Realtek Ameba IoT Solutions

- **Ameba IoT SoC / MCU Solutions**
- **RTL8722DM_MIN Evaluation Board User Manual**

CONTENT

RTL8722DM_MINI

RTL8722DM

More

Ameba D

RTL8722DM_MINI/RTL8722DM/ RTL8721DM/RTL8720DN

– Arm Cortex-M4 @200MHz(Max.)
– Integrated with 802.11a/n Wi-Fi & BLE5.0
– Support Trustzone & Secure boot
– Integrated with Codec/LCDC/Key Matrix
– Up to 4.5MB RAM(Max.)
– Ultra Low Power design

2019 COMPUTEX Best Choice Award

RTL8195AM

More

Ameba 1

RTL8195AM/RTL8711AM/RTL8710AF

– Arm Cortex-M3 @166MHz(Max.)
– Integrated with 802.11n Wi-Fi
– Hardware SSL Engine
– Integrated with NFC tag
– Up to 2.5MB RAM(Max.)
– Ultra Low Power design

2015 COMPUTEX Best Choice Award

Information Platforms

Code Related

- Ameba IoT GitHub

Ameba Resources

- Ameba IoT Website

Demo Videos

- YouTube

Social Media

- Facebook

Asking Questions?

- Forum

RTL8722DM_MINI Evaluation Board User Manual

V1.0 2021/07

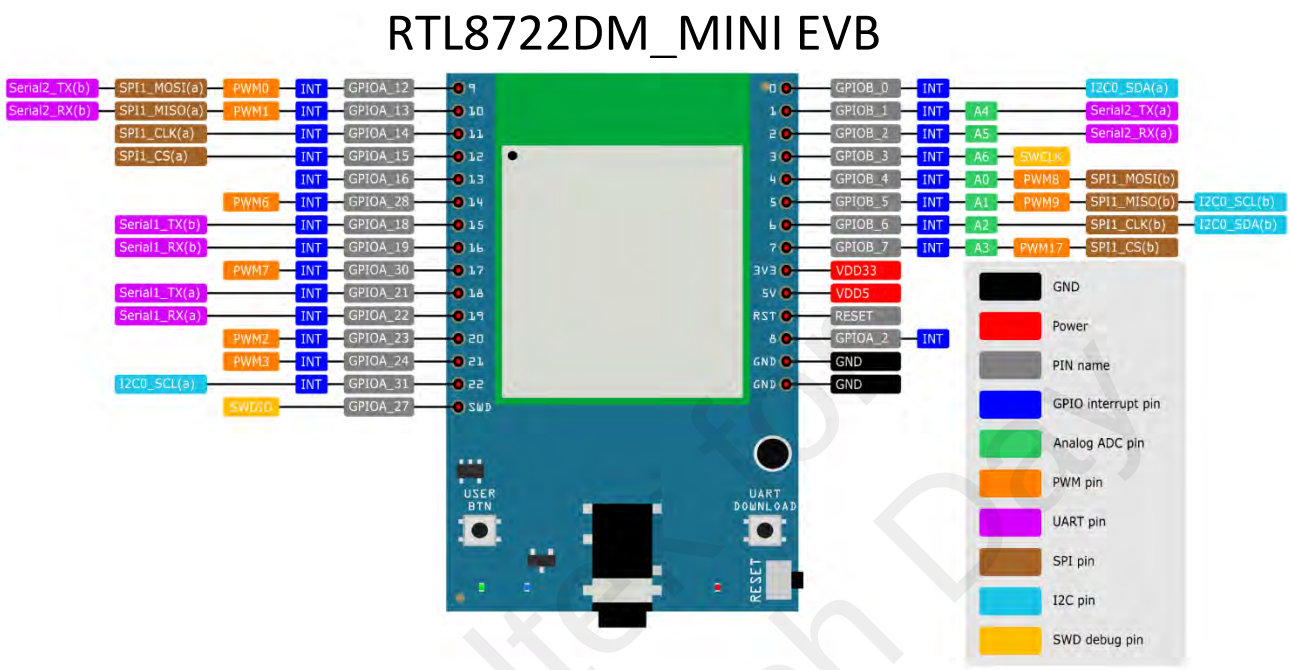
Contents

Hardware Introduction

Pin map

Please refer to the following figure and table for the pin diagram and function of RTL8722DM_MINI.

Picture



Table

	PIN name	GPIO INT	ADC	PWM	UART	SPI	I2C
D0	GPIOB_0	✓					I2C0 SDA
D1	GPIOB_1	✓	A4		Serial2_TX		
D2	GPIOB_2	✓	A5		Serial2_RX		
D3	GPIOB_3	✓	A6				
D4	GPIOB_4	✓	A0	✓			
D5	GPIOB_5	✓	A1	✓			I2C0 SCL
D6	GPIOB_6	✓	A2				I2C0 SDA
D7	GPIOB_7	✓	A3	✓			
D8	GPIOA_2	✓					
D9	GPIOA_12	✓		✓	Serial2_TX	SPI1_MOSI	

D10	**GPIOA_13**	✓		✓	**Serial2_RX**	**SPI1_MISO**	
D11	**GPIOA_14**	✓				**SPI1_CLK**	
D12	**GPIOA_15**	✓				**SPI1_CS**	
D13	**GPIOA_16**	✓					
D14	**GPIOA_28**	✓		✓			
D15	**GPIOA_18**	✓			**Serial1_TX**		
D16	**GPIOA_19**	✓			**Serial1_RX**		
D17	**GPIOA_30**	✓		✓			
D18	**GPIOA_21**	✓			**Serial1_TX**		
D19	**GPIOA_22**	✓			**Serial1_RX**		
D20	**GPIOA_23**	✓		✓			
D21	**GPIOA_24**	✓		✓			
D22	**GPIOA_31**	✓					**I2C0 SCL**

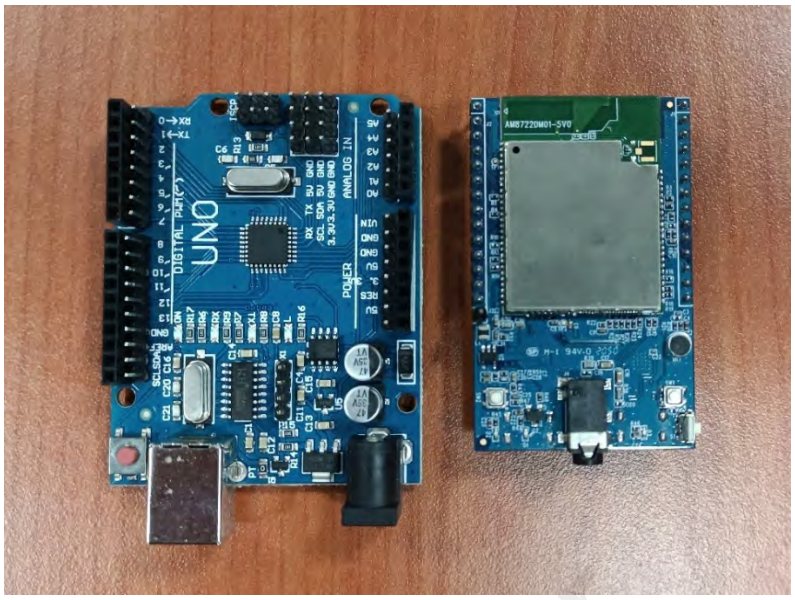
Open-source SDK Getting Start

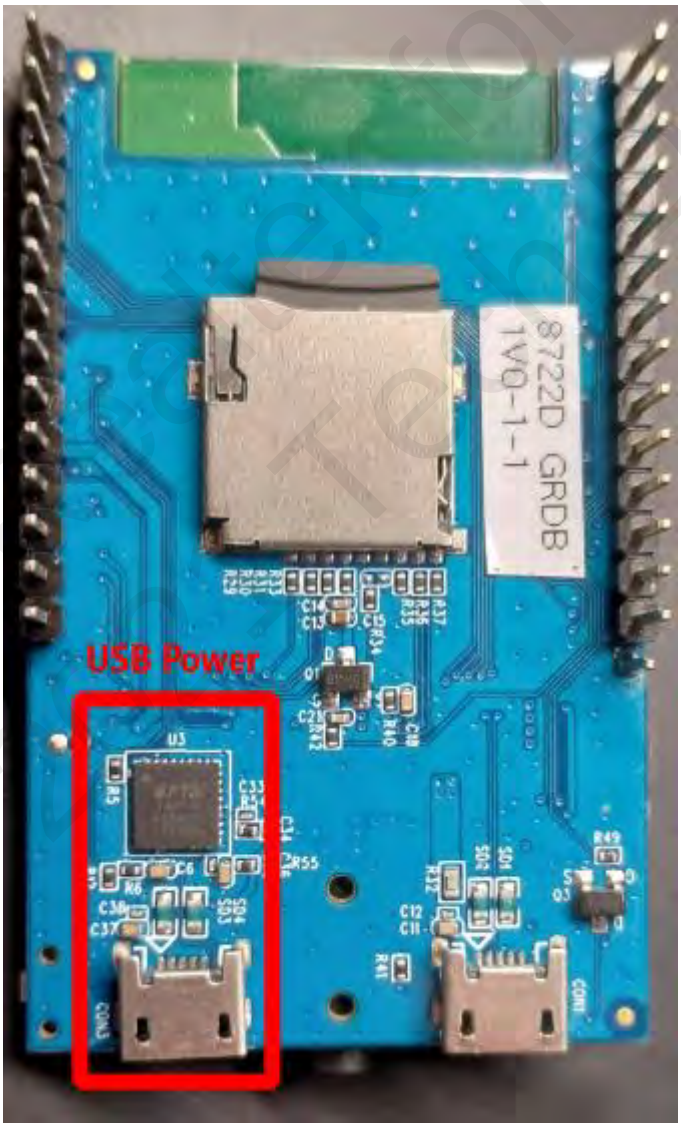
AmebaD RTL8722DM_MINI currently supports Windows XP/7/8/10 32-bits and 64-bits, Linux, and Mac operating systems.

Ameba is an easy-to-program platform for developing all kind of IoT applications. AmebaD is equipped with various peripheral interfaces, including WiFi, GPIO INT, I2C, UART, SPI, PWM, ADC. Through these interfaces, AmebaD can connect with electronic components such as LED, switches, manometer, hygrometer, PM2.5 dust sensors, …etc.

The collected data can be uploaded via WiFi and be utilized by applications on smart devices to realize IoT implementation.

RTL8722DM_MINI has smaller size compared to existing development boards. For example, Arduino Uno.





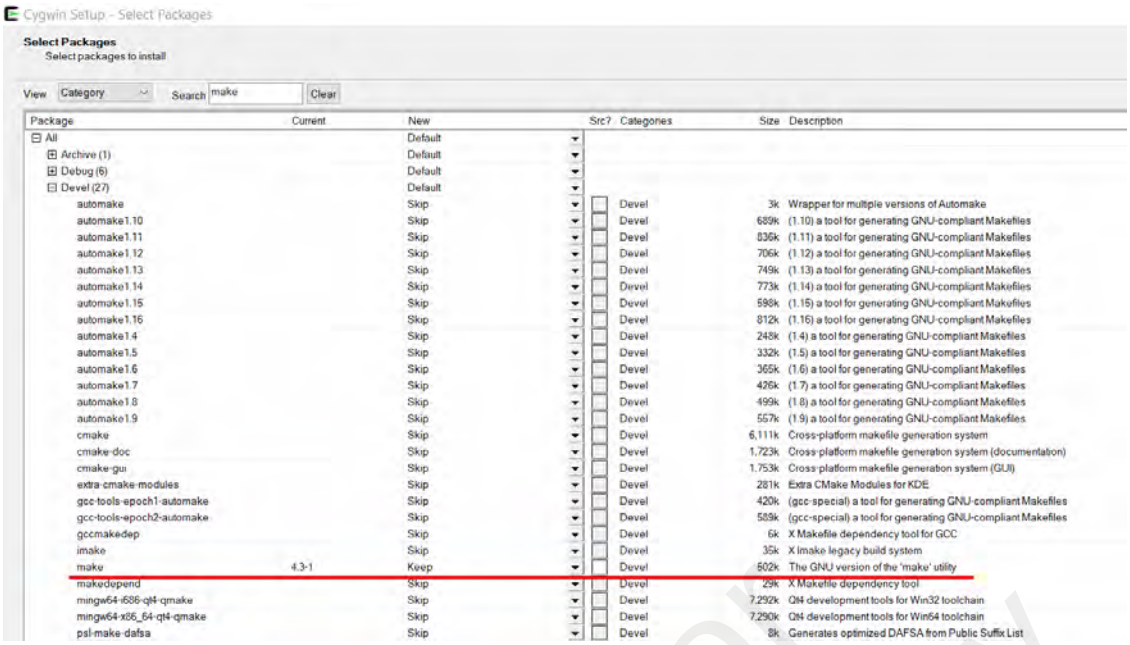
RTL8722DM_MINI uses Micro USB to supply power, which is common in many smart devices.

In order to use RTL8722DM_MINI at any platform, it is necessary to install the driver first.

First, connect RTL8722DM_MINI to the computer via Micro USB(same as power). If this is the first time you connect RTL8722DM_MINI to your computer, the USB driver for RTL8722DM_MINI will be automatic installed. Check the COM port number in Device Manager of computer:

On Windows there is a tool required when using GCC.

For developer who wish to use GCC compiler, then please be sure to install Cygwin, which is a Linux-like environment running on Windows system. When selecting the Cygwin installer, we recommend using the Cygwin 32-bit version. During Cygwin installation, installer will prompt user if wish to install other software, please make sure to select the GNU version of make from the Devel category (see picture below) and pick the latest edition.



Standard SDK

Please download the Standard SDK at https://github.com/ambiot/ambd_sdk/releases

There are 2 ways to development with Standard SDK. GCC project or IAR project. For both projects' developments, there needs 2 steps Code Compilation and Upload Image. For Code Compilation, "lp" and "hp" stands for 2 cores and needs to be compiled. Refer the following simplified steps to process development. For more detail information, please refer to "datasheet" section https://www.amebaiot.com/en/ameba-sdk-summary/.

GCC project

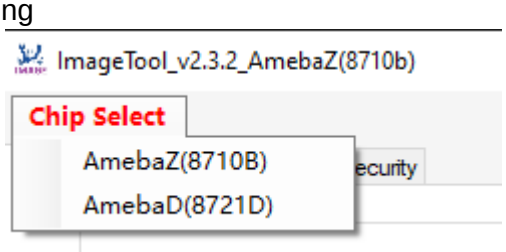
Code Compilation

- Open the Cygwin terminal at "/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp". Then apply commend "make".
- Open the Cygwin terminal at "/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp". Then apply commend "make".

Code Compilation should be done after applied the 2 "make" with no error.

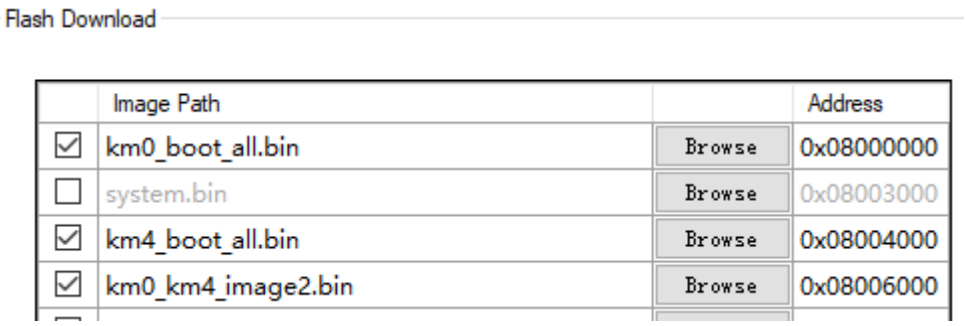
Upload Image

- Open "/tools/AmebaD/Image_Tool/ImageTool.exe"
- Select "AmebaD" as following



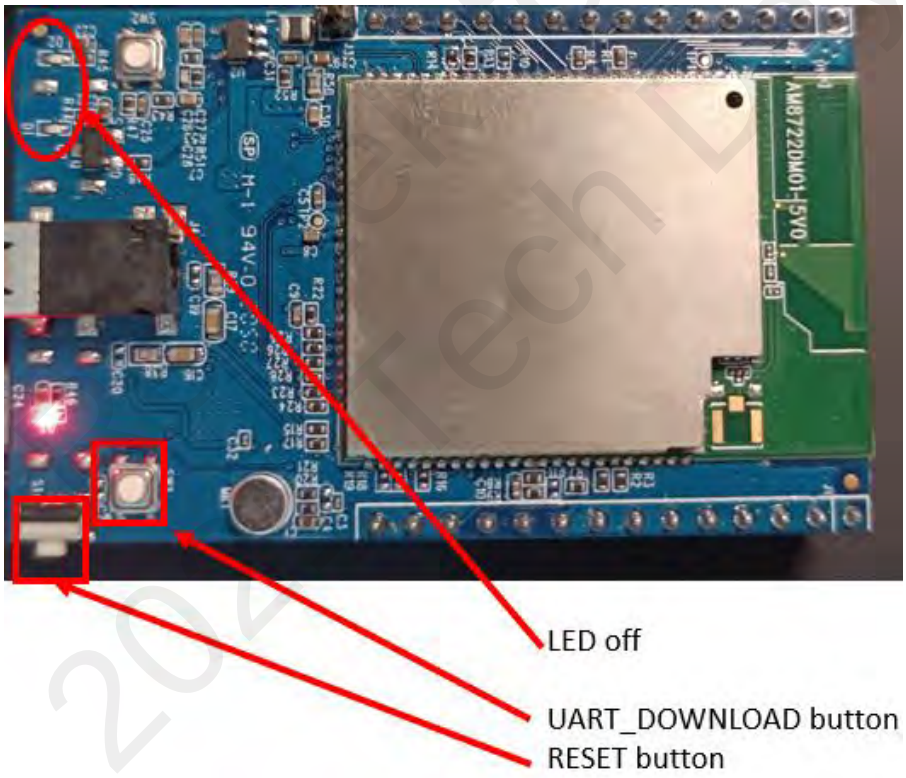
- Set serial information. COM port and Baudrate at 115200.

- Refer to below, set the select the correct bins to be download.

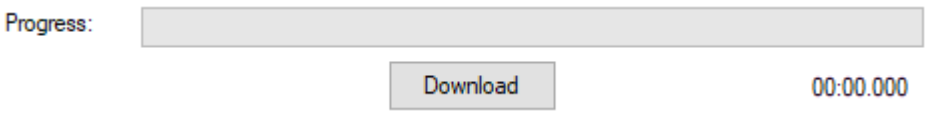


The bin files can be found at "\project\realtek_amebaD_va0_example\GCC-RELEASE\project_hp\asdk\image"

- Enter the download mode.
  First press and hold the UART_DOWNLOAD button, then press the RESET button. If success, you should see the onboard green LED and blue LED all turned off.



- Press the "Download" button.



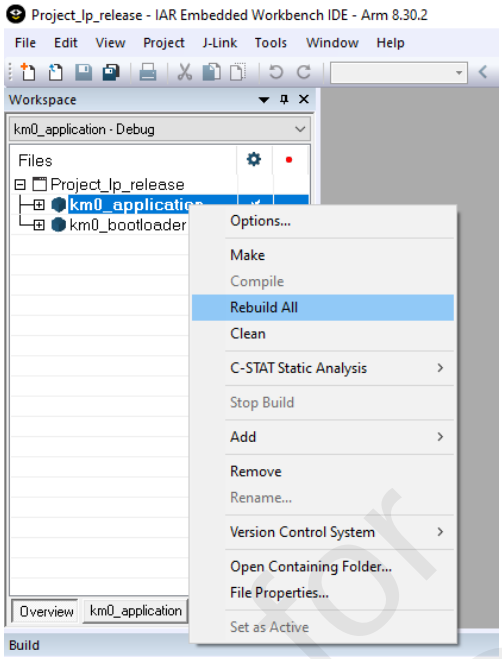
Upload Image should be done after download process finished.

IAR project

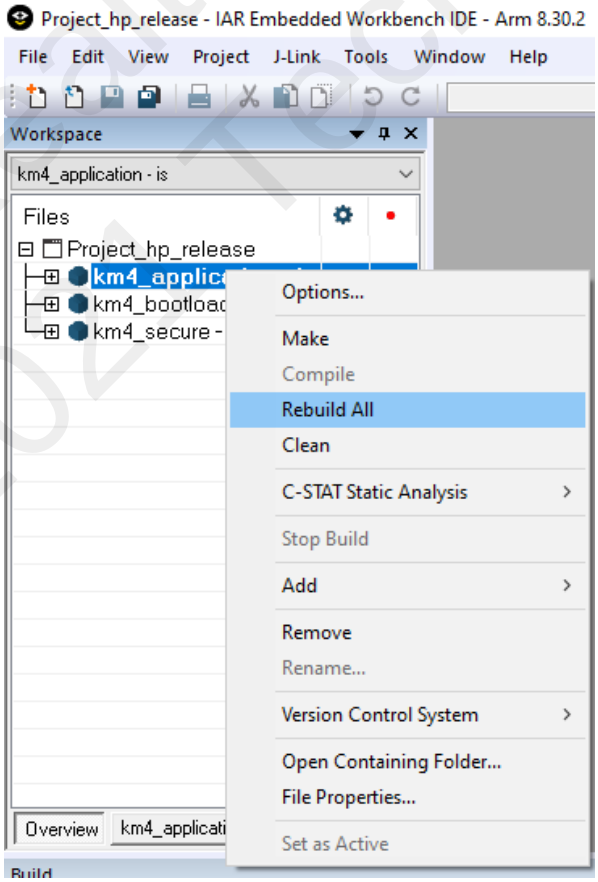
Code Compilation

- Use IAR Embedded Workbench (version should be 8 or above) to open all eww files.

- Open “/project/realtek_amebaD_va0_example/EWARM-RELEASE/Project_lp_release.eww”. Then right click “km0_application” and “Rebuild All”.



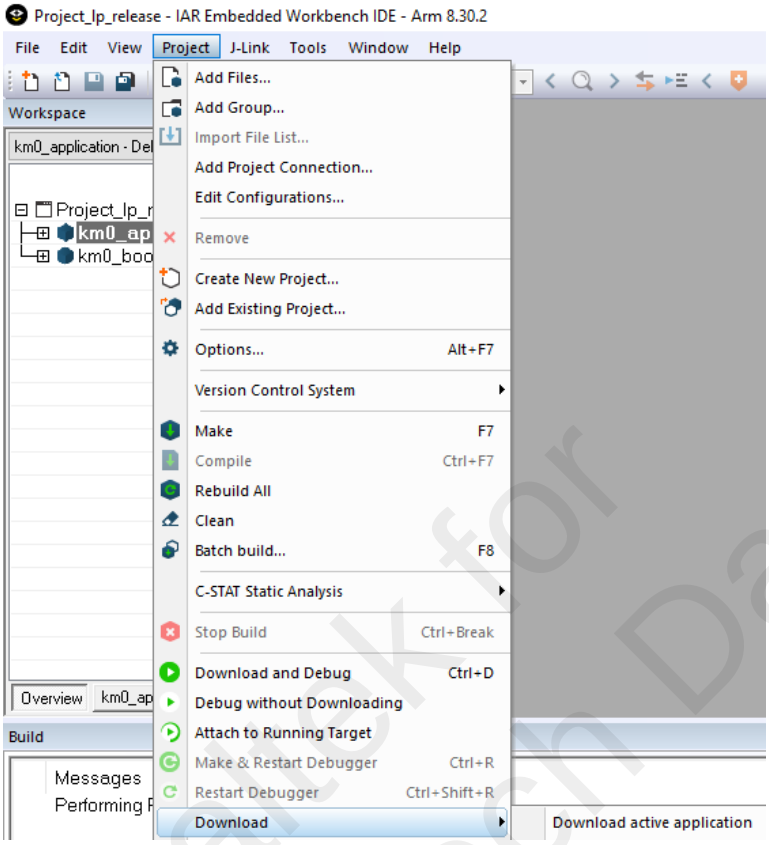
- Open “/project/realtek_amebaD_va0_example/EWARM-RELEASE/Project_lp_release.eww”. Then right click “km4_application” and “Rebuild All”.

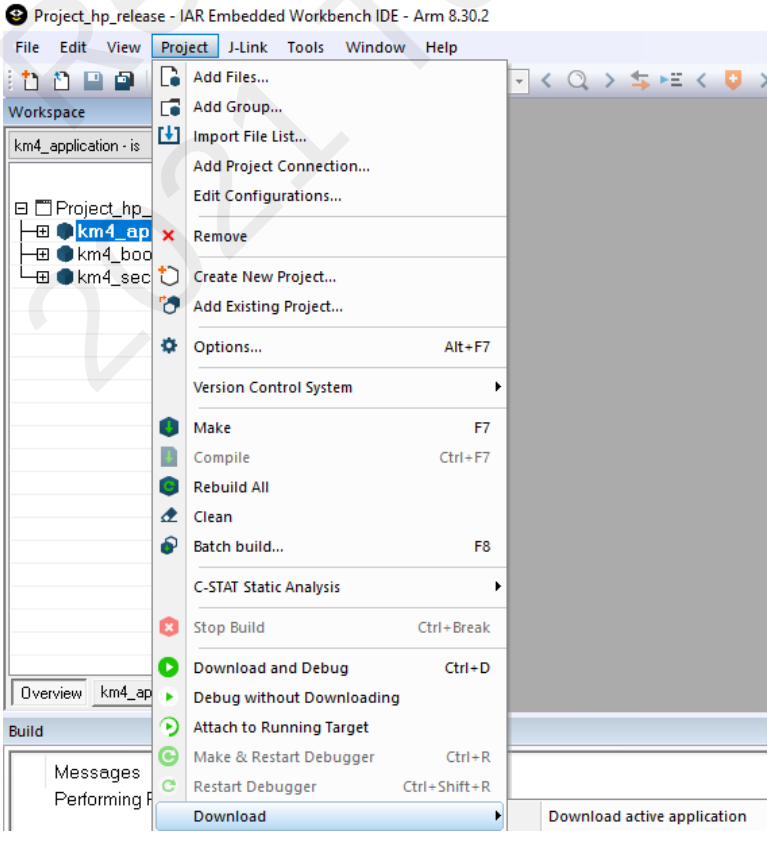


Code Compilation should be done after applied the 2 “Rebuild” with no error.

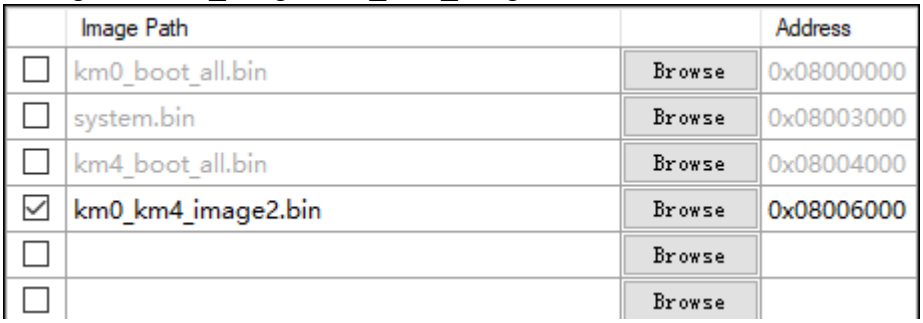
Upload Image

- Connect the Jlink and apply “download” as following. Both “lp” and “hp” use the same way to download images.





- If there is issue using Jlink. The image tool also can be used for download.
- Only upload "\project\realtek_amebaD_va0_example\EWARM-RELEASE\Debug\Exe\km4_image\km0_km4_image2.bin"



Arduino SDK

In this documentation, please use Arduino IDE with version 1.8.15 or later.

From version 1.6.5, Arduino IDE supports third-party hardware. Therefore, we can use Arduino IDE to develop applications on RTL8722DM_MINI, and the examples of Arduino can run on RTL8722DM_MINI too. Refer to basic example table below,

Category	Name	Comment	Remarks
01. Basics	AnalogReadSerial	Connect potentiometer. Reading voltage range 0 to 3.3V.	ADC pin options A0, A1, A2, A3, A4, A5 and A6.
	BareMinimum		
	Blink		Onboard LEDs options LED_B and LED_G. (blue and green)
	DigitalReadSerial		Onboard button PUSH_BTN.
	Fade	Replace "led = 9;" by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. "led = 4;"	
	ReadAnalogVoltage	ADC can read a maximum of 3.3V.	
02. Digital	BlinkWithoutDelay	The onboard blue LED (LED_B) has been used.	Onboard LEDs options LED_G.

	Button		**Onboard LEDs options LED_B and LED_G. Onboard button PUSH_BTN.**
	Debounce		**Onboard LEDs options LED_B and LED_G. Onboard button PUSH_BTN.**
	DigitalInputPullup		**Onboard LEDs options LED_B and LED_G.**
	StateChangeDetection		**Onboard LEDs options LED_B and LED_G. Onboard button PUSH_BTN.**
	toneKeyboard	**Replace "tone(8, notes[thisSensor], 20);" by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. "tone(21, notes[thisSensor], 20);"**	
	toneMelody	**Replace "tone(8, notes[thisSensor], 20);" by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. "tone(21, notes[thisSensor], 20);"**	
	toneMultiple	**Replace "tone(8, notes[thisSensor], 20);" by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. "tone(21, notes[thisSensor], 20);"**	
	tonePitchFollower	**Replace "tone(8, notes[thisSensor], 20);" by a PWM pin (D4,**	

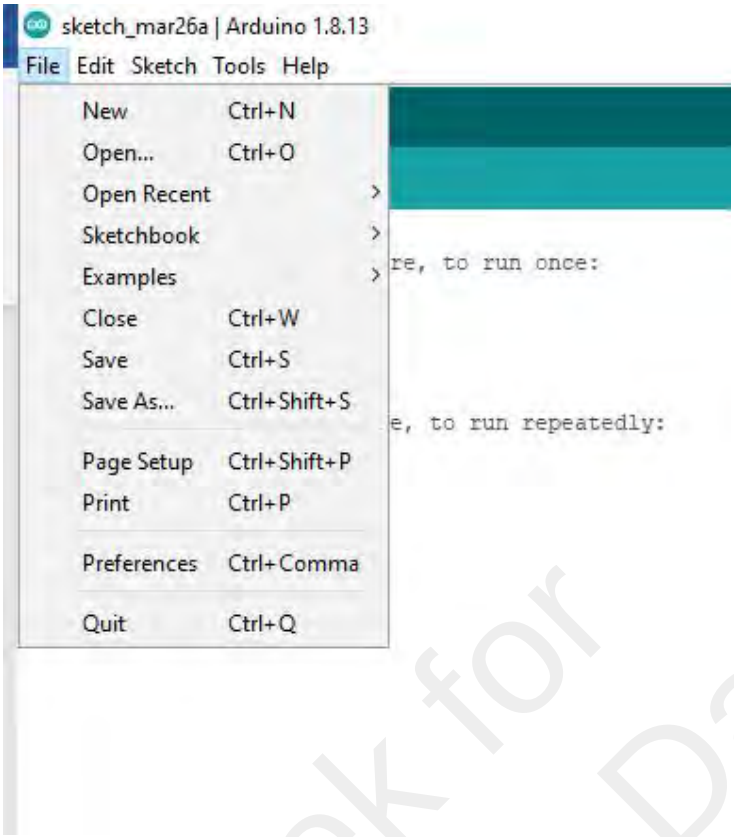
		D5, D7, D12, D13, D14, D17, D20, or D21). e.g. “tone(21, notes[thisSensor], 20);”	
03. Analog	**AnalogInOutSerial**	**Replace “const int analogOutPin = 9;” by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. “const int analogOutPin = 4;**	
	AnalogInput	**Onboard LEDs options LED_B and LED_G.**	
	Analog Write Mega		
	Calibration	**Replace “ledPin = 9;” by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. “ledPin = 4;”**	**Onboard LEDs options LED_B and LED_G. Onboard button PUSH_BTN.**
	Fading	**edPin = 9;” by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. “ledPin = 4;”**	
	Smoothing		
04. Communication	**ASCIITable**		
	Dimmer	**Replace “ledPin = 9;” by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. “ledPin = 4;”**	
	Graph	**Connect potentiometer. Reading voltage range 0 to 3.3V.**	**ADC pin options A0, A1, A2, A3, A4, A5 and A6.**
	Midi	**Please use Serial1 and pin 26, or use Serial2 and pin 17.**	
	MultiSerial		
	PhysicalPixel		**Onboard LEDs options LED_B and LED_G.**

	ReadASCIIString	**Use PWM pin for LED (D4, D5, D7, D12, D13, D14, D17, D20, or D21).**	
	SerialCallResponse		
	SerialCallResponseASCII		
	SerialEvent		
	SerialPassthrough		
	VirtualColorMixer		**ADC pin options A0, A1, A2, A3, A4, A5 and A6.**
05. Control	**Arrays**		
	ForLoopIteration		
	IfStatementConditional		**ADC pin options A0, A1, A2, A3, A4, A5 and A6. Onboard LEDs options LED_B and LED_G.**
	switchCase		
	switchCase2		
	WhileStatementConditional	**Replace "ledPin = 9;" by a PWM pin (D4, D5, D7, D12, D13, D14, D17, D20, or D21). e.g. "ledPin = 4;**	**ADC pin options A0, A1, A2, A3, A4, A5 and A6.**
06. Display	**barGraph**		**ADC pin options A0, A1, A2, A3, A4, A5 and A6. Onboard LEDs options LED_B and LED_G.**
	RowColumnScanning		**ADC pin options A0, A1, A2, A3, A4, A5 and A6.**
07. Strings	**CharacterAnalysis**		
	StringAdditionOperator		

	StringAppendOperator		
	StringCaseChanges		
	StringCharacters		
	StringComparisonOperators		**ADC pin options A0, A1, A2, A3, A4, A5 and A6.**
	StringIndexOf		
	StringLength		
	StringLengthTrim		
	StringReplace		
	StringStartsWithEndsWith		
	StringSubstring		
	StringToInt		

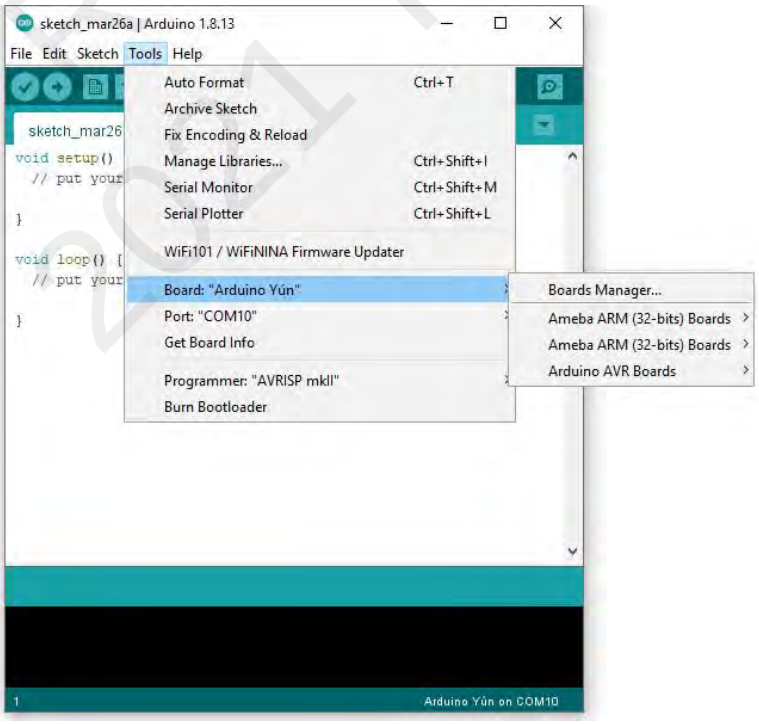
Arduino IDE can be downloaded in the Arduino website: https://www.arduino.cc/en/Main/Software

When the installation is finished, open Arduino IDE. To set up RTL8722DM_MINI correctly in Arduino IDE, go to “File” -> “Preferences”.



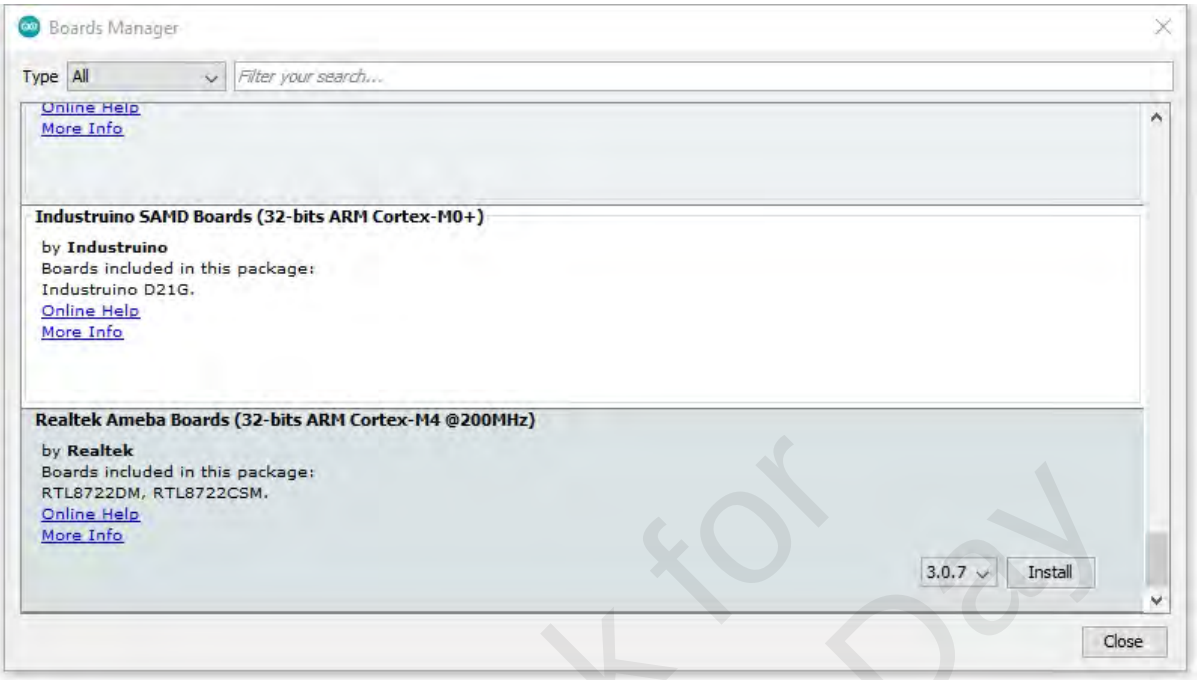
And paste the following URL into “Additional Boards Manager URLs” field:
https://github.com/ambiot/ambd_arduino/raw/master/Arduino_package/package_realtek.com_amebad_index.json

Next, go to “Tools” -> “Board” -> “Boards Manager”:

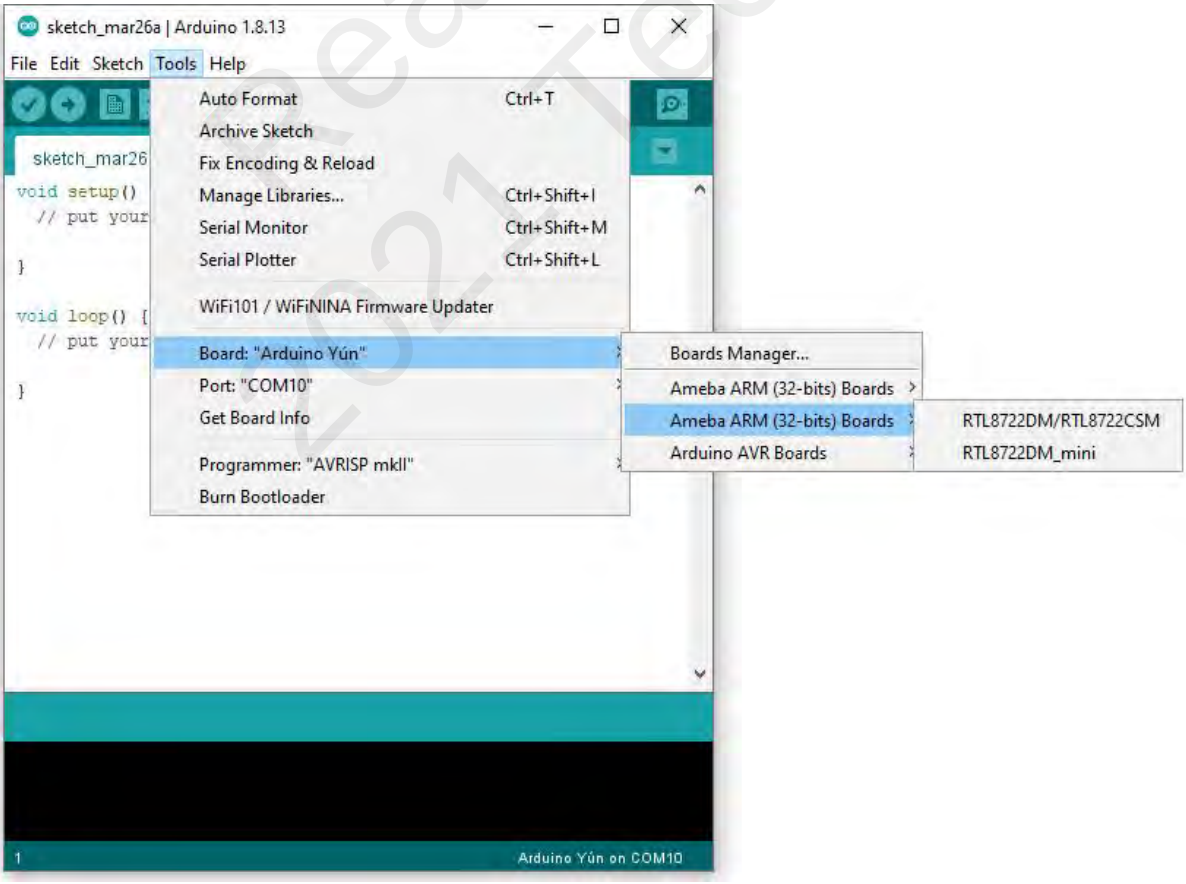


The “Boards Manager” requires about 10~20 seconds to refresh all hardware files (if the network is in bad condition, it may take longer). Every time the new hardware is connected, we need to reopen

the Board Manager. So, we close the Boards Manager, and then open it again. Find “Realtek AmebaD Boards (32-bits ARM Cortex-M4 @200MHz)” in the list, click “Install”, then the Arduino IDE starts to download required files for AmebaD.



Finally, we select AmebaD as current connected board in “tools” -> “Board” -> “Ameba ARM (32-bits) Boards” ->” RTL8722DM_MINI”：



For more information, please refer to https://www.amebaiot.com/en/amebad-mini-arduino-getting-started/

Micropython SDK

REPL stands for Read-Evaluation-Print-Loop, it is an interactive prompt that you can use to access and control your microcontroller.

REPL has been equipped with other powerful features such as tab completion, line editing, auto-indentation, input history and more. It basically functions like the classic Python IDLE but running on microcontroller.

To use REPL, simply open any serial terminal software (most common ones are teraterm, putty etc.) on your PC and connect to your microcontroller's serial port, then set baudrate to 115200 before manually reset the board, then you will see >>> MicroPython prompt appear on the terminal. Now you may type in any Python script on REPL as long as it's support by MicroPython and your microcontroller's MicroPython port.

Most importantly, try to abuse "help()" function as much as possible to gain more information. For example, upon microcontroller power up and REPL shown, just type help(). You will see a help page giving you more details about this port; also, if you type help(modules), it will list out all available builtin modules that are at your disposal.

Furthermore, if you want to learn more about a module, such as its API and CONSTANT available, simply type the following code and details of that module will be returned to you, help(the module of your interest).

Let us take Pin module (GPIO) as an example:

```
>>> help(Pin)
object <class 'Pin'> is of type type
  id --
  init --
  value --
  off --
  on --
  toggle --
  board -- <class 'board'>
  IN -- 0
  OUT -- 1
  PULL_NONE -- 0
  PULL_UP -- 1
  PULL_DOWN -- 2
```

REPL Hotkeys

• Ctrl + d :

Soft reboot MicroPython will perform software reboot, this is useful when your microcontroller is behaving abnormally. This will also run scripts in 'boot.py' once again. Note that this will only reset the MicroPython interpreter not the hardware, all your previously configured hardware will stay the way it is until you manually hard-reset the board.

• Ctrl + e :

Paste mode Paste mode allow you to perform pasting a large trunk of code into REPL at once without executing code line by line. This is useful when you have found a MicroPython library and wish to test it out immediately by copy and paste.

• Ctrl + b :

Normal mode This hotkey will set REPL back to normal mode. This is useful if you are stuck in certain mode and cannot get out.

• Ctrl + c :

Quick cancel This hotkey help you to cancel any input and return a new line.

For different platforms, there are different tools.

On Windows

For windows users, please install a serial terminal software to interact with MicroPython. The most common serial terminals are Tera Term and Putty, here we recommend using Tera Term, which can be downloaded from internet.

Also, Python3 is required during firmware compilation, so be sure to download the latest Python3 from its official website and have it added as environment variable when asked during installation.

On Linux

For Linux user, please install a serial terminal software of your choice using apt-get install command. Here we recommend using picocom for its lightweight.

For advanced developer interested in developing MicroPython module in C, please make sure the GNU make of at least version 3.82 or newer and Python3 are installed and can be found using terminal.

After downloading the MicroPython repository from Github(https://github.com/ambiot/ambd_micropython), at the “Release” folder in the root directory of this repository, enter this folder and locate a tool named “Double-Click-Me-to-Upload”.

Enter UART Download mode. Double click on the file(“Double-Click-Me-to-Upload”) to run it, follow instructions printed on the screen to update the ameba’s serial COM port (this is known to us during the driver installation step mentioned above) so the uploading can be carried out successfully. Once the uploading is successful, you will see a line of log printed on the screen – “All images are sent successfully”.

For more information, please refer to https://www.amebaiot.com/en/amebad-micropython-getting-started/