MTRE 2610 Engineering Algorithms and Visualization – Dr. Kevin McFall

Laboratory – Controlling motor motion

Introduction

Work continues this week on the color sorting robot depicted in Figure 1. The completed system involves all the major elements listed below:

A. Motor to rotate arm
B. Switch for rotary encoder
C. Limit switch to determine home position
D. Air compressor
E. Solenoid valve for generating suction
F. Solenoid valve for lowering arm
G. Piston-cylinder driving vacuum
H. Piston-cylinder delivering vacuum
I. Piston-cylinder to lower arm
J. Light source for part detection
K. Phototransistor for part detection
L. Analog color sensor
M. Cable channels
N. Proprietary Fischertechnik microcontroller to be replaced with an Arduino Uno

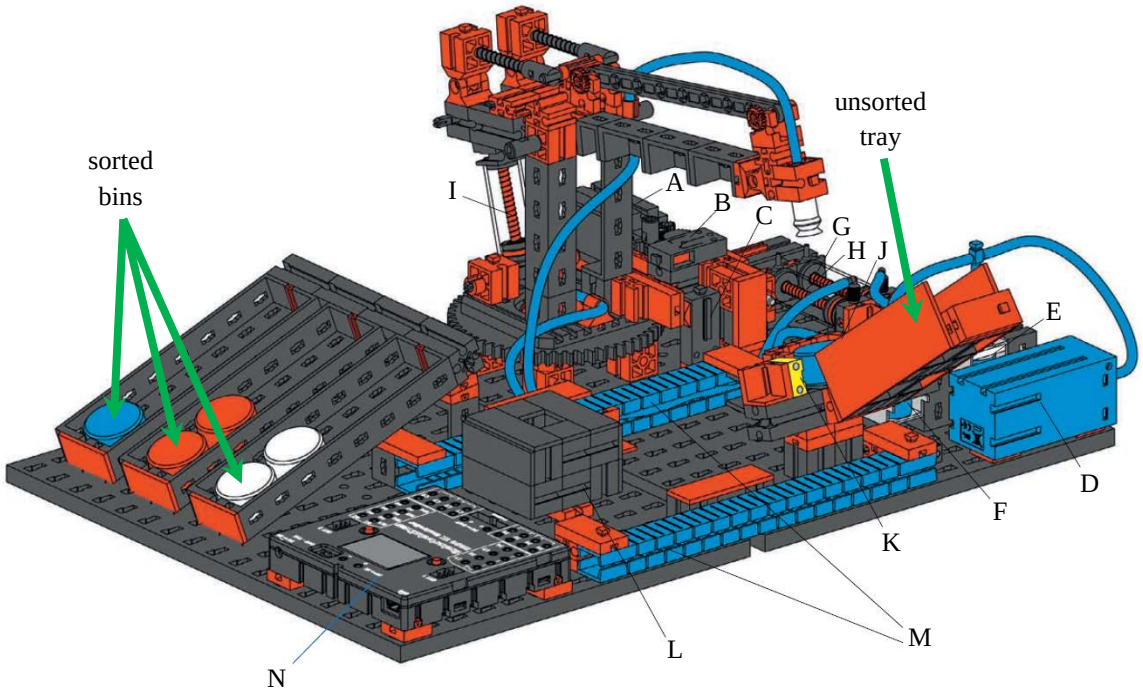

Figure 1: Schematic of completed color sorting robot including major elements.

This laboratory exercise deals with programming precise movements of motor element A. Motion will be controlled using the limit switch C and rotary encoder B to move between the unsorted tray, color sensor L, and sorted bins.

Important modification

The hoses and stop in Figure 2 should still be removed after completing the previous laboratory. If they have been reattached since then, disassemble them again to allow complete rotation of the robot arm. The stop will be used later, but only after confirmation that limit switch C will disengage the motor.

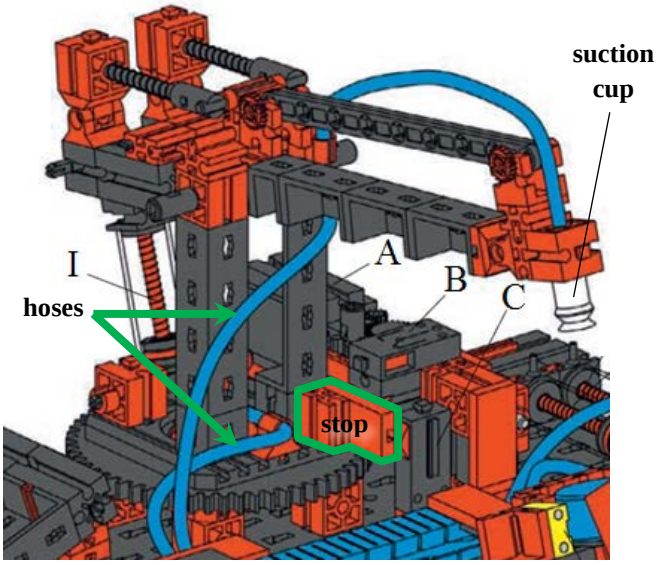

Figure 2: Parts to remove before attempting to operate arm rotation motor.

Fischertechnik pushbutton switches

Switches B and C are identical, although will be used for different purposes. Switch B is part of the rotary encoder and C acts as a limit switch to identify the robotic arm's home position. In any case, both switches are connected to the Arduino in the same manner. Although the switches, depicted in Figure 3, have three terminals, only two are necessary. Ground is connected to terminal 1 and either terminal 2 or 3 is used as the other side of the switch, depending on whether a normally close or open switch is desired, respectively. Together with a pull-up resistor, the value of the pressed switch can be changed by choosing terminal 2 or 3 rather than switching to a pull-down resistor as in the first laboratory exercise.

Figure 3: Fischertechnik pushbutton switch with its three terminals.

Rotary encoder

Limit switch B is installed adjacent a gear which rotates as the motor moves. The resulting switch signal is a square wave where the switch signal alternates between true and false as the robotic arm turns. The arm turntable is a gear with 58 teeth. It is driven by a 10-tooth gear on the underside of the motor assembly, which is directly connected to the 4-tooth gear pressing on the switch. When the 58-tooth gear traverses one revolution, the interlocking 10-tooth

gear will experience 5.8 turns, as will the 4-tooth gear pressing on the switch. As a result, the switch will be pressed 5.8×4 = 23.2 times for every revolution of the robotic arm. The switch will therefore be pressed every 360°/23.2 = 15.5° of arm rotation. This is not especially fine resolution for controlling exact angular position of the arm. Resolution could be improved with more than 4 teeth on the gear, but the number of teeth is constrained by the physical size of the switch mechanism. One way resolution can be improved is by recognizing that while the switch is pressed 23.2 times per revolution, it is also released 23.2 times as well. If falling edges (switch value dropping from high to low) are counted as well as rising edges (switch value jumping from low to high), 23.2×2 = 46.4 counts will occur per revolution, translating to a resolution of 7.76° per count. Such resolution is sufficient for this exercise, as the position of elements on the plastic base is somewhat adjustable.

Laboratory exercise procedure

First, confirm operation of the encoder switch by taking the final code from the last laboratory and having the motor switch direction for every 5 clicks of the encoder switch rather than the pushbutton used last time. Remember, the encoder switch is like any other pushbutton in that it requires a pull-up resistor

Adjust the time threshold as necessary since the bounce duration of the encoder switch may be different than the previously used pushbutton. Using the count variable modified by the interrupt, have `loop` change motor directions every 46 encoder edges (both rising and falling), which should move the arm one full rotation before changing direction.

Now that rotation of the arm can be measured, it is time to consider reintroduction of the stop. While the *amount* the arm rotates is now known, the *location* of arm is still indeterminable. To resolve this type of problem, a limit switch is typically used to identify when a machine is in its “home” position. Program a new sketch so that the arm always begins rotating in the counter-clockwise direction, stops once limit switch C is pressed, and continues counter-clockwise when the switch is released. Be sure to appropriately connect leads from limit switch C to ground and its pull-up resistor. Perform this task by manually pressing the limit switch, i.e. *without* reattaching the stop. **Only introduce the stop once proper operation is confirmed.** Notice that the machine is designed so that the arm is directly over the unsorted tray when reaching the home position. Always be prepared to pull out a lead the 9 V battery in case the motor is unexpectedly constrained by the stop and does not halt; running motors in a stalled state can permanently damage them.

The real work begins now that both position and rotation amount can be determined, at least within a tolerance of 7.76°. The motor will be programmed to cycle between several modes. First, the arm will return to the home position, pause for a short time, and then enter the next mode moving to the color sensor L. Pause again before reverting back to the mode returning home. One method for coding operation in the various modes is to declare an integer variable storing the currently activated mode. The loop then can consist of an `if-else` structure with one case for each mode, writing the appropriate values to output pins depending on the current mode. Inside each case, the mode number would change whenever the trigger for entering the next mode occurs. For example, the homing mode continues until the limit switch is depressed. To begin with, only two modes are necessary but more will be added later.

A skeleton of the code for moving through the modes is:

```
int mode;
void setup() {
  mode = 0;
}
void loop() {
  if(mode == 0) {
    // Do stuff for mode 0
    if(???) { // Condition for changing from mode 0 to 1
        // Do whatever needs to be done before entering mode 1
        mode = 1;
    }
  }
  if(mode == 1) {
    // Do stuff for mode 1
    if(???) { // Condition for changing from mode 1 to 2
        // Do whatever needs to be done before entering mode 2
        mode = 2;
    }
  }
  if(mode == 2) {
    // Do stuff for mode 2
    if(???) { // Condition for changing from mode 2 to 3
        // Do whatever needs to be done before entering mode 3
        mode = 3;
    }
  }
  if(mode == 3) {
    // Do stuff for mode 3
    if(???) { // Condition for changing from mode 3 to starting over at 0
        // Do whatever needs to be done before starting over
        mode = 0;
    }
  }
}
```

Program the sketch to alternate between travelling home and travelling to the color sensor, with a short pause between each mode transition. To do so, determine the correct number of encoder counts for the arm to travel from the unsorted tray to the color sensor. Adjust the physical position of the color sensor so the arm will lower centered directly on the sensor. Be certain to always program the sketch to begin in homing mode.

Finally, add another mode to the sequence travelling from the color sensor to the sorted bins. Create a variable that can be changed to appropriate edge count values so the arm would stop at each of the three sorted bins. Adjust the precise physical positioning of the bins as needed.

Grading rubric

1. 25 points: Rotate arm back and forth one revolution by counting encoder edges with an interrupt
2. 25 points: Arm rotates counter-clockwise when limit switch is not pressed, and stops otherwise
3. 25 points: Modes alternate between homing and travelling to the color sensor
4. 25 points: Mode sequence includes moving to the sorted bins, with an adjustable value for each bin