Lauren Courtney

February, 2020

EN 605.715.81 Embedded Systems

**Project 3 – Measure Fan Speed with IR Detector**

**Requirements**

1. The Arduino shall use an IR Emitter and Detector to measure fan speed

i. The fan speed shall be measured in rotations per minute (RPMs)

ii. The IR emitter and detector shall be placed far enough apart from each other that the fan speed is calculated by counting the number of times the beam is broken between them.

2. The fan shall be powered by a power supply module separate from the Arduino

3. The setup of the Arduino and breadboard with IR emitter/detector shall follow the example of [CMPalmer](https://www.instructables.com/id/Arduino-Based-Optical-Tachometer/) in the Arduino-Based Optical Tachometer instructions

4. The RPMs shall be captured over time and sent over the serial connection to the host where they will be graphed in real time.

5. An LED will serve as a sensor to indicate when the beam between the IR emitter and detector has been broken.

6. The implementation shall use a Round Robin with interrupts design.

i. This implementation shall implement this design using the attach and detach interrupts functions.

**Design**

**Hardware**

1 Arduino – Mega 2650 (program should run on any Arduino)

1 IR Emitter

1 IR Detector

1 10KΩ Resistor

7 Jumper Cables

1 LED

1 KΩ Resistor

1 Power Supply Module

1 9V1A Adapter

1 Fan Blade

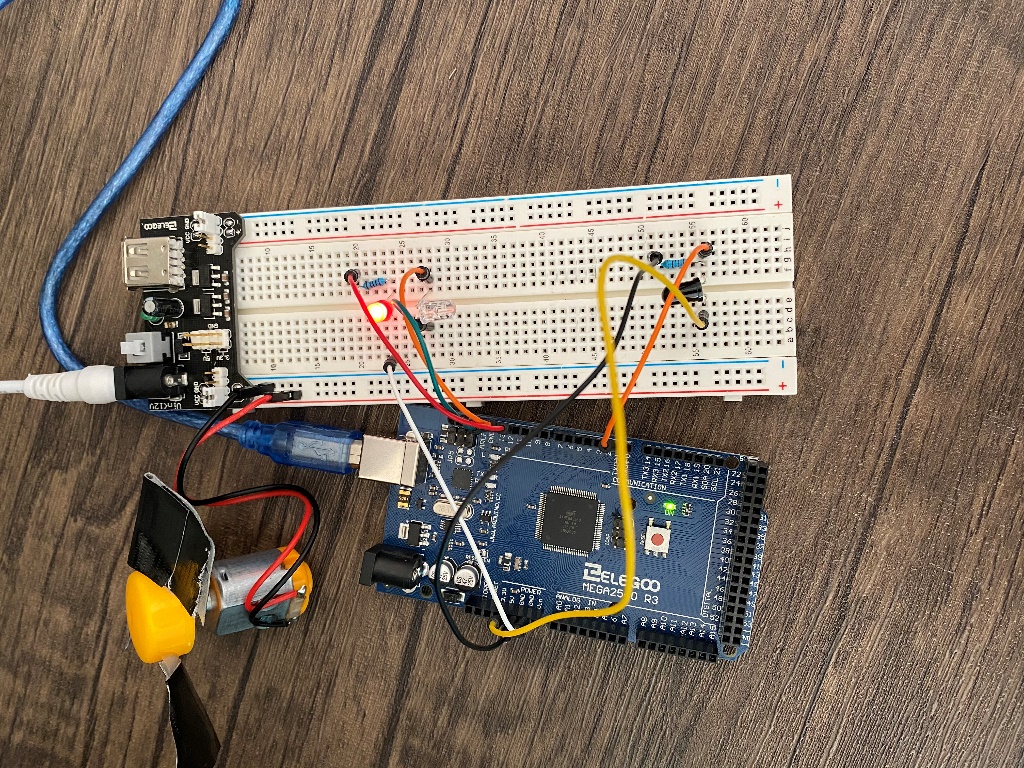
1 3-6V Motor with wiring

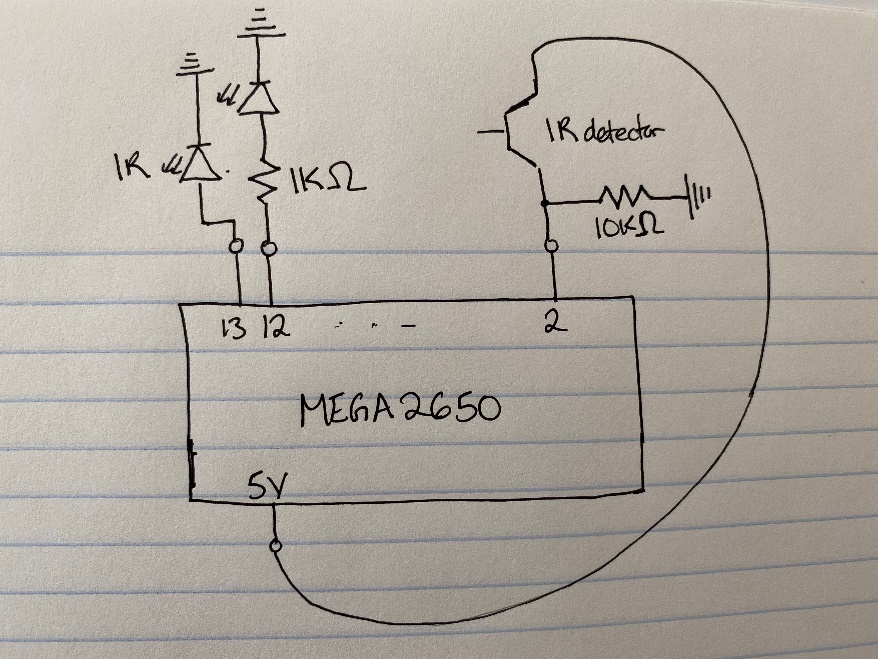
Black tape

1 Breadboard

1 Laptop

1 USB cable





Connect the positive end of the IR emitter to pin 13 and the negative end to ground. Then connect an LED to pin 12 using a 1K Ohm resister (just as in project 1). Connect the short end of the IR detector to the 5V power source, and connect the long end to pin 2 and a 10K Ohm resister. Connect the other end of the resistor to ground. This is seemingly opposite to the configuration used by CMPalmer, but it was recommended by a [fellow student](https://www.youtube.com/watch?v=5qeKiJ8wD8g) and works consistently. Note that any artificial light in the room may overload the IR detector and make it appear that your circuit is not working. In addition to the above, plug in the Power Supply module using the Adapter, and connect the positive end of the fan to the positive end of the power, negative to negative. When you turn the power supply on, the fan will turn. Cover the blades of the fan with clear tape so the IR light cannot penetrate them. Use the serial port and USB cable to connect the Arduino to the laptop.

**Software**

The code will be written in C using Arduino Sketch for the rotation detection and RPM calculations. Reading the Serial data and plotting the real-time graph will be written in Python using Jupyter notebooks. This code is copied directly from [CMPalmer](https://www.instructables.com/id/Arduino-Based-Optical-Tachometer/).

I. Initialization:

- Set ledPin to 13 for the IR LED

- Set status pin to 12 for the sensor LED

- Initialize arrays with hardcoded Morse Code letters and numbers as “..-- “, etc.

- Initialize base time unit to 100 milliseconds

- Create a buffer for string input and initialize a maximum size of 100 chars

II. ISR:

- Note that this is run three times every rotation when calculating RPM (since my fan has three blades)

- Increment the RPM count

- toggle the LED sensor light

III. Setup:

- Open the Serial connection

- attach an interrupt to interrupt 0, which is digital pin 2 where the IR detector is connected. It will trigger on the falling edge (so when pin 2 goes from HIGH to LOW – when the connection between the emitter and detector is broken)

- Turn on the IR LED, this will run continuously

- Set up the sensor pin so it can flash when the interrupts happen (when the beam is broken)

- Initialize values

IV. Loop:

- Update the RPM every second, so delay 1000 milliseconds

- detach interrupts while calculating RPMs

- calculate the RPM using the RPM count per second and the fact that the interrupt happens three times every revolution (since my fan has three blades)

- write the data to the serial port

- re-attach the interrupt once the calculations are complete

IV. Python notebook

- Set-up a serial connection to the Arduino through COM3

- Initialize a graph with time on the x-axis and RPMs on the y-axis.

- Make the maximum size of the live graph 100 seconds

- While the demo is running:

- Read in time and RPM data over the serial connection

- Format the data so it can be displayed on the graph

- Update the graph as each new data point arrives as outlined in Source [3](https://gist.github.com/brandoncurtis/33a67d9d402973face8d)

- After the demo, plot another graph with all the data

- Write the data to a CSV so it can be exported and loaded into another graphing program

**Demo**

Demo can be found on YouTube at <https://www.youtube.com/watch?v=OCgUtS_07WA>

**References**

Source 1: <https://www.instructables.com/id/Arduino-Based-Optical-Tachometer/>

Source 2: <https://www.youtube.com/watch?v=5qeKiJ8wD8g>

Source 3: <https://forum.arduino.cc/index.php?topic=625904.0>