

USB DMX 512 контроллер на основе Arduino своими руками

Создать свой [dmx](#) 512 контроллер не составит труда, на базе микроконтроллера Arduino.

Технические характеристики

В конфигурации, мы имеем 6 каналов с переменными значениями, предусмотренные 6 потенциометров, соединенных с Arduino на аналоговых входах, 10-битовые значения сводятся к 8 битам (0-255 которые используются [DMX](#)), и 12 каналов с включения-выключения значения с 12 кнопками, подключенных к цифровых входов Arduino, цифровые входы с использованием Arduino внутренние резисторы подтяжки, так что если кнопка нажата входное значение 0, и, если он свободен входное значение 1.

Общая информация

Исходный код открыт, и выложен в сети интернетах в открытом доступе.

Мы будем использовать **Arduino Mega**, или **Arduino Uno** для основной базы контроллера.

Известные проблемы:

Когда мы будем компилировать исходный код и получим ошибку: *'ArduinoDmxN' was not Declared In This scope*:

Проверьте, правильно ли настроен правильный тип правления в **Arduino IDE**, меню Сервис > доска (с Arduino Nano может использовать только один **USART = ArduinoDmx0 = USART0**)

Перед тем как перепрограммировать Arduino, необходимо отключить его от общей сборки.

Исходный код и библиотеки

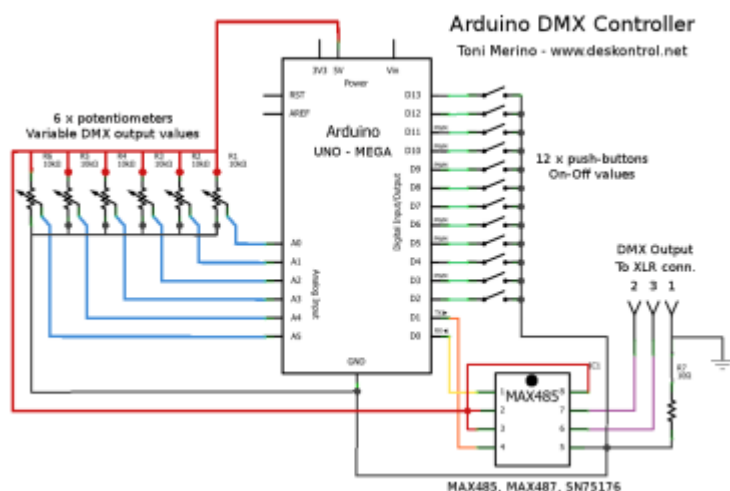
Подключаемая библиотека [arduino-four.zip](#)

Сборка

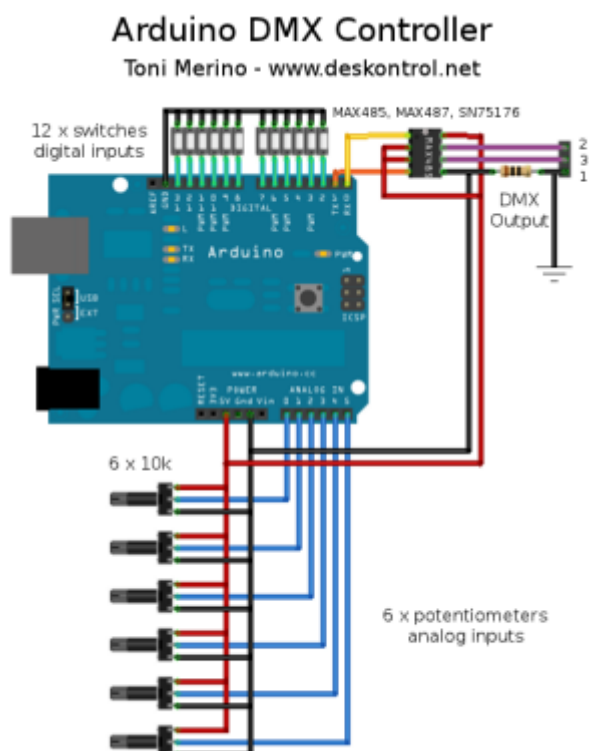
Сборка состоит из микроконтроллера

- 1 Arduino UNO(MEGA)
- 1 Маломощного трансивера, [MAX485](#)
- 1 резистора на 220khm
- 6 потенциометров(димеров)

- 12 кнопок.



1. Подключаем к входам ардуино с 2 по 13 выходы кнопки, а к gnd цепляем землю.
2. К аналоговым входам подключаем потенциометры с 1 по 5 выходы. Подключаем землю и выход 5V. Как показано на рисунке.
3. Подключаем к входу и выходу RX TX микросхему [MAX485](#), как вариант можно использовать аналог MAX487 или SN75176 к земле цепляем резистор на 220 килоом. Выход с микросхемы и есть управляющий сигнал [DMX](#)



Прошивка контроллера

Будем использовать 1 библиотеку.

Примеры

Пример тестового кода прошивки микроконтроллера.

Пример кода №1

[ArduinoDMXExample1.ino](#)

```
//*****
//*****
#include <lib_dmx.h>  // Подключаем скаченную нами библиотеку

//*****
//*****
//          Пример кода №1 *** Экспериментальный***
//*****
//*****
#define    DMX512      (0)    // (250 kbaud - 2 от 512 каналов) Стандарт
USITT DMX-512
#define    DMX1024     (1)    // (500 kbaud - 2 от 1024 каналов) Не стандарт,
проверка.
#define    DMX2048     (2)    // (1000 kbaud - 2 от 2048 каналов) called
by manufacturers DMX1000K, [[:wiki:dmx_512|DMX]] 4x or
[[:wiki:dmx_512|DMX]] 1M ???

void setup()
{
    // configurar pines arduino del 2 al 13 como entradas con pullup,
    (cuando se pulsa el boton = 0 si no = 1)
    // configure arduino pins 2 to 13 as inputs with pullup, (button
    pressed = 0, button free = 1)
    for (int i=2;i<=13;i++)
    {
        pinMode(i,INPUT);           // pines como entradas
                                     // pins as inputs
        digitalWrite(i, HIGH);      // activar resistencias pullup
internas
                                     // turn on pullup internal resistors
    }

    ArduinoDmx0.set_tx_address(1);  // poner aqui la direccion de
inicio de [[:wiki:dmx_512|DMX]]
                                     // put here [[:wiki:dmx_512|DMX]]
start address

    ArduinoDmx0.set_tx_channels(100); // poner aqui el numero de
canales a transmitir
                                     // put here the number of
```

```
[[[:wiki:dmx_512|DMX]] channels to transmit

  ArduinoDmx0.init_tx(DMX512);          // iniciar transmision universo
0, modo estandar DMX512                  // starts universe 0 as TX,
                                         standard mode DMX512
} //end setup()

void loop()
{
  // seis entradas con potenciómetros que envían valores
  [[[:wiki:dmx_512|DMX]] entre 0 y 255 a los canales 1 al 6
  // six analog inputs with potentiometers, sending values from 0 to
  255, to [[[:wiki:dmx_512|dmx]] output channels 1 to 6

  ArduinoDmx0.TxBuffer[0] = scale(analogRead(0)); // copiar valor de la
  entrada analogica 0 al canal DMX 1
                                         // copy value from
  analog input 0 to DMX channel 1
  ArduinoDmx0.TxBuffer[1] = scale(analogRead(1)); // copiar valor de la
  entrada analogica 1 al canal [[[:wiki:dmx_512|DMX]] 2
                                         // copy value from
  analog input 1 to [[[:wiki:dmx_512|DMX]] channel 2
  ArduinoDmx0.TxBuffer[2] = scale(analogRead(2)); // copiar valor de la
  entrada analogica 2 al canal DMX 3
                                         // copy value from
  analog input 2 to DMX channel 3
  ArduinoDmx0.TxBuffer[3] = scale(analogRead(3)); // copiar valor de la
  entrada analogica 3 al canal [[[:wiki:dmx_512|DMX]] 4
                                         // copy value from
  analog input 3 to [[[:wiki:dmx_512|DMX]] channel 4
  ArduinoDmx0.TxBuffer[4] = scale(analogRead(4)); // copiar valor de la
  entrada analogica 4 al canal DMX 5
                                         // copy value from
  analog input 4 to DMX channel 5
  ArduinoDmx0.TxBuffer[5] = scale(analogRead(5)); // copiar valor de la
  entrada analogica 5 al canal [[[:wiki:dmx_512|DMX]] 6
                                         // copy value from
  analog input 5 to [[[:wiki:dmx_512|DMX]] channel 6

  if (digitalRead(2) == LOW)             // pulsador en pin 2 apretado //
  push-button on pin 2, is pressed
    ArduinoDmx0.TxBuffer[6] = 255; // enviar 255 al canal
  [[[:wiki:dmx_512|DMX]] 7 // send value 255 to [[[:wiki:dmx_512|DMX]]
  channel 7
  else
    ArduinoDmx0.TxBuffer[6] = 0; // si no enviar 0 //
  push-button free, send 0

  if (digitalRead(3) == LOW)             // pulsador en pin 3 apretado
```

```

    ArduinoDmx0.TxBuffer[7] = 255; // enviar 255 al canal DMX 8
else
    ArduinoDmx0.TxBuffer[7] = 0;    // si no enviar 0

    if (digitalRead(4) == LOW)      // pulsador en pin 4 apretado
        ArduinoDmx0.TxBuffer[8] = 255; // enviar 255 al canal
[[:wiki:dmx_512|DMX]] 9
else
    ArduinoDmx0.TxBuffer[8] = 0;    // si no enviar 0

    // añadir aqui hasta el pin 13
    // add here the others inputs

} //end loop()

uint8_t scale(uint16_t value) // scale values from 10 bits to 8 bits
{
    if(value > 1023) // test for 10 bits limit
        value = 1023;

    return (value >> 2); // scale
} //end scale()

//*****
*****

```

Пример кода №2

ArduinoDMXExample2.ino

```

/*****
*****
*
*   Title           : Controlador [[:wiki:dmx_512|DMX]] con Arduino
*   Version         : v 0.1
*   Last updated    : 28.08.2012
*   Target          : Arduino mega 2560, Arduino mega 1280, Arduino
UNO, Arduino nano
*   Author          : Toni Merino - merino.toni at gmail.com
*   Web             : www.deskontrol.net/blog
*
*
*   Ejemplo de codigo para el mini-controlador [[:wiki:dmx_512|DMX]]
con Arduino:
*   Este codigo lee los valores de 6 potenciómetros conectados a 6
entradas analogicas y escribe cada valor en la salida
[[:wiki:dmx_512|DMX]]
*   solo cuando se presiona el pulsador correspondiente. (canales de

```

```
salida 1 al 6)
*
*****
*****/
#include <lib_dmx.h> // libreria [[:wiki:dmx_512|DMX]] 4 universos
// deskontrol four universes [[:wiki:dmx_512|DMX]] library -
www.deskontrol.net/blog

//*****
*****
//
New [[:wiki:dmx_512|DMX]] modes ***
EXPERIMENTAL ***
//*****
*****
#define DMX512 (0) // (250 kbaud - 2 to 512 channels)
Standard USITT DMX-512
#define DMX1024 (1) // (500 kbaud - 2 to 1024 channels)
Completely non standard - TESTED ok
#define DMX2048 (2) // (1000 kbaud - 2 to 2048 channels)
called by manufacturers DMX1000K, [[:wiki:dmx_512|DMX]] 4x or
[[:wiki:dmx_512|DMX]] 1M ???

uint8_t buffer[6]; // buffer [[:wiki:dmx_512|DMX]] data

void setup()
{
    // configurar pines arduino del 2 al 13 como entradas con pullup,
    (cuando se pulsa el boton = 0 si no = 1)
    // configure arduino pins 2 to 13 as inputs with pullup, (button
    pressed = 0, button free = 1)
    for (int i=2;i<=13;i++)
    {
        pinMode(i,INPUT); // pines como entradas
                           // pins as inputs
        digitalWrite(i, HIGH); // activar resistencias pullup
internas
                           // turn on pullup internal resistors
    }

    ArduinoDmx0.set_tx_address(1); // poner aqui la direccion de
inicio de [[:wiki:dmx_512|DMX]]
                           // put here [[:wiki:dmx_512|DMX]]
start address

    ArduinoDmx0.set_tx_channels(100); // poner aqui el numero de
canales a transmitir
                           // put here the number of
[[:wiki:dmx_512|DMX]] channels to transmmit

    ArduinoDmx0.init_tx(DMX512); // iniciar transmision universo
```

```
0, modo estandar DMX512                                     // starts universe 0 as TX,
                                                             // standard mode DMX512
} //end setup()

void loop()
{
    // seis entradas con potenciómetros que envían valores
    // entre 0 y 255 a los canales 1 al 6, cuando se
    // presiona el pulsador
    // six analog inputs with potentiometers, sending values from 0 to
    // 255, to output channels 1 to 6, when switch is
    // pressed

    buffer[0] = scale(analogRead(0)); // copiar valor de la entrada
    // analógica 0 al canal DMX 1
    // copy value from analog input 0
    // to DMX channel 1
    buffer[1] = scale(analogRead(1)); // copiar valor de la entrada
    // analógica 1 al canal [[:wiki:dmx_512|DMX]] 2
    // copy value from analog input 1
    // to [[:wiki:dmx_512|DMX]] channel 2
    buffer[2] = scale(analogRead(2)); // copiar valor de la entrada
    // analógica 2 al canal DMX 3
    // copy value from analog input 2
    // to DMX channel 3
    buffer[3] = scale(analogRead(3)); // copiar valor de la entrada
    // analógica 3 al canal [[:wiki:dmx_512|DMX]] 4
    // copy value from analog input 3
    // to [[:wiki:dmx_512|DMX]] channel 4
    buffer[4] = scale(analogRead(4)); // copiar valor de la entrada
    // analógica 4 al canal DMX 5
    // copy value from analog input 4
    // to DMX channel 5
    buffer[5] = scale(analogRead(5)); // copiar valor de la entrada
    // analógica 5 al canal [[:wiki:dmx_512|DMX]] 6
    // copy value from analog input 5
    // to [[:wiki:dmx_512|DMX]] channel 6

    if (digitalRead(2) == LOW) // pulsador en pin 2 apretado
    // push-button on pin 2, is pressed
        ArduinoDmx0.TxBuffer[0] = buffer[0]; // enviar valor potenciómetro
        // al canal [[:wiki:dmx_512|DMX]] 1 // send analog value to
        // [[:wiki:dmx_512|DMX]] channel 1
    else
        ArduinoDmx0.TxBuffer[0] = 0; // si no enviar 0
    // push-button free, send 0

    if (digitalRead(3) == LOW) // pulsador en pin 3 apretado
    // push-button on pin 3, is pressed
        ArduinoDmx0.TxBuffer[1] = buffer[1]; // enviar valor potenciómetro
```

```
al canal DMX 2 // send analog value to DMX channel 2
else
    ArduinoDmx0.TxBuffer[1] = 0; // si no enviar 0
// push-button free, send 0

if (digitalRead(4) == LOW)
    ArduinoDmx0.TxBuffer[2] = buffer[2];
else
    ArduinoDmx0.TxBuffer[2] = 0;

if (digitalRead(5) == LOW)
    ArduinoDmx0.TxBuffer[3] = buffer[3];
else
    ArduinoDmx0.TxBuffer[3] = 0;

if (digitalRead(6) == LOW)
    ArduinoDmx0.TxBuffer[4] = buffer[4];
else
    ArduinoDmx0.TxBuffer[4] = 0;

if (digitalRead(7) == LOW)
    ArduinoDmx0.TxBuffer[5] = buffer[5];
else
    ArduinoDmx0.TxBuffer[5] = 0;

} //end loop()

uint8_t scale(uint16_t value) // scale values from 10 bits to 8 bits
{
    if(value > 1023) // test for 10 bits limit
        value = 1023;

    return (value >> 2); // scale
} //end scale()

//*****
*****
```

Заключение

Мы могли бы, например, использовать EEPROM¹⁾ чтобы хранить значения DMX в EEPROM и извлекать для загрузки сцен ...

Где взять комплектующие

Все необходимое можно недорого заказать на алиэкспресс ссылки приведены ниже:

Arduino UNO ~ 253-400 руб.

MAX485 RS-485 ~ 46-52 руб. партия минимум 10 штук.

Резисторы 220 кОм ~ 44-240 руб. партия минимум 100 штук.

Потенциометры ~ 240-600руб. партия минимум 10 штук.

Кнопки ~ 200руб. 25 кнопок.

{{(rater>id=17|name=Понравилась статья?Удели секунду!Отцени!|type=rate|trace=ip)}}

1)

Энергонезависимая память устройства

From:

<https://dmx-512.ru/> - **DMX512.RU Управление светом**

Permanent link:

https://dmx-512.ru/zheleznaja_chast/dmx_512_kontroller_na_osnove_arduino?rev=1497038717

Last update: **2017/06/09 20:05**

