

iDTRONIC GmbH

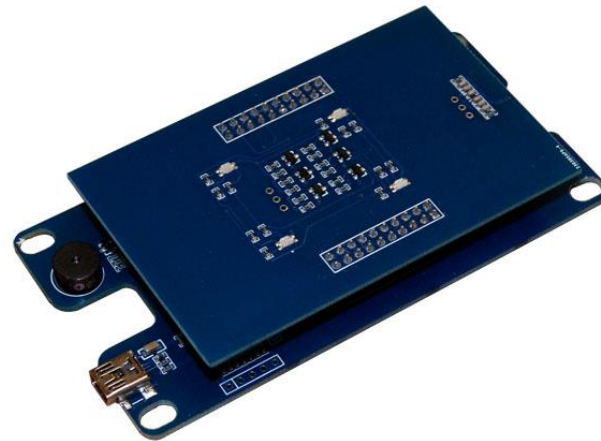


Kemu-Settings

Kemu Settings



Kemu Setting is free of charge and included in the supporting file package for the OEM-HF-R840-SET-V2 RFID reader. It is the graphical configuration tool that with just few clicks customizes the functions of the OEM-HF-R840-SET-V2 device. The software itself will show only the parameters that are meant to be managed. This guide refers to OEM-HF-R840-USB-V2.



Connection



Connectivity

In order to connect the device to the KEMU-Software, first of all it is necessary to choose the proper communication parameters. For the OEM-HF-R840-SET-V2 please select the Com Port.

Connect : Connects your device to the Kemu Software. If the Com Port was chosen right you are going to hear a beep sound.

Settings

Tag Data : Select the data type you are using.

Output Format : Choose the Format of output you desire Hex or ASCII.

Set Reader to Keyboard Mode: Starts the keyboard emulation.

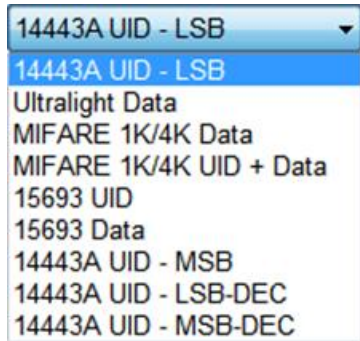
Set Reader : Sets the reader to your desired Settings, if it works you are going to hear a beep.

Protocol Screen

You can monitor the communication between this configuration software and the RFID device.

The screenshot shows the "KEMU Setting" window with three tabs: "Connectivity", "Settings", and "Protocol Screen".
- The "Connectivity" tab is active, showing "Connection:" with "COM" selected and "USB" unchecked. "Com Port:" is set to "COM3", "Baudrate:" is "9600", and "Address:" is "0". A "Connect" button is on the right.
- The "Settings" tab is below, showing "Set Reader to Keyboard Mode" as a toggle switch. "Tag Data" is set to "14443A UID - LSB", "Data Position" is "0", and "Data Length" is "16". "Memory Position" is "0". "Memory Key (if applicable)" is "KeyA" and "Key" is "FF FF FF FF FF FF". "Output Format" is "HEX". A "Set Reader" button is on the right.
- The "Protocol Screen" tab is at the bottom, displaying a log of communication: ">> AA 00 01 83 82 BB", "<< AA 00 0A 00 00 FF FF FF FF FF FF FF FF 0A BB", ">> AA 00 01 86 87 BB", and "<< AA 00 11 00 49 44 54 35 32 37 45 2D 56 35 2E 30 2D 53 45 54 02 BB".

Keyboard Emulation – Function Overview.



| Tag data type | Function |
|-------------------------|-------------------------------------------------------------------------------------------------------------------|
| 14443 A UID – LSB | Outputs the UID compatible with our other readers as hexadecimal number. |
| 14443 A UID – LSB-DEC | Outputs the UID compatible with our other readers as decimal number. |
| 14443 A UID – MSB | Outputs the UID in reverse byte order as hexadecimal number. |
| 14443 A UID – MSB-DEC | Outputs the UID in reverse byte order as decimal number. |
| Mifare 1K/4K Data | Outputs selectable Bytes from a selectable memory block. |
| Mifare 1K/4K UID + Data | Outputs selectable Bytes from a selectable memory block. |
| Ultralight Data | Outputs selectable memory page (4 Bytes). |
| 15693 UID | Outputs the UID compatible with our other readers as hexadecimal number. This is 8 Bytes = 16 characters in size. |
| 15693 Data | Outputs selectable Bytes from a selectable memory block. |

Configuration




1. Connect your Device for configuration

Plug in the device first before starting the configuration software "KEMU Setting".

If the device is connected to a PC for the first time, it can take some time for automatic installation of the Silicon Labs 210x Series VCP driver. If this is the case, pls. wait until this is fully done. Now start the configuration software "KEMU Setting".

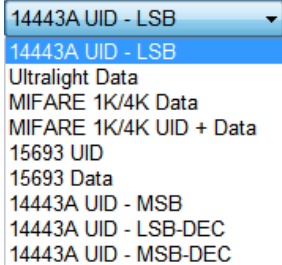
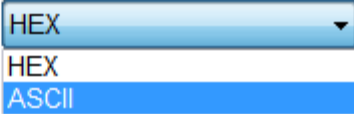
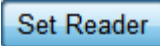
2. Connectivity

Select the correct com port, which has been set by the driver.
Baudrate 9600 and device address 0 are factory default values.
Then connect with 

Connectivity

Connection: ☒ COM ☐ USB
Com Port: Baudrate: Address: 

3. Select Your Data Tag Type and Output Format then press Set Reader

 Output Format  


14443A UID - LSB
14443A UID - LSB
Ultralight Data
MIFARE 1K/4K Data
MIFARE 1K/4K UID + Data
15693 UID
15693 Data
14443A UID - MSB
14443A UID - LSB-DEC
14443A UID - MSB-DEC

HEX
HEX
ASCII

4. Settings with most Tag Types

- Memory Position: Select the memory block (16 Bytes) from which data is read.
- Data Position: Cut a part from the data read from the selected memory block. Data - Position gives the start byte.
- Data Length: Cut a part from the data read from the selected memory block. Data Length gives the number of bytes.

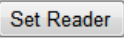
Settings

Set Reader to Keyboard Mode 

Tag Data Data Position Data Length

Memory Position

Memory Key (if applicable) Key

Output Format 

Structure of the most common Transponder memories - Mifare



- The memory of Mifare transponders is divided into different sectors and every sector is divided into 4 (or more if its a Mifare 4k) different blocks.
It is important to know that you can't address a sector you have always to address the block where the information is saved.
- Every sector has a password, it is always in the last block. The password in the last block is divided into two keys. Key A the first six bytes and Key B the last six bytes, Key B can be configured to have more rights than Key A. For example with Key A you can read the balance of your card but with Key B you can change the balance.
- The purple 3 bytes determine what you can do with each key for each block.

MiFare 4K, 3360 usable bytes

MiFare 2K, 1440 usable bytes

MiFare 1K, 720 usable bytes

| | | | | | | | | | | | | | | | | | |
|-----------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Sector #0 | Block #0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | Block #1 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | Block #2 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| | Block #3 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| Sector #1 | Block #4 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| | Block #5 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| | Block #6 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| | Block #7 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| Sector #2 | Block #8 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| | Block #9 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| | Block #10 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| | Block #11 | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |

Note : It is absolutely normal that you can't change Block 0

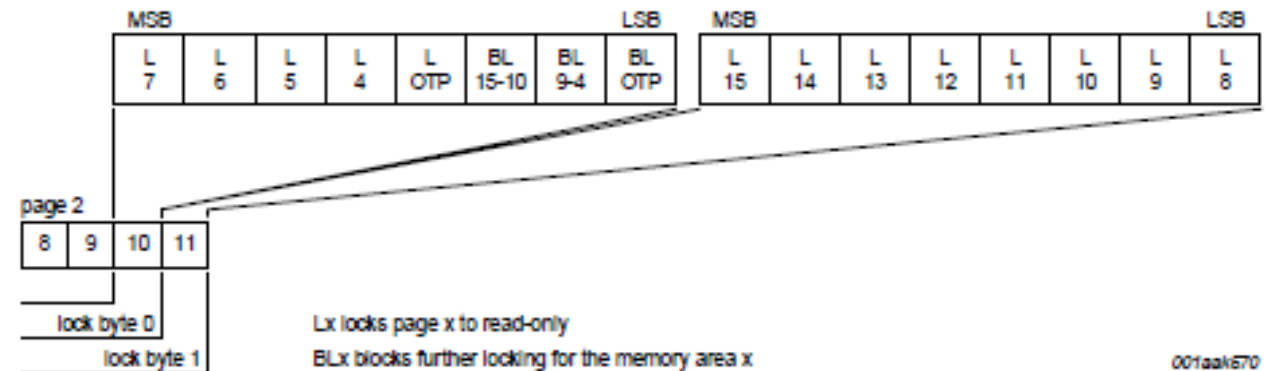
Structure of the most common

Transponder memories – Mifare UL



- The memory of the Transponder Mifare Ultralight is divided into the page address and these consist of four bytes. On the first two pages (0, 1) the serial number is saved.
- The last two bytes of page 2 are lock bytes. With them you can lock every page from 3-15 to read only. The last 3 bits of lock byte 0 are the block locking bits. The first of the three blocks pages 15-10 the second 9-4 and the last the OTP, every other bit blocks a single page. After these bits are set to written state (1) they cannot be changed back
- Page 3 consists of OTP bytes (One time programming), these bytes are set by default to erased (0). If a bit in these bytes is changed to written state (1) it can not be changed back. But if a bit is free you can still write into this bit.

| Page address | | Byte number | | | |
|--------------|------------|---------------|----------|------------|------------|
| Decimal | Hex | 0 | 1 | 2 | 3 |
| 0 | 00h | serial number | | | |
| 1 | 01h | serial number | | | |
| 2 | 02h | serial number | internal | lock bytes | lock bytes |
| 3 | 03h | OTP | OTP | OTP | OTP |
| 4 to 15 | 04h to 0Fh | user memory | | | |



001ask570

Structure of the most common Transponder memories – I-Code SLI



The 1024 bit SLI memory is divided into 32 Blocks. A block is the smallest access unit, each of them consists of 4 bytes. The memory is divided into 2 parts, the configuration area and the user memory.

In the first two blocks (-4, -3) the unique identifier ID is saved, it is programmed in the production process and can't be changed afterwards. The TAG type is a part of the UID (bit41 to 48, after the manufacturer code which is "04" hex for Philips Semiconductors). The TAG type of the SL2 ICS20 is "01" hex.

Within the EAS byte the last bit of it can turn the EAS (Electronic Article Surveillance) just by writing 1 for on or 0 for off.

The Write Access Condition bits in block -1 determine the write access conditions for the other blocks. These bits can be set only to 1 with a lock command (and never be changed back to 0)

| | Byte 0 | Byte 1 | Byte 2 | Byte 3 | |
|----------|-----------------|--------|--------|--------|----------------------------------|
| Block -4 | UID0 | UID1 | UID2 | UID3 | Unique Identifier (lower bytes) |
| Block -3 | UID4 | UID5 | UID6 | UID7 | Unique Identifier (higher bytes) |
| Block -2 | Internally used | EAS | AFI | DSFID | EAS, AFI, DSFID |
| Block -1 | 00 | 00 | 00 | 00 | Write Access Conditions |
| Block 0 | X | X | X | X | User Data |
| Block 1 | X | X | X | X | User Data |

| MSB | | | | | | | | | | LSB | |
|-------|----|-------|----|-------|----|-------------------------------|-------|-------|-------|-------|---|
| 64 | 57 | 56 | 49 | 48 | 41 | 40 | | | | | 1 |
| "E0" | | "04" | | "01" | | IC manufacturer serial number | | | | | |
| UID 7 | | UID 6 | | UID 5 | | UID 4 | UID 3 | UID 2 | UID 1 | UID 0 | |

| Block -1 | | | | | | | | | | | | | | | |
|-------------------------------|---|---|---|---|--------|--------|--------|--------|----|----|---|---|---|---|---|
| Byte 0 | | | | | | | | Byte 1 | | | | | | | |
| MSB | | | | | | | | LSB | | | | | | | |
| Condition | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Write access for block number | 3 | 2 | 1 | 0 | -2 (3) | -2 (2) | -2 (1) | -2 (0) | 11 | 10 | 9 | 8 | 7 | 6 | 4 |

!! Note : to write in negative blocks you have to use the special commands as displayed in ISO- 15693

**Your trusted partner for
RFID hardware solutions**

iDTRONIC GmbH

www.idtronic-rfid.com