

# Internet connectino Watchdog

this is about creating a raspberry pi based appliance that monitors your internet conenction and if it goes down, automatically debugs it and restarts stuff like modem / router etc. as necessary. the device should also be reachable via an on-demand gsm link that can be initiated by sms to login to the device and debug the link manually

so far, this is just a collection of snippets of information and links, i'll write it up into a nice article one day, but not today ;)

## software installation

```
apt update && apt install wvdial gammu gammu-smsd picocom
```

## SMS

- Software: [Gammu](#)
- Hardware: TP-Link MA260
- on the tp-link stick the last serial device seems to be the one that can be used for sms (gammu identify only gets information from that device).

```
gammu-detect | grep -B1 TP-LINK | grep ttyUSB | tail -n 1
```

to get the device name for the config

- /etc/gammu-smsdrc.template

```
# Configuration file for Gammu SMS Daemon

# Gammu library configuration, see gammurc(5)
[gammu]
port = %PORT%
connection = at
#logformat = textall

# SMSD configuration, see gammu-smsdrc(5)
[smsd]
service = null
logfile = syslog
# Increase for debugging information
debuglevel = 0
RunOnReceive = /opt/process-sms.sh
```

- /root/.gammurc

```
[gammu]
```

```
device = %PORT%  
connection = at
```

- [/opt/updateModemPort.sh](#)

#### [updateModemPort.sh](#)

```
#!/bin/bash  
port=$(gammu-detect | grep -B1 TP-LINK | grep ttyUSB | tail -n 1 |  
awk -F = '{print $2}' | tr -d " ")  
sed -e "s|%PORT%|${port}|" /etc/gammu-smsdrc.template >  
/etc/gammu-smsdrc  
sed -e "s|%PORT%|${port}|" /root/.gammurc.template >  
/root/.gammurc
```

- installation:

```
apt install gammu gammu-smsd
```

- [process-sms.sh](#)

```
#!/bin/bash  
for i in `seq $SMS_MESSAGES` ; do  
    numvar="SMS_${i}_NUMBER"  
    textvar="SMS_${i}_TEXT";  
    echo "${!numvar}:${!textvar}" >> /tmp/messages  
    if [ "${!textvar}" == "connect" ]; then  
        echo "connect!" >> /tmp/messages  
    fi  
done
```

## GSM Link

- Hardware: See above
- Software: wvdial
- config: /etc/wvdial.conf.template

```
[Dialer Defaults]  
Init1 = ATZ  
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0  
Init3 = AT+CGDCONT=1,"IP","internet"  
Modem Type = Analog Modem  
Baud = 960000  
New PPPD = yes  
Modem = %PORT%  
ISDN = 0
```

```
Phone = *99#  
Password = { }  
Username = { }
```

## Relay Card

- Hardware: Conrad electronics, part number 967720 (Relaiskarte 8Fach Seriell)
- Software: [http://www.thomas-dohl.de/?Projekte\\_\\_Relaiskarte](http://www.thomas-dohl.de/?Projekte__Relaiskarte)

## serial console to router

- via dual prolific basead usb to serial converter, one link goes to the relay card, the other to the serial terminal of the pfsense router.
- usb link is extended down to the basement via cat7 link, I simply cut a usb extension cord into two and attached RJ45 plugs to both ends with identical pin layout. works like a charm :)
- software: picocom

From:

<http://wiki.psuter.ch/> - **pswiki**

Permanent link:

[http://wiki.psuter.ch/doku.php?id=internet\\_connection\\_watchdog](http://wiki.psuter.ch/doku.php?id=internet_connection_watchdog)

Last update: **17.06.2018 22:21**

