com/mdk5/learn.

Arduino IDE Software

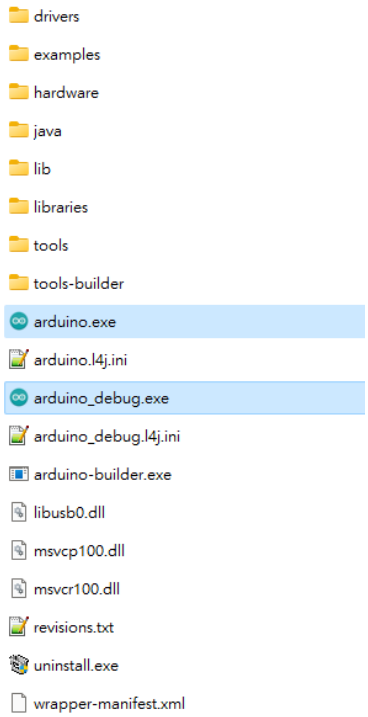
Arduino IDE Download and Installation

Open the Arduino official website (Software | Arduino) to download the corresponding Arduino IDE software according to the computer operation system and execute the installation.



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.

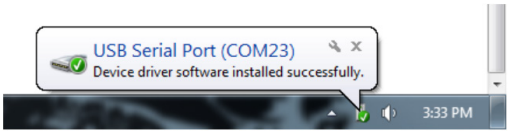


Arduino IDE Folder

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.

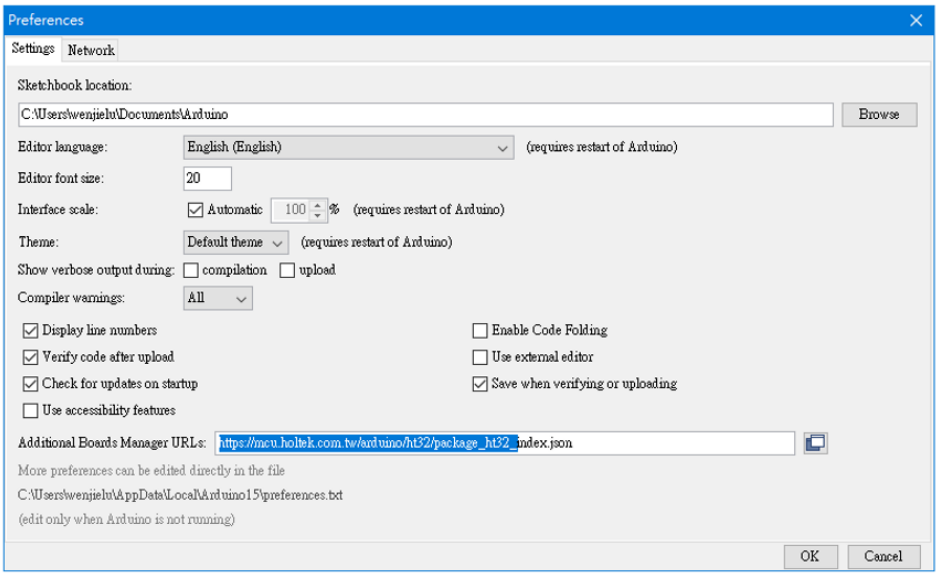


Mount VCP

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

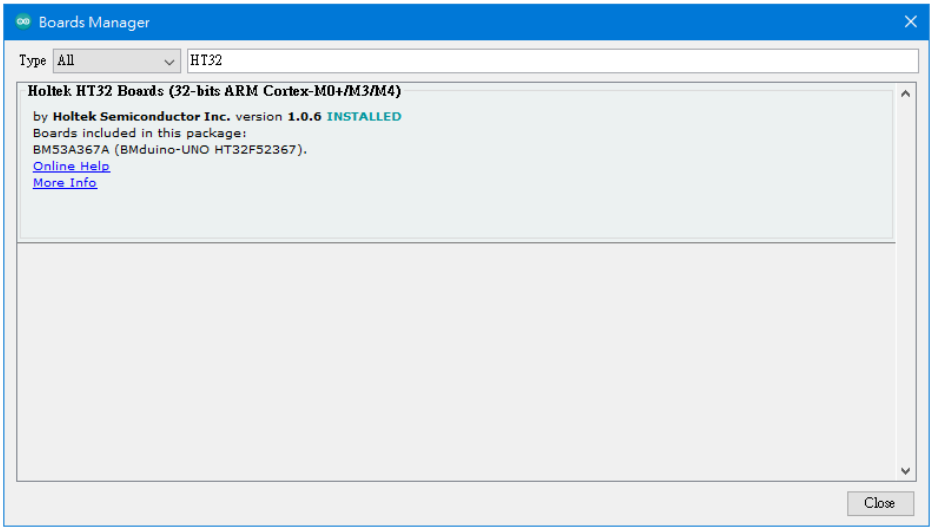
IDE Setup

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.



Type Json Path

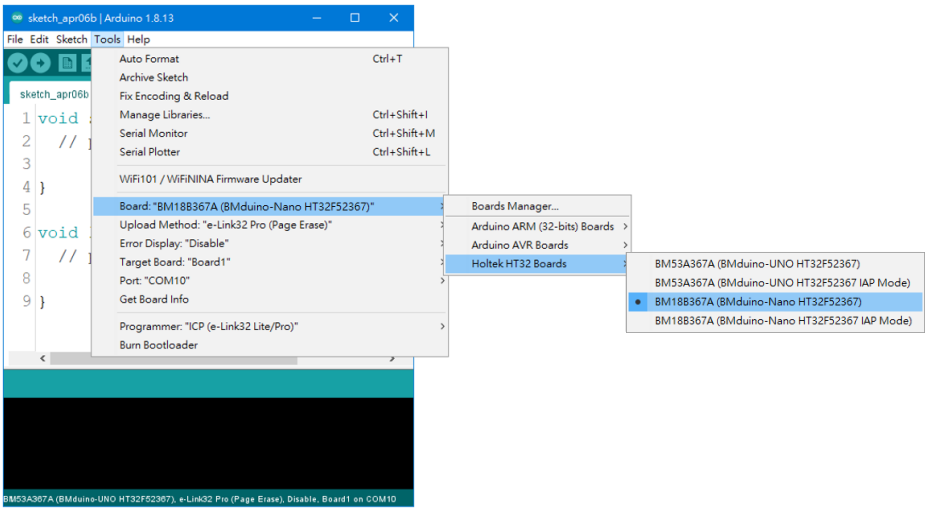
2. Click “Tools → Board: “Arduino Nano” → Board Manager” to enter the Boards Manager window and search for “HT32”. Locate the Holtek HT32 Boards installation window. Select the latest version and click “Install”. After the installation has completed, click on “Close”.



Install Holtek Library

ICP Mode Settings – recommended

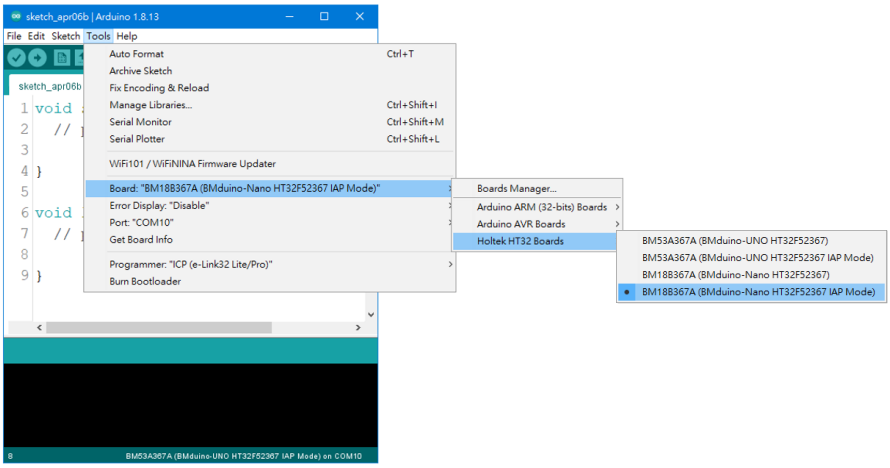
Click “Tools → Board → Holtek HT32 Boards → BM18B367A (BMduino-Nano HT32F52367)” to complete the initialisation.



Select BM18B367A

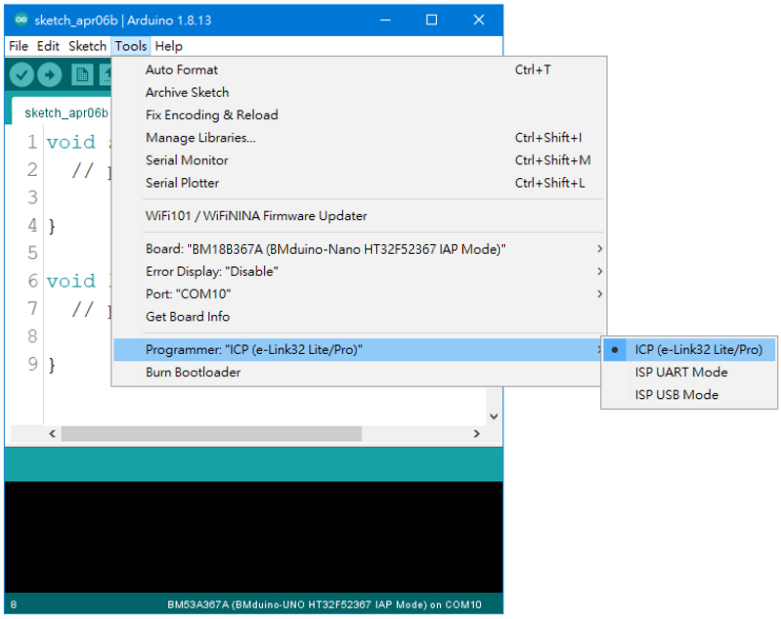
IAP Mode Settings

Step 1. Click “Tools → Board → Holtek HT32 Boards → BM18B367A (BMduino- Nano HT32F52367 IAP Mode)”.



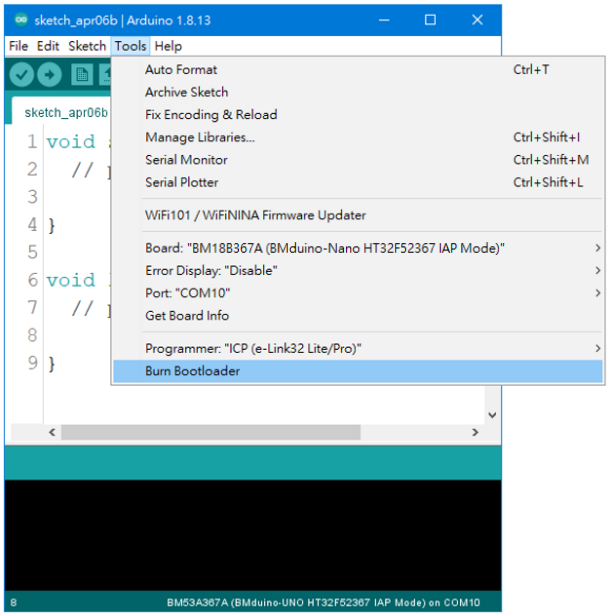
Select BM18B367A IAP Mode

Step 2. The BM18B367A 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)”.



Select ICP (e-Link32 Lite/Pro)

Step 3. To implement Bootloader programming, click “Tools → Burn Bootloader”.



Select Burn Bootloader

Setp 4. After completion of the Burn Bootloader, the L 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-Nano BM18B367A and the Arduino Nano 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-Nano BM18B367A	Arduino Nano
1	Serial	Data bits: 7~9 bits	Data bits: 5~8 bits
2	analogReference()	Internal Vref: 1.215V, 2V, 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	PWM frequency: 490/980Hz
6	SoftwareSerial	TX: 230400bps, RX: 115200bps	TX/RX: 57600bps
7	Servo	Disables analogWrite() on D4	Disables analogWrite() on D9, D8
8	MsTimer2	Integrated Library, renamed MsTimer to prevent duplicate naming	Third-party Library

1. Serial.begin()

Syntax: Serail.begin(speed, config)

Valid values for config are as follows:

BMduino-Nano BM18B367A	Arduino Nano
	SERIAL_5N1
	SERIAL_6N1
	SERIAL_7N1
SERIAL_7N1	SERIAL_8N1 (default)
SERIAL_8N1 (default)	SERIAL_5N2
SERIAL_9N1	SERIAL_6N2
SERIAL_7N2	SERIAL_7N2
SERIAL_8N2	SERIAL_8N2
SERIAL_9N2	SERIAL_5E1: even parity
SERIAL_7E1: even parity	SERIAL_6E1
SERIAL_8E1	SERIAL_7E1
SERIAL_9E1	SERIAL_8E1
SERIAL_7E2	SERIAL_5E2
SERIAL_8E2	SERIAL_6E2
SERIAL_9E2	SERIAL_7E2
SERIAL_7O1: odd parity	SERIAL_8E2
SERIAL_8O1	SERIAL_5O1: odd parity
SERIAL_9O1	SERIAL_6O1
SERIAL_7O2	SERIAL_7O1
SERIAL_8O2	SERIAL_8O1
SERIAL_9O2	SERIAL_5O2
	SERIAL_6O2
	SERIAL_7O2
	SERIAL_8O2

2. analogReference()

   Syntax: analogReference(type)

   Valid values for type are as follows:

BMduino-Nano BM18B367A	Arduino Nano
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 to 3.3V 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)

   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 BM18B367A has an operating frequency of 60MHz while the Arduino Nano has 16MHz. For example:

BMduino-Nano BM18B367A	Arduino Nano
SPI.setClockDivider(4) → SCK = 60MHz / 4 = 15MHz	SPI.setClockDivider(4) → SCK = 16MHz / 4 = 4MHz

4. tone()

   Syntax: tone(pin, frequency)

   tone(pin, frequency, duration)

   BM18B367A has a minimum output frequency of 1Hz while the Arduino Nano has 31Hz.

5. analogWrite()

   Syntax: analogWrite(pin, value)

   BM18B367A has a PWM cycle of 1000Hz while the Arduino Nano has 490Hz/980Hz.

6. SoftwareSerial

   BM18B367A: TX supports up to 230400bps and RX supports up to 115200bps

   Nano: TX/RX supports up to 57600bps

7. Servo

   BM18B367A: the D4 analogWrite() PWM output function is disabled.

   Nano: the D8 and D9 analogWrite() PWM output function is disabled.

8. MsTimer2

   BM18B367A: The Library is integrated, whose name has been changed to MsTimer. Related examples can be found in "File → Example → MsTimer".

   Nano: The Library provided by the third party should be downloaded through the Library Manager.

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 ON 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 L 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_vxxxxxxx\Tools\Holtek.HT32_DFP.xx.xx.xx.pack".

3. Install the VCP driver and execute "\HT32_M0p_vxxxxxxxx\Tools\HT32_VCP_Driver_vxxx.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_18b367a.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 L LED flashes 5 times quickly which indicates test completion.

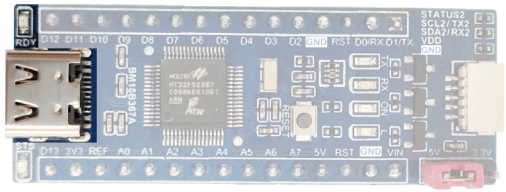
Supplement: This example also illustrates the Serial (115200, 8, N, 1) function, enables the COM Port and observes the prompts using the terminal software such as Tera Term.

Troubleshooting

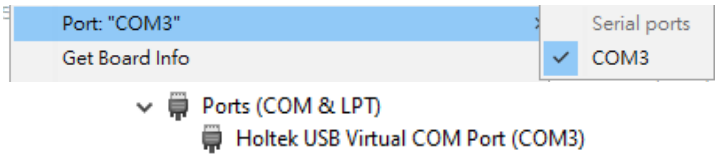
Serial Ports not Appearing in Port Menu

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.

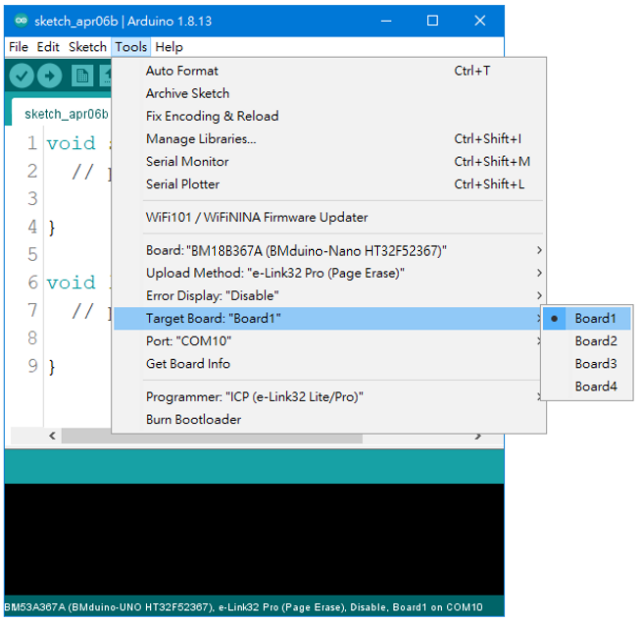
   Refer to the IDE Setup step 3 in the Keil IDE Software section to install the VCP driver.

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

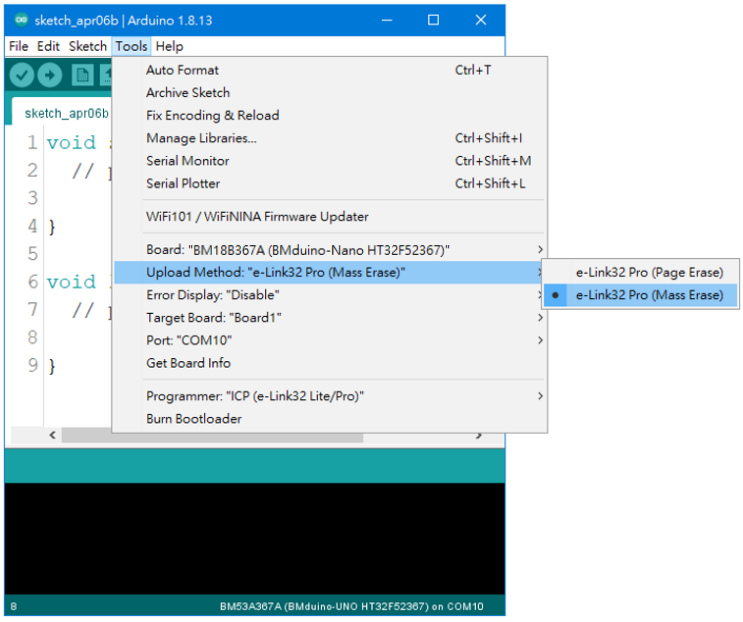
illuminated, remove the e-Link Lite USB and insert it into another USB port on the PC. Restart the PC if the RDY LED has still not illuminated after 10 seconds.

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".



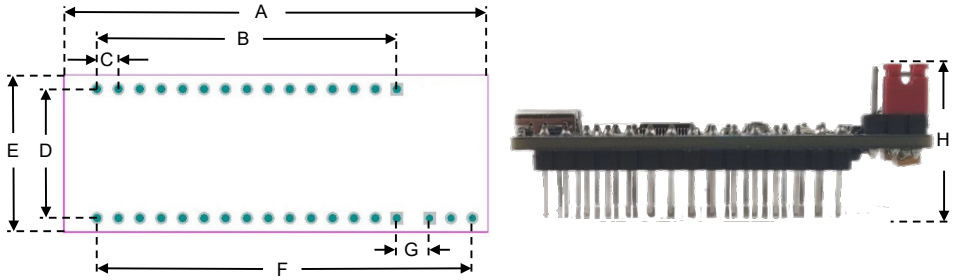
Select Board1

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".



Mass Erase

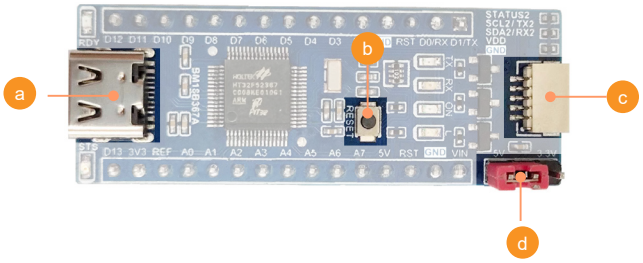
Dimensions



Dimension Information

Unit No.	mm	inch
A	50	1.968
B	35.56	1.4
C	2.54	0.1
D	15.24	0.6
E	18	0.709
F	44.53	1.753
G	3.89	0.153
H	18.4	0.724

Dimension List



Component Size – Height Information

Size No.	Length		Width		Height	
	mm	inch	mm	inch	mm	inch
a	7.65	0.301	8.94	0.352	3	0.118
b	3	0.118	2.5	0.098	1.6	0.063
c	5.2	0.205	7	0.276	2.9	0.114
d	7.62	0.3	2.54	0.1	8.6	0.339

Default height: 11mm / 0.433 inch

Component Size – Height List