

Sprawozdanie z projektu

Realizowanego w ramach przedmiotu *Zaawansowane Architektury Procesorów*

Temat projektu:

Projekt z użyciem płytki STM32L4R9I-DISCOVERY, korzystający z okrągłego wyświetlacza i mechanizmów jego obsługi – zegarek.

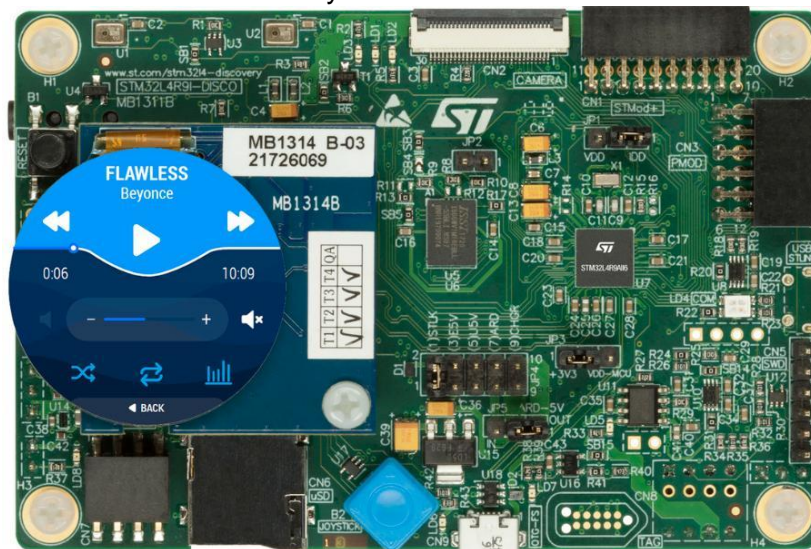
Cele projektu:

Wykorzystanie okrągłego wyświetlacza płytki STM32L4R9I-DISCOVERY do stworzenia aplikacji zegarka wskazującego czas rzeczywisty. Praktyczne wykorzystanie wyświetlacza płytki oraz dostępnych narzędzi programistycznych i projektowych. Zapoznanie z architekturą płytek STM32.

Wykorzystany sprzęt:

Do realizacji projektu wykorzystano:

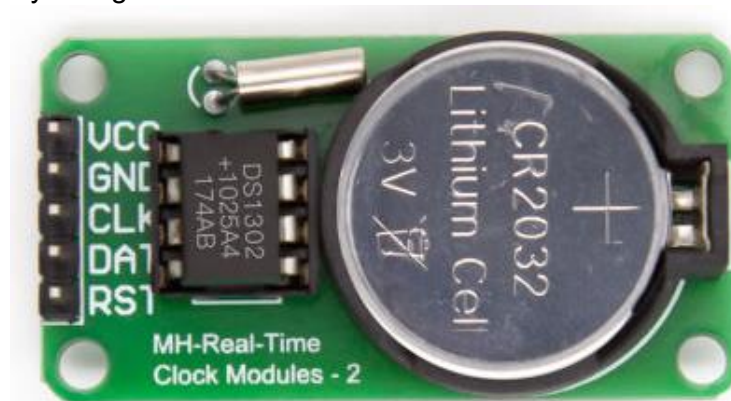
1. Płytkę STM32L4R9I-DISCOVERY firmy STMicroelectronics



Źródło grafiki: strona producenta www.st.com

Najważniejsze cechy wykorzystanej płytki:

- procesor ARM® Cortex®-M4 taktowany zegarem 120 MHz i wykonujący maksymalnie 150 milionów operacji na sekundę (150 MIPS)
 - dotykowy wyświetlacz LCD o przekątnej 1,2 cala i rozdzielczości 390x390 pikseli
 - 640 KB statycznej pamięci RAM i 2 MB pamięci Flash
2. Moduł czasu rzeczywistego RTC DS1302



Źródło grafiki: www.kuongshun-ks.com

Najważniejsze cechy wykorzystanego zegara czasu rzeczywistego:

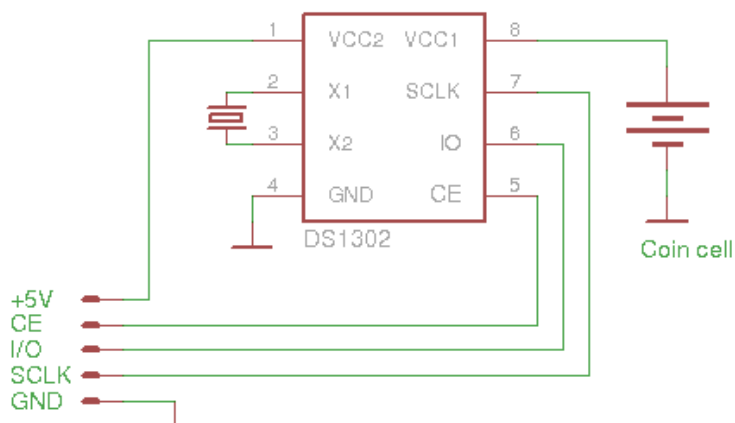
- zasilanie napięciem od 2.0V do 5.5V (wykorzystano baterię CR2032 o napięciu 3V)
- zużycie prądu na poziomie niższym niż 300 nA przy napięciu zasilania 2V
- pięć wyprowadzeń do komunikacji z płytą, połączenia zrealizowane z pomocą kabli F/F oraz M/M

Wykorzystane narzędzia programistyczne i projektowe:

1. EWARM (*Embedded Workbench for ARM*) 8.11.1 firmy IAR Systems – zintegrowane środowisko programistyczne (*IDE*) dla rozwijania i debugowania aplikacji dla sprzętu opartego o rdzenie ARM®, wykorzystane na mocy licencji ewaluacyjnej.
Odnosnik do strony poświęconej EWARM na stronie producenta:
www.iar.com/iar-embedded-workbench/
2. TouchGFX firmy STMicroelectronics – bezpłatne narzędzie do tworzenia aplikacji z graficznym interfejsem użytkownika dla mikrokontrolerów STM32. Pozwala na wygenerowanie kodu w języku C++ dla dalszego rozwijania w wybranych środowiskach programistycznych, w tym w EWARM.
Odnosnik do strony poświęconej TouchGFX na stronie producenta:
www.st.com/en/development-tools/touchgfxdesigner.html
3. CubeMX firmy STMicroelectronics – narzędzie graficzne umożliwiające początkową konfigurację pinów mikrokontrolerów STM32, oraz wygenerowanie kodu w języku C++ dla dalszego rozwijania w wybranych środowiskach programistycznych, w tym w EWARM.
Odnosnik do strony poświęconej CubeMX na stronie producenta:
<https://www.st.com/en/development-tools/stm32cubemx.html>
4. STM32 ST-LINK firmy STMicroelectronics – bezpłatne, nieskomplikowane narzędzie do programowania pamięci (Flash, RAM) mikrokontrolerów STM32 oraz czyszczenia zawartości pamięci urządzenia. Odnosnik do strony poświęconej STM32 ST-LINK na stronie producenta:
<https://www.st.com/en/development-tools/stsw-link004.html>

Przebieg prac projektowych:

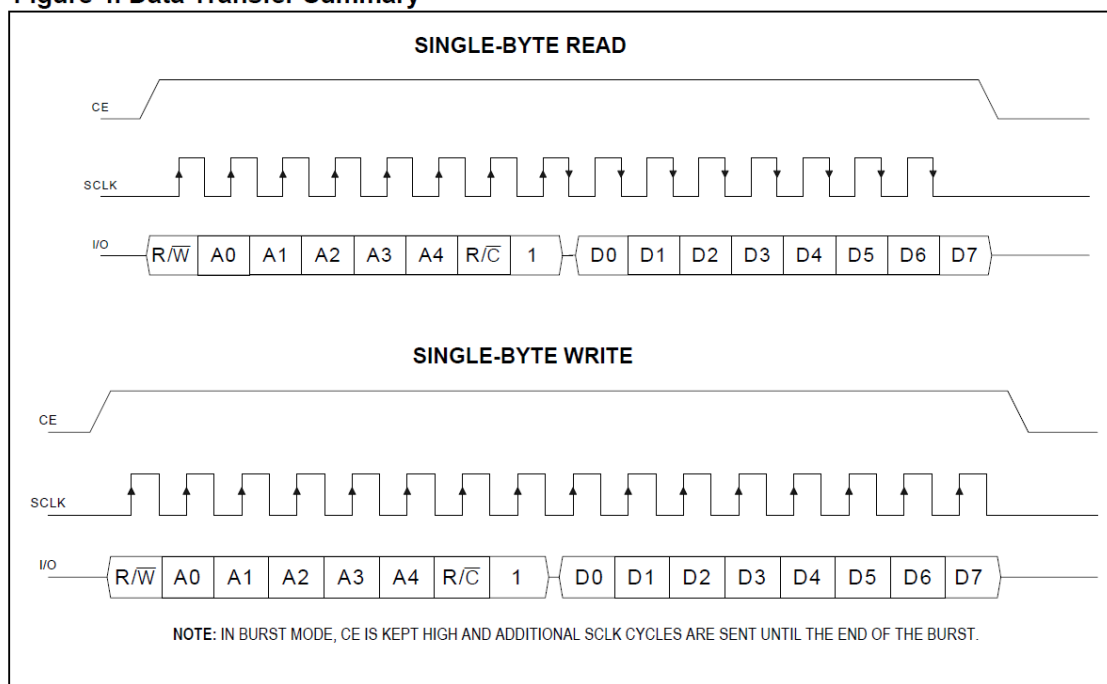
Z uwagi na chęć wykorzystania wymienionych wyżej narzędzi, pierwszą dużą trudnością już na początkowym etapie realizacji projektu była odpowiednia konfiguracja narzędzi i znalezienie sposobu na wykorzystanie plików wynikowych jednego narzędzia do dalszego rozwijania w innym narzędziu. Największym kłopotem okazało się wykorzystanie plików projektu z interfejsem graficznym, wygenerowanych przez program *TouchGFX*, do dalszego rozwijania w środowisku *EWARM* i debugowanie zmodyfikowanej aplikacji. Po znalezieniu rozwiązania tego problemu i uruchomieniu prostych przykładów dla potwierdzenia poprawności rozwiązania można było przystąpić do implementacji zasadniczej części rozwiązania – komunikacji płytki z zegarem czasu rzeczywistego. Niezbędna była w tym celu dokumentacja modułu zegara czasu rzeczywistego DS1302. Jako podstawa do zrealizowania fizycznego połączenia płytki STM32L4R9I-Discovery z modułem DS1302 posłużył schemat przedstawiony na Zdjęciu 1.



Zdjęcie 1. Schemat połączenia płytki z modulem DS1302 (źródło: www.kuongshun-ks.com)

Aby móc poprawnie komunikować się z zegarem czasu rzeczywistego, konieczne było przeanalizowanie operacji odczytu oraz zapisu pojedynczego bajtu do RTC. Podsumowanie tych operacji przedstawia Zdjęcie 2.

Figure 4. Data Transfer Summary



Zdjęcie 2. Operacje transferu danych między RTC i płytką (źródło: <https://datasheets.maximintegrated.com/en/ds/DS1302.pdf>)

Dla właściwej interpretacji wyników otrzymanych odczytów z RTC, konieczna jest znajomość znaczenia bitów przy wykonywaniu odpowiednich operacji, takich jak na przykład odczyt minut z modułu zegara. Tabela obrazująca te własności jest przedstawiona na Zdjęciu 3.

RTC

READ	WRITE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	RANGE
81h	80h	CH	10 Seconds			Seconds				00–59
83h	82h		10 Minutes			Minutes				00–59
85h	84h	12/24	0	10	Hour	Hour				1–12/0–23
				AM/PM						
87h	86h	0	0	10 Date		Date				1–31
89h	88h	0	0	0	10 Month	Month				1–12
8Bh	8Ah	0	0	0	0	0	Day			1–7
8Dh	8Ch	10 Year				Year				00–99
8Fh	8Eh	WP	0	0	0	0	0	0	0	—
91h	90h	TCS	TCS	TCS	TCS	DS	DS	RS	RS	—

Zdjęcie 3. Znaczenie odpowiednich bitów wyniku przy wykonywaniu operacji odczytu/zapisu z/do RTC (źródło: <https://datasheets.maximintegrated.com/en/ds/DS1302.pdf>)

Źródła wykorzystywane podczas realizacji projektu:

1. Dokumentacja płytki STM32L4R9I-Discovery, dostępna na stronie producenta: <https://www.st.com/en/evaluation-tools/32l4r9idiscovery.html>
2. Dokumentacja zastosowanego zegara czasu rzeczywistego DS1302: <https://datasheets.maximintegrated.com/en/ds/DS1302.pdf>
3. Instrukcje oraz dokumentacje wykorzystanych narzędzi, dostępne na wymienionych wyżej stronach producentów – w szczególności instrukcja obsługi narzędzia TouchGFX.
4. Liczne fora internetowe, pomagające w rozwiązaniu problemów związanych z używanymi narzędziami oraz odnajdywaniu rozwiązań innych napotykanymi trudnościami przy realizacji projektu.