

## UDC Controller - SLSM3 Synthesizer Commands Luis Quintero, Arecibo Observatory

### 1 Introduction

All the commands must start with the same header (SYN - case sensitive) and followed by the ID number of the board. The ID numbers must be between 00 and 31 (two characters format). Commands without these parameters are ignored. The examples below show commands from the computer to the device >>, and the the device response <<. All command and responses terminate with Carriage Return (CR - \r - 0x0d).

### 2 Commands

The status request ? command will return the configuration of the SLSM3 synthesizer (four 24bit latch registers, little endian format). If the device is restarted, these values correspond to the default values stored in the EEPROM memory. This is the command syntax:

Command: SYNxx?

where xx: Synth. board ID, 00<=xx<=31

Response: synxxsXXXXXXXXYYYYZZZZZZWWWWLLL,  
where

s: multiple latches

XXXXXX: reference counter latch, ctrl. bits 00

YYYYYY: N counter latch, ctrl. bits 01

ZZZZZZ: function latch, ctrl. bits 10

WWWWW: initialization latch, ctrl. bits 11

LLL: Lock status bits. L: Locked, U: Un-Locked. Bit description from left to right:

- SLSM3 Lock Detect, Arduino pin No. PB1/9.
- TLSD Lock Loss, Arduino pin No. PB0/8.
- TLSD Lock Detect, Arduino pin No. PD7/7.

Example:

```
>> SYN01?
<< syn01s010200FFFFF1FFFFF0010203UUU
```

You can use the "R" command to print out default configuration (EEPROM) of the device:

Command: SYNxxR

Response: synxxsXXXXXXXXYYYYZZZZZZWWWWWixx

where i: is the ID response command

Example:

```
>> SYN01R
<< syn01sFFFFFF0FFFFF1FFFFF0FFFFF3i01
```

The L command is used to set only one latch:

Command: SYNxxLXXXXXX, where

xx: synth. board ID, 00<=x<=31

XXXXXX: 24bit latch value in hex

Response: synxxok

Example:

```
>> SYN01?
<< syn01s010200FFFFF1FFFFF0010203
>> SYN01LAAAAAA
<< syn01ok
>> SYN01?
<< syn01s010200FFFFF1AAAAAA010203
```

You can use the S command if you want to configure all the latches. Latch values are in hex and little-endian. You must enter the latches in order, according to the control bits.

Command: SYNxxSXXXXXXXXYYYYZZZZZZWWWWW, where

S: multiple latches

XXXXXX: reference counter latch, ctrl. bits 00

YYYYYY: N counter latch, ctrl. bits 01

ZZZZZZ: function latch, ctrl. bits 10

WWWWW: initialization latch, ctrl. bits 11

Response: synxxok

Example:

```
>> SYN01S000000111111222222333333
<< syn01ok
>> SYN01?
<< syn01s000000111111222222333333UUU
```

The W command is used if you want to store your actual configuration as default (EEPROM):

Command: SYNxxW

Response: synxxok

Example:

```
>> SYN01R
<< syn01sFFFFFF0FFFFF1FFFFF0FFFFF3i01
>> SYN01S000000111111222222333333
<< syn01ok
>> SYN01W
<< syn01ok
>> SYN01R
<< syn01s000000111111000000333333i01
```

Use the D command to configure the default (EEPROM):

Command: UDCxxD

Response: udcxxok

Example:

```
>> SYN01R
<< syn01s000000111111000000333333i01
>> SYN01D
<< syn01ok
>> SYN01?
<< syn01s000000111111000000333333
```

If you want to change the device ID, you can use the I command to change the device ID: WARNING: this command does not resolve ID conflicts if several devices are connected to the same bus (e.g. RS485/422).

Command: SYNxxIww, where

xx: actual device ID

ww: new device ID

Response: synnwok

Example:

```
>> SYN01I02
<< syn02ok
>> SYN02?
<< syn02s000000111111000000333333
```

If you do not know the device ID (e.g. first time programmed device), and you want to reset this value, you can send an I command to everybody. WARNING: only one device connected, this could produce ID conflict later.

Setup ID, read status and check EEPROM:

```
>> SYNXXI02
>> SYN02?
<< syn02s000000111111000000333333
>> SYN02R
<< syn01s000000111111000000333333i01
```

Store new ID in EEPROM (permanent change):

```
>> SYN02W
<< syn02ok
>> SYN02R
<< syn02s000000111111000000333333i02
```

### 3 Error Codes

The device is expecting digits for the ID. If there is an error, the device will return an error code 01 (udcxxERR01, xx: device ID). The example shows the a as not digit characters:

```
>> SYN01I0a
<< syn01ERR01
```

If the device ID is out of range, the device will return an error code 02. For example an ID change from 01 to 80, the valid range is between 00 and 31:

```
>> SYN01I33
<< syn01ERR02
```

The device will return an error code 03 if at least one character is not in the hex range. You can see in the following examples that U and G are not valid characters:

```
>> SYN01LFFFFFFU
<< syn01ERR03
>> SYN01S5566778899AABCCDDEEFFGG
<< syn01ERR03
```

The device will return an error code 04 if the latches are not in order in the S command:

```
>> SYN01SFFFFFF0FFFFF1FFFFF3FFFFF2
<< syn01ERR04
```

The latch setup command L only accepts 12 characters of configuration. If a different number is provided, the device returns an error code 09. For example:

```
>> SYN01L0123456
<< syn01ERR09
```

If you are trying to use the S command, and do not provide the exact number of parameters (4x24bits registers, 12 characters), the device will return an error code 10:

```
>> SYN01S00112233445566778899
<< syn01ERR10
```

The ID change command I only accepts eight (8) characters of configuration. If a different number is provided, the device returns an error code 08. For example:

```
>> SYN01I2
<< syn01ERR08
```

Commands related to status and EEPROM only accepts six (6) characters. If a different number is provided, the device returns an error code 07. Example:

```
>> SYN01
<< syn01ERR07
```

Note: this error message is disabled, message ignored. Finally, if the command does not exits, the device will return an error code 06:

```
>> SYN01K
<< syn01ERR06
```

#### 4 Examples

- Setup the last latch (control bits 0b11) to 0x010203:  
SYN01L010203
- Setup all ones of third latch (control bits 0b10):  
SYN01LFFFFFFE
- Multiple latch configuration examples:  
SYN01S000000111111222222333333  
SYN01S000008002315920012920013  
SYN01SFFFFFF0FFFFF1FFFFF2FFFFF3  
SYN01Saaaaa8aaaaa9aaaaaAaaaaaB