



**Politechnika
Śląska**

Dokumentacja do Project Basic Learning

Framer - Wizualizacja danych z kamery termowizyjnej

Michał LUBCZYŃSKI

Andrzej ZAGÓRSKI

Eryk SZMYT

PROWADZĄCY PRACĘ

Dr. Inż. Aleksander Iwaniak

KATEDRA

Wydział Inżynierii Materialowej

Katowice 2023

Tytuł pracy

Framer - Wizualizacja danych z kamery termowizyjnej

Streszczenie

Framer.py to aplikacja w języku Python, która wizualizuje dane z kamery termowizyjnej MLX90640 podłączonej do Raspberry Pi 4B za pośrednictwem magistrali I2C. Aplikacja wykorzystuje SocketIO, biblioteki Adafruit i Flask, aby zapewnić interaktywne pomiary i dane w czasie rzeczywistym. Przeglądaj i analizuj rozkłady temperatur w przyjaznej dla użytkownika kanwie HTML.

Słowa kluczowe

Raspberry, Termowizja, MLX90640, Python, HTML, Canvas

Thesis title

Framer - Thermal Camera Data Visualization

Abstract

Framer.py is a Python application that visualizes thermal camera data from the MLX90640 model connected to a Raspberry Pi 4B via the I2C bus. The application utilizes SocketIO, Adafruit libraries, and Flask to provide a real-time interactive experience. Explore and analyze temperature distributions in a user-friendly HTML canvas.

Key words

Raspberry, Thermo-vision, MLX90640, Python, HTML, Canvas

Spis treści

1	Wstęp	1
2	Ustawienia dla prowadzącego	3
3	Ustawienia dla klienta	5
4	[Właściwy dla kierunku – np. Specyfikacja wewnętrzna]	7
5	Podsumowanie i wnioski	9

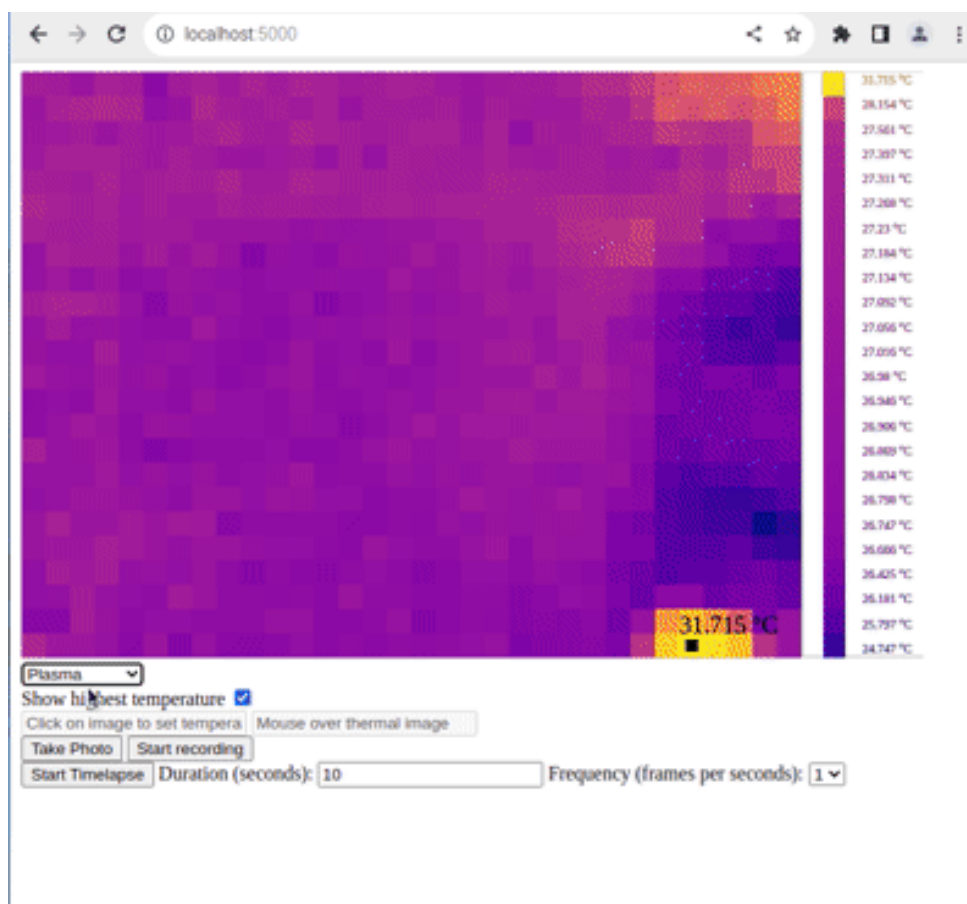
Rozdział 1

Wstęp

Celem pracy jest stworzenie oprogramowania do przechwytywania danych z modułu kamery termowizyjnej - model MLX 90640. Do tego celu wykorzystano Raspberry Pi 4b do którego podłączono za pomocą przewodów połączeniowych "I2C/UART - 4-pinowy wtyk żeński", moduł kamery wspomniany wcześniej z kątem widzenia 110° , komunikujący się poprzez włączony w systemie Ubuntu, interfejs I2C. Matryca kamery 32×24 umożliwia pomiar temperatury w zakresie od -40°C do 300°C z dużą dokładnością do $1,5^\circ\text{C}$. Płytkę kamery zasilana jest napięciem 3,3 V ale umożliwia także podpięcie pod 5V. Częstotliwość odświeżania została ustawiona na 2Hz. Program framer.py na platformie raspberry został napisany z myślą o przedstawianiu zjawisk termicznych obserwowanych za pomocą mobilnego stanowiska kierowanego przez prowadzącego. Umożliwia klientom w sieci lokalnej do obserwacji w czasie rzeczywistym zjawisk które mogą wykonywać się także w znacznej odległości od nich za pomocą dowolnego urządzenia z dostępem do sieci w której znajduje się aparat do badania termicznego. Administrator sieci lokalnej, którym w założeniach jest prowadzący zajęcia i nadzorujący obserwacje, ma prawo wpuszczenia konkretnych hostów co jednoznacznie daje im prawo do korzystania z uruchomionej aplikacji na urządzeniu do przechwytywania obrazu. Klienci mają dostęp do:

- Zmiany mapy kolorów za pomocą pola wyboru w celu lepszej interpretacji danych.
- Wyświetlanie temperatury pod kursorem myszy w celu uzyskania natychmiastowej informacji zwrotnej
- Śledzenie najgorętszego miejsca wraz z jego temperatura wypisaną w kolorze widocznym na tle.
- Śledzenie temperatury w określonym punkcie kanwy
- Skala temperatury dynamicznie aktualizowana po prawej stronie obszaru roboczego.
- Przechwytywanie obrazu .png rzeczywistych danych wraz z skalą i każdym znacznikiem który śledzimy.

- Nagrywanie wideo reagujące na start/stop przez maksymalnie 10 minut w najwyższej możliwej jakości 2 FPS
- Nagrywanie poklatkowe z możliwością wyboru liczby klatek na sekundę.
- Zapisywanie w lokalnych plikach hosta z dokładną datą i godziną przechwycenia.



Rysunek 1.1: Interfejs użytkownika

Rozdział 2

Ustawienia dla prowadzącego

Aby aplikacja została uruchomiona w środowisku przygotowanym przez jednego z współtwórców, wystarczy po uruchomieniu systemu, upewnić się o wystąpieniu zielonej sygnalizacji przez diodę led, na module kamery potwierdzające poprawne podłączenie jej do zasilania. Następnie należy upewnić się o poprawnym połączeniu z modulem poprzez polecenie: `i2cdetect -y 1`. Jeżeli wystąpi problem z uruchomieniem polecenia należy doinstalować brakującą bibliotekę poleceniem: `sudo apt-get install i2c-tools`. Jeżeli po pomyślnym wykonaniu pierwszej komendy, występuje zapis 33 mamy potwierdzenie o poprawnej komunikacji mikrokontrolera przez interfejs i2c z kamerą. W terminalu należy wywołać wówczas za pomocą polecenia `python program` z ścieżka o ile nie znajdujemy się w jego lokalizacji. Przykładowe polecenie może wyglądać: `python3 Desktop\Framer\framer.py`

Aby aplikacja została uruchomiona w nowym środowisku tj Raspberry z kamerą MLX90640, wystarczające powinno być skopiowanie repozytorium pod adresem github.com/michalubczynski/Framer. W przypadku braku dostępu do sieci, niezbędne biblioteki to kolejno:

- `adafruit-circuitpython-mlx90640`
- `flask`
- `python-socketio`

W przypadku braku dostępu do repozytorium lecz posiadając dostęp do sieci, należy wprowadzić w terminalu następujące polecenia:

```
pip install adafruit-circuitpython-mlx90640 flask python-socketio
python3 framer.py
```

Częstotliwość odświeżania zadeklarowana przez producenta 0,5 Hz do 64 Hz po przeprowadzonych testach została zdementowana albowiem wartości powyżej 2Hz wpływały wyłącznie negatywnie na czas odświeżania klatek. Aby klienci mogli korzystać z danych rozpropagowywanych przez mikrokontroler, musi zostać na nim uruchomiona aplikacja

Aby nieprzerwanie przechwytywać obraz z wybranymi ustawieniami jak i podczas jakichkolwiek operacji klient jak i operator zobowiązani są do zapewnienia stałego, stosunkowo szybkiego łącza internetowego jeśli mówimy o sieci lokalnej z wieloma instancjami klientów. Dla samodzielnej aplikacji na aparacie nie jest wymagane połączenie z internetem. Połączenie następuje poprzez

Rozdział 3

Ustawienia dla klienta

Dostęp do aplikacji uzyskujemy poprzez wprowadzenie w przeglądarce IPaparatu::5000 co jest kolejno adresem ip aparatu, o który należy zapytać operatora, oraz port na którym hostowana aplikacja. Nie zostały przetestowane maksymalne ilości jednocześnie korzystających hostów z portu z dostępem do aplikacji.

Rozdział 4

Ograniczenia aplikacji

- Przeglądarki
- Ilość klientów
- Odświeżanie
- UDP a nie TCP
- Interpolacja obrazu a wypełnionej canvy

Rozdział 5

Problemy podczas developmentu

- Zrzuty ekranu i gify
- Poszukiwania biblioteki do zapisu wideo i kodeki

Rozdział 6

Podsumowanie i wnioski

- uzyskane wyniki w świetle postawionych celów i zdefiniowanych wyżej wymagań
- kierunki ewentualnych danych prac (rozbudowa funkcjonalna ...)
- problemy napotkane w trakcie pracy

