PDMGR Raport

Jakub Ostrzołek (nr albumu: 310864)

2025-02-02

Osiągnięcia

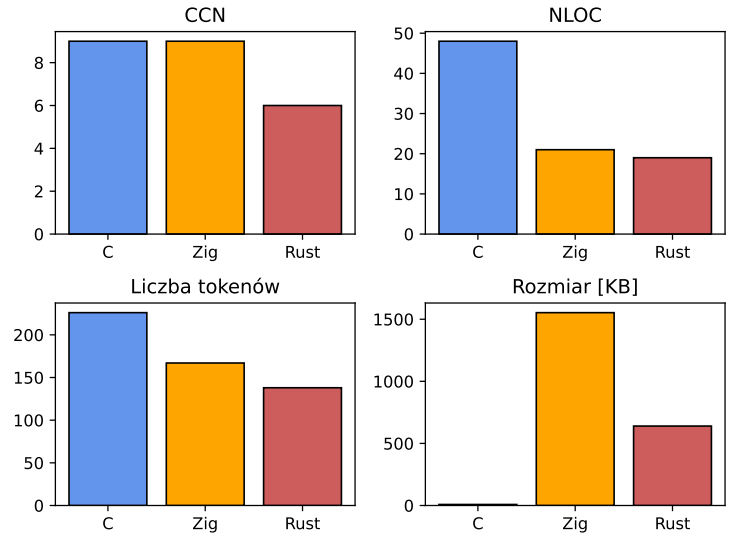
1. Opracowano sposoby mierzenia metryk:
   - czas wykonania: pomiar w kodzie,
   - zużycie pamięci operacyjnej:,
     - Linux → skrypt odpytujący system (do zrobienia),
     - Bare metal → na podstawie analizy pliku wykonywalnego oraz wskaźnika sterty,
   - liczba linii kodu (bez komentarzy): narzędzie `lizard`,
   - całkowita złożoność cyklometryczna funkcji: narzędzie `lizard`,
   - miary kody Helstead: porzucone ze względu na brak dostępnych narzędzi,
   - wielkość pliku wykonywalnego: trywialne.
2. Rozszerzono wspomniane narzędzie `lizard` o wsparcie dla języka *Zig*. Autor narzędzia zaakceptował zmiany.
3. Zamówiono komponenty sprzętowe do realizacji zadań.
4. Zrealizowano w każdym z języków zadanie numer 1 („Hello World'').

Wyniki

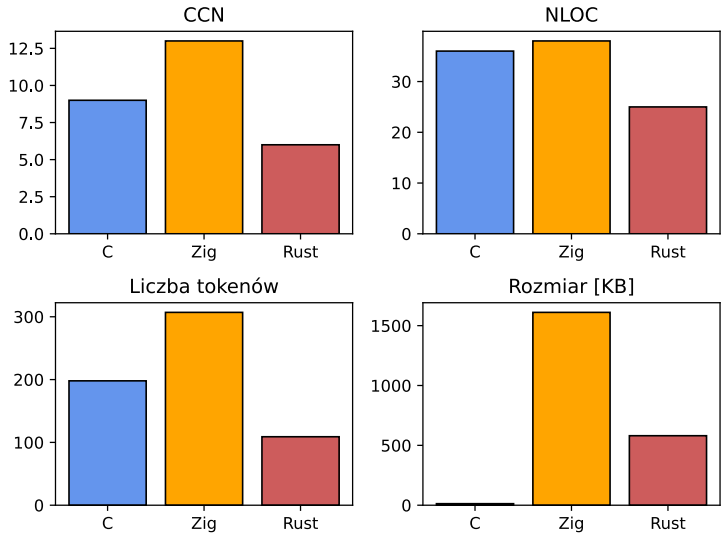
Poniższe wykresy przedstawiają porównanie mierzonych statycznie metryk między językami dla zadań.

Uwaga: Zadanie 2 nie jest na ten moment w pełni skończone, więc końcowe wyniki się zmienią.

Uwaga 2: Pliki języka *Zig* są linkowane z biblioteką standardową c, by umożliwić obsługę sygnałów SIGINT. Zwiększa to zauważalnie rozmiar plików, więc w końcowym porównaniu być może warto uniknąć tej obsługi bądź zrealizować ją bez użycia tej biblioteki.



Rysunek 1: Wartości statycznych metryk dla zadania 1



Rysunek 2: Wartości statycznych metryk dla zadania 2

Oznaczenia:

- CCN (*ang. Cyclomatic Complexity Number*) – łączna złożoność cyklometryczna funkcji
- NLOC (*ang. Non-comment Line Of Code*) – łączna liczba linii kodu bez komentarzy
- LOC (*ang. Line Of Code*) – łączna liczba linii kodu (z komentarzami)

Zadania

Brzmienie zadań uległo drobnym modyfikacjom.

1. Program typu „Hello World'' mrugający diodą LED. Pozwoli ustanowić środowisko pracy dla każdego z języków i zidentyfikować trudności z tym związane. **(zrobione)**
2. Regulacja obrotów silnika sterowanego sygnałem PWM generowanym sprzętowo i regulowanym za pomocą potencjometru. Do odczytu ustawienia potencjometru wykorzystany zostanie zewnętrzny konwerter ADC podłączony interfejsem I2C lub SPI. **(w trakcie realizacji)**
3. Programowy regulator PID częstotliwości obrotu silnika. Wartość zadana będzie przekazywana przez protokół HTTP/Modbus TCP z komputera zewnętrznego, gdzie będzie również raportowany i wyświetlany przebieg w czasie stanu regulatora (odczyty z czujnika, wartość zadana, sterowanie). Część oprogramowania na komputerze będzie wspólna dla każdego z testowanych języków i nie będzie poddawana testom. **(do zrobienia)**
4. Regulator tak jak w zadaniu 3, tylko zaimplementowany przy użyciu algorytmu DMC zamiast regulatora PID. **(do zrobienia)**

Harmonogram

- 01.2025 – zadanie 2.
- 02.2025 – zadanie 3.
- 03.2025 – zadanie 4.
- 04.2025 – powtórzenie eksperymentów na ESP32, jeśli harmonogram dotrzymany lub nadrobienie zaległości w przeciwnym wypadku.
- 05-06.2025 – pisanie dokumentu wynikowego pracy magisterskiej.