

Research Project I: Drone-based Autonomous Generation of 3D Maps of Indoor Radio Environments

Ken Mendes

Department of Computer Science

University of Antwerp

Antwerp, Belgium

ken.mendes@student.uantwerpen.be

Abstract—A Radio Environment Map (REM) is an important instrument for increasing the radio awareness of different agents. Building such an REM - especially in three dimensions - can be a cumbersome process. This project shows how data for a Radio Environment Map can be gathered for an indoor three-dimensional space using a drone. It discusses the requirements that need to be met for such a drone and proposes a design with the required positioning and beacon scanning functionality. It demonstrates this can be achieved in an autonomous way, with a positioning accuracy of 10cm and a flight time of up to 6 minutes, allowing it to visit and scan more than 30 locations. It concludes with tests demonstrating the concept, capabilities and endurance of the design.

Index Terms—drone, autonomous, REM, Radio Environment Map, Indoor, Crazyflie, ESP8266

I. INTRODUCTION

A Radio Environment Map (REM) documents radio signals and their properties over a geographic area. These properties can among others include the frequency, strength and protocol of the radio signal and are stored together with the location where they were measured. These REMs and the data they hold can then be used for a variety of purposes including data mining or aid in cognitive radio.

This project's aim is to be able to build such an REM in 3D of indoor volumes with the use of a drone. The scope is limited to mapping IEEE 802.11b, 802.11g and 802.11n Wi-Fi signals (2.4 GHz band) in a large room. The drone takes radio measurements at a set of $\langle x, y, z \rangle$ coordinates within the room and stores the collected properties of the different radio signals together with the coordinates where they were collected.

The most similar existing effort to our work is [1], where the authors propose a drone-based system for the generation of a training dataset for 3-dimensional WiFi fingerprinting-based indoor localization. In contrast to their work focusing solely on WiFi-based fingerprinting, we argue that REMs can be beneficial and utilized more broadly, for example in optimizing the positioning of UAVs serving as mobile relays [2] or planning the extensions of any wireless networking infrastructure by adding APs/base stations to cover “dark” connectivity regions in an environment of interest [3].

With that in mind, in our work we posit two additional design requirements compared to [1]. The first one is a modular design of the interface between the drone-based part of the system and a given chipset under study. This requirement allows for a simple integration of different types of chipsets (e.g., LoRa, BLE, mmWave) with the drone, extending our future experimentation capabilities well beyond 3D REM generation in traditional WiFi. The second is to guarantee no interference between the drone-based part of the system and the chipset the drone is carrying, allowing highly repeatable measurement collections with minimized external influences.

The main components of our design are a small Crazyflie drone, a high-accuracy UWB based positioning system and an ESP8266 Wi-Fi module. The drone is operated by an application residing on a base station, communicating over radio with the drone. This allows to autonomously collect measurements with a positioning accuracy of 10cm at a rate of about 10 seconds per waypoint for up to 6 minutes.

The repeatability of experimentation is unprecedented compared to a person carrying a measurement device, allowing us to e.g., obtain multiple 3D REMs at effectively the same locations at different time instances or testing the effects of different (types of) WiFi chipsets on the quantity of the generated REM.

II. DRONE SELECTION

A. Requirements

There are several hard requirements for a drone that gathers the data for building an REM of an indoor volume:

- 1) The drone needs to be able to fly and be controlled well indoors
- 2) Its position in 3 dimensions within a room needs to be determined with relatively high accuracy (decimetre-level) as this directly influences the accuracy of the REM
- 3) It needs to be able to handle Wi-Fi signals, more specifically receive IEEE 802.11 beacon frames [4] and should at least:
 - Record the MAC address of the transmitting Access Point (AP)

- Record the Service Set Identifier (SSID) stored in the beacon's body
- Calculate the Received Signal Strength Indicator (RSSI)

In addition it would be practical if this could be done in a completely autonomous way, making use of waypoints to discover the volume.

B. Implications and challenges

1) *Indoor positioning:* Many off-the-shelf drones are equipped with Global Navigation Satellite Systems (GNSS) receivers like GPS or GLONASS. These are however not usable in many indoor environments due to signal loss leading to reduced accuracy and long acquisition times as shown by Kjærgaard et al. in [6].

A variety of Indoor Positioning Systems (IPS) is available as surveyed by Al-Ammar et al. in [7] Several of these technologies provide the room-scale range and decimetre-level accuracy required for this project including Ultrasonic, Computer Vision and UWB as shown by Song et al. in [8].

2) *Wi-Fi receiver:* Drones come with a radio transceiver for control and often come with a Wi-Fi adapter for video transmission, these transceivers might not be usable for custom purposes like frequently scanning for Wi-Fi beacons as is our intention. If that is the case, an additional Wi-Fi transceiver will need to be foreseen and integrated in order to collect beacon data.

An additional difficulty is that these different radio transceivers will likely interfere with each other.

C. Selection

A range of sub €1000 off-the-shelf drones from brands like DJI, Parrot, Yuneec and BitCraze were compared based on the requirements and challenges above, this comparison can be found in appendix A . Eventually the choice was made to use a BitCraze Crazyflie 2.1 nano-drone [9] as shown in figure 1 for this project for the following reasons:

- The Crazyflie is an open platform - open hardware and open software - allowing it to be easily extended and adapted to the needs of the project
- It supports a Loco Positioning System, an IPS that uses Ultra Wide Band (UWB) technology which provides the required accuracy as discussed earlier
- The ability to mount and integrate additional custom electronics, this would make it possible to integrate a Wi-Fi module into the drone for beacon scanning
- The Crazyflie can be controlled by software running on a base station connected over a USB radio transceiver (The CrazyRadio). This allows to fully automate the project from sending waypoints to parsing and showing the results of scans.



Fig. 1. The Crazyflie 2.1

III. DESIGN

A Crazyflie 2.1 drone comes with a FreeRTOS [10] based operating system and a radio and Bluetooth LE transceiver for control. It provides an accelerometer, gyroscope, magnetometer and a high precision pressure sensor through its 10-DOF Inertial Measurement Unit (IMU).

These capabilities can be extended by adding up to 2 expansion boards or decks as shown in figure 2. For this project, both expansion slots are used: one for the Loco Positioning Deck, the other to add a custom integrated Espressif ESP8266 Wi-Fi module. So there is one deck that provides positioning functionality and one deck that will be responsible for scanning Wi-Fi beacons.

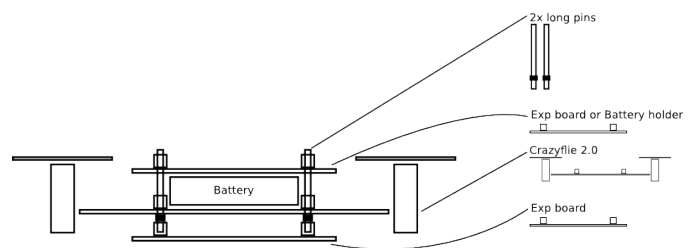


Fig. 2. Crazyflie expansion boards

These expansion decks need a way to communicate with the drone itself. To allow this, the Crazyflie provides a set of interfaces that can be addressed over 20 pins accessible to each expansion deck. Figure 3 documents the pin allocations, one can choose between 2 UART, an I²C, an SPI and an STM32 IO interface. The Loco Positioning Deck uses the UART1, STM32 IO and SPI interfaces [11], the custom ESP8266 deck uses the UART2 interface.

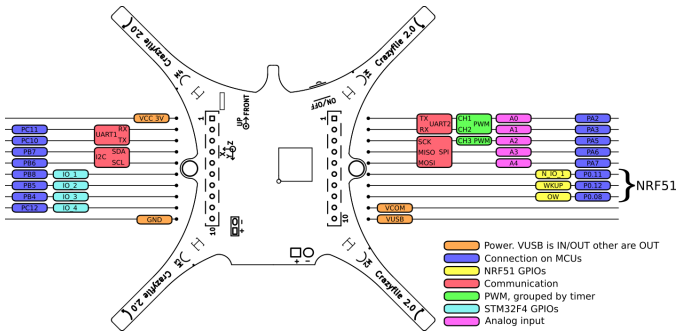


Fig. 3. Crazyflie interfaces and pin allocation

Controlling the drone is done through a Python application that makes use of the Crazyflie Python library [12]. This application can communicate with the drone over the Crazyradio, send instructions to move to a waypoint and scan for beacons, parse the results and store them for later processing.

A more detailed explanation of how the Crazyflie platform is customised and set up for this project can be discovered in the following sections:

A. Positioning

This project makes use of the Crazyflie’s Loco Positioning System (LPS) to provide accurate indoor positioning. This system works with a tag, the Loco Positioning Deck, and multiple anchors distributed over the volume to scan. This way, the tag can calculate its own position based on the signals it receives from the anchors.

The Loco Positioning System is based on the Decawave DWM1000 module and uses Ultra-Wideband (UWB) radio technology for communication, it allows localization to a precision of 10 cm [13]. Using UWB for indoor positioning is interesting for several reasons. It has a decent range of about 10m, operates in the 3.1-10.6 GHz range which is outside the 2.4 GHz band of the IEEE 802.11 specifications. Additionally, it can operate without causing interference in the same spectrum occupied by other radio services as discussed by D. Porcino and W. Hirt [14].

A minimum of 4 anchors is required to perform multilateration in three dimensions. While this is theoretically sufficient, Bitcraze advises to use at least 6 LPS anchors. Negative influences in the system’s setup or environment like packet loss, no line-of-sight to an anchor or tag orientation can all hinder performance. A Crazyflie equipped with an LPS deck makes use of an extended Kalman filter to estimate its state (orientation and position), its implementation is based on [15]. More anchors provide more sensor data leading to an increased robustness of this process. This project uses 8 LPS anchors since this also allows to work with Time Difference of Arrival (TDoA) in addition to Two-Way Ranging (TWR).

Chekuri and Won showed in their tests [1] that an LPS with 6 anchors can achieve an accuracy of 9 cm when the drone is hovering, this is important as we will record the drone’s position and scan while the drone is holding its position at its

target waypoint. A reference setup for an 8 anchor setup like we’ll be using is shown in figure 4.

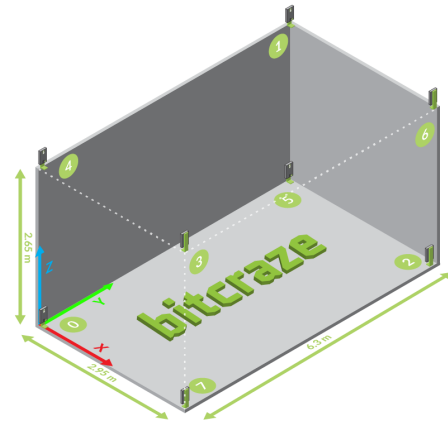


Fig. 4. 8 Anchors LPS reference setup

In addition to mounting the 8 anchors in a room they also need to be configured with their location as an $\langle x, y, z \rangle$ tuple. When the drone starts, it will connect to the anchors and download their locations for use in its estimation of its own position.

As mentioned earlier, this can be done in 2 different ways and both are supported in this project:

- **Two way ranging** mode where the drone will actively poll the anchors and calculate how far the anchor is removed by measuring the Time of Flight (ToF) of the packets it sent and their reply, then multiplying that time by the speed of light.
- **Time Difference of Arrival** mode works by measuring the difference in distance between two anchors instead of the drone and an anchor. This information is then added to the packets the anchors send. The drone (tag) listens passively to those packets and calculates its position accordingly.

Both modes have their advantages. TWR is more accurate and doesn’t require clock synchronization between the anchors, making it more robust as well. Additionally, it allows the drone to also calculate its position accurately when flying outside of the convex hull of the set of anchors.

TDOA on the other hand allows for multiple drones to fly at the same time and is more energy efficient for the drone since it’s only required to listen to (sniff) packets instead of actively communicating.

B. Wi-Fi module

An AI Thinker ESP-01 module with an Espressif Systems ESP8266 chip is used. It has 1MiB flash memory and its firmware was upgraded to the ESP8266_NonOS_AT_Bin_V1.7.4 which allows to work with more advanced AT instructions [17]. The module is soldered on a Crazyflie prototyping deck, making it possible

to let it interact with the drone as one of its expansion decks. The resulting soldered deck is shown in figure 5

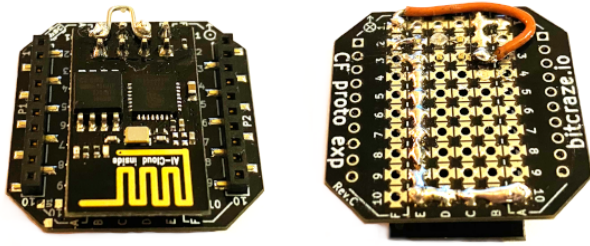


Fig. 5. An ESP-01 module soldered on a Crazyflie prototyping deck

Using the Crazyflie 2020.06 firmware release as basis, a custom driver for this ESP-01 module was written [16]. This driver communicates with the ESP-01 module over its UART interface by sending AT instructions and parsing the output. An overview of these AT instructions and their function can be found in [17]. Since the module is only used to scan for available access points, it suffices that the driver supports just the following AT instructions:

- AT - Tests AT start-up
- AT+CW_MODE_CUR - Sets the current Wi-Fi mode, used to put the module in station mode
- AT+CWLAP - Lists the available Access Points, this is the essential instruction that scans for beacons in the IEEE 802.11 2.4 GHz band. Different parameters that influence the scan can be given. For this project, some of these parameters are hard-coded making sure that all WLAN channels and all supported modes (IEEE 802.11b, 802.11g and 802.11n) are included. Other parameters are configurable and discussed in more detail below.
- AT+CWLAPOPT - Allows to format the output of the AT+CWLAP instruction, we use this to limit the output to just $\langle ssid, rssi, mac, channel \rangle$ tuples.

The driver can operate in two modes:

- **Periodic scanning:** The driver initiates a beacon scan every s seconds
- **On-demand scanning:** The driver initiates a beacon scan on-demand, triggered by setting a parameter. This is the more flexible method but requires the application that controls the drone to initiate every scan.

This operating mode together with several other options have been added to the Crazyflie parameter framework, allowing them to be changed on-the-fly by the controlling application. The following parameters are available:

- `scanOnDemand` (default = 1):
 - 0 for periodic scanning
 - 1 for on-demand scanning
- `scanInterval` (default = 5): for periodic scanning, the interval in seconds between the end of a scan and the start of the next one

- `scanNow` (default = 0): for on-demand scanning, triggers an immediate scan when set to 1, will be reset to 0 by the driver when the scan is complete
- `scanNowDelay` (default = 0): Number of seconds to wait before executing a scan triggered by `scanNow`
- `atScanType` (default = 0): the `scan_type` passed to AT+CWLAP
 - 0 for active scan
 - 1 for passive scan
- `atScanTimeMin` (default = 0) AT+CWLAP $\langle scan_time_min \rangle$, minimum scan time in ms for each channel
- `atScanTimeMax` (default = 120) AT+CWLAP $\langle scan_time_max \rangle$, maximum scan time in ms for each channel

When the scan is done, the driver will parse the results and make them immediately available to the controlling application over the Crazyradio's console port. The drone will also add its own location at the time of initiating the scan, thus providing all data for building the eventual REM.

Below an example of the output of such a scan that gets sent to the controlling application, the results have been truncated and the mac addresses obfuscated:

```
AT+CWLAP=,,,0,0,120
POS: x=1.131997 y=2.378417 z=0.744113
AP: WiFi-2.4-54AE, -80, 3835FB-----, 1
AP: telenet-FC734, -73, 5C353B-----, 1
<snip>
AP: TelenetWiFree, -81, 362CB4-----, 11
Finished scanning, waiting for next scan
request...
```

C. Communicating using the Crazyradio

Communicating with and controlling a Crazyflie remotely can be done using a custom USB dongle: the Crazyradio. This radio uses a nRF24LU1 chip providing 126 channels and works with a custom protocol to communicate with the Crazyflie: the Crazyradio RealTime Protocol (CRTP). These 126 channels [0,125] are uniformly distributed over the 2400 MHz - 2525 MHz range that the radio can operate in.

This frequency range overlaps with the IEEE 802.11 2.4 GHz band that our Wi-Fi module uses, leading to concerns over interference between the two. In order to get an idea of how pronounced this expected interference is, the Crazyradio was run on different frequencies over its range in 25 Mhz increments. (2400, 2425, 2450, 2475, 2500 and 2525 MHz) At each of these frequencies, 3 access point scans were done using the ESP-01 module on the Crazyflie. To have a baseline for comparison, an additional three scans were done with the Crazyradio turned off. The data gathered in this way is documented in appendix B.

The interference generated by the Crazyradio while scanning for access points is illustrated in figure 6. It shows for every Wi-Fi channel in the 2.4 GHz band the average count

of access points that were detected over the 3 runs and shows this for 6 different frequencies of the Crazyradio as well as the radio turned off. Wi-Fi channels that didn't have any detected access points were left out of the chart for clarity. These scans were done in a short timespan and with the Crazyflie and Crazyradio in a fixed position.

Not only does this data clearly show that the interference of the Crazyradio is significant, it also excludes possible solutions like dynamically adjusting the Crazyradio frequency to be as far as possible from the Wi-Fi channel currently being scanned. We can conclude that turning off the Crazyradio while performing an access point scan is the only viable way forward.

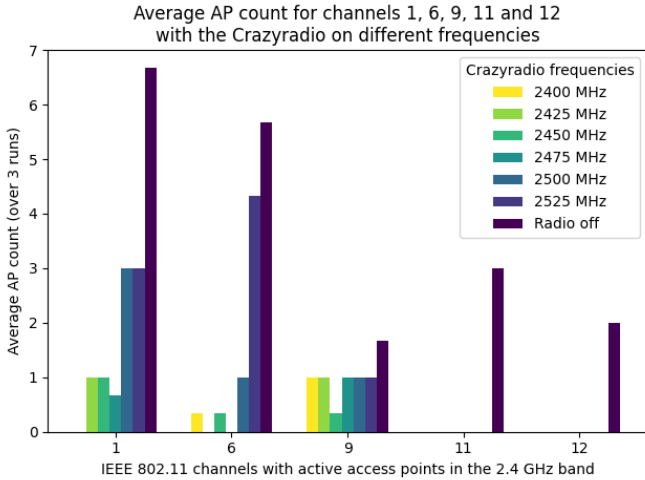


Fig. 6. The number of access points detected per IEEE 802.11 channel with the Crazyradio set at different frequencies or completely turned off. The number of access points is an average over 3 runs per frequency.

D. Controlling application

A python application that controls the drone in real-time over the Crazyradio was developed as part of this project. It uses the Crazyflie python library and is configurable with a list of $\langle x, y, z \rangle$ waypoints. It runs through the following sequence:

- 1) Initialization and instruct the drone to take-off
- 2) For every configured waypoint, make the drone:
 - Move to the waypoint
 - Initiate an on-demand scan
 - Shutdown the Crazyradio while the scan is running
 - Restart the radio connection after the scan is done
 - Fetch the scan results, parse and store them
- 3) Land the drone and shut it down
- 4) Store the results (the REM)

To avoid interference as discussed in III-C, the radio is shut down right before the scan starts and restarted again after the scan has finished. The Crazyflie will go into *position hold* mode while the radio is shut down.

Small adjustments were made to the firmware of the Crazyflie to make working with a shut down radio possible:

- the `CRTX_QUEUE_SIZE` was increased so that full scan results can be temporarily stored until the radio comes back online and the results can be sent to the controlling application
- the `COMMANDER_WDT_TIMEOUT_SHUTDOWN` was increased to 10 seconds. This timeout is a safety measure, if there is no setpoint (target $\langle x, y, z \rangle$) received within this interval, the Crazyflie will shut down under the assumption that something went wrong. The default value doesn't allow to bridge the radio shutdown period.

IV. TESTING & RESULTS

A. Design test

A short test of this design was run to demonstrate its functionality. The eight anchors were set up in a small room with the controlling application instructing the drone to visit waypoints and scan when reaching them. The results of this test can be found in appendix C.

The drone's starting position was set at $\langle 1.2, 2.75, 0.25 \rangle$ and the following waypoints were set in the controlling application. Values are in meters relative to the drone's starting position:

WP	x	y	z
0	0	0	0.5
1	-0.4	0	0.5
2	-0.4	0.4	0.5
3	0	0.4	0.5
4	0.4	0.4	0.5
5	0.4	0.0	0.5
6	0.4	-0.4	0.5
7	0	-0.4	0.5
8	-0.4	-0.4	0.5
9	0	0.5	0.5

It took the drone 100 seconds to take off, visit the 10 waypoints and perform a scan there before eventually landing again. For this test, the drone was given 5 seconds to reach a waypoint and 3 seconds to perform a beacon scan at that position while the Crazyradio was turned off.

The drone had to visit 10 waypoints where it scanned before returning to its landing position, this accounts for $(10 + 1) \times 5 + 10 \times 3 = 85$ seconds. An explanation for the remaining 15 seconds can be found with the other functions that have to be performed by the controlling application including communication, parsing and awaiting feedback from the drone.

Figure 7 shows the whole room where the test was done, intersections of the axes mark the locations of 1 of the 8 anchors. The 10 waypoints are indicated with black cross markers and the location where the drone initiated the scan with red dots.

Waypoint vs scanned locations

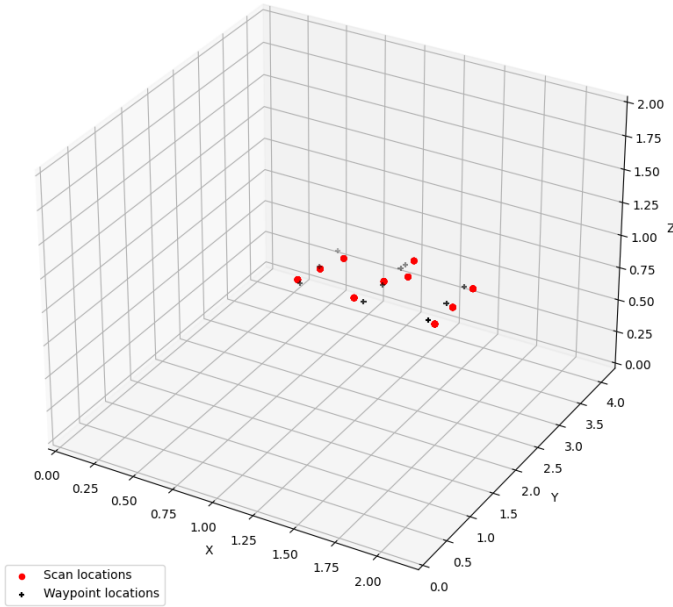


Fig. 7. Waypoint locations vs scan locations in the whole volume

This graph shows that the drone is able to match the given waypoints closely, only being off by a few centimetres. We do need to carefully interpret this though as the $\langle x, y, z \rangle$ coordinates of the scan locations are where the drone estimates its position is at, hence this is not an externally verified position.

Measured RSSI's for a single access point

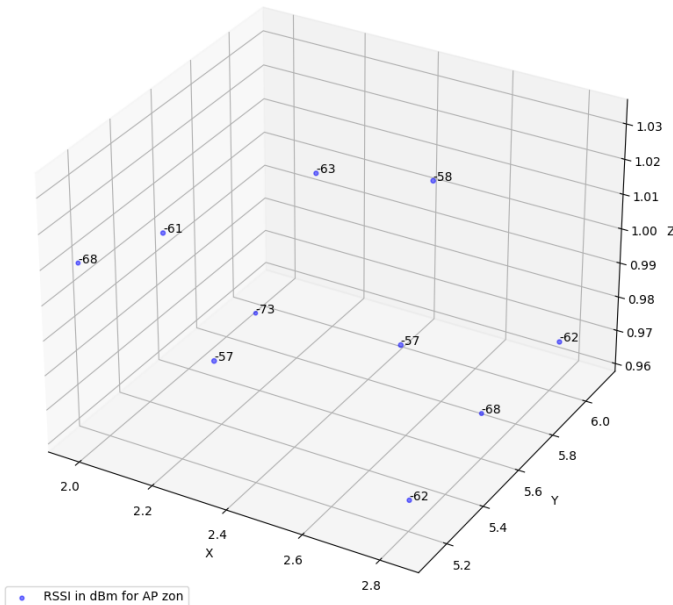


Fig. 8. RSSI Values in dBm for a single scanned access point

For an REM not only the locations are important, but also the RSSI values of the different IEEE 802.11 beacons. Figure 8 shows these RSSI values in dBm for a single access point with SSID "zon", the access point with the strongest signal in the test. Note that for visibility reasons, the graph has been zoomed in to the actual scan locations. The dots shown match the red dots of the previous image but on a different scale.

B. Endurance test

Endurance is an important factor as it dictates how many waypoints can be visited and scanned in a single run. The Crazyflie is advertised as having a flight time of up to 7 minutes depending on how you fly it. This is however without the weight and power consumed by the Loco Positioning Deck and the custom ESP8266 deck. There are also several other factors that can influence the drone's endurance including flight and scan parameters, the choice between TWR and TDoA or the distance to the anchors, to name a few.

To be able to still get an idea of the drone's endurance, it was manually flown under the following conditions until it became less responsive and its motions erratic:

- A fully charged standard battery
- 8 active anchors in TWR mode
- Periodic scanning mode with an interval of 8 seconds. With a beacon scan taking around 2 seconds, this puts the scanning frequency in line with the design test of IV-A.
- The drone was kept in a steady position about 1m above ground level for the duration of the test

Under these conditions, the drone was able to perform 35 scans over a timespan of 6 minutes and 12 seconds before it started to show erratic behaviour due to a low battery charge.

V. FUTURE WORK

In future work, post-processing the data generated by this drone could be done where the drone takes measurements at strategic points in the room which would then be enriched by using interpolation, propagation or AI based techniques, filling in the blanks and creating a dense REM.

Another interesting avenue would be to increase the endurance of the setup by for example deploying multiple drones at the same time, each covering a part of the room and aggregating the results. Other options include reducing the Crazyradio's power to the minimum required, making use of the ESP8266 module's light sleep mode and making the traveltime to a waypoint depend on its distance instead of a fixed interval.

Making use of the open and modular nature, integrating a different sensor or communication module instead of the ESP8266 would be appealing as well. Many modules exist that are small, lightweight (a few grams), work with 3.3V and provide an I²C or UART interface. Bluetooth LE or LoRa transceivers would be a possibility but also environment sensors for air quality, gas or humidity could be integrated with small adjustments to the driver and controlling application.

Finally, having an upper bound on the accuracy of the Loco Positioning System as it's used in this project with the full 8 anchor setup in both TWR and TDoA mode would be valuable. This can be achieved by for example using an additional vision-based positioning system with high millimetre-level accuracy providing the ground truth for the locations.

VI. CONCLUSION

This research project has shown how IEEE 802.11b, 802.11g and 802.11n data in the 2.4 GHz band for an indoor REM can be gathered using a small drone outfitted with an accurate decimetre-level positioning system and a custom integrated Wi-Fi module. It shows a fully autonomous solution where waypoints can be provided to an application which in turn instructs the drone in real-time to visit those waypoints, scan for IEEE 802.11 beacons, parse the results and store them on disk for later analysis and processing. Interference of the Crazyradio while scanning is avoided by doing strategic shutdowns of the radio. We have shown that this setup is able to visit dozens of waypoints in a single run.

While this project's aim was to make the autonomous gathering of data for a three-dimensional REM possible, the same design can be easily repurposed. It could for example be used for building training databases for 3D fingerprinting-based positioning, frequent scanning for rogue access points in high-security environments or detecting low Wi-Fi coverage areas.

As a result of this research project, the following paper is to be submitted for publication:

- Ken Mendes, Filip Lemic, Jeroen Famaey: "Demonstration Abstract: Enabling Autonomous and Repeatable Drone-based Experimentation in 3-Dimensional Indoor Wireless Environments", to be submitted to 18th ACM Conference on Embedded Networked Sensor Systems (SenSys 2020), 2020 [19].

ACKNOWLEDGMENT

I would like to thank my mentor for this project, Dr. Filip Lemic, for his guidance, patience, expertise and providing valuable feedback over the course of this project. I would also like to thank Prof. Dr. Jeroen Famaey for giving me the opportunity to do research in this interesting field.

Crazyflie related schematics and images (Figures 1, 2, 3 and 4) are copyright Bitcraze AB and are used according to their Creative Commons Attribution 3.0 License.

REFERENCES

- [1] A. Chekuri and M. Won, "Automating WiFi Fingerprinting Based on Nano-Scale Unmanned Aerial Vehicles," 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, 2017, pp. 1-5, doi: 10.1109/VTCspring.2017.8108561.
- [2] I. Rubin and R. Zhang, "Placement of UAVs as Communication Relays Aiding Mobile Ad Hoc Wireless Networks," MILCOM 2007 - IEEE Military Communications Conference, Orlando, FL, USA, 2007, pp. 1-7, doi: 10.1109/MILCOM.2007.4455114.
- [3] Daniel, Kai, and Christian Wietfeld. Using public network infrastructures for UAV remote sensing in civilian security operations. DORTMUND UNIV (GERMANY FR), 2011.

- [4] IEEE Std 802.11™-2016 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, pp. 692–698
- [5] S. H. Jung, B. Moon and D. Han, "Performance Evaluation of Radio Map Construction Methods for Wi-Fi Positioning Systems," in IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 4, pp. 880-889, April 2017, doi: 10.1109/TITS.2016.2594479.
- [6] Kjærgaard M.B., Blunck H., Godsk T., Toftkjær T., Christensen D.L., Grønbæk K. (2010) Indoor Positioning Using GPS Revisited. In: Floréen P., Krüger A., Spasojevic M. (eds) Pervasive Computing, Pervasive 2010. Lecture Notes in Computer Science, vol 6030. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-12654-3_3
- [7] M. A. Al-Ammar et al., "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms," 2014 International Conference on Cyberworlds, Santander, 2014, pp. 245-252, doi: 10.1109/CW.2014.41.
- [8] Song Z., Jiang G., Huang C. (2011) A Survey on Indoor Positioning Technologies. In: Zhou Q. (eds) Theoretical and Mathematical Foundations of Computer Science. ICTMF 2011. Communications in Computer and Information Science, vol 164. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24999-0_28
- [9] Bitcraze Crazyflie 2.1 <https://www.bitcraze.io/products/crazyflie-2-1/>
- [10] FreeRTOS Real-time operating system for microcontrollers <https://www.freertos.org>
- [11] Crazyflie Deck Pin Allocation https://wiki.bitcraze.io/projects/crazyflie2:expansionboards:index#deck_pin_allocation
- [12] Crazyflie Python Library <https://github.com/bitcraze/crazyflie-lib-python>
- [13] Decawave DWM1000 module <https://www.decawave.com/product/dwm1000-module/>
- [14] D. Porcino and W. Hirt, "Ultra-wideband radio technology: potential and challenges ahead," in IEEE Communications Magazine, vol. 41, no. 7, pp. 66-74, July 2003, doi: 10.1109/MCOM.2003.1215641.
- [15] M. W. Mueller, M. Hamer and R. D'Andrea, "Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 1730-1736, doi: 10.1109/ICRA.2015.7139421.
- [16] ESP-01 driver for the Crazyflie 2.1 <https://github.com/mendesk/crazyflie-firmware-esp8266>
- [17] ESP8266 Non-OS AT Instruction Set V3.0.2 by Espressif Systems https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf
- [18] Controlling application for building the REM <https://github.com/mendesk/drone-based-rem/tree/master/src/rem-builder>
- [19] SenSys <https://sensys.acm.org/>

APPENDIX A DRONE COMPARISON

Manufacturer	DJI	Parrot	Yuneec	Blitzreze
Model	Mavic 2 Zoom	Parrot Bebop 2	Mantis Q	Crazyflie
Price	€1,249	€350	€499	€220
Link	link	link	link	link
Weight	907g	500g	480g	27g (nano) max takeoff: 42g
Remote included?	Yes	No, but available	Yes	No
Connectivity	WiFi	Wi-Fi 802.11 a/b/g/n/ac	5.470-5.725MHz	low-latency/long-range radio (Crazyradio) Bluetooth LE
Radio frequency	2.400 - 2.483 GHz; 5.725 - 5.850 GHz	2.4 and 5 GHz dual dipole aerials	2.4 / 5.8 GHz	2.4 Ghz (Crazyradio)
Output Power	<= 18 dBm	Up to 21 dBm	27 dBm	
Signal range	Controller: 5km	300m	800m (EU)	1000m
Positioning	GPS / GLONASS / Vision	GPS	GPS / GLONASS / IPS (indoor PS)	
Sensors	See Vision system below	Barometer, Magnetometer, Ultrasonic, Vertical Camera, 3-axis accelerometer, 3-axis gyroscope	Ultrasonic, IPS	accelerometer, gyro, magnetometer and a high precision pressure sensor
Camera	12 MP (+ gimbal)	14mp with fisheye	4K	
Video	Full HD	3-axis full HD 1080p	4K UHD	
Battery	16 mins	25 mins	33 mins	7 mins
Processor		Dual core processor with quad-core GPU		
Storage	SD Card up to 64GB?	8 GB flash storage system	16 GB MicroSD pre-installed	
Works indoors confirmed?	Yes	Yes	Yes	Yes, made for indoors
Hovering Accuracy	Vertical: +/- 0.1 m (When Vision Positioning is active) or +/- 0.5 m Positioning is active) or +/- 1.5 m	Vertical: ±0.1 m (with Vision Positioning) ±0.5 m (with GPS Positioning) Horizontal: ±0.3 m (with Vision Positioning) ±1.5 m (with GPS Positioning)		
Flight Options	Vision system (= 3D infra + cam) allows for precision hovering indoors and collision avoidance. Intelligent flight modes: QuickShots, ActiveTrack, TapFly, TapGo, TapGo mode, Gesture mode (more of interest) Flight recorder: Flight data is automatically recorded to the internal storage of the aircraft status information and other parameters. To access this data, connect the aircraft to the PC through the Micro USB port.	Down, forward and backward vision systems allowing for precision hovering waypoint mode, 2 to 5 wps and takes intelligent flight modes: QuickShots, ActiveTrack, SmartCapture (gestures), TapFly, TapGo mode, Cinematic mode, TapFly, Flight recorder: Flight data is automatically recorded to the internal storage of the aircraft status information, and other parameters. To access this data, connect the aircraft to the PC through the Micro USB port.	- IPS, requires recognizable non-reflective, non-repeating terrain/environment. - Only with GPS signal, Journey, P.O.I, Follow-me, Watch-me - No route / waypointing	
Software		Windows SDK available	PX4 based flight controller but no API or access? Only Yuneec H520 allows SDK access but expensive (~ €1600)	Completely open and hackable, can do realtime controlling from PC with Crazyradio PA dongle
Remarks	Vision systems (indoor) require good lighting and different non-reflective patterns API / ROS only on the even more expensive models			
Payload		250 grams? (unofficial)		

APPENDIX B
CRAZYRADIO INTERFERENCE

CR Channel CR Frequency	0 2400 MHz	25 2425 MHz	50 2450 MHz	75 2475 MHz	100 2500 MHz	125 2525 MHz	No Radio
Scan 1	-65, AC2205-----, 6 -44, 04F021-----, 9	-52, 06353B-----, 1 -45, 04F021-----, 9	-46, 06353B-----, 1 -60, AC2205-----, 6	-45, 04F021-----, 9	-65, 5C353B-----, 1 -52, 06353B-----, 1 -67, 06353B-----, 1 -62, 566731-----, 6 -43, 04F021-----, 9	-64, 5C353B-----, 1 -51, 06353B-----, 1 -65, 06353B-----, 1 -54, AE2235-----, 6 -64, 566731-----, 6 -65, 566731-----, 6 -43, 04F021-----, 9	-67, 5C353B-----, 1 -83, C8D12A-----, 1 -90, 362CB4-----, 1 -50, 06353B-----, 1 -67, 06353B-----, 1 -95, 5C353B-----, 1 -71, AC2205-----, 6 -61, AC2205-----, 6 -59, AE2235-----, 6 -67, 546751-----, 6 -69, AE2235-----, 6 -50, 04F021-----, 9 -94, 342CC4-----, 1 -83, DC537C-----, 11 -78, 342CC4-----, 11 -76, 362CB4-----, 11 -85, 3835FB-----, 11 -76, 42F201-----, 12 -82, 40F201-----, 12
Scan 2	-41, 04F021-----, 9	-65, 06353B-----, 1 -47, 04F021-----, 9	-47, 06353B-----, 1	-65, 5C353B-----, 1 -42, 04F021-----, 9	-54, 06353B-----, 1 -66, 06353B-----, 1 -66, 546751-----, 6 -56, AC2205-----, 6 -70, AC2205-----, 6 -45, 04F021-----, 9	-66, 5C353B-----, 1 -50, 06353B-----, 1 -66, 546751-----, 6 -56, AC2205-----, 6 -56, AE2235-----, 6 -65, 566731-----, 6 -43, 04F021-----, 9 -67, 06353B-----, 1	-67, 5C353B-----, 1 -83, C8D12A-----, 1 -49, 06353B-----, 1 -90, 5C353B-----, 1 -68, 06353B-----, 1 -69, FA8FCA-----, 6 -64, 546751-----, 6 -57, AC2205-----, 6 -66, AC2205-----, 6 -88, AE2235-----, 6 -66, 566731-----, 6 -57, AE2235-----, 6 -90, 362CB4-----, 1 -89, 342CC4-----, 1 -46, 04F021-----, 9 -76, 342CC4-----, 11 -75, 362CB4-----, 11 -81, 3835FB-----, 11 -77, 40F201-----, 12 -75, 42F201-----, 12
Scan 3	-44, 04F021-----, 9	-49, 06353B-----, 1 -44, 04F021-----, 9	-45, 06353B-----, 1 -42, 04F021-----, 9	-65, 06353B-----, 1 -46, 04F021-----, 9	-54, 06353B-----, 1 -67, 06353B-----, 1 -67, 5C353B-----, 1 -62, AE2235-----, 6 -45, 04F021-----, 9	-64, 5C353B-----, 1 -49, 06353B-----, 1 -64, 06353B-----, 1 -55, AC2205-----, 6 -64, 546751-----, 6 -55, AE2235-----, 6 -64, 566731-----, 6 -64, FA8FCA-----, 6 -46, 04F021-----, 9	-68, 5C353B-----, 1 -84, C8D12A-----, 1 -53, 06353B-----, 1 -68, 06353B-----, 1 -90, 342CC4-----, 1 -93, 5C353B-----, 1 -66, 546751-----, 6 -86, AC2205-----, 6 -66, AE2235-----, 6 -64, 566731-----, 6 -55, AE2235-----, 6 -91, 362CB4-----, 1 -46, 04F021-----, 9 -89, 342CC4-----, 1 -81, 3835FB-----, 11 -82, 06353B-----, 1 -75, 42F201-----, 12 -76, 40F201-----, 12

APPENDIX C
DESIGN TEST RESULTS

Parsed output of the test results with the following fields

- 1) X position
- 2) Y position
- 3) Z position
- 4) SSID
- 5) RSSI value in dBm
- 6) MAC Address (obfuscated for privacy reasons)
- 7) Channel

```
(1.211822, 2.745754, 0.781816) WiFi-2.4-54AE | -77 | 3835FB----- | 1
(1.211822, 2.745754, 0.781816) telenet-FC734 | -83 | 5C353B----- | 1
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -81 | 06353B----- | 1
(1.211822, 2.745754, 0.781816) The Cara Loft | -81 | AC2205----- | 6
(1.211822, 2.745754, 0.781816) Glitzers | -73 | 546751----- | 6
(1.211822, 2.745754, 0.781816) telenet-B29F5F9 | -73 | AC2205----- | 6
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -81 | AE2235----- | 6
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -75 | AE2235----- | 6
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -75 | 566731----- | 6
(1.211822, 2.745754, 0.781816) zon | -63 | 04F021----- | 9
(1.211822, 2.745754, 0.781816) telenet-63B5A | -70 | 342CC4----- | 11
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -70 | 362CB4----- | 11
(1.211822, 2.745754, 0.781816) Orange-e957f | -75 | D8FB5E----- | 11
(1.211822, 2.745754, 0.781816) Guest-Orange-e957f | -76 | DAFB5E----- | 11
(1.211822, 2.745754, 0.781816) TelenetWiFiFree | -84 | 362CB4----- | 11
(0.820605, 2.701933, 0.755447) WiFi-2.4-54AE | -73 | 3835FB----- | 1
(0.820605, 2.701933, 0.755447) telenet-FC734 | -75 | 5C353B----- | 1
(0.820605, 2.701933, 0.755447) WiFi-2.4-CA53 | -82 | C4EA1D----- | 1
(0.820605, 2.701933, 0.755447) Glitzers | -71 | 546751----- | 6
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -72 | 566731----- | 6
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -73 | AE2235----- | 6
(0.820605, 2.701933, 0.755447) telenet-B29F5F9 | -77 | AC2205----- | 6
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -78 | AE2235----- | 6
(0.820605, 2.701933, 0.755447) The Cara Loft | -76 | AC2205----- | 6
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -72 | 06353B----- | 1
(0.820605, 2.701933, 0.755447) zon | -61 | 04F021----- | 9
(0.820605, 2.701933, 0.755447) telenet-63B5A | -72 | 342CC4----- | 11
(0.820605, 2.701933, 0.755447) Orange-e957f | -68 | D8FB5E----- | 11
(0.820605, 2.701933, 0.755447) Guest-Orange-e957f | -69 | DAFB5E----- | 11
(0.820605, 2.701933, 0.755447) telenet-74643 | -79 | 342CC4----- | 11
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -78 | 362CB4----- | 11
(0.820605, 2.701933, 0.755447) TelenetWiFiFree | -79 | 362CB4----- | 11
(0.844855, 3.125482, 0.712712) WiFi-2.4-54AE | -77 | 3835FB----- | 1
(0.844855, 3.125482, 0.712712) telenet-FC734 | -69 | 5C353B----- | 1
(0.844855, 3.125482, 0.712712) WiFi-2.4-CA53 | -88 | C4EA1D----- | 1
(0.844855, 3.125482, 0.712712) TelenetWiFiFree | -68 | 06353B----- | 1
(0.844855, 3.125482, 0.712712) Glitzers | -80 | 546751----- | 6
(0.844855, 3.125482, 0.712712) TelenetWiFiFree | -79 | 566731----- | 6
(0.844855, 3.125482, 0.712712) telenet-B29F5F9 | -76 | AC2205----- | 6
(0.844855, 3.125482, 0.712712) TelenetWiFiFree | -76 | AE2235----- | 6
(0.844855, 3.125482, 0.712712) The Cara Loft | -74 | AC2205----- | 6
(0.844855, 3.125482, 0.712712) TelenetWiFiFree | -75 | AE2235----- | 6
(0.844855, 3.125482, 0.712712) zon | -73 | 04F021----- | 9
(0.844855, 3.125482, 0.712712) Orange-e957f | -76 | D8FB5E----- | 11
(0.844855, 3.125482, 0.712712) Guest-Orange-e957f | -73 | DAFB5E----- | 11
(0.844855, 3.125482, 0.712712) telenet-63B5A | -70 | 342CC4----- | 11
(0.844855, 3.125482, 0.712712) TelenetWiFiFree | -70 | 362CB4----- | 11
(1.25751, 3.115851, 0.716725) WiFi-2.4-54AE | -74 | 3835FB----- | 1
(1.25751, 3.115851, 0.716725) telenet-FC734 | -69 | 5C353B----- | 1
(1.25751, 3.115851, 0.716725) TelenetWiFiFree | -70 | 06353B----- | 1
(1.25751, 3.115851, 0.716725) Glitzers | -79 | 546751----- | 6
(1.25751, 3.115851, 0.716725) TelenetWiFiFree | -72 | AE2235----- | 6
(1.25751, 3.115851, 0.716725) The Cara Loft | -73 | AC2205----- | 6
(1.25751, 3.115851, 0.716725) telenet-B29F5F9 | -78 | AC2205----- | 6
```

(1.25751, 3.115851, 0.716725) TelenetWiFree | -75 | 566731----- | 6
(1.25751, 3.115851, 0.716725) TelenetWiFree | -77 | AE2235----- | 6
(1.25751, 3.115851, 0.716725) zon | -57 | 04F021----- | 9
(1.25751, 3.115851, 0.716725) telenet-63B5A | -72 | 342CC4----- | 11
(1.25751, 3.115851, 0.716725) TelenetWiFree | -77 | 362CB4----- | 11
(1.25751, 3.115851, 0.716725) TelenetWiFree | -71 | 362CB4----- | 11
(1.25751, 3.115851, 0.716725) Orange-e957f | -73 | D8FB5E----- | 11
(1.25751, 3.115851, 0.716725) Guest-Orange-e957f | -71 | DAFB5E----- | 11
(1.25751, 3.115851, 0.716725) telenet-74643 | -78 | 342CC4----- | 11
(1.635196, 3.220749, 0.724767) WiFi-2.4-54AE | -73 | 3835FB----- | 1
(1.635196, 3.220749, 0.724767) telenet-FC734 | -72 | 5C353B----- | 1
(1.635196, 3.220749, 0.724767) TelenetWiFree | -72 | 06353B----- | 1
(1.635196, 3.220749, 0.724767) Glitzers | -76 | 546751----- | 6
(1.635196, 3.220749, 0.724767) TelenetWiFree | -79 | AE2235----- | 6
(1.635196, 3.220749, 0.724767) TelenetWiFree | -77 | 566731----- | 6
(1.635196, 3.220749, 0.724767) TelenetWiFree | -78 | AE2235----- | 6
(1.635196, 3.220749, 0.724767) The Cara Loft | -77 | AC2205----- | 6
(1.635196, 3.220749, 0.724767) zon | -62 | 04F021----- | 9
(1.635196, 3.220749, 0.724767) Orange-e957f | -74 | D8FB5E----- | 11
(1.635196, 3.220749, 0.724767) Guest-Orange-e957f | -74 | DAFB5E----- | 11
(1.635196, 3.220749, 0.724767) telenet-63B5A | -68 | 342CC4----- | 11
(1.635196, 3.220749, 0.724767) telenet-74643 | -82 | 342CC4----- | 11
(1.635196, 3.220749, 0.724767) TelenetWiFree | -68 | 362CB4----- | 11
(1.635196, 3.220749, 0.724767) TelenetWiFree | -83 | 362CB4----- | 11
(1.635549, 2.767902, 0.726354) WiFi-2.4-54AE | -70 | 3835FB----- | 1
(1.635549, 2.767902, 0.726354) telenet-FC734 | -72 | 5C353B----- | 1
(1.635549, 2.767902, 0.726354) TelenetWiFree | -71 | 06353B----- | 1
(1.635549, 2.767902, 0.726354) WiFi-2.4-CA53 | -86 | C4EA1D----- | 1
(1.635549, 2.767902, 0.726354) Glitzers | -72 | 546751----- | 6
(1.635549, 2.767902, 0.726354) telenet-B29F5F9 | -75 | AC2205----- | 6
(1.635549, 2.767902, 0.726354) TelenetWiFree | -76 | 566731----- | 6
(1.635549, 2.767902, 0.726354) TelenetWiFree | -74 | AE2235----- | 6
(1.635549, 2.767902, 0.726354) TelenetWiFree | -72 | AE2235----- | 6
(1.635549, 2.767902, 0.726354) zon | -68 | 04F021----- | 9
(1.635549, 2.767902, 0.726354) telenet-63B5A | -61 | 342CC4----- | 11
(1.635549, 2.767902, 0.726354) Orange-e957f | -75 | D8FB5E----- | 11
(1.635549, 2.767902, 0.726354) TelenetWiFree | -62 | 362CB4----- | 11
(1.635549, 2.767902, 0.726354) Guest-Orange-e957f | -74 | DAFB5E----- | 11
(1.635549, 2.767902, 0.726354) telenet-74643 | -73 | 342CC4----- | 11
(1.635549, 2.767902, 0.726354) TelenetWiFree | -77 | 362CB4----- | 11
(1.632986, 2.383654, 0.721445) WiFi-2.4-54AE | -79 | 3835FB----- | 1
(1.632986, 2.383654, 0.721445) telenet-FC734 | -70 | 5C353B----- | 1
(1.632986, 2.383654, 0.721445) WiFi-2.4-CA53 | -81 | C4EA1D----- | 1
(1.632986, 2.383654, 0.721445) TelenetWiFree | -69 | 06353B----- | 1
(1.632986, 2.383654, 0.721445) Glitzers | -75 | 546751----- | 6
(1.632986, 2.383654, 0.721445) telenet-B29F5F9 | -80 | AC2205----- | 6
(1.632986, 2.383654, 0.721445) The Cara Loft | -72 | AC2205----- | 6
(1.632986, 2.383654, 0.721445) TelenetWiFree | -83 | AE2235----- | 6
(1.632986, 2.383654, 0.721445) TelenetWiFree | -73 | AE2235----- | 6
(1.632986, 2.383654, 0.721445) TelenetWiFree | -74 | 566731----- | 6
(1.632986, 2.383654, 0.721445) telenet-Sam en Ana | -80 | 546751----- | 6
(1.632986, 2.383654, 0.721445) zon | -62 | 04F021----- | 9
(1.632986, 2.383654, 0.721445) telenet-63B5A | -63 | 342CC4----- | 11
(1.632986, 2.383654, 0.721445) telenet-74643 | -84 | 342CC4----- | 11
(1.632986, 2.383654, 0.721445) Orange-e957f | -75 | D8FB5E----- | 11
(1.632986, 2.383654, 0.721445) TelenetWiFree | -63 | 362CB4----- | 11
(1.632986, 2.383654, 0.721445) TelenetWiFree | -84 | 362CB4----- | 11
(1.632986, 2.383654, 0.721445) Guest-Orange-e957f | -77 | DAFB5E----- | 11
(1.131997, 2.378417, 0.744113) WiFi-2.4-54AE | -80 | 3835FB----- | 1
(1.131997, 2.378417, 0.744113) telenet-FC734 | -73 | 5C353B----- | 1
(1.131997, 2.378417, 0.744113) TelenetWiFree | -68 | 06353B----- | 1
(1.131997, 2.378417, 0.744113) Glitzers | -73 | 546751----- | 6
(1.131997, 2.378417, 0.744113) The Cara Loft | -71 | AC2205----- | 6
(1.131997, 2.378417, 0.744113) TelenetWiFree | -73 | 566731----- | 6
(1.131997, 2.378417, 0.744113) TelenetWiFree | -71 | AE2235----- | 6
(1.131997, 2.378417, 0.744113) TelenetWiFree | -78 | AE2235----- | 6
(1.131997, 2.378417, 0.744113) telenet-B29F5F9 | -76 | AC2205----- | 6

(1.131997, 2.378417, 0.744113) zon | -57 | 04F021----- | 9
(1.131997, 2.378417, 0.744113) telenet-63B5A | -60 | 342CC4----- | 11
(1.131997, 2.378417, 0.744113) telenet-74643 | -80 | 342CC4----- | 11
(1.131997, 2.378417, 0.744113) Orange-e957f | -79 | D8FB5E----- | 11
(1.131997, 2.378417, 0.744113) TelenetWiFiFree | -61 | 362CB4----- | 11
(1.131997, 2.378417, 0.744113) Guest-Orange-e957f | -79 | DAFB5E----- | 11
(1.131997, 2.378417, 0.744113) TelenetWiFiFree | -81 | 362CB4----- | 11
(0.776738, 2.371472, 0.760733) WiFi-2.4-54AE | -86 | 3835FB----- | 1
(0.776738, 2.371472, 0.760733) telenet-FC734 | -78 | 5C353B----- | 1
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -75 | 06353B----- | 1
(0.776738, 2.371472, 0.760733) WiFi-2.4-CA53 | -87 | C4EA1D----- | 1
(0.776738, 2.371472, 0.760733) Glitzers | -73 | 546751----- | 6
(0.776738, 2.371472, 0.760733) The Cara Loft | -68 | AC2205----- | 6
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -68 | AE2235----- | 6
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -74 | 566731----- | 6
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -74 | AE2235----- | 6
(0.776738, 2.371472, 0.760733) telenet-B29F5F9 | -74 | AC2205----- | 6
(0.776738, 2.371472, 0.760733) zon | -68 | 04F021----- | 9
(0.776738, 2.371472, 0.760733) telenet-63B5A | -61 | 342CC4----- | 11
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -82 | 362CB4----- | 11
(0.776738, 2.371472, 0.760733) telenet-74643 | -83 | 342CC4----- | 11
(0.776738, 2.371472, 0.760733) TelenetWiFiFree | -61 | 362CB4----- | 11
(1.225038, 3.359706, 0.753282) telenet-FC734 | -75 | 5C353B----- | 1
(1.225038, 3.359706, 0.753282) WiFi-2.4-54AE | -83 | 3835FB----- | 1
(1.225038, 3.359706, 0.753282) TelenetWiFiFree | -73 | 06353B----- | 1
(1.225038, 3.359706, 0.753282) The Cara Loft | -74 | AC2205----- | 6
(1.225038, 3.359706, 0.753282) TelenetWiFiFree | -72 | AE2235----- | 6
(1.225038, 3.359706, 0.753282) Glitzers | -71 | 546751----- | 6
(1.225038, 3.359706, 0.753282) TelenetWiFiFree | -71 | 566731----- | 6
(1.225038, 3.359706, 0.753282) zon | -58 | 04F021----- | 9
(1.225038, 3.359706, 0.753282) telenet-74643 | -77 | 342CC4----- | 11
(1.225038, 3.359706, 0.753282) telenet-63B5A | -74 | 342CC4----- | 11
(1.225038, 3.359706, 0.753282) TelenetWiFiFree | -75 | 362CB4----- | 11
(1.225038, 3.359706, 0.753282) TelenetWiFiFree | -79 | 362CB4----- | 11
(1.225038, 3.359706, 0.753282) Orange-e957f | -72 | D8FB5E----- | 11
(1.225038, 3.359706, 0.753282) Guest-Orange-e957f | -73 | DAFB5E----- | 11