

## Biblioteka kompas

`void compassInitialize(void);`

Inicjalizuje działanie dwóch kompasów MAG3110 oraz LSM303D, uruchamia komunikację I2C.

`void startCalibration(void);`

`void stopCalibration(void);`

Funkcje rozpoczynające i kończące kalibrację kompasu. Po wykonaniu funkcji należy obrócić kompas o co najmniej 360 stopni. Wtedy można zakończyć wykonywanie funkcji. Funkcja zbiera informacje i największym i najmniejszym natężeniu pola magnetycznego przy osiach X,Y. Następnie wprowadza do kompasów  $\text{offset} = \text{max} + \text{min}$  (dla MAG3110, offset nie wynisi  $(\text{min} + \text{max})/2$ , tak jak by można przypuszczać).

Funkcja `stopCalibration()` może być również napisana jako `uint8_t stopCalibration(void)`. Wtedy zwracaną wartością będzie 0 – nie powinno się kończyć kalibracji, za mało danych zostało zebranych, lub 1 – kalibracja zakończona poprawnie.

W funkcji też ustalany jest współczynnik proporcjonalności AR pomiędzy osiami X i Y. Wynosi on  $(\text{maxX} - \text{minX}) / (\text{maxY} - \text{minY})$ ;

`void setAveraging(uint8_t num);`

Aby kompas posiadał większą dokładność dane będą zbierane z dwóch kompasów i uśredniane. Funkcja ta ustala ile próbek ma być pobranych aby dać wynik funkcji natężenia pola w danym kierunku. Domyślna wartość uśredniania to 5. Jeżeli zmiany kierunku robota są wolne to należy ustawić dużą wartość. Dla szybkich zmian lepsza jest wartość mniejsza.

`uint16_t get_mag_X(void);`

`uint16_t get_mag_Y(void);`

Funkcje zwracające natężenie pola w 2 kierunkach po uśrednieniu i użyciu współczynnika AR. Nie ma funkcji zwracającej wartość w osi Z, ponieważ wymagało by to kalibracji w tym kierunku.

`uint16_t get_mag_Dir(void);`

Funkcja zwracająca kierunek na podstawie funkcji `get_mag_X()` oraz `get_mag_Y()`;

`uint8_t set_mag_int(uint8_t angle);`

Funkcja wywołująca przerwanie, kiedy wartość obrotu zmieni się o `angle` od wartości obrotu odczytanej w chwili wykonania funkcji. Możliwe, że będzie konieczne użycie timera i co pewien czas odczytywanie wartości funkcji `get_mag_Dir()`. Wtedy zalecane byłoby użycie niskiej wartości uśredniania. Funkcja zwraca 1 jeżeli pomyślnie ustawiono przerwanie. 0 kiedy wystąpił błąd.