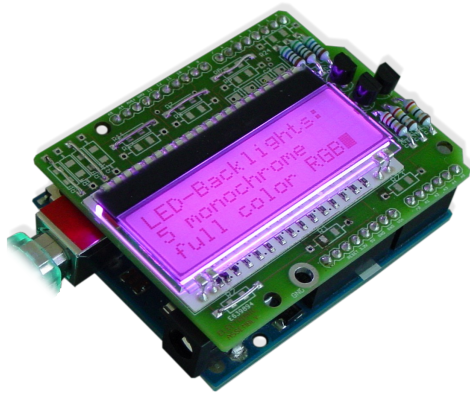


Arduino meets EA DOGM-Character series with ST7036 controller



This little project describes the connection between an Arduino-Board and a ELECTRONIC ASSEMBLY DOG-Display with ST7036 controller, like the EA DOGM081-A, EA DOGM162-A, EA DOGM163-A.

For more details, you will find the display's under <http://www.lcd-module.de/eng/pdf/doma/dogme.pdf> and the controller's under <http://www.lcd-module.de/eng/pdf/zubehoer/st7036.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
1.5.1.3.3 V Mode with monochrome backlight.....	4
1.5.2.5 V Mode with monochrome backlight.....	5
1.5.3.3.3 V Mode with RGB backlight.....	5
1.5.4.5 V Mode with RGB backlight.....	6
2.Software library classname: dogm_7036.....	7
2.1.Initialize.....	8
2.2.string.....	8
2.3.ascii.....	9
2.4.position.....	9
2.5.displ_onoff.....	9
2.6.cursor_onoff.....	10
2.7.define_char.....	10
2.8.clear_display.....	10
2.9.contrast.....	11
2.10.Backlight.....	11
3.The Library in action.....	12

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. transfective	it's fine with and without backlight	readable even without backlight	black on white	black on backlight color	white, blue, 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, RGB
	STN pos. yellow/green reflective	no backlight possible	finest readable without backlight	dark green on yellow/green	---	---

Depending on the content you want to show and the needed character size, we have 3 different solutions of lines and characters per line:

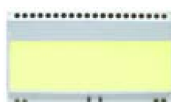


1.2. Backlight

We have 6 different backlight modules, 5 monochrome versions and 1 RGB. Depending on your choice you have to assemble different resistors and transistors. Please refer to the part list.



EA LED55x31-W
White



EA LED55x31-G
Yellow/Green



EA LED55x31-B
Blue



EA LED55x31-R
Red



EA LED55x31-A
Amber

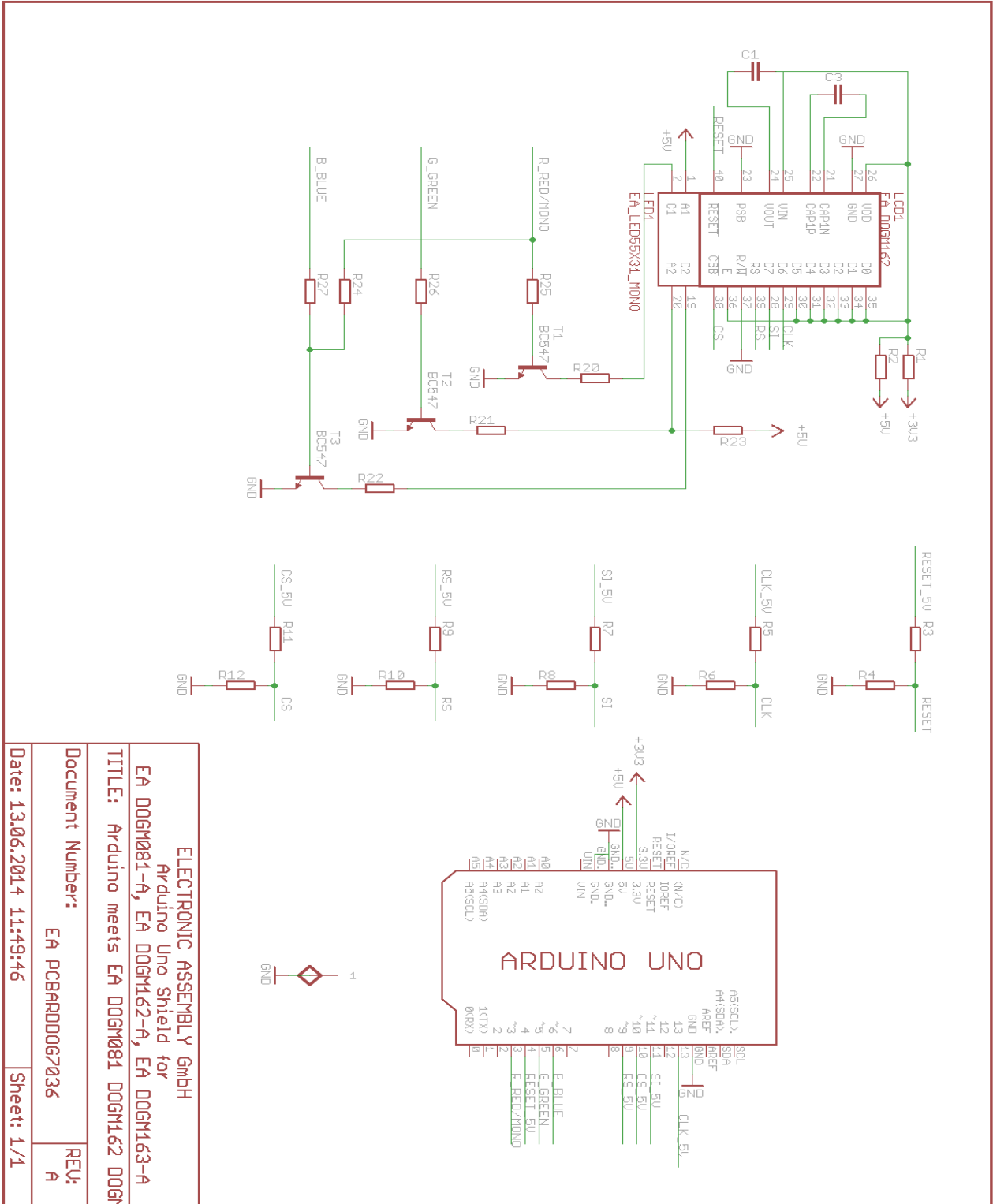


EA LED55x31-RGB
Full Color

Arduino meets EA DOGM-Character series

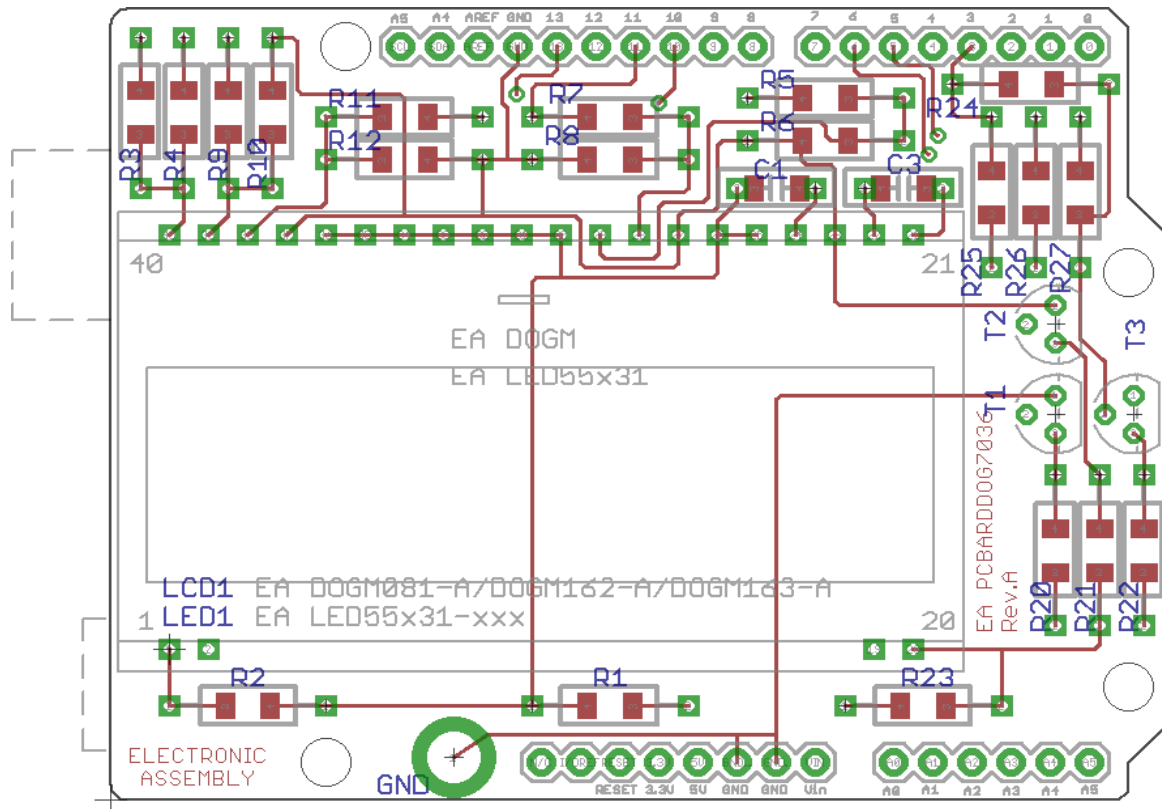
with ST7036 controller

1.3. Schematic



ELECTRONIC ASSEMBLY GmbH	
Arduino Uno Shield for	
EA DOGM081-A, EA DOGM162-A, EA DOGM163-A	
TITLE: Arduino meets EA DOGM081 DOGM162 DOGM163 Rev_A	
Document Number:	EA PCBARD0067036
Date: 13.06.2014 11:49:46	Sheet: 1/1
REV:	A

1.4. PCB



1.5. Bill of Materials

1.5.1. 3.3 V Mode with monochrome backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C3	1 μ F	80-ARR07D105KGS	X7R-G1206 1,0/50
R1, R23	0 Ω	299-0-RC	SMD 1/4W 0,0
R24, R25	470 Ω	273-470-RC	SMD 1/4W 470
R3, R5, R7, R9, R11	2.7 K Ω	273-2.7K-RC	SMD 1/4W 2,7K
R4, R6, R8, R10, R12	4.7 K Ω	273-4.7K-RC	SMD 1/4W 4,7K
R20, R22	Yellow/Green BL: 35R EA LED55X31-G	71-CCF02-F-35	SMD 1/4W 36
	White BL: 30R EA LED55X31-W	273-30-RC	SMD 1/4W 30

Arduino meets EA DOGM-Character series

with ST7036 controller

	Blue BL: 30R EA LED55X31-B	273-30-RC	SMD 1/4W 30
	Amber BL: 32R EA LED55X31-A	71-RN60E-B-32	SMD 1/4W 33
	Red BL: 36R EA LED55X31-R	271-36-RC	SMD 1/4W 36
T1, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGMxxxx-A	790-EADOGM162BA	EA DOGM162B-A
LED1	EA LED55X31-x	790-EALED55X31W	EA LED55X31-W

1.5.2. 5 V Mode with monochrome backlight

Designator	Value	Mouser p/n	Reichelt p/n
R2, R3, R5, R7, R9, R11, R23	0 Ω	299-0-RC	SMD 1/4W 0,0
R24, R25	470 Ω	273-470-RC	SMD 1/4W 470
R20, R22	Yellow/Green BL: 35R EA LED55X31-G	71-CCF02-F-35	SMD 1/4W 36
	White BL: 30R EA LED55X31-W	273-30-RC	SMD 1/4W 30
	Blue BL: 30R EA LED55X31-B	273-30-RC	SMD 1/4W 30
	Amber BL: 32R EA LED55X31-A	71-RN60E-B-32	SMD 1/4W 33
	Red BL: 36R EA LED55X31-R	271-36-RC	SMD 1/4W 36
T1, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGMxxxx-A	790-EADOGM162BA	EA DOGM162B-A
LED1	EA LED55X31-x	790-EALED55X31W	EA LED55X31-W

1.5.3. 3.3 V Mode with RGB backlight

Designator	Value	Mouser p/n	Reichelt p/n
C1, C3	1 μF	80-ARR07D105KGS	X7R-G1206 1,0/50
R1	0 Ω	299-0-RC	SMD 1/4W 0,0

Arduino meets EA DOGM-Character series

with ST7036 controller

R25, R26, R27	470 Ω	273-470-RC	SMD 1/4W 470
R3, R5, R7, R9, R11	2.7 K Ω	273-2.7K-RC	SMD 1/4W 2,7K
R4, R6, R8, R10, R12	4.7 K Ω	273-4.7K-RC	SMD 1/4W 4,7K
R20, R21, R22	24 Ω	271-24-RC	SMD 1/4W 24
T1, T2, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGMxxxx-A	790-EADOGM162WA	EA DOGM162W-A
LED1	EA LED55X31-RGB	790-EALED55X31RGB	EA LED55X31-RGB

1.5.4. 5 V Mode with RGB backlight

Designator	Value	Mouser p/n	Reichelt p/n
R2, R3, R5, R7, R9, R11	0 Ω	299-0-RC	SMD 1/4W 0,0
R25, R26, R27	470 Ω	273-470-RC	SMD 1/4W 470
R20, R21, R22	24 Ω	271-24-RC	SMD 1/4W 24
T1, T2, T3	BC238 (or: BC548)	512-BC548CTA	BC 548C
LCD1	EA DOGMxxxx-A	790-EADOGM162WA	EA DOGM162W-A
LED1	EA LED55X31-RGB	790-EALED55X31RGB	EA LED55X31-RGB

2. Software library classname: dogm_7036

The library provides all necessary functions for using a EA DOGM-Textdisplay.

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

Please add dogm_7036.zip Your sketch should look like this:

```
#include <Arduino.h>
#include <SPI.h>
#include <dogm_7036.h>

dogm_7036 DOG;

void setup()
{

}

void loop()
{
  delay(2000);
}
```

In addition you will find examples for all EA DOG-Displays with ST7036 controller in the library folder “examples”.

These functions and definitions are provided within the library:

Classname: **dogm_7036**

```
#define DOGM081 1
#define DOGM162 2
#define DOGM163 3
```

```
void initialize      (byte p_cs, byte p_si, byte p_clk, byte rs, boolean sup_5V, byte lines);
void string          (const char *str);
void ascii           (char character);
void position        (byte column, byte line);
void displ_onoff     (boolean on);
void cursor_onoff    (boolean on);
void define_char     (byte mem_adress, byte *dat);
void clear_display   (void);
void contrast        (byte contr);
```

2.1. Initialize

Name: void initialize (byte p_cs, byte p_si, byte p_clk, byte p_rs, byte p_res, boolean sup_5V, byte lines);

Vars: CS-Pin; MOSI-Pin; SCK-PIN; RS-Pin (data or command), Reset-Pin, 5V=true 3.3V=false, 1=EA DOGM081-A 2=EA DOGM162-A 3=EA DOGM163-A

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 **DOGM081**, **DOGM162**, **DOGM163**). It clears the screen and sets the cursor to the top left position. Cursor is not flashing.

Example: Initialize EA DOGM162-A with 3.3V using Hardware SPI:

```
DOG.initialize(10,0,0,9,4,0,DOGM162);
```

SS = 10, 0,0= use Hardware SPI, 9 = RS, 4=Reset, 0 = 3.3V, EA DOGM162-A (=2 lines)

Initialize EA DOGM081-A with 5V using Software SPI

```
DOG.initialize(10,12,13,9,4,1,DOGM081);
```

SS = 10, MOSI = 12, SCK = 13 use Software SPI, 9 = RS, 4=Reset, 1 = 5V, EA DOGM081-A (=1 line)

2.2. string

Name: void string (const char *str);

Vars: char array with closing 0 (String)

Description: The function shows the given string on the actual position

Example: Show "Hello World" on display

```
DOG.string("Hello World");
```

Show array on display (Hello)

```
const char array[] = {'H', 'e', 'l', 'l', 'o', 0x00};
```

```
DOG.string(char_array);
```


2.3. ascii

Name: void ascii (char character);

Vars: one byte

Description: The function shows the given character on the actual position

Example: Show '1' on display

```
DOG.ascii('1');
```

Show array on display '1'

```
DOG.ascii(0x31);
```

2.4. position

Name: void position (byte column, byte line);

Vars: column = x-postion (1..16), line = y-postion (1..3)

Description: The function sets the cursor at the given position, the left upper corner is 1,1

Example: Set cursor to 2nd line, 3rd character

```
DOG.position(3,2);
```

2.5. displ_onoff

Name: void displ_onoff (boolean on);

Vars: on = true/false

Description: The function turns the display on (=true), or off (=false)

Example: Turn display on

```
DOG.displ_onoff(TRUE);
```

2.6. cursor_onoff

Name: void cursor_onoff (boolean on);

Vars: on = true/false

Description: The function turns cursor blinking on (=true), or off (=false)

Example: Turn cursor blinking off

```
DOG.cursor_onoff(FALSE);
```

2.7. define_char

Name: void define_char (byte mem_adress , byte *dat);

Vars: CGRAM-Adress of char (0..7), Array of data bytes

Description: The function defines an own character pattern, maximum number of own defined characters is 8

Example: Store ↓ on position 2 in ram and show it

```
byte arrow_down[8] = {0x04, 0x04, 0x04, 0x04, 0x15, 0x0E, 0x04, 0x00};  
pattern for own defined character
```

```
DOG.define_char(2, arrow_down);
```

```
DOG.ascii(0x02); show that character on the actual position
```

2.8. clear_display

Name: void clear_display (void);

Vars: ---

Description: The function clears the entire display

Example: Turn cursor blinking off

```
DOG.clear_display();
```

2.9. contrast

Name: `void contrast (byte contrast);`

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

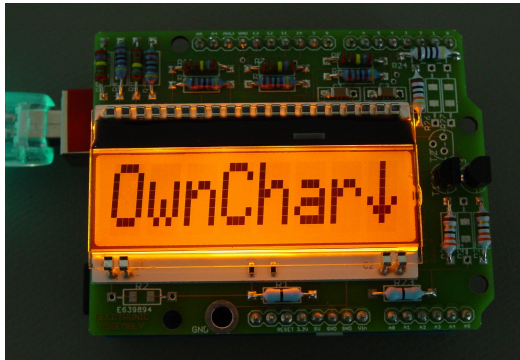
```
DOG.contrast(25);
```

2.10. Backlight

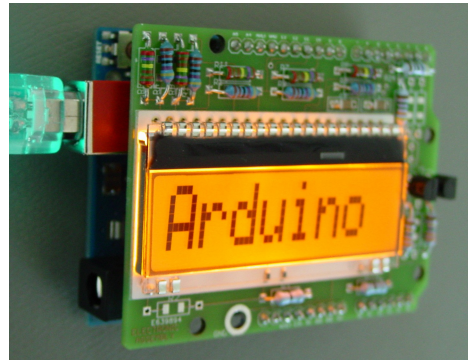
For driving the backlight you should use the Arduino internal functions, like `digitalWrite(port pin , HIGH/LOW)` or `analogWrite(port pin, 0..255)`. Where `digitalWrite` can be used for on (HIGH) and off (LOW) and `analogWrite` fades the backlight from off (0) to on (255).

3. The Library in action

Please feel free to explore some pictures of our modules working with the Arduino library the shield "EA PCBARDDOG7036" together with our EA DOGM-Textseries displays



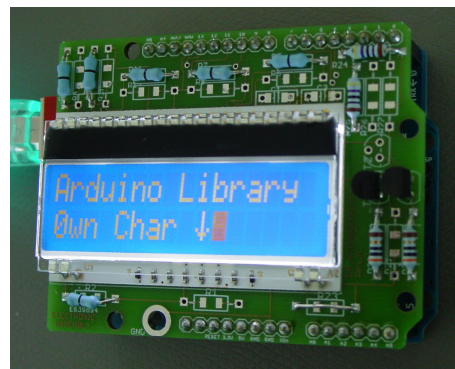
EA DOGM081W-A + EA LED55X31-A



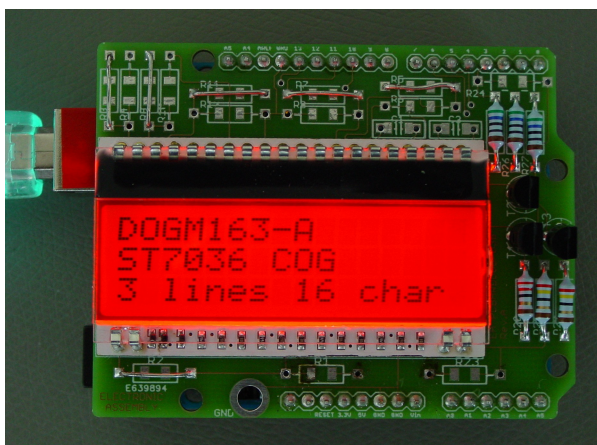
EA DOGM081W-A + EA LED55X31-A



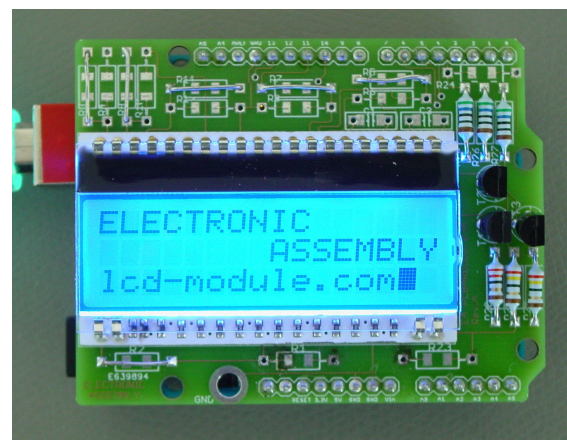
EA DOGM162B-A + EA LED55X31W



EA DOGM162B-A + EA LED55X31W



EA DOGM163W-A + EA LED55X31-RGB



EA DOGM163W-A + EA LED55X31-RGB