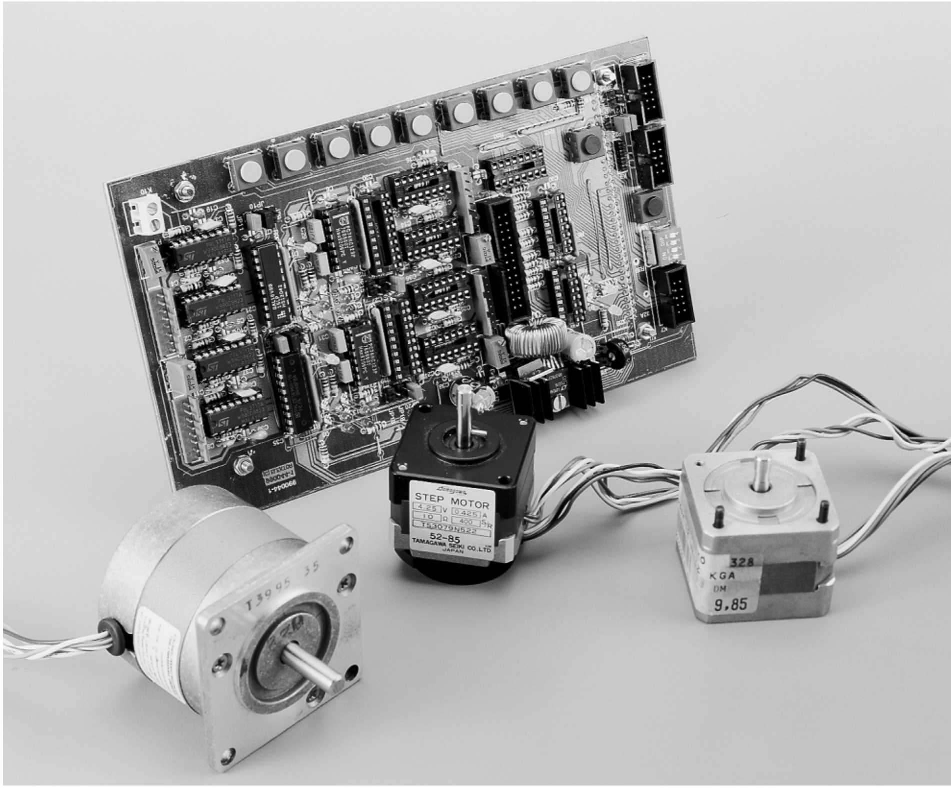
stepper motor control

part 2: SMC, a 80C166-powered driver for 4 motors

Over the past few years, we've witnessed a significant increase in the use of stepper motors. This result is largely due to technological advances and refinements achieved by the semiconductor industry, which currently offers a galaxy of integrated circuits, drivers and power stages specifically developed for the control of stepper motors. The number of ready-made controls complete with power drivers has also seen a remarkable increase.



Main features

- *4 power drivers, each having 2 bipolar output pairs for 0.5A or 1.2A phase winding current (PBL3717/PBL3717/2N)*
- *Current adjusted digitally in 3 steps, or analogue via presets*
- *Current reduction at no motor action*
- *Cut-off time 35 µs*
- *Full and half-step operation, other modes available by reprogramming system GALs.*
- *LED indicators for clock and direction*
- *4 optocoupler inputs for zeropoint sensors or contacts*
- *8 optocoupler inputs for position sensors or contacts*
- *2 serial interfaces*
- *10 pushbuttons for manual control*
- *1 reset pushbutton*
- *Power supply 12-40VDC, approx. 2.0A or 4.8A*

Design by K. H. Domnick

The key advantage of using stepper motors is the ability to control motor speed and spindle position without the need for a closed control circuit. Drivers and power stages for stepper motors determine the amount of current and the direction of the current sent through the stepper motor windings. For spindle movement, the direction signal has to be complemented with a variable-frequency clock signal. For accurate positioning of the motor spindle, all pulses have to be counted and tallied. Clearly these functions challenge us to use a powerful microcontroller system like the 80C166 board described in the March & April 1999 issues of *Elektor Electronics*.

The **S**tepper **M**otor **C**ontrol (SMC) described in this article is the perfect mechanical end electrical companion to the 80C166 Evaluation System. The 80C166 employs tailor-made software to generate all the necessary signals for up to four power drivers on the motor control board, and also handles all information supplied by the available optocoupler inputs, the keyboard and the serial interface.

The construction, dimensions and mechanical assembly details of the 80C166 board and the SMC board are such that they can be housed together under a 42-TE wide front panel in a 19-inch case. The power supply is a simple one, turning the mains voltage into 12 to 40 volts d.c. (unregulated). Of course, the output current of the PSU should be adequate for the power drivers and stepper motors you want to use. A step-down voltage converter forms the 5-volt power supply for the HCMOS logic on the motor board.

The 80C166 board may also be used to provide clock signals to other stepper motor drivers, power stages and direction inputs. This however does require an additional regulated supply voltage of 5 V/300 mA for the board. Because all inputs and outputs are designed for TTL levels, level converters may be necessary where the board is connected to power drivers or sensors. Pushbuttons and a serial interface on the other hand may be directly connected.

Pushbuttons (Halt/Go) are provided to allow manual control of the stepper motors. As soon as the control becomes complex, however, you'll soon find that the necessary control signals have to be created with the aid of a PC.

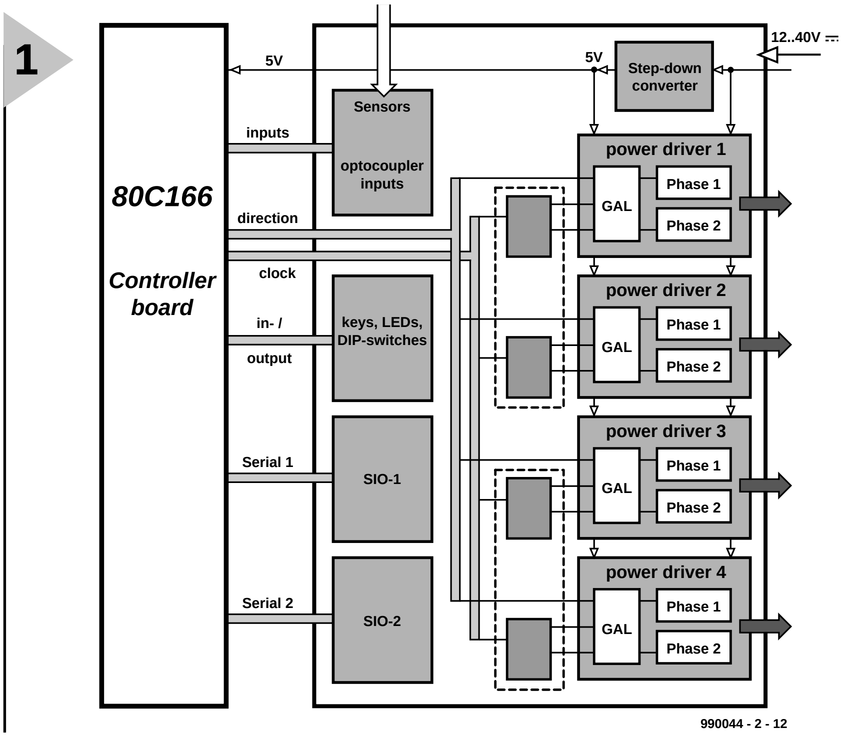


Figure 1. Block diagram of the stepper motor control. The program is hidden in blocks SIO1 and SIO2. The 80C166 microcontroller board supplies the necessary direction and clock information.

THE HARDWARE IN DETAIL

The block diagram of the SMC (stepper motor control) is given in **Figure 1**. Each of the four (identical) power drivers consists of two stepper motor ICs type PBL3717 or PBL3717/2, a programmed GAL type 16V8 and one half of a re-triggerable monostable multivibrator type 74HC123. The output of the HC123 goes inactive when no clock signal is detected for 35-50 ms. This signal may be used to reduce the motor current when the spindle is not moving. The GAL, or rather its contents, determines the step order and current distribution through the windings — its outputs eventually lead to the two stepper motor ICs. These components, in turn, use just a handful of external components to generate the phase currents for the stepper motor windings. Their outputs toggle (change polarity) as a function of the TTL level applied to pin 8 (PH). Two other TTL inputs, pin 9 and pin 7 (I0 and I1) set up a comparator threshold of about 420 mV, 250 mV, 80 mV or 0 mV. When a coil current starts to flow, the resulting voltage drop across the 1-Ω resistor (Rx1/Rx5) at pin 16 is applied to input pin 10 of the comparator. When this voltage exceeds the set level, the comparator cuts off the current. After a 'cut-off' time of about 35 µs (determined by Rx3/Rx7 and Cx2/Cx4), the current flow is established again, and the process starts again.

The power stage will only supply enough current required to either build up or maintain the magnetic field in the phase winding. This is done because the high inductive coil reactance drops to the much lower ohmic resistance of the coil once the magnetic field has been established. Alternatively, this current may be adjusted with the aid of a potentiometer. The bias level set up by voltage dividers Rx4/Rx8 and Rx2/Rx6 is applied to the comparator input where it is effectively added to the voltage drop across the 1-Ω sense resistor that enables the comparator to cut the motor current. The disadvantage of this arrangement is that it is no longer possible to control the current distribution across the phase windings because all current control inputs on the PBL3717 have to be held low with this method.

Three jumpers at the GAL inputs enable different step modes to be selected:

Jumper	GAL Pin	Function	open	closed
1	7	Step mode	Full step	Half step
2	8	Phase current	60%	100%
3	9	(Stand by)	(with Stand-by)	(w/o Stand-by)
		Stationary current	60%	100%

Other modes are possible by changing the GAL contents.

For zeropoint searching (spindle 'home' calibration) four optocoupler inputs are available for sensors or contact-less switches with a 'make' function. During a zeropoint search, the stepper motor first turns in the direction of a sensor. When this is actuated, the motor is halted for about 0.2 s. Next, the spindle is turned into the opposite direction until the sensor switches off again. The locating of the zeropoint is acknowledged, and the current motor position is defined as 'zero' or 'home'. When a stepper motor is connected up, you should take into account that the spindle will turn in the direction of the sensor during the zeropoint search.

The activation of one of the other 8 optocoupler inputs is reported individually. In this way, it is possible to detect certain states or when a certain position is reached.

The pushbuttons on the SMC front panel allow the stepper motors to be controlled by hand, as well as parameters to be entered. The preferred

2

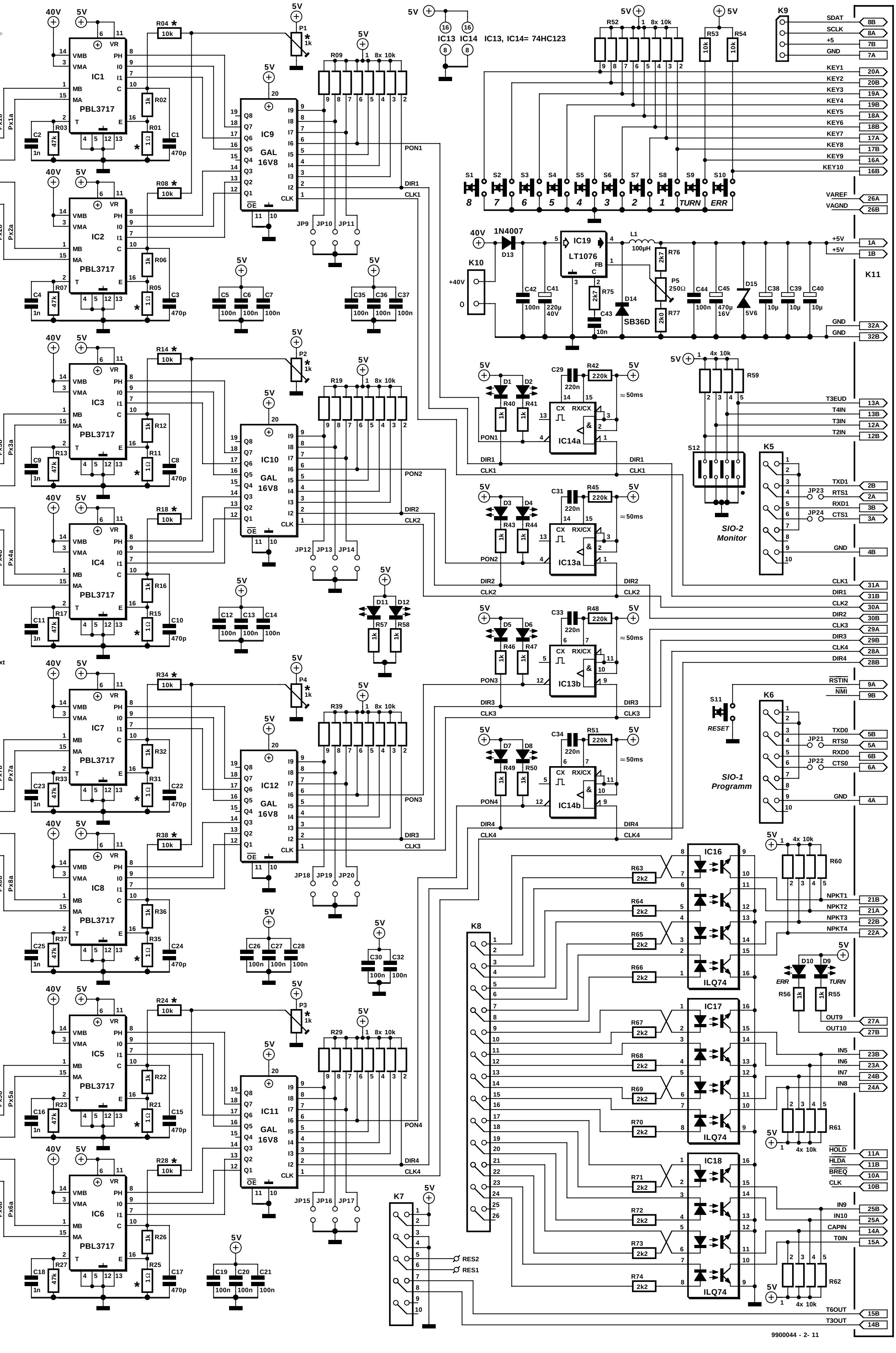


Figure 2. Complete circuit diagram of the stepper motor control (SMC). Individual circuit functions are clearly identifiable.

Connector pinouts

Function	Label	Pin (X1)			Pin (X1)		Label	Function
+5V pwr supply	+5V	1bc	1		2	1a	+5V	+5V pwr supply
O SIO 2 TxD Mon	P3.8 / TXD1	2bc	3		4	2a	P1.12	O SIO 2 RTS
I SIO 2 RxD Mon	P3.9 / RXD1	3bc	5		6	3a	P2.12/CC12IO	I SIO 2 CTS
GND SIO 2	GND	4bc	7		8	4a	GND	GND SIO 1
O SIO 1 TxD Prog	P3.10 / TXD0	5bc	9		10	5a	P1.11	O SIO 1 RTS
E SIO 1 RxD Prog	P3.11 / RXD0	6bc	11		12	6a	P2.11/CC11IO	I SIO 1 CTS
+5V Serial	+5V	7bc	13		14	7a	GND	GND Serial
I/O Serial Data	P1.10	8bc	15		16	8a	P2.10/CC10IO	I/O Serial Clock
I (not used)	NMI#	9bc	17		18	9a	RSTIN#	I Reset key
O 4	P3.15 / CLK	10bc	19		20	10a	P2.13/BREQ#	O 1
O 2	P2.14/HLDA#	11bc	21		22	11a	P2.15/HOLD#	O 3
I DIP switch Bit 4	P3.7 / T2IN	12bc	23		24	12a	P3.6 / T3IN	I DIP switch Bit 3
I DIP switch Bit 2	P3.5 / T4IN	13bc	25		26	13a	P3.4 / T3EUD	I DIP switch Bit 1
O Output T3	P3.3 / T3OUT	14bc	27		28	14a	P3.2 / CAPIN	I Pos. Sensor 7
O Output T6	P3.1 / T6OUT	15bc	29		30	15a	P3.0 / T0IN	I Pos. Sensor 8
I Key 10 (Cntrl)	P2.9 / CC9IO	16bc	31		32	16a	P2.8 / CC8IO	I Key 9 (control)
I Key 8 (1)	P2.7 / CC7IO	17bc	33		34	17a	P2.6 / CC6IO	I Key 7 (2)
I Key 6 (3)	P2.5 / CC5IO	18bc	35		36	18a	P2.4 / CC4IO	I Key 5 (4)
I Key 4 (5)	P2.3 / CC3IO	19bc	37		38	19a	P2.2 / CC2IO	I Key 3 (6)
I Key 2 (7)	P2.1 / CC1IO	20bc	39		40	20a	P2.0 / CC0IO	I Key 1 (8)
I Motor zero SM-1	P5.0 / AN0	21bc	41		42	21a	P5.1 / AN1	I Motor zero SM-2
I Motor zero SM-3	P5.2 / AN2	22bc	43		44	22a	P5.3 / AN3	I Motor zero SM-4
I Pos. Sensor 1	P5.4 / AN4	23bc	45		46	23a	P5.5 / AN5	I Pos. Sensor 2
I Pos. Sensor 3	P5.6 / AN6	24bc	47		48	24a	P5.7 / AN7	I Pos. Sensor 4
I Pos. Sensor 5	P5.8 / AN8	25bc	49		50	25a	P5.9 / AN9	I Pos. Sensor 6
GND	VAGND	26bc	51		52	26a	VAREF	+5V
O LED Cntrl	P1.9	27bc	53		54	27a	P1.8	A LED control
O Direction SM-4	P1.7	28bc	55		56	28a	P1.6	O Clock SM-4
O Direction SM-3	P1.5	29bc	57		58	29a	P1.4	O Clock SM-3
O Direction SM-2	P1.3	30bc	59		60	30a	P1.2	O Clock SM-2
O Direction SM-1	P1.1	31bc	61		62	31a	P1.0	O Clock SM-1
GND pwr supply	GND	32bc	63		64	32a	GND	GND pwr supply

Function	Name	Pin (X2)			Pin (X2)		Name	Function
+5V pwr supply	+5V	1	1		2	6	+5V	+5V pwr supply
GND pwr supply	GND	2	3		4	7	GND	GND pwr supply
Spare 2	Res 2	3	5		6	8	Res 1	Spare 1
Output T6	P3.1 / T6OUT	4	7		8	9	P3.3 / T3OUT	Output T3
		5	9		10		-	
(X2 is not normally used and provided for extensions)								

Function	Name	Pin (X3)			Pin (X3)		Name	Function
	DTR / DSR0	1	1		2	6	DTR / DSR0	
Output TxD1 Prog	P3.10 / TXD0	2	3		4	7	P1.11	Output RTS1
Input RxD1 Prog	P3.11 / RXD0	3	5		6	8	P2.11 / CC11IO	Input CTS1
	DTR / DSR0	4	7		8	9	GND	GND pwr supply
GND pwr supply	GND	5	9		10		-	

Function	Name	Pin (X4)			Pin (X4)		Name	Function
	DTR / DSR1	1	1		2	6	DTR / DSR1	
Output TxD2 Mon	P3.8 / TXD1	2	3		4	7	P1.12	Output RTS2
Input RxD2 Mon	P3.9 / RXD1	3	5		6	8	P2.12 / CC12IO	Input CTS2
	DTR / DSR1	4	7		8	9	GND	GND pwr supply
GND pwr supply	GND	5	9		10		-	

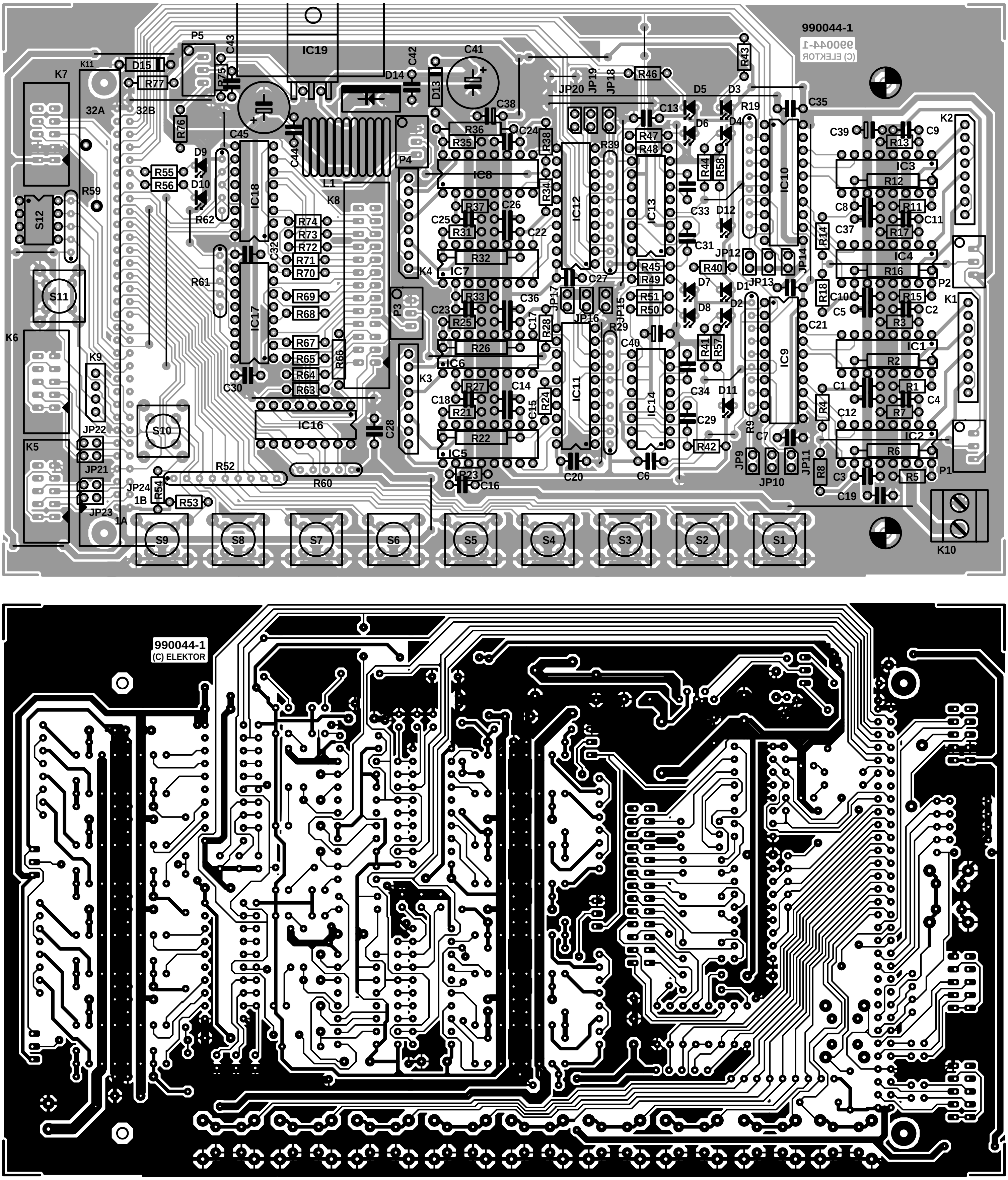


Figure 3. Copper track layout and component mounting plan of the single-sided PCB designed for the SMC. Mechanically it is a perfect match to the 80C166 controller board.

method of parameter inputting and control is however by means of the PC and its serial interface. The SMC is capable of reading parameters and commands via its serial input. The serial link is also used to return information to the PC.

With a hot needle...

To build up the stepper motor control board you will need solder tin with a diameter of not more than 1 mm. The solder iron should have a fine bit and a tip temperature of about 340 degrees

COMPONENTS LIST

Resistors:
R1,R5,R11,R15,R21,R25,R31,R35 = 1Ω (1W for 1.2-A version)
R2,R6,R12,R16,R22,R26,R32,R36,R40, R41,R43,R44,R46,R47,R49,R50, R55-R58 = 1kΩ
R3,R7,R13,R17,R23,R27,R33, R37 = 47kΩ
R4,R8,R14,R18,R24,R28,R34,R38,R53, R54 = 10kΩ
R9,R19,R29,R39,R52 = 8×-way 10kΩ SIL array
R10,R20,R30 = not used
R42,R45,R48,R51 = 220kΩ
R59-R62 = 4-way 10kΩ SIL array
R63-R74 = 2kΩ2
R75,R76 = 2kΩ7
R77 = 2kΩ0
P1-P4 = 1kΩ multiturn preset, vertical, (top adjustment)
P5 = 250Ω multiturn preset, vertical, (top adjustment)

Capacitors:
C1,C3,C8,C10,C15,C17,C22,C24 = 470pF
C2,C4,C9,C11,C16,C18,C23,C25 = 1nF
C5,C6,C7,C12,C13,C14,C19,C20,C21, C26,C27,C28,C30,C32,C35,C36,C37, C42,C44 = 100nF
C29,C31,C33,C34 = 220nF
C38,C39,C40 = 10µF 16V tantalum
C41 = 220µF 40V
C43 = 10nF
C45 = 470µF 16V

Inductors:
L1 = 100µH 1A, max. dia. 15mm

Semiconductors:
D1-D8,D10 = LED, red
D9 = LED, yellow
D11,D12 = LED, green
D13 = 1N4007
D14 = SB360 or SB550 (Schottky diode, 3A)
D15 = 5V6 1W zener diode
IC1-IC8 = PBL3717A or PBL3717/2N (ST-Microelectronics)
IC9-IC12 = GAL16V8 (order code **996524-1-a+b**, 2 pcs)
IC13,IC14 = 74HC123
IC15 = not used
IC16,IC7,IC18 = ILQ74
IC19 = LT1076 CT

Miscellaneous:
JP1-JP8 = not used
JP9-JP24 = jumper
K1-K4 = 8-way PCB plug (Conrad Electronics o/n 741256)
K5,K6,K7 = 10-way boxheader
K8 = 26-way boxheader
K9 = 4-way SIL header
K10 = 2-way PCB terminal block
K11 = 64-way pinheader
S1-S11 = pushbutton, 1 make contact (ITT/Schadow PVA10AH2)
S12 = 4-way DIP-switch
Heatsink for voltage regulator
PCB, order code **990044-1**
Disk (source code files), order code **996031-1**
On 80C166 board:
H-EPROM = order code **996525-1**
L-EPROM = order code **996525-2**

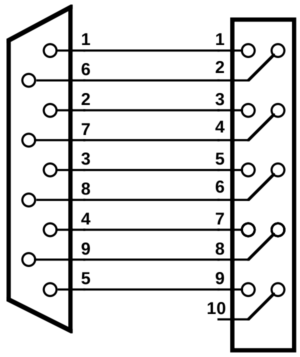
Celsius. The wire jumpers being the 'components' with the lowest profile, they are first bent to shape, fitted and soldered. Next come the resistors, IC sockets, capacitors and so on, until all components are fitted except the integrated circuits, the pushbuttons and the LEDs. The step-down regulator IC is a tall component that has to be fitted flat on the board. Carefully insert its pins into the holes in the board, secure a small heatsink to it, and then solder the pins.

The connection to the 80C166 board is made via a 20-mm long pinheader at the solder side of the board. To make sure it fits correctly, insert the pinheader in the socket on the 80C166 board and secure the SMC board on top using fours PCB spacers with a height of 20 mm. Next, fix the pinheader by soldering its corner pins and two centrally located pins. Finally you remove the controller board again and solder the remaining pins on the pinheader.

Next, concentrate on the pushbuttons and the LEDs. The pushbuttons are fitted with their caps mounted on them, and the LEDs, with spacers. Now mount the front panel on to the SMC board using four 15-mm high PCB spacers. The layout and drilling template of the front panel will be given in next month's instalment. Check that no components are squashed between the board and the front panel! Once the LEDs and the pushbuttons are seated in their respective holes, their connecting terminals may be soldered.

A cable adapter is needed to enable the SMC to communicate with the outside world. It consists of piece of flatcable, an IDC-style 10-way pinheader and a 9-way sub-D socket, also for flatcable mounting. Its 'schematic' is given in **Figure 4**.

4



990044 - 2 - 13

Figure 4. A DIY cable adaptor.

The pinheader and the sub-D connector are aligned on the flatcable ends, clamped secure in a vise and then pressed on to the cable by slowly closing the vise. Do not forget the strain relief for the flatcable. On the pinheader, an arrow marks pin 1. Unused wires at either end of the cable may be cut off. A 26/25-way way cable adaptor for the inputs is made in the same way.

Having read this instalment you have roughly a month to build the electronics. Next time, we will again tackle practical matters like taking the SMC in use and operating it. Faultfinding will also be discussed, as well as the operation of the system software developed by the author.

(990044-2)

Figure 5. Completed prototype of the SMC board. Compare this carefully with your own work!