Inżyniera Oprogramowania

Dokumentacja Projektu: MousEye

|  |
| --- |
| Sekcja 8 |
| Łukasz Dąbek |
| Andrzej Bydzicki |
| Karol Adamus |
| Damian Klapiński |

Spis treści

[1. Opis Projektu 3](#_Toc422744505)

[1.1. Cel projektu 3](#_Toc422744506)

[1.2. Zasada działania 3](#_Toc422744507)

[1.2.1. Hardware 3](#_Toc422744508)

[1.2.2. Proces modyfikacji 3](#_Toc422744509)

[1.2.3. Software 7](#_Toc422744510)

[2. Wymaganie funkcjonalne i pozafunkcjonalne. 8](#_Toc422744511)

[2.1. Wymaganie funkcjonalne 8](#_Toc422744512)

[2.2. Wymaganie pozafunkcjonalne 8](#_Toc422744513)

[3. Strategia postępowania 8](#_Toc422744514)

[4. Zastosowane narzędzia 9](#_Toc422744515)

[5. Słownik Terminów 9](#_Toc422744516)

[6. Architektury systemu 9](#_Toc422744517)

[7. Wybrane diagramy UML 9](#_Toc422744518)

[8. Dokumentacja użytkownika 9](#_Toc422744519)

[9. Harmonogram 9](#_Toc422744520)

[10. Podział prac i obowiązków 11](#_Toc422744521)

[11. Wnioski 11](#_Toc422744522)

# Opis Projektu

## Cel projektu

Cele projektu jest stworzenie oprogramowania wraz z urządzeniem peryferyjnym, które pozwolą kontrolować ruchem gałki ocznej wskaźnik kursora myszy.

## Zasada działania

W projekcie wyróżnić można dwa zasadnicze elementy – hardware oraz software, które to stanowią integralną całość.

### Hardware

Wykorzystywanym sprzętem jest zmodyfikowana kamera Playstation Eye. Urządzenie poddane następującym modyfikacją:

* Oryginalna obudowa urządzenia została usunięta.
* Układ optyki urządzenia pozbawiony został filtru światła podczerwonego.
* Do urządzenia dołączony został układ wyposażony w cztery diody emitujące światło podczerwone.
* Urządzenie zawieszone zostało na ramieniu pozwalającym na utrzymanie stałej pozycji i obserwację oka użytkownika.

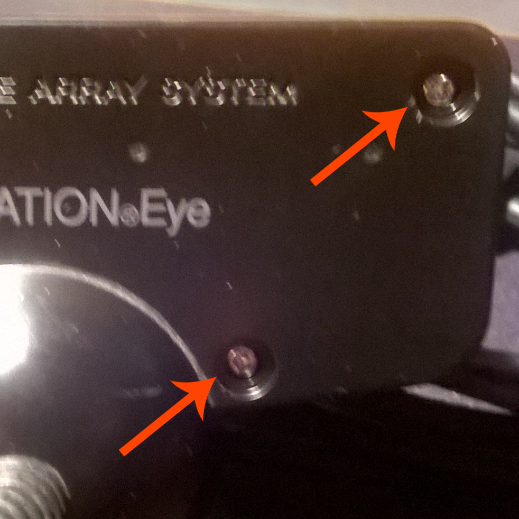
### Proces modyfikacji

Proces modyfikacji należy rozpocząć od usunięcia zaślepek ukrywających śrubki.



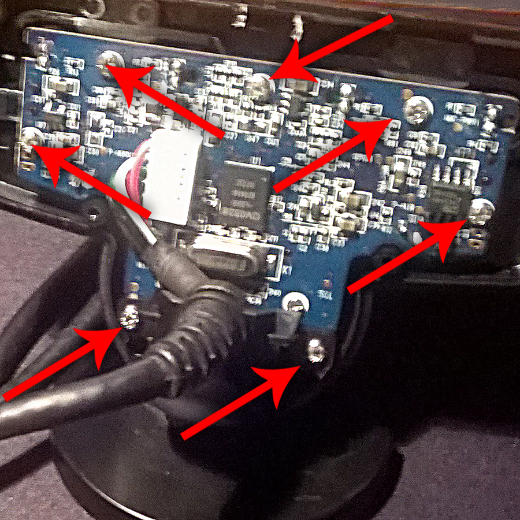
Rysunek 1 Usunięcie zaślepek.

W następnym kronu należy wykręcić uwidocznione śrubki mocujące obudowę.



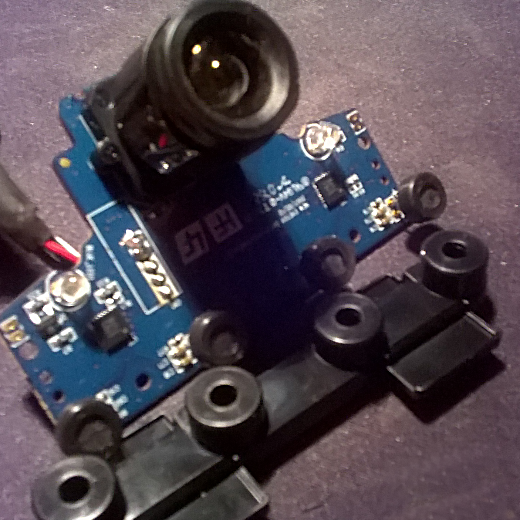
Rysunek 2 Wykręcenie śrubek i zdjęcie tylnego panelu.

Kamerę należy wymontować z obudowy wykręcając zaznaczone śrubki.



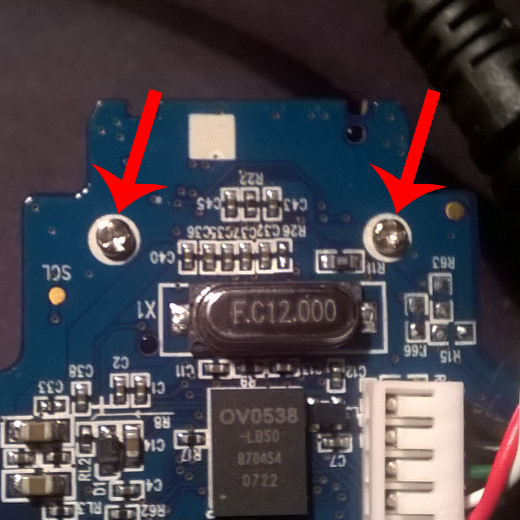
Rysunek 3 Wykręcenie śrubek mocujących urządzenie do obudowy.

Jeśli wszystkie śrubki zostały wykręcone osłona mikrofonów powinna odejść z łatwością.



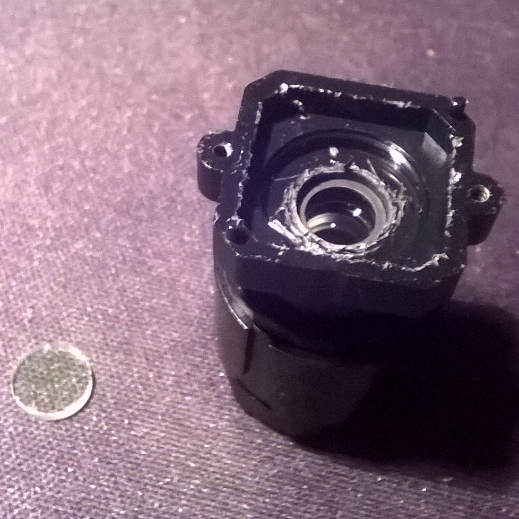
Rysunek 4 Usunięcie osłony mikrofonów.

By przystąpić do usunięcia filtru podczerwieni należy wykręcić śrubki montujące obiektyw do płytki kamery.



Rysunek 5 Wykręcenie śrubek podtrzymujących obiektyw.

Za pomocą noża do tapet bądź inne ostrego narzędzia należy wyciąć plastik trzymający filtr podczerwony. Po delikatnym podważeniu powinien on odejść od obudowy obiektywu.



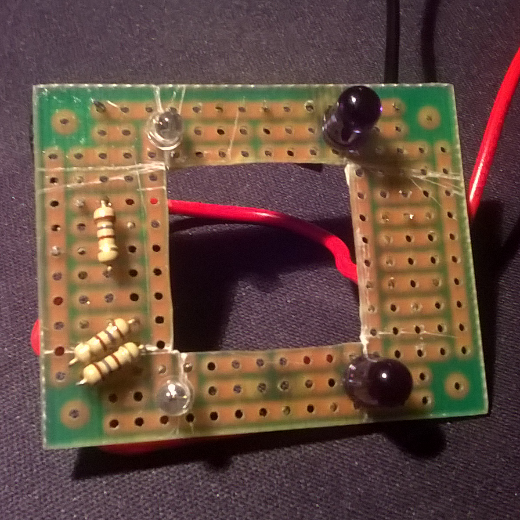
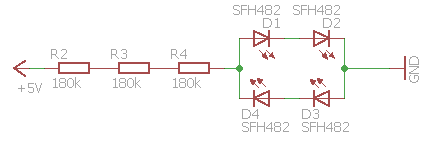
Rysunek 6 Usunięcie plastikowej obwódki mocującej filtr podczerwieni w obudowie obiektywu.

Po usunięciu filtru podczernieni obiektyw należy z powrotem przymocować do kamery.



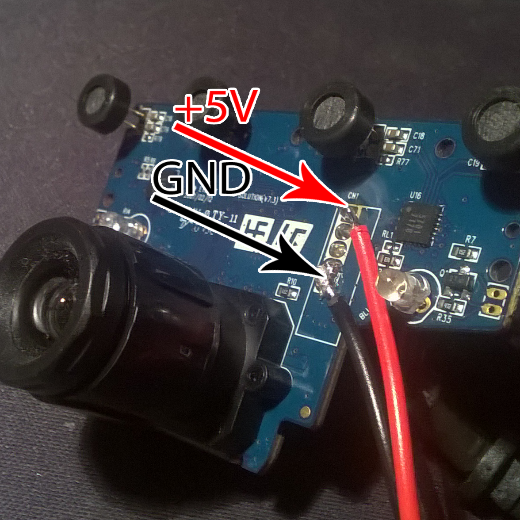
Rysunek 7 Ponowne przykręcenie obiektywu.

Wykorzystując schemat przedstawiony poniżej należy zmontować układ diod.



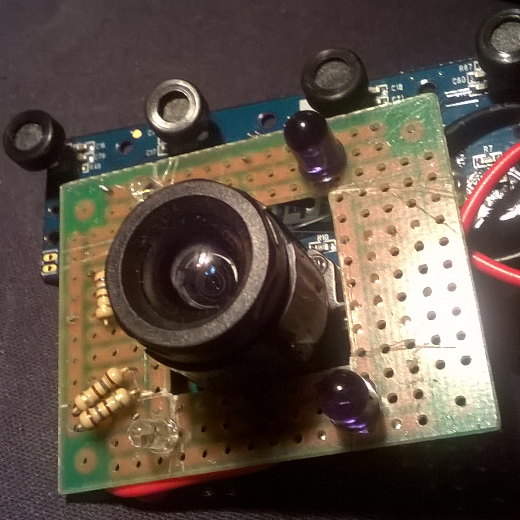
Rysunek 8 Zmontowanie układu diod podczerwonych.

Przewody zasilające wyprowadzone z układu diod należy przylutować w miejscach wskazanych poniżej.



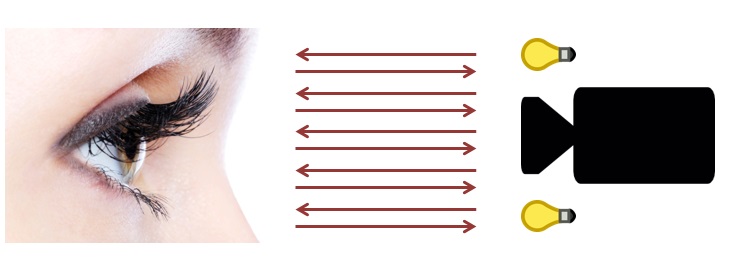
Rysunek 9 Przylutowanie przewodów zasilających układu diod do pinów zasilających kamery.

Poprzez nałożenie ciepłego kleju na diody sygnalizacyjne kamery należy przymocować układ diod podczerwonych do urządzenia.

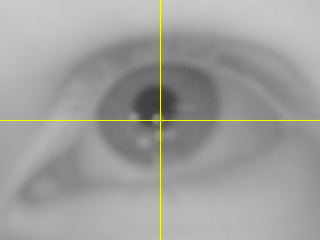


Rysunek 10 Przyklejenie układy diod podczerwonych do kamery za pomocą ciepłego kleju.

Modyfikacja urządzenia ma na celu ułatwienie lokalizacji źrenicy oka użytkownika. Diody emitują światło podczerwone, które odbija się od źrenicy użytkowania uwydatniając ją tym samym.



Rysunek 11 Schemat odbicia światła podczerwonego od okna użytkownika..

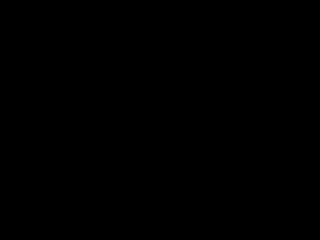


Rysunek 12 Obraz z kamery po dokonaniu modyfikacji.

## Software

Podstawowym zadaniem software’u jest komunikacja komputera z kamerą, odbieranie i przetwarzanie obrazu, umożliwienie konfiguracji całego zestawu oraz ostatecznie sterowanie kursorem za pomocą gałki ocznej. Do poprawnego działanie niezbędnie jest zainstalowanie sterownika obsługującego urządzeniem, który pobrać można ze strony: <https://codelaboratories.com/products/eye/driver/>.

Po uruchomieniu programu jesteśmy proszeni o przejście przez proces kalibracji. Jego zadaniem jest zadbanie by urządzenie zostało poprawne ustawienie na głowie użytkownika, jak i dobór parametrów pozwalających na poprawną interpretację obrazu. Ostatnim etapem kalibracji jest ustalenie, gdzie znajdują się cztery rogi ekranu względem położenia gałki ocznej użytkownika. Po tym procesie program przechodź w stan pracy normalnej. Po odebraniu oryginalnego obrazu z kamery dokuje odwrócenia kolorów, co sprawia, że źrenica jest białym elementem obrazu, następnie tak przetworzony obraz sprowadzany jest do czarno białej postaci. Na czarno zaznaczane są piksele, których kolor nie był biały, natomiast piksele białe zachowują swoją barwę. Operacja ta pozwana na zaznaczenie na oryginalnym obrazie gdzie znajduje się źrenica, jak i obliczenie jej środka.



Rysunek 13 Poszczególne etapy przetwarzania obaru.

# Wymaganie funkcjonalne i pozafunkcjonalne.

## Wymaganie funkcjonalne

* Obserwacja oka użytkownika.
* Przetwarzanie zarejestrowanego obrazu.
* Interpretacja zarejestrowanego obrazu.
* Sterowanie pracą kursora myszy, na podstawie zgromadzonych danych.

## Wymaganie pozafunkcjonalne

* Zapewnienie precyzyjnego i wygodnego sterowania kursorem myszki
* Symulacja wszystkich funkcji myszki:
  + Klikniecie prawym i lewym przyciskiem myszki
  + Zaznaczanie
  + Przeciąganie i upuszczanie
  + Skrolowanie

# Strategia postępowania

Przyjętą przez nas metodologią prowadzenia pra była metodologia scrum wraz z modelem kaskadowym. Prace podzieliśmy na cztery spinty o długości czterech tygodni każdy. W realizacji naszego projektu wyróżnić możemy następujące etapy:

* Analiza problemu
* Komunikacja z warstwą hardwaru
* Przetwarzanie otrzymanego obrazu
* Właściwe sterowanie pracą kursora myszy

Rysunek 14 Organizacja prac w projekcie.

Analiza problemu

Komunikacja z warstwą hardwaru

Przetwarzanie otrzymanego obrazu

Właściwe sterowanie pracą kursora myszy

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Nasze podejście podyktowane było faktem, iż do realizacji całości projektu niezbędne było ukończenie każdego etapu. Dodatkowo zamkniecie każdego z nich w ramach czterech tygodni pomogła nam zachować optymalne tempo prac.

# Zastosowane narzędzia

Do realizacji projektu zastosowaliśmy narzędzia, które wymieniliśmy w tabeli poniżej.

Tabela 1 Zestaw wykorzystanych narzędzi.

|  |  |
| --- | --- |
| System kontroli wersji | Git |
| Zarządzanie projektem | Trello, Redmine |
| Środowisko | Visual Studo 2013 |
| Język | C#, WPF |
| oprogramowanie dodatkowe i pluginy | |
| TeamViever | |
| CideMaid | |
| Resharper | |

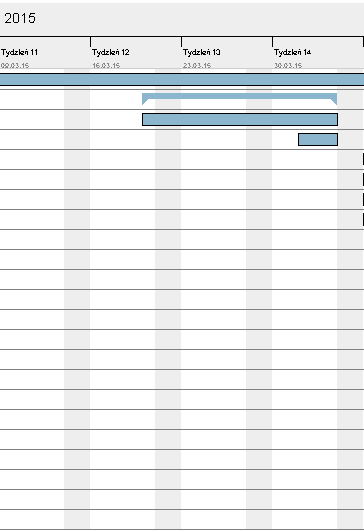
# Słownik Terminów

# Architektury systemu

# Wybrane diagramy UML

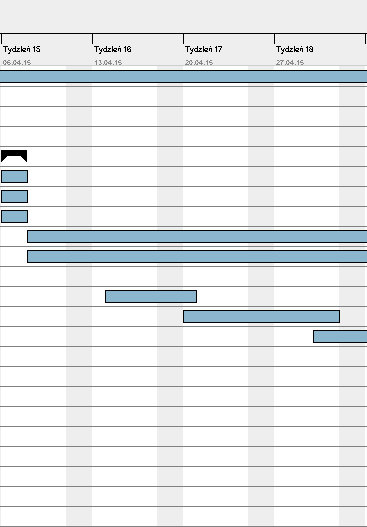
# Dokumentacja użytkownika

# Harmonogram

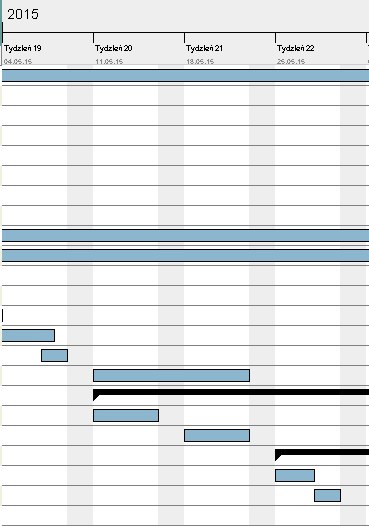


Rysunek 15 Sprint 1.

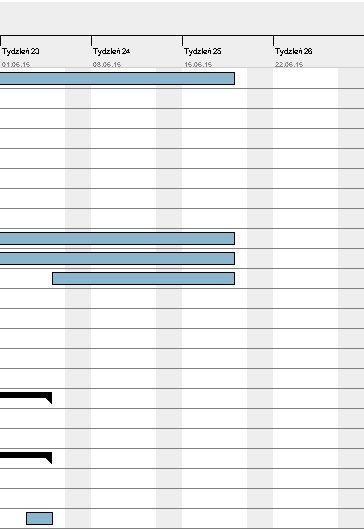
Realizując pierwszy sprit skupiliśmy się na zapoznanie się z problemem. Staraliśmy się ustalić jakie funkcjonalności powinna implementować budowana przez nas aplikacja.



Rysunek 16 Sprint 2.



Rysunek 17 Sprint 3.



Rysunek 18 Sprint 4.

# Podział prac i obowiązków

|  |  |
| --- | --- |
| Łukasz Dąbek | Organizacja i nadzorowanie prac nad projektem i prezentacjami. Modyfikacja urządzenia. Prace programistyczne. |
| Andrzej Bydzicki | Implementacja i opracowanie algorytmów przetwarzania obrazu. |
| Damian Klapinski | Opracowanie interfejsu użytkownika |
| Karol Adamus | Sporządzanie prezentacji i dokumentacji. |

# Wnioski