readme.md

Assignment 3: Channel Sensing

Made by: Morten Sahlertz, Trung Thai, Andreas Arberg and Jacob Kjærager

Introduction

This is the handin for the third task of the course WSN conducted in the Spring of 2021 at the University of Aarhus. The project is conducted on the embedded HW-platform TelosB that runs the Contiki OS. The TelosB mote got a IEEE 802.15.4 interface which uses the 2.4 GHz ISM band. This assignment seeks to measure the noise/interference at each of 16 channels(11-26).

Technologies

To this assignment two different technologies has been used. The RSSI on the individual channels is found with the help of the Contiki CC2420 C-library on the mote. The second technology is the data visualization, which is done in Python

Data visualization

To compare the 16 RSSI signals a Python based data visualization implementation has been made. This will display the results in a generated .html-file with the help from the libary Plotly. The html-file can be seen in the root folder.

Experiments

For the experiments a function is made, which adds the RSSI readings from a designated channel. For this experiment 300 samples has been recorded for each channel. The mean value of the samples has been calculated for each channel. To find the RSSI from the CC2420(The RF chip on the mote) register, the following formula is needed according to the datasheet.

$$P = RSSI\_VAL + RSSI\_OFFSET \text{ [dBm]}$$

The function used in the assignment takes this into account and can be seen in the following code snippet:

```
int
cc2420_rssi(void)
{
  int rssi;
  int radio_was_off = 0;

  if(locked) {
    return 0;
  }

  GET_LOCK();

  if(!receive_on) {
    radio_was_off = 1;
    cc2420_on();
  }
  wait_for_status(BV(CC2420_RSSI_VALID));

  rssi = (int)((signed char) getreg(CC2420_RSSI));
  rssi += RSSI_OFFSET;

  if(radio_was_off) {
    cc2420_off();
  }
  RELEASE_LOCK();
  return rssi;
}
```

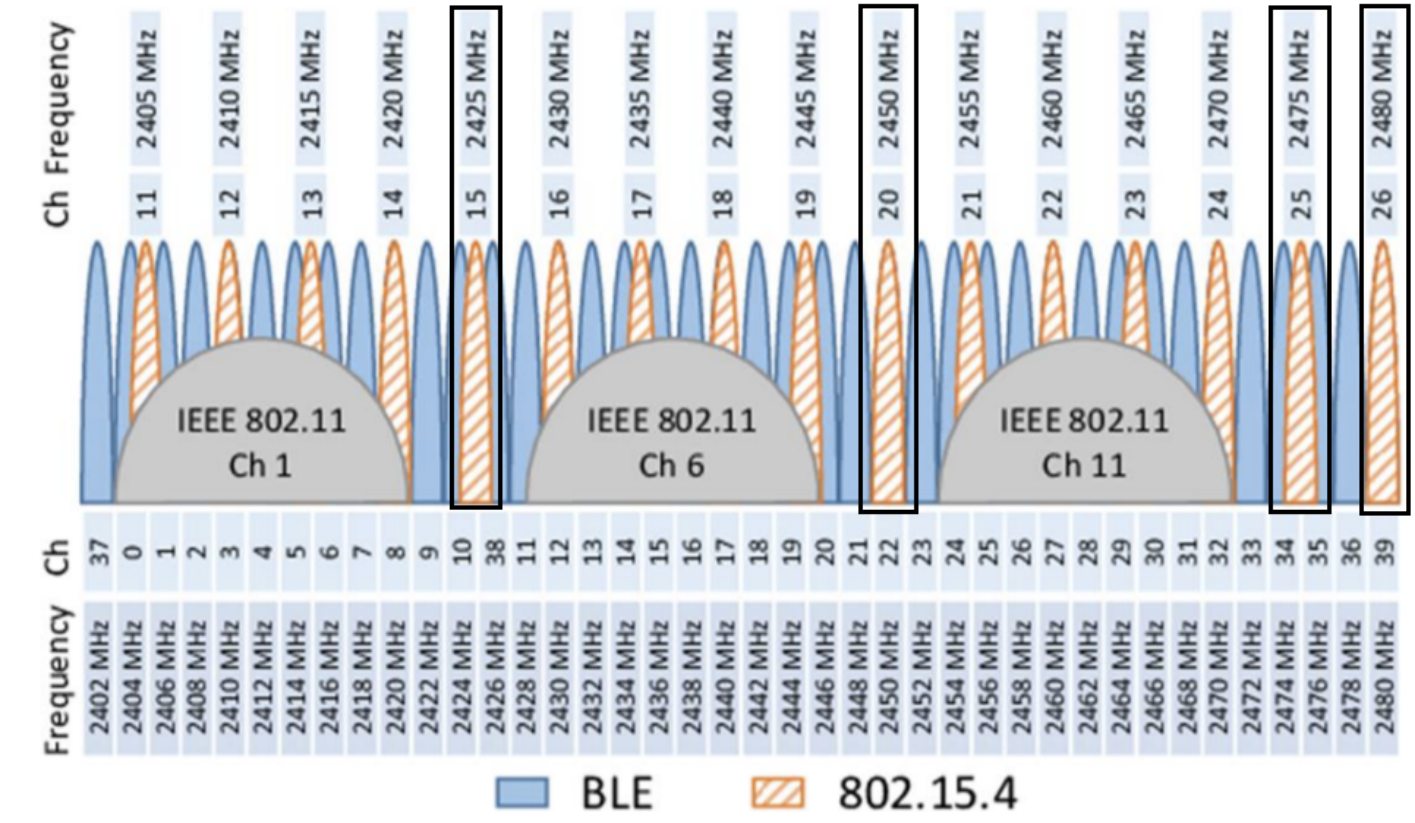
And the part shown below, is the part that takes the formula into account, so we get a correct RSSI value

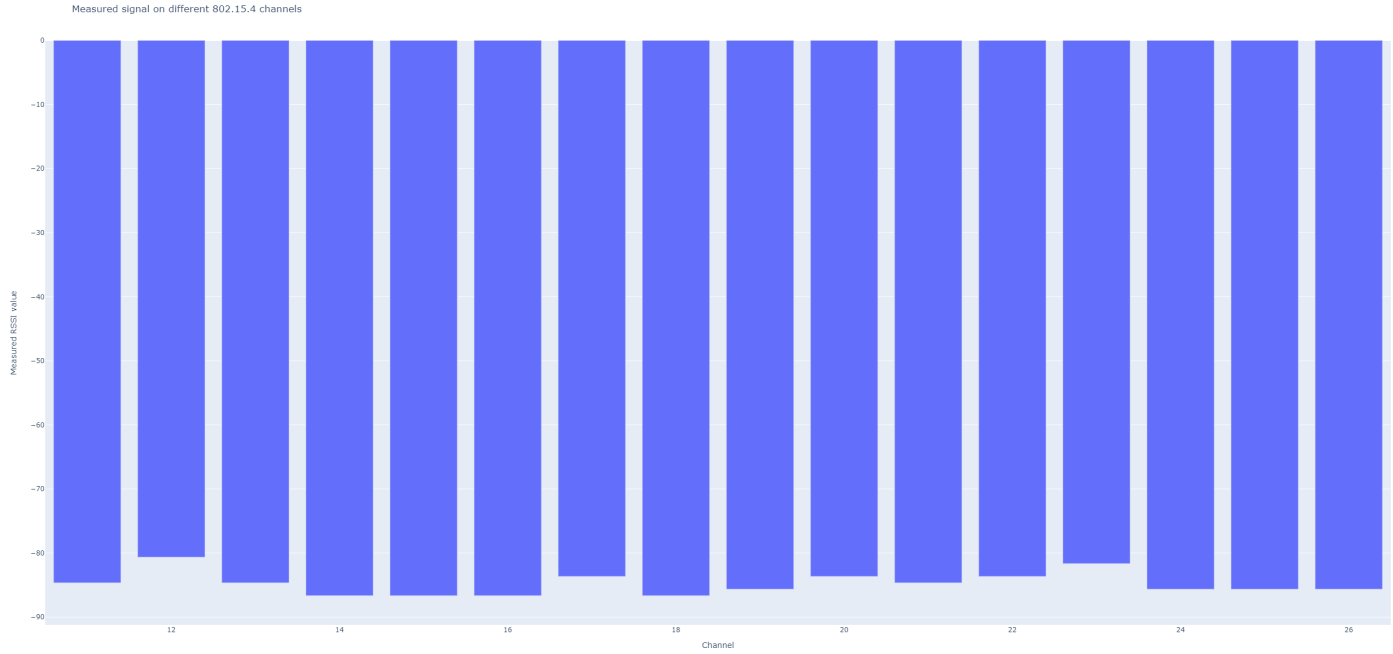
```
rssi = (int)((signed char) getreg(CC2420_RSSI));
rssi += RSSI_OFFSET;
```

This is done from channel 11 to channel 26 and the results will then be written down in a CSV-file. The results are then visualized in python and compared to the illustration shown in the assignment description.

Results

Comparing the two images we notice the theory matches the practice quite good. The best channels must be the bars with the lowest value(largest bar) as this gives the lowest noise energy in the channel. The expected best channels is marked on the figure below. The reason this is belived to be the best is because of no overlapping with the very powerful WiFi-channels which is belived to be a large source of interference, especially channel 15 and 25 are expected to be low on noise as they do not fully overlap with the BLE channels either. As seen when comparing the results with the theory it matches quite good. The values' interference are small at the channels: 12, 15, 16, 19, 25, 26. Which is close to the expected set of 15, 20, 25, 26 from the theory.





Conclusion

During this experiment we can conclude that the theory matches the obtained results quite good. The reason they don't match perfectly, is believed to be caused by different sources of errors as the small sample size and that it is not guaranteed that the WiFi/BLE-channels are in use at the time of the experiment. These type of experiments could be way better to conduct at e.g an Airport/international conference where the WiFi are heavily in use.