WCU Wind Tunnel

July 21, 2025

Contents

1 Introduction

1.1 Overview of System

This document describes the functioning of the WCU Wind Tunnel. This system is capable of up to 70m/s (160 mph) and has a Xm^3 test section. It can gather data on lift, drag, and stall characteristics of models tested, and can serve as an excellent educational aid, and may be useful in advancing various forms of aerodynamic research.

1.2 Nomenclature

Term	Description
VFD	Variable Frequency Device, used to "throttle" the fan. The box next to the fan with the dials.
CAB	"Clear Acrylic Box" housed underneath the table that is the hub between all sensors.
The DAQ	The central hub for all sensor information, populated by "modules" that read and write analog and digital voltage levels.
NI-9239 or "9239"	A module for the DAQ that acts as a multimeter, reading analog DC voltage across its terminals
NI-9219 or "9219"	A module for the DAQ that acts as a multimeter, reading analog DC voltage across its terminals with additional functionality for load cells
NI-9263 or "9263"	A module for the DAQ that acts as a 0-10v power supply, outputting an analog voltage signal
VI	A VI or "Virtual Instrument" is the program created in LabVIEW. Equivalent to .py or .c etc

1.3 Quick Start Guide

While it is recommended that the user read the User Guide in its entirety, this section can be used to quickly understand the basic operation of the system. To get data quickly, the following steps can be used to safely start and operate the system.

2 Developer Guide

2.1 Hardware Guide

The electrical system for the wind tunnel consists of a central hub called the CAB that interfaces with a suite of sensors and the VFD that allows the operation of the system. This section will describe each subsystem in detail.

CAB Overview

The central part of the CAB is the DAQ. This device acts as the firmware between the sensors and LabVIEW. There are three modules used to do this. The NI-9239 analog input module, that acts basically as a multimeter. It has a positive and negative terminal per channel.

The NI-9219. Which is similar to the 9239, having additional functionality for load cell sensors. The NI-9263 analog output module that is used to throttle the VFD, akin to a programmable power-supply. Below is an annotated image of the DAQ, and its associated hardware.

DAQ Hardware Description

As stated above, the brain of this system is the DAQ. This device and its modules acts essentially like a combined multimeter and power supply. It uses these capabilities to monitor and control each individual subsystem. The 9239 module has 2 terminals per channel, labeled 0-1, where 0 is the positive terminal and 1 the negative terminal.
Pressure Transmitters: To read the pressure transmitters this module is required to read in the current outputted from the device, 4-20ma, mapping to -14.7 to 30PSIG. (Note that PSIG reads the relative pressure against ambient room pressure). To accomplish this, shunt resistors of 220Ω are used, and a conversion ma = V/220 * 1000 is done in software. Attached Below is a schematic of the pressure transmitter read circuit.
VFD Frequency In: In addition to reading the pressure transmitters, the DAQ is responsible for reading in frequency data from the VFD on the 9239 using voltage. The VFD can output this information via AO1 (See VFD), in a percentage from $0 - 10V$. This voltage is then multiplied by six in software and outputted as a frequency, giving the user a useful sanity checker and debug tool.
VFD Frequency Out: The DAQ is also capable of outputting a desired voltage via the NI-9263. This is used to send voltage commands to the VFD via AI0 (See VFD) from 0-10v mapping to 0-60hz. The user can either select the option to set a desired percentage of max fan speed (useful for when the system lacks pitot tube readings). There is also an option to set a PID controlled velocity setting (see *TODO SOFTWARE DOCS*).
Sting: The sting currently is manually operated with a level, with three load cells leading into the NI-9219 module, providing useful information to derive lift, drag, and moment. Schematics of this are Below.

VFD
The VFD or Variable Frequency Driver is the throttle for the wind tunnel. It is located on the fan inside the box with the dials and switches above the power switch. This box takes a signal from a potentiometer, either physical or emulated, from $0 - 10v$, and uses this to throttle the fan from $0 - 100\%$ power. There is a big red emergency stop button that will quickly spin the fan down, an on off switch, a potentiometer dial, as well as a switch between "AUTOMATED" and "MANUAL" modes. If switched to MANUAL, the VFD will use the potentiometer physically attached to the box as a throttle. In this mode, there should be an operator at the box at all times to both throttle the fan as well as to monitor the frequency the VFD is currently running at for accuracy. MANUAL mode should only be used if the AUTOMATED mode is not accessible for whatever reason. The AUTOMATED mode uses the controls within LabVIEW to send a voltage $0 - 10v$ directly into the VFD. The user can specify either a frequency $(0 - 60hz)$ or a desired airspeed $(0 - 70m/s)$ and the fan will spin up.

2.2 Programming Guide

Labview
The majority of the functionality of this system is run in a program called LabVIEW. This program is a little cumbersome for anybody with any experience with a traditional programming language, yet was chosen for its ease of entry for a slew of engineers with limited programming knowledge. This section will strive to understand the web of code that is the VI.

Descriptions and Tool-tips
Before I get into detail on the functionality of the VI, note that (most) parts of said VI contain a tool-tip briefly explaining their functionality. These can be accessed via the "Help-Context Help" tab on the left side of the toolbar. Clicking on parts of the VI will show a developer comment on the component as well as a hyperlink to the relevant link on LabVIEW's documentation. **Main Program Loop**
The Wind Tunnel Main VI is (mostly) encapsulated in a while loop. In the VI this consists of a gray box that surrounds most of the components. Everything outside of this loop is ran once on program start, while everything inside is ran throughout the execution of the program. If you are familiar with Arduino you can think of the outside being "void Setup()" and the insides being "void loop()".

Shift Registers
Along the border of the while loops are boxes with a downward facing arrow inside. These components are called Shift Registers and are similar to global variables. During setup, these can be set to a base value. During the execution of the program, the previous data stored is outputted from the left terminal and the right terminal is an input to store the next value. This is seen in the program with the "Tare SR's". There is a boolean controlled by the user that tells the vi to set the value of the shift register to be changed to the current value of the sensor, otherwise it keeps the previous tare. The value of the SR is then subtracted from the unadjusted value read in from the sensors.

DAQ
The DAQ is the tool used by the VI in order to read in raw data from the various sensors and VFD, as well as to output a control voltage to the VFD. The **DAQ Input** component contains the settings for data collection. Double clicking on it will bring up a menu that lists different channels among other information. Each channel is an input from an individual DAQ module, and is outputted as a dynamic signal. This dynamic signal is an array with additional metadata attached. In the VI, this dynamic signal (blue line) is split into two paths. One that is stripped

of this metadata and broken up into its individual parts for more data processing. The other is passed directly into a spreadsheet for raw data collection. The **DAQ Output** is interfaced in the same way as the DAQ Input, it however, controls the output modules.

3 Notes

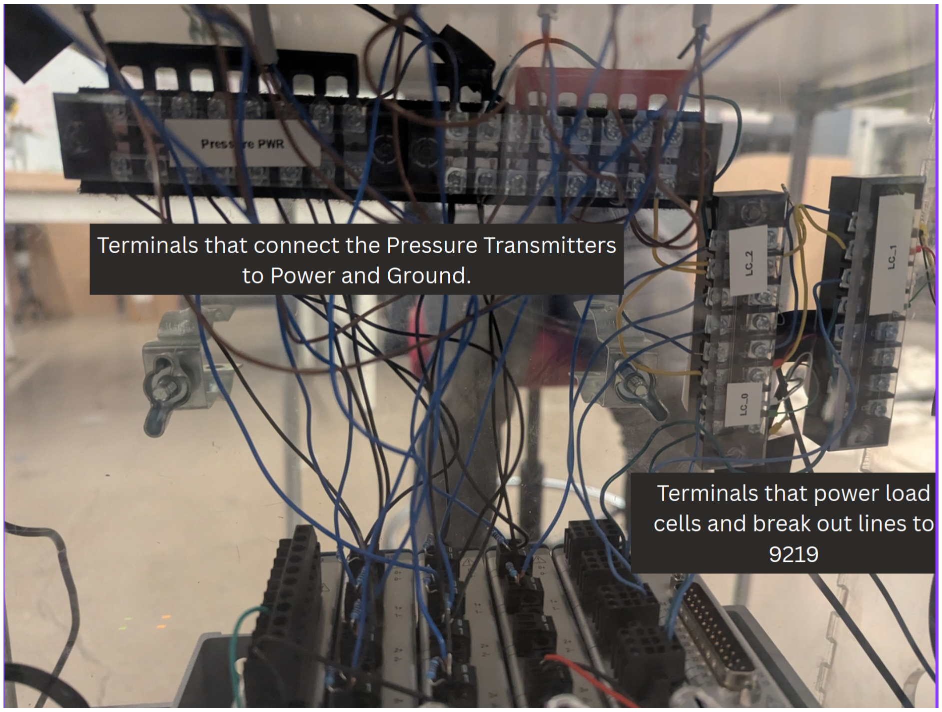

Figure 1: CAB Power Block Terminals

4 Figures

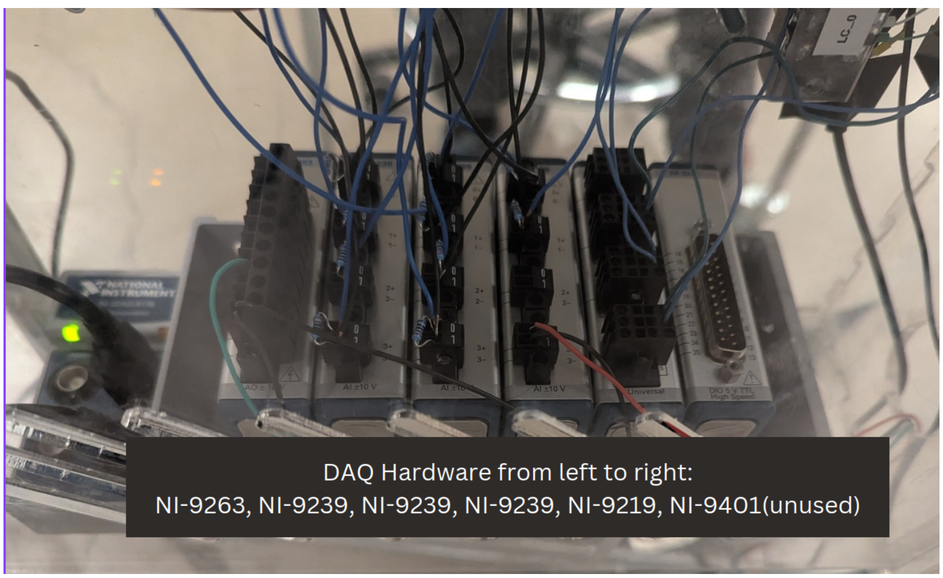

Figure 2: Top down view of DAQ in CAB