
Remote control circuitry via mobile phones and SMS

Gunther Zielosko

1. Introduction

In application note No. 56 (“BASIC-Tiger® sends text messages”, in which we described a BASIC-Tiger® sending text messages from one mobile phone to another as an alarm system) we already promised: Now we will use this universal technology vice versa. We send a text message to a mobile phone in any distance, which will receive the text message and will implement the commands with BASIC-Tiger®, depending on the content of the message. Like this we are able to construct remote controls which are practically unlimited regarding distance and universality. Just imagine an application, which turns on the heating system of your holiday home long before you arrive or if frost is being forecasted. Similarly you can preheat your car equipped with a car heating system no matter where you parked it. Many applications can be found in the business world, too. E.g. hotels can use this application for updating their advertisement panels at the city borders (“vacancies” or “no vacancies”). Water resources management could open distant weirs. When great distances do not allow independent wire connection or radio communication, our simple SMS remote control is the first choice. In addition it is possible for different users to access control functions, provided that the number of controlling mobile phone and the stipulated text messages are known. In application note No. 056 we discovered many hints how to apply a mobile phone’s hardware and software to a BASIC-Tiger®. Please see No. 056 for details. Here we also use a Siemens S25 as a guinea pig due to the following reasons:

- First of all the S25 is the standard mobile phone for data transmission. You can find plenty of literature, control hints and it lacks the gimmicks of modern mobile phones.
- Second it was probably produced in large quantities and now a large number of this version is unused.
- Third the author uses the S25 now and then, because he has many data wires and solderable plugs at his disposal which you need in development.
- Even if you do not possess a S25 you can use the below presented strategy. Different mobile phones use the same commands or even the same hardware. You do, however, need a mobile phone which is capable of exchanging data and provides a serial interface. Depending on the version you need to know if serial communication uses RS232 or TTL levels, what your data cable does etc. If you know about the connection conditions of your mobile phone, you should be able to solve almost every problem.

2. Hardware

Principally this procedure resembles application note 056, since we will use a simple Economy-Tiger®, which works serially with TTL levels and therefore can be connected

almost directly to the S25. We use pins L80 to L82 as switching outputs, port 6 (with corresponding control lines) functions as a channel for an optional data output via LCD. In contrast to note No. 056 SER1 is a serial connection to the mobile phone, SER0 optionally displays the mobile phone reactions via a terminal PC software. This feature requires the adaption of levels to the RS232 signal levels (see figure 1). Please do not use the hardware from application note 056, since there the serial interfaces are used vice versa!

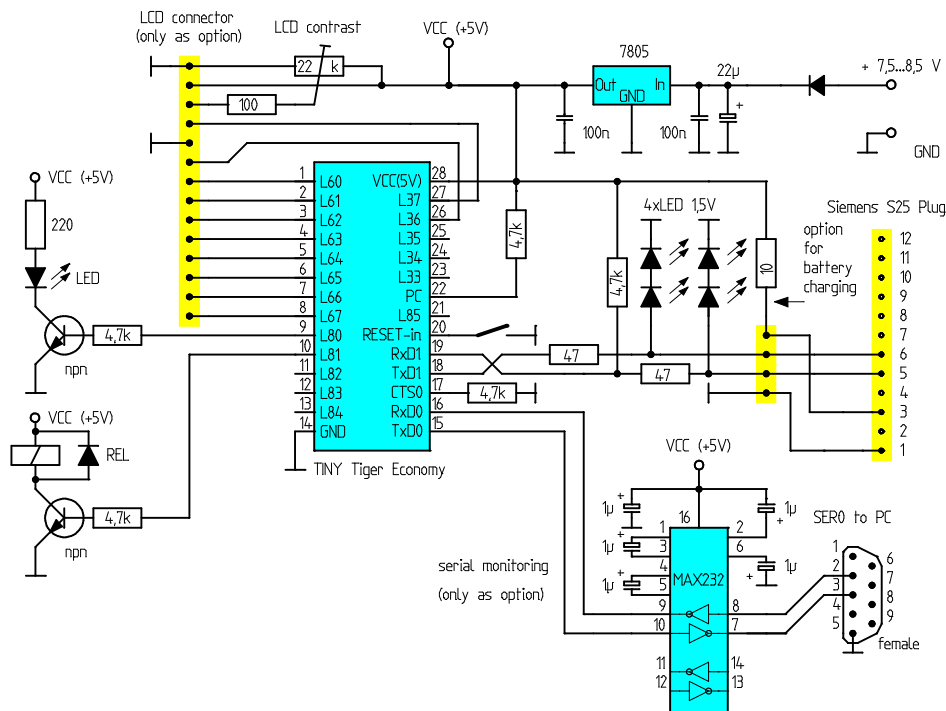


Fig. 1 Circuitry

You can leave out the circuit of SER0 (MAX232, SUB-D socket), the LCD connector including both resistors as well as drawn in, as an example for a LED or relay stage, components on L80 and L81, without impairing the fundamental functionality. The circuit and the appropriate software will still evaluate messages received via SMS.

Subsequently we will shortly explain the components:

TxD1: the Economy-Tiger[®] transmission exit outputs TTL signals on a 5V level (can be attached to pin 6 of the mobile phone plug - figure 2). The phone, however, uses 3.3 V levels. That is why measures to limit levels need to be taken. The 47 Ω resistor is supposed to limit the mobile phones voltage (high ca. 3.3 V!) and current with a series connection of the two LEDs.

RxD1: Of course the mobile phone only sends 3V levels to the plug connector. This will not be adequate to control the Economy-Tiger[®] properly.

Therefore we use a pull-up resistor (4.7 k Ω on +5V). For safeness a voltage limit on 3.3V with two LEDs is provided.

Rechargeable battery: Pin 3 of the S25 plug serves as a recharge input of the mobile phone. The manufacturer likes to see 6.1 to 8 V with currents about 1 A there. If you can ensure that the raw voltage in front of the regulator remains stable in that range you may place pin 3 there. This elegantly allows an unattended continuous operation of our remote control system. Possibly a sufficiently powerful VCC in the BASIC-Tiger[®] environment will be able to recharge the phone's batteries at 5 V. Since the mobile phone needs a few 100 mA to recharge an empty battery, power supply should be designed generously. 10 Ω again serve as a "fear" resistor. If you plan to do first experiments without recharging the batteries, just leave pin 3 of the plug unoccupied.



Fig. 2 S25 and standard Siemens plug

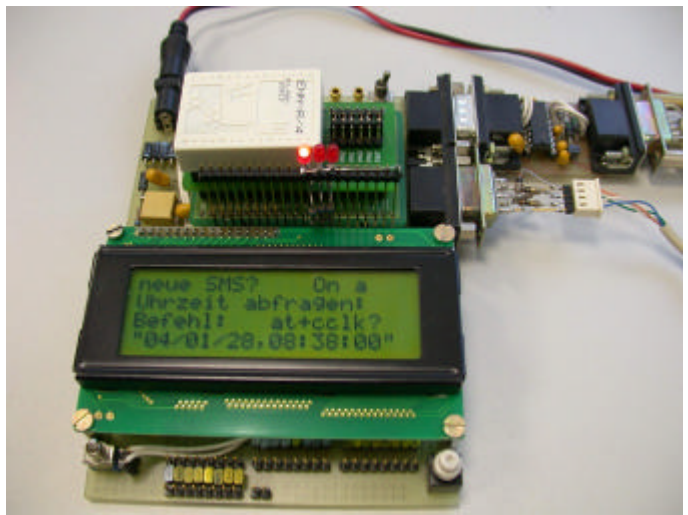


Fig. 3 The author's breadboard construction

Just some more words about the switching outputs. The outputs of Economy-Tiger[®] carry a level of about 5V when ON and about 0V when OFF at a driving capacity of a few mA. For switching tasks requiring more power, e.g. on the 220 V network, appropriate measures have to be taken to ensure potential isolation (danger for human-beings), safety against short-circuits (fire) and safety against retroactivities from the power gates on the electronics (transients etc.). Those applications should be carried out only by power electronics experts. A suggestion for laymen: Use for example radio controlled remote controls for power sockets from the DIY store. The transmitter works on batteries and is electrically isolated from the mains supply. Small relays at the Economy-Tiger[®] bridge the corresponding remote control keys and so easily switch consumers on and off (also see application note 026).

3. Evaluating a text message

We already know the problem – a SMS transmission is not just a transmission of texts. In application note 056 we had to deal with transforming a text into the PDU format, tedious work... And also an incoming text message leaves us with the problem of dealing with this cryptic kind of communication. At least with the Siemens S25 we will receive a message in PDU format with all its bits and bytes transformations. Bad luck – isn't it? If you just want to operate a few switching channels, like the author, you can use an ingenious trick to avoid this problem: We don't really want to read the text message, we just want to decide upon the message content when to put which output to L or H. This is how we can simplify the evaluation considerably. The following example should make the procedure clear:

We will send a text message to turn a consuming unit on or off like this:

On a turn on consuming unit "a" at pin L80 (LED in circuit diagram)

Off b turn off consuming unit "b" at pin L81 (relay in circuit diagram)

The mobile phone transforms this simple text into a complex PDU which also contains the receiver's mobile phone number, message length information and more. Finally the PDU text is transformed by an extensive calculation from 8 bit into 7 bit representation and is converted by redistribution of bytes. You find details in application note 056 and there published links.

In the underlying case the PDU representation looks as follows:

On a becomes **00010000810000044F37280C**

If the receiver's phone number is a different one it converts into:

0001000E91946110325476980000044F37280C

Off b becomes **00010000810000054FB3192406**

In case of a different number:

0001000D91946110325476F80000054FB3192406

Looking carefully you notice that the last bytes of our message are identical, meaning we normally do not need to know the whole content of this cryptic message. We just need to compare the last part of the PDU with a given message, using a few simple string commands:

If the right part of the PDU string equals "**3192406**" the consuming unit **b is turned off**. Using the PC software "PDUspy" that you can find at

<http://www.nobbi.com/download/pduspy.zip>

or our construction (via SER0 and level converter) we can look at text messages in PDU format and design the BASIC-Tiger® evaluation software accordingly. Of course you can recalculate the original text of the message, which makes you independent in the choice of texts, you should, however, keep the decoding simple. Test what your mobile phone turns the given messages into and use the last 7 characters to verify the control command. Pay attention

to a correct spelling, because even a (invisible!) space can disturb the PDU string. Disturbances can also occur in text messages which are sent via the Internet or other non-telephone-connections. This is because of attached advertisements or other. Our simple decoding mechanism would be swamped in such a case. If necessary the incoming and on SER0 in PDU format displayed message has to be used as an (maybe alternative) comparative string.

4. HANDY_02.TIG software

The rest is simple. According to the “HANDY_01.TIG” software (application note 056) AT commands are sent to the mobile phone before the retrieval of reaction. When your construction is finished and you have connected the S25, start the also delivered software”HANDY_02.TIG”. If you have nothing connected but the mobile phone (no display, no RS232 level converter), your S25 should at least ring once – a sign for working communication. Since especially the ring command AT^SRTC is specific for Siemens, it can fail with other mobile phones. If your phone has a different command for ringing, you should adapt the software command. With a connected display and/or terminal program on SER0 you can observe all sent and received messages. Afterwards the software starts an infinite loop and checks the phone on new text messages every 15 seconds (AT command AT+SMGL=1). The mobile phone will answer with either OK or by displaying the message in PDU format. If this is the case, all unneeded characters (e.g. linefeed, OK etc.) are deleted, the message’s content is tested and the according operation is initialized. In order to avoid the phone’s memory to overflow, the commands AT+CMGD=1 to AT+CMGD=4 are used to delete all saved text messages. You can use a terminal program (and a levels converter!) to log the infinite loop on SER0 in a simple way, using 19200 baud (figure 3). You will notice that our demo software additionally checks and transmits time and date every time, a great extra service for optional inquiries and further tasks.

```
Uhrzeit abfragen:
at+cclk?
+cclk: "04/01/28,08:20:00"

OK

at+cmgl=0
+CMGL: 3,0,,24
0791947101670000240091946190039184F9000040108280122040044F37280C

OK
```

Fig. 3 Extract from the serial software protocol

The underlying demo software HANDY_02.TIG only “knows” the following commands:

| | | |
|------|---------|----------------------------------|
| On a | Off a | Switches L80 to high or low |
| On b | Off b | Switches L81 to high or low |
| On c | Off c | Switches L82 to high or low |
| | Off all | Switches L80, L81 and L82 to low |

Other messages are ignored and do not lead to switching operations, even if they contain such a text.

If you are familiar with AT commands you will be able to think of further options (identify the text message sender, check the state of charge, transmit measured data etc.) Always keep in mind that a text message does not always reach the receiver, e.g. if the network is running to capacity etc. Time-critical applications can hardly be realised this way. Nevertheless remote controlling via SMS is an attractive enrichment for electronics engineers. If you consider that receiving messages on your mobile phone is for free, you do not need an expensive contract for your “control phone”. A discarded phone and a cheap contract, e.g. with prepaid card, is favourable. This costs you nothing more than a one-time charge for the mobile phone contract. For the transmitting part you can use another mobile phone or, with a bit of luck, find a cheap or even free SMS service, so costs of a “worldwide remote control” are negligible.

5. Troubleshooting and further experiments

At the latest when something does not work, we need some tools to investigate the communication with a mobile phone. This requires at least a serial data cable that matches our mobile phone. This cable contains more than just a few wires – we know that the device only uses TTL levels (only 3V), the serial PC interface, however, uses RS232 levels (-3 to -15 V and 3 to 15 V). The cable has to contain a level converter, which is supplied by unused signal lines of the RS232 PC interface. If we connect such a cable to the mobile phone and to the PC, data interchange should already be possible. Every Windows PC provides communication software. With Windows 3.1 it used to be the software “Terminal”, now a program called “HyperTerminal” can often be found under “Accessories\Communication”. The latter, however, is much more complex than the old one, that is why the author uses the more simple version “Terminal” which is attached to this note. It should also work with Windows 98, 2000, ME and XP. With this software you can operate almost all communication tests that we need. Let’s start this software (it is a simple .exe which does not have to be installed) and change some settings:



Terminal

Under “Settings” and “Communications” resp. “Terminal Preferences” we make the following settings:

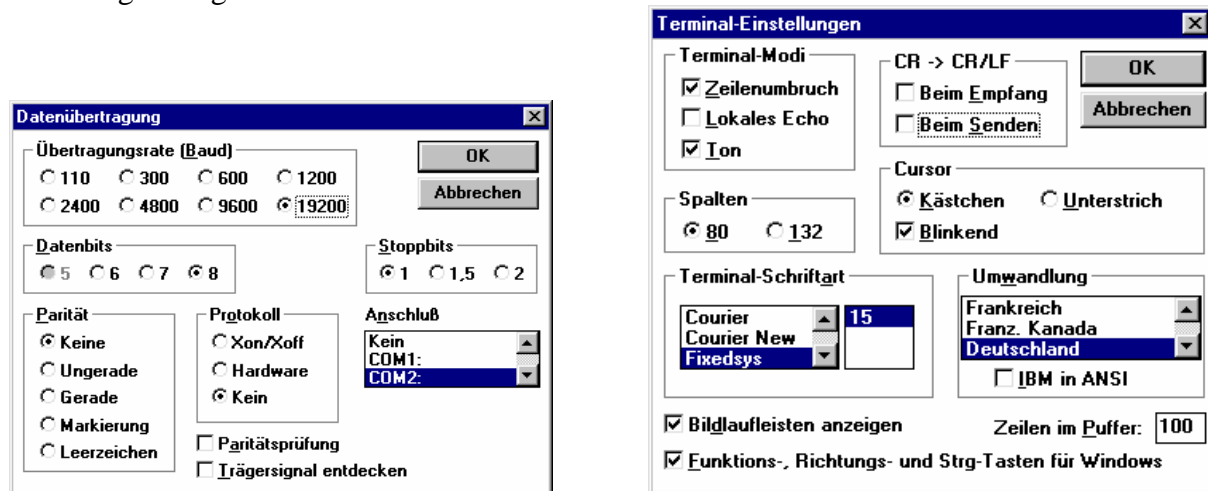


Fig. 4 Settings in program “Terminal”

You can, of course, choose an interface according to your PC. Finally we set TTY (general) under “Settings” and “Terminal Emulation”.

Now a blank window appears, in which we can enter our first command like this (capital letters are irrelevant):

at+cclk? (followed by ENTER)

With a bit of luck now the phone’s answer appears right under the entered line:

cclk: 04/01/28, 08:20:00 (current date and time)

Using the terminal software and the connected mobile phone you can now travel the world of AT commands. Please only try commands which do not change your S25 dramatically – and there are some of them! Carefully read the list of valid S25 commands and their consequences before you start off! It is also important to wait for the S25’s reaction on your command before you start with another one.

After these first steps you can use this knowledge and the construction with Economy-Tiger® and Tiger-BASIC® to approach new tasks. Everything you can do with the terminal software on your mobile phone can also be executed by BASIC-Tiger®. Even different mobile phones do not pose an unsolvable problem any more.

For those who would like to exploit the medium SMS for BASIC-Tiger® Wilke Technology provides a module set to encode and decode messages in PDU mode. It contains of three files (Include file Sms_v004.inc and the two Tiger programs sendSMS.TIG and receiveSMS.TIG which all are attached to this note).

With this further options for using SMS technology open up to the operator. The needed software development by the operator is immensely reduced.

Have fun with you new toy!