AIRBAG SYSTEM IMPLEMENTATION WITH ATMEGA328P

1. Introduction

This report describes the design and implementation of a simplified airbag deployment demonstration system using an ATmega328P microcontroller on Proteus. The system is inspired by the principles of real SRS airbags found in automotive applications. The primary goal is to simulate the key functionalities of frontal (stage 1 and stage 2) and side airbag deployments based on collision sensors and safety lock mechanisms.

2. System Overview

2.1 Hardware Components

1. **Power Supply**
   - 9 V battery regulated down to 5 V using a 7805 regulator.
   - Two 220 pF capacitors for smoothing and noise filtering.
   - Output: stable 5 V supply to the ATmega328P.
2. **Microcontroller: ATmega328P**
   - 16 MHz external crystal oscillator for clock.
   - AVCC pin attached to an LC low-pass filter to reduce noise during ADC operations.
   - All IO pins configured for various sensor inputs and LED outputs.
3. **Sensors and Inputs**
   - **Potentiometer** used to emulate a G sensor (i.e., measuring acceleration or collision severity).
   - **Push button** for side impact simulation (side airbag deployment).
   - **Push button** for system reset.
   - **Logic state input** for Safety Sensor (enables or disables overall airbag deployment).
4. **Indicators (LEDs)**
   - **Startup LED** (indicates system power-on).
   - **Stage 1 Deployment LED** (partial deployment).
   - **Stage 2 Deployment LED** (full frontal deployment).
   - **Side Airbag LED** (emulates the side airbag module firing).

2.2 Functional Behavior

1. **System Startup and Self-Test**
   Upon powering on, the system lights the "startup" LED on PORTD pin 0 for a brief period (6 seconds). This is akin to the real-world airbag ECU (Electronic Control Unit) self-test, where the airbag warning light illuminates to confirm that the system diagnostics are working. After the delay, the LED is turned off, indicating that the system is ready.

2. **Collision Sensing and Safety Logic**

   - The G sensor reading is taken from the ADC channel connected to the potentiometer.
   - A digital safety sensor on PC4 must be set for deployment to be allowed. If the safety sensor is off (bit not set), the system prevents any airbag deployment.
   - A side impact sensor is read on PC5. If triggered (push button pressed/activated), the system sets the side airbag LED.
3. **Frontal Airbag Deployment Stages**

   - Stage 1 deployment activates if the G sensor reading crosses a threshold of 512.
   - Stage 2 is triggered when the ADC reading is higher than 728, simulating a more severe collision..
4. **Output Control**

   - Stage 1 LED and Stage 2 LED are mapped to PORTB bits.
   - Side airbag LED also mapped to a PORTB bit.
   - The register `r18` is used as a bitmask, with each bit controlling a particular LED.

3. Relationship to Real-World Airbag Systems

3.1 Real SRS Airbag Operation

- **Airbag Control Module (ACM)** with internal G sensors, which continuously monitors vehicle acceleration or deceleration.
- **Driver/Passenger Airbag Modules** for frontal collisions.
- **Side Airbag/Curtain Airbag Modules** for lateral collisions.
- **Seat Belt Pretensioners** that lock belts upon collision.

In a real scenario, multiple sensors (front sensors, side sensors, seat position sensors, occupant detection, etc.) communicate with the central control module to judge crash severity. Two-stage inflators then adjust how quickly (and how forcefully) airbags deploy.

3.2 Simplification in This Project

- **Potentiometer in place of an accelerometer or G sensor**: Real vehicles use MEMS-based accelerometers. Here, rotating the potentiometer simulates various levels of impact force.
- **Push Button for Side Impact**: Instead of having a dedicated side crash sensor, a push button simply toggles the side impact bit.
- **Safety Sensor**: Rather than a complex seatbelt sensor or occupant classification system, a single digital input on PC4 simulates whether the system can or cannot fire.
- **LED Indicators**: Real vehicles have squibs (inflators) that physically deploy the airbag. For demonstration, LEDs indicate which airbag stage would be triggered.

4. Software Implementation

Below is an overview of how the assembly code works:

Reset and Initialization

```
.org 0x0000
  rjmp initialize
```

1.
    - CPU vectors to `initialize` at reset.

Delay Subroutine

```
delay_1s:
  ldi r24, 82
  ...
  ret
```

2.
    - Approximate 1-second delay at 16 MHz.
    - A six-second startup delay is achieved by calling this subroutine six times.

I/O Configuration

```
initialize:
  ; PORTD for startup LED, PD0 as output
  ldi r16, 0b00000001
  out DDRD, r16
  out PORTD, r16   ; Turn on PD0 LED

  ; PORTB for LED outputs
  ser r16
  out DDRB, r16    ; All PORTB pins as output
```

3.
    - **PORTD** pin 0 (PD0) → Startup LED.

- **PORTB** used for stage and side airbag LEDs.

ADC Setup

```
ldi r16, (1<<REFS0)    ; AVCC as reference
sts ADMUX, r16
ldi r16, (1<<ADEN) | (1<<ADPS0) | (1<<ADPS1) | (1<<ADPS2)
sts ADCSRA, r16        ; Enable ADC, prescaler=128
```

4.
    - ADC enabled, AVCC as reference, 128 prescaler for stable ADC reads.
    - The potentiometer's output is connected to the ADC channel (default is ADC0 if lower bits of ADMUX are zero).

Main Loop and Collision Detection

```
adc_loop:
  in r19, PINC       ; Read safety sensor (PC4) and side sensor (PC5)
  sbrs r19, 4        ; If PC4 is not active (0), jump to no_deployment
  ...
```

5.
    - **Safety sensor check**: If safety is off, the system disallows deployment.

ADC Conversion:

```
sbr r16, (1<<ADSC)   ; Start conversion
...
lds r16, ADCL
lds r17, ADCH
```

- The upper ADC result (`r17`) is used to decide the threshold (Stage 1 or Stage 2).

Stage 1 and Stage 2 Logic

```
cpi r17, 3
brsh check_high    ; If ADC ≥ 3, check for Stage 2

cpi r17, 2
brsh set_stage1    ; If ADC ≥ 2, set Stage 1
```

6.
    - **Stage 1**: `r17 >= 2 but < 3`
    - **Stage 2**: `r17 >= 3 or more`
    - Both thresholds are simplified for demonstration.

Side Airbag Logic

sbrs r19, 5
ori r18, 0b00001000

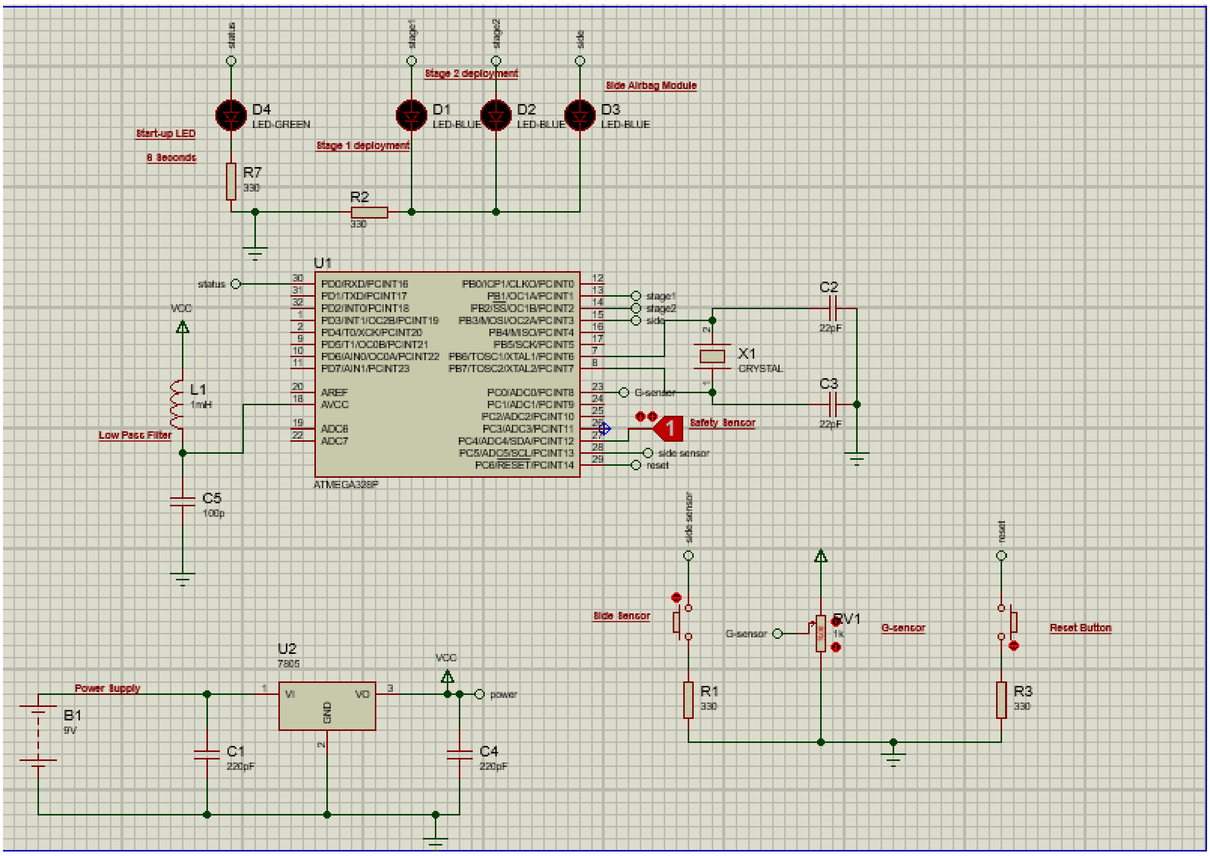
7.
   - Checks if PC5 (side sensor) is triggered; sets the side airbag LED bit if so.

LED Updates

out PORTB, r18
rjmp adc_loop

8.
   - The system continuously updates PORTB to reflect the current stage(s).
   - Bits in `r18`:
     - `0b00000010` → Stage 1 LED
     - `0b00000100` → Stage 2 LED
     - `0b00001000` → Side airbag LED

5. Testing and Results



- **Startup Test**: The “startup” LED (PD0) lights up for 6 seconds, confirming the system powers on and runs the self-test delay.

- **Safety Sensor Test**: If the safety sensor input (PC4) is deactivated, no airbag LEDs will light, regardless of the potentiometer or side impact inputs.
- **Frontal Collision Simulation**:
    1. Setting the potentiometer to above 50% but below 75% triggers Stage 1 LED.
    2. Increasing it further triggers Stage 2 LED.
- **Side Collision Simulation**: Pressing the side impact button (PC5) activates the side airbag LED.

6. Conclusion

By combining the ATmega328P microcontroller, basic sensor inputs (potentiometer and push buttons), and a series of output LEDs, this system demonstrates how multiple airbag stages can be triggered based on varying collision severities.

This project highlights:

- How ADC readings can emulate impact severity thresholds.
- The role of a safety sensor in preventing accidental deployments.
- How side collisions can be independently detected and triggered.

In a real vehicle, considerably more sophistication is involved, including occupant sensing, seat belt pretensioners, and robust diagnostic routines. Nonetheless, the basic architecture and logic flow here echo the fundamental principles of modern SRS airbag systems.

References

- “AIRBAG SYSTEM” Manual
- ATmega328P Datasheet.
- AVR instruction set.