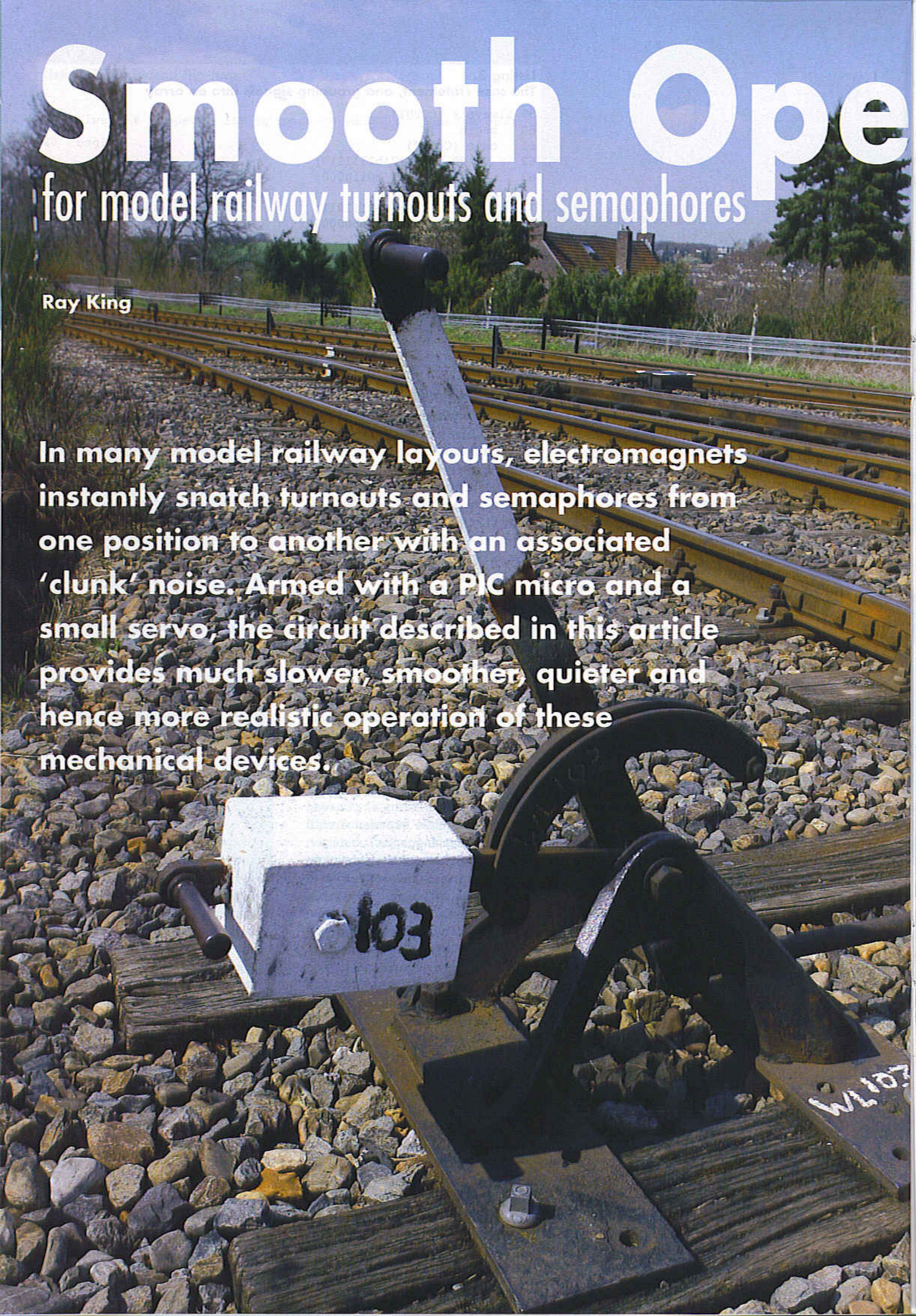
Smooth Ope

for model railway turnouts and semaphores

Ray King

In many model railway layouts, electromagnets instantly snatch turnouts and semaphores from one position to another with an associated 'clunk' noise. Armed with a PIC micro and a small servo, the circuit described in this article provides much slower, smoother, quieter and hence more realistic operation of these mechanical devices.



rator

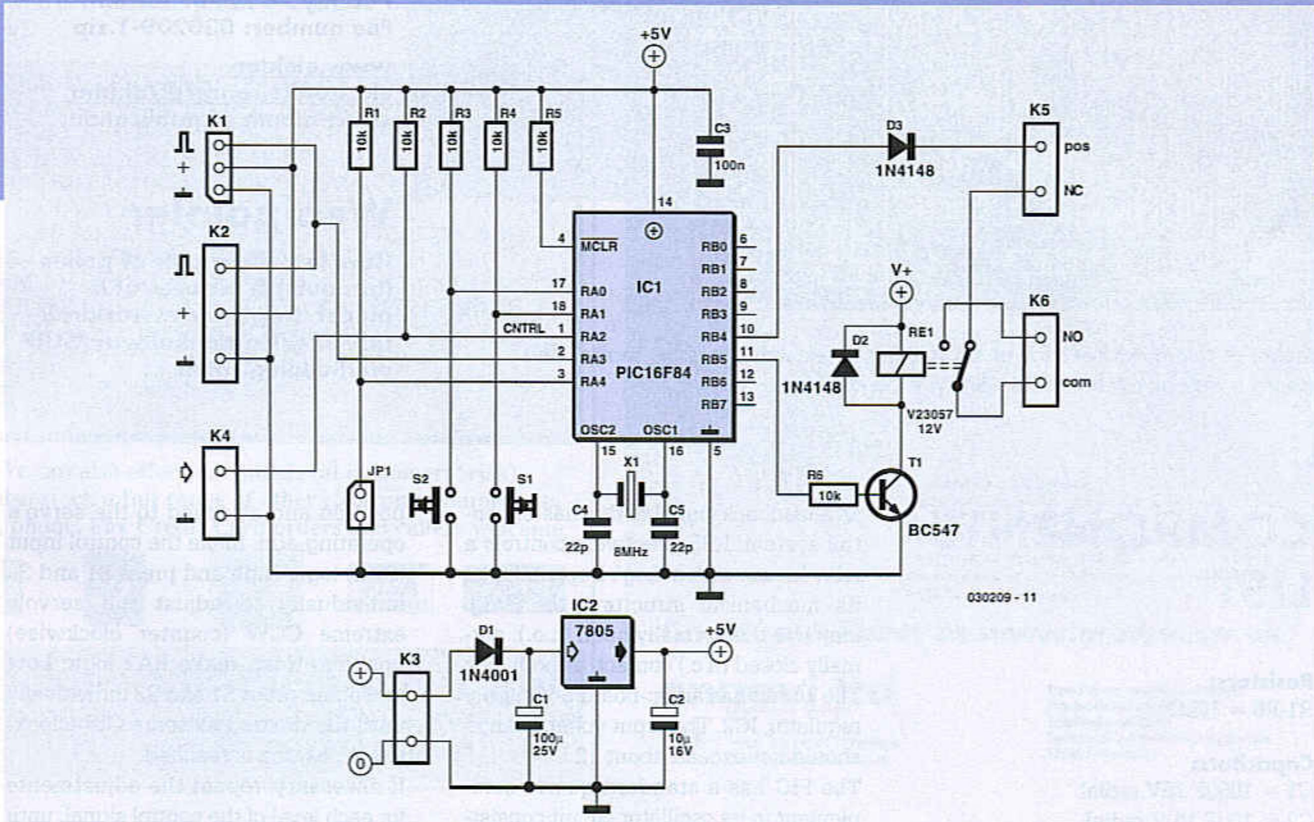


Figure 1. At the heart of the circuit we find the trusty PIC16F84 microcontroller.

There are a large number of model railway builders using proprietary track and points to realise their layouts. When adding remote controls for points (turnouts) there is often disappointment with the unrealistic and noisy movement of the mechanism. There are commercially available motorised units but these are expensive as well as difficult to install. A further disadvantage of commercial units is that the force they use is often detrimental to fine scale turnouts in that it can easily cause damage if not precisely adjusted and maintained.

On a different track

The proposed design uses standard radio-control (R/C) servos to drive the turnouts. A PIC microcontroller is used to generate the necessary PWM pulses under the control of a simple on/off switch. The circuit includes a facility to switch the 'frog' polarity depending on the position of the turnout. In railway terminology, a frog is 'a grooved piece of iron at place in railways where tracks cross'. There is also a signal returned to the operator to indicate correct operation of the unit for display on a track diagram, for example. The turnout will move slowly and smoothly from one position to the other with no excess travel or brute force that will damage the turnout. If you are interested in making your own pair of points (also known as a 'turnout') then visit the webpage listed at the end of this article — it also shows a 'frog'.

Since control is by a single on/off switch using standard TTL levels, it is also simple to interface the unit to an automatic or computer-controlled system (like the one described elsewhere in this issue).

So why use a PIC when the design could be accommodated with some simple hardware? The present design uses very few components, is easy to set up and the speed of movement can be adjusted by changing parameters in the code.

Circuit description

Figure 1 shows the circuit diagram of the smooth control for model railway turnouts. Basically, a PIC microcontroller sits between input and output connectors. The inputs include the 'turnout change' control signal supplied by the master control unit or a simple switch on your control panel. This signal arrives at PIC input RA2 via connector K4. The two other input devices are pushbuttons S1 and S2 which are read via PIC port lines RA1 and RA0 respectively. They are used to set the servo action required to make the turnout reach its two extreme positions. Looking at output connectivity, we find the servo control pulses being supplied by PIC port line RA3 and fed to the servo by way of connector K1 and /or K2. Port line RA4 is read to detect the presence or absence of jumper JP1 which selects between two memory settings (more about this further on). Port line RB4 supplies control information about the turnouts status, for use

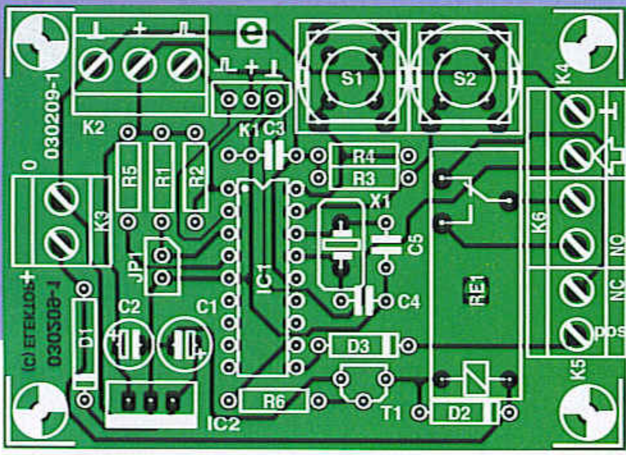


Figure 2. Component overlay of the PCB designed for the turnouts control. The board is available ready-made through The PCBShop.

Free Downloads

PIC microcontroller software.
File number: 030209-11.zip

PCB layout in PDF format.
File number: 030209-1.zip

www.elektor-electronics.co.uk/dl/dl.htm, select month of publication.

Web pointer

How to build a pair of points (turnouts) & pictures of a model 'frog': www.worldrail-fans.org/GardenRailway/SLRPointBuilding.shtml

COMPONENTS LIST

Resistors:
R1-R6 = 10kΩ

Capacitors:
C1 = 100µF 25V radial
C2 = 10µF 16 V radial
C3 = 100nF
C4,C5 = 22pF

Semiconductors:
D1 = 1N4001
D2,D3 = 1N4148
T1 = BC547
IC1 = PIC16F84-10P, programmed, order code **030209-41**
IC2 = 7805

Miscellaneous:
JP1 = 2-way pinheader with jumper
K1 = 3-way pinheader
K2 = 3-way PCB terminal block, lead pitch 5mm
K3-K6 = 2-way PCB terminal block, lead pitch 5mm
S1,S2 = pushbutton with 1 make contact, e.g., type D6-R
X1 = 8MHz quartz crystal
RE1 = relay, PCB-mount, 12V SPDT, e.g., Siemens V23057
PCB, available through **The PCBShop**
Disk, Proton PIC Basic Plus source code, order code **030209-11** or **Free Download**

as a feedback signal to the master control system. RB5 effectively controls a relay for use with a 'frog'. Depending on its mechanical structure, the 'frog' employs the normally open (n.o.), normally closed (n.c.) contact, or both.

The circuit has an on-board 5-V supply regulator, IC2. The input voltage range should not exceed about 12 VDC.

The PIC has a standard quartz complement in its oscillator circuit consisting of an inexpensive 8-MHz quartz crystal X1 and the two usual small loading capacitors, here C4 and C5.

Printed circuit board

The printed circuit board (PCB) for the turnouts control has been spaciously laid out. What's more, it contains standard size components only. The component overlay is given in **Figure 2**. Easy to use PCB terminal blocks with 5 mm pin spacing are used for the connectors, except K1 which is a 3-pin pinheader for use with ready-made servo cables. However, in all cases where you are not certain about the servo connections, use terminal block K2 instead. The completed and tested board should be mounted out of sight, which in nearly all cases will mean securing it to the underside of the model railway tabletop. If you do not need 'frog' control, then components R6, T1, D2, K6 and Re1 may be omitted.

Setting up

There are two calibration pushbuttons, S1 and S2. Press them simultaneously and the servo will adopt a central position. The turnout is held in its central position and attached to the servo's operating arm. Make the control input (RA2) logic High and press S1 and S2 individually to adjust the servo's extreme CCW (counter clockwise) position. Next, make RA2 logic Low and again press S1 and S2 individually until the desired extreme CW (clockwise) position is reached.

If necessary, repeat the adjustments for each level of the control signal, until the servo drives accurately and smoothly from one position to the other. The settings are automatically stored in the PIC's internal EEPROM. Jumper JP1 directs the PIC to use an alternative memory location permitting two settings to be used. Its use is optional.

Software

The source file (.BAS) was written in Proton PIC Basic Plus and should not be too difficult to convert to other PIC compilers. The Proton environment also generates an assembler file which, together with the Basic listing should provide enough clues to adapt the program and assemble it with your favourite assembler for the PIC16F84.

The circuit could also be used to operate semaphore signals. By adjusting the parameters for operating speed and/or modifying the program code it should even be possible to mimic the 'bounce' of the semaphore arm as it rises and falls.

(030209-1)

Location photograph courtesy South Limburg Steam Railway Foundation (www.zlsm.nl)