History, how to and troubleshooting guide v0.1

# Gas Sensor Shield version history

## AirSensEUR R10

|  |  |
| --- | --- |
| 1. Revision date: February 2014, quantity delivered: 5pcs 2. Tested: Lab controlled chamber 3. Main problems found: Low resolution; high temperature drift; problems when connecting to some Windows XP as the serial line was not properly detected; limited to a specific sensor footprint/brand   (the picture is taken from an not completely assembled prototype used for firmware development only) |  |

## AirSensEUR R20

|  |  |
| --- | --- |
| 1. Revision date: May 2014, quantity delivered: 5pcs 2. Tested: Lab controlled chamber 3. Main problems found: Noise affecting measurements; limited to a specific sensor footprint/brand |  |

## AirSensEUR R22

|  |  |
| --- | --- |
| 1. Revision date: September 2015, quantity delivered: 5pcs 2. Tested: Lab controlled chamber / field campaign 3. Main problems found: footprint problems with some specific sensors/brands; mechanical incompatibilities with custom enclosure |  |

## AirSensEUR R23

|  |  |
| --- | --- |
| 1. Revision date: February 2016, quantity delivered: > 30pcs 2. Tested: Lab controlled chamber / field campaign 3. Main problems found: temperature, pressure and humidity sensors discontinued; suggested 4. improvements for 4 leads chemical sensors (not yet officially supported) |  |

# Host version history

## Hardware

I know we have 2 versions at least, the second one having an improved battery manager

### Host R1.1

|  |  |
| --- | --- |
| Test board: December 2015, 1 pcs |  |

### Host R1.2

|  |  |
| --- | --- |
| 1. Revision date: February 2016, quantity delivered: 5 pcs 2. Added filters on USB port. 3. Shape modified to have the GPRS modem at the bottom of the shield, to avoid magnetic interference with the sensors | Air quality monitoring host board |

### Host R1.3

|  |  |
| --- | --- |
| 1. Revision date: May 2016, quantity delivered: > 30pcs 2. Improved battery manager, with new step up converter giving the possibility to power a MOx sensor mounted on aI2C board (VZ89 SGX Sensortech). | C:\Users\gerbomi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Electronic parts.jpg |

### Host R1.4

In questo periodo ho portato avanti la migrazione del sistema di comunicazione da point to point a point to multipoint.... e sembra funzionare!

Potete vedere la foto allegata: un AirSensEUR Host R1.2 (come i vostri) ha collegato un adattatore verso il SerialBus. Ad esso e' collegato, con tre fili, l'adattatore che converte un vecchio Shield rendendolo compatibile con il SerialBus.

Ho modificato gli applicativi Java per gestire il nuovo protocollo multi punto. Ho poi implementato un firmware che gira sull'adattatore. In questo momento l'adattatore e' impostato per lavorare come identificativo "boardID = 0" ma e' facile modificarlo attraverso gli switch presenti sull'adattatore stesso.

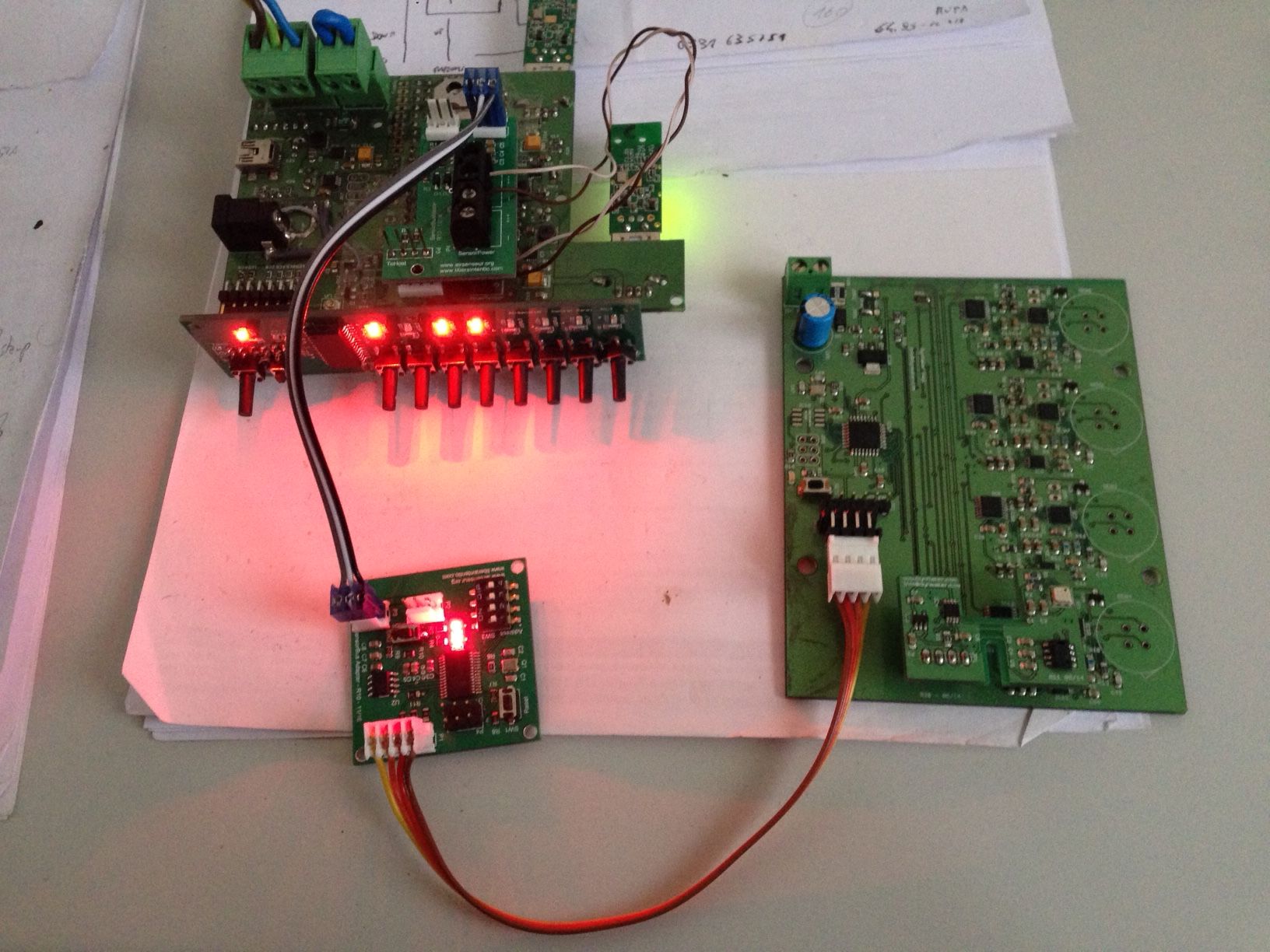
Come potete vedere, lo Shield non riceve piu' l'alimentazione dal connettore "solito" ma tutto viene portato attraverso il SerialBus (questo e' un esperimento ed e' da provare sul campo).

Ho trovato un unico problema: l'Host R1.2 montava un alimentatore per lo shield un po' problematico... ho risolto cortocircuitando un componente. Sembra che funzioni, ora.

Tra le modifiche che ho fatto al software c'e' ora la lettura del nome dei sensori direttamente dalle impostazioni memorizzato sullo shield tramite il Java Panel. In questo modo il nome dei sensori e' sempre identico a quello memorizzato in fase di configurazione dello shield. Non viene piu' letto dal sensor.properties.

La cosa si puo' estendere anche agli altri parametri dal front end, se puo' interessare.

Ora che ho verificato il funzionamento del tutto mi dedico alla parte del sensore PM.



## OS Software

### Version 0.1

1. Adding driver for Key dongle Huawei E303 (Alex) - Huawei 2511
2. Adding driver for Key dongle Huawei E303 (Jan) - Huawei E303- H-2
3. Kernel fix for battery charger

### Version 0.2

1. Solving bug: problem with system date not updating that crashes of system
2. Non official release for RIVM: WIFI connection for datapush instead of GPRS. As the Host is a full working Linux device, as soon as you will be able to have a stable network connection and you install/configure the proper services. In a recent campaign with 10 units, we help the user to implement VPNs through an external server. Each box was connected 24/7 to Internet through WiFi dongles linked to their private access point. We installed a VPN client and certificates on each AirSensEUR box and instruct to point to an external Linux server we handle. Each unit was accessible 24/7 through this server. This allows for remote management through SSH. When equipped with GPRS dongles, instead, to save power we decided to have the dongle switched off except when pushing data to the external server. In this type of installation the unit is not connected to Internet but only when required.

### Version 0.3 (see Updating the SD to the system OS 0.3:)

1. Support InfluxDB 1.0
2. New field "calibrated" in airsenseur.db controlled by AirSensEURDataAggregator and AirSensEURDataPush
3. Bootloader updated to light power LED del power as soon as AirSensEUR is switch on.

### Version 0.4, not released yet

1. Solved bug: when a blank space is inserted in the sensor names of sensor.properties, the data push return an error.
2. A new jar file has been released which updates the DataPush for InfluxDB with the sensor names containing blank spaces
3. Additionally, the HTTPS connection has been implemented in the AirSensEURDataPush java program.

# Sensor box version history

|  |  |
| --- | --- |
| 3D printed box, May 2016 about 20 pcs | C:\Users\gerbomi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\DSC_0076.jpg |
| Teflon box; August 2016 about 25 pcs | C:\Users\gerbomi\AppData\Local\Microsoft\Windows\INetCache\Content.Word\IMG_4667.jpg |

# New features added in the next version (Version 0.5?)

## Hardware

Adding a PM sampler. We're working on a second type of Shield targeted to PM measurements (and VOC?). The idea is to interface to PM sensors available in the market and transmit results to the Host through a LIN data bus (this is the reason we're migrating to a simple serial line to something more flexible). Some developments has been already started and we're investigating about PM sensors available in the market as we would implement something "flexible" and not targeted to a specific brand/type. May you probably address us to some you already use?

Implement a LIN Bus able to control several shields

Adding MOx support (partially added since Host version R1.3)

Adding support for amperometric sensor with 4 electrodes (partially tested in shield version R2.3)

## Software

New shield firmware, transfer to host, database and datapush for the OPC.

New shield firmware, transfer to host, database and datapush for the MOx sensor

Remote connection (VPN connection) for managing AirSensEUR (ssh and Web interface), (inform B. Wastine Air Normand and Sean Kahn UNEP) (See 6.8). Added on 16-12-15: Ho installato un certificato VPN sulla microSD In questo modo, tutte le volte che viene fatto un data push si instaura una VPN temporanea con il server sossvr1.liberaintentio.com e, da li', e' possible connettersi verso la box in SSH. Questo continuerà a funzionare anche quando sarà li' da voi. Se vuoi si puo' estendere a tutte le altre box... cosi' almeno abbiamo la possibilita' di monitorarle e aggiornarle da remoto (se la connessione GPRS avviene correttamente) per lo meno negli istanti di tempo in cui viene fatto il push. Per "aumentare" la finestra, ho aggiunto un "sleep 600" all'interno del runDataPush, poco prima di spegnere il dongle USB...In questo modo la finestra temporale e' lunga almeno 10 minuti.

When configuring the sensor shield with the JAVAPanelConfiguration, the file sensor.properties is automatically created.

In the local WEB interface, it is no more possible to edit the name of sensors that goes into sensor.properties, and in airsenseur.db to avoid confusion. The name of the sensor can only be changed with the JAVAControl Panel for example for a complete change of sensors. Reason : avoid confusion during data treatment and visualisation in grafana between old and new gas sensors. The change of the names of sensor should only be possible using the JAVAControl Panel. For the calibration, the R script shall recognize the sensor type from a list. So that the name of sensors shall be controlled. For the calibration what if the user change a sensor from e. g. O3 to NO2 ! I think a change of sensor (name or model) should trigger a new airsenseur.db database. (this is solved in the future OS 0.5?)

Time delay of measurements: on 16-12-15: Bug Solved: Ho approfittato per installare l'ultimo kernel (4.8.12) al posto del precedente (4.2.16). Ho poi notato che, in mancanza di connessione internet ma in presenza del segnale GPS, l'NTP locale tendeva a sincronizzarsi con il segnale di riferimento al secondo del GPS (PPS). Secondo me questo e' il problema. Il segnale PPS non ha informazione assoluta del tempo ma solo quella di precisione del secondo. Quindi cio' che accade e' che l'orologio locale si sincronizza al centesimo di secondo con l'inizio del secondo fornito dal GPS... ma a un orario assoluto casuale! Ho commentato, quindi, la definizione del segnale di riferimento a 1 secondo all'interno del file /etc/ntp.conf .2017-01-23. The tTime delay of AirSensEUR solved by keeping the GSM on for 10 mins after data push: Looking at the plots in grafana after the push at 10:00, the last values are : - AirSensEUR01 : 10:03:20, - AirSensEUR02 : sampling stop at 01:26, - AirSensEUR03 : 10:03:40, - AirSensEUR04 : 10:02:20, - AirSensEUR05 : 10:03:00, - AirSensEUR\_B06\_01: 10:03:20. The difference between last values 1:20 . Quite good! I checked in ssh the date of AirSensEUR01, the time was correct at 10:09:55. I missed the other AirSensEURs before the end of the connection. Laurent checked AirSensEUR02, exact same time between his PC and AirSensEUR02. If you remember, last time we visually compared the last values in grafana we had more differences in time between AirSensEURs. I guess the synchronisation took place thanks to the longer GSM connections: +1 for VPN and 10 mins GSM. Explanation from Marco: GPS fast , remove local clock, gsm update ntp no adjustment. Mistake of TimeStamp even when GPS and NTP are active ***Explanation needed by Marco***, he has done 2 things among which the GPRS connection lasting for 10 minutes for the Datapush plus s2 cahnge of software for the NTP including forcing of the time without relying on locat time. Check that the sampling is up or restart with a watchdog process, use official oracle JAVA virtual machine. Update the OS in order for the sampling process, to start automatically with on\_sensor add a delay 3 minutes for start\_samplingprocess (to avoid spikes and outliers), start WIFI.

As the sampling process stops is becoming one of the most annoying problems within the AirSensEUR project, I've spent half a day to implement a set of scripts that should be able to minimise it. The script implements a background daemon that periodically monitor the AirSensEURDataAggregator data folder by looking for modifications on the files. If no modifications are performed within three minutes (due to a crash in the airsenseurhost and/or the airsenseurdataaggregator processes) the daemon restarts the two processes then waits for next three minutes to check if modifications happen. If not, it removes the power to the sensor power supply then power up again and restart the sampling processes. This loop is performed each two minutes. The daemon does nothing if the sampling process has not been enabled in the control panel. To implement this, a flag is set/reset when the user pushes the sampling button in the control panel. Being the flag persisted in a file, the good consequence is that the system auto-restarts the sampling when booting, if required. I've already installed a patch in AirSensEUR\_04. If you're confident I would like to install on all the remaining boxes (through VPN). I would test this till the end of the week and, if not surprises will be found, I'll propose to apply the same to NILU and RIVM for further deep testing. Hope this will drastically reduce this annoying problem! MG: - if the power on and/off of the shield does not allow restarting of the data aggregator why not implementing a reboot of the host or sending a mail/message to warn the user? MG "The daemon does nothing if the sampling process has not been enabled in the control panel". MG: I suppose that the daemon will work if the sampling has been enabled with the Web interface. MG: When booting the "push Enabled" button should also auto restart so that both sampling and pushing restart. Sampling Stopping (Marco: implement a watchdog, use the official Oracvle JVM): NILU/Norway: Looking at all past logs (and the one you're sent me too) is possible to see that the common issue is the action "GenCollectForAllocation" performed by the JVM. I'm convincing myself that this is something related to a bug present in the JVM we're using. But it's not a big issue because I'm quite sure we can override on that. Yesterday I've implemented a watchdog daemon that should be able to restart the sampling process if, for a whatever reason, the folder containing the dataset is not updated every 3 minutes. I tested for a while and seems working (I've manually killed several times the sampling process to test it). Today I'll install on all Ispra boxes and, if no surprises will be found and if it's ok for you, I'll propagate the patch to your boxes on Monday.

New Java Panel running with Chemical Shield and PM Shield. As you can see, the "board ID" concept has been added. The Java Panel has been split in "tabs". The first tab is reserved for Chemical Sensor Shield boards, the second is reserved for the new OPC-N2 and MOX sensor. The screenshot is reporting a sampling procedure working on two boards connected on the same SensorBus. Questions: Will the JAVAControl Panel be able to manage more than one sensor shields? (like 2 in case of 4 electrodes sensors we may need 2 shields of more), It is possible that the MOx sensor will be used without having an OPC?, Maybe we could have a plot of the MOx sensor values like for the chemical sensors. Wish list: to read the the configuration of a schield in WIFI, to double chec the upload of configuration file, to add an ID for each sensors + OPC, and ID of the AirSensEUR, to transfer the configuration + IDs to InfluxDB

|  |  |
| --- | --- |
|  |  |

# To be solved or features to be added with futures version

## Hardware

To avoid that the WIFI switched off, set the USB baud rate to full speed.

Adding active sampling system

Add button function outdoor/Indoor (USER4), 1 when indoor and add a column “indoor” in airsenseur.db

Add button function measure/calibration (USER5), 1 when calibration at a monitoring place, and a column “station” in airsenseur.db

Add button function valid/invalid measurements (USER6), 1 when invalid and add a column “invalid” in airsenseur.db

Power supply with solar Panel (B. Wastine Air Normand). Marco is actively testing a first prototype but it is not ready yet. “I personally made some tests with a 40Watts poly and a commercial DC/DC converter. My debug unit is working since a couple of days with a WiFi connection (no GPRS) but very raining days. AirSensEUR is battery (LiFePO4 64Wh cell) operated and has a 5V 3A socket for charging battery. Thanks to the embedded battery manager is possible to have the unit working when charging the battery. The battery manager is able to limit battery charging current prioritising the load... so in theory it would be possible to supply an external 5V with variable current but provided by a DC converter connected to PV panels. This is what I'm working on... but I can't promise anything in a short time as it's done in my (low) spare time.”

Command external devices (such as a relay for example) as Digital or Analogic outputs (B. Wastine Air Normand). “The Host is plenty of spare I/O lines and AD lines. But is not our strategy to use them, mainly for two reasons: 1. It's impossible for us to manage different board revisions 2. All I/O lines on the Host should be managed by Linux drivers. This requires to configure and install the proper drivers and, at the end, to have users skilled to access on. As anticipated, we're pushing to have a single bus where multiple devices can be connected. This will allow us to propose different Shield typologies targeted to different needs. An I/O device would be an interesting unit to be added to the bus. Our idea is to provide a software abstraction on the Host where external processes can interact with all interfaces through simple API calls.”

## Software

Modify the JAVAControlPanel for amperometric 4 electrode sensors, removing sensors channel 2 and 3 when 4-electrode sensor are used on channel 1 and 4.

Adding : page web server for using WIFI instead of GPRS for data push. Adding WIFI pushing in the OS.

Adding: page web server for selecting POSIX time meridian for the first data push? Not sure we can do it automatically with function get\_google\_tz

In the local web site: add function backup and update of the system using the shell script, from a repository?

Adding the calibration algorithm in R or shiny?

The file dataSOSDB.properties should be directly created within the local web server

JAVAPanelConfiguration could be operated through the Local Web server for ease – IS this possible? The jar should be on AirSensEUR.

Add the possibility to read a sensor configuration file of the 7 sensors of a gas sensor shield directly from the JAVAPanelConfiguration. Read the firmware of shield with the JAVAControl Panel

Add the download of coordinates in the SOS protocol (software, Michel).

Add a new table in airsenseur.db that should be pushed giving the configuration of sensor created by the JAVAPanelConfiguration. For example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| namesensor | "CO\_3E300" , , , NA , NA ) | "NO\_3E100" | "NO2\_3E50" , | "O3\_3E1F" |
| ID | 22011587 | 22011587 | 22011587 | 22011587 |
| Sensor.raw.unit | V | V | V | V |
| board.REF | 1.699 | 1.342 | 1.674 | 1.674 |
| board.REF.AD | 0.5 | 0.5 | 0.5 | 0.5 |
| board.zero.set | 1.678 | 1.344 | 1.672 | 1.667 |

The following parameters should be also added maybe in the local WEB server or in the JAVAControl Panel!

Intercept : c(1.678 , 1.344 , 1.672 , 1.667 )

Slope : c(-0.02189/1000 , 0.00001194 , -0.00004302 , -0.00002379 )

Sensor.Unit : c("ppb" , "ppb" , "ppb" , "ppb" )

The control Panel (and likely the whole AirSensEUR host) is sometime blocked in Norway. We think it is related to low outdoor temperature. It was found that the Ram of the Arietta G25 can be used between 0 and 70°C. First tentative solution will be to use a polystyrene insulation. Other possibility: use a heating pad (https://www.sparkfun.com/products/11288) to connect such a micro-heating device to the 5V power > supply from the shield? I used a lab power supply to power up one of the platforms. It looked like it did not draw higher current than 0.67A. The power supply you delivered with the platforms gives 4.3A. It looks like using a bit more current would not be a problem. That would be an interesting test and yes, we have a 4Amps power supply because we want to be able to charge the battery at full speed (it's more than 3Amps when it's completely down. When the battery is charged, the system will not drain more than 250mA so you have a lot of margin. I would take power directly in parallel from the internal DC connector on the Host, instead, where are the wire coming from the Bulgin connector. You can easily solder two wires on the female DC connector. (The wires powering the shield have 7V@200mA maximum). But before going this step further, did you tried to take advantage of the RAM self heating by mounting a small piece of polystyrene in contact with the RAM chip? I'm quite confident that this would be sufficient to have the RAM working in a safe zone. Control Panel Freezing (Marco): NILU: At the same time I'm working to understand the problem related to the control panel freeze. I've found very interesting behaviour in some of the Linux kernel driver we're using to handle the panel... something like "if you have an error when reading on the bus, let's do like if you receive this fixed value". I think this lazy treatment of exceptions is somewhat generates avalanche of events and the following high load on the CPU. I think that we can patch those drivers in order to reduce the effects of the problems, but I'm still investigating on that. I've also found a possible problem in a resistor mounted on the I2C bus. The selected value is at the very limit of the bus specifications. This, added to standard component tolerances and temperature variations, could amplify I2C data transaction exceptions in some AirSensEUR boxes.

# How-to

## Updating the SD to the system OS 0.3

1. Download file at <http://www.liberaintentio.com/output-0-3.tar>
2. Insert the SD found AirSensEUR SD in a linux system (tested with Ubuntu 16.04)
3. You need to unmount SD if an automatic mounting of the SD took place, for example using a graphic file manager
4. Type “dmesg “
5. You need to identify the SD. Something like” /dev/mmc0blk0”

### Make a backup of your data using the script BackupSD.sh. See Appendix 1

1. Go to the directory where you have the script BackupSD.sh, typing “cd” followed by the name of directory
2. type “sudo chmod 775 BackupSD.sh”
3. type ./backupSD.sh and add 2 arguments to ./BackupSD.sh separated with one blank space:
4. the directory where to backup (e.g. /home/oper/Downloads/AirSensEURxx/ASE05/161110\_0.2)
5. and the SD directory found in point 5 (e. g. /dev/mmcblk0).

### Update your system with the new OS version 0.3 using the script MakeNewSD\_03.sh. See Appendix 2

1. with the SD installed in your linux system
2. You need to unmount SD in case of automatic mounting of the SD, for example using a graphic file manager
3. Go to the directory where you have the script MakeNewSD\_03.sh, typing “cd” followed by the name of directory
4. type “sudo chmod 775 MakeNewSD\_03.sh”
5. type ./MakeNewSD\_0.3.sh and add the 3 following arguments to ./MakeNewSD\_0.3.sh separated with one blank space:
6. the directory where is output03.tar, e.g. /home/oper/Downloads/
7. the directory where are the old data of ASE, e. g. /home/oper/Downloads/AirSensEURxx/ ASE01/161114\_0.2/
8. and the SD directory, e. g. /dev/mmcblk0.
9. Stop the terminal and put back the SD in AirSensEUR
10. To update airsenseur.db adding the column “calibrated” filled with 0, switch on AirSensEUR, and switch on the WIFI, start a ssh session as root, type “./updateASDB” in the root directory.
11. Since MakeNewSD\_03.sh deleted the file /usr/local/airsenseur/AirSensEURDataPush/history.db, at the next datapush all data will be pushed again to influx and SOS!?!
12. Update file /usr/local/etc/influxpush.properties to insert new credential and key allowing data push. The file should be:
13. datapath=/usr/local/airsenseur/AirSensEURDataAggregator
14. influxdbhost=influxdb1.liberaintentio.com
15. influxdbport=8086
16. influxdbname=jrcispra
17. influxdbdataset=AirSensEUR01
18. influxdbuser=jrcuser
19. influxdbpasswd=?????????
20. uselineprotocol=true
21. debug=1
22. For JRC Ispra, the only change between each AirSensEUR should be the key influxdbdataset=AirSensEUR01

*Come dicevamo, la nuova tabella contiene una nuova colonna con nome "calibrated". Questa e' attualmente riempita con 0 (vecchi dati) oppure con lo stesso valore float risultato della valutazione dell'espressione locale impostata tramite panello di amministrazione sull'AirSensEUR. L'idea e' che il tool di calibrazione possa sovrascrivere il database sqlite locale e, successivamente, si possa riutilizzare lo stesso DataPush per inviarlo all'influxDB sovrascrivendo cosi' il campo calibrated.*

## Updating the SD to the system OS 0.4

1. Download file <http://www.liberaintentio.com/AirSensEURDataPush.tar> in a linux machine
2. Move the file to for example ...Airsenseurxx/tar/0.4/ for example in a terminal go to the Downlaod directory and type cp AirSensEURDataPush.tar Airsenseurxx/tar/0.4/
3. Go to directory, type cd ...Airsenseurxx/tar/0.4/
4. Extract the compressed archive: tar -xvf ./AirSensEURDataPush.tar
5. connect to AirSensEUR in WIFI (password AirSensEUR)
6. open FileZila (see 6.3 Connect to AirSensEUR in WIFI using FileZila under Windows or Linux with a graphic interface) with AirSensEUR in one pane (/usr/local/airsenseur/AirSensEURDataPush) and the directory of your laptop on the other one (...Airsenseurxx/tar/0.4/AirSensEURDataPush/)
7. substitute the 3 files of the tar file

The new Java software supports https transfer for InfluxDB 1.0 and HTTPS connection has been implemented in the AirSensEURDataPush java program.

1. To enable HTTPS, please add the key: useHTTPS=true into the influxpush.properties file located at /usr/local/etc in ssh
2. Do not delete the AirSensEURDataPush folder but use the .tar to override their contents, otherwise you'll loose the history.db file forcing AirSensEUR to upload again the whole local database to your influxdb server.

## Connect to AirSensEUR in WIFI using FileZila under Windows or Linuz with a graphic interface

1. Install Filezila
2. Open Filezila
3. Menu FILE|SITE MANAGER
4. Click on New Site, select name AirSensEUR,
5. Under GENERALL TAB:
6. Host: 192.168.100.1
7. Protocol: SFTP – SSH file transfer protocol
8. Login: root, Password: password
9. Click on save
10. Click on connect – Next time just connect to AirSensEUR from menu FILE|SITE MANAGER

## Connecting to AirSensEUR in ssh under windows using Putty

1. connect in WIFI to AirSensEUR Password “AirSensEUR”
2. under windows install Putty (<http://www.putty.org/>), connect as root in sftp to 192.168.100.1 password: pasword
3. under linux, open a terminal and type ssh [root@192.168.100.1](mailto:root@192.168.100.1), password: password

## Manual Datapush

1. Connect in WIFI to your AirSensEUR, password AirSensEUR
2. Open a browser at 192.168.100.1
3. Click on “Force Now” in the Web server under the window “General Settings”
4. Connect in SSH (see Putty)
5. To monitor what is going on: in ssh enter: tail –f /var/log/runDataPush.log.

## Cloning a SD in Linux

Since version 0.3, you can use the shell scripts in appendix 1 and 2 on a linux machine using the guide in 6.1 Updating the SD to the system OS 0.3.

Otherwise the main command are:

1. sudo dd if=/dev/sdc of=/home/oper/Downloads/ASE\_OS\_161010.iso to make an iso, check with cd /dev/ then ls to get the name of the SD (sda b c …)
2. sudo dd if=/home/oper/Downloads/ASE\_OS\_161010.iso of=/dev/sdc to burn the iso after changing the SD mother with the SD daughter

## Setting SOS-T dapaPush parameters

1. Configure the SOS server and generate a configuration file for the host (dataPushSOSdb.properties).
2. upload the dataPushSOSdb.properties file, substituting the existing one in /usr/local/etc/
3. uncomment the line in /usr/local/aursenseur/runDataPush which is executing the java for SOS. It should state something like "java –jar AirSensEURDataPush.java ../../etc/datapushsosdb.properties"

### Example of SOS conf file for AirSens02-Ispra

# Defines here the remote SOS server endpoint coordinates

sos.hostname=sossvr1.liberaintentio.com

sos.port=8080

sos.endpoint=52nSOS/sos/json

sos.offering.name=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1/offering/raw/1

# Define here the FOI relevant SOS properties

sos.foi.name=AirSensEUR of c05\_emep\_05

sos.foi.id=http://www.airsenseur.org/featureOfInterest/c05\_emep\_05

# If true, the location will be updated with current GPS info

sos.foi.updatelocation=true

# If true, each observation will be timestamped by a unique Id

sos.observation.byid=true

# Define here, for each sensor, the relevant SOS properties

sos.sensor\_00.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_00.observedprop=http://www.airsenseur.org/ch0/no2\_b43f/202400501/1

sos.sensor\_00.uom=raw

sos.sensor\_01.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_01.observedprop=http://www.airsenseur.org/ch1/co\_a4/132180127/1

sos.sensor\_01.uom=raw

sos.sensor\_02.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_02.observedprop=http://www.airsenseur.org/ch2/o3\_a431/214830322/1

sos.sensor\_02.uom=raw

sos.sensor\_03.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_03.observedprop=http://www.airsenseur.org/ch3/no\_b4/160400162/1

sos.sensor\_03.uom=raw

sos.sensor\_04.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_04.observedprop=http://www.airsenseur.org/ch4/bmp180/p3\_r13/1

sos.sensor\_04.uom=hPa

sos.sensor\_05.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_05.observedprop=http://www.airsenseur.org/ch5/ur100cd/t3\_r13/1

sos.sensor\_05.uom=degrees

sos.sensor\_06.procedure=http://www.airsenseur.org/host\_r12\_ispra\_05/ase1\_r22\_ispra\_05/1

sos.sensor\_06.observedprop=http://www.airsenseur.org/ch6/ur100cd/h3\_r13/1

sos.sensor\_06.uom=RH

sos.filedump=transactions.txt

sos.timeout=90

# Other information for datapush

datapath=/usr/local/airsenseur/AirSensEURDataAggregator

datatimespan=12000000

debug=1

## Adding VPN possibility

### Installing VPN

Here below you can find the instructions to install cryptographic certificates on the AirSensEUR boxes. As soon as they'll be installed and the process has been restarted, each box will instantiate a private virtual network with the sossvr1.liberaintentio.com server. Accessing to this server will allow you and all of us to connect to the remote boxes for administrative and monitoring purposes.

Certificates install is quite simple as the engine has been already setup on the latest microSD card image you deployed some days ago. You don't need to physically access to the microSD card as the install can be done through WiFi.

Below are reported the needed steps for a single box. Please repeat the same for each box you have but replacing the certificates with the proper one for the box you're working on (I've generated a total of 4 certificates, one for each board, thus giving a unique identifier for each board on the server).

1. Download the .tar file with configuration and certificates from this link: http://sossvr1.liberaintentio.com/ovncerts/???\_p00.tar

2. Copy the .tar file to the box and move it (you can use a WiFi connection to the board and operate through SSH and secure copy) to the folder: /etc/openvpn

3. Open an SSH remote connections and move to /etc/openvpn folder: cd /etc/openvpn

4. Uncompress the .tar file: tar -xvf ./???\_p00.tar

5. Remove the .tar file: rm ./???\_p00.tar

6. Restart the openvpn service: /etc/init.d/openvpn restart

This is all you need. As soon as the Internet connection will raise up, the box will connect to the server and generates a private network.

To increase the "time window" where the modem is turned on, I suggest to modify the data push script by artificially add an empty timeframe as soon as the push process is terminated.

To do this, you should modify the /usr/local/airsenseur/runDataPush script by adding a "sleep" before the modem disconnection.

Please open the /usr/local/airsenseur/runDataPush script with a text editor and locate the label "Stop GPRS"

Insert the "sleep" command as shown in the example below:

# Wait 10 minutes to let remote VPN accesses

sleep 600

# Stop GPRS

sleep 30

killall wvdial

sleep 20

In this situation the modem connection will be persisted for 600 seconds after the data push process. It would be also useful to increase the data push process scheduling (through the web administrative panel) to have the box reachable more frequently.

### Connecting in VPN with ssh

Since this morning all AirSensEUR boxes are linked with a VPN to the sossvr1.liberaintentio.com server. The VPN is active for the data push time and 10 minutes after.

Here are the details for accessing to our server

URL: sossvr1.liberaintentio.com

Username: sosadmin

Password: \*\*\*\*\*\*\*

As soon as you connected through SSH to the sossvr1 server, you should use it to start another SSH connection to the IP. Ispra boxes have been allocated to server's local IP range 10.20.100.40, (Alex's box) and 10.20.100.41 to 10.20.100.45 for Ispra01 to Ispra05 boxes.

### Connect to the AirSensEUR Web interface in ssh with tunnel

For what is related to the possibility to have the web interface... you already have it! It's all in the magic of the SSH protocol. SSH allows you to "map" remote ports/IP addresses into local port.

As soon as you're connected to the sossvr1 through SSH (for example using putty) you can assign a port (or a range of ports) to your local IP (or a range of ports) thus reflecting ports for specific IPs that are local to the server.

Example:

The server "see" the AirSensEUR box at the 10.20.100.41 IP (it's on his private network through the VPN) so the server could access to port 22 and

port 80 of the box.

You want to access to the port 80 for the 10.20.100.41 server's local IP and you have a working SSH connection to the server.

Solution:

You should ask to SSH to "reflect" the port 10.20.100.41:80 from the server to your local machine's port (you should select a local port that's greater than 1000 as the first 1000th are reserver by the kernel in your machine). Let's assume, for example, you decided for your local 8010 (localhost:8010).

How to do it.

1. Open putty

2. Log in to the sossvr1.liberaintentio.com as usual

3. Click on the icon in the left top of the putty window

4. Go to the "Change Settings"

5. Expand the SSH node under "Connection" in the menu tree on the left

side in the new window

6. Select "Tunnels"

7. In Source Port set 8010

8. In Destination set: 10.20.100.41:80

9. Click on "Add" button

10. Click on "Apply"

11. Open your browser and point to http://localhost:8010

You should receive the AirSensEUR Web Console Panel for Ispra01 box (the

one mapped to 10.20.100.41).

You can "save" your current session settings as a "preset" using the

"Session" tab in the Putty configuration panel. In this way you "tunnels"

will be maintained through sessions.

Have a nice day.

Best regards,

Marco.

### Restarting sampling

Per verificare se il sampling sta funzionando, guardo il timestamp del file airsenseur.db in /usr/local/airsenseur/AirSensEURDataAggregator

Una alternativa, per vedere "live" i dati, e' quella di usare il commando tail -f sull'ultimo file .csv disponibile:

tail -f /usr/local/airsenseur/AirSensEURDataAggregator/DataSet\_000XX.csv

Cosi' facendo vedi "scorrere" i valori mentre arrivano dal DataAggregator.

Per fare ripartire i processi di sampling (sono due: l'Host e il DataAggregator) tramite SSH e' possibile lanciare i due comandi qui sotto:

/usr/local/airsenseur/stop\_samplingprocess

/usr/local/airsenseur/start\_samplingprocess

(Conviene sempre prima fare uno stop e poi uno start perche' cosi' si e' sicuri che entrambi i processi sono stati fermati). Per una verifica del funzionamento puoi andare a vedere se in /usr/local/airsenseur/AirSensEURDataAggregator viene creato un nuovo DataSet\_000XX.csv e viene popolato con i nuovi dati.

NOTA: Utilizzando SSH si riesce a far ripartire i due processi di sampling senza togliere l'alimentazione alla scheda sensori. Questo evita gli "spike" iniziali generati da alcuni sensori quando gli togli tensione. Nel caso in cui non fosse sufficiente (perche' lo shield ha necessita' di un RESET) allora e' possibile togliere e ridare l'alimentazione allo shield tramite i comandi:

/usr/local/airsenseur/off\_sensor

/usr/local/airsenseur/on\_sensor

# Troubleshooting

## No values in Live View of the web server: values 32768, 32768, 32768, 32768, 0, -40 and 0)

Pen the AirSensEUR box and check the voltage of the power supply of the shield.

* If the voltage is 0 V, the cable is not well fix or the polarity is wrong. The electric wire is not well connected please connect it correctly, there is a “+” sign on the host to get the good polarity and the negative wire is to be connected to the screw port that is the nearest to the corner of the shield.

|  |  |
| --- | --- |
| +  Host | +  Shield |

* If it is 2.7 or 2.8 V. The data cable between the shield and the host is incorrect. It is important to observe the version of the shield. Currently, version R2.2 uses a cable called: “Dritti” in the photo below while the shield version R2.3 uses the one noted “90ºC”.



## WIFI does not work

When pressing the WIFI button, if the led light a few times (5) and then switched off, it is likely that the MAC Address of the WIFI USB is incorrect in the file /etc/udev/rules.d/70-persistent-net.rules . Just delete this file and reboot AirSensEUR. The 70-persistent-net.rules will be updated at next startup.

You can connect with Fileila (see 6.3 Connect to AirSensEUR in WIFI using FileZila under Windows or Linuz with a graphic interface) or in ssh to edit this file (see 6.4 Connecting to AirSensEUR in ssh under windows using Putty).

The name of the SSID can be edited in /etc/hostapd/hostapd.conf.

## Data Push or DataAggregator does not work

### Problem with date

It could be because the system date is incorrect, in particular when AirSensEUR is kept indoor because the date of the system is not updated by the GSM or GPS. This problem was solved in version 0.2 of the system. Nevertheles, if this problem occurs, you may need to update manually the date and time with the following command in a linux terminal:

1. Switch on WIFI, see figure below.
2. Switch off data acquisition (DAQ), see figure below.

|  |
| --- |
| **WIFI**  **DAQ**  Control Panel |

1. In a linux terminal type [ssh root@192.168.100.1](mailto:ssh%20root@192.168.100.1) , password: password (under windows, it is possible to use putty (see Connecting to AirSensEUR in ssh under windows using Putty)
2. Type “cd /usr/local/airsenseur/airsenseurdataAggregator”
3. Type “ls –ll”.
4. If the date of airsenseur.db should be updated to current date. Type “date MMDDhhmmYYYY”
5. Restart data acquisition
6. After a few minutes check the date of the of the airsenseur.db and \*.csv files
7. Type “cd /usr/local/airsenseur/airsenseurdataAggregator”
8. Type “ls –ll”. The date of airsenseur.db should be updated to current date.

### The GPRS dongle does not work (16-12-15)

If the GPRS does not connect to GSM signal even outside, you may need to insert the dongle until it starts again. Sono sempre piu' convinto, quindi, che ci sia qualche occasione in cui la SIM e/o la coppia SIM+network si portino in uno stato in cui e' necessaria la comunicazione GSM (e non GPRS). Questa necessita piu' di 1A disponibile sulla porta USB dell'AirSensEUR e quindi il DC/DC converter va in protezione. Il tutto viene risolto alimentando almeno una volta il dongle con una presa USB in grado di erogare piu' di 1A.

Forse converra', basandosi sull'esperienza che si maturerà per il DC/DC converter dell'OPC, fare in modo che la porta USB del GSM, nei prossimi Host, sia in grado di erogare piu' di 1A...

## AirSensEUR does not push data on its own while manual pushing still works

This error was solved in the current version of MakeNewSD\_0.3.sh

* Scheduling of cron is carried out using a file that is created with the web interface: /etc/cron.d/datapushcron.
* This file is normally saved and copied in the backup script as the tar file extracted in the microSD does not include it.
* If the file does not exist, local web interface shows the data in cron.defaults of /usr/local/etc but never carries out a cron job because the /etc/cron.d/datapushcron does not exist.
* Anyhow saving the scheduling in the Web server would create a /etc/cron.d/datapushcron file and hence solve the problem.

## If SD is damaged

* To be able to resume you system you should always make a backup of the OS using BackupSD. Sh as explained in 6.1.1 Make a backup of your data using the script BackupSD.sh. See Appendix 1.
* Resume your system with the script MakeNewSD\_??.sh, see in 6.1.2 Update your system with the new OS version 0.3 using the script MakeNewSD\_03.sh. See Appendix 2

Solution of Marco

*Having Windows showing only the 1st 128MB partition is normal. This is because Windows is not able to read ext4 partitions. You should try to re-partition the whole microSD through a disk management console (you normally find on somewhere in the control panel, under maintenance services). You probably need to delete the partitioned space, then partition again.*

*Instead, it would be interesting to test this tool, if you have a spare time:*

*https://sourceforge.net/projects/win32diskimager/*

*It allows you to completely rewrite an SDCard/USBStick/microSD sourcing from an image file. There is a not updated and standard (i.e. with no WiFi Client) image file at this link (we used Win32diskimager to generate it):*

*http://www.liberaintentio.com/output-0-3.zip*

*You can unzip and use as the source file for the Win32DiskImager. At the end of the process you must be able to take the microSD card and properly boot an AirSensEUR Box.*

*Some days ago I had a conversation with a colleague that was working on a microSD for the Raspberry and was asking a consultancy because the system stopped working (and booting) after some days. We found a very strange behaviour for the microSD. The system was able to format and re-partition but... the microSD contents was never changed. All worked like the processes were done but nothing changed on the microSD. We supposed the microSD embedded controller was "bugged" and/or corrupted. It was like the microSD was no more able to erase their content.*

*I suspect that microSD price drop is raising some concerns about their reliability... seems like what happened on 1.4Mb 3.5" floppies. At the end of their life they were very cheap.. but several time happened to loose datas when writing on them.*

## WIFI Led switch off

1. Put USB in full speed not high speed for stability (ask to Marco, how to do it)
2. Ask Marco for Battery manager to flash – not to switch-off
3. Resistance 4.7 to 1 Mohms stability for battery charging

## Sampling stopping on it own – increase memory

Hi Tore,

thank you for the logs.

So you had at least two JVM crashes in the AirSensEURHost application.

Looking at the log I see something interesting related to the memory. I don't know if this is the cause, but I can see the "Perm Gen" memory is used for more than 90%.

compacting perm gen total 7168K, used 6956K [0xb0200000, 0xb0900000,

0xb4200000)

the space 7168K, 97% used [0xb0200000, 0xb08cb160, 0xb08cb200,

0xb0900000)

It would be interesting to try to assign more memory to the JVM running the application.

You can start doing by yourself, if you want test it, by simply changing the startup line on the:

/urs/local/airsenseur/AirSensEURHost/airsenseurhost file

Just add the -XX:MaxPermSize=32M as a third token in order to have the following:

nohup java -server -XX:MaxPermSize=32M -Djna.nosys=true -jar $JARPACKAGE

$CONFIGFILE </dev/null > $LOGFILE 2>&1 &

And restart the service.

I'll do the same with our debug systems.

Thank you for the feedback.

Have a nice day.

Marco Signorini.

# New SD – configuration in WEB Server

See section 6 in Gerboles M., et al, AirSensEUR: and open data/software/hardware multi-sensor platform for air quality monitoring: Part B: Host, influx datapush and assembling of AirSensEUR, 2016, http://dx.doi.org/10.2790/214743

With the target SD mounted in Linux or using WIFI+Filezilla check that the push are correctly set both for Influx/grafana and SOS in /usr/local/airsenseur/runDataPush, under #Push Updates, comment/uncomment if needed.

# Appendix 1 : BackupSD.sh

#!/bin/sh

# User config of directory and SD dev -----------------

echo "Before runnung the script, be sure to type sudo su in the terminal prompt"

echo "Before running the script, type chmod 775 MakeNewSD\_03.sh to make the script executable "

echo "You need to unmount SD in case of automatic mounting of the SD"

echo ""

echo "you have to add 2 arguments to ./BackupSD.sh:"

echo "1: the directory where to backup, ex /home/oper/Downloads/AirSensEURxx/ASE05/161110\_0.2"

echo "2: the SD directory, ex /dev/mmcblk0. Do you want to continue ? (yes/no)"

read WaitEnter

if [ ! ${WaitEnter} = "yes" ]; then

exit # is kill necessary?

fi

# Define the folder where you want to backup the microSD data

# AirSensEUR=/home/oper/Downloads/AirSensEURxx/ASE05/161110\_0.3

AirSensEUR=${1}

echo "Directory where to backup the SD is ${AirSensEUR} - Make sure that it is empty"

if [ -d ${AirSensEUR} ]; then

echo "Directory ${AirSensEUR} already exists, press any key"

else

mkdir ${AirSensEUR}

echo "Directory ${AirSensEUR} did not exist and it was just created, press any key"

fi

read WaitEnter

# -- Customize here with the SD device target name

echo "Defining the variables to mount the SD, check with dmesg before"

#TGTDEV=/dev/mmcblk0

TGTDEV=${2}

if [ ! -f ${TGTDEV} ]; then

echo "SD found at ${TGTDEV}"

else

echo "Directory ${AirSensEUR} did not exist and it was just created"

fi

TGTDEVP1=${2}p1

TGTDEVP2=${2}p2

# -- End of user customization -----------------

# go to AirSensEUR

Init\_Dir=$(pwd)

echo "Initial directory is ${Init\_Dir}"

echo "Moving to directory: ${AirSensEUR}"

cd ${AirSensEUR}

# Mount

echo "Mounting the SD that is going to be backup and creating dirs for backup (boot, rootfs)"

mkdir boot

mkdir rootfs

mount ${TGTDEVP1} boot

mount ${TGTDEVP2} rootfs

mkdir BOOT

mkdir ROOTFS

# Backup of files

echo "Copying files in rootfs"

# if gcp is not installed use cp -a, gcp options: recursive, overwrite all and keep all atributes

# gcp -rf --preserve=mode,ownership,timestamps ../rootfs/\*

rsync -avc rootfs/\* ROOTFS/.

echo "Copying files in boot"

# if gcp is not installed use cp -a, gcp options: recursive, overwrite all and keep all atributes

# gcp -rf --preserve=mode,ownership,timestamps ../boot/\*

rsync -avc boot/\* BOOT/.

# Unmount

echo "UnMounting the SD"

umount boot

umount rootfs

rmdir boot

rmdir rootfs

#go back to the initial directory

echo "Go back to the initial directory is ${Init\_Dir}"

cd ${Init\_Dir}

# Appendix 2 : MakeNewSD\_0.3sh

#!/bin/sh

clear

# User config of directory and SD dev -----------------

echo "Before runnung the script be sure to type sudo su in the terminal prompt"

echo "Before type chmod 775 MakeNewSD\_03.sh to make the script executable "

echo "You need to unmount SD in case of automatic mounting of the SD"

echo ""

echo "Add 3 arguments to ./MakeNewSD\_0.3.sh:"

echo "1 the directory where is output03.tar (ex /home/oper/Downloads/),"

echo "2 the dir where are the old data of ASE (ex /home/oper/Downloads/AirSensEURxx/ASE01/161114\_0.2/)"

echo "3 the SD directory (eg /dev/mmcblk0). Do you want to continue ? (yes/no)"

read WaitEnter

if [ ! ${WaitEnter} = "yes" ]; then

exit # is kill necessary?

fi

# Define the folder where is output\_0.3.tar

#diroutputtar=/home/oper/Downloads/

diroutputtar=${1}

if [ -f ${diroutputtar}/output-0-3.tar ]; then

echo "The file output-0-3.tar is found in directory ${diroutputtar}, press any key"

else

echo "The file output-0-3.tar is not found in directory ${diroutputtar}, stopping the execution of the script"

exit

fi

read WaitEnter

#Old\_ASE=/home/oper/Downloads/AirSensEURxx/ASE05/161110\_0.2/

Old\_ASE=${2}

echo "Previous system of AirSensEUR shall be in ${Old\_ASE}, press anykey"

if [ -d ${Old\_ASE} ]; then

echo "Directory ${Old\_ASE} exists, press any key"

else

echo "Directory ${Old\_ASE} does not exist, stopping the execution of the script"

exit

fi

read WaitEnter

# -- Customize here with the SD device target name

echo "Defining the variables to mount the SD, check with dmesg before, press anykey"

read WaitEnter

#TGTDEV=/dev/mmcblk0

TGTDEV=${3}

#if [ -f ${TGTDEV} ]; then

# echo "SD found at ${TGTDEV}"

#else

# echo "SD not found at ${TGTDEV}"

# exit # should it be a kill?

#fi

TGTDEVP1=${TGTDEV}p1

TGTDEVP2=${TGTDEV}p2

echo "The SD is in ${TGTDEV}, press anykey"

read WaitEnter

# -- End of user customization -----------------

#Create the partitions

echo "Creating the partion on the SD, press anykey"

read WaitEnter

sed -e 's/\t\([\+0-9a-zA-Z]\*\)[ \t].\*/\1/' << EOF | fdisk ${TGTDEV}

o # clear the in memory partition table

n # new partition

p # primary partition

1 # partition number 1

# default - start at beginning of disk

+128M # 128 MB boot partition

t # change to FAT16

6 # FAT16

n # new partition

p # primary partition

2 # partion number 2

# default, start immediately after preceding partition

# default, extend partition to end of disk

p # print the in-memory partition table

w # write the partition table

q # and we're done

EOF

#Format SD

echo "Formating the 2 partion of the SD, press anykey"

read WaitEnter

mkfs.vfat -n boot ${TGTDEVP1}

mkfs.ext4 -L rootfs ${TGTDEVP2}

# Mount SD

# go where is the file output.tar

Init\_Dir=$(pwd)

echo "Initial directory is ${Init\_Dir}"

cd ${diroutputtar}

echo "Mounting the SD in ${diroutputtar}, press anykey"

read WaitEnter

mkdir sdboot

mkdir sdrootfs

mount ${TGTDEVP1} sdboot

mount ${TGTDEVP2} sdrootfs

# Extract to SD

echo "Extract output-0-3.tar to SD, press anykey"

read WaitEnter

tar -xvf ./output-0-3.tar --strip-components=3 -C ./sdboot/ ./output/fat

tar -xvf ./output-0-3.tar --strip-components=3 -C ./sdrootfs/ ./output/rootfs

# add the boot of Mother directory in case the boot of output\_0.3.rar is incorrect

#cd Mother\_161017/

#cp -a \* ${diroutputtar}/sdboot/.

# Copy the var log from previous version

echo "Copying the old logs to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/var/log

cp -a \*.log ${diroutputtar}/sdrootfs/var/.

# Copy the cron scheduling from previous version

echo "Copying the cron scheduling from previous version to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/etc/cron.d

cp -a datapushcron ${diroutputtar}/sdrootfs/etc/cron.d/.

#Update database, \*.csv files and history.db to avoid to push twice

echo "Copying the old data (db and csv) to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/usr/local/airsenseur/AirSensEURDataAggregator/

cp -a airsenseur.db ${diroutputtar}/sdrootfs/usr/local/airsenseur/AirSensEURDataAggregator/.

cp -a \*.csv ${diroutputtar}/sdrootfs/usr/local/airsenseur/AirSensEURDataAggregator/.

cd ${Old\_ASE}/ROOTFS/usr/local/airsenseur/AirSensEURDataPush/

cp -a history.db ${Old\_ASE}/ROOTFS/usr/local/airsenseur/AirSensEURDataPush/

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

# in the new Influx1.0 (OS vs0.3) we need to push all data, then I delete the file history.db. For other OS versiosn we should not delete it

cd ${Old\_ASE}/ROOTFS/usr/local/airsenseur/AirSensEURDataPush/

rm history.db

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

# cp -a history.db ${diroutputtar}/sdrootfs/usr/local/airsenseur/AirSensEURDataPush/.

echo "Copying the old config files to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/usr/local/etc/

cp -a datapushsosdb.properties ${diroutputtar}/sdrootfs/usr/local/etc/.

cp -a influxpush.properties ${diroutputtar}/sdrootfs/usr/local/etc/.

cp -a sensor.properties ${diroutputtar}/sdrootfs/usr/local/etc/.

cp -a wvdial.conf ${diroutputtar}/sdrootfs/usr/local/etc/.

cp -a datapushcron ${diroutputtar}/sdrootfs/usr/local/etc/.

cp -a aggregator.properties ${diroutputtar}/sdrootfs/usr/local/etc/.

# name SSID

echo "Copying the old SSID files to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/etc/hostapd

cp hostapd.conf ${diroutputtar}/sdrootfs/etc/hostapd/.

# Wvdial.conf fpor the name of the GPRS provider

echo "Copying the config file of GPRS to SD, press anykey"

read WaitEnter

cd ${Old\_ASE}/ROOTFS/etc/

cp wvdial.conf ${diroutputtar}/sdrootfs/etc/.

# updating the airsenseur.db

#echo "updating airsenseur.db, adding field calibrated, press anykey"

#read WaitEnter

#cd /

#./updateASDB

# Unmount and remove dir

echo "unmounting SD, press anykey "

read WaitEnter

cd ${diroutputtar}

umount sdboot

umount sdrootfs

rmdir sdboot

rmdir sdrootfs

#go back to the initial directory

echo "Go back to the initial directory is ${Init\_Dir}, press anykey"

read WaitEnter

cd ${Init\_Dir}