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 Materiałowej

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 Visualizatioh

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

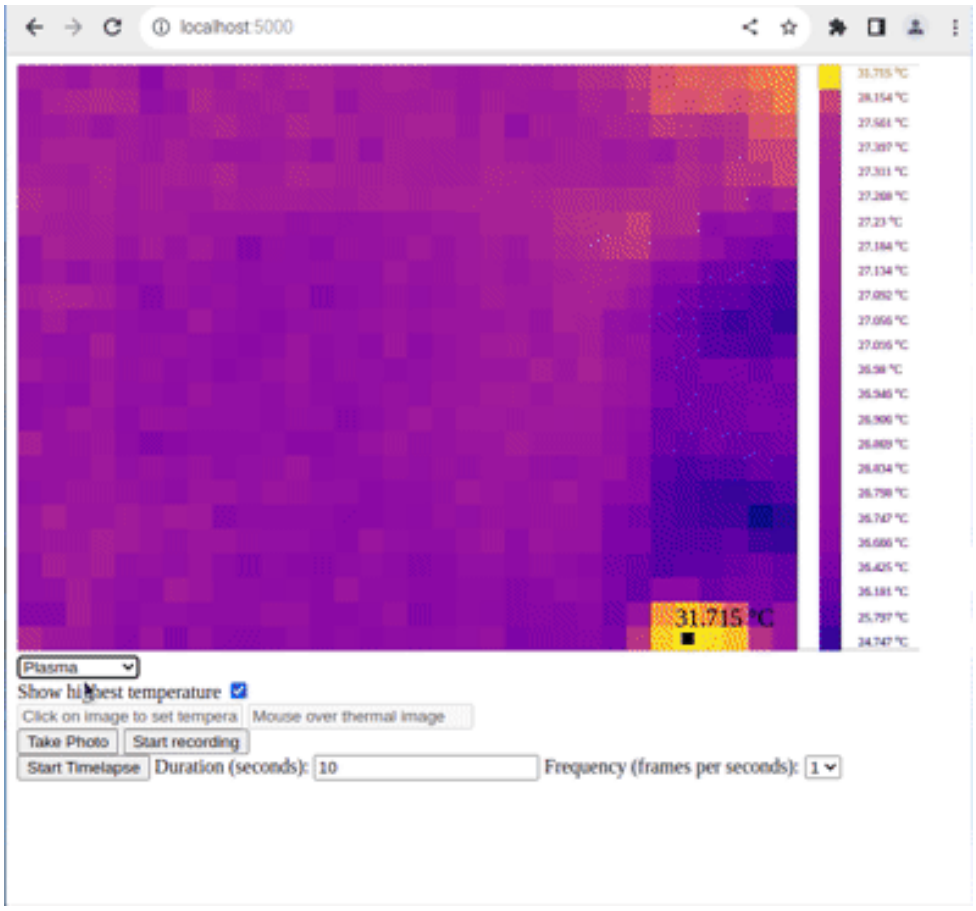
Rozdział 1

Wstęp

Celem pracy jest stworzenie oprogramowania do przechwytywania danych z modulu 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 32x24 umożliwia pomiar temperatury w zakresie od -40°C do 300°C z dużą dokładnością do 1,5°C. Płytka kamery zasilana jest napięciem 3,3 V ale umożliwia także podpięcie pod 5V. Częstotliwość odświeżania zostala ustawiona na 2Hz. Program framer.py na platforme 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

Uruchomienie

2.1 Dla prowadzącego

Aby aplikacja została uruchomiona w środowisku przygotowanym przez jednego z współtwórców należy:

1. Upewnić się o poprawnym zasilaniu Raspberry Pi
2. Uruchomić system
3. Upewnić się o wystąpieniu zielonej sygnalizacji przez diodę led, na module kamery, potwierdzające poprawne podłączenie jej do zasilania.
4. Upewnić się o poprawnym połączeniu z modułem poprzez polecenie:
   $sudo i2cdetect −y 1. (Jeżeli wystąpi problem z uruchomieniem polecenia należy doinstalować brakującą bibliotekę poleceniem:
   $sudo apt−get install i2c−tools.) Poprawna komunikacja Raspberry z kamerą przez interfejs i2c potwierdza wystąpienie liczby 33 w tablicy wypisanej w konsoli.
5. W terminalu należy wywołać wówczas za pomoca polecenia python program z sciezka o ile nie znajdujemy sie 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
```

2.2 Dla uczestnika

Dostęp do aplikacji uzyskujemy poprzez wprowadzenie w przeglądarce "adresIpAparatu:5000"np: 192.168.0.11:5000 co jest kolejno adresem ip aparatu 192.168.0.11, o ktory należy zapytać operatora, oraz stały port 5000 na ktorym hostowana jest aplikacja. Jeśli na Twojej sieci lokalnej działa firewall, upewnij się, że nie blokuje dostępu do portów, które są używane przez serwer WWW. W przypadku braku dostępu do aplikacji po zapewnieniu poprzednio wspomnianych wymagań, należy skontaktować się z administratorem. Testowane przeglądarki dostępne dla użytkowników aplikacji:

- Opera v106.0.4998.2
- Opera gx v105.0.4970.37
- Google Chrome v119.0.6045.209
- Mozilla Firefox v120.0.1
- Microsoft Edge v120.0.2210.61
- Safari v17.1
- Brave v1.60.125
- Microsoft Edge v120.0.2210.61

Rozdział 3

Funkcjonalności

3.1 Obraz

Wyświetlany obraz ma format 32x24 zgodnie z matrycą i jest skalowany na obiekt Canvas o wymiarach 640x480 co daje nam powiększenie rzędu 2000%. Obraz jest odświeżany dwa razy na sekundę tj. z częstotliwością 2Hz.

3.2 Skala temperatury

Skala występujących temperatur na obrazie, znajduję się po prawej stronie od obrazu. Posiada dokładnie 24 wartości, najwyższą u góry, najniższą u dołu. Wypisane temperatury odpowiadają kolorowi przy jakim zostały wypisane oraz w jakim kolorze zostały napisane (kolor cyfr odpowiada kolorowi do którego się odnoszą). Tekst posiada czarną obwódkę dla zwiększenia czytelności tekstu. Skala jest dopisywana do zdjęcia o którym mowa w podpunkcie 3.7.

3.3 Kolor mapy

Wyświetlany obraz jest widoczny w jednym dostępnych map kolorów. Oznacza to, że użytkownik w zależności od indywidualnych preferencji, może wybrać dla siebie najbardziej czytelny zestaw wyświetlanych kolorów. Dostępne zestawy to następująco:

- **Inferno**

Rysunek 3.1: Colormap inferno

- **Plasma**

Rysunek 3.2: Colormap plasma

- **Rainbow**

Rysunek 3.3: Colormap rainbow

- **Seismic**

Rysunek 3.4: Colormap seismic

- **Greys**

Rysunek 3.5: Colormap greys

- **Hot**

Rysunek 3.6: Colormap hot

- **Hot and Cold**

Rysunek 3.7: Colormap hot and cold

3.4 Najgorętszy punkt na obrazie

Hot spot to opcja do włączenia, znajdująca się pod obrazem z podpisem "Show highest temperature"która włącza i wyłącza zaznaczanie i wypisywanie najgorętszej temperatury wraz z jej miejscem na obrazie. Do malowania wspomnianego punktu wykorzystywane jest wyznaczanie koloru tła i jego konwersja na biały bądź czarny w zależności który jest bardziej widoczny na tle.

3.5 Śledzenie wyznaczonego punktu

Po kliknięciu na obraz, pojawi się na nim biały punkt którego temperatura jest wypisywana w polu pod obrazem z opisem w zależności od scenariusza. Jeżeli nie został wybrany punkt do śledzenia pojawi się informacja "Click on image to set temperature point". W przeciwnym razie pojawi się "Point position:"z dodaną, aktualizowaną temperaturą punktu który śledzimy.

3.6 Śledzenie temperatury pod kursorem

W czasie rzeczywistym odczytywana jest stale temperatura spod kursora, jeśli kursor znajduje się na wyświetlanym obrazie z kamery, Wówczas temperatura wyświetlana jest jako "Mouse position: (temperatura *C)". W przeciwnym razie użytkownik jest powiadamiany o wyjechaniu kursorem poza obraz informacją "Mouse over thermal image". Informacje te znajdują się w kontrolce pod obrazem a po lewej stronie od kontrolki z punktu 3.5.

3.7 Robienie zdjęć

Do funkcji robienia zdjęć służy przycisk umiejscowiony pod obrazem, opisany "Take Photo". W momencie jego przyciśnięcia, zostaje pobierany lokalnie przez klienta obraz, wraz ze skalą oraz każdym wskaźnikiem namalowanym na obrazie. Zdjęcie to posiada format .png oraz jest opisane jako thermalImage<dzien_miesiac_rok@godzina_minuta_sekunda .png> czasu UTC. Obraz ma wymiary 740x480 i jego rozmiar nie powinien przekraczać 50kB. Domyślnym folderem zapisywania jest folder C:\\Downloads .

3.8 Nagrywanie wideo

Do funkcji nagrywania wideo służy przycisk Start/Stop recording. Jest to przycisk rotacyjny co oznacza, że przy pierwszym wyzwoleniu jest widoczny napis Start recording a przy drugim Stop recording. Takie zachowanie jest naturalne ze względu na możliwości użytkownika do uruchamiania i zatrzymywania nagrywania w dowolnym momencie. Maksymalny czas nagrania został ustawiony na 10 minut po którym w przypadku braku akcji ze strony użytkownika samoistnie zakończy się nagranie i zostanie pobrane lokalnie do folderu C:\\Downloads . Wideo to posiada format .mp4 oraz jest opisane jako thermalVideo<dzien_miesiac_rok@godzina_minuta_sekunda.png> czasu UTC. Nagrywanie jest przechwytywane w maksymalnym dostępnym odczycie dwie klatki na sekundę. Podczas nagrywania nie jest dostępna żadna inna funkcja przechwytywania obrazu czyt. robienie zdjęć oraz Timelapse.

3.9 Timelapse

Do funkcji Timelapse służy przycisk Start Timelapse. Do ustawiania czasu trwania timelapse służy pole opisane "Duration"i jest ono opisane w sekundach. W odróżninieniu od wideo, możemu tutaj zmienić ilość klatek na sekundę w celu zaoszczędzenia miejsca na dysku. Dostępne pola to 1 oraz 2 fps. Po zakończeniu czasu trwania timelapse, wideo zostanie pobrane lokalnie do folderu C:\\Downloads . Wideo to posiada format .mp4 oraz jest opisane jako thermalVideo<dzien_miesiac_rok@godzina_minuta_sekunda.png> czasu UTC. Uruchomiony Timelapse nie może zostać przerwany w konwencjonalny sposób, lecz może zostać przerwane poprzez odświeżenie przeglądarki.

Rozdział 4

Proces rozwoju aplikacji

4.1 Wybór środowiska i narzędzi

Wybór środowiska i narzędzi w projekcie Framer.py wynikał z potrzeby stworzenia aplikacji do wizualizacji danych z kamery termowizyjnej w czasie rzeczywistym, która zapewnia interaktywne i przyjazne dla użytkownika doświadczenie.

- MLX90640 Thermal Camera i Raspberry Pi 4B
  Kamera termowizyjna MLX90640 została wybrana ze względu na jej zdolność do przechwytywania danych temperatury w formacie matrycy. Raspberry Pi 4B została wybrana jako platforma ze względu na jej możliwości, w tym obsługę GPIO i możliwość połączenia z MLX90640 za pośrednictwem magistrali I2C.

- SocketIO
  SocketIO został wykorzystany do komunikacji w czasie rzeczywistym między serwerem a klientem. Wybór ten umożliwia płynne i natychmiastowe aktualizacje danych termicznych na kanwie HTML, zapewniając użytkownikom interaktywne wrażenia.

- Adafruit
  Biblioteka Adafruit CircuitPython dla MLX90640 została wykorzystana do połączenia z kamerą termowizyjną poprzez magistralę I2C. Biblioteki te ułatwiają integrację i komunikację z czujnikiem, zapewniając dokładne i niezawodne pobieranie danych o temperaturze.

- Flask
  Flask, lekki i rozszerzalny framework sieciowy, został wybrany do obsługi interfejsu HTML i interakcji z użytkownikiem. Jego prostota sprawia, że nadaje się do tego projektu, umożliwiając szybkie opracowanie aplikacji internetowej do wyświetlania danych termicznych.

- HTML i Canvas
  Element HTML canvas został wykorzystany do renderowania danych termicznych w czasie rzeczywistym. Jego elastyczność i łatwość użycia sprawiają, że jest to odpowiedni wybór do tworzenia dynamicznych wizualizacji w przeglądarce internetowej.

- Python
  Z racji użytkowania klas udostępnionej biblioteki w pythonie adafruit-circuitpython-mlx90640 zdecydowaliśmy nie multiplikować kodu i wykorzystać sprawne rozwiązanie

4.2 Napotkane problemy

- Ilość klientow
  Z racji posiadanych środków logistycznie zostały przeprowadzone testy na maksymalnie trzech urządzeniach jednocześnie. W razie problemów z większą ilością użytkowników, proszony jest natychmiastowy kontakt z jednym z współtwórców aplikacji.

- Odswiezanie obrazu Częstotliwość odświeżania zadeklarowana przez producenta 0,5 Hz do 64 Hz po przeprowadzonych testach została zdementowana albowiem wartosci powyzej 2Hz wpływały wyłącznie negatywnie na czas odświeżania klatek.

- UDP a nie TCP Istnieją przesłanki iż wykorzystywany podczas projektu system dostarczania pakietów TCP nieznacznie spowalnia wyświetlanie obrazu lecz implementacja systemu UDP w sposób znaczący opóźniłaby dostarczenie gotowego produktu. Jest to jeden z problemów do poruszenia w przypadku rozwijania aplikacji.

- Interpolacja obrazu a wypelnionej canvy Dodanie obrazu z pliku pozwala używać gotowych implementacji interpolacji, które są zoptymalizowane pod kątem efektywności i jakości. Z racji iż obraz nie jest gotowym obrazem z rozszerzeniem .png a w naszym przypadku jest w każdej iteracji wypełniany kolorami piksel po pikselu, interpolacja musiałaby zostać wykonana własnoręcznie poprzez stworzenie odpowiedniej funkcji. Kod aplikacji zawiera już odpowiednio przygotowaną metodę lecz nie została ona w obecnej wersji przetestowana i zaimplementowana pod obecne warunki środowiska.

- Zrzuty ekranu i gify Jednym z problemów które napotkano podczas próby dokumentacji poczyniań był problem z przechwytywaniem ekranu. Instalacja oprogramowania jak: "peek"oraz "byzanz"i wielorazowe próby przechwycenia interfejsu użytkownika, kończyły się zapisem do pliku wyłącznie czarnego koloru. Choć wszystkie ustawienia aplikacji działały i możliwość zmiany czasu nagrywania, rozdzielczości,

kompresji czy też pola które będzie przchwytywane, żadno z powyższych nie wpływało na rezultat co zawiera nagranie. Instalacja oprogramowania "simplescreenrecorder"przy uruchamianiu aplikacji, jako pierwsze nakierowało na problem, wyświetlając komunikat o treści braku wsparcia dla sesji z systemem Wayland cyt. :"You are using a non-X11 window system (e.g. Wayland) which is currently not supported". Zapoznając się z dokumentacją oprogramowań które do tej pory zastosowano zaobserwowano trend w tego typu aplikacjach do wsparcia bazowo systemów opartych na X11. Wykorzystując kolejno:

- $sudo raspi−conifg
  - Advanced Options-> Wayland-> W1 x11 (Openbox window manager with X11 backend)
    Zamiast dotychczas
  - Advanced Options-> Wayland-> W2 Wayfire (Wayfire window manager with manager with Wayland backend)
- $ reboot
- $ echo $XDG_CURRENT_DESKTOP
  LXDE
- $ echo $XDG_SESSION_TYPE
  x11
  Wówczas każdy z poprzednich edytorów zaczął prawidłowo nagrywać

- Biblioteka do zapisu wideo i kodeki Początek problemu leżał już w znalezieniu odpowiedniej biblioteki, gdyż ffmeg która była zalecana przez społeczność, nie posiadała odpowiednich dla projektu build'ów. Co za tym idzie kod wcześniej przygotowany wykorzystujący ją, musiał zostać zmodernizowany gdyż sieciowa wersja plików źródłowych nie miała by prawa działania lokalnie bez internetu a lokalnie nie można jej było pobrać z racji praw autorskich. Sama implementacja tej biblioteki także budziła kontrowersje w zespole przez co zdecydowaliśmy się ją odsunąć i zastosować MediaStream recording API która bazowo zapisuje w formacje webm. Według samego producenta posiada ona wsparcie dla rozszerzenie .mp4 oraz kodeków h264 lecz po ich przypisaniu w inicjacji instancji, program się nie uruchamiał. Ku zdziwieniu ustawienia sprzeczne sobie tj inicjacja za pomoca webm a następnie przypisanie zmiennej rozszerzenia .mp4 oraz bazowe kodeki dla webm działają prawidłowo i obraz jest zapisywany zgodnie z oczekiwaniami.

4.3 Kontakt i osoby odpowiedzialne

- Budowa hardware, instalacja software, logika aplikacji - Michał Lubczyński: michlub322@student.polsl.pl
- Szata graficzna strony WWW - Andrzej Zagórski: andrzag533@student.polsl.pl
- Wykorzystane biblioteki, uruchomienie aplikacji - Eryk Szmyt: erykszm832@student.polsl.pl

Rozdział 5

Podsumowanie i wnioski

Praca nad aplikacją Framer.py była wyzwaniem, które pozwoliło na zdobycie doświadczenia w zakresie programowania aplikacji internetowych oraz integracji sprzętu z oprogramowaniem. Projekt umożliwił zastosowanie różnorodnych technologii i narzędzi, takich jak SocketIO, Flask, Adafruit, czy HTML Canvas, w celu stworzenia interaktywnej aplikacji do wizualizacji danych z kamery termowizyjnej MLX90640.
Projekt Framer.py jest udanym przykładem zastosowania technologii webowych do wizualizacji danych w czasie rzeczywistym. Praca nad projektem pozwoliła na zdobycie praktycznych umiejętności w obszarze programowania w języku Python, korzystania z różnych bibliotek i frameworków, oraz rozwiązywania problemów związanych z integracją sprzętu i oprogramowania.
Aplikacja Framer.py stanowi solidną podstawę do dalszego rozwoju. W przyszłości można by rozważyć dodanie nowych funkcjonalności, takich jak obsługa większej liczby klientów, poprawa algorytmów interpolacji obrazu, czy rozwinięcie opcji nagrywania wideo. Istotnym aspektem jest również dostosowanie aplikacji do różnych platform i systemów operacyjnych oraz testowanie jej działania w różnych środowiskach.
Ostatecznie, praca nad projektem Framer.py była wartościowym doświadczeniem, umożliwiającym zdobycie nowych umiejętności i pogłębienie wiedzy w obszarze programowania aplikacji internetowych i integracji sprzętu z oprogramowaniem.