

Отправка DMX 512 данных с помощью Arduino

Еще один простой скетч, использующий Arduino и микросхему [MAX485](#) для отправки данных [DMX](#). По моему это самый простой скетч, чтобы разобраться как устроен протокол [DMX 512](#). На нем видно все задержки, где и как используется BREAK.

Скетч

```
/* [[wiki:dmx_512|DMX]] Shift Out for arduino - 004 and 005
 * -----
 *
 * Shifts data in [[wiki:dmx_512|DMX]] format out to [[wiki:dmx_512|DMX]]
enabled devices
 * it is extremely restrictive in terms of timing. Therefore
 * the program will stop the interrupts when sending data
 *
 * The elektronik foundation for [[wiki:dmx_512|DMX]] is RS 485, so you
have to use
 * a MAX-485 or a 75176.
 *
 * wiring for sending [[wiki:dmx_512|dmx]] with a MAX-485
1 - R0 - Receiver Output --- set to ground with a 100 ohm resistor
2 - RE - Receiver Output Enable -- set to ground
3 - DE - Driver Output Enable -- set to 5v
4 - DI - Driver Input -- Input from Arduino
5 - GND - Ground Connection -- set to ground -- reference for the
[[wiki:dmx_512|DMX]] signal --- ([[wiki:dmx_512|DMX]] pin 1)
6 - A - Driver Output / Receiver Input -- [[wiki:dmx_512|DMX]] Signal
(hot)----- ([[wiki:dmx_512|DMX]] pin 3)
7 - B - Driver Output / Receiver Input -- [[wiki:dmx_512|DMX]] Signal
inversion ( cold)----- ([[wiki:dmx_512|DMX]] pin 2)
8 - Vcc - Positive Supply -- 4,75V < Vcc < 5,25V

 * Every [[wiki:dmx_512|dmx]] packet contains 512 bytes of information (for
512 channels).
 * The start of each packet is marked by a start byte (shiftDmxOut(sig,0);),
 * you should always send all 512 bytes even if you don't use all 512
channels.
 * The time between every [[wiki:dmx_512|dmx]] packet is marked by a break
 * between 88us and 1s ( digitalWrite(sig, LOW); delay(10);)
 *
 * (c) 2006 by Tomek Ness and D. Cuartielles
 * K3 - School of Arts and Communication
 * fhp - University of Applied Sciences
 * <http://www.arduino.cc>
```

```
* <http://www.mah.se/k3>
* <http://www.design.fh-potsdam.de>
*
* @date: 2006-09-30
* @idea: Tomek Ness
* @code: D. Cuartielles and Tomek Ness
* @acknowledgements: Johny Lowgren for his \[\[:wiki:dmx\_512|DMX\]\] devices
*
*/
int sig = 11; // signal (hot / dmx pin 3)
int count = 0;
int swing = 0;
int updown = 0;
/* Sends a \[\[:wiki:dmx\_512|DMX\]\] byte out on a pin. Assumes a 16 MHz clock.
 * Disables interrupts, which will disrupt the millis() function if used
 * too frequently. */
void shiftDmxOut(int pin, int theByte)
{
    int theDelay = 1;
    int count = 0;
    int portNumber = port_to_output[digital_pin_to_port[pin].port];
    int pinNumber = digital_pin_to_port[pin].bit;
    // the first thing we do is to write te pin to high
    // it will be the mark between bytes. It may be also
    // high from before
    _SFR_BYTE(_SFR_IO8(portNumber)) |= _BV(pinNumber);
    delayMicroseconds(20);
    // disable interrupts, otherwise the timer 0 overflow interrupt that
    // tracks milliseconds will make us delay longer than we want.
    cli();
    // DMX starts with a start-bit that must always be zero
    _SFR_BYTE(_SFR_IO8(portNumber)) &= ~_BV(pinNumber);
    //we need a delay of 4us (then one bit is transfert)
    // at the arduino just the delay for 1us is precise every thing between 2
and 12 is jsut luke
    // to get excatly 4us we have do delay 1us 4 times
    delayMicroseconds(theDelay);
    delayMicroseconds(theDelay);
    delayMicroseconds(theDelay);
    delayMicroseconds(theDelay);

    for (count = 0; count < 8; count++) {

        if (theByte & 01) {
            _SFR_BYTE(_SFR_IO8(portNumber)) |= _BV(pinNumber);
        }
        else {
            _SFR_BYTE(_SFR_IO8(portNumber)) &= ~_BV(pinNumber);
        }
    }
}
```

```
delayMicroseconds(theDelay);
delayMicroseconds(theDelay);
delayMicroseconds(theDelay);
delayMicroseconds(theDelay);
theByte>>=1;
}

// the last thing we do is to write the pin to high
// it will be the mark between bytes. (this break is have to be between 8
us and 1 sec)
_SFR_BYTE(_SFR_IO8(portNumber)) |= _BV(pinNumber);
// reenale interrupts.
sei();
}
void setup() {
  pinMode(sig, OUTPUT);
  digitalWrite(13, HIGH);
}
void loop() {

  // sending the break (the break can be between 88us and 1sec)
  digitalWrite(sig, LOW);
  delay(10);
  //sending the start byte
  shiftDmxOut(sig,0);
  //sending the 512 bytes for the channels
  shiftDmxOut(sig, 150); //1
  shiftDmxOut(sig, 150); //2
  shiftDmxOut(sig, 150); //3
  shiftDmxOut(sig, 150); //4
  for (count = 1; count<=508; count++){
    shiftDmxOut(sig, 0);
  }
}
```

From:

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

Permanent link:

https://dmx-512.ru/zheleznaja_chast/dmx_512_send_dmx_data_arduino?rev=1490746657

Last update: **2017/06/09 20:04**

