

---

---

# Embedded Workshop

— Jan 2016 —

---

---



# Agenda for Workshop

## Programs on USB Key

### Project 1 I2C Scanner

Scans I2C cbus for addresses

### Project\_2\_DHT.ino

DHT11 Read Sensor and Display  
Temperature & Humidity to Serial  
Console

### Project\_3>HelloWorld\_U8glib.ino

Display "Hello World" using an OLED  
128x64 0.96, Organic Light Emitting  
Diodes and the U8glib library

### Project\_4\_bitmap\_pic\_DMS

How to Draw a bitmap image

### Project\_4b\_bitmap\_pic\_blank

Use your own bitmap image

### Project\_5\_Weather\_Station\_DHT11\_OLED

Read Temp & Humidity sensor and  
pass to the OLED

### Project\_5b\_Weather\_Station\_Controller

## Present DHTxx Sensor

Bread board Wiring & DHT pinout.

DHT Description & Specifications

Interfacing with Arduino

Review for using libraries

Review sketch [Program\\_2\\_DHT.ino](#)

## Present OLED 128x64 Display

Bread board Wiring & OLED I2C pinout.

OLED Description & Specifications

Interfacing with Arduino

Review for using libraries

Review sketch [Project\\_3>HelloWorld\\_U8glib.ino](#)

Review sketch [Project\\_4\\_bitmap\\_pic\\_DMS](#)

Review sketch [Project\\_4b\\_bitmap\\_pic\\_blank](#)

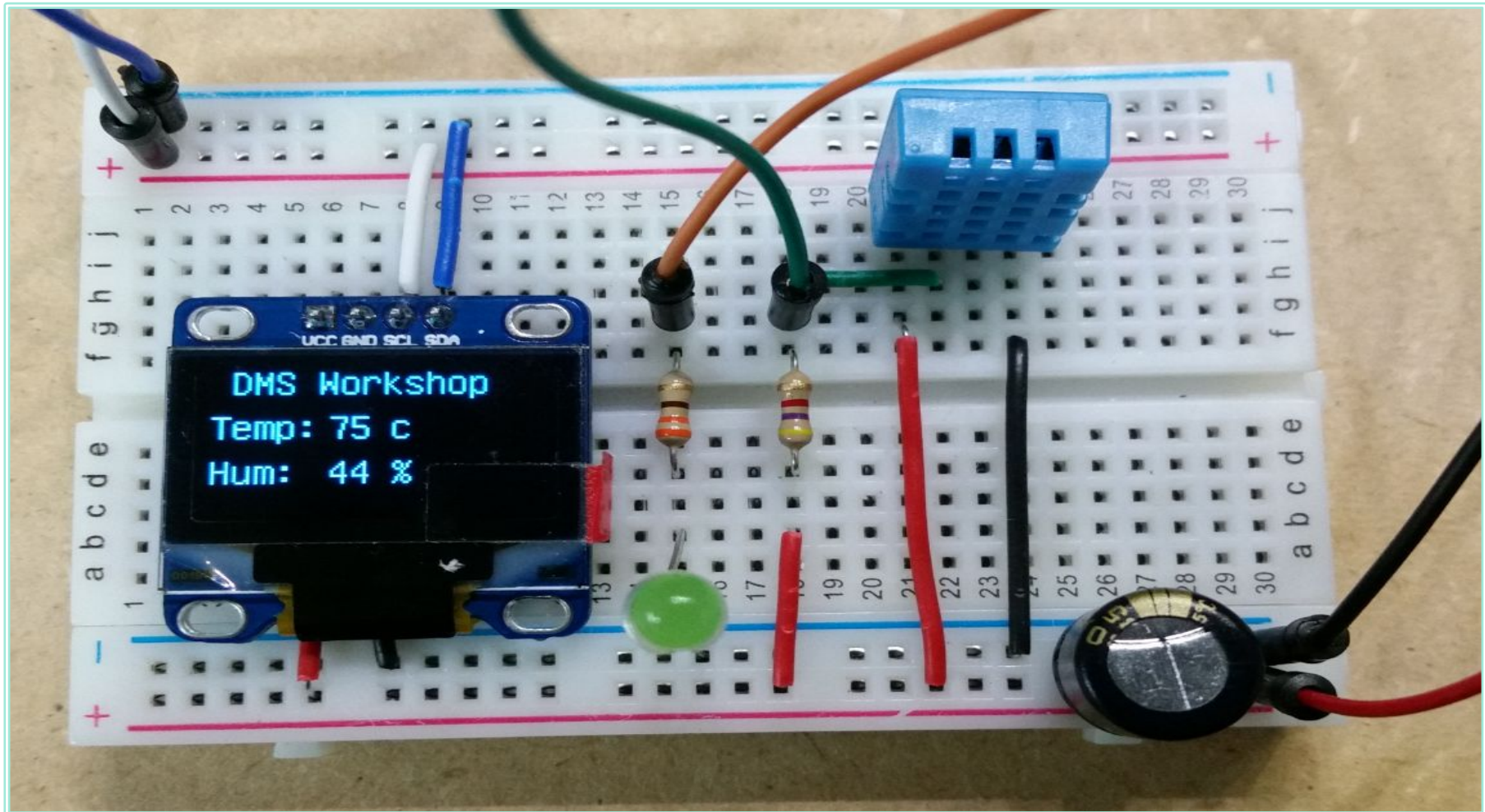
## Present DHT & OLED Display working together

Review sketch [Project\\_5\\_Weather\\_Station\\_DHT\\_OLED](#)

Review sketch [Project\\_5b\\_Weather\\_Station\\_Controller](#)

## Going Beyond

Creating you own Weather Station.



# DHT11-Humidity-Temperature Sensor

The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and sends out a digital signal on the data pin (no analog input pins needed). Fairly simple to use, but requires careful timing to grab data. The sensor can only get new data from it once every 2 seconds. When using the Arduino library, sensor readings can be up to 2 seconds old. 3 to 5V power and I/O (V)oltage, (S)ignal, (G)round

When the connecting cable is shorter than 20 metres use a 5K-10K pull-up resistor.

## Specifications:

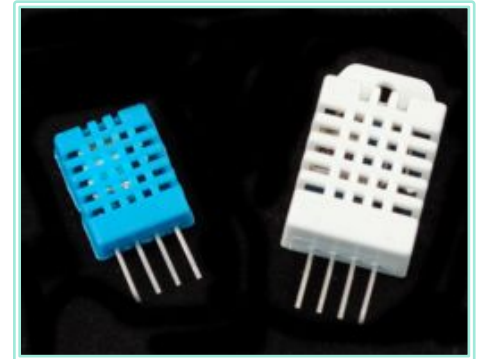
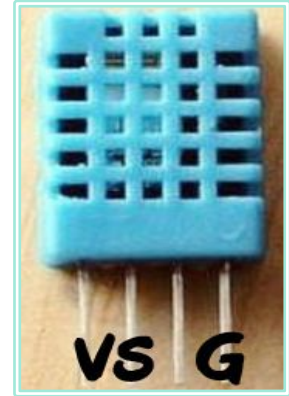
- Humidity measuring range: 20% ~ 90% RH (0-50 ? temperature compensation)
- Temperature measuring range: 0 ~ +50 ?C

## Features

- .. Single wire digital interface
- .. Ultra-small size ( 12X15.5X5.5 mm )
- .. High reliability
- .. Optimized long-term stability

Datasheet Adafruit.com: <https://www.adafruit.com/datasheets/DHT22.pdf>

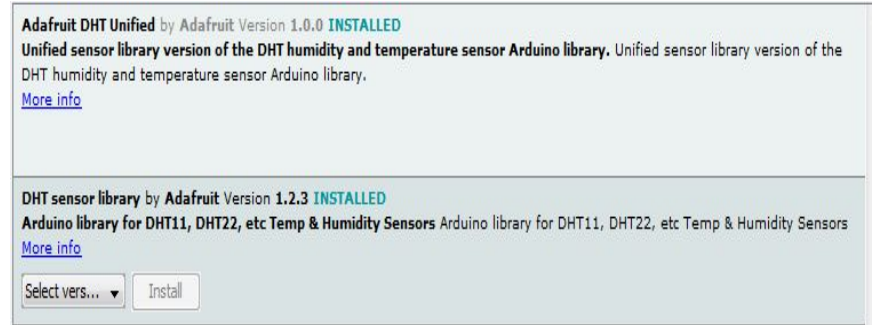
Datasheet Robocraft.ru <http://robocraft.ru/files/datasheet/DHT11.pdf>



# Installing Drivers DHT.h

## DHT Instructions

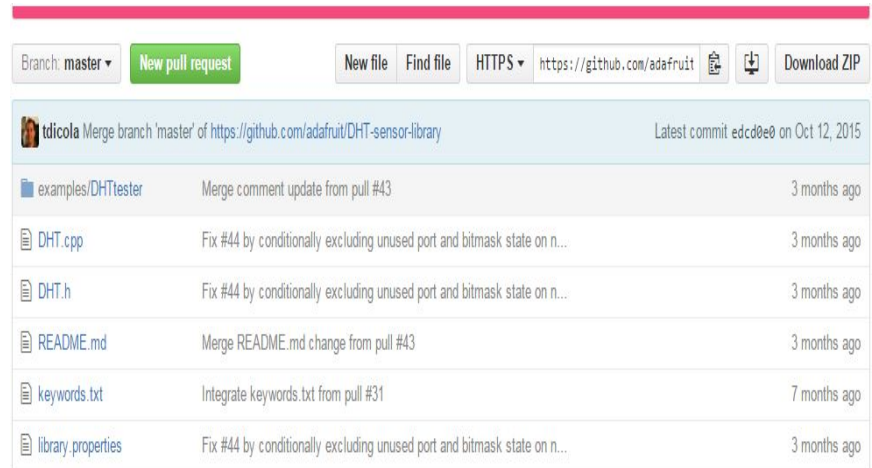
- Arduino (Installation with IDE)
    1. Start Arduino IDE
    2. In the Arduino IDE, import the library from the "Add Library" Menu.
- 
- **Older versions or LINUX**
  - <https://github.com/adafruit/DHT-sensor-library>
  - **Select** Download Zip button in the top right corner from above URL.
  - Rename the uncompressed folder DHT.
  - Check that the DHT folder contains **DHT.cpp** and **DHT.h**.
  - Place the DHT library folder your `/libraries/` folder.



The screenshot shows the Arduino IDE library manager interface. It displays two installed libraries:

- Adafruit DHT Unified** by Adafruit Version 1.0.0 **INSTALLED**. Description: Unified sensor library version of the DHT humidity and temperature sensor Arduino library. [More info](#)
- DHT sensor library** by Adafruit Version 1.2.3 **INSTALLED**. Description: Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors. [More info](#)

Below the second library, there is a "Select vers..." dropdown menu and an "Install" button.



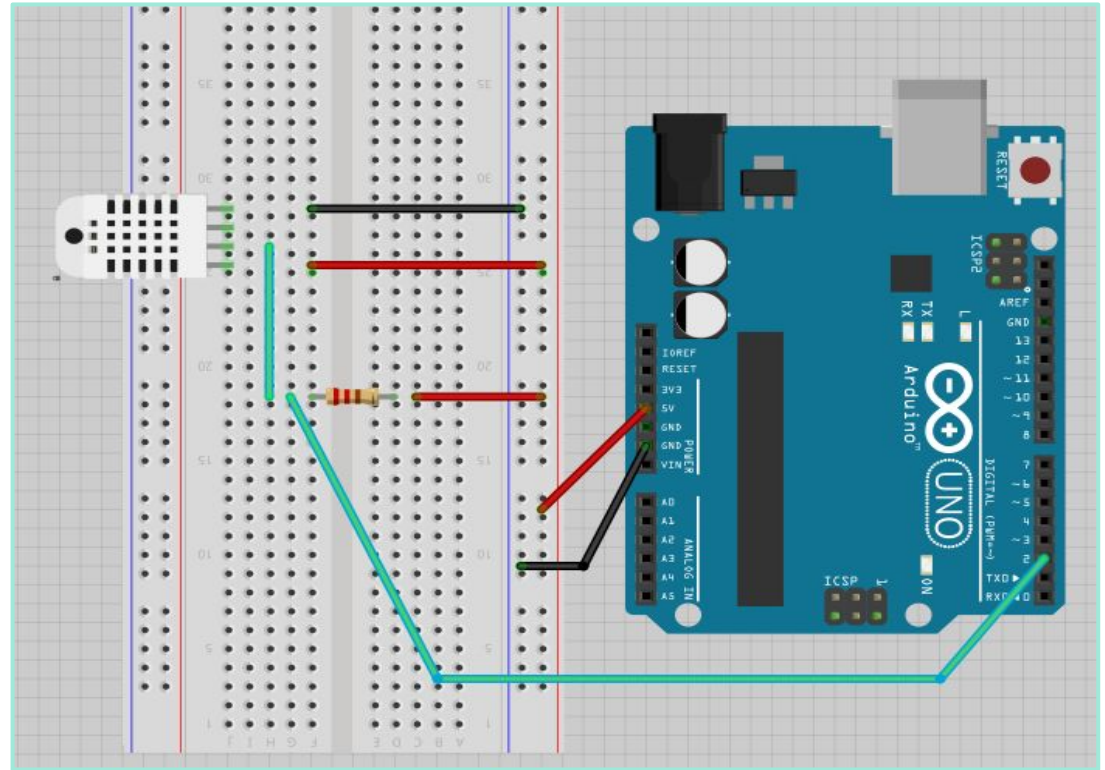
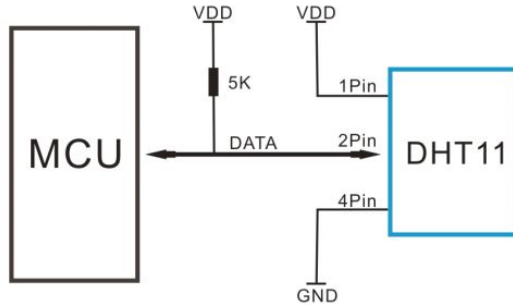
The screenshot shows the GitHub repository page for `adafruit/DHT-sensor-library`. The repository is on the `master` branch. The latest commit is `edcd0e0` on Oct 12, 2015. The repository contains several files and folders:

File/Folder	Description	Time
<code>examples/DHTtester</code>	Merge comment update from pull #43	3 months ago
<code>DHT.cpp</code>	Fix #44 by conditionally excluding unused port and bitmask state on n...	3 months ago
<code>DHT.h</code>	Fix #44 by conditionally excluding unused port and bitmask state on n...	3 months ago
<code>README.md</code>	Merge README.md change from pull #43	3 months ago
<code>keywords.txt</code>	Integrate keywords.txt from pull #31	7 months ago
<code>library.properties</code>	Fix #44 by conditionally excluding unused port and bitmask state on n...	3 months ago

# Wiring up DHT11

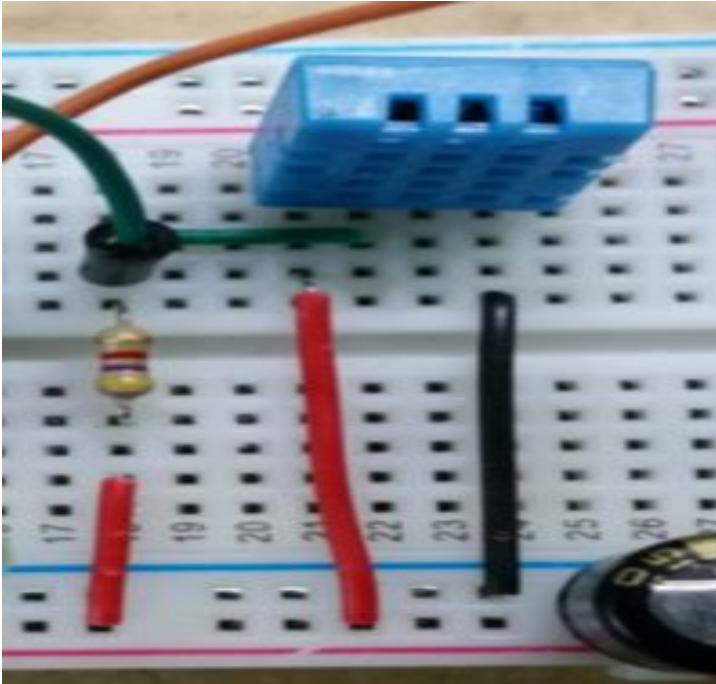
Pin	Name	Description
1	VDD	Power supply 3 - 5.5 V DC
2	DATA	Serial data output
3	NC	Not connected
4	GND	Ground

3. Typical Application (Figure 1)



# Using DHTxx with the Arduino

Connect pin 1 (on the left) of the sensor to +5V  
Connect pin 2 of the sensor to Pin2 Arduino DHTPIN  
Connect pin 4 (on the right) of the sensor to GROUND  
Connect a 5K- 10K resistor from pin 2 (data) to pin 1 (power)  
of the sensor



## Connect

DHTxx Sensor from diagram on left side.

## Arduino Setup: Select:

Tools - Board - Arduino

Tools - Port - (Select the connected USB Port)

**Load** Project\_2\_DHT.ino sketch to your Arduino.

## Open Serial Monitor:

Tools -Serial Monitor

## Output:

**DMS Jan 2016 Workshop DHTxx  
Program Project\_2\_DHT.ino  
Initializing Sensor Please Wait**

**Temperature 77.00F Humidity 36.00 %**



# Organic Light Emitting Diodes (OLED) Display

An organic light-emitting diode (OLED) is a light-emitting diode (LED) in which the emissive electroluminescent layer is a film of organic compound that emits light in response to an electric current.

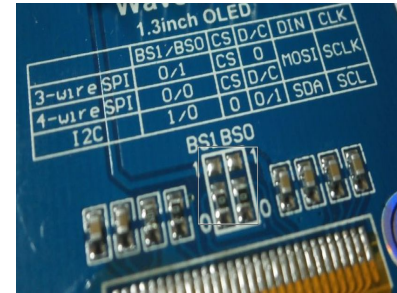
This layer of organic semiconductor is situated between two electrodes; typically, at least one of these electrodes is transparent. - [Wikipedia](#)

## This Workshop is using Model SSD1306

.96" inch in size, features 128×64 pixels and uses the 4-wire SPI and I2C Bus. It is monochrome blue in color. Also it consumes 0.04 W of energy which is one tenth that is required to run traditional 16x2 LCD display.

Note:

Some versions of the OLED can be modified to work with different either I2C or SPI by moving one of the two resistors on the back of the board. I2C remove BS1 resistor on 1 but keep BS0 on 0 as it is. SPI BS1 must be 0 and BS0 on 1



## u8glib Library Instruction Set

Examples:

- [begin](#)
- [drawBox](#)
- [drawStr](#)
- [drawFrame](#)
- [drawBitmapP](#)

<https://code.google.com/p/u8glib/wiki/userreference>  
<https://github.com/olikraus/u8glib>

# Universal Graphics Library for 8 Bit Embedded Systems **u8glib**

## Picture Loop

U8glib needs a special programming construct, called "picture loop". It can be usually placed in the loop() procedure of the Arduino Sketch:

```
void loop(void) {  
  // picture loop  
  u8g.firstPage();  
  do {  
    draw();  
  } while( u8g.nextPage() );  
  
  // rebuild the picture after some delay  
  delay(1000);  
}
```

The full picture of the graphics device is drawn by looping through the picture loop.

Best practice is to place all graphics commands into one procedure (draw()):

All graphics commands are listed in the [User Reference Page](#) of U8glib.

```
void draw(void) {  
  // graphic commands to redraw the complete screen should be placed here  
  u8g.setFont(u8g_font_unifont);  
  u8g.drawStr( 0, 20, "Hello World!");  
}
```

The first command in this little example selects a font. All fonts start with u8g\_font\_. Fonts can be selected by their [size](#) or by their [origin](#).

```
u8g.setFont(u8g_font_unifont); Fonts listed by size
```

The second command writes a string to the display. The first two arguments define the position on the screen. The drawStr procedure uses the previously selected font and will use the current color index (defaults to 1) to render the string on the display.

## U8glib device select:

```
U8GLIB_DOGM128(sck, mosi, cs, a0 [, reset]) // SPI
```

```
U8GLIB //I2C
```

- [List of devices supported by U8glib](#)



# Installing Drivers u8glib.h

## u8glib.h Instructions:

- Arduino (Installation with IDE)
  1. Start Arduino IDE
  2. In the Arduino IDE, import the library from the "Add Library" Menu.

U8glib by oliver Version 1.18.0 **INSTALLED**

A library for monochrome TFTs and OLEDs Supported display controller: SSD1306, SSD1309, SSD1322, SSD1325, SSD1327, SH1106, UC1601, UC1610, UC1611, UC1701, ST7565, ST7920, KSD108, LC7981, PCD8544, PCF8812, SBN1661, TLS8204, T6963.  
[More info](#)



## u8glib.h for LINUX and older version of Arduino

- Open <https://code.google.com/p/u8glib/>
- Select U8glib for Arduino: Select **Files** tab
- Download (ver:1.18.1) [u8glib\\_arduino\\_v1.18.1.zip](#)  
Select - **Sketch** - **Import Library** - **Add Library**
  - i. Select file in Download Folder:
- Verify Library - Select - **Sketch** - **Import Library** - **U8glib**
- This should add `#include "U8glib.h"` to your program

General	Readme	Release Notes	Reviews (1)	Statistics	Files
/					
Name		Updated		Size	
•		-		-	
	<a href="#">u8glib_arduino_v1.14.zip</a>	2 years ago		1 MB	
	<a href="#">u8glib_arduino_v1.15.zip</a>	2 years ago		1.1 MB	
	<a href="#">u8glib_arduino_v1.16.zip</a>	2 years ago		1.1 MB	
	<a href="#">u8glib_arduino_v1.17.zip</a>	1 year ago		1.1 MB	
	<a href="#">u8glib_arduino_v1.18.1.zip</a>	6 months ago		1.1 MB	

# Using OLED with the Arduino

## Connection for OLED 128x64 I2C

Connect pin 1 (on the left) of the OLED to +5V

Connect pin 2 of the sensor to GND

Connect pin 3 of the OLED to SCL

Connect pin 4 of the OLED to SDA

## SPI connect the OLED pins

### Adafruit Version

GND goes to ground

Vin goes to 5V

DATA to digital 9

CLK to digital 10

D/C to digital 11

RST to digital 13

CS to digital 12

DIN to pin 13

CLK to pin 11

CS to pin 10

D/C to pin 9

RES to pin 8



## Connect

OLED Sensor from diagram on left.

## Arduino Setup: Select:

Tools - Board - Arduino

Tools - Port - (Select the connected USB Port)

Load [Project\\_3>HelloWorld\\_U8glib.ino](#) sketch to your Arduino.

## OLED Display:

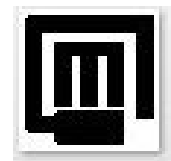
DMS Embedded

Workshop Jan2016

Hello World!

Load [Project\\_4\\_bitmap\\_pic\\_DMS.ino](#)

## OLED Display:



# Create Bitmap image to display

Download LCD Assistant

[http://en.radzio.dxp.pl/bitmap\\_converter](http://en.radzio.dxp.pl/bitmap_converter)

File - Load image: **DMS\_32x32.bmp**

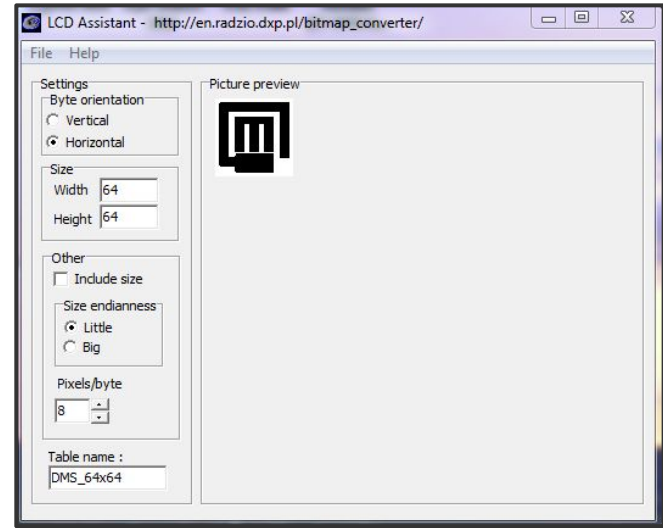
File - Save output: **DMS\_hex**

Double click on DMS\_hex and use **Notepad** to open file.

Cut and paste hex data into sketch.

[Project\\_4\\_bitmap\\_pic\\_blank.ino](#)

Flash Arduino with new code.



```
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0x00, 0x00, 0x00, 0x7F, 0x1F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x7F, 0x1F, 0x00, 0x1F, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x7F, 0x1F, 0xFF, 0xFF, 0xFF, 0x00,
0x1F, 0x00, 0xFF, 0xFF, 0xFF, 0x00, 0x7F, 0x1F, 0x00, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x7F,
0xFF, 0xFF, 0xFF, 0x00, 0x7F, 0x1F, 0x00, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x7F, 0x1F,
0xFF, 0x00, 0x7F, 0x1F, 0x00, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x7F, 0x1F, 0xFF, 0xFF,
0xFF, 0x00, 0x7F, 0x1F, 0x00, 0x1F, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x7F, 0x1F, 0xFF, 0xFF,
0x00, 0x1F, 0x00, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x18, 0x30, 0x30, 0x2C, 0x20, 0x78, 0x30, 0x30, 0x2C, 0x20, 0x30, 0x30, 0x2C, 0x20, 0x30,
0x30, 0x2C, 0x20, 0x30, 0x78, 0x30, 0x30, 0x2C, 0x20,
};
```

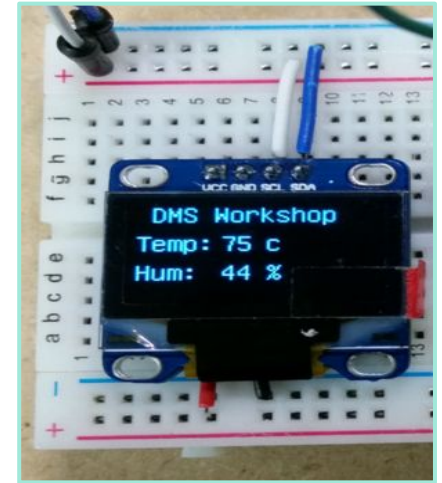
# Weather Station

## Load: [Project\\_5\\_Weather\\_Station\\_DHT11\\_OLED](#)

Read Temp & Humidity sensor and pass the values to the OLED Display

## Load: [Project\\_5b\\_Weather\\_Station\\_Controller](#)

Read Temp & Humidity sensor and pass the values to the OLED Display.  
If Temp > 82 turn on Fan Controller Pin.



# Going Beyond

## Build your own Weather Station:

- Write your code.
- Add graphics to Display.
- Draw the schematic.
- Create the PCB.
- Populate your board with parts.
- Design housing using Sketchup.
- Use a 3D printer for the housing.

Thanks to **Jay Box**

The image below is from an ATtiny 85 schematic CAD made simple for the workshop.  
There is a downloadable .ZIP file below it containing BMP/GIF/PNG/JPG versions.

- Pull-up resistors used \*IF\* internal one is insufficient
- Pullups could use output pin instead of Vcc
- Power switch could be added
- For I2C bus expansion
  - Real-time clock
  - EEprom memory
  - or Sensors

