GW16159 + EKH05 Setup Guide

Summary

This guide describes the steps to bring up the MorseMicro EKH05 evaluation kit with the VeniceFLEX + GW16159 HaLow Radio. We will first create an AP ”Hotspot” with the Venice board using a bringup script, then use the EKH05 scan application to sanity-check the signal. Then, we will connect the EKH05 to the Hotspot, allowing the connection to be verified and tested. Lastly we will create a bridged setup to allow the EKH05 web interface to be viewed on a LAN with the Venice board serving as the bridge between networks.

Introduction to the EKH05

The EKH05 (MM6108-EKH05) is an evaluation and development kit for the MM6108, a HaLow (sub-1 GHz) system-on-chip that provides wireless S1G connectivity for both microcontrollers (MCUs) and microprocessors (MPUs). The EKH05 integrates an onboard STM32 microcontroller, running Morse Micro's MCU software frameworks, which serve as the basis for application development and evaluation. To learn more, see the MM6108-EKH05 product brief.

Introduction to the GW16159

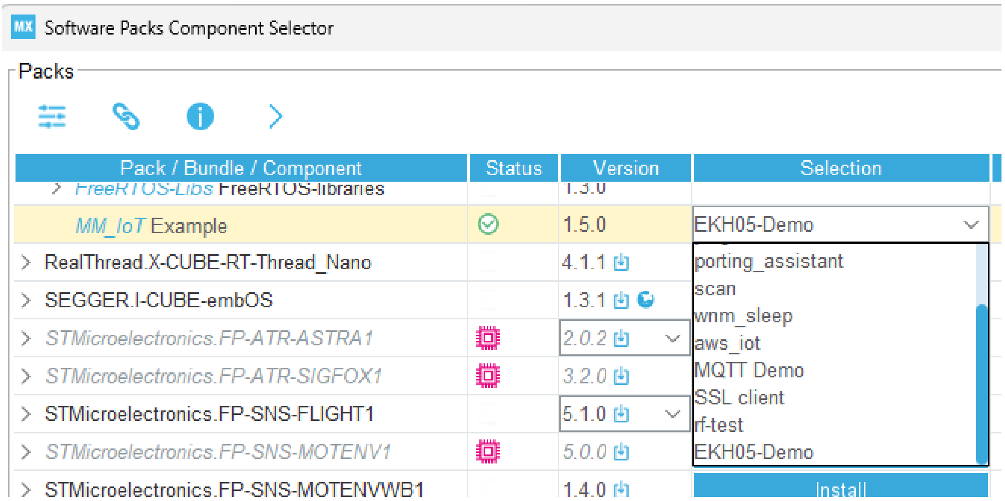
The Gateworks GW16159 is a HaLow (802.11ah, sub-1 GHz) radio card compatible with the Gateworks VeniceFLEX boards. It uses a Silex SX-SDMAH module, which inside uses the MM6108 chip. Connects to the VeniceFLEX board through the M.2 E-Key socket using SDIO. Gateworks includes kernel drivers and userspace programs which allow the GW16159 to register a standard wireless network interface in Linux. To learn more, see the GW16159 Wiki page and product page.

Setup Instructions

If you are having problems during the following steps, see the Troubleshooting section at the end of the document.

1. **Install IDE and dependencies for Scan program**
   - Please review the (EKH05 user guide) before following this documentation as it is highly in-depth for installing the dependencies necessary for communicating with the EKH05.g
   - Install STM32CubeIDE and CMSIS dependencies as described in the EKH05 User Guide (Section 5)

- Follow Section 5.3 of the guide and select the “scan” Example Application. Then press **Generate Code**, install any necessary dependencies, and build the program (not flashing yet). We do not need to have the EKH05 setup until later in the guide.



Software Packs Component Selector

Packs

Pack / Bundle / Component	Status	Version	Selection
FreeRTOS-Libs FreeRTOS-libraries		1.3.0	
MM_IoT Example	✓	1.5.0	EKH05-Demo
RealThread.X-CUBE-RT-Thread_Nano		4.1.1	
SEGGER.I-CUBE-embOS		1.3.1	
STMicroelectronics.FP-ATR-ASTRA1		2.0.2	
STMicroelectronics.FP-ATR-SIGFOX1		3.2.0	
STMicroelectronics.FP-SNS-FLIGHT1		5.1.0	
STMicroelectronics.FP-SNS-MOTENV1		5.0.0	
STMicroelectronics.FP-SNS-MOTENVWB1		1.4.0	

MM_IoT Example Applications configuration

2. **Prepare VeniceFLEX SBC**

- On a fresh Ubuntu Noble install, connect the GW16159 HaLow radio card and antenna.
- Connect the VeniceFLEX to the internet via ethernet or an adapter.
- Connect the VeniceFLEX to a host computer using the JTAG debugger or other desired method (ssh). Note: ssh for root is disabled on default images. See the Gateworks Getting Started wiki page for instructions
- Install the necessary dependencies, configurations, and scripts.

```
sudo apt update
sudo apt install -y git hostapd dnsmasq
git clone https://github.com/ChipDev/ekh05-venice-halow.git
```

- Perform an initial operation check by running the following.

```
#Bring up the wlan0 interface so morse_cli can communicate
ifconfig wlan0 up
morse_cli --interface=wlan0 hw_version
morse_cli --interface=wlan0 version
morse_cli channel
```

A valid example response (dmesg truncated):

```
#ifconfig wlan0 up
[  245.076157] morse_sdio mmc0:0001:2: morse_mac_ops_add_interface: ...
[  245.088112] morse_sdio mmc0:0001:2: morse_mac_ops_add_interface: ...
[  245.100964] morse_sdio mmc0:0001:2: morse_mac_ops_add_interface: ...
```

```
# morse_cli --interface=wlan0 hw_version
HW Version: MM6108-A1
# morse_cli --interface=wlan0 version
Morse_cli Version: rel_1_12_5_2024_Jul_25-4-g3541610
FW Version: rel_1_12_4_2024_Jun_11
# morse_cli channel
Full Channel Information
        Operating Frequency: 916000 kHz
        Operating BW: 8 MHz
        Primary BW: 2 MHz
        Primary Channel Index: 0
```

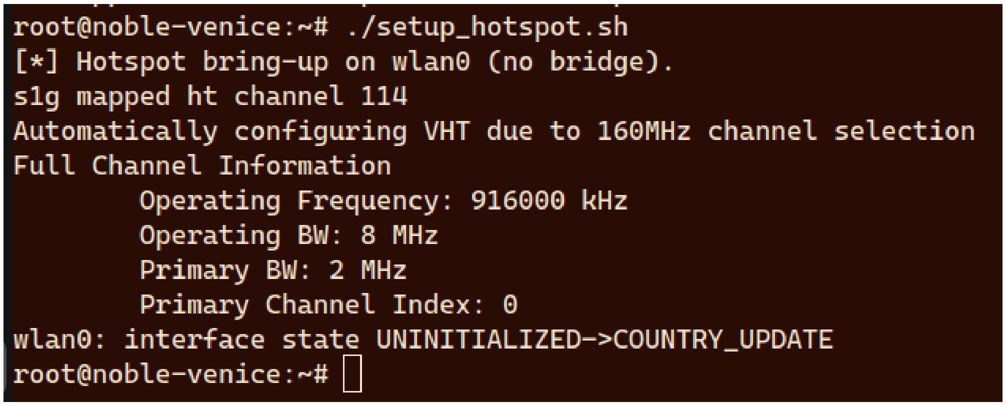
3. **Start AP on SBC**

- Start the AP in ”Hotspot” mode by running the following

  ```
  cd ekh05-venice-halow
  ./setup_hotspot.sh
  ```

- The SBC should now broadcast a Wi-Fi HaLow WPA3 SAE network.

  NOTE: The GW16159 supports different modes of WiFi security such as WPA2/3, but the EKH05 does not support connecting to WPA2 networks. Refer to the Gateworks 16159 Wiki page to find WPA2 bringup configuration.

  ```
  root@noble-venice:~# ./setup_hotspot.sh
  [*] Hotspot bring-up on wlan0 (no bridge).
  s1g mapped ht channel 114
  Automatically configuring VHT due to 160MHz channel selection
  Full Channel Information
          Operating Frequency: 916000 kHz
          Operating BW: 8 MHz
          Primary BW: 2 MHz
          Primary Channel Index: 0
  wlan0: interface state UNINITIALIZED->COUNTRY_UPDATE
  root@noble-venice:~#
  ```

  Desired console output from setup_hotspot.sh

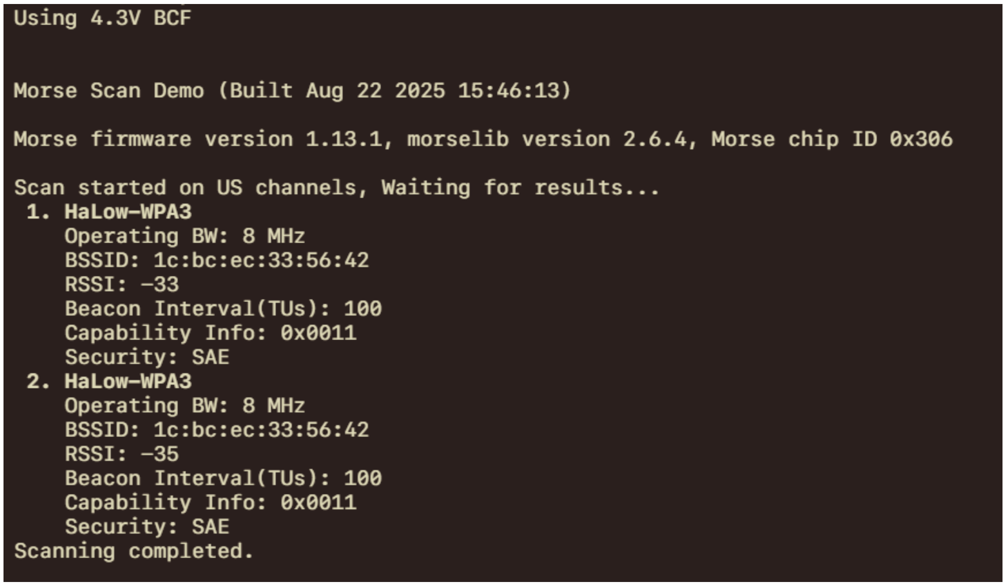
4. **Connect EKH05 hardware to development computer**

- Attach the desired antennae to the EKH05 *before* powering it.
- Connect the onboard ST-Link to the development host via MicroUSB and install drivers as per the user guide.
- Open your desired serial console and connect to the ST-LINK (i.e. PuTTY or screen) at 115200 baud
- Flash the EKH05 with the scan demo by **building and flashing** the program as per the setup guide. NOTE: Ensure that the scan program was chosen and

"Generate Code" was pressed before flashing, as per Step 1, otherwise the example program will not be changed.

5. **Verify scan demo & Confirm AP detection**

    - Pres the "STM RESET" button on the EKH05 to restart the microcontroller.
    - Use the serial console to confirm the *Morse Scan Demo* runs. (If not, see Troubleshooting).
    - Ensure the HaLow AP (HaLow-Demo) is detected with `Security: SAE`.
    - If it shows "Other," the AP was created as WPA2 and the EKH05 cannot connect. See the Troubleshooting section at the end of the document



```
Using 4.3V BCF


Morse Scan Demo (Built Aug 22 2025 15:46:13)

Morse firmware version 1.13.1, morselib version 2.6.4, Morse chip ID 0x306

Scan started on US channels, Waiting for results...
 1. HaLow-WPA3
    Operating BW: 8 MHz
    BSSID: 1c:bc:ec:33:56:42
    RSSI: -33
    Beacon Interval(TUs): 100
    Capability Info: 0x0011
    Security: SAE
 2. HaLow-WPA3
    Operating BW: 8 MHz
    BSSID: 1c:bc:ec:33:56:42
    RSSI: -35
    Beacon Interval(TUs): 100
    Capability Info: 0x0011
    Security: SAE
Scanning completed.
```

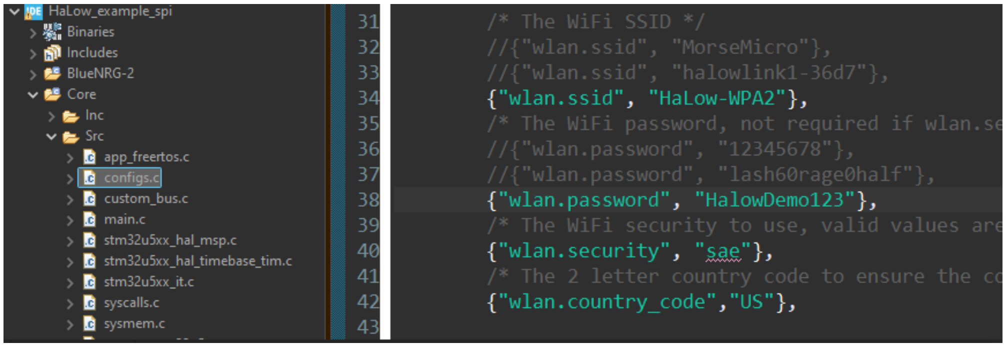
EKH05 scan output detecting HaLow AP.

6. **Switch EKH05 to demo mode**

    - Now that we've confirmed that the EKH05 can see the AP, we'll switch the EKH05 from scanning to connecting to the AP.
    - In STM32CubeIDE, change the component selector from `scan` to `EKH05-Demo`, then **Generate Code**. Do not build or flash yet.

7. **Configure connection parameters**

    - Edit `configs.c` *(Core/Src/configs.c* to set `wlan.ssid` and `wlan.password` to match the AP. (i.e. ssid HaLow-WPA3, password HalowDemo123)

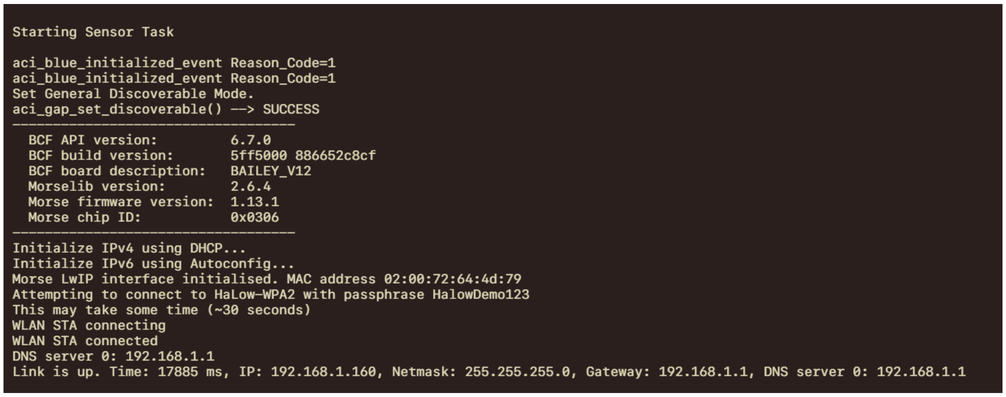


```
        /* The WiFi SSID */
        //{"wlan.ssid", "MorseMicro"},
        //{"wlan.ssid", "halowlink1-36d7"},
        {"wlan.ssid", "HaLow-WPA2"},
        /* The WiFi password, not required if wlan.se
        //{"wlan.password", "12345678"},
        //{"wlan.password", "lash60rage0half"},
        {"wlan.password", "HalowDemo123"},
        /* The WiFi security to use, valid values are
        {"wlan.security", "sae"},
        /* The 2 letter country code to ensure the co
        {"wlan.country_code","US"},
```

Setting WiFi connection parameters in configs.c

8. **Build and flash EKH05**

   - Compile and flash the EKH05 firmware.
   - Confirm the device prints `WLAN STA connected` and receives a DHCP IP.



```
Starting Sensor Task

aci_blue_initialized_event Reason_Code=1
aci_blue_initialized_event Reason_Code=1
Set General Discoverable Mode.
aci_gap_set_discoverable() --> SUCCESS
-----------------------------------
  BCF API version:         6.7.0
  BCF build version:       5ff5000 886652c8cf
  BCF board description:   BAILEY_V12
  Morselib version:        2.6.4
  Morse firmware version:  1.13.1
  Morse chip ID:           0x0306
-----------------------------------
Initialize IPv4 using DHCP...
Initialize IPv6 using Autoconfig...
Morse LwIP interface initialised. MAC address 02:00:72:64:4d:79
Attempting to connect to HaLow-WPA2 with passphrase HalowDemo123
This may take some time (~30 seconds)
WLAN STA connecting
WLAN STA connected
DNS server 0: 192.168.1.1
Link is up. Time: 17885 ms, IP: 192.168.1.160, Netmask: 255.255.255.0, Gateway: 192.168.1.1, DNS server 0: 192.168.1.1
```

EKH05 receives IP (192.x) via DHCP from Venice board

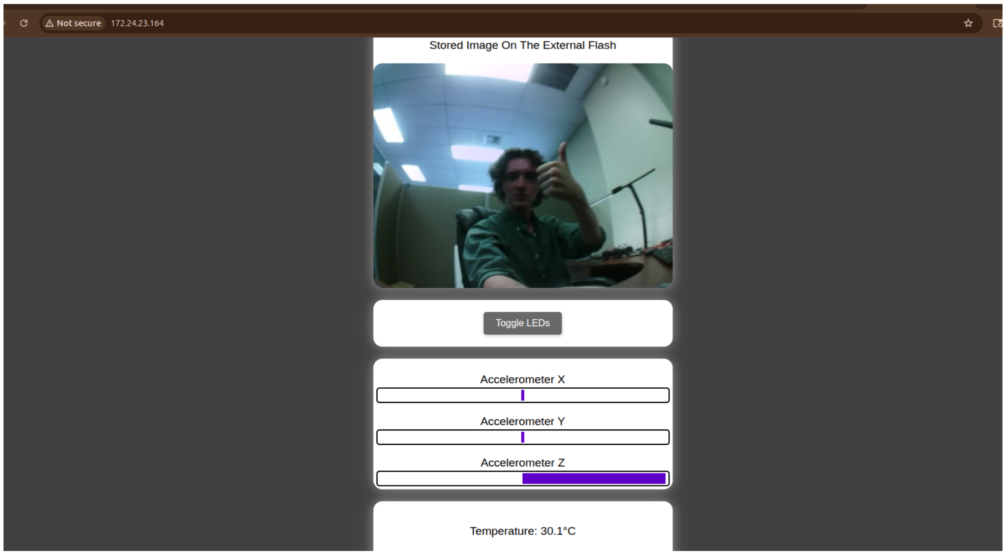
   - Test connectivity by pinging the EKH05 from the SBC. Stress testing can be done via tools such as iperf.

9. **Enable LAN bridging for webserver access via the browser**

   - Reboot the SBC to clear networking state.
   - Start the AP in ”Bridged” mode while SBC is connected to LAN via Ethernet:

```
cd ekh05-venice-halow
./setup_bridged.sh
```

   - Power Cycle / Press the STM RESET button on the EKH05.
   - EKH05 will now obtain an IP from the LAN DHCP server once connected to the SBC.
   - Devices on the LAN can now access the EKH05 webserver using its assigned IP.



Stored Image On The External Flash

Temperature: 30.1°C

EKH05 Web server can be accessed by LAN devices through Venice bridge.

Your EKH05 is now connected to your LAN through via HaLow and a bridge on your Gateworks SBC. To go back to the non-bridge (hotspot) example for direct peer-to-peer testing, restart the SBC to reset all the current bridging configuration.

10. **Next steps**

    (a) The GW16159 has many other builtin features such as Wi-Fi mesh (802.11s) For building HaLow mesh systems, Gateworks provides some example hostapd_s1g configurations on the Gateworks GW16159 Wiki Page, along with detailed documentation on the capabilities of the GW16159.

Troubleshooting

1. **morse_cli: command not found**

   (a) Ensure you’re on an updated Gateworks image with the proper MorseMicro drivers and CLI tools installed. Download a prebuilt updated image (noble-venice.img.gz) from the Gateworks Ubuntu page and flash it through your preferred method (i.e. tftp, see the Venice Firmware wiki page)

2. **EKH05 not connecting to network**

   (a) First check if the AP is being detected by the EKH05 with the ’scan’ application as listed in step 5. If no network is not being detected there, ensure proper antenna hardware and installation.

   (b) To see debug messages from hostapd_s1g (to detect hardware or kernel errors with hostapd) edit the setup_hotspot.sh file to include debugs and be in the foreground.

   ```
   #Change the previous hostapd_s1g line
   "${HOSTAPD_BIN}" "${HOSTAPD_CFG}" -B
   #to be debug-printing and in the foreground.
   "${HOSTAPD_BIN}" "${HOSTAPD_CFG}" -d
   ```

3. **Scan demo is not running / Stuck in scan mode**

   (a) If after changing the application to ”scan” in the STM32CubeIDE and flashing the EKH05 via the ST-LINK, the serial console still prints ”Starting Sensor Task” or something other than ”Morse Scan Demo”, ensure you ”compile” the desired program into the code by pressing **Generate Code** after changing the MM_Iot Example selection as per step 1. If the correct example is selected, and the code compiles without errors, ensure the board was properly flashed and verified in the STM32CubeIDE. Power cycle the board and repeat the flashing process if the program is not changing.

   (b) **Generate Code** also applies when changing the program back to EKH05-Demo

4. **morse_cli returning firmware errors**

   (a) morse_cli cannot talk to the device without the interface being up. This is intended behavior, documented on the GW16159 wiki page

   ```
   # morse_cli version
   Morse_cli Version: rel_1_12_5_2024_Jul_25-4-g3541610
   NL80211, code -100: Error callback called
   NL80211, code -1: Failed to rcvmsgs
   Get firmware version failed (-1)

   # ifconfig wlan0 up
   # morse_cli version
   Morse_cli Version: rel_1_12_5_2024_Jul_25-4-g3541610
   FW Version: rel_1_12_4_2024_Jun_11
   ```