L A B O R A T O R I U M P R O J E K T O W E

Piotr Klasa

CompassRobot

Projekt CompassRobot to robot zbudowany z ZUMO Shield i płytki Freescale FRDM sterowany za pomocą komputera i analizujący odbierane dane o kierunku obrotu robota. Projekt składa się z programu na płytkę FRDM napisanego a języku C oraz aplikacji na komputer stworzonej w LabView. Komunikacja z komputerem odbywa się za pomocą bluetooth.

Opis działania robota

W głównej funkcji programu, program inicjalizuje wszystkie elementy z których korzysta – diody LED , wyświetlacz LCD, bluetooth, TMP dla sterowania ruchem oraz interfejsy I2C i kompasy. Diody, wyświetlacz LCD oraz przyciski były używane podczas testów.

Po inicjalizacji program wchodzi do pętli nieskończonej while(1) i czeka na przerwanie. Jednym z przerwań jest wywoływane co pewien okres przerwanie od timera, które wysyła dane na temat obroty robota do komputera. Drugim możliwym przerwaniem jest odebrana informacja od komputera z numerem komendy. Robot na podstawie numeru lub dwóch kolejnych danych rozróżnia czynność, którą ma wykonać. Większość komend jest związana z biblioteką motodriverinit i poruszaniem się robota. Biblioteka ta powstała na bazie biblioteki pobranej ze strony fpga.agh.edu.pl, ale dodałem do niej funkcje ułatwiające sterowanie: (wszystkie funkcje, które rzeczywiście wykorzystuję w robocie poza funkcjami samego TPM są moje) obrót o dany kąt i obrót do danego kąta z odpytywaniem czy robot jest już obrócony prawidłowo. Obrót w prawo i lewo jest też teraz możliwy przy jednoczesnym ruchu do przodu lub tyłu. Jeżeli prędkość gąsienicy jest mniejsza od 0 to prędkość jest równa wartości bezwzględnej, a kierunek jest przeciwny. Takie funkcje pozwalają na jednakowe traktowanie jazdy w przód i tył, co bardzo ułatwia, kiedy chcemy dodać skręt w prawo przy prędkości 5. Wtedy prawa gąsienica będzie jechać do tyłu z prędkością 5, a lewa do przodu z prędkością 15.

```
SetTracksDir( (left>0)?FORWARD:REVERSE,(right>0)?FORWARD:REVERSE );
setTracksSpeed( fabs(left),fabs(right) );
```

Kolejną z komend, którą możemy odebrać poprzez moduł bluetooth jest kalibracja. Wtedy wewnątrz przerwania od bluetooth jest uruchamiane przerwanie od timera. To przerwane musi mieć większy priorytet, aby było obsługiwane podczas przerwania bluetooth. Robot sam wykonuje obrót w lewo, po pewnym czasie przerywa i zapisuje dane o kalibracji do rejestrów w magnetometrach – oba są jednocześnie kalibrowane, ale korzystać z nich można osobno. Kalibracja polega na znalezieniu maximum i minimum na każdej z osi. Jedna niepoprawna wartość, która będzie dużo większa od pozostałych bardzo zniekształciła by dane kalibracji, dlatego dane dla kalibracji dane odczytywane są 4 razy i z nich jest liczona średnia. Zapamiętywana jest wtedy również wartość AR, która jest stosunkiem amplitudy wartości na osi X do osi Y. Kąt jest odczytywany jako atg2(x,y*AR) – czyli funkcja atg z uwzględnieniem znaków argumentów. Taki sposób nie sprawdzi się kiedy robot będzie jechał po nierównej powierzchni, ale daje najdokładniejsze wyniki dla powierzchni płaskiej.

Wysyłając polecenie wyboru kompasu zmieniamy tylko jedną zmienną w programie – compassSelect. Komunikacja z kompasami odbywa się poprzez porty I2C0 oraz I2C1. W plikach obsługi magnetometrów podczas wysyłania danych lub odbierania chwilowo wyłączam przerwania, ponieważ czasami zakłócały one prace programu.

Pozostałe, bardziej skomplikowane funkcje i dane są przechowywane na komputerze. LabView odbiera kolejne dane od robota i łącząc z obecną prędkością jazdy tworzy mapę po której robot przejechał. Na tej podstawie może obliczyć kąt w kierunku którym robot się powinien obrócić, aby wskazywał punkt, w którym zaczynał trasę.

Pliki z biblioteki obsługi kompasu częściowo różnią się od tych użytych w projekcie. Jedną z różnic jest, w bibliotece do skrętu o dany kąt lub do danego kątku użyte są przerwania. W robocie zastąpione są odpytywaniem.