Projekt

Roboty mobilne

Założenia projektowe

Track Observing Fully Independent Car
TOFIC

Skład grupy:
Patryk Szelewski, 241490
Daniel Śliwowski, 241166
Jakub Tomaszewski, 241576

Termin: ptTP17

Prowadzący:
dr inż. Wojciech Domski

8 maja 2020

Spis treści

1 Opis projektu

Celem projektu jest zaprojektowanie oraz wykonanie robota mobilnego zdolnego do autonomicznego przemieszczania się po makiecie ulicy. Robot reagować będzie na pasy ruchu na drodze oraz wybrane znaki drogowe (znak STOP oraz sygnalizację świetlną). Całość zrealizowana zostanie w oparciu o minikomputer Raspberry Pi 3B+.

2 Założenia projektowe

Głównym założeniem projektu jest skonstruowanie autonomicznie przemieszczającego się po makiecie ulicy samochodu. Zakłada się implementację wykrywania pasa ruchu którym porusza się pojazd oraz algorytm sterowania mający za zadanie pozostanie na nim. Ponadto przewiduje się zaprojektowanie oraz wdrożenie algorytmów identyfikacji oznaczeń drogowych takich jak znak ŚTOPóraz sygnalizacja świetlna

3 Urządzenia zewnętrzne

3.1 Raspberry Pi 3 model B+

Mikrokomputer Raspberry Pi 3 model B+ charakteryzuje się następującymi parametrami:

- Rdzeń procesora Quad-Core ARM Cortex-A53
- Taktowanie $1,4GHz$
- Architektura $ARMv8-A$
- Pamięć RAM 1 GB LPDDR2 @ 900 MHz

Dzięki mocy mikrokomputera możliwe jest wykonywanie wielu obliczeń w czasie rzeczywistym.

3.2 Google Edge TPU ARM Accelerator

Edge TPU to mały układ ASIC zaprojektowany przez Google, który zapewnia wysoką wydajność wnioskowania uczenia maszynowego przy niskim poborze energii. Na przykład, może wykonywać najnowocześniejsze mobilne modele wizyjne, takie jak MobileNet v2 przy 100+ fps, w energooszczędny sposób. Posiada również dostępne do pobrania oprogramowanie TensorFlow lite.

3.3 Serwo Feetch FS5103B

Serwomechanizm sterowany jest za pomocą sygnały PWM. Zakres jego ruchów wynosi 180 stopni. Zasilany jest napięciem stałym $4,8-6V$

3.4 Kamera Waveshare Camera HD OV5647

Moduł kamery z sensorem OV5647 podłączany do dedykowanego złącza minikomputera Raspberry Pi w wersji 4B, 3B+, 3B, 3, 2, B+, A+ oraz starszych A i B. Urządzenie posiada matrycę o rozdzielczości 5 Mpx, wspiera tryb HD 1080 px.

4 Projekt elektroniki

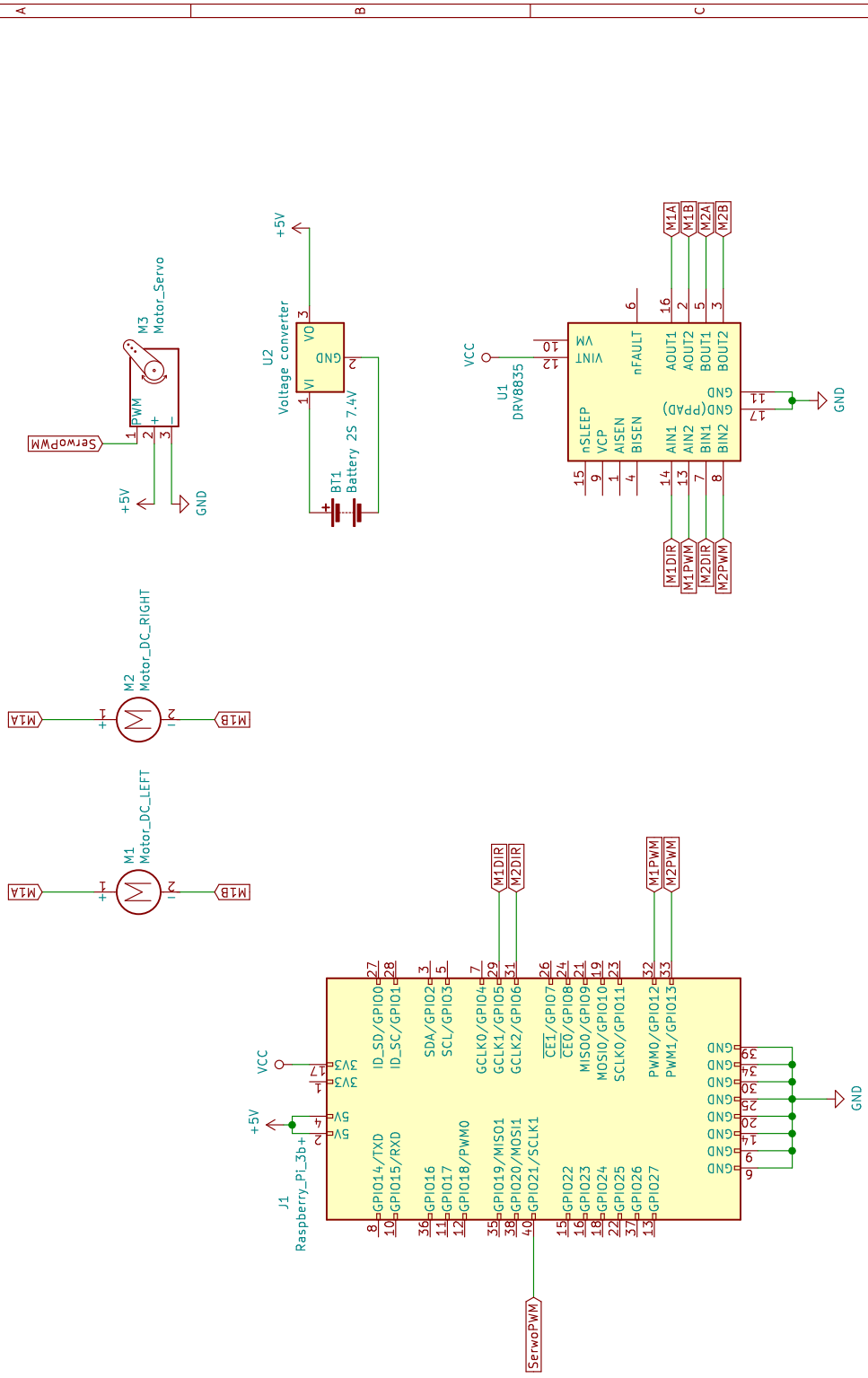
Udało się zrealizować wszystkie zadania związane z częścią elektroniczną robota. Poniżej przedstawiono wyniki wykonanych prac:

4.1 Spis użytych elementów

- Raspberry Pi 3 model B+ [6]
- Silnik DC Dagu DG02S-L 2szt
- Wskaźnik napięcia Li-pol 1-8S z buzzerem P309

- Taśma Raspverry Pi - kamera 50cm
- Koła Dagu RW002 65x26mm do silników DG - 4szt.
- Kamera Waveshare Camera HD OV5647
- Serwo Feetch FS5103B
- Pakiet Li-Pol Redox 2200mAh 30C 2S 7.4V
- Pololu DRV8835 - dwukanałowy sterownik silników 11V/1,2A
- Przetwornica napięcia - ładowarka 2x USB 5V
- Google Edge TPU ARM Accelerator

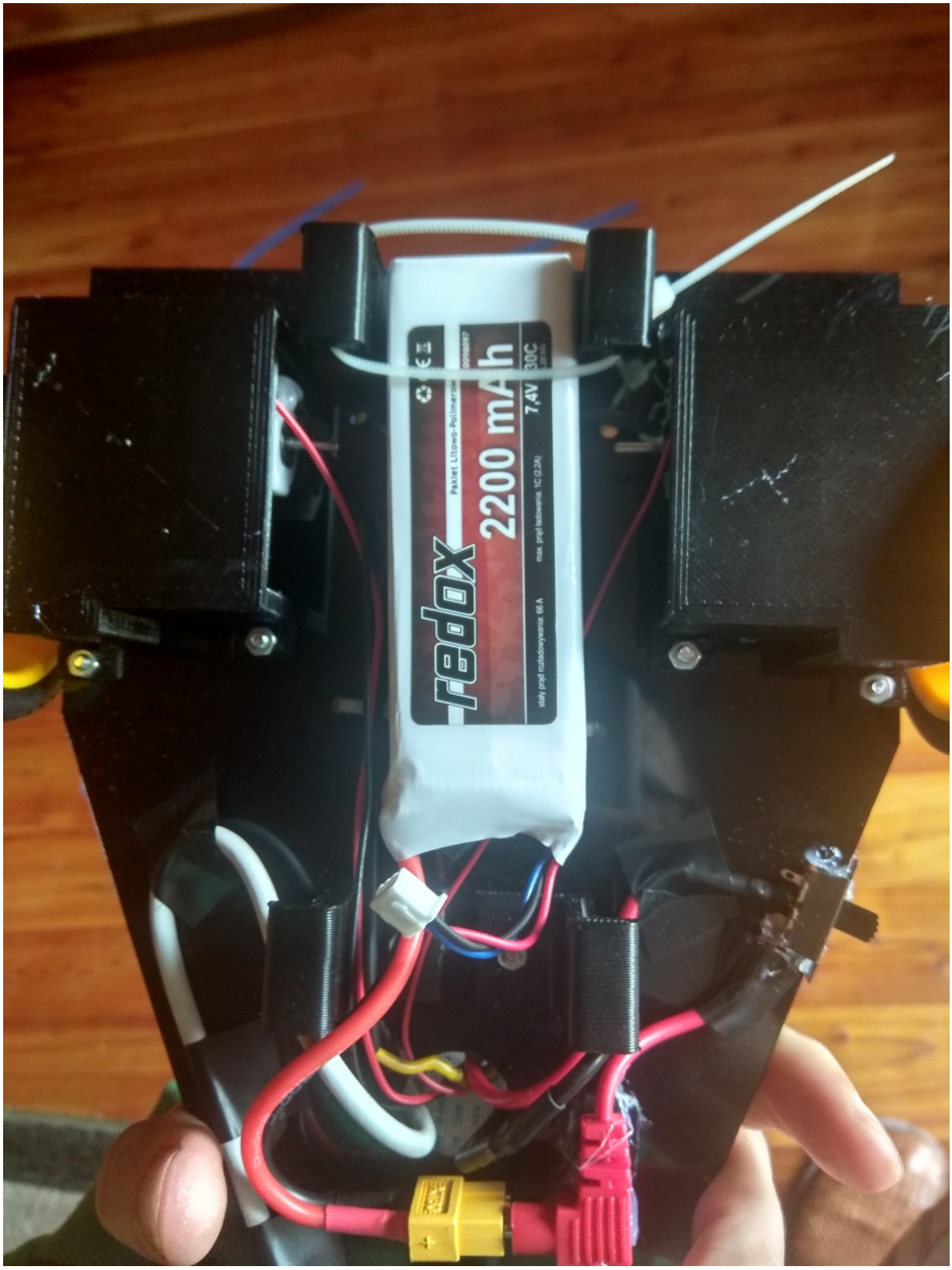
4.2 Schemat połączeń elektrycznych



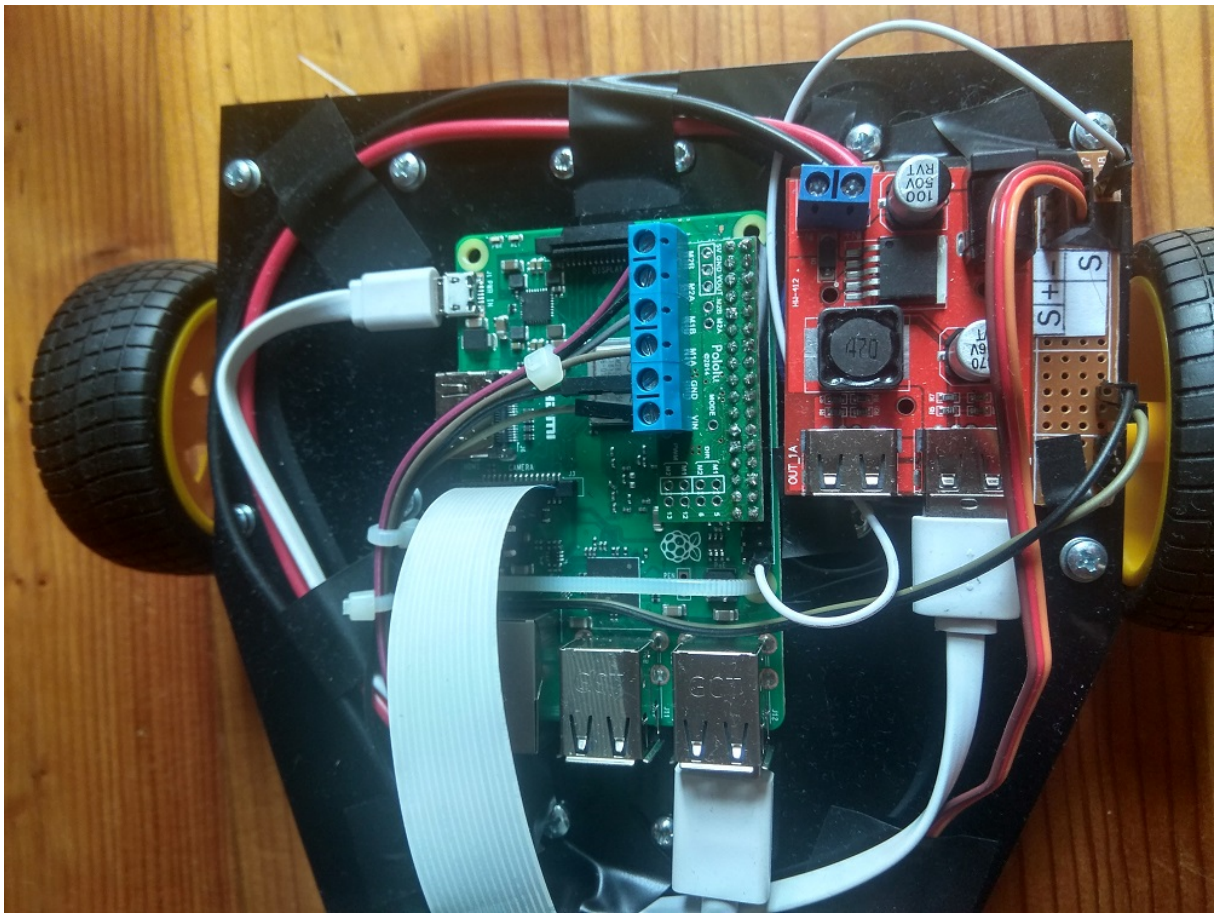
Rysunek 1: Schemat elektryczny robota mobilnego

4.3 Wykonane połączenia elektroniczne

Rezultaty wykonanych połączeń elektronicznych zamieszczono na



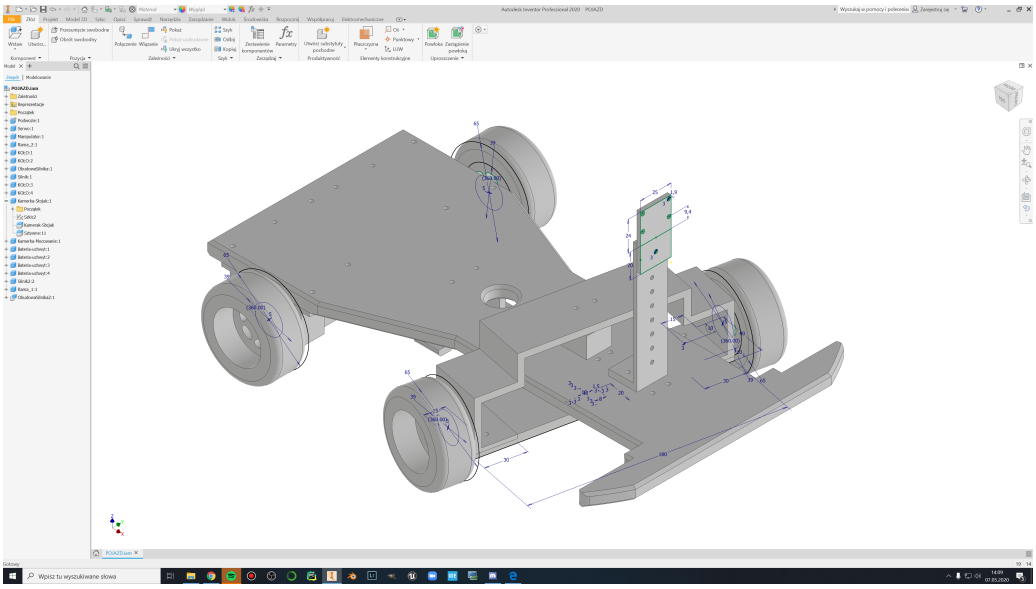
Rysunek 2: Gotowe połączenia elektroniczne w robocie mobilnym



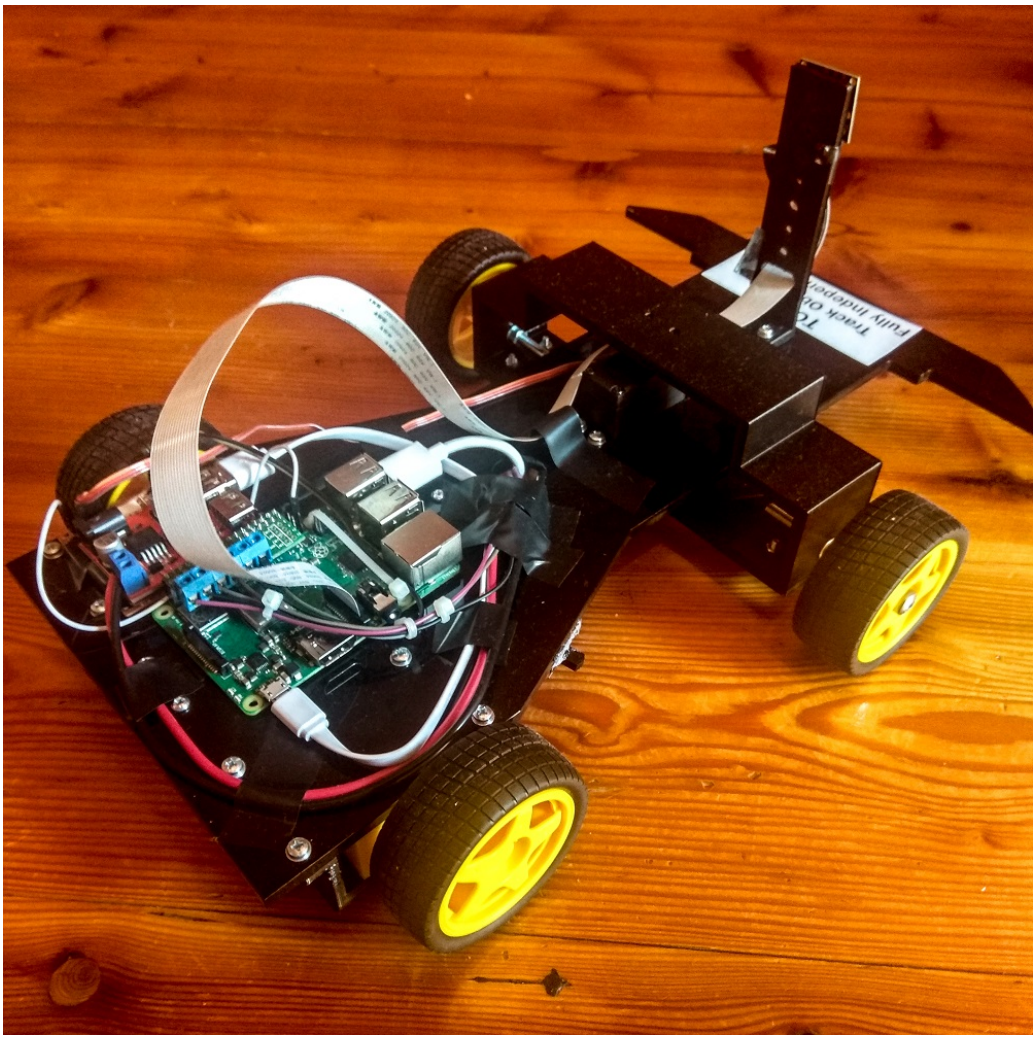
Rysunek 3: Gotowe połączenia elektroniczne w robocie mobilnym

5 Konstrukcja mechaniczna

Konstrukcja Pojazdu składa się z 10 wydrukowanych na drukarce 3D elementów z materiału `PLA`, oraz jednego elementu (podwozia pojazdu) wyciętego laserowo w `plexi`. Całość konstrukcji mechanicznej zaprojektowana została w programie Autodesk Inventor [4]. Udało się zrealizować wszystkie przewidziane zadania dla kamienia milowego `Mechanika`, co widać na Rys. 4 oraz Rys. 5. Konstrukcję mechaniczną pojazdu uznaje się za zakończoną.



Rysunek 4: Model 3D pojazdu w programie Autodesk Inventor

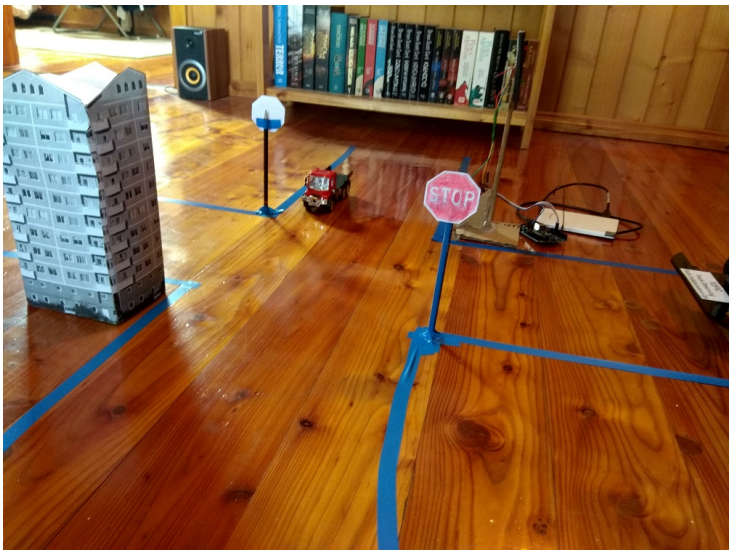


Rysunek 5: Gotowa konstrukcja mechaniczna pojazdu

6 Makieta drogi i znaki

W ramach tego etapu należało zbudować makietę drogi oraz znak stopu i sygnalizacje świetlną. Udało się zrealizować te zadanie czego dowodem są zdjęcia na Rys. 6, Rys. **??**. Sygnalizacja świetlna została wykonana za pomocą płytki Arduino.

Rysunek 6: Makieta drogi



Rysunek 7: Znak stopu

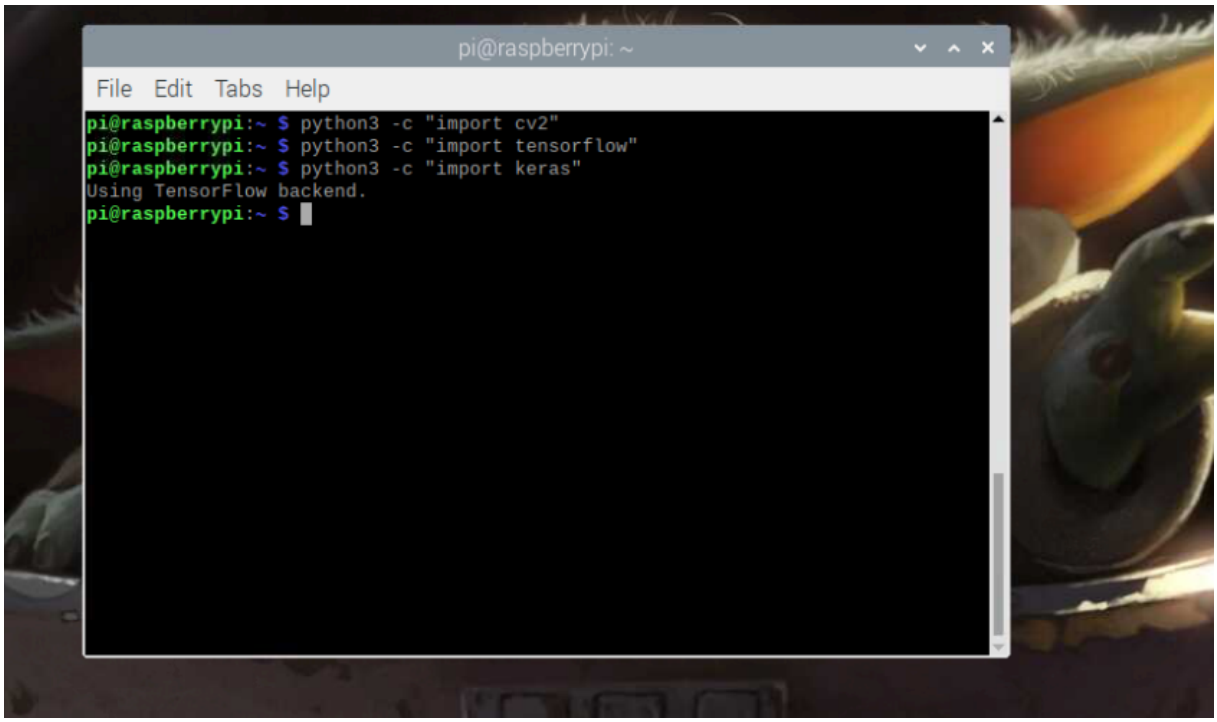
Rysunek 8: Światła drogowe

7 Konfiguracja RaspberryPi

W ramach tego zadania należało zainstalować wszelkie biblioteki służące do obsługi kamery i sterownika silników. Ponadto napisano skrypty pozwalające na sterowanie serwomechanizmem oraz ręczne sterowanie robotem.

7.1 OpenCV

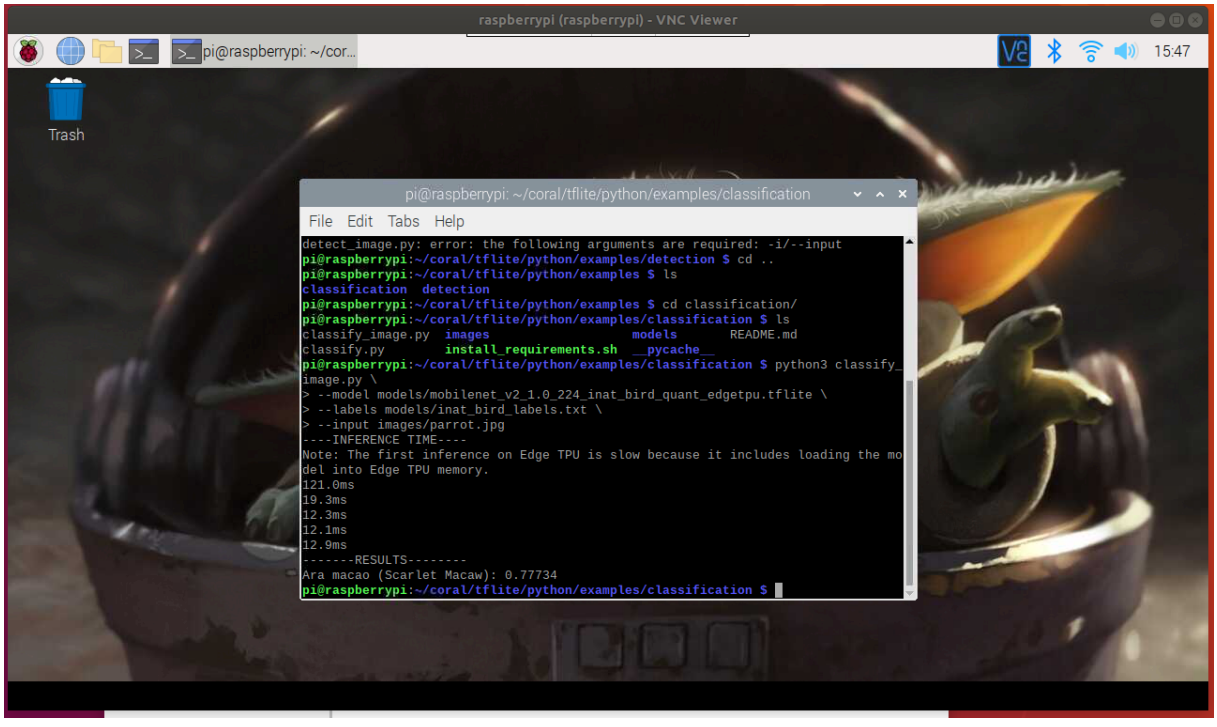
W celu przetwarzanie obrazu z kamery zainstalowano bibliotekę `OpenCV`, która jest zbiorem funkcji pozwalających na odczytywanie obrazu z kamery i jego przetwarzanie. Poprawne zainstalowanie biblioteki obrazuje Rys. 9



Rysunek 9: Poprawne importowanie `OnepCV`

7.2 TensorFlow i TensorFlow Lite

W celu tworzenia sieci neuronowej [1], a następnym uruchomieniu inferencji należało zainstalować bibliotekę do uczenia głębokiego `TensorFlow` [3] oraz bibliotekę pozwalającą na skorzystanie za zakupionym TPU [2]. Na Rys. 10 znajdują się zrzut ekranu z konsoli pokazujący poprawne działanie inferencji przeprowadzonej przez testowy program załączony z urządzeniem.



Rysunek 10: Przeprowadzona inferencja

7.3 Sterownik silników i serwa

Pobrano i skonfigurowano sterownik serwa. Ponadto napisano skrypty, który pozwala na ustawienie serwa w zadanej pozycji. Działanie obu skryptów obrazuje filmik: .

7.4 RealVNC Viewer i Server

W związku z obecną sytuacją postanowiono skorzystać z oprogramowania RealVNC Viewer i Server, które pozwalają na zdalny dostęp do Raspberry. Tym sposobem przy koordynacji członków zespołu możliwy jest zdalny dostęp do pulpitu urządzenia. Wadą tego rozwiązania jest to, że tylko jedna osoba na raz może kontrolować myszą. Zrzut ekranu przedstawiający ten program znajduje się na Rys. 10. W chwili robienia zdjęcia Raspberry znajdował się w innym domu niż komputer z którego robiono zdjęcie.

7.5 Sterowanie ręczne

W celu sprawdzenia działania robota napisano prosty skrypt korzystający z zainstalowanych bibliotek, który pozwalał na ręczne sterowanie robotem za pomocą klawiatury. Działanie obrazuje filmik: .

8 System kontroli wersji

W celu realizacji projektu członkowie zespołu wykorzystują system kontroli wersji Git. Link do publicznego zdalnego repozytorium zamieszczonego w serwisie `GitHub`: *https://github.com/DSliwowski1/TOFIC*. W repozytorium zamieszczane na bieżąco są postępy w pracach nad projektem dokonywane przez członków zespołu.

9 Podsumowanie

W etapie II udało się zrealizować wszystkie etapy projektu. Przykładową jazdę pojazdu w serwisie YoutTube znaleźć można pod następującym linkiem: próbna jazda. Do sterowania pojazdem wykorzystano bibliotekę udostępnioną przez producenta sterownika silników [5]

Literatura

[1] F. Chollet. *Deep Learning. Praca z językiem Python i biblioteką Keras.* Helion, 2019.

[2] Coral. Instrukcja instalacji oprogramowania do obsługi Edge TPU. 2020.

[3] Google Brain Team. Dokumentacja API Tensorflow. 2020.

[4] A. Jaskulski. *Autodesk Inventor 2020 PL / 2020+.* Helion, 2019.

[5] Pololu Robotics and Electronics. Biblioteka do obsługi sterownika silników. 2020.

[6] Raspberry Pi Foundation. Dokumentacja Raspberry Pi. 2020.