Humidity Sensing Fan Controller

Hardware Description

Etec 474

05/04/10

Jason Huft
Introduction:

The humidity sensing fan controller uses a few hardware components to function. The main components used are the microcontroller, LCD screen, humidity sensor, solid state relay, a power supply, and a voltage regulator. In addition to this, I use a few mountain switches for input, an internal crystal oscillator, and a microcontroller to control a fan motor unit. This document is intended as an explanation of each parts intended function and requirements.

Microcontroller:

The microcontroller I am using for my schematic is the Freescale 9s08qe32 8-bit brand of microcontroller. I found that the 32-bit series of microcontrollers would drive up cost considerably and provide many unused features. The 9s16 series was also slightly more expensive, and is mostly intended for CANN systems. It is worth mentioning that my demonstration prototype will be the 9s08qe128, which I did most of my testing on. The QE 8-bit brand of microcontroller is powered by a 3.3Vdc input. It has a maximum current draw of 120 mA. The package I have chosen is the 44 pin package. This provides me enough IO for this project. I can’t go lower than 44 pins though, as lower packages do not have the IO I need. Dropping lower than the 44 pin package loses PTD, which I use to keep my LCD in 8 bit mode. I suppose it might be possible to go lower if I switch my LCD to 4 bit mode however, but with the 44 pin package it is not needed. The difference in price between the 44 pin package and the 48 pin package varies, but it appears to be about 50 cents. The difference between the 32 and 44 pin packages appears to be more like a dollar to a dollar and a half.

The MCU does not apply an internal voltage regulator. This means I need to reduce the voltage from my power supply before powering the microcontroller. This increases the cost of the unit. In future revisions of this design, I would look for parts that are all powered by 3.3 volts. This should allow removal of the voltage regulator and a decent reduction of cost. The internal frequency of the microcontroller will be 8Mhz. The MCU also will have an available BDM hookup for reprogram ability.

Display:

I’m using a Newhaven Display LCD screen, specifically the NHD-0108RFZ-RN-YBW . It is 1x8 characters, yellow in color, and requires 5vDC to operate. It will attach the microcontroller through eight display bits, and two control bits. The R/W bit will be tied low, since the LCD screen should never be set into write mode. The LCD bits will only read information from the microcontroller, so hooking this pin to the MCU is unnecessary. This LCD does not come with a backlight, since I decided that was unnecessary power consumption for this project. The contrast pin will be connected to a voltage divider to provide the contrast with 0.6V as specified by the manufacturer.

Relay:

The relay I am using is the AQZ207. The relay has a voltage range on the control from 1.5 – 5V. On the load side of the relay, it is hooked up between wall voltage and the fan. The relay can support a maximum of 200 VAC, and 1A current across the load. This allows for a wide range of possible load voltages; however it will not work in a 240 VAC system. The load side of my relay will be attached to PTB0 on my microcontroller and then to ground. Thusly, if the microcontroller does not send a high output to the relay, the fan will not start.

Power Supply:
My power supply is a linear power supply, part number PS-05-5. It can take in anywhere from 85 VAC – 264 VAC, and output 5VDC at 1 Amp of current. The 5V is useful for powering my LCD and humidity sensor, both of which require that specific voltage. Of course, the microcontroller cannot be powered by a 5V supply, as it requires less. I had to include a voltage regulator to output a 3.3V output for the microcontroller. I decided to go with this supply to ensure reliable operation. However, the cost is around 10$ per unit. I can drastically reduce the price by using a 5V AC/DC converter such as a common wall wart. The price range falls to 5-6 dollars per unit on the lower end, but reliability might be a factor which limits the usefulness of such a part. I have decided to use the PS-05-5 for my schematic, with the idea that further cost saving parts could be used.

Voltage Regulator:

The voltage regulator is a 5 pin package, part number 497-6450-1-ND, that takes anywhere from 1.6 to 16VDC, and will output 3.3V at around 4.8 VDC. The maximum current it can output is 200 mA. Because this regulator is going to be powering my MCU exclusively, this shouldn’t be a problem. SHDN pin is the shutdown pin that will turn off the regulator if tied to ground or anything below .6V. I have tied it directly to 5V from the power supply to keep it tied high. Bypass is tied to ground through a .1uF capacitor to reduce thermal noise. Otherwise, the other pins are Vin, Vout, and ground respectively.

Humidity Sensor:

I am using the HTM1735LF humidity/temperature sensing module. This sensor takes a 5 VDC input on pin 3, and outputs a varying voltage range from 1.5-3.6 VDC from pin 4. Pin 1 is a resistance varying pin that is intended to indicate temperature, and is not connected in this project. The humidity sensor varies in voltage as relative humidity of the surrounding area changes. The humidity sensor has a linear voltage relationship within the range of 10-90% relative humidity. About every 26mA represents another percentage value of humidity. The sensor is intended for 10-90% relative humidity with about 3.5V correlating to 90% RH. It can output slightly higher voltages up to a maximum of 3.6-3.7V. This output voltage will be attached to my microcontroller using one of the AtD pins. But since my microcontroller can only take 3.8 volts maximum, I have passed this output voltage through a voltage divider. This will lower the voltage into the pin to around 3V, which will be within safe levels.

Fan:

The fan I am using for this project is the Air King BFQ90. This is a small fan that is capable of outputting 90 CFM. This means that the fan can circulate 90 cubic feet of air per minute. The fan is also around 2.5 sones which is fairly quiet by fan standard ratings. While the fan is not the main portion of this project, the relay is tied into the hot wire of the fan. This is to allow the relay the switch the fan on/off. The fan can be easily swapped with any other model on the market that draws less than 1A current. The BFQ90 only draws 400mA of current, making it fine for my purpose and prototyping.

Switches:

The switches for this project are rather straightforward. I am using On-Off momentary switches. These three switches are attached to the microcontroller’s GPIO pins and will change states between On/Off/Automatic depending on the user input. This is the only user input my project will receive. The switches can handle upwards of 3A passed through them. This makes them more than useable for my purpose. While not displayed on the schematic, the microcontroller will provide internal pull-ups for these switches, which allows them to be tied directly to ground.
In addition to the switches, the schematic calls for a few extra resistors and capacitors. Most of the capacitors are noise reduction capacitors with the exception of one electrolytic charging cap. The resistors are primarily used in voltage dividers. I have a few dividers to fit the specs of both my LCD contrast and the voltage of the microcontroller. Additionally, I have one ceramic cap to be used near my clock.