KickStart™ KS480 LCD Controller

480 segment,
KickStart™ protocol,
intelligent LCD front panel controller and driver,
with SPI and I²C interfaces

GPEG International Ltd

Revision History

Date	Rev#	Description
2012-07-04	0.1	Internal release
2012-07-31	1.0	First public release

Contents

1. Introduction

KickStart™ is a segmented LCD design suite that speeds up development process of a custom LCD, simplifies its use and reduces costs and time to market. Its main components are

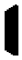
1. KickStart™ on-line tool
2. KickStart™ development kit
3. KickStart™ LCD Driver chip

These will be explained in the following chapters in details, but first a few definition is needed, because these will be used all over the document.

1.1. What is a segment?

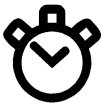
A segment is a single, usually continuous, area of the LCD surface that can be on or off. This is the smallest controllable unit on the LCD.

Example 1



This is a simple segment.

Example 2

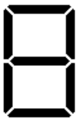


This is a more complicated segment. It consist of two areas, but these areas are always controlled together.

1.2. What is an object?

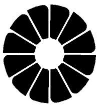
An object is a group of segments that are usually used together.

Example 1



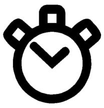
A 7-segment digit is an object consisting of seven segments which are controlled independently to show various numbers.

Example 2



A disk shape consisting of 12 segments, where the segments are controlled independently, is also an object.

Example 3

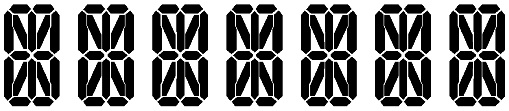


This object consist of only one segment.

1.3. What is a component?

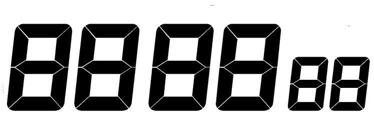
A component is an array of identical type objects that are usually used together.

Example 1



Seven identical 16-segment objects are a components.

Example 2



But six 7-segment objects are also a component even if the size of the digits are different.

2. KickStart™ on-line tool

KickStart™ on-line tool is an easy to use web based LCD drawing tool. With simple drag-and-drop operation it is possible to design a segmented LCD within seconds. After the LCD layout is ready, GPEG can create samples, define the segment to memory mapping, and table definition files. A slower, but more flexible, alternative is to make a normal mechanical drawing and submit it for processing.

At the time of writing the tool was here:

http://sidekick.web1.sf.tigauk.net/home/create

I am not sure this is its final location.

The design process is quite straightforward At first the tool asks for the size of the LCD, then creates a blank design screen. Initial size is not really important, because the project name and size of the LCD can be changed any time

Most common objects can be selected from the Toolbox on the left side. Once selected they appear in the top left corner of the design screen. Name, size and position can be defined by typing in the properties or dragging the component with the mouse or with the arrow keys.

If an object is not available in the Toolbox then it can be uploaded. Small size, high resolution, black and white PNG files are the easiest to create, but JPG and SVG files can also be used.

Objects that are part of a component shall be placed consecutively

The on-line tool can be used in two way:

1. Either as an engineering tool, by specifying the exact size and position of each elements.

   This often require the generation and uploading of custom graphic file, because no two LCDs are exactly the same.

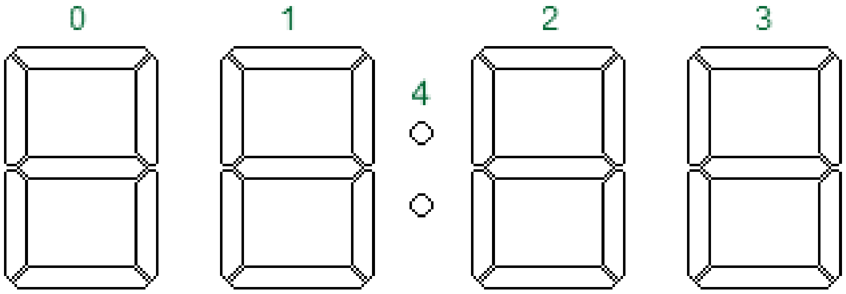
2. Or as a sketching tool to get an overall idea how the glass will look like.

   There is a huge library of objects and components available to choose from. Even if not exactly the right size and shape, they can be used to create an animated sketch quickly.

The following examples show the use of the tool better than a lengthy explanation.

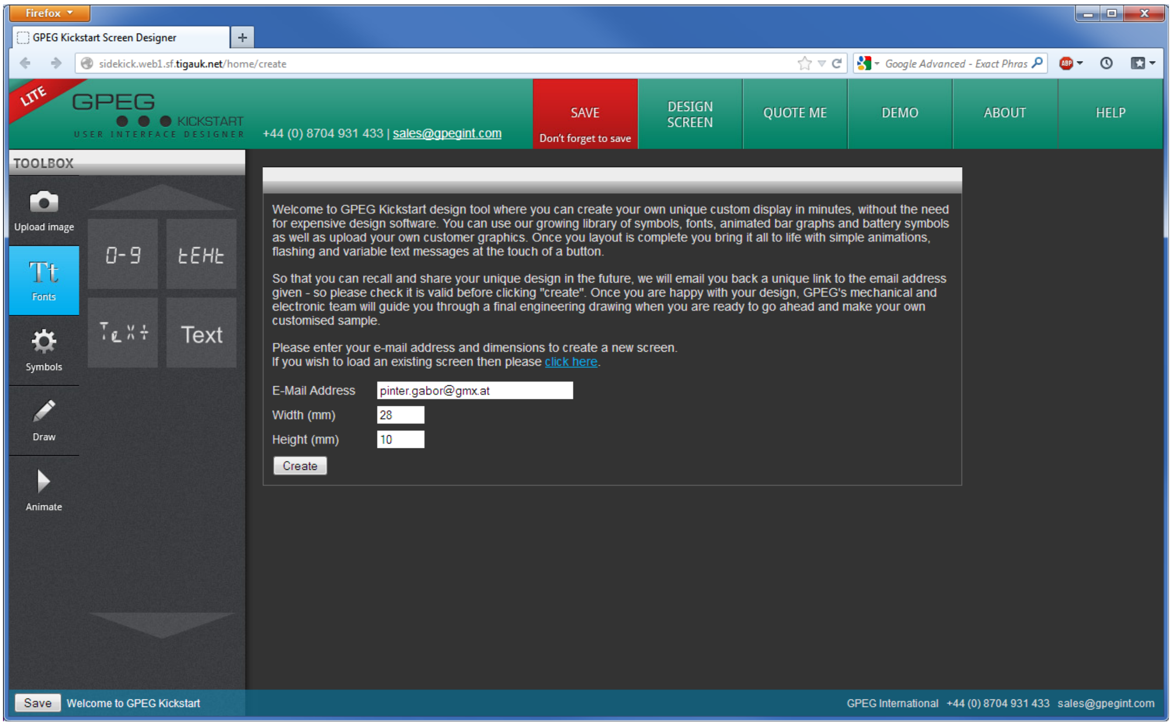
Example 1: A simple clock

The target is to design a simple clock that displays time in hh:mm format.



It can be designed the following way:

1. Open the KickStart™ on-line tool.
2. Enter your email address.
3. Define the size of the LCD as 28x10 mm and click "Create".


Firefox
GPEG Kickstart Screen Designer
sidekick.web1.sf.tigauk.net/home/create
Google Advanced - Exact Phras
LITE
GPEG
KICKSTART
USER INTERFACE DESIGNER
+44 (0) 8704 931 433 | sales@gpegint.com
SAVE
Don't forget to save
DESIGN SCREEN
QUOTE ME
DEMO
ABOUT
HELP
TOOLBOX
Upload image
Fonts
Symbols
Draw
Animate
0-9
Text
Text
Save
Welcome to GPEG Kickstart
GPEG International +44 (0) 8704 931 433 sales@gpegint.com

Welcome to GPEG Kickstart design tool where you can create your own unique custom display in minutes, without the need for expensive design software. You can use our growing library of symbols, fonts, animated bar graphs and battery symbols as well as upload your own customer graphics. Once you layout is complete you bring it all to life with simple animations, flashing and variable text messages at the touch of a button.

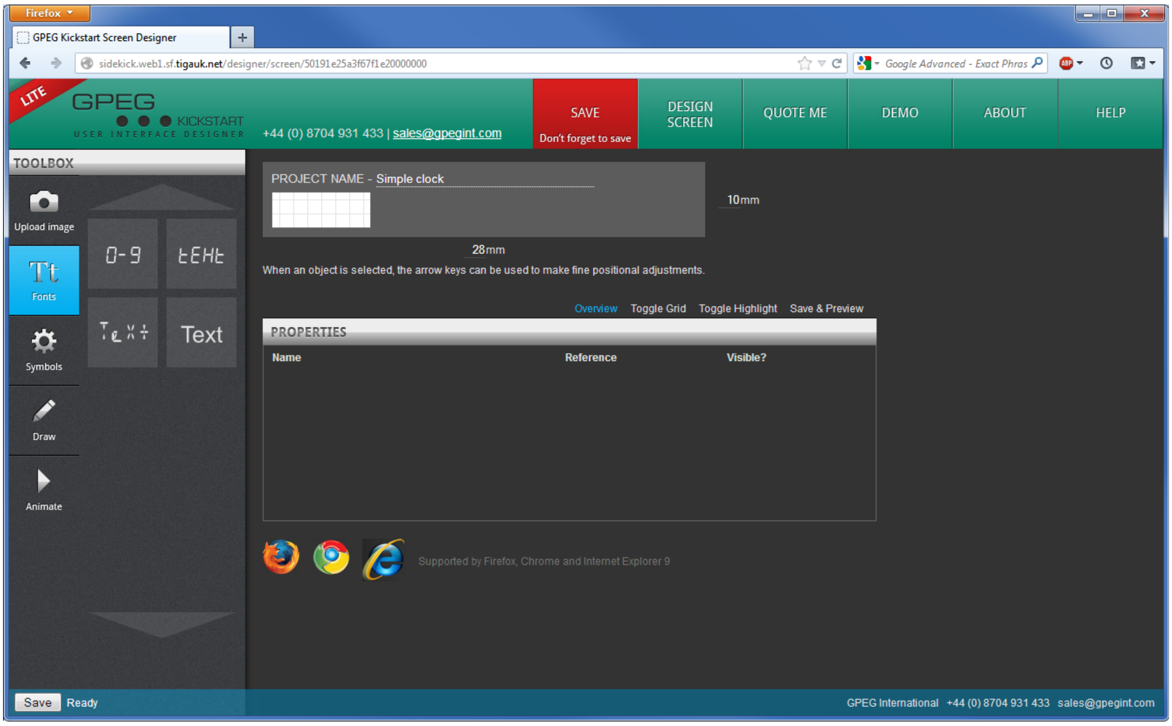
So that you can recall and share your unique design in the future, we will email you back a unique link to the email address given - so please check it is valid before clicking "create". Once you are happy with your design, GPEG's mechanical and electronic team will guide you through a final engineering drawing when you are ready to go ahead and make your own customised sample.

Please enter your e-mail address and dimensions to create a new screen.
If you wish to load an existing screen then please click here.

E-Mail Address: pinter.gabor@gmx.at
Width (mm): 28
Height (mm): 10

Create

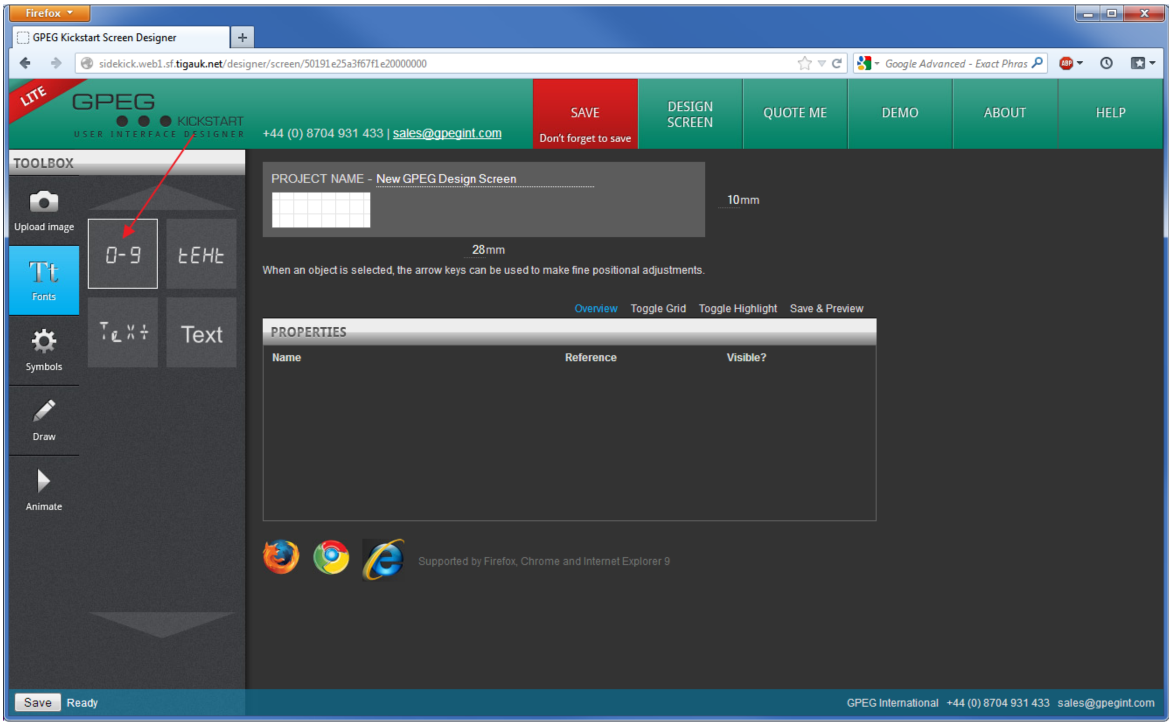
4. Click on PROJECT NAME and type "Simple clock".


GPEG Kickstart Screen Designer
sidekick.web1.sf.tigauk.net/designer/screen/50191e25a3f67f1e20000000
LITE
GPEG KICKSTART
USER INTERFACE DESIGNER
+44 (0) 8704 931 433 | sales@gpegint.com
SAVE
Don't forget to save
DESIGN SCREEN
QUOTE ME
DEMO
ABOUT
HELP
TOOLBOX
Upload image
Fonts
Symbols
Draw
Animate
0-9
Text
10mm
28mm
Overview
Toggle Grid
Toggle Highlight
Save & Preview
PROPERTIES
Name
Reference
Visible?
Supported by Firefox, Chrome and Internet Explorer 9
Save
Ready
GPEG International +44 (0) 8704 931 433 sales@gpegint.com

PROJECT NAME - Simple clock

When an object is selected, the arrow keys can be used to make fine positional adjustments.

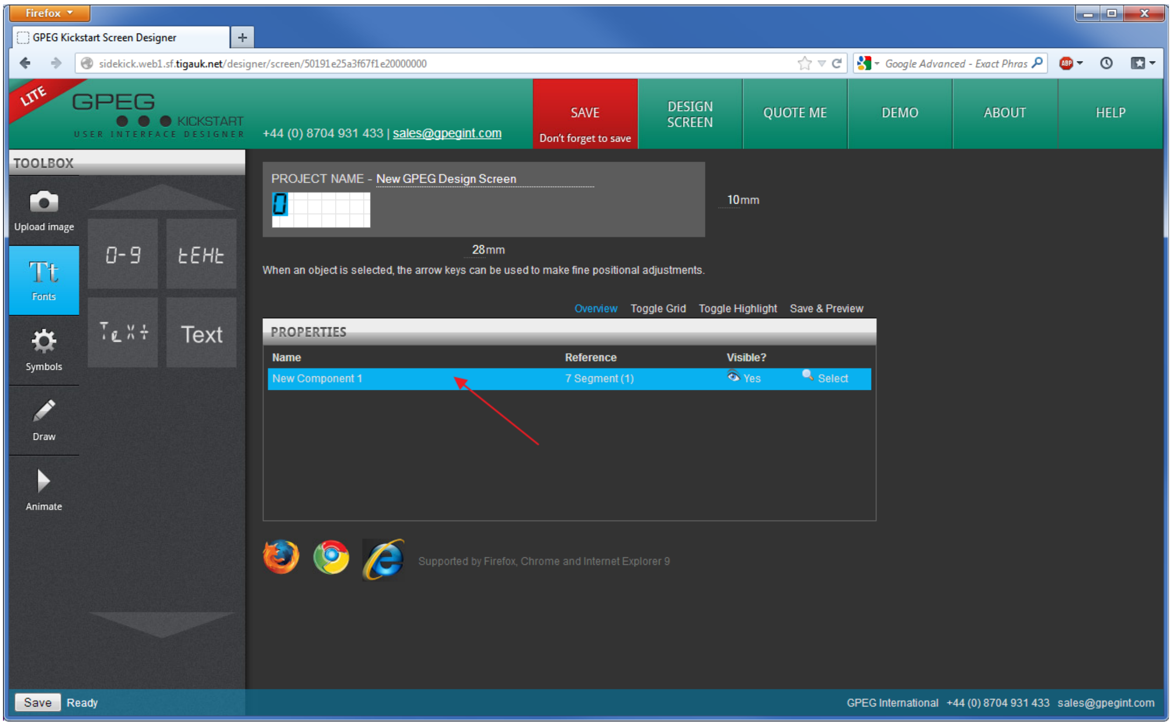
5. Select on the 7-segment display icon.


GPEG Kickstart Screen Designer
sidekick.web1.sf.tigauk.net/designer/screen/50191e25a3f67f1e20000000
LITE
GPEG KICKSTART
USER INTERFACE DESIGNER
+44 (0) 8704 931 433 | sales@gpegint.com
SAVE
Don't forget to save
DESIGN SCREEN
QUOTE ME
DEMO
ABOUT
HELP
TOOLBOX
Upload image
Fonts
Symbols
Draw
Animate
0-9
Text
10mm
28mm
Overview
Toggle Grid
Toggle Highlight
Save & Preview
PROPERTIES
Name
Reference
Visible?
Supported by Firefox, Chrome and Internet Explorer 9
Save
Ready
GPEG International +44 (0) 8704 931 433 sales@gpegint.com

PROJECT NAME - New GPEG Design Screen

When an object is selected, the arrow keys can be used to make fine positional adjustments.

6. A 7-segment object will appear in the top left corner of the design screen. Position and size are wrong, but do not worry, just click on "New Component 1" in PROPERTIES window.



Firefox
GPEG Kickstart Screen Designer
sidekick.web1.sf.tigauk.net/designer/screen/50191e25a3f67f1e20000000
Google Advanced - Exact Phras
LITE
GPEG KICKSTART
USER INTERFACE DESIGNER
+44 (0) 8704 931 433 | sales@gpegint.com
SAVE
Don't forget to save
DESIGN SCREEN
QUOTE ME
DEMO
ABOUT
HELP
TOOLBOX
Upload image
Fonts
Symbols
Draw
Animate
0-9
TEXT
Text
10mm
28mm
Overview
Toggle Grid
Toggle Highlight
Save & Preview
Supported by Firefox, Chrome and Internet Explorer 9
Save
Ready
GPEG International +44 (0) 8704 931 433 sales@gpegint.com

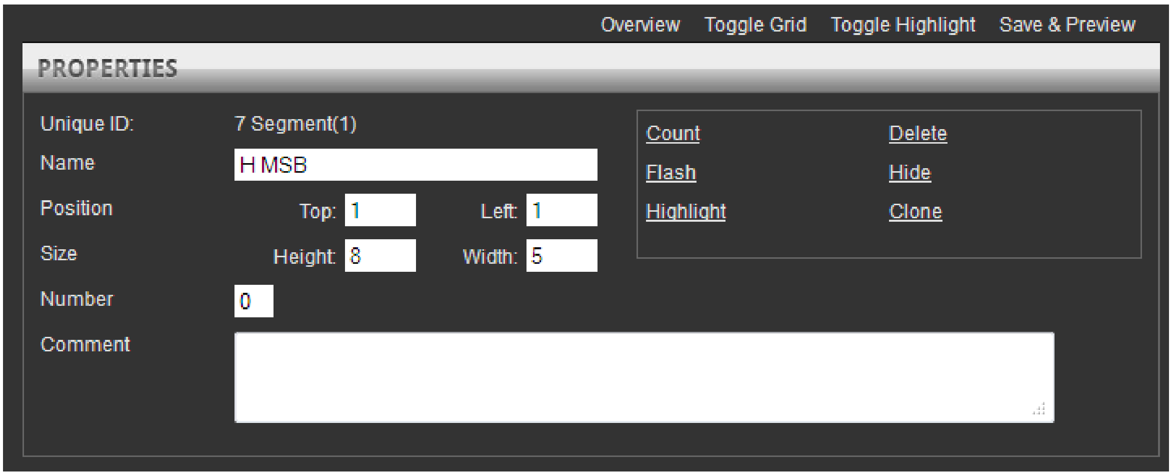
PROJECT NAME - New GPEG Design Screen

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
New Component 1	7 Segment (1)	Yes	Select

7. Enter name Name="H MSB", position Top=1, Left=1, size Height=8, Width=5.



PROPERTIES

Unique ID: 7 Segment(1)

Name: H MSB

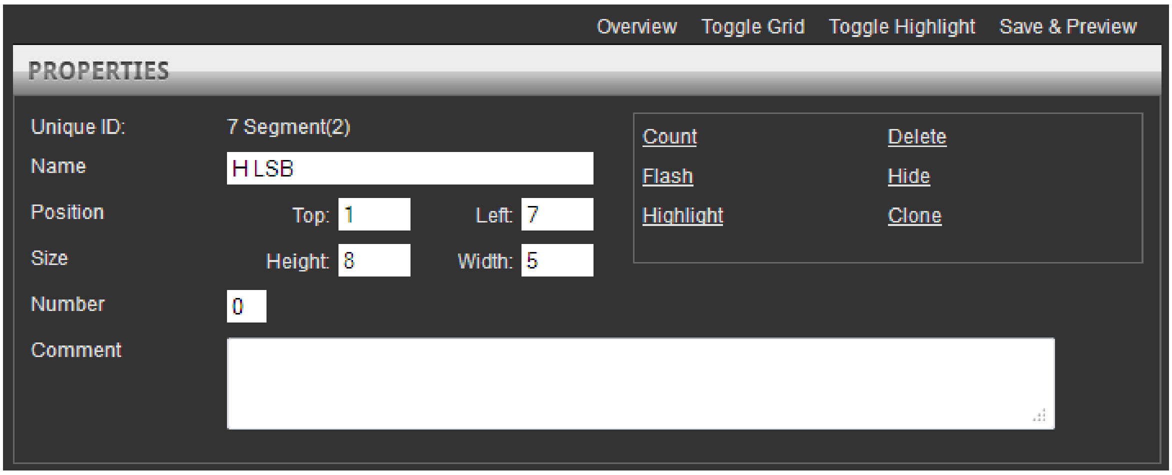
Position: Top: 1 Left: 1

Size: Height: 8 Width: 5

Number: 0

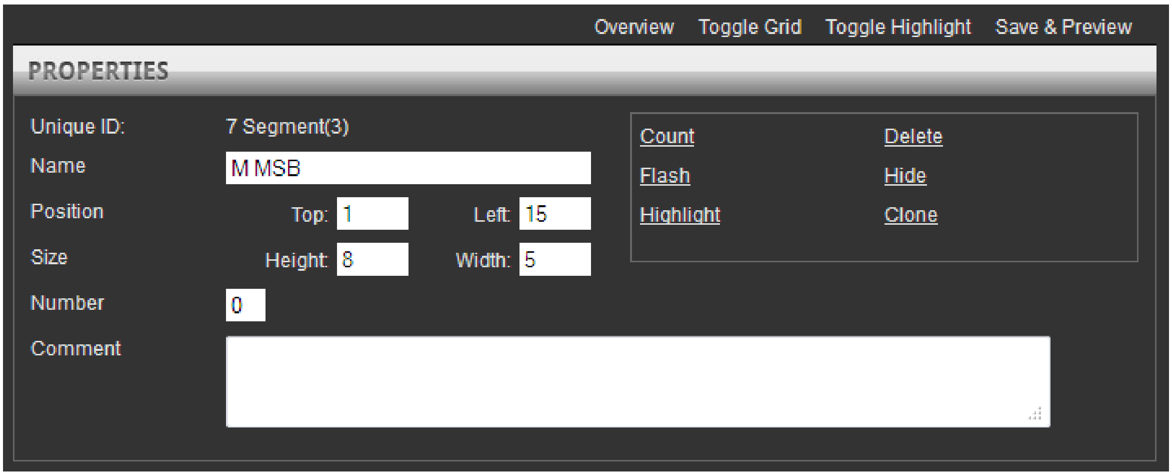
Comment:

8. Select the 7-segment display icon again, click on the "New Component 2" line in the PROPERTIES window, and enter name Name="H LSB", position Top=1, Left=7, size Height=8, Width=5.



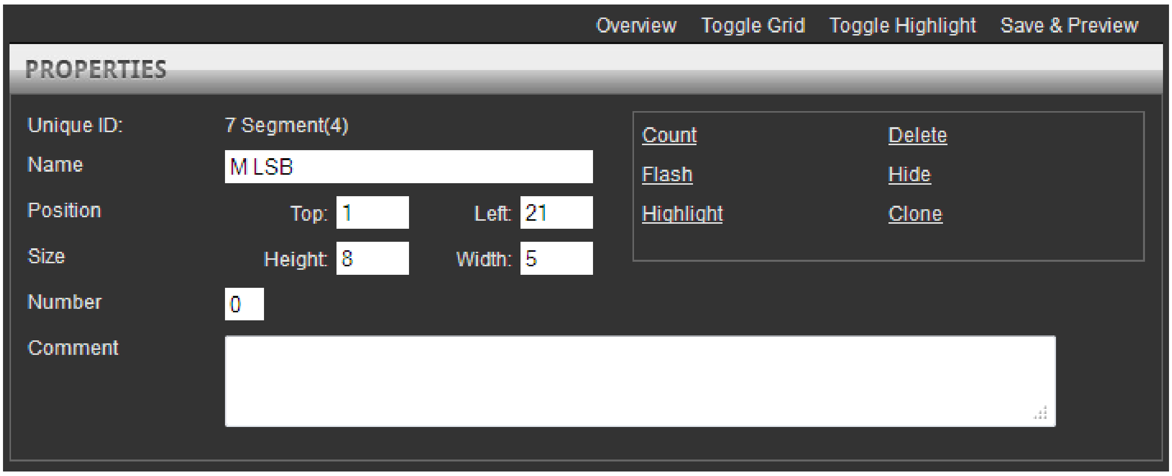
Unique ID: 7 Segment(2)
Name H LSB
Position Top: 1 Left: 7
Size Height: 8 Width: 5
Number 0
Comment

9. Select the 7-segment display icon again, click on the "New Component 3" line in the PROPERTIES window, and enter name Name="M MSB", position Top=1, Left=15, size Height=8, Width=5.



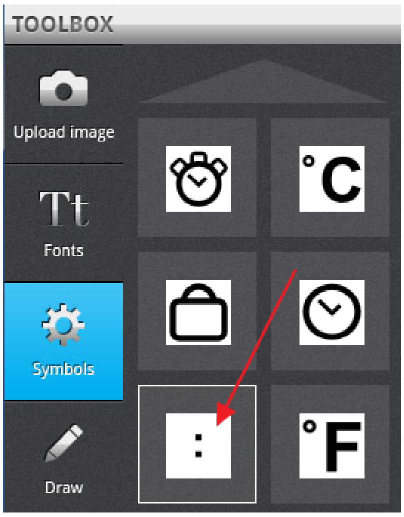
Unique ID: 7 Segment(3)
Name M MSB
Position Top: 1 Left: 15
Size Height: 8 Width: 5
Number 0
Comment

10. Select the 7-segment display icon again, click on the "New Component 4" line in the PROPERTIES window, and enter name Name="M LSB", position Top=1, Left=21, size Height=8, Width=5.

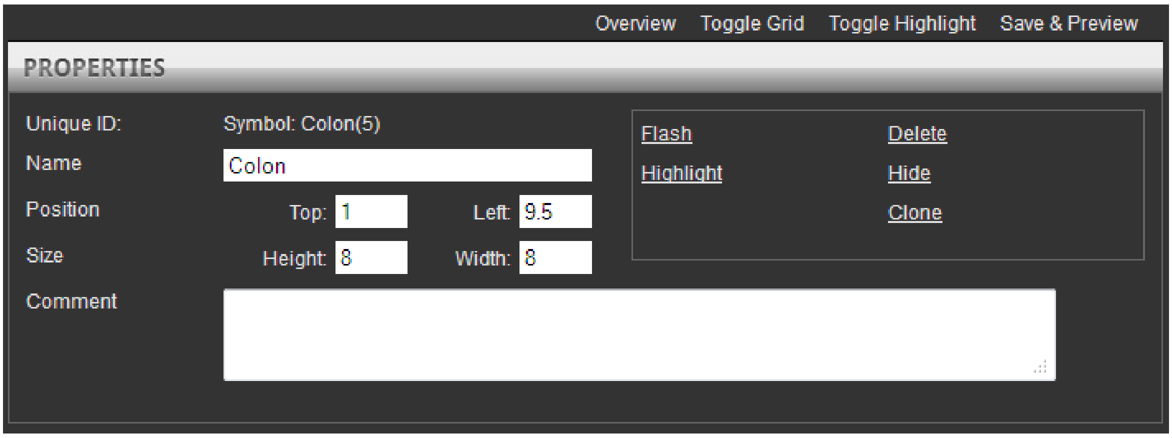


Unique ID: 7 Segment(4)
Name M LSB
Position Top: 1 Left: 21
Size Height: 8 Width: 5
Number 0
Comment

11. Select the colon symbol, from Symbols on the left side



12. Click on the "New Component 5" line in in PROPERTIES window, and enter name Name="Colon", position Top=1, Left=9.5, size Height=8, Width=8.



PROPERTIES

Unique ID: Symbol: Colon(5)

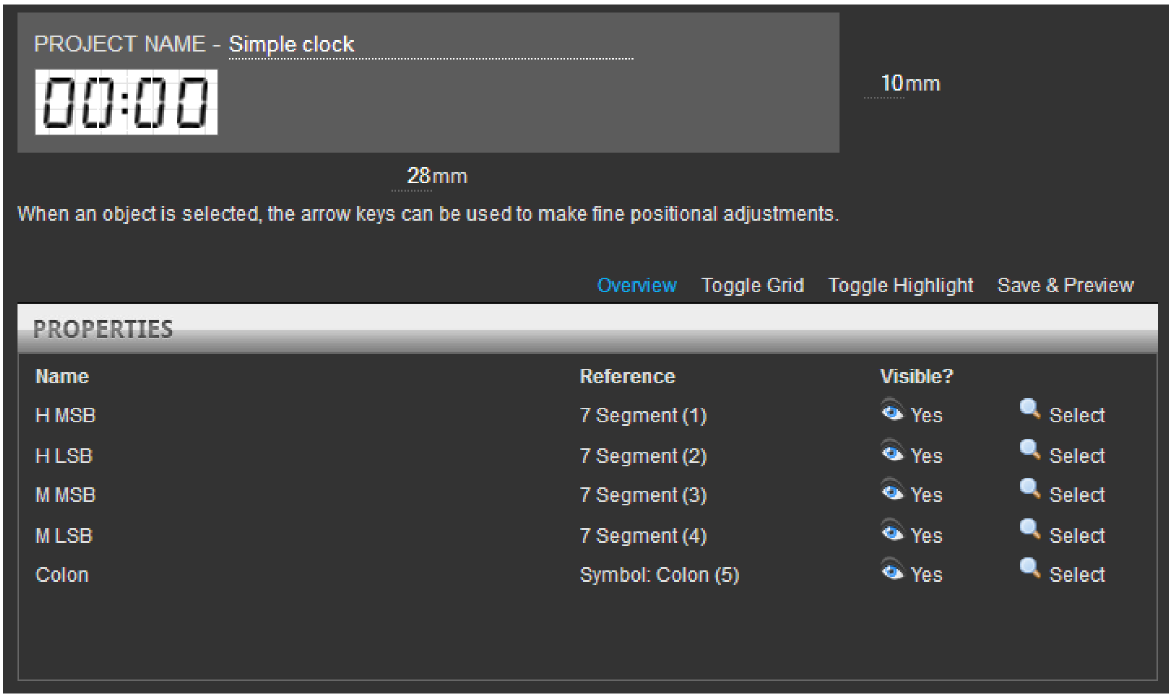
Name Colon

Position Top: 1 Left: 9.5

Size Height: 8 Width: 8

Comment

13. Click on Overview and check the result. It should look like this:



PROJECT NAME - Simple clock

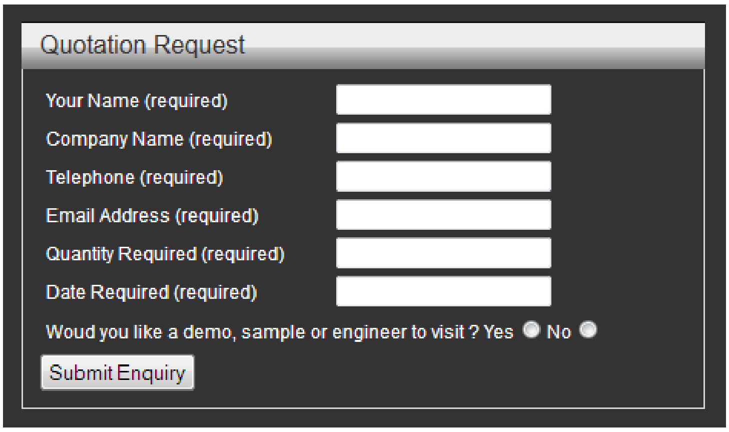
When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
H MSB	7 Segment (1)	Yes	Select
H LSB	7 Segment (2)	Yes	Select
M MSB	7 Segment (3)	Yes	Select
M LSB	7 Segment (4)	Yes	Select
Colon	Symbol: Colon (5)	Yes	Select

If it is different, then click on the faulty object and edit its properties.

14. Click on "QUOTE ME", fill in the "Quotation Request Form" and "Submit Enquiry"



Quotation Request

Your Name (required)

Company Name (required)

Telephone (required)

Email Address (required)

Quantity Required (required)

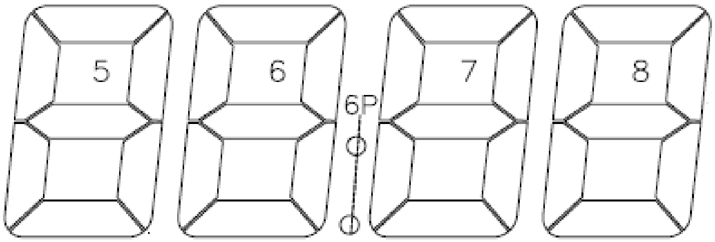
Date Required (required)

Woud you like a demo, sample or engineer to visit ? Yes ○ No ○

After a few days you will receive a price quotation for designing and manufacturing the glass.

If you are happy with the costs then our engineers will proceed designing the glass and you will receive the following data:

1. A drawing showing the names and the sizes of the segments:



2. A segment definition table

Number	Name
0	5A
1	5B
2	5C
3	5D
4	5E
5	5F
6	5G
7	

Number	Name
8	6A
9	6B
10	6C
11	6D
12	6E
13	6F
14	6G
15	6P

Number	Name
16	7A
17	7B
18	7C
19	7D
20	7E
21	7F
22	7G
23	

Number	Name
24	8A
25	8B
26	8C
27	8D
28	8E
29	8F
30	8G
31	

This information is needed to access the LCD controller at segment level.

Please note that this table is not the connection of the glass. The advantage of using KickStart™ is that there is no need to know the physical connection of the LCD. In fact the same firmware can be used to access differently connected LCDs.

3. Object definition table

Object	Type	Description
0	7-segment object	1st digit of the 4 * 14-segment component
1	7-segment object	2nd digit of the 4 * 14-segment component
2	7-segment object	3rd digit of the 4 * 14-segment component
3	7-segment object	4th digit of the 4 * 14-segment component
4	icon	colon between object 1 and object 2

This information is needed to access the LCD at object and component level.

4. Arrays of data needed for LCD initialization

Cmd_Seg_Tbl command with parameters
1, 119, 219, 218, 213, 209, 207, 223, 201, 215, 203, 210, 197, 208, 223, 205, 193, 201, 195, 202, 205, 203, 207, 195, 209, 199, 211, 199, 221, 197, 199, 159, 185, 253, 163, 251, 165, 153, 167, 191, 177, 189, 163, 163, 189, 241, 191, 191, 169, 189, 163, 139, 181, 185, 191, 135, 177, 229, 162, 162, 172, 160, 182, 158, 129, 29, 139, 27, 157, 25, 151, 23, 144, 145, 154, 151, 132, 149, 142, 141, 152, 143, 138, 137, 132, 139, 150, 134, 184, 5, 178, 3, 172, 1, 166, 63, 80, 61, 66, 59, 76, 57, 69, 117, 67, 119, 89, 113, 84, 241, 93, 110, 83, 108, 89, 106, 87, 97, 93, 111, 75, 109, 65, 97, 79, 101, 93, 91, 107, 89, 101, 219, 107, 217, 113, 215, 119, 21, 125, 19, 107, 17, 97, 15, 118, 79, 124, 73, 98, 75, 104, 70, 110, 68, 124, 66, 114, 64, 120, 55, 14, 53, 20, 51, 26, 61, 16, 51, 14, 49, 4, 55, 23, 48, 17, 175, 18, 41, 16, 11, 30, 9, 12, 47, 2, 33, 0, 33, 6, 32, 12, 27, 42, 21, 40, 11, 46, 29, 36, 31, 42, 5, 56, 27, 62, 1, 60, 47, 50, 77, 55, 9, 57, 11, 43, 5, 37, 4, 32, 35, 38, 17, 44, 191, 202, 125, 207, 250, 206, 185, 196, 119, 205, 244, 216, 115, 217, 240, 219, 237, 213, 233, 215, 203, 217, 201, 203, 199, 197, 245, 199, 235, 193, 229, 203, 219, 237, 213, 239, 203, 233, 221, 227, 223, 237, 197, 255, 219, 249, 193, 251, 239, 245, 141, 246, 201, 248, 203, 234, 197, 228, 196, 231, 67, 225, 129, 235, 255, 141, 61, 142, 186, 137, 249, 131, 55, 140, 180, 159, 51, 152, 176, 154, 173, 148, 169, 150, 139, 152, 137, 138, 135, 132, 181, 134, 171, 128, 165, 138, 155, 172, 149, 174, 139, 168, 157, 162, 159, 172, 133, 190, 155, 184, 129, 186, 175, 180, 205, 181, 137, 187, 139, 169, 133, 167, 132, 166, 3, 160, 193, 170, 63, 76, 253, 77, 122, 72, 57, 66, 247, 79, 116, 94, 243, 91, 112, 89, 109, 87, 105, 85, 75, 91, 73, 73, 71, 71, 117, 69, 107, 67, 101, 73, 91, 111, 85, 109, 75, 107, 93, 97, 95, 111, 69, 125, 91, 123, 65, 121, 111, 115, 93, 121, 91, 103, 89, 109, 79, 99, 101, 105, 3, 119, 1, 125, 127, 3, 29, 9, 27, 23, 25, 29, 39, 11, 181, 1, 179, 31, 177, 21, 111, 2, 44, 24, 42, 22, 40, 20, 37, 18, 39, 8, 33, 7, 161, 62, 27, 24, 61, 18, 59, 12, 57, 6, 55, 24, 29, 10, 27, 20, 25, 6, 7, 8, 29, 18, 27, 28, 25, 6, 23, 38, 7, 36, 11, 39, 5

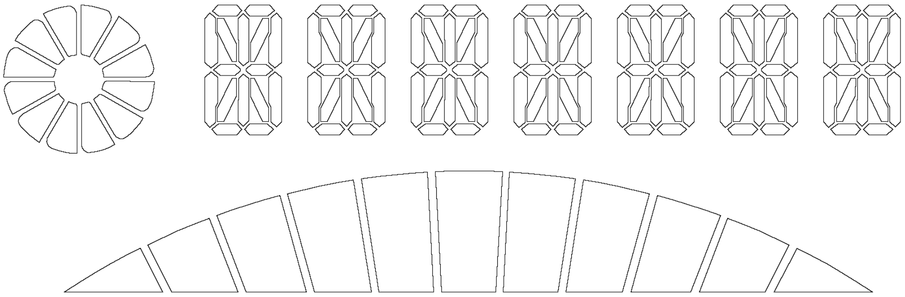
Cmd_Obj_Tbl command with parameters
2, 66, 66, 117, 92, 11, 6, 37, 122, 35, 110, 57, 110, 47, 82, 45, 70, 43, 178, 41, 190, 55, 178, 53, 134, 43, 134, 41, 134, 27, 232, 25, 202, 31, 44, 28, 14, 34, 80, 32, 114, 38, 148, 52, 190, 62, 170, 4, 210, 18, 218, 16, 218, 14, 194, 12, 194, 10, 218, 8, 218, 6, 210, 4, 210, 2, 218, 0

Cmd_LCD_cnf command with parameters
16, 3, 3, 1, 34, 185

The contents of these tables are used in Cmd_Seg_Tbl, Cmd_Obj_Tbl and Cmd_LCD_cnf commands. See Chapter "4.3.5. LCD initialization example" for details. Once these tables are downloaded to Kickstart™ LCD Controller, the segments and objects, as defined above, are accessible.

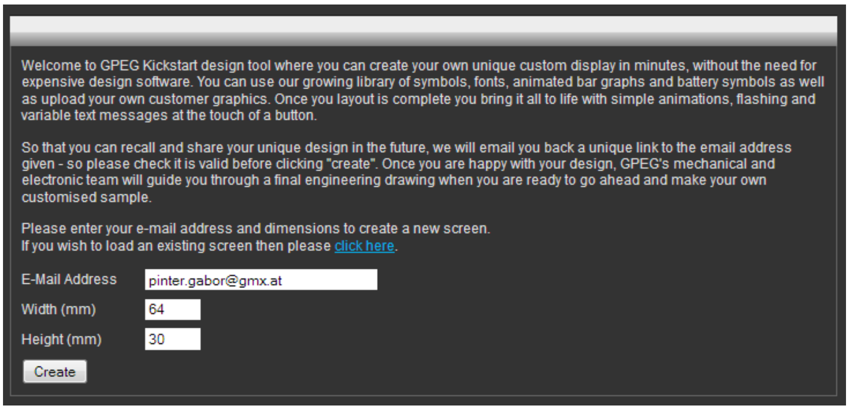
Example 2: Part of the LCD used in KickStart™ development kit

The target is to design the following LCD:



The on-line tool is used for approximation only in this example, because neither the 16-segment digit nor the rounded top bar are available in the TOOLBOX. For detailed work see the next example.

1. Open the KickStart™ on-line tool.
2. Enter your email address.
3. Define the size of the LCD as 64x30 mm and click "Create".



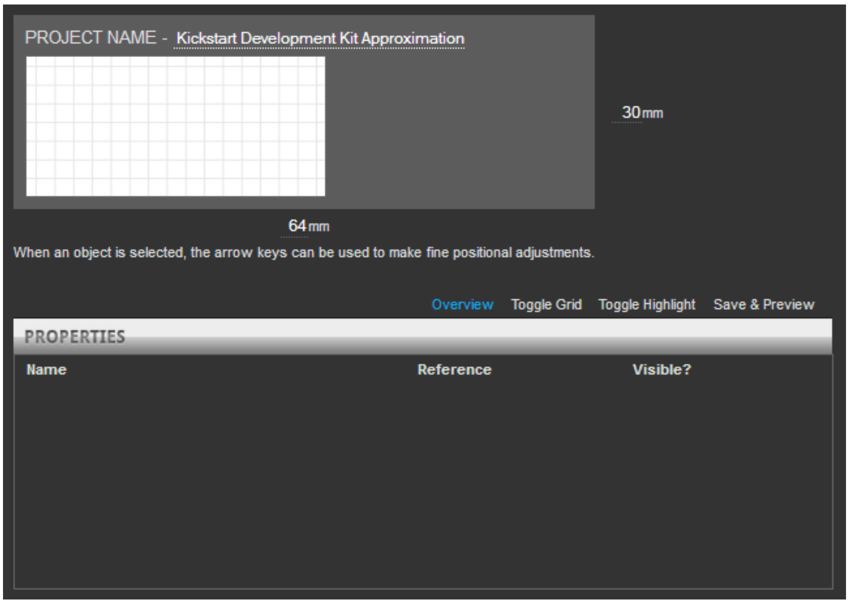
Welcome to GPEG Kickstart design tool where you can create your own unique custom display in minutes, without the need for expensive design software. You can use our growing library of symbols, fonts, animated bar graphs and battery symbols as well as upload your own customer graphics. Once you layout is complete you bring it all to life with simple animations, flashing and variable text messages at the touch of a button.

So that you can recall and share your unique design in the future, we will email you back a unique link to the email address given - so please check it is valid before clicking "create". Once you are happy with your design, GPEG's mechanical and electronic team will guide you through a final engineering drawing when you are ready to go ahead and make your own customised sample.

Please enter your e-mail address and dimensions to create a new screen.
If you wish to load an existing screen then please click here.

E-Mail Address: pinter.gabor@gmx.at
Width (mm): 64
Height (mm): 30

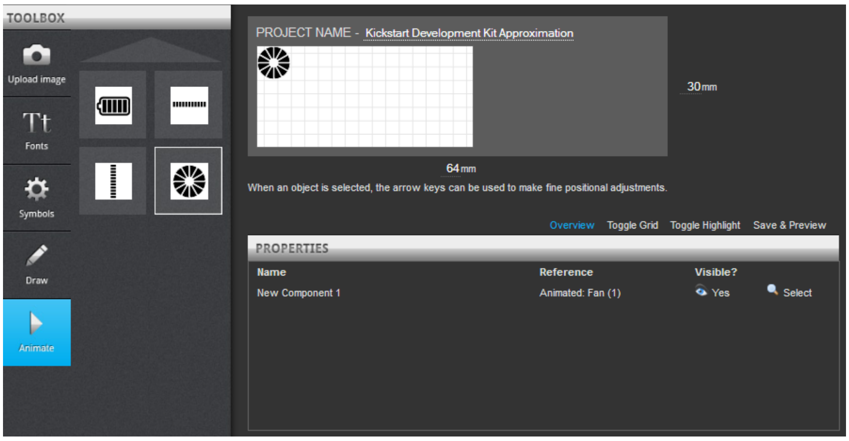
Click on PROJECT NAME and type "KickStart™ Development Kit Approximation".



PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

4. Select the disk shape from the Animate menu.



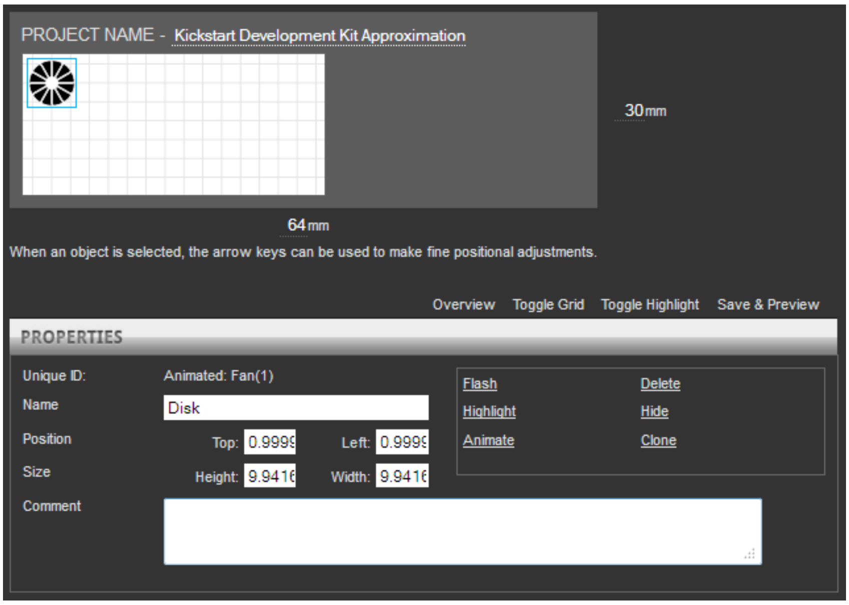
PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
New Component 1	Animated: Fan (1)	Yes	Select

5. A disk shape object will appear in the top left corner of the design screen. Position and size are wrong, but do not worry, just click on "New Component 1" in PROPERTIES window and enter name Name="Disk", position Top=1, Left=1, size Height=10, Width=10.



PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID: Animated: Fan(1)

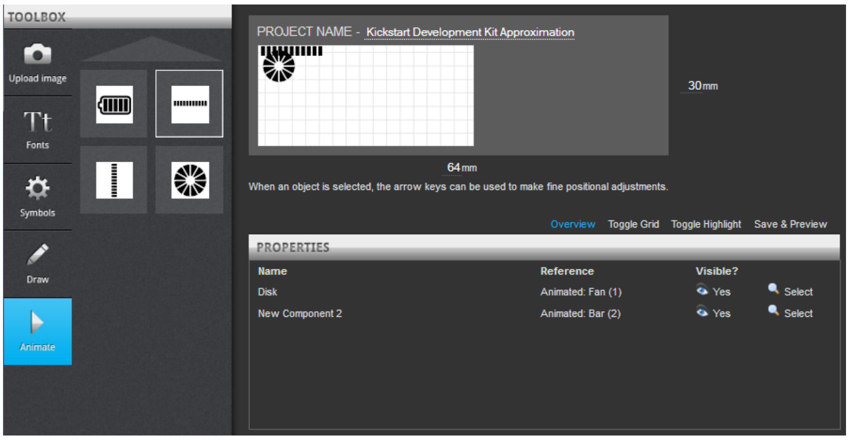
Name: Disk

Position: Top: 0.9999 Left: 0.9999

Size: Height: 9.9416 Width: 9.9416

Comment

6. Select the bar shape object from the Animate menu.



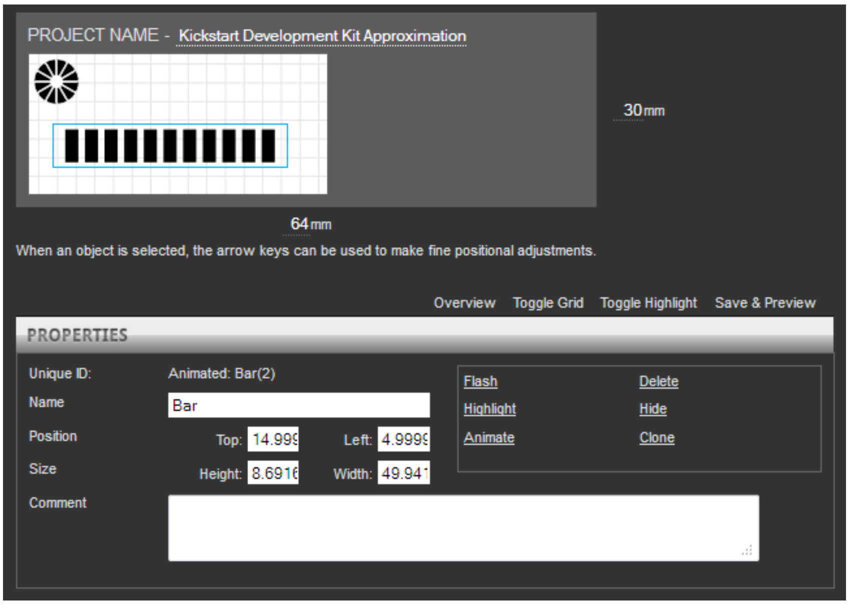
PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
Disk	Animated: Fan (1)	Yes	Select
New Component 2	Animated: Bar (2)	Yes	Select

7. A bar shape object will appear in the top left corner of the design screen. Position and size are wrong, and the top is not rounded, but we need it only for approximation. Click on "New Component 2" in PROPERTIES window and enter name Name="Bar", position Top=1, Left=1, size Height=12, Width=50.



PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID: Animated: Bar(2)

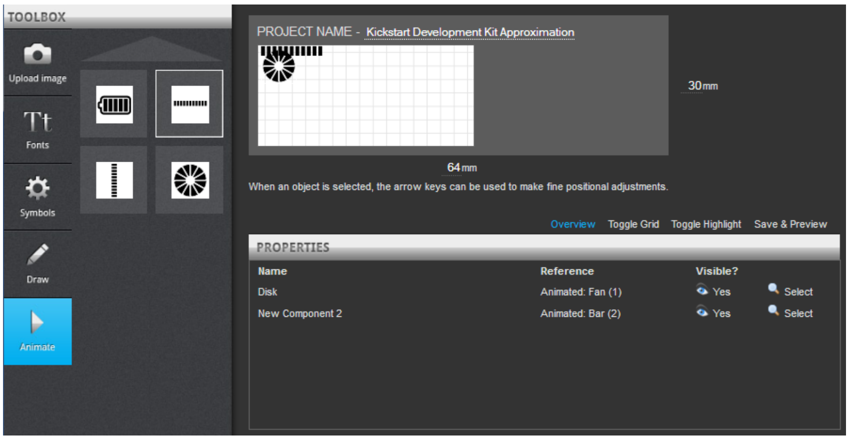
Name: Bar

Position: Top: 14.999 Left: 4.9999

Size: Height: 8.6916 Width: 49.941

Comment

8. Select the segmented text shape component from the Fonts menu.

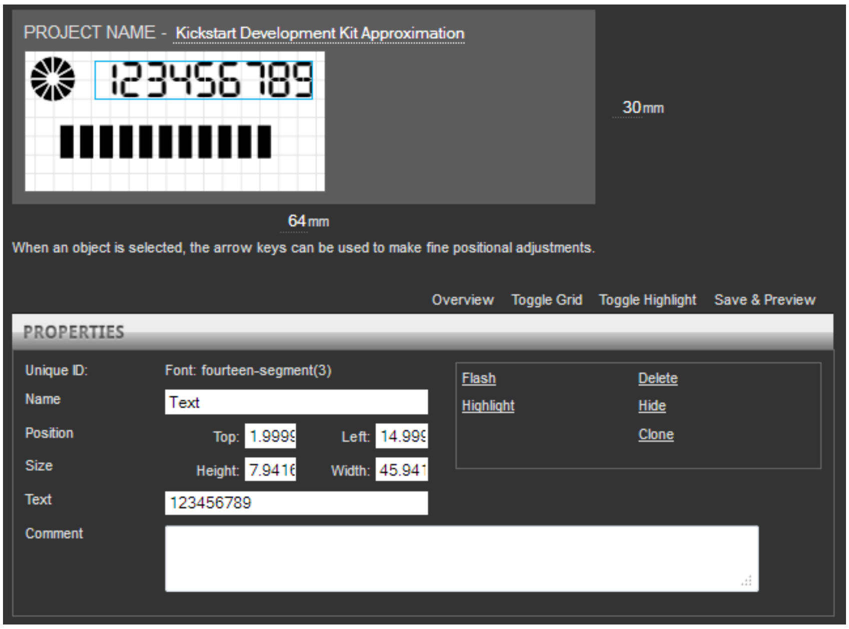


When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
Disk	Animated: Fan (1)	Yes	Select
New Component 2	Animated: Bar (2)	Yes	Select

9. A segmented text shape component will appear in the top left corner of the design screen. Position and size are wrong, and the font is wrong type, but we need it only for approximation. Click on "New Component 3" in PROPERTIES window and enter name Name="Text", position Top=2, Left=15, size Height=8, Text="1234566789".



When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID: Font: fourteen-segment(3)

Name: Text

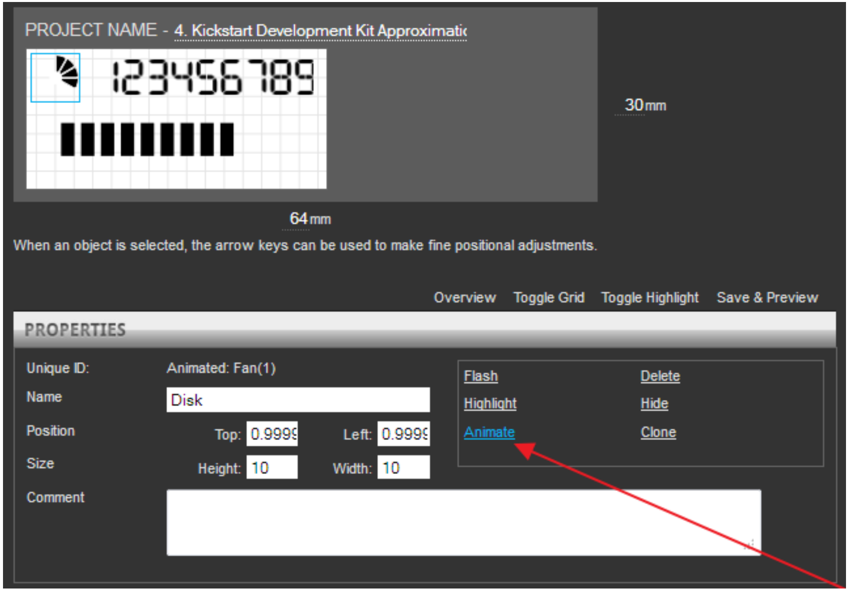
Position: Top: 1.9999 Left: 14.999

Size: Height: 7.9416 Width: 45.941

Text: 123456789

Comment:

10. Click on the disk shape and then on "Animate".



PROJECT NAME - 4. Kickstart Development Kit Approximatic

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID: Animated: Fan(1)

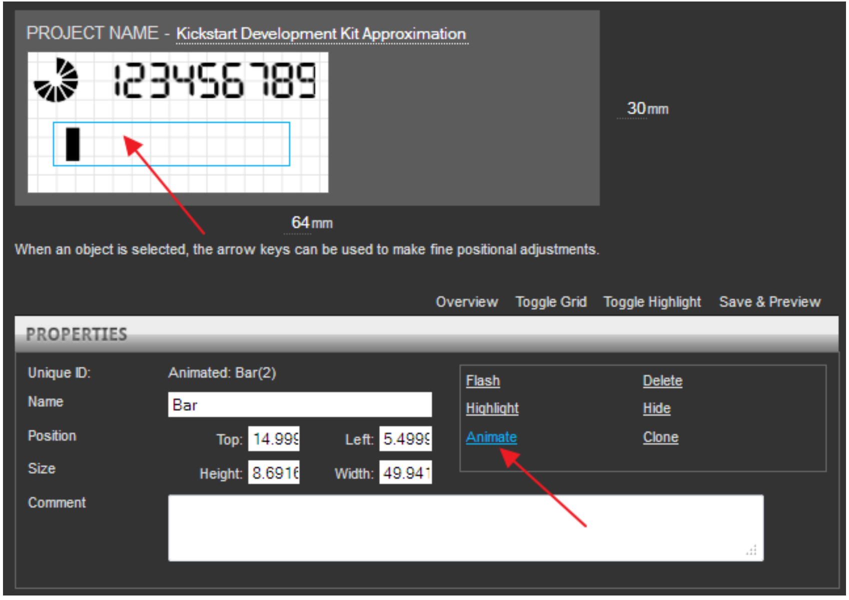
Name: Disk

Position: Top: 0.9999 Left: 0.9999

Size: Height: 10 Width: 10

Comment:

11. Repeat it with the bar shape too, and your design will come to life.



PROJECT NAME - Kickstart Development Kit Approximation

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID: Animated: Bar(2)

Name: Bar

Position: Top: 14.999 Left: 5.4999

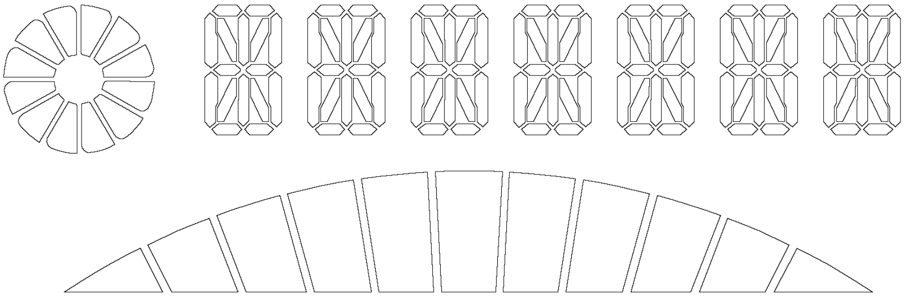
Size: Height: 8.6916 Width: 49.941

Comment:

12. And if you are interested, how to design this screen with engineering precision, read the next example.

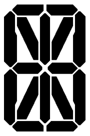
13. Example 3: Part of the LCD used in KickStart™ development kit

The target is to design the following LCD with engineering precision:

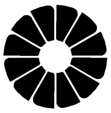


1. Before you start using the on-line tool you have to create the precise drawing of all objects.

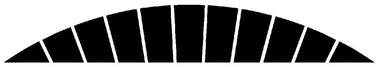
2. Open your favourite drawing program and create a blank black and white PNG image, 600 dpi resolution, 10x16 mm size, draw the following picture and save it as "16-segment.png".



3. Create a blank black and white PNG image, 600 dpi resolution, 15x15 mm size, draw the following picture and save it as "Disk.png".



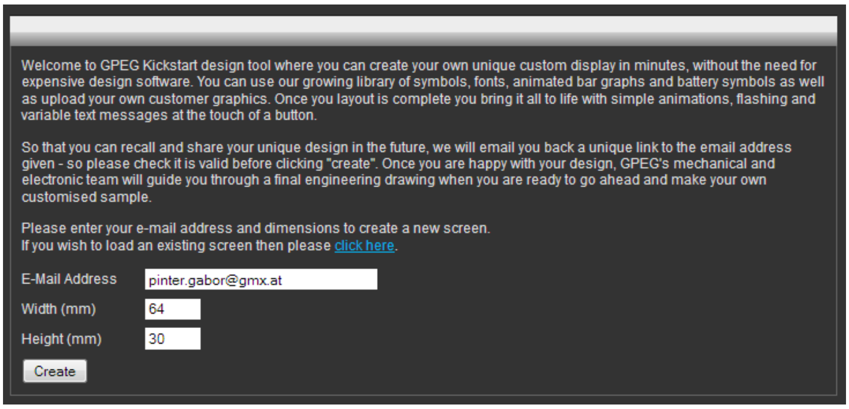
4. Create a blank black and white PNG image, 600 dpi resolution, 50x8 mm size, draw the following picture and save it as "Bar.png".



5. Open the KickStart™ on-line tool.

6. Enter your email address.

7. Define the size of the LCD as 64x30 mm and click "Create".



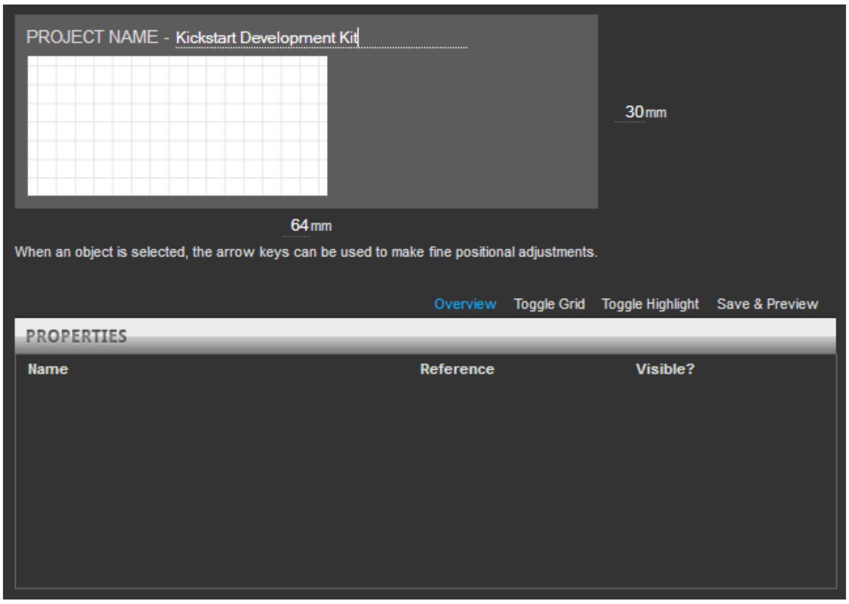
Welcome to GPEG Kickstart design tool where you can create your own unique custom display in minutes, without the need for expensive design software. You can use our growing library of symbols, fonts, animated bar graphs and battery symbols as well as upload your own customer graphics. Once you layout is complete you bring it all to life with simple animations, flashing and variable text messages at the touch of a button.

So that you can recall and share your unique design in the future, we will email you back a unique link to the email address given - so please check it is valid before clicking "create". Once you are happy with your design, GPEG's mechanical and electronic team will guide you through a final engineering drawing when you are ready to go ahead and make your own customised sample.

Please enter your e-mail address and dimensions to create a new screen.
If you wish to load an existing screen then please click here.

E-Mail Address: pinter.gabor@gmx.at
Width (mm): 64
Height (mm): 30

8. Click on PROJECT NAME and type "KickStart™ Development Kit".



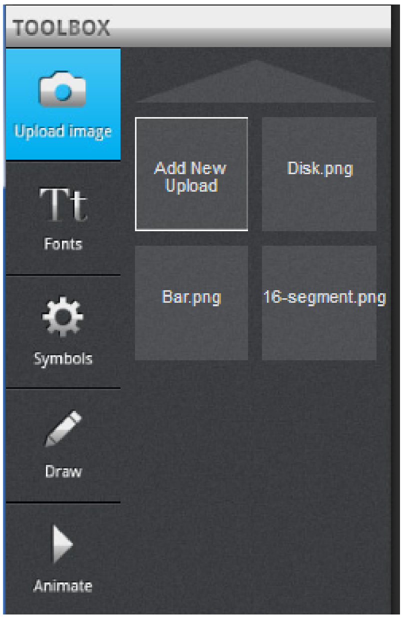
PROJECT NAME - Kickstart Development Kit

30mm

64mm

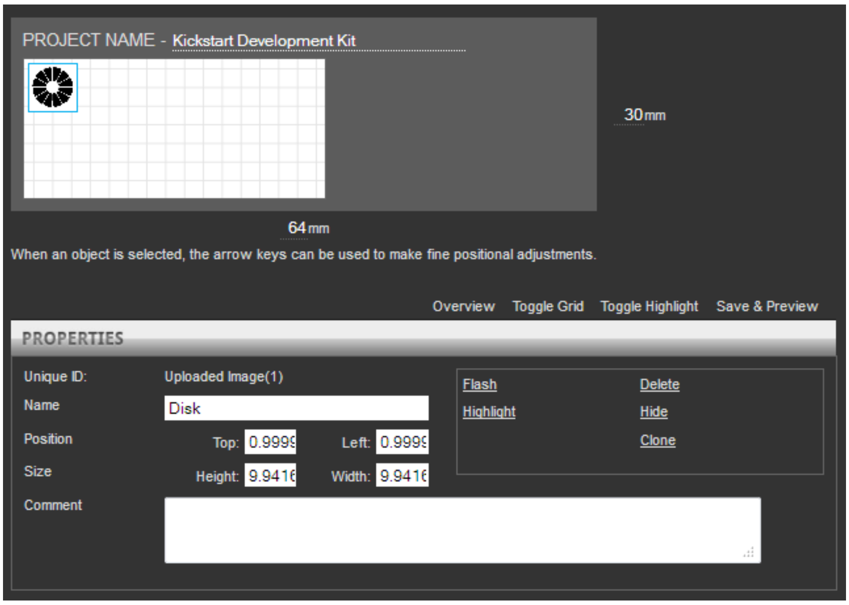
When an object is selected, the arrow keys can be used to make fine positional adjustments.

9. Upload your images in TOOLBOX, "Upload image", Add New Upload". After you uploaded all three images the screen should look like this:



10. Select Disk.png from the "Upload image" menu.

11. A disk shape object will appear in the top left corner of the design screen. Position and size are wrong, but do not worry, just click on "New Component 1" in PROPERTIES window and enter name Name="Disk", position Top=1, Left=1, size Height=10, Width=10.



PROJECT NAME - Kickstart Development Kit

30mm

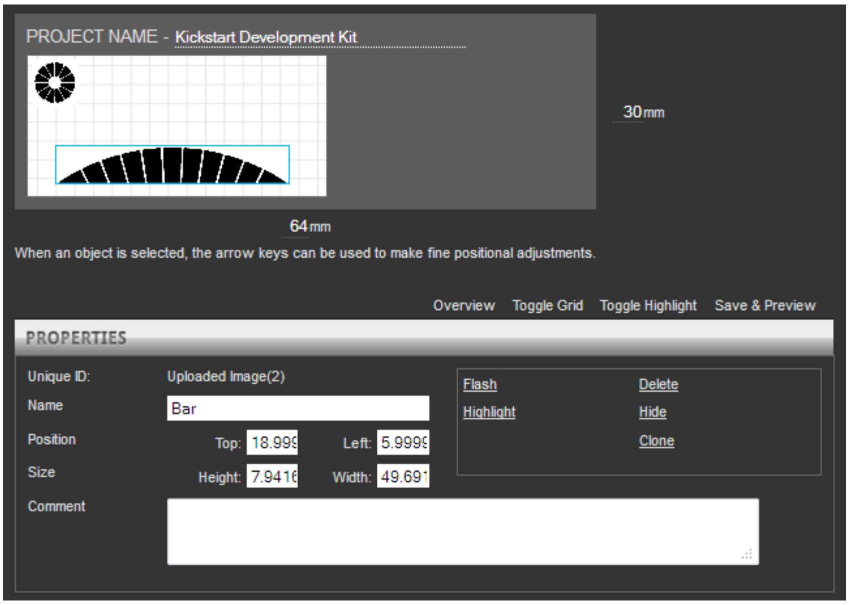
64mm

When an object is selected, the arrow keys can be used to make fine positional adjustments.

Unique ID:	Uploaded Image(1)	
Name	Disk	
Position	Top: 0.9999	Left: 0.9999
Size	Height: 9.9416	Width: 9.9416
Comment		

12. Select Bar.png from the "Upload image" menu.

13. A bar shape object will appear in the top left corner of the design screen. Position and size are wrong, but do not worry, just click on "New Component 2" in PROPERTIES window and enter name Name="Bar", position Top=19, Left=6, size Height=8, Width=50.



PROJECT NAME - Kickstart Development Kit

30mm

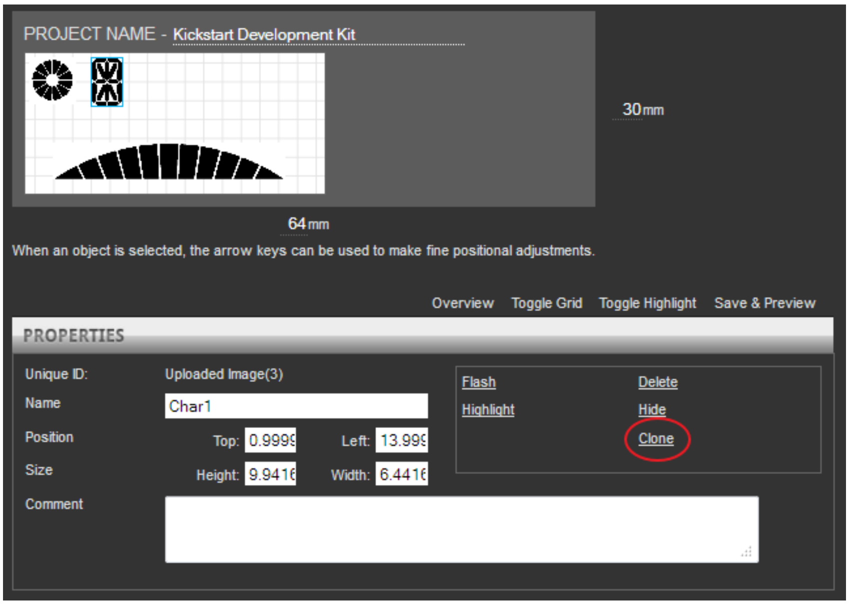
64mm

When an object is selected, the arrow keys can be used to make fine positional adjustments.

Unique ID:	Uploaded Image(2)	
Name	Bar	
Position	Top: 18.999	Left: 5.9999
Size	Height: 7.9416	Width: 49.691
Comment		

14. Select 16-segment.png from the "Upload image" menu.

15. A 16-segment character shape object will appear in the top left corner of the design screen. Position and size are wrong, but do not worry, just click on "New Component 3" in PROPERTIES window and enter name Name="Char1", position Top=1, Left=14, size Height=10, Width=6.4.



PROJECT NAME - Kickstart Development Kit

30mm

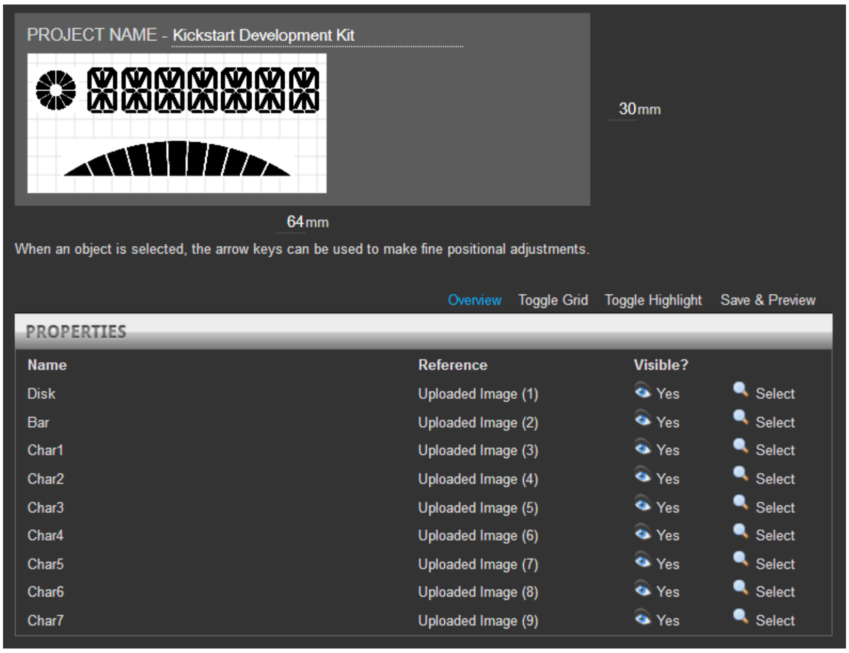
64mm

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Unique ID:	Uploaded Image(3)
Name	Char1
Position	Top: 0.9999 Left: 13.999
Size	Height: 9.9416 Width: 6.4416
Comment	

16. Use the "Clone" function to create six more characters and arrange the objects on the design screen until they look like this:



PROJECT NAME - Kickstart Development Kit

30mm

64mm

When an object is selected, the arrow keys can be used to make fine positional adjustments.

PROPERTIES

Name	Reference	Visible?	
Disk	Uploaded Image (1)	Yes	Select
Bar	Uploaded Image (2)	Yes	Select
Char1	Uploaded Image (3)	Yes	Select
Char2	Uploaded Image (4)	Yes	Select
Char3	Uploaded Image (5)	Yes	Select
Char4	Uploaded Image (6)	Yes	Select
Char5	Uploaded Image (7)	Yes	Select
Char6	Uploaded Image (8)	Yes	Select
Char7	Uploaded Image (9)	Yes	Select

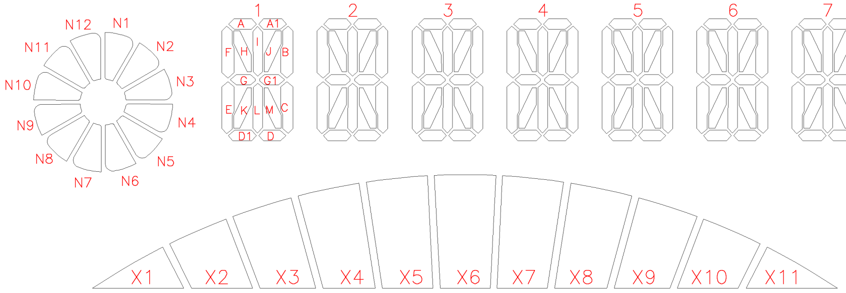
If it is different, then click on the faulty object and edit its properties.

17. Click on "QUOTE ME", fill in the "Quotation Request Form" and "Submit Enquiry"

After a few days you will receive a price quotation for designing and manufacturing the glass.

If you are happy with the costs then our engineers will proceed designing the glass and you will receive the following data:

1. A drawing showing the names and the sizes of the segments:



2. A segment definition table

Number	Name
0	1A
1	1A1
2	1B
3	1C
4	1D
5	1D1
6	1E
7	1F
8	1G
9	1G1
10	1H
11	1I
12	1J
13	1M
14	1L
15	1K

Number	Name
16	2A
17	2A1
18	2B
19	2C
20	2D
21	2D1
22	2E
23	2F
24	2G
25	2G1
26	2H
27	2I
28	2J
29	2M
30	2L
31	2K

Number	Name
24	3A
25	3A1
26	3B
27	3C
28	3D
29	3D1
30	3E
31	3F
32	3G
33	3G1
34	3H
35	3I
36	3J
37	3M
38	3L
39	3K

Number	Name
32	4A
33	4A1
34	4B
35	4C
36	4D
37	4D1
38	4E
39	4F
40	4G
41	4G1
42	4H
43	4I
44	4J
45	4M
46	4L
47	4K

Number	Name
48	5A
49	5A1
50	5B
51	5C
52	5D
53	5D1
54	5E
55	5F
56	5G
57	5G1
58	5H
59	5I
60	5J
61	5M
62	5L
63	5K

Number	Name
64	6A
65	6A1
66	6B
67	6C
68	6D
69	6D1
70	6E
71	6F
72	6G
73	6G1
74	6H
75	6I
76	6J
77	6M
78	6L
79	6K

Number	Name
80	7A
81	7A1
82	7B
83	7C
84	7D
85	7D1
86	7E
87	7F
88	7G
89	7G1
90	7H
91	7I
92	7J
93	7M
94	7L
95	7K

Number	Name
96	N1
97	N2
98	N3
99	N4
100	N5
101	N6
102	N7
103	N8
104	N9
105	N10
106	N11
107	N12
108	
109	
110	
111	

Number	Name
112	X1
113	X2
114	X3
115	X4
116	X5
117	X6
118	X7
119	X8
120	X9
121	X10
122	X11
123	
124	
125	
126	
127	

This information is needed to access the LCD controller at segment level.

Please note that this table is not the connection of the glass. The advantage of using KickStart™ is that there is no need to know the physical connection of the LCD. In fact the same firmware can be used to access differently connected LCDs.

3. Object definition table

Object	Type	Description
0	16-segment object	1st digit of the 7 * 16-segment component
1	16-segment object	2nd digit of the 7 * 16-segment component
2	16-segment object	3rd digit of the 7 * 16-segment component
3	16-segment object	4th digit of the 7 * 16-segment component
4	16-segment object	5th digit of the 7 * 16-segment component
5	16-segment object	6th digit of the 7 * 16-segment component
6	16-segment object	7th digit of the 7 * 16-segment component
7	bar object	Disk shape
8	bar object	Bar shape

This information is needed to access the LCD at object and component level.

4. Arrays of data needed for LCD initialization commands.

Cmd_Seg_Tbl command with parameters
1, 119, 219, 218, 213, 209, 207, 223, 201, 215, 203, 210, 197, 208, 223, 205, 193, 201, 195, 202, 205, 203, 207, 195, 209, 199, 211, 199, 221, 197, 199, 159, 185, 253, 163, 251, 165, 153, 167, 191, 177, 189, 163, 163, 189, 241, 191, 191, 169, 189, 163, 139, 181, 185, 191, 135, 177, 229, 162, 162, 172, 160, 182, 158, 129, 29, 139, 27, 157, 25, 151, 23, 144, 145, 154, 151, 132, 149, 142, 141, 152, 143, 138, 137, 132, 139, 150, 134, 184, 5, 178, 3, 172, 1, 166, 63, 80, 61, 66, 59, 76, 57, 69, 117, 67, 119, 89, 113, 84, 241, 93, 110, 83, 108, 89, 106, 87, 97, 93, 111, 75, 109, 65, 97, 79, 101, 93, 91, 107, 89, 101, 219, 107, 217, 113, 215, 119, 21, 125, 19, 107, 17, 97, 15, 118, 79, 124, 73, 98, 75, 104, 70, 110, 68, 124, 66, 114, 64, 120, 55, 14, 53, 20, 51, 26, 61, 16, 51, 14, 49, 4, 55, 23, 48, 17, 175, 18, 41, 16, 11, 30, 9, 12, 47, 2, 33, 0, 33, 6, 32, 12, 27, 42, 21, 40, 11, 46, 29, 36, 31, 42, 5, 56

Cmd_Obj_Tbl command with parameters
2, 66, 66, 117, 92, 11, 6, 37, 122, 35, 110, 57, 110, 47, 82, 45, 70, 43, 178, 41, 190, 55, 178, 53, 134, 43, 134, 41, 134, 27, 232, 25, 202, 31, 44, 28, 14, 34, 80, 32, 114, 38, 148, 52, 190, 62, 170

Cmd_LCD_cnf command with parameters
16, 3, 3, 1, 34, 185

The contents of these tables are used in Cmd_Seg_Tbl, Cmd_Obj_Tbl and Cmd_LCD_cnf commands. See Chapter "4.3.5. LCD initialization example" for details.

You do not have to understand the meaning of these tables. In fact they are meant to be cryptic. Once they are downloaded to KickStart™ LCD Controller, the objects, as defined above are accessible.

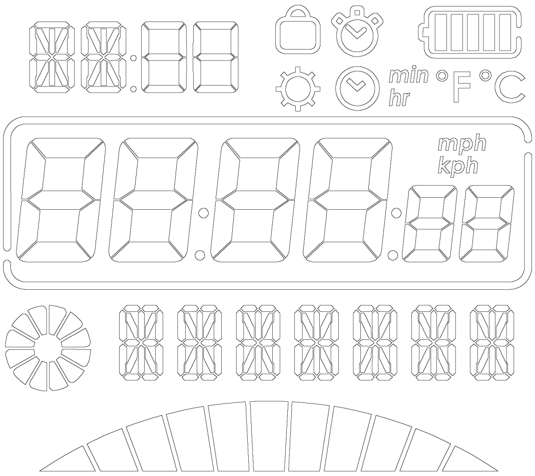
3. KickStart™ development kit

Kickstart™ development kit consist of the following elements:

1. LCD designed with KickStart™ on-line tool with KickStart™ LCD driver chip built into its flat cable
2. KickStart™ Development Board to connect the KickStart™ LCD to PC
3. Kickdev demo program to test the LCD

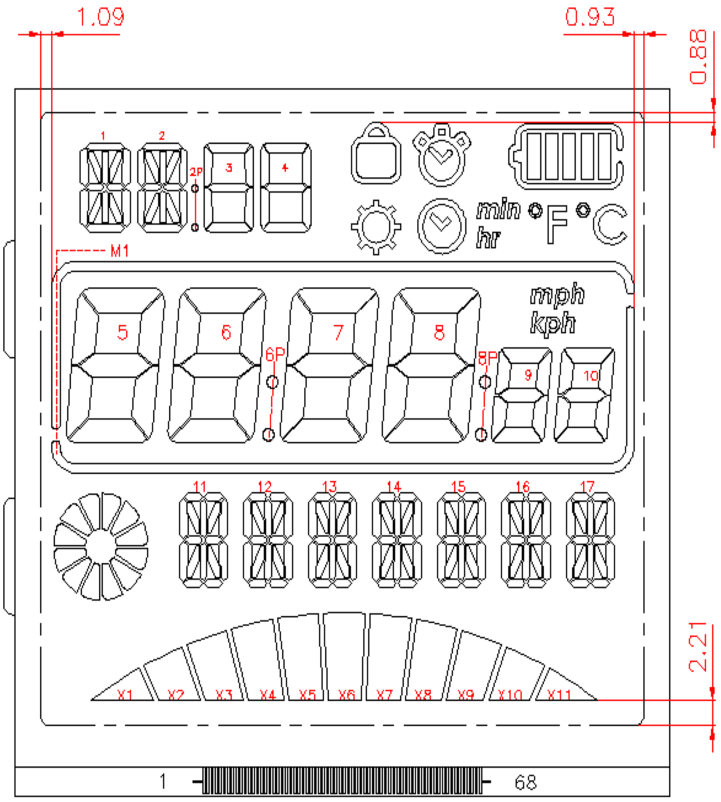
3.1. LCD of the development kit

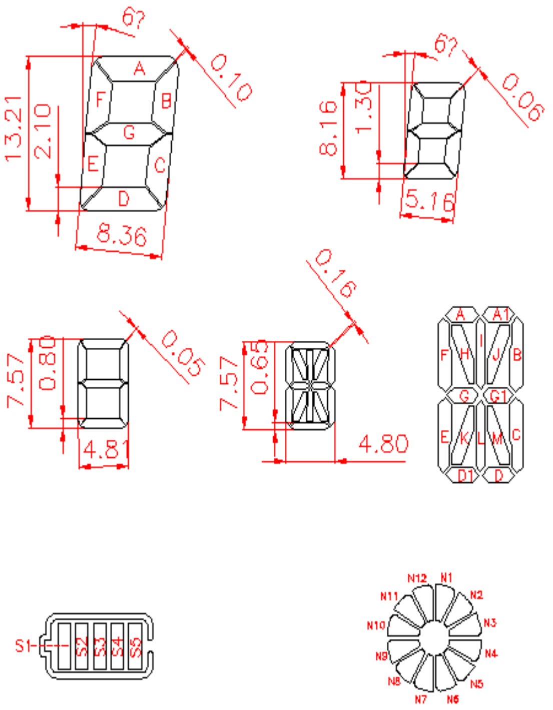
This LCD can be designed with the on-line tool. Part of the design process was shown in Example 2 and 3 of Chapter "2. KickStart™ on-line tool".



The following chapters list the segment and object assignments of the LCD in details. These are needed to access the LCD.

3.1.1. Segment names





3.1.2. Segment definition table

Number	Name
0	1A
1	1B
2	1C
3	1D
4	1E
5	1F
6	1G
7	1G1
8	1H
9	1I
10	1J
11	1K
12	1L
13	1M
14	2A
15	2B
16	2C
17	2D
18	2E
19	2F
20	2G
21	2G1
22	2H
23	2I
24	2J
25	2K
26	2L
27	2M
28	3A
29	3B
30	3C
31	3D
32	3E
33	3F
34	3G
35	4A
36	4B
37	4C
38	4D
39	4E
40	4F
41	4G
42	2P
43	5A
44	5B
45	5C
46	5D
47	5E
48	5F
49	5G
50	6A
51	6B
52	6C
53	6D
54	6E
55	6F
56	6G
57	7A
58	7B
59	7C
60	7D
61	7E
62	7F
63	7G
64	8A
65	8B
66	8C
67	8D
68	8E
69	8F
70	8G
71	9A
72	9B
73	9C
74	9D
75	9E
76	9F
77	9G
78	10A
79	10B
80	10C
81	10D
82	10E
83	10F
84	10G
85	6P
86	8P
87	11C
88	11D
89	11E
90	11C
91	11D
92	11D1
93	11E
94	11F
95	11G
96	11G1
97	11H
98	11I
99	11J
100	11M
101	11L
102	11K
103	12C
104	12D
105	12E
106	12C
107	12D
108	12D1
109	12E
110	12F
111	12G
112	12G1
113	12H
114	12I
115	12J
116	12M
117	12L
118	12K
119	13C
120	13D
121	13E
122	13C
123	13D
124	13D1
125	13E
126	13F
127	13G
128	13G1
129	13H
130	13I
131	13J
132	13M
133	13L
134	13K
135	14C
136	14D
137	14E
138	14C
139	14D
140	14D1
141	14E
142	14F
143	14G
144	14G1
145	14H
146	14I
147	14J
148	14M
149	14L
150	14K
151	15C
152	15D
153	15E
154	15C
155	15D
156	15D1
157	15E
158	15F
159	15G
160	15G1
161	15H
162	15I
163	15J
164	15M
165	15L
166	15K
167	16C
168	16D
169	16E
170	16C
171	16D
172	16D1
173	16E
174	16F
175	16G
176	16G1
177	16H
178	16I
179	16J
180	16M
181	16L
182	16K
183	17C
184	17D
185	17E
186	17C
187	17D
188	17D1
189	17E
190	17F
191	17G
192	17G1
193	17H
194	17I
195	17J
196	17M
197	17L
198	17K
199	N1
200	N2
201	N3
202	N4
203	N5
204	N6
205	N7
206	N8
207	N9
208	N10
209	N11
210	N12
211	X1
212	X2
213	X3
214	X4
215	X5
216	X6
217	X7
218	X8
219	X9
220	X10
221	X11
222	S1
223	S2
224	S3
225	S4
226	S5
227	lock
228	sun
229	alarm
230	clock
231	min
232	hr
233	F
234	C
235	mph
236	kph
237	frame

This information is needed to access the LCD controller at segment level.

Please note that this table is not the connection of the glass. The advantage of using KickStart™ is that there is no need to know the physical connection of the LCD. In fact the same firmware can be used to access differently connected

3.1.3. Object definition table

Object	Type	Description
0	14-segment object	1st digit of top left 2 * 14-segment component
1	14-segment object	2nd digit of top left 2 * 14-segment component
2	7-segment object	1st digit of top left 2 * 7-segment component
3	7-segment object	2nd digit of top left 2 * 7-segment component
4	icon	colon between object 1 and object 2
5	7-segment object	1st digit of the centre 6 * 7-segment component
6	7-segment object	2nd digit of the centre 6 * 7-segment component
7	7-segment object	3rd digit of the centre 6 * 7-segment component
8	7-segment object	4th digit of the centre 6 * 7-segment component
9	7-segment object	5th digit of the centre 6 * 7-segment component
10	7-segment object	6th digit of the centre 6 * 7-segment component
11	icon	colon between object 6 and object 7
12	icon	colon between object 8 and object 9
13	16-segment object	1st digit of the bottom 7 * 16-segment component
14	16-segment object	2nd digit of the bottom 7 * 16-segment component
15	16-segment object	3rd digit of the bottom 7 * 16-segment component
16	16-segment object	4th digit of the bottom 7 * 16-segment component
17	16-segment object	5th digit of the bottom 7 * 16-segment component
18	16-segment object	6th digit of the bottom 7 * 16-segment component
19	16-segment object	7th digit of the bottom 7 * 16-segment component
20	bar	disk-shaped, 12-segment object
21	bar	bottom, 11-segment object
22	bar	battery, 5-segment object
23	icon	lock
24	icon	sun
25	icon	alarm
26	icon	clock
27	icon	min
28	icon	hr
29	icon	F
30	icon	C
31	icon	mph
32	icon	kph
33	icon	frame

This information is needed to access the LCD at object and component level.

3.1.4. LCD initialization data

Cmd_Seg_Tbl command with parameters
1, 119, 219, 218, 213, 209, 207, 223, 201, 215, 203, 210, 197, 208, 223, 205, 193, 201, 195, 202, 205, 203, 207, 195, 209, 199, 211, 199, 221, 197, 199, 159, 185, 253, 163, 251, 165, 153, 167, 191, 177, 189, 163, 163, 189, 241, 191, 191, 169, 189, 163, 139, 181, 185, 191, 135, 177, 229, 162, 162, 172, 160, 182, 158, 129, 29, 139, 27, 157, 25, 151, 23, 144, 145, 154, 151, 132, 149, 142, 141, 152, 143, 138, 137, 132, 139, 150, 134, 184, 5, 178, 3, 172, 1, 166, 63, 80, 61, 66, 59, 76, 57, 69, 117, 67, 119, 89, 113, 84, 241, 93, 110, 83, 108, 89, 106, 87, 97, 93, 111, 75, 109, 65, 97, 79, 101, 93, 91, 107, 89, 101, 219, 107, 217, 113, 215, 119, 21, 125, 19, 107, 17, 97, 15, 118, 79, 124, 73, 98, 75, 104, 70, 110, 68, 124, 66, 114, 64, 120, 55, 14, 53, 20, 51, 26, 61, 16, 51, 14, 49, 4, 55, 23, 48, 17, 175, 18, 41, 16, 11, 30, 9, 12, 47, 2, 33, 0, 33, 6, 32, 12, 27, 42, 21, 40, 11, 46, 29, 36, 31, 42, 5, 56

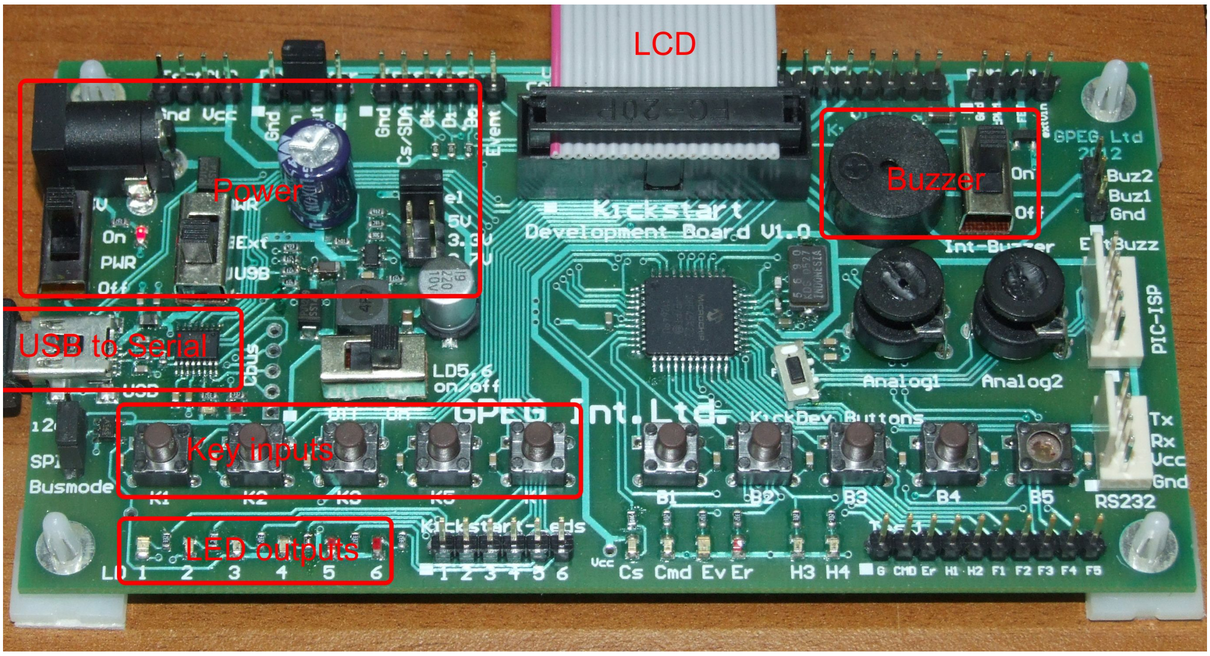
Cmd_Obj_Tbl command with parameters
2, 66, 66, 117, 92, 11, 6, 37, 122, 35, 110, 57, 110, 47, 82, 45, 70, 43, 178, 41, 190, 55, 178, 53, 134, 43, 134, 41, 134, 27, 232, 25, 202, 31, 44, 28, 14, 34, 80, 32, 114, 38, 148, 52, 190, 62, 170

Cmd_LCD_cnf command with parameters
16, 3, 3, 1, 34, 185

The contents of these tables are used in Cmd_Seg_Tbl, Cmd_Obj_Tbl and Cmd_LCD_cnf commands. See Chapter "4.3.5. LCD initialization example" for details. Once these tables are downloaded to KickStart™ LCD Controller, the segments and objects, as defined above, are accessible.

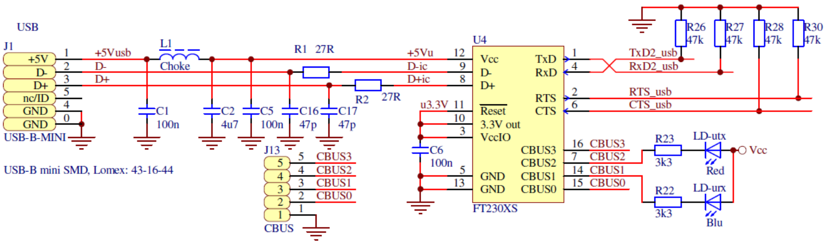
3.2. KickStart™ Development Board

The interface board uses FTDI FT230XS to convert USB data to serial line and Microchip PIC18F45K22 to convert serial line data to SPI or I²C used by KickStart™ LCD Controller.



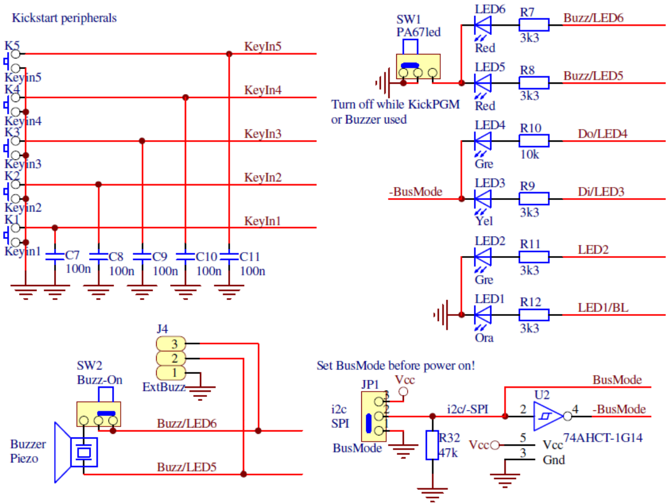
The board was designed to be both an evaluation and a development board. Only the parts marked in red are interesting for evaluation.

3.2.1. USB to serial converter



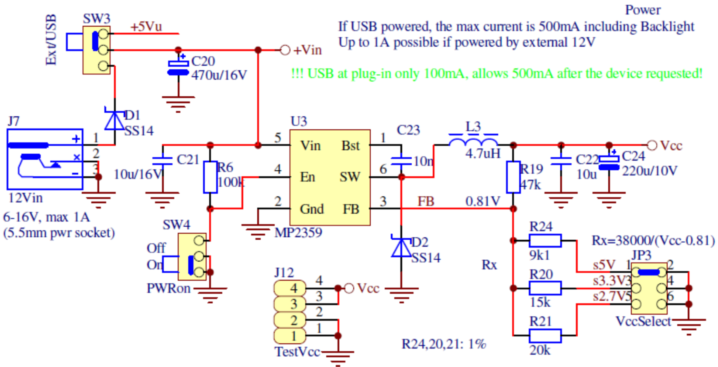
This part of the circuit converts the USB to serial line.

3.2.2. KickStart™ peripheries



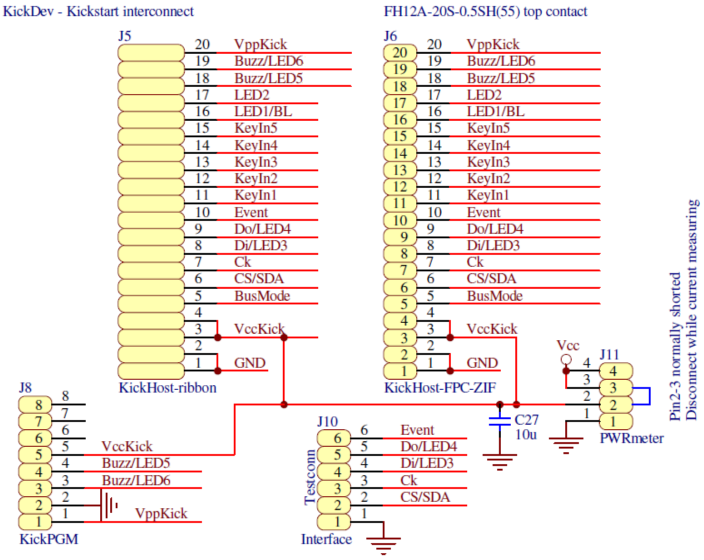
These are the key inputs and LED outputs of the KickStart™ LCD controller.

3.2.3. Power



This part of the circuit provides power for the LCD, the LCD controller and the PIC.

3.2.4. Connection to KickStart™ LCD controller



This is the connection to LCD through the KickStart™ LCD Controller.

3.2.5. Rest of the circuit

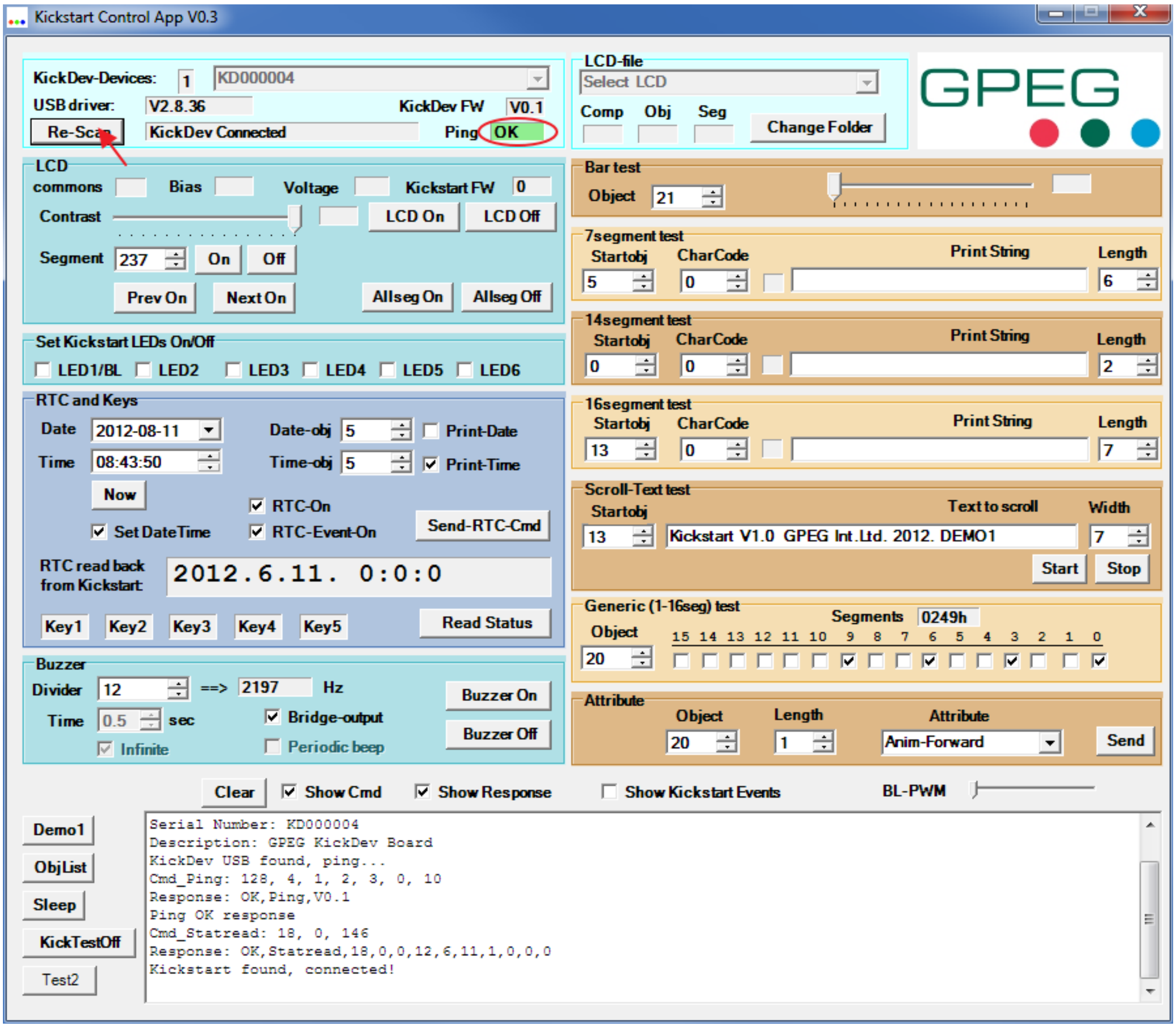
The rest of the circuits are the PIC processor, some unused peripheries and internally used auxiliary circuits.

3.3. Kickdev demo program

Kickdev demo program requires Windows XP or 7. It is downloadable from ... or can be installed from the KickStart™ development kit CD. To install the program, run Setup.exe and follow instructions on screen. After installation the computer may have to be rebooted.

3.3.1. Starting Kickdev for the first time

When KickStart™ Development Board is properly connected to an USB port of the PC and powered, the Kickdev application should start up like this:



KickDev-Devices: 1 KD000004
USB driver: V2.8.36
KickDev FW V0.1
KickDev Connected
Ping OK

commons Bias Voltage Kickstart FW 0
Contrast
Segment 237

□ LED1/BL □ LED2 □ LED3 □ LED4 □ LED5 □ LED6

Date 2012-08-11
Time 08:43:50
Date-obj 5 □ Print-Date
Time-obj 5 ☑ Print-Time
☑ Set DateTime ☑ RTC-On ☑ RTC-Event-On
RTC read back from Kickstart 2012.6.11. 0:0:0

Divider 12 ==> 2197 Hz
Time 0.5 sec
☑ Infinite ☑ Bridge-output □ Periodic beep

☑ Show Cmd ☑ Show Response □ Show Kickstart Events

Select LCD
Comp Obj Seg

Bar test: Object 21

	Startobj	CharCode	Print String	Length
7segment test	5	0		6
14segment test	0	0		2
16segment test	13	0		7

Scroll-Text test: Startobj 13, Text to scroll: Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1, Width 7

Generic (1-16seg) test: Object 20, Segments 0249h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
□	□	□	□	□	□	☑	□	□	☑	□	□	☑	□	□	☑

Attribute: Object 20, Length 1, Attribute Anim-Forward

```
Serial Number: KD000004
Description: GPEG KickDev Board
KickDev USB found, ping...
Cmd_Ping: 128, 4, 1, 2, 3, 0, 10
Response: OK,Ping,V0.1
Ping OK response
Cmd_Statread: 18, 0, 146
Response: OK,Statread,18,0,0,12,6,11,1,0,0,0
Kickstart found, connected!
```

If it is not the case then, reboot your PC, check USB cable, turn power off and on again and press the rescan button until the green OK appears.

After the green OK has appeared, firmware version and real time clock are read back and displayed correctly, the segment and object tables are automatically downloaded and all functions of the program are available.

4. KickStart™ LCD driver chip interface

KickStart™ supports modified I²C (2 wires) and SPI (4 wires) bus modes. For new development SPI is recommended, because it is faster.

Interface is selected by "BusMode" pin on the host connector. 1=I²C, 0=SPI. On the KickStart™ Development Board there is a jumper in the bottom left corner of the board which is used to select the BusMode.

The KickStart™ Development Board translates the bytes received from its USB/serial interface.

Detailed timing diagrams of both modes can be found in Chapter "5. KickStart™ LCD Controller hardware".

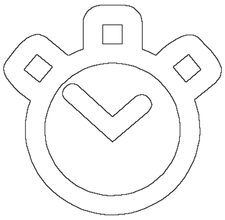
4.1. Standard objects

The definition of an object can be found in Chapter "1.2. What is an object?" at the beginning of this manual. KickStart™ can use any group of up to 16 segments as an object, and simplifies the use of some common objects. The following chapters will describe these common objects.

KickStart™ is also capable of animating object by automatically updating them in every 0.5 seconds. It can display digits of the real time clock, part of a scrolling text, can blink or form an animated icon.

4.1.1. Simple icon object

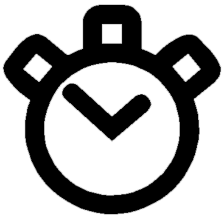
An object that consists of only one, sometimes discontinuous, segment is a simple icon. This is a typical simple icon object:



The discontinuous part are internally connected.

Example: A simple icon object accessed raw

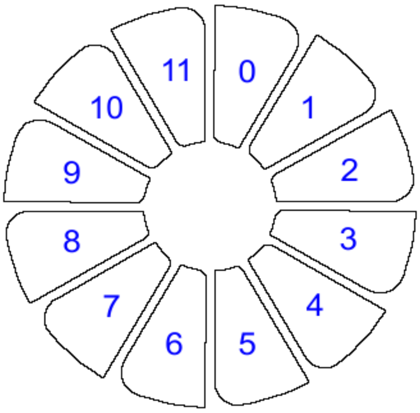
Writing the value of 1 with the Cmd_Obj_W command to a simple icon object, that looks like the one above, will create the following pattern:



More information about this command is in Chapter "4.3.9. Cmd_Obj_W".

4.1.2. Complex icon object

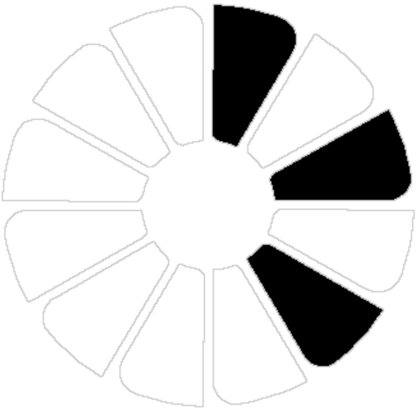
A complex icon consists of up to 16 segments. This is a typical complex icon object:



The blue numbers inside the segment define the order of the segments, and also define the bit position when the object is accessed in raw format.

Example: A complex icon object accessed raw

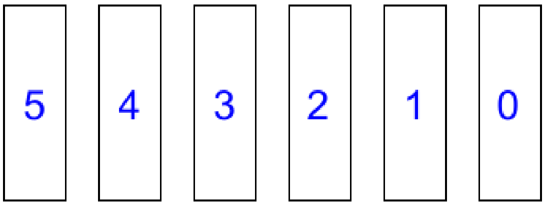
Writing binary pattern '010101' or decimal 21 with the Cmd_Obj_W command to a complex icon object, that looks like the one above, will create the following pattern:



More information about this command is in Chapter "4.3.9. Cmd_Obj_W".

4.1.3. Bar object

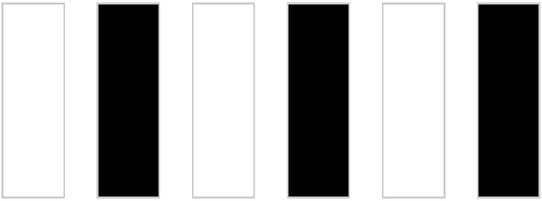
A bar object consists of up to 16 segments arranged vertically, horizontally or circularly side by side. This is a typical bar object:



The blue numbers inside the segment define the order of the segments, and also define the bit position when the object is accessed in raw format. Segments could be defined in the opposite order, but then the examples bellow would show mirror image.

Example 1: A bar object accessed raw

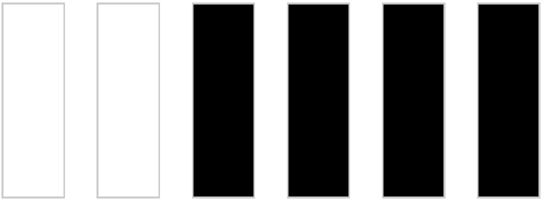
Writing binary pattern '010101' or decimal 21 with the Cmd_Obj_W command to a bar object, that looks like the one above, will create the following pattern:



More about this command is in Chapter "4.3.9. Cmd_Obj_W".

Example 2: A bar object accessed normally

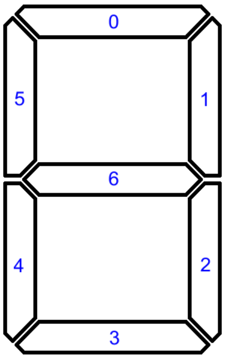
Writing the value of 4 with the Cmd_Bar command to a bar object, that looks like the one above, will create the following pattern:



More about this command is in Chapter "4.3.13. Cmd_Bar".

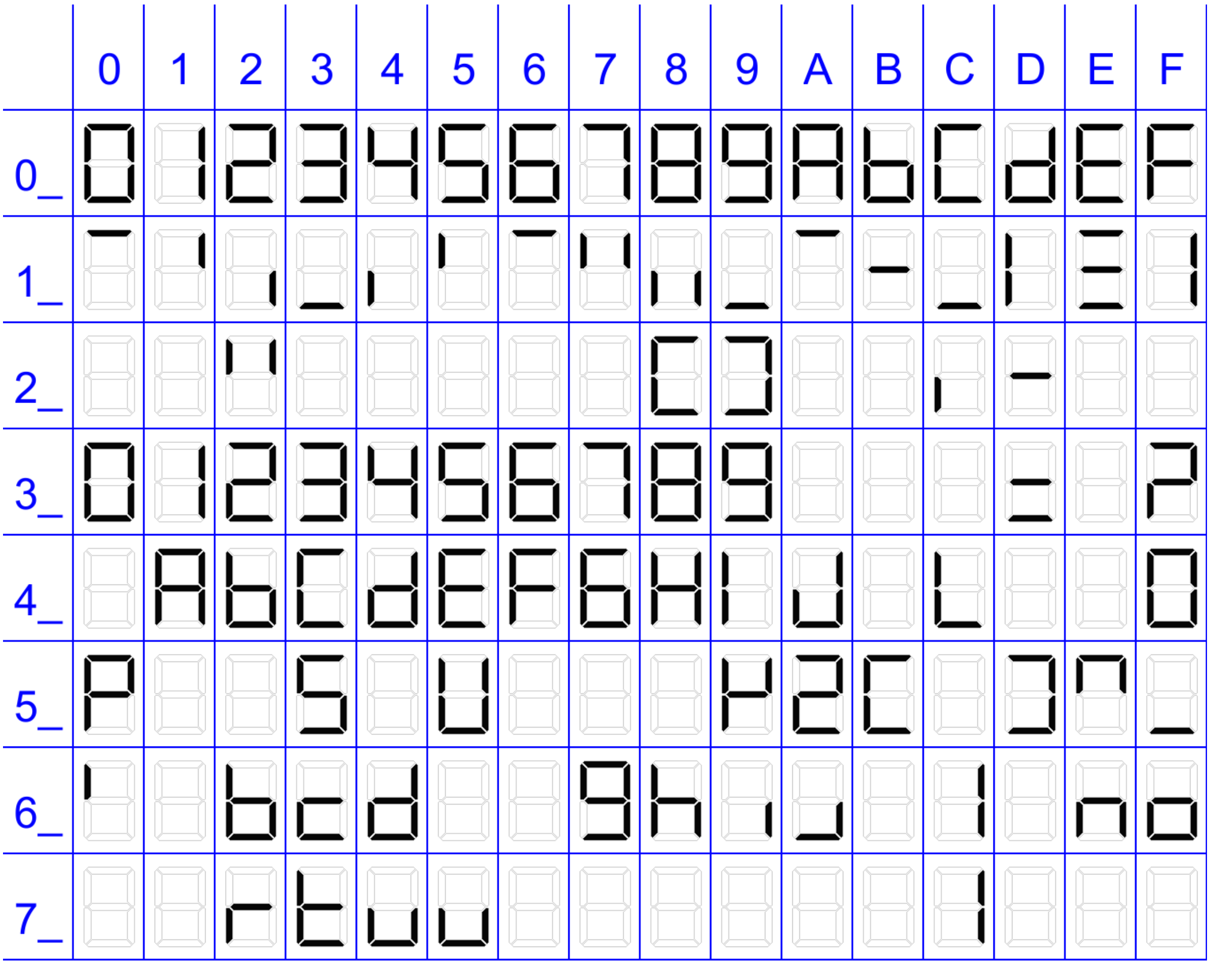
4.1.4. 7-segment object

A 7-segment object consists of 7 segments that are arranged in this general layout:



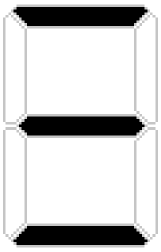
The blue numbers inside the segment define the order of the segment, and also defines the bit position when the object is accessed in raw format.

A 7-segment object is usually used to display one decimal number, but it is also capable of displaying a limited number of other characters. The full character table is here:


0
1
2
3
4
5
6
7
8
9
A
B
C
D
E
F
0_
1_
2_
3_
4_
5_
6_
7_

Example 1: A 7-segment object accessed raw

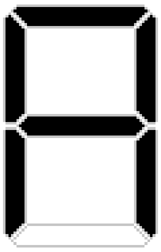
Writing binary pattern '01001001' or decimal 73 with the Cmd_Obj_W command to a 7-segment object will create the following pattern:



More about this command is in Chapter "4.3.9. Cmd_Obj_W".

Example 2: A 7-segment object accessed normally

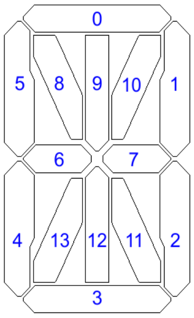
Writing the letter 'A' or decimal 65 with the Cmd_7seg command to a 7-segment object will create the following pattern:



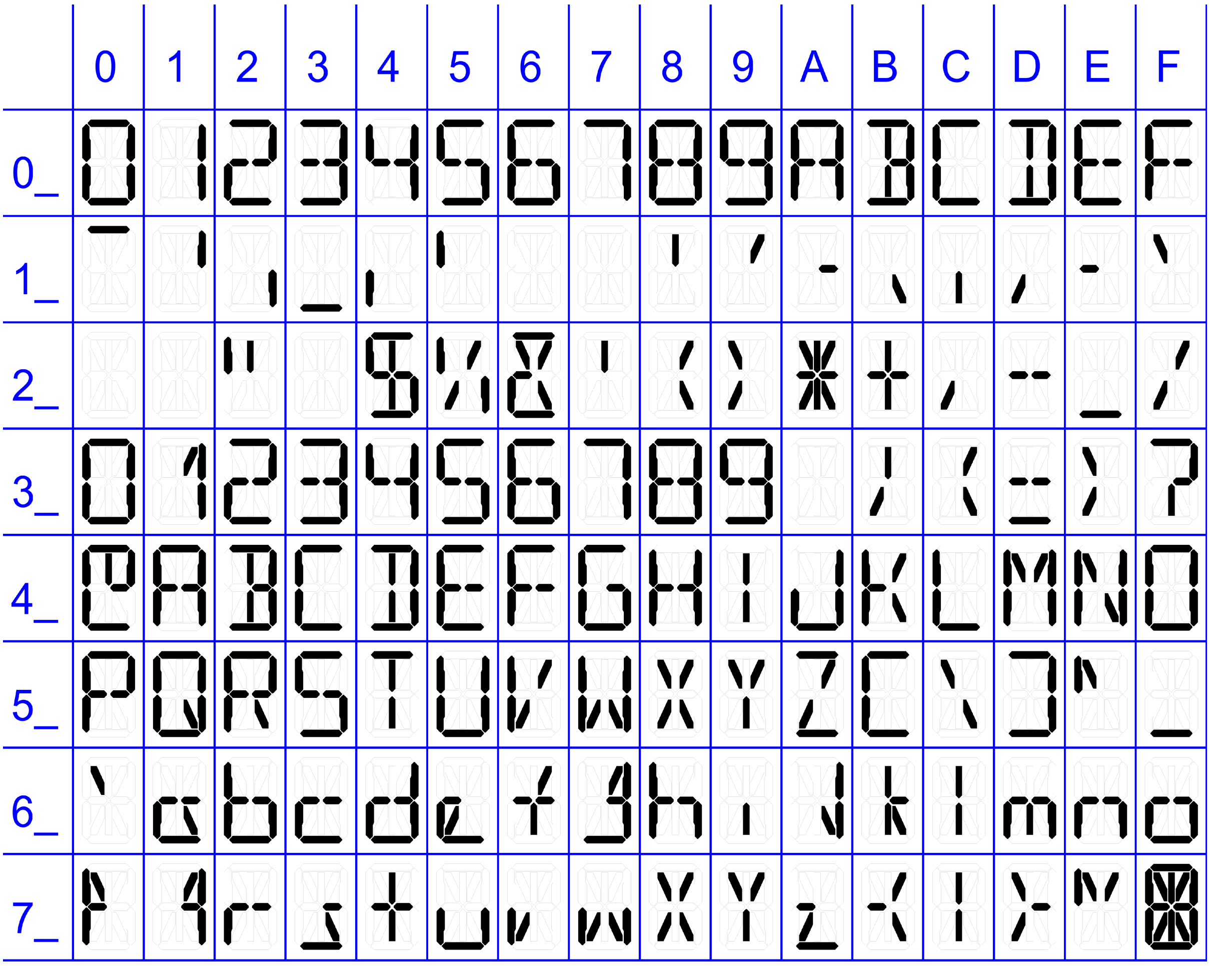
More about this command is in Chapter "4.3.10. Cmd_7seg".

4.1.5. 14-segment object

A 14-segment object consists of 14 segments that are arranged in this general layout:

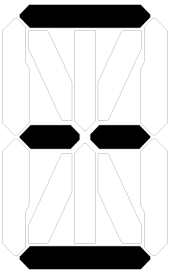


The blue numbers inside the segment define the order of the segment, and also defines the bit position when the object is accessed in raw format.
A 14-segment object is usually used to display one character. The full character table is here:



Example 1: A 14-segment object accessed raw

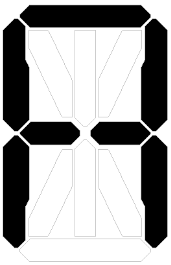
Writing binary pattern '00000011001001' or decimal 201 with the Cmd_Obj_W command to a 14-segment object will create the following pattern:



More about this command is in Chapter "4.3.9. Cmd_Obj_W".

Example 2: A 14-segment object accessed normally

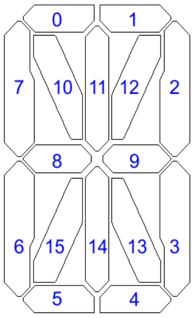
Writing the letter 'A' or decimal 65 with the Cmd_14seg command to a 14-segment object will create the following pattern:



More about this command is in Chapter "4.3.11. Cmd_14seg".

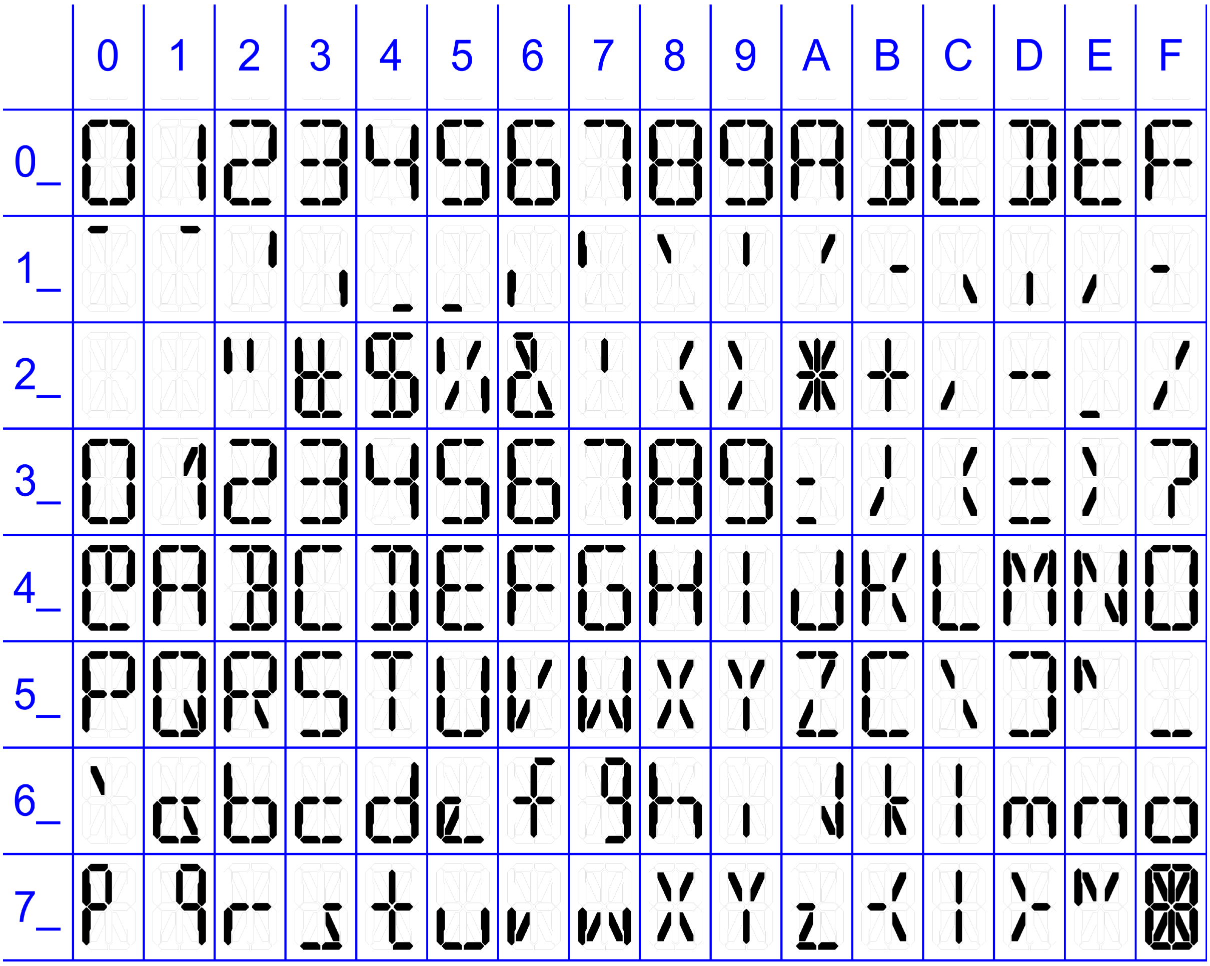
4.1.6. 16-segment object

A 16-segment object consists of 16 segments that are arranged in this general layout:



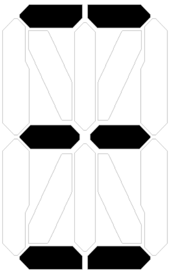
The blue numbers inside the segment define the order of the segment, and also defines the bit position when the object is accessed in raw format.

A 16-segment object is usually used to display one character. The full character table is here:



Example 1: A 16-segment object accessed raw

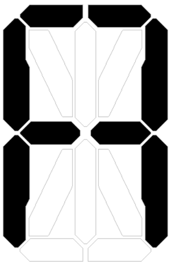
Writing binary pattern '0000001100110011' or decimal 819 with the Cmd_Obj_W command to a 16-segment object will create the following pattern:



More about this command is in Chapter "4.3.9. Cmd_Obj_W".

Example 2: A 16-segment object accessed normally

Writing the letter 'A' or decimal 65 with the Cmd_16seg command to a 16-segment object will create the following pattern:



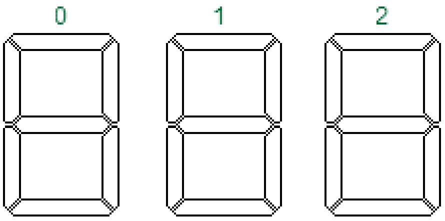
More about this command is in Chapter "4.3.12. Cmd_16seg".

4.2. Standard components

Any array of objects, placed consecutively in the objects table can be treated as a component, and KickStart™ simplifies the use of some common components.

4.2.1. String component

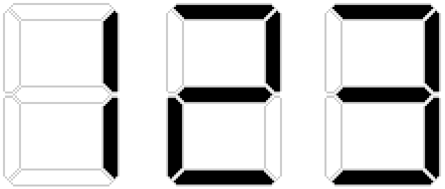
A group of 7-segment, 14-segment or 16-segment objects can be treated as a string component. A typical 7-segment string component, consisting of three 7-segment digits, is the following:



The green numbers inside the segment define the order of the components.

Example 1: A 7-segment string accessed normally

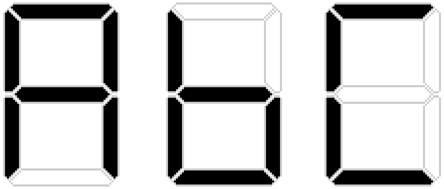
Writing the string "123", or decimal numbers 49, 50, 51 with the Cmd_7Seg_Str command to a 7-segment string object, that looks like the one above, will create the following pattern:



More about this command is in Chapter "4.3.14. Cmd_7seg_Str".

Example 1: A 7-segment string accessed normally

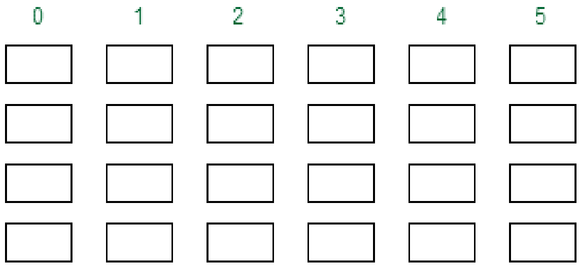
Writing the string "ABC", or decimal numbers 65, 66, 67 with the Cmd_7Seg_Str command to a 7-segment string object, that looks like the one above, will create the following pattern:



More about this command is in Chapter "4.3.14. Cmd_7seg_Str".

4.2.2. Bargraph component

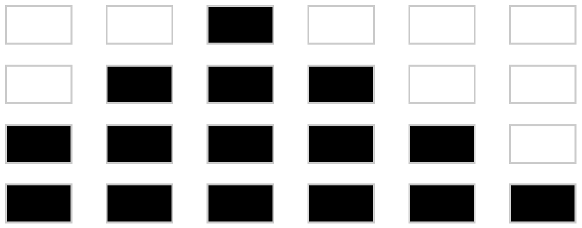
A group of bar objects treated as a string and it is call bargraph component. A typical bargraph component, consisting of six bars, is the following:



The green numbers inside the segment define the order of the components.

Example: A bargraph component accessed normally

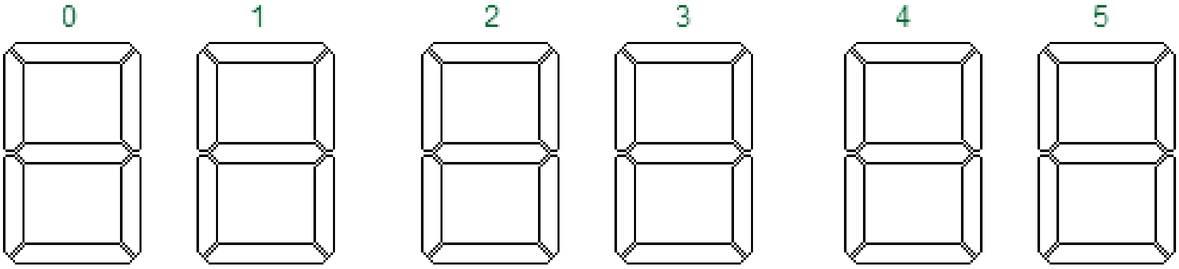
Writing the string of decimal numbers 2, 3, 4, 3, 2, 1 with the Cmd_Bar_Str command to a bargraph component, that looks like the one above, will create the following pattern:



More about this command is in Chapter "4.3.17. Cmd_Bar_Str".

4.2.3. Date component

A special group of six 7-segment object, that is used to automatically display the date in YYMMDD format is the date component. A typical date component is the following:

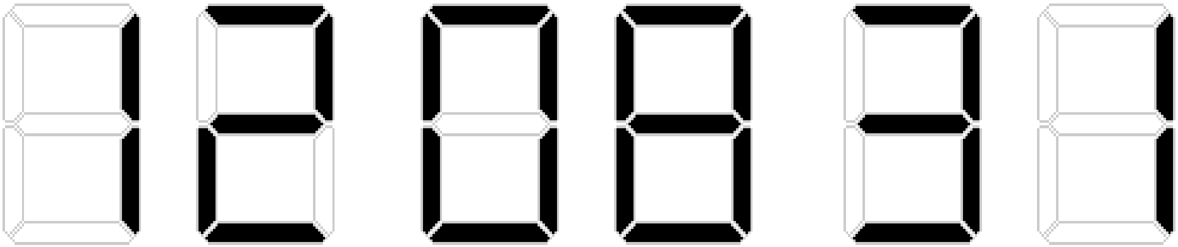


The green numbers inside the segment define the order of the objects in the component. If the order is different then the digits of the date appear in different order. It is possible to define more than one components that contain the same objects in different order to display the date in different, culture-specific order.

More information about displaying the date is in Chapter "4.3.22. Cmd_RTC".

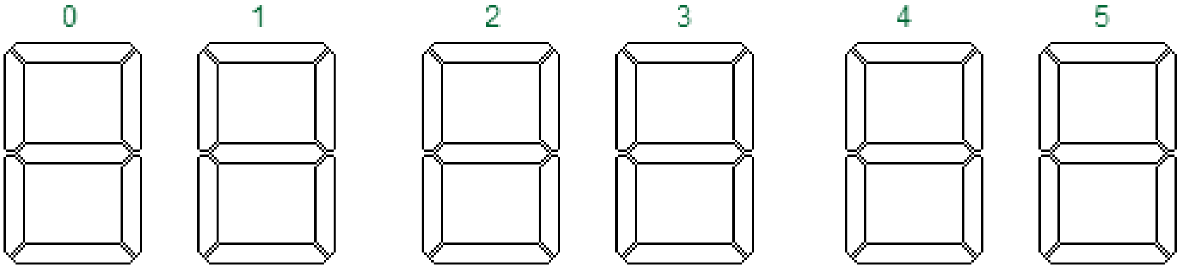
Example: A date component displaying the date

A date component displaying the date on August 31th, 2012 will look like this:



4.2.4. Time component

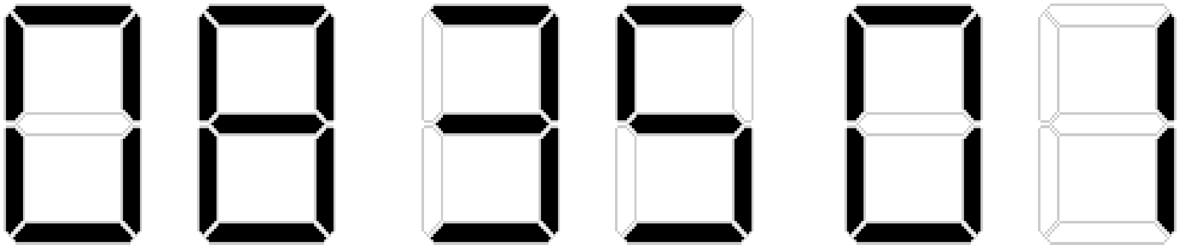
A special group of six 7-segment objects, that is used to automatically display the time in hhmmss format is the time component. A typical time component is the following:



The green numbers inside the segment define the order of the components.
More information about displaying the time is in Chapter "4.3.22. Cmd_RTC".

Example: A time component displaying the time

A time component displaying the time on 8:35:01 will look like this:



4.3. Commands

Commands differ in their mode of execution. There are three types of commands:

1. Status read command

   This is a very special command, and the only one that returns data from KickStart™ LCD controller. It is different for I²C and SPI interface.

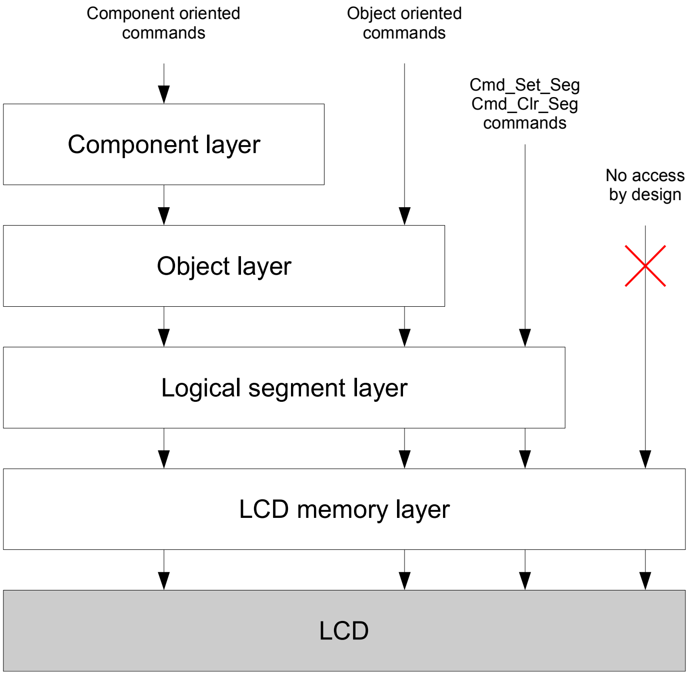
2. Configuration commands

   Configuration commands are always the first commands to send KickStart™ LCD controller. They define layout and connection of the LCD.

3. Normal commands

   All other commands cause some change in the way KickStart™ LCD controller displays information.

Normal commands also differ on which abstraction layer they act upon.



The lowest layer is the "LCD memory layer". KickStart™ LCD controller has 64 bytes LCD memory. Each bit in this memory corresponds to one segment. The memory to segment mapping is determined by the segment/common line connections to the LCD and it cannot be changed. This is the only layer most dumb controllers implement, and in KickStart™ it is not accessible directly. Because the memory map is directly determined by the physical connection of the LCD it may change between different revisions of the glass.

The next level is the "Logical segment layer". Each segment has a unique number between 0 and 509, and can be turned on by the Cmd_Set_Seg or cleared by Cmd_Clr_Seg commands. Each logical segment is mapped to one bit of LCD memory. Logical segments are usually nicely ordered and the order does not depend on the physical connection. The segment definition table of the KickStart™ development kit is in Chapter "3.1.2. Segment definition table". The "Logical segment layer" is accessible after downloading the LCD Segment Table.

The next level is the "Object layer". Commands that access this layer operate on objects, (which are group of segments.) The object definition table of the KickStart™ development kit is in Chapter "3.1.3. Object definition table". The "Object layer" is accessible after downloading the LCD Object Table.

The top level is the Component layer. Commands that access this layer operate on components, (which are array of objects.)

Normal commands have a general format of

command, length, payload[], checksum

where

Symbol	Length	Description
command	1 byte	Command code
length	1 byte	Length of payload[]
payload[]	length	Meaning depends on command code
checksum	1 byte	Sum of all bytes starting from command plus 128

List of commands:

Code	Symbol	Description	Chapter
1	Cmd_Seg_Tbl	Load LCD Segment Table	4.3.2.
2	Cmd_Obj_Tbl	Load LCD Object Table	4.3.3.
3	Cmd_ScrollTxt	Load text for automatic scrolling	4.3.20.
16	Cmd_LCD_cnf	Set commons of glass, bias, object count	4.3.4.
17	Cmd_Contrast	Set LCD contrast level	4.3.30.
18	Cmd_Statread	Read status bytes. Only for SPI mode!	4.3.1.
19	Cmd_Reset	Reset KickStart™ LCD Controller	4.3.6.
20	Cmd_Set_Seg	Set logical segment on	4.3.7.
21	Cmd_Clr_Seg	Set logical segment off	4.3.8.
22	Cmd_Obj_W	Write object without conversion	4.3.9.
23	Cmd_7seg	Set a 7-segment object to a value	4.3.10.
24	Cmd_14seg	Set a 14-segment object to a value	4.3.11.
25	Cmd_16seg	Set a 16-segment object to a value	4.3.12.
27	Cmd_Bar	Set a bar object to a value	4.3.13.
28	Cmd_7seg_Str	Set a 7-segment component to a value	4.3.14.
29	Cmd_14seg_Str	Set a 14-segment component to a value	4.3.15.
30	Cmd_16seg_Str	Set a 16-segment component to a value	4.3.16.
32	Cmd_Bar_Str	Set a bargraph component to a value	4.3.17.
33	Cmd_Shift_L	Shifts the values of a component left by one object	4.3.18.
34	Cmd_Shift_R	Shifts the values of a component right by one object	4.3.19.
35	Cmd_Scroll2Hz	Controls automatic scrolling	4.3.21.
36	Cmd_RTC	Set RTC mode, date/time, events, auto-display	4.3.22.
37	Cmd_Attr	Set attribute of a component	4.3.23.
38	Cmd_LED	Set one LED, including the backlight, on or off	4.3.25.
39	Cmd_Buzz	Turn on/off buzzer, set output-mode, frequency	4.3.26.
40	Cmd_LCD_OnOff	Turn the LCD on or off	4.3.27.
41	Cmd_TestLED	Turn on or off the test functions of the LEDs	4.3.24.
42	Cmd_AllSeg_On	Turn on all segments	4.3.28.
43	Cmd_AllSeg_Off	Turn off all segments	4.3.29.
44	Cmd_Sleep	Set KickStart™ power mode	4.3.31.

All other command codes are reserved.

4.3.1. Status read and Cmd_Statread

This is a special command, the only one which reads status and RTC (Real Time Clock) data from Kickstart, and it differs in SPI and I²C mode.

See Chapter "4.3.22. Cmd_RTC" about setting and displaying the RTC.

The Event pin is set if

- a key is pressed
- in every second, if RTC_On=1 and RTC_event_On=1

The Event pin is cleared by reading the status.

I²C mode:
write I²C Read Address, then read 9 status bytes and the checksum. The checksum is the sum of all status bytes, plus 128.

The format of the returned data is:

version, keys, year, month, day, dow, hour, min, sec, checksum

where

Symbol	Length	Description
version	1 byte	Firmware version
keys	1 byte	See separate table
year	1 byte	RTC years since 2000
month	1 byte	RTC month (1..12)
day	1 byte	RTC day (1..31)
dow	1 byte	RTC day of the week (1..7)
hour	1 byte	RTC hours (0..23)
min	1 byte	RTC minutes (0..59)
sec	1 byte	RTC seconds (0..59)
checksum	1 byte	Sum of all bytes, **plus 128**

Please note that the numbers are **not** BCD numbers.

If a key was pressed, (one KeyIn signal went low), since last status read, a bit in the keys field is set.

Bit	Key	KeyIn signal went low
0	1	KeyIn1
1	2	KeyIn2
2	3	KeyIn3
3	4	KeyIn4
4	5	KeyIn5

Only key press events are detectable, key release events are not.
There is no way to determine the length of the keypress.

SPI mode:
write Cmd_Statread, then read 9 status bytes and the checksum. The checksum is the sum of all status bytes, plus 18 (which is the command code of Cmd_Statread), plus 128.

The format of the command is:

Cmd_Statread, X, X, X, X, X, X, X, X, X, X

where X can be anything.

Or with numbers:

18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

The first status byte comes while the host transmitting the first 0.

The format of the returned data is:

0, version, keys, year, month, day, dow, hour, min, sec, checksum

where

Symbol	Length	Description
0	1 byte	Dummy
version	1 byte	Firmware version
keys	1 byte	See separate table above, at I²C mode
year	1 byte	RTC years since 2000
month	1 byte	RTC month (1..12)
day	1 byte	RTC day (1..31)
dow	1 byte	RTC day of the week (1..7)
hour	1 byte	RTC hours (0..23)
min	1 byte	RTC minutes (0..59)
sec	1 byte	RTC seconds (0..59)
checksum	1 byte	Sum of all bytes, **plus 146**

Please note that the numbers are **not** BCD numbers.

Example 1:

In SPI mode the host sends the following command

18, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

and reads back the following response:

0, 16, 0, 12, 8, 2, 5, 6, 4, 54, 253

which means
the first 0 is a dummy byte, that comes during the transmission of the command code of 18
the version number is 16
there was no keypress since last status read
the date is 2012-08-02, Thursday, and the time is 06:04:54
18+(16+0+12+8+2+5+6+4+54)+128=253, checksum is correct

Example 2:

In I²C mode the host selects read mode by sending the I²C read address of 203 and then reads 10 bytes back:

16, 0, 12, 8, 2, 5, 6, 4, 54, 235

which means
the version number is 16
there was no keypress since last status read
the date is 2012-08-02, Thursday, and the time is 06:04:54
(16+0+12+8+2+5+6+4+54)+128=235, checksum is correct

4.3.2. Cmd_Seg_Tbl

Load LCD Segment table. This command shall be used once after startup before any other commands which accesses segments or objects.

The format of this command is

table[]

where

Symbol	Length	Description
table[]	variable	The segment table as defined for this LCD

The first byte of the table is always the command code of Cmd_Seg_Tbl, which is 1. The rest depends on the LCD.

The segment table of KickStart™ development kit is in Chapter "3.1.4. LCD initialization data"

The Kickdev application initializes KickStart™ Development Board with the table selected from the "Select LCD" dropbox. The name of the default LCD is "GP07820B0-F011". It is possible to load a different table if a different LCD is connected to the KickStart™ Development Board and the tables copied to the "LCDFiles" directory.

Example:

See Chapter "4.3.5. LCD initialization example"

4.3.3. Cmd_Obj_Tbl

This command shall be used once after startup before any other commands which accesses segments or objects.

The format of this command is

table[]

where

Symbol	Length	Description
table[]	variable	The object table as defined for this LCD

The first byte of the table is always the command code of Cmd_Obj_Tbl, which is 2. The rest depends on the LCD.

The object table of KickStart™ development kit is in Chapter "3.1.4. LCD initialization data"

The Kickdev application initializes KickStart™ Development Board with the table selected from the "Select LCD" dropbox. The name of the default LCD is "GP07820B0-F011". It is possible to load a different table if a different LCD is connected to the KickStart™ Development Board and the tables copied to the "LCDFiles" directory.

Example:

See Chapter "4.3.5. LCD initialization example"

4.3.4. Cmd_LCD_cnf

Set LCD configuration. This standard command is the last of the initialization commands.
The format of this command is

Cmd_LCD_cnf, length, duty, bias, objcount, checksum

where

Symbol	Length	Description
Cmd_LCD_cnf	1 byte	Command code (always 16)
length	1 byte	Length of the parameters (always 3)
duty	1 byte	0=1/4, 1=1/5, 2=1/6, 3=1/8
bias	1 byte	1=1/4, 2=1/5
objcount	1 byte	Number of objects in object table
checksum	1 byte	Sum of all bytes, plus 128

There are recommended values for these parameters which are provided for each LCD by the manufacturer. The use of other values are not recommended.

The recommended values for KickStart™ development kit is in Chapter "3.1.4. LCD initialization data"

Example:

See Chapter "4.3.5. LCD initialization example"

4.3.5. LCD initialization example

After startup the Kickdev program initializes the LCD controller through the KickStart™ Development Board with the following commands:

1, 119, 219, 218, 213, 209, 207, ...

2, 66, 66, 117, 92, 11, 6, 37, 122, ...

16, 3, 3, 1, 34, 185

The first is the Cmd_Seg_Tbl, the second is the Cmd_Obj_Tbl and the third is the Cmd_LCD_cnf, exactly as defined for the default LCD.

4.3.6. Cmd_Reset

Reset the LCD controller.
The format of this command is

Cmd_Reset, length, checksum

where

Symbol	Length	Description
Cmd_Reset	1 byte	Command code (always 19)
length	1 byte	Always 0
checksum	1 byte	Sum of all bytes, plus 128 (always 147)

After sending this command the host must wait for 100ms before sending the next command.

Example:

The host send the following command:

19, 0, 147

and the KickStart™ LCD controller is reset.

4.3.7. Cmd_Set_Seg

Set one segment.
The format of this command is

Cmd_Set_Seg, length, segment, checksum

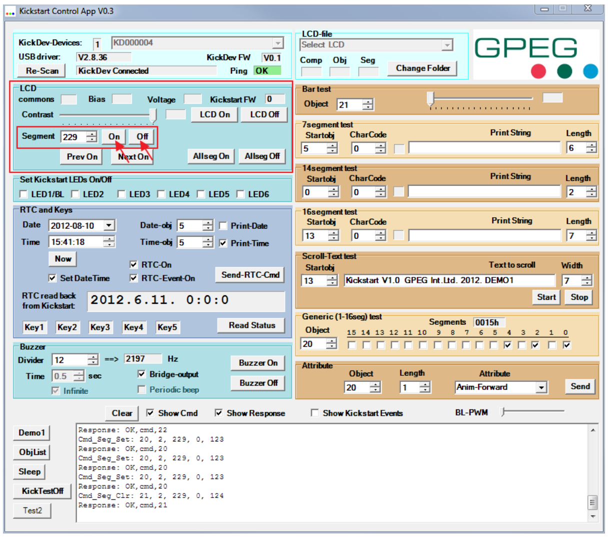
where

Symbol	Length	Description
Cmd_Set_Seg	1 byte	Command code (always 20)
length	1 byte	Always 2
segment	2 bytes	Segment number (LSB first)
checksum	1 byte	Sum of all bytes, plus 128

Segment numbers for the default LCD are in the table in Chapter "3.1.2. Segment definition table". For other LCDs, they are provided with the LCD.

Example:

The user selects segment 229 in "LCD" frame of KickDevApp and then clicks the "On" button.



Kickstart Control App V0.3

KickDev-Devices: 1 KD000004
USB driver: V2.8.36 KickDev FW V0.1
KickDev Connected Ping OK

LCD
Kickstart FW 0
Segment 229

Set Kickstart LEDs On/Off
□ LED1/BL □ LED2 □ LED3 □ LED4 □ LED5 □ LED6

RTC and Keys
Date 2012-08-10 Date-obj 5 □ Print-Date
Time 15:41:18 Time-obj 5 ☑ Print-Time
☑ Set DateTime ☑ RTC-On ☑ RTC-Event-On
RTC read back from Kickstart 2012.6.11. 0:0:0

Buzzer
Divider 12 ==> 2197 Hz
Time 0.5 sec ☑ Bridge-output
☑ Infinite □ Periodic beep

LCD-file
Select LCD

Bar test
Object 21

7segment test
Startobj 5 CharCode 0 Length 6

14segment test
Startobj 0 CharCode 0 Length 2

16segment test
Startobj 13 CharCode 0 Length 7

Scroll-Text test
Startobj 13 Text to scroll Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1 Width 7

Generic (1-16seg) test
Object 20 Segments 0015h

Attribute
Object 20 Length 1 Attribute Anim-Forward

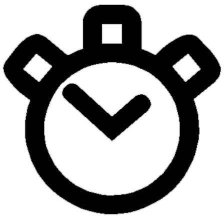
☑ Show Cmd ☑ Show Response □ Show Kickstart Events

```
Response: OK,cmd,22
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Clr: 21, 2, 229, 0, 124
Response: OK,cmd,21
```

And the host sends the following commands:

20, 2, 229, 0, 123

and on the KickStart™ LCD the alarm clock icon, which is segment 229, lights up.



4.3.8. Cmd_Clr_Seg

Clear one segment.
The format of this command is

Cmd_Clr_Seg, length, segment, checksum

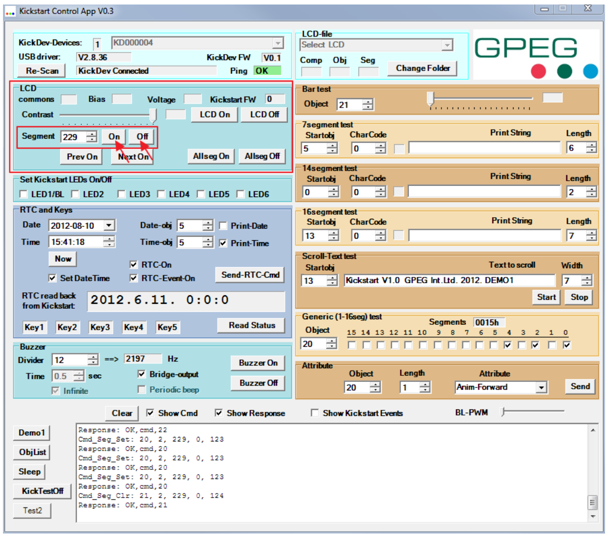
where

Symbol	Length	Description
Cmd_Clr_Seg	1 byte	Command code (always 21)
length	1 byte	Always 2
segment	2 bytes	Segment number (LSB first)
checksum	1 byte	Sum of all bytes, plus 128

Segment numbers for the default LCD are in the table in Chapter "3.1.2. Segment definition table". For other LCDs, they are provided with the LCD.

Example:

The user selects segment 229 in "LCD" frame of KickDevApp and then clicks the "Off" button.



Kickstart Control App V0.3

KickDev-Devices: 1 KD000004
USB driver: V2.8.36
KickDev FW V0.1
KickDev Connected
Ping OK

LCD
Kickstart FW 0
Segment 229

Set Kickstart LEDs On/Off
☐ LED1/BL ☐ LED2 ☐ LED3 ☐ LED4 ☐ LED5 ☐ LED6

RTC and Keys
Date 2012-08-10
Time 15:41:18
Date-obj 5 ☐ Print-Date
Time-obj 5 ☑ Print-Time
☑ RTC-On
☑ Set DateTime ☑ RTC-Event-On
RTC read back from Kickstart: 2012.6.11. 0:0:0

Buzzer
Divider 12 ==> 2197 Hz
Time 0.5 sec
☑ Infinite
☑ Bridge-output
☐ Periodic beep

LCD-file
Select LCD

Bar test
Object 21

7segment test
Startobj 5 CharCode 0 Length 6

14segment test
Startobj 0 CharCode 0 Length 2

16segment test
Startobj 13 CharCode 0 Length 7

Scroll-Text test
Startobj 13 Text to scroll Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1 Width 7

Generic (1-16seg) test
Object 20 Segments 0015h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☑	☐	☑	☐	☑

Attribute
Object 20 Length 1 Attribute Anim-Forward

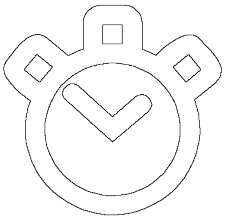
☑ Show Cmd ☑ Show Response ☐ Show Kickstart Events

```
Response: OK,cmd,22
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Set: 20, 2, 229, 0, 123
Response: OK,cmd,20
Cmd_Seg_Clr: 21, 2, 229, 0, 124
Response: OK,cmd,21
```

And the host sends the following commands:

21, 2, 229, 0, 124

and on the KickStart™ LCD the alarm clock icon, which is segment 229, turns off.



4.3.9. Cmd_Obj_W

Write the value of one object raw

The format of this command is

Cmd_Obj_W, length, object, value, checksum

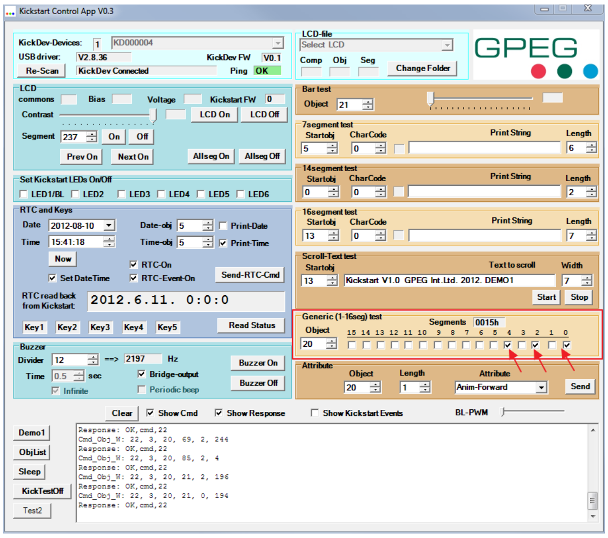
where

Symbol	Length	Description
Cmd_Obj_W	1 byte	Command code (always 22)
length	1 byte	Always 3
object	1 byte	Object number
value	2 bytes	New value (LSB first)
checksum	1 byte	Sum of all bytes, plus 128

Object numbers of the default LCD are in the table in Chapter "3.1.3. Object definition table". For other LCDs, they are provided with the LCD.

Example:

The user selects object 20 in "Generic (1-16seg) test" frame of KickDevApp and then sets "0000000000010101" pattern by toggling the checkboxes:

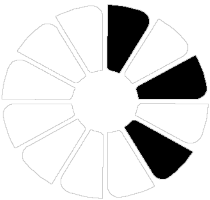


```
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 69, 2, 244
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 85, 2, 4
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 21, 2, 196
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 21, 0, 194
Response: OK,cmd,22
```

And the host sends the following commands:

22, 3, 20, 21, 0, 194

and on the KickStart™ LCD the disk shape, which is object 20, shows the following pattern:



4.3.10. Cmd_7seg

Write the value of one 7-segment object with conversion.

The format of this command is

Cmd_7seg, length, object, value, checksum

where

Symbol	Length	Description
Cmd_7seg	1 byte	Command code (always 23)
length	1 byte	Always 2
object	1 byte	Object number
value	1 byte	New value
checksum	1 byte	Sum of all bytes, plus 128

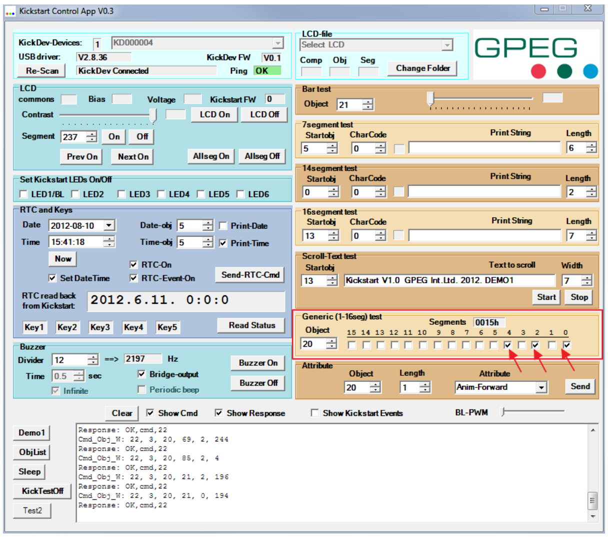
Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".

For other LCDs, they are provided with the LCD.

The character generator is defined in Chapter "4.1.4. 7-segment object".

Example:

The user selects object 5 in "7segment test" frame of KickDevApp and then sets CharCode to 65, which is letter 'A' and presses Enter.

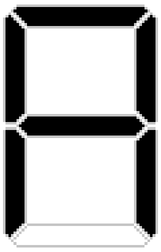


```
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 69, 2, 244
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 85, 2, 4
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 21, 2, 196
Response: OK,cmd,22
Cmd_Obj_W: 22, 3, 20, 21, 0, 194
Response: OK,cmd,22
```

And the host sends the following command:

23, 2, 5, 65, 223

and on the KickStart™ first big 7-segment digit, which is object 5, appears the following:



4.3.11. Cmd_14seg

Write the value of one 14-segment object with conversion.
The format of this command is

Cmd_14seg, length, object, value, checksum

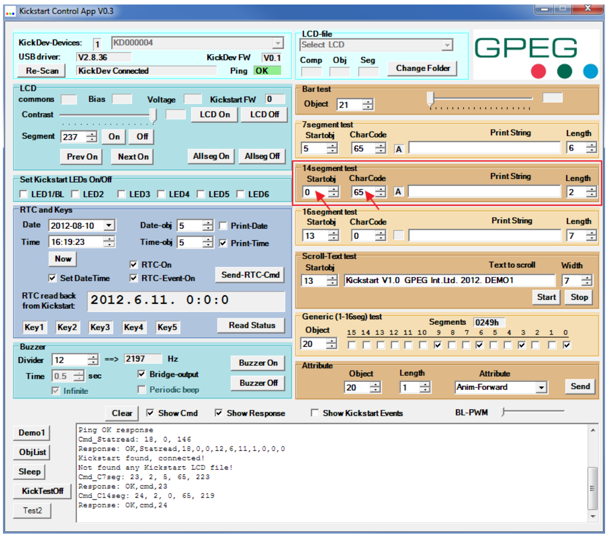
where

Symbol	Length	Description
Cmd_14seg	1 byte	Command code (always 24)
length	1 byte	Always 2
object	1 byte	Object number
value	1 byte	New value
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
The character generator is defined in Chapter "4.1.5. 14-segment object".

Example:

The user selects object 0 in "14segment test" frame of KickDevApp and then sets CharCode to 65, which is letter 'A' and presses Enter.



14segment test: Startobj 0, CharCode 65, A, Length 2

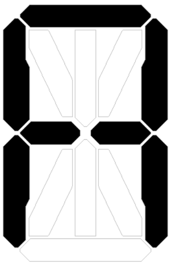
☑ Show Cmd ☑ Show Response ☐ Show Kickstart Events

```
Ping OK response
Cmd_Statread: 18, 0, 146
Response: OK,Statread,18,0,0,12,6,11,1,0,0,0
Kickstart found, connected!
Not found any Kickstart LCD file!
Cmd_C7seg: 23, 2, 5, 65, 223
Response: OK,cmd,23
Cmd_C14seg: 24, 2, 0, 65, 219
Response: OK,cmd,24
```

And the host sends the following command:

24, 2, 0, 65, 219

and on the KickStart™ first 14-segment digit, which is object 0, the following image appears:



4.3.12. Cmd_16seg

Write the value of one 16-segment object with conversion.
The format of this command is

Cmd_16seg, length, object, value, checksum

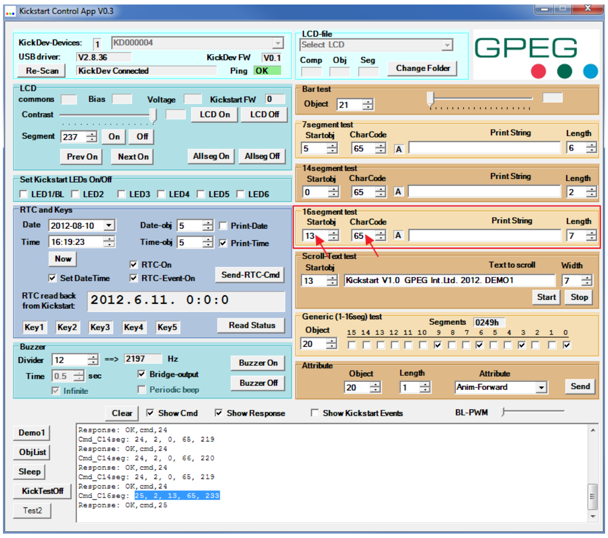
where

Symbol	Length	Description
Cmd_16seg	1 byte	Command code (always 25)
length	1 byte	Always 2
object	1 byte	Object number
value	1 byte	New value
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
The character generator is defined in Chapter "4.1.6. 16-segment object".

Example:

The user selects object 13 in "16segment test" frame of KickDevApp and then sets CharCode to 65, which is letter 'A' and presses Enter.

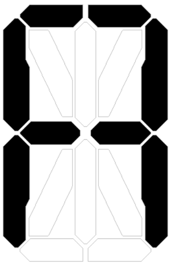


```
Response: OK,cmd,24
Cmd_C14seg: 24, 2, 0, 65, 219
Response: OK,cmd,24
Cmd_C14seg: 24, 2, 0, 66, 220
Response: OK,cmd,24
Cmd_C14seg: 24, 2, 0, 65, 219
Response: OK,cmd,24
Cmd_C16seg: 25, 2, 13, 65, 233
Response: OK,cmd,25
```

And the host sends the following command:

25, 2, 13, 65, 233

and on the KickStart™ first 16-segment digit, which is object 13, following image appears:



4.3.13. Cmd_Bar

Write the value of one Bar object with conversion.
The format of this command is

Cmd_Bar, length, object, value, checksum

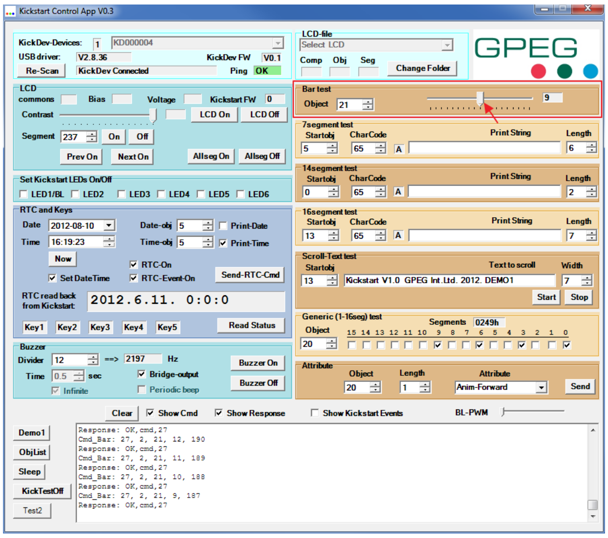
where

Symbol	Length	Description
Cmd_Bar	1 byte	Command code (always 27)
length	1 byte	Always 2
object	1 byte	Object number
value	1 byte	New value
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
value can be 0-16 for normal bar images and 17-18 for special effects.
More about the Bar object is in Chapter "4.1.3. Bar object".

Example:

The user selects object 21 in "Bar test" frame of KickDevApp and then pulls the slider to middle position.


Kickstart Control App V0.3

KickDev-Devices: 1 KD000004
USB driver: V2.8.36
KickDev Connected
KickDev FW V0.1
Ping OK

LCD-file: Select LCD

LCD
Kickstart FW 0
Segment 237

Set Kickstart LEDs On/Off
□ LED1/BL □ LED2 □ LED3 □ LED4 □ LED5 □ LED6

RTC and Keys
Date 2012-08-10
Time 16:19:23
Date-obj 5 □ Print-Date
Time-obj 5 ☑ Print-Time
☑ RTC-On
☑ Set DateTime ☑ RTC-Event-On
RTC read back from Kickstart: 2012.6.11. 0:0:0

Buzzer
Divider 12 ==> 2197 Hz
Time 0.5 sec ☑ Bridge-output
☑ Infinite □ Periodic beep

Bar test
Object 21 9

7segment test
Startobj 5 CharCode 65 A Print String Length 6

14segment test
Startobj 0 CharCode 65 A Print String Length 2

16segment test
Startobj 13 CharCode 65 A Print String Length 7

Scroll-Text test
Startobj 13 Text to scroll: Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1 Width 7

Generic (1-16seg) test
Object 20 Segments 0249h
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
□ □ □ □ □ □ ☑ □ □ ☑ □ □ ☑ □ □ ☑

Attribute
Object 20 Length 1 Attribute Anim-Forward

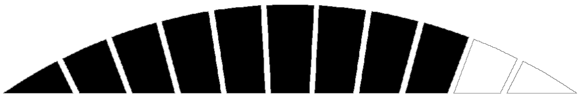
☑ Show Cmd ☑ Show Response □ Show Kickstart Events

```
Response: OK,cmd,27
Cmd_Bar: 27, 2, 21, 12, 190
Response: OK,cmd,27
Cmd_Bar: 27, 2, 21, 11, 189
Response: OK,cmd,27
Cmd_Bar: 27, 2, 21, 10, 188
Response: OK,cmd,27
Cmd_Bar: 27, 2, 21, 9, 187
Response: OK,cmd,27
```

And the host sends the following command:

27, 2, 21, 9, 187

and on the bottom of the LCD, which is object 21, the following image appears:



4.3.14. Cmd_7seg_Str

Write a string of values to a continuous array of 7-segment objects, with conversion.
The format of this command is

Cmd_7seg_Str, length, object, value[], checksum

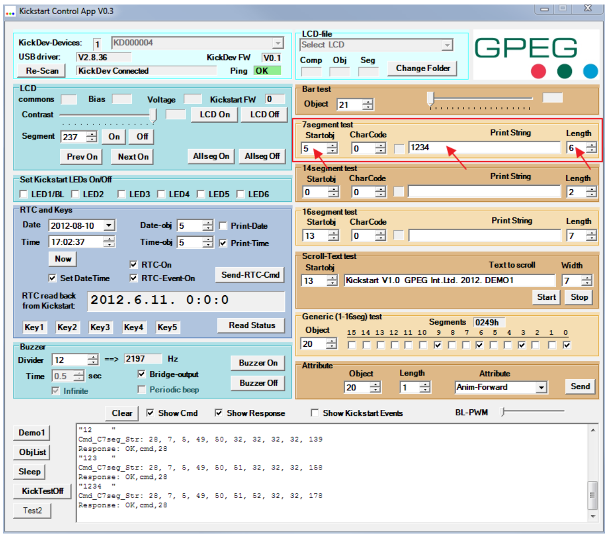
where

Symbol	Length	Description
Cmd_7seg	1 byte	Command code (always 28)
length	1 byte	Number of objects plus one
object	1 byte	Object number of the first object
value[]	length-1 bytes	New values
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
The character generator is defined in Chapter "4.1.4. 7-segment object".
More about strings is in Chapter "4.2.1. String component".

Example:

The user selects object 5 and length 6 in "7segment test" frame of KickDevApp and then types the string "1234 " into the edit box.



7segment test: Startobj 5, CharCode 0, Print String 1234, Length 6

```
"12    "
Cmd_C7seg_Str: 28, 7, 5, 49, 50, 32, 32, 32, 32, 139
Response: OK,cmd,28
"123   "
Cmd_C7seg_Str: 28, 7, 5, 49, 50, 51, 32, 32, 32, 158
Response: OK,cmd,28
"1234  "
Cmd_C7seg_Str: 28, 7, 5, 49, 50, 51, 52, 32, 32, 178
Response: OK,cmd,28
```

And the host sends the following command:

28, 7, 5, 49, 50, 51, 52, 32, 32, 178

and on the KickStart™ first 4 big 7-segment digits, which are objects 5, 6, 7, 8, appears the string of "1234" and the next two small 7-segment digits, which are objects 9 and 10, are cleared.

4.3.15. Cmd_14seg_Str

Write a string of values to a continuous array of 14-segment objects, with conversion.
The format of this command is

Cmd_14seg_Str, length, object, value[], checksum

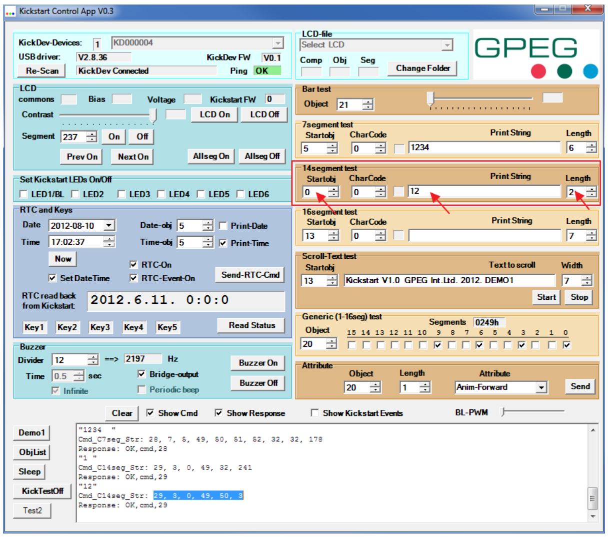
where

Symbol	Length	Description
Cmd_14seg	1 byte	Command code (always 29)
length	1 byte	Number of objects plus one
object	1 byte	Object number of the first object
value[]	length-1 bytes	New values
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
The character generator is defined in Chapter "4.1.5. 14-segment object".
More about strings is in Chapter "4.2.1. String component".

Example:

The user selects object 0 and length 2 in "14segment test" frame of KickDevApp and then types the string "12" into the edit box.



```
"1234  "
Cmd_C7seg_Str: 28, 7, 5, 49, 50, 51, 52, 32, 32, 178
Response: OK,cmd,28
"1 "
Cmd_C14seg_Str: 29, 3, 0, 49, 32, 241
Response: OK,cmd,29
"12"
Cmd_C14seg_Str: 29, 3, 0, 49, 50, 3
Response: OK,cmd,29
```

And the host sends the following command:

29, 3, 0, 49, 50, 3

and on the LCD on the first two 14-segment digits, which are objects 0 and 1, appears the string of "12".

4.3.16. Cmd_16seg_Str

Write a string of values to a continuous array of 16-segment objects, with conversion.
The format of this command is

Cmd_16seg_Str, length, object, value[], checksum

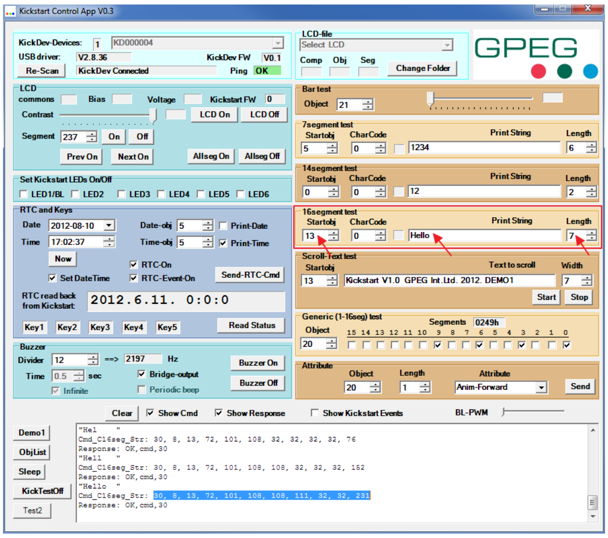
where

Symbol	Length	Description
Cmd_16seg	1 byte	Command code (always 30)
length	1 byte	Number of objects plus one
object	1 byte	Object number of the first object
value[]	length-1 bytes	New values
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.
The character generator is defined in Chapter "4.1.6. 16-segment object".
More about strings is in Chapter "4.2.1. String component".

Example:

The user selects object 13 and length 7 in "16segment test" frame of KickDevApp and then types the string "Hello " into the edit box.



```
"Hel    "
Cmd_C16seg_Str: 30, 8, 13, 72, 101, 108, 32, 32, 32, 32, 76
Response: OK,cmd,30
"Hell   "
Cmd_C16seg_Str: 30, 8, 13, 72, 101, 108, 108, 32, 32, 32, 152
Response: OK,cmd,30
"Hello  "
Cmd_C16seg_Str: 30, 8, 13, 72, 101, 108, 108, 111, 32, 32, 231
Response: OK,cmd,30
```

And the host sends the following command:

30, 8, 13, 72, 101, 108, 108, 111, 32, 32, 231

and on the LCD in the line of 16-segment digits, which are objects 13, 14, 15, 16, 17, 18 and 19, appears the string of "Hello ".

4.3.17. Cmd_Bar_Str

Write a string of values to a continuous array of Bar objects, which is also called Bar component, with conversion.

The format of this command is

Cmd_Bar_Str, length, object, value[], checksum

where

Symbol	Length	Description
Cmd_16seg	1 byte	Command code (always 32)
length	1 byte	Number of objects plus one
object	1 byte	Object number of the first object
value[]	length-1 bytes	New values
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".

For other LCDs, they are provided with the LCD.

More about the Bar object is in Chapter "4.1.3. Bar object" and about the Bargraph component is in Chapter "4.2.2. Bargraph component".

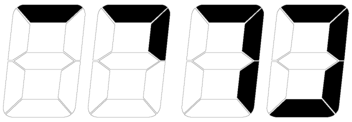
Example:

This command is intended for updating a bargraph component. Unfortunately there is no such component on the default glass, so this example is a bit awkward

The host sends the following command:

32, 5, 5, 1, 2, 3, 4, 180

and on the LCD in the line of big 7-segment digits, which are objects 5, 6, 7 and 8, the following image appears:



This example shows that a 7-segment component can be treated as Bargraph component, although it is not very useful.

4.3.18. Cmd_Shift_L

Shift the values of an array of objects left by one. The rightmost object is cleared.
The format of this command is

Cmd_Shift_L, length, object, size, checksum

where

Symbol	Length	Description
Cmd_Shift_L	1 byte	Command code (always 33)
length	1 byte	Always 2
object	1 byte	Object number of the first object
size	1 byte	Number of objects
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.

Example:

Assume there is string of "1234" on the big 7-segments of the LCD. This can happen after executing the example in Chapter "4.3.14. Cmd_7seg_Str".
Then the host sends the following command:

33, 2, 5, 4, 172

and on the LCD the string changes to "234 ".

4.3.19. Cmd_Shift_R

Shift the values of an array of objects right by one. The leftmost object is cleared.
The format of this command is

Cmd_Shift_R, length, object, size, checksum

where

Symbol	Length	Description
Cmd_Shift_R	1 byte	Command code (always 34)
length	1 byte	Always 2
object	1 byte	Object number of the first object
size	1 byte	Number of objects
checksum	1 byte	Sum of all bytes, plus 128

Object numbers for the default LCD are in the table in Chapter "3.1.3. Object definition table".
For other LCDs, they are provided with the LCD.

Example:

Assume there is string of "1234" on the big 7-segments of the LCD. This can happen after executing the example in Chapter "4.3.14. Cmd_7seg_Str".
Then the host sends the following command:

34, 2, 5, 4, 173

and on the LCD the string changes to " 123".

4.3.20. Cmd_ScrollTxt

Load text for later use by autoscroll commands. The text can be 255 character long.
The format of this command is

Cmd_ScrollTxt, length, text[], checksum

where

Symbol	Length	Description
Cmd_ScrollTxt	1 byte	Command code (always 3)
length	1 byte	Length of text
text[]	length	Text
checksum	1 byte	Sum of all bytes, plus 128

This command does not start scrolling, it just defines the text.
See in Chapter "4.3.21. Cmd_Scroll2Hz" how to use this text.

Example:

See example in Chapter "4.3.21. Cmd_Scroll2Hz".

4.3.21. Cmd_Scroll2Hz

Automatically scroll text in a component at the rate of 2 characters per second. Text is loaded with Cmd_ScrollTxt. See Chapter "4.3.20. Cmd_ScrollTxt". The component can be 7-segment, 14-segment or 14-segment component, type is automatically detected.

After receiving this command the contents of the component is cleared and then characters shift in from right, one by one. After all characters are shifted in the first character comes again and it continues endlessly. If the component is shorter than the text, only a part of text will be shown at a time. If the component is longer than the text, the text appears multiple times.

The format of this command is

Cmd_Scroll2Hz, length, object, size, textsize, checksum

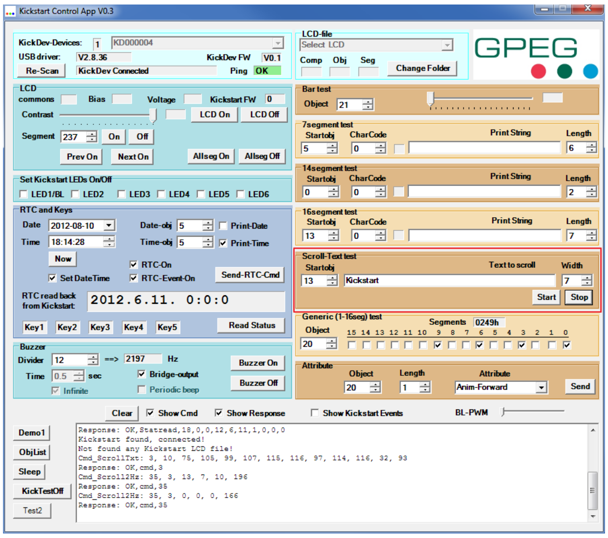
where

Symbol	Length	Description
Cmd_ScrollTxt	1 byte	Command code (always 3)
length	1 byte	Length of text
object	1 byte	Object number of the first object
size	1 byte	Number of objects
textsize	1 byte	Number of bytes to use from text
checksum	1 byte	Sum of all bytes, plus 128

To stop scrolling, send this command with size=0.

Example:

The user selects object 13 and length 7 in "Scroll-Text test" frame of KickDevApp and then types the string "Kickstart " into the edit box and then click on "Start" button.



```
Response: OK,Statread,18,0,0,12,6,11,1,0,0,0
Kickstart found, connected!
Not found any Kickstart LCD file!
Cmd_ScrollTxt: 3, 10, 75, 105, 99, 107, 115, 116, 97, 114, 116, 32, 93
Response: OK,cmd,3
Cmd_Scroll2Hz: 35, 3, 13, 7, 10, 196
Response: OK,cmd,35
Cmd_Scroll2Hz: 35, 3, 0, 0, 0, 166
Response: OK,cmd,35
```

And the host sends the following commands:

3, 10, 75, 105, 99, 107, 115, 116, 97, 114, 116, 32, 93

35, 3, 13, 7, 10, 196

and on the LCD in the line of 16-segment digits, which are objects 13, 14, 15, 16, 17, 18 and 19, appears the string of "Kickstart " scrolling in.

4.3.22. Cmd_RTC

Set RTC (Real Time Clock) date and time, events, and controls automatic display.
The format of this command is

Cmd_RTC, length, config, year, month, day, dow, hour, min, sec, dobj, tobj, checksum

where

Symbol	Length	Description
Cmd_ScrollTxt	1 byte	Command code (always 36)
length	1 byte	Always 10
config	1 byte	Configuration, see separate table
year	1 byte	RTC years since 2000
month	1 byte	RTC month (1..12)
day	1 byte	RTC day (1..31)
dow	1 byte	RTC day of the week (1..7)
hour	1 byte	RTC hours (0..23)
min	1 byte	RTC minutes (0..59)
sec	1 byte	RTC seconds (0..59)
dobj	1 byte	Object number of the first object that displays the date
tobj	1 byte	Object number of the first object that displays the time
checksum	1 byte	Sum of all bytes, plus 128

Please note that the numbers are **not** BCD numbers.
If year=0 then the date and time are **not** set.
The meaning of configuration bits are as follows:

Bit	Name	Description
7	RTC_On	1=RTC on, 0=RTC off
6	RTC_event_On	1=Set Event pin in every second The Event pin remains set till the host reads status
5	RTC_print_date	1=Automatically display date on six objects, starting from dobj in YYMMDD format
4	RTC_print_time	1=Automatically display time on six objects, starting from tobj in hhmmss format
3-0	Reserved	Reserved, always 0

dobj and tobj can be the first object of an array of six 7-segment or 14-segments objects. Other combinations are not allowed. dobj is used only if RTC_print_date=1. tobj is used only if RTC_print_time=1.

If RTC_print_time=1, the time is displayed in six objects in the following order when the command is received, and then only those objects are updated which value changes at every second.

Object number	Symbol	Description
tobj+0	h	Hours tens
tobj+1	h	Hours ones
tobj+2	m	Minutes tens
tobj+3	m	Minutes ones
tobj+4	s	Seconds tens
tobj+5	s	Seconds ones

Leading zeros are not suppressed

Additional symbols, like the colons between the digits, are not handled.

If RTC_print_date=1, the date is displayed in six objects in the following order when the command is received, and then only those objects are updated which value changes at every second.

Object number	Symbol	Description
dobj+0	Y	Year tens
dobj+1	Y	Year ones ones
dobj+2	M	Month tens
dobj+3	M	Month ones
dobj+4	D	Day tens
dobj+5	D	Day ones

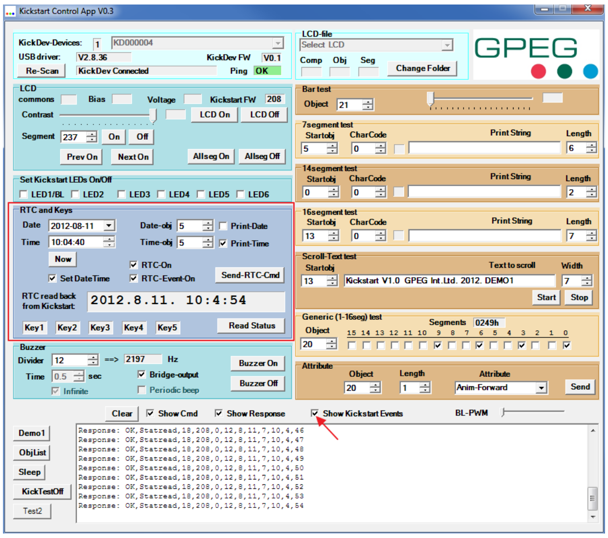
Leading zeros are not suppressed

Additional symbols, like separators between the digits, are not handled.

See Chapter "4.3.1. Status read and Cmd_Statread" about reading back the RTC and clearing the Event pin.

Example:

The user sets the selects object 13 and length 7 in "Scroll-Text test" frame of KickDevApp and then types the string "Kickstart " into the edit box and then click on "Start" button.



KickDev-Devices: 1 KD000004
USB driver: V2.8.36 KickDev FW V0.1
KickDev Connected Ping OK

LCD
commons Bias Voltage Kickstart FW 208
Contrast
Segment 237

Set Kickstart LEDs On/Off
□ LED1/BL □ LED2 □ LED3 □ LED4 □ LED5 □ LED6

RTC and Keys
Date 2012-08-11 Date-obj 5 □ Print-Date
Time 10:04:40 Time-obj 5 ☑ Print-Time
☑ RTC-On
☑ Set DateTime ☑ RTC-Event-On
RTC read back from Kickstart: 2012.8.11. 10:4:54

Buzzer
Divider 12 ==> 2197 Hz
Time 0.5 sec ☑ Bridge-output
☑ Infinite □ Periodic beep

LCD-file
Select LCD
Comp Obj Seg

Bar test
Object 21

7segment test
Startobj 5 CharCode 0 Print String Length 6

14segment test
Startobj 0 CharCode 0 Print String Length 2

16segment test
Startobj 13 CharCode 0 Print String Length 7

Scroll-Text test
Startobj 13 Text to scroll Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1 Width 7

Generic (1-16seg) test
Object 20 Segments 0249h
15 □ 14 □ 13 □ 12 □ 11 □ 10 □ 9 ☑ 8 □ 7 □ 6 ☑ 5 □ 4 □ 3 ☑ 2 □ 1 □ 0 ☑

Attribute
Object 20 Length 1 Attribute Anim-Forward

☑ Show Cmd ☑ Show Response ☑ Show Kickstart Events

```
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,46
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,47
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,48
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,49
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,50
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,51
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,52
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,53
Response:  OK,Statread,18,208,0,12,8,11,7,10,4,54
```

And the host sends the following commands:

36, 10, 208, 12, 8, 11, 7, 10, 4, 40, 5, 5, 228

where

Number	Description
36	Command code (always 36)
10	Always 10
208	RTC_On=1 and RTC_print_time=1
12	Set RTC year to 2012
8	Set RTC month to August
11	Set RTC day to 11
7	Set RTC day of the week to Saturday
10	Set RTC hours to 10
4	Set RTC minutes to 4
40	Set RTC seconds to
5	Unimportant, because RTC_print_date=0
5	Object number of the first big 7-segment object that displays the time
228	Sum of all bytes, plus 128

See Example in Chapter "4.3.1. Status read and Cmd_Statread" about reading back the RTC and clearing the Event pin.

4.3.23. Cmd_Attr

Set attribute of a component.
The format of this command is

Cmd_Attr, length, object, attribute, size, checksum

where

Symbol	Length	Description
Cmd_ScrollTxt	1 byte	Command code (always 37)
length	1 byte	Always 3
object	1 byte	Object number of the first object of the component
attribute	1 byte	Attribute, see separate table
size	1 byte	Size of the component (=number of objects)
checksum	1 byte	Sum of all bytes, plus 128

The meaning of attribute bits are as follows:

Bit	Description
7-6	0=Static 1=Blink at 2 Hz rate 2=Animate forward at 2 Hz rate 3=Animate backward at 2 Hz rate
5-0	Reserved, always 0

Animation, bit rotation, is performed by rotating left/right the pre-filled segment pattern of the object.
Blinking repeatedly clears and replaces the original contents of the object, so it works only on
Cmd_Obj_W can be used to set the initial pattern. See Chapter "4.3.9. Cmd_Obj_W" for details.
After the attribute is reset to static the state of the previously blinking or animated object freezes in the state it last displayed, which may be entirely cleared.

4.3.24. Cmd_TestLED

Disable or enable test function of LEDs. Test function is enabled after startup. To use LEDs normally, test function must be disabled. See Chapter "4.3.25. Cmd_LED" for details.
The format of this command is

Cmd_TestLED, length, onoff, checksum

where

Symbol	Length	Description
Cmd_TestLED	1 byte	Command code (always 41)
length	1 byte	Always 1
onoff	1 byte	0=enable, 255=disable
checksum	1 byte	Sum of all bytes, plus 128

Example:

See Example in Chapter "4.3.25. Cmd_LED".

4.3.25. Cmd_LED

Set or clear the state of LEDs. To use LEDs normally, test function must be disabled. See Chapter "4.3.25. Cmd_LED" for details.
The format of this command is

Cmd_LED, length, bits, checksum

where

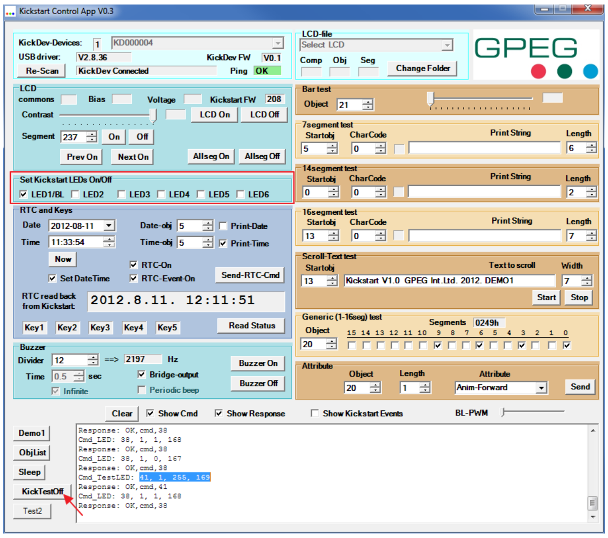
Symbol	Length	Description
Cmd_LED	1 byte	Command code (always 38)
length	1 byte	Always 1
bits	1 byte	See separate table
checksum	1 byte	Sum of all bytes, plus 128

The meaning of LED bits are as follows:

Bit	Description
7-6	Reserved, always 0
5	1=LED6 on, 0=LED6 off, don't care if LED6 is used by the buzzer
4	1=LED5 on, 0=LED5 off, don't care if LED5 is used by the buzzer
3	1=LED4 on, 0=LED4 off, don't care in SPI mode
2	1=LED3 on, 0=LED3 off, don't care in SPI mode
1	1=LED2 on, 0=LED2 off
0	1=LED1 and backlight on, 0=LED1 and backlight off

Example:

The user turns off test function of LEDs by clicking the "KickTestOff" button and then turns on the backlight by checking checkbox "LED1/BL" in "Set Kickstart LEDs On/Off" frame of KickDevApp.



Kickstart Control App V0.3

KickDev-Devices: 1 KD000004
USB driver: V2.8.36
KickDev FW V0.1
KickDev Connected
Ping OK
Kickstart FW 208
Segment 237
Set Kickstart LEDs On/Off
☑ LED1/BL ☐ LED2 ☐ LED3 ☐ LED4 ☐ LED5 ☐ LED6
Date 2012-08-11
Time 11:33:54
Date-obj 5 ☐ Print-Date
Time-obj 5 ☑ Print-Time
☑ RTC-On
☑ Set DateTime ☑ RTC-Event-On
RTC read back from Kickstart: 2012.8.11. 12:11:51
Divider 12 ==> 2197 Hz
Time 0.5 sec ☑ Bridge-output
☑ Infinite ☐ Periodic beep
Bar test Object 21
7segment test Startobj 5 CharCode 0 Length 6
14segment test Startobj 0 CharCode 0 Length 2
16segment test Startobj 13 CharCode 0 Length 7
Scroll-Text test Startobj 13 Text to scroll: Kickstart V1.0 GPEG Int.Ltd. 2012. DEMO1 Width 7
Generic (1-16seg) test Object 20 Segments 0249h
Attribute Object 20 Length 1 Attribute Anim-Forward
☑ Show Cmd ☑ Show Response ☐ Show Kickstart Events

```
Response: OK,cmd,38
Cmd_LED: 38, 1, 1, 168
Response: OK,cmd,38
Cmd_LED: 38, 1, 0, 167
Response: OK,cmd,38
Cmd_TestLED: 41, 1, 255, 169
Response: OK,cmd,41
Cmd_LED: 38, 1, 1, 168
Response: OK,cmd,38
```

And the host sends the following commands:

41, 1, 255, 169

38, 1, 1, 168

The first command turns off the test function of LEDs and the second command turns on the backlight.

4.3.26. Cmd_Buzz

Turn on and off buzzer and set its frequency.
The format of this command is

Cmd_Buzz, length, mode, divider, time, checksum

where

Symbol	Length	Description
Cmd_Buzz	1 byte	Command code (always 39)
length	1 byte	Always 3
mode	1 byte	Mode, see separate table
divider	1 byte	Frequency is 23438/(divider+1) Hz
time	1 byte	Reserved, 0
checksum	1 byte	Sum of all bytes, plus 128

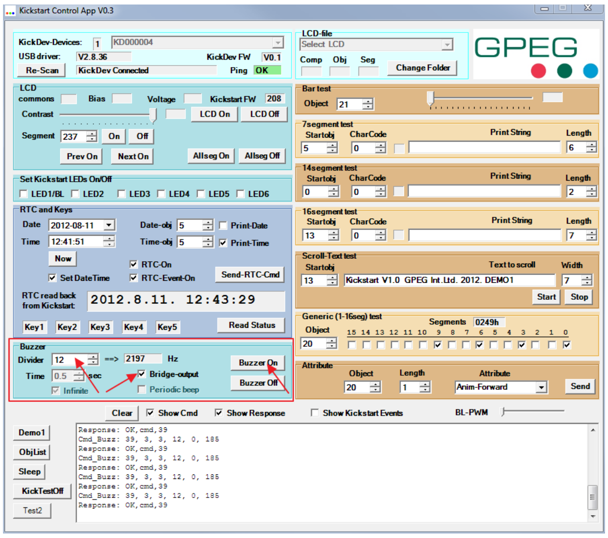
The meaning of mode bits are as follows:

Bit	Description
7-2	Reserved, always 0
1	1=Use LED6 for buzzer (normal polarity), 0=Use LED6 for LED control
0	1=Use LED5 for buzzer (inverted polarity), 0=Use LED5 for LED control

If neither LED6 nor LED5 are assigned to be used as buzzer outputs then they are controlled by the Cmd_LED and the Cmd_TestLED commands.
See Chapters "4.3.24. Cmd_TestLED" and "4.3.25. Cmd_LED". If mode=0 then the buzzer is off.
The buzzer operates only in RUN mode. It shall be disabled before entering SLEEP mode.

Example:

The user selects divider 12, and checks "Bridge-output" in "Buzzer" frame of KickDevApp and then clicks on "Buzzer On" button.



Buzzer
Divider 12 ==> 2197 Hz
Time 0.5 sec
☑ Infinite
☑ Bridge-output
☐ Periodic beep

```
Response: OK,cmd,39
Cmd_Buzz: 39, 3, 3, 12, 0, 185
Response: OK,cmd,39
Cmd_Buzz: 39, 3, 3, 12, 0, 185
Response: OK,cmd,39
Cmd_Buzz: 39, 3, 3, 12, 0, 185
Response: OK,cmd,39
Cmd_Buzz: 39, 3, 3, 12, 0, 185
Response: OK,cmd,39
```

And the host sends the following commands:

39, 3, 3, 12, 0, 185

and buzzer turns on. Frequency is approximately 2197 Hz.

4.3.27. Cmd_LCD_OnOff

Turns LCD on or off. After initialization the LCD is on.
The format of this command is

Cmd_LCD_OnOff, length, mode, checksum

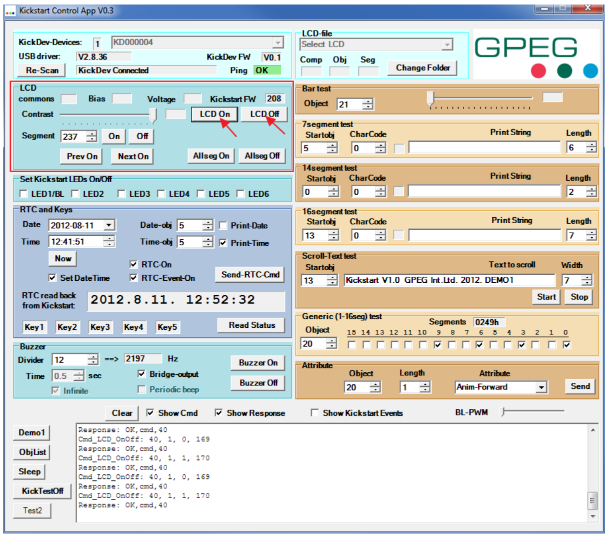
where

Symbol	Length	Description
Cmd_LCD_OnOff	1 byte	Command code (always 40)
length	1 byte	Always 1
mode	1 byte	1=on, 0=off
checksum	1 byte	Sum of all bytes, plus 128

If LCD is off and the controller is in SLEEP mode then keypresses cannot wake it up. For more information see Chapter "4.3.31. Cmd_Sleep".

Example:

The user clicks on "LCD Off" and then on "LCD On" button in "LCD" frame of KickDevApp.



```
Response: OK,cmd,40
Cmd_LCD_OnOff: 40, 1, 0, 169
Response: OK,cmd,40
Cmd_LCD_OnOff: 40, 1, 1, 170
Response: OK,cmd,40
Cmd_LCD_OnOff: 40, 1, 0, 169
Response: OK,cmd,40
Cmd_LCD_OnOff: 40, 1, 1, 170
Response: OK,cmd,40
```

And the host sends the following commands:

40, 1, 0, 169

40, 1, 1, 170

The first command turns off the display and the second turns it on again.

4.3.28. Cmd_AllSeg_On

Sets all segments of the LCD. Useful for testing.
The format of this command is

Cmd_AllSeg_On, length, checksum

where

Symbol	Length	Description
Cmd_LCD_OnOff	1 byte	Command code (always 42)
length	1 byte	Always 0
checksum	1 byte	Sum of all bytes, plus 128 (always 170)

This command sets the entire contents of the LCDRAM to 1, sets all bits of all objects to 1, and clears all attributes. It does not affect automatic scrolling and automatic display of time and date.

Example:

See Example in Chapter "4.3.29. Cmd_AllSeg_Off".

4.3.29. Cmd_AllSeg_Off

Clears all segments of the LCD. Useful for testing and initialization.
The format of this command is

Cmd_AllSeg_Off, length, checksum

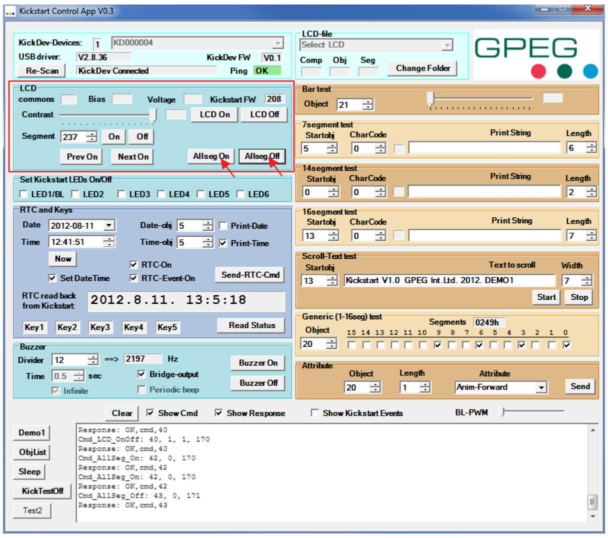
where

Symbol	Length	Description
Cmd_LCD_OnOff	1 byte	Command code (always 43)
length	1 byte	Always 0
checksum	1 byte	Sum of all bytes, plus 128 (always 171)

This command clears the entire contents of the LCDRAM, clears all bits of all objects, and clears all attributes. It does not affect automatic scrolling and automatic display of time and date.

Example:

The user clicks on "Allseg On" and then on "Allseg Off" button in "LCD" frame of KickDevApp.



```
Response: OK,cmd,40
Cmd_LCD_OnOff: 40, 1, 1, 170
Response: OK,cmd,40
Cmd_AllSeg_On: 42, 0, 170
Response: OK,cmd,42
Cmd_AllSeg_On: 42, 0, 170
Response: OK,cmd,42
Cmd_AllSeg_Off: 43, 0, 171
Response: OK,cmd,43
```

And the host sends the following commands:

42, 0, 170

43, 0, 171

The first command turns on all segments and the second turns off all segments.

4.3.30. Cmd_Contrast

Sets the contrast of the LCD.
The format of this command is

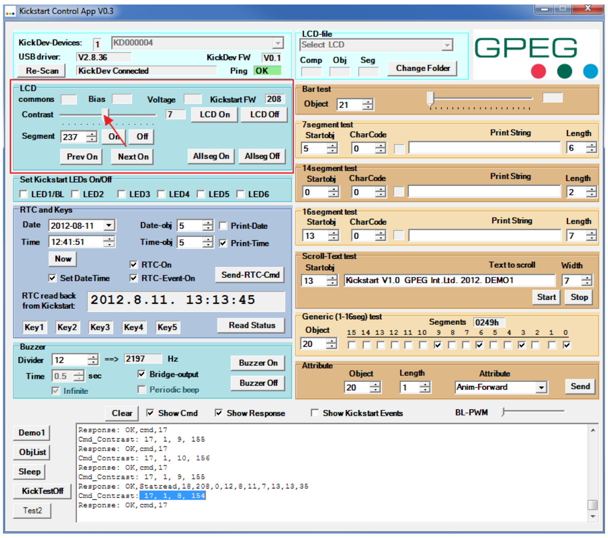
Cmd_Contrast, length, contrast, checksum

where

Symbol	Length	Description
Cmd_Contrast	1 byte	Command code (always 17)
length	1 byte	Always 1
contrast	1 byte	0..15, 0=maximum, 15= minimum
checksum	1 byte	Sum of all bytes, plus 128

Example:

The user moves the "Contrast" slider in "LCD" frame of KickDevApp into middle position.



```
Response: OK,cmd,17
Cmd_Contrast: 17, 1, 9, 155
Response: OK,cmd,17
Cmd_Contrast: 17, 1, 10, 156
Response: OK,cmd,17
Cmd_Contrast: 17, 1, 9, 155
Response: OK,Statread,18,208,0,12,8,11,7,13,13,35
Cmd_Contrast: 17, 1, 8, 154
Response: OK,cmd,17
```

And the host sends the following command:

17, 1, 8, 154

and the contrast of the LCD changes.

4.3.31. Cmd_Sleep

KickStart™ LCD Controller has three operating modes.

Mode	Command execution	RTC, blinking, animation	Keypress detection	LCD	Power
RUN	Yes	Yes	Yes	Yes	1300 µA
SLEEP	Yes	Yes	Yes	Yes	30 µA
DEEPSLEEP	Yes	No	No	No	1 µA

If a command, a keypress or an RTC event wakes up KickStart™ LCD Controller from SLEEP, it operates in RUN mode until the command or the event is processed and then returns to SLEEP, if the command was not a Cmd_Sleep command that changed the operating mode.

If a command wakes up KickStart™ LCD Controller from DEEPSLEEP, it operates in RUN mode until the command is received and executed and then returns to DEEPSLEEP, if the command was not a Cmd_Sleep command that changed the operating mode.

The format of this command is

Cmd_Sleep, length, mode, checksum

where

Symbol	Length	Description
Cmd_Sleep	1 byte	Command code (always 44)
length	1 byte	Always 1
mode	1 byte	0=RUN, 1=SLEEP, 2=DEEPSLEEP
checksum	1 byte	Sum of all bytes, plus 128

Example:

The host sends the following command:

44, 1, 1, 174

and KickStart™ LCD Controller goes to SLEEP.

5. KickStart™ LCD Controller hardware

The SK480 is an intelligent LCD Controller with capability of driving LCDs up to 480 segments. The device can be configured to operate in any LCD system. Interface via modified SPI and modified I²C is supported, along with a simple, high level command structure, significantly reducing programming time and errors. When the SK480 LCD Controller is used in conjunction with the KickStart™ on-line tool it eliminates the need for the traditional LCD pin lookup table, time consuming font tables and power hungry animated elements.

SK480 is available in two packages:

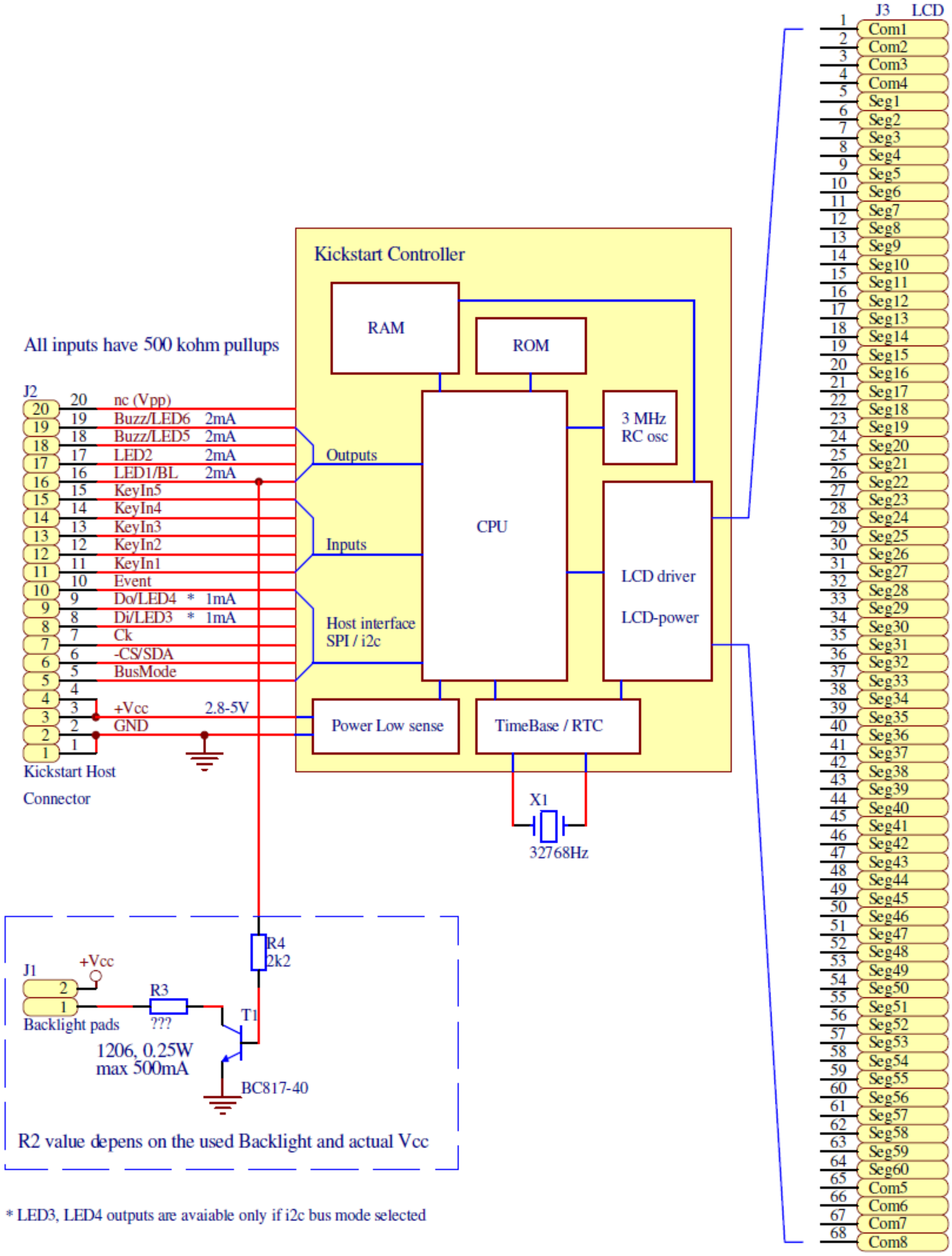
- COF - Intended for general use
- DIE - Intended for high volume, cost sensitive or space constrained applications

5.1. Features

- Selectable duty cycle: 1/4, 1/5, 1/6, 1/8
- Max. 60 segment lines
- Max. 8 common lines
- Max. 480 segments
- Monochrome, no grayscale
- SPI and modified I²C interfaces
- Operating voltage 2.7 - 5.5V
- Low power SLEEP mode (< 30µA) with LCD on, keypress monitoring, RTC running
- Ultra-low power DEEPSLEEP mode (< 1µA), only command wakeup
- Built-in RTC (Real Time Clock) with automatic display update, and event generation
- Built-in 7/14/16 segment character generator
- Built-in Bar-converter
- Built-in oscillator and external 32.768kHz crystal
- Built-in symbol-animation and symbol blinking
- Text string and bargraph display
- Max. 255 character scrolling text
- Intuitive and easy to use high level command set
- On/off backlight control
- Buzzer output with variable frequencies
- 5 general purpose input pins, for key connection
- 6 general purpose output pins, for backlight control, LED or buzzer

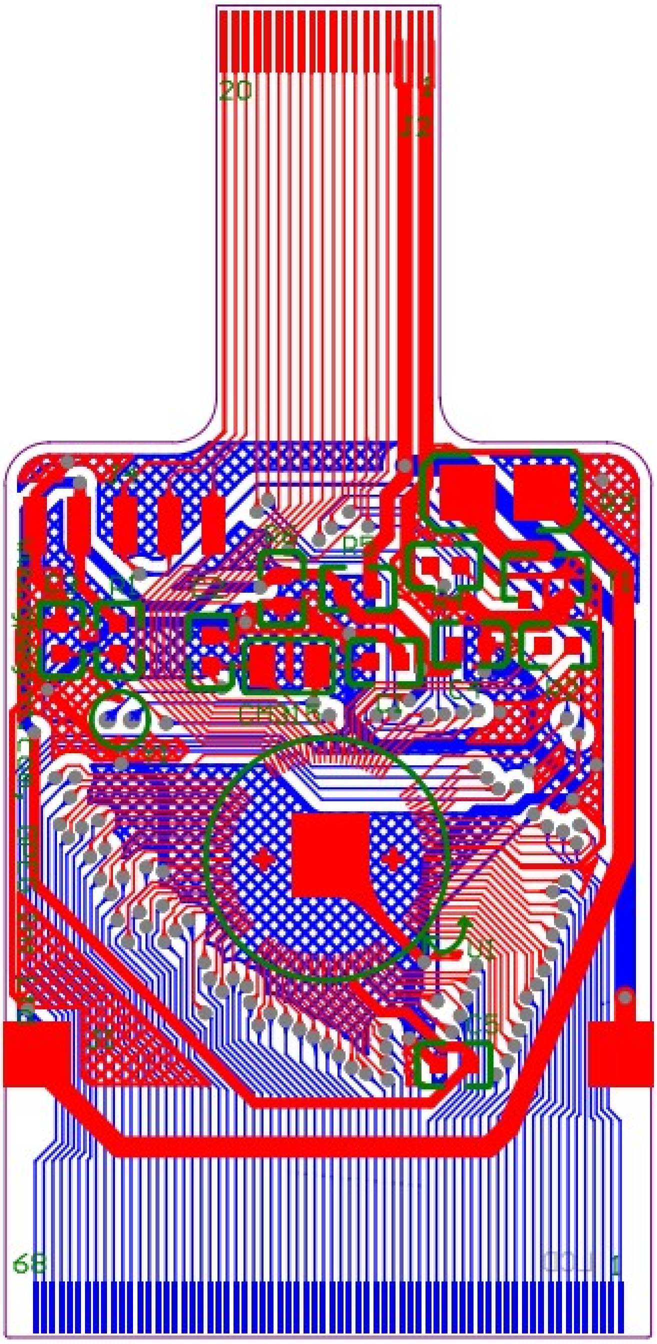
5.2. COF version

5.2.1. Block diagram



* LED3, LED4 outputs are avaiable only if i2c bus mode selected

5.2.2. Dimensions



5.2.3. Interface connector pinout

Number	Name	Direction	Description
1	GND		Signal and power ground
2	GND		Signal and power ground
3	VCC	I	Power supply to LCD and LCD controller
4	VCC	I	Power supply to LCD and LCD controller
5	BusMode	I	Interface mode (1=I²C, 0=SPI)
6	CS/SDA	I or I/O	Chip select in SPI mode, SDA in I²C mode
7	Ck/SCL	I or I/O	Clock in SPI mode, SCL in I²C mode
8	DI/LED3	I or O	Data input in SPI mode, LED3 in I²C mode
9	DO/LED4	O	Data output in SPI mode, LED4 in I²C mode
10	Event	O	Event output
11	KeyIn1	I	Key 1 input
12	KeyIn2	I	Key 2 input
13	KeyIn3	I	Key 3 input
14	KeyIn4	I	Key 4 input
15	KeyIn5	I	Key 5 input
16	LED1/BL	O	LED1 or backlight control output
17	LED2	O	LED2 output
18	Buzz/LED5	O	Main buzzer output or LED5 output
19	Buzz/LED6	O	Inverted buzzer output or LED6 output
20	Vpp		Reserved, leave it unconnected

Directions are as seen by the LCD controller.

5.2.4. Output connector pinout

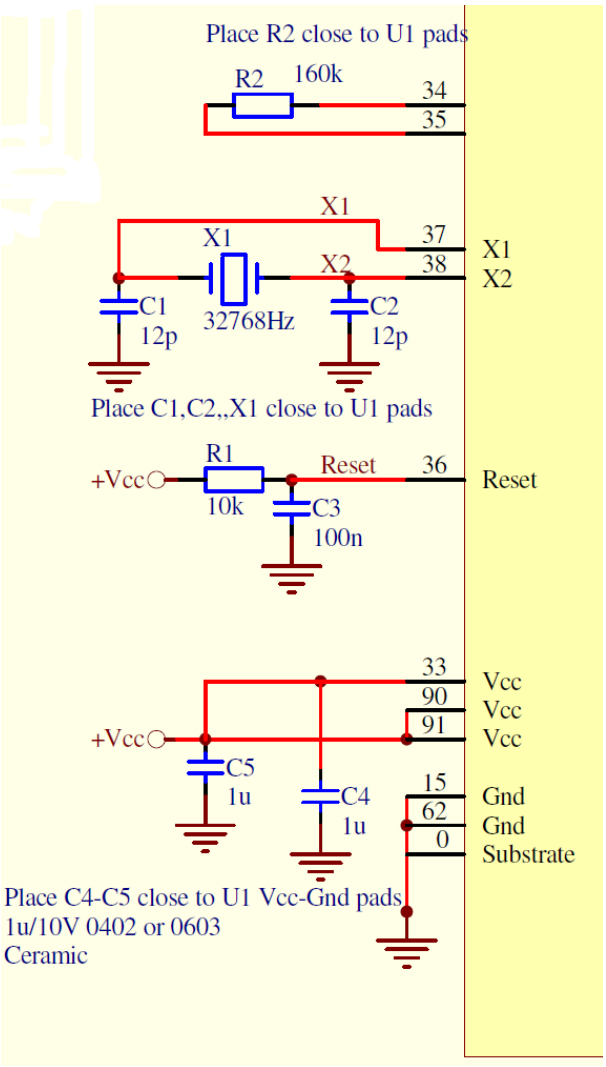
#	Name	Description
1	Com1	Common line 1
2	Com2	Common line 2
3	Com3	Common line 3
4	Com4	Common line 4
5	Seg1	Segment line 1
6	Seg2	Segment line 2
7	Seg3	Segment line 3
8	Seg4	Segment line 4
9	Seg5	Segment line 5
10	Seg6	Segment line 6
11	Seg7	Segment line 7
12	Seg8	Segment line 8
13	Seg9	Segment line 9
14	Seg10	Segment line 10
15	Seg11	Segment line 11
16	Seg12	Segment line 12
17	Seg13	Segment line 13
18	Seg14	Segment line 14
19	Seg15	Segment line 15
20	Seg16	Segment line 16
21	Seg17	Segment line 17
22	Seg18	Segment line 18
23	Seg19	Segment line 19
24	Seg20	Segment line 20
25	Seg21	Segment line 21
26	Seg22	Segment line 22
27	Seg23	Segment line 23
28	Seg24	Segment line 24
29	Seg25	Segment line 25
30	Seg26	Segment line 26
31	Seg27	Segment line 27
32	Seg28	Segment line 28
33	Seg29	Segment line 29
34	Seg30	Segment line 30

#	Name	Description
35	Seg31	Segment line 31
36	Seg32	Segment line 32
37	Seg33	Segment line 33
38	Seg34	Segment line 34
39	Seg35	Segment line 35
40	Seg36	Segment line 36
41	Seg37	Segment line 37
42	Seg38	Segment line 38
43	Seg39	Segment line 39
44	Seg40	Segment line 40
45	Seg41	Segment line 41
46	Seg42	Segment line 42
47	Seg43	Segment line 43
48	Seg44	Segment line 44
49	Seg45	Segment line 45
50	Seg46	Segment line 46
51	Seg47	Segment line 47
52	Seg48	Segment line 48
53	Seg49	Segment line 49
54	Seg50	Segment line 50
55	Seg51	Segment line 51
56	Seg52	Segment line 52
57	Seg53	Segment line 53
58	Seg54	Segment line 54
59	Seg55	Segment line 55
60	Seg56	Segment line 56
61	Seg57	Segment line 57
62	Seg58	Segment line 58
63	Seg59	Segment line 59
64	Seg60	Segment line 60
65	Com5	Common line 5
66	Com6	Common line 6
67	Com7	Common line 7
68	Com8	Common line 8

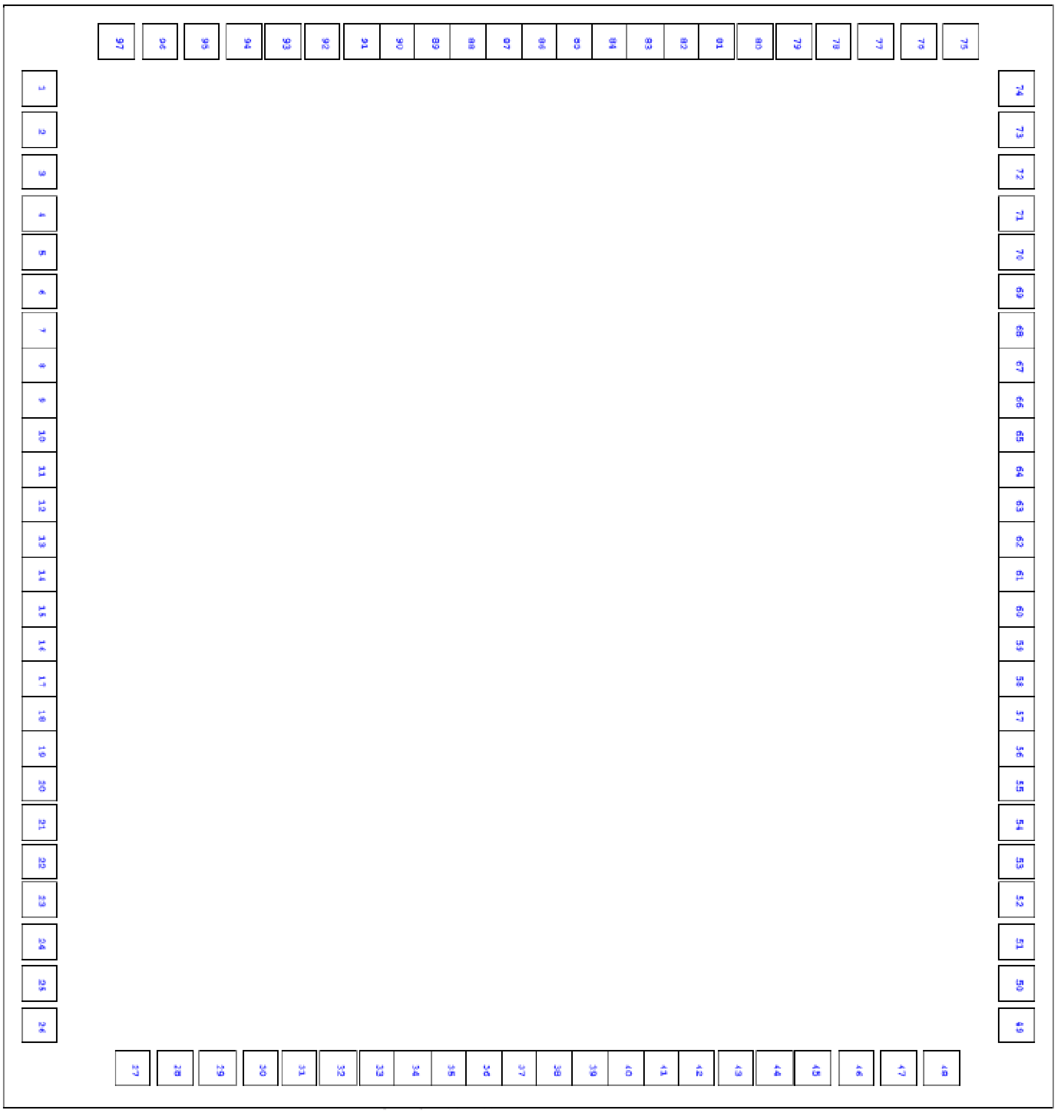
All lines are outputs of the LCD controller.

5.3. DIE version

Some passive components are integrated into the COF version, which has to be added externally to the DIE version. This schematics shows these passive components:



5.3.1. Pad numbers and names of the DIE version



Chip Size: 3240 x 3060 µm

Note: the IC substrate should be connected to GND on the PCB layout.

5.3.2. Pad coordinates

PAD	Name	X (µm)	Y (µm)
1	SEG14	-1340.000	-1378.500
2	SEG13	-1220.000	-1378.500
3	SEG12	-1100.000	-1378.500
4	SEG11	-980.000	-1378.500
5	SEG10	-870.000	-1378.500
6	SEG9	-760.000	-1378.500
7	SEG8	-650.000	-1378.500
8	SEG7	-550.000	-1378.500
9	SEG6	-450.000	-1378.500
10	SEG5	-350.000	-1378.500
11	SEG4	-250.000	-1378.500
12	SEG3	-150.000	-1378.500
13	SEG2	-50.000	-1378.500
14	SEG1	50.000	-1378.500
15	GND	150.000	-1378.500
16	COM1	250.000	-1378.500
17	COM2	350.000	-1378.500
18	COM3	450.000	-1378.500
19	COM4	550.000	-1378.500
20	COM5	650.000	-1378.500
21	COM6	760.000	-1378.500
22	COM7	870.000	-1378.500
23	COM8	980.000	-1378.500
24	NC	1100.000	-1378.500
25	NC	1220.000	-1378.500
26	GP9	1340.000	-1378.500
27	GP12	1463.000	-1115.000
28	GP13	1463.000	-995.000
29	GP14	1463.000	-875.000
30	GP15	1463.000	-755.000
31	GP11	1463.000	-645.000
32	GP10	1463.000	-535.000
33	VCC	1463.000	-425.000
34	VR1	1463.000	-325.000
35	VR2	1463.000	-225.000
36	RESET	1463.000	-125.000
37	X1	1463.000	-25.000
38	X2	1463.000	75.000
39	NC	1463.000	175.000
40	GP0	1463.000	275.000
41	GP1	1463.000	375.000
42	GP2	1463.000	475.000
43	GP3	1463.000	585.000
44	GP4	1463.000	695.000
45	GP5	1463.000	805.000
46	GP6	1463.000	925.000
47	GP7	1463.000	1045.000
48	SEG60	1463.000	1165.000
49	SEG59	1340.000	1378.500
50	SEG58	1220.000	1378.500
51	SEG57	1100.000	1378.500
52	SEG56	980.000	1378.500
53	SEG55	870.000	1378.500
54	SEG54	760.000	1378.500
55	SEG53	650.000	1378.500
56	SEG52	550.000	1378.500
57	SEG51	450.000	1378.500
58	SEG50	350.000	1378.500
59	SEG49	250.000	1378.500
60	SEG48	150.000	1378.500
61	SEG47	50.000	1378.500
62	GND	-50.000	1378.500
63	SEG46	-150.000	1378.500
64	SEG45	-250.000	1378.500
65	SEG44	-350.000	1378.500
66	SEG43	-450.000	1378.500
67	SEG42	-550.000	1378.500
68	SEG41	-440.000	1378.500
69	SEG40	-760.000	1378.500
70	SEG39	-870.000	1378.500
71	SEG38	-980.000	1378.500
72	SEG37	-1100.000	1378.500
73	SEG36	-1220.000	1378.500
74	SEG35	-1340.000	1378.500
75	SEG34	-1473.500	1221.000
76	SEG33	-1473.500	1101.000
77	SEG32	-1473.500	981.000
78	SEG31	-1473.500	861.000
79	SEG30	-1473.500	751.000
80	SEG29	-1473.500	641.000
81	SEG28	-1473.500	531.000
82	SEG27	-1473.500	431.000
83	SEG26	-1473.500	331.000
84	SEG25	-1473.500	231.000
85	SEG24	-1473.500	131.000
86	SEG23	-1473.500	31.000
87	SEG22	-1473.500	-69.000
88	SEG21	-1473.500	-169.000
89	SEG20	-1473.500	-269.000
90	VCC	-1473.500	-369.000
91	VCC	-1473.500	-469.000
92	SEG19	-1473.500	-579.000
93	VPP	-1473.500	-689.000
94	SEG18	-1473.500	-799.000
95	SEG17	-1473.500	-919.000
96	SEG16	-1473.500	-1039.000
97	SEG15	-1473.500	-1159.000

5.3.3. Pin description

PAD	Name	Description or COF equivalent
1	SEG14	=Seg14
2	SEG13	=Seg13
3	SEG12	=Seg12
4	SEG11	=Seg11
5	SEG10	=Seg10
6	SEG9	=Seg9
7	SEG8	=Seg8
8	SEG7	=Seg7
9	SEG6	=Seg6
10	SEG5	=Seg5
11	SEG4	=Seg4
12	SEG3	=Seg3
13	SEG2	=Seg2
14	SEG1	=Seg1
15	GND	=GND
16	COM1	=Com1
17	COM2	=Com2
18	COM3	=Com3
19	COM4	=Com4
20	COM5	=Com5
21	COM6	=Com6
22	COM7	=Com7
23	COM8	=Com8
24	NC	NC, Reserved
25	NC	NC, Reserved
26	GP9	=Do/LED4
27	GP12	=Ck/SCL
28	GP13	=Di/LED3
29	GP14	=BusMode
30	GP15	=Event
31	GP11	=LED2
32	GP10	=LED1/BL
33	VCC	=Vcc
34	VR1	* Note 1
35	VR2	* Note 1
36	RESET	* Note 1
37	X1	* Note 1
38	X2	* Note 1
39	NC	Reserved
40	GP0	=CS/SDA
41	GP1	=KeyIn1
42	GP2	=KeyIn2
43	GP3	=KeyIn3
44	GP4	=KeyIn4
45	GP5	=KeyIn5
46	GP6	=Buzz/LED5
47	GP7	=Buzz/LED6
48	SEG60	=Seg60
49	SEG59	=Seg59
50	SEG58	=Seg58
51	SEG57	=Seg57
52	SEG56	=Seg56
53	SEG55	=Seg55
54	SEG54	=Seg54
55	SEG53	=Seg53
56	SEG52	=Seg52
57	SEG51	=Seg51
58	SEG50	=Seg50
59	SEG49	=Seg49
60	SEG48	=Seg48
61	SEG47	=Seg47
62	GND	=GND
63	SEG46	=Seg46
64	SEG45	=Seg45
65	SEG44	=Seg44
66	SEG43	=Seg43
67	SEG42	=Seg42
68	SEG41	=Seg41
69	SEG40	=Seg40
70	SEG39	=Seg39
71	SEG38	=Seg38
72	SEG37	=Seg37
73	SEG36	=Seg36
74	SEG35	=Seg35
75	SEG34	=Seg34
76	SEG33	=Seg33
77	SEG32	=Seg32
78	SEG31	=Seg31
79	SEG30	=Seg30
80	SEG29	=Seg29
81	SEG28	=Seg28
82	SEG27	=Seg27
83	SEG26	=Seg26
84	SEG25	=Seg25
85	SEG24	=Seg24
86	SEG23	=Seg23
87	SEG22	=Seg22
88	SEG21	=Seg21
89	SEG20	=Seg20
90	VCC	=Vcc
91	VCC	=Vcc
92	SEG19	=Seg19
93	VPP	=Vpp, Reserved
94	SEG18	=Seg18
95	SEG17	=Seg17
96	SEG16	=Seg16
97	SEG15	=Seg15

* Note 1: See COF and DIE differences.

5.4. Absolute maximum ratings

Parameter	Symbol	Ratings
DC supply voltage	V_{CC}	<6.0V
Input voltage range	V_{IN}	-0.5V to VCC +0.5V
Operating temperature range	T_A	0°C to +60°C
Storage temperature range	T_{STO}	-50°C to +150°C

Note: Stresses beyond these may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

5.5. DC characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Operating voltage	V_{CC}	2.6	3	5.5	V
Operating current	I_{OP}		1.3		mA
DEEPSLEEP current	$I_{DEEPSLEEP}$			1	µA
SLEEP current	I_{SLEEP}			30	µA
Input high level	V_{IH}	0.7 VCC			
Input low level	V_{IL}			0.2 VCC	

5.6. Modified I²C mode

KickStart™ LCD Controller uses a modified version of the I²C protocol.

- Clock speed is limited to 25kHz, or with some restriction to 50kHz.
- Repeated start condition is not used.
- Premature termination of a command by sending stop condition in the middle of a command sequence is not allowed.

Because of these limitations it is not recommended to have other I²C devices on the same bus. If it is unavoidable, then these must operate at less than 25kHz and must not use repeated start conditions.

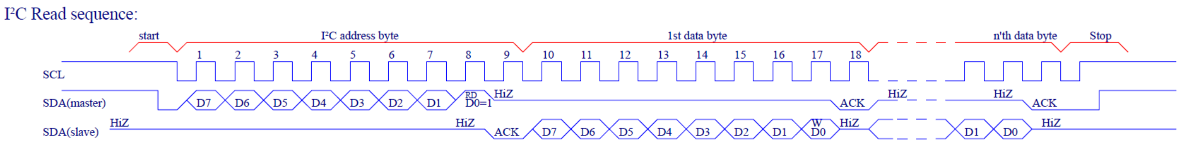
KickStart™ LCD Controller is always slave on the I²C bus and has fixed read and write address. The read address is decimal 202, the write address is decimal 203.

All SDA falling edges wake up KickStart™ LCD Controller from SLEEP or DEEPSLEEP. KickStart™ LCD Controller remains in RUN mode until it receives and executes the command or decides that it is not the start of a valid command.

5.6.1. I²C read

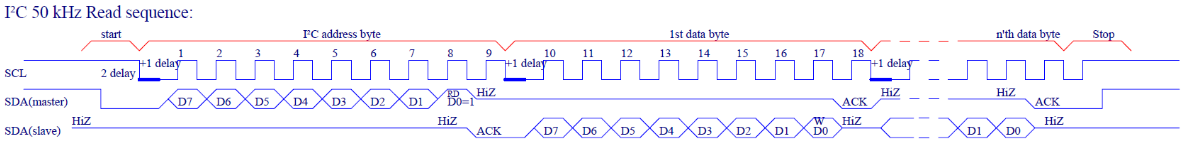
The host can read only the status in I²C mode. See more details in Chapter "4.3.1. Status read and Cmd_Statread".

Up to 25 kHz the communication happens exactly as defined by the I²C specification



1. The host generates a start condition.
2. The host sends the read address of KickStart™ LCD Controller, which is decimal 202.
3. The host checks if KickStart™ LCD Controller acknowledged the read address.
4. If not then KickStart™ LCD Controller is busy and the host continues with step 8.
5. If yes then KickStart™ LCD Controller it is ready to send status bytes.
6. The host generates clocks and receives the status bytes and the checksum.
7. The host acknowledges all status bytes and the checksum.
8. The host generates stop condition.

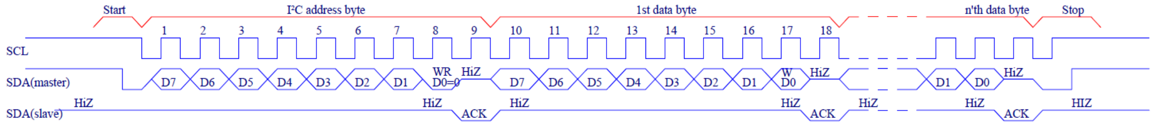
KickStart™ LCD Controller can be used up to 50 kHz clock, if a few additional cycles are inserted where it is indicated in the following drawing:



5.6.2. I²C write

Up to 25 kHz the communication happens exactly as defined by the I²C specification

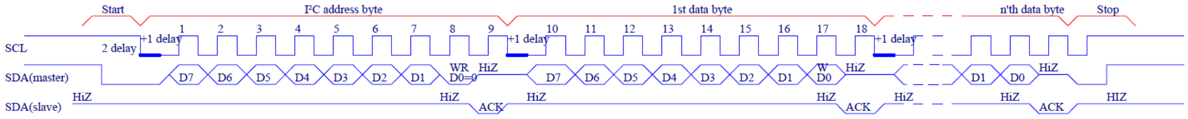
I²C Write sequence:



1. The host generates a start condition.
2. The host sends the write address of KickStart™ LCD Controller, which is decimal 203.
3. The host checks if KickStart™ LCD Controller acknowledged the write address.
4. If not then KickStart™ LCD Controller is busy and the host continues with step 8.
5. If yes then KickStart™ LCD Controller it is ready to receive data.
6. The host sends the command, its parameters and the checksum.
7. KickStart™ LCD Controller acknowledges all bytes and the checksum.
8. The host generates stop condition.

KickStart™ LCD Controller can be used up to 50 kHz clock, if a few additional cycles are inserted where it is indicated in the following drawing:

I²C 50 kHz Write sequence:



5.6.3. I²C timing characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Clock frequency	F_{CLK}	10	50	50	kHz
Clock high time	T_{HIGH}	10			µs
Clock low time	T_{LOW}	10			µs
Clock low time before 1st clock	T_{LOW1}	20			µs
SDA, SCL rise time	T_R			1	µs
SDA, SCL fall time	T_F			1	µs
Start condition hold time	$T_{HD:STA}$	20			µs
Start condition setup time	$T_{SU:STA}$	20			µs
Data input hold time time	$T_{HD:DAT}$	0			µs
Data input setup time	$T_{SU:DAT}$	0			µs
Stop condition setup time	$T_{HD:STO}$	10			µs

5.7. Modified SPI mode

KickStart™ LCD Controller uses a modified version of the SPI protocol.

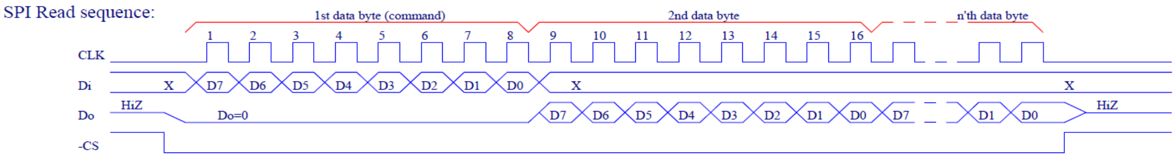
- Clock speed is limited to 25kHz, or with some restriction to 50kHz.
- Non-standard handshaking is used.

KickStart™ LCD Controller is always slave on the SPI bus.

5.7.1. SPI read

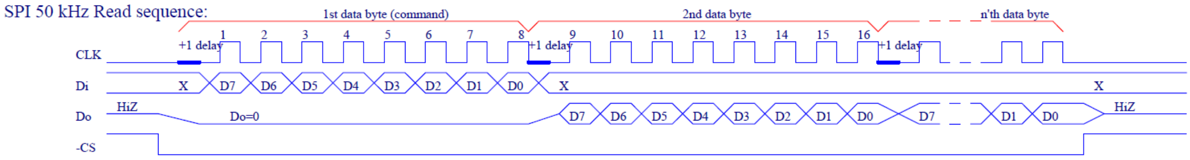
The host can read only the status in SPI mode. See more details in Chapter "4.3.1. Status read and Cmd_Statread".

Up to 25 kHz the communication happens exactly as defined by the SPI specification, with the addition of non-standard handshaking.



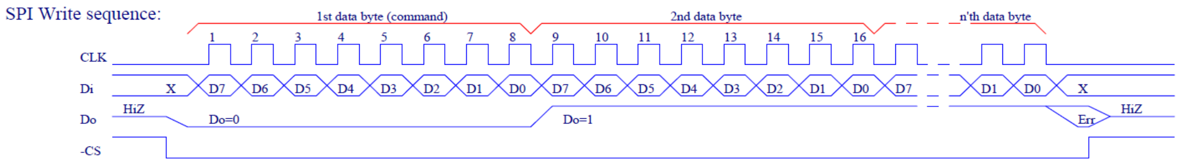
1. The host asserts CS.
2. The host checks if DI is low.
3. If not then KickStart™ LCD Controller is busy and the host continues with step 7.
4. If yes then KickStart™ LCD Controller it is ready to receive commands.
5. The host generates clocks and sends the Cmd_Statread command.
6. The host generates clocks and receives the status bytes and the checksum.
7. The host deasserts CS.

KickStart™ LCD Controller can be used up to 50 kHz clock, if a few additional cycles are inserted where it is indicated in the following drawing:



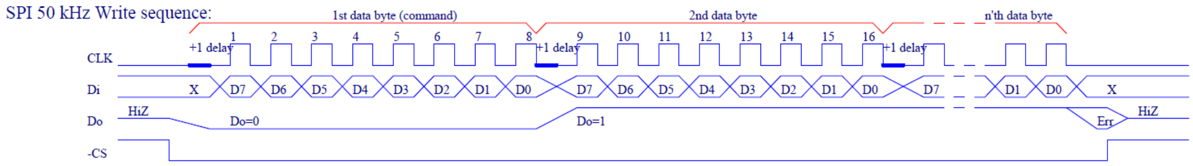
5.7.2. SPI write

Up to 25 kHz the communication happens exactly as defined by the SPI specification, with the addition of non-standard handshaking and error checking.



1. The host asserts CS.
2. The host checks if DI is low.
3. If not then KickStart™ LCD Controller is busy and the host continues with step 7.
4. If yes then KickStart™ LCD Controller it is ready to receive commands.
5. The host generates clocks and sends the command, its data and the checksum.
6. The host checks DI after the last clock. If it is low, the data is received correctly.
7. The host deasserts CS.

KickStart™ LCD Controller can be used up to 50 kHz clock, if a few additional cycles are inserted where it is indicated in the following drawing:



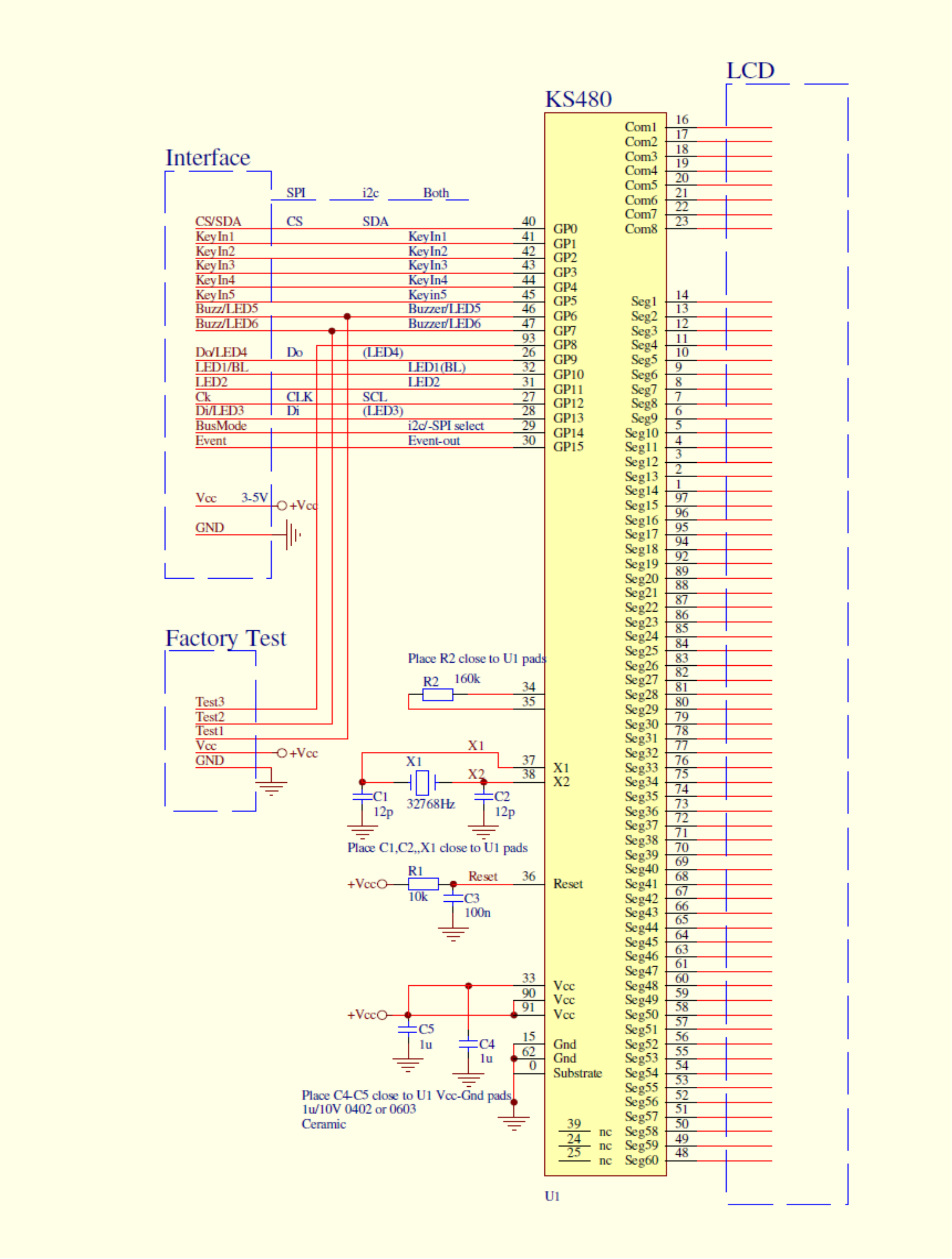
The CS pin is monitored continuously, the host can prematurely terminate the command any time. The partially received command will not be executed. Partially received Cmd_Seg_Tbl, Cmd_Obj_Tbl and Cmd_ScrollTxt commands have undefined effect.

5.7.3. SPI timing characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Clock frequency	F_{CLK}	10	50	50	kHz
Clock high time	T_{HIGH}	10			µs
Clock low time	T_{LOW}	10			µs
Clock low time before 1st clock	T_{LOW1}	20			µs
Signal rise time	T_{R}			1	µs
Signal fall time	T_{F}			1	µs
CS hold time	$T_{HD:CS}$	20			µs
CS setup time	$T_{SU:CS}$	20			µs
Data input hold time	$T_{HD:DI}$	0			µs
Data input setup time	$T_{SU:DI}$	0			µs
Data output hold time	$T_{HD:DO}$	0			µs
Data output setup time	$T_{SU:DI}$	0			µs

5.8. Recommended use

This is the schematics of the COF package, with signal names that indicate recommended use.



5.8.1. Using SLEEP mode effectively

The power consumption of KickStart™ LCD Controller is very low in SLEEP mode, which is especially important in battery powered application. But to use SLEEP mode effectively the following precautions shall be observed:

- Every I²C bus activity wakes up KickStart™ LCD Controller and increases its power consumption. This is why it is not recommended to have other devices on the I²C bus.
- If segments that are never used as part of objects are removed from the objects table, the time the processor is in RUN mode can be reduced. The objects table is not directly editable by the end user, but our engineers can make this change in no time for a small fee.
- The time it takes to receive a command is comparable to the time it takes to execute it. Therefore it is advisable to use the highest possible clock rate to reduce command reception time.

5.8.2. Displaying the date in different formats

Different counties display date in different order. If an LCD is intended for international market, it is often required that the order shall be configurable at runtime.

In KickStart™ LCD Controller it is possible to define more than one component that contains the same objects in different order. The objects table is not directly editable by the end user, but our engineers can make this change in no time for a small fee. We can also create object tables, with virtual (invisible) objects, that displays only part of the date or part of the time. The Cmd_RTC command always displays date and time in six consequtive objects, but those digits which updates the virtual (invisible) objects will not be visible.