File needed

#include <c8051\_SDCC.h>

#include <stdlib.h>// needed for abs function

#include <stdio.h>

#include <i2c.h>

8051 initialize functions

void Port\_Init(void);

void PCA\_Init (void);

void SMB\_Init (void);

void Interrupt\_Init(void); void PCA\_ISR ( void ) \_\_interrupt 9;

void read\_accel (void); //Sets global variables gx & gy

void set\_servo\_PWM (void);

void set\_drive\_PWM(void);

void updateLCD(void);

void set\_gains(void); // function which allow operator to set feedback gains

//define global variables

unsigned int PW\_CENTER = \_\_\_\_;

unsigned int PW\_RIGHT = \_\_\_\_;

unsigned int PW\_LEFT = \_\_\_\_;

unsigned int SERVO\_PW = \_\_\_\_;

unsigned int SERVO\_MAX= \_\_\_\_\_;

unsigned int SERVO\_MIN= \_\_\_\_\_;

unsigned int heading;

unsigned int range;

unsigned int light;

int compass\_adj = 0; // correction value from compass

int range\_adj = 0; // correction value from ranger

unsigned char r\_count; // overflow count for range

unsigned char h\_count; // overflow count for heading

unsigned char print\_count; // overflow count for printing

\_\_sbit \_\_at \_\_\_\_ RUN // a slide switch

\_\_sbit\_\_at \_\_\_\_ BILED0

\_\_sbit\_\_at \_\_\_\_ BILED1

Main function

Declare local variables

None

Funcion initialization

Do infinite while loop

Print battery voltage for check

if run out of battery

charge the battery

else

if (run switch is off)

Set the motor stop

Set the steer parallel to the car

BILED is red

Else if (run switch is on)

Set gain first (only once)

If (enough overflows to update accel)

read\_accels();

set\_servo\_PWM(); // set the servo PWM

set\_drive\_PWM(); // set drive PWM

new\_accels = 0; //set the flag off

a\_count = 0; //clear the accel counts

if (enough overflows to update LCD)

updateLCD(); // display values

new\_lcd = 0;

lcd\_count = 0;

finish the loop

end main function

void PCA\_ISR ( void ) \_\_interrupt 9 {

if (CF) {

CF = 0; // clear overflow indicator

a\_count++;

if(a\_count>=\_\_\_\_) {

new\_accel=1;

a\_count = 0;

}

lcd\_count++;

if (lcd\_count>=\_\_\_\_) {

new\_lcd = 1;

lcd\_count = 0;

}

PCA0 = PCA\_start;

} // handle other PCA interrupt sources

PCA0CN &= 0xC0;

}