



TECHNISCH ONTWERP VERKEERSSIMULATIE

Versie 1 | Informatica Communicatie Academie | Nijmegen

02-04-2020

Embeded Software Development Docent: Ralph Niels

Ontwerp geschreven door: Lauren Meenhorst SN632206 Klas 1DC

Inhoud

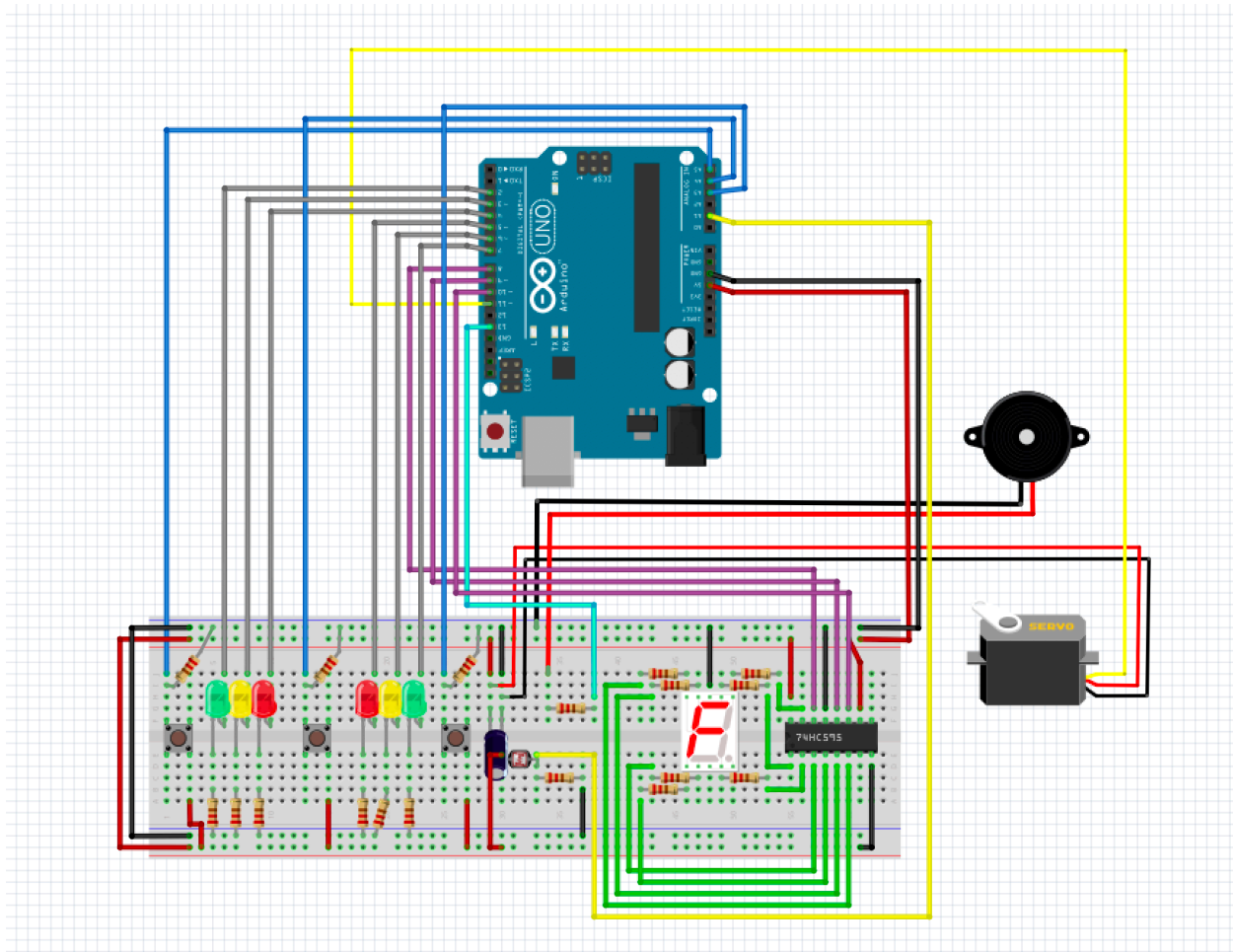
Inhoud	2
Inleiding.....	3
1. Aansluiting.....	4
3.1 Aansluitschema	4
3.2 onderdelen	4
2. Tabbladen.....	5
2.1 Tabbladen diagram.....	5
2.2 Tabbladen toelichting.....	6
2.2.1 BPEPDTrafficSimulationLauren.....	6
2.2.2 Barrier.....	6
2.2.3 Buttons	6
2.2.4 Buzzer	7
2.2.5 DeadTime	7
2.2.6 LDR	7
2.2.7 LEDControl.....	8
2.2.8 NightShift.....	8
2.2.9 PedestrianDisplay	8
2.2.10 Queue	9
2.2.11 ServoMotor	9
2.2.12 ShiftRegister	9
2.1.13 SimulationStateMachine	10
2.1.14 SimulationStateMachineActions	10
2.2.15 TrafficLights	11
3. Toestanden.....	12
3.1 Toestandsdiagram	12

Inleiding

Onderstaand document is een technische uitwerking van de specificaties die in het functioneel ontwerp zijn gespecificeerd.

1. Aansluiting

3.1 Aansluitschema



Figuur 1 Aansluitschema (Figuur is ook te vinden in de map diagrammen en schema's)

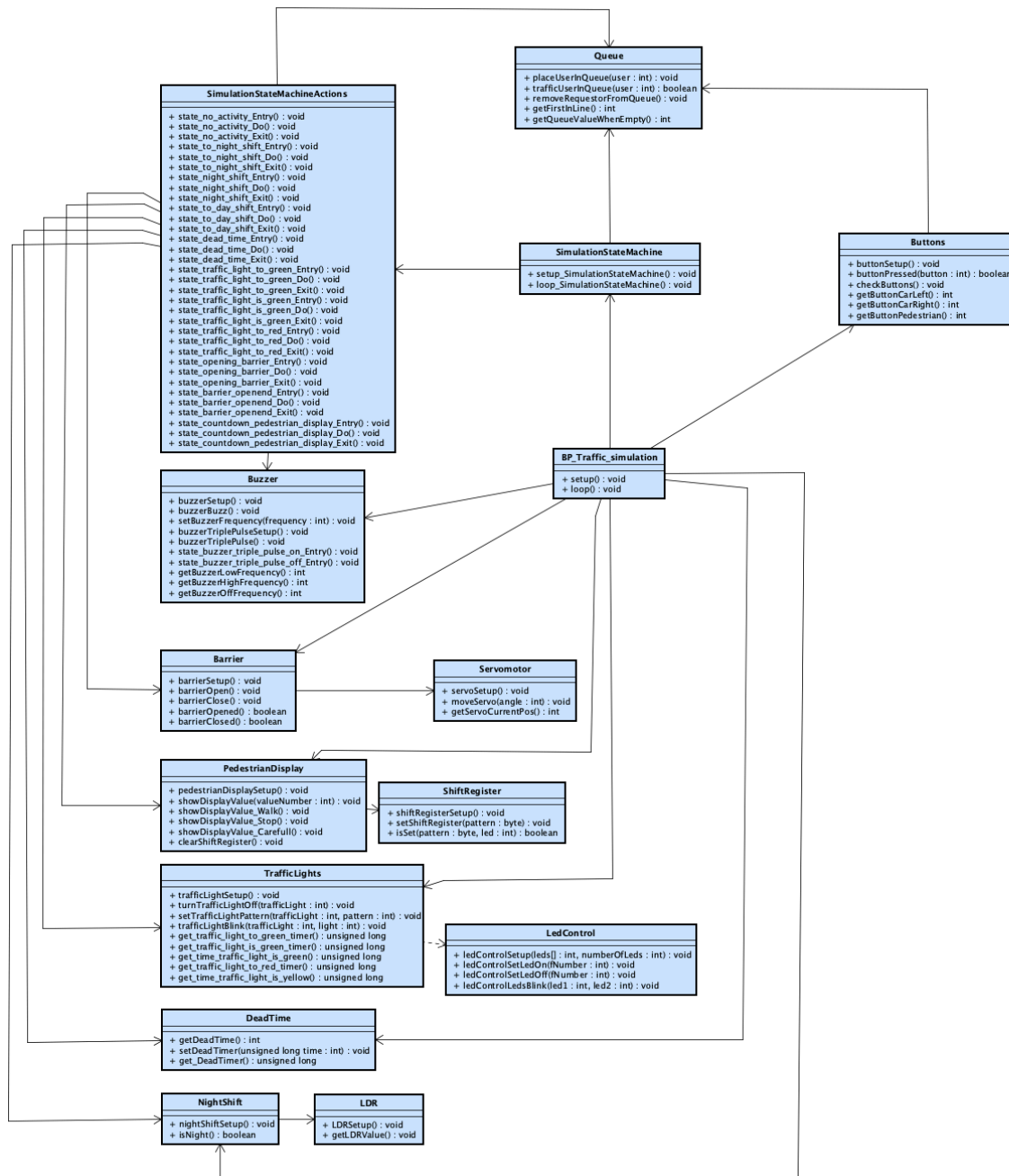
3.2 onderdelen

In bovenstaand model zijn de volgende elementen terug te vinden.

Type	Aantal	Gebruik
LED groen	2	Stoplicht 1 en 2
LED geel	2	Stoplicht 1 en 2
LED rood	2	Stoplicht 1 en 2
7-segment display	1	Voetgangers overgang licht
Shiftregister	1	Aansluiten 7-segment display
Servomotor	1	Voetgangers slagboom
Elco	1	Aansluiten servomotor
Buzzer	1	Tik state indicatie
LDR	1	Bepalen of het dag of nacht is
Push button	3	Vullen queue
220 Ω resistor	17	Aansluiten elementen
10 k Ω resistor	1	Aansluiten LDR

2. Tabbladen

2.1 Tabbladen diagram



Figuur 2 State tabbladen diagram (Figuur is ook te vinden in de map diagrammen en schema's)

2.2 Tabbladen toelichting

2.2.1 BPEPDTrafficSimulationLauren

Setup()	void	Voert alle setup functies die nodig zijn voor de loop uit.
Loop()	void	Controleert of het nacht is. Laat de buzzer geluid maken, kijkt of er op een knop gedrukt is en loopt door de state machine. Indien het nacht is worden de knoppen en buzzer functies niet uitgevoerd.

2.2.2 Barrier

Attributen

OPENED	const int	Hoek wanneer open.
CLOSED	const int	Hoek wanneer dicht.

Methoden

barrierSetup()	void	Roept de functie servoSetup aan.
barrierOpen()	void	Roept de functie moveServo aan en geeft de waarde mee van de hoek die de servomotor moet hebben wanneer deze geopend is.
barrierClose()	void	Roept de functie moveServo aan en geeft de waarde mee van de hoek die de servomotor moet hebben wanneer deze gesloten is.
barrierOpened()	boolean	Retourneert true indien de huidige servomotor positie gelijk is aan open.
barrierClosed()	boolean	Retourneert true indien de huidige servomotor positie gelijk is aan dicht.

2.2.3 Buttons

Attributen

NUMBER_OF_BUTTONS	const int	Het aantal buttons.
BUTTONS [NUMBER_OF_BUTTONS]	const int	De button pinarray.
BUTTON_CAR_LEFT	const int	De index van de knop die een auto van links representeert.
BUTTON_CAR_RIGHT	const int	De index van de knop die een auto van rechts representeert.
BUTTON_PEDESTRIAN	const int	De index van de knop die een voetganger representeert.
buttonPreviousMillis [NUMBER_OF_BUTTONS]	unsigned long	De button previous millis array.
DEBOUNCE_TIME	const int	De knop debounce tijd.

Methoden

buttonSetup()	void	Initialiseert de button pins als input pins.
buttonPressed()	boolean	Retourneert true indien een analog read van de button pin HIGH retourneert.
checkButtons()	Void	Loopt door de buttons. Met de functie hierboven wordt gecontroleerd of een button is ingedrukt hierna wordt gecontroleerd of de button langer ingedrukt is dan de debounce tijd. Indien dit zo is wordt het volgnummer van de button in de queue geplaatst.
getButtonCarLeft()	int	Retourneert het volgnummer van de button die verbonden is aan auto's die van links komen.
getButtonCarRight()	int	Retourneert het volgnummer van de button die verbonden is aan auto's die van rechts komen.
getButtonPedestrian()	int	Retourneert het volgnummer van de button die verbonden is aan voetgangers.

2.2.4 Buzzer

Attributen

BUZZER	const int	De buzzer pin.
BUZZER_LOW_FREQUENCY	const int	Buzzer frequentie wanneer hij langzaam moet tikken.
BUZZER_HIGH_FREQUENCY	const int	Buzzer frequentie wanneer hij snel moet tikken.
BUZZER_TRIPLE_PULSE_INTERVAL	const int	Buzzer interval wanneer hij 3 keer moet tikken.
BUZZER_TRIPLE_PULSE_FREQUENCY	const int	Buzzer frequentie wanneer hij 3 keer moet tikken.
BUZZER_OFF_FREQUENCY	const int	Buzzer frequentie wanneer hij stil moet zijn.
buzzerPreviousMillis	unsigned long	De buzzer previous millis.
buzzerFrequency	int	De huidige buzzer frequentie.

Methoden

setBuzzerFrequency(int frequency)	void	Maakt buzzerFrequency gelijk aan frequency.
buzzerSetup()	void	Initialiseert de buzzer pin als output pin.
buzzerBuzz()	void	Laat de buzzer geluid maken met de huidige frequentie.
buzzerTriplePulseSetup()	void	Maakt de huidige tripple pulse state gelijk aan aan en stuurt het programma naar de entry functie van de tripple pulse state machine.
buzzerTriplePulse()	void	Voert de tripple pulse state machine uit.
state_buzzer_triple_pulse_on_Entry()	void	Maakt de buzzer frequentie gelijk aan de buzzer drie keer tikken frequentie.
state_buzzer_triple_pulse_off_Entry()	void	Maakt de buzzer frequentie gelijk aan de buzzer uit frequentie.
getBuzzerLowFrequency()	int	Retourneert de lage buzzer frequentie waarde.
getBuzzerHighFrequency()	int	Retourneert de hoge buzzer frequentie waarde.
getBuzzerOffFrequency()	int	Retourneert de buzzer frequentie waarde voor stilte.

2.2.5 DeadTime

Attributen

DEAD_TIME	const int	Hoe lang dead time moet duren.
deadTimer	unsigned long	Hoe lang de timer bezig is.

Methoden

getDeadTimer	unsigned long	Retourneert de huidige waarde van deadTimer.
setDeadTimer (unsigned long time)	void	Maakt deadTimer gelijk aan time.
getDeadTime	int	Retourneert DEAD_TIME deze waarde is gelijk aan het aantal milliseconden dat state_dead_time_do() uitgevoerd moet worden.

2.2.6 LDR

Attributen

LDR_PIN	const int	De LDR pin.
MIN_VALUE	const int	De minimale waarde die gelezen kan worden door de LDR.
MAX_VALUE_ANALOG	const int	De maximale waarde die gelezen kan worden door de LDR
MAX_VALUE_RANGE	const int	Het bereik waarin de LDR waarde geformatteerd wordt.

Methoden

LDRSetup()	void	Initialiseert de LDR pin als input pin.
getLDRValue()	int	Maakt de huidige lichtwaarde en retourneert deze.

2.2.7 LEDControl

Methoden

ledControlSetup (int leds[], int number_of_leds)	void	Ontvangt een array van ledpins en zijn lengte om deze vervolgens te initialiseren als output pins.
ledControlSetLedOn (int led)	void	Schrijft ledwaarde naar high.
ledControlSetLedOff (int led)	void	Schrijft ledwaarde naar low.
ledControlLedsBlink (int led1, int led2)	void	Controleert of de knipper tijd voorbij is. Zo ja wordt de huidige waarde van de 2 leds omgezet naar de tegenovergestelde waarde.

2.2.8 NightShift

Attributen

MAX_NIGHT_LIGHT_VALUE	const int	De maximale waarde voor nacht.
MIN_DAY_LIGHT_VALUE	const int	De minimale waarde voor dag.

Methoden

nightShiftSetup()	void	Roept de LDRsetup functie aan.
isNight()	boolean	Retourneert true als de huidige LDR-waarde kleiner of gelijk is aan de maximale nacht lichtwaarde.

2.2.9 PedestrianDisplay

Attributen

NUMBER_OF_DISPLAYVALUES	const int	Het aantal waarden dat op het display wordt afgebeeld.
DISPLAYVALUES [NUMBER_OF_DISPLAYVALUES]	const byte	De waarden die op het display worden afgebeeld als bytes in een array.
countdownPedestrianDisplayTimer	unsigned long	Een timer die gebruikt wordt voor de countdown.
displayWalkTimer	unsigned long	Een timer die gebruikt wordt voor het tonen van de walk value.
timePedestriansCanWalk	int	Hoe lang voetgangers mogen lopen.
timeNeededForCountdown	int	Hoe lang de countdown duurt.

Methoden

pedestrianDisplaySetup	void	Roept de shiftRegisterSetup aan.
showDisplayValue(int valueNumber)	void	Roept de functie setShiftRegister aan en geeft deze valueNumber mee als patroon.
showDisplayValue_Walk()	void	Roept de functie setShiftRegister aan en geeft deze het loop patroon mee als patroon.
showDisplayValue_Stop()	void	Roept de functie setShiftRegister aan en geeft deze het stop patroon mee als patroon.
showDisplayValue_Carefull()	void	Roept de functie setShiftRegister aan en geeft deze het wees voorzichtig mee als patroon.
getDisplayWalkTimer()	unsigned long	Retourneert de huidige waarde van de walk timer.
setDisplayWalkTimer(unsigned long time)	void	Maakt de display walk timer gelijk aan time.
getTimePedestriansCanWalk()	int	Retourneert timePedestriansCanWalk.
setCountdownPedestrianDisplayTimer()	void	Maakt countdownPedestrianDisplayTimer gelijk aan millis() + de tijd die nodig is voor de countdown.
getCountdownPedestrianDisplayTimer()	unsigned long	Retourneert de huidige waarde van de countdown timer.

2.2.10 Queue

Attributen

QUEUE_POSITIONS	const int	Het aantal beschikbare plaatsen in de rij.
QUEUE_VALUE_WHEN_EMPTY	const int	De waarde van de rij positie wanneer deze leeg is.
queue[QUEUE_POSITIONS]	int	De array die de rij representeert.

Methoden

requestorInQueue(int requestor)	boolean	Loopt door alle requestors. Als hij er 1 tegen komt met dezelfde waarde als requestor tegenkomt retourneert hij true.
placeRequestorInQueue(int requestor)	void	Controleert of de requestor al in de wachtrij staat. Zo niet wordt hij op de eerste lege positie geplaatst.
removeRequestorFromQueue()	void	Verwijdert de eerste requestor in de rij en schuift alle anderen naar voren.
getFirstInLine()	int	Retourneert de eerste requestor in de rij.
getQueueValueWhenEmpty()	int	Retourneert QueueValueWhenEmpty.

2.2.11 ServoMotor

Attributen

servo	Servo	
SERVOPIN	const int	De servo pin.
SERVO_START_POS	const int	De start positie van de servo motor.
SERVO_TIME_INTERVAL	const int	De tijd interval van de servo.
SERVO_STEP_SIZE	const int	Het aantal graden dat de servo per interval beweegt.

Methoden

servoSetup()	void	Initialiseert de servo motor.
moveServo(int goalPos)	void	Wacht tot de servo interval voorbij is om de servo met stapgrootte te verlagen of verhogen. Dit wordt gedaan tot de gewenste positie is bereikt.
getServoCurrentPos()	int	Retourneert de huidige servo positie.

2.2.12 ShiftRegister

Attributen

DS_PIN	const int	De data serial pin.
SH_CP	const int	De clock pin.
ST_CP	const int	De latch pin.

Methoden

shiftRegisterSetup()	int	Initialiseert de DS, ST en SH pins als output pins.
setShiftRegister(byte pattern)	void	Loopt door een byte. SH wordt high. Als i isSet schrijft DS is high anders Ds is low. SH wordt low. Hierna wordt ST eerst high, daarna low.
isSet(byte pattern, int led)	boolean	Loop door een byte heen. Als i is 1 retourneer true anders false.

2.1.13 SimulationStateMachine

Attributen

STATE_NO_ACTIVITY	const int	De state wanneer er geen activiteit plaats vindt.
STATE_TO_NIGHT_SHIFT	const int	De state wanneer het systeem naar nachtstand overschakelt.
STATE_NIGHT_SHIFT	const int	De state waarin het systeem verkeerd wanneer het nacht is.
STATE_TO_DAY_SHIFT	const int	De state wanneer het systeem naar dagstand overschakelt.
STATE_DEAD_TIME	const int	De state waarin het systeem verkeerd wanneer er even niets gebeurt tussen activiteiten in.
STATE_TRAFFIC_LIGHT_TO_GREEN	const int	De state wanneer het systeem een verkeerslicht naar groen schakelt.
STATE_TRAFFIC_LIGHT_IS_GREEN	const int	De state waarin het systeem verkeerd wanneer een stoplicht groen licht toont.
STATE_TRAFFIC_LIGHT_TO_RED	const int	De state waarin het systeem verkeerd wanneer een stoplicht geel licht toont.
STATE_OPENING_BARRIER	const int	De state waarin het systeem verkeerd wanneer de slagboom aan het openen is.
STATE_BARRIER_OPENED	const int	De state waarin het systeem verkeerd wanneer de slagboom open is.
STATE_COUNTDOWN_PEDESTRIAN_DISPLAY	const int	De state waarin het systeem verkeerd wanneer zij aftelt van 6 naar 0 op het voetgangers display.
STATE_CLOSE_BARRIER	const int	De state waarin het systeem verkeerd wanneer de slagboom aan het sluiten is.
currentState	int	De huidige staat waarin het systeem zich verkeerd.
direction_requesting_passing	int	De eerste richting in de queue.

Methoden

getDirectionRequestingPassing()	int	Deze functie roept getFirstInLine() aan.
setDirectionRequestingPassing()	void	Deze functie maakt direction_requesting_passing gelijk aan getFirstInLine().
simulationStateMachineSetup()	void	Roept pedestrianDisplaySetup(), trafficLightSetup() en barrierSetup() aan. currentState wordt gelijk gemaakt aan STATE_NO_ACTIVITY en de entry functie van de state wordt aangeroepen.
simulationStateMachineLoop()	void	In deze functie staat een switch die voor de huidige state de Entry en Do functie aanroept. Als de state voorbij is wordt de Exit functie aangeroepen en currentState gelijk gemaakt aan de volgende state.

2.1.14 SimulationStateMachineActions

Dit tabblad heeft geen attributen. Het bevat alle entry, do en exit functies van de states.

2.2.15 TrafficLights

Attributen

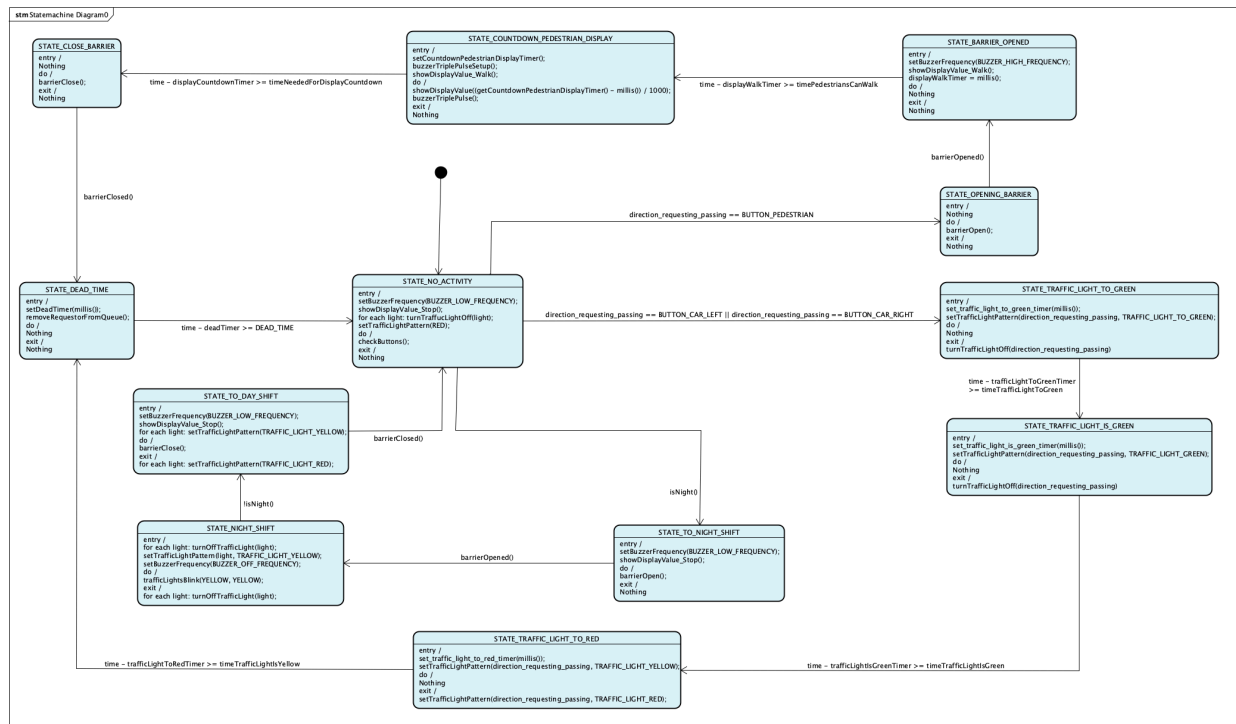
NUMBER_OF_TRAFFIC_LIGHTS	const int	De data serial pin.
NUMBER_OF_COLORS_PER_TRAFFIC_LIGHT	const int	De clock pin.
NUMBER_OF_TRAFFIC_LIGHT_PATTERNS	const int	De latch pin.
LEFT_TRAFFIC_LIGHT	const int	Het volgnummer van het linker verkeerslicht.
RIGHT_TRAFFIC_LIGHT	const int	Het volgnummer van het rechter verkeerslicht.
TRAFFICLIGHTS [NUMBER_OF_TRAFFIC_LIGHTS] [NUMBER_OF_COLORS_PER_TRAFFIC_LIGHT]	const int	Een tweedimensionale array met hierin de licht pinnen per verkeerslicht.
trafficLightToGreenTimer	unsigned long	Timer.
timeTrafficLightToGreen	const int	Hoe lang een verkeerslicht leeg is.
trafficLightIsGreenTimer	unsigned long	Timer.
timeTrafficLightIsGreen	const int	Hoe lang een verkeerslicht groen is.
trafficLightToRedTimer	unsigned long	Timer.
timeTrafficLightIsYellow	const int	Hoe lang een verkeerslicht geel is.
trafficLightPreviousMillis	unsigned long	Timer.
TRAFFIC_LIGHT_RED	const int	Het verkeerslicht patroon voor rood.
TRAFFIC_LIGHT_TO_GREEN	const int	Het verkeerslicht patroon voor naar groen.
TRAFFIC_LIGHT_GREEN	const int	Het verkeerslicht patroon voor groen.
TRAFFIC_LIGHT_YELLOW	const int	Het verkeerslicht patroon voor geel.
TRAFFIC_LIGHT_PATTERNS [NUMBER_OF_TRAFFIC_LIGHT_PATTERNS] [NUMBER_OF_COLORS_PER_TRAFFIC_LIGHT]	const int	Een tweedimensionale array met hierin alle mogelijke stoplicht patronen.

Methoden

trafficLightSetup()	void	Voert voor alle stoplichten de led control setup functie uit.
turnTrafficLightOff(int trafficLight)	void	Zet alle drie de lampen van een verkeerslicht uit.
setTrafficLightPattern (int trafficLight, int pattern)	void	Deze functie loopt door de patronen array. Als hij een 0 tegen komt zet hij de lamp uit. Als hij een 1 tegenkomt zet hij deze lamp aan.
trafficLightsBlink(int light1, int light2)	void	Deze functie roept ledControlLedsBlink (int led1, int led2) aan.
getTrafficLightRed()	int	Retourneert patroon TRAFFIC_LIGHT_RED.
getTrafficLightToGreen()	int	Retourneert patroon TRAFFIC_LIGHT_TO_GREEN.
getTrafficLightGreen()	int	Retourneert patroon TRAFFIC_LIGHT_GREEN.
getTrafficLightToRed()	int	Retourneert patroon TRAFFIC_LIGHT_YELLOW.
getNumberOfTrafficLights()	int	Retourneert NUMBER_OF_TRAFFIC_LIGHTS.
get traffic light to green timer()	unsigned long	Retourneert trafficLightToGreenTimer.
get time to green()	int	Retourneert timeTrafficLightToGreen.
set_traffic_light_to_green_timer (unsigned long time)	void	Maakt trafficLightToGreenTimer gelijk aan time.
get traffic light is green timer()	unsigned long	Retourneert trafficLightIsGreenTimer.
set_traffic_light_is_green_timer (unsigned long time)	void	Maakt trafficLightIsGreenTimer gelijk aan time.
get time traffic light is green()	int	Retourneert timeTrafficLightIsGreen.
get traffic light to red timer()	int	Retourneert trafficLightToRedTimer.
set_traffic_light_to_red_timer (unsigned long time)	void	Maakt trafficLightToRedTimer gelijk aan time.
get time traffic light is yellow()	int	Retourneert timeTrafficLightIsYellow.

3. Toestanden

3.1 Toestandsdiagram



Figuur 3 State machine diagram (Figuur is ook te vinden in de map diagrammen en schema's)