Lab 3 GPIO Access and Robot Tactile Navigation

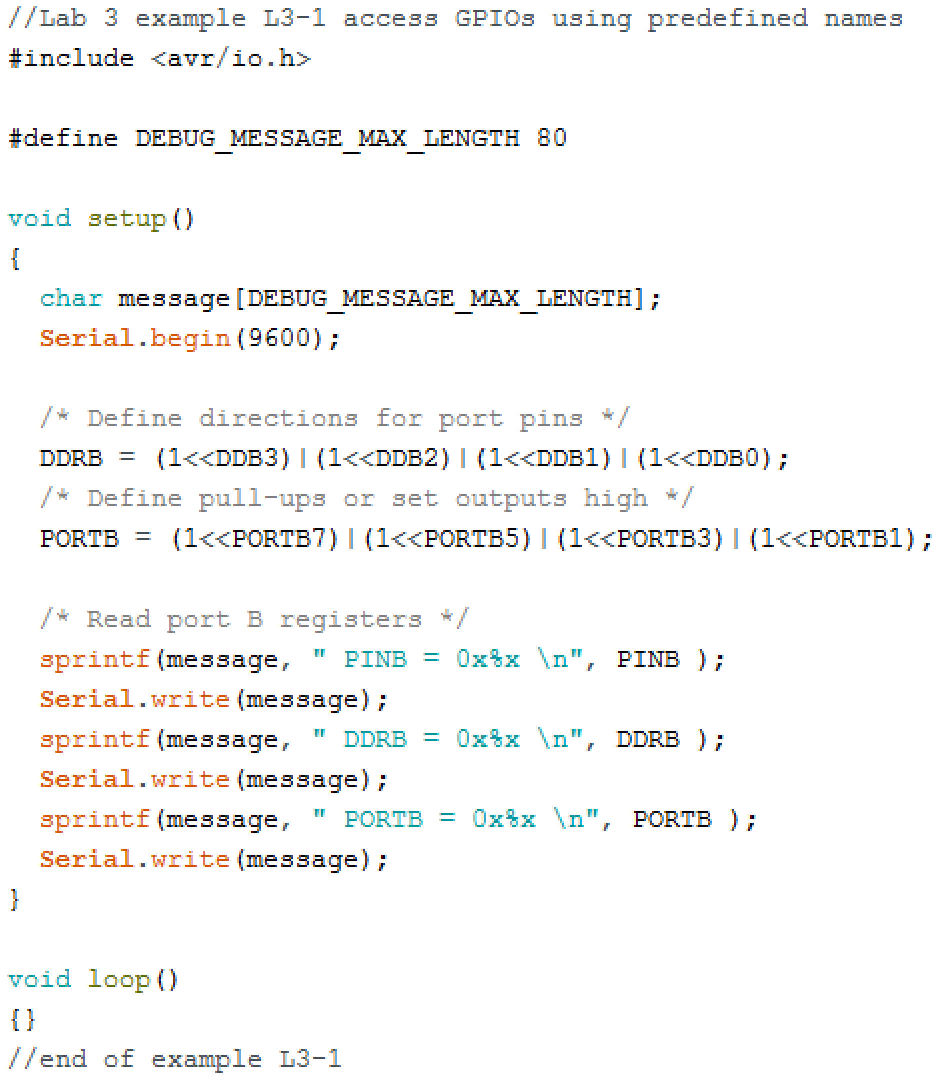
Objectives:

1. Learn to access GPIOs using predefined names.
2. Learn to control the robot navigation using tactile switches.
3. Learn to interface with seven-segment LEDs using GPIOs.

Lab Activities:

1. Access GPIO registers using predefined names

Read the example program L3-1 as shown below. Input and run the program. Show results in the serial monitor window.



```
//Lab 3 example L3-1 access GPIOs using predefined names
#include <avr/io.h>

#define DEBUG_MESSAGE_MAX_LENGTH 80

void setup()
{
  char message[DEBUG_MESSAGE_MAX_LENGTH];
  Serial.begin(9600);

  /* Define directions for port pins */
  DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
  /* Define pull-ups or set outputs high */
  PORTB = (1<<PORTB7)|(1<<PORTB5)|(1<<PORTB3)|(1<<PORTB1);

  /* Read port B registers */
  sprintf(message, " PINB = 0x%x \n", PINB );
  Serial.write(message);
  sprintf(message, " DDRB = 0x%x \n", DDRB );
  Serial.write(message);
  sprintf(message, " PORTB = 0x%x \n", PORTB );
  Serial.write(message);
}

void loop()
{}
//end of example L3-1
```

Lab requirement 1: Make sure you understand the program. Check the serial debug window. What are the values displayed in the debug window? Are the results the same as what you expect to have? Please explain your results.

2. Interface with pushbuttons

Next, we will learn to interface with a switch or pushbutton. Please connect a switch or pushbutton as shown in Figure L3-1. This switch output is active-LOW (i.e. when the switch is closed, the input to pin 8 will be LOW). Input and run the program example L3-2. Check the debug window output. Read the program and understand how it works.

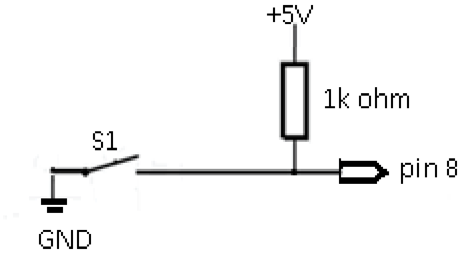


Figure L3-1 SPST switch connection

```
//Lab 3 example L3-2
//Display message if the input from pin 8 (i.e. PORTB.0) changes
#define DEBUG_MESSAGE_MAX_LENGTH 80

void setup()
{
 PORTB |= (1<<PORTB0);     //Turn on the pull-up resister for pin 8
 DDRB &= 0b11111110;      // set pin 8 as an input
 Serial.begin(9600);
}

void loop()
{
 char message[DEBUG_MESSAGE_MAX_LENGTH];
 static unsigned char previousSample;
 static unsigned char currentSample;

 currentSample = PINB & 0x01 ;  //obtain the status of input from pin 8
 sprintf(message, "previousSample = 0x%x \n", previousSample);
 Serial.write(message);
 sprintf(message, "currentSample = 0x%x \n", currentSample);
 Serial.write(message);
 if (currentSample != previousSample)
 {
    //input has changed, display promot
    Serial.write("change occurs...........................\n");
  }
 //store the current sample as the next previous sample
 previousSample = currentSample;
 delay(1000);
}
//end of Lab 3 example L3-2
```

Lab requirement 2: Please prepare a program that may use a switch or pushbutton to control the LED patterns. If the switch is on, four LEDs blinks simultaneously every second; if the switch is off, four LEDs take turns to turn on and off in sequence and repeat the pattern. Show the demo to the instructor. You may refer to the examples in the lecture 6 for control of multiple LEDs. Please remember to include the circuit connection diagram and the fully commented program for your report.

3. Tactile Navigation with Whiskers

Tactile switches are also called bumper switches or touch switches, and they have many applications in robotics. Automated factory lines might use tactile switches to count objects, and to align parts for a certain step in a manufacturing process. In each case, switches provide inputs that trigger the actions of a robot. A microprocessor monitors the input signals from the switches and takes different actions depending on if the switch is on or off.

The robot kit comes with a pair of whisker switches that give the BOE Shield-Bot the ability to sense its surroundings through touch as it roams around, much like a cat's whiskers. The whiskers are connected to ground (Vss) because the plated holes at the outer edge of the board are all connected to GND. The metal standoffs and screws provide the electrical connection to each whisker. Please follow the instruction in the following link to install the whiskers: http://learn.parallax.com/node/236

The connection diagram is shown in Figure L3-2. The pin we used to connect left whisker tactile switch is pin 8 and that for right whisker is pin 9. Run the following program to test if the whisker switch works. It should display in the serial monitor the status of both whiskers. When the whisker is pressed to close the switch, the input read will be 0, otherwise it should be 1. Verify it works before you continue to next step.

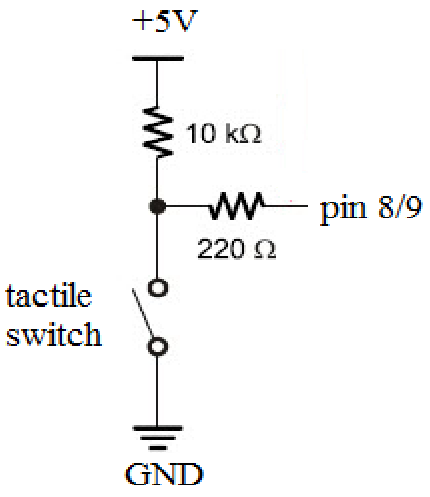


Figure L3-2 tactile

```
/*TECH 3157 Lab 3 example 3
  Tactile navigation with whiskers
  pin 8: left whisker, active LOW
  pin 9: right whisker, active LOW
  pin 10: piezo buzzer
  pin 11: control signal to Left servo
  pin 12: control signal to right servo
  pin 10: tone to the piezo buzzer */

#define piezo 10

void setup()                          // Built-in initialization block
{
  Serial.begin(9600);

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

  DDRB &= 0b11111100;                  //set pin 8 and 9 as inputs
}

void loop()                           // Main loop auto-repeats
{
  char message[80];
  char leftWhisker;
  char rightWhisker;

  leftWhisker = PINB & 0x01;               //pin 8 (PINB0) is connected to the left whisker
  rightWhisker = (PINB & 0x02)>>1;          //pin 9 (PINB1) is connected to the right whisker
  sprintf(message, "Left Whisker: %x, Right Whisker: %x \n",leftWhisker, rightWhisker);
  Serial.write(message);
}
```

Lab requirement 3:

Please modify the program so that the robot can navigate with whiskers. When the left whisker is pressed, the robot should be able to go backward a little and then turn right; when the right whisker is pressed, the robot should be backward and turn left to avoid the obstacles. Use the following rules for navigation:

1) If both whiskers are touched, the robot move backward for 1 second and then pivot turn left for 1 second.
2) If only left whisker is pressed, the robot move backward for 1 second and then turn right for half second;
3) If only right whisker is pressed, the robot move backward for 1 second and then turn left for half second;
4) If no whisker is pressed, the robot move forward for 20 ms;

Please use a flowchart to help you implement the system. Prepare the program and show demo to the instructor. Please remember to include the circuit connect diagram, flowchart and properly commented program for your report.

4. Interface with 7-segment LEDs

In this part, you will learn to interface with a seven segment LED using GPIO. You will use the 7-segment LED to indicate number of times a robot meets obstacles. A common cathode seven-segment led connections to the board is shown in Figure L3-3. Please do not forget to use the proper size of current limit resistors.

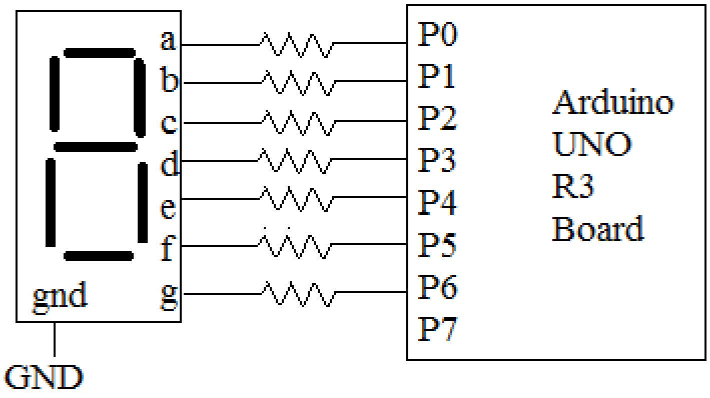


Figure L3-3 Common cathode 7-segment LED connection.

Segments a to g are connected to pin 0 to 6 in the Arduino UNO board. You need to convert from the decimal number (the counted touches) to 7-seg display output. One sample function that can be used to perform the conversion is shown in the following:

```
//return the seven segment LED output (common cathode) for decimal number count (0-9)
unsigned char sevenSegLED(char count)
{
  //for common cathode 7-seg LEDs)
  unsigned char ledDisplay[] ={63,6,91,79,102,109,125,7,127,111};

  if ((0 <= count) && (count < 10))
    return ledDisplay[count];
  else
    return 0;
}
```

This function will return the corresponding common cathode seven segment LED outputs for a given decimal number ***count***. The valid ***count*** range is from 0 to 9. Other values will disable the display. The values stored in ledDisplay[] array are the decimal numbers corresponding to the outputs needed to show the digital, as shown in Figure L3-5. For example, to display number 0, we need to turn on segments a, b, c, d, e, f. So the corresponding PortD value should be 0b00111111 in consistent with the connections in Figure L3-4. The decimal value of binary 00111111 is 63. Therefore, if count = 0, ledDisplay[count] = 63, which enables segments a, b, c, d, e, f and digital 0 is displayed.

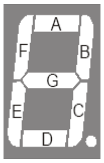
Number displayed	LED segments lit on	.GFEDCBA	PortD value in binary	PortD value in decimal	7-segment LED display--segment definition
0	A,b,c,d,e,f	**00111111**	%00111111	63	
1	B,c	**00000110**	%00000110	6	
2	A,b,g,e,d	**01011011**	%01010011	91	
3	A,b,c,d,g	**01001111**	%01001111	79	
4	B,c,f,g	**01100110**	%01100110	102	

Figure L3-4Value determination for seven-segment LED display

Lab requirement 4:

You need use a seven segment LED to indicate number of times a robot meets obstacles. The robot moves the same way as indicated in Lab requirement 3. Each time the whisker is pressed, the number displayed will increment by 1 and the buzzer beeps briefly (assume the piezo buzzer is connected to pin 10):

```
tone(10, 3000, 100);
delay(100);
```

Once the number exceeds 9, it will reset the number back to 0 and make a warning of two beeps using:

```
tone(10, 1850, 200);
tone(10, 2250, 300);
delay(300);
```

Design the complete system and prepare the program. Show the demo to the instructor. Please remember to include the circuit connection diagram, flowchart and properly commented program for your report.

Report requirements:

Please follow the general requirements for lab reports. Answer the questions for each given parts. Circuit connection diagrams need to be provided if there is external components used. Please also include the flow chart for Lab requirement 3 and 4 part. Please properly comment your programs.