Lab 6 Interrupt Program and Robot Navigation using Infrared Receivers

Objectives:

1. Understand the operation of interrupts in ATmega328P.
2. Learn to program interrupt in C.
3. Learn to interface with Infrared receivers.

Lab Activities:

1. Interrupt program – external interrupt

In this part, you will learn to program external interrupt INT0 and INT1. A global variable is modified whenever the pushbutton is pressed and the value is displayed in Serial Monitor. Please connect two pushbuttons to the pin 2 (INT0) and pin3 (INT1) as shown in Figure L6.1.

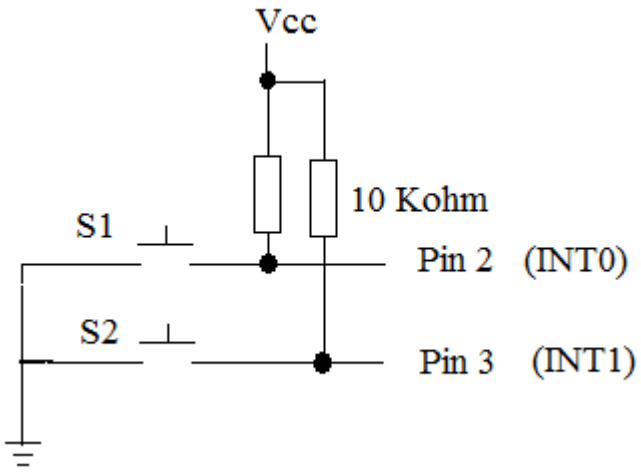

Figure L6.1 External interrupt connection

In the program, both external interrupt is set as rising edge triggered. When the pushbutton S1 is released after press, you should observe that the pos value displayed in Serial Monitor decreases, while pos increases when S2 is released.

```
#include <avr/io.h>
#include <avr/interrupt.h>
int pos;

void setup()
{
   EICRA = 0x0F;  //set rising edge triggered
```

```
    EIMSK = (1<<INT0)|(1<<INT1);  //set external interrupt mask
    sei();          //enable global interrupt bit

    Serial.begin(9600);
}

void loop()
{
    char message[80];

    sprintf(message, "pos = %d\n", pos);
    Serial.write(message);
    delay(500);
}

ISR (INT0_vect)
{
  pos ++;   //increment pos when SW1 is pressed
}

ISR (INT1_vect)
{
    pos --;  //decrement pos when SW2 is pressed
}
```

Lab requirement 1.1:

Use the whisker tactile switch used in Lab 3 and interrupt, display the number of touches in left and right side, respectively. Please use falling edge triggered mode for quick response on the display. Please also turn on the right LED when the right whisker is touched and turn on the left LED when the left whisker is touched. Observe the results and show the demo the instructor. **Report the observation and include the modified program in your report.**

2. Infrared (IR) Receiver

In this part, you will learn how to use IR receivers to detect objects, the similar mechanism that used for a TV remote controller. A TV remote controller sends signals at a rate of about 38 kHz and the IR receiver in the TV only responds to infrared signal at this frequency. The advantage of this method is that the ambient light does not affect the IR receiver.

We will use an IR LED to keep sending the 38 kHz signal and if the IR receiver received the signal, it will output an active signal and let the microcontroller know.

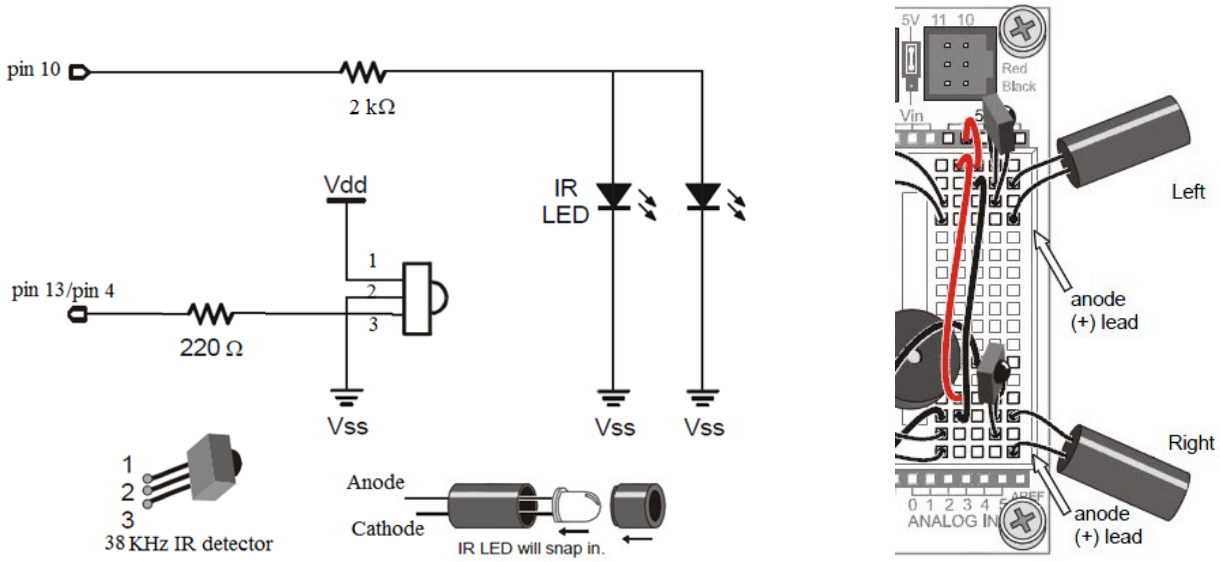

Figure L6.2 IR transmitter and Receiver connection diagram

Please assemble the IR LED and connect the circuit as shown in Figure L6.2. Please point the IR LED away from the detector. The black shield used is to prevent the IR flashing light leakage to strike to IR detector directly. The IR receivers used here are active LOW. When an object is close enough and not dark black, it reflects the 38 kHz IR signal back to the IR receiver and then it will output a LOW, otherwise it outputs a HIGH, as shown in Figure L6.3.

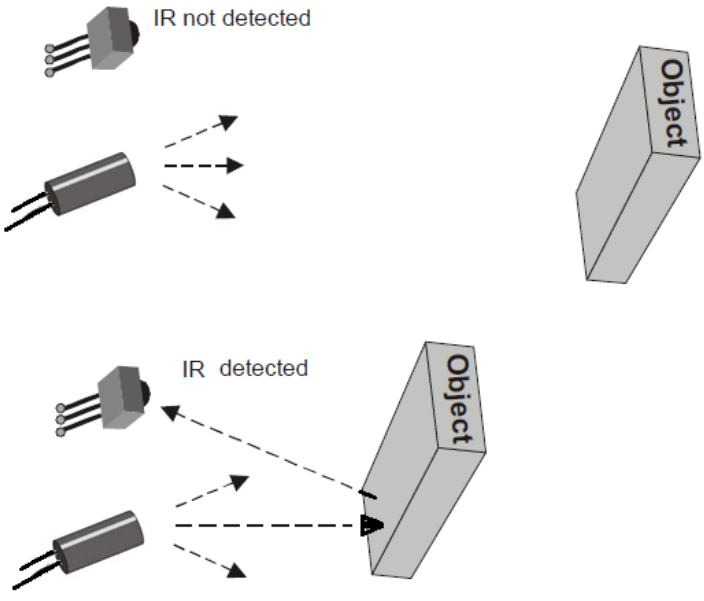

Figure L6.3 Setup for object detection.

In the first step, we will use only the left IR receiver to test its object detection function. Input and run the program example 2 given in the following:

```
/*Lab 6 Example 2
 * Display IR detector output resutls
 * pin 10 - Timer 1 Ch. B is used to generate 38 KHz signal
 * pin 13 -- left IR receiver output
 */
 #include <avr/io.h>

 void carrierInit();
 void pwmEnable();
 void pwmDisable();

 char message[80];

 void setup()
 {
   //USE Timer1 to generate 38 KHz carrier
   carrierInit();
   Serial.begin(9600);
 }

 void loop()
 {
   char irLeft;

   pwmEnable();
   delay(1);    //1 ms delay. The transmission length should <= 1.2 ms
   pwmDisable();
   irLeft = (PINB & 0x20)>>5; //get the status infrared receiver at pin 13
   sprintf(message, "irLeft = %d \n", irLeft);
   Serial.write(message);
   delay(500);
 }

 void carrierInit()
{
   /*intialized timer 1 for frequency 38KHz, OC1B at PB2(pin 10)
```

```
    phase-correct pwm, Ch.B, prescaler=1, 50% duty ratio*/
  DDRB |= 0x04;
  TCCR1A = 0x23; //0b00100011
  TCCR1B = 0x11; //0b00010001
  OCR1A = 211; //16M/2/38K = 211
  OCR1B = OCR1A *0.5;
}

void pwmEnable()
{
   /*enable Timer1 Channel B OC1B at PB2(pin 10)
   */
  TCCR1A = 0x23; //0b00100011
}

void pwmDisable()
{
   TCCR1A = 0x0; //discount OC0A and OC0B.
}
```

Lab requirement 2.1:

Run the program and move an object (such as your hand or a sheet of paper) in front of the IR LED. Observe the results displayed in Serial Monitor. It should display a 1 when there is no object and display 0 when the object is present. Report your observation in the report.

Lab requirement 2.2:

Modify the program so that it can detect the object in front of both sides. Please prepare a function to return the proximity detection result for each IR receiver. Use the function prototype as shown below:

char irDetect(char irReceiverPin);

For example, `irLeft = irDetect(13)` should get the output from the IR receiver connected to pin 13. Observe the results and show the demo the instructor.

Note: Before you continue to utilize the IR receivers to guide your robot, you also want to detect if there is any IR interference from other sources. You can accomplish this by not enabling the LED and testing the IR receiver output. Move your robot around and to see if there is 38 kHz signal detected. You may add additional output for easy observation in the Serial Monitor.

Lab requirement 2.3:

The IR LED light level, which is proportional to the current flow through the LED, will affect the object detection range. Therefore, you may adjust the resistor that is in series with LED to adjust the detection distance. Try several different resistors size and estimate the corresponding maximum detection distance.

Resistance (Ω)	Maximum Detection Distance
4.7 k	
2 k	
470	
220	

Pick one distance resistance that has the maximum detection distance about 2 inches for next step. Report the detection distances and your selection of IR LED resistor size.

Lab requirement 2.4:

Please modify the program so that the robot can navigate with its IR headlights and eyes. When an object in the left is detected, the robot should be able to turn right; when the object in the right is detected, the robot should turn left to avoid the obstacles. Please also add two LEDs (one in left, one in right, but not IR LED) to indicate if the object is detected on the side. Use the following rules for navigation:

- If both sides are detected, turn both LEDs on, and the robot turn around (180 degrees).
- If only left side is detected, turn left LED on and right LED off, and the robot turn right for about 45 degrees;
- If only right side object is detected, turn left LED off and right LED on, and the robot turn left side for about 45 degrees;
- If no object is detected, the robot move forward for 20 ms;

Implement the program and show demo to the instructor. Please include circuit connect diagram, flowchart and complete program for this part.

Report requirements:

Please follow the general requirements for lab reports. The part in Lab requirement 2.4 needs to include both the flowchart and a complete circuit connection diagram. Please fully comment your programs.