Lab 2 Program Arduino UNO board using ANSI C and Robot Navigation Functions

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.

Lab Activities:

1. Test blink program using standard C

Program the Arduino IDE and test blink program as shown in Example Lab2-1. You may use onboard LED that is connected to pin 13, i.e. PortB.5, or connect an additional LED to pin 13 for easy display.

Example Lab2-1

```
void MyDelay(unsigned long mSecondsApx); //user defined delay

void setup()
{
   unsigned char *portDDRB;

   portDDRB = (unsigned char *) 0x24;  //address of Port B Data Direction Register

   *portDDRB |= 0x20;  //set pin 13 as an output
}

void loop()
{
   unsigned char *portB;

   portB = (unsigned char *) 0x25; //address of Port B Data Register

   *portB |= 0x20;  //turn on LED
   MyDelay(500);
   *portB &= 0xDF;  //turn off LED
   MyDelay(500);
}

void MyDelay(unsigned long mSecondsApx)
{
   volatile unsigned long i;
   unsigned long endTime = 800 * mSecondsApx;
   for (i = 0; i < endTime; i++);  //loops for delay
}
```

Lab Requirement 1: Please modify the program so that onboard LED blinks faster and in a different pattern (such as blink fast slow fast). Show the demo to the instructor.

PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Figure L2-1 PortB I/O registers in ATmega328P

Table L2-1 Pin mapping between UNO R3 Board Digital I/O and ATmega328P I/O

UNO pin	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATmega328P	PORTB [5:0]						PORTD[7:0]							

2. Debugging and Troubleshooting

Two programs (Lab2_21_BlinkInC.ino and Lab2_22_BlinkInC.ino) that try to implement the blink function similar to the program example 2-1 are uploaded to e-Learning. Please identify if it has compile errors(s) or runtime error(s). If yes, please also indicate the errors clearly and remove the bugs.

Lab Requirement 2: Please clearly indicate the bugs in the programs in your report and show the fixed program to the instructor.

3. More functions on Robot Navigation

Connect the right servo motor to pin 12, the left one to pin 11 via the servo port, and the piezo buzzer to pin 10 as you did in lab 1.

Lab Requirement 3.1: control the robot turns

Please modify the program so that the robot will complete the following move sequences:

a. move forward for 2 second;
b. pivot forward to left for 4 second;

c. pivot backward to left for 4 second;
d. move forward for 2 second;
e. pivot backward to right for 1 second;
f. pivot forward to right for 2 second;
g. pivot back to right for 1 second;
h. move backward for 3 second.

You can make the BOE Shield-Bot turn by pivoting around one wheel. To complete a pivoting turn, you need to keep one wheel still while the other rotates. For example, for pivot forward to left turn, the following code can be used:

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

Please define pivotForwardLeft, pivotBackwardLeft, pivotForwardRight, pivotBackwardRight functions in your program to implement the above requirements. The function prototype should look like:

```
void pivotForwardLeft(unsigned int time);
```

Please use lab 1 as your reference. Show the demo to the instructor and the program codes need to be included for your report.

Manipulate the speed

Gradually increase or decrease the speed of the servos instead of abruptly starting or stopping can increase the life expectancy of both your BOE Shield-Bot's batteries and your servos. According to Figure L2-2, we can see the rotational velocity is approximately proportional to the pulse width linearly from 1400 – 1600 μs. Try the example 2 as shown in the following. Remember to include the functions *forward(unsigned int time)* and *disableServos()* in your program (not included below to save space). The robot will ramp up to full speed and continue forward for 2 seconds. Observe the difference from the case without ramping up. Read and understand the program.

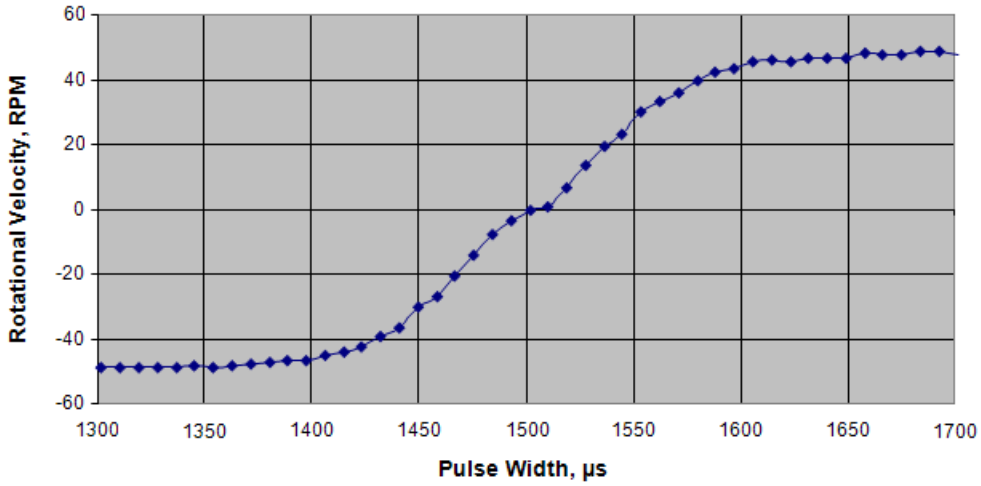

Figure L2-2 Transfer curve for Parallax Continuous Rotation Servo motor 900-00008 (from www. Parallax.com)

```
/*TECH 3157 Lab 2 example 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 */

#include <Servo.h>                    // Include servo library
void rampingUpForward(unsigned char speedStep =1 ); // Ramp up to full speed (speedStep
between 1 to 20, default 1)
void disableServos();                 // Halt servo signals

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

void setup()                          // Built-in initialization block
{
  tone(10, 3000, 1000);               // Play tone for 1 second
  delay(1000);                        // Delay to finish tone

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

  rampingUpForward();                 //ramping up to full speed in 2 seconds
  forward(2000);                      //continue forward for 2 seconds
  disableServos();                    // Stay still indefinitely
}

void loop()                           // Main loop auto-repeats
```

```
{                              // Empty, nothing needs repeating
}

/* Ramp up to full speed. Allowed speed step is between 1 to 20.
 Time to ramp up to full speed from stop = (2/speedStep) seconds
 For example, speedStep is 1: ramp up to full speed in 2 seconds
          2: ramp up to full speed in 1 seconds
          4: ramp up to full speed in 0.5 seconds
          20: ramp up to full speed in 0.1 seconds  */
void rampingUpForward(unsigned char speedStep)
{
  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
  }
}
/* end of example 3 */
```

You may use similar method to ramp down from full speed to stop, instead of a sudden stop. An example code is given below.

```
   for (int speed = 100; speed >= 0; speed -= speedStep) //
   {
   servoLeft.writeMicroseconds(1500+speed);   // t_pulse = 1600,1598,...1502,1500 us
   servoRight.writeMicroseconds(1500-speed);  // t_pulse = 1400, 1402,...1498,1500 us
   delay(20);                          // 20 ms at each speed
   }
```

Lab requirement 3.2:

Please 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. Functions should be used. In this part, you need to implement a rampMove( ) function with the following definitions and prototype:

```
#define FORWARD 0x0
#define BACKWARD 0x2
```

```
#define UP 0x0
#define DOWN 0x1
void rampMove ( unsigned char control, unsigned char speedStep);
```

The function rampMove will perform the function according to:

control = 0x0: ramp up forward
control = 0x1: ramp down forward
control = 0x2: ramp up backward
control = 0x3: ramp down backward
others: do not move and output warning in the serial output

For example, function rampMove(FORWARD+UP, 2) should make the robot to ramp up from stop to full speed forward in 1 second.

Show the demo to the instructor. Include the codes for your lab reports.

Report requirement:

Please follow the general requirements for Lab reports. Programs for Lab requirements 1 and 2 are not needed to be included in your report. The complete program in Lab requirement 3 is needed for your report. Please fully comment your program.