King Fahd University of Petroleum and Minerals

Computer Engineering (COE)

COE 306 – Project Report Term 181

Remote Controlled Car

By Hassan Alawadh

201325050

Instructor	Dr. Aiman El-Maleh
Lab Instructor	Hazem Helmi

Abstract

This report will describe the Boe-Bot Robot (as a car only) controlled by PC keyboard or controller.

Dec 11, 2018

Table of Content

I. Introduction

The project aims to build a small car that can be controlled through a PC controller or keyboard. The remote-controlled cars are very interesting subject both in scientific research and practical applications. The RC cars have many indoor applications like, Surveillance buildings, Research entertainment and providing children with entertainment. RC cars have many outdoor applications as well like Military, Firefighting and spying.

II. Problem Statement

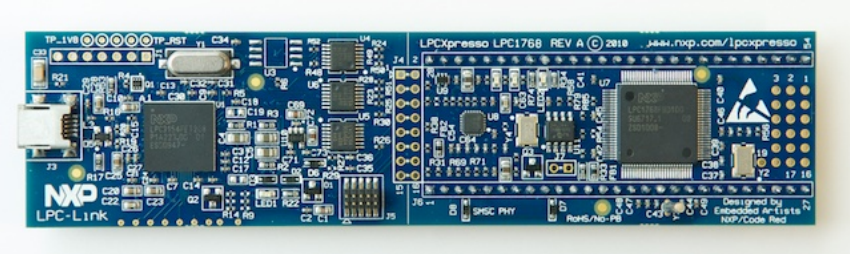
We have a huge amount of problems that can be solved by the remote-controlled cars. This essay will illustrate only some of these problems. First of all, Firefighting which is an important job, but it is very dangerous occupation. Due to that, RC cars could be used to work with fire fighters to reduce the risk of injury to victims. Secondly, the need of spy cameras has increased in the recent past. Remote controlled cars could help to keep us and our assets safe if we put cameras on them. Lastly RC cars could help in increase responsibility in our children and improve their fine motor skills.

III. Used Parts

These parts below are the parts used for this project:

- **LPC 1769 microcontroller:**

  The brain of the system, and it is responsible of sending and receiving signals between the different parts

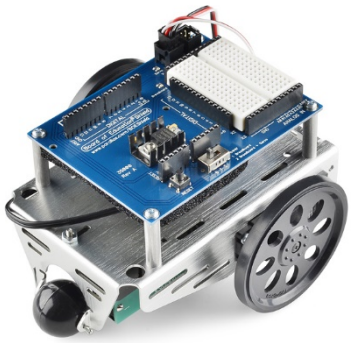


- **Two servo motors with continuous rotation:**

  A Parallax continuous rotation servo meter is used to rotate the two wheels of the car.

- **Boe-Bot Robot:**

  We can use any alternative car. I used this boe-bot robot because it already has two wheels suitable for the Parallax continuous rotation servo meter.



- **PC controller:**

  The PC controller will be used to send signals to the microcontroller through the PC. We can use the keyboard of the PC instead of the controller.

IV. The System

The purpose of the project is to control a car remotely. First, to remote the car we have to use pulse width modulation (PWM) to rotate the wheels that

connected to a servo meter. I let MR0 register which control the period duration of the PWM signal equal to 500,000 which is equivalent to half a second. Then I make the duty cycle equal to 50,000. When I latch MRx register to less than its duty cycle, it will rotate to the reverse direction of latching it to bigger than its duty cycle. Thus, I used the values 25000 and 75000 for latching the MRx register.

Secondly, I write switch statement in an infinite while loop to take a scanned integer and move the car based on the given number as follows:

- 4 to move to the left (by making only the right wheel moving).
- 8 to move forward.
- 5 to move backward.
- 6 to move to the right (by making only the left wheel moving).
- 7 to move sharply to the left (by making the right wheel moving forward and the left wheel moving backward).
- 9 to move sharply to the right (by making the left wheel moving forward and the right wheel moving backward).
- 0 to stop the car.

The sharp left and right moves are very efficient in time and space that the car needs to rotate.

Finally, to connect the PC controller to the keyboard I used Antimicro app to map the keyboard buttons to the controller. And Type Pilot app to enter ‘enter button’ automatically whenever I press a number in the keyboard, so I do not have to press Enter permanently while execution.

V. The code

```
#ifdef __USE_CMSIS
#include "LPC17xx.h"
#endif

#include <cr_section_macros.h>


volatile static int i;
int n;

int main(void) {
	LPC_GPIO0->FIODIR |= 1<<22;

	LPC_PINCON -> PINSEL4 |= (1<<6);
	LPC_PINCON -> PINSEL4 |= (1<<10);
	LPC_PWM1 -> MR0 = 500000;// period
	LPC_PWM1 -> MR6 = 50000; //pulse
	LPC_PWM1 -> MR4 = 50000; //pulse
	LPC_PWM1 -> TCR |= (1<<0) | (1<<3); // 0 Counter Enable 3 PWM
Enable
	LPC_PWM1 -> PCR |= (1<<12);
```

```
        LPC_PWM1 -> PCR |= (1<<14);
        LPC_PWM1 -> MR4 = 0; LPC_PWM1 ->LER |= (1<<4);
        LPC_PWM1 -> MR6 = 0; LPC_PWM1 ->LER |= (1<<6);


    while (1) {

        printf("enter");
        scanf ("%d", &n);
        LPC_PWM1 -> MR4 = 0; LPC_PWM1 ->LER |= (1<<4);
          LPC_PWM1 -> MR6 = 0; LPC_PWM1 ->LER |= (1<<6);

switch (n) {

      case 6:
            LPC_PWM1 -> MR4 = 75000;
            LPC_PWM1 ->LER |= (1<<4);
            LPC_PWM1 -> MR6 = 0;
            LPC_PWM1 ->LER |= (1<<6);
            break;

      case 8:
            LPC_PWM1 -> MR6 = 25000;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 75000;
            LPC_PWM1 ->LER |= (1<<4);
            break;

      case 5:
            LPC_PWM1 -> MR6 = 75000;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 25000;
            LPC_PWM1 ->LER |= (1<<4);
            break;

      case 4:
            LPC_PWM1 -> MR6 = 25000;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 0;
            LPC_PWM1 ->LER |= (1<<4);
            break;

      case 0:
            LPC_PWM1 -> MR6 = 0;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 0;
            LPC_PWM1 ->LER |= (1<<4);
            break;

      case 9:
            LPC_PWM1 -> MR6 = 75000;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 75000;
            LPC_PWM1 ->LER |= (1<<4);
            break;

      case 7:
```

```
            LPC_PWM1 -> MR6 = 25000;
            LPC_PWM1 ->LER |= (1<<6);
            LPC_PWM1 -> MR4 = 25000;
            LPC_PWM1 ->LER |= (1<<4);
              break;
      }
    }
    return 0;
}
```

VI. Conclusion

To summaries, the PC controller which is connected to the PC by Bluetooth or cable will send specific characters to the PC which then take these characters and pass them to the LPC1769. Which will send the corresponding instructions to move the car based on these characters.

VII. References

- https://github.com/AntiMicro/antimicro
-
- https://type-pilot.soft112.com/
- https://www.embeddedartists.com/wp-ontent/uploads/2018/07/LPCXpressoLPC1769revB.pdf
- https://www.nxp.com/docs/en/user-guide/UM10360.pdf
- https://www.nxp.com/support/developer-resources/evaluation-and-development-boards/lpcxpresso-boards/lpcxpresso-board-for-lpc1769-with-cmsis-dap-probe:OM13085
- https://www.parallax.com/product/900-00008
- https://www.parallax.com/product/boe-bot-robot
- https://www.xbox.com/en-US/xbox-one/accessories/controllers/recon-tech-special-edition