الجامعة الإسلامية العالمية ماليزيا

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

يونيبرسيتي إسلام انتارابغسا مليسيا

Mechatronics LAB III

MCT 4159

Section 1

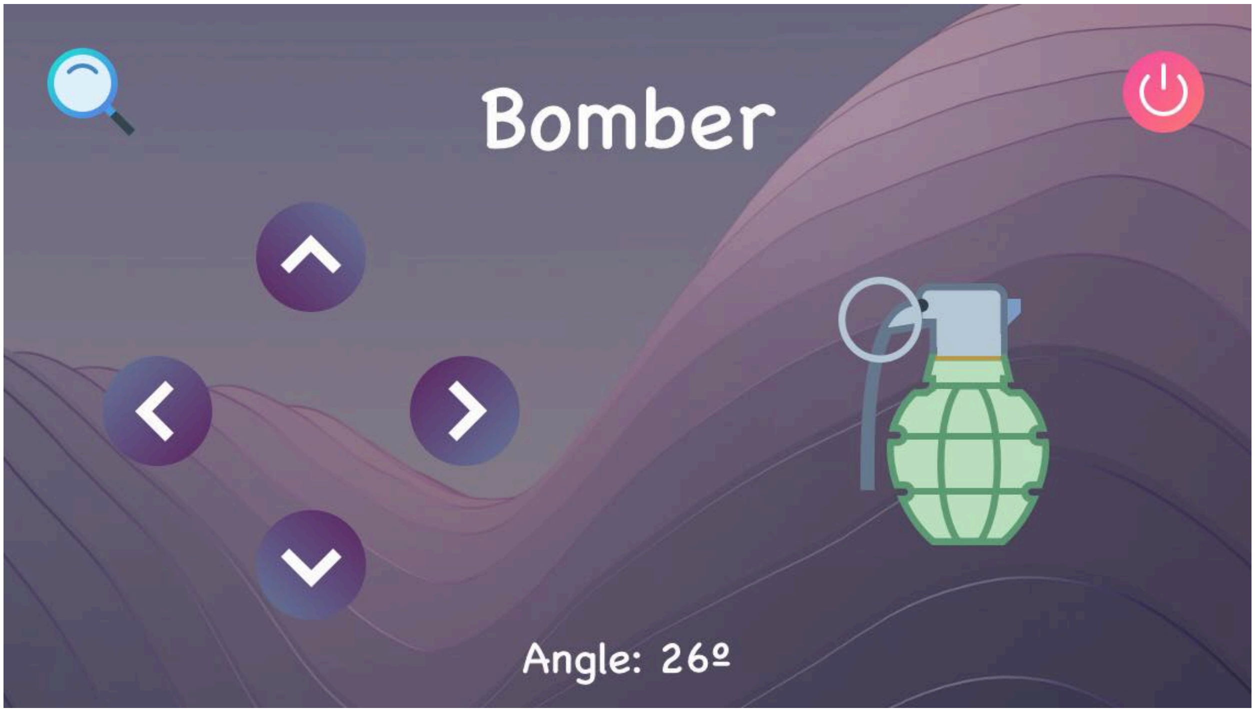
Bomber Project

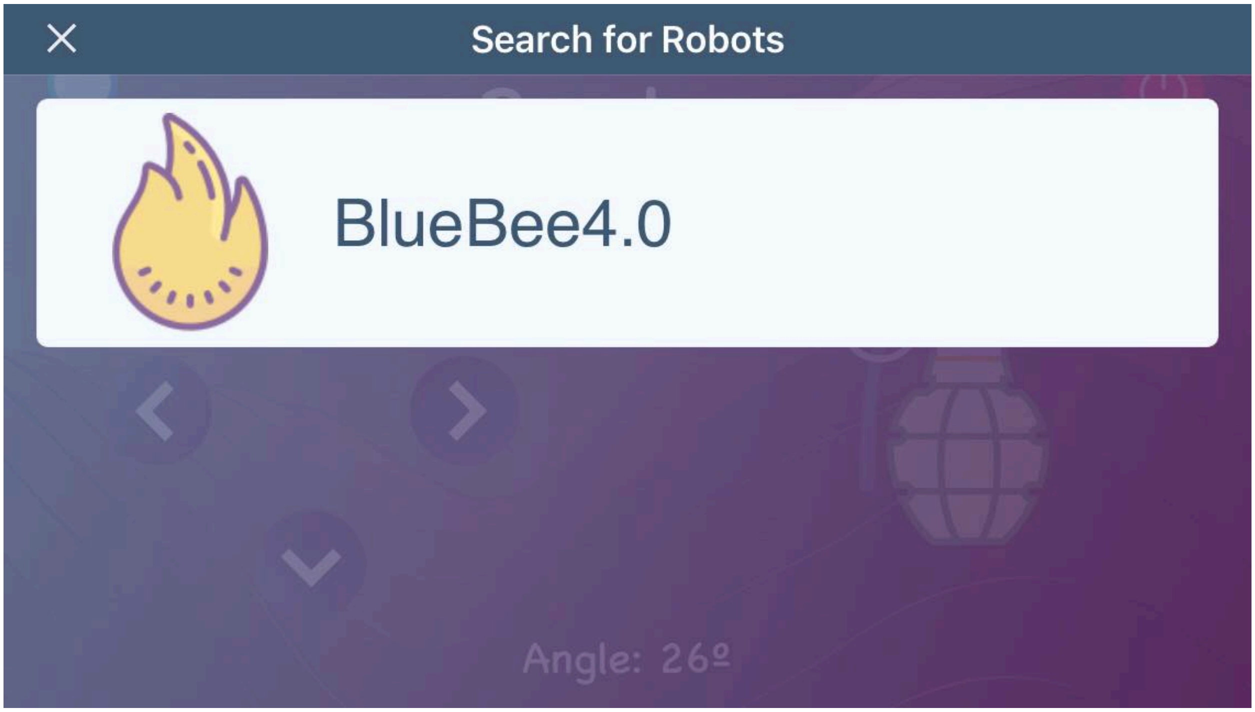
Group Members:

1. Saleem Hadad 1337479
2. Mahsin Saifullah 1424689
3. Hussein Al-Aidaros 1239417

Introduction & Description:

In this project we design Differential mobile robot applied projectile concept controlled via custom iOS application, embedded code is written in assembly.





Objectives:

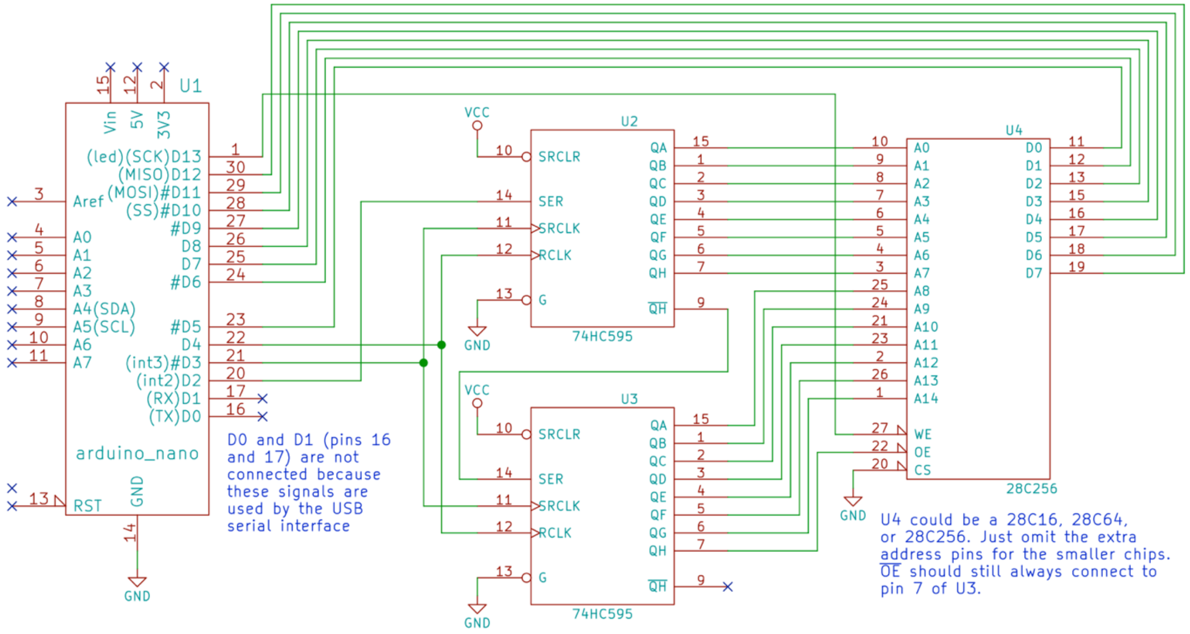
- To design a project that combine between MicroP and DLD.
- To demonstrate microprocessor controller 7 segment display and motors interface.
- To build a combinational circuit using several logic gates.

Equipment:

1. Atmega32 Controller
2. Potentiometer
3. LEDs
4. Buzzer
5. DC Motors
6. Motor Driver
7. Battery
8. 74LS08
9. Bluetooth Module
10. EEPROM
11. 7 Segment
12. iOS App
13. Jumper Wires

Circuit design:

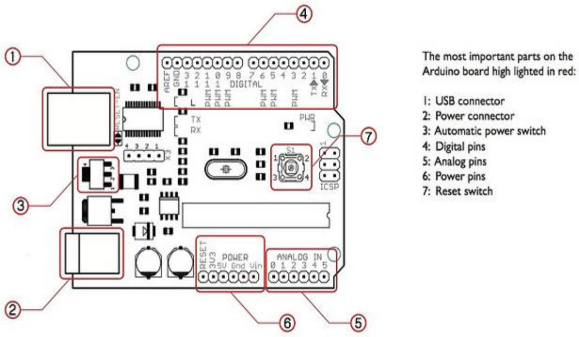
For DLD parts:



D0 and D1 (pins 16 and 17) are not connected because these signals are used by the USB serial interface

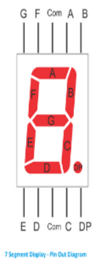
U4 could be a 28C16, 28C64, or 28C256. Just omit the extra address pins for the smaller chips. OE should still always connect to pin 7 of U3.

For microP parts:

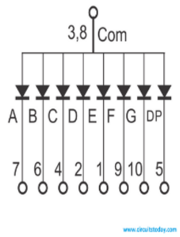


The most important parts on the Arduino board high lighted in red:

1: USB connector
2: Power connector
3: Automatic power switch
4: Digital pins
5: Analog pins
6: Power pins
7: Reset switch

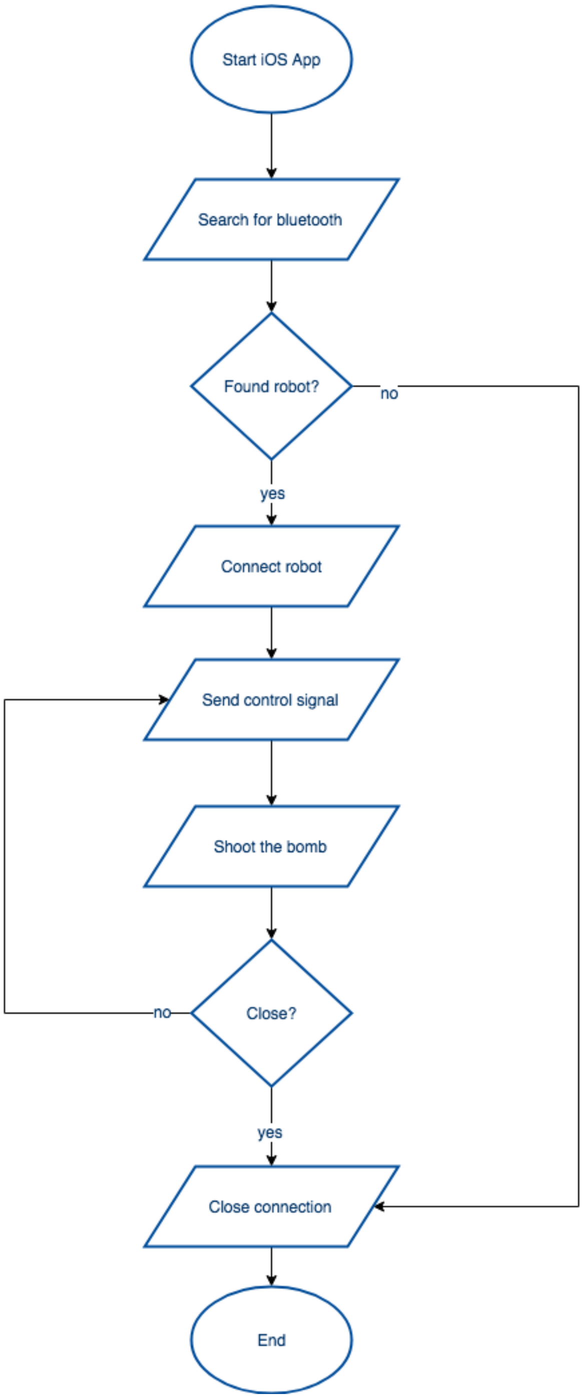


7 Segment Display - Pin Out Diagram

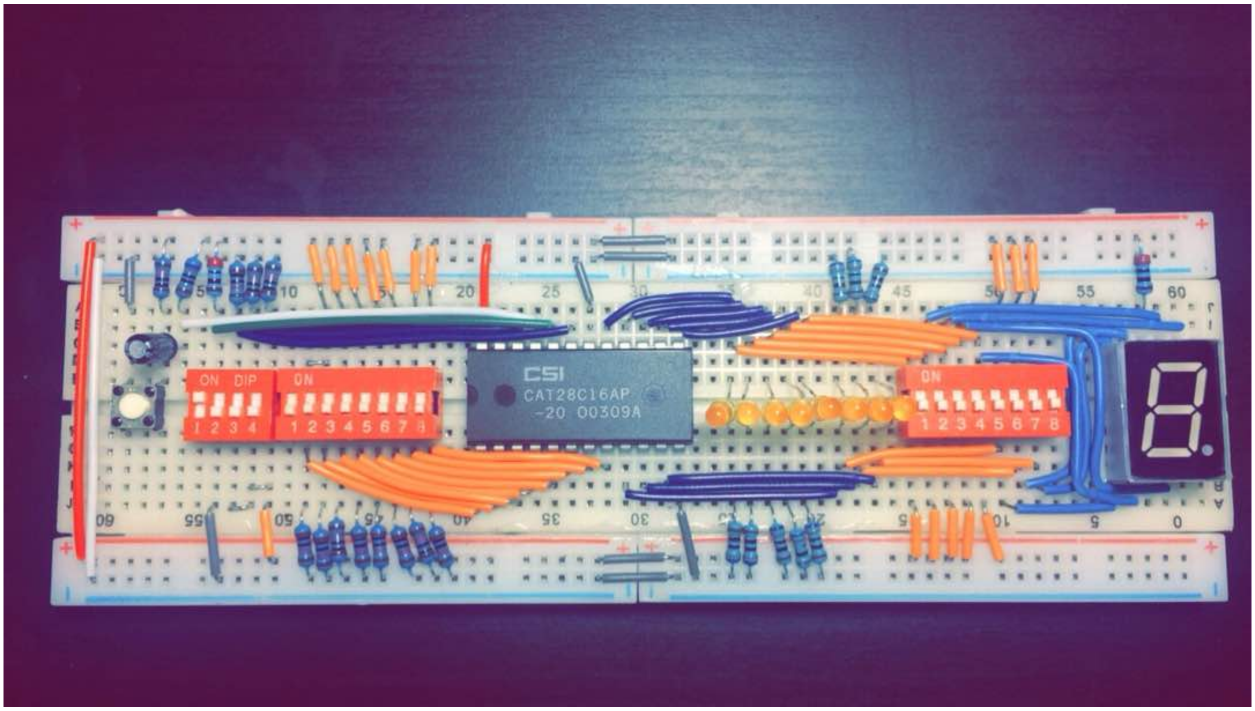


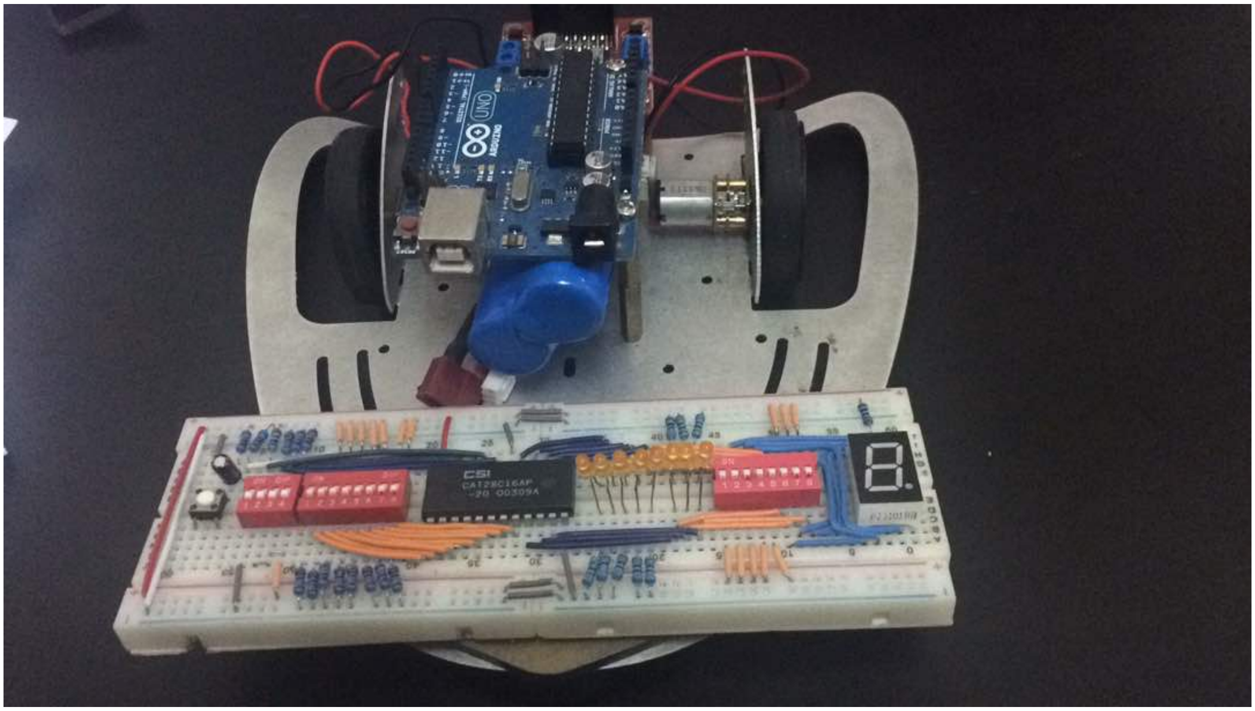
www.circuitstoday.com

Flowchart diagram:



All together:





Code:

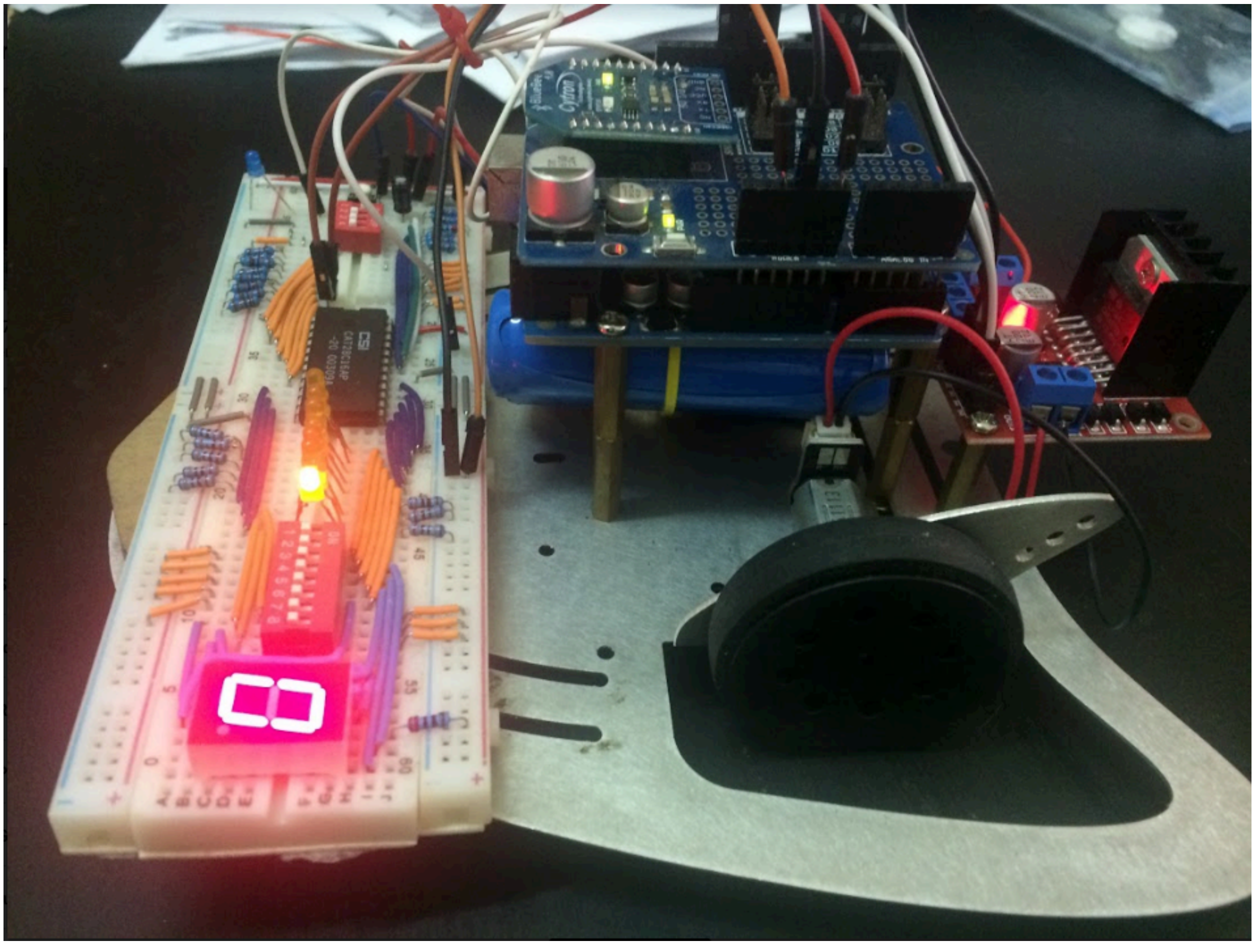
```
.file
"main.S"
           #include <avr/io.h>


           .text
           .global setup
           .global loop
           setup:
               RCALL setupPins
               RCALL setupUSART
               RCALL setupPWM
               RCALL setupADC
               RET


           loop:
               RCALL serialReceive
               RJMP loop


           setupPins:
               ; Set DP 3, 4 and 5 as Outputs
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               ; | DDRD | DDRD7 | DDRD6 | DDRD5 | DDRD4 | DDRD3 | DDRD2 | DDRD1 | DDRD0 |
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               ; | 0x2A |     0 |     0 |     1 |     1 |     1 |     0 |     0 |     0 |
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               LDI r16, 0b00111000
               STS DDRD, r16


               ; Set DP 8, 9, 10, 11 as Outputs
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               ; | DDRB | DDRB7 | DDRB6 | DDRB5 | DDRB4 | DDRB3 | DDRB2 | DDRB1 | DDRB0 |
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               ; | 0x2A |     0 |     0 |     0 |     0 |     1 |     1 |     1 |     1 |
               ; +------+-------+-------+-------+-------+-------+-------+-------+-------+
               LDI r16, 0b00001111
               STS DDRB, r16
               RET
```

```
setupUSART:
    ; set the baud rate to 115200
    ; +--------+---+---+---+---+---+---+---+---+
    ; | UBRR0L | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
    ; +--------+---+---+---+---+---+---+---+---+
    ; | 0xC4   | W | W | W | W | W | W | W | W |
    ; +--------+---+---+---+---+---+---+---+---+


    ; +--------+---+---+---+---+---+---+---+---+
    ; | UBRR0H | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
    ; +--------+---+---+---+---+---+---+---+---+
    ; | 0xC5   | W | W | W | W | W | W | W | W |
    ; +--------+---+---+---+---+---+---+---+---+
    LDI r16, 0x08
    LDI r17, 0x00
    STS UBRR0L, r16
    STS UBRR0H, r17


    ; set frame, 8 bits, 2 stop bit
    ; +--------+---------+---------+-------+-------+-------+--------+--------+--
------+
    ; | UCSR0C | UMSEL01 | UMSEL00 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 |
UCPOL0 |
    ; +--------+---------+---------+-------+-------+-------+--------+--------+--
------+
    ; | 0xC2   |       0 |       0 |     0 |     0 |     1 |      1 |      1 |
0 |
    ; +--------+---------+---------+-------+-------+-------+--------+--------+--
------+
    LDI r16, 0b00001110
    STS UCSR0C, r16


    ; enable the Tx and Rx of the serial
    ; +--------+--------+--------+--------+-------+-------+--------+-------+----
---+
    ; | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 |
TXB80 |
    ; +--------+--------+--------+--------+-------+-------+--------+-------+----
---+
    ; | 0xC1   |      0 |      0 |      0 |     1 |     1 |      0 |     0 |
0 |
    ; +--------+--------+--------+--------+-------+-------+--------+-------+----
---+
    LDI r16, (1 << RXEN0) | (1 << TXEN0)
    STS UCSR0B, r16
    RET


setupPWM:
    ; Timer 1 (right motor):
    ; pin 9     ->  OC1A
    ; pin 10    ->  OC1B


    ; Timer 2 (left motor)
    ; pin 11    ->  OC2A
    ; pin 3     ->  OC2B


    ; Timer 1:
    ; set PWM mode to phase-correct PWM with 8 bits
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    ; | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | - | - | WGM11 | WGM10 |
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    ; | 0x80   |      1 |      0 |      1 |      0 | 0 | 0 |     0 |     1 |
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    LDI r16, (1 << COM1A1) | (1 << COM1B1) | (1 << WGM10)
    STS TCCR1A, r16


    ; set pre-scaler to 64
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    ; | TCCR1A | ICNC1 | ICES1 | - | WGM13 | WGM12 | CS12 | CS11 | CS10 |
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    ; | 0x81   |     0 |     0 | 0 |     0 |     0 |    0 |    1 |    1 |
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    LDI r16, (1 << CS11) | (1 << CS10)
    STS TCCR1B, r16
    ; Timer 2:
    ; set PWM mode to phase-correct PWM
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    ; | TCCR2A | COM2A1 | COM2A0 | COM2B1 | COM2B0 | - | - | WGM21 | WGM20 |
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    ; | 0xB0   |      1 |      0 |      1 |      0 | 0 | 0 |     0 |     1 |
    ; +--------+--------+--------+--------+--------+---+---+-------+-------+
    LDI r16, (1 << COM2A1) | (1 << COM2B1) | (1 << WGM20)
    STS TCCR2A, r16


    ; set pre-scaler to 64
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    ; | TCCR2A | ICNC2 | ICES2 | - | WGM23 | WGM22 | CS22 | CS21 | CS20 |
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    ; | 0x81   |     0 |     0 | 0 |     0 |     0 |    1 |    0 |    0 |
    ; +--------+-------+-------+---+-------+-------+------+------+------+
    LDI r16, (1 << CS22)
    STS TCCR2B, r16
    RET


setupADC:
    ; set the ADC conversion on pin 0 port c, with AVCC ref.
    ; +-------+-------+-------+-------+---+------+------+------+------+
    ; | ADMUX | REFS1 | REFS0 | ADLAR | - | MUX3 | MUX2 | MUX1 | MUX0 |
    ; +-------+-------+-------+-------+---+------+------+------+------+
    ; | 0x7C  |     0 |     1 |     0 | 0 |    0 |    0 |    0 |    0 |
    ; +-------+-------+-------+-------+---+------+------+------+------+
    LDI r16, 0b01000000
    STS ADMUX, r16


    ; set pre-scaler to 128 and mode to auto-triggering, and start.
    ; +--------+------+------+-------+------+------+-------+-------+-------+
    ; | ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 |
    ; +--------+------+------+-------+------+------+-------+-------+-------+
    ; | 0x7A   |    1 |    1 |     1 |    0 |    0 |     1 |     1 |     1 |
    ; +--------+------+------+-------+------+------+-------+-------+-------+
    LDI r16, 0b11100111
    STS ADCSRA, r16
    RET
```

```
serialReceive:
    ; check first if the data has been received.
    LDS r17, UCSR0A
    SBRS r17, RXC0
    RJMP serialReceive


    ; get the ready data from buffer
    LDS r16, UDR0


    ; check if the action is left
    LDI r17, 75
    CP r16, r17
    BREQ moveLeft


    ; check if the action is right
    LDI r17, 80
    CP r16, r17
    BREQ moveRight


    ; check if the action is forward
    LDI r17, 85
    CP r16, r17
    BREQ moveForward


    ; check if the action is backward
    LDI r17, 90
    CP r16, r17
    BREQ moveBackward


    ; check if the action is shoot
    LDI r17, 95
    CP r16, r17
    BREQ shoot


    ; check if the action is stop
    LDI r17, 100
    CP r16, r17
    BREQ stop
    RET


moveLeft:
    LDI r16, 0
    LDS r17, ADCL
    LDS r18, ADCH


    STS OCR1A, r17
    STS OCR2A, r16
    STS OCR1B, r16
    STS OCR2B, r17
    RET


moveRight:
    LDI r16, 0
    LDS r17, ADCL
    LDS r18, ADCH


    STS OCR1A, r16
    STS OCR2A, r17
    STS OCR1B, r17
    STS OCR2B, r16
    RET


moveForward:
    LDS r16, ADCL
    LDS r18, ADCH
    LDI r17, 0


    STS OCR1A, r16
    STS OCR2A, r16
    STS OCR1B, r17
    STS OCR2B, r17
    RET
```

```
moveBackward:
    LDI r16, 0
    LDS r17, ADCL
    LDS r18, ADCH


    STS OCR1A, r16
    STS OCR2A, r16
    STS OCR1B, r17
    STS OCR2B, r17
    RET


shoot:
    ; 3
    SBI PORTD - 0x20, 4
    SBI PORTD - 0x20, 5
    RCALL delay
    ; 2
    CBI PORTD - 0x20, 4
    SBI PORTD - 0x20, 5
    RCALL delay
    ; 1
    SBI PORTD - 0x20, 4
    CBI PORTD - 0x20, 5
    RCALL delay
    ; fire the bomb
    SBI PORTB - 0x20, 0
    RCALL delay
    CBI PORTB - 0x20, 0
    RET


stop:
    ; stop the robot movement
    LDI r16, 0


    STS OCR1A, r16
    STS OCR1B, r16
    STS OCR2A, r16
    STS OCR2B, r16
    RET


delay:
    ; custom delay for 1 second
    ldi  r18, 82
    ldi  r19, 43
    ldi  r20, 100
temp:
    dec  r20
    brne temp
    dec  r19
    brne temp
    dec  r18
    brne temp
    nop
    RET
```

Discussion & Conclusion:



The concept in this project is to control the circuit, motors by a software that built by assembly language which to send a command by an app to move our robot in any direction. Furthermore, this app can send other command to run the circuit which display count down time on 7 segment to run the buzzer.

This project help us to learn more about DLD and MicroP which let us to learn how to combine two different circuits in one to construct a robot that can fellow the orders by the user.