NuttX in Long Range RFID Readers

Mattias Edlund
info@mattiase.com
Agenda

- RFID
- TagMaster
- Why NuttX?
- Our Implementation
- A Minimal Boot Loader
- My NuttX Wish List
RFID
RFID

• Radio Frequency IDentification

• Uses radio waves to identify and/or track tagged objects

• A system consists of readers and tags

• Tags contain electronically stored information

• Active tags have a local power source and an active transmitter

• Passive tags collect energy from the radio waves and do not have an active transmitter
RFID Frequencies

- 125 kHz
- 13.56 MHz
- 900 MHz
- 2.4 GHz
- 5.8 GHz
Backscattering

- A reader has a radio transmitter and receiver
- A tag does not have a radio transmitter
- Compare with a flashlight and a mirror
RAIN RFID

- Global standard for backscattering RFID @ 860-960 MHz
- Used to identify, locate and authenticate things
- Typical read range up to 10 meters
TagMaster

- Founded 1994 with HQ in Kista, Sweden
- Originally a 2.45 GHz RFID company
- Now a global group of companies, focusing on traffic and rail solutions for Smart Cities
TagMaster - RAIN RFID Tags

WindShield Tag

ISO Card

HeadLight Tag

Other tags...
TagMaster - RFID for Parking
TagMaster - RFID for Rail
TagMaster - RFID for Road Tolls

24/7
Why NuttX?
2005 Reader Platform Getting Old

- RF module approaching end of life
- Unused legacy RF interface
- Aging Linux system
- Diverging user requirements
  - Most users want a cheap reader and don’t care about the Linux system.
  - Many users only need a few of the interfaces
  - Power users need the programmable Linux system but wants higher performance and more memory.
A New Scalable Architecture
Needing a New Operating System
Existing Software Architecture

- Existing applications depending on shared library and Linux APIs for networking, serial ports, file system, etc.
- Kernel driver for time critical radio control and application specific interfaces.
Operating System Wish List

- Open source
- Available for STM32
- As close to Linux as possible
- Drivers for RS-232, RS-485, microSD, GPIOs, USB, and Ethernet
- Networking support (TCP/UDP)
- Web server
Selecting NuttX (this was 2013)

- NuttX seemed to be the perfect fit, but...
  - There was no big organization behind it
  - There was no big user group
  - The future was unclear
- Due to our previous good experience with open source software we decided to build a prototype.
- After a few days we had a basic but working RFID reader!
Our Implementation
Hardware Examples

RFID Reader with NuttX (2013)  
RFID Reader with NuttX + Linux (2018)
Modified Software Architecture

- **Applications**
- **Shared Library**
- **Daemon**
- **Kernel Driver**
  - **Linux Kernel**

**TCP (local/remote)**

**read(), write(), ioctl()**

---

- **NuttX Tasks**
- **Shared Library**
- **NuttX Driver**
- **NuttX**

**TCP (remote)**

**read(), write(), ioctl()**
NuttX + Linux

- Linux system communicates with NuttX system over on-board USB
- Applications and web interface compiles from the same code for both NuttX and Linux
- The microcontroller binary is exactly the same in both systems
Demo
A Minimal Boot Loader
A Minimal Boot Loader

- Invisible during normal start
- Public key verification of stored firmware
- Web interface for firmware upgrade
- Fail safe environment for user settings
- Minimal footprint (< 16 KB + environment)
The boot loader is started if:

- The “Force boot loader” DIP switch is active
- The user has requested start of the boot loader from software (through magic number in RAM)
- The RSA signature of the firmware is not ok

In all other cases, the MCU is reset and the firmware is started.
Web Interface for Firmware Upgrade

![Firmware Upgrade Interface](image-url)
Fail Safe Environment

- Stores user settings as typical environment variables: NAME=VALUE
- Shared between boot loader and firmware (IP settings, etc.)
- Keeps settings when firmware is upgraded
- Guarantees that a value is either completely written or not written at all even if power is lost during a write
- Requires two erasable flash sectors with single byte write capability
Minimal Footprint

- Boot loader and environment fits in the first three 16 KB sectors on STM32F407
- These sectors are not overwritten when firmware is upgraded (all user settings are saved)
My NuttX Wish List
My NuttX Wish List

● Let NuttX (continue) to be “Linux on a microcontroller”
  – Many developers are familiar with Linux
  – Use similar APIs whenever possible

● The Linux features that were missing in NuttX became a way for us to differentiate our products. If available we would have used:
  – Discovery protocols: UPnP, mDNS/DNS-SD (Bonjour/Avahi)
  – Secure network protocols: TLS, HTTPS
  – Network Time Protocol: NTP
Thank you for listening!