

## Arduino meets EA DOG-Graphic series with ST7565R controller



This little project describes the connection between an Arduino-Board and a ELECTRONIC ASSEMBLY DOG-Display with ST7565R controller, like the EA DOGM132-5, EA DOGM128-6, EA DOGL128-6.

For more details, you will find the display's datasheets under <http://www.lcd-module.com/datasheets.html> and the controller's under <http://www.lcd-module.de/eng/pdf/zubehoer/st7565r.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 DOGM132-5 with monochrome backlight.....	4
1.5.2.EA DOGM132-5 with RGB backlight.....	5
1.5.3.EA DOGM128-6 / EA DOGL128-6 with monochrome backlight.....	5
1.5.4.EA DOGM128-6 / EA DOGL128-6 with RGB backlight.....	6
2.Software library classname: dog_7565R.....	7
2.1.initialize.....	8
2.2.clear.....	8
2.3.contrast.....	9
2.4.view.....	9
2.5.string.....	9
2.6.rectangle.....	10
2.7.picture.....	10
2.8.Backlight.....	10
3.The Library in action.....	11

## 1. Hardware

### 1.1. Display

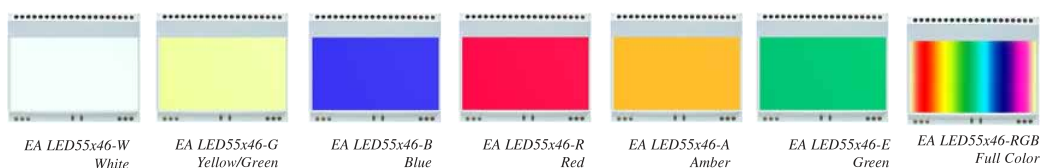
We have 5 different display technologies in the DOG Series to match every area of application:

display type	technology	optional backlight	readability	display color non backlighted	display color with backlighted	recommended backlight color
	FSTN pos. transmissive	it's fine with and without backlight	readable even without backlight	black on white	black on backlight color	white, blue, green, RGB
	STN pos. yellow/green transmissive	backlight unit required	readable even without backlight	dark green on yellow/green	black on yellow/green or amber	yellow/green, amber
	STN neg. blue transmissive	usage only with backlight	---	---	backlight color on blue background	white, yellow/green
	FSTN neg. transmissive	usage only with backlight	---	---	backlight color on black background	white, green, RGB
	STN pos. yellow/green reflective	no backlight possible	finest readable without backlight	dark green on yellow/green	---	---

Depending on your area of application you can choose between various sizes. With the ST7565R controller we have 55 x 31 mm (132 x 32 dots), 55 x 46 mm (128 x 64 dots) and 68 x 51 mm (128 x 64 dots).

### 1.2. Backlight

We have 7 different backlight modules, 6 monochrome versions and RGB. Depending on your choice you have to assemble different resistors and transistors. Please refer to the part list.



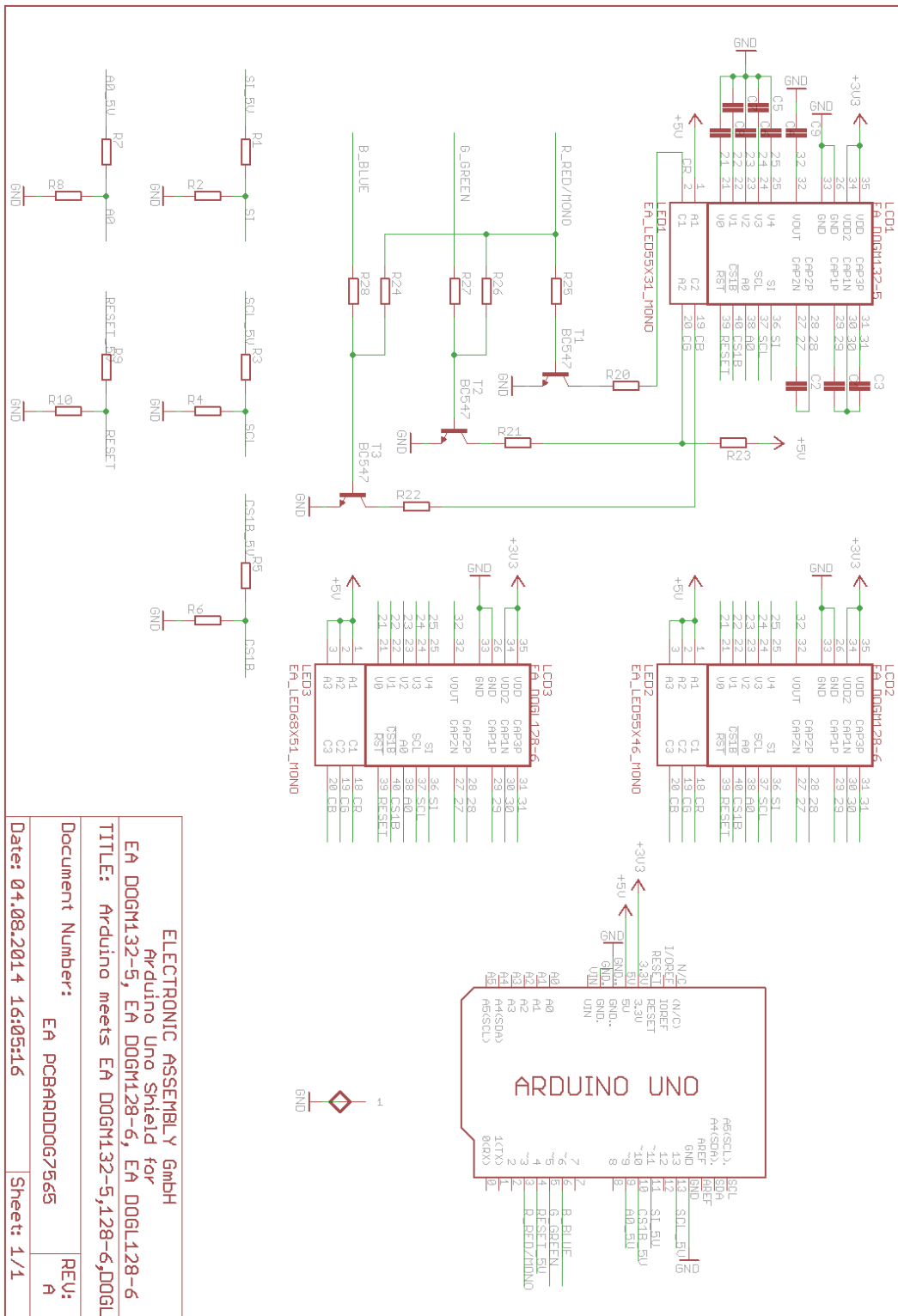
# Arduino meets EA DOG-Graphic series

with ST7565R controller

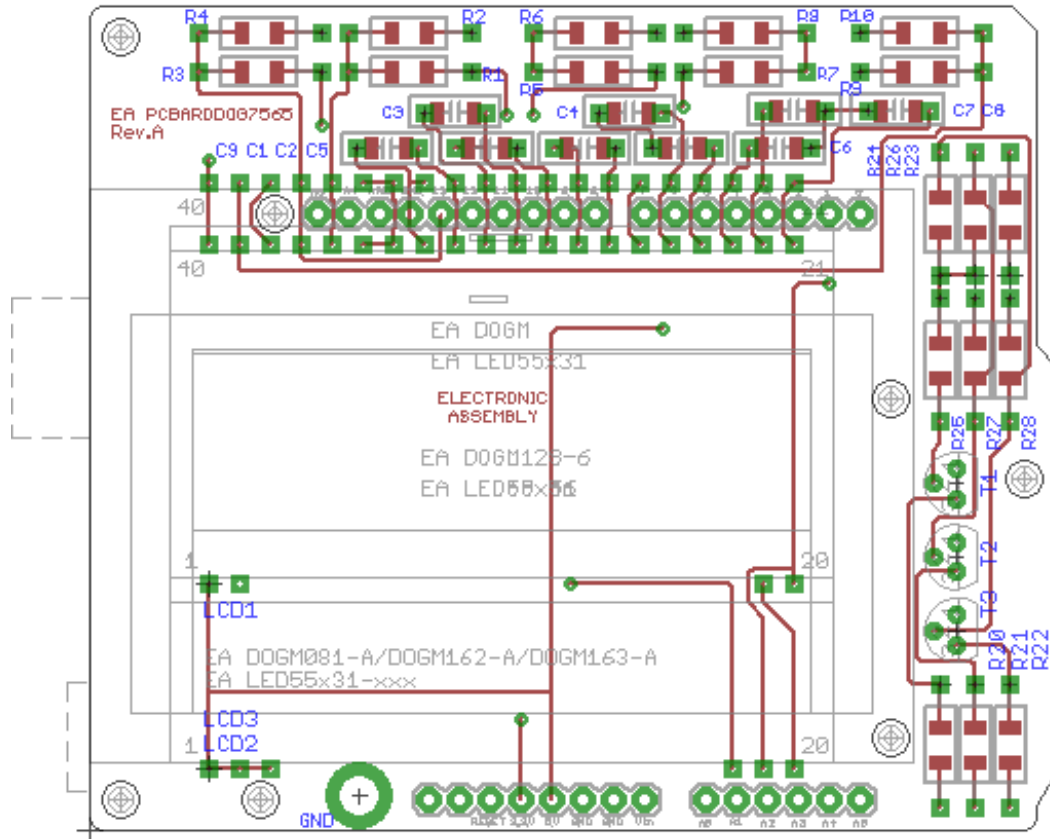
**ELECTRONIC ASSEMBLY**

*new display design*

## 1.3. Schematic



## 1.4. PCB



## 1.5. Bill of Materials

### 1.5.1. EA DOGM132-5 with monochrome backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C2, C3, C4, C5, C6, C7, C8, C9	1 $\mu$ F	80-ARR07D105KGS	X7R-G1206 1,0/50
R23	0 $\Omega$	299-0-RC	SMD 1/4W 0,0
R24, R25	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
R20, R22	Yellow/Green BL: 26 $\Omega$ EA LED55X31-G	271-36-RC	SMD 1/4W 36

## Arduino meets EA DOG-Graphic series

with ST7565R controller

	White BL: 30 Ω EA LED55X31-W	594-5083NW30R00J	SMD 1/4W 30
	Blue BL: 30 Ω EA LED55X31-B	594-5083NW30R00J	SMD 1/4W 30
	Amber BL: 33 Ω EA LED55X31-A	594-5083NW33R00J	SMD 1/4W 33
	Red BL: 36 Ω EA LED55X31-R	594-5093NW36R00J	SMD 1/4W 36
T1, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGM132X-5	790-EADOGM132B5	EA DOGM132B-5
LED1	EA LED55X31-x	790-EALED55X31W	EA LED55X31-W

### 1.5.2. EA DOGM132-5 with RGB backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C2, C3, C4, C5, C6, C7, C8, C9	1 µF	80-ARR07D105KGS	X7R-G1206 1,0/50
R25, R27, R28	470 Ω	273-470-RC	SMD 1/4W 470
R1, R3, R5, R7, R9	2.7 KΩ	273-2.7K-RC	SMD 1/4W 2,7K
R2, R4, R6, R8, R10	4.7 KΩ	273-4.7K-RC	SMD 1/4W 4,7K
R20, R21, R22	24 Ω	594-5083NW24R00J	SMD 1/4W 24
T1, T2, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGM132X-5	790-EADOGM132B5	EA DOGM132B-5
LED1	EA LED55X31-RGB	790-EALED55X31RGB	EA LED55X31-RGB

### 1.5.3. EA DOGM128-6 / EA DOGL128-6 with monochrome backlight

Designator	Value	DOGM DOGL	Mouser p/n	Reichelt p/n
C1, C2, C3, C4, C5, C6, C7, C8, C9	1 µF		80-ARR07D105KGS	X7R-G1206 1,0/50
R24, R25, R26	470 Ω		273-470-RC	SMD 1/4W 470
R1, R3, R5, R7, R9	2.7 KΩ		273-2.7K-RC	SMD 1/4W 2,7K
R2, R4, R6, R8, R10	4.7 KΩ		273-4.7K-RC	SMD 1/4W 4,7K
R20, R21, R22	Yellow/Green BL: 100 Ω		594-5083NW100R0J	SMD 1/4W 100

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

# Arduino meets EA DOG-Graphic series

with ST7565R controller

T1, T2, T3	EA LED55X31-G	47 Ω	594-5073NW47R00J	SMD 1/4W 47
	White BL:	82 Ω	270-82-RC	SMD 1/4W 82
	EA LED55X31-W	39 Ω	594-5083NW39R00J	SMD 1/4W 39
	Blue BL:	68 Ω	594-5073NW68R00J	SMD 1/4W 68
	EA LED55X31-B	39 Ω	594-5083NW39R00J	SMD 1/4W 39
	Amber BL:	100 Ω	594-5083NW100R0J	SMD 1/4W 100
	EA LED55X31-A	56 Ω	271-56-RC	SMD 1/4W 56
	Red BL:	100 Ω	594-5083NW100R0J	SMD 1/4W 100
	EA LED55X31-R	51 Ω	594-5083NW51R00J	SMD 1/4W 51
	Green BL:	56 Ω	271-56-RC	SMD 1/4W 56
	EA LED55X31-E	33 Ω	594-5083NW33R00J	SMD 1/4W 33
	LCD1	BC238 (or: BC548)	512-BC548CTA	BC 548C
	LED1	EA DOGM128X-6 or EA DOGL128X-6	790-EADOGM128B6	EA DOGM128B-6
	EA LED55X46-X or EA LED68X51-X	790-EALED55X46W	EA LED55X46-W	

## 1.5.4. EA DOGM128-6 / EA DOGL128-6 with RGB backlight

Designator	Value	DOGM DOGL	Mouser p/n	Reichelt p/n
C1, C2, C3, C4, C5, C6, C7, C8, C9	1 µF		80-ARR07D105KGS	X7R-G1206 1,0/50
R25, R27, R28	470 Ω		273-470-RC	SMD 1/4W 470
R1, R3, R5, R7, R9	2.7 KΩ		273-2.7K-RC	SMD 1/4W 2,7K
R2, R4, R6, R8, R10	4.7 KΩ		273-4.7K-RC	SMD 1/4W 4,7K
R20, R21, R22	24 Ω		594-5083NW24R00J	SMD 1/4W 24
	15 Ω		94-5073NW15R00J	SMD 1/4W 15
T1, T2, T3	BC238 (or: BC548)		512-BC548CTA	BC 548C
LCD1	EA DOGM128X-6 or EA DOGL128X-6		790-EADOGM132B5	EA DOGM132B-5
LED1	EA LED55X46-X or EA LED68X51-X		790-EALED55X31RGB	EA LED55X31-RGB

## 2. Software library classname: dog\_7565R

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\_7565r.zip Your sketch should look like this:

```
#include <Arduino.h>
#include <SPI.h>
#include <dog_7565R.h>
#include <font_16x32nums.h>
#include <font_6x8.h>
#include <font_8x16.h>
#include <font_8x8.h>
#include <logo_small_BLH.h>
```

```
dog_7565R DOG;
```

```
void setup()
{
}
```

```
void loop()
{
}
```

In addition you will find examples for all EA DOG-Displays with ST7565R controller in the library folder “examples”.

These functions and definitions are provided within the library:

Classname: **dog\_7565R**

```
#define DOGM128 1
#define DOGL128 2
#define DOGM132 3
#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);
```

# Arduino meets EA DOG-Graphic series

with ST7565R controller

```
void contrast (byte contr);  
void view (byte direction);  
void string (byte column, byte page, const byte *font_adress, const char *str);  
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  
DOGM128-6 2=EA DOGL128-6 3=EA DOGM132-5

**Description:** The function inits the SPI. You can select whether 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 defines like **DOGM128**, **DOGL128**, **DOGM132**). It clears the screen.

**Example:** Initialize EA DOGM128-5 using Hardware SPI:

```
DOG.initialize(10,0,0,9,4,DOGM128);
```

*SS = 10, 0,0= use Hardware SPI, 9 = A0, 4=Reset, EA DOGM128-6(=128x64 dots)*

Initialize EA DOGL128-6 using Software SPI

```
DOG.initialize(10,12,13,9,4,DOGL128);
```

*SS = 10, MOSI = 12, SCK = 13 use Software SPI, 9 = RS, 4=Reset, EA DOGL128-6 (=128x64 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..131), page (0..7 / 0..3), 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..127 / 0..131), start page (0..7 / 0..3), end column (0..127 / 0..131), end page (0..7 / 0..3), 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(0,0,127,5,0x55);
```

## 2.7. picture

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

**Vars:** upper left corner: column (0..127 / 0..131) and page (0..7 / 0..3), 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 DOG-Graphic series

with ST7565R 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 PCBARDDOG7565" together with our EA DOG-Graphic series displays



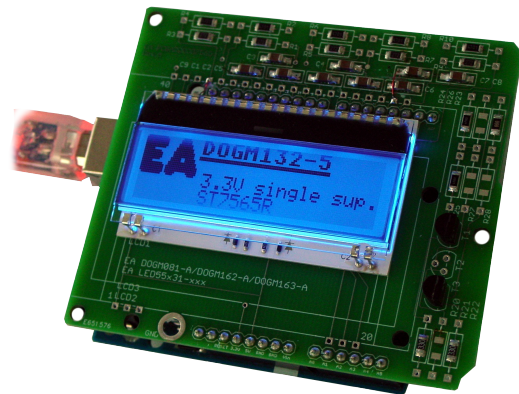
EA DOGL128W-6 + LED68X51-RGB



EA DOGM128E-6 + LED55X46-G



EA DOGM132W-5 + LED55X31-RGB



EA DOGM132W-5 + LED55X31-B