Z-StickC

ZME_ZSTICKC (EU) USB Stick with Z-Wave transceiver

The Z-Wave.Me Z-StickC connects a PC platform with USB interface to the wireless Z-Wave network. The interface to the Z-Wave network is a standard Z-Wave transceiver module ZM3102 implementing a serial control protocol. The USB stick connects this serial interface to the PC platform. Once the USB stick is connected to the PC the OS of the PC will create a new virtual serial device (/dev/tty.xxxx for Unix/MAC, COM X for Windows) that is used by control software on the PC to communicate with the Z-Wave network. The Z-StickC implements all necessary communication functions of Z-Wave but the functionality depends on the software that uses the Z-StickC.

Installation

The USB Stick will work on every USB 1.1/2.0/3.0 compatible USB interface and is also powered by USB. The USB Serial conversion function is implemented using a serial-2-usb interface chip from the vendor FTDI. Modern operating systems have a device driver for this chip already installed on default. Device drivers for MS Windows and MAC OSX can be found on the support web site mentioned below. Updated device drivers can also be downloaded from the FTDI website mentioned below.

It is possible to upgrade the firmware on the Z-StickC. Several different firmware versions are available, one is built on the SDK 5.03 (SUC/SIS Support) and one is built on SDK 4.54 (Explorer Frame support). You find the upgrade tool and the most recent firmware for SDK 4 and 5 on the support web site mentioned below.

Usage / Serial protocol

The Z-StickC serial protocol is based on the Z-Wave Serial Interface Protocol, recommended by Sigma Designs. It is optimized for a static controller role of the device using this USB device. Therefore the protocol specification supports the following functions.

- Network Management (Inclusion, Exclusion of devices, Controller handling, ...)
- Application Data exchange (Send Data, Application Handler)

The full specification of the Serial API protocol is not publicly available but part of the Sigma Designs SDK. However you can expect that most software solutions for Z-Wave including the Controllers from Z-Wave.Me (Z-Cloud, RaZberry, Z-box,…) implement the standard Sigma Designs API.

This Z-StickC supports the Sigma Designs API as well but offers some enhanced functions that are not part of this spec. Please refer to the Annex below for information about the Z-StickC specific functions.

The Z-StickC enhances the standard Z-Wave transceiver by the following functions:

- **Restore function** Allows to recover all network data in a different hardware
- **Wakeup Helper** If enabled, this function saves battery lifetime of sleeping devices when the Z-Wave controller software is not running on host processor. The USB stick needs to be powerd by a power supply, e.g.

an notmale USB Power plug used to charge mobile phones.

- **Firmware Update** This allows to update the firmware. Please refer to the firmware update section of this manual for more information. **Note!** Only Z-Wave.Me firmwares (.ehex) can be loaded to this Z-StickC. Uploading standard firmwares from other USB sticks may make this device nonfunctional.
- **VendorId Change** Allows to change VendorId value reported by Z-StickC. This can be useful to limit own software developments to certain hardware nd use htis hardware as protection for the software. By default Z-Wave.Me Vendor Id is reported. In order to change the vendor ID a special key is required. Contact to Z-Wave.Me for details on the commercial arrangement needed to obtain this key sequence.

Z-Connector

Z-Connector is a free application software for Z-Wave available for Linux, MAC OSX and Windows. It connects your system to a free online cloud service (Z-Cloud) running Z-Way a full Z-Wave controller application with JSON interface and AJAX User Interface. Z-Connector can be downloaded from the web site z-cloud.z-wave.me. Please register at this site and follow the instructions there.

LED

The Z-StickC uses two LEDs to indicate status information. They are inside the enclosure but shine thru the translucent plastics.

- RED: permanent on Inclusion mode and Exclusion Mode
- GREEN: Blink on transmission success

Firmware Update

There are several reasons to update the firmware. The Z-StickC comes with a firmware that's is based on the most recent Z-Wave SDK 4.5. This firmware however is lacking some of the support functions that help to manage older devices based on SDK 5.x. A network with majority of devices with SDK 5.x may be better controlled using a Z-Wave transceiver based on SDK 5.x as well.

Everybody makes mistakes. Even more, the Z-StickC is based on Z-Wave protocol libraries, which can also contain errors and bugs. Firmware upgrades delivers bug fixes if needed. It is always recommended to use the latest version of firmware.

New features developed for the Z-StickC become available with a firmware upgrade avoiding the need to purchase a new hardware.

Upgrade process

Note that the upgrade process is dangerous! Do not unplug or power off the stick during the firmware upgrade process. To upgrade the firmware follow instructions below.

1. Download the firmware upgrade utility from the device support site below and unpack it. Make sure you have the python interpreter installed. For more information about python and the python interpreter please refer to the website mentioned below.
2. Download a new firmware from the device support site below and unpack it. Then connect your Z-StickC to your PC
3. Make sure you have PL2303 driver installed (if you have used your Z-Stick previously, it is installed). Locate the correct port name of your stick (use `dmesg` on Linux and OSX and Device Manager in Windows; if you have used your Z-Stick already, you should know it)
4. Make sure there is no other software using the stick running (shutdown Z-Way, Z-Connector, HomeSeer or any other software you use)
5. Run ZStickUpdater utility from terminal: *python ZStickUpdater.pyc -f <path to ehex file> -d <port name>* On Linux it would be something like: *python ZStickUpdater.pyc -f ~/Downloads/Z-Stick4.ehex -d /dev/ttyUSB0* On Windows: *c:\Program Files\Python\bin\python ZStickUpdater.pyc -f c:\Uses\Joe\Downloads\Z-Stick4.ehex -d COM8*
6. The process should end with information message: *Progrm.INFO Done.* If you have encountered an error, check log

messages to find a hint how to solve the problem.

7. If you have not changed between SDK 4 and SDK 5 firmware, you are done. If you have changed, use Restore function to fix the content of internal memory of the stick. Consult you software manual for details.

Important Web Resources

- SDK Information: http://wiki.zwaveeurope.com/index.php?title=SDK_Versions_and_Explorer_Frames
- Device Support Site: http://www.zwaveeurope.com/support/ZME/ZSTICK/
- FTDI Web Site for Device drivers: http://www.ftdichip.com/Drivers/VCP.htm
- Python: http://www.python.org/

Annex A: Restore Function

General syntax:

HOST->ZW: REQ | 0x73 | cmd | data | funcID
ZW->HOST: RES | 0x73 | RetVal
RetVal: 0 = failed. 1 = queued
ZW->HOST: REQ | 0x73 | funcID

Commands:

cmd = ZME_RESTORE_HOMEID_FIRST, data: homeID (4 bytes)
cmd = ZME_RESTORE_HOMEID_SECOND, data: homeID (4 bytes) | myNodeID
cmd = ZME_RESTORE_MAXNODE, data: maxNodeId
cmd = nodeID (1-232), data: deviceType (5 bytes)

ZME_RESTORE_HOMEID_FIRST = 251
ZME_RESTORE_HOMEID_SECOND = 252
ZME_RESTORE_MAXNODE = 255

Z-Way usage:

// set NodeId device information
ZMERestore(nodeId, [deviceTypeArray])

// set HomeID of controller when NodeId == 1 and primary
ZMERestore(ZME_RESTORE_HOMEID_FIRST, [homeId])

// set HomeID of controller when NodeId != 1. set nodeId = 0 to use HomeId from ZME_RESTORE_HOMEID_FIRST
ZMERestore(ZME_RESTORE_HOMEID_SECOND, [homeId, nodeId])

// set maximal NodeId registered in network
ZMERestore(ZME_RESTORE_MAXNODE, [maxNodeId])

Annex B: Wakeup Helper

If enabled, this function saves battery life of sleeping devices while Z-Wave softwere is not running on host processor.

SerialAPI:

HOST->ZW: 0xf0 | state
state: 0 disabled, 1 enabled

Z-Way usage:

ZMEWakeupHelper(state)

Annex C: Firmware upgrade

Turn firmware upgrade mode. Note that Z-Stick become non-functional until new firmware is uploaded. Special software from Z-Wave.Me is required to upload the new firmware.

SerialAPI:

HOST->ZW: 0xf1

Z-Way usage:

ZMEFreqReflash()

Annex D: VendorId change

Allows to change VendorId value reported by Z-Stick 4 and Z-Stick 5. By default Z-Wave.Me VendorId is reported. Note that a special 10 Byte key is required to change the vendor ID. Contact info@z-wave.me for details.

SerialAPI:

HOST->ZW: REQ | 0x24 | 0 | 24 | [key (10 bytes)] | funcID
ZW->HOST: RES | 0x24 | RetVal

RetVal: 0 = failed. 1 = queued
ZW->HOST: REQ | 0x24 | funcID

Z-Way usage:

MemoryPutBuffer(24, [key (10 bytes)])

Annex E: Frequency Change

Switches the transceiver frequency between Russian and European.

SerialAPI:

HOST->ZW: REQ | 0xf2 | freq
freq: 0x00 = EU, 0x01 = RU

Technical Data

Z-Wave Chip	Sigma Designs ZM3102
Indicator	one green LED and one red LED
Frequency	EU: 868.4 MHz (EN 300 220) RU: 869.0 MHz (GKRCh/EN 300 200) US: 908.4 MHz (FCC CFR47 P 15.249)
Wireless Range	Up to 100 m outside, on average up to
Explorer Frame	Yes
SDK	4.54 pl1
Device Type	Depends on Application firmware
Routing	yes
Serial API Version	4.2
Dimensions	65 x 24 x 8 mm
Weight	15 gr.
Origin	Made in Germany

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:

(1) Reorient or relocate the receiving antenna.

(2) Increase the separation between the equipment and receiver.

(3) Connect the equipment into an outlet on a circuit different from that to which the receiver is connected. (4) Consult the dealer or an experienced radio/TV technician for help. Any changes or modification not expressly approved by the party responsible for compliance could void the user's authority to operate the device.

Where shielded interface cables have been provided with the product or specified additional components or accessories elsewhere to be used with the installation of the product, they must be used in order to ensure compliance with FCC regulations.

CE for Class B ITE (Following European standard EN55022/1998; EN61000- 3-2/1995; EN61000-3-3/1995, EN55024/1998, EN60950-1/2001)