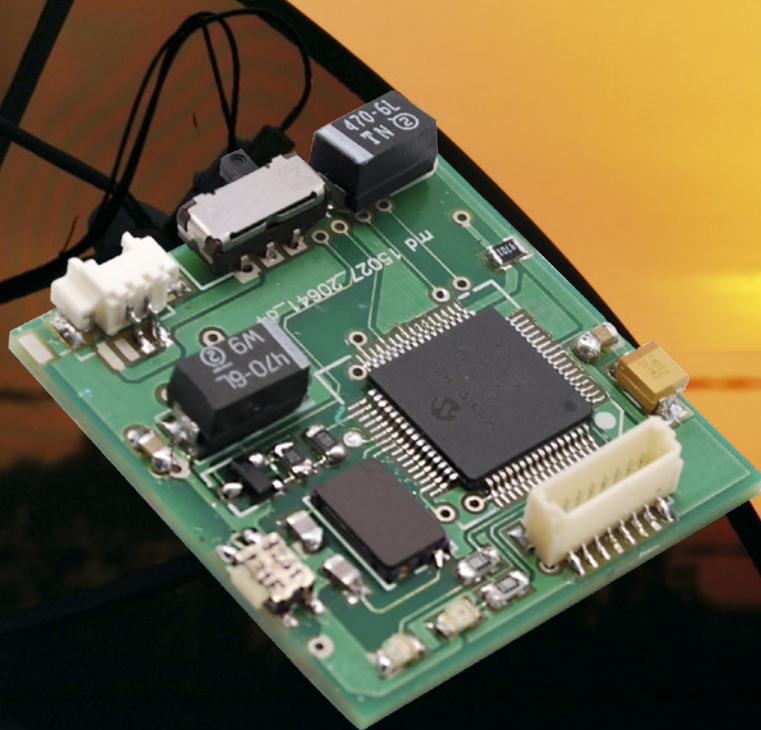


GPS RECEIVER? ... NO LONGER NEEDED!

GSM LOCALIZER

by BORIS LANDONI

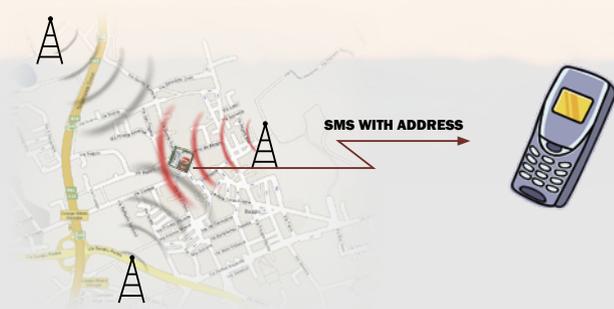
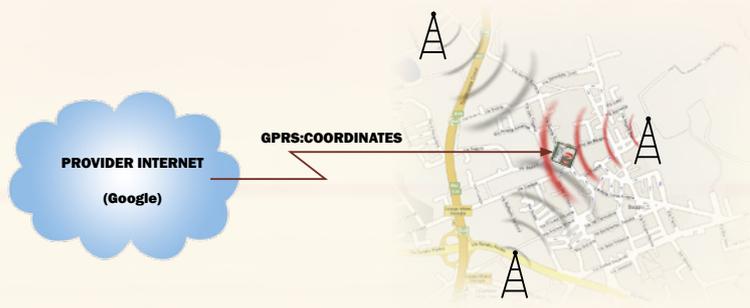
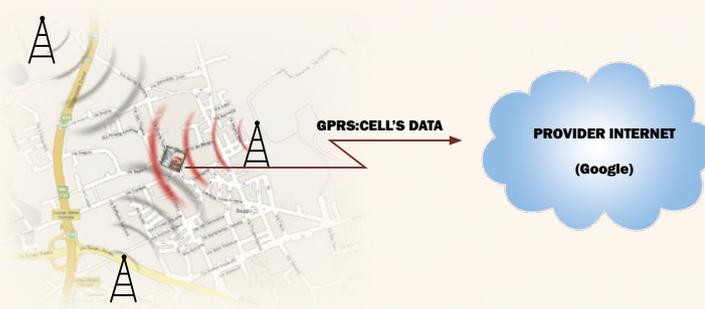
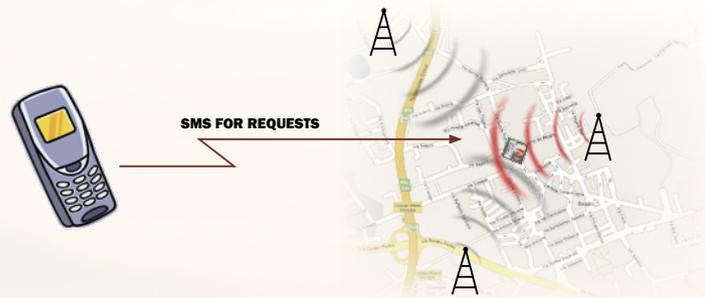
Do you still think you need GPS to know where a person or vehicle is? Well, here is some news for you: by using cellular network data and properly querying Google, even a simple GSM module is able to determine one's position, with a fairly acceptable error margin.



Although so far all localization systems have been based on GPS technology, it is now conceivable to be able to fairly precisely locate a vehicle or person carrying a GSM

cellular device without having to resort to a satellite dish. Is this a miracle? Of course, not! It is simply an ingenious marriage between the information provided by the radio mobile

phone network and the cell coordinate data drawn from Internet portals such as Google Maps. To be sure, GPS is not entirely dispensed with, but this new system allows for localiza-



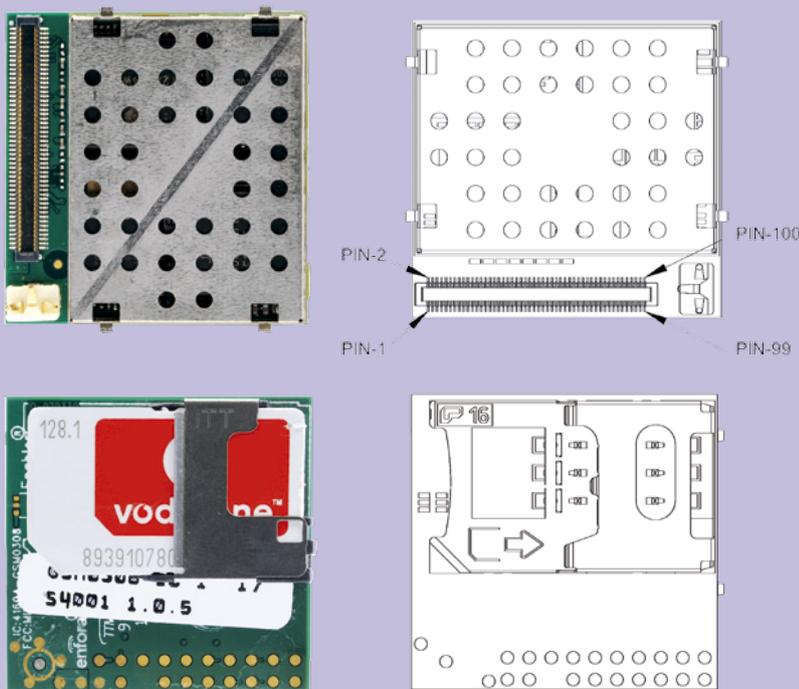
tion without directly utilizing GPS technology; simply put, we are able to locate the desired object fairly precisely by using database availability together with the geographic position of

The GSM/GPRS Module

The module used in our localizer is the Enabler IIIG GSM0308 by Enfora, one of the smallest yet best performing modules of this type currently available. The device, which utilizes a chipset made by Texas Instruments, is but 1.06 x 1.10 inches, is a quad-band that works both in Europe and the US, and has all the functions needed to build M2M devices, including the TCP/IP stack. The technical documentation regarding this device is thorough: hardware and software manuals, as well as numerous tutorials, can easily be downloaded directly from www.enfora.com. Also available for this module is an inexpensive development system. The Enfora module is a GSM/GPRS/Edge device that has everything except loudspeakers, microphone, and keyboard,

which, by the way, are not necessary in our application given that the cell phone is employed for SMS messages as well as to transmit and receive data using the TCP/IP system of the GPRS network. Even the antenna is external: we resolved to connect it to only one pad, where the antenna cable can be soldered. This module consumes very little energy: during transmission, it consumes 230 mA in GSM 850/900 MHz and 175 in GSM 1.800/1.900 MHz; when not in transmission, it consumes only 10 mA (and goes down to when idle, that is, when it is disabled through the PWRCTL line). The antenna's transmitted energy is 2 watts at 850/900 MHz and 1 W at 1.800/1.900 MHz; the reception sensitivity is perfect: -106 dB.

of its clients' cell phones; such data are provided by the phones themselves: once a smartphone connects to the Google service, it sends out both the cell identifier with which it is associated and the geographic location obtained from its own GPS. However, some sites embrace an *open-content* philosophy and utilize information willfully sent by their users in order to create databases such as the one mentioned above. These systems allow them to receive automatically and update the coordinates of the various cells that play a fundamental role in GSM localization. It is important to point out that, unlike in Italy, in some countries, phone companies make those cell coordinates available, which greatly facilitates the entire localization process. Though this is but one hypothesis among others, it may very well be that Google has relied on its Maps Mobile service to obtain the data it needed. Some, on the other hand, claim that the vehicles employed to shoot films to be used for Street View were equipped with cell phones so that the position of cells with respect to the coordinates provided by GPS devices could be detected and recorded. Thus, whenever the phone switched cells, a computer would record their coordinates, which in turn would feed the database. This latter hypothesis may in part be true; however, it is more likely that Google built its archive using its Maps Mobile service. If this is correct, in exchange for a free service, Google used its own clients to obtain information that would have otherwise taken time (simply to go around and record the position of cells) and money to pay for staff and moving expenses or to buy data from phone companies, providing



the cells themselves. Officially, cell coordinates are not publicly known; in fact, they are carefully guarded by those companies that use them for the services they provide. If so, where do we find such data? Who can give them to us, and how? The answer is

very simple. There are numerous Internet portals that have stored the information we need using those very services they provide to their users. For instance, through *Google Maps Mobile*, Google has been able to store billions of data regarding the location

they were willing to sell them. Naturally, there is nothing wrong with this strategy: a similar strategy is used by big supermarkets, which give their most loyal clients discounts in exchange for a membership card through which they are able to determine how much everyone spends and to get precious information that allows them to optimize sales according to geographic area and time of year. A favor in exchange for a favor... But how does GSM localization work? The radio mobile network is made up of a number of adjacent radio cells, each of which is characterized by an identifier consisting of four data: a progressive number (Cell ID), a code related to the area in which that given cell is (LAC, or *Local Area Code*), the code of national network to which the cell belongs (MCC, an acronym for *Mobile Country Code*), and finally the company code (MNC, or *Mobile Network Code*), which obviously identifies the phone company itself. For this reason, once a cell name and coordinates are known, and considering the maximum distance allowed between this cell and a phone before the phone connects to a new cell, it is possible to find out, approximately, the most distant position of the phone itself. For example, if the maximum distance has been determined to be one mile, the cell phone can be within a one-mile radius. It can be deduced that the more cells are found in a given area, the more precisely one can determine where the phone is located (up to 200-350 feet). The idea of employing only a GSM device to build a remote localization system occurred to us when we realized that *Google Maps Mobile*, which had been conceived to allow smartphones equipped with a GPS receiver to

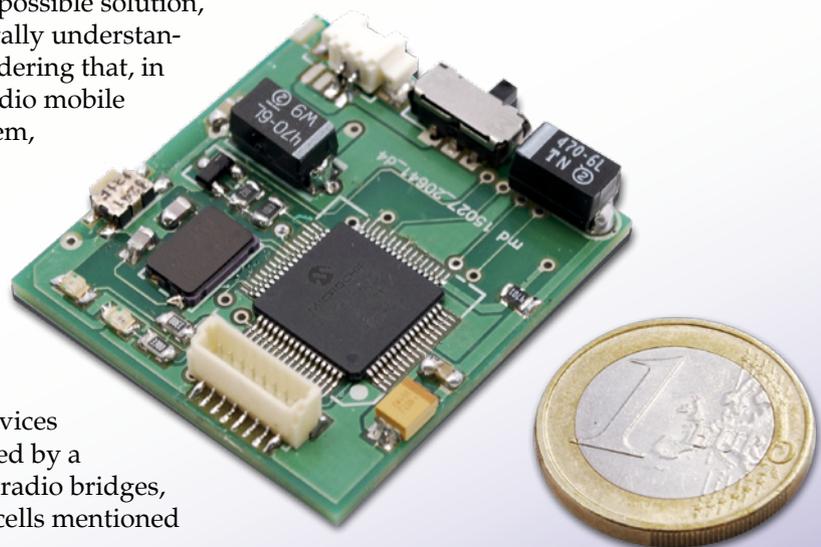
The screenshot shows the OpenCellID website interface. At the top, there's a navigation bar with 'opencellid' logo and links for 'About', 'Stats', 'Download', 'Api', 'Browse cells', 'Cells Map', and 'Raw data'. The main content area is titled 'Listing cells' and includes a note: 'Note: you can get the Kml view here:'. Below this is a table with columns: Lat, Lon, Mcc, Mnc, Lac, Cellid, Range, Nbsamples, Created at, Updated at, Needs computation, Country, and Operator. The table lists several cell entries with their respective coordinates and details.

Lat	Lon	Mcc	Mnc	Lac	Cellid	Range	Nbsamples	Created at	Updated at	Needs computation	Country	Operator
47.2480344772339	39.7961282730103	250	20	26221	2632	0	2	Tue Aug 11 11:15:19	Tue Aug 11 11:22:38	false	Russia Russian Federation	Tele2 / ECC
53.34433	-6.26229	272	1	31	1411640		1	Tue Aug 11 11:15:07	Tue Aug 11 11:07:04	false	Republic of Ireland Ireland	Vodafone Ireland
53.3816614285714	6.35114142857143	272	1	3014	3403440		7	Tue Aug 11 11:15:06	Tue Aug 11 11:06:36	false	Republic of Ireland Ireland	Vodafone Ireland
53.37979	-6.28111	272	1	3014	36479	0	1	Tue Aug 11 11:15:05	Tue Aug 11 11:05:38	false	Republic of Ireland Ireland	Vodafone

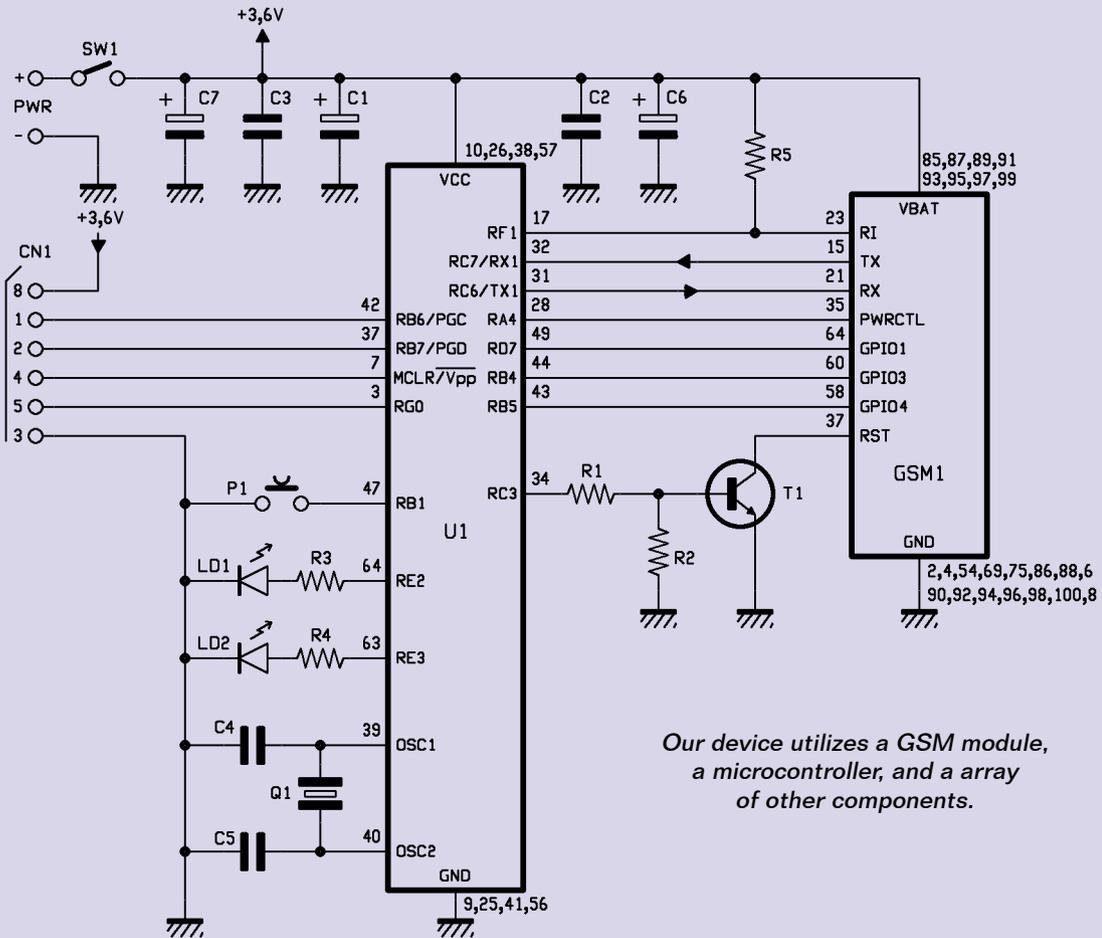
Browse Page of our site www.opencellid.org.

use Google for satellite navigation, was extended to all cell phones, as long as they were able to support GPRS or UMTS data. That realization made a light bulb go on, and not only in our heads. Indeed, many people started wondering how on earth Google was able to tell someone whose phone was devoid of a GPS receiver where he or she was at that particular time. Given that the technique in question has inflexible rules, we thought of the only possible solution, which is totally understandable considering that, in the GSM radio mobile phone system, the radio connection allowing for phone calls and access to data and Internet services is maintained by a network of radio bridges, that is, the cells mentioned

above. When a cell phone is on, it connects to the cell that allows for the best possible connection, but also detects signals from *neighboring cells* so that the phone can switch to a new one when the cell being used starts performing poorly due, for example, to the fact that the user is moving. The communication message between a cell phone and a cell contains the cell's identifier, sent out by the cell itself; every



[Localizer SCHEMATIC]



Our device utilizes a GSM module, a microcontroller, and a array of other components.

identifier is unique. This entails that every cell phone, when connected to the network of a given network, has specific information about the cell it is relying on. We, therefore, concluded that *Google Maps Mobile* is able to track the position of a cell phone by simply associating the name of

a given cell with the data regarding its position; in other words, a cell phone transmits the data of the cell it is connected to via the GPRS network, and the *Google Maps Mobile Server* associates such data to the position of the cell itself. While working out the details of our project, we were still testing the ground, given that Google not only is disinclined from publishing its

data on the web, but also refrains from explaining how to use them. So, what we did was connect a USB-interfaced GPRS cell phone (on which we had uploaded *Google Maps Mobile*) to a computer which, in turn, was connected to the web via LAN. Using programs such as *Microsoft Network Monitor*, we sniffed the communication between the cell phone and the Google server and realized that our hypothesis was indeed correct: Google knows the position (or coordinates) of the cells of the radio mobile system and can even state where a cell is if provided with that cell's ID. Once the position of a cell is known, and considering that the coverage radius for each cell is also known, it is easy to find the circular zone, whose radius equals the coverage



radius, where the cell phone requesting location tracking is located. Naturally, this method allows but for a rough estimate: determining the precise position of the cell phone hinges on data

and slimmer, since it lacks one of the pieces of the other phones. It is also less expensive, and most important, it lasts longer, given that it does not have a GPS receiver, which is the component

people engaged in developing cellular localization, several websites have popped up where one can find a large number of data regarding the position of the cells available for free. And there are

[board LAYOUT]

Parts List:

R1, R5: 4,7 kohm (0805)

R2: 10 kohm (0805)

R3, R4: 330 ohm (0805)

C1: 100 μ F 6,3 VL tantalum (CASE-B)

C2, C3: 100 nF multilayer (0805)

C4, C5: 10 pF ceramic (0805)

C6, C7: 470 μ F 6,3 VL tantalum (CASE-X)

U1: PIC18LF6722 (MF833)

LD1: red LED (0805)

LD2: green LED (0805)

Q1: 20 MHz quartz (12SMX)

T1: BC817

SW1: Slide switch

GSM1: GSM0308

P1: Microswitch 90° SMD

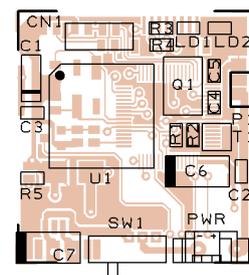
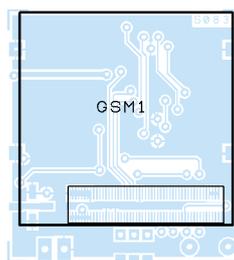
Varie:

- CS connector 100 pin for GSM

- 8 pole connector 90°

1 mm

- PCB



regarding the coverage of a given cell which can only be provided by the Google server.

OUR PROJECT

Regardless of how Google has compiled its database, once we discovered how cell phones using *Google Maps Mobile* send their requests, we immediately thought of employing that functionality to build a specific remote localizer, which, unlike those that have been proposed so far, relies solely on GSM/GPRS for its radio connection, rather than use a GPS receiver. Well, here it is! This article describes our new localizer, which has numerous advantages over those using a GPS receiver, though it does not guarantee the same precision as they do. For one thing, our localizer is lighter

that consumes the most energy. This means that one 1-Ah lithium battery, for instance, allows our localizer to work for several days (depending on the number of calls), while a 13-Ah battery will make it work for a couple of months. In addition, a localizer based on a cellular network is likely to pick up more quickly than a conventional phone: there are times when a GPS receiver may need a few minutes to determine its own position, whereas cellular network data allow for a speedy answer to the call to the Google server. What's more, our cell phone has reception even in many highway tunnels and basements where a GPS receiver has no reception whatsoever. Before looking more closely at the localizer circuit, it is important to note that, due to the work of the many

also services available at a charge; one such service is provided by OctoTelematics (www.octotelematics.it), which features a carefully guarded database known for its thoroughness. Among the free websites we have found, two are worth mentioning: www.opencellid.org and www.cellldb.org; in both of them, you can not only find valuable information, but also check the position of a cell by simply entering its identifier (Cell ID). Bear in mind that you can also manually get the identifier from the cell phone itself: All you need to do is install programs such as *Nokia Net Monitor* or *WM Cell Catcher*, a Windows Mobile service pack we have used to verify the data we sniffed during the communication between our cell phone and the Google server. Notably,

You would be well advised to lend an ear ... connecting to Google

In order to be able to associate a GSM cell with its geographic coordinates, we utilize a Google Maps Mobile function, which is not documented, though. Before doing this, however, we need to know which cell the remote GSM device is connected to, that is, we need information regarding MCC, MNC, LAC, and CID. To obtain the data we need, the micro-controller must send the following commands to the Enfora module:

AT+COPS? to obtain MCC and MNC

AT+CREG? to obtain LAC e CID

Recall that during the setup stage of the GSM module it is necessary to program Enfora in such a way that it can respond to questions such as the two immediately above. To this end, PIC needs to have sent the module the following commands:

AT+COPS=0,2

AT+CREG=2

Once the data regarding MCC,MNC,LAC and CID have been obtained, the remote device can send Google (www.google.com) a request, using the function HTTP POST. Thus, what one needs to do is first connect to Google and then send a request with the following procedure:

POST /glm/mmap HTTP/1.1

HOST: google.com

Content-Type: application/binary

Content-Length: 82

Connection: close

The relevant string consists of 82 characters which must contain the cell's data, among other data:

21	2 byte	2
0	8 byte	8 10
2	2 byte	2 12
it	1 byte	2 14
13	2 byte	2 16
Boris Landoni	1 byte	13 29
7	2 byte	2 31
3.0.1.6	1 byte	7 38
3	2 byte	2 40
Web	1 byte	3 43
27	1 byte	1 44
0 0 3	4 byte	12 56
0	2 byte	2 58
cid lac mnc mcc 0 0	4 byte	24 82



It is essential that the length of the data be adhered to; for instance, header Content-Length: 82 explicitly specifies how many data are to be sent : if the string length should be different, the server will reject the request. After processing the request, the server will respond with a string containing the cell's geographic data (if that is in the Google databases, obviously) as well as the approximate position of the remote device, that is, its maximum distance from the point in question:

2 byte	a
1 byte	b
4 byte	c
4 byte	latitude
4 byte	longitude
4 byte	approximation
4 byte	d
2 byte	e

While testing the device, we were not able to determine with certainty what the data a,b,c,d, correspond to, in part because, as was mentioned above, Google fails to document such functionality; our conclusions are thus based solely on our sniffing the data exchanged between the Google Maps Mobile cellular application and the Google server. We encourage others to look further into this issue ...

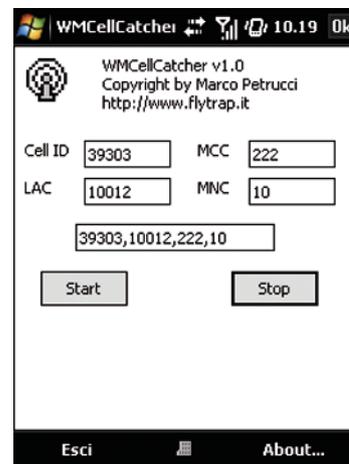
opencellid.org makes APIs available; some of them allow users to download information on cells needed for localizing purposes, while at the same time providing the site itself with data, drawn from those users, on the position of new cells. For instance, when

a GPS cell phone connects to a cell, its ID code and position can easily be obtained and transmitted to the site, which, in turn, will verify and publish them in behalf of all site's visitors. The same site contains a thread that, utilizing Google Earth, shows how many

and what cells are found in each area, distinguishing them by MCC and MNC. Regarding company codes, keep in mind that in Italy MNC 1 identifies TIM, 10 Vodafone, 88 Wind and 99 Tre (H3g). The file can be downloaded by clicking on Google Earth

in the "Browse Cells" section of www.opencellid.org. However, one needs to first download and install Google Earth from www.google.com; once the plug-in has been installed, the heading "View Cells" is displayed in the Google Earth Places panel, under "Temporary Places." At any rate, while developing our application, we relied on data provided by *Google Maps Mobile*, given that, unlike other sites, Google provides information regarding not only the conjectured but also approximate position of a cell phone; in other words, it indicates how far a cell phone can possibly be in relation to any given point. Among others, the closest possible position is indicated. Note that at the moment Google does not employ the triangular method, that is, the capability of obtaining data from neighboring cells, which would allow to determine a given position more precisely, nor does it verify the strength of the cell signal, since that could

weather conditions and even radioelectric disturbances. Our localizer runs on batteries, so it can be carried by those who feel they may need to call for help or to be easily reachable by others; however, it can also be used in vehicles (without having been installed) or simply be part of moving goods. In a nutshell, it can do whatever a localizer can do, short of telling us which side of the street it is on; yes, that is something it cannot do, though it can definitely help us figure out whether a driver is going the right direction or is ending up in a different neighborhood in some town nearby. In order to save battery life, we have set up our localizer to provide its position when requested via SMS, by means of a call or periodically; we have also implemented an SOS function, which can be activated by pressing a button: an SMS requesting help, and containing the coordinates of the relevant position, is immediately sent to



WM Cell Catcher allows one to find out the data of the cell the phone is connected to.

spend money. When asked, the localizer will send an SMS with the current localization result. When in auto-report mode (that is, sending periodic notifications), it sends SMS messages with localization data to one or more of the stored numbers, depending on the user's settings. Please see the drawing for details of how it works. Basically, if we want to find out the position of the remote device, we must send it an SMS. Once the message has been received, the device verifies the data of the cell it is connected to and sends Google a request via GPRS; in turn, Google will send back the coordinates together with data regarding how precise the information is. As soon as the device receives this information, it forwards it to us by means of another SMS. All of this takes but a few seconds.

LED Signaling

OPERATING CONDITION	SIGNAL
Start-up	10 rapid flashings of both LEDs
GSM Setup	Solid green LED
Waiting for easy setup	Solid red LED (starts flashing when getting a call)
GPRS connection	Both LEDs are solid
Sending an SMS	Solid green LED
Pressing SOS button	Solid red LED; then a sequence of messages (GPRS -> SMS message)
Receiving a call	Red LED lit; then a sequence of messages (if the number is one of the authorized numbers)
Receiving an SMS	Red LED flashes rapidly 5 times
Stand by	Green LED flashes continuously

be misleading. Indeed, determining the distance from the cell on the basis of how strong its signal is does not necessarily provide accurate information, as the reception can be hindered by a number of factors, such as

the number that has been stored. More specifically, up to eight numbers can be stored from phones that can send out SMS messages; to this end, a phone call will do, too, although this call won't be answered in order not to

ELECTRIC SCHEME

The circuit our localizer is based on consists only of a quad-band Gsm/GPRS module interfaced with a microchip microcontroller which, after initializing the I/Os and UARTs it has at its disposal, puts the main program in a loop waiting for some event,



Going the Right Direction

Relatives and nurses dealing with those affected by Alzheimer's can now feel more secure: in the United States, GTX and Aetrex Worldwide (a shoe manufacturing company) have created shoes equipped with a GPS localizer, through which patients can be found in the event they should leave their houses and start wandering aimlessly. This is great news, given that Alzheimer's is a disease that causes patients to gradually lose their memories, often producing such mental confusion and disorientation that they may be able to neither find their ways back home nor ask for help once they have left their houses for a simple walk. Indeed, 60% of those affected by this condition have gotten lost at

least once. This new technology makes it possible to locate a patient regardless of where he or she may have gone, up to about 30 feet away. The technology was developed by GTX Corporation utilizing GPS micro-receivers incorporated into one of the shoes. The device is mounted inside the heel of a shoe, rather than in a bracelet or pendant, because those affected by Alzheimer's don't like to wear objects they are not familiar with, so they take them off as soon as they can. In the United States, these shoes cost between 200 and 300 dollars, plus a monthly rate of about 20 dollars for monitoring services. These shoes are now being tested and should become available in 2010.

which could be the arrival of an SMS or someone pressing the button P1 (that is, the logical 1/0 transition of the RB1 line, internally equipped with a pull-up resistor). In the event that an SMS is received, the localizer distinguishes between a message that is setting up a configuration and one that is requesting information regarding the position. Let us look more closely at the procedure that allows to find out the position, which is essentially the same in the case of manual requests as well as when pres-

sing P1 (alarm or SOS): as soon as a request is received, the PIC orders the cellular module to connect to the Google Maps server in data mode (via GPRS, hence on the Internet) and to send a position request along with the cell ID to which Enfora is connected; now the localizer is ready to receive Internet-transmitted data on the RX channel of its own UART. Once the localizer receives the data with conjectured position (latitude and longitude, approximate street or square location) and accuracy, it prepares

an SMS and sends it to the number that requested it or to the one that is stored and associated with this particular alarm function. That is all: simple and functional. The cell phone is handled by the microcontroller by means of the RF1 line (through which it reads the RI, that is, the Ring Indicator used by the module to signal the arrival of a call), as well as RB4 line, which allows it to detect receipt of SMS messages. RC7/RX1 and RC6/TX1 are, respectively, the reception and transmission lines of the UART, which is used not only to handle the cell phone's functions (except for reset and power), but also to read and send SMS messages. The power function is handled by the RA4 line, which relies on PWRCTL to turn GSM1 on and off and which is essentially used to start the cell phone after initialization, that is, to turn it off and on again in the event it should get stuck and the reset function should not be enough to set it back to normal. The reset function is handled by the PIC through the RC3 line, which relies on the T1 transistor, employed here as a static switch that effectively changes the module's RST.

This compound unit (that is, the cell phone and microcontroller) is powered through the SW1 switch by a continuous 3.6-volt tension, applied to both + and - PWR, which can easily be obtained from a 1.2-V NiMh AAA rechargeable battery or from a 3.7-V Li-Ion battery. Notice how many filter condensers are placed along the power line. They are needed to filter out disturbances produced by the input when the cell phone is transmitting -disturbances that would otherwise create enough power fluctuations to block the microcontroller.

Functions and Commands

FUNCTION	SMS COMMAND	PARAMETERS	DEFAULT VALUES
CHANGE PASSWORD	PWDxxxx;12345	xxxx=new password	12345
STORING A NUMBER (up to 8 numbers) (UP TO 19 DIGITS PER NUMBER)	NUMx+39nnnnnnnnn;12345	x=position of number nnnnnnn=number to be stored	
ERASING A NUMBER	NUMx;12345	x=position of number	
CHECKING STORED NUMBERS	NUM?;12345		
TOTAL RESET	RES;12345		
SETTING NUMBERS SMSs ARE TO BE SENT TO (FOR AUTO-REPORT / SOS / POLLING)	SMSxxxxxxx;z	xxx=positions of numbers to be notified z=can be either ON or OFF	All numbers will be notified
REQUESTING IMEI	IMEI?		
SETTING NAME	NAME:xxxx	xxxx=name of localizer (up to 15 digits)	
REQUESTING NOME	NAME?		
SENDING NOTIFICATION WHEN REQUESTED THROUGH A RING	ARI:x	x=M calls only x=S all enabled users	S
SELECTING COORDINATE FORMAT FOR SENDING SMS'S	FORS:x	See text	
REQUESTING FORMAT	FORS?		
CONTINUOUS AUTO-REPORT	AUTOC:z	z=can be either ON or OFF	
TIME INTERVAL FOR SENDING NOTIFICATIONS (CONTINUOUS)	AUTOC:hh/mm	hh=hours mm=minutes	
REQUESTING AUTO-REPORT SETTINGS	AUTO?		
SETTING APN	GPRSAPN:apn	apn=apn of provider (e.g., web.omnitel.it)	
ERASING APN	GPRSAPN		
SETTING GPRS USERNAME	GPRSUSR:xxxxxxx	xxxx=username (usually not necessary)	
ERASING USERNAME	GPRSUSR		
SETTING GPRS PWD	GPRSPWD:xxxxxxx	xxxx=password (usually not necessary)	
ERASING PASSWORD	GPRSPWD		
CHECKING GPRS SETTINGS	GPRS?		
DISABLING RESPONSE TO A MULTIPLE MESSAGE	RISP		
REQUESTING COORDINATES	COO		

BUILDING THE LOCALIZER

Our localizer is definitely simple, yet it must be built with care, as all of its components are SMD. Once the two-sided board is ready (the two tracks can be downloaded from our site www.elettronica.in in TIFF format), be sure you have a very fine-pointed welder of up to 20 watts and a 0.02-inch-diametered (possibly, even smaller) welding thread. First of all, take care of the microcontroller; solder a couple of stems opposite one another in order to stabilize it and then

solder the rest of the stems. Then go on to the passive components, quartz and transistor. For the GSM/GPRS module, there is a small high-density connector to be soldered on the printed board, possibly with the aid of a magnifying lens to make sure there are no solder "whiskers" short-circuiting neighboring areas. On the mini-module just inserted you can attach and connect the SIM-card holder. Cellular antenna connection involves a triple golden spring contact which, pushing the component all the way

into its connector, touches a pad purposely created in one of the corners of the c.s.; there one can solder the antenna cable or, depending on its shape, the antenna itself. For our model, we simply used a thread fragment about 3.15 inches long as an antenna; however, depending on the space available, one can choose a cellular antenna, even a printed-board one. Once everything has been assembled, one can proceed to the in-circuit programming of the microcontroller through the CN1 connector (8 poles). An

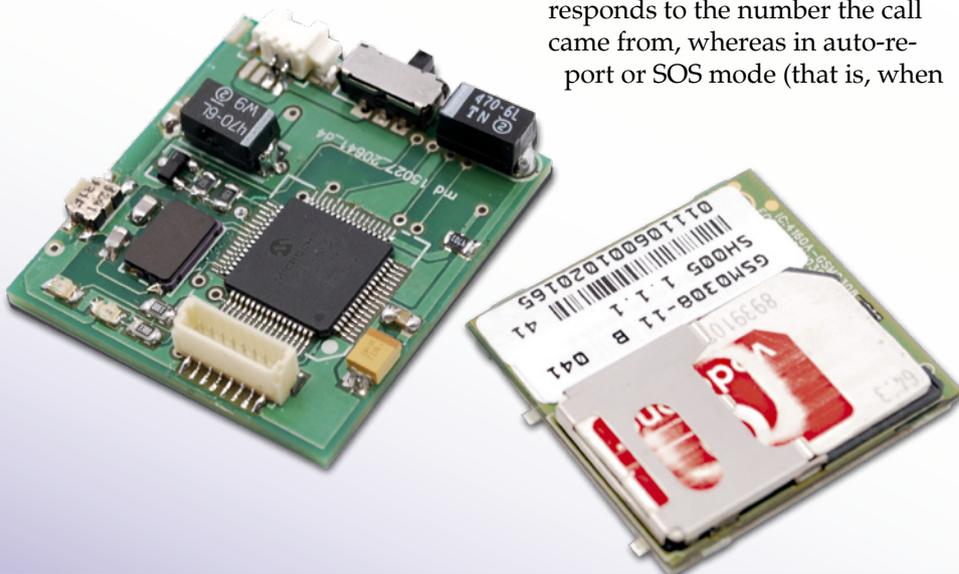
additional two-pole miniature connector must be mounted for battery connection.

SETTINGS AND COMMANDS

All configurations can be set remotely using any cell phone; what follows is a description of how settings and commands work. Let us start with the list of numbers that can be used to request location data: there can be up to eight numbers –the same numbers to which the localizer sends SMS messages containing location data. Before continuing, let us emphasize that some of the commands can be executed only after entering a password; others can be executed as long as they come from one of those eight phones that have been enabled to deliver commands and request coordinates. The default password (automatically reset whenever the system is reset) is 12345; the default password can be replaced with a private one (provided it is not more than five digits) by sending a command with an SMS containing "PWD *newpwd*;pwd," where "*newpwd*" is the password to be stored and "*pwd*" the current one. Having said that, let us

look at the commands regarding the phone number list: in order to store a given number, one must send an SMS containing "*num**x**nnnnnnnnnnnnnn*;pwd," where *x* indicates the position (1÷8) in which to save the number represented by "*nnnnnnnnnnnn*" and "*pwd*" is the current password, which must be entered if the SMS message is coming from an unlisted phone (though it is always necessary when the first number is stored...) or, even when the phone is listed, if the specified location is momentarily busy with another number. Each number can contain up to 19 digits and must include the country code (+39 for Italian cell phones). To erase a number from a position in the list, one must use the command "*NUM**x*;pwd," where *x* is the position (1÷8) and "*pwd*" the current password. It is also possible to find out what the stored numbers are; all one has to do is send an SMS containing the command "*NUM?*;pwd." Once the numbers have been saved, one must determine to which ones the localizer is to send SMS messages containing the coordinates of a given position. As was mentioned above, when a call is made, the localizer responds to the number the call came from, whereas in auto-report or SOS mode (that is, when

the alarm button P1 is pressed), the localizer can be programmed to respond to all eight numbers. The command, in this case, would be "*SMS**xxxxxxxx*;ON," where "*xxxxxxxx*" stands for the positions where the numbers are. For example, "*SMS*1346;ON" means that those numbers in positions 1, 3, 4 e 6 will be receiving the sent SMS messages. In order to disable sending SMS's to one or more numbers, it is necessary to send the command "*SMS**xxxxxxxx*;OFF," which tells the localizer to disabled the positions specified. For instance, "*SMS*2,3,4,6;OFF" disables sending SMS's to those numbers stored in positions 2, 3, 4, 6. By default, the first time a phone is turned on or after it is reset, as well as in auto-report or alarm mode, the localizer will send SMS's to all the numbers that have been stored. Regarding requests through a mere ring, it is possible to program the localizer's response through the command "*ARL**x*," where *x* should be replaced with *M* if you want the device to respond only to the number the call came from; on the other hand, if you would like all the numbers enabled with the command "*SMS**xxxxxxxx*;ON" to receive the SMS, then *x* should be replaced with *S*. Now let's take a look at those commands that define the localizer's IDs: an SMS containing the text "*IMEI?*" sends the number the call came from the phone's IMEI code. The command "*NAME*:*xxxx*" allows a user to name the device, which comes in handy when dealing with more than one localizer. "*NAME?*" is what one should ask the device if he or she wants to find out its name. All these commands can be sent by the numbers in the list, as well as by any other phone providing that



a password is entered. "FORs:x" determines how SMS messages are displayed. The two configurable formats (1 or 2) yield the following results:

1:

Probable position:

FT833

LAT: 45.643851

LON: 8.814346

Range: 1159 m

Gallarate VA, Italy

SOS

2:

<http://maps.google.it/maps?f=q&hl=it&q=45.643830,+8.814598>

Gallarate VA, Italy

SOS

Through the command "REV:1," one can also view the information regarding the conjectured address, whereas REV:0 doesn't allow for that. In the examples you see SOS because the SOS button was pressed remotely when the message was sent; if the request is made by calling or by means of an SMS, the following is displayed:

POLLING

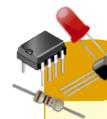
or

AUTO_CONT

when in auto-report mode. The system can be queried at any time using the commands "FORs?" in order to find out which format is currently configured. Regarding the auto-report mode, there is a specific set of commands: "AUTO?" allows a user to remotely check auto-report settings. "AUTOc:xx" enables or disables the periodic sending of coordinates to the list of numbers stored or enabled to receive SMS's. In order to enable

auto-report, write ON in the place of xx; to disable it, write OFF. The command "AUTOc:hh/mm" allows one to determine how often periodic sending is to occur; to this end, you need to "hh/mm" with the number of hours and minutes the localizer is supposed to send messages with its position to the numbers in the list with the usual command "SMSxxxxxxxx:ON." For instance, if a message must be sent every 15 minutes, you need the following command: "AUTOc:00/15." Let's now turn to the set of commands regarding Internet settings, indispensable when accessing the Google server during location requests: APN is set with "GPRSAPN xxxxxxxxxx," where the x's must be replaced with the parameter associated with the company providing cellular Internet service; for Vodafone, it would be "web.omnitel.it." When needed, "GPRSAPN" erases the existing setting. "GPRSUSR xxxxxxxx" can be used to set the username (which needs to be written in place of the x's), while "GPRSUSR" erases the current setting. "GPRSPWD xxxxxxxxxx" sets the password needed to access Internet, while "GPRSPWD" erases the current password. Finally, those who wish to check their Internet settings can use the command "GPRS?," which will respond by sending an SMS containing the relevant data to the number that requested them. Recall that our localizer responds to commands that come from one of the numbers in the list or from any other phone as long as the SMS contains the correct password. The command "COO" allows users to request the coordinates of the current position at any time; this command causes an SMS to be sent to the phone the command itself came from.

Total reset, which restores default settings and erases the phone number list, is achieved through the command "RES;pwd," where "pwd" is the current password. The system can handle multiple messages, that is, SMS's containing each more than one command, as long as such commands are separated by commas. In the event one wishes to save money, he or she can disable answering some of the commands by inserting in the SMS the command "RISP," which blocks commands that need an SMS confirmation from receiving answers. Our localizer can automatically store the first number in the list (Easy Setup): once it has been turned on and for about three minutes, it waits for a call; when the call arrives, the localizer saves the number of the phone the call is coming from and then starts functioning normally wherever it is, even if no one else calls. To go back to Easy Setup, it is necessary to switch the power off and then on again. ■



WHERE TO BUY

All the components utilized in this circuit can be easily found on the market. The printed circuit master board can be downloaded from the site of our magazine (www.elettronica.in), and so can the firmware needed to program the microcontroller. Our localizer is also available fully assembled and tested (cod. FT833M) for 105.00 Euro, including tax.

Please send your order to:
Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)
Tel:+39-0331-799775 • Fax:+39-0331-792287
<http://www.futurashop.it>