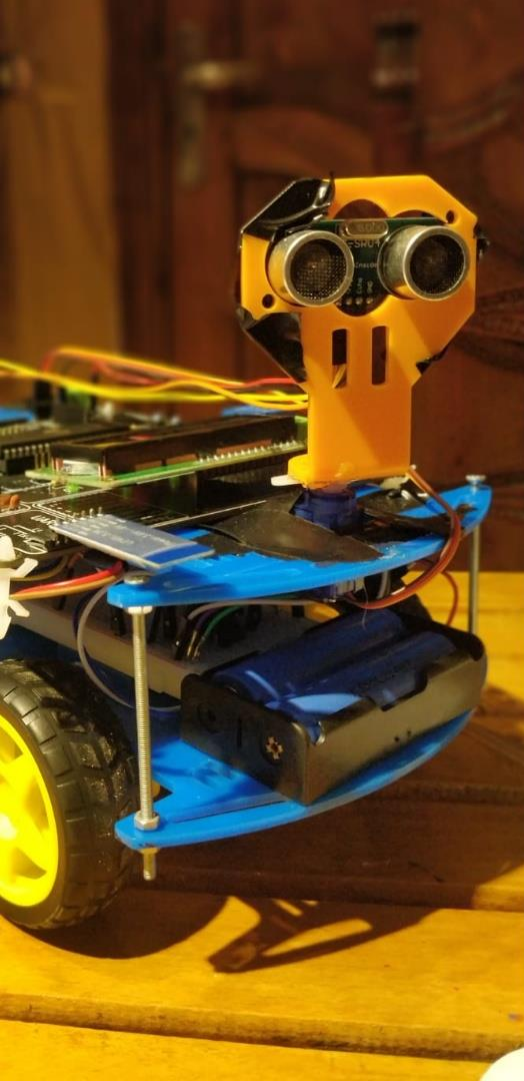


GRADUATION PROJECT

Autonomous Car

WHAT IS AN AUTONOMOUS CAR?



It is a vehicle that is capable of sensing its environment and operating without human involvement.

It senses using an ultrasonic sensor rotating with the help of a servo with range of 180° .

When the Atmega32 chip receives a signal that says that there is an obstacle, it sends orders to the servo to rotate left and right.

Then the sensor checks if there is an obstacle and the while (1) keep going.

COMPONENTS

- **Ultrasonic sensor**

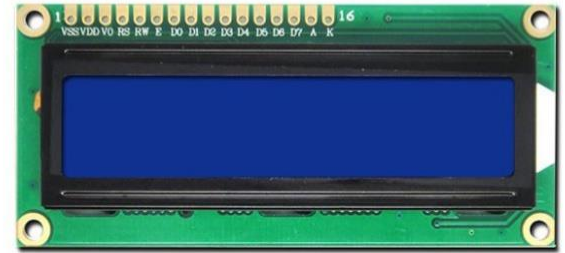
To detect any obstacles towards to the car.

- **Servo**

Used to make ultrasonic scanning around 180°.

- **LCD**

To display the current direction that car immediately move.



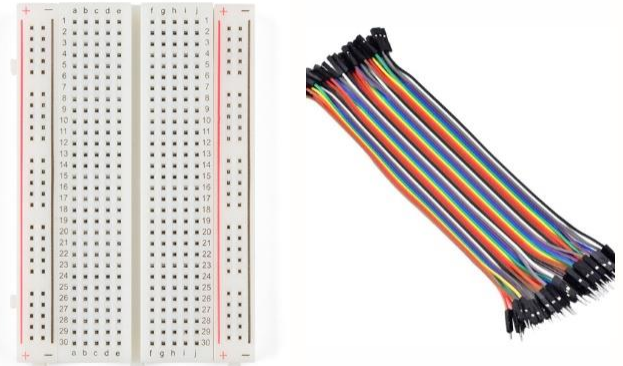
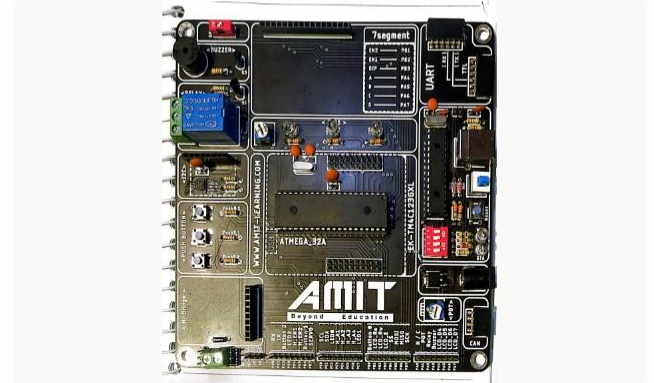
COMPONENTS

- DC Motor ×4
- L293D Chip
To control the speed and the direction of motors.
- Robot Platform 4WD (4 Gear Motors + 4 Wheels)



COMPONENTS

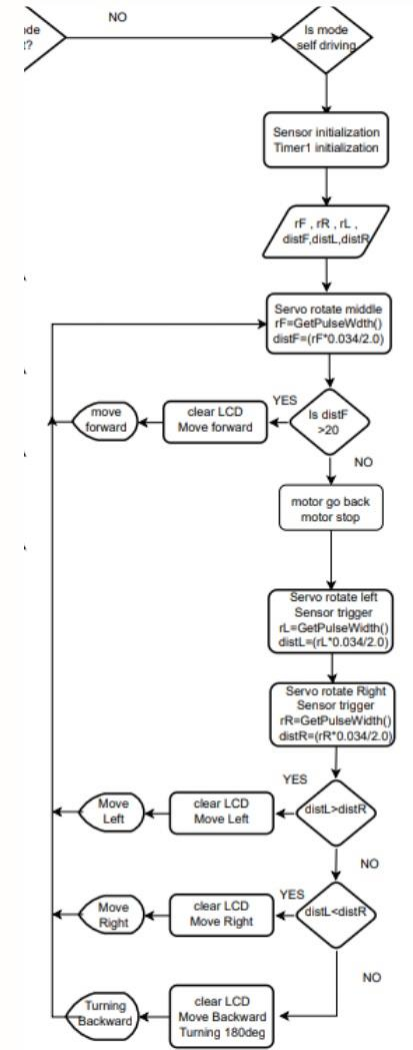
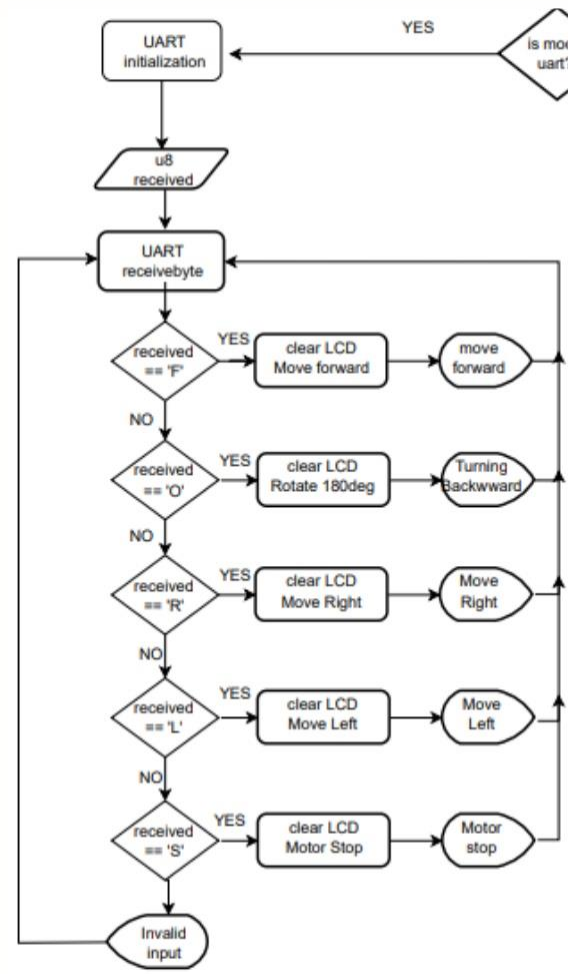
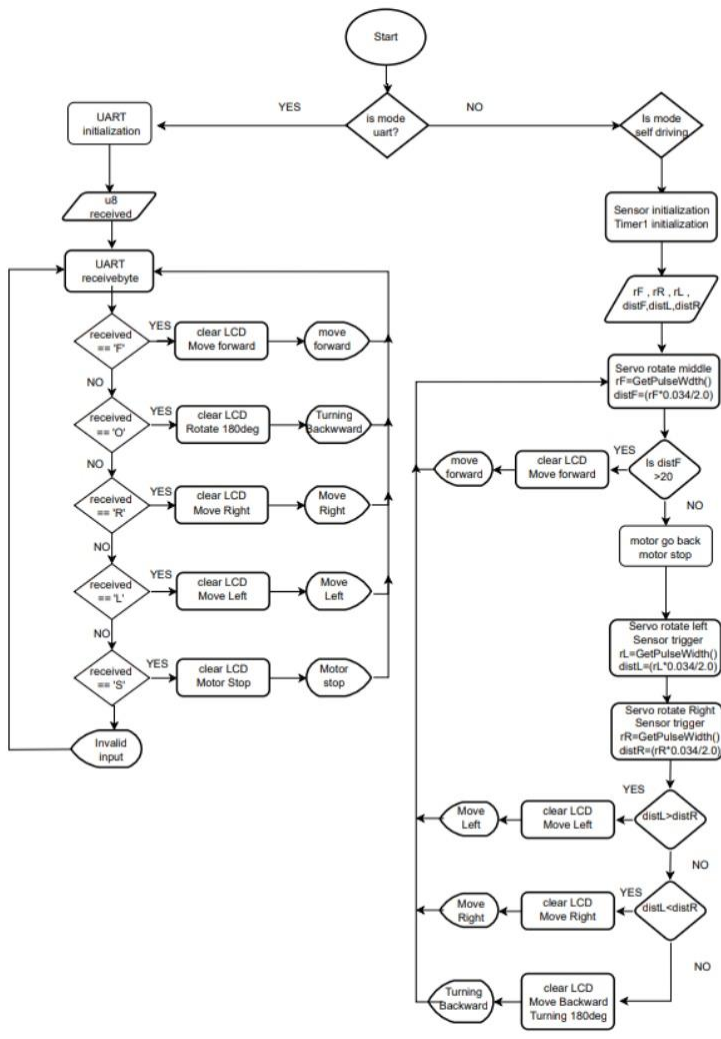
- AMIT Kit
- Breadboard
- Jumpers



COMPONENTS

- Battery (16850) ×4
- Battery holder ×2
- Power Supply (Power Bank)





ULTRASONIC SENSOR FUNCTIONS:

```
void HCSR04Init()
{
    // we're setting the trigger pin as output as it will generate ultrasonic sound wave
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN0,DIO_PIN_OUTPUT);    //TRIG
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN1,DIO_PIN_INPUT);     //ECHO
}

void HCSR04Trigger()
{
    // this function will generate ultrasonic sound wave for 15 microseconds
    //Send a 10uS pulse on trigger line

    DIO_SetPinValue(DIO_PORTC,DIO_PIN0,DIO_PIN_HIGH);    //high

    _delay_us(10);    //wait 15uS

    DIO_SetPinValue(DIO_PORTC,DIO_PIN0,DIO_PIN_LOW);    //low
}
```


ULTRASONIC SENSOR FUNCTIONS:

```
u16 GetPulseWidth()
{
    // this function will be used to measure the pulse duration. When the ultra sound echo back after hitting an object
    // the microcontroller will read the pulse using the echo pin of the ultrasonic sensor connected to it.

    u32 i;
    u16 result;

    //checking if the ultrasonic is working or not
    // it check the echo pin for a certain amount of time. If there is no signal it means the sensor is not working or not connect properly
    for(i=0;i<600000;i++)
    {
        if(!GET_BIT(PINC,1) )
            continue;    //Line is still low, so wait
        else
            break;        //High edge detected, so break.
    }

    if(i==600000)
        return US_ERROR;    //Indicates time out
}
```

ULTRASONIC SENSOR FUNCTIONS:

```
TMR_TIMER1_Start();

for(i=0;i<600000;i++)          // the 600000 value is used randomly to denote a very small amount of time, almost 40 milliseconds
{
    if(GET_BIT(PINC,1))
    {
        if(TCNT1L_u16 > 60000) break; else continue; //if the TCNT1 value gets higher than 60000 it means there is not object
                                                    //in its range
    }
    else
        break;
}

if(i==600000)
return US_NO_OBSTACLE; //Indicates time out

//Falling edge found

result=TCNT1L_u16; // microcontroller stores the the value of the counted pulse time in the TCNT1 register. So, we're returning
// this value to the main function for utilizing it later

//Stop Timer
TMR_TIMER1_Stop();

if(result > 60000)
return US_NO_OBSTACLE; //No obstacle
else
return (result>>1);
}
```

SERVO FUNCTIONS:

```
void ServoRotation_RIGHT(void)
{
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_HIGH);
    _delay_us(550);
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_LOW);
    _delay_ms(20);
}
```

```
void ServoRotation_LEFT(void)
{
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_HIGH);
    _delay_us(2500);
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_LOW);
    _delay_ms(20);
}
```

```
void ServoRotation_MID(void)
{
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_HIGH);
    _delay_us(1550);
    DIO_SetPinValue(DIO_PORTD,DIO_PIN7,DIO_PIN_LOW);
    _delay_ms(20);
}
```

MOTORS FUNCTIONS:

Motors initialization

```
void Motor_PinsOutput(void)
{
    //motors pins
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN6,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN5,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN4,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTC,DIO_PIN3,DIO_PIN_OUTPUT);

    DIO_SetPinDirection(DIO_PORTB,DIO_PIN7,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTB,DIO_PIN6,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTB,DIO_PIN5,DIO_PIN_OUTPUT);
    DIO_SetPinDirection(DIO_PORTB,DIO_PIN4,DIO_PIN_OUTPUT);
}
```

MOTORS FUNCTIONS:

```
void move_forward(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_HIGH);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```


MOTORS FUNCTIONS:

```
void move_right(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_HIGH);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);

    _delay_ms(600);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```


MOTORS FUNCTIONS:

```
void move_left(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(600);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```

MOTORS FUNCTIONS:

It rotates 180 degree.

```
void move_backward(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(1050);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```

MOTORS FUNCTIONS:

It moves back.

```
void move_GoBack(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(500);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```

MOTORS FUNCTIONS:

It moves back.

```
void move_GoBack(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(500);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```


MOTORS FUNCTIONS:

It brakes and go back a little.

```
void motor_brake(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(200);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```

MOTORS FUNCTIONS:

It brakes and go back a little.

```
void motor_brake(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_HIGH);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_HIGH);

    _delay_ms(200);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```


MOTORS FUNCTIONS:

```
void motor_stop(void)
{
    DIO_SetPinValue(DIO_PORTC,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN4,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTC,DIO_PIN3,DIO_PIN_LOW);

    DIO_SetPinValue(DIO_PORTB,DIO_PIN7,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN6,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN5,DIO_PIN_LOW);
    DIO_SetPinValue(DIO_PORTB,DIO_PIN4,DIO_PIN_LOW);
}
```

TIMER1 FUNCTIONS:

Timer one is functioning in normal mode.

We are selecting prescaler = 8

```
void TMR_TIMER1Init(void)
{
    //normal mode
    TCCR1A=0X00;
}

void TMR_TIMER1_Start(void)
{
    //Select Prescalar value = 8
    SET_BIT(TCCR1B,1);
    CLR_BIT(TCCR1B,0);
    CLR_BIT(TCCR1B,2);
    TCNT1L_u16=0x00;
}

void TMR_TIMER1_Stop(void)
{
    //STOP TIMER0
    CLR_BIT(TCCR1B,0);
    CLR_BIT(TCCR1B,1);
    CLR_BIT(TCCR1B,2);
}
```

Thank you!

Project by:

Michael Mina

Marten Wafik

Rafik Sameh