Changelog of SBFspot install on a Raspberry Pi

```
Version 1.0 - 21/09/2014 (Snowmiss)
        Edited content. Complete howto for installing SBFspot V3.x

Version 1.1 - 15/08/2015 (SBF)
        Typo corrected (page 4)

Version 1.2 - 29/03/2015 (SBF)
        Some minor corrections - See https://sbfspot.codeplex.com/discussions/652997
```

Introduction

This document includes all steps on how to install SBFspot https://sbfspot.codeplex.com/ on a Raspberry Pi http://www.raspberrypi.org/

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 options (not needed).

Preparing the Raspberry Pi for SBFspot

Since the start of this document, a lot has changed. For instructions on how to install an OS to your PI, take a look at : http://www.raspberrypi.org/ After the initial setup of your Pi you may use a remote control over your Pi. Windows users will need PuTTY to connect to the Pi: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
When using Linux, go to a terminal and type *ssh pi@192.168.x.x*

I recommend giving the Pi a fixed IP address. It 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.

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

How to install SBFspot on the Raspberry Pi?

This is what it is all about, but first things first. Unlike SBFspot 2.x, version 3.x no longer uploads to PVoutput.org directly. This functionality is now in the hands of an upload service (Windows) or a daemon (Linux). SBFspot retrieves the data from the inverter and stores it in a SQLite or MySQL database. The service/daemon reads the data from the database and sends it to Pvoutput.org
Since SQLite is a simple system without a lot of bells and whistles (only 1 file) it is the preferred choice for use on a Raspberry Pi and the only option discussed in this document

---

Backup

This installation procedure should not touch existing ones as we will use other directories, but is's always a good idea to have a backup. So, before proceeding with this SBFspot installation, make a copy of your .cfg file(s) and other things you want to keep. This isn't necessary if this is your first installation though ;-)

Install a bluetooth driver, it 's not there by default:

sudo apt-get --no-install-recommends install bluetooth libbluetooth-dev

On screen you'll see:

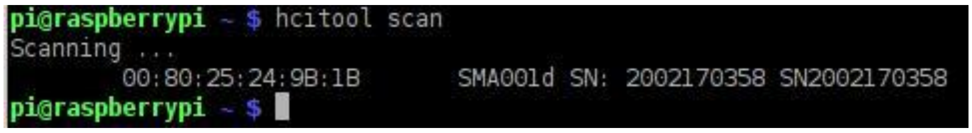
```
pi@raspberrypi ~ $ sudo apt-get install –no-install-recommends bluetooth liblbluetooth-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
…..
and maybe a lot more, depending on how up-to-date your Raspberry Pi is
```

Remark: even if you're using Speedwire, you need to install this bluetooth software.

Check for bluetooth connectivity with the SMA converter:

hcitool scan

You should see something like this:



```
pi@raspberrypi ~ $ hcitool scan
Scanning ...
        00:80:25:24:9B:1B       SMA001d SN: 2002170358 SN2002170358
pi@raspberrypi ~ $ 
```

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

Install boost:
sudo apt-get install libboost-all-dev

Install SQLite:
sudo apt-get install sqlite3 libsqlite3-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 SBFspot on the Pi

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

First we create the folders for SBFspot and it's data then switch to that folder

```
cd /home/pi
mkdir sbfspot.3
mkdir smadata
mkdir smadata/logs
cd sbfspot.3
```

**

Now we download the source code **to our main computer** and copy the sourcecode to /home/pi/sbfspot.3

For Linux you can put the downloaded source code to your Pi, using:

```
scp /home/yourname/SBFspot_SRC_301_Linux_Win32.tar.gz pi@192.168.x.x:/home/pi/sbfspot.3
```

Windows users use PuTTY or get a copy of psftp.exe (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.htm) and copy it to your %SystemRoot%\system32 folder

Open a command box, change directory to the folder containing the downloaded source code and start PSFTP:

```
C:\TEMP>psftp.exe pi@192.168.x.x
Using username "pi"
pi@192.168.1.2's password:
Remote working directory is /home/pi
psftp> cd sbfspot.3
psftp> put SBFspot_SRC_301_Linux_Win32.tar.gz
local:SBFspot_SRC_301_Linux_Win32.tar.gz =>
remote:/home/pi/SBFspot.3/SBFspot_SRC_301_Linux_Win32.tar.gz
psftp> exit
```

Now your file should be copied into /home/pi/sbfspot.3 folder and we go back to the commandline of the Pi

Let's untar the sourcecode:

```
cd /home/pi/sbfspot.3
```

tar -xvf SBFspot_SRC_301_Linux_Win32.tar.gz

and compile SBFspot with Sqlite support in /usr/local/bin/sbfspot.3:

cd /home/pi/sbfspot.3/SBFspot
sudo make install_sqlite

When you see this warning, simply ignore it: *"Warning: swp{b} use is deprecated for this architecture"*. Don't worry too much about the SWP warnings.

You should see this on your screen:
```
test -d bin/Release_SQlite || mkdir -p bin/Release_SQLite
test -d obj/Release_SQlite || mkdir -p obj/Release_SQLite
g++ -Wall -O2 -DUSE_SQLITE -c boost_ext.cpp -o obj/Release_SQLite/boost_ext.o
g++ -Wall -O2 -DUSE_SQLITE -c db_SQLite.cpp -o obj/Release_SQLite/db_SQlite.o
g++ -Wall -O2 -DUSE_SQLITE -c db_SQLite_Export.cpp -o
obj/Release_SQLite/db_SQLite_Export.o
g++ -Wall -O2 -DUSE_SQLITE -c misc.cpp -o obj/Release_SQLite/misc.o
g++ -Wall -O2 -DUSE_SQLITE -c strptime.cpp -o obj/Release_SQLite/strptime.o
g++ -Wall -O2 -DUSE_SQLITE -c sunrise_sunset.cpp -o
obj/Release_SQLite/sunrise_sunset.o
g++ -Wall -O2 -DUSE_SQLITE -c SBFNet.cpp -o obj/Release_SQLite/SBFNet.o
g++ -Wall -O2 -DUSE_SQLITE -c Bluetooth.cpp -o obj/Release_SQLite/Bluetooth.o
g++ -Wall -O2 -DUSE_SQLITE -c CSVexport.cpp -o obj/Release_SQLite/CSVexport.o
g++ -Wall -O2 -DUSE_SQLITE -c Ethernet.cpp -o obj/Release_SQLite/Ethernet.o
g++ -Wall -O2 -DUSE_SQLITE -c Eventdata.cpp -o obj/Release_SQLite/Eventdata.o
g++ -Wall -O2 -DUSE_SQLITE -c Archdata.cpp -o obj/Release_SQLite/Archdata.o
g++ -Wall -O2 -DUSE_SQLITE -c SBFspot.cpp -o obj/Release_SQLite/SBFspot.o
/tmp/ccmAjspk.s:32096: Warning: swp{b} use is deprecated for ARMv6 and ARMv7
** this warning is shown a lot of times **
g++ -Wall -O2 -DUSE_SQLITE  -c TagDefs.cpp -o obj/Release_SQLite/TagDefs.o
g++  -o bin/Release_SQLite/SBFspot obj/Release_SQLite/boost_ext.o
obj/Release_SQLite/db_SQLite.o obj/Release_SQLite/db_SQLite_Export.o
obj/Release_SQLite/misc.o obj/Release_SQLite/strptime.o
obj/Release_SQLite/sunrise_sunset.o obj/Release_SQLite/SBFNet.o
obj/Release_SQLite/Bluetooth.o obj/Release_SQLite/CSVexport.o
obj/Release_SQLite/Ethernet.o obj/Release_SQLite/EventData.o
obj/Release_SQLite/ArchData.o obj/Release_SQLite/SBFspot.o
obj/Release_SQLite/TagDefs.o   -s -lbluetooth -lboost_date_time -lsqlite3
cp TagList*.txt bin/Release_SQLite
cp date_time_zonespec.csv bin/Release_SQLite
test -d /usr/local/bin/sbfspot.3 || mkdir -p /usr/local/bin/sbfspot.3
test -f /usr/local/bin/sbfspot.3/SBFspot.cfg || cp SBFspot.cfg
/usr/local/bin/sbfspot.3/
cp -R bin/Release_SQLite/* /usr/local/bin/sbfspot.3/
```

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

rm SBFspot_SRC_301_Linux_Win32.tar

Create an empty database

```
cd /home/pi/smadata
sqlite3 SBFspot.db < /home/pi/sbfspot.3/SBFspot/CreateSQLiteDB.sql
```

This should create /home/pi/smadata/SBFspot.db

As a quick test, you can already execute a simple query:

```
sqlite3 SBFspot.db
SQLite version 3.7.13 2012-06-11 02:05:22
Enter “.help” for instructions
Enter SQL statements terminated with a “;”
sqlite> select * from config;
SchemaVersion|1
sqlite> .quit
```

Configuration of SBFspot

The best way is to start with a new config file. Normally, there should be a default SBFspot.cfg in the install folder (/usr/local/bin/sbfspot.3)

```
sudo nano /usr/local/bin/sbfspot.3/SBFspot.cfg
```

```
################################################################
#                     ____  ____  _____                _
#                    / ___|| __ )|  ___|__ _ __   ___ | |_
#                    \___ \|  _ \| |_ / __| '_ \ / _ \| __|
#                     ___) | |_) |  _|\__ \ |_) | (_) | |_
#                    |____/|____/|_|  |___/ .__/ \___/ \__|
#                                         |_|
#
#
#   SBFspot.cfg - Configuration file for SBFspot.exe
#   SBFspot - Yet another tool to read power production of SMA® solar inverters
#   (c)2012-2014, SBF
#
#   DISCLAIMER:
#   A user of SBFspot 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 and Speedwire are registered trademarks of SMA Solar Technology AG
######################################################################

# SMA Inverter's Bluetooth address
# Windows: smaspot -scan
# Linux  : hcitool scan
# IMPORTANT FOR SPEEDWIRE USERS: COMMENT OUT BTADDRESS (PUT # IN FRONT)
BTAddress=00:00:00:00:00:00
```

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

```
# SMA Inverter's Speedwire IP address
# If IP_Address is not set or is 0.0.0.0 SBFspot will try to detect the 
speedwire inverter by broadcast
# If IP_Address is set to a valid IP, SBFspot will try to connect directly to 
that IP without broadcast detection
#IP_Address=0.0.0.0
```

Above has to be uncommented if you are a speedwire user

```
# User password (default 0000)
Password=0000

# MIS_Enabled (Multi Inverter Support: Default=0 Disabled)
# +------------+-------+-------------+
# | #Inverters | NetID | MIS_Enabled |
# +------------+-------+-------------+
# |      1     |   1   | Don't Care  |
# +------------+-------+-------------+
# |      1     |   >1  |      0      |
# +------------+-------+-------------+
# |      >1    |   >1  |      1      |
# +------------+-------+-------------+
MIS_Enabled=0

# Plantname
Plantname=MyPlant
```

spaces are allowed

```
# OutputPath (Place to store CSV files)
# 
# Windows: C:\Users\Public\SMAdata\%Y
# Linux  : /home/pi/smadata/%Y
# %Y %m and %d will be expanded to Year Month and Day
OutputPath=/home/pi/smadata/%Y
```

As example, you already created the folder smadata.
%Y can be left out if you don't want a Year folder.

```
# OutputPathEvents (Place to store CSV files for events)
# If omitted, OutputPath is used
OutputPathEvents=/home/pi/smadata/%Y/Events

# 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 1 = On)
# If set to 1 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

# Locale
# Translate Entries in CSV files
# Supported locales: de-DE;en-US;fr-FR;nl-NL;es-ES;it-IT
# Default en-US
Locale=en-US

# Timezone
# Select the right timezone in date_time_zonespec.csv
# e.g. Timezone=Europe/Brussels
Timezone=Europe/Brussels

# BTConnectRetries
# Number of Bluetooth Connection attempts (1-15; Default=10)
BTConnectRetries=10

###########################
### 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

# CSV_ExtendedHeader (default 1 = On)
# Enables or disables the SMA extended header info (8 lines)
# isep=;
# Version CSV1|Tool SBFspot|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

# CSV_Spot_WebboxHeader (Default 0 = Off)
# When enabled, use Webbox style header (DcMs.Watt[A];DcMs.Watt[B]...)
CSV_Spot_WebboxHeader=0

###########################
###   SQL DB Settings           ###
###########################

# SQLite
# SQL_Database (Fullpath to SQLite DB)
# Windows: C:\Users\Public\SMAdata\SBFspot.db
# Linux   : /home/pi/smadata/SBFspot.db
SQL_Database=/home/pi/smadata/SBFspot.db

# MySQL
#SQL_Database=SBFspot
#SQL_Hostname=<Network Name> or <IP-address>
#SQL_Username=SBFspotUser
#SQL_Password=SBFspotPassword
```

Save and exit, Ctrl-O and Ctrl-X

Testing

/usr/local/bin/sbfspot.3/SBFspot -v -finq -nocsv

Watch out for “Error: Can't open SCLite db [/home/pi/smadata/SBFspot.db]: unable to open database file” in the output

If all went well:

```
SBFspot V3.0.1
Yet another tool to read power production of SMA solar inverters
(c) 2012-2014, SBF (https://sbfspot.codeplex.com)
Compiled for Linux 32 bit

Commandline Args: -v
Reading config '/usr/local/bin/sbfspot.3/SBFspot.cfg'
Tue Sep 23 13:51:33 2014: INFO: Starting...
sunrise: 05:31
sunset : 17:38
Connecting to 00:80:25:24:9B:1B (1/10)
Initializing...
SUSyID: 125 - SN: 852675450 (0x32D2CB7A)
SMA netID=01
Serial Nr: 7756B1F6 (2002170358)
BT Signal=69%
Logon OK
Local Time: 23/09/2014 13:51:35
TZ offset (s): 0 - DST: Off
SUSyID: 99 - SN: 2002170358
```

```
Device Name:      SN: 2002170358
Device Class:     Solar Inverters
Device Type:      SB 1600TL-10
Software Version: 12.12.208.R
Serial number:    2002170358
SUSyID: 99 - SN: 2002170358
Device Status:      Ok
SUSyID: 99 - SN: 2002170358
Device Temperature: 0.0° C
SUSyID: 99 - SN: 2002170358
GridRelay Status:      ?
SUSyID: 99 - SN: 2002170358
Pac max phase 1: 1600W
Pac max phase 2: 0W
Pac max phase 3: 0W
SUSyID: 99 - SN: 2002170358
Energy Production:
        EToday: 2.050kWh
        ETotal: 3585.376kWh
        Operation Time: 10319.38h
        Feed-In Time  : 8355.91h
SUSyID: 99 - SN: 2002170358
DC Spot Data:
        String 1 Pdc:   0.296kW - Udc: 175.00V - Idc:  1.695A
        String 2 Pdc:   0.000kW - Udc:   0.00V - Idc:  0.000A
SUSyID: 99 - SN: 2002170358
AC Spot Data:
        Phase 1 Pac :   0.000kW - Uac: 234.30V - 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   :   0.272kW
SUSyID: 99 - SN: 2002170358
Grid Freq. : 49.96Hz
SUSyID: 99 - SN: 2002170358
Current Inverter Time: 23/09/2014 13:51:54
Inverter Wake-Up Time: 23/09/2014 13:51:54
Inverter Sleep Time  : 23/09/2014 13:51:55
ExportSpotDataToCSV()
********************
* ArchiveDayData() *
********************
startTime = 5420B800 -> 23/09/2014 00:00:00
ExportDayDataToCSV()
**********************
* ArchiveMonthData() *
**********************
startTime = 54045FC0 -> 01/09/2014 12:00:00
ExportMonthDataToCSV()
Reading events: 2014-Sep-01
ExportEventsToCSV()
Tue Sep 23 13:51:37 2014: INFO: Done.
```

Now we can check if there is something in the db (don't forget the ';' at the end of each query):

```
cd /home/pi/smadata
sqlite3 SBFspot.db
SQLite version 3.7.13 2012-06-11 02:05:22
Enter “.help” for instructions
Enter SQL statements terminated with a “;”
sqlite> select * from vwspotdata;
2014-09-21 15:11:10|2014-09-21 15:10:00|SN: 2002170358|SB 1600TL-
10|2002170358|208|0|1.213|0.0|172.0|0.0|0|0|0|0.0|0.0|0.0|233.8|0.0|0.0|208|0|0.0|4412|3
580032|50.01|10297.1|8336.77|68.6274|OK|?|0.0
sqlite> select * from inverters;
2002170358|SN: 2002170358|SB 1600TL-
10|12.12.208.R|1411312270|187|4.412|3580.03|10297.1|8336.77|OK|?|0.0
sqlite> .quit
```

Remark: make sure you are in the smadata folder (or provide the full path to the folder). If not, you create an empty db

If you're not too familiar with SQL syntac, there are plenty of examples and tutorials on the internet. For automation and fine-tuning of SBFspot, I refer to the online documentation: https://sbfspot.codeplex.com/documentation

5 Minute data

A small script is created, which will be executed by cronjob, so SBFspot 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 SBFspot.sh
```

```
#!/bin/bash
/usr/local/bin/sbfspot.3/SBFspot -v
```

Extra options besides -v are:

```
SBFspot [-options]
 -scan          Scan for bluetooth enabled SMA inverters.
 -d#            Set debug level: 0-5 (0=none, default=2)
 -v#            Set verbose output level: 0-5 (0=none, default=2)
 -ad#           Set #days for archived daydata: 0-300
                0=disabled, 1=today (default), ...
 -am#           Set #months for archived monthdata: 0-300
                0=disabled, 1=current month (default), ...
 -ae#           Set #months for archived events: 0-300
                0=disabled, 1=current month (default), ...
 -cfgX.Y        Set alternative config file to X.Y (multiple inverters)
 -u             Upload to online monitoring system (see config file)
 -finq          Force Inquiry (Inquire inverter also during the night)
 -q             Quiet (No output)
 -nocsv         Disables CSV export (Overrules CSV_Export in config)
 -nosql         Disables SQL export
 -sp0           Disables Spot.csv export
 -installer     Login as installer
 -password:xxxx Installer password
 -loadlive      Use predefined settings for manual upload to pvoutput.org
```

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

This script needs to be executable:

```
sudo chmod 755 SBFspot.sh
```

test the script:

```
./SBFspot.sh
```

Start a cronjob:

```
crontab -e
```

Add this line:

```
*/5 6-23 * * * /home/pi/scripts/SBFspot.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 *

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 ./SBFspot -v. By sending it to /dev/null it is considered to be lost in a big black hole.

Uploading to PVOutput with UploadDaemon and SQLite

Install curl:
sudo apt-get install libcurl3-dev

Compile and install SBFspotUploadDaemon with SQLite support in /usr/local/bin/sbfspot.3:
cd /home/pi/sbfspot.3/SBFspotUploadDaemon
sudo make install_sqlite

```
test -d bin/Release_SQLite || mkdir -p bin/Release_SQLite
test -d obj/Release_SQLite/SBFspot || mkdir -p obj/Release_SQLite/SBFspot
test -d obj/Release_SQLite || mkdir -p obj/Release_SQLite
test -d obj/Release_SQLite/SBFspotUploadCommon || mkdir -p
obj/Release_SQLite/SBFspotUploadCommon
g++ -Wall -DUSE_SQLITE  -I../SBFspot -c ../SBFspot/db_SQLite.cpp -o
obj/Release_SQLite/SBFspot/db_SQLite.o
g++ -Wall -DUSE_SQLITE  -I../SBFspot -c main.cpp -o obj/Release_SQLite/main.o
g++ -Wall -DUSE_SQLITE  -I../SBFspot -c ../SBFspotUploadCommon/Configuration.cpp -o
obj/Release_SQLite/SBFspotUploadCommon/Configuration.o
g++ -Wall -DUSE_SQLITE  -I../SBFspot -c ../SBFspotUploadCommon/PVOutput.cpp -o
obj/Release_SQLite/SBFspotUploadCommon/PVOutput.o
g++ -Wall -DUSE_SQLITE  -I../SBFspot -c ../SBFspotUploadCommon/CommonServiceCode.cpp -o
obj/Release_SQLite/SBFspotUploadCommon/CommonServiceCode.o
g++  -o bin/Release_SQLite/SBFspotUploadDaemon obj/Release_SQLite/SBFspot/db_SQLite.o
obj/Release_SQLite/main.o obj/Release_SQLite/SBFspotUploadCommon/Configuration.o
obj/Release_SQLite/SBFspotUploadCommon/PVOutput.o
obj/Release_SQLite/SBFspotUploadCommon/CommonServiceCode.o   -s  -lcurl -lsqlite3
test -d /usr/local/bin/sbfspot.3 || mkdir -p /usr/local/bin/sbfspot.3
test -f /usr/local/bin/sbfspot.3/SBFspotUpload.cfg || cp SBFspotUpload.cfg
/usr/local/bin/sbfspot.3/
cp -R bin/Release_SQLite/* /usr/local/bin/sbfspot.3/
```

Edit the UploadDaemon config:

sudo nano /usr/local/bin/sbfspot.3/SBFspotUpload.cfg

```
################################################################################
#  SBFspotUpload.cfg - Configuration file for SBFspotUploadService/Daemon
#  (c)2012-2014, SBF (https://sbfspot.codeplex.com)
#
#  DISCLAIMER:
#  A user of SBFspotUploadService/Daemon 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.
#
################################################################################


################################
### Log Settings             ###
################################
# Windows: C:¥Users¥Public¥SMAdata¥Logs
# Linux  : /home/pi/smadata/logs
#LogDir=C:¥Users¥Public¥SMAdata¥Logs
LogDir=/home/pi/smadata/logs

################################
### PVoutput Upload Settings ###
################################
#PVoutput_SID
#Map inverters to PVoutput System ID's
#PVoutput_SID=SerialNmbrInverter_1:SID_1,SerialNmbrInverter_2:SID_2
#e.g. PVoutput_SID=200212345:4321
PVoutput_SID=200212345:4321
```

Of course yours is needed

Remark: first the serial number followed by SystemID

```
#PVoutput_Key
#Sets PVoutput API Key
PVoutput_Key=12347babcge2a9d50aklm0ael1t323737f6d1ab3a8f1337

################################
### SQL DB Settings          ###
################################
# SQL_Database (Fullpath to SQLite DB)
# Windows: C:¥Users¥Public¥SMAdata¥SBFspot.db
# Linux  : /home/pi/smadata/SBFspot.db
#SQL_Database=C:¥Users¥Public¥SMAdata¥SBFspot.db
SQL_Database=/home/pi/smadata/SBFspot.db

# Reserved for MySQL
#SQL_Database=SBFspot
#SQL_Hostname=<Network Name> or <IP-address>
#SQL_Username=SBFspotUser
#SQL_Password=SBFspotPassword
```

Testing

Start the daemon:
/usr/local/bin/sbfspot.3/SBFspotUploadDaemon

If all goes well, you see nothing on the commandline but there should be a logfile in /home/pi/smadata/logs:

```
[09:49:47] INFO: : Starting...
[09:49:47] INFO: : Uploading 30 datapoints, starting with
20140922,06:20,3580445,0 => OK (200)
[09:50:54] INFO: : Uploading 12 datapoints, starting with
20140922,08:50,3580788,264 => OK (200)
[10:13:02] INFO: : Uploading 30 datapoints, starting with
20140918,06:10,3563122,0 => OK (200)
[10:14:09] INFO: : Uploading 30 datapoints, starting with
20140916,06:05,3551781,0 => OK (200)
[10:15:16] INFO: : Uploading 30 datapoints, starting with
20140916,08:35,3552119,420 => OK (200)
[10:16:22] INFO: : Uploading 30 datapoints, starting with
20140916,11:05,3553541,864 => OK (200)
```

A few minutes later, your Pvoutput graphs should be updated.

Since SBFspot is autorun with a cronjob every 5 minutes we let the UploadDaemon auto start:

sudo nano /etc/rc.local

Add these lines:

```
#Start SBFspotUploadDaemon
sudo /usr/local/bin/sbfspot.3/SBFspotUploadDaemon
```

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 with SBFspot and the daemon is automatically started.

Have fun with SBFspot on your Raspberry Pi

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

http://www.pvoutput.org/listteam.jsp?tid=613

Cleanup

When everything is running fine, you can remove the source folder:
sudo rm -r /home/pi/sbfspot.3/

Cleanup the data folder

SBFspot 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

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 2014, next year the script won't work unless you change it in 2015.

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

Make executable:

sudo chmod 755 cleanup.sh

and test:

./cleanup.sh

We edit the cronjob which runs SBFspot 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”

** Install SBFspot on an usb-stick **

Installing SBFspot 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 SBFspot 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 SBFspot

```
cd /mnt/usb
mkdir sbfspot.3
mkdir /smadata
mkdir /smadata/logs
cd sbfspot.3
```

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

Go to **

Thanx to SBF (SBFspot owner, Pierre Gorissen (ictoblog.nl/raspberry-pi) and others.