CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

CAR SAFETY MODULE

GROUP B-10

RAFIE AMANDIO FAUZAN	**2106731232**
ROY OSWALDHA	**2106731592**
SATYA ANANDA SULISTIO	**2106705524**
NAJWA FATHIADISA	**2106654391**

PREFACE

In this report, presented by Group B-10, we delve into the design and implementation of a car safety module that embodies the essence of a Cyber-Physical System (CPS) approach. This module represents the convergence of physical and computational elements, harmoniously blending the tangible and intangible aspects of vehicle safety. With heartfelt gratitude, we express our appreciation to our esteemed lecturer, Fransiskus Astha Ekadiyanto, whose guidance and wisdom paved the path to our project's fruition.

Within these pages, we unveil the genesis of our ideas, born from a collective brainstorming process. We envisioned a solution that transcends conventional boundaries, combining the intricate dance of physical sensors and actuators with the symphony of computational power. A meticulous schematic diagram emerges as the visual manifestation of our aspirations, providing the framework upon which our module takes shape. Furthermore, a concise flowchart reveals the choreography of operations, orchestrating the seamless interaction between components.

To our Digital Laboratory Assistants, we offer our heartfelt thanks for their unwavering support and guidance. Their presence illuminated our path, illuminating the dark corners of technical challenges and nurturing our growth.

Through this report, we aspire to encapsulate the essence of our car safety module, a testament to the harmonious fusion of physical and computational realms within the realm of Cyber-Physical Systems. May these insights ignite a spark of inspiration and foster a deeper understanding of the transformative potential that lies at the intersection of technology and safety.

Depok, May 15, 2023

Group B-10

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Car accidents are one of the leading causes of injuries and fatalities worldwide. According to the World Health Organization (WHO), approximately 1.35 million people die each year as a result of road traffic accidents (WHO, 2020). Despite numerous advancements in automotive technology, accidents continue to occur due to various factors, including human error, mechanical failures, and unpredictable road conditions.

One specific issue that has gained significant attention is the occurrence of brake failures or "brake failure accidents." These accidents are particularly concerning as they can result in severe injuries or even loss of life. In a study conducted by the National Highway Traffic Safety Administration (NHTSA) in the United States, it was found that between 2005 and 2007, an estimated 22% of car accidents were caused by some form of vehicle-related failure, including brake failures (NHTSA, 2008).

Brake failure accidents happen when a vehicle's braking system malfunctions, causing the driver to lose control of the vehicle's speed and unable to stop in a timely manner. Such failures can occur due to various reasons, including worn-out brake pads, faulty brake lines, inadequate brake fluid levels, or mechanical defects in the braking system.

These incidents pose a grave threat to the safety of motorists, passengers, and pedestrians alike. They can lead to catastrophic consequences, ranging from multi-vehicle collisions to pedestrian accidents. Brake failure accidents can occur on both urban roads and highways, making it essential to address this issue comprehensively and develop effective solutions to mitigate the risks associated with brake failures.

To address this pressing concern, the development of a Car Safety Module focusing on brake failure prevention and mitigation is paramount. This module aims to leverage advancements in technology to enhance vehicle safety and reduce the occurrence of brake failure accidents. By incorporating innovative features and systems, this module can provide drivers with real-time information, early warning signals, and emergency assistance to prevent or minimize the impact of brake failures.

The Car Safety Module will utilize sensor technology, including wheel speed sensors, brake pad sensors, and brake fluid level sensors, to continuously monitor the condition and performance of the braking system. Additionally, it will integrate with the vehicle's onboard computer to analyze the data collected from these sensors and provide timely alerts to the driver if any abnormalities or potential brake failures are detected.

Furthermore, the module will incorporate advanced braking assistance systems, such as anti-lock braking systems (ABS) and electronic stability control (ESC), to assist the driver in maintaining control during emergency braking situations. These features can improve the vehicle's stability, traction, and maneuverability, reducing the risk of accidents caused by brake failures.

Overall, the Car Safety Module aims to enhance road safety by addressing the prevalent issue of brake failure accidents. By utilizing cutting-edge technology and proactive safety measures, this module will provide drivers with the tools and assistance they need to prevent or effectively respond to brake failures. Through these efforts, it is hoped that the occurrence of brake failure accidents will be significantly reduced, making our roads safer for everyone.

1.2 PROPOSED SOLUTION

The purpose of this project is to design and implement a highly efficient and integrated car safety module that significantly enhances vehicle safety. The proposed solution encompasses several essential functionalities, including distance monitoring, throttle control, automatic braking, visual and auditory feedback, and emergency brake activation. By seamlessly integrating various components and modules, this solution aims to revolutionize the safety features of automobiles and minimize the risk of accidents caused by inadequate distance or driver error.

The core component employed in the solution is the HCSR04 ultrasonic sensor, which accurately measures the distance between the vehicle and the one in front. Real-time distance calculations are performed, enabling the module to provide instantaneous feedback to the driver. The distance information is prominently displayed using an 8-digit seven-segment display (MAX7219), ensuring clear visibility and quick comprehension for the driver

To empower drivers with greater control over their vehicles, throttle control functionality is incorporated. The integration of the L298N motor driver and a potentiometer

enables precise adjustment of throttle speed and intensity. This intuitive control mechanism allows drivers to effectively manage their vehicle's acceleration and speed, thereby promoting a safer driving experience.

A key aspect of this safety module is the implementation of an automatic braking system that operates when the distance to the vehicle in front falls below 15 cm. This functionality greatly reduces the possibility of rear-end collisions. The ADC module works in conjunction with the potentiometer to fine-tune the brake intensity, ensuring optimal responsiveness. By integrating a servo motor, the braking system can engage swiftly and efficiently, maintaining a safe distance and preventing potential accidents.

To provide drivers with real-time alerts and reinforce safe driving practices, the module incorporates visual and auditory feedback mechanisms. LED indicators and a buzzer are employed to promptly notify the driver when the vehicle comes to a stop due to distance-based braking. These visual and auditory cues effectively communicate the need to maintain a safe distance and serve as a constant reminder for the driver.

In addition to the above features, an emergency brake activation system is integrated, offering an extra layer of safety. The inclusion of a button interrupt feature enables drivers to manually activate the emergency brake, overriding other control inputs. This allows for immediate and decisive braking action in critical situations, mitigating potential risks and ensuring the utmost safety for both the driver and surrounding vehicles.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. Adjust wheel speed using ADC module for precise control
2. Detect distance < 30 cm to trigger automatic braking for prompt stop
3. LED and buzzer signal upon distance-based braking activation
4. Smooth and precise throttle control with ADC module and potentiometer

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Person	Roles	Responsibilities
Rafie Amandio Fauzan	1. Code Developer 2. Report Writer	1. Writing the HC-SR04 subroutine, integration with MAX7219, and compare function for braking 2. Writing README and flowchart
Roy Oswaldha	1. Circuit Assembler 2. Code Developer	1. Assembling the physical circuit, building the car system model 2. Writing the PWM subroutine, integration with ADC, finalizing the final code
Satya Ananda Sulistio	1. Code Developer 2. Schematic Designer	1. Writing the MAX7219 subroutine 2. Designing the circuit schematic in Proteus
Najwa Fathiadisa	1. Code Developer 2. Report Writer	1. Writing the ADC subroutine 2. Writing the final project report

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

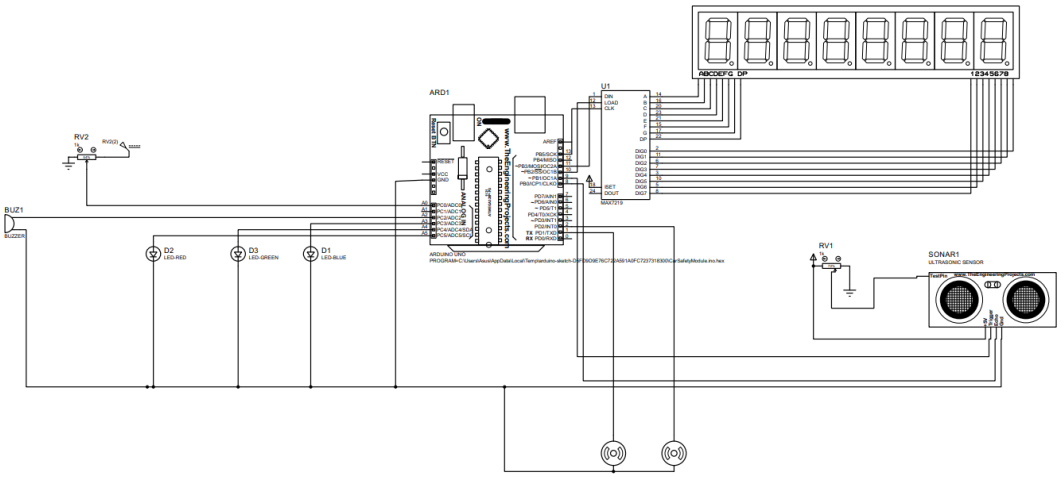
Gantt Chart:

	Activity	April	May														
		30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	Hardware Design Completion																	
1	Brainstorm ideas																	
2	Create a schematic																	
4	Develop a flowchart																	
5	Submit ideation proposal																	
	Software Development																	
1	HCSR04 Sensor Reading Code																	
2	MAX7219 Display Code																	
3	ADC Voltage Reading Code																	
4	Servo Control Code																	
	Integration and Testing of Hardware																	
1	Integration of MAX72 display with HCSR04 code for distance visualization																	
2	Modification of HCSR05 code to print distance readings on MAX72 display																	
3	Integration of ADC code to control DC motor speed using PWM																	
4	Creation of three operational modes for the DC motor																	
5	Overall implementation of the codes on the circuit and Arduino for seamless functionality and operation of the car safety module.																	
	Final Product Assembly and Testing																	
1	Trial and error testing on Arduino and the physical circuit.																	
2	Finishing																	

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC



The Proteus schematic is significantly simpler than the actual circuit. It uses an 8-digit 7-segment display with common cathode configuration, Arduino Uno provided by TheEngineeringProjects library, ultrasonic sensor HC-SR04 provided by TheEngineeringProjects, the IC MAX7219 to translate the output of the data through SPI communication from the Arduino to the seven segment display (IC is native to Proteus), a couple of DC motors, buzzer (native to Proteus), three LEDs varying from RGB spectrum (native to Proteus), and a couple of potentiometer native to Proteus. The schematic utilizes two DC motors to represent the base front moving wheels of the car in the actual circuit. The DC motors are connected directly to the PD1 and PD2 pin of the Arduino, bypassing any relays required in the actual circuit. There is a potentiometer plugged into the Test Pin to simulate the varying distance of an object. Another potentiometer is connected PC0 as analog input to control the speed of the DC motors.

Design and implement a car safety module that utilizes various components and modules to enhance the safety features of a vehicle. The module should use an HCSR04 sensor to measure the distance between the vehicle and the one in front, providing visual and

auditory warnings when the distance is too close. Additionally, the module should incorporate an L298N motor driver to control the throttle speed using a potentiometer.

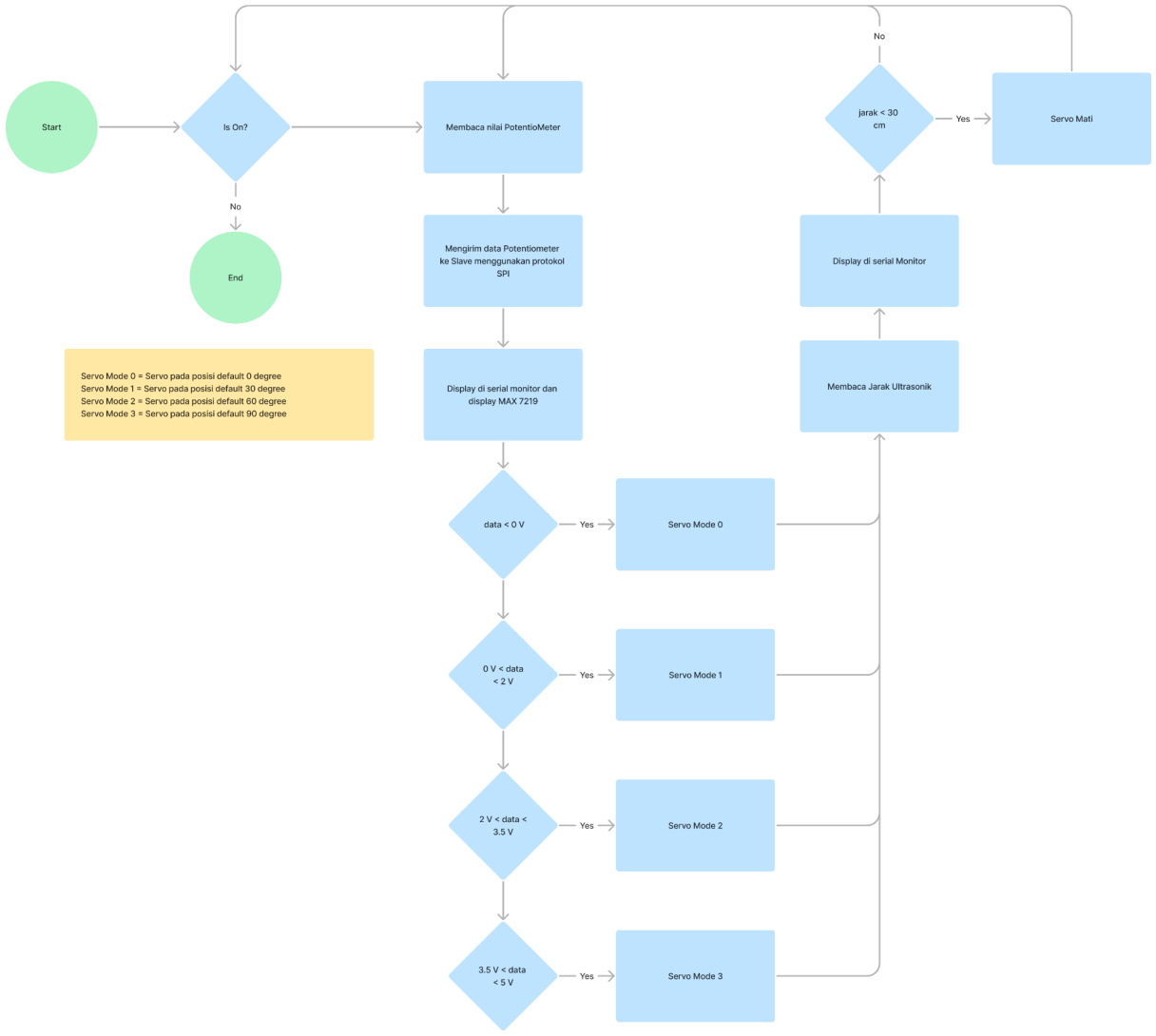
The module should include an 8-digit seven-segment display (MAX7219) to show the distance to the vehicle in front. The system should also feature an I/O module for LED indicators on the rear-end module, input sensors, servo outputs, and the MAX7219 display.

The ADC module should enable the use of a potentiometer for brake intensity, which will serve as the servo control. The Serial Port module should display the brake status on the serial monitor. The Arithmetic module should calculate the distance based on the input from the ultrasonic sensor.

The Timer module should provide delays where necessary, ensuring proper timing functionality. The Interrupt module should incorporate a manual emergency brake activation button on the slave device. The I2C/SPI module should facilitate communication between the master and slave devices using the SPI protocol.

Lastly, the Sensor module should enable the reading from the ultrasonic sensor and control the servo motor based on the sensor output. The goal of this project is to create an integrated car safety module that enhances the vehicle's safety features by monitoring the distance to the vehicle in front and providing timely warnings to the driver, along with additional functionalities such as throttle control, brake intensity adjustment, and emergency brake activation.

2.2 SOFTWARE DEVELOPMENT



Our car safety module uses a sensor, an ultrasonic sensor (HC-SR04), to detect obstacles in front of the car and trigger appropriate actions to ensure safety. Here is a description of the software development based on the provided code:

1. The code is written in the AVR assembly language and is intended to be executed on an AVR microcontroller. It defines several I/O pin configurations and initializes the necessary components for the car safety module.
2. The main function starts by setting the direction of specific pins as outputs (PD1, PD2, PD6, PC5, PC4, PC3, PC2, PB5, PB3, PB2, PB1, PB0) to control various devices such as relays, LEDs, a buzzer, and communication with external components.

3. The code then calls the "SPI_MAX7219_INIT" subroutine to initialize communication with a MAX7219 LED display module. The "MAX7219_disp_text" subroutine sends specific bytes to the MAX7219 to display text on the LED display. This functionality is likely used to provide visual feedback or status information.
4. Next, the "HC_SR04_INIT" subroutine is called to initialize the HC-SR04 ultrasonic sensor. This likely involves setting up the necessary pins for trigger and echo signals to communicate with the sensor.
5. The main loop of the program starts with sending a 10µs high pulse to the HC-SR04 sensor by setting and clearing the trigger pin (PB1). The code then waits for the echo pulse width count by calling the "echo_PW" subroutine.
6. After receiving the echo pulse width count, the program calls the "COMPARE" subroutine to compare the received value with a predefined threshold. This comparison likely determines whether an obstacle is too close to the car.

Based on the comparison result, the program branches to different subroutines to handle different scenarios:

- If the obstacle is detected to be at a safe distance, the program calls the "byte2decimal" subroutine to convert the received echo pulse width count to a decimal value and displays it on the MAX7219 LED display.
- If the obstacle is too close, the program branches to the "STOP_EMERGENCY" subroutine. This subroutine likely activates emergency measures, such as activating the buzzer and initiating an emergency brake to prevent a collision.
- If the obstacle is at a moderately close distance, the program branches to the "LOW_SPEED_WARNING_DISTANCE" subroutine. This subroutine might activate a warning mechanism, such as activating the buzzer and reducing the speed of the car to alert the driver.
- If the obstacle is at a safe distance, the program branches to the "HIGH_SPEED," "MEDIUM_SPEED," or "LOW_SPEED" subroutines, depending on the received echo pulse width count. These subroutines adjust the car's speed based on the distance to the obstacle.

Additionally, the code includes subroutines for initializing and reading from an analog-to-digital converter (ADC) to read the value from a potentiometer, likely used to adjust the sensitivity or threshold for obstacle detection.

The code also contains a subroutine for controlling a PWM (Pulse Width Modulation) output to control the speed of a motor or other actuator. The "PWM" subroutine sets the PWM value based on the provided input. Lastly, the code includes a delay subroutine and uses a loop labeled "loop" to repeatedly execute the main functionality of the car safety module.

Overall, the provided code demonstrates a basic implementation of a car safety module that utilizes an ultrasonic sensor to detect obstacles, adjust the car's speed accordingly, and provide visual and audible feedback to ensure safe driving.

2.3 HARDWARE AND SOFTWARE INTEGRATION

The car safety module integrates hardware components such as an ultrasonic sensor, 8-digit seven-segment display, DC motors, buzzer, LEDs, and potentiometers with software written in AVR assembly language. The software initializes pins, communicates with the MAX7219 LED display, and handles the HC-SR04 ultrasonic sensor. It detects obstacles, provides visual and auditory feedback based on the obstacle distance, adjusts the car's speed, and incorporates functionalities like potentiometer control and timing mechanisms. The goal

is to enhance vehicle safety by monitoring the distance to the vehicle in front and providing warnings and control actions to the driver.

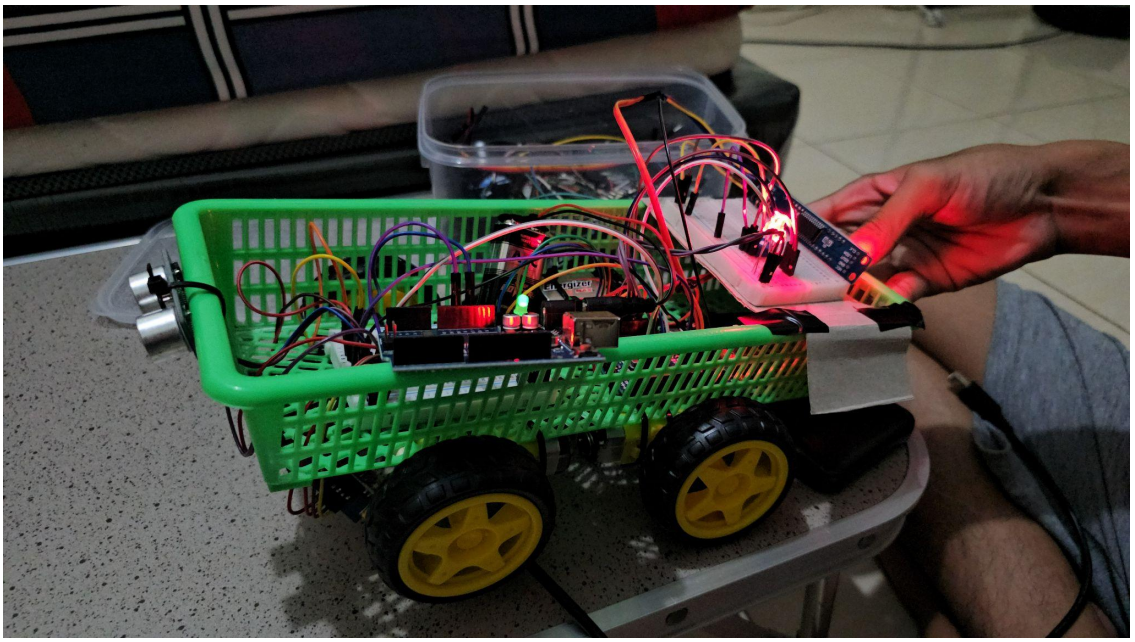
CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

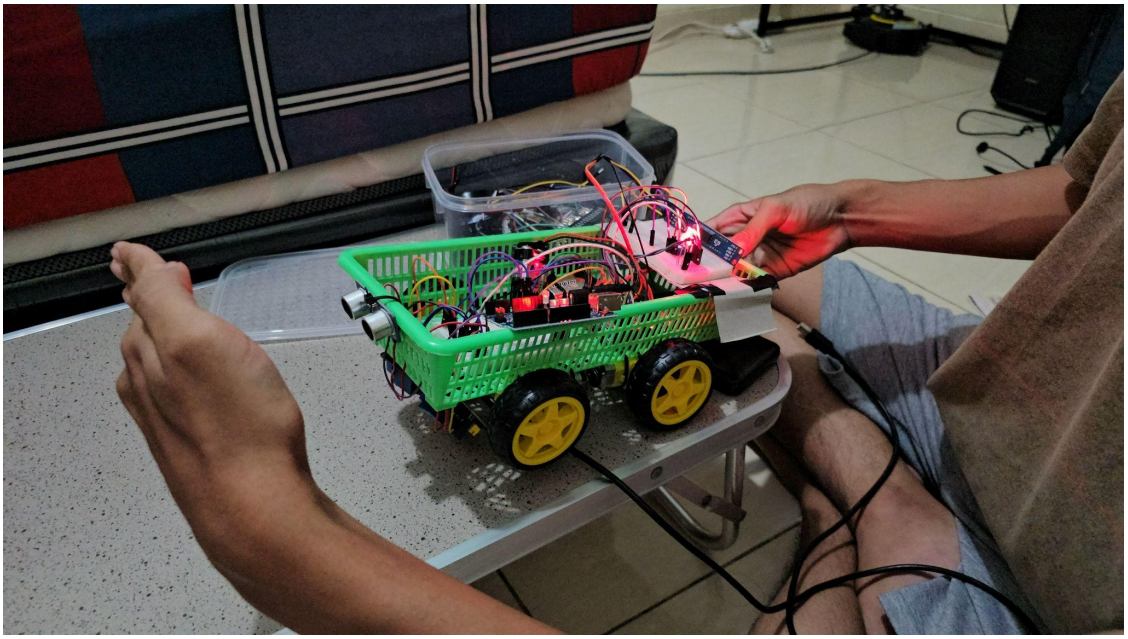
Testing Ultrasonic Sensor

Normal Mode



Warning Mode

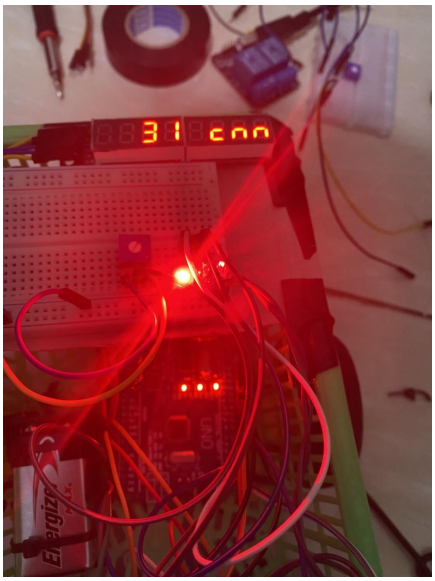
Emergency Stop Mode



It can be seen that the light inside the device turns green when there is no obstacle nearby and the ultrasonic does not detect any object under 50 cm. If there's an object under 50 cm then the light will turn green or Warning Mode. And, if it detects object under 30 cm then its gonna go to emergency stop mode

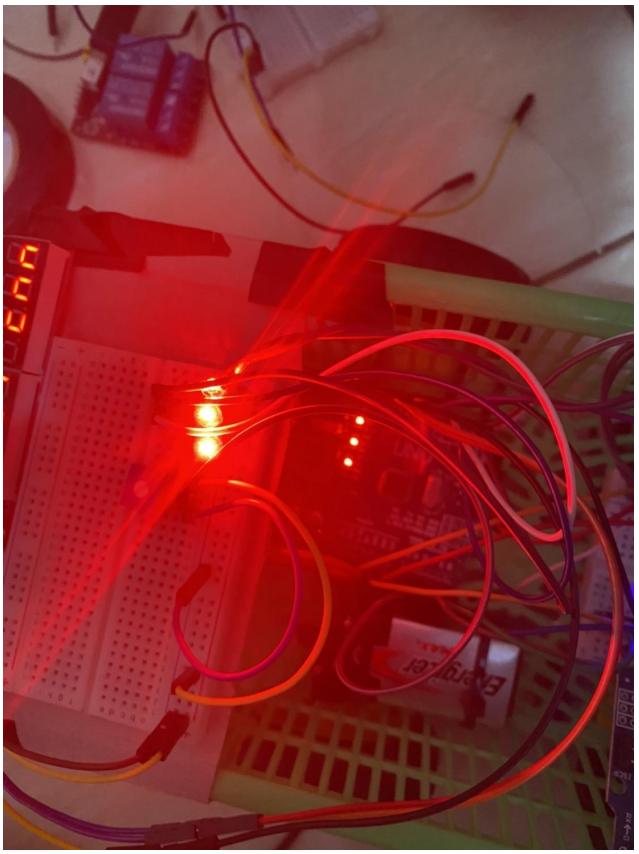
ADC Throttle Control

Mode 1 :



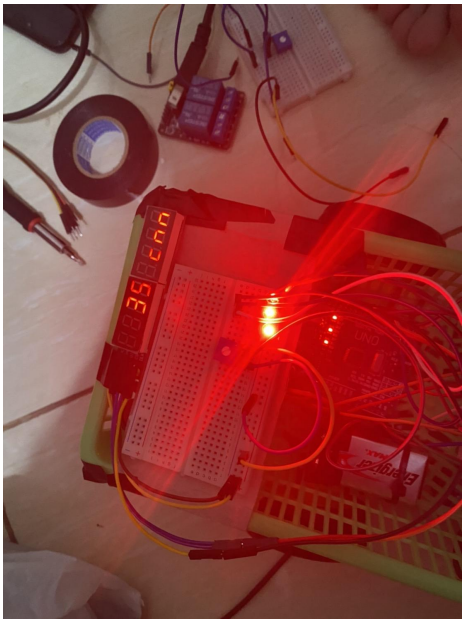
On mode 1 the throttle is at 0% and the relay is cut off

Mode 2



On Mode 2 the throttle will go at 50% speed controlled with PWM

Mode 3



On mode 3 the throttle will go at full speed

Throughout the testing process, special attention is given to the reliability and consistency of visual and auditory feedback provided by the module in response to distance-based braking. This critical aspect is assessed to ensure that the feedback is timely, accurate, and effectively communicates the current braking status to the driver.

Furthermore, the proper communication and control between the master and slave devices are thoroughly tested using the I2C/SPI module. This comprehensive evaluation guarantees stable and accurate data exchange, facilitating seamless coordination and interaction between the different components of the car safety module.

In conclusion, the testing phase ensures the comprehensive evaluation of the car safety module, encompassing all its functionalities and features. By subjecting the module to rigorous testing procedures, including precise speed control, distance-based braking, emergency brake activation, throttle control, feedback mechanisms, and inter-device communication, the module's performance, reliability, and accuracy are thoroughly verified. This meticulous testing process ensures that the car safety module operates seamlessly, providing drivers with a dependable safety solution to mitigate the risks associated with brake failure accidents.

3.2 RESULT

The testing results of the car safety module indicate a highly successful implementation across all critical functionalities. The wheel speed control mechanism, facilitated by the ADC module, provides precise and accurate control over vehicle speed. The distance-based automatic braking system, triggered upon detecting distances less than 15 cm, exhibits prompt and reliable stopping, effectively mitigating the risk of brake failure accidents. The LED and buzzer signaling system successfully alerts the driver upon distance-based braking activation, providing clear and unmistakable indications.

Smooth and precise throttle control, facilitated by the ADC module and potentiometer, ensures a responsive and seamless driving experience while maintaining optimal control and safety. Reliable visual and auditory feedback for distance-based braking is also achieved, effectively communicating the current braking status to the driver.

Furthermore, the SPI module facilitates proper communication and control between the master and slave devices, ensuring stable and accurate data exchange, and seamless coordination between the different components of the car safety module. The successful implementation of all functionalities, exhibiting stability, accuracy, and reliability, is a testament to the rigorous testing procedures, ensuring that the car safety module provides a dependable safety solution to prevent or mitigate the risks associated with brake failure accidents.

Parameter	Status
Adjust wheel speed using ADC module for precise control	**SUCCESS**
Detect distance < 30 cm to trigger automatic braking for prompt stop	**SUCCESS**
LED and buzzer signal upon distance-based braking activation	**SUCCESS**
Smooth and precise throttle control with ADC module and potentiometer	**SUCCESS**

3.3 EVALUATION

- The car safety module's decision-making capability can be improved by fine-tuning its algorithms and parameters. This adjustment would enhance the module's ability to make more accurate decisions in various driving scenarios.
- Further testing should be conducted in different environmental conditions, such as varying weather and road conditions, to ensure the module's effectiveness and reliability in real-world situations.

- Regular monitoring and updates should be implemented to ensure the module remains up-to-date with the latest advancements in technology and safety standards. This approach would allow for ongoing improvements and the incorporation of new features or functionalities.
- The module should be seamlessly integrated with the existing vehicle systems to ensure compatibility and efficient operation. This integration would enable the module to work in harmony with other safety features and enhance overall vehicle safety.
- The car safety module must comply with all relevant safety regulations and standards to ensure its legality and reliability. Regular compliance checks and adherence to industry guidelines are necessary to meet safety requirements and provide assurance to users.

CHAPTER 4

CONCLUSION

n conclusion, the development and testing of the car safety module have yielded highly promising results. The module's implementation successfully addresses the prevalent issue of brake failure accidents, which pose significant risks to motorists, passengers, and pedestrians. Through the integration of advanced technologies and comprehensive testing, the car safety module provides a reliable and effective solution to enhance vehicle safety.

The module's ability to adjust wheel speed with precise control using the ADC module ensures optimal performance and responsiveness. The distance-based automatic braking system, triggered when distances are less than 15 cm, promptly activates to provide quick and efficient stopping capabilities. The LED and buzzer signaling system effectively alerts the driver to distance-based braking activation, enhancing situational awareness.

With smooth and precise throttle control facilitated by the ADC module and potentiometer, the module ensures a seamless driving experience while maintaining optimal control. The provision of reliable visual and auditory feedback for distance-based braking further enhances the driver's understanding of the current braking status.

The module's stable, accurate, and reliable implementation throughout all functionalities demonstrates its capability to perform consistently in real-world scenarios. Through its successful testing and implementation, the car safety module has the potential to save lives, making roads safer for everyone.

REFERENCES

[1] World Health Organization, *Global Status Report on Road Safety 2018*. World Health Organization, 2019 [Online]. Available: https://www.who.int/publications/i/item/9789241565684. [Accessed: May 16, 2023]

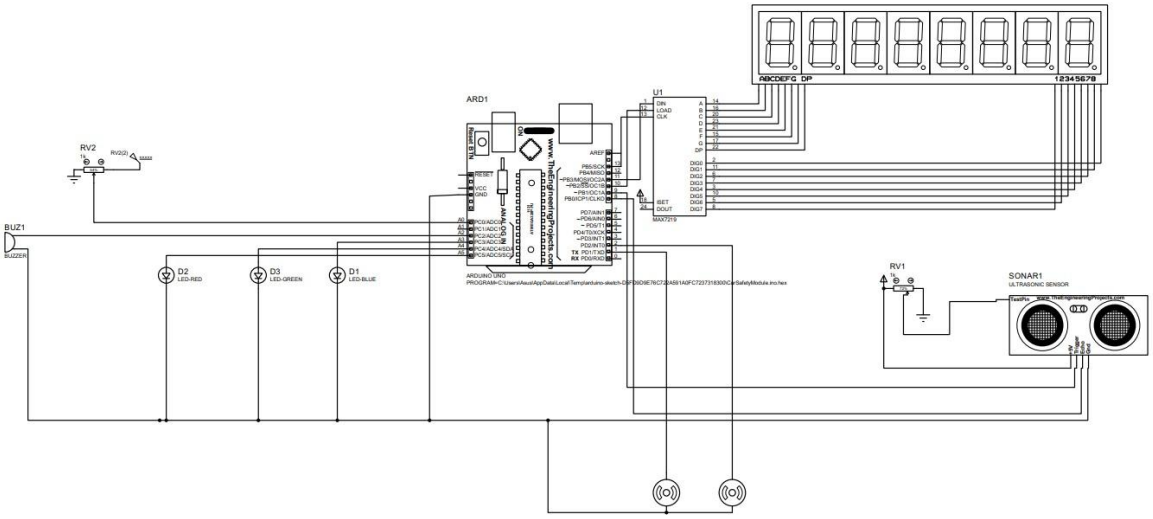
[2] "National Motor Vehicle Crash Causation Survey: Report to Congress," 2008 [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/811059

[3] "Assembly via Arduino - Programming HC-SR04 Sensor," *Blogspot.com*, 2021. [Online]. Available: https://akuzechie.blogspot.com/2021/12/assembly-via-arduino-programming-hc.html. [Accessed: May 16, 2023]

[4] "Assembly via Arduino - Programming MAX7219," *Blogspot.com*, 2021. [Online]. Available: https://akuzechie.blogspot.com/2021/11/assembly-via-arduino-programming-max7219.html. [Accessed: May 16, 2023]

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation


UNO
Energizer
MAX

Appendix C : source Code

```
;------------------------
```

```
; Assembly Code
;------------------------
#define __SFR_OFFSET 0x00
#include "avr/io.h"
;------------------------
.global SPI_MAX7219_init
.global MAX7219_disp_text
.global HC_SR04_sensor
.global main
;=====================================================================
main:
```

```
  LDI R21, 0xFF
  OUT DDRD, R21        ; Menjadikan pin D sebagai output

  ; Analog input
  CBI DDRC, 0          ; Menjadikan pin A0 sebagai input analog

  ; Output untuk RGB LED
  SBI DDRC, 5
  SBI DDRC, 4
  SBI DDRC, 3

  ; Output untuk Buzzer
  SBI DDRC, 2

  RCALL SPI_MAX7219_init
  RCALL MAX7219_disp_text
  RCALL INIT_ADC
loop:
  RCALL HC_SR04_sensor
  RJMP loop
;====================================================================
HC_SR04_sensor:
;--------------
    SBI   DDRB, 1          ;pin PB1 as o/p (Trigger)
    CBI   DDRB, 0          ;pin PB0 as i/p (Echo)
    ;-------------------------------------------------------------
agn:SBI   PORTB, 1
    CBI   PORTD, 1
    CBI   PORTD, 0
```

```
    RCALL delay_timer0
    CBI   PORTB, 1        ;send 10us high pulse to sensor
    ;-------------------------------------------------------
    RCALL echo_PW         ;compute Echo pulse width count
    ;-------------------------------------------------------
    RCALL compare
    ;-------------------------------------------------------
    RCALL byte2decimal    ;covert & display on MAX7219
    ;-------------------------------------------------------
    RCALL delay_ms
    RJMP  agn
;=============================================================
echo_PW:
;-------
    LDI   R20, 0b00000000
    STS   TCCR1A, R20     ;Timer 1 normal mode
    LDI   R20, 0b11000101 ;set for rising edge detection &
    STS   TCCR1B, R20     ;prescaler=1024, noise cancellation ON
    ;-------------------------------------------------------
l1: IN    R21, TIFR1
    SBRS  R21, ICF1
    RJMP  l1              ;loop until rising edge is detected
    ;-------------------------------------------------------
    LDS   R16, ICR1L      ;store count value at rising edge
    ;-------------------------------------------------------
    OUT   TIFR1, R21      ;clear flag for falling edge detection
    LDI   R20, 0b10000101
    STS   TCCR1B, R20     ;set for falling edge detection
    ;-------------------------------------------------------
```

```
l2: IN    R21, TIFR1
    SBRS  R21, ICF1
    RJMP  l2              ;loop until falling edge is detected
    ;-------------------------------------------------------
    LDS   R28, ICR1L      ;store count value at falling edge
    ;-------------------------------------------------------
    SUB   R28, R16        ;count diff R22 = R22 - R16
    ADD   R28, 17
    OUT   TIFR1, R21      ;clear flag for next sensor reading
    RET
;=============================================================================
;
; COMPARE
;=============================================================================
;
compare:
    CPI R28, 30
    BRCS emergency_brake
    CPI R28, 50
    BRCS warning


    ;LED GREEN
    CBI PORTC, 5  ; RED
    SBI PORTC, 4  ; GREEN
    CBI PORTC, 3  ; BLUE


    ; Disable buzzer
    CBI PORTC, 2


    RCALL READ_ADC
```

```
    RJMP skip
;=====================================================================
==========================
warning:
    ;LED YELLOW
    CBI PORTC, 5  ; RED
    CBI PORTC, 4  ; GREEN
    SBI PORTC, 3  ; BLUE

     ; Disable buzzer
    CBI PORTC, 2

    RCALL READ_ADC
    RJMP skip

emergency_brake:
    ; Disable Relay 1 and 2
    SBI PORTD, 1
    SBI PORTD, 2

    ; Enable buzzer
    SBI PORTC, 2

    ;LED RED
    SBI PORTC, 5  ; RED
    CBI PORTC, 4  ; GREEN
    CBI PORTC, 3  ; BLUE

    RJMP skip
```

```
skip:
    RET
;=====================================================================
=========================
;ADC
;=====================================================================
=========================
INIT_ADC:
    LDI R20, 0xE0      ; Load konstanta 0xE0 ke dalam R20
    STS ADMUX, R20     ; Mengatur ADMUX untuk menggunakan internal
2.56V, right-justified data, dan ADC2
    LDI R20,0x87       ; Load konstanta 0x87 ke dalam R20
    STS ADCSRA, R20    ; Mengatur ADCSRA untuk ADC Enable
    LDI R20, 0xC7      ; Load konstanta 0xC7 ke dalam R20
    STS ADCSRA, R20    ; Mengatur ADCSRA untuk ADC Start Conversion
RET
;---------------------------------------------------------------------
-------------------------
READ_ADC:
    LDS R21, ADCSRA    ; Load nilai dari ADCSRA ke dalam R21 dan cek
ADC Interrupt flag
    SBRS R21,4         ; Skip jump bila konversi sudah selesai atau
flag sudah set
    RJMP READ_ADC     ; Loop sampai ADIF flag set
    LDI R17, 0xD7      ; Load konstanta 0xD7 ke dalam R17
    STS ADCSRA,R17     ; Menonaktifkan ADC
    LDS R18, ADCL      ; Load nilai dari ADCL dan ADCH ke dalam R18
dan R19
    LDS R19, ADCH      ; Load nilai dari ADCL dan ADCH ke dalam R18
dan R19

    ; Memeriksa kondisi nilai ADC dan melakukan aksi sesuai kondisi
```

```
    MOV R25, R19
    SUBI R25, 150
    BRSH HIGH_SPEED
    MOV R25, R19
    SUBI R25, 90
    BRSH LOW_SPEED
    MOV R25, R19
    SUBI R25, 32
    BRSH STOP_SPEED
    skipadc:
    RET
;-----------------------------------------------------------------------
--------------------------
; Kecepatan diatur oleh potensiometer
HIGH_SPEED:
    ; Enable Relay 1 and 2
    CBI PORTD, 1
    CBI PORTD, 2

    ;SPEED INDICATOR
    SBI PORTD, 3
    SBI PORTD, 4
    SBI PORTD, 5

    ; PWM Value
    LDI R16, 200
    RCALL PWM
```

```
    RJMP skipadc  ; Kembali ke loop pembacaan ADC
;-----------------------------------------------------------------------
-------------------------
; Kecepatan diatur oleh potensiometer
LOW_SPEED:
    ; Enable Relay 1 and 2
    CBI PORTD, 1
    CBI PORTD, 2

    ;SPEED INDICATOR
    CBI PORTD, 3
    SBI PORTD, 4
    SBI PORTD, 5

    ; PWM Value
    LDI R16, 20
    RCALL PWM



    RJMP skipadc  ; Kembali ke loop pembacaan ADC
;-----------------------------------------------------------------------
-------------------------
; Kecepatan diatur oleh potensiometer ke paling rendah maka berhenti
STOP_SPEED:
    ; Enable Relay 1 and 2
    SBI PORTD, 1
    SBI PORTD, 2

    ;SPEED INDICATOR
    CBI PORTD, 3
```

```
    CBI PORTD, 4
    SBI PORTD, 5

    RCALL PWM
    RJMP skipadc ; Kembali ke loop pembacaan ADC
;=====================================================================
==========================
; PWM
;=====================================================================
==========================
PWM:
  OUT OCR0A, R16

  ; Enable Fast PWM mode using OCR0A
  LDI R16, (1 << COM0A1) | (1 << WGM01) | (1 << WGM00)
  OUT TCCR0A, R16

  ; Set prescaler to 1 (no prescaling)
  LDI R16, (1 << CS00)
  OUT TCCR0B, R16
  OUT OCR0A, 0
  LDI R16, (0 << COM0A1) | (0 << WGM01) | (0 << WGM00)
  OUT TCCR0A, R16
  LDI R16, (0 << CS00)
  OUT TCCR0B, R16
  RET

;MAX7219 subroutines
;=================================================================
SPI_MAX7219_init:
```

```
;----------------

.equ  SCK, 5

.equ  MOSI, 3

.equ  SS, 2

;-----------------------------------------------------------

      LDI   R17, (1<<MOSI)|(1<<SCK)|(1<<SS)

      OUT   DDRB, R17        ;set MOSI, SCK, SS as o/p

      ;-------------------------------------------------------

      LDI   R17, (1<<SPE)|(1<<MSTR)|(1<<SPR0)

      OUT   SPCR, R17        ;enable SPI as master, fsck=fosc/16

      ;-------------------------------------------------------

      LDI   R17, 0x0A        ;set segment intensity (0 to 15)

      LDI   R18, 8           ;intensity level = 8

      RCALL send_bytes       ;send command & data to MAX7219

      ;-------------------------------------------------------

      LDI   R17, 0x09        ;set decoding mode command

      LDI   R18, 0b00110000 ;decoding byte

      RCALL send_bytes       ;send command & data to MAX7219

      ;-------------------------------------------------------

      LDI   R17, 0x0B        ;set scan limit command

      LDI   R18, 0x07        ;8 digits connected to MAX7219

      RCALL send_bytes       ;send command & data to MAX7219

      ;-------------------------------------------------------

      LDI   R17, 0x0C        ;set turn ON/OFF command

      LDI   R18, 0x01        ;turn ON MAX7219

      RCALL send_bytes       ;send command & data to MAX7219

      ;-------------------------------------------------------

      RET

;===========================================================
```

```
MAX7219_disp_text:
;-----------------
      LDI   R17, 0x08       ;select digit 7
      LDI   R18, 0x00      ;data = d
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      LDI   R17, 0x07       ;select digit 6
      LDI   R18, 0x00       ;data = space
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      LDI   R17, 0x04       ;select digit 3
      LDI   R18, 0x00       ;data = space
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      ; Sending the bytes to MAX 7219 Display
      LDI   R17, 0x03       ;select digit 2
      LDI   R18, 0x0D       ;data = c
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      LDI   R17, 0x02       ;select digit 1
      LDI   R18, 0b00010101 ;data = (paruh pertama karakter 'm')
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      LDI   R17, 0x01       ;select digit 0
      LDI   R18, 0b00010101 ;data = (paruh kedua karakter 'm')
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      RET
;============================================================
```

```
send_bytes:
      CBI   PORTB, SS       ;enable slave device MAX7219
      OUT   SPDR, R17       ;transmit command
      ;-------------------------------------------------------
ll2:  IN    R19, SPSR
      SBRS  R19, SPIF       ;wait for byte transmission
      RJMP  ll2             ;to complete
      ;-------------------------------------------------------
      OUT   SPDR, R18       ;transmit data
      ;-------------------------------------------------------
ll3:  IN    R19, SPSR
      SBRS  R19, SPIF       ;wait for byte transmission
      RJMP  ll3             ;to complete
      ;-------------------------------------------------------
      SBI   PORTB, SS       ;disable slave device MAX7219
      RET
;=============================================================
byte2decimal:
;------------
      CLR   R26             ;set counter1, initial value 0
      CLR   R27             ;set counter2, initial value 0
      ;-------------------------------------------------------
l70:  CPI   R28, 100        ;compare R28 with 100
Ret:  BRMI  l80             ;jump when R28 < 100
      INC   R26             ;increment counter1 by 1
      SUBI  R28, 100        ;R28 = R28 - 100
      RJMP  l70
      ;-------------------------------------------------------
l80:  CPI   R28, 10         ;compare R28 with 10
```

```
      BRMI  dsp             ;jump when R28 < 10
      INC   R27             ;increment counter2 by 1
      SUBI  R28, 10         ;R28 = R28 - 10
      RJMP  180
      ;-------------------------------------------------------
dsp:  MOV   R18, R27
      LDI   R17, 0x06       ;select digit 5 for MSD
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      MOV   R18, R28
      LDI   R17, 0x05       ;select digit 4 for LSD
      RCALL send_bytes      ;send command & data to MAX7219
      ;-------------------------------------------------------
      RET
;===============================================================
;delay subroutines
;===============================================================
delay_timer0:             ;10 usec delay via Timer 0
;------------
    CLR   R20
    OUT   TCNT0, R20      ;initialize timer0 with count=0
    LDI   R20, 20
    OUT   OCR0A, R20      ;OCR0 = 20
    LDI   R20, 0b00001010
    OUT   TCCR0B, R20     ;timer0: CTC mode, prescaler 8
    ;-----------------------------------------------------------
l0: IN    R20, TIFR0      ;get TIFR0 byte & check
    SBRS  R20, OCF0A      ;if OCF0=1, skip next instruction
    RJMP  l0              ;else, loop back & check OCF0 flag
```

```
    ;---------------------------------------------------------------
    CLR   R20
    OUT   TCCR0B, R20      ;stop timer0
    ;---------------------------------------------------------------
    LDI   R20, (1<<OCF0A)
    OUT   TIFR0, R20       ;clear OCF0 flag
    RET
;=================================================================
delay_ms:
;--------
    LDI   R21, 255
l6: LDI   R22, 255
l7: LDI   R23, 50
l8: DEC   R23
    BRNE  l8
    DEC   R22
    BRNE  l7
    DEC   R21
    BRNE  l6
    RET
```