

robot mobilny, układ sterowania, algorytm sterowania, Pocket PC, mikrokontroler

Robert LUKIERSKI¹

SYSTEM PROGRAMOWANIA ALGORYTMÓW STEROWANIA ROBOTÓW MOBILNYCH

Implementacja algorytmu sterowania robotów mobilnych w mikrokontrolerze o niewielkich zasobach sprzętowych i małych możliwościach obliczeniowych wymaga od programisty dużego doświadczenia. Ponadto uruchamianie w takich warunkach oprogramowania po jego implementacji jest znacznie utrudnione. W pracy przedstawiona została koncepcja i realizacja systemu wspomagającego tworzenie i testowanie algorytmów sterowania. System składa się z mikrokontrolera oraz komputera klasy PocketPC. Mikrokontroler jest wykorzystywany wyłącznie do implementacji niskopoziomowej części algorytmu sterowania (sterowanie napędami, odczyt danych z czujników itp.), natomiast komputer PocketPC pozwala na wykonywanie pełnego algorytmu sterowania. W przyjętym rozwiązaniu jest on zaimplementowany w języku C#. Takie podejście dostarcza znacznie większej mocy obliczeniowej i pozwala na łatwe dołączenie dodatkowego sprzętu (typu kamera, komunikacja WiFi TCP/IP itp.) przy jednoczesnym zachowaniu prostoty i zapewnieniu wydajności procesu programowania (język C#). Moduły zaproponowanego systemu komunikują się za pośrednictwem interfejsu IrDA. W tym celu opracowano oraz zaimplementowano protokół komunikacji, zarówno w języku C# dla komputera PocketPC jak i w języku C dla mikrokontrolera ARM7. W celu zilustrowania działania całego systemu dodatkowo zaimplementowano w nim przykładowe algorytmy sterowania.

PROBLEM

Obecnie robotyka mobilna zaczyna odgrywać istotną rolę w codziennym życiu. Roboty opuszczają laboratoria i znajdują zastosowanie w przemyśle wytwórczym, rozrywkowym oraz militarnym. Stawiane przed nimi zadania wymagają od nich

¹ Koło Naukowe Robotyków „KoNaR”
Zakład Podstaw Cybernetyki i Robotyki
Politechnika Wroclawska

interakcji ze światem. Wiąże się to z zastosowaniem złożonych systemów sensorycznych oraz ze wzrostem złożoności obliczeniowej algorytmów. Coraz częściej problemem twórcy robota jest warstwa wyższa oprogramowania robota (planowanie trajektorii, sztuczna inteligencja itp.).

Celem realizowanego projektu jest opracowanie środowiska programowania robotów, które pozwoli skupić się programiście na realizacji złożonych algorytmów sterowania, odciążając go równocześnie od żmudnej implementacji niższych warstw sterownika robota. W niniejszym opracowaniu zastosowano komputer typu PocketPC oraz język C# (wraz z platformą .NET Compact Framework) jako docelowe środowisko programowania, oraz mikrokontroler NXP LPC2148 ARM7 wykorzystany do obsługi sprzętowych elementów robota oraz niższych warstw.

ROZWIĄZANIE PROBLEMU

ARCHITEKTURA SYSTEMU

Na rysunku 1 przedstawiono schematyczną hierarchię warstw zaproponowanego rozwiązania.



Rys. 1. Hierarchia systemu

Zadaniem warstwy fizycznej jest komunikacja z mikrokontrolerem poprzez protokół IrDA bądź port szeregowy (UART). Warstwa ta została zaimplementowana obiektowo co pozwala na zmianę stosowanego nośnika informacji bez modyfikacji kodu dalszych warstw.

Struktura zaproponowanego protokołu jest wzorowana na protokole „MODBUS” [1]. Charakteryzuje się on prostą specyfikacją oraz implementacją. W przypadku transmisji UART komunikacja może jednak zachodzić w trybie full-duplex. Na rysunku 2 przedstawiono ramkę protokołu gdzie:



Rys. 2. Ramka protokołu

- K – kod komendy (1 bajt)
- R – rozmiar danych (1 bajt)
- D – dane
- A – suma kontrolna Adler-32 (4 bajty)

Zadaniem warstwy BSP (ang. „board support package”) jest transmisja komend zgodnych z zastosowanym typem robota. Zapewnia ona obsługę protokołu oraz warstwy fizycznej. Warstwa BSP pozwala na stosowanie obiektowo ujętego opisu w celu ujednoczenia sterowania dla różnych typów robotów (inny protokół, inne wyposażenie robota). Warstwę aplikacji stanowi program użytkownika, korzystający wyłącznie z warstwy BSP do zadawania sterowań oraz obsługi danych sensorycznych.

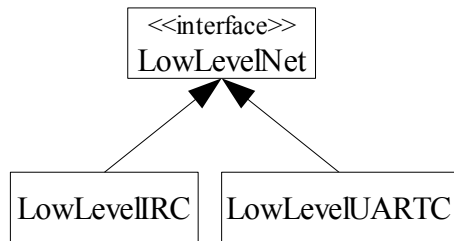
IMPLEMENTACJA

MIKROKONTROLER

Przykładową realizację układu sterowania oparto o robota mobilnego Ulisses 2 [2]. Zastosowano w niej mikrokontroler klasy ARM7 firmy NXP. Wykorzystując język C oraz peryferia zawarte w układzie zaimplementowano sterowanie silnikami firmy PortEscap z enkoderami optycznymi, w zamkniętej pętli sprzężenia zwrotnego z regulatorem PID. Dodatkowo mikrokontroler poprzez wbudowany przetwornik ADC obsługuje dalmierz laserowy OCD5000 firmy Pepperl-Fuchs. Korzystając z układu MCP2150 oraz wbudowanych portów UART zaimplementowano warstwy fizyczne, oraz obsługę protokołu.

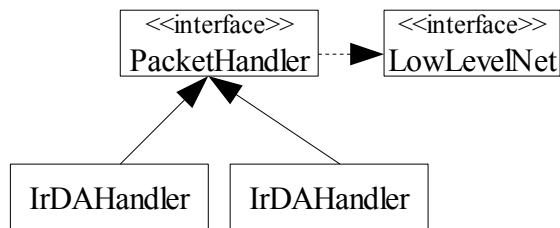
POCKET PC

Zastosowano komputer klasy Pocket PC firmy Acer, model N50 Premium, wyposażony w system operacyjny Microsoft Windows Mobile 2003. Komputer ten pozwala na komunikację przy pomocy interfejsów IrDA, WiFi oraz RS-232 (przy zastosowaniu podstawki dokującej). Producent zastosował w nim szybki procesor StrongARM Intel PXA272 o częstotliwości zegara 520MHz, co wraz z 128 MB pamięci RAM dostarcza wydajnej platformy sprzętowej. Dodatkowo możliwe jest również wykorzystanie wbudowanego złącza kart SD/MMC oraz CompactFlash do rozbudowy systemu.



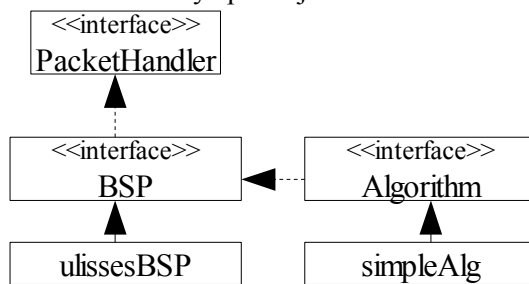
Rys. 3. Diagram klas warstwy fizycznej

Na rysunku 3 przedstawiono diagram dziedziczenia dla klas obsługujących warstwę fizyczną. Powyższe klasy implementują metody interfejsu „LowLevelNet”, przy czym klasa LowLevelIRC stanowi klienta protokołu IrDA (wykorzystano bibliotekę 32feet.net), a klasa LowLevelUARTC stanowi klienta protokołu szeregowego. Rysunek 4 przedstawia diagram klas obsługujących warstwę protokołu oraz ich związek z niższą warstwą.



Rys. 4. Diagram klas warstwy protokołu

Klasy implementujące interfejs „PacketHandler” pośredniczą pomiędzy warstwą fizyczną a warstwą BSP. Rysunek 5 przedstawia diagram relacji między klasami warstwy protokołu, BSP oraz warstwy aplikacji.

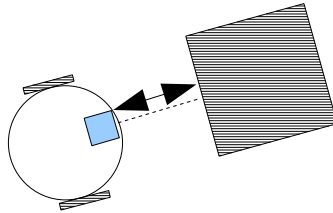


Rys. 5. Diagram klas warstw BSP oraz aplikacji.

Interfejs „BSP” implementuje klasa, która współpracuje z danym modelem/oprogramowaniem mikrokontrolera robota. Natomiast interfejs „Algorithm” implementuje użytkownik systemu robotycznego.

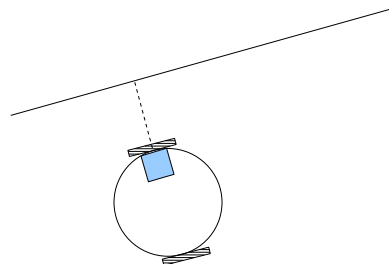
WYNIKI BADAŃ

W ramach testów systemu zaimplementowano proste algorytmy sterowania reakcyjnego wykorzystujące zastosowany przemysłowy dalmierz laserowy, charakteryzujący się wysoką dokładnością. W pierwszym przypadku robot bazując na danych z dalmierza utrzymuje zadany dystans do obiektu w polu czujnika, poruszając się po linii prostej (rys. 6).



Rys. 6. Utrzymywanie zadanej odległości

Po przesunięciu czujnika możliwa była realizacja algorytmu jazdy wzdłuż ściany, gdzie robot poruszając się utrzymuje jednocześnie zadaną odległość od ściany bazując na pomiarach dalmierzem (rys. 7).



Rys. 7. Jazda wzdłuż ściany

W początkowej fazie projektu zdecydowano się, na wykorzystanie w procesie komunikacji wyłącznie interfejsu IrDA, mając na uwadze jego uniwersalność. Jednak w trakcie prac ujawniła się jego istotna wada, mianowicie wyłącznie praca w trybie half-duplex. Pozwala to odczytywać dane sensoryczne na zasadzie „pytanie-odpowiedź”, jednak takie rozwiązanie nie jest wygodne w robocie mobilnym, który często musi się cechować szybką reakcją na bodźce zewnętrzne. Z tego powodu zastosowano klasyczny interfejs RS-232, charakteryzujący się pracą w trybie full-duplex. Zaimplementowano dyskretne regulatory PID co wymagało szeregu testów oraz doboru odpowiednich nastaw. W projekcie położono nacisk na stworzenie systemu programistycznego oraz zbadanie zagadnień komunikacji, nie na stworzenie rozbudowanego robota mobilnego, dlatego w przykładowym robocie zastosowano jedynie podstawowy zestaw układów sensorycznych i wykonawczych.

PODSUMOWANIE

Wyniki prac potwierdzają sens tworzenia systemów programowania robotów mobilnych, ponieważ stosowanie metod programowania obiektowego stwarza możliwość ponownego wykorzystania kodu oraz ułatwia proces opracowywania kodu. Dodatkowo zasoby mocy obliczeniowej komputera PocketPC pozwalają na realizację złożonych algorytmów stosujących metody sztucznej inteligencji, planowanie trasy czy system wizyjny. Mając za bazę taką platformę sprzętową łatwo możemy rozszerzyć możliwości robota o:

- rejestrację pomiarów na karcie pamięci SD/MMC
- wyposażenie robota w kamerę VGA (również złącze SD/MMC)
- komunikacja z panelem operatorskim przy pomocy protokołu WiFi (IEEE 802.11)

Powyższe kwestie są przedmiotem dalszych prac, mających na celu usprawnienie i rozbudowanie istniejącego systemu.

LITERATURA

[1] <http://www.modbus.org/>, 15 marca 2007

[2] SŁOWIŃSKI M. Komputer pokładowy dla robota mobilnego, Praca Dyplomowa. Instytut Informatyki, Automatyki i Robotyki.

PROGRAMMING SYSTEM FOR MOBILE ROBOTS CONTROL ALGORITHMS

Implementing control algorithms on microcontrollers used in mobile robots requires from the developer lot of experience, and process is difficult. Insufficient processing power of microcontrollers us an additional problem. In this article a concept and implementation of the system to aid creating and testing control algorithms is presented. The project hardware includes a microcontroller and a PocketPC handheld. The microcontroller covers the low-level part of the robot control (motor control, sensor data aquisition), the PocketPC is used to execute the high-level control algorithm. In the proposed solution the algorithm is implemented in the C# language. Such approach provides much more processing power and facilitate the hardware extension (for example camera, WiFi communication etc.) at the same time having simplicity and ensuring performance of the programming process (C# language). The modules of the offered system communiante with use of the IrDA interface. The communication protocol was developed and implemented both in the C# language for the PocketPC and the C on the ARM7 microcontroller side. Simple control algoritmhms were created to exemplify the project goals.