Projekt

Wizualizacja danych sensorycznych

Założenia projektowe

Wizualizacja samopozycjonującej się platformy fotowoltanicznej

Albert Lis, 235534

Termin: Śr 17:05

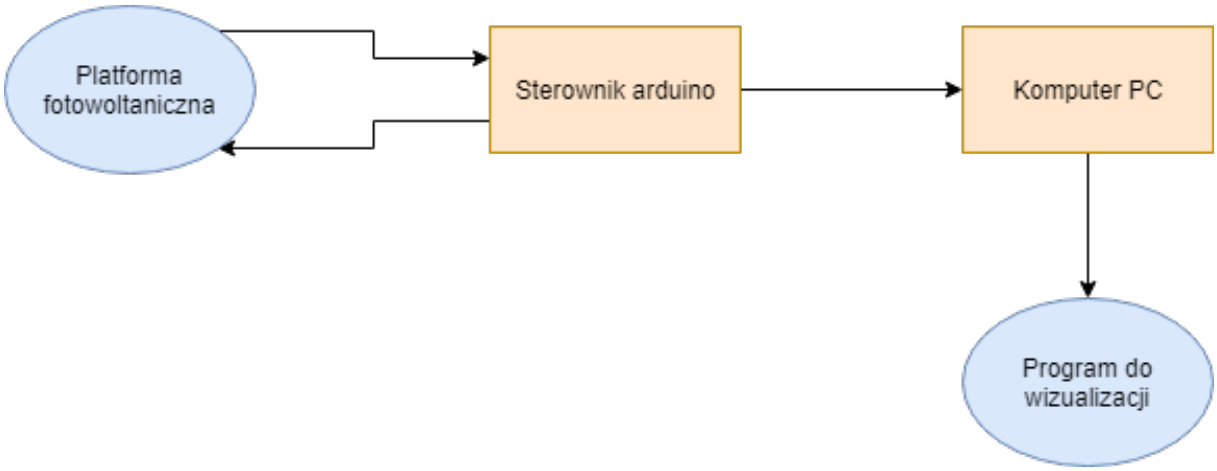
Prowadzący:
dr inż. Bogdan Kreczmer

11 maja 2019

Spis treści

1 Opis projektu

Celem projektu jest stworzenie wizualizacji 3D platformy fotowoltanicznej. Platforma jest sterowana za pomocą mikrokontrolera i czterech czujników. Dzięki temu ma możliwość podążania za najintensywniejszym źródłem światła i pozycjonowania się w sposób umożliwiający optymalne korzystanie z energii słonecznej. Dane o pozycji platformy zostaną przesłane do komputera PC. W komputerze zostanie uruchomiona aplikacja pozwalająca pokazywać aktualną pozycję platformy.



Rysunek 1: Architektura systemu

2 Założenia projektowe

2.1 Komunikacja

1. Połączenie ze sterownikiem
   Realizowane za pomocą modułu Wi-Fi ESP8266 i protokołu UDP/TCP lub bez łączności bezprzewodowej z użyciem portu szeregowego.

2. Połączenie modułu Wi-Fi z mikrokontrolerem
   Realizowane za pomocą portu szeregowego.

2.2 Wizualizacja

1. Środowisko
   Zostanie wykorzystany silnik graficzny UNITY w darmowej wersji.

2. Modele
   Zostaną wygenerowane za pomocą programu Blender.

3. Tekstury
   Zostaną stworzone za pomocą programu GIMP lub pobrane z dowolnej internetowej bazy z darmowymi teksturami.

3 Harmonogram pracy

3.1 Zakres prac

1. Zapoznanie się ze środowiskiem UNITY
   Stworzenie kilku prostych projektów tak aby zapoznać się ze środowiskiem i jego możliwościami.

2. Stworzenie podstawowego modelu 3D
   Stworzenie prostego modelu platformy bez dbałości o detale.

3. Implementacja obrotów platformy za pomocą klawiatury
   Stworzenie wizualizacji poruszania się modelu za pomocą strzałek na klawiaturze.

4. Stworzenie dokładnych modeli w programie Blender
   Stworzenie dokładnego odwzorowania platformy z uwzględnieniem połączeń krawędzi.

5. Wygenerowanie tekstur
   Stworzenie lub pobranie z internetu tekstur dla obiektów.

6. Opracowanie standardu komunikacji sterownik - PC
   Zastanowienie się nad sposobem przesyłania informacji oraz ich kodowaniem.

7. Implementacja komunikacji sterownik - PC
   Implementacja jednostronnej komunikacji między sterownikiem a PC.

8. Implementacja obrotów platformy za pomocą danych ze sterownika
   Modyfikacja istniejącego sterowania w taki sposób aby zwizualizowany stan platformy zgadzał się z rzeczywistym.

9. Poprawki stylistyczne
   Poprawa elementów które okazały się niedopracowane w trakcie projektu.

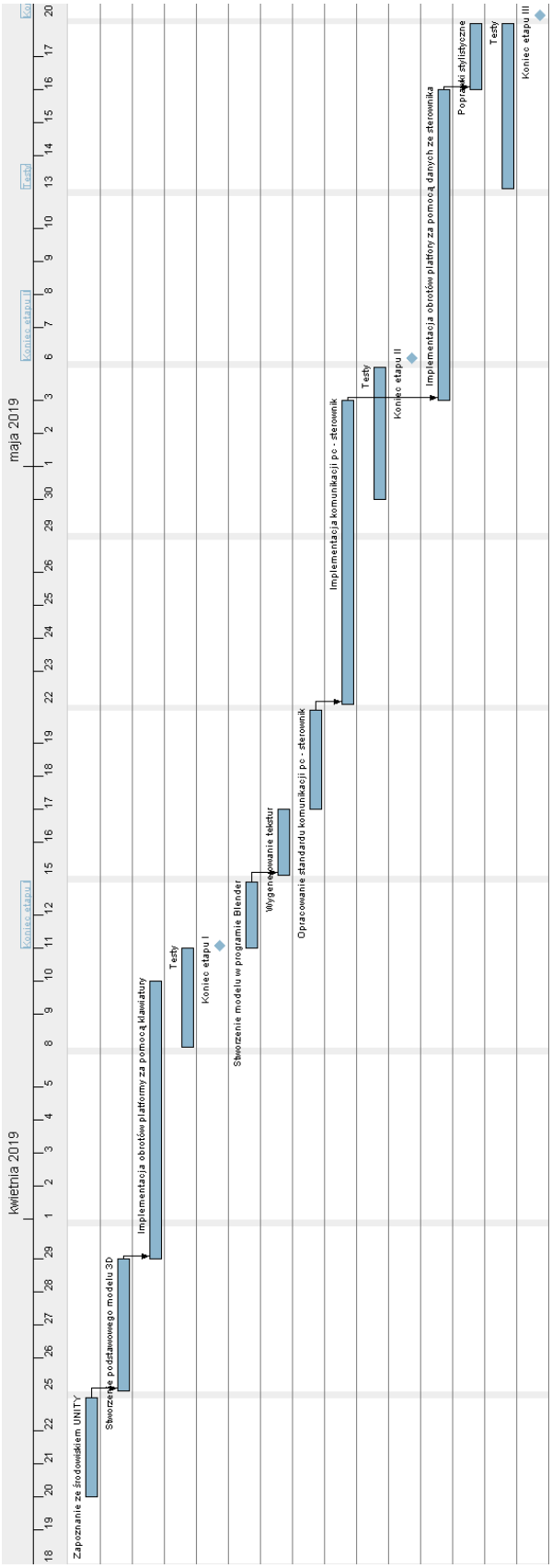
3.2 Kamienie milowe

1. Implementacja działającej wizualizacji w oparciu o sterowanie klawiaturą

2. Implementacja poprawnej komunikacji sterownik - PC

3. Implementacja wizualizacji w oparciu o dane ze sterownika

3.3 Wykres Gantta



Rysunek 2: Diagram Gantta

4 Projekt interfejsu graficznego

4.1 Funkcjonalność UI

1. Lista wyboru nazwy portu szeregowego
   Powinna umożliwić wybranie portu do którego podłączony jest sterownik.

2. Lista wyboru prędkości połączenia
   Powinna zawierać takie prędkości wyrażone w bodach względem których przesłanie pakietu danych będzie trwało mniej niż 1/60[s].

3. Przycisk nawiązania połączenia
   Przycisk umożliwiający nawiązanie połączenia ze sterownikiem po wybraniu odpowiednich parametrów.

4. Przycisk zakończenia połączenia
   Przycisk umożliwiający zakończenie połączenia ze sterownikiem.

5. Przycisk zamknięcia aplikacji
   Przycisk umożliwiający zamknięcie aplikacji. Powinien realizować również akcję zamykania połączenia jeśli nadal by było ono otwarte.

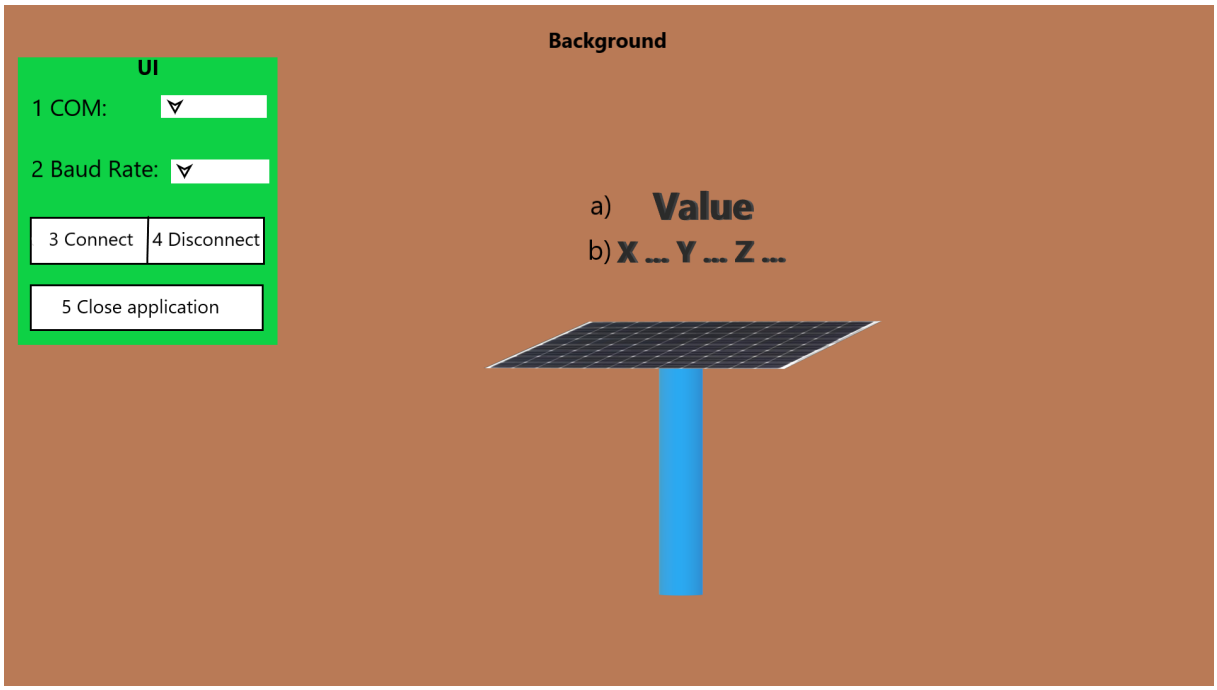
4.2 Funkcjonalność aplikacji

1. Wyświetlanie aktualnej wartości natężenia światła
   Wartość natężenia światła powinna być wyświetlana nad platformą np w postaci napisu 3D.

2. Wyświetlanie aktualnej pozycji
   Powyżej/poniżej wartości natężenia światła powinna być wyświetlana informacja o aktualnej pozycji. Za pomocą schematu XYZ.

4.3 Graficzna reprezentacja aplikacji

I. Schemat



Rysunek 3: Wygląd aplikacji

II. Szczegółowy opis UI

1 COM
Lista modyfikująca parametr odpowiedzialny za nazwę portu w skrypcie obsługi portu szeregowego.

2 Baud Rate
Lista modyfikująca parametr odpowiedzialny za prędkość transmisji w skrypcie obsługi portu szeregowego.

3 Connect
Przycisk wywołujący funkcję odpowiedzialną za nawiązanie połączenia w skrypcie obsługi portu szeregowego.

4 Disconnect
Przycisk wywołujący funkcję odpowiedzialną za zamknięcie połączenia w skrypcie obsługi portu szeregowego.

5 Close application
Przycisk wywołujący skrypt odpowiedzialny za zamknięcie aplikacji.

III. Szczegółowy opis napisów interaktywnych

a) Value
Napis wyświetlający bieżącą wartość natężenia światła. Połączony ze skryptem rotacji aby dostosowywał swoje położenie względem kamery.

b) X . . . Y . . . Z . . .
Napis wyświetlający aktualne położenie we współrzędnych kartezjańskich. Połączony ze skryptem rotacji aby dostosowywał swoje położenie względem kamery.

5 Wstępne rezultaty

5.1 Zmiany w projekcie

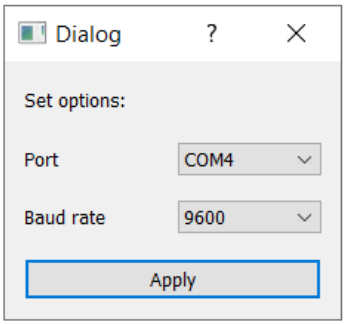
Nastąpiła zmiana środowiska programistycznego z Unity na Qt + OpenGL. To pociągnęło za sobą zmiany w harmonogramie pracy i podejście do projektu. Najpierw zostanie stworzona komunikacja między PC a sterownikiem a następnie wizualizacja 3D. Dodatkowo zmieni się format przesyłanych danych. Aktualnie przewiduję że pakiet danych będzie wyglądał następująco:

$$\text{"H}^{1} \text{ ... V}^{2} \text{ ... L}^{3} \text{ ... I}^{4} \text{ ... CRC}^{5} \text{ ... \textbackslash n}^{6}\text{".}$$

Gdzie ... to poszczególne wartości. Natomiast separator to znak spacji.

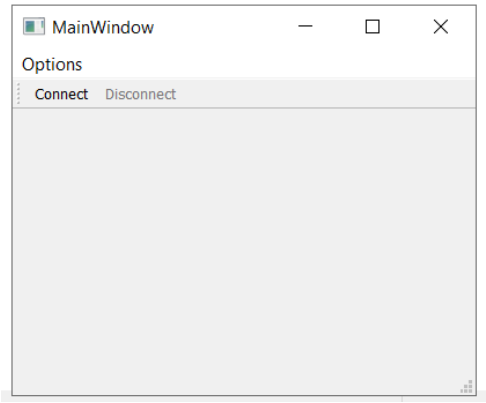
5.2 Zrealizowane zadania

1. **Graficzy interfejs użytkownika**
   Stworzyłem aplikację komunikującą się ze sterownikiem za pomocą portu szeregowego. Aplikacja po uruchomieniu prosi o podanie prędkości komunikacji w bodach oraz portu do którego został przyłączony sterownik. Domyślna prędkość to 9600 bodów. Natomiast lista portów wczytuje tylko te dostępne.



Rysunek 4: Okno opcji

Po zaakceptowaniu ustawień uruchamia się okno główne w którym mamy opcje Connect oraz Disconnect. Obie wzajemnie się wykluczają. Dodatkowo gdy połączenie jest aktywne wygaszona zostaje opcja zmiany ustawień połączenia.



Rysunek 5: Okno główne

[1] rotacja horyzontalna
[2] rotacja wertykalna
[3] natężenie światła
[4] Zmierzone natężenie prądu
[5] 32-bitowa suma kontrolna
[6] znak końca pakietu danych

2. **Komunikacja**
   Komunikacja jest uruchamiana w osobnym wątku tak aby nie zakłócać pracy głównego okna. Port szeregowy został skonfigurowany z 8 bitami danych, bitem parzystości oraz bitem stopu. Aktualnie przesyłane dane wyświetlam za pomocą konsoli. Gdy suma kontrolna się nie zgadza wyświetlam komunikat o niepoprawnej ramce danych.