Electronics 1

Electronics for Interactive Media Design

Les 5

Emma Pareschi

From the last time

From class 2:
- How to count a switch
- For statement
- Array structure
- Generate sound: Buzzer
- Light sensor

Assignments:
1 - The Dice
2 - The buzzer

Today:
- Neo pixel
- Functions and libraries
- LCD screen

Light - intelligent LEDs

Red, green and blue LEDs are integrated alongside a driver chip into a tiny surface-mount package controlled through a single wire.

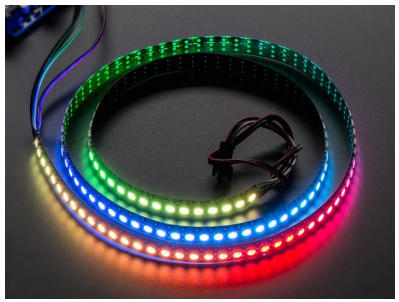
They require a microcontroller (such as Arduino) and some programming. We provide some sample code to get you started.

RGB: WS2812, WS2811
RGBW: SK6812
NeoPixel -> Adafruit

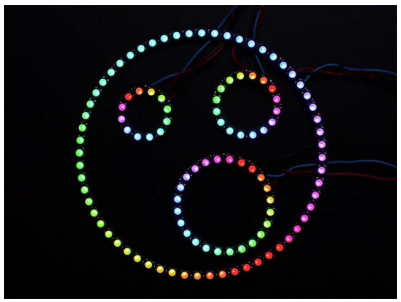
Light - intelligent LEDs

Different shapes, Form factors 

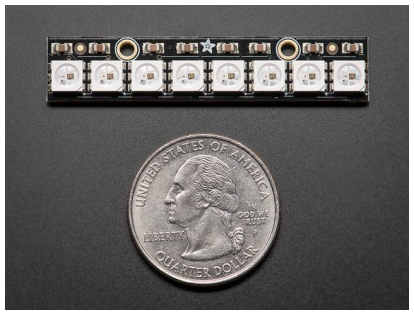
NeoPixel Strips

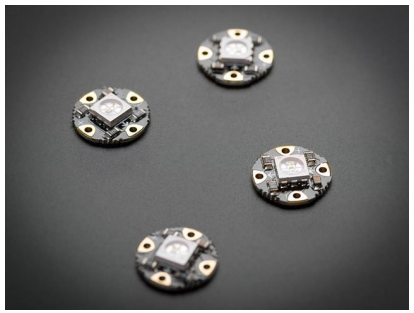


NeoPixel Rings

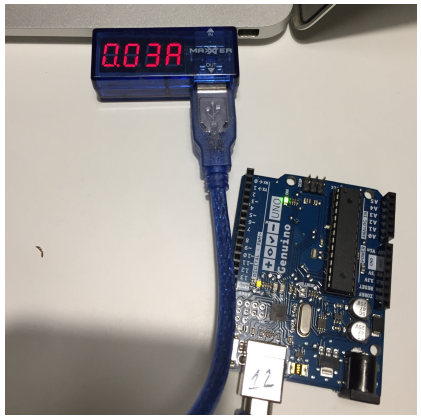


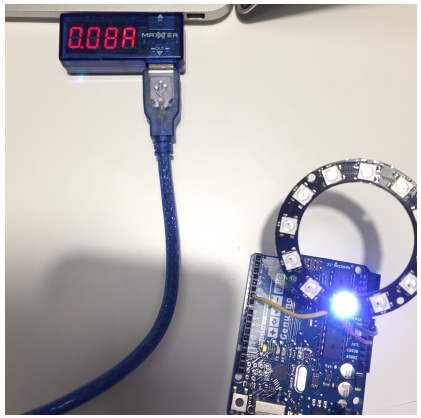
NeoPixel Matrix

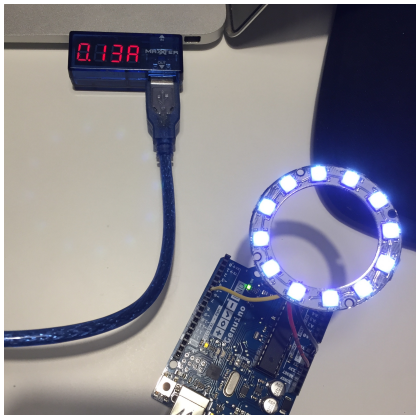




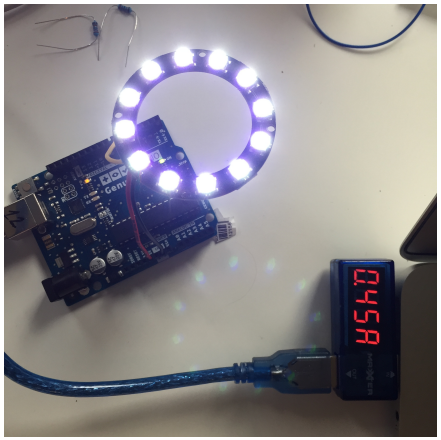
Light - intelligent LEDs







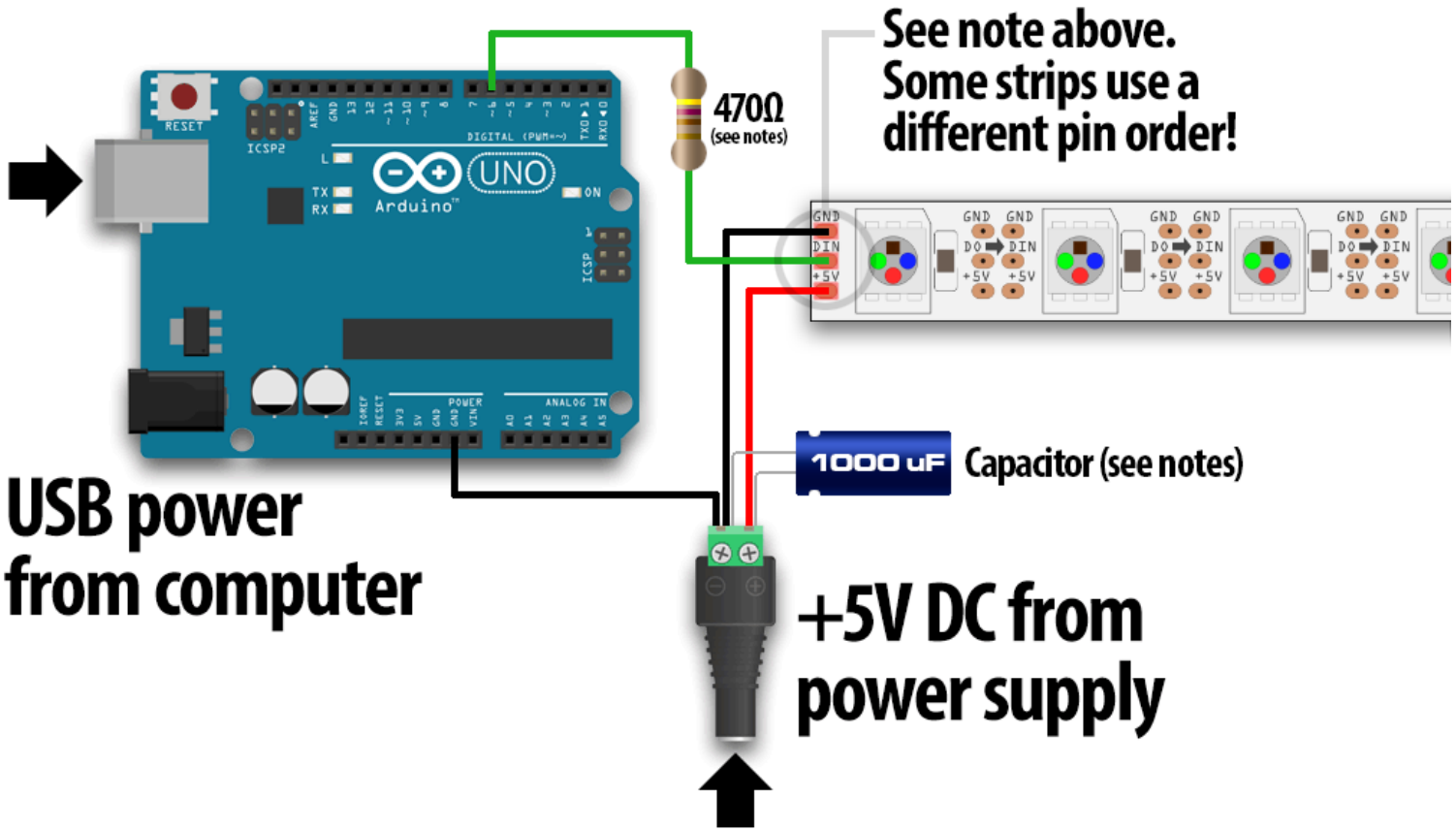
1 Neopixel: 20mA x 3 = 60mA
12 Neopixel: 60mA x 12 = 720 mA



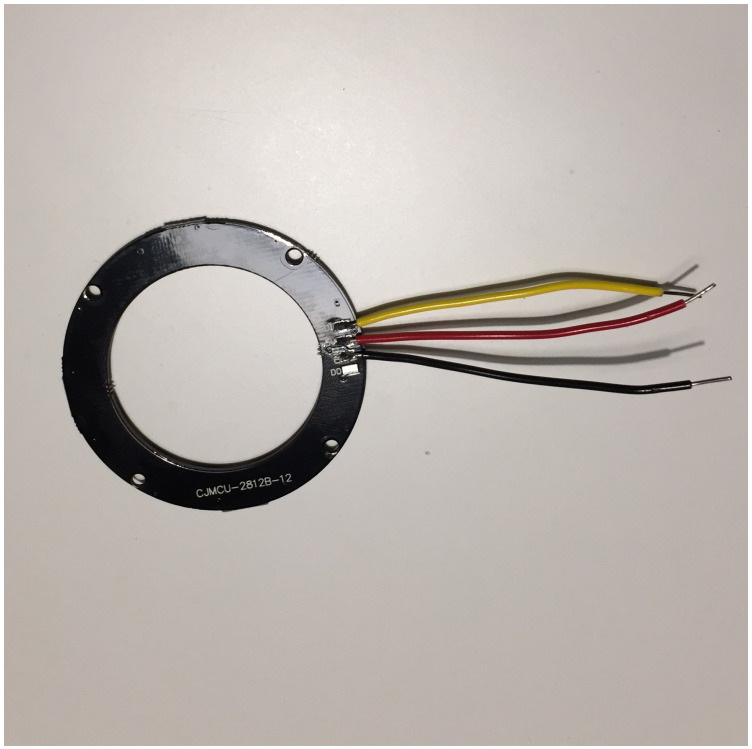
…using Arduino +5V => 0.450A

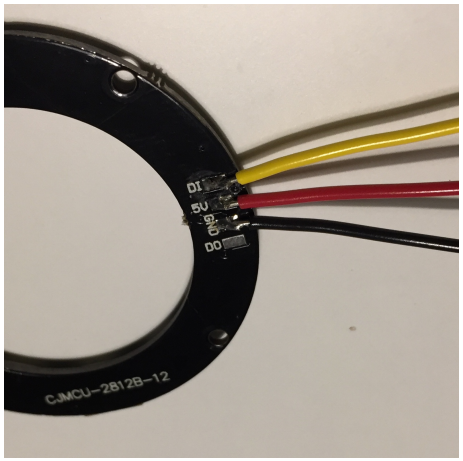
Light - intelligent LEDs

Power



Solder





DI: Yellow
5V: Red
GND: black

NeoPixel Library

NeoPixel_single

Code 00_NeoPixel_single

```
/*
 * Emma Pareschi
 *
 * Basic commands of Library Adafruit_NeoPixel
 * Using some parts from the example "strandtest"
 */

#include <Adafruit_NeoPixel.h>

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN            6

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      12

// Parameter 1 = number of pixels in strip
// Parameter 2 = Arduino pin number (most are valid)
// Parameter 3 = pixel type flags, add together as needed:
//   NEO_KHZ800  800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
//   NEO_KHZ400  400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
//   NEO_GRB     Pixels are wired for GRB bitstream (most NeoPixel products)
//   NEO_RGB     Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
//   NEO_RGBW    Pixels are wired for RGBW bitstream (NeoPixel RGBW products)
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

int delayval = 2000; // delay for half a second

void setup() {

  pixels.begin(); // This initializes the NeoPixel library.

}
```

Install Library

Create an Object Of the library NeoPixel

pixels.setBrightness(255);

NeoPixel Library

```
void loop() {

  // For a set of NeoPixels the first NeoPixel is 0, second is 1

  pixels.setPixelColor(0, 255, 255, 255);
  pixels.show();

  delay(delayval);

  pixels.setPixelColor(0, 0, 0, 0);
  pixels.show();

  delay(delayval);

}
```

Code 00_NeoPixel_single

NeoPixel Library

01_NeoPixel_single

```
 */

#include <Adafruit_NeoPixel.h>

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN            6

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      12

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_RGB + NEO_KHZ800);


uint32_t magenta = pixels.Color(255, 0, 255);
uint32_t off = pixels.Color(0, 0, 0);

int delayval = 2000; // delay for half a second

void setup() {

  pixels.begin(); // This initializes the NeoPixel library.

}

void loop() {

  pixels.setPixelColor(0, magenta);
  pixels.show();

  delay(delayval);

  pixels.setPixelColor(0, off);
  pixels.show();

  delay(delayval);

}
```

Code 01_NeoPixel_single

NeoPixel Library

02_NeoPixel_sequence

```
 * Emma Pareschi
 *
 * from example: simple (Neopixel library)
 */

#include <Adafruit_NeoPixel.h>

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN            6

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS      12

// When we setup the NeoPixel library, we tell it how many pixels, and which pin to use to send signals.
// Note that for older NeoPixel strips you might need to change the third parameter.
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_RGB + NEO_KHZ800);

int delayval = 50; // delay for half a second

uint32_t aqua = pixels.Color(0, 255, 255);  //define the color aqua
uint32_t off = pixels.Color(0, 0, 0);       // set to off when the LEDs are off

void setup() {

  pixels.begin(); // This initializes the NeoPixel library.

}
```

Code 02_NeoPixel_sequence

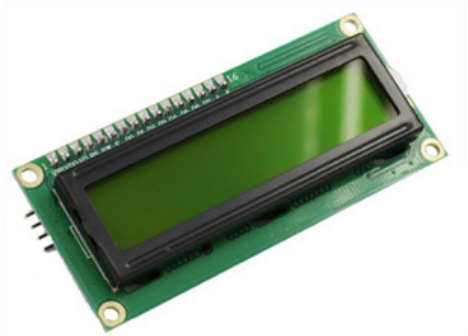
NeoPixel Library

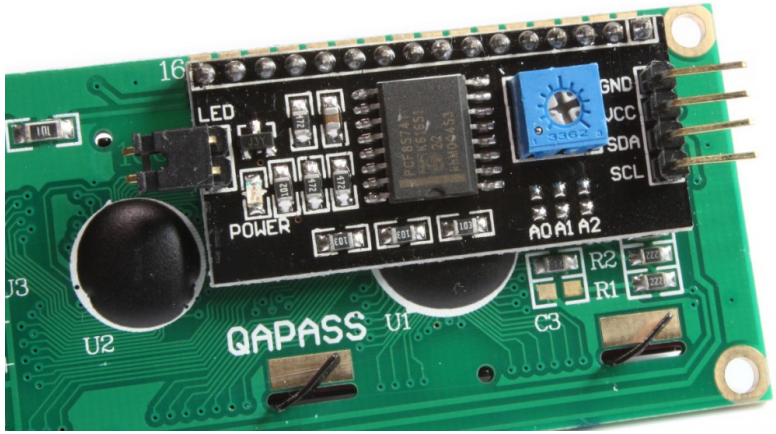
```
void loop() {

  //turn on cw, one after one, the ring.
  for(int i=0;i<NUMPIXELS;i++){

    pixels.setPixelColor(i, aqua); // aqua colour.

    pixels.show(); // This sends the updated pixel color to the hardware.

    delay(delayval); // Delay for a period of time (in milliseconds).

  }

  delay(50);

  //turn off ccw, one after one, the ring.
  for(int i = NUMPIXELS; i>=0; i--){

    // pixels.Color takes RGB values, from 0,0,0 up to 255,255,255
    pixels.setPixelColor(i, off); // turn off the ring

    pixels.show(); // This sends the updated pixel color to the hardware.

    delay(delayval); // Delay for a period of time (in milliseconds).

  }

    delay(50);
}
```

Done Saving.

Code 02_NeoPixel_sequence

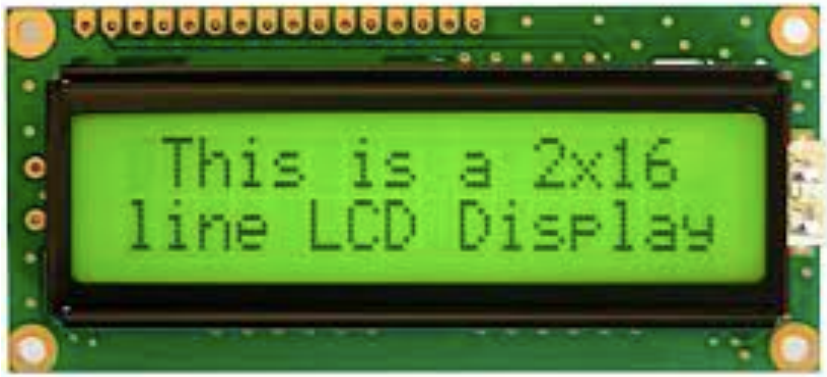
LCD SCREEN





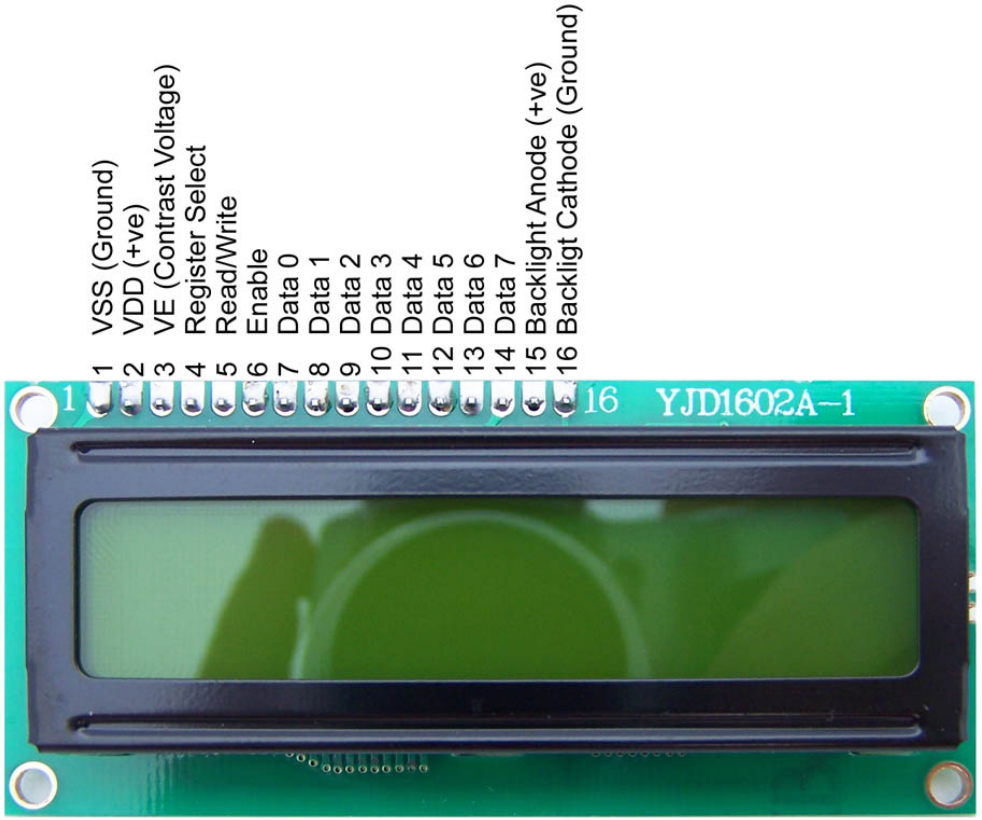
LCD SCREENS

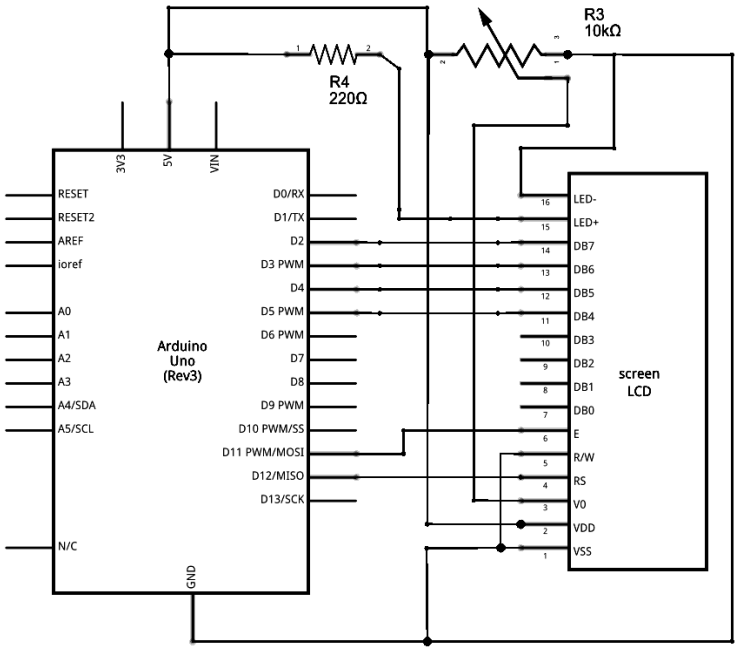
A liquid-crystal display (LCD) is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals. Liquid crystals do not emit light directly, instead using a backlight or reflector to produce images in colour or monochrome



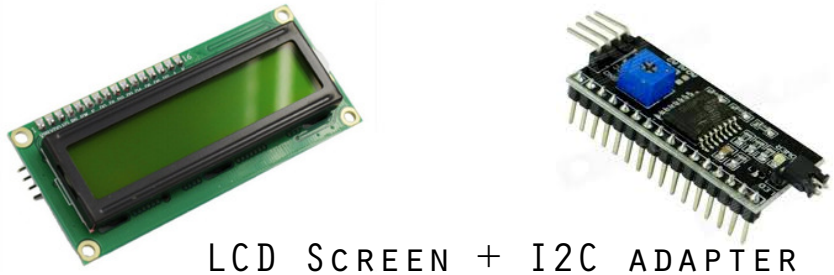
Monochrome, screen size: 8×2, 16×2 and 20×4

LCD - PINS


1 VSS (Ground)
2 VDD (+ve)
3 VE (Contrast Voltage)
4 Register Select
5 Read/Write
6 Enable
7 Data 0
8 Data 1
9 Data 2
10 Data 3
11 Data 4
12 Data 5
13 Data 6
14 Data 7
15 Backlight Anode (+ve)
16 Backligt Cathode (Ground)
1
16
YJD1602A-1


R3
10kΩ
R4
220Ω
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
Arduino
Uno
(Rev3)
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
LED-
LED+
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
E
R/W
RS
V0
VDD
VSS
screen
LCD

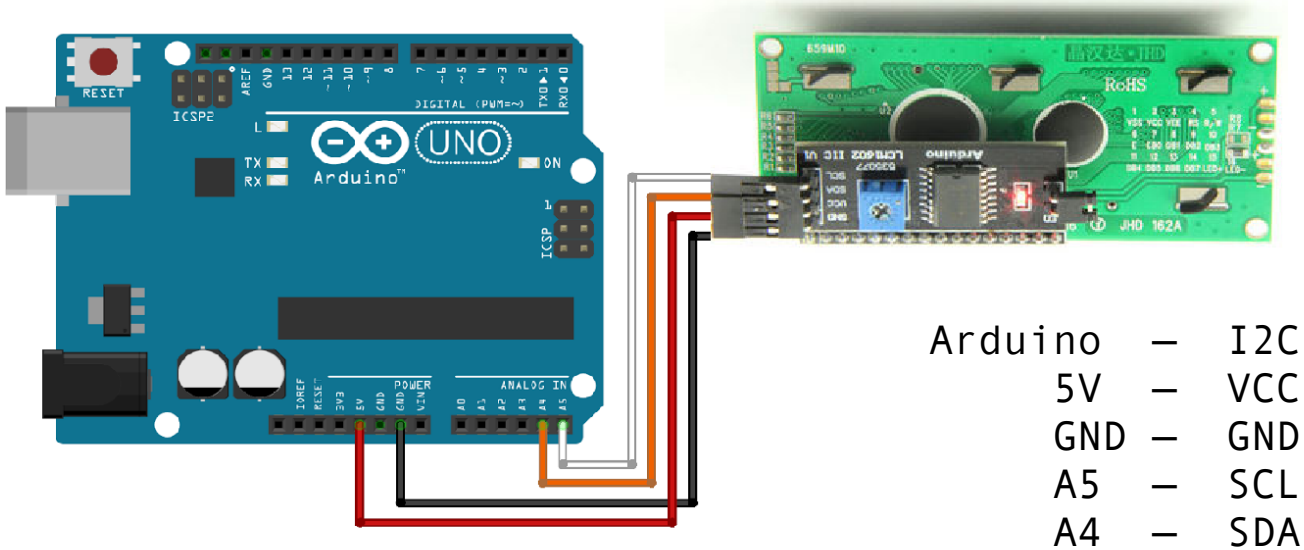
LCD WITH I2C ADAPTER



LCD SCREEN + I2C ADAPTER

Using the code “lcd_scanner”, you get the address of the adapter:

```
Scanning...
I2C device found at address 0x3F  !
done
```

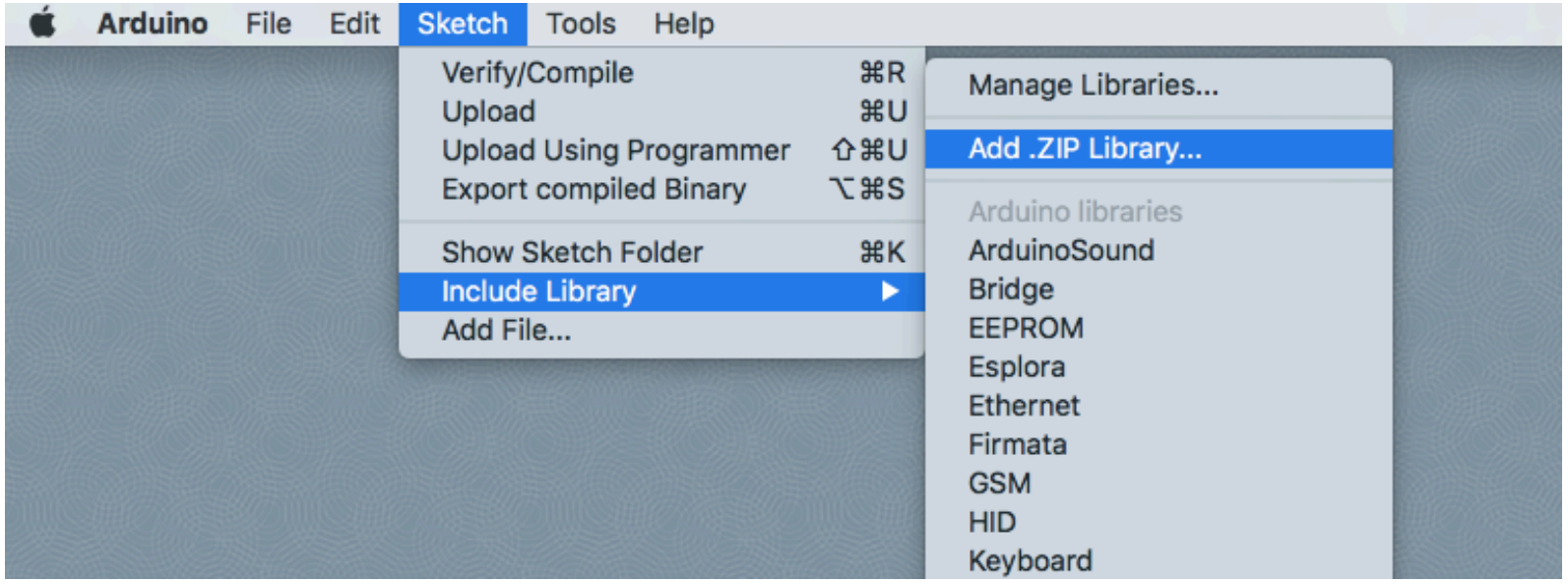


Arduino — I2C adapter
5V — VCC
GND — GND
A5 — SCL
A4 — SDA

LCD LIBRARY

Download the library Newliquidcrystal_1.3.5.zip
from https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads/

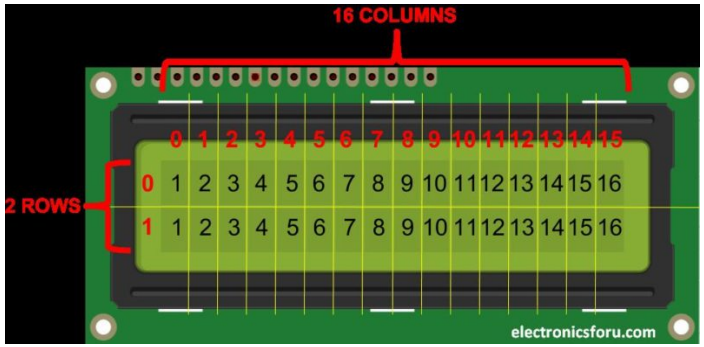
And install the library:



LCD HELLO WORLD!

lcd_00

```
/*
 * Emma Pareschi, Nov 2018
 *
 * hello world sketch for LCD
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x3F for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  // Set the LCD I2C address

void setup()
{
  // initialize the LCD
  lcd.begin(12,2);

  // Turn on the blacklight and print a message.
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Hello, world!");
}

void loop()
{
  // Do nothing here...
}
```



LCD Library

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);  create an object lcd
```

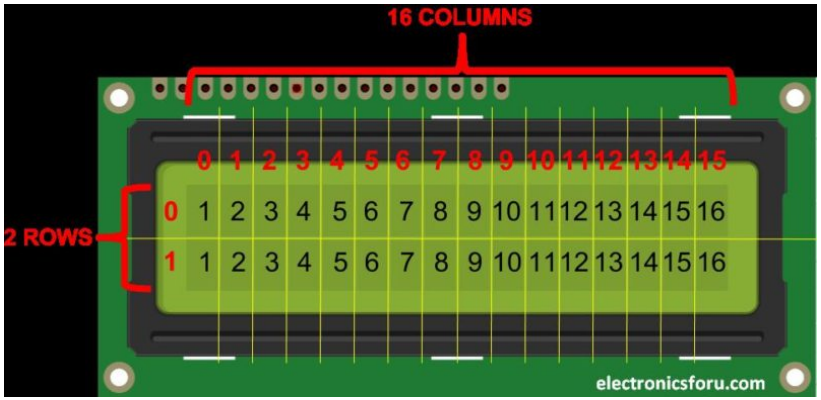
```
lcd.begin(16,2);     initialise the object lcd
lcd.backlight();     set the backlight
lcd.setCursor(column, row);     move the cursor the in the wanted location
```

```
lcd.write("text");     to display text or variables
lcd.print("text");     to display custom char or special symbol
```

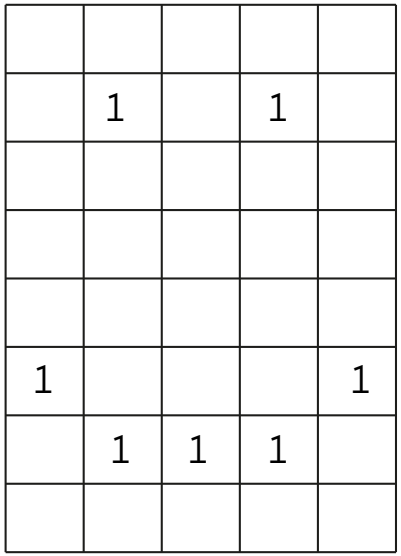
```
.blink();  //  .noBlink();
.cursor();  //  .noCursor();
```

```
.display();  //  .noDisplay();
.clear();
.home();
```

```
.scrollDisplayRight();
.scrollDisplayLeft();
```



LCD CUSTOM CHAR



```
uint8_t happy[8] = {
B00000,
B01010,
B00000,
B00000,
B00000,
B10001,
B01110,
B00000
};
```

Define the char

```
lcd.createChar(1, happy);
```

Create a char called 1
That contains Happy.

```
lcd.write(1);
```

Display char 1

LCD SPECIAL SYMBOLS

HTTPS://WWW.SPARKFUN.COM/DATASHEETS/LCD/HD44780.PDF

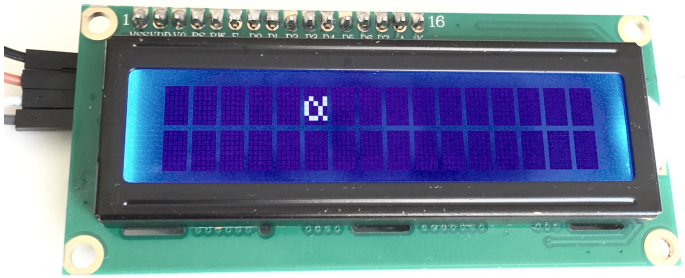
Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits / Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)			!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)			"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)			#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)			$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)			%	5	E	U	e	u			・	オ	ナ	ユ	σ	ü
xxxx0110	(7)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)			'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)			(	8	H	X	h	x			ィ	ク	ネ	リ	ｊ	x̄
xxxx1001	(2)			)	9	I	Y	i	y			ゥ	ケ	ノ	ル	ｰ	y
xxxx1010	(3)			*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)			+	;	K	[	k	{			ォ	サ	ヒ	ロ	×	万
xxxx1100	(5)			,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)			−	=	M	]	m	}			ュ	ス	ヘ	ン	ŧ	÷
xxxx1110	(7)			.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)			/	?	O	_	o	←			ッ	ソ	マ	°	ö	

Note: The user can specify any pattern for character-generator RAM.

B 1110 0000

lcd.write(B11100000);



Assignment

Think about the semester project that you will present at the individual and collective assessment.

The semester project consists of an interactive device with at least one input device and at least one output device.

Output devices:
- LCD screen
- Neo Pixel
- Buzzer

Input devices:
- light sensor
- push button
- potentiometer
- ——————
- Water sensor
- Touch sensor
- Reed switch
- Force sensor
- Distance sensor
- Tilt sensor
- Motion sensor

Assignment

The name of the semester project will be: make it interactive.
Select an object that originally is not interactive and transform it.
Examples:

- a pillow that doesn't allow you to sit because it generates an annoying sound.
- A cereal box that actually create a rainbow light when you open it.
- A vase with a plant that asks for hugs.
- A shoe that behave as a door alarm.
- A paint/frame that is a dice

Answer the following questions:

- what object will you use?
- What will the object do? What does it trigger? What is the generated feedback?
- Which sensor will you use?
- Which output will you use?
- How the sensor will trigger the output device?

Assignment

Start the development of the project:

- input device:
    - Design the schematic for the sensor
    - Make the circuit of breadboard
    - Write and test the code to read the sensor and print on serial monitor the values

- output device:
    - Design the schematic for the output device
    - Make the circuit on breadboard
    - Write and test the code to control the output device.