



**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

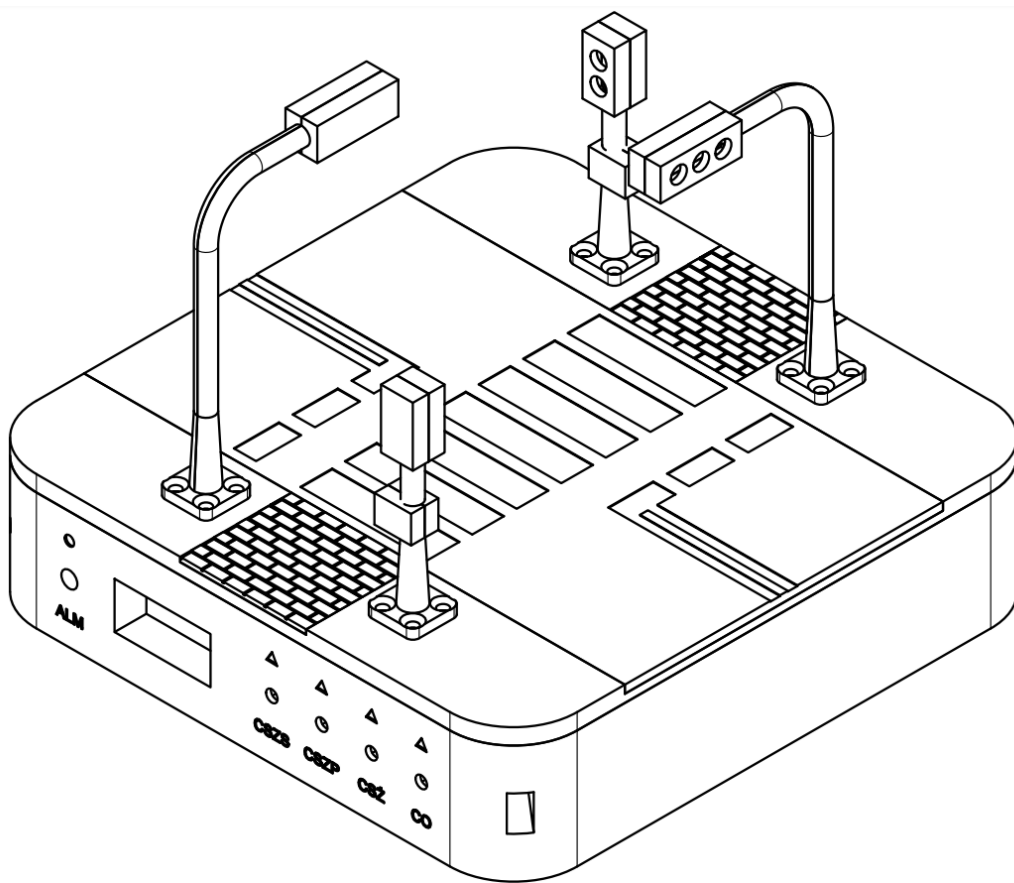
## Wprowadzenie

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, czerwony)
- 2 sygnalizacji dla pieszych z przyciskiem (zielony, czerwony)
- 1 wyświetlacza 4 cyfrowego 7 segmentowego z przyciskami
- 1 przetwornika 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 światła 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 światła. 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
- 8 magnesów neodymowych 2mm x 10mm

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.

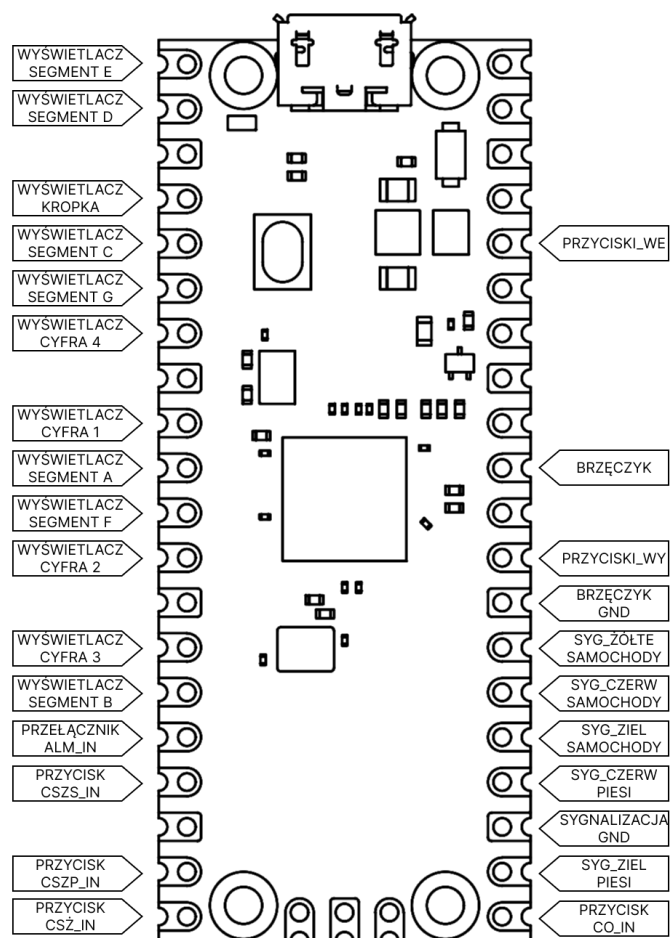
Po wydrukowaniu wszystkie części zostały ze sobą połączone. Dla lepszego zagospodarowania przewodami, pod wieczkiem makiety zostały przytwierdzone kostki zaciskowe, do wejść których zostały zgrupowane przewody odpowiadające za świecenie poszczególnych diod w sygnalizatorach, wejścia i wyjścia przycisków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

mikrokontrolera, do których zostały przyporządkowane poszczególne sygnały.

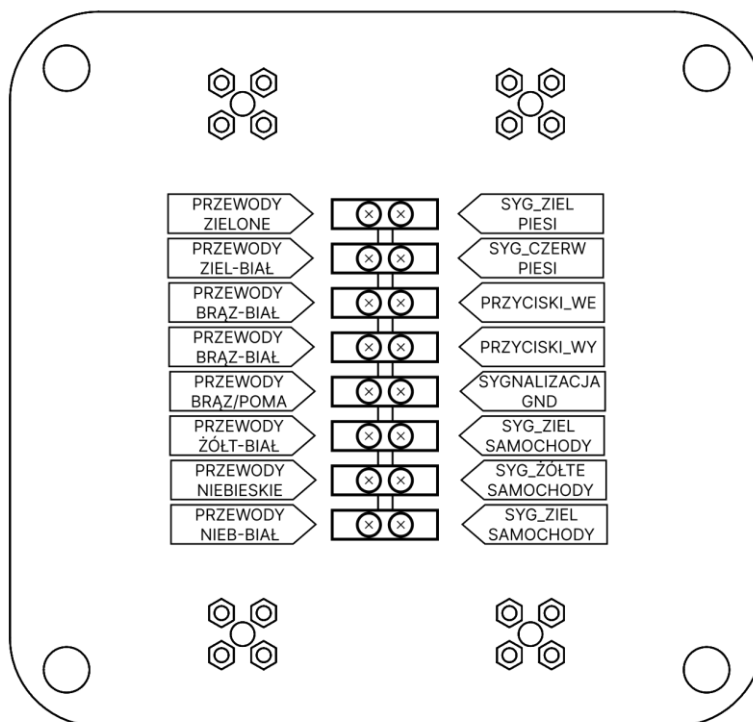
Tabela 1. Wykaz sygnałów i wejść/wyjść mikrokontrolera

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.

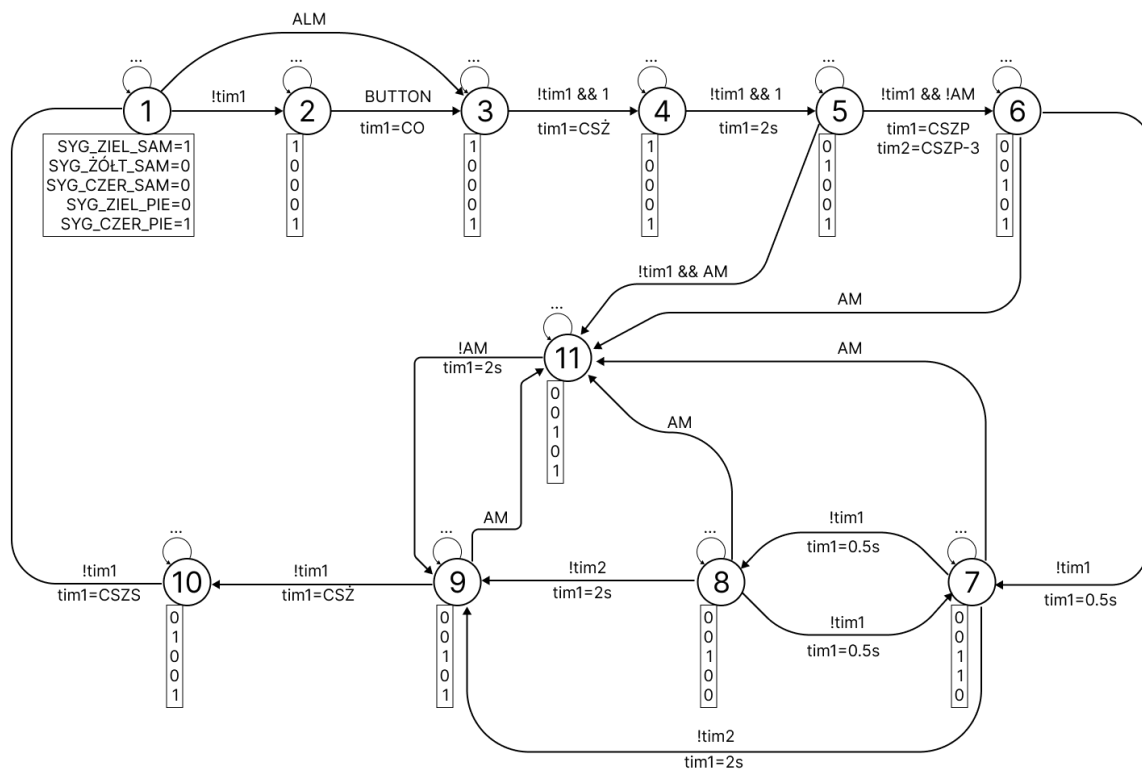


Rysunek 2. Wyprowadzenia mikrokontrolera



Rysunek 3. Aranżacja przewodów pod wieczkiem

## Graf stanów



Rysunek 4. Graf stanów

1. Światło zielone dla samochodów świeci max (CSZS)
2. Oczekiwanie na wciśnięcie przycisku dla pieszych
3. Oczekiwanie na zmianę światła dokładnie CO
4. Światło żółte przed czerwonym dokładnie CSŻ
5. Oczekiwanie dokładnie 2s na zielone dla pieszych
6. Zielone świeci dla pieszych max CSZP
7. Zielone miga max 0.5s (trwa odliczanie dokładnie 3s)
8. Zielone nie miga max 0.5s (trwa odliczanie dokładnie 3s)
9. Oczekiwanie max 2s na żółte
10. Światło żółte po czerwonym max CSŻ
11. ALARM – wszystkie światła czerwone

## Instrukcja obsługi

1. Przed rozpoczęciem korzystania z makiety należy ściągnąć wieko obudowy (górny zielony element) aby upewnić się, czy do układu są dołączone 3 sztuki baterii AA.
2. W kolejnym kroku należy przykręcić sygnalizatory za pomocą śrubokręta. Wszystkie potrzebne elementy znajdują się w tym samym opakowaniu co makietą. Każdy sygnalizator powinien być przymocowany za pomocą 4 śrub.
3. Po skończonych przygotowaniach powinno się zamknąć wieko w taki sposób, aby linia wyznaczona przez przejście dla pieszych i chodnik przecinała się z wyświetlaczem i przyciskami, które powinny znajdować się wprost przed osobą korzystającą z makiety.
4. W tym momencie makietą jest gotowa do użytku. Przed jej uruchomieniem warto zadbać aby przełącznik ALM był wyłączony (pozycja dolna). Aby włączyć makietę należy ustawić przełącznik zasilający znajdujący się na prawym narożniku panelu operatora w pozycję ON. Po dołączeniu zasilania do mikrokontrolera, automatycznie rozpocznie się praca sygnalizacji w stanie początkowym czyli kiedy światło zielone dla samochodów dopiero rozpoczęło działanie a pieszy nie nacisnął przycisku wybudzenia.
5. Na wyświetlaczu pokazywane są ustawione czasy kolejno CSZS, CSZP, CSŻ i CO. Poprzez wciskanie odpowiednich przycisków w panelu operatora można inkrementować poszczególne czasy w przedziale od 1 do 9 (wartości są wyrażone w sekundach). Świecąca się kropka obok którejś z wyświetlanych wartości informuje, że aktualnie sygnalizacja korzysta z tego czasu i nie możliwa jest jego zmiana. Ponadto na wyświetlaczu pokazywany jest pozostały czas do wyzerowania timera i przejścia do kolejnego stanu.



6. Załączenie przełącznika ALM spowoduje przejście sygnalizacji w tryb alarmowy. Wszystkie światła przejdą w odpowiedniej sekwencji do koloru czerwonego.
7. Wciśnięcie przycisku (wybudzenia) znajdującego się na sygnalizatorze dla pieszych spowoduje przejście sygnalizacji w tryb zielonego światła dla pieszych ale nie wcześniej niż przed upływem czasu świecenia zielonego światła dla samochodów (CSZS).
8. Kiedy użytkownik skończy korzystanie z makiety, należy odłączyć poprzez ustawienie przełącznika zasilającego w pozycję OFF. Można to zrobić w każdym momencie działania makiety.
9. Przed schowaniem makiety do pudełka, należy koniecznie odkręcić wszystkie śrubki utrzymujące sygnalizatory, a je same położyć poziomo na wieczku makiety. Nie należy wprowadzać żadnych zmian w połączeniach przewodów. Śrubki i śrubokręt należy schować do woreczka strunowego znajdującego się w zestawie a całość zabezpieczyć folią bąbelkową.

## 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
CzasSwieceniaZielonegoDlaPiesznych = 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
```



```

        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)
```

Cały program w wersji cyfrowej można znaleźć na repozytorium GitHub. Link został zamieszczony poniżej.

**<https://github.com/Rav-Naz/SW-Sygnalizacja>**



*Rysunek 5. Kod QR do repozytorium*