MONITORING SYSTEM FOR ROTATING TURBINES:

Final Document

By
Asm Fariz Khandker, Alaa Qarooni

UIN: 676283060, 670832813
NetID: akhand8, qarooni2

Fall 2017

Abstract

This paper culminates the different design choices, requirements and verifications, in addition to the overall functionality, of the monitoring system we built over the course of this semester. Modularity, low power consumption and production cost are key, and have been achieved by carefully chosen circuit setups and system blocks. Our system detects the rotations per minute (RPM) of a rotating turbine, and, depending on whether this RPM drops below a certain threshold, switches between user-provided main or secondary power supplies. It also includes a LCD screen to display the RPM to the user.

Table of Contents

1. Introduction

Turbines are an essential electromechanical component in many large systems. They are used for propulsion in planes and submarines. They are also used in cooling systems such as the ones in nuclear power plants, high performance computers, etc. Most of these systems, however, do not have a sophisticated monitoring and control system that maintains the operational speed of the fan.

Our design provides a cheap, power-efficient and modular way of solving this problem through power switching: the RPM of a turbine is detected through a phototransistor-laser pair and calculated through a microcontroller chip, it is cross-referenced with a threshold RPM that is unique to the turbine, then, depending on whether the RPM is above or below this threshold, a main or secondary power supply is provided to the turbine.

It is worth noting that the standard method of detecting RPM is through the use of a dynamo that generates a current proportional to the torque of the turbine. The problem with this approach, however, is that it is bulky, requires a mechanical split-ring commutator, requires various rectifying circuits to convert the alternating current to an acceptable DC, and is unreliable overall due to the complexity of the components involved. Furthermore, it is not a purely electrical system and relies on the integrity of various mechanical components. Our solution, on the other hand, uses a photonic integrated system that can be implemented into existing machines in a relatively non-invasive way, it is more precise and reliable due to the highly sensitive, but still cheap, phototransistor that is used for detection, and is setup to rely on purely digital signals for processing and analysis, making it more robust.

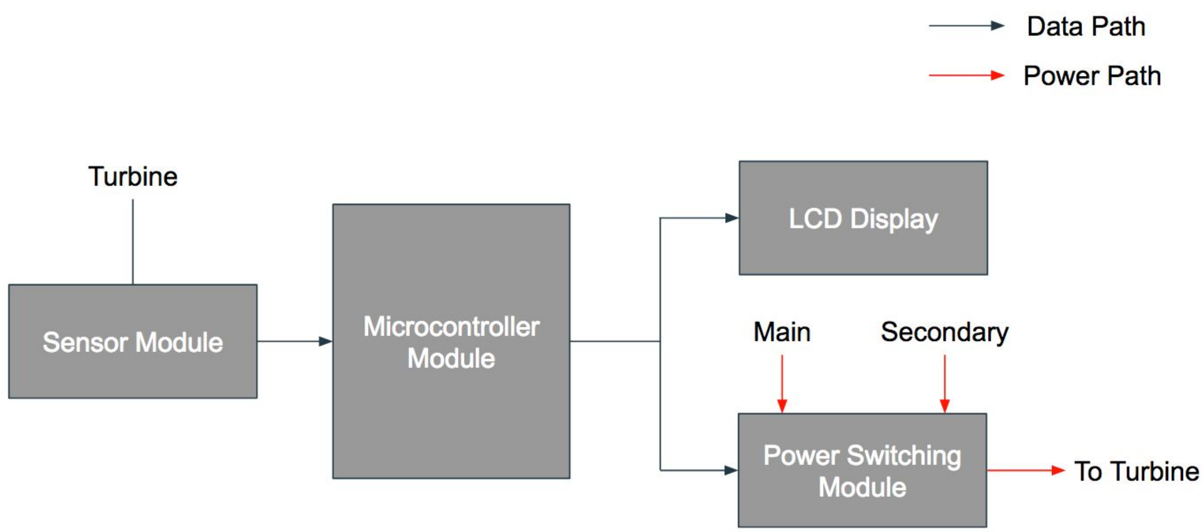


Figure 1: Block Diagram

Figure 1 shows the block diagram of our system. The sensor module generates information through a pulse train induced by the insertion of a rotating turbine into our photonic integrated system. It sends that information to the microcontroller module, which calculates the RPM value. The microcontroller module then sends print commands to an LCD display to print this value, but meanwhile, it checks to see if the RPM is above a certain user-specified threshold. If it is, then it sends commands to route the main supply to the turbine and the secondary supply away. If it isn't, then opposite is done.

2. Design

2.1 Design Procedure

2.1.1 Sensor Module:

We have chosen to use a BJT phototransistor and laser photodiode pair as our detector-laser pair in our sensor module. Several considerations have been taken when making this choice, the most important of which are:

1) The sensitivity of the detector to light coming from sources other than the laser.
    - We do not want it to incorrectly detect light at times when the laser is not directly hitting it. This makes for separating logic high and logic low values difficult. We therefore need a detector that has a high signal-to-noise ratio.
2) The time the detector takes to switch between high and low voltage values corresponding to when the laser is shone or not shone onto it.
    - The turbines this system will be dealing with rotate at a minimum speed of 200rpm, and, in addition, usually have 5-10 blades, so the time between each blade is usually in the order of a few milliseconds. We need a detector that is sensitive enough to detect this fast switching.

Initially, we were testing using a very expensive photodetector provided to us by Professor Peter Dragic. It was very precise and powerful, but came with the exact drawbacks we wanted to avoid: it had a very low signal to noise ratio, and a really slow response time. We therefore experimented with different photodetectors and found the BJT phototransistor to be perfectly suitable for our purposes, with response times dropping down to the microseconds.

2.1.2 Power Switching:

The core functionality is as follows: The microcontroller continuously calculates RPM. It also sends two signals MC1 and MC2 to maintain MAIN SUPPLY and disconnect SECONDARY supply as long as the fan is spinning optimally. A threshold is set for RPM, below which the microcontroller changes MC1 and MC2 to “switch” supplies. Based on this, certain design requirements become immediately apparent:

1) The power block is logically controlled. The output signals MC1 and MC2 from the controller are low voltages (logic level voltages of 0V and 5V). A MOSFET based design will be optimal for this. The IRF510 Power MOSFET can be controlled by logic level signals (low turn on voltage of 4 V) while the MOSFET

channel itself can handle large currents. Large Drain to Source voltages are also permitted. The tolerance for the device is 32 A I_{ds} and 60 V V_{ds}

2) Fast switching is also important. FETS are also ideal for this because the channel is inverted almost as soon the gate voltage exceeds threshold.

2.1.3 Microcontroller Module:

The ATmega328P microcontroller IC was chosen for this module because of its ability to act as a standalone Arduino. To process the RPM we need software that is capable of doing so and the Arduino IDE + programming language is suitable for this purpose. In addition, it operates at 5V 5mA, making it extremely power-efficient given that the bulk of information processing is done through it.

2.1.4 LCD Display Module:

We chose the "RioRand LCD Module for Arduino 20 x 4, White on Blue" for this module. The only requirement we considered when choosing this LCD display is whether it is HD44780 compatible, which means if it can communicate with the ATmega328P through the Arduino IDE.

2.2 Design Details

2.2.1 Sensor Module:

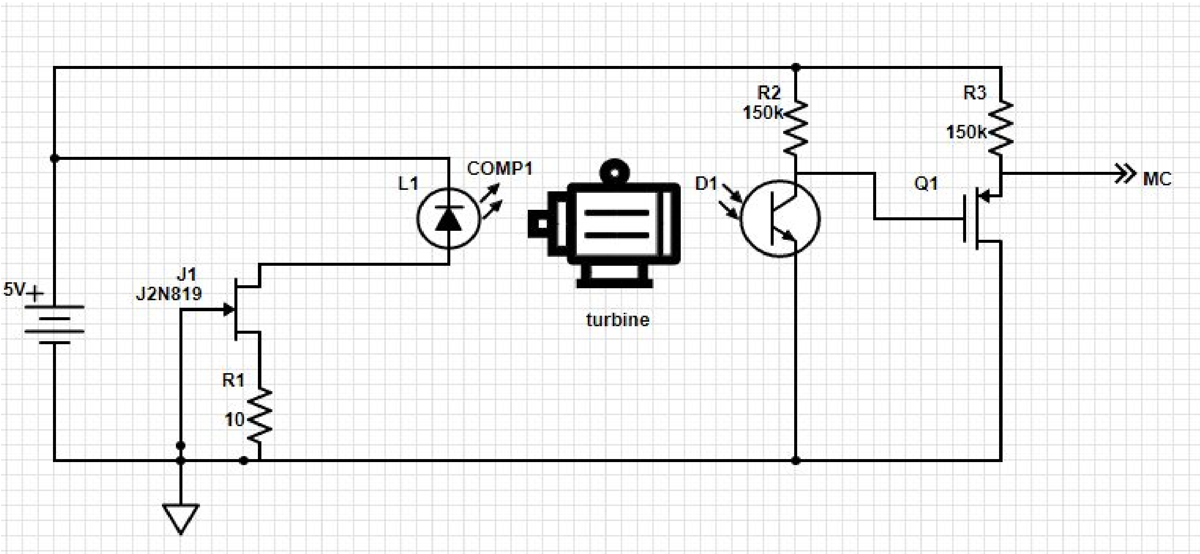


Figure 2: Circuit Layout of Sensor Module

In the above layout, a correctly matched Vertical Cavity Surface Emitting Laser and phototransistor (Photodiode-BJT) pair form the basis for the overall RPM detection scheme. Broadly speaking, the laser (L1) feeds a continuous signal onto the exposed base region of the BJT, resulting in continuous electron-hole pair generation. This sets up an applied electric field across the depletion region of the NPN transistor, and a current density due to the drift of excess minority carriers within the device. Whenever the turbine pulses the laser the current density in the BJT is reduced to zero corresponding to the blade of the fan interfering with the laser beam.

This produces a series of high and low signals to be fed into the microcontroller due to the common emitter configuration. The dynamic resistance of the BJT plays an important role in the high speed switching. Unlike standard BJTs the dynamic resistance, r_d, of the phototransistor depends on the light falling on it; with a sufficiently large pull up resistor tied to the collector we can get the desired "high" and "low" signals, as shown in Figure 3.

One problem with the above set up is that we need our "high" signals to be distinct highs (4.5-5V) and the "low" signals to be distinct lows (0-300mV). This creates enough of a difference in the voltage levels for the microcontroller to distinguish between them. For this purpose, we use a PMOS discriminator circuit. The PMOS also prevents negative voltages from appearing at the sensor output.

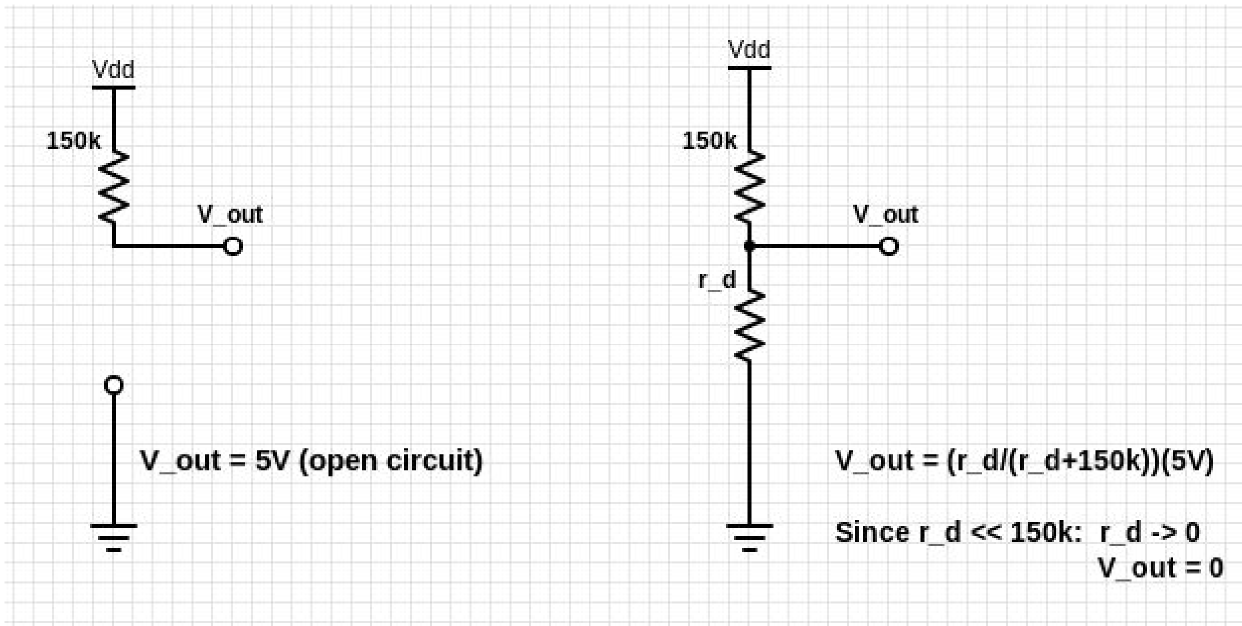


Figure 3: Phototransistor dynamic resistance

Functionally, the PMOS Q1 discriminator circuit guarantees that the alternating signals generated by the "light" and "no-light" conditions are distinctly "high" and "low". The particular PMOS used is the 771-BUK98150-55A/CUF chip by Mouser Electronics and has a low V_{gs} threshold of about 1V meaning that when V_{out} from the above circuit is 0V, V_{gs} of the FET is also 0V and an open circuit appears across the drain-source. This means that the full 5V from the internal supply gets passed on to the microcontroller, MC (V_{ds} = 5V). Alternatively, when V_{out} = "high" the microcontroller gets 0V.

Another important aspect of the sensor module is correctly matching the laser to the photodetector (phototransistor circuit). Although the laser needs to be compatible with the 5V source, it operates with a current source. Furthermore, we are aiming for a low power sensor design, so the selected laser (QUARTON 623nm red-dot VCSEL) has a maximum tolerance of 10mW at a 50mA drive current. Keeping this in mind the fixed 5V battery has to somehow be converted to a constant current source while ensuring that the drive current is restricted to well below the maximum tolerance. Keeping this in mind, we move forward with the following design choices.

First we show that the voltage appearing across the PN junction of the semiconductor laser is pinned at population inversion, and that the optical power output of the laser is a function of the drive current and not the voltage. The output power of the laser is determined using equation (1)

$$P_{opt}=\frac{hv\left(i_d-I_{th}\right)}{q}\eta_i\eta_e \qquad (1)$$

Here, η_i is the current utilization energy which is fixed for a given material. η_e is the quantum extraction efficiency which is also fixed by the dimensions of the material:

$$\eta_e=\frac{\alpha_I}{\alpha_m+\alpha_I} \qquad (2)$$

$$\alpha_m=\frac{1}{2L}\ln\left(\frac{1}{R_1 R_2}\right) \qquad (3)$$

Here, α_{I} and is cavity loss in the laser diode [fixed] and α_{m} is the mirror loss which is also fixed by the laser cavity length, L and "mirror" reflectivities, R_1 and R_2 inside the diode. All this means that we can combine the efficiencies into a differential quantum efficiency, η_{ed}:

$$\eta_i\eta_e=\eta_{ed}=const. \qquad (4)$$

The important takeaway from all this is that the efficiencies are only determined by physical factors and the output power is directly proportional to the drive current, i_d. Here the I_{th} is the minimum current required by the device. The input voltage does not matter. Matter of fact, once I_{th} is passed, the voltage across the photodiode is clamped (by the quasi-fermi levels) and is no longer relevant as seen in Figure 4.

It is intuitive to say that the laser behaves like a diode (it is in fact just that - a PN junction). For our laser the turn-on voltage is 3.5V (constant during lasing) and any further increase in input power due to increase in drive current results in an increase in optical output power.

Another significance of the above behavior is that we can now assume a linear relationship between the current and optical power and this allows us to select our operating point based on the detector without really needing to know the quantum efficiencies. The phototransistor saturates at 2mW optical power and we know that the laser is rated at 10mW at 50mA. So the required i_d is:

$$\frac{id}{2mW}=\frac{50mA}{10mW}$$

$$\Rightarrow i_d = 10\text{mA}$$

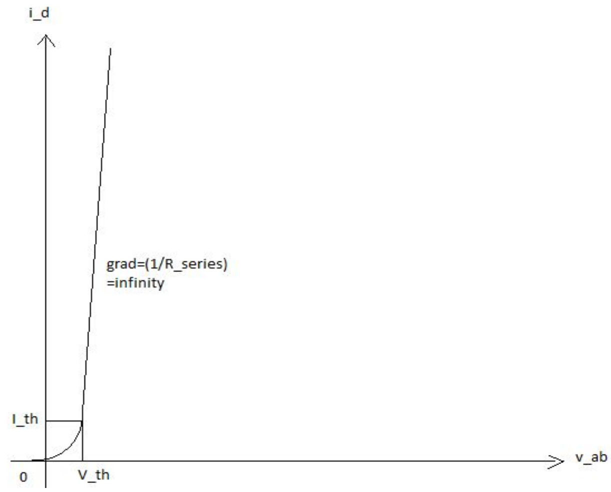


Figure 4: laser diode I_d Vs. V_d characteristics

Now that we know our drive current we can design a simple current control circuit using a the JFET, J2N3819 in a self biased configuration. LTspice simulations of the required circuit yield the results shown in Figure 5 for the different resistance values of the resistor R1 in Figure 2.

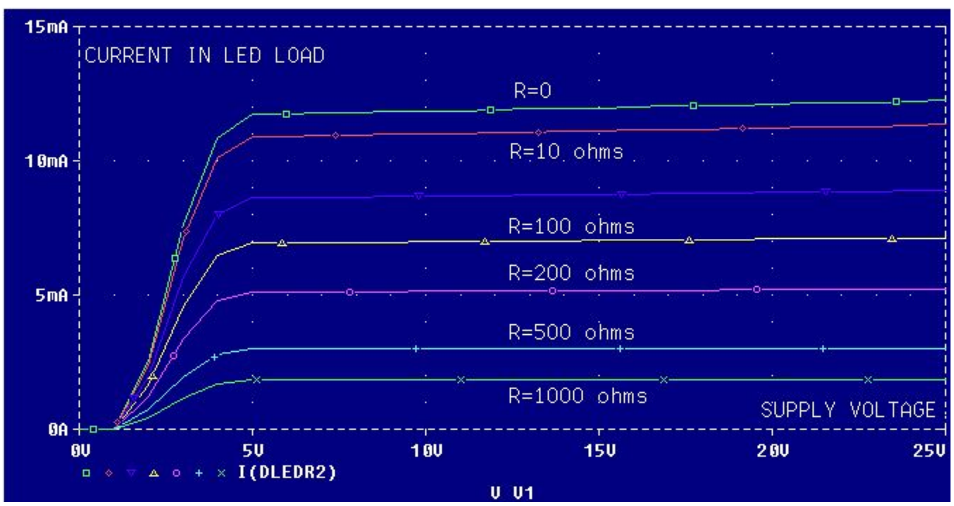


Figure 5: I_d Vs. V_ds for J2N3819 at different resistance values

So a 10Ω resistor is the closest we can achieve if we are to clamp the drive current at ~10mA.

2.2.2 Power Switching:

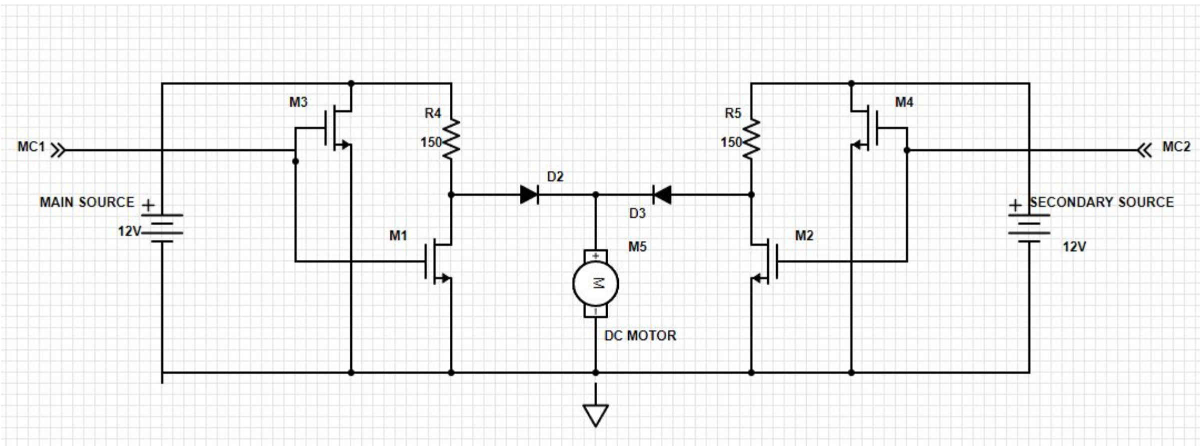


Figure 6:

For our design we used a NOCTUA 19cm brushless fan with an RPM of 800. The fan runs off of a 12 V DC supply. Therefore, the above circuit is a unique instance of 12 V being used for the MAIN and SECONDARY sources. The design, however, is consistent with any source. Figure 7 shows how the microcontroller will perform the switching action (based on the MC1 and MC2 signals it sends to this module). There will be two distinct states.

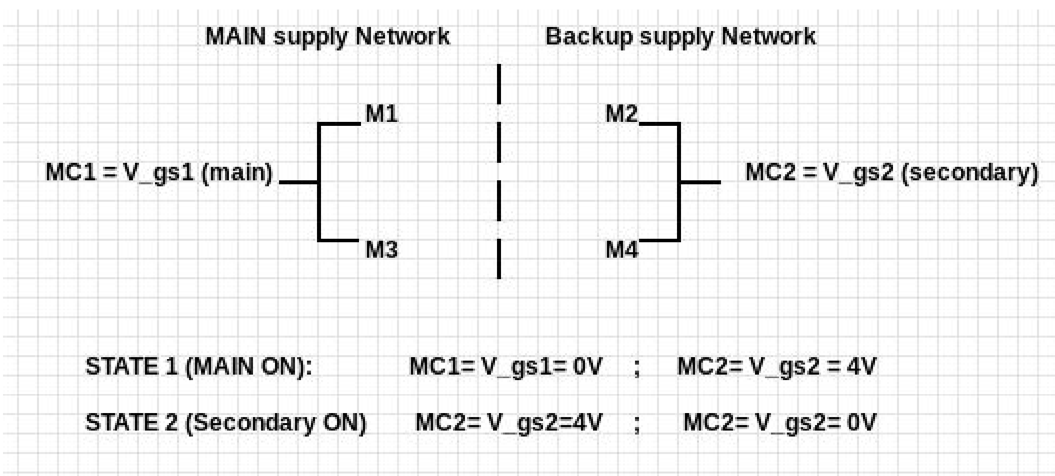


Figure 7

The design is symmetrical, with the two distinct power blocks (corresponding to the two sources) connecting parallely to the fan. For understanding the core functionality, it is worthwhile looking at one of the two states. Here we will discuss the state where the MAIN supply is ON and the

SECONDARY supply is OFF. The same logic will the be extended to the opposite state where the MAIN is OFF and the SECONDARY is ON.

Figure 8 shows the STATE 1 where MC1 = 0 V and MC2 = 5 V:

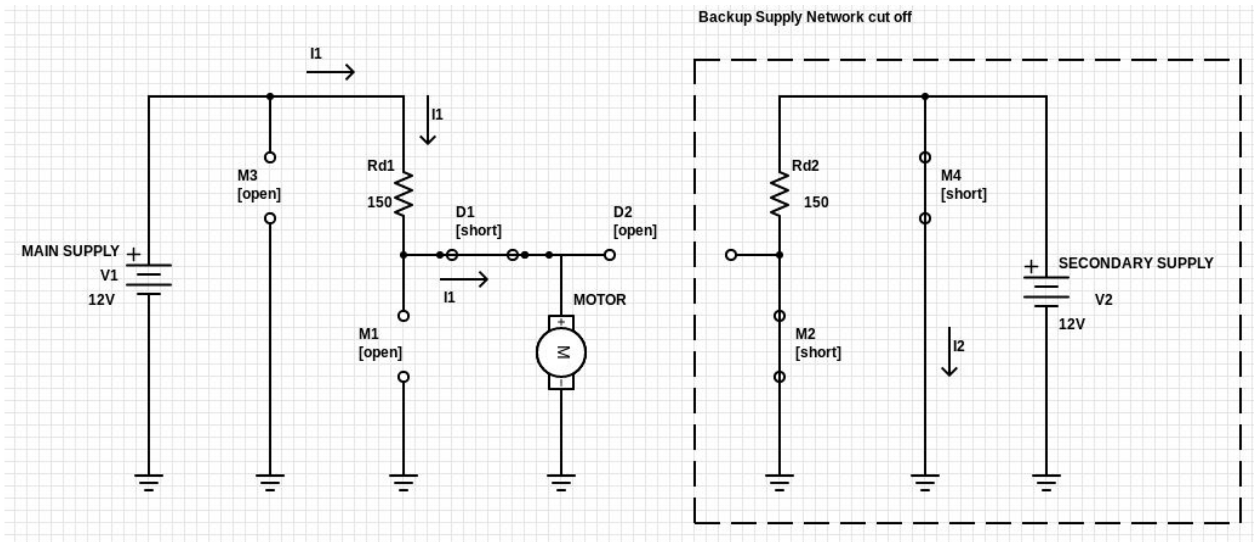


Figure 8:

This is a high level abstraction of what is going on. The open circuits at M1 and M3 redirect all the current, I1, to the motor through the pull-up resistor Rd1. Although shown to be a short circuit, the segment at diode D1 is not really a short. It will have a small voltage drop whilst providing a conducting path to the turbine. We therefore selected diodes with low turn-on voltages of 0.7 V, so that the majority of the voltage from the power supply is being drawn by the turbine when either of the diodes is conducting. Ideally, we would also use a relatively small resistance for Rd1 to ensure minimal dissipation across it. This applies to Rd2 in the opposite state when the secondary supply is providing power. We have elected to use 150 Ω, a small value compared to the motor resistance, which has an overall impedance in the order of 10^6 Ω. M4 is there to protect it from drawing too much current and burning, since the gate voltages may induce a drain current that it wouldn't be able to tolerate.

The same analysis is true for the opposite configuration (STATE 2) where where MC1 = 5 V and MC2 = 0 V, resulting in the MAIN network being cut off similar to the above diagram.

2.2.3 LCD Display Module

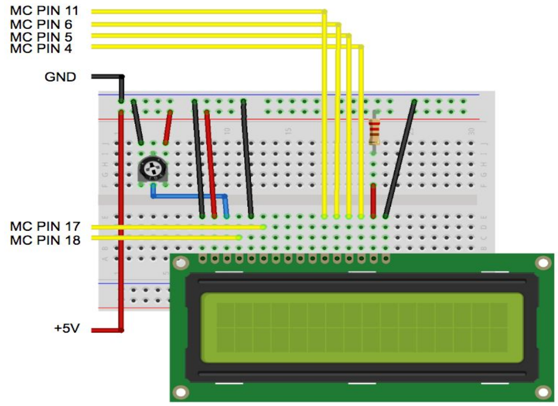


Figure 9: LCD Screen connections

The LCD Display is a very simple part of this design and only requires correct connections to and corresponding code in the microcontroller to function properly. We chose the “RioRand LCD Module for Arduino 20 x 4, White on Blue” display because it is Arduino compatible as indicated by its ‘HD44780 compatible’ specification. In addition, this display draws a maximum of 4 mA of current, meeting our power-efficiency requirement.

2.2.4 Microcontroller Module:

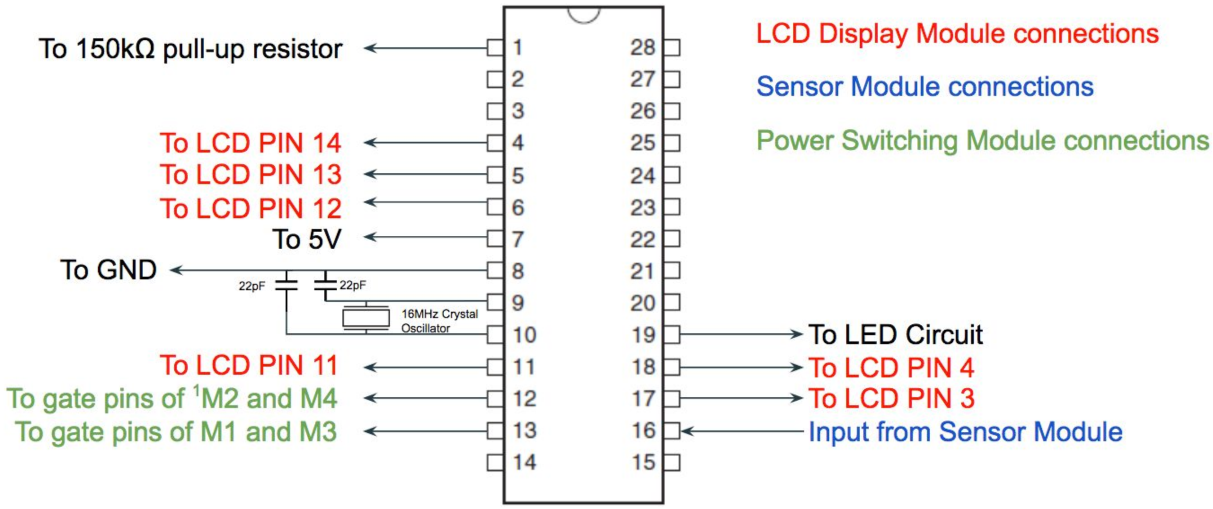


Figure 10: ATmega328P input and output connections

[1]See Figure 6 for to reference components M1, M2, M3 and M4

The ATmega328P microcontroller IC makes up our entire microcontroller module. It serves as a bridge from the pulse train information received from the sensor module to the power switching and LCD display modules. Using a test turbine with a known 800 rpm and 15 blades, we get the waveform depicted in Figure 11 from the sensor module.

Software design details:

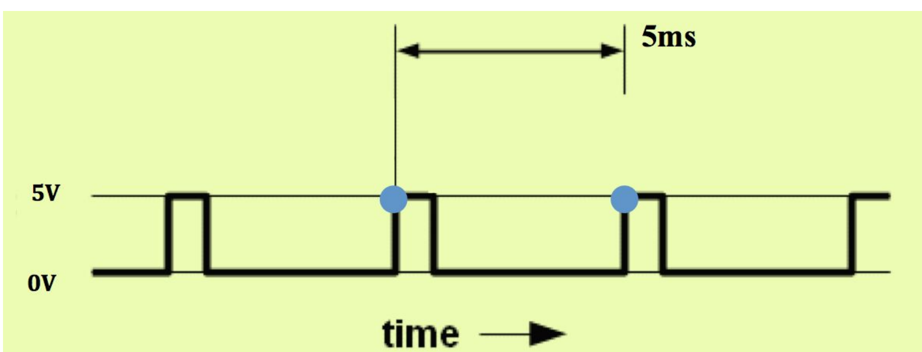


Figure 11: Pulse Train Waveform from phototransistor

The microcontroller reads this waveform as a digital input. The microcontroller can process data at very high frequencies, down to the microsecond level, so the 5ms interval is not too fast for it to handle. To begin measuring the time between pulses, the microcontroller records the time at the first rising edge it sees,

prevTime = current time in milliseconds

then does nothing until it detects another rising edge. The microcontroller then records that time as well,

$$\text{curTime} = \text{current time in milliseconds}$$

and subtracts it from the previously recorded time, to give the time it takes for one blade to traverse the laser's path.

$$\text{timePerBlade} = \text{curTime} - \text{prevTime}$$

Next, since our turbine consists of 15 blades, we multiply the time per blade by 15 to get the time per revolution in milliseconds.

$$\text{timePerRev} = 15 * \text{timePerBlade}$$

We then convert from time per revolution to rpm,

$$rpm = \frac{60 * 1000}{timePerRev}$$

To display this calculated rpm value, we send simple print commands to the LCD screen telling it to print this rpm value. Namely through the lcd.print(variable) command.

With the rpm calculated, we can begin to send the two signals that control the gate voltages of the MOSFETs that control the main and secondary power supplies. It is necessary that those two signals be opposite each other at all times, so that when one is digital HIGH (5 V), the other is necessarily digital LOW (0 V). This ensures that only one power supply is being used at all times, as discussed in section 2.2.2.

The switching occurs when the rpm drops below a certain threshold value, usually chosen to be 75% of the original rpm. This choice is arbitrary and is up to the user to decide. We believe our choice of 75% is reasonable since the turbine is losing more than a quarter of its speed and needs to switch to a functioning power supply that takes it back up to its original mode of operation.

Initially, however, we set the ATmega328P pin connected to the gate pins of the main supply MOSFETs to digital LOW and the pin connected to the gate pins of the secondary supply MOSFETs to HIGH. This turns on the main power supply and turns off the secondary power supply. We have this run for a couple of seconds to ensure that the turbine gets up to its standard rpm upon turning on. The turbine we tested with operates at 800 rpm. The threshold value is therefore 600 rpm in our case.

Once the rpm drops below this value, the gate pins of the main supply MOSFETs receive a digital HIGH and vice versa for the gate pins of the secondary supply MOSFETs. Since the rpm drop indicates a malfunction in the main supply, we need to remain in this state and never revert back to turning on the main supply.

Hardware design details:

The ATmega328P is a very low power device that draws a total of 5mA of current. This is essential to our overall design requirements because the bulk of the system's functionality is being achieved in this module.

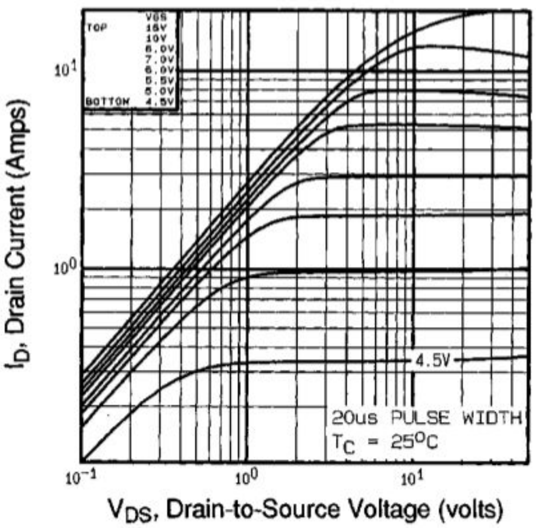
To code the ATmega328P, we used two methods: a USB to TTL Serial adapter with the proper connections, and, more reliably, an Arduino UNO R3 Board with a detachable ATmega328P socket.

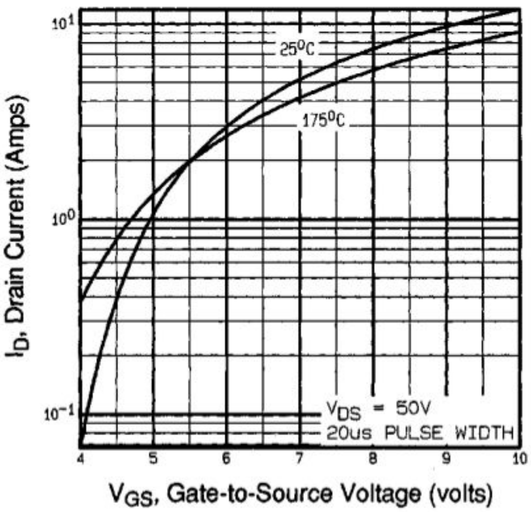
For it to function properly, there has to be a 16 MHz crystal oscillator placed between pins 9 and 10, and two pull-down 22 pF capacitors connecting pins 9 and 10 to ground, respectively. In addition, a 150 kΩ pull-up resistor connects pin 1 to the common 5V supply that powers the microcontroller, LCD display and sensor modules. Whenever the code needs to start over again, this resistor needs to be pulled down and back up again.

3. Verification of Requirements

Module	*Requirement*	*Verification*
Detection	1) Laser receives nominal 10mA from current limiter (J2N3819) and outputs 2mW optical power. 2) Phototransistor saturates at 2mW optical power. 3) Discriminator Circuit gives distinct "high" and "low" 4) No negative voltages being passed to microcontroller. 5) Good coupling efficiency between laser and sensor	1) Ammeter is used to measure the current at the JFET drain/laser diode junction. The 2mW optical power is verified using Keysight 8163xA/B oscilloscope. 2) This is checked by shining the laser on the detector and using a voltmeter to check if the output voltage (going into PMOS gate) is zero. When the light is off the voltmeter should give an open circuit 5V. 3) Voltmeter reading at the PMOS drain terminal gives nominal 4.7V at "light" and 200mV $\pm$ 50 mV at "no-light" conditions. 4) This is verified at the same time as the "no-light" condition. 5) This is done by checking the beam diameter and ensuring that the entire spot is contained within the collecting lens of the BJT phototransistor
Power Switching	1) IRF510 MOSFETS used for M1, M2, M3 and M4 are logic controlled i.e. they switch at low threshold gate voltages (at 4V). These are MC1 and MC2 from controller. 2) The motor never gets more than 12V when both supplies are ON 3) Saturation current for MOSFETS at V_gs = 5V is kept at 800mA so that when one side (MAIN or SECONDARY network) is not supplying, the idle	1) This is verified by supplying 0V and 5V to the MC1 and MC2 nodes respectively (microcontroller temporarily disconnected) . M1 and M3 drain to source current, I_ds, is measured by ammeter and should give 0A. The drain to source current for M2 and M4 should be >0 (in our case this is ~800mA). Finally, MC2 and MC1 gate voltages are switched (5V and 0V respectively) and the opposite behavior is verified. 2) This is verified by disconnecting the microcontroller and supplying 0V through both MC1 and MC2 (both supplies ON). The node between D1 and D2 (output to motor) should read

	current on that side of circuit is not too large. 4) Both the MAIN and SECONDARY are never ON at the same time. 5) High temperature tolerance. The FETS in the design become hot when drawing idle current	~12V. 3) This is verified along with the 1st requirement. The current measured in Verification 1) should be 800mA or less. 4) This verification is done during power switch testing. Instead of diodes we use LEDS as indicators for D1 and D2. This allows us to verify that both LEDS are never simultaneously ON during operation. 5) To check that the MOSFETS are behaving optimally, we replace the motor with an equivalent dummy load and check (with ammeter) if the current is kept at 800mA over a long period of time.
LCD Display	1) Compatible with Arduino 2) Low power consumption	1) Check for 'HD44780 compatible' label upon purchase 2) Use ammeter to ensure that no more than a few milliamps is being drawn
Microcontroller	1) Takes input from detection block and processes it to calculate RPM 2) Prints RPM to LCD Block 3) Initially provides LOW gate voltage for main supply mosfet and HIGH gate voltage for secondary supply mosfet in power switching block 4) Switches to Secondary power supply (providing LOW to secondary supply mosfet and HIGH to main supply mosfet) when RPM drops below threshold. 5) Low power consumption	1) Use Serial Monitor to display calculated rpm values and do debugging of code if rpm values not consistent with what we expect 2) Gate voltage switching verification a) Turn off main supply and wait for rpm to drop. Use serial monitor (or working LCD block) to view this b) Once rpm drops below threshold, observe rpm rising again due to switching gate voltages, indicating power is now given through secondary supply 3) Use an ammeter to verify that microcontroller circuit not drawing more than a few milliamps of current





The above graphs helped us determine the suitability of the IRF510 prior to implementation in the circuit. The verifications tabled above support the expected behavior in the graphs, especially the I_d Vs. V_gs behavior.

4. Cost

Labor costs are calculated as follows:

Per partner = $10/hr * 78 hrs * 2.5 = $1950.00

This gives a total labor cost of **$3900.00**

Part	Cost
BJT Phototransistor	$4.00
Quarton 635 nm red dot Laser diode	$12.00
RioRand LCD Module for Arduino 20 x 4, White on Blue	$7.99
4-pack NEW Atmega328p-pu Chip & DIP IC Socket Bundle w/ Arduino UNO Bootloader	$16.49
IRF510 Power FETs	$12.99
NOCTUA 19cm Brushless Fan	$18.99
Arduino UNO R3	$29.99
FT232RL FTDI USB to TTL Serial Converter Adapter Module Mini USB 3.3V 5.5V	$5.99
TOTAL:	**$108.44**

5. Conclusion

The project was a good way to bridge the gap between theoretical knowledge obtained from class, and real life engineering applications. It was also a good exercise in combining different engineering concepts from different subfields to one cohesive product oriented design. The whole process came with its own set of difficulties such as compatibility issues between components, to circumvent which careful design choices had to be made. Modular testing and overall design testing had to be constantly done every time adjustments were made. Finally, the whole design had to be optimized based on certain priorities, such as motor getting maximum power from the individual power networks, maximum coupling efficiency between laser module and phototransistor, etc. The result was a final working product that accomplishes what it sets out to do - a modular, non-contact, RPM detection scheme and built in backup supply with automatic power switching for when the MAIN source is being depleted.

References

[1] IRF510 Power FET datasheet:
https://www.google.com/search?q=IRF510&ie=utf-8&oe=utf-8

[2] “How to Program ATmega328 Chip Using Arduino IDE.” Performance by Fahim Rahman, Youtube.com, 2 Nov. 2016, www.youtube.com/watch?v=CojmolVruDk.

[3] O'Neill, Brendon. “Pedestal, Desk, Tower & Bladeless Fan Running Costs.” Canstar Blue, 29 June 2017, www.canstarblue.com.au/energy/electricity/portable-fans-running-costs/.
Desk Fan operating at Low/Medium Setting

[4] Ben G. Streetman, Sanjay Bannerjee, “Solid State Electronic Devices” Pearson Publishing (UK 1972)

[5] R. S. Muller and T. I. Kamins, “*Device Electronics for Integrated Circuits*”, 3rd ed., (Wiley, New York 2003)

[6] Mauro Nisoli, “Semiconductor photonics. Principles and Applications”, Società Editrice Esculapio Publishing, (2016)