UNIVERSITY OF NORTHERN IOWA

TECH 3157 MICROCONTROLLERS APPLICATIONS SECTION 01

LAB 2 : PROGRAM ARDUINO UNO BOARD USING ANSI C AND ROBOT NAVIGATION FUNCTION

PROFESSOR: Dr. JIN ZHU

STUDENT: DIDAN JUNQUEIRA RIBEIRO

OCTOBER 1, 2015
CEDAR FALLS, IA

Objectives

1. Learn to program Arduino board using Standard C functions.
2. Understand the operations of pointer, address access and bitwise operators.
3. Learn to troubleshoot and debugging the program.
4. Learn to use user defined functions
5. Learn to control robot navigation using ramping up/down.

Equipment

BOE-BOT Shield Kit, Red LED, Piezo Buzzer.

Procedure and Design

The first part of the the Lab 2 was test blink program and the requirement was blink faster and in a different pattern for a red LED using standard C. Second, the goal was debugging two programs to blink a LED which was running but without blink the LED. For these first two parts of Lab 2 was used the followed LED connection:

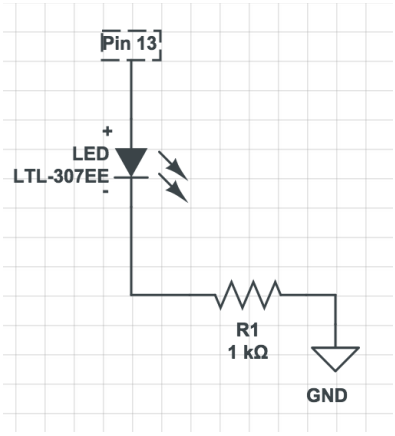

Figure 1: LED Connection on Arduino Board

Finally, the last part of Lab 2 was about to learn more functions on robot navigation. First requirement was be able to control the robot turns in the following sequence:

- Move forward for 2 second;
- pivot forward to left for 4 second;
- pivot backward to left for 4 second;
- move forward for 2 seconds;
- pivot backward to right for 1 second;
- pivot forward to right for 2 seconds;

- pivot back to right for 1 second;
- move backward for 3 seconds.

For that task, was defined on the code functions to do each movement. After that, was asked on Lab requirement 3.2 to modify the program to control the robot to switch from move forward to backward smoothly. The robot should ramp up to full speed and continue forward for 3 seconds, then ramp down to stop and ramp up again to full speed backward and continue backward for 3 seconds, and finally ramp down to stop. In this part was implemented a new function called rampMove() and defined some parameters.

Observation and Results

The following code contain the results for lab requirement 3.1:

```
/*TECH 3157 Lab 2 Lab requirement 3.1 Control the robot turns */

#include <Servo.h>      // Include servo library

void forward  (unsigned int time);  //full speed forward for a duration of time (ms)
void turnLeft (unsigned int time);  //Left turn for a duration of time(ms)
void turnRight(unsigned int time);  //Right turn for a duration of time(ms)
void backward (unsigned int time);  // Full -speed Backward for a duration of time (ms)
```

```
void pivotForwardLeft  (unsigned int time);
void pivotBackwardLeft (unsigned int time);
void pivotForwardRight (unsigned int time);
void pivotBackwardRight(unsigned int time);

void disableServos();   // Halt servo signals

Servo servoLeft;        // Declare left and right servos
Servo servoRight;

void setup()
{

tone(10, 3000, 1000);              // Play tone for 1 second
delay(1000);                       // Delay to finish the tone

servoLeft.attach(11);              // Attach left signal to pin 11
servoRight.attach(12);             // Attach right signal to pin 12

forward(2000);
pivotForwardLeft(4000);
pivotBackwardLeft(4000);
```

```
forward(2000);
pivotBackwardRight(1000);
pivotForwardRight(2000);
pivotBackwardRight(1000);
backward(3000);
}

void loop()
{
}

void forward (unsigned int time)   // Full-speed Forward function

{
servoLeft.writeMicroseconds(1625);
servoRight.writeMicroseconds(1300);
delay(time);    // Maneuver for time ms
}

void turnLeft(unsigned int time)  // Left turn function

{
```

```
servoLeft.writeMicroseconds(1300);
servoRight.writeMicroseconds(1300);
delay(time);  // Maneuver for time ms
}

void turnRight(unsigned int time)   // Right turn function

{
servoLeft.writeMicroseconds(1700);
servoRight.writeMicroseconds(1700);
delay(time);     // Maneuver for time ms
}

void backward(unsigned int time)  // Full-speed Backward function

{
servoLeft.writeMicroseconds(1300);
servoRight.writeMicroseconds(1613);
delay(time);
}

void pivotForwardLeft  (unsigned int time)   // Right wheel clockwise
```

```
{
  servoLeft.writeMicroseconds(1500);  // Left wheel stay still
  servoRight.writeMicroseconds(1300); // Right wheel clockwise
  delay(time);
}

void pivotBackwardLeft (unsigned int time) // Right wheel counterclockwise
{
  servoLeft.writeMicroseconds(1500);  // Left wheel stay still
  servoRight.writeMicroseconds(1700); // Right wheel counterclockwise
  delay(time);
}

void pivotForwardRight (unsigned int time)  // Left wheel clockwise
{
  servoLeft.writeMicroseconds(1300);  // Left wheel clockwise
  servoRight.writeMicroseconds(1500);  // Right wheel stay still
  delay(time);
}

void pivotBackwardRight (unsigned int time)          // Left wheel counterclockwise
{
```

```
  servoLeft.writeMicroseconds(1700);          // Left wheel counterclockwise
  servoRight.writeMicroseconds(1500);         // Right wheel stay still
  delay(time);
}

void disableServos()

{
servoLeft.detach();
servoRight.detach();
}

/* end of Lab 3.1 requirement */
```

Through this code the robot could execute the move sequences as asked on lab requirement 3.1

Next, the following code contain the results for Lab requirement 3.2:

```
/*TECH 3157 Lab 2 requirement 3.2: Ramping up to full speed forward pin 11: control signal
to Left servo pin 12: control signal to right servo pin 10: tone to the piezo buzzer
*/
```

```
#define FORWARD  0x0         // ramp up forward
#define BACKWARD 0x2       // ramp up backward
#define UP 0x0                    // ramp up forward
#define DOWN 0x1                // ramp down forward

#include <Servo.h>      // Include servo library

// Functions Prototype
void forward  (unsigned int time);      //Full speed forward for a duration of time (ms)
void backward (unsigned int time);    // Full -speed Backward for a duration of time (ms)
void rampMove (unsigned char control, unsigned char speedStep);
void rampingUpForward     (unsigned char speedStep =1 ); /* Ramp up to full speed (speedStep
between 1 to 20, default 1)  */
void rampingUpBackward   (unsigned char speedStep =1);  // Ramp down to stop
void rampingDownForward  (unsigned char speedStep =1); // Ramp up to full speed backward
void rampingDownBackward (unsigned char speedStep =1); // Ramp down to stop
void disableServos();   // Halt servo signals

Servo servoLeft;        // Declare left and right servos
Servo servoRight;
```

```
void setup()
{

  tone(10, 3000, 1000);      // Play tone for 1 second
  delay(1000);                // Delay to finish the tone
  servoLeft.attach(11);      // Attach left signal to pin 11
  servoRight.attach(12);    // Attach right signal to pin 12

  rampMove(FORWARD+UP,2);              // ramping up to full speed in 1 seconds
  forward(3000);                                // continue forward for 3 seconds
  rampMove(DOWN+FORWARD,2);         // ramping down to stop in 1 sec
  rampMove(BACKWARD+UP,2);           // ramping up to full speed backward in 1 sec
  backward(3000);                              // continue backward for 3 seconds
  rampMove(DOWN+BACKWARD,1);       // ramping down to stop in 1 second

  disableServos();          // Stay still indefinitely

}

void loop()
{
 }
```

```
void disableServos()
{
  servoLeft.detach();
  servoRight.detach();
  }

void forward (unsigned int time)   // Full-speed Forward function

{
  servoLeft.writeMicroseconds(1625);
  servoRight.writeMicroseconds(1300);
  delay(time);    // Maneuver for time ms
  }

void backward(unsigned int time)      // Full speed backward function

{
  servoLeft.writeMicroseconds(1300);
  servoRight.writeMicroseconds(1613);
  delay(time);
  }
```

```
void rampingUpForward(unsigned char speedStep)  // ramping up forward
{
  if (speedStep == 0 or speedStep > 20)
  speedStep = 20; // if the speedStep is out of range [1..20], set to 20
  for(int speed = 0; speed <= 100; speed += speedStep) //
{
  servoLeft.writeMicroseconds (1500+speed);  // us = 1500,1502,...1598,1600
  servoRight.writeMicroseconds(1500-speed);  // us = 1500,1498,...1402,1400
  delay(20);                                 // 20 ms at each speed
}

}

void rampingUpBackward (unsigned char speedStep) // ramping up backward
{
  if (speedStep == 0 or speedStep > 20)
  speedStep = 20; // if the speedStep is out of range [1..20], set to 20
  for(int speed = 0; speed <=100 ; speed += speedStep) //
{
  servoLeft.writeMicroseconds (1600-speed);  // us = 1500,1502,...1598,1600
```

```
  servoRight.writeMicroseconds(1500+speed);  // us = 1500,1498,...1402,1400
  delay(20);                                 // 20 ms at each speed
  }
}

void rampingDownForward(unsigned char speedStep) // ramping down forward
{
    if (speedStep == 0 or speedStep > 20)
        speedStep = 20; // if the speedStep is out of range [1..20], set to 20
        for (int speed = 100; speed >= 0; speed -= speedStep)
        {
          servoLeft.writeMicroseconds(1500+speed);  // t_pulse = 1600,1598,...1502,1500 us
          servoRight.writeMicroseconds(1600-speed); // t_pulse = 1400,1402,...1498,1500 us
          delay(20);            // 20 ms at each speed
          }
}

void rampingDownBackward (unsigned char speedStep)      // ramping down backward
{
  if (speedStep == 0 or speedStep > 20)
  speedStep = 20; // if the speedStep is out of range [1..20], set to 20
  for (int speed = 100; speed >= 0; speed -= speedStep)
```

```
  {
  servoLeft.writeMicroseconds(1500-speed);  // t_pulse = 1600,1598,...1502,1500 us
  servoRight.writeMicroseconds(1600+speed); // t_pulse = 1400,1402,...1498,1500 us
  delay(20);            // 20 ms at each speed
  }
}

void rampMove (unsigned char control, unsigned char speedStep)
{
//   control = 0x0; //ramp up forward   FORWARD+UP
//   control = 0x1; //ramp down forward   DOWN+FORWARD
//   control = 0x2; // ramp up backward   BACKWARD+UP
//   control = 0x3; // ramp down backward     DOWN+BACKWARD
          if (control == 0x0) rampingUpForward(speedStep);
          if (control == 0x1) rampingDownForward(speedStep);
          if (control == 0x2) rampingUpBackward(speedStep);
          if (control == 0x3) rampingDownBackward(speedStep);
}
/* end of Lab requirement 3.2 */
```

Conclusion

All the lab parts was done successful. All the steps was followed as in the lab 2 manual and in this way was possible got all the results expected from it. It applies for lab requirements which was accomplish the goal asked for it. There was some complications for last lab requirement but through questions was possible figure out what was asked to do and implement it.