SINGLE2010
Installation Guide

RJ ANGEL

rossangelsoftware@gmail.com

http://www.rossangel.com

INTRODUCTION

SINGLE is a program maintained and developed by RJ Angel to carry out the calculations necessary for controlling a four-circle Eulerian-cradle diffractometer. It is developed from earlier versions of SINGLE written by LW Finger (Geophysical Laboratory, CIW) and RJ Angel. Much of the code is derived from software written earlier by LW Finger.

No statement or information in this document should be taken as contradicting instructions and information in manuals and documents from the manufacturer of your diffractometer. No responsibility is taken for any mental or physical damage arising from information provided in this document.

The software is currently supplied on a non-commercial basis. The authors would therefore appreciate acknowledgement of the use of the code being made in all publications that make use of data collected or processed using SINGLE by reference to: Angel RJ, Finger LW (2011) SINGLE: a program to control single-crystal diffractometers. *J. Appl. Cryst.*, 44:247-251.

1. VERSION NOTES

Single2010 can run under Windows-98, Windows-XP, and 32-bit and 64-bit Windows-7. It has not been used under Vista, or NT. It does not work under Windows-2000.

The current version of the code is **Single2010**. It was originally established in summer 2010 as the final version of Single2008 which was developed over the previous two years. Single2008 was created from the previous release Single04 in Spring 2008, and installed and tested in Bayreuth and Virginia Tech Crystallography Laboratory (VTX) in the period May-June 2008, further modified and tested at Padua, BGI, Heidelberg and VTX in fall 2008, and VTX, BGI, Padua in Summer 2009. The P4 interface was developed in Innsbruck in September 2009-May 2010, and the interface for the SMC9300 was developed at the same time in Bayreuth.

Significant changes in **Single2010** from **Single04**, included:

- Restructuring of code to separate diffractometer-dependent code from the communications subroutines. Standardisation of the diffractometer-dependent code so allows easier implementation of new diffractometer interfaces.
- Implementation of interfaces for Stoe Stadi4, Siemens P4 and Huber SMC9300.
- Support for both Windows-98 and Windows-XP communications via serial ports or USB/Serial adaptors.
- More traps of communication errors, and more recovery routines.
- Support for choice of line terminators via the *difprof.dat* file.
- Reduction of interface calls by keeping a log of motor positions.
- Generalisation of motor names, to make internal motor numbers always to correspond to 1 = 2θ, 2 = ω(bisecting), 3 = χ, 4 = ϕ, 5 = vertical slits, 6 = horizontal slits. Local translation of motor names to these numbers is via entries in the *difprof.dat* file.
- Introduction of motor numbers 7 = detector drive, 8 = sample height.
- If motorised slits are not present, the slit settings can be loaded with the **ldmt** command.
- Introduction of a *STOP* button.
- Introduction of ZREM command to do zref on multiple crystals from one command, and CEN8 command to do 8-position centering on one reflection.
- Introduction of the INL command to input reflection indices in to the list by Laue group.
- Re-introduction of SRCH and CONE searches.
- Program is distributed as one *exe* file. Choice of diffractometer done via *difprof.dat*

Further changes to commands in **Single2010** from May 2010 to March 2012 include:

- Re-introduction of PHOT and CALP to do rotation photographs.
- Introduction of UBIM and UBEX commands to import and export UB matrices. These replace the previous OMX command.
- Introduction of STEP command to perform step scan of any diffractometer circle with the shutter closed.
- Intensity data collection commands have been renamed:
    - INTA to DC S
    - INTL to DC L
    - INTS to DC I
    - PSI to DC P
- The CALC command can now report 8 positions.
- The SORT command can now sort on all items in the list, not just 2theta
- Introduction of the MACR command to run a list of commands stored in a text file.
- Introduction of the FREF command to initiate a search for circle reference points on the Stadi4.

- The refinement of trigonal unit-cells was added to **clsq** and **tlsq**.
- The telescope message is now controller-dependent so that it does not appear on the P4.
- More information is provided after a failure in centering.
- Additional parameters have been introduced for working with high-temperature furnaces.
- More information about the program version and configuration is written to the user log file when the program starts.

For details of the new and changed commands, please read the section on COMMANDS in the user manual.

There have also been a number of developments in the code, and changes to controllers, in order to improve the precision of the unit-cell parameters determined by the ZREF procedure:

- When motor steps are not exact multiples of 0.001°, correct operation requires that the Huber SMC9300 motor controller is set to report positions to 4 decimal places. See the section in this manual on the SMC9300 controller.
- The fitting of omega peak profiles collected on instruments fitted with a monochromator was corrected.
- The angular positions are now written in to the *rfl* file with additional decimal places to accommodate motor steps that are not exact multiples of 0.001°.

In combination with changes to the separate Zref and WinIntegrStp programs, these code changes and changes to the SMC9300 controller configuration means that the unit-cells obtained by fitting data with these programs is identical to the results from LSQ and CLSQ in Single.

2. NEW INSTALLATION

2.1. The normal distribution consists of:

- One executable, *single_*.exe*
- One or more diffractometer profile files; *difprof_*.dat*
- several *dll* files
- *gino.con*
- *pdf* files of manuals

2.2. Copy the exe file to a directory such as C:\SINGLE or to a directory under \My Documents.

2.3. Copy the following files to the same directory:

- the appropriate diffractometer profile file. Rename it *difprof.dat*
- all the *dll* files
- *gino.con*

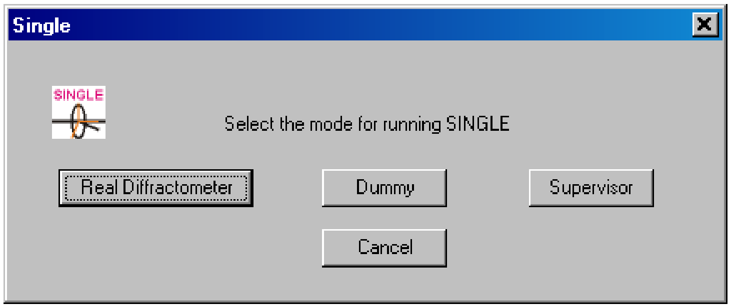
This directory should be kept for the program files only. Data files should be placed in other directories.

2.4. Copy the *pdf* files to a separate directory (e.g. \SINGLE\docs).

2.5. Create a directory for test data.

2.6. Create a shortcut to the executable on the desktop.

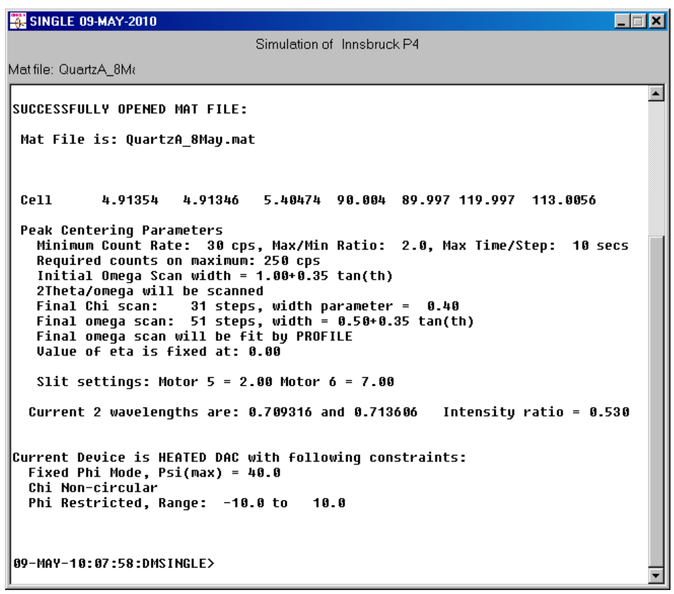
2.7. Test the installation. Start the program from the desktop. A dialog box should appear:



If this box does not appear and/or you get an error about missing *dll* files, or a missing or corrupted *con* file, then check that you have copied all of the files listed in section 2.3 to the same directory as the executable.

2.8. Click on the "Dummy" button in the dialog box. If the installation is ok, a file browser will appear. Proceed as follows:

- Use the browser to select the data directory created in 2.5.
- Type in the name for the experiment ("mat") file, or select an existing file. Do not use an extension.
- If you create a new matfile, answer "YES" to the question "do you want to create a new matfile?"
- The program should now load some default parameters and the window should look something like this:



2.9. If this window appears then you have installed the program successfully, and it can be used to do crystallographic calculations without a connection to a diffractometer. See the SINGLE users' manual for details of commands and functions. If you now want to control a real diffractometer, type "exit" at the prompt to close the program and proceed to section 4.

3. UPGRADE

3.1. If you are just upgrading the program from a previous version of SINGLE, do not overwrite the previous version of the SINGLE executable! (This allows you to use the old version if the new version does not work.)

3.2. Copy any distribution notes to the documents directory. Read them for any specific instructions. Take care not to over-write old versions of files!

3.3. Read the Version Notes at the start of this manual to see what changes have been made.

3.4. Unless the distribution notes say otherwise, you only need to copy the new SINGLE executable to a new filename in the same directory as the previous executable. A good name would be single_*date* where *date* is the date of installation. Redirect the desktop shortcut to point at the new executable and run the program. The new program will use the old *difprof.dat* file.

3.5. Do not overwrite your copy of *difprof.dat* with the distribution version unless the distribution notes say otherwise! The version of this file in the distribution is generic and **will not contain data specific for your diffractometer!**

3.6. Now proceed to section 5 to test the new version of SINGLE.

4. COMMUNICATIONS

4.1. Your diffractometer control computer must have either a serial com port, or a USB to serial-port adaptor attached to a USB port, for communications with the diffractometer. If you use a USB port with an external adaptor, then the serial line must appear as a COM port to the Windows system, and it must always retain the same com port number every time you boot up the system. You can check which com ports are available as follows (in Windows-XP):

- Select Control Panel, then System.
- Select the Hardware tab, then press the Device Manager button.
- Expand the list for “Ports (LPT and COM)”.
- The list should include a COM port that is your connection to the diffractometer controller. Note the number of this port.

4.2. Make up a serial cable to connect your diffractometer controller and PC. See the specific instructions at the end of this document for your type of diffractometer.

4.3. Consult the manufacturers manual for your diffractometer controller to determine the communications protocols (baud rate, bits, line terminators, etc). Edit the *difprof.dat* file, line COMPORT to match the settings of your controller (see section 5).

4.4. Set the first entry of the COMPORT line to match your operating system (see section 5).

4.5. Set the controller type in the CONTROL line to match your diffractometer controller (see section 5).

4.6. Consult the specific section at the end of this manual for details of hardware and software settings used for your type of diffractometer controller.

5. DIFFRACTOMETER PROFILE FILE

The diffractometer profile file named *difprof.dat* contains two types of information:

- Diffractometer characteristics (motor settings, hardware limits, configurations).
- Default scan and data collection parameters.

First, you will have to modify the *difprof.dat* file to match your hardware and the settings on your diffractometer controller. These entries in the file that you need to change are listed under section 5.1. Later you can modify the default scan and data collection parameters listed under section 5.2. For details specific to your type of diffractometer controller, consult the specific section at the end of this manual.

The *difprof.dat* file contains a number of lines, which may be present in any order. Each line begins with a 6 letter code word, or 6 blank spaces. All lines with 6 blank spaces at the beginning are ignored; they can be used to insert comments. The allowed code words (in bold) with examples from the Bayreuth Huber on an SMC9000 controller are:

5.1 Diffractometer characteristics.

DIFNAM Name of the diffractometer. This will appear on the GUI.

CONFIG -1,-1,-1,-1,1,0,0,1,0,0,500,1
parity(4),ndcp,nspcchi,i_mono,dslit,hslit,ifilt,shtdly,ishut_check
- Interface and diffractometer configuration:

Parity(4)	Parities (+/-1) of the 4 circles, relative to a Busing-Levy diffractometer
ndcp	ndcp=1 for a diffractometer interface which treats ω as absolute, 0 for an interface that works with ω relative (e.g. a Picker). At this time all supported instruments work with ndcp=1
nspcchi	Compensation on ϕ for χ drives (not used at this time)
i_mono	1 if monochromator installed, 0 otherwise
dslit	1 if motor driven slits available, 0 otherwise
hslit	1 if half-slits available, 0 otherwise
ifilt	1 if filter-wheel available, 0 otherwise
shtdly	Shutter delay in msecs (delay before next motor drive or scan)
ishut_check	1 if shutter status feedback is enabled, 0 otherwise

CONTROL 1
- Defines the type of motor controller/diffractometer interface:

1 – SMC9000 Huber
2 – Stadi4, Stoe
3 – AMS motor controllers plus Keithley counter card (as at VTX)
4 – Not used
5 – Not used
6 – Siemens P4
7 – SMC9300 Huber

GONCON 470.0,370.0
- Diffractometer size. Set them initially to the physical distances on the diffractometer. The exact values do not matter to within 5mm. Then see the section on Fine Alignment for instructions on how to determine more precise values.

U0	source to crystal distance in millimetres
V0	crystal to detector distance in millimetres

COMPORT 2,1,9600,0,8,0,3,1,0,0

- Communications parameters (*all integers*). **These must match the settings in the interface**. See the diffractometer manual, and the diffractometer-specific notes at the end of this document, for details. In order the parameters are:

icommtype	1=use Win API (Windows-XP, Windows-7), 2=use sport routines (Windows-98).
iport	Port number on PC. iport=1 means COM1.
baud	Baud rate for the com line.
parity	0=none, 1=odd, 2=even, 3=mark, 4=space (normally 0).
databits	Number of databits on the comport (normally 8).
stopbits	The stop bits for the port (0, 1, 2 = 1, 1.5, 2 respectively). Normally set 0 to mean 1 stop bit.
interm	Comm line terminator on strings sent back from the diffractometer.
outterm	Comm line terminator attached to strings sent from the PC to the diffractometer. Both of these variables have the following meaning: 1=<CR>; 2=<LF>; 3=<CR><LF>
itimeout	Not used for Win98, ever. Not used on Stadi4 or P4 controllers, ever. On other controllers and systems, if itimeout is zero, no timeouts are set. If it is non-zero, this is used as the maximum read timeout interval, in msecs. itimeout=1000 seems to work for the Huber SMC9000 controller under XP, but start with this parameter set to zero.
isenddelay	Delay in msecs between sending a command to comm. port, and issuing the read request to the comm port. If zero, there is no delay. If isenddelay is positive, then it is used as the delay in msecs. Not yet implemented for interfaces other than Stadi4, as no delay has been used in the past for Huber controllers or the P4.

MOTORn

- Motor parameters, including motor names, default slew and ramp speeds, and circle limits. The exact meaning of the entries depends on the diffractometer. See diffractometer-specific sections at the end of this manual. The interpretation is done in the device-dependent subroutine check_motors.

The circle limits should be set in the *difprof.dat* file to be slightly less than the position at which the physical limit switches are activated. Once you have got the program talking to the diffractometer, check these values carefully! The values for the ω limits in the file are absolute ω.

MONOC 12.2,90.0,0

- Monochromator information. Not needed if no monochromator (imono=0 on config card). If monochromator is present, the numbers are:

- Monochromator 2θ value.
- Monochromator dihedral angle (0 if monochromator diffraction plane is parallel to sample plane, 90 if perpendicular).
- Iparallel. Only needed if dihedral angle=0. Then 1 means dispersions of crystal and monochromator add on positive 2θ side of diffractometer, -1 if dispersions add on negative side.

WAVEL 2,0.709316,0.713606,0.52

NWAVE,WAVE1,WAVE2,WRATIO

NWAVE is the number of wavelengths in the incident beam, WAVE1 and WAVE2 their wavelengths, and WRATIO the intensity ratio I(WAVE2)/I(WAVE1). Determine

WRATIO from fitting scans of well-resolved high-angle strong reflections.

NOSHUT
- Suppresses all calls (open, shut, and status) to the shutter. Useful for testing and debugging. Not useful for doing measurements!

VIEW 30.,0.,0.,50.
- Preset 2θ, ω, χ, and ϕ for the view position. Adjust these so that the slides of the goniometer head are parallel and perpendicular to the axis of the telescope at the view position. Make sure that these positions do not result in collision, as the drive is not checked against the software limits.

PARK 0.,0.,90.,0.,2.,9.
- Preset 2θ, ω, χ, ϕ and slits positions for the park position to which the diffractometer is driven on normal program termination. Make sure that these positions do not result in collision, as the drive is not checked against the software limits.

PHOTO 0.,0.,0.,0.,300.
- Preset 2θ, ω, χ, and ϕ for the photo position, and the crystal-to-film distance (*not* the distance to the point detector). Make sure that these positions do not result in collision, as the drive is not checked against the software limits.

OFFSET 0.,0.,0.,0.,0.,0.,0.
- Offsets of diffraction-defined zero positions of the circles from the zero positions of the motor controllers. The diffractometer is driven to a position of the angle *plus* the offset. Thus, if the direct beam hits the detector when the controller value for 2θ value is $+10^{o}$ the offset for 2θ should be $+10^{o}$. When you use the software to then drive to $+15^{o}$ 2θ, the motor controller will drive to $+25^{o}$. The ω offset is specified in absolute ω.

UBTRAN 0.,1.,0.,-1.,0.,0.,0.,0.,1.0,name
- Preloaded matrices for transforming UB matrices for import from, and export to other diffractometer control systems. These matrices may vary from installation to installation because of the way in which the goniometer head mount is aligned on this diffractometer and on the other diffractometer, in addition to the change in Cartesian axis choice between Single and the other system. The matrix is given as that required for importing the UB in to Single. There are nine entries, the first three being the first line of the matrix. The string is the identifying name that will appear for commands UBIM and UBEX. Up to 9 matrices may be given on separate lines, for example:
UBTRAN 0.,1.4098,0.,-1.4098,0.,0.,0.,1.4098,Crysalis-Mo - Single
UBTRAN 0.,-1.4098,0.,1.4098,0.,0.,0.,1.4098,Stadi CCD - Single/Stadi4
UBTRAN 0.,1.,0.,-1.,0.,0.,0.,0.,-1.,Xscans - Single
- These matrices may vary from installation to installation because of the way in which the goniometer head mount is aligned on this diffractometer and on the other diffractometer. The matrices should be tested and changed as appropriate.

5.2. Default scan parameters

SCANS1 30.0,2.0,10.,250.,1.0,0.35,2.0,9.0
- Default parameters for centering reflections, only used when creating a new mat file. In order:

MINCT Minimum count on a centering scan for a reflection to be centered
MAXMIN The required ratio of the max to min counts on a centering scan
TMAX Max allowed time per step
maxct Required counts at max
OMW_I1,OMW_I2 Width of the first ω scan during the centering procedure is OMW_I1 + OMW_I2*tan(θ)
M5slit Width of the vertical detector slit for zref
M6slit Width of the horizontal detector slit for zref

SCANS2 31,51,0,0,0,0.2,0.4,0.35,0.0

- More default parameters for centering reflections, only used when creating a new mat file. In order:
N_CHI,N_OMEGA,IBAD,FIXTHETA,FIXETA,CHIW,OMW_F1,OMW_F2,ETAVAL

Final χ scans in centering are performed with N_CHI steps, and a width given by CHIW*sqrt(slit5width/sin(θ)).

Final ω scans in centering are performed with N_OMEGA steps, and a width OMW_F1+OMW_F2*tan(θ).

If ibad=0 then the final ω scans are fitted with a Pseudo-Voigt function with ETAVAL as the default value of the mixing parameter (ETAVAL=0 is Gaussian, ETAVAL=1 is Lorentzian). The value of eta is fixed in the fitting routine if fixeta=0, and refined if fixeta=1.

If ibad=1 then the final ω scans are fitted with a parabolic function instead of a Pseudo-Voigt. Recommended for crystals with very poor peak profiles.

If fixtheta=1, then the centering procedure does not use 2θ−ω scans. This is sometimes useful for crystals with poor peak shapes.

SGROUP 0 P 1

- Space group for data collection. The leading integer is the lattice type number, 0=P.

DSCAN1 1.00 0.35

- The scan width for data collection. The first number is the basic width and the second term is the dispersion term. The example will set a width of 1.0+0.35tanθ. The width is either ω or 2θ, depending on the scan type specified in the DSCAN2 line.

DSCAN2 1.000 0.000 0.020 5.000 0.000 10.00 1.000 0.000 0.000 0.000 0. 0.

- Data collection parameters, in order:
  - Scan type, =1 for ω scan, =2 for ω/2θ
  - Number of psi positions collected for each reflections
  - Step size for scan
  - Time per step (secs) for initial scan
  - Mode: 0 for single scan, 1 for constant precision
  - Required precision I/sigI
  - Maximum no. repeat scans in constant precision mode
  - Space group absence flag: =1 collect only sp gp allowed, =2 collect only sp gp absent, =0 ignore sp gp, collect all
  - Increment in psi for psi scans

- Reorientation flag; =0 no reorientation, =1 otherwise (not currently used)
- Minimum I/sigI to allow rescan in constant precision mode
- = 0 for no psi scans, =1 for psi scans

HKLORD 0 1 0 1 0 1 1 2 3
- Limits and sequence for indices.

ALLSTD 2 100
- Number of standards and frequency (here 2 standards every 100 reflections).

HKLST 1 1 0 1 1
- Standard reflection; standard number followed by *hkl* and use flag.

SHELL 1,0.,60.,1
- Shell information; shell number followed by 2θ limits (min, max) to shell, and use flag.

6. COMMUNICATIONS TEST

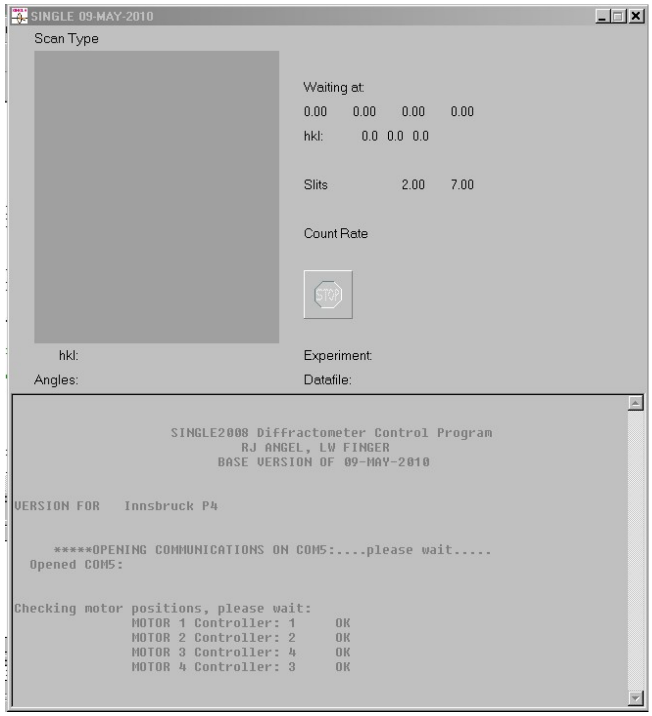
6.1. Before you try to drive the diffractometer, run through this checklist:

- The SINGLE program works in "dummy" mode (sections 2.7-2.9).
- The cable is properly connected between the motor controller and the com port.
- The com port and the com port parameters are correctly written in to the diffractometer profile file.
- The motor controller communications parameters are set to the same values as those in the *difprof.dat* file. See the controller-specific information at the end of this manual.

6.2. Start the SINGLE program from the shortcut on the desktop.

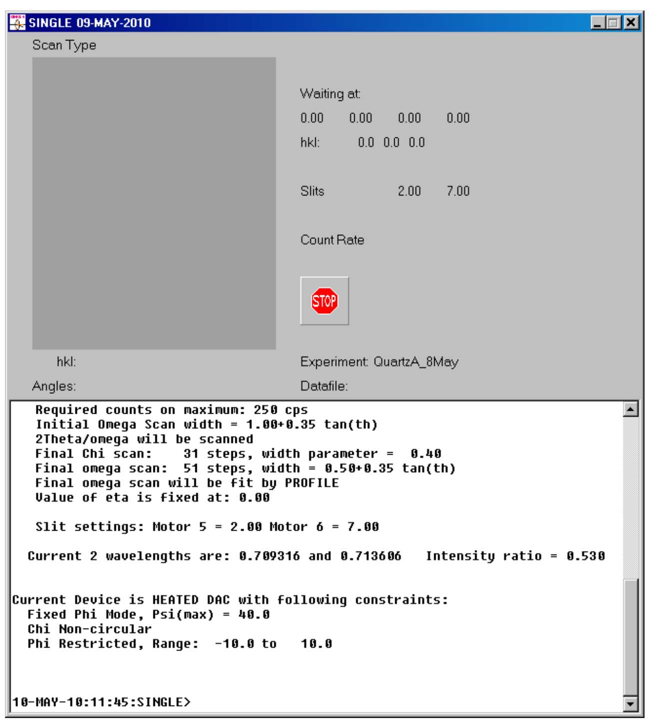
6.3. In the dialog box, select "Real Diffractometer". If the installation is ok the main window will appear. In the scrolling region of the window, the following messages will appear in order:

- The Version information.
- "OPENING COMMUNICATIONS ON COM**n:** ....please wait**….**"
  - check that **n** is the com port number you want
  - if the com port cannot be opened, or comms stall, then you will either get an error message or nothing will happen. Close the program and fix the problem – it is probably a parameter set incorrectly in the comport line of *difprof.dat*
  - if successful you will see "Opened COM**n**:" and it will proceed to...
- "Checking motor positions, please wait:"
  - For each motor you should see either "OK"
  - or a message to say that the stored position in the controller does not match the park position
  - or some other error message. Correct the problem and restart the program.



6.4. Then the motor positions will be updated on the display, and a file browser will appear. Proceed as follows:

- Use the browser to select your data directory.
- Type in the name for the experiment ("mat") file. Do not use an extension.
- Answer "YES" to the question "do you want to create a new matfile?".
- The program should now load some default parameters and the window should look something like this:



6.5. The program is now in communication with the diffractometer controller.

7. TESTING

7.1. It is assumed that the diffractometer is at least approximately aligned so that Xrays hit the crystal, the diffractometer axes intersect properly, etc. If not, a coarse diffractometer alignment should be performed according to the instructions from the manufacturer.

7.2. Now do the following, carefully to determine whether the program is working correctly:

- mot command (test ϕ first! Make sure that the physical movement, the position on the controller, and the display on the SINGLE GUI agree)
- ldmt command
- open/close shutter: check error handling when enclosure door open
- drives: check error handling when limit switch is activated during a drive (after doing this you will have to press the "reset" button on the controller and may have to reinitialise the controller. See local instructions)
- Carefully (I mean it) check that the motor limits in *difprof.dat* are correct. If they are, then you should be able to drive the diffractometer to close to the limit switches, but not on to them. Double, double check for collisions not protected by limit switches, such as the detector arm on to the χ circle, and the ϕ cradle on to either the collimator or the detector arm. If you need to change the limits you must first close the program, then edit the *difprof.dat* file, and restart the program
- count
- prof (all five options for motor scans)

7.3. If all of these facilities work then try, in order, the following centering routines for one strong low-angle reflection. Time the process:

  - cent
  - cntr
  - zref

You may have to adjust the scan parameters (use **set center**) to get the best results. As a guide, the motor 5 slit should be adjusted so as to get symmetric and close to Gaussian peak shapes on the final χ scan of centering. The final ω scan should be of a width so that the peak occupies about 1/3-1/2 of the total scan width. Once you have good parameters, edit them into the *difprof.dat* file.

8. FINE ALIGNMENT

8.1. Use a gem-quality crystal with perfect peak profiles. Spend the time to get this as perfectly-centered as possible! Start zref's on low-angle strong reflections. Adjust the scan parameters (see previous section) to get the most accurate centering. If the zref's cannot be completed due to the reflections not being found, then it is likely that the circle zeroes are incorrect:

  - Failure on the 2nd reflection indicates 2θ or ω zero error.
  - Failure on the 5th reflection indicates a χ-zero error.

In any of these cases, center the equivalent reflections by hand with cent, and try to adjust the circle zeroes (with ldmt) so that zref completes. It is sometimes possible to get zref to complete on reflections with low χ values, when it fails for reflections with higher χ-values, or vice-versa.

8.2. Repeat zref's on strong low-angle reflections. Reset the circle zeroes and adjust the crystal position to correct the reported errors (see details in the user manual for zref for the sense of these corrections). Adjust either the tube height or the detector height (as appropriate or possible for your diffractometer) so as to reduce deltaH to zero (within the reproducibility of the values). Repeat zrefs until all the circle zero errors and deltaH are zero within uncertainties.

8.3. Do a zref on a set of reflections from your standard crystal and examine the resulting scans with the WinIntegrStp program. You should obtain good unit-cell parameters, and there should be no significant variation in the delta-d/d plot. If there is variation, then you may need to adjust either the $\alpha 1/\alpha 2$ intensity ratio and/or the profile parameter eta. Adjust these with WinIntegrStp until you minimise the slope and scatter in the delta-d/d plot. Once you have good values, edit them back into *difprof.dat* so that they are the default values for all new experiments, create a new matfile so that the new parameters are used, and then repeat the measurement to check the parameters are correct.

8.4. Typical esd's for the unit-cell volumes of gem-quality crystals are 1 part in 20,000 for the volume and 1 part in 50-60,000 in the unit-cell edges and 0.002° in the unit-cell angles. Unconstrained unit-cell angles of high-symmetry crystals should conform to the symmetry within 1-2 esd's.

8.5. Determine the effective goniometer size for the correct calculation of crystal offsets from the zref measurements. Start with the crystal as well-centered as is possible. Find a strong reflection at $\chi \sim 0$ (within 5°).

- Determine the effective crystal-source distance:
    - Perform a zref. Note the crystal offsets.
    - Displace the crystal a known distance (e.g. 100 microns) along +X as measured with your telescope system. Repeat the zref on strong reflection at $\chi \sim 0$ and note the crystal offsets.
    - Displace the crystal the same distance to the other side of the centered position (i.e. along –X). Repeat the zref and note the crystal offsets.
    - Recenter the crystal along X, and repeat the measurements with the crystal displace in turn along +Y and –Y, noting the offsets in each case.
    - Average the magnitudes of the 4 crystal offsets along the axes. Call this average value Δ. Compare it to the physical offset of the crystal that you measured with the telescope (100 micron = 0.1mm). Calculate the ratio $0.1/\Delta = R$.
    - Close Single. Open *difprof.dat* in Notepad. Note the *first* value on the GONCON line. Multiply this number by the ratio *R*, and replace the number with the new value. Save *difprof.dat*.
    - Displace the crystal again by 100 microns along +X, do a zref on the strong reflection at $\chi \sim 0$. Check that the crystal offset along X reported by zref is +0.10 within the expected uncertainty. If not, close Single, adjust the *first* value on the GONCON line in *difprof.dat*, save the file, and repeat the zref.
    - When you are sure you are getting reproducible and correct values for the X and Y offsets from the reflection at $\chi \sim 0$, recenter the crystal on X and Y.
- Determine the effective crystal-detector distance:
    - Now offset the crystal vertically (+Z) by 100 micron, do zref on the reflection at $\chi \sim 0$ and note the offset. Repeat for a displacement of 100 micron down (-Z) from the center.
    - Compare the Z offsets reported by Single to the known offset. If they differ, adjust the *second* value on the GONCON line in *difprof.dat*; if the reported offset is smaller than the offset measured by the telescope, increase the value.
    - Repeat the measurements with the Z offsets, and adjust the value in *difprof.dat* until the values reported by Single agree with the known offsets.

- Check the final parameters:
    - Offset the crystal by a known amount (e.g. 50 micron) in two axial directions (as viewed in the telescope). Perform zref on a complete list of reflections and check that the reported offset values are consistent and reasonable.

SMC9000 HUBER CONTROLLERS

Communications

Make up an RS232 cable to connect a COM port on your PC to the serial port on the SMC9000. The socket on the SMC9000 accepts a 25-way D-plug. The serial line on your PC may be a 9-way or a 25 way.

If no com port is available, use an external USB-to-serial convertor. See section 4 of this manual.

Wiring:
At PC end of the cable short together CTS, DSR and DTR. On a 25-way plug these are pins 5,6, and 20. On a 9-way plug these are pins 8,6, and 4.

At Huber end of the cable short together pins 5,8, 25.

Connect through cable:	PC end pin	2	to Huber end pin	3
		3		2
		7		7

The normal configuration for the communications on the SMC9000 controller are:

9600 baud
8 data bits
1 stop bit
parity none
terminator CR/LF

Edit the *difprof.dat* file and change the COMPORT line to read (for using port 1, usually com1):

COMPORT *N*,1,1200,0,8,0,3,1

The first number indicates the type of serial port drivers. If you are using Windows-98, then *N*=2. For Windows-XP or Windows-7, use *N*=1. The next digit is the port number, so change that if necessary. If you experience a lot of comm errors, reset the speed in both *difprof.dat* and the controller to 1200 baud as the slower communications rate is more reliable for debugging problems. Once you are sure things are working, then you can exit the program and reset the SMC9000 controller to a higher baud rate and the value on this COMPORT line to the same value. See section 5.1 for the meaning of the other numbers.

Shutter control

The shutter will be controlled through the additional digital I/O port on the Huber. The user needs to connect this I/O port via the 37-way D-connector to a separate box that drives the shutter and also provides the safety interlocks between the shutter operation and the doors of the radiation enclosure.

To open the shutter the SINGLE software sends “IO1” to the SMC9000.
To close the shutter the SINGLE software sends “IO0” to the SMC9000.

These commands set OUT0 (Pin 10) to low or high. By wiring a circuit through Pin 34 (+12V) a +12V signal can be sent to the shutter box to open the shutter. When IO0 is sent, the circuit is broken.

If feedback on the shutter status from the shutter box or interlock box is available, it should open and close a circuit between pins pin 1 (IN0) and pin 20 (IN0A). This should be wired so that the command ?IO to the SMC9000 returns a “1” when the shutter is open, or a “0” when the shutter is shut. See the SMC9000 manual for details of signal levels needed to obtain this result.

Motor parameters

Each motor of the controller must be described by a line in the *difprof.dat* file that starts with the characters **MOTOR*n***, where ***n*** corresponds to a motor number in SINGLE (1 = 2θ, 2 = ω(bisecting), 3 = χ, 4 = ϕ, 5 = vertical slits, 6 = horizontal slits).

For the Huber SMC9000 interface the parameters are (in order):

Motor number in the controller…1 character preceeding a comma.
Steps/degree
Slew Rate (in Hz)
Ramp rate to slewing speed
Med speed rate (in Hz)
Ramp rate to medium speed
Minimum circle limit
Maximum circle limit

For ω, circle limits are given in absolute ω, and the last number is the minimum angle of inclination between the 2θ arm and the plane of the χ circle.

For motorised slits, only the stpdg (interpreted as step per mm) and the limits are used.

Motor drive instructions are always rounded off explicitly to the resolution of the diffractometer circle as defined in the file *difprof.dat*.

The slew rates are given in Hz which means motor steps per second. So, if you have 400 steps/deg, 400Hz will move the motor at 1deg/sec and 2000Hz at 5deg/sec.

The steps/deg in the *difprof.dat* file must match those loaded directly to the SMC9000 interface. As an example, the settings on the SMC9000 controller for the Bayreuth old Huber are:

Axis	**Type**	**GRN**	**GRD**	**Ref Off**	**Ref Freq**	**Run Freq**	**Fast freq**
1	Goniometer	1000	1	90.0	10,000	1,000	10,000
2	Goniometer	1000	1	0.0	10,000	1,000	10,000
3	Linear Table	1000	1	0.0	8,000	1,000	8,000
4	Linear Table	1000	2	0.0	10,000	1,000	10,000
5	Slit screen	400	1	0.0	1,000	200	1,000
6	Slit screen	400	1	0.0	1,000	200	1,000

Notes:
1: The gear ratios GRN and GRD define the number of motor steps per degree. GRN/GRD = 1000/1 means 1000 steps per degree, or 0.001°/step. GRN/GRD = 1000/2 is 0.002°/step, and stpdg for motor 4 in the *difprof.dat* file is therefore 500. See the Huber manual for the SMC9000 series controller for more details.
2: All motors on the BGI Huber have level for positive direction =0 and limit switch attachment =1.

The corresponding entries in the *difprof.dat* file are:

```
    NAME STPDG  SLEW   SLEW   MED   MED   MIN  MAX       SPARE
           RATE   RAMP   RATE  RAMP   LIMITS
MOTOR1 1, 1000., 3000., 20., 1000.,  10.,  -50.,   90.,   0., 0.,0.
MOTOR2 2, 1000., 1500., 20., 1000.,    5.,  -42.,  42.,  40., 0.,0.
MOTOR3 3, 1000., 5000., 20., 1000.,  10.,     0.,    0.,    0., 0.,0.
MOTOR4 4,   500., 9000., 20., 1000.,  10.,     0.,    0.,    0., 0.,0.
MOTOR5 5,   200.,      0.,    0.,       0.,    0.,   0.2,  9.5,    0., 0.,0.
MOTOR6 6,   200.,      0.,    0.,       0.,    0.,   0.2,  9.5,    0., 0.,0.
```

SIEMENS P4

Communications

Communications are via an RS232 cable from a COM port on your PC to the serial port on the controller. This should already be installed, to drive the controller with XSCANS. If no com port is available on your PC, use an external USB-to-serial adaptor. See section 4 of this manual.

The normal configuration for the communications on the P4 controller are:

9600 baud
8 data bits
1 stop bit
parity none
terminator CR

Edit the *difprof.dat* file and change the COMPORT line to read (for using port 1, usually com1):

COMPORT *N*,1,9600,0,8,0,1,1

The first number indicates the type of serial port drivers. If you are using Windows-98, then *N*=2. For Windows-XP or Windows-7, use *N*=1. The next digit is the port number, so change that if necessary. See section 5.1 for the meaning of the other numbers.

Shutter control

The shutter is controlled via the Siemens motor controller. Because the controller returns as shutter status the status in the controller (i.e. the most recent request) and not the true state of the controller, there is no point in SINGLE checking the shutter status. Therefore, ishut_check in the CONFIG line of *difprof.dat* is set to 0.

Motor parameters

Each motor of the controller must be described by a line in the *difprof.dat* file that starts with the characters **MOTOR*n***, where ***n*** corresponds to a motor number in SINGLE (1 = 2θ, 2 = ω(bisecting), 3 = χ, 4 = ϕ). There are no motorised slits on the P4.

The parameters on each motor entry are (in order):

Motor number in the controller…1 character preceeding a comma.
Steps/degree
Slew Rate (units unknown)
Unused
Unused
Unused
Minimum circle limit
Maximum circle limit

The motor numbers in the P4 motor controller do not correspond to the assignment of motor numbers in SINGLE: for the controller, motor 3 is ϕ and motor 4 is χ. This is handled by the entries in the *difprof.dat* file.

The motor encoders on the P4 give positions to a very high resolution; any reasonable lower resolution can be set in SINGLE. For the Innsbruck P4 the steps per degree were set to 500 for each motor except for ω, which was set to 1000stp/dg in order to allow for precise and consistent ω/2θ positioning. Setting other axes to resolutions of 1000stp/deg or higher results in an increase in the number of motor drive errors due to mechanical instabilities in the motors and encoders.

The units for the slew rates are unknown, but should be set to the values used by XSCANS, except for ω, which should be set to one-half of the slew-speed of 2θ. The normal slew rates for 2θ, χ and ϕ are 600. If a motor loses zero position the problem can sometimes be alleviated by reducing the slew speed.

The SINGLE program works with ω values as the deviation from bisecting. The ω values used and communicated by the P4 motor controller are absolute ω: ω(abs)= ω(bi)+θ.The user does not see this, as SINGLE only displays bisecting ω. Note that the ω scale on the diffractometer reads absolute ω values.

For ω, circle limits are given in absolute ω, and the last number in the parameters list reported above is the minimum angle of inclination between the 2θ arm and the plane of the χ circle. The circle limits for the Innsbruck P4 were set to lie inside the physical positions of the limit switches. The α parameter is set to 35^{o} after inspection of physical motions of the diffractometer. The limits for 2θ will require adjustment if the detector is moved back on the detector arm.

The corresponding entries in the *difprof.dat* file for the Innsbruck P4 are:

NAME		STPDG	SLEW RATE	SLEW RAMP	MED RATE	MED RAMP	MIN LIMITS	MAX	SPARE
MOTOR1	1,	500.,	300.,	0.,	0.,	0.,	-60.,	60.,	0., 0.,0.
MOTOR2	2,	1000.,	150.,	0.,	0.,	0.,	-50.,	50.,	35., 0.,0.
MOTOR3	4,	500.,	600.,	0.,	0.,	0.,	-179.999,	180.000,	0., 0.,0.
MOTOR4	3,	500.,	600.,	0.,	0.,	0.,	-179.999,	180.000,	0., 0.,0.

The slew speeds for 2θ and ω were reduced from 600 and 300 respectively to reduce the slipping and loss of position on the 2θ drive.

The coupling and compensating motion of ϕ when the χ axis is driven is handled internally by the P4 controller. Nspcchi is therefore not set in *difprof.dat*.

The P4 motor controller can accept any angle input, but returns motor positions within the range of 0 to 360^{o} for all motors. This is handled in SINGLE as follows:

1. For 2θ and ω values greater than 180^{o} are converted to negative by subtracting 360^{o}.
2. For χ and ϕ the limits are set to -179.999^{o} and +180.000^{o} in *difprof.dat* and values sent to, and returned by, the motor controller are put into this range.
3. These limits then mean that the χ and the ϕ axes are non-circular at all times and the diffractometer will not drive through χ = 180.

The circle parities for the P4 should be set to -1,-1,-1,1

STOE STADI4

Communications

Communications are via an RS232 cable from a COM port on your PC to the serial port on the controller. This should already be installed, to drive the controller with the Stoe software.

If no com port is available on your PC, use an external USB-to-serial adaptor. See section 4 of this manual.

The normal configuration for the communications on the Stadi4 controller are:

9600 baud
8 data bits
1 stop bit
parity none
terminator CR

Edit the *difprof.dat* file and change the COMPORT line to read (for using port 1, usually com1):

COMPORT *N*,1,9600,0,8,0,1,1

The first number indicates the type of serial port drivers. If you are using Windows-98, then *N*=2. For Windows-XP or Windows-7, use *N*=1. The next digit is the port number, so change that if necessary. See section 5.1 for the meaning of the other numbers.

Shutter control

The shutter is controlled via the Stoe motor controller. There is no feedback from the controller on the true shutter status, so ishut_check in the CONFIG line of *difprof.dat* is set to 0.

Motor parameters

Each motor of the controller must be described by a line in the *difprof.dat* file that starts with the characters **MOTOR*n***, where ***n*** corresponds to a motor number in SINGLE (1 = 2θ, 2 = ω(bisecting), 3 = χ, 4 = ϕ). There are no motorised slits on the Stadi4.

The parameters on each motor entry are (in order):

Motor number in the controller…1 character preceeding a comma.
Steps/degree
Unused
Unused
Unused
Unused
Minimum circle limit
Maximum circle limit

The only motor parameters in *difprof.dat* are the steps/degree and the circle limits.

For the Padua Stadi4 all of the motors were set to 400 steps per degree. All ramp and slew speeds are stored and handled internally by the Stoe motor controller, and cannot be rewritten by a SINGLE command.

The circle limits have been set to lie inside the physical positions of the limit switches. For ω, circle limits are given in absolute ω. Although the 2θ will not hit the χ circle, the α parameter is set to 20^{o} to prevent driving to positions at which the detector is obscured by the χ cradle. The limits for 2θ will require adjustment if the detector is moved back on the detector arm.

The program works with ω values as the deviation from bisecting. The ω values used and displayed by the Stoe motor controller are absolute ω: $\omega(abs)= \omega(bi)+\theta$.

The Stoe motor controller limits the values of the χ axis to $-360^{o} \leq\phi\leq 360^{o}$. Therefore these limits must be set for motor 3 in *difprof.dat* to allow SINGLE to run properly. These limits then mean that the χ axis is non-circular at all times.

The Stoe motor controller limits the values of the ϕ axis to $-180^{o} \leq\phi\leq 180^{o}$ (the init_motors routine sends a ‘P0’ to the interface). Therefore these limits must be set for motor 4 in *difprof.dat* to allow SINGLE to run properly. These limits then mean that the ϕ axis is non-circular at all times.

The corresponding entries in the *difprof.dat* file for the Padua Stadi4 are:

```
    NAME STPDG  SLEW   SLEW   MED   MED   MIN  MAX      SPARE
          RATE   RAMP   RATE  RAMP   LIMITS
MOTOR1        1,       400.,   0.,      0.,      0.,      0.,      -80.,    65.,      0., 0.,0.
MOTOR2        2,       400.,   0.,      0.,      0.,      0.,      -60.,    60.,      20., 0.,0.
MOTOR3        4,       400.,   0.,      0.,      0.,      0.,-360.000,360.000,      0., 0.,0.
MOTOR4        3,       400.,   0.,      0.,      0.,      0.,-180.001,180.001,      0., 0.,0.
```

The coupling and compensating motion of ϕ when the χ axis is driven is handled internally by the Stadi4 controller, and the SINGLE program just sends values for ϕ and χ as normal. Nspcchi is therefore not set in *difprof.dat*.

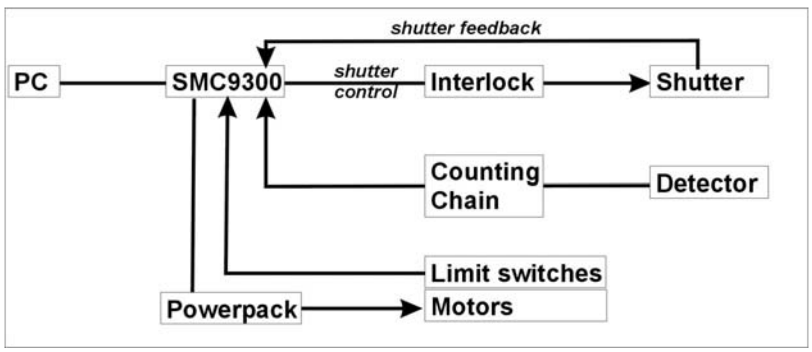
The circle parities for the Stadi4 should be set to 1,1,-1,1.

SMC9300 HUBER CONTROLLERS

Basic Installation instructions

Cable up the system:

- Huber powerpack to motors
- Huber SMC9300 to diffractometer limit switches, or insert blanking plugs in to SMC9300. The standard cable from Huber has one connector on the diffractometer end, and two connectors (one to SMC9300, one for power pack) on the other.
- Connect the power pack and SMC9300 with cable provided.
- Connect the earth terminals on the back of the SMC9300 and power pack.
- Connect the signal output from the counting chain to the counter card (BNC connector) on the SMC9300



Power up the SMC9300:

- Check and note all of the motor parameters. Change if required (see below).
- Change χ and ϕ to linear drives, not circles.
- Verify that you can drive all of the diffractometer motors manually. Be careful not to drive the slits to their limits.
- Check that the limit switches all work and are correctly placed to prevent collisions. Do not drive the slits in to the limit switches, as the slits can jam.
- Check that the motor motion direction and distance matches the change in value on the display. If not, adjust *gnum* and *gdef* parameters (see below).

You may find it easier to monitor the diffractometer motions by connecting a small computer monitor to the VGA port on the back of the SMC9300. This will then just duplicate the display on the front of the SMC9300 (but without the touch-screen operations).

SMC9300 motor parameters

If you have purchased a complete system of motor controller and a diffractometer, Huber will normally configure the motor controller parameters to match your instrument. The default values for the VTX instrument which is non-standard (two 420 drives, a 511.1 cradle, and 2 motorised slits, but with Anaheim motors) are:

	1	**2**	**3**	**4**	**5**	**6**
Name	2θ	ω	χ[1]	ϕ	Vslit	Hslit
Type	circle	circle	linear[2]	linear[2]	linear	linear
gnum[3]	400	400	400	400	400	400
gdef[3]	1	1	1	2	1	1
ref offset	0	0	0	0	0	0
direction	normal	normal	normal	normal	invert[4]	invert[4]
encoder	none	none	none	none	none	none
start/stop speed	400	400	400	400	400	400
slew speed	3000	3000	3000	3000	1000	1000
acc (and dec)	10	10	10	10	10	10
man mode acc/dec	5	5	5	5	5	5

[1]Note that, as delivered, χ was motor 4 and ϕ was motor 3. It is recommended to swap these and cable ϕ to motor 4, provided that the motors on the two circles are physically identical. If not, this swap has to be made in the *difprof.dat* file (see below).

[2]As delivered, the χ and ϕ were configured as circles, but this results in conflicts with the SINGLE software. So they must be changed to “linear”.

[3]The gear ratios *gnum* and *gdef* define the number of motor steps per degree. *gnum/gdef* = 400/1 means 400 steps per degree, or 0.0025°/step. *gnum/gdef* = 400/2 is 0.005°/step or 200 steps per degree. See the Huber manual for the SMC9300 series controller for more details.

[4]The direction of the slit motors on the VTX Huber was inverted to allow for them being wired backwards. That means with direction set as “normal” they moved positive when the negative button was pressed on the controller.

Note that the motor speeds are given in Hz which means motor steps per second. So, if you have a motor with 400 steps/deg, 400Hz will move the motor at 1deg/sec and 2000Hz at 5deg/sec.

These parameters only affect the manual driving of the diffractometer from the front panel of the SMC9300. The SINGLE software explicitly sets ramp rates and slew speeds with every drive instruction, based on parameters in the *difprof.dat* file (see below). These parameters may be based on the ones set in the controller, but are not required to be the same.

Communications

Huber normally supply an RS232 cable to connect the COM port on your PC to the serial port on the SMC9300. The socket on the SMC9300 accepts a 9-way D-sub plug. The serial line on your PC may be a 9-way or a 25 way. If no com port is available on your PC, use an external USB-to-serial adaptor (see section 4).

If you have to make your own cable:

If you are using a 25-way plug at the PC end of the cable short together CTS, DSR and DTR (pins 5,6, and 20).

Connect through cable:	PC end pin	2	to Huber end pin	3
		3		2
		7 (or 5 on 9-way)		5

Configure the communications on the SMC9300 controller to the following:

9600 baud
8 data bits
1 stop bit
parity none
terminator CR/LF

Edit the *difprof.dat* file and change the COMPORT line to read (for using port 1, usually com1):

COMPORT *N*,1,9600,0,8,0,3,1

The first number *N* indicates the type of serial port drivers. If you are using Windows-98, then *N*=2. For Windows-XP and Windows-7, use *N*=1. The next digit is the port number, so change that if necessary. See section 5.1 for the meaning of the other numbers.

To verify correct communications, use the supervisor mode to send the string "LOCAL;" (not the quotation marks, but include the semi-colon and make it all capitals). The word "local" should be displayed on the SMC9300 display. Now send "REMOTE;". If the SMC9300 display changes to "remote" you are communicating.

Use the Huber commands from the supervisor mode to change the names of circles on the SMC9300 display:

- alias1:2TH — changes name of axis 1 on display to 2TH
- unit3:deg — changes displayed units of axis 3 on display to deg
- update — saves changes

Motor parameters

Each motor of the controller must be described by a line in the *difprof.dat* file that starts with the characters **MOTOR*n***, where ***n*** corresponds to a motor number in SINGLE (1 = 2θ, 2 = ω(bisecting), 3 = χ, 4 = ϕ, 5 = vertical slits, 6 = horizontal slits).

The parameters are (in order):

Description	Value	varname
Motor number in the controller	1 character preceeding a comma	mname
Steps/degree	Must equal *gnum/gdef* in controller	stpdg
Slew Rate (in Hz)		rates(i,1)
Ramp rate to slewing speed	Use Huber recommended values	rates(i,2)
Med speed rate (in Hz)	Usually set to half of slew rate	rates(i,3)
Start speed rate (in Hz)	Typically 300. Must be less than med speed. Also used for step scan speed	rates(i,4)
Minimum circle limit	Set inside physical limit switch	limits(i,1)
Maximum circle limit	Set inside physical limit switch	limits(i,2)

For ω, circle limits are given in absolute ω, and the last number is the minimum angle of inclination between the 2θ arm and the plane of the χ circle.

For motorised slits and other linear tables (e.g. detector drive, sample height) the value of steps/deg is interpreted as steps per mm.

Motor drive instructions are always rounded off explicitly to the resolution of the diffractometer circle as defined in the file *difprof.dat* by steps/degree. This parameter must therefore match those loaded directly to the SMC9300 interface as the ratio *gnum/gdef* :

> The gear ratios *gnum* and *gdef* define the number of motor steps per degree. *gnum/gdef* = 1000/1 means 1000 steps per degree, or 0.001^{o}/step. *gnum/gdef* = 1000/2 is 0.002^{o}/step, and *stpdg* is therefore 500. See the Huber manual for the SMC9300 series controller for more details.

The slew rates are given in Hz which means motor steps per second. So, if you have 400 steps/deg, 400Hz will move the motor at 1deg/sec and 2000Hz at 5deg/sec.

It is safest to set the slew rates of ω to one-half those of 2θ, and start with values below those set as default by Huber in to the controller.

At VTX the limit switch cables for χ and ϕ were removed, and replaced them with blanking plugs supplied by Huber, as these operate as “circular”. Also the pins that activate the limit switches for these two circles were removed. They are set circular in the software by both limits for motors 3 and 4 being both set zero.

The corresponding entries in the *difprof.dat* file for the VTX Huber are:

```
          NAME STPDG  SLEW  RAMP  MED   START      MIN     MAX      SPARE
                      RATE  RATE  RATE  RATE         LIMITS
MOTOR1 1,     400.,   2000.,  10.,    1000.,  400.,   -75.,    110.,    0., 0.
MOTOR2 2,     400.,   1000.,  10.,    500.,   200.,   -46.,    46.,     45., 0.
MOTOR3 3,     400.,   3000.,  10.,    1000.,  400.,   0.,      0.,      0., 0.
MOTOR4 4,     200.,   3000.,  10.,    1000.,  200.,   0.,      0.,      0., 0.
MOTOR5 5,     400.,   500.,   10.,    200.,   100.,   0.1,     9.9,     0., 0.
MOTOR6 6,     400.,   500.,   10.,    200.,   100.,   0.1,     9.9,     0., 0.
```

For the large Huber at the BGI, on an SMC9300 controller are:

MOTOR1	1,	400.,	1500.,	10.,	800.,	300.,	-55.,	55.,	0.,	0.,	0.
MOTOR2	2,	400.,	750.,	5.,	500.,	300.,	-45.,	45.,	45.,	0.,	0.
MOTOR3	3,	400.,	1600.,	12.,	800.,	300.,	-180.,	180.0,	0.,	0.,	0.
MOTOR4	4,	200.,	2000.,	12.,	1000.,	300.,	-180.,	180.0,	0.,	0.,	0.
MOTOR8	5,	400.,	1000.,	25.,	400.,	200.,	2.,	10.,	0.,	0.,	0.

Write the standard motor positions (VIEW, PARK, PHOTO) and the zero offsets in to the *difprof.dat* file. If in doubt, set these all to zero. Make sure that the positions entered do not represent collisions.

Important

Most SMC9300 controllers have been configured on delivery to return motor positions to 3 decimal places. If the number of steps per degree in *difprof.dat* result in a step size that is not an exact multiple of 0.001deg, the output format from the controller must be changed. So, for example, 400steps/deg means each step is 0.0025deg, which needs to be reported to 4 decimal places by the controller. If this is the case, use the Huber commands from the supervisor mode, or the DCOM command, to change the returned resolution on axes requiring this. For example:

- dcpl1:4 makes the controller report axis 1 position to 4 decimal places
- dcpl2:4 makes the controller report axis 2 position to 4 decimal places
- dcpl3:4 makes the controller report axis 3 position to 4 decimal places
- update saves changes

When the motor entries in *difprof.dat* are complete, put NOSHUT in *difprof.dat* (to suppress all shutter operations), start SINGLE in normal mode and check:

- correct motor operation
- operation of stop button
- operation of limit switches
- check software limits prevent collisions and contact with limit switches.

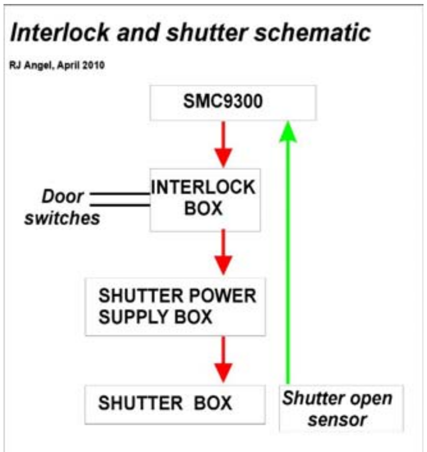
If you hear resonances or the motors stall, or do not always reach their targets, or you note that the zero position is drifting (watch the odometers), adjust the slew speeds or the ramp rates in *difprof.dat.*

Further commissioning

- Once the motors are driving reliably, test with the DEMO command from SINGLE.
- Remove the NOSHUT command from *difprof.dat* and test shutter operation.
- Test counting.
- Try peak profiling, then centering (if scans are too jerky, adjust start speed).

Shutter control

The shutter will be controlled through the additional digital I/O port on the Huber SMC9300 controller. The user needs to connect this I/O port via the 37-way D-connector to a separate box that drives the shutter and also provides the safety interlocks between the shutter operation and the doors of the radiation enclosure. The following figure provides a schematic indication of the relationship of the various control elements.

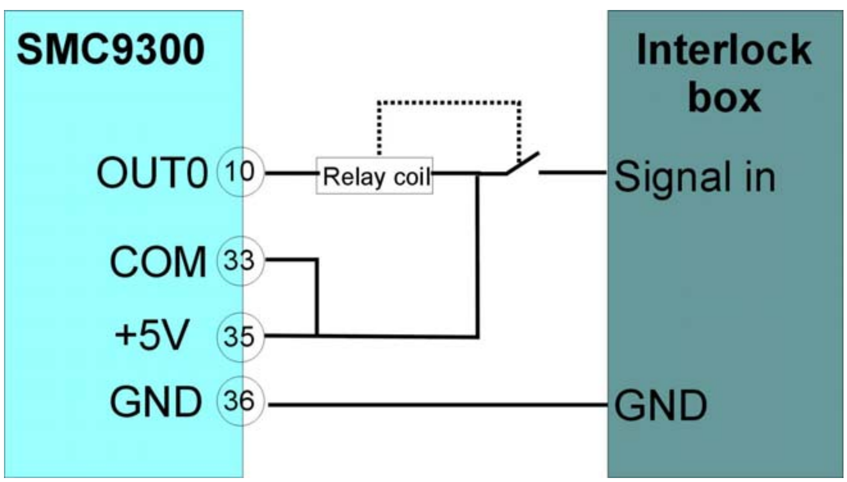


The red arrows indicate that the SMC9300 sends the open signal to an interlock box. The absence of a signal from the SMC9300 is the "close" signal, so the natural power-off or fault state is always shutter closed.

If the signal from the SMC9300 is "open", the radiation enclosure is closed and any other safety criteria are met, then the interlock box sends a shutter open signal to the shutter power supply or shutter controller (depending on your system), which opens the shutter.

The green arrow indicates that, if there is a sensor in the shutter box that detects the shutter is open, then this signal can be returned to the SMC9300, and read from the SMC9300 by the SINGLE software. There is a parameter *ishut_check* on the CONFIG line of *difprof.dat* that specifies whether this signal is read in (see section 5).

The following is the wiring used for the VTX Huber:



The power-up state of the controller is that the photodiode in the SMC9300 attached to pin 10 does not conduct to ground, so no current flows in the relay, and therefore there is no voltage on the signal line, and the interlock box sees no voltage at "signal in". See diagram at top of page 23 of the Huber *SMC Hardware Reference Manual.*

To open the shutter the SINGLE software sends "IO1" to the SMC9300. This allows conduction in the SMC9300 from pin 10 to ground, completing the circuit through the relay which then supplies +5V to the interlock box.

To close the shutter the SINGLE software sends "IO0" to the SMC9300, which returns the circuit to the original state.

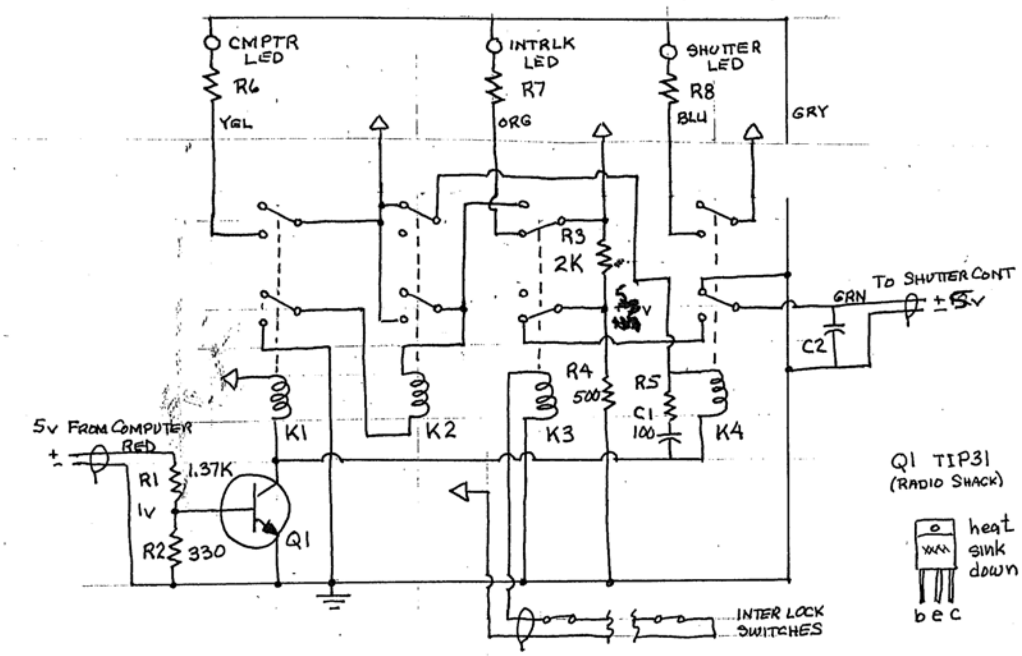
The interlock box and these connections are the responsibility of the constructor of the diffractometer and/or radiation enclosure. None of the instructions here should be read as over-riding the information and instructions provided by Huber. In particular, the user must ensure that his circuitry does not overload the IO card of the SMC9300.

The function of the interlock box is:

- To prevent the shutter from opening when the enclosure doors are open.
- When the shutter is open, to close the shutter when the doors are opened and, on closure of the doors, to stop the shutter immediately reopening.

At VTX a relay circuit that performs these functions was designed. A full description of the function of the circuit and the circuit diagram are on the next two pages.

At VTX this interlock box issues an open signal to the shutter power supply as a +5V signal. A close signal is no voltage. The shutter power supply then drives the shutter in the shutter box.



Note that the output signal level is changed by changing the value of R3; use 2Kohm for +5V output and 250ohm for +12V.

Interlock System

The interlock system on the Huber XRD was compounded by the presence of a four-volt spike on the ground of the system. This was a very consistent level occurring at about 16kc. The shutter control circuit worked properly until the interlock switches were connected. The long run of wire acted as an antenna distributing the 4v interfering signal to the electronics rendering the TTL logic useless. So the interlock system was designed using 12v-relay logic

When the doors are closed the completed circuit applies power to K3 causing it to energize. This would be the normal operating condition. With K3 energized a 5-volt signal from the computer is fed to the base of Q1 through a voltage divider R1, R2 reducing the signal to one volt. This signal turns on the transistor, which energizes K1. When both K3 and K4 are energized a 12-volt signal is generated as an output for the shutter control unit. Energizing the relay in the shutter control unit. When the 5-volt signal from the computer is removed K1 and K4 de-energize and K4 no longer provides voltage to energize the relay in the shutter control unit.

If the doors are opened while the computer is generating a 5 volt signal commanding the shutter to open K3 de-energizes. The 5 volts from the computer energizes K1 and K4. The energized K1 places one side of K2 to ground and when K3 de-energizes power is applied to the other side of K2 causing it to energize removing power from K4 causing it to de-energize no longer providing a voltage to energize the relay in the shutter control unit

K2 has a self-holding contact that provides power as long as it is energized. The only way to de-energize K2 is by sending a low signal (shutter close) from the computer. This causes Q1 to stop conducting and K1 and K2 de-energize removing the return (ground) path from K2 This prevents the signal from the interlock circuit going high when the door is closed and K3 energizes. K4 is de-energized provided a low out to the shutter control.

When the doors are closed and K3 is energized the circuit works normally essentially echoing the input from the computer

If feedback on the shutter status from the shutter box or interlock box is available, it should open and close a circuit between either pin 34 (+12V) or pin 35 (+5V) and pin 20 (IN0A). Pin 1 (IN0K) must be connected to ground at pin 36 or 37.

The circuit to pin 20 should be completed when the shutter is open. Then the command ?IO to the SMC9300 returns a “1” when the shutter is open, or a “0” when the shutter is shut. See the SMC9300 manual for details of signal levels needed to obtain this result.

The VTX Huber is equipped with a shutter sensor that conducts when the shutter is open, and is open circuit when the shutter is shut. The wiring is as follows: