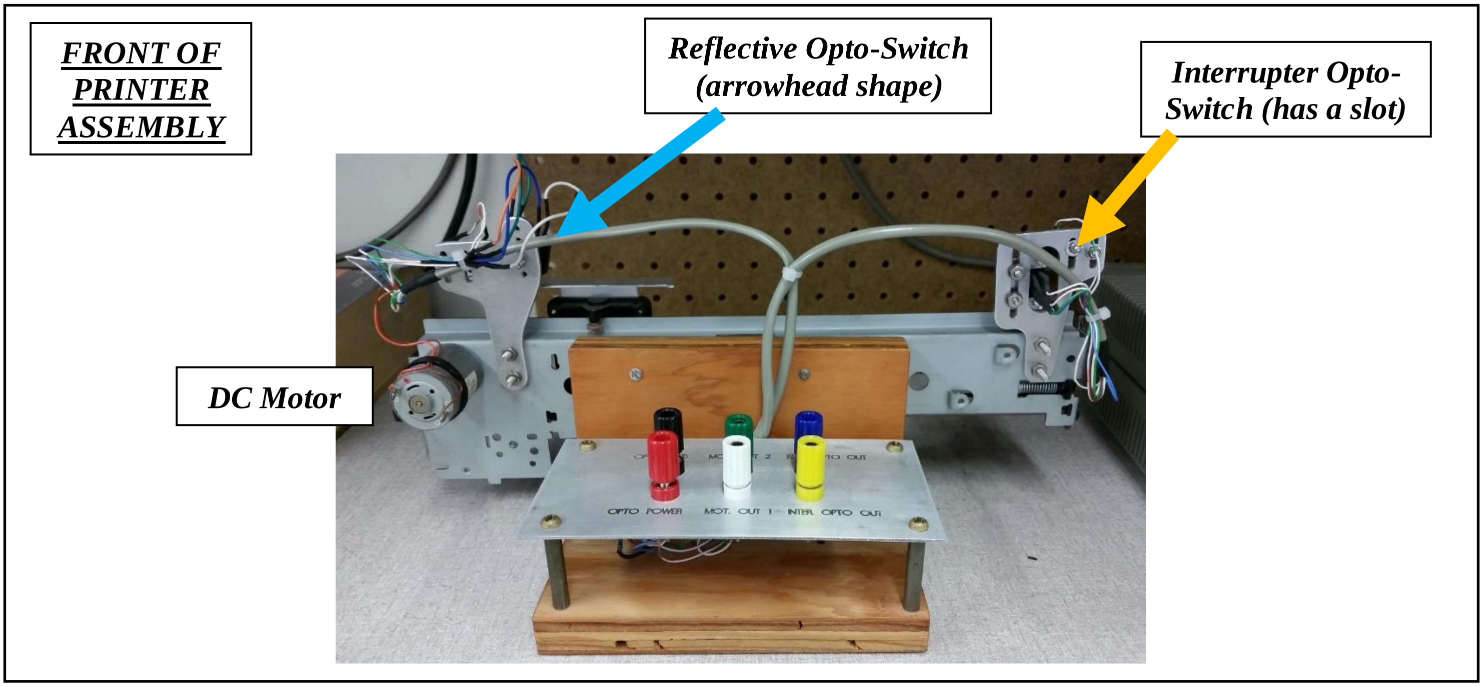
FRONT OF PRINTER ASSEMBLY

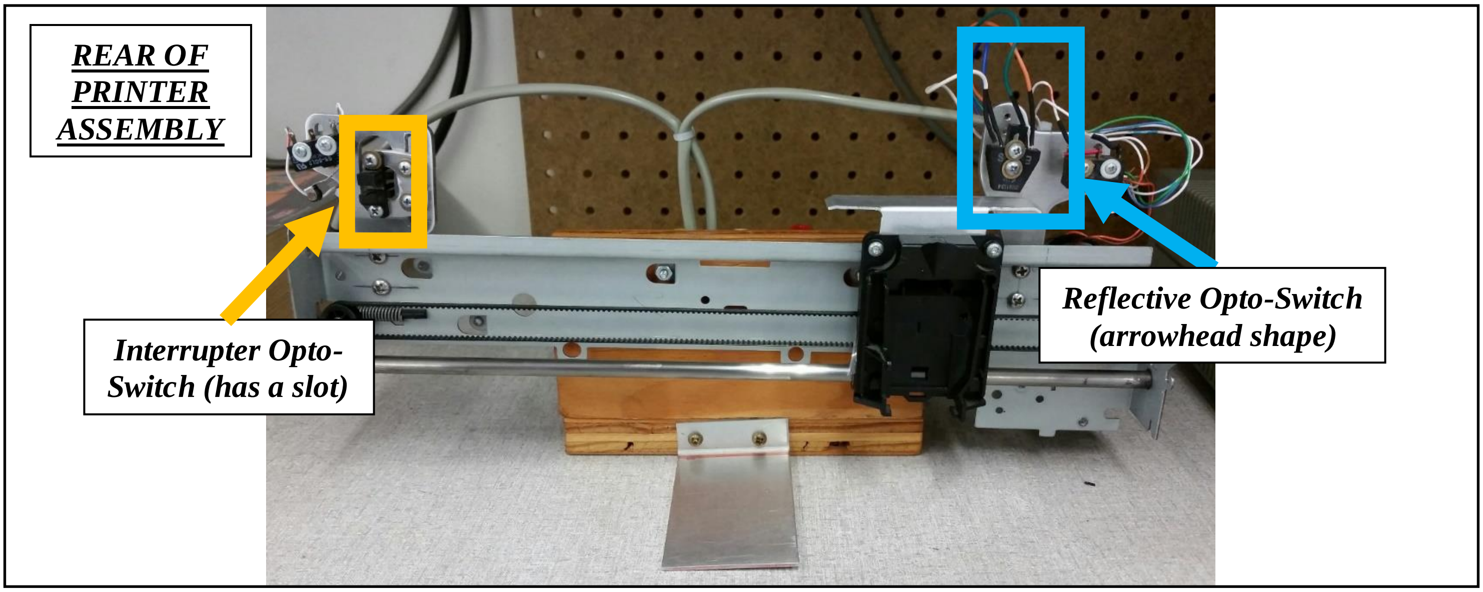


Reflective Opto-Switch (arrowhead shape)

Interrupter Opto-Switch (has a slot)

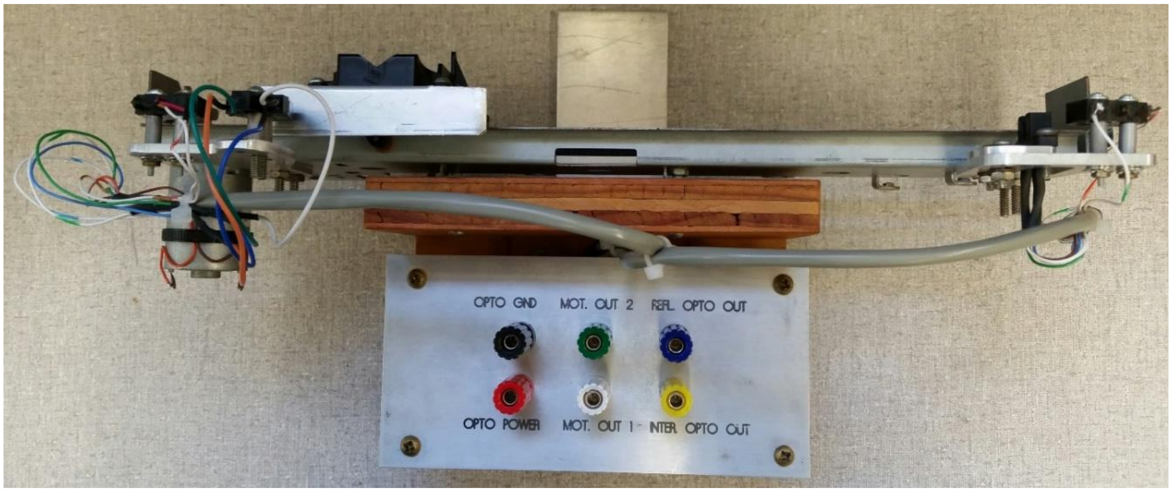
DC Motor

REAR OF PRINTER ASSEMBLY



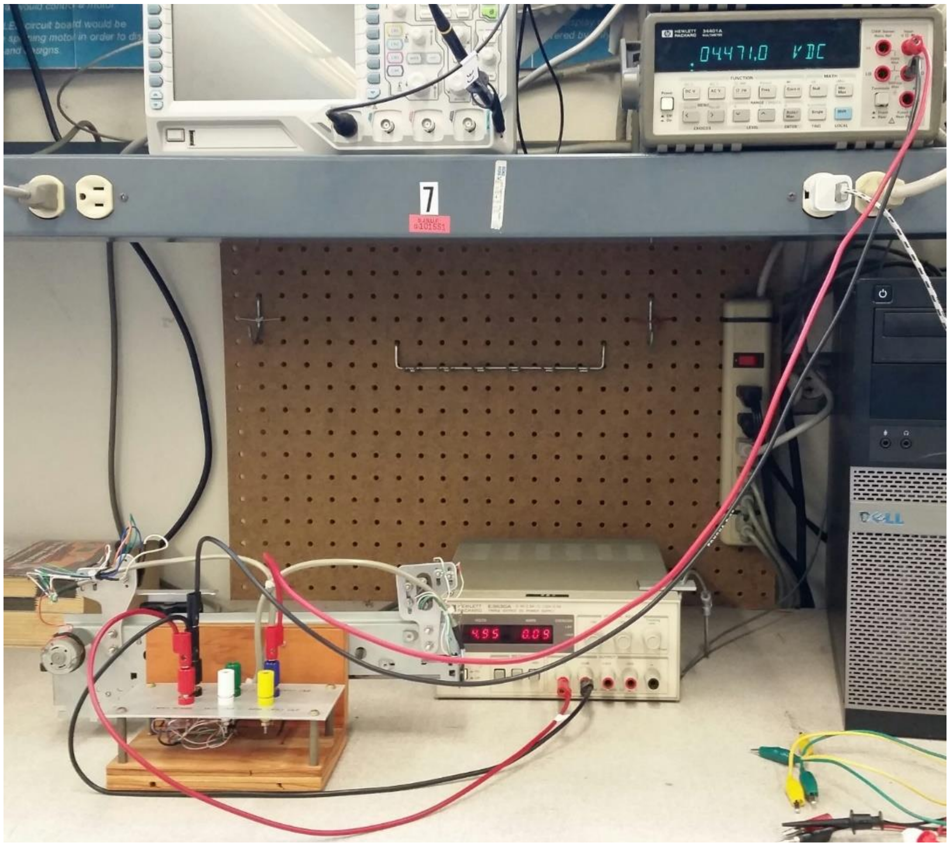
Interrupter Opto-Switch (has a slot)

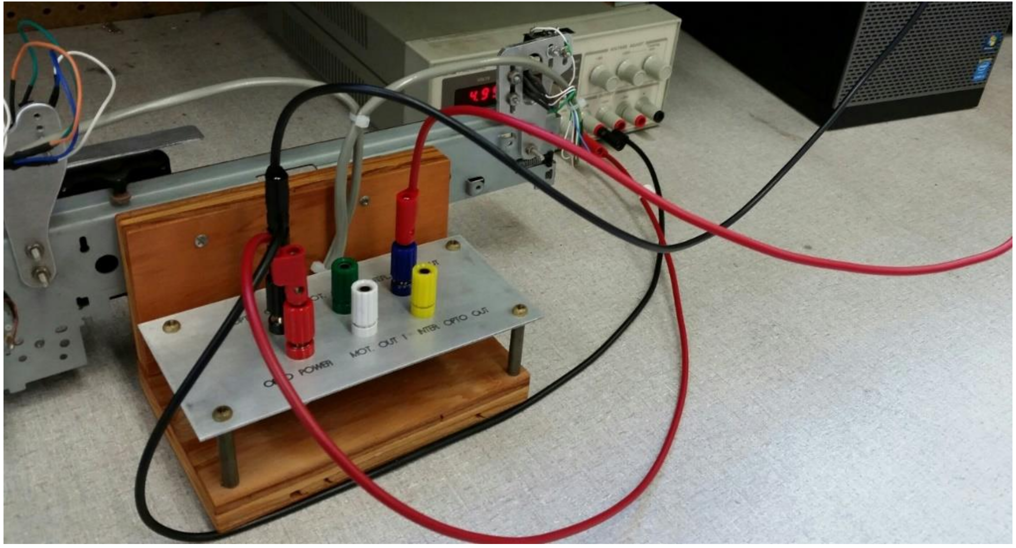
Reflective Opto-Switch (arrowhead shape)



To measure voltage using banana cables:

- Connect the +5 V terminal of the power supply to the OPTO POWER
- Connect the COM terminal of the power supply to the OPTO GND
- Always connect the LO terminal of the DMM to the OPTO GND on the printer assembly, no matter which sensor you are measuring the voltage of; need a common ground reference.
- Connect the HI terminal of the DMM to the appropriate sensor binding post on the printer assembly (blue for reflector opto-switch, yellow for interrupter opto-switch)





Reflective Opto-switch (arrowhead-shaped sensor)

(**Lab Report Question 1**) When printer carriage is in front of reflective opto-switch (arrowhead-shaped sensor), voltage is 4.3739 V (reflective opto-switch's blocked condition). This corresponds to logic high because it is very close to +5 V. The sensor can also be seen as in the ON state because this corresponds to a binary logic 1.

(**Lab Report Question 2**) When printer carriage is no longer in front of reflective opto-switch (arrowhead-shaped sensor), voltage is 11.172 mV (reflective opto-switch's unblocked condition). This corresponds to logic low because it can be estimated to be 0 V. The sensor can also be seen as in the OFF state because this corresponds to a binary logic 0.

(**Lab Report Question 3**) As opposed to analog inputs and outputs, there are only two logic level signals for digital inputs and outputs: logic high is +5 V, and logic low is 0 V (Alciatore & Histand, 2011, p. 227). Therefore, the voltage for Question 1 (when the opto-reflector is blocked) should be +5 V. And the voltage for Question 2 (when the opto-reflector is not blocked) should be 0 V. In terms of binary representations, the state for Question 1 should be binary 1 (ON), and the state for Question 2 should be binary 0 (OFF) (Alciatore & Histand, 2011, p. 215) ("Logic Levels").

Opto-interrupter (sensor with the big slot in it)

(**Lab Report Question 4**) When printer carriage is in the slot of the opto-interrupter, voltage is 1.673 mV. This corresponds to logic low because it can be estimated to be 0 V. The sensor can also be seen as in the OFF state because this corresponds to a binary logic 0.

(**Lab Report Question 5**) When printer carriage is not in the slot of the opto-interrupter, voltage is 3.8597 V. This corresponds to logic high because it is very close to +5 V. The sensor can also be seen as in the ON state because this corresponds to a binary logic 1.

(**Lab Report Question 6**) As opposed to analog inputs and outputs, there are only two logic level signals for digital inputs and outputs: logic high is +5 V, and logic low is 0 V (Alciatore & Histand, 2011, p. 227). Therefore, the voltage for Question 4 (when the opto-interrupter is blocked) should be 0 V. And the voltage for Question 5 (when the opto-interrupter is not blocked) should be +5 V. In terms of binary representations, the state for Question 4 should be binary 0 (OFF), and the state for Question 5 should be binary 1 (ON) (Alciatore & Histand, 2011, p. 215) ("Logic Levels").

To Summarize:

	Blocked	**Unblocked**
Reflective opto-switch	4.3739 V	11.172 mV
Interrupter opto-switch	1.673 mV	3.8597 V

	Blocked	**Unblocked**
Reflective opto-switch	1	0
Interrupter opto-switch	0	1

Limit Switches

To check for continuity, connect the HI and LO terminals on the DMM to the MOT OUT 1 and MOT OUT 2 binding posts on the printer assembly. Press Cont on the DMM.

When the Limit switches are not pressed, there is a connection: between 30 and 40 Ohms of Impedance, which is the connection to the motor. This means neither of the limit switches have been activated and the carriage has not gone past the limit, so the limit switches still allow current to flow.

When either the left roller or the right roller was pressed, the DMM displayed an OPEN (open circuit, OC). When the carriage has gone past either left or right limits, either of the limit switches are activated and create an OC to discontinue current flow and to prevent the carriage from over-travelling.

Connecting the Optical Sensors to the Arduino

(**Lab Report Question 7**) Wrote a program and used Digital Pin 3 to test the state of the reflective opto-switch and Digital Pin 2 to test the state of the opto-interrupter.

```
// I) Variables
/* Pin Assignments: Declare and Initialize Input, Output, and Variables.
 *    The printer carriage's opto-interrupter was connected to Arduino
 *       Digital Pin 2.
 *    The printer carriage's photo-reflector was
 *       connected to Arduino Digital Pin 3.
 */
#define INTERRUPTER 2
#define REFLECTOR 3



// II) setup() is analogous to Constructors
void setup()
{
  pinMode(INTERRUPTER, INPUT);
  pinMode(REFLECTOR, INPUT);
  Serial.begin(9600);
}



// III) loop() is analogous to main()
void loop()
{
  // Creates int's to store the state (logic level) from the digital
  // reading of the sensors
     int interrupterLogicLevel = digitalRead(INTERRUPTER);
     int reflectorLogicLevel = digitalRead(REFLECTOR);

  // Prints the stored logic level onto the Serial Monitor
     Serial.print("Interrupter Logic Level: ");
     Serial.println(interrupterLogicLevel);
     Serial.print("Reflector Logic Level: ");
     Serial.println(reflectorLogicLevel);
}
```

74LS04 Hex Inverter && Quad Half-H Driver CONNECTION

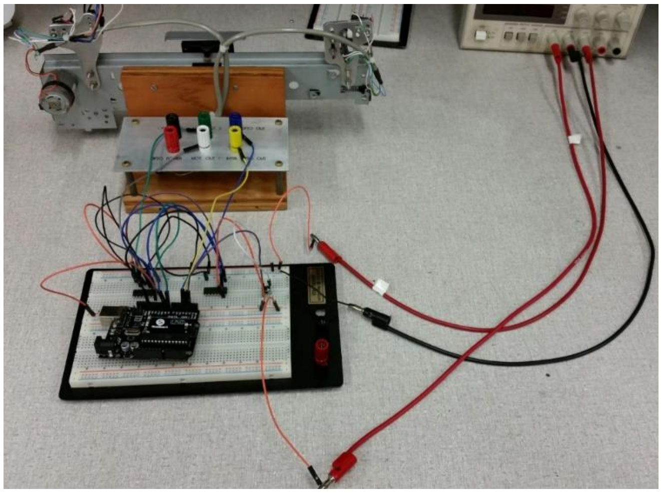
- 74LS04 has one notch at top and two holes: one near top, one near bottom; 14 pins.
- SN754410 has one notch at top and one hole near the bottom; 16 pins.

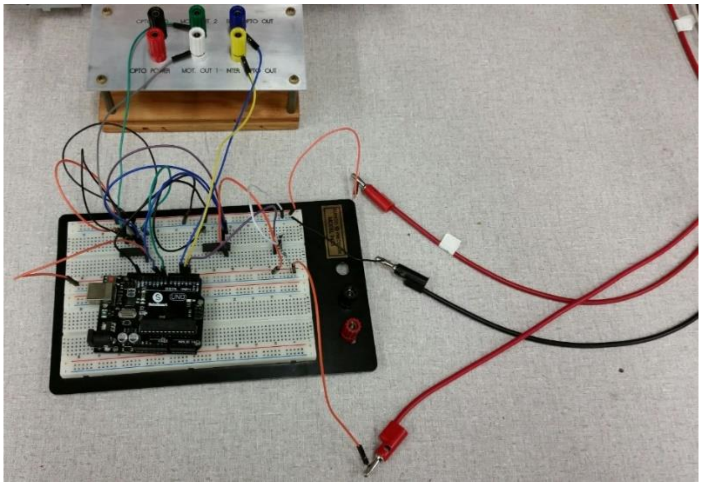
Write a short program (label it appropriately) that looks for the switch to be pressed, enables EN1 of the SN754410, and toggles IN 1 and IN 2 continuously.

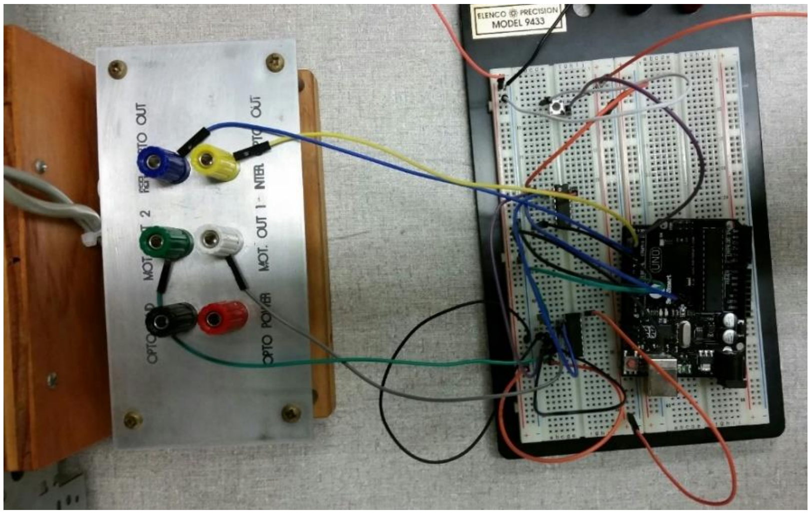
	Blocked	**Unblocked**
Reflective opto-switch	1	0
Interrupter opto-switch	0	1

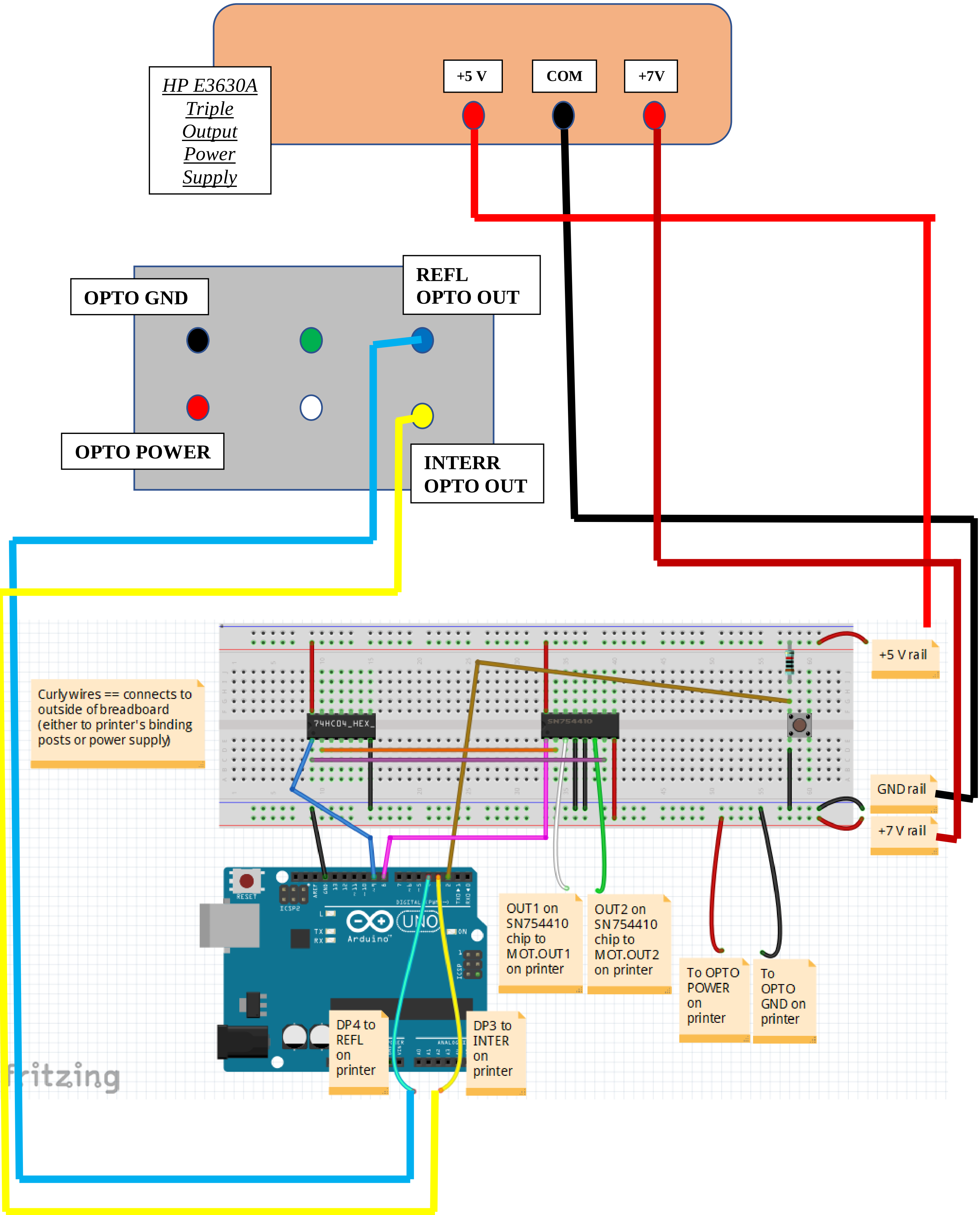
```
#define BUTTON 2
#define ENABLE 8
#define IN 9
#define REFLECT 4
#define INTERUPT 3

void setup()
{
  pinMode(BUTTON, INPUT);
  pinMode(REFLECT, INPUT);
  pinMode(INTERUPT, INPUT);
  pinMode(ENABLE, OUTPUT);
  pinMode(IN, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  while(digitalRead(BUTTON) == LOW)
  {
    motor();
    digitalWrite(ENABLE, LOW);
  }
}

void motor()
{
  digitalWrite(ENABLE, HIGH);
  int reflectorState = digitalRead(REFLECT);
  int interrupterState = digitalRead(INTERUPT);
  delay(10);

  /*  Create a boolean, and assign it
   *   a value of 1 (1 == true)    */
  static bool toggle = 1;

  if(reflectorState == 0 && interrupterState == 1)
  {
    if(toggle)
      digitalWrite(IN, toggle);
    else
      digitalWrite(IN, !toggle);
    toggle ^= 1;    // OR: toggle = !toggle;
    delay(10);
  }
  else if(reflectorState == 1 && interrupterState == 1)
  {
    digitalWrite(IN, HIGH);
  }
  else if(reflectorState == 0 && interrupterState == 0)
  {
    digitalWrite(IN, LOW);
  }
}
```

Motor Connection









Curly wires == connects to outside of breadboard (either to printer's binding posts or power supply)

<u>Motor Connection (Final Exercise) Instructions</u>

Power Supply Connections:

1. Make a +7 V rail on breadboard
2. Make a GND rail on breadboard (suggest making GND rail next to +7 V rail)
3. Make a +5 V rail on breadboard

74LS04 Hex Inverter, SN754410 Quad Half-H Motor Driver, and Printer Assembly Connections:

0. Place 74LS04 and SN754410 in the middle, between two rails, as shown in the *very colorful* Figure above
1. SN754410 Pin 1 (EN1) == connect to Arduino Digital Pin 8 (no rail needed)
2. SN754410 Pin 2 (IN1) == connect to Pin 2 of 74LS04 (Output of Hex Inverter #1)
3. SN754410 Pin 3 (OUT1) == connect to MOT OUT 1 on printer assembly (white binding post)
4. SN754410 Pin 4 (GND #1) == connect to GND rail
5. SN754410 Pin 5 (GND #2) == connect to GND rail
6. SN754410 Pin 6 (OUT2) == connect to MOT OUT 2 on printer assembly (green binding post)
7. SN754410 Pin 7 (IN2) == connect to Arduino Digital Pin 9 (**must make a rail for Digital Pin 9**)
8. SN754410 Pin 8 (+Vs) == connect to +7 V rail
9. SN754410 Pin 16 (+5 V) == connect to +5 V rail
10. 74LS04 Pin 1 (Input of Hex Inverter #1) == connect to Arduino Digital Pin 9's **Rail**
11. 74LS04 Pin 7 (GND) == connect to GND rail
12. 74LS04 Pin 14 (+5 V) == connect to +5 V rail

More Printer Assembly Connections:

1. Reflector opto-switch of printer assembly == connected to Arduino Digital Pin 4
2. Interrupter opto-switch of printer assembly == connected to Arduino Digital Pin 3
3. OPTO POWER of printer assembly == connected to +5 V rail
4. OPTO GND of printer assembly == connected to GND rail

Tactile Switch Button Connections:

0. Also Place the Tactile Switch Button in the middle, between two rails, as shown in the *very colorful* Figure above
1. Connect 10 kOhm resistor from +5 V rail to rail that the button is on
2. Connect the other side of the button to GND rail
3. Connect Arduino Digital Pin 2 in between the 10 kOhm resistor and the button

Double-check the Arduino Digital Pin Connections:

1. Tactile Switch Button == connected to Arduino Digital Pin 2
2. ENABLE of SN754410 Motor Driver IC chip == connected to Arduino Digital Pin 8
3. IN1 & IN2 of SN754410 Motor Driver IC chip == connected to Arduino Digital Pin 9
4. Reflector opto-switch of printer assembly == connected to Arduino Digital Pin 4
5. Interrupter opto-switch of printer assembly == connected to Arduino Digital Pin 3

Motor Connection (Final Exercise) Code

```
#define BUTTON 2
#define ENABLE 8
#define IN     9
#define REFLECT 4
#define INTERUPT 3

void setup()
{
  pinMode(BUTTON, INPUT);
  pinMode(REFLECT, INPUT);
  pinMode(INTERUPT, INPUT);
  pinMode(ENABLE, OUTPUT);
  pinMode(IN, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  static bool toggle=1;
  Serial.print("Sensors.");
  Serial.print(digitalRead(REFLECT));
  Serial.print("   ");
  Serial.println(digitalRead(INTERUPT));
  if(digitalRead(REFLECT) == digitalRead(INTERUPT))
  {
    digitalWrite(ENABLE, HIGH);
    toggle=!toggle;
    Serial.println(toggle);
    digitalWrite(IN, toggle);
    delay(500);
  }
}
```