Complier directives

```
#include<c8051_SDCC.h>
#include <stdio.h>
#include<stdlib.h>
#include<i2c.h>
```

Function Prototypes

```
Void Port_Init(void);
Void Timer_Init(void);
Void Interrupt_Init(void);
Void Timer0_ISR(void) __interrupt 1;
void PCA_Init (void)
void read_driver(void)
void readcompass(void)
void readLED (void)
void drive_motar(void)
void steering servo(void)
void LEDblink(void)
```

global variables

```
Sbit LED0 SLDSW
unsigned int MOTOR_PW = 0;
unsigned int steering-servo_PW=0;
unsigned int LED brightness_PW=0;
unsigned int distance=0;
int heading=0;
unsigned int roombrightness=0;
unsigned int desired_speed=0;
unsigned int desired_heading=0;
unsigned int desired_brightness=0;
unsigned int current_speed=0;
```

```
        unsigned int current_heading=0;
        unsigned int current_brightness=0;
main function
        Declare local variables
                (none)
        Initialize function
        Sys_Init();
        putchar(‘ ‘); //the quotes in this line may not format correctly
        Port_Init();
        XBR0_Init();
         PCA_Init();
        Print some message to indicate start
        While(1){
                Ranger task()
                Compass task()
                LED task()
        }
        End main function




Ranger task()
         If switch is on
                read ranger distance
                If (distance <= 10cm)
                        Full speed ahead
                Else If (>=80cm)
                        Full reverse speed
```

Else

Speed linearly changes depends on the reading distance

Neutral at about 45cm

Else

Motor stop

End ranger task

Compass task

If switch is on

Read compass heading

Set a desired heading

Calculate error

Use a value k to calculate the pw then new pw wil turn the wheel

Else

Wheels are parallel to the car

End compass task

LED task

If switch is on

Read light sensors value

If (<40)

Brightest

Else If(>215)

Dimmest

Else

Change the brightness linearly

Else if switch is off

Turn off the led

End LED task

Other function

Initialize i2c and XBR

EVB Pin			Port Bit	Bit Addresses & Labels
1	2	1.		
		2.		
3	4	3.	3.3Volt	3.3Volts
		4.		
5	6	5.		
		6.		
7	8	7.		
		8.		
9	10	9.		
		10.	P1.2	Motor
11	12	11.	P1.3	LED
		12.	P1.0	Steering
13	14	13.		
		14.	P0.6	SDA
15	16	15.	P0.7	SCL
		16.		
17	18	17.		
		18.		
19	20	19.		
		20.		
21	22	21.		
		22.		
23	24	23.		
		24.		
25	26	25.		
		26.		
27	28	27.		
		28.		
29	30	29.		
		30.		
31	32	31.		
		32.		
33	34	33.		
		34.		
35	36	35.		
		36.		
37	38	37.		
		38.	P3.0	Slide Switch
39	40	39.		
		40.		
41 ⟷ 60				

Software Initializations

A) Port I/0

```
P3MDOUT &= ~0x01
P3 = 0x01
```

B) Timers

C) Interrupts

```
EA = 1
EIE1 |= 0x08
```

D) A/D

E) PCA

```
PCA0MD = 0x81
PCA0CPM1 = 0xC2
PCA0CN = 0x40
```

F) XBAR

```
XBR0 = 0x27
```

G) I2C

```
ENSMB = 1
SMB0CR = 0x93
```