

Automotive Software Concepts

Assignment #1

Indicator Lights
Patryk Stefanski 20089013
version4
31/03/2020

Design

The system has three inputs button Left, button Right and button hazard. These buttons are used to control the two outputs led left and led right. Pressing the left button flashes the left led and pressing the right button flashes the right led. Pressing the hazard button flashes both Leds. The hazards can be turned on while the left or right indicator is on, once the hazards are turned off it resumes the action before. The left and right indicators can't be turned on at the same time.

The left led represents the left indicator and right led represents the right indicator. The leds are being flashed every 500milliseconds(1hz) while the indicators or hazards are on otherwise, the leds are off.

Code

The buttons are detected by the detectEventLeft ,detectEventRight and detectEventHazards functions which return a number when the button is pressed down and a different number when the button is released.Each of these numbers is then assigned to an action. For example, the when the left button is pressed down it returns the number 0 which is assigned to downLeft.

```
int detectEventLeft() {  
    if (digitalRead(buttonPinLeft) == HIGH) {  
        return 0;  
    }  
    if (digitalRead(buttonPinLeft) == LOW) {  
        return 1;  
    }  
}
```

```
}
```

The code starts off by assigning a `currentState` of 0. The state can be changed if a specified action occurs while in a specified state. For example, if `currentState` is 0 and the action `downLeft` is detected the `currentState` changes to 1.

```
if ((currentState == 0) && (eventLeft == downLeft)) {  
    currentState = 1;  
}  
if ((currentState == 1) && (eventLeft == upLeft)) {  
    currentState = 2;  
}  
if ((currentState == 2) && (eventHazards == downHazards)) {  
    currentState = 3;  
}
```

The leds are flashed at a rate of 1hz using the `millis()` function. This `millis` function returns how long the program has been running in milliseconds. Using an `if` statement the current time in `millis` is subtracted from previous time which is initially set to 0 and if the difference is greater than 500 it turns the led on (sets `ledState` to 1) if it's off or it turns the led off (sets `ledState` to 0) if it's on and this keeps on happening every 500ms causing the led to flash at 1hz because it's inside the loop.

```
int ledState = LOW;  
unsigned long previousMillis = 0;  
const long interval = 500;  
if (currentMillis - previousMillis >= interval) {  
    // save the last time you blinked the LED  
    previousMillis = currentMillis;  
  
    // if the LED is off turn it on and vice-versa:  
    if (ledState == LOW) {  
        ledState = HIGH;  
    } else {  
        ledState = LOW;  
    }  
}
```

```

    }
}

```

The lights are flashed by checking the `currentState` then turning on the correct led on and flashing it. eg if `currentState` is 2 (ie right indicator is on) the program uses `digitalWrite()` to connect the correct led and then that led is assigned the `ledState` which causes it to change from low to high every 500ms because its inside the loop.

```

if ((currentState == 0) || (currentState == 1) || (currentState ==
7) || (currentState == 13)) {

```

```

    digitalWrite(ledPinLeft, LOW);

```

```

    digitalWrite(ledPinRight, LOW);

```

```

}

```

```

if ((currentState == 2) || (currentState == 3) || (currentState
== 6)) {

```

```

    digitalWrite(ledPinLeft, ledState);

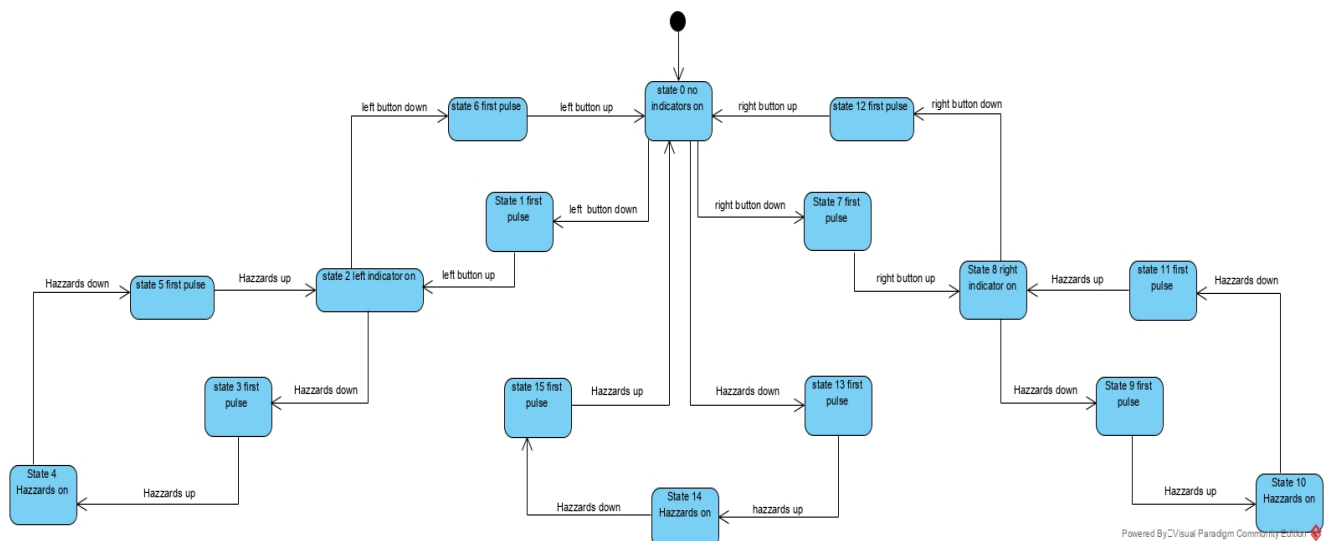
```

```

    digitalWrite(ledPinRight, LOW);

```

States



- State 0 no leds are on.
- State 1 leftDown from buttonLeft was received.
- State 2 leftUp from buttonLeft received and leftLed is turned on.
- State 3 hazardsDown from buttonHazards received and leftLed is still on.
- State 4 hazardsUp from buttonHazards received and leftLed and rightLed are turned on.
- State 5 hazardsDown from buttonHazards received and leftLed and rightLed are turned on.
- State 6 leftDown from buttonLeft received and leftLed is turned on
- State 7 rightDown from buttonRight received
- State 8 rightUp from buttonRight received and rightLed is turned on
- State 9 HazardsDown from buttonHazards received and rightLed is turned on.
- State 10 hazardsUp from buttonHazards received and rightLed and leftLed are turned on.
- State 11 hazardsDown from buttonHazards received and rightLed and leftLed are turned on.
- State 12 rightDown from buttonRight received and rightLed is turned on.
- State 13 HazardsDown from buttonHazards received.
- State 14 HazardsUp from buttonHazards received and rightLed and leftLed is turned on.
- State 15 HazardsDown from buttonHazards received and rightLed and leftLed is turned on.