

Overview

This document describes the communication protocol „USP3“ used for the ChromoFlex modules. This is very simple protocol. We have tried explain it very detailed, so this document has become somewhat longer than originally expected...

Basic function - USP3

The used protocol is named USP3. This is a very simple, but reliable way to transfer data in packets. USP3 is not time critical and best suited for all kind of media (cables, wireless, infrared, ...). For the ChromoFlex only a small subset of USP3 is required.

USP3 can be used for point-2-point connections as well as for multidrop networks (like RS485, ChromoFlex-Bus, etc...)

USP3 was originally developed by the people from www.chromoflex.com for all kind of embedded communications.

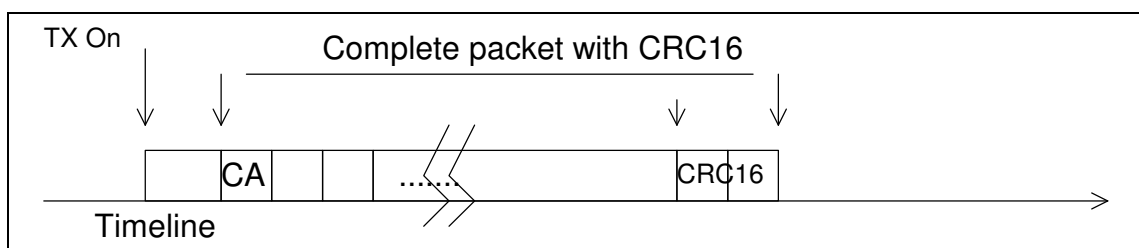
Sentinels

USP3 incorporates two sentinel characters: \$CA always signals the start of a new packed and hence will never occur inside a packed. Inside of packets \$CA is replaced by the 2 bytes sequence \$CB \$00, whereas the charater \$CB is replaced by the 2 Bytes sequence \$CB \$01.

So, detecting a packet is very easy: simply look for \$CA.

Preamble, Baudrate and Timing

Normally in a multidrop network, the bus is „free“, meaning the signal on the bus is undefined. If any network member wants to send a packet, it first has to turn on its transmitter. Because of the previous undefined network state, there must be a default state larger than 1 Bytes need to be transmitted. USP3 here recommends to wait circa 1.5-2 Byte times before transmitting the initial \$CA:



For the ChromoFlex a Baudrate of 9600 is used (Settings: 8 Statabits, 1 Stopbit, No Parity, commonly known as „9600 8N1“). Hence 1 Byte requires 10 bit times for transmission, this is 1.041667 msec/Byte. So the transmitter must wait between 1.5 and 2 msec before transmitting the \$CA.

Note: If any driver is constantly on, like a PC's RS232 if driving a bus with one or more ChomoFlex modules, the leading pause is not required...

Frame format

The frame format is very easy. In the following the structure is shown without the escaped characters:

- 1 Byte Frame Start sentinel \$CA
- 3 Bytes Address
 - Address \$000000 is the broadcast address.
 - Address \$000001 to \$0000FF is for „Group addresses“
(Note: Groups 1 - 8 can be selected for each module with the DIP switches)
 - Addresses \$000100 to \$FFFFFF may be used to address individual modules.
- 2 Bytes length (n) of the following datablock (max. \$FFFF, use \$0000 for none).
- 1 Byte command
- Optionally (n) data Bytes
- 2 Bytes CRC16 (error checksum)

Endianess format

All numbers are transmitted in „Big Endian“ format (= most significant Bytes first).

Error checking, CRC16

Each packet is protected with a CRC16 (industrial standard poly nominal, see apendix) Only error free packed are accepted. Packets with errors are discarded.. The CRC16 is calculated over all sent, unescaped data, including the Start sentinel. The CRC16 is initialised with \$FFF FF. (Note: since the first Byte is always constant, it should be mentioned, that the CRC16 will always be \$173F after the \$CA is inserted into the output buffer, as shown later).

The algorithm for the CRC16 is easy: The variable 'usp_crc' is continuously updated for each new byte. This could be either done with a table or (slightly slower) by some shift operations:

```
static UINT16  usp_crc; // initialise per packet with $FFFF.

void process_crc(UCHAR8 ucData){
    int i;
    usp_crc^=ucData;
    for(i=0;i<8;i++){ // Process each Bit
        if(usp_crc&1){ usp_crc >>=1; usp_crc^=0xA001;}
        else{          usp_crc >>=1; }
    }
}
```

Replies

No replies will be sent to any of the mentioned commands.

Retries

It is allowed to send a command as several repetitions

Transmission schematics

- Switch Transmitter On and wait 1.5 - 2 msec (this might be not necessary if a PC (RS232) is driving ChromoFlex a bus)
- Send \$CA
- Set variable 'usp_crc' to \$173F
- Pass all other Bytes of the packet (3 Bytes address, 2 Bytes len, 1 Byte command, optionally the data Bytes) to a send function. The send function calls 'process_crc()' for each Byte.
- Make a copy of 'usp_crc' and pass it to the send function as 2 Bytes (High Byte first, because of Big Endian format). Note: It is important to send the copy of 'usp_crc', because it is modified by the send function.
- Then switch transmitter Off.

The send function

It processes the CRC16 for each argument and escapes the characters \$CB and \$CB:

```
void serial_send_cooked(uchar sb){
    process_crc(sb);    // track CRC
    if (sb==0xCA) {
        serial_send_raw(0xCB); // Physically out
        serial_send_raw(0);
    }else if (sb==0xCB) {
        serial_send_raw(0xCB);
        serial_send_raw(1);
    } else{
        serial_send_raw(sb);
    }
}
```

Commands for the ChromoFlex

Currently there are only 2 commands important:

- Reset (= 254 or \$FE) : This command resets all modules with the matched address. It does not contain data, hence the len is 0. In total 9 bytes are sent.

Example:

This is a packet for a broadcast Reset. The address is \$000000, The CRC16 is 8c f0

```
ca 00 00 00 00 00 fe 8c f0
```

- Write to Registers (= 126 or \$7E): The ChromoFlex contains internally some registers that control the colours, transition times, ... In the data the register number is defined. An explanation of all important registers follows. This command contains up to 5 Bytes in the

datablock. The first byte in the datablock is the address.

Example:

The address is \$000000, The CRC16 is underlined This is a packet to write 4 Bytes (len in blue is \$0005), starting at address \$04 (in red). The 4 bytes are (in *italic* letters): 201 (== \$C9), 202 (== \$CA), 203 (== \$CB), 204 (== \$CC). The data Bytes \$CA and \$CB are transmitted as escaped characters:

ca 00 00 03 00 05 7e 04 c9 cb 00 cb 01 cc b2 8d

The ChromoFlex registers

Note: The ChromoFlex internally has reserved registers for 4 channels (R G B X). X is reserved, implemented are only the registers R G B.

All registers are of size Byte (UCHAR8):

Adr 0: Red_Level
Adr 1: Green_Level
Adr 2: Blue_Level
Adr 3 X_Level

The „Level“ is the current value of a channel (0 is „OFF“, 255 is full „ON“).
If different from the appropriate „Set“ it is adjusted periodically.

Adr 4: Red_Set
Adr 5: Green_Set
Adr 6: Blue_Set
Adr 7: X-Set

The „Set“ is the target value. Of course, to allow changes it must be >0.

Adr 8: Red_Increment
Adr 9: Green_Increment
Adr 10: Blue_Increment
Adr 11: X_Increment

The „Increment“ is the increment which is used to change the „Level“ until „Set“ is reached.

<GAP>

Adr 17: Track

The „Track“ determines the speed of the color transitions. It specifies the number of 1/100 seconds after which the „Level“ is adjusted to the „Set“. If set to 1 and „Increment“ is 1 too, a

complete change from Level 0 to 255 of any color needs 2.55 seconds. For slower changes it is recommended to set all „Increments“ to 1 and to set Track to a value of ≥ 1 .

Adr 18: Status

The Status Byte contains some bits for external control. „Status“ must be set to 1. In this case any on-board effect is stopped and only color changes (via „Level“, „Set“ and „Increment“ are performed).

<GAP>

Adr 21: Program_High

Adr 22: Program_Low

The „Program“ specifies which of the 8 programs of the Chromoflex should be started. Since the entry table starts at 200 and each entry is 3 bytes in size, the number of any program to start can easily be calculated as: „Program_High/_Low“ = $200 + 3 * \text{Program_Number}$ (Program_Number from 0..7 and 9). A special program is 9: It simply shuts all output down.

Note: In the calculation from above the High Byte „Program_High“ is always 0.

Example (commands are sent as broadcast):

To take over control over a ChromoFlex first Adr. 18 („Status“) should be set to 1. Since Adr 17 „Track“ is the predecessor of „Status“ it is best to set these two Register with one command. Here „Track“ is set to 1 too:

```
ca 00 00 03 00 03 7e 11 01 01 66 aa
```

In the next step the „Increments“ are all set to 1 (a transition time of 2.55 sec is wanted):

```
ca 00 00 03 00 05 7e 08 01 01 01 01 18 45
```

Now simply set the „Set“ values of colours from time to time and the ChromoFlex will automatically change (within max. 2.55 sec) to this colour.

With this packet the „Set“ values are set to 64 64 64 64:

```
ca 00 00 00 00 05 7e 04 40 40 40 40 a1 f5
```

To exit this mode, simply send a „Reset“ command to all ChromoFlex.

Hint: Setting the „Level“ with enabled „Set“ and „Increment“ can be used for some kind of „Flash“-Effect.