Development of a Remote Serial Cable Over GSM

Paulo J. Nóbrega, Telmo R. Cunha

Abstract: **Diverse monitoring and control applications require that a host computer communicates with one or several remote units (sensors, actuators, …). Usual off-the-shelf communication links require that both host computer and remote units adapt themselves to data protocols or physical properties of the devices that implement the link. Since almost all remote devices have a built-in microprocessor, which usually has one or more UART transceiver, a communication link was designed to work over this interface with the objective of minimizing the hardware and firmware changes required on both ends of the link. This article presents the design and implementation details of this communication link, which was denominated Remote Serial Cable – RSC. The RSC is a hardware interface that allows bidirectional communication between remote equipment and host computer, being that communication channel seen by both ends as a transparent RS232 connection, as if they were connected directly through a standard serial cable. It uses the GSM network to communicate between RSC Units, transporting the data through SMS, Circuit Switched Data (CSD), or GPRS.**

I. INTRODUCTION

NOWADAYS, it is common to find applications where there is the need of a host computer to gather data from a number of widespread sensors, or to control a group of remotely located actuators or devices. And with the mobility tendency of people and equipment, this host computer does not necessarily need to be a fixed desktop computer on a server room, but can be a laptop or even a PDA. Also, the remote equipment can be in motion, for some applications. Since the nature of the remote devices is usually very simple in terms of processing capabilities, having in most cases a simple microprocessor controlling the device functions, it was thought in the presented work that it could be possible to use the basic hardware features presented in such remote devices in order to communicate bi-directionally with the host computer, causing the least possible changes on the remote devices, and keeping these as simple as possible (from the hardware perspective). It was then recognized that almost microprocessors have available an UART transceiver, which is, in most cases, accessible though a RS232 serial port (for local configuration of the devices, and other operations). With this in mind, a communication link, named Remote Serial Cable – RSC, was designed and implemented using hardware units (RSCU) to convert the RS232 channel into a communication over a GSM network (between two RSCUs) converting again, at the other end, the data back to the RS232 protocol. This operation is performed in a transparent way from the point of view of the end-devices, that is, either host or remote equipment look at the communication link as if it was a standard serial cable connecting them. Fig.1 illustrates the concept of the implemented link.

Depending on the application, the communication required between host and remote devices can have different usage characteristics, in terms of channel usage with time. That is, some applications may require an open link for a very short period once a day, or once a week, and other applications may require data to be sent/received every 5 seconds, for example. In order to accommodate these distinct operation modes, enabling a configuration of the link that would minimize the communication costs, the RSCUs can operate in SMS mode (for sporadic low volume data communication), in CSD mode (for sporadic high volume data communication) and also in GPRS mode (for frequent, open channel, data communication). In the latter case, no RSCU is required at the host computer because it simply uses a TCP/IP port to connect to the remote units (which continue to look at the link as if it was an RS232 link).

The following sections present the hardware details of the RSCU devices, and also the description of some of the firmware features that were implemented.

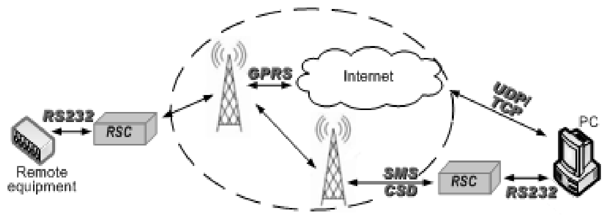

Fig.1 - Transparent RS232 communication between remote and host equipment, over a GSM network, with RSC units controlling the link.

This section describes the hardware details of a RSCU. Fig.2 shows a block diagram of its main hardware blocks. It should be mentioned that all RSCUs are exactly equal in terms of hardware construction. There is no difference between the host-end RSCU and the remote device-end RSCU.

Starting with the power supply unit, this was designed to allow a high range of DC power voltages, from 7V to 40V, so that it could be used as is in diverse applications. As some remote equipment have to rely on batteries for their power supply, and in order to reduce unnecessary power consumption, a DC-DC switched converter was implemented, capable of delivering 3A (mode than the necessary to enable the GSM communication).

The unit is controlled by a microcontroller, a PIC18 family MCU, which interfaces the communications between the external UART and the GSM module. The microcontroller is capable of buffering the communication data from/to the UART or the GMS module, and of initiating the peripherals with the configuration provided by a hardware switch, or by a configuration word stored in the MCU flash (the configuration reference is selectable). This configuration consists on the baud rate used for serial communication, the GSM operation mode (CSD, SMS or GPRS), and other less relevant parameters.

The UART has 2 internal FIFOs, one for transmission and other for reception, each with 16 bytes. The main function of the UART is to perform serial to parallel data conversion, so the microcontroller can exchange communication (or configuration) data with the UART through its I/O ports.

It should be mentioned that the selected MCU only contained a single internal USART, which was dedicated to the communication between MCU and GSM module. Also foreseeing possible future extensions for the RSCUs, aiming to simultaneous communication with more than one remote equipment per RSCU, it was decided to use a general external UART in this stage.

The UART has a programmable baud rate generator, and a selectable auto flow control feature that can significantly reduce software overload on the microcontroller and increase the RSCU efficiency by automatically controlling the serial flow control signals (RTS and CTS).

To interface the external equipment, a line driver was inserted in order to switch the serial signal level from TTL (at the UART side) to RS232 (at the external equipment side). Also, this driver has built-in protections that avoid voltage peaks (or over-voltages caused by device misuse or equipment malfunction) to damage the RSCU internal components.

The connector to the external equipment is a serial female DB9 plug, since the RSCU is seen as a DCE (Data Circuit-terminating Equipment) unit.

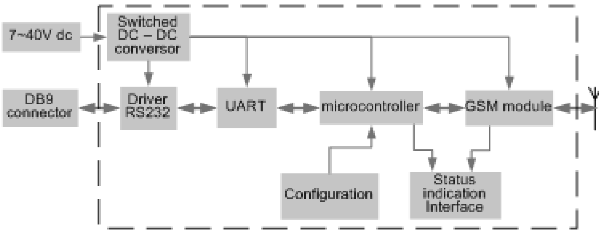

Fig.2 - Basic block diagram of the RSC unit.

The connection to the GSM network is made by a dual-band GSM/GPRS module (the GM862-GPRS from Telit Mobile Terminals, S.p.A.) that allows data calls (CSD), GPRS and SMS handling. It has an embedded SIM card reader, and exchanges data and configuration commands with the microcontroller, through the connection between the microcontroller and GSM module USARTs. The module operates in two possible modes: data mode (all the data received by the module is send to the network), and command mode, being the communication protocol based on AT commands.

Also considered in the RSCU implementation was a LED interface which allows a quick overview of the RSCU status at every instant. This contemplates LEDs indicating communication activity between external equipment and the MCU (receiving and transmitting), and between the MCU and the GSM module (receiving and transmitting). The MCU execution status was also mapped into a LED interface which, through the frequency of its blinking, would indicate if the MCU was in normal operation, if it encountered a problem, if it was running in debug mode or in boot mode, and so on. Also, the GSM module status could be monitored through LEDs, indicating if it was connected to the GSM network, if a call was (or not) currently established, etc.

The PCB design of the RSCU included, for debug purposes, a series of inside connectors to which a computer can be plugged into. For example, it is possible for a debug laptop to snoop the data exchanged between MCU and GSM module, or even to overwrite the MCU commands for the GSM module.

Two views of the implemented RSC units are shown in Fig.3 and Fig.4, presenting the outside view and the PCB board, respectively.

Fig.3 - Outside view of the RSC unit.

Fig.4 - Inside view of the RSC unit.

II. OPERATION

As referred in the above sections, the RSC has 3 different operation modes: SMS, CSD and GPRS. In SMS mode, all serial data received from the equipment are converted into SMS messages, and all SMS messages received from the network are sent to the serial interface, to the external equipment, after being parsed by the microcontroller firmware. The SMS mode is preferred for exchanges of small sporadic bursts of data. A SMS message is sent whenever a total of 160 (maximum size of an SMS) bytes are received from the external equipment, or when the equipment stops sending data for more that a certain time period (which can be programmed). In the other direction, as soon as a SMS is received from the network, its data field is sent to the external equipment. Note that, in this mode, the time required for data to arrive its destiny is dependant of the GSM network efficiency.

More adjusted to real time applications which sporadically transfer large data amounts is the CSD mode. This mode has real time specifications similar to a voice call, which requires a small delay between both callers. In this mode, in order to exchange data, a call must be established between both RSC units. This takes about 30 seconds and is triggered as soon as the first serial data byte is received from the external equipment. The data call ends when one of the RSC units stops the call, or when a time out condition is reached from the reception of the last serial data byte from the external equipment. This time out is adjustable. Another advantage of CSD mode is that the communication channel is full duplex.

In GPRS mode, data is sent directly to the internet, so there is no reserved path or dedicated circuit. The GSM/GPRS module organizes data into TCP/IP packets and sends them to the mobile operator's network where they will be forwarded to their destiny, through the network routers. This mode of operation requires the base computer to have internet access and a known static public IP address. When the RSC unit is configured to work in GPRS mode, as soon as it is powered, it establishes a TCP/IP connection to the base computer so data can be exchanged. Note that the base computer IP is known by the remote RSC unit, but usually there is no reserved static IP address for the GSM/GPRS module, so its IP is dynamic and unknown to the base computer. GPRS mode is advisable for applications that are constantly exchanging data, either in high or low volume.

III. CONFIGURATIONS

It's possible to configure communication parameters, mode of operation, and even change the firmware of the RSC unit without any extra hardware.

There is a hardware dipswitch that allows choosing the external equipment serial baud rate, the RSC mode (or leaving this choice to the PC software), and also to force the unit to run in the boot/debug mode.

In boot/debug mode, after the RSC unit resets, it listens to its serial port and if the PC wants to send a new firmware configuration, the RSC unit starts its bootloader. Note that the RSC unit bootloader was specially designed, because the firmware is uploaded to the microcontroller through the external UART connected to the microcontroller I/O pins. It is possible with the developed bootloader (and associated PC loader), to upload to the microcontroller's flash, special registers and EEPROM. Taking advantage of this feature, the RSCU configuration is kept in the microcontroller's EEPROM, and an API for the PC was developed, which allows on-line configuration of the RSCU operation mode.

With this API, the user can choose the communication mode, and the individual parameters for each communication mode.

In SMS mode only the destiny's SIM phone number needs to be configured.

When in CSD mode, the user needs to configure not only the destiny's SIM phone number, but also the time out threshold from the last received byte (after which the RSC will end the call).

If the GPRS mode is chosen, the SIM operator's access point name, login and password will have to be configured. Also, the destination IP address and port (those of the base computer) need to be specified.

When boot/debug mode is active, all the steps taken by the RSC unit are reported back to the serial port. This mode is helpful mostly when the RSC is first time installed or a change was made to the system firmware. In normal operation, this option remains disabled, otherwise the reports of the RSC unit would be interpreted as communication data.

IV. THE RSC UNIT FIRMWARE

The microcontroller is the responsible of initiating the UART, the GSM/GPRS module and itself according to the predefined configuration. After the entire configuration is executed, the microcontroller acts like a bridge between the UART and the GSM/GPRS module.

Excluding the GSM module's internal FIFOs, there are 4 FIFOs in the RSC unit. Two of them are in the UART and can store 16 bytes, the other two are circular buffers in the microcontroller's RAM that can store each 256 bytes of data. Fig.5 shows an overall buffering scheme implemented in the microcontroller.

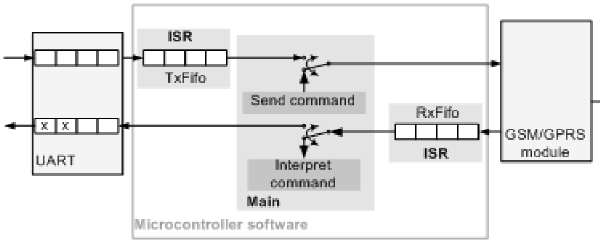

Fig.5 - Simple scheme of the microcontroller's buffering scheme.

The microcontroller's transmitting FIFO is required for buffering data while, for instance, the microcontroller is waiting for the GSM/GPRS module to establish a connection.

The other FIFO is required for buffering data while the external equipment is busy and data is arriving from the GSM/GPRS module. Data is automatically input in both FIFOs. After a byte is received into the UART or the microcontroller's USART, the microcontroller interrupt service routine is executed, processing it and storing the byte in the respective FIFO. As soon as the external device, or the GSM/GPRS module, is ready to receive that byte, the byte is sent towards its destiny.

V. CONCLUSIONS

A flexible and easy to install solution was implemented which enables the serial communication between two remote devices using the GSM network in a totally transparent way for those end-devices. To these, the communication link would be (in terms of hardware and firmware) as if they were connected through a direct RS232 serial cable. Through the developed RSC units, a base computer can be connected to diverse remote devices without having to drastically modify the hardware or software of such remote equipment.

A wide range of configuration settings was considered in the RSC system design so that it can be easily adapted to applications of distinct communication requirements. For example, and to allow an optimization of the costs associated with communications, the GSM mode can be chosen from SMS-based (sporadic and small volume data transfers), CSD-based (sporadic but medium/large volume transfers) and GPRS-based (frequent data transfers).

REFERENCES

[1] C. Petzold, "*Programming Windows,*" Microsoft Press.

[2] B. Miles, "the Forger's Win32 API Tutorial", 1998-2006. Available:

http://winprog.org/tutorial/

[3] M. Shepherd, Available:

http://www.gomorgan89.com/winsock/winsock.html

[4] Wikipedia.org. Available:

http://en.wikipedia.org/wiki/Hayes_command_set

http://en.wikipedia.org/wiki/Circuit_Switched_Data

http://en.wikipedia.org/wiki/GPRS

[5] www.gs-modem.de, Available:

http://www.gsm-modem.de/sms-pdu-mode.html

[6] R. Fosler, R. Richey, "A FLASH Bootloader for PIC16 and PIC18 Devices, " Microchip Technology Inc, 2002