



właścicielem praw jest Marvel Entertainment i Paramount Pictures

Polski Ironman (1)

Pojazd sterowany rękawicą

Informując w tytule „polski Ironman” przedstawiam wszystkim samochód sterowany rękawicą, a dokładniej akcelerometrem umieszczonym na dłoni. Zmieniając wychylenie rękawicy w płaszczyźnie XY jest możliwe sterowanie pojazdem, tak aby poruszał się płynnie do przodu, do tyłu oraz skręcał. Projekt został wymyślony oraz wykonany przez mnie podczas studiów, co na tamtą chwilę dawało masę frajdy oraz godziny dobrej zabawy. Pomysł i inspirację zawdzięczam postaci z komiksu oraz filmu – Ironman. Przedstawiony projekt ma służyć celom edukacyjnym.

Obecnie jest kilka metod używanych do sterowania pojazdami. Wiele z takich rozwiązań jest używanych w modelarstwie. Dawniej dużo frajdy sprawiał pojazd sterowany przewodowo, obecnie takie rozwiązanie zostało całkowicie wyparte przez łączność radiową. Nie wymieniając wielu zalet tej metody należy podkreślić, że na rynku dostępnych jest wiele układów oraz gotowych modułów komunikacyjnych, które w prosty sposób mogą być zastosowane do budowy własnych urządzeń modelarskich.

To co zostało nam do przeanalizowania, to przyszłe zastosowanie naszego pojazdu. Projekt przedstawiony w tym artykule, ma służyć głównie celom edukacyjnym, przy-

najmniej takie było założenie. Nietypowa forma sterowania niewątpliwie przyciąga uwagę widzów.

Opis

Urządzenie składa się z dwóch części:

- **część nadawcza.** Rękawica, na której jest umieszczony akcelerometr. Dzięki zmianie położenia rękawicy operatora (manipulacja dłonią w przestrzeni XY), jest możliwe sterowanie pojazdem: przód/tył oraz skręcanie w lewo/prawo,
- **część odbiorcza.** Pojazd mobilny z umieszczonym modułem odbiorczym. Otrzymuje on informację z części nadawczej oraz realizuje sterowanie silnikami.

Akcelerometr MMA7260

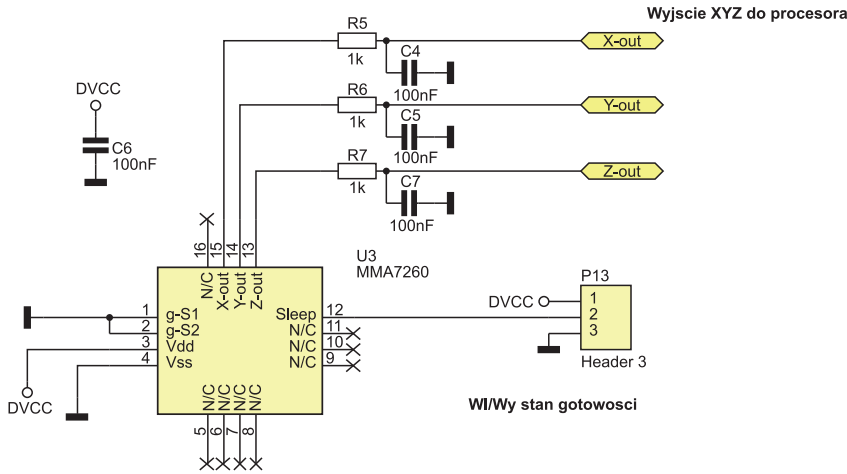
Do sterowania pojazdem (przód/tył/skręt) zastosowano akcelerometr MMA7260 firmy Freescale, umieszczony na rękawicy części nadawczej operatora.

Akcelerometr służy do pomiaru przyspieszenia. Może np. wykrywać silne uderzenie podczas wypadku samochodowego, co wyzwała zadziałanie poduszki powietrznej. Układ ten nadaje się również do wykrywania wychylenia w płaszczyźnie XY.

Na rynku elementów elektronicznych dostępnych jest kilka rodzajów scalonych układów akcelerometrów min.:

- firma Freescale oferuje układy z rodziny MMA,
- firma Analog Devices, rodzina ADXL,
- firma Memsic, rodzina MX.

Użyty akcelerometr to 3 osiowy układ analogowy, z możliwością wyboru czułości z zakresu 1,5...6g. Pomiar przyspieszenia/wchylenia realizowany jest przez pomiar pojemności pomiędzy dwiema okładzinami. Zasady działania różnych akcelerometrów były dokładnie opisane w „Wyborze konstruktora” w EP 2/2010 (str. 60).



Rys. 1. Schemat ideowy połączenia akcelerometru

W układzie wielkością wyjściową dla każdego z kanałów jest napięcie 1,65 V, które oscyluje w granicach do ± 1 V zależnie od wychylenia.

Schemat ideowy sposobu podłączenia akcelerometru umieszczono na rys. 1. Akcelerometr MMA7260 (U3) ma wyprowadzone wyjścia sygnałów analogowych trzech osi XYZ. Rezystory R5, R6, R7 zapewniają ograniczanie prądowe do zewnętrznych układów. Wszystkie kondensatory 100 n zastosowano w celu likwi-

dacji szumów napięcia wyższych częstotliwości. Sygnały g-S1, g-S2 podłączone do masy zasilania ustawiają czułość na wartość 1,5 g, co do opisanego zastosowania jest zupełnie wystarczające.

Napięcie z wyjścia akcelerometru jest podawane na wewnętrzny przetwornik analogowo-cyfrowy procesora LPC2146 firmy NXP. Użyte zostały dwie osie – X oraz Y. Wychylenie w osi X steruje jazdą do przodu/tyłu, natomiast w osi Y powoduje skręcenie pojazdu.



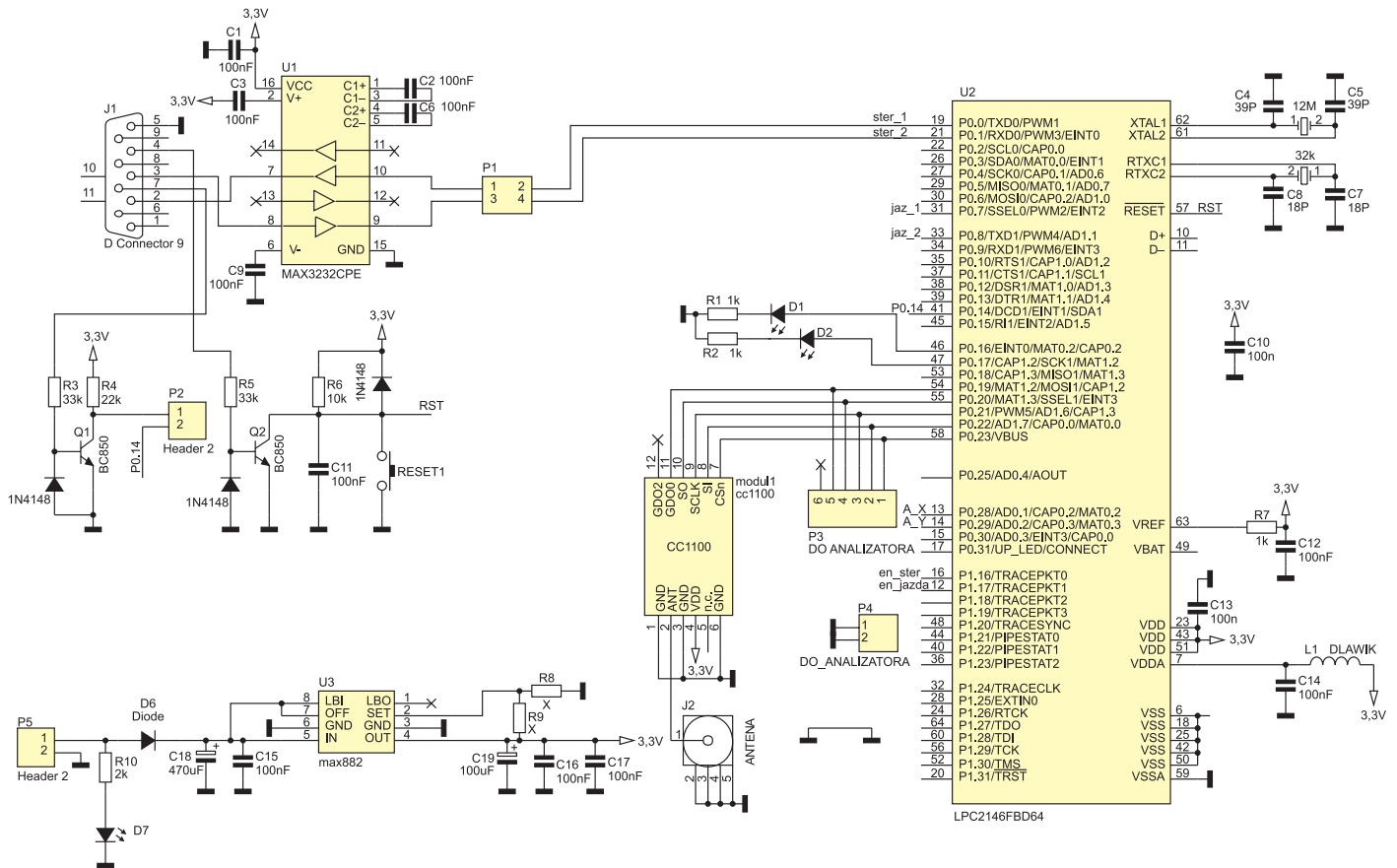
Rys. 2. Sugerowane umiejscowienie elementów części nadawczej

Nadajnik sterujący

Jako moduł radiowy zastosowano produkt firmy ARS. Moduł bazuje na układzie CC1100 firmy Texas Instruments. Sposób programowania oraz konfiguracji CC1100 był opisywany we wcześniejszych numerach EP, począwszy od 03/2008.

Część nadawczą stanowi rękawica, na której został umieszczony akcelerometr. Poprzez zmianę położenia rękawicy operatora (manipulacja dłonią w przestrzeni XY), moż-





Rys. 3. Schemat układu ideowy części nadawczej. Opis sygnałów ze schematu: Ster_1 – [wyjście] sygnał sterowania lewo; Ster_2 – [wyjście] sygnał sterowania prawo; Jaz_1 – [wyjście] sygnał sterowania przód; Jaz_2 – [wyjście] sygnał sterowania tył; A_X – [wejście] sygnał z kanału X akcelerometru, przód-tył; A_Y – [wejście] sygnał z kanału Y akcelerometru, lewo-prawo; En_ster – [wyjście]sygnał włączenia modułu sterowania; En_jazda – [wyjście]sygnał włączenia modułu sterowania

liwie jest sterowanie pojazdem w przód/tył oraz skręcanie w lewo/prawo.

Sugerowane umiejscowienie elementów części nadawczej na ramieniu operatora przedstawia rys. 2, natomiast jej schemat ideowy rys. 3.

Jednostka wykonawcza, układ ARM7TD-MI LPC2146 U1, to 32-bitowy mikrokontroler

firmy NXP (dawniej Philips) taktowany kwarcem 12 MHz. Na płytce pozostawiono możliwość dodania rezonatora kwarcowego 32768 kHz. C4, C5, C7, C8 są częścią obwodów generatorów kwarcowych. Wewnętrzny przetwornik A/C procesora wymaga zewnętrznego napięcia referencyjnego 3,3 V. Jest ono dostarczane ze źródła zasilania

przez filtr składający się z L1 i C14. Ta część jest nazywana sygnałem analogowym.

Pomimo tego, że procesor ma możliwość programowania przez interfejs J-TAG, można go również programować przez port UART (Tx0, Rx0) ustawiając odpowiednią kombinację stanów na doprowadzeniach RESET i P0.14. Za dopasowanie poziomów napięć



28-30.09.2010

zapraszamy do odwiedzenia

XII Targów Przemysłowej Techniki Pomiarowej

CONTROL-TECH

Menedżer Targów: Joanna Adamczyk
tel. 41 365 12 14, fax 41 365 13 13, e-mail: adamczyk.j@targikielce.pl

www.control-tech.pl

Patronat medialny:  

między mikrokontrolerem a interfejsem RS232 komputera PC przesyłającego program, odpowiada MAX332 (U1).

Programowanie rozpoczyna się od podania sygnału *reset* i napięcia logicznego „0” a następnie „1” na P0.14. Wówczas mikrokontroler wchodzi w tryb programowania. Po jego zakończeniu należy ponownie podać sygnał *reset*.

Za wygenerowanie odpowiednich sygnałów odpowiadają obwody zbudowane z użyciem elementów R3...R6, C11, Q1, Q2, oraz diody 1N4148. Są one dołączone do układu U1 zgodnie z zaleceniami producenta. Całym procesem steruje program FlashMagic.

Moduł radiowy dołączono do mikrokontrolera poprzez porty P0.19...P0.23. Dodatkowo, w celu umożliwienia podłączenia analizatora stanów logicznych dającego możliwość obserwacji przebiegu transmisji, dodano złącza P3 i P4. Zasilanie dostarcza układ U3 (MAX882) będący stabilizatorem LDO napięcia stałego 3,3 V do zasilania procesora, modułu radiowego oraz układu programowania procesora. Elementy C18, C19 to kondensatory tantalowe o niskiej rezystancji szeregowej LESR. Dioda D6 zabezpiecza przed zmianą polaryzacji napięcia zasilania. Do sygnalizacji obecności napięcia służy dioda D7, natomiast D1 i D2 sygnalizują odbiór bądź nadawanie. Wszystkie pojemności wartości 100 n zostały zastosowane w celu likwidacji szumów napięcia wyższych częstotliwości.

Program sterujący.

Na rys. 4 umieszczono algorytm ilustrujący sposób działania programu obsługi nadajnika.

Po starcie procesora w pętli głównej *main(void)*; następuje konfiguracja portów I/O (diody LED, sterowanie SPI), a następnie jest zerowany układ CC1100 *Rf_Power_Up_Reset()*; Po przeprowadzeniu tej operacji są wgrywane parametry konfiguracyjne pracy CC1100 *RfWriteRfSettings(&rfSettings)*; Układ wprowadzany jest w tryb nadawania *SpiWriteCommand(CCxxx0_STX)*; oraz następuje konfiguracja pracy przetwornika A/D *init_ad()*;

W pętli *while(1){*; napięcie z wyjścia X,Y akcelerometru podane na przetwornik analogowo-cyfrowy procesora LPC2146, jest w sposób ciągle próbkowa-

ne a następnie mierzone co przedstawia fragment kodu programu:

```
chan1 = get_ad_chanel(1);
chan2 = get_ad_chanel(2);
dana_x=(chan1 >> 2); //do wartości 8-bitowej
dana_y=(chan2 >> 2); //do wartości 8-bitowej
dana_xx = ((dana_x >> 4) & 0x0F);
//cztery najstarsze bity
dana_yy = ((dana_y >> 4) & 0x0F);
//cztery najstarsze bity
```

Użyte zostały dwie osie akcelerometru X oraz Y. Oś X podaje informację o poruszaniu się pojazdu do przodu oraz tyłu, oś Y podaje informację o skręcie pojazdu. Sterowanie silnikami odbywa się przez regulację wypełnienia (PWM).

Wartość analogowa każdego z kanałów pobrana przez przetwornik daje wartość 10-bitową, która jest przesuwana w prawo do pozycji czterech najstarszych bitów. Prowadzi to do analizy wychylenia akcelerometru, bazującej wyłącznie na czterech najstarszych bitach.

Wartość czterech najstarszych bitów jest analizowana przez konstrukcję *case-switch*:

```
switch(dana_xx) {
case 5 : dana_xxx=0x2; break; //+60°
case 6 : dana_xxx=0x1; break; //+30°
case 7 : dana_xxx=0x0; break; //0°
case 8 : dana_xxx=0x0; break; //0°
case 9 : dana_xxx=0x0; break; //0°
case 10: dana_xxx=0x3; break; //-30°
case 11: dana_xxx=0x4; break; //-60°
}
```

Analogicznie wygląda przetwarzanie danych dla kanału Y. Czułość kąta wychylenia dłoni, na który ma reagować układ dobrano empirycznie. Metodą prób i błędów wyznaczono kąty, przy których powinna następować reakcja pojazdu (rys. 5).

Po odpowiednim przetworzeniu danych, wysyłana jest ramka transmisyjna składająca się z jednego bajtu. Opis znaczenia poszczególnych bitów w ramce zawiera **tab. 1**.

Moduły zostały tak zaprojektowane, aby był swobodny dostęp do potencjalnych portów procesora (przez złącza potocznie nazywane goldpinami), jednocześnie zapewniając połączenie z modułem radiowym, zasilanie oraz możliwość zaprogramowania mikrokontrolera.

Procesor zakupiony jako oddzielny element, został wlutowany w uniwersalny adapter do procesora MSP430 model AVT1442.

Całość zajęła powierzchnię 110×61 mm.

Część odbiorcza

Pojazd zbudowano z gotowego zestawu kupionego w sklepie modelarskim. Został on fabrycznie wyposażony w silniki prądu stałego, regulatory ich obrotów oraz odpowiednią konstrukcję mechaniczną.

Nadajnik i odbiornik są identyczne, dlatego przedstawianie schematu oraz opis nawiązujące do opisywanego wcześniej nadajnika.

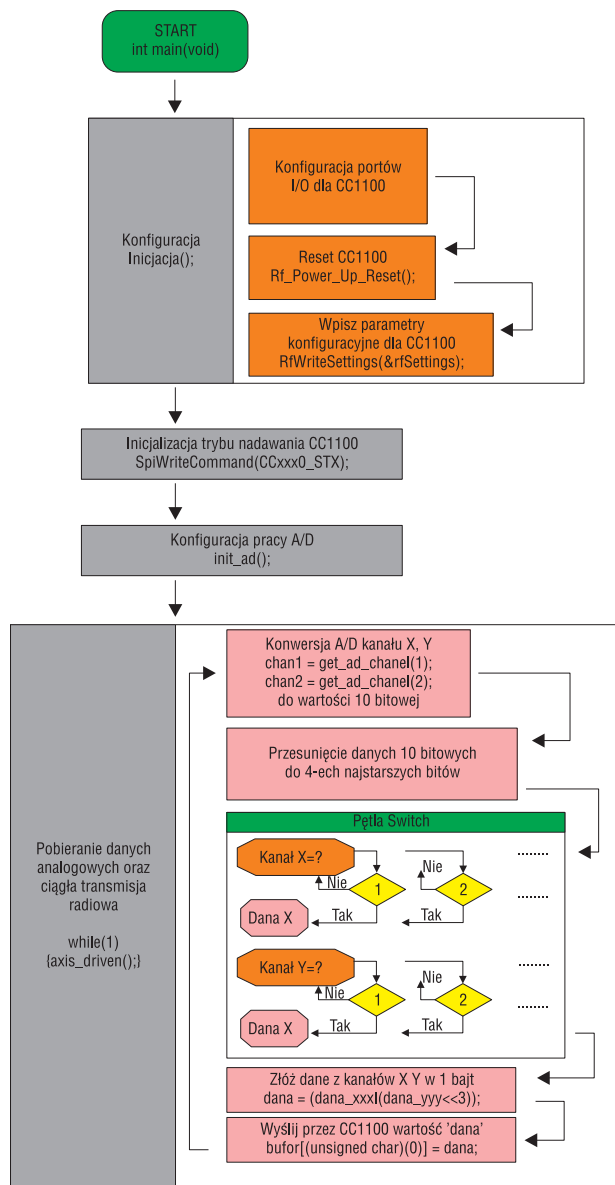
Na rys. 6 umieszczono algorytm funkcjonowania programu sterującego pracą części odbiorczej.

Po starcie procesora w pętli głównej *main(void)*; następuje konfiguracja portów I/O (diody LED, sterowanie SPI), a następnie jest zerowany układ CC1100 *Rf_Power_Up_Reset()*; Następnie są wgrywane parametry konfiguracyjne pracy CC1100 *RfWriteRfSettings(&rfSettings)*; Układ wprowadzany jest w tryb odbioru *SpiWriteCommand(CCxxx0_SRX)*; oraz następuje konfiguracja kanałów PWM *pwm_init()*;

W chwili odbioru informacji z nadajnika (pojawienie się stanu „1” na porcie G0) włącza się funkcja odbioru oraz przetwarzania danych z modułu radiowego.

Dane zostają przepisane z bufora odbiorczego do zmiennej *val*:

```
val |= bufor[0]
val_jazda = (val & 0x07);
```



Rys. 4. Algorytm programu części nadawczej



**DLA NAS
 ROZMIAR NIE MA
 ZNACZENIA...**

Moduły liniowe o dowolnej
 długości w atrakcyjnej cenie

**Wypróbuj
 nasz
 konfigurator**



www.ml.wobit.com.pl



(061) 2912 225
 (061) 8350 620



wobit@wobit.com.pl
 www.wobit.com.pl

Tab. 1. Znaczenie bitów w bajcie wysyłanym przez nadajnik							
B7	B6	B5	B4	B3	B2	B1	B0
		Tryb nadawania	2 bity: skręt w lewo/prawo		3 bity: jazda w przód/tył		
		0 = wyłączony 1 = włączony	00 = brak zmian 01 = skręt w lewo 10 = skręt w prawo		000 - bez zmian 001 - jedź prosto z prędkością 70% 010 - jedź prosto z prędkością 100% 011 - jedź wstecz z prędkością 70% 100 - jedź wstecz z prędkością 100%		

`val_ster = ((val>>3) & 0x03);`

W zmiennej `val_jazda` przechowywane są tylko te trzy bity, które są odpowiedzialne za sterowanie jazdą, reszta bitów jest zerowana. Podobna operacja jest przeprowadzana dla zmiennej `val_ster` wraz z przesunięciem o 3 pozycje w lewo (miejsce zajmowane do tej pory przez informację o jeździe pojazdu). Otrzymane w ten sposób dane są poddane analizie w `case-switch`:

`switch(val_jazda) {`

`case 0 :`

`PWMR2=0;`

`PWMR4=0;`

`jazda_enable;`

`break;`

`case 1 :`

`PWMR2=0;`

`PWMR4=70;`

`jazda_enable;`

`break;`

`case 2 :`

`PWMR2=0;`

`PWMR4=100;`

`jazda_enable;`

`break;`

`case 3 :`

`PWMR2=70;`

`PWMR4=0;`

`jazda_enable;`

`break;`

`case 4 :`

`PWMR2=100;`

`PWMR4=0;`

`jazda_enable;`

`break;`

`}`

Program dla procesora LPC2146 został napisany z użyciem kompilatora C firmy Keil. Częściowo bazuje on na przykładowym kodzie do CC1100 firmy www.ars.info.pl. Darmową wersję kompilatora można pobrać ze strony producenta www.keil.com.

Mariusz Piotr Lasota
 murphy5@o2.pl

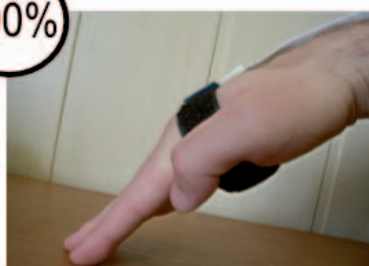
0 (prędkość)



70%



100%



70%



100%



Rys. 5. Kąty, przy których powinna następować reakcja samochodu