

# HSAJET CU REMOTE COMMUNICATION



**A guide to remote communication with the HSAJET CU, CU2 and CUF controllers**

Last update 1 Mar 2011

# INTRODUCTION

The CU series is a powerful stand-alone controller with ability to perform almost any print task using the print jobs as templates and external communication through RS232 or Ethernet. Design layouts once, and fill with content as required. Select jobs. Change parameters. Ask status.

You do not get a pre-defined software package with CU. Instead, you get the building blocks which allow you to custom build a tailored solution for the individual application.

## ***What you should understand before continuing***

In order to understand the protocol for communicating with the CU, you should understand:

- Basic principles of request-response communication
- Fundamentals of RS232 and the ASCII character set, such as hexadecimal values and escape codes
- Ethernet communication principles such as IP address, subnet mask and sockets.

It does not matter what programming language you use to connect to the CU.

Perl, Ruby, VB.NET, C++, C# or AutoIT Script are all among possible alternatives.

**Please bear in mind that HSA can not support you in the above fundamentals. You should consult an external programmer if you do not fully understand this documentation..**

## ***Examples are available***

Even if HSA can not do the programming for you, we do have resources available you can learn from.

- CUMunicator: application to debug communication with CU.
- Scripts in Perl for CU
- Scripts in AutoIT3 code to demonstrate a front end.

## **Basics of communicating with the CU**

- All commands are terminated by “#” character
- For RS232: Use COM2 port. You must use a crossed cable (Null Modem), and use the following parameters:

Rate: 9600 bits/second  
 Stop bits: 1  
 Data bits: 8  
 Paritet: None  
 HW flow None

- For Ethernet: Use device IP address, port 1500. Password is same as parameter password, if set.
- CU will send a response to EVERY command you send with a result code. It is not NECESSARY to evaluate response, but strongly recommended.
- CU objects are grouped into numeric types, and then again by numbers. For example for counters (object type 1), you may have objects 1;1 1;2 and 1;3 for the 3 counters in your message. Additionally, objects have a descriptive name which you can **read** from the unit but NOT use to address objects. It is recommended that you after connecting to the unit ask which objects are available.

Object type	Object type number
Text	0
Counters	1
Time	2
Date	3
Barcode	4
Logo	5

- Commands are grouped by keyword. The following keywords are available:

Keyword	Function
CMD	Execute commands
REQ	Get information
OBJ	Change objects
PAR	Change parameters
FIL	Transmit a file

In the following section, each keyword will be detailed.

## Example of communication with CU

The following are examples of a communication session over Ethernet to a CU. Read from top down. RED is what is SENT. GREEN is what is RECEIVED.

Transmit	Receive	Description
CMD:C#		Remote connect with password = no password.
	RES:10#	Password accepted. Notice that this is RES:10 not RES:0# (zero)
CMD:R#		Start print mode.
	RES:0#	Command accepted.
OBJ:1;1;CUR;0#		Set counter object 1 current value to 0.
	RES:0#	Counter updated.
CMD:U#		Update next print buffer with any changes made (counter object 1 will be updated on next print) If CMD:U is not send after an update, there will be a 2 print delay before update.
	RES:0#	Print buffer updated.
CMD:S#		Stop print mode.
	RES:0#	Print mode ended.
CMD:D#		Disconnect.
	RES:0#	Disconnected.

### Example, showing response from CU to REQ

```
>>> Logging in ! (password set to 'a')
--> CMD:C;a#
RES:10#
>>> Starting print
--> CMD:R#
RES:0#
>>> Updating Object 0;1 to 4963
--> OBJ:0;1;TEX;4963#
RES:0#
>>> update
--> CMD:U#
RES:0#
>>> Asking for status <<<
--> REQ:status#
DAT:printactive;1#
DAT:selectedfile;TEST #
DAT:printcount;0#
DAT:systime;11;53;0;30;7;2010#
RES:0#
>>> Asking version <<<
--> REQ:version#
DAT:1.02;1.17;2.0.9;1.02#
RES:0#
>>> Asking object list <<<
--> REQ:object list#
DAT:0;1;CUText#
DAT:1;1;CUCount#
DAT:3;1;CUDate#
DAT:5;1;Background#
RES:0#
```

## ***Example, showing selection of different logos***

It is not possible to select different logo files from the CF card, but it IS possible to “layer” logos in the design process. You can use a maximum of 8 different CU logos.

These will be of type 5 (logo), numbered 2..9 since background of design is always logo no 1. Logos are numbered from bottom up in the object list.

When you want to change logo, deactivate currently active logo, and activate the logo you want to print. If multiple logos are enabled, the top most logo “wins”.

```
>>> Logging in ! (password set to 'a')
      --> CMD:C;a#
      RES:10#
>>> Disable logo 2
      --> OBJ:5;2;deactivate#
      RES:0#
>>> Enable logo 3
      --> OBJ:5;3;activate#
      RES:0#
>>> update
      --> CMD:U#
      RES:0#
```

**If you do not fully understand the above examples, after consulting this documentation, you should contact a programmer to help you communicate with the CU.**

# PROTOCOL REFERENCE

## COMMAND

### CMD:<parameters>#

Parameters	Description	Response
S	Stop print mode	RES:0#
R	Start print mode	RES:0#
F;<filename>	Load file (max 8 upper case chars). File must be present on the compact flash card.	RES:0#
D	Disconnect (PC network only)	RES:0#
U	Update next print buffer	RES:0#
C;<password>	Remote Connect with password (PC network only), not serial.	RES:10#
P	Purges all engines for 0,3 seconds	RES:0#
I;<1   2   3   4>	Ink counter reset for stall number 1..4. Only on HP version. Number must be 1,2,3 or 4.	RES:0#

**REQUEST****REQ:<parameters>#**

<b>Parameters</b>	<b>Description</b>	<b>Response</b>
<b>object list</b>	Read objects list	DAT:<object type>;<object nr>;<object name>#(all)RES:0#
<b>dir</b>	Read CU files list	DAT:<filename1># DAT:<filename2>...(all)RES:0#
<b>object data;&lt;object type&gt;;&lt;object nr&gt;</b>	Read data of specific object.	See <b>OBJECT DATA RESPONSE</b>
<b>parameters</b>	Read CU device parameters	See <b>PARAMETERS RESPONSE</b>
<b>status</b>	Read CU device status	See <b>STATUS RESPONSE</b>
<b>device name</b>	Read device name	DAT:device name;<device name>#RES:0#
<b>version</b>	Read version information	DAT:<release>;<upver.>;<fpgaver.>;<enetver.>#RES:0#
<b>wait next</b>	Return response when next print is started. Used as interrupt for next print.	RES:0#

**You can use REQ without being logged in when using Ethernet.**

All response values from CU are sent as a literal value. Boolean parameters are sent as 0/1.

**CHANGE OBJECT****OBJ:<object type>; <object nr>; <parameter>; <value>#**

Parameters	Description	Allowed Values (Response = RES:0# on success)
activate / deactivate	activate or deactivate object. Usable for logos to select different logos.	None
TEX;<text>	Set text of text objects.	Text string max 80 chars.
CUR;<value>	Set current count of counter objects.	integer value (32 bit)
TIM;<value>	Set minutes offset of time objects.	integer 16bit value (signed)
DAT;<value>	Set days offset for date objects	integer 16bit value (signed)

It is only possible to change CONTENT of objects. Other parameters (position, length, counter properties etc) are changed in design phase. If you need to change these, simply create different layouts and load files as needed.

Integer value 32 bit             $2^{32} - 1$  or 4,294,967,295 maximum

Integer 16bit value signed   -32768 to 32767



## SET PARAMETERS

**PAR:<parameter>; <value>#**

*Set parameter is used to modify the print parameters of the unit.*

Parameters	Description	Allowed Values (Response = RES:0#)
start;<value>	Set start distance.	integer value 0..65535 in 1/100 mm
edge;<+/->	Set startswitch edge.	<+> : positive edge. <-> : negative edge.
endless;<+/->	Set endless print mode.	<+> : endless print. <-> : normal print
mode;<vel/pos>	Set encoder mode.	<vel> : set velocity mode. <pos> : set position mode.
encoder;<value>	Set encoder value.	integer value 0..65535 in 1/100000 mm encoder pulses.
velocity;<value>	Set velocity value.	integer value 0..65535 in 1/100 m/min. However only 0..30000 is practically possible due to HP maximum frequency.

## FILE

**FIL:<parameters>#**

**<file>** (after receiving response to send)

Parameters	Description	Response
N;<filename>; <file extension>; <file size>	Initialize new file transmission to CF card in CU unit.	FIL:G;<filename>#
	Send file data after FIL:G response	RES:0#

Example:                                      FIL:N;myfile;CU;347#                                      prepare myfile.CU 347 bytes.

## PING (only Ethernet connections)

**PNG:<parameters>#**

Parameters	Description	Response
CU	Ping the CU, for broadcast on port 1600. All CUs on the network will reply.	<PNG:CU#000(10 bytes)><IP ADDRESS(4 bytes)><ENET VERSION(4 bytes)><0xFFFF>

PNG should be sent as a UDP broadcast on port 1600. It is used to locate CUs in a network.

The response is also a UDP packet containing

- PNG:CU#000      10 bytes
- CU IP address    4 bytes
- CU enet version   4 bytes
- End as 0xFFFF    2 bytes, always FF FF

# Parsing response from CU

## OBJECT DATA RESPONSE

*This is the response you get on REQ:object data*

Object type	Responses	parameters
All		active, name, font, prompt, locked, layer
Text	DAT:<parameters>;<value>#	text
Counter	... (RES:0# after last DAT response)	decimals, counter, direction, leadin, maxcount, mincount, countstep, repeats
Time		format, minoffset
Date		format, dayoffset, houroffset
Barcode		content, objtype, objnr

The CU sends common parameters (listed under “All”) before sending object-specific values.

Each parameter is returned in a separate DAT response, example

```
DAT:active;1#
DAT:name;Object1#
DAT:font;14#
...
RES:0#
```

## PARAMETERS RESPONSE

*This is the response you get on REQ:parameters#*

Responses	parameters
DAT:<parameters>;<value># ... (RES:0# after last DAT response)	startdelay, startedge, direction, upsidedown, otherside, spitdelay, spitlength, tickle, repeatnr, repeatdis, endless, firemode, modular, position, qudrature, velocity

## STATUS RESPONSE

*This is the response you get on REQ:status#*

Responses	parameters
DAT:<parameters>;<value># ... (RES:0# after last DAT response)	printactive, file, prints, time, fault(0 = no CU system fault), lowink (1=inklow) See CU STATUS FAULT CODES

## STATUS FAULT CODES

*On sending DAT:fault;xx#, these are the values of xx different from 0.*

Responses	parameters
1	File Error: CU file missing
2	File error: CF card full
3	File error: FNT file missing
4	File error: CU file corrupt
5	Flash error: FPGA corrupt
6	Flash error: Language corrupt
7	XJ128 files are invalid on XJ500 CU
8	XJ128 files are invalid on HP CU
9	XJ500 files are invalid on XJ128 CU
10	XJ500 files are invalid on HP CU
11	HP files are invalid on XJ128 CU
12	HP files are invalid on XJ500 CU
13	File error: HW file missing
14	Default restored Please restart
15	CF card corrupt or missing
16	CF format error. Format for FAT
18	Warning Inklow
19	Warning Maximum print speed

## RESULT CODES

*The codes following RES:nn#.*

Error number	Error Description
0	No errors
1	Wrong password
2	Unknown command
10	Password accepted (ethernet only, not error)
11	Not connected (ethernet only. To connect use CMD:C;<password>#)
21	FIL: File not found / CF card full
22	File receive timeout error (file length error or transmission error)
23	File receive error (file length error)
24	File name error (use upper letters or numbers only)
100	CMD: Unknown command
101	Printer is not running
102	Printer is running
103	File not found
106	Printer active
200	Unknown request
300	Object not found
301	OBJ: Unknown or missing parameter
320	OBJ: illegal data
1000	PAR: Unknown or missing parameter
1020	PAR: Illegal edge
1040	PAR: Illegal data
1050	PAR: Illegal mode