

10.3 RS232 Interface (X112/X113)

10.3.1 Hardware

All the control signals of the control module can be read and written through the RS232 interface.

Every GENIUS control module is equipped with two RS232 interfaces, meaning a data chain can be created with several control modules and a computer. The interface isolates the control module and computer electrically.



The plugs and cables connected to the interface must be shielded. The shielding must be connected to the housing ground.

Every GENIUS can take over the function of the MASTER (starts transmission) or SLAVE (replies to the master) at both interfaces. An overriding computer always has the status of master.

| Pin | Master interface X112 (9-pin socket) | Slave interface X113 (9-pin plug) |
|-----|---|--------------------------------------|
| 1 | | |
| 2 | TxD | RxD |
| 3 | RxD | TxD |
| 4 | Connected to pin 6 | Connected to pin 6 |
| 5 | RS232_GND | RS232_GND |
| 6 | Connected to pin 4 | Connected to pin 4 |
| 7 | Connected to pin 8 | Connected to pin 8 |
| 8 | Connected to pin 7 | Connected to pin 7 |
| 9 | | |

10.3.2 Protocol

The protocol of the interface is structured in such a way that the master first sends a number of characters, to which the slave answers. If the slave does not answer (pause > 100ms) or it answers with an error code, the process must be repeated (first wait approx. 50ms). Only after the fifth failed transmission attempt by the master is it assumed that there is a transmission fault that has to be dealt with by the master.

- RS232 with TxD and RxD
- Transmission rate: maximum 19 200 Baud
- Parity: none
- Data bits: 8
- Stop bits: 1

10.3.2.1 Explanation of characters

- 'z' Target address (`abc...z`) =PC, a=GENIUS1, b=GENIUS2, ...
- 's' Value > 31: Checksum
Value < 32: Error code

The checksum of a telegram is OK if the following equation is true:
 $255 \& (\text{sum of all transmitted characters excluding 'EOT'}) = 0 \text{ or } 32$

- 'q' Source address (`abc...z`) =PC, a=GENIUS1, b=GENIUS2, ...
- 'o' Number of the object
- 'd' Number of the datum
- 'c' Characters (all characters between 1 and 255 excluding the control characters listed).
Numbers are transferred in hex (2, 4 or 8 characters).
- SO Control character 14
Start of a transmission to write data
- SI Control character 15
Start of a transmission to read data
- ACK Control character 6
Feedback from slave to master
- EOT Control character 4
Final character
- NAK Control character 21
Data transmission aborted

10.3.2.2 Telegram structure

Send data to slave

The master sends:

| | | | | | | | | |
|-----|----|-----|-----|-----|-----|-----|--------------|-----|
| 'z' | SO | 's' | 'q' | 'o' | 'd' | 'c' | Further data | EOT |
|-----|----|-----|-----|-----|-----|-----|--------------|-----|

The slave replies:

Data received and accepted:

| | | | |
|-----|-----|-----|-----|
| 'q' | ACK | 's' | EOT |
|-----|-----|-----|-----|

Data was not accepted:

| | | | |
|-----|-----|---|-----|
| 'q' | ACK | 's' < 32 (error code) There is an error. | EOT |
|-----|-----|---|-----|

Read data from slave

The master sends:

| | | | | | | |
|-----|----|-----|-----|-----|-----|-----|
| 'z' | SI | 's' | 'q' | 'o' | 'd' | EOT |
|-----|----|-----|-----|-----|-----|-----|

The slave replies:

Query received and accepted:

| | | | | | |
|-----|-----|-----|-----|--------------|-----|
| 'q' | ACK | 's' | 'c' | Further data | EOT |
|-----|-----|-----|-----|--------------|-----|

Query was not accepted:

| | | | |
|-----|-----|---|-----|
| 'q' | ACK | 's' < 32 (error code) There is an error. | EOT |
|-----|-----|---|-----|

10.3.2.3 Error codes

| Error number | Name | Explanation |
|--------------|-----------|--|
| 1 | Object_No | Unknown object number |
| 2 | Datum_No | Unknown datum number |
| 3 | Type | The data type is not identical. |
| 4 | Access | There is no access to the datum selected, e.g. datum may only be read or datum is protected by a password. |
| 5 .. 31 | - | Free |

10.3.3 Data

Name: Name of the datum

No.: Number of the datum, specified as ASCII characters or decimal number

Type: t: Text with 1 to 8 characters followed by a zero
 b, w, u: Datum without sign (byte, word, ulong)
 c, s, l: Datum with sign (char, short, long)
 *: Only for Pro SW

Range: Value range of the datum (- = all possible numbers)
 Attention!! System errors can occur if the value range is not kept to!

Unit: Resolution or unit of the datum

10.3.3.1 System constants

The constants are set during installation of the system and only have to be changed in the event of a system change.

Object number: 32 (in hex = 20H, ASCII character = blank)

| Name | No. | Type | Range | Resolution | Description |
|-----------------|-----|------|-----------------------|------------|--|
| Serial | 'A' | w | - | - | Not yet implemented |
| Version | 'B' | w | - | - | (Read only) Version number of the software; H-byte main version, L-byte sub-version |
| CRC | 'C' | w | - | - | Not yet implemented |
| Address | 'D' | c | 'a'...'z' | - | Own address (addressing with address 'a' is always possible) |
| Code | 'E' | w | - | - | Logon code H-byte for level 2, L-byte for level 1. Every 2 bits specify a digit (0...3). (Default 8d55 -> code for level 1 = 1111, code for level 2 = 2031) |
| User_Level | 'F' | w | - | - | Logon level: 0 = logged off, 1 = user, 2 = service, (100 = switches the CRC check off) |
| Pro | 'G' | t | "" or "Pro" | - | (Read only) Pro SW is active? |
| Language | 'L' | c | 0, 'D', 'F', 'E', 'I' | - | Language: 0 = English, 'G' = German 'F' = French, 'S' = Spanish 'I' = Italian |
| Contrast | '@' | b | 0 ... 7 | - | LCD contrast |
| HV_Min | 'N' | w | 0 ... HV_Max | 1V | Min. high voltage |
| HV_Max | 'O' | w | HV_Min ... 10000 | 1V | Max. high voltage |
| Max_Emission | 'P' | w | 1 ... 10000 | 0.1mA | Maximum emission current |
| Number_Guns | 'Q' | b | 1, 2, 3 | - | No. of guns connected to the Genius |
| Voltage_Timeout | 'R' | w | 10 ... 3000 | 1ms | Maximum time the voltage may deviate from the high voltage setpoint by more than 500V |
| Water_Timeout | 'S' | w | 0 ... 10000 | 1ms | Time until an open water interlock leads to switching off |
| Step_Emission | 'T' | b | 1 ... 50 | 1% | Step width in percent of data. Max_Emission when the emission button is turned further by one step. |
| Step_percent | 'V' | b | 1 ... 50 | 1% | Step width of all percent data |
| Slave_Address | 'H' | c | 0, 'a' ... 'z' | - | Address of connected slave 0 or 'a' = no slave (from FW1.3) |

| | | | | | |
|----------------------|-----|---|-------------|---------|---|
| Master_Slave | 'l' | b | 0, 1, 2 | - | Specifies whether several devices are connected together for simultaneous evaporation (from FW1.3). 0 = standard 1 = master mode 2 = slave mode |
| Fil_Ramp | 'J' | w | 1 ... 10000 | 0,1 A/s | When the AFC level has been undercut, the filament current is run down with the ramp (from FW1.3) |
| HV_for_Fil_Ramp | 'K' | b | 0% ... 99% | 1 % | Start point for the AFC (from FW1.3) |
| ADC_Max_Emission | '1' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Max_Emission10 | '2' | w | - | - | Only for FERROFLUIDICS T Service |
| ADC_Max_Voltage | '3' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Max_Filament | '4' | w | - | - | Only for FERROFLUIDICS Service |
| OUT_ADC_Max_Voltage | '5' | w | - | - | Only for FERROFLUIDICS Service |
| OUT_ADC_Max_Filament | '6' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Max_Current_X | '7' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Max_Current_Y | '8' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Max_FPS_Emission | '9' | w | - | - | Only for FERROFLUIDICS Service |
| ADC_Offset_E10 | '0' | w | - | - | Only for FERROFLUIDICS Service |
| OUT_CPU | 'a' | b | - | - | Status of the outputs after switching on of the device (for further info see under "Actual values") |
| OUT_K1 | 'b' | b | - | - | ... |
| OUT_K2 | 'c' | b | - | - | ... |
| OUT_K3 | 'd' | b | - | - | ... |
| OUT_K4 | 'e' | b | - | - | ... |
| OUT_T11 | 'f' | b | - | - | ... |
| OUT_T12 | 'g' | b | - | - | ... |
| OUT_T21 | 'h' | b | - | - | ... |
| OUT_T22 | 'i' | b | - | - | ... |
| OUT_T1_ON | 'j' | b | - | - | ... |
| OUT_T2_ON | 'k' | b | - | - | ... |
| OUT_T1_PWM | 'l' | b | - | - | ... |
| OUT_T2_PWM | 'm' | b | - | - | ... |
| Pocket_in_Select | 'n' | b | 0,1,2 | - | Defines the function of the pocket selection (Source_Pocket_x) 0: No. of the input corresponds to no. of the pocket. If no input is active, the pocket can be changed via the remote control or RS232 1: Input code 0 ... 63 corresponds to pocket 1 ... 64 2: Input code 1 ... 64 corresponds to pocket 1 ... 64 In the case of input code 0 the pocket can be changed via the remote control or RS232 |

10.3.3.2 Configuration of the inputs and outputs

The Genius has inputs and outputs (IO), with a distinction being drawn between real and internal IOs. The *real IOs* are located on X110 of the CPU card or on the IO cards. Up to five cards are supported. *Internal IOs* are internal variables treated by the software like real IOs.

In order to be able to link the inputs and outputs, there is a variable in the range of the constants for every output stating to which input the output should react.

Outputs

Object number: 32 (in hex = 20H, ASCII character = blank)

| Name | No. | Type | Description |
|----------------------------------|-----|------|--|
| Source_A_CPU1 | 'o' | w | Output 1 of the CPU card |
| Source_A_CPU2 | 'p' | w | Output 2 of the CPU card |
| Source_A_K1_CARD1 | 128 | w | Relay output K1 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_K1_CARD5 | 132 | | |
| Source_A_K2_CARD1 | 134 | w | Relay output K2 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_K2_CARD5 | 137 | | |
| Source_A_K3_CARD1 | 138 | w | Relay output K3 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_K3_CARD5 | 142 | | |
| Source_A_K4_CARD1 | 143 | w | Relay output K4 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_K4_CARD5 | 147 | | |
| Source_A_T1_ON_CARD1 | 148 | w | Switch on transistor T1 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_T1_ON_CARD5 | 152 | | |
| Source_A_T1_PWM_CARD1 | 153 | w | Switch transistor T1 of IO card 1 to IO card 5 into PWM mode |
| ... | ... | | |
| Source_A_T1_PWM_CARD5 | 157 | | |
| Source_A_T11_CARD1 | 158 | w | Output 1 of transistor T1 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_T11_CARD5 | 162 | | |
| Source_A_T12_CARD1 | 163 | w | Output 2 of transistor T1 of IO card 1 to IO card 5 |
| ... | ... | | |
| Source_A_T12_CARD5 | 167 | | |
| Source_A_T2_ON_CARD1 | 168 | w | Transistor T2 |
| ... | ... | | |
| Source_A_T22_CARD5 | 187 | | |
| Source_HV_on | 'r' | w | A change from 0 to 1 switches HV-ON |
| Source_HV_off | 's' | w | When active: switch off HV |
| Source_Fil_on | 't' | w | When active: switch on filament |
| Source_Magnet_ON | 203 | w | When active: switch on magnet (from FW 1.3) |
| Source_GUN_ON | 188 | w | 1 = gun is active (from FW 1.4) 0 = gun off => emission inhibited, filament current = 0 A, water and magnet and filament would not be checked. |
| Source_Set_Auto | '.' | w | When active: switch over to automatic |
| Source_Set_Manual | '' | w | When active: switch over to manual |
| Source_Emission_release_external | 'u' | w | When active: emission enabled (e.g.: when the pocket is controlled externally) |
| Source_Interlock_Chamber | 'v' | w | When active: chamber is closed |
| Source_Interlock_External | 'w' | w | When active: an additional interlock is OK |
| Source_Pocket_rotate | 'x' | w | Pocket rotates constantly (only if the number of pockets is set on 1) |

| | | | |
|------------------------|---------|---|---|
| Source_Error_Quit | 'y' | w | A change from L to H acknowledges an error or a warning (from FW 1.3) |
| Source_Switch1_Set | 'z' | w | 0 = switch 1 remains unchanged, 1 = switch 1 is set (from FW 1.4) |
| Source_Switch1_Reset | '{' | w | 0 = switch 1 remains unchanged, 1 = switch 1 is reset (from FW 1.4) |
| Source_Switch1_Toggle | ' ' | w | An LH change changes the contents of switch 1 (from FW 1.4) |
| Source_Switch2_Set | '}' | w | 0 = switch 2 remains unchanged, 1 = switch 2 is set (from FW 1.4) |
| Source_Switch2_Reset | '~' | w | 0 = switch 2 remains unchanged, 1 = switch 2 is reset (from FW 1.4) |
| Source_Switch2_Toggle | 127 | w | An LH change changes the contents of switch 2 (from FW 1.4) |
| Source_Pocket_external | 'q' | w | When active, pocket selection via digital inputs is possible |
| Source_Pocket_1 | 191 | w | Specifies which inputs are used for pocket selection. The allocation is defined by the constant "Pocket_in_Select". |
| ... | ... | | |
| Source_Pocket_12 | 202 | | |
| Source_Data_external | 'X' | w | When active, pocket selection via digital inputs is possible (when pocket and data set have been separated). |
| Source_Data_1 | 'Y'-'Z' | w | Specifies which inputs are used for data selection. The allocation is defined by the constant "Pocket_in_Select". |
| ... | '['-'/' | | |
| Source_Data_6 | ']'-'^' | | |
| Source_AND1_IN1 | 207 | w | Logic function (from FW 1.3) |
| Source_AND1_IN2 | 208 | | |
| Source_AND1_IN3 | 209 | | |
| Source_AND2_IN1 | 210 | | |
| Source_AND2_IN2 | 211 | | |
| Source_AND2_IN3 | 212 | | |
| Source_OR1_IN1 | 213 | w | Logic function (from FW 1.3) |
| Source_OR1_IN2 | 214 | | |
| Source_OR1_IN3 | 215 | | |
| Source_OR2_IN1 | 216 | | |
| Source_OR2_IN2 | 217 | | |
| Source_OR2_IN3 | 218 | | |
| Source_MUX1_Function | 219 | w | Control input for Mux 1 (from FW 1.4) |
| Source_MUX1_IN_Low | 220 | w | User input for Mux 1 when the control input = 0 (from FW 1.4) |
| Source_MUX1_IN_High | 221 | w | User input for Mux 1 when the control input = 1 (from FW 1.4) |
| Source_MUX2_Function | 222 | w | Control input for Mux 2 (from FW 1.4) |
| Source_MUX2_IN_Low | 223 | w | User input for Mux 2 when the control input = 0 (from FW 1.4) |
| Source_MUX2_IN_High | 224 | w | User input for Mux 1 when the control input = 1 |

Coding of the inputs

Object number: 32 (in hex = 20H, ASCII character = blank)

When bit 15 in the data word is set, the input is inverted.

Data word = 0000H -> the output is not changed (open)

Data word = 0100H -> the output is always 0 (off)

Data word = 8100H -> the output is always 1 (on)

Data word = 0001H -> the output corresponds to CPU input 1

Data word = 0002H -> the output corresponds to CPU input 2

Data word = 0003H -> the output corresponds to CPU input 3

Data word = 0004H -> the output corresponds to CPU input 4

Data word = 0005H -> the output corresponds to CPU input 5

Data word = 0006H -> the output corresponds to CPU input 6

Data word = 0101H -> the output corresponds to input 1 of IO card 1

Data word = 0102H -> the output corresponds to input 2 of IO card 1

...

Data word = 010CH -> the output corresponds to input 12 of IO card 1

Data word = 0201H -> the output corresponds to input 1 of IO card 2

...

Data word = 050CH -> the output corresponds to input 12 of IO card 5

Data word = 1000H -> the output is used for the pocket drive

Data word = 1001H -> the output is 1 when HV is switched on

Data word = 1002H -> the output is 1 when there is an error

Data word = 1003H -> the output is 1 when there is a warning

Data word = 1004H -> the output is 1 when emission is enabled

Data word = 1005H -> the output is 1 when all interlocks are OK

Data word = 1006H -> the output corresponds to the value of "Auto" (actual values)

Data word = 1007H -> the output corresponds to the value of "Status_Magnet" (actual values) (from FW1.3)

Data word = 1008H -> the output corresponds to the value of "Status_FPS" (actual values) (from FW1.3)

Data word = 1009H -> the output corresponds to the value of "Status_HVP" (actual values) (from FW1.3)

10.3.3.3 Gun data

Explanation: Data and constants relating to the guns.

The data type is designed for up to three guns.

Object number for gun 1: 33 (in hex = 21H, ASCII character = '!')

Object number for gun 2: 34 (in hex = 22H, ASCII character = ' " ')

Object number for gun 3: 35 (in hex = 23H, ASCII character = '#')

| Name | No. | Type | Range | Resolution | Description |
|----------------------------|-----------|--------|------------------------------------|------------|---|
| Gun_Type | '0' | b | 0... | - | Type of gun 0 = (for gun 1) demo 0 = (for gun 2 or 3) not used 1 = EV M-6 2 = EV M-5 3 = EV M-8 4 = EV 4-12 5 = EV 1-8 6 = EV 1-63 7 = EV 1-15 8 = EB3 9 = EB1 10 = no rotary pocket 11 = unknown rotary pocket (from FW1.4) |
| Pocketnumber | '1' | b | 1 ... 12 | - | Number of pockets |
| Pocketspeed | '2' | b | 1 ... 255 | ? | Rotation speed |
| Max_Power | '3' | w | 100 ... 10000 | 1W | Maximum power of the gun |
| Fil_Min | '4' | w | 0 ... < Fil_Max | 1A | Filament minimum |
| Fil_Max | '5' | w | Fil_Min < ... 50000 | 1A | Filament maximum |
| Min_X_Current | '6' | s | -3000 ... Max_X_S. | 1mA | Minimum magnet current for X |
| Max_X_Current | '7' | s | Min_X_S. ... 3000 | 1mA | Maximum magnet current for X |
| Min_Y_Current | '8' | s | -3000 ... Max_Y_S. | 1mA | Minimum magnet current for Y |
| Max_Y_Current | '9' | s | Min_Y_S. ... 3000 | 1mA | Maximum magnet current for Y |
| Step_X_Limit | 'A' | s | -200 ... 200 | 1mA | Step width for limit adjustment |
| Step_Y_Limit | 'B' | s | -200 ... 200 | 1mA | Step width for limit adjustment |
| Controller_PH | 'J' | | 0 ... 10000 | - | P-share of the controller for high emission (I > 300 mA) |
| Controller_PL | 'K' | w | 0 ... 10000 | - | P-share of the controller for low emission (I < 30 mA) |
| Controller_I | 'L' | w | 0 ... 1000 | - | I-share of the controller |
| Limit_X_V | 'M' | s | Min_X_Current ... Max_X_Current | 1mA | Global limit for X front |
| Limit_X_H | 'N' | s | Min_X_Current ... Max_X_Current | 1mA | Global limit for X back |
| Limit_Y_L | 'O' | s | Min_Y_Current ... Max_Y_Current | 1mA | Global limit for Y left |
| Limit_Y_R | 'P' | s | Min_Y_Current ... Max_Y_Current | 1mA | Global limit for Y right |
| Voltage | 'Q' | w | HV_Min ... HV_Max | 1V | Global value for the voltage |
| Pocket1_POS Pocket2_POS | ('')' | w w | - | - | Pocket positions of the unknown rotary pocket (from FW1.4). |

| | | | | | |
|-------------|-----|---|--|--|---|
| Pocket3_POS | '*' | w | | | Pocket1_POS = code for pocket 1, ... Bit 0 = input 7, bit 1 = input 8, ..., bit 5 = input 12 When bit 15 = 1 => pendulum pocket, second turning point bit 8 to bit 13 |
| Pocket4_POS | '+' | w | | | |
| Pocket5_POS | '.' | w | | | |
| Pocket6_POS | '_' | w | | | |
| Pocket7_POS | ' ' | w | | | |
| Pocket8_POS | '/' | w | | | |

10.3.3.4 Actual values

The actual values reflect the current status as well as the status during the last 10 errors. The structure of the actual values is therefore set up a total of 11 times. When an error leading to switching off of the high voltage occurs, the GENIUS places the actual values in object number 37 for the last error and stores the previous actual values under the following object numbers.

Object number of the actual values: 36 (in hex = 24H, ASCII character = '\$')

Object number of the last error: 37 (in hex = 25H, ASCII character = '%')

Object number of the second-to-last error: 38 (in hex = 26H, ASCII character = '&')

...

Object number of the tenth-to-last error: 46 (in hex = 2EH, ASCII character = '.')

| Name | No. | Type | Range | Resolution | Description |
|---------------------------|------|------|--------------|------------|--|
| RS232_WD | ' ' | w | 0 ... 65535 | 10 ms | RS232 monitoring (from FW1.4). If RS232_WD is set on a value, the variable is counted down. When the variable 0 is reached, the high voltage is switched off with error no. 25. |
| Magnet_on | 'A' | b | 0, 1 | - | 1 = switch on magnet |
| Fil_on | 'B' | b | 0, 1 | - | 1 = switch on magnet and filament |
| HV_on | 'C' | b | 0, 1 | - | 1 = switch on magnet, filament and voltage and enable emission |
| Gun_on | '=' | b | 0, 1 | - | 1 = gun is active (from FW1.4) 0 = gun off => emission inhibited, filament current = 0 A, water and magnet and filament would not be checked. |
| Emission_release_external | 'D' | b | 0, 1 | - | 0 = emission cannot be enabled |
| Error_Quit | 'E' | b | 0, 1 | - | Write with 1: error is acknowledged |
| Group_set | 'F' | b | 0 ... 49 | - | Write with a process index and the new process becomes active |
| Pocket_set | 'G' | b | 1 ... 64 | - | Write with a pocket number and the new pocket becomes active |
| Data_set | '@' | b | 1 ... 64 | - | If data set and pocket are separated: write with a data set number and the new data set becomes active |
| Auto_set | 'H' | b | 0, 1 | - | 0 or 1 transferred in 'Auto' |
| Save_Data | 'I' | b | 0, 1 | - | Write with 1: the actual data set is saved |
| Target_Emission | 'J' | w | 0 ... 10000 | 0.1mA | Emission current desired |
| Out_lock | '\' | b | 0,1,2 | - | 0 = outputs enabled 1 = outputs briefly inhibited after switching on of the device 2 = constant inhibit |
| PocketTest | '[' | b | 0, 1 | - | 0 = normal, 1 = pocket in test mode PocketSpeed can be written |
| Switch1 | '"' | b | 0,1 | - | Buffer variable of IO control (from FW 1.3) |
| Switch2 | '(' | b | 0,1 | - | Buffer variable of IO control (from FW 1.3) |
| Switch3 | ')' | b | 0,1 | - | Buffer variable of IO control (from FW 1.3) |
| Switch4 | '*' | b | 0,1 | - | Buffer variable of IO control (from FW 1.3) |
| Speed1 | '\$' | c | -100 ... 100 | 1% | Speed of rotary pocket 1 |
| Speed2 | '%' | c | -100 ... 100 | 1% | Speed of rotary pocket 2 |
| Speed3 | '&' | c | -100 ... 100 | 1% | Speed of rotary pocket 3 |
| PWM_Value_T1 | 'n' | w | 0 ... 65536 | - | Pulse-pause ratio for transistor 1 |

| | | | | | |
|--|-----|---|-------------|---|--|
| PWM_Value_T2 | 'o' | w | 0 ... 65536 | - | Pulse-pause ratio for transistor 2 |
| OUT_CPU | 'a' | b | - | - | Bit 0 = relay output 1 of CPU card Bit 1 = relay output 2 of CPU card |
| Bit allocation for outputs on the IO cards: Bit 4 = IO card 1, bit 3 = IO card 2, bit 2 = IO card 3, bit 1 = IO card 4, bit 0 = IO card 5 | | | | | |
| OUT_K1 | 'b' | b | - | - | Relay output K1 to K4 |
| ... | ... | | | | |
| OUT_K4 | 'e' | | | | |
| OUT_T11 | 'f' | b | - | - | Output 1 of transistor 1 |
| OUT_T12 | 'g' | b | - | - | Output 2 of transistor 1 |
| OUT_T21 | 'h' | b | - | - | Output 1 of transistor 2 |
| OUT_T22 | 'i' | b | - | - | Output 2 of transistor 2 |
| OUT_T1_ON | 'j' | b | - | - | Switch on transistor 1 |
| OUT_T2_ON | 'k' | b | - | - | Switch on transistor 2 |
| OUT_T1_PWM | 'l' | b | - | - | Switch transistor 1 into PWM mode |
| OUT_T2_PWM | 'm' | b | - | - | Switch transistor 2 into PWM mode |
| State_Master | ':' | b | 0, 1 | - | Status of master (for simultaneous evaporation) |
| The following data can only be read. | | | | | |
| State | 'K' | b | 0 ... 4 | - | Status of control: 0 = all off 1 = magnet on 2 = magnet and filament on 3 = magnet, filament and HV on 4 = emission enabled |
| State_External | 'L' | b | 0, 1 | - | 1 = additional interlock OK |
| State_Chamber | 'M' | b | 0, 1 | - | 1 = chamber interlock OK |
| State_Vacuum | 'N' | b | 0, 1 | - | 1 = vacuum interlock OK |
| State_Water | 'O' | b | 0, 1 | - | 1 = water interlock OK |
| State_Magnet | 'P' | b | 0, 1 | - | 1 = magnet OK (only changed if magnet is on) |
| State_FPS | 'Q' | b | 0, 1 | - | 1 = FPS OK (only changed if FPS is on) |
| State_HVP | 'R' | b | 0, 1 | - | 1 = HVP OK (only changed if HVP is on) |
| State_Slave | ',' | b | 0, 1 | - | 1 when State_External, State_Chamber, State_Vacuum and State_Water are 1 |
| State_all | '<' | w | | | Summary of all statuses (from FW1.3): Bit 0: State_External Bit 1: State_Chamber Bit 2: State_Vacuum Bit 3: State_Magnet Bit 4: State_FPS Bit 5: State_HVP Bit 6: State_Slave Bit 7: State_Master Bit 8: ErrorNumber != 0 Bit 9: WarningNumber != 0 Bit 10: Auto Bit 11: Reserve Bit 12: Emission is enabled (Status = 4) Bit 13: High voltage is on (Status >= 3) Bit 14: FPS is on (Status >= 2) Bit 15: Magnet is on (Status >= 1) |
| All_Actual | '^' | t | | | Summary of selected actual values, output as text with hex numbers (from FW 1.3): 4 characters: State_all 4 characters: Voltage 4 characters: Actual_Emission |

| | | | | | |
|---|------|---|--------------|-------|---|
| | | | | | 4 characters: Target_Emission 4 characters: FilCurrent 4 characters: X_Current 4 characters: Y_Current 2 characters: Pocket |
| All_limit | '_' | t | | | Summary of limits (data set), output as text with hex numbers (from FW 1.3): 4 characters: Voltage 4 characters: Max_Emission 4 characters: Limit_X_V 4 characters: Limit_X_H 4 characters: Limit_Y_L 4 characters: Limit_Y_R |
| All_Pos | '' | t | | | Summary of selected data, output as text with hex numbers (from FW1.3): 4 characters: X_Frequency 4 characters: Y_Frequency 2 characters: X_Position 2 characters: Y_Position 2 characters: X_Amplitude 2 characters: Y_Amplitude |
| Auto | 'S' | b | 0, 1 | - | 0 = manual, 1= automatic |
| ErrorNumber | 'T' | w | 0 ... | - | The actual error number |
| WarningNumber | 'U' | w | 0 ... | - | The actual warning number |
| Actual_Group | 'V' | b | 0 ... 49 | - | Index of the actual process |
| Process | 'W' | t | 8 char + 0 | - | Name of the actual process |
| Material | 'X' | t | 8 char + 0 | - | Name of the actual material |
| Data | 'Y' | t | 8 char + 0 | - | Not supported (Use name from data set 0!) |
| Emission_release_internal | 'Z' | b | 0, 1 | - | 0 = emission cannot be enabled because inhibited internally (the pocket is not in position) |
| Emission_release_HV | 'J' | b | 0, 1 | - | 0 = emission cannot be enabled because inhibited internally (the HV is not OK) |
| Data_Index | '/' | b | 1 ... 64 | - | Index of the actual data set |
| Pocket | '0' | b | 1 ... 64 | - | The actual pocket |
| Pocket_Text | '1' | t | e.g. "3B" | - | The actual pocket |
| Data_changed | '2' | t | " ", "?" | - | "?" = data has been changed but not yet saved |
| Actual_Emission | '3' | w | 0 ... 10000 | 0.1mA | Actual emission |
| Voltage | '4' | w | 0 ... 10000 | 1V | Actual voltage |
| FilCurrent | '5' | w | 0 ... 50000 | 1A | Actual filament current |
| Gun | '6' | b | 1, 2, 3 | - | Actual gun |
| Real_Pocket | '7' | b | 1 ... 12 | - | Actual real pocket |
| X_Current | '8' | s | -3000...3000 | 1mA | Actual magnet current X |
| Y_Current | '9' | s | -3000...3000 | 1mA | Actual magnet current Y |
| IN_CPU | 'p' | b | - | - | Bit 0 = CPU input 1 Bit 1 = CPU input 2 ... Bit 6 = CPU input 6 |
| Bit allocation for inputs on the IO cards: Bit 0 = IO card 1, bit 1 = IO card 2, bit 2 = IO card 3, bit 3 = IO card 4, bit 4 = IO card 5 | | | | | |
| IN_0 | 'q' | b | - | - | Input 1 to input 12 of the IO cards |
| ... | ... | | | | |
| IN_9 | 'z' | | | | |
| IN_10 | '!'' | | | | |
| IN_11 | '?' | | | | |
| DAC_1 | '{' | w | - | - | Will only be used in future software |

| | | | | | |
|-------|-----|---|---|---|---|
| DAC_2 | ' | | | | versions |
| DAC_3 | ' | | | | |
| DAC_4 | ' | | | | |
| ADC_1 | 127 | w | - | - | Will only be used in future software versions |
| ADC_2 | 128 | | | | |
| ADC_3 | 129 | | | | |
| ADC_4 | 130 | | | | |
| ADC_5 | 131 | | | | |
| ADC_6 | 132 | | | | |

10.3.3.5 Data set

The data set contains the evaporation parameters.

Up to 99 data sets can be saved. The actual data set is copied into the work data set. A change in the work data set has a direct effect on the process. If the active data set is not the same as the work data set, the variable Data_changed = "?" is set (data should be saved).

In the first data set the active data set is copied (with the name '_backup_') before it is overwritten during saving.

A data set is deleted if the 'Name' begins with a zero or blank.



Data set parameters whose type is marked with „“ are only supported by the model GENIUS PRO from Firmware Version 1.2.*

Object number of the work data set: 47 (in hex = 2fH, ASCII character = '/')

Object number of the first data set: 48 (in hex = 30H, ASCII character = '0')

Object number of the second data set: 49 (in hex = 31H, ASCII character = '1')

...

Object number of the 99th data set: 146 (in hex = 92H)

| Name | No. | Type | Range | Resolution | Description |
|-----------------|-----|------|---------------------------|------------|--|
| Name | '0' | t | 8 char + 0 | - | Name of the data set |
| Use_Global_Data | '1' | b | 0, 1 | - | 0 = local limits and voltage 1 = global limits and voltage Only for FSC card: 16 = local for gun 2 32 = local for gun 3 |
| Voltage | '2' | w | HV_Min HV_Max | 1V | Local value of the voltage |
| Max_Emission | '3' | w | 0 Const.Max_E. | 0.1mA | Limit of the maximum emission current |
| Auto_Emission | '4' | w | 0 ... 10000 | 0.1mA | Specifies the current arising at 10V in automatic mode |
| SweepMode | '8' | b* | 1, 2, 3, 4, 20, 21, 22 | - | 1 = sweep off 2 = sweep Lissajous 3 = sweep grid 4 = circle / spiral (up to FW 1.2) 20 = spiral (centre cold) (up to FW 1.2) 21 = spiral (centre normal) (up to FW 1.2) 22 = spiral (centre hot) |
| SpiralSort | '9' | c | -1, 0, 1, ..., 15 | - | -1 = self-defined spirals (from FW1.3) 0 = circle 1 = spiral centre cold 8 = spiral centre normal 15 = spiral centre hot |
| X_Waveform | 'A' | b* | 0, 1, 2, 3, 4 | - | Specifies the waveform 0 = sine 1 = triangle 2 = square 3 = rectified sine 4 = self-defined |

| | | | | | |
|--|--|--------|------------------------------------|----------|---|
| Y_Waveform | 'B' | b | 0, 1, 2, 3, 4 | - | Like X_Waveform |
| Limit_X_V | 'C' | s | Min_X_Current ... Max_X_Current | 1mA | Local limit for X front |
| Limit_X_H | 'D' | s | Min_X_Current ... Max_X_Current | 1mA | Local limit for X back |
| Limit_Y_L | 'E' | s | Min_Y_Current ... Max_Y_Current | 1mA | Local limit for Y left |
| Limit_Y_R | 'F' | s | Min_Y_Current ... Max_Y_Current | 1mA | Local limit for Y right |
| X_Position | 'G' | c | -100 ... 100 | 1% | Position of the beam in X-direction. The data relates to the limits. |
| Y_Position | 'H' | c | -100 ... 100 | 1% | Position of the beam in Y-direction. |
| X_Frequency | 'I' | w | 10 ... 10000 | 0.01Hz | Sweep frequency for X |
| Y_Frequency | 'J' | w | 10 ... 10000 | 0.01Hz | Sweep frequency for Y |
| X_Amplitude | 'K' | b | 0 ... 100 | 1% | Sweep amplitude for X Position(value) + amplitude may not exceed 100% |
| Y_Amplitude | 'L' | b | 0 ... 100 | 1% | Sweep amplitude for Y |
| Defocus_Amplitude | 'M' | w* | 0 ... 25 | 1% | Amplitude of defocus circle 0 = defocus off |
| Defocus_Frequency | 'N' | c* | 10 ... 15000 | 0.01Hz | Frequency of defocus circle |
| Pulse0 Pulse1 Pulse2 Pulse3 Pulse4 Pulse5 Pulse6 Pulse7 | 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' | b* | 10 ... 200 | 1% | When SpiralSort = -1 the dwell time can be changed (from FW1.3). All values 100 => spiral centre cold. Pulse0 = centre ... Pulse7 = outside |
| CircleSector0 CircleSector45 CircleSector90 CircleSector135 CircleSector180 CircleSector225 CircleSector270 CircleSector315 | 'X' 'Y' 'Z' '[' '\ ']' '^' '_' | b* | 10 ... 200 | 1% | Specifies the energy distribution to the circle sector in the sweep mode "circle" or "spiral". |
| Grid_0_0 Grid_0_1 ... Grid_0_7 Grid_1_0 ... Grid_7_7 | 160 161 ... 167 168 ... 223 | b* | 0 ... 200 | 1% | Specifies the energy distribution to the 64 fields in the sweep mode "grid". Allocation: <i>Grid_7_0, ..., Grid_7_7</i> ... <i>Grid_0_0, ..., Grid_0_7</i> Filament |
| X_Function_0 ... | 96 ... | c c | -100 ... 100 -100 ... 100 | 1% 1% | 32 values that define the waveform for the X-sweep |
| X_Function_31 | 127 | c | -100 ... 100 | 1% | |
| Y_Function_0 ... | 128 ... | c c | -100 ... 100 -100 ... 100 | 1% 1% | 32 values that define the waveform for the Y-sweep |
| Y_Function_31 | 159 | c | -100 ... 100 | 1% | |

10.3.3.6 Process

A process defines the allocation of pocket, material and data set.

A process is deleted if the 'Name' begins with a zero or blank.

If the data set and pocket are connected, a pocket is not used if 'Data_?' begins with a zero or blank.

Up to 50 processes can be saved.

Object number of the first process: 147 (in hex = 93H)

Object number of the second process: 148 (in hex = 94H)

...

Object number of the 50th process: 196 (in hex = C4H)

| Name | No. | Type | Range | Resolution | Description |
|----------------------|-----|------|------------|------------|---|
| Name | '0' | t | 8 char + 0 | - | Name of the process |
| Pocket_Data_separate | '/' | b | 0, 1 | - | 0 = every pocket is allocated a data set. 1 = pocket and data set can be changed separately of each other. |
| Material_1 | '1' | t | 8 char + 0 | - | Material in real pocket 1 |
| ... | ... | t | 8 char + 0 | - | Material in real pocket x |
| Material_36 | 'T' | t | 8 char + 0 | - | Material in real pocket 36 (3 guns with 12 pockets) |
| Data_1 | 'a' | t | 8 char + 0 | - | Data set for pocket 1 |
| ... | ... | t | 8 char + 0 | - | Data set for pocket x |
| Data_64 | 160 | t | 8 char + 0 | - | Data set for pocket 64 |

10.3.3.7 Examples

10.3.3.7.1 Example in C to test the checksum:

```

/* Parameter      : Telegram      */
/* Return value   : -1 = faulty sum */
Int Sum_Test(char *Telegram)
{
    char Sum = 0;
    int Index = 0;

    while (Telegram[Index] != EOT)
        Sum += Telegram[Index++];

    if ((Sum == 0) || (Sum == 32))
        return(0);
    else
        return(-1);
}

```

10.3.3.7.2 Example in C to enter the checksum in a telegram:

```

/* Parameter      : Telegram      */
void Sum_Test(char *Telegram)
{
    int Index;

    Telegram[2] = 0 - (Telegram[0] + Telegram[1]);
    for (Index=3; Telegram[Index] != EOT; Index++)
        Telegram[2] = Telegram[2] - Telegram[i];

    if ((byte)Telegram[2] < 32)
        Telegram[2] += 32;
}

```

10.3.3.7.3 Example of a telegram to switch on the high voltage:

In order to switch on the high voltage, the datum "HV_on" (number of the datum = ASCII 'C' or hex 43H) in the object "Actual values" (object number = Hex 24H) must be written with 1.

The address of the master computer is ASCII '^' or hex 60H.

The default address of the Genius is ASCII 'a' or hex 61H. If several Geniuses are connected to a master computer, the address must be changed with the remote control.

Data exchange in hex:

| | | | | | | | | | |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| The master computer sends: | 61H | 0eH | 69H | 60H | 24H | 43H | 30H | 31H | 04H |
| The Genius replies: | 60H | 06H | 9aH | 04H | | | | | |

Example of a telegram to read the value of the emission current:

The actual emission current is in datum "Actual_Emission" (number of the datum = ASCII '3' or hex 33H) in the object "Actual values" (object number = hex 24H).

Data exchange in hex:

| | | | | | | | | |
|----------------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| The master computer sends: | 61H | 0fH | d9H | 60H | 24H | 33H | 04H | |
| The Genius replies: | 60H | 06H | aeH | 30H | 42H | 42H | 38H | 04H |

Received were 30H = '0', 42H = 'B', 42H = 'B' and 38H = 8. That means the hex value for the emission is 0BB8H, which corresponds to 3000 * 0.1 mA.

Example of a telegram to change a process:

The data set "ABC" is to be assigned to the fourth pocket in the third process.
(Pocket and data set are coupled.)

Object of third process (object number = hex 95H)

Datum "Data_4" (number of the datum = ASCII 'd' or hex 64H)

Data exchange in hex:

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Send | 61H | 0eH | d2H | 60H | 95H | 64H | 41H | 42H | 43H | 20H | 20H | 20H | 20H | 20H | 00H | 04H |
| Reply | 60H | 06H | 9aH | 04H | | | | | | | | | | | | |

or:

| | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Send | 61H | 0eH | 72H | 60H | 95H | 64H | 41H | 42H | 43H | 00H | 04H |
| Reply | 60H | 06H | 9aH | 04H | | | | | | | |

Example of a telegram to delete a data set:

The fifth data set is to be deleted.

Object of fifth data set (object number = hex 34H)

Datum "Name" (number of the datum = ASCII '0' or hex 30H)

In the software version V1.2 the memory is not released correctly. After deleting via RS232 it can only be occupied via RS232.

Data exchange in hex:

| | | | | | | | | | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Send | 61H | 0eH | cdH | 60H | 34H | 30H | 20H | 20H | 20H | 20H | 20H | 20H | 20H | 20H | 00H | 04H |
| Reply | 60H | 06H | 9aH | 04H | | | | | | | | | | | | |

or:

| | | | | | | | | |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| Send | 61H | 0eH | cdH | 60H | 34H | 30H | 00H | 04H |
| Reply | 60H | 06H | 9aH | 04H | | | | |