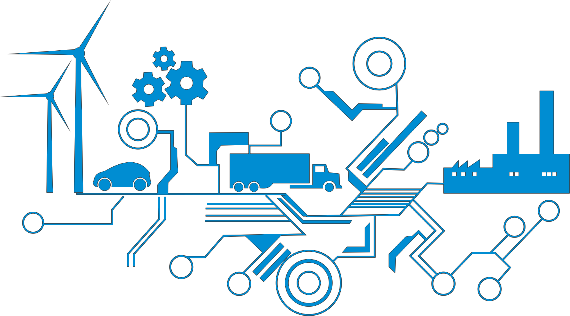
HVAC and Air Outlet Control



Group Title

Group Members:

|  |  |  |
| --- | --- | --- |
| Sl. No. | Name | Employee ID |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

© KPIT Technologies Ltd. All rights reserved. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form, or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior permission in writing of KPIT Technologies Ltd. Confidential & for internal circulation only.

Table of Contents

[HVAC Control 3](#_Toc511306054)

[Detailed Requirements 3](#_Toc511306055)

[Functional Diagram 3](#_Toc511306056)

[Comparison of Existing (Infineon, Renesas , Atmel etc ) 3](#_Toc511306057)

[Proposed Design 4](#_Toc511306058)

[Functional Requirements 4](#_Toc511306059)

[Systems and Subsystems 4](#_Toc511306060)

[Inputs and outputs of the Systems 6](#_Toc511306061)

[Test Cases 6](#_Toc511306062)

[UML Diagrams 6](#_Toc511306063)

[Model Implementation 7](#_Toc511306064)

[Code Generation of the Model 7](#_Toc511306065)

[Autocode Generation 7](#_Toc511306066)

[AUTOSAR complaint code generation 7](#_Toc511306067)

[Test case Validation 8](#_Toc511306068)

[Air Outlet Control 9](#_Toc511306069)

[Detailed Requirements 9](#_Toc511306070)

[Functional Diagram 9](#_Toc511306071)

[Comparison of Existing (Infineon, Renesas , Atmel etc ) 9](#_Toc511306072)

[Proposed Design 10](#_Toc511306073)

[Functional requirements 10](#_Toc511306074)

[Systems and Subsystems 10](#_Toc511306075)

[Inputs and outputs of the Systems 12](#_Toc511306076)

[Test Cases 12](#_Toc511306077)

[UML Diagrams 12](#_Toc511306078)

[Model Implementation 13](#_Toc511306079)

[Code Generation of the Model 13](#_Toc511306080)

[Autocode Generation 13](#_Toc511306081)

[AUTOSAR complaint code generation 13](#_Toc511306082)

[Test case Validation 13](#_Toc511306083)

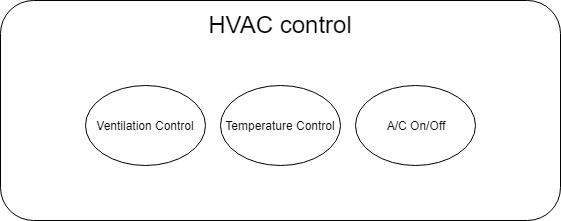
# HVAC Control

HVAC Control enables users to control in-cabin Heating, Ventilation and Air Conditioning. It has two knobs, Ventilation Knob and Temperature Knob. It also have a switch to turn on and off the air conditioning if the desired temperature is cooling.

## Detailed Requirements

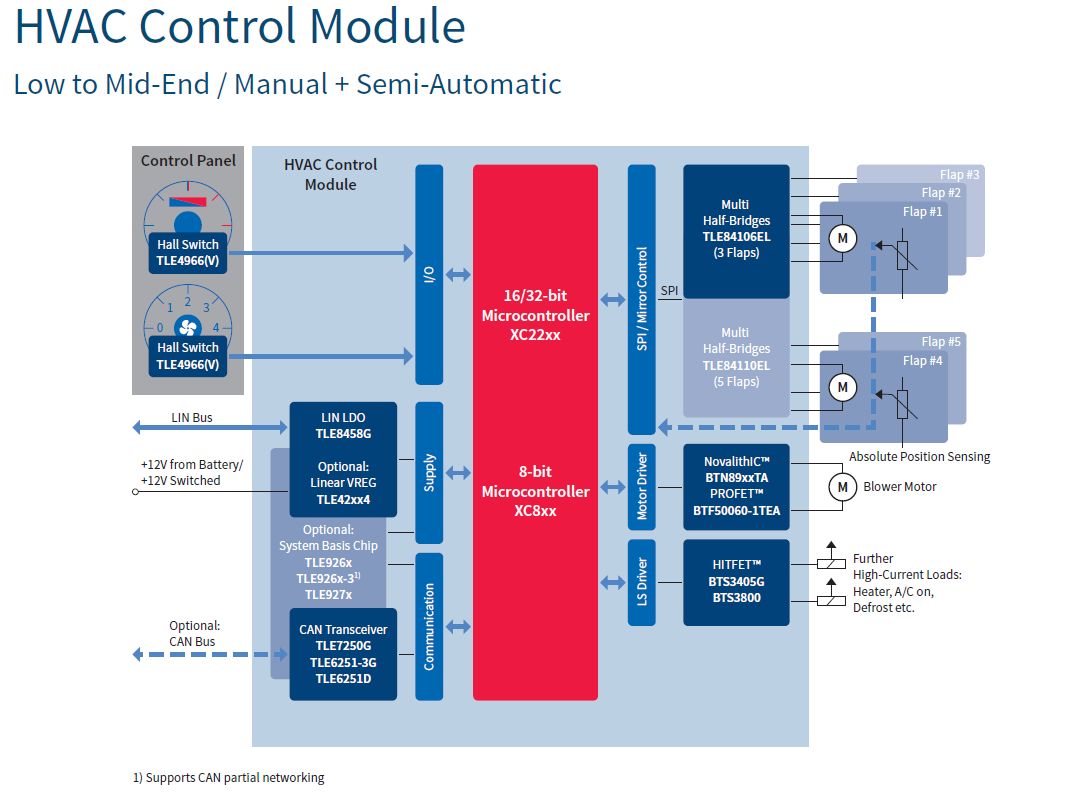
|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| HVAC000 | Ventilation Knob controls the rotation speed on fans. |
| HVAC100 | Temperature Knob sets the desired temperature. |
| HVAC200 | A/C Switch turns on and off the A/C if desired temperature is set to cooling. |

## Functional Diagram

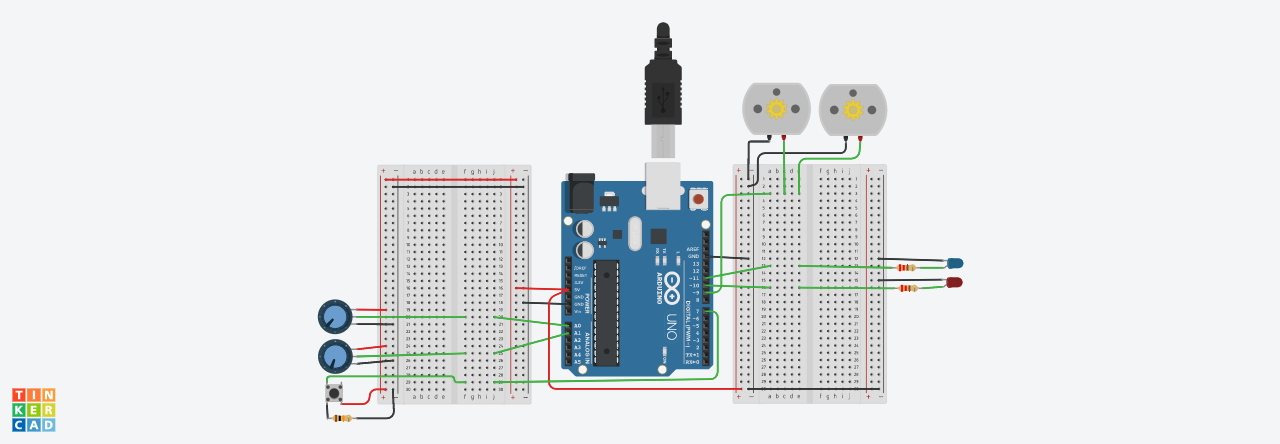


### Comparison of Existing (Infineon, Renesas , Atmel etc )

Infineon



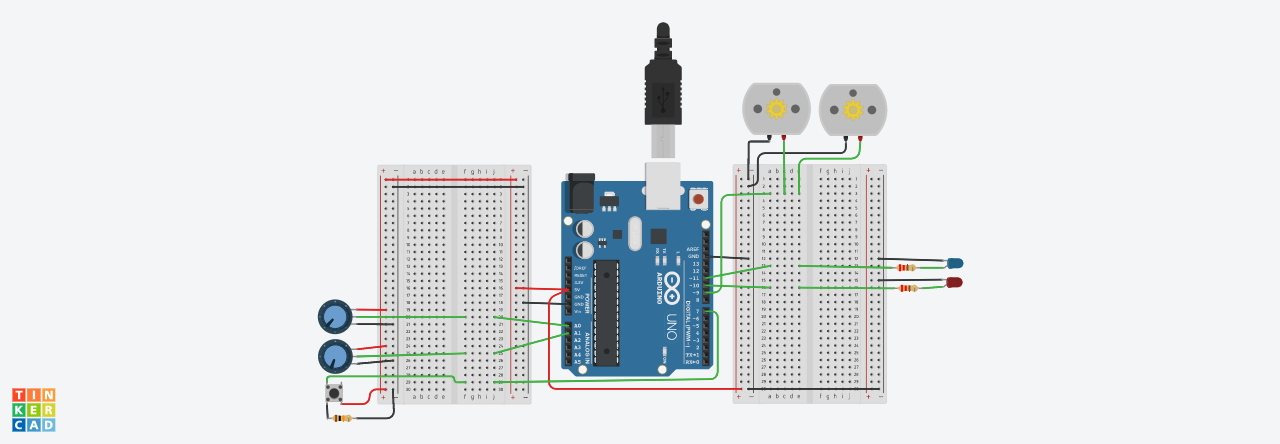
### Proposed Design



## Functional Requirements

|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| HVAC001 | Ventilation Knob sends analog signal to the MCU to determine the speed of ventilation fans. |
| HVAC002 | MCU controls the speed of ventilation fans depends on the input of Ventilation Knob. |
| HVAC101 | Temperature Knob sends analog signal to the MCU to determine the desired temperature. |
| HVAC102 | MCU controls the power of heating and air conditioning depends on the input of Temperature Knob. |
| HVAC301 | A/C Switch sends digital signal to MCU to indicate the power on/off of the A/C. |
| HVAC302 | MCU turns on or off the air conditioning depends on the input of A/C Switch. |

### Systems and Subsystems



|  |  |  |
| --- | --- | --- |
| **Name** | **Quantity** | **Component** |
| U1 | 1 | Arduino Uno R3 |
| M1, M2 | 2 | DC Motor |
| Rpot4, Rpot1 | 2 | 250 kOhm, Potentiometer |
| R1, R2 | 2 | 220 ohm Resistor |
| D1 | 1 | Red LED |
| D2 | 1 | Blue LED |
| S1 | 1 | Pushbutton |
| R3 | 1 | 10 kohm Resistor |

/\*

This program simulate HAVC Control.

Input: Ventilation Knob, Temperature Knob, A/C switch

Output: Fans and LEDs simulation of HVAC

- Copyright 2018 YuPui

\*/

#include<stdint.h>

#include<avr/io.h>

#include<avr/interrupt.h>

#include<util/delay.h>

#define sbi(sfr, bit) (\_SFR\_BYTE(sfr) |= \_BV(bit))

#define SET\_BIT(PORT, BIT) PORT|= (1 << BIT)

#define CLR\_BIT(PORT, BIT) PORT&= ~(1 << BIT)

# define F\_CPU 16000000UL

volatile uint16\_t ventIn = 0;

volatile uint16\_t tempIn = 0;

void setup() {

// Set up port C pin 0 as analog input

// Ventilation Knob

DDRC = (0 << 0);

PORTC = (0 << 0);

// Set up port C pin 0 as analog input

// Temperature Knob

DDRC = (0 << 1);

PORTC = (0 << 1);

// Analog input initializtion

ADMUX = 0x00;

ADMUX |=\_BV(REFS0)|\_BV(MUX3)|\_BV(MUX2)|\_BV(MUX1)|\_BV(MUX0);

ADCSRA = 0x00;

ADCSRB = 0x00;

ADCSRA |=\_BV(ADEN) | \_BV(ADPS2) |\_BV(ADPS1) | \_BV(ADPS0);

ADCSRA |= \_BV(ADSC);

// Set port D pin 7 as digital input

// A/C on/off switch

CLR\_BIT(PORTD, PD7);

// Set up pin 9, 10, 11 as output

SET\_BIT(DDRB, PB3);

SET\_BIT(DDRB, PB2);

SET\_BIT(DDRB, PB1);

CLR\_BIT(PORTB, PB3);

CLR\_BIT(PORTB, PB2);

CLR\_BIT(PORTB, PB1);

// Set up timer 1 channel A for PWM output on pin 9

sbi(TCCR1A, COM1A1);

// Set up timer 1 channel B for PWM output on pin 10

sbi(TCCR1A, COM1B1);

// Set up timer 2 channel A for PWM output on pin 11

sbi(TCCR2A, COM2A1);

}

void loop() {

// Read the analog in value:

// Start reading value from port C pin 0

ADMUX &= 0xf0;

ADMUX |= 0X0; // port C pin 0

ADCSRA |= \_BV(ADSC);

\_delay\_ms(10);

ventIn = ADC;

// Start reading value from port C pin 0

ADMUX &= 0xf0;

ADMUX |= 0X1; // port C pin 1

ADCSRA |= \_BV(ADSC);

\_delay\_ms(10);

tempIn = ADC;

// analog output to pin 9 drive the fans

OCR1A = ventIn/4;

// determine heating or cooling

if (tempIn > 600) {

// Heating

// analog output to pin 10 and 11

OCR1B = (tempIn - 600)/2;

OCR2A = 0;

}

if (tempIn <= 600 && tempIn > 400) {

// Off

OCR1B = 0;

OCR2A = 0;

}

if (tempIn < 400) {

// Cooling, A/C on depend on switch input

OCR1B = 0;

OCR2A = (200 - tempIn/2)\*(PIND >> 7);

}

}

int main() {

setup();

while (1) { // super loop

loop();

}

}

### Inputs and outputs of the Systems

Inputs:

Ventilation Knob

Temperature Knob

A/C Switch

Outputs:

2X Ventilation Fans

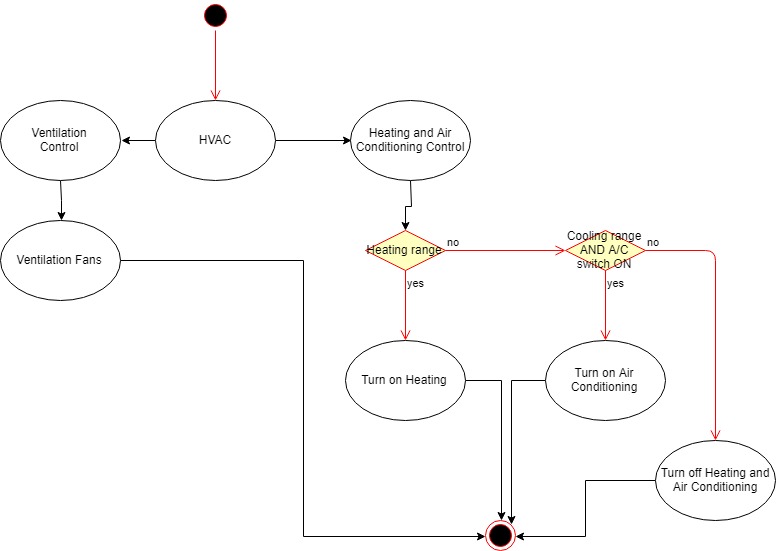
Heating Elements (Simulated by a Red LED)

Cooling Elements (Simulated by a Blue LED)

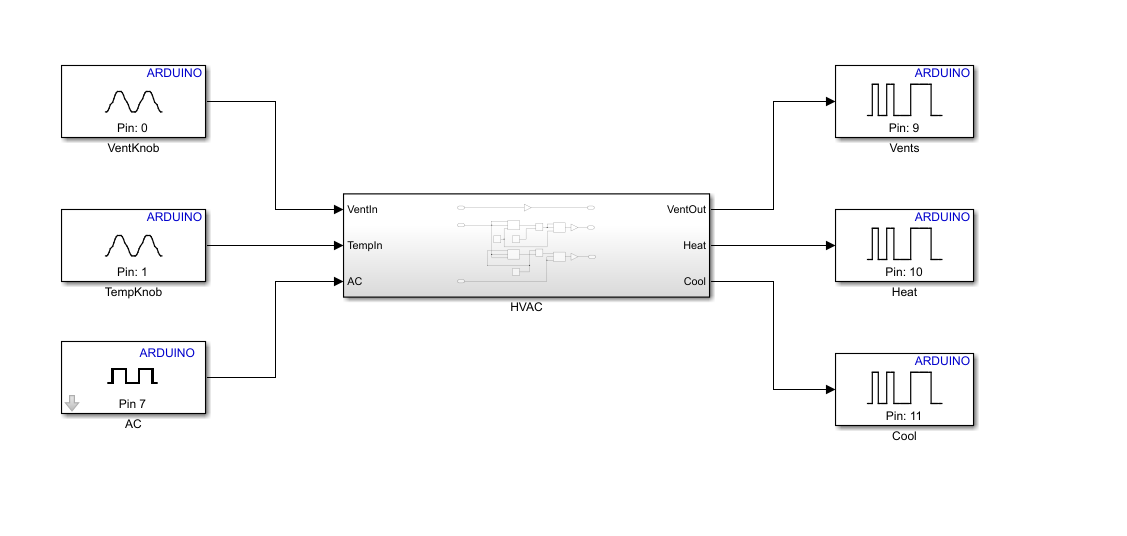
## Test Cases

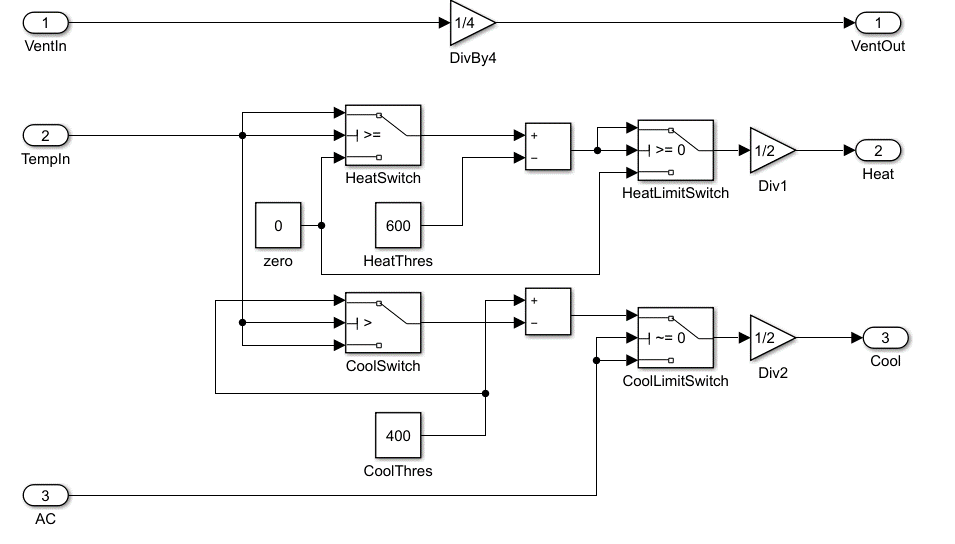
|  |  |  |
| --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Result** |
| HVACT001 | Turn Ventilation Knob to low | Ventilation fans set to low |
| HVACT002 | Turn Ventilation Knob to high | Ventilation fans set to high |
| HVACT003 | Turn Temperature Knob to high and A/C Switch off | Red led glows, blue led off |
| HVACT004 | Turn Temperature Knob to high and A/C Switch on | Red led glows, blue led off |
| HVACT005 | Turn Temperature Knob to low and A/C Switch off | Red led off, blue led off |
| HVACT006 | Turn Temperature Knob to low and A/C Switch on | Red led off, blue led glows |

## UML Diagrams



## Model Implementation





## Code Generation of the Model

Describe ..

### Autocode Generation

Describe ..

### AUTOSAR complaint code generation

Describe ..

## Test case Validation

Tests was perform in hardware with Embedded C code after they pass in the simulation on TinkerCAD and Simulink.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Result** | **Test Result** |
| HVACT001 | Turn Ventilation Knob to low | Ventilation fans set to low | Ventilation fans set to low |
| HVACT002 | Turn Ventilation Knob to high | Ventilation fans set to high | Ventilation fans set to high |
| HVACT003 | Turn Temperature Knob to high and A/C Switch off | Red led glows, blue led off | Red led glows, blue led off |
| HVACT004 | Turn Temperature Knob to high and A/C Switch on | Red led glows, blue led off | Red led glows, blue led off |
| HVACT005 | Turn Temperature Knob to low and A/C Switch off | Red led off, blue led off | Red led off, blue led off |
| HVACT006 | Turn Temperature Knob to low and A/C Switch on | Red led off, blue led glows | Red led off, blue led glows |

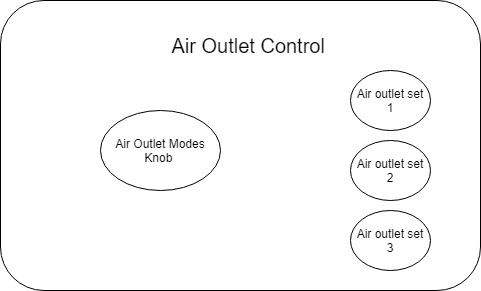
# Air Outlet Control

Air Outlet Control enables user to control air outlet for ventilation fans. It has one knob that allow user to switch 5 modes. Mode 1 turns on outlet set 1 only. Mode 2 turns on outlet set 2 only. Mode 3 turns on outlet set 1 and 2. Mode 4 turns on outlet set 1 and 3. Mode 5 turns on outlet set 3 only.

## Detailed Requirements

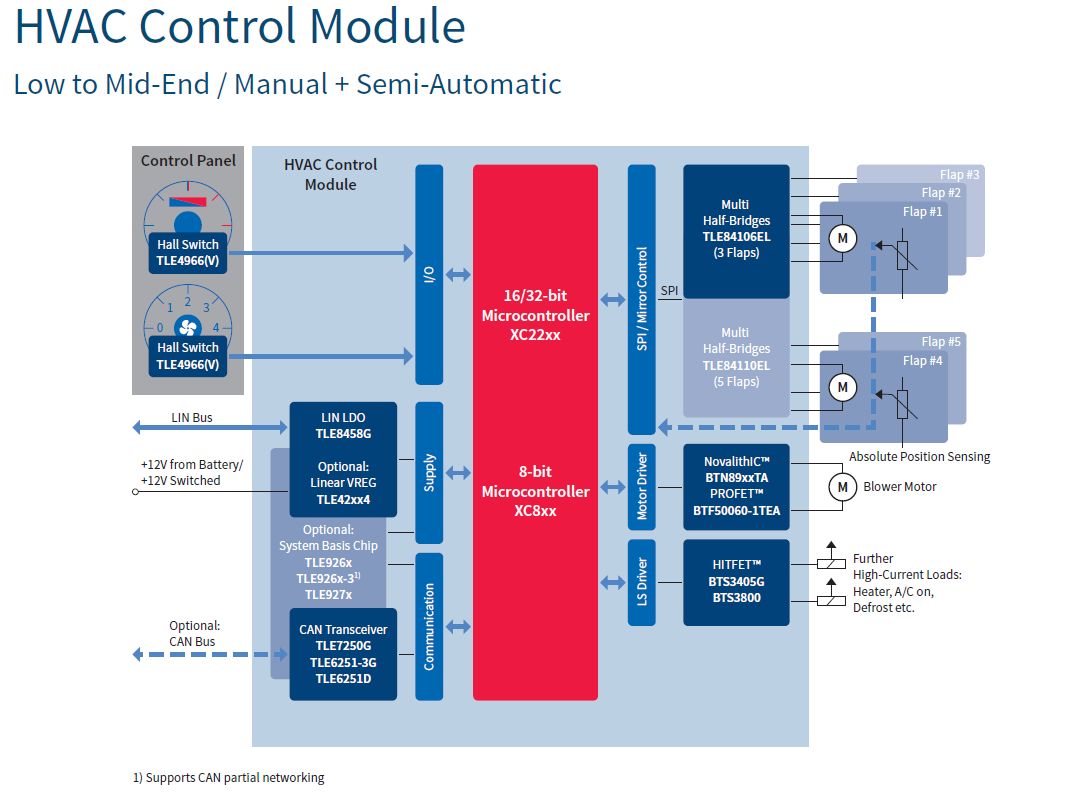
|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| AOC000 | Air Outlet Modes Knob controls air outlet modes. |
| AOC100 | At mode 1, air outlet set 1 only shall be on. |
| AOC200 | At mode 2, air outlet set 2 only shall be on. |
| AOC300 | At mode 3, air outlet set 1 and 2 shall be on. |
| AOC400 | At mode 4, air outlet set 1 and 3 shall be on. |
| AOC500 | At mode 5, air outlet set 3 only shall be on. |

## Functional Diagram

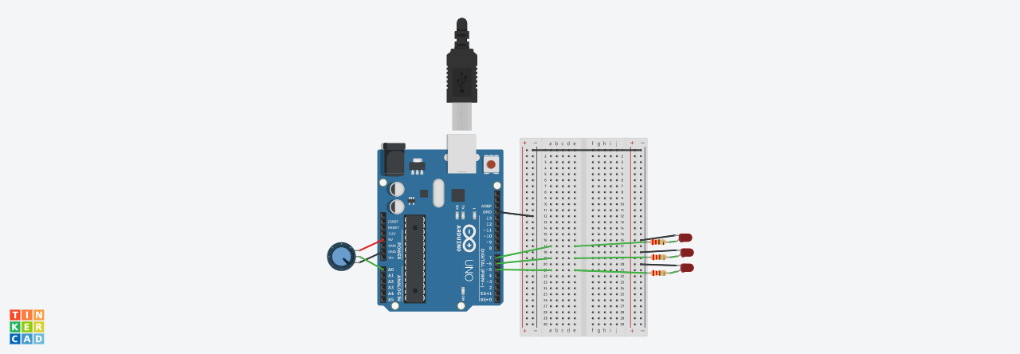


### Comparison of Existing (Infineon, Renesas , Atmel etc )

Infineon



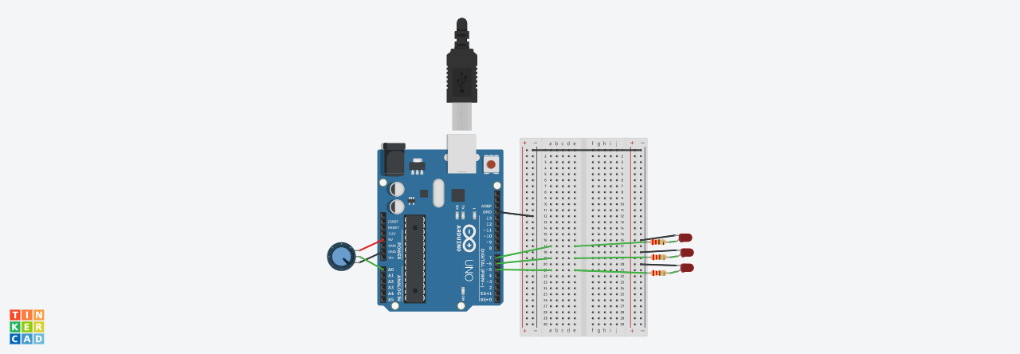
### Proposed Design



## Functional requirements

|  |  |
| --- | --- |
| **Requirement ID** | **Requirement Description** |
| AOC001 | Air Outlet Modes Knob set the air outlet mode value, range from 0 to 1023. |
| AOC101 | When mode value < 200, air outlet set 1 on, rest is off. |
| AOC201 | When 200 <= mode value < 400, air outlet set 2 on, rest is off. |
| AOC301 | When 400 <= mode value < 600, air outlet set 1 and 2, rest is off. |
| AOC401 | When 600 <= mode value < 800, air outlet set 1 and 3, rest is off. |
| AOC501 | When mode value >= 800, air outlet set 3 on, rest is off. |

### Systems and Subsystems



|  |  |  |
| --- | --- | --- |
| **Name** | **Quantity** | **Component** |
| U1 | 1 | Arduino Uno R3 |
| RpotAir Outlet Modes Knob1 | 1 | 250 kOhm, Potentiometer |
| D2, D3, D4 | 3 | Red LED |
| R1, R2, R3 | 3 | 220 ohm Resistor |

/\*

This program simulate Air Outlet Control.

Input: Air Outlet Modes Knob

Output: LEDs simulation of air outlets

- Copyright 2018 YuPui

\*/

#include<stdint.h>

#include<avr/io.h>

#include<util/delay.h>

#define SET\_BIT(PORT, BIT) PORT|= (1 << BIT)

#define CLR\_BIT(PORT, BIT) PORT&= ~(1 << BIT)

# define F\_CPU 16000000UL

int main() {

// Set up port C pin 0 as analog input

DDRC = (0 << 0);

PORTC = (0 << 0);

// Analog input initializtion

ADMUX = 0x00;

ADMUX |=\_BV(REFS0)|\_BV(MUX3)|\_BV(MUX2)|\_BV(MUX1)|\_BV(MUX0);

ADCSRA = 0x00;

ADCSRB = 0x00;

ADCSRA |=\_BV(ADEN) | \_BV(ADPS2) |\_BV(ADPS1) | \_BV(ADPS0);

ADCSRA |= \_BV(ADSC);

// Set up port D pin 5,6,and 7 as digital output

SET\_BIT(DDRD, PD5);

SET\_BIT(DDRD, PD6);

SET\_BIT(DDRD, PD7);

CLR\_BIT(PORTD, PD5);

CLR\_BIT(PORTD, PD6);

CLR\_BIT(PORTD, PD7);

while (1) { // super loop

// Start reading value from port C pin 0

ADMUX &= 0xf0;

ADMUX |= 0X0; // port C pin 0

ADCSRA |= \_BV(ADSC);

if (ADC < 200) {

SET\_BIT(PORTD, PD5);

CLR\_BIT(PORTD, PD6);

CLR\_BIT(PORTD, PD7);

}

if (ADC < 400 && ADC >= 200) {

CLR\_BIT(PORTD, PD5);

SET\_BIT(PORTD, PD6);

CLR\_BIT(PORTD, PD7);

}

if (ADC < 600 && ADC >= 400) {

SET\_BIT(PORTD, PD5);

SET\_BIT(PORTD, PD6);

CLR\_BIT(PORTD, PD7);

}

if (ADC < 800 && ADC >= 600) {

SET\_BIT(PORTD, PD5);

CLR\_BIT(PORTD, PD6);

SET\_BIT(PORTD, PD7);

}

if (ADC > 800) {

CLR\_BIT(PORTD, PD5);

CLR\_BIT(PORTD, PD6);

SET\_BIT(PORTD, PD7);

}

\_delay\_ms(100);

}

}

### Inputs and outputs of the Systems

Inputs:

Air Outlet Mode Knob

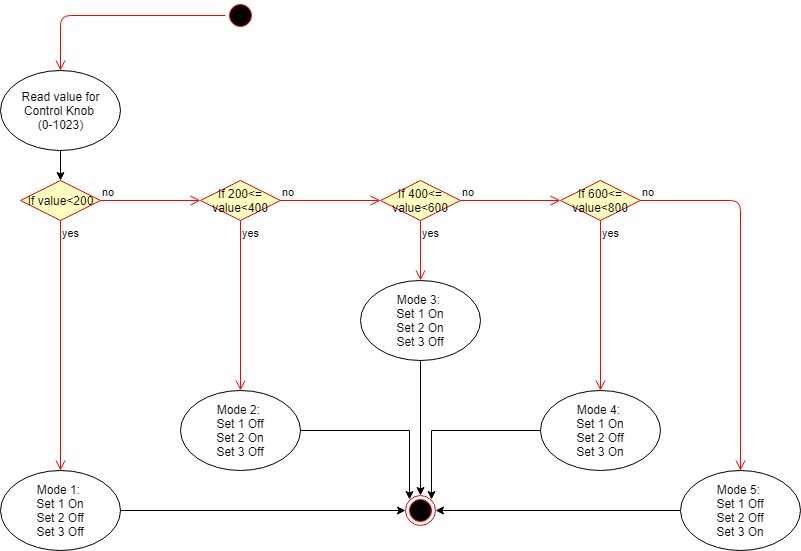
Outputs:

2X RED LED simulating the switch of air outlet sets

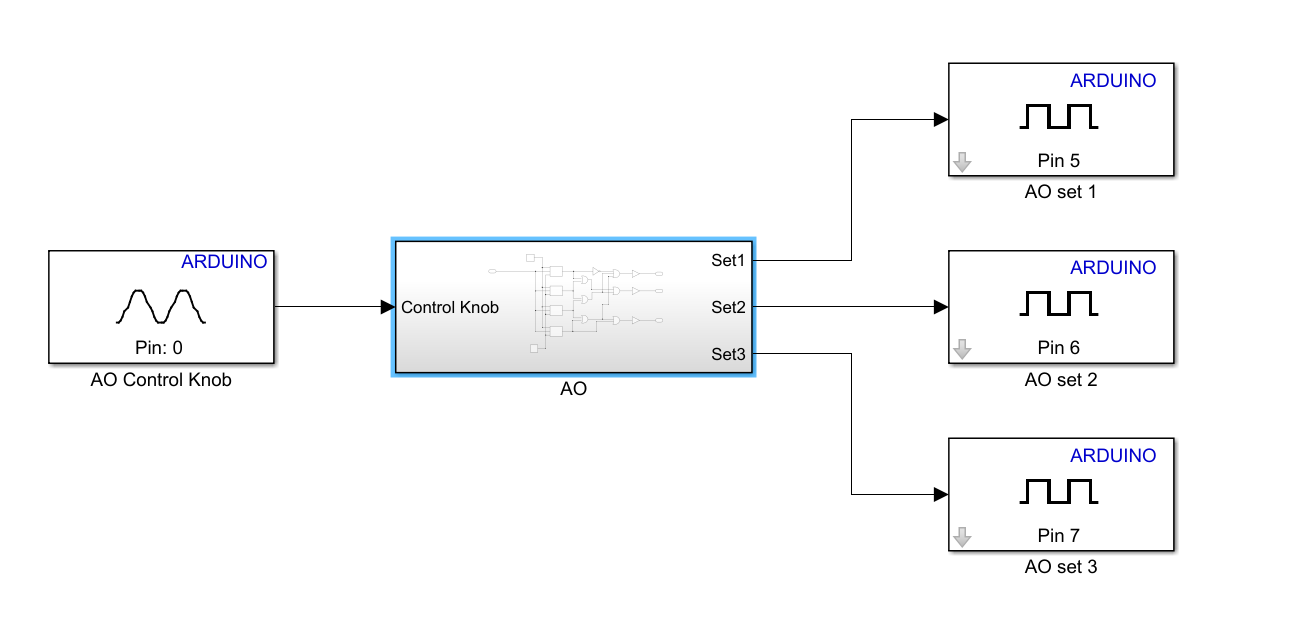
## Test Cases

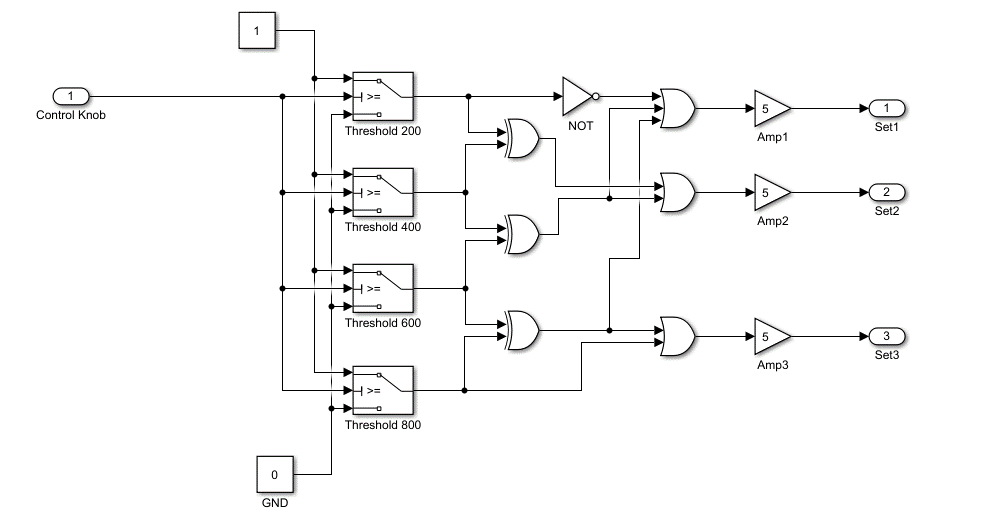
|  |  |  |
| --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Result** |
| AOCT001 | Turn Air Outlet Mode Knob to very low (below 200) | Mode 1: set 1 on. |
| AOCT002 | Turn Air Outlet Mode Knob to low (between 200 to 400) | Mode 2: set 2 on. |
| AOCT003 | Turn Air Outlet Mode Knob to middle (between 400 to 600) | Mode 3: set 1 and 2 on. |
| AOCT004 | Turn Air Outlet Mode Knob to high (between 600 to 800) | Mode 4: set 1 and 3 on. |
| AOCT005 | Turn Air Outlet Mode Knob to very high (above 800) | Mode 5: set 3 on. |
| AOCT006 | Turn Air Outlet Mode Knob back to high (between 600 to 800) | Mode 4: set 1 and 3 on. |

## UML Diagrams



## Model Implementation





## Code Generation of the Model

Describe ..

### Autocode Generation

Describe ..

### AUTOSAR complaint code generation

Describe ..

## Test case Validation

Tests was perform in hardware with Embedded C code after they pass in the simulation on TinkerCAD and Simulink.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Result** | **Test Result** |
| AOCT001 | Turn Air Outlet Mode Knob to very low (below 200) | Mode 1: set 1 on. | Mode 1: set 1 on. |
| AOCT002 | Turn Air Outlet Mode Knob to low (between 200 to 400) | Mode 2: set 2 on. | Mode 2: set 2 on. |
| AOCT003 | Turn Air Outlet Mode Knob to middle (between 400 to 600) | Mode 3: set 1 and 2 on. | Mode 3: set 1 and 2 on. |
| AOCT004 | Turn Air Outlet Mode Knob to high (between 600 to 800) | Mode 4: set 1 and 3 on. | Mode 4: set 1 and 3 on. |
| AOCT005 | Turn Air Outlet Mode Knob to very high (above 800) | Mode 5: set 3 on. | Mode 5: set 3 on. |
| AOCT006 | Turn Air Outlet Mode Knob back to high (between 600 to 800) | Mode 4: set 1 and 3 on. | Mode 4: set 1 and 3 on. |