Ćwiczenie 19: Magistrala SPI na przykładzie akcelerometru

Instrukcja laboratorium

Mariusz Chilmon <mariusz.chilmon@ctm.gdynia.pl>

2024-06-04

> I'm not a great programmer; I'm just a good programmer with great habits.
>
> — *Kent Beck*

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z:

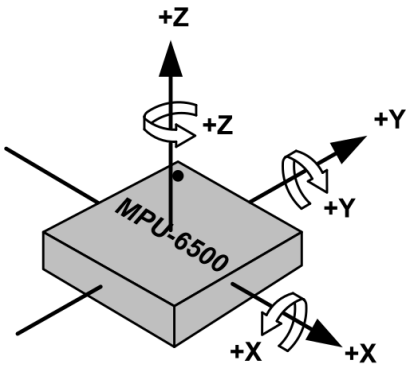
- działaniem magistrali SPI,
- odczytem danych z akcelerometru.

Uruchomienie programu wyjściowego

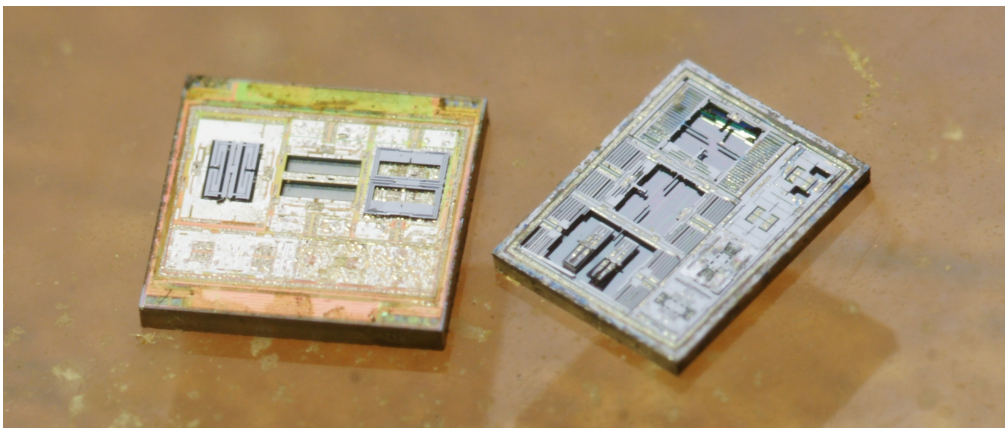
1. Podłącz płytkę *LCD Keypad Shield* do *Arduino Uno*.
2. Podłącz akcelerometr.
3. Na wyświetlaczu wyświetlane są zerowe wartości parametrów.

Rysunek 1: Wyjściowy stan wyświetlacza

Do płytki podłączony jest układ scalony MPU-6500 firmy InvenSense (obecnie wchodzącej w skład TDK), który jest czujnikiem typu MEMS (*Microelectromechanical System*), integrującym obwody elektroniczne z czujnikami mechanicznymi: trzyosiowym akcelerometrem i trzyosiowym żyroskopem. Umożliwiają one pomiar przyspieszenia (akcelerometry) oraz prędkości obrotowej (żyroskopy) w trzech ortogonalnych osiach.



Rysunek 2: Orientacja osi w czujniku MPU-6500



Rysunek 3: Budowa wewnętrzna podobnego czujnika — MPU-6050. Obok struktur elektronicznych widoczne są elementy mechaniczne, dokonujące pomiaru

Akcelerometry znajdują bardzo szerokie zastosowanie: oceniają przechył telefonu komórkowego, stabilizują obraz w aparatach fotograficznych, zliczają kroki w smartwatchu, uruchamiają poduszki powietrzne w samochodzie czy mierzą przeciążenia mózgu graczy NHL po uderzeniu krążkiem hokejowym (do tego zastosowania firma STM musiała poszerzyć zakres pomiarowy swoich czujników).

Dzięki zasadzie równoważności, która mówi, że jednorodone statyczne pole grawitacyjne jest dla obserwatora nieodróżnialne od ruchu z przyspieszeniem (*vide* pojęcie przyspieszenia ziemskiego), akcelerometr nadaje się również do pomiaru siły pola grawitacyjnego. Z kolei siła grawitacji w określonej osi jest zależna od nachylenia tej osi do np. powierzchni Ziemi, co łatwo możemy zaobserwować przechylając stół — przy pewnym kącie nachylenia składowa siły grawitacji dociskająca przedmioty do powierzchni stołu staje się zbyt mała do ich utrzymania[1].

Wobec powyższego akcelerometr zmienia wskazania w zakresie $\pm 1\ g$ w zależności od nachylenia danej osi do powierzchni Ziemi.

Zadanie podstawowe

Czujnik MPU-6500 może komunikować się z mikrokontrolerem za pomocą magistrali I^{2}C lub SPI. Ze względu na dostępne wyprowadzenia mikrokontrolera użyto interfejsu SPI.

[1] Jest to zagadnienie powszechnie wykorzystywane do dręczenia uczniów i studentów, gdyż jest praktycznym przykładem użycia trygonometrii.

Celem zadania podstawowego jest skonfigurowanie SPI i odczytanie z czujnika numeru identyfikacyjnego.

Wymagania funkcjonalne

1. Na wyświetlaczu widoczny jest identyfikator czujnika.

Modyfikacja programu

Konfiguracja interfejsu SPI

Interfejs SPI (*Serial Peripheral Interface*) jest prostym interfejsem, w którym na jednym zboczu zegara kontroler magistrali (*master*) i układ peryferyjny (*slave*) wystawiają dane, a na przeciwnym — odczytują je. Transmisja zawsze jest możliwa w obie strony, choć nie zawsze oba urządzenia rzeczywiście wystawiają dane (jeżeli nie — odbiorca ignoruje przychodzące bity).

Transmisja z kontrolera do układu peryferyjnego zachodzi na linii *MOSI* (*Master Out, Slave In*) zaś w drugą stronę — na linii MISO (*Master In, Slave Out*). Kontroler magistrali odpowiada za generowanie sygnału zegarowego. Steruje również liniami *CS* (*Chip Select*). Każdy układ peryferyjny ma swoją linię *CS*, na której stan aktywny (logiczne 0) oznacza, że dane urządzenie zostało przez kontroler wybrane do komunikacji (pozostałe linie są współdzielone). Nawet jeżeli komunikujemy się tylko z jednym urządzeniem peryferyjnym, wskazane jest sterowanie linią *CS*, gdyż łatwo wtedy wskazać początek i koniec transmisji.

Mimo zasadniczej prostoty interfejs SPI ma wiele trybów pracy, różniących się:

- polaryzacją zegara (stanem zegara w stanie uśpienia),
- fazą zegara (zboczem, na którym są odczytywane dane),
- kolejnością bitów w bajcie,
- szybkością zegara.

W celu skonfigurowania interfejsu uzupełnij metodę `SPI::initialize()` o ustawienie odpowiednich bitów w rejestrze `SPCR`. Ustaw bity `SPE` (*SPI Enable*) oraz `MSTR` (*Master/Slave Select*), aby włączyć SPI w trybie *master*, a pozostałe bity ustaw zgodnie ze sprawozdaniem z ćwiczenia.

Implementacja transferu bajtu przez SPI

Uzupełnij metodę `Spi::transfer()` o następujące kroki:

1. Wpisanie wysyłanego bajtu do rejestru `SPDR`.

2. Oczekiwanie na ustawienie flagi `SPIF`, świadczącej o zakończeniu transferu.
3. Zwrócenie odebranego bajtu z rejestru `SPDR`.

Ustawienie flagi można sprawdzać w pustej pętli:

```
1 while (bit_is_clear(SPSR, SPIF)) ;
```

Implementacja odczytu rejestru

Uzupełnij metodę `Spi::readRegister()` o następujące kroki:

1. Ustawienie linii *CS* za pomocą `gpio.chipSelect(true)`, co ustawia odpowiedni pin w stan niski (aktywny).
2. Wysłanie bajtu z adresem za pomocą wcześniej uzupełnionej metody `Spi::transfer()`. Wysyłany bajt musi mieć dodatkowo ustawiony najstarszy bit, co oznacza operację odczytu.
3. Zapamiętanie w zmiennej bajtu zwróconego przez drugie wywołanie metody `Spi::transfer()`.
4. Zwolnienie linii *CS* za pomocą `gpio.chipSelect(false)`, co ustawia odpowiedni pin w stan wysoki (nieaktywny).
5. Zwrócenie odebranego bajtu.

Odczytanie wartości rejestru *Who Am I*

W metodzie `Accelerometer::whoAmI()` wywołaj metodę `Spi::readRegister()` jako argument podając adres rejestru *Who Am I*. Na początku drugiej linii wyświetlacza powinien być widoczny identyfikator czujnika.

Adresy rejestrów znajdziesz w mapie rejestrów czujnika.

Zadanie rozszerzone

Wymagania funkcjonalne

1. Odczytywany jest pomiar przyspieszenia w osi X.
2. Wychylenie czujnika jest wizualizowane w pierwszej linii wyświetlacza.

Modyfikacja programu

Odczyt podwójnego rejestru

Uzupełnij metodę `Spi::readRegisterWord()` tak, by odczytywała dwa kolejne bajty. W tym celu po wysłaniu bajtu adresu, jak w funkcji `Spi::readRegister()` należy odczytać nie jeden, ale dwa bajty, a następnie zsumować je w zmiennej 16-bitowej, używając operatora przesunięcia bitowego o 8, czyli `<< 8`.

Odczyt pomiaru w osi X

Uzupełnij metodę `Accelerometer::measure()` tak, by z użyciem wcześniej przygotowanej funkcji `Spi::readRegisterWord()` odczytywała pomiar w osi X. Na ekranie powinien być widoczny pomiar w postaci szesnastkowej i dziesiętnej.

Zmniejsze częstotliwości pomiaru

W celu poprawienia czytelności odczytu na wyświetlaczu zmniejsz częstotliwość pracy czujnika, ustawiając na końcu funkcji `Accelerometer::initialize()` maksymalną wartość w rejestrze *Sample Rate Divider*. Funkcja do zapisu rejestrów jest już gotowa.

Wizualizacja pomiaru za pomocą bargrafu

W pętli głównej `mainLoop()` zwizualizuj wychylenie czujnika, rysując w pierwszej linii wyświetlacza myślniki odpowiednio po lewej lub prawej stronie znacznika `||` proporcjonalnie do wychylenia. Zakres pracy i rozdzielczość tego bargrafu dobierz wedle uznania.

Rysunek 4: Przykładowy stan wyświetlacza