

Printer Carriage Motion Control

Learning Objectives

By the end of this laboratory experiment, you should be able to:

- Use a quad push-pull driver chip for bi-directional DC motor control with an Arduino.
- Use reflective photo-switches, opto-interrupter switches, and micro-switches for limit sensing.
- Explain how and why limit switches are used to prevent over-travel of moving components.

Components

| Qty. | Item |
|------|--|
| 1 | Arduino UNO board w/ ATmega328 microcontroller |
| 1 | USB A to B cable |
| 1 | Solderless breadboard |
| 1 | Printer carriage assembly |
| 1 | Optical-interrupter switch (Fairchild H21LTB) |
| 1 | Photo-reflective opto sensor (Fairchild QRB1134) |
| 2 | Roller lever momentary switch (Omron SS-5GL2) |
| 1 | SN754410 (Quad Half-H driver) or L293D (Quad Push-Pull driver with diodes) |
| 1 | 74LS04 hex inverter |

Overview

The purpose of this lab is to help you learn how construct a system with multiple sensors and an actuator, and interface these with the Arduino/ATmega328. Figure 1 shows how all the components come together to complete the printer carriage motion system. The procedure below will guide you in building and testing pieces of the system and then integrating them into a complete system.

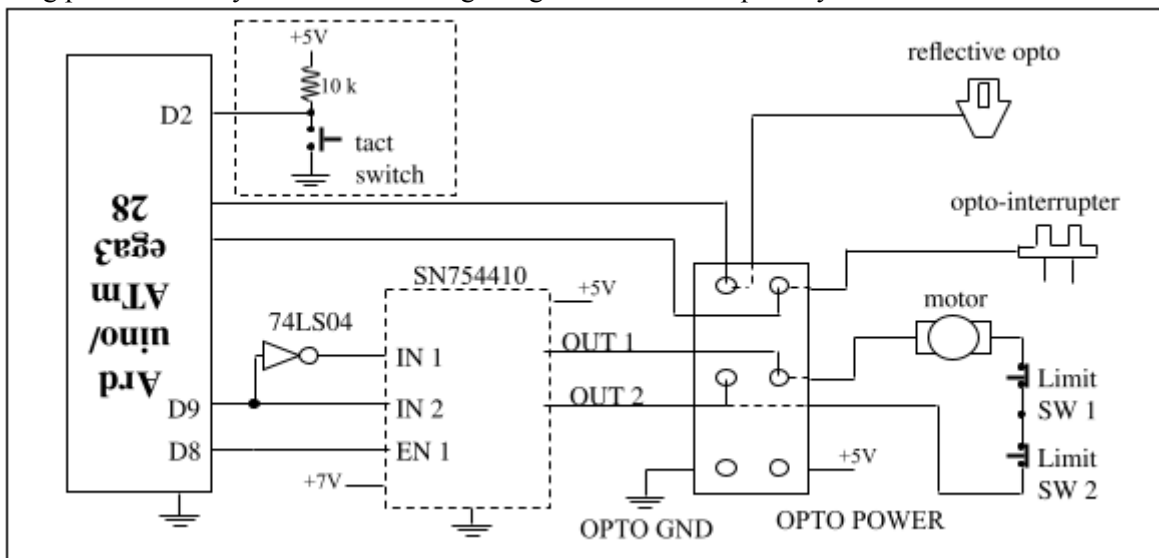


Figure 1. Printer carriage system. The printer carriage system contains: two sensors: a photo-reflective opto sensor and a photo-interrupter switch; a power interface (the SN754410 and 1/6 of the 74LS04); the carriage motor, and two SPST switches to provide fail-safe end-of-travel limits. The connections between the modules and Arduino are made through 5-way binding posts on the printer carriage assembly. The opto sensors and the SN754410 need +5 V to operate, whereas +7 V will be applied to the V_S pin of the SN754410 to power the motor.

Introduction

Switches are common devices that permit or interrupt the flow of current. In addition to simply controlling electrical power to a device (such as a motor or light), they can be used in motion control to detect whether or not a movable element has reached a predetermined position.

Mechanical switches come in a variety of designs. They are categorized by the number of poles and throws they have. The number of poles represents the number of separate circuits that can be completed by the same action of the actuating lever or button. The number of throws represents the number of individual contacts for each pole. The most common types are shown in Figure 2 below.

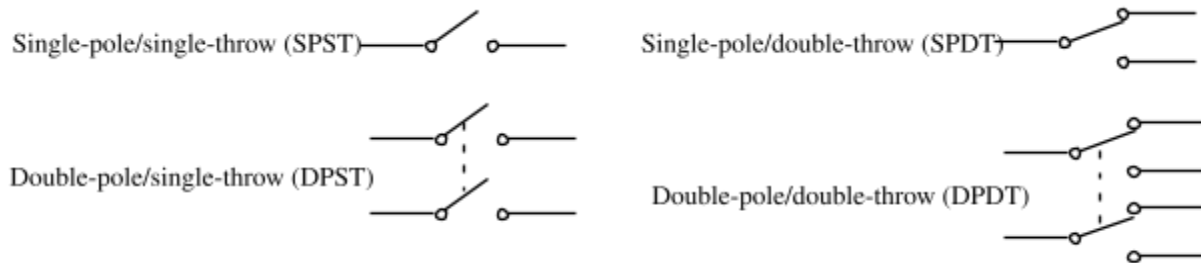


Figure 2. Examples of different switch configurations. The number of poles represents the number of separate circuits that can be completed by the same action of the movable contact(s) inside the switch. The number of throws represents the number of individual contacts for each pole.

Micro-switches typically refer to mechanical switches of small size that have a spring-loaded, momentary contact, operated by a push-button directly or via a pivoted cantilever. Micro-switches find use in many consumer products such as notebook computers, appliances (to detect if a cover is closed), etc. Switches are comprised of moving or sliding mechanical elements are designed to operate for thousands to hundreds of thousands of on-off cycles.

Optical switches (e.g. opto-interrupters, opto-reflectors, etc.) are non-mechanical switches made up of a light-emitting diode (LED) and a phototransistor. As shown in Figure 3, light from the LED shines toward the base of a phototransistor across a gap in the housing of the switch. The output of the phototransistor will indicate whether or not something is blocking the light from the LED. Opto-interrupters are often used in mechatronic devices to indicate that a movable element has reached a specific position or an end-of-travel limit. Devices that use optical switches include printers, copiers, and manufacturing automation systems. Opto-interrupters are attractive to use, because they are solid-state, reliable, relatively inexpensive, and are straightforward to interface with logic circuits. Since they operate without physical contact, they have the advantages of longer lifetime, higher reliability, and faster actuation time compared to mechanical switches.

You are going to use two types of optical switches, a reflective (Fairchild QRB1134) and an interrupter-type (Fairchild H21LTB) on a printer mechanism and use them in driving a printer carriage between preset limits. You will also use two SPST microswitches to provide a fail-safe limit on the end-of-travel of the carriage in case one of the opto-switches fails.

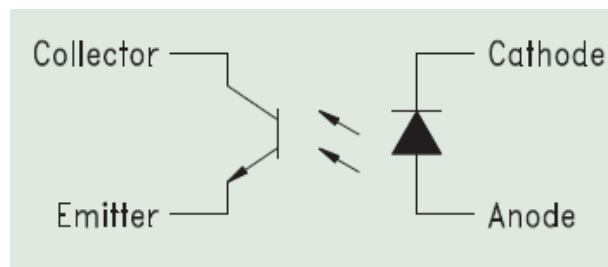


Figure 3. Opto switch schematic. The opto interrupters used on the printer carriage are solid state devices consisting of an infrared-LED and an NPN transistor. When the LED shines on the phototransistor, this causes a current to flow from the base to the emitter. The base-to-emitter current in turn can control current from collector-to-emitter assuming that there is a circuit connected to the collector from a voltage source.

Assembling the System

You will first work with the sensors and make sure that they are working properly. Then you will connect the sensors to the Arduino. Next, you will work with the SN754410 chip, and finally you will pull everything together to complete the printer system.

Get a printer carriage assembly from your lab instructor. The following three sections will guide you through some tests to ensure that all the sensors on the assembly are working properly.

Reflective Opto-switch

Before connecting anything, verify the proper functioning of the reflective opto-switch, which is mounted on a bracket and attached to the side of the printer closest to the motor. Verify that it is operating properly using the procedure below.

1. Set the +6V output terminal on the HP E3630A Triple Output Power Supply to output +5V. Using banana-to-banana test leads, connect the +5V output and the Common output from the power supply to the respective OPTO POWER and the OPTO GND jacks on the printer carriage assembly.
2. Measure the voltage at the binding post that is connected to the output of the *reflective* opto-sensor (the one shaped like an ‘arrow head’) when the aluminum plate on the carriage is in front of its face (this is the ‘blocked’ condition). **(Lab Report Question 1) What voltage do you measure, and what does this voltage represent, logic high or low?**
3. Move the printer carriage, so that the reflective opto-sensor is no longer blocked, and measure its output voltage. **(Lab Report Question 2) What voltage do you measure, and what does this voltage represent, logic high or low?**
4. **(Lab Report Question 3) What should the voltages be in parts 1 and 2?**

If you are satisfied that the opto-reflector is working correctly, go on. If not, figure out what is wrong, and fix it, or ask you lab instructor for help.

Opto-interrupter

Check that the photo-interrupter (the sensor with the big slot in it) that is mounted on a bracket and attached to the side of the printer frame furthest away from the motor, is operating properly.

5. Measure the voltage at the binding post that is connected to the output of the opto-interrupter when the aluminum plate on the carriage is in its slot. **(Lab Report Question 4) What voltage do you measure, and what does this voltage represent, logic high or low?**
6. Measure the voltage at the output when the sensor is unblocked. **(Lab Report Question) What voltage do you measure, and what does this voltage represent, logic high or low?**
7. **(Lab Report Question 6) What should the voltages be in parts 4 and 5?**

If you are satisfied that the photo-interrupter is working correctly, go on. If not, figure out what is wrong, and fix it, or ask you lab instructor for help.

Limit Switches

Move the printer carriage by hand, and make sure that each of the two microswitches with the roller levers at the far right and left limits of travel will be actuated when the aluminum “flag” on the carriage passes beneath them. Use the DMM to check for continuity between the OUT1 and OUT2 binding posts, first with the roller levers from both switches NOT pressed, then with each switch, one at a time, with its roller lever pressed. Continuity should be broken when *either* of the roller levers is pressed. These switches will be used to cut the power to the motor if the carriage travels beyond the opto-switches. Note that it is very important that any device you design with moving components have some means to **reliably** prevent or deal with *over-travel*. Over-travel protection is important to avoid damage to the device and ensure the safety of others working with or near the machine while it is operating.



Connecting the Optical Sensors to the Arduino

Connect the Arduino to the photo-reflective sensor and the opto-interrupter via the appropriate 5-way binding post, **and write a short program that reads the state of each sensor and prints the corresponding logic level to the serial monitor.** You may use any input port and pins you wish (except for PD0 and PD1, which are used for the Serial Monitor). Verify that your program correctly detects the change of state of the switches. Show your lab instructor that your program functions properly before going on. **(Lab Report Question 7) Draw your own schematic, and document which pins you used. Include this code and schematics, and label it appropriately in your lab report.**

74LS04 Hex Inverter

The 74LS04 is a logic IC containing six independent inverters (or *NOT gates*). Figure 4 shows how the chip is laid-out. The output of an inverter simply *inverts* (reverses) the logic level that is presented at its input. So, if logic HIGH is presented at the input, the output of the inverter will be logic LOW.

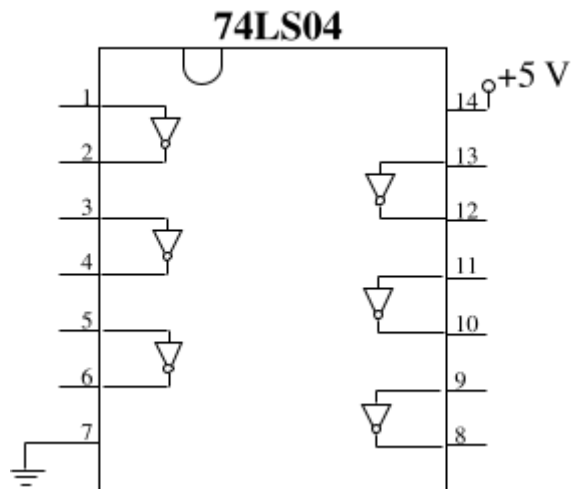


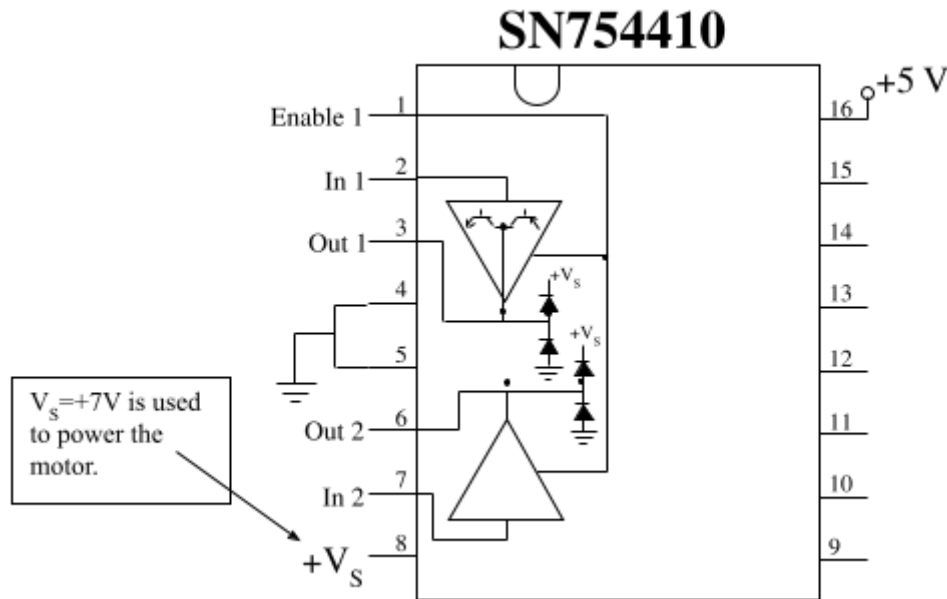
Figure 4. 74LS04 pinout. The 74LS04 is a logic IC containing six *inverters*, also called NOT gates. The schematic symbol for a NOT gate is a triangle with an open circle at the end. Note that you must apply +5V and ground for the 74LS04 chip to function.

IMPORTANT: DO NOT ASSUME any logic IC chip is working when you receive it. You MUST test the logic level on the pins you are working from. For example, observing Figure 4, if you chose to work with pin 1 and pin 2 as your designated inverter, test them by applying +5 V to pin 1 and verify that pin 2 is 0 V, then apply 0 V to pin 1 and verify that pin 2 is +5 V. Don't forget that you need to apply +5V and ground to power the 74LS04 for it to function.

NOTE: The hex inverter's function can also be done entirely by software. By using the *bitwise-not* operator (\sim), you can also reverse the logic level presented at an input, and use it to set the state of an output pin (i.e. $\text{PORTB} = \sim\text{PINA}$; will set all pins on output PORTB to the inverted state of PORTA's inputs).

Quad Half-H Driver

The SN754410 (or L293D) is an IC designed for driving moderate-current inductive loads, such as motors



and solenoids, from *logic* level signals (0 V or 5 V in our case). The chip has four, half H-bridges (also known as ‘push-pull’ channels, because each of the four output can either *source* (‘push’) or *sink* (‘pull’) current), and each pair has an enabling input, and integral clamping (flyback) diodes as shown in Figure 5. Each channel can source or sink up to 600 mA continuous current. A push-pull channel consists of two transistors, a PNP and an NPN, in which the collectors of the two transistors are connected and their base leads are common. When a logic-level signal is applied to the common base, one of the transistors will be saturated (full-on) and the other cutoff (full-off). This arrangement allows the channel to either source (push) or sink (pull) current from the common collector junction, hence the name “push-pull”. If two channels are used, a DC motor can be driven bi-directionally from a power supply of single polarity.

Figure 5. SN754410 (or L293D) connections. The SN754410 is a four-channel push-pull driver chip with integral clamping diodes. Only two of the four channels are shown. The chip needs +5 V applied to pin 16 to operate. +Vs is the voltage of the supply used to drive the motor (use +7 V for Vs). Channels 1 and 2 will be enabled when logic high (+5 V) is applied to pin 1 (Enable 1).

74LS04 and SN754410 connection

Pin 1 of the SN754410 is the *enable* input (abbreviated EN1) for channels 1 and 2 (see Figure 5). When pin 1 is taken to logic high, the *pair* of push-pull channels (IN1/OUT1 and IN2/OUT2) is “enabled”, meaning that channels 1 and 2 are made operational. Thus, if a logic high is applied to pin 1 (In 1), then pin 3 (Out 1) will go ‘high’ (to about 1.4 V lower than V_s). If a logic low is applied to pin 2 (In 1), then pin 3 (Out 1) will go ‘low’ (to about 1.2 V above ground). When pin1 (EN1) is taken to logic low, the two channels are “disabled”, which means that their outputs are effectively disconnected from the circuit.

Connect the microcontroller to the 74LS04 and the SN754410 as shown in Figure 6 using Arduino pins of your choice.

Set the +20V output terminal on the HP power supply so that it will output +7 V (check the voltage BEFORE you make the connection described next). Connect the tact switch to an input pin of the Arduino using an appropriate jumper. **Write a short program (label it appropriately) that looks for the switch to be pressed, enables EN1 of the SN754410, and toggles IN 1 and IN 2 continuously.** With the DMM, verify that the voltages measured at OUT 1 and OUT 2 on the SN754410 alternate between high (+7V) and low (0V) when the state written to IN 1/IN 2 toggles.

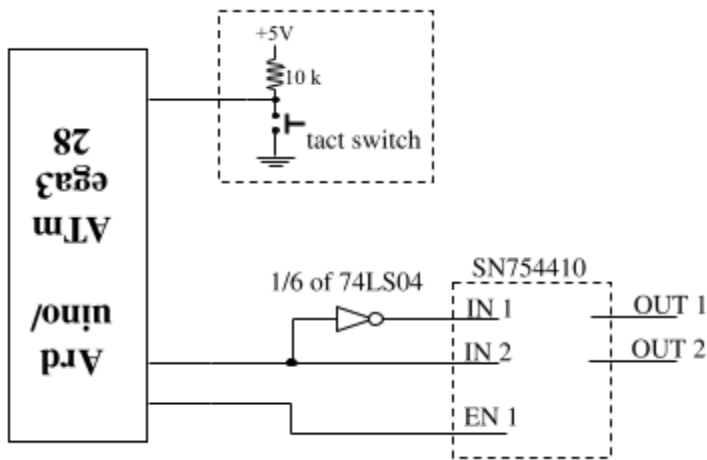


Figure 6. Connections to the Arduino, the 74LS04, and the SN754410 (or L293D). Note that the connections for +5 and Vs are not shown on the SN754410, but need to be connected.

Motor Connection

Assuming that you have gotten all the pieces to work so far, you are now ready to interface the printer carriage motor, and get things moving! Connect the output of the opto-switches to the Arduino if they are not already connected. The general layout of your system should look like Figure 7 below.

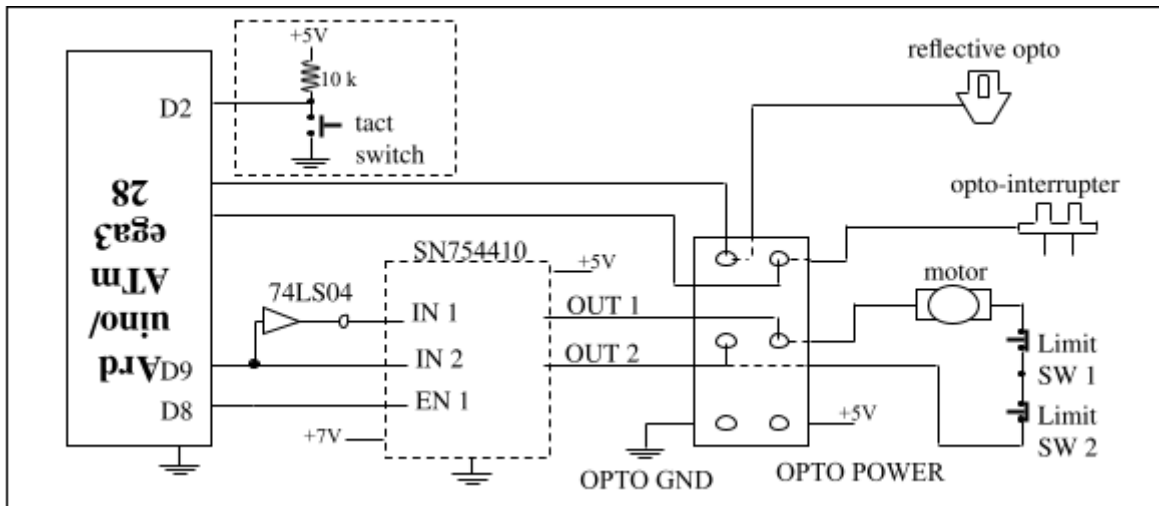


Figure 7. Printer Carriage Motion Controller System. The carriage is driven between limits set by the two opto-switches continuously after the program detects that the tact switch has closed.

Motor Connection

Write a program that will drive the carriage back and forth between the opto-switches when the tact switch is pressed. **Before you jump in and start coding, think about the logic of how you are going accomplish the task! Write a flow chart and develop the logic of your code before you try to move the carriage.** Use your experience from previous labs to fashion your program, and do so by building and testing pieces of your code, rather than trying to write the whole program from start to finish in one shot. (Lab Report Question 8) Write a summary on how your printer works using this code.

IT IS VERY IMPORTANT THAT THE LIMIT SWITCHES FUNCTION PROPERLY. IF YOU DRIVE THE CARRIAGE BEYOND THE OPTO-SWITCHES, MAKE SURE THAT THE LIMIT SWITCHES CUT POWER TO THE MOTOR. Otherwise, the motor, belt, and/or carriage assembly may be damaged if the carriage crashes at the end of the stroke. We don't want you to have to buy us new printer carriage if your break yours!



For more information on the SN754410, see www.ti.com/product/sn754410.

For more information on the 74LS04, look at the data sheets on the Texas Instruments web site:
www.ti.com

