

Звіт

1. Датчик для визначення зайнятості паркомісця:

Вирішив обрати поки HMC5883 based QMC5883L I2C GY-273 (чи якийсь IMU модуль з компасом, де також є цей чіп, за потреби потім можна дозамовити). Причини:

- Досить дешеві (50-100 грн кожен модуль (1.5 доларів на аліекспресі)).
- Можуть працювати від 2V16 до 3V6 (що дозволить дещо довший час роботи з батареєю "на межі")
- $\pm 2\text{mG} \sim \pm 8\text{G}$ (чого буде більше ніж досить, адже середня значення для магнітного поля землі: 0.6 gauss)
- Низьке енергоспоживання (100 μA за активного використання та 2 μA у режимі Standby (тобто можливість реалізувати Deep Sleep))
- У нас на них роблять лаби, тому досить багато має бути в наявності

За потреби, у Adafruit є MLX90393, які мають "large dynamic range of **5-50 mT** (1 mT or millitesla = 10 G or Gauss). By comparison, the LSM303DLHC saturates at +/-8.1 G (0.81 mT) at maximum range setting. Втім вони дорожчі, і дещо наадлишкові для наших цілей.

Трохи моїх нотаток по сенсорах:

Magnetometers, HMC5883 included, measure the **magnetic field strength**. These magnetometer chips can measure field strengths from **2mGauss** ($1.0 \times 10^{-7}\text{T}$) to **8Gauss**. Since the average field strength of the Earth's magnetic field is between **30uT** (0.3 Gauss, $3 \times 10^{-5}\text{T}$) and **60uT** (0.6G) These chips easily measure the Earth's magnetic field strength, as well as magnetic fields from higher strength magnetic materials. Then, if one wants to calculate the heading (use magnetometer as compass), he/she should use the following formula: **D =**
 $\arctan(y\text{GaussData}/x\text{GaussData}) * (180/\pi)$.

According to the Adafruit's Library realisation, the sensor's measurement will consist of:

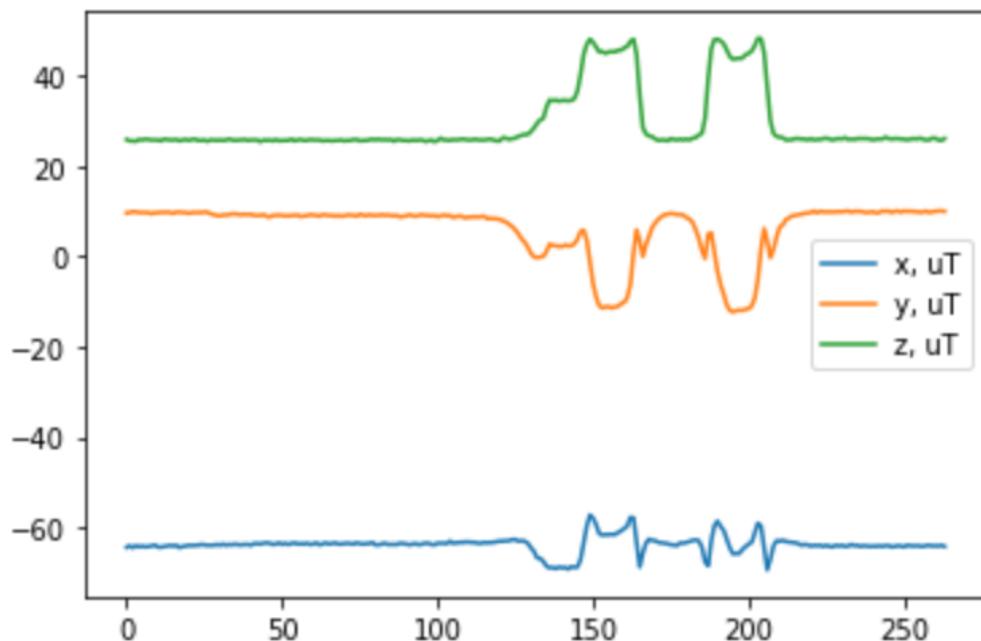
```

_magData.x / _hmc5883_Gauss LSB_XY * SENSORS_GAUSS_TO_MICROTESLA;
event->magnetic.y =
    _magData.y / _hmc5883_Gauss LSB_XY * SENSORS_GAUSS_TO_MICROTESLA;
event->magnetic.z =
    _magData.z / _hmc5883_Gauss LSB_Z * SENSORS_GAUSS_TO_MICROTESLA;

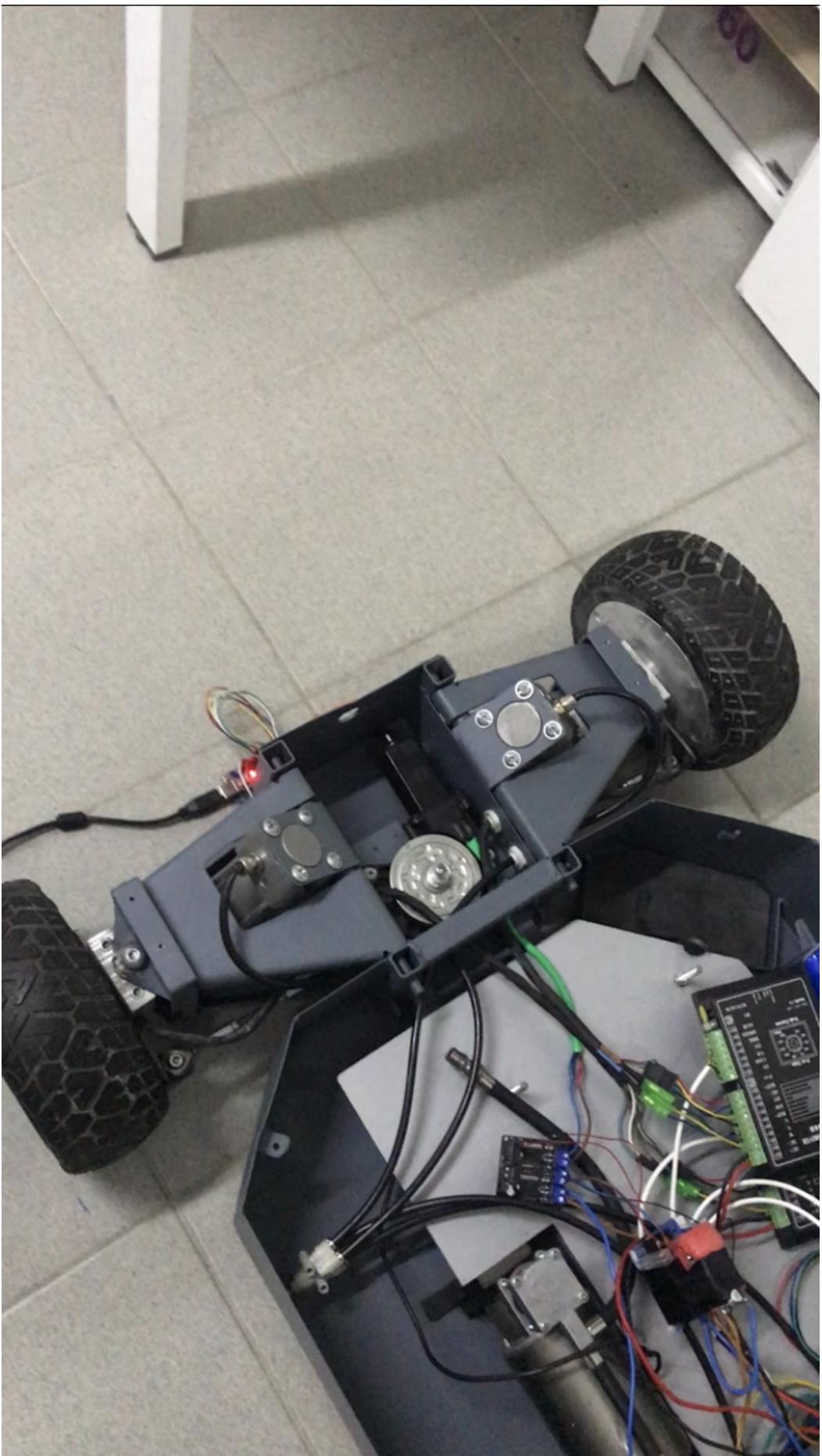
```

, where magData.x/y/z is a raw data from a sensor, _hmc5883_Gauss LSB_XY - default gain, which is equal to 1100.0F LSB/gauss (where LSB stands for Least Significant Bits and the 1100 LSB / Gauss number indicates that a reading of 1100 is equivalent to 1 Gauss, a reading of 2200 is equivalent to 2 Gauss and so on). This gain is provided in the datasheet, by the manufacturer

Щодо результативності: на цьому графіку можна бачити два послідовні наїзди машиною з бляхи на датчик (наїхав - від'їхав - наїхав (вісь x була направлена в сторону до машинки)).



Виглядало це якось так:





Враховуючи ще початкову валідацію ідеї з магнетометром (коли збирав якісь дані на парковці), можу сказати, що при приближенні машинки та великої машини - дані ведуть себе однаковим чином (машина дещо дальше до сенсора, втім і значно більша). З машинкою теж проводились заміри, коли вона поряд і показники датчиків вели себе таким ж чином.

В одному з досліджень знайшов таку симуляцію (за принципом відносної магнітної проникності):

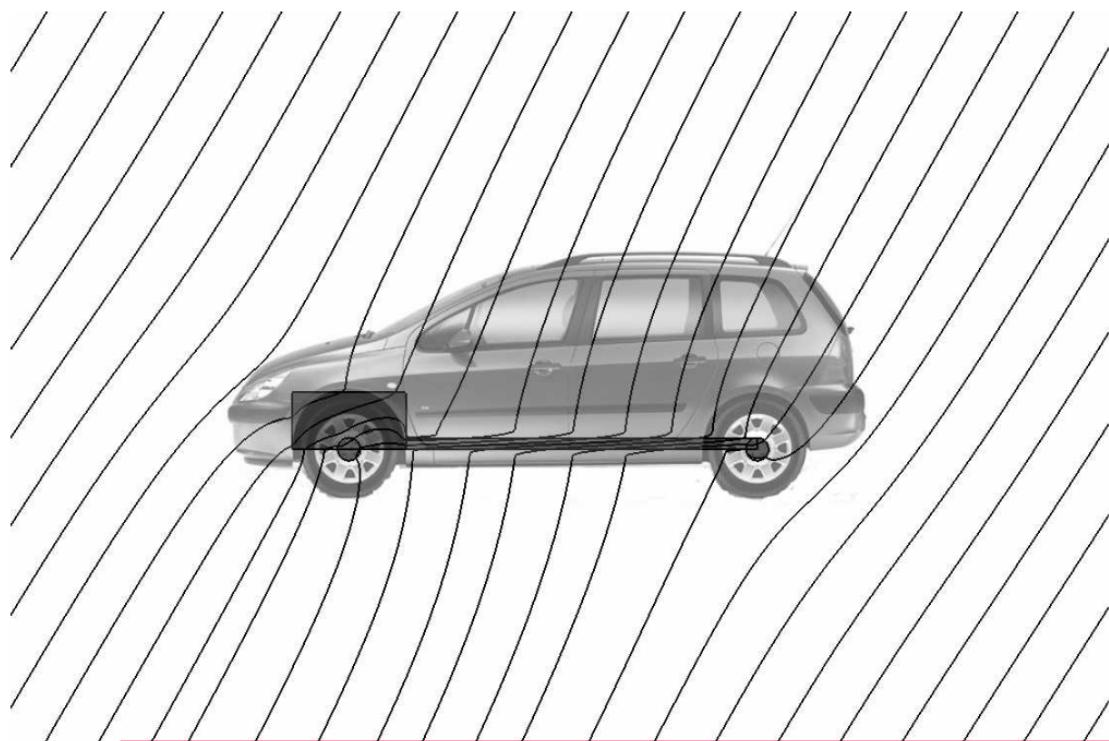
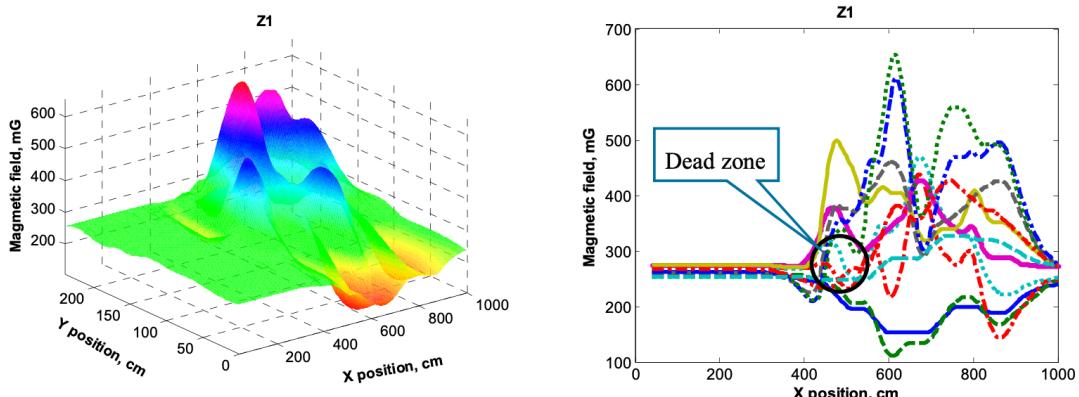


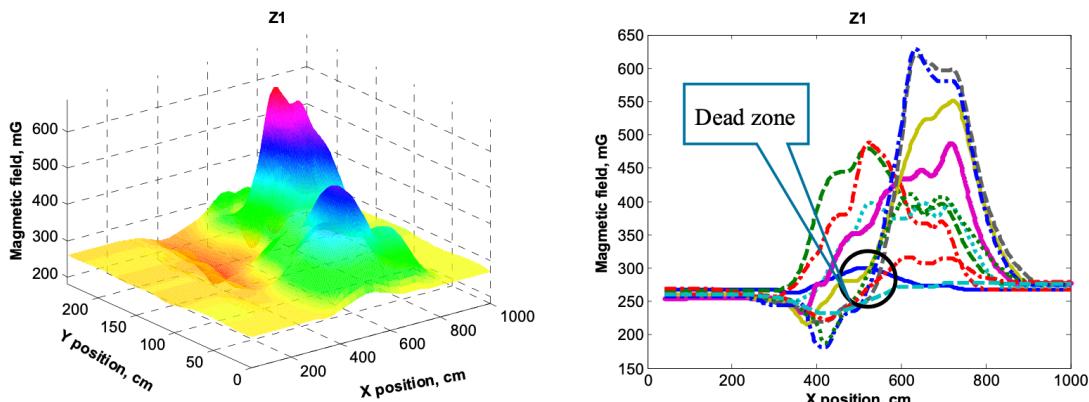
Fig. 1. Simulation of the deformation of the Earth's magnetic field by a car

Можливі недоліки:

Vehicle No 2



Vehicle No. 3



Також в дослідженні було знайдено, що в залежності від розташування датчика, інколи машина може потрапляти в "сліпу зону" (вирішення: встановлення датчика з краю (під двигун чи задню частину автомобіля); крайній випадок: встановлення додаткових датчиків на дорогу, щоб при приближенні автомобіля кидати через радіо інтервал та збільшувати частоту збору інформації (що дозволить встановити факт зміни при паркуванні));

2. Радіо модуль

Досить довго шукав, що підійде краще. Можливі варіанти:

- HC 12 (дозволяє створювати one-to-many star topology до 255 сенсорів, втім передає інформацію побайтово (через завади буде дуже складно make it work))
- LoRa (значно дорожі, та й надлишкові)

Враховуючи ціну/функціонал, поки найкращим варіантом видається nRF24 від Nordic Semiconductors.

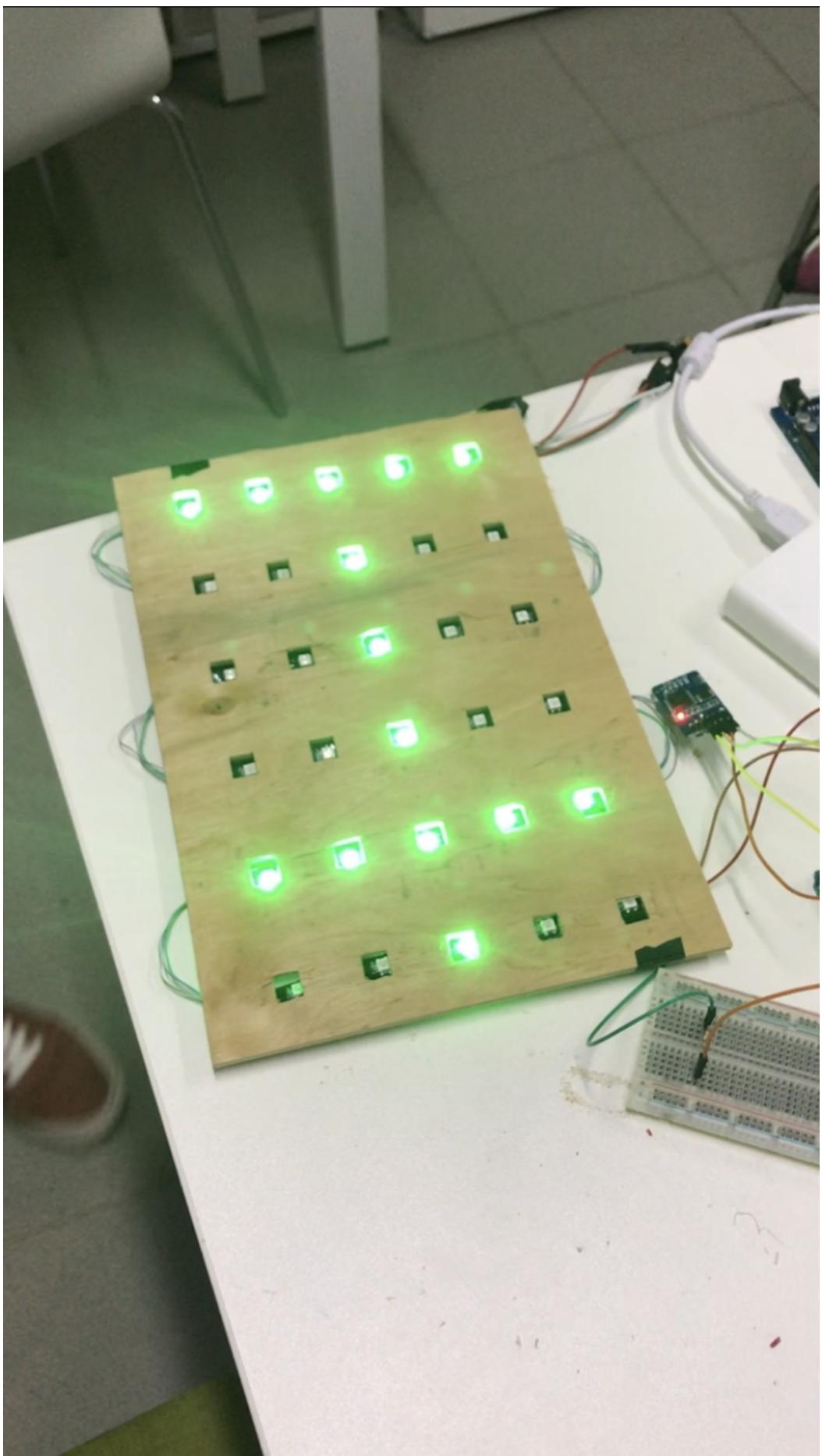
Обмеження: кожен receiver може отримувати передачу з 6 інших transmitters (each RF channel is logically divided into 6 parallel data channels called Data Pipes). Втім, знайшов вирішення цієї проблеми: замість того щоб надавати кожному датчику логічну адресу, можна кільком надати одну й ту саму та визначати адресу за кодом у надісланому повідомленні. Тоді можна розробити систему синхронізації, щоб для однієї адреси не відбувалось надсилання з кількох датчиків (що не повинно бути складно дуже). Таким чином, можна розширити систему для одного ресівера до, для прикладу, 36 модулів, що видається непоганим результатом.

За таких умов, у нас буде кілька гейтвеїв (радіо модуль + NodeMCU з виходом до інтернету), що підвищить стабільність (у разі одного, шанси колізій збільшуються), та дозволить легко скейліти (збільшувати) систему.

3. Щоб показувати користувачу інформацію про вільні паркомісця пропоную такі підходити:

- Arduino + LED Strip з постійним живленням, на яку буде приходити інформація з сервера про зайнятість парко-місць

Щось таке:



- Також придумав, як все ж використати BLE (дозвіл оплати через смартфон: дуже приблизна локалізація, що синхронізується з даними із магнетометрів)

План роботи:

- Заселепити таки варіант radio нетворк, про який згадано вище.
- Зібрати End Node (магнетометр + датчик температури + радіо модуль) на основі PSoC
- Протестити в реальних умовах, внести певні зміни
- Якщо працюватиме добре, повернусь до BLE (щоб реалізувати оплату через смартфон)

Питання/прохання:

- Якщо є якісь датчики саме від Cypress, про які я можу не знати, було б добре почути про них!

P.S останній два тижні зробив собі дещо канікули, тому трохи мало працював над проектом, втім зараз планую активно працювати, щоб до кінця літа мати перший готовий варіант системи (який потім ітеративно покращити).

P.P.S щодо BLE пробував багато, результатами незадоволений. За потреби пояснення можу підготувати звіт по них, втім здається, що зараз це не зовсім потрібно (хоч якісь напрацювання безперечно є).