Changelog of SMAspot install on a Raspberry Pi

Changelog 0.5 20/06/2013

- Added update procedure

Changelog 0.4 22/05/2013

- Added installation on an usb-stick

Changelog 0.3 21/05/2013

- Update on raspi-config

Changelog 0.2 13/05/2013

- minor change in explanation of watchdog.conf

Changelog 0.1 05/05/2013

- Extra option in SMAspot.cfg
- Added option in cronjob to reduce output in the logfiles
- Added watchdog (with extra's)
- Added cleanup script

Changelog 0.0 09/04/2013

- First version online, downloadable on the SMAspot site

Introduction

This document includes all steps on how to install SMAspot
https://code/google.com/p/sma-spot/ on a Raspberry Pi http://www.raspberrypi.org/

Hardware used:
1x – Raspberry Pi version B
1x – SD-card 4GB (The OS will need 2GB, the rest is for storage)
1x – Wifi dongle (only if needed, the Pi has a LAN-port)
1x – USB bluetooth dongle (some work some don't, see
http://elinux.org/RPi_VerifiedPeripherals)
1x – Power adapter with micro usb connector
1x – Enclosure

All can be bought at Element14 (Farnell) or RS Components:
http://www.farnell.com/
http://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi

In general costs will be around €70-100

Windows users will need PuTTY to connect to the Pi:
Download PuTTY here: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

---
Commands to be entered on the Pi's commandline will be *arial italic* and are not preceded by an > or $ as can be seen in other tutorials. Text for scripts and configfiles are `in this small font` and text in blue means you have to enter your own settings/name
Finally small but same font is used for not necessary options.

Installing the Raspberry Pi

Let's start with information on the installation of the Pi itself. First the latest Raspbian OS is downloaded from: http://www.raspberrypi.org/downloads and is transferred to a sd-card. The procedure for Windows, Mac OS X and Linux can be found here:
http://elinux.org/RPi_Easy_SD_Card_Setup

In this how-to the image used is: 2013-02-09-wheezy-raspbian.img

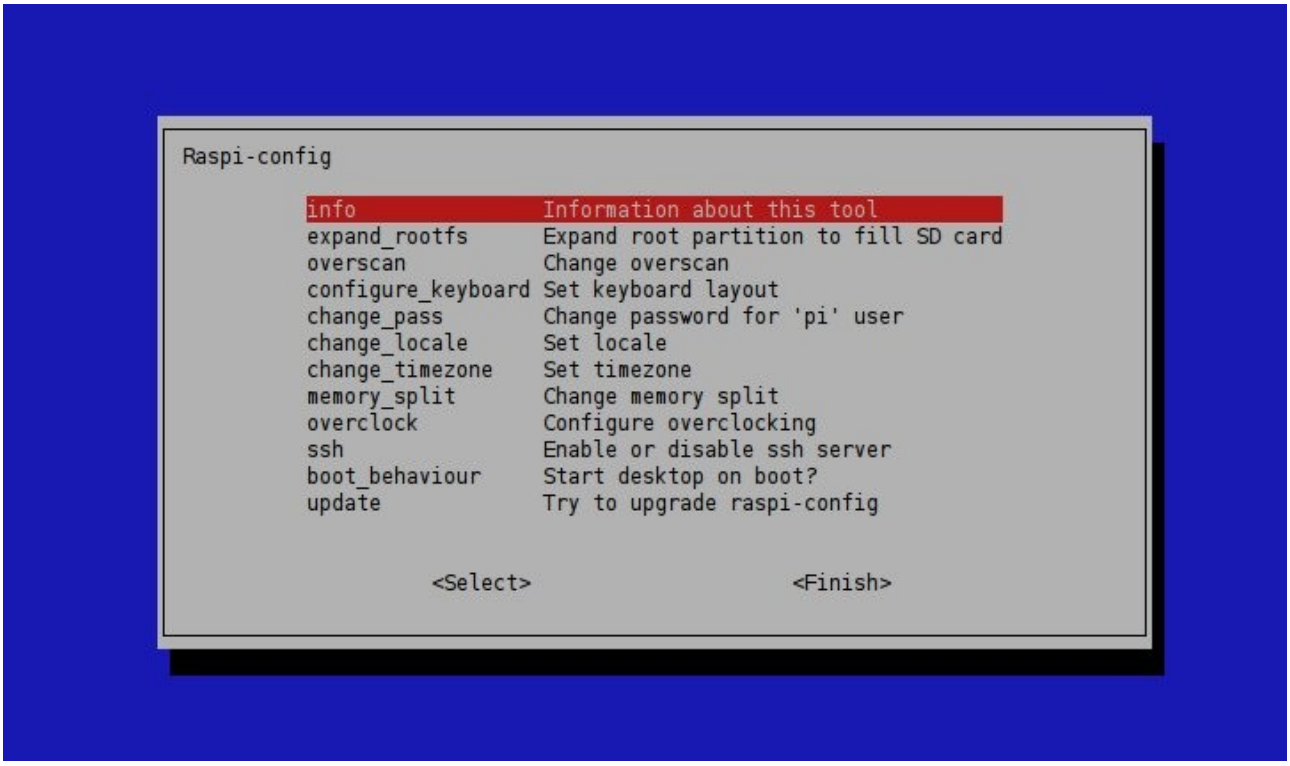
Start the Pi for the first time:
Put the prepared sd-card into the Pi, connect the power and a LAN cable. When the Pi starts, you should see a red led on and a green led flashing. Log on to your router and find the corresponding IP-address for your Pi (in your DHCP client list).

Start PuTTY (for windows users, copy/paste does work in PuTTY to make it easier with the longer commands) or if you are using Linux, go to a terminal and type *ssh pi@192.168.x.x*

Standard login
User-id: *pi*
Password: *raspberry*

Configuring the system:

sudo raspi-config



To navigate in above menu use the up and down arrow:

- expand_rootfs

The installed OS uses 2GB, this will make the rest of the sd-card accessiblechange_pass
Give the Pi your own password

- change_time zone

According to where you live ;-)

- memory_split

Since there will be no screen attached to the Pi, I selected only 16Mb for the GPU, the rest can be used by the CPU

- ssh

Standard enabled, don't change this!

- boot_behaviour

no, the Pi will be connected to by ssh (PuTTY), without a monitor

- update

First time use for update/upgrade of raspi-config

Afterwards you return to the raspi-config menu. You can see a few extra options:

- overscan
- change_hostname
- camera
- rastrack

None of the above are discussed here, since they are not necessary for the purpose of this

how-to. So continue with:

- Finish

When you're at the bottom of the menu, use the right arrow to jump to Finish

You have to reboot now, some users experience that raspi-config ask for a Reboot Yes/No?, otherwise use the following command:

sudo shutdown -r now

You can use *sudo reboot*, but I prefer the other command, because it stops services first instead of a direct reboot. If for some reason you want to shutdown the Pi (for example to place it somewhere else):

sudo shutdown -h now

the argument -h is for halt, where -r is reboot.

The connection is closed, so you'll have to log on with the new password.

Start PuTTY again and make connection with the Pi (ip-address from the DHCP-client list, we will change this in a fixed address later)

Make the Raspbian OS up-to-date

To do this, we install RPi-update (software from the Pi community) after an upgrade of all installed packages so far:

sudo apt-get upgrade

After this command, you can see how many packages need an upgrade and how much space it will use on your sd-card. Answer **y** on the question if this allowed. Depending on how much packages need an upgrade this will take from several minutes up to a (half) hour.

When using commands like *sudo apt-get install thequickbrownfox* there always will be the question if you want to install the extra Mb's. The answer should be **y** otherwise it's of no use to enter the command in the first place. There will be plenty of text scrolling over the screen and it might take a while. Just consider the Pi with its 700MHz processor a Pentium 2 computer from last century.

Installing Rpi-update:

Rpi-update depends on the software package git:

sudo apt-get install git-core

Followed by the command:

sudo wget http://goo.gl/1BOfJ -O /usr/bin/rpi-update && sudo chmod +x /usr/bin/rpi-update

To update the Pi-firmware:

sudo rpi-update

After a successful update, the Pi needs a reboot. If it doesn't ask for it or goes automatically, you can give a reboot, connect again and retry above command again.

Option
The Pi doesn't have a build-in hardware clock, instead you can use the NetworkTimeProtocol
http://www.pool.ntp.org

sudo apt-get install ntpdate
sudo ntpdate -u ntp.ubuntu.com
/Option

Give the Pi his fixed IP-address

Comes in handy when your DHCP server mixes things up or to know on which address to log on each time you want to do something with the Pi.

sudo nano /etc/network/interfaces

You enter the interface configuration file of the Pi with the use of the nano text editor. The preface sudo runs the command with superuser rights.

The text should start with:

```
auto lo

iface lo inet loopback
iface eth0 inet dhcp
```

Replace dhcp with static and add address / netmask and gateway (things you can get from your modem/router)

It now will look like:

```
auto lo

iface lo inet loopback
iface eth0 inet static
address xxx.xxx.x.xxx
netmask xxx.xxx.xxx.x
gateway xxx.xxx.x.xxx
```

Depending on your home network you enter the right numbers. Best is to choose an IP-address outside the range of your DHCP server, but of course in the same subnetwork.

Save the file with Ctrl-O. Exit with Ctrl-X. If you forget to save, nano will ask on exit if you want to save or not. Answer: **y**

The Pi now has his own fixed IP-address, so you reboot:

sudo shutdown -r now

and log on with the newly given address.

How to install SMAspot on the Raspberry Pi?

Finally you might think, but first things first. Install a bluetooth driver, it 's not there by default:

sudo apt-get install bluetooth

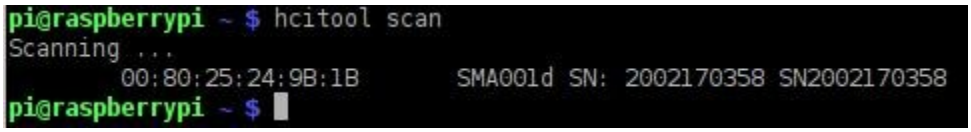
On screen you'll see:
pi@raspberrypi ~ $ sudo apt-get install bluetooth
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libblas3gf liblapack3gf
Use 'apt-get autoremove' to remove them.

Option
Besides sudo apt-get install and sudo apt-get upgrade, there are the commands sudo apt-get autoremove and sudo apt-get clean to get rid of unnecessary software packages
/Option

Check for bluetooth connectivity with the SMA converter

hcitool scan

You should see something like this:



Make a note of the Bluetooth address xx:xx:xx:xx:xx
We will use it later in the SMAspot.cfg file

Now there are two more package dependencies:

sudo apt-get install libbluetooth-dev

and the next one:

sudo apt-get install libcurl3-dev

It's oké to go with the new version libcurl4-openssl-dev

Option
You should consider to stop the Pi and make a copy of the image on the sd-card. In case you need it, next time you can skip a few steps and directly use the newly made backup-image on a (new) sd-card.
/Option

Now it's really time to install SMAspot on the Pi

If you want your SMAspot installed on an usb-stick, go to Install SMAspot on an usb-stick and when that's done return here ** but read /mnt/usb instead of /home/pi

First we create the folder where to install SMAspot and go to that folder

cd /home/pi
mkdir smaspot
cd smaspot

**

Now we download the source code, in this case the latest version 2.0.6

wget https://sma-spot.googlecode.com/files/SMAspot_SRC_206_Linux_Win32.tar

Let's unwrap:

tar -xvf SMAspot_SRC_206_Linux_Win32.tar

and compile SMAspot:

make release

You should see this on your screen:

```
pi@raspberrypi ~/smaspot $ make release
test -d bin/Release || mkdir -p bin/Release
test -d obj/Release || mkdir -p obj/Release
g++ -Wall -O2  -c Bluetooth.cpp -o obj/Release/Bluetooth.o
g++ -Wall -O2  -c SMANet.cpp -o obj/Release/SMANet.o
g++ -Wall -O2  -c SMAspot.cpp -o obj/Release/SMAspot.o
g++ -Wall -O2  -c misc.cpp -o obj/Release/misc.o
g++ -Wall -O2  -c strptime.cpp -o obj/Release/strptime.o
g++ -Wall -O2  -c sunrise_sunset.cpp -o obj/Release/sunrise_sunset.o
g++ -Wall -O2  -c CSVexport.cpp -o obj/Release/CSVexport.o
g++ -Wall -O2  -c PVOutput.cpp -o obj/Release/PVOutput.o
g++  -s  obj/Release/Bluetooth.o obj/Release/SMANet.o obj/Release/SMAspot.o obj/Release/misc.o
obj/Release/strptime.o obj/Release/sunrise_sunset.o obj/Release/CSVexport.o obj/Release/PVOutput.o
-lbluetooth -lcurl -o bin/Release/SMAspot
```

If so, you are doing oké and the source code can be removed:

rm SMAspot_SRC_206_Linux_Win32.tar

After compiling is done, there is a new folder in your smaspot folder called bin, and in bin there is the subfolder Release. In the Release folder is the actual SMAspot software. Make sure you are in the smaspot folder (if you didn't move you are still there):

cd /home/pi/smaspot

Let's configure the config file:

nano SMAspot.cfg

The following will appear on your screen, some things need to be changed some don't.

```
#################################################################################
#                 _____  __  __    _                          _
#                / ___||  \/  |  / \    ___ _ __    ___ | |_
#                \___ \| |\/| | / _ \ / __| '_ \ /  _ \| __|
#                 ___) | |  | |/ ___ \\__ \ |_) | (_) | |_
#                |____/|_|  |_/_/   \_\___/ .__/ \___/ \__|
#                                         |_|
#
#  SMAspot.cfg - Configuration file for SMAspot.exe
#  SMAspot - Yet another tool to read power production of SMA solar
inverters
#  (c)2012-2013, SBF (mailto:s.b.f@skynet.be)
#
#  DISCLAIMER:
#  A user of SMAspot software acknowledges that he or she is receiving this
#  software on an "as is" basis and the user is not relying on the accuracy
#  or functionality of the software for any purpose. The user further
#  acknowledges that any use of this software will be at his own risk
#  and the copyright owner accepts no responsibility whatsoever arising from
#  the use or application of the software.
#
#################################################################################

# SMA Inverter's Bluetooth address
# Windows: smaspot -scan
# Linux  : hcitool scan
BTAddress=00:00:00:00:00:00

Here goes your SMA inverter's BT address found with hcitool

# User password (default 0000)
Password=0000

# Plantname
Plantname=MyPlant

spaces are allowed

# OutputPath (Place to store CSV files)
#
# Windows: C:\TEMP\SMA\%Y
# Linux  : /home/sbf/Documents/sma/%Y
# %Y %m and %d will be expanded to Year Month and Day
OutputPath=/home/pi/smadata/%Y

As example, no need to create the folder smadata as SMAspot will do so.
%Y can be left out if you don't want a Year folder. Some might want to have
the datafolder inside the smaspot folder: /home/pi/smaspot/smadata

# Position of pv-plant http://itouchmap.com/latlong.html
# Example for Ukkel, Belgium
Latitude=50.80
Longitude=4.33
```

Search for the correct position on the site. This will be used to calculate sunrise and sunset.

```
# Calculate Missing SpotValues
# If set to 1, values not provided by inverter will be calculated
# eg: Pdc1 = Idc1 * Udc1
CalculateMissingSpotValues=1

# DateTimeFormat (default %d/%m/%Y %H:%M:%S)
# For details see strftime() function
# http://www.cplusplus.com/reference/clibrary/ctime/strftime/
DateTimeFormat=%d/%m/%Y %H:%M:%S

# DateFormat (default %d/%m/%Y)
DateFormat=%d/%m/%Y

# DecimalPoint (comma/point default comma)
DecimalPoint=comma

# TimeFormat (default %H:%M:%S)
TimeFormat=%H:%M:%S

# SynchTime (default 0 = Off)
# If set to 1 and Inverter time differs more than 1min from pc time,
# the Inverter time is synchronised with pc time
# Some inverters don't have a real-time clock
SynchTime=1

# SunRSOffset
# Offset to start before sunrise and end after sunset (0-3600 - default 900 seconds)
SunRSOffset=900

############################
### CSV Export Settings ###
############################
# With CSV_* settings you can define the CSV file format

# CSV_Export (default 1 = Enabled)
# Enables or disables the CSV Export functionality
CSV_Export=1
```

If you only want upload to PVOutput and there is no need for .csv files, choose option 0

```
# CSV_ExtendedHeader (default 1 = On)
# Enables or disables the SMA extended header info (8 lines)
# sep=;
# Version CSV1|Tool SMAspot|Linebreaks CR/LF|Delimiter semicolon|
Decimalpoint comma|Precision 3
# etc...
# This is usefull for manual data upload to pvoutput.org
CSV_ExtendedHeader=1

# CSV_Header (default 1 = On)
# Enables or disables the CSV data header info (1 line)
# dd/MM/yyyy HH:mm:ss;kWh;kW
# This is usefull for manual data upload to pvoutput.org
# If CSV_ExtendedHeader is enabled, CSV_Header is also enabled
CSV_Header=1
```

```
# CSV_SaveZeroPower (default 1 = On)
# When enabled, daily csv files contain all data from 00:00 to 23:55
# This is usefull for manual data upload to pvoutput.org
CSV_SaveZeroPower=1

# CSV_Delimiter (comma/semicolon default semicolon)
CSV_Delimiter=semicolon

# CSV_Spot_TimeSource (Inverter|Computer default Inverter)
CSV_Spot_TimeSource=Inverter

################################
### Online Monitoring Systems ###
################################
#
# In the future, multiple online monitoring systems can be defined
# Here we can activate the ones we like
#
################################
### PVoutput Upload Settings ###
################################
# PVoutput (default 0 = Disabled)
# Enables or disables the upload functionality to pvoutput.org
# When enabled, be sure to use -u switch on the command line
PVoutput=1

Default there is no upload to PVOutput, but this is what we want so option 1

#PVoutput_SID
#Sets PVoutput System ID
PVoutput_SID=00001

Of course yours is needed

#Pvoutput_Key
#Sets PVoutput API Key
PVoutput_Key=12347babcge2a9d50aklm0ael1t323737f6d1ab3a8f1337

likewise

# VOltageLogging sets AC or DC logging.
# Possible values are:
# MAX(AC) (default)
# AC(PH1) or AC(PH2) or AC(PH3)
# MAX(DC) or DC(ST1) or DC(ST2)
VoltLogging=MAX(AC)

In case you want something else logged
```

Save and exit, Ctrl-O and Ctrl-X

Let's place this config file in it's right spot

cp SMAspot.cfg /home/pi/smaspot/bin/Release

and test it

cd bin/Release
./SMAspot -v -u

If all went well:

```
SMAspot V2.0.6
Yet another tool to read power production of SMA solar inverters
(c) 2012-2013, SBF (http://code.google.com/p/sma-spot)
Commandline Args: -v -u
Mon Jun 10 14:30:51 2013: INFO: Starting...
sunrise: 05:24
sunset : 22:00
Connecting to XX:XX:XX:15:D3:E7 (1/10)
Initializing...
SMA netID=01
Serial Nr: XXXXABE5 (XXXXXX6197)
BT Signal=64% <<---- The strenght of the BT signal, if low, consider placing the
Pi closer to the SMA inverter
Logon OK
Local PC Time: 10/06/2013 14:30:52
Inverter Time: 10/06/2013 14:31:37
Time diff (s): -45
TZ offset (s): 7200
Device Name:      SN: 2002133758
Device Class:     Solar Inverters
Device Type:      SB1600TL
Software Version: 12.12.205.R
Serial number:    2002133758
Device Status:      OK
GridRelay Status:      ?
Energy Production:
      EToday: 1.875kWh
      ETotal: 8341.071kWh
      Operation Time: 9241.46h
      Feed-In Time  : 8945.33h
DC Spot Data:
      String 1 Pdc:  0.000kW - Udc:  0.00V - Idc: 0.000A
      String 2 Pdc:  0.000kW - Udc:  0.00V - Idc: 0.000A
AC Spot Data:
      Phase 1 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Phase 2 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Phase 3 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Total Pac   :  0.000kW
Grid Freq. : 0.00Hz
Current Inverter Time: 10/03/2013 19:33:29
Inverter Wake-Up Time: 10/03/2013 07:50:06
Inverter Sleep Time  : 10/03/2013 18:14:51
ExportSpotDataToCSV()
**********
* ArchiveDayData() *
**********
startTime = 513BBEF0 -> 10/03/2013 00:00:00
ExportDayDataToCSV()
**********
* ArchiveMonthData() *
**********
startTime = 51308A30 -> 01/03/2013 12:00:00
ExportMonthDataToCSV()
PVOutputExport()
OK 200: Added StatusMon Jun 10 14:30:54 2013: INFO: Done.
```

A data folder is created, as specified in the config file and your first data on PVOutput should be there!

Now for the magic to happen itself

The part of the automatic upload. A small script is created, which will be executed by cronjob, so smaspot is run every 5 minutes:

For update reasons this (and other scripts) are placed in a separate folder

cd /home/pi/
mkdir scripts
cd /home/pi/scripts
nano smaspot.sh

```
#!/bin/bash
cd /home/pi/smaspot/bin/Release
./SMAspot -v -u
```

After the script is written with nano, save and exit. Ctrl-O and Ctrl-X

This script needs to be executable:

sudo chmod 755 smaspot.sh

test the script:

./smaspot.sh

Start a cronjob:

crontab -e

Add this line:

```
*/5 6-23 * * * /home/pi/scripts/smaspot.sh > /dev/null
```

Press Ctrl-O, Enter (proposed filename is oké), Ctrl-X
You should see the message “crontab: installing new crontab”

This will make the above script to run every 5 minutes between 6.00 and 23.00 hours. If you want it to run for 24hrs, just replace 6-23 with *

Changelog 0.1
The extra '> /dev/null' is added, because otherwise the root will send an email every 5 minutes to user pi, just to say that everything went well. Including the complete output from ./SMAspot -v -u. By sending it to /dev/null it is considered to be lost in a big black hole.

Final

Keep an eye on PVOutput during the day. You should see the values being updated. The Pi will be doing it's work 24/7 for the rest of his life.....

For a correct shutdown:

sudo shutdown -h now

place it where ever suitable and connect again: no command needed. The cronjob remembers what to do an so the upload to PVOutput continues.

Have fun with SMAspot on your Raspberry Pi

Don't forget these:
http://www.pvoutput.org/listteam.jsp?tid=502
http://www.pvoutput.org/listteam.jsp?tid=613

Changelog 0.1

Some might experience the Pi to be stuck in the middle of something and not responding anymore. This might have to do with the wifi-connection, an usb-conflict or perhaps the BT-dongle is working it's magic. The solution for most of the time is to pull out and plug in again the power supply.

Watchdog

However a watchdog can be installed, who reboots the Pi when it is gonna hang itself and send an email when this happened. This program also works without the email notification, so the first part below is optional.

Changelog 0.4

Watchdog and it's components need to be installed on the sd-card not on the usb-stick

mail
First we have to install some additional software packages:

```
sudo apt-get install sendmail-bin
sudo apt-get install sensible-mda
sudo apt-get install mutt
```

Let's retrieve and configure the configfile .muttrc

```
cd /home/pi
wget http://cache.gawker.com/assets/images/lifehacker/2010/06/muttrc-gmail.txt
mv muttrc-gmail.txt .muttrc
nano .muttrc
```

Edit the first six 6 lines of the file to match your gmail account info. Since I do have a gmail account I didn't investigate what to change to set this up for another mail-account

Edit the file /etc/hosts:

```
sudo nano /etc/hosts
```

```
127.0.0.1       localhost
::1             localhost ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

127.0.1.1       raspberrypi
```

In this file we change the first line to:

```
127.0.0.1       raspberrypi.local. raspberrypi
```

save and exit (Ctrl+O, Ctrl-X)

Mail notification

cd /home/pi
mkdir bin
cd bin
sudo nano mailIP

In this mailIP file we enter the following:

```
#!/bin/sh
mailreciever=YOURMAIL This can be any mailaddress
today=$(date)
my_ip=`ifconfig | grep 'inet addr:'| grep -v '127.0.0.1' | cut -d: -f2 | awk
'{print $1}'`
my_pi="Your RaspberryPi has rebooted! "
message="Your Pi has rebooted at $today. Current IP address = $my_ip"
echo $message > message.txt
mutt -s "${my_pi}" ${mailreciever} < message.txt
```

Save as usual and make it executable:

sudo chmod 0755 mailIP

Test the mail notification and see if you get an email in the specified mailbox:

./mailIP

It seems nothing is happening, but you will receive 'immediately' an email in the specified mailbox. And there is a file called message.txt in the bin folder:

```
pi@raspberrypi ~/bin $ cd /home/pi/bin
pi@raspberrypi ~/bin $ ./mailIP
pi@raspberrypi ~/bin $ ls -l
total 8
-rwxr-xr-x 1 root root 336 Mar 23 19:12 mailIP
-rw-r--r-- 1 pi   pi    90 Apr 29 21:37 message.txt
```

Now we want to make sure that the script gets run when the Raspberry Pi boots.

Therefore edit the file /etc/rc.local:

sudo nano /etc/rc.local

Add this above the line with '*exit 0*':

```
sudo /home/pi/bin/mailIP &
```

Make sure the rc.local can be executed:

sudo chmod 0755 /etc/rc.local

I had to put a copy of .muttrc in the /root folder for this to work

sudo bash
cp /home/pi/.muttrc /root
exit

After the command *sudo bash*, the commandline starts with a #, enter the line *cp /home/pi/.muttrc /*root and hit enter. To go back to the normal commandline with $ you need the command *exit*

Easiest way to test all above:

sudo reboot

The real Watchdog with auto-reboot

Watchdog can also be run without the mail notification:

sudo modprobe bcm2708_wdog
sudo nano /etc/modules

Add the following:

```
bcm2708_wdog
```

save and exit (Ctrl+O, Ctrl-X)

Install the watchdog daemon:

sudo apt-get install watchdog chkconfig
chkconfig watchdog on
sudo /etc/init.d/watchdog start
sudo nano /etc/watchdog.conf

Uncomment (delete the #) the lines in this config file with:

```
max-load-1 = 24
watchdog-device = /dev/watchdog
```

Watchdog is now setup, in case you choose the mail notification option, I do hope you don't get to much mails.

Cleanup the data folder

SMAspot produces a daily, a monthly and a Spot .csv file. Last one containing very different spot values from your inverter. In case you don't need this info I wrote this little cleanup script

Changelog 0.4
Dont not forget to use the right folder navigation

cd /home/pi/scripts
nano cleanup.sh

```
#!/bin/bash
cd /home/pi/smadata/%Y
sudo rm *Spot*
```

smadata/%Y depends on your original setup. *In case of %Y ,which is now 2013, next year the script won't work unless you change it in 2014.*

Save and exit (Ctrl-O, Ctrl-X)

Make executable:

sudo chmod 755 cleanup.sh

and test:

./cleanup.sh

We edit the cronjob which runs SMAspot every five minutes, so every night at 23.30 the cleanup.sh also gets executed

crontab -e

Add the next line:

```
30 23 * * * /home/pi/scripts/cleanup.sh > /dev/null
```

Possible output is send to /dev/null to avoid growing logfiles

Press Ctrl-O, Enter (proposed filename is oké), Ctrl-X
You should see the message “crontab: installing new crontab”

Changelog 0.4

Install SMAspot on an usb-stick

Installing SMAspot and its output on an usb-stick makes the Pi to read/write less on the sd-card and thus make it less vulnerable to crash.

cd /mnt
sudo mkdir usb

Attach an usb-stick

dmesg | tail

The message you receive is about the non mounted usb-stick called *sda* (presuming this is the only attached usb-stick) otherwise it could be sdb or sdc

```
[  588.238008] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

To know which filesystem is used:

sudo blkid /dev/sda1

The result kinda looks like:

```
pi@raspberry /mnt $ blkid /dev/sda1
/dev/sda1: UUID="b395cd0c-0a2c-4cb5-9415-64564643d221" TYPE="vfat"
```

Where vfat is another name for fat32

Since the Pi has a Linux OS we are gonna change the FS on the stick to ext2

!! make sure you definitely know which device is the USB stick, since this change is gonna format everything on it!!

sudo mkfs -t ext2 /dev/sda1

Now we mount the usb-stick to it's mountpoint made earlier:

sudo mount /dev/sda1 -t ext2 /mnt/usb

Make the stick owned by the SMAspot user (pi):

sudo chown pi:pi /mnt/usb

Let's make sure the stick get automatically mounted when the Pi is started:

sudo nano /etc/fstab

Add the following:

```
/dev/sda1       /mnt/usb       ext2       rw,noatime,defaults       0       0
```

Use <TAB> between entries and because there is only 1 partition on the usb-stick, you have to use /dev/sda**1**

Let's reboot and see if the stick is automatically mounted:

sudo shutdown -r now

After login, navigate to the stick and create the folder for SMAspot

cd /mnt/usb
mkdir smaspot
cd smaspot

The rest of the installation is described above.
Don't forget to read */mnt/usb* instead of */home/pi*

Go to **

Changelog 0.5

Update procedure

In case you want to update your SMAspot installation, here are some easy steps to follow:

Go to the original folder where SMAspot was installed and create a new folder:

```
cd /home/pi
mkdir smaspot_new
cd /home/pi/smaspot_new
```

In this new folder the steps are similar to the original installation, only with the new version of SMAspot:

```
wget https://sma-spot.googlecode.com/files/SMAspot_SRC_new_Linux_Win32.tar
tar -xvf SMAspot_SRC_new_Linux_Win32.tar
```

Compile the new SMAspot and afterwards in case it went oké remove the source code:

```
make release
rm SMAspot_SRC_new_Linux_Win32.tar
```

After compiling is done, there is a new folder in your smaspot_new folder called bin, and in bin there is the subfolder Release. In the Release folder is the actual SMAspot software. Make sure you are in the smaspot_new folder (if you didn't move you are still there):

Make sure the config file hasn't been changed since last time and if so you can copy the original SMAspot.cfg to the new version:

```
cd /home/pi/smaspot/bin/Release
cp SMAspot.cfg /home/pi/smaspot_new/bin/Release/
```

and test it

```
cd /home/pi/smaspot_new/bin/Release
./SMAspot -v -u
```

If all went well:

```
SMAspot V2.0.new
Yet another tool to read power production of SMA solar inverters
(c) 2012-2013, SBF (http://code.google.com/p/sma-spot)
Commandline Args: -v -u
Mon Jun 10 14:30:51 2013: INFO: Starting...
sunrise: 05:24
sunset : 22:00
Connecting to XX:XX:XX:15:D3:E7 (1/10)
Initializing...
SMA netID=01
Serial Nr: XXXXABE5 (XXXXXX6197)
BT Signal=64% <<---- The strenght of the BT signal, if low, consider placing the
Pi closer to the SMA inverter
Logon OK
Local PC Time: 10/06/2013 14:30:52
Inverter Time: 10/06/2013 14:31:37
```

```
Time diff (s): -45
TZ offset (s): 7200
Device Name:      SN: 2002133758
Device Class:     Solar Inverters
Device Type:      SB1600TL
Software Version: 12.12.205.R
Serial number:    2002133758
Device Status:      OK
GridRelay Status:      ?
Energy Production:
      EToday: 1.875kWh
      ETotal: 8341.071kWh
      Operation Time: 9241.46h
      Feed-In Time  : 8945.33h
DC Spot Data:
      String 1 Pdc:  0.000kW - Udc:  0.00V - Idc: 0.000A
      String 2 Pdc:  0.000kW - Udc:  0.00V - Idc: 0.000A
AC Spot Data:
      Phase 1 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Phase 2 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Phase 3 Pac :  0.000kW - Uac:  0.00V - Iac: 0.000A
      Total Pac   :  0.000kW
Grid Freq. : 0.00Hz
Current Inverter Time: 10/03/2013 19:33:29
Inverter Wake-Up Time: 10/03/2013 07:50:06
Inverter Sleep Time  : 10/03/2013 18:14:51
ExportSpotDataToCSV()
********************
* ArchiveDayData() *
********************
startTime = 513BBEF0 -> 10/03/2013 00:00:00
ExportDayDataToCSV()
**********************
* ArchiveMonthData() *
**********************
startTime = 51308A30 -> 01/03/2013 12:00:00
ExportMonthDataToCSV()
PVOutputExport()
OK 200: Added StatusMon Jun 10 14:30:54 2013: INFO: Done.
```

A data folder is created, as specified in the config file and your first data with the new version on PVOutput should be there!

Option
There are two ways of letting the Pi use the new installed version of SMAspot. One involves editing the script(1) that runs SMAspot the other involves some moving(2) around files. Please read further for both of them to be explained
/Option

(1)

Now it's time to let your Pi know, that it needs to use the new version:

crontab -e

place an # in front of this line:

```
*/5 6-23 * * * /home/pi/scripts/smaspot.sh > /dev/null
```

Press Ctrl-O, Enter (proposed filename is oké), Ctrl-X
You should see the message “crontab: installing new crontab”. Because of the # in front of the line this new cronjob does nothing.

Now we are gonna edit the original script:

cd /home/pi/scripts
nano smaspot.sh

```
#!/bin/bash
cd /home/pi/smaspot_new/bin/Release
./SMAspot -v -u
```

change the foldername to the new one and save on exit

Start the cronjob again:

crontab -e

Remove the # in front of this line:

```
*/5 6-23 * * * /home/pi/scripts/smaspot.sh > /dev/null
```

Press Ctrl-O, Enter (proposed filename is oké), Ctrl-X
You should see the message “crontab: installing new crontab”

The new cronjob will now use the newly installed version of SMAspot. If you satisfied with the new build and everything goes oké you can simply remove the old SMAspot folder, if not change the script back to the way it was and your old version of SMAspot will be used.

This last commando completely removes the (original) smaspot folder, it's subfolder and all content within. Make sure you point to the right folder!

cd /home/pi
sudo rm -r smaspot/

(2)
If correct you are still in this folder /home/pi/smaspot_new/bin/Release otherwise:

cd /home/pi/smaspot_new/bin/Release

Copy the binary (with a new name) to the other smaspot folder (or perhaps an other working directory):

cp -a SMAspot /home/pi/smaspot/bin/Release/smaspot.new

move to this folder and move the old binary out of the way and the new binary in to place:

mv SMAspot smaspot.old
mv smaspot.new SMAspot

Since the cronjob uses this old location, nothing has to change on that part and your Pi will continue as usual with the newly installed version of SMAspot. If you are satisfied with the new build and everything goes oké you can simply remove the newly created SMAspot folder, if not reverse what you did:

mv SMAspot smaspot.new
mv SMAspot.old SMAspot

This last commando completely removes the new smaspot folder, it's subfolder and all content within. Make sure you point to the right folder!

cd /home/pi
sudo rm -r smaspot_new/

Thanx to Pierre Gorissen (ictoblog.nl/raspberry-pi), s.b.f (smaspot owner) and others