```
/*
 * 6 Wheel Robot Experiment
 * --------------
  */

// Definition of variables and classes

#include <Servo.h>
Servo servo_camera_pan;
Servo servo_camera_tilt;
Servo servo_lights_pan;
int servo_camera_angle_center = 90;
int servo_camera_angle_min = 20;      //Left position of servo with wireless camera
int servo_camera_angle_max = 160;     //Right position, camera will pan between angles of 20 and 160 degrees.
int servo_camera_angle_circle = 300;  // Variable needed to calculate the movement of the servo back to its original position
int servo_camera_angle = 90;
boolean camera_toggle = true;
int servo_light_angle =90;
int servo_light_angle_max = 135;
int servo_light_angle_center = 90;
int servo_light_angle_min =45;          //Left postion of servo with LED
int servo_light_angle_circle = 225;     //Right position of servo with LED, servo will pan between angles defined by the variables ..._circle and... _min
boolean light_toggle_1 = true;
boolean light_toggle_2 = true;
#include <SoftwareSerial.h>
#define TXPIN 4                        //Assign pin 4 for communication with the motor controller
#define RXPIN 5                        //Assign pin 5 for communication with the motor controller
#define LIGHT_1 6
#define LIGHT_2 7
SoftwareSerial TREXJr(RXPIN, TXPIN);  // Motor Controller TRex Jr
int speed_min = 0;
unsigned char Send_Byte = 0;
int speed_turn = 115;              // fixed speed to turn left or right
int speed_fixed = 85;              // fixed speed
char bytein;                       // raw user input

void setup()              //Initial setup of motor controllers, servos and LEDs, definition of pins
{
  Serial.begin(19200);
  TREXJr.begin(19200);
  Serial.println("      Arduino Serial Motor Control");
  Serial.println();
  pinMode(RXPIN, INPUT);              //Pin 4 Arduino is input pin for signal from TREX Jr.
  pinMode(TXPIN, OUTPUT);              //Pin 5 Arduino is output pin for signal to TREX Jr.
  pinMode (LIGHT_1, OUTPUT);           // Pin 6 turn LED lights 1 on/off
  pinMode (LIGHT_2, OUTPUT);           //Pin 7 turn LED lights 2 on/off
  Send_Byte = 0xDA;                   //Addressing both motor controllers
  TREXJr.print(Send_Byte, BYTE);
  TREXJr.print(speed_min, BYTE);        //Setting speed for controller 1
  TREXJr.print(speed_min, BYTE);        //Setting speed for controller 2
  servo_camera_pan.attach(13);                      // Servo 1 attached to pin 13
  servo_camera_tilt.attach(11);                     // servo 2 attached to pin 11
  servo_camera_tilt.write(servo_camera_angle_center);  //setting servo 2 to 90 degrees
  servo_camera_pan.write(servo_camera_angle_center);   //setting servo 1 to 90 degrees
  servo_lights_pan.attach(10);                        // servo 3 attached to pin 10
  servo_lights_pan.write(servo_light_angle_center);    //setting servo 3 to 90 degrees
  digitalWrite(LIGHT_1, LOW);                          //Lights OFF during setup.
  digitalWrite(LIGHT_2, LOW);
  }
void Setspeed_robot(char robot_direction, int robot_speed, int robot_turn, int speed_low)  // Set speed and direction of motors
{
  unsigned char SendByte = 0;
  switch (robot_direction)
  {
   case 'f':                                 // Pressing "f": robot moves forward at a fixed speed
    SendByte = 0xDA;
    Serial.print("  Forward, setting speed to:   ");
    TREXJr.print(SendByte, BYTE);
    TREXJr.print(robot_speed, BYTE);
    TREXJr.print(robot_speed, BYTE);
    Serial.println(robot_speed, DEC);
    break;
    //delay(100);
  case 'b':                                  //"b" Backward
      Serial.print("  Backward, setting speed to:   ");
      SendByte = 0xD5;
      TREXJr.print(SendByte, BYTE);
      TREXJr.print(robot_speed, BYTE);
      TREXJr.print(robot_speed, BYTE);
      Serial.println(robot_speed, DEC);
      break;
      //delay(100);
  case 'l':                                // "l" Left turn at fixed speed
      Serial.print("  Turn left, setting speed to:   ");
      SendByte = 0xD6;
      TREXJr.print(SendByte, BYTE);
      TREXJr.print(robot_turn, BYTE);
      TREXJr.print(robot_turn, BYTE);
      Serial.println(robot_turn, DEC);
      break;
      //delay(100);
  case 'r':                                   // "r" Right turn at fixed speed
      Serial.print("  Turn right, setting speed to:   ");
      SendByte = 0xD9;
      TREXJr.print(SendByte, BYTE);
      TREXJr.print(robot_turn, BYTE);
      TREXJr.print(robot_turn, BYTE);
      Serial.println(robot_turn, DEC);
      break;
      //delay(100);
  case 'q':                                 // "q"  Quit = STOP !!!
      Serial.print("  STOP, setting speed to:  0 ");
      SendByte = 0xDA;
      TREXJr.print(SendByte, BYTE);
      TREXJr.print(0, BYTE);
      TREXJr.print(0, BYTE);
      //Serial.println(0, DEC);
      break;
    }
}
void setlights_1()                        //Function to turn one LED on/off
  {
    light_toggle_1 =!light_toggle_1;    //toggle LED lights on/off each time this routine is called
    switch (light_toggle_1)
    {
      case false:
          digitalWrite(LIGHT_1, HIGH);
          break;
     case true:
      digitalWrite(LIGHT_1, LOW);
          break;
    }
  }
void setlights_2()                     //Function to turn 2nd LED on/off
  {
    light_toggle_2 =!light_toggle_2;
    switch (light_toggle_2)
    {
      case false:
          digitalWrite(LIGHT_2, HIGH);
          break;
     case true:
      digitalWrite(LIGHT_2, LOW);
          break;
    }
  }

//*****************************************************************
void setcamera_pan(char pan_input, int pan_angle, int pan_angle_min, int pan_angle_max, int pan_angle_circle)      //Function to create pan motion of the servo with the wireless camera
  {
    int pan_current;
     if (pan_input == 'p')
      camera_toggle =!camera_toggle;          //toggle servo between pan and a fixed forward position when key "p" is pressed on the keyboard
    if (camera_toggle ==false)
    servo_camera_pan.write(90);
     else
     {
    if (pan_angle_min<=pan_angle && pan_angle <= pan_angle_max)
      pan_current=pan_angle;
    if (pan_angle_max < pan_angle && pan_angle <= pan_angle_circle)
      pan_current=pan_angle_max-(pan_angle-pan_angle_max);
    servo_camera_pan.write(pan_current);
    delay (10);
     }
  }

void setcamera_tilt()
  {
  servo_camera_tilt.write(90);      //Tilt servo is fixed in horizontal position
  delay(10);
  }
 void setlights_pan(char lights_input, int pan_angle, int pan_angle_min, int pan_angle_max, int pan_angle_circle)  //Function to create pan motion of the LDED
 {
    int pan_current;
    if (pan_angle_min<=pan_angle && pan_angle <= pan_angle_max)
      pan_current=pan_angle;
    if (pan_angle_max < pan_angle && pan_angle <= pan_angle_circle)
      pan_current=pan_angle_max-(pan_angle-pan_angle_max);
    servo_lights_pan.write(pan_current);
    delay (10);
   }
```

```
void loop()                              //********** Main program************************************
{
  setlights_2();                         //Toggle the 2nd LED everytime the function is called
 if (Serial.available() > 0)
      bytein = Serial.read();              // Read the incoming byte from the keyboard via USB cable or the Xbee module
 Setspeed_robot(bytein, speed_fixed, speed_turn, speed_min);    //set robot direction based on keys pressed on the keyboard
      delay(50);
 servo_camera_angle= servo_camera_angle + 3;                    //rotate camera from left to right and back by 3 degrees everytime the main program loop is called
   if (servo_camera_angle > servo_camera_angle_circle)
     servo_camera_angle =servo_camera_angle_min;
 setcamera_pan(bytein, servo_camera_angle, servo_camera_angle_min, servo_camera_angle_max,servo_camera_angle_circle);
   delay (50);
 setcamera_tilt();                                              //set camera for tilt to 90 degrees=horizontal position
 servo_light_angle=servo_light_angle+5;                         //rotate servo with 1st LED from left to right and back within predefined angles, turn 5 degrees at a time.
    if (servo_light_angle > servo_light_angle_circle)
     servo_light_angle =servo_light_angle_min;
setlights_pan(bytein, servo_light_angle, servo_light_angle_min, servo_light_angle_max,servo_light_angle_circle);
   delay (10);
setlights_1();                           //toggle the 1st LED on/off everytime this function is called, since the program is about 90ms long a blinking effect is visible.
   delay (10);
// ************ Main Program END ************
  }
```