

# **Automated Industrial Wind Tunnel Functional Description and Complete System Block Diagram**

Daniel Monahan

Nicholas DeTrempe

Project Advisors

Dr. Aleksander Malinowski

September 20, 2011

The overall goal of this project is to allow for a wind tunnel to be monitored and controlled remotely via the Internet. The wind speed can be modified via a blower and a damper while the position of the target object inside the wind tunnel can be modified via two linear actuators. The wind speed and position of the target should be modified either by the existing manual controls (switches) or via an existing 8051F120 microcontroller unit. The command signals should be delivered to the MCU via a serial connection from a PC located in the laboratory. This PC will also be set up as a server which can be controlled either by a local user or through a remote client. This project is a continuation of previous senior projects and most of the hardware needed has already been purchased.

The actuators will be controlled by H-bridges which are connected to the MCU. The wind tunnel controls are currently connected to steady-state relays which will be controlled through the MCU as well. The lab PC interfaces with the MCU through a serial port and LabVIEW is used for the user interface.

The existing setup had an A/D converter connected to the PC which provided interfacing to the MCU as well as some of the wind tunnel controls. All control signals and sensor measurements will now go through the MCU. Also, some type of switch or set of switches will be needed to allow for easy transfer of control from the server to the manual switches. See Figure 1 for the revised overall block diagram.

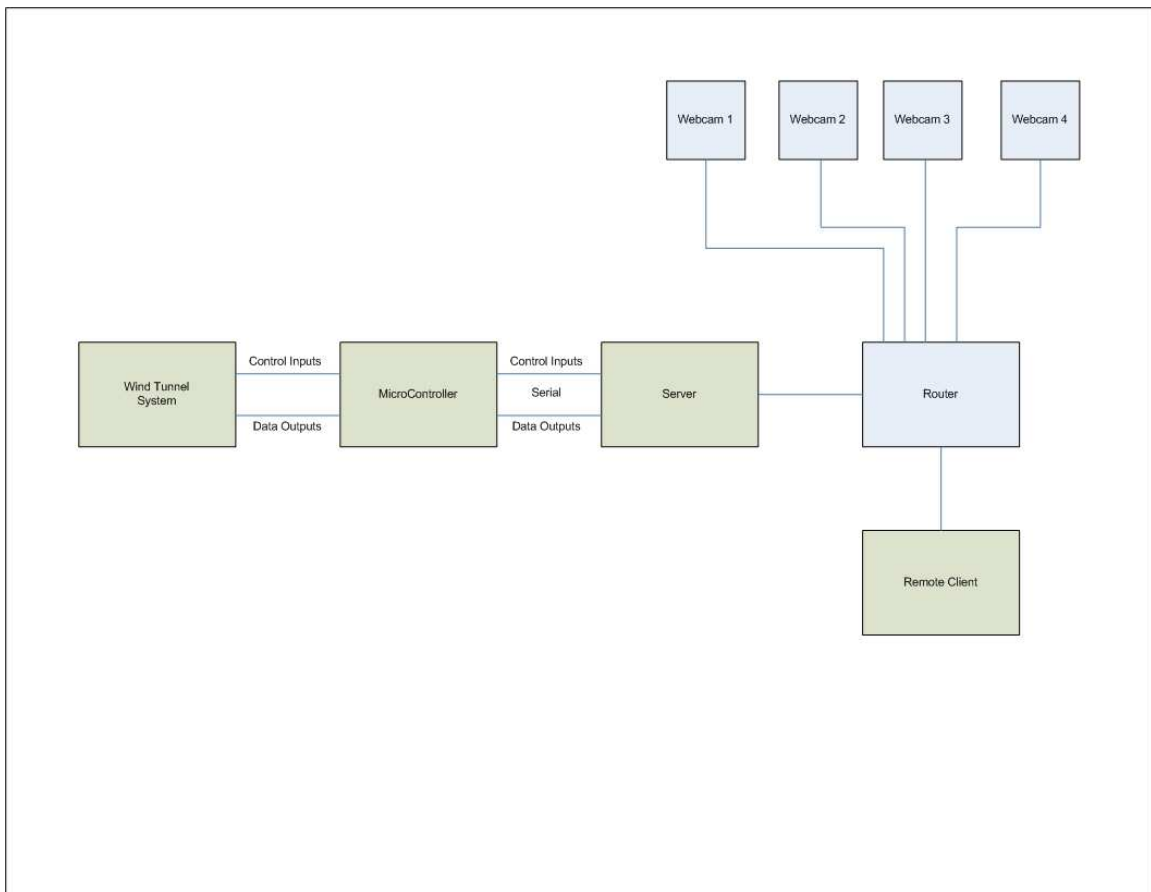


Figure 1 – System Block Diagram

Webcams: Three webcams have already been purchased and are ready to be used. They will be mounted as indicated by the ME department and will be connected to the local router to be accessed by the local PC.

Router: The router will allow for the local network containing the multiple webcams and local PC to be connected to the internet.

Local PC/Server: This PC will serve for the local user interface and act as a server to grant access to remote users. This PC will run LabVIEW software as the user interface to command and read data from the wind tunnel. All command information will be delivered to the C8051F120 microcontroller unit via a serial port. All sensor data will be received from this serial port as well.

Wind Tunnel System: The wind tunnel contains a blower and damper to control wind speed and direction. A pressure sensor inside the tunnel is used to measure the wind speed. Furthermore, two force sensors are used to measure lift and drag on the target object. These three measurements will be needed by users for experimental data collection. Two linear actuators are used to control the position of the target object. These actuators include position sensors that will be used for closed-loop controls but will not likely be needed for user data. See Figure 2 for more details regarding this subsystem.

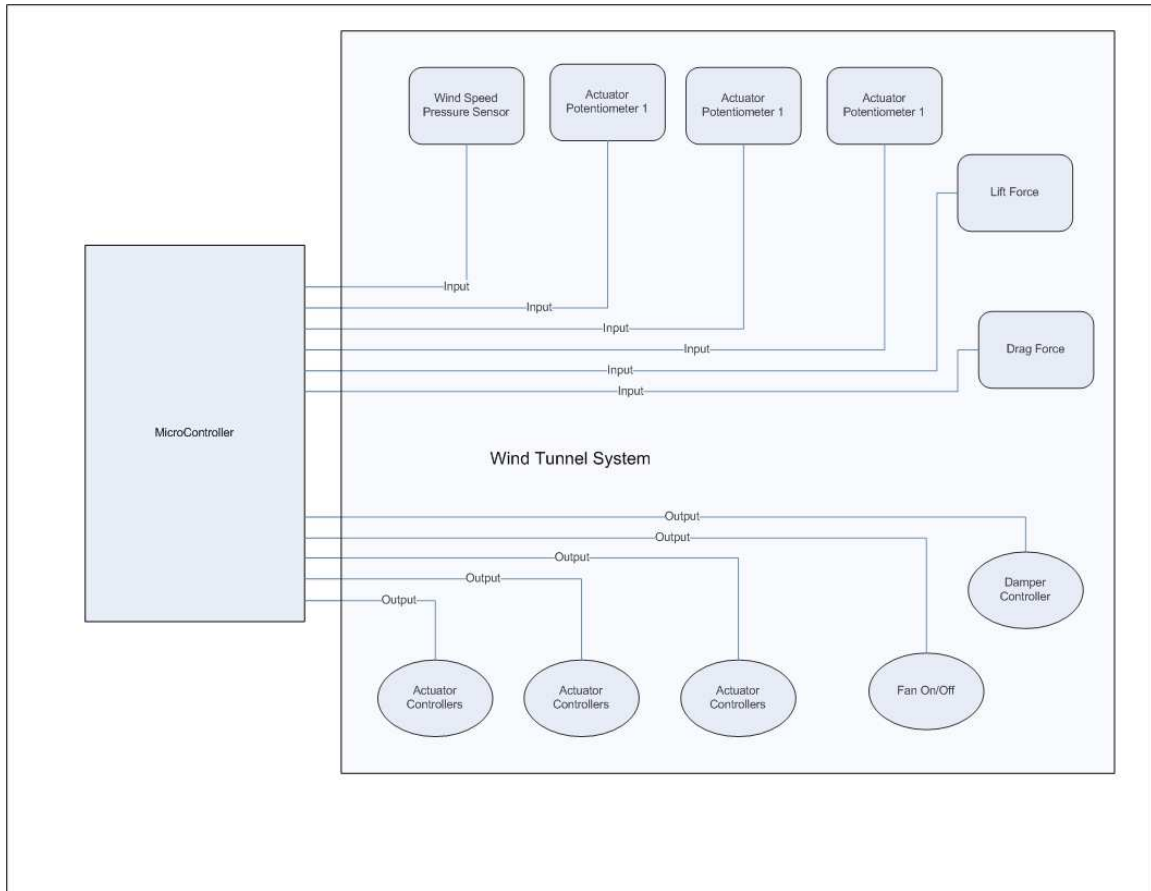


Figure 2 Wind Tunnel Control Block Diagram

Remote Client: The remote PC will be used by a remote user to login to the server via the internet and control and monitor the wind tunnel.

Microcontroller: This MCU will control the blower and damper for the wind tunnel as well as the target actuators. The MCU will also read information regarding actuator position, lift, drag and wind speed from sensors with channels on the A/D converter. The A/D converter contains 8 channels and 6 will be needed. See the figure 3 for more information regarding this subsystem. Figures 3a through 3e show microcontroller software flowcharts.

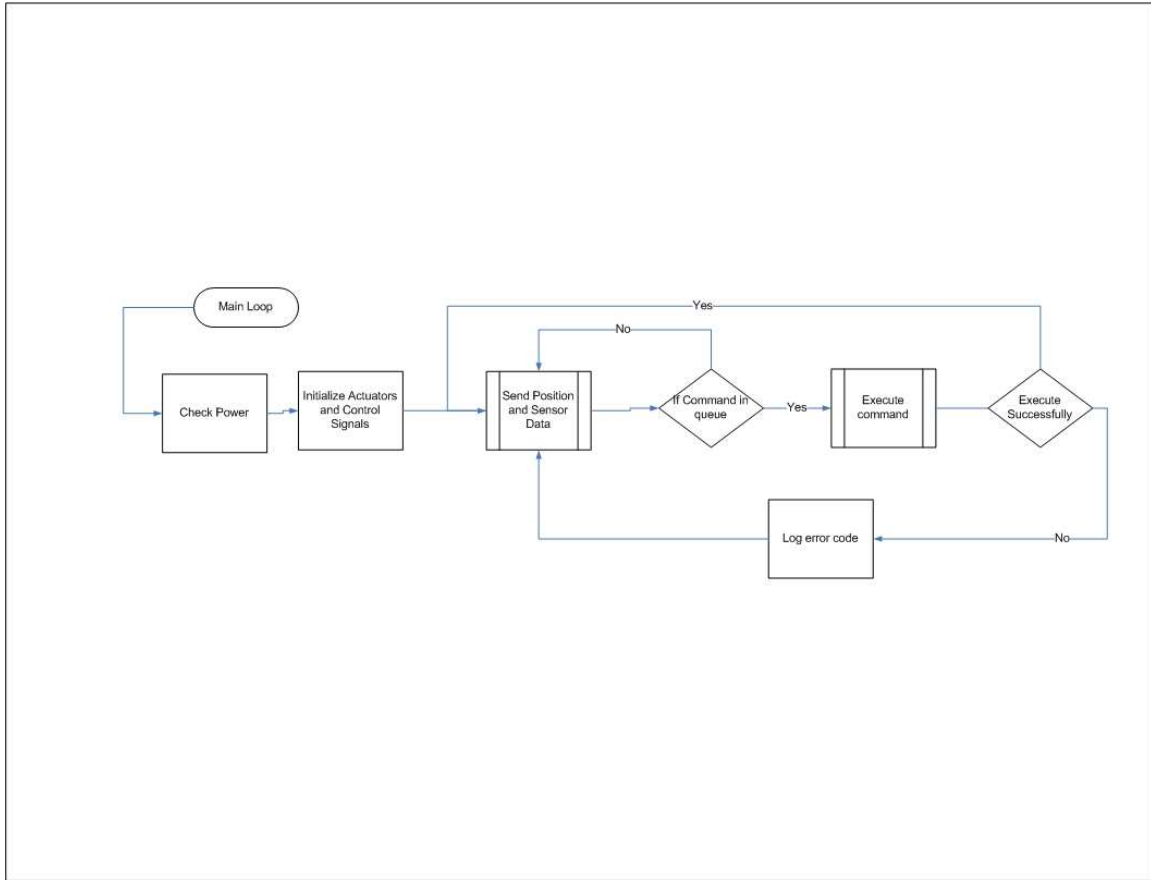


Figure 3a Main Loop for microcontroller

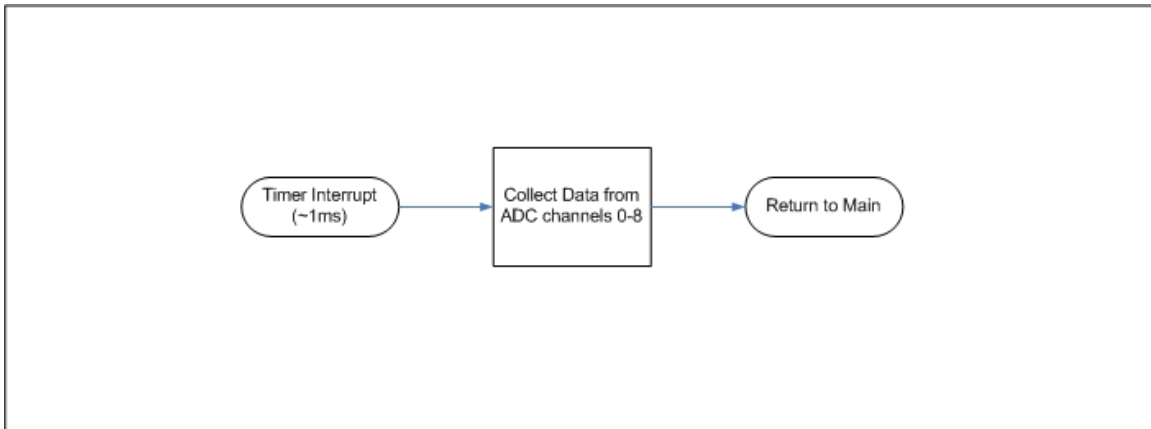


Figure 3b Data collection interrupt.

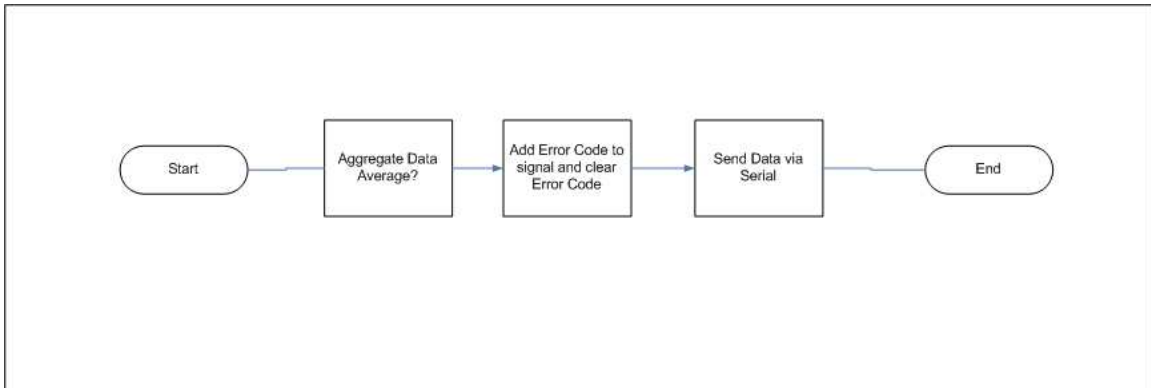


Figure 3c – Data Sending Interrupt

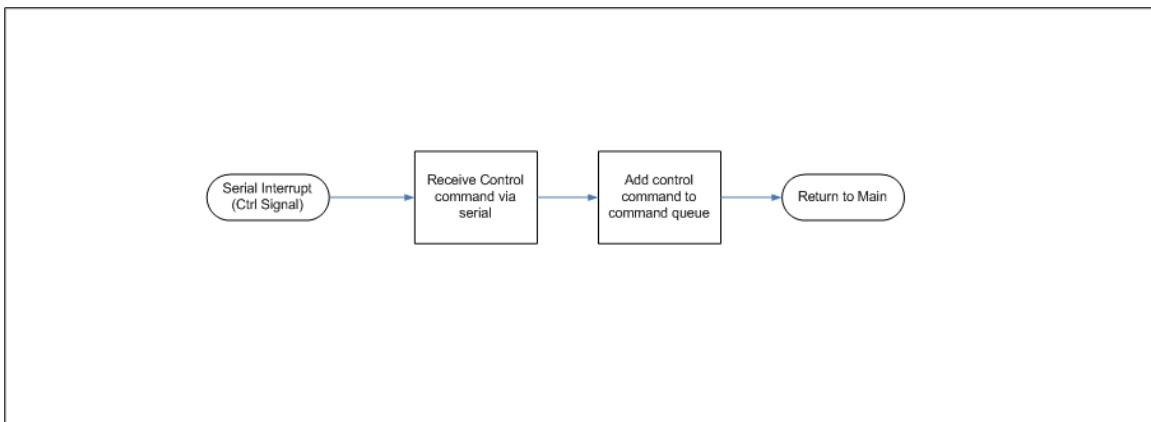


Figure 3d – Command Receive Interrupt

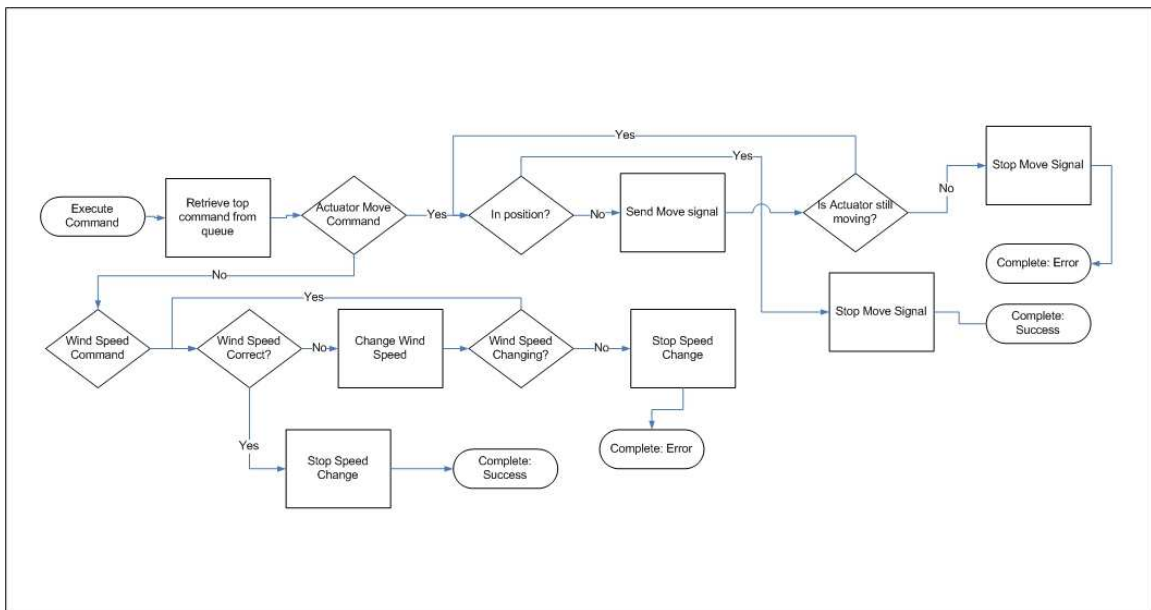


Figure 3e – Process for Controlling the Wind Tunnel

## **References**

Dr. Aleksander Malinowski

Dr Scott Post

Dr. Martin Morris