Lab 3 – Fan Control

Goals of the Lab

1. Control the speed of a computer fan using PWM.
2. Measure the fan speed using tachometer circuitry.

Links to Tutorials and Specifications

Intel_Motherboard_4_Wire_PWM_Spec.pdf (in Library section of Canvas)

Parts List for the Lab

F14 PWM Fan

Optocoupler

LM340T5 Voltage Regulator

Resistors: 1k, 10k, 660, 150

Capacitors: 100nF

Instructions

Some of the grade of this lab will be based on the report you turn in. The report is simply this lab manual (only Lab 3, not the entire manual) with answers and schematics. Please write your answers in green below the questions and paste schematics in from Altium where requested.

Part 1: Beginning Analysis

Fan Analysis:

1. What voltage does the F14 PWM fan require to operate?
    a. 12VDC Typical, 6VDC Starting
2. According to the 4-wire PWM Controlled Fans Specification document, what is the maximum steady state current allowed for a 4-wire computer fan?
    a. 1.5A
3. How much current does the F14 PWM fan use?
    a. 0.11A @ 12VDC
4. What voltage is required for the PWM input signal to the fan? Where does this voltage come from?
    a. $V_{LOW} = 0.8V$
    b. $V_{MAX} = 5.25V$
    c. The voltage is provided by an external power supply, with the signal controlled by the RasPi through an Optoisolator
5. How could you turn the fan off without removing power from the fan? This is not directly stated in the documentation but was discussed in the lecture material.
    a. By Grounding the PWM pin
6. What frequency is required for the PWM input signal to the fan?
    a. 21kHZ -> 28kHz (25kHz target)
7. What is the maximum current for the PWM input signal to the fan?

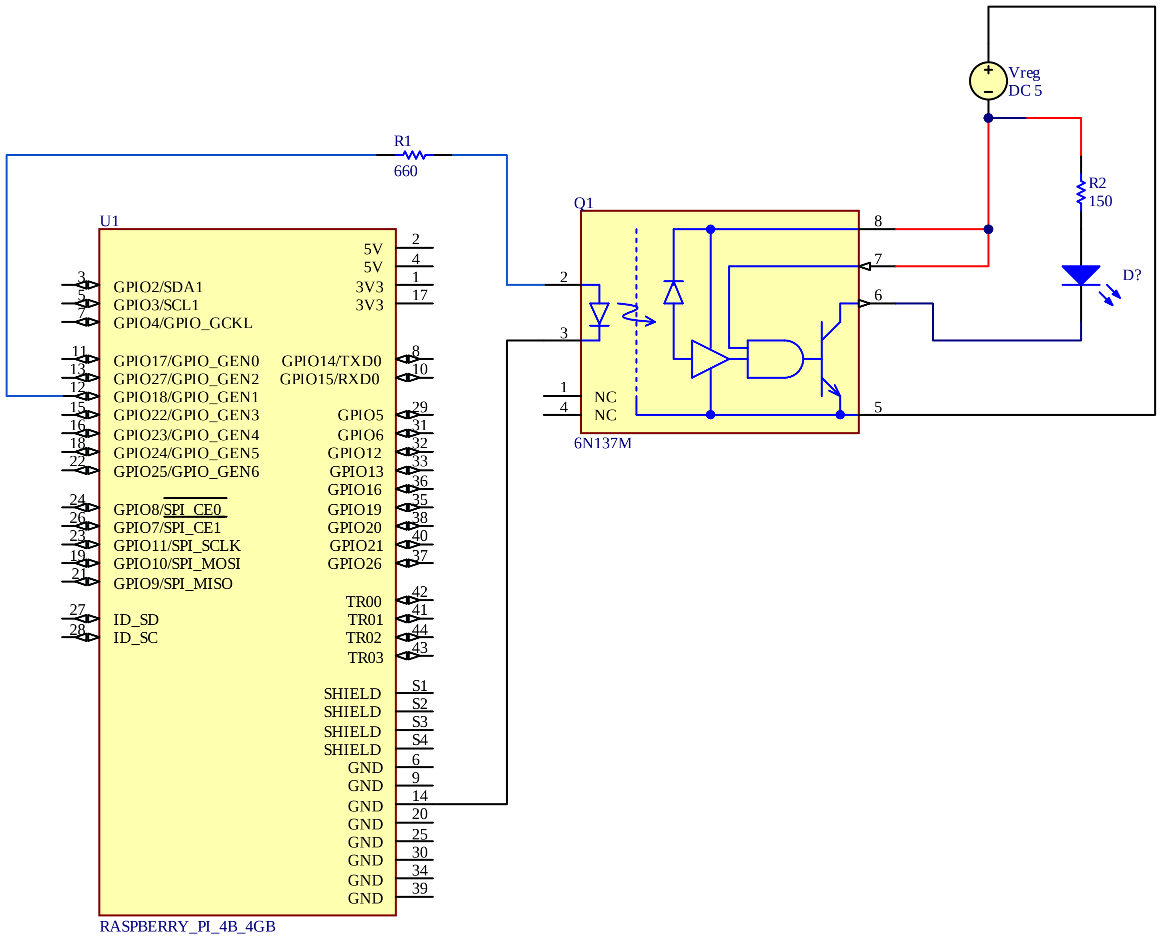
    a. 5mA

8. Complete the table for the pinout of the fan:

Pin Number	**Pin Function**	**Voltage Level**
1	Ground	0V
2	Motor Power	12V
3	Sense / Tach	12V
4	Control / PWM	0.8V – 5.25V

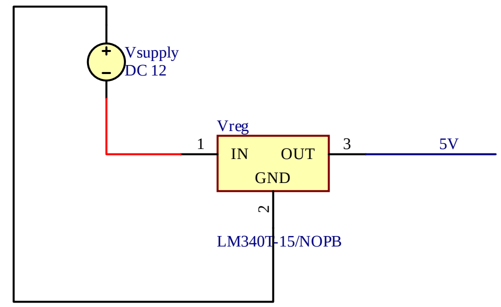
Optocoupler Analysis

1. What is the recommended supply voltage of the optocoupler?
    a. 4.5 – 5.5V
2. What is the typical input forward voltage of the emitter?
    a. 1.45V
3. Design a circuit using the single channel optocoupler described in the datasheet.
    a. The input side of the optocoupler should receive a signal from RPi pin 12.
        i. The current going into the optocoupler should not exceed 5mA
    b. The output side of the optocoupler should mirror the output side of the MOSFET LED circuit used in Pi Lab 2.
    c. When RPi pin 12 is 0V, the LED should be off. When RPi pin 12 is 3.3V, the LED should be on.
    d. The input and output sides of the circuit should NOT be connected in any way, including that they should not share a common ground.
    e. Draw this circuit in Altium. Paste the schematic here.



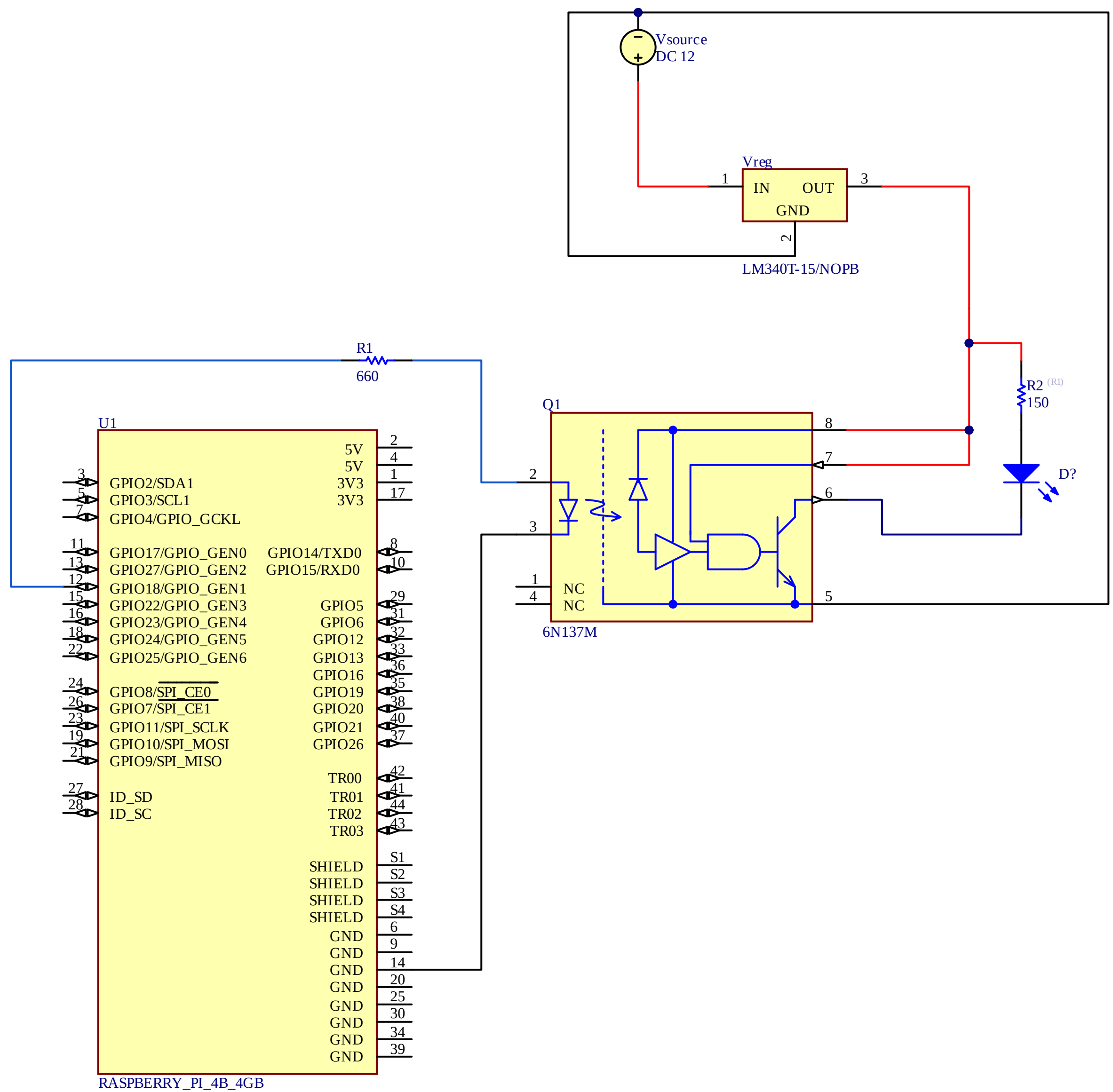
Voltage Regulator Analysis

1. What is the maximum output current?
    a. 1.5A
2. What is the maximum input voltage?
    a. 35V
3. Using Table 6.6, what is the output voltage?
    a. 5V
4. In Altium, draw a circuit that produces a 5V output when the input is the fan power supply. Paste the schematic here.



Combined Circuits

1. Now use the circuits previously designed and combine them into a single circuit where voltage regulator powers the output side of the optocoupler circuit. Paste the schematic here.



Part 2: Build and Test Power Circuitry

1. On your breadboard, build a power supply circuit that does the following:
    a. Uses 12V power supply connected to DC barrel jack connector
    b. The power rail on the left side of the breadboard is 12V
    c. The power rail on the right side of the breadboard is 5V
    d. The power rails share a common ground
2. Use your multimeter to verify the power rails have correct voltages.

Part 3: Build and test the Optocoupler

1. Build the 'Combined Circuit' designed above. Use 5V from the voltage regulator to power the optcoupler and LED output side of the circuit. Instead of connecting the input side of the circuit to the RPi, temporarily connect the Digilent to supply the input signal .

a. Please note that connecting a voltage directly to the input pin on the optoisolator will damage the optoisolator. You must use a current-limiting resistor as designed.
b. Start with just providing 3.3V or 0V
    i. When 3.3V is applied, the LED should turn on. When 0V is applied, the LED should turn off.
    ii. When 3.3V is applied, test the current between Digilent and optoisolator
c. Then use the function generator in Waveforms and make a square wave to emulate PWM
    i. Set the offset to 1.65 and the amplitude to 1.65 to generate a 0-3.3V range
    ii. Verify that you can brighten and dim the LED by changing the duty cycle (symmetry).

2. Once it works with the Digilent and you have verified that it does not pull too much current, replace the Digilent with the RPi and run the led_dimmer code from Lab 2.
    a. RPi Pin 12 should be connected to the resistor on the input side of the optoisolator.
    b. An RPi Ground pin should be connected to the ground pin on the input side of the optoisolator.
        i. This ground should not be connected to the other grounds to maintain electrical isolation.

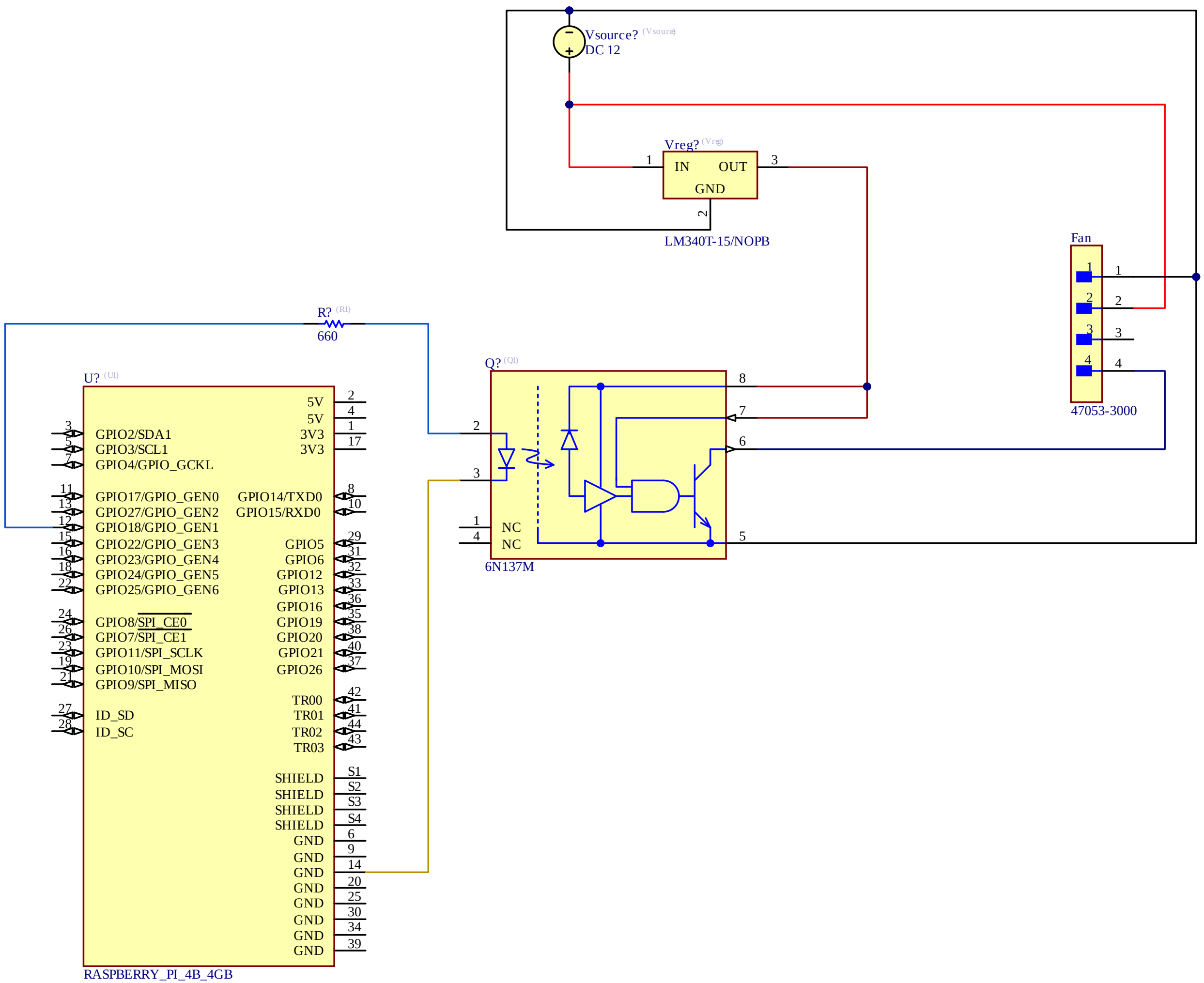
Part 4: Test the F14 PWM Fan

1. To connect the fan to your breadboard, use header pins that stick into the breadboard (they are in the supply cabinet). Then you can connect the fan connector to the header pins in the breadboard, providing a much more secure connection than just plugging wires into the fan connector.
2. Connect the 12V power and Ground rails to the Power and Ground pins on the F14 PWM fan and verify that the fan runs.
3. Use your multimeter to measure the voltage of the Control pin on the fan and record the voltage below.
    a. 4.82V
4. What is the value of the pullup resistor in the hub of the fan? You will need to measure this using your multimeter.
    a. Add a 10k resistor to ground, use your multimeter to measure the voltage across the resistor, and then use voltage divider equation to determine the internal resistance. Show your results and work below.
        i. 1.8V across resistor
        ii. $V_{out} = \frac{Z_2}{Z_1+Z_2} * V_{in}$
        iii. $Z_1 = \frac{Z_2 V_{in}}{V_{out}} - Z_2$
        iv. $Z_1 = 14{,}896\Omega$
5. To turn the fan off, you need to set the Control pin to 0V. To do this, you need to ground the Control pin.
    a. Ground the Control pin and make sure that the fan turns off. Disconnect the Control pin from ground and make sure that the fan turns back on.

i. The fans with black blades are defective and do not turn completely off. For the rest of the lab, when I say to verify that the fan turns off, we will just verify that it almost turns off.

Part 5: Control Pin Integration

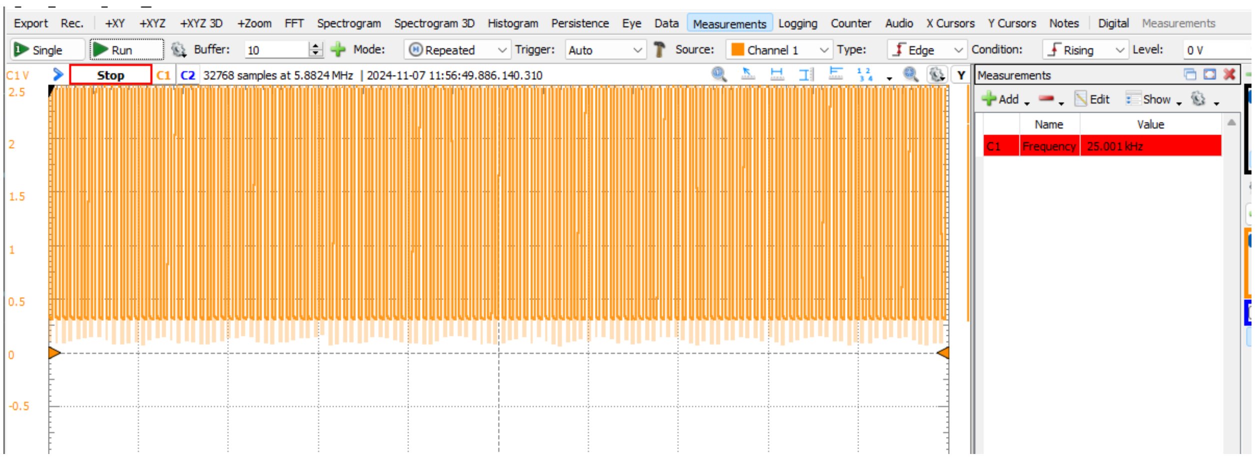
1. Disconnect the LED circuitry that was used for testing purposes (green LED and 150 ohm resistor). We will not be using these again.
2. Design a circuit to connect the Control pin on the fan to the output side of the optoisolator (instead of the LED circuitry). The open/closed behavior of the optoisolator should determine whether the fan is on or off. Draw the circuit in Altium and insert the circuit schematic below.



3. Build the circuit and test it using the Digilent on the input side of the optoisolator.
    a. You should be able to turn the fan on and off with the Digilent, and you should be able to control the speed of the fan using the duty cycle on the Digilent.
4. Use the pwm_fan_test code provided in Canvas to test the operation of the fan.
    a. The code will ask for a duty cycle value. Provide a value from 0.0 to 100.0
    b. Go through a variety of duty cycle values and verify that the fan speeds up and slows down when the duty cycle is changed.

i. You will notice something interesting about the behavior at different duty cycle values. Why do we see this behavior?
    1. The values are swapped (i.e 100% duty cycle means 0% fan)
    2. When the optoisolator is "powered", it bridges the fan to ground, which turns it off. When is it not "powered", the fan is floating, which means it's on full blast. Thus the inverted behavior.

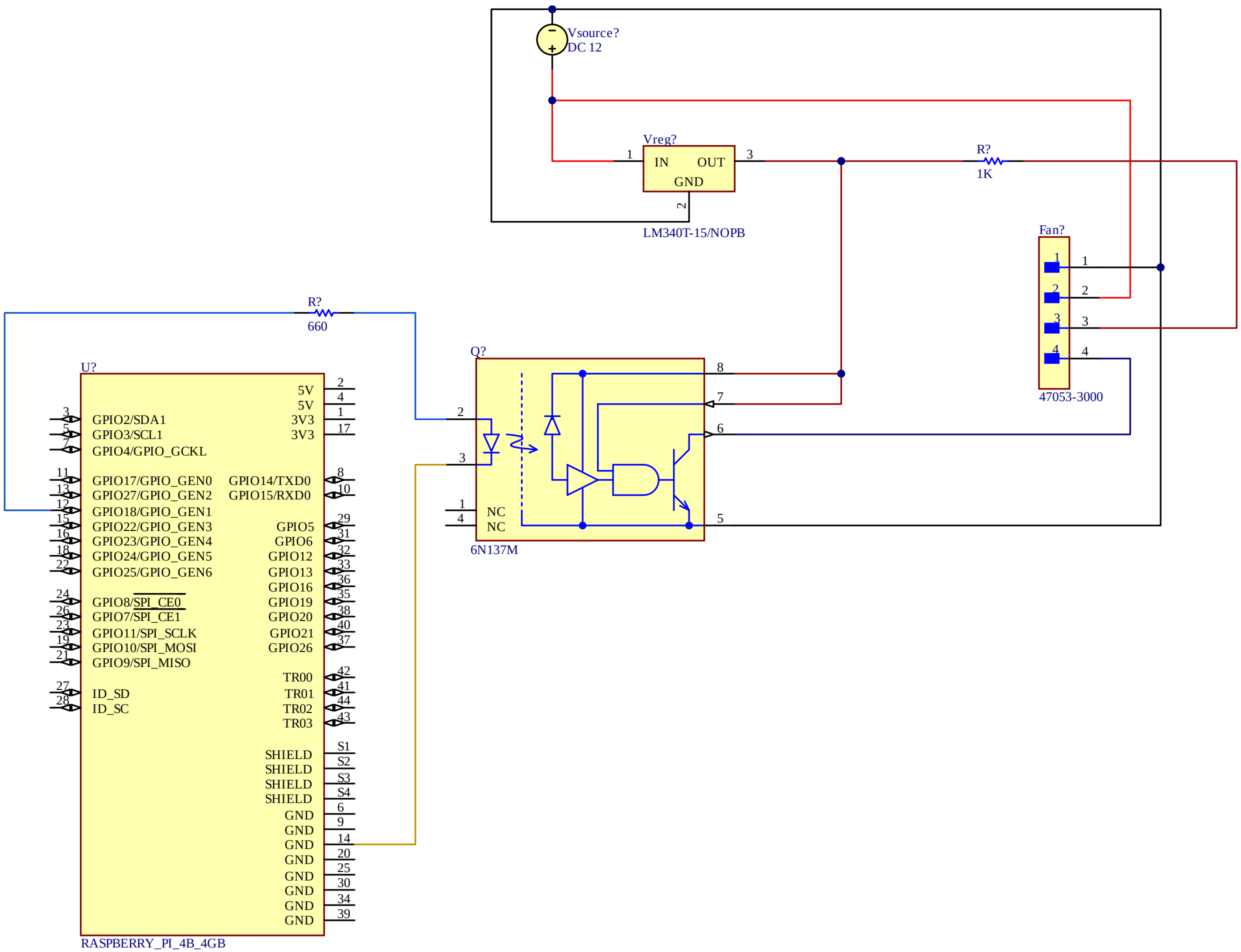
5. Use the Digilent Oscilloscope functionality to verify that the appropriate 25kHz PWM signal is being provided to the Control pin on the fan.
    a. Capture a screenshot of your PWM signal that shows your measurement of 25kHz.
    b. Insert the screenshot below.



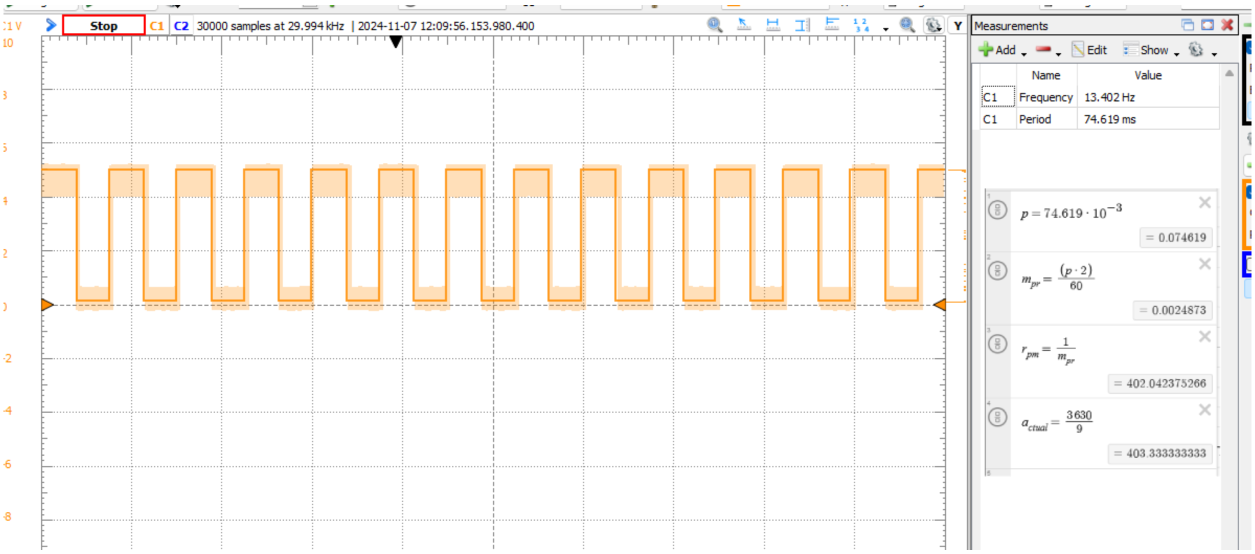
Part 6: Tachometer (Sense) Signal Design

The tachometer signal is briefly described in Section 2.1.3 in the Intel_Motherboard_4_Wire_PWM Spec. To explain further, there is a hall effect sensor in the hub of the fan. The fan has 2 magnets that are detected by the hall effect sensor, therefore producing two pulses for every revolution of the fan. The spec tells us that the tachometer signal has an open-collector or open-drain output. It also tells us that the "motherboard" will have a pull up to 12V. In a computer, the motherboard provides the circuitry that is connected to the fan. In our case, we are building the circuitry to connect to the fan. Therefore, we are the "motherboard". However, there is no need for us to pull the voltage up to 12V, since the rest of our system does not use 12V. In our case, we should pull the output up to 5V.

1. Design a basic Tachometer circuit that only includes the 5V source, 1k pullup resistor, and the Sense pin (and ground of course).
    a. In Altium, draw the schematic for the circuit and insert the schematic at the top of this part.



2. Build and test the basic Tachometer (Sense) circuitry. This circuit should not be connected to the RPi, but you should use the RPi to drive the fan at various speeds using the pwm_fan_test program.
    a. Use the Digilent Oscilloscope functionality to analyze the signal produced by the tachometer pin. Try to determine the RPM value based on the output of the Digilent.
        i. Capture a screenshot of the Digilent where you measure the RPM value. Insert the screenshot below, and also show the calculation that you used to determine the RPM value. (75% duty cycle)



b. Use the laser tachometer to measure the fan speed. Compare the measurement to the RPM value calculated from the Digilent and verify that they are similar. Please note that a fan with white blades will pick up every blade turning, so you will need to divide the laser tachometer reading by the number of blades.

   i. Record your calculated values and your laser tachometer below. Do this for three different duty cycle values

   ii. 10% -> 24.385 ms = 1230RPM; Measured 1222RPM

   iii. 32% -> 30.466 ms = 984RPM; Measured 988RPM

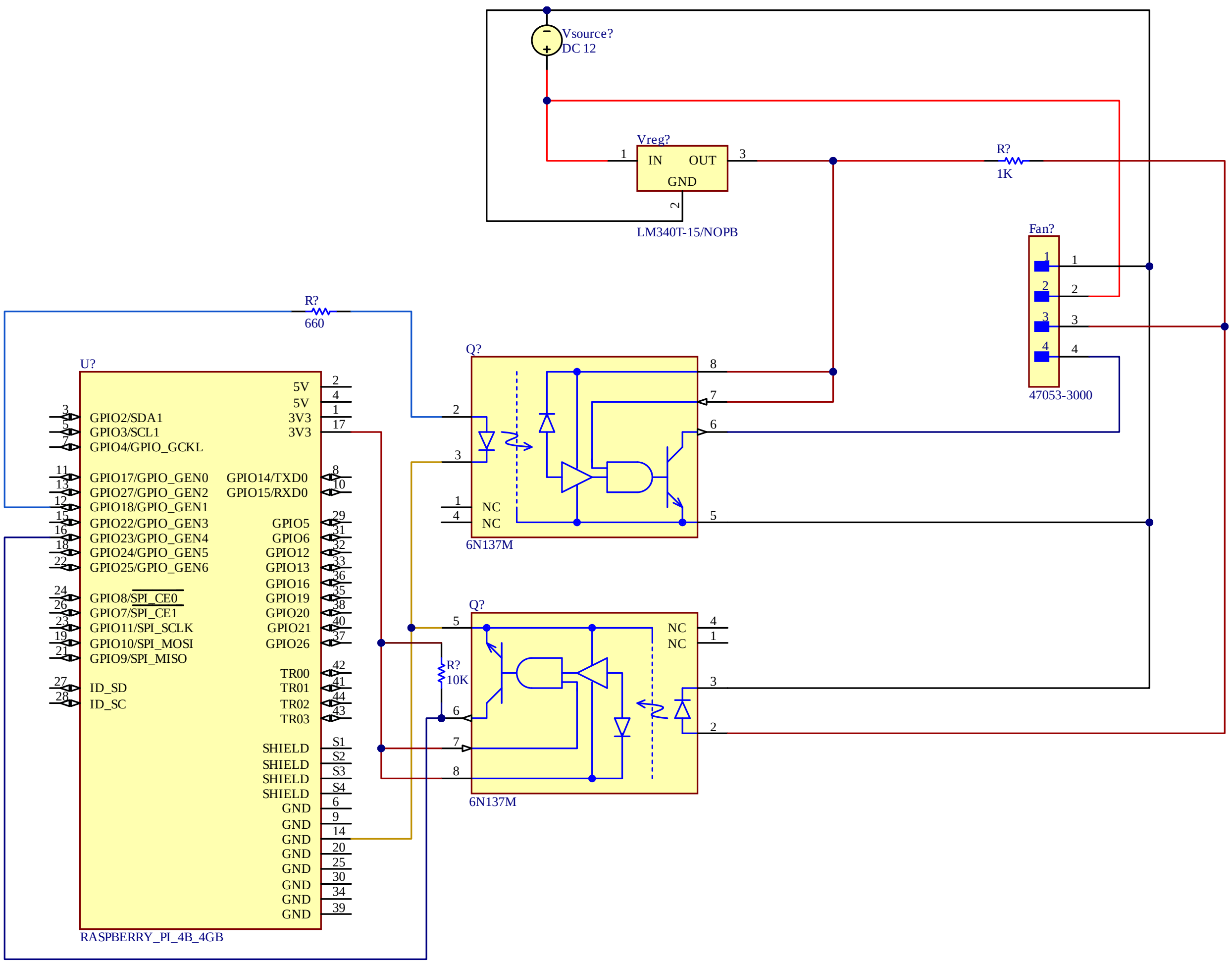
   iv. 85% -> 130 ms = 230RPM; Measured 217RPM

Part 7: Integrate Tachometer with the RPi

1. Once the basic tachometer circuit works, use a second optocoupler as an electrical barrier between this tachometer signal and the RPi.
   a. Design a circuit that connects the basic tachometer circuit to the input side of the second optocoupler. The tachometer signal will drive the emitter in the optoisolator.
   b. Build the circuit and test that the tachometer circuit works as expected by using the Digilent to view the voltage of the optoisolator input pin and verifying that it is changing in a similar way that the basic tachometer circuit worked.
2. Design a circuit on the output side of the optocoupler that will read the output of the second optocoupler with RPi Pin 16. Do NOT connect this circuit to the RPi yet. We will test it first.
   a. Draw and insert the schematic for this circuit. This schematic should include the control pin circuitry connected to the input side of the optoisolator as well as the output side of the optoisolator that is connected to the RPi.
      i. This circuit is relatively similar to the push button circuit used in Lab 1. Please make sure to use a 10k resistor, and make sure to check the resistor value when you build it in case the resistors are not sorted well.
      ii. All connections on the output side of the optoisolator should be connected to the RPi and should not be connected to the rest of the circuitry on the breadboard. This includes power pins, ground, and the input pin. We should

design a circuit that is completely electrically isolated from the 12V source, 5V source, fan, etc.

1. For Vcc and Ve, you should use the 3.3V power rail from the RPi. Do not use the 5V rail on the RPi.



3. Build the circuit but do not connect it to RPi Pin 16 yet. Test the output of the circuit (where you would connect RPi Pin 16) by checking:
   a. The voltage with the Digilent Oscilloscope
   b. The current to make sure that it is an acceptable current for the RPi.
4. Once tested, remove the Digilent and connect the circuit to pin RPi Pin 16.

Part 8: Final Schematic

Now that the circuitry is designed and complete, draw a schematic that encompasses all of the circuitry for this lab. Please submit this as a PDF to the Lab 3 assignment in Canvas.

Part 9: Tachometer Software

1. Now that circuitry is complete, please create a file called fan_controller_monitor.py and write software that:

    a. Allows the user to input a desired fan speed between 0.0 and 100.0. If the user inputs 0.0, the fan should be off (or close). If the user inputs 100.0, it should run at full speed.
    b. Reads the RPM value and prints the RPM value to a file called rpm_value_file.txt (about once per second). It should only print to the file after getting 10 samples and averaging those samples.
        i. To read the tachometer pin, please use the 'GPIO.add_event_detect' function and do your calculations and file writing in the callback.
        ii. Since the callback is running in a separate thread, you will need to wait for the callback to finish before exiting the program (when you get a keyboard interrupt). To do this, you can just set a flag when you begin the callback and clear it when you end the program. Otherwise, you will get a segmentation fault if the program terminates before the callback finishes.
2. Use the laser tachometer to measure the fan speed and make sure that your RPM values are similar to the laser tachometer's values.

Submitting the Lab

1. When complete, demonstrate it for me so that I can verify its operation.
2. Files to submit:
    a. fan_controller_monitor.py
    b. Your completed Lab 3 section of the manual.
    c. A PDF of your final schematic.