/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* rc\_motor\_control.c

\*

\* This is a simple dc motor control program. It takes in a character

\* and then controls the motors to move forward, reverse, left or right

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <rc\_usefulincludes.h>

#include <roboticscape.h>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* int main()

\*

\* This main function contains these critical components

\* - call to initialize\_cape

\* - main while loop that checks for EXITING condition

\* - switch statement to send proper controls to the motors

\* - cleanup\_roboticscape() at the end

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(){

char input;

// always initialize cape library first

rc\_initialize();

printf("\nHello BeagleBone\n");

// done initializing so set state to RUNNING

rc\_set\_state(RUNNING);

// bring H-bridges of of standby

rc\_enable\_motors();

rc\_set\_led(GREEN,ON);

rc\_set\_led(RED,ON);

rc\_set\_motor\_free\_spin(1);

rc\_set\_motor\_free\_spin(2);

printf("Motors are now ready.\n");

// Turn on a raw terminal to get a single character

system("stty raw");

do

{

printf("> ");

input = getchar();

switch(input){

case 'f':

rc\_set\_motor(1, 0.5);

rc\_set\_motor(2, 0.5);

break;

case 'r':

rc\_set\_motor(1, 0.5);

rc\_set\_motor(2, -0.5);

break;

case 'l':

rc\_set\_motor(1, -0.5);

rc\_set\_motor(2, 0.5);

break;

case 'b':

rc\_set\_motor(1, -0.5);

rc\_set\_motor(2, -0.5);

break;

case 's':

rc\_set\_motor\_brake\_all();

break;

case 'q':

rc\_disable\_motors();

break;

default:

printf("Invalid Character.\n");

}

}

while(input != 'q');

printf("Done\n");

rc\_cleanup();

system("stty cooked");

return 0;

}

-------------------------------------------------------------------------------------------------------------------

if(rc\_get\_state()==RUNNING){

rc\_led\_set(RC\_LED\_GREEN, 1);

rc\_led\_set(RC\_LED\_RED, 0);

}

else{

rc\_led\_set(RC\_LED\_GREEN, 0);

rc\_led\_set(RC\_LED\_RED, 1);