Multi-channel Failsafe for Radio Controlled Models

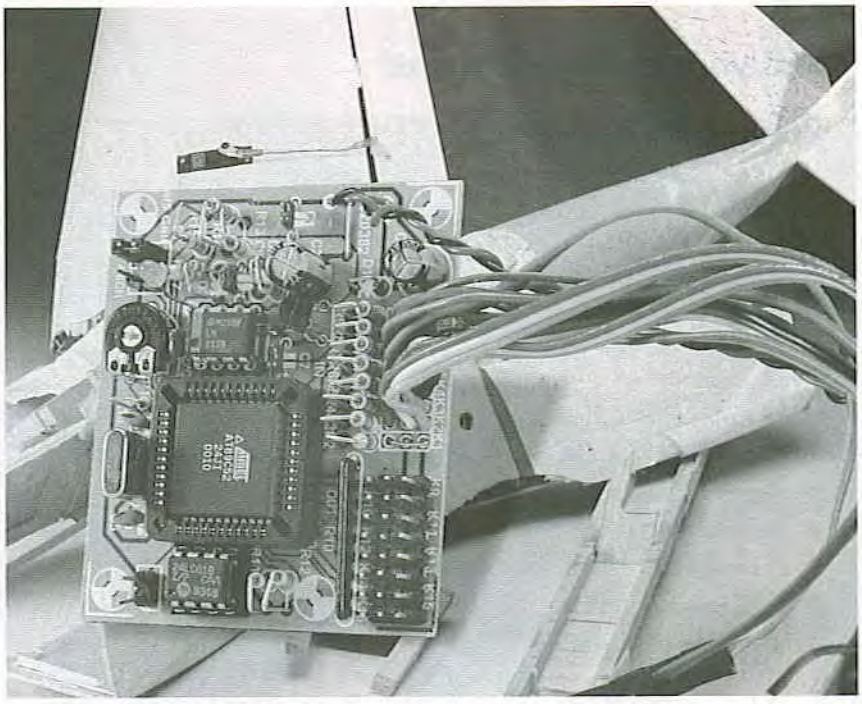
Happy landings!

Design by Lex Cunningham

If you do not want your expensive R/C model to crash or disappear forever when the radio link fails or is subject to heavy interference, you need to preload your model with servo settings that 'take over' under adverse radio or battery conditions ensuring a landing that's as gentle and safe as possible. This circuit does it all, and more!

The Multi-Channel Failsafe was developed to prevent damage to radio controlled models when the on-board radio control receiver starts to receive erratic control information, or when the craft supply voltage drops below a predefined 'safe' level. Detection of corrupt frames in the received signal is handled by a microcontroller. At a certain level of insecurity, the micro loads a number of servo settings from an EEPROM that (hopefully) result in a safe landing, or at least, minimum damage to the model. The most important setting in these cases is, without doubt: cut off the engine (i.e., get it to idle)!

While such 'failsafe' facilities are available on high-end radio control systems, the cost is prohibitive to many occasional and weekend modellers. Many failsafe circuits have been published and commercial units are available, however these are only for a single channel and do not detect low voltage of the on-board battery. Setting up these units is also fiddly with the failsafe position set with a potentiometer.



Advantages and basic functionality

The Multi Channel Failsafe described in this article is flexible and easy to use, fits between the receiver and servos, provides up to eight simultaneous channels, monitors the battery voltage and is switch selectable for use with four or five cell battery packs. It will continually measure and compare each servo pulse, as well as the entire frame and also provide a 'lost model' alarm.

The failsafe state is entered if one of the following three conditions is detected by the microcontroller software in conjunction with suitable hardware:

- an entire frame is missing due to complete loss of received signal;
- a distorted-pulse counter reaches a threshold — this is useful when the model is near the edge of radio range;
- a low battery voltage is detected —this causes a repeated cycle of one second of failsafe positions, followed by five seconds of normal control. The resulting motions of the model are easily seen and heard from a distance and represent rather compelling reasons to get your expensive model back ashore, on the ground or in the pit!

The failsafe state is left, and the circuits acts as a bypass again for received servo signals when one of the following conditions is detected:

- a valid frame is received;
- the distorted-pulse counter drops below the threshold;
- the battery voltage returns to normal.

Programming the failsafe positions consists of setting the desired 'safe' positions using the RC transmitter, i.e., throttle to idle and other controls (rudder, aileron, steering, etc.) to neutral, and then pressing the Store button.

This causes the desired servo positions to be stored in EEPROM, from where they can be retrieved by the micro, but only when failsafe action is required. In all other cases, the circuit 'does nothing' to the servo commands you transmit to your model.

Circuit description

The failsafe unit is based on an AT89C52 microcontroller in a PLCC case. The circuit diagram in **Figure 1** shows a typical application of a microcontroller. The main thing to understand about the circuit is that it is totally transparent to signals travelling from the eight IN connectors to the associated OUT connectors **as long as no problems exist with the radio link and/or the on-board battery**.

Having stated that the circuit is a standard microcontroller application, we should hasten to add that some analogue circuitry has been thrown in! A Low Battery condition is detected by a comparator circuit around IC3.A. The circuit supply voltage is normally stabilized to 5 volts by the circuit around IC4 — after all, five NiCd cells can supply anything between about 5.5 volts and 6.75 volts when exhausted or fully charged respectively.

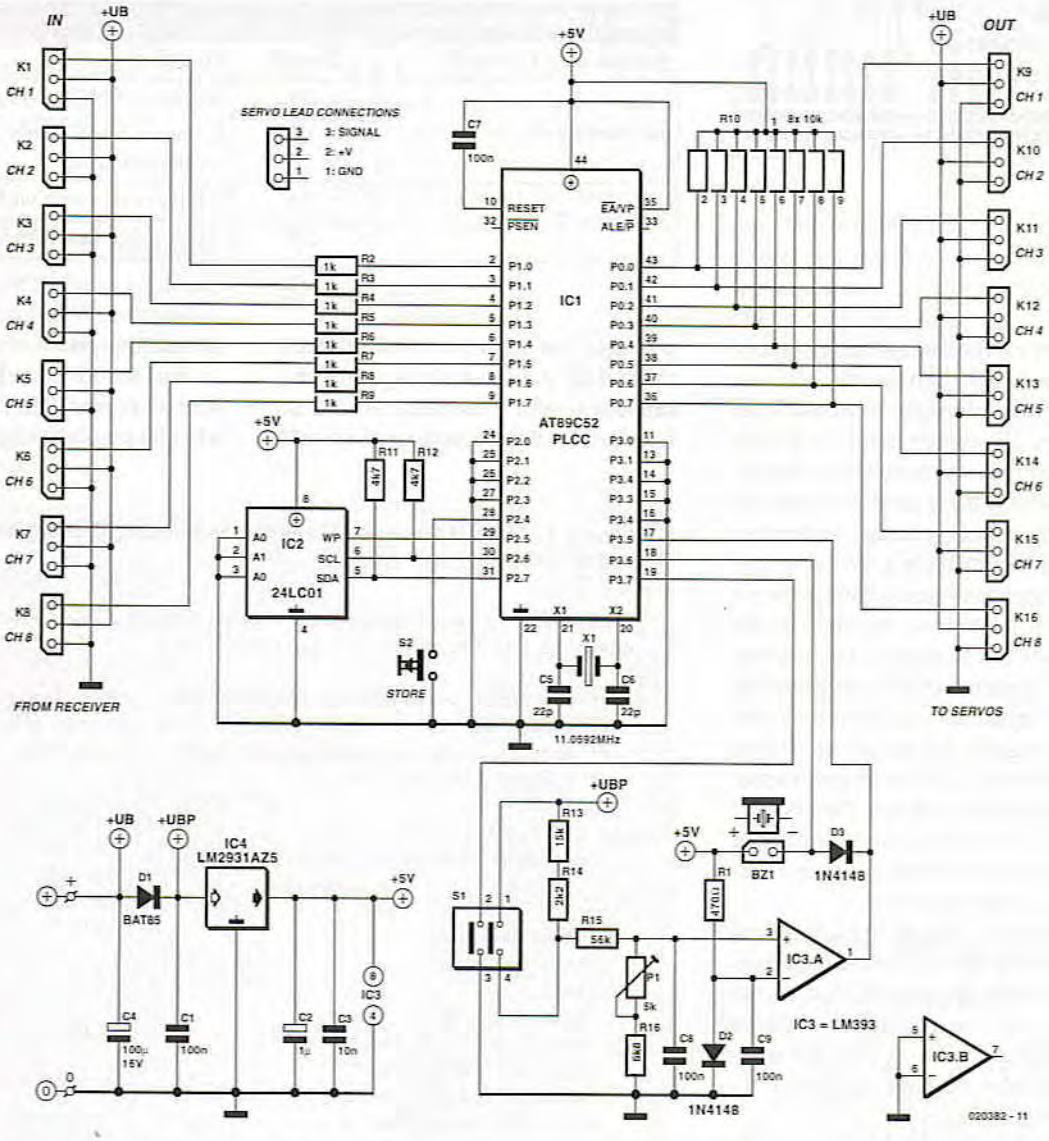


Figure 1. Circuit diagram of the Failsafe for R/C Models.

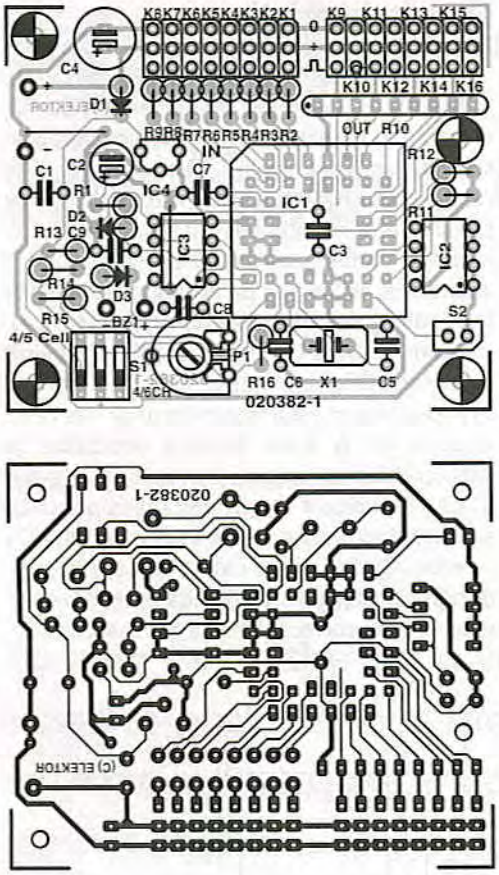


Figure 2. Printed circuit board layout (board available ready-made).

COMPONENTS LIST

Resistors:
R1 = 470Ω
R2-R9 = 1kΩ
R10 = 8-way 2kΩ2 SIL array
R11,R12 = 4kΩ7
R13 = 15kΩ
R14 = 2kΩ2
R15 = 56kΩ
R16 = 6kΩ8
P1 = 5kΩ preset H

Capacitors:
C1,C7,C8,C9 = 100nF
C2 = 1μF 16V radial
C3 = 10nF
C4 = 100μF 16V radial
C5,C6 = 22pF

Semiconductors:
D1 = BAT85
D2,D3 = 1N4148
IC1 = AT89C52-24JI, programmed, order code **020382-41** (see Readers Services page)
IC2 = 24LC01 (2V7)
IC3 = LM393
IC4 = LM2931AZ5 (Farnell # 412480)

Miscellaneous:
BZ1 = active (DC) buzzer, 5V
K1-K8 = 3-way servo plug
K9-K16 = 3-way SIL pinheader
S1 = 2-way DIP-switch (3-way optional, one contact not used)
S2 = pushbutton, 1 make contact
X1 = 11.0592MHz quartz crystal
PCB, order code **020382-1** (see Readers Services page)
Disk, microcontroller source code, order code **020382-11** (see Readers Services page) or **Free Download**.

Table 1 DIP switch functions

Switch contact on S1	On/Off	Function
S1-1 (pins 1-4)	On	Supply = 4 NiCd cells (4.8 V nom.)
	Off	Supply = 5 NiCd cells (6.0 V nom.) (recommended)
S1-2 (pins 2-3)	On	5-8 servo channels with failsafe
	Off	4 servo channels with failsafe

The LM2931A was chosen for its very low quiescent current, low voltage drop (max. 0.6 V), short-circuit and overload protections.

One switch in DIP switch block S1 allows you to select between operation from four or five NiCd cells. The setting you'll want to use for your particular model boat, car, helicopter or plane is picked from **Table 1**. When a four-cell NiCd battery pack is used, voltage regulator IC4 will not be able to regulate as its input voltage will be just 4.4-5.4V less the drop across D1 (approx. 0.2 V). Adding the drop across the (inactive) regulator itself, the supply voltage to the microcontroller may well become as low as 3.25 V and vary in proportion with the battery voltage. That is why we recommend the five-cell variety unless it is absolutely mandatory to use a four-cell battery pack for the model involved.

The 'battery-low' voltage level is accurately adjusted by means of preset P1. When the desired threshold is reached, that is, the battery voltage you consider unsafe for your model, buzzer Bz1 will sound. The 'battery-low' condition is also noticed by the microprocessor via port line P3.5.

The micro can also actuate the buzzer via P3.6 which is programmed to act as an output line. The buzzer will sound when the model is 'lost' and may help you to recover your prized model aircraft from a corn field, tree branches or a swimming pool.

Port line P3.7 reads the state of contacts 2 and 3 in DIP switch block S1. The available options are again

Figure 3. Extract from the C source code listing showing how the pulse width is measured in software.

```
unsigned char get_input_pulse_widths(unsigned char *seq_ptr,
unsigned int *sav_ptr, unsigned char *ft)
{
  /* get_input_pulse_widths measures the input pulse widths and   */
  /* stores the results in an array. 0 is returned when the frame */
  /* appears within a predetermined time. 1 is returned if there is */
  /* a frame timeout.                                             */

    bit timer_flag, first_flag;
    unsigned char fail, int_hi, int_lo;
    unsigned int input_timer;

    timer_flag = 0;
    first_flag = 1;
    fail = 0;
    fail_int = 0;
    int_hi = INT_HI_FRAME;
    int_lo = INT_LO_FRAME;
    while (*seq_ptr != 0xFF && !fail) {
        if (!timer_flag) {
            set_int_timer(int_hi, int_lo);
```

given in Table 1.

When pushbutton S2 (STORE) is pressed, the failsafe settings for the CH1-CH8 servos on the corresponding connectors K9-K16 are copied from the transmitter (i.e., read from K1-K8) and stored in a 24LC01 I^2C EEPROM (IC2). The microcontroller software contains a special routine to do all the talking with the 24LC01 using just three port lines. Two of these have been programmed to act as the SDA (serial data, P2.7) and SCL (serial clock, P2.6) lines, the other, to control the Write Protect (WP) function of the 24LC01. Note that a low-voltage (2.7 V) version of the 24LC01 is specified for this circuit.

One more point to note is the use of an 11.0592 MHz crystal. Although this is a dead common frequency in MCS51/52 land, we should not forget to mention that it was deliberately chosen so that any oscillator harmonics would not fall within the 27, 29, 35, 36, 40, 50, and 72 MHz R/C bands allowed in the UK.

Construction

As usual, Elektor's renowned PCB design department has succeeded in producing an extremely compact printed circuit board. What's more, the board is **single-sided** hence relatively cheap and easy to make yourself. Even better, the board can be supplied ready-made through our Readers Services, the order code being **020382-1**.

The PCB artwork is shown in **Figure 2**. Start the construction by fitting the two wire links so they are not forgotten later. SIL resistor array R10 is a polarized component whose 'common' terminal is usually identified by a dot on the body. This dot is also shown on the component overlay. Still on polarised components, the microcontroller is seated in a 44-pin PLCC socket — note the bevelled edge at one of its corners. Many parts are mounted upright. The areas where the two groups of eight (maximum) servo cables are connected to the board are quite crowded. If you do not want to use pinheaders, the servo wires may be soldered directly to the board. Either way, you need to get the servo wires connected the right way around, so if necessary consult the manufacturer's data in this respect. The location of the 'pulse' pins is clearly marked on the overlay, as well as by the inset drawing in the circuit diagram. The other two pins of each servo plug are ground and the positive supply (+UB).

Software

The software executed by the AT89C52 micro in this circuit was written in C using the freeware SDCC C compiler. The source code is available on diskette (order code **020382-11**) or a Free Download from our website (same number, see month of publication). Those of you without access to an assembler or a programmer for the Atmel AT series can buy the microcontroller ready-programmed through our Readers Services under order code **020382-41**.

The source code has been extensively commented. The main parts of the code are the input sequencing and pulse width measurement functions. The latter is illustrated by the code snippet in **Figure 3**. Lex tells us that some innovative techniques have been developed to achieve correct framing of the input signals. By making the software freely accessible, Lex and *Elektor Electronics* jointly encourage readers to modify and extend the code.

(020382-1)

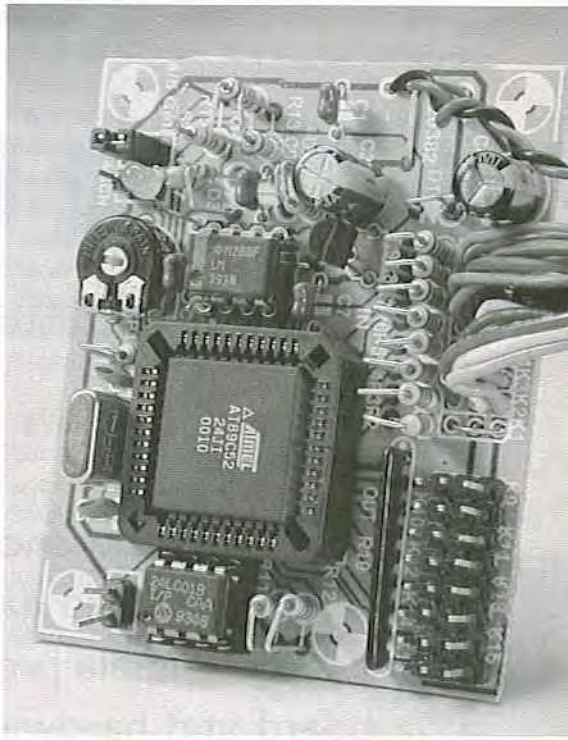


Figure 4. A close-up of our finished and tested prototype. As you can see, we soldered the servo wires directly to the board.

```
            j = *seq_ptr;
_asm
            mov     a,_j
00001$:
            cjne    a,p1,00002$
            sjmp     00003$
00002$:
            jnb     _fail_int,00001$
00003$:
_endasm ;
            set_start_timer(8);
            if (first_flag) {
                *ft = TIMER_2_HI;
                first_flag = 0;
                TIMER_2_LO = 0x00;
                TIMER_2_HI = 0x00;
            }
        }
        else {
            timer_flag = 0;
        }
        if (fail_int) {
            fail = 1;
            fail_int = 0;
```

Free Downloads

Microcontroller source code. File number: 020382-11.zip.

PCB layout in PDF format. File number: 020382-1.zip.

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