

## **AN113: Transmission of DMX with AVR**

### **Content**

- Introduction
- Terms of Use
- DMX 512
- Code description

### **Introduction**

This application note describes the generation of a DMX signal using AVR microcontrollers. The state machine was written in C for the DMX Transceiver from Henne's Sites, but should also be portable to other circuits.

### **Terms of Use**

The state machines can be used in accordance with the gnu general public licence (GPL).

If GPL-compliant use is not possible, please contact the author.

## DMX 512

DMX 512 is a unidirectional differential serial protocol according to the RS485 standard with a transmission rate of 250kBaud. One byte consists of one start bit [4], eight data bits [14, 15] and two stop bits [7, 8]. There is no parity bit (8n2). Inter byte gaps [9] can be inserted between the bytes to allow slower receivers a reliable reception.

A bus (or universe) consists of one master and up to 32 slaves. The number of slaves can be increased by using splitters or boosters. A universe can have up to 512 channels. The transmission is initiated by a break [1] of at least 88 $\mu$ s followed by a mark-after-break [2] of at least 8 $\mu$ s. Afterwards, the start byte [14] is transmitted, which always has a zero value for DMX.

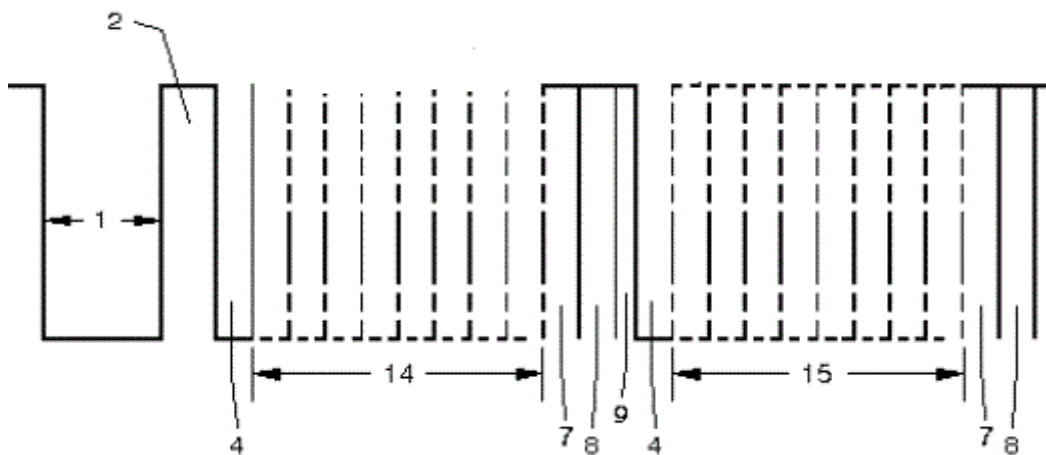


Figure 1: Sketch of the DMX protocol acc. to ANSI E1.11

## Code description

Besides the initialisation, the code consists of an interrupt service routine:

In the "USART Transmission Complete" ISR, it is first checked whether a break is to be sent. If yes, a zero byte is sent with a much lower baud rate, which the receivers interpret as a break with over 100µs and a mark after break of over 8µs.

To send the start byte after the break, the baud rate is set to 250kBit/s again and another zero byte is transmitted. Finally, the channel counter is reset.

After sending the start byte, the individual channels are transmitted. When the universe is completely transmitted, another break is started.

A special feature is the variable inter byte gap: The larger it is, the lower the DMX refresh rate and the longer the ISR blocks the AVR core. However, some receivers need this delay (the Transceiver does not!) to process the byte just received. The gap should therefore be chosen as small as necessary. In addition, at least 50 channels should be transmitted so that receivers can handle the refresh rate.

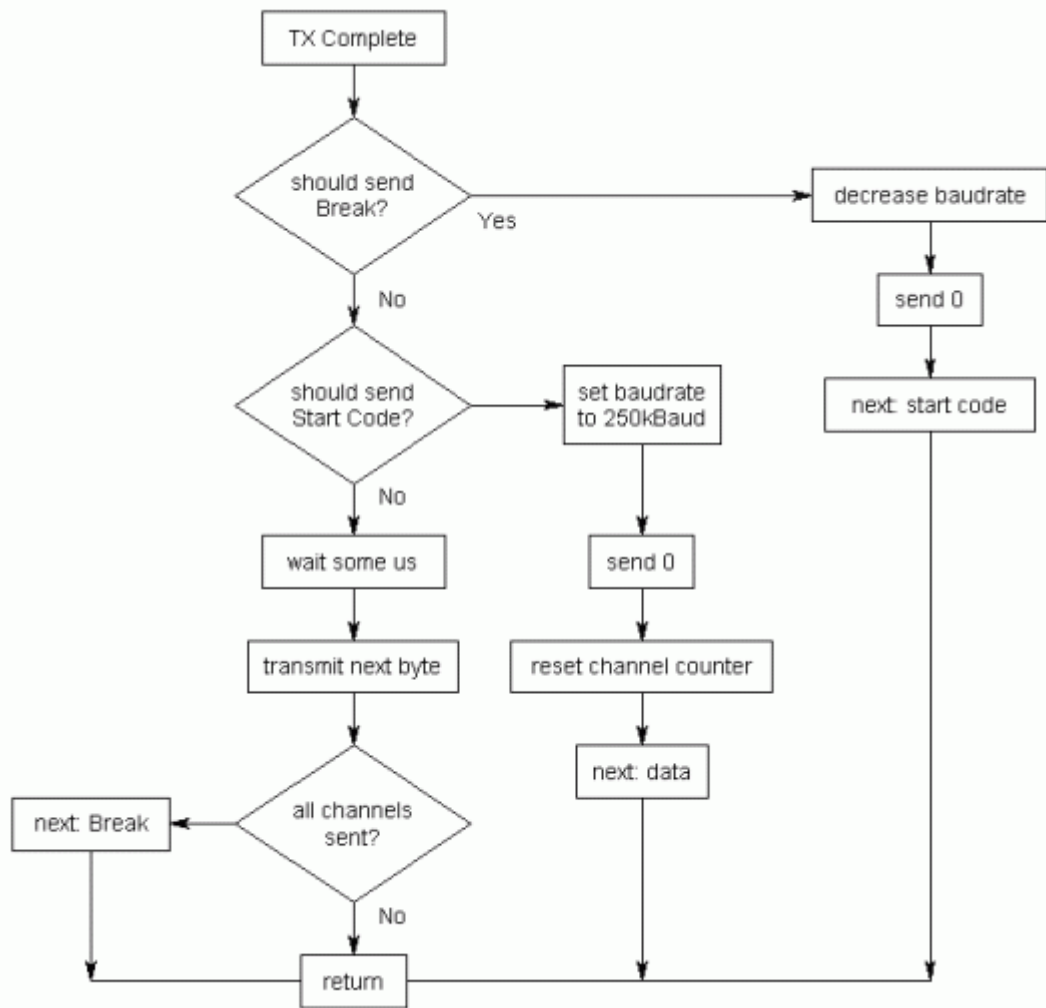


Figure 2: flow sheet of the state machine for DMX transmission

The code was written with the ATMEL Studio 7.

The following variables and constants must be defined in the header file:

```
#define F_CPU (8000000UL)           //oscillator freq.(typical 8MHz or 16MHz)

volatile uint8_t DmxField[50];     //array of DMX vals
```

F\_CPU is the crystal frequency in Hz.

DmxField is the array with the channels to be transmitted.

With "init\_dm", the USART is initialised for transmitting DMX data and the DmxField is flushed.