



REVELPROG IS

user manual



Last update (v1.6.2): 2017-11-20

Please read user manual before using the device.

Contents

1. Technical data and requirements	- 5 -
2. Features and applications	- 6 -
3. Installation of software and USB drivers.....	- 7 -
4. First steps with REVELPROG-IS application.....	- 8 -
4.1. User interface.....	- 8 -
4.2. Device programming	- 10 -
4.2.1. Reading	- 10 -
4.2.2. Writing.....	- 11 -
4.2.3. Erasing	- 11 -
4.3. HEX editor features (buffer editing).....	- 12 -
4.3.1. HEX editor	- 12 -
4.3.2. Appearance	- 12 -
4.3.3. Fill buffer.....	- 13 -
4.3.4. Patterns - export changes to file.....	- 14 -
4.3.5. Swap bytes.....	- 14 -
4.3.6. Clear buffer or generate random buffer	- 15 -
4.3.7. CRC32 and Checksum-16	- 15 -
4.3.8. File format conversion: .bin <-> .hex	- 15 -

4.3.9. DiffChecker (file comparing)	- 16 -
4.4. Additional features	- 17 -
4.4.1. Blank test.....	- 17 -
4.4.2. Compare with the buffer.....	- 17 -
4.4.3. VPP selection	- 17 -
4.4.4. Device identification (Chip ID) for FLASH SPI	- 18 -
4.4.5. Quad SPI (QSPI) configuration for FLASH SPI	- 18 -
4.4.6. Auto backup	- 18 -
5. In-Circuit Serial Programming (ICSP).....	- 19 -
5.1. Addressing in external circuit.....	- 19 -
5.2. Write protection in circuit (WP/PRE pin)	- 20 -
5.2. Power supply in external circuit	- 20 -
5.3. Limits and possible issues during in-circuit programming	- 21 -
5.4. Troubleshooting for in-circuit programming	- 22 -
6. Custom Script Programming (I2C / SPI bus)	- 24 -
6.1. Script construction	- 24 -
6.2. Script syntax	- 25 -
6.3. Script maker with GUI	- 25 -
6.4. Script execution sequence.....	- 26 -
6.5. Script examples.....	- 26 -
7. Updates.....	- 27 -

8. Research and development	- 28 -
9. APPENDIX A - List of supported devices.....	- 29 -
8.1. EEPROM I2C 24... series	- 29 -
8.2. EEPROM SPI 25... series	- 31 -
8.3. EEPROM SPI 35... series	- 32 -
8.4. EEPROM SPI 95... series	- 32 -
8.5. EEPROM MICROWIRE 93... series.....	- 33 -
8.6. FLASH SPI 25... series.....	- 35 -
8.7. FLASH SPI 26... series.....	- 37 -
8.7. FLASH SPI 45... series.....	- 37 -
8.8. FRAM I2C 24... series.....	- 38 -
8.9. FRAM SPI 25... series	- 38 -
8.10. Optical and ethernet modules (SFP/SFP+ transceivers).....	- 38 -
8.11. Your suggestions.....	- 39 -
10. APPENDIX B - video tutorials and technical support	- 39 -

1. Technical data and requirements

REVELPROG-IS made by **REVELTRONICS** is a professional tool for serial memory programming. It supports wide range of **EEPROM**, **FLASH** and **FRAM** devices with **I2C**, **SPI** and **µWire** protocols powered with **1.0V - 5.0V** voltage levels. It has integrated **ZIF** socket (for THT devices) and **SOIC** socket (for SMD devices). It has also connector for **in-circuit serial programming (ICSP)** with buffered I/O. Due to **USB 2.0** interface memory operations are really fast.

Technical data:

- power supply and communication via USB 2.0 (hardware level, bulk transfer),
- supported protocols: I2C, SPI, MICROWIRE,
- supported devices with 1.0V - 5.0V power supply,
- programming device powered from programmer (1.0 - 5.0V, I_{max} = 100mA) or from external circuit (1.0 - 5.5V),
- operation speed (read/write) up to 10Mbit/s,
- ZIF-32 + SOIC-8 (150mil) integrated sockets,
- connector for in-circuit serial programming (ICSP) with buffered I/O,
- auto adaptive I/O pins connected only for programming time (Hi-Z in IDLE state),
- short circuit and overload protection,
- device manufactured in EU, made in Poland.

Requirements:

- USB 2.0 or above,
- Windows XP, Vista, 7, 8 or 10 (32 and 64 bit) + .NET Framework 4¹ installed in OS.

¹ .NET 4.0 Client Profile can be downloaded from MICROSOFT: <http://www.microsoft.com/en-us/download/details.aspx?id=17718>

2. Features and applications

REVELPROG-IS is used in many applications such as: **car electronics & automotive** (reprogramming modules, radio-code, mileage correction, immobilizers, gps navigations), **computer service** (data recovery, reflashing/upgrading/programming BIOS chips on motherboards of PC, notebooks, copier & fax machines, printers etc.), **RTV/AGD service** (TV & TV tuners, digital cameras, multifunction devices etc.) and many other devices equipped with serial I2C/SPI/MICROWIRE EEPROM and FLASH chips. Due to the possibility of change voltage level (in range of **1.0 - 5.0V**) programmer supports both older and newer devices (e.g. latest devices equipped with 1.8V low-voltage chips). Device is equipped with **ICSP connector** that allows in-circuit serial programming without unsoldering the chip from board - **built in additional protection** (buffered I/O, short circuit and overload protection) increase reliability and safety of such operations.

Main features:

- reading / writing / erasing functions with user-friendly application interface,
- supported binary files (*.bin, *.rom, *.epp etc.) and Intel-Hex (*.hex) with possibility to load file to the buffer, edit it, calculate checksums, swap-bytes, save modified bytes as patterns, compare with chip etc.,
- programming in socket or in external circuit (without unsoldering the chip),
- the target application device can be either powered by REVELPROG (possibility to change VPP and logic levels: 1.0V / 1.5V / 1.8V / 2.5V / 2.8V / 3.3V / 5.0V) or powered by an application with external power supply (1.0 – 5.5V),
- build in additional protection: buffered I/O pins internal connected only during programming + short circuit and overload protection (reduces the risk of damaging parts due to connection mistake),
- integrated sockets (ZIF + SOIC) with auto adaptive I/O pins reduces requirement of additional adapters,
- short programming time², e.g. WINBOND W25Q80BV (8Mb): read 0.9s (8,4Mb/s), write 3.3s (2,4Mb/s), MICRON N25Q128A (128Mb): read 12.8s (10Mb/s), write 37.6s (3,4Mb/s),
- useful tools such as: save/load patterns with changes, i2c/spi custom scripts, auto backups, file comparing etc.

² Due to USB 2.0 hardware layer (BULK mode) maximum real transfer is up to 10Mbit/s for read/write operations but in many cases transfer is limited by read/write time cycle from/to memory (e.g. write is always slower than read). Write time does not include chip erase time (you can find tCE in device datasheet) and verification (read) time. There is possibility to change write sequence (e.g. without erasing and verifying) but most devices require at least chip erase before write operation.

3. Installation of software and USB drivers

Requirements:

PC application is compatible with MS Windows³ XP, Vista, 7 , 8 and 10 (32 and 64-bit versions). You will need also .NET Framework 4.0 to run application (probably already installed⁴ in your OS)

Software installation:

Software installation is typical for Windows. Just run setup (*REVELPROG-IS_v1-x_Setup*) and follow instructions (next, next, finish;). You may be prompted for USB driver installation (accept it). Shortcut to REVELPROG-IS app will be added to the desktop. Application is ready to run.

Drivers installation:

In most cases drivers will be successful installed during application installation. Drivers may be installed also later from application (menu "help" -> "install drivers").

Windows XP users will have to install drivers after first device plugged-in - just select drivers location in installation folder (by default it is "*C:\Program Files\REVELTRONICS\REVELPROG-IS\drivers\REVELPROG-IS*"). If you miss it, you can install drivers later manually (run device manager, find "REVLPROG-IS" device on the list, right click and update driver, select driver location).

Windows 8/10 x64 users will have to install drivers from REVELPROG-IS application by using "Zadig" installer (menu "help" -> "install drivers with Zadig"). You should select REVELPROG-IS device on the list and install WinUSB driver, video: <https://www.youtube.com/watch?v=6AFD3yBsRnc>

³ Windows XP with SP3 installed (service pack 3). Windows 7 with SP1 installed.

In case of "fresh" install of OS you may need also "Microsoft Visual C++ 2010 Redistributable Package" (<http://www.microsoft.com/en-us/download/details.aspx?id=5555>)

⁴ if NOT, you can download and install it directly from Microsoft web page: <http://www.microsoft.com/en-us/download/details.aspx?id=17718>

4. First steps with REVELPROG-IS application

4.1. User interface

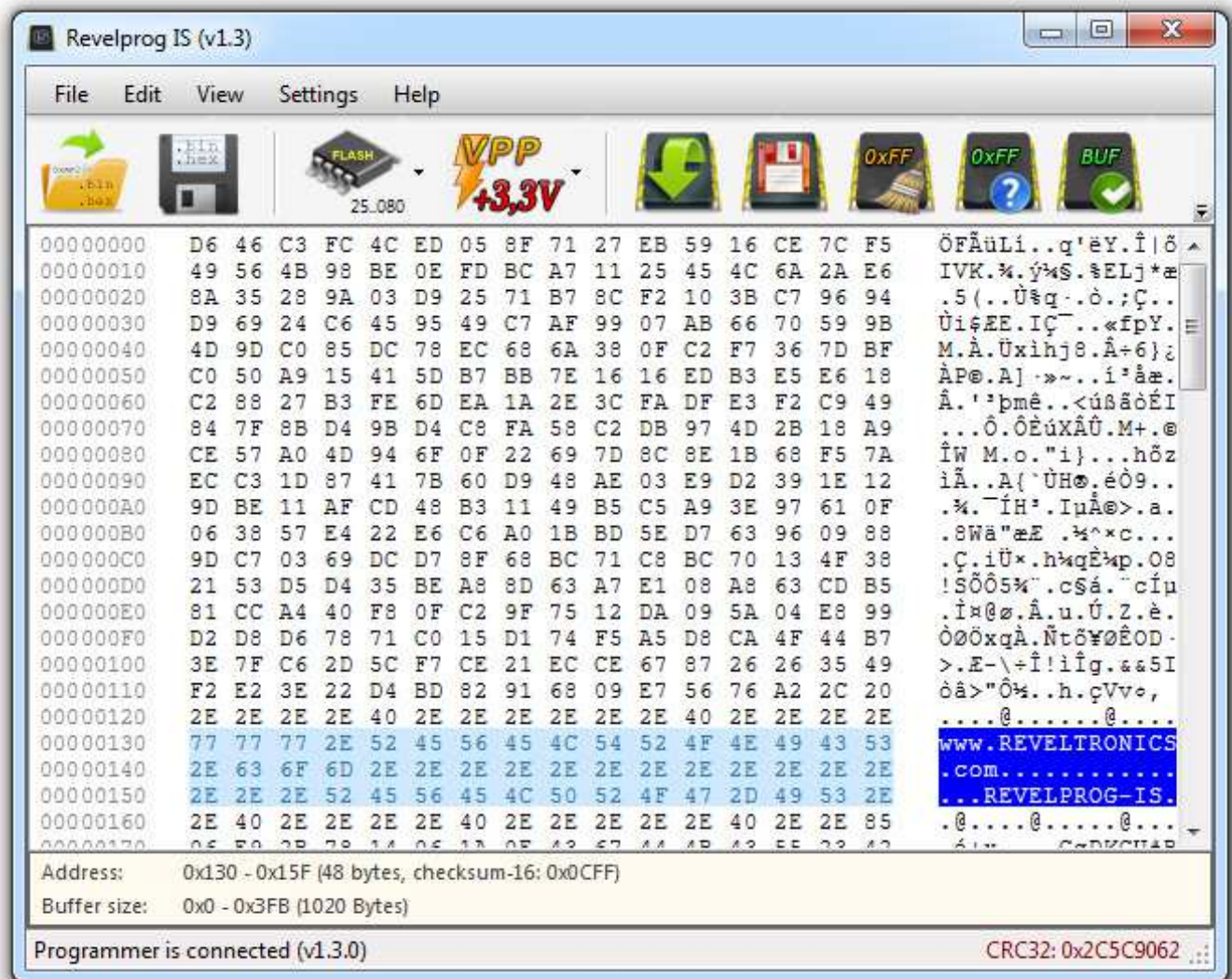
Application interface is user-friendly. All main functions can be run directly from shortcut icons (all icons are activated after device connected). Please move cursor above specific icon to see additional tooltip.



Memory buffer is displayed in hexadecimal + ASCII format. There is possibility to edit buffer as bytes and as ASCII characters.

Additional information is displayed in lower area:

- buffer information: buffer size in bytes, buffer addresses range and current address (or addresses range) of cursor position (selection) with calculated checksum-16 (from selected bytes),
- status bar (programmer version info or last operation info),
- CRC-32 checksum calculated from buffer or opened file (click on "CRC32" to recalculate checksum)



4.2. Device programming

Device type should be selected from the device database before any operation ("chip select" icon). Voltage will be set automatically. It can be changed manually by clicking VPP icon. In case of ICSP programming with external power supply, VPP should be set to the same level as memory power supply.

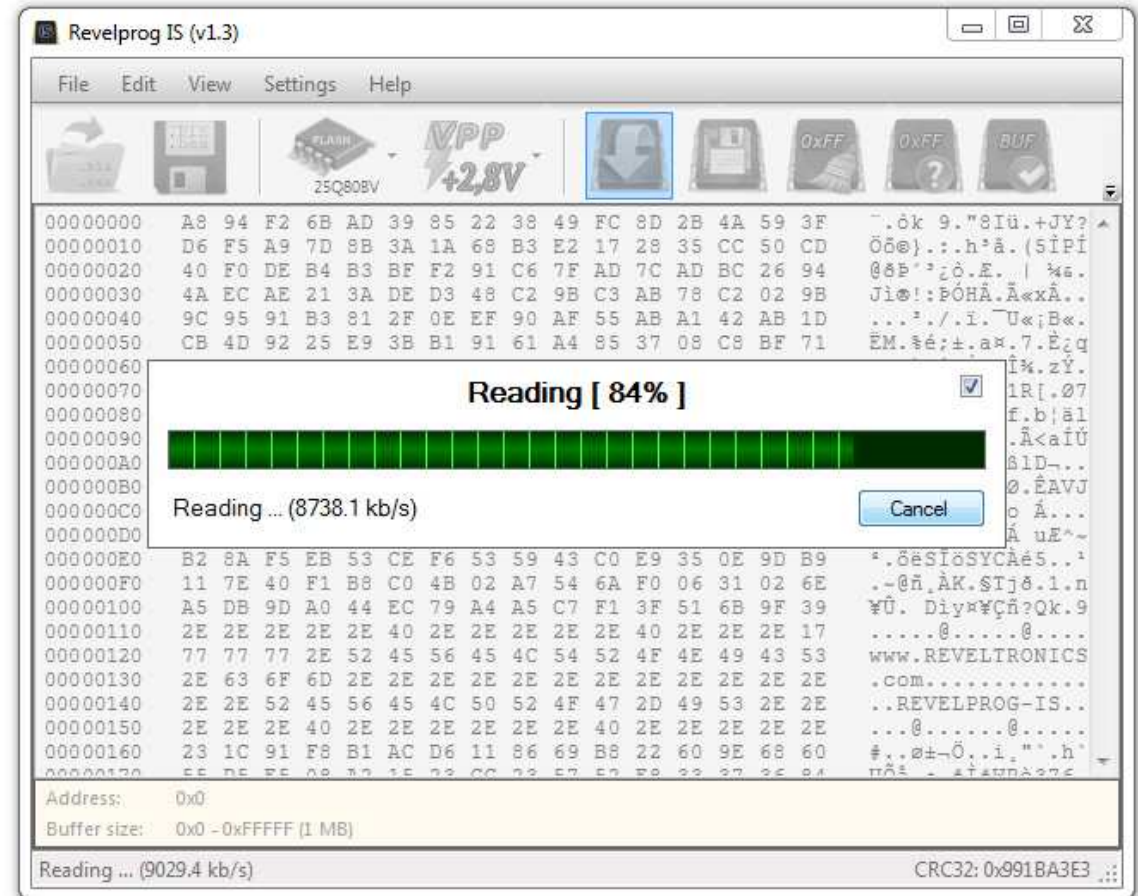
4.2.1. Reading

Before device reading operation, memory type should be chosen. Please check also VPP voltage (you may need to change it in case of low-voltage chips). Click READ icon (memory chip with green arrow).

Programmer will read device. Memory content will be loaded into the buffer. Buffer can be saved to external file. You can also edit the buffer and reprogram the device.

Please note at the CRC32 checksum (right corner at the bottom). If checksum is equal after each read than the risk of read with error usually does not exist. It is a good practice to make memory backup to external file before any writing.

The reading may be incorrect due to communication error or broken chip.



4.2.2. Writing

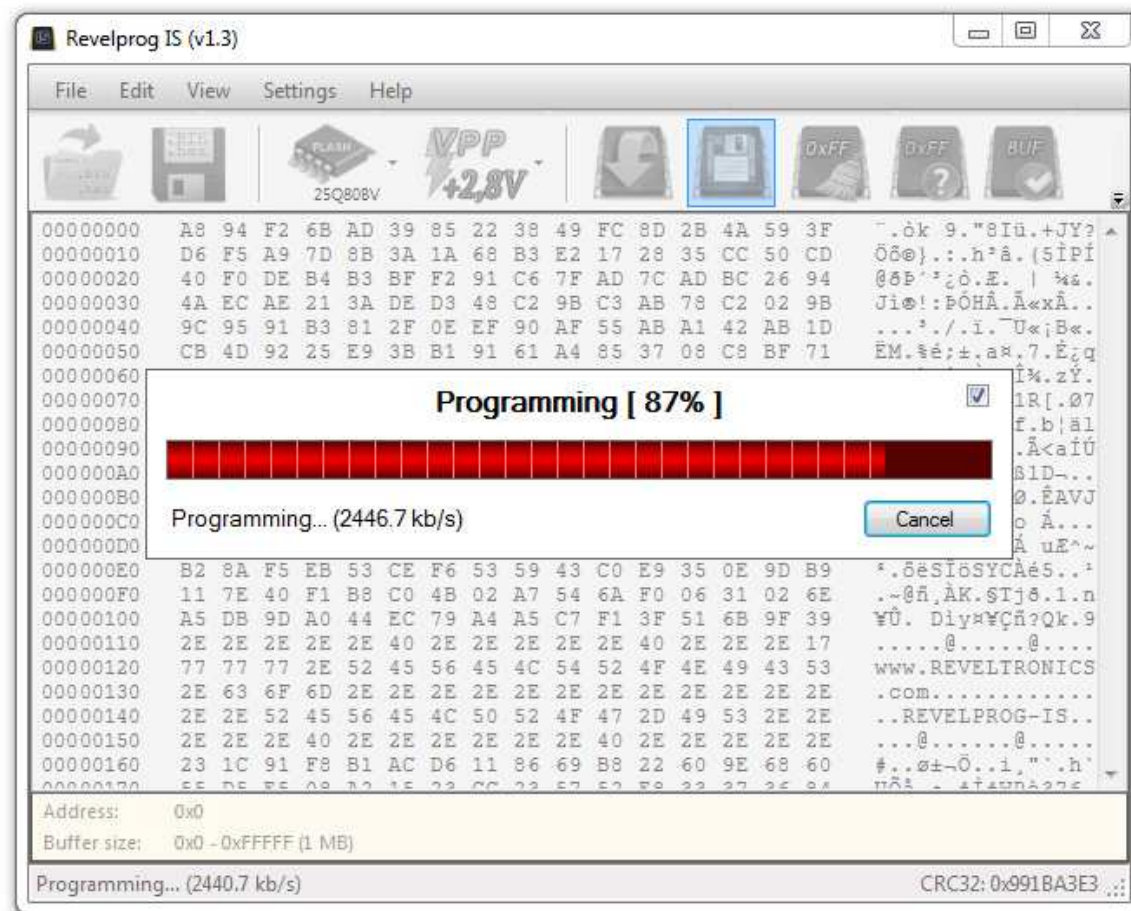
Before device programming operation, (1) memory type should be chosen and (2) buffer should be filled with data. You can load data from external binary (*.bin, *.rom, *.epp etc.) or Intel-Hex (*.hex) file. Please check also VPP voltage – it may be changed in case of low-voltage chips. Please click “WRITE” icon (memory chip with red-orange floppy disk).

Programmer will erase the device, write data to memory from the buffer and make verification process. There is possibility to change writing sequence from menu: *Settings* -> *Write sequence*. Verification process will read the device after writing and compare reading with the buffer. If verification pass than memory is successful programmer. Verification error may occur due to communication error, secured or broken chip.

4.2.3. Erasing

To erase memory, empty buffer can be loaded and programmed into a memory. In most cases this operation is not necessary, because programmer supports fast erase procedures (Page Erase, Block Erase and Chip Erase). Please click “ERASE” icon (memory chip with 0xFF and broom) for erasing operation.

Please note that erasing progress is not displayed at progress bar. For high density FLASH memories erase process may took quite long (even few minutes!).



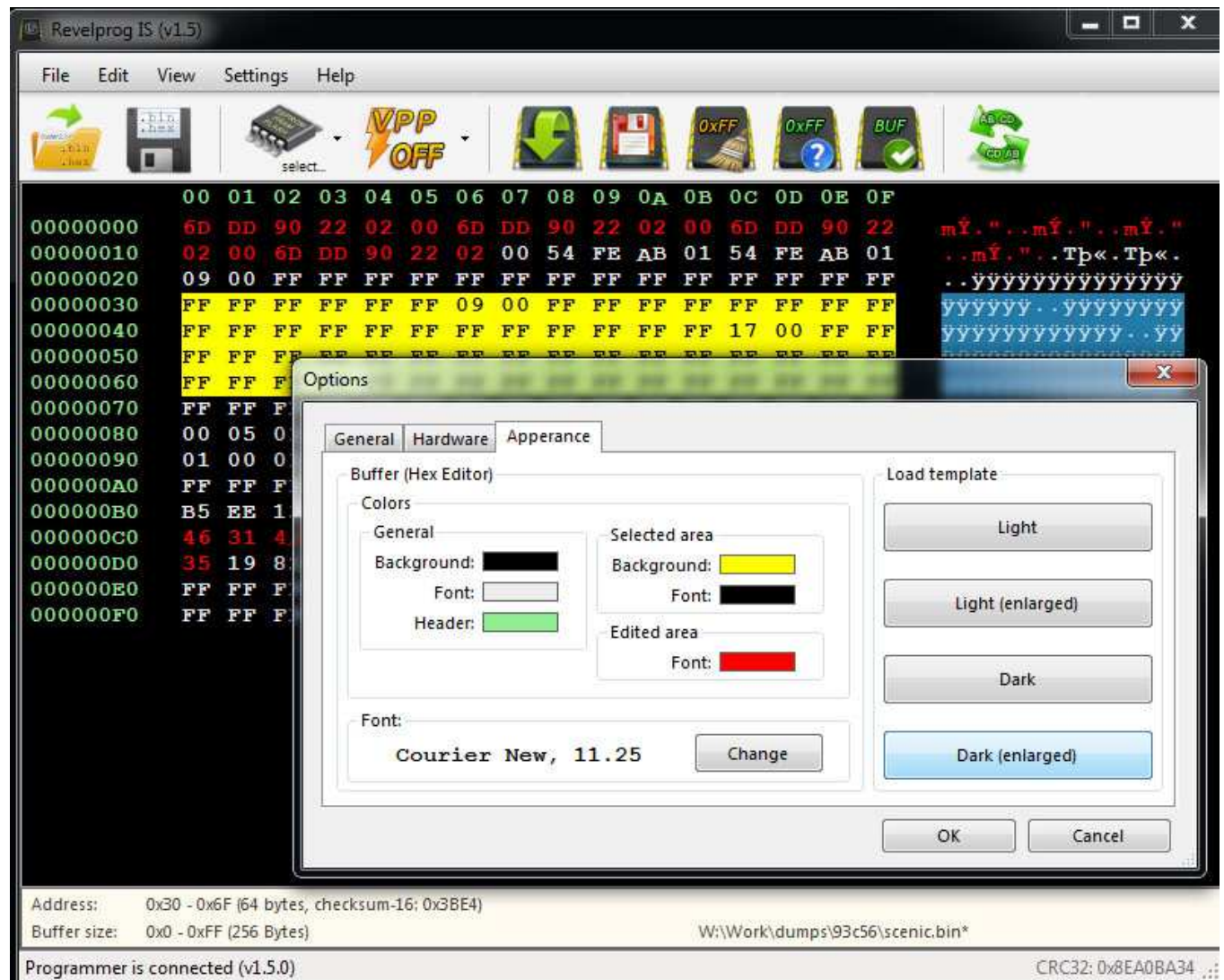
4.3. HEX editor features (buffer editing)

4.3.1. HEX editor

Buffer can be edited from application level. In the first column there is current address displayed. In the second column there is memory contents displayed (as bytes in hexadecimal mode). In the third column there is ASCII representation of memory contents. All buffer operations are allowed (e.g. CTRL+C, CTRL+V etc.). Modified buffer can be saved to external .bin or .hex file.

4.3.2. Appearance

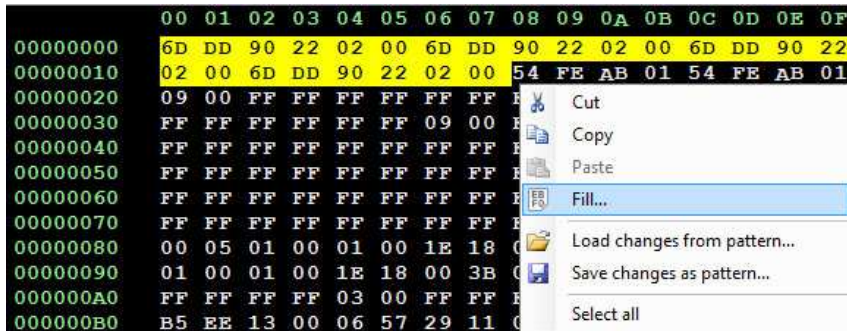
There is possibility to personalize hex editor, e.g. color and font changes. Bigger font and dark contrast colors will enhance your work with editor especially on high resolution displays.



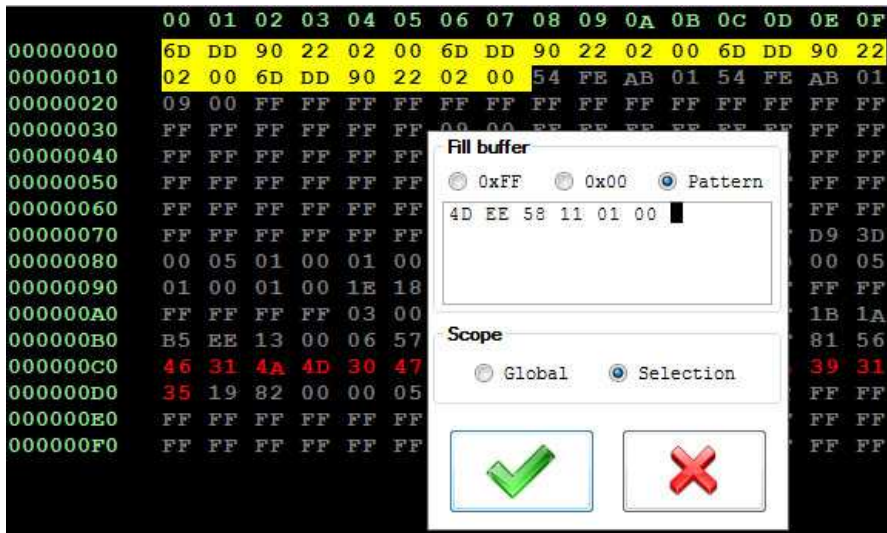
4.3.3. Fill buffer...

Fill selected bytes with constant bytes or defined pattern.

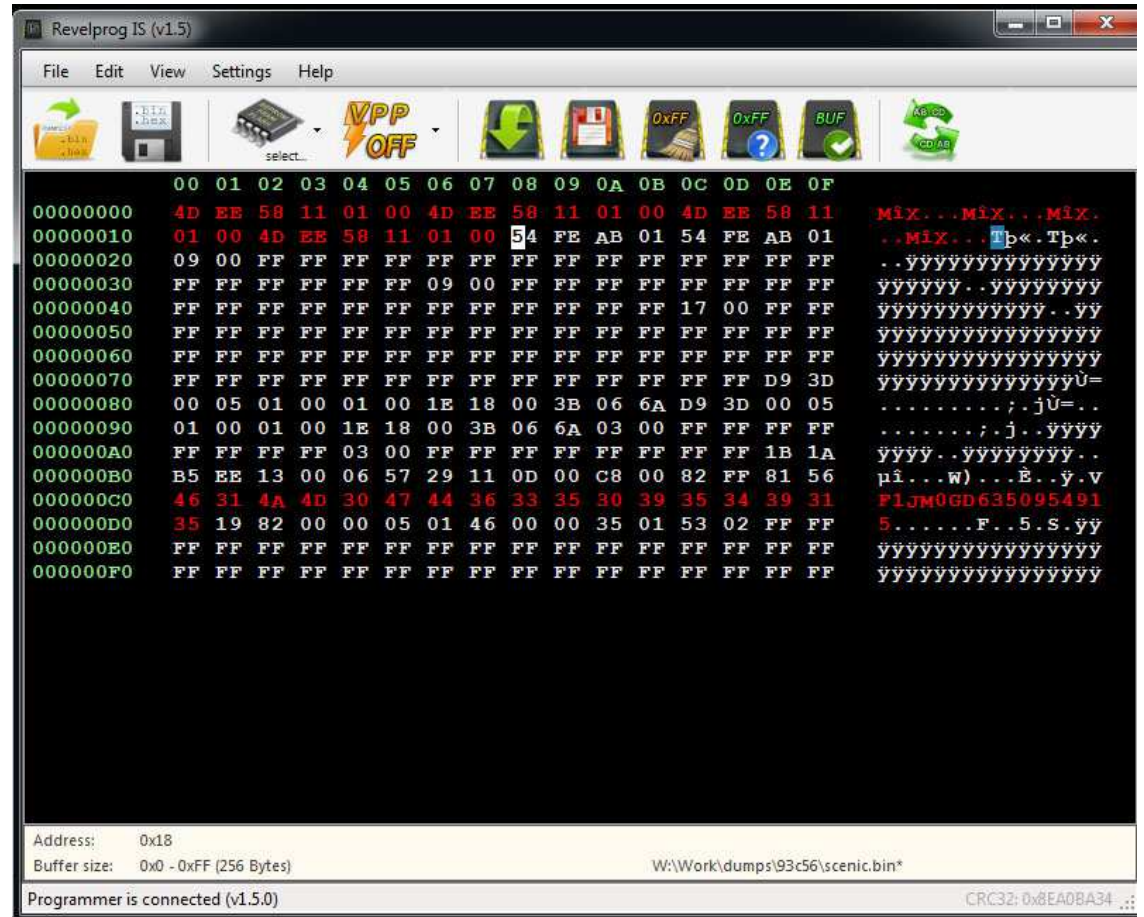
1. Open context menu (right click) and choose fill...



2. Enter pattern...

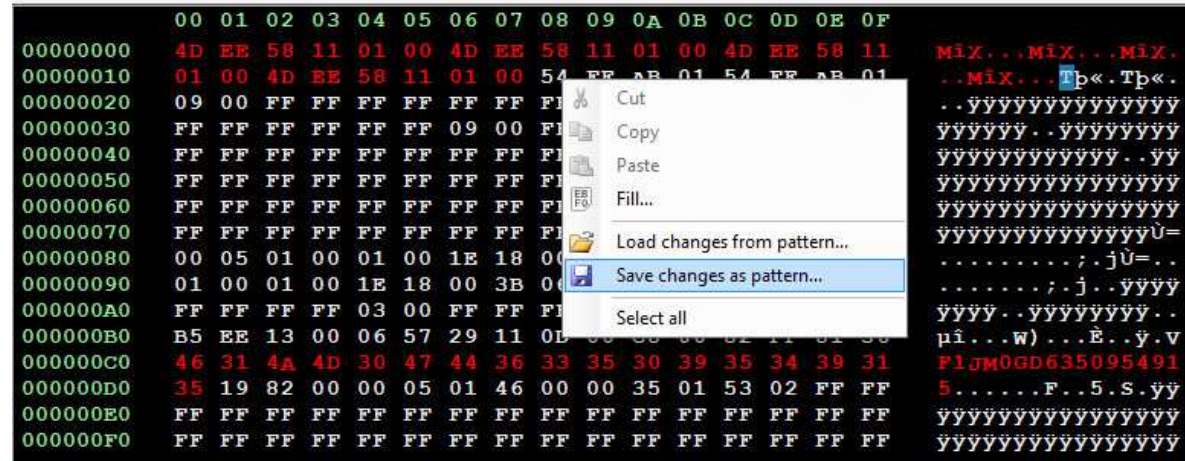


3. Take profit - you have just saved some time...



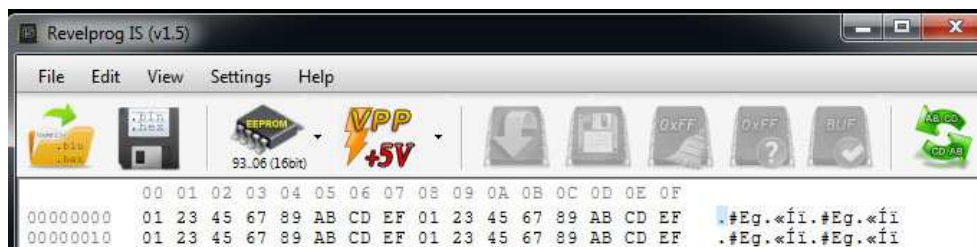
4.3.4. Patterns - export changes to file

All manual changes (highlighted bytes) can be exported to external file (pattern). Later, at any moment there is possibility to load pattern file and all changes will be imported, without modifying other bytes in the buffer. Simple, fast and powerful.



4.3.5. Swap bytes

Swap bytes operation will change byte order in 2-bytes word. For example, swapped word 0xABCD will be equal to 0xDCBA. This functionality is very useful for memories with two addressing modes, e.g. for microwire 93... series 16-bit chips. REVELPROG supports both addressing modes (8-bit and 16-bit) for these devices.

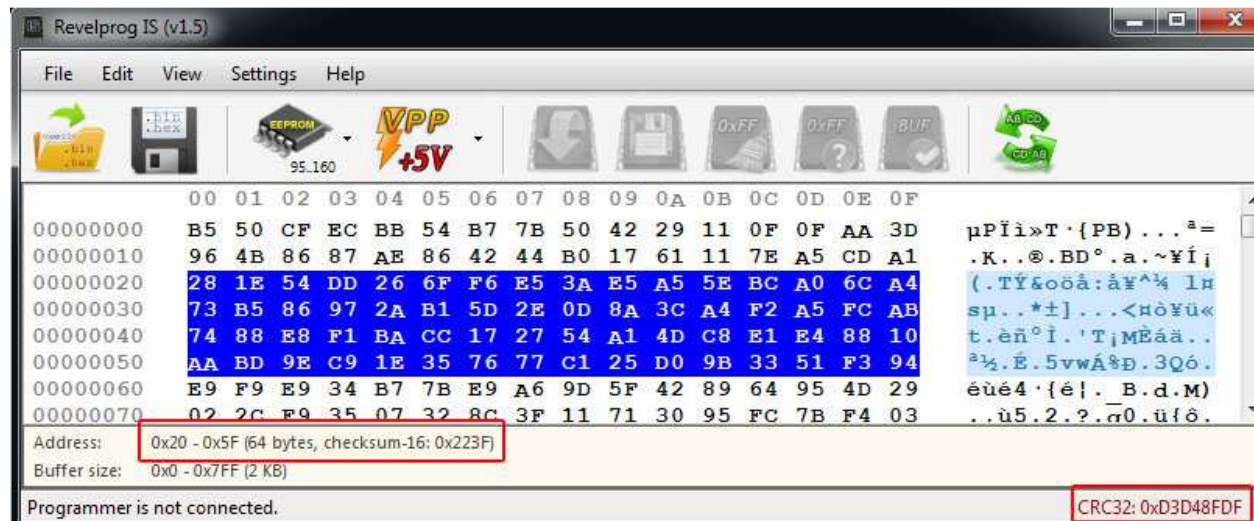


4.3.6. Clear buffer or generate random buffer

There is possibility to clear buffer (0xFF) or generate random buffer from *Edit* menu. Buffer length will be equal to memory size. *CTRL + N* shortcut will clear buffer and *CTRL + R* shortcut will generate random buffer.

4.3.7. CRC32 and Checksum-16

Click "CRC32" in right-bottom corner to calculate CRC32 checksum from the buffer. Changing single byte in the buffer will change CRC32 checksum so this feature is useful in comparison of sequential reads or files. Checksum-16 is calculated in real time from selected bytes in the buffer area and it's displayed in buffer info. This is most common algorithm for checksum used in wide range of devices (checksum of mileage, operation hours, serial numbers etc.) To calculate Checksum-16 from all addresses select all bytes in the buffer area (*CTRL+A*).



4.3.8. File format conversion: .bin <-> .hex

I Application can be used also as a simple file converter (binary to intel-hex and vice versa). To convert file from one type to another, just open file (load to buffer) and save it in another format.

4.3.9. DiffChecker (file comparing)

DiffChecker tool allows to compare 2 bin or hex files. All differences will be selected and grouped. Press F3/F2 to find next/previous difference region. You can save all differences to external file as a pattern (feature described in Chapter 4.3.4). If files are totally different then first 1000 different regions are marked.

The screenshot shows the DiffChecker application window. The title bar reads "DiffChecker". The main window is divided into two sections, each showing a comparison of a binary file with its ASCII representation. The first section is for "W:\Work\dumps\93c56\scenic1.bin" (256B (0xA48E)). The second section is for "W:\Work\dumps\93c56\scenic2.bin" (256B (0xA70B)).

In the first section, the hex dump shows differences at addresses 0x0000000A, 0x00000017, and 0x00000036. In the second section, differences are shown at addresses 0x0000000A, 0x00000017, and 0x00000032.

The status bar at the bottom of the window displays the message: "Files are different (3 different regions). Current address: 0xFF".

4.4. Additional features

4.4.1. Blank test

Blank test checks if chip is empty. Test will be canceled as soon as first programmed cells will be detected (there is no need to read full memory).

4.4.2. Compare with the buffer

Operation compares buffer with the contents of memory. Operation will be cancelled as soon as first difference will be detected.

4.4.3. VPP selection

VPP is set for both: programming voltage and signal levels. VPP can be changed with 0.1V step in range of 1.0 - 5.0 volts. This feature is very useful in few cases:

- in order to program low-voltage variant of the device (e.g. 1.8V instead of 3.3V),
- in case of in-circuit programming with external power supply -> programmer voltage should be the same as external circuit voltage or VPP should be disabled (VPP OFF),
- in case of in-circuit programming without external power supply. In many cases there is no need to set nominal voltage for external board. For example: memory operates in range of 2.6 to 3.6 volts. Nominal voltage for external board is 3.3V. Setting the lowest operational voltage (2.6V) will be enough to power and communicate with the memory and at the same time will be too low to power all circuits around. First of all, if there is main microprocessor on the board than it will not communicate with the external memory (because it will be in "reset" state due to low voltage to operate) so it will be possible to communicate with this memory using external programmer. Secondly, you will save current consumption because devices in reset state consume much less power. This trick is not always possible, so if you can't communicate with the device in external circuit try to plug-in external power supply or unsolder the chip and program it in socket. Please read Chapter 5 for more details.

4.4.4. Device identification (Chip ID) for FLASH SPI

In order to read Chip ID (only FLASH SPI 25x supports this function) please select "Tools" -> "Read ID". Operator should know operating voltage for the device because REVELPROG-IS supports also low voltage chips and reading chip ID without knowledge about chip power supply level is risky (setting 3,3V for 1,8V chip will damage it even during chip id reading). Programmer will read 4 identification bytes of data (JEDEC ID) – 1st byte is Manufacturer ID, 2nd and 3rd bytes are Device ID and 4th byte is Extended ID. You can read about identification bytes in device datasheet.

4.4.5. Quad SPI (QSPI) configuration for FLASH SPI

Some of FLASH SPI devices may require to enable QSPI mode before soldering it in external device. This will require to change one of configuration or status registers in device memory map.

TIP: in most cases, specially when you reprogram FLASH SPI there is not necessary to change any configuration bits.

To enable or disable Quad mode you can use dedicated tool (Tools -> Quad SPI) – it will work with most popular FLASH SPI devices (e.g. Winbond and Spansion). For other devices it may be necessary to write custom script (with Custom Script tool – described in Chapter 6) to enable or disable QSPI mode manually by changing QSPI bits in required registers. You can read how to change Quad SPI mode for your device in device datasheet.

4.4.6. Auto backup

Every read operation there is backup file (.bin) generated (by default it is user documents folder in system). By default maximum 16 files are created (oldest are deleted) and size is limited to 8Mbit devices. You can change these parameters in Settings -> Option -> File -> Autobackup (you can set unlimited number of files and size).

5. In-Circuit Serial Programming (ICSP)

Caution! Before ICSP programming please read following instructions!

ICSP programming is recommended only for advanced users. There is required technical knowledge and being familiar with specific chip. You should read carefully device datasheet, especially memory pinout, addressing modes and voltage levels.

Programmer is equipped with 10-pin ICSP socket (IDC10). There's only 8 pins used (pin-9 and pin-10 are not internally connected). Pinout numeration (1-8) is the same as memory pinout in datasheet for DIL8/SO8 package.

There will be various signals at these pins during chip programming (digital and analog signals). These signals are assigned automatically after memory chosen (in application) and are set only during memory operation.

5.1. Addressing in external circuit

All hardware addressing outputs for EEPROM 24.. series (pin 1,2,3 ie A0, A1, A2) must have low voltage level (GND). If chip in circuit is addressed in other way (e.g. there are few the same chips, which require various addressing) then communication will fail. In this case you can (1) set correct addressing in application ("External addressing" option during memory selection) or (2) set default "0" addressing with the programmer (connect addressing wires to the chip and in the same time disconnect them in external circuit). The safest option is to unsolder the chip and put it to integrated programmer socket.

Addressing 8/16-bit for EEPROM 93 series also needs special care. In circuit addressing wire (pin 6) can be connected to low or high state (it's defining addressing mode if memory support both modes). Depends on your chip selection in application (8/16 bit) programmer gives LOW or HIGH state on this pin. If you choose wrong addressing (e.g. in circuit it is low state and programmer outputs high state) than you will make short circuit. In most cases it's not necessary to connect addressing wire (pin 6) for microwire 93 series - it's enough to select right mode from application (16-bit in most cases).

If you are not 100% sure about connections, DO NOT program in circuit – unsolder the chip and program it with integrated socket.

5.2. Write protection in circuit (WP/PRE pin)

A lot of serial devices have WP (Write Protect) or PRE (Protection Enable) input. Low state on this pin (ground) protect the device from erasing and writing. Programmer outputs a high state during erase or write operation. If in external circuit write protect pin is connected to ground (common case), and programmer will output high state on this pin, than it will lead to short circuit. There are 3 common solutions in this case: (1) do not connect write protect pin (erase/write will not work), (2) disconnect write protect pin in circuit (connect it only to programmer) or (3) unsolder the chip and program it in programmer's socket.

WP (Write-Protect) or PRE (Protection Enable) pin can be found e.g. in FLASH SPI 25.. (pin 3), FLASH SPI 26... (pin 3), FLASH SPI 45... (pin 5), EEPROM I2C 24... (pin 7), EEPROM SPI 25.. (pin 3), EEPROM SPI 35... (pin 3), EEPROM SPI 95... (pin 3), EEPROM MICROWIRE 93... (pin 7).

5.2. Power supply in external circuit

The target application device can be either powered by programmer [voltage level (VPP) can be chosen from: 1.0V – 5.0V - please read Chapter 4.4.3] or powered by an application within a voltage range of 1.0V to 5.5V. If chip is already powered in external circuit, than VPP should be set to the same voltage as external power supply (0,1V accuracy is tolerated). Absolute maximum rating at programmer's I/O pins is 5.5V - exceed this value may damage protection buffers in programmer.

5.3. Limits and possible issues during in-circuit programming

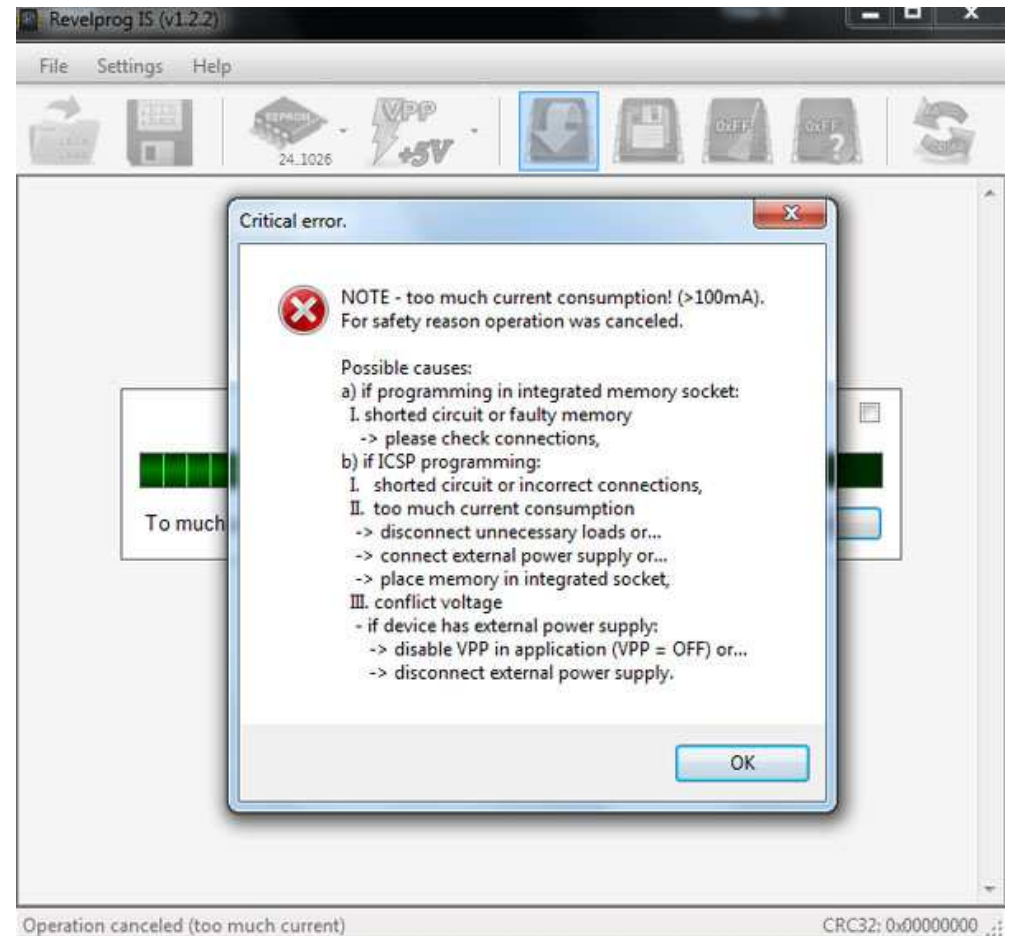
In some cases there is no possibility to program chip in application. This can be due by many reasons, such as:

- to long signal wires,
- too much load capacity in circuit,
- incorrect pull-ups or pull-downs on signal wires,
- incorrect addressing,
- bus conflict (memory is during communication with other device),
- impedance not match,
- additional capacitors, diodes bus-switches etc. in circuit.

In case of power supply from programmer, current is limited to 100mA for safety reason. If application requires more power, than external power supply should be connected.

In case of incorrect connections, short circuit or overload, following error (pict.) will be displayed and all I/O will be disconnected. Delay of protection trigger is 5-35ms so you still have to be careful. Programmer implements "VPP-boost" mode which allows for 3x higher current consumption at the first 30ms of operation (e.g. for capacitor charging). There is also possibility to increase in settings overload protection to 200mA.

If you are not sure about connections or voltage levels DO NOT program in-circuit. It is much safer to unsolder the chip and program it in the socket.



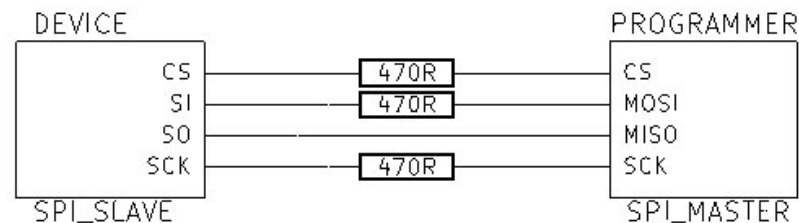
5.4. Troubleshooting for in-circuit programming

If you can program device in programmer's socket (or adapter) with success, but you cannot program it in circuit than there are few things that you can check. First of all, make sure that:

- all wires are connected properly (please take special care for pin contact in case of programming with SOIC CLIP),
- external circuit is powered properly and there is no voltage conflict on power bus,
- SPI lines: CS, (MOSI), (MISO) and (SCK) have pull-up resistors to power supply (4.7k - 10k),
- communication lines do not have additional elements, such as diodes, bus-switches, transistors etc.
- there is no communication conflict on the bus (e.g. processor and all other devices connected to shared SPI bus should be in RESET state).

Secondly, if problem still exist, there are still may be some tricky issues. Few examples with solutions below:

- noise and overshooting (communication errors, random read) – possible solutions:
 - decrease SPI clock to 10% for FLASH SPI devices (menu *Settings* -> *Options* -> Hardware -> *FLASH SPI SCK*),
 - impedance matching and termination for external circuit (and wires) by using e.g. termination resistors on signal wires. There is no universal value of resistance (each circuit is different) but in most cases 100 - 1000 Ohm resistors in serial for CS, (MOSI) and (SCK) will be enough:



- CS line decoupling (33pF - 220pF to ground as close as possible to the device in-circuit),
- to low voltage level during erasing and/or writing (erase error / communication error):
 - when external device (in external circuit) is powered from programmer, than there are also other devices powered too (on the same power line) and due to higher current consumption output voltage may be too low for erase and write operations. In this case it is recommended to set VPP slightly higher (e.g. +0.3V) or use external power source and set the same VPP level in application.

- bus conflict (random read or communication error during read):
 - in some cases after device power up, the main processor is starting communication with slave devices (e.g. with memory you are trying to read) – in this case it will be conflict on the bus (random read or communication error). All devices connected to the same communication bus should be forced to be in RESET state. Sometimes it is enough to use external power supply and wait about 2-3 minutes for initial procedure finished and then try to communicate with the memory (when all devices are in idle state and bus is free). The other tricky way is to use minimum voltage level to communicate with memory and this voltage level may be too low for other devices so they will stay in RESET state and will not interrupt communication. On the other hand, too low voltage level may be not enough to success for erase and write operations.

In some cases design of external circuit does not physically allow for in-circuit programming. In this case chip should be unsoldered and programmed in programmer's socket or adapter.

6. Custom Script Programming (I2C / SPI bus)

Custom script programming for I2C or SPI bus in case of REVELPROG is a some kind of "bus pirate" tool for serial communication. This tool is still in development phase but it is already powerful and functional. With REVELPROG-IS Customs Script tool **there is possibility to execute any commands in any sequence with precise delays on SPI or I2C bus**. You can write script to execute commands for unlocking device, read and write hidden (or not) registers, change sector protection bits, write OTP registers and much more.

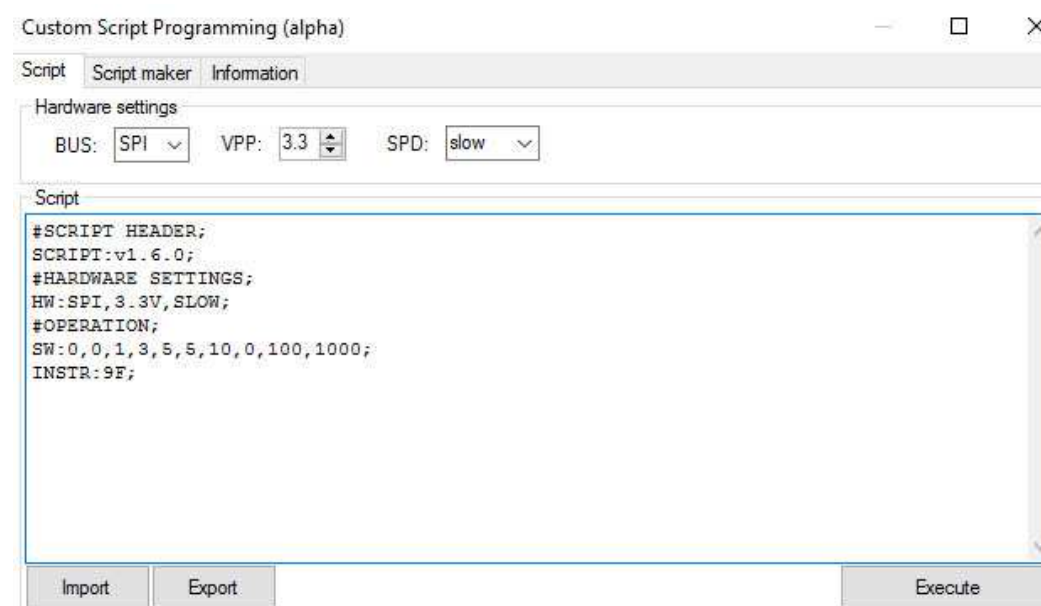
In practice it is also very useful tool for electronic enginners and embedded programmers, because **there is possibility to communicate with any serial (I2C or SPI) device**. External device or module can be also powered from REVELPROG-IS (1.0 - 5.0V with 0.1V threshold). Only few connection lines are required for communication (4 wires for SPI or 2 wires for I2C + optional 2 wires for external device power supply).

6.1. Script construction

Typical script consists of (1) #SCRIPT HEADER - information about software version, (2) #HARDWARE SETTINGS - voltage selection (VPP), bus selection (I2C/SPI) and communication speed, (3) #OPERATION (1 to n). Each script has to have at least one operation defined.

Each #OPERATION consists of (1) 'SW' operation parameters: number of bytes to read/write, delay settings, (2) 'INSTR' instruction - sequence of bytes for single instruction, (3) 'DATA' data to write (optional) - sequence of optional bytes for write after single instruction

Example construction of typical script with header (software version v.1.6.0), hardware settings (SPI bus, 3.3V voltage, slowest speed) and single operation with 1-byte 9Fh command is presented on right image:



6.2. Script syntax

- "#" defines sections, e.g. #SCRIPT HEADER; (header), #HARDWARE SETTINGS; (hardware settings), #OPERATION; (for each operation),
- ";" is placed at the end of each command,
- ":" is placed directly after command words, e.g. SCRIPT: (script version), HW: (hardware settings), SW: (software settings - operation parameters), INSTR: (instruction), DATA: (write data);
- "," separates parameters for each command (e.g. between "INSTR:" and ";" comma separates instruction bytes)
- "//" precedes comments which are ignored by script executor

Each script should have single #SCRIPT HEADER, single #HARDWARE SETTINGS and at least one #OPERATION (or more).

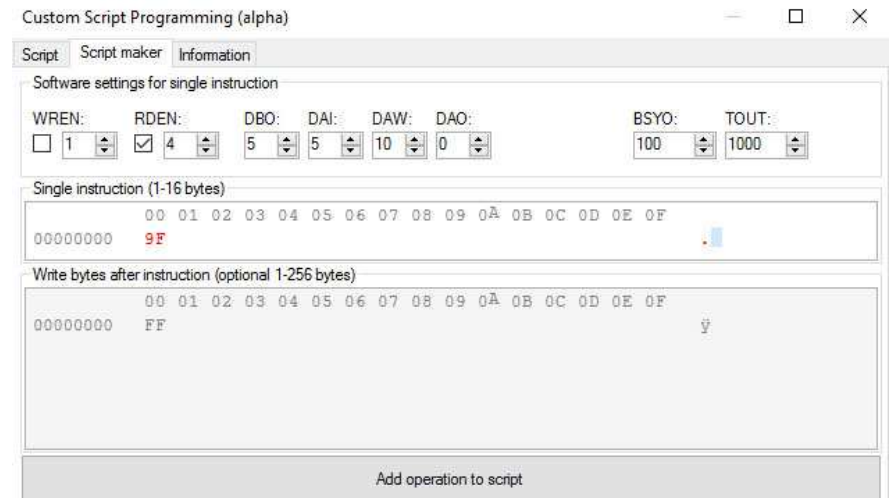
6.3. Script maker with GUI

Script maker will help you prepare multiple operations with correct syntax. You need to set few parameters for each operation:

- WREN (Write Enable) + number of bytes to write in single operation,
- RDEN (Read Enable) + number of bytes to read in single operation,
- DBO (Delay Before Operation) - optional delay in ms before operation,
- DAI (Delay After Instruction) - optional delay in ms after instruction,
- DAW (Delay After Write) - optional delay in ms after write,
- DAO (Delay After Operation) - optional delay in ms after operation,
- BSYO (Busy Operation) - maximum time in ms for write end (e.g. maximum wait time for busy flag in FLASH SPI or for ACK in I2C devices)
- TOUT (Timeout) - maximum PC timeout in ms for operation end (it is recommended to set timeout greater than sum of all other delays).

Please note that:

- Read/Write length are 0 based (0 means 1 byte, 255 means 256 bytes) and with WREN=0/RDEN=0 length parameter is ignored,
- "SW:" and "HW:" byte format is decimal,



- "INSTR:" and "DATA:" byte format is hexadecimal.

Operation code generated by script maker from above example::

```
#OPERATION;  
SW:0,0,1,3,5,5,10,0,100,1000;  
INSTR:9F;
```

As you can see SW parameters are equal to script maker parameters:

```
SW:WREN[0-1],WriteLength[0-255],RDEN[0-1],ReadLength[0-255],DBO[0-250],DAI[0-250],DAW[0-250],DAO[0-250],BSYO[0-n],TOUT[0-n];
```

6.4. Script execution sequence

Each script is executed with following sequence:

1. SET HW (BUS, VPP, SPEED)
2. EXECUTE OPERATION 1 (SET SW -> DBO -> INSTR -> DAI -> WRITE -> DAW -> BSYO -> READ -> DAO)
3. EXECUTE OPERATION 2
4. ...
5. EXECUTE OPERATION N
6. RESET HARDWARE

6.5. Script examples

Please take a look at forum tutorial for more information and script examples:

<https://forum.reveltronics.com/viewtopic.php?f=22&t=280>

7. Updates

New trends on the market and latest devices require from users newer and newer programming devices. Customer's satisfaction is our number one priority and we share updates free of charge for all users of REVELPROG-IS. Update consist of software update (pc application) and firmware update (software inside the device). Latest software can be downloaded directly from manufacturer's webpage. You will be informed by the software about possible updates after connecting out of date device.

Caution! If a firmware update process is interrupted, your programmer may not function properly. Updates are optional – warranty does not cover damage due improper care.

During update process, device will start in DFU (Device Firmware Upgrade) mode. First time it may be necessary to install additional drivers (for DFU mode). Windows should find drivers automatically on the Internet (if not, there are also available in `/drivers/DFU` subdirectory in the installation folder). Installing drivers process should take about 30 seconds. If DFU drivers are installed, application will update the firmware device. Appropriate instructions will be displayed before.

Please close all other applications and disable antivirus software during update process. Do not disconnect and do not connect any USB devices during update process. Make sure that you have stable power supply for PC.

Update process may take about half a minute. Progress will be displayed on the screen. There is no possibility to cancel this process! After successful update, device will restart and will be ready to use.



8. Research and development

We are keen to still develop and improve the device. Please contact us at lab@reveltronics.com or visit our contact page at www.reveltronics.com if you have any comments or tips about REVELPROG-IS.

Very welcome are ideas:

- how to improve REVELPROG-IS - what functionality add to it,
- what chips add to supported device list in the next update (please suggest only serial I2C, SPI, uWire devices).

Repeated suggestions from many users will have more chance for implementation in the future. We encourage you to contact us because in this way we already added many devices popular only in small segments of automotive market, computer service, RTV/AGD devices etc.

9. APPENDIX A - List of supported devices

The following list details the family of supported devices (left column) with selected examples of supported memories (right column). Most chips have the same programming procedures – programmer supports more devices than the examples given in the list.

8.1. EEPROM I2C 24... series

24..01 (1kbit)	24AA01 24AA014 24AA014H 24AA01H 24C01 24C01A 24C01B 24C01C 24LC01 24LC014 24LC014H 24LC01B 24LC01BH AF24BC01 AT24C01 AT24C01A AT24C01B BR24A01AF BR24A01AFJ BR24C01 BR24C01A BR24C01AF BR24C01AFJ BR24C01AFV BR24C21F BR24C21FJ BR24C21FV BR24G01 BR24G01F BR24G01FJ BR24G01FV BR24G01FVT BR24L01 BR24L01A BR24L01AF BR24L01AFJ BR24L01AFV BR24L01AFVM BR24L01AFVT BR24T01A CAT24AA01 CAT24C01 CAT24C01B CAT24C21 CAT24FC01 CAT24LC01 CAT24WC01 CR24C01 CY24C01 FM24C01 FT24C01 FT24C01A GE24C01 GP24BC01 HM24LC01 HT24C01 HT24LC01 IS24C01 IS24C01B LT24C01 LX24C01 M24C01 MM24C01 S-24C01A S-24C01B S-24C01C S-24CS01A SLA24C01 SLE24C01 ST24C01 ST24W01 TMC24A01 TU24C01B UG24C01 X24012 X24C01 X24C01A XL24C01A
24..02 (2kbit)	24AA02 24AA024 24AA024H 24AA025 24AA02H 24C02 24C02A 24C02B 24C02C 24LC02 24LC024 24LC024H 24LC025 24LC02B 24LC02BH 24LC02W 24LLC02 A24C02 ACE24C02 ACE24LC02 AF24BC02 AM24LC02 AT24C02 AT24C02A AT24C02B AT24C02C AT24HC02B AT24HC04B AX24C02A BJX25LC02 BL24C02 BR24A02F BR24A02FJ BR24A02FVM BR24C02 BR24C02F BR24C02FV BR24G02F BR24G02FJ BR24G02FVT BR24L02 BR24L02F BR24L02FJ BR24L02FV BR24L02FVM BR24L02FVT BR24T02 CAT24AA02 CAT24C02 CAT24C03 CAT24LC02 CAT24LC02A CAT24LC02AZ CAT24LC02Z CAT24WC02 CAT24WC03 CKD24A02 CR24C02 CW24C02 CW24C02B CY24C02 EC24C02 EC24C02A EM24LC02 FM24C02 FM24C02U FM24C02UF FM24C03U FM24C03UF FT24C02 FT24C02A GE24C02 GM24LC02B GP24BC02 HM24LC02 HN58X2402 HT24C02 HT24LC02 IN24AA02A IN24AA02B IN24LC02B IS24C02 IS24C02A IS24C02B IS24C02E K24C02 KK24LC02B L24C02 L24W02 LE24C0221 LE24C023 LE24L0221 LT24C02 LX24C02 LY24C02 M24C02 MAX24A02 MC24C02 MC24LC02 MM24C02 NM24C02 NM24C02F NM24C02U NM24C03 NM24C03F NM24W02 NSC24C02 PTK24A02T RT24A02 S-24C02A S-24C02B S-24C02C S-24CS02A SLA24C02 SLE24C02 SM24C02 ST24C02 ST24W02 T24C02 T24C02A TMC24A02 TU24C02B UG24C02 WS24C02 X24022 X24C02 XL24C02 Y24LC02
24..04 (4kbit)	24AA04 24AA04H 24C04 24C04A 24LC04 24LC04B 24LC04BH A24C04 ACE24C04 ACE24LC02 AF24BC04 AM24LC04 AT24C04 AT24C04A AT24C04B AX24C04A BJX24LC04 BL24C04 BR24A04F BR24A04FJ BR24C04 BR24C04F BR24C04FJ BR24C04FV BR24G04F BR24G04FJ BR24G04FV BR24G04FVT BR24L04 BR24L04F BR24L04FJ BR24L04FV BR24L04FVM BR24L04FVT BR24T04 CAT24C04 CAT24C05 CAT24LC04 CAT24WC04 CAT24WC05 CR24C04 CW24C04 CW24C04A CY24C04 EC24C04 EC24C04A FM24C04 FM24C04U FM24C04UF FM24C05U FM24C05UF FM24CL04 FT24C04 FT24C04A GE24C04 GM24LC04B HM24LC04 HN58X2404 HT24C04 HT24LC04 IN24LC04B IS24C04 IS24C04A IS24C04B K24C04 KK24LC04 KK24LC04B L24C04 L24W04 LE24C042 LE24C043 LT24C04 LX24C04 LY24C04 M24C04 M24LC04 MM24C04 NM24C04 NM24C04F NM24C04U NM24C05 NM24C05F NM24W04 S-24C04A S-24C04B S-24CS04A SLA24C04 SLE24C04 ST24C04 ST24W04 T24C04 T24C04A TMC24A04 TU24C04B UG24C04 WS24W04 X24042 X24C04 XL24C04 Y24LC04
24..08 (8kbit)	24AA08 24AA08H 24C08 24C08B 24LC08 24LC08B 24LC08BH 24LLC08 A24C08 ACE24C08 ACE24C08A ACE24LC08 ACE24LC08A AF24BC08 AM24LC08 AT24C08 AT24C08A AT24C08B AT24RF08C AX24C08A BL24C08 BR24A08F BR24A08FJ BR24C08 BR24C08F BR24C08FJ BR24C08FV

	BR24G08F BR24G08FJ BR24G08FV BR24G08FVT BR24L08 BR24L08F BR24L08FJ BR24L08FV BR24L08FVM BR24L08FVT BR24T08 CAT24C08 CAT24LC08 CAT24WC08 CR24C08 CW24C08 CW24C08B CY24C08 EC24C08 EC24C08A FM24C08 FM24C08U FM24C08UF FM24C09UF FM24CL08 FT24C08 FT24C08A GM24LC08B GP24BC08 HN58X2408 HT24C08 HT24LC08 IN24AA08B IN24LC08B IS24C08 IS24C08A K24C08 KK24LC08 KK24LC08B L24M08 L24W08 LE24C081 LE24C082 LE24L082 LT24C08 LX24C08 LY24C08 M24C08 MAX24A08 MC24C08 MC24LC08 MM24C08 MTV24C08 MTV24LC08 NM24C08 NM24C08F NM24C08U NM24C09 NM24C09F NM24C09U NM24W08 NSC24C08 PTK24A08T S-24C08A S-24C08C S-24CS08A SLA24C08 SLE24C08 ST24C08 ST24W08 T24C08 T24C08A TMC24A08 TU24C08B UG24C08 WS24W08 X24C08 XL24C08 Y24LC08
24..16 (16kbit)	24AA16 24AA16H 24C16 24C16B 24LC16 24LC164 24LC16B 24LC16BH 24LLC16 A24C16 ACE24C16 ACE24LC16 AF24BC16 AM24LC16 AT24C16 AT24C164 AT24C16A AT24C16B AT24C16C AX24C16A BJX24LC16 BL24C16 BR24A16F BR24A16FJ BR24C16 BR24C16F BR24C16FJ BR24C16FV BR24E16 BR24E16F BR24E16FJ BR24E16FV BR24G16F BR24G16FJ BR24G16FV BR24G16FVT BR24L16 BR24L16F BR24L16FJ BR24L16FV BR24L16FVT BR24S16F BR24S16FJ BR24S16FV BR24S16FVM BR24S16FVT BR24T16 CAT24C16 CAT24C16L CAT24C16Z CAT24FC16 CAT24FC17 CAT24WC16 CAT24WC17 CR24C16 CW24C16 CW24C16A CY24C16 EC24C16 EC24C16A FM24C16 FM24C16U FM24C16UF FM24C17 FM24C17U FM24C17UF FM24CL16 FT24C16 FT24C16A GM24LC16B GP24BC16 HM24LC16 HN58X2416 HT24C16 HT24LC16 IN24LC16B IS24C16 IS24C16A KK24LC16 KK24LC16B L24W16 LE24C162 LE24L162 LT24C16 LX24C16 LY24C16 M24C16 MAX24A16 MC24C16 MC24LC16 MM24C16 MTV24C16 MTV24LC16 NM24C16 NM24C16F NM24C16U NM24C17 NM24C17F NM24C17U NM24W16 NSC24C16 PTK24A16T S-24C16A S-24CS16A SLA24C16 SLE24C16 SLE24C164 ST24C16 ST24W16 T24C16 T24C16A TMC24A16 TU24C16B UG24C16 WS24W16 X24164 X24165 X24C16 XL24C16 XL24C164 Y24LC16
24..32 (32kbit)	24AA32 24AA32A 24C32 24LC32 24LC32A 24LC32B A24C32 ACE24C32 ACE24C32A AF24BC32 AT24C32 AT24C32A AT24C32B AT24C32C AT24C32D AX24C32A BR24A32F BR24A32FJ BR24C32 BR24C32F BR24G32F BR24G32FJ BR24G32FV BR24G32FVT BR24L32 BR24L32F BR24L32FJ BR24L32FV BR24L32FVT BR24S32F BR24S32FJ BR24S32FV BR24S32FVM BR24S32FVT BR24T32 CAT24C32 CAT24FC32 CAT24FC32A CAT24WC32 CAT24WC33 CR24C32 CW24C32 EC24C32 EC24C32A FM24C32 FM24C32U FM24C32UF FT24C32 FT24C32A HM24LC32 HN58X2432 HT24C32 HT24LC32 IN24AA32A IS24C32 IS24C32A IS24C32B IS24C32C K24C32 L24W32 LE24C322 LE24L322 M24C32 MC24C32 MM24C32 NM24C32 NM24C32U NM24C32UF NSC24C32 PTK24A32T S-24C32C SLA24C32 SLE24C32 ST24C32 T24C32 T24C32A TU24C32 WS24W32 X24320
24..64 (64kbit)	24AA64 24AA64F 24AA65 24C64 24C65 24FC64 24LC64 24LC64B 24LC64F 24LC65 A24C64 ACE24C64 ACE24C64A AF24BC64 AT24C64 AT24C64A AT24C64B AT24C64C AT24C64D AX24C64A BR24A64F BR24A64FJ BR24C64 BR24C64F BR24G64F BR24G64FJ BR24G64FV BR24G64FVT BR24L64 BR24L64F BR24L64FJ BR24S64F BR24S64FJ BR24S64FV BR24S64FVM BR24S64FVT BR24T64 CAT24C64 CAT24FC64 CAT24FC65 CAT24FC66 CAT24WC64 CAT24WC65 CR24C64 CW24C64 EC24C64 EC24C64A FM24C64 FM24C64F FM24CL64 FT24C64 FT24C64A HM24LC64 HN58X2464 HT24C64 HT24LC64 IS24C64 IS24C64A IS24C64B K24C64 L24W64 LE24CB642 M24C64 MM24C64 NM24C65 NM24C65F NSC24C64 PTK24A64T S-24C64C SLA24C64 SLE24C64 ST24C64 T24C64 T24C64A TU24C64B WS24W64 X24640 X24641 X24645
24..128 (128kbit)	24AA128 24C128 24FC128 24LC128 24LC21 24LC21A 24LCS21A ACE24C128 AF24BC128 AT24C128 AT24C128B AT24C128C AT24CS128 AX24C128A BL24C128 BR24G128F BR24G128FJ BR24G128FV BR24G128FVT BR24S128F BR24S128FJ BR24S128FV BR24S128FVT BR24T128 CAT24C128 CAT24WC128 CAT24WC129 CR24C128 CW24C128 EC24C128 EC24C128A FM24C128 FM24C128F FM24CL128 FT24C128 FT24C128A HN58X24128 HT24C128 HT24LC128 IS24C128 IS24C128A IS24C128B IS24L128 LE24CB1283 M24128 S-24C128C T24C128 T24C128A TU24C128
24..256 (256kbit)	24AA256 24C256 24FC256 24LC256 A24C256 ACE24C256 AT24C256 AT24C256B AT24C256C AT24CS256 AX24C256A BL24C256 BR24S256F BR24S256FJ BR24T256 CAT24C256 CAT24FC256 CAT24WC256 CAT24WC257 CW24C256 EC24C256A FM24C256 FM24C256F FM24CL256 FT24C256

	FT24C256A HN58X24256 HT24C256 HT24LC256 IS24C256 IS24C256A IS24L256 M24256 MC24C256 T24C256 T24C256A TU24C256
24..512 (512kbit)	24AA512 24C512 24FC512 24LC512 AT24C512 AT24C512B AT24C512C CAT24C512 CAT24WC512 FM24C512 FM24CL512 FT24C512A HN58X24512 HT24C512 HT24LC512 IS24C512 M24512 SA24C512
24..1024 (1Mbit)	24AA1024 24C1024 24FC1024 24LC1024 AT24C1024 AT24C1024B CAT24C1024 CAT24WC1024 FM24C1024 FM24CL1024 HT24C1024 HT24LC1024 IS24C1024
24..1025 (1Mbit)	24AA1025 24C1025 24LC1025
24..1026 (1Mbit)	24AA1026 24C1026 24LC1026
Other	Philips: PCF8582C-2

8.2. EEPROM SPI 25... series

25..010 (1kbit)	25010 25AA010 25AA010A 25C010 25LC010 25LC010A A25010A AT25010 BR25H010-W BR25L010-W CAT25C01 CAT25C11 CY25C01 IS25C01 S-25A010A ST25C01 ST25W01 XC25010
25..020 (2kbit)	25020 25AA020 25AA020A 25C020 25LC020 25LC020A AT25020 AT25020A BR25H020-W BR25L020-W CAT25C02 CAT25C03 CY25C02 FM25C020U FM25C020UL IS25C02 NM25C020 S-25A020A ST25C02 ST25W02 X25020 XC25020
25..040 (4kbit)	25040 25AA040 25AA040A 25C040 25LC040 25LC040A AT25040 AT25040ACAT25C04 BR25H040-W BR25L040-W CAT25C05 CY25C04 FM25C040U FM25C040UL FM25C041U FM25L04 IN25AA040 IS25C04 NM24C040 NM24C041 S-25A040A ST25C04 ST25W04 X25040 X25041 XC25040
25..080 (8kbit)	25080 25AA080 25AA080A 25AA080B 25C080 25LC080 25LC080A 25LC080B AT25080 AT25080A AT25080B BR25H080-W BR25L080-W CAT25C08 CAT25C09 CY25C08 FM25L08 IN25AA080 IS25C08 S-25A080A ST25C08 ST25W08 X25080 X25F008 XC25080
25..160 (16kbit)	25160 25AA160 25AA160A 25AA160B 25C160 25LC160 25LC160A 25LC160B AT25160 AT25160A AT25160B BR25H160-W BR25L160-W CAT25C16 CAT25C17 CY25C16 FM25C160 FM25C160U FM25C160UL FM25L016 IN25AA160 IS25C16 NM24C160 S-25A160A ST25C16 ST25W16 X25160 X25170 X25F016 XC25160
25..320 (32kbit)	25320 25AA320 25AA320A 25C320 25LC320 25LC320A AT25320 AT25320A AT25320B BR25H320-W BR25L320-W CAT25C32 CAT25C33 FM25C320U FM25C320UL IS25C32 IS25C32A IS25C32B S-25A320A TI2532 X25320 X25F032 XC25320
25..640 (64kbit)	25640 25AA640 25C640 25LC640 AT25640 AT25640A AT25640B BR25L640-W CAT25C64 CAT25C65 FM25C640U FM25CL64 FM25L64 IS25C64 IS25C64A IS25C64B NM24C640 S-25A640A TI2564 X25640 X25642 X25650 X25F064 XC25640
25..128 (128kbit)	25128 25AA128 25C128 25LC128 AT25128 AT25128A CAT25C128 IS25C128 IS25C128A S-25C128A X25128 X25138 X25F128 XC25128
25..256 (256kbit)	25256 25AA256 25C256 25LC256 AT25256 AT25256A CAT25C256 FM25L256 IS25C256 IS25C256A X25256 XC25256

25..512 (512kbit)	25512 25AA512 25C512 25LC512 AT25512 XC25512
25..1024 (1Mbit)	251024 25AA1024 25C1024 25LC1024 AT251024 XC251024

8.3. EEPROM SPI 35... series

NOTE: Incremental register (first 32 bytes) can be written only with higher value.

35..080 (8kbit)	D80D0WQ D80D0WQ M35080 M35080-3 M35080-6 M35080V6 M35080-VP
35..160 (16kbit)	D160D0WQ D160D0WT D160D0WQ D160D0WT M35160

8.4. EEPROM SPI 95... series

95..010, 95..020 95..040, 95..080 95..160, 95..320 95..640, 95..128 95..256, 95..512 95..10, 95..20 (1kbit – 2Mbit)	M95010, M95020, M95040, M95080, M95160, M95320, M95640, M95128, M95256, M95512, M95M01, M95M02
95P08 (8kbit)	5P08 5P08C3 95P08

8.5. EEPROM MICROWIRE 93... series

Supported both addressing (8-bit and 16-bit) modes.

93..06 (256bit)	9306 93C06 KM93C06 KM93C06GD M93C06 NMC9306 NMC93C06L NMC93C06LZ
93..46 (1kbit)	93AA46 93AA46A 93AA46B 93AA46C 93C46 93C46A 93C46B 93C46C 93CX46 93LC46 93LC46A 93LC46B 93LC46C 93X46 A93C46(x16) A93C46(x8) AF93BC46(x16) AF93BC46(x8) AK93C46 AM93LC46 AT93C46(x16) AT93C46(x8) AT93C46A AT93C46B AT93C46C AT93C46D(x16) AT93C46D(x8) AT93C46E BL93C46(x16) BL93C46(x8) BR93A46 BR93C46 BR93H76 BR93L46 BR93LC46 CAT93C46(x16) CAT93C46(x8) CAT93C46A CAT93C46AH CAT93C46B(x16) CAT93C46B(x8) CAT93C46H(x16) CAT93C46H(x8) CAT93C46R(x8) CAT93HC46(x16) CAT93HC46(x8) CR93C46(x16) CR93C46(x8) EC93C46A(x16) EC93C46A(x8) EM93LC46(x16) EM93LC46(x8) FM93C46 FM93C46A(x16) FM93C46A(x8) FT93C46(x16) FT93C46(x8) FT93C46A(x16) FT93C46A(x8) GSC93BC46A(x16) GSC93BC46A(x8) GT93C46(x16) GT93C46(x8) GT93C46A(x16) GT93C46A(x8) HT93LC46-A(x16) HT93LC46-A(x8) IN93AA46A IN93AA46B IN93AA46C(x16) IN93AA46C(x8) IN93LC46A IN93LC46B IN93LC46C(x16) IN93LC46C(x8) IS93C46-3 IS93C46A(x16) IS93C46A(x8) IS93C46AGR(x16) IS93C46AGR(x8) IS93C46B IS93C46BGR IS93C46D(x16) IS93C46D(x8) IS93C46DGR(x16) IS93C46DGR(x8) K93C46 KM93C46 KM93C46GD KM93C46V KM93C46VGD LC46 M93C46(x16) M93C46(x8) MAX93LC46 MC93C46 MM93C46-3 MM93C46-3GR NM93C46 NM93C46A NM93C46A(x16) NM93C46A(x8) NM93C46L NM93C46LZ NMC9346 PTK93LC46(x16) PTK93LC46(x8) RH76 RT93LC46(x16) RT93LC46(x8) S93462 S93463 S-93A46A S-93A46B S-93L46AD S93VP462 S93VP463 S93WD462 S93WD463 T93C46(x16) T93C46(x8) T93C46A(x16) T93C46A(x8) TMC93LC46 TMC93LC46(x16) TMC93LC46(x8) TU93C46(x16) TU93C46(x8) UG93C46-A(x16) UG93C46-A(x8) W93C46 W93C46B(x16) W93C46B(x8) XL93C46 XL93LC46 XL93LC46A XL93LC46ARY XL93LC46B(x16) XL93LC46B(x8) XL93LC46BRY(x16) XL93LC46BRY(x8) XL93LC46RY Y93LC46(x16)
93..56 (2kbit)	93AA56 93AA56A 93AA56B 93AA56C 93C56 93C56A 93C56B 93C56C 93CX56 93LC56 93LC56A 93LC56B 93LC56C 93X56 A93C56(x16) A93C56(x8) AF93BC56(x16) AF93BC56(x8) AK93C56 AM93LC56 AT93C56(x16) AT93C56(x8) AT93C56A AT93C56B AT93C56C AT93C56D(x16) AT93C56D(x8) AT93C56E AT93C57(x16) AT93C57(x8) BL93C56(x16) BL93C56(x8) BR93A56 BR93C56 BR93L56 BR93LC56 CAT93C56(x16) CAT93C56(x8) CAT93C56A CAT93C56AH CAT93C56B(x16) CAT93C56B(x8) CAT93C56H(x16) CAT93C56H(x8) CAT93C56R(x8) CAT93C57(x16) CAT93C57(x8) CAT93HC56(x16) CAT93HC56(x8) CR93C56(x16) CR93C56(x8) EC93C56A(x16) EC93C56A(x8) EM93LC56(x16) EM93LC56(x8) EM93LC57(x16) EM93LC57(x8) FM93C56 FM93C56A(x16) FM93C56A(x8) FT93C56(x16) FT93C56(x8) FT93C56A(x16) FT93C56A(x8) GSC93BC56A(x16) GSC93BC56A(x8) GT93C56(x16) GT93C56(x8) GT93C56A(x16) GT93C56A(x8) HT93LC56-A(x16) HT93LC56-A(x8) IN93AA56A IN93AA56B IN93AA56C(x16) IN93AA56C(x8) IN93LC56A IN93LC56B IN93LC56C(x16) IN93LC56C(x8) IS93C56-3 IS93C56A(x16) IS93C56A(x8) IS93C56AGR(x16) IS93C56AGR(x8) IS93C56B IS93C56BGR IS93C56D(x16) IS93C56D(x8) IS93C56DGR(x16) IS93C56DGR(x8) K93C56 KM93C56 KM93C56GD KM93C56V KM93C56VGD KM93C57(x16) KM93C57GD(x16) KM93C57V(x16) M93C56(x16) M93C56(x8) MC93C56 MM93C56-3 MM93C56-3GR NM93C56 NM93C56A NM93C56A(x16) NM93C56A(x8) NM93C56L NM93C56LZ PTK93LC56(x16) PTK93LC56(x8) S-93A56A S-93A56B S-93L56AD T93C56(x16) T93C56(x8) T93C56A(x16) T93C56A(x8) TMC93LC56 TMC93LC56(x16) TMC93LC56(x8) TMC93LC57(x16) TMC93LC57(x8) TU93C56(x16) TU93C56(x8) UG93C56-A(x16) UG93C56-A(x8) UG93C56-C W93C56 W93C56B(x16) W93C56B(x8) XL93C56 XL93LC56 XL93LC56A XL93LC56ARY XL93LC56B(x16) XL93LC56B(x8) XL93LC56BRY(x16) XL93LC56BRY(x8) XL93LC56RY Y93LC56-A(x16)

<p>93..66 (4kbit)</p>	<p>93AA66 93AA66A 93AA66B 93AA66C 93C66 93C66A 93C66B 93C66C 93CX66 93LC66 93LC66A 93LC66B 93LC66C 93X66 A93C66(x16) A93C66(x8) AF93BC66(x16) AF93BC66(x8) AK93C66 AM93LC66 AT93C66(x16) AT93C66(x8) AT93C66A AT93C66B AT93C66C AT93C66D(x16) AT93C66D(x8) AT93C66E BL93C66(x16) BL93C66(x8) BR93A66 BR93C66 BR93L66 BR93LC66 CAT93C66(x16) CAT93C66(x8) CAT93C66A CAT93C66AH CAT93C66B(x16) CAT93C66B(x8) CAT93C66H(x16) CAT93C66H(x8) CAT93C66R(x8) CAT93HC66(x16) CAT93HC66(x8) CR93C66(x16) CR93C66(x8) EC93C66A(x16) EC93C66A(x8) EM93LC66(x16) EM93LC66(x8) FM93C66 FM93C66A(x16) FM93C66A(x8) FT93C66(x16) FT93C66(x8) FT93C66A(x16) FT93C66A(x8) GSC93BC66A(x16) GSC93BC66A(x8) GT93C66(x16) GT93C66(x8) GT93C66A(x16) GT93C66A(x8) HT93LC66-A(x16) HT93LC66-A(x8) IN93AA66A IN93AA66B IN93AA66C(x16) IN93AA66C(x8) IN93LC66A IN93LC66B IN93LC66C(x16) IN93LC66C(x8) IS93C66-3 IS93C66A(x16) IS93C66A(x8) IS93C66AGR(x16) IS93C66AGR(x8) IS93C66B IS93C66BGR IS93C66D(x16) IS93C66D(x8) IS93C66DGR(x16) IS93C66DGR(x8) K93C66 KM93C66 KM93C66GD KM93C66V KM93C66VGD KM93C67(x16) KM93C67GD(x16) KM93C67V(x16) M93C66(x16) M93C66(x8) MC93C66 MM93C66-3 MM93C66-3GR NM93C66 NM93C66A NM93C66A(x16) NM93C66A(x8) NM93C66L NM93C66LZ PTK93LC66(x16) PTK93LC66(x8) S93662 S93663 S-93A66A S-93A66B S-93L66AD S93VP662 S93VP663 S93WD662 S93WD663 T93C66(x16) T93C66(x8) T93C66A(x16) T93C66A(x8) TMC93LC66 TMC93LC66(x16) TMC93LC66(x8) TU93C66(x16) TU93C66(x8) UG93C66-A(x16) UG93C66-A(x8) W93C66 W93C66B(x16) W93C66B(x8) XL93C66 XL93LC66 XL93LC66A XL93LC66ARY XL93LC66B(x16) XL93LC66B(x8) XL93LC66BRY(x16) XL93LC66BRY(x8) XL93LC66RY Y93LC66-A(x16)</p>
<p>93..76 (8kbit)</p>	<p>93AA76 93AA76A 93AA76B 93AA76C 93C76 93C76A 93C76B 93C76C 93CX76 93LC76 93LC76A 93LC76B 93LC76C 93X76 A93C76(x16) A93C76(x8) AF93BC76(x16) AF93BC76(x8) AK93C76 AM93LC76 AT93C76(x16) AT93C76(x8) AT93C76A AT93C76B AT93C76C AT93C76D(x16) AT93C76D(x8) AT93C76E BL93C76(x16) BL93C76(x8) BR93A76 BR93C76 BR93L76 BR93LC76 CAT93C76(x16) CAT93C76(x8) CAT93C76A CAT93C76AH CAT93C76B(x16) CAT93C76B(x8) CAT93C76H(x16) CAT93C76H(x8) CAT93C76R(x8) CAT93HC76(x16) CAT93HC76(x8) CR93C76(x16) CR93C76(x8) EC93C76A(x16) EC93C76A(x8) EM93LC76(x16) EM93LC76(x8) FM93C76 FM93C76A(x16) FM93C76A(x8) FT93C76(x16) FT93C76(x8) FT93C76A(x16) FT93C76A(x8) GSC93BC76A(x16) GSC93BC76A(x8) GT93C76(x16) GT93C76(x8) GT93C76A(x16) GT93C76A(x8) HT93LC76-A(x16) HT93LC76-A(x8) IN93AA76A IN93AA76B IN93AA76C(x16) IN93AA76C(x8) IN93LC76A IN93LC76B IN93LC76C(x16) IN93LC76C(x8) IS93C76-3 IS93C76A(x16) IS93C76A(x8) IS93C76AGR(x16) IS93C76AGR(x8) IS93C76B IS93C76BGR IS93C76D(x16) IS93C76D(x8) IS93C76DGR(x16) IS93C76DGR(x8) K93C76 KM93C76 KM93C76GD KM93C76V KM93C76VGD M93C76(x16) M93C76(x8) MC93C76 MM93C76-3 MM93C76-3GR NM93C76 NM93C76A NM93C76A(x16) NM93C76A(x8) NM93C76L NM93C76LZ PTK93LC76(x16) PTK93LC76(x8) S-93A76A S-93A76B S-93L76AD T93C76(x16) T93C76(x8) T93C76A(x16) T93C76A(x8) TMC93LC76 TMC93LC76(x16) TMC93LC76(x8) W93C76 W93C76B(x16) W93C76B(x8) XL93C76 XL93LC76 XL93LC76A XL93LC76ARY XL93LC76B(x16) XL93LC76B(x8) XL93LC76BRY(x16) XL93LC76BRY(x8) XL93LC76RY</p>
<p>93..86 (16kbit)</p>	<p>93AA86 93AA86A 93AA86B 93AA86C 93C86 93C86A 93C86B 93C86C 93CX86 93LC86 93LC86A 93LC86B 93LC86C 93X86 A93C86(x16) A93C86(x8) AF93BC86(x16) AF93BC86(x8) AK93C86 AM93LC86 AT93C86(x16) AT93C86(x8) AT93C86A AT93C86B AT93C86C AT93C86D(x16) AT93C86D(x8) AT93C86E BL93C86(x16) BL93C86(x8) BR93A86 BR93C86 BR93L86 BR93LC86 CAT93C86(x16) CAT93C86(x8) CAT93C86A CAT93C86AH CAT93C86B(x16) CAT93C86B(x8) CAT93C86H(x16) CAT93C86H(x8) CAT93C86R(x8) CAT93HC86(x16) CAT93HC86(x8) CR93C86(x16) CR93C86(x8) EC93C86A(x16) EC93C86A(x8) EM93LC86(x16) EM93LC86(x8) FM93C86 FM93C86A(x16) FM93C86A(x8) FT93C86(x16) FT93C86(x8) FT93C86A(x16) FT93C86A(x8) GSC93BC86A(x16) GSC93BC86A(x8) GT93C86(x16) GT93C86(x8) GT93C86A(x16) GT93C86A(x8) HT93LC86-A(x16) HT93LC86-A(x8) IN93AA86A IN93AA86B IN93AA86C(x16) IN93AA86C(x8) IN93LC86A IN93LC86B IN93LC86C(x16) IN93LC86C(x8) IS93C86-3 IS93C86A(x16) IS93C86A(x8) IS93C86AGR(x16) IS93C86AGR(x8) IS93C86B IS93C86BGR IS93C86D(x16) IS93C86D(x8) IS93C86DGR(x16) </p>

	IS93C86DGR(x8) K93C86 KM93C86 KM93C86GD KM93C86V KM93C86VGD M93C86(x16) M93C86(x8) MC93C86 MM93C86-3 MM93C86-3GR NM93C86 NM93C86A NM93C86A(x16) NM93C86A(x8) NM93C86L NM93C86LZ PTK93LC86(x16) PTK93LC86(x8) S-93A86A S-93A86B S-93L86AD T93C86(x16) T93C86(x8) T93C86A(x16) T93C86A(x8) TMC93LC86 TMC93LC86(x16) TMC93LC86(x8) W93C86 W93C86B(x16) W93C86B(x8) XL93C86 XL93LC86 XL93LC86A XL93LC86ARY XL93LC86B(x16) XL93LC86B(x8) XL93LC86BRY(x16) XL93LC86BRY(x8) XL93LC86RY
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

93S46 (1kbit)	93S46 M93S46 NM93CS46
93S56 (2kbit)	93S56 M93S56 NM93CS56 ST93CS56 ST93CS57
93S66 (4kbit)	93S66 M93S66 NM93CS66

8.6. FLASH SPI 25... series

Supports low-voltage serial flash memories (1.5V, 1.8V, 2.5V) and standard (3.3V)

25..05 (512kbit)	A25L05P A25L512 A25L512A A25LS512A AC25LV512 AT25DL5121 AT25F512 AT26DF5121 EN25B05 EN25F05 EN25P05 F25D512Q F25L0512 F25L05P F25L05PA F25L512P F25L512Q GD25Q05 KH25L512 M25P05 M25P05-A MS25X512 MX25L512 MX25L5121 MX25L5121E MX25L512C MX25L512E MX25V512 MX25V5121E MX25V512C MX25V512E MX25V512F PM25LV512 PM25LV512A SA25F005 SST25VF512 SST25VF512A SST25WF512 SST25WF512B W25Q05CL W25Q05CV W25X05CL
25..010 (1Mbit)	A25L010 A25L010A A25L10P AC25LV010 AT24F1024 AT25FS010 EN24LF10 EN25B10 EN25F10 EN25LF10 EN25P10 EN25S10 ES25P10 F25L01P F25L01PA GD25Q10 IS25CD010 IS25LQ010B KH25L1005 LE25FU106 LE25FU106B M25P10 M25P10-A M25PE10 MS25X10 MX25L1005 MX25L1005C MX25L1005ZM MX25L1006 MX25L1006E MX25L1021 MX25L1021E MX25L1025 MX25L1025C MX25L1026 MX25L1026E MX25R1035F MX25U1001E MX25V1006 MX25V1006E MX25V1035F NX25P10 PM25LV010 PM25LV010A S25FL001D SA25F010 SST25VF010 SST25VF010A SST25WF010 TS25L010A W25D10V W25P10V W25Q10BV W25Q10CL W25Q10CV W25Q10EW W25X10AV W25X10BL W25X10BV W25X10CL W25X10L W25X10V
25..020 (2Mbit)	A25L020 A25L020C A25L20P AT25DF021 AT25F2048 EN25B20 EN25F20 EN25LF20 EN25P20 EN25S20 ES25P20 F25D02P F25D02QA F25L02P F25L02PA GD25Q20B GD25Q21B IS25LQ020 IS25WD020 IS25WP020D IS25WQ020 KH25L2005 LE25FU206 LE25S20 LE25U20 M25P20 M25PE20 M25U2033E MS25X20 MX25L2005 MX25L2005C MX25L2005ZM MX25L2006 MX25L2025C MX25L2026E MX25L26C MX25R2035F MX25U2032E MX25U2033E MX25U2035F MX25V2006E MX25V2033F MX25V2035F NX25P20 PM20LV020 S25FL002D SA25F020 SST25VF020 SST25VF020A SST25VF020B SST25WF020 SST25WF020B TS25L020A W25D20V W25P20V W25Q20BW W25Q20CL W25Q20CV W25Q20EW W25X20BL W25X20BV W25X20CL

<p>25..040 (4Mbit)</p>	<p>A25L040 A25L040A A25L40P AT25DF041 AT25F4096 AT25FS040 AT25SF041 AT26F004 EN25B40 EN25F40 EN25LF40 EN25P40 EN25Q40 EN25S40 EN25T40 ES25M40 ES25P40 F25D04Q F25D04QA F25L004 F25L04P F25L04PA GD25LQ40 GD25Q40B GD25Q41B IS25LD040 IS25LQ040B IS25WD040 IS25WP040 IS25WP040D IS25WQ040 KH25L4005A KH25L4006E LE25FU406B LE25S40 LE25U40 M25P40 M25PE40 MS25X40 MX25L4005 MX25L4005A MX25L4005C MX25L4006E MX25L4025C MX25L4026E MX25R4035F MX25U4032F MX25U4033E MX25U4035 MX25U4035F MX25V4005 MX25V4005C MX25V4006E MX25V4035 NX25P40 PM25LV040 S25FL004 S25FL004A S25FL004K S25FL204K SA25F040 SST25VF040 SST25VF040A SST25VF040B SST25WF040 SST25WF040B W25D40V W25P40V W25Q40BL W25Q40BV W25Q40BW W25Q40CL W25Q40EW W25X40AL W25X40AV W25X40BL W25X40BV W25X40CL W25X40L W25X40V</p>
<p>25..080 (8Mbit)</p>	<p>A25L080 A25L80P AT25DF081 AT25DF081A AT25DL081 AT25SF081 AT26DF081 EN25B80 EN25D80 EN25F80 EN25P80 EN25Q80 EN25Q80A EN25S80 EN25T80 ES25M80 ES25P80 F25D08QA F25L08QA FM25Q08 FM25Q08A GD25LQ80 GD25Q80B IS25LP080D IS25LQ080 IS25LQ080B IS25WP080 IS25WP080D KH25L8005 KH25L8006E KH25L8036D LE25S81 M25P80 M25PE80 M25PX80 M25X80 MS25X80 MX25L8005 MX25L8005ZM MX25L8006E MX25L8008E MX25L8035E MX25L8036E MX25L8073E MX25L8075E MX25R8035F MX25U8032E MX25U8033E MX25U8035 MX25U8035E MX25U8035F MX25V8005 MX25V8006E MX25V8035F NX25P80 PM25LV080 PM25LV080B S25FL008A S25FL008K S25FL208K SST25VF080B SST25WF080 SST25WF080B TS25L80P W25D80V W25P80V W25Q40CL W25Q80BL W25Q80BV W25Q80BW W25Q80DL W25Q80DV W25Q80EW W25Q80FV W25X80AL W25X80AV W25X80BV W25X80L W25X80V</p>
<p>25..016 (16Mbit)</p>	<p>A25L016 A25L16P A25LQ16 AT25DF161 AT25DL161 AT25SF161 AT26DF161 EN25B16 EN25D16 EN25F16 EN25P16 EN25Q16A EN25QA16 EN25QH16 EN25S16 EN25T16 ES25M16 ES25P16 F25L016 F25L16P F25L16PA F25L16Q F25L16QA FM25Q16 FM25Q16A FM25Q16B FM25S16A GD25LQ16 GD25Q16B IS25LP016D IS25LQ016B IS25WP016 IS25WP016D KH25L1605A KH25L1605D KH25L1606E KH25L1635D LE25S161 M25P16 M25PE16 M25PX16 M25X16 MS25X16 MX25L1605 MX25L1605A MX25L1605D MX25L1606E MX25L1608D MX25L1608E MX25L1633E MX25L1635D MX25L1635E MX25L1636D MX25L1636E MX25L1673E MX25L1675E MX25R1635F MX25U1633F MX25U1635E MX25U1635F MX25V1635F N25Q016 NX25P16 PM25LV016 PM25LV016B QB25F016S33B QH25F016S33B S25FL016 S25FL016A S25FL016K S25FL116K S25FL216K SA25F160 SST25VF016B TS25L16AP TS25L16BP TS25L16P W25Q16BF W25Q16BV W25Q16CL W25Q16CV W25Q16DV W25Q16DW W25Q16FW W25Q16JL W25Q16JV W25Q16VF W25X16AL W25X16AV W25X16BV W25X16V</p>
<p>25..032 (32Mbit)</p>	<p>A25L032 A25LQ32A AT25DF321 AT25DF321A AT25QL321 AT25SF321 AT25SL321 AT26DF321 AT26SD321 EN25B32 EN25F32 EN25P32 EN25Q32B EN25QA32 EN25QH32 EN25S32 ES25P32 F25L32P F25L32PA F25L32Q F25L32QA FM25Q32 FM25Q32A GD25LQ32 GD25LQ32C GD25Q32B GD25Q32C IS25CQ032 IS25CQ032A IS25LP032D IS25LQ032B IS25WP032 IS25WP032D KH25L3205D KH25L3206E M25P32 M25PE32 M25PX32 M25X32 MX25L3205A MX25L3205D MX25L3206 MX25L3206E MX25L3208D MX25L3208E MX25L3225D MX25L3233F MX25L3235D MX25L3235E MX25L3236D MX25L3236E MX25L3236F MX25L3237D MX25L3239E MX25L3255E MX25L3273E MX25L3273F MX25L3275E MX25R3235F MX25U3235E MX25U3235F MX25U3273F N25Q032 NX25P32 QB25F320S33B QH25F320S33B S25FL032 S25FL032A S25FL032K S25FL032P S25FL132K SA25F320 SST25VF032B W25Q32BV W25Q32DW W25Q32FV W25Q32FW W25Q32JV W25Q32V W25Q32VF W25X32AV W25X32BV W25X32V</p>
<p>25..064 (64Mbit)</p>	<p>A25LMQ64 A25LQ64 AT25DF641A AT25QF641 AT25QL641 AT25SF641 AT25SL641 EN25F64 EN25Q64 EN25QA64 EN25QH64 EN25S64 F25D64QA F25L64QA FM25Q64 FM25Q64A GD25B64 GD25LQ64B GD25LQ64C GD25Q64B GD25Q64C IS25LP064 IS25LP064A IS25WP064 IS25WP064A KH25L6405D KH25L6406E M25P64 M25PX64 MT25QL064A MT25QU064A MX25L64 MX25L6405 MX25L6405D MX25L6406E MX25L6408 MX25L6408D MX25L6408E MX25L6433 MX25L6433F MX25L6435 MX25L6435E MX25L6436E MX25L6436F MX25L6439 MX25L6439E MX25L6445E MX25L6455E MX25L6456 MX25L6456E MX25L6456F MX25L6465E MX25L6473E MX25L6473F MX25L6475E MX25R6435F MX25U6435 MX25U6435E MX25U6435F </p>

	MX25U6473 MX25U6473F N25Q064 N25Q064A11 N25Q064A13 QB25F640S33B QH25F640S33B S25FL064A S25FL064K S25FL064L S25FL064P S25FL164K S25FS064S SST25VF064C W25Q64BV W25Q64CV W25Q64DW W25Q64FV W25Q64FW W25Q64JV W25Q64VF W25X64BV W25X64V
25..128 (128Mbit)	AT25QL128A AT25SL128A EN25Q128 EN25QA128 EN25QH128 EN25QH128A F25D128QA F25L128QA GD25B128 GD25LQ128 GD25LQ128C GD25Q127 GD25Q128B GD25Q128C IS25LP128 IS25LP128F IS25WP128 IS25WP128F KH25L12845E M25P128 MT25QL128A MT25QU128A MX25L128 MX25L12805D MX25L12835E MX25L12835F MX25L12836E MX25L12839F MX25L12845E MX25L12845G MX25L12850F MX25L12855E MX25L12855F MX25L12865E MX25L12865F MX25L12873F MX25L12873G MX25L12875F MX25LM12845G MX25U128 MX25U12835F MX25U12873F MX25UM12845G N25Q128 N25Q128A11 N25Q128A13 Q128 S25FL127S S25FL128K S25FL128L S25FL128P S25FL128S S25FL129P S25FS128S W25Q128BV W25Q128FV W25Q128FW W25Q128JV W25Q128JW
25..256 (256Mbit)	EN25QA256 EN25QH256 EN25QH256A EN25S256 GD25LQ256 GD25Q256 IS25LP256D IS25WP256D MT25QL256A MT25QU256A MX25L25635E MX25L25635F MX25L25639F MX25L25645G MX25L25655E MX25L25655F MX25L25673G MX25L25735E MX25L25735F MX25L25835E MX25L25855E MX25LM25645G MX25U25635F MX25U25645G MX25UM25645G N25Q256A11 N25Q256A13 N25Q256A33 N25Q256A73 N25Q256A81 N25Q256A83 S25FL256L S25FL256P S25FL256S S25FS256S W25Q256FV W25Q256JV W25Q256JW W25Q257FV W25Q257JV
25..512 (512Mbit)	GD25Q512 IS25LP512M IS25WP512M MT25QL512A MT25QU512A MX25L51237G MX25L51245G MX25L51255G MX25LM51245G MX25U51245G MX25UM51245G MX66L51235F MX66U51235F S25FL512S S25FS512S

8.7. FLASH SPI 26... series

Supports low-voltage serial flash memories (1.8V) and standard (3.3V)

26.. (4 - 64MBit)	SST26WF040B SST26WF040BA SST26WF080B SST26WF080BA SST26VF016 SST26VF016B SST26WF016B SST26WF016BA SST26VF032 SST26VF032B SST26VF032BA SST26VF064B SST26VF064BA SST26VF064C SST26VF064CA SST26WF064C SST26WF064CA	SS
------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

8.7. FLASH SPI 45... series

45.. (1 - 64MBit)	Atmel / Adesto: AT45DB021 AT45DB021D AT45DB021E AT45DB041 AT45DB041D AT45DB041E AT45DB081 AT45DB081D AT45DB081E AT45DB161 AT45DB161D AT45DB161E AT45DQ161 AT45DB321 AT45DB321D AT45DB321E AT45DQ321 AT45DB642 AT45DB642D AT45DB641E	
------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

8.8. FRAM I2C 24... series

24..04	FM24V01
24..16	FM24V02 FM24W256
24..64	FM24C04B FM24CL04B
24..01	FM24V05
24..02	FM24V10 FM24VN10
24..05	FM24C16B FM24CL16B
24..10	FM24C64B FM24CL64B

8.9. FRAM SPI 25... series

25..04	FM25V01
25..16	FM25V02 FM25W256
25..64	FM25040B FM25L04B
25..01	FM25V05
25..02	FM25V10 FM25VN10
25..05	FM25C160B FM25L16B
25..10	FM25H20 FM25V20
25..20	FM25640B FM25CL64B

8.10. Optical and ethernet modules (SFP/SFP+ transceivers)

Please note that there is SFP adapter required with external power supply!

SFP / SFP+ (MSA standard: 0xA0, 0xA2)	SFP and SFP+, read/write block 0xA0 (256B), read block 0xA2 (256B), standard MSA Supports not secured chips - example manufacturers: GBC, HUAWEI, GPON, JDSU, MIKROTIK, OPTEC, OPTION, WTD and other OEM Possibility to change manufacturers info, model, serial number etc. and recalculate checksum directly from REVELPROG-IS application.
------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8.11. Your suggestions

If you have any suggestion for supported device list please contact us at lab@reveltronics.com. There is possibility to add any serial device with I2C, SPI or uWire interface (EEPROM/FLASH/FRAM).

10. **APPENDIX B - video tutorials and technical support**

Programming examples on youtube:

https://www.youtube.com/playlist?list=PLKdrjJPMLd8KBX12vHdyCg_0JfNT0Aw7U

REVELTRONICS' forum:

<http://www.reveltronics.com/forum/viewforum.php?f=10>