

Arduino_Fast_LSM6DS3

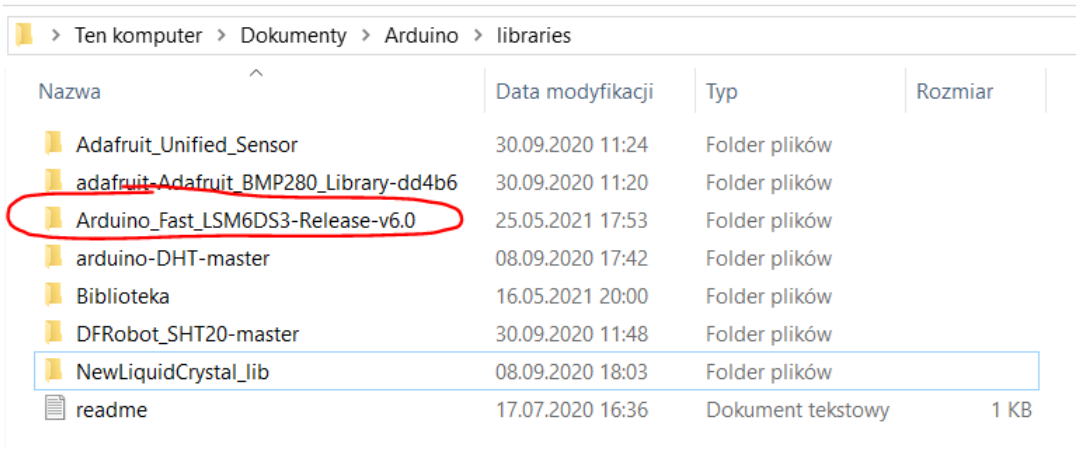
Jakub Kolton

INSTRUKCJA OBSŁUGI BIBLIOTEKI

Niniejsza instrukcja opisuje wykorzystanie biblioteki w środowisku Arduino IDE 2.0 (Beta 5) uruchamianym w systemie operacyjnym Windows 10.

1. Folder z biblioteką (wypakować archiwum) należy w lokalizacji:

C:\Users\<nazwa_użytkownika>\Documents\Arduino\libraries



Ten komputer > Dokumenty > Arduino > libraries				
Nazwa		Data modyfikacji	Typ	Rozmiar
Adafruit_Unified_Sensor		30.09.2020 11:24	Folder plików	
adafruit-Adafruit_BMP280_Library-dd4b6		30.09.2020 11:20	Folder plików	
Arduino_Fast_LSM6DS3-Release-v6.0		25.05.2021 17:53	Folder plików	
arduino-DHT-master		08.09.2020 17:42	Folder plików	
Biblioteka		16.05.2021 20:00	Folder plików	
DFRobot_SHT20-master		30.09.2020 11:48	Folder plików	
NewLiquidCrystal_lib		08.09.2020 18:03	Folder plików	
readme		17.07.2020 16:36	Dokument tekstowy	1 KB

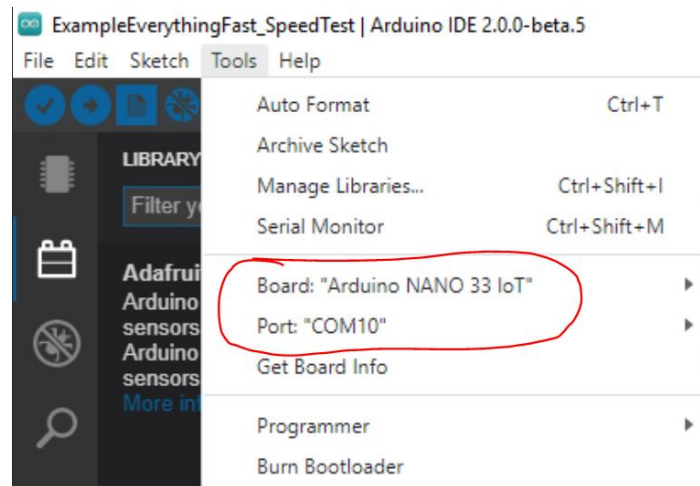
Od teraz pliki biblioteki będą widoczne w środowisku Arduino IDE 2.0.

2. Aby wykorzystać zawartość biblioteki w projekcie, należy załączyć plik nagłówkowy:

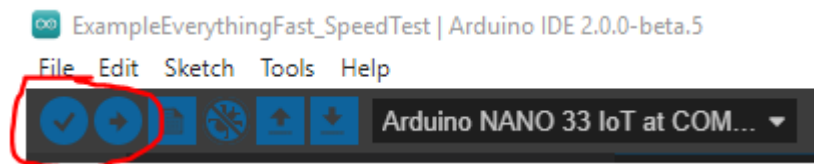
#include "Fast_LSM6DS3.h"

Po jego załączeniu dostępna jest cała funkcjonalność biblioteki.

3. Biblioteka przeznaczona jest dla Arduino Nano 33 IoT. Przed kompilacją i wgraniem programu należy wybrać tę płytkę z listy dostępnych płytek (jeśli jest niedostępna, należy pobrać rozszerzenie *Arduino SAMD Boards*) oraz wybrać port COM, którym jest dołączona do komputera:



4. Po napisaniu programu należy go skompilować lub skompilować i wgrać do podłączonego Arduino Nano 33 IoT z użyciem przycisków:



5. Biblioteka napisana jest obiektowo. Oferuje szereg stałych, pól i metod ułatwiających pracę z czujnikiem LSM6DS3. Część z nich jest prywatna, aby biblioteka stanowiła niezawodny interfejs między czujnikiem a mikrokontrolerem. Z punktu widzenia programisty interfejsem takim jest obiekt klasy *LSM6DS3*, który należy powołać z użyciem konstruktora. Na obiekcie tym wywoływane są wszystkie metody.

6. Zawartość biblioteki:

- Stałe predefiniujące adres I2C czujnika oraz adresy jego rejestrów (linie 15-141 pliku *.h*).
- Stałe predefiniujące poszczególne fragmenty wpisywane do rejestrów kontrolnych, odpowiadające za m.in. częstotliwość pomiarów, ich zakres, konfigurację filtru antyaliasingowego itp. (linie 144-185 pliku *.h*).
- Konstruktor i destruktor obiektu klasy *LSM6DS3*.
- Metody klasy *LSM6DS3*. Większość metod posiada dwie wersje: standardową (pozwalającą na samodzielną konfigurację czujnika i zebranie pomiarów) oraz szybką (konfigurującą czujnik w sposób domyślny, i pozwalającą zebrać pomiary w takich warunkach). Użycie funkcji szybkiego odczytu pomiarów nie jest zalecane w przypadku samodzielnej konfiguracji zakresu pomiarowego. Funkcje standardowe każdorazowo odczytują z rejestru zakres pomiarowy.

7. Opis metod:

Metody odczytu:

Metody te odczytują do parametru/ów jeden z pomiarów. Wyjątkiem jest pierwsza z metod, która zwraca zawartość rejestru.

Metody odczytu:

Metoda	Zastosowanie
<i>int readRegister (uint8_t addr)</i>	Zwraca wartość zapisaną w rejestrze
<i>int readAcceleration (float &x, float &y, float &z)</i>	Odczytuje do parametrów pomiar z akcelerometru (dla dowolnej jego konfiguracji) w jednostkach g
<i>int readAccelerationFast (float &x, float &y, float &z)</i>	Szybko odczytuje do parametrów pomiar z akcelerometru (dla domyślnej jego konfiguracji) w jednostkach g
<i>int readAngular(float &x, float &y, float &z)</i>	Odczytuje do parametrów pomiar z żyroskopu (dla dowolnej jego konfiguracji) w jednostkach %/sek
<i>int readAngularFast(float &x, float &y, float &z)</i>	Szybko odczytuje do parametrów pomiar z żyroskopu (dla domyślnej jego konfiguracji) w jednostkach %/sek
<i>int readTemperature(float &x)</i>	Odczytuje do parametru pomiar z termometru w jednostkach °C

Metody zapisu:

Metody te wpisują do rejestrów pewne wartości – dla metod ustawiających parametry czujnika (metody rodzin *set...*, *begin...*) przygotowane są predefiniowane stałe ułatwiające korzystanie z nich. Wyjątkiem jest pierwsza z metod, która pozwala wpisywać wartości do dowolnych rejestrów, które nie są na liście rejestrów zabronionych (*Reserved*).

Metoda	Zastosowanie
<i>int writeRegister (uint8_t addr, uint8_t value)</i>	Wpisuje podaną wartość do podanego rejestru (o ile wolno do niego wpisywać)
<i>int setFreq_XL(uint8_t value)</i>	Ustawia ODR (częstotliwość wysyłania pomiarów) akcelerometru
<i>int setScale_XL(uint8_t value)</i>	Ustawia FS (zakres pomiarowy) akcelerometru
<i>int setBandwith_XL(uint8_t value)</i>	Ustawia BW (pasmo filtru anty-aliasingowego) akcelerometru
<i>int setFreq_G(uint8_t value)</i>	Ustawia ODR (częstotliwość wysyłania pomiarów) żyroskopu
<i>int setScale_G(uint8_t value)</i>	Ustawia FS (zakres pomiarowy) żyroskopu
<i>int beginFast()</i>	Szybko inicjuje czujnik z jego domyślnymi ustawieniami.
<i>int begin(uint8_t ODR_XL, uint8_t FS_XL, uint8_t BW_XL, uint8_t ODR_G, uint8_t FS_G)</i>	Inicjuje czujnik z ustawieniami podanymi jako parametry.
<i>void end()</i>	Wyłącza czujnik.

8. Prócz plików źródłowego i nagłówkowego, w katalogu *Examples* przygotowane są przykładowe projekty w formacie *.ino* korzystające z biblioteki:

📁 > Arduino_Fast_LSM6DS3-Release-v6.0 > Examples			
Nazwa	Data modyfikacji	Typ	Rozmiar
📁 ExampleAccelerationCustom	13.06.2021 18:51	Folder plików	
📁 ExampleAccelerationFast	13.06.2021 18:51	Folder plików	
📁 ExampleEverythingCustom	13.06.2021 18:51	Folder plików	
📁 ExampleGyroscopeCustom	13.06.2021 18:51	Folder plików	
📁 ExampleGyroscopeFast	13.06.2021 18:51	Folder plików	
📁 ExampleTemperature	13.06.2021 18:51	Folder plików	

Można wykorzystać je do lepszego zrozumienia struktury i działania biblioteki, szybkiego uruchomienia pomiarów lub sprawdzenia polecanego sposobu użycia poszczególnych funkcjonalności.