



Stinger USB Developer's Guide

Please read carefully!

May 12th , 2014

Document revision 1.07

*Valid from Stinger USB
Firmware version 1.25- Stinger Mode, Phoenix
Mode or UniProg Mode.*

Stinger USB – Setting Manager 1.5.01 or later

Index

| | |
|---|----|
| 1. Change Log | 4 |
| 2. Premise..... | 6 |
| Firmware Features Cross Reference Table | 7 |
| 3. Phoenix Mode (<i>working with Phoenix Mode Firmware</i>) | 8 |
| 3.2 Commands Description (<i>Phoenix Mode</i>) | 10 |
| Command 00 – Get Stinger USB information..... | 10 |
| Command 77 – Set Stinger Clock in <i>Phoenix Mode</i> | 11 |
| 4. Stinger Mode (<i>working with Stinger Mode Firmware</i>) | 12 |
| Clock Speed and Baudrate. | 14 |
| Parity Setting. | 16 |
| Smartcard Presence Detection..... | 16 |
| Smartcard Logger - Debug Mode..... | 17 |
| Extra hardware needed | 19 |
| 4.2 Commands Description (<i>Stinger Mode</i>) | 20 |
| Command 00 – Get Stinger USB information..... | 21 |
| Command 01 – Set USB speed between Stinger and PC. <i>“On the flight” Command. Needs to be saved to be effective after power off.</i> | 23 |
| Command 02 – Set Smartcard Clock. <i>“On the flight” Command. Needs to be saved to be effective after power off.</i> | 24 |
| Command 03 – Set Smartcard COM parameters. <i>“On the flight” Command. Needs to be saved to be effective after power off.</i> | 25 |
| Command 05 – Store user setting to flash memory. <i>“On the flight” Command. Needs to be saved to be effective after power off.</i> | 26 |
| Command 06 – Reset to factory default. <i>“Non On the flight” Command. Effective after Power OFF and Power ON.</i> | 27 |
| Command FD – ATR Baudrate and Debug Mode. <i>“Non On the flight” Command. Effective after Power OFF and Power ON.</i> | 28 |
| Command FD – Activate Debug Mode | 29 |
| 5. UniProg Mode (<i>working with UniProg Firmware</i>)..... | 30 |
| 5.2 Commands Description (<i>UniProg Mode</i>)..... | 32 |

| | |
|--|----|
| Command 00 – Get Stinger USB information..... | 32 |
| Command 44 – Set Stinger Clock and Baudrates in <i>UniProg Mode</i> | 33 |
| Command 42 – Read Stinger Clock and Baudrates in <i>UniProg Mode</i> | 34 |
| Command 45 – Erase Stinger Clock and Baudrates in <i>UniProg Mode</i> | 35 |
| 6. C Source code examples | 36 |
| 6.2 Copy and Paste Code..... | 38 |
| 6. How to compile and run the <i>conf.c</i> | 39 |
| 7. Appendix A - " <i>Dummy Card</i> " | 41 |

1. Change Log

May 13th 2013 – from Stinger Firmware 1.25

- New UniProg firmware version 1.25 available.
- Serial Number added on USB Descriptor. To set the 8 Bytes serial number use **Stinger USB – Setting Manager** or **FTDI - FT_PROG 2.8.2.0 - EEPROM Programming Utility** (<http://www.ftdichip.com/Support/Utilities.htm>). **Stinger USB – Setting Manager** will set the last 8 digits of the serial number of the Stinger USB connected. Example: 0000245D9D9100070016300522254E45 will be serialized as: 22254E45. This number can be changed and programmed internally to appear as iSerial on Linux or MAC OS X.



Please use the latest version of **Stinger USB – Setting Manager** downloadable from <http://www.stingerusb.com>. If you are using an old version and you load the latest UniProg Firmware 1.25 you cannot load previous versions unless you use **Stinger USB – Setting Manager**.

December 02nd 2013 – from Stinger Firmware 1.25

Introduced two different firmwares and two new functions.

Available Firmware:

- **Stinger Mode** firmware which is now supporting *Smartcard Logger* function like a Season Interface.
- **Phoenix Mode** firmware transforms a Stinger USB in a normal Phoenix Programmer with programmable Clock Speed.

Please refer to the following pages to fully discover both firmware functions.

The Developer's Guide has significantly changed from release 1.05 so please read carefully!

September 24th 2013 – from Stinger Firmware 1.24

- ATR Speed implemented – Command FD XX

We added the ATR support at 9600 baud instead of 115200baud to be more compatible with third party software through command FD.

FD 00 – ATR at 115200 baud

FD 80 – ATR at 9600 baud.

This mode is NOT set by default so it is needed to set it by default and then save the configuration.

After the ATR the smartcard will communicate at the speed set with Command 01.

2. Premise

This document is dedicated to third parties software's developers who want to control the Stinger USB configuration using their own software routines.

Stinger USB is an extremely flexible double Smartcard reader and writer. It is possible to connect the Stinger USB to a Windows, MAC OSX or Linux Operating System and no special drivers are needed to detect the device.

Stinger USB is based on a FTDI Chipset to support the USB to Serial 232 protocol. FTDI drivers are normally natively embedded in any Operating System or can be downloaded from the FTDI web site at <http://www.ftdichip.com/>

Each Smartcard reader is independently associated to Virtual COM port on your Personal Computer. This means that once the Stinger USB is connected and the drivers are properly loaded, two new COM Ports will be displayed on Windows based machines and two new devices are available on MAC OSX and Linux based machines.

From firmware version 1.25 we split the single firmware in three different firmware-versions **Stinger mode**, **Phoenix Mode** and **UniProg Mode**.

In **Stinger Mode** you can use your Stinger USB as a smart programmer with dedicated command to perform some CPU controlled functions. The new *Smartcard Logger – Debug Mode* (Season Logger) function is implemented in **Stinger Mode only**. See section 4.

In **Phoenix Mode** the Stinger USB works as a normal Phoenix Programmer where is possible to set the Clock Frequency only. This is a “*Pass-Through*” function which allows the programmer to be a simple USB to Serial Port converter with the addition of a controlled Clock to the Smartcard. See section 3.

In **UniProg Mode** the Stinger USB works as an Emulation of UniProg Protocol which is similar but not equal to Phoenix Mode. In UniProg Mode some settings must be given to the Stinger USB such as Clock Speed for each connected Smartcard and two cards-talk speeds (Baudrate). In UniProg Mode user must set the Baudrate between Stinger USB and PC (after the ATR) and the Baudrate between Stinger USB and Smartcard (after the ATR). ATR is always returned at 9600 Baud. See section 5

To run in *Stinger Mode*, *Phoenix Mode* or *UniProg Mode* you need to change the firmware by loading it time by time accordingly to your needs. To load the firmware please use the **Stinger USB – Setting Manager** available for MACOX or Windows or the Command Line version available for Linux or MAC OSX.

Please make sure you loaded the correct firmware version to access the correct function.

Here next we present a cross reference table for Stinger USB Functions:

Firmware Features Cross Reference Table

| Features | Firmware Version | | |
|---------------------------------------|---|-------------------------------------|--|
| | <u>Stinger USB Mode</u> | <u>Phoenix Mode</u> | <u>UniProg Mode</u> |
| Customizable ATR Baudrate | YES | Fixed – 9600 Baud | Fixed – 9600 Baud |
| Customizable Smartcard Clock Speed(*) | YES | YES | YES |
| Customizable Baudrate | YES PC <-> Stinger: • 9600 • 115200 Stinger <-> Smartcard: • 9600 • 38400 • 115200 | Pass-Through | YES PC <-> Stinger: • 9600 • 38400 • 115200 Stinger <-> Smartcard: • 9600 • 38400 • 115200 |
| Customizable Smartcard Parity | YES | Pass-Through | Automatic Detection |
| Smartcard Reset PIN | RTS | RTS | RTS |
| Command Mode PIN (settings) | DTR From HIGH To LOW 115200 8N1 | DTR From HIGH To LOW 9600 8N1 | RTS + DTR RTS: From HIGH To LOW after 50ms DTR: From LOW To HIGH 9600 8N1 |
| Smartcard Detection PIN | CTS | CTS | Auto Detect + CTS |
| Smartcard Debug Mode (Season Logger) | YES | NO | NO |

(*)Clock Speeds: 3.43 Mhz, 3.69 Mhz, 4.00 Mhz, 4.80 Mhz, 6.00 Mhz, 8.00 Mhz, Mhz,12.00.

3. Phoenix Mode

(working with Phoenix Mode Firmware)

The *Phoenix Mode* requires to load on the Stinger USB the *Phoenix Mode* dedicated firmware that can be downloaded from www.stingerusb.com official web site.

Clock for Stinger USB in *Phoenix Mode* can be set at the following speed:

| Value for Configuration: (See Command 77) | HEX: | DEC: | MHZ: |
|--|--------|----------|-------|
| 01 | 368120 | 3572000 | 3.572 |
| 02 | 384E10 | 3690000 | 3.69 |
| 03 | 3D0900 | 4000000 | 4.00 |
| 04 | 493E00 | 4800000 | 4.80 |
| 05 | 5B8D80 | 6000000 | 6.00 |
| 06 | 7A1200 | 8000000 | 8.00 |
| 07 | B71B00 | 12000000 | 12.00 |

While operating in *Phoenix Mode* Baudrate, Parity and Stop Bits do not need to be set as those settings are handled directly from the software. The Stinger USB gets and transmits raw bytes from and to the Smartcard transparently up to 115200 Baud.

The Smartcard Clock once it is set does not need to be saved as the Stinger USB is automatically setting and saving the Smartcard Clock speed.

After loading the *Phoenix Mode* firmware both LEDs will light up in Yellow Color. To set the clock speed of the Upper Slot and Lower Slot it is necessary to hold **DTR signal HIGH** for at least 500ms and then right after **DTR signal LOW** and within maximum 100ms and minimum 10ms since the **DTR signal is LOW** send the Configuration command explained in the next section.

To successfully communicate to the Stinger USB while is *Phoenix Mode* in command mode (settings) the Virtual COM port must be set at:

9600 Baud - 8N1

Depending on the programming language and Operating System the programming routines may vary but it is strictly necessary to communicate to the Stinger USB at the above Baudrate and follow the port configuration at 8Bit, No Parity and 1 Stop Bit.

It is alternatively possible to use **Stinger USB – Setting Manager** or the other open source code available on www.stingerusb.com to set the clock speed and load the appropriate firmware.



NOTE!

The Stinger USB in *Stinger Mode* communicates at 115200Baud at 8N1 but in *Phoenix Mode* the setting must be 9600 8N1 while in command mode (settings).

It is absolutely mandatory to follow the DTR HIGH - DTR LOW sequence exactly as explained above and to repeat the DTR HIGH - DTR LOW sequence every each command that we need to send to the Stinger while in Phoenix Mode.

This procedure is slightly different from the INIT procedure in *Stinger Mode!*

Please remember to observe the timing of DTR HIGH – DTR LOW sequence and send the desired command within maximum 100ms and minimum 10 ms after the DTR is set to LOW!

3.2 Commands Description *(Phoenix Mode)*

Command 00 – Get Stinger USB information.

Send from PC: 00

Answer from Stinger USB: 32 bytes

Answer example:

0001 0000313A 0000245D9D9100070016300522254E455B8D80 00 00 00 0000 0122

| | | | | | | | | | | | |
|--------------|--------------|-----------|----------|----------|----------|--------------|-----------|-----------|--------------|-----------|-----------|
| 00-01 | 02-05 | 06 | - | - | - | 22-24 | 25 | 26 | 27-29 | 30 | 31 |
| 00 0X | 0000313A | 0000245D | 9D910007 | 00163005 | 22254E45 | 5B8D80 | 00 | 00 | NA | 0000 | 0119 |

| Byte | Description | Values | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|--|--|------|------|------|---------------|---------|-------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|----------|-------|
| 00 | Echo Command (1 Byte) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | Reader Number (1 Byte) | 00 – Upper Smartcard reader 01 – Lower Smartcard reader | | | | | | | | | | | | | | | | | | | | | | | | |
| 02-05 | Chip Family (4 Bytes) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06-21 | Unique Serial Number (20 Bytes) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22-24 | Clock speed (3 byte) <i>CMD – 77 to set</i> | HEX Value must be converted in DEC and it will show up to clock Speed in Hz. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>HEX:</th> <th>DEC:</th> <th>MHZ:</th> </tr> </thead> <tbody> <tr> <td>368120</td> <td>3572000</td> <td>3.572</td> </tr> <tr> <td>384E10</td> <td>3690000</td> <td>3.69</td> </tr> <tr> <td>3D0900</td> <td>4000000</td> <td>4.00</td> </tr> <tr> <td>493E00</td> <td>4800000</td> <td>4.80</td> </tr> <tr> <td>5B8D80</td> <td>6000000</td> <td>6.00</td> </tr> <tr> <td>7A1200</td> <td>8000000</td> <td>8.00</td> </tr> <tr> <td>B71B00</td> <td>12000000</td> <td>12.00</td> </tr> </tbody> </table> | HEX: | DEC: | MHZ: | 368120 | 3572000 | 3.572 | 384E10 | 3690000 | 3.69 | 3D0900 | 4000000 | 4.00 | 493E00 | 4800000 | 4.80 | 5B8D80 | 6000000 | 6.00 | 7A1200 | 8000000 | 8.00 | B71B00 | 12000000 | 12.00 |
| HEX: | DEC: | MHZ: | | | | | | | | | | | | | | | | | | | | | | | | |
| 368120 | 3572000 | 3.572 | | | | | | | | | | | | | | | | | | | | | | | | |
| 384E10 | 3690000 | 3.69 | | | | | | | | | | | | | | | | | | | | | | | | |
| 3D0900 | 4000000 | 4.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 493E00 | 4800000 | 4.80 | | | | | | | | | | | | | | | | | | | | | | | | |
| 5B8D80 | 6000000 | 6.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 7A1200 | 8000000 | 8.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| B71B00 | 12000000 | 12.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | ND | ND | | | | | | | | | | | | | | | | | | | | | | | | |

| | | |
|-------|----------------------------|--------------------------------------|
| 26 | ND | ND |
| 27-29 | <i>Reserved</i> | <i>Reserved</i> |
| 30-31 | Firmware version (2 bytes) | 01.25 (Decimal Value when converted) |

Command 77 – Set Stinger Clock in Phoenix Mode.

Send from PC: 77 xx yy

Clock Speed:

xx --> Virtual COM Port number number (00 or 01)

yy == 01 == 3.572 Mhz

yy == 02 == 3,69 Mhz

yy == 03 == 4.00 Mhz

yy == 04 == 4.80 Mhz

yy == 05 == 6.00 Mhz

yy == 06 == 8.00 Mhz

yy == 07 == 12.00 Mhz

Example:

77 00 01 Sets Upper Smartcard Slot at 3.572 Mhz

4. Stinger Mode

(working with Stinger Mode Firmware)

Stinger Mode allows the user to perform two different functions:

- Use the Stinger USB as a CPU controlled Smartcard reader.
- Use the Stinger USB as a Session Logger which allows the user to log the traffic between a Smartcard and a Host (remote) device where the Smartcard is supposed to be working inside (example Mobile Phones, Satellite Receivers etc). No specific software will be provided to log the traffic from and to the Smartcard. This mode is called *Debug Mode*.

To run the above operating modes it is necessary to load the Stinger USB with *Stinger Mode* firmware.

The Stinger USB operating settings can be modified using the official GUI Software called **Stinger USB - Setting Manager** available on the official web site www.stingerusb.com. This GUI software is working under Windows (from Windows XP) and on MAC OSX (from version 10.7.x). There is also a command line version available for MAC OSX and Linux and related Source Codes.

Normal Smartcard programmer settings:

- **Reset Mode** (*Smartmouse or Phoenix mode, [read note below](#)*),
- **Clock Speed** (3.43 Mhz, 4.00 Mhz, 4.80 Mhz, 6.00 Mhz, 8.00 Mhz, 12.00 Mhz),
- **Baudrate Speed** after the ATR (*from standard 9600bps to 115200bps*),
- **Parity Mode** (*Odd, Even None*) while **Stop Bits** value is fixed to 1.

Such settings can be also be changed using a non-GUI interface made by third parties. To change settings programmatically the developer needs to enter *Configuration Mode*.

To enter *Configuration Mode* hold the **RS-232 DTR signal LOW, RTS signal HIGH and wait 1500 ms** after powering up the Stinger USB and then sending configuration settings to the device.

The list of settings commands is described below.

The Stinger USB configuration must be saved with Command 05 (see below) otherwise the programmer once turned OFF will restart with previous settings.

It is also possible to restore factory settings using with Command 06 (*see below*).

As described above to enter the *Configuration Mode* open your Virtual COM port and hold the **DTR signal LOW, RTS signal HIGH and wait 1500 ms** so the Stinger USB will detect the *Configuration Mode* (differently it will work as a Smartcard reader using the predefined settings).

DTR signal can be held LOW on any of the two Virtual COM ports to enter the *Configuration Mode*.

To successfully talk to the Stinger USB while is *Configuration Mode* the Virtual COM port must be set at:

115200 Baud - 8N1

Depending on the programming language and Operating System the programming routines may vary but it is strictly necessary to talk at the above Baudrate.

Settings are effective depending on the given command. Some Commands are considered "*On the flight*" which means immediately effective and some others take effect after powering OFF and powering ON the Stinger USB ("*Non On the flight*").

To get more specific information please read the *Stinger Mode Command* section.

Clock Speed and Baudrate.

Another critical point on Smartcard is the calculation of the correct Baudrate and Clock speed. Basically the problem raises because on some Operating Systems it is not possible to set an arbitrary Baudrate on the Serial COM port when opening and setting the device preferences.

It is common practice to use a 3,579 Mhz Clock Speed to get an exact 9600 Baud (or multiples, i.e. 115200) on the Serial Port. This is a hardware adaptation to match a software limitation due to the Operating Systems. Some Smartcards need to be fed with a 6,00Mhz Clock to match the correct transmission at 9600 Baud.

This means that a Smartcard that is fed with 6,00Mhz is slower than a Smartcard fed at 3,579Mhz. The final result - Elementary Time Unit (ETU) - must be at 9600 Baud to match the Serial COM port speed.

Using the Stinger USB the concept is completely different from what described above as the Stinger USB detects exactly how to handle the matching between Clock Speed and Serial Port. The Clock speed is needed just to make the card operate slower or faster to perform the requested operation. Overclocking may damage the Smarcard. Make sure you do not overclock the Smartcard over the manufacturer's specifications.

The Stinger USB independently from the Smartcard Clock Speed will always output the bytes from the Virtual COM Port at the predefined speed (see *Command 01 – Set USB speed between Stinger and PC*). In this case any Smartcard can be running at any speed included in the Stinger frequencies (see *Command 02 – Set Smartcard clock*) and the Stinger will output the bytes at the desired Baudrate (see *Command 03 – Set Smartcard COM parameters*) as explained in Figure 4).

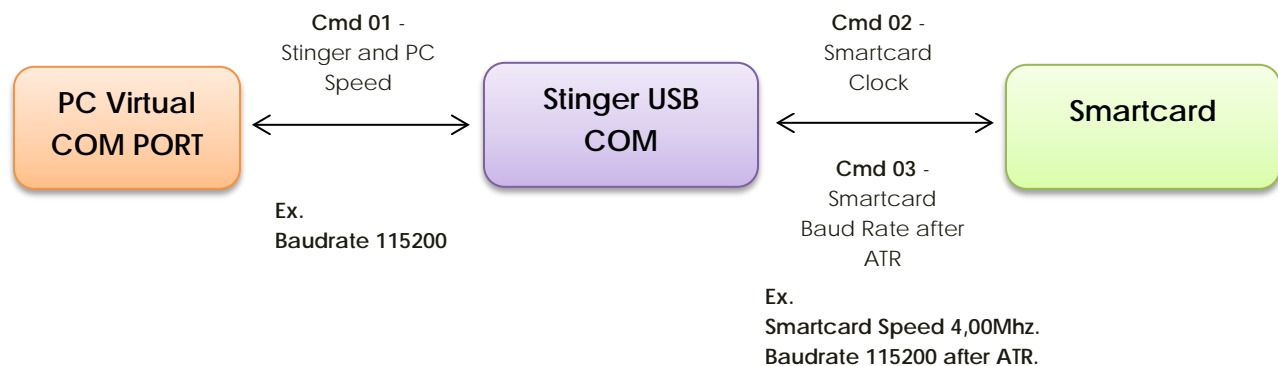


Figure 4. Command 01, 02 and 03 Setting Layout.

There is another important Setting to pay attention to which is the Baudrate after the ATR (see *Command 03 – Set Smartcard COM parameters*).

The Stinger USB by default and as a fixed value resets the Smartcard at 9600 Baud which is a standard from ISO 7816. The communication speed after the ATR may vary from 9600 to 115200 bauds accordingly to each specific Smartcard.

TIME LINE on Smartcard Pin during ATR

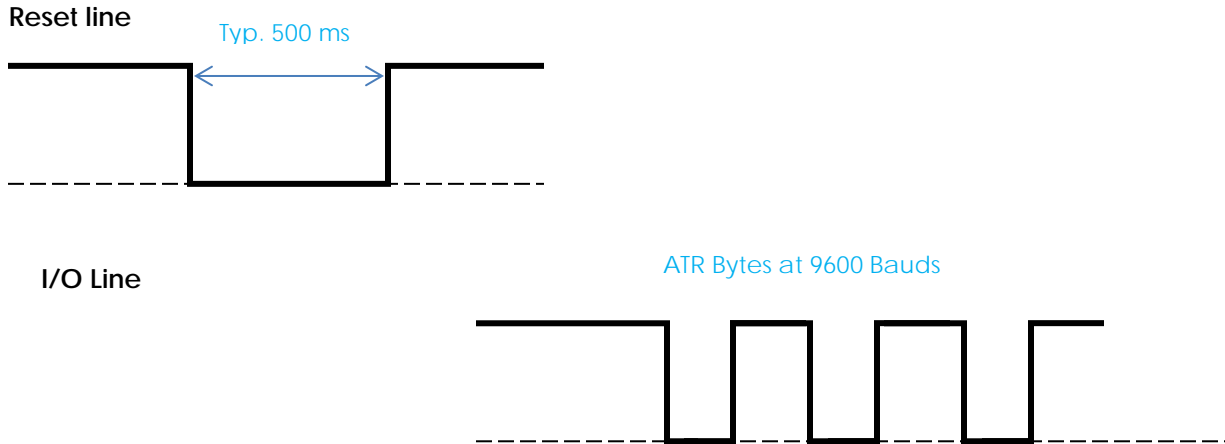


Figure 5. ATR Process on RST PIN

TIME LINE on Smartcard Pin after ATR and on PC Serial

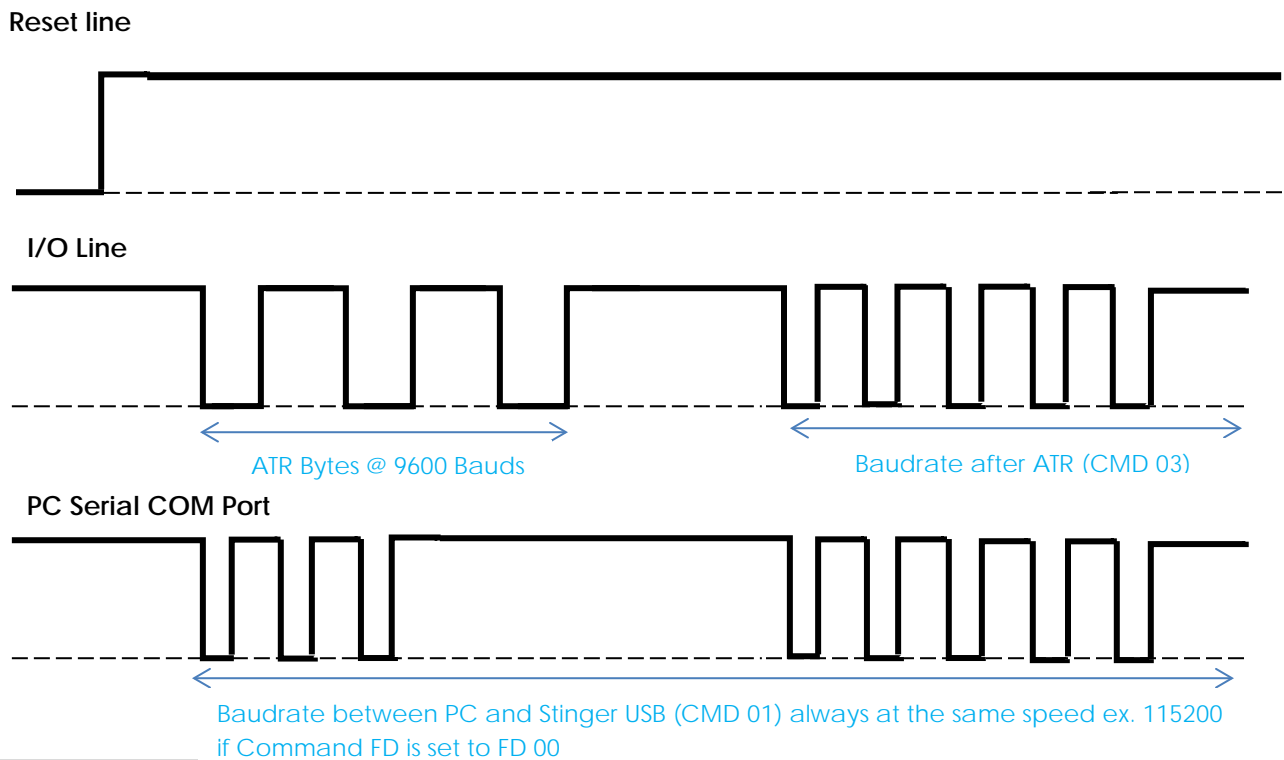


Figure 6. Smartcard Communication after ATR.

**NOTE!**

Accordingly to the above instruction the ATR comes out from the Stinger USB at the desired Baudrate set by Command 01 (ex. 115200 Baud, see *Figure 6*).

It is recommended to set the Baudrate equivalent to the standard communication speed to prevent loss of data after the ATR.

Parity Setting.

The Stinger USB after the ATR will set the Parity automatically accordingly ATR Self Detection. ATR Self Detection is a special function inside the Stinger USB that detects the 03 (Inverted ATR) or 3F (Direct ATR) first ATR byte.

If you want to set your specific Parity after the ATR you can set it using Command 03.

Smartcard Presence Detection.

The Smartcard detection is provided by the Stinger USB observing the status of **CTS Signal** on the Virtual Com Port. If the **CTS Signal is LOW** indicates that the Smartcard is inserted. If the **CTS Signal is HIGH** it means that the Smartcard is not inserted.

Smartcard Logger - Debug Mode

This is an extremely useful function which allows the user to log the traffic between a Smartcard and a Host device for debugging purposes. This function is available only in *Stinger Mode* firmware. Do not load the *Phoenix Mode* firmware to run the Smartcard logger function (*Debug Mode*).

The Stinger USB while in *Debug Mode* will automatically detect and set the clock speed accordingly to the Host Device where the Dummy Smartcard is inserted and will output on the second Virtual Com Port of the Stinger USB the results of passive logging between the Smartcard and the Host Device at the fixed speed of 115200 Baud at 8N1.

Stinger USB is capable to internally automatically adapt the clock speed of the Host device and return on the COM port a clean log of the traffic between the Smartcard and the Host Device.

To activate the *Debug Mode* please refer to Commands Description. Some parameters need to be given to the Stinger USB while setting the *Debug Mode* to allow the Stinger USB to be fully compatible with the Host Device. Please read the dedicated section to *Debug Mode* in *Command FD*.

Once the *Debug Mode* is activated it is absolutely necessary to follow the next steps to make it work properly:

Stinger – Debug Mode Initialization

1. **Make sure the Host Device is turned ON.**
2. **Activate the Debug Mode** using **Stinger USB – Setting Manager** or manually using a dedicated software or command line. (Plug out and Plug in the Stinger USB to make the sure it is now set in *Debug Mode*).
3. **Insert Dummy Smartcards as follows:**

With the Stinger USB disconnected from the PC USB Port.

- a. **Insert the Dummy Smartcard in the Lower Slot of the Stinger USB.**
 - b. **Insert the Original Smartcard in the Upper Slot of the Stinger USB.**
 - c. **Insert the Host Dummy Smartcard in the Host device.**
4. **Turn on the Stinger USB** with the three Smartcards inserted in the relative positions. This is needed to allow the Stinger USB to get and internally set the correct Baudrate obtained from the Host Device. Once the Stinger USB has successfully

obtained the correct Baudrate from the Host device a BLUE LED Light will light up. This is the sign that the Stinger USB is now ready to operate in *Debug Mode*. If you do not see the BLUE LED light UP please do TURN OFF the Stinger USB by unplugging it from the USB Port of your PC and repeat steps 1, 2, 3 and 4 again. Make sure that the three Smartcards are inserted properly in the relative Slots. A PURPLE LED Light indicates the Reset PIN on the Smartcard and will flash each time the Smartcard is resetted.

Stinger – Debug Mode Logging.

Once the initialization procedure is terminated successfully and the BLUE LED Light is turned on together with the PURPLE LED Light the Stinger USB is ready to log on the second Virtual COM port at the speed of 115200 Baud in 8N1.

1. Remove the **Dummy Smartcard** from the Host Device.
2. Remove the **Original Smartcard** from the Upper Slot.
3. Insert the **Original Smartcard** into the Upper Slot.
4. Insert the **Dummy Smartcard** in the Host Device.

Logging output will be displayed on the second Virtual COM port. Please make some tests to identify the correct Virtual COM port assigned by the Operating System in use.

Hint: In Linux the logging port should be `/dev/ttyUSB1` if Stinger USB is recognized as `/dev/ttyUSB0` and `/dev/ttyUSB1`. Using **Stinger USB – Setting Manager** the output COM port will be automatically recognized.

The output is set at 115200 Baud 8N1. Please make sure you opened the Virtual COM port with the above settings.



NOTE!

It is extremely important to follow the above step in the right sequence to make sure that the whole procedure!

Do not Unplug the Stinger USB while doing steps 1,2,3,4. In case you do so the *Debug Mode* setup procedure will fail and you will need to repeat it again.

Do not remove the Stinger Dummy Card inserted in the lower Smartcard Slot after the setup. To safely remove it please turn the Stinger USB off by

unplugging it from the USB Port.

Extra hardware needed

This function is implemented with the need of a simple adaptor that can be made as a homebrew.

At the moment there is no plan to make Dummy Smartcard adaptors but you can alternatively use existing Smartcards in PCB format to adapt an build your own Dummy Adaptors.

Please refer *Appendix A* how to make your own Smartcard debug interface.

4.2 Commands Description *(Stinger Mode)*

When the *Stinger Mode* is in use with the Stinger Mode firmware it is possible to send some commands to the Stinger USB in order to obtain information or set some values. There are two categories of Commands while in *Stinger Mode*.

1. *"On the flight" Commands.*
2. *"Non - On the flight" Commands.*

The main difference is that *"On the flight" Commands* become immediately effective without turning OFF and ON the Stinger USB and need to be saved in order to be effective after the Stinger USB is turned OFF otherwise such settings will be lost.

"Non - On the flight" Commands need the Stinger USB to be turned OFF and tuned ON to be effective and do not need to be saved after being set.

Command 00 is just a read command so it is not part of *"On the flight" Commands* nor *"Non - On the flight" Commands*.

The usage of *"On the flight" Commands* is recommended when you need to dynamically change the Smartcard parameters upon certain values normally depending upon the ATR that is received.

Using *"Non - On the flight" Commands* normally means that the developer already knows what kind of Smartcard is used and some pre-defined settings must be set at the startup. It is possible to combine the use of *"On the flight"* after the initial startup upon certain condition as the ATR result for example.

| <i>"On the Flight" Commands.</i> | <i>"Non - On the flight" Commands</i> |
|--|--|
| Command 01 <i>Set USB speed between Stinger and PC.</i> | Command FD <i>ATR Baudrate and Debug Mode</i> |
| Command 02 <i>Set Smartcard Clock.</i> | Command 06 <i>Reset to factory default</i> |
| Command 03 <i>Set Smartcard COM parameters.</i> | |
| Command 05 <i>Store user setting to flash memory.</i> | |

Command 00 – Get Stinger USB information.

Send from PC: 00

Answer from Stinger USB: 32 bytes

Answer example:

0001 0000313A 0000245D9D9100070016300522254E455B8D80 00 00 00 0000 0122

| | | | | | | | | | | | |
|-------|----------|----------|----------|----------|----------|--------|----|----|-------|------|------|
| 00-01 | 02-05 | 06 | - | - | - | 22-24 | 25 | 26 | 27-29 | 30 | 31 |
| 00 0X | 0000313A | 0000245D | 9D910007 | 00163005 | 22254E45 | 5B8D80 | 00 | 00 | NA | 0000 | 0119 |

| Byte | Description | Values | | | | | | | | | | | | | | | | | | | | | |
|---------------|---|--|------|------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|----------|-------|
| 00 | Echo Command (1 Byte) | | | | | | | | | | | | | | | | | | | | | | |
| 01 | Reader Number (1 Byte) | 01 – Upper Smartcard reader 02 – Lower Smartcard reader | | | | | | | | | | | | | | | | | | | | | |
| 02-05 | Chip Family (4 Bytes) | | | | | | | | | | | | | | | | | | | | | | |
| 06-21 | Unique Serial Number (20 Bytes) | | | | | | | | | | | | | | | | | | | | | | |
| 22-24 | Clock speed (3 byte) CMD – 02 to set | HEX Value must be converted in DEC and it will show up to clock Speed in Hz. <table border="1"> <thead> <tr> <th>HEX:</th> <th>DEC:</th> <th>MHZ:</th> </tr> </thead> <tbody> <tr> <td>345670</td> <td>3430000</td> <td>3.43</td> </tr> <tr> <td>3D0900</td> <td>4000000</td> <td>4.00</td> </tr> <tr> <td>493E00</td> <td>4800000</td> <td>4.80</td> </tr> <tr> <td>5B8D80</td> <td>6000000</td> <td>6.00</td> </tr> <tr> <td>7A1200</td> <td>8000000</td> <td>8.00</td> </tr> <tr> <td>B71B00</td> <td>12000000</td> <td>12.00</td> </tr> </tbody> </table> | HEX: | DEC: | MHZ: | 345670 | 3430000 | 3.43 | 3D0900 | 4000000 | 4.00 | 493E00 | 4800000 | 4.80 | 5B8D80 | 6000000 | 6.00 | 7A1200 | 8000000 | 8.00 | B71B00 | 12000000 | 12.00 |
| HEX: | DEC: | MHZ: | | | | | | | | | | | | | | | | | | | | | |
| 345670 | 3430000 | 3.43 | | | | | | | | | | | | | | | | | | | | | |
| 3D0900 | 4000000 | 4.00 | | | | | | | | | | | | | | | | | | | | | |
| 493E00 | 4800000 | 4.80 | | | | | | | | | | | | | | | | | | | | | |
| 5B8D80 | 6000000 | 6.00 | | | | | | | | | | | | | | | | | | | | | |
| 7A1200 | 8000000 | 8.00 | | | | | | | | | | | | | | | | | | | | | |
| B71B00 | 12000000 | 12.00 | | | | | | | | | | | | | | | | | | | | | |
| 25 | Smartcard Speed (1 byte) CMD – 03 to set | 00 == 9600 01 == 38400 02 == 115200 | | | | | | | | | | | | | | | | | | | | | |

| | | |
|-------|---|---------------------------------------|
| 26 | Smartcard Parity (1 byte) <i>CMD - 03 to set</i> | 00 == none 01 == odd 02 == even |
| 27-29 | <i>Reserved</i> | <i>Reserved</i> |
| 30-31 | Firmware version (2 bytes) | 01.25 (Decimal Value when converted) |

Command 01 – Set USB speed between Stinger and PC.

"On the flight" Command. Needs to be saved to be effective after power off.

Send from PC: 01 xx yy

Set Baudrate between Virtual COM Port USB <--> Stinger USB.
This will be the speed between Virtual COM Port and the Stinger USB.

xx --> Virtual COM Port number (00 or 01)

yy == 00 == 9600

yy == 01 == 115200

yy == 02 == 230400 **(not tested USE AT YOUR OWN RISK!)**

yy == 03 == 460800 **(not tested USE AT YOUR OWN RISK!)**

Answer from Stinger USB:

01 00 <-- OK

!=01 00 <-- KO

Command 02 – Set Smartcard Clock.

"On the flight" Command. Needs to be saved to be effective after power off.

Send from PC: 02 ss xx

Set Clock Smartcard

ss == Smartcard Number == 00 Smartcard 1 (Virtual COM Port)

ss == Smartcard Number == 01 Smartcard 2 (Virtual COM Port)

xx == Clock == 01 --> Clock1 --> 3.43

xx == Clock == 02 --> Clock1 --> 4.00

xx == Clock == 03 --> Clock1 --> 4.80

xx == Clock == 04 --> Clock1 --> 6.00

xx == Clock == 05 --> Clock1 --> 8.00

xx == Clock == 06 --> Clock1 --> 12.00

Answer from Stinger USB:

0200 <-- OK

!=0200 <-- KO (Error Number)

Command 03 – Set Smartcard COM parameters.

"On the flight" Command. Needs to be saved to be effective after power off.

Send from PC: 03 ss yy xx (nn)

ss == Smartcard Number == 00 Smartcard 1

ss == Smartcard Number == 01 Smartcard 2

Baudrate

yy -> 00 == 9600,
01 == 38400,
02 == 115200



NOTE!

This is the Serial COM Port speed and Parity after the Smartcard Answer To Reset. ATR Baudrate is always at 9600 Baud.

Parity

xx -> 00 == none, 01 == odd, 02 == even



NOTE!

STOP BITs is fixed to 1.

Answer from Stinger USB:

0300 <-- OK

!=0300 <-- KO

Command 05 – Store user setting to flash memory.

“On the flight” Command. Needs to be saved to be effective after power off.

Send from PC: 05

Store user setting to flash memory

Answer from Stinger USB:

0500 <-- OK

!=0500 <-- KO

Command 06 – Reset to factory default.

“Non On the flight” Command. Effective after Power OFF and Power ON.

Send from PC: 06

Factory resets are:

9600 Bauds (*between Reader and Smartcard after the ATR*) – **8N1** – **4.00 Mhz**

Answer from Stinger USB:

0600 <-- OK

!= 0600 <-- KO

Command FD – ATR Baudrate and Debug Mode.

“Non On the flight” Command. Effective after Power OFF and Power ON.

FD Command is extremely important to set the correct working mode on your Stinger USB while in *Stinger Mode*. FD Command is dedicated to set three possible configurations:

Send from PC: FD XX

1. FD 00 – Sets ATR output at 115200 Baud instead of standard 9600 baud.
2. FD 01 – Sets ATR output at 9600 Baud. This is the conventional ATR value.
3. FD 02 – Sets the Stinger USB in *Debug Mode*.

Factory resets are:

FD 00

Answer from Stinger USB:

FD00 <-- OK

!= FD00 <-- KO

It is also possible to set directly the Smartcard parameters in one shot using the FD extended expression:

| CMD | MODE | COM_0 | CLK_0 | PARITY_0 | BAUDRATE_0 | COM_1 | CLK_1 | PARITY_1 | BAUDRATE_1 |
|-----|------|-------|-------|----------|------------|-------|-------|----------|------------|
| FD | 00 | 00 | CC | PP | BB | 01 | CC | PP | BB |

| CLOCK | PARITY | BAUDRATE |
|-----------------|---------------|---------------|
| CC | PP | BB |
| 01 == 3,43 Mhz | 00 == None | 00 == 9600 |
| 02 == 4,00 Mhz | 01 == Odd | 01 == 38400 |
| 03 == 4,80 Mhz | 02 == Even | 02 == 115200 |
| 04 == 6,00 Mhz | FF == Default | FF == Default |
| 05 == 8,00 Mhz | | |
| 06 == 12,00 Mhz | | |
| FF == Default | | |

Example:

FD 00 00 01 00 02 01 04 02 01

Sets ATR at 115200, COM 00 at 3,43Mhz with Parity NONE and Baudrate after ATR at 115200 baud + COM 01 at 6,00Mhz with Parity EVEN and Baudrate after ATR at 38400 baud.

Command FD – Activate Debug Mode

Please set the following parameters during activation:

| CMD | MODE | PARITY | BAUDRATE | ADDED_STOP_BIT |
|-----|------|---------------|---------------|----------------|
| FD | 02 | PP | BB | NN |
| | | 00 == None | 00 == 9600 | From 0 to 200 |
| | | 01 == Odd | 01 == 38400 | FF == Default |
| | | 02 == Even | 02 == 115200 | |
| | | FF == Default | FF == Default | |

Example:

FD 02 00 01 100

Sets Debug Mode at No Parity, Baudrate (after ATR) at 38400 and 100 Stop Bits. Stop bits may be adjusted accordingly to the Host Device.

5. UniProg Mode

(working with UniProg Firmware)

(working with UniProg Mode Firmware)

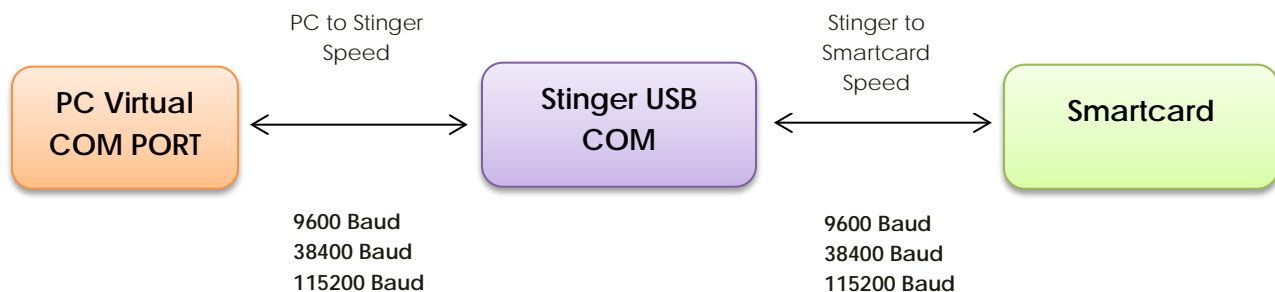
The *UniProg Mode* requires to load on the Stinger USB the *UniProg Mode* dedicated firmware that can be downloaded from www.stingerusb.com official web site.

In *UniProg Mode* it is possible to set the Clock Speed to communicate with the Smartcard and the Baudrate between the PC to Stinger USB and the Stinger USB to Smartcard after the ATR.

Clock for Stinger USB in *UniProg Mode* can be set at the following speed:

| Value for Configuration: | HEX: | DEC: | MHZ: |
|--------------------------|--------|----------|-------|
| 01 | 368120 | 3572000 | 3.572 |
| 02 | 384E10 | 3690000 | 3.69 |
| 03 | 3D0900 | 4000000 | 4.00 |
| 04 | 493E00 | 4800000 | 4.80 |
| 05 | 5B8D80 | 6000000 | 6.00 |
| 06 | 7A1200 | 8000000 | 8.00 |
| 07 | B71B00 | 12000000 | 12.00 |

Baudrate Speed between PC to Stinger USB and Stinger USB to Smartcard can be represented as follows:



While operating in *UniProg Mode* Parity and Stop Bits do not need to be set as those settings are handled directly from the Stinger USB during the automatic ATR at smartcard insertion. The ATR will be returned to the PC after the classic ATR command through the RTS PIN. The

After loading the *UniProg Mode* firmware both LEDs will light up in Blue Color and when the Smartcard is inserted and the automatic ATR is performed the LEDs will blink shortly in Red Color.

To enter the command mode and change the settings (two Baudrate speeds and Clock) of the Upper Slot and Lower Slot it is necessary to:

1. Remove all Smartcards from Upper and Lower Slot.
2. Set **RTS signal LOW** for at least 50ms
3. Set **DTR signal HIGH**
4. Wait 50ms since the **DTR signal is HIGH** send the Configuration command explained in the next section.

To successfully communicate to the Stinger USB while is *UniProg Mode* in command mode (settings) the Virtual COM port must be set at:

9600 Baud - 8N1

Depending on the programming language and Operating System the programming routines may vary but it is strictly necessary to communicate to the Stinger USB at the above Baudrate and follow the port configuration at 8Bit, No Parity and 1 Stop Bit.

It is alternatively possible to use **Stinger USB – Setting Manager** or the other open source code available on www.stingerusb.com to set the clock speed and load the appropriate firmware.



NOTE!

The Stinger USB in *Stinger Mode* communicates at 115200Baud at 8N1 but in *UniProg Mode* the setting must be 9600 8N1 while in command mode (settings).

It is absolutely mandatory to follow the command mode entering sequence to send to the Stinger while in UniProg Mode.

After every setting change please plug off and plug in the Stinger USB in order to activate the new configuration!

5.2 Commands Description *(UniProg Mode)*

Command 00 – Get Stinger USB information.

Send from PC: 00

Answer from Stinger USB: 32 bytes

Answer example:

0001 0000313A 0000245D9D9100070016300522254E455B8D80 00 00 00 0000 0122

| | | | | | | | | | | | |
|--------------|--------------|-----------|----------|----------|----------|--------------|-----------|-----------|--------------|-----------|-----------|
| 00-01 | 02-05 | 06 | - | - | - | 22-24 | 25 | 26 | 27-29 | 30 | 31 |
| 00 0X | 0000313A | 0000245D | 9D910007 | 00163005 | 22254E45 | 5B8D80 | 00 | 00 | NA | 0000 | 0119 |

| Byte | Description | Values | | | | | | | | | | | | | | | | | | | | | | | | |
|---------------|--|--|------|------|------|---------------|---------|-------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|---------|------|---------------|----------|-------|
| 00 | Echo Command (1 Byte) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 01 | Reader Number (1 Byte) | 00 – Upper Smartcard reader 01 – Lower Smartcard reader | | | | | | | | | | | | | | | | | | | | | | | | |
| 02-05 | Chip Family (4 Bytes) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 06-21 | Unique Serial Number (20 Bytes) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 22-24 | Clock speed (3 byte) <i>CMD – 44 to set</i> | HEX Value must be converted in DEC and it will show up to clock Speed in Hz. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>HEX:</th> <th>DEC:</th> <th>MHZ:</th> </tr> </thead> <tbody> <tr> <td>368120</td> <td>3572000</td> <td>3.572</td> </tr> <tr> <td>384E10</td> <td>3690000</td> <td>3.69</td> </tr> <tr> <td>3D0900</td> <td>4000000</td> <td>4.00</td> </tr> <tr> <td>493E00</td> <td>4800000</td> <td>4.80</td> </tr> <tr> <td>5B8D80</td> <td>6000000</td> <td>6.00</td> </tr> <tr> <td>7A1200</td> <td>8000000</td> <td>8.00</td> </tr> <tr> <td>B71B00</td> <td>12000000</td> <td>12.00</td> </tr> </tbody> </table> | HEX: | DEC: | MHZ: | 368120 | 3572000 | 3.572 | 384E10 | 3690000 | 3.69 | 3D0900 | 4000000 | 4.00 | 493E00 | 4800000 | 4.80 | 5B8D80 | 6000000 | 6.00 | 7A1200 | 8000000 | 8.00 | B71B00 | 12000000 | 12.00 |
| HEX: | DEC: | MHZ: | | | | | | | | | | | | | | | | | | | | | | | | |
| 368120 | 3572000 | 3.572 | | | | | | | | | | | | | | | | | | | | | | | | |
| 384E10 | 3690000 | 3.69 | | | | | | | | | | | | | | | | | | | | | | | | |
| 3D0900 | 4000000 | 4.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 493E00 | 4800000 | 4.80 | | | | | | | | | | | | | | | | | | | | | | | | |
| 5B8D80 | 6000000 | 6.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 7A1200 | 8000000 | 8.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| B71B00 | 12000000 | 12.00 | | | | | | | | | | | | | | | | | | | | | | | | |
| 25 | ND | ND | | | | | | | | | | | | | | | | | | | | | | | | |
| 26 | ND | ND | | | | | | | | | | | | | | | | | | | | | | | | |
| 27-29 | <i>Reserved</i> | <i>Reserved</i> | | | | | | | | | | | | | | | | | | | | | | | | |
| 30-31 | Firmware version (2 bytes) | 01.25 (Decimal Value when converted) | | | | | | | | | | | | | | | | | | | | | | | | |

Command 44 – Set Stinger Clock and Baudrates in UniProg Mode.

Send from PC: 0x44 0x00 **c0 c1 br0 br1 bc0 bc1** 0x00 0x00

c0 = Smartcard Upper Slot Clock Speed

c1 = Smartcard Lower Slot Clock Speed

br0 = Baudrate from Stinger USB to Smartcard Upper Slot

br1 = Baudrate from Stinger USB to Smartcard Lower Slot

bc0 = Baudrate from PC to Stinger USB for Smartcard Upper Slot

bc1 = Baudrate from PC to Stinger USB for Smartcard Lower Slot

Leave the last two bytes at 0x00 0x00

Clock Speed Values:

01 == 3.572 Mhz

02 == 3,69 Mhz

03 == 4.00 Mhz

04 == 4.80 Mhz

05 == 6.00 Mhz

06 == 8.00 Mhz

07 == 12.00 Mhz

FF == Default Value (3,69Mhz)

Baudrate Values:

00 == 9600

01 == 38400

02 == 115200

FF == Default Value (9600 Baud)

Example:

0x44 0x00 **0x02 0x05 0x02 0x00 0x00 0x00** 0x00 0x00

Sets Upper Smartcard Slot at 3.69 Mhz Baudrate from Stinger USB to Smartcard 115200 baud and from PC to Stinger USB at 9600 baud.

Sets Lower Smartcard Slot at 6.00 Mhz Baudrate from Stinger USB to Smartcard 9600 baud and from PC to Stinger USB at 9600 baud.

From Stinger USB to PC:

Write OK: 0x4400

Write ERROR: 0x44xx

Command 42 – Read Stinger Clock and Baudrates in UniProg Mode.

Send from PC: 0x42

Receive from Stinger USB: 0x42 0x00 0x00 **c0 c1 br0 br1 bc0 bc1** 0x00 0x00

c0 = Smartcard Upper Slot Clock Speed

c1 = Smartcard Lower Slot Clock Speed

br0 = Baudrate from Stinger USB to Smartcard Upper Slot

br1 = Baudrate from Stinger USB to Smartcard Lower Slot

bc0 = Baudrate from PC to Stinger USB for Smartcard Upper Slot

bc1 = Baudrate from PC to Stinger USB for Smartcard Lower Slot

Clock Speed Values:

01 == 3.572 Mhz

02 == 3,69 Mhz

03 == 4.00 Mhz

04 == 4.80 Mhz

05 == 6.00 Mhz

06 == 8.00 Mhz

07 == 12.00 Mhz

FF == Default Value

Baudrate Values:

00 == 9600

01 == 38400

02 == 115200

FF == Default Value

Example:

0x42 0x00 0x00 **0x02 0x05 0x02 0x00 0x00 0x00** 0x00 0x00

Upper Smartcard Slot is configured at 3.69 Mhz Baudrate from Stinger USB to Smartcard 115200 baud and from PC to Stinger USB at 9600 baud.

Lower Smartcard Slot is configured at 6.00 Mhz Baudrate from Stinger USB to Smartcard 9600 baud and from PC to Stinger USB at 9600 baud.

Command 45 – Erase Stinger Clock and Baudrates in *UniProg Mode*.

Send from PC: 0x45

Resets to standard values:

Upper Smartcard Slot is configured at 3.69 Mhz Baudrate from Stinger USB to Smartcard 9600 baud and from PC to Stinger USB at 9600 baud by default.

Lower Smartcard Slot is configured at 3.69 Mhz Baudrate from Stinger USB to Smartcard 9600 baud and from PC to Stinger USB at 9600 baud by default.

From Stinger USB to PC:

Write OK: 0x4500

Write ERROR: 0x45xx

6. C Source code examples

This code can be used in POSIX systems like Linux or MACOSX.

```
/* Common Libraries to be used */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <limits.h>
#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <termios.h>
```

- Step 1 – Open COM Port

```
int fd = -1;
fd = open ("/dev/ttyUSB0", O_RDWR | O_NOCTTY | O_NDELAY);

if (fd<0)
{
    fprintf(stderr, "*** ERROR - Couldn't open COM PORT\n ");
    exit(1);
}
```

- Step 2 – Configure COM Port

```
struct termios port_settings; // structure to store the port settings in
fcntl (fd, F_SETFL, 0); // clear all flags
tcgetattr(fd, &port_settings); // Read Options
cfsetspeed(&port_settings, B115200);
port_settings.c_cflag |= (CLOCAL | CREAD | CS8); // set no parity, stop bits, data bits
port_settings.c_cflag &= ~(PARENB | CSTOPB); // set no parity, stop bits
tcsetattr(fd, TCSANOW, &port_settings); // Write new settings
```

- Step 3 – Set DTR to LOW

```
int controlbits = TIOCM_DTR;
ioctl (fd, TIOCMBIS, &controlbits);
```

- Step 4 – Set RTS to HIGH

```
int controlbits = TIOCM_RTS;
ioctl (fd, TIOCMBIS, &controlbits);
```

- Step 5 – Send a Command

```
char result [256];
char temp [256];

int res = 0;

memset (result, 0, 256);
memset (temp, 0, 256);

temp [0] = 0x00;

// Write 1 byte to device
res = write (fd, temp, 1);

if (res<0)
{
    fprintf(stderr,"Write Command Failed!\n");
    return;
}

// Wait 200ms
usleep (200000);

// Read 256 bytes as answer from device
res = read (fd, &result, 256);

if (res<0)
{
    fprintf(stderr,"No answer\n");
    return;
}
else
{
    fprintf(stderr,"No answer\n");
    for (i = 0; i < res; i++)
    {
        fprintf(stderr,"%02X", result [i] & 0xFF);
    }
}

// Close File Descriptor (COM Port)
close (fd);
```

6.2 Copy and Paste Code

```
// Created for www.stingerusb.com on 25/10/13.
// Copyright (c) 2013 cc. All rights reserved.

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <limits.h>
#include <assert.h>
#include <errno.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <sys/mount.h>
#include <termios.h>

int main(int argc, const char * argv[])
{
    int fd = -1;
    int i = 0;
    char result [256];
    char temp [256];
    int res = 0;
    int controlbits = 0;
    struct termios port_settings; // structure to store the port settings in
    fd = open ("/dev/tty.usbserial-000012FDA", O_RDWR | O_NOCTTY | O_NDELAY);
    if (fd<0)
    {
        fprintf(stderr, "*** ERROR - Couldn't open COM PORT\n ");
        exit(1);
    }

    fcntl (fd, F_SETFL, 0); // clear all flags
    tcgetattr(fd, &port_settings); // Read Options
    cfsetspeed(&port_settings, B115200);
    port_settings.c_cflag |= (CLOCAL | CREAD | CS8); // set no parity, stop bits, data bits
    port_settings.c_cflag &= ~(PARENB | CSTOPB); // set no parity, stop bits
    tcsetattr(fd, TCSANOW, &port_settings); // Write new settings

    controlbits = TIOCM_DTR;
    ioctl (fd, TIOCMBIS, &controlbits);
    controlbits = TIOCM_RTS;
    ioctl (fd, TIOCMBIS, &controlbits);

    memset (result, 0, 256);
    memset (temp, 0, 256);
    temp [0] = 0x00;

    // Write 1 byte to device
    res = write (fd, temp, 1);
    if (res<0)
    {
        fprintf(stderr, "Write Command Failed!\n");
    }

    usleep (200000); // Wait 200ms

    // Read 256 bytes as answer from device
    res = read (fd, &result, 256);
    if (res<0)
        fprintf(stderr, "No answer\n");
    else
    {
        fprintf(stderr, "Answer:\n");
        for (i = 0; i < res; i++)
        {
            fprintf(stderr, "%02X ", result[i]& 0xFF);
        }
    }

    close (fd); // Close File Descriptor (COM Port)
}
```

6. How to compile and run the `conf.c`

Tested and compiled under Ubuntu 12.xx and Ubuntu 13.xx and MACOSX

Make your own source code or use the `config.c` open source code to test.

No special libraries are needed.

1. Download the latest open source code from www.stingerusb.com and update your Stinger to the latest firmware version.

2. Unpack the archive.

3. Compile as follows:

```
gcc config.c -o config
```

The compile `config` file will be generated.

4. Check on which Virtual Com Port your Stinger is associated to with command:

```
ls /dev/tty*
```

5. Identify your programmer device ID for example `/dev/ttyUSB0` in Linux or `/dev/tty.usbserial-000012xxx` in MACOSX.

6. Launch a terminal shell and run this command line:

```
sudo ./config /dev/ttyxxxxx [arguments]
```

Check your Stinger configuration

[arguments] ? shows actual Stinger configuration

Example: `sudo ./config /dev/ttyUSB0 ?`

Set your Stinger configuration

[arguments] clock_speed baudrate parity

Example: `sudo ./config /dev/tty.USB0 400 115200 NONE`

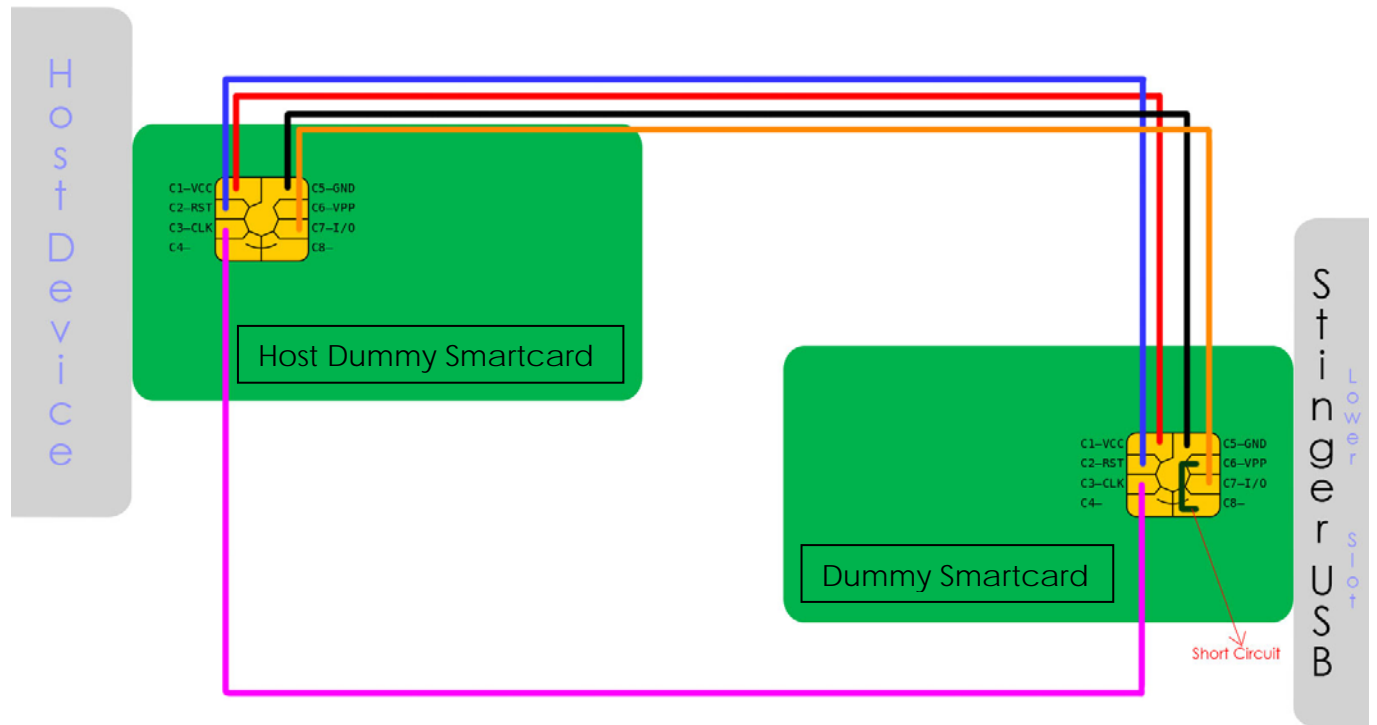
7. At the prompt type your system password and let it go!

8. When finished unplug and plug in the Stinger again.

Remember that the new configuration will be active after powering off and on the Stinger.

Make sure you run program using the sudo and make sure your file path is correct!

7. Appendix A - "Dummy Card"



| Dummy Card - Host Device | Dummy Card - Stinger |
|--------------------------|--|
| VCC | VCC |
| GND | GND |
| RST | RST |
| CLK | CLK |
| I/O (Data Line) | I/O (Data Line) |
| | <i>Short Circuit Between C6 and C8</i> |