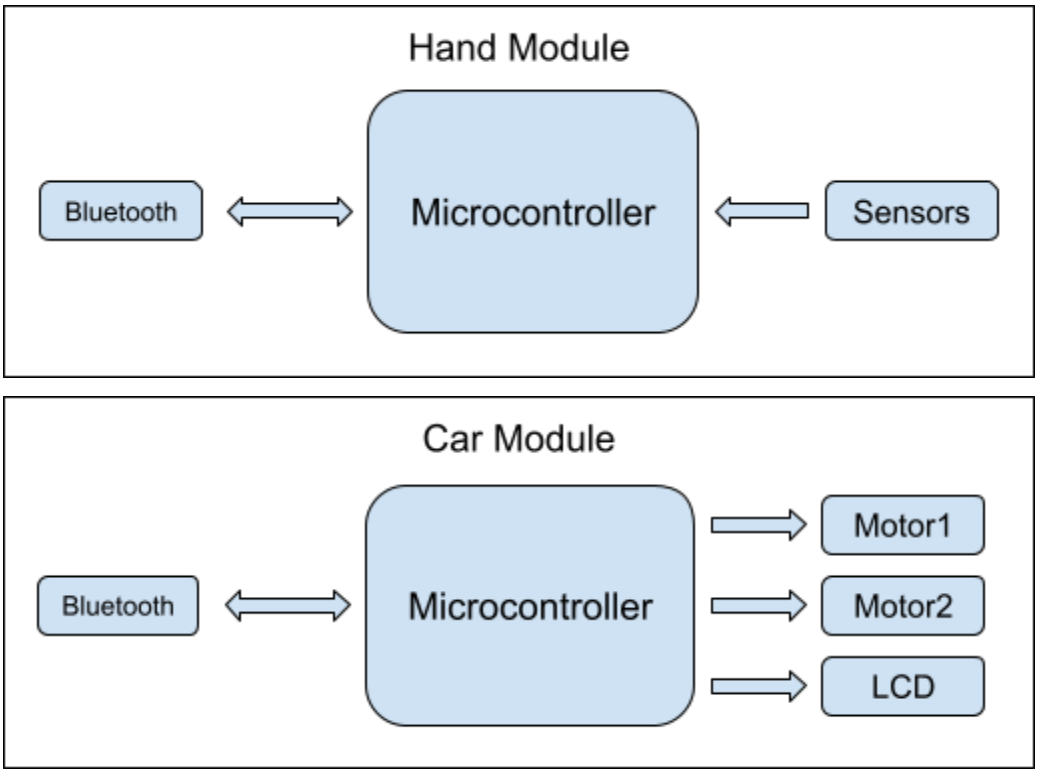
Name: Yasser Waleed
Date: 31/12/2022

Embedded Systems Diploma
Graduation Project

CAR CONTROL

Project Requirements

This project aims to control a car's speed and direction using hand's motion or fingers' motion. There are two modules required: Hand Module and Car Module. The Hand Module consists of a microcontroller, flexible sensors, and bluetooth. The Car Module consists of a microcontroller, DC motors, LCD, and bluetooth.

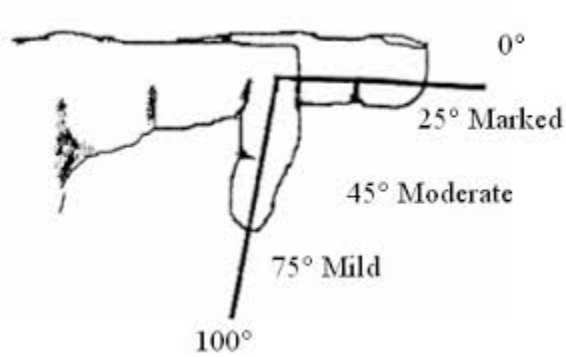


The Hand Module should detect the fingers posture and continuously send updated data to the car's module. The Car Module on the other hand has to receive the updated data and quickly adjust its motion. The fingers' posture should translate into the car's speed and direction.

(Note: Since this project will be demonstrated as a simulation on Proteus, a potentiometer could be used instead of flexible sensors.)

Analysis

To achieve the required functionality, a good approach is dividing the range of the flexible sensors so that the central value represents the neutral position of the finger which signals stopping. For example, if the finger has a range of 100 degrees, having the finger around 50 degrees should signal stopping. It is inconvenient to define the stop signal as exactly 50, so it could be the range from 40 to 60 degrees, or 45 to 55 degrees for instance.



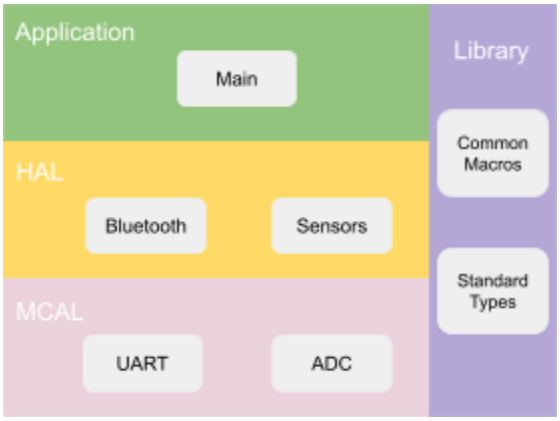
If the finger is at an angle lower than the neutral position (below 40 degrees), it should signal a motor moving forward, and if the finger is at an angle higher than the neutral position (above 60 degrees), it should signal a motor moving backward. Since there are two motors, two fingers can be used to control each motor. The speed could be controlled by a third finger, or it could be using the same two fingers.

If a third finger is used, then the first range for that finger should mean zero speed, then the higher the angle, the higher the speed. However, if the same two fingers are to be used, then the speed will depend on how far the finger is from the neutral position. So, below 40 means moving forward, but the nearer it is to zero, the higher the speed. And, above 60 is moving backwards, but the nearer it is to 100, the higher the speed.

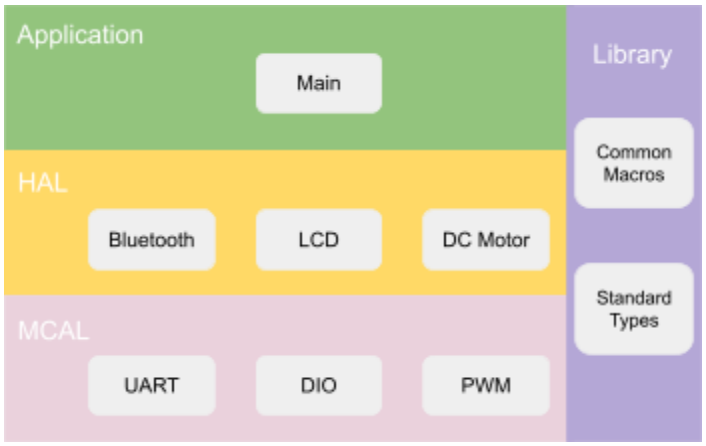
Name: Yasser Waleed
Date: 31/12/2022

Software Design

Shown below is the layered architecture for the Hand Module. The main program uses the sensors module to receive current values from the sensors, and uses the bluetooth module to send data to the Car Module. The bluetooth module needs the UART module to send and receive data. The sensors module needs the DIO module to read data from the input pins.



Shown below is the layered architecture for the Car Module. The main program uses the bluetooth module to receive data from the Hand Module, uses the DC motor module to control the two motors depending on the received data, and uses LCD module to display the current state. The bluetooth module needs the UART module to send and receive data. The LCD module needs the DIO module to display data by writing to the output pins. The DC motor module needs the PWM module to control the speed.

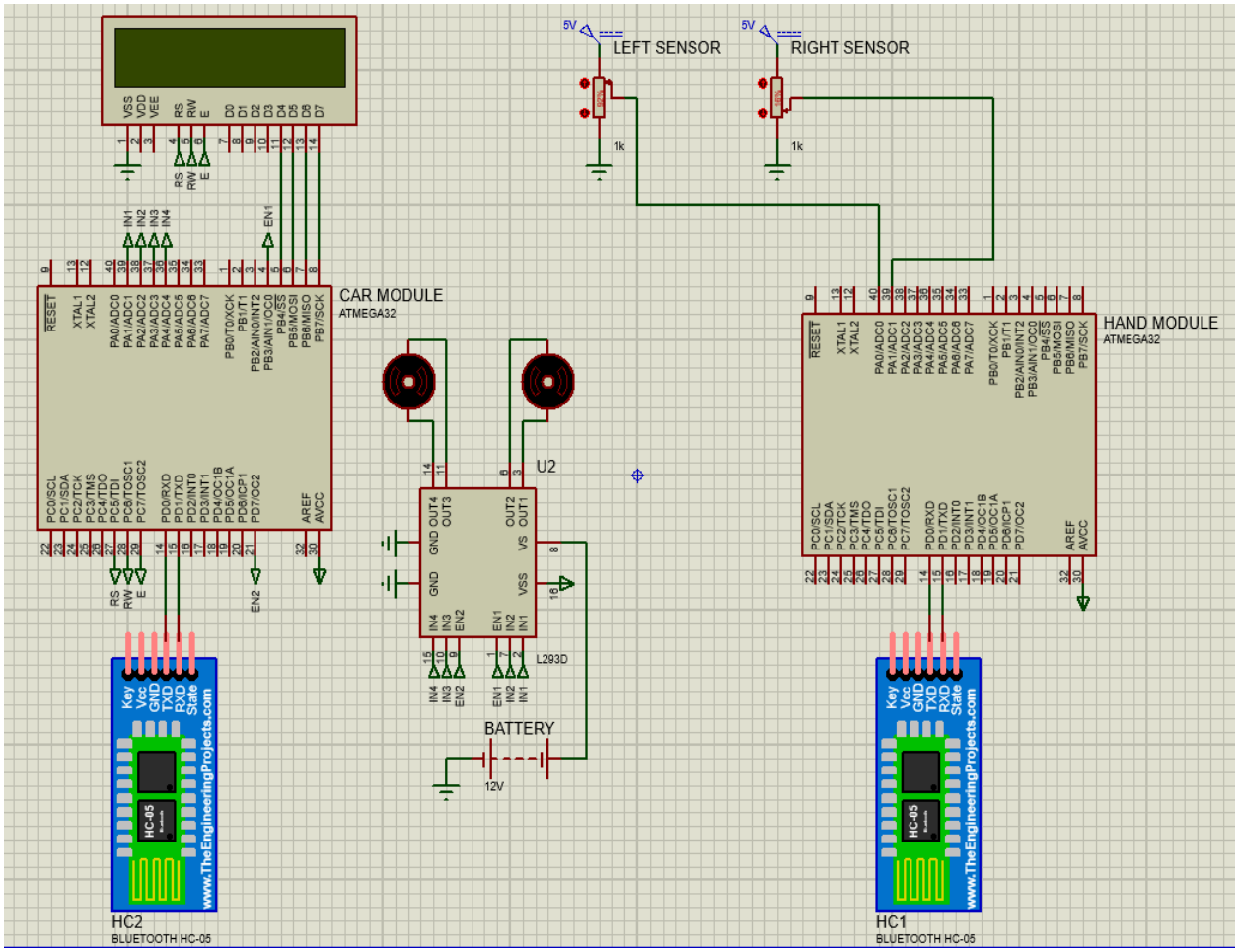


Name: Yasser Waleed
Date: 31/12/2022

Components

- Microcontrollers (ATMEGA##)
- Bluetooth Modules (HC-05)
- Flexible sensors
- Batteries
- L293D
- LCD (16x2)
- Motors
- Wires
- Caster wheel
- Chassis

Electrical Circuit

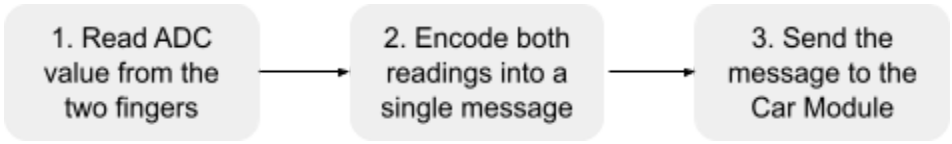


Shown above is the circuit design on Proteus. According to Bluetooth Library for Proteus - The Engineering Projects, the HC-05 modules shown do not accept AT Commands. Thus, it is only shown above to illustrate the circuit's connections. Normally, one of the bluetooth modules should be configured as master, and the other is configured as slave. This is explained in this link: How To Configure and Pair Two HC-05 Bluetooth Modules as Master and Slave | AT Commands.

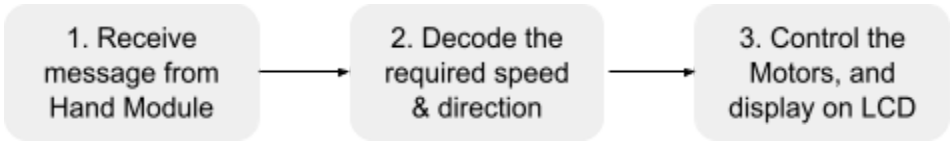
However, for the purpose of this project's simulation, the communication between the two microcontrollers will be through the normal UART connection.

Implementations

The flowchart for the Hand Module is as follows:



The flowchart for the Car Module is as follows:



==========

The main loop for the Hand Module is as follows:

```
while (1)
{
    /*Reading ADC input, scaled by 0.4 to fit range of 100 degrees*/
    left_finger = ADC_Read(0)*0.4;
    right_finger = ADC_Read(1)*0.4;
    message = 0;
```

```
        /*The first digit represents the right finger*/
        if                              (right_finger <= 10) {message += 1;}
        else if (10 < right_finger && right_finger <= 20) {message += 2;}
        else if (20 < right_finger && right_finger <= 30) {message += 3;}
        else if (30 < right_finger && right_finger <= 40) {message += 4;}
        else if (40 < right_finger && right_finger <= 60) {message += 5;}
        else if (60 < right_finger && right_finger <= 70) {message += 6;}
        else if (70 < right_finger && right_finger <= 80) {message += 7;}
        else if (80 < right_finger && right_finger <= 90) {message += 8;}
        else if (90 < right_finger)                       {message += 9;}
        /*The second digit represents the left finger*/
        if                             (left_finger <= 10) {message += 10;}
        else if (10 < left_finger && left_finger <= 20) {message += 20;}
        else if (20 < left_finger && left_finger <= 30) {message += 30;}
        else if (30 < left_finger && left_finger <= 40) {message += 40;}
        else if (40 < left_finger && left_finger <= 60) {message += 50;}
        else if (60 < left_finger && left_finger <= 70) {message += 60;}
        else if (70 < left_finger && left_finger <= 80) {message += 70;}
        else if (80 < left_finger && left_finger <= 90) {message += 80;}
        else if (90 < left_finger)                      {message += 90;}

        /* Transmit Message */
        UART_TransmitChar(message);
        _delay_ms(100);
    }
}
```

The loop goes through the 3 main steps illustrated in the flowchart. ADC reading is scaled by multiplying by 0.4. This way we can match the angle range mentioned in the analysis section. The message is decoded into a single value. Could be as little as 11 and as large as 99. Each digit represents a finger. The message is sent via UART. From this message, the Car Module will be able to decide the required speed and direction for each of the motors.

==========

The main loop for the Car Module is as follows:

```
while (1)
{
Message = UART_ReceiveChar();
R_Finger = Message % 10;
L_Finger = (Message - R_Finger) / 10;

if      (R_Finger == 1) { R_Speed = -4; DCMotor_SetRotation(R_MOTOR, -100);}
else if (R_Finger == 2){ R_Speed = -3; DCMotor_SetRotation(R_MOTOR, -75);}
else if (R_Finger == 3){ R_Speed = -2; DCMotor_SetRotation(R_MOTOR, -50);}
else if (R_Finger == 4){ R_Speed = -1; DCMotor_SetRotation(R_MOTOR, -25);}
else if (R_Finger == 5){ R_Speed =  0; DCMotor_SetRotation(R_MOTOR,  0  );}
else if (R_Finger == 6){ R_Speed =  1; DCMotor_SetRotation(R_MOTOR,  25);}
else if (R_Finger == 7){ R_Speed =  2; DCMotor_SetRotation(R_MOTOR,  50);}
else if (R_Finger == 8){ R_Speed =  3; DCMotor_SetRotation(R_MOTOR,  75);}
else if (R_Finger == 9){ R_Speed =  4; DCMotor_SetRotation(R_MOTOR,100);}

if      (L_Finger == 1) { L_Speed = -4; DCMotor_SetRotation(L_MOTOR, -100);}
else if (L_Finger == 2){ L_Speed = -3; DCMotor_SetRotation(L_MOTOR, -75);}
else if (L_Finger == 3){ L_Speed = -2; DCMotor_SetRotation(L_MOTOR, -50);}
else if (L_Finger == 4){ L_Speed = -1; DCMotor_SetRotation(L_MOTOR, -25);}
else if (L_Finger == 5){ L_Speed =  0; DCMotor_SetRotation(L_MOTOR,  0  );}
else if (L_Finger == 6){ L_Speed =  1; DCMotor_SetRotation(L_MOTOR,  25);}
else if (L_Finger == 7){ L_Speed =  2; DCMotor_SetRotation(L_MOTOR,  50);}
else if (L_Finger == 8){ L_Speed =  3; DCMotor_SetRotation(L_MOTOR,  75);}
else if (L_Finger == 9){ L_Speed =  4; DCMotor_SetRotation(L_MOTOR,100);}

sprintf(display, "R:%d ", R_Speed);
lcd_gotoxy(0,0);
lcd_puts(display);
sprintf(display, "L:%d ", L_Speed);
lcd_gotoxy(0,1);
lcd_puts(display);
_delay_ms(50);
}
```

References:

- https://www.theengineeringprojects.com/2016/03/bluetooth-library-for-proteus.html
- https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf
- Arduino Bluetooth Car using HC05 Master-Slave Transmitter Receive…
- Direction and speed control of DC motor using ATmega32