Control equipment from anywhere, anytime, using SMS and an old Nokia mobile phone! – By Peter Smith

SMS Controller Pt.1

Would you like to be immediately informed when your burglar alarm is activated, as well as which sectors were tripped? What about if you could reset the alarm or even isolate one or more sectors? Well, this is just one of a huge number of possible applications for our new SMS Controller. Other applications include switching home appliances, rebooting a server or locating your car in a car park.

USING THE CONVENIENCE of SMS, this project lets you remotely control equipment by sending plain text messages, such as 'pump on', 'aircon off', 'reset' or 'blast horn' – all of which can be pre-programmed into the controller and easily remembered later. It can control up to eight external devices and report the condition of up to four digital inputs.

Short Message Service (SMS) is defined as a text-based service that enables up to 160 characters to be sent from one mobile phone to another. In a similar vein to email, messages are stored and forwarded at an SMS centre, allowing messages to be retrieved later if you are not immediately available to receive them. Unlike voice calls, SMS messages travel over the mobile network's low-speed control channel. 'Texting', as it's also known, is a fast and convenient way of communicating.

Users have been quick to make use of this technology, with millions of handsets currently in use. As new models with 'must have' features hit the market, older models become virtually worthless and if not recycled, end up in landfill.

With this in mind, we've designed this project to work with several popular (but now outdated) Nokia models. Chances are, you'll already have one of these on the shelf. If not, secondhand units are readily available for a song.

Nokia rebirth

While a number of models would have been suitable for this project, the

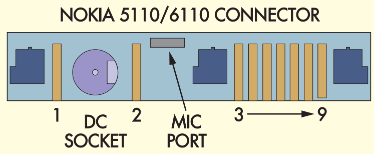


Fig.1: the interface connector on 5110 & 6110 models is located at the bottom of the phone. Here's the pin numbering schematic.

Table 1: Connector Pinouts For The Nokia 5110 & 6110 Models		
Pin No.	**Name**	**Function**
1	V_{IN}	Charger input voltage
2	CHRG_CTRL	Charger control (PWM) signal
3	XMIC	External microphone input
4	SGND	Signal ground
5	XEAR	External earphone output
6	MBUS	MBUS serial receive/transmit
7	FBUS_RX	FBUS serial receive
8	FBUS_TX	FBUS serial transmit
9	L_GND	Charger/logic ground
DC Jack		Charger input (centre positive)

Nokia 3210, 3310, 5110 and 6110 were obvious choices, as they all include a common serial data interface necessary for remote control.

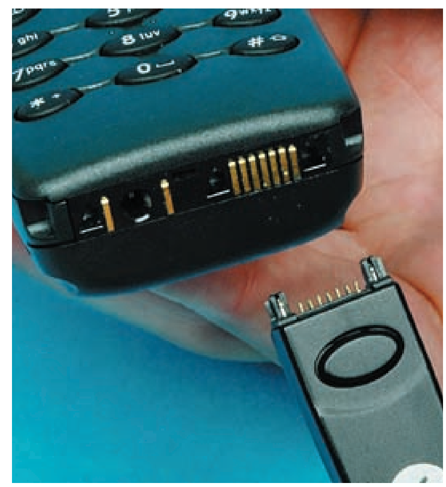
Of these four models, the 5110 & 6110 are preferred for two important reasons. First, Nokia specifically designed the serial interface on these models for user access. In fact, they marketed accessories such as car kits and PC-based software that makes use of the interface. The interface connectors are therefore reliable and easily accessible.

Second, both models include the functions necessary for the SMS Controller to monitor battery level, as well as 'push' the power button should power be lost for any reason.

By contrast, the 3210 & 3310 interface connectors are hidden beneath the rear covers, and in the case of the 3310, beneath the battery! In addition, they lack the battery monitoring and remote power-up functions. It's therefore necessary to manually push the power button if a battery runs flat. Despite these limitations, both models operate satisfactorily with this project. If you don't already have a suitable model, you can often pick one up on eBay for under £10. Look for a unit with a good battery; this will save you money later, as a functioning battery is mandatory, even when connected to a DC power source.

You'll also need a data cable for the phone to controller link. Nokia no longer offers cables for these older phones but after-market equivalents are readily available on the Internet. Alternatively, ask your local mobile phone dealer for advice on suitable suppliers.

Note that some vendors offer cables designed specifically for updating, or 'flashing', phone memory. Some of these will not work with this project! When in doubt, look for a cable that works with 'LogoManager' or 'Oxygen Phone Manager'. Both these PC



This close-up view shows the interface connector on the Nokia 6110, together with the matching plug from the data cable. The DC jack and the adjacent charger input and control signal pins (pins 1 & 2) are to the left.

System Limitations & Cautions

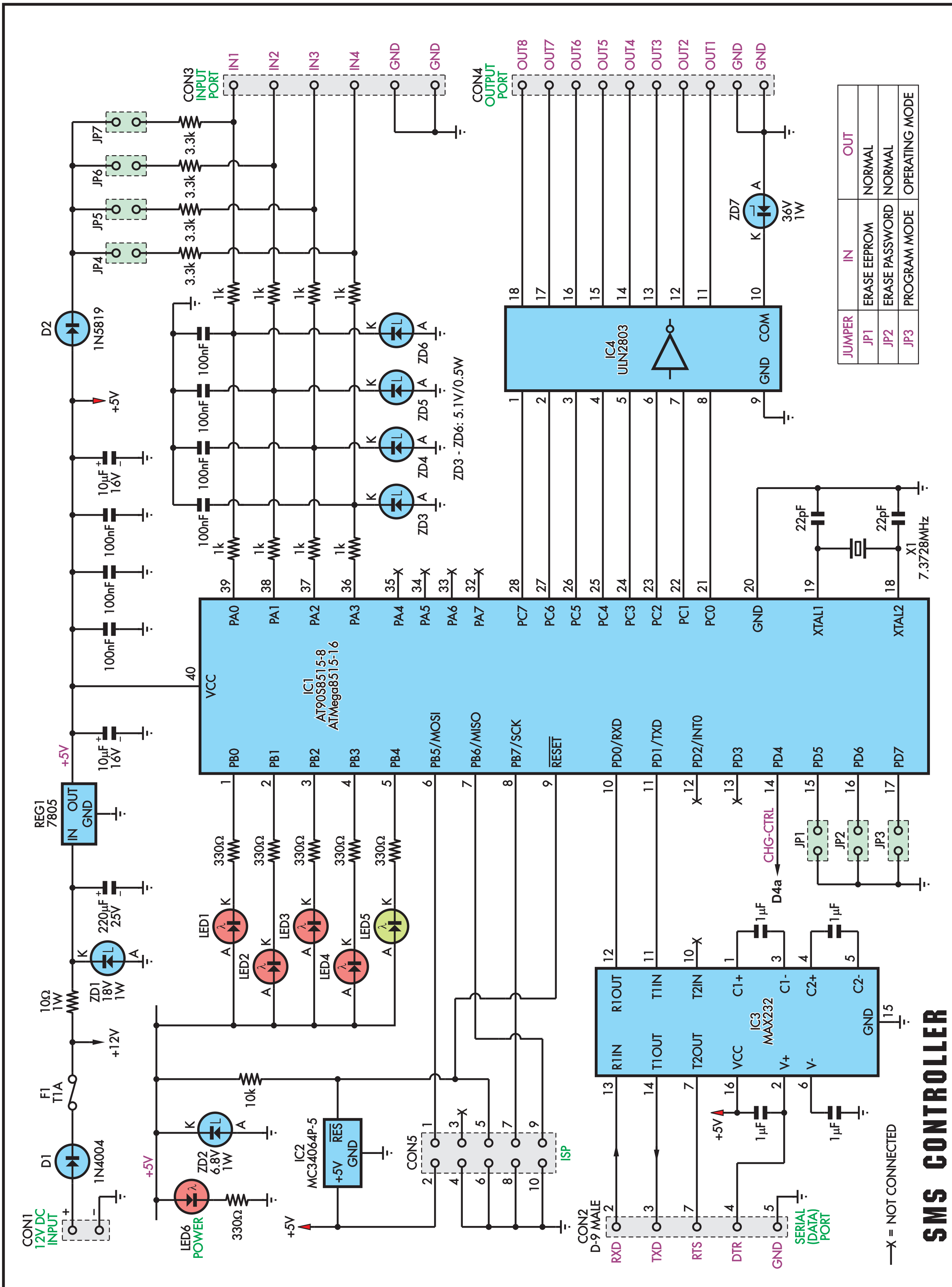
Before building this project, you should first make sure that it suits your intended application. Note that this is not a *real-time* control system. The time taken for a message to be sent by the controller can vary from anywhere between a few seconds to minutes, depending on network load.

This means that rapidly changing inputs will go undetected. Effectively, you will be left not knowing what the real state of the input port is, despite having received a host of state-change messages. In other words, the inputs should only be used to sense signals that change infrequently over time. Alarm signals are a typical example, as they're expected to change only during exceptional conditions.

A second pitfall has to do with SMS costs. You *must* use a pre-paid mobile phone account. A malfunctioning system could cost you a fortune on an open-ended plan. In theory, if the controller were to send messages as fast as the network would allow, more than 17,000 messages could be sent in one day alone. This would really be a disaster!

We therefore strongly recommend that a pre-paid account be set up for the controller-connected phone. This ensures that if something goes wrong, you already know how much it's going to cost you.

Finally, do not use the phone connected to the controller to program or test the system by sending messages to yourself. Doing so will confuse the controller, resulting in messages echoing backwards and forwards until your account balance is empty!



JUMPER	IN	OUT
JP1	ERASE EEPROM	NORMAL
JP2	ERASE PASSWORD	NORMAL
JP3	PROGRAM MODE	OPERATING MODE

Fig.2: the circuit diagram for the main part of the controller. A 40-pin microcontroller (IC1) handles almost everything, including communications with the mobile phone and control of the input and output ports.

Parts List – SMS Controller

1 PC board, code 609 available from the *EPE PCB Service*, size 130mm x 85mm
4 2-way 5mm terminal blocks (CON1, CON4, CON6)
4 3-way 5mm terminal blocks (CON3, CON4)
1 9-way 90° PC-mount male 'D' connector (CON2)
1 10-way 2.54mm DIL shrouded header (CON5)
1 8-way 2.54mm DIL header (JP4 - JP7)
1 6-way 2.54mm DIL header (JP1 - JP3)
7 jumper shunts
1 40-pin IC socket
1 18-pin IC socket
1 16-pin IC socket
1 220µH ferrite choke (L1)
2 M205 PC-mount fuse clips
1 M205 1A slow-blow fuse
4 M3 x 10mm tapped spacers
5 M3 x 6mm pan head screws
1 M3 x 6mm nut & washer
Nokia mobile phone (see text)
Serial (data) cable to suit phone (see text)
DC power cable to suit phone (see text)

Semiconductors
1 AT90S8515-8 or ATMega8515-16 microcontroller (40 pin) (IC1), programmed with SMS.HEX
1 MC34064P-5 under-voltage sensor (IC2)
1 MAX232 RS232 receiver/driver (IC3)
1 ULN2803 Darlington transistor array (IC4)
1 MC34063 switching regulator (IC5)
1 7.3728MHz crystal, HC49 package (X1)
1 1N4004 diode (D1)
2 1N5819 Schottky diodes (D2, D3)
1 1N4148 diode (D4)
1 1N4746 18V 1W Zener diode (ZD1)
1 1N4736 6.8V 1W Zener diode (ZD2)
4 1N751 5.1V 0.5W Zener diodes (ZD3 – ZD6)
1 1N4753 36V 1W Zener diode (ZD7)
5 3mm red LEDs (LED1 - LED4, LED6)
1 3mm green LED (LED5)

Capacitors
1 220µF 50V PC electrolytic
2 220µF 25V PC electrolytic
2 10µF 16V tag tantalum
4 1µF 50V monolithic ceramic
9 100nF 50V monolithic ceramic
1 1nF 50V ceramic disc
2 22pF 50V ceramic disc

Resistors (0.25W 1%)

1 22kΩ	8 1kΩ
1 10kΩ	6 330Ω
4 3.3kΩ	2 1.5Ω 5%
2 4.7kΩ	1 10Ω 1W 5%
1 10Ω 5W 5% (for testing)	

software products communicate with the phone in a similar manner to this project.

Phone power

The controller includes an on-board current-limited power supply for charging the phone's battery. The original plugpack charger (ACP-7A) cannot be used, as it provides no mechanism for disconnecting power once the battery is sufficiently charged.

To connect the controller's power supply output to the phone's DC input, a simple two-wire cable with a standard 1.1mm (3.5mm OD) DC plug on one end is required. You can either make one yourself, or scrounge a ready-made cable from an old in-car charger. All you need to do is disconnect the cigarette lighter plug end and you have the necessary cable complete with a moulded-in DC plug!

Serial interface

The Nokia phones mentioned earlier incorporate two proprietary serial interfaces known as 'MBUS' and 'FBUS'. MBUS is half-duplex, meaning that it provides just one signal line for both sending and receiving data. Data is exchanged over the MBUS at 9600bps (bits per second). This interface is intended primarily for factory test and adjustment, so we won't be using it here.

FBUS, on the other hand, provides separate send and receive lines and operates at the much higher speed of 115.2kbps. Nokia designed FBUS for connection to external accessories, such as their PC Data Suite. However, not all models work with this particular software. Nevertheless, the FBUS interface is present on all these models and ready to do duty in this project.

Note: although earlier model phones also include an FBUS interface, the protocol used is different to that used on the models mentioned here. This project uses FBUS 'version 2' protocol, which according to one source is supported only on the following models: 6110, 6130, 6150, 6190, 5110, 5130, 5190, 3210, 3310, 3330, 3360, 3390 & 3410. However, we've only tested this project with the 6110, 5110, 3210 & 3310 and therefore cannot guarantee operation with other models!

The physical location of the interface pins varies according to the model. In addition, some models provide extra contacts for hands-free adapters and chargers.

Fig.1 and Table 1 show the connector layout and pin assignments for the 5110 and 6110 models. This information is shown for interest only, as the data cable includes all the electronics necessary to interface these signals to a standard PC's serial port. We've designed the controller so that the cable plugs directly into the on-board 9-pin 'D' connector – no PC is required!

Circuit basics

For convenience, we've divided the circuit diagram for the controller into two sections. The main circuit appears in Fig.2, while the phone power supply is shown in Fig.3.

Looking first at Fig.2, you'll note that an Atmel microcontroller (IC1) dominates the circuit, with just a handful of external interface components and a 5V power supply. As first glance, it may seem odd that we've selected a 40-pin micro for the job, as quite a few pins are unused. Wouldn't a

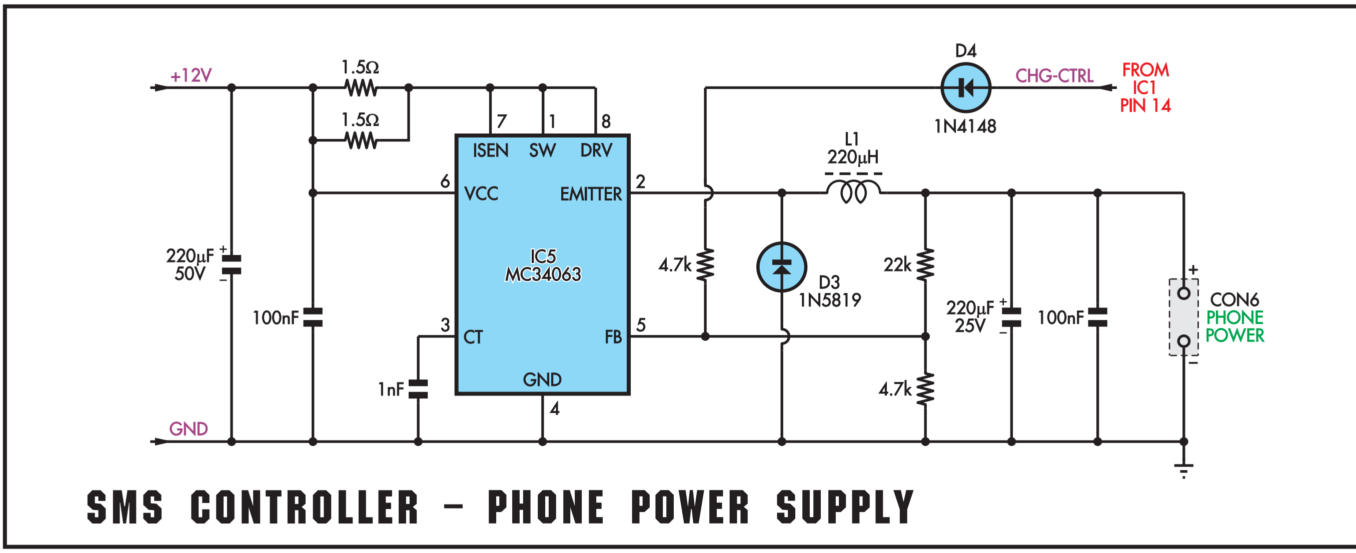


Fig.3: the on-board power supply for the phone is based on a common switchmode regulator (IC5).

smaller, cheaper device have been sufficient? Well, no, because we needed to make use of the generous code and data memory spaces available in this particular device. The AT90S8515/ATMega8515 includes 8192 bytes of code (FLASH) memory, 512 bytes of RAM and 512 bytes of data (EEPROM) memory.

The micro includes four 8-bit input/output (I/O) ports. Ports 'A' and 'C' are used for the external interface, which we'll come back to shortly. Port 'B' drives the five status LEDs (LED1 - LED5) and is also used for in-system programming (ISP) via CON5 – see the panels entitled 'LED Indicators' and 'Microcontroller Programming' in Pt.2 next month.

The upper three bits of Port 'D' (PD5-PD7) are used to read the state of jumpers JP1-JP3. The lower two bits

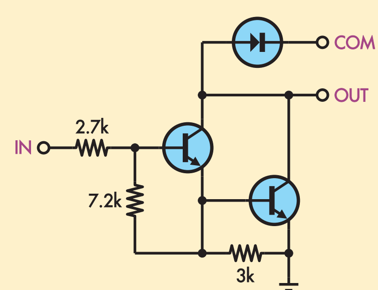


Fig.4: eight open-collector outputs are provided by IC4, a ULN2803 Darlington transistor array. The equivalent circuit for each output channel is shown here.

(PD0 & PD1) are programmed as serial transmit and receive lines for communication with the phone. A MAX232 level converter (IC3) transforms the TTL levels on these pins to about ±9V to drive the electronics embedded in the data cable.

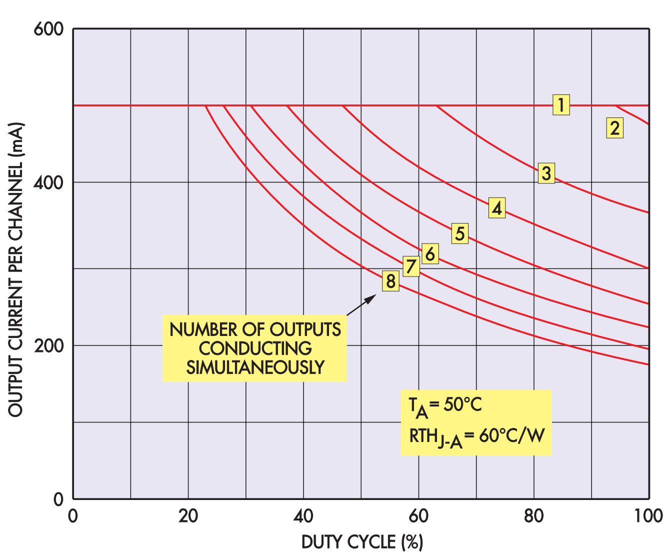


Fig.5: the amount of current the ULN2803 can sink depends on the number of outputs in use. Reproduced from the datasheet, this graph shows the maximum current per channel for 1-8 simultaneously conducting outputs. For most controller applications, a duty cycle of 100% should be assumed.

By way of explanation, electronic circuitry is included in the data cable to convert the logic levels from the phone (0 – 2.8V) into RS232 levels (about ±9V), so that the phone can be plugged into the serial port of a PC. We've therefore included a 'PC-like' interface for use with common types of cables.

The MAX232 also provides simulated 'RTS' and 'DTR' signals to the cable. 'RTS' is used by 'dual mode' cables to switch between the MBUS and the FBUS. In this design, 'RTS' is permanently driven to a negative voltage to select the FBUS connection. Conversely, 'DTR' is permanently driven positive by virtue of the direct connection to the positive output on V+ (pin 2) of IC3. This is used to power the circuits in the cable.

Power for the micro and its associated circuitry is provided by a 7805 +5V regulator (REG1). The input to the regulator is reverse-polarity protected by D1. Following this, a 10Ω series resistor and Zener diode ZD1 are included to provide transient over-voltage protection.

A 6.8V Zener diode (ZD2) provides limited protection in the case of serious over-voltage transients on the 5V rail. Note that if subjected to a substantial over-voltage, such as might occur during a nearby lightning strike, ZD2 would probably be destroyed. Always check the condition of this Zener if the fuse blows or the 10Ω 1W resistor is found to be open-circuit.

An under-voltage sensor (IC2) is used to reset the micro whenever the power supply voltage drops below about 4.6V.

Output switching

Eight outputs are provided for controlling external devices. Each output line is driven by one open-collector transistor pair in a ULN2803 (IC4). Fig.4 shows the equivalent circuit for one channel of the ULN2803.

All outputs are diode-connected to the 'COM' pin, which is then externally clamped to ground using a 36V Zener diode (ZD7). To allow for plenty of headroom, the open-circuit voltage at any output pin should not exceed +28V.

One ULN2803 output can switch a maximum load current of 500mA. However, when more than one output is used, this must be derated according to the graph in Fig.5. For example, with four outputs in use, the maximum current per channel is slightly less than 300mA.

Note that for this application, a duty cycle of 100% should be assumed. More information is available in the ULN2803 datasheet, which can be downloaded from www.allegromicro.com.

Fig.6(a) shows how to connect a simple relay circuit to any of the eight outputs. Note that a high-speed diode must be soldered directly across the relay coil terminals as shown. This diode limits the flyback voltage that occurs at relay switch-off, thus preventing high-voltage spikes from appearing across the driver output. We've specified UF4001 diodes for the job but of course, the higher voltage UF4002 and UF4003 devices can also be used.

If more current is required than can be provided by the ULN2803, the circuit shown in Fig.6(b) can be used. This circuit will handle at least 500mA, at the same time allowing all eight outputs to be used without overloading the driver. However, by substituting a power transistor and increasing the base drive, the current handling can be increased to over 1A – see Fig.6(c).

Input sensing

Four digital inputs (at CON3, Fig.2) are available for sensing the state of external trigger devices. Each input is current-limited by a 1kΩ resistor and is then clamped to +5.1V using a Zener diode (ZD3-ZD6). This scheme allows a maximum trigger input of 16V.

As shown in Fig.7(a), an input voltage of between 0 and 1.5V will be sensed as a logic 'low', whereas 3-16V will be sensed as a logic 'high'. Voltages in between these two ranges are considered invalid and may be sensed either 'low' or 'high'.

The micro samples these inputs every 128ms. Any single input change must be present for at least twice that time (256ms), otherwise it will be rejected as noise. If additional inputs change state within this 256ms window, they must remain valid for 500ms or more to be recognised.

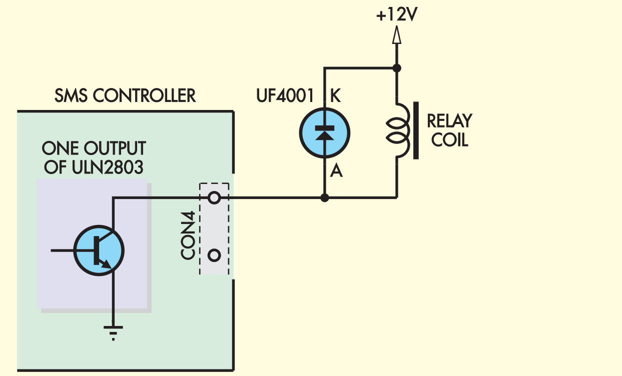


Fig.6(a): here's how to hook up a relay to any of the eight outputs. The diode must be soldered directly across the relay coil terminals. Take particular care that you have the cathode (banded) end to +12V, otherwise the ULN2803 will be destroyed!

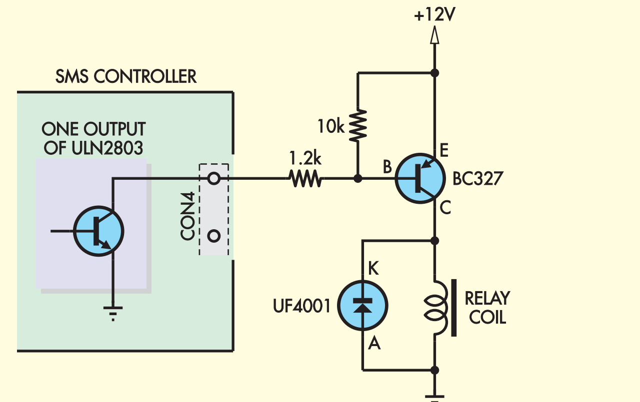


Fig.6(b): if more current is required than the ULN2803 can handle, then a transistor buffer circuit can be added. This circuit will switch at least 500mA.

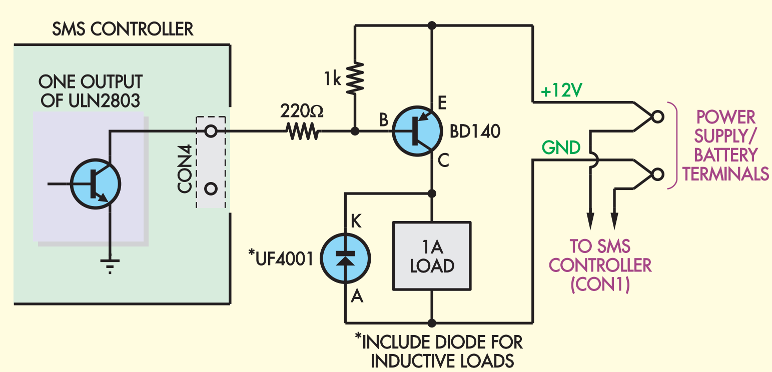


Fig.6(c): a power transistor can be used to boost current handling to over 1A. As shown here, power and ground for all external circuits must be independently wired to the power source.

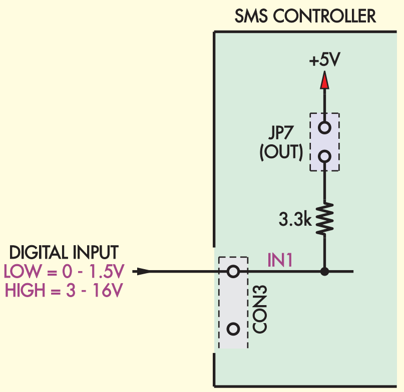


Fig.7(a): any of the four inputs can be used to detect the logic level of a digital signal. An input voltage of between 0 and 1.5V will be sensed as a logic 'low', whereas 3-16V will be sensed as a logic 'high'.

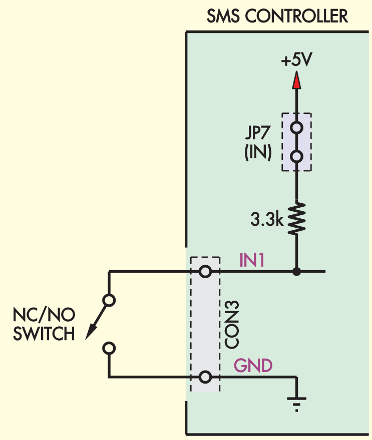


Fig.7(b): with the aid of the SMS Controller's on-board pull-up resistors, the state of a switch is easily sensed.

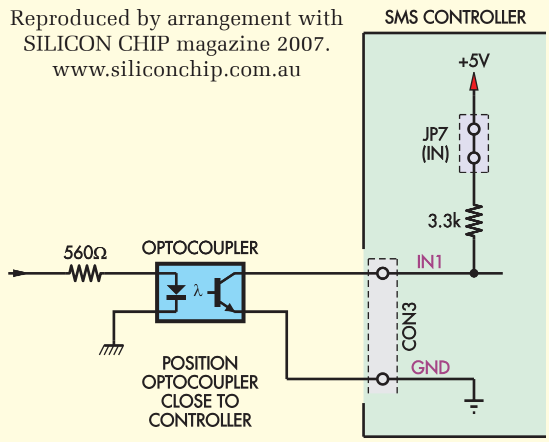


Fig.7(c): optocouplers can be used when the two systems lack a common ground. This is also an effective means of eliminating false triggering in noisy electrical environments and when using long cable runs. Select a resistor value that limits LED current flow to between 4mA and 20mA.

Jumpers JP4-JP7 allow a 3.3kΩ pull-up to be applied to any of the inputs for use with a switch (Fig.7(b)) or optocoupler (Fig.7(c)). The optocoupler scheme is necessary when the two systems do not share a common ground. It can also be used to eliminate false level sensing in noisy electrical environments and provides an effective isolation barrier against high-voltage transients. Any general-purpose optocoupler (eg, 4N25 or 4N28) would be suitable.

Important: *when using the circuits shown in Figs.7(a) & 7(b), the wiring between the equipment and/or switches and the input terminal block must be kept as short as possible. Do not connect long cable runs directly to the digital inputs! If you need to sense a signal over any significant distance, then use an optocoupler for isolation, as shown in Fig.7(c).*

Phone power supply

A simple step-down switchmode regulator circuit is used to power the phone – see Fig.3. It is based on the well-known MC34063 switchmode controller IC (IC5), which includes an oscillator, PWM controller and switching transistor – ie, most of the elements needed for a step-down design.

In short, the MC34063 regulates the output voltage by varying the amount of time an internal NPN transistor is switched on. The transistor's collector is connected to pin 1 and the emitter to pin 2. When the transistor is conducting, energy is transferred to inductor L1 and a 220µF capacitor. When it turns off, the energy is discharged into the load via D3.

In operation, the MC34063 attempts to maintain the output voltage at 7.0V, as set by the 22kΩ and 4.7kΩ resistors connected to pin 5. However, once the load current reaches about 350mA, internal current-limiting circuits begin to take effect.

The peak current level during each 'on' cycle is determined by the voltage at pin 7, which is developed across the paralleled 1.5Ω resistors. At about 350mA, the MC34063 begins to shorten the transistor 'on' time, thus limiting the output current. This also causes a drop in output voltage.

The result is a current-limited output of between 360mA and 400mA. When charging the phone's battery, the output voltage will typically fall to around 5-6V. This closely follows the performance of the standard ACP-7A plugpack charger.

Battery charging

According to Nokia, the batteries in these models must not be continually charged. In use, we found that the phone's battery charging circuits disconnect the DC input once the terminal voltage exceeds a certain absolute value. Some models also include a thermistor inside the battery pack and will terminate charging after a certain temperature rise. However, neither method eliminates overcharging.

To minimise overcharging, it is therefore necessary for the controller to be able to switch the current-limited supply on and off at the appropriate times. This is achieved in the circuit using diode D4 and a 4.7kΩ resistor between pin 14 of IC1 and pin 5 of IC5. When the micro drives this line high, it pulls the MC34063 feedback signal (FB) above the set point, forcing it to stop switching. In this condition, the internal switching transistor is off, so the input is disconnected from the output and no current flows to the phone.

In operation, the micro adopts one of two charging strategies, dependent on the particular model of phone. For the 5110 & 6110, battery level is monitored over the FBUS. When the level drops to '1', the power supply is switched on. When it reaches '4', the supply is switched off after a short 'top-up' period. To prevent sudden death due to a marginal battery, the supply is also switched on just prior to message transmission if the battery level is less than '3'. These numbers relate to the battery indicator bar on the right-hand side of the display.

As battery level information is not available on the 3110 & 3210 models, a simple timed charge regime is used

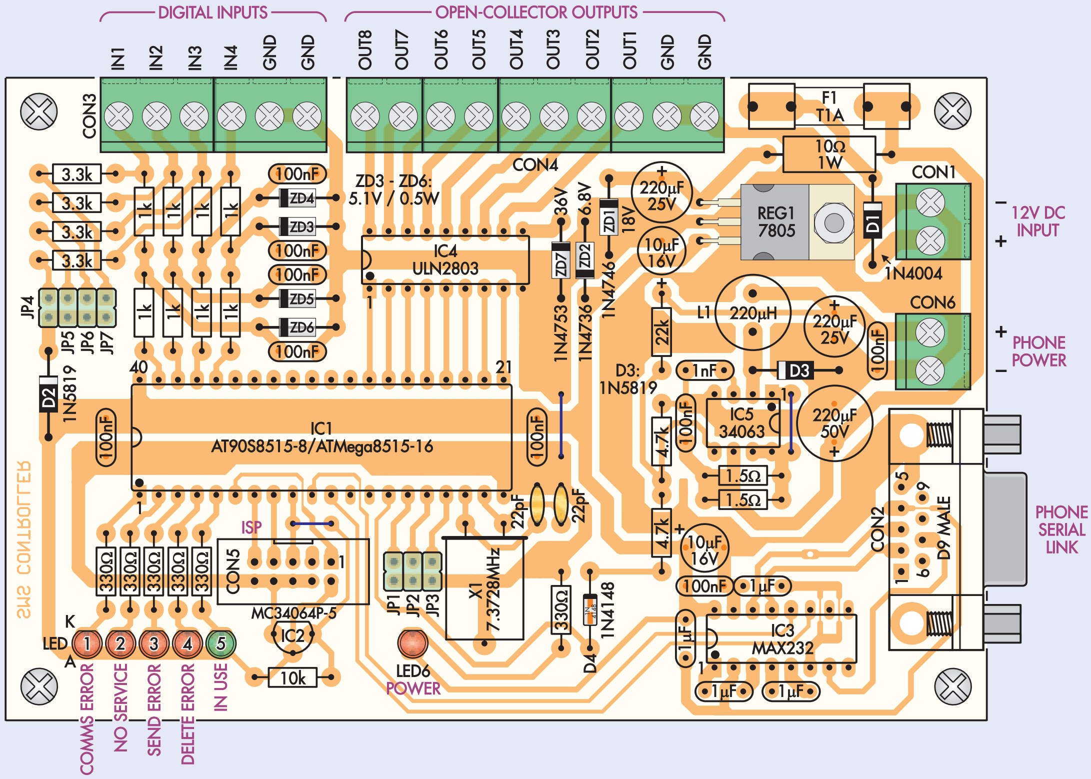
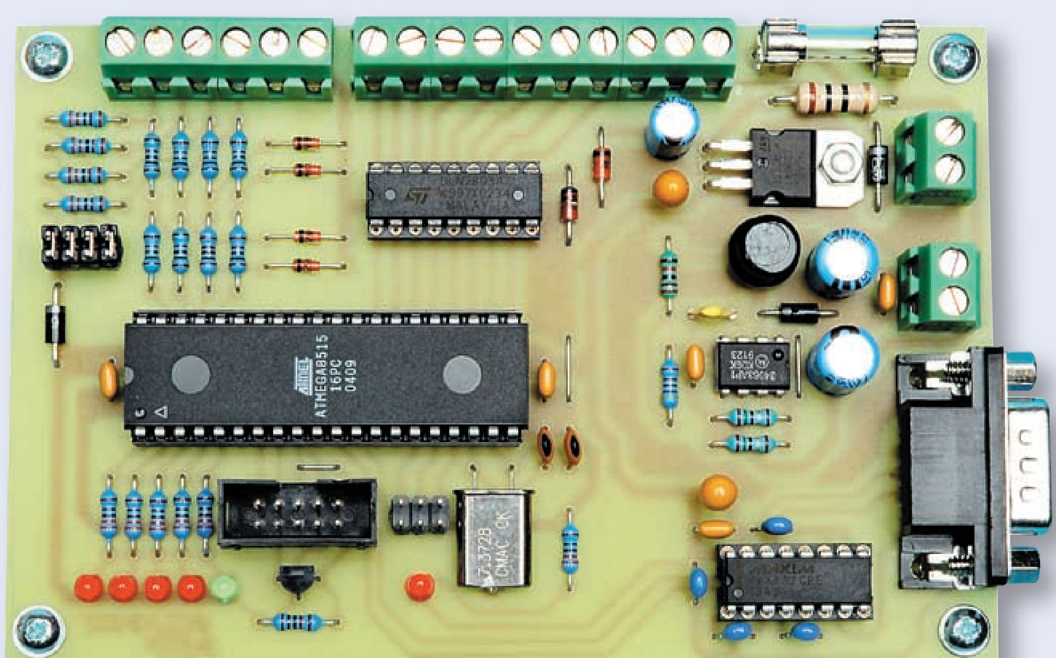


Fig.8: follow this diagram when assembling the controller. The orientation of all the ICs, diodes, LEDs and polarised capacitors is critical, so double-check all of these before applying power.



This is what the fully-assembled PC board looks like. Note that there are a few minor differences between this prototype and the final version depicted in Fig.8.

instead. At switch-on, the battery is charged for 40 minutes. Following this, the power supply is switched off for eight hours and then the cycle repeats over again. As we'll see next month, the default 40-minute charge time can be altered if desired.

This charge-discharge cycling continues indefinitely. Should a marginal battery cause the phone to switch off prematurely or an extended power failure occurs, the controller automatically brings the phone back on-line and resumes charging. Without this feature, you'd

Table 1: Resistor Colour Codes

❏	No.	Value	4-Band Code (1%)	5-Band Code (1%)
❏	1	22kΩ	red red orange brown	red red black red brown
❏	1	10kΩ	brown black orange brown	brown black black red brown
❏	4	3.3kΩ	orange orange red brown	orange orange black brown brown
❏	2	4.7kΩ	yellow violet red brown	yellow violet black brown brown
❏	8	1kΩ	brown black red brown	brown black black brown brown
❏	6	330Ω	orange orange brown brown	orange orange black black brown
❏	2	1.5Ω	brown green gold brown	brown green black silver brown
❏	1	10Ω 1W 5%	brown black black gold	not applicable
❏	1	10Ω 5W	not applicable	not applicable

Main Features

- Works with several popular Nokia brand phones
- Eight open-collector outputs
- Four digital inputs
- User-programmable plain text message control
- Communicate from any other mobile
- Password protected
- On-board phone power supply
- Ideal for alarm control panels
- Can be used in vehicles

have to press the power button to restore operation!

Unfortunately, this cannot be achieved with the 3210 & 3310 models, which lack support for remote control of the power button. In other words, a marginal battery or extended power failure will require that you physically press the power button to get the system back on-line.

Assembly

All the circuitry, including the phone power supply, is accommodated on a single PC board measuring 130mm × 85mm and coded 609. This has a row of screw terminals for the inputs (CON3) and outputs (CON4), as well as a 9-pin D socket (CON2) and screw terminals for phone power (CON6) and 12V (CON1). There are also the five status LEDs for the micro.

Fig.8 shows the assembly details. Begin by installing the three wire links using 0.7mm tinned copper wire or similar. Follow up with all the low-profile components, starting with the resistors, diodes (D1-D4) and Zener diodes (ZD1–ZD7). Take care to orient the banded (cathode) ends of the diodes as indicated. Also, double-check the numbers printed on the Zener diodes to ensure you have them all in their intended positions.

Install the IC sockets next, aligning the notched (pin 1) ends as shown. IC5 must be installed without a socket, noting that it goes around the opposite way to the other three DIL-packaged ICs. We also recommend that IC4 be soldered directly to the board (no socket), as in some applications it will have to dissipate considerable power. However, for low-power applications, such as when you'll only be driving one or two relays, an IC socket can be used if desired. Don't plug the micro (IC1) or MAX232 (IC3) into their sockets just yet – that comes later.

All remaining components can now be installed, leaving the connectors until last. Note that the flat (cathode) sides of the LEDs must all face towards the micro (IC1). In addition, the positive leads of the three electrolytic and two tantalum capacitors must be aligned with the '+' markings on the diagram.

To mount the 3-terminal regulator (REG1), first bend its leads at right angles about 5mm from the body. That done, slip it into position, checking that the hole in its metal tab lines up with the hole in the PC board. Adjust as necessary, then secure it to the board using an M3 × 6mm screw, nut and washer and tighten up before soldering the leads.

The leads of the crystal (X1) must also be bent at right angles, this time about 3mm from the body. Once in place, a short length of tinned copper wire can be soldered to the opposite end of the crystal case and to the pad underneath, grounding the case and securing it in position.

Finally, the 10-way and 6-way screw terminal blocks (CON3 & CON4) are made up by sliding 2-way and 3-way sections together, before mounting them on the board. Push them all the way down onto the board surface and hold them in place while soldering. The same goes for the remaining connectors; make sure they're fully in contact with the board surface before soldering their pins.

Controller checkout

The first job is to check out the power supply circuitry. Without IC1 or IC3 installed, connect a 12V DC power supply to the DC input terminals (CON1). A plugpack, 12V SLA battery or bench supply can be used for testing and it must be able to source at least 500mA of current.

Switch on and check that the power LED (LED6) illuminates. If it doesn't, switch off immediately and check that LED6, D1, ZD1 & ZD2 are all correctly installed. Also, check for a possible short circuit between the +5V rail and ground (0V) using your multimeter. Note that a short circuit will probably blow the fuse.

Assuming all is well, set your meter to read volts and measure between pins 20 & 40 of IC1's socket and pins 15 & 16 of IC3's socket. Both readings must be close to 5.0V (±5%). Any problems here must be rectified before continuing with the testing procedure.

Next, measure the voltage across the phone power supply output terminals (CON6). With nothing connected to these terminals, you should get a reading of about 7.0V.

If this is correct, switch off and install a 10Ω 5W resistor across the '+' and '-' terminals of CON6 to act as a load. This resistor will get quite warm

Cutting Corners: Using A Homemade Data Cable

Some readers will already be familiar with the Nokia FBUS/MBUS and software such as LogoManager and Oxygen Phone Manager. These products enable you to upload and download phone books and ring tones, create logos and more, using a PC.

Some may even have made up their own cables for connection to a PC. Making your own cable can save a few pounds but it's risky. A wrong connection and your phone or project may not survive. The results may also not be completely reliable. We'd therefore strongly recommend that you use a commercially made data cable for this project.

Having said that, we know that some diehards will want to have a go at making their own cable for the phone to controller connection, so here are the basics – use them at your own risk!

Commercial data cables include electronics for conversion between the FBUS/MBUS signal levels (0 - 2.8V) and RS232 levels (about ±9V) so that you can plug the phone into your PC. However, when using the phone with a microcontroller, a much less complicated level conversion scheme can be employed.

To modify the standard layout for direct phone to controller connection, leave out the MAX232 (IC3), the four 1µF capacitors and 100nF capacitor and install three resistors instead, as shown in Fig.9(b). The transmit (TXD), receive (RXD) and ground (GND) pins from the on-board D9 connector are then wired to the FBUS_RX, FBUS_TX and L_GND pins of the phone using shielded data cable. The length of this cable should be 550-600mm and the cable shield must be connected to ground.

We note that some circuits published on the Internet join MBUS to FBUS_RX and use a diode to connect back to the serial transmit line. This may work but it provides no protection for the microcontroller or phone signal lines.

The method used here translates the 5V logic levels from the micro's serial data output to about 2.7V for the FBUS serial input using a simple 2.7kΩ and 3.3kΩ resistive divider. On the return side, data transmitted on the FBUS is connected directly to the micro's serial data input via a 2.7kΩ current-limiting resistor.

The 2.8V logic levels from the FBUS mean that this scheme is running right on the margin and is not noise-immune. However, if you make the cable as we've described, you should find that it works reliably.

For the 5110 & 6110 models, an old hands-free set is a cheap source for the phone-side connector. For other models, you're on your own! Pinouts for the Nokia 3210 and 3310 models are readily available on the Internet.

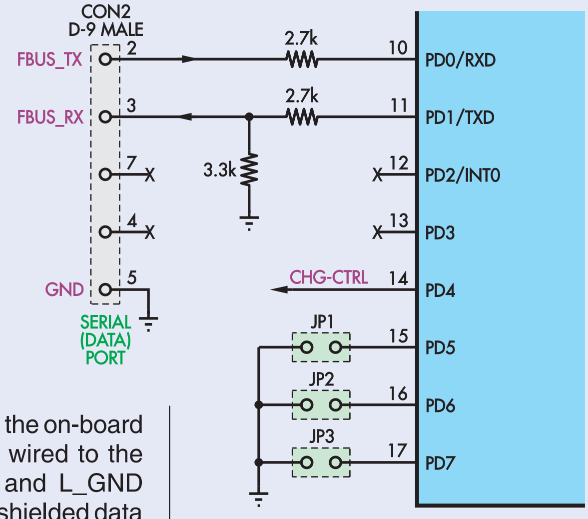


Fig.9(a): use this modified circuit if you intend using a homemade data cable (see text).

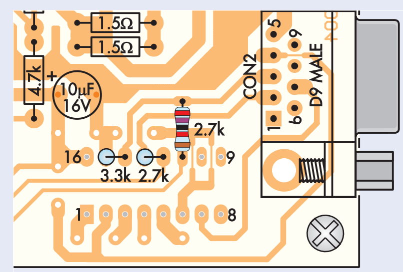


Fig.9(b): the modified board layout – just leave out the MAX232 (IC3) and five associated capacitors and install three resistors instead (note: the pads are numbered in line with IC3's original pin positions). Two resistors mount vertically between pads 15 & 14 (3.3kΩ) and pads 13 & 12 (2.7kΩ), while the third (2.7kΩ) goes between pad 11 and the spare pad directly above.

in operation, so make sure that it's not touching anything. Now power up again and measure the voltage across the 10Ω load resistor – it should be between about 3.6V and 3.9V.

In some cases though, this voltage may be higher than specified due to tolerances in the MC34063 and the 1.5Ω resistors. If it's 4.7V or less, it can be safely used as is. Alternatively, you can reduce the voltage to the specified level (3.6V - 3.9V) by increasing one of the 1.5Ω resistors to 1.8Ω.

If the voltage is still out of range, the first step is to make sure that the DC input voltage on CON1 is between 12.0V and 14.5V. If so, there is a problem somewhere in the switching regulator section shown in Fig.3. In particular, check that D2 is oriented correctly and that you've installed the wire link that goes between pins 1 & 8 of IC5.

Once the power supply checks out, disconnect the 10Ω test resistor and connect your phone's power cable leads. Be particularly careful that you have the polarity correct. This can be verified by measuring the voltage directly at the DC plug tip. With the black (-) probe on the barrel (outer) surface of the plug and red (+) probe on the inner contact, your meter should display a positive (not negative!) voltage.

That's all for this month. In Pt.2, we'll show you how to check out the remainder of the circuit, including the microcontroller and serial interface, and describe how it's used.