ALCOHOL DETECTION WITH VEHICLE CONTROLLING SYSTEM

Team Members :(S1-T13)

1. 221CS116, Bhakti Raju Karchi , bhaktirajukarchi.221cs116@nitk.edu.in , 7483144229

2. 221CS130, Hitha N , hithan.221cs130@nitk.edu.in , 9380256163

3. 221CS138, P Devi Deepika , pdevideepika.221cs138@nitk.edu.in , 6360920665

Abstract:

The purpose of this project is to develop vehicle accident prevention by method of alcohol detector in an effort to reduce traffic accident cases based on driving under the influence of alcohol. This project is developed by integrating the alcohol sensor with the microcontroller 16F877A. The alcohol the sensor used in this project is MQ-2 which detects the alcohol content in human breath. An ignition system

which will produce spark plugs is built up as a prototype to act like the ignition starter over the vehicle's engine. The ignition system will operate based on the level of blood alcohol content (BAC) from human breaths detected by alcohol sensor. The main purpose is "Drunk driving detection". Nowadays, many accidents are happening because of the alcohol consumption of the driver or the person who is driving the vehicle. Thus, Drunken driving is a major reason for accidents in almost all countries all over the world. The Alcohol Detector in vehicle project is designed for the safety of the people seating inside the car.

In this simplified test bench code, where we are using test case-1 and teat case-2.Test case-1 simulate a scenario where no alcohol is detected, and it checks if the detection flag is 0 and detection counter is 0 as expected. Test case-2 simulates a scenario where alcohol is detected, and it check if the detection flag is 1 and detection counter is not 0.

Keywords: Arduino, L293D Motor Driver.

Brief Description:

Alcohol detection and vehical control systems typically involve hardware and software components to function effectively.

In hardware we need to use breathalyzer Sensor that is MQ3.This is key component that measures the alcohol content in a person's breath.It's based on principles of chemical or infrared analysis.often a, Microcontroller (e.g,Arduino,Raspberry Pi)is used to process signals from the breathalyzer sensor and make decision based on the alcohol concentration.ignition interlock device ,this hardware components is responsible for controlling the vehicale's ignition system based on the input from the microcontroller.If the alcohol level is above the legel limit ,it can prevent the vehical from starting.LED or Buzzer indicate the status of the alcohol level(e.g,above the threshold)

In Software Data Processing ,The microcontroller reads data from the breathalyzer sensor ,processes it ,and converts it into a usable alcohol concentration value. Decision Making Algorithm ,A software algorithm is implemented to analyze the alcohol

concentration.if it's above a specified threshold ,it triggers action to prevent the vehical from starting or halt its operation. Communication Protocols use for communication protocols to transmit data to other components or to external monitoring system.Integrating with Vehical Systems ,the software needs to integrate with the vehical's ignition system to control the starting or operation of the vehical based on the detected alcohol levels.

Code part involves ,Read and process data from the breathalyzer sensor.Implement the decision -making algorithm to determine if the alcohol conentration exceeds the limit.control the ignition syatem based on the outputs.Handle communication with other components or external systems if applicable. In flowchart intialize the system,read analog data from MQ3 sensor.Compare the read value with the predefined threshould .If alcohol level is above threshold,disable ignition .If alcohol level is below or at threshold ,enable ignition. Display status (e.g,"Ignition enabled" or "Ignition Disabled") ona

screen or communication externally. Continuously monitor and repeat the process.

The code you provided is a Verilog testbench (tb_alcohol_vehicle) for the Alcohol Detection and Vehicle Control system. In this testbench, you instantiate the AlcoholDetection and VehicleControl modules, declare signals to connect them, and apply test cases to verify the functionality of the system. Here's a breakdown of the key components of the testbench:

1.Module Instantiations->Alcohol Detection and Vehicle Control

2.Signal Declarations->sensor_data,ignition_request,

speed_request,alcohol_detected,ignition_enabled and speed_limit.

3.Stimulus Generation (initial block)

4.Monitor(always block)

In this simplified test bench code, where we are using test case-1 and teat case-2.Test case-1 simulate a scenario

where no alcohol is detected, and it checks if the detection flag is 0 and detection counter is 0 as expected. Test case-2 simulates a scenario where alcohol is detected, and it check if the detection flag is 1 and detection counter is not

Working:

Alcohol detection and vehicle control systems are designed to enhance road safety by preventing individuals under the influence of alcohol from operating vehicles. These systems typically incorporate breathalyzer technology to measure the driver's blood alcohol concentration (BAC). If a driver's BAC exceeds a certain threshold, the system may immobilize the vehicle or prevent it from starting, ensuring a safer driving environment. MQ-3 Alcohol Sensor: The MQ-3 sensor detects alcohol vapor concentration in the air and provides an analog output proportional to the alcohol level.Microcontroller (e.g., Arduino): This will interface with the MQ-3 sensor and process the analog

output.LED or Buzzer (Optional): To indicate the status of the alcohol level (e.g., above the threshold).

Here we have should use sequential logical gates (flip flops).there

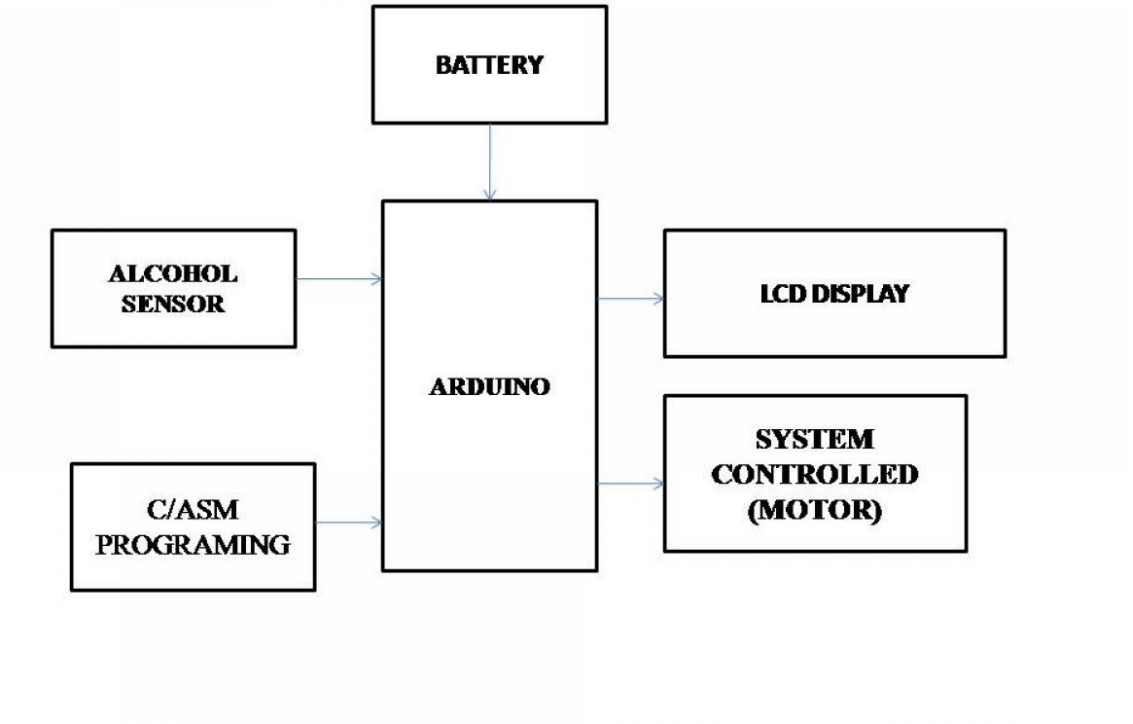
- Input –clock , alcohol level, rest.
- Output- Detection flag, detection counter.
- Alcohol level is 0 if alcohol is less than 127.
- Alcohol level is 1 if alcohol is greater than 127.

FUNCTIONAL TABLE:

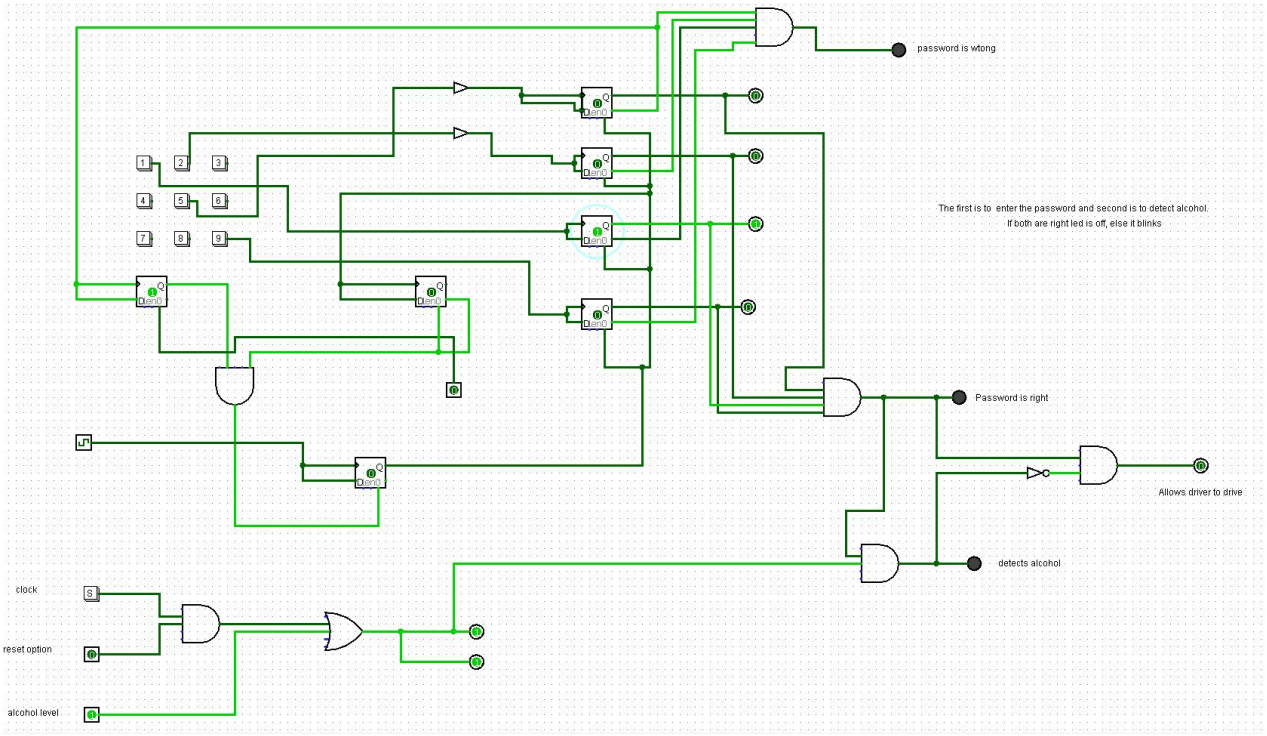
CLOCK	RESET	ALCOHOL LEVEL	DETECTION FLAG	DETECTION COUNTER
1	0	0-127	0	0
1	0	128-255	1	1
1	0	X	0	0

1	1	X	0	0
0	0	0-127	0	0
0	0	128-255	1	1
0	X	X	0	0
0	1	X	0	0

FLOW CHART:


BATTERY
ALCOHOL SENSOR
ARDUINO
LCD DISPLAY
SYSTEM CONTROLLED (MOTOR)
C/ASM PROGRAMING

Logisim daigram:



Verilog Code:

Testbench:

```
`timescale 1ns/1ps

module tb_alcohol_vehicle;


    // Instantiate the modules
     AlcoholDetection alcohol_detection_inst (
            .sensor_data(sensor_data),
            .alcohol_detected(alcohol_detected)
     );
  VehicleControl vehicle_control_inst (
            .alcohol_detected(alcohol_detected),
            .ignition_request(ignition_request),
            .speed_request(speed_request),
            .ignition_enabled(ignition_enabled),
            .speed_limit(speed_limit)
     );
 // Declare signals for the modules
     wire sensor_data;
     wire ignition_request;
     wire speed_request;
     wire alcohol_detected;
     wire ignition_enabled;
     wire speed_limit;
```

```
// Stimulus generation
    initial begin
        // Initialize inputs
        sensor_data = 0;
        ignition_request = 0;
        speed_request = 0;
  // Apply some test cases
  // Test case 1: No alcohol detecte
  // Set sensor_data accordingly
  // Assert expected values for ignition_enabled and speed_limit

  // Test case 2: Alcohol detected
  // Set sensor_data accordingly
  // Assert expected values for ignition_enabled and speed_limit

  // ... Add more test cases as needed

  // Finish simulation after some time
    $finish;
    End

    // Monitor to display outputs
    always @ (posedge alcohol_detected or posedge ignition_enabled or posedge speed_limit) begin
        $display("Alcohol Detected: %b, Ignition Enabled: %b, Speed Limit: %b",
                alcohol_detected, ignition_enabled, speed_limit);
    end

endmodule
```

Source file:

```
module AlcoholDetection (
    input wire sensor_data,
    output wire alcohol_detected
);
    // Alcohol detection logic based on sensor_data
    // Set alcohol_detected high if alcohol is detected
endmodule

module VehicleControl (
    input wire alcohol_detected,
    input wire ignition_request,
    input wire speed_request,
    output wire ignition_enabled,
    output wire speed_limit
);
    // Vehicle control logic based on alcohol_detected, ignition_request, and speed_request
    // Control ignition_enabled and set speed_limit accordingly
endmodule

module AlcoholDetectionVehicleControl (
    input wire sensor_data,
    input wire ignition_request,
    input wire speed_request,
    output wire alcohol_detected,
    output wire ignition_enabled,
```

```
        output wire speed_limit
);
        // Instantiate the AlcoholDetection module
        AlcoholDetection alcohol_detection_inst (
                .sensor_data(sensor_data),
                .alcohol_detected(alcohol_detected)
        );

        // Instantiate the VehicleControl module
        VehicleControl vehicle_control_inst (
                .alcohol_detected(alcohol_detected),
                .ignition_request(ignition_request),
                .speed_request(speed_request),
                .ignition_enabled(ignition_enabled),
                .speed_limit(speed_limit)
        );
Endmodule
```

REFERENCES:

● , International Journal of Control and Automation, Vol.9, No.2, 2016,

● https://youtu.be/f6uldsZ7SAI?si=cUVmN7X1tOGFGGBT

● https://youtu.be/p7JhPJG_JA0?si=MVZnvQYXWsx8x7Mx

● https://www.researchgate.net/figure/The-vehicle-safety-system-with-alcohol-detection-

● https://www.ijert.org/research/alcohol-detection-system-to-reduce-drunk-driving-IJERTCONV9IS03077.pdf

● https://www.researchgate.net/figure/The-vehicle-safety-system-with-alcohol-detection-circuit-diagram_fig2_327445845

● J.Dai ,J Teng , X.Bai , Z.Shen , and D.Xuan. ″ Mobile phone drunk driving detection ″ . In 2010 4thInternational Conference on Pervasive Computing Technologies for Healthcare,pp.1-8.IEEE,2010.

- ● .Babal , ″Accident avoiance and detection on highways″.

● https://www.tinkercad.com/--- HYPERLINK "https://www.tinkercad.com/---%3E(Tinkercad"> HYPERLINK "https://www.tinkercad.com/---%3E(Tinkercad"(Tinkercad app).

●https://www.slideshare.net/PankajSingh678/synopsis-for-alcohol-detection-with-vehicle-controlling-1