



SGD-A

Serial Graphical and text LCD

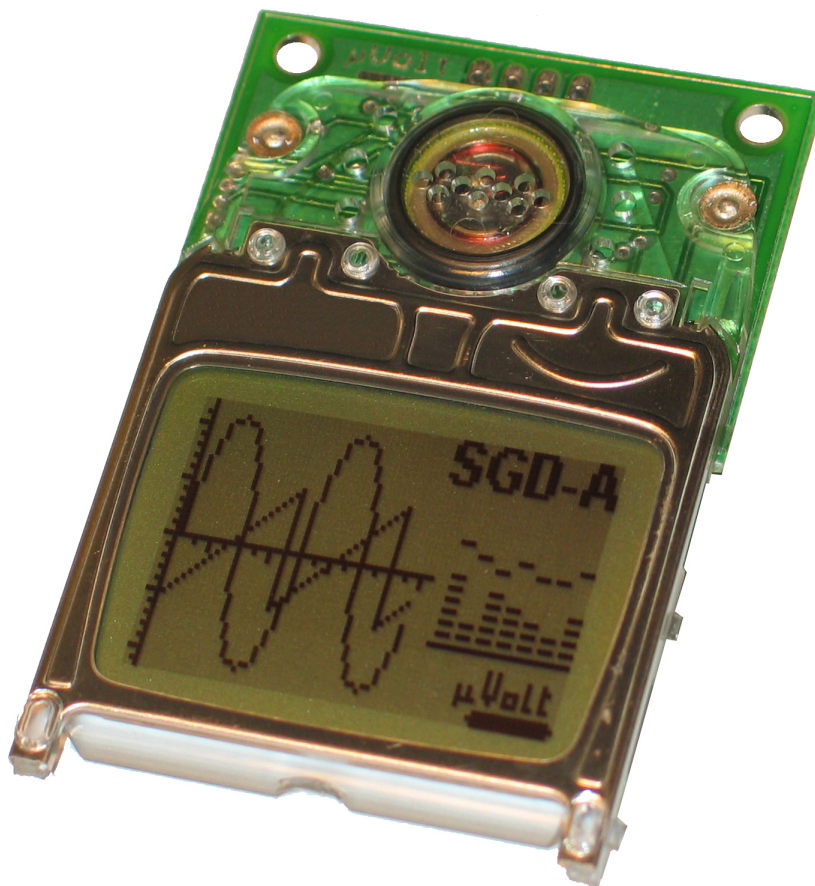
Index

<u>1.- INTRODUCTION</u>	<u>3</u>
<u>2.- PHISICAL DIMENSIONS</u>	<u>5</u>
<u>3.- ELECTRICAL SPECIFICATION</u>	<u>6</u>
<u>4.- SGD-A CONTROL</u>	<u>8</u>
B: BITMAP	10
C: CONTRAST	11
D: READ-RAM	12
E: WRITE-RAM	13
L: LINE	14
N: CLS	15
P: LOCATE	16
R: READ-EEPROM	17
W: WRITE-EEPROM	18
S: STRING	19
T: TONE	21
F: FIRMWARE RELEASE	22
X: PING	23
Z: SLEEP	24

1.- Introduction.

SGD-A is a serial, graphic and text LCD with following features:

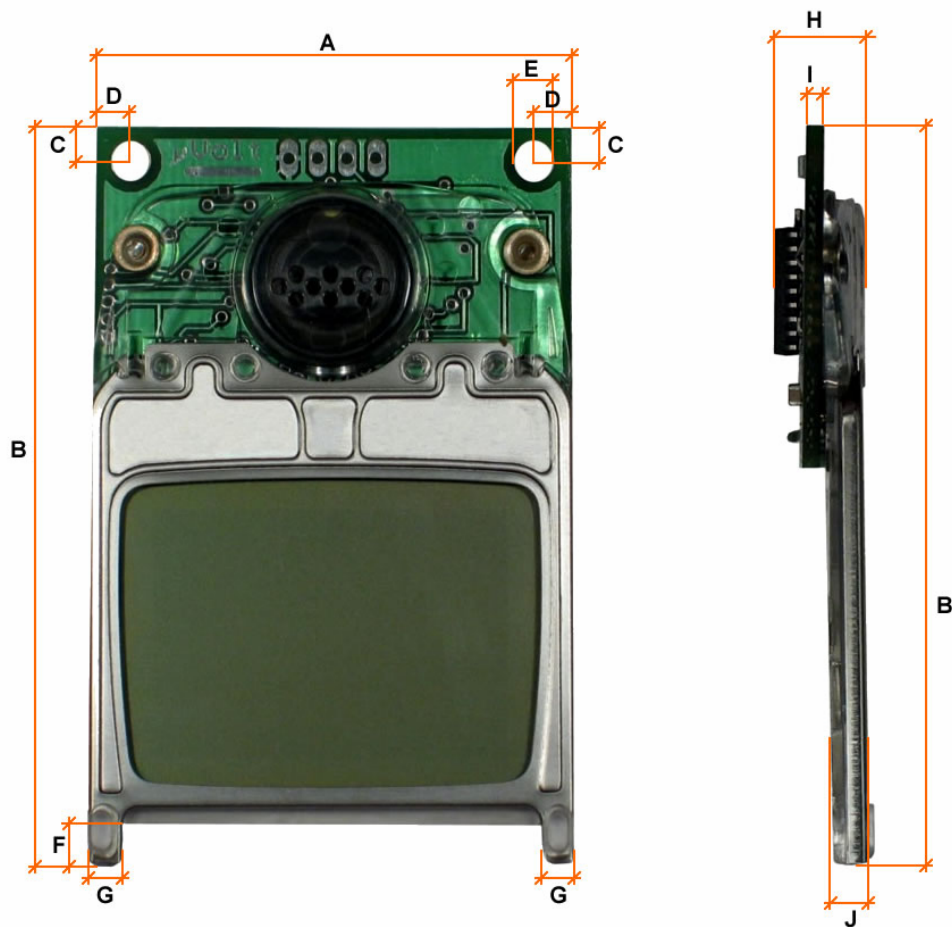
- 84x48 pixels (text: 6 lines of 14 characters)
- 4 pins (Vcc,Tx,Rx,GND). Only 2 pins of microcontroller needed for LCD
- High level commands (line, bitmaps, strings)
- Serial communication 9600 bps, 8bits, no parity
- predefined ascii characters (94 characters)
- Vcc between 3v and 5v +-5%
- 128 bytes user E²PROM
- 64 bytes user SRAM
- Speaker with tone generation by simple commands
- Power consumption of 0,9mA @ 3v (1,2mA @ 5V)



Control is simple. All commands are identified by a byte plus the arguments needed for execution. You can see all commands in the table below. Syntax of every command is not referred here. For detailed information of syntax, see chapter "SGD-A Control"

Command	Name	Description
'B'	Bitmap	Sends to display the bitmap that is being received in Rx pin
'C'	Contrast	Adjusts LCD contrast. SGD-A is temperature compensated, so this command normally is unnecessary
'D'	Read-Ram	Reads a byte of user RAM , that is 64 bytes.
'E'	Write-Ram	Writes a byte of user RAM, that is 64 bytes.
'L'	Line	Draws a line in the display. It starts from last coordinates of pointer (Look at command 'P') and ends in selected coordinates.
'N'	CLS	Clears screen and places the pointer at (0,0) (top left corner)
'P'	Locate	Places the pointer in the selected coordinates. It is suitable to place text or lines.
'R'	Read-eprom	Read a byte of user EEPROM, that is 128 bytes.
'W'	Write-eprom	Writes a byte of user EEPROM, that is 128 bytes.
'S'	String	Writes a string of text in current coordinates.
'T'	Tone	Outputs a tone of frequency and time selected.
'F'	Firmware Revision	Displays a FW version of SGD-A.
'X'	Ping	Outputs a 'X' in Tx. Useful to debug a connection and operation of display.
'Z'	Sleep	Start low power mode.

2.- Phisical dimensions.



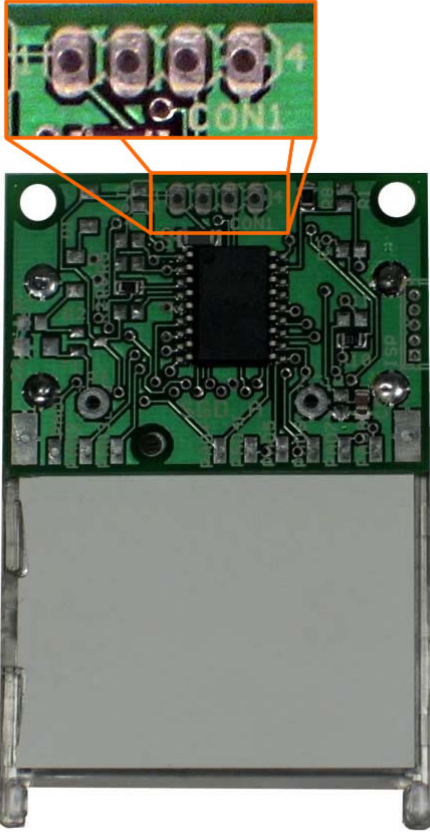
Dim*	mm	Inch
A	42	1.65
B	64	2.5
C	3	0.12
D	3.3	0.13
E	3.1	0.13
F	2.5	0.1
G	3	0.12
H	8.2	0.32
I	1.6	0.062
J	3.5	0,14
* These dimensions can be slightly different due to tolerances in manufacturing process.		

Display is 84*48 pixels, (6 lines of 14 characters), with an active area of 30*22mm.

3.- Electrical specification.

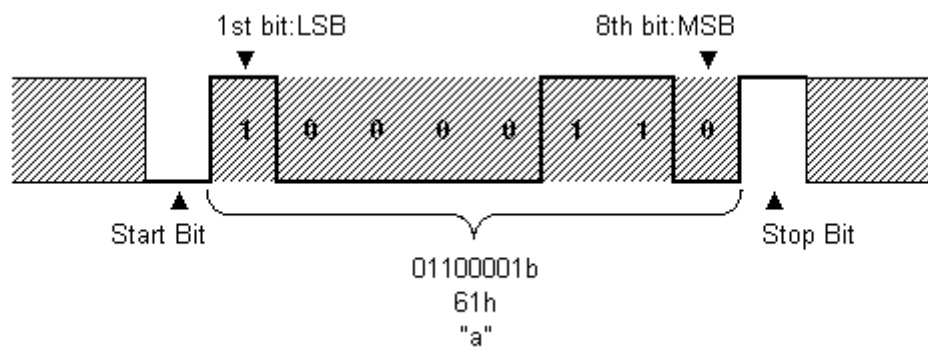
Value	Name	Min	Typ	Max	Unit
Vcc	Power voltage	3		5	V
Icc	Standby current	0.9		1.2	mA
Temperature	Temperature	0		40	°C
Humidity	Humidity	10		80	%HR

The display needs only 4 pins to operates: Vcc ,Tx ,Rx , GND.

	Pin	Name	Description
	1	Vcc	Input. DC power voltage required from 3v to 5v. It is necessary to place an external 1uF condenser near this pin.
	2	Rx	Input pin. Commands are received in this pin. Commands are sent byte after byte, with following serial format: 9600 bps, 8bits, no parity, one stop bit.
	3	Tx	Output pin. The lcd answers by this pin. If you send a command that has answer (read RAM, read EEPROM or "ping") display outputs a serial transmission: 9600 bps, 8bits, no parity, one bit stop.
	4	GND	Ground pin.

The transmission format to send or receive commands is serial (as RS-232). For most microcontrollers and high level programming languages there are libraries available for this kind of bus. Anyway if you need to implement a new one, protocol is simple and is described below.

Here you can see how to send a byte.



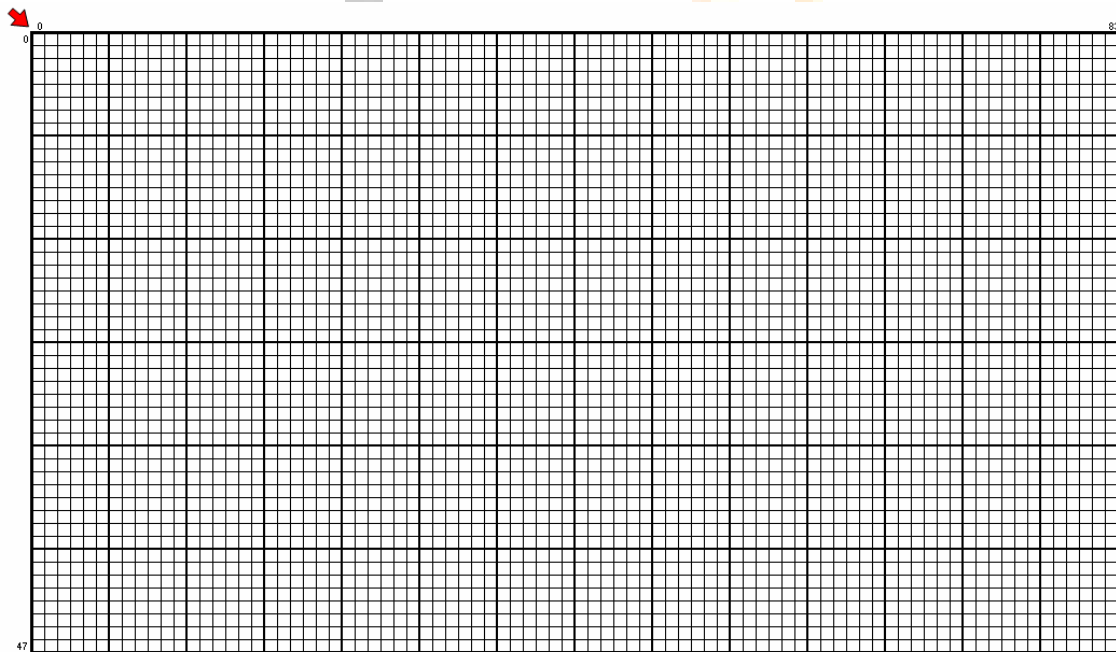
Line is at high level or '1' (Vcc), and data stream begins with Start bit. It always is a '0'. After this you must send 8 bits of data, LSB first. In the example above we are sending an 'a'. Data stream ends with the Stop bit, that always is a '1'.

At 9600 Bps the time of bit is around 104 μ s.

4.- SGD-A control.

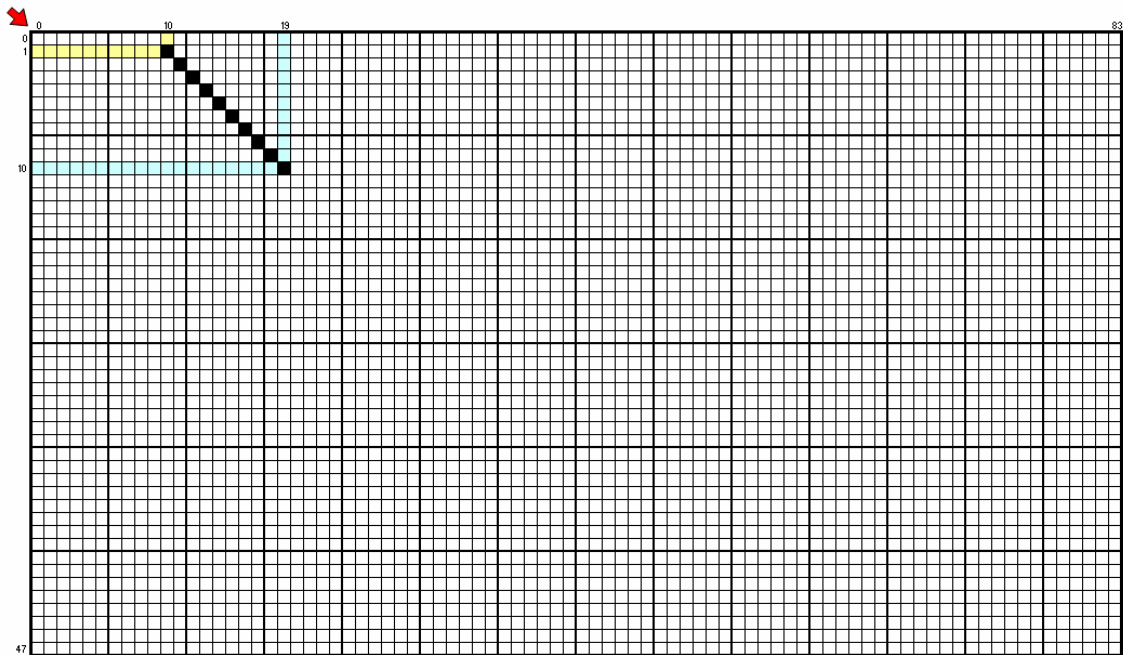
SGD-A is controlled by 12 simple commands. In this chapter each one will be explained, but before it is important to know some general information about the display.

SGD-A is graphical and alphanumeric. It is 84 x 48 pixels and has 6 lines of 14 characters each. You can send text and drawing instructions at same time, it does not exist text mode or graphical mode.



The idea behind screen control is that there is a pointer (coordinates x and y) from where next command is executed, if it affects the screen appearance. This pointer is not visible, only a register that contains a screen position. If you send a command that changes screen appearance (i.e. a line) it starts from the coordinates of pointer, and, at the end, the pointer will be set to the coordinates where the command ends. After a clear screen (or switch on display) pointer is placed at (0,0). In the picture above, it is the pixel on top left corner, pointed with a red arrow. From this point, coordinate x is growing to 83 following right direction, and coordinate y go down until 47.

Lets see an example:



Suppose we just switch on the display. Next we send a command that changes the coordinates of the pointer to the new coordinates (10,1). At this moment display pointer is exactly in the point indicated by yellow lines. Next we send the command to draw lines ('L') with coordinates (19,10) as argument. Then display draws the black line of the example above, begins in old coordinates of pointer and ends in the new ones. After this, the pointer is updated to new coordinates marked by blue lines in the example above.

Sequence will be as follows.

```
P,10,1  
L,19,10
```

Next you will find a detailed explanation of each command. You can find the syntax of each command at top and follows the next rules:

CAPITAL LETTER:	ascii character associated to instruction code. It means send 1 byte.
"," (comma):	Spacer between arguments, it does not mean send anything.
Small letter:	Numerical argument needed for the command. It means send 1 byte.

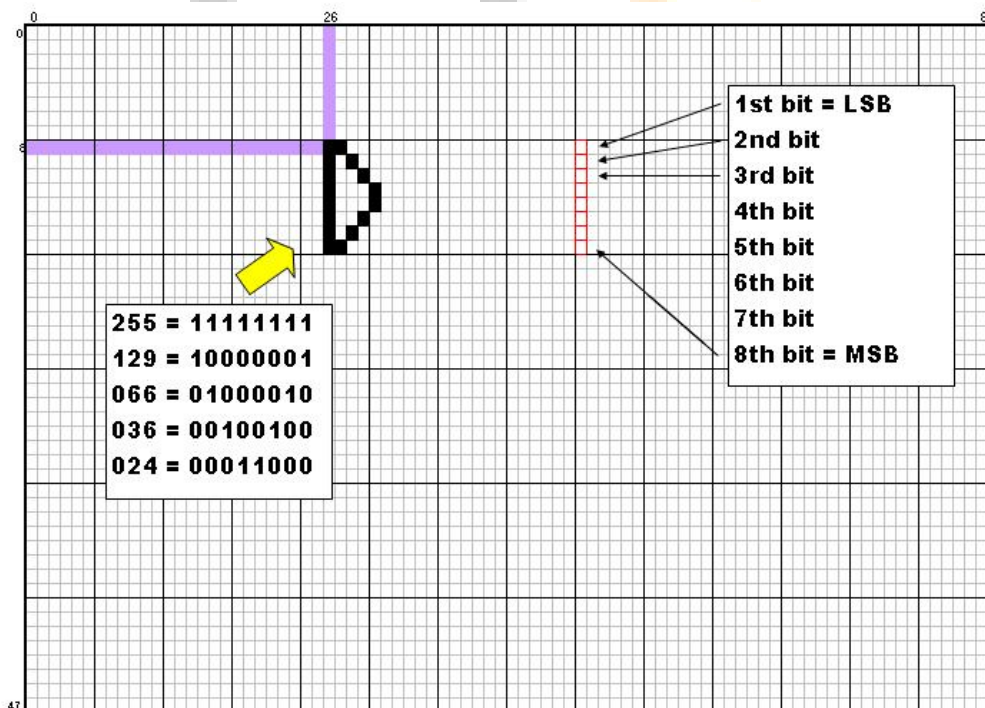
B: Bitmap

Syntax: **B,n,b₁,b₂,...,b_n**

B : Command. 'B' = 66d = 42h
n : Number of bytes of bitmap. Max. 255
b₁ : First byte of bitmap.
b₂ : Second byte of bitmap.
b_n : last (nth) byte of bitmap.

Description:

This function sends a data stream to display that will be translated to pixels on and off. n is the number of bytes to be sent. Each byte modifies an 8 pixel column, as you can see in the picture below.



For example, suppose that the pointer is in coordinates (26,8). Then we send command:

B,5,255,129,66,36,24

Result would be the one in the picture above, indicated by yellow arrow.

C: Contrast

Syntax: **C,n**

C : Command. 'C' = 67d = 43h
n : Value to adjust the contrast. Between 0 and 127.

Description:

This function, in some SGD-A, allows changing screen contrast, to set darker or clearer. Once you send a new value contrast changes instantaneously. It is useful only if display changes a lot its power voltage.

In most SGD-A this command produces no change, because most units are compensated. These models have a self adjust contrast that compensates voltage and temperature changes.

D: Read-Ram

Syntax: **D,n**

D : Command. 'D' = 68d = 44h
n : RAM address to be read. Between 0 and 63.

Description:

This function reads an address in a RAM user memory. This RAM memory is an "extra device" of SGD-A, and it is 64 bytes long. Memory addresses are from 0 to 63.

If this command is sent, SGD-A answers with a value stored in the indicated address.

E: Write-Ram

Syntax: **E,p,v**

E : Command. 'E' = 69d = 45h
p : RAM address to be written. Between 0 and 63.
v : Value to store in memory. Between 0 and 255.

Description:

This function writes an address in a RAM user memory. This RAM memory is an "extra device" of SGD-A, and it is 64 bytes long. Memory addresses are from 0 to 63.

If this command is sent, SGD-A stores a value in the indicated address. This memory is RAM so this value is stored as long as power is on.

L: Line

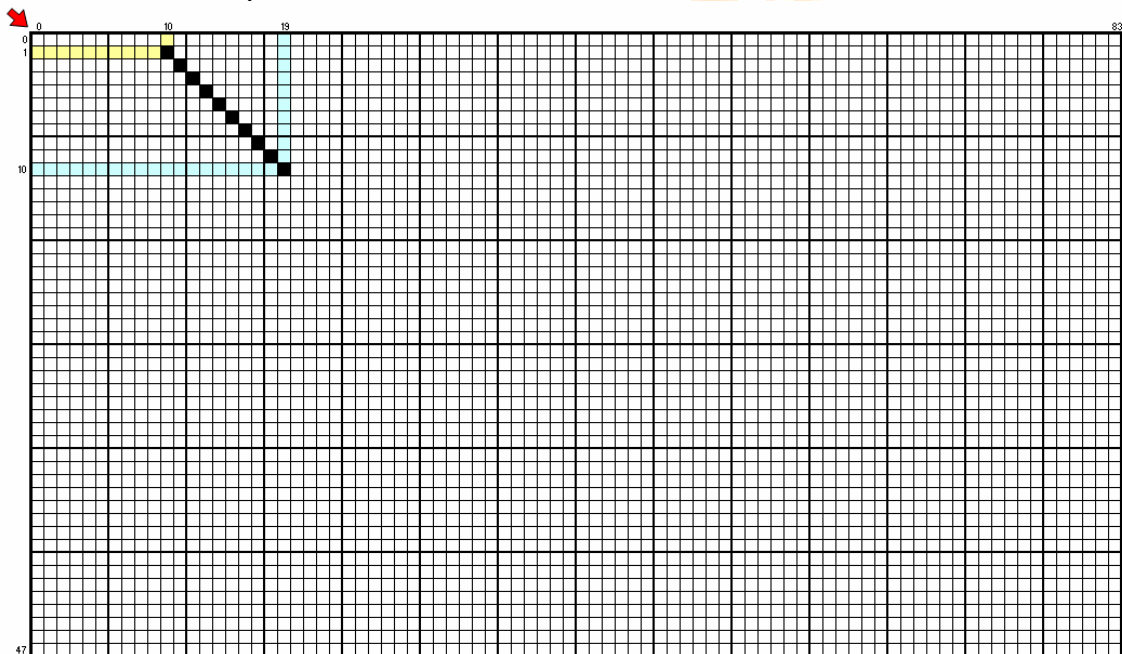
Syntax: **L,x,y**

L : Command. 'L' = 76d = 4Ch
x : Coordinate x of line end. Between 0 and 83.
y : Coordinate y of line end. Between 0 and 47.

Description:

This function draws a line from current pointer coordinates to the indicated coordinates. Afterwards pointer is updated to new coordinates, where line ends.

For example:



If send:

P,10,1
L,19,10

Result would be the line in the picture above.

This command needs aprox. 40 ms to execute.

N: CLS

Syntax: **N**

N : Command. 'N' = 78d = 4Eh

Description:

This function clears the screen and leaves the pointer at (0,0).

This command needs aprox. 61 ms to execute.

P: Locate

Syntax: **P,x,y**

L : Command. 'P' = 80d = 50h
x : New coordinate x of pointer. Between 0 and 83.
y : New coordinate y of pointer. Between 0 and 47.

Description:

This function changes the current coordinates of pointer. It can be used as an origin for writing functions (Line, Bitmap and String). All of them change pointer coordinates, leaving it at the end of "drawing".

R: Read-eprom

Syntax: **R,n**

R : Command. 'R' = 82d = 52h
n : EEPROM address to read. Between 0 and 127.

Description:

This function reads an address of EEPROM memory. This EEPROM memory is an "extra device" of SGD-A, and it is 128 bytes long. Memory addresses are from 0 to 127.

If this command is sent, SGD-A answers with a value stored in the indicated memory address.

This command needs aprox. 1.5 ms to execute.

W: Write-eprom

Syntax: **W,p,v**

W : Command. 'W' = 87d = 57h
p : EEPROM address to write. Between 0 and 127.
v : Value to store in memory. Between 0 and 255.

Description:

This function writes an address in a EEPROM user memory. EEPROM memory is an "extra device" of SGD-A, and it is 128 bytes long. Memory addresses are from 0 to 127.

If this command is sent, SGD-A stores a value in the indicated address. This memory is EEPROM, that means it is a non-volatile memory.

This kind of memory can be written about 100000 times during device's life.

This command needs aprox. 6.5 ms to execute.

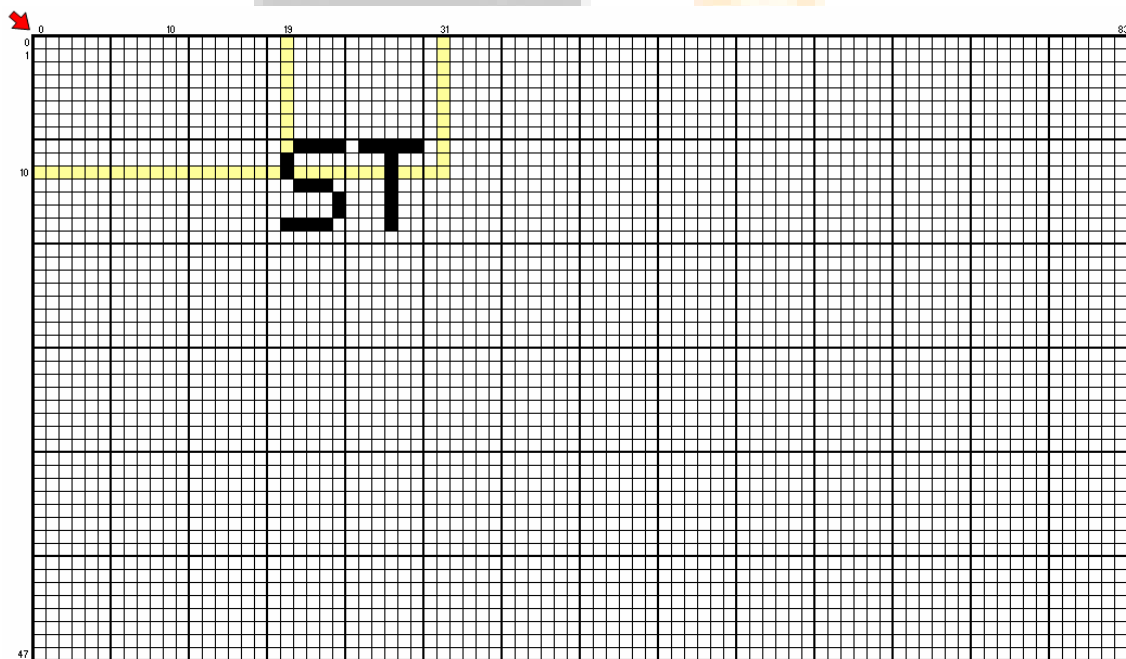
S: String

Syntax: **S,n,b₁,b₂,...,b_n**

S : Command. 'S' = 83d = 53h
n : Number of characters of String. Max. 255
b₁ : First character of string.
b₂ : Second character of string.
B_n : Last (nth) character of string.

Description:

This function writes a string of characters in the current pointer coordinates. Coordinate x is completely respected, but coordinate y is "rounded" to text line where pointer is.



For example, we suppose current coordinates are (19,10). Then we send:

S,2,S,T

Result would be the one in the picture above and new pointer coordinates will be (31,10).

SGD-A uses a very complete characters map.
ASCII codes available are:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0											10 0000 1010 LF ≡			13 0000 1101 CR ≤		
1																
2	32 0010 0000	33 0010 0001	34 0010 0010	35 0010 0011	36 0010 0100	37 0010 0101	38 0010 0110	39 0010 0111	40 0010 1000	41 0010 1001	42 0010 1010	43 0010 1011	44 0010 1100	45 0010 1101	46 0010 1110	47 0010 1111
3	48 0011 0000	49 0011 0001	50 0011 0010	51 0011 0011	52 0011 0100	53 0011 0101	54 0011 0110	55 0011 0111	56 0011 1000	57 0011 1001	58 0011 1010	59 0011 1011	60 0011 1100	61 0011 1101	62 0011 1110	63 0011 1111
4	64 0100 0000	65 0100 0001	66 0100 0010	67 0100 0011	68 0100 0100	69 0100 0101	70 0100 0110	71 0100 0111	72 0100 1000	73 0100 1001	74 0100 1010	75 0100 1011	76 0100 1100	77 0100 1101	78 0100 1110	79 0100 1111
5	80 0101 0000	81 0101 0001	82 0101 0010	83 0101 0011	84 0101 0100	85 0101 0101	86 0101 0110	87 0101 0111	88 0101 1000	89 0101 1001	90 0101 1010	91 0101 1011	92 0101 1100	93 0101 1101	94 0101 1110	95 0101 1111
6	96 0110 0000	97 0110 0001	98 0110 0010	99 0110 0011	100 0110 0100	101 0110 0101	102 0110 0110	103 0110 0111	104 0110 1000	105 0110 1001	106 0110 1010	107 0110 1011	108 0110 1100	109 0110 1101	110 0110 1110	111 0110 1111
7	112 0111 0000	113 0111 0001	114 0111 0010	115 0111 0011	116 0111 0100	117 0111 0101	118 0111 0110	119 0111 0111	120 0111 1000	121 0111 1001	122 0111 1010					
	p	q	r	s	t	u	v	w	x	y	z					

SGD-A also uses characters 10 and 13 to jump to beginning of next line. That means that if you send this character, coordinate y of the pointer will increase in 8, and coordinate x will become 0, pointing to the start of next line.

Apart from these characters, SGD-A also answers to 4 more ASCII codes:

Code	Symbol
230	μ
164	ñ
165	Ñ
241	ñ

T: Tone

Syntax: **T,p,e,t**

T : Command. 'T' = 84d = 54h
p : Timer counter period. Between 0 and 255.
e : Prescaler of timer counter. Between 1 and 16.
t : Tone duration. Between 0 and 255.

Description:

This function generates a tone in SGD-A speaker of frequency and duration chosen. This speaker is an "extra device" of SGD-A, and its control is simple.

Tone duration is:

$$Duration = 8.1 * t \quad [ms]$$

Frequency of tone follows the equation:

$$Frequency = \frac{1}{2 * (p + 1) * e * 10^{-6}} \quad [Hz]$$

For example, command T,12,249,2 generates a tone of 1 kHz that last aprox. 97 ms.

In the MicroVolt Engineering web page (www.uvolt.com) it is available an excel file to get easily from a given frequency, the values of **p** and **e**.

This command needs aprox. the tone duration time to execute.

F: Firmware Release

Syntax: **F**

F : Command. 'F' = 70d = 46h

Description:

This function outputs to the screen the firmware release of SGD-A. It is useful to debug the connection of SGD-A, and its operation.

X: Ping

Syntax: **X**

X : Command. 'X' = 88d = 58h

Description:

 This function outputs, in SGD-A transmission pin, another 'X' character. It is useful to debug the connection of SGD-A.

Z: Sleep

Syntax: **Z**

Z : Command. 'Z' = 90d = 5Ah

Description:

This function starts the low power mode. Internal oscillator is switched off, and almost all operation, but screen appearance. SGD-A wakes up with next command received.

ATTENTION: SGD-A could not execute properly the first command after a SLEEP. It is recommended to wake it up with "X" (Ping) command, and repeat it until you get the answer. After this first command, operation is normal and reliable.