

Step by step setup/tuning procedure for R9xH drives.

Introductions:

This document outlines how to setup and tune the Rutex brush motor drives R990H (100V/20A), R990H2 (200V/10A), R90H (200V/40A), brush-less drive R992H (100V/20A) and the R991H (+/-10V).

This document is targeted for new users of the Rutex products. The entire test should be run on the bench before installation of the servo drives on the machine. The purpose of this document is to familiarize the user with the R9xH servo drive prior to installation on a machine. It deals with following issues:

- Connection of the R9xH drive to the PC.
- Installation and the functionality check of test software such as RT990H.EXE for DOS or the RutexTune.EXE for Windows XP/2000.
- Connection of the different types of encoders to the R9xH servo drive as well as describing advantages and disadvantages of different types of encoders
- Connection of the servomotor, phasing the hall feedback / motor phases for brush-less drives and testing the DC output of the R991H servo drive.
- And finally, tuning of the drive.

1. Connection of the drives:

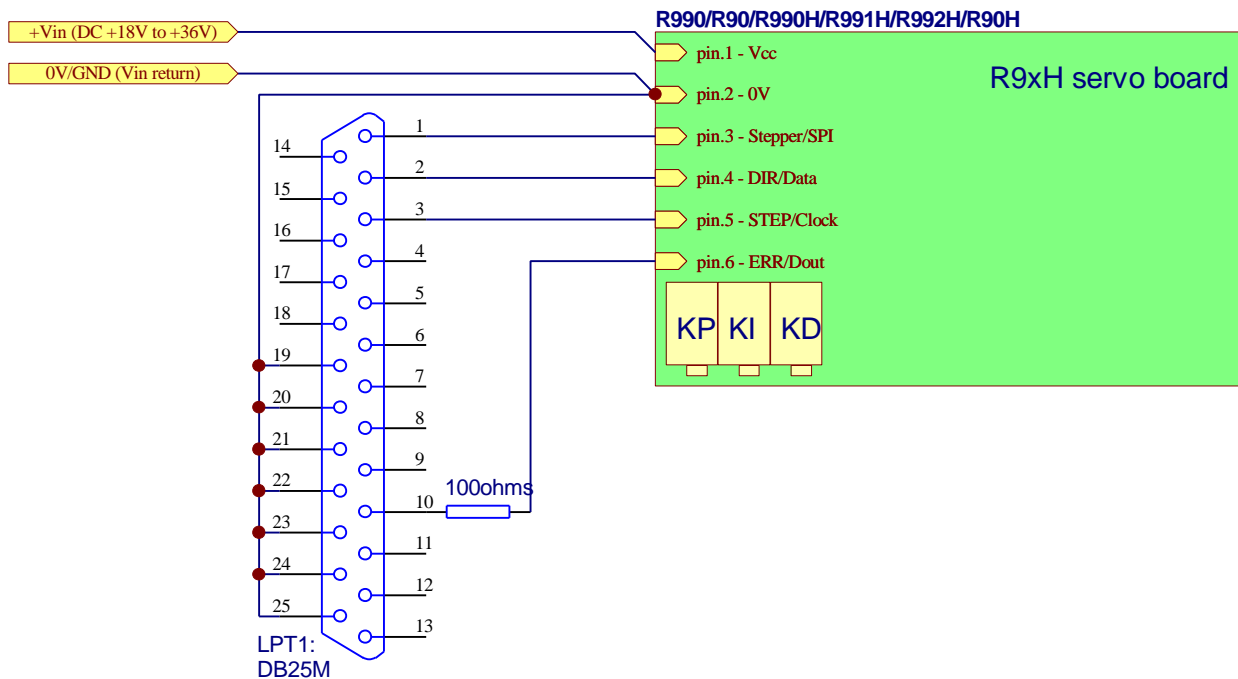
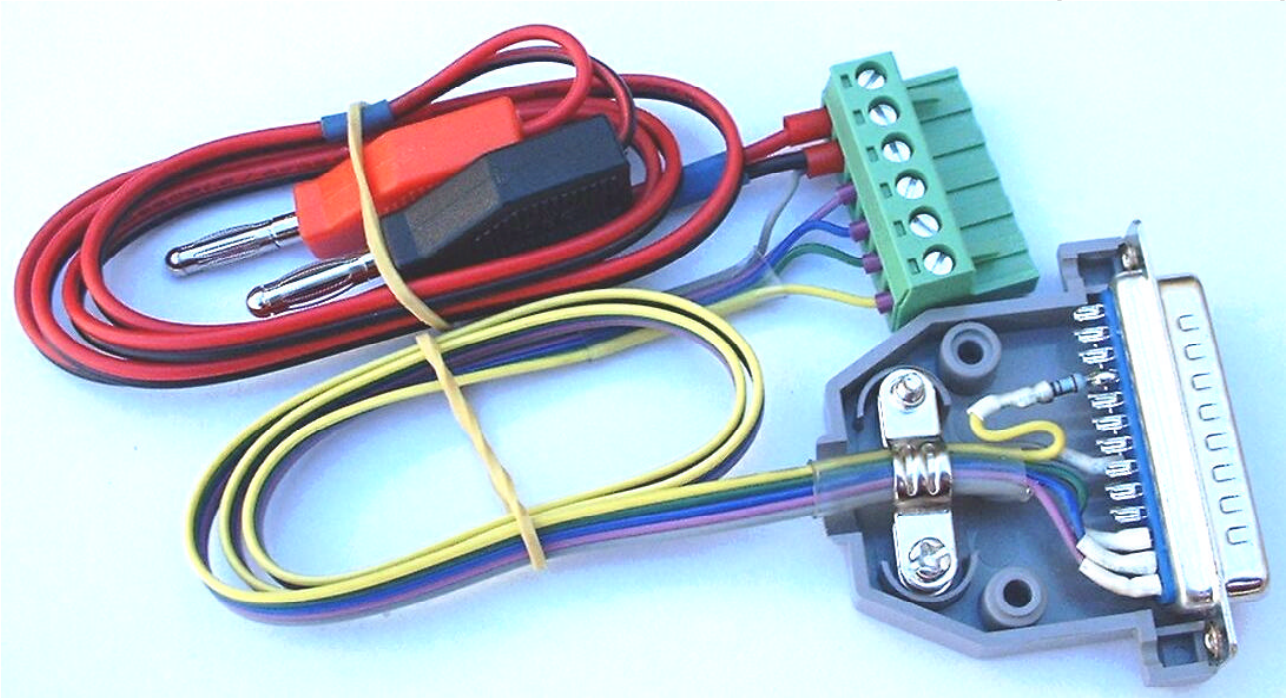
1.1. The PC based test software communicates with the R9xH drives via the printer port. It uses a SPI protocol (Serial Peripheral Interface). The Step input is used as the Clock input and the Direction input is used as Data-in. There are two more lines required for PC/R9xH drive communication: Data-out and Stepper/SPI. The Stepper/SPI input to the drive is the signal which directs the drive to switch between the standard Step&Dir mode and the SPI mode. The Data-out and Stepper/SPI lines have to be connected back to printer port pins either by a custom cable or jumpers on J7 of the R990MB motherboard – if a R990MB is used. Once the SPI tuning is done, these two lines can be disconnected from printer port pins and again used for their normal I/O functions.

1.2. Start with one drive only.

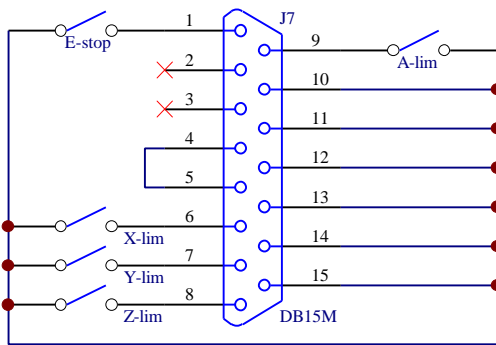
1.2.1. If you are **not using the R990MB** (motherboard), make a short *custom cable* having a DB25M (25 pin D connector available from your local computer/electronic store or www.didgikey.com p/n 1225M-ND) on one end, and a R990CON connector on the other (available from www.rutex.com or www.digikey.com p/n 277-1015-ND). You will also need a 100 ohm resistor (www.digikey.com p/n 100XBK-ND). Follow the layout shown in the picture below. A quarter watt resistor will work fine.

R9xHbdi (preliminary)

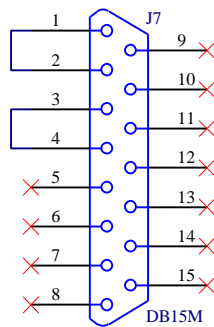
[R9xHbdi.doc 5/11/2004]



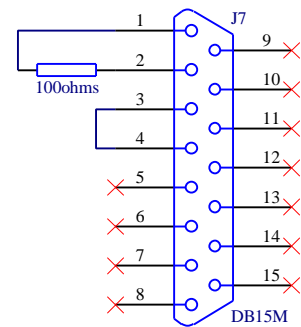
1.2.2. If you are **using a R990MB**, first of all check the revision of the PCB. If your board is revision 030618 or later then you will need only jumpers. If the revision is a lower number then you will have to use one jumper and one 100-ohms resistor. It is handy to have two plugs. One for the SPI tune only and a second one for the normal machine operation. You will need one DB15M connector for each plug that you make (15 pin D connector available from your local computer/electronic store or www.didgikey.com p/n 1215M-ND). Please follow the schematic below. You will also need a straight through cable (DB25M-DB25F) for connection between the PC and R990MB. This cable is sometimes called as printer extension cable. Use quality-shielded cable not longer than 2 meters (www.digikey.com p/n AE1012-ND).



R990MB Normal Step&Dir run - No SPI
Limit/Home switches connection



R990MB rev.030618 or later
SPI Enabled



R990MB early revision
SPI Enabled

1.3.1. If you are using the R990mb/R990io combination, connect the R990IO relay board and one R9xH drive into the R990MB as an X-drive into the R990mb mother board. Connect 18VAC into the R990IO. Alternatively, you may use 24VDC instead of AC voltage. The polarity of the DC voltage connected to the R990io does not matter, because it is passing through the bridge rectifier.

1.3.2. If you are using the custom cable as described above, then connect the supply to DC24V power supply. Any voltage between DC18V to DC35V is fine. You can use a low cost 300mA or more wall outlet power supply available from Radio Shack or Jameco. **Do not connect any AC voltage directly to the R9xH drives!**

1.4. Do not connect the encoder, motor or the DC motor power supply to the Vm terminals at this stage.

1.5. Make sure that the LED lights up on the R9xH drive after the power is switched on.

2. Test software:

2.1. Currently there are two versions of the SPI software available for the R9xH drive: The DOS version RT990H.EXE and the RutexTune.EXE for Windows XP/2000. The RutexTune can be found in some versions of the Mach-2 as well. Both tuning programs, the RT990H.EXE (DOS) and RutexTune.EXE (XP/2K) are functionally about the same. They are using the exact same pins on the printer port and Rutex drives can be tuned with either of them. Currently the DOS version has few more features such as a higher resolution graph for Step Response Tuning and manual jogs. The DOS version has been developed in-house and the Windows version by Art Fenerty (thank you Art), who is also the creator of Mach2 (www.artofcnc.ca).

2.1.1. Installation of the DOS RT990H.EXE:

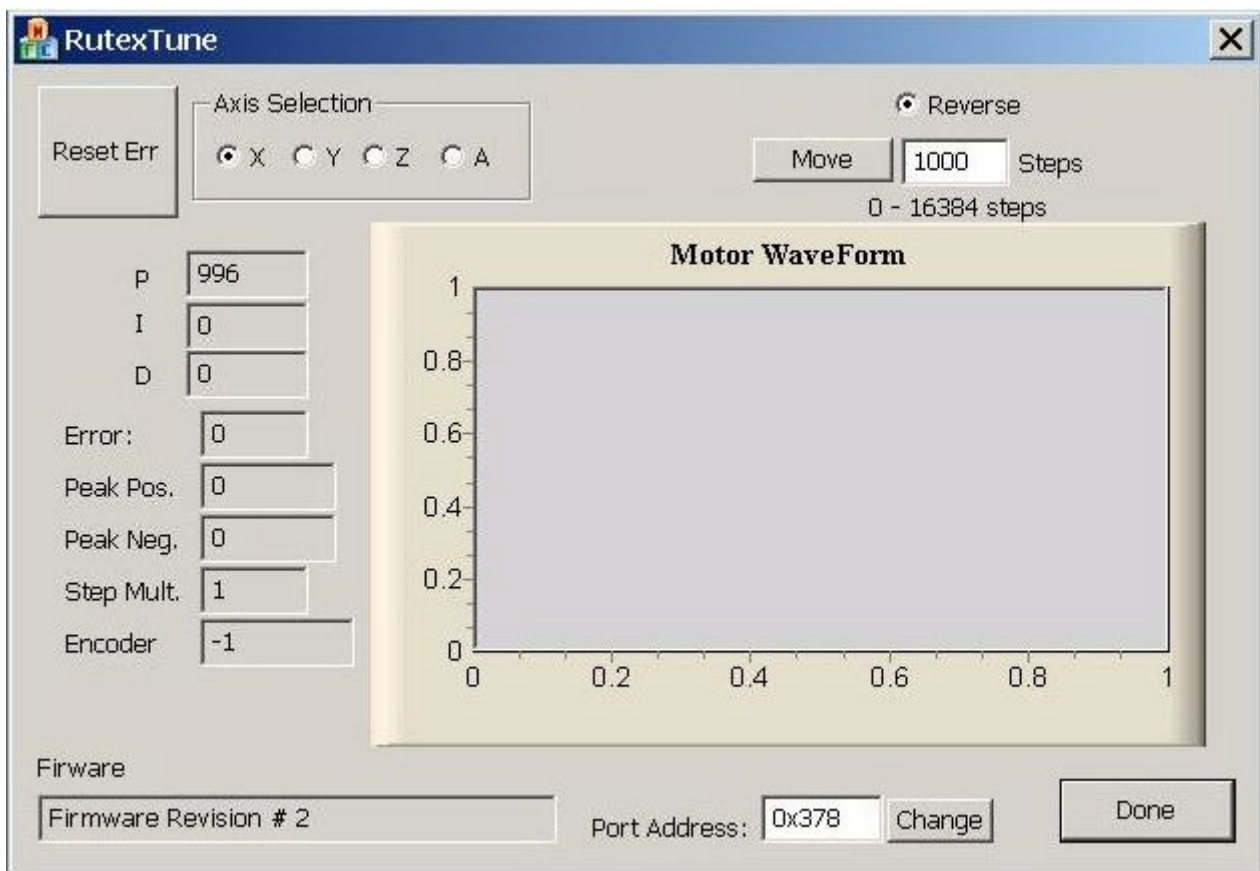
Download the RT990H.ZIP from <http://www.rutex.com/zip/RT990H.zip>. Make a folder on your hard disk, (named "Rutex" for example), and extract the files into this folder. You can run RT990H.EXE from DOS. Note that this program runs only in DOS or a DOS Window under Windows 95 or 98. It does not run in the DOS window in the XP or 2000 operating systems. If you wish to run RT990H.EXE on your XP/2K machine you will have to reboot your computer with a DOS/W95/W98 bootable floppy. Download the RT990Hboot.ZIP from <http://www.rutex.com/zip/RT990Hboot.zip>, extract it into a folder and then run the MakeBoot.BAT batch file. It will create bootable floppy disk for you. Note that if you are running directly from DOS, you might need to run the Mouse.COM or Mouse.EXE before the RT990H.EXE to install the mouse driver first.

2.1.2. Installation of the Win XP/2K RutexTune.EXE:

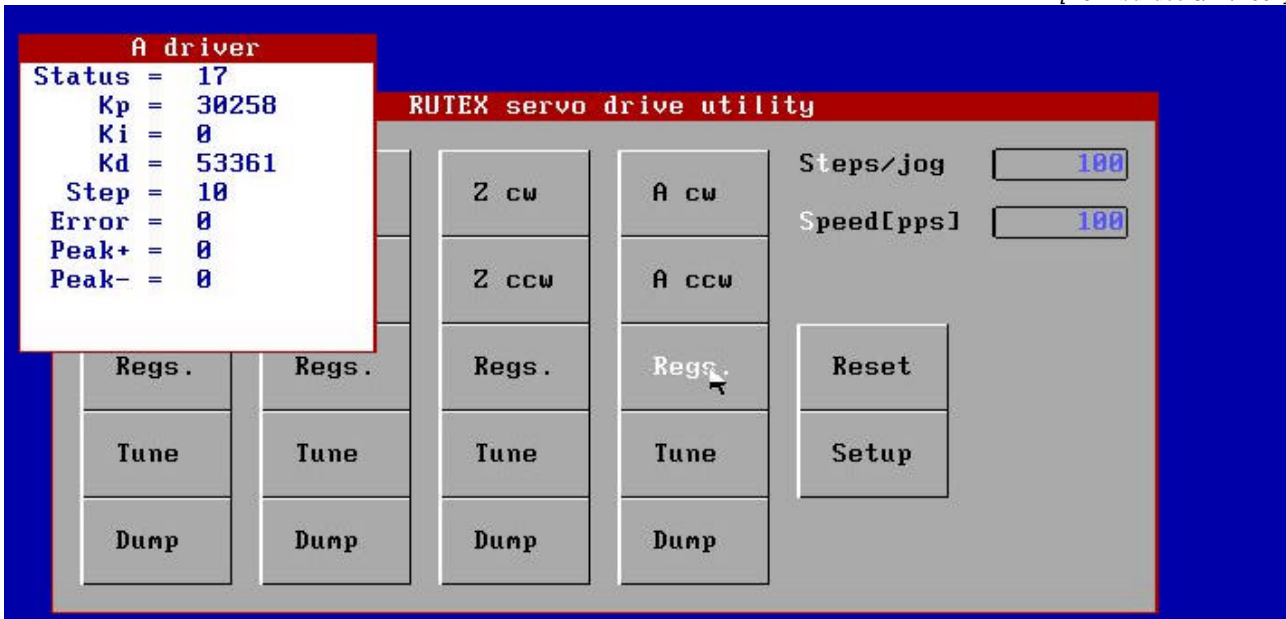
Download the RutexSetup.EXE from <http://www.rutex.com/zip/RutexSetup.exe>. This is a self-installing program. (You might need to install the latest Mach2 driver to run the program properly. Just install the Mach2demo from www.artofcnc.ca).

Note that the DOS program can be used with LPT1: only. The RutexSetup program for Windows can be used with LPT1: or LPT2:.

2.2. Switch on the power to the drive and then run the program. The program should display the value of the Kp, Ki, Kd potentiometers and the setting of the DIPswitches for step multiplier for X drive. The RutexTune displays these values directly in the main window. In the RT990H, click on the Regs button to display these drive register values. Re-adjust each of the pots (about +/-3 turns) and see that the readings of the registers change as you adjust the pots. Once you can see these registers change, you know that the R9xH is communicating with the PC and you may go on to the next step. Do not go to step #3 if you have no SPI communication. Re-check your configuration and find out why there is no communication.



RutexSetup.EXE (XP/2K) above – adjust the P, I, D, and Step Mult.



RT990H.EXE (DOS) above – adjust Kp, Ki, Kd and Step.

2.3. Note that it is quite common that during the very first run of the program the initial readings might be incorrect due to the fact that the printer port might have some internal bits set as port defaults. If this happens, exit the program, switch the drive off and then back on and then run the program again. Do not reboot the PC.

2.4. Secondly, you might get encoder readings of plus or minus one count without an encoder connected on the initial power up. This can happen because of the uneven transient of the rising voltages during the power up sequence.

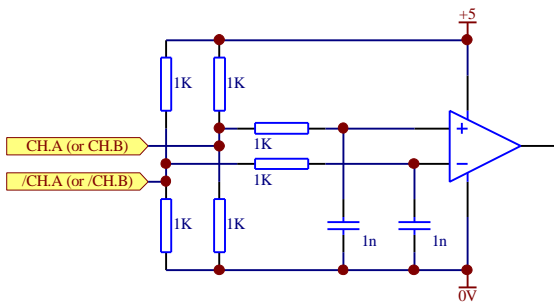
2.4. Fault finding hints:

- Go to your PC BIOS (usually press F2 or DEL during boot-up) and check that printer port is set to LPT1: (0x378 hexadecimal) and that it is set as a standard printer port (SPP) or bi-directional. Make sure that it is not set to EPP or ECP.
- Some of the older laptops are not able to produce robust 5Volt level signals. If you are using an older laptop (some brands of 486 or early Pentiums) and you are not able to establish communications, try to run the program on another PC – a standard desktop model.
- By far, the most common problem is incorrect wiring. Check it again.
- If above fails, contact Rutex.

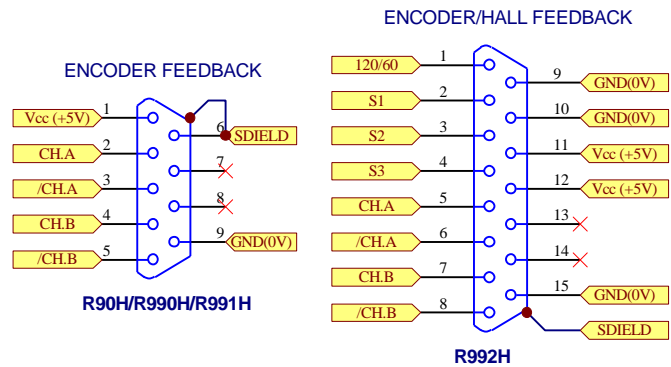
3. Connection of Encoders:

3.1. R9xH drives accept signals from standard 5V two channel incremental encoders. Incremental encoders have 2 channel square wave outputs shifted 90deg. Both channels are identical – they can be swapped. Quite often encoders have one more output which is the Index pulse. The index pulse is a one pulse per revolution marker and it is usually used for homing or synchronizing more drives for one axis. The R9xH boards do not use the index pulse. Encoders can have *single ended* or *differential* outputs.

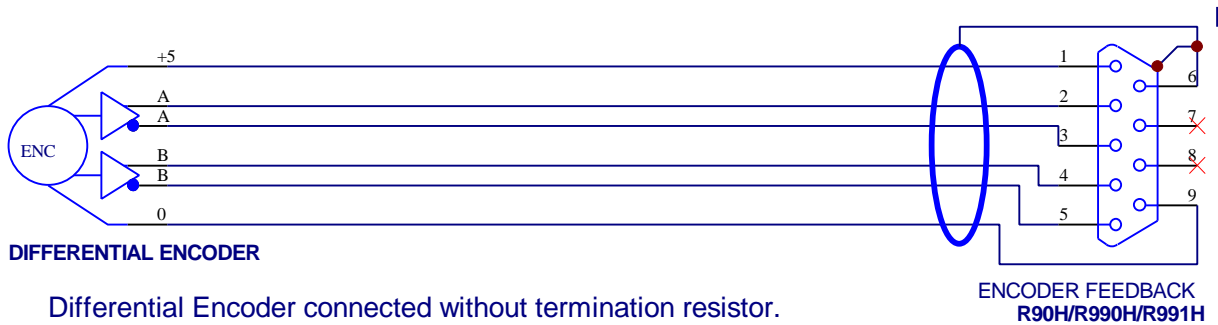
See the picture below which illustrates the internal circuitry of the encoder inputs of the R9xH drives and the encoder connector pin-layouts on the R9xH boards.



Simplified one channel encoder input on the R9xH

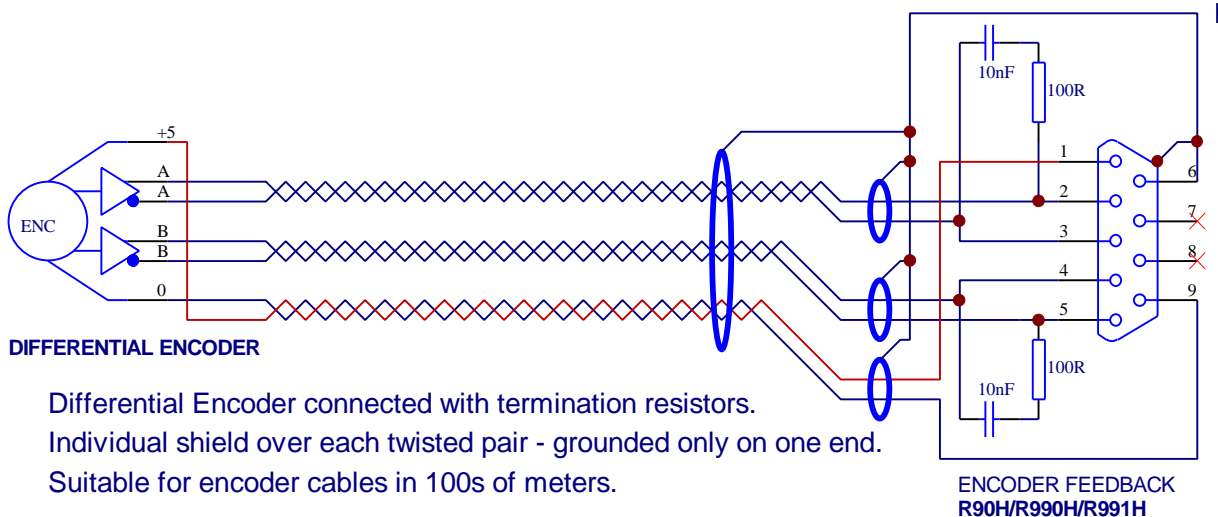


3.1.1. **Differential output** encoders have two complementary wires for each channel. If one is a logic high (5V), then its compliment is a logic low (0V). These encoders have 6 or 8 wires – depending on whether or not there is an Index pulse. There are two wires for the power supply and two wires for each channel. Differential encoders are suitable for long encoder cables and high noise environments and in general are more desirable than single ended encoders. See the picture below for connection.



DIFFERENTIAL ENCODER

- Differential Encoder connected without termination resistor.
- One overall shield over all the cores - grounded only on one end.
- Suitable for encoders cables typically less then 5 to 10 meters.



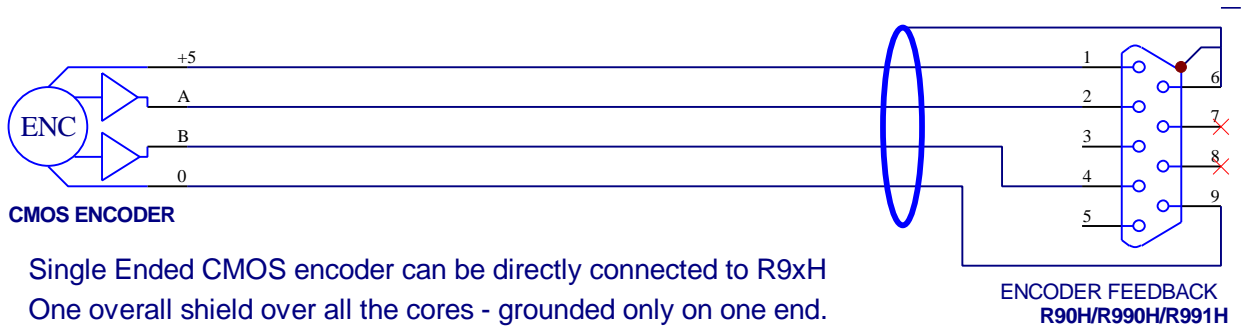
DIFFERENTIAL ENCODER

- Differential Encoder connected with termination resistors.
- Individual shield over each twisted pair - grounded only on one end.
- Suitable for encoder cables in 100s of meters.

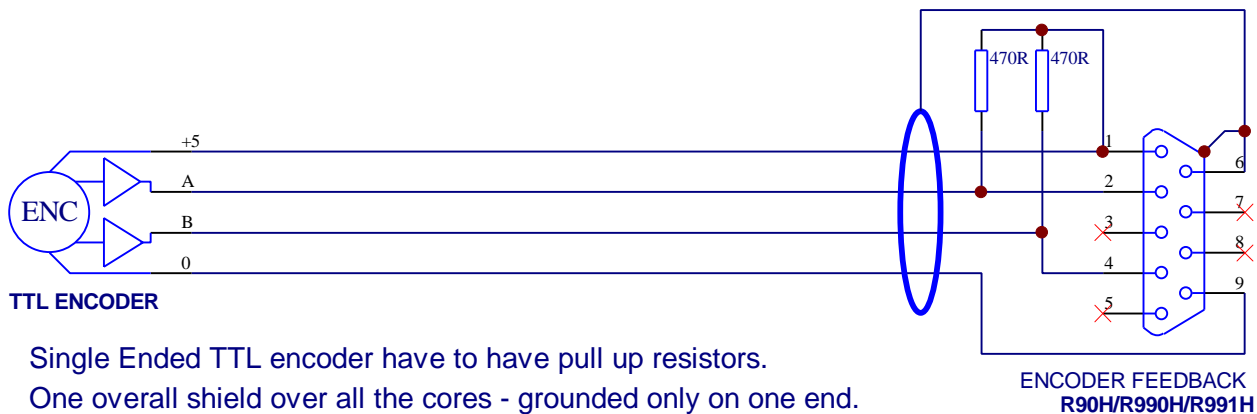
3.1.2. The **single ended output** encoders have only one wire for each output. They have 4 or 5 wires - two for supply, one for A, one for B and one for I (index where applicable). Single ended encoders can be of different types such as *CMOS*, *TTL* and *OC (open collector)*. The difference is in the amount of current they can sink (internally connect to ground) and source (internally connect to +5vdc) - encoder output current and

voltage for logic low and logic high. Note that although it is not common today, some encoders draw considerable current. Note the drive specification for the maximum current that a R9xH drive can provide for the encoder power. A single ended encoder is only suitable for short encoder cables. It is possible to pick up noise on encoder cables as short as half of a meter, although in most applications with good shielded cable which is routed away from motor wires it possible to use single ended encoders for a few meters.

3.1.2.1. The **single ended CMOS** can sink as much current as they can source, and the voltage swing of the pulse can reach close to the supply rails (+5v and 0v, for example). They usually can sink/source in orders of 10's of milliamps. Strictly speaking, the true CMOS specification is just few milliamps, but encoder manufacturers usually use high current CMOS drivers. These encoders can be directly connected to R9xH.



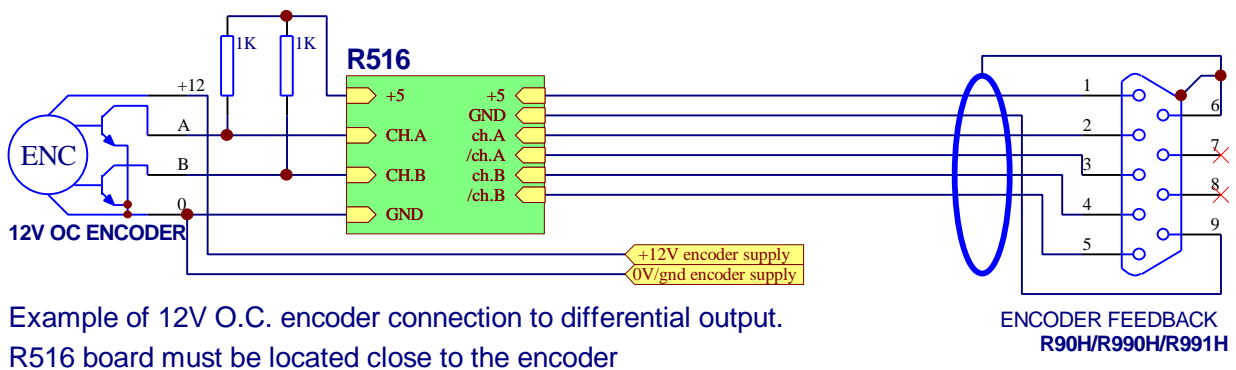
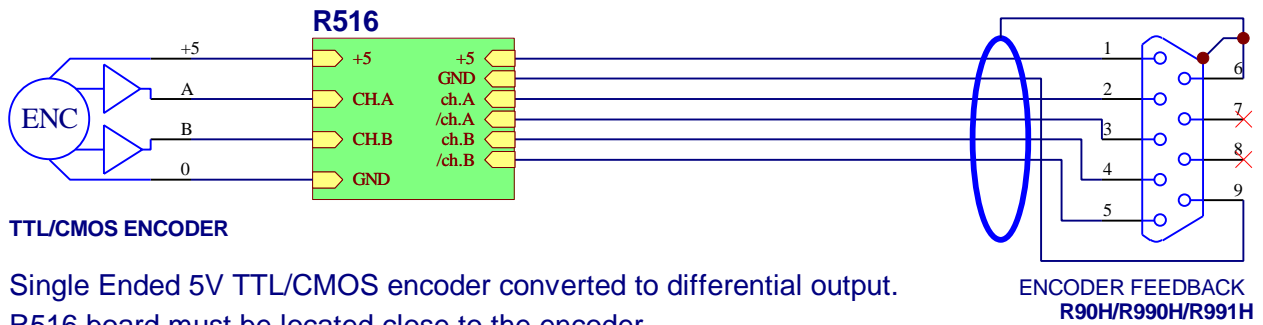
3.1.2.2. The **single ended TTL** can typically sink -24mA and source 4mA. The logic low is less than 0.5V and logic high is more than 2.4V. They need an external pull-up resistor (about 270 to 470 ohms) for each channel for direct connection to R9xH boards. The pull-up resistors are connected between the encoder outputs and +5V. They can be carefully housed inside the DB9 connector shell.



3.1.2.3. The **single ended Open Collector** encoder can usually sink plenty of current (100mA+) but they cannot source any current. They again need a pull up resistor. Check the manufacturers specification for the maximum sink current to match the pull-up resistor. Single ended O.C. encoders should be connected in the same way as the TTL encoder.

3.2. As noted above, it is better to use differential output encoders – unless the drive is physically located in close proximity to motor. This is a general rule for any encoder/servo-drive configuration. Twisted pairs pick up the same noise on the both wires and having them loaded with same impedance, the differential amplitude on the both wires is unchanged. A single ended encoder can be easily converted to differential using board such as R516 (www.rutex.com) or similar, for example, from www.usdigital.com. These are sometimes called as RS422 line drivers. Using the US-Digital board such as EA-D10 has the advantage

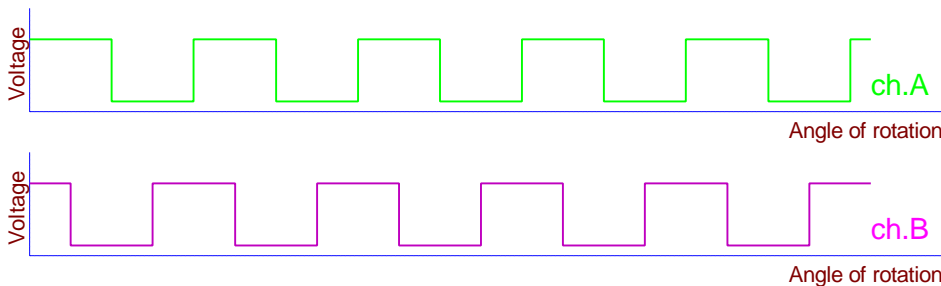
that you can order a suitable cable such as CA-4217-6FT and then just solder the DB9M on the other end. See the picture below for converting a single ended encoder to differential output using the R516 board.

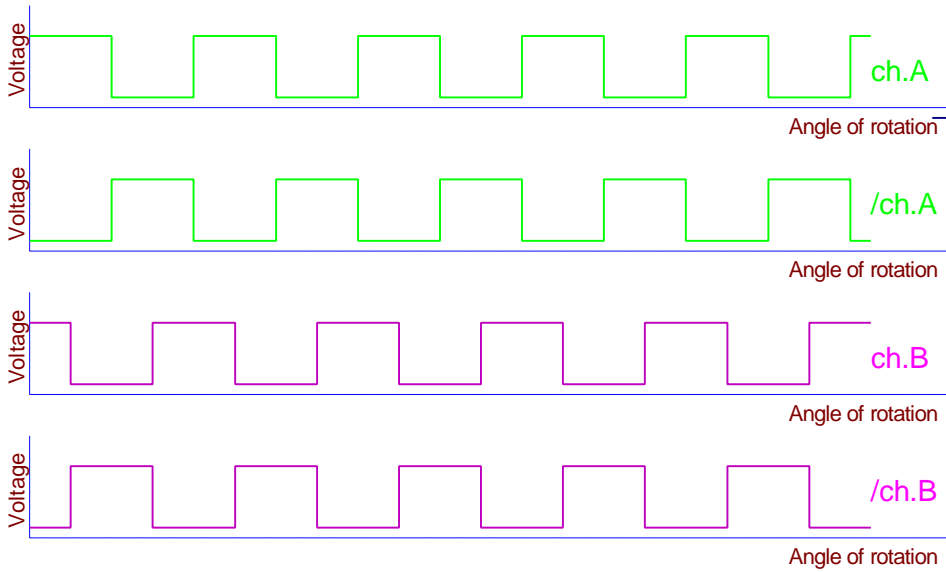


3.3. Connect your encoder per one of the configurations above with a short cable. Do not connect the motor or the supply for the motor to the Vm connector as yet. Connect Vc power to the drive. Start the test program (RT990H.EXE or RutexTune.EXE). Manually rotate the shaft of the motor and observe that the position feedback from the encoder is changing proportionally to the displacement of the shaft. If it does change, then you can go to step number 4.

3.4. Fault finding hints:

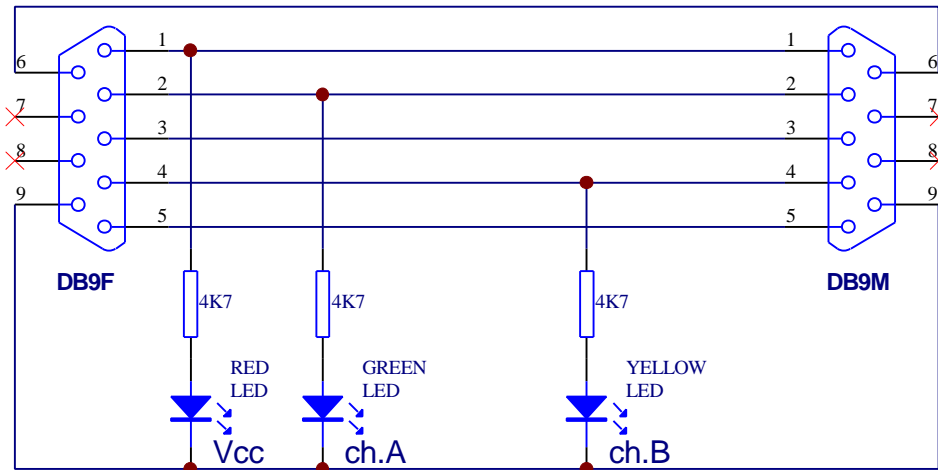
Encoder problems are the most common problems with the whole setup. You should check the signal directly on the DB9 (or DB15 for R992H) encoder connector. The signal must look like the pattern below:



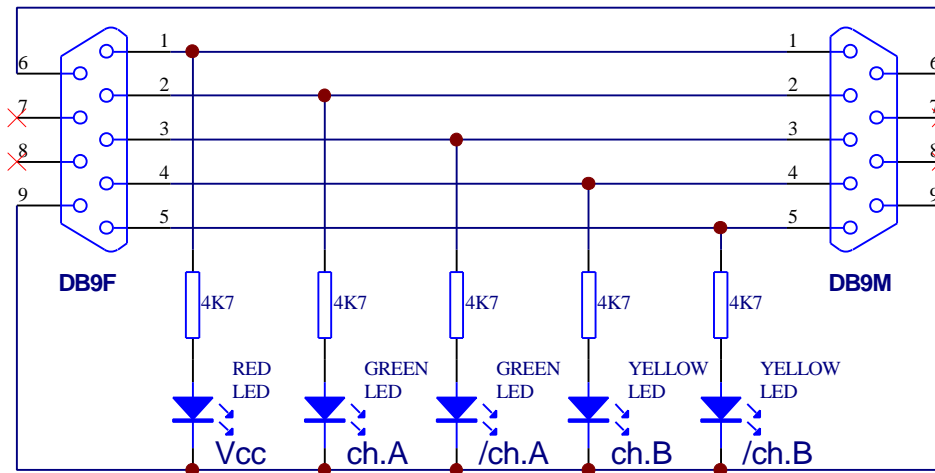


Output of differential output encoder.

The best way to check the signals is to use 4-channel oscilloscope, although the 2-channel can be used as well. Single channel oscilloscope is not much help. If you have no access to oscilloscope you can make a little test jig that can be connected inline with the encoder.



Single ended encoder test jig.



Differential encoder test jig.

Note that the LEDs will not light up very brightly. The LED current is less than 1mA. Rotate the encoder very slowly. A typical 500-line encoder will change the status of the LEDs every 0.18deg. Measure the voltage for logic low and logic high. It should be more than 4V for a high and less than 1V for a logic low when the shaft is not being rotated.

4.1. Connection of the DC PM brush motor - R990H/R990H2/R90H only:

4.1.1. Do not connect the motor to the drive unless you are sure that the R9xH board is communicating with the PC and the encoder is working properly. Initially, use a lower motor voltage and have the power-on switch close to your hand at all times. A one horse power motor can do plenty of damage – just imagine a full size horse out of control.

4.1.2. Make sure that Kp is set between 500 and 1000, Ki=0 and Kd=0 with the step multiplier dip switches set to 1. Connect the motor and supply for the Vm to R9xH board. You can use DC24V for Vm even for a 200 volt motor. You will get enough motion for testing, but you will not get the full speed or torque.

4.1.3. The motor should stay steady after the power is switched on. Carefully try to displace the shaft by hand and the motor should “fight” you back proportionally to the displacement. If it runs away for second or two, then “fights” you back with a huge dead band, then you have to swap the polarity of the motor or encoder. Do not swap both the motor and encoder, just one of them. If the motor breaks into oscillation, lower the kp value by adjusting the kp and apply power again.

4.1.3.1. For the R990H/R90H drives you can simply swap the polarity of motor wires and leave the encoder as it is OR

4.1.3.2. If you are using a single ended encoder it may be easier to swap the ch.A with ch.B

4.1.3.3. If you are using a differential output encoder, swap the ch.A with /ch.A OR ch.B with /ch.B. Do not swap both.

Now you can go on to step 5. – Tuning of the drive.

4.2. Connection of the DC PM brush-less motor to the R992H:

4.2.1.1. Determinate if the motor feedback is 60-deg or 120-deg.

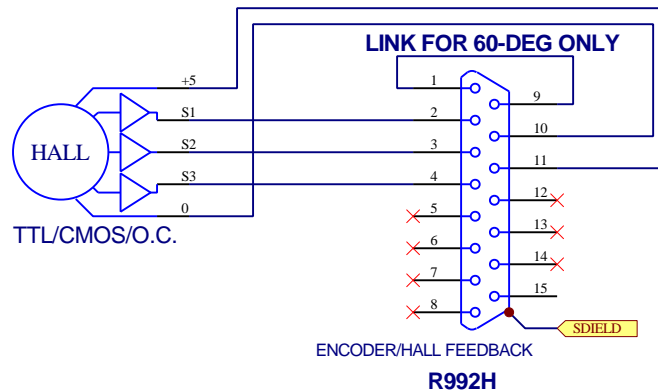
Most of the motors are 120-deg. Table below shows the 6-step communication feedback S1-S2-S3:

Rotation angle (electrical)	S1	S2	S3	S1	S2	S3
	60-deg	60-deg	60-deg	120-deg	120-deg	120-deg
0-deg	0	0	0	0	0	1
60-deg	1	0	0	1	0	1
120-deg	1	1	0	1	0	0
180-deg	1	1	1	1	1	0
240-deg	0	1	1	0	1	0
300-deg	0	0	1	0	1	1

As you can see from above the 60-deg motor NEVER goes through 0-1-0 and 1-0-1 sequence. The 120-deg motor NEVER goes through 0-0-0 and 1-1-1 sequence. You can use this rule to determinate type of your motor. Use oscilloscope or LED test jig similar as per 3.4. Note that typical 4-pole motor has two 360-deg electrical revolutions per one 360-deg mechanical revolution.

4.2.1.2. If the feedback from your motor is 60-deg then connect link between pins #1 and #9 of DB15 – Encoder/Hall feedback connector. If your motor is 120-deg, then leave pin #1 not connected. It has internal pull-up resistor.

4.2.1.3. Connect hall sensors to the DB15 as per picture below:



Do not connect the incremental encoder at this stage.

Note that for the 120-deg motors the sequence S1, S2, S3 does not have to match. They can be freely swapped at this stage. You will match the phases of motor to match to commutation feedback. Although for the 60-deg motor the S2 feedback must be connected to S2 input. Only the S1 and S3 can be freely swapped.

Note that the commutation feedback input to R992H is optimised for standard 5V single ended CMOS, TTL or Open Collector hall sensors. If the commutation feedback is different voltage (for example 12 volts) or differential output, then contact Rutex for details how to connect it to the drive.

4.2.2. Connect the Vc supply to the drive and the RT990H.EXE or RutexTune.EXE. Adjust Kp=0, Kd=0 and Ki between 5 to 10.

4.2.3. Phasing of the motor:

4.2.3.1. Set the current limit potentiometer fully CCW and use current limiting power supply (bench top laboratory supply). If you have no current limiting power supply then connect in series with +Vm a 10 ohms 50 Watts or more power resistor as well as Amperemeter. Use lower Vm – about 15-25 Volts is fine for test. Incorrectly phased motor can draw plenty of current and that can damage the motor or drive.

4.2.3.2. There are 6 possible combinations of motor phases and only one is correct. For example, if the three phase wires from the motor are Red, White, and Black then the possible combinations are:

#1	Red	White	Black
#2	Red	Black	White
#3	White	Black	Red
#4	White	Red	Black
#5	Black	Red	White
#6	Black	White	Red

The idea is to try them all to find “best” one. You can find one or two, which might seem to be correct, but when you find the right one you’ll notice the difference.

4.2.3.3. Make sure that encoder is not connected, $K_p = 0$, $K_d = 0$ and $K_i = 5$ to 10. PC does not have to be connected at this stage. Try the #1 combination and then connect at the **same time** the Vm and Vc to the drive. Within few second (between 1 to 10) the motor should start moving, slowly accelerating and smoothly increasing the torque and current. Check the torque with your hand and make sure that there is no ripple in torque or current. Incorrectly phased motor can stall in certain position or have very high torque and current ripple during motion. Try all 6 combinations to make sure that you found the right one.

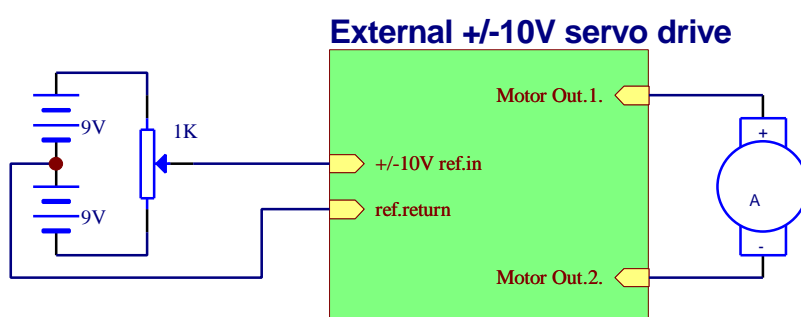
The encoder paragraph 2.4. explains why the K_i cause the motion of the motor. Alternatively you could use a small jog from RT990H.EXE (1 - 5 steps) to simulate the error for K_i accumulator.

4.2.4. Once the commutation feedback and motor are correctly phased, then you can follow the procedure for the brush motors: **4.1. Connection of the DC PM brush motor - R990H/R990H2/R90H only:** to connect the encoder.

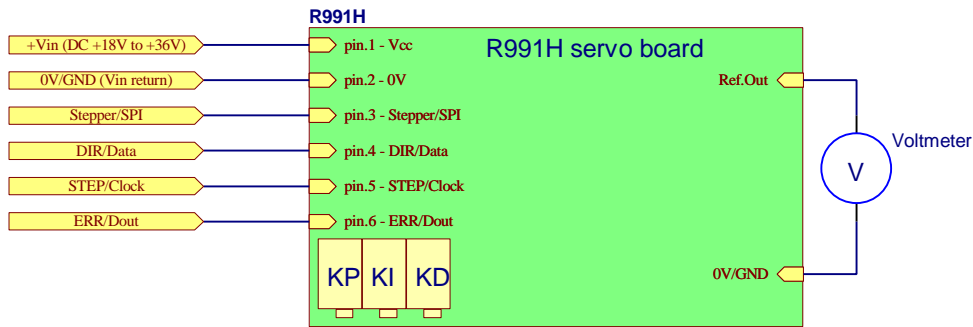
4.3. Connection of external +/-10V servo drive to R991H:

4.3.1. Do not connect the external 3rd party servo drive amplifier to the R991H unless you are sure that R991H board is communicating with the PC and the encoder is working properly.

4.3.2. Tune the external servo drive as per the manufacturer’s specification. You may want to test the drive before connection to the R991H using two 9V batteries and 1 kilo-ohms potentiometer. See the picture below.

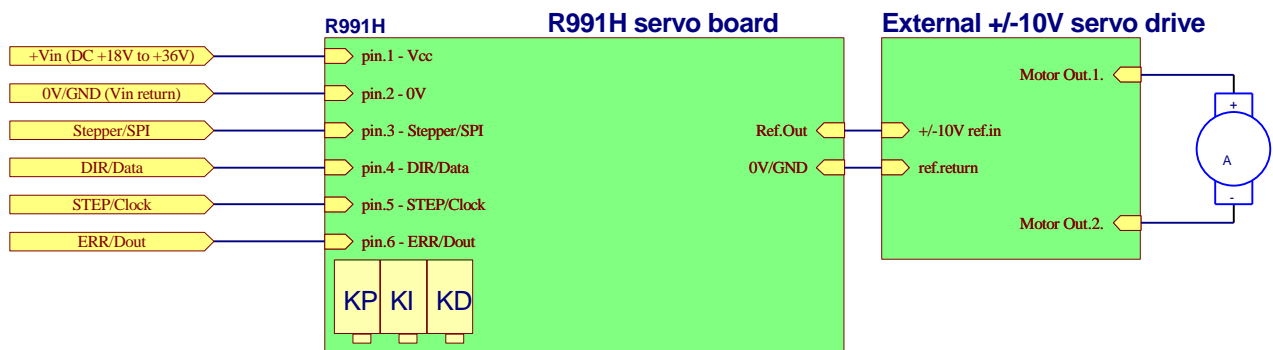


4.3.3. Make sure that K_p is set between 500 to 1000, $K_i=0$ and $K_d=0$, Step=1 (multiplier). Connect the multimeter to the output of R991H board.



4.3.4. Connect the Vc power to R991H servo board and monitor the voltage. It should be close to zero (typical offset is less than 50mV). Then rotate the shaft of the encoder by hand and the output voltage should be proportional to the angle of displacement.

4.3.5. Having tested the external servo drive and the R991H then we can connect the drives together.



4.3.6. The motor should stay steady after the power is switched on. Try to displace the shaft by hand and motor should “fight” you back proportionally to the displacement. If it runs away for second or two, then “fights “ you back with huge dead band, then you have to swap the polarity of the encoder:

4.3.6.1. If using a single ended encoder then swap the ch.A with ch.B

4.3.6.2. If you are using a differential output encoder then swap ch.A with /ch.A OR ch.B with /ch.B. Do not swap both.

5. Tuning of the servo drives:

5.1. There are 3 multi-turn potentiometers for the PID gains, one single turn pot for current limiting (not on the R991H) and one DIP switch setting on the R9xH drives. Each pot sets a value to multiply the corresponding error value in the PID algorithm. Let's consider each of them:

5.1.1. Kp – proportional gain. Proportional gain is directly proportional to the torque. It is the main force in the whole PID loop. The kp provides a multiplier for the position error between the commanded position (from the CNC control) and the counter, which tracks the feedback from the encoder.

5.1.2. Kd – differential gain. It is like oil in a shock absorber. It makes the servo settle in the desired velocity.

5.1.3. Ki – integral gain. It is like a flywheel in a toy car. You have to push it harder (more Kp) to move it, but it runs smoother then a toy car without a flywheel. The I part of the PID filter integrates the following

error and this integral is directly added to the P term and keeps the following error small during the motion or in steady condition. In fact, correctly tuned Ki can keep the following error within +/- 1 encoder count during constant velocity motion (or in steady condition) while the motor can deliver full torque. The other great advantage of Ki is that it can make from a simple trapezoid velocity profile a profile very similar to a typical "S" profile. A trapezoid velocity profile is when the motor acceleration is constant (an increment of velocity) until it reaches the desired velocity. In S profile, the acceleration at the start is rather low, then it exponentially goes up till it reaches the maximum possible acceleration. Then, just before reaching the desired velocity the acceleration gradually decreases. You can see the typical S velocity profile on the faces of the astronauts in the space rocket while taking of.

5.1.4. Step multiplier. It is a dual DIPswitch located next to the DB9 encoder connector. It can be set to 1, 4, 10 or 20. It multiplies the steps from the step input. For example, if the step multiplier is set to 10 then each one step on the Step&Dir input represents the motion of 10 steps from the encoder (2.5 optical lines), although the 10x step multiplier decreases the resolution of the system by 10. You should match the step multiplier with the maximum required speed of your system. For example, running software with 45kHz sampling frequency can generate a maximum pulse train of 45,000 samples per second, or 2,700,000 steps per minute. Using a 500 optical line (2,000 quadrature pulses) encoder and a step multiplier set to 1, the maximum available speed is 1350 RPM (2700000/2000). If the step multiplier is set to 10 the maximum speed is 13,500RPM.

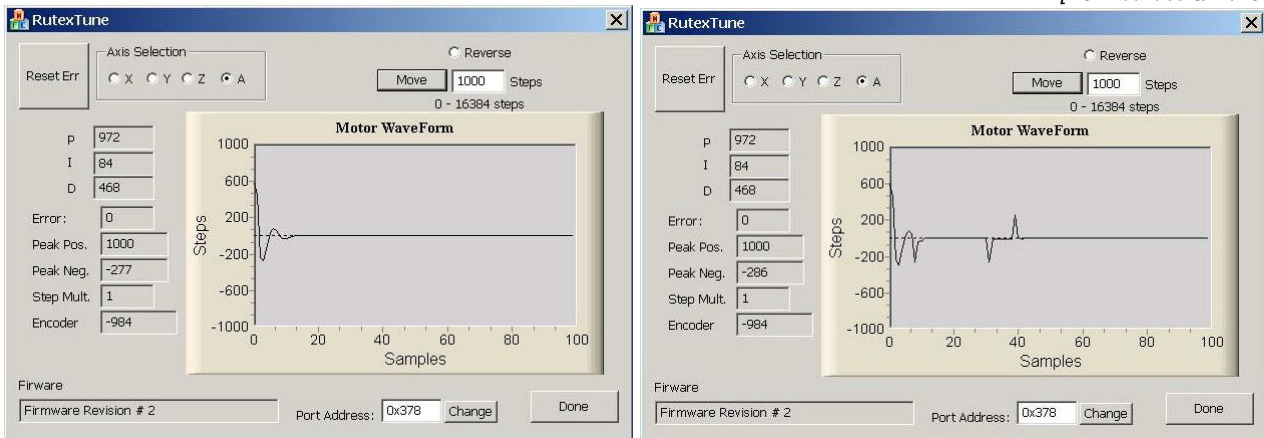
5.1.5. The (average) Current Limiting pot. It is a square potentiometer located close to the center of the R9xH boards (excluding R991H). Fully CCW sets the current to minimum and fully CW disables the average current limiter – although, the peak, cycle-by-cycle, current limiter is not disabled. The minimum current is about 1 Amp for the R990H/R992H drives, 2 Amps for the R90H and about 0.5 Amp for the R990H2. The adjustment is non linear – more like an exponential response (the middle of the pot adjustment range sets current to about double the minimum current setting).

5.2. Step Response Tuning:

5.2.1. Step Response tuning is a commonly used method of tuning a servo system. In tuning a Rutex drive, the tuning software induces a very large following error through the spi communication with the servo drive and then the user monitors the settling time of the correction on the monitoring screen. The aim is to adjust the drive for shortest possible settling time and acceptable error, but keep in mind that some overshoot is normal in this tuning method. The size of the Step should be large enough to let the motor reach its full speed. Usually the step has to be between ¼ to 1 turns of the motor shaft. This tuning method does not employ a trapezoid profile with the programmed acceleration/deceleration, and this is why the motor might overshoot. If the step (or induced following error) were very low or provided correctly programmed acceleration/deceleration, then the motor should not overshoot. A properly tuned servo, matched with a control that employs acceleration and deceleration will not overshoot, but the tuning software allows the user to tune the servo system under somewhat extreme conditions.

The Step Response tuning can be started by clicking on the 'Move' button in the RutexTune.EXE or the 'Tune' button in the RT990H.EXE.

Note two graphs from below. Both graphs show the results of the same motor/drive tuning. The second graph is displaying 3 incorrect samples due to the fact that the reading has changed during sampling. Each 16-bit sample is read via SPI in two sequential un-buffered 8-bit bytes. For example, if the value changes from -1 to 0 (0xFFFF to 0x0000) then the first byte is read as 0xFF and second one as 0x00 which makes -255 (0xFF00). If this happens, just disregard this graph – there is nothing wrong with the drive.



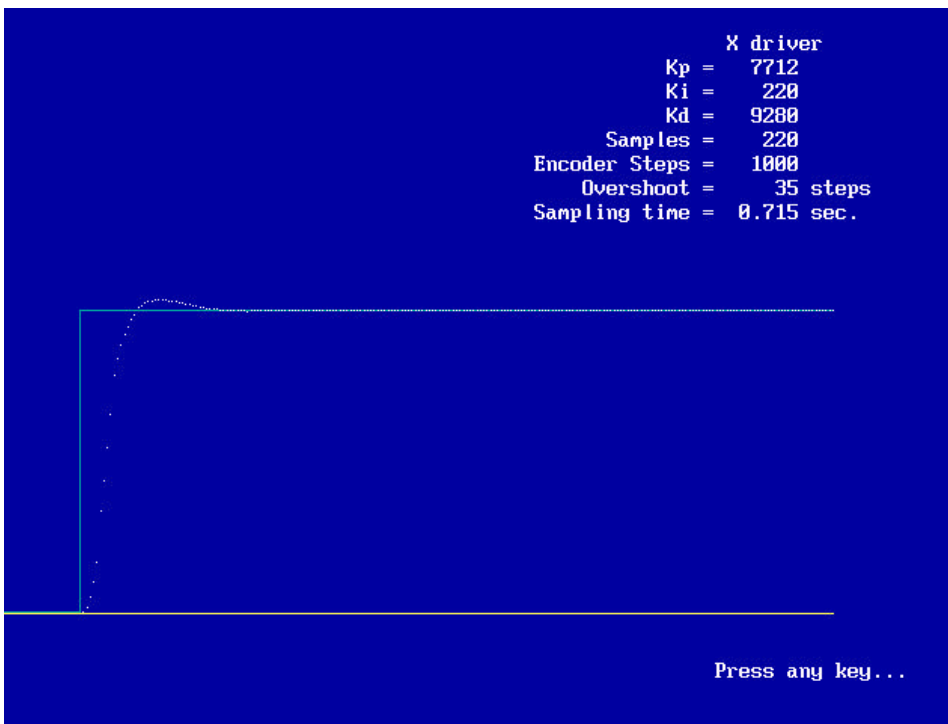
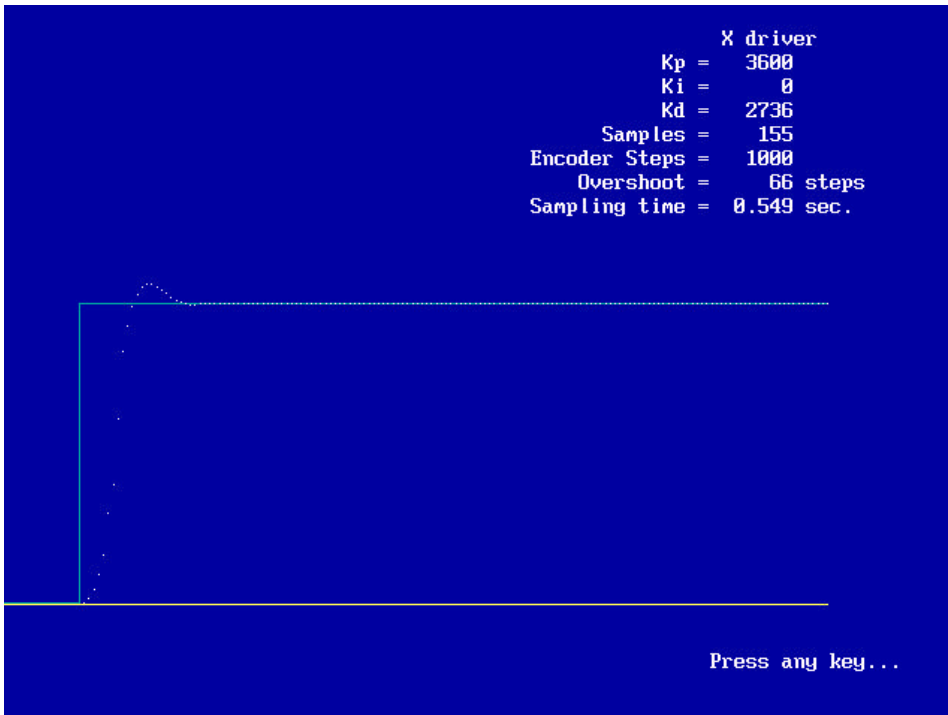
5.2.2. The tuning is very much motor/machine/encoder dependant. There is no universal procedure. Before you start to tune drive you should mechanically disconnect motor from the machine and you should have the E-stop switch very close to your hand. An incorrectly tuned drive can violently oscillate which can do mechanical damage to the machine/motor or overheat the drive. It is also a good idea to start the tuning at a lower voltage (if you can drop down the voltage) to familiarize yourself with tuning. There is much less mechanical/electrical energy.

5.2.3. You should know if your system needs the Ki gain or not. The difference is that without Ki the system is bit faster - sort of 'snappier' when traveling, but without ki, there is an inherent error of few encoder counts - somewhere around 5-50 encoder counts. With Ki the system might be slightly less dynamic but it will correct the steady error down to +/-1 encoder count. Most applications are tuned incorporating some Ki.

A typical setup for a DC PM motor running at DC30V with a 500-line encoder could be something in the neighborhood of $K_p=1000$, $K_i=0$, $K_d=2000$. If you apply K_i , then K_p and K_d usually have to be dramatically increased. Also, the K_i has to be adjusted over a certain value to work correctly. Usually very low K_i does not do a good job at all. So the setting could be somewhere around $K_p=5000$, $K_i=150$, $K_d=3000$. If you double the Voltage or double the encoder resolution, then the K_x values should be about half. The above values are very rough estimates and you might end up with completely different values.

5.2.4. The adjustment of the drive is to find the best combination of setting of the K_p , K_d and K_i . First, find the right combination of K_p and K_d – keep the $K_i=0$. You should change the value of potentiometers, and run the step response and view the graph. Repeat this process until you get a good response. After that add K_i and adjust all three pots, one at the time and run the step response. When you are familiar with the tuning procedure, mount and connect your motor shaft to the machine and re-tune to the dynamics of the whole machine.

5.2.4. Note the two graphs below of a correctly tuned servo drive with the same motor. The first one is not using K_i and second one is with the K_i . Note that the first graph has bit faster setting time and lower K_p and K_d . In second graph the motor is bit slower.



(End of the document)