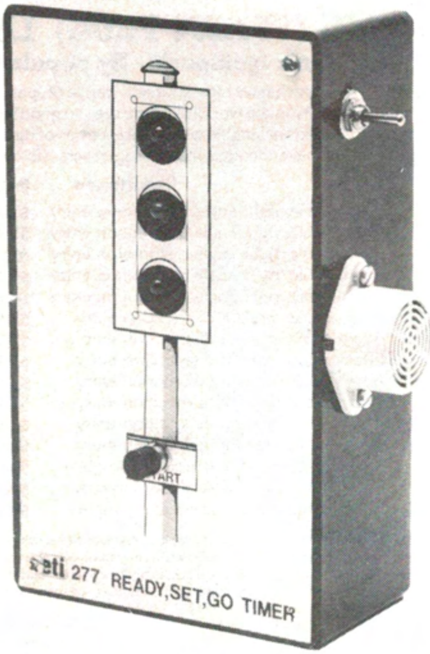
Project 277

Ready-set-go! lights



No more cries of "cheat, cheat!" during starts on the slot car track or other simultaneous start games. This project will start competitors with the familiar red-amber-green light system which has a random delay between the amber and green indicators.

Robert Irwin

MOST PROJECTS are conceived by the 'suggestive' technique. That is, someone — a staff member, a reader or an advertiser — suggests such-and-such a project would be a good idea. After a little massaging through the office consensus system (rubbish!/ripper!/might be/ho ho), the idea makes it to the 'breadboard' stage in the lab. or gets discarded along the way.

This project went through a somewhat *different* gestation.

Prologue

One dark and stormy night in an uncharted portion of the ETI lab, strange events were taking place which would have a profound and deep-reaching effect on the lives of many of us here at ETI. It started innocently enough. Peter Ihnat had just finished his electronic Tug'o'War game (ETI, August 1984) and had called in a few of the editorial staff for a quick demonstration. Roger played with the on/off switch for several seconds and grunted his approval.

"Nice colour!" exclaimed Jennie enthusiastically from the doorway. She was not noted for her wit (thought Robert in his usual dim-witted and humourless fashion).

The games began innocuously enough, with much button pushing from one and all and titters of mirth resounding around the cavernous interior of the lab. The place took on the air of a carnival and there was even talk of abandoning the August issue in favour of a party.

"Hey! This Tug'o'War is really fun," said Jon, who was not known for his wit. "Why don't we keep score?"

And so it came to pass that the great lab staff versus editorial staff Tug'o'War games began. It was soon noted, however, that the scoreline somewhat favoured the lab staff by the margin of ten to nil. This trend may have continued well into the night but, alas, one of the more cerebral of the editorial staff put two and two together (finally getting four with the aid of an HP-15C) and realised that it was also the lab staff who were saying the ubiquitous "Ready, Set, Go".

It was decided that someone, other than the participants, should say the sacred words. David, a member of the elitist Draughting Club, was given the honour. All should have been well, and would have been had not David been caught giving secret signals to the lab staff before reciting his trusted oath (it was later discovered that the lab staff had anticipated the events and had plied David with liberal quantitites of scotch and promises of a free engineering degree to gain his cooperation).

The editorial staff were livid, Roger, his face red and his left eye twitching, mumbled something about placing an ad in the employment section of the Herald. From all round the lab cries of "cheat, cheat" could be heard coming angrily from the advancing mob.

All of the sudden a piercing scream like that of a tortured animal filled the air, Jennie, clutching a live soldering iron (Weller, type WTCPN) and harbouring years of pent-up frustrations over late projects, lunged madly at Geoff's throat and attempted to solder his head to Peter's right leg.

Things could have gotten ugly then had I not been overcome with a blinding vision. "Wait!" I cried, "I have the answer".

Prying the soldering iron out of Jennie's trembling fingers I set furiously to work and, many hours and a myriad of burnt out CMOSs later, I emerged from the lab triumphant, carrying the prototype of what is now the ETI-277 Ready, Set, Go Timer. This secret can now be yours. Read on . . .

Design details

This circuit was beyond my design capabilities and I couldn't find anything similar to copy. I asked Geoff for some advice but he said he didn't know how to do it (he's a bit shy). Anyway, he was too busy to be bothered about electronics as he had to organise the next curry lunch at the Broadway pub. And after that he wouldn't care.

I asked Peter if he could give me a few hints on the circuit — like a complete design and layout of the pc board. Then I could probably manage the 'Parts List'. But Peter had taxed his mental capacities to their limit on a wooden box. One side was a bit shorter than the other (like Roger's arms) and the solder wasn't holding the wood together very well. So I left him to sit on it and I phoned the crisis counselling service (there was no aid from the venereal diseases hotline).

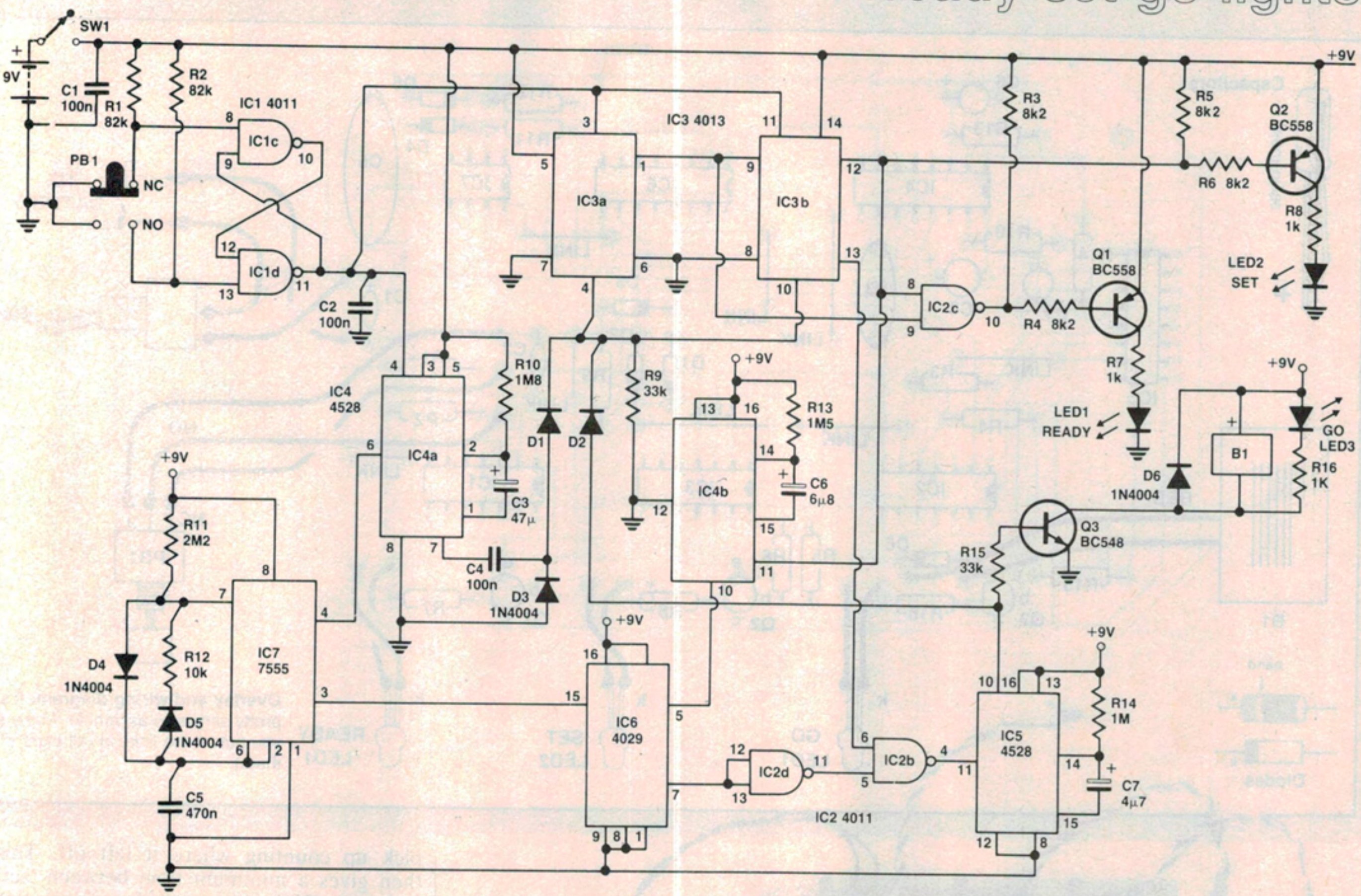
The aim of the exercise was to develop a device which would give some sort of random time period between the press of a pushbutton and the firing of a buzzer or some such thing. This would enable a game, such as the ETI Tug'o'War or a slot car race, to be started fairly without any of the competitors knowing exactly when the gun was going to go off. The final design consists of a four-bit counter, a couple of flip-flops and a handful of monostables.

It was decided to use CMOS chips because of their ability to be run directly off a 9 V battery without any regulation and their low power consumption.

The heart of the circuit is a 4029 four-bit counter. This is configured as a binary-up counter. The counter will count continuously from 0 to 15 and repeat this cycle until it is disabled. At the end of every 0-to-15 count a *terminal count* signal is generated in the form of a low-going pulse.

The terminal count signal is connected through some gating circuitry to the trigger of a monostable which, when fired, lights a green LED and sounds a buzzer. The gating on the trigger is arranged so that the mono will only fire on a terminal count pulse immediately following the press of a pushbutton and will ignore all other terminal count pulses.

Three LEDs are used to indicate the status of the circuit. A red LED indicates



HOW IT WORKS ETI-277

To understand how this circuit works let us look at the sequence of events after each press of the pushbutton. IC1c and IC1d form an RS flip-flop to debounce the pushbutton. Initially the two D-type flip-flops of IC3 are reset i.e: the outputs are low. When the pushbutton is pressed once, two things happen. Firstly, IC3a and IC3b both receive a clock pulse. The input to the first flip-flop IC3a is tied high and the input of the second flip-flop is connected to the output of the first. Therefore, on the first clock pulse the output of flip-flop one goes high and the output of flip-flop two remains low. This causes the output of the NAND gate IC2c to go low which turns on Q1 and lights LED1.

The second thing that happens is that the monostable IC4a is triggered. IC4a is configured as a retriggerable monostable which is triggered on a low to high transition of the trigger input. The output pulse width is set by R10 and C3 and is given, for a 4528 with C $>0.01\mu F$ and $V_{dd} = 9$ V, by the formula

$$T_w = 0.32RC$$

In this case a pulse width of around 30 seconds is given. When the mono is triggered a high is given at the output which is connected to the reset pin of the 7555, IC7. The 7555 is connected as an astable multivibrator with a low time of

$$T_L = 1.1R11C5$$

and a high time of

$$T_H = 1.1R12C5$$

This gives a period of

$$T = T_H + T_L$$

With the values given, this gives a period of one second.

A high on the reset pin enables the astable. The output of the astable clocks a four-bit counter IC6. If the pushbutton is not pressed again within 30 seconds then the monostable output will go low disabling the 7555. A positive pulse is also applied to the clear inputs of the flip-flops which resets the outputs to low and turns off the LED1 (the mechanism for obtaining this pulse is discussed later). The circuit is now returned to its 'standby' state.

If, however, the pushbutton is pressed a second time within 30 seconds, then a high is clocked through to the output of the second flip-flop. This turns Q2 on which lights LED2. At the same time the output of the NAND gate IC2c goes high which turns off Q1 and thus LED1. The second press also triggers IC4a.

When the high is clocked through to the output of the flip-flop this triggers a second monostable IC4b which disables the counter for a period given by

$$T_W = 0.32R13C6$$

(about three seconds).

After this time the counter resumes counting up. When a terminal count occurs, i.e: when a four-bit counter reaches 15, a low going pulse occurs on pin 7 of the counter. This is inverted to a high going pulse by IC2d. This drives the NAND gate IC2b which gives a low pulse to trigger a third monostable IC5. This mono gives a pulse which turns on Q3 for a period of 1.5 seconds given by

$$T_W = 0.32\ R14C7$$

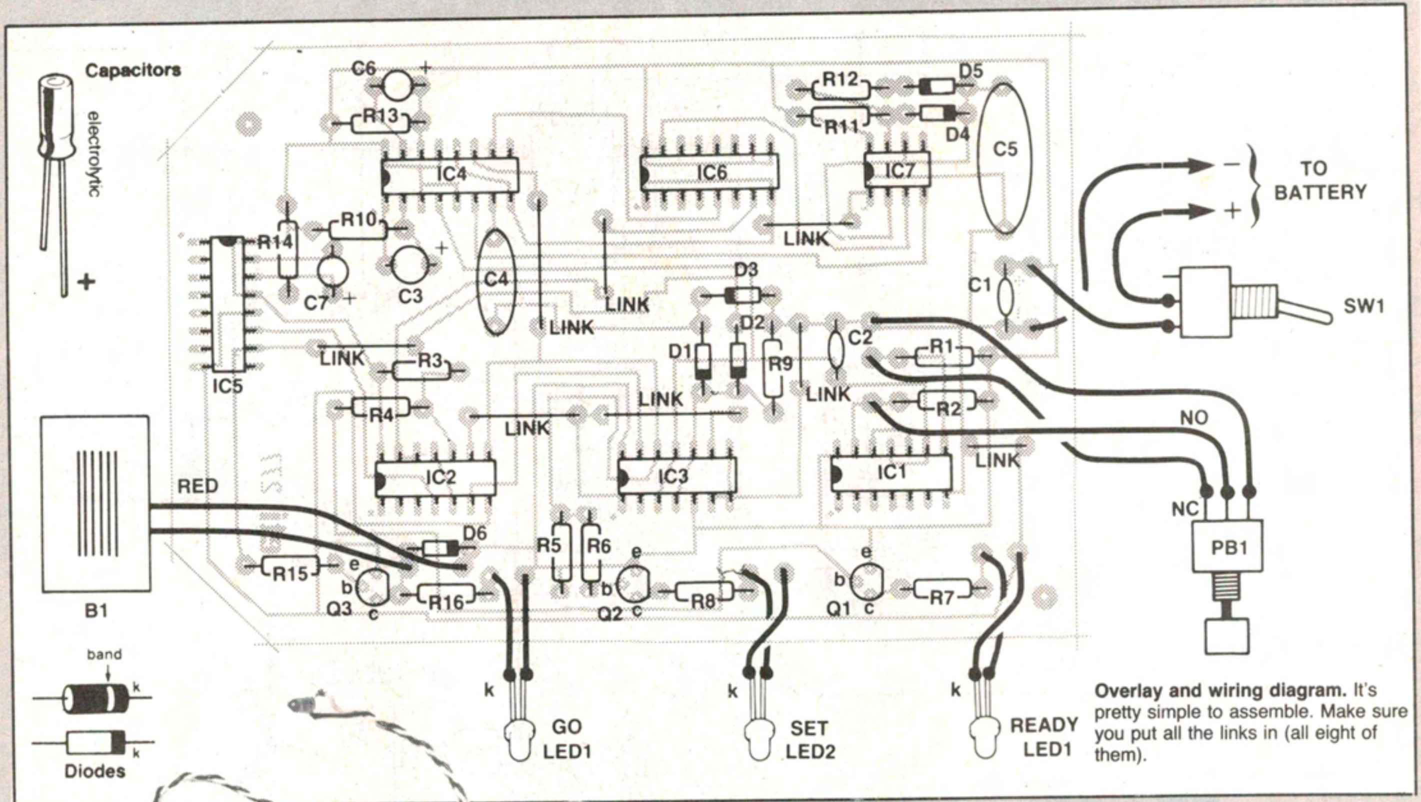
This lights LED3 and turns on the buzzer. The pulse from this third mono also resets the two flip-flops which turns off LED2.

The other input of the NAND gate IC2b is connected to the output of flip-flop two which has the effect of only letting a trigger pulse through to the monostable when LED2 is lit. This prevents the mono from being triggered by a terminal pulse from the counter at some time other than after the second pushbutton press.

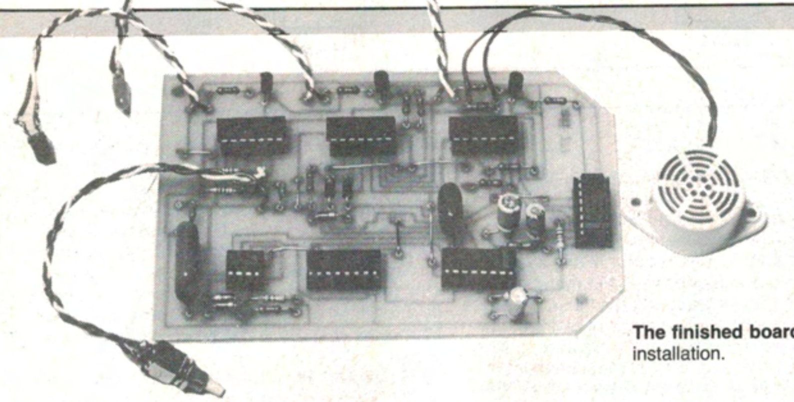
D1, D2 and R9 form a two input OR gate which lets the output from IC4a or IC5b reset the flip-flops. Pin 7 of IC4a is normally high and goes low when the mono is triggered. To prevent this from keeping the flip-flops reset, the pulse is ac coupled to D1 by C4. This provides a short positive pulse on a low to high transition on pin 7 of IC4a. D3 allows C4 to discharge when pin 7 goes low. D6 is put across the buzzer to protect IC5b and Q3 from any back EMF induced because of the inductive load of the buzzer.

C1 filters the supply and C2 filters the clock pulse derived from the pushbutton.

The overall effect is that on the first press of the pushbutton LED1 lights indicating 'ready'. If the button is pressed again within 30 seconds then the 'ready' LED will go out and the 'set' LED will light. Some time later, between three and 18 seconds depending on at which point in the count the button is pressed, the 'set' LED will go out and the 'go' LED and buzzer will turn on for about 1.5 seconds. The circuit will then reset itself. If the button is not pressed a second time within 30 seconds then the circuit will reset itself.



Overlay and wiring diagram. It's pretty simple to assemble. Make sure you put all the links in (all eight of them).



The finished board. Ready for installation.

that the circuit is in the 'ready' mode. This LED is lit after the pushbutton is pressed once. It indicated that the circuit is ready for action.

At this stage the counter is enabled but the gating circuitry on the terminal count output is not yet primed so the mono will not yet fire on a terminal count pulse. The circuit will, in fact, just sit there with the red LED lit and do nothing.

When the pushbutton is pressed a second time and red LED will go out and the yellow LED will light. This indicates that the circuit is in the 'set' mode. At this stage the output monostable will be enabled and will sit and wait for a terminal count pulse to appear. The length of time between the lighting of the yellow LED and the firing of the mono will depend on whereabouts in the count the button was pressed.

If, for instance, the button was pressed when the counter was at five, then the counter would have to count ten more times before giving a terminal count pulse. The counter is clocked once every second, so ten counts is equivalent to ten seconds. If, on the other hand the button is pressed when the counter is at 11, then only four more counts are required.

When the output mono is fired then the yellow LED goes out and the green LED comes on and a buzzer sounds. This means *GO, GO, GO*. At this stage you should be 'going at' whatever you wanted to say 'ready, set, go' for and won't be interested in what the timer is doing which is fine since all the timer will be doing is resetting itself ready for the next round.

If the counter was at 15 when the button was pressed a second time then the green LED and buzzer will come on almost immediately. To overcome this problem a second mono is used to disable the counter for a period of time when the yellow LED is lit.

After the monostable pulse has finished then the counter will be enabled again and pick up counting where it left off. This then gives a minimum time between 'set' and 'go'. With this particular circuit this minimum time is three seconds which should give you enough time to get your hot little hands off the timer and on to your slot car controller or whatever. This time can be lengthened if desired and will be discussed later.

A third monostable is used to shut the whole circuit down after a period of time if it is not re-used. This time is 30 seconds at present but can also be changed if necessary. When the circuit is shut down the clock generator for the counter is disabled and any LED which may be on will be turned off. This is done to save any excessive drain on the battery if you forget to turn the unit off.

Constructional details

The construction is quite straightforward. Begin with the pc board. Check that the tracks on the pc board are all OK and that there are no breaks or shorts. Start by cutting suitable lengths of tinned copper wire to form the eight links on the pc board and solder them in position.

Next locate and solder in all the resistors and capaitors. Take care to get the electrolytics the right way round. The three transistors can be mounted next followed by the six diodes. Make sure you follow the overlay carefully to get the diodes the correct way round.

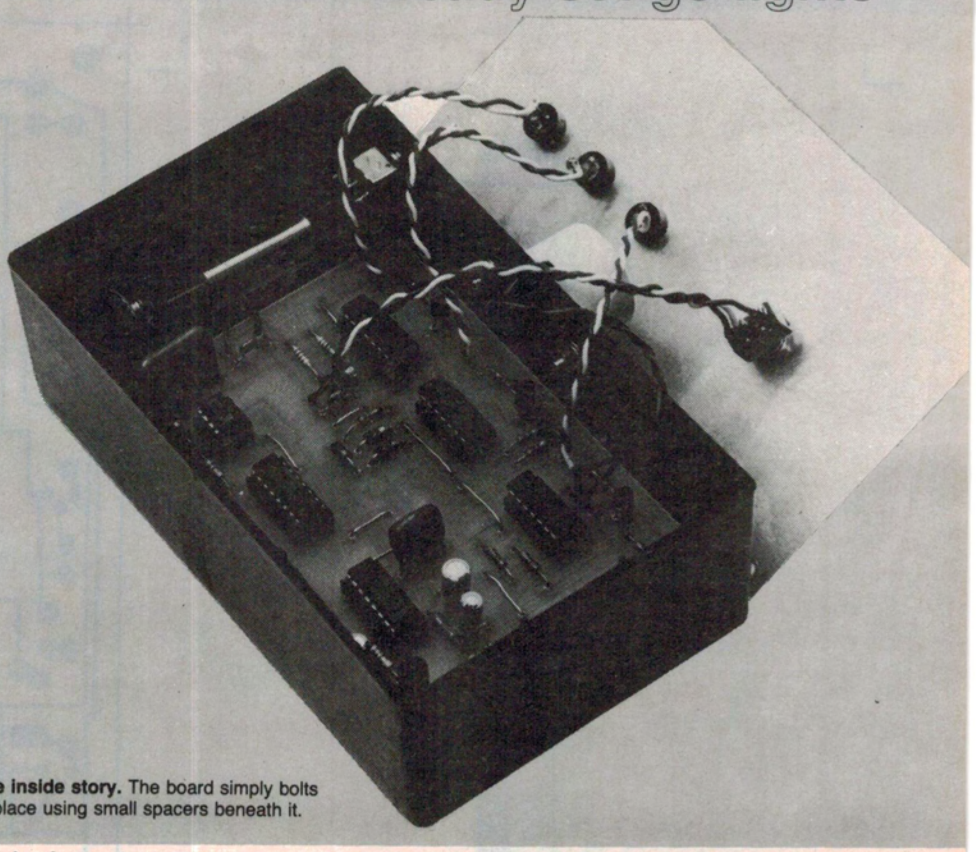
The only remaining components are the ICs. It is recommended that IC sockets be used with CMOS ICs unless you are an experienced solderer. If you're not using

PARTS LIST — ETI-277

Resistors	
R1, R2	82k
R3, R4, R5, R6	8k2
R7, R8, R16	1k
R9, R15	33k
R10	1M8
R11	2M2
R12	10k
R13	1M5
R14	1M
Capacitors	
C1, C2	100n ceramic bypass
C3	47µ/25V electro
C4	100n greencap
C5	470n greencap
C6	6µ8/25V electro
C7	4µ7/electro
Semiconductors	
Q1, Q2	BC558
Q3	BC548
IC1, IC2	4011
IC3	4013
IC4, IC5	4528
IC6	4029
IC7	7555
LED1	Red LED 5 mm
LED2	Yellow LED 5 mm
LED3	Green LED 5 mm
D1, D2, D3, D4, D5, D6	1N4002 or similar
Miscellaneous	
SW1	SPDT toggle
PB1	SPDT momentary action pushbutton switch
B1	9V buzzer

ETI-277 pc board; Scotchcal front panel; 50 x 90 x 150 jiffy box; No. 216 battery clip; three LED mounting bezels; hookup wire, double-sided tape etc.

Price estimate: $20-$25



The inside story. The board simply bolts in place using small spacers beneath it.

sockets then solder in the ICs quickly and avoid excessive heat on the IC pins. It is advisable to solder in all the 'earthy' pins first to minimise the risk of destroying the IC with stray voltages induced by the soldering iron. Make sure, with all the ICS that you get the correct orientation.

The only remaining thing to do is to attach lengths of hook-up wire to the connection points of the LEDs, battery terminal and switches. If you make each wire about 100 mm long then you can trim them later. Note that the negative lead from the battery terminal is connected straight to the pc board.

At this stage you should have a fully constructed pc board with a battery terminal and about a dozen wires dangling from it. The next step is to prepare the box to put it in.

The prototype was housed in a 90x50x150 mm Jiffy box. The pc board has been designed to fit snuggly into this box with room for the battery at one end. Two corners of the pc board are chamfered to allow the board to sit between the screw points in the corners of the box.

Firstly unscrew the face plate of the box and mark out the holes for the LEDs and pushbutton using the front panel artwork as a template. Drill ten holes to 6.5 mm. The Scotchcal label can now be attached to the front plate. The easiest way to do this is to prick holes at the centre marks on the Scotchcal with a pin. This will give you something to line up with.

Peel the backing off the Scotchcal carefully and make sure it doesn't stick to itself. Next, wet the sticky side of the Scotchcal by putting it under a running tap. Do the same to the face plate. The Scotchcal can now be placed on the face plate and the water will allow you to move the Scotchcal into position. When you have the Scotchcal lined up press it firmly down and wipe off the excess water with a dry, soft cloth. Work out all the air bubbles beginning at the centre and working outwards. Leave the front panel aside for a couple of hours to dry thoroughly. When it is dry carefully cut out the holes with a sharp knife or scalpel.

The prototype box had small ledges protruding from the sides which allowed the pc board to sit about 10 mm from the bottom of the box. If your box doesn't have these ridges then mounting holes have been provided on the board so that nuts and spacers can be used. Sit the pc board in the box and make sure there is room for the battery at one end.

Locate the ON/OFF switch on the side of the box so tht it will not foul on the pc board or battery. Once you are happy with the position drill a 6.5 mm hole in the side of the box to take the switch. Also locate the buzzer on the side of the box in a convenient position and drill holes to take the mounting bolts for the buzzer.

A small slot should be filed in the top edge of the box just above the buzzer to allow the leads to enter. If necessary the LED leads can be trimmed at this stage. Don't cut them too short or there won't be enough free play to take the lid off without ripping the pc board out.

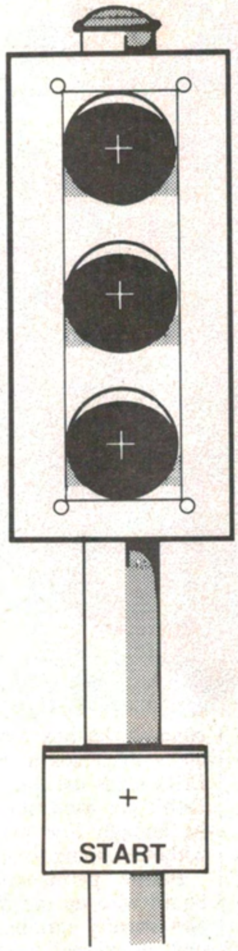
Before permanently mounting the pc board wire up the buzzer ON/OFF switch and battery terminal. The pc board can now be placed in the box. A couple of small blobs of glue will hold it in place (don't use superglue or you'll never be able to get it out again). Mount the ON/-OFF switch and buzzer.

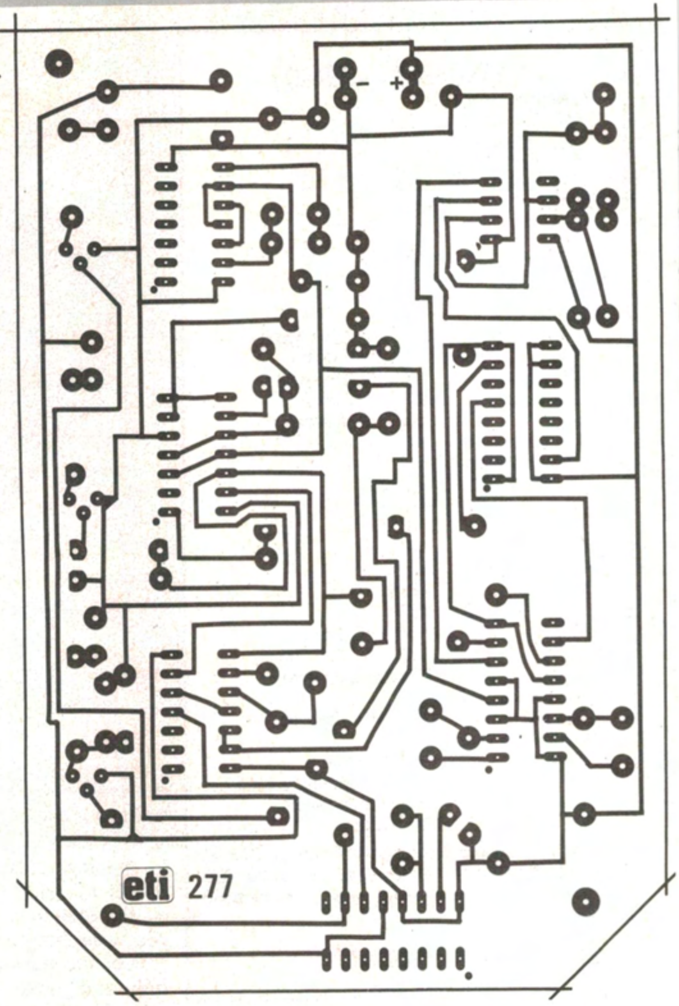
The pushbutton and LEDs can now be attached to the lid (use LED grommets and hacking washers for the LEDs). The only thing left to do is attach a battery and screw on the lid. The battery can be secured to the bottom of the case with a piece of double-sided tape.

Using it

The unit is very easy to use. Simply turn the ON/OFF switch on. The green LED should light and the buzzer will sound for about five seconds. After this all the LEDs should be off. Now all you need to do is press the pushbutton once. The red LED should light. If you do not press the button again the red LED will go out after about 30 seconds.

If you press the button again the yellow LED will light and then, somewhere between three and 18 seconds later, the yellow LED will go out and the green LED and buzzer will turn on for about 1.5 seconds. There is no need to turn the unit off again, unless you are not going to use it ▶





eti 277 READY,SET,GO TIMER

Artwork. Full-size reproductions of the printed circuit board and front panel.

again for a long time, as the unit will automatically go into a standby mode and will draw only tens of microamps from the battery.

A note on time constants

There are three main time constants that may be altered in this circuit. Firstly, the minimum time between 'set' and 'go'. This is set by R13 and C6. The formula for the time delay is:

$$Tw = 0.32 \times R13 \times C6$$

It should be noted, however, that R13 should not be greater than 2M. C6 can be altered to any value to give the desired time.

The second time which can be altered is the period of the 7555 astable. At the moment the counter is being clocked at the rate of one per second. This will give a time of 16 seconds between terminal count pulses. If the period of the clock pulses is decreased to, say, 0.5 seconds then the terminal count pulses will come about every eight seconds. The period is approximately given by:

$$T = 1.1 \times R11 \times C5$$

This is best altered by altering C5. R11 should not be increased past 2M2. If a longer period is required then C5 can be an electrolytic capacitor with the negative pin going to earth.

The third time constant you may wish to alter is that of the automatic reset which puts the circuit into standby when it is not used for a period of time. The monostable that controls this is retriggered on each press of the pushbutton so this time constant must always be set greater than the maximum time between 'set' and 'go'.

At the moment it is set for around 30 seconds. If you increase either of the previous two time constants then you should also increase this one. If the either of the previous time constants are decreased then you don't need to change this one. The formula for this time constant is:

$$Tm = 0.32 \times R10 \times C3$$

Once again R10 should be kept below 2 M but C3 can be as high as practical.

Epilogue

After implementing the Ready, Set, Go timer to start the Tug 'o' War games, editorial/lab staff relations improved remarkably. It should be noted that the lab staff still trounced the opposition regardless of the starting method. This was put down to the exhaustive research in pinball parlours that we were forced to do in order to gain a greater understanding of the state of-the-art technology used in today's sophisticated arcade games.

Jennie has made a remarkable recovery but has been advised that a long sea voyage could do her the world of good, and so leaves on a slow boat to China in the near future. We lose a lot of good assistant editors that way. ●