



UNO-01M-C105W54H256 SCPI Programming Manual

Overview

UNO synthesizer consists of two blocks: control PCB based on the microcontroller and RF-module based on three PLL loops. First PLL with very narrow band is used to clear external reference and to produce high quality 100 MHz internal signal. Second PLL with frequency offset is used to produce fine frequency resolution signal. This signal is fed to third N-integer PLL following by variable attenuator and amplifier. Low band signal is fed directly from DDS which is used in the second PLL.

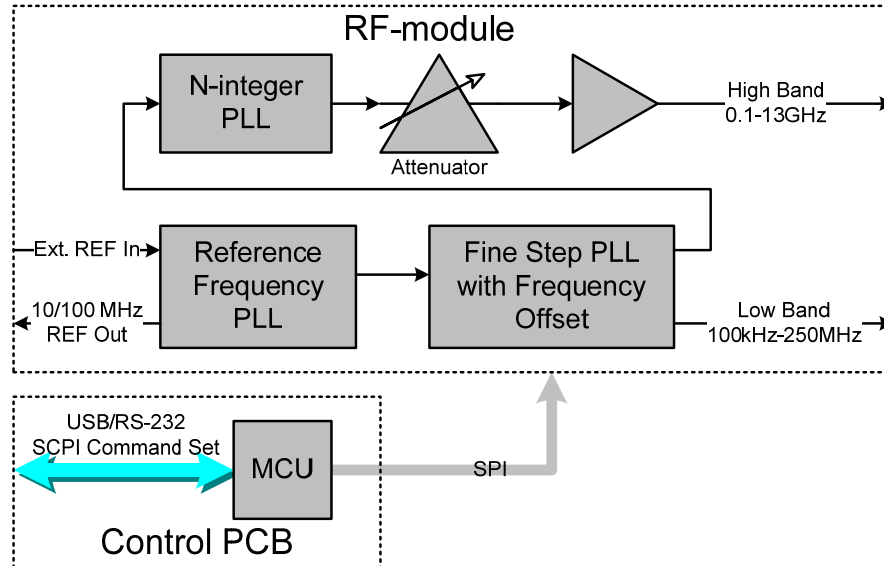


Figure: Block diagram of UNO Synthesizer

Remote Control Operating Manual

The remote control of UNO synthesizer is based on the SCPI (Standard Commands for Programmable Instruments) protocol. It is implemented via RS-232 and USB interfaces located rear panel of the instrument, fig. 1, (USB looks in OS like COM-port), so it can be easily managed by any software which has access to the COM-port of the PC. Only one port can be active, the choice is automatic and remains the same until the power off. The choice of the active port occurs when any byte comes to the one of the ports (not necessary a valid command). The connection via USB is implemented with aid of USB to UART bridge, so from the view point of PC software it looks like a COM-port, fig. 16. This method requires driver installation (integral circuit CP2102). The driver is available for the following OSes: Win2K/XP/2K3, Vista, Windows 7, Mac OS, Linux 3.1. The driver (its last version) can be downloaded from CP2102 bridge vendor site Silicon Labs



Figure 1: Rear panel

1 Quick Start

As a simple remote control application you can use the standard Windows application – HyperTerminal (Start ▸ Programs ▸ Accessories ▸ Communications ▸ HyperTerminal) to send the SCPI commands to the instrument, fig. 2. COMport settings are the following: 115200 bps, 8 data bits, parity none, 1 stop bit, flow control none, fig. 3.

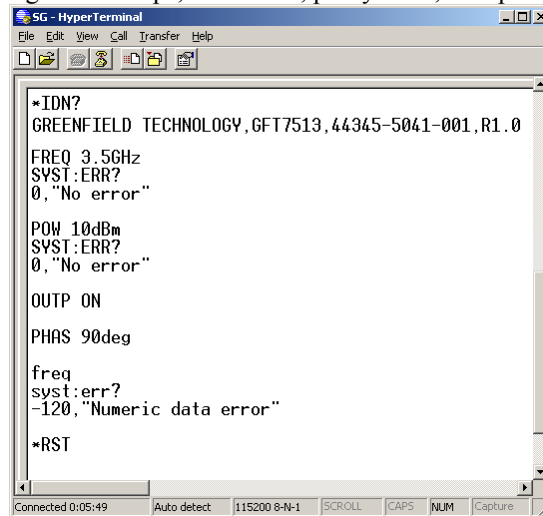


Figure 2: Example of the remote control using HyperTerminal application

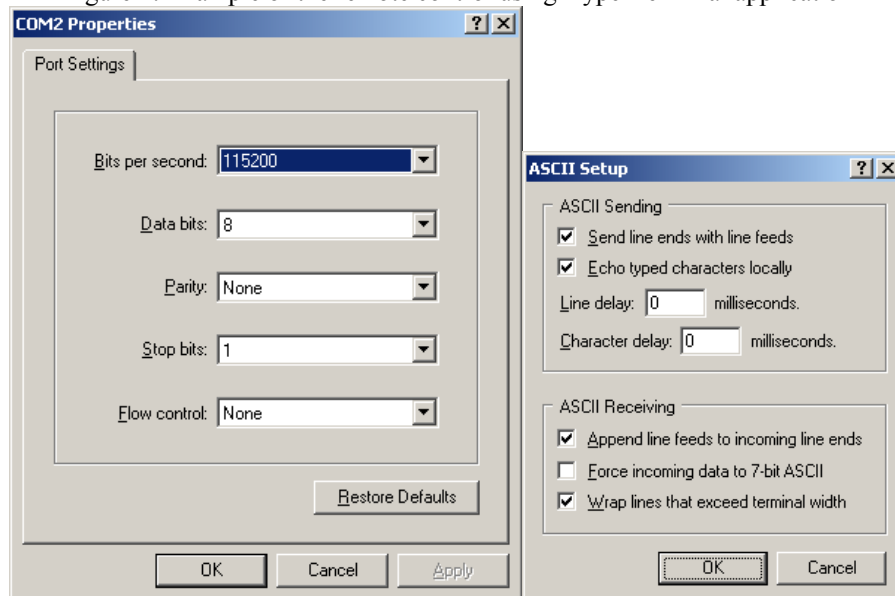


Figure 3: COM-port settings

For more convenient use of the HyperTerminal it's recommended to configure the settings (File ▸ Properties, Settings tab, ASCII Setup... button) “Echo typed characters locally” and “Send line ends with line feeds”. For example, the following typed sequence in HyperTerminal application

```
*rst
freq 100MHz
pow -1dBm
```

will reset the instrument, set the output frequency to 100 MHz, and output RF level to -1 dBm.

2 SCPI Commands

Figure 4 shows the model of SCPI command processing.

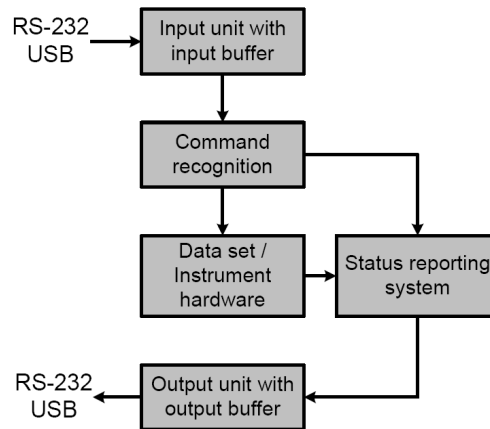


Figure 4: Instrument control and command processing model

2.1 SCPI Compliance

Instrument control command set is based on the SCPI v. 1999.0, but it's not fully compatible. The differences are listed below:

- not all required commands are implemented (see the command list);
- command parser recognizes only one command in single string, and the
- string size shall not exceed 64 symbols;
- command buffer length equals two, i.e. it is possible to send second command
- after the first one immediately (not waiting for completion of the first command), but not more; In applications where one command follows after another immediately without any time delay, for proper SCPI operation it is recommended to use *OPC? command
- status information is not fully supported;
- not all data formats are supported;
- documentation doesn't meet all the requirements of the standard.

2.2 SCPI Summary

Some notes listed below can help the newcomers to start the work with SCPI instrument.

- SCPI-commands are not case sensitive, i.e. commands *RST and *rst are identical.
- The first part of the command name, which is written in capitals, corresponds to a short form of the command. For example, to set frequency you can write `FREQ`, as well as `FREQuency`.
- Commands placed to the brackets are optional. For example the command `[SOURce:]FREQuency[:CW]` have the same effect when written in following forms `SOURce:FREQuency:CW`, `FREQuency:CW`, `SOURce:FREQuency`, `FREQuency`.
- If command requires the numerical parameter expressed in some units then the default unit exists. For example commands `FREQ 1GHz`, `FREQ 1E9Hz` and `FREQ 1000000000` sets the same frequency value 1 GHz, and the default unit is one Hz.
- Commands ending with a question mark, are the queries and return a response. For example the `*IDN?` command returns an instrument ID. Many commands that set some value, also have a query form to read the current value, e.g. `FREQ?` – returns the current value of frequency.
- Commands that provide input of numerical parameters, can also take the following: `MINimum` – minimum value, `MAXimum` – maximum value, `DEFault` – default value. For example `FREQ MAX` command sets the maximum frequency (13 GHz), and `FREQ DEF` sets 1 GHz (default value).
- If the entered command didn't produce the desired effect, it's worth checking the error buffer by means of request `SYSTem:ERRor:[NEXT]?`. If it were not an error, the return value will be the following string 0,"No error", otherwise it returns error code and short description.

2.3 SCPI Command List

2.3.1 *CLS

*CLS command clears error buffer.

2.3.2 *IDN?

This query command returns the string containing the information about the instrument in the following format: *manufacturer,part number,serial number,firmware info*.

2.3.3 *RST

This command presets the instrument: mode CW, frequency 1 GHz, level 0 dBm, output off, internal reference, reference output off.

It's a good practice to start the remote control process with the *RST command.

2.3.4 *OPC?

This query command returns 1 after its completion. It means that all previous commands are completed. It's good practice to use this command instead of fixed time delay between commands.

Examples: If you need to send several commands following immediately one after another, for example to set frequency and level, it's recommended to do it this way:

```
freq 100 mhz
*opc?
1
pow 1 dbm
*opc?
1
```

2.3.5 SYSTem:ERRor[:NEXT]?

This query command returns the string containing the error code and its description from the error buffer. If the buffer is empty then the returned string contains 0,"No error".

Error buffer is organized in form of FIFO (First In First Out). If input command doesn't meet the requirements of parser, or for whatever reason cannot be executed, the corresponding message is placed to the error buffer. The buffer can contain 2 messages. There are two ways to clear the buffer: by reading the errors with aid of SYST:ERR? command one by another, or using *CLS command. When the buffer is already full, and one more error message comes, then the last error message will be rewritten by the following -350,"Queue overflow".

2.3.6 OUTPut[:STATe]

The command turns on or off the RF output as the hardware RF OUT ON/OFF button located at the front panel do.

Parameters: To turn on the output: 1 or ON, to turn off – 0 or OFF.

In query form the command returns 0 – if RF output is off, and 1 – if it's on.

Examples:

```
output on
output off
outp:state 1
OUTPUT 0
OUTP:STAT?
```

2.3.7 OUTPut:ROSCillator[:STATe]

The command turns on and off the REF OUT output located on the front panel of the instrument.

Parameters: To turn on the output: 1 or ON, to turn off – 0 or OFF.

In query form the command returns 0 – if REF OUT output is off, and 1 – if it's on.

Examples:

```
output:rosc on
outp:rosc off
outp:rosc:state 1
```

2.3.8 [SOURce:]FREQuency[:CW]

The command sets output frequency

Parameters: Parameter has the following form

[+/-]float_num[E[+/-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ].

The default unit is HZ. Input value is rounded with accuracy of 10^{-4} . If the specified value is out of valid range, then the nearest limit value is applied, error message will not be formed.

Query form of the command returns current frequency in Hz. The result has the following form: [+/-]float_num.

Examples:

```
freq 2.1GHZ
frequency 21e-1ghz
sour:freq:cw 21E8
freq max
```

2.3.9 [SOURce:]POWer[:LEVel][:IMMediate][:AMPLitude]

The command sets RF output level.

Parameters: Parameter has the following form[+/-]float_num[E[+/-]int_num][DBM].

The default unit is DBM. Input value is rounded with accuracy of 10^{-2} . If the specified value is out of valid range, then the nearest limit value is applied, error message will not be formed.

Query form of the command returns current output level in dBm. The result has the following form: [+/-]float_num.

Examples:

```
pow 5.1dbm
source:power 1.23
POWER 123E-2DBM
POW MAX
```

2.3.10 [SOURce:]PHASe[:ADJust]

The command sets RF signal phase offset. The command is active only if PHASe:ENABLe ON.

Parameters: Parameter has the following form[+/-]float_num[E[+/-]int_num][DEGREE].

The default unit is DEGREE. Input value is rounded with accuracy of 10^{-2} . If the specified value is out of valid range, then the nearest limit value is applied, error message will not be formed.

Query form of the command returns current phase offset in degrees. The result has the following form: [+-]float_num.

Example:

```
phas 90deg
PHASE 90DEG
phase:adj 90.1e-1
```

2.3.11 [SOURCE:]ROSCillator:SOURCE

The command turns on and off the REF IN input, i.e. switches between internal and external reference frequency source.

Parameters: Input parameter value may be INTERNAL – internal reference source is used, or EXTERNAL – external reference source is used.

Query form of the command returns current source in use: INT or EXT.

Examples:

```
rosc:source INT
rocs:sour ext
```

2.3.12 [SOURCE:]ROSCillator:EXTERNAL:FREQUENCY

The command sets the value of external reference frequency at REF IN input.

Parameters: Input parameter has the following form: [+-]float_num[E[+-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is HZ. Input value is rounded with accuracy of 10^{-4} . If the specified value is out of valid range, then the nearest limit value is applied, error message will not be formed.

Query form of the command returns current reference frequency value in Hz.

The result has the following form: [+-]float_num.

Examples:

```
rosc:ext:freq 100MHZ
SOURCE:ROSC:EXTERNAL:FREQUENCY 32MHz
rosc:ext:freq DEF
```

2.3.13 MEASure[:SCALar]:TEMPerature?

The command reads the internal temperature of RF synthesizer block in °C.

The result has the following form: [+-]float_num.

Examples:

```
meas:scal:temp?
meas:temp?
```

2.3.14 STATus:QUESTIONable:CONDition?

The command returns current status of the instrument. The result has the following form: integer_num. Zero value means that all is OK, “1” in 4-th bit means that power which was set to the instrument is outside of the calibrated area, “1” in 6-th bit means that PLL is not locked.

Examples:

```
STAT:QUES:COND?
8
```

2.3.15 STATus:QUESTIONable[:EVENT]?

The command returns the value of the Questionable Status Event Register. The result has the following form: integer_num. The command clears Questionable Status Event Register. The 6-th bit of the register is set to “1” if the PLL unlock event happens. Next command will return this value, and then register is cleared until the next event. Zero value means that all is OK, “1” in 6-th bit means that PLL unlock event happened.

Examples:

```
STAT:QUES:EVEN?
32
```

2.3.16 [SOURCE:]PHASe[:ADJust]:ENABLE

The command turns on the phase adjustment mode. This mode has greater minimum phase step (about 3 Hz) and a bit higher phase noise than in normal mode.

Parameters: Input parameter value may be 0 or OFF – phase adjustment mode is off, and 1 or ON – mode is on.

Query form of the command returns current mode in use: 0 or 1.

Examples:

```
phas:enab ON
phase:adjust:enable off
```

2.3.17 [SOURCE:]FREQUENCY[:CW]:BAND

The command sets the active output - high band output (HB) or low band (LB) output which are located on the front panel of the instrument.

Parameters: Input parameter value may be LB – Low band (100kHz-250MHz) is on, and HB – High band (0.1-13GHz) is on.

Query form of the command returns current mode in use: LB or HB.

Examples:

```
freq:band hb
freq:band LB
```

2.3.18 OUTPut:ROSCillator:DIVider

The synthesizer has 100 MHz internal reference frequency. To obtain 10 MHz signal (from REF Out connector located on the front panel of the instrument) the internal divider should be enabled. This command turns on this divider. When the divider is off, REF Out corresponds to 100 MHz signal.

Parameters: Input parameter value may be 0 or OFF – reference frequency output divider is off (i.e. 100 MHz REF Out), and 1 or ON – output divider is on (i.e. 10 MHz REF Out).

Examples:

```
outp:rocs:div on
outp:rocs:div 0
```

2.3.19 SAVE:CURRENT

The command saves current state of the instrument to its nonvolatile memory. These previously saved settings are automatically loaded just after power on. There is one exception: outputs are always in the off state after power on (for safety).

2.3.20 [SOURce:]FREQuency:MODE

The command turns on Sweep Frequency mode.

Parameters: Input parameter has the following form: [CW|SWEep]. Query form of the command returns current mode in use: CW or SWEEP.

Examples:

```
freq:mode cw
freq:mode swe
```

2.3.21 [SOURce:]FREQuency:CENTer

The command sets center frequency for sweep frequency mode.

Parameters: Input parameter has the following form: [+|-]float_num[E[+|-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is HZ.

Query form of the command returns current center frequency value in Hz.

The result has the following form: [+|-]float_num.

Examples:

```
freq:cent 10MHz
freq:cent 1 GHz
```

2.3.22 [SOURce:]FREQuency:SPAN

The command sets frequency span for sweep frequency mode.

Parameters: Input parameter has the following form: [+|-]float_num[E[+|-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is HZ.

Query form of the command returns current span frequency value in Hz.

The result has the following form: [+|-]float_num.

Examples:

```
freq:span 10MHz
freq:span 1 GHz
```

2.3.23 [SOURce:]FREQuency:STARt

The command sets start frequency for sweep frequency mode.

Parameters: Input parameter has the following form: [+|-]float_num[E[+|-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is HZ.

Query form of the command returns current start frequency value in Hz.

The result has the following form: [+|-]float_num.

Examples:

```
freq:star 100MHz
freq:start 1 GHz
```

2.3.24 [SOURce:]FREQuency:STOP

The command sets stop frequency for sweep frequency mode.

Parameters: Input parameter has the following form: [+|-]float_num[E[+|-]int_num][GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is HZ.

Query form of the command returns current stop frequency value in Hz.

The result has the following form: [+|-]float_num.

Examples:

```
freq:stop 1500 MHz
```

2.3.25 [SOURce:]SWEep[:FREQuency]:DWELL

The command sets dwell time for sweep frequency mode. Dwell time is the time between two adjacent frequency steps.

Parameters: Input parameter has the following form: float_num[E[+|-]int_num] [S|MS|US]. The default unit is US.

Query form of the command returns current dwell time in us.

The result has the following form: `[+|-]float_num`.

Examples:

```
swe:dwe1 5 ms
```

2.3.26 [SOURce]:[SWEep]:FREQuency]:STEP[:LINear]

The command sets frequency step for sweep mode.

Parameters: Input parameter has the following form: `[+|-]float_num[E[+|-]int_num]` [GHZ|MHZ|MAHZ|KHZ|HZ]. The default unit is Hz.

Query form of the command returns current frequency step value for sweep mode in Hz.

The result has the following form: `[+|-]float_num`.

Examples:

```
swe:step 1 MHz
```

2.3.27 [SOURce]:[SWEep]:FREQuency]:SHAPE

The command sets the shape of frequency sweep process.

Parameters: Input parameter has the following form: [SAWTooth|TRIangle]. Query form of the command returns current mode in use: SAWTooth or TRIangle.

Examples:

```
swe:shap sawt
```

```
swe:shap tri
```

2.3.28 [SOURce]:[SWEep]:FREQuency]:MODE

The command sets the triggering mode for frequency sweep process.

Parameters: Input parameter has the following form: [AUTO|SINGLE|STEP]. AUTO mode means that frequency sweeps from start to its stop value and this process repeats automatically without input triggering signal.

SINGLE mode means that frequency sweeps from start to its stop value and waits for triggering signal on TRIG In input. After triggering signal event the frequency repeats the process.

STEP mode means that the instrument waits for triggering event at TRIG In input for each frequency step.

Query form of the command returns current mode in use: AUTO, SINGLE or STEP.

Examples:

```
swe:mode auto
```

```
swe:mode single
```

```
swe:mode step
```

2.3.29 [SOURce]:[SWEep]:RESet[:ALL]

The command resets the current sweep cycle.

Examples:

```
swe:res
```

2.3.30 [SOURce]:[ROSCillator]:INTernal:FREQuency:ADJust

The command adjusts internal reference oscillator frequency.

Parameters: Input is the value from 0 to 1023. 0 is the lowest frequency setting (about -1 ppm), 1023 is the highest (about +1 ppm).

Query form of the command returns current adjustment setting.

Examples:

```
rosc:int:freq:adj 625
```

2.3.31 [SOURce]:[ROSCillator]:INTernal:FREQuency:SAVE

The command saves current internal reference oscillator frequency adjustment setting into nonvolatile memory of RF-module.

Examples:

```
rosc:int:freq:save
```

3 Firmware Update

MCU which controls the synthesizer RF-module has two types of nonvolatile memory: Flash that contains program code, and EEPROM which stores all information about the device, including information about current firmware version. Thus the firmware update consists of two files in Intel Hex format, one – for Flash, and the other – for EEPROM. Firmware update is carried out through RS-232 or USB interface, connected to the PC. The connection via USB is implemented with aid of USB to UART bridge, so from the view point of PC software it looks like a COM-port, fig. 5. This method requires driver installation 2 (integral circuit CP2102). After driver installation and when the instrument is connected to the PC, the COM-port appears in the hardware list. Its number is indicated at the end of the string (fig. 5).

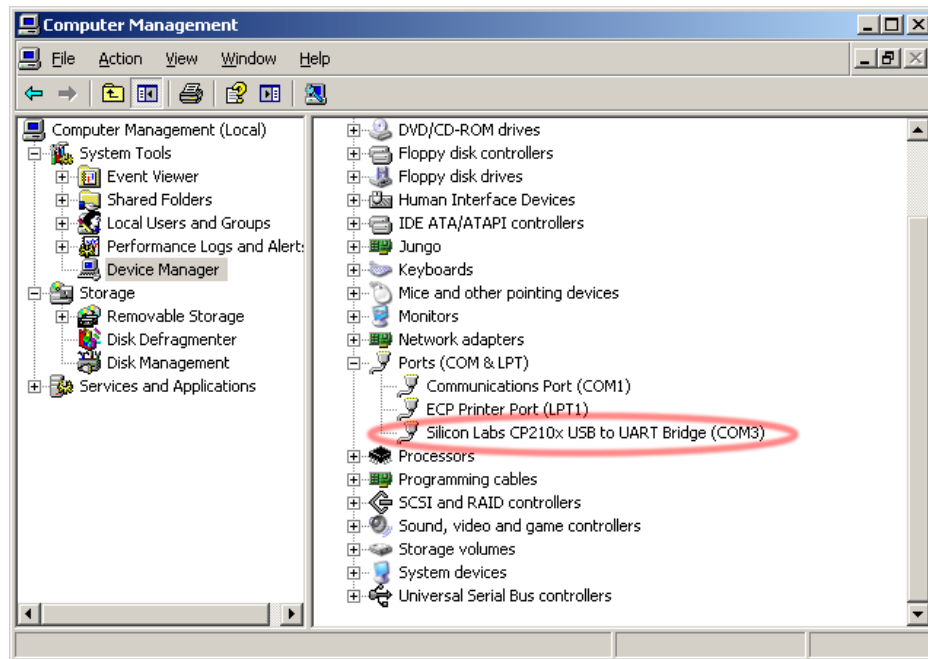


Figure 5: Device Manager window (My Computer > Manage)

When powering up the evaluation board, it enters firmware update mode and waits for the appropriate command from the PC. If it doesn't receive the command within a half of a second, it exits the update mode and no longer responds to the read/write Flash or EEPROM commands, since in regular mode RS-232 and USB interfaces are used for remote control. For the firmware updating the XMI Programmer (XMEGA Instrument Programmer) application is used, fig. 6. The sequence of actions is the following:

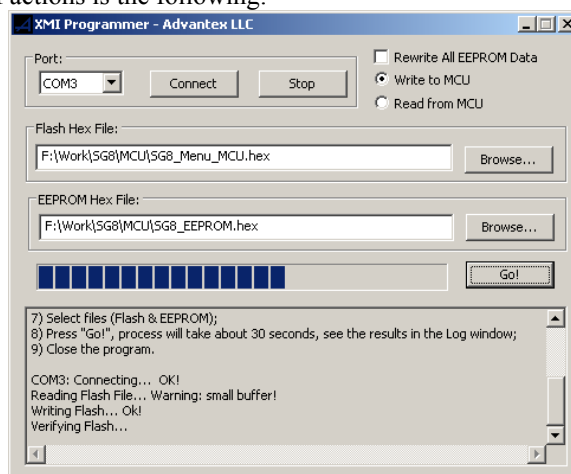


Figure 6: Firmware update application – XMI Programmer

1. Turn off the instrument;
2. Connect it to the PC via RS-232 or USB cable;
3. Launch the XMI Programmer software;
4. Select the port, then press Connect button. If you chose the wrong port, then press the Stop button, select the right port, then press Connect;
5. Turn on the instrument. The connection should be established within a second (you will see the string Connecting... Ok!). If it's not happened then probably wrong port was selected. In such a case press the Stop button, turn off the instrument, change the port in drop-down list, press Connect and then turn on the instrument. The string Connecting... Ok! should be displayed;
6. Select the Write to MCU radio-box, the Rewrite All EEPROM Data flag should remain inactive; If you select this flag, then all EEPROM memory will be re written, not only the part, that contains the information about current firmware revision. That is the personal data of the instrument (serial number, operation time, power-on count) will be erased, which is not desirable
7. Select the files for Flash and EEPROM;
8. Press Go! button. The firmware update process takes about 30 seconds and comprises the data integrity check, the results of the operations are displayed in the log window;
9. After successful verifying the written data you may close the application. You can save the current firmware and EEPROM data by selecting Read from MCU radio-button at 6-th step.