# KeeUnloq

Open-source tools for exploring Microchip KeeLoq keyless remote systems (the old ones).

|  |  |
| --- | --- |
|  |  |
| *KUL FULL v1 PCB rev 0* | *KUL LITE v1 PCB rev 0* |

## Features

Device comes in two PCB variants: **KUL FULL** and **KUL LITE**.

Note: version **KUL FULL** contains RF receiver and transmitter while **KUL LITE** does not.

* Receiver module with optocouplers (only with PCB „KUL FULL“)
* Upgrading third-party insecure systems that use fixed-code HCS101 with rolling-code HCS versions (HCS200, HCS300, ...)
* Emulating HCS transmitter (only with PCB „KUL FULL“ or with external RF ASK transmitter)
* KeeLoq HCS series programmer tool (requires PC software)
* PC software for programming HCS series chips, upload & download of collected transmitter data and more
* Firmware upgrade via USB
* Powered externally or by USB
  + 5V DC or 9-24 AC/DC (only with PCB „KUL FULL“)
  + 5V DC (PCB „KUL LITE“)
* Open source

## Modes of operation

Devices with **FW v1** support 4 modes of operation:

1. **Mode 1: Receiver module** with memory of up to 1000 remote transmitters
2. **Mode 2: MITM upgrader** for upgrading third-party systems (insecure garage door openers)
3. **Mode 3:** **Grab, collect and re-transmit** with memory of up to 1000 transmissions
4. **Mode 4:** **Transmitter emulator**

Mode of operation is changed by holding down **button** **S0** and resetting the device (via **reset button** or by supplying power). **LED A** will remain ON and current operating mode will be displayed by blinking the **LED B**. Selecting different mode of operation can now be accomplished by again pressing the **button S0** within 5 seconds. If **button S0** is not pressed within 5 seconds, the device will exit mode-change procedure and start with operation in currently selected mode.

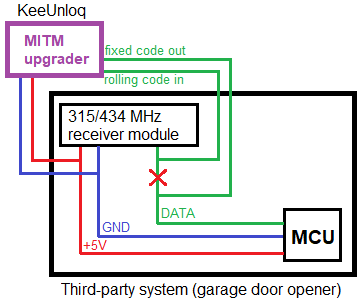
### **Mode 1**: Receiver module

Receiver module is used as a stand-alone receiver which is capable of learning up to 2000 KeeLoq remote transmitters. It has 4 digital optocoupler **outputs S0-S3** (only PCB „KUL FULL“) which are usually connected to garage door motor controllers. It is also capable of learning insecure HCS101 chips and rolling-code HCS series that require programming before usage.

Please refer to **Table 1** in *Programming remote transmitters and encoder ICs.*

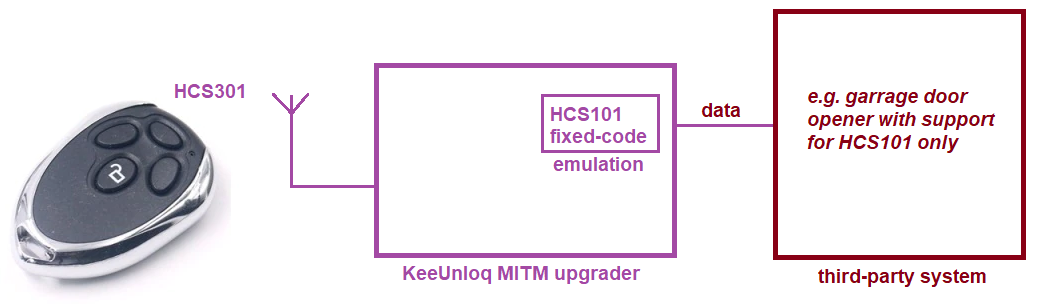
### **Mode 2**: MITM upgrader

Man-in-the-middle upgrader is a mode of operation where the goal is to upgrade a fixed-code insecure third-party system with a rolling-code transmitter. Rolling-code HCS series are pin-compatible with HCS101 fixed code encoder so modifying existing insecure transmitters should be an easy task by simply replacing ICs and programming new HCS with KeeUnloq device.



In order for this mode of operation to be successful, a third-party system must be modified as shown in the schematic diagram above.

The data line of the third-party system must be intercepted so that any data received by the RF receiver is passed into the KeeUnloq device, and special data line from the KeeUnloq device passed into the third-party system.



This means that KeeUnloq device must learn:

1. Fixed-code HCS101 remote control (**only one** HCS101) that third-party system **already recognizes** and works with
2. Rolling-code transmitter that will be learned by the KeeUnloq device (as many as required)

After installing the KeeUnloq MITM device into the third-party system, regular programming must take place with one exception: The remote transmitter that has been last learned by the KeeUnloq device will be passed/proxied into the third-party system upon reception of a valid learned rolling-code remote transmitter.

**Important note:** fixed-code remote transmitters **cannot** be used in this mode during normal operation of the device because learning the last fixed-code remote transmitter is used to pass into the third-party system, so beware.

Please refer to **Table 1** in *Programming remote transmitters and encoder ICs.*

### **Mode 3**: Grab, collect & re-transmit

This mode is intended for collecting KeeLoq data from the air in order to evaluate the security of third-party systems for vulnerabilities such as: **code re-transmission**.

When device is in this state, it will listen, grab and collect information and save to internal EEPROM memory. All data can be downloaded to PC software via USB connection.

### **Mode 4**: Transmitter emulator

In this mode it is possible to emulate any of the HCS series encoders. It is required to previously program the “transmitter identity” via USB connection and PC software:

1. Type of encoder (HCS\*)
2. Serial number
3. Encryption key (if applicable)
4. Counter value

It is also possible to use previously collected remote transmitters in **Mode 3**.

Once device is in this mode, all four buttons are used to transmit the data (S0 – S3).

In order to change/select the remote transmitter “identity” from memory (make sure device is previously set to **Mode 4**) and hold **button S3** and reset the device (via **reset button** or by supplying power). All LEDs will light up for 1 second and then turn off. Currently selected transmitter (ordinal number) will be displayed on LEDs by blinking:

* **LED A** – hundreds
* **LED B** – tens
* **LED C** - ones

Example: **LED A** blinks **2 times**, **LED B** does **not** **blink**, **LED C** blinks **7 times**; the resulting number is **207**. This means that 207th remote transmitter identity is selected from memory.

Selecting the transmitter from memory is done with buttons S0, S1 and S2:

* **Button S2** – hundreds
* **Button S1** – tens
* **Button S0** - ones

Device will remain in this state for 5 seconds and return to operation. Pressing **buttons S0 – S3** will start the transmission.

Notes:

If collected transmitter is of a rolling code type and encryption key is unknown (set to 0), pressing any button will simply re-transmit the last stored transmission. If collected transmitter is a fixed code type (HCS101) it will use the last known synchronization counter and increment on every transmission.

If remote transmitter is collected/seen **only once** in **Mode 3** the encoder type is unknown and it will be simply re-transmitted on any button press.

## Programming remote transmitters and encoder ICs

Programming is done by accessing **buttons S0-S3** or via PC software when device is connected via USB.

|  |  |
| --- | --- |
| **Program & learn** | To learn rolling-code encoder into the device memory, place programming clips on the HCS IC or place it on the PCB programming slot and hold it down firmly. Hold **button S0** to enter programming mode and **LED B** will turn ON. Within **20 seconds** press one of the buttons:   * S0 – program&enroll **66 bit** encoder (HCS200) * S1 – program&enroll **66 bit** encoder (HCS201) * S2 – program&enroll **66 bit** encoder (HCS300, HCS301, HCS320) * S3 – program&enroll **67 bit** encoder (HCS360, HCS361)   Each HCS encoder IC will be programmed with a **random serial number,** random **encryption key,** random **discrimination value** (except HCS360 & HCS361 which mirror the serial number value)andrandom **counter value** (but keptin the range of 0 – 255). This means that encoder ICs programmed using KeeUnloq device cannot be later enrolled using “**Learn from RF**” function in case KeeUnloq memory is later deleted - encoder ICs will need to be re-programmed. If it is required that all encoder ICs share the same encryption key, then use the PC tool to program them individually to the **master encryption key** of your choice. This master key resides in KeeUnloq memory.  The device will remain in this state for 20 seconds after the last transmitter has been programmed and will exit automatically. |
| **Learn from RF** | To learn a remote transmitter into the device memory from RF, hold **button S1** to enter programming mode and **LED B** will turn ON. Within 10 seconds press any button on a remote control and **LED B** will start blinking. Press again **the same button** on remote control within 5 seconds.  If learning process was successful **LED C** will blink:   * **2 times** if remote has been recognized using pre-configured master key as a rolling-code encoder * **5 times** if remote has been recognized as a fixed-code encoder or if using the pre-configured master key did not decrypt the data successfully so the encoder received will be treated as fixed-code type   If programming was not successful **LED A** will blink:   * **2 times** if remote is already programmed into the memory (you can try first deleting it and then learning it again) * **3 times** if device is in **Mode 2** and there is already HCS101 emulating encoder enrolled * **5 times** if remote failed to program because second reception did not match the first reception (you can simply try again) * **10 times** if this remote is not supported by current firmware   The device will remain in this state for 10 seconds after the last transmitter has been received and will exit automatically.  This process will enroll a fixed-code remote transmitter HCS101 any time, or a rolling-code transmitter with the **encryption key** that has been earlier loaded into the KeeUnloq device via USB and PC configuration tool. This also means that every remote transmitter must have this exact encryption key already programmed into them. It will also enroll a rolling-code transmitter even if the encryption key does not match, **however it will be treated as a fixed-code transmitter** where only serial number is used just like with HCS101.  **~~Important note:~~** ~~When device is in~~ **~~Mode 2 (MITM Upgrader)~~** ~~last learned fixed-code HCS101 will be used to pass/proxy into the third-party system.~~ |
| **Remove transmitter from memory (one-by-one)** | In order to delete a specific remote transmitter from the memory, press and hold **button S2**. **LED B** will start blinking. Press any button on a remote transmitter that needs to be deleted from the memory. **LED A** will turn ON to indicate the successful reception of the code. When remote transmitter has been deleted from the memory, **LED C** will blink 2 times to confirm deletion, or **LED A** will blink 2 times to confirm that remote has not been found in memory.  The device will remain in this state for 10 seconds after the last transmitter has been received and will exit automatically. |
| **Clear all memory** | In order to clear all memory from the device, press and hold **button S3** until **LED A** starts blinking. Now within 5 seconds press and hold **button S2** until **LEDs B&C** turn on. Release all buttons and wait until all LEDs finally turn off – the memory is clear.  When device is in:   * **MODE 1**: this will delete only learned/enrolled encoders that can operate the device * **MODE 2**: this will delete only learned/enrolled MITM emulating encoder * **MODE 3**: this will delete entire grabbed/log table * **MODE 4**: this will delete nothing |

*Table 1: programming remote transmitters and encoder ICs*

## Appendix

Example of installation of **MITM Upgrader** into a garage door opener that supports HCS101. Before continuing make sure that garage door opener is programmed and controllable by the working remote transmitter that has HCS101 encoder inside.

Step 1: access the garage door openers RF receiver

Step 2: cut the **data** line and pass it through the KeeUnloq device and power up the garage door opener. Now change KeeUnloq mode into the **MITM Upgrader** (refer to the instructions in this document on how to change the operating mode)

Step 3: perform the **Program & learn** procedure of the existing HCS101 remote controller into the KeeUnloq

Step 4: order new remote transmitters with rolling-code HCS encoder IC (such as: HCS200, HCS300, …) or remove existing HCS101 encoder IC from your current remote controller

Step 5: perform the **Program & learn** procedure of the new HCS encoder IC by using the programming clips or by using the pads on KeeUnloq device (in case you haven’t already soldered new HCS encoder IC onto the remote transmitter)

Step 6: test for proper operation of garage door opener with KeeUnloq device and your new rolling-code transmitter