



MM6108-EKH05 USER GUIDE

Copyright © 2024 Morse Micro

Table of Contents

1 Introduction	4
2 Features	5
3 Development Environment	9
3.1 System Environment	9
3.2 Development Options	9
3.2.1 CMSIS Pack	9
3.2.2 MM IoT SDK	10
3.2.3 PlatformIO + MM IoT SDK	10
4 Getting Started	11
4.1 Default Jumper Configuration	12
4.2 AP Device Settings	13
4.2.1 Changing Channel, Bandwidth, DTIM Period	17
4.3 Software Examples	18
4.4 Viewing MM6108-EKH05 Demo HTTP Server	19
5 Software Development	22
5.1 Installing CMSIS Package	22
5.2 Build and Run Example Applications	25
5.2.1 UART Output	31
5.2.1.1 Windows	31
5.2.1.2 Linux	32
5.2.1.3 Mac OS	32
5.3 Changing Example Application	33
5.4 Changing Example Configurations	35
5.5 Changing Between SPI and SDIO	36
5.6 Changing Network Stacks	43
5.7 Updating BLUENRG-M2SP Module Firmware	45
6 Hardware Layout and Configuration	48
6.1 Power Selection	48
6.2 Using an External Debugger/Programmer	49
6.3 Changing VFEM Voltage	50
6.4 Switching Between SDIO and SPI	51
6.5 Switching Between SMA and U.FL Connector	52
6.6 Disconnecting Sensors	53
7 Power Consumption Measurements	54
7.1 Power Consumption Measurement Points	54

7.1.1 General Structure	54
7.1.2 HaLow and VFEM	55
7.1.3 Whole System Power Consumption	56
7.2 Power Consumption Measurement Procedure	57
7.3 Optional Debug GPIO Connections	58
7.4 Measuring DTIM Power	60
7.5 Measuring WNM Sleep Power	62
8 Troubleshooting	65
8.1 Device Is Not Programming	65
8.2 Poor HaLow Performance	67
8.3 High Power Consumption	68
9 Errata	69
10 Revision History	70
Appendix A: Schematics	71
Appendix B: Mechanical Drawing	79

1 Introduction

The MM6108-EKH05 evaluation kit is a fully integrated Wi-Fi HaLow development platform, designed for a wide range of IoT applications—from smart home devices to industrial automation systems. It features an Azurewave AW-HM593 module, incorporating Morse Micro's MM6108 Wi-Fi HaLow low-power SoC, alongside the STM32U585 low-power microcontroller (MCU) and the BlueNRG-M2 Bluetooth® SoC, the board provides robust wireless connectivity, low power consumption, and an extensive range of programmable interfaces and sensors. This versatile platform is ideal for software engineers developing energy-efficient IoT solutions.

This user guide provides a comprehensive overview of the MM6108-EKH05 evaluation kit, offering step-by-step instructions for setup, configuration, and programming. It covers all aspects of the board's capabilities, including the integration of Wi-Fi HaLow and Bluetooth® connectivity, utilization of the STM32U585 MCU, and leveraging the onboard sensors and interfaces. Detailed examples and practical use cases are provided to help developers quickly prototype and validate IoT applications, ensuring they can harness the full potential of this evaluation kit. The guide also includes troubleshooting tips, optimization techniques for low-power operation, and insights into best practices for IoT system design. Whether the user is new to IoT development or an experienced engineer, this guide is structured to help efficiently create innovative, connected solutions.

Note: At this time, a HaLow access point is <u>required</u> in addition to the MM6108-EKH05. Refer to <u>Section 4</u> for setup details.



2 Features

This section highlights the key features of the MM6108-EKH05 Wi-Fi HaLow evaluation kit, showcasing the advanced capabilities that make it a versatile platform for IoT development.

Key features:

- Wi-Fi HaLow support for long-range, low-power wireless connectivity.
- STM32U585 microcontroller with energy-efficient architecture.
- Integrated programmable interface for flexible development.
- Integrated sensors for enhanced IoT applications.
- Multiple power input sources, including battery connectors.



#	ltem	Description
1.	USB-C Connector	5V power supply. USB data lines connecting to STM32U585 host microcontroller.
2. Micro USB STLINK-V3MODS	MicroUSB	5V power supply. Programming and debugging access to STM32U585 host microcontroller. Access to STM32U585 host microcontroller UART.
	STLINK-V3MODS	Note: The header pins below the STLINK-V3MODS module allow for disconnection from the host microcontroller for low-power testing.
3.	Camera Header	Camera connector to suit a DCMI (Digital Camera Interface) camera module. Connects to DCMI interface on STM32U585 host microcontroller.
		Note: A Camera module is provided in the MM6108-EKH05 kit.
4.	mikroBUS Connector	Standard connector for mikroBUS modules. Interfaces connect to STM32U585 host microcontroller.
5.,	QWIIC Connector	Standard connector for QWIIC modules. Interface connects to STM32U585 host microcontroller.
6.	HaLow Module	Azurewave AW-HM593 module with the MM6108 Morse Micro chipset. Connects to SPI interface (by default) on STM32U585 host microcontroller.
7 SPI/SDIO Selection		Option to populate 0-ohm resistors for either SPI (default) or SDIO operation to STM32U585 host microcontroller.
	Kesistors	Note: Changing this setting also requires software modifications.
8.	U.FL Connector	Antenna output for U.FL connections.

		Note: SMA Connector is set by default. O-ohm resistor change required to use U.FL connector. Instructions available in a chapter below.
0	SMA Connector	Antenna output for SMA connections.
9.	SMA Connector	Note: SMA is the default antenna output.
10.	BLUENRG-M2SP Module	ST BLUENRG-M2SP BLE module for provisioning devices. Connects to SPI interface on STM32U585 host microcontroller.
Module		Note: Access to SPI data lines and SWD data lines through header pins beside the BLUENRG module.
11.	4-20mA Current Sensor Screw Terminal	Connects to an ADC pin on the STM32U585 host microcontroller for 4-20mA operation.
12.	User Button	Programmable button for user applications. Connects to GPIO on STM32U585 host microcontroller.
13.	Status LEDs	Individual red, green, and blue LEDs for user applications. Connects to GPIO on STM32U585 host microcontroller.
14.	Reset Button	Resets the STM32U585 host microcontroller.
15.	STM32U585 host microcontroller	Ultra-low-power MCU with FPU ARM Cortex-M33 MCU with TrustZone, 160 MHz with 2 MBytes of Flash memory.
16.	16MBit SPI Flash	Winbond W25Q16JV SPI flash. Available as extra memory for user applications on STM32U585 host microcontroller
17.	Temperature Humidity Sensor	Sensirion SHT40 Ultra-Low-Power, 16-bit relative humidity and temperature sensor. Connects to I2C Interface on STM32U585 host microcontroller

18.	Accelerometer	ST IIS328DQ Ultra low-power 3-axis accelerometer with digital output. Connects to I2C Interface on STM32U585 host microcontroller.
19.	RGB LED	RGB LED connected to PWM interface on STM32U585 host microcontroller.
20.	Current Measurement Headers	Headers for power measurement. Follows a tree structure allowing measurement of either the entire system or individual blocks.
21.	Power Selection Header	 Header to select system power source. Options: USB power from USB-C or STLINK Battery JST Connector or Battery screw terminals
22.	Battery Screw Terminals	Option for connecting a battery to the system through screw terminals.
23.	JST Battery Connector	Option for connecting a battery to a system through a standard JST connector.

3 Development Environment

3.1 System Environment

All major operating systems are supported:

- Windows 10, Windows 11
- Linux 64-bit
- macOS (macOS 14 and below.)

The development options are described in the following section.

3.2 Development Options

Morse Micro provides three development options for the MM6108-EKH05. The CMSIS Pack with STM32CubeIDE fully supports all hardware and includes pre-loaded example applications for immediate, out-of-box use. The other development options are available but have limited support for the hardware peripherals of the MM6108-EKH05.

3.2.1 CMSIS Pack

A CMSIS (Cortex Microcontroller Software Interface Standard) pack is a standardized software package format defined by ARM to facilitate the development of software for ARM Cortex-based microcontrollers. It bundles together all necessary components, such as device drivers, middleware, libraries, and configuration files, into a single package that can be easily distributed and integrated into development environments. The Morse Micro CMSIS pack includes everything needed to integrate Wi-Fi HaLow applications using this standardized package.

Morse Micro CMSIS pack releases can be found via:

https://www.morsemicro.com/downloads-dashboard/#/software-releases/iot/cmsis-pa ck

The instructions for using this package with STM32CubeIDE as the supported IDE are included in section 5 of this document.

3.2.2 MM IoT SDK

The Morse Micro IoT SDK provides a framework for developing embedded applications that include connectivity provided by Morse Micro MM6108 chip. This SDK package contains the software components to build the application firmware for the STM32 processor. These components include the RTOS (FreeRTOS), network stack (LwIP or FreeRTOS + TCP), Morse Micro host software (morselib), and the platform-specific board support package (BSP).

The MM IoT SDK release packages can be found via: https://www.morsemicro.com/downloads-dashboard/#/software-releases/iot

The documentation for this package is included in each release.

3.2.3 PlatformIO + MM IoT SDK

PlatformIO is an open source multi-platform build system with integration available for various IDEs. It has powerful built-in features such as debugging, unit testing, and static code analysis and it supports a multitude of hardware platforms. PlatformIO also has a vast amount of libraries and example code available for various sensors and actuators used in IoT applications. The MM IoT SDK provides an integration for PlatformIO that supports multiple development boards from ST Microelectronics combined with Morse Micro extension board for developing Wi-Fi HaLow enabled applications.

The user guide for the PlatformIO can be found from the following link on the Morse Micro support portal:

https://www.morsemicro.com/download/ug-platform-io-mm-iot-sdk-user-guide/.

4 Getting Started

The most straightforward setup involves connecting a host PC to both the MM6108-EKH05 as a station (STA), and a separate Wi-Fi HaLow access point (AP). The recommended AP is the MM6108-EKH01-05US evaluation kit, which will be referenced throughout this document.

Note: At this time, the MM6108-EKH05 cannot operate as an AP.

The MM6108-EKH05 connects through the STLINK Micro USB connector, and the AP connects through an Ethernet cable. The Host PC must set the Ethernet interface to a DHCP client - see <u>AP Device Settings</u>. The HaLow connection can either be Over The Air (OTA) through antennas, or cabled. For cabled, it is recommended to use 40 dB of external attenuation as shown below. The device can be damaged if connected directly without attenuation.



4.1 Default Jumper Configuration

By default, the following headers are populated:

- 1. USB power selection header
- 2. All STLINK headers (except for T_VCC)
- 3. Current measurement headers



4.2 AP Device Settings

This section describes how to set up a MM6108-EKH01-05US evaluation kit as an access point (AP). These steps assume the EKH01 AP has been freshly flashed with a <u>OpenWRT 2.6.6 image</u>. For more information, the <u>User Guide</u> for the EKH01 can be found on the Customer Portal.



Navigate to 10.42.0.1 in a web browser. Click Log in.

Select which **Country**, and optionally, there is the ability to set a new hostname and passphrase. Click **Apply** after making changes.

♥ Ø #H01-06b1-LuCl X +			- 0 X
← → ♂ ▲ Not secure 10.42.0.1/cgi-bin/luci/admin/morse/lan	ding		Guest 1
Morse Micro reaching farther			English 日本語 🗙
	Welcome!		
	You can exit now if you'd prefer to configure manually.	devide.	
	HaLow Configuration	US v The models determines the counterflow of each barrier of the Working IP you are device, for details, see the regulatory data table.	
	System Configuration	ekh01-9661	
		Heatmanne is used for many device ld purposes, including DNS.	
	Password	We recommend setting a password. This will protect both the web interface and sah access.	
	Confirmation		
			Apply

Select Standard Wi-Fi HaLow.

	rd		- # ×
Morse Nicro reaching forther			English 日本語 🗙
	Select a Wizard		
	Standard WiFi HaLow	802.11s Mesh	
	EasyMesh		

Select Access Point, and click Next.

• • • • • • • • • • • • • • • • • • •	- ø x
	(③ Guett) : Englishi 日本語 ×
WiFi HaLow Wizard	
This witzard will guide you in setting up your device as a simple Client or Access Point . You can exit now if you prefer to complete your configuration manually.	
This Device	
	Next

Here is where the SSID and Passphrase, Bandwidth and Channel can be set. Default

SSID and passphrases for Morse Micro example applications are:

SSID: MorseMicro

Passphrase: 12345678

After configuring, click **Next**. These parameters can be changed later if needed.

ekh01-96b1 - LuCi X +		- 0 X
← → C ▲ Not secure 10.42.0.1/cgl-bin/luci/admin/morse/wizard		@ Guet I
	This Device	
	SSID MorseMicro Pessphrase 12345678 Operating Frequency With B Mitz Channel 44 (124 MHz) © Available Bandwidths and Channels differ grastly across regions. The higher your bandwidth, the greater the potential throughput of	
	the connection. If you're optioying multiple HisLow access points you may want to select distinct channels and a lower bandwidth to reduce interference.	
Back		Next

Set the Upstream Network to None, and click Next.

Y × 140-1661-1660 € Y	- o x
← → C △ Not server 10.42.0.1/cgi-bin/luci/admin/morsa/wizard	(@ Guet) 1
Morse Micro reaching farther	English 日本語 🗙
Upstream Network	
This Device whole states UP address and run a DHCP server on all interfaces, the Hallow and non-Hallow retenorts	
Back	Next

Click **Apply**.

O ekb01-9651-LuCi X +		- 0 X
Mu Morse Micro reaching further		English E3438 🗙
Ain	most there	
	his Device ADDE-SRUT SEC Merseller SEC Merseller Halene Chert Halene Chert Halene Chert Halene Chert	
• Co • Co	onnect other Hallow-enabled devices to use your new Hallow network. annect Ethernet devices to your network. :::: Apply to persist your configuration.	
Back		Apply

The device is now set up as an AP.

4.2.1 Changing Channel, Bandwidth, DTIM Period

After initial configuration, the channel, bandwidth, and DTIM period can all be changed by navigating to the **Network** \rightarrow **Wireless** page, and then clicking **Edit** on the HaLow radio.



The **Bandwidth** and **Channel** can then be changed from the **Device Configuration** menu. The **DTIM Interval** can then be changed by selecting **Advanced Settings** from the **Interface Configuration** menu.

♥ Ø ekh01-96b1 - LuCi × +			- 0 X
← → ♂ ▲ Not secure 10.42.0.1/ogi-bin/luci/adm	nin/network/wireless		@ Guest I
	Vireless Network: Access Point "MorseMicro" (Wianu)		4ODE: AP REFRESHING English 日本版
M _µ	Device Configuration		
There	General Setup Advanced Settings		
Home Win	Status	Mode/ Master (SSID: MorseMicro BSSD: Oc.BF740.6;461 Encryption: VMA3 SAF (ECMP) — dBm tracket: 44 (924.0;47) Signal: 0 dBm (Moise 0 dBm Bitrate: 0 Moltin) & Country, AU	Co to password configuration
🖉 Wizards	Wireless network is enabled	Obable	Restart Scare Add
 Advanced Config 	Country Code	AU v	Disable Edit Remove
🗈 Status > 🔀 System > Asse	Operating frequency	8 MHz * 44(924 MHz) *	
B Services >	Dynamic Channel Selection	0	RX Rate / TX Rate
di Network 🗸			
Interfaces	Interface Configuration		
Wireless	General Setup Wireless Security Advanced Set	ttings Power Save	Save & Apply • Save
DHCP and DNS	Isolate Clients	0	
Diagnostics		Prevents client-to-client communication	
Firewall	DTIM Interval	40	
		Delivery Traffic Indication Message Interval	
(i) Help	Station Inactivity limit	300	

4.3 Software Examples

The following table will describe the list of supported software examples that are included with the CMSIS pack for the MM6108-EKH05. The "EKH05 Demo" example is loaded on the device out of the box.

Software Example	Description
EKH05 Demo	As the default out-of-box example, the EKH05 Demo showcases the operation of all on-board peripherals of the MM6108-EKH05. A HTTP server runs on the device and displays images from the camera, temperature, humidity and accelerometer values. The BLE module is also activated and acts as a sensor.
Ping	Simple ping demonstration. Also used to stay connected to an AP while idling to measure DTIM sleep currents.
WNM Sleep	Example utilizing WNM sleep to conserve power in between periodic transmissions.
iPerf	Example showing how to perform throughput measurements using iPerf.
AWS IoT	AWS IoT example to demonstrate connecting to AWS Shadow service and demonstrate a simple light bulb example.
Scan	Example application to demonstrate scanning for Wi-Fi HaLow networks.

4.4 Viewing MM6108-EKH05 Demo HTTP Server

To view the HTTP server running on the MM6108-EKH05 device from a PC, the AP will need to bridge the HaLow and Ethernet interfaces together. To do this, first click **Quick Config** from the left hand panel. Scroll down to the **Wireless** section, click the drop down arrow under **Network**, and select **Ian**. Click **Save & Apply** for the changes to take effect.

👻 🌒 ekh01-96b1 - LuCI	×	+													- 0	×
← → C ▲ Not secure	10.42.0.1	l/cgi-bin/luci/admir	n/config											Ŧ	@ Guest	:
					Laptop/Device											
М		Networ	k Interfaces	5											¢	
μĽμ		Name	Forward	Wir	eless	Ethernet	DH	P Server		Protocol			IPv4 address			
		lan	⇒ None	·		eth0 •		D	Static IP		•	10.42.0.1				
🆀 Home		ahwlan	⇒ None	• Morsel	Micro 🕹	None •		0	Static IP		~	192.168.1.	1			
Quick Config			_				_						_		~	ľ
 wvizards 		Wireles	S												ŝ	
 Advanced Config 		Morse M	icro HaLow V	ViFi 802.11ah (ra	dio0)											
🕑 Status	>			Cou	intry AU 🗸											
🔀 System	>			Preferred frequ	ency Width	Channel										
B Services	>				8 MHz	✓ 44 (924 M	1Hz) 🗸									
🚓 Network	>	Enabled	Device	Network	Mod	e 🕖	DPP 🥝	SSI	D/Mesh ID		Encrypti	on	Key/Security			
∠ Statistics	>		wlan0	ahwlan 🔹	Access Point (WD	S)	•	MorseMicro		w	PA3-SAE	~	•••••		Î	
(i) Help	>	+		ahwlan												
□ → Log out				lan												
													Save & Apply 💌	Save	Reset	
								All rights reserv	/ed.							

The MM6108-EKH05 device needs to be restarted at this point to obtain the new DHCP lease. To determine the new IP address it has taken, navigate back to the **Home** sec**uent**ion, and expand the **Local Network** tile.



Here is where to find the DHCP leases the AP has distributed. The first lease given out is the HaLow connection at 10.42.0.225, and the second is the lease given to the host PC.

Lo	cal Network						\times
IPv4 IPv6	10.42.0.1/24 fd6a:cf94:7b45::1/60						
	MAC Address	Hostname	IPv4	Expiry	IPv6	IPv6 Expiry	
	02:00:73:64:4D:79		10.42.0.225	720 min(s)			
	00:E0:4D:6C:88:9A	jono_xps.lan	10.42.0.209	686 min(s)			

Navigate to the IP address of the MM6108-EKH05 (10.42.0.225 in this case) in a web browser to view the contents of the HTTP server.



Important: This will change the HaLow IP address of the AP to be on the same 10.42.0.x IP range. All other example applications such as Ping default to using 192.168.1.1 as the AP IP address. To ensure other example applications still function as expected, the IP address must either be changed in the example applications, or the bridging step can be reversed by selecting **ahwlan** under the **Network** drop down menu. **Save & Apply** must once again be pressed for this change to take effect.



5 Software Development

This section will describe how to install the CMSIS package, how to build and run Morse Micro example applications within the STM32CubeIDE and how to update the firmware on the BLUENRG-M2SP module.

STM32CubeIDE can be downloaded from the ST website - <u>https://www.st.com/en/development-tools/stm32cubeide.html</u>.

The CMSIS package can be downloaded from the Morse Micro customer portal - <u>https://www.morsemicro.com/downloads-dashboard/#/software-releases/iot/cmsis-pa</u> <u>ck</u>.

5.1 Installing CMSIS Package

After Starting STM32CubeIDE, click on **Manage Embedded Software Packages** from the **Help** menu:



Click on From Local ...:

Embedded Softwar	e Packages Manager						×
STM32Cube MCU Packages and embedded software packs releases							
Releases	s Information was last r	efreshed less than o	one hour ago.				T -
RealThread	RoweBots MCU Packages		WES	emotas Embeddedi	port Office		wolfSSL
Stat	us Descri	ption	ectionics	Lupedded	onice	Avail	able Version
•	X-CUBE-R	T-Thread_Nano					
Details							
			-				01
From Local	From Url			Refresh	Install	Remove	<u>Close</u>

Navigate to where the Morse Micro CMSIS package is downloaded, and click **Open**:

MX Select a STM	32Cube Package File	×
Look <u>I</u> n 📒 Do	ownloads 🗸 🛱 🙆	Co 🔠 🗄
MorseMicro	.MM_loT.1.0.0.pack	
File <u>N</u> ame	MorseMicro.MM_lo1.1.0.0.pack	
Files of <u>T</u> ypes	STM32Cube Packages File (*.zip, *.pack)	\sim
	Open	Cancel

Accept the license agreement, and click **Finish**:

MX Licensing Agreement	×
MorseMicro MM_IoT 1.0.0 License Agreement	
Please read and accept the following agreement carefully to finish the installation:	
Morse Micro is licensing this software to you upon the condition that you accept all the terms of the license agreement below ("Agreement") below. By installing this software you accept all the terms of the Agreement.	
"Software" means the software in binary or source code that you have received from Morse Micro or its authorized licensees and/or those portions of such software produced by Programâ€ [™] s code within the Software, as well as any other machine readable materials (including, but not limited to, libraries, source files, header files, and data files), any updates or error corrections provided by Morse Micro, and any user manuals, programming guides and other documentation provided to you by Morse Micro under this Agreement.	r
I have read, and I agree to the terms of this license agreement	
○ I do not accept the terms of this license agreement	
Finish Cancel	

The package is now installed.

5.2 Build and Run Example Applications

After installing the Morse Micro CMSIS package, an example application can be run. As an example, this will run the HaLow Example application.

First, create and log into a myST account. This is needed to install other packages included within the HaLow example.



From File, click Open Projects from File System:



Click on **Directory...**:

Import Projects from File System or Archive					\times
Import Projects from File System or Archive This wizard analyzes the content of your folder or archive file to find projects and im	port them in the IDE.				5
Import source:		~	Directory	Archive.	
type filter text			Sele	ect All	
Folder	Import as		Dese	lect All	
 Close newly imported projects upon completion Use installed project configurators to: Search for nested projects Detect and configure project natures Working sets 			0 of 0 selected	y open proj	jects
Add project to working sets				New	
Working sets:			~	Select	
		Show	other specialize	d import wi	<u>izards</u>
0	< Back	Next >	Finish	Cance	el 🛛

Navigate to the default STM32 package location, and then find the HaLow_example application.

On Windows, the default path is:

```
C:\Users\USER_NAME\STM32Cube\Repository\Packs\MorseMicro\MM_IoT\1.3.3\H aLow_example
```

On Linux, the default path is:

```
/home/USER_NAME/STM32Cube/Repository/Packs/MorseMicro/MM_IoT/1.3.3/HaLo
w_example
```

On Mac, the default path is:

```
/Users/USER_NAME/STM32Cube/Repository/Packs/MorseMicro/MM_IoT/1.3.3/HaL
ow_example
```

Note that the version number may change depending on which version of the CMSIS pack is downloaded.

IDE Browse for Folder							×
\leftrightarrow \rightarrow \checkmark \uparrow	«Packs > I	MorseMicro > MM_	_loT > 1.0.0 > Hal	ow_example >	~ C	Search HaLow_exan	nple ,0
Organize 👻 New folder							≣ • 🕐
> 🦲 OneDrive - Personal		Name	^	Date modified	Туре	Size	
		Core		18/09/2024 11:18	AM File folder		
E Desktop	*						
🚽 Downloads	*						
Documents	*						
Pictures	*						
📒 repos	*						
🚞 repos	*						
늘 Hardware Designs	*						
🚞 mm6108-ekh05	*						
🗖 rpi pico 2 hat	*						
Folder:	HaLow_exa	mple					
						Select Folder	Cancel

Click Finish:

IDE Import Projects from File System or Archive		– D X
Import Projects from File System or Archive This wizard analyzes the content of your folder or archive file to find projects and impor	t them in the IDE.	
Import source: C:\Users\JonathanGarnier\STM32Cube\Repository\Packs\MorseMicro	o\MM_loT\1.0.0\HaLow_example v	Directory Archive
type filter text		Select All
Folder HaLow_example	Import as STM32CubelDE project	Deselect All
Close newly imported projects upon completion		1 of 1 selected Hide already open projects
Use installed project comparators to: Search for nested projects Compared project and configure project natures		
Working sets		
Add project to working sets		New
Working sets:		∨ Select
	<u>Sh</u>	ow other specialized import wizards
0	< Back Next >	Finish Cancel

repos - STM32CubeIDE IDE <u>N</u>avigate Refactor Se<u>a</u>rch Edit Source File Pr - 🔜 🞺 **N**3 0: Project Explorer 🗙 F 8 Interpretation Halow_example > 🔊 Includes 冯 Core HaLow_example.ioc SH.Id T STM32U585VITXQ_RAM.Id

Expand the Halow_example, and double click on the .ioc file:

Click Continue.



There may be additional software packs that need to be installed including STMicroelectonics.X-CUBE-FREERTOS.1.2.0, and STMicroelectronics.X-CUBE-BLE2.3.3.0. Download these items in the subsequent pop-up windows.

Click **Project**, **Generate Code**. This may also start a download of the STM32U5 firmware if this is not downloaded already.



The example can then be built by clicking the hammer icon in the toolbar. If successful, there should be no errors in the Console window:



The program can now be run by clicking on the green Run icon:

IDE re	epos -	HaLow_ex	ample/Cor	e/Src/main	.c - STM	32CubeIDE					
File	Edit	Source	Refactor	Navigate	Search	Project	Run	Window	Help	1	
1	•	0 8	- 🔨 -	🔜 i 🞺 (۵ (۵ 🍐	- 🚳	• 💣 •	ଙ -	₩ 0 •	• 😂 🔗 • 🌛
P P	roject	Explorer	×	-		MX HaLow	examp	le.ioc	🖻 mai	n.c 🔨 Kuirriad	ow_example
				🖻 🛠 5	7 8	1 /	* USH	ER CODE	BEGIN	Header */	

This will then bring up a window for the Run Configuration. All settings can be left as default. Click **Ok**:

dit launch configuration properties				
Jame: HaLow_example				
🗎 Main 🕸 Debugger 🐌 Startup 🖏 Source 📰 Com	mon			
Project:				
HaLow_example				Browse
C/C++ Application:				
Debug/HaLow_example.elf		Search Project	Workspace	Browse
Build (if required) before launching				
Build Configuration: Select Automatically				~
○ Enable auto build	O Disable auto	build		
Use workspace settings	Configure Work	space Settings		

The example program will now build again if not built already, and then be flashed to the device.

5.2.1 UART Output

The UART output can be monitored by opening a serial monitor with the following settings:

ltem	Setting
Baud Rate	115200
Hardware Flow Control	None
Data Bits	8
Stop Bits	1
Parity	None

5.2.1.1 Windows

The STLINK-V3MODS will show up as a Virtual COM port on windows machines.



This COM port can then be opened with tools such as PuTTY.

putty -serial COM10 -sercfg 115200

5.2.1.2 Linux

On Linux machines, the device can be detected with the following command:

```
ls /dev/serial/by-id/usb-STMicroelectronics_STLINK-V3_*
```

And a serial terminal can be opened with tools such as minicom:

```
minicom -D /dev/serial/by-id/usb-STMicroelectronics_STLINK-V3_*
-b 115200
```

5.2.1.3 Mac OS

On Mac machines, the device can be detected with the following command:

ls /dev/tty.*

And similarly, a tool such as minicom can be used to open the serial terminal:

minicom -D /dev/tty<YOUR-DEVICE-HERE> -b 115200

5.3 Changing Example Application

To change the example application, open the HaLow_example.ioc file and click **Select Components** from the **Software Packs** menu.

File Edit Navigate Search Project	it Run Window Help 🎩 myST		
🔁 = 🗟 🖏 🕹 = 🇞 = 🗎 🧰	11111111111111111111111111111111111111		Q, 📰 🖬 🛄 🏘
💫 Project Explorer 🗙 👘 🗖	HaLow_example.ioc X main.c		- 0
E 🕸 🖓 !	Pinout & Con	guration 🗲 LPBAM Scenario & Configuration 🔰	CHECK LPBAM DESIGN
HaLow_example Second	Direct & Confer	Olask Cashermation Design Harran	Taala
> 😥 Includes	Pinout & Configu	ation Oject Manager	LOOIS
> 25 BlueNRG-2		 Software Packs Pinout 	
> 😅 Drivers	<u>م</u> ا	Manage Software Package Compositive System view	
> 🥶 MM_IoT	Categories Ar>2	Add pack software component to the project	1
> 😂 Debug	System Core >	96 O M	
> BSP-EKH05	Andre)		
HaLow_example.ioc			
STM32U585VITXQ_FLASHJ	Timers >		
STMB2U585VITXQ_RAM.Jd	Connectivity >		
	Multimedia		
	Security >		
	Computing	SAME OF A CONTRACT	
	Company		
	Middleware and Software Pac>	M 0000,00 M 0000,00	
	Trace and Debug		
		Note that the second se	
	Power and Thermal 2	100_00007_01 (ext	
	Utilities >		
		STM32U585VITxQ	
		ange ange ange ange ange ange ange ange	
		1860_01_01_0 100 100 100 100 100 100 100 10	
		and a	
		f fan Gwei	
		- 8 4	
		Q [] Q 🕒 🗐 🗏 Q 🛶	

Click on the MorseMicro.MM_IoT arrow, and then select the desired application from the drop down list.



For some example applications, there may need to be an extra component added. There will be a yellow warning symbol in the **Status** column. Clicking on the yellow warning symbol will bring up a menu detailing the missing component. Click on the **Resolve** button to add the component, and then click **Ok**.

Software Packs Component Selector						
Packs						
🗮 💊 🕕 >						
Pack / Bundle / Component	Status	Version	Selection			Ę
> EmbeddedOffice.I-CUBE-FS-RTOS		1.0.1	🔓 Install			
> ITTIA_DB.I-CUBE-ITTIADB		8.8.0 😐	🔒 Install			
> Infineon.AIROC-Wi-Fi-Bluetooth-STM32		1.5.1 😐	🚣 Install			
✓ MorseMicro.MM_IoT	A	1.0.0				
> MM_IOT MM_IOT	\odot	2.5.2				
> AWS AWS-libraries		2.5.2				
> FreeRTOS-Libs FreeRTOS-libraries	171	2.5.2				
MM_IoT Example	A	.5.2	ping 🗸 🗸			
> RealThread.X-CUBE-RT-Thread_Nano	1	4.1.1 🖻	an moran	•		
> SEGGER.I-CUBE-embOS		1.3.1 👜 😂	🚣 Install			
> STMicroelectronics.FP-ATR-ASTRA1		2.0.0 🖻	🚣 Install			
> STMicroelectronics.FP-ATR-SIGFOX1		3.2.0 🕒	🚣 Install			
> STMicroelectronics.FP-SNS-FLIGHT1	Ø	5.0.2 🖻	🚣 Install			
> STMicroelectronics.FP-SNS-MOTENV1	Ø	5.0.0 ڬ 🗸 🗸	📥 Install			
> STMicroelectronics.FP-SNS-MOTENVWB1		1.4.0 🖢 🗸 🗸	🐴 Install			
> STMicroelectronics.FP-SNS-SMARTAG2		1.2.0 🕒	🚣 Install			
Component dependencies						
✓ MM_IoT Example					Show	Resolve
Requires: component class MM_IoT, group	mmping					🔺 Missing
 Solutions in MorseMicro.MM_IoT.1.0.0: 						
Component mmping					Show	Select
l					Ok	Cancel
					Ok	Cancel

Click **Generate Code** from the **Project** menu again. The **Run** button can then be clicked again to build and run the new example.

5.4 Changing Example Configurations

Parameters like country code, SSID, passphrase, IP addresses, ping count etc can be changed by navigating to the **Core** \rightarrow **Src** \rightarrow **configs.c** file. Replace the parameter and save the file. The changes will take effect upon the next build of the firmware.

IDE repos - HaLow_example/Core/Src/configs.c - S	TM32CubeIDE
File Edit Source Refactor Navigate Search	:h Project Run Window Help 💄 myST
📑 🕶 🔚 🐚 🕲 🕶 🗞 🕶 🔜 🖉 🗠 🖉	. ∲ @ ▼ & ▼ C ▼ C ▼ C ▼ ☆ ▼ O ▼ 9₄ ▼ 参 タ ▼ 🕖 ≅ 🗐 11 ∲ ▼ ∛ ▼ ♡
🏠 Project Explorer 🗙 🗖 🗖	HaLow_example.ioc 🖻 main.c 🗟 configs.c 🗙
E 😫 🍸 🖇	10/*
V DE HaLow example	2 * Copyright 2021-2023 Morse Micro
> 🖑 Binaries	3 *
> 🔊 Includes	4 * This file is licensed under terms that can be found in the LIC
> 🔑 BlueNRG-2	5 * directory of the Morse Micro IoT SDK software package.
V 📇 Core	6 */
> 🍋 Inc	7 0. Hissinds Human film bill
V 🍋 Src	a #include "mmCOnfig.n"
Contraction of the state of the	3 10 config entry t mm confige[]=/
> configs.c	10 configenciation configence */
.c	12 {"iperf.mode", "udp server"}.
> 🖸 main.c	13 /* IP address of server to connect to when in client mode
> c stm32u5xx_hal_msp.c	14 {"iperf.server", "192.168.1.1"},
> c stm32u5xx_hal_timebase_tim.c	15 /* Specifies the port to listen on in server mode. */
> 💽 stm32u5xx_it.c	<pre>16 {"iperf.port", "5001"},</pre>
> 尾 syscalls.c	170 /* Duration for client transfers specified either in seco
> 🖻 sysmem.c	18 * negative, it specifies a time in seconds; if positive,
> 🖻 system_stm32u5xx.c	<pre>19 * of bytes to transmit. */</pre>
> 🗁 Startup	20 {"iperf.amount", "-10"},
> 🐸 Drivers	
> 😂 MM_loT	22 {"ping.target", "192.168.1.1"},
> 🔑 Middlewares	23 {"ping.count", "10"}, 24 {"ping.interval", "1000"}
> 📂 Debug	25 ("ping.size", "56").
> 🧀 BSP-EKH05	26
Abstract.txt	27 /* The WiFi SSID */
HaLow_example.ioc	<pre>28 {"wlan.ssid", "MorseMicro"},</pre>
HaLow_example.launch	29 /* The WiFi password, not required if wlan.security is op
STM32U585VITXQ_FLASH.Id	<pre>30 {"wlan.password", "12345678"},</pre>
STM32U585VITXQ_RAM.Id	31 /* The WiFi security to use, valid values are sae, owe an
	<pre>32 {"wlan.security", "sae"},</pre>
	33 /* The 2 letter country code to ensure the correct regula
	<pre>34 {"wlan.country_code","AU"},</pre>
	35
	36 /* If true use DHCP, else the static IP configuration wil
	•
	■ Console ×
	<terminated> HaLow_example [STM32 C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Sep 1</terminated>

5.5 Changing Between SPI and SDIO

This section will document how to change between using SPI and SDIO as the communication protocol between the STM32U585 and the HaLow module. Note that hardware changes are also required to function properly. These changes are described in section 6.4.

Open the HaLow_example.ioc file and click **Select Components** from the **Software Packs** menu.



Select the **MM_IoT** and **BSP-EKH05** drop down menus, and then select the desired communication protocol from the **MM6108** drop down box. When using SDIO, there will be a yellow warning label next to morselib. This can be ignored.

cks				
÷ • • •				
Pack / Bundle / Component	Status	Version	Selection	
> EmbeddedOfficeJ-CUBE-FS-RTOS		1.0.1 🗎	👗 install	
> ITTIA_DB.I-CUBE-ITTIADB		8.8.0 🖻	🚣 install	
> Infineon.AIROC-Wi-Fi-Bluetooth-STM32		1.5.1 🖻	👗 Install	
V MorseMicro.MM_IoT	A	1.3.1		
✓ MM_IoT MM_IoT	A	2.5.2		
morselib	A	2.5.2	1	
mmconfig	\odot	2.5.2	S	
mmipal	\odot	2.5.2	V	
mmutils	0	2.5.2	V	
mmLwIP	\odot	2.2.0	.	
mmping	\odot	2.5.2		
mmiperf	0	2.5.2		
mbedtis		3.4.0		
mm-tinycbor		0.6.0		
V BSP-EKH05	0			
BSP	0	1.3.1		
SHT4x	0	1.1.0		
IIS328DQ	0	2.0.0		
OV5640	0	4.0.2		
MM6108	0	252	SDIO V	
> AWS AWS-libraries		1.3.0		
> EmeRTOS Libs EreeRTOS libraries		130		
	0	25.2	inorf v	
> DealThread V.CURE DT.Thread Nano	0	4.1.1 m	lostal	
> Rearriead_A-GODRT-Titlead_Nano		1210	- Install	
> SEGGERI-COBE-embos	-	1.3.1	an mistal	
STMICroelectronics.FP-ATR-ASTRAT	2.9	2.0.0	🚜 install	
Software Packs Pinul Software Packs Software Packs Pinul Software Packs Software Packs Pinul Software Pack Software Pack Pinul Software Pack Software Pack Pinul Software Pack Mode Mode Pinul Mode Mode Pinul Pinul Pinul Software Packs Mode Pinul Pinul Pinul Pinul Software Pack Mode Mode Pinul	Pinout & Configuration	Clock Configuration	Project Manager	Tools
--	------------------------	-------------------------------	--	---
SUMMC2 SUMMC2 SUMMC2 Mode Preset view P Preset		✓ Software Packs	✓ Pinout	
Mode Mode <td< th=""><th> ©</th><th>SDMMC2 Mode and Configuration</th><th>Pinout view</th><th>System view</th></td<>	©	SDMMC2 Mode and Configuration	Pinout view	System view
Optimized Image	egories A->Z	Mode	SCL SCL	
shalling > times > Connectivity • PEANI • Construction • PEANI • Construction • PEANI • Construction • Construction <th>Mode,D</th> <th>isable</th> <th>RESE 2C1_ 2C1_ 2C1_ 2C1_ 2CM_</th> <th>SPL_N BRNG SPI3_ SPI3_ SPI3_ DEBU</th>	Mode,D	isable	RESE 2C1_ 2C1_ 2C1_ 2C1_ 2CM_	SPL_N BRNG SPI3_ SPI3_ SPI3_ DEBU
Image: Signed	nalog >			- 33333
Piccuri Piccuri <t< td=""><td>mers ></td><td></td><td></td><td>P01 P01 P02 P03 P03 P03 P03 P03 P03 P03 P03 P03 P03</td></t<>	mers >			P01 P01 P02 P03
PDDWI PEDWI PEDWI <td< td=""><td>onnectivity ~</td><td></td><td>PE2 PE3 Reset_Stat</td><td></td></td<>	onnectivity ~		PE2 PE3 Reset_Stat	
FDCAN1 PEC DOAL (1)2 VOUL I2C1 I2C2 DEBUG_/IRST PLA DEBUG_/IRST	•		PE3 COMP2 IN	P VSS
Incide PES Decognition PA1 PA1 1222 1223	FDCAN1		PE4 DCMI_D12	VDDU
Inclusion Inclu	12C1		PE5 I2C3_SDA	PA13. DEBUG_
LIC4 INTU Configuration PLAN I	12C2		PE6 LPTIM1_CH	12 PA12
Configuration Configuration Configuration Configuration Configuration Configuration Configuration PCL PCL PCL PCL PCL PCL PCL PC	12C4		PC12	PATT SPARE_C
OCTOSPH PR05 PR05 PR05 PR06 DC SSMMMC1 YSS YSS PR06 DC PR06 DC SSM1 SSM2 YSS PR06 DC PR07 DC PR07 DC PR06 DC PR07 DC PR06 DC PR07	LPUART1	Configuration	PC14-	PAG MM DEB
> SOMAC1 YSS PCS DC PC PC DC PC PC DC PC	OCTOSPI1 OCTOSPI2		PC15	PAB CAMERA
SPI VOD PC2 DC SPI2 VOD PC2 DC N SPI3 VOD PC2 DC V XATTA PHA PHA PC2 DC V XATTA PHA PHA PC2 DC V XATTA PHA PHA PC2 PC3 PC4 DC V XATTA PHA PE3 PC4 PC5 PC5 PC4 PC5 PC5 PC4 PC5 PC5 PC5 PC5 PC4 PC5 PC5 PC4 PC5 PC5 PC5 PC4 PC5 PC5 PC5 PC5 PC4 PC5 PC6 PC5 PC6 </td <td>SDMMC1</td> <td></td> <td>VSS</td> <td>PC9 DCMLD3</td>	SDMMC1		VSS	PC9 DCMLD3
SP2 P40 P40 LUARTA P40 Pcc Pcc LUARTA Pcc Pcc Pcc Pcc LUARTA Pcc Pcc Pcc Pcc Pcc LUARTA Pcc	SPI1			PCB DCMI_D2
UART3 P01. P01. <t< td=""><td>SPI2 SPI3</td><td></td><td>PHO</td><td>PC7 DCML_D1</td></t<>	SPI2 SPI3		PHO	PC7 DCML_D1
VCPD1 VCPD1 VCANT2 USANT2 USANT2 USANT2 USANT2 USANT2 Street Provide Action of the second sec	UART4		PH1	PC6 DCMI_D0
USART USART USART USART USARTS	UCPD1		NRST	PD15 SPARE_G
USART3 DUSE_OTG_FS PC2 PC2 PC3	USART1 USART2		PCI	
	USART3		PC2	PD12 - RGB LFC
	JUSE_OIG_FS		Q [] Q 🗅 🖆	

Click on the **PB4** pin and click **Reset State**.

Click on the **PB15** pin and click **Reset State**.





Click on **Connectivity**, and **SPI2** from the left hand panel. Select, **Disable**.

Click on **System Core**, and **GPDMA1** from the left-hand panel. Select, **Disable** for channels 14 and 15.

repos - Device Configuration Tool - STM32CubelD	DI						
Eile Edit Navigate Search Project Bun W	indow Help 🌡 myST						
[] • 🗟 🕲 • 🐐 • 🔒 🗇 🗅 🍐 🕸 • O	 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	C 0					
🗞 Project Explorer × 🛛 🖻 🛸 🐨 🔋 💳 🗖	HaLow_example.ioc × 🕜 main.c						
~ 🤜 HaLow_example	HaLow example.ioc - Pinout & Cor	nfiguration	LPBAM Scenario & Configu	ration			
> 🕼 Binaries	-		/	/			
 Blueble 	Pinout &		ation	Clock C	onfiguration		Project Manag
> IB Core					N. Collumna Databa	No. Dimond	
> @ Drivers					 Soliware Packs 	• Pinout	
> 😂 MM_loT	Q	0		GPDMA1 Mode and Configuration			
> 49 Middlewares	4.57			Mode			
> 📴 Debug			Channel 15 Words Internal FIFO	2D addressin Disable	~		
B Abstract bt	System Core		Channel 14 Winds Internal EEO	2D addressia Disable			
Halow_example.ioc			Gramer 14 Words memar PPO	2D addressing Disable			
HaLow_example Debug.launch	CORTEX_M33		Words Internal FIFO	2D addressing Unsable Linkort List Mode			1
STM32U585VITXQ_FLASH.id	DCACHE1	_	Channel 12 - 8 Words Internal FIFO	2D addressing Standard Request Mode			
STM32U585VITXQ_RAM.Id	GPDMA1		Channel 11 - 2 Words Internal FIFO	Disable	~		8 3 5
			Channel 10 - 2 Words Internal EIEO	Disable	~		
	✓ ICACHE			LT-SAUND			CAM INTEN
	INDG		Channel 9 - 2 Words Internal HIFO	Lisable			00H_04 ===4
	NVIC		Channel 8 - 2 Words Internal FIFO	Disable	×		DOM_D6 45
	RAMCFG		Channel 7 - 2 Words Internal FIFO	Disable	~		00M_07 44
	¥ RCC		Channel 6 - 2 Words Internal FIFO	Disable	~		18AT
	✓ SYS		Channel 5 - 2 Words Internal EEO	Dirable			9446_940_9413
	WWDG		Channel 5 - 2 Words Internal PPO	CASHDRE .			RC DSCD OF
			Channel 4 - 2 Words Internal FIFO	Disable	¥		155
			Channel 3 - 2 Words Internal FIFO	Disable	~		000
	Analog	\rightarrow	Channel 2 - 2 Words Internal FIFO	Disable	~		HG-0.
			Channel 4 3 Works Internal DEO	Dirable			PH1-Q.
	Timers	· · ·		Configuration			MALE .
	Compatible .		Read Code and an				LOG USART TX
	Connectivity		Reseccomputation				ADC VEDH NO
	Mutimedia		Al Channels SECURITY	CH12 User Constants NVIC S	Settings		K3
		I	Configure the below parameters :				USEA.
	Security		G Search (Chi+F) (C) (O)		0		Logi-
			V Channel 12				LCDA
	Computing)	Request	LinkedList Hosting			GRO DIA
	Middleway and Callman Darks						942
	Modeware and Software Packs						PA3
	Trace and Debug	>					8 8 3 3
							2 2 2
	Power and Thermal	>					



For the pins PD6, PD7, PB3, PB4, PB14 and PB15, click on them, and select the SDMMC2 pin functions.

Click on **Connectivity**, and **SDMMC2** from the left-hand panel. Select, **SD 4 bit Wide bus**.



Click on **Middleware and Software Packs** from the left-hand panel, and then **X-CUBE-FREERTOS**. Set **USE_TICKLESS_IDLE** to **Disabled**.

*HaLow_example.ioc × 🖻 main.c 🗈 main.h	
HaLow_example.ioc - Pinout & Configuration $>$ L	PBAM Scenario & Configuration 🛛 🔪
Pinout & Configuration	Clock Configuration
	✓ Software Packs
Q × 4	STMicroelectronics.X-CUBE-FREERTOS.1.2.0 Mode and Configuration
Categories A->Z	Mode
I-CUBE-ITTIADB	CMCIC DTOCA
🛃 I-CUBE-embOS	V CIVISIS RTUSZ
I-CUBE-wolfMQTT	
I-CUBE-wolfSSH	
I-CUBE-WOITSSL	
L.Cube-SoMuGOAI	
LEVELX	Configuration
V MM IoT	Depart Configuration
NETXDUO	Reset Configuration
Ø THREADX	Advanced settings User Constants CMSIS RTOS2
TOUCHSENSING	Config parameters Include parameters
	Configure the below parameters :
X-CUBE-ALSOBULD	IDLE_SHOULD_YIELD Enabled
X-CUBE-BLE1	USE_MUTEXES Enabled
✓ X-CUBE-BLE2	USE_RECURSIVE_MU Enabled
🛃 X-CUBE-BLEMGR	USE_COUNTING_SEM Enabled
X-CUBE-DPower	QUEUE_REGISTRY_SI 8
X-CUBE-EEPRMA1	USE_APPLICATION_T Disabled
X-CUBE-FREERTOS	ENABLE_BACKWARD Enabled
X-CUBE-GNSS1	
	USE_IICKLESS_IDLE Disabled ~
X-CUBE-NEC4	
A X-CUBE-NFC6	RECORD_STACK_HIG Disabled
X-CUBE-NFC7	USE_MINI_LIST_TEM Enabled
X-CUBE-NFC9	USE_SB_COMPLETE Disabled
X-CUBE-NFC10	USE TICKLESS IDLE
X-CUBE-RT-Thread_Nano	USE TICKLESS IDLE
X-CUBE-SAFEA1	Parameter Description:
X-CUBE-SFXS2LP1	

Next, click on **Project Manager**, **Advanced Settings**. Untick the checkbox under **Visibility (Static)** and tick the checkbox under **Do Not Generate Function Call** for SDMMC2.

Hal.ow_example.loc - Project Manager 👌 LPBAM Scenario & Configuration 🖒				
	Pinout & Configuration	Clock Configuration	Project Manager	Tools
Project	Draw Elector (General Currier) (December Curr			
Code Generator	> SDMAC CORTEX (MA) IS Generated Function Cals Generated Code 2 1 2 2	Ren Function Name Ren Particip Name Ren Ren Ren Ren Ren Ren Ren Ren Ren Ren	N. N. Perpheral Induce Hane Dis Not Generals Function Cal	UT 52
Advanced Settings	0 - 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0 11 0 12 0 13	LKL_COPENIL_INI COPENIL MKL_CAPELINI COPENIL MKL_CAPENIL LPARAL MKL_FERRIT_LAND_INI LPARAL MKL_FERRIT_LAND_INI LPARAL MKL_FERRIT_LAND_INI LPARAL MKL_FERRIT_LAND_INI REG MKL_FERRIT_LAND_INI		

Click **Generate Code** from the **Project** menu to regenerate the code.



laior X R main

Next, open the **Core** \rightarrow **Inc** \rightarrow **main.h** file, and delete or comment out the following lines:

#define	SPI_IRQn	(EXTI15_IRQn)
#define	SPI_IRQ_LINE	(LL_EXTI_LINE_15)
#define	SPI_IRQ_HANDLER	EXTI15_IRQHandler

Replace them with:

#define	WLAN_SDMMC	SDMMC2
#define	WLAN_SDMMC_IRQ	SDMMC2_IRQn
#define	WLAN_SDMMC_IRQHandler	SDMMC2_IRQHandler

Contract Project Explorer 🛛 🕹	🖻 😫 🎖 🕴 🗖 🗖	□ HaLow_example.ioc
 HaLow_example Sinaries 		54 /* USER CODE END ET */ 55
> Includes		57 /* USER CODE BEGIN EC */ 58
 ✓ ➡ Core ✓ ➡ Inc 		59 /* USER CODE END EC */ 60
 in app_freertos.h in custom_bus.h 		61@/* Exported macro 62 /* USER CODE BEGIN EM */
 in custom_conf.h in custom_errno.h 		63 #define MMHAL_MAC_ADDR_OVERRIDE_ENABLED 64 #define JPEG_BUFFER_TIMEOUT_MS (500)
FreeRTOSConfig.h M main.h		65 66 #define SPI_PERIPH (SPI2) 67 #define SPI_DMA_PERIPH (GPDMA1)
 RTE_Components.h stm32_assert.h 		68 #define SPI_RX_DMA_CHANNEL (LL_DMA_CHANNEL_14) 69 #define SPI_TX_DMA_CHANNEL (LL_DMA_CHANNEL_15)
 istm32u5xx_hal_conf.h istm32u5xx_it.h 		70 71 71 //#define SPI_IRQn 72 //#define SPI_IRQ_LINE (LL EXTI LINE 15)
> 🧀 Src > 🇁 Startup > 😂 Drivers		73 //#define SPI_IRQ_HANDLER EXTI15_IRQHandler 74 #define WLAN_SDMMC SDMMC2 75 #define WLAN_SDMMC_IRQ SDMMC2
> 🐸 MM_IoT		76 #define WLAN_SDMMC_IRQHandler SDMMC2_IRQHandler

The **Run** button can now be clicked again to build and run the example using SDIO.

5.6 Changing Network Stacks

This section will document how to change between using LwIP and FreeRTOS+TCP as the network stack. For almost all situations, LwIP should be used. FreeRTOS+TCP should only be used when validating the best low-power performance.

Open the HaLow_example.ioc file and click **Select Components** from the **Software Packs** menu.



Select the **MM_IoT** and **FreeRTOS-Libs** drop down menus. Select the desired network stack and ensure the other is deselected. There should be all green ticks if there are no issues.

MorseMicro.MM_IoT	\odot	1.0.0	
✓ MM_IoT MM_IoT	\odot	2.5.2	
morselib	\odot	2.5.2	V
mmconfig	\odot	2.5.2	V
mmipal	\odot	2.5.2	V
mmutils	\odot	2.5.2	V
mmLwIP		2.2.0	
mmping	\odot	2.5.2	V
mmiperf		2.5.2	
mbedtls		2.5.2	
mm-tinycbor		2.5.2	
> BSP-EKH05	\odot		
> AWS AWS-libraries		2.5.2	
FreeRTOS-Libs FreeRTOS-libraries	\odot	2.5.2	
backoffAlgorithm		2.5.2	
freertos-libs-common		2.5.2	
coreJSON		2.5.2	
coreMQTT		2.5.2	
coreMQTT-Agent		2.5.2	
coreSNTP	2	2.5.2	
FreeRTOS-Plus-TCP	\odot	4.1.0	V
MM_IoT Example	\odot	2.5.2	ping ~
> RealThread.X-CUBE-RT-Thread_Nano		4.1.1 🖻	📥 Install
		131 th 🕰	A Install
> SEGGER.I-CUBE-embOS		1.0.1	
SEGGER.I-CUBE-embOS STMicroelectronics.FP-ATR-ASTRA1	Ö	2.0.0	🔒 Install

Next, click the **Middleware and Software Packs** menu from the .ioc file. Select **MM_IoT** and ensure the **FreeRTOS-Libs FreeeRTOS-libraries** checkbox is ticked.

aLow_example.ioc - Pinout & Conf		
		PBAM Scenario & Configuration
Pinout & Configu	ration	Clock Configuration
		✓ Software Packs
~	0	MorseMicro.MM_IoT.1.0.0 Mode and Configuration
ategories A->Z		Mode
Security	>	MM IoT Example
Computing	>	MM IoT MM IoT
1 5		FreeRTOS-Libs FreeRTOS-libraries
Middleware and Software Packs	~	
 FP-SNS-SMARTAG2 FP-SNS-SMARTAG2 FP-SNS-STBOX1 I-CUBE-CANOPEN I-CUBE-Cesium I-CUBE-FS-RTOS I-CUBE-FSRTOS I-CUBE-WolfNQTT I-CUBE-wolfSSH I-CUBE-wolfSSH I-CUBE-wolfSSL I-CUBE-WOLFSSL		Configuration Marning: This peripheral has no parameters to be configured.

The code will need to be regenerated, and the firmware rebuilt and uploaded to the device for the changes to take effect.

5.7 Updating BLUENRG-M2SP Module Firmware

The most straightforward way to flash the firmware on the BLUENRG-M2SP module is to use the on board STLINK V3. The SWCLK and SWDIO jumpers can be removed, and wires connecting to the BLE SWD headers can be attached as shown below.



Ensure the wires are attached to the top pins of the header as these are connected to the STLINK V3.



Before doing this, the BLE RESET line must be held high by the STM32U585. Please ensure that pin PD2 is configured as an output, and is set to high. The BLUENRG-M2SP module will be held in reset if this is not done, and will not be able to be programmed.

The next step is to download the RF-Flasher Utility from ST. This can be found here - <u>https://www.st.com/en/embedded-software/stsw-bnrgflasher.html</u>

Once installed, open the program and do the following:

- 1. Click on the **SWD** tab.
- 2. Select your connected STLINK device
- 3. Click Select Image File. Select the firmware image file to load.
- 4. Click Flash.



To verify this operation was successful, click the **Verify Flash Content** button. A popup window will appear notifying if the memory content matches the file or not.

FFFlasher Utility v4.5.0	- 0 X
File Tools Help	
Core File Save File As Close File Congrete Two files Settings Mass Erase Files Arges Fall Vort/Files/ Content, Compare Device Memory with file Read Bootinder Sector (SNO) Read OTP Area (SNI)	P) About
Select Image File C Allvers/JonathanGamier/Dewritadu/BLUEHRG-M2SP_DTM_SPI.hex	Start Address: 0x10040000 Plash Stop
UART SWD	MAC Address
	MAC Address: Start: 0x0000000000 - End: 0x0000000000
Control & Marcelland Control & Contro & Control & Control & Control & Control	MAC Flash Location
PlugPlay Mode United All Section 2	File Name 🔄 🖾 tmestamp Set Mac Address
Pash Memory Verbjing Device ID: 003/F02F3233510729362	Interface: ST-Link
Device 03/P02F923951079953554 -> Memory content match.	
Device ID: 003F0/2F3233510739363834 Image File Company Device Memory with, See File ox	
0 1 2 3 4 5 1 2	3 4 5 6
4 b	b.
Stor Log	
11-00-01_389: Device 003P022733751078945454 -> Memory context match. 11-00-01_389: Device Disconnection -> SUCCESS	She Log.

6 Hardware Layout and Configuration

6.1 Power Selection

The voltage supply input to the buck-boost regulators can come from multiple sources. If the jumper is placed on the USB and VIN header, the supply can come from either the USB-C or STLINK Micro USB connectors. If the jumper is placed on the BAT and VIN header, the supply will come from the battery connections.



Powered From Either USB-C or STLink Micro USB Connectors



Powered From Either JST Battery Connector or Screw Terminal Battery Connector

6.2 Using an External Debugger/Programmer

The STLINK V3 MODS is a modular in-circuit debugger and programmer for STM32 and STM8 microcontrollers. It programs the device through a Single Wire Debug (SWD) interface, and connects to a UART interface on the STM32U585 microcontroller in the one cable.

Developers can opt to use their own external debug tool if desired. The jumpers below the module can be removed, and the bottom side of the headers can be attached to with an external debug tool to connect to the relevant pins on the STM32U585 microcontroller.



Connects to 3.3V. Attach target voltage pin of external debugger here. This side connects to STM32U585 Microcontroller. Connect external debug tool to this side of header.

6.3 Changing VFEM Voltage

The voltage supplied to the Front End Module (FEM) on the HaLow module can be changed between 3.3V and 4.3V. Setting to 4.3V will increase the power output from the Power Amplifier, and increase the maximum range. This will also increase power consumption of the VFEM rail when actively transmitting. Note that different Board Configuration Files (BCF) are required for each of the settings. In the Morse Micro examples, an ADC from the STM32U585 is connected to VFEM and will automatically select the correct BCF. For user applications, please take care to ensure the correct BCF is being used.

To switch between the two, the jumper can be placed on the leftmost header and VFEM (3.3V), or the rightmost header and VFEM (4.3V). Power cycle on change.



6.4 Switching Between SDIO and SPI

The MM6108-EKH05 gives users the option to interface the STM32U585 Microcontroller with the HaLow module either through SDIO or SPI. SPI is the default and has been optimised for lowest power consumption. If using SDIO however, the 47 kOhm pull-up resistor must be populated as pointed out by the arrow below. For SPI, this pull up resistor will cause extra leakage current on the VDDIO rail and should be removed.

A different pin configuration in the software is also needed when changing between the two. Section 5.6 details how to configure this.



SPI Configuration



SDIO Configuration

6.5 Switching Between SMA and U.FL Connector

The user has the option to use either a SMA or U.FL connector for the HaLow antenna. The two can be switched between by changing the orientation of a 0 ohm resistor.



SMA Antenna



U.FL Antenna

6.6 Disconnecting Sensors

The sensor blocks on the MM6108-EKH05 feature a 0 ohm resistor that supplies power from the SENSOR header in the power tree to each individual sensor. These sensors include the temperature and humidity sensor, accelerometer, RGB LED, camera header, mikroBUS connector (3.3V on the left, 5V on the right), QWIIC connector, and the BLE module. By removing these 0 ohm resistors, the power can be cut off to the corresponding sensor individually. All these resistors are populated by default.



7 Power Consumption Measurements

7.1 Power Consumption Measurement Points

7.1.1 General Structure

The MM6108-EKH05 is designed to make key power consumption measurements straightforward. Its power distribution follows a tree structure, with each sub-power section branching from the main supply. For instance, to measure the total current draw on the 3.3V rail, an ammeter can be connected across the 3V3 header. If the user wishes to measure only the power consumption of the MCU for example, the ammeter can instead be placed across the MCU header.



7.1.2 HaLow and VFEM

For measuring HaLow power consumption, all HaLow power rails can be assessed together from the HALOW header when VFEM is set to 3.3V. Each individual HaLow supply can also be measured separately from the corresponding header. Keep in mind that VFEM can be configured to either 3.3V or 4.3V. When set to 4.3V, VFEM is disconnected from the rest of the power tree, meaning its current consumption is not included when measuring from the 3V3 or HALOW headers. To measure power consumption at 4.3V, a separate measurement is required.



VFEM set to 3.3V. VFEM will be included in power consumption measurements from 3V3 and HALOW headers.



VFEM set to 4.3V. VFEM will not be included in power consumption measurements from 3V3 and HALOW headers. Separate measurement is needed.

7.1.3 Whole System Power Consumption

Note that the 3V3 header in the current measurement tree is the output from the onboard buck-boost regulators and does not include the power consumption from the regulators themselves. To measure the entire system including the 3.3V and 4.3V buck-boost regulators, an ammeter can be attached in the following position.



If using only 3.3V for VFEM, the user can disable the 4.3V buck-boost regulator by moving the 0 ohm resistor as shown below. This connects the enable line of the regulator to GND, reducing its power consumption to negligible amounts.



4.3V Regulator Enabled



4.3V Regulator Disabled

7.2 Power Consumption Measurement Procedure

A high-precision power analyzer, such as the Joulescope <u>JS220</u>, is recommended for low power measurements. This device can measure both voltage and current, allowing for a precise calculation of overall power. The suggested configuration is shown below.



This configuration can be used for the 3V3, MCU, HALOW, SENSORS, VBAT and VDDIO headers, with the positive ammeter position at the top pin of the header, and the negative at the bottom.

For VFEM, the positive ammeter position should be placed on either the leftmost header (3.3V), or the rightmost header (4.3V) depending on which voltage is desired for VFEM. The negative ammeter position should always be connected to the middle header as shown below.



7.3 Optional Debug GPIO Connections

Certain example applications utilise PA9 and PA10 of the STM32U585 as debug GPIOs. These debug GPIOs will toggle when the device is transitioning between different states and can be useful for measuring different parts of the example applications as they execute. These GPIOs are broken out to the SPARE GPIO header on the MM6108-EKH05 and can be connected to the Joulescope as follows.



On the Joulescope UI, the toggling debug GPIOs can be seen at the bottom and are annotated over to show the different states in a ping example application.



Definitions for each debug state can be found in the example folder under each application of interest. For example, the definitions for the ping application can be found in the following locations.

Windows:

C:\Users\USER_NAME\STM32Cube\Repository\Packs\MorseMicro\MM_IoT\1.3.3\E xample\ping\ping.c

Linux:

/home/USER_NAME/STM32Cube/Repository/Packs/MorseMicro/MM_IoT/1.3.3/Exam
ple/ping/ping.c

Mac:

/Users/USER_NAME/STM32Cube/Repository/Packs/MorseMicro/MM_IoT/1.3.3/Exa
mple/ping/ping.c

A search for the **enum** of **debug_state** will show the definitions. The function **set_debug_state** is used to set these throughout the application.

7.4 Measuring DTIM Power

To measure DTIM power consumption, use the ping application. Make sure to use the correct country code, BCF, SSID, and passphrase to connect the device to the AP. The DTIM period and bandwidth/channel can also be changed by following the steps in Section 4.2.1. To get an accurate measurement for just the HaLow and MCU power consumption, it is recommended to remove the SENSORS header on the power tree. The ping application will perform the following actions:

- Boot the MM6108 chip.
- Connect to the AP.
- Ping the AP 10 times.
- Remain connected in power save mode while idling for 60 seconds.
- Disconnect from the AP.

Before starting, ensure that the ping application is loaded onto the device and that the device is properly connected to the AP. This can be confirmed by monitoring the UART output. Note that the UART baud rate is set to 115200.

```
Morse Ping Demo (Built Sep 16 2024 13:44:00)
    BCF API version:
                                             6.7.0
    BCF build version:
                                             fbed8cb 244ffd771b
   BCF board description:
                                            BAILEY_V12
   Morselib version:
                                             998b9a497_NFP
   Morse firmware version: 1.13.0
   Morse chip ID:
                                             0x0306
Initialize IPv4 with static IP: 192.168.1.2...
Initialize IPv6 using Autoconfig...
Morse FreeRTOS+ TCP interface initialised. MAC address 94:bb:43:dc:f9:52
Attempting to connect to MorseMicro with passphrase 12345678
This may take some time (~30 seconds)
WLAN STA connecting
 WLAN STA connected
DNS server 0: 192.168.1.1
Link is up. Time: 7068 ms, IP: 192.168.1.2, Netmask: 255.255.255.0, Gateway: 192.168.1.1, D1
Ping 192.168.1.1 56(64) bytes of data.
(192.168.1.1) packets transmitted/received = 2/2, round-trip min/avg/max = 12/12/12 ms
(192.168.1.1) packets transmitted/received = 3/3, round-trip min/avg/max = 11/11/12 ms
(192.168.1.1) packets transmitted/received = 4/4, round-trip min/avg/max = 11/11/12 ms
(192.168.1.1) packets transmitted/received = 5/5, round-trip min/avg/max = 11/11/12 ms
(192.168.1.1) packets transmitted/received = 5/6, round-trip min/avg/max = 11/11/13 ms
(192.168.1.1) packets transmitted/received = 6/6, round-trip min/avg/max = 11/11/13 ms
(192.168.1.1) packets transmitted/received = 7/7, round-trip min/avg/max = 11/11/13 ms
(192.168.1.1) packets transmitted/received = 8/8, round-trip min/avg/max = 11/11/13 ms
(192.168.1.1) packets transmitted/received = 9/9, round-trip min/avg/max = 11/11/13 ms
(192.168.1.1) packets transmitted/received = 10/10, round-trip min/avg/max = 11/11/13 ms
--- 192.168.1.1 ping statistics ---
10 packets transmitted, 10 packets received, 0.000% packet loss
round-trip min/avg/max = 11/11/13 ms
 WLAN STA disabled
Link is down. Time: 27868 ms
```



The below screenshot shows the power consumption profile as the program executes.

The green highlighted section below illustrates the measurement of a DTIM period, zoomed in from the DTIM Wakeups section displayed above. In this example, the measurement is for a DTIM3 with an 8 MHz channel and a 102.4 ms beacon interval. Adjusting these parameters will impact the measured power consumption.



7.5 Measuring WNM Sleep Power

WNM Sleep provides a more sophisticated sleep mechanism, where the sleep schedule is explicitly negotiated between the station and the AP. This allows for longer sleep intervals and more efficient data retrieval when the client wakes up.

To measure WNM sleep power consumption, use the wnm_sleep application. Ensure that the correct country code, BCF, SSID, and passphrases are configured so the device can connect to the AP. To get an accurate measurement for just the HaLow and MCU power consumption, remove the SENSORS header on the power tree.

The wnm_sleep application will perform the following actions:

- Boot the MM6108 chip.
- Connect to the AP.
- Ping the AP 10 times.
- Enter WNM sleep for 20 seconds.
- Exit WNM sleep.
- Ping the AP another 10 times.
- Enter WNM sleep with chip in shutdown mode for 20 seconds.
- Exit WNM sleep.
- Disconnect from the AP.

As with DTIM measurements, start by verifying that the wnm_sleep example is correctly loaded and that the device can successfully associate with the AP via the MM6108-EKH05's UART. Below are screenshots of the complete UART output from the program, along with an annotated power trace.

Morse WNM Sleep Demo (Built Sep 16 2024 14:15:05) BCF API version: 6.7.0 BCF build version: fbed8cb 244ffd771b BCF board description: BAILEY_V12 Morselib version: 998b9a497_NFP Morse firmware version: 1.13.0 Morse chip ID: 0x0306 Initialize IPv4 with static IP: 192.168.1.2... Initialize IPv6 using Autoconfig... Morse FreeRTOS+ TCP interface initialised. MAC address 94:bb:43:dc:f9:52 Attempting to connect to MorseMicro with passphrase 12345678 This may take some time (~30 seconds) WLAN STA connecting WLAN STA connected DNS server 0: 192.168.1.1 Link is up. Time: 8176 ms, IP: 192.168.1.2, Netmask: 255.255.255.0, Gateway: 192.168.1.1, D1 Ping 192.168.1.1 56(64) bytes of data. (192.168.1.1) packets transmitted/received = 2/2, round-trip min/avg/max = 12/12/12 ms (192.168.1.1) packets transmitted/received = 3/3, round-trip min/avg/max = 12/12/12 ms (192.168.1.1) packets transmitted/received = 4/4, round-trip min/avg/max = 11/11/12 ms (192.168.1.1) packets transmitted/received = 5/5, round-trip min/avg/max = 11/11/12 ms (192.168.1.1) packets transmitted/received = 6/6, round-trip min/avg/max = 11/12/14 ms (192.168.1.1) packets transmitted/received = 7/7, round-trip min/avg/max = 11/12/14 ms (192.168.1.1) packets transmitted/received = 8/8, round-trip min/avg/max = 11/12/14 ms (192.168.1.1) packets transmitted/received = 9/9, round-trip min/avg/max = 11/11/14 ms (192.168.1.1) packets transmitted/received = 10/10, round-trip min/avg/max = 11/11/14 ms - 192.168.1.1 ping statistics --10 packets transmitted, 10 packets received, 0.000% packet loss round-trip min/avg/max = 11/11/14 ms Entering WNM sleep with chip power down disabled Expected sleep time 20000ms. Enter WNM sleep took 16 ms. Exit WNM sleep took 17 ms. Ping 192.168.1.1 56(64) bytes of data. (192.168.1.1) packets transmitted/received = 2/2, round-trip min/avg/max = 3/7/11 ms (192.168.1.1) packets transmitted/received = 3/3, round-trip min/avg/max = 3/9/13 ms (192.168.1.1) packets transmitted/received = 4/4, round-trip min/avg/max = 3/9/13 ms (192.168.1.1) packets transmitted/received = 5/5, round-trip min/avg/max = 3/14/35 ms (192.168.1.1) packets transmitted/received = 6/6, round-trip min/avg/max = 3/14/35 ms (192.168.1.1) packets transmitted/received = 7/7, round-trip min/avg/max = 3/14/35 ms (192.168.1.1) packets transmitted/received = 8/8, round-trip min/avg/max = 3/14/35 ms (192.168.1.1) packets transmitted/received = 9/9, round-trip min/avg/max = 3/14/35 ms (192.168.1.1) packets transmitted/received = 10/10, round-trip min/avg/max = 3/14/35 ms - 192.168.1.1 ping statistics -10 packets transmitted, 10 packets received, 0.000% packet loss round-trip min/avg/max = 3/14/35 ms Entering WNM sleep with chip power down enabled. Expected sleep time 20000ms. Enter WNM sleep took 13 ms. Exit WNM sleep took 623 ms. WLAN STA disabled Link is down. Time: 69345 ms



8 Troubleshooting

8.1 Device Is Not Programming

If the STLINK V3-MODS is unable to connect to the STM32U585, please check the following steps.

1. Ensure the STLINK jumpers are populated. Without these, the programmer will be disconnected from the microcontroller.



2. Ensure the 3V3 and MCU jumpers are populated. The microcontroller will not be powered without these.



3. Ensure the power select jumper is on USB. If using batteries, check VIN voltage to ensure the battery is functioning normally.



4. Ensure the device is not in a low power state. If the microcontroller is in certain deep sleep states, the programmer will not be able to attach. To recover in this situation, hold the STM RESET button as the device is attempting to connect, and release. This will ensure the microcontroller can be attached to.



8.2 Poor HaLow Performance

If the HaLow performance is poor, the following items should be checked.

- 1. Confirm the BCF loaded on to the device matches the VFEM setting. The bcf_aw_hm593.bin file should be used for 3.3V, and the bcf_aw_hm593_4v3.bin file should be used for 4.3V. If using a Morse Micro example from the CMSIS pack, the software will automatically detect which setting is used and select the correct BCF. If this is the case, the device might just need to be reset with the STM RESET button.
- 2. Ensure the AP configuration is expected. I.e. the correct channel and bandwidth are configured.
- 3. Ensure there are no other HaLow devices on the same channel causing in-band interference.
- 4. Ensure antenna connection is on securely.
- 5. If cabled, ensure there is sufficient attenuation (~40 dB) between the devices. If the power is too high, it will saturate the LNA.

8.3 High Power Consumption

There are many reasons that high power consumption may be measured. Please check the following:

- 1. If the T_VCC jumper is populated, this will add ~300uA to the 3V3 power consumption.
- 2. When measuring just HaLow and MCU power consumption, ensure the SENSORS jumper is not populated.
- 3. To replicate Morse Micro DTIM power consumption numbers, the devices should be placed inside a shielded box away from interference. If the device fails to receive the beacon from the AP, it will spend a longer time in active RX mode and consume more power resulting in misleading DTIM measurements.



- 4. The temperature plays a big factor in power consumption. All testing should be done at room temperature of ~77 degree Fahrenheit to match Morse Micro DTIM power consumption numbers.
- 5. Ensure the MM6108-EKH05 is associated with the AP. Double check SSID and passphrase match.

9 Errata

10 Revision History

Release Number	Release Date	Release Notes
01	19/11/2024	Initial release.

Appendix A: Schematics








74 | morsemicro.com







		1		2	3			4			
]
	Revision	History									
	1.0	- Initial release.									
A	2.0	Connect USB data to STM Connect RTS/CTS to BLE Module. Swap UART pins of BLE Module. Update buck boost output cap to 22uF Reversed UART RX/TX connections on Moved power supply for QSPI Flash to V Updated battery connector to JST Added bulk cap on Power select Removed load switch for sensors power s Added price to charge VEFM to 4.3V	STLINK 'DD_SENSORS upply								A
в	3.0	 Added option to change VFEM 104.3V Updated SMA footprint. Updated BLE module communication pri SPI. Updated pinout as needed. Removed 00hm links to unneeded SPI/SI Combined the MM_SPI_CS and MM_SI Pin assignment updates to red status LED MIKRO_PWM Updated VFEM High Power Select to 4.3 Removed Vin & VFEM from power mea 	otocol from UART to DIO connections. PLINT pins with SDIO , MIKRO_AN and W sure tree								в
с											с
D						Drawn: J. Garn Checked: Matthe Approved: Chad O Date Released:2/	Orse Micro	Level 8, 10-14 Surry Hills, NS AUSTRALIA ion History _EKH05.PrjPcb	Waterloo Street W 2010 (c) Copyright Drawing 3.0 Revision: 3.0	Size:	. D
		1		2	3		rile: kevision_Hi	4	Sneet8	01.0	1

Appendix B: Mechanical Drawing





Morse Micro Pty. Ltd. Corporate Headquarters Level 8, 10-14 Waterloo Street, Surry Hills, NSW 2010, AUSTRALIA

Morse Micro Inc. – USA

40 Waterworks Way Irvine, CA 92618, UNITED STATES

Morse Micro – China

Floor 7, Room 08-210, Wework office 292 Yan'an Rd, Shangcheng District, Hangzhou 310003

USA: +1.949.501.7080 Australia: +61.02.905.45922 China: +86.571.229.30277 Email: sales@morsemicro.com www.morsemicro.com

© 2024 Morse Micro Pty. Ltd.

All rights reserved. Morse Micro and the Morse Micro logo are trademarks of Morse Micro ltd. All other trademarks and service marks are the property of their respective owners.

About Morse Micro

Morse Micro is producing IEEE 802.11ah / Wi-Fi HaLow solutions for Internet of Things (IOT) - based on a newly certified Wi-Fi standard called HaLow. Morse Micro is a VC-backed Startup headquartered in Sydney, Australia. Learn more at www.morsemicro.com



Morse Micro reaching farther[™]

4.