

Niskopoziomowa emulacja sprzętowa implementacja architektury systemowej konsoli Game Boy

Patryk Michalak

16 lutego 2026

Streszczenie

Your abstract.

Spis treści

1 Wstęp	3
1.1 Opis Problemu	3
1.2 Emulatory	3
1.3 Założenia pracy	3
1.4 Grupa docelowa	4
2 Analiza architektury systemowej konsoli Game Boy	4
2.1 Specyfikacja techniczna	4
2.2 Procesor i SoC	4
2.3 Płyta główna	5
2.4 Rewizje	5
3 Some examples to get started	5
3.1 How to create Sections and Subsections	5
3.2 How to include Figures	5
3.3 How to add Tables	5
3.4 How to add Comments and Track Changes	6
3.5 How to add Lists	6
3.6 How to write Mathematics	6
3.7 How to change the margins and paper size	6
3.8 How to change the document language and spell check settings	6
3.9 Good luck!	7

1 Wstęp

1.1 Opis Problemu

Konsola Game Boy została wydana w Japonii 21 kwietnia 1989 roku i od razu odniosła ogromny sukces komercyjny. W ciągu pierwszych dwóch lat sprzedawały się ponad 20 milionów egzemplarzy, co czyniło ją jedną z najlepiej sprzedających się konsoli gier wideo w historii. W 1990 roku Game Boy została wydana na zachodzie i odniosła podobny sukces komercyjny. Konsola była dostępna w kilku różnych barwach i wersjach, co dodatkowo zwiększyło jej popularność. Gry były przekazywane na kartridżach "GamePak." które zawierały kod gry, grafiki i dźwięk a i także dodatkowe element jak dodatkowa pamięć RAM czy bateria do przechowywania stanu gry pomiędzy sesjami.

[[wikipedia-gameboy](#)]

Jednakże, kiedy Nintendo zaprzestała produkcji Game Boya, wiele gier zostało opracowanych tylko dla tej konsoli. Dzisiaj większość tych gier nie jest dostępna na żadnym innym platformie, a gracze muszą szukać innych rozwiązań.

W związku z brakiem dostępności oryginalnej konsoli Game Boya, gracze zaczęli poszukiwać sposobów na zagranie tych klasycznych gier. Jednym z takich rozwiązań są emulatory konsol - oprogramowanie które pozwala na symulację pracy oryginalnej konsoli wirtualnie.

1.2 Emulatory

Emulatory to oprogramowanie które pozwala na symulację pracy oryginalnej konsoli wirtualnie. Są one zdolne do wyświetlania grafiki, generowania dźwięku i obsługi sterowników w sposób podobny do oryginalnego sprzętu.

Emulatory są stosowane głównie przez graczy, którzy chcą zagrać w klasyczne gry bez potrzeby posiadania oryginalnego sprzętu. Są one również używane przez muzea i archiwalia, które pragną zachować treści konsolowej historii w formie cyfrowej.

Emulatory są ważne dla archiwizacji treści z kilku powodów:

- Zachowanie historycznej wartości: Emulatory pozwalają na zachowanie historycznej wartości konsoli i gier, które mogłyby inaczej być utracone.
- Dostępność dla nowej generacji graczy: Emulatory umożliwiają nowej generacji graczom dostęp do klasycznych gier, które mogą nie być dostępne w oryginalnej formie.
- Zachowanie jakości: Emulatory pozwalają na zachowanie jakości grafiki i dźwięku oryginalnej konsoli, co jest ważne dla archiwizacji treści.
- Możliwość dostępu do gier nieopublikowanych: Emulatory umożliwiają dostęp do gier nieopublikowanych lub eksperymentalnych, które mogłyby inaczej pozostać ukryte.

Istnieją dwa rodzaje emulacji:

- Emulacja pełna: Emulacja pełna jest zdolna do symulowania pracy oryginalnej konsoli w całości, w tym wyświetlania grafiki, generowania dźwięku i obsługi sterowników.
- Emulacja częściowa: Emulacja częściowa jest zdolna do symulowania tylko części pracy oryginalnej konsoli, takiej jak grafika lub dźwięk.

1.3 Założenia pracy

Cel pracy to stworzenie emulacji konsoli Game Boy w języku programowania C++, wykorzystującej bibliotekę Raylib do obsługi interfejsu graficznego. Użytkownik posiada dostępne w postaci cyfrowej własne kopie gier.

- Emulacja musi być implementowana w języku programowania C++ i wykorzystać bibliotekę Raylib do obsługi interfejsu graficznego. Muszą być zapewnione mechanizmy kontrolne, by użytkownik mógł sterować grą za pomocą klawiatury.
- Końcowy program musi być dostępny do uruchomienia na różnych platformach, takich jak Windows, macOS i Linux.
- Emulator ma mieć wsparcie dla podstawowego rodzaju cart'a - zawierający jedynie kod gry ROM Only.

- Dostępne w postaci cyfrowej własne rzuty gier Game Boy będą używane do testowania emulacji. Muszą być zapewnione możliwości odtwarzania i grania gier Game Boy z własnymi rzutami gier.

Plan realizacji pracy obejmuje zrozumienie architektury konsoli Game Boy, implementację emulacji w języku programowania C++ oraz wykorzystanie biblioteki Raylib do obsługi interfejsu graficznego. Emulacja musi być zweryfikowana poprzez porównanie wyników z testami tworzonymi przez wierną społeczność dokumentującą konsolę Game Boy.

1.4 Grupa docelowa

Grupa docelowa dla emulatora Game Boy to gracze, którzy chcą ponownie grać w swoje ulubione gry z lat 80. i 90., ale nie mają dostępu do oryginalnej konsoli lub posiadają sposób na skopiowanie danych ze swoich kartridżów bez dostępu do konsoli.

Oni są osobami, które chcą odtworzyć swoje dzieciństwo i ponownie grać w gry, które lubili w młodości. Grupa ta obejmuje również graczy, którzy zakupili cyfrowe gry homebrew i chcieli je zagrać, które bez oryginalnej konsoli i narzędzi do tworzenia własnych kaset z grą.

Emulator Game Boya jest dla nich sposobem na grę w ich ulubione gry bez potrzeby posiadania oryginalnej konsoli lub posiadania specjalistycznych narzędzi.

2 Analiza architektury systemowej konsoli Game Boy

2.1 Specyfikacja techniczna

Nintendo Research And Development 1, pod przewodnictwem Gunpei Yokoi i Satoru Okady, zaprojektowało 8-bitową konsolę Game Boya. Charakteryzuje się wyświetlaczem matrycowym o rozdzielcości 160x144 pikseli, D-padem, czterema przyciskami gry i jednym głośnikiem. Używa stworzonych na potrzebe konsoli kartridżów 'GamePak'. Zasilany był przez 4 baterie AA. Gracze mogli korzystać z kabla Game Link Cable do połączenia dwóch Game Boyów dla rozgrywek wieloosobowej bądź transferu danych dla wspierających gier. Chociaż wyświetlacz monochromatyczny był gorszy niż u konkurencji, umożliwił on bardziej tanie i długotrwałe życie baterii. '[Ref: https://en.wikipedia.org/wiki/Game_Boy,_po_przetłumaczeniu]'

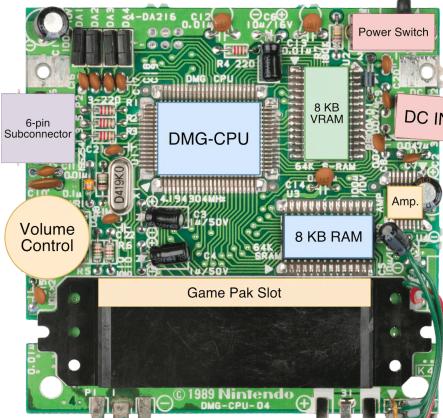
2.2 Procesor i SoC

Procesorem Gameboya był specjalnie wytworzony na potrzeby konsoli 8-bitowy procesor SM83 firmy Sharp. '[Źródło: <https://gekkio.fi/files/gb-docs/gbctr.pdf>]' . Procesor był mieszanką dwóch innych procesorów - 8080 firmy Intel i Z80 firmy Zilog. Procesor znajdował się na płycie wraz innymi komponentami takimi jak pamięć RAM i ROM. Cała płytką jest określana jako SoC (System on Chip) i oryginalny Game Boy zawierał DMG-CPU znany także jako Sharp LR35902 [copetti]. SoC był wpinany do płyty głównej, która zawierała dodatkową pamięć RAM i Video RAM (VRAM).

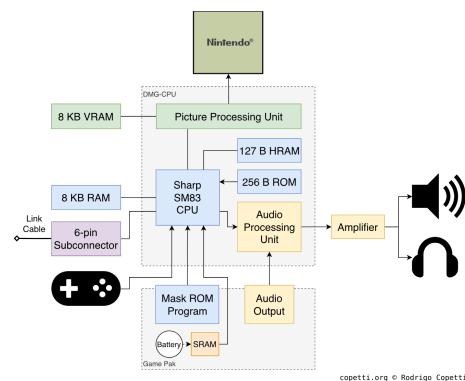
Procesor zawiera osiem 8-bitowych rejestrów: A,B,C,D,E,F,H,L. Rejestry mogą łączyć się w 16-bitowe rejesty: AF, BC, DE, HL . Rejestr F jest używany jako flagi procesora w wypadku operacji arytmetycznych:

- Bit 7 - Zero (Z), jeśli operacja zwróciła zero
- Bit 6 - Negacja (N), używana gdy ostatnia operacją była porównaniem bądź odejmowaniem
- Bit 5 - Half Carry (H), jeśli doszło do przesunięcia na bicie 3
- Bit 4 - Carry (C), jeśli doszło do przesunięcia na bicie 7

2.3 Płyta główna



Rysunek 1: Zdjęcie płyty głównej oryginalnego GameBoy'a z opisami komponentów



Rysunek 2: Schemat płyty głównej konsoli GameBoy

Rysunek 3: Źródło: [copetti]

2.4 Rewizje

3 Some examples to get started

3.1 How to create Sections and Subsections

Simply use the section and subsection commands, as in this example document! With Overleaf, all the formatting and numbering is handled automatically according to the template you've chosen. If you're using the Visual Editor, you can also create new section and subsections via the buttons in the editor toolbar.

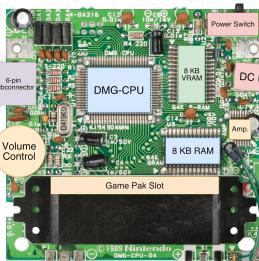
3.2 How to include Figures

First you have to upload the image file from your computer using the upload link in the file-tree menu. Then use the `\includegraphics` command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 4 in this section for an example.

Note that your figure will automatically be placed in the most appropriate place for it, given the surrounding text and taking into account other figures or tables that may be close by. You can find out more about adding images to your documents in this help article on [including images on Overleaf](#).

3.3 How to add Tables

Use the `\begin{table}` and `\end{table}` environments for basic tables — see Table 1, for example. For more information, please see this help article on [tables](#).



Rysunek 4: This frog was uploaded via the file-tree menu.

Item	Quantity
Widgets	42
Gadgets	13

Tabela 1: An example table.

3.4 How to add Comments and Track Changes

Comments can be added to your project by highlighting some text and clicking “Add comment” in the top right of the editor pane. To view existing comments, click on the Review menu in the toolbar above. To reply to a comment, click on the Reply button in the lower right corner of the comment. You can close the Review pane by clicking its name on the toolbar when you’re done reviewing for the time being.

Track changes are available on all our [premium plans](#), and can be toggled on or off using the option at the top of the Review pane. Track changes allow you to keep track of every change made to the document, along with the person making the change.

3.5 How to add Lists

You can make lists with automatic numbering ...

1. Like this,
2. and like this.

...or bullet points ...

- Like this,
- and like this.

3.6 How to write Mathematics

\LaTeX is great at typesetting mathematics. Let X_1, X_2, \dots, X_n be a sequence of independent and identically distributed random variables with $E[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

3.7 How to change the margins and paper size

Usually the template you’re using will have the page margins and paper size set correctly for that use-case. For example, if you’re using a journal article template provided by the journal publisher, that template will be formatted according to their requirements. In these cases, it’s best not to alter the margins directly.

If however you’re using a more general template, such as this one, and would like to alter the margins, a common way to do so is via the geometry package. You can find the geometry package loaded in the preamble at the top of this example file, and if you’d like to learn more about how to adjust the settings, please visit this help article on [page size and margins](#).

3.8 How to change the document language and spell check settings

Overleaf supports many different languages, including multiple different languages within one document.

To configure the document language, simply edit the option provided to the babel package in the preamble at the top of this example project. To learn more about the different options, please visit this help article on [international language support](#).

To change the spell check language, simply open the Overleaf menu at the top left of the editor window, scroll down to the spell check setting, and adjust accordingly.

3.9 Good luck!

We hope you find Overleaf useful, and do take a look at our [help library](#) for more tutorials and user guides! Please also let us know if you have any feedback using the **Contact us** link at the bottom of the Overleaf menu — or use the contact form at <https://www.overleaf.com/contact>.