Spectrometric solutions from components to systems

JETI TECHNISCHE INSTRUMENTE GMBH

Technical Note 18

Direct operation of specbos 1211-2 and 1201 with Firmware Commands

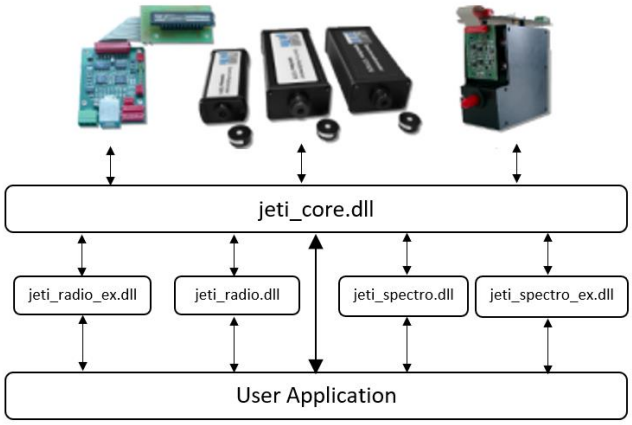
JETI's **specbos 1201 and 1211-2** can be controlled directly to make possible an implementaton into customer specific programs. The possibilities to control the instrument are as follows:

- by DLLs (in SDK) or
- by firmware commands

It can be freely chosen between these possibilities, but one should note, that DLLs are the easier way to control the device (without almost no disadvantages or limitations in comparison to the firmware commands). Generally it is possible:

- to change settings
- to control hardware features of the device (like shutter or laser)
- to proceed measurements
- to get measurement results
- to make calculations (a number of radiometric, photometric and colorimetric calculations can be done directly by the device).

DLLs and a firmware command list are included in the standard delivery of a JETI specbos device. Please note that the documentations on DLLs and the firmware are provided with full description of all functions for completeness, but concerning many of them it is hard to imagine an external application where they could be useful. For example, functions which give spectra in terms of pixels (of the CCD) are widely used in internal testing software of JETI, but they can hardly be of interest for an external user. So it is recommended not to depart from examples and schemes that are given hereafter.



This application note concerns operation with firmware commands.

Content

1 First Steps with Firmware Commands

1.1 COM-port Conventions

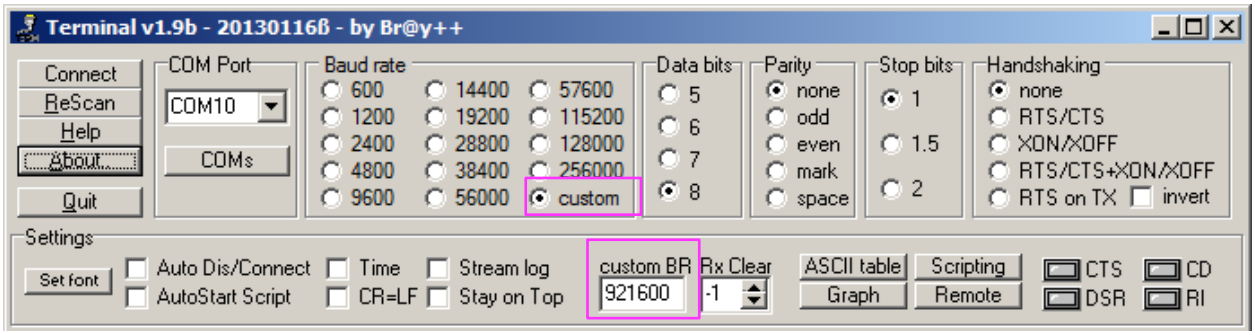
A COM-Port with a connected JETI-device should be opened with following parameters:

Baud rate (Baud):	921 (921 600 Bd – specbos 1201 and specbos 1211)
	115 (115 200 Bd – specbos 1211BT/ RS)
Data-Bits:	8
Stop-Bits:	1
Parity:	none
Flow-Control:	none

1.2 Getting in Touch with Firmware Commands

The easiest way to start operating with firmware commands is to use any terminal program, which allows communication through a COM-port.

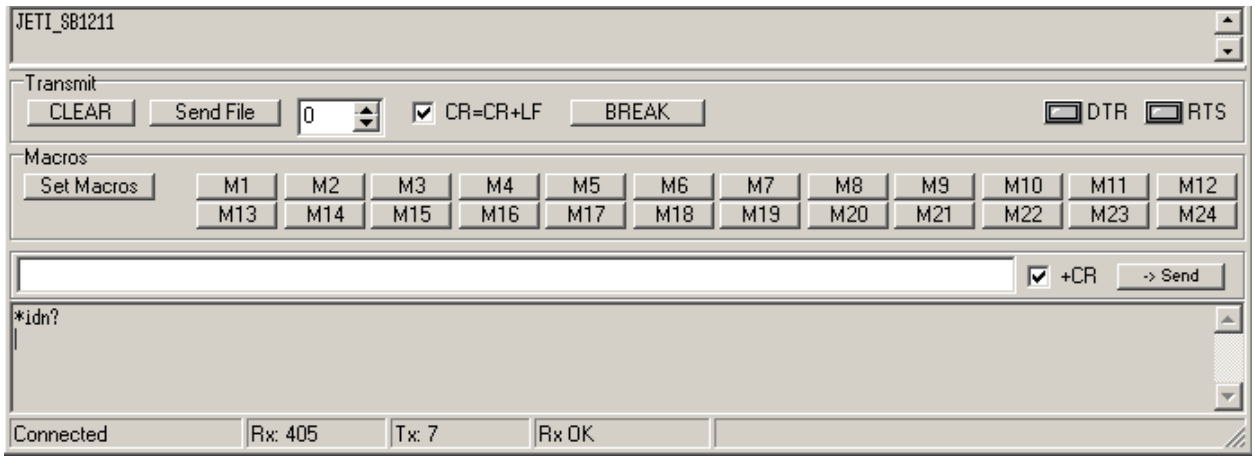
Let us take the *Terminal* freeware program. First of all, we do the necessary settings as follows:



Important are: baud rate – we set its radio button onto “custom” and input the value of 921 600 in the “custom BR”-input field below. If you have a specbos 1211-BT-2 or specbos 1211-RS-2, use 115 200 setting. *Data bits* is set to 8, stop bits is set to 1, *handshaking* is set to “none”.

Then choose the COM-port, to which the device is connected, from the drop-down list and click *Connect*.

Now we are about to send something to the COM-port. Type “*idn?” and press *Enter*. If all the settings are correct, the device will give you an answer, for example, “JETI_SB1211”.



Now the communication works.

1.3 Twiddling with Different Commands

Now direct your device to some light source, for example, to this document opened on the monitor of your computer.
Go to *Terminal*, type "*contr:laser 1<CR>" and send this string to the COM-port.
The laser pointer of the device switches on, and you receive one byte as an answer. The ASCII code of this byte is 0x06, which means *acknowledge* (ACK).
Try to make a mistake in the command. If you would send to the COM-port something, that the device doesn't understand, you will receive a byte with the code 0x15, which means *not acknowledege (NACK).*
It is always possible to ask the device for the state of its controls and parameters. Send the string "*contr:laser?<CR>" to the COM-port. The answer will be "laser: 1", what in ASCII codes looks like 0x6C 0x61 0x73 0x65 0x72 0x3A 0x09 0x31 0x0D.
It is to notice, that every string received as an answer ends with a *carriage return* character 0x0D. And every meaningful number is preceded by a tab character 0x09.
So an important thing to do when you start with programming is to create a routine for splitting of strings according to this pattern. Note, that an answer can contain more than only one string.
Send the string "*contr:laser 0<CR>" to the COM-port to turn the laser off.
Now it is time to make a measurement.
Send "*meas:sprad 0 1 7<CR>" to the COM-port.
First you will get an acknowledge byte 0x06. After some time, which the device has used to make a measurement, comes another byte: *bell* 0x07. This one means, that the measurement is completed. Then comes a large table with wavelengths and corresponding values of measured spectral radiance (for the certain format was responsible the 3rd argument of *meas:sprad command, the "7"). The result looks like as follows:

After executing of "*meas:sprad" command, many photometric values are computed automatically. Send "*fetch:chromxy<CR>" to the COM-port. And then – "*fetch:photo<CR>". The answers contain the xy color coordinates and the luminance of the measured light source.

2 Remarks on Writing a Program

The rest of this paper aims to give you an idea, how a program for controlling of JETI-devices looks like. It is assumed, that the programming language that you use has necessary tools for communicating through COM-ports, and you have already written a routine for extracting numbers from strings.

2.1 Interface

The most essential controls, which your software can include, are:

- Measure-button, Cancel-button (it can be one button, changing its meaning before and after every measurement) to start and to break a measurement.
- Averages-field. Input field for an integer number of averages.
- Laser on/off-toggle-button for switching of the laser pointer (a nice idea is to make its color dependable on its state: for example, if the laser is on, paint it red).
- Sync. measure-button, to measure synchronization frequency.
- Sync. frequency-field to display the measured synchronization frequency or to let a user input it manually.
- Sync.-toggle-button for switching the synchronization on and off.

Also a couple of optional controls are possible. For example, a check-box for making measurements continuously, or start and end wavelengths if you want to display spectra.

2.2 Integration Time Adaption

Both **specbos 1201** and **1211-2** can automatically adapt the integration time to the brightness of the light source. Although it is always possible to use fixed integration times, there might be only very special radiometric applications, where fixed integration times should be used. Fixed integration times can lead to over- and underexposure and in this case all further calculations are meaningless (expecially in the case of overexposure). So it is strongly recommended **always to use the automatic adaption**.

2.3 Low Speed of specbos 1201

A **specbos 1211-2** is approximately 10 times faster than a **specbos 1201**. For example, while **specbos 1211-2** needs 5 seconds to proceed a measurement of some light source, **specbos 1201** needs 50 seconds (such integration times would practically mean, that the light source is pretty dark, e.g. a black color on a TV-set). In such applications like monitor calibration, where many measurements are required, these times can add up to huge time periods of many hours.

So it can have a sense to narrow the range of possible integration times of **specbos 1201** (for monitor calibrations it is recommended to set the maximal integration time to 4 seconds). Although it does lead to underexposure and therefore to measurement inaccuracy, it can be often tolerated in practice because users

rarely have an appropriately darkened room which allows really accurate measurements of dark light sources.

2.4 Synchronization

Some light sources (PWM driven LEDs, displays and TV-sets are among them) are modulated. It means that their brightness changes very quickly in time – the light source "flickers". For example, for a display with a frequency of 50 Hz the periodicity of its "flickering" (modulation period) is 20 ms. While integration times of a **specbos 1201** are nearly always much longer than that, integration times of a **specbos 1211-2** can be almost of the same order and its advantage turns to disadvantage: high speed of **1211-2** leads to loss of reproducibility due to the fact, that not always an even number of flickering-periods fits in the integration time of the device.

To solve this problem, **specbos 1211-2** is equipped with a special sensor on its front panel, that allows synchronization with a light source, i.e. to provide a guarantee that the integration time contains an integer number of modulation periods.

Making synchronization correctly and user-friendly at the same time is may be the most difficult problem for a software engineer.

First of all it is necessary to distinguish between **specbos 1201** and **specbos 1211-2** and to grey out all synchronization features in case of **specbos 1201** if they are implemented in user interface.

Secondly, it is possible to measure synchronization frequency only of a reasonably bright light source (for monitor calibration it would be a good idea to turn – or ask a user to turn the monitor to 80 % white) and in absence of parasitic light of other light sources like another monitor in the immediate neighborhood or ceiling lights.

Thirdly, sometimes a measurement of the synchronization frequency is still impossible even if all the conditions are fulfilled. For example, some monitors lose their modulation if they are set to maximal brightness (100 % PWM setting). That is quite normal and in such a case synchronization can be switched off.

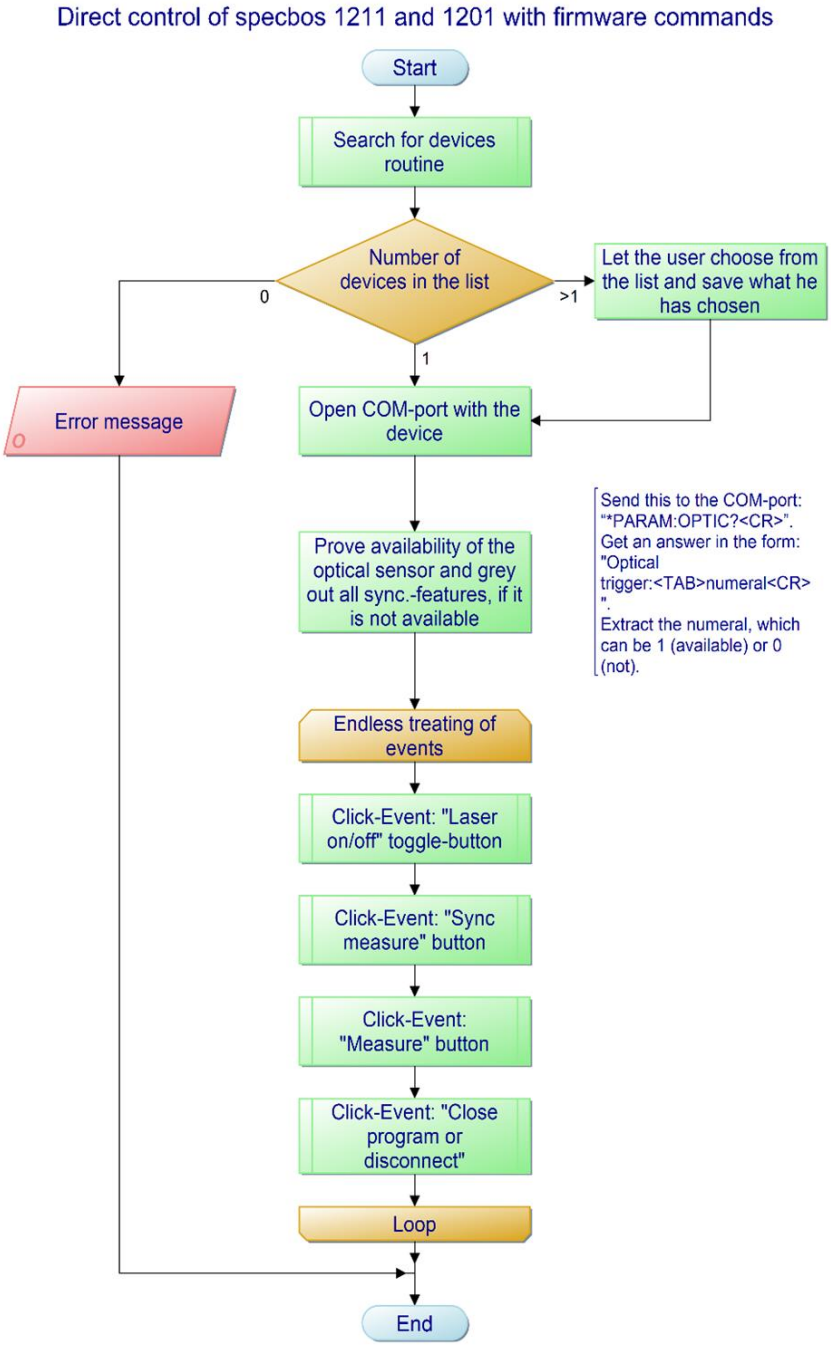
If the modulation frequency of a source is known, it is possible to set it to this value for the measurements.

Summing up, it is important to give the user of a custom software necessary information and carefully differentiate cases when an error-message or a warning must be shown or neither.

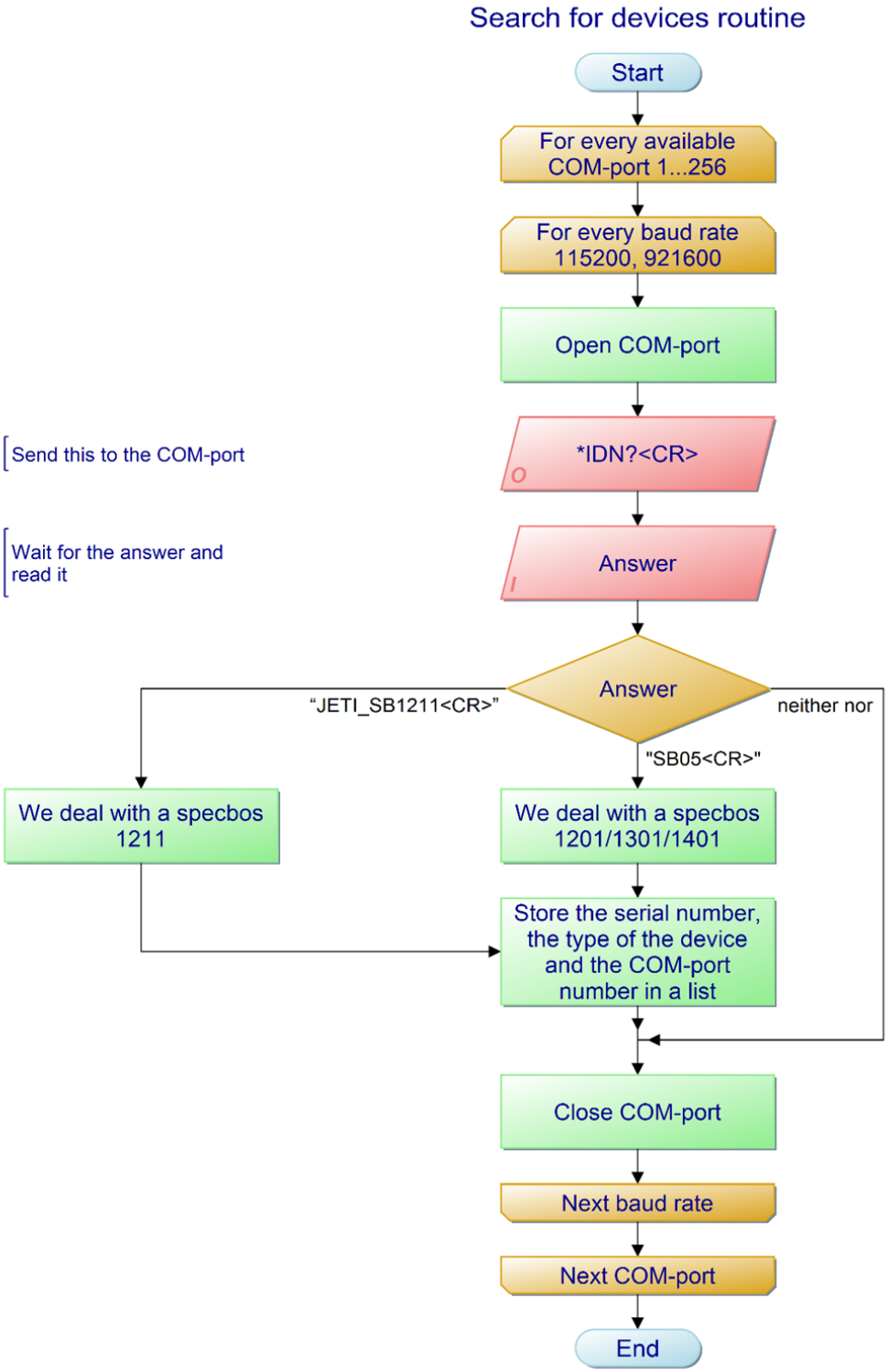
3 Program Structure

The following sequences demonstrate possibilities of device controlling using firmware commands directly.

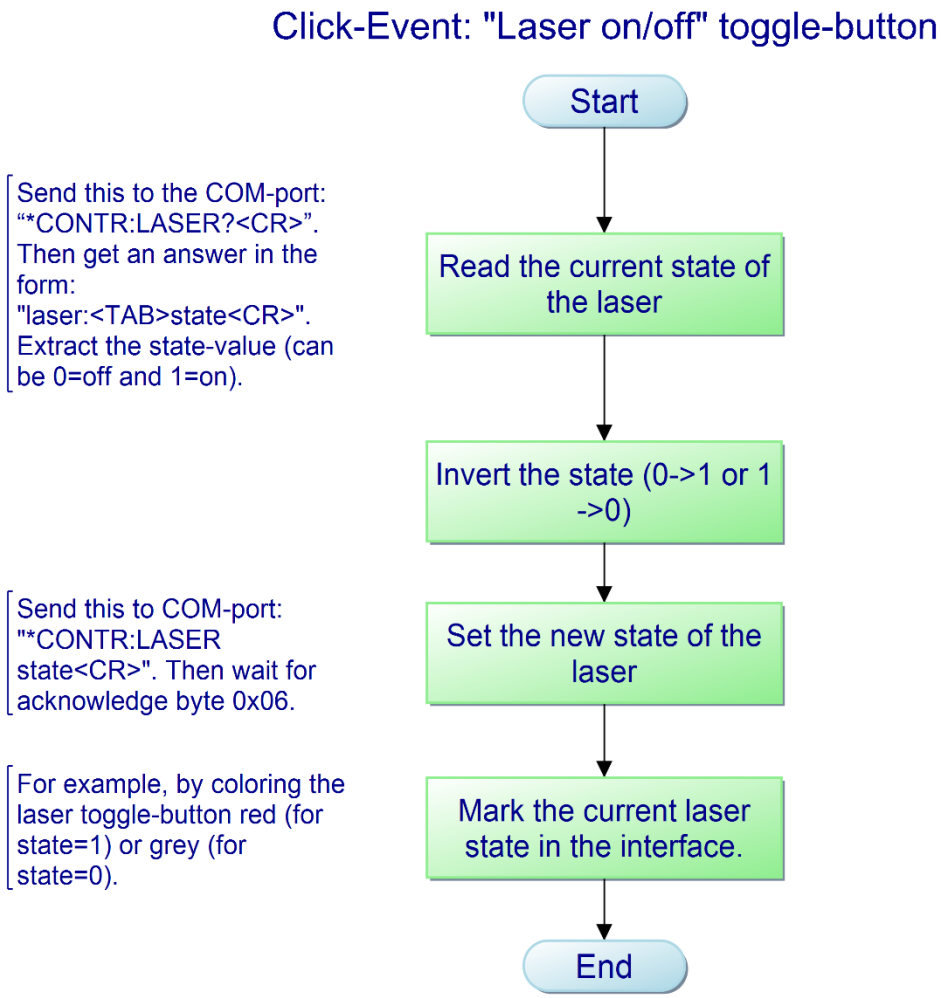
3.1 Main Program Body



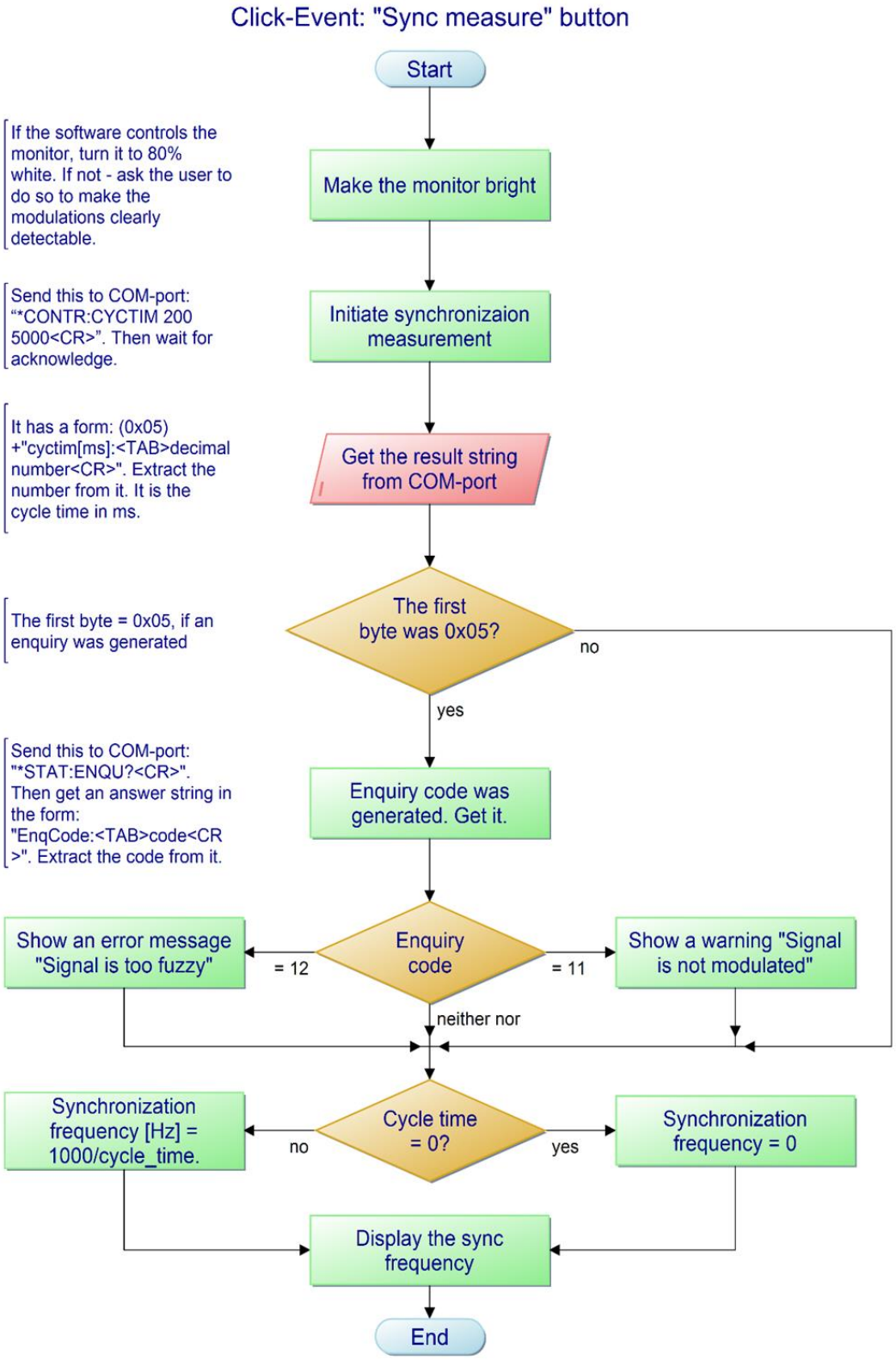
3.2 Search-for-Devices Routine



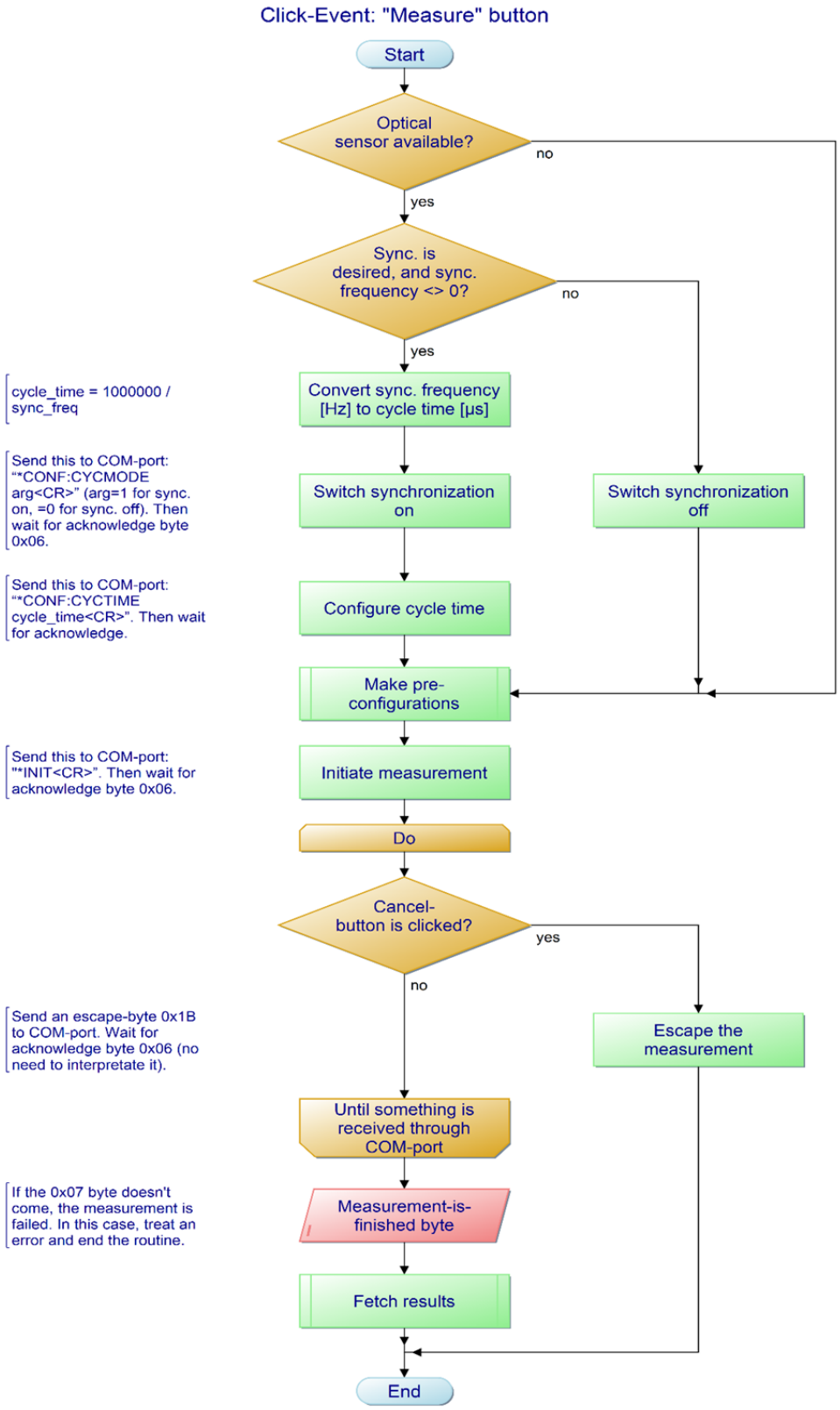
3.3 Laser Switching Routine



3.4 Measure Synchronization Frequency

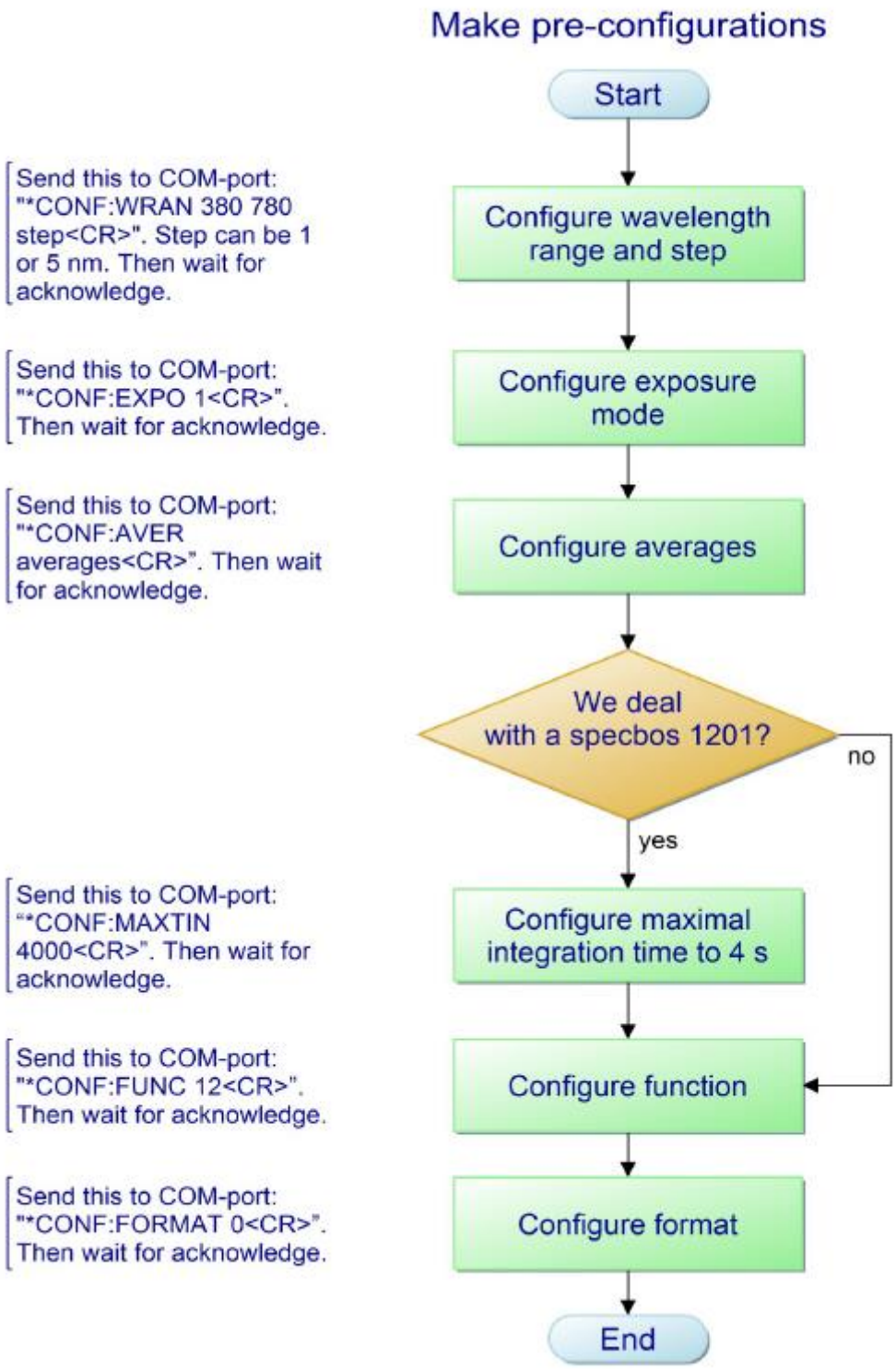


3.5 Measuring Routine

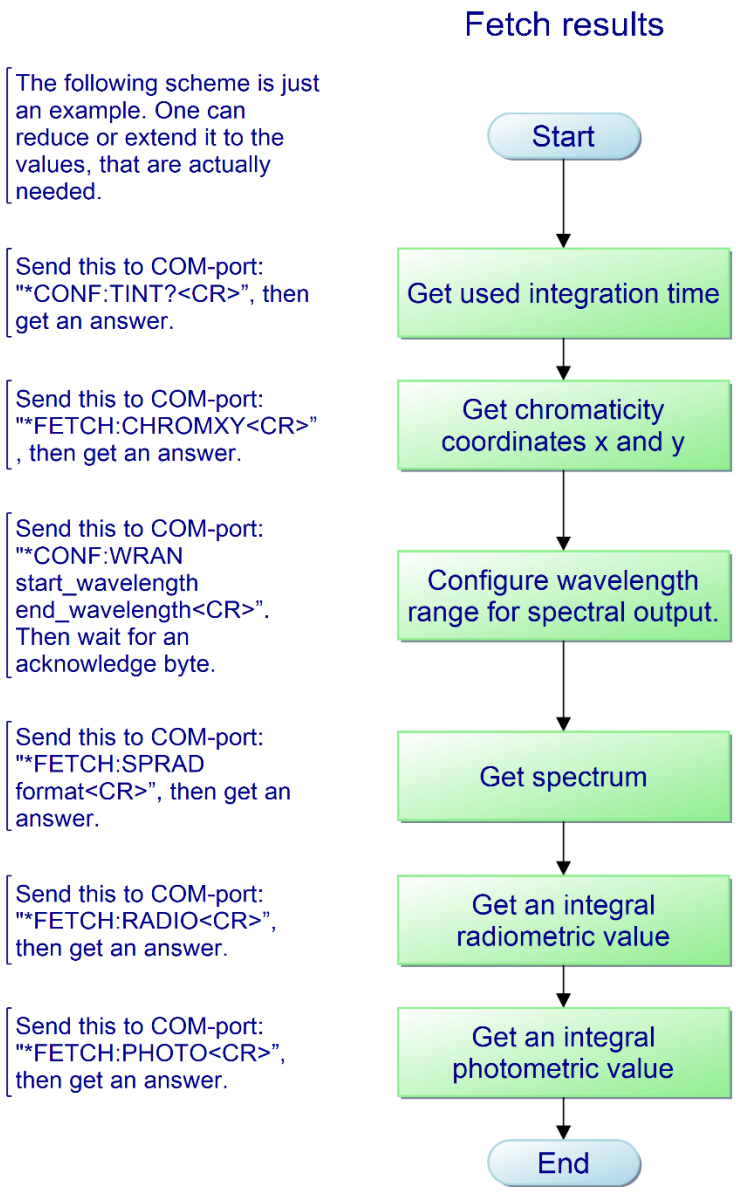


3.6 Pre-configuration Sub-routine

It is assumed, that the number of averages is set by user, and wavelength step for spectral output is chosen (for all non-critical applications it is recommended to use step = 1).



3.7 Fetch Results Sub-routine



For information concerning response formats of certain commands, refer to firmware documentation. Pay attention to the role of „format“-parameter of *FETCH:SPRAD command.

3.8 Disconnect-Routine

Last modified: April 2023