Universiteit
Antwerpen

DIFFERENT MICROCONTROLLERS

De Borrekens Gauthier

Intro

Wireless microcontrollers are divided in groups based on which protocol they use.
There are plenty of different protocols based upon the RF (Radio Frequency) spectrum.
In the next paragraph we will discuss the (dis)advantages of the most important ones. Thereafter we will decide which one we will be using and why. Finally we compare the different MCU's using this protocol and pick one out.

Materials and Methods

As this is just research, no materials were used. The sources will be added.

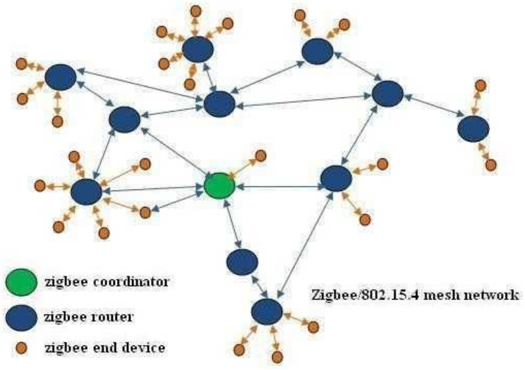
Information and conclusion

Wi-Fi

Wi-fi, or the IEEE 802.11 Protocol has a very high data rate and low delay, but comes with the downside of a high power consumption.
It is mostly used to connect devices to an IoT-setup which do not need to run for a long time on some batteries.
Wi-Fi is very often used which makes it easy to work with. Almost all devices can connect to it and it suits most needs.
Specifications:
Frequency: 2.4 GHz, 5.8 GHz
Data rate: 100-200 Mbps realistically (can go up to 1Gbps)
Range: 50 meters
Power consumption: High

Zigbee (802.15.4 Standard)

Zigbee is a mesh networking protocol that is based upon the 802.15.4 standard. The IEEE 802.15.4 standard defines the WPAN network. A mesh network is a network where points can communicate to and through each other, making a net or a 'mesh'.
A device that can use the Zigbee protocol is XBee.
Specifications:
Frequency: 2.4GHz, others possible
Data rate: 250Kbps
Range: 10 meters (can go up to 100m)
Power consumption: Low
[1] [2]



Bluetooth

Bluetooth is a short-range technology which Is gaining popularity lately. It is mostly used to connect smartphones to computers or to one another. Bluetooth is not designed for data transfer and more for connectivity. Specifications:
Frequency: 2.4 GHz
Data rate: 1 Mbps
Range: 10 meters
Power consumption: Medium

[3] [2] [4]

Conclusion

It is obvious that Zigbee and Bluetooth do not fit our need as they are both short ranged and have relatively low data transfer. The highest priority are high throughput and low delays as the lack of these can cause life-threatening crashes. Power consumption is not as important as it is neglectable in the scope of a car.

In this project we will be using Wi-Fi, as the power consumption is not a big deal. Of course this only counts for the miniature version we will be making as the project in the real world will extend over a really large area. In this case it is probably recommended to use cellular connection and let a satellite or a big server act as a 'hub' for a big area.

It is also possible that each car will be his own 'hub' and they communicate only between each other. This will all be discussed later on when we try to apply this project in the real world.

Recap: This setup will feature an RC car with a MCU connected on each one which will establish a Wi-Fi connection to a server running on a laptop.

Which MCU to use

Now that we narrowed our choice down to Wi-Fi-based MCU's, lets see which one.

The most commonly used MCU's for a Wi-Fi are Arduino, Raspberry pi and a MCU using the ESP8266.

Arduino & Raspberry Pi

These two MCU's are by far the best known and rings a bell with most people interested in electronics. This is because they are mainly an all-round MCU that is easy to learn and can do a variety of projects. They are often suggested to people interested in starting out with some electronic tinkering.

The downside of those two MCU's is that they are relatively expensive, especially when you need a feature not implemented in the base board. You'll have to buy these modules separately and they add up to the total cost. Both of them do not contain a Wi-Fi component; Arduino needs a 'Wi-Fi Shield' which easily cost up to 20 euros and, ironically, uses the ESP8266 component.

Raspberry pi needs a Wi-Fi adapter which cost 8 euros.

Both of these are already more expensive than MCU's that use ESP8266. [5] This brings us to the alternative for Wi-Fi connection, ESP8266.

ESP8266

By far the biggest and most used Wi-Fi component is the ESP8266, a Wi-Fi component made by 'Espressif', an electronic company based in Shanghai. Its successor, ESP32, has a lot more features.

Finally, 'Microchip' also makes a variety of Wi-Fi components.

The biggest advantage and the greatest factor in the decision is the amount of information found online. Here, ESP8266 clearly excels as there are so many tutorials and user-made modules plus the fact that everything is open source makes this an excellent choice for us.

The Wi-Fi-component is used in a variety of different boards created by different manufacturers like Adafruit, Sparkfun and NodeMCU. The component is also used on the Arduino-shield that adds Wi-Fi to the Arduino but as explained in the previous paragraph we won't be using this.

A quick comparison of the different boards:

- Adafruit: Great documentation and without bugs, but on the pricey side. Uses the Arduino IDE
- Sparkfun: Even pricier, also uses Arduino IDE.
- NodeMCU: Very cheap, fully open source. Uses the 'lua' language and runs the 'NodeMCU' firmware. Can also run other firmwares like the Arduino IDE and microPython.
- Wemos: Wemos D1 is a small and cheaper variant of the ESP8266. It contains only 16 total pins in respect to the 30 of the NodeMCU. It's also only available in china which takes too long to arrive.

[6] [7]

We will be using the NodeMCU for this project as it is the cheapest option which still does everything we want. The only choice left to make is the difference in the three versions, gen1 gen2 and gen3.

- Gen1: (V1 / v0.9) Original but outdated, uses ESP12 module and has 4MB flash memory
- Gen2: (V2 / v1.0) Fixed sizing to fit on a breadboard. The most commonly used.
- Gen3: (V3) Is wider and very inconvenient to use, apparently a spinoff as it isn't made by the NodeMCU team

[8]

Conclusion

We will be working with the ESP8266 Wi-Fi module placed on a NodeMCU board. The nodeMCU board will be explained in another document.

BIBLIOGRAPHY

[1] RF Wireless world, „Advantages and disadvantages of Zigbee," RF Wireless World, [Online]. Available: http://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-zigbee.html. [Geopend 7 Maart 2018].

[2] RS online, „eleven IoT protocols you need to know about," RS Online, [Online]. Available: https://www.rsonline.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about. [Geopend 7 Maart 2018].

[3] Ali Mostafa, „Understanding differences between Bluetooth, Zigbee and Wi-Fi," SlideShare, [Online]. Available: https://www.slideshare.net/MostafaAli39/understanding-differences-between-bluetooth-zigbee-and-Wi-Fi. [Geopend 7 Maart 2018].

[4] geometrikal, „Whats the difference between RF options (Wi-Fi, xbee, NRF24L01)," Stackexchange, [Online]. Available: https://arduino.stackexchange.com/questions/3270/whats-the-difference-between-rf-options-Wi-Fi-xbee-nrf24l01. [Geopend 7 Maart 2018].

[5] makeuseof, „5 dollar microcontrollers: Arduino, raspberry pi or nodeMCU?," makeuseof, [Online]. Available: https://www.makeuseof.com/tag/5-microcontrollers-arduino-raspberry-pi-zero-nodemcu/. [Geopend 13 Maart 2018].

[6] squix, „esp8266 module comparison," squix, [Online]. Available: https://blog.squix.org/2015/03/esp8266-modulecomparison-esp-01-esp-05.html. [Geopend 13 Maart 2018].

[7] retro moe, „nodemcu vs adafruit vs sparkfun," retro moe, [Online]. Available: https://retro.moe/2016/03/27/c64-remotecontroller-nodemcu-vs-adafruit-huzzah-vs-sparkfun-thing-vs/. [Geopend 13 Maart 2018].

[8] frightanic, „Comparison of ESP8266 nodemcu development boards," frightanic, [Online]. Available: https://frightanic.com/iot/comparison-of-esp8266-nodemcu-development-boards/. [Geopend 13 Maart 2018].

[9] Github, „NodeMCU," Github, [Online]. Available: https://github.com/nodemcu/. [Geopend 7 Maart 2018].

[10] ReadTheDocs, „NodeMCU Documentation," Read The Docs, [Online]. Available: https://nodemcu.readthedocs.io/en/master/. [Geopend 7 Maart 2018].