Practical 3: Incremental Encoder

What is it about?

This week, you will build and fit to your DC motor an optical incremental encoder. Plug it to an Arduino and measure the angular velocity of your motor.

Aims

At the end of the practical, you should:

- have build a basic yet accurate incremental encoder
- have fit the encoder on your motor shaft
- have connected it to an Arduino board and programed the Arduino to read the angular velocity of your motor.

Specific Challenges

- There's a lot to do! Be quick!

Note

In your lab journal, **describe the basic design of each component**, **how it was constructed** and **how it was tested**. Add **pictures** and link to **videos** as needed.
Describe as well **the overall system** and how it performs.

And do not forget: **write your lab journal as a text file using the Markdown syntax** and **push your journal and the pictures on GitHub**.

Part I

Introduction

Principle of operation

This assignment consists of using an LED light source and phototransistor detector. When a slotted disc is placed between the light source and detector and rotated, the circuit will detect each time light can pass from the LED to the phototransistor, as shown in Figure 1. Using an Arduino to count these pulses, it is possible to estimate the rotational position and angular velocity of the disc.

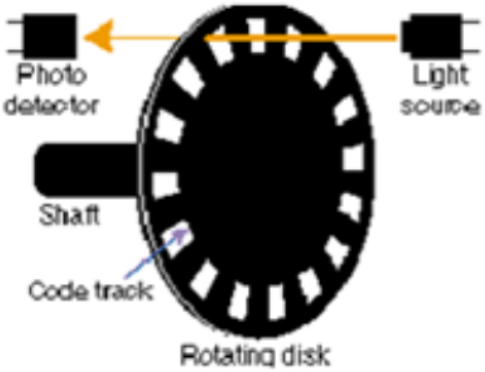


Figure 1: Basic operation of the simple optical incremental encoder. A light source shines light through holes in a disc, which can be detected by a photo detector. As the disc rotates, this gives rise to one pulse per hole.

Step 1 – Circuit diagram

The circuit is very simple and shown in Figure 2. We use an IR LED as a light source and an IR sensitive transistor to detect its light. They should be aligned to the LED shines onto the phototransistor, as illustrated in Figure 3.

Important

Set the power supply to 5V! By doing so, you will not destroy the semiconductors if you accidentally connect it up the circuit with the wrong supply voltage polarity.

You will be provided will all the necessary electronic components: an IR LED, a IR photo transistor, 1K and 10KOhms resistors, connecting wire and a piece of Vero board to solder them onto. You need to use a bench power supply to run the circuit. You will need to use an oscilloscope to check it is working. Figures 3 and 4 provide details of these components.

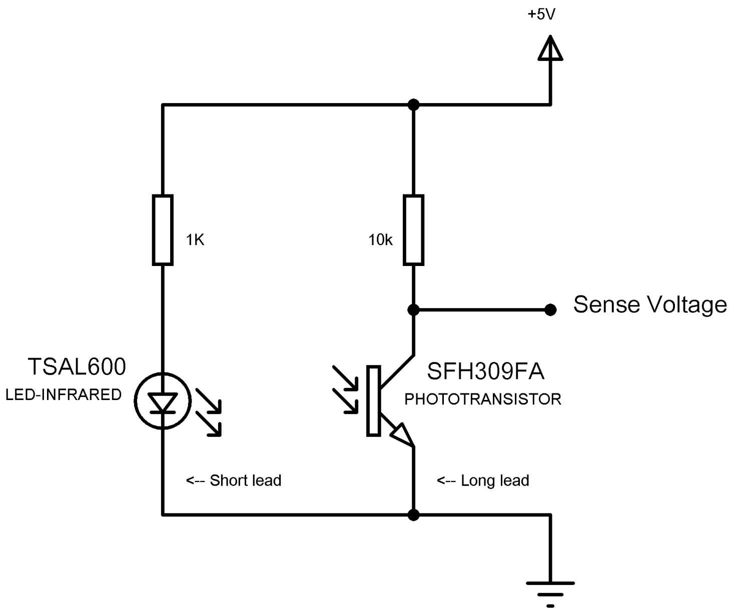


Figure 2: Simple IE circuit. The IR LED provides a light source and a phototransistor detects it. When the light is blocked, the phototransistor switches off.

Note

If needed, use the resistor colour codes provided at the end of the instruction sheet to check your resistors' values.

Build the circuit in stages and test each stage before you progress to the next stage.

Step 2 – Build the IR LED light source

Set the power supply to 5V. Connect the red and black wires to the Vero board to supply the circuit with power. Use a 1KOhm resistor to drive the IR LED from the 5V supply. Note the output light from the IR LED is invisible to the human eye, although some digital cameras can detect it.

Step 3 – Build the light detector

Use a 10KOhms resistor to drive the phototransistor from the 5V supply. Measure the voltage across the phototransistor. When it receives IR light from the IR LED, the phototransistor turns on and it should drop only a

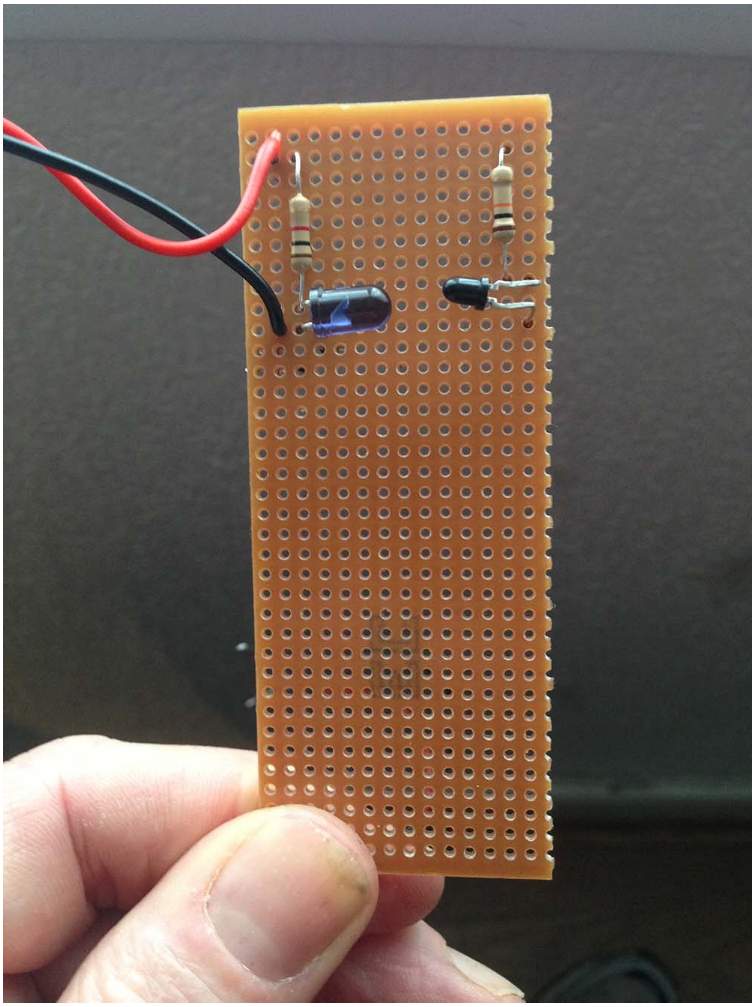


Figure 3: Finished Veroboard circuit. Make sure that the LED and the phototransistor point towards each other and are far enough off the board that you can measure the rotational speed of the rotating disc attached to the motor in Figure 4!

small voltage across the emitter and collector. When the light path is blocked, the transistor will turn off and the voltage across the emitter and collector will rise to almost the supply voltage. By monitoring this voltage it is possible to detect when the light path is blocked. Block the light with a piece of card and check that it is working.

Step 4 – Place an encoder disc on the motor

Cut out a cardboard disc and make a hole in the centre so it fits onto your motor shaft. Attach it with blue tack. Cut a sector so that the disc will allow the passage of light once per revolution (see Figure 4).

Hold the disc between the IR LED and phototransistor while it is rotating. You need to work in pairs to do this. When light is blocked by the passage a disc rotated a motor, an oscilloscope can be used to determine the speed of rotation of motor. How can this be achieved? Run the motor and estimate its speed of rotation in RPM.

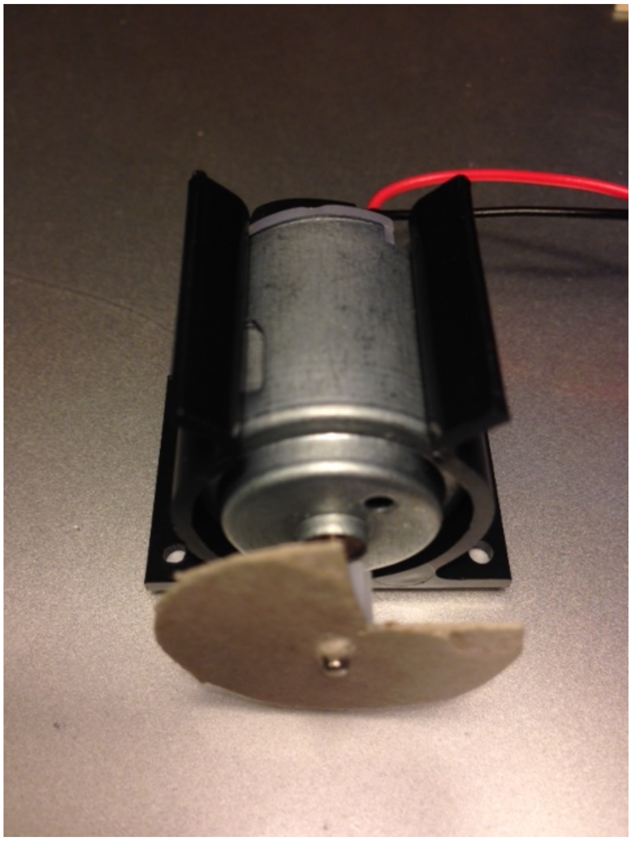


Figure 4: Cardboard disc attached to motor to break light beam from LED to phototransistor. You may need blue tack to attach the disc. Be careful not to cut too much away to make the sector to let light pass or it will no longer attach to the motor spindle!

Note

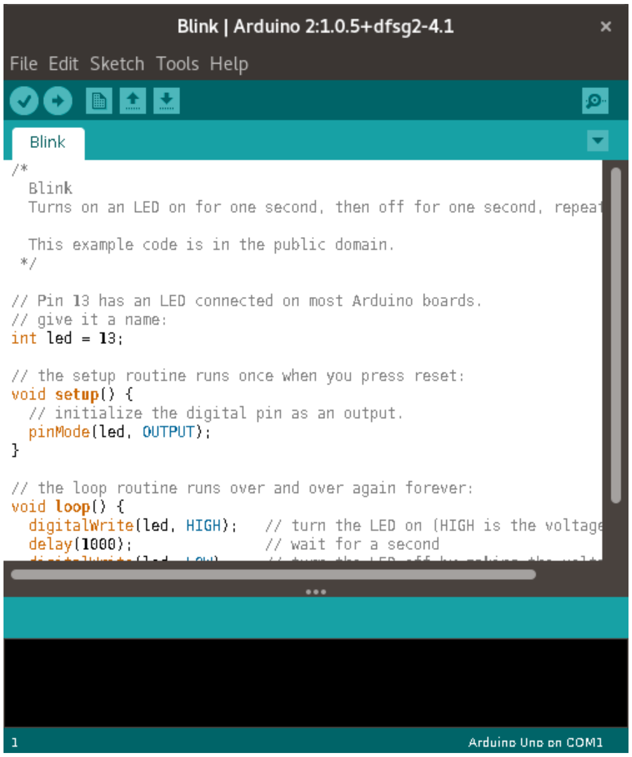
Do not forget to make **photos & videos of your system** for your journal!

Part II

Arduino

Step 1 – Launch the Arduino IDE

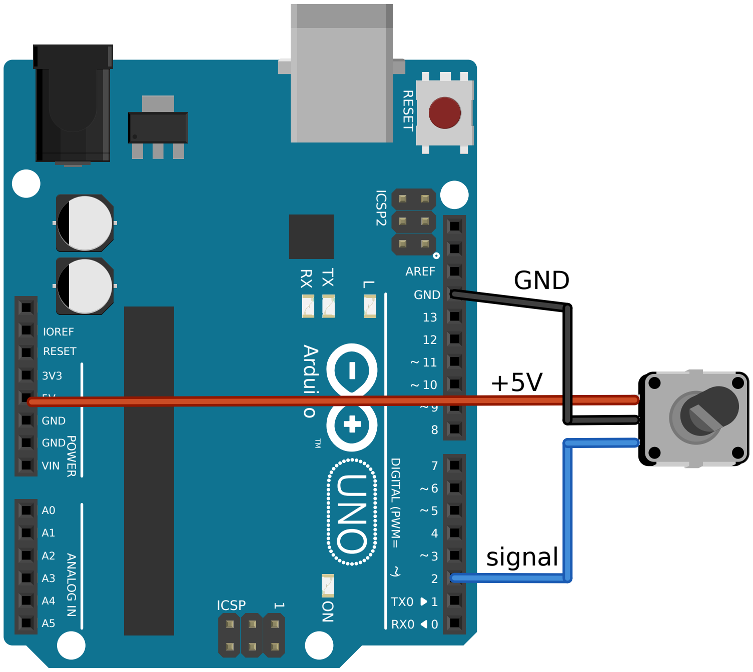
Open the Arduino IDE, plug the provided Arduino, and make sure the IDE is correctly configured for your card (check the card type – Arduino UNO – and the port – likely `/dev/ttyACM0`).


Blink | Arduino 2:1.0.5+dfsg2-4.1
File Edit Sketch Tools Help
Blink
1
Arduino Uno on COM1

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeat

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage
  delay(1000);               // wait for a second
```

Step 2 – Connect your encoder

Use the following diagram to connect your encoder to the Arduino.



Using the following program, test whether your encoder is correctly connected. Manually block the light path of your light detector: it should cause the Arduino LED to blink.

```
const byte ledPin = 13;
const byte interruptPin = 2;
volatile byte state = LOW;

void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(interruptPin, INPUT);

  // configure the interrupt call-back: blink is called everytime the pin
  // goes from low to high.
  attachInterrupt(digitalPinToInterrupt(interruptPin), blink, RISING);
}

void loop() {
  digitalWrite(ledPin, state);
}

void blink() {
  state = !state;
}
```

Step 3 – Calculate the angular velocity of your motor

Modify the program to count the number of pulses. Use this count to output on the serial port the speed of your motor every second.

The following code example shows how to write to the serial port every second:

```
void setup()
{
  Serial.begin(9600);             // set up Serial library at 9600 bps
}

void loop()
{
  Serial.println("Hello world!");  // prints hello with ending line break
  delay(1000);                     // wait 1s
}
```

Taking it further

Check on the web the other ***modes*** of `attachInterrupt`. A wise choice and a bit of math should let you improve the resolution of your encoder.

Part III

Components

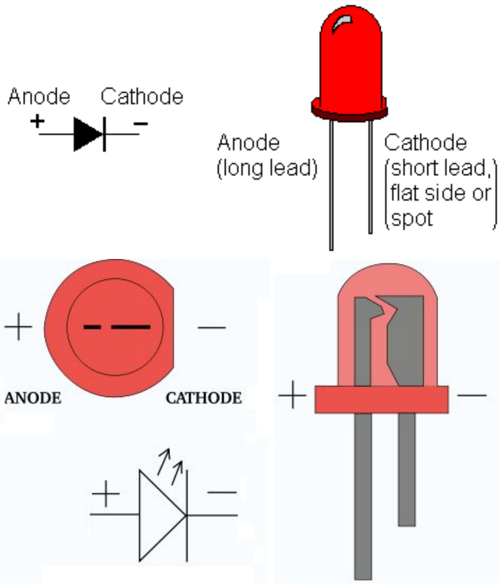


Figure 5: Led pins – same for IR LED

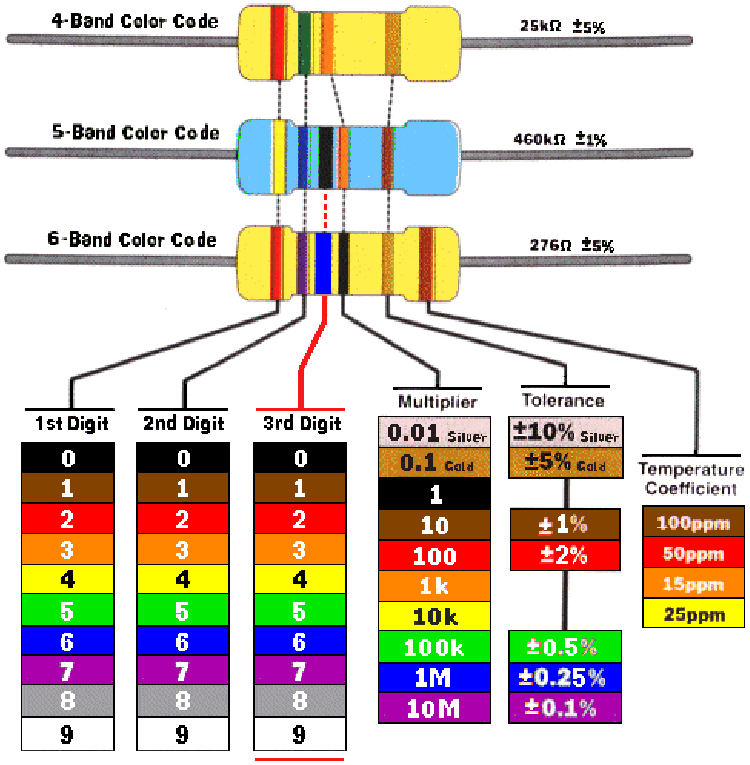


Figure 6: Resistor colour codes

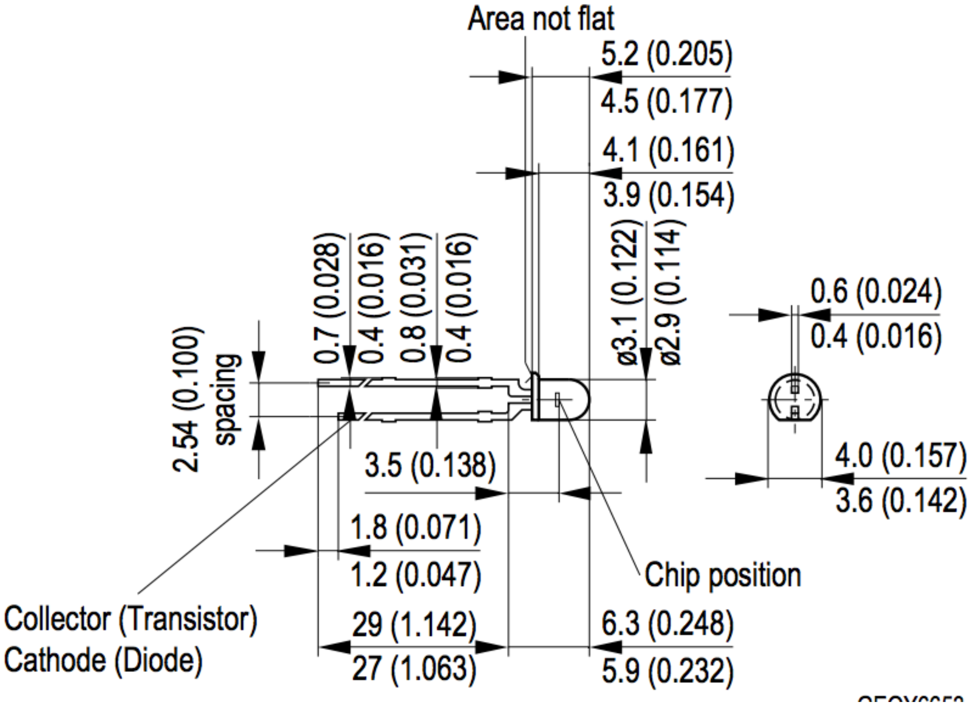


Figure 7: SFH309FA connection polarity