

anyfeed SXM-DD Serial Communication Interface

Revision 1.3

1. Interface Cable

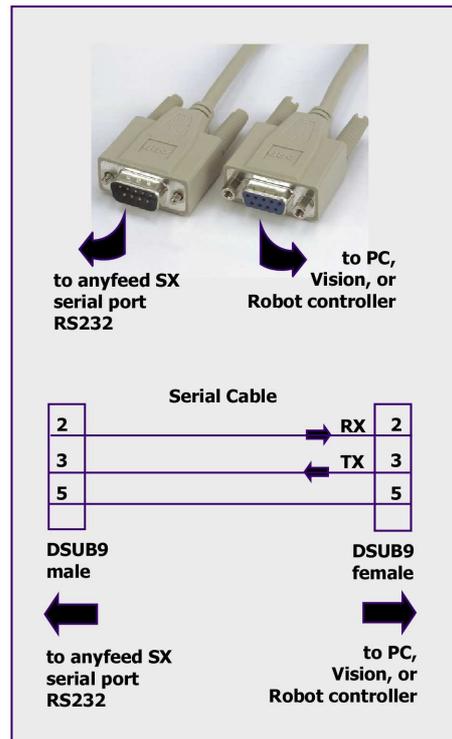


Figure 1: Serial cable specification

2. Serial Communication Setup

2.1 Setting up the communication port

Typical devices which communicate with the anyfeed SXM over a serial line are:

- PC Terminal
- Robot controller
- Smart Camera
- Vision System

On the device you use to communicate with the anyfeed SXM, setup the serial port as follows:

| | |
|---------------------|-------|
| Baudrate | 9'600 |
| Start bit | 1 |
| Stop bit | 1 |
| Parity | none |
| Flow control | none |

Table 1: Serial port setup

Install an RS232 cable as described in chapter 1 and connect the feeder to 24VDC.

3. anyfeed SXM Serial Communication

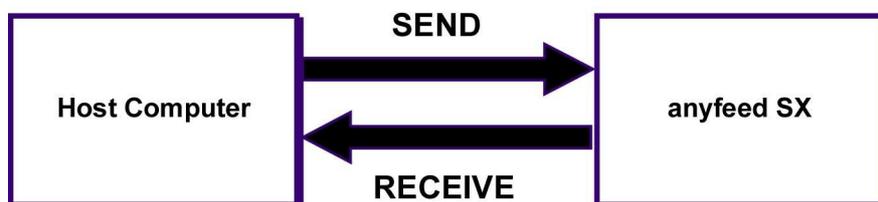
Below is list of all available anyfeed SXM commands. With each command there is a maximum of 2 associated parameters which influence that command. Some commands do not need any parameters as you can see in the table following.

During startup, the anyfeed SXM firmware assigns „reasonable“ default values to all parameters. So if you send a feeder motion command without having set any of its parameters, the feeder control system will apply its default parameters.

Once a feeder parameter has been changed from the host, the value is maintained until you power-cycle the feeder or send a „Restart firmware“ command. This means you can prepare the feeder for a given part by downloading all the parameters upfront and during operation you just send the action commands such as feed forward, feed backward, flip etc.

3.1 Sending commands to the anyfeed SXM

This chapter describes what commands can be *sent* to the anyfeed SXM and what effect they have. The computer *sending* commands to the anyfeed SXM is referred to as host computer. Chapter 3.3 describes how the anyfeed *responds* to received commands



Explanations for the following Table

| | |
|--|---|
| <speed> | integer in the range 0..10 |
| <turns> | integer in the range 1..127 |
| <cr> | Carriage return, ASCII code 13 (decimal) |
| > | Symbolizes the prompt of a text terminal window |
| <p>Note that commands shown in the syntax column of table 1 are case sensitive, which means x is not equal X or AB[1] is not ab[1] ! All characters shown in column „Syntax“ in table 2 must be lower case.</p> | |

Motion commands for anyfeed SXM

| Command + (Short form) | Description | Syntax | Explanations |
|-------------------------------------|---|---|---|
| Feed forward (ffwd) | Feeds parts forward | > x=1<cr> > ab[1]=<turns><cr> > ab[17]=<speed> x=17<cr> | Executes a ffwd motion Sets number of ffwd turns Changes ffwd speed |
| Feed backward (fbwd) | Feeds parts backward | > x=2<cr> > ab[2]=<turns><cr> > ab[18]=<speed> x=18<cr> | Executes a fbwd motion Sets number of fbwd turns Changes fbwd speed |
| Feed/flip forward (flipfwd) | Feeds part forward and flips them | > x=3<cr> > ab[3]=<turns><cr> > ab[19]=<speed> x=19<cr> | Executes a flipfwd motion Sets number of flipfwd turns Changes flipfwd speed |
| Feed/flip backward (flipbwd) | Feeds part backward and flips them | > x=4<cr> > ab[4]=<turns><cr> > ab[20]=<speed> x=20<cr> | Executes a flipbwd motion Sets number of flipbwd turns Changes flipbwd speed |
| Flip neutral (flip) | Flips parts without moving them substantially forward or backward | > x=5<cr> > ab[5]=<turns><cr> > ab[21]=<speed> x=21<cr> | Executes a flip motion Sets number of flip turns Changes flip speed/intensity |
| Dispense (dispense) | Adds parts from the bulk feeder into the pick area | > x=6<cr> > ab[6]=<turns><cr> > ab[22]=<speed> x=22<cr> | Executes a dispense motion Sets number of dispense turns Changes dispense speed |
| Purge (purge) | Feeds all parts backward out; purge gate must be opened manually | > x=7<cr> > ab[7]=<turns><cr> > ab[23]=<speed> x=23<cr> | Executes a purge motion Sets number of purge turns Changes purge speed |
| Initialize (init) | Initializes the feeders home position; | > x=16<cr> | Init is required before any other action can take place This command also resets all error flags |
| Stop (stop) | Stops current action and moves feeder to home position | > x=15<cr> | |

Note: Motion settings (turns and speed) are stored in volatile memory. Parameters remain active until they are overwritten by new values. Power cycle resets all Parameters to their default values.

Setup commands for anyfeed SXM

| Command | Description | Syntax | Explanations |
|--------------------------------|---|--|---|
| Report firmware version | Outputs the currently loaded firmware | > x=10<cr> | Feeder will output two strings, one for each drive. |
| Startup feeder firmware | Startup feeder firmware (this also happens automatically when the feeder is powered on) | > S RUN<cr> | Feeder will output m10<cr> m20<cr> as a greeting message. Which indicates that the feeder is online. |
| Restart firmware | Restarts feeder firmware; <u>all parameters are reset to default values</u> | > x=31<cr> | |
| Reset error | Resets error status and moves feeder to home position | > x=30<cr> | Must be used after feeder reported an error before new actions commands can be executed. |
| Set trigger output | There is a dedicated output which can be used to control camera triggering and/or an LED illumination device | > ab[12]=<mode> | Values for <mode>: 0: output is OFF 1: output is ON 2: output is strobed as described under command „set trigger delay“ and „set trigger interval“ |
| Set trigger interval | Sets the vision trigger interval in multiples of 50ms. | > ab[25]=30 x=25<cr> | Sets the trigger interval to 1500ms range ab[25]=10..100 default is 1000ms |
| Set trigger delay | Sets the vision trigger delay in multiples of 20ms. The trigger delay starts after a feeder action has completed. It is used to avoid taking pictures from parts still in motion. | > ab[26]=20 x=26<cr> | Sets the trigger delay to 400ms range ab[25]=1..100 default value is 200ms |
| Switch backlight on/off | Allows you to turn the backlight on or off. | > ab[27]=2 x=27<cr> > ab[27]=3 x=27<cr> | Turns the backlight off Turns the backlight on the backlight is on by default |

3.2 How the anyfeed SXM responds to commands (receive side)

For each action command line the feeder receives, either drive (flip or dispense drive or also referred to as drive 1 and drive 2) will respond as follows:

`mix<cr>` The *i* after the *m* indicates whether the message is coming from motor/drive 1 or 2 if *x* has a value of 1, command is valid/accepted; if *x* has a value of 0, the command completed successfully; *x* values larger than 1 represent an error number

Motor/Drive 1: that's the so called flip drive. With respect to part flow, that's the drive in front near the pickup area

Motor/Drive 2: that's the so called dispense drive. With respect to part flow, that's the motor in the rear underneath the bulk container.

3.3 Dialog examples

| Initializing the feeder | |
|---|---|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>x=16<cr></code> | Send the 'Init' command |
| <code>x=16<cr></code> | Echo of the 'Init' command |
| <code>m21<cr></code> | Motor/Drive 2 understood command and is now busy, indicated by the '1' |
| <code>m11<cr></code> | Motor/Drive 1 understood command and is now busy, indicated by the second '1' |
| <code>m10<cr></code> | Motor/Drive 1 completed action successfully indicated by '0' |
| <code>m20<cr></code> | Motor/Drive 2 completed action successfully |

| Feeding parts forward | |
|---|---|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>x=1<cr></code> | Send the 'feed forward' command |
| <code>x=1<cr></code> | Echo of the 'feed forward' command |
| <code>m21<cr></code> | Motor/Drive 2 understood command and is now busy, indicated by the '1' |
| <code>m11<cr></code> | Motor/Drive 1 understood command and is now busy, indicated by the second '1' |
| <code>m20<cr></code> | Motor/Drive 2 completed action successfully indicated by '0'; in this case (and other feed-commands) the 'dispense drive' does not execute a motion, but still confirms that it accepted the command and is 'alive'. A consistent drive reporting scheme is easier to handle by the receiving host computer and the additional communication overhead is minimal. |
| <code>m10<cr></code> | Motor/Drive 1 completed action successfully |

| Changing the flip speed/intensity | |
|--|---|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>ab[21]=10 x=21<cr></code> | Send the command line to set flip speed to 10 (max.) |
| <code><i>ab[21]=10 x=21<cr></i></code> | Echo of the command line to set flip speed to 10 (max.) |
| <code><i>m21<cr></i></code> | Drive 2 command understood |
| <code><i>m11<cr></i></code> | Drive 1 command understood |
| <code><i>m10<cr></i></code> | Drive 1 finished |
| <code><i>m20<cr></i></code> | Drive 2 finished |

| Setting the number of turns for flip forward motion | |
|--|---|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>ab[3]=6<cr></code> | Set the number of turns for feed/flip forward motion to 6 |
| <code><i>ab[3]=6<cr></i></code> | Echo of the command to set feed/flip forward turns to 6 |
| | No answers are output. Only the echo of the command will be received. The drives do NOT answer to this setting-command if it is issued without the motion-command ('x=3<cr>'). This can be done at startup for all turn-settings and then ,during production, only motion commands can be issued, since the parameters stay present in memory. |

| Feeder not initialized | |
|--|--|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>x=5<cr></code> | Send the 'flip' command to the feeder but the feeder has not been initialized before |
| <code><i>x=5<cr></i></code> | Echo of the 'flip' command |
| <code><i>m26<cr></i></code> | Drive 2 reports that it is not initialized |
| <code><i>m16<cr></i></code> | Drive 1 reports that it is not initialized |

| Feeder error/servo overload | |
|--|---|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>x=6<cr></code> | Send the 'dispense' command to the feeder |
| <code><i>x=6<cr></i></code> | Echo of the 'dispense' command |
| <code><i>m11<cr></i></code> | Drive 1: command understood and processing |
| <code><i>m21<cr></i></code> | Drive 2: command understood and processing |
| <code><i>m23<cr></i></code> | Drive 2: reports servo overload |
| <code><i>m17<cr></i></code> | Drive 1: stops motion and reports error state |

| Unknown Command | |
|---|--|
| Terminal window | Explanation |
| For clarity, echo and answer messages are in <i>bold/inclined</i> | |
| <code>x=9<cr></code> | Sends command '9' – an unknown command - to the feeder |
| <code>x=9<cr></code> | Echo of the command |
| <code>m22<cr></code> | Drive 2 reports that this is an unknown command |
| <code>m12<cr></code> | Drive 1 reports that this is an unknown command |

4. Recommendations for implementing communication with anyfeed SXM

We recommend that you download all parameters for a given part first. During runtime you can then only send short commands, such as:

feed forward, flip, feed backard etc.

To get started, we recommend you use a terminal emulator program, such as Hyperterminal that comes with each Windows installation and try out some of the commands directly by entering them in Hyperterminal and see the feeder's response.

Here is the setup again:

```
Baudrate      9'600
Start bit     1
Stop bit      1
Parity        none
Flow control  none
```

also,

```
Disable local echo
Append 'line feed' to each line send
Append 'line feed' to lines received
```

5. Error messages

| Error | Error Message | Details / possible cause |
|--------------|-------------------------------|--|
| 12 | Invalid command | Motor 1 received a command which is not supported |
| 13 | Motor 1 reports a servo error | Motor 1 is overloaded, drive blocked ... |
| 16 | Motor 1 not initialized | Motor 1 hasn't been initialized yet or lost it's state due to power cycle |
| 17 | Error state on Motor 1 | Servo error has occurred on Motor 1. No commands will be accepted until error state is reset. |
| 22 | Invalid command | Motor 2 received a command which is not supported |
| 23 | Motor 2 reports a servo error | Motor 2 is overloaded, drive blocked ... |
| 26 | Motor 2 not initialized | Motor 2 hasn't been initialized yet or lost it's state due to power cycle |
| 27 | Error state on Motor 2 | Servo error has occurred on Motor 2. No commands will be accepted until error state is reset. |