

OSCII-bot 0.3 Celphor 0.1

1. General

This document describes the changes from OSCII-bot 0.3 to the extensions in OSCII-bot 0.3 Celphor 0.1

2. Extensions

2. 1 Receiving sysex messages

Incoming sysex messages trigger the @midi code section (like 3 byte MIDI messages).

To distinguish between regular MIDI messages and sysex messages the variables msgdev and sysexdev are used.

If a regular MIDI message arrives, msgdev will be > 0. On arrival of a sysex message the variable sysexdev will be > 0.

The msg1..3 variables remain untouched on an incoming sysex message.

To read the sysex messages content two functions are provided:

sysexrecv_buf(device, buffer, maxlen)

and

sysexrecv_str(device, str)

Both functions return the effectively returned bytes, or 0 on error.

A more detailed description can be found below.

2. 2. Sending sysex messages

To send sysex messages, there are two functions available:

midisend_buf(device,buffer,length)

and

midisend_str(device, str)

Both functions return the effectively sent bytes, or 0 on error.

A more detailed description can be found below.

3. Function description

sysexrecv_buf(device, buffer, maxlen)

```
@midi

sysexdev > 0 ? (
    buf = 10000;
    maxlen = 65536;
    while ((rcvlen = sysexrecv_buf(sysexdev,buf,maxlen)) > 0) (
        ...
    );
);
```

Receives a sysex message from device to a buffer up to maxlen bytes. If there is no message waiting/remaining or an error occurred, 0 is returned. Otherwise the number of bytes totally read is returned.

If maxlen is smaller than the length of the MIDI message, the rest of the MIDI message will remain in the internal buffer for later reading using this function.

sysexrecv_str(device, str)

```
@midi

sysexdev > 0 ? (
    while ((rcvlen = sysexrecv_str(sysexdev,#str)) > 0) (
        ...
    );
);
```

Receives a sysex message from device to a string. If there is no message waiting/remaining or an error occurred, 0 is returned. Otherwise the number of bytes totally read is returned. A max of 16384 can be read into one string.

If maxlen is smaller than the length of the MIDI message, the rest of the MIDI message will remain in the internal buffer for later reading using this function.

midisend_buf(device,buffer,length)

```
buf = 1000;
buf[0] = 0xF0;
buf[1] = 0x37;
buf[2] = 0x11;
buf[3] = 0x11;
buf[4] = 0xF7;
midisend_buf(midi_out, buf, 5); // send a 5-byte message to device midi_out
```

Sends a variable length MIDI message. Can be used to send normal MIDI messages or sysex messages. When sending sysex, no leading 0xf0 and trailing 0xf7 bytes are prepended or appended. Returns the length sent, or 0 on error.

midisend_str(device, str)

```
midisend_str(midi_out, "\xF0\x43\x10\x3E\x18\x00\x01\x02\x00\xF7");  
// send the message in the second argument to midi device named midi_out
```

Sends a variable length MIDI message from [a](#) string. Can be used to send normal MIDI messages, or sysex messages. When sending sysex, no leading 0xf0 and trailing 0xf7 bytes are prepended or appended. Returns the length sent, or 0 on error.

4. Restrictions

This version is only available for Windows.

5. Source code

The source code is under GPL will be publicly available later. In the mean time you can request a copy at celphor@gmx.de