GARDENA IoT Gateway

A Journey from Anxiety to Collaboration

Andreas Müller & Reto Schneider
29.10.2019
Table of Contents

- Introduction
- Technical Background
- Open-Source from Scratch
- Keeping the Products Open
- Mainline FTW
- Questions & Contact Info
- Fun & Adventures (Backup)
- Hacking (Reference / Backup)
Introduction
About Us

Reto Schneider

- Embedded Developer GARDENA in Zurich
- little prior experience with Embedded Linux

Andreas Müller

- Embedded Developer (and Head of Embedded) GARDENA in Zurich
- no prior experience with Yocto Linux
Gateway/Project Background

Project Goals

- lower hardware costs
- maintainability for 10+ years
- open-source compliance

Challenges

- distributed part-time team
- no backup plan
- timeline: 1 year from project start to sale
Goals Of This Talk

We would like to talk about

- the technical aspects of our gateway
- GARDENA's open-source journey
- mainlining: status, benefits & how to convince your boss
- (our pitfalls and how to avoid them)
- (hacking our gateway)

Caveats

- the journey is ongoing; a lot is still left to do
- we are neither Yocto nor Embedded Linux experts
Technical Background
Hardware

Semi-custom Linux module with

- MediaTek MT7688 SoC
  - 580MHz MIPS 24KEc CPU
  - integrated Wi-Fi
- 128MB DDR2 RAM
- 8MB SPI NOR flash
- 128MB SPI NAND flash

Radio module for communication with devices

- SiM3U167 ARM Cortex-M3 MCU
- Si4476 Sub-GHz transceiver
Hardware - PCBA
Hardware - Details

- nRF52840 Bluetooth
- ARM Cortex M3
- Sub-GHz TRX
- EEPROM
- LNA
- SAW filter

- Linux Module
  - MT7688 SoC
  - 128 MB RAM
  - 8 MB NOR flash
  - 128 MB NAND flash

- MFi chip (HomeKit)

- Radio Module
  - ARM Cortex M3
  - Sub-GHz TRX
  - EEPROM
  - LNA
  - SAW filter

- Power supply

- Ethernet
- UART

Antennas:
- 868 MHz antenna
- 2.4 GHz antenna (WiFi, BLE)
Software – Flash Layouts

8 MB NOR:

- uboot: U-Boot code (640 kB)
- uboot_env0: U-Boot environment #1 (64 kB)
- uboot_env1: U-Boot environment #2 (64 kB)
- factory: calibration and configuration for MT7688 (64 kB)
- unused: free space (7360 kB)

128 MB NAND (UBI device with multiple volumes):

- rootfs0: squashfs (40 MB)
- rootfs1: squashfs (40 MB)
- kernel0: fitImage (4 MB)
- kernel1: fitImage (4 MB)
- overlay: UBIFS (30 MB)
Software – Build System

Mostly off the shelf plus own layers for...

- distribution
- custom packages
- 3rd-party software
- MediaTek MT7688 BSP
- device firmwares
Open-Source from Scratch

https://www.golem.de/1605/120646
## Legal issues with gateway

<table>
<thead>
<tr>
<th>Type</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug</td>
<td>In Progress</td>
<td>Unresolved</td>
</tr>
</tbody>
</table>

### Details

- **Type:** Bug
- **Priority:** Highest
- **Affects Versions:** None
- **Components:** Gateway
- **Labels:** None
- **P-Priority:** P3
- **Sub-Group:** 9031

The issue was reported by a user who encountered problems with the gateway functionality. The user reported that the gateway was not responding in certain scenarios, leading to application failures. The team is investigating the root cause and has been working on resolving the issue. The current status is 'In Progress', and the resolution is set to 'Unresolved' until the problem is fully addressed.

Additional notes:

- The gateway was updated to version 5.1.2, and the issue persisted.
- Users have reported similar issues on the support forum under the topic 'Gateway Performance Issues.' The community is actively discussing possible solutions.
- The team is looking into potential configuration issues and memory leaks as potential causes.

The issue is being marked as high priority and is being closely monitored by the development and testing teams.
Progress?

- **Added a comment - 2017-05-12 15:33**
  I prioritized this issue down to "MEDIUM", since it is no release blocker.

- **Reto Schneider added a comment - 2017-07-24 09:24**
  How much progress here?

- **Reto Schneider added a comment - 2017-07-28 12:35**
  Back to highest as this issue has a chance of hampering or even killing our business and/or reputation.
Die angedrohte Strafe ist hapig: **250.000 Euro** Ordnungsgeld oder ersatzweise **bis zu sechs Monate Haft** drohen Mike Decker, Geschäftsführer der Firma Geniatech Europe aus Herzogenrath bei Aachen, sollte er noch einmal das freie Betriebssystem Linux verbreiten und dabei gegen die komplizierten Lizenzbedingungen verstoßen. Erstritten hat das Urteil des Landgerichtes Köln der in der Szene umstrittene Entwickler Patrick McHardy. Am kommenden Mittwoch wird sich das Oberlandesgericht Köln mit dem Fall beschäftigen.

https://heise.de/-3986181
Legalize Gateway

Type: Epic
Priority: High
Affects Version/s: None
Component/s: Device: Gateway
Labels: None
Epic Name: Legalize Gateway

Status: IN PROGRESS
Resolution: Unresolved
Fix Version/s: None
Problem (legally) resolved, but some internal improvements still outstanding.

We should aim for a solution as soon as possible due to high financial risk.
https://github.com/husqvarnagroup/smart-garden-gateway-public
Keeping the Products Open
First Gateway: Reverse Engineering by Customers

Rooting

Description
If you hard reset the device via the reset button, it will copy a recovery partition over the root filesystem. After that the device will boot up, load its settings from the uboot environment, generating ssh-keys and then setting a random root password. If we manage to quickly login before the password is set, we are in.

Process
1. Remove the power plug
2. Press and hold the hard reset button on the back of the device to reset it to factory
3. Insert the power plug.
4. Wait a while and release the hard reset button, when you see the device reading / writing from nand
5. Wait a while for a login prompt
6. Quickly enter the username "root"
7. Enter the top command and wait until the ssh-key generation is done.
8. Set your desired root password
9. You are in ;)

https://github.com/gardena-smart-reverse-engineering/
Developers Perspective

- reliable lockdown is difficult and expensive
- local security is not important here
  - much easier for an attacker to just turn off the water in the garden, than to access the gateway inside the house
- you buy it, you own it
- conclusion: we would prefer to just leave local root access via UART open
Quality Department: Convincing & Explaining

Empty password on LC-Gateway (SSH) is seen as critical from point of claim analysis

(note: SSH password was never empty)
[ OK ] Reached target Multi-User System.
Starting Update UTMP about System Runlevel Changes...

Dear customer

You can login to your GARDENA smart Gateway with user 'root' without a password. Please understand that by doing so, you will void your warranty. If anything breaks, you can try to do a factory reset (hold the reset button while powering on the gateway); however if that does not resolve the problem, customer support will not be able to help. Please refrain from contacting customer support as this will cause us to lock down future devices in the long run. For questions regarding open source, please contact smart.open.source@husqvarnagroup.com.

GARDENA smart Gateway 3.11.1 GARDENA-0b9b03 ttyS0
GARDENA-0b9b03 login: •
Mainline FTW
MT7688 Support in April 2018

MediaTek:
- U-Boot: based on 1.1.3 from 2005
- Linux: 2.6.x
- OpenWrt: 3.10 from 2014

Upstream:
- U-Boot: No support at all
- Linux: Not supported
- OpenWrt: Supported, Linux 4.14 (LTS)
MT7688 Support: Way Out (Plan)

U-Boot

- hired U-Boot initiator (DENX) to implement support for our hardware

Linux

- ported patches over from OpenWrt

Management Rationale

- lower project risk
- cost savings with smaller NOR flash
November 2018

- U-Boot support worked very well
- keeping DENX to work on updating to Linux 4.19 (LTS)
- impending manufacturing start and sale for gardening season 2019
Mainlining Win #1: Wi-Fi Alliance Certification

Solution: Hire (very motivated) mt76 maintainer, solved 45 man hours later
Mainlining Win #2: BitFlips

January 2019: 30 out of 250 devices failed selftest

[16401.605487] ubi0: scrubbed PEB 340 (LEB 0:76), data moved to PEB 354
[16401.616238] ubi0: fixable bit-flip detected at PEB 354
[16401.616254] ubi0: schedule PEB 354 for scrubbing
[16401.661569] ubi0: fixable bit-flip detected at PEB 354
[16401.730866] ubi0: scrubbed PEB 354 (LEB 0:76), data moved to PEB 340
[16401.736318] ubi0: fixable bit-flip detected at PEB 340
[16401.736335] ubi0: schedule PEB 340 for scrubbing

Solution: Kernel 4.19 with new, mainline SPI-NAND framework
October 2019

Results so far:

- project successful
- MT7688 well supported in upstream Linux and U-Boot
- mt76 WLAN driver passed certification for at least two other companies

Follow up project:

- porting old gateway to new code base
- upstream support to Linux and U-Boot
Mainlining Conclusion

Probably best risk reduction we had

- mainlining cost less than 10% of the project budget
- project likely would have failed without it

Upstreaming as much as possible is in our own best interest

- small embedded team
- maintenance done by 2 embedded devs (us; part-time)
- upstreaming keeps our future workload low
- synergies with other MT7688 users

Outsourcing helped with speed and quality

- DENX and embeDD are much better with low-level work
Questions & Contact Info
Thank you for your attention

Questions?

Feedback? Ideas? Let us know!

- pull requests are also welcome :-)

Contact

- smart.open.source@husqvarnagroup.com
- reto.schneider@husqvarnagroup.com
- andreas.mueller@husqvarnagroup.com
Fun & Adventures (Backup)
Certificates

- during manufacturing, the bootstrapping server generates a key/certificate pair
- the certificate is signed by a service on an external server (bad idea!)
- at one point, this server started simply returning an empty body, rather than the certificate (but still HTTP status 200 OK)
- we were lucky to have a test

```python
def test_013_openvpn_client_certificate(self):
    """Check the plausibility of the stored OpenVPN certificate"""
    conf_openvpn_crt = fw_getenv("conf_openvpn_crt")
    self.assertTrue(conf_openvpn_crt)
    self.assertTrue('%-----BEGIN CERTIFICATE-----' in conf_openvpn_crt)
    self.assertTrue('%-----END CERTIFICATE-----' in conf_openvpn_crt)
```
Python Cache Problems (I)

Symptoms:

root@GARDENA-0b8a80:/usr/lib/python3.5/site-packages# /usr/bin/ipr-setup
Traceback (most recent call last):
  File "/usr/bin/ipr-tool", line 186, in <module>
    main()
  File "/usr/bin/ipr-tool", line 178, in main
    initialize_gateway_id()
  File "/usr/bin/ipr-tool", line 56, in initialize_gateway_id
    batch_id = get_batch_id()
  File "/usr/bin/ipr-tool", line 32, in get_batch_id
    body = bootstrap_get_batch()
  File "/usr/lib/python3.5/site-packages/bootstrap.py", line 64, in bootstrap_
    return bootstrap_server_get(BOOTSTRAP_SERVER_BATCH_CONFIG_PATH)
  File "/usr/lib/python3.5/site-packages/bootstrap.py", line 30, in bootstrap_
    connection.request("GET", path)
  File "/usr/lib/python3.5/http/client.py", line 1107, in request
    self._send_request(method, url, body, headers)
  File "/usr/lib/python3.5/http/client.py", line 1152, in _send_request
Python Cache Problems (II)

Problem analysis:

- during manufacturing, we ran a python script, which turned an LED green when done
- manufacturer immediately removed power when LED green
- python cache was generated but not fully written to disk

Lessons learned:

- always sync before turning LED green
- barrier when copying files (cp; sync; mv)
- generate python cache during build time

Manual fixing:

```bash
fw_printenv | grep "^[a-z_]\+/(done|finalized|passed\)=" find /usr/lib/python3.5/ -type d -name __pycache__ -exec rm -r {} \;
sbin/fw_setenv fct_finalized && rm -f /etc/fct_finalized
fw_printenv fct_finalized || [ "$(fw_printenv -n self_test_passed)" == "1" ]
sbin/fw_printenv -n self_test_passed && fct-tool --set-leds green && echo OK
```
Hacking (Reference / Backup)
Building the Software

REPO="https://github.com/husqvarnagroup/smart-garden-gateway-public"
git clone --recurse-submodules $REPO
cd smart-garden-gateway-public
export TEMPLATECONF=${PWD}/yocto/meta-distribution/conf
source yocto/openembedded-core/oe-init-build-env build-gardena
bitbake gardena-image-opensource-prod
Flashing Custom Software

- assemble UART connector and connect via terminal (settings: 115200n8)
- set up TFTP
- place kernel & rootfs image in TFTP directory
  - Rename gardena-image-opensource-prod-gardena-sg-mt7688.squashfs-xz to gardena-image-prod-gardena-sg-mt7688.squashfs-xz
- reboot
- send any character to enter U-Boot
- run "ubi part nand" and "do_flash" for programming
- run "env set update_url invalid && saveenv" to prevent fetching of updates
  - run "env set update_url && saveenv" to re-enable them

CPU: MT7628 Rev 1.2 - Boot from XTAL (3-Byte SPI Addr)
Model: Gardena smart-Gateway-MT7688
DRAM: 128 MiB

Loading Environment from SPI Flash... SF: Detected xm25qh64a with page size 256

OK

Watchdog: Started

F-Data: factory-data version 1 detected

Net: eth0: eth@10110000

Hit any key to stop autoboot: 0

=> run do_flash

BOOTP broadcast 1
DHCP client bound to address 10.42.0.2 (10 ms)

Using eth@10110000 device

TFTP from server 10.42.0.1; our IP address is 10.42.0.2
Filename 'fitImage-mt7688.bin'.
Installing Additional Packages via OPKG

root@GARDENA-0b9b03:~# opkg update
Updated source 'uri-all-0'.
Download http://sg-low-cost-gateway-dev.s3.amazonaws.com/feeds/opkg-head/mi
Updated source 'uri-mips32r2el-24kc-nf-0'.
Download http://sg-low-cost-gateway-dev.s3.amazonaws.com/feeds/opkg-head/mt
Updated source 'uri-mt7688-0'.
root@GARDENA-0b9b03:~# opkg install tcpdump
Installing tcpdump (4.9.2) on root.
Download http://sg-low-cost-gateway-dev.s3.amazonaws.com/feeds/opkg-head/mi
Configuring tcpdump.
root@GARDENA-0b9b03:~#