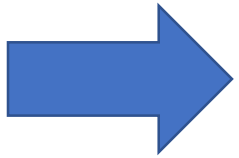
Aerodust

Using (rooted) Smart Home Devices for cool things

By Dennis Giese, Andrew Tu @HackBeanpot 2018, Boston

Introduction

- developed on the HackBeanpot 2018
  - Very hacky implementation due to limited time
- Prior work:
  - Dennis Giese, Daniel Wegemer: Rooting of Xiaomi Smart Home Devices
  - Presented on Chaos Communication Congress (34C3), Recon BRX 2018 *
- Initial idea: What to do with a rooted Vacuum Cleaning Robot?
  - Rooting already done, but cool applications were missing



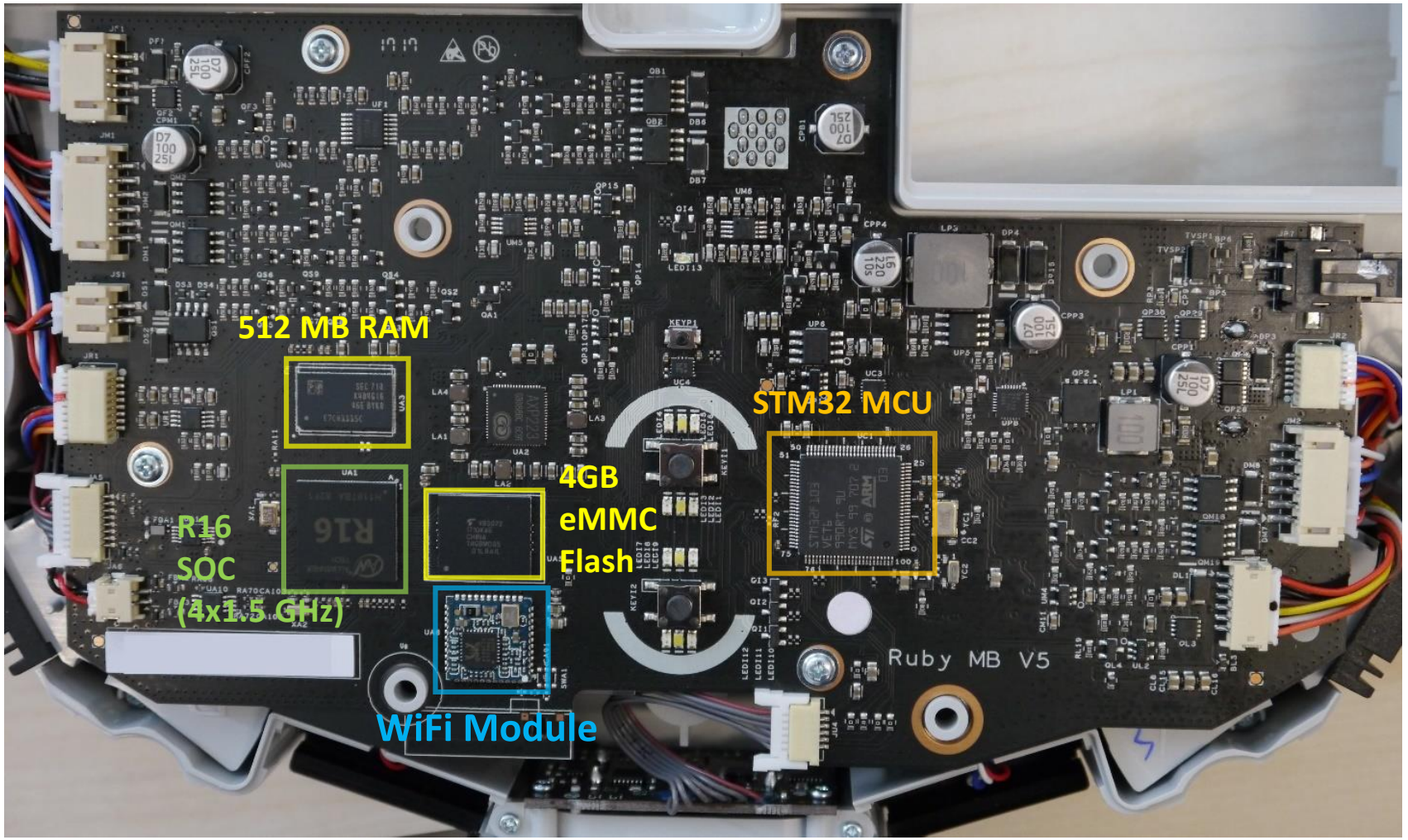
Let’s build a mobile WiFi mapping device

*See: https://github.com/dgiese/dustcloud/

Why WiFi mapping device?

- Demonstation of features of vacuum robot
- Measuring Signal strength correlated with position
  - Due Lidar based navigation: high precision
  - Automated measurement
  - Easily repeatable
  - Cleaning floor while measurement ;)
- Goal: create Map of WiFi signal, map available WiFi APs/SSIDs

Hardware: Xiaomi Vacuum Cleaning Robot

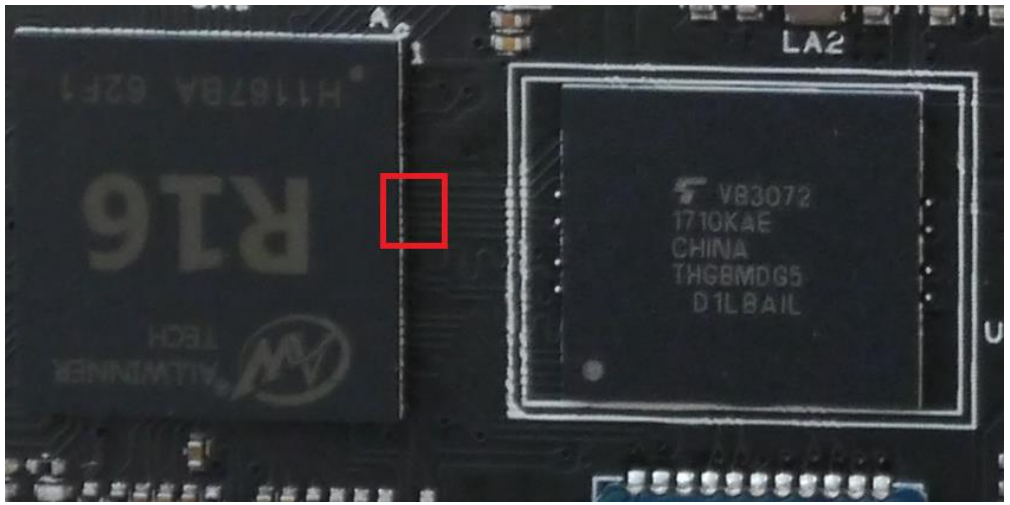


Overview sensors

- 2D **LIDAR** SLAM (5*360°/s)
- Gen1 only**: Ultrasonic** distance sensor
- multiple **IR** sensors
- 3-axis **Magnetic** Sensor
- 3-axis **accelerometer**
- 3-axis **gyroscope**
- **Bump** sensors

Rooting

The weapon of choice:



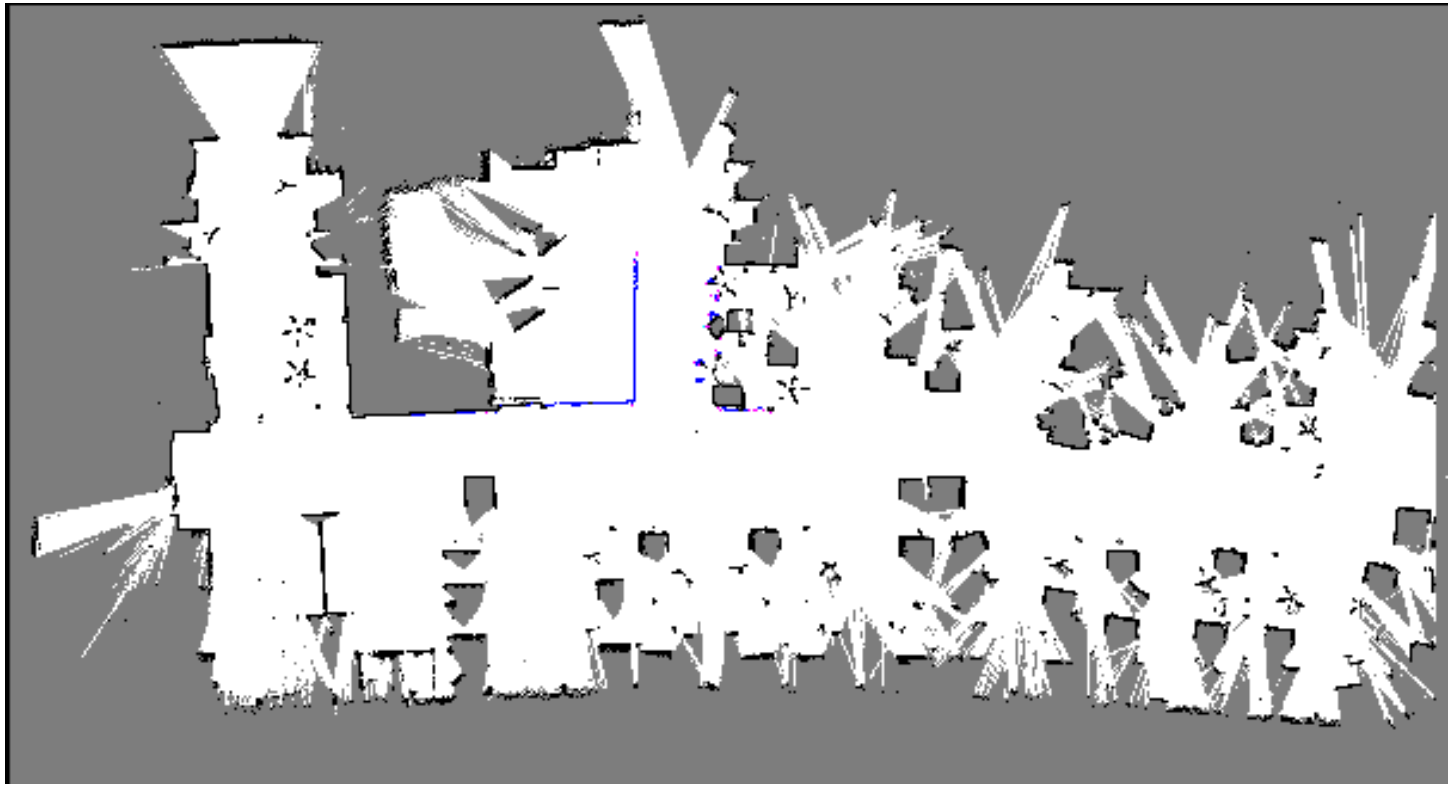
Exact procedure see here:
https://github.com/dgiese/dustcloud/blob/master/presentations/Recon-BRX2018/recon_brx_2018-final-split.pdf

Software

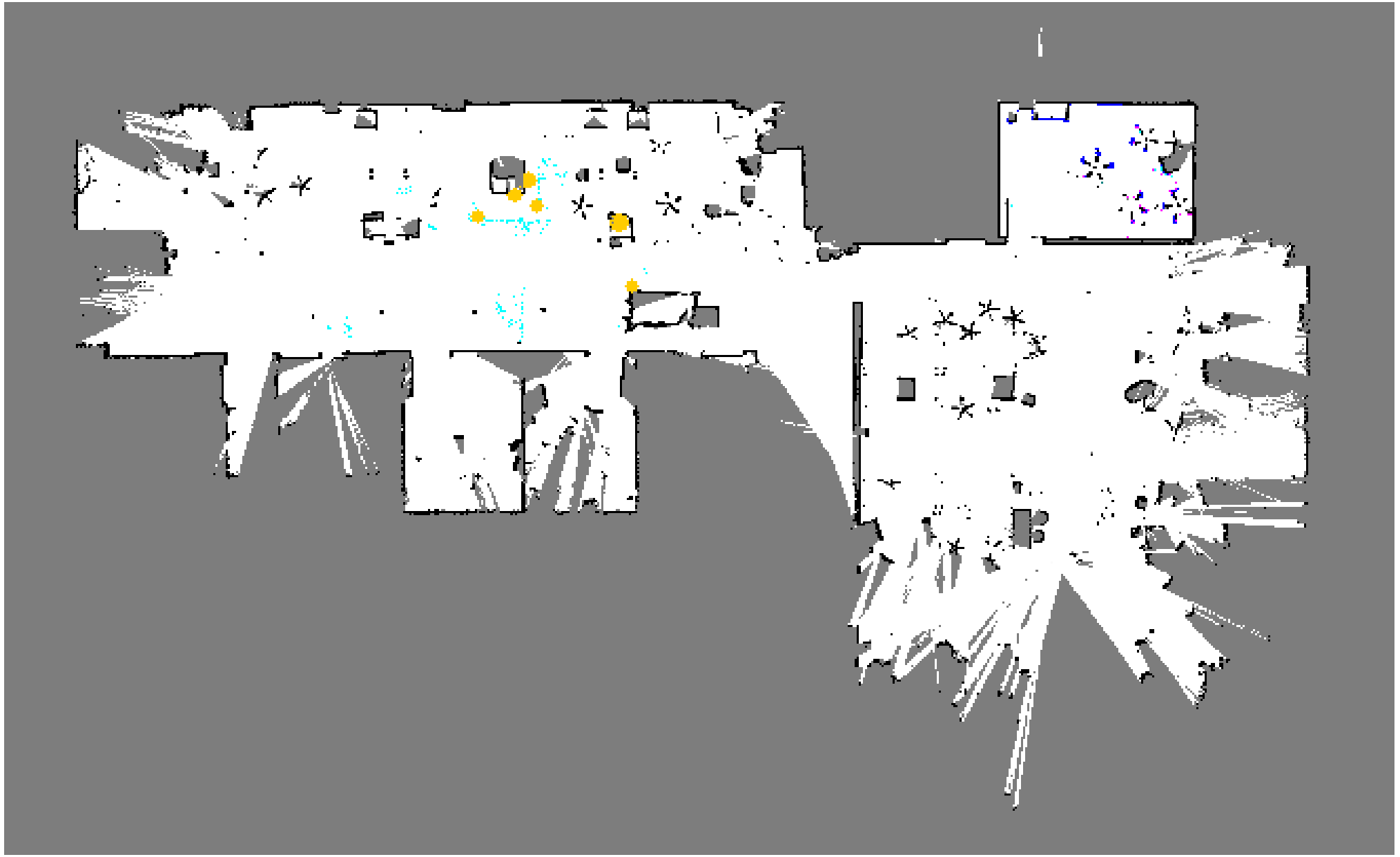
- Ubuntu 14.04.3 LTS (Kernel 3.4.xxx)
  - Mostly untouched, patched on a regular base
- Player 3.10-svn
  - Open-Source Cross-platform robot device interface & server
- Proprietary software (/opt/rockrobo)
  - AppProxy
  - RoboController
  - Miio_Client
  - Custom adbd-version
- iptables firewall enabled
  - Blocks Port 22 (SSHd) + Port 6665 (player)

Available data on device

- Maps
  - Created by player
  - 1024px * 1024px
  - 1px = 5cm
  - PPM format
- Logfiles
  - Navigation
  - Sensor Data



Northeastern University, ISEC Building, 6th floor



Genuine + Jack Morton Office, NE Side,5th floor

Genuine + Jack Morton Office, NE Side,5th floor

Approaches

- Initial approach: compile driver into player software
  - On robot: only binary format of player available
  - Not successful as player not modular after compilation
- Second approach: usage of player client interface
  - Client interface allows subscription of sensor data
  - Problem: Position data not supported (custom plugin by Xiaomi/Rockrobo)
- Third approach: parsing of the runtime logs
  - Player Software: creation of massive logs with a lot of debug information
  - Sub-sensor “position2d” returns useful data

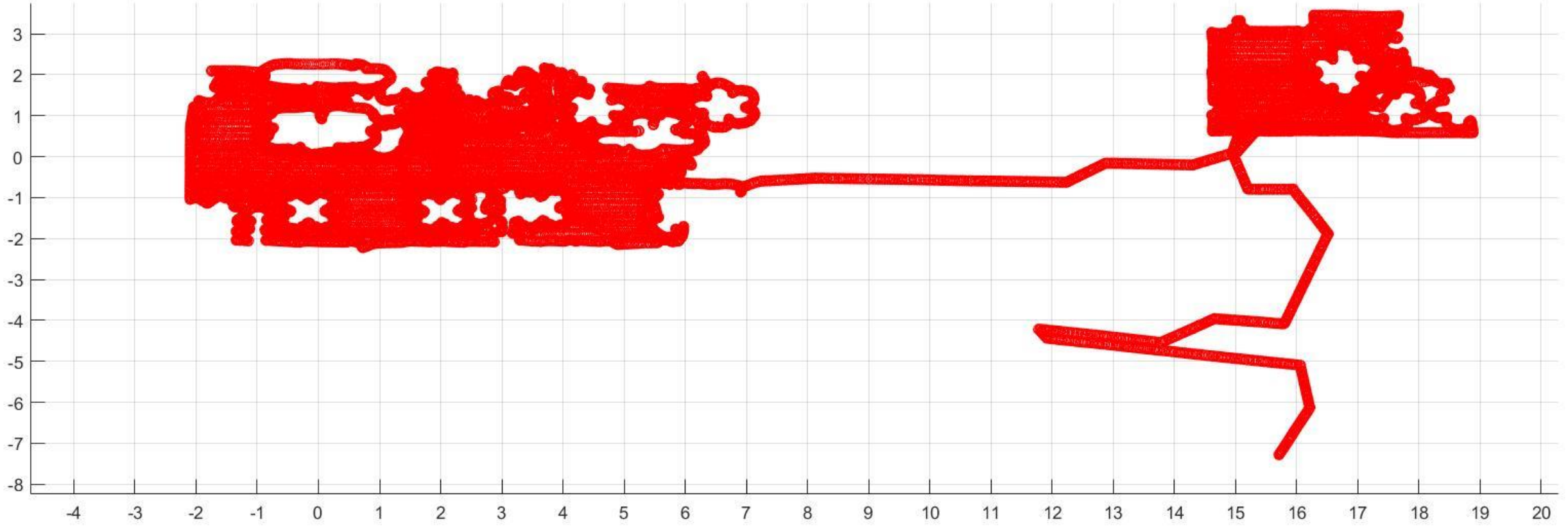
Implementation

- Installation of Python3 on vacuum robot
- Python script:
  - Parsing of position2d from player logfile

    {x_pos, y_pos, yaw_pos, x_vel, y_vel, yaw_vel}
  - Retrieving WiFi information from Linux kernel

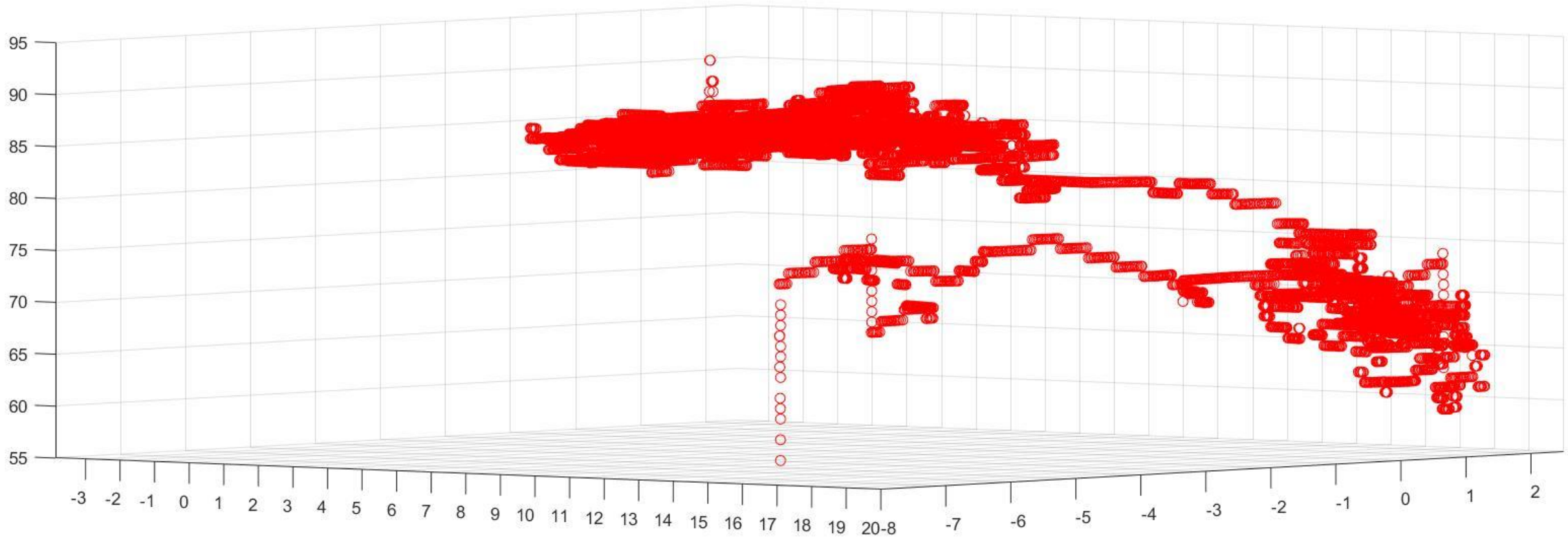
    {link, level, noice, SSID, BSSID)
  - Creation of CSV-File with position and signal strength
    - 100ms per entry
    - Import into Matlab
  - Future work: direct representation of signal in map

Matlab output: Point cloud



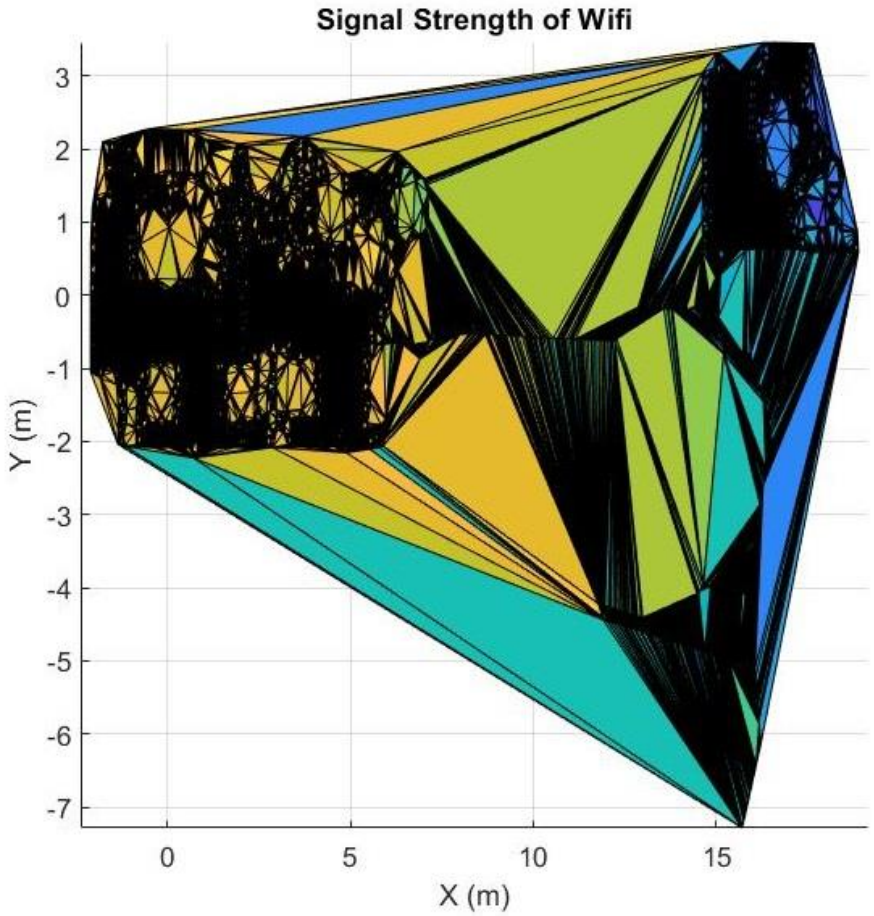
Not in scale
Genuine + Jack Morton Office, NE Side,5th floor
SSID: Vaccums, Ch: 11, Tx_PW: 20 dbm

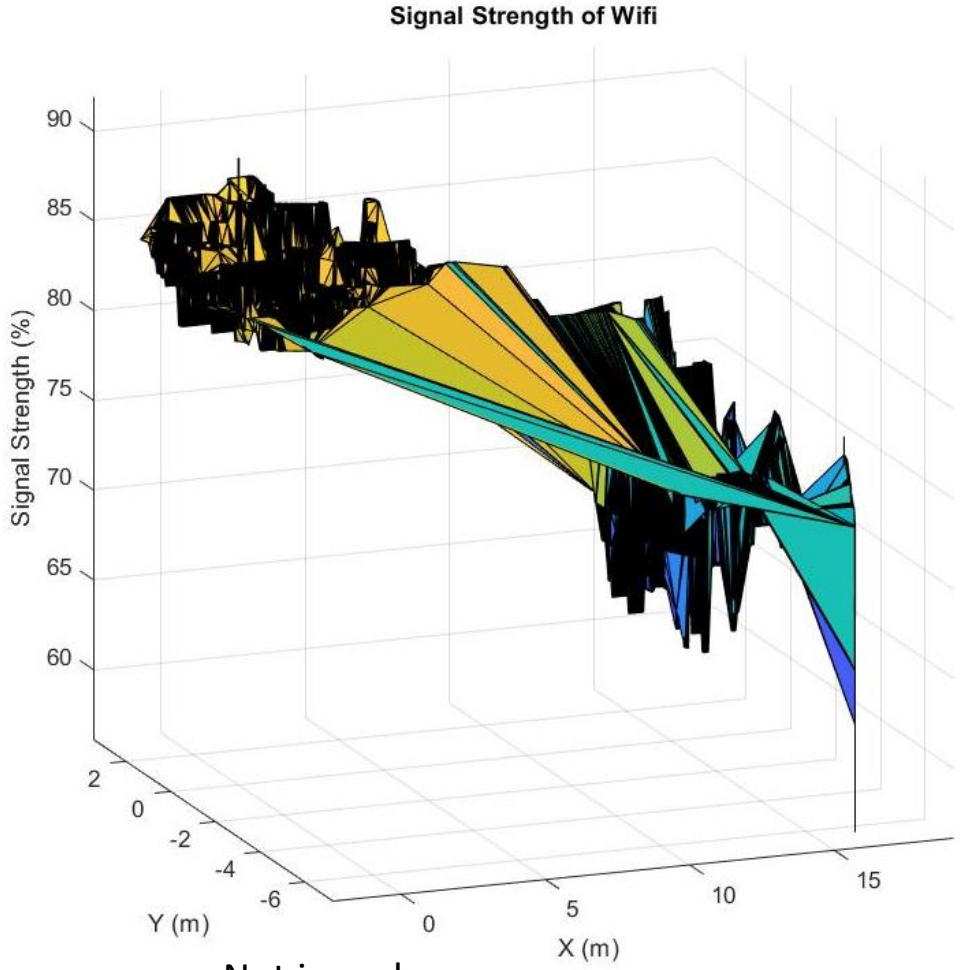
Matlab output: Point cloud



Not in scale
Genuine + Jack Morton Office, NE Side,5th floor
SSID: Vaccums, Ch: 11, Tx_PW: 20 dbm

Matlab output





Not in scale

Genuine + Jack Morton Office, NE Side,5th floor

SSID: Vaccums, Ch: 11, Tx_PW: 20 dbm

Limitations

- WiFi: 1x1 single band b/g/n adapter (2.4 GHz)
  - Only 2.4 GHz WiFi measurable
  - RSS-value instead of more complex “Channel state information”(CSI)-values
    - CSI would enable to localize the position of the AP or to detect reflections
  - Possible solution: resolder WiFi card or attach one to USB ;)
- Height
  - Measurement is only “precise” for the level of the vacuum robot

Not realized while HackBeanpot

- Multiple SSIDs
- Nice GUI / Webinterface
- Realtime WiFi heatmap generation
- Correct Matlab scaling
- Automatic alignment player map <-> WiFi Heatmap