

Gameboy Printer Format

Information gathered by *Martin Eyre* Email: Martin.Eyre@crispgroup.co.uk special thanks to all who contributed especially *ClapOn/ClapOff*.

Last Updated : 24-August-1998

Reading data from the GB Camera

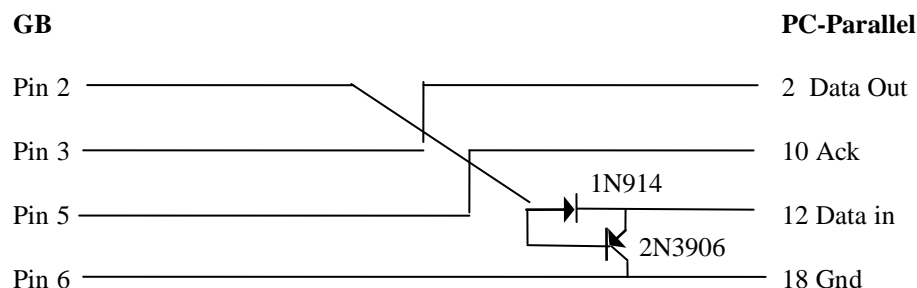
The following document contains information on how to make your PC emulate the Gameboy printer thus allowing you to download gameboy camera pictures, it also details the gameboy printer format.

The information has been gained by trial and error and even though I've successfully managed to download data I don't guarantee all the information is correct.

A Gameboy connector cable (Gameboy to Parallel port)

I constructed a cable a bit like the one used with GBBasic except I joined the **clk** to pin 11 instead of pin 2, this allows the PC to read the **CLK** using interrupts on IRQ 7.

Note : When constructing a cable I noticed the colours of wires differed to what was outlined in a diagram describing the GBBasic cable.



Reading the Data

Data is sent from the GB Camera down the serial port one bit at a time, we know when a bit has arrived because the CLK goes low causing a interrupt to fire, we then read the data from pin 12.

The GB Camera starts by sending a *Print Initialise ~ Command 1* (see below) on byte 9 the Gameboy will expect a message back in *Message Data ~ footer* (see below) in this case the message is 129.

Do I need to send any other messages back to GB Camera ?

I've successfully read all the data by only sending the 1 message as above, but I know that the GBPrinter sends back many other messages within the footer see *Message Data ~ footer* below for more information.

Software

I've written a utility to emulate the GBPrinter and store the downloaded pictures, I'm upgrading the software all the time and at the moment is only half finished.

It currently only works under DOS.

All printer commands start with \$88 \$33 then the 3rd byte is the command type.

Print Initialise ~ Command 1

This is the first data that needs to be sent to initialise the printer so it knows that data is about to arrive.

HEADER				Length 6 bytes
Byte	Hex	Bin	Dec	Description
00	88	10001000	136	Command Type 1
01	33	00110011	51	
02	01	00000001	1	
03	00	00000000	0	
04	00	00000000	0	
05	00	00000000	0	
FOOTER				Length 4 bytes
CRC = 1 + 0 + 0 + 0				
Byte	Hex	Bin	Dec	Description
00	01	00000001	1	CRC Low Byte
01	00	00000000	0	CRC High Byte
02	00	00000000	0	Message data byte 0
03	00	00000000	0	Message data byte 1

Start Printing ~ Command 2

After the first data pack has been sent send this data to tell the printer to start printing.

HEADER				Length 10 bytes
Byte	Hex	Bin	Dec	Description
00	88	10001000	136	Command Type 2
01	33	00110011	51	
02	02	00000010	2	
03	00	00000000	0	
04	04	00000100	4	
05	00	00000000	0	
06	01	00000001	1	Margins (see below)
07	13	00010011	19	
08	e4	11100100	228	
09	40	01000000	64	
FOOTER				Length 4 bytes
CRC = 2 + 0 + 4 + 0 + 1 + 19 + 228 + 64 = 318 (100111110 bin)				
Byte	Hex	Bin	Dec	Description
00	3e	00111110	62	CRC Low Byte
01	01	00000001	1	CRC High Byte
02	00	00000000	0	Message data byte 0
03	00	00000000	0	Message data byte 1

Margins (Byte 07)

There are before picture and after picture margins arranged as follows :-

Before Margin 

07	13	00010011	19	Margins
----	----	----------	----	---------

↑

————— After Margin

Data Pack ~ Command 4

This is how the data is sent in 2 rows of 20 columns

HEADER Length 6 bytes

Byte	Hex	Bin	Dec	Description
00	88	10001000	136	Command Type 4
01	33	00110011	51	
02	04	00000100	4	
03	00	00000000	0	
04	80	10000000	128	Data Length Low Byte
05	02	00000010	2	Data Length High Byte

DATA Length 640 bytes

640 bytes of data = 16 bytes tile * 20 columns* 2 Rows

Byte	Hex	Bin	Dec	Description
00	??	????????	?	640 bytes of data
...	
639	??	????????	?	

FOOTER Length 4 bytes

CRC= 4 + 0 + 128 + 2 + value of each data byte

Byte	Hex	Bin	Dec	Description
00	??	????????	?	CRC Low Byte
01	??	????????	?	CRC High Byte
02	00	00000000	0	Message data byte 0
03	00	00000000	0	Message data byte 1

Status ~ Command 15

This is sent to the printer so status can be read.

HEADER Length 6 bytes

Byte	Hex	Bin	Dec	Description
00	88	10001000	136	Command Type 15
01	33	00110011	51	
02	0F	00001111	15	
03	00	00000000	0	
04	00	00000000	0	
05	00	00000000	0	

FOOTER Length 4 bytes

CRC = 15 + 0 + 0 + 0 = 15 (00001111 bin)

Byte	Hex	Bin	Dec	Description
00	0F	00001111	15	CRC Low Byte
01	00	00000000	0	CRC High Byte
02	00	00000000	0	Message data byte 0
03	00	00000000	0	Message data byte 1

Message Data ~ footer

The printer returns messages when the gameboy is transmitting the last 2 bytes of the footer. So reading 2 bytes of data at this point will have the following format :-

First Byte will always be **one** of the following

Byte	Hex	Bin	Dec	Description
02	80	10000000	128	
02	81	10000001	129	

Second byte is the printer status code described by the bits

Bit	Bin	Description
0	00000001	unknown
1	00000010	Print in progress (when set)
2	00000100	unknown
3	00001000	unknown
4	00010000	unknown
5	00100000	Error #3 Paper Jam (when set)
6	01000000	Error #4 Too Hot/Cold
7	10000000	Error #1 Low Batteries