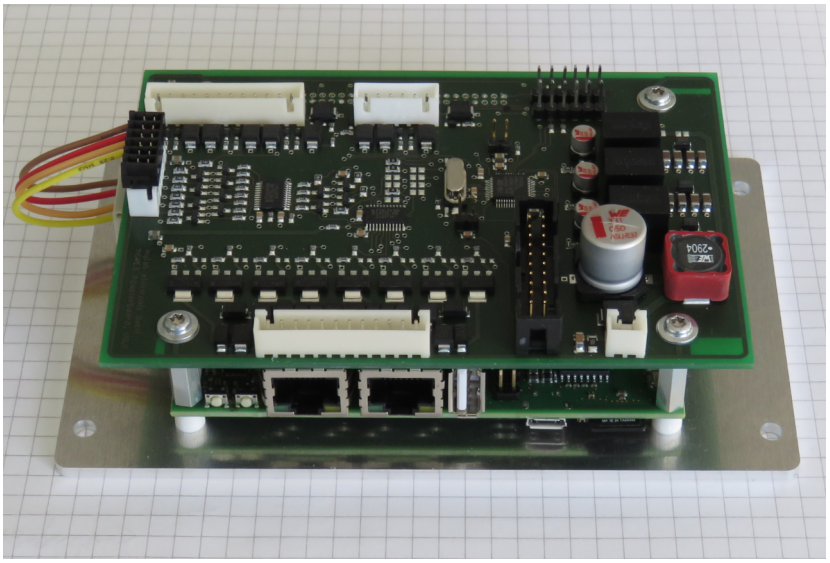

IO Controller Software

Version 2.1

Bubikon, 13. April 2021

mylab
elektronik GmbH

Inhalt

1 IO Controller Software - Bestandteile

Die IO Controller Software besteht im Wesentlichen aus folgenden Teilen:

- Bootloader und Linux Betriebssystem
  - Basiert auf Phytec phytec-headless-image-phyboard-segin-imx6ul-2, Version PD19.1.0 (RAUC ready)
  - Folgende Komponenten wurden hinzugefügt:
    - curl
      openembedded-core
      https://layers.openembedded.org/layerindex/recipe/5765/
      Tool für client-side URL transfers. z.B. für Webserver-Tests
    - tcpdump
      meta-networking
      https://layers.openembedded.org/layerindex/recipe/23477/
      Tool für Netzwerk Diagnose (u.a.)
    - netcat-openbsd (openBSD)
      meta-networking
      https://layers.openembedded.org/layerindex/recipe/5210/
      Tool für Netzwerk Diagnose (u.a.)
    - lsof
      openembedded-core
      https://layers.openembedded.org/layerindex/recipe/136/
      Anzeigen offener Files (inkl. TCP/UDP ports, domain sockets,..)
    - chrony / chronyc
      meta-networking
      https://layers.openembedded.org/layerindex/recipe/51402/
      Zeitmanagement über Netzwerk (NTP)
    - cronie
      openembedded-core
      https://layers.openembedded.org/layerindex/recipe/5707/
      Periodisches Ausführen von Tasks genannt «cron jobs»
    - Fcgi
      meta-webserver
      https://layers.openembedded.org/layerindex/recipe/28567/
      Bibliothek für Kommunikation zwischen Applikationen und Webserver
    - spawn-fcgi
      meta-webserver
      https://layers.openembedded.org/layerindex/recipe/33352/

Tool zum starten von fcgi Applikationen. (Wird nur für Tests verwendet.)

- syslog-ng
  meta-oe
  https://layers.openembedded.org/layerindex/recipe/1030/
  Log system mit Möglichkeit von Remote-Logging
- logrotate
  openembedded-core
  https://layers.openembedded.org/layerindex/recipe/97/
  Tool um Logfiles zu rotieren (Anzahl/Grösse limitieren)
- esmtp
  meta-networking
  https://layers.openembedded.org/layerindex/recipe/31914/
  send-only E-Mail agent für das Verschicken von Status-E-Mails.
- monkey
  meta-webserver
  https://layers.openembedded.org/layerindex/recipe/27933/
  Webserver mit FCGI Schnittstelle.

- ioControl
  - Kommandozeilen Tool um Ein-/Ausgänge anzusteuern.
  - C++ source code erstellt von mylab.
- ioControlFcgi
  - Tool um Ein-/Ausgänge via FCGI anzusteuern.
  - Wird von systemd als Service gestartet.
  - C++ source code erstellt von mylab.
- ioPeriodicPinCheck
  - Skripte (bash) und Konfigurationsdateien für die periodische Überwachung bestimmter Ein- und Ausgänge.
  - Wird via cron job alle 2 Minuten angestossen.
    Nach einem Reboot wird die Konfiguration der I2C-Schnittstelle und des Ports ebenfalls mittels cron job durchgeführt.
  - Verschickt via esmtp E-Mails.
  - Benutzt ioControl (siehe oben).
- Webinterface
  - Webseite basierend auf Bootstrap 4
  - Wird von monkey Webserver auf Port 80 (HTTP) zur Verfügung gestellt.

    - Es können optional einige stationsspezifischen Angaben (Links, Text) eingebettet werden.
- Installationsskript und Standard-Konfigurationssatz
    - Skript um die Installation und Konfiguration effizient durchzuführen.

2 Installation

2.1 NAND Flash via SD Card updaten

Dieser Schritt wird grundsätzlich von mylab ausgeführt um das Betriebssystem der vorhandenen Units auf den aktuellen Stand zu bringen. (Optional kann am Ende dieses Schritts auch die Installation der anderen Komponenten durchgeführt werden.)

Dieser Schritt erfordert direkten Zugang zum Unit um die micro SD Card einzusetzen und über die serielle Schnittstelle den Bootloader zu bedienen.

Der Vorgang ist im folgendem Dokument detailliert beschrieben:
Agnes_ioControlUnit_NandUpdateUsingSDCard.md
(als PDF: Agnes_ioControlUnit_NandUpdateUsingSDCard.pdf)

2.2 Installation der IO Controller Software

Alle Komponenten der IO Controller Software, die nicht zum Betriebssystem gehören, werden in einem Archiv zur Verfügung gestellt:

ioControlDelivery_<DATE>.tar.gz

Dieses kann via SCP auf das Unit geladen werden. Z.B. aus Linux:

scp ioControlDelivery_<DATE>.tar.gz root@<IP>:/home/root/ioControlDelivery.tar.gz

Oder entsprechend mit WinSCP, FileZilla,..

Hinweis: mylab kopiert das aktuelle Archiv im Anschluss des NAND Update (siehe oben) ins Verzeichnis /home/root/.

Danach muss via SSH auf das Unit als root eingeloggt werden. Folgende Befehlskette entpackt das Archive und startet die interaktive Installation:

```
cd ~ && tar -xzpf ioControlDelivery.tar.gz &&
 cd ./stage && chmod +x install.sh && ./install.sh
```

3 Anpassungen / Konfiguration auf swisstopo Umgebung

Diverse Punkte müssen für die Umgebung im künftigem Netzwerk von swisstopo angepasst werden. Diese Anpassungen wurde von mylab soweit möglich übernommen und im Archiv bzw. im Installationsskript (install.sh) integriert.

Manche Punkte müssen für jede Station individuell konfiguriert werden. Dies kann entweder interaktiv während der Installation erfolgen oder auch nachträglich angepasst werden.

Hinweis: Viele Details zur Konfiguration sind im Installationsskript (install.sh) ersichtlich. Um dieses einzusehen kann das Archiv ioControlDelivery_<DATE>.tar.gz entpackt werden. Damit sind auch alle Konfigurationsdateien und Skripte leicht einsehbar.

3.1 Benutzer

Der Super-User Account «root» ist in jedem Fall vorhanden. Standardpasswort (bei mylab): keines (nur im Ausnahmefall «root»).

Dieses muss angepasst werden. Falls keines gesetzt ist, kann dies während der Installation gesetzt werden.

Die Installation fragt, ob ein weiterer Benutzer erzeugt werden soll.

3.1.1 Verwaltung der Logins (swisstopo interner Prozess)

swisstopo erzeugt für jede Station eigene Passwörter, die den entsprechenden Anforderungen genügen, und verwaltet diese mittels Keepass (oder einem ähnlichem Tool). Alle Passwörter werden jährlich erneuert.
Als root Passwort darf auf keinen Fall das gleiche Passwort wie für das Webinterface (siehe unten) verwendet werden!

Die IO Control Units implementieren keine Massnahmen um dies sicherzustellen.

Folgender Befehl kann verwendet werden, um die Passwörter zu ändern (nach Login als root):

passwd <USER>

(Ohne Angabe <USER> wird das Passwort des aktuell eingeloggten Benutzer verändert.)

3.2 Hostname / Stations-ID

Die Stations-ID («Four Character ID»; z.B. DAC2, KALT,…) wird mit dem fixen Prefix «control» als Hostname verwendet. Die Installation verwendet sowohl den Hostname als auch die Stations-ID als Standartwerte für weitere Einstellungen. Grundsätzlich kann aber auch ein beliebiger anderer Hostname angegeben werden.

3.3 Netzwerk

Die Konfiguration für die beiden Netzwerkinterfaces eth0 und eth1 sind in den Dateien config/10-eth0.network und config/10-eth1.network im Archiv abgelegt und werden während der Installation ins entsprechende Systemverzeichnis kopiert.

Das Interfaces eth0 (EN1) wird mit der fixen IP 192.168.3.11 und DHCP konfiguriert. Es wird momentan nur für die Installation und zur allfälligen Fehlersuche vor Ort verwendet.

Das Interface eth1 (EN2) wird während der Installation für den Anschluss an das Netzwerk (BIT) konfiguriert. (Vor der Installation ist es mit der fixen IP 192.168.4.11 und DHCP konfiguriert.) Falls nach der Installation Änderungen nötig sind, können diese direkt z.B. mit folgenden Befehlen durchgeführt werden:

vi /etc/systemd/network/10-eth1.network
systemctl restart systemd-networkd

(Siehe auch
https://www.freedesktop.org/software/systemd/man/systemd.network.html)

3.4 NTP

Es wird folgender NTP Server verwendet:

ntp.metas.admin.ch

Falls dies angepasst werden muss, können mit folgenden Befehlen die nötigen Konfigurationsdateien geändert und geladen werden:

vi /etc/chrony.conf
vi etc/chrony.key
systemctl restart chronyd

(Die Dateien können auch per SCP auf die Stationen kopiert und via SSH geladen werden.)

3.5 Remote logging

Aktuell werden nur lokale Log geschrieben (/var/log/*.log). Falls erwünscht und entsprechende Infrastruktur (Log-Server) vorhanden ist, kann syslog-ng entsprechen konfiguriert und nachgeladen werden:

vi /etc/syslog-ng/syslog-ng.conf
systemctl daemon-reload
systemctl restart syslog-ng

(Die Dateien können auch per SCP auf die Stationen kopiert und via SSH nachgeladen werden.)

3.6 E-Mail-Versand

Aktuell wird folgender SMTP Server verwendet:

identity = <HOSTNAME>@swisstopo.ch
    mailhost.admin.ch:25

Hinweis: indentity= wird während der Installation entsprechend gesetzt. Dabei kann auch eine andere E-Mail-Adresse eingetragen werden.

Falls dies geändert werden muss, kann der gewünschte Mailserver (SMTP) mit folgendem Befehl (als root) konfiguriert werden:

vi ~/.esmtprc

Testen:
echo "Mein Testmail vom dd.mm.yyy HH:MM" > mail.txt
sendmail -v MYNAME@MYMAIL.MY < mail.txt
rm mail.txt

(Die Datei .esmtp kann auch per SCP auf die Stationen kopiert werden.)

3.7 Periodische Pin-Überprüfung

Aktuell werden die folgenden Pins periodisch (alle 2 Minuten) überprüft:

pin_list='IN1 IN2 CHARGER_FAIL POWER_FAIL BAT_FAIL OUT1'

Dies kann direkt im Skript testpinchange.sh angepasst werden:

vi /home/root/ioPeriodicPinCheck/testpinchange.sh

(Die Datei kann auch per SCP auf die Stationen kopiert werden. Achtung: Die Berechtigung zum Ausführen muss mit kopiert oder per SSH gesetzt werden.)

3.7.1.1 E-Mail-Adressen (für Pin-Überprüfung)

In der Datei /home/root/ioPeriodicPinCheck/emailreport.conf sind «from», «to», «cc» und «bcc» wie folgt definiert:

from=<HOSTNAME>@swisstopo.ch
to=AgnesOp@swisstopo.ch
cc=
bcc=

Hinweis: from= wird während der Installation entsprechend gesetzt. Dabei kann auch eine andere E-Mail-Adresse eingetragen werden.

Falls dies geändert werden muss, kann dies mit folgendem Befehl (als root) gemacht werden:

cd /home/root/ioPeriodicPinCheck/
vi emailreport.conf

Testen:
./emailreport.sh EinTest body.txt

3.8 Web Interface

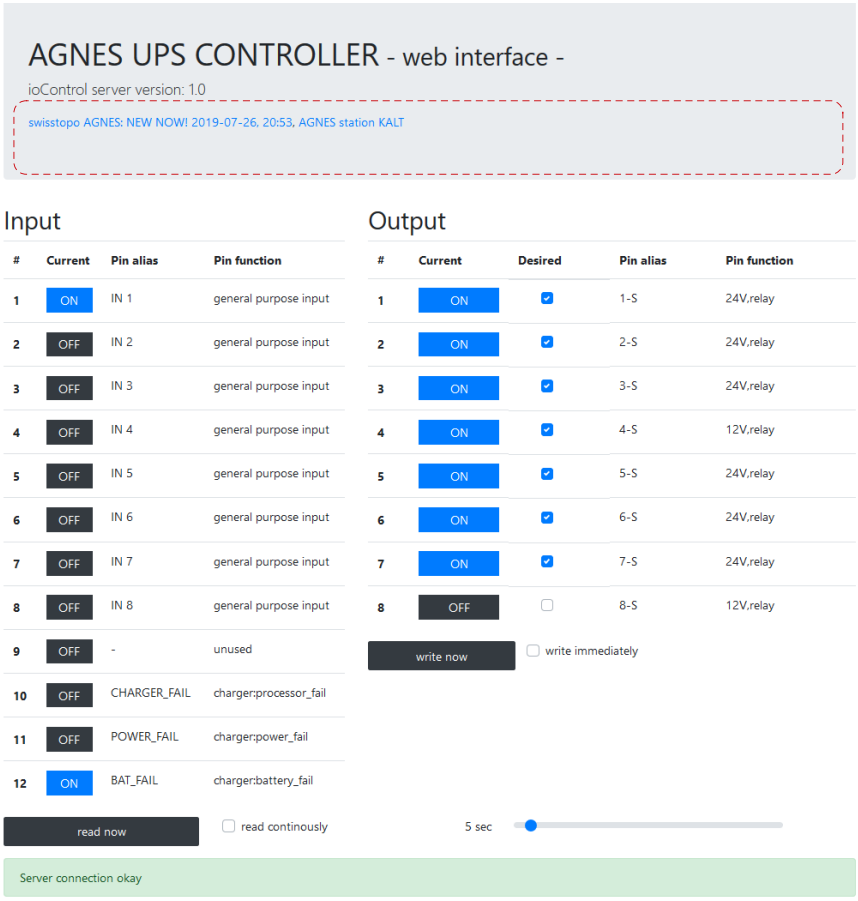

Abb. 1: Zusätzliches HTML im Jumbotron (agnes_links.html) rot gestrichelt.

Im Titelbalken der Webseite («Jumbotron») wird zusätzliches HTML aus der Datei /var/www/monkey/ioControlWeb/agnes_links.html eingefügt. Die Idee ist, dass so einfach stations-spezifische Angaben oder Links eingefügt werden können.

Während der Installation wird diese Datei mit der Stations-ID versehen. Zudem kann während der Installation die Datei direkt mit vi editiert werden.

Man kann sie auch jederzeit z.B. via SCP neu hochladen oder direkt mit folgendem Befehl editieren:

vi /var/www/monkey/ioControlWeb/agnes_links.html

Ein SHIFT+Reload im Webbrowser reicht, um die Änderung zu sehen.

3.8.1 Start-up Ausgangskonfiguration

Falls erwünscht, kann mit dem folgenden Script ein cronjob erzeugt werden, der beim Aufstarten die Ausgänge auf einen definierten Zustand setzt:

add_startup_pinconfiguration.sh

Das Script wird separat bereitgestellt. Der gewünschte Zustand ist im Script hard-codiert.

Ohne diesen cronjob, bleiben die Ausgänge bei einem Reboot (ohne Stromunterbruch) unverändert; bei einem Stromunterbruch sind die Ausgänge grundsätzlich auf «OFF», sie können aber mittels Bestückungsoption verändert werden. (Widerstände: R301|R305; R302|R306; R307|R311; R308|R312; R313|R317; R314|R318; R319|R323; R320|R324)

3.8.2 Authentifizierung

Es wird aktuell «basic access authentication» für HTTP verwendet.

Während der Installation kann ein Login erstellt werden. Als Standard-Vorgabe wird der Benutzername «webadmin» vorgeschlagen. Dieser kann aber geändert werden.

swisstopo erzeugt dazu für jede Station ein Passwort und verwaltet diese mittels Keepass (oder einem ähnlichem Tool). Diese Passwörter werden jährlich erneuert.

Weder das root Passwort noch Passwörter anderer System-Benutzer (siehe oben) darf für das Webinterface verwendet werden!

Die IO Control Units implementieren keine Massnahmen um dies sicherzustellen.

Mit folgendem Befehl kann das Passwort geändert oder neue Benutzer hinzugefügt werden:

mk_passwd /etc/monkey/plugins/auth/users.mk <USER> <NEW PASSWORD>

Mit folgendem Befehl kann ein Benutzer entfernt werden:

mk_passwd -D /etc/monkey/plugins/auth/users.mk <USER>

Mit folgendem Befehl können alle Benutzer angezeigt werden:

cat /etc/monkey/plugins/auth/users.mk

(Siehe auch:
http://monkey-project.com/documentation/1.5/plugins/basic_auth.html)

3.8.3 Ausblick: HTTPS

HTTPS wird aktuell nicht unterstützt. Falls dies in Zukunft erwünscht ist, wird empfohlen auf den deutlich weiter verbreiteten und umfangreicheren Webserver nginx zu wechseln. Dieser benötigt mehr Ressourcen als der aktuell verwendete Webserver, dies sollte aber kein Problem für das System darstellen.

Falls signierte Zertifikate eingesetzt werden (empfohlen), muss geklärt werden, wie diese erstellt und verteilt werden. Vermutlich ist es dazu von Vorteil eine sub-domain für die IO Control Units zu erstellen und darin «wildcard» Zertifikate zu verwenden.

Falls für die Erstellung der Zertifikate einen ACME Client benötigt wird (z.B. für letsencrypt.org), muss geprüft werden ob dieser auf den Units ausgeführt werden muss oder ob ein anderer Rechner in der gleiche sub-domain diesen ausführen kann. Ersteres könnte schwierig werden, da üblich ACME Clients auf Grund deren Ressourcenbedarf kaum werden können. (Es gibt offenbar auch reine Bash-Script basierte Clients die möglicherweise eingesetzt werden könnten: https://github.com/Neilpang/acme.sh)

4 Update mittels RAUC bundle mit integriertem post-install hook

4.1 Hintergrundinformation

Das NAND auf dem SBC verfügt über zwei Partitionen auf auf denen jeweils ein System (OS und diverse Software) installiert ist. Die Partitionen sind wiederum unterteilt ins slots (dtb, rootfs, kernel).

Diese werden von RAUC in Zusammenarbeit mit dem Bootloader barebox verwaltet. Der Bootloader selbst liegt auf einer separaten Partition (nicht durch RAUC verwaltet).

Beim booten lädt der Bootloader das aktive System. Das andere System ist entsprechend inaktiv. Der Befehlen rauc status zeigt, welches System aktiv ist und welche Zustände die slots haben. Der Befehl rauc status mark-active other aktiviert das aktuell inaktive System, so dass es vom Bootloader beim nächsten Start geladen wird.

RAUC erlaubt ein Überschreiben des inaktiven Systems (während dem das aktive läuft). Um z.B. Konfigurationen vom aktiven

System in das neu geschriebene System zu übernehmen, bietet RAUC sogenannte hooks an. In unserem Fall wird ein post-install hook für den slot rootfs verwendet, um die Netzwerkkonfiguration zu übernehmen, genauer sind dies:

- /etc/hostname
- /etc/systemd/network/10-eth0.network
- /etc/systemd/network/10-eth1.network

Damit kann das ioControlUnit nach dem Update und Reboot wieder unter der gleichen Netzwerk-Adresse erreicht werden.

Alle anderen Konfigurationen müssen manuell wiederhergestellt werden. Dazu wird der unter 2.2 beschriebene Vorgang Installation-Vorgang erneut durchgeführt. (Eine umfangreichere Migration der aktuellen Konfiguration auf das neue System ist technisch zwar gut möglich, ist aber schnell sehr komplex und damit anfällig auf Fehler. Beispielsweise müsste möglicherweise Kombinationen von Version separat behandelt werden.)

4.2 Anleitung

RAUC bundle (.raucb) z.B. mit WinSCP oder aus Linux mit folgendem Befehl auf den SBC kopieren.

scp {DATEINAME}.raucb root@192.168.3.11:/home/root/

Auf SBC einloggen (als root) und RAUC Status prüfen:

rauc status

Typische Ausgabe:

```
Compatible:  phyboard-segin-imx6ul-2
Variant:
Booted from: system0
Activated:   rootfs.0 (system0)
slot states:
  dtb.1: class=dtb, device=/dev/ubi0_4, type=ubivol, bootname=(null)
      state=inactive, description=, parent=rootfs.1, mountpoint=(none)
  rootfs.0: class=rootfs, device=/dev/ubi0_2, type=ubifs,
bootname=system0
      state=booted, description=, parent=(none), mountpoint=(none)
      boot status=good
  kernel.1: class=kernel, device=/dev/ubi0_3, type=ubivol,
bootname=(null)
      state=inactive, description=, parent=rootfs.1, mountpoint=(none)
  rootfs.1: class=rootfs, device=/dev/ubi0_5, type=ubifs,
bootname=system1
      state=inactive, description=, parent=(none), mountpoint=(none)
      boot status=good
  kernel.0: class=kernel, device=/dev/ubi0_0, type=ubivol,
bootname=(null)
      state=active, description=, parent=rootfs.0, mountpoint=(none)
  dtb.0: class=dtb, device=/dev/ubi0_1, type=ubivol, bootname=(null)
      state=active, description=, parent=rootfs.0, mountpoint=(none)
```

Merken welches System aktiv ist. Im obigen Fall system0.

RAUC bundle überprüfen; insbesondere die fett markierten Stellen.

rauc info {DATEINAME}.raucb

Typische Ausgabe:

```
rauc-Message: Reading bundle: {DATEINAME}.raucb
rauc-Message: Verifying bundle...
Compatible:  'phyboard-segin-imx6ul-2'
Version:     'r0'
Description: 'PHYTEC rauc bundle based on BSP-Yocto-i.MX6UL-PD19.1.0'
Build:       '20210412163807'
Hooks:       ''
3 Images:
(1)  phytec-headless-image-phyboard-segin-imx6ul-2.ubifs
     Slotclass: rootfs
     Checksum:  7dbca7a7ecd1e13556ac47672871301849fb842c700{...}
     Size:      81645568
     Hooks:     post-install
(2)  zImage-phyboard-segin-imx6ul-2.bin.img
     Slotclass: kernel
     Checksum:  923752b89560026e8850a42c629dcb0830db6e4a1b{...}
     Size:      6725712
     Hooks:
(3)  imx6ul-phytec-segin-ff-rdk-nand.dtb.img
     Slotclass: dtb
     Checksum:  b3e3415676e768c3dbd29460b975a0f70d75b78b0f{...}
     Size:      33383
     Hooks:
0 Files

Certificate Chain:
 0 Subject: /O=PHYTEC Messtechnik GmbH/CN=PHYTEC Messtechnik GmbH
Development-1
   Issuer: /O=PHYTEC Messtechnik GmbH/CN=PHYTEC Messtechnik GmbH
PHYTEC BSP CA Development
   SPKI sha256:
E2:47:5F:32:05:37:04:D4:8C:48:8D:A6:74:A8:21:2E:97:41:EE:88:74:B5:F4:6
5:75:97:76:1D:FF:1D:7B:EE
   Not Before: Jan  1 00:00:00 1970 GMT
   Not After:  Dec 31 23:59:59 9999 GMT
 1 Subject: /O=PHYTEC Messtechnik GmbH/CN=PHYTEC Messtechnik GmbH
PHYTEC BSP CA Development
   Issuer: /O=PHYTEC Messtechnik GmbH/CN=PHYTEC Messtechnik GmbH
PHYTEC BSP CA Development
   SPKI sha256:
AB:5C:DB:C6:0A:ED:A4:48:B9:40:AC:B1:48:06:AA:BA:92:09:83:8C:DC:6F:E1:5
F:B6:FB:0C:39:3C:3B:E6:A2
   Not Before: Jan  1 00:00:00 1970 GMT
   Not After:  Dec 31 23:59:59 9999 GMT
```

Falls vorhanden Hook log file /home/root/rauc_hook.log löschen:

rm /home/root/rauc_hook.log

RAUC bundle installieren und Ergebnis überprüfen; insbesondere die fett markierten Stellen:

rauc install {DATETIME}.raucb

Typische Ausgabe:

```
rauc-Message: installing /home/root/phytec-headl{...}.raucb:
installing
rauc-Message: installing /home/root/phytec-headl{...}.raucb:   0%
Installing
rauc-Message: installing /home/root/phytec-headl{...}.raucb:   0%
Determining slot states
```

```
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  20%
Determining slot states done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  20%
Checking bundle
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  20%
Verifying signature
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  40%
Verifying signature done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  40%
Checking bundle done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  40%
Loading manifest file
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  60%
Loading manifest file done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  60%
Determining target install group
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  80%
Determining target install group done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  80%
Updating slots
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  80%
Checking slot rootfs.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  83%
Checking slot rootfs.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  83%
Copying image to rootfs.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  86%
Copying image to rootfs.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  86%
Checking slot kernel.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  90%
Checking slot kernel.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  90%
Copying image to kernel.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  93%
Copying image to kernel.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  93%
Checking slot dtb.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  96%
Checking slot dtb.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb:  96%
Copying image to dtb.0
rauc-Message: installing /home/root/phytec-headl{...}.raucb: 100%
Copying image to dtb.0 done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb: 100%
Updating slots done.
rauc-Message: installing /home/root/phytec-headl{...}.raucb: 100%
Installing done.
Installing `/home/root/phytec-headl{...}.raucb` succeeded
```

Überprüfen ob der post-install hook ausgeführt werden. Der hook schreibt dazu ein einfaches Logfile: /home/root/rauc_hook.log.

Sicherstellen, dass die Netzwerkkonfiguration kopiert wurde:

cat rauc_hook.log

Typische Ausgabe:

```
RAUC HOOK for AGNES called
post install hook
slot rootfs
RAUC_SLOT_MOUNT_POINT: /run/rauc/rootfs.0
copied hostame
copied 10-eth0.network:
[Match]
Name=eth0

[Network]
DHCP=ipv4
Address=192.168.3.11/24
```

```
copied 10-eth1.network:
[Match]
Name=eth1

[Network]
Address=10.180.252.85/28
DNS=131.102.100.110
DNS=131.102.37.142

[Route]
Gateway=10.180.252.81
```

(!) Wenn die Ausgabe nicht korrekt ist und das SBC notfalls nicht über die serielle Schnittstelle erreicht werden kann, den Vorgang abbrechen. (Falls keine Netzwerkkonfiguration kopiert werden konnte, sollte das SBC auf beiden Schnittstellen DHCP (ipv4) und default IPs eth0:192.168.3.11, eth1:192.168.4.11 verwenden. Falls falsche Angaben kopiert wurden, ist das Verhalten möglicherweise unbestimmt.)

Neu installiertes System aktivieren:

rauc status mark-active other

SBC neustarten:

reboot

Wieder mit dem SBC verbinden und als root einloggen (kein Passwort). Hinweis: Der SSH Client wird vermutlich vor einem veränderten Fingerprint warnen.

Prüfen, ob nun das frisch installierte System aktiv ist:

rauc status

Typische Ausgabe:

```
Compatible:  phyboard-segin-imx6ul-2
Variant:
Booted from: system1
Activated:   rootfs.1 (system1)
slot states:
  dtb.1: class=dtb, device=/dev/ubi0_4, type=ubivol, bootname=(null)
      state=active, description=, parent=rootfs.1, mountpoint=(none)
       rootfs.0:    class=rootfs,    device=/dev/ubi0_2,    type=ubifs,
bootname=system0
      state=inactive, description=, parent=(none), mountpoint=(none)
      boot status=good
       kernel.1:    class=kernel,    device=/dev/ubi0_3,    type=ubivol,
bootname=(null)
      state=active, description=, parent=rootfs.1, mountpoint=(none)
       rootfs.1:    class=rootfs,    device=/dev/ubi0_5,    type=ubifs,
bootname=system1
      state=booted, description=, parent=(none), mountpoint=(none)
      boot status=good
```

```
       kernel.0:    class=kernel,    device=/dev/ubi0_0,    type=ubivol,
bootname=(null)
      state=inactive, description=, parent=rootfs.0, mountpoint=(none)
  dtb.0: class=dtb, device=/dev/ubi0_1, type=ubivol, bootname=(null)
      state=inactive, description=, parent=rootfs.0, mountpoint=(none)
```

Überprüfen ob folgende, leere Datei vorhanden ist:

/home/root/hook-post-install-done

Wenn ja, wurde der hook ausgeführt. (Nur für Debug-Zwecke sinnvoll.) Datei löschen:

rm /home/root/hook-post-install-done

Das System ist nun bereit für die Installation gemäss Abschnitt 2.2.

Achtung: Die Installation erlaubt das Wiederherstellen der default Netzwerkkonfiguration. Unbedingt sicherstellen, dass dies nur in gewollten Fällen gemacht wird.

5 Versionierung

Version	Änderungen	Autor Datum
1.0	Initial	Steven Brossi, 2019-08-02
2.0	Update mit Anpassungen gemäss swisstopo. Streichung irrelevanter Punkte/Fragen (diese sind nun definiert).	Steven Brossi, 2019-08-30
2.1	Update mittels RAUC bundle mit integriertem post-install hook. Info zu Start-up Ausgangskonfiguration.	Steven Brossi, 2021-04-13