CASIO fx-9860 Communication Protocol Specification

Andreas Bertheussen
Manuel Naranjo

April 10, 2007

Contents

1 Introduction

This document is written to cover the protocol used for communication with the CASIO fx-9860G (SD) Graphing Calculator (codename Gy-363). This document is *not* compiled by CASIO, but is based on their original (japanese) documentation. Therefore, the information may be inaccurate or wrong due to poor translation. I apologize for my english, but I promise you that this is better than babelfish'ed japanese ;).

2 Notation and definitions

I will try to keep this document consistent;

- Hexadecimal values are represented with a 0x prefix, for example 0xFA.
- ASCII strings are enclosed in quotation marks, "05".
- In charts, an X means "unused" or "don't care", while a V means "used".

3 Interface

The calculator uses two interfaces for communication;

1. 3-pin serial interface
2. USB (1.1 standard)

3.1 Cable type

The 3-pin serial interface uses three conductors, GND, Tx and Rx. Tx in one end of the cable is connected to Rx in the other, and vice versa. The voltage levels used are +4.2V for logical 1, and 0V (GND) for logical 0. [1]

3.1.1 Signal format

<omitted>

3.2 USB Interface

The calculator can be identified on the USB bus by the following properties;

- Product ID: 0x6101 (Applies only to the fx-9860G)
- Provider ID: 0x07CF (Should apply to all Casio Products)

Even though the USB standard is well-defined, some things need to be done to set up the USB connetion with the calculator. By examining transfer logs after using FA-124, Manuel wrote the following code, utilizing libusb[2]:

3.2.1 Code

```
int init_9860(struct usb_dev_handle *usb_handle) {
        int retval;
        char* buffer;
        buffer = calloc(0x29, sizeof(char));
        retval = usb_control_msg(usb_handle, 0x80, 0x6, 0x100, 0, buffer, 0x12, 200);
        if (retval < 0) {
                fprintf(stderr, "Unable to send first message\n");
```

[1] Because the EIA-232-E standard uses signal levels at ±12V, a converter is necessary to use the calculator with EIA-232-E compliant devices such as computers.

[2] http://libusb.sourceforge.net/doc/

```
                return retval;
        }
        retval = usb_control_msg(usb_handle, 0x80, 0x6, 0x200, 0, buffer, 0x29, 250);
        if (retval < 0) {
                fprintf(stderr, "Unable to send second message\n");
                return retval;
        }
        retval = usb_control_msg(usb_handle, 0x41, 0x1, 0x0, 0, buffer, 0x0, 250);
        if (retval < 0) {
                fprintf(stderr, "Unable to send third message\n");
                return retval;
        }
        free(buffer);
        return 0;
}
```

3.2.2 Explanation

There are three control messages that need to be sent to the calculator to put it into the data transfer mode. The structure usb_handle is a structure that points to the device, it has been completed by usb_open. You should check the libUSB code for more information.

4 Protocol summary

4.1 Example of communication flow

Note that the primary equipment is usually the computer, while the calculator is the secondary equipment.

Type	Primary equipment	Direction	Secondary equipment
Start session	Connection verification	>	
		<	Response
Single operation	Command	>	
		<	Response
Send data	Command	>	
		<	Response
	Data	>	
		<	Response
Request data	Command (request)	>	
		<	Response
	Direction change	>	
		<	Command
	Response	>	
		<	Data
	Response	>	
		<	Direction change
End session	End communication	>	
		<	Response

4.2 The packet

Communication over the interface is done with *packets*. All packets transmitted are acknowledged with a *response packet*. There are several types of response packets, they will all be covered in later sections.

4.3 Packet format

Size	1 b	2 b	1 b	4 b	0–n b	2 b
Name	Type (T)	Subtype (ST)	Data follows (DF)	Data size (DS)	Data (D)	Checksum (CS)

- The T field decides the basic type of packet, command, data, response, or directional change packet.
  - 0x00–0x1F
- The ST field decides the specific meaning of the packet, – which command, what type of response etc. Using ASCII hex format.
- The DF field indicates if the packet contains a data area (DS and D) or not.
  - "0" or "1" (0x30 or 0x31)
- The DS field only exists when DF="1", and indicaties the size of the following D field using ASCII-hex formatting.
  - "0000"– "FFFF"
- The D field only exists when DF="1", and contains data with a length defined by DS.
- The CS field contains a checksum of field ST through D.
  - The contents of this field is made by calculating the sum of all bytes from the beginning of ST to the end of D, NOT that sub, and then add 1 to the result. The two least significant hex digits are converted to ASCII and used as CS.

4.4 Packet types (T)

The basic types of packets used are:

- The *command packet* information on what to do, and sometimes how, depending on the type.
- The *data packet* only makes sense following a command packet, and is used to transfer big chunks of data. In a file transfer, the command packet contains the files metadata (name etc.), while the data packet contains the file data.
- The *directional change packet* is used after requesting the secondary device to do something. It allows the other side (secondary), to send commands, on request by the primary device. When the request has been fulfilled, the secondary device returns the command right by sending back a direction change packet.
- The *response packet* is used to acknowlede received command and data packets. There are *positive response packets* and *negative response packets*, that contain information (responses) to the last received data.
- The *connection verification packet* is used to verify the secondary equipment while starting communication and during communication.
- The *end communication packet* is sent when a host wants to stop communicating, and the session is ended.

This table shows the value of T and the respective packet type.

T	Type
0x01	Command packet
0x02	Data packet
0x03	Directional change packet
0x05	Connection verification packet
0x06	Positive response packet
0x15	Negative response packet
0x18	End communication packet

4.5 Handling packet loss

In addition to the CS field in packets, the protocol defines how to handle packet loss, where both sides have entered a wait state (waiting for response or waiting for data). After two failed tries at connection verification, communication is ended.

4.5.1 Unavailable secondary

Primary equipment	Direction	Secondary equipment
[Packet X]	> (destroyed)	
10 second timeout, no response		
Connection verification	>	
	<	Response (retransmission request)
[Packet X] (retransmitted)	>	
	<	Response

4.5.2 Unavailable primary

Primary equipment	Direction	Secondary equipment
[Packet X]	>	
	(destroyed) <	Response
10 second timeout, no response		
Connection verification	>	
	<	Response (retransmission request)
[Packet X] (retransmitted)	>	
	<	Response

4.6 Automatic protocol recognition

<omitted>

4.7 Timeouts

<can wait>

5 The command packet

There are three groups of command packets, depending on their operation;

- System command
- MCS (RAM) command
- Flash memory command

5.1 Packet format (D)

Size	2 b	2 b	8 b	2 b	2 b	2 b	2 b	2 b	2 b	$n1$ b	$n2$ b	$n3$ b	$n4$ b	$n5$ b	$n6$ b
Name	Options (OPT)	Data Type (DT)	Filesize (FS)	SD1	SD2	SD3	SD4	SD5	SD6	D1	D2	D3	D4	D5	D6

- The OW field is used to decide if data should be overwritten or not where appropriate.
  - "0" Verify before overwriting.
  - "1" Do not overwrite.
  - "2" Overwrite anyway.
- The DT field is sometimes filled with "MCS Management Code".
- The FS field contains filesizes or is used to report the amount of free space.
- SD1–SD6 contain the sizes of field D1–D6, in ASCII-decimal. "00" indicates that the corresponding data field does not exist.

  - “00”–“99”

- D1–D6 contain the data/parameters for the command.

5.2 System commands

The following tables show how system command packets are stuffed.

ST	Description	Communication flow
“00”	Reset or restart	Command, response
“01”	Device identification	Command, response (private)
“02”	Comm. settings change[3]	Command, response
“03”	Enable protocol security	Command, response
“04”	Disable protocol security	Command, response
“05”	Cancel data regulation	Command, response
“06”	Set timeout	Command, response

ST	OW	DT	FS	D1	D2	D3
“00”	X	X	X			
“01”	X	X	X			
“02”	X	X	X	Baud Rate (ASCII)	Parity: “ODD” “EVEN” “NONE”	Stop bit: “1” “2”
“03”	X	X	X	Security code		
“04”	X	X	X			
“05”	X	X	X	User ID code		
“06”	X	X	X	Minutes (ASCII). Max “1440”		

An X means “don’t care”, a V means “used”, empty D-fields means not used. Note that DS1–DS6 will represent the size of D1–D6, and are not specified.

5.3 MCS (RAM) commands

These commands operate on the built in file system (1.5MB).

ST	Description	Communication flow
“20”	Create folder	Command, response
“21”	Delete folder	Command, response
“22”	Rename folder	Command, response
“23”	Change folder	Command, response
“24”	Request file	Command, response, directional change
“25”	Send file	Command, response, data
“26”	Delete file	Command, response
“27”	Rename file	Command, response
“28”	Copy file	Command, response
“29”	Request all files	Command, response, directional change
“2A”	Initialization	Command, response
“2B”	Request capacity	Command, response, directional change
“2C”	Send capacity	Command, response
“2D”	Request all fileinfo	Command, response, directional change
“2E”	Send fileinfo	Command, response
“2F”	Request RAM image	Command, response, directional change
“30”	Send RAM image	Command, response, data
“31”	Request setting info	Command, response, directional change
“32”	Send setting info	Command, response
“33”	Request all setttings	Command, response, directional change

ST	OW	DT	FS	D1	D2	D3	D4	D5	D6
“20”	X	X	X	New name					
“21”	X	X	X	Folder to delete					
“22”	X	X	X	Orig. folder name	New folder name				
“23”	X	X	X	Folder name					
“24”	X	V	X	Folder name	File to send	Group			
“25”	V	V	V	Folder name	File to send	Group			
“26”	X	V	X	Folder name	File to delete	Group			
“27”	X	V	X	Folder name	Orig. file name	New file name			
“28”	X	V	X	Source folder	Source file	Dest. folder	Dest. file		
“29”	X	X	X						
“2A”	X	X	X						
“2B”	X	X	X						
“2C”	X	X	V						
“2D”	X	X	X						
“2E”	X	V	V	Folder name	File name	Group			
“2F”	X	X	X						
“30”	X	X	X						
“31”	X	X	X	Setting name					
“32”	X	X	X	Setting name	Setting data				
“33”	X	X	X						

5.4 Flash memory commands

<elegantly skipped, for now>

6 The data packet

Data packets are always sent in relation to commands, never alone. The largest data size per packet is limited to 256 bytes. The ST of data packets is the same as for the command packet it follows.

6.1 Packet format (D)

Size	4 b	4 b	*n* b (max 256 b)
Name	Total number	Packet number	Transmitted data

Both the total number of data packets and packet number contain hexadecimal values in ASCII format.