

Raport z projektu

Testowanie i Niezawodność

Paweł Majewski

Jakub Kołodziej

Wykorzystane urządzenia:

- Arduino Nano
- Kabel USB
- Laptop

Oprogramowanie:

- Arduino IDE version 1.8.19
- Visual Studio Code version 1.78.2
- Python 3.10

Założenia projektowe:

- Przedział wartości jakie mogą przyjmować współczynniki wielomianów (0;255) i (-100;0) U (0;100)
- Dane przesyłane przez UART
- Wielomiany stopnia I i II
- Stworzenie aplikacji konsolowej

Cel projektu:

- Część embedded – stworzenie oprogramowania dla mikrokontrolera arduino nano, który będzie odbierał dane (współczynniki wielomianów poprzez UART, obliczał miejsca zerowe funkcji stopnia pierwszego i drugiego, a następnie przysyłał wynik poprzez UART.
- Testowanie głównie polegała na weryfikacji poprawnego działania zaprogramowanego urządzenia, na tą potrzebę zostaną stworzone dwa testy w języku Python, które przysyłać będą przykładowe wartości współczynników poprzez UART i odbierać wynik tym samym interfejsem.

Szczegółowy opis działania poszczególnych kodów:

Test sprawdzający wszystkie możliwe przypadki:

Ten kod wykonuje testy weryfikujące wszystkie możliwe kombinacje współczynników o wartościach z przedziału (0;255) funkcji kwadratowej i liniowej. Poniżej znajduje się krótki opis poszczególnych elementów kodu:

1. Importowane moduły:

- **serial**: moduł umożliwiający komunikację szeregową z urządzeniami, w tym z Arduino.
- **random**: moduł do generowania liczb losowych.

- **time**: moduł do obsługi funkcji związanych z czasem.
 - **math**: moduł zawierający matematyczne funkcje i stałe.
 - **numpy** (importowane jako **np**): biblioteka do pracy z macierzami i tablicami numerycznymi.
2. Funkcja **funKwadratowa()**:
- Funkcja ta testuje funkcje kwadratowe o postaci $ax^2 + bx + c$.
 - Początkowo zdefiniowane są zmienne, takie jak **FAIL** (wartość początkowa 0) oraz **zero1_out** i **zero2_out** (wartości początkowe 1).
 - Tworzona jest lista **coefficients_list**, która zawiera kombinacje liczb (**a**, **b**, **c**) dla wartości z przedziału od 1 do **max_number** (255 w tym przypadku).
 - Następnie dla każdej kombinacji liczb z **coefficients_list** wykonywane są następujące operacje:
 - Wyświetlana jest informacja o funkcji.
 - Oczekiwanie na 2 sekundy za pomocą funkcji **time.sleep(2)**.
 - Wysyłanie danych do portu szeregowego (**ser.write(...)**) w formacie **(c) (b) (a)\n**.
 - Oczekiwanie na odpowiedź z portu szeregowego i odczytanie jej (**result**) przy użyciu **ser.readline()**.
 - Przetwarzanie otrzymanych danych i porównywanie ich z oczekiwanymi wartościami.
 - Wyświetlanie wyników i ewentualne oznaczenie testu jako "PASS" lub "FAIL".
 - Na końcu jest wyświetlane podsumowanie testu.
3. Funkcja **funLiniowa()**:
- Funkcja ta testuje funkcje liniowe o postaci $bx + c$.
 - Jest podobna do funkcji **funKwadratowa()**, ale z inaczej formatowanymi danymi do wysłania na port szeregowy.
 - Testowanie funkcji odbywa się w sposób analogiczny do **funKwadratowa()**, porównując otrzymaną wartość **zero_out** z oczekiwanym miejscem zerowym funkcji liniowej.
4. Warunek **if __name__ == "__main__":**:
- Następuje inicjalizacja połączenia z portem szeregowym (**ser = serial.Serial('COM3', 9600)**), gdzie 'COM3' to nazwa portu, który należy dostosować do swojego środowiska.
 - Wyświetlany jest komunikat **""" start tests """**.
 - Wywołanie funkcji **funKwadratowa()** i **funLiniowa()**, które wykonują testy.
 - Na końcu port szeregowy jest zamykany (**ser.close()**).

Podsumowując, ten kod przeprowadza testy funkcji kwadratowej i liniowej, wysyłając dane do Arduino przez port szeregowy i porównując otrzymane wyniki z oczekiwanymi wartościami.

Test sprawdzający losowe wartości współczynników:

Ten kod również wykonuje testy funkcji kwadratowej i liniowej, ale dla losowych wartości współczynników.

Oto krótkie wyjaśnienie poszczególnych części kodu:

5. Na początku kodu importowane są moduły:
 - **serial** - moduł do obsługi komunikacji szeregowej z urządzeniami,
 - **random** - moduł do generowania liczb losowych,
 - **time** - moduł do obsługi czasu,
 - **math** - moduł zawierający funkcje matematyczne,
 - **numpy** (importowane jako **np**) - biblioteka do obliczeń numerycznych.
6. Następnie definiowane są dwie funkcje pomocnicze:
 - **random_positive()** - generuje trzy losowe dodatnie liczby całkowite z zakresu od 1 do 100 i zwraca je jako krotkę,
 - **random_negative()** - generuje trzy losowe ujemne liczby całkowite z zakresu od -100 do -1 i zwraca je jako krotkę.
7. Funkcja **funKwadratowa()** przeprowadza testy dla funkcji kwadratowej. Wykonuje się 100 rund testowych. W zależności od numeru rundy, generowane są losowe liczby dodatnie lub ujemne. Funkcja wywołuje funkcję **random_positive()** lub **random_negative()** w zależności od warunku **i < max_round / 2**.
8. W każdej rundzie:
 - Generowane są losowe współczynniki **a**, **b** i **c** dla funkcji kwadratowej.
 - Wysyłane są te współczynniki do urządzenia za pomocą portu szeregowego (**ser.write()**).
 - Oczekuje na odpowiedź od urządzenia i odczytuje wynik z portu szeregowego (**ser.readline()**).
 - Porównuje otrzymany wynik z wartościami obliczonymi na podstawie współczynników i sprawdza poprawność.
9. Funkcja **funLiniowa()** przeprowadza testy dla funkcji liniowej. Zasada działania jest podobna do funkcji **funKwadratowa()**, ale oblicza tylko jeden pierwiastek i porównuje go z wynikiem odczytanym z portu szeregowego.
10. Blok **if __name__ == "__main__":** Tworzony jest obiekt **ser** dla komunikacji szeregowej z portem COM3. Następnie wywoływane są funkcje **funKwadratowa()** i **funLiniowa()**, a na końcu port szeregowy jest zamykany.

Ten kod testuje poprawność obliczania pierwiastków dla funkcji kwadratowej i liniowej, porównując wyniki z otrzymanymi z urządzenia.

Kod dla Arduino:

W ArduinoIDE został napisany program obsługujący rozwiązywanie równań liniowych, kwadratowych i sześciennych za pomocą Arduino. Oto jego działanie:

1. Program inicjalizuje port szeregowy o prędkości 9600 bps w funkcji `setup()`.
2. Główna pętla programu (`loop()`) oczekuje na dane wejściowe dostępne na porcie szeregowym.
3. Następnie program odczytuje cztery liczby z portu szeregowego i zapisuje je w tablicy `data`.
4. W zależności od wartości stopnia wielomianu (`stopien`), program rozwiązuje równanie.

- Jeśli stopień to 2, program oblicza deltę i znajduje miejsca zerowe równania kwadratowego.
- Jeśli stopień to 1, program rozwiązuje równanie liniowe.
- Jeśli stopień to 3, program rozwiązuje równanie sześciennne.

5. Wyniki obliczeń są wysyłane przez port szeregowy.

6. Na końcu pętli program oczekuje na wyczyszczenie bufora wejściowego.

Kod ma zdefiniowaną dyrektywę `#define DEBUG`, która jest zakomentowana. Jeśli ta dyrektywa zostanie odkomentowana, wówczas w trakcie działania programu będą wyświetlane dodatkowe informacje diagnostyczne na porcie szeregowym.

W kodzie rozpoczęta jest implementacja rozwiązywania równania sześciennego, która nie jest zawarta w założeniach projektowych.

Program może być wgrany na dowolną platformę wspieraną przez Arduino IDE, ponieważ nie używamy zewnętrznych bibliotek i nie są używane żadne peryferia oprócz portu szeregowego.

Instrukcja:

Zaprogramowane urządzenie (Arduino Nano) podłączamy kablem USB do laptopa. (UWAGA w menedżer urządzeń należy sprawdzić do jakiego portu podłączona jest nasza płytka, jeśli jest inny niż COM3 należy zmienić na właściwą wartość w plikach z testami). Uruchamiamy konsolę, przechodzimy do folderu, gdzie znajdują się nasze pliki .py z testami. Używając polecenia python <nazwa_pliku>.py uruchamiamy wybrany test. Wyniki i poszczególne wartości w danej rundzie testu widzimy na konsoli, na koniec testu zostanie wyświetlone podsumowanie testu.

Podsumowanie:

Testy i kod dla arduino zostały napisane przez dwie osoby niezależnie. Po czym przeprowadzono testy i otrzymano pozytywny rezultat.

Źródła:

Github - https://github.com/PawelMajew/Testowanie_i_Niezawodnosc_projekt.git