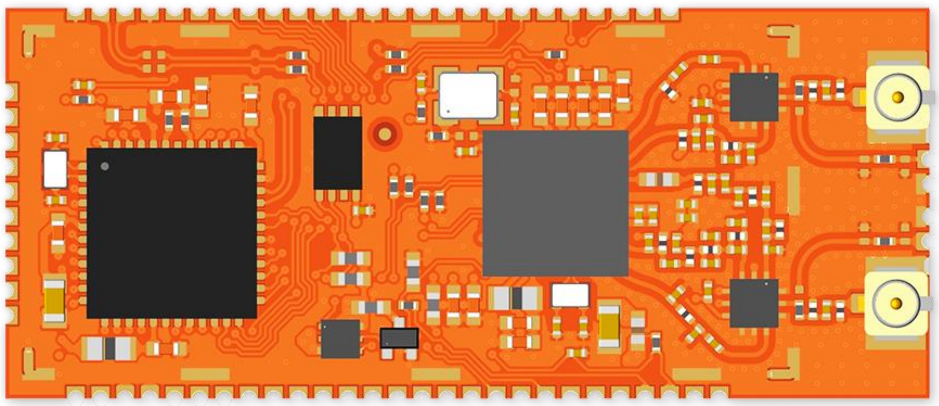
MBC-LR

Modular Brick Concept

Combo UHF 433/868/915 LoRa® SoM
User Manual



Revision History

Revision number	Revision date	Summary of changes	Authors
1.0	26/09/2023	Initial version	D.Trimarchi C. Ruggeri

FCC Warning Statement
Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment. This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not
installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
-- Reorient or relocate the receiving antenna.
-- Increase the separation between the equipment and receiver.
--Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-- Consult the dealer or an experienced radio/TV technician for help.
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and
(2) this device must accept any interference received, including interference that may cause undesired operation.

RF Exposure Statement
To maintain compliance with FCC's RF Exposure guidelines, this equipment should be installed and operated with minimum distance of 20cm the radiator your body. This device and its antenna(s) must not be co-located or operation in conjunction with any other antenna or transmitter.

Tips & Tricks

Notes

Warning messages

```
Code snippets
```

Getting Started

The MBC-LR SoM is part of Briki MBC family of devices that share the same pinout and form factor regardless of each module's characteristics, such as MCU selection, peripherals, or field of application. This compact and certified SoM is the ideal solution for designers who want a unique device for their IoT or IIoT projects requiring an UHF radio interface ranging from 137 MHz up 1020 MHz featuring LoRa® modulation and an additional MCU dedicated to add GPIOs, wired interfaces and computational power.

It features (depending on module versions):

- an ATSAML21G18B ultra-low power ARM® Cortex®-M0+ MCU with 256 KB Flash, 32 KB of SRAM, operating at a maximum frequency of 48 MHz and reaching 2.46 CoreMark/MHz;
- an ATSAMR34J18 ultra-low power ARM® Cortex®-M0+ MCU with 256 KB Flash, 32 KB of SRAM + UHF transceiver SX1276 by Semtech capable of LoRa®, (G)FSK, (G)MSK e OOK modulations;
- a CryptoAuth ECC608B chip for the module security;
- Up to 3 pre-configured radio bands: 433/868/915 MHz with u.fl connectors (or pads)S antenna interface.

This ultra-low power SoM has been designed to be configured with a minimal configuration featuring the only radio frontend (the SAMR34) or in case the application requires more power, in conjunction with a second SAML21 to increase the computational power, adding more wired interface and GPIOs, offering to the customer the most versatile and complete SoM.

Both the configurations, minimal and full, can be further personalized by choosing which band must be used for the wireless communication, offering on the same module the compatibility with US, European and many other markets at the same time.

The MBC-LR makes all these features available to the user in a tiny 38x16mm SoM with 62 pins.

A complete module pinout, with each pin functions description, is available here, while the block diagram of the module is available here.

Two separate power supplies are exposed on the module pinout to allow the maximum flexibility and control over its power supply requirements. Detailed information about power supply voltages and current consumptions can be found in the Electrical characteristics section of this manual.

Detailed information about the supported power and sleep modes are available in the Power modes section.

The USB interface is available on the module pinout. Its default configuration depends on the module configuration in turn: in the minimal version, it is configured in device mode with just one USB-CDC interface exposed by the SAMR34; in the full version, there are two separate USB-CDC device interfaces, one for the SAML21 and one for the SAMR34. For more information, read the related USB Interface section.

Both MCUs are fully user accessible (both in programming and debugging) to allow a full configuration/customization for each design's needs. Both the SAML21 and the SAMR34 can be reprogrammed in-system through the SWD interfaces exposed on the module pinout. For non-intrusive on-chip debug of application code, the same interfaces can be used.

Each chip can update its own or the counterpart's firmware (even in a secured environment, thanks to the crypto chip supplied with the board). This leads to two more ways to reprogram the MBC-LR:

- through the USB bootloader in the SAML21/R34. For more information about how the SAML21/R34's bootloader works, click here
- through the radio interface (1) offered by the SAMR34.

Since the firmware of both MCUs can be customized, the user has the power to decide which of the two should act as main or companion and what task each chip should perform. This allows the designer to achieve a perfect balance between power consumption, functionalities implemented and overall performance. Even a "fluid" configuration can be achieved, where each MCU works based on its own tasks, exchanging data and messages thanks to the shared communication interface.

One communication interface, plus some synchronizing signals, is shared between the two MCUs: one low-power UART, suitable for data exchange and firmware update. All the other peripherals and GPIOs are available for the user and exposed on the module pinout. To better understand how the communication interface works, please check the Communication section of this manual.

Every MBC features a special ID line that can be configured to output to a carrier board the SoM's characteristic (MCUs mounted on the SoM, operating voltages, power consumptions, features enabled, etc.) during the board boot.

Key features

Features of the SAMR34:

- Processor
  - ARM©, Cortex-M0+ CPU running at up to 48 MHz
  - Single-cycle hardware multiplier
  - Micro Trace Buffer

- Memories
  - 256KB in-system self-programmable Flash
  - 32KB SRAM Memory
  - 8 KB low-power RAM with battery backup retention

- System
  - Power-on reset (POR) and brown-out detection (BOD)
  - Internal and external clock options with 48MHz Digital Frequency Locked Loop (DFLL48M) and 48MHz to 96MHz Fractional Digital Phase Locked Loop (FDPLL96M)
  - External Interrupt Controller (EIC), 15 external interrupts, one non-maskable interrupt
  - Two-pin Serial Wire Debug (SWD) programming, test and debugging interfaces

- Low Power Consumption
  - Transceiver:
    - RX = 16 mA (typ)
    - RFO_HF = 33 mA (typ)
    - PA_BOOST = 95 mA (typ)
  - MCU:
    - Idle and Standby Sleep Modes
    - SleepWalking peripherals

- Temperature Range: -40°C to + 85°C (Industrial)

- RF/Analog Features
  - Integrated LoRa Technology Transceiver:
    - Tri-band Coverage
      - 137 MHz to 175 MHz
      - 410 MHz to 525 MHz
      - 862 MHz to 1020 MHz
    - +20 dBm (100 mW) Max Power (VDDANA > 2.4 VDC)
    - +17 dBm (50 mW) Max Power (Regulated PA)
    - +13 dBm (20 mW) High-efficency PA
  - High Sensitivity:
    - Down to -136 dBm (LoRaWAN™ protocol compliant modes)
    - Down to -148 dBm (proprietary narrowband modes)
  - Up to 168 dB Maximum Link Budget
  - Robust Front-End: IIp3 = -11 dBm
  - Excellent Blocking Immunity
  - Low RX Current of 17 mA (typical)
  - Fully Integrated Synthesizer with a Resolution of 61 Hz
  - LoRa Technology, (G)FSK, (G)MSK and OOK Modulation
  - Preamble Detection
  - 127 dB Dynamic Range RSSI
  - Automatic RF Sense and CAD with Ultra-Fast Automatic Frequency Control (AFC) Packet Engine up to 256 bytes with Cyclic Redundancy Check (CRC)

- Peripherals
  - 12-channel Direct Memory Access Controller (DMAC)
  - 12-channel Event System
  - Up to Three 16-bit Timer/Counters (TC), configurable as either:
  - One 8-bit TC with compare/capture channels

- One 16-bit TC with compare/capture channels
- One 32-bit TC with compare/capture channels, by using two TCs
- Three 16-bit Timer/Counters for Control (TCC), with extended functions:
  - Up to four compare channels with optional complementary output
  - Generation of synchronized pulse width modulation (PWM) pattern across port pins,
  - Deterministic fault protection, fast decay and configurable dead-time between complementary output
  - Dithering that increases resolution with up to 5 bit and reduce quantization error
- 32-bit Real Time Counter (RTC) with clock/calendar function
- Watchdog Timer (WDT)
- CRC-32 generator
- One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface with Device and embedded Host and 8 endpoints
- Five Serial Communication Interfaces (SERCOM), each configurable to operate as either:
  - USART with full-duplex and single-wire half-duplex configuration
  - I2C Bus up to 3.4MHz
  - SPI
  - LIN slave
- One 12-bit, 1 Msps Analog-to-Digital Converter (ADC) with up to 8 external channels
  - Differential and single-ended input
  - 1/2x to 16x programmable gain stage
  - Automatic offset and gain error compensation
  - Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution
- Two Analog Comparators (AC) with window compare function
- Peripheral Touch Controller (PTC) with 48-channel capacitive touch and proximity sensing I/O
- 27 GPIO pins

Features of the ATSAML21:

- Processor
  - ARM©, Cortex-M0+ CPU running at up to 48MHz
  - Single-cycle hardware multiplier
  - Micro Trace Buffer

- Memories
  - 256KB in-system self-programmable Flash
  - 32KB SRAM Memory

- System
  - Power-on reset (POR) and brown-out detection (BOD)
  - Internal and external clock options with 48MHz Digital Frequency Locked Loop (DFLL48M) and 48MHz to 96MHz fractional
  - External Interrupt Controller (EIC), 16 external interrupts, one non-maskable interrupt
  - Two-pin Serial Wire Debug (SWD) programming, test and debugging interface

- Low Power
  - Idle, stand-by, backup and Off sleep modes
  - SleepWalking peripherals
  - Static and Dynamic Power Gating Architecture
  - Battery backup peripherals
  - Two performance levels
  - Embedded Buck/LDO regulator supporting on-the-fly selection

- Temperature Range: -40°C to + 85°C (Industrial)

- Peripherals
  - 16-channel Direct Memory Access Controller (DMAC)
  - 12-channel Event System
  - Up to five 16-bit Timer/Counters (TC) (one low-power TC), configurable as either:
  - One 16-bit TC with compare/capture channels
  - One 8-bit TC with compare/capture channels
  - One 32-bit TC with compare/capture channels, by using two TCs
  - Two 24-bit and one 16-bit Timer/Counters for Control (TCC), with extended functions:
    - Up to four compare channels with optional complementary output
    - Generation of synchronized pulse width modulation (PWM) pattern across port pins,
    - Deterministic fault protection, fast decay and configurable dead-time between complementary output
    - Dithering that increase resolution with up to 5 bit and reduce quantization error
  - 32-bit Real Time Counter (RTC) with clock/calendar function
  - Watchdog Timer (WDT)
  - CRC-32 generator
  - One full-speed (12Mbps) Universal Serial Bus (USB) 2.0 interface with Device and embedded Host and 8 endpoints
  - Six Serial Communication Interfaces (SERCOM) (one low-power SERCOM), each configurable to operate as either:
    - USART with full-duplex and single-wire half-duplex configuration
    - I2C Bus up to 3.4MHz
    - SPI
    - LIN slave
  - One AES encryption engine
  - One True Random Generator (TRNG)
  - One Configurable Custom Logic (CGL)
  - One 12-bit, 1 Msps Analog-to-Digital Converter (ADC) with up to 20 channels
    - Differential and single-ended input
    - 1/2x to 16x programmable gain stage
    - Automatic offset and gain error compensation

    - Oversampling and decimation in hardware to support 13-, 14-, 15- or 16-bit resolution
- Two 12-bit, 1Msps dual output Digital-to-Analog Converter (DAC)
- Two Analog Comparators (AC) with window compare function
- Three Operational Amplifiers (OPAMP)
- Peripheral Touch Controller (PTC) with 169-channel capacitive touch and proximity sensing I/O, wake up on touch in Stand-by mode
- 38 GPIO pins

Features of the ATECC608B:

- Cloud authentication for AWS IoT
- Cloud authentication for Google Cloud IoT Core
- Secure Boot implementation with an ATSAML21 Cortex-M0+
- Cryptographic coprocessor with secure hardware-based key storage
  - Protected storage for up to 16 Keys, certificates or data
- Hardware support for asymmetric sign, verify, key agreement – ECDSA: FIPS186-3 Elliptic Curve Digital Signature
- ECDH: FIPS SP800-56A Elliptic Curve Diffie-Hellman
- NIST standard P256 elliptic curve support
- Hardware support for symmetric algorithms
  - SHA-256 & HMAC hash including off-chip context save/restore
  - AES-128: encrypt/decrypt, galois field multiply for GCM
- Networking key management support
  - Turnkey PRF/HKDF calculation for TLS 1.2 & 1.3
  - Ephemeral key generation and key agreement in SRAM – Small message encryption with keys entirely protected
- Secure boot support
  - Full ECDSA code signature validation, optional stored digest/signature – optional communication key disablement prior to secure boot
  - Encryption/Authentication for messages to prevent on-board attacks
- Internal high-quality FIPS 800-90 A/B/C Random Number Generator (RNG)
- Two high-endurance monotonic counters
- Guaranteed unique 72-bit serial number
- Two interface options available
  - High-speed single pin interface with One GPIO pin
  - 1MHz Standard I2C interface
- 1.8V to 5.5V IO levels, 2.0V to 5.5V supply voltage
- <150nA Sleep current

Security

A key point of IoT and IIoT is security. Since many industrial companies are extensively implementing the Internet of Things at the edge of their networks and increasing the capabilities of the network itself, connecting many devices presents a huge security threat. This big number of connected devices offers a much larger surface prone to cyber-attack than the IT space where, by comparison, the volumes of data are lower, and its exchanging can be more precisely controlled.

In the industrial sector, huge amounts of data are being processed by physical devices at the edge (through their firmware), sent back to the cloud for further analysis and used by different applications or, after processing, by the devices themselves to control the processes or the plant environment. Attackers can exploit these devices and their software to subvert and compromise the hardware itself. Significant numbers of IoT devices are not being used with security in mind, so every single device and sensor in the IoT represent a potential risk and an easy entry point to the network.

The MBC-LR01 thanks to its embedded crypto auth chip can ensure a good security level.

The Microchip ATECC608B that integrates ECDH (Elliptic Curve Diffie Hellman) security protocol - an ultra-secure method to provide key agreement for encryption/decryption - along with ECDSA (Elliptic Curve Digital Signature Algorithm) sign-verify authentication.

It also offers an integrated AES hardware accelerator ensuring the secure boot features to both the MBC's MCU and supplying the full range of security such as confidentiality, data integrity, and authentication to MBC's system. The ATECC608B employs ultra-secure hardware-based cryptographic key storage and cryptographic countermeasures which eliminate potential backdoors linked to software weaknesses.

It can also perform Elliptic Curve Diffie Hellman Key Exchange which means that the part can securely store the asymmetric keys (private key) for a TLS (v. 1.3) exchange and deliver the master secret to the microcontroller for the symmetric portions of the protocol, simplifying the connection and the authentication to Azure, AWS, and Google cloud platforms.

Ordering information

TBD

USB Interface

The Universal Serial Bus (USB) module of the SAML21/SAMR34, complies with the Universal Serial Bus (USB) 2.1 specification, supporting both device and embedded host modes. It supports full (12Mbit/s) and low (1.5Mbit/s) speed communication and Link Power Management (LPM-L1) protocol. More information can be found in the devices' datasheet.

By default, and in case both the MCUs are populated, the firmware implementation of the MBC-LR comes with a Dual CDC interface over the same physical USB interface: one for the SAML21 itself and one for the SAMR34, leaving PA24 and PA25 of the latter one, free for the user convenience. For the SAMR34, the SAML21 acts as a USB-to-UART transceiver allowing a Host PC to access the UART interface the two MCUs share, both for communication and programming. To modify this configuration, read the section SAML21 CDC configuration.

In case only the SAMR34 is populated on the module (for more information on the versions available, please read the Ordering Information section of this manual), the external USB interface is directly tied to the SAMR34 USB interface. In this case a single CDC is implemented since no companion chip is available.

On Windows, the USB driver discriminates between the two different interfaces embedded in the device descriptor of the MBC-LR. This leads to the enumeration of two separated USB ports: BRIKI MBC-LR (saml) for the SAML21 and BRIKI MBC-LR (samr) for the SAMR34.

In case a previous version of the driver has already been installed, the port names will not be displayed correctly. The suggested procedure is to connect the MBC-LR with an USB cable to the host PC, enter the Device Manager and, under the COM Port list, select the port belonging to the MBC. Now by clicking the right mouse button, select "Device uninstall" from the drop-down menu. Once the procedure has finished, disconnect the USB cable, launch the driver installer contained in the platform installation directory under the "driver" folder, and finally connect the board again. If the names are still not correct it is possible that the firmware on the target board is not updated to the last version. In that case, just simply upload a new firmware to the SAML21/SAMR34 using one of the supported IDEs.

On other operating systems, such as Linux or MacOS, both ports show the same descriptor, and thus there is no way to distinguish them at a glance; usually, with the two-chip module version, the SAML21 port is enumerated as first and the SAMR34 virtual port as second. So, the general rule to recognize which is which, is to look at the final number of the attached port. For example, under Linux, if the ports are enumerated as ttyACM0 and ttyACM1, it is likely they are referred to SAML21 and SAMR34 respectively.

A simple firmware that prints a different message on both the MCUs can be loaded to check how your system enumerated the two ports.

Regarding the firmware update via USB bootloader, it will be performed anyway, independently by which of the two ports has been selected for. This means that the SAML21 can be programmed even if the SAMR34 port has been selected and vice versa. To better understand how the firmware update process works over the USB interface, please read the USB bootloader section.

SAML21 USB-to-serial transceiver function

As already explained, in the dual-core MBC-LR version, the SAML21 virtualizes a secondary CDC interface for the SAMR34 over its own physical USB interface. By default, it performs this job in the background, along with the handling of the other messages exchanged with the companion MCU to enable the GPIO virtualization and the Communication functions.

Since the SAML21 comes with a single-thread firmware implementation (there is no RTOS that allows multi-threading), the execution time of the code loop is affected by the execution time of each task it embeds. Thus, if a task requires a long time to be executed, this will have a negative impact on the overall system performances and consequently also on those tasks that runs in background.

To speed up the execution of all background tasks, the function communication.handleCommEvents() can be added in those functions that require a long time to end. This function will check for messages to be forwarded to/from USB as well as any other communication event that needs to be processed.

When the double USB-CDC interface is enabled, the communication between the two MCUs is performed via the shared Low-Power UART interface. To disable the secondary CDC interface, please read the section SAML21 CDC configuration.

Depending on the hardware version, in case both the MCUs are present on the board, the SAML21 is capable of programming both, in case the only SAMR34 is present, it can only program itself.

In case a single CDC interface is enabled, the SAMR34's serial interface can be mirrored on the SAML21 USB interface using the UartBridge example supplied.

Hardware communication interface

The MBC-LR features one communication interface based on the SERCOM5 low-power UART, plus some control and synchronization signals shared among the two MCUs.

UART interface

The maximum throughput achievable is approximately 19 kBps.

Table 3 - SAML21-SAMR34 shared UART interface signals

Pin # (HW)	Pin # (FW)	Function	Notes	Direction (default)
PA24 (SAMR34) **PB23 (SAML21)**	5 (SAMR34) 51 (SAML21)	SAMR34 UART TX / SAML21 UART RX	In SAMR34 this GPIO is shared with USB interface, so if USB is active on SAMR34 → no UART	SAML21 ← SAMR34
PA25 (SAMR34) **PB22 (SAML21)**	6 (SAMR34) 50 (SAML21)	SAMR34 UART RX / SAML21 UART TX	In SAMR34 this GPIO is shared with USB interface, so if USB is active on SAMR34 → no UART	SAML21 → SAMR34
PB03 (SAMR34) **RESET SAML21**	59 (SAMR34)	SAML21 reset control	LOW = L21 Reset, HIGH = L21 Run Direction cannot be swapped	SAML21 ← SAMR34
PA15 (SAMR34) **PA18 (SAML21)**	45 (SAMR34) 48 (SAML21)	Companion EVT	This GPIO can be used as a sinc signal between the two MCU Direction swappable	SAML21 ↔ SAMR34
PA14 (SAMR34) **PA19 (SAML21)**	44 (SAMR34) 49 (SAML21)	Internal EVT	This GPIO can be used as a sinc signal between the two MCU Direction swappable	SAML21 ↔ SAMR34
PA20 (SAML21) **RESET SAMR34**	42 (SAML21)	SAMR34 reset control	LOW = R34 Reset, HIGH = R34 Run Direction cannot be swapped	SAML21 → SAMR34
PB22 (SAMR34)	41 (**SAMR34**)	SAML21 power control	Hi-Z / HIGH = L21 power on, LOW = L21 power off Direction cannot be swapped	SAML21 ← SAMR34

Firmware platform

Installation prerequisites

To simplify the platform installation, Meteca provides an executable installer for all the main OSs. It contains the required drivers (only for the Windows version), the firmware platform as well as all the software tools needed for configuring and programming the MBC-LR.

Before launching the installer, one of the two supported IDEs must be installed: the Arduino IDE or Visual Studio Code with the PlatformIO plugin. The former is more suited for beginners while the latter is recommended as a more professional and complete IDE

In case of any issue with the installer, or in case a manual installation is required, please check the manual installation section.

Arduino platform automatic installation

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

Once the Arduino IDE has been installed, download the MBC-LR platform installer for the preferred OS:

Windows:	https://www.briki.org/download/resources/briki_arduino_installer.exe
Linux 64 bit:	https://www.briki.org/download/resources/briki_arduino_installer_linux64.tar.gz
Linux 32 bit:	https://www.briki.org/download/resources/briki_arduino_installer_linux32.tar.gz
MAC OSX:	https://www.briki.org/download/resources/briki_arduino_installer_osx.tar.gz

Before launching the installer, make sure that:

- the Arduino IDE is closed, or it will overwrite the settings the installer makes
- the Host PC has a stable connection to the internet since the installer will download the platform from the web
- on Linux and Mac systems the downloaded archive has been extracted before launching it

When ready, the installation can be started by double clicking on the installer executable.

Under Linux and Mac OS, in case of issues with the installer when launched using the UI, a terminal can be opened to drag and drop the file there. Pressing the enter key will then start the installation.

For Linux users: to complete the installation, serial port permission must be set. Please, read this reference or run arduino-linux-setup.sh script, that can be found in the Arduino IDE folder.

Drivers installation

Using Linux or Mac, no driver installation is required to start using an MBC-LR board. On the other hand, Windows requires the installation of the proper USB drivers to make the Host PC interact with the board through the USB connection. Drivers can be found in the MBC-LR installation directory, under “drivers” folder. The path should look like the following:

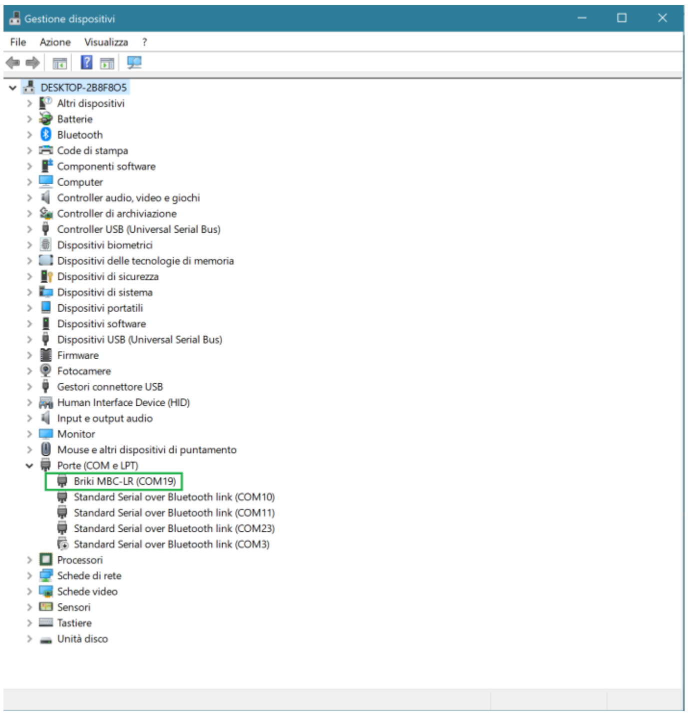
C:\Users\CurrentUser\AppData\Local\Arduino15\packages\briki\hardware\MBC-LR\x.x.x\drivers

Search this folder for the file *briki_drv_installer.exe* and execute it to start the driver installation. After the process is completed, the MBC-LR board can be connected to the Host PC, and the associated COM ports should be listed under the Device Manager. You can use one of these ports to interact with your board. The number of the available ports depends on the firmware loaded in the SAML21 as depicted in this section.

To check the correct installation of drivers, it is recommended to try loading a sketch on the board (on both the SAML21 and the SAMR34)

Troubleshooting with the driver’s installation

If the sketch upload fails, the first thing to do is check if the drivers are installed correctly. Open the Windows Device Manager on the Host PC with the board connected via USB and check if a window like the following is prompted:



```
DESKTOP-2B8F8O5
  Altri dispositivi
  Batterie
  Bluetooth
  Code di stampa
  Componenti software
  Computer
  Controller audio, video e giochi
  Controller di archiviazione
  Controller USB (Universal Serial Bus)
  Dispositivi biometrici
  Dispositivi delle tecnologie di memoria
  Dispositivi di sicurezza
  Dispositivi di sistema
  Dispositivi portatili
  Dispositivi software
  Dispositivi USB (Universal Serial Bus)
  Firmware
  Fotocamere
  Gestori connettore USB
  Human Interface Device (HID)
  Input e output audio
  Monitor
  Mouse e altri dispositivi di puntamento
  Porte (COM e LPT)
    Briki MBC-LR (COM19)
    Standard Serial over Bluetooth link (COM10)
    Standard Serial over Bluetooth link (COM11)
    Standard Serial over Bluetooth link (COM23)
    Standard Serial over Bluetooth link (COM3)
  Processori
  Schede di rete
  Schede video
  Sensori
  Tastiere
  Unità disco
```

Figure 7 - Windows driver's troubleshooting

To invoke the Windows Device Manager on Windows 10:

- press W + X and select Device Manager
- from a command prompt write devmgmt.msc and press enter
- click the start button on the Taskbar and search for devmgmt.msc

In case the Briki MBC-LR board is not listed under the Ports (COM & LPT) – like in the following picture, it probably means the drivers install procedure went wrong.

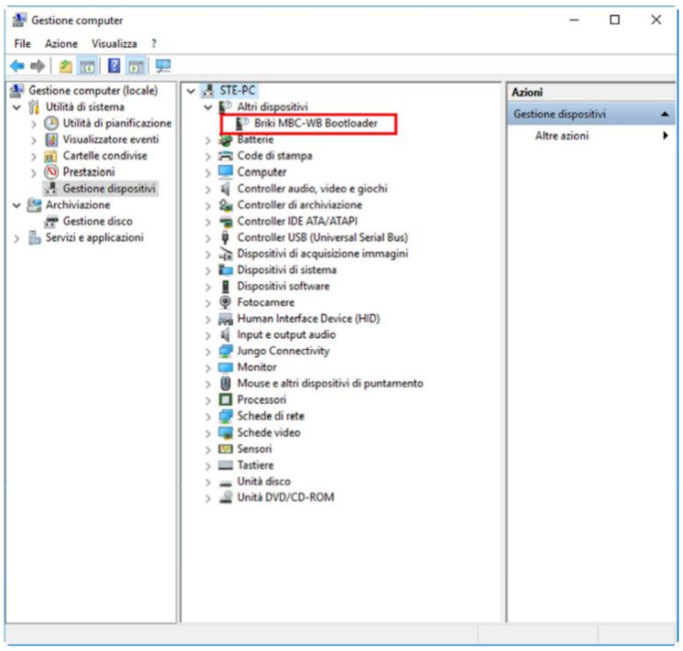


Figure 8 - Windows driver's Installation

In this case, try to manually install the drivers to fix the problem. From the Device Manager window, open Briki MBC-LR device properties (right click on Briki MBC-LR and then select Properties). From the popup window that will appear, select the *Driver* tab and then click on the *Update Driver*. When prompted, select the *Browse my computer for driver software* option. Then search for the driver's folder of MBC-LR platform (the same as the one where briki_drv_installer.exe is located) to start the installation.

At the end of the installation procedure, try to unplug and re-plug in the board on the Host PC and check if a new COM Port has now been correctly enumerated.

Software

mbctool

The mbctool is a command line executable written in Python. A copy of this tool can be downloaded for Windows, Linux and Mac. It is the only software that must be used to perform a firmware update on the MBC-LR, whatever OS or IDE is used. It is based on bossac and its tasks are:

- to generate the triggering event to invoke the SAML21/SAMR34 bootloader; this is performed by opening and closing the USB connection at a given baud rate and with a specific timing. Part of the code, that underlies the application running in the SAML21/SAMR34, is always listening for this event on the USB interface.
- to establish the communication with the bootloader, specifying which is the target device for the upload process SAML21 or SAMR34.
- wait for the MBC to be ready and finally launch bossac for both the SAML21 and the SAMR34 that will upload the firmware.

Therefore, a copy of bossac command line applications is placed in the same folder of mbctool

SAML21 programming

To load a binary on the SAML21 using the Command Line Interface, the following command can be used:

```
mbctool -d <saml> -p <port> -u <firmware.bin>
```

where: -d specifies the target device (--device); -p specifies the USB port the board is connected to (--port); -u specifies the file to be loaded (--upload).

SAMR34 programming

To load a binary on the SAMR34 using the Command Line Interface, the following command can be used:

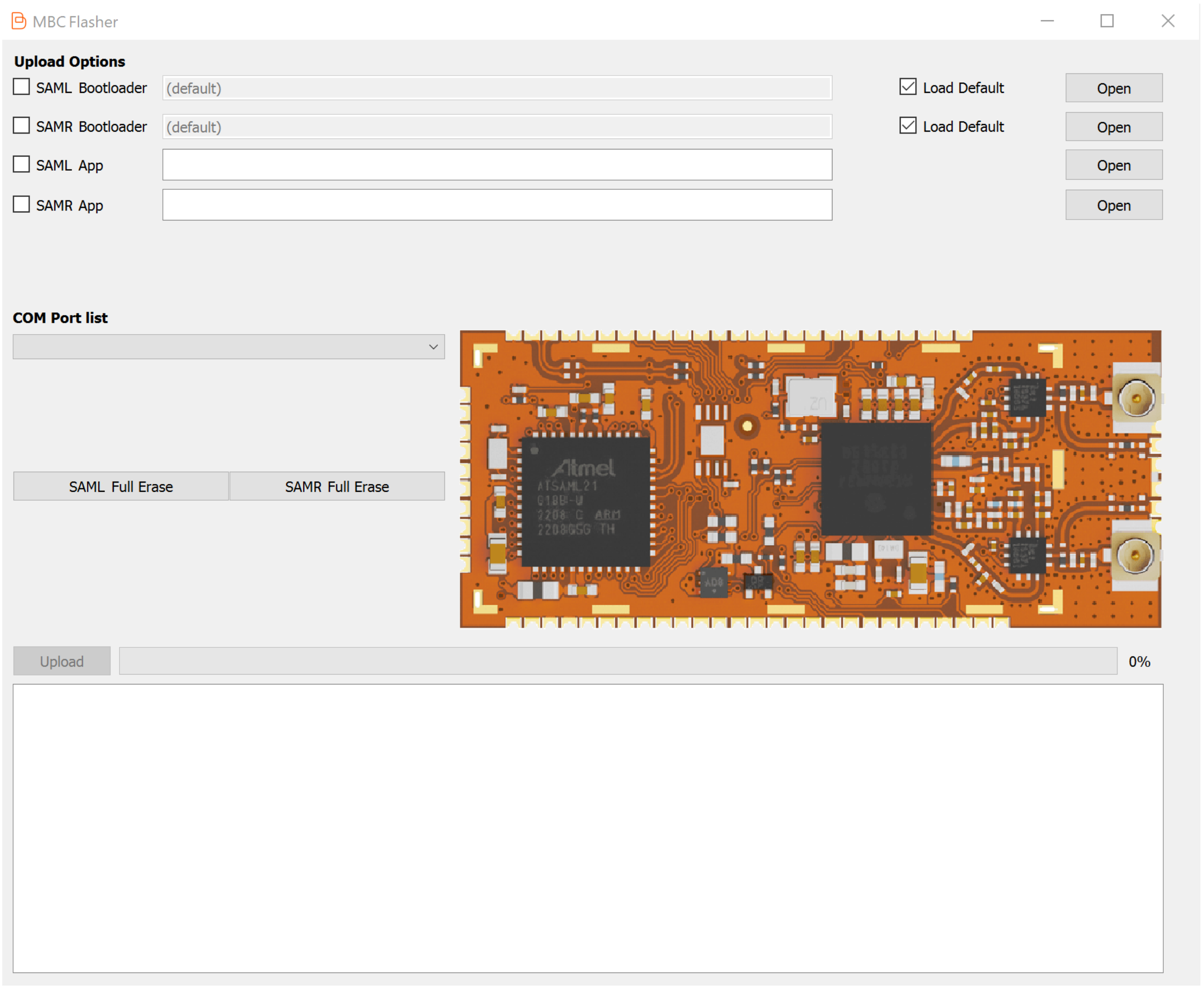
```
mbctool -d <samr> -p <port> -u <firmware.bin>
```

where: -d specifies the target device (--device); -p specifies the USB port the board is connected to (--port); -u specifies file to be loaded (--upload).

Several firmwares are available, both for test purposes and as a reference for the production release.

For testing purposes, a software (MBC Flasher) is provided to perform the firmware update procedure, allowing the tester to load the correct binary required for the specific testing phase.

MBC Flasher



Using one of the buttons on the right is it possible to choose the binary to load into the corresponding MCU. Once the binary has been chosen, its path and name become visible in the related textbox. Enabling the radio-button on the left, that specific binary will be loaded to the related MCU.

The COM Port list menu allows the operator to choose what COM port could be used to start the upload process. In the specific case of this board, the only SAMR34 USB port will be listed and displayed since the SAML21 is not populated.

Once the port has been chosen, clicking on Upload button will start the upload process. The process log can be read out using the textbox below.

Testing Software

To test the module several binaries and executables/python files has been prepared.

All the required material is contained in a zip file named: *mbc_lr_testing_software.zip*

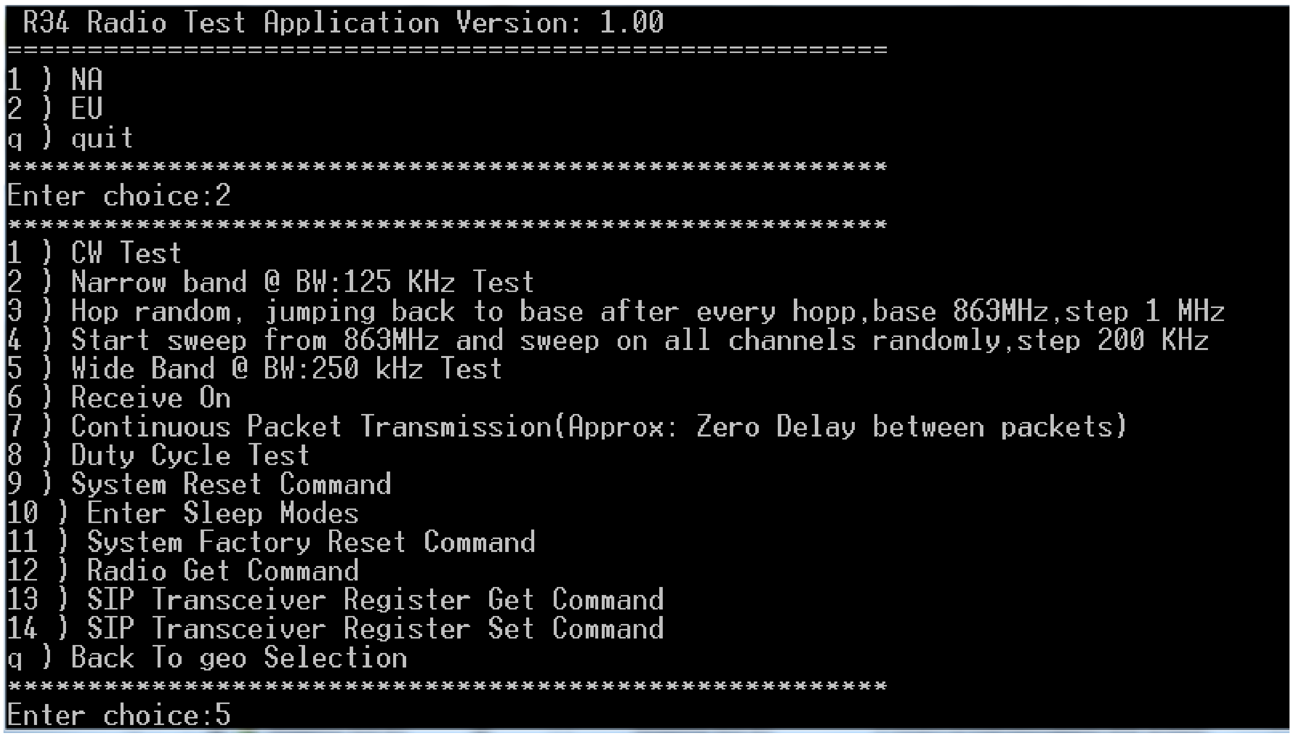
This file contains the following folders:

- bin – that contains all the binaries that must be uploaded to the module to test it
- testing software – that contains the executables and the python test software that must run on the Host PC
- tools – that contains the previously mentioned MBC-LR-Flasher, needed to upload the firmware

The module is pre-configured with a serial-to-usb bridge (*SAML_bridge.bin*) firmware uploaded into the SAML21, and the RadioUtility firmware (*MBC-LR_full_version_Radio_Utility.bin*) uploaded to the SAMR34.

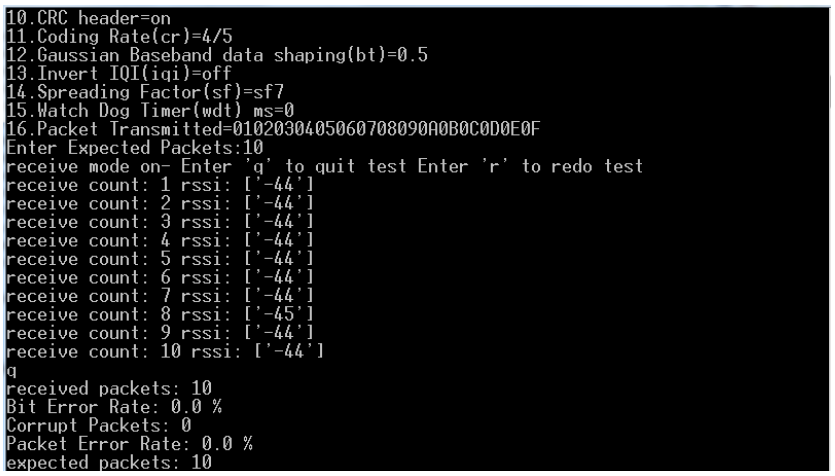
To effectively use the RadioUtility firmware, two executables that must run on Windows PCs have been produced. These two software control the execution of the firmware through instructions given by the operator and selected from a text menu shown by the software itself during all operational phases.

The picture below shows how the *"MBC-LR_RF_TransmitterTests.exe"* text menu appears, where the first 9-10 choices are the most useful for your testing procedure.

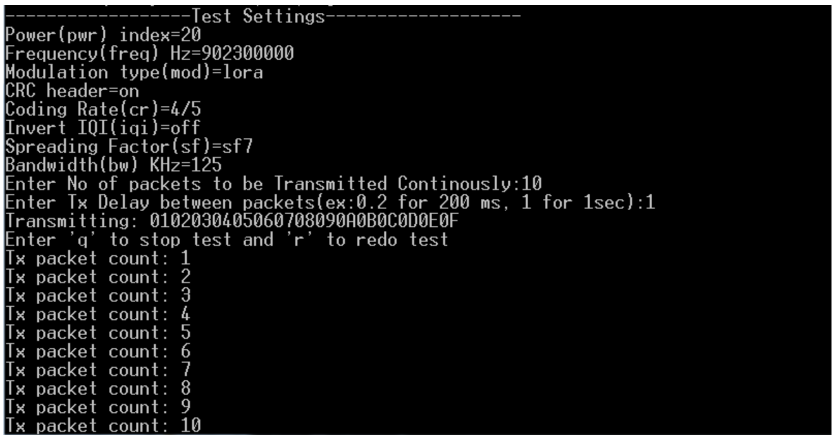


```
 R34 Radio Test Application Version: 1.00
=======================================================
1 ) NA
2 ) EU
q ) quit
*******************************************************
Enter choice:2
*******************************************************
1 ) CW Test
2 ) Narrow band @ BW:125 KHz Test
3 ) Hop random, jumping back to base after every hopp,base 863MHz,step 1 MHz
4 ) Start sweep from 863MHz and sweep on all channels randomly,step 200 KHz
5 ) Wide Band @ BW:250 kHz Test
6 ) Receive On
7 ) Continuous Packet Transmission(Approx: Zero Delay between packets)
8 ) Duty Cycle Test
9 ) System Reset Command
10 ) Enter Sleep Modes
11 ) System Factory Reset Command
12 ) Radio Get Command
13 ) SIP Transceiver Register Get Command
14 ) SIP Transceiver Register Set Command
q ) Back To geo Selection
*******************************************************
Enter choice:5
```

Beside the "*MBC-LR_RF_TransmitterTests.exe*" another software has been released: the "*MBC-LR_RF_ReceiverTests.exe*". This second software controls a second MBC_LR module running the same RadioUtility firmware to allow communication between two MBC-LR modules; one module configured as a transmitter using the "*MBC-LR_RF_TransmitterTests.exe*" as a controller and the second one configured as a receiver using the "*MBC-LR_RF_ReceiverTests.exe*" as a controller.


```
10.CRC header=on
11.Coding Rate(cr)=4/5
12.Gaussian Baseband data shaping(bt)=0.5
13.Invert IQI(iqi)=off
14.Spreading Factor(sf)=sf7
15.Watch Dog Timer(wdt) ms=0
16.Packet Transmitted=0102030405060708090A0B0C0D0E0F
Enter Expected Packets:10
receive mode on- Enter 'q' to quit test Enter 'r' to redo test
receive count: 1 rssi: ['-44']
receive count: 2 rssi: ['-44']
receive count: 3 rssi: ['-44']
receive count: 4 rssi: ['-44']
receive count: 5 rssi: ['-44']
receive count: 6 rssi: ['-44']
receive count: 7 rssi: ['-44']
receive count: 8 rssi: ['-45']
receive count: 9 rssi: ['-44']
receive count: 10 rssi: ['-44']
q
received packets: 10
Bit Error Rate: 0.0 %
Corrupt Packets: 0
Packet Error Rate: 0.0 %
expected packets: 10
```

Receiver side


```
------------------Test Settings-------------------
Power(pwr) index=20
Frequency(freq) Hz=902300000
Modulation type(mod)=lora
CRC header=on
Coding Rate(cr)=4/5
Invert IQI(iqi)=off
Spreading Factor(sf)=sf7
Bandwidth(bw) KHz=125
Enter No of packets to be Transmitted Continously:10
Enter Tx Delay between packets(ex:0.2 for 200 ms, 1 for 1sec):1
Transmitting: 0102030405060708090A0B0C0D0E0F
Enter 'q' to stop test and 'r' to redo test
Tx packet count: 1
Tx packet count: 2
Tx packet count: 3
Tx packet count: 4
Tx packet count: 5
Tx packet count: 6
Tx packet count: 7
Tx packet count: 8
Tx packet count: 9
Tx packet count: 10
```

Transmitter side

Beside this firmware – software couple, a "production" binary has been prepared: "*SAMR_Production.bin*".

This firmware runs on the SAMR34 MCU in parallel with the "*SAML_bridge.bin*" running on the SAML21 MCU.

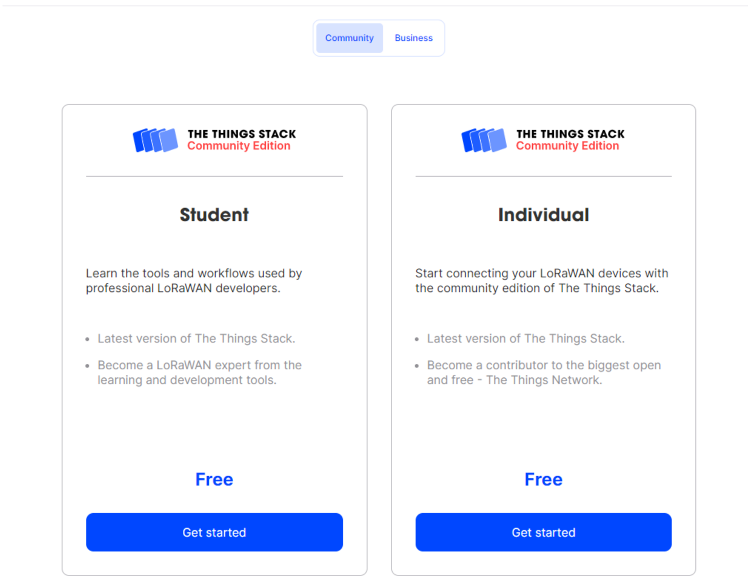
It allows the user to issue a set of parameters to connect the module to the "TTN – The Things Network" cloud service.

To register your device, you'll first need to create an account in The Things Network.

1. Go to account.thethingsnetwork.org and click create an account.
   You will receive an email to confirm your email address. You have 24 hours to do so, so let's wait for the mail and carry on! 🚀

   You can change all but your username later via your Profile.

2. Select Console from the top menu.
3. From the top right menu, select your name and then Settings. Then change the default Handler if the one currently selected is not where you'll be deploying most of your devices.


Community Business

THE THINGS STACK Community Edition

Student

Learn the tools and workflows used by professional LoRaWAN developers.

- Latest version of The Things Stack.
- Become a LoRaWAN expert from the learning and development tools.

Free

Get started

THE THINGS STACK Community Edition

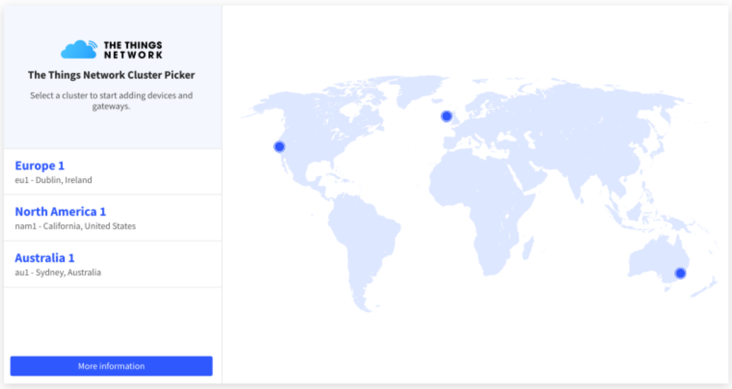
Individual

Start connecting your LoRaWAN devices with the community edition of The Things Stack.

- Latest version of The Things Stack.
- Become a contributor to the biggest open and free - The Things Network.

Free

Get started


THE THINGS NETWORK

The Things Network Cluster Picker

Select a cluster to start adding devices and gateways.

Europe 1
eu1 - Dublin, Ireland

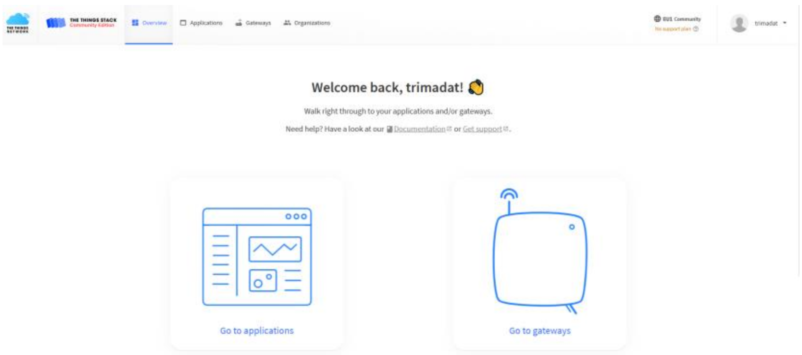
North America 1
nam1 - California, United States

Australia 1
au1 - Sydney, Australia

More information

Then an application must be added to the account.

1. For Application ID, choose a unique ID of lower case, alphanumeric characters and nonconsecutive - and _.
2. For Description, enter anything you like.
3. Leave the Handler registration on the default selected value, unless you are going to use this application in a different region than your default region



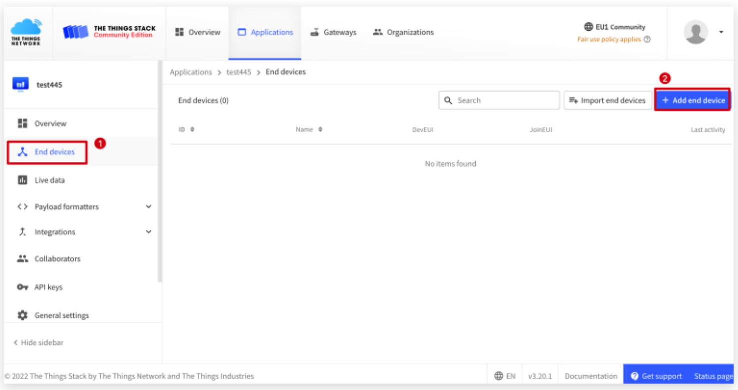
Welcome back, trimadat!

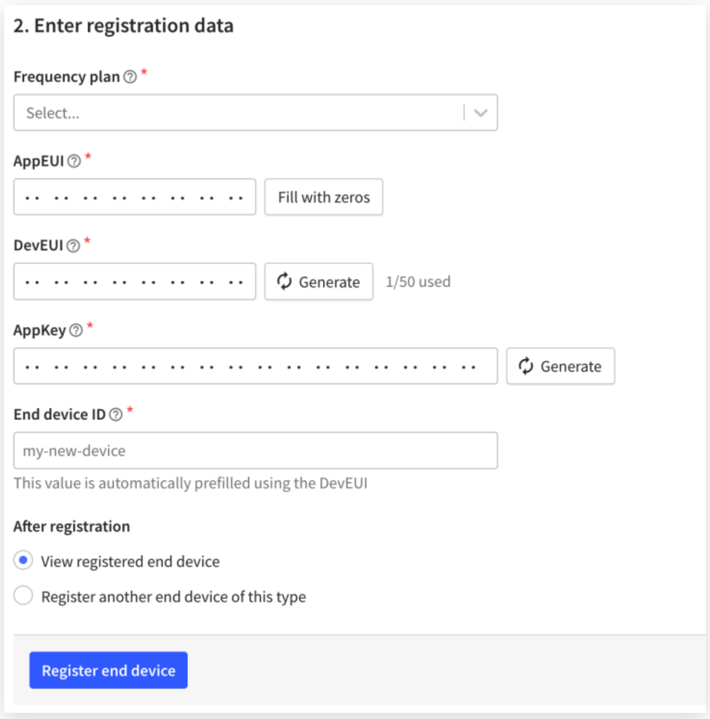
Walk right through to your applications and/or gateways.

Need help? Have a look at our Documentation or Get support.

Finally, to communicate via The Things Network, you need to register your device.

1. On the application's page, scroll down to Devices or select Devices from the top right menu.
2. In the Devices box, click register device.
    a. For Device ID, choose an ID of lower case, alphanumeric characters and nonconsecutive - and _. A device ID needs to be unique per application but can be reused in another application.
    b. For Device EUI, copy-paste the DevEUI you retrieved from your device (in our case it is 0x70, 0xB3, 0xD5, 0x7E, 0xD0, 0x05, 0x58, 0x45). By the way since the SAMR34 is programmable, you could create an EUI using the Generate button and then issue it using the serial interface offered by the production firmware.
    c. Leave the App Key on "this field will be generated".
    d. Leave the default App EUI selected.





2. Enter registration data

Frequency plan *
Select...

AppEUI *
Fill with zeros

DevEUI *
Generate 1/50 used

AppKey *
Generate

End device ID *
my-new-device
This value is automatically prefilled using the DevEUI

After registration
- [✓ View registered end device]
- Register another end device of this type

Register end device

For more information related to the quick start, please refers to: [Quick Start | The Things Network]

Electrical characteristics

Table 4 - Electrical characteristics

Symbol	Description	Min.	Typ.	Max.	Unit
VCC_SAML	Input supply voltage for SAML21	1.62	3.3	3.63	V
VCC_SAMR	Input supply voltage for SAMR34	1.8	3.3	3.63	V
ICC_SAML	Current absorption from SAML21	510n	-	5m	A
ICC_SAMR	Current absorption from SAMR34	790n	-	95m	A
OPE_T	Operating temperature	-40		80	°C

2.2 List of applicable FCC rules
List the FCC rules that are applicable to the modular transmitter. These are the rules that specifically establish the bands of operation, the power, spurious emissions, and operating fundamental frequencies.
DO NOT list compliance to unintentional-radiator rules (Part 15 Subpart B) since that is not a condition of a module grant that is extended to a host manufacturer. See also Section 2.10 below concerning the need to notify host manufacturers that further testing is required.3
Explanation: This module meets the requirements of Part 15 Subpart C Section 15.249

2.3 Summarize the specific operational use conditions
Describe use conditions that are applicable to the modular transmitter, including for example any limits on antennas, etc. For example, if point-to-point antennas are used that require reduction in power or compensation for cable loss, then this information must be in the instructions. If the use condition limitations extend to professional users, then instructions must state that this information also extends to the host manufacturer's instruction manual. In addition, certain information may also be needed, such as peak gain per frequency band and minimum gain, specifically for master devices in 5 GHz DFS bands.
Explanation: The EUT uses ISM Antenna, antenna gain: -0.3dBi. There is no restriction on the installation method.

2.4 Limited module procedures
If a modular transmitter is approved as a "limited module", then the module manufacturer is responsible for approving the host environment that the limited module is used with. The manufacturer of a limited module must describe, both in the filing and in the installation instructions, the alternative means that the limited module manufacturer uses to verify that the host meets the necessary requirements to satisfy the module limiting conditions.
A limited module manufacturer has the flexibility to define its alternative method to address the conditions that limit the initial approval, such as: shielding, minimum signaling amplitude, buffered modulation/data inputs, or power supply regulation. The alternative method could include that the limited module manufacturer reviews detailed test data or host designs prior to giving the host manufacturer approval.
This limited module procedure is also applicable for RF exposure evaluation when it is necessary to demonstrate compliance in a specific host. The module manufacturer must state how control of the product into which the modular transmitter will be installed will be maintained such that full compliance of the product is always ensured. For additional hosts other than the specific host originally granted with a limited module, a Class II permissive change is required on the module grant to register the additional host as a specific host also approved with the module.
Explanation: The module is not limited module.

2.5 Trace antenna designs
For a modular transmitter with trace antenna designs, see the guidance in Question 11 of KDB Publication 996369 D02 FAQ – Modules for Micro-Strip Antennas and traces. The integration information shall include for the TCB review the integration instructions for the following aspects:

layout of trace design, parts list (BOM), antenna, connectors, and isolation requirements.4
a) Information that includes permitted variances (e.g., trace boundary limits, thickness, length, width, shape(s), dielectric constant, and impedance as applicable for each type of antenna);
b) Each design shall be considered a different type (e.g., antenna length in multiple(s) of frequency, the wavelength, and antenna shape (traces in phase) can affect antenna gain and must be considered);
c) The parameters shall be provided in a manner permitting host manufacturers to design the printed circuit (PC) board layout;
d) Appropriate parts by manufacturer and specifications;
e) Test procedures for design verification; and
f) Production test procedures for ensuring compliance.
The module grantee shall provide a notice that any deviation(s) from the defined parameters of the antenna trace, as described by the instructions, require that the host product manufacturer must notify the module grantee that they wish to change the antenna trace design. In this case, a Class II permissive change application is required to be filed by the grantee, or the host manufacturer can take responsibility through the change in FCC ID (new application) procedure followed by a Class II permissive change application.
Explanation: Yes. The module with trace antenna designs.

2.6 RF exposure considerations
It is essential for module grantees to clearly and explicitly state the RF exposure conditions that permit a host product manufacturer to use the module. Two types of instructions are required for RF exposure information: (1) to the host product manufacturer, to define the application conditions (mobile, portable – xx cm from a person’s body); and (2) additional text needed for the host product manufacturer to provide to end users in their end-product manuals. If RF exposure statements and use conditions are not provided, then the host product manufacturer is required to take responsibility of the module through a change in FCC ID (new application).
Explanation: This module complies with FCC RF radiation exposure limits set forth for an uncontrolled environment. The device is mobile, and the use distance is 20 cm. This module is designed to comply with the FCC statement, FCC ID: 2ATX7-MBC-LR01.

2.7 Antennas
A list of antennas included in the application for certification must be provided in the instructions. For modular transmitters approved as limited modules, all applicable professional installer instructions must be included as part of the information to the host product manufacturer. The antenna list shall also identify the antenna types (monopole, PIFA, dipole, etc. (note that for example an “omni-directional antenna” is not considered to be a specific “antenna type”)).
For situations where the host product manufacturer is responsible for an external connector, for Example with an RF pin and antenna trace design, the integration instructions shall inform the installer that unique antenna connector must be used on the Part 15 authorized transmitters used in the host product. The module manufacturers shall provide a list of acceptable unique connectors.
Explanation: The EUT uses ISM Antenna, antenna gain: -0.3dBi.

2.8 Label and compliance information
Grantees are responsible for the continued compliance of their modules to the FCC rules. This includes advising host product manufacturers that they need to provide a physical or e-label stating “Contains FCC ID” with their finished product. See Guidelines for Labeling and User Information for RF Devices – KDB Publication 784748.
Explanation: The host system using this module, should have label in a visible area indicated he following texts: "Contains FCC ID: 2ATX7-MBC-LR01

2.9 Information on test modes and additional testing requirements5
Additional guidance for testing host products is given in KDB Publication 996369 D04 Module Integration Guide. Test modes should take into consideration different operational conditions for a standalone modular transmitter in a host, as well as for multiple simultaneously transmitting modules or other transmitters in a host product.
The grantee should provide information on how to configure test modes for host product evaluation for different operational conditions for a stand-alone modular transmitter in a host, versus with multiple, simultaneously transmitting modules or other transmitters in a host.
Grantees can increase the utility of their modular transmitters by providing special means, modes, or instructions that simulates or characterizes a connection by enabling a transmitter. This can greatly simplify a host manufacturer’s determination that a module as installed in a host complies with FCC requirements.
Explanation: Data transfer module demo board can control the EUT work in RF test mode at specified test channel

2.10 Additional testing, Part 15 Subpart B disclaimer
The grantee should include a statement that the modular transmitter is only FCC authorized for the specific rule parts (i.e., FCC transmitter rules) listed on the grant, and that the host product manufacturer is responsible for compliance to any other FCC rules that apply to the host not covered by the modular transmitter grant of certification. If the grantee markets their product as being Part 15 Subpart B compliant (when it also contains unintentional-radiator digital circuity), then the grantee shall provide a notice stating that the final host product still requires Part 15 Subpart B compliance testing with the modular transmitter installed.

Explanation: The module without unintentional-radiator digital circuity, so the module does not require an evaluation by FCC Part 15 Subpart B. The host should be evaluated by the FCC Subpart B.