anyfeed SX Serial Communication Interface

Revision 2.1

1. Interface Cable

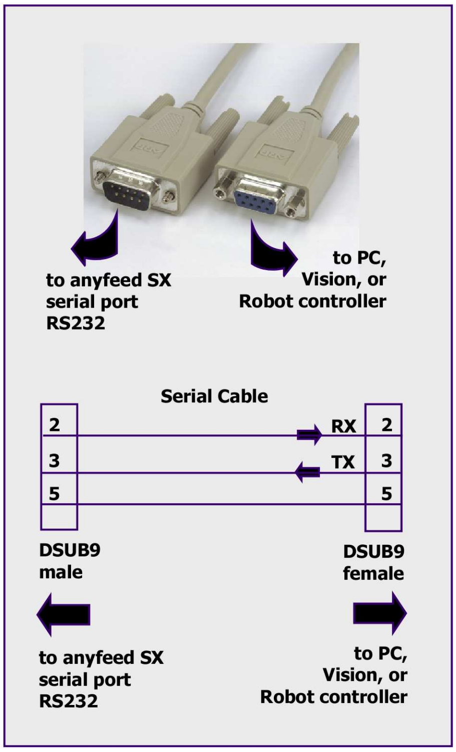

to anyfeed SX serial port RS232
to PC, Vision, or Robot controller
Serial Cable
2
3
5
RX
TX
2
3
5
DSUB9 male
DSUB9 female
to anyfeed SX serial port RS232
to PC, Vision, or Robot controller

Figure 1: Serial cable specification

2. Serial Communication Setup

2.1 Setting up the communication port

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

- PC
- Robot controller
- Smart Camera
- Vision System

On the device you use to communicate with the anyfeed SX, 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 SX Serial Communication

Below is list of all available anyfeed SX 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.

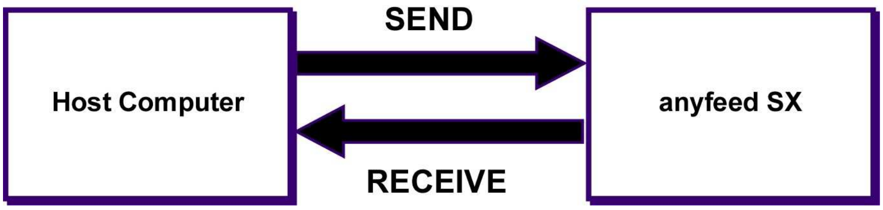
During startup, the anyfeed SX firmware assigns „reasonable‟ default values to all parameters. So if you send a feeder motion command without having set any of its parameters, the feeder control system will apply its default parameters (see chapter 3.2).

Once a parameter has been changed from the outside, it maintains its value 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 then, during production, you just send the action commands such as feed forward, feed backward, flip etc.

3.1 Sending commands to the anyfeed SX

This chapter describes what commands can be sent to the anyfeed SX and what effect they have.
The computer sending commands to the anyfeed SX is refered 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
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.	

Motion commands for anyfeed SX

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 (disp)	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 by getting reference from inductive sensor	> 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 or the feeder looses power. Power cycle resetts all Parameters to their default values.

Setup commands for anyfeed SX

Command	Description	Syntax	Explanations
Startup feeder firmware	Normally the firmware starts automatically at power-up. This command has the same effect as power-cycling the feeder and is only needed if the firmware crashed	> RUN<cr>	Feeder will output m10<cr> m20<cr> as a greeting message. Which indicates that the feeder is ready.
Restart firmware	Restarts feeder firmware; **all parameters are reset to default values**	> x=31<cr>	Feeder will output m10<cr> m20<cr>
Reset error	Resets error status and moves the feeder to the buffered 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 according to the following settings. (puls width is always 20ms)
Set trigger interval	Sets the vision trigger interval in multiples of 50ms. The interval starts over right after the trigger delay time.	> ab[25]=60 x=25<cr>	Sets the trigger interval to 3sec range ab[25]=10..100 (range intervall: 500...5000ms) default is 2000ms
Set trigger delay	Sets the vision trigger delay in multiples of 20ms. The trigger delay starts after a feeder action has completed.	> ab[26]=15 x=26<cr>	Sets the trigger delay to 300ms range ab[26]=1..100 (range intervall: 20...2000ms) default value is 200ms
Set prefeeder type	Some applications need an external prefeeder instead of the standard bulk-container. This value must match the hardware setup.	> ab[13]=<mode>	Set <mode> according to your feeder layout: 1: standard bulk-container 2: external prefeeder
Set prefeeder sensitivity (only applicable if prefeeder type is set to 2)	Sets the sensitivity of the light barrier that controls the external prefeeder in multiples of 50ms. If the light barrier is open during the specified time, the prefeeder starts to run during ON-time.	> ab[28]=32 x=28<cr>	Sets the sensitivity to 1.6sec range ab[28]=1..100 (range intervall: 50...5000ms) default is 1000ms
Set prefeeder ON-time (only applicable if prefeeder type is set to 2)	Sets the duration which the prefeeder is working in multiples of 50ms. After that time the light barrier checks again if more parts need to be supplied.	> ab[29]=24 x=29<cr>	Sets the ON-time to 1.2sec range ab[29]=1..100 (range intervall: 50...5000ms) default value is 1000ms
Set digital outputs	Allows you to change the state of feeder internal digital outputs	> ab[27]=0 x=27<cr> > ab[27]=1 x=27<cr> > ab[27]=2 x=27<cr> > ab[27]=3 x=27<cr> > ab[27]=4 x=27<cr> > ab[27]=5 x=27<cr>	Engages the dispense clutch Engages the flip clutch Turns the backlight off Turns the backlight on Moves the retainer gate up or turns ext. prefeeder OFF Moves the retainer gate down or turns ext. prefeeder ON

3.2 Parameter default values

```
ab[1]=6         'default feed forward turns
ab[2]=6         'default feed backward turns
ab[3]=4         'default flip forward turns
ab[4]=4         'default flip backward turns
ab[5]=4         'default flip turns
ab[6]=4         'default dispense turns
ab[7]=20        'default purge turns
ab[12]=2        'default trigger mode
ab[13]=1        'default prefeeder type (standard bulk-container)
ab[17]=4        'feed forward speed
ab[18]=4        'feed backward speed
ab[19]=4        'feed/flip forward speed
ab[20]=4        'feed/flip backward speed
ab[21]=4        'flip speed
ab[22]=4        'dispense speed
ab[23]=6        'purge speed
ab[25]=40       'equals 2000ms trigger interval
ab[26]=10       'equals 200ms trigger delay time
ab[28]=20       'equals 1000ms prefeeder sensitivity
ab[29]=20       'equals 1000ms prefeeder ON-tine
```

3.3 How the anyfeed SX responds to commands (receive side)

Each command line the feeder receivesis echoed back to the serial line. Additionally, all x-commands are answered by the two feeder-drives (drive 1 and drive 2, also refered to as 'flip' or 'dispense' drive). The response-messages always have the following format:

mix<cr>	The i after the m indicates whether the message is comming from drive 1 or 2. If x has a value of 1, the command is accepted and the drive is busy processing it. If x has a value of 0, the command completed successfully. x-values larger then 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. That's the motor in the rear underneath the bulk container.

3.4 Dialog examples

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

Feeding parts forward	
Terminal window	Explanation
For clarity, echo and answer messages are in ***bold/inclined***	
x=1<cr>	Send the 'feed forward' command
x=1<cr>	Echo of the 'feed forward' command
m11<cr>	Motor/Drive 1: understood command and is now busy, indicated by the second '1'
m21<cr>	Motor/Drive 2: understood command and is now busy, indicated by the '1'
m20<cr>	Motor/Drive 2: completed action successfully indicated by '0'. In this case (and other feed-commands) the 'dispense drive' does not execute any motion, but still reports acceptance and end of processing of the command to fulfill a consistent drive reporting scheme.
m10<cr>	Motor/Drive 1: completed action successfully

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

Set number of turns (without executing motion)	
Terminal window	Explanation
For clarity, echo and answer messages are in ***bold/inclined***	
ab[3]=6<cr>	Set the number of turns for feed/flip forward motion to 6
ab[3]=6<cr>	Echo of the command to set feed/flip forward turns to 6
	No answers are output. Only the echo of the command will be recieved. 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 ***bold/inclined***	
x=5<cr>	Send the 'flip' command to the feeder but the feeder has not been initialized before
x=5<cr>	Echo of the 'flip' command
m16<cr>	Drive 1: reports that it is not initialized
m26<cr>	Drive 2: reports that it is not initialized

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

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

4. Recommendations for implementing communication with anyfeed SX

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

feed forward (x=1), flip (x=5), flip backard (x=4), etc.

To get started, we recommend using a terminal emulator program, such as Hyperterminal that comes with Windows operating system, and try sending some of the commands to the feeder by entering them in the terminal window. This way you can test the feeder motions and also see the feeder's response.

Here is the setup for the serial communication line:

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	Drive 1 received a command which is not supported
13	Servo error on Drive 1	Drive 1 is overloaded or blocked.
16	Drive 1 not initialized	Drive 1 hasn’t been initialized yet or lost it’s state due to power cycle
17	Error state on Drive 1	Servo error on Drive 2 is detected and reported by Drive 1. No commands will be accepted until error state is reset.
22	Invalid command	Drive 2 received a command which is not supported
23	Servo error on Drive 2	Drive 2 is overloaded or blocked.
24	Backlight error	A backlight problem has been detected and it is switched off. The checking is performed at startup and during initialization.
26	Drive 2 not initialized	Drive 2 hasn’t been initialized yet or lost it’s state due to power cycle
27	Error state on Drive 2	Servo error on Drive 1 is detected and reported by Drive 2. No commands will be accepted until error state is reset.
28	Timeout - no sync-signal	Drive 2 did not receive the sync-signal from Drive1 to start action.