GPS-GSM Mobile Navigator

By Ma Chao & Lin Ming

What's the more laudable engineering feat, designing a navigation system capable of tracking ships in Shanghai Port or placing at the top of a competitive design contest? With the award-winning GPS-GSM Mobile Navigator, Ma and Lin accomplished both.

Ma Chao is a professor of Electronic Engineering at East China Normal University in Shanghai, China. He is a specialist in digital image compression and processing, embedded control systems, and computer network systems. You may reach Ma at ma-chao@online.sh.cn.

Lin Ming is a graduate student completing a Master's degree in Electronic Engineering at East China Normal University. He works primarily with embedded systems and microcontroller-based applications. You may reach him at lmcrr@online.sh.cn.

With today's stand-alone global position system (GPS) receivers, you are able to pinpoint your own position. But, what's more useful about stand-alone GPS receivers is that they can transmit your position information to other receivers. We decided to use both of these features to create a wireless vehicle tracking and control system for the Design Logic 2001 Contest, sponsored by Atmel and Circuit Cellar.

To design the Port Navigation System, we combined the GPS's ability to pinpoint location along with the ability of the Global System for Mobile Communications (GSM) to communicate with a control center in a wireless fashion. The system includes many GPS-GSM Mobile Navigators and a base station called the control center.

Let us briefly explain how it works. In order to monitor ships around a port, each ship is equipped with a GPS-GSM Mobile Navigator. The navigator on each ship receives GPS signals from satellites, computes the location information, and then sends it to the control center. With the ship location information, the control center displays all of the ships' positions on an electronic map in order to easily monitor and control their routes. Besides tracking control, the control center can also maintain wireless communication with the GPS units to provide other services such as alarms, status control, and system updates.

Hardware

GPS became available in 1978 with the successful launch of NAVSSTAR 1. Later, in May of 2000, the U.S. government ended selective availability (SA); as a result, the GPS accuracy is now within 10 to 30 m in the horizontal plane and slightly more in the vertical plane. For more information on GPS and its accuracy, read Jeff Stefan's article, "Navigating with GPS" (Circuit Cellar 123).

The GPS-GSM Mobile Navigator is the main part of the Port Navigation System. The design takes into consideration important factors regarding both position and data communication. Thus, the project integrates location determination (GPS) and cellular (GSM)—two distinct and powerful technologies—in a single handset (see Photo 1).

The navigator is based on a microcontroller-based system equipped with a GPS receiver and a GSM module operating in the 900-MHz band. We housed the parts in one small plastic unit, which was then mounted on the ships and connected to GPS and GSM antennas. The position, identity, heading, and speed are transmitted either automatically at user-defined time intervals or when a certain event occurs with an assigned message (e.g., accident, alert, or leaving/entering an admissible geographical area).

This information is received by the system in the dispatching or operations center, where it is presented as a Short Message Service (SMS) message on a PC monitor. SMS is a bidirectional service for sending short alphanumeric (up to 160 bytes) messages in a store√and√forward fashion. If the only data received is time and position, then the data can be displayed on a digitized map and also recorded in a database file; the recorded information can be replayed later for debriefing or evaluation of a mission.

The hardware block diagram is shown in Figure 1. The AT90S8515 microcontroller assures that all of the components work well together; it controls all incoming and outgoing messages as well as the I/O channels, serial interfaces (RS-232), peripheral devices (e.g., LCD and buttons), and all other parts. The GPS module receives the GPS signals and outputs the data to the AT90S8515 microcontroller via a TTL-level asynchronous serial (UART) interface. The microcontroller works with the GSM module by communicating with the GSM network. The interface between the GSM module and AT90S8515 is also TTL async serial. An RS-232 interface is used to exchange data with the PC.

Photo 1—On the front side of the main board, you can see an LCD, four programmable keys, a GSM module, an RS-232 connector, and some other components.

Because the AT90S8151 has only one UART, a three-channel multiplexer is used to switch among three working modes. The location information and other data is stored in the 2-Mb serial data flash memory of the AT45D021. The flash memory stores up to 2160 pieces of location information in 12 h, because the GPS-GSM Mobile Navigator saves GPS signals every 20 s. Four buttons, an LCD, and a buzzer enable you to display the system status and information and control the navigator.

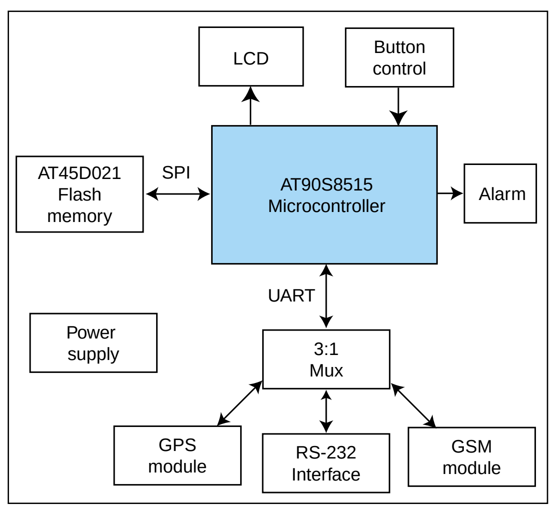

Figure 1—The AT90S8515 microcontroller is the basis for the GPS-GSM Mobile Navigator.

System Features

As we explained, the GPS module outputs the ship location information such as longitude, latitude, and Greenwich Time every 2 s. The location information is then stored every 20 s in flash memory, which has enough power to memorize the track of a ship even when the power is off.

Name	Example	Units	Description
Message ID	$GPRMC	–	RMC protocol header
UTC Position	161229.487	–	hhmmss.sss
Status	A	–	A = data valid; V = data not valid
Latitude	3723.2475	–	ddmm.mmmm
N/S Indicator	N	–	N = north; S = south
Longitude	12158.3416	–	dddmm.mmmm
E/W Indicator	W	–	E = east; W = west
Speed over ground	0.13	Knots	–
Course over ground	309.62	Degrees	True
Date	120598	–	ddmmyy
Magnetic variation	–	Degrees	E = east; W = west
Checksum	*10	–	–
<CR><LF>	–	–	End of message termination

Table 1—The NMEA RMC data values are based on the following example: $GPRMC,161229.487,A,3723.2475,N,12158.3146,W,0.13,309.62,120598,,*10.

Note that the GSM wireless communications function is based on a GSM network established in a valid region and with a valid service provider. Via the SMS provided by the GSM network, the location information and the status of the GPS-GSM Mobile Navigator are sent to the control center. Meanwhile, the mobile navigator receives the control information from the control center via the same SMS. Next, the GPS-GSM Mobile Navigator sends the information stored in flash memory to the PC via an RS-232 interface. (Note that you can set up the navigator using an RS-232 interface.)

There are two ways to use the mobile navigator's alarm function, which can be signified by either a buzzer or presented on the LCD. The first way is to receive the command from the control center; the second way is to manually send the alarm information to the control center with the push of a button.

The GPS-GSM Mobile Navigator is powered by either a rechargeable battery or DC input.

Getting GPS Data

After the GPS module computes the positioning and other useful information, it then transmits the data in some standard format—normally in NMEA-0183 format. When you're building this project, it's nice to be able to buy stand-alone GPS OEM modules. Just check the pages of Circuit Cellar for manufacturers. We used a Sandpiper GPS receiver from Axiom for this project. The Sandpiper is intended as a component for an OEM product that continuously tracks all satellites in view and provides accurate satellite positioning data. With differential GPS signal input, the accuracy ranges from 1 to 5 m; however, without differential input, the accuracy can be 25 m.

The Sandpiper has two full-duplex TTL-level asynchronous serial data interfaces (ports A and B). Both binary and NMEA initialization and configuration data messages are transmitted and received through port A. Port B is configured to receive RTCM DGPS correction data messages, which enable the GPS unit to provide more accurate positioning information. But, we didn't require the use of port B for this project.

About 45 s after the GPS module is cold booted it begins to output a set of data (according to the NMEA format) through port A once every second at 9600 bps, 8 data bits, 1 stop bit, and no parity. NMEA GPS messages include

Reprinted with permission of Circuit Cellar® - Issue 151 February 2003

Name	Byte	Definition	Description
Start byte	1	:	Start symbol of data package
Data package ID	1	0~9	Package ID is repeated from 0 to 9
System password	3	000~999	System password
Terminal ID	4	0000~9999	Terminal ID
Position data	19	E000000000~E180000000	E means east longitude, which is from 000° and 00.0000 min. to 180° and 00.0000 min.
		N00000000~N90000000	N means north latitude, which is from 00° and 00.0000 min. to 90° and 00.0000 min.
UTC	6	hhmmss	Greenwich Time (hour, minute, second)
Upload time rate	3	001~255(003)	Upload time interval = basic upload time ˇ upload time rate
Alarm information	4	xxxx	0 means OK; 1 means alarm Byte 1: aberrance alarm Byte 2: over-speed alarm Byte 3: dangerous area alarm Byte 4: manual alarm
Stop byte	1	#	Stop symbol of data package

Table 2—Take a look at the 42-byte data package format and the following example ready to be saved: :10019999E121263457N311864290742160030000#.

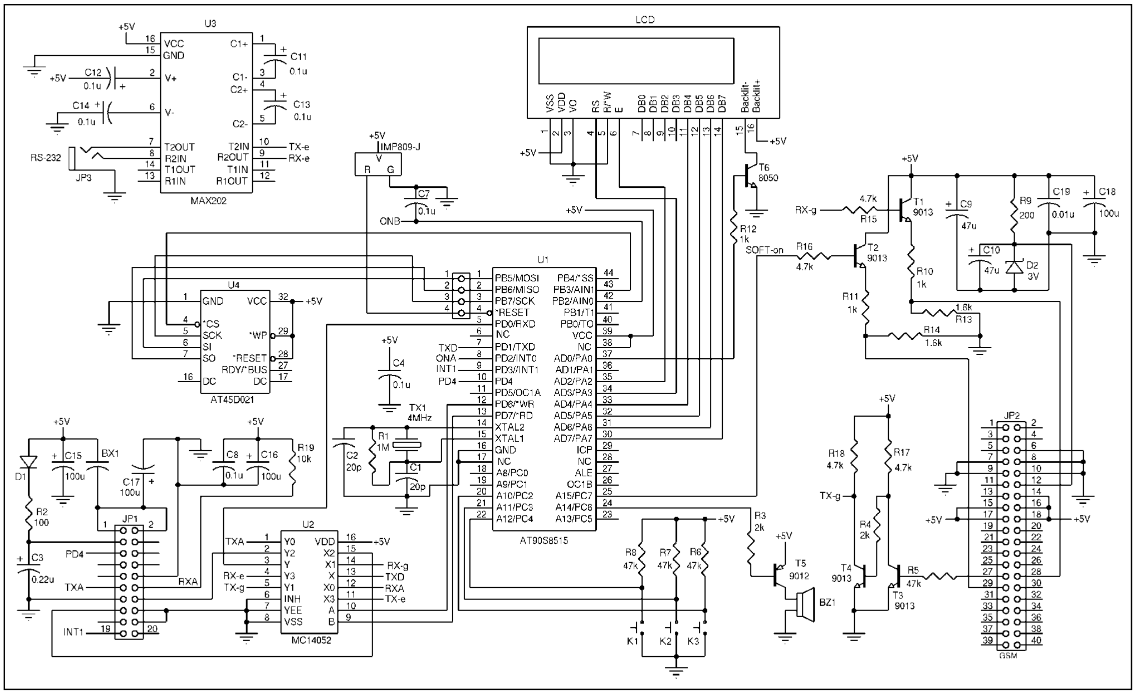

Figure 2—Jack port JP1 is the 20-pin GPS socket header. Jack port JP2 is the 40-pin GSM socket header. U2 is the dual four-channel multiplexer controlled by PA2 through PA3. All of the data traffic runs at 9600 bps.

six groups of data sets: GGA, GLL, GSA, GSV, RMC, and VTG. We use only the most useful RMC message—Recommended Minimum Specific GNSS Data—which contains all of the basic information required to build a navigation system. Table 1 lists the RMC data format.

We only need position and time data, so the UTC position, longitude with east/west indicator, and latitude with north/south indicator are picked out from the RMC message. All of this data will be formatted into a standard fixed-length packet with some other helpful information. Next, this data packet will be transmitted to the control center and stored in the AT45D021's flash memory.

The data packet is a 42-byte long ASCII string, which includes the package ID, system password, terminal ID, position data, UTC, and other operational information. Table 2 shows the definition of a reforming data packet and an example ready to be saved or transmitted.

GSM TRANSMITS DATA

A committee of telecom vendors and manufacturers in Europe—the European Telecommunications Standards Institute (ETSI)—designed GSM as a digital wireless communications system. Commercial service began in mid√1991, and by 1993 there were 36 GSM networks in 22 countries, with 25 additional countries looking to participate. Furthermore, the standard spread quickly beyond Europe to South Africa, Australia, and many Middle and Far Eastern countries. By the beginning of 1994, there were 1.3 million subscribers worldwide. Today, GSM is also the most widely used communications standard in China, and covers almost all of the country. So, we didn't need to set up a communications base station for our system; this, of course, significantly reduced the total cost of the project. The most basic teleservice supported by GSM is telephony. Group 3 fax, an analog method described in ITU√T recommendation T.30, is also supported by the use of an appropriate fax adapter. SMS is one of the unique features of GSM compared to older analog systems. For point√to√point SMS, a message can be sent to another subscriber to the service, and an acknowledgment of receipt is sent to the sender. SMS also can be used in Cell Broadcast mode to send messages such as traffic or news updates. Messages can be stored on the SIM card for later retrieval.

SMS is effective because it can transmit short messages within 3 to 5 s via the GSM network and doesn't occupy a telephony channel. Moreover, the cost savings makes it a worthwhile choice (i.e., in China, each message sent costs $ 0.01 and receiving messages is free). With SMS transmitting, gathering position data is easy and convenient.

PROJECT FILES
To download the pin assignments and source code, go to ftp.circuitcellar.com/pub/Circuit_Cellar/2003/151/.

Command	Definition
AT+CSCA	Set the SMS center address. Mobile-originated messages are transmitted through this s ervice center.
AT+CMGS	Send short message to the SMS center
AT+CMGR	Read one message from the SIM card storage
AT+CMGD	Delete a message from the SIM card storage
AT+CMGF	Select format for incoming and outgoing messages: zero for PDU mode, one for Text mode
AT+CSMP	Set additional parameters for Text mode messages

Table 3—To send SMS messages, you can use these (mainly) AT commands. For more details, you may want to read the GSM 07.07 protocol on the ETSI web site.

REFERENCES

[1] European Telecommunications Standards Institute, "ETSI GTS GSM 07.05," V.5.5.0, 1998.

[2] ———, "ETSI GTS GSM 07.07," V.5.0.0, 1996.

RESOURCE

NMEA Specification
National Marine Electronics Association
(919) 638-2626
www.nmea.org

SOURCES

AT90S8515 and AT45D021
Atmel Corp.
(714) 282-8080
www.atmel.com

Sandpiper GPS receiver
Axiom Navigation, Inc.
(714) 444-0200
www.axiomnav.com

FALCOM A2D GSM module
Falcom Wireless Communications GmbH
(800) 268-8628
www.falcom.de

BASCOM-AVR
MCS Electronics
+31 75 6148799
www.mcselec.com

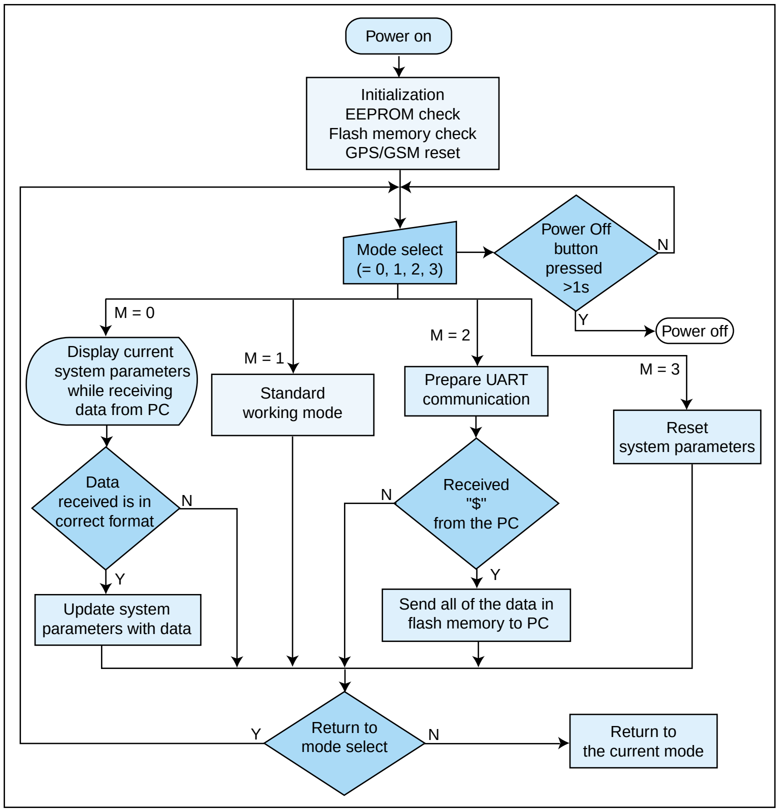

Figure 3—After initialization, you can select the function mode by pressing the Menu button and Enter button. The LCD will show the status and system parameters.

As with GPS modules, stand-alone GSM OEM modules are available. We used the FALCOM A2D from Wave-com for this project. The FALCOM A2D is a dual-band embedded GSM module (GSM900/DCS1800). It features the following services: telephony, SMS, data, and fax. The GSM module has one TTL-level serial data interface. We use AT commands to control and program the FALCOM A2D. The data and control commands are exchanged between the microcontroller and GSM module through the serial interface.

There are many groups of AT commands, including: Call Control, Data Card Control, Phone Control, Computer Data Card Control, Reporting Operation, Network Communication Parameter, Miscellaneous, and Short Message Service. We use some of the SMS commands to communicate with the control center. The main AT commands for using SMS are listed in Table 3. You can download the GSM 07.07 and GSM 07.05 protocols for more details about the AT commands that are used in GSM communications. [1, 2]

Let's review an example of how to make a GSM module send and read a sample SMS in Text mode. First, initialize the GSM module with AT commands AT+CSCA and AT+CMGF. Using the former sets the SMS center number to be used with outgoing SMS messages. Remember, the number will be saved on the SIM card just like in normal mobile phones. There are two different modes—Text mode and Protocol Data Unit (PDU) mode—for handling short messages. The system default is PDU mode; however, Text mode is easier to understand. So, use the AT+CMGF=1 command to set the module to the GSM 07.05 standard SMS Text mode.

The AT+CMGS command is used to send a short message. The format of this command is:

```
AT+CMGS=<da><CR>Message
   Texts<CTRL-Z>
```

Here, <da> is a subscriber's mobile phone number that you want to send the short message to. The GSM module can receive incoming short messages and save them on the SIM card automatically. You can use the AT+CMGR command to read an incoming short message from the SIM card storage, and then use the AT+CMGD command to delete it when you're finished.

If you want to read an SMS message, then send a AT+CMGR=x command to tell the GSM module which short message you want to read. Next, check the serial port to receive the message from the GSM module. Rs232_r is a subroutine used to receive data from the UART. Listing 1 demonstrates sending and reading a short message in a BASCOM-AVR program. In this code segment, chr(34) converts the ASCII value 34 to the right quote character ("). It also converts chr(13) to <CR> and chr(26) to <CTRL-Z>. As you can see, "My SMS Message" is the message you want to send.

Circuit Description

The difficult part of designing this project was learning both the NMEA GPS message and GSM AT command protocols. The easy part was designing the hardware circuit (see Figure 2). You may download a table of the pin assignments from the Circuit Cellar ftp site. As you can see from the schematic, there are three jack ports. JP1 (20 pins) is used for the GPS module, JP2 (40 pins) is for the GSM module, and JP3 is used for communication with the PC.

The AT90S8515 (U1) is the core of the circuit. This low-power CMOS 8-bit microcontroller is based on the AVR-enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8515 achieves throughputs approaching 1 MIPS per megahertz, allowing you to optimize power consumption versus processing speed. The AT90S8515 features 8 KB of in-system programmable flash memory, 512 bytes of EEPROM, 512 bytes of SRAM, and 32 general-purpose I/O lines. Flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, an SPI serial port, and two software-selectable power-saving modes are also available. The high speed of the AT90S8515 makes it possible to complete multiple tasks between the GPS and GSM modules, although it has only one UART serial port. With the programmable flash memory, you have high reliability and can update your system. The EEPROM makes it possible to store system parameters such as the SMS center number, control center number, and predetermined time intervals.

Other components on the board are the four-channel multiplexer, a large capacity data memory, and the user interface. The latter consists of a 2 ˘ 16 LCD, a buzzer, and three push buttons.

Accessories

An AT45D021's serial-interface flash memory is used as a black box to store data packages. The 2,162,688 bits of memory are organized as 1024 pages of 264 bytes each. In addition to the main memory, the micro also contains two data SRAM buffers of 264 bytes each. The simple SPI serial interface facilitates the hardware layout, increases system reliability, and reduces the package size and active pin count. The AT90S8515 saves GPS data to flash memory via an SPI port at a user-defined specific interval. Or it reads data from the

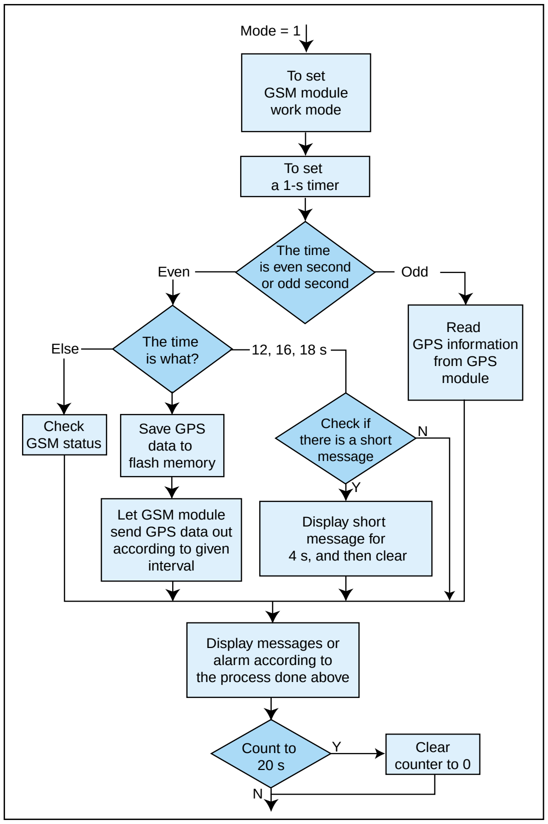

Figure 4—The main function is mode 1. The AT90S8515 microcontroller receives the ship location data every 2 s from the GPS module, and then saves the data in flash memory every 20 s. At a user-defined time interval, the AT90S8515 sends the location data to the control center, and then receives the control information from the control center via the GSM module.

flash memory to backup to PC. Up to 2160 pieces of information can be stored in flash memory. Because the AT90S8515 has only one UART port, another chip is used to expand the serial port for three kinds of different functions. The digitally controlled MC14052B analog switch is a dual four-channel multiplexer. With two I/O pins, the AVR controls it to switch among three channels, all of which are UART serial interfaces.

One MAX202 chip accomplishes the conversion between TTL/CMOS level and RS-232 level, which is necessary for the RS-232 interface between the navigator and PC. Using the RS-232 port, the system can backup the data in flash memory to the PC. Also, you can change some system parameters through the PC via the RS-232 port.

With two control pins and four data pins, the AVR gives the LCD specific information to display. Port pins PC2 through PC4 individually sense the three push-button switches. There is a Menu button to select the work mode, and an Enter button to confirm the selection. The third is an SOS button used to send an alarm message to the control center.

Software Description

We used the powerful BASCOM-AVR to develop the software. An IDE is provided with an internal assembler. You can also generate Atmel OBJ code. Additionally, the BASCOM-AVR has a built-in STK200/300 programmer and terminal emulator. Other notable features include: structured BASIC with labels; fast machine code instead of interpreted code; special commands for LCDs; I2C; one wire; PC keyboard and matrix keyboard; RC5 reception; and RS-232 communications. The BASCOM-AVR has an integrated terminal emulator with download option, an integrated simulator for testing, and an integrated ISP programmer.

You can easily write the firmware for this project using the BASCOM-AVR. And with the ISP benefit of AVR, on-line emulation is almost unnecessary, so you can program and test with ease. The flow charts in Figures 3 and 4 describe the AT90S8515 program that controls the devices. The software handles a number of key functions, such as initializing the system and starting the GPS and GSM modules. The software also selects the working mode. Additionally, it checks and sets the system parameters in mode 0, backs up the trace data stored in flash memory to the PC in mode 2, and resets the system parameters in mode 3.

Mode 1 is the standard working mode during which many tasks are completed. During mode 1, the GPS signals are read every 2 s from a satellite; the location information is saved in flash memory every 20 s; and the GSM module sends location data to the control center according to the given interval time. Meanwhile, the navigator receives the control information from the control center from the GSM module.

❑

Our system is now being used in Shanghai Port, China for navigation and monitoring of ships. Aside from tracking ships, the GPS-GSM Mobile Navigator can also find use in other applications, such as navigating taxis. The system works quite well, and we plan to adapt it for future projects.

Listing 1—We created a program to send an SMS message to a mobile phone (13916315573). The program directs the GPS-GSM Mobile Navigator to read and delete an incoming short message. The Print command is a BASCOM-AVR instruction that sends output to the serial port. The Rs232_r subroutine is used to read input from the serial port.

```
constant definition
Const Gsm_center = "+8613800210500"//SMS center number
Const Send_number = "13916315573"          //Phone number the SMS sends to
Const Sms_texts = "My SMS Message"         //Message texts to be sent

//Initialize the GSM module
Print "AT+CMGF=1"
//Set GSM module in Text mode
Print "AT+CSCA=" ; Chr(34) ; Gsm_center ; Chr(34)
                                           //Set SMS center number
//Send a message
Print "AT+CMGS="; Chr(34); Send_number; Chr(34); Chr(13); Sms_texts; Chr(26)

//Read and delete an incoming short message
Print "AT+CMGR=1"                          //Read first short message from SIM card storage
Gosub Rs232_r                              //Receive message
Print "AT+CMGD=1"                          //Delete message from SIM card storage
```