TELEPHONE C

Have you ever opened your phone bill and thought "Woh! No Way! I can't have made so many calls". This call logger will prove the point – who you called and for how long!

Design by Frank Crivelli – Article by Peter Crowcroft*

Most phone bills itemise long distance calls and give details of the number dialled. But local calls are just totalled up – you have no way of knowing who those calls were made to, or when they were made, or for how long.

What about small businesses that operate from home? They can usually claim a percentage of phone calls as a deduction for income tax purposes.

With the call logger, ALL outgoing call details are recorded and it is simply a matter of sorting them to determine how many were business calls.

You don't have to do a thing: all the data is recorded automatically. And all phones that use DTMF tone dialing are supported.

The logger records the start and stop time of outgoing calls along with the number dialled (plus any other digits pressed during the call – for example, BPay and credit card number entry). It operates "stand alone" – there is no need for any connection to a PC until you want to use the data.

When you want to analyse the data, it can be easily imported into Microsoft Excel. Various Excel functions can then be used to analyse and sort the data and produce formatted printouts.

Data is stored in non-volatile EEPROM memory, so there is no loss of data in the event of a power failure. The kit is supplied with 16K of memory and is expandable in 16K blocks up to 64K total. The circuit "auto detects" the amount of available memory – just plug it in and go.

The number of calls that can be recorded depends on the amount of memory installed and the number dialled. Each call requires the following minimum number of bytes:

- Start date & time   -   6 bytes
- Stop data & time   -   6 bytes
- Record terminator   -   1 byte

Each digit dialled requires one byte. So, each call occupies 13 bytes of memory PLUS one byte for each digit dialled. Allowing for 11 bytes of sys-

PLEASE NOTE:

This is NOT an Austel-approved device. Anyone building this kit should make appropriate enquiries to ensure they are not contravening telecommunications legislation. Also, the use of telephone number and duration recording devices may contravene privacy legislation in some areas.
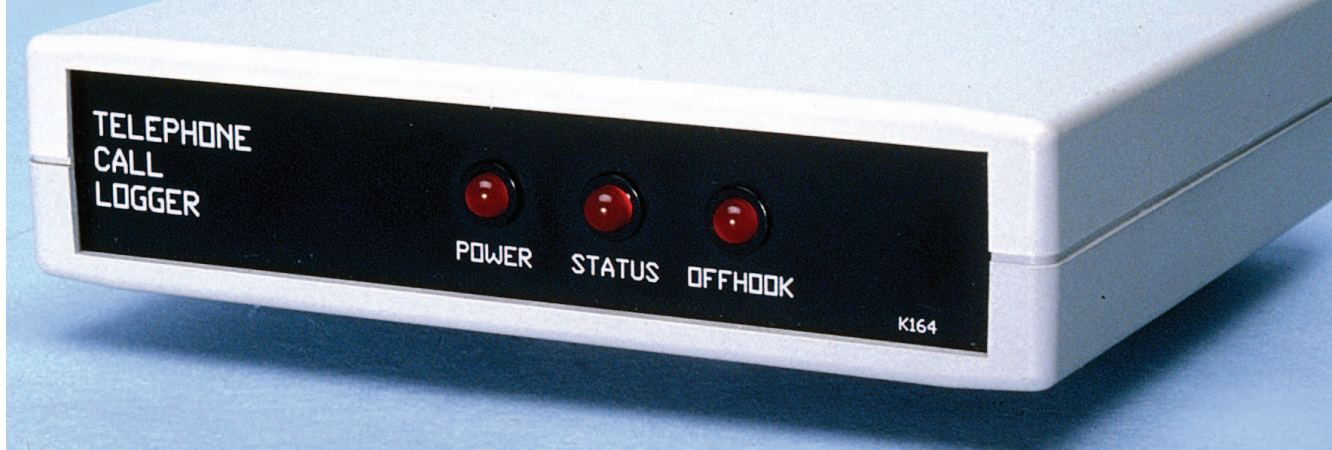
tem overhead (password storage), this leaves 711 x 10-digit phone numbers that can be stored in the basic design with 16K of memory installed [(16384-11)/23] bytes.)

If your phone bill records more local calls than this (aah, so you have teenage daughters too?), simply install extra memory.

With the full 64K memory installed this is 2848x10-digit phone numbers. Ten digits is just taken as an example. If you are in a country where only 7 or 8 digits are used per call then you will get more calls stored. If you pay your bills by credit card then about 50 to 55 bytes will be used per call.

Access to all logger functions, including downloading of call data, is password protected.

Three LEDs are provided on the front panel to indicate power on, offhook (call in progress) and memory status. The memory status LED flashes when the memory is 75% full and is permanently on when 100% full.

As a bonus, call data is also output in "real time" to the serial port as well as being recorded in memory. This is useful when immediate call analysis and/or external data logging are required. This "real time" output is unaffected by the memory status – even if the memory is full the call data is still output.

You can attach this serial output to the parallel port of a printer by going through a serial to parallel converter. These interface boards are readily available commercially.

Phone line connection

Two RJ12 (USA-type) telephone sockets are available on the rear panel, marked "LINE" and "PHONE". The telephone line connects to the "LINE" connector and an optional telephone can be connected to the "PHONE" connector. This allows a telephone to be connected and used even when the logger is put in its place. Telephone line cords are not supplied with the logger kit but are very commonly available.

The project is constructed on a dou-

The logger is housed in a small plastic case and is fully self-contained. It does not need to be connected to a PC until you want to download the data which is stored in EEPROMs.

ble-sided, plated-through hole PC board and fits in a plastic case measuring 130(W) x 100(D) x 30(H)mm. Screen-printed front and rear panels are supplied.

The kit requires a 9-to-12V DC power supply. A 12VDC plug-pack rated at 300mA is suitable.

Circuit description

The kit is controlled by a preprogrammed 89C2051 microcontroller from Atmel (U3). This was chosen because it has the required number of I/O pins and an inbuilt serial port. It has 2K bytes of flash memory which is pre-programmed with the control firmware.

U2, a Maxim MAX666, provides two functions:

1: 5V regulator (via pin 2) to power the rest of the circuitry. The diode bridge B1 provides polarity protection. The MAX666 maximum input voltage is 16.5 volts. Some 12V (nominal) plugpacks actually deliver a lot higher voltage at low output currents, up to 17V in some cases. Diodes D1 & 2, along with B1, allow voltages up to 19V to be used to power the kit.

2: Low Battery Detect, intended for use with battery powered equipment. In this application it is used to detect when the telephone line is 'looped' (in use). The DC voltage of a telephone line is nominally 48V. This drops to around 12V or less when looped. The chip detects this voltage drop and pulls pin 7 low. Diode bridge B2 provides polarity protection on the tele-phone line.

U1 (MT8870CE) is a DTMF decoder. It connects to the telephone line via resistors R1 and R2 and capacitors C1 and C2. When a DTMF tone is de-
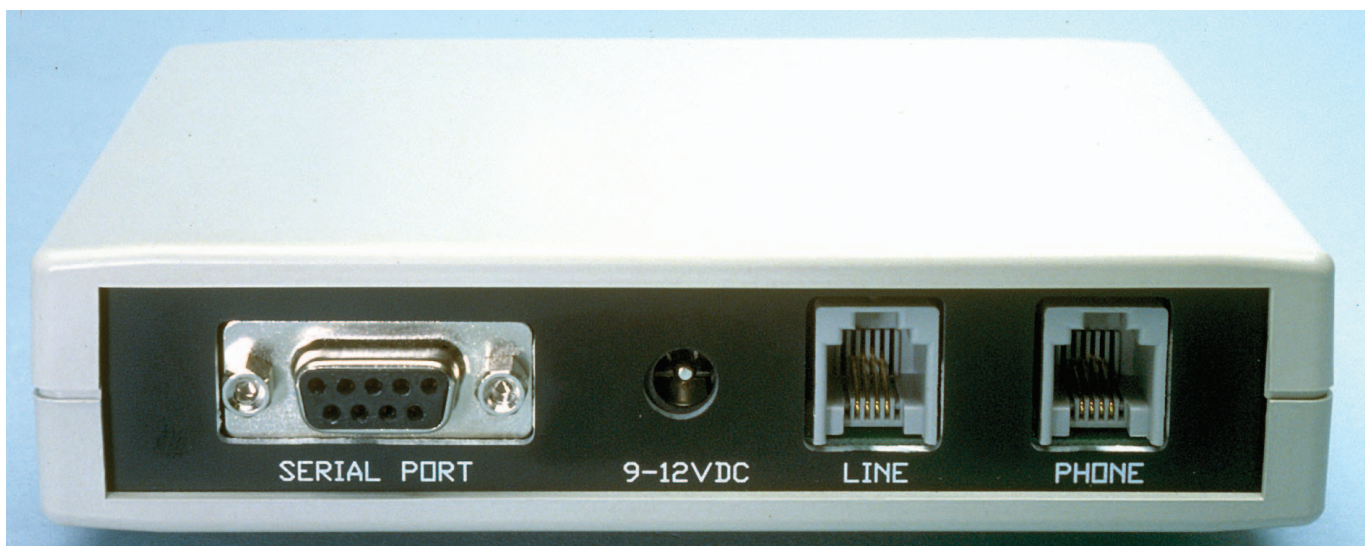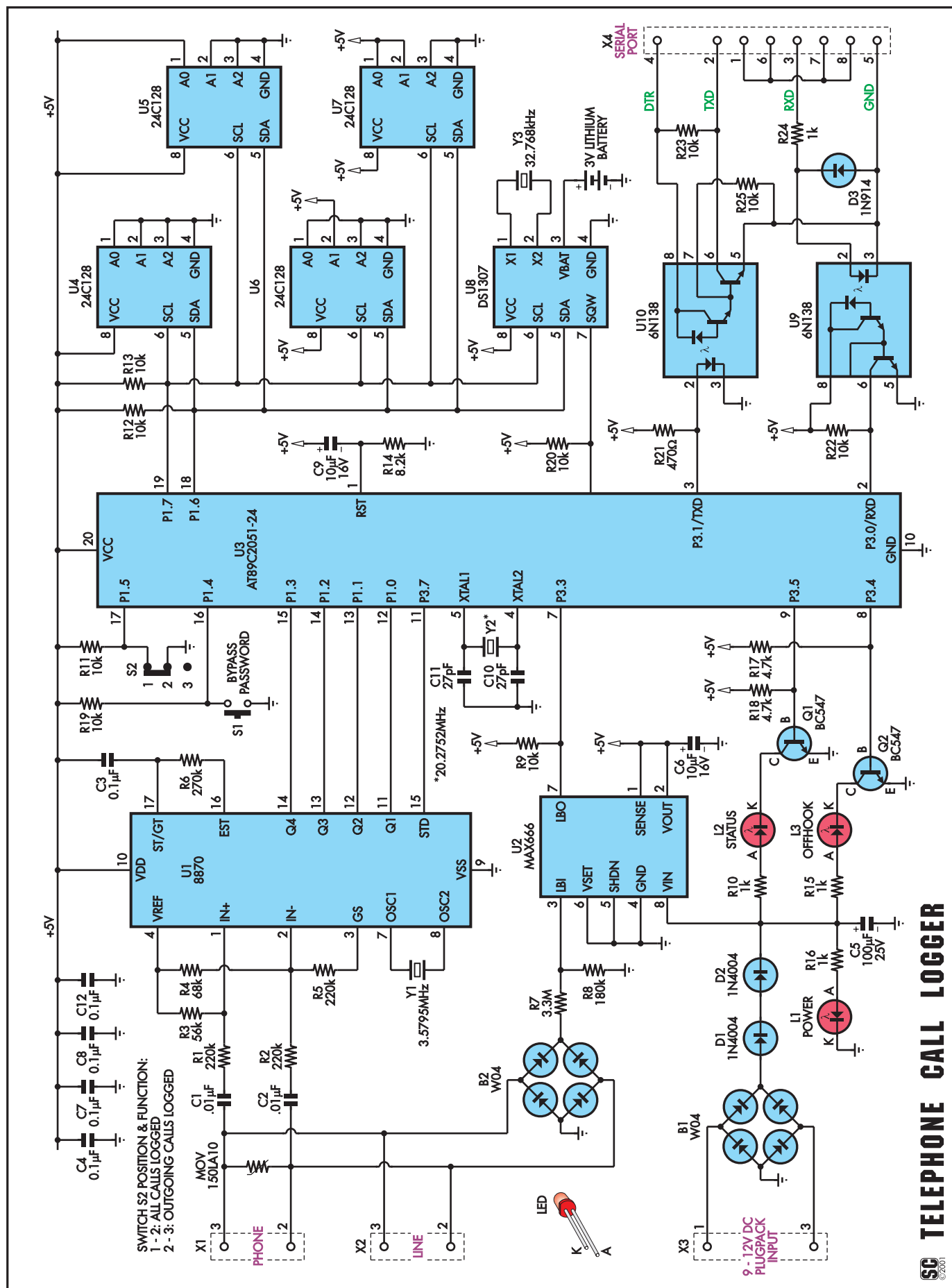

The rear of the case has four sockets. The telephone line connects to the "line" socket and you can plug a phone into the other RJ12 socket.

TELEPHONE CALL LOGGER

SWITCH S2 POSITION & FUNCTION:
1 - 2: ALL CALLS LOGGED
2 - 3: OUTGOING CALLS LOGGED

tected, pin 15 goes high and the 4-bit digit data is presented on pins 11-14. The metal oxide varistor (MOV) protects the kit from damage by any short duration high-voltage spikes on the telephone line.

All clock/calendar functions are provided by U8 (DS1307), a Serial Real Time Clock from Dallas Semiconductor. Using a 2-wire I^{2}C serial interface to communicate with the microcontroller, it features real time clock counts for seconds, minutes, hours, date of the month, month, day of the week and year with leap year compensation valid until 2100. It has a square wave output (pin 7), programmed to give a 1-second signal. This is used by the microcontroller firmware to flash the memory status LED. The chip also features automatic power-fail detect and switch circuitry. The 3V lithium battery provides back-up power during a power failure.

Non-volatile memory storage is provided by up to four AT24C128 EEPROMs. These also use a 2-wire I^{2}C serial interface like the clock chip. The EEPROMs are 128K bit types organised as 16K x 8. At least one EEPROM must be present. The rest are optional and can be inserted in sequence by the user as required.

Two switches are provided. S1 is a pushbutton type and is used for 'password bypass'. S2 is a slide switch and is used to set the type of calls recorded. More on these later.

The serial interface is a bit unusual. It is not the regular type offering RS232 type signal levels. Instead we have used two high-speed optoisolators, U9 and U10.

These were necessary to provide complete electrical isolation between any PC connected to the serial port and the telephone line.

An RS232 interface uses a common 'ground' signal between each end of the interface. At the PC end this 'ground' signal is connected to mains earth. This mains earth would then be connected to the telephone line via on-board circuitry in the data logger. Since both sides of a telephone line are normally 'floating' above earth, earthing one side of the line would cause a fault condition at the telephone exchange and the line would stop working properly.

The solution is to use an opto-isolated serial port for connection to a PC. Otherwise the kit would have to
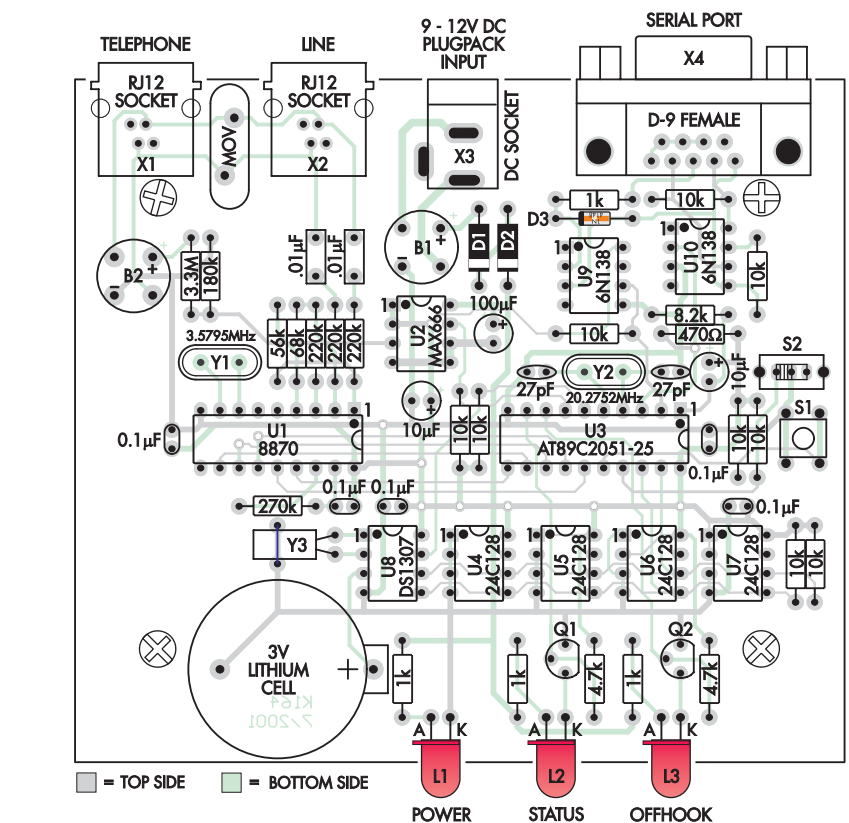
The PC board for the Telephone Call Logger is double-sided – make sure you mount the components from the top (this view looks down on the top side)

be disconnected from the phone line before connecting a PC to it. It would also mean that a PC could not be permanently connected to capture the 'real time' output from the logger.

The receive side of the serial interface (U9) is simple. The RS232 voltage levels from the PC drive the LED in U9 via R24. Diode D3 protects the LED from reverse currents when the RS232 level goes negative.

The transmit side of the serial interface (U10) requires a positive voltage to bias the output transistor.

To maintain electrical isolation, this voltage is provided by the PC side of the interface via the DTR line. The 'active' state of DTR is +12V (nominal). R23 is the collector load resistor for U10's open-collector output, while R25 provides bias for the Darlington output stage, reducing the turn-off time of the output.

What about the power supply you say? Yes, it also must be isolated from mains earth. This is not really a problem when using a plugpack. Plugpacks normally use a 2-pin connection to mains (no earth pin). They also use a transformer to convert the mains voltage, which provides further electrical isolation from earth.

Construction

The PC board has the component overlay printed on it. This is the same as the diagram above. Use the component overlay and the photographs to help with the assembly. Note that the LEDs are left until last.

The following is a suggested order of assembly:

1. Resistors and diodes. Put in the four 1% metal film resistors (R3-6) around the DTMF decoder first.

2. IC sockets

3. Crystals and capacitors. Use a resistor lead offcut to hold down crystal Y3 against the PCB.

4. Transistors and the two switches

5. Battery holder and diode bridges

6. DC jack and D9 connector

7. Both telephone connectors

8. Metal Oxide Varistor (MOV). Push this as *far down onto the PC board* as possible. It is tall and will need to be pushed right in so the lid will fit. Do not insert any ICs yet.

The plastic box is in two halves. Take the bottom half and fit the rear panel into the slot provided. Now drop in the PC board and slide it back so that the telephone and D9 connectors protrude through the panel. Fix the PC board into place using the four

self-tapping screws provided.

Now comes assembly of the front panel. Start by inserting the three LED clips into the holes, pushing them in from the front. Insert the LEDS into the clips from the rear. The LEDs should "click" into place.

Turn the LEDs in the clips so that the long lead is to the left when viewed from the front. This means that the flat edge of the LED (short lead, cathode) will be to the right when viewed from the front and will correspond to the flat marked on the overlay. This is very important so make sure it is right.

Now bend the leads down at right angles, about 3mm from the LED body, as shown in the diagram.

Use a pair of long-nose pliers to hold the leads while bending. Cut the leads off to a length of about 9mm from the bend.

Position the front panel on to the case and drop it into place, making sure that the LED leads are inserted into their matching holes on the PC board. Once the panel is in place solder the LED leads *from the top of the PC board.*

Insert the 5V regulator IC, U2 (MAX666), into its socket. Connect a power supply to the DC jack. The power LED on the front panel should light. Measure the DC voltage output of the regulator. The easiest place to do this is to measure across pins 10 (GND) and 20 (+5) of the U3 IC socket. It should read very close to 5V.

If OK, disconnect the power and insert the rest of the ICs. Take care that the ICs are the correct way around and none of the leads are bent under the body of the IC or miss their places in the IC sockets.

The EEPROM (24C128) *must* be inserted in position 'MEM 1' (U4). Any extra EEPROMs must be inserted in order from left to right starting at 'MEM 2' (U5). The DS1307 clock/calendar chip, U8, is located to the left of all the EEPROMs.

The lithium battery is inserted in the battery holder with the positive side up. Note we have a lithium battery because fresh CR2032 batteries are readily available everywhere. You would not want a kit with a flat battery!

Do not fit the plastic case lid just yet. Proceed to the section "Installation and setup".

Serial port parameters

Any communications program used to 'talk' to the logger must configure its serial port to:

9600 baud, 8 data bits, 1 stop bit, no parity bit

No 'handshaking' or 'flow control' is used. The logger ***does*** require that the DTR line is 'active', ie +12V. This is the default for most comms programs anyway.

Installation and setup

The logger needs to be 'set up' before use. You will need a PC running a communications program. Any comms program will do. Windows users can use HyperTerminal, which comes with Windows itself. We use our own **term.exe**. which you may download from:

http://kitsrus.com/zip/term.zip

The zip file includes a detailed explanation of how to set up and use term.exe under Windows 98.

● Connect a PC to the logger via one of the serial ports. Use a 9-way, male-to-female 'straight though' cable.

● Start the comms program and set the serial port parameters as described

---

Parts List - Telephone Call Logger

1 Plastic case, 130(W) x 100(D) x 30(H)mm
1 PC board, 102 x 86mm, coded K164
1 Set of front & rear panels
1 Battery holder
2 RJ12 (6-way 4 pin) phone connectors
1 2.5mm DC Jack
1 female D9 connector PCB mounting
8 8-pin IC sockets
1 18-pin IC socket
1 20-pin IC socket
3 LED clips, 5mm black
1 Pushbutton switch
1 SPDT Slide switch
1 Lithium battery, 3V CR2032 or equivalent (not supplied in kit)
4 Self tapping screws for fixing PC board to case
1 3.579MHz crystal
1 20.2752MHz crystal
1 32.768kHz crystal, tuning fork type
1 metal oxide varistor (MOV)

Semiconductors
2 1N4004 power diodes
1 1N4148 signal diode
2 WO2 bridge rectifier
2 BC547 NPN transistors (Q1, Q2)
3 LEDs, 5mm red
1 MT8870CE DTMF receiver (U1)
1 MAX666 5V regulator (U2)
1 AT89C2051-24PC (Pre-programmed with K164 firmware) (U3)
1-4 AT24C128 Serial EEPROM (1 supplied with kit ñ see text) (U4-U7)
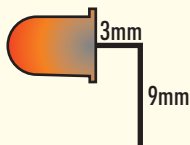1 DS1307 Serial Time Clock (U8)
2 6N138 or 139 optocoupler (U9, U10)

Resistors (0.25W carbon film unless specified)

| 1 | 3.3MΩ | 1 | 270kΩ∗ | 1 | 220kΩ∗ | 2 | 220kΩ | 1 | 180kΩ | 1 | 68kΩ∗ |
| 1 | 56kΩ∗ | 9 | 10kΩ | 1 | 8.2kΩ | 2 | 4.7kΩ | 4 | 1kΩ | 1 | 470Ω |

*1% metal film

Capacitors
1 100µF 25V electrolytic
2 10µF 16V electrolytic
5 0.1µF MKT polyester (104 or 100n)
2 .01µF MKT polyester (103 or 10n)
2 27pF ceramic (27 or 27p)

Here's how to bend the LED leads so they are a neat fit in the front panel and the PC board. Use a pair of needle-nose pliers to grip the lead and bend against.

3mm

9mm

Compare the PC board photograph with the component overlay. The component overlay is also screened onto the top side of the PC board.

above. Make sure the correct serial port is selected.

● Power up the logger. A banner is printed showing the firmware version number and the amount of EEPROM memory installed.

● Hold the "PWD BYPASS" switch (S1) down and press the 'Enter' or 'Return' key on the PC keyboard.

● A menu will be displayed, as follows:

1. Download data
2. Clear data
3. Set date
4. Set time
5. Change password
0. Exit

● Set the date – option 3. The current date will be displayed and you will be prompted to enter the new data. The date is entered as "mm/dd/yy" (including the '/').

Note: The logger does **not** check for valid dates. For example, if you enter "15/66/01" it will accept it.

● Set the time – option 4. The current time will be displayed and you will be prompted to enter the new time. The time is entered in 24-hour format as "hh:mm:ss" (including the ':').

Note: As with the date, the logger does **not** check for valid times. For example, if you enter "29:77:99" it will accept it.

● Set the password – option 5. You

can use any letter, number or punctuation mark but it must be at least four characters long, up to a maximum of eight characters.

Note: Lower and upper case letters are treated as different characters; eg, the lower case letter 'a' is not the same as capital 'A'.

● The last thing is option 2 - Clear data. This **MUST** be done so that the memory is initialised correctly.

● Now exit the menu using option 0. A "READY" prompt is displayed.

Select the type of calls to log, as described next.

Types of calls

The logger has an on-board switch (S2) that is used to select the type of calls to record. A call is defined as any time the phone line is looped (that is, the handset is picked up & the OFFHOOK LED is on.)

Types of calls are defined as:

1. Outgoing Calls Only

These are calls where the handset is picked up and at least one digit is dialled. If no digits are dialled, no data is recorded.

2. All Calls

Data is recorded any time the line is looped, regardless of whether any digits are dialled or not. If no digits are dialled then it was probably an incoming call.

The logger does not have 'caller ID' circuitry and is not able to record the phone number of incoming calls.

Testing

Use a telephone line cord to connect the logger to the telephone line via the 'LINE' connector on the back of the logger and connect a suitable power

No.	Value	4-Band Code (1%)	5-Band Code (1%)
1	3.3MΩ	orange orange green brown	orange orange black yellow brown
1	270kΩ	red purple yellow brown	red purple black orange brown
3	220kΩ	red red yellow brown	red red black orange brown
1	180kΩ	brown grey yellow brown	brown grey black orange brown
1	68kΩ	blue grey orange brown	blue grey black red brown
1	56kΩ	green blue orange brown	green blue black red brown
9	10kΩ	brown black orange brown	brown black black red brown
1	8.2kΩ	grey red red brown	grey red black brown brown
2	4.7kΩ	yellow purple red brown	yellow purple black brown brown
4	1kΩ	brown black red brown	brown black black brown brown
1	470Ω	yellow purple brown brown	yellow purple black black brown

Table 1: RESISTOR COLOUR CODES
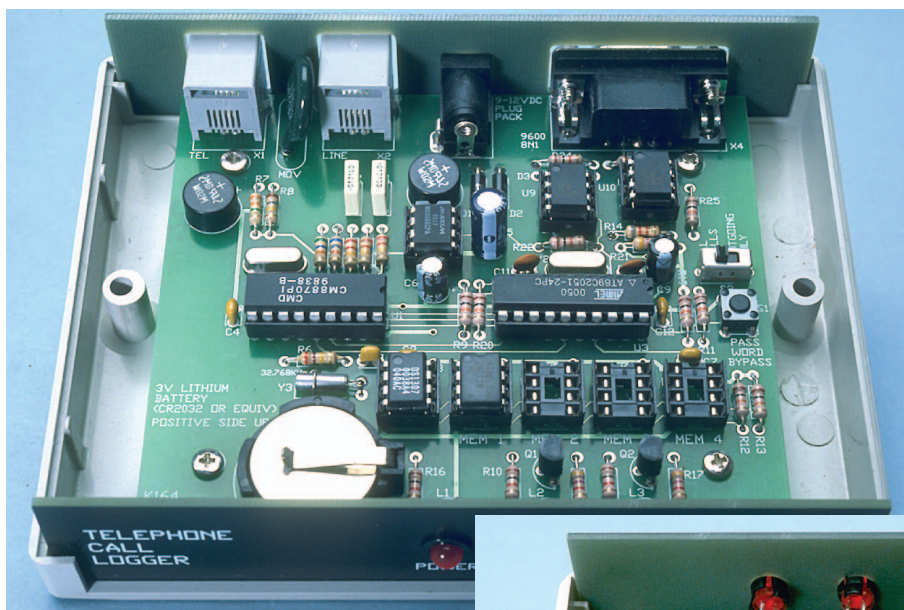
Here's the view inside the case from the front (above) and back (right). Everything mounts on the PC board.

64K of memory. Only one EEPROM (16K) is supplied as standard with the kit.

The memory is user upgradeable by simply 'plugging in' more EEPROMs. Disconnect the power and telephone line and remove the lid. Insert the EEPROM(s) according to the following table.

Memory	Position
16K	MEM 1 (mandatory)
32K	MEM 1, 2
48K	MEM 1, 2, 3
64K	MEM 1, 2, 3, 4

supply (12VDC plugpack).

Once connected to the phone line we can test that it is actually recording calls. Connect a PC to the logger and run the comms program. This will enable us to see the 'real time' output.

Now pick up the phone. The 'OFFHOOK' indicator should light and the current date and time printed on the PC screen. Dial a few digits – they should be displayed as they are pressed. Now hang up. The 'OFFHOOK' indicator should go off and the date and time printed on the screen.

Now we will check if the data was recorded into memory. Press the 'Enter' or 'Return' key on the PC keyboard. You will be prompted for a password. Enter your password. You only have five seconds to start and between each character. If the password is correct a menu is displayed. Choose option 1 – Download data. The call you just made should be displayed there.

Notice that the recorded format is slightly different to the 'real time' output. In the 'real time' output, everything is printed as it happens; ie, the START date/time, followed by the digits dialled, followed by the END date/time. The recorded format has the START date/time followed by the END date/time followed by the digits dialled. This makes the data easier to read if simply making a printed copy.

Operation and use

The logger does not need a PC connected to log calls. A PC is only re-

quired to set or check the onboard clock and download any recorded call data.
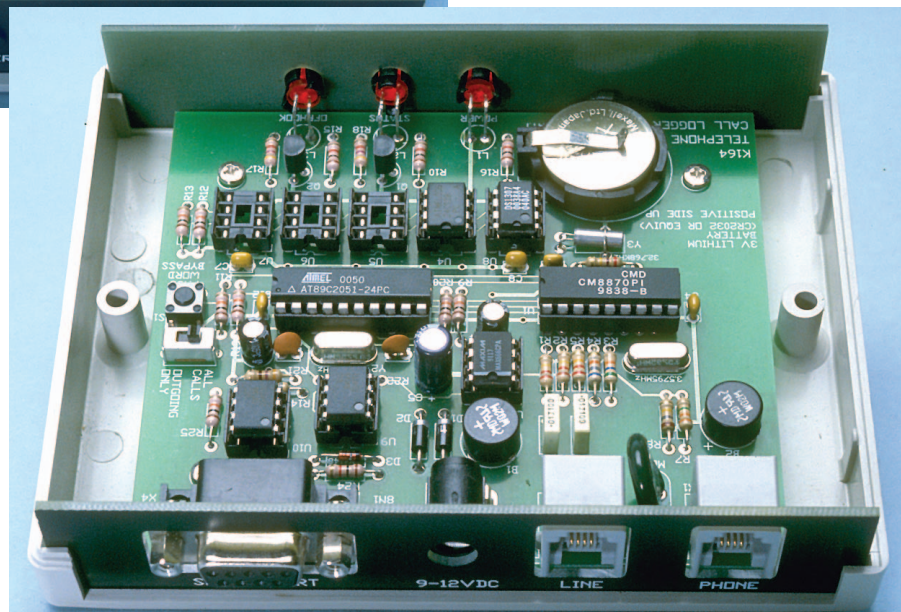
If the data memory becomes full the logger will not log any more calls. In this case connect a PC and download the data (see "***Saving call data to a disk file***"). Once saved, the memory MUST be cleared for the logger to continue logging further calls.

NOTE: the logger will not log any calls while the menu is displayed. ***You must exit the menu (enter '0') to resume normal operation.***

What if I forget my password?

In this case you will need to remove the lid from the logger and hold down the 'PWD BYPASS' switch before you press 'Enter' or 'Return'.

Adding more memory

The logger uses up to 4 x AT24C128 EEPROMs for data storage. Each EEPROM is 16K bytes giving a total of

What do the lights mean?

POWER indicates that power is connected to the logger.

OFFHOOK indicates that the phone line is looped (in use). It will only indicate if the line was looped AFTER the logger was connected. If the line is already in use when the logger is connected then the indicator will be remain off until the NEXT time the line is used.

STATUS indicator has a number of meanings. It is used to indicate the amount of memory left for logging calls. When the memory capacity reaches 75% full the STATUS light starts flashing. When the memory is completely full the light is permanently ON and no more calls will be logged.

If the STATUS light is permanently ON it could indicate battery failure on the clock. If the battery has failed then the clock data may be corrupted or

incorrect. In this case you will need to use the logger's menu to check the date and time. Test the battery and replace if necessary.

If both the STATUS and OFFHOOK lights are ON when the logger is powered up it means that the logger could not detect any EEPROM memory at all. If you connect a PC to the logger and run a comms program you will see the words "MEMORY ERROR" continuously printed. Check that there is an EEPROM in posistion 'MEM 1' or re-seat the chip if one is present.

Download data format

The recorded call data is printed out in the following format:

|óStart date and timeó| |óóEnd date and timeóó| |-digits dialledó| (dd-mon-year hours:mins:secs, dd-mon-year hours:mins:secs, all numbers pressed)

The date is of the form "12-May-01" and the time is in 24-hour format as in "13:30:00" (1.30pm). Each data record (call) is terminated by a "carriage return/line feed" combination.

Saving call data to a disk file

Now all the call logged data is sitting in MEM1 (and MEM2, 3 and 4 if present.) We want to get this data from these IC(s) into our PC. How do we do this?

Answer: we download the data using a 'comms', or communications, program running on a PC that 'captures' the data to a disk file as it is displayed on the screen. All comms programs have a 'data capture' or 'logging' function where everything displayed on the screen is also saved to a disk file. The file is a simple ASCII text file. Windows 9x has its own comms program, HyperTerminal, built in and you can use this. But we use *TERM.EXE*, our own program which can be downloaded from our website http://kitsrus.com/zip/term.zip

The file includes a detailed explanation of how to set up and use *TERM.EXE* under Windows 98.

The following description assumes you are running *TERM.EXE*. (Similar event sequences apply to all comms programs.)

Connect your PC to the logger and start term running by clicking on the *TERM.EXE* icon. After the screen has appeared press 'Enter'. You will be prompted for a password. Enter the password and a menu will be displayed. (The phone must not be in use – offhook LED lit – when you do this.) If you do not enter your password within 5s the software will return to ready mode and will log data. Hit Enter to get the password prompt back.

After successful password entry you will get the Menu. Now turn on 'data capture' by pressing alt-L. You should be prompted for a file name to save to (term.log is the default). Hit Enter. LOG will appear on the bottom line of the display to show logging is on. Then choose option 1 – Download data. Hit Enter. All recorded data will be printed out (in the format described previously).

When finished, turn off 'data capture' by pressing alt-L again. Note that LOG disappears from the bottom line. All the data has now been saved to the disk file 'term.log'. Finally, press '0', press Enter – the logger will exit the menu and return to normal operation. To exit TERM.EXE hit alt-Q. Note logging must be turned off to allow you to exit from TERM.EXE.

Note that you cannot stop the scrolling of the data after you press '1'. The whole data in the EEPROMs is dumped to term.log in one go. You have to go to term.log with a text editor to look at the individual data lines or load it into Excel as described next.
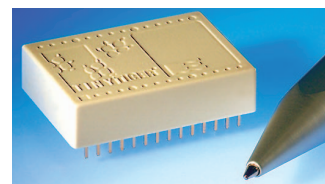
Importing data into Excel

The data in term.log is in a format that can be easily imported into Microsoft Excel. First, edit term.log to remove unwanted data from the start and end of the file. Once done the data can now be imported into Excel.

The following step-by-step guide will show you how to import the file into Microsoft Excel. It has been tested using Excel 97 and 2000.
1. Click on ìFile, Openî
2. In the ëFiles of typeí box select ëAll Files (*.*)í
3. Browse to the saved data file, select it and click ëOpení
4. A ëText Import Wizardí dialog box will appear. Under ëOriginal data typeí select ëDelimitedí. Click ëNextí
5. A second dialog box will appear. Here you can choose which delimiters to use to break up the file into columns. If you want the data and time to be one column then select ëCommaí only. If you want the date and time to be separate columns (recommended) then select both ëCommaí and ëSpaceí as delimiters. A preview will show you
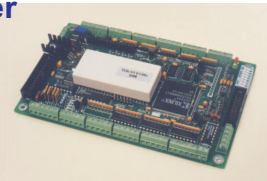
how choosing each delimiter affects the file. Click ëNextí.

6. A third and last dialog box appears. Here you can choose how Excel interprets each column of data. Select each column in turn and set the ëColumn data formatí to ëTextí. Click ëFinishí when done.

Calculating call duration

Once you have imported the data we can use Excel's date and time functions to calculate the length of each call. We can then sort the file into order starting with the longest duration call first.

The data should have been imported in five columns labelled as follows:

Column	Data
A	Start date
B	Start time
C	End date
D	End time
E	Digits dialled

The data should start in row 1. We will now create a column of data that is the duration of each call.

1. Click on cell ëF1í
2. Type in the following formula:
    =DATEVALUE(C1)
    +TIMEVALUE(D1)
    -DATEVALUE(A1)
    -TIMEVALUE(B1)

This tells Excel to subtract the start date (A1) and time (B1) from the end date (C1) and time (D1).

3. Click on cell ëF1í again
4. From the menu select ìFormat Cellsî
5. The ìFormat Cellsî dialog box will appear. In the ëNumberí tab under ëCategoryí list select ëTimeí. Under ëTypeí select í37:30:55í. The call duration is now shown in ìhours:minutes:secondsî and can span multiple days.

We now want to copy this formula and formatting to the rest of the cells in this column down to the last data entry.

6. Click on cell F1 again. This cell contains the formula and formatting information we need.
7. Drag the fill handle over the cells to be filled.

The fill handle is the small black square in the corner of the selected cell (F1). When you point to the fill handle, the mouse pointer changes to a black cross.

The cells should now contain the duration of each call.

Note: some calls may span across midnight from one day to the next. In this case the end time will be earlier than the start time. Using the Excel DATEVALUE and TIMEVALUE functions allows the correct time duration to be calculated.

Sorting the data

The data can be sorted in any order using any column as required. The following example sorts the data according to call duration (as calculated previously).

1. Click on any data cell
2. From the menu select ìData Sortî
3. The ëSortí dialog box will appear. In the ëSort byí box select Column F (the call duration column) and ëDescendingí
4. Click ëOKí

The call data is now sorted by call duration with the longest call first.

Of course you can sort the data in any order you like. Using the 'digits dialled' column will sort the data by phone numbers dialled, with all similar numbers grouped together.

Sorting by "start time" will group all numbers dialled at a certain time of day.

If it does not work . . .

Poor soldering ("dry joints") is the most common reason for the circuit not working. Check all soldered joints carefully under a good light. Re-solder any that look suspicious. And then check:

• Are all the components in their correct position on the PC board?

• Are the electrolytic capacitors the right way round? What about the diodes and diode bridges?

• Are the ICs the right way around?

• Are any IC leads bent up under the IC body (ie, not in the sockets)?

• Is the regulator output = 5V?

• Is it connected to the telephone line?

• Is it connected to the right serial port on your PC?

• Are you using a straight through serial cable?

• Is the correct serial port selected in your comms program?

• Is the serial port configured correctly? Is DTR active (+12V)?

• Is an EEPROM inserted in position "MEM 1" (U4)?   **SC**