

## **Mini-Sys software**

The Mini-Sys software was written by Brian Markey in Borland Pascal. It can most easily be compiled within the 'Borland Pascal for Windows' environment.

This document briefly describes the procedure for compiling the software, and then gives a list of the modifications that have been made to the original software as produced by Brian.

### ***Compiling MiniSys.EXE***

The unit Mini-Sys.Pas should be loaded into BPW. Under the Options|Compiler section you need to ensure that the 'Conditional defines' section include the word 'minisys'. This is needed because the MiniSys program was originally designed to come in two forms - one designed to run on a true MiniSys system, and the other as a Terminate and Stay Resident (TSR) program controlling the old IBM card. Setting the conditional define to MiniSys tells the compiler that you want the version for the true MiniSys system.

To compile the program select Compile|Build. This will generate a file called MiniSys.EXE which can then be uploaded to the MiniSys instrument. Attempting to run the program on a standard PC (i.e. without all the associated MiniSys hardware) will probably crash the system.

### **Minisys builds, Newest first:**

#### **January 2019 version 4.14**

Version 4.12 and 4.13 was only tested on DASH on classic DASH it didn't work, it is fixed in version 4.14

#### **November 2018 version 4.13**

LOG was added to version 4.12, removed again in this version

#### **November 2018 version 4.12**

DASH focusing unit supported  
SI (set iris) & SF (set focus) commands added.

Support for DASH CPLD version 115.

**Previous version of minisys does not support DASH driver board CPLD version 115.**

#### **March 2018 version 4.11**

The support for “old” pulsed OSL didn’t work stable in version 4.10 or before version 4.10. The SPI bus were not always reads the pulser and avr version correct. This means that the pulsed OSL power value (0-99% without regulation and 100 – 199 ~ 0 -99% power with regulation on to be able to power adjusting the pulsing power) were sometimes handled wrong (as BCD and not as decimal as pulser version 1.09 expects).

In the SPI bus timing parameters no. 116 & 117 are now fixed values for the “old” pulsing board. 116 = 200us and 117 = 500us.

For DASH the timing parameters 116 was previously set to 10us, and parameter 117 was previously set to 100us by default. These default values was causing many retries and sometimes even SPI error 16.

In version 4.11 the default values are changed:

- Parameter 116 default change from 10us to 30us. The allowed range is from 20-200us
- Parameter 117 default change from 100us to 500us. The allowed range is from 300us to 20000us (20ms).

When version 4.11 start-up it checks Parameter 116 and 117 if they are out of bounds they will be changed to the new defaults values.

In the low level command list there was wrong definition of minimum values at parameter 91,93,95,97,99,101,103,105

These was defined as minimum 0, but supposed to be 60000 which it already is in the minisys. Therefore, this has no importance of practice.

## **February 2018 version 4.10**

Ext1. And Ext2. Light source was not working in the PU command  
Reason. The SP arm for osl shall always be the last command executed before stimulation, otherwise it won’t work.

Support for “old” pulsed OSL without power regulation. Pulser ver. 1.09 and AVR version. 1.03.

Fixed issue with a bug in interrupt handling.

Retriggering of 300s. Communication timeout. The 300s timer is retriggered every 0.5s as long as an event or acquisition is active.

## **May 2016 version 4.09**

DASH lightsource ext1 and ext2 in OS, PO & POSL ect. Were missing.

While the DASH detector (DU) or and filter (FS) commands are running the controller LCD display will show the command and its final destination.

Factory configuration file added

From version 4.09 there are two configuration files in the controller.

All parameter in the Mini-sys are hardcoded initialized to default values. See the default values in the low level command documentation.

Each time the Mini-sys powers up the configuration is read from the configuration file from the flash disk and copied to the mini-sys parameters in RAM, The default parameters/reader settings are then overwritten by the values in the configuration file.

Previously there was only one configuration file.

A few times we have seen that the values reinitialize to the hardcoded values, for unknown reasons.

In version 4.09 there are two configuration files, one is as on previous versions of Mini-sys. The new configuration file small be written just after final production test.

The interface to both the configuration files goes thru the RAM parameters/reader settings in the mini-sys these are written by the SA command and read by the RA command.

Parameters added to commands WP and LP. By adding an F as parameter to the commands WP or LP, write and read can access the factory configuration file. WP and LP commands without the F parameter access the normal old and previous configuration file.

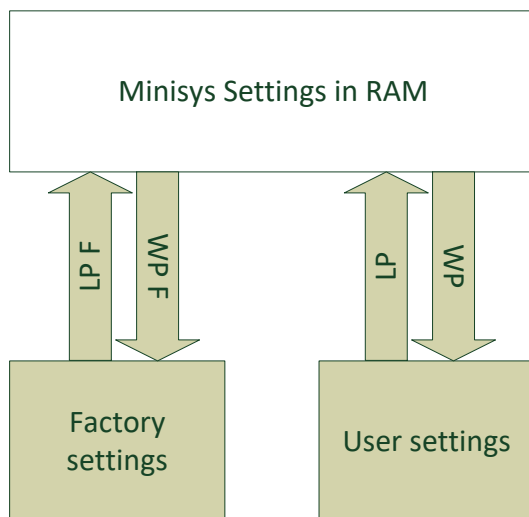


Figure of the mini-sys parameter/readere settings in RAM and the two configuration files.

#### **After the factory final testing:**

Use the dialog tab in the control program

Type *ep syspass* ; enters the password

Type *wp f* ; write the mini-sys parameters to the Factory configuration file

If the parameters/reader settings is lost:

Use the dialog tab in the control program

Type *ep syspass* ;enters the password

Type *lp f* ;loads the factory parameters/reader settings onto the Mini-sys

Type *wp* ;overwrite the lost configuration

## January 2016 version 4.08

### Changes in TL command for EMCCD acquisition

From : **TL t r p f [m]**

Record a TL glow curve. The turntable must be on position.

*t* = maximum temperature

*r* = heating rate

*p* = number of data points

*f* = final temperature ( $f \leq t$ )

*m* = omitted or 0: data points generated from PMT counts (\*note 1)

*m* = 1: no data points generated, but HW ACQ (ACQuisition) signal set for EMCCD camera acquisition (\*note 1)

To: **TL t r p f [m]**

note 1: when camera option *m* 0 or 1 is defined, option *f* must also be defined.

Record a TL glow curve. The turntable must be on position.

*t* = maximum temperature

*r* = heating rate

*p* = number of data points (see note1)

*f* = final temperature ( $f \leq t$ )

*m* = omitted or 0: data points generated from PMT counts (\*note 1)

*m* = 1: no data points generated, but HW ACQ (ACQuisition) signal set for EMCCD camera acquisition (see note 2)

note 1: If  $p=0$  no data points are generated and the Controller immediately starts setting final temperature to *f*.

note 2: When camera option *m* 0 or 1 is defined, option *f* must also be defined.

Better reading of configuration file to avoiding lost settings.

Parameter 20 was on version possible to have values above 1. So the changes is

Parameter 20 = 1...65535

the SR command calculate the end position in the same manner as version 3.35

Parameter 20 = 0 the SR command calculate the end position in the same manner as version 3.36

The equations can be seen here:

[XY endpoint calculation.xlsx](#)

## November 2015 version 4.07

SF command added

Option to parameter 20 added for scanmode of XY rails:

Parameter 20 = 1      the SR command calculate the end position in the same manner as version 3.35

Parameter 20 = 0      the SR command calculate the end position in the same manner as version 3.36

The equations can be seen here:

[XY endpoint calculation.xlsx](#)

### **September 2015 version 4.06**

TO command SP37 added for DASH

### **July 2015 version 4.05**

Changes to DASH FS and DU commands

### **May 2015 version 4.04**

Minor bug fix

### **April 2015 version 4.03**

Illumination command added

### **Marts 2015 version 4.02**

Minor bug fixing

### **February 2015 version 4.01**

Minor bug fixing

### **February 2015 version 4.00**

Support for DASH. Build on version 3.37

### **February 2015 version 3.37**

Better noise immunity in detection of COM connection (USB / RS 232)

### **December 2014 version 3.36**

Improved SR command

### **September 2014 version 3.35**

Wrong timing in previous version 3.34.

### **September 2014 version 3.34**

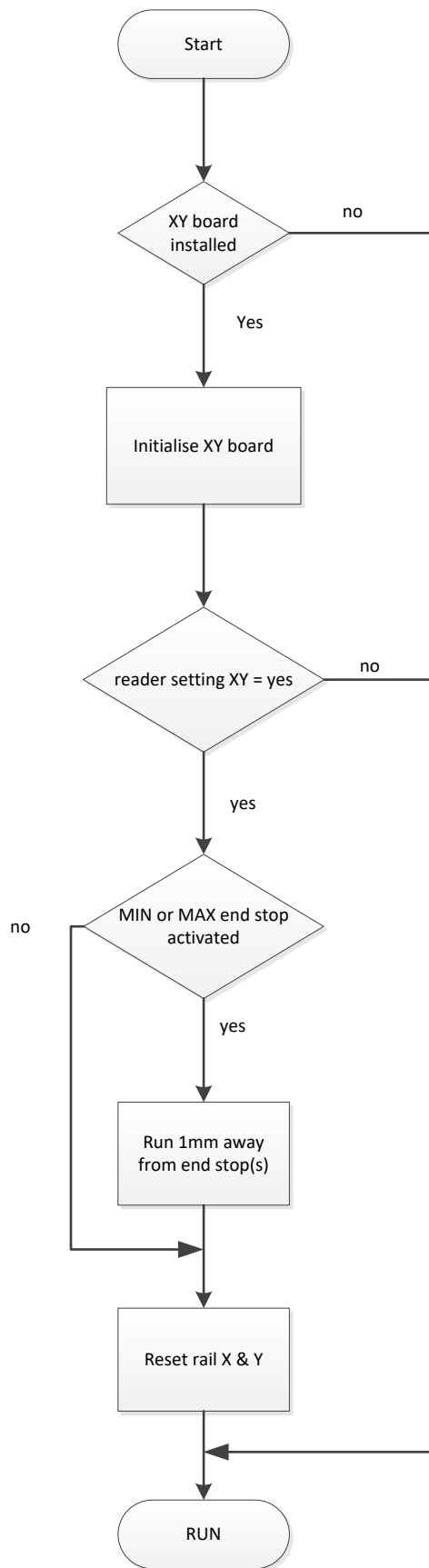
### **DO NOT USE VERSION 3.34**

Changes in SR command

Changes in RS 3 for XY end stop, High and low end stop was swopped in all previous versions!

New feature to automatically get free of XY end stop (min and max on both rails.

FLOW:



**September 2014 version 3.33**

This version was similar to 3.32 with only small changes to SR command.

### **August 2014 version 3.32**

In the PO command with inactive channels before stimulation errors occurs if channels were longer than 0.655 seconds.

Channels can now be defined from 1ms to 429seconds.

### **May 2014 version 3.31**

#### **Bug fix.**

Heat check reading, in previous versions a noisy signal on the read temp could force the heating to fail with an error “heating failed”.

### **June 2013 version 3.30**

Support for EMCCD camera in the following commands OS,PO,TL and TO.

Problem with electrical spikes on green XY laser signal has been solved.

Parameter for maximum laser power has been added.

### **February 2013 version 3.29**

Problem with “system reset” after 5 minutes with no communication to the controller, it hangs the program and a reboot was the only solution to come in contact with the controller again.

This problem only occurs on controllers with the B-Plus CPU Dimm-PC boards.

### **January 2013 version 3.28**

Problem with update software UP and UW command on controllers with B-plus CPU boards fixed.

### **December 2012 version 3.27**

“bug” fixing in command ST (set temperature) this cause that the command ST 0 was not possible since version 3.22

### **November 2012 version 3.26**

This version is the first version which can be used with the B-Plus DIMMBoard, **previous versions CANNOT BE used with B-PLUS DIMMBoard AS the program will hang if there is no graphic board present.**

The only change from version 3.25 is that this version checks whether there is a graphic board present. If a graphic board is present it will be used otherwise it will not be accessed.

### **June 2012 version 3.25**

For readers with lift underneath the beta irradiator, the delay for checking the lift position was changed from 20milliseconds to 100milliseconds

### **September 2011 version 3.24**

Parameter 88. Beta irradiator check was default set to “0” and not as documented “1”. This is changed in this version. So the Beta source check by the mechanical irradiator microswitch now is checked by default. Parameter 88 = 1.

### June 2011 version 3.23

Default lift time out changed from 60second to 300 seconds  
Max. Xray mA default value changed from 1mA to 2mA.

Active hi or lo bleach shutter signal can be changed with parameter 112.

### May 2011 version 3.22

Changes in command ST so that ramping lower temperature is possible

### March 2011 version 3.21

#### Hardware versions command

Software support for calibration of analog input channels, implemented in Eeprom.

Two user commands implemented

HW reads hardware version

HT rread hardware date of test, who has tested the CPU board, possible repair notes.

#### EEPROM read & write commands

The layout of the EEPROM is:

Decimal Address	Description
0..1	Table type *1 (table 0x1000)
2..257	Hardware version, string 256bytes
258..513	Date of test + engineer + repair ect, string 256 bytes
514..515	Ai channel 0 offset, msb first, word
516..517	Ai channel 0 full-scale, msb first, word
518..519	Ai channel 1 offset, msb first, word
520..521	Ai Channel 1 full-scale, msb first, word
522..523	Ai Channel 2 offset, msb first, word
524..525	Ai Channel 2 full-scale, msb first, word
526..527	Ai Channel 3 offset, msb first, word
528..529	Ai Channel 3 full-scale, msb first, word
530..531	Ai Channel 4 offset, msb first, word
532..533	Ai Channel 4 full-scale, msb first, word
534..535	Ai Channel 5 offset, msb first, word
536..537	Ai Channel 5 full-scale, msb first, word
538..539	Ai Channel 6 offset, msb first, word
540..541	Ai Channel 6 full-scale, msb first, word
542..543	Ai Channel 7 offset, msb first, word
544..545	Ai Channel 7 full-scale, msb first, word
546..547	Checksum

The table type is unique, should be set to 0x1000



Hardware version & Date of test string are a zero terminated string which length can be between 2 and 255 characters long.

The Checksum is calculated by adding all bytes into a word (16bit) variable.

The analogue values for offset and fullscale are also stored as parameters in case of the EEPROM lost the values.

The parameters used for this are:

90 offset ch 0  
91 full scale ch 0  
92 offset ch 1  
93 full scale ch 1  
94 offset ch 2  
95 full scale ch 2  
96 offset ch 3  
97 full scale ch 3  
98 offset ch 4  
99 full scale ch 4  
100 offset ch 5  
101 full scale ch 5  
102 offset ch 6  
103 full scale ch 6  
104 offset ch 7  
105 full scale ch 7.

Commands for reading and writing to the EE PROM are not listed in the command list as they only are supposed to be used by Risoe staff.

Write command:

*WE addr. Data*

There *addr.* Is the address in EEPROM range 0 – 0x8000, *Data* is the 16bit data word to be written.

Example we 0 0xAA55

Writes 0xAA to Address byte 0 and 0x55 in address byte 1 in the EEPROM.

Read command:

*RE addr.*

There *addr.* Is the address in EEPROM range 0 – 0x8000. The 16bit data are returned.

Example re 0 returns 43605dec = 0xAA55 this means that EEPROM address 0 contains 0xAA and address 1 contains 0x55.

An extra status byte is also implemented: byte 6

Bit.0 = EEPROM checksum failure;

Bit.1 = N/A

Bit.2 = N/A

Bit.3 = N/A

Bit.4 = N/A  
Bit.5 = N/A  
Bit.6 = N/A  
Bit.7 = N/A

### **IR set value**

Value is changed so that fractional values can be written.

### **Readback heater element**

temperature RT 1 does now have linearization. A four grade polynomial is added in the following parameters:

106: 0=linearization disabled, 1=linearization enabled

107:  $X^4$  5.86008E-0010

108:  $X^3$  -8.80585E-0007

109:  $X^2$  3.86796E-0004

110:  $X$  9.48209E-0001

111:  $X$  offset -4.84946E-0002

## **January 2011 version 3.20**

Unused single grain code removed

Parameter 20 – 38 removed

Parameter 43 – 49 removed

The parameters are still there but they are not used.

Detection of XY board: if parameter 17 is set to 1 and no XY board is present the parameter 17 will be cleared.

BI command changes:

Depending of parameter 88 setting the function are as follow:

1 = Hardware check via microswitch whether betasource is on or off the result is shown on status register 2 bit 7 if the microswitch does not change to active state within 1.5s. a irradiation failure will be set in status register 5 .

0 = The Beta source irradiator control signal is copied to status register 2 bit 7 one minisys syscheck circle (approx 0,5s) after. No changes in status register 4 or 5.

## **September 2010 version 3.19**

An error in temperature linearization was introduced from version 3.16. this is fixed in version 3.19.

Default parameter for single grain max X and max Y are changed from 12000 micron to 14000 micron.

Parameter 88 then this parameter is zero the status register 2 bit 7 are copied one syscheck circle ( $>0,5s$ ) after beta irradiator is on.

### **August 2010 version 3.18**

Another bug in OSL linearization was fixed.

Mechanical check if beta source irradiator is on copied to status register 2 bit 7  
This feature can be disabled by setting parameter 88 to 0 in this case the beta irradiator control signal is copied to status register 2 bit 7.

### **July 2010 – version 3.17**

Bug fixing is OSL linearization.

### **May 2010. Version 3.16**

USB communication implemented, but only with controller boards with version D or newer. The controller/minisys detects whether the USB cable is connected to the host or not, by default the serial RS232 comport is active. If the USB cable is connected the communication will work as USB-serial converter (comport) or as USB, if such are present and implemented in the sequence editor. When the USB cable are removed from the controller switch back to the serial RS232 comport.

### **March 2009. Version 3.15**

Timing bug was fixed in the following commands OS, PO, TO, OP: the error coursed empty channels with zero contents in up to 80mS after an OSL stimulation command was issued to the controller.

Rounding error in stimulation time was implemented to the minisys in between version 2.06 and 2.11 and has been there for years! Until this version, it is now possible to define stimulation times with two decimals.

It is now possible to define the polarity of the OSL shutter signal. This is done by setting of parameter 67 which is added to this version.

Parameter 67 = 0 the shutter signal is active low

Parameter 67 = 1 the shutter signal is active high

### **October 2008. version 3.14**

Pulser PMT gate open in following commands

OS : StartOSL

PO : Pulsed osl

TO : StartTOL

### **July to October 2008. version 3.13**

There are several different releases of this version deployed worldwide!.

The main things

1. Bug in the PC & OP command so for use with pulsed OSL please use the next version.
2. SPI retry on pulser settings inside PC & OP commands.
3. Gearbox variable removed from single grain sample time.

### **July 2008, Version 3.12**

Support for built-in Pulsed OSL implemented

Following command was added:

SP (sending SPI commands direct to Pulser board

PC (command setup pulser ei. used in calibration of pulsed OSL board)

OP (Pulse stimulation command)

Parameter for setting time resolution in singlegrain reflection scan (SR command)

XY default parameter changed so they fit the new rails

Parameter	Changed from	Changed to	Function
39	4960	2158	Y steps per 10000 micron
40	4960	2158	X steps per 10000 micron
41	22000	12000	Max X travel in microns
42	22000	12000	Max Y travel in microns

### **April 2008, Version 3.10 & 3.11**

Software filter in XY end stop detection implemented.

In version 3.10 the filter has no effect and this version was released by a mistake!

### **December 2007, version 3.09**

Read of X-ray KV and mA was swopped in previous versions.

### **June/July 2007, version 3.08**

The amount of datapoints are increased from 5750 to 9999.

The on limitation is at this time the Sequence editor. 1..7998 datapoints are tested with the TL and OSL command.

### **June 2007, version 3.07**

New version of XY system added.

Parameter 39 and 40 which are definition of encoder steps per 10000 micron.

Rails before June 2007 has 4960 steps per 10000 micron.

Rails after June 2007 has 2158 steps per 10000 micron.

The sequence editor now deals with those two parameters

Also the Gearbox ratio, to get the right velocity, can be defined as parameter 56  
to

Gearboxes before June 2007 has gearbox ratio 16

Gearboxes after June 2007 has gearbox ratio 35.55

Set of parameter 56 has to be done manually.

### **September 2006, version 3.06.**

Parameters for heater curve fitting added. the curvefit is made as a four grade polynomial there parameter no.

51 is  $x^4$

52 is  $x^3$

53 is  $x^2$

54 is  $x^1$

55 is offset

The parameters can be read/written by the commands RA for read and SA for write. Write of the five parameters are password protected.

### **July 2006, Version 3.05.**

Minor change in start-up text. The countdown from 5 seconds count right now. So that 0 doesn't take 1 second.

The lift timeout was concerning raising lift only and the lowering lift timeout time was fixed to 10 second.

The lift timeout can now be set by two separate system parameters

Raise lift timeout (eg SA/RA 2, called Lifttimeout in the control program)

A new parameter is added in this version

SA/RA 50 as Lift down timeout, the default value is 60 seconds.

### **February 2006 Version 3.04.**

Changes in cancelall procedure.

If the event doesn't vent Idle within 20sec (40sec. because of two event wait idle functions in the cancelall procedure). This happens if no reader is connected to the controller.

In command UP (Update program) acknowledge is send after each transmitted frame, but the Minisys EOT was missing. The EOT chars (default CR + LF) are now added.

In Hardware the analogue input channel 2 and 4 are swapped so that laserpower can be read as in previous Minisys system.

### **February 2006 Version 3.03.**

Reader systems with two heater plates does not always lowering both lifts.  
The cancelall procedure now act as below  
take relay 2 to off  
wait 20mS  
if lift1 not down then lowering the lift  
write message to display line 2 “wait until lift down”  
repeat until event = idle (This will hang the program if the reader is disconnected)  
take relay 2 to on  
wait 20 mS  
if lift2 not down then add commands to queue to lowering the lift  
repeat until event = idle.  
take relay 2 to off.

this changes in the cancelall procedure will toggle the relay 2 in reader idle state with 5 minutes interval.

A problem with LM-OSL was discovered. with a ramp from “something% – 100% diodelevel and only ramps to 100% the DAC will over roll the DAC resolution.

In MS\_DATA the StartOsl procedure the formula for calculating was changed from :

$$\text{DiodeStep} = ((\text{StopDiodePower} - \text{StartDiodePower}) * \text{DiodeVMax} / 100) / (P-1)$$
there:

P is the count of channels.

the formula is changed to:

$$\text{DiodeStep} = ((\text{StopDiodePower} - \text{StartDiodePower}) * \text{DiodeVMax} / 100) / P$$

### **January 2006 version 3.02.**

Again changes in SG timing. The two procedures read and write registers was rewritten instead of keeping /ALE within /CS which seems to force the encoder motors to the end stop sometimes Furthermore the read registers was done twice for some reason.

The changes of /ALE and /CS signals to be non overlapped, does that read of registers was more stable and only done once.

A test with finding grains was done for 121 disks with 100 grain holes at each disk.

### **January 2006                      Version. 3.01.**

Changes in SG timing. Microsecond delay procedure added.

### **December 2005                      Version 3.00**

The new Controller software is build on the Minisys/Controller version 2.18 which was the first working version on the Controller.

There was a bug fixing in hardware done in this version 3.00.

The bug was that room/ambient temperature & heater temperature was swooped in hardware. These two are now swooped back in software. This means that this version only can be used on the Controller type. Later on there will be a version which can run on previous MiniSys systems.

Two new commands are implemented, for updating the Controller software. These two new commands are both high priority commands and also password protected.

### **UP command:**

Then the UP command is issued from host the Controller assign a new file called "temp.exe". If the Controller succeeded this, an ascii ACK is transmitted, otherwise an ascii NAK is transmitted, by the Controller.

If an ACK is received by the host the file transmission can begin. The file is send by the host by a simple ascii protocol, described below.  
The maximum length of a frame is 255 byte.

The frames from host begins with an ascii start of text (STX) and ends with end of text (ETX) or an end of transmission (EOT), in between STX and EXT/EOT the data and a checksum (CRC) are included.

STX	DATA	CRC	ETX
-----	------	-----	-----

If one or more data bytes contains one of the following control characters, NULL=0, STX = 2, ETX = 3, EOT = 4, LF = 10, CR = 13, DLE = 16, NAK = 21. A data link escape (DLE) is inserted before the data byte, which was a control character, this data byte is also added by the constant of 0x10 (16 decimal).

Example:  
Original data is:

1	3	15	4	2
---	---	----	---	---

CRC (checksum) will in the frame above, be calculated as  $1 + 3 + 15 + 4 + 2 = 25$ . CRC (checksum) are 16bit wide. Which means then CRC exceed 65535 CRC rolls over zero, so that 65537 will give 1 as CRC.

As data bytes 3, 4 & 2 are control characters ETX, EOT & STX, three DLE sequences will be inserted and 0x10 has to be added to each control character in the frame.

1	16 (DLE)	19 (ETX + 0x10)	15	16 (DLE)	20 (EOT + 0x10)	16 (DLE)	18 (STX + 0x10)
---	-------------	--------------------	----	-------------	--------------------	-------------	-----------------------

Therefore the frame will look like

STX	data 0	data 1	data 2	data 3	data 4	data 5	data 6	data 7	CRC hi byte	CRC low byte	ETX
2	1	16 (DLE)	19 (ETX + 0x10)	15	16 (DLE)	20 (EOT + 0x10)	16 (DLE)	18 (STX + 0x10)	0	25	3

If CRC hi byte or CRC low byte are one of the abovementioned control characters, these will also be expanded to DLE sequences.  
When the last frame in the file is to be sent from the host to the Controller the ETX is replaced by an EOT.

STX	data 0	data 1	data 2	data 3	data 4	data 5	data 6	data 7	CRC hi byte	CRC low byte	EOT
2	1	16 (DLE)	19 (ETX + 0x10)	15	16 (DLE)	20 (EOT + 0x10)	16 (DLE)	18 (STX + 0x10)	0	25	4

Each frame sent from host to Controller is receipted by the Controller with ACK for Ok and NAK for fail.  
If the Controller receipted a frame with a NAK, the host can retransmit the frame again up to 10 times. After that the Controller times out and no further file transmission is possible before the host issue a new UP command.

### UW command:

syntax:

UW CRC

Where CRC is the expected checksum of the file (temp.exe) received by the Controller. If the temp.exe file did not have the same CRC as received, the Controller respond with an errorcode 119. And the existing program remains as before. Otherwise the old program is replaced with the new received file.

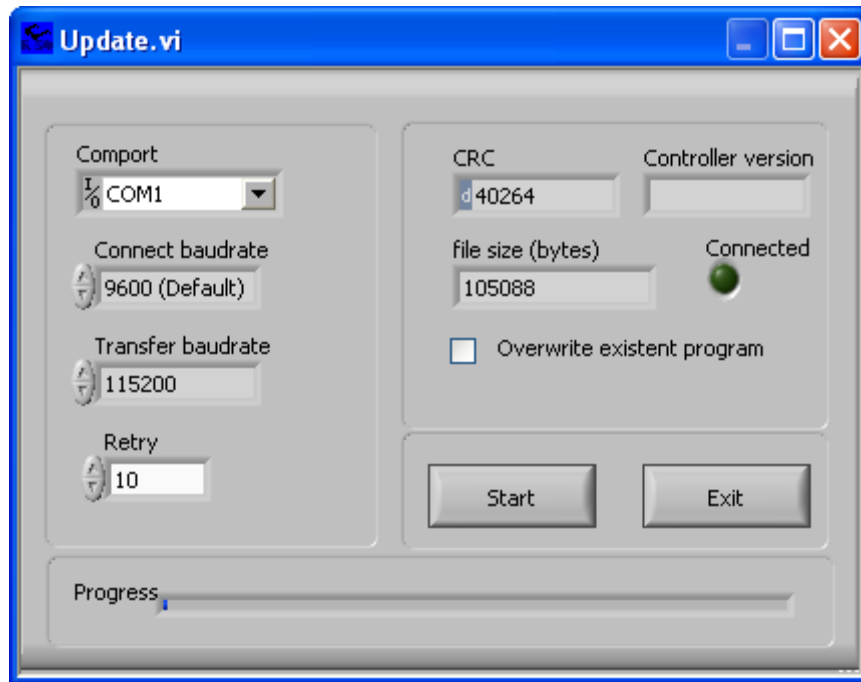
**If something vents wrong overwriting the existing program the Controller respond with an errorcode 120. in that case it is important not to turn power off from the controller. But try to update the controller again!**



A prototype of a program which can update the controller are written in Labview. The installations files can be loaded from the Risoe S:\OSL\Minisys 2005\lapi\Update software.

To install the program on your local machine, find and execute the Setup.exe and follow the instructions.

The program looks like this



### November 2005 Version 2.18

Change from MiniSys to Controller, this was done because of the new design where MiniSys and Controlbox are build together in one box.

Startup text in display (LCD only) are changed from Riso to Risø.

Use of numeric coprocessor added.

Default settings were changed, the values are :

- Multispeed turntable = yes
- Turntable timeout = 60S
- Turntable start delay = 500mS
- Turntable stop delay = 15mS
- Turntable half position = 2000mS
- Num. Position on the carousel = 48
- Check heat = yes
- Max. hardware temperature = 700°C
- Software temperature limit = 700°C
- Lift stop delay = 450mS

Lift timeout = 60S  
Max heating rate = 10°C  
Monochromator reset timeout = 100S  
Comport output delay = 100µS  
Default diode power = 40%  
Irradiation offset = 0mS  
Reader type = New  
Single grain reader = No.

This version were build on version 2.17.

**May 2005 Version 2.17**

(a) Lamp Failure checks in HLSYSCTRL have been disabled so that this no longer operates. (b) Additional code added to Cancel All routine in HLSYSCTRL to check whether the second lift is lowered on systems with a hot plate under the beta source. This uses relay number 2 to switch the hotplate controls from one hotplate to another. (c) Port 782, bit 1 has been assigned to Control IO Enable for the MiniSys2005 designed by Lars. This bit has to be toggled 4 times at start up to reset all the outputs. Code for this has been added to MINISYS.PAS.

**June 2004 Version 2.16**

"Irr Failure" system was not correctly implemented in 2.15. Main error was that the system also checked for the X-ray being OK when doing a beta or alpha irradiation! In 2.16 this bug was removed. Note that Irr Failure can only occur when doing an X-ray irradiation since the source control line is not connected for the beta or alpha sources.

Also added a simple debugging command "BR x". This stands for "Byte Read" and will return the value of the port at address "x". This is solely for debugging purposes and is not listed in the Commands document.

**2003 Version 2.15**

Add new failure message "Irr Failure" that will occur if the x-ray fails during an irradiation.

**May 2003 Version 2.14.**

Minor changes.

- (a) Alter default values for Com Read Delay from 5 to 100 (reflecting change in units made in 2.12).
- (b) Alter default for Turntable Stop Delay to 15 ms.
- (c) Make check for thermal failure more robust by requiring it to find the thermal failure 5 times, each separated by a half second.
- (d) Alter the Display module so that it resets the cursor position when the MinSys code first starts up. This should prevent the display from getting too messed up.

### Mar 2003 Version 2.12.

The major hardware addition at this point is the X-ray system. This has involved a number of new MiniSys commands and new hardware. It has only been implemented on the MiniSysII system since the older systems cannot have the new output capability. The new hardware is 2 additional DAC convertors. These are DAC's number 3 and 4 and the number of DACs specified in MS\_PORTS has been increased to 4. The new commands are:-

XI [t]            Open X-ray shutter for a specified period of time. Before the shutter is opened, and during irradiation, the monitor signals from the X-ray control box are checked to see that they are close to the values requested using SX. If they are not then a failure will occur (failure code 11).

XC                Close X-ray shutter and end any timed irradiation

XP                Move the carousel to the X-ray position (this is MAXPOS div 2)

SX v I            Set the power for the X-ray system using the DAC outputs. The values specified are [v] is kV and [i] in mA. The maximum values for these are currently set at 50 kV and 1 mA corresponding to 10V on the DAC output. The power is ramped from its current value to the new value to avoid spikes in the X-ray control. This is done using the HeatTimer (EventTimer) and a new event called "SetXRay".

RR n             This reads the x-ray settings of kV and mA.

As part of the safety features of the X-ray system a watchdog timer is built into the x-ray control box. If a pulse is not received from the MiniSys for a period of 1.6 seconds then this watchdog will disable the x-ray system. The pulse is generated in the main housekeeping routine (checksystemstatus in hlsysctr.pas) that is run every 0.5s to update the MiniSys display. An additional safety is that if the MiniSys goes to sleep (because no communication has occurred for 5 minutes) then the X-ray power is ramped down to zero. Also, the baud rate is reset to 9600 in case communication has broken down.

As part of testing the watchdog feature it was realised that the one process that locks the MiniSys for any period of time is the RD instruction to read data. This is written to the Com port in a tight loop within MS\_DATA.pas. It becomes very slow, especially if the output buffer becomes full (maximum size is 8192 bytes). To correct this, reading data becomes a timed event. EventTimer is used to send one channel of data at a time to the Com port. If the buffer is full then no data is sent. The variable OutBufChCount in RS232.pas was implemented fully to allow the program to check whether there is space to write data to the buffer or not. In ReadDataPoint in MS\_DATA.pas, Event Timer is set up to provide pulses at an interval specified by ComOutputDelay (parameter 13 in the MiniSys). A new event called 'SendData' has been created for this, Everytime the interrupt occurs, a data point is written to the Com port. This makes the process more robust and allows it to go quicker. The ComOutputDelay is now specified in microseconds instead of milliseconds, and seems to be stable with values of 100  $\mu$ s. This has a great advantage during transfer of data during reflection scans. If higher baud rates are used, then this also helps.

Bit 2 of status byte 1 now indicates whether any type of irradiation (alpha, beta or x-ray) is being carried out. In addition, bit 6 of status byte 2 is now used to

indicate the condition of the X-ray interlock to see whether the X-ray system is ready to generate X-rays.

#### **Mar 2002 version 2.06.**

Jakob added support for the new infrared laser on the single grain system. This uses commands based around the instruction 'LI' and mirrors the commands for the green laser using 'LA'. These two light sources are defined for use by any commands (OS, BL, PO etc) as 'G' for the green laser and 'A' for the infrared laser.

The 'CB' command to alter the baud rate used by the MiniSys was changed. Previously this could not operate above 9600. The BIOS call to Int 14h set this limit. However direct programming of the registers in MS\_COMM.Pas allows baud rates up to 115200. The other change was to the CommTimer. This is started when a CB command is received. The idea is that if communication has not been reestablished with the MiniSys within 30s then it will revert to the previous baud rate. However this CommTimer event was not reset in the code. This was added to the '!' command in HlSysCtr.Pas.

#### **Feb 2002 version 2.05**

In previous versions the halogen lamp had been disabled in all systems fitted with a single grain unit. In order to simplify programming, the halogen shutter control routines had been redirected so that they switched the green laser on the single grain system on and off (This happened in the routine ChangeSysParams in MS\_PORTS and GS\_PORTS. This meant that issuing a command such as BL L 5 would switch the green laser on for 5 seconds. It also meant that we lost control of the halogen lamp output! Since this is now being used again, we have got rid of this botch. In "pcsys" systems the halogen lamp is left so that a BL L 5 command will operate the halogen lamp. The green laser on the single grain has been added as an extra light source 'G'. So the XYWIN software needs to be altered so that it issues a PO G .... command instead of a PO L .... command.

Also, we need to be able to ask MiniSys to go to the position appropriate for bleaching with a specified light source. This new command is PL (position of light source). The command PL 5 L will move the carousel so that sample 5 is under the light source L (halogen lamp). This means that sample 5 will be on the measurement position. All light sources except W will move the sample to the measurement position. Issuing PL 5 W will move sample 5 to the white light position. This is exactly opposite the measurement position (MAXPOS div 2).

Switching the white light shutter on and off will AUTOMATICALLY switch the HV supply off and on.

#### **Jan 2002 Version 2.02**

Introduction of new light source NOT on the measurement position. This was requested by Lamothe. This is a xenon type lamp requiring warm up times and a shutter. This has been added as an extra light source 'W'. New routines have been added to control the WHITELIGHTSHUTTER and the WHITELIGHT. Also, four new commands have been added:-

WA – switch on the xenon lamp  
WD – switch off the xenon lamp  
WC – close the shutter on the xenon lamp  
WO – open the shutter on the xenon lamp

Also added control of the high voltage supply to the photomultiplier tube via the commands

HV ON &amp; HV OFF

**2001** A substantial part of the Single Grain software has been implemented in the MiniSys code using a set of new commands and internal data structures. This allows the MiniSys alone to undertake the hole search routines and location of single grain discs without having to communicate continuously with the control computer. These are described in the COMMANDS.DOC file.

Jakob added power control of blue diodes for TLDA systems running GeniSys. To activate this control one has to include the compiler directive ‘specielgeni’ along with ‘genisys’.

New hardware based around a standard Pentium PC was introduced with new interface cards designed at Riso – replacing the AX104 boards that had previously been used in MiniSys design. The change in boards was designed to be largely invisible, but there were some changes to addresses and functions. Therefore the MiniSys code has been altered to allow an additional mode of compilation. There are now three ways to compile the MiniSys code, each for a different hardware platform....

1. GeniSys      Designed to run on the TL-DA interface card
2. MiniSys     Designed to run on the AX104 boards
3. MiniSys II   Designed to run on the Pentium systems with the in-house interface cards

The compiler directives are ‘genisys’ for option 1; ‘minisys’ for option 2 and ‘minisys:pcsys’ for option 3.

## May 2000

Bleach – there is an error in the bleaching routine that means that if one uses the command (BL I 10 – bleach with IR for 10 seconds) once the lift is in the up position already then it is OK. However, if the auto-lift procedure is used then it all works correctly except that after the function has finished the MiniSys thinks that it still has a command running (shown as 64 in the status byte).

This was caused by the change made in April 99 by altering the lift down event. When the lift down delay was added, the AutoLift = false command was removed, so whenever ANY instruction used AutoLift it would never finish!  
Corrected in version 1.10.

Added a new parameter to the MiniSys (18) that is a software limit for the temperature of the hotplate. Since the user cannot alter the hardware limit on the hotplate, this provides software protection that will prevent the MiniSys from

accepting any commands in excess of the Software Temperature Limit. I have also made it so that the hardware temperature limit can only be changed if you enter the system password. Simply entering the standard password (risoe) no longer allows access to that parameter.

### **Summer 1999**

GENISYS – Enver and Andy realised that one could use almost the same code to drive the old TLDA card directly as a ‘MiniSys’ system that is used to drive a real MiniSys. This is called a GeniSys. The main change is simply that the addresses for IO and timing are different. The chips involved (8253 for timing and 8255 for IO) are the same in both the MiniSys and the TLDA card.

The only substantial IO difference is in the collection of counts from the PMT – otherwise the two systems are very close.

The main structural difference is that the TLDA card only has two interrupt driven processes. On the TLDA card as it was previously used by the MT software (v4.65) both these interrupts were disabled. To use the GeniSys these interrupts were reconnected so that both the heater and the PMT are interrupt driven. On a ‘real’ MiniSys there are a series of other processes (such as Lift Up, Lift Down, Next Position, Position 1) that also generate interrupts. This does NOT happen on the GeniSys – instead they are polled by a small routine in the main loop and this then generates a ‘software interrupt’ if they have changed state.

### **April 1999 version 1.09.**

Henrik noticed a problem with the lift control. When using the vacuum pump, the drop in pressure within the measurement chamber could sometimes move the lift so that it was no longer in the down position. This is presumably because the lift motor stops as soon as the lift down sensor is triggered and there is no overrun. The problem appeared to be in the LowerLift procedure (in llsysctrl.pas), where there is not a delay,

```
delay(LiftStopDelay)
```

as there is in the RaiseLift procedure. However, altering these routines had no effect! The reason is that these procedures are never called! They appear to be totally superfluous - the raise and lower lift procedures are in fact controlled via the hardware interrupt procedure in hlsysctrl. Here there are two events RaisingLift and LoweringLift. When a delay was added to the LoweringLift event then this successfully changed the timing.

The timing is not critical (as it is for the RaisingLift event) but it will force the system to overrun so avoiding the problem with the vacuum pump.

The RaiseLift and LowerLift procedures in llsysctrl have been commented out since they are never used.

## Nov 98 version 1.07

The current versions of MiniSys will allow non-integer heating rates, but the minimum heating rate is approximately 0.27C/s. This is set by the maximum value that can be programmed into the count down timer used for the interval between increment the heating DAC. The value is 65535, and each clock pulse is 100us. Hence the maximum time is 0.65s, and each DAC increment is equivalent to 0.17C. This gives a heating rate of about 0.27C/s. In order to be able to use slower heating rates a new global variable has been defined in the routine ms\_heat.pas called *NumHeatRepeats*. This variable, which is normally set to 1, instructs the MiniSys how many times it has to decrement the count down timer before incrementing the DAC. In the ms\_heat routine *SetUpHeater* the variable NumHeatRepeats is set to a value above 1 if a very slow heating rate is needed and the length of time between incrementing the DAC needs to be more than 0.65s.

In the interrupt handling routine HardwareIntHandler in hlsysctr.pas, each time the count down timer for the DAC reaches zero, the value of NumHeatRepeats is decremented, and the DAC is only increased when NumHeatRepeats reaches zero. Since NumHeatRepeats is a 2 byte integer, in theory this allows the time between DAC increments to be up to 65535 \* 0.65s. However, in practice for most applications the limit is now set by the longest interval between data acquisition points. This is set to 65s.

## Aug 98

The position information maintained by the MiniSys contains two potential problems. The first arises when the lid is opened. Currently the system does not check this unless it is attempting to do an operation. In other words it is possible to open the lid, move the carousel (or reposition a new carousel), close the lid, and the system will still think that it knows the position of the carousel!

This is clearly a problem. Currently this situation rarely occurs since the TL/OSL program issues a TR (turntable reset) prior to every run. However this bug should be corrected.

The second 'problem' is only a problem if some error occurs in the system. If a locating hole on the periphery of the carousel is blocked for instance. In this case the MiniSys will have incorrect position information. However there is no check on when the carousel passes the position 1 sensor. This should occur in order to confirm the current position information.

### The Lid

Part of the Control Port monitors the status of the lid. This can be checked using the function *LidIsClosed* in *LLSysCtrl.Pas*. A routine could be added that resets the position control to -1 (i.e. undetermined).

### Checking Pos 1.

This will need coding into the HLSysCtrl part of the Hardware Event Handler HardwareIntHandler, where TTPosN events are handled.

In version 1.05 it was not possible to make an OSL measurement using the Calibration LED as the stimulation source. It is useful to be able to do this for testing system performance etc.

The reason why it was not possible is that in the routine *StartOSL* in unit *MS\_Data.Pas* there was a line of code that converted an input to the OS command with the parameter 'C' (for calibration LED) to 'IR'. This is not a valid light source, and caused the OSL command to fail to take any data! The problem was simply solved by removing this line. This allowed the *StartOSL* routine to pass the parameter 'C' to the routine *LightStarterStopper* correctly.

This was changed and included in version 1.06 of the MiniSys software.

**Jul. 1998** During the development of the XY system a new instruction was required to allow one of the analogue inputs to be measured during a monochromator scan instead of the PMT (as would occur during a MS command). The new instruction is called 'SR' - short for Scan Reflectance. The syntax is very similar to MS, but with one additional parameter that defines which analogue input to read.

**SR** *id s ew r p ch* Perform a monochromator scan from the current wavelength to *ew*. During this the value at the analogue input *ch* is recorded. The turntable must be on position and the monochromator must have been previously reset.

*id* = monochrometer ID\*  
*s* = light source ID†  
*ew* = ending wavelength (nm)  
*r* = scan rate (nm/s)  
*p* = number of data  
*ch* = analogue channel to read

The modifications to the MiniSys code were as follows:-

- 1) The new instruction was added in the *ExecuteLowPriority* routine of the unit *hlsysctr.pas*.
- 2) A new AcquisitionCode was inserted called *ReflDAQ*. This allowed specific code to be written to handle the data acquisition.
- 3) A new routine called *FastAtoD* to read the analogue to digital converter was written in *hlsysctr.pas*. The standard routine (*ReadAtoD*) starts the AtoD conversion, and then waits 5ms before polling the AtoD convertor to see whether a result is ready. This is too slow for the SR routine. *FastAtoD* copes as follows. When a call is made to *FastAtoD* the current value is read, and just before exiting the routine, the next conversion is started. This means that the 'dead time' of the conversion occurs while the delay until the next acquisition occurs. This is ideal for fast conversions, but is likely to create an offset of the data when data acquisition is slow.
- 4) In *hlsysctr* the *PMTCounterISR* routine was altered so that during data acquisition the AtoD is read instead of the PMT counter.
- 5) A new routine, *StartReflScan*, was added to *ms\_data.pas*. This sets up the parameters for the scan. The new variable *AtoDChannel* keeps track of which AtoD channel is to be monitored during the scan.
- 6) The version number was changed to 1.06 in *ms\_ports.pas*.

Version 1.04 Nov. 8, 1996

Prior to this version the Mini-Sys could not properly recover from a lift timeout condition unless the timeout status had been cleared by reading status byte 5. In version 1.04 the status byte is cleared automatically whenever a



timed process is started. These processes include lift operations, turntable operation and data acquisitions.

Also, some minor changes were made so that the Mini-Sys and the TL-DA driver would be more compatible. These changes include not clearing the command Queue upon receipt of a 'CL' command. In the Mini-Sys, the 'CL' command has no effect and in the TL-DA driver it simply has the effect of clearing the serial output buffer. If it is necessary to clear the command queue then a 'CA' command must be sent.

Modifications prior to 1997 were by BGM, after that by GATD after that by LAPI...

#### Version 1.03 Oct. 23, 1996

It was discovered that the windows software sent a real number for the number of seconds in a pause command (PA) if the time exceeded 999sec. In other words, if a delay of 900 seconds was requested then the command "PA 900" was sent, if however, a delay of 1800 seconds was requested then the command "PA 1.8E3" was sent. Since 1.8E3 is a real number the Mini-Sys generated an error and skipped the pause. The pause command has now been modified to accept real parameters. Commands such as "PA 1.8E3" or "PA 5.6" are now acceptable.

#### Version 1.02 Sept. 26, 1996

In previous versions the heater DAC was always incremented by 5 regardless of the heating rate. In this version the heater DAC is incremented by 1 if the total heat time exceeds 200 seconds. This allows a more linear heating for low heating rates.

Also, in this version, a new parameter has been added (#16). This parameter allows one to define an irradiator offset time which helps correct for the time delay involved in activating and deactivating the irradiator. The value of parameter #16 is added to the Beta irradiation time. For example, if it is known that there is a 0.4sec delay in the irradiator then parameter #16 should be set to 400ms. Whenever a Beta irradiation is performed the Mini-Sys will automatically add 400ms (0.4sec) to the irradiation time.

#### Version 1.01 Sept. 19, 1996

Corrected implementation the TOL and POSL commands. In version 1.00 the light source did not toggle on and off correctly.

#### Version 1.00 completed and released September 1996