Printer Carriage Motion Control

Learning Objectives

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

- Use a quad push-pull driver chip for bi-directional dc motor control.
- Use reflective photo-switches, opto-interrupter switches, and micro-switches for motion control.
- Explain how and why limit switches are used to prevent over-travel.

Components

<table>
<thead>
<tr>
<th>Qty.</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ATmega16 microcontroller, STK500 board, and serial port cable.</td>
</tr>
<tr>
<td>1</td>
<td>Solderless breadboard</td>
</tr>
<tr>
<td>1</td>
<td>Printer carriage assembly</td>
</tr>
<tr>
<td>1</td>
<td>Optical-interrupter switch (Fairchild H21LTB)</td>
</tr>
<tr>
<td>1</td>
<td>Photo-reflective opto sensor (Fairchild QRB1134)</td>
</tr>
<tr>
<td>2</td>
<td>Roller lever momentary switch (Omron SS-5GL2)</td>
</tr>
<tr>
<td>1</td>
<td>SN754410 (Quad Half-H driver) or L293D (Quad Push-Pull driver with diodes)</td>
</tr>
<tr>
<td>1</td>
<td>74LS04 hex inverter</td>
</tr>
</tbody>
</table>

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 ATmega16 microcontroller. Figure 1 shows how all the components come together to complete the printer 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 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 the STK500 are made through 5-way binding posts on the printer carriage assembly. The opto sensors and the SN754410 need +5 V to operate, and +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.

![Switch Configurations](image)

**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. (What type of switch are the buttons on the STK500?)

Opto-switches (e.g., opto-interrupters, etc.) are non-mechanical switches that use a light-emitting diode (LED) and a phototransistor. 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 its desired position or end-of-travel. Examples include printers, copiers, and manufacturing automation. 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 opto-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.

**Assembling the System**

Like we have done in past experiments, we will again construct the system in a *modular* way by building and testing *pieces* of the system as we go. This is a good general approach in any kind of electronic work, system design, or even computer programming. Avoid the temptation to wire everything up first and hope that it will all work the first time. Such an approach is doomed
to fail, and you’ll end up spending more time trying to figure out what is not working than if you just build and test subsystems as you go along.

First you will work with the sensors, and make sure that they are working properly; then you’ll connect the sensors to the ATmega16; then you’ll work with the SN754410 chip; then you will pull everything together to complete the printer system.

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

**Reflective Opto-switch**

First, before connecting anything, check that the reflective opto-switch, which is mounted on a bracket and attached to the side of the printer closest to the motor, is operating properly using the procedure below.

1. Set the HP E3630A Triple Output Power Supply to output +5V and +7V. 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 (which is shaped like an ‘arrow head’) when the aluminum plate on the carriage is in front of its face (this is the ‘blocked’ condition). **What voltage do you measure, and what does this voltage represent, logic high or low?**

3. Move the printer carriage, so that the reflective optosensor is no longer blocked, and measure its output voltage. **What voltage do you measure, and what does this voltage represent, logic high or low?**

4. **What should the voltages be in parts 1 and 2?**

If you are satisfied that the reflective opto-switch 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 one 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.

4. 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. **What voltage do you measure, and what does this voltage represent, logic high or low?**

5. Measure the voltage at the output when the sensor is unblocked. **What voltage do you measure, and what does this voltage represent, logic high or low?**

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 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 unpressed, 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. 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 the machine or in its near vicinity.

**Connecting the Optical Sensors to the ATmega16**

Connect the ATmega16 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 port. You may use any input port and pins you wish (except for PD0 and PD1, which are used for the serial port). If you need help remembering how to setup and use the serial port, refer to your old labs. 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. 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 3 shows how the chip is layed out. The output of an inverter or NOT gate simply “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 at logic low.

![74LS04 pinout](image)

**Figure 3.** 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 to function.

**IMPORTANT:** DO NOT ASSUME any logic IC chip is working when you receive it. You MUST test the logic level on the pin-outs you are working from. For example, observing Figure 3, if you chose to work with pin 1 and pin 2 as your designated inverter, test them by applying +5 Volts to pin 1 and verify that pin 2 is 0 Volts, then apply 0 Volts to pin 1 and verify that pin 2 is +5 Volts. Don’t forget that you need to apply +5V and ground for the 74LS04 to function.

**NOTE:** The hex inverter’s function can also be done entirely by software. By using the ‘not’ operator (~), you can also reverse the logic level presented at an input, and use it to set the state of an output pin (i.e. PORTB = ~PINA; will set all pins on output port B to the inverted state of corresponding port A inputs).
Quad Half-H Driver

The SN754410 (or L293D) is an IC designed for driving inductive loads, such as motors and solenoids from logic level signals. 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 4. 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 and bases of the two devices are connected. When a logic-level signal is applied to the common base, one of the transistors will be saturated and the other cut-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 4. SN754410 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 the channels 1 and 2 (see Figure 4). When pin 1 is taken to logic high, the pair of push-pull channels (IN1/OUT1 and IN2/OUT2) is “enabled”, meaning that they 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 Vs). If a logic low is applied to pin 2 (In 1), 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 5.

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 one of the tact switches provided on the STK500 interface board to an input pin of the ATmega16 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 (by toggling the bit**
written to PB1) continuously. Verify that the voltage measured between OUT 1 and OUT 2 on the SN754410 reverses polarity when the bit written to PB1 toggles.

Figure 5. Connections to the ATmega16, the 74LS04, and the SN754410 (or L293). Note that the connections for +5 and Vs are not shown on the SN754410, but need to be connected. Also, the switch and resistor shown is already on the STK500. You just need to jumper the switch pin to PA0 on the microcontroller.

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 optoswitches to the ATmega16 if they are not still connected. The general layout of your system should look like Figure 6 below.

Figure 6. 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.

**IT IS EXTREMELY 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.** 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 if your break this one!

For more information on the SN754410, see [http://focus.ti.com/docs/prod/folders/print/sn754410.html](http://focus.ti.com/docs/prod/folders/print/sn754410.html).

For more information on the 74LS04, look at the data sheets on the Texas Instruments web site: [http://www.ti.com](http://www.ti.com)