```
##################################################################################################################
#############################################
#sendSMS.py - a-gsm 2.064 send sms example utility
#COPYRIGHT (c) 2014 Dragos Iosub / R&D Software Solutions srl
#
#You are legaly entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH a-gsm DEVICES USAGE.
Modifications, derivates and redistribution
#of this software must include unmodified this COPYRIGHT NOTICE. You can redistribute this
SOFTWARE and/or modify it under the terms
#of this COPYRIGHT NOTICE. Any other usage may be permited only after written notice of
Dragos Iosub / R&D Software Solutions srl.
#
#This SOFTWARE is distributed is provide "AS IS" in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied
#warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
#
#Dragos Iosub, Bucharest 2014.
#http://itbrainpower.net
##################################################################################################################
#############################################
#Raspberry PI - a-gsm wiring connection:
# Legal disclaimer:
# Incorrect or faulty wiring and/or connection can damage your RPi and/or your a-gsm board!
# Following directives are provided "AS IS" in the hope that it will be useful, but WITHOUT
ANY WARRANTY!
# Do the wiring on your own risk!

#name       RPi     a-gsm shield
#
#POWER a-gsm    16      D7  - power(UP/DOWN)
#RESET a-gsm    18      D6  - reset
#a-gsm STATUS   12      D5  - status
#
#serial TXD0    08      D4  - tx(rxd)
#serial RXD0    10      D3  - rx(txd)
#
#5V         02/04       5V  - on Arduino power IN connector
#GND        06/14       GND - on Arduino power IN connector
#
#IMPORTANT:
# a-gsm's POWER supply input selector must be in "use 5V pin" position
##################################################################################################################
#############################################

# this utility must be runned as root (just use: sudo python sendSMS.py)

#next: no more than 160 chars -text SMS maximum lenght
message="Hi!\r\nThis message has been sent from a-gsm v2.064 shield connected with my RPi
board."
#destinationNumber="+40123456789"#EXAMPLE FOR BELLOW
destinationNumber=""#usually phone number with International prefix eg. +40 for Romania - in
some networks you must use domestic numbers
serialSpeed = 19200#we recommend usage of 19200 bps speed. If you want to use other speed,
first set the a-gsm speed using setSerial.py
usePoweringControl = 1#change it to 0 if you do not want to control powerUP/powerDown the
a-gsm board. In this case, please be sure the a-gsm board is powered UP(the a-gsm green led
lights continuous) before run this utility
```

```
#Do not change under following line! Insteed make one copy of the file and play with!
##############################################################################################################
###############################################

#definitions for a-gsm control(RPi GPIO mode)
POWER = 16
RESET = 18
STATUS = 12

i=0
buffd =""
sreadlen = 100#how many chars to read in one try over serial

import os
import serial
from time import sleep, time
import RPi.GPIO as GPIO

if destinationNumber=="":
    print("No destination number has been set for your SMS!")
    print("Enter: \"sudo nano sendSMS.py\" and fill in the destinationNumber in line 42\r\n")
    exit(0)

if not os.getuid() == 0:
    print("please use root privileges! try: \"sudo python sendSMS.py\"")
    exit(0)

agsm = serial.Serial("/dev/ttyAMA0", serialSpeed, timeout=1)
agsm.open()

print "Hy folks. Lets send some SMS with your a-gsm shield"


#poweron() - power up the modem
def poweron():
    #...function code here

#poweroff() - shutdown the modem
def poweroff():
    #...function code here

#recUARTdata(endchars,to,tm)
#   read from modem - read string is loaded in global var buffd
#   looking for endchars [SUCCESS STRING,FAILURE STRING] and to - TIMEOUT
#   return 0 for SUCCESS, 1 for FAILURE, -1 for timeout
#   tm how many chars to read(maximum) in one loop from serial
def recUARTdata(endchars,to,tm):
    #...function code here
    return SuccessErrorTimeout

#sendATcommand(command, endchars,to)
#   command +"\r\n" is forwarded to modem
#   looking for endchars [SUCCESS STRING,FAILURE STRING] and to - TIMEOUT
#   return 0 for SUCCESS, 1 for FAILURE, -1 for timeout
#   modem response is loaded in global var buffd
def sendATcommand(command, endchars,to):
```

```
    global sreadlen
    agsm.write(command+"\r\n")
    return (recUARTdata(endchars,to,sreadlen))

#aGsmWRITE(command)
#   just write command to serial without CR LF
def aGsmWRITE(command):
    agsm.write(command)

#sendSMS(phno, phtype, message)
#   phno    - phone number - see exambles bellow
#   phtype  - 147/129 - see exambles bellow
#   message - SMS message to be transmited
#   returns 0 on success, 1 on failure
#
#   res = sendSMS(destinationNumber, "129", message)#domestic format numbers
#   res = sendSMS(destinationNumber, "145", message)#international format numbers
#   anyway, check AT command pdf for proper 129/145 parameter/number format usage
def sendSMS(phno, phtype, message):
    #...function code here
    return sendSMSSuccessorError

#setupMODEMforSMSusage()
#   what do you think this baby doing here?
def setupMODEMforSMSusage():
    #...function code here


#RaspberryPI hardware setup section start
if usePoweringControl==1:
    GPIO.setmode(GPIO.BOARD)
    GPIO.setwarnings(False)
    try:
        GPIO.setup(STATUS, GPIO.IN)
        GPIO.setup(POWER, GPIO.OUT, initial=GPIO.LOW)
        GPIO.setup(RESET, GPIO.OUT, initial=GPIO.LOW)
    except:
        GPIO.cleanup()#free GPIO
        GPIO.setup(STATUS, GPIO.IN)
        GPIO.setup(POWER, GPIO.OUT, initial=GPIO.LOW)
        GPIO.setup(RESET, GPIO.OUT, initial=GPIO.LOW)
    GPIO.setwarnings(True)
#RaspberryPI hardware setup section end

#here start the main code
if usePoweringControl==1:
    poweron()

setupMODEMforSMSusage()

#check AT command pdf for proper 129/145 parameter/number format usage
#res = sendSMS(destinationNumber, "129", message)#domestic format numbers
res = sendSMS(destinationNumber, "145", message)#international format numbers
if res==0:
    print "SMS has been sent with succes"

agsm.close()#close serial
```

```
sleep(5)

if usePoweringControl==1:
    poweroff()#shutdown a-gsm

GPIO.cleanup()#free GPIO
```