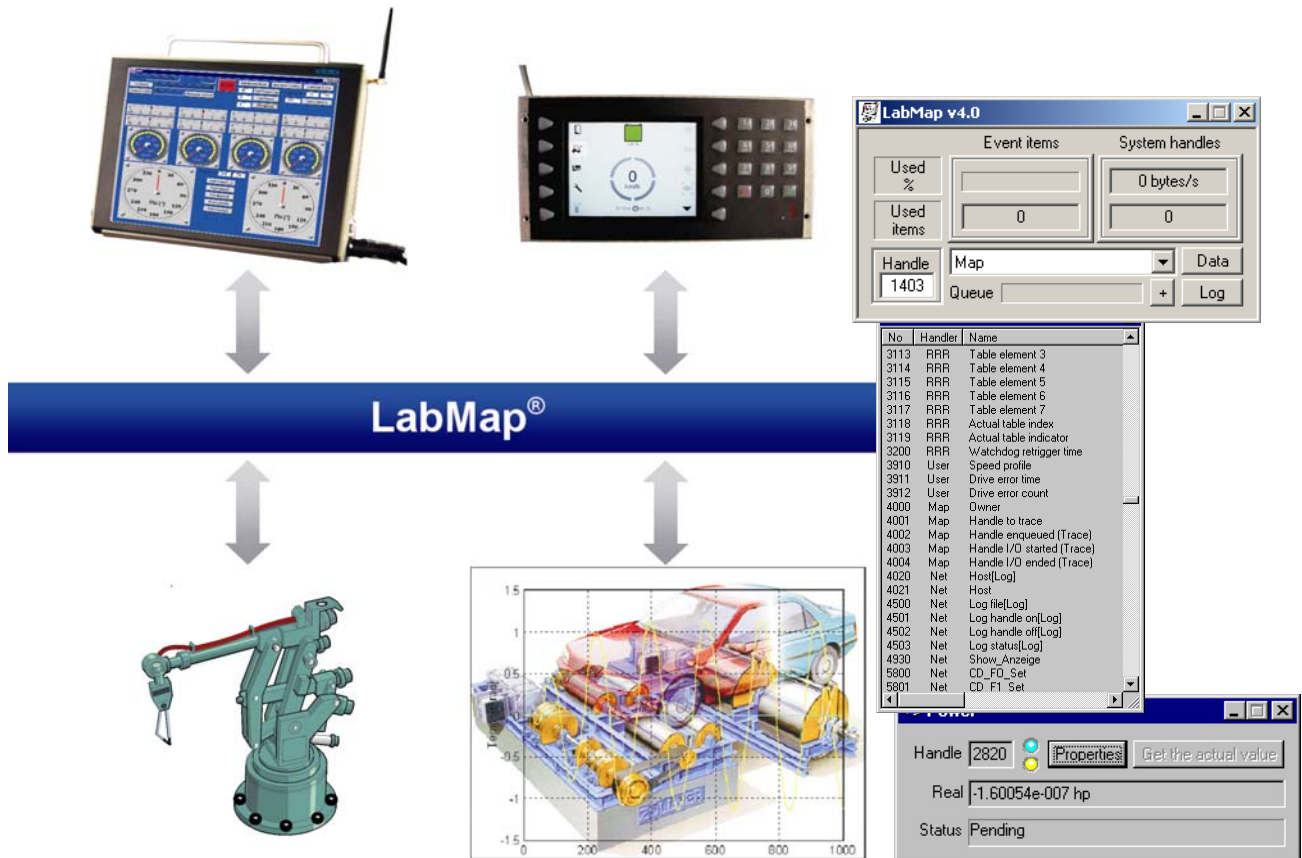


LabAK Server



Author: Dr Dmitry A.Kazakov

The 1st edition, July 2001

LabAKServer

The first edition

Dmitry A.Kazakov

© Copyright [cbb software GmbH](#)

July 2001

LabAKServer is a 32-bit MS-Windows[®] application functioning as an AK server for [LabMap[®]](#). It allows to query the registers of [LabMap[®]](#) using the [AK protocol](#).

1. Register management commands

The server exports [LabMap[®]](#) registers as AK commands with the tags representing the number of the corresponding register. For instance, the register 900 can be accessed using the AK command **0900** (AK commands have 4-character names). Registers of all types and I/O directions can be used.

- The **input** (to be sent) registers accept the register value in the data field of the AK command. When the server receives an AK command associated with an input registers it tries to send the new value to [LabMap[®]](#) (see [LabMapSend](#)). Upon the operation completion it acknowledges the command.
- The **output** (to be requested) registers return the actual value in the data field of the AK command. When the server receives an AK command associated with an output register it tries to request the new value from [LabMap[®]](#) (see [LabMapRequest](#)). Upon the operation completion it acknowledges the command.

Record registers can be requested, but the command acknowledge will contain no data.

2. Access control. SMAN and SREM

Note that the server acts as a separate application. It means that it should have access to the registers to send and request them. Two AK commands are used for this. The command **SREM** is used to get access to [LabMap®](#) registers. Upon receiving the server issues [LabMapAccess](#) for the user associated with the AK server. By default it is *nobody*. The user name can be changed in the main dialog window (see fig 1) of the server. The command **SMAN** causes the server to issue [LabMapDeaccess](#).

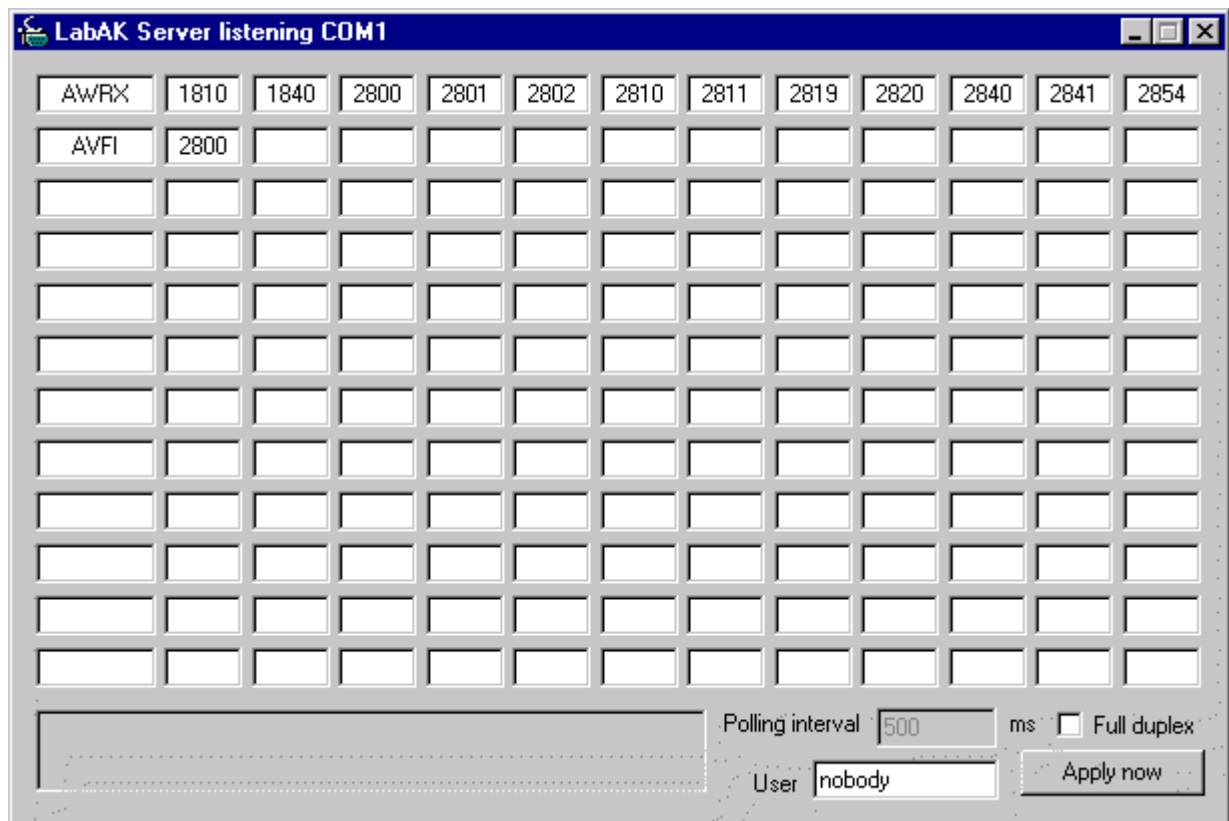


Figure 1. The main dialog panel

3. User defined commands

In addition to **SMAN**, **SREM** and the register commands, an operator can define up to 12 block commands. The block commands can be used to request values of several registers simultaneously and without issuing send or request at the host side. Usually it is state query commands. Upon receiving a block command the server gets the actual values of the specified registers (up to 12), puts them into a space separated list and sends the list in the data field of the acknowledge. In the example on fig 1 the server has two block commands defined: **AWRX** and **AVFI**. If an AK client sends the command **AWRX** to the server, the latter responds with the actual values of the registers 1810, 1840, .. , 2854.

4. Error processing

The server reports errors using the error status field of the [AK protocol](#). There are three possible values of the error status:

- '0' (ASCII) indicates a successful operation completion.
- '1' indicates an execution error on AK host side. The data field contains the error explanation obtained from [LabMap®](#). Usually it is the text returned by [LabMapGetErrorText](#). For instance, if the server has no access, an attempt to send or request a register would fail and '1' error flag will be set..
- '2' indicates a severe fault such as unexisting command etc. The data field contains the error text explaining the fault reason.

5. Full duplex mode

The full duplex mode is an extension of [AK protocol](#) that allows the server to send data to the AK client at its own initiative. This lifts the burden of polling the data from the AK client. In the full duplex mode the server monitors the registers of [LabMap®](#) and sends the modified ones to the client in the format of acknowledge of a register AK command. Note that in the full duplex mode there is no need to poll any registers. They will be automatically sent to the AK client as their values change. When the full duplex mode is activated or the server starts in the full duplex mode, the actual values of all registers are sent to the client. The full duplex mode can be activated using the corresponding check box on the main dialog panel. The parameter polling interval on the main panel specifies the minimal time in ms between two sending of the same register. The default value is 500ms. Note that this has nothing to do with the serial link baud rate. If the link is too slow the efficient polling interval might be longer than specified by the parameter. Note also that the full duplex mode has no effect on the conventional AK commands, that can be sent and acknowledged in parallel to the updates sent by the server. It is also recommended that the AK client software would be able to work in the full duplex mode, as the [AK client](#) by cbb-automation does.