Casio CFX-9950G communications protocol

Erik Grindheim, August ~ October 2001

This document describes the communication process for transferring individual values to or from the graphic Casio calculators. Everything in this document is based on experiments carried out with the CFX-9950G. Still the information should be valid for all the compatible units that go into the so-called CFX-9850G series of *Power Graphic* calculators from Casio.

The communication is initiated by the Casio when the **Send(** or **Receive(** command is run. This transfers an individual value to or from one of the “Alpha Memory” variables. These commands can also transfer *Matrix data*, *List data* and *Picture data*, but none of these subjects are covered in this document.

Electrical:

TTL-level (High=5V and Low=0V), asynchronous serial half-duplex communication, connects to a PC´s RS-232 COM-port with a MAX232 or equivalent level-converter circuit.

The Casio-plug is a standard 2.5mm stereo jack with the following pin-designations:

Sleeve	Ground
Ring	Data to Casio
Tip	Data from Casio

Baud-rate:	9600 bps	
Parity:	none	
Byte-size:	8 bits	
Stop-bits:	from Casio:	2 bits
	to Casio:	1 bit

Communication process to SEND variables:

This section describes the data blocks transferred when using the **Send(** function to send a variable from the Casio to the outer world.

When the command is issued the Casio sends an “attention request”, one byte: 0x15. Then the external device must reply with a “device present”, one byte: 0x13. If the Casio doesn’t receive this byte (0x13) within 0.5 ~ 1 second there will be a timeout and the Casio displays the **Com ERROR** message.

When the first handshaking-procedure (0x15 and 0x13) is over the Casio starts to send the *Variable-description packet*, which consists of 50 bytes.

The external device receives this and confirms with one byte: 0x06.

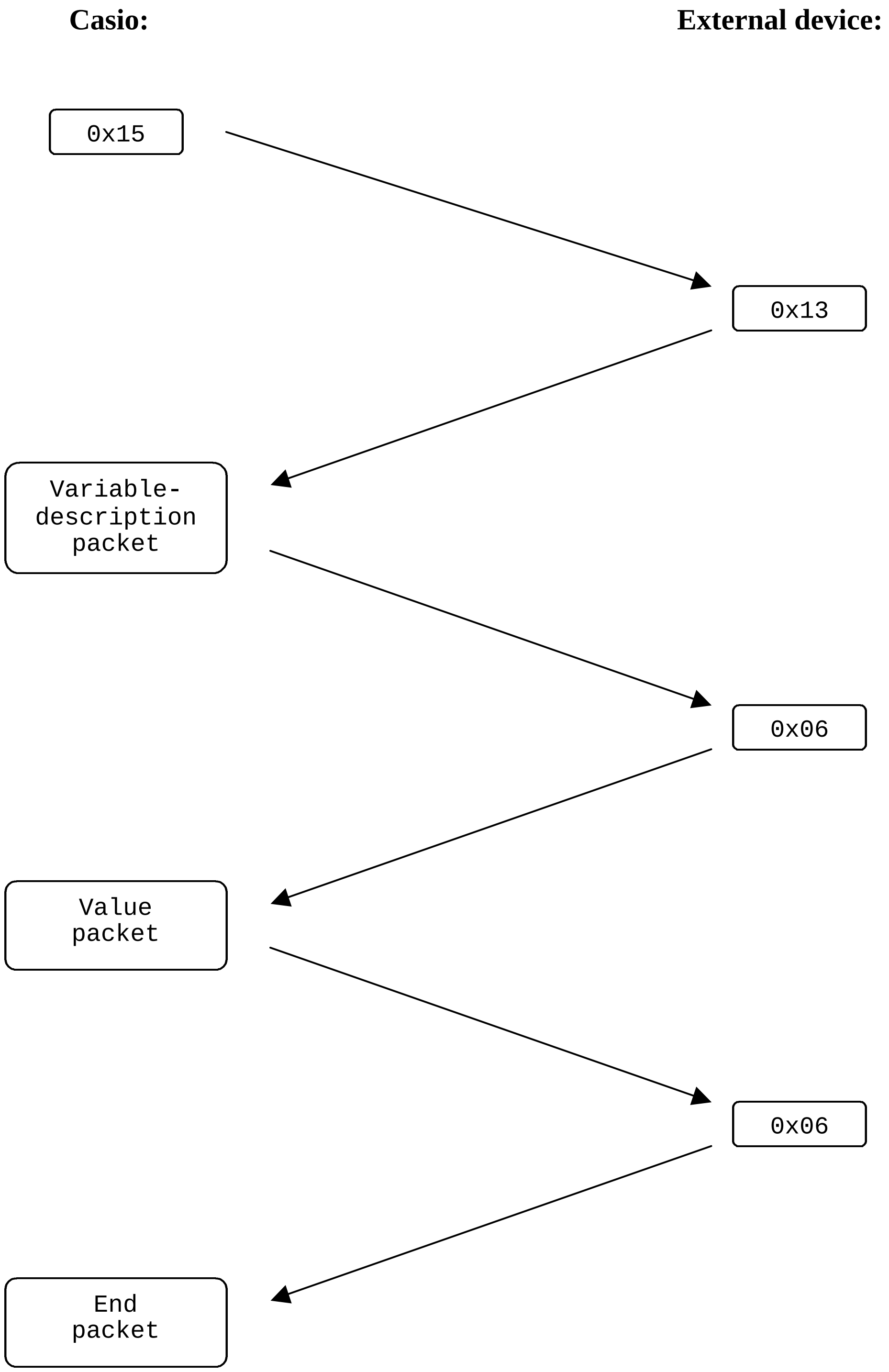
Then the Casio sends its *Value packet*, consisting of either 16 or 26 bytes. Packet length depends on whether the variable also contains an imaginary part or not. If the variable has never been assigned a value since the last reset of “Alpha Memory” this packet will not be sent at all.

The external device also receives this packet and confirms with 0x06.
(If an empty variable is sent, then there is no *Value packet*, and this byte is not sent either.)

To close the communication cycle the Casio sends an *End packet* of 50 bytes. The end packet is always the same; none of the bytes change their value.

The diagram at the next page clarifies how the process is carried out:

Instruction: *Send(*



Communication process to RECEIVE variables:

This section describes the data blocks transferred when using the **Receive(** function to read a variable from the outer world into the Casio.

When the command is issued the Casio sends an “attention request”, one byte: 0x15. Then the external device must reply with a “device present”, one byte: 0x13. If the Casio doesn’t receive this byte (0x13) within 0.5 ~ 1 second there will be a timeout and the Casio displays the **Com ERROR** message.

When the first handshaking-procedure (0x15 and 0x13) is over the Casio starts to send the *Request packet*, which consist of 50 bytes.

The external device receives this and confirms with one byte: 0x06.

The Casio confirms (that it’s ready for a *Variable-description packet*) with one byte: 0x06.

Then the external device sends a *Variable-description packet* (consisting of 50 bytes). Byte 12 in this packet (variable name) seems to be totally ignored by the Casio, as the variable is stored under the name indicated in the *Request packet* anyway. Still it’s recommended to send back the same name/byte as received in the *Request packet* byte 12.

Again the Casio confirms that it’s ready (for a *Value packet* or an *End packet*, depending on byte 9 and 11 in the previous packet) with one byte: 0x06.

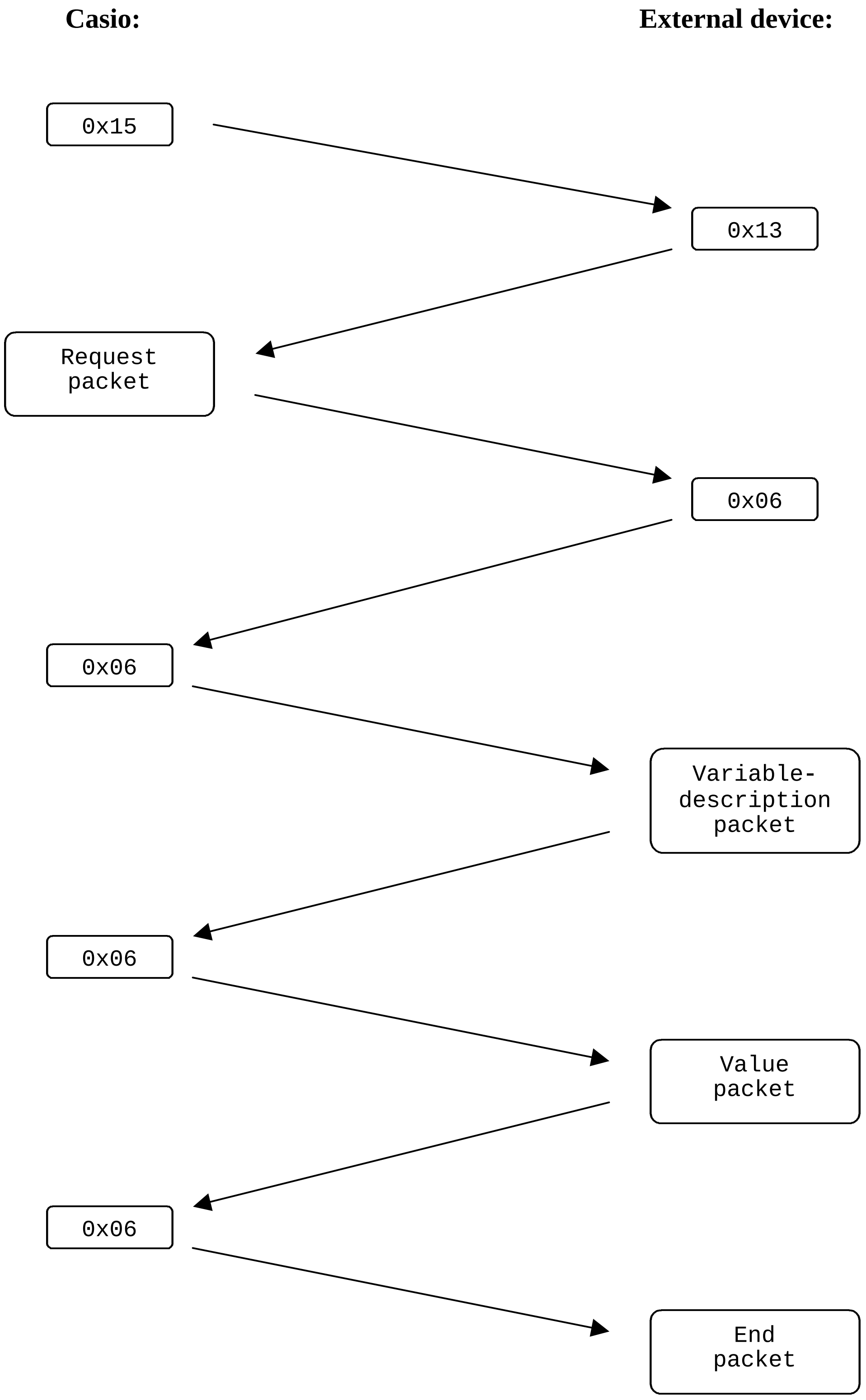
A *Value packet* is then sent from the external device. It is still consisting of either 16 or 26 bytes, depending on whether the variable also contains an imaginary part or not. If the variable is empty this packet is not sent at all (length: 0 bytes). This means that the actual “Alpha Memory” variable in the Casio is deleted. This will free 10 (real) or 20 (complex) bytes of memory in the Casio. In calculations the deleted variable’s value equals zero (0).

If a *Value packet* was sent then the Casio confirms (as usual) with a 0x06.
(If the variable is empty there is no *Value packet*, and this byte is not sent either.)

To close the communication cycle the external device sends an *End packet* of 50 bytes. The end packet is always the same; none of the bytes change their value.

The diagram at the next page clarifies how the process is carried out:

Instruction: *Receive(*



The different data packets:

There are four different types of data packets in use to transfer variables, in addition to the single-byte packets. This section describes the structure of these packets in detail:

Request packet:

Byte no (1-50)	ASCII	HEX
1-4	:REQ	3A 52 45 51
5		00
6-7	VM (Variable) PC (Picture) LT (List) MT (Matrix)	56 4D <-- This is the bytes we want... 50 43 (Byte no 12-49 below is based) 4C 54 (on Variable transfers, VM.) 4D 54 ( )
8-11		FF FF FF FF
12	A-Z / r / è	(The name of the Variable:) 41-5A / CD / CE
13-49		all FF
50		Checksum = 01+not((sum bytes 1-49)-3A)

Variable-description packet:

Byte no (1-50)	ASCII	HEX
1-4	:VAL	3A 56 41 4C
5		00
6-7	VM	56 4D
8		00
9		if the variable is reset/unused: 00 ** if variable is in use (normal): 01
10		00
11		the same value as byte no 9: 00 or 01
12	A-Z / r / è	(The name of the Variable:) 41-5A / CD / CE
13-19		FF FF FF FF FF FF FF
20-27	Variable	56 61 72 69 61 62 6C 65
28	C or R	43 or 52 (If the variable has an imaginary part: C-complex. Otherwise: R-real)
29		0A
30-49		all FF
50		Checksum = 01+not((sum bytes 1-49)-3A)
		**note: If bytes 9 and 11 has the HEX-value 00 this means that the variable has not been used after last "Alpha Memory" reset. Then no Value packet will be transmitted at all. After this packet comes the End packet.

Value packet, real numbers:

Byte no (1-16)	ASCII	HEX
1	:	3A
2-5		00 01 00 01
6		00-09 (BCD 0 and BCD 0-9 (integer part))
7-13		00-99 (BCD for 14 digits (decimal part))
14		SignInfoByte. Bit set if...: bit 7 - Variable has an imaginary part bit 6&4 - Value (real part) is negative bit 0 - Absolute value of (real part of) value is 1,0... or more bits 1, 2, 3 and 5 are always 0 / Low
15		Exponent, 00-99 Exp.: Byte: +99 99 \\ +01 01 -} byte 14 bit 0 = 1 (High) ** 00 00 / -01 99 \\ -02 98 -} byte 14 bit 0 = 0 (Low) -99 01 /
16		Checksum = 01+not((sum bytes 1-15)-3A)
		**note: If the variable is (exactly) 0,0 then the exponent is 00 (instead of minus infinite) but this bit (bit 0 in byte 14) is still low because variable value is less than 1.

The smallest value the Casio accepts is $\pm\ 1.00000000000000 \times 10^{-99}$

The greatest number the Casio accepts is $\pm\ 9.99999999999999 \times 10^{+99}$

Zero is stored as $+\ 0.00000000000000 \times 10^{+00}$

Or, at a general form: $\pm\ \underline{\texttt{I}}.\underline{\texttt{decimals}} \times 10^{+\underline{\texttt{EE}}}$

In the *Value packet*, byte no. 6 is containing the integer part $\underline{\texttt{I}}$, always from 0x01 to 0x09 (except if the variable value is exactly 0,0…). Bytes no. 7 to 13 contain the 14 $\underline{\texttt{decimals}}$ represented as BCD data. The exponent $\underline{\texttt{EE}}$ is represented in byte no. 15 and bit 0 of the SignInfoByte (byte no. 14). The sign indicating whether the value is negative or positive is held in bits 6 and 4 of the SignInfoByte.

Value packet, numbers with an imaginary part:

Byte no (1-26)	ASCII	HEX
1-15		Equal to bytes 1-15 in the 16 bytes long Value packet for real numbers. These bytes only describe the real part of the variable, while the next 10 bytes describe the imaginary part of the complex number:
16		00-09 (BCD 0 and BCD 0-9 (integer part))
17-23		00-99 (BCD for 14 digits (decimal part))
24		SignInfoByte. Bit set if...: bit 7 - Always set to 1 (imaginary) bit 6&4 - Value (img. part) is negative bit 0 - Absolute value of (img. part of) value is 1,0... or more bits 1, 2, 3 and 5 are always 0 / Low
25		Exponent for the imaginary part; 00-99 Exp.: Byte: +99 99 \\ +01 01 -} byte 24 bit 0 = 1 (High) 00 00 / -01 99 \\ -02 98 -} byte 24 bit 0 = 0 (Low) -99 01 /
26		Checksum = 01+not((sum bytes 1-25)-3A)

End packet:

Byte no (1-50)	ASCII	HEX
1-4	:END	3A 45 4E 44
5-49		all FF
50	V	56 (checksum, always the same value)

Checksum calculation:

As an example I’ll show how the checksum of the *End packet* is calculated. This value will always remain the same since nothing in this packet changes from time to time; everything is constant. First we add all the previous 49 bytes. This gives us a sum of 0xE4 (the excess carry digits are thrown away). Then we subtract 0x3A and the result is 0xAA. We invert it (=0x55) and add 0x01. The result is 0x56, which is the correct checksum-byte; byte no. 50.

Miscellaneous:

The only way for an external device to initiate communication with the Casio is if the calculator is already in LINK-mode and F2 is pressed (F2: Receive). Then the external device must start the communication by sending a 0x16. This is not covered further in this document.

If the Casio receives an incorrect packet (something is wrong or un-logical; for instance a value packet of 26 bytes, complex number, is transferred when the Casio expects a real value with packet length of 16 bytes) then the Casio immediately transmits a 0x22 to inform that the communication is aborted. It also displays the **Com ERROR** message. Whenever an error occurs in the communication process the Casio shows this message and transmits a 0x22 byte.

It is possible to release memory by sending empty/un-used variables (no value packet, and bytes 9 & 11 with value 0x00) into the Casio. An un-used variable takes 0 byte, a real value takes 10 bytes and a complex value takes 20 bytes of memory.

Byte 12 in the *Variable-description packet* is totally ignored by the Casio, as the transferred value is stored in the variable first indicated by byte 12 in the *Request packet* anyway. It is not possible to change any other variables than the one indicated by the Casio in the *Request packet*.

This document:

Revision:	1.2
Date:	07 October 2001
Author:	Erik Grindheim

You can contact me by e-mail if you have any suggestions, comments or additional information regarding the Casio calculators' communications protocol that you want to share with me and the rest of the world.

Erik Grindheim
erik@grindheim.net