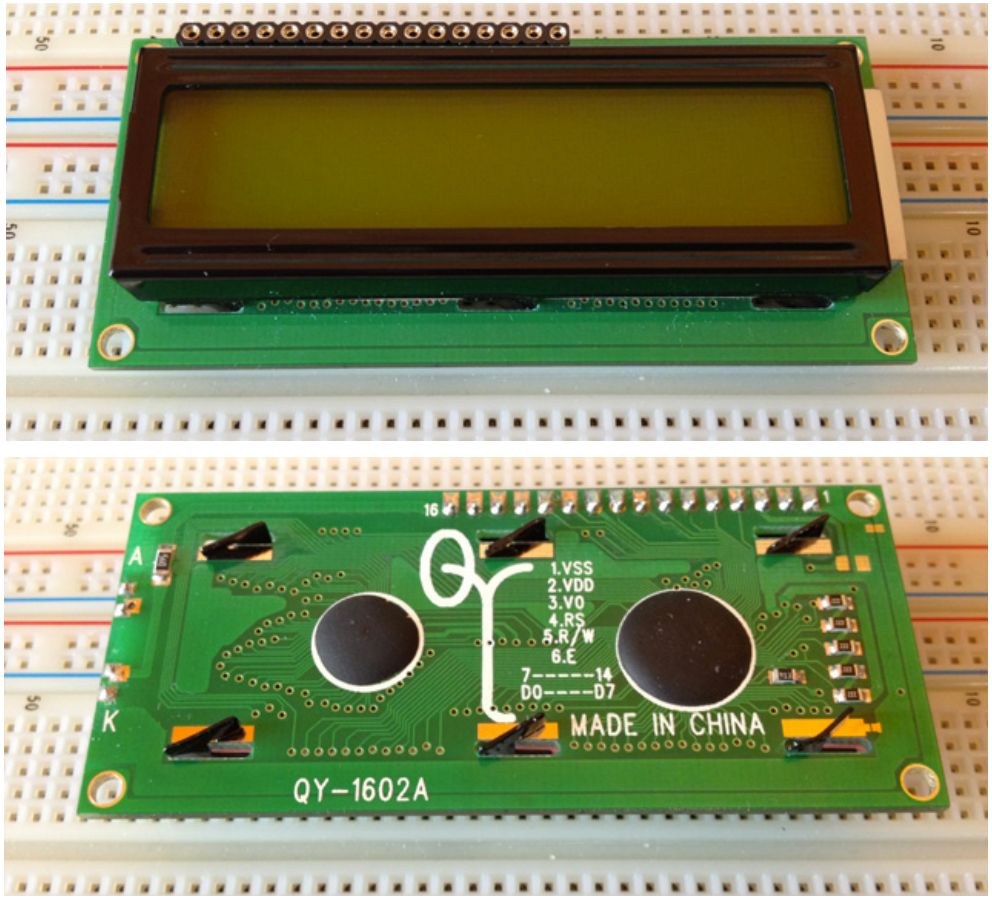
Proogramming with arduino

Fifth Program

LCD Monitor

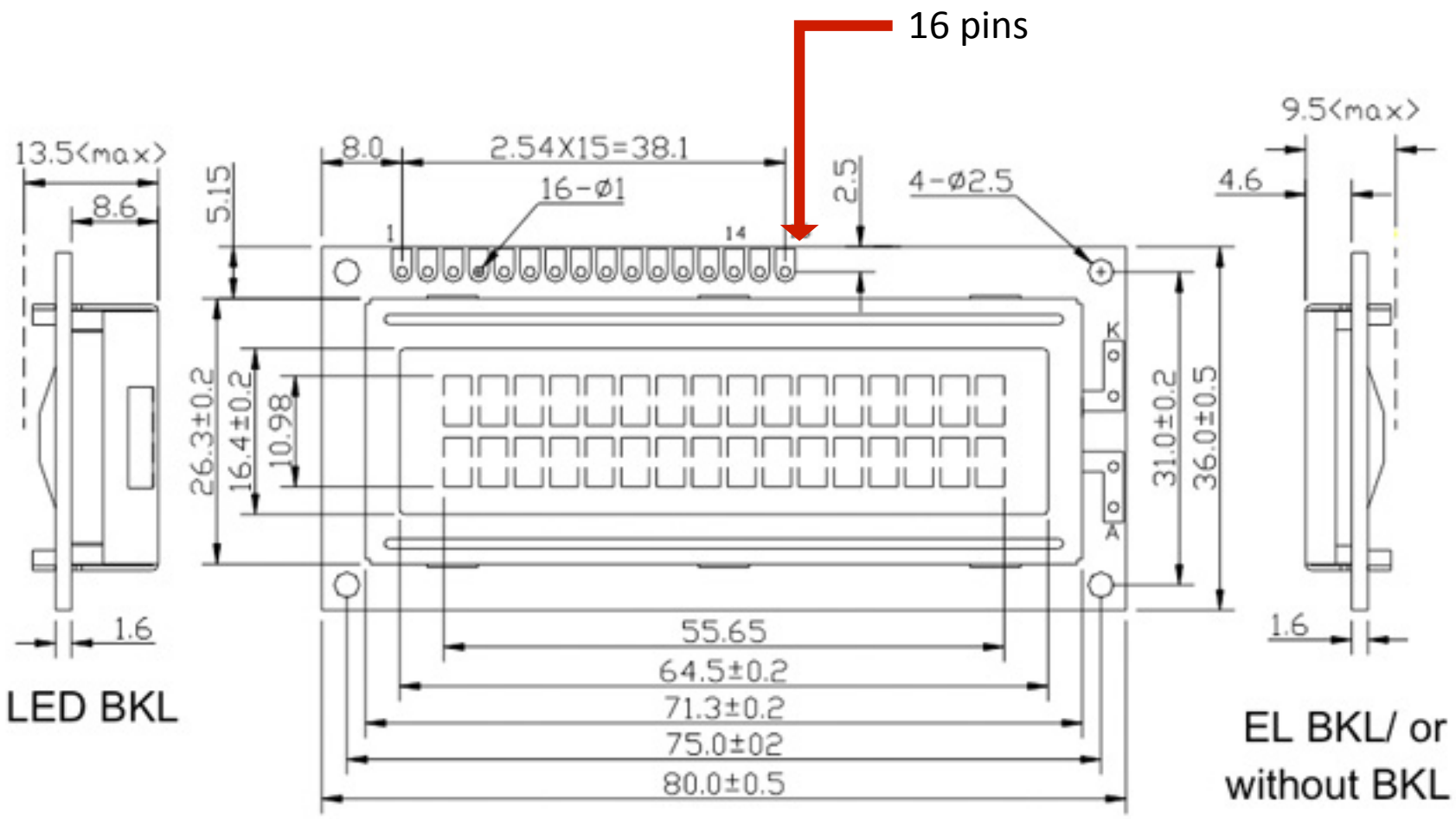
Monitor LCD 16(columns)x2(rows)



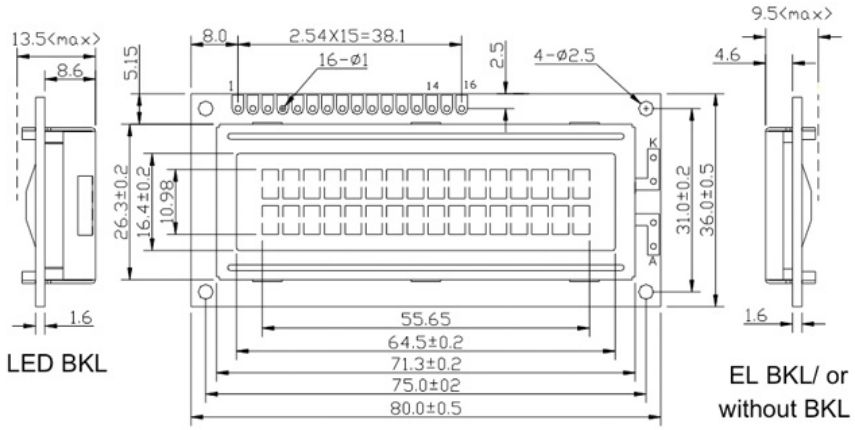
- Library: LiquidCrystal.h
- You can communicate with 4 bit (4 lines) or 8 bit (8 lines)
- Total lines: 4 or 8 + 2 controller lines:
  - Register Select (RS)
  - Enable (E)
  - + 1 optional line Read/ write (RW)

https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf

From the datasheet



Pin description



16 Digital Pins

PIN NO	Symbol	Fuction
1	VSS	GND
2	VDD	+5V
3	V0	Contrast adjustment
4	RS	H/L Register select signal
5	R/W	H/L Read/Write signal
6	E	H/L Enable signal
7	DB0	H/L Data bus line
8	DB1	H/L Data bus line
9	DB2	H/L Data bus line
10	DB3	H/L Data bus line
11	DB4	H/L Data bus line
12	DB5	H/L Data bus line
13	DB6	H/L Data bus line
14	DB7	H/L Data bus line
15	A	+4.2V for LED
16	K	Power supply for BKL(0V)

Pin Function

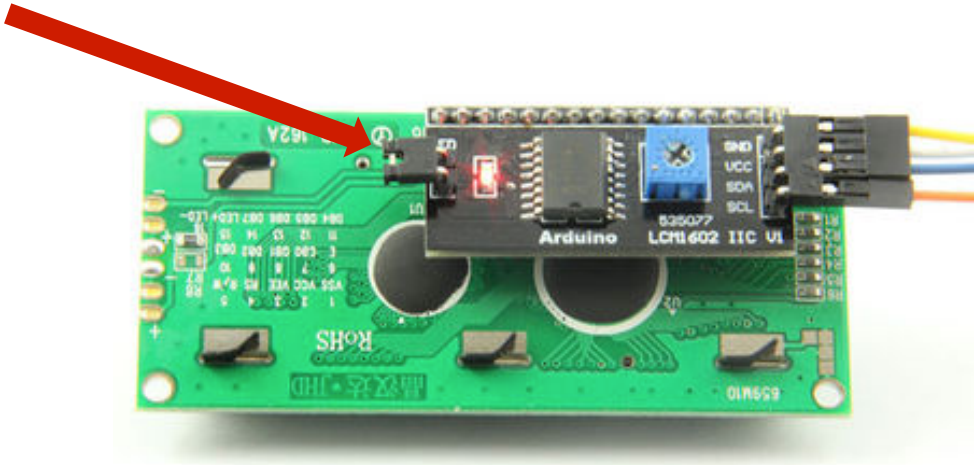
- Pin 1: Vss – connected to the GND
- Pin 2: VDD – connected to +5V
- Pin 3: V0 – control of the letters contrast. It is general connected to a potentiometer (or trimmer). (in this way it is possible apply a changeable voltage between 0V and 5V . Changing the voltage the contrast changes as well. Pin 4: RS signal– to select the register where register what appear on the LCD
- Pin 5: Read/Write signal – to select functional mode: RW – connected to GND
- Pin 6: Enable (E) signal – to enable writing to the registers
- From Pin 7 to Pin 14: lines used to communicate with the registers:
  - HIGH (H) value indicates a written value (WRITE) of the bit on the display register
  - LOW (L) indicates a read value (READ) from the register
- Pin 15: A (Anode) – Pin to which a positive voltage is applied (+5V) to have the backlight of the display.
- Pin 16: K (Cathode) – Pin to which a GND is applied to have the backlight of the display.

I2C Communication

The I2C protocol involves using two lines to send and receive data: a serial clock pin (SCL) that the Arduino or Genuino Master board pulses at a regular interval, and a serial data pin (SDA) over which data is sent between the two devices. As the clock line changes from low to high (known as the rising edge of the clock pulse), a single bit of information - that will form in sequence the address of a specific device and a a command or data - is transferred from the board to the I2C device over the SDA line. When this information is sent - bit after bit -, the called upon device executes the request and transmits it's data back - if required - to the board over the same line using the clock signal still generated by the Master on SCL as timing. The initial eight bits (i.e. eight clock pulses) from the Master to Slaves contain the address of the device the Master wants data from. The bits after contain the memory address on the Slave that the Master wants to read data from or write data to, and the data to be written, if any.

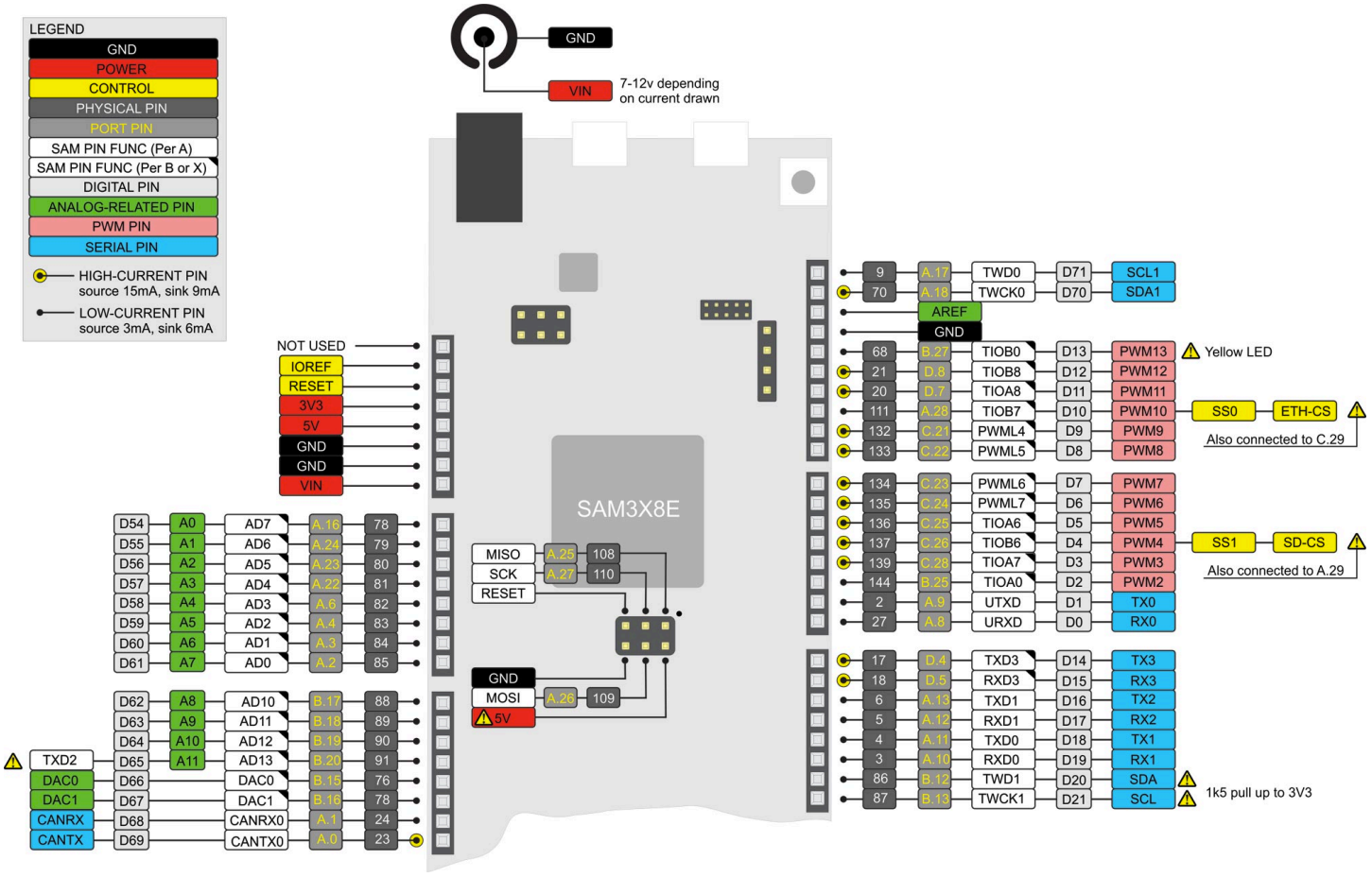
LCD monitor with I2C Driver

Jumper Blackligth

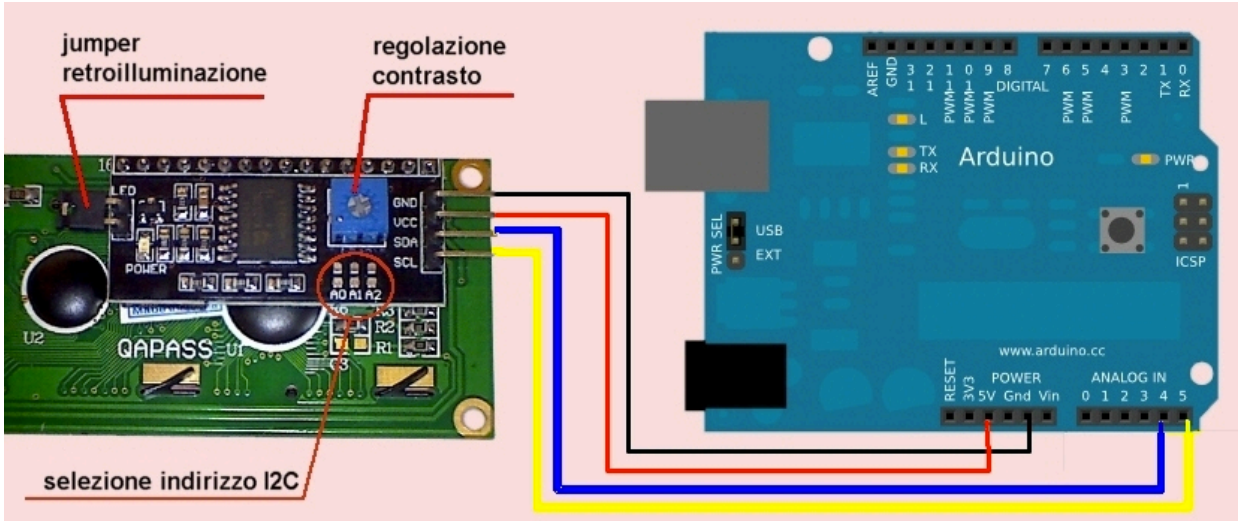


Board	I2C/TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

Arduino DUE Pin Map



The Circuit



PCF8574T	Arduino
SDA	SDA
SCL	SCL
Vcc	+5V
GND	GND

The Code

https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library

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

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  // put your setup code here, to run once:
lcd.begin();
lcd.backlight();
lcd.print("ciao");
lcd.setCursor(0,1);
lcd.print("rucola");
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Sixth Program

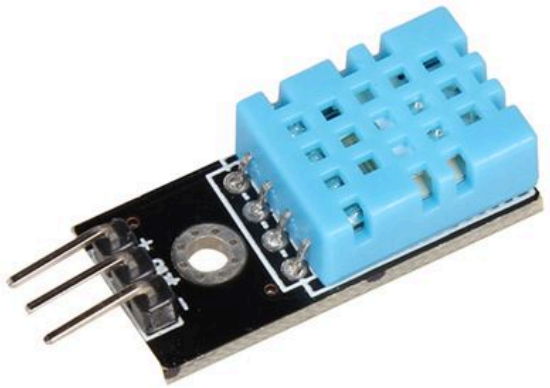
Temperature and humidity sensor

DHT11 Sensor

The DHT11 is a relatively cheap sensor for measuring temperature and humidity.

The DHT11 has three lines:

- GND,
- +5V
- and a single data line.

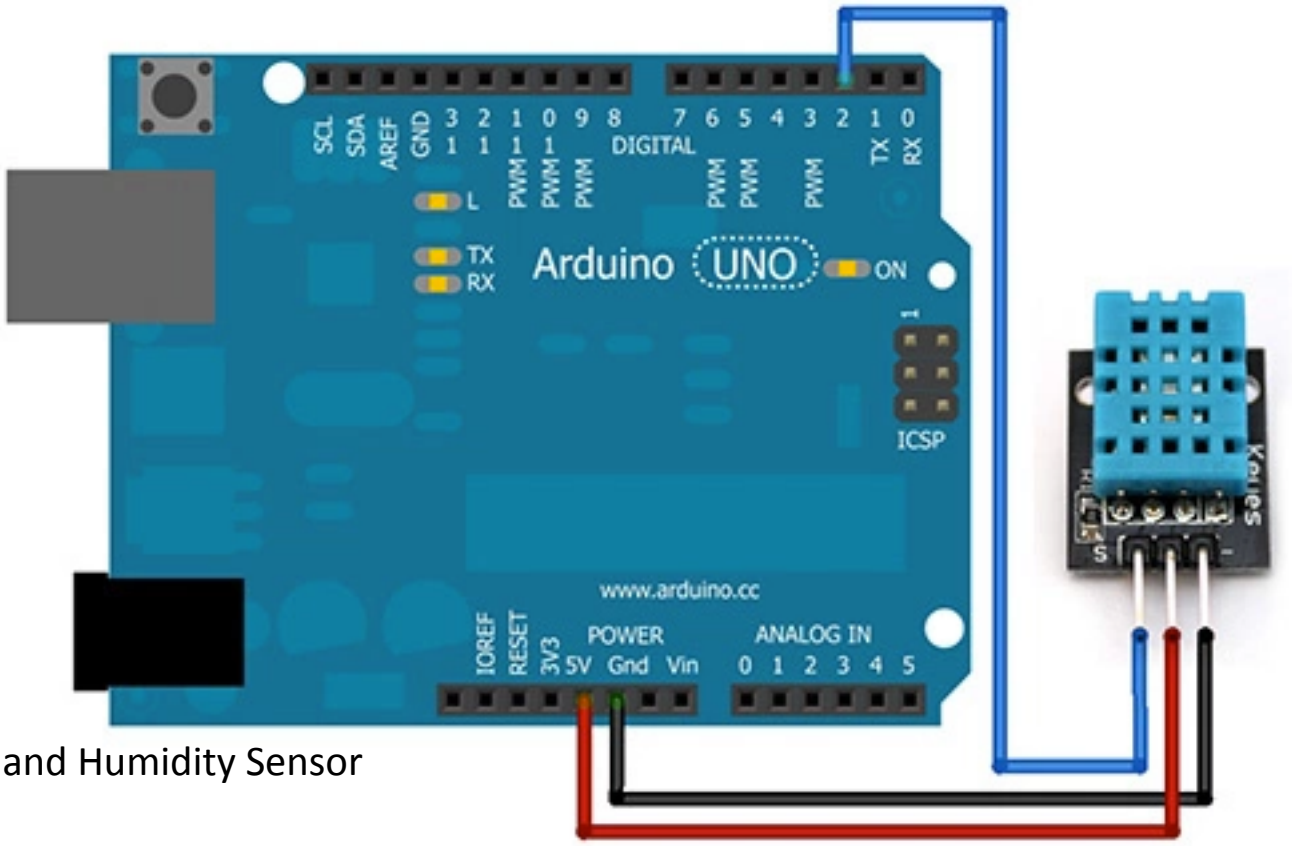


Signal transmission range: 20m
Temperature range: 0-50°C
Humidity range: 20-90%RH

By means of a handshake, the values are clocked out over the single digital line.

Datasheet:
http://www.micro4you.com/files/sensor/DHT11.pdf

The Circuit



Parts List

- Arduino
- I2C LCD display 16×2
- DHT11 Temperature and Humidity Sensor
- 4.7k resistor

Libraries

- DHT11 Library (Sensor)
- Wire Library (I2C communication)
- LiquidCrystal_I2C Library (LCD Monitor)

The Code

```
#include <DHT11.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

Int pin=4;
DHT11 dht11(pin);
LiquidCrystal_I2C lcd(0x27, 16, 2);

double Kelvin(double celsius){
  return celsius + 273.15;
 }

void setup() {
  // put your setup code here, to run once:
lcd.begin();
lcd.backlight();
lcd.clear();

lcd.print("HUM & TEMP");
delay(2000);
lcd.clear();

lcd.print("Starting .....");
delay(2000);
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  int err;
  float temp, humi;
  if((err=dht11.read(humi, temp))==0)
  {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("temp: ");
    lcd.print(Kelvin(temp));
    lcd.print(" K");
    lcd.setCursor(0,1);
    lcd.print(" hum: ");
    lcd.print(humi);
    lcd.print(" %");
  }
  else
  {
    lcd.println();
    lcd.print("Error No :");
    lcd.print(err);
  }
  delay(DHT11_RETRY_DELAY);

}
```

Contenuto kit

- 1 scheda ProtoShield con mini breadboard;
  1 Breadboard MB102 830 punti;
  15 LED (5 rossi, 5 verdi, 5 gialli);
  10 Resistori metal film da 10KOhm;
  10 Resistori metal film da 1KOhm;
  10 Resistori metal film da 220 Ohm;
  1 Circuito integrato SN74HC595 8-bit Shift Register;
  1 Buzzer attivo;
  1 Buzzer passivo;
  10 pulsanti 12*12*7.3;
  3 Fotoresistori;
  1 Potenziometro da 10kOhm;
  1 Sensore di temperatura LM35DZ LM35 TO-92;
  1 Sensore di fiamma ad infrarossi;
  1 Ricevitore ad infrarossi HS0038B HS0038 SIP3 TO-92;
  2 Sensori di inclinazione(Tilt Sensor);
  1 Tilt Sensor Switch;
  1 modulo LCD Blu 1602 con interfaccia seriale IIC/I2C/TWI;
  1 modulo joyStick PS2;
  1 Motore Stepper 5V con scheda drive ULN2003;
  1 Servo motore compatibile SG90;
  1 Modulo RGB 3 Colori;
  1 Modulo Relè ad 1 canale 5V;
  1 kit jumper wire (65 pezzi);
  10 pezzi Dupont Line (F a M);
  1 Header femmina 2.54mm 1x40 Pin Single Row;
  1 case per batterie 6-AA-1.5V;
  1 Modulo Real Time Clock RTC DS1302;
  1 kit RFID con Card read/write;
  1 Modulo Sensore di suono;
  1 Modulo Sensore di Temperatura e umidità DHT11;
  1 Tastiera a matrice 4x4;
  1 Modulo Display 8x8 Dot-Matrix 32x32 mm;
  1 Modulo Sensore del livello dell'acqua;

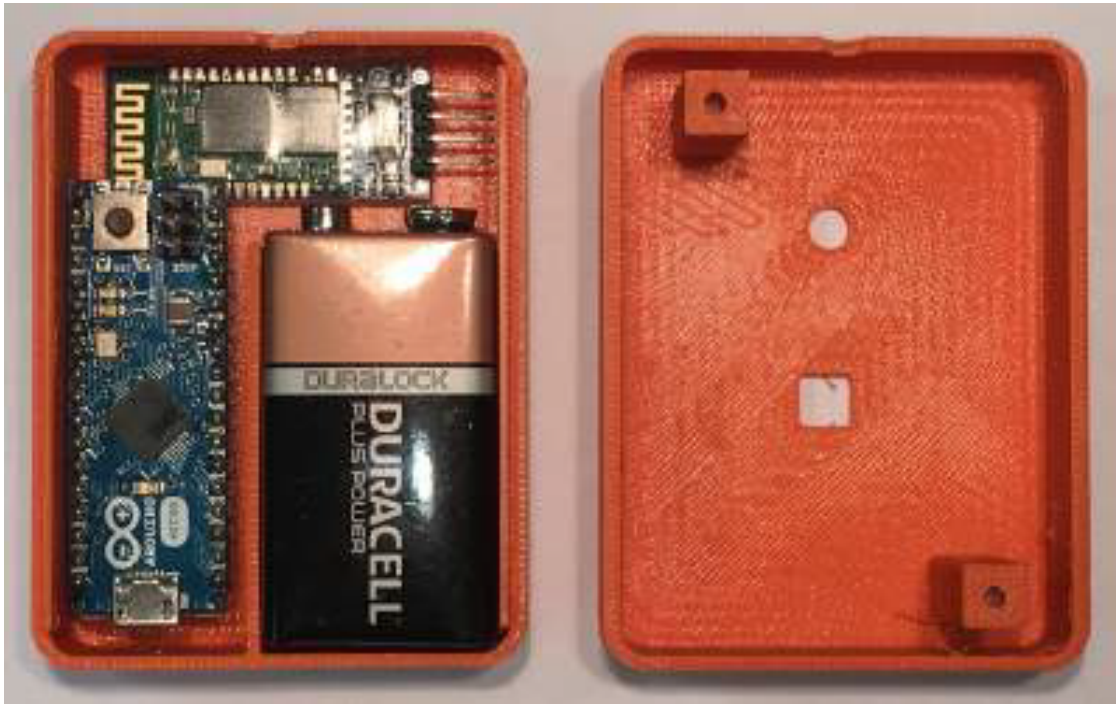
Fritzing

Dimensionamento dei componenti

- Dimensionamento batteria

- Dimensionamento attuatore

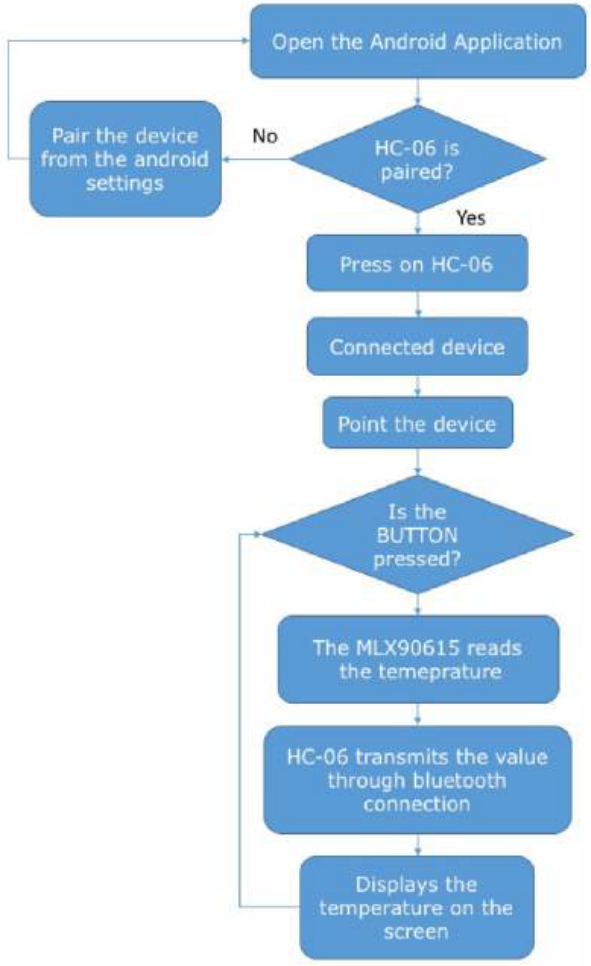
Contactless thermometer

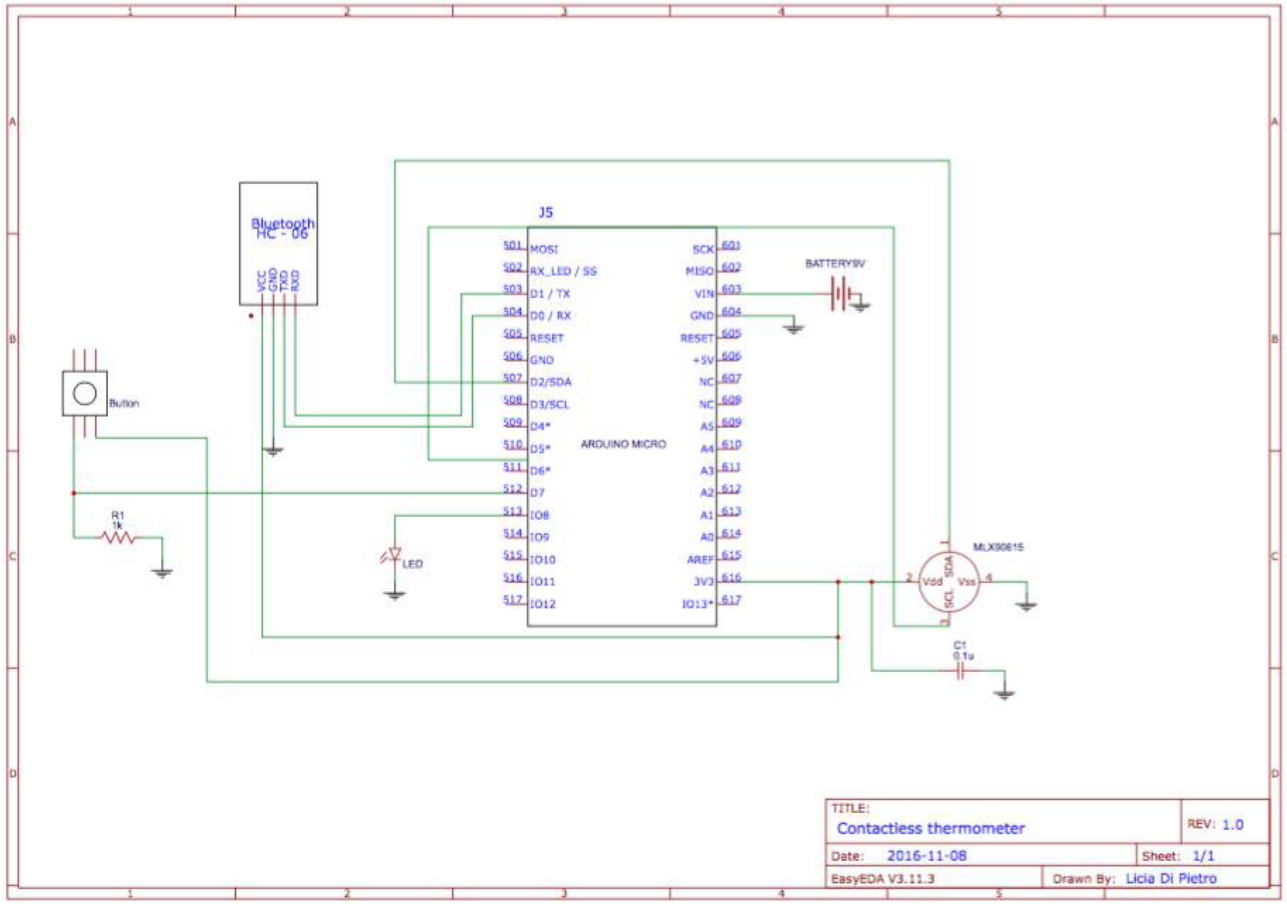


MLX90615 by Melexis

Battery capacity 550 mAh @9 V

Contactless thermometer

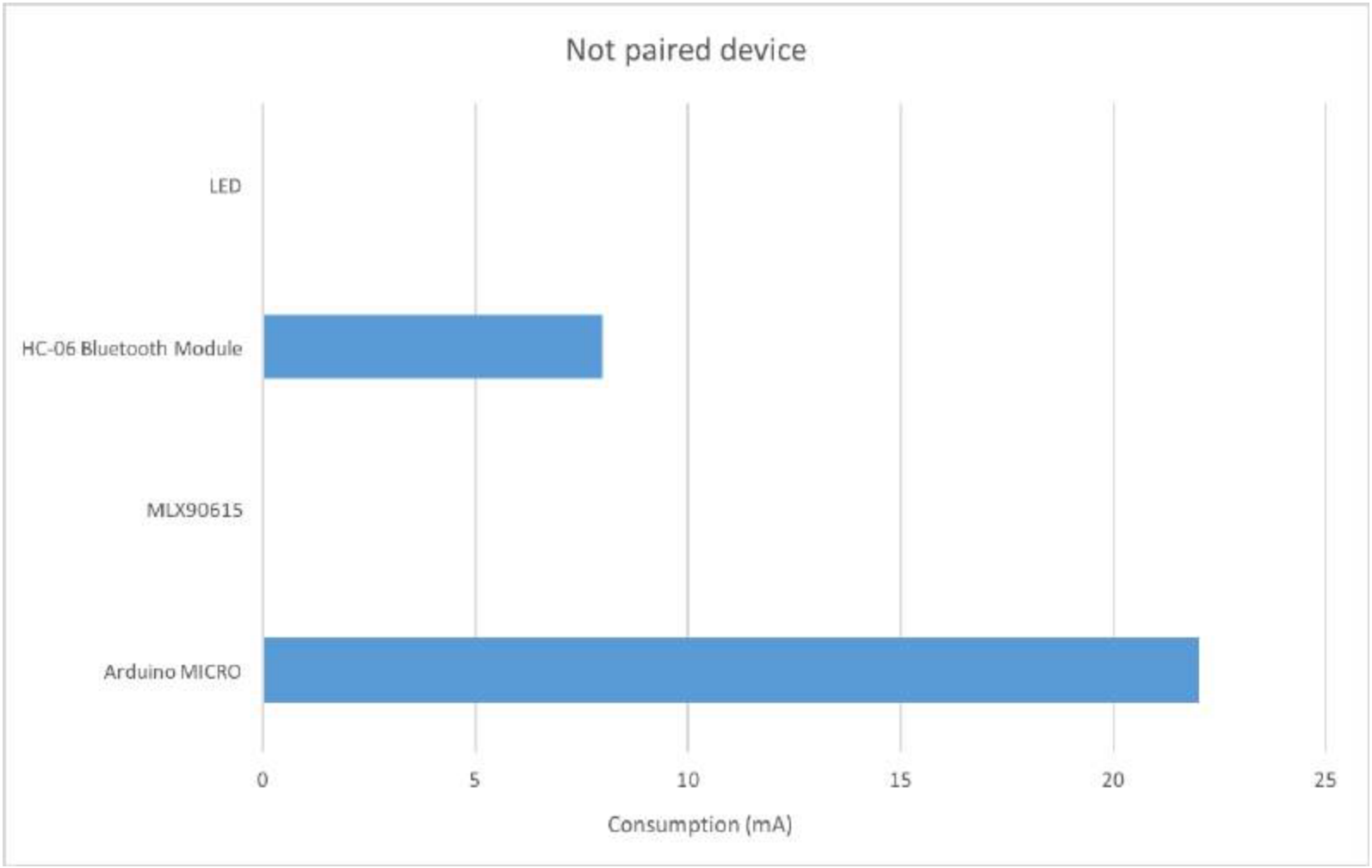




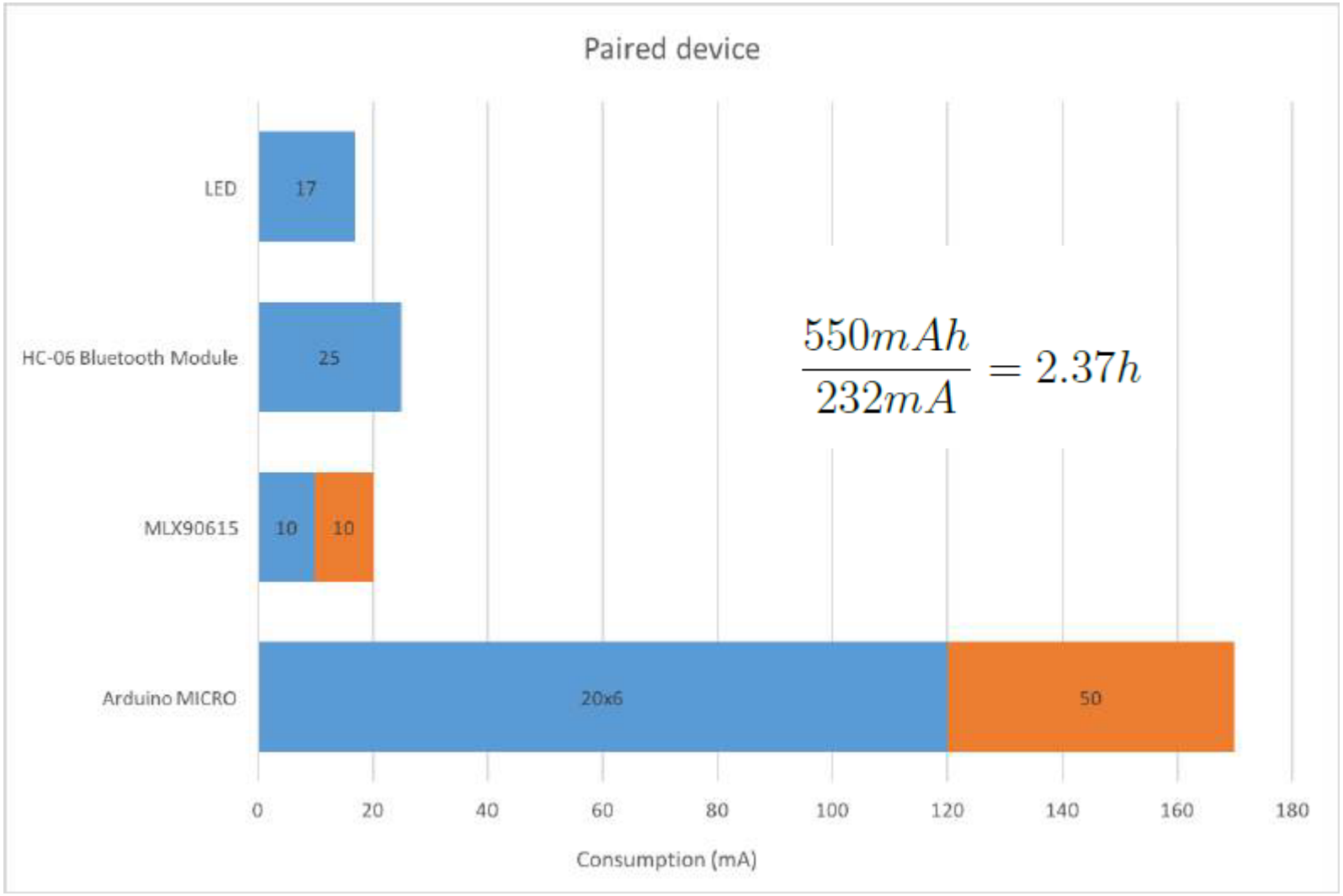
Current consuption

Arduino Micro Board	
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Sleep Mode	20 mA
MLX90615 InfraRed Sensor	
DC clamp current, SDA pin	10 mA
DC clamp current, SCL pin	10 mA
HC-06 Bluetooth Module	
During the pairing	25 mA
After pairing	8 mA
LED-Basic Led 5mm	
Using current	16-18 mA

Current consuption

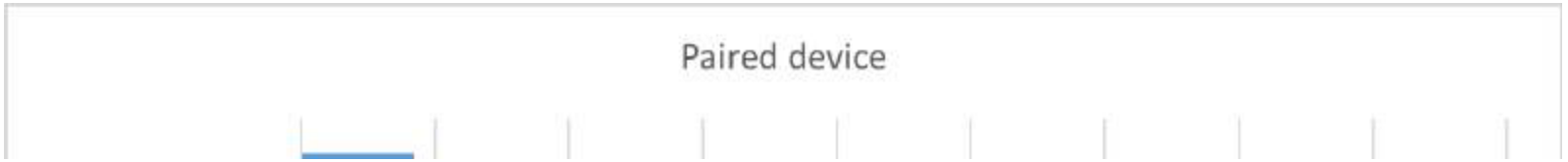


Current consuption



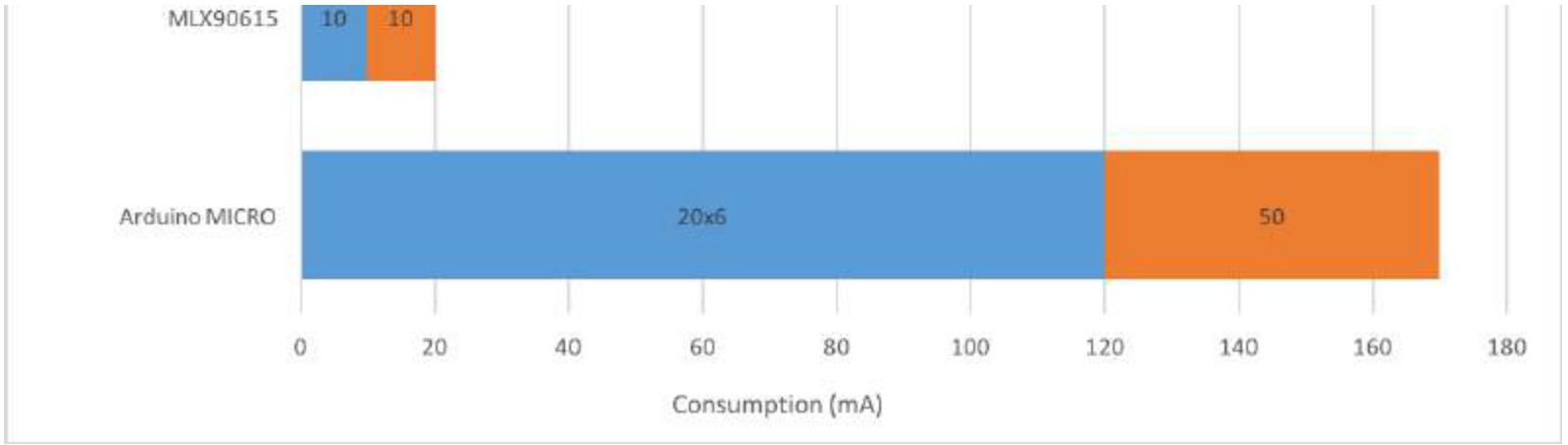
$$\frac{550mAh}{232mA} = 2.37h$$

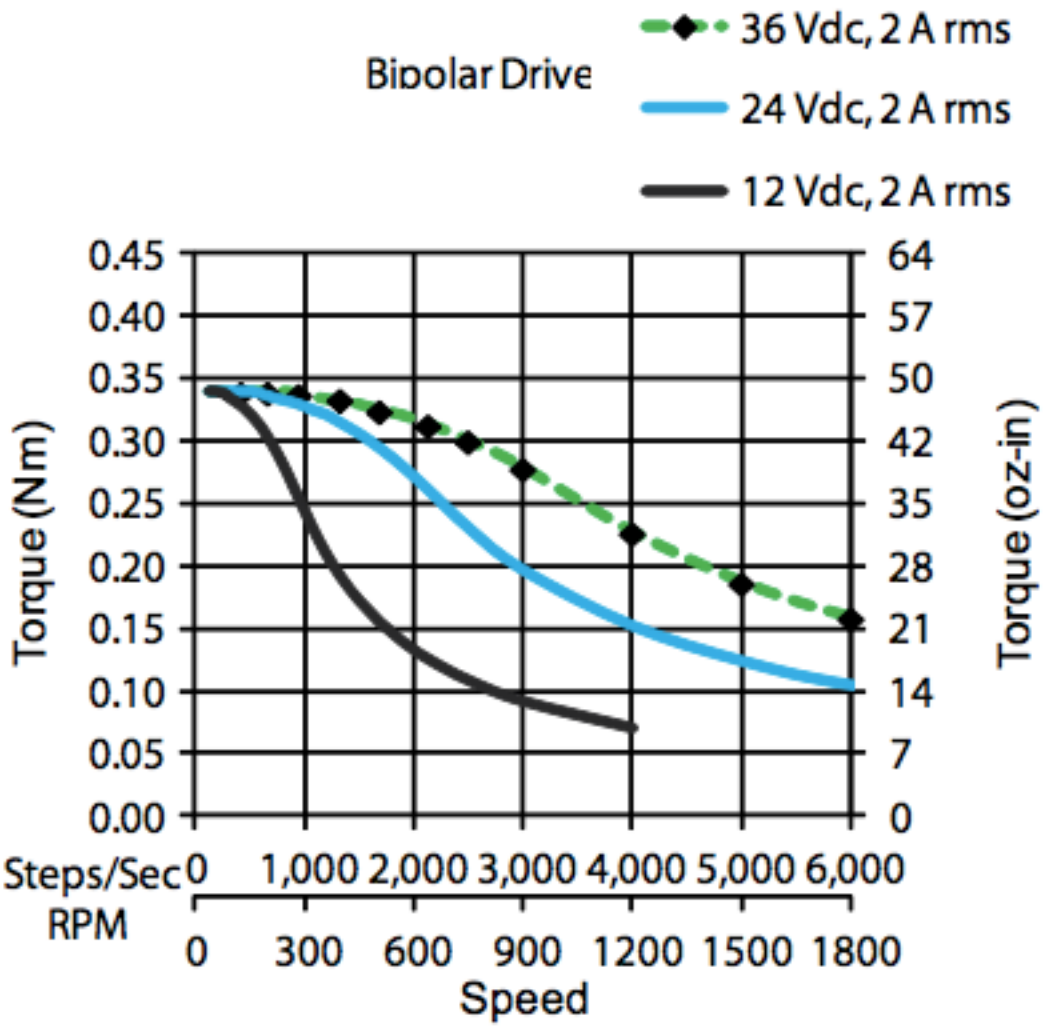
Current consuption



Using the device in continuous way the battery life is 2h 22m 12s but considering that the duration of only one read is about 6s, is possible using the thermometer for 1422 times, as calculated in Equation 3.4.

$$\frac{12s}{6s} + \frac{22 \times 60s}{6s} + \frac{2 \times 60 \times 60s}{6s} = 1422 \tag{3.4}$$




Bipolar Drive
36 Vdc, 2 A rms
24 Vdc, 2 A rms
12 Vdc, 2 A rms
Torque (Nm)
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Torque (oz-in)
64
57
50
42
35
28
21
14
7
0
Steps/Sec 0 1,000 2,000 3,000 4,000 5,000 6,000
RPM 0 300 600 900 1200 1500 1800
Speed

References

- http://predictabledesigns.com/how-to-develop-and-prototype-a-new-product/
- https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard
- http://www.electronics-tutorials.ws/resistor/res_1.html
- http://www.studyelectrical.com/2016/12/different-types-classification-of-capacitors.html
- https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds
- http://www.madehow.com/Volume-2/Printed-Circuit-Board.html
- https://www.arduino.cc/en/Guide/Introduction
- https://www.engineersgarage.com/tutorials/twi-i2c-interface