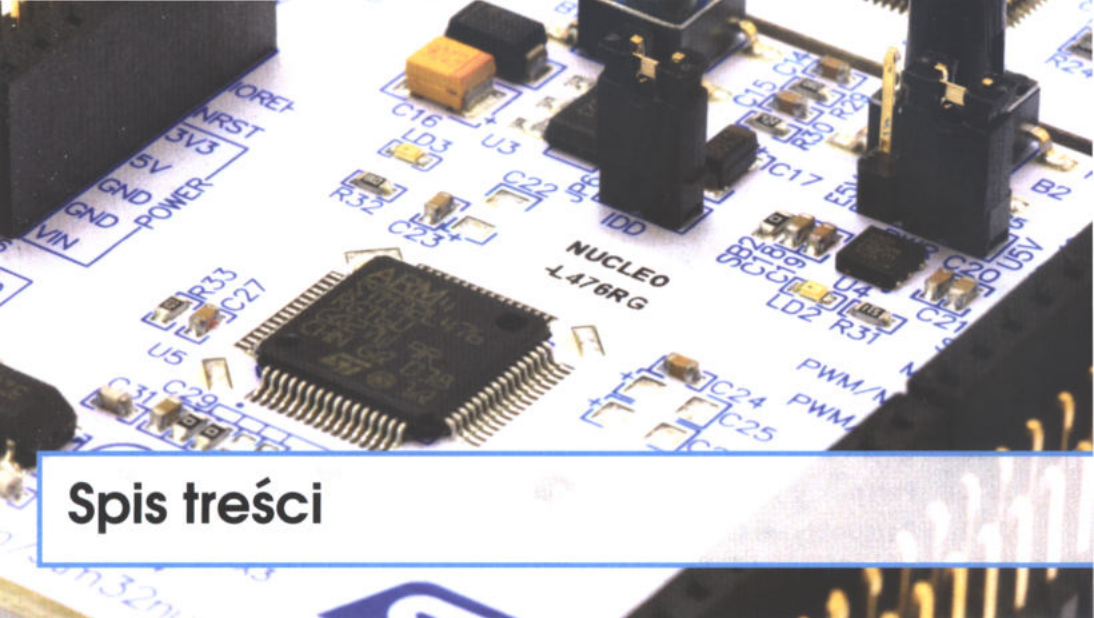




Spis treści

1	Dla kogo jest ten kurs?	5
1.1	Czego nauczysz się z kursu STM32L4?	5
1.1.1	Jak wygląda kurs?	6
1.2	Dla kogo jest kurs STM32L4?	6
1.2.1	Co już trzeba umieć?	7
1.3	Jakie elementy do wykonania ćwiczeń?	7
1.3.1	Kieszonkowe ściągki do kursu STM32L4	8
1.4	Dlaczego (kolejny) kurs o STM32?	8
1.5	Czy warto uczyć się z kursów Forbota?	8
1.5.1	Kto jest autorem tego kursu?	9
1.6	Nasza współpraca z firmą STMicroelectronics	9
1.7	Ważne: aktualizacje, kody programów i grafiki	10
1.8	Podsumowanie – jak zacząć naukę z kursu STM32L4?	10
2	Platforma sprzętowa i środowisko	11
2.1	Czego dowiesz się z tej części kursu STM32L4?	11
2.2	Dlaczego STM32?	11
2.3	Jak odnaleźć się w świecie STM32?	12
2.3.1	Czym jest ARM?	13
2.3.2	Architektury i rdzenie ARM	13
2.3.3	Różne rodziny mikrokontrolerów	14
2.4	Płytki uruchomieniowa NUCLEO-L476RG	15
2.5	Mikrokontroler STM32L476RG	17

2.6	Dokumentacja techniczna dla STM32L4	18
2.6.1	Dokumentacja płytki Nucleo	18
2.6.2	Dokumentacja mikrokontrolera STM32L476RG	19
2.6.3	Dokumentacja bibliotek i programów	20
2.7	Instalacja środowiska STM32CubeIDE	20
2.8	Wstępna konfiguracja STM32CubeIDE	23
2.8.1	Uwaga: interfejsy graficzne mogą się zmieniać!	23
2.9	Pierwszy projekt w STM32CubeIDE	24
2.10	Czym są widoki i perspektywy w STM32CubeIDE?	27
2.11	Perspektywa Device Configuration Tool	27
2.12	Perspektywa C/C++ w STM32CubeIDE	28
2.13	Kompilowanie programu w STM32CubeIDE	30
2.14	Perspektywa Debug w STM32CubeIDE	30
2.14.1	Błąd: nie wykryto programatora	31
2.14.2	Informacja: możliwość aktualizacji programatora	32
2.14.3	Sukces: przejście do perspektywy Debug	32
2.15	Zadania domowe	35
2.16	Podsumowanie – co powinieneś zapamiętać?	35
3	GPIO w praktyce	37
3.1	Czego dowiesz się z tej części kursu STM32L4?	37
3.2	Uniwersalne wejścia/wyjścia – GPIO w praktyce	38
3.2.1	Jak oznaczone są GPIO w STM32?	38
3.3	Pierwsze kroki z Device Configuration Tool	38
3.3.1	Omówienie interfejsu CubeMX	39
3.3.2	Konfiguracja pinu jako wyjście	42
3.3.3	Nadawanie pinom nazw (etykiet)	42
3.3.4	Generowanie kodu	43
3.4	Budowa i zawartość projektu	44
3.5	Zmiana stanu pinu na przeciwny	46
3.6	Wprowadzanie prostych opóźnień do programu	47
3.7	Podpowiadanie składni	48
3.8	Kompilacja programu	48
3.9	Konfiguracja pinu jako wejście	49
3.9.1	Odczytywanie stanu wejścia cyfrowego	50
3.10	Podłączenie zewnętrznej diody	51
3.11	Długi dodatek (?) na temat linijki LED	53
3.11.1	Linijka LED, czyli używanie wielu (różnych) wyjść	54
3.11.2	Własne funkcje pomocnicze	57

3.11.3	Diody podłączone do różnych portów	59
3.11.4	Używanie tablic zamiast duplikacji kodu	60
3.11.5	Używanie struktury zamiast dwóch tablic	61
3.11.6	Sterowanie linijką LED za pomocą przycisku	62
3.11.7	Podłączanie własnego przycisku	66
3.11.8	Drgania styków	69
3.12	O co chodzi z tym generowaniem kodu?	71
3.13	Zadanie domowe	72
3.14	Podsumowanie – co powinieneś zapamiętać?	72
4	Komunikacja UART, debugger	73
4.1	Czego dowiesz się z tej części kursu STM32L4?	73
4.2	Czym jest RS-232, UART i USART?	73
4.3	STM32 i UART – pierwszy projekt	75
4.3.1	Ręczna konfiguracja pinów debuggera	75
4.3.2	Konfiguracja UART-a na STM32	76
4.3.3	Wysyłanie danych przez UART	77
4.3.4	Uruchomienie programu	79
4.4	Odbieranie danych przez Tera Term	80
4.4.1	Terminal wbudowany w STM32CubeIDE	81
4.5	Różnice między funkcjami <i>sizeof</i> oraz <i>strlen</i>	82
4.5.1	Wysyłanie danych z użyciem <i>printf</i>	82
4.5.2	Pułapka nr 1: znaki końca linii	84
4.5.3	Pułapka nr 2: buforowanie	87
4.5.4	Obsługa liczb <i>float</i> przez funkcję <i>printf</i>	88
4.6	Odbieranie danych przez UART	90
4.6.1	Timeout i blokowanie programu	93
4.6.2	Odbieranie linii danych	93
4.7	Programy nie zawsze są bezbłędne	95
4.7.1	Pierwsze prawdziwe wykorzystanie debuggera	97
4.8	Zadanie domowe	102
4.9	Podsumowanie – co powinieneś zapamiętać?	102
5	Taktowanie, RTC, watchdog	103
5.1	Czego dowiesz się z tej części kursu STM32L4?	103
5.2	O co chodzi z taktowaniem układów?	103
5.3	Jakie są typy sygnałów zegarowych?	104
5.3.1	Jaki jest cel różnych źródeł taktowania?	105
5.3.2	Kiedy przydają się dwa źródła sygnałów zegarowych?	106

5.4	Jakie są źródła sygnałów zegarowych?	106
5.4.1	Zewnętrzny generator sygnału zegarowego	107
5.4.2	Rezonator kwarcowy	108
5.4.3	Generator RC	108
5.5	Dokładność źródeł sygnału zegarowego	109
5.5.1	Dokładność kwarcu zegarkowego	109
5.5.2	Dokładność generatora RC	110
5.6	Konfiguracja zegarów w CubeMX	110
5.7	Włączenie RTC	111
5.7.1	Jak ustawić czas zegara RTC?	115
5.7.2	Użycie rezonatora kwarcowego dla RTC	117
5.7.3	Odczytywanie daty z RTC	118
5.7.4	Zapisywanie daty do RTC	119
5.7.5	Ustawianie i podtrzymywanie zegara RTC	119
5.8	Autokalibracja MSI	120
5.9	Główny zegar systemowy	121
5.10	Watchdog – kontrola działania układu	123
5.11	Quiz – sprawdź, ile już wiesz!	127
5.12	Zadanie domowe	127
5.13	Podsumowanie – co powinieneś zapamiętać?	128
6	Oszczędzanie energii	129
6.1	Czego dowiesz się z tej części kursu STM32L4?	129
6.1.1	Ta część kursu jest trochę inna...	130
6.2	Krótki wstęp teoretyczny: powtórka z fizyki	130
6.2.1	Moc elektryczna i praca	130
6.2.2	Pojemność akumulatorów	131
6.2.3	Zmienność poboru prądu w czasie	132
6.2.4	Podejście praktyczne	132
6.3	STM32L4, czyli L jak <i>ultra-low-power</i>	133
6.4	Praktyczny eksperyment z oszczędzaniem energii	134
6.4.1	Program testowy – symulacja czujnika	136
6.5	Uruchamianie programów bez debuggera	136
6.5.1	Ile prądu pobiera ten program?	137
6.5.2	Pomiar prądu samego mikrokontrolera na Nucleo	138
6.5.3	Zmiana częstotliwości pomiarów	140
6.5.4	Uspianie mikrokontrolera, gdy nic nie robi	140
6.5.5	Wyłączanie niepotrzebnych peryferii	142
6.5.6	Obniżanie częstotliwości taktowania	143
6.5.7	Zmiany napięcia zasilania?	146
6.5.8	Pełne uspianie mikrokontrolera	147

6.6	Uwzględnianie czasu pomiaru	150
6.7	PCC – rozbudowany kalkulator zużycia energii	150
6.7.1	Konfiguracja projektu w Power Consumption Calculator	151
6.7.2	Ciekawostka: dokładniejsze pomiary	155
6.8	Zadanie domowe	157
6.9	Podsumowanie – co powinienś zapamiętać?	157
7	Przerwania i obsługa błędów	159
7.1	Czego dowiesz się z tej części kursu STM32L4?	159
7.2	Czym są przerwania?	159
7.3	Obsługa przerw sprzętowych	160
7.4	Miganie diodą w przerwaniu	163
7.5	Jak działają przerwania?	165
7.5.1	Czym są zdarzenia?	165
7.5.2	Czym są przerwania i wyjątki?	165
7.5.3	Procedury obsługi przerw	165
7.5.4	Obsługa błędów	167
7.6	Przerwania od przycisku	168
7.6.1	Licznik naciśnięć przycisku	170
7.6.2	Więcej opcji związanych z przerwaniami	172
7.7	Niepoprawna wersja z <i>printf</i> w przerwaniu	174
7.7.1	Priorytety przerw	177
7.7.2	Zmiana priorytetu przerw	179
7.7.3	Obsługa UART-a w przerwaniach	180
7.8	Wysyłanie danych z użyciem przerw	181
7.8.1	Sprawdzanie komunikatu błędów	181
7.8.2	Ukrywanie problemów	184
7.8.3	Informacja o zakończeniu transmisji	184
7.8.4	Wysyłanie komunikatów w przerwaniu	185
7.8.5	Bardzo prosty automat skończony	186
7.8.6	Wysyłanie komunikatów w przerwaniu, ulepszona wersja	187
7.8.7	Ulepszanie kodu	189
7.8.8	Problemy z buforem pamięci	189
7.8.9	Odbieranie danych w przerwaniach	191
7.9	Zadanie domowe	194
7.10	Podsumowanie – co powinienś zapamiętać?	194
8	Liczniki sprzętowe (PWM, enkoder)	195
8.1	Czego dowiesz się z tej części kursu STM32L4?	195
8.2	Czym w mikrokontrolerach są liczniki?	195

8.3	Liczniki dostępne w STM32L476RG	196
8.4	Liczniki podstawowe	197
8.4.1	Generowanie cyklicznych przerwań	198
8.4.2	Obliczenie preskalera i okresu	198
8.4.3	Wykorzystanie przerwania od licznika	200
8.5	Liczniki ogólnego zastosowania	201
8.5.1	Generowanie cyklicznych przerwań	203
8.5.2	Wykorzystanie wielu kanałów licznika	204
8.5.3	Generowanie sygnału PWM	207
8.5.4	Zmiana częstotliwości sygnału PWM	209
8.5.5	Sterowanie diodą RGB	210
8.6	Obsługa enkodera obrotowego	213
8.6.1	Zliczanie impulsów za pomocą licznika	214
8.6.2	Pomiar szerokości impulsu za pomocą licznika	218
8.6.3	Tryb obsługi enkoderów w STM32L4	221
8.7	Zadanie domowe	224
8.8	Podsumowanie – co powinieneś zapamiętać?	224
9	Przetworniki analogowo-cyfrowe	225
9.1	Czego dowiesz się z tej części kursu STM32L4?	225
9.2	Przetworniki analogowo-cyfrowe w STM32L476RG	225
9.3	Pierwszy pomiar przez ADC	226
9.3.1	Konfiguracja projektu z ADC	226
9.3.2	Kod programu korzystającego z ADC	228
9.3.3	Jak interpretować wynik z ADC?	230
9.4	Pomiar napięcia zewnętrznego przez STM32	231
9.4.1	Ile czasu zajmuje pomiar ADC?	235
9.5	Ciągły odczyt z przetwornika ADC	235
9.5.1	Odczyt z wielu kanałów ADC	237
9.6	Pomiary przy użyciu DMA	240
9.6.1	Pałapki pracy z DMA	243
9.6.2	Oversampling, czyli filtrowanie pomiarów	244
9.7	Obsługa joysticka na STM32L4	245
9.8	Rysowanie wykresów w STM Studio	247
9.9	Inne moduły analogowe w STM32L4	251
9.9.1	Wykorzystanie komparatora z STM32L4	252
9.10	Zadanie domowe	255
9.11	Podsumowanie – co powinieneś zapamiętać?	255

10	Ekspander portów (SPI)	257
10.1	Czego nauczysz się z tej części kursu STM32L4?	257
10.2	Szeregowe interfejsy komunikacyjne	257
10.2.1	SPI vs. UART – najważniejsze różnice	258
10.3	Komunikacja przez SPI	260
10.3.1	Konfiguracja SPI na STM32L4	260
10.3.2	Programowe sterowanie linią CS	262
10.3.3	Podłączenie układu	263
10.3.4	Wysyłanie i odbieranie danych przez SPI	264
10.4	Omówienie rejestrów MCP23S08	264
10.5	Komunikacja z MCP23S08	267
10.5.1	Sterowanie diodą za pomocą ekspandera	267
10.5.2	Obsługa przycisku za pomocą ekspandera	270
10.6	Zmiana struktury projektu CubeMX	273
10.7	Wydzielanie kodu do własnej biblioteki	274
10.8	Quiz – sprawdź, ile już wiesz!	276
10.9	Zadanie domowe	277
10.10	Podsumowanie – co powinieneś zapamiętać?	277
11	Kolorowy wyświetlacz TFT (SPI)	279
11.1	Czego dowiesz się z tej części kursu STM32L4?	279
11.2	Programowanie hobbystyczne vs. zawodowe	279
11.3	Wyświetlacz graficzny TFT ze sterownikiem ST7735S	280
11.3.1	Podłączenie wyświetlacza	281
11.3.2	Konfiguracja projektu z wyświetlaczem	281
11.4	Własna biblioteka dla wyświetlacza TFT	283
11.4.1	Inicjalizacja wyświetlacza	283
11.4.2	Rysowanie prostokątów	290
11.4.3	Włączanie pojedynczych pikseli	293
11.4.4	Wyświetlanie obrazów	294
11.4.5	Optymalizacja rysowania	298
11.4.6	Buforowanie obrazu	299
11.4.7	Nowa wersja biblioteki	302
11.5	Wykorzystanie gotowej biblioteki graficznej	304
11.5.1	Dodanie brakujących plików	306
11.5.2	Program demonstracyjny	307
11.6	Użycie pamięci	308
11.7	Wykorzystanie DMA	311
11.7.1	Programy demonstracyjne	314
11.8	Zadanie domowe	316

11.9	Podsumowanie – co powinieneś zapamiętać?	316
12	Zewnętrzna pamięć EEPROM	317
12.1	Czego dowiesz się z tej części kursu STM32L4?	317
12.2	Czym jest I2C?	317
12.2.1	Dlaczego potrzebne są dodatkowe rezystory?	318
12.2.2	Jak przebiega komunikacja przez I2C?	321
12.3	Zewnętrzna pamięć EEPROM	321
12.3.1	Podłączenie pamięci EEPROM do STM32	322
12.3.2	Zapis do pamięci EEPROM – teoria	322
12.3.3	Zapis do pamięci EEPROM – praktyka	323
12.3.4	Odczyt z pamięci EEPROM	326
12.3.5	Kontrola transmisji za pomocą analizatora	327
12.3.6	Czas zapisu i obsługa błędów	329
12.4	Własna biblioteka do obsługi pamięci EEPROM	333
12.5	Licznik uruchomień układu	335
12.6	Zadanie domowe	336
12.7	Podsumowanie – co powinieneś zapamiętać?	336
13	Czujnik ciśnienia (I2C)	337
13.1	Czego dowiesz się z tej części kursu STM32L4?	337
13.2	Dlaczego czujniki ciśnienia nie są nudne?	338
13.3	LPS25HB – czujnik ciśnienia i temperatury	338
13.4	Komunikacja z czujnikiem LPS25HB	340
13.4.1	Test komunikacji z czujnikiem LPS25HB	343
13.4.2	Odczyt temperatury z czujnika LPS25HB	344
13.4.3	Pomiar ciśnienia za pomocą czujnika LPS25HB	347
13.4.4	Biblioteka dla czujnika LPS25HB	348
13.5	Interpretacja pomiarów ciśnienia	351
13.5.1	Jednostki ciśnienia atmosferycznego	351
13.5.2	Wzór barometryczny	352
13.5.3	Ciśnienie względne, bezwzględne i prognoza pogody	353
13.5.4	Przeliczanie ciśnienia na inne jednostki	353
13.5.5	Pomiar ciśnienia względnego	354
13.5.6	Kalibracja czujnika	356
13.6	Pomiar wysokości bezwzględnej	357
13.7	Pomiar wysokości względnej	359
13.7.1	Uśrednianie wyników z czujnika LPS25HB	360
13.8	Zadanie domowe	363
13.9	Podsumowanie – co powinieneś zapamiętać?	363

14	Dalmierz, wyświetlacz 7-seg.	365
14.1	Czego dowiesz się z tej części kursu STM32L4?	365
14.2	HC-SR04 – ultradźwiękowy czujnik odległości	365
14.3	Podłączenie HC-SR04 do STM32L4	367
14.3.1	Współpraca STM32L4 z układami zasilanymi z 5 V	367
14.3.2	Obsługa czujnika HC-SR04 za pomocą liczników	368
14.3.3	Sprzętowy pomiar obu zboczy sygnału	372
14.4	Obsługa wyświetlaczy 7-segmentowych	373
14.4.1	Wyświetlanie jednej cyfry	374
14.4.2	Wyświetlanie dwóch cyfr	379
14.4.3	Wyświetlanie pomiarów na wyświetlaczu	382
14.5	Prędkość dźwięku w powietrzu?	384
14.5.1	LM35 – analogowy czujnik temperatury	384
14.5.2	Podłączenie i test czujnika	385
14.6	Wzmacniacz operacyjny	387
14.6.1	Pomiar napięcia wyjściowego wzmacniacza	389
14.7	Zadanie domowe	391
14.8	Podsumowanie – co powinieneś zapamiętać?	391
15	Diody RGB WS2812B (liczniki)	393
15.1	Czego dowiesz się z tej części kursu STM32L4?	393
15.2	WS2812 – cyfrowe diody RGB	393
15.2.1	Zasilanie diod WS2812B	396
15.2.2	Metody konwersji poziomów logicznych	396
15.3	Podłączenie diod WS2812B do STM32L4	399
15.3.1	Komunikacja z diodami WS2812B	400
15.3.2	Konfiguracja projektu	401
15.3.3	Pierwszy program obsługujący diody WS2812B	402
15.3.4	Optymalizacja kodu do obsługi WS2812B	405
15.3.5	Korekcja gamma	408
15.4	Dla zainteresowanych używaniem PWM oraz DMA	410
15.5	Quiz – sprawdź, ile już wiesz!	413
15.6	Zadanie domowe	414
15.7	Podsumowanie – co warto zapamiętać?	414
16	Zdalne sterowanie IR (NEC)	415
16.1	Czego dowiesz się z tej części kursu STM32L4?	415
16.2	Zdalna komunikacja za pomocą IR	415
16.2.1	Pilot IR	416

16.2.2	Protokół NEC	417
16.2.3	Scalony odbiornik podczerwieni	419
16.3	Praktyczny test odbiornika VS1838B	421
16.3.1	Analiza odebranych danych	421
16.4	Dekodowanie NEC za pomocą STM32L4	424
16.4.1	Dekodowanie impulsów	427
16.4.2	Uprozczone dekodowanie poleceń	431
16.4.3	Wykrywanie przytrzymania przycisków	433
16.4.4	Biblioteka do dekodowania NEC	434
16.5	Lampka RGB – WS2812B sterowane pilotem	437
16.6	Zadanie domowe	439
16.7	Podsumowanie – co warto zapamiętać?	439
17	Termometry DS18B20 (1-wire)	441
17.1	Czego dowiesz się z tej części kursu STM32L4?	441
17.2	DS18B20 – czujnik temperatury	441
17.3	Protokół 1-wire	442
17.3.1	Sekwencja reset w 1-wire	443
17.3.2	Odczyt danych przez 1-wire	443
17.3.3	Wysyłanie danych (kodowanie bitów) przez 1-wire	444
17.4	Podłączenie DS18B20 do STM32L4	445
17.5	Komunikacja STM32L4 z DS18B20	445
17.5.1	Programowy 1-wire na STM32L4 (bit-banging)	446
17.5.2	Odczyt adresu czujnika DS18B20	448
17.5.3	Sprawdzanie sumy kontrolnej DS18B20	450
17.5.4	Odczyt zawartości pamięci DS18B20	451
17.5.5	Pomiar temperatury DS18B20	453
17.6	Biblioteka do obsługi 1-wire	454
17.7	Biblioteka do obsługi DS18B20	456
17.8	Wykorzystanie bibliotek do 1-wire i DS18B20	460
17.9	Odczyt z wielu DS18B20	462
17.10	Opóźnienia, czyli kiedy pojawią się problemy	464
17.10.1	Wyłączanie przerwań (styl Arduino)	464
17.10.2	Wyłączanie przerwań (styl ARM)	466
17.10.3	Użycie modułu USART do komunikacji po 1-wire?	466
17.11	Implementacja 1-wire przez UART	468
17.11.1	Udoskonalona obsługa 1-wire przez UART (Half-Duplex)	472
17.12	Zadanie domowe	473
17.13	Podsumowanie – co powinieneś zapamiętać?	473

18	Podsumowanie kursu, quiz	475
18.1	Zadanie 1: Oceń ten kurs	475
18.2	Zadanie 2: Wybierz kolejny kurs	476
18.3	Zadanie 3: Sprawdź się w quizie	476
18.4	Gratulacje!	476