EE 3612 Microprocessor Systems Lab Design Project

Servo Motor Control and Speedometer Implementation.

Designed by Dr. Helferty

4/12/2020

OBJECTIVES

- To learn how to operate IO port of ATmega324PB Xplained Pro board using C
- To learn how to generate timing signals using the AVR timer
- To learn how to control Servo motor speed
- To learn how to utilize interrupt functionality of the processor
- To learn how to measure the duty cycle and calculate RPM of the motor
- Display speed of Servo motor on a LCD

Parts needed

1. Atmel board
2. Male-male connectors (lots of them!)
3. Servo motor
4. Resistors 1 300Ω (or 2 150Ω in series) and a 2.2KΩ
5. Optical switch
6. LED
7. Wheel with slot cut into it
8. Base to stabilize motor.

Discussion

In this project we will measure the speed of a servo motor and display the results on the LCD screen. The setup is shown below in Fig. 1. As the motor rotates a slot in the wheel will allow the optical sensor to send pulses to PB3 and use interrupt pin there to activate interrupt INT1. By counting the number of interrupts and timing the pulses you can calculate the rotations per second and display that on the LCD.

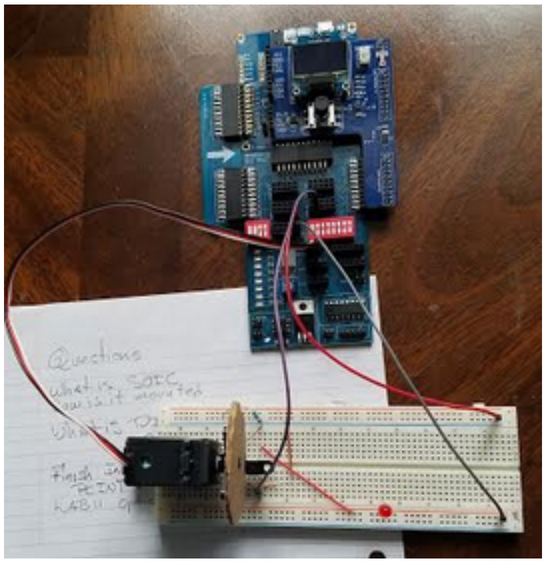


Figure 1: The servo motor setup.

Basically, the idea is to have a “wheel” with a slot cut into it and have it attached to the servo horn using the screw that hold the horn to the shaft. If you don’t have the screw hot glue works but please don’t use a permanent glue so it can be removed easily and replaced. See figure 2.

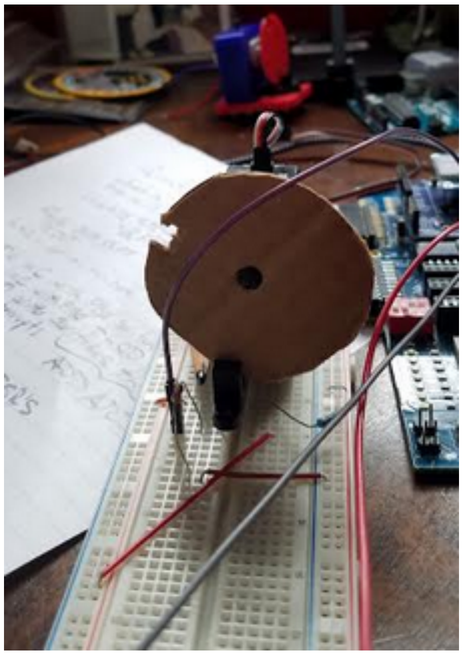


Fig. 2. Wheel mounted to servo with slot cut into the wheel.

You will also have to tape some kind of board behind the screws on the motor bottom so the motor “sits” on the breadboard and is in an upright position. The motor can then be taped to the breadboard to stabilize it as it rotates. See Figure 3.

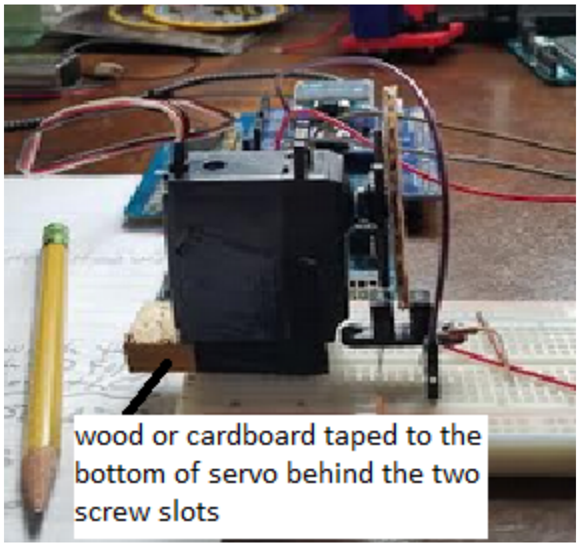
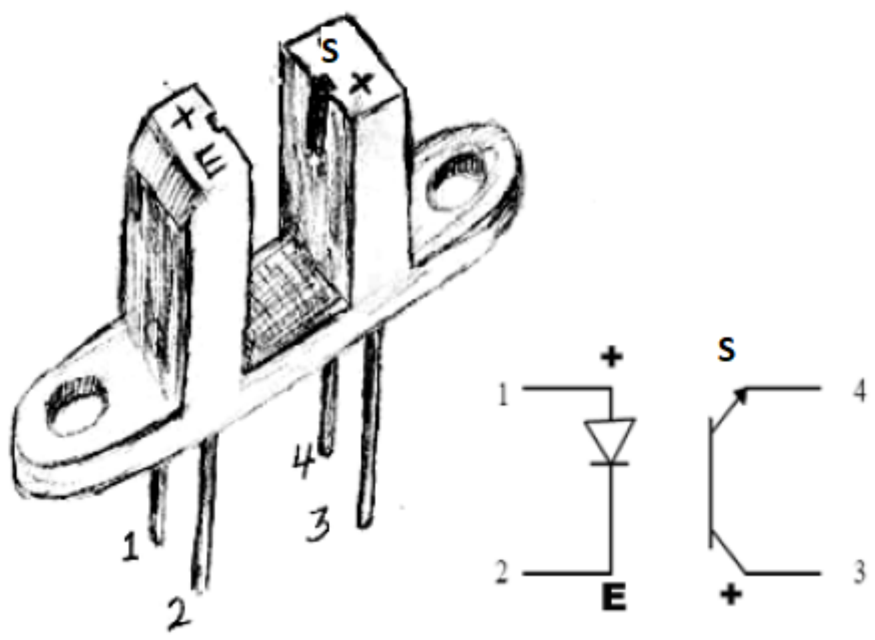
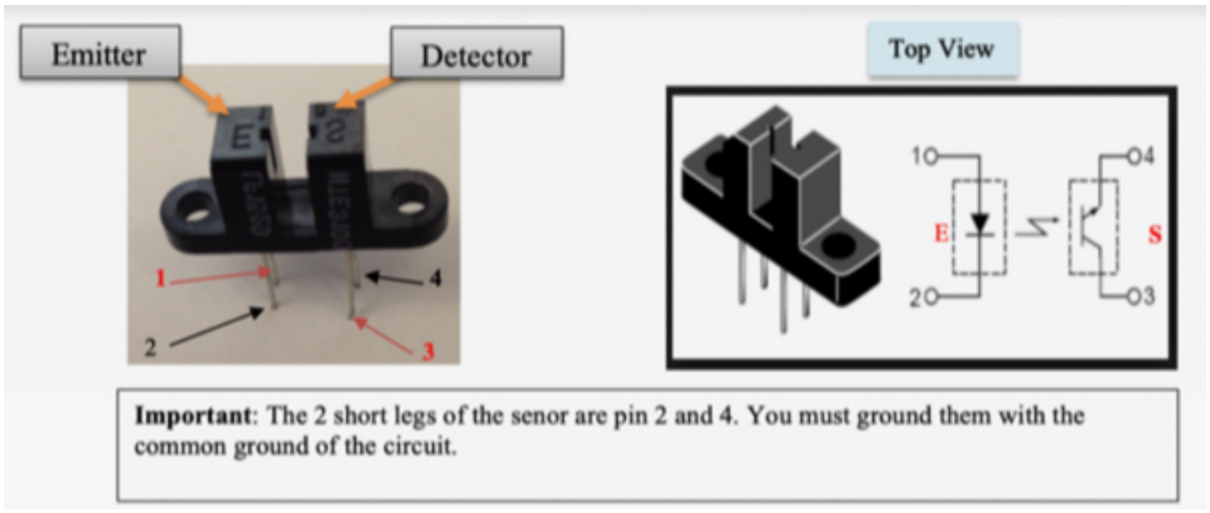

wood or cardboard taped to the bottom of servo behind the two screw slots

Fig. 3. Stable mount of the base of the motor.

Optical Switch

First, we want to test the optical switch to see that it works. This will be done using an LED to see when the switch transitions from high to low and low to high. I do this if you don’t have a meter to measure the output voltage of the switch. Let’s first look into the optical switch.
The device shown in Figure 4 is an opto-interrupter or optical switch, which has an air channel between the IR light emitting diode) noted with the symbol “E” in Figure 4 and in our case the receiver side on your switch has an “S” on it which is the IR detector transistor, see Figure 4. An opaque object passed between the diode and the detector causes the transistor to turn off thus ‘interrupting’ the IR light beam and thus the devices output current. We can tie the output of the optical switch to an external interrupt pin on the AVR board (our case PB3) and detect the interruption. You will have to make an opening cut in the wheel in Figure 3 for this device to work (when you cut out the slot, screw of hot glue or tape to the servo motor horns If you rig up the motor base as shown in Fig. 4 so that the wheel spins thru the slot in the opto-interrupter, each time the opening passes; the output of the optical switch turns on and back off when the slot has passed. If we write our software so that a voltage change on the pin attached to the opto-interrupter causes an interrupt in the AVR processor, we can count those interrupts. If we count for exactly one second we have the number of times the wheel rotates per second, which is the rotational speed in Hz. Cool!





Important: The 2 short legs of the senor are pin 2 and 4. You must ground them with the common ground of the circuit.

Figure 4: Opto-Interrupt Switch - H21A1

The schematic for the external connections is shown in Figure 5. We need to use a 300Ω to pin 1 of switch to Vcc=3.3v from the AVR board and a 2.2kΩ from pin 3 to VCC.

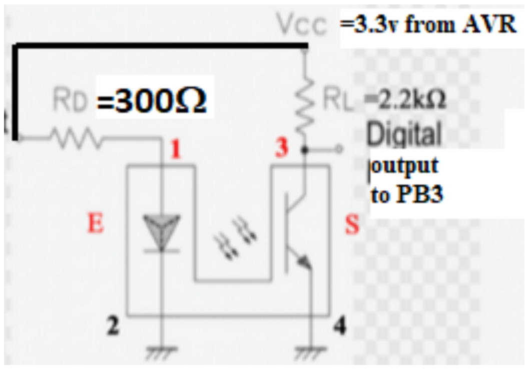
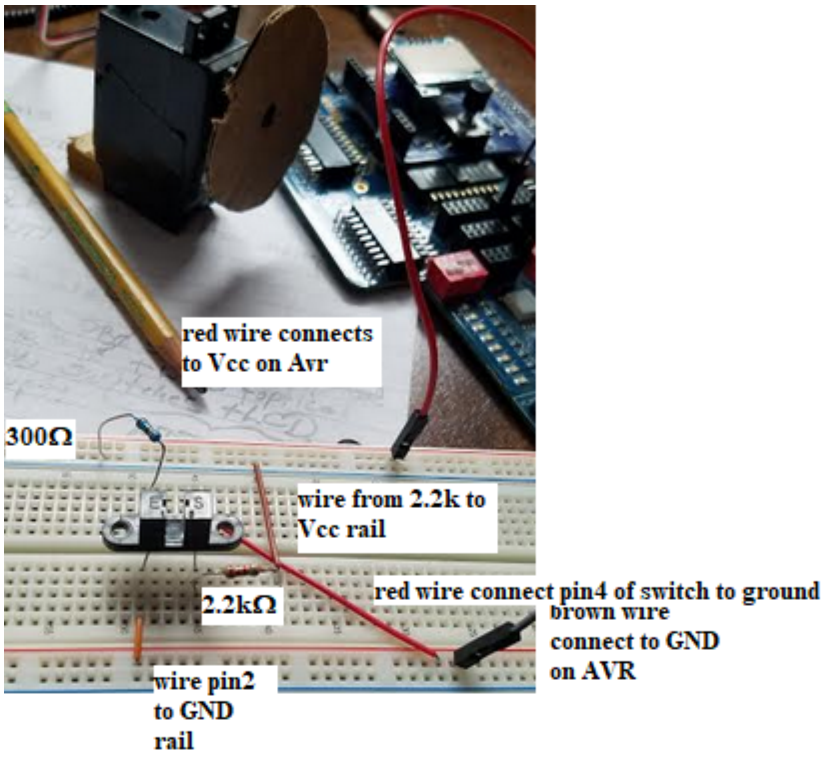


Fig. 5 Electrical schematic connecting the resistors, VCC and ground.

Now, you need to connect the opto-interrupter IC to the bread board. Make sure you use both "sides" of the breadboard so as to not to short out either transmitter pins or output pins as shown in Figure 6 and a top view is shown in Figure 7. You must connect the **Anode** pin with a **current limiting resistor** (300-1000 ohms) to prevent damaging the IR led. **Make sure you do this part or you will fry the IR transmitter side of the switch.** The output of the switch will be connected to any input pin PB3 of the AVR board similar. Make sure you enable pull up resistor for PB3 bit in your code.


red wire connects to Vcc on Avr
300Ω
wire from 2.2k to Vcc rail
red wire connect pin4 of switch to ground
2.2kΩ
brown wire connect to GND on AVR
wire pin2 to GND rail
Figure 6: Breadboard layout of the optical switch connections to AVR board.

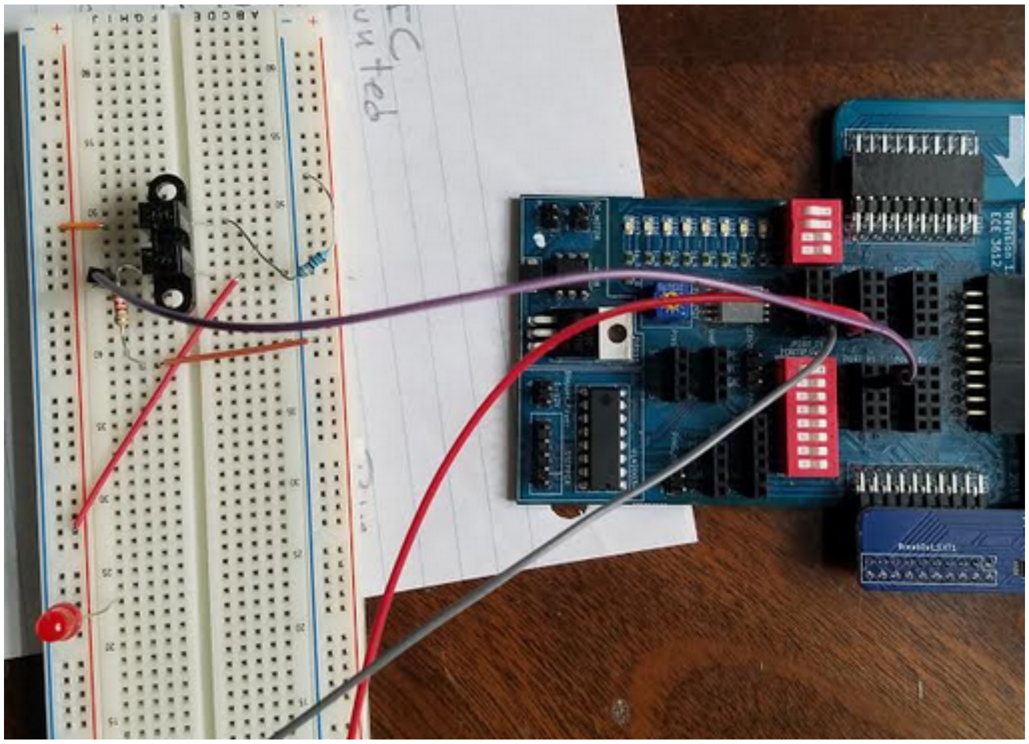


Figure 7: Breadboard layout of the optical switch connections to AVR board.

After we have wired up the bread board, we can now test it with an LED attached between pin3 of the optical switch to ground on the breadboard. When the connections to the Vcc and GND of the AVR to the rails of the breadboard the switch is now powered up and ready to test. With nothing in between the slot of the switch the LED is off as shown in Figure 8.

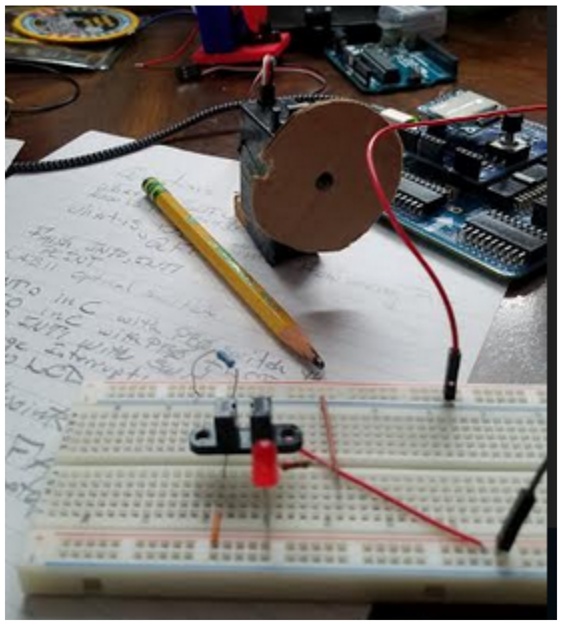


Fig. 8. Nothing in slot of optical switch and the LED is off.

Now place a piece of paper in the slot and hopefully the LED lights up as shown in Figure 9. If this happens you are good to go!

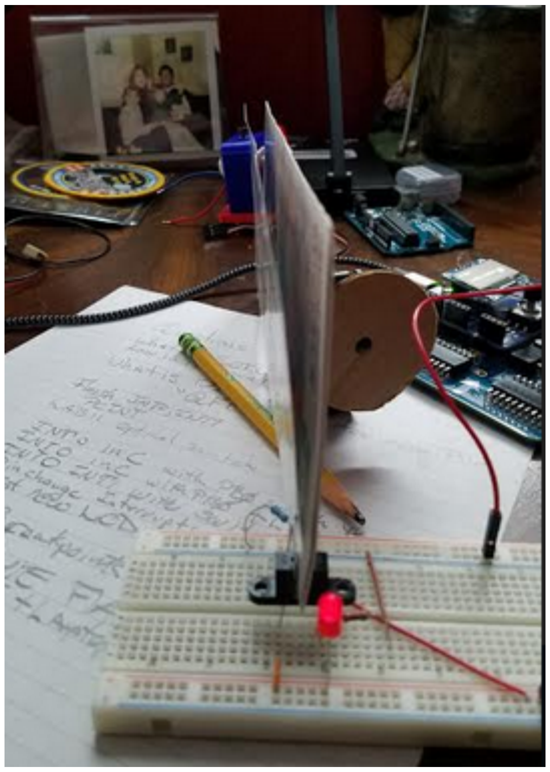


Fig. 9. Paper in slot of optical switch and the LED is on.

If all went well you can now mount the servo motor and the wheel so that the wheel lines up in the optical switch's slot as shown above in Figure 3. Now the fun begins with coding up the AVR with 1) servo software to turn the motor at a constant rate (you can use the internal _delay_ms() functions to do this very simply like on slide 21 of chapter 9 timer1 slides, 2) interrupt software to count pulses on PB3 (similar to slide 22 or 23 of chapter 10 section 10.3 slides and make sure to change INT0 to INT1, 3) run a timer for 1 second count the pulses and display on the LCD. Good luck!

Report and Performance Grading

The lab final will be graded in two parts: Report (50%) and Performance Presentation (50%). The lab final report must include the full report format's sections; Introduction, Method and Procedure, Result (Result and Verification), Conclusion, and Appendix (CODE).

- **Total points for the lab final**: 200 points
- **Use the lab final worksheet (inside the lab final folder on Canvas)**
- **Due date**: Friday 5/1/20 11:59pm (Section 02 and Section 03)

Part I. Report (100 points)

1. Introduction (10 points)

- Objective
- Background Information

2. Method and Procedure (40 points)

- Equipment and Materials
- Hardware – circuit, connections of components, operation of each device
- Software – flowchart and key code blocks with explanation

3. Result and Verification (40 points)

Show the result (eg. screenshots, pictures) with the appropriate labels and verify the result. Explain the method to prove whether the result is correctly working or not.

- Motor running – PWM
- Sensor (Op-to switch) – reading the signal
- Counting and displaying
- Calculation and display of RPM on displaying device

4. Conclusion (10 points)

- Achievement of the objectives
- Discussion in procedure or result
- Summary of the project

5. Appendix

- Full code with full comments

Part II. Performance

Show the code and system is running in the video. Explain the procedure and result.

1. Motor and Sensor (85% points)

Show the following operations;

- motor running
- sensor reading
- displaying the sensor's reading (counting the input)

2. Calculating and Displaying (15% points)

Show the following operations;

- calculating the RPM
- displaying RPM

Lab Final Rubric

Part	Section	Points	Earned Points	Comments
I. Report	Introduction	10		
	Method and Procedure	40		Flowchart (10 pts), Equipment and Components (10pts), Hardware (10pts), and Software(10pts) Description
	Result and Verification	40		Show the results and verify them with your own method. • Motor running – PWM (10pts), • Sensor (Op-to switch) – reading the signal (10pts) • Counting and displaying (10pts) • Calculation and display of RPM on displaying device (LCD) (10pts)
	Conclusion	10		Summary and Discussion
	Appendix	0		Code with full comment IMPORTANT: **If you do not put your code**, no credit will be given for your project)
Subtotal		100		

II. Performance (video)	Motor and Sensor	85		Motor running (25pts), sensor reading (30pts), counted value displaying (30pts)
	RPM	15		Calculating and Displaying RPM (LCD)
Subtotal		100		
Total		200		
% of Total		100		