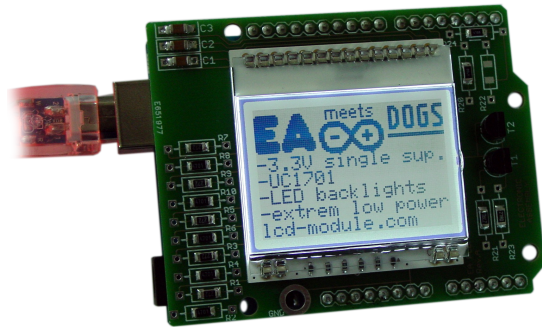


## Arduino meets EA DOG-Graphic series with UC1701 controller



This little project describes the connection between an Arduino-Board and a ELECTRONIC ASSEMBLY DOG Display with UC1701 controller, like the EA DOGS102-6. For more details, you will find the display's datasheet under <http://www.lcd-module.de/eng/pdf/grafik/dogs102-6e.pdf> and the controller's under <http://www.lcd-module.de/eng/pdf/zubehoer/uc1701.pdf>.




### Table of content

1. Hardware.....	2
1.1. Display.....	2
1.2. Backlight.....	2
1.3. Schematic.....	3
1.4. PCB.....	4
1.5. Bill of Materials.....	4
1.5.1. EA DOGS102-6 with monochrome backlight.....	4
1.5.2. EA DOGS102-6 with GR backlight.....	5
2. Software library classname: dog_1701.....	6
2.1. initialize.....	7
2.2. clear.....	7
2.3. contrast.....	7
2.4. view.....	8
2.5. string.....	8
2.6. rectangle.....	8
2.7. picture.....	9
2.8. Backlight.....	9
3. The Library in action.....	10

## 1. Hardware

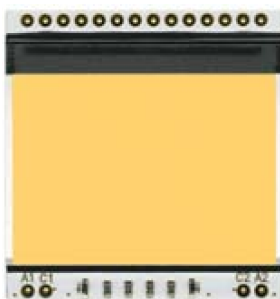
### 1.1. Display

We have 3 different display technologies in the DOGS-Series to match every area of application:

display type	technology	optional backlight	readability	display color non backlighted	display color with backlighted	recommend backlight color
	FSTN pos. transfective	it's fine with and without backlight	readable even without backlight	black on white	black on backlight color	all
	STN neg. blue transmissive	usage only with backlight	---	---	white backlight on blue background	white
	FSTN pos. white reflective	no backlight possible	finest readable without backlight	black on white	---	---

### 1.2. Backlight

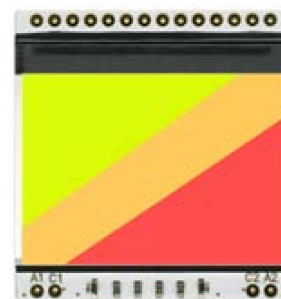
We have 3 different backlight modules, 2 monochrome versions and RGB. Depending on your choice you have to assemble different resistors and transistors. Please refer to the part list.



EA LED39x41-A Amber



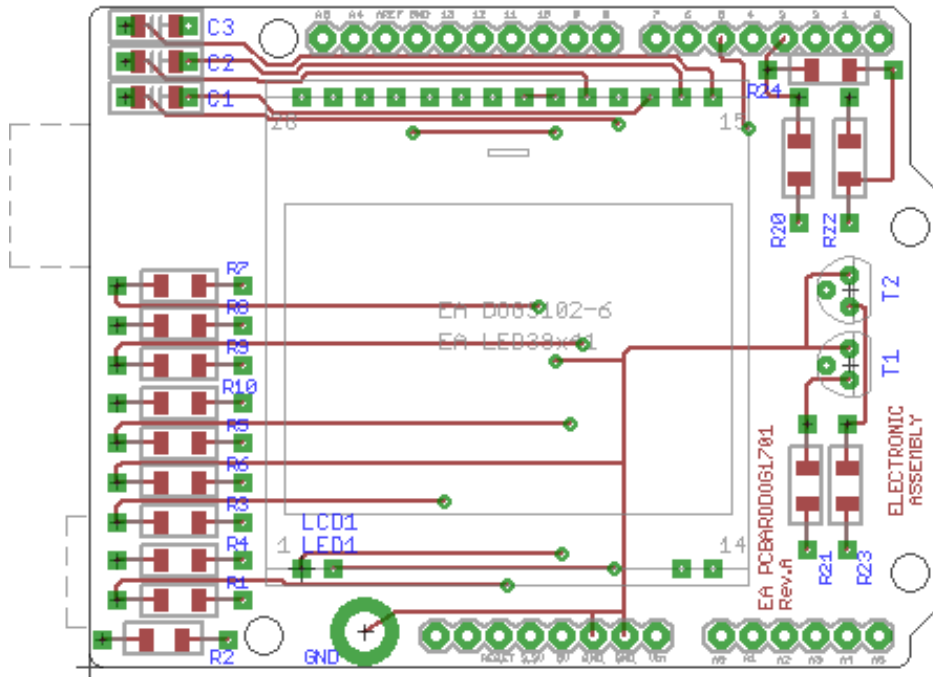
EA LED39x41-W White



EA LED39x41-GR Green/Red



## 1.4. PCB



## 1.5. Bill of Materials

### 1.5.1. EA DOGS102-6 with monochrome backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C2, C3	1 $\mu$ F	80-ARR07D105KGS	X7R-G1206 1,0/50
R20, R24	470 $\Omega$	273-470-RC	SMD 1/4W 470
R1, R3, R5, R7, R9	2.7 K $\Omega$	273-2.7K-RC	SMD 1/4W 2,7K
R2, R4, R6, R8, R10	4.7 K $\Omega$	273-4.7K-RC	SMD 1/4W 4,7K
R21, R23	White BL: 27 $\Omega$ EA LED39X41-W	594-5083NW27R00J	SMD 1/4W 27
	Amber BL: 36 $\Omega$ EA LED39X41-A	594-5093NW36R00J	SMD 1/4W 36
T1, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGS102X-6	790-EADOGS102W6	EA DOGS102W-6
LED1	EA LED39X41-GR	790-EALED39X41GR	EA LED39X41-GR

# Arduino meets EA DOGS-Graphic series

with UC1701 controller

## 1.5.2. EA DOGS102-6 with GR backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C2, C3	1 $\mu$ F	80-ARR07D105KGS	X7R-G1206 1,0/50
R21	39 $\Omega$	594-5083NW39R00J	SMD 1/4W 39
R23	51 $\Omega$	594-5083NW51R00J	SMD 1/4W 51
R20, R22	470 $\Omega$	273-470-RC	SMD 1/4W 470
R1, R3, R5, R7, R9	2.7 K $\Omega$	273-2.7K-RC	SMD 1/4W 2,7K
R2, R4, R6, R8, R10	4.7 K $\Omega$	273-4.7K-RC	SMD 1/4W 4,7K
T1, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGS102X-6	790-EADOGS102W6	EA DOGS102W-6

## 2. Software library classname: dog\_1701

The library provides all necessary functions for using a EA DOG graphic display with ST7565R controller.

Importing the library is very easy: start Arduino application, Sketch → Import Library → Add Library

Please add dog\_1701.zip Your sketch should look like this:

```
#include <Arduino.h>
#include <SPI.h>
#include <dog_1701.h>
#include <font_16x32nums.h>
#include <font_6x8.h>
#include <font_8x16.h>
#include <font_8x8.h>
#include <logo_BLH.h>

dog_1701 DOG;

void setup()
{

}

void loop()
{

}
```

In addition you will find examples for all EA DOG-Displays with UC1701 controller in the library folder “examples”.

These functions and definitions are provided within the library:

Classname: **dog\_1701**

```
#define DOGS102 1
#define VIEW_BOTTOM 0xC0
#define VIEW_TOP 0xC8
```

**void initialize** (byte p\_cs, byte p\_si, byte p\_clk, byte p\_a0, byte p\_res, byte type);

**void clear** (void);

**void contrast** (byte contr);

**void view** (byte direction);

**void string** (byte column, byte page, const byte \*font\_adress, const char \*str);

Printing and typographical errors reserved. ELECTRONIC ASSEMBLY reserves the right to change specifications without prior notice.

ELECTRONIC ASSEMBLY \* Zeppelinstr.19 \* D-82205 Gilching \* Phone: +49-8105-778090 \* [www.lcd-module.de](http://www.lcd-module.de) \* [info@lcd-module.de](mailto:info@lcd-module.de)

# Arduino meets EA DOGS-Graphic series

with UC1701 controller

```
void rectangle (byte start_column, byte start_page, byte end_column, byte end_page, byte  
pattern);
```

```
void picture (byte column, byte page, const byte *pic_adress);
```

## 2.1. initialize

**Name:** void initialize (byte p\_cs, byte p\_si, byte p\_clk, byte p\_a0, byte p\_res, byte type);

**Vars:** CS-Pin; MOSI-Pin; SCK-PIN; A0-Pin (data or command), Reset-Pin, 1=EA DOGS102-6

**Description:** The function inits the SPI. You can select weather the Arduino uses a software SPI (p\_si and p\_clk are different and name the port) or you can select hardware SPI by setting p\_si=p\_clk=0. Second the function inits the EA DOG-Display depending on the given display type (You can use a define **DOGS102**). It clears the screen.

**Example:** Initialize EA DOGS102-6 using Hardware SPI:

```
DOG.initialize(10,0,0,9,4,DOGS102);
```

*SS = 10, 0,0= use Hardware SPI, 9 = A0, 4=Reset, EA DOGS102-6(=102x64 dots)*

Initialize EA DOGS102-6 using Software SPI

```
DOG.initialize(10,12,13,9,4,DOGS102);
```

*SS = 10, MOSI = 12, SCK = 13 use Software SPI, 9 = RS, 4=Reset, EA DOGS102-6 (=102x64 dots)*

## 2.2. clear

**Name:** void clear (void);

**Vars:** ---

**Description:** The function clears the entire display

**Example:**

```
DOG.clear();
```

## 2.3. contrast

**Name:** void contrast (byte contr);

**Vars:** contrast (0..63)

**Description:** The function sets the contrast of the display. The function “initialize(..)” sets the default contrast. Please change it to your favorite look and feel.

**Example:** Sets the contrast value to 25

```
DOG.contrast(25);
```

## 2.4. view

**Name:** void view (byte direction);

**Vars:** VIEW\_BOTTOM = 0xC0 / VIEW\_TOP = 0xC8

**Description:** The function sets the viewing direction and clears the screen, as old content is not useable any more.

**Example:** Set Bottom view (default)

```
DOG.view(VIEW_BOTTOM);
```

## 2.5. string

**Name:** void string (byte column, byte page, const byte \*font\_adress, const char \*str);

**Vars:** column (0..127 / 0..132), page (0..5 / 0..4), ptr to font data, ptr to 0 terminated string

**Description:** The function writes a string with selected font to a specific position. Fonts that are included in the library are: **font\_6x8**, **font\_8x8**, **font\_8x16**, **font\_16x32nums**. The font needs to be a 'FV' font. You can generate it with ELECTRONIC ASSEMBLY's FontEditor (EA USBSTICKFONT).

**Example:** “Hello World” at 5<sup>th</sup> column, 2<sup>nd</sup> row with big characters (8x16)

```
DOG.string(6, 1, font_8x16, “Hello World”);
```

## 2.6. rectangle

**Name:** void rectangle (byte start\_column, byte start\_page, byte end\_column, byte



end\_page, byte pattern);

**Vars:** start column (0..101), start page (0..7), end column (0..101), end page (0..7), pattern-byte to fill area

**Description:** Draws a filled rectangle on the screen. The filling is given through the pattern byte

**Example:** Draw horizontal stripes on the screen

```
DOG.rectangle(0,0,101,7,0x55);
```

## 2.7. picture

**Name:** void picture (byte column, byte page, const byte \*pic\_adress);

**Vars:** upper left corner: column (0..101) and page (0..7), ptr to picture data

**Description:** The function shows a picture. The picture needs to be 'BLV' formatted. You can generate it with ELECTRONIC ASSEMBLY's BitMapEdit, which is available in the EA LCD-Tools. You can download them from <http://www.lcd-module.com/support.html>. We have included two demo pictures, ea\_logo and ea\_logo\_small.

**Example:** show EA logo in the top left corner

```
DOG.picture(0,0, ea_logo);
```

## 2.8. Backlight

For driving the backlight you should use the Arduino internal functions, like digitalWrite(port pin , HIGH/LOW) or analogWrite(port pin, 0..255). Where digitalWrite can be used for on (HIGH) and off (LOW) and analogWrite fades the backlight from off (0) to on (255).

# Arduino meets EA DOGS-Graphic series

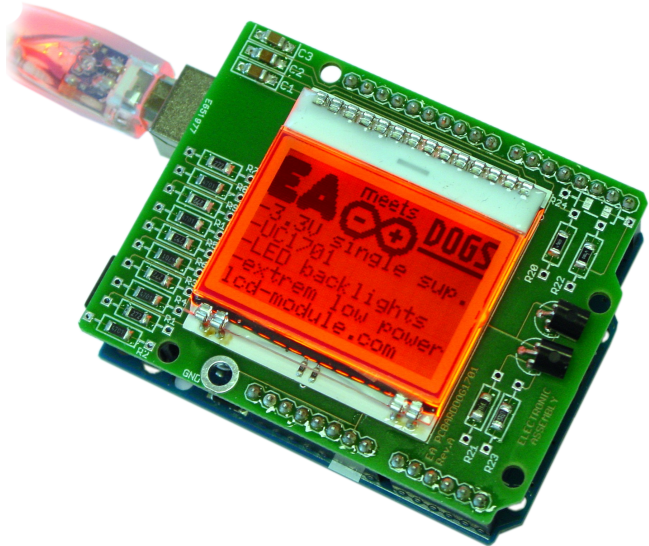
with UC1701 controller

**ELECTRONIC  
ASSEMBLY**

*new display design*

## 3. The Library in action

Please feel free to explore some pictures of our modules working with the Arduino library the shield "EA PCBARDDOG1701" together with our EA DOG-Graphic series displays



EA DOGS102W-6 + EA LED39X41-GR



EA DOGS102W-6 + EA LED39X41-W