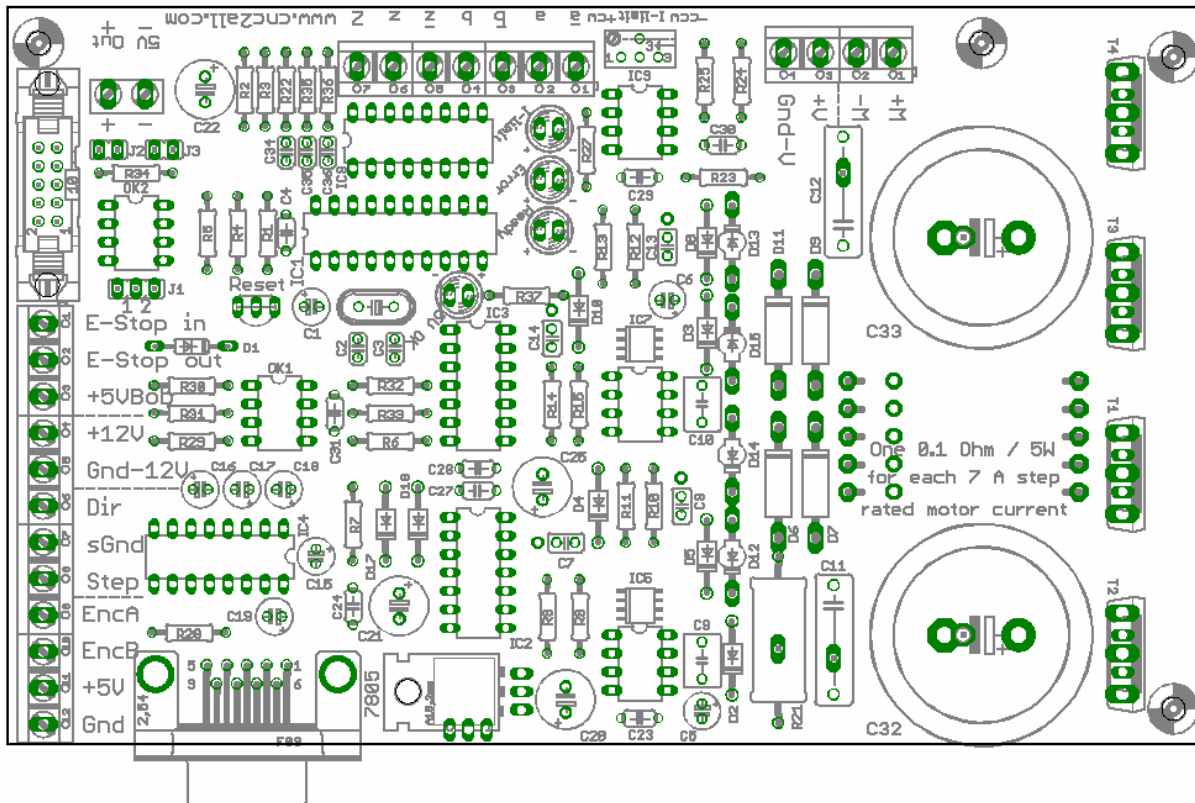


# Tuning Manual for SD-1



## Settings

Parametrizing is done solely over the serial interface. The Windows-program UHU.exe is the preferred choice. Any other terminal program can be used, if it allows configuring the PC:s RS232 port to the parameters 38400, N, 8, 1 without handshake and with echo. But usage is limited to setting and reading of parameters. The serial connection is only used for setup and control functions. It is not needed for normal operation.

**Automated machines are potentially dangerous. It is highly recommended to read the entire documentation before activating the servo controller.**

In terminal operations all inputs are done on a command line after the prompt '>'  
All input lines are submitted by the press of the enter key, displayed below as '↵'

Datam vom UHU Controller:

UHU servo Controller 3.00e-24 (c) Snr: xxxx

Help by (?)

EEPROM:

OK

>

In principle a letter has to be pressed which represents the function. A help function is available by the press of the '?' key. All commands reflect English expressions.

Example

>? ↵

(P)rop.	1000
(I)nt.	50
(D)iff.	255
(H)iOrd.	4000
Ma(X)	300
(T)orque	255
i(N)ert.	200
(E)rror	2000

(M)ult. 0  
c(Y)cle 63  
(O)verrun 0  
(L)oad [0,1,2]  
(S)ave [0,1,2]  
(G)o [Value]  
(C)ounter  
(R)eset

>

Parameters can be changed by the press of a letter , followed by the target value of the desired parameter. It will be confirmed with 'OK'.

> **P600** ↵

OK

>

If no value is supplied, the actual value is displayed with the possibility to enter a new value

> **P** ↵

(P)rop. 900: **600** ↵

OK

No input value leads to an unchanged parameter.

All input values integers, no decimals.

## Explanation of each setting

### **Proportional constant(P)**

As the name is suggesting, this value generates a control variable which is proportional to the actual position difference. The higher the value is, the more aggressively is the reaction on new position commands.

If the value is to high, it causes instabilities and low frequency oscillations with slowly declining or inclining amplitude.

Possible values are in between 50 and 5000 dependent on the encoder resolution and motor dynamics.

### **Integral constant (I)**

This value generates a control variable which increases over time with all unbalanced positions. This ensures to compensate even smallest errors (i.e. 1 encoder line) by increasing the control variable by 1/256 of the given integral constant in every recalculation interval until it is sufficient to make the motor start.

If the value is to high, it causes high frequency oscillations with inclining amplitude.

Possible values are in between 1 and 120 dependent on the encoder resolution and motor dynamics.

Attention: In Rel 3.00 the effect of the I constant was reduced by a factor 4 compared to the old Versions. This makes it easier to handle the mostly small values.

### **Differential constant (D)**

This value generates a degenerative feedback which is proportional to the actual speed to stabilize the control loop. The drawback is a tracking error, increasing with the speed. This makes the H parameter the better choice in most cases. If not needed, enter 255.

If the value is to low, it causes high frequency oscillations with inclining amplitude.

Possible values are in between 255 and 20.000 dependent on the encoder resolution and motor dynamics.

### **Higher derivative constant (H)**

This value generates a degenerative feedback which is proportional to the actual speed difference in the last recalculation interval. It stabilizes the control loop without negative influence on the tracking error. Hence it is the better choice over increasing the 'D' constant.

If the value is to low, it causes high frequency oscillations with inclining amplitude.

Possible values are in between 200 and 20.000 dependent on the encoder resolution and motor dynamics.

The influence of the H constant was significantly increased in Rel 3.00.

## Maximum speed(X)

This value limits the maximum speed of the system. It is entirely useless for step/dir operations and should be set to 5000. If 'X' is below the stepping frequency, it causes a tracking error shutdown. The speed limitation can be used in special applications, where position commands are given by the 'G' command. The tracking error value 'E' must be adjusted to a value bigger than any intended single move. Possible values are 1 to 5000.

## Maximum torque (T)

This values limits the maximum current by limiting the PWM duty cycle. It can be used if a motor with low impedance has to be attached to a controller with higher voltages. Attention: If the T - value is changed in these setups, it can destroy the output stage or the motor. Possible values are 50 to 255 (255 is the max. duty cycle 87%).

## System inertance (N)

This value affects the dynamic the control variable. It is intended only for fine tuning and is mostly not needed.

Higher values lead to 'sharper' control reactions.  
Possible values are from 50 to 1000

## Tracking error shutoff (E)

This value is the maximum allowed tracking error in encoder steps in 4 -fold interpretation before the output stage is shut off and the error -LED is activated.

The output stage stays deactivated till the next soft- or hard-reset. If you use the cnc2all.com:s own Control/Interface board just make an E-stop, this will make an reset occur.

If connected, the error output can be used to shutoff surrounding systems.  
Possible values are 10 to 5000. Lower values are better/safer.

## Step multiplier (M)

The provided integer value is added supplementary with every step pulse. I.e. if 'M' is set to '2', the motor moves exactly three 4-fold interpreted encoder lines with every step pulse.

This makes it possible to drive setups with high encoder resolutions to high speeds even if the commanding PC is limited in terms of stepping rate. As it also reduces overhead operations, it is always desirable to increase this parameter if the resolution of the encoders is not needed.

Attention: Setting 'M' to a high value leads to higher tracking errors with every step. The tracking error shutoff 'E' has to be increased accordingly.

Useful values are 0 to 20.

## Recalculation interval (Y)

The provided integer value changes the recalculation interval of the control loop.

Basic setup is 63 (interval 666us or 1.5 kHz).

The smaller the value is, the shorter is the interval and the smaller are the position differences in each interval. This can be used to tune critical setups with high encoder resolutions as the influence of the 'T' parameter is reduced.

Higher values reduces the needed calculation power and makes it easier to tune setups with lower dynamics.

As all speeds are calculated over position deltas, the change of 'Y' also influences the P-, I-, D-, H-, X and N-constants.

Possible values are 30 to max. 200 (equals from 0.3 ms to 2 ms ).

Attention: Values smaller than 30 lead to sequential processing and the loss of exact timing. This can cause unreliable results.

## “Overrun-counter” (O)

This value has no influence to the control process. It is just a counter for false transitions on the encoder lines. These can be detected because of the fact that transitions always happen sequentially and the two phases must not change both in a time interval (Gray Code).

If the do, it is always due to an error:

- speed/frequency is to high (unlikely in normal setups)
- uneven or disturbed transitions of the encoder lines
- defective encoder or defective input pin on the controller

The displayed value should always be '0'. Any higher value indicates a severe problem which will for sure lead to step losses and inaccurate operations.

The counter can be zeroed or preset.

## ***Saving configuration parameters***

All parameters can be saved in the EEPROM area of the controller. There are 3 independent sets of parameters available, numbered from 0 to 2.

The values are stored with a checksum and forward error correction to give maximum reliance and to ensure that the controller never operates with untuned or random control loop parameters.

Any hard reset loads and activates parameter set '0'

### **Reading (L)**

Allowed commands are L0, L1 and L2 for the three sets of parameters.

If the configuration cannot be reconstructed despite of forward error correction, the output stage is shut off, an error message is displayed and the error LED is activated.

### **Writing (S)**

Allowed commands are S0, S1 and S2 for the three sets of parameters.

While saving the values to EEPROM, a checksum is generated and saved in addition. There is no logical check of the values.

**Attention: Remind that any values stored to set '0' will be activated at the next startup, even if they make no sense or lead to improper operation.**

It is a good idea to store good configurations in S2 and test configurations in S1.

## ***Additional functions***

### **Direct move (G)**

This function allows to send an incremental position update to the controller. The controller moves then to the new position with a speed limited by the 'X' parameter. As this is done by setting the position error to the desired value, the tracking error value 'E' has to be always higher than any 'G' value to prevent an undesired shutoff.

This function was implemented to support a PC based auto tuning functionality in future.

Possible values are -32.000 to +32000

### **Soft-reset (R)**

The input of 'R' (without any parameter) resets the controller with the actual (unsaved) set of parameters.

The output stage is activated and the error LED is cleared.

The analyze mode is deactivated.

The command is confirmed with 'OK'.

### **Hard-reset (+)**

The input of '+' (without any parameter) restarts the controller.

Parameter set '0' is loaded and the output stage is activated. Error LED and analyze mode are cleared.

## ***Getting started***

### **Terminal program**

Interface setting is fixed to com1, 38400,n,8,echo and hence there is no need (or possibility) to change settings.

The program forwards all key presses to the controller, where they are echoed and then displayed on the screen.

If you don't see your key presses on the PC, check connection and/or proper operation of your controller board.

### **Controller**

**Attention: Prior to the initial startup of a controller board, always disconnect the motor power supply ! Doing so, you prevent your machine from hazardous movements which could be caused by faulty setup values or hardware defects.**

**If possible, start with a lower motor voltage and a current limitation by employing a regulated power supply. Rise voltage and current limit only if there is no indication for any malfunctions.**

**Never operate a machine without a emergency stop button which directly disrupts the motor power supply !**

Directly after power up, the controller executes a reset procedure, where the operating parameters are read from EEPROM and the control process is initialized.

After the initialization procedure the controller sends a startup message to the terminal,

Datam vom UHU Controller:

UHU servo Controller 3.00e-24 (c) Snr: xxxx

Help by (?)

EEPROM:

OK

>

## **Setup strategy**

Luckily the rough setup of a PID controller is not rocket science (if the bandwidth of the parameters is wide enough).

The extended algorithm which takes care of acceleration effects makes it even easier as it provides a degenerative feedback with less impact on the tracking error.

But there will still be setups where the controller fails - for example those where backlash between motor and encoder is to high and stabilizing the loop comes close to impossible.

Fine tuning on the other hand is something you need an intuition for. Playing with the effects of changing parameters will bring you this proficiency slow by slow.

If the system is not to powerful, the best way to find the best basic setup is to move the motor jerkily against its resistance and to observe the reactions on the UHU.exe terminal.

Using the 'G' command is an alternative for moving the motor by hand for more powerful systems.

### **Step 1**

Connect the controller board to your CNC PC and a PC with UHU.exe running.

### **Step 2**

Set the control parameters to 'pretty safe' starting values .(This is pre-programmed)

P = 1000

I = 50

D = 255

H = 4000

X = 5000

T = 255

N = 200

Check the System for stability. If it starts oscillating, rise the H and/or D value or reduce N.

### **Step 3**

Increase P step by step until the system starts to oscillate and compensate with H until oscillating cannot be stopped any more. D should be raised only if H does not lead to satisfying stability in very nervous setups.

Remember that D leads to higher tracking errors with high operating speeds.

There is an optimum where a high enough P -value ensures fast system response and damping caused by H does not reduce the dynamic significantly.

Ensure, that a ramp is defined in your CNC program which meets the possibilities of your motor and mechanics. Tuning can only be done in your final system setup !

Once you found a stable operating point, reduce P about 10 % to give a safety margin.

If the motor impedance does not match at all, try to decrease M. This stabilizes the loop but reduces the torque of the motor.

### **Step 4**

Increase I step by step until the system starts to oscillate. Lower I about 10%.

**Attention: Powerful systems make powerful oscillations ! Always stay in reachable distance of an emergency stop button !**

### **Step 5**

Don't forget to save your configuration with the S0 command. Save a second parameter set with S2 if needed.