Automated Car Avoidance System

By: Jacob Reyes & Ricardo Rodriguez

Overview

The Automated Car Avoidance System, is a obstacle detection / avoidance system that uses a microcontroller in conjunction with ultrasonic sensors in order to detect and avoid obstacles.

Project objective / Goals

The main objective of this project is to design and develop a obstacle detection / avoidance system that can be installed on small vehicles making them autonomous.

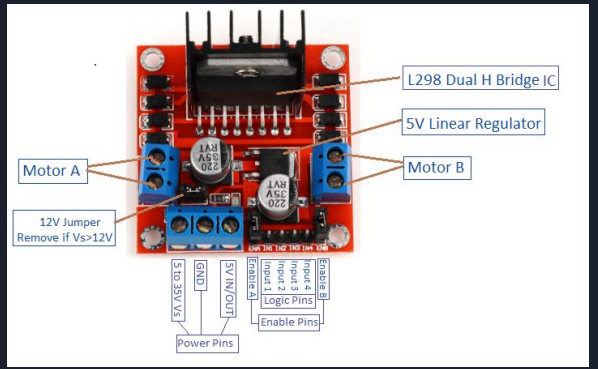
This project will allow us to gain a better understanding of programming a microcontroller to interface with ultrasonic sensors as well as DC motors.

Implementation

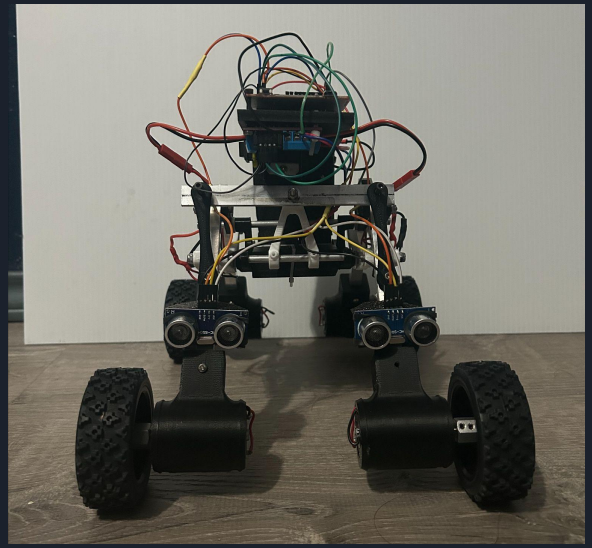
- Using a microcontroller and ultrasonic sensors is a cost efficient way of creating a obstacle detection and avoidance system.
- State of the art autonomous vehicles use expensive sensors, cameras, and artificial intelligence software to function.
- Our obstacle detection and avoidance system can be used at a small scale to automate small vehicles in predictable environments.
- Applications for small autonomous vehicles include, delivery, transportation, maintenance, security / surveillance, and even entertainment.

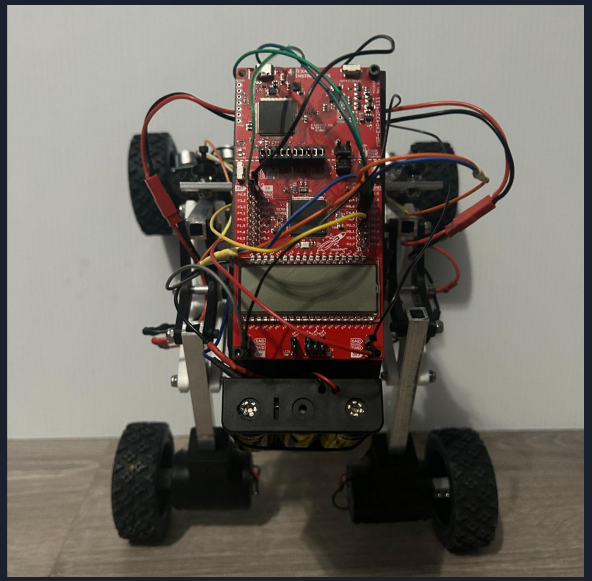
Components of Automated Car Avoidance System

- TI MSP-EXP432P4111 - Microcontroller
- HC-SR04 Ultrasonic Sensors (x2)
- L298N H-Bridge Motor Driver
- 12-V Battery Pack equipped with an On/Off switch
- T Star DSTR robot

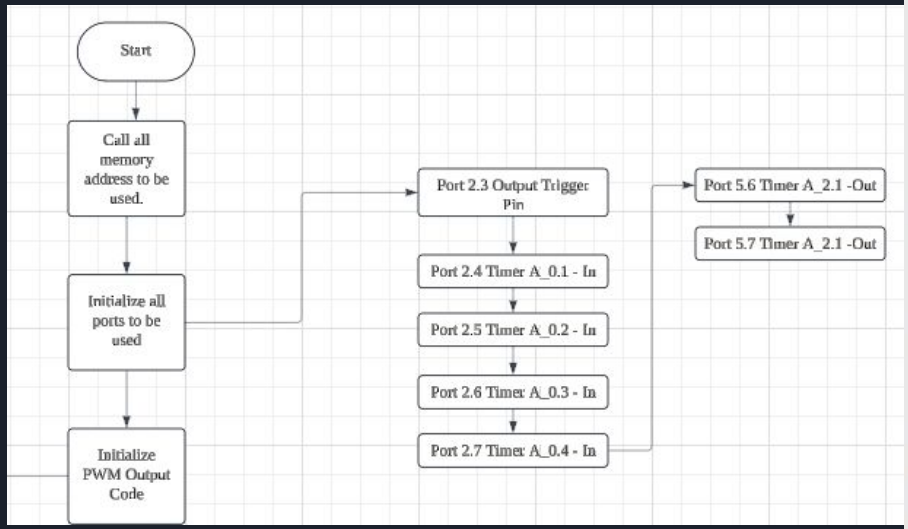


Completed Vehicle





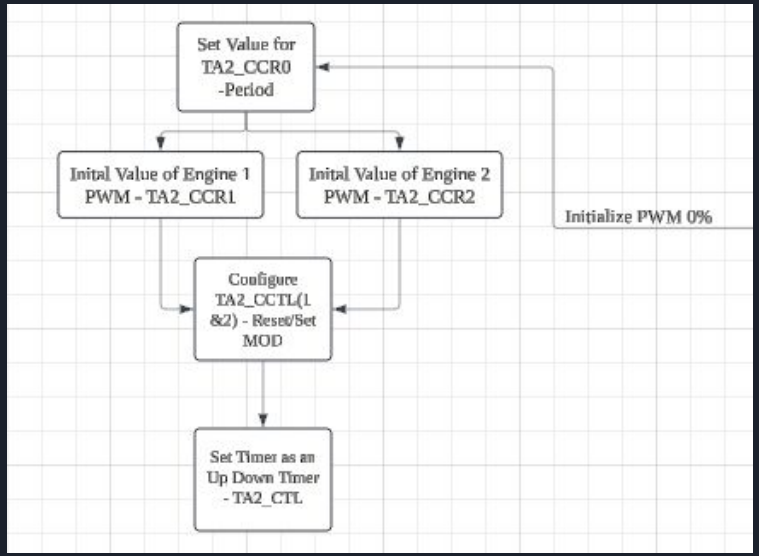
Schematics Done + Code Alignment -Software Portion



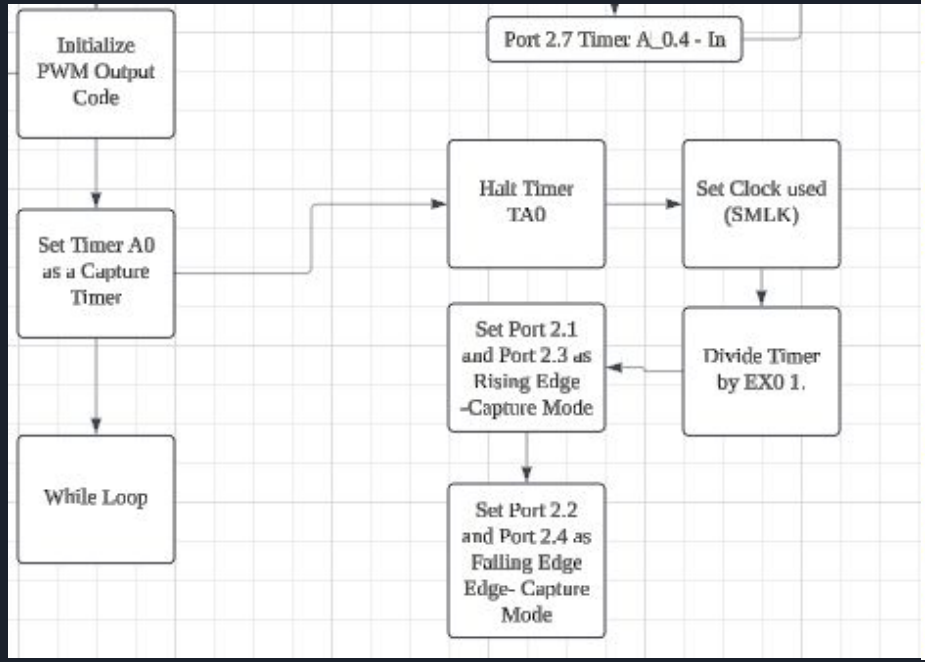

```
1   P2SEL0  EQU 0x40004C0B      ;Port 2 Select 0
2   P2SEL1  EQU 0x40004C0D      ;Port 2 Select 1
3   P2DIR   EQU 0x40004C05      ;Port 2 Direction
4   P2_OUT  EQU 0x40004C03      ;Port 2 Output
5   TA0_CTL EQU 0x40000000      ;Timer A_0 Control
6   TA0_CCR1 EQU 0x40000014     ;Timer A_0 Read P2.4
7   TA0_CCTL1 EQU 0x40000004    ;Timer A_0 Control P2.4
8   TA0_CCR2 EQU 0x40000016     ;Timer A_0 Read P2.5
9   TA0_CCTL2 EQU 0x40000006    ;Timer A_0 Control P2.5
10  TA0_CCR3 EQU 0x40000018     ;Timer A_0 Read P2.6
11  TA0_CCTL3 EQU 0x40000008    ;Timer A_0 Control P2.6
12  TA0_CCR4 EQU 0x4000001A     ;Timer A_0 Read P2.7
13  TA0_CCTL4 EQU 0x4000000A    ;Timer A_0 Control P2.7
14  TA0_EX0   EQU 0x40000020    ;Timer A_0 EX0
15  P5SEL0  EQU 0x40004C4A      ;Port 5 Select 0
16  P5SEL1  EQU 0x40004C4C      ;Port 5 Select 1
17  P5DIR   EQU 0x40004C44      ;Port 5 Direction
18  TA2_CTL EQU 0x40000800      ;Timer A_2 Control
19  TA2_CCR0 EQU 0x40000812     ;Timer A_2 Read - Period
20  TA2_CCTL1 EQU 0x40000804    ;Timer A_2 Control P5.6 - Left PWM
21  TA2_CCR1 EQU 0x40000814     ;Timer A_2 Read P5.6 - Left PWM
22  TA2_CCTL2 EQU 0x40000806    ;Timer A_2 Control P5.7 - Right PWM
23  TA2_CCR2 EQU 0x40000816     ;Timer A_2 Read P5.7 - Right PWM
24  STACK_TOP EQU 0x20000004    ;Stack Location
```

```
; Select 1 and Select 0 Port 5.6 & 5.7 PWM Output Signal
LDR     R0, =P5SEL1
LDRB    R1, [R0]            ;Select 1 = 0
MVN     R2, #0xC0           ; 5.6 & 5.7
AND     R3, R1, R2
STRB    R3, [R0]

;P5 Select_0
LDR     R0, =P5SEL0         ;Select 0 = 1
LDRB    R1, [R0]            ;5.6 & 5.7
ORR     R1, #0xC0
STRB    R1, [R0]

;P5 Direction
LDR     R0, =P5DIR      ;Set Timers at 5.6 and 5.7 as Outputs
LDRB    R1, [R0]
ORR     R1, #0xC0
STRB    R1, [R0]
```

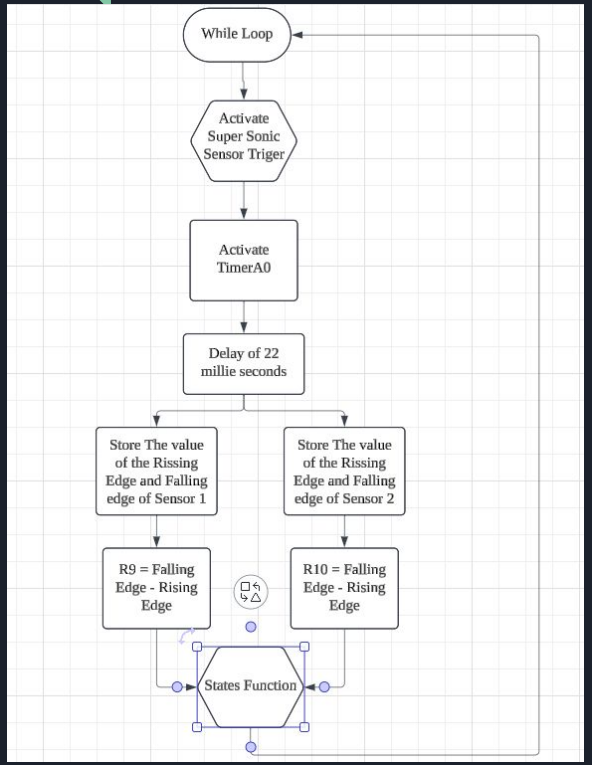
```
; Select 1 and Select 0 Port P2.4 & 2.5 & 2.6 & 2.7 PWM INPUT Signal
LDR     R0, =P2SEL1
LDRB    R1, [R0]
MVN     R2, #0xF0
AND     R3, R1, R2      ; P2.4 & 2.5 & 2.6 & 2.7 Pins as Input...
STRB    R3, [R0]        ;PWM signal

;Port 2 Select_0
LDR     R0, =P2SEL0
LDRB    R1, [R0]
ORR     R1, #0xF0       ; P2.4 & 2.5 & 2.6 & 2.7 Pins as Input...
STRB    R1, [R0]        ;PWM signal

;P2 Direction - INPUTS
LDR     R0, =P2DIR
LDRB    R1, [R0]        ;Set 6.6 & 6.7 as a Input
MVN     R2, #0xF0
AND     R3, R1,R2
STRB    R3, [R0]

;Set Port 2.3 as OUTPUTS For Activation Sensor Triger
LDR     R0, =P2DIR
LDRB    R1, [R0]
ORR     R1, #0x08
STRB    R1, [R0]
```



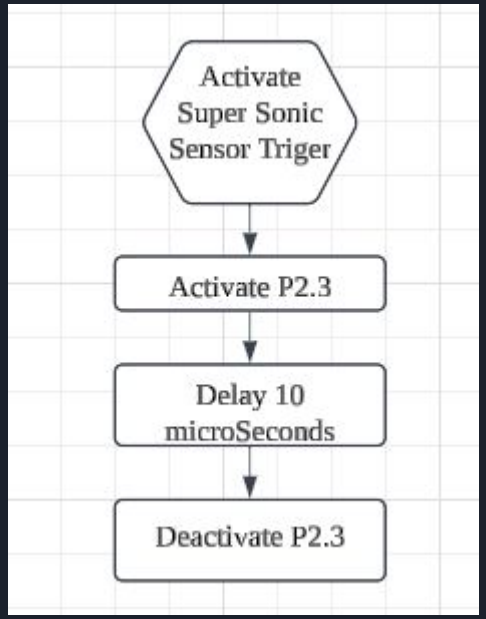

```
LDR     R11, =0x00000000   ;5.6 Port Initial PWM
LDR     R12, =0x00000000    ;5.7 Port Initial PWM

LDR     R0, =TA2_CCR0
LDR     R1, =0x0000EA5F     ;SETTING THE VALUE FOR THE TOTAL 20MS -50Hz Period
STRH    R1, [R0]
LDRH    R2, [R0]

LDR     R0, =TA2_CCR1       ;SETTING THE INITIAL PWM VALUE FOR LEFT Motor
STRH    R11, [R0]
LDRH    R2, [R0]

LDR     R0, =TA2_CCR2       ;SETTING THE INITIAL PWM VALUE FOR RIGHT Motor
STRH    R12, [R0]
LDRH    R2, [R0]

LDR     R0, =TA2_CCTL1      ;CONFIGURING AS A RESET/SET - Port 5.6
LDR     R1, =0x000000E0
STRH    R1, [R0]
LDRH    R2, [R0]

LDR     R0, =TA2_CCTL2      ;CONFIGURING AS A RESET/SET - Port 5.7
LDR     R1, =0x000000E0
STRH    R1, [R0]
LDRH    R2, [R0]

LDR     R0, =TA2_CTL
LDR     R1, =0x00000214  ; SETTING THE UP-DOWN TIMER - Infinitly
STRH    R1, [R0]
```



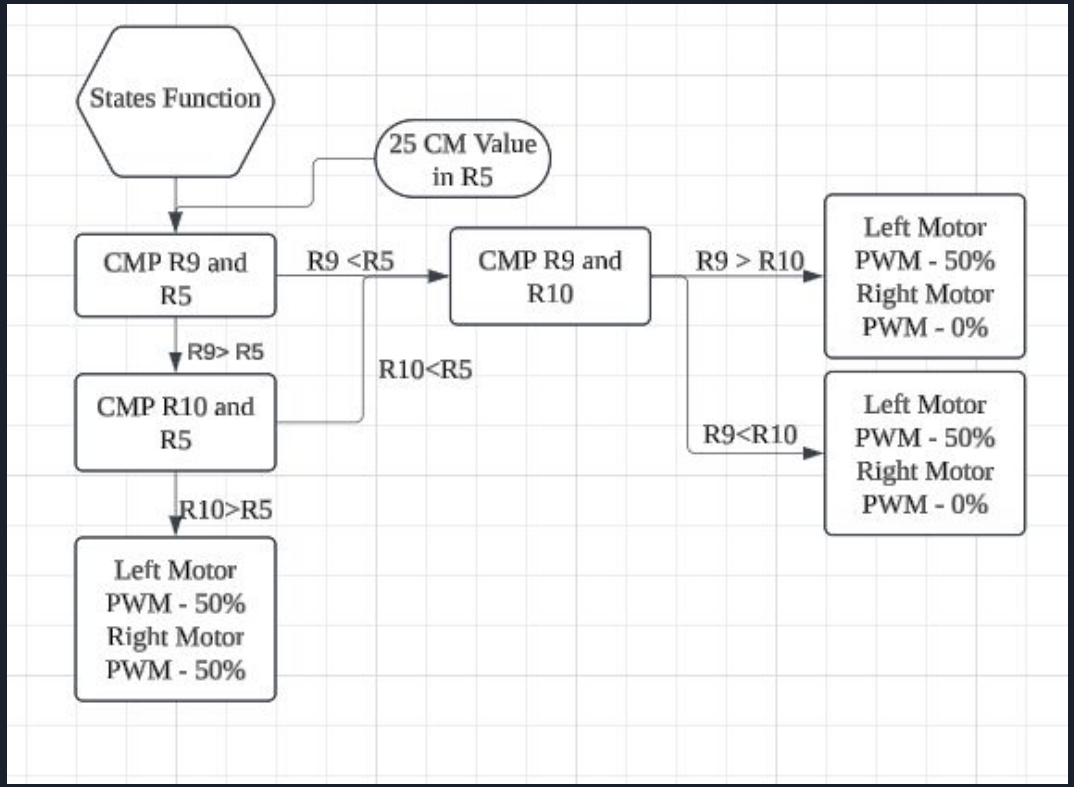

```
; Port P2.4 & 2.5 & 2.6 & 2.7 as PWM CAPTURE MODE
LDR     R0, =TA0_CTL        ;Halt Timer to Configure
LDR     R1, =0x00000030
MVN     R2, R1
LDRH    R3, [R0]
AND     R4, R3, R2
STRH    R4, [R0]

LDR     R0, =TA0_CTL        ; Set Clock :SMCLK, ID =/1
LDR     R1, =0x00000200
STRH    R1, [R0]


LDR     R0, =TA0_EX0        ; INDEX = /1
LDR     R1, =0x00000007
MVN     R2, R1
LDRH    R3, [R0]
AND     R4, R3, R2
STRH    R4, [R0]


LDR     R0, =TA0_CCTL1      ; Rising Edge , CCI2A, SCS, Capture Mode
LDR     R1, =0x00004900     ;P2.4
STRH    R1, [R0]

LDR     R0, =TA0_CCTL2      ;  Falling Edge, CCI2A, SCS, Capture Mode
LDR     R1, =0x00008900     ;P2.5
STRH    R1, [R0]

LDR     R0, =TA0_CCTL3      ; Rising Edge, CCI2A, SCS, Capture Mode
LDR     R1, =0x00004900     ;P2.6
STRH    R1, [R0]

LDR     R0, =TA0_CCTL4      ; Falling Edge, CCI2A, SCS, Capture Mode
LDR     R1, =0x00008900     ;P2.7
STRH    R1, [R0]
```




```
while
        LDR     R9, =TA0_CTL        ;Halt Timer
        LDR     R10, =0x00000024
        LDRH    R11, [R9]

        ; Sub-Routine Function
        BL      Activate_Super_Sonic_Sensor

                                    ;The whole Timer takes 21.845 millie seconds.
        ORR     R12, R10, R11
        STRH    R12,[R9]            ;Activates The CLOCK


        MOV     R0, #22         ; 22 Second Delay
        BL      delayMs

        LDR     R0, =TA0_CCR3       ;Strore The RISING EDGE Capture in R7
        LDRH    R7, [R0]
        LDR     R1, =TA0_CCR4       ;Stores the Falling EDGE CApture in R5
        LDRH    R5, [R1]
        SUB     R10, R5, R7         ;Value of the PWM Length By Subtracting...
                                        ;Falling Edge - Rising Edge

        LDR     R0, =TA0_CCR1       ;Strore The RISING EDGE Capture in R7
        LDRH    R7, [R0]
        LDR     R1, =TA0_CCR2       ;Stores the Falling EDGE CApture in R5
        LDRH    R6, [R1]
        SUB     R9, R6, R7          ;Value of the PWM Length By Subtracting...
                                        ;Falling Edge - Rising Edge

        ; Compare R6 and R7
        LDR     R5, =0x000010EC
        BL      States              ; Subroutine - All States Here

        B   while                   ; Loops Back
Here    B   Here
```




```
Activate_Super_Sonic_Sensor

        LDR     R0, =P2_OUT
        LDRB    R1, [R0]        ; Activate the Super-sonic Sensor Trigger
        ORR     R1, #0x08       ; P 2.3 OUTPUT - ON
        STRB    R1, [R0]

        MOV     R1, #29        ; do inner loop 29 times
L4_1    SUBS    R1, #1          ; inner loop
        BNE     L4_1            ;Basicaly Wait 10 MicroSeconds

        LDR     R0, =P2_OUT     ;Deactivate the Super-Sonic Sensor Trigger
        LDRB    R1, [R0]
        MVN     R2, #0x08       ; P 2.3 OUTPUT -OFF
        AND     R3, R2, R1
        STRB    R3, [R0]

        BX      LR
```


States Function
25 CM Value in R5
CMP R9 and R5
R9 <R5
CMP R9 and R10
R9 > R10
Left Motor PWM - 50% Right Motor PWM - 0%
R9> R5
R10<R5
CMP R10 and R5
R9<R10
Left Motor PWM - 50% Right Motor PWM - 0%
R10>R5
Left Motor PWM - 50% Right Motor PWM - 50%

```
States
        STR     LR,[R13]
        SUB     R13, R13, #4    ;Stack Pointer Activated
        ;PWM Subroutine uses R11 and R12
        ;Sensor 1 uses R10 and Sensor 2 uses R9
        LDR     R4, =0x000006D5     ; 15CM
        LDR     R5, =0x00000A3F     ; 40CM


        CMP     R10,R5          ;Compare Sensor 1 if grater than 25 CM
        BHS     L1              ; Will Branch if SENSOR 1 is grater than 25 CM
        B       Check
L1      CMP     R9, R5
        BHS     L12             ; Will Branch if SENSOR 2 is grater than 25 CM
        B       Check


Check   CMP     R9, R4
        BHS     Less_Than_5CM
        LDR     R11, =0x00000000    ;PWM SIGNAL 0% - LEFT MOTOR
        LDR     R12, =0x00000000    ;PWM SIGNAL 0% - RIGHT MOTOR
        BL      PWM

Less_Than_5CM
        CMP     R10, R4
        BHS     Less_Than_5CM1
        LDR     R11, =0x00000000    ;PWM SIGNAL 0% - LEFT MOTOR
        LDR     R12, =0x00000000    ;PWM SIGNAL 0% - RIGHT MOTOR
        BL      PWM
        B       enp_loop

ess_Than_5CM1
        CMP     R10, R9
        BHS     LEFT            ; Will Branch if SENSOR 1 is grater than SENSOR 2
Right   LDR     R11, = 0x00000000   ;PWM SIGNAL 0% - LEFT MOTOR
        LDR     R12, = 0x00004650   ;PWM SIGNAL 30% - RIGHT MOTOR
        BL      PWM
        B       Leave_1
```

```
259  LEFT    CMP     R9, R10
260          BHS     Right
261          LDR     R11, = 0x00004650   ;PWM SIGNAL 30% - LEFT MOTOR
262          LDR     R12, = 0x00000000   ;PWM SIGNAL 0% - RIGHT MOTOR
263          BL      PWM
264          B   Leave_2
265
266
267  L12     LDR     R11, =0x00004650    ;PWM SIGNAL 30% - LEFT MOTOR
268          LDR     R12, =0x00004650    ;PWM SIGNAL 30% - RIGHT MOTOR
269          BL      PWM
270          B end_loop
271
272  enp_loop
273  Leave_1
274  Leave_2
275  end_loop
276  Less_Than_5CM1
277          ADD     R13, R13, #4            ;Loades the stack register info into the link register
278          LDR     LR,[R13]                ;Stack Pointer
279          BX  LR
```

Ultrasonic Sensor Calculation / PWM Calculation

$$Distance = \frac{1}{2} * 343(m/s) * \frac{1}{3MHz} * (\alpha)$$

Sensor Equation Used:

- ∝ = Decimal Value of Value Generated by Sensor
- Final Units: Meters

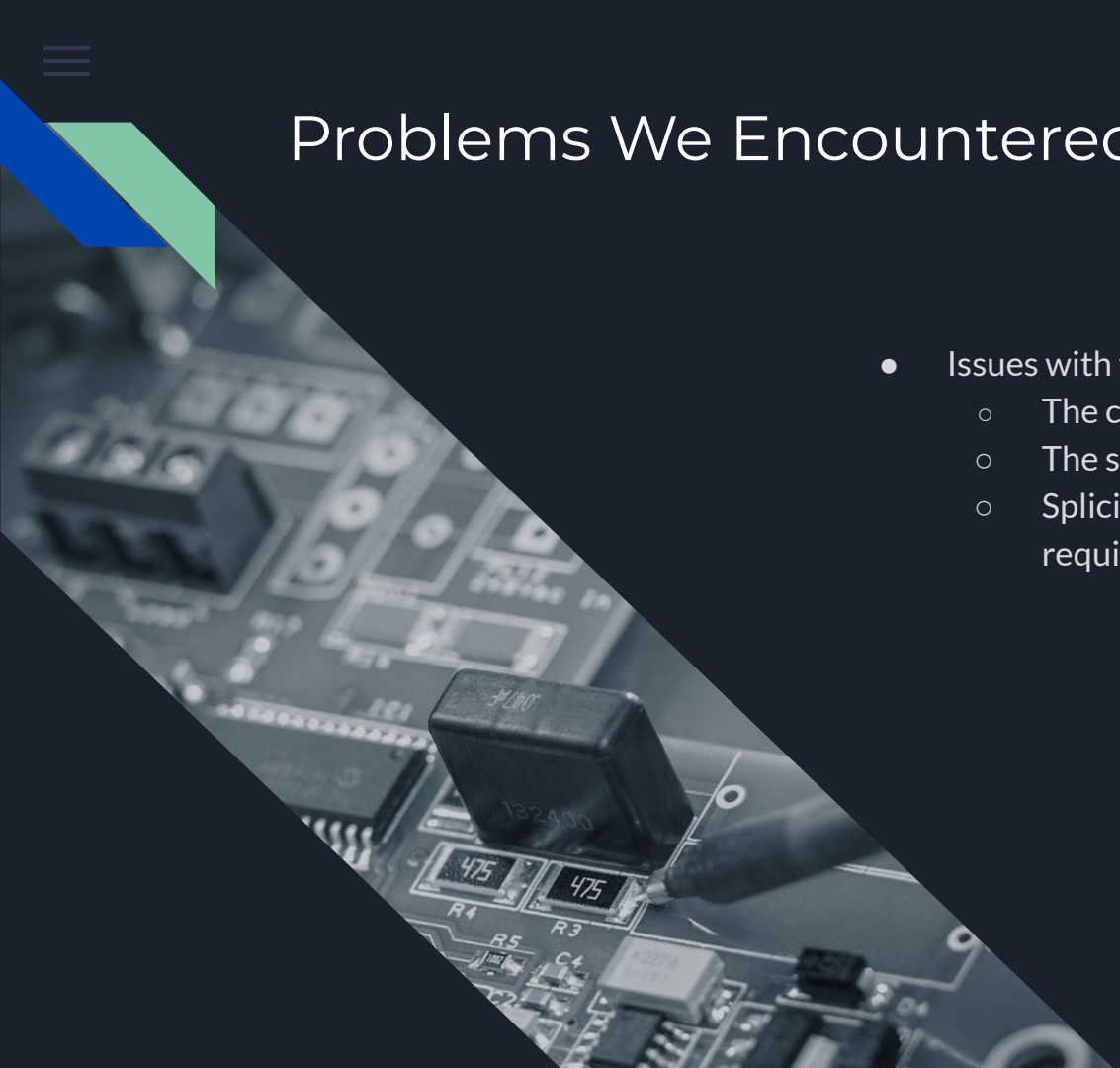
$$PWM\ Value = \frac{3MHz}{50Hz} * \beta$$

PWM Equation Used:

β = Percentage Wanted ex. .20 - 20%
Final Units: Unitless (Convert Decimal -> Hex)

Automated Car Avoidance System Demonstration

Problems We Encountered.



- Issues with the ultrasonic sensors
  - The capture mode not working correctly
  - The sensors detecting the vehicle itself
  - Splicing wires for each pin of the sensor was required

Future Improvements

Features that we would like to add / improve upon for future modifications would include:

- Adding a reverse feature, that would allow the vehicle to back up and reposition itself.
- Adjusting the detection distance of the vehicle according to its speed. Ex If the vehicle is traveling at a higher rate of speed the detection distance would increase giving the vehicle more time to react to the obstacle.
- Including a speaker on to the vehicle to announce its presence when encountering an obstacle.

Thank you!

Any Questions?