

# JG Electronics

## Arduino™ Projects

### Compendium

For a complete understanding of Art-Net™ and DMX read JGE Remote DMX manual

**Arduino™ and IoT.....3**

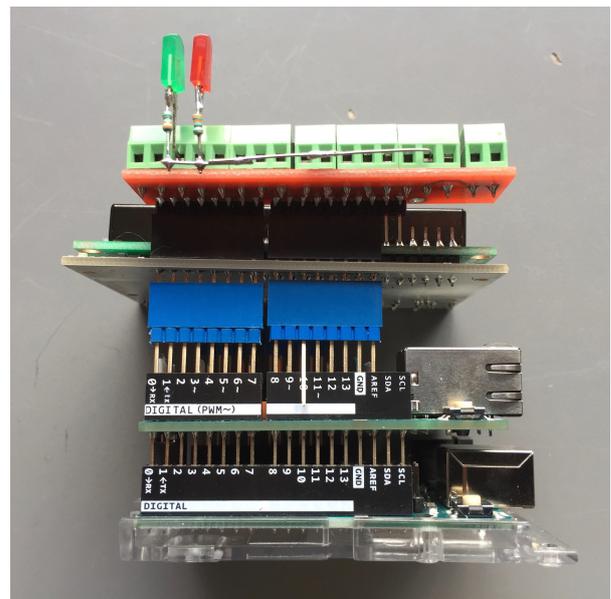
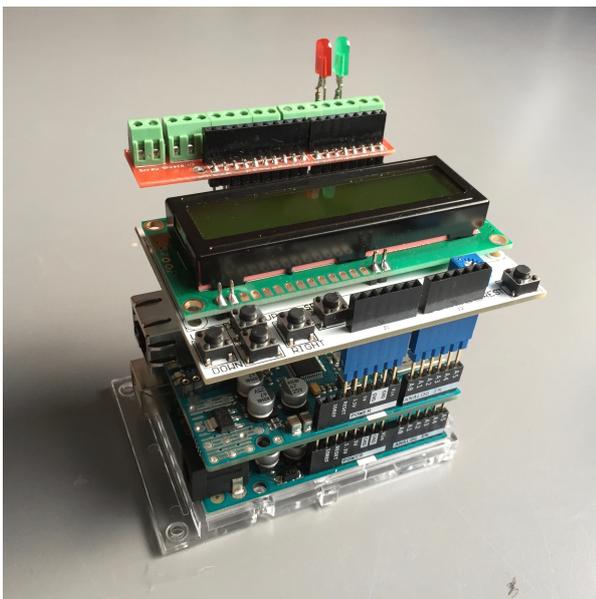
- Arduino™ Hardware Set Up #1 .....3
- Arduino Project #1 .....4
- Art-Net protocol.....4
- DMX Protocol .....4
- Arduino Project #2.....6
- Arduino Project #3.....9
- Arduino Project #4.....10

# Arduino™ and IoT

Remote DMX for iPhone™ or MacBook™ is an app specifically created for theatre technicians and lighting designers to control, adjust and focus spotlights on lighting catwalks (bridges), trusses or fly-bars with a iPhone™ or iPad™ without a light board operator but can also be used with an Arduino™ to control additional IoT (Internet of Things) devices or equipment that are a part for example of home automation. These devices can include lighting, heating and air conditioning, media and security systems. With Remote DMX you can control 1024 channels and every channel can have a value between 0 and 255.

## Arduino™ Hardware Set Up #1

Visit for information of the original Arduino™ processor boards and shields and to download the free IDE for MacBook™ to program the Arduino™ processor boards: <https://www.arduino.cc> and goto: <https://www.arduino.cc/en/Guide/HomePage> for information about software libraries a.s.o.



Arduino UNO Rev3

Arduino Ethernet Shield 2

LCD Shield: 16 x 2 LCD Keypad

LED's:

<https://store.arduino.cc/arduino-uno-smd-rev3>

<https://store.arduino.cc/arduino-ethernet-shield-2>

<https://www.sparkfun.com/products/13293>

connected to output 2 and output 3 of the Arduino UNO

Connect the Ethernet Shield to the Arduino UNO and connect the LCD Shield with extension connectors to the Ethernet Shield but don't connect output pin 10 to the LCD Shield because that causes flickering of the back light of the LCD because pin 10 is used by the Ethernet Shield.

A excellent book with sketches and videos for those who start exploring the Arduino™ from the beginning. <https://jeremyblum.com/portfolio/exploring-arduino>

# Arduino Project #1

<http://www.jgelectronics.nl/art-net-dmx/arduino-project-1.html>

Arduino Hardware Set Up #1

Arduino Software Sketch: 20171214\_DhcpAddressPrinter.ino

Connect the Arduino UNO board with an USB cable to your MacBook™ and the Ethernet Shield with a RJ45 Ethernet cable to your router. Start up the Arduino IDE, open the sketch file “20171214\_DhcpAddressPrinter.ino” and upload the sketch to the Arduino. On the LCD display and on the serial monitor screen you will see the IP Address that the Arduino received from your router. The address of your router you can find in: system preferences → network → advanced → TCP/IP and the Mac address of your MacBook™ in system preferences → network → advanced → Hardware.

To get a better understanding of how communication over Internet works here a list of common concepts:

MAC Address: [https://en.wikipedia.org/wiki/MAC\\_address](https://en.wikipedia.org/wiki/MAC_address)  
IP Address: [https://en.wikipedia.org/wiki/IP\\_address](https://en.wikipedia.org/wiki/IP_address)  
Subnet: <https://en.wikipedia.org/wiki/Subnetwork>  
Router: [https://en.wikipedia.org/wiki/Router\\_\(computing\)](https://en.wikipedia.org/wiki/Router_(computing))  
Gateway: [https://en.wikipedia.org/wiki/Gateway\\_\(computer\\_program\)](https://en.wikipedia.org/wiki/Gateway_(computer_program))  
DHCP: [https://en.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)  
HTTP: [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)  
DNS: [https://en.wikipedia.org/wiki/Domain\\_Name\\_System](https://en.wikipedia.org/wiki/Domain_Name_System)  
HTML: <https://en.wikipedia.org/wiki/HTML>  
Client and Server: [https://en.wikipedia.org/wiki/Client-server\\_model](https://en.wikipedia.org/wiki/Client-server_model)

## Art-Net protocol

<http://art-net.org.uk>

Het Art-Net™ protocol consist (in short) of a header “Art-Net” , one byte that indicates the SubNet number, one byte that indicates Universe-1, one byte that indicates Universe-2 and a packet of maximum 512 data bytes.

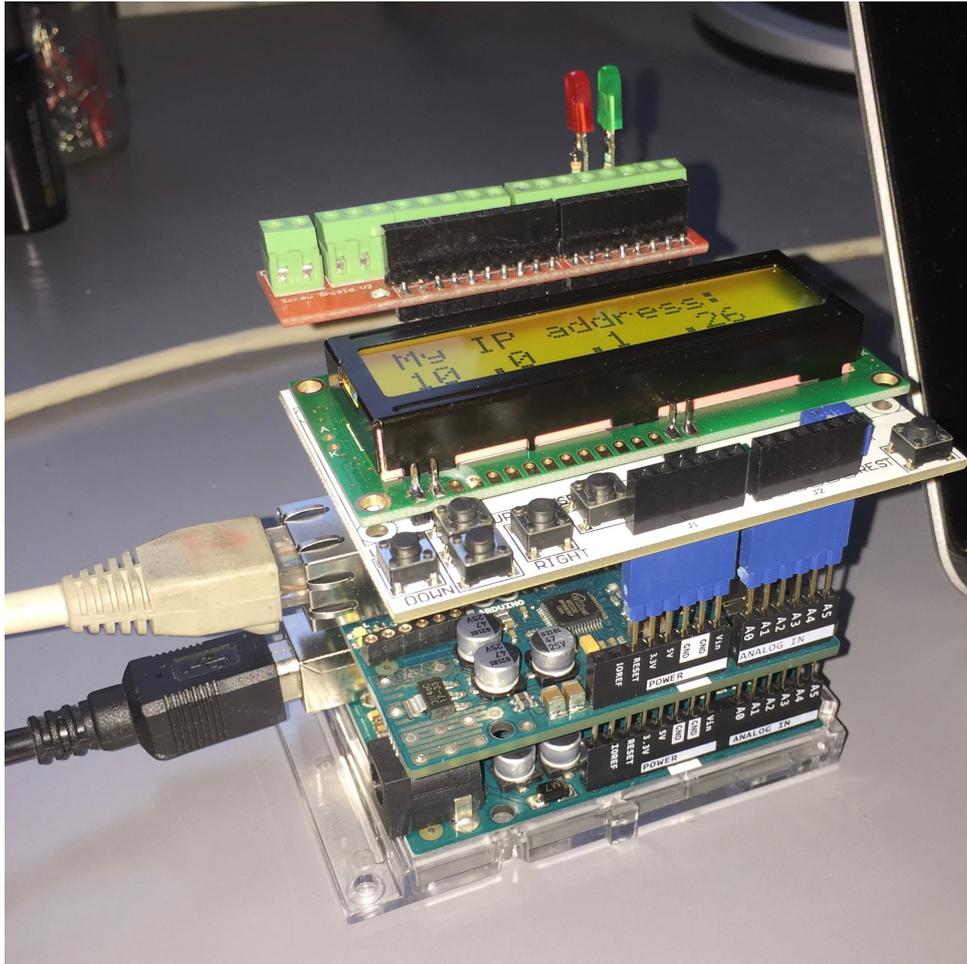
## DMX Protocol

<https://en.wikipedia.org/wiki/DMX512>

The serial RS485 DMX signal consist of a maximum 512 bytes data.

A data Packet is preceded by a Break, signal is low for minimum time of 88 uSec and a Mark After Break, signal is high for a minimum time of 8 uSec, followed by a packet of maximum of 512 data bytes.

# Pictures Arduino Project #1



## Arduino Project #2

<http://www.jgelectronics.nl/art-net-dmx/arduino-project-2.html>

Arduino Hardware Set Up #1

Arduino Software Sketch: 20171212\_Art-Net\_UDP\_Recv.ino

Apple MacBook App: Remote DMX free available at the Apple App Store

Apple iPhone, iPad App: Remote DMX available at the Apple App Store

YouTube Movie: [https://www.youtube.com/](https://www.youtube.com/watch?v=eOQIOQs4Ug0&list=PLnkzrtwChEibzial2QR-f2IpWDSbUuTdw&index=3)

[watch?v=eOQIOQs4Ug0&list=PLnkzrtwChEibzial2QR-f2IpWDSbUuTdw&index=3](https://www.youtube.com/watch?v=eOQIOQs4Ug0&list=PLnkzrtwChEibzial2QR-f2IpWDSbUuTdw&index=3)

Follow the steps described in Remote DMX Manual chapter “Step by step Art-Net™ Network set up on MacBook™” to set up an ethernet location.

If the Art-Net™ location is applied switch Wi-Fi off and connect your MacBook™ with an Ethernet cable with the Arduino Ethernet Shield. The Art-Net™ Universes and Sub-Net in the sketch are set as follows: SubNet = 0 and Universe = 1. You can alter this values in the sketch if you like but these values must be the same as in the DMX Remote app.

In the set up screen of the DMX Remote App set Universe 1 (DMX channel 1 - 512) to one and set Universe 2 (DMX channel 513 - 1024) to two. Make sure that Universe 1 is set to a different number than Universe 2 because if they have the same number DMX channel 1 and DMX channel 513 will both be received which causes the value of DMX channel 1 in the Arduino to alternate between the value of DMX channel 1 and DMX channel 513.

Set Sub-Net 1 and 2 to zero.

Set the broadcast IP address to 255.255.255.255.

Select in the Remote DMX screen DMX address 1 and press ON. The control LED's on the Arduino must go on if the value of DMX channel 1 reaches a value of more than 235 and go off if the value of DMX channel 1 reaches a value of less than 20. You can alter this values in the sketch if you like.

On the LCD Screen you can read the values of DMX channel 1 to 4. You can alter the DMX channels that are displayed in the LCD in the sketch if you like.

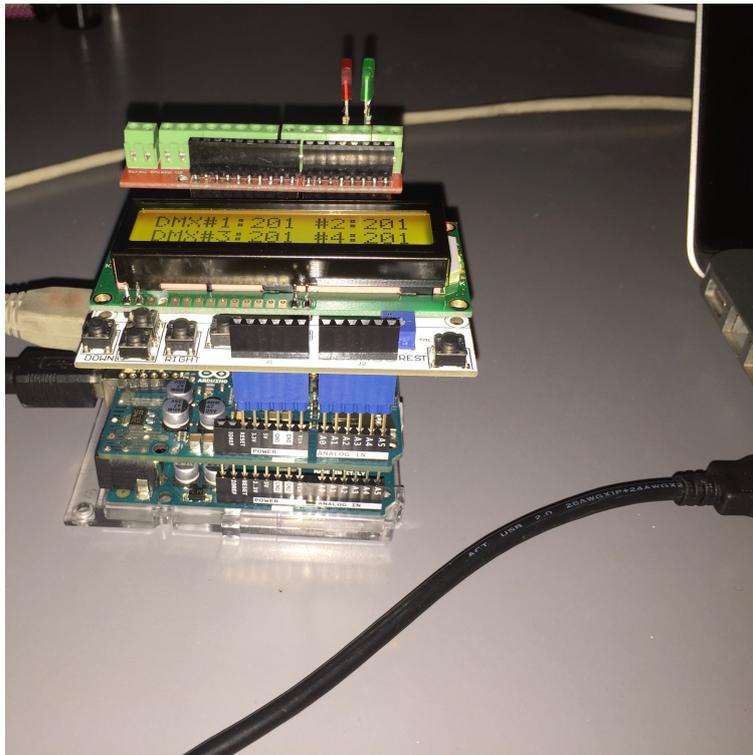
If you have a WiFi router you can unplug the ethernet cable from the MacBook™ and connect it to the router and switch WiFi on, on the MacBook™. See chapter “WiFi Router set up” on how to set up an Apple Airport Express. With the Remote DMX app for iPhone and iPad you can control the Arduino.

In the set up screen of the MacBook, iPhone, iPad app set the broadcast IP address to 255.255.255.255 and the Art-Net SubNet 1 and 2 to zero, Universe 1 to one and Universe 2 to two.

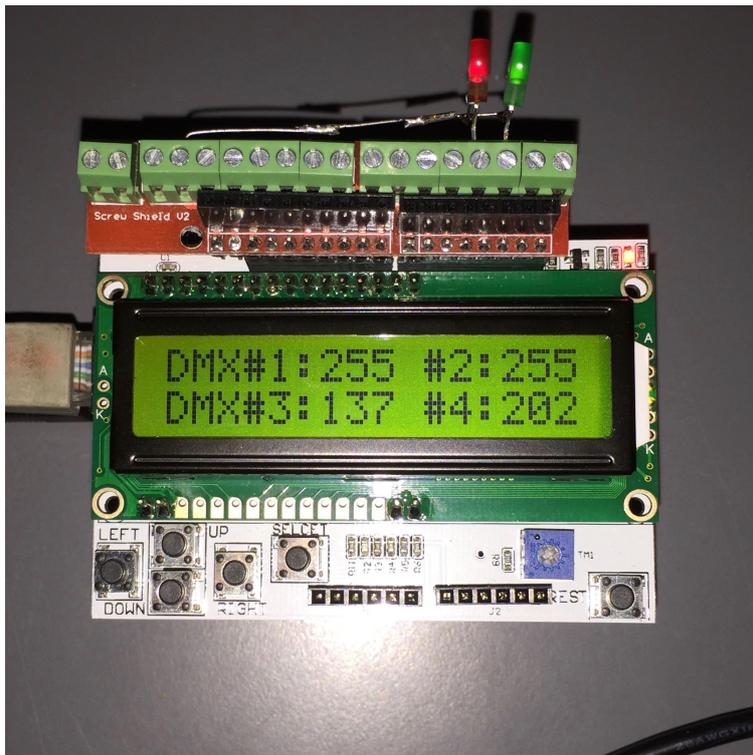
Set up the WiFi of your iPhone or iPad as described in chapter “Step by step Art-Net™ Network set up with iPhone™ or iPad™”.

DMX channel 1 to 4. DMX values set with the slider on the app to different values.

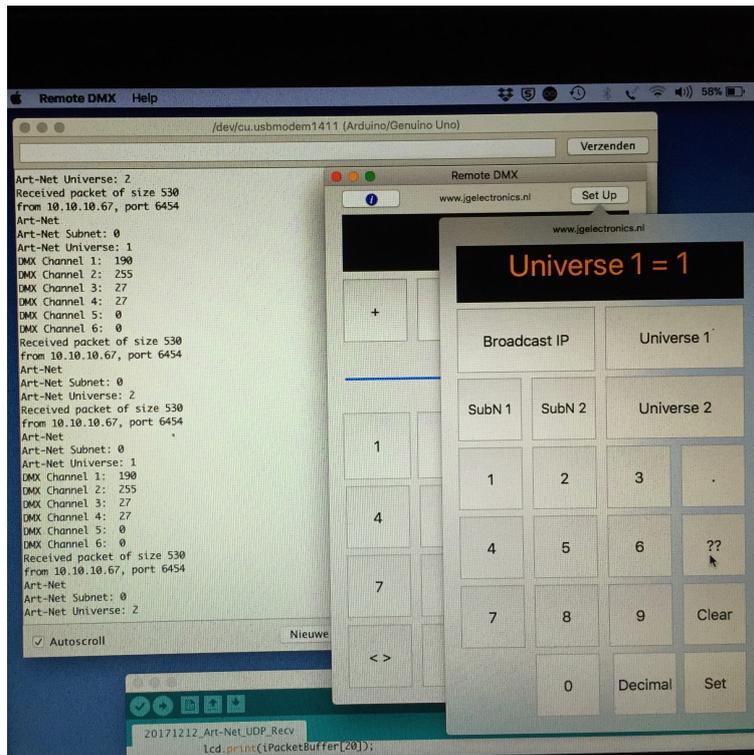
DMX channel 1 and 2 set to 255, green and red LED's output pin 2 and 3 are on.



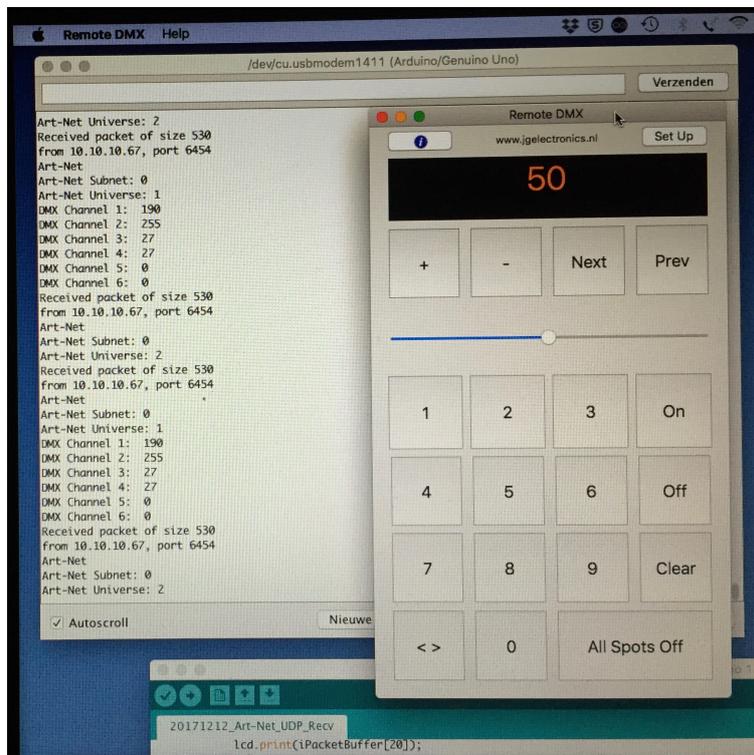
Ethernet connected to MacBook™, USB cable connected to MacBook for serial monitor



DMX channel 1 to 4. DMX values set with the slider on the app to different values. DMX channel 1 and 2 set to 255, green and red LED 's output pin 2 and 3 are on.



In the Remote DMX app for MacBook in the Set Up Screen set Broadcast IP to 255.255.255.255. SubN1 and SubN2 to zero.



Values in serial monitor Arduino IDE. Received Art-Net, Subnet = 0, Universe = 1, DMX channel 1 to 6. Then a new packet with Universe = 2, then again a packet with Universe 1

## Arduino Project #3

<http://www.jgelectronics.nl/art-net-dmx/arduino-project-3.html>

Arduino Hardware Set Up #2

Arduino MKR1000

<https://store.arduino.cc/arduino-mkr1000>

LCD Shield: 16 x 2 LCD Keypad

<https://www.sparkfun.com/products/13293>

LED's:

connected to output ? and ? of the Arduino MKR1000

Arduino Software Sketch:

???.ino

Apple MacBook App:

Remote DMX free available at the Apple App Store

Apple iPhone, iPad App:

Remote DMX available at the Apple App Store

## Arduino Project #4

<http://www.jgelectronics.nl/art-net-dmx/arduino-project-4.html>

Arduino Hardware Set Up #3

Arduino UNO Rev3

<https://store.arduino.cc/arduino-uno-smd-rev3>

LCD Shield: 16 x 2 LCD Keypad

<https://www.sparkfun.com/products/13293>

Encoder Interface: <http://www.jgelectronics.nl/art-net-dmx/arduino-project-4.html>

Absolute SSI Encoder:

Hengstler AC58-1213AK

Connect the LCD Shield with all pins to the Arduino UNO. You can control the intensity of the back light with a block wave at pin 10 if you wish but in the sketch pin 10 is not used.

The SSI encoder should be connected to the Arduino via an interface that converts the complimentary balanced signal of the data from the encoder to an unbalanced signal and the unbalanced clock pulse from the Arduino to a balanced signal.

The multi turn absolute SSI encoder used with the software is a Hengstler AC58-1213AK with a data format of 25 bits. 11 Bits for the current continuous angular distance in number of revolutions and 13 bits for the continuous angular distance in one rotation. The resolution is very high because one rotation is divided in 8192 steps. The data format is reflected binary code (RBC) also know as Gray code. [https://en.wikipedia.org/wiki/Gray\\_code](https://en.wikipedia.org/wiki/Gray_code)

Because of speed issues the program is written in C++ with the Atmel and not in the Arduino Sketch. We use an Atmel-ICE to upload the code to the Arduino. Be aware that the software starts at address 2048 in the Arduino other wise you overwrite the boot-loader program in the Arduino. It's also possible to upload Arduino sketches to an Arduino with the Arduino IDE and the Atmel-ICE. You don't need the boot-loader program any more in the Arduino and that gives you 2048 extra program memory space. We often use cheap Arduino clones that don't have an Arduino boot-loader program and program those with the Atmel-ICE and so we can recover the cost of the Atmel-ICE.

