Computer- Control of Surface Science Experiments with Labview

K.D. Schierbaum
Heinrich-Heine-
University Düsseldorf, 40225
Düsseldorf, Germany

M. Cherradi
Heinrich-Heine-
University Düsseldorf, 40225
Düsseldorf, Germany

M. Bouchtaoui
Heinrich-Heine-
University Düsseldorf, 40225
Düsseldorf, Germany

ABSTRACT

A LabVIEW^{TM} 2009-based remote and process control program for an ultra-high vacuum system and a variety of surface science experiments is presented. The apparatus consists of five individual recipients for sample preparation, surface analysis, scanning tunneling microscopy and a central chamber connecting them by means of gate-valves. All processes like electron-beam heating, ion sputtering, gas exposure, thermal and electron-beam evaporation, analysis techniques, sample manipulation can run under full automatic control; transfers are supported.

A variety of useful auxiliary functions are implemented to control bake-out processes, titanium sublimation pumps, pressure regulation of leak valves but also data acquiring, automatic reporting through ftp and emails, as well as external communication and control via a TCP/IP connection with a portable PC. Important features of the software include a time-control module to operate up to five channels (repetitive heating, sputtering, gas exposure, evaporation and manipulation to evaporation positions) as well as full process control. Such tools allows us to set up automatically processed experiments that include e.g. the preparation of single-crystal surfaces, their AES (Auger electron spectroscopy) and LEED (low energy electron diffraction) analysis as well as the evaporation and various treatments of thin films.

General Terms

Process control

Keywords

LabVIEW, surface science, remote control.

1. INTRODUCTION

Process monitoring, data acquisition and instrument control are widely used for surface science experiments but software developers are focusing on isolated applications, preferentially for high-sophisticated spectroscopy rather than providing a single program environment for many different needs of such experiments.

One of the first reports on the implementation of LabVIEW, a well-known graphical programming language developed by National Instruments [1], for this type of experiments goes back to the beginning of the 1990s. Such a program was developed for the use at the ELETTRA synchrotron radiation source [2]. Already the first version offered, apart from interfacing the x-ray diffraction and spectroscopy instrumentation, a number of additional features like sample manipulation and vacuum control. Of course, remote data acquisition and instrument control have become of broader interest for a number of other experiments in research [3,4]. Nowadays the development of commercial solutions of LabVIEW-based control systems for surface science experimental stations is being advertised. Detailed block diagrams of such LabVIEW codes are, however, not available.

We are reporting here a LabVIEW program capable to control a variety of components of an ultra-high vacuum (uhv) setup; our approach also aims at a suitable automation of repetitive experiments such as the preparation of single-crystal surfaces by ion sputtering, thermal treatments and gas exposure, evaporation of thin films, and a subsequent basic characterization by techniques like low-energy diffraction or Auger and photo electron spectroscopy. Block diagrams will be presented that outlines the basic structure of the software; we also demonstrate how the basic LabVIEW nodes are working in connection with the electronic components.

2. THE VACUUM SYSTEM

The home made ultra-high vacuum system consists of five vacuum chambers that are radially arranged around a central chamber; they are equipped with a variety of standard surface science tools. Gate valves between all chambers enable the non-interfering use of the different chambers with respect to vacuum requirements. They are all of electro pneumatic type; instead of mechanical position indicators, micro switches produce status signals. The vacuum is generated by two rotary pumps, four turbomolecular pumps, five ion pumps and four titanium sublimation pumps. Gate or right-angle valves, placed between turbo molecular pumps and the chambers, as well as safety valves in the rough vacuum lines allow scanning tunneling microscopy without running mechanical pumps. A number of gauges monitor the pressure in all chambers. For gas exposure during experiments, leak valves are used. The originally manually operated all-metal leak valves have been modified by replacing the turning knob by assemblies of high-torque stepper motors and gear boxes. Also, a few manually driven gate valves were equipped with pneumatic double cylinders, 5/2 ways valves and micro switches to enable their remote operation. In total, fifteen valves are in use. The number of stepper motors equals 25; they operate three x,y,z,ϕ-manipulators, the extractable LEED optics, the thickness monitor and other devices.

3. HARDWARE AND SOFTWARE

3.1 Computer boards and control electronics

Two desktop PCs are applied in the laboratory, a third portable PC can be used for remote control and monitoring via the internet.

All in all the desktop PCs are equipped with four PCI digital I/O boards (Quancom), each of them providing two 34-pin connectors with 32 input or output channels, two multiple RS232 controller boards (NetMos), one 16x16Bit-AD/4x12-Bit-DA-converter board (Quancom), one PCI-GPIB board (National Instruments) connected with a frequency counter (Agilent 53131A), a DSP-lock-in-amplifier (Stanford Research SR-810), a 6½ digital multi meter (Picotest M3500A) and a 2x16Bit-DA-converter (4861A-12, Meilhaus electronic) as well as one fire wire IEEE1394 board for the LEED digital CCD camera (Hamamatsu C8484-03G01). The RS232 interfaces enable serial communication with two 6½ digital multi meters (Hameg HM 8112-3), vacuum gauges and pumps, two pyrometers (Impac) and a high-voltage supply (1200V-60mA, MCP 140-2000, FUG Elektronik) that is used for the three sources of the electron beam evaporator.

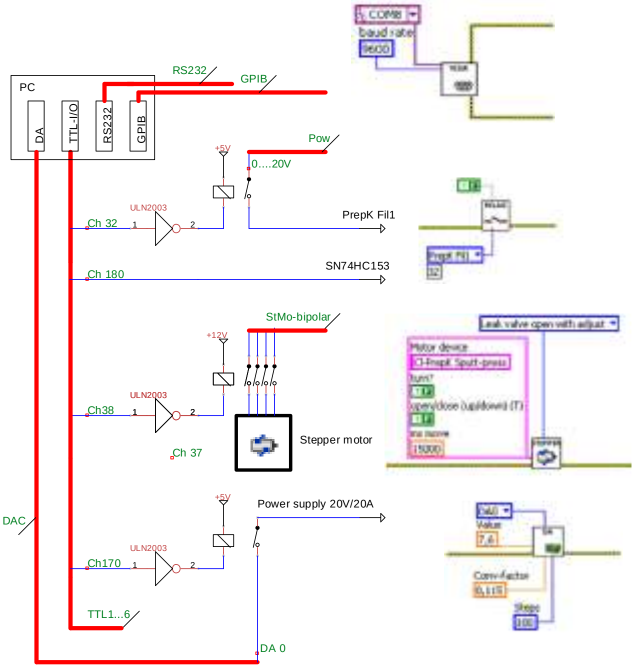


Figure 1. Some key functions of the computer boards and their relation to corresponding nodes of the LabVIEW subVIs. Red (thick) lines in the circuit diagram represent multi pin connections with components, labeled according to their function. "StMo-bipolar" is linked with a bipolar driving board. Most outputs of the I/O TTL board are used with driver chips; a few transmit their signals directly to other components or units with appropriate inputs. Also electronic switches can be used in some cases. The four DAC converter output voltages can be transmitted to a number of instruments, according to the relay switches.

The central electronics, mounted in a standard 19" rack housing, consists of various boards and is connected at the rear panel with all control units of pumps, gauges and valves, the 16x16Bit-AD/4x12-Bit-DA-converter board, the power supplies (constant 5V, 12V, 24V, two voltage-controlled units, 20V/20A and 40V/10A), a 300 s^{-1} pulse generator and a control cabinet placed at the vacuum system. It also contains two driving boards for the unipolar and the bipolar stepper motors (MSE570 EVO2, McLennan, and AIT134, RS components). A 10mA-constant current source is employed to generate temperature-dependent voltages from the various Pt100 elements that are attached to the vacuum chambers. They are required for the vacuum bake-out process. The electronics provides driving voltages for valve solenoids and relays by means of ULN2003 chips, controllable voltages (0 to 10V, with a resolution of 12 bit) used for equipment with corresponding inputs, and two lines with pulsed voltages for the stepper motors; it also bundles and transmits the input voltages from other control units and from the valves to the PC. In addition, a board ensures a proper "computer-off" operation of the uhv system by means of logical IC chips. The development of such a central electronics was the key to significantly reduce the number of interconnections between various individual control units as well as connections to the PC. Figure 1 reveals the basic functions of the electronics.

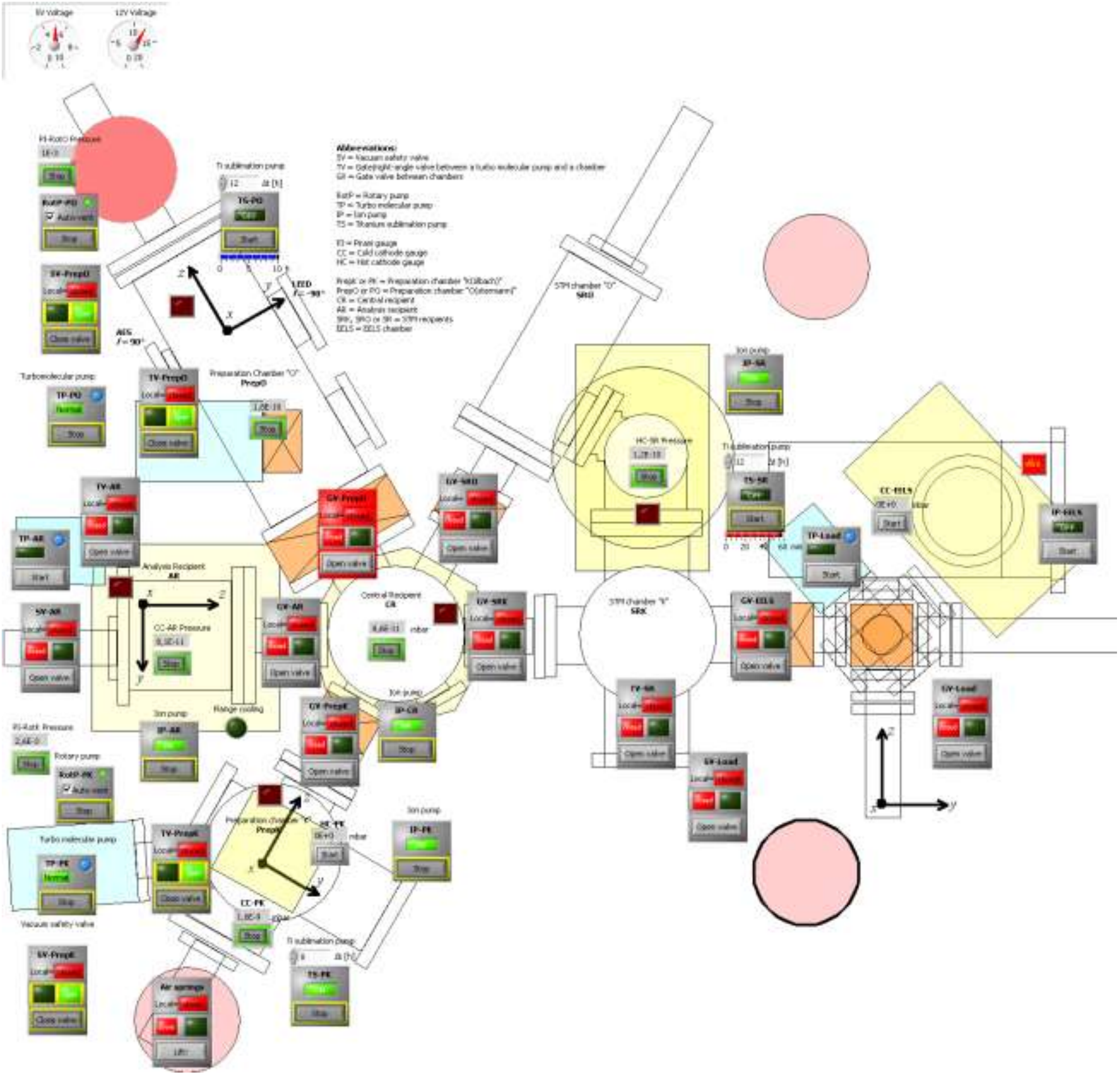
The control cabinet consists of high-current relays for the titanium sublimation pumps, high-voltage relays connected to electron beam heaters and evaporators and for sample grounding, 4-contact relays associated with the stepper motors, and power relays for the baking plates. Thus, a permanent installation of the electrical wiring is attained. Another benefit of this concept is that only a few power supplies and multi meters are required that can be used for a number of different functions.

3.2 How the software works

The software is written using LabVIEW 2009. The initial intention of the program was to control and operate an ultra-high vacuum apparatus, which is used for research and academic training, thereby aiding students to avoid accidental mal-operation of vacuum components. An important issue was also to perform repetitive processes, necessary in their experimental surface science studies, over night in an automated and reproducible mode; the implementation of spectroscopic instruments and the availability of a single program environment were another leading points. And finally, a detailed software-based documentation about the experiments, based on the recording of all available physical data as a function of time, as well as remote control and operation from an external PC were relevant objectives of our approach.

An executable program in LabVIEW is called Virtual Instrument (VI). In accordance with [5], we indicate technical terms associated with the programming language in *Italics* and names of VIs in `Courier`; we use the double quote to refer to *terminals*. A part of the *front panel* of the top-level-VI `GG-control.vi` that we have developed for our purposes is displayed in figure 2.

Here, the *controls* and *indicators* of the vacuum components are labeled according to the type of component and the vacuum chamber to which the component is attached. Abbreviations are explained in the front panel. The control elements are arranged together with an underlying scheme of the vacuum system. They are composed of start/stop or open/close *push buttons* and *indicators* that display the output state of the associated channel of the I/O-TTL board and, in case of gauges, the pressures that are either received via RS232 interfaces or from the AD converter board.



Abbreviations:
SV = Vacuum safety valve
TV = Gate/right-angle valve between a turbo molecular pump and a chamber
GV = Gate valve between chambers

RotP = Rotary pump
TP = Turbo molecular pump
IP = Ion pump
TS = Titanium sublimation pump

PI = Pirani gauge
CC = Cold cathode gauge
HC = Hot cathode gauge

PrepK or PK = Preparation chamber "K(ilbach)"
PrepO or PO = Preparation chamber "O(ttermann)"
CR = Central recipient
AR = Analysis recipient
SRK, SRO or SR = STM recipients
EELS = EELS chamber

Figure 2. Front panel of `GG-control.vi` with control elements for the various valves, pumps and gauges of the ultra-high vacuum setup. The *x,y,z* coordinate axes refer to the motion of sample manipulators. The horizontal extension of the setup equals approximately 3 m. The "vent"-LED lights up if the VI detects a vented chamber; this initiates a guard circuit disabling manual operation of certain gate valves at the central electronics. The alarm state of the valve "GV-PrepO" is shown to highlight this feature.

The control elements of the electro-pneumatically driven valves, which can also be operated locally by real switches at the control electronics, have additional *indicators* that represent the output levels of the electronics, the settings of local switches and the micro switches of the valves; in the *front panel* they are visible as tiny yellow fringes around the *controls* and *indicators*. The remote control outputs of the I/O-TTL board are automatically adjusted by the program to the actual valve state if the valves are **not** operated from the VI; so the valves maintain their state when changing the local switch to remote operation. The electric valves of the cooling water of the turbo pumps, vibration sensors attached to the rotary pumps, automatic ventilation valves, timers of titanium sublimation pumps and alarm displays are useful additions to some control elements.

The operation of other instruments and various processes is organized by a *register card* in the center of the front panel of `GG-control.vi`. Figure 3 highlights the *page* for the preparation chambers. *Controls* are grouped together in *clusters*, according to their functionality, and have a start/stop *push button* and *LED indicators* that show their on/off status.

Every technique and device, available in the two different preparation chambers of the vacuum setup, can be operated with the *controls* of this *page*: gas exposure with leak valves, ion

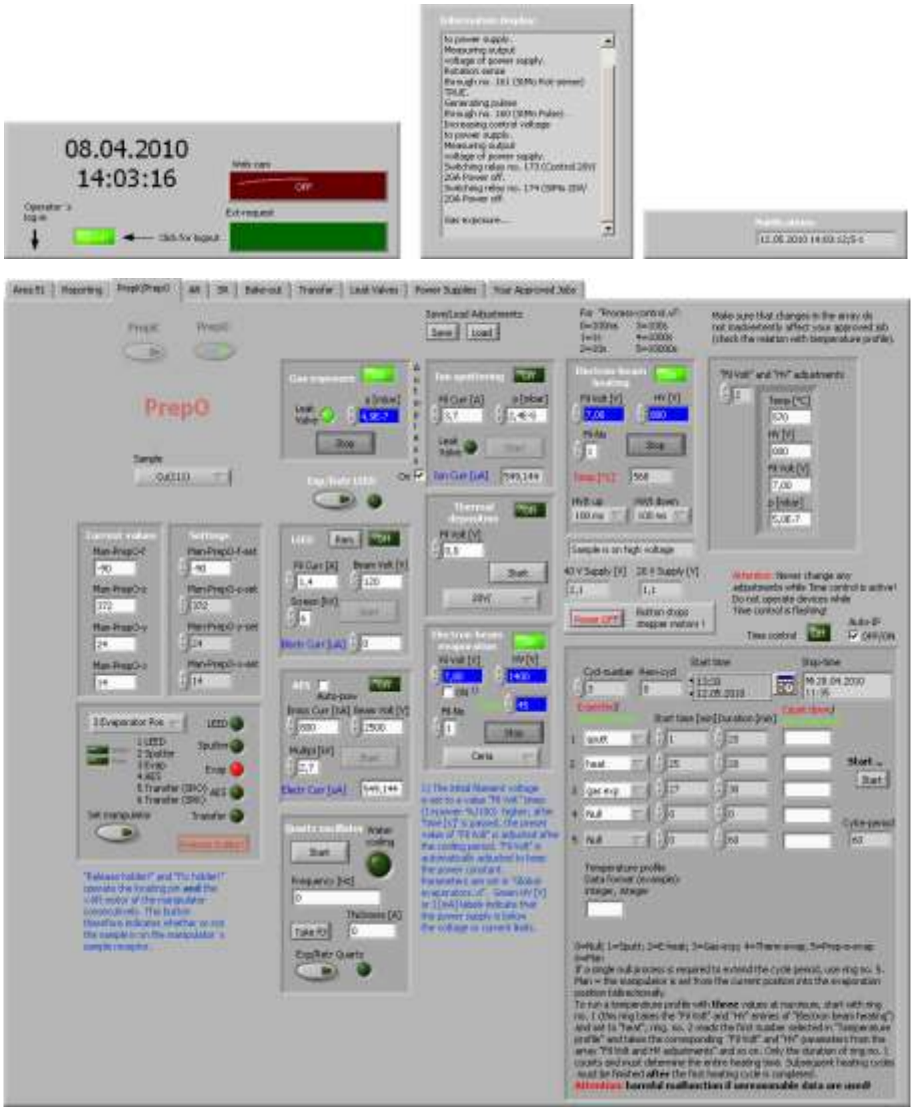


Figure 3. *Register card* of `GG-control.vi`. The elements above the *register card* are used for login, time, display, and the *string field* “Notifications”. Further explanations are given in the text.

sputtering, electron-beam heating, thermal deposition, electron-beam evaporation, the quartz-oscillator for thickness monitoring, low-energy electron diffraction including the movement of the optics, Auger electron spectroscopy and sample manipulation. The operation of a *push or OK button* can only initiate a **complete** process, e.g. ion sputtering or heating of the sample are started or stopped. For example, pushing the button of the ion sputtering module opens the leak valve of the ion gun to a predefined value and increases the filament current of the gun subsequently, both by operating corresponding stepper motors. The status *LED indicator* lights up and the ion current, which is measured by connecting the sample with a digital current meter, is displayed. The electrical connection is made automatically and the instrument is read out via an RS232 interface. Pressing the button again terminates the ion sputtering in a reverse sequence. Parameters can also be changed in the “on” state; the new values are directly converted physically. An automated regulation of the chamber´s pressure, defined in the *controls* for gas-exposure and ion-sputtering, can be activated by a *system checkbox*. One can start as many processes as it is experimentally meaningful.

The manipulation of the sample between various fixed positions of the preparation chambers, selected by a *control ring*, occurs automatically in four or five steps, according to stored sequences of movements in the x, y, and z direction and the angle ϕ about the axis of the manipulator. These positions refer to certain experimental sites in the vacuum chamber; other positions are accessible by setting the numeric inputs. If the manipulator arrives at a fixed position, the corresponding *LED indicator* lights up.

The operation of the low-energy electron diffraction and the Auger electron spectrometer activates also processes on the second laboratory desktop PC which is connected to a CCD camera, a counting unit, a lock-in amplifier and various precision voltage generators and multi meters. The input data like start and stop energy, step width, sensitivity or time constant, necessary to collect the spectra, are typed in the *subpanel* of `GG-control.vi` on the screen on the left hand side (not shown here). These data are transferred from the first to the second PC using *data sockets*.

The *cluster* “Time control” is used for repetitive operations and has five channels that can be assigned to all processes, including the manipulation to the evaporation site. A start time is defined on which the individual start times of the channels refer to. The duration of an operation means the time between its individual start and stop time. If “Time control” is started, the color of the *ring* indicates the active process and the next expected one by means of different colors; in addition, the count-down and remaining times are displayed.

Other pages of the register card are available to access the I/O TTL board and change individual channels, log-in, settings of the vacuum system (“Area51” for administrators), for reporting (short messages stored along with the experiments with date and time, email notifications), operation of other chambers (analysis chamber and scanning tunneling chambers), bake-out, calibration of leak valves, and operation the 220V relays of all power supplies and some control units. The leak valves can be calibrated automatically in the pressure regime of interest; this routine determines the parameters of the semi-logarithmic relationship between chamber pressure and the angle of rotation of the valve. Sophisticated time-controlled experiments can be performed if the user navigates to the *page* “Approved Jobs” of the *register card*. This is explained in more detail below.

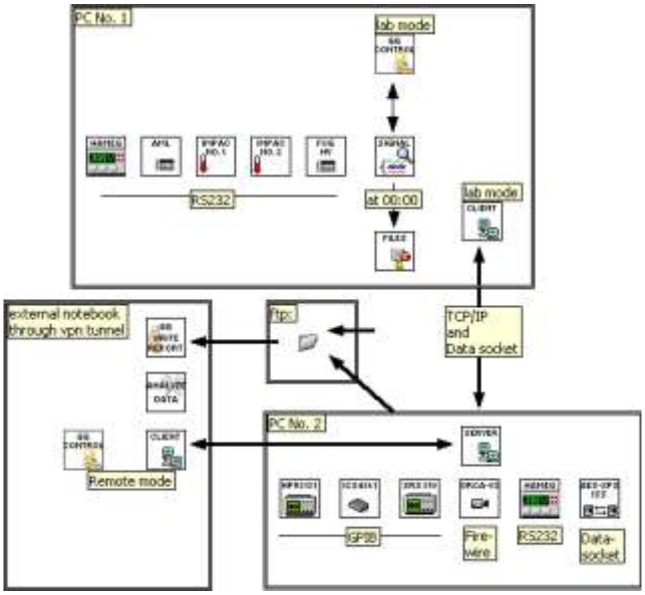


Figure 4. Main VIs running on the two lab PCs and the external portable PC. The ftp server has file directories for LabVIEW data and general control files.

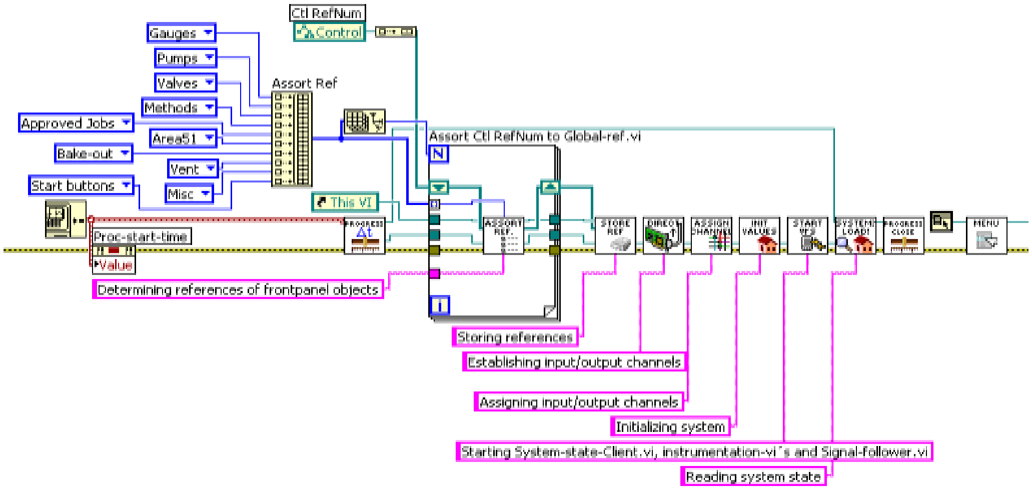


Figure 5. Block diagram of the initialization of `GG-control.vi`. During the initialization a new window appears, created by `Show-progress.vi`; its front panel shows a status bar and is closed after approx. 6 seconds by `Progress-close.vi`. For clarity, the *terminals* of the front panel objects are not indicated. Further explanations are given in the text.

The full functionality of `GG-control.vi` is obtained in connection with the second desk-top PC on which a *data socket server* and a server-VI is running (figure 4). SubVIs for the initialization and operation of instruments, connected via GPIB, RS232 and fire wire interfaces to that PC, are called by `GG-control.vi` through the server-VI.

Auger, photoemission and ion scattering spectra can be collected here, also initiated by commands from `GG-control.vi`. Spectra and LEED images, that are determined with a CCD camera, are stored and transmitted to an ftp server. In addition, the *file paths* of these data are written in a login-file which is read and saved on the ftp server. From this login-file, the user can easily access and retrieve own data by means of `GG-Write-report.vi` and `Analyze-data.vi` that run on an external PC.

Next is presented the *block diagram* of `GG-control.vi` in the figures 5 and 6 which allows a closer look on how the program is working. It can be used in two quite different modes. It runs in the “lab” mode (*options field* set to listening/talking) on the PC in the laboratory, which is connected to the equipment; it runs in the “remote” mode on a remote PC (e.g. a portable computer at home). We will first explain the “lab” mode of the program. As usual, the dataflow starts at the left-hand side of the *block diagram*; the system is initialized by a number of subVIs (figure 5). The *wires* between *error-in/error-out clusters* ensure that they are sequentially processed in the right way. Then, the five loops *A*, *B1*, *B2*, *C* and *D* were executed parallel (figure 6). In `ASSORT REF.vi` all *references* of the front panel objects are determined. This is done in the following way: The *path* of the *panel* is wired to the *property node* “*Controls*[]”; this is an array whose individual elements are the *objects refnums* that are *down casted* by the *“To More Specific Class function”* to *strictly typed control refnums* that have the **identical** *owned labels* as the corresponding front panel objects. *Target classes* are the *classes* of the front panel objects. The *control refnums* are organized in different *clusters* labeled “Gauges”, “Pumps”, “Valves” and so on and collected in one *global variable* `Global-ref.vi.` By doing so, the number of *wires* between the different nodes in the block diagram is kept small in spite of the large number of *terminals* that equals approximately 500. References to the front panel objects do not need to be wired with subVIs; in the subVI the *refnum* of an object is extracted from the *cluster* in `Global-ref.vi` by means of the “*Unbundle by Names function*” and its properties are determined or changed by means of corresponding *property nodes.* In `STORE REF.VI` the *refnums* themselves are stored in an array of dimension one; in this array every *refnum* is always indexed with the same number. The array is primarily used for the programmatic control of `GG-control.vi.` The subVI `DIRECT.vi` determines the input and output character of the various I/O channels of the TTL boards. We used 32 I/O channels of the three TTL boards in the PC as input lines, the residual ones as output lines. There are several other subVIs that refer to the I/O channels of the TTL boards; `ASSIGN CHANNELS.vi` transfers the descriptions (i.e.

short names) of the channels to the *free labels* of corresponding *clusters* of all other relevant subVIs. Reasonable default values of *terminals* and *global variables*, data ranges of numeric inputs and certain style settings are configured in `INIT VALUES.vi`; in addition, each stepper motor is associated here with a *cluster* that contains its type (unipolar or bipolar), rotational direction (which may depend on the transmission ratios of the gear boxes), operation voltage, output channel of the associated relay, type of calibration (linear, logarithmic or specified by formula) and corresponding calibration parameters.

In `START VIs.vi`, other programs are started that run independently without synchronization with the top-level VI and maintain the communication with instruments, the second desktop PC, and collect all data and operations as a function of time. One of these VIs, `Signal-follower.vi` activates a file compression and a report generation-VI at midnight, apart from other operations such as alarm settings. In addition `Signal-follower.vi` supervises the function of titanium sublimation pumps. The next VI in this queue is `SYSTEM LOAD.vi`. It reads the stored system state (binary information, manipulator positions). The initialization is then finished by creating a *top-down menu* in `Menu-create.vi` that makes other VIs and subVIs accessible at the front panel of `GG-control.vi`.

The residual program consists of five *while loops* that run parallel after the initialization. Their *block diagrams* are shown in figure 6. Loop *A* operates the *top-down menu* and the *subpanel*, loops *B1* and *B2* are connected with a *queue* and a *named notifier*, respectively, and are paused until an *dequeue element* or a *notification* are received, loop *C* is a *timed loop* with a 5 s-periodicity in which various *nodes* are processed sequentially, and in loop *D* all front panel actions are organized in an *event structure*. The two subVIs `Select Menu.vi` and `Select Sub-VI.vi` of the loop *A* contain modified procedures from the LabVIEW examples list; they will not be discussed here. The other *nodes*, implemented in *A*, need to be processed more frequently as it would be the case in loop *D*; they are responsible

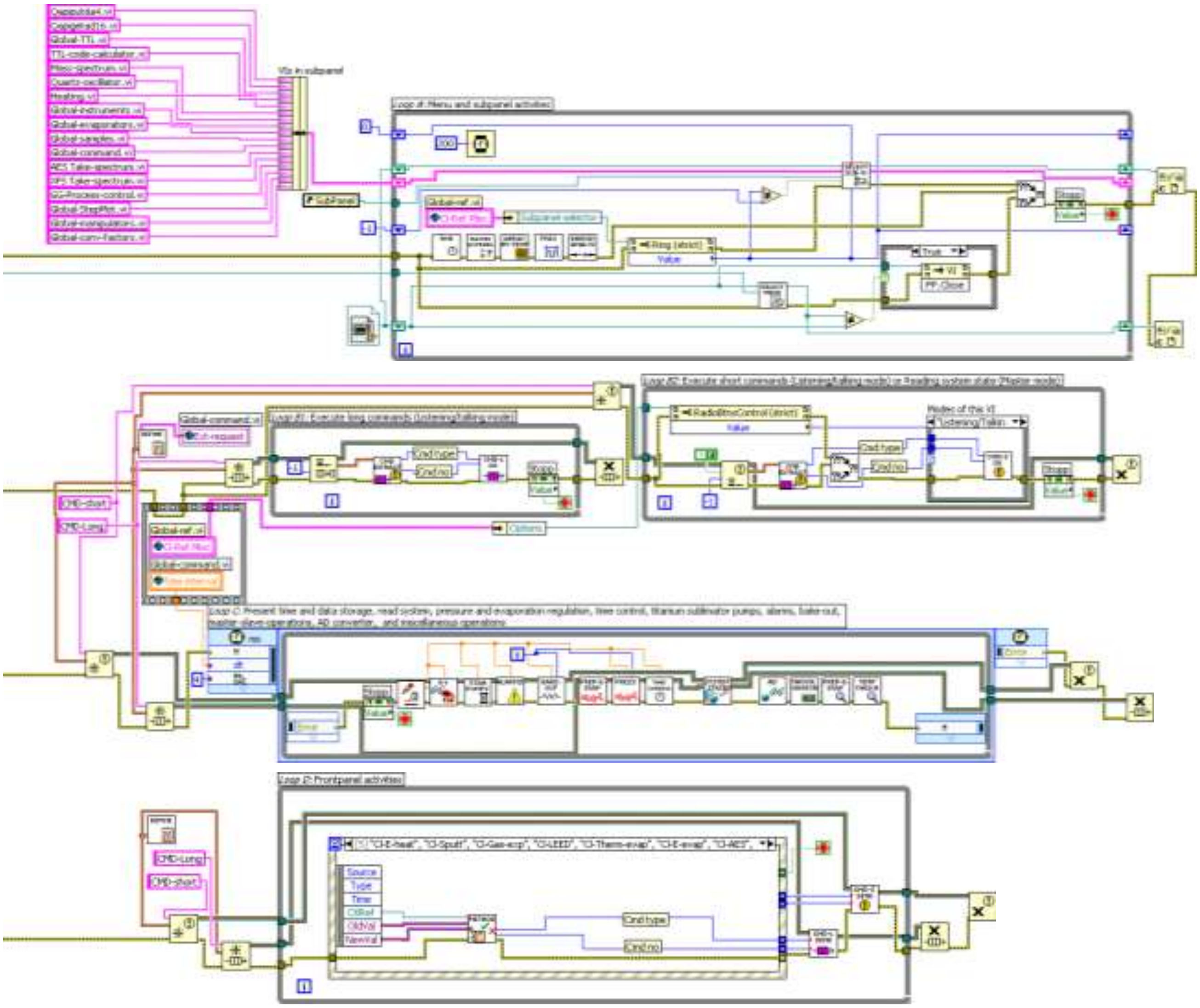

VIs in subpanel
Global-TTL.vi
TTL-code-calculator.vi
Mass-spectrum.vi
Quartz-oscillator.vi
Heating.vi
Global-instruments.vi
Global-evaporators.vi
Global-samples.vi
Global-command.vi
AES Take-spectrum.vi
XPS Take-spectrum.vi
GG-Process-control.vi
Global-StepMot.vi
Global-manipulators.vi
Global-conv-factors.vi
SubPanel
Loop A: Menu and subpanel activities
Global-ref.vi
Subpanel selector
Value
True
PF.Close
Global-command.vi
Exit-request
Loop B1: Execute long commands (Listening/talking mode)
Cmd type
Cmd no
Stopp
Value
Options
Loop B2: Execute short commands (Listening/talking mode) or Reading system state (Master mode)
RadioBtnsControl (strict)
Modes of this VI
"Listening/Talkin
CMD-short
CMD-Long
Loop C: Present time and data storage, read system, pressure and evaporation regulation, time control, titanium sublimator pumps, alarms, bake-out, master slave operations, AD converter, and miscellaneous operations
Error
Loop D: Frontpanel activities
"CI-E-heat", "CI-Sputt", "CI-Gas-exp", "CI-LEED", "CI-Therm-evap", "CI-E-evap", "CI-AES",
Source
Type
Time
CtlRef
OldVal
NewVal
Cmd type
Cmd no

Figure 6. Block diagram of the *loops A*, *B*, *C* and *D* of `GG-control.vi`. The option "Listening/Talking" in the *case structure* of *loop B* corresponds to the laboratory mode. The *event structure* in *loop D* handles 33 different *event cases*.

for a quick transfer of certain information and data to some front panel objects.

The *loops B1* and *B2* form the interface to the electronic units of our apparatus. We use the *queue function* as well as *notifications* as internal instructions; the subVI `Code-extract.vi` converts them into a type and a number. The type categorizes them into methods, regulation processes, operation of gauges, pumps or valves, bake-out, power supplies, and system operations. The numbers are associated with certain devices or actions. In *loop B1* longer instructions which are associated with methods (like electron-beam heating or sputtering, typically composed of a **series of operations**) and regulation (e.g. change of parameters of methods) are processed in the sub-VI `CMD-L do.vi`. In *loop B2* **single operations** of an individual relay are executed. This occurs in `CMD-S do.vi`. Such operations concern pumps, valves and gauges. Because they are completed in a very short time, there is no risk of a data loss and hence *notifications* are a proper choice. In `CMD-L do.vi` and the various subVIs called herein, the execution of instructions may last longer in contrast. We have chosen the *queue function* since all instructions are buffered and stored until they are read in loop *B1*. Similar to `CMD-S do.vi`, instructions in *loop B1* are also processed without interruption, for instance, by operating relays, stepper motors, DAC converters and VISA resources. In most cases, step-by-step procedures are used to transform the instructions into real physical operations; time constraints do not play a significant role in our application. In fact, the implementation of the procedures into our LabVIEW program could have been done in many ways. The TTL board would allow switching many channels at practically the same time but the step-by-step procedure is an appropriate and secure method, too, and has therefore been chosen here. An example of such a step-by-step process is shown in figure 7. It describes the key elements of the subVI for electron-beam heating; *nodes* that control the representation at the front panel are omitted here for clarity. (The representation plays, however, a role that should not be underestimated: we used the *controls* also to indicate actual values and modify their appearance during operations, see figure 3). The *refnums* of the front panel controls “E-heat-fil-volt Value” (i.e. the voltage applied to the filament) and “E-heat-HV Value” (i.e. the high voltage applied to the sample) are passed to the subVI by means of `Global-ref.VI`, in which they are categorized under a *cluster* “Cl-Ref Methods”. The physical steps include disconnection of the sample from the grounded current meter, connecting the sample with a high-voltage supply and the filament with a power supply, connecting the control input of both power supplies with two channels of the DAC converter board. The electrical connections are made with relays; finally the DAC converter is set to the calculated voltages. Conversion factors (and functions in some cases) are passed from a *cluster* of `Cl-con-factors.vi`; the number of steps, each of them lasts 100 ms, determines how quick the temperature of the sample is raised and is here represented as a constant although more complicated algorithms are used in the program to enable different rise and decay times. Most of the other processes follow this technique; relevant basic functions have been already shown in figure 1. The most complex *node* concerns the stepper motors; here the time period, during which pulses are transmitted to a certain stepper, determines its operation time and the execution of *loop B1* is kept in *while loops* that are terminated by a time condition. We can handle situations in which the stepper is simply rotating in forward or reverse direction, a feedback signal is used to control the axis´ final rotation angle, or a periodic operation is required (e.g. for pressure regulation with the leak valves). All physical operations are also explained in the *string field* “Information display” on the front panel, to allow students to understand the way how it works.

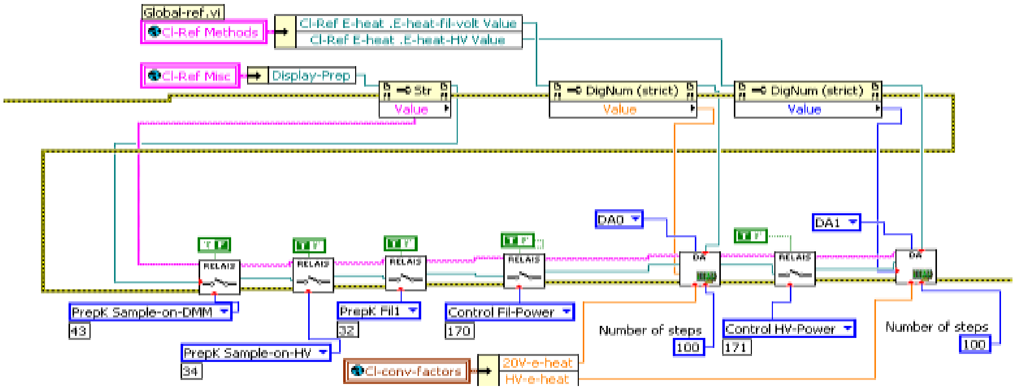


Figure 7. Block diagram for starting the electron-beam heating. It has been rearranged for clarity. The upper part shows how the *refnums* of objects of the top-level VI are extracted from `Global-ref.vi` to read the numeric values of the objects. The lower part indicates the *nodes* to convert them into real physical operations.

In the *timed loop C* a number of subVIs are invoked sequentially. Data storage by `Signal-follower.vi`, running in the background, is requested by the first one every hour. The second subVI in this queue calls a “system read”, assigned with 5-1, by the `CMD-S do.vi` in loop *B2;* thus the binary signals of the valves, pumps and alarms (e.g. cooling water flow, failure in the compressed air pipes) are acquired from the thirty-two input channels; half of them are multiplexed by a factor of four using dual 4-line to 1-line data selectors (SN74HC153) localized in the central electronics. Then three subVIs follow that control the titanium sublimation pumps, alarms, and bake-out operations.

If the electron-beam evaporation is active, the subVI `Prep-e-evap-control.vi` implements a proportional regulation of the filament voltage; the power during the evaporation is kept constant by means of this subVI. `Press-regulate.vi` keeps the pressure during gas exposure and ion sputtering constant; corresponding commands are send out to `CMD-L do.vi` in *loop B2*. `Time-control.vi` executes all operations associated with the time control function of our program. In `System-state.vi`, instructions from the external PC enter *loop C* and are converted into *queued elements* or *notifications*, depending on their nature. The residual subVIs transfer data from the AD converter to the front panel and from the thickness monitor and the power supply of the electron-beam evaporator if the corresponding components are active and hence their LED indicators are on. The last subVI generates temperature-*versus*-time or thickness-*versus*-time signals if such data are required for *charts* of other subVIs that are selected in the *subpanel* of `GG-control.vi`.

If a user clicks on a button or generates another front panel activity, *notifications* or *queued elements* are also emitted from the loop *D*. *Dialog boxes* may appear that announce messages to the user; warnings indicate forbidden operations, for example. The control routine comprises nodes that check certain conditions; an example is given in figure 8. Only if the planned action is feasible and has passed the control routine, an instruction is sent out.

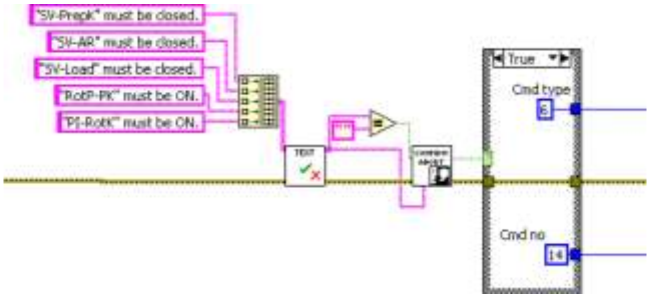
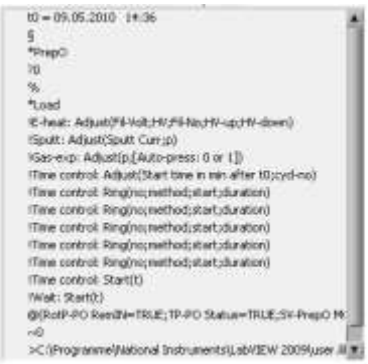


Figure 8. Concept of the control routine that check whether an operation can be performed. In `Text.vi` the *strings constants* are parsed and analyzed. If the Boolean value is true, the condition is fulfilled and an empty string appears at the output. If the Boolean value is false and hence the condition is not met, a dialog box is opened with a corresponding message. In response to the message, the numbers "Cmd type" and "Cmd no" appear at the output of the VI, defining the instruction.

Access to `GG-control.vi` is possible from an external computer by sending a *cluster* of data that addresses a front panel object and a change of its property. For example, a *value (signaling)* activates the corresponding front panel object and the corresponding action is processed like if this object is operated on the local machine.

Another subVI, called `GG-Process-control.VI`, allows us to control the top-level VI programmatically in which instructions are processed as a function of time. This makes it possible to run experiments on the vacuum system in an automated way. `GG-Process-control.vi` helps to create a *list-box* which represents a series of commands to `GG-control.vi` (Figure 9). Upon operation, the *list-box* is read line by line, and generates either a front panel activity of `GG-control.vi` or changes its front panel objects by means of property nodes "Value" or "Value (signaling)". The *list-box* always starts with setting a "t0" time, i.e. the start time of the entire job. Exclamation signs in connection with the process name, followed by the words Adjust, Start, Stop or Modify are used to control the execution of processes. The sign ">" terminates `GG-Process-control.vi`. Complicated experiments require extended *list-boxes*. They can, however, also be split into smaller partitions, each of them assigned with a brief label. These short jobs can be invoked on the *page* "Approved Jobs" on the *register card* of `GG-control.vi` in a *tree*, in which they are categorized under different names like manipulation, heating, and so on. A long lasting experiment can then be assembled and stored. The user defines a start time of this task and clicks on a *start button*. `GG-Process-control.vi` is called in the background, the start times "t0" of all jobs are omitted and the jobs are executed consecutively.



```
t0 = 09.05.2010  14:36
§
*PrepO
?0
%
*Load
!E-heat: Adjust(Fil-Volt;HV;Fil-No;HV-up;HV-down)
!Sputt: Adjust(Sputt Curr;p)
!Gas-exp: Adjust(p;[Auto-press: 0 or 1])
!Time control: Adjust(Start time in min after t0;cycl-no)
!Time control: Ring(no;method;start;duration)
!Time control: Ring(no;method;start;duration)
!Time control: Ring(no;method;start;duration)
!Time control: Ring(no;method;start;duration)
!Time control: Ring(no;method;start;duration)
!Time control: Start(t)
!Wait: Start(t)
@(RotP-PO RemIN=TRUE;TP-PO Status=TRUE;SV-PrepO M
~0
>C:\Programme\National Instruments\LabVIEW 2009\user
```

Figure 9. The *list-box* is used to run an experiment with `GG-control.vi`. §, *, ?, %, !, @, ~ are control signs. Further explanations are given in the text.

3.3 Summary

We have presented a LabVIEW 2009 package for the computer control of an ultra-high vacuum system and surface science experiments. The block diagram of the top-level VI has been presented and discussed. Apart from data acquisition and instrument control, we use the program to run such experiments automatically; the corresponding control instructions can include electron-beam heating, ion sputtering, gas exposure, electron-beam and thermal evaporation, sample movements, operation of vacuum components, low energy diffraction as well as Auger and photoelectron spectroscopy. Remote access is another benefit of the software enabling the user to monitor and control the vacuum system from a portable PC outside the laboratory.

4. ACKNOWLEDGMENTS

The authors gratefully acknowledge the assistance of M. El Achhab, M. Steger and T. Garbowski.

5. REFERENCES

[1] http://www.ni.com/labview/. [Online] National Instruments.

[2] Cautero, G., et al., Meas. Sci. Technol. 5 (1994) 1002-1011.

[3] Sumathi, S. und Surekha, P., LabVIEW based Advanced Instrumentation Systems. Springer-Verlag Berlin Heidelberg 2007.

[4] Sudhaker, Samuel R D und Jagadeesh, Chandra A.P., International Journal of Computer Applications 10 (2010) 103–110.

[5] Mütterlein, B., Handbuch für die Programmierung mit LabVIEW: Spektrum Akademischer Verlag, Heidelberg 2009