Remote Data Logging with Cell-Phones and the Internet

By means of the GSM cell-phone network and the Internet, data can be delivered from remote locations to anywhere in the world. Using this existing infrastructure can be an advantage over the more traditional point-to-point radio telemetry. An obvious use in speleology is the remote monitoring of long-term cave-based experiments or water levels and the progress of flood pulses. ***David Gibson*** *outlines the principles.*

In many fields of work, the gathering of data from remote locations and sending it to a central point for processing has become commonplace. The public utilities (gas, water, electricity) need to monitor remote pumping stations, reservoirs and so on, and this data is usually collected by radio telemetry. As cavers and cave scientists, we also have a requirement for remote instrumentation. Applications could include water level monitoring as well as any number of scientific studies.

The topic has been covered in several articles in the CREG journal. In (France, 1999; 2000) a data-logger for flood-pulse detection in caves was described, and some further practical results from this project were given in (Thomas, 2000). In (Lokhorst, 2002) an electronic detector for fluorescein added to the capabilities of remote loggers. (Gázquez et al, 2003) described a telemetry system for use in caves and (Bedford, 2005) described the use of radio modems for remote telemetry. Most recently, (Hurni, *et al.*, 2006) described a cave data-logger that used an earth-current cave-to-surface link, followed by a GSM (Global Standard for Mobile Communications) radio link.

The purpose of the present article is to describe the use of the GSM network and the Internet for data-logging – a technique of gathering data that is more flexible and adaptable than using a radio modem.

Choice of methods

Our requirement is to transmit data from a remote location to a personal computer based in an office or home. A simple method would be to use a radio modem that transmitted data to a radio receiver connected to a computer. Systems such as this have been available for some years. However, the disadvantages are many; for example...

- the transmission range is limited
- the transmission is point-to-point requiring a dedicated base station

These problems can be mitigated by making use of the public telephone network as an onward stage in the transmission. However, although such systems are widespread, they are not suitable for amateur or low-cost implementation; and we can see that there are advantages in being able to make use of an existing cellular radio network.

For commercial applications there is a choice of several systems including the TETRA network (terrestrial trunked radio) which is used by the emergency services and by large private organisations. However, for simplicity and ease of implementation it is clear that – in the UK at least – it is worth making use of the mobile phone network. Using GSM cell-phones and the Internet is not a new technique; it is already well-established in industrial applications and it is an obvious choice for remote data logging.

Using the GSM Network

In some situations, lack of coverage by the GSM network may require a different system to be considered but there is always the possibility of installing a radio modem at the remote end of the communications link. Additionally, affordable Wi-Fi systems can reportedly work over a distance of 80km with suitable antennas.

The interface to the GSM telephone is simple. Some phones will accept a serial input and will behave exactly like a modem. For example, using an enhanced 'AT' command set (the commands that control modems) they can send and receive text (SMS; short message service) messages; maintain an address book, and so on. Essentially we can use cell-phones and the GSM network as a long-distance point-to-point radio modem, for only a small amount of programming effort.

If a cell-phone has a serial modem-like interface then we will be able to access the SMS facility via an AT commands. Utilising text messaging has the advantage that such messages are often free of charge. The salient point is that a text message does not have to be read by a human. It can be extracted from the phone via its AT-command interface, so the SMS facility can be used to send small amounts of data between machines.

Using a GSM Modem

The above scenario depends on having a cell-phone with the required interface. A better alternative is to use an industrial 'GSM modem'. These are essentially mobile phones without the keypad, display and antenna. (See photo below).

An audio i/o connection allows a microphone and speaker to be connected, they contain a slot for a SIM card, and they often have advanced functionality that includes SMS and fax capabilities.

We could envisage, therefore, a small low-powered data-logger (e.g. a Microchip PIC-based device) that sent serial commands and data to a GSM modem, causing it to

- dial a number and **upload data** to a modem at the other end of the line, which was expecting it.
- dial a service centre and **send a text message** to another phone.
- receive a data call and **download data.**

Figure 1 – A GSM / GPRS modem
Photo from web site **www.rfsolutions.co.uk**.
A SIM card is inserted using the tray, bottom right.

Internet Operations

Once the data has been injected into the network it can, of course, be sent anywhere in the world. However there is the possible disadvantage that it has to be sent to another telephone or, via a modem, into a computer. This could be restricting in some applications, so it is worth attempting to combine the operation with transmission over the Internet. In this way, the data can be stored on a file server for access by anyone who has permission, or it can be placed on a web

server and accessed via a web site (again; by anyone who has the required password). This is clearly a much more flexible system.

To send data into the Internet we can use a GSM telephone as a modem and dial up the Internet as one would for any other dial-up Internet connection. However one difficulty is apparent. The simplest implementation (i.e. the one that requires the least thought) would be a PC running an Internet access program. But, unless we are going to supply each of our remote locations with a portable PC, batteries and software that does not crash periodically, this will not be possible.

Industrial PCs and similar products exist, and have the powerful functionality needed to implement an Internet link, but they can be expensive, and they consume a fair amount of power. (See Box *Power Consumption*).

It is possible to write an application for a small microcontroller that will dial an Internet 'point of presence'; negotiate a connection and send data. However, this is a difficult programming exercise because there are several levels of negotiation that must be done 'outside' the modem; at a higher level. At the very least, one must send a username and password when asked; and there are other aspects of the connection that require configuration too. Furthermore, Internet data has to be assembled into packets, with a routing address, etc., and this is a large overhead for a small microcontroller.

This may seem to suggest that a dedicated point-to-point data link over the telephone network would be simpler, as it would need only a modem at either end and, once the modems had negotiated with each other, character data would appear directly at the receiver. However, an Internet link is possible with less difficulty than suggested, if we make use of GPRS – General Packet Radio Services.

Power Consumption

In addition to the power consumption of the computer or microcontroller to consider, the power drain of the cell-phone or modem could be an issue for remote data-logging. However, if we use a GSM/GPRS modem it is easily possible to power it down by disconnecting its external power supply (which would be one of the functions of our microcontroller interface). The standby power of a phone is typically low, in any case. A phone that draws 24mW in standby would last for nearly four months on three D-size alkaline cells. As an alternative, it is not unreasonable for a small solar panel to be expected to deliver 5–10W in bright sunlight.

Many implementations by public utilities make use of solar panels, which have become a familiar road-side sight (see photo below). Solar panels are not used for environmental reasons, but because they are far cheaper to install than a hard-wired connection to the electricity grid.

Using GPRS

The latest mobile phones can assemble packet data for the Internet using GPRS. As before, we probably would not use an actual telephone but an industrially-packaged device. GPRS modems are widely available, although strictly speaking, the devices are not called modems, but 'terminal adapters'. Instead of accepting a data-stream at the 'modem' level, they accept high-level data. In technical parlance, they operate at the 'application layer' of the link. That means that the device will do all the link negotiation; it will assemble the data into packets, it will handle the TCP/IP network layer; and it leaves the user to deal solely with the application data. Depending on the commands one sends the device, one can...

- use **SMTP** to send an email
- use **POP3** to collect an email
- use **FTP** to transfer files
- use **HTTP** to access web servers

All this can be done using relatively simple programming on a microcontroller.

Application Layer Commands

When we use a PC for the above functions, the actual commands used to access the services are hidden from the user, but they are usually simple text strings. For example, when sending an email, the microcontroller will send data to identify who the mail is from, then it will send the command **DATA** which means "I would like to send you the message now". The server usually responds by sending

```
354 Enter message, ending with
"." on a line by itself.
```

Note that it sends a human-readable response even though a human does not usually get to see it. The important part is the code 354, which the program should interpret as an invitation to send the message text. The salient point is that there are other response codes that mean "do *not* send the data", and the client program running in the microcontroller must be capable of coping with them. A group of response codes you may be familiar with from email are the '500s', e.g.

```
553 User Not Known
```

If you have used a command-line FTP program (e.g. at the Command Prompt in Windows XP type **FTP**↵ then type **help**↵ to get a list of commands) you will be familiar with how an Application Layer protocol is constructed. SMTP, POP3 and HTTP all work in similar ways and, provided one has access to the specifications, it is easy to write MCU-based routines to handle them. This is simpler by several orders of magnitude than writing low-level code to assemble the data into packets and handle its transportation, so a GPRS terminal adapter is far more useful than a mere GSM modem.

There is probably only one circumstance when you might consider writing 'packet-level' code and that is if you are sending very small packets of data for which you do not require to know if they were delivered successfully. Time synchronisation data is one common example; 'ping' tests are another. Some information about low-level applications is given in the box, *Using a PIC to Access the Internet*.

A follow-up article – to appear in my new venture, the *Journal of Speleo-Electronics* (**caves.org.uk / radio / jspe**) – will look in more detail at Application Layer protocols and their use in remote data telemetry.

A demonstration is currently at **caves.org.uk / tests** although this assumes that you already have a good knowledge of the HTTP protocol.

References and Further Reading

Bedford, Mike (2005), *Radio Devices for Remote Data Logging*, CREGJ **59**, pp19-22, March 2005.

France, Stuart (1999), *Flood Pulse Logging with PICs*, CREGJ **37**, pp26-27, Sept. 1999.

France, Stuart (2000), *Flood Pulse Measurement: Preliminary Results from DYO*, CREGJ **39**, pp3-4, March 2000.

Gázquez, J., J. Calaforra, N. Novas, A. Fernández-Cortés, J. Verger (2003), *An Intelligent Telemetry System for Show Caves*, CREGJ **53**, pp6-8, Sept. 2003.

Hurni, J., F. Ziegler & C. Ebi (2006), *Robust Text and Data Exchange with Earth Current Transceivers*, CREGJ **63**, pp21-22, June 2006.

Lokhorst, Erwin (2002), *An Electronic Detector for Fluorescein*, CREGJ **49**, pp10-13, Sept. 2002.

Thomas, Adrian (2004), *A Flood Early Warning System for Irish Caves*, CREGJ **56**, pp4-9, June '04.

Using a PIC to Access the Internet

These are Microchip application notes.

Application AN724. Using PICmicro® MCUs to Connect to Internet via PPP. (Doc. DS00724c)

This describes a 'bare bones' application for the PIC16C63A. The demonstration software, written in C, allows the PIC to communicate with the Internet via its serial port and an external modem. A dial-up script using PPP is given, and a simple Ping routine that keeps the link alive, using ICMP (Internet Control Message Protocol). The article gives brief information on UDP (User Datagram Protocol) that can be used to send information by TFTP (Trivial File Transfer Pr'col).

Application AN731. Embedding PICmicro® Microcontrollers in the Internet. (Doc DS00731a)

This document describes a web server built around a PIC16F877 and the Seiko iChip™ S-7600A TCP/IP stack IC.

Application AN833. Microchip TCP/IP Stack. (Document DS00833a)

This describes Microchip's TCP/IP software, designed for a more powerful PIC18 device.

CREG