

How did you find this web page?

Did you googled: GSM alarm SMS control DIY do-it-your-self Microchip 16f628 ?

Well, this is exact what it is all about.

What is this.

This is seen before, but this time it is complete!

A simple GSM based monitoring and controlling system on a small board with a PIC processor holding a solid program able to control a various types of GSM modems and Cell phones.

Four inputs, one power output prepared for monitoring and controlling through SMS messages on your mobile phone. Send a SMS and turn on the relay. Get a SMS if the inputs are activated or the power is lost.

You can build it in your existing alarm system, or as a stand alone system. Use it in you car or where ever you like. You decide it.

I call it 'the bug'..

Introduction

This is a description on how you can build your own system allowing you to receive a SMS message when an input in changed and have the ability to switch on/off an output by sending a SMS to the system. The included software supports 4 input and 1 output, and 1 phone number for receiver of the SMS. You can send a SMS turning the output on/off, enable/disable the input lines or request a status for the outputs. Switching on/off (or enable/disable) is protected by a 4 digit pin code.

There is a circuit for battery backup up included and the main power supply is monitored by the Bug, so you will get a SMS if it is lost (if you use the battery backup).

That's it. It is simple and nothing more (you can find a lot of other application showing similar functions but they are often very early prototypes, no source code, with the 'wrong' processor or including a lot of fancy functionalities. This one is completed – all inclusive!)

The Bug uses a standard GSM modem (Wavecom or Siemens works fine) but some used old mobile phones will do it too if they support Text format for SMS (if they don't buy you a new one. It is more than 10 years ago since text mode became a standard and if you don't have a real modem buy you one. You can find a cheap GSM modem on ebay).

This article includes the full documentation and all you will need. It for free and you can use it or modify if you like (if you like it, or do the ultimate modification feel free to drop me a mail).

The ZIP file includes:

- Schematic diagram
- Part list incl. component layout
- Gerber & NC drill files for PCB.
- Source code for the application.
- HEX code ready for the PIC.

Enjoy the project,

Ole Wulff (owdk at yahoo-dot-com)

Prepare the SIM card

- remove pin code on the card.
 - create an entry in phone book on the SIM card with name 'PRI' and with the mobile phone number on which you want to receive the messages. The phone number should be in international format, e.g. +45xxxxxxxx for Denmark.
- Put in SIM card in card holder on the modem.

I recommend a pre-paid SIM card or a card with limit amount to make sure that you won't lose a fortune if your application goes crazy and switching on and off the inputs constantly (up to 30 SMS will in this case be send per minute!)

Power on the system

The LED will turn after a few seconds showing that the unit is initializing. After a few seconds the LED turns off again and the system is ready (this period can be longer – 1 min. - depending on how fast your modem got registered at the GSM net). From now on the LED will light shortly every 10th second when the system is checking for at new SMS or when an input is activated showing an SMS is being send.

If the LED does not turn off and continue to flash something went wrong when initializing (either no contact to modem or phone number no read correctly from the SIM card. Check the card and the communication).

Receiving an alarm

The system will send you a SMS every times on of the input in changed from normal state to alarm state (before a new SMS will be send for this input, the input need to be in normal state again).

The text messages are fixed for this system ("INx Activated" for inputs and "Main Power Lost/Restored" for power monitoring). If you need special text you will need to compile your own version (see later in the technical description).

If a battery backup is connected to the system you will receive a SMS when the main power is lost and when it is restored.

Input1 and 4 are 'normal closed' circuits, which means that a SMS is send when the connection to ground is removed. The input 2 and 3 are 'normal open' circuit.

All the input must be connected to potential free contacts or to an open-collector switch. Do not under any circumstances put power directly to the input as this will kill the processor.

Controlling the system

You can control the output by sending a SMS to the system. In every 10th second the system will check if a new SMS is received. Three simple commands are implemented:

#xxxx:0	Turn the output OFF
#xxxx:1	Turn the output ON
#xxxx:A	Arm/Enable the inputs
#xxxx:D	Disarm/disable the inputs
#xxxx:?	Return a SMS with a status of the output.
#xxxx:v	Return firmware version

xxxx is the pin code for the system. This is the last for digits of the phone number 'PRI' read when then system initialized.

When the system is powered up, the input will be enabled as default and the output will be off.

If you disable the inputs the LED will reverse its function: Staying ON must of the time and goes OFF when the SMS is checked or an SMS is send.

The input can also be disabled if the D1 input is connected to minus (normal state is high). Note: If you have turn off the input by D1 you will not be able to turn the input on by sending a SMS (if you are using the system as an alarm system you wont like the system to be armed remotely while you are in the house...)

When the system receives the command it will be executed and a status SMS showing the result of the command will be returned to you so always know if the operation is completed.

Using it as an alarm system

On the board you will also find a open-collector output on pin O3. This pin will be activated for 3 minutes if input 1 to 4 is activated and the inputs are enabled. This output can be connected to a siren (max. 500 mA) to build a simple system.

If you send an SMS turning the alarm off or disable the input by a switch on O3 the output will immediate turn off.

Building the system

You can build the system using a proto type board or build the system using the shown PCB (you can buy both PCB and a pre-programmed processor from me).

If you don't modify the source code just find a PIC programmer and put the .HEX file into PIC. Connect the bug to the modem and you are ready to go.

The inputs 1 and 4 are 'normal closed' and 2 and 3 are 'nomal open'. 'Normal Closed' means that they need to be connected to minus/ground when not activated and left open when activated (this does exactly fit the output from passive infrared sensors and other alarm equipment, so you can build a simple alarm system).

If you want to use the option for battery backup in case of power failure, you can connect a standard 9-12 V battery to the 'B+' terminal.

Prepare the GSM modem

You can use every kind of modem supporting the shown commands (see figure) – all modem types from Wavecom, Siemens M20 modem, Siemens S45 telephone and Nokia 6210 have been tested,. The modem must be connected through a real RS-232 connection (the same as the one you find on a PC). Connect the modem to a PC and check the commands AND the respond (some modem sends other respond and you might need to adapt the software).

The modem will normally have been set for auto baud rate as factory default. To ensure proper functionality lock the modem to fixed baud rate: 9600 baud using the following AT commands send to the modem from your PC (using eg. hyperterminal):

```
AT+IPR=9600 (9600 baud)
AT+ICF=3,0 (8 data bit-No parity-1 stop bit)
AT+IFC=0,0 (no flow control)
AT&W (store as default)
```

Cycle power and check if the modem respond to baud rate set to: 9600, 8 bit data, 1 stop bit and no parity as default.

The technical part

The software is written in a combination of Basic and Assembler using the PIC Simulator from Validimir Soso. A super fine development tool. You can test it for free or you can buy you own license for a few Euro's (check www.oshonsoft.com).

The Basic in the simulator includes commands for communicating through the serial port on the PIC, but there is no error handling in the receiver routine and to ensure full stability in the software a new receive routine has been written in assembler (this is based on the studio of PIC hardware serial port written in PICUART.ASM document. You can find it on the net too). Rest of the code is written in Basic so it is straight forward to debug if you want to modify.

After initializing the modem (the software check for a respond from the modem before trying to do further communication) the software reads the phone number from the SIM card and stores it in local memory. After this it goes into a loop checking the inputs incl. power and very 10 seconds checking for a new SMS.

If an input is activated the phone number is read from memory and send to the modem followed by the messages.

If a new SMS is received the modem will store it in position 1 if free otherwise it will be stored on the first free location. By clearing the first position we ensure that any new message will be stored here. This simplifies the software as it only needs to check for new message on this position. If a new message has arrived it will be read, decoded and deleted.

The decoding part checks that the pin code in the message is identically with the last four digits of the phone number. If ok the included command is executed.

Polling the modem every 10th second is simple method but it has a disadvantage. If the system receives two new messages in a row it will only be able to handle the first one and the other one will be lost. Due to this you should disable any 'service information' from your GSM operator to limit the amount of crappy SMS received. If you send a command to the system and you don't get at respond within a minute, then just resend it.

For this simple setup only five commands are implemented: Two for switching the output ON/OFF, two for enable/disable the inputs and one for asking for status for the output. It is easy to add more commands if you like (find the section in the code where the pin code is checked and put in you own commands)

SMS / GSM fundamental

The following commands are used for communication with the modem.

Command	Respond	Comment
AT	OK	Check if modem is ready for communication
ATE0	OK	Disable echo from modem. If you are debugging it might be nice to switch echo on by ate1
AT+CNMI=0,0,0,0	OK	Setup respond from modem when a SMS is received (send nothing – we use polling technique on the modem)
AT+CMGF=1	OK	Set text mode for modem. If you modem does not support txt modem but only pdu, buy you a new one!
AT+CMGD=1	OK	Delete the message in 1 st position. If 1 st position is empty a new messages will always be stored there.
AT+CPMS="SM"	+CPMS: x,y,x,y x,y	Use SIM card phone book.
AT+CMGS="+xxxxxx xx"	>	Tell the modem to send a SMS to phone number: xxxxx The text is send right after the command and ended by Ctrl-Z character (ASCII 26).
AT+CMGR=1	+CMGR: "REC UNREAD", "+45xxxxxxx", "dd/mm/yy, hh:mm:ss+08"#1234:0'	Read the SMS in 1 st . position. You will get an error if not available
AT+CPBF="PRI"	+CPBF: 1, "+45xxxxxxx", 145, "PRIMARY"	Read the entry 'PRI' in the phone book. Note: Some modems are case sensitive other responds if only a part of the string matches.

All commands are followed by CrLf, except after the message where is should be Ctrl-Z

GSM SMS commands in a nut shell