Lab Assignment # 5

Remote control of Instruments Using IEEE488 Interface and Agilent VEE
(Introduction)

By: Prof. Dr. rer. nat. habil. Albrecht Rost
University of Applied Sciences Merseburg
Department 1: Computer Science and Applied Natural Sciences
Physics, Laboratory of Measurement Technology
e-mail: albrecht.rost@in.fh-merseburg.de

Objective

- Control different instruments using soft front-panels of the instrument drivers
- Control instruments using direct I/O
- Save measurement data

Equipment

- Agilent 34401A Digital Multimeter
- Agilent 33120A Function Generator
- Agilent VEE software

Laboratory Experiments

1. Exercises

With these exercises you should learn the basics of remote instrument control, data transfer and data saving using simple programs developed with Agilent VEE.

1.1. Application of instrument drivers with soft front-panel

Use loaded instrument drivers to control different instrument functions and to transfer measurement data from the instrument into the computer. Display the values on the screen.

1.2. Application of direct I/O

Use the object *Direct I/O* to transfer commands and data between the computer and the instruments. Develop a simple program for remote instrument control, data acquisition and data display on the screen.

1.3. Saving data

Investigate different possibilities of data conversion and extend your first programs to save the measurement data.

2. Theoretical Background

(See also the instrument manuals and the book of R. Helsel: Cutting your test development time.)
Today the IEEE488 interface is an international standard of the computer-controlled measurement technology. The IEEE488 bus consists of 24 lines (8 data lines, 3 handshake lines, 5 bus control lines, 7 ground lines and 1 shield line) and realizes a bit-parallel albeit byte-serial data transfer with a

high transfer rate (max. 500 kbytes/s) along short distances (max. 20 m). In a further international agreement the SCPI standard (**s**tandard **c**ommands for **p**rogrammable **i**nstruments) was developed, which defines a set of commands together with the syntax and semantics simplifying instrument control. This standard is compatible horizontally as well as vertically, i.e., all SCPI instruments of the same type will be programmed identically, and the same command is used for the same function at different instruments.

All programs should be developed using Agilent VEE (Agilent **V**isual **E**ngineering **E**nvironment), an object-oriented programming language with graphical user interface, which is especially suited for the development of virtual instruments used as automatic measurement systems. With this tool many problems of measurement technology and data processing can be solved.

Using this engineering environment two methods of instrument control exist: the application of instrument drivers with soft front-panel and the application of the direct I/O, using the special instrument commands and taking notice to the roles of syntax and semantics (see appendix: Remotely Controlled Instrumentation Using Direct I/O and SCPI commands).

3. Instructions to the Laboratory Exercises

Attention: At the beginning all instruments must be switched off. Carefully check all power connections and interface connections. If everything is okay, then start the computer, run Agilent VEE and switch on the other instruments.

The following instruments are used:

- Agilent function generator 33120A,
- Agilent digital multimeter 34401A.

All instruments fulfill the SCPI standard. Use the instrument manager (from the I/O menu) to inform yourself about the installed drivers as well as the properties of the interface and the instruments and the possibilities to edit the interface parameters.

1.1.

In the first exercise you should use the soft front-panels of the instrument drivers to control the instruments and to transmit messages from the instrument to the computer. Proceed as follows:

- Connect the output of the function generator with the input of the digital multimeter;
- Load the soft front-panels of the instruments on the screen;
- Adjust the instruments using the soft front-panels and take notice of the response of the instrument;
- Display the interface communication using the bus I/O monitor;
- Add data terminals to the soft front-panels (function generator: data input *amplitude* and *frequency*; multimeter: data output *reading*) and use suitable objects to enter input variables and to display measurement data;
- Develop a program, which changes a parameter (e.g. the amplitude) in a loop and measures a set of data;
- Document your program.

If there are initial questions, use the online help.

Note: After running the program all instruments are in the remote state, and it is impossible to control the instruments using the hardware front-panel.

The soft front-panel of an instrument driver (see Fig. 1) makes possible to control online the instrument with the mouse and to read the state and the measured value on the screen. The instrument manager shows all available drivers (if necessary further drivers can be loaded). Professional drivers contain an online help (user manual). Using the edit function of the instrument manager, many special parameters of the driver (e.g. name, address, status) can be changed. It is possible to add terminals to the soft front-panel for input or output of data, parameters, instrument functions and control signals.

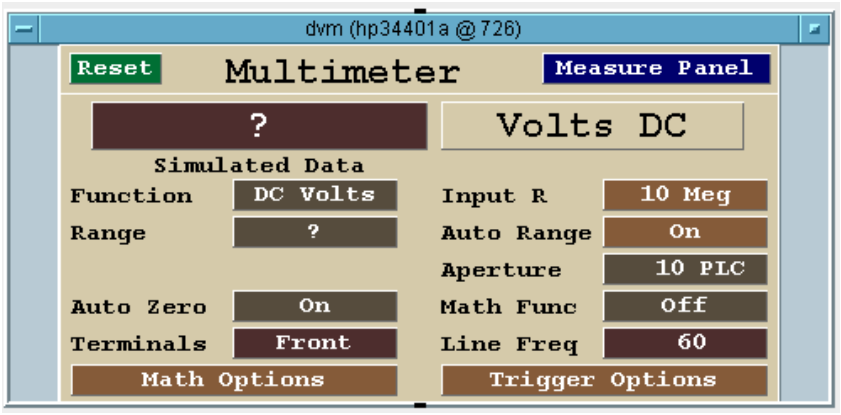

Fig. 1: Soft front-panel of the multimeter 34401A

Note: Instrument drivers can be used also without the instruments working in the *NOT LIVE* mode.

1.2.

(see appendix: Remotely Controlled Instrumentation Using Direct I/O and SCPI commands).
In this exercise you are to program the same functions as in the exercise 1.1., but using direct I/O (see Fig. 2). Additional, the program should be terminated with all instruments reset and switched the local state. Use the bus I/O monitor to display the bus communication, and compare the communication list with that of your program of the exercise 1.1. Document your program.

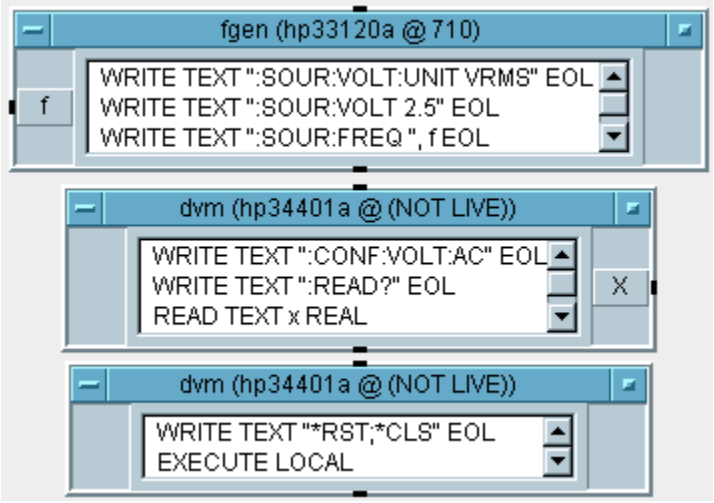

Fig. 2: Examples of direct I/O

The application of direct I/O is useful if no instrument driver is available or if only few functions of an instrument are used. The use of direct I/O shortens the running time of the program. On the other hand, the user must know the instrument commands and take notice of the syntax and semantics of the instrument control language. (See manuals and programming guides of the instruments.)

1.3.

Extend your programs from the exercises 1.1. and 1.2. with possibilities to save the measurement data on a floppy disc using the objects *To Data Set* or *To File*, respectively (see Fig. 3). Verify data correctness by reading back the data using the objects *From Data Set* or *From File*, respectively, and displaying it on the screen. Document the extended programs.

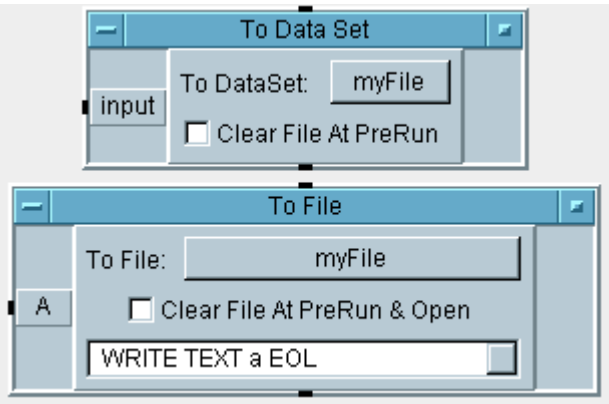

Fig. 3: Objects for data saving

There are two possibilities for saving. It is always necessary to write the correct path to the file into the field with the default name *myFile*. But of the two possibilities of saving data the easiest way is to apply the object *To Data Set* in connection with the built-up of a record of data, which should be saved, using the object *Build Record*: in this case no additional information must be known for reading data, if the objects *From Data Set* and *Unbuild Record* are used.

Appendix 1:

Remotely Controlled Instrumentation Using Direct I/O and SCPI Commands

Syntax of program messages

A command or query is called a program message unit. Such a program message unit consists of a header, or a header separated by a space from one or more parameters (see Fig. 1). The header consists of one or more key words.

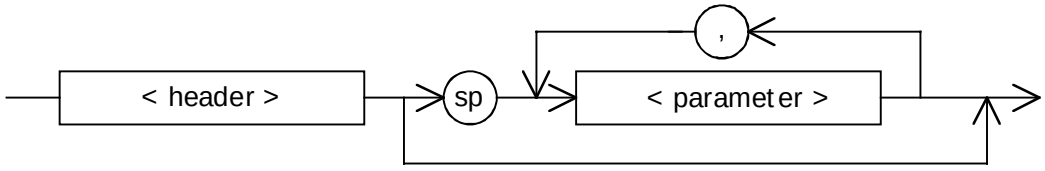

Fig. 1: Syntax of a program message unit

One or more program message units (commands) may be sent within a single program message, separated by a semicolon (;). A program message must be finished by a program message terminator; Agilent VEE sends this terminator automatically.
SCPI commands have a tree structure. The syntax of a command begins from the root level downwards to the leaf-nodes at the bottom of the command tree. All key words begin with a double point (:). Leaf-nodes are the last key words in the command header. The following notes apply to commands:

- Key words written into angular brackets can be left out.
- Basically all commands also have a query form. The response to the query is often the parameter of the command.
- Commands that do not cause a setting or state to be changed (so-called event commands), do not have a query form.
- There are also queries that do not have a command form.
  A command may or may not have parameters. Following data can be used as a parameter:
- numeric values,
- several key words as a special form of numeric data, i.e. MINimum, MAXimum, DEFault,
- other data types, e.g. string, character, Boolean.

Long and short form

Key words (mnemonics) in SCPI program messages may be sent either in long or short form. The long and short forms essentially obey the following rules:

- The long form is the full word of the mnemonic but has a maximum of 12 characters (example: INITiate).
- The short form consists of the first four characters of the long form (example: INIT).
- Exception: When the fourth character is a vowel and the four characters do not form the long form, then the short form consists of the first three characters only (e.g. long form: SWEEP, short form: SWE).

Sometimes a mnemonic is created from a concatenated word or phrase, i.e. PTPeak for peak to peak or AC for alternating current.

Note: Either the long form or the short form may be used in a command message. In a response message only the short form will be used.

Measurement instructions

SCPI offers different possibilities to obtain measurement results. These group of measurement instruction sets have different levels of complexibility and flexibility.

1. :MEASure?

(Simple to use, only few features.)
The *:MEASure?* query configures the instrument, starts the data acquisition, and returns the result. Parameters may be added to give further details about the signal to be measured (see Fig. 2).

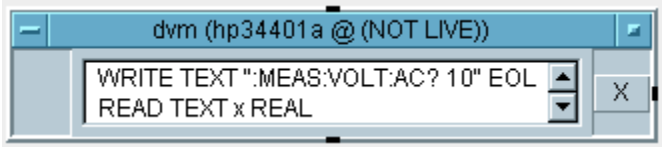

Fig. 2: Application of the *:MEASure?* instruction with parameter (1)

In this case a numeric value is used as a parameter: the parameter 10 is the expected signal range. It is possible to use a data input of the direct I/O object to transfer the parameter as a variable into the direct I/O transaction (the Fig. 3: the parameter is the variable A).

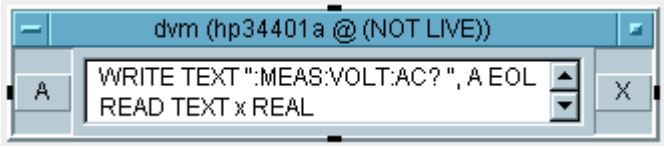

Fig. 3: Application of the *:MEASure?* instruction with parameter (2)

2. **:CONFigure**
   :READ?
   (Somewhat more difficult, but with some extra possibilities.)
   The *:CONFigure* command causes the instrument to choose an optimal setting for a specified measurement. *:READ?* starts the acquisition and returns the result (see Fig. 4).

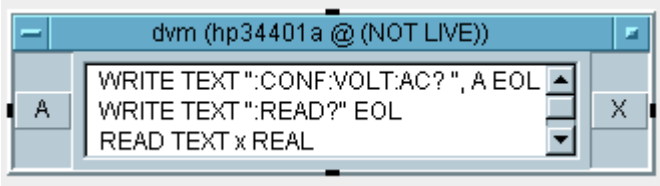

Fig. 4: Application of the *:CONFigure* - *:READ?* sequence.

3. **:CONFigure**
   :INITiate
   :FETCh?
   (Rather difficult to use, but many extra features.)
   The *:READ?* query can be divided further into the *:INITiate* command, which starts the measurement, and the *:FETCh?* query, which requests the instrument to return the measured result (see Fig. 5). (By using this set of instructions it is possible to obtain more than one characteristic of the measured signal with successive *:FETCh?* queries.)

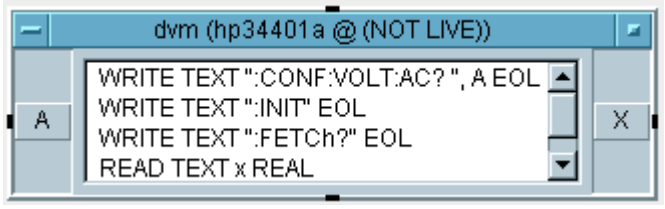

Fig. 5: Application of the *:CONFigure* - *:INITiate* - *:FETCh?* sequence.

Appendix 2: Program Examples

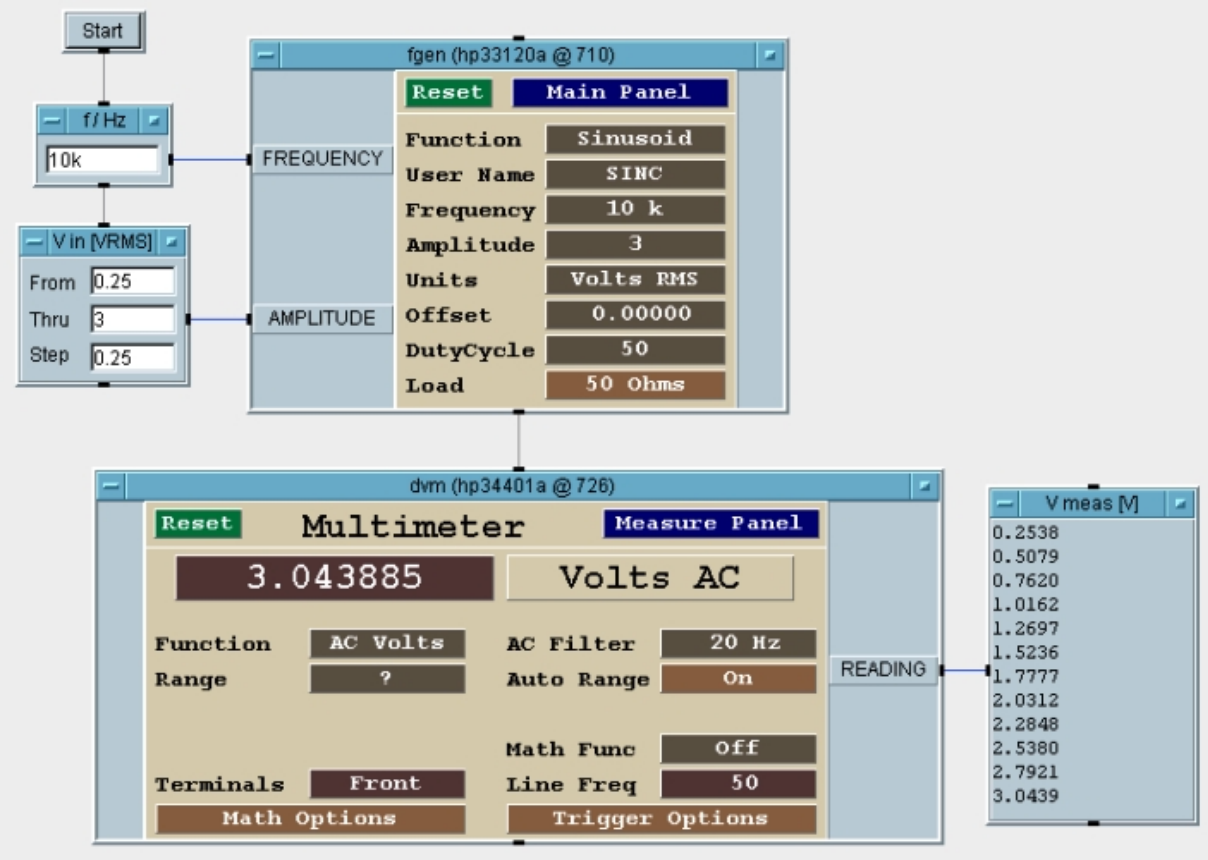

Fig. 1: Instrument remote control using soft front-panels

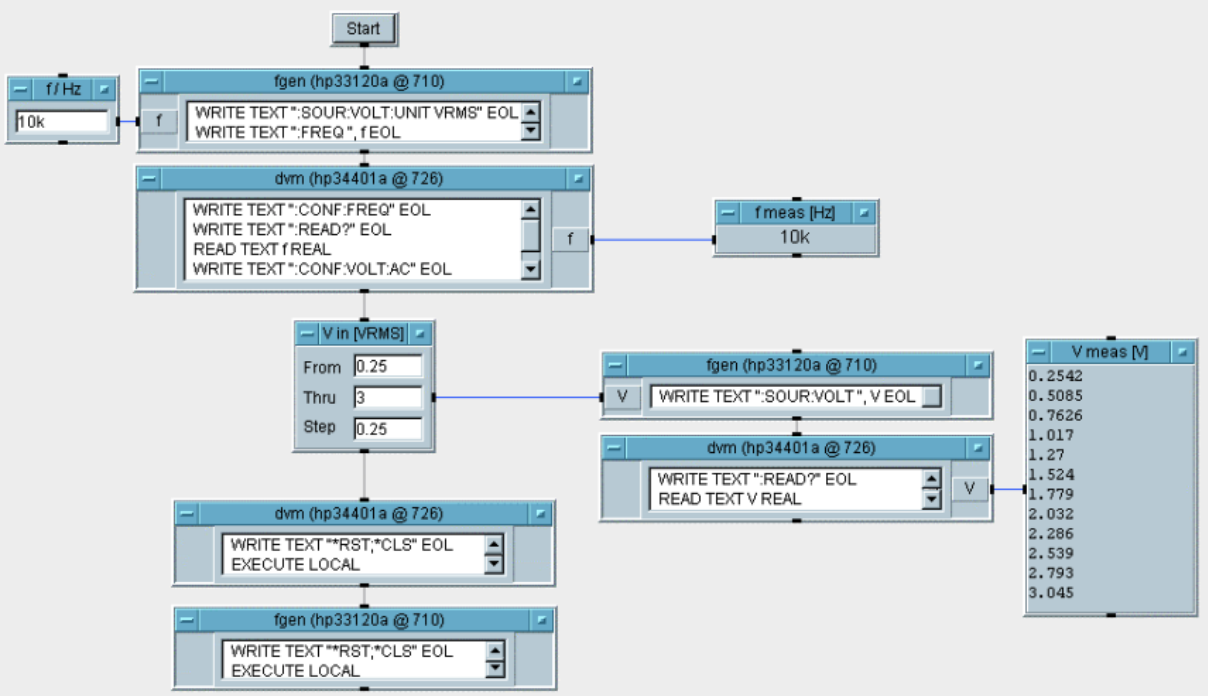

Fig. 2: Instrument remote control using Direct I/O