

Temat: Sprawozdanie z testowania zaawansowanych czujników Lego mindstorm NXC.

1. Uwagi ogólne:

Wszystkie czujniki są bardzo wrażliwe na zakłócenie w kablach (mogło być ono powodem problemów, które zostały opisane poniżej związanych ze skalowaniem wartości) – należy w związku z tym wszystkie kable na stałe przymocować (taśmą, bądź napiąć je mocno) z daleka od siebie i innych elementów elektronicznych.

2. Kompas

Używany poprzez funkcje:

```
SetSensorLowspeed(IN_2);
```

```
int Comp_value = (SensorHTCompass(IN_2));
```

Czujnik należy zamontować (wg opisu) ok 20 cm od najbliższego innego czujnika/silnika/kostki. W rzeczywistości wydaje się, że wartość 10-15 cm jest wystarczająca. Czujnik daje zadowalające odczyty nawet w pokoju pełnym komputerów i fal elektromagnetycznych z różnych źródeł (praktycznie zerowe wahania), ale jego zbliżenie na odległość mniejszą niż 10 cm do przyrządu/kabla elektrycznego bądź kawałka metalu znacznie zaburza odczyty. W przypadku oddalenia od takich elementów czujnik działa bardzo dobrze.

Najlepszym ustawieniem kompasu wydaje się być podniesienie go wysoko ponad silniki i kostkę.

3. Akcelerometr:

Używany poprzez funkcje:

```
SetSensorLowspeed(IN_1);
```

```
bool check = ReadSensorHTAccel (IN_1, Accel_value_x, Accel_value_y, Accel_value_z);
```

Używany poprzez 'całkowanie' (sumowanie) wartości (do predkości i przesunięcia) i przeskalowywanie ich do odpowiednich jednostek. Nie należy skalować pojedynczych odczytów ($1 * 19/10 = 1$), lecz ich sumę. To prowadzi do niedogodności związanych z dużymi liczbami (int ma zakres ok +/- 32k) – w naszym programie (mierzącym przemieszczenie) oznaczało to ograniczenie do ok. 90cm oddalenia.

Kolejny problem który się pojawił był związany ze skalowaniem wartości – okazało się, że wartości dodatnie i ujemne należy skalować oddzielnie (z różnymi współczynnikami) – tylko wtedy wynik zsumowania dawał zero. Niestety brak czasu nie pozwolił nam na sprawdzenie, czy

to był jedyny problem – nie udało nam się uzyskać wyniku w centymetrach (ani wyników dot. Prędkości chwilowej) które byłyby przez dłuższy czas zgodne ze stanem rzeczywistym robota.

Wyznaczenie wartości zerowej przyspieszeń także sprawia problemy – czujnik czasem po uruchomieniu daje kilka odczytów bardzo dużych. Dodatkowo ustawienie czujnika idealnie poziomo przy klockach LEGO jest bardzo trudne – może z tego wyniknąć stałe przyspieszenie, które nie jest w jednostkach przekazywanych przez czujnik całkowite (w idealnym przypadku zerowe przyspieszenie oznacza brak ruchu w danym kierunku), w związku z czym wahać się pomiędzy dwoma wartościami (wahania większe niż o 1 jednostkę są rzadkie, ale występują).

Czujnik naszym zdaniem najrozsądniej zamontować pomiędzy kołami napędowymi.

4. Żyroskop

Używany poprzez funkcje:

```
SetSensorTouch(IN_3);
```

```
int Gyro_value = SensorRaw(IN_3);
```

Wartość opisywana jako zerowa wynosi dla testowanego czujnika 611 (przy innym humorze czujnika 612), czyli tę wartość należy odjąć od odczytanej, aby otrzymać wartość prędkości kątowej.

Używany poprzez ‘całkowanie’ (sumowanie) wartości i przeskalowywanie ich do odpowiednich jednostek. Dalsze uwagi jak w przypadku akcelerometru. Standardowe wahania wartości zaobserwowane przez nas wynoszą +/-2, ale zdarzają się też dużo większe.

Również w tym przypadku napotkaliśmy problem ze skalowaniem wartości dodatnich i ujemnych – użyte przez nas wzmocnienia 580/400 dla wartości dodatnich i 567/400 dla wartości ujemnych pozwoliły uzyskać błąd ok. 1 stopień/pełen obrót. Po zmianie baterii ten błąd się nie zmienił, zaś po przełożeniu kabli w inne miejsce (bliżej innego kabla) już tak.

Czujnika zdecydowanie nie należy montować na osi obrotu pojazdu, lepiej z dala od niej.