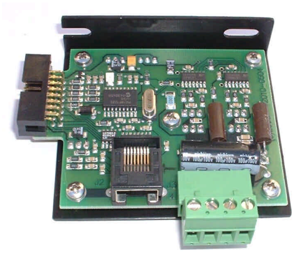
Step&Dir and SPI
Servo Drive Boards:

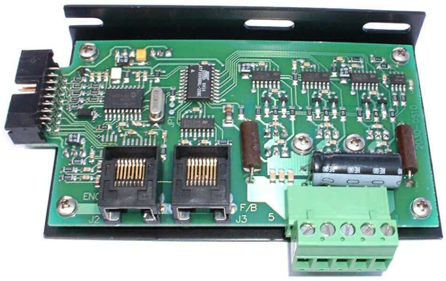
R2010 - 100V/20A brush motors
R2020 - 200V/40A brush motors
R2030 - 100V/20A brushless motors
R2040 - +/-10V analogue output.

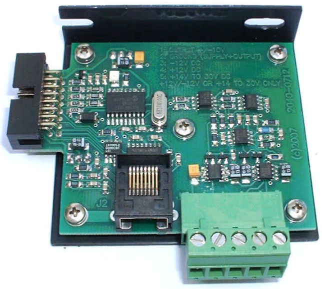
Features:

- Standard stepper input interface - STEP & DIR.
- Advanced SPI (Serial Peripheral Interface) for host control and motion command with 16 deep FIFO for motion command and up to 128 points interpolation – W2K/XP DLL available.
- 5 volts incremental differential encoder input – shielded RJ45.
- Output for driving brushed or brushless servo motor (AC/DC).
- Output bridge rated 100V 34A (R2010/R2030) and 200V 56A (R2020) continuous.
- Optoisolated +/-10V output (R2040) to drive external, 3rd party, analogue servo drives (AC, DC, brushless, etc.).
- Peak and average current limiter with 2 second fall-back.
- Enhanced PID filtering.
- Servo loop time adjustable from 125 us to 13 ms with optional external synchronization.
- True 32-bit encoder counter (DRO).
- Raw and decoded (clock up/down) TTL encoder “taps” output.
- Overvoltage and undevoltage detection.
- Following error trip (adjustable from +/-2 to +/-30000)
- Stall rotor detection trip.
- Encoder fault detection trip – no “run-away” motor.
- Limit switch input (hold or disable motor).
- Support for encoder index counter.
- Replacement for R9xH series.
- Windows (2K/XP) based tunning software.
- Low cost


R2010 100V/20A brush motors

R2020 200V/40A Brush motor


R2030 100V/20A Brushless motor


R2040 +/-10V analogue output.

Description:

The R20x0 Servo Drive Board is a motion controller for brush or brushless motors or external 3rd analogue servo drives. It accepts either STEP and DIRECTION commands from host or motion command via SPI. The SPI motion command is simply a 16-bit number of steps per each particular servo loop cycle (or quite often called as velocity word). Rather than clocking each step individually, the host CNC tells the drive the number of step via SPI. For example, with a 1ms loop-time, the maximum speed of SPI motion is +/-32767 steps/ms or 32,767,000 steps per second. There is a build-in fractional interpolator in R20x0 servo drive which allows the servo loop time to be faster than the effective loop, for example optimal loop time for motor might be 500 us and optimal loop time for host CNC calculation might be 4 ms. If so, the interpolator should be set to 8. Also, there is a 16 x 16-bit FIFO for the velocity word buffering in the drive. These also ease the requirements for the host CNC. The host has to service (talk to) the servo drive only every few tens or even hundreds of milliseconds to keep the motion smooth and uninterrupted.

SPI communication is little known to average PC users, compared to RS232, USB etc, although it is well established in the PC. The standard PS2 keyboard is using variation of SPI and video cards read the information from the monitor using a modified SPI during boot-up. The SPI can be easily implemented in the PC using standard printer port (LPT) similarly as it is used for step & dir generation. The printer port was not build to be use for SPI nor step&dir, but both of them can be implemented similarly. The SPI uses 3 communication lines: clock, data-out, and data-in. Clock and Data-out are the same as Step and Dir signals. In the R20x0 drive they are multiplexed in between. The data-out from drive is connected to data-in of the PC and it is an open collector signal so more then one drive can be connected together. As a matter of fact, 7 drives (axis) can be controlled via single printer port using only 9 signal lines: 7 independent data-out lines, one common clock and one common data-in line. The host can talk to all 7 drives at the same time – typically sending the motion command (velocity word) and reading individually from each drive, for example each encoder reading (DRO) once the FIFO is full.

There are several interface products available for R20x0 servo drives such as motherboards and interfaces for encoder connection.

Electrical specifications:

Model:		R2010	R2020	R2030	R2040	
Supply voltage for motor:	V_M	12-100	12-200	12-100	N/a	V
Supply voltage for control circuit:	V_C	18-36	18-36	18-36	18-36	V
Isolated supply	Vi	N/a	N/a	N/a	+12/-12 or +15to35	V
Output voltage	$Vout$	~Vc	~Vc	~Vc	+/-10	V
Supply Current (no encoder connected, Vc = 24 V):	I_C	40	40	40	40	mA
Maximum peak motor current for t < 1 ms	I_{PEAK}	110	220	110	N/a	A
Maximum continuous peak motor current	I_{MAX}	20	40	20	N/a	A
Maximum continuous average motor current	I_{AVG}	10	20	10	N/a	A
Default PWM frequency	f_{PWM}	19.5	19.5	19.5	19.5	kHz
Adjustable peak current limit [typical]	I_{LIM}	0.5-20	1-40	0.5-20	N/a	A
Operating temperature	T	0-70	0-70	0-70	0-70	^{O}C

Pin descriptions:

J1 – interface (IDC20) – R2010, R2020, R2030, R2040

Pin No	Name	Description
1	ENC-B	TTL Encoder ch.B output (for external DRO)
2	ENC-A	TTL Encoder ch.A output (for external DRO)

3	0V	Signal/supply ground
4	ENC-I	TTL Encoder Index output
5	Input.1	Limit switch input (switched to 0V)
6	Input.2	Index input – linked to pin.4 if index is used
7	ENC_DN	TTL Decoded encoder down clock output.
8	ENC_UP	TTL Decoded encoder up clock output.
9	+5V out	+5V supply output
10	0V	Signal/supply ground
11	DIR/Din	Dir (in Step&Dir mode) or Data-in (in SPI mode) input
12	STEP/Clk	Step (in Step&Dir mode) or Clock-in (in SPI mode) input
13	StepMode/SPImode	Logic high = Step&Dir mode (internal pull-up) Logic low = SPI mode
14	Err/Dout	ErrorOut (in Step&Dir mode) or Data-out (in SPI mode) output
15	0V	Vc return (ground)
16	+Vc	Controll supply DC 18 to 36 Volts

J2 – RJ45 – Encoder connection – R2010, R2020, R2030, R2040

Pin#	Signal Name	Description
1	Ch – A	Encoder input channel A – non inverted
2	Ch – $\underline{A}$	Encored input channel A – inverted
3	Ch – B	Encoder input channel B – non inverted
4	+ 5V	Positive supply for encoder
5	0V	Negative (ground) supply for encoder
6	Ch – $\underline{B}$	Encoder input channel B – inverted
7	Ch – $\underline{I}$	Encoder input Index – inverted
8	Ch – I	Encoder input Index – non inverted

Note that pin 1 of J2 is the pin closest to J1 connector (IDC20).

J3 – RJ45 – Commutation feedback – ‘halls connection’ – R2040 only

Pin#	Signal name	Description
1	S2	Hall feedback sensor #2 – non inverted

2	S2	Hall feedback sensor #2 – inveteted
3	S1	Hall feedback sensor #1 – non inverted
4	+ 5V	Positive supply – connected to pin 4 of encoder feedback
5	0V	Negative (ground) supply – connected to 5 of encoder feedback
6	S1	Hall feedback sensor #1 – inverted
7	S3	Hall feedback sensor #3 – non inverted
8	S3	Hall feedback sensor #3 – inverted

Note that pin 1 of J2 is the pin closest to J1 connector (IDC20).

J3 – 4 pin plug – Motor connection – R2010only

Pin#	Signal Name	Description
1	+Vm	DC Motor supply positive
2	0V	DC Motor supply return
3	Mot.1	Motor terminal 1
4	Mot.2	Motor terminal 2

Note that pin 4 of J3 is the pin closest to J2 connector (RJ45).

J4 – 5 pin plug – Motor connection – R2030only

Pin#	Signal Name	Description
1	+Vm	DC Motor supply positive
2	0V	DC Motor supply return
3	Mot.1	Motor terminal 1. Phase – S
4	Mot.2	Motor terminal 2. Phase – T
5	Mot.3	Motor terminal 2 . Phase – R

Note that pin 5 of J4 is the pin closest to J2 connector (RJ45).

J3 – 5 pin plug – Output connection – R2040only

Pin#	Signal Name	Description
1	+/-10V	Reference output +/-10V
2	0V	Reference ground, isolated supply ground

3	-12V	Isolated -12V dual suppy input
4	+12V	Isolated +12V dual supply input
5	+15V to +30V	Alternative isolated +15V to 30V single supply input.

Use only one type of supply. Eighter the complementary +12V and –12V regulated supply (connected to pins: 2, 3 and 4) or single ended +15V to 30V (unregulated) supply (connected to pins: 2 and 5). Note that pin 5 of J3 is the pin closest to J2 connector (RJ45).

Setup procedure:

Please follow the instruction in the **R2000setup.PDF** and **R2000faq.PDF** document available for download at www.rutex.com.

Internal SPI registers:

Name	Adr	Size	Type	Description
Dummy8	10h	8-bit	R/W	Dummy FFh register
StatusReg	11h	8-bit	R/w	Status Register – holds the status of FIFO, input and error
Revision	12h	8-bit	R/W	Firmware revision
StepSize	13h	8-bit	E/W/E	Size of the step multiplier
Im	14h	8-bit	R	Motor current - real time reading
TimingReg	15h	8-bit	R/W/E	Timing Register - setting of loop time and SPI interpolation
ConfigReg	16h	8-bit	R/W/E	Configuration Register -
ErrReg	17h	8-bit	R/W	Error Register – Status of Errors
ImLim	18h	8-bit	R/W/E	Motor current – limiter setting
KdIndex	19h	8-bit	R/W/E	Kd Index - part of PID filter
ImFall	1Ah	8-bit	R/W/E	Motor current – fallback setting
ImPeak	1Bh	8-bit	R/W	Motor current – peak detector
CommandReg	1Ch	8-bit	R/W/E	Command Register – command for limit switches, motor etc.
Dummy16	20h	16-bit	R/W	Dummy FFFFh register
Kp	22h	16-bit	R/W/E	Kp gain
Ki	24h	16-bit	R/W/E	Ki gain
Kd	26h	16-bit	R/W/E	Kd gain

Cnt16	28h	16-bit	R/W	Following Error counter – real time
Cnt16up	2Ah	16-bit	R/W	Following Error counter – peak “up” detector
Cnt16dn	2Bh	16-bit	R/W	Following Error counter – peak “down” detector
Cnt16Off	2Eh	16-bit	R/W/E	Following Error counter – trip detection
Acc16	30h	16-bit	W	Velocity word FIFO input
Vm	32h	16-bit	R	Motor supply – real time reading
VmPeak	34h	16-bit	R/W	Motor supply – peak detector
DriveType	38h	16-bit	R/W	Drive type
Dummy32	40h	32-bit	R/W	Dummy FFFFFFFFh register
DROu	44h	32-bit	R/W	Unbuffered DRO output
DRO	48h	32-bit	R/W	Buffered DRO output
SaveE2	54h	32-bit	W	= 12345678h to save internal /E register into permanent E2PROM memory
DROIndex	58h	32-bit	R/W	Index counter. Whenever index pulse is detected, then the value of DRO counter is latched into DROindex

For programming details consult the **STEPEX2** protocol documentation available from Rutex.