POLITECHNIKA
RZESZOWSKA
im. IGNACEGO ŁUKASIEWICZA

Projekt z przedmiotu

Systemy wbudowane

Makieta sygnalizacji świetlnej z przejściem dla pieszych

Imię i nazwisko: Rafał Nazarko

Grupa laboratoryjna: L2

Numer indeksu: 163982

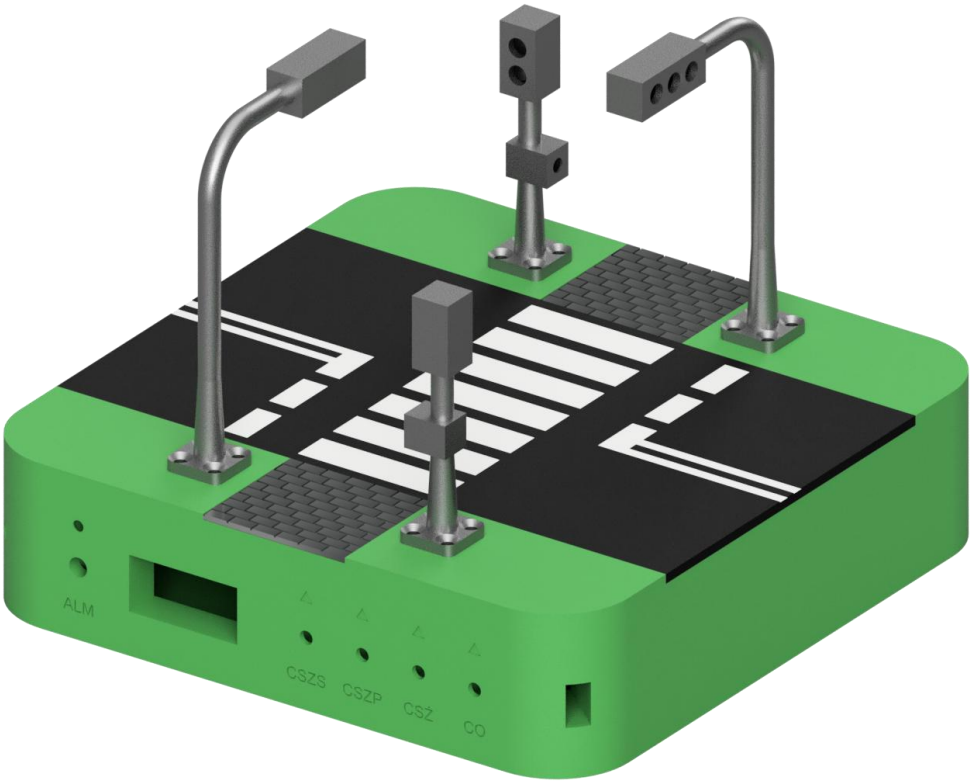
Wprowadzenie

Tematem projektu będzie stworzenie interaktywnej makiety sygnalizacji świetlnej na przejściu dla pieszych sterowanej przez mikrokontroler.

Założenia

Makieta będzie składać się z:

- 2 sygnalizacji dla samochodów (zielony, żółty, czerowny)
- 2 sygnalizacji dla pieszych z przyciskiem (zielony,czerwony)
- 1 wyświetlacza 4 cyfrowego 7 segmentowego z przyciskami
- 1 przełącznika uruchamiającego alarm
- Wydrukowanego w 3D modelu dwukierunkowej drogi z przejściem dla pieszych



Rysunek 1. Wizualizacja makiety

Układ będzie oparty o mikrokontroler Raspberry Pi Pico, który posiada odpowiednią ilość złącz GPIO aby móc podłączyć wszystkie komponenty. Zasilanie układu odbywać się będzie poprzez mikroport USB lub 3 baterie AAA. Z panelu operatora będzie można ustawić:

- CSZS - czas świecenia zielonego światła dla samochodów,
- CSZP - czas świecenia zielonego światła dla pieszych,
- CSŻ - czas świecenia światła żółtego,
- CO - czas opóźnienia,
- ALM - przełączyć sygnalizację w tryb alarmowy (w tym czasie wszystkie światła zarówno dla pieszych jak i samochodów są czerwone).

Działanie

Czas świecenia świateł oraz opóźnienia jest regulowany poprzez przyciski w panelu operatora. Czas może być dodany poprzez wciśnięcie przycisku ale tylko w przedziale od 1 sekundy do 9 sekund. Przekroczenie maksymalnej wartości czasu przywróci go do minimum. Zielone światło dla samochodów będzie się świecić, dopóki pieszy nie naciśnie przycisku na sygnalizacji świetlnej, ale nie krócej niż ustawiony czas świecenia zielonego światła. Po wciśnięciu przycisku na sygnalizacji dla pieszych, zostanie odmierzony czas opóźnienia, po którym rozpocznie się zmiana świateł. Po 2 sekundach od zmiany koloru światła dla samochodów na czerwone, światło dla pieszych zmieni się na zielone na czas równy czasowi trwania zielonego światła dla pieszych, które przez ostatnie 3 sekundy trwania będzie migać. Po upływie tego czasu odmierzone zostaną kolejne 2 sekundy, po których zaświeci się żółte światło dla samochodów. Podczas pierwszych n-sekund, gdzie n to minimalny czas świecenia zielonego światła, wciśnięcie przycisku dla pieszych nie spowoduje od razu odliczania czasu opóźnienia, ale zapisze się w pamięci i zostanie wywołane po upływie n-sekund od włączenia światła

zielonego dla samochodów. Ponadto w każdym momencie, sygnalizacja może przejść w stan alarmowy, w którym to wszystkie światła zostają zmienione na czerwone niezależnie od czasu trwania oraz aktualnego stanu. Po wyjściu z trybu przejazdu dla karetki, sygnalizacja startuje od początkowego stanu po upływie czasu opóźnienia.

Wykonanie

Pierwszym było skompletowanie potrzebnych komponentów elektronicznych według założeń. W ich skład wchodzi:

- 6x przycisków 6mm x 6mm
- 1x przełącznik dźwigniowy
- 1x przełącznik ON/OFF
- 1x wyświetlacz 4 cyfrowy 7 segmentowy
- 1x mikrokontroler Raspberry Pi Pico
- 4x diod zielonych, 4x diod czerwonych, 2 diod żółtych i 1 dioda niebieska
- 16 śrubek i nakrętek M3x10
- 2m skrętki ośmiożyłowej
- 10 rezystorów 220Ohm

Na podstawie wymiarów z zakupionych elementów stworzono modele 3D całej makiety aby następnie móc je wydrukować. Cały projekt 3D można zobaczyć na *Rysunku 1*. Proces drukowania zajął około 20 godzin i zużył łącznie około 300g materiału w kolorze srebrnym, szarym, czarnym i zielonym.

Po wydrukowaniu wszystkie części zostały ze sobą połączone. Dla lepszego zagospodarowania przewodami, pod wieczkiem makiety zostały przytwierdzone kostki zaciskowe, do wejść których zostały zgrupowane przewody odpowiadające za świecenie poszczególnych diod w sygnalizatorach, wejścia i wyjścia przycisków oraz uziemienie. W poniższej tabeli zostały zaprezentowane wejścia/wyjścia ogólnego przeznaczenia

mikrokontrolera, do których zostały przyporządkowane poszczególne sygnały.

Tabela 1. Wykaz sygnałów i wejść/wyjść mikrokontrolera

Pin GPIO	Sygnał
0	Segment E wyświetlacza 7 segmentowego
1	Segment D wyświetlacza 7 segmentowego
2	Kropka w wyświetlaczu 7 segmentowym
3	Segment C wyświetlacza 7 segmentowego
4	Segment G wyświetlacza 7 segmentowego
5	Cyfra 4 wyświetlacza 7 segmentowego
6	Cyfra 1 wyświetlacza 7 segmentowego
7	Segment A wyświetlacza 7 segmentowego
8	Segment F wyświetlacza 7 segmentowego
9	Cyfra 2 wyświetlacza 7 segmentowego
10	Cyfra 3 wyświetlacza 7 segmentowego
11	Segment B wyświetlacza 7 segmentowego
12	Przełącznik alarmowy
13	Przycisk zmiany czasu świecenia zielonego światła dla samochodów
14	Przycisk zmiany czasu świecenia zielonego światła dla pieszych
15	Przycisk zmiany czasu świecenia żółtego światła
16	Przycisk zmiany czasu opóźnienia
17	Zielone światło dla pieszych
18	Czerwone światło dla pieszych
19	Zielone światło dla samochodów
20	Żółte światło dla samochodów
21	Czerwone światło dla samochodów
22	Przycisk oczekiwania pieszego
26	Brzęczyk

Z racji na napięcie operacyjne mikrokontrolera wynoszące 3.3V, brzęczyk cechuje się niewielką głośnością a światła (w szczególności zielone) nie są maksymalnie jasne. Napięciem wystarczającym do odpowiedniego działania wymienionych komponentów jest 5V, niemożliwe jednak do osiągnięcia na użytym mikrokontrolerze bez dodatkowych rozszerzeń.

Przebieg czasowy i graf stanów

Pierwszym

Program

Program został napisany w języku MicroPython – rozwiązaniu dedykowanym do programowania mikrokontrolera Raspberry Pi Pico.

Listing 1. Program wgrany na mikrokontroler

```
# Import required libraries
from machine import Pin
import time

czasOdswiezaniaWyswietlacza = 0.006

CzasSwieceniaZielonegoDlaSamochodow = 7
CzasSwieceniaZielonegoDlaPieszych = 5
CzasSwieceniaZoltego = 2
CzasOpoznienia = 2

pinyPrzyciskow = [12,   13,   14,   15,  16, 22]
# Zmienne:        ALM,  CSZS, CSZP, CSZ, CO, BUTTON
stanyPrzyciskow = []

pinyCyfr = [6,9,10,5]
# Cyfry:    1,2, 3,4

pinySegmentow = [7,11,3,1,0,8,4]
#Lista ref:      A, B,C,D,E,F,G

pinySygnalizacji = {
    "ZIEL_S": 19,
    "ZOLT_S": 20,
    "CZER_S": 21,
    "ZIEL_P": 17,
    "CZER_P": 18,
    "BUZZER": 26
}

pinKropki = 2
gdzieKropki = [0,0,0,0]
alarm = 0
stan = 1
tim1 = 0
tim2 = 0
cykle = 0
czasCyklu = 1 #ms
czyOczekujePieszy = 0

# ustawia wszystkie piny przyciskow jako wyjście
for pin in pinyPrzyciskow:
    button = Pin(pin,Pin.IN, Pin.PULL_DOWN)
    stanyPrzyciskow.append(button.value())
    if pin == 12: alarm = stanyPrzyciskow[0]

# ustawia wszystkie piny wyświetlacza jako wyjście
for pin in (pinySegmentow + pinyCyfr + [pinKropki]):
    Pin(pin,Pin.OUT)

# ustawia wszystkie piny sygnalizacji jako wyjscie
for key in pinySygnalizacji:
```

```
    Pin(pinySygnalizacji[key],Pin.OUT)


tablicaZnakow = {
    "0": [0,0,0,0,0,0,1],
    "1": [1,0,0,1,1,1,1],
    "2": [0,0,1,0,0,1,0],
    "3": [0,0,0,0,1,1,0],
    "4": [1,0,0,1,1,0,0],
    "5": [0,1,0,0,1,0,0],
    "6": [0,1,0,0,0,0,0],
    "7": [0,0,0,1,1,1,1],
    "8": [0,0,0,0,0,0,0],
    "9": [0,0,0,0,1,0,0],
    "A": [0,0,0,1,0,0,0],
    "L": [1,1,1,0,0,0,1],
    "o": [1,1,0,0,0,1,0],
    "O": [0,0,1,1,1,0,0]
}

def wyswietl4Znaki(znaki,kropkiONlista):
    global alarm

    for idx, znak in enumerate(znaki[0:4]):
        Pin(pinyCyfr[0]).value(1 if idx == 0 else 0)
        Pin(pinyCyfr[1]).value(1 if idx == 1 else 0)
        Pin(pinyCyfr[2]).value(1 if idx == 2 else 0)
        Pin(pinyCyfr[3]).value(1 if idx == 3 else 0)
        Pin(pinKropki).value(not kropkiONlista[idx])
        if ((stan in [1,2] and idx == 0) or (stan in [4,10] and idx == 2) or (stan in [3] and idx ==
3)) and not alarm:
            for idj, bit in enumerate(tablicaZnakow[str(pozostaleSekundyTimera(tim1))]):
                Pin(pinySegmentow[idj]).value(bit)
        elif (stan in [6,7,8] and idx == 1) and not alarm:
            for idj, bit in enumerate(tablicaZnakow[str(pozostaleSekundyTimera(tim2))]):
                Pin(pinySegmentow[idj]).value(bit)
        else:
            for idj, bit in enumerate(tablicaZnakow[znaki[idx]]):
                Pin(pinySegmentow[idj]).value(bit)

        time.sleep(czasOdswiezaniaWyswietlacza)

def obslugaPrzyciskow(przycisk,wartosc):
    global CzasSwieceniaZielonegoDlaSamochodow
    global CzasSwieceniaZielonegoDlaPieszych
    global CzasSwieceniaZoltego
    global CzasOpoznienia
    global alarm
    global stan
    global czyOczekujePieszy

    if przycisk == 0:
        alarm = 1 if wartosc else 0
    if przycisk == 1 and not alarm and not stan in [1,2]:
        if wartosc:
            CzasSwieceniaZielonegoDlaSamochodow = CzasSwieceniaZielonegoDlaSamochodow+1 if
CzasSwieceniaZielonegoDlaSamochodow < 9 else 1
    if przycisk == 2 and not alarm and not stan in [6,7,8]:
        if wartosc:
            CzasSwieceniaZielonegoDlaPieszych = CzasSwieceniaZielonegoDlaPieszych+1 if
CzasSwieceniaZielonegoDlaPieszych < 9 else 1
    if przycisk == 3 and not alarm and not stan in [4,10]:
        if wartosc:
            CzasSwieceniaZoltego = CzasSwieceniaZoltego+1 if CzasSwieceniaZoltego < 9 else 1
    if przycisk == 4 and not alarm and not stan in [3]:
        if wartosc:
            CzasOpoznienia = CzasOpoznienia+1 if CzasOpoznienia < 9 else 1
    if przycisk == 5:
        if wartosc:
            czyOczekujePieszy = 1


def wyswietlAktualneZmienne():
    if not alarm:
```

```
wyswietl4Znaki((str(CzasSwieceniaZielonegoDlaSamochodow)+str(CzasSwieceniaZielonegoDlaPieszych)+str(Cza
sSwieceniaZoltego)+str(CzasOpoznienia)), gdzieKropki)
    else:
        wyswietl4Znaki("oALO" if cykle < 5 else "OALo", [0,0,0,0])

def ustawTimerNaSekundy(sekundy):
    return time.ticks_ms() + (sekundy * 1000)

def sprawdzCzyTimerAktywny(timer):
    return time.ticks_ms() < timer

def pozostaleSekundyTimera(timer):
    aktualnyTick = time.ticks_ms()
    return round((timer-aktualnyTick) / 1000) if (timer-aktualnyTick) > 0 else 0

def obslugaStanow():
    global CzasSwieceniaZielonegoDlaSamochodow
    global CzasSwieceniaZielonegoDlaPieszych
    global CzasSwieceniaZoltego
    global CzasOpoznienia
    global alarm
    global stan
    global tim1
    global tim2
    global gdzieKropki
    global cykle
    global czyOczekujePieszy

    if stan == 1:
        Pin(pinySygnalizacji["ZIEL_S"]).value(1)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(0)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [1,0,0,0]

        if not sprawdzCzyTimerAktywny(tim1): stan = 2
        elif alarm:
            tim1 = ustawTimerNaSekundy(CzasOpoznienia)
            stan=3

    elif stan == 2:
        Pin(pinySygnalizacji["ZIEL_S"]).value(1)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(0)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [1,0,0,0]

        if stanyPrzyciskow[5] or alarm or czyOczekujePieszy:
            tim1 = ustawTimerNaSekundy(CzasOpoznienia)
            stan=3

    elif stan == 3:
        Pin(pinySygnalizacji["ZIEL_S"]).value(1)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(0)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [0,0,0,1]

        if not sprawdzCzyTimerAktywny(tim1) and 1:
            tim1 = ustawTimerNaSekundy(CzasSwieceniaZoltego)
            stan=4

    elif stan == 4:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(1)
        Pin(pinySygnalizacji["CZER_S"]).value(0)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [0,0,1,0]
```

```
        if not sprawdzCzyTimerAktywny(tim1) and 1:
            tim1 = ustawTimerNaSekundy(2)
            stan=5

    elif stan == 5:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [0,0,0,0]

        if not sprawdzCzyTimerAktywny(tim1) and not alarm:
            tim1 = ustawTimerNaSekundy(CzasSwieceniaZielonegoDlaPieszych-3)
            tim2 = ustawTimerNaSekundy(CzasSwieceniaZielonegoDlaPieszych)
            czyOczekujePieszy = 0
            stan=6
        elif not sprawdzCzyTimerAktywny(tim1) and alarm:
            stan=11

    elif stan == 6:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(1)
        Pin(pinySygnalizacji["CZER_P"]).value(0)
        Pin(pinySygnalizacji["BUZZER"]).value(cykle%10)
        gdzieKropki = [0,1,0,0]

        if not sprawdzCzyTimerAktywny(tim1):
            tim1 = ustawTimerNaSekundy(0.5)
            stan=7
        elif alarm: stan=11

    elif stan == 7:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(1)
        Pin(pinySygnalizacji["CZER_P"]).value(0)
        Pin(pinySygnalizacji["BUZZER"]).value(cykle%5)
        gdzieKropki = [0,1,0,0]

        if not sprawdzCzyTimerAktywny(tim1):
            tim1 = ustawTimerNaSekundy(0.5)
            stan=8
        elif alarm: stan=11
        elif not sprawdzCzyTimerAktywny(tim2):
            tim1 = ustawTimerNaSekundy(2)
            stan = 9

    elif stan == 8:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(0)
        Pin(pinySygnalizacji["BUZZER"]).value(cykle%5)
        gdzieKropki = [0,1,0,0]

        if not sprawdzCzyTimerAktywny(tim1):
            tim1 = ustawTimerNaSekundy(0.5)
            stan=7
        elif alarm: stan=11
        elif not sprawdzCzyTimerAktywny(tim2):
            tim1 = ustawTimerNaSekundy(2)
            stan = 9

    elif stan == 9:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
```

```
        gdzieKropki = [0,0,0,0]

        if not sprawdzCzyTimerAktywny(tim1):
            tim1 = ustawTimerNaSekundy(CzasSwieceniaZoltego)
            stan=10
        elif alarm: stan=11

    elif stan == 10:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(1)
        Pin(pinySygnalizacji["CZER_S"]).value(0)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [0,0,1,0]

        if not sprawdzCzyTimerAktywny(tim1):
            tim1 = ustawTimerNaSekundy(CzasSwieceniaZielonegoDlaSamochodow)
            stan=1
        elif alarm: stan=11

    elif stan == 11:
        Pin(pinySygnalizacji["ZIEL_S"]).value(0)
        Pin(pinySygnalizacji["ZOLT_S"]).value(0)
        Pin(pinySygnalizacji["CZER_S"]).value(1)
        Pin(pinySygnalizacji["ZIEL_P"]).value(0)
        Pin(pinySygnalizacji["CZER_P"]).value(1)
        gdzieKropki = [0,0,0,0]

        if not alarm:
            tim1 = ustawTimerNaSekundy(2)
            stan=9

tim1 = ustawTimerNaSekundy(CzasSwieceniaZielonegoDlaSamochodow)
ostatniCykl = time.ticks_ms()
while True:
    for przycisk in range(0,len(pinyPrzyciskow)):
        if Pin(pinyPrzyciskow[przycisk]).value() and not stanyPrzyciskow[przycisk]:
            stanyPrzyciskow[przycisk] = 1
            obslugaPrzyciskow(przycisk,stanyPrzyciskow[przycisk])
        elif not Pin(pinyPrzyciskow[przycisk]).value() and stanyPrzyciskow[przycisk]:
            stanyPrzyciskow[przycisk] = 0
            obslugaPrzyciskow(przycisk,stanyPrzyciskow[przycisk])
    obslugaStanow()
    wyswietlAktualneZmienne()
    cykle = (cykle+1)%10
    while (time.ticks_ms()-ostatniCykl) < czasCyklu:
        pass
    ostatniCykl = time.ticks_ms()
    print(stan)
```