|  |  |
| --- | --- |
| Electronic Throttle  control (ETC) Name: Rutvij Joshi ID: 2005502 |  |
|  |  |

Contents

[Name: Rutvij Joshi ID: 2005502 1](file:///C:\Users\rutvi\Desktop\STUDY\KPIT\Embedded%20C\2005502_EmbeddedMilestone_1\2005502_Report.docx#_Toc68536261)

[Introduction to Problem 3](#_Toc68536262)

[Requirement Specifications 3](#_Toc68536263)

[Functional Modelling 4](#_Toc68536264)

[System Description 4](#_Toc68536265)

[ETC System Fail Safe Mode 4](#_Toc68536266)

[Software Design using UML 5](#_Toc68536267)

[Static Analysis 8](#_Toc68536268)

[CPPCheck Results 8](#_Toc68536269)

[Memory Configuration 8](#_Toc68536270)

List of Tables

[Table 1: Requirement specifications. 3](#_Toc68435964)

[Table 2: Fail safe modes. 4](#_Toc68435965)

[Table 3: CPPCheck Results. 8](#_Toc68435966)

[Table 4: Memory configuration 8](#_Toc68435967)

List of Figures

[Figure 1: Block diagram. 4](#_Toc68435968)

[Figure 2: Component relationship diagram. 5](file:///C:\Users\rutvi\Desktop\STUDY\KPIT\Embedded%20C\Rutvij_2005502_Embedded_milestone\2005502_Report.docx#_Toc68435969)

[Figure 3: Flowchart. 6](#_Toc68435970)

[Figure 4: File diagram. 7](#_Toc68435971)

## Introduction to Problem

Electronic throttle control (ETC) is an automobile technology which electronically connects the accelerator pedal to the throttle, replacing a mechanical linkage. It consists of an accelerator pedal sensor (APS), engine control module (ECM) and throttle servo. The aim of ETC is to make powertrain drive by wire to support systems like cruise control, stability control, traction control and safety system. ETC provides some benefit in areas such as air-fuel ratio control, exhaust emissions and fuel consumption reduction, and works in concert with other technologies such as gasoline direct injection. An ETC also has redundant pedal sensor which is connected to same pedal, and it also deals with various faults like throttle motor failure, pedal sensor failure.

This system works with three fail-safe modes:

1. **APS1 mode**: - In this mode data from pedal position sensor (APS1) is taken. It is activated when there is failure in second pedal position sensor (APS2) or no failure.
2. **APS2 mode**: - In this mode data from pedal position sensor 2 (APS2) is taken. It is activated when

there is failure in second pedal position sensor (APS2).

1. **Limp mode**: - In this mode engine runs at idle speed. This mode is activated when there is failure of both pedal position sensor (APS1 and APS2) or there is failure in throttle position sensor (TPS).

In actual system sensors communicate fault code with ECM (Engine control module) as they use communication protocols like CAN, FlexRay etc. Here switches are used.

## Requirement Specifications

|  |  |  |
| --- | --- | --- |
| Requirement ID | Requirement | Requirement Description |
| 1 | Power supply | 5V for Atmega328p and APS (Accelerator pedal position sensors). |
| 2 | Throttle servo | 60Hz PWM waveform for throttle servo and values of APS mapped to 9%-90% duty cycle range. |
| 3 | Pedal position sensor | Linear potentiometer giving values between  0-5 V. |
| 4 | Fault management | Algorithm with proper fault management and two pedal position sensors (APS1 and APS2). |
| 5 | Visual indicator of failure | LED is used as MIL (Malfunction indicator lamp) to notify driver about failure in ETC system. |

Table : Requirement specifications.

## Functional Modelling

### System Description

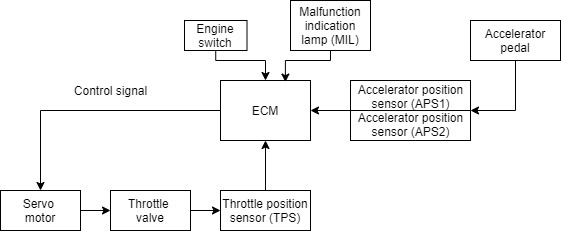


Figure : Block diagram.

This system uses an engine control module (ECM) to take input from pedal position sensor (APS) to control throttle servo. It also takes care of fail-safe modes. An ECM is brain of system, we have used Atmega328 as ECM since, it has clock speed of 16MHz, and it has PWM and ADC channels.

Potentiometer is used as pedal position sensor as it provides linear response in terms of voltage.

For failure management two such sensors are used. These sensors are connected to ADC0 and ADC1 channels of Atmega328.

The electronic throttle body (ETB) is a main part of an electronic throttle control (ETC) system. The ETB controls the volume of air flowing into the engine. It features a butterfly valve which opens and closes according to a signal from the engine control unit. Throttle is controlled by servo which uses 60Hz PWM signal. By changing the duty cycle of PWM from 9%-90% we can control throttle position from fully open to fully closed. Servo motor uses a potentiometer (throttle position sensor) as feedback to get into correct position. We can observe PWM signal on oscilloscope. Malfunction indication lamp (MIL) is used to notify driver about failure in the system.

### ETC System Fail Safe Mode

|  |  |
| --- | --- |
| Failure | Fail-safe |
| APS1 failure | Replace with APS2 |
| APS2 failure | Replace with APS1 |
| Both APS fails | Engine idle (Limp mode) |
| Throttle sensor failure | Engine idle (Limp mode) |

Table : Fail safe modes.

To simulate fail-safe modes switches are used to indicate failure.

## Software Design using UML

#### Component relationship diagram

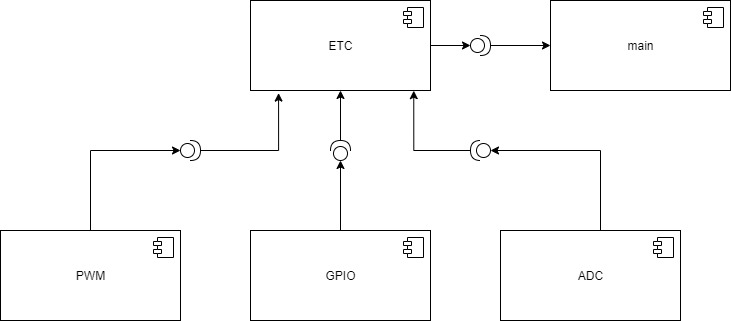


Figure 2: Component relationship diagram.

Above diagram shows relationship between various components. Driver level components are PWM, GPIO and ADC. An ETC component acts as a middle layer which implements ETC functionality. A main component implements logic of ETC as shown in below flowchart.

#### flowchart

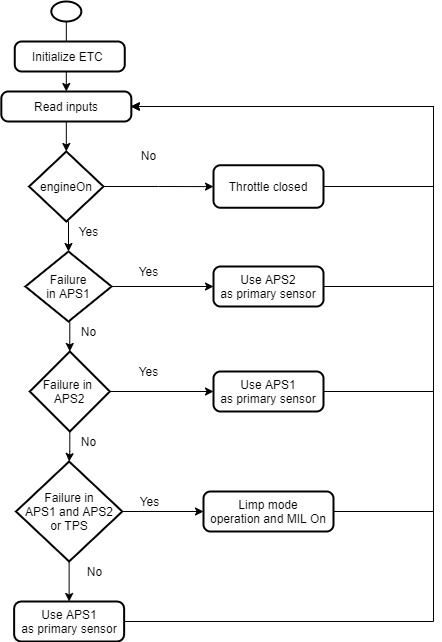


Figure : Flowchart.

As shown in above flowchart ECM monitors status of APS (Accelerator pedal position sensors) and decides the mode of operation. When no fault is detected APS1 is used as primary sensor. When both APS or TPS (Throttle position sensor) fails ECM uses limp mode. In the limp mode engine runs at an idle speed and driver is notified about failure by MIL (Malfunction indication lamp).

#### Algorithm

1. Start
2. Initialize ETC module
3. Read inputs from switches.
4. If engine on then
5. If failure in pedal position sensor1 then
6. Take data from pedal position sensor 2.
7. If failure in pedal position sensor2 then
8. Take data from pedal position sensor 1.
9. If failure in both pedal position sensor or throttle sensor then
10. Use limp mode operation.
11. No failure
12. Take data from pedal position sensor 1.
13. Else
14. Turn off the ETC module.
15. Stop

#### File diagram

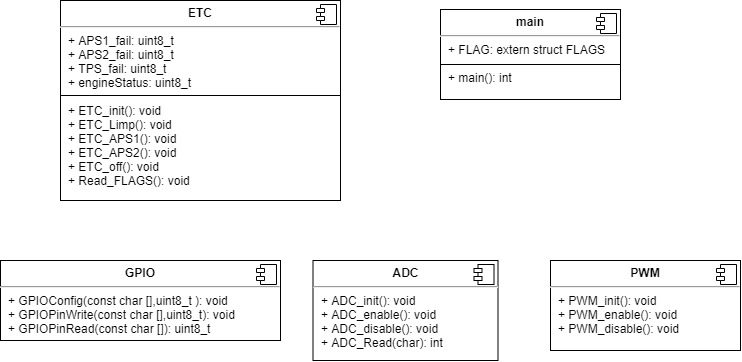


Figure : File diagram.

Above is file diagram of a project. The GPIO file consists of functions which enable application to configure and access I/O pins for reading or setting the level on pin. The pin name is passed as string argument e.g., GPIOConfig(“PB4”, OUTPUT). The ADC.c and ADC.h enables application to enable, disable or read the ADC value from selected channel. It uses ADC pre-scalar value of 2. The PWM.c and PWM.h has functions to initialize the PWM module with 60Hz frequency. It provides functions to enable and disable the PWM module. The ETC.c and ETC.h implements ETC operational modes. The ETC\_APS1() and ETC\_APS2() are used when one of the sensors fails. The ETC\_Limp() mode is used when either throttle sensor or both pedal sensor (APS1 and APS2) fails. The Read\_FLAGS() is used to monitor all the faults and engine switch status. The main.c implements ETC logic.

## Static Analysis

### CPPCheck Results

#### Run Test 1:

|  |  |
| --- | --- |
| Warnings | Relevant File |
| The function 'PWM\_disable' is never used. | PWM.c |
| The function 'ADC\_disable' is never used. | ADC.c |

Table : CPPCheck Results.

#### RUN TEST 2:

No warnings, as above functions are used to reduce power consumption by disabling the PWM and ADC module whenever not used.

## Memory Configuration

|  |  |
| --- | --- |
| Application Test | Size in bytes |
| 1 | .text segment  6074 bytes  .data segment  0 bytes  .bss  4 bytes |
| 2 | .text segment  6066 bytes  .data segment  0 bytes  .bss  4 bytes |

Table : Memory configuration