



GPIO SMS Remote Control and Monitoring Software

The GPIO software allows the user to remotely control physically connected hardware e.g. turn lights on / off via SMS message sent from a mobile/cell phone or any other means that can send SMS messages (e.g. via server side software using an SMS Gateway such as provided by clickatell.com). It can also signal changes in state e.g. switch turned on / off via SMS message back to a pre specified telephone number.

The GPIO software is controlled via the use of SMS Command messages, these are standard text messages send via a mobile phone however they are formatted in a specific way that instruct the GPIO software to perform a set action, such as turning an output on or off.

The GPIO Python software is designed to operate on a range of embedded Python supporting hardware modules from Telit. For further details see <http://www.telit.com/en/products.php> It can be used either directly by those designing their own hardware incorporating the Telit modules (GM862-QUAD-PY, GE863-PY, GC864-PY etc) or those who purchase off the shelf pre boxed hardware such as the GT863-PY, GT864-PY, Gatetel EZ10 PY or EZ863 PY Terminals.





SMS Commands

smspassword

Format: `[password] smspassword [newpassword]`

Example: `1234 smspassword 1111`

Allows setting of a new 1-12 character password. The pin number is required in all SMS command messages sent to the unit. The default password is 1234.

tel

Format: `[password] tel [telephone number]`

Example: `1234 tel +441234567890`

Allows setting of the telephone number where changes in GPIO input state to the unit are reported.

status

Format: `[password] status`

Example: `1234 status`

Example Reply: `Status: sw1 in on, sw2 in off, led1 out off...`

(assumes default GPIO alias names of "gpio1", "gpio2" etc have been changed by the user, and input output type of the GPIOs changed appropriately)

On receipt of this message the software will respond, detailing the current state of all GPIOs i.e. if they are set as input (and if so if the input is reading on or off), or if they are outputs (if so they are set as on or off). The comma separated order of which GPIOs states are reported in the status message is undefined.





alias

Format: `[password] alias [gpio no] [gpio alias], [gpio no] [gpio alias]`

Example: `1234 alias 1 gpio1, 2 gpio2, 3 gpio3`

All commands reference physical hardware GPIOs e.g. 1-16 via the use of alias names. These names are set by default to "gpio1" for GPIO 1, "gpio2" for GPIO 2 etc. Alias names can be changed by specifying a comma separated list of GPIO number followed by the new name. Alias names may be 1-10 characters a-z, 0-9 only. The user can set as few or as many alias names as they like in a command message (in the example above 3 alias names are set at once) as long as the total length of the message does not exceed 160 characters.

iotype

Format: `[password] iotype [gpio alias] in, [gpio alias] [in/out]`

Example: `1234 iotype gpio1 in, gpio2 in, gpio4 out, gpio5 out, gpio6 out`

Allows setting of a GPIO to either an input or an output. Inputs report state changes (i.e. connected switch is turned on) via SMS to a pre specified telephone number. Outputs can be controlled by sending an SMS to the software setting their state (e.g. LED is turned on).

Note: Setting a GPIO alias to an output will automatically turn on reporting for that output (for an explanation of "reporting" see below).

report

Format: `[password] report [gpio alias] on, [gpio alias] [on/off]`

Example: `1234 report gpio4 on, gpio5 on, gpio6 off`

For those GPIOs which are set as inputs (using the iotype command) it is additionally possible to turn reporting on and off for them. If reporting is turned off, changes in state will not be sent to the pre specified telephone number. The current status of all input GPIOs however can still be determined via the status command on request. The user may turn off reporting for a GPIO if instant notification of a change is not required, and may then manually query it by sending an SMS message requesting the status of all GPIOs as required.





output

Format: `[password] output [gpio alias] on, [gpio alias]
[on/off]`

Example: `[password] output gpio4 on, gpio5 on, gpio6 off`

Allows setting of GPIO alias names which are currently set as iotype output as on or off, e.g. to turn on or off a connected LED.





General Notes

- The software will respond to all command messages sent to it with a text message returned to the sender detailing if the command was successful or if it failed. The only exception to this is when an incorrect password is specified, in this case no response will be given.
- A change in state is defined as a state different to that state observed at power on or different to the previous state. E.g. if an input GPIO is in the high state at power on, an SMS message will be sent when it changes to low, and then subsequently when it changed back to high, and on any further state changes.
- State changes are generally detected within a few seconds however if the software is busy (i.e. processing previous SMS command messages) the time taken to detect a state change may increase.
- SMS text messages have a 160 character limit. In order to prevent SMS replies from the software from spilling over into multiple messages it is recommended to use names with the minimum possible amount of characters.
- When sending SMS command messages to the software using more advanced phones (which automatically split text greater than 160 characters into multiple messages) care should be taken not to exceed the 160 character limit. To set alias names for e.g. 16 messages at once with the full length of 10 characters for each of 16 GPIOs, and a pin consisting of 12 characters would require 2 separate messages to be sent.
- If there is an error encountered with a command e.g. setting the alias names of multiple GPIOs, or turning on / off multiple GPIOs, all actions will be processed upto the point where the error is encountered. Any actions after the error will not be processed.

