**Weather Station Report**

**By:** *Ian Cromheecke, Jimmy Vang*

# **Abstract**

The purpose of this report is to explain how we did our project and how our Weather Station works.

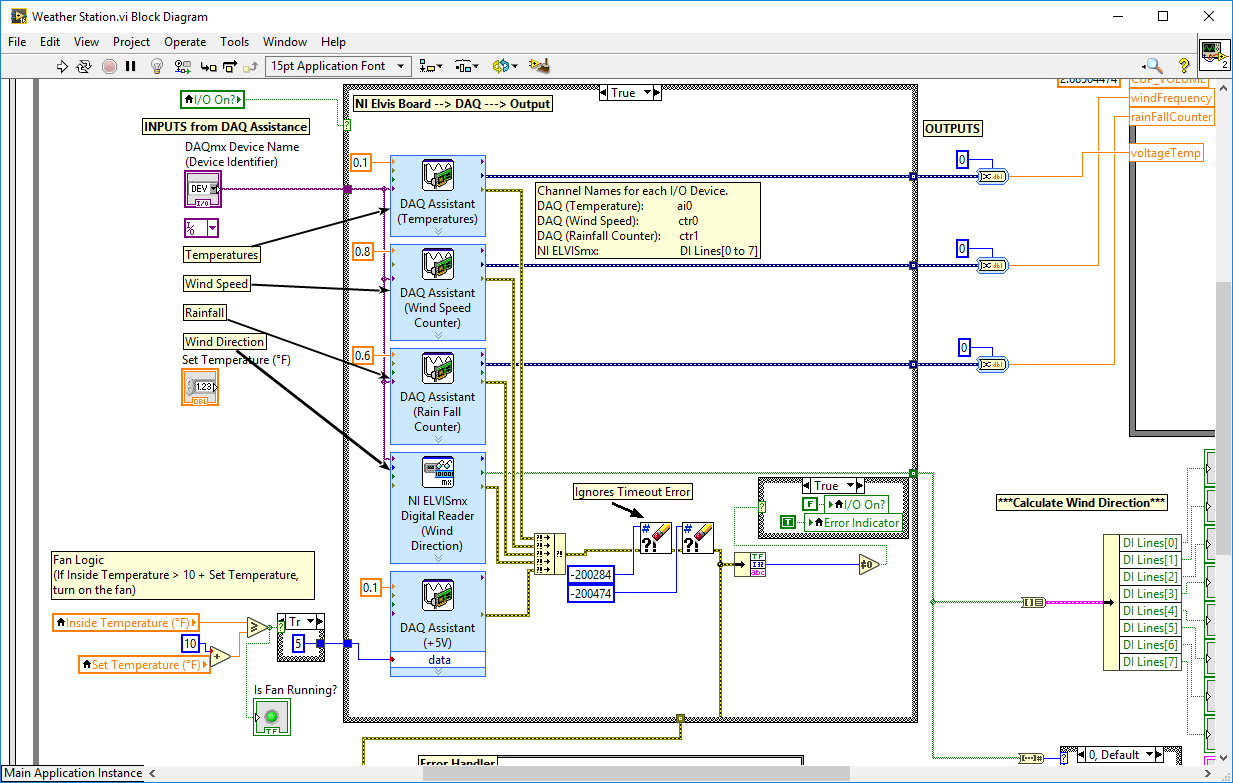
# **Methods**

We used LabVIEW 2016 to create our VI software and while the NI ELVIS board was used to make our physical circuits. To take physical inputs, we used DAQ Assistants. To build the Weather Station, we gathered and bought cups, cardboard, pencils, straws, wood, washers (weight for the rainmeter), and paper clips. We used hot glue and duct tape, given by our instructor, to attach those parts together. The circuits were created using jumper wires, SPST switches, and IC chips. The 74LS14 schmitt trigger was used for the RC switch debouncer circuits. The temperature sensor we used was a LM34 chip. For the fan, it uses a special circuit with the TIP102 chip.

# **How the Weather Station VI works**

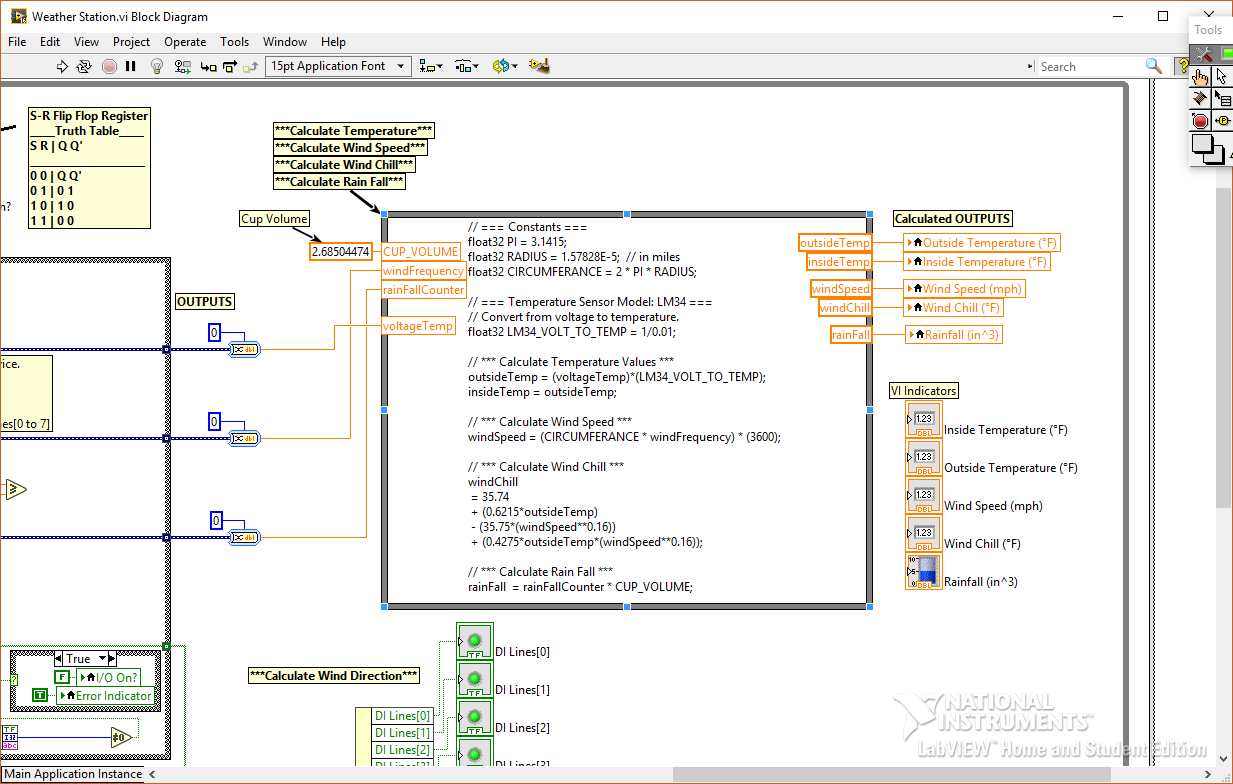
Our Weather Station VI or virtual instrument’s purpose is to mainly display 6 of the following outputs: outside temperature, inside temperature, wind speed, wind chill, wind direction, and rainfall.

**The DAQs**

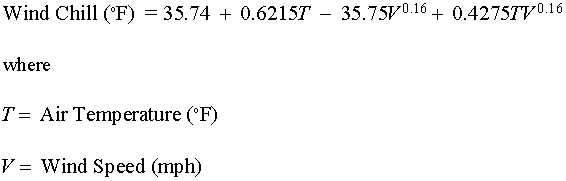


Before we can display anything, we must first get analog inputs from the real world. Our inputs are gathered using special types of circuits. The inputs gathered from the DAQs include temperature, wind speed, and rainfall. These inputs are then converted by DAQs or DAQ Assistants to be usable by the LabVIEW software. Notice that each DAQ gives different input data. The wind chill and direction will be calculated differently next topics. *(DAQs shown above)*

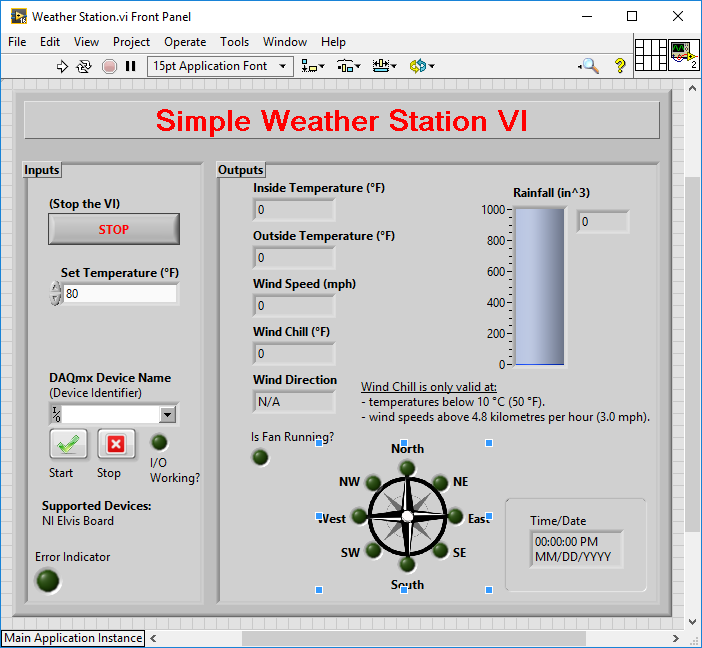
**The Formula Node**



The input data then goes to what is known as a Formula Node. The node takes in each input data from the DAQs, except wind direction. *(Wind chill is calculated in this node using the formula below.)* The node then does specific calculations for each input and finally outputs them. The outputs then travel to their respective local variable, which represents an indicator that will display the appropriate data to the VI front panel. *(Formula Node shown above)*

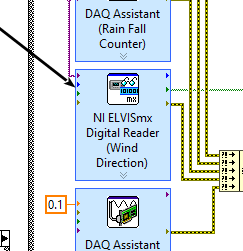


**The Front Panel**

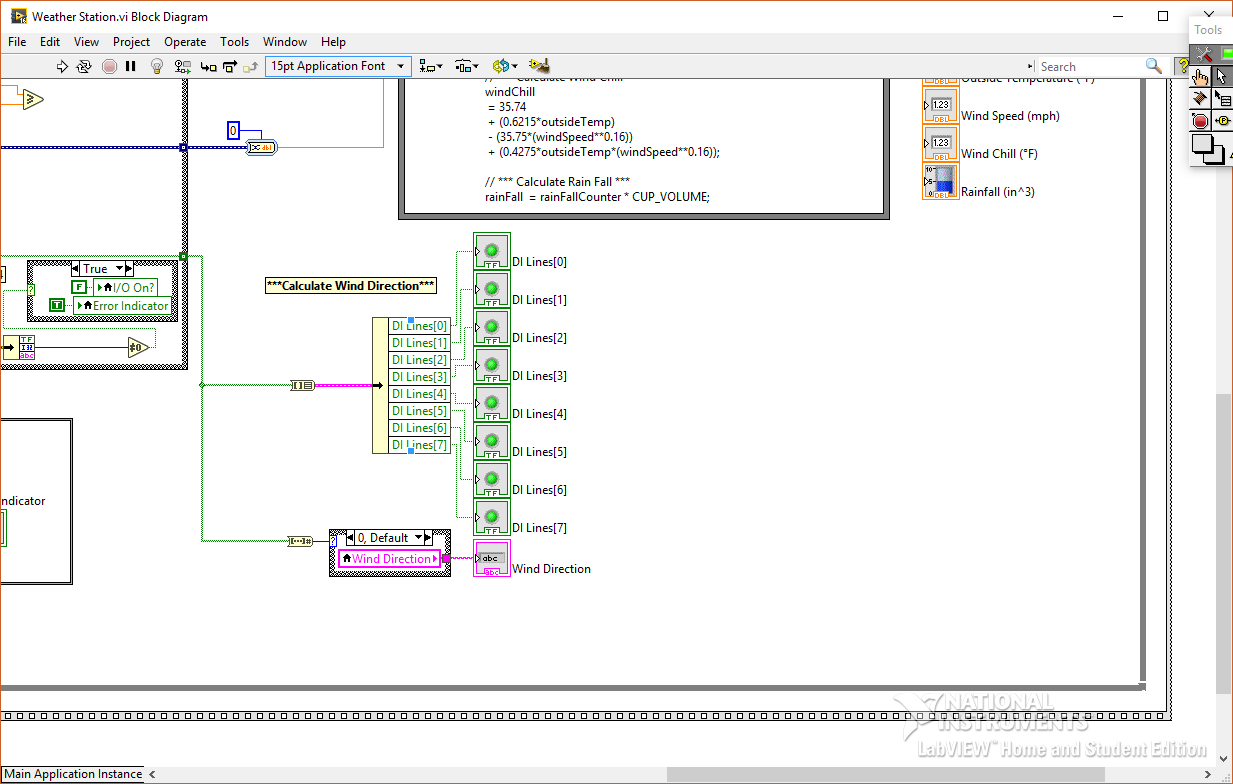


This picture shows what the VI’s front panel looks like. This will display the 6 main outputs mentioned earlier. *(Wind direction is outputted as the Green LED’s and a String.)*

**Wind Direction Calculations**

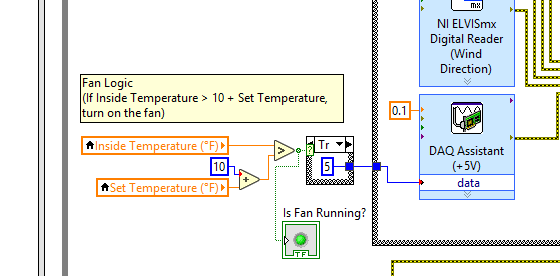


To calculate wind direction, we had to use a different method. Instead of pulling input data from the DAQs, we used the NI ELVISmx Digital Reader. The reader pulls data from 8 different inputs based on the way we set it up. The inputs will come from the NI Elvis board from DI0 to DI7.



The data then gets converted into a visible readable format using the above block diagram. When the switches are wired correctly on the NI ELVIS board, this will display the appropriate LED and message to the VI.

**Fan Logic**



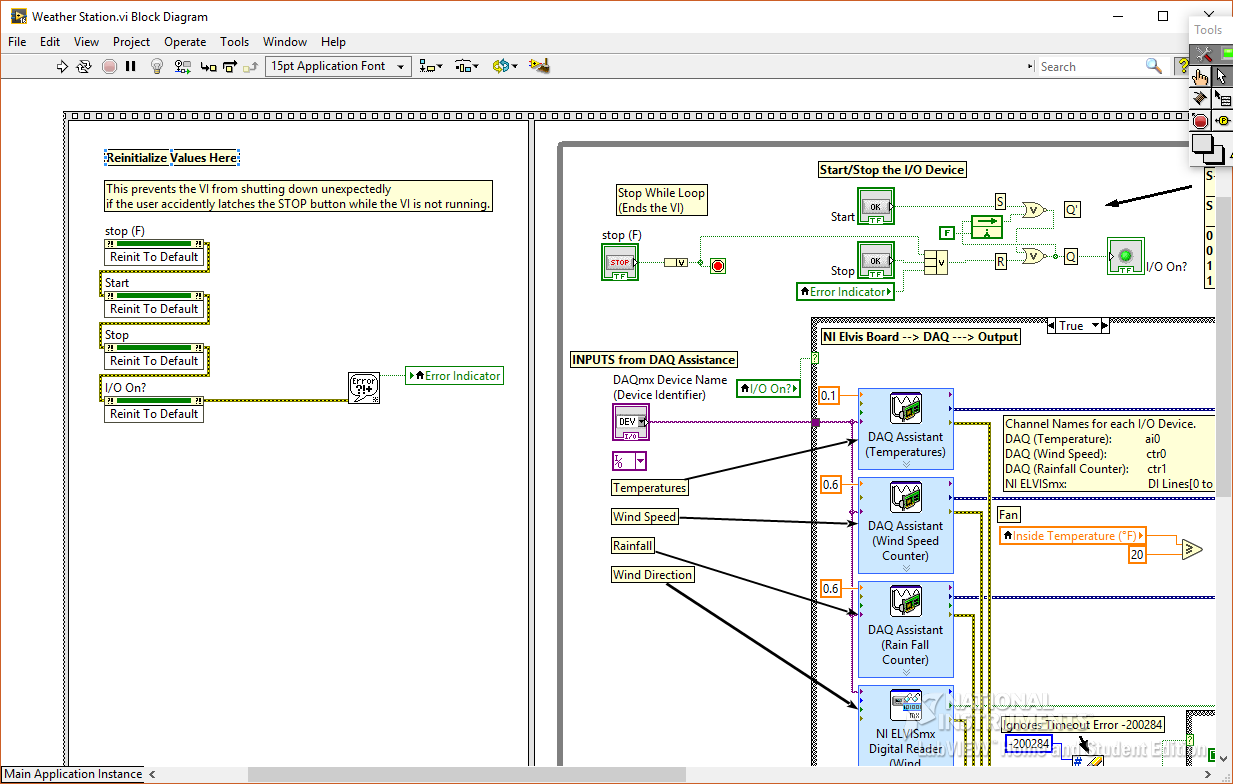
The block diagram above shows the fan logic. To explain, the “Inside Temperature” must be greater than the “Set Temperature” by a difference of 10. Here’s the math equation below:

* **Inside Temperature > Set Temperature + 10**

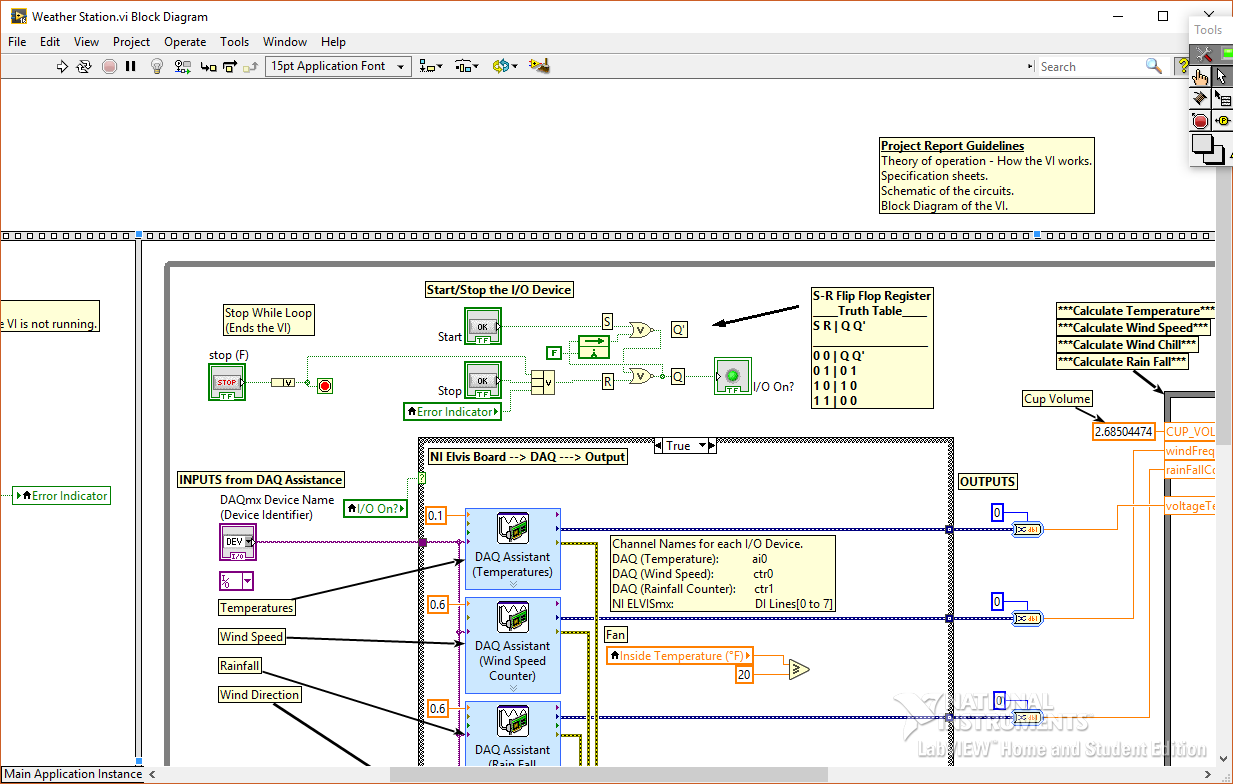
If the equation above is true, a voltage of 5V will be sent to the DAQ Assistant. The NI ELVIS board will output 5V from “ao0” into the Fan Circuit, thus the fan will spin.

**Other parts of the VI**

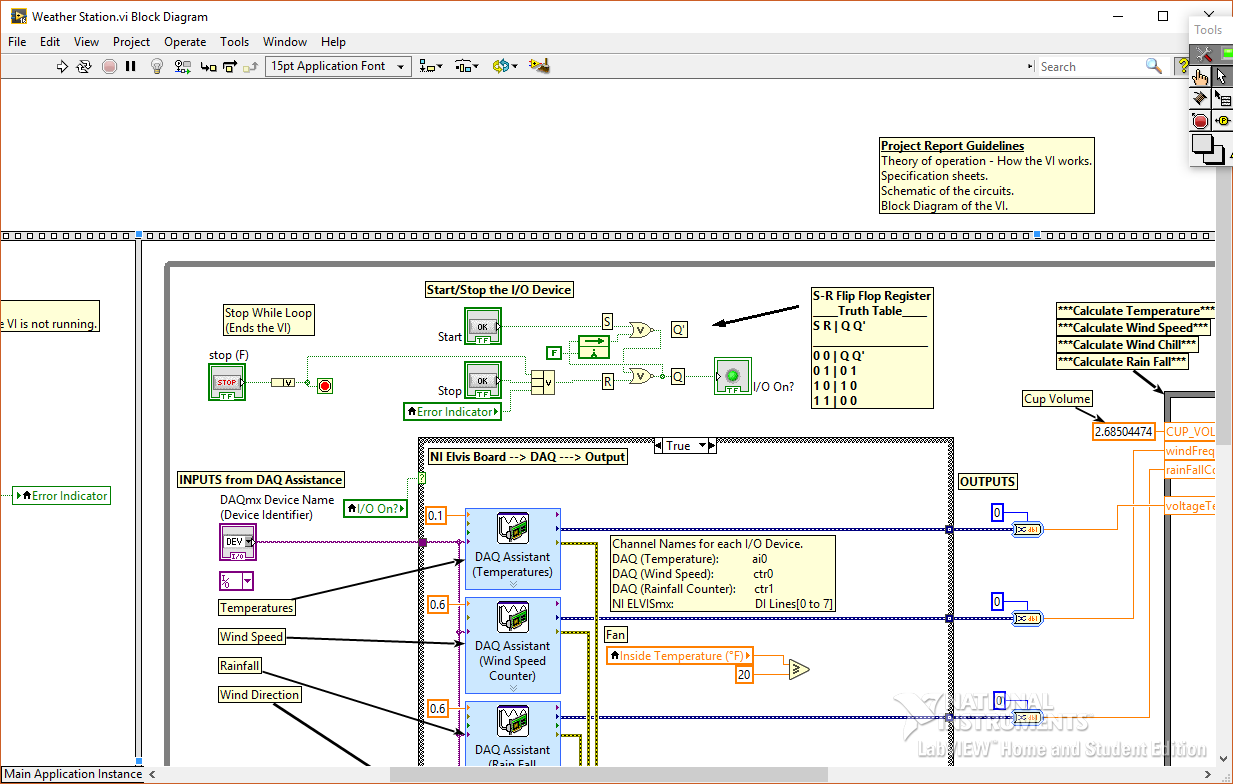
There are other parts of the VI that are not as important, but are still needed to make this VI work properly.



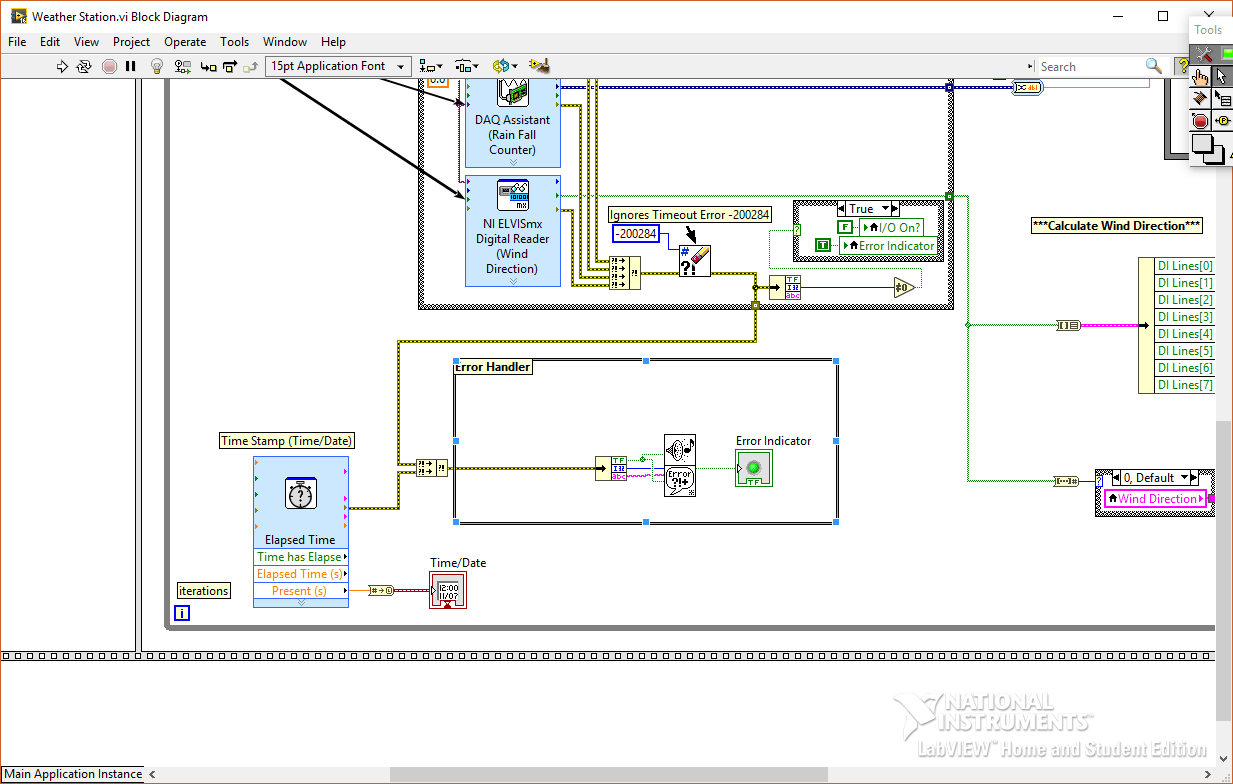
In the block diagram above, this area is needed to prevent unexpected errors. It just basically resets some of the important buttons and LEDS by creating “Invoke Node” properties. It requires a Frame Sequence Structure and must be in the first frame before running the main code and calculations



This block above simply stops the VI by ending the while loop.

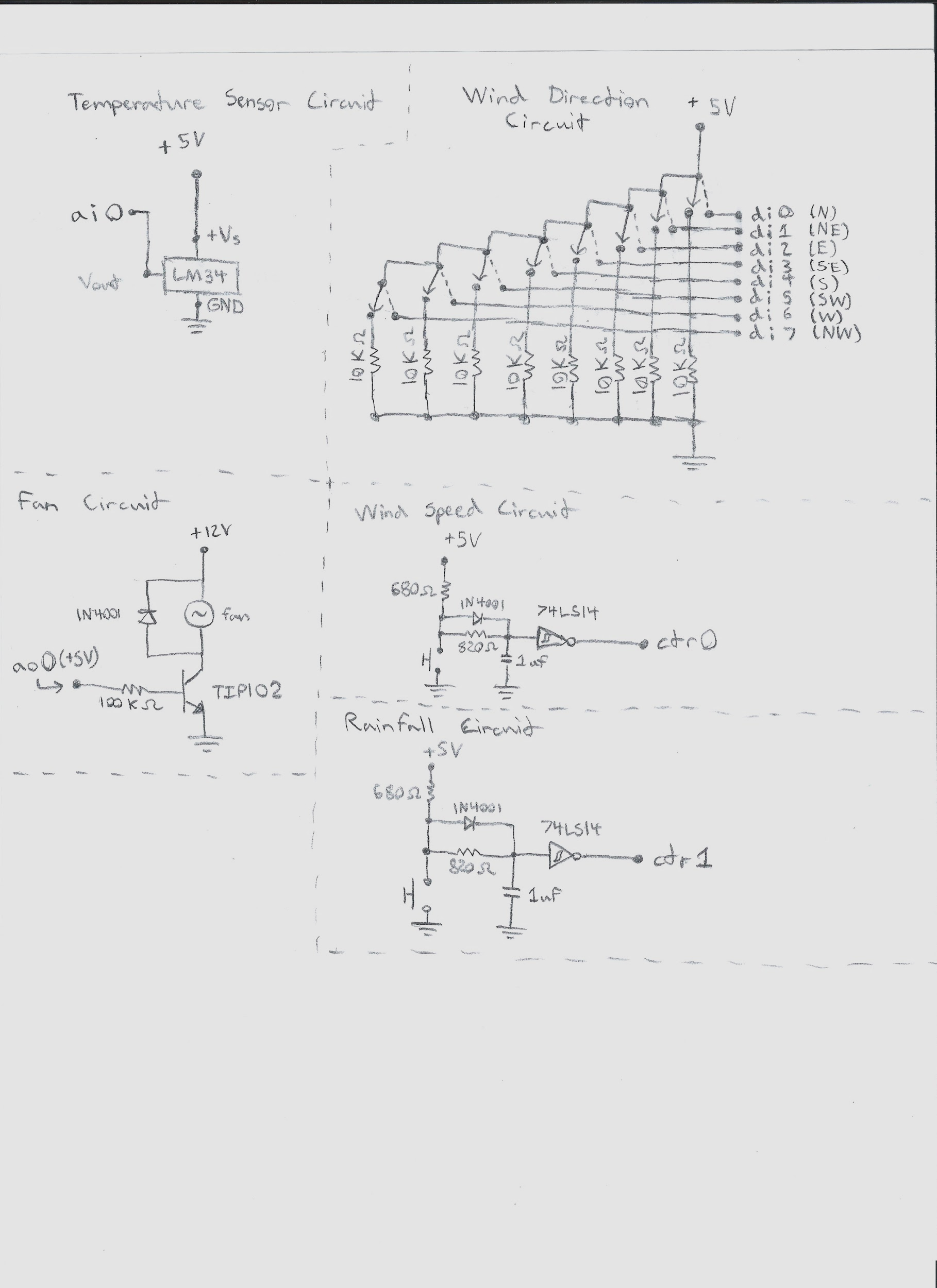


This block above starts and stops the I/O Device (NI ELVIS board). The I/O Device is turned on by the

.

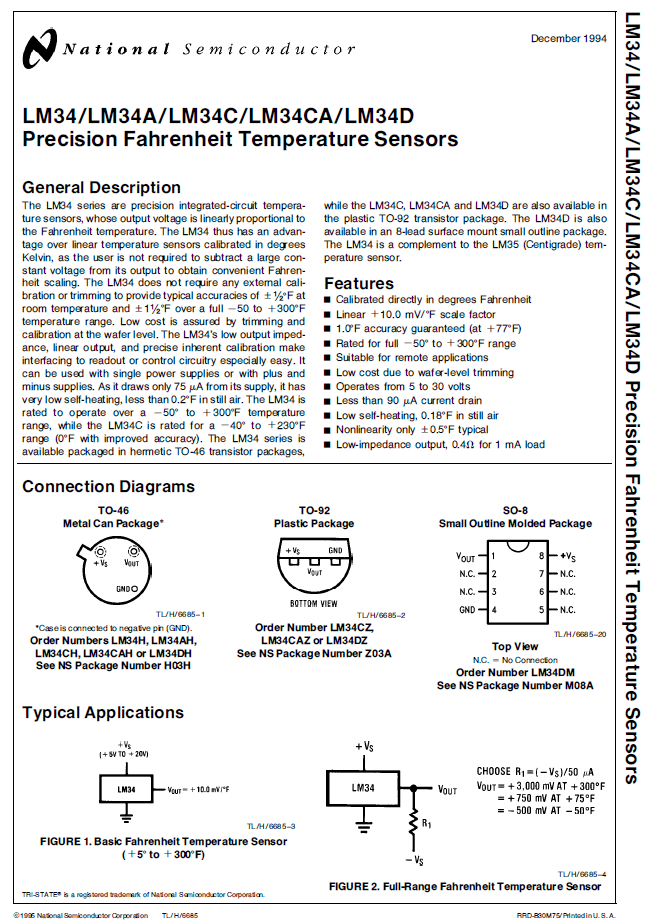
And this is the last part of the VI. The “Elapsed Time”, records the time and date, then displays it. The “Error Handler”, which handles most of the errors produced by the VI and reacts properly.

# **Schematic Diagrams**



**Specification Sheet References**

**LM34 - Temperature Sensor**



**TIP102 - Darlington Transistor**

