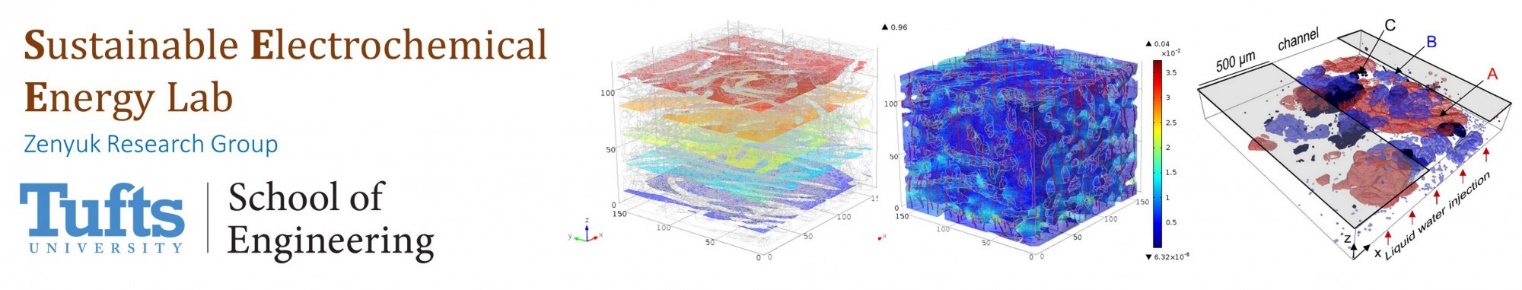
Mass Flow Controller GFC

Control System Operating Manual



# Stanley Normile

# With help from Liam Connolly

Contents

[1. Basic Setup 1](#_Toc454963746)

[1.1 Version Control 1](#_Toc454963747)

[1.2 Software Requirements 1](#_Toc454963748)

[1.3 System Setup 2](#_Toc454963749)

[1.4 Arduino Pairing 2](#_Toc454963750)

[2. Flow Control 3](#_Toc454963751)

[2.1 Setting Flow Rate 3](#_Toc454963752)

[2.2 Dynamic Averaging 3](#_Toc454963753)

[2.3 Data Logging 3](#_Toc454963754)

[3. Hardware Setup 3](#_Toc454963755)

[3.1 Power Supply 3](#_Toc454963756)

[3.2 Arduino Mega 3](#_Toc454963757)

[3.3 Breadboard 3](#_Toc454963758)

[3.4 Mass Flow Controllers 4](#_Toc454963759)

[4. Software 4](#_Toc454963760)

[4.1 Data Logging 5](#_Toc454963761)

[4.2 Control System Functions 5](#_Toc454963762)

[4.2.1 Arduino Setup 5](#_Toc454963763)

[4.2.2 Arduino Control 5](#_Toc454963764)

[4.2.3 Average Flow 5](#_Toc454963765)

[4.2.4 Write Data 5](#_Toc454963766)

[4.2.5 Proportional Control 5](#_Toc454963767)

[4.3 Callbacks 5](#_Toc454963768)

[4.4 User Interface Figure 5](#_Toc454963769)

[Appendix 6](#_Toc454963770)

# 1. Basic Setup

## 1.1 Version Control

As new features are added, they are added to the GitHub in a new branch. When a branch has been shown to be stable, it is merged into the master branch.

## 1.2 Software Requirements

This program runs on Matlab using Arduino Support for Matlab. Arduino support can be downloaded at <http://www.mathworks.com/hardware-support/arduino-matlab.html> Note Simulink support is not required. In order for the data logging to execute, Microsoft Excel must be installed as well. The files required are Control\_ui.m and Control\_ui.fig. They must be in the same directory which must be selected as your current working directory. The data log will be automatically created in that same directory with the file name Flow\_Data\_date\_time.

## 1.3 System Setup

1. Attach the hoses to the Mass Flow Controllers
2. Plug the 15 pin connectors into the Mass Flow Controllers
3. Plug the control box into 120V AC power
4. Plug the USB into the Computer

## 1.4 Arduino Pairing

Connect the control box to the computer via USB. When the run button is pressed, the screen below pops up with the defaults already selected. By default, the Auto Detect Arduino feature is enabled, thus the Board and COM port need not be specified. Click connect to connect to the Arduino. Wait for the pop up that says “Arduino Ready” - this may take up to a minute.

If the Arduino fails to pair, an error message will be displayed in the Matlab Command Window. Uncheck the Auto Detect Arduino and specify the Arduino board type and COM port in the appropriate windows. Note: COM port must be formatted COM#. The appropriate COM port may be found in Settings>Devices>Connected Devices on Windows machines.

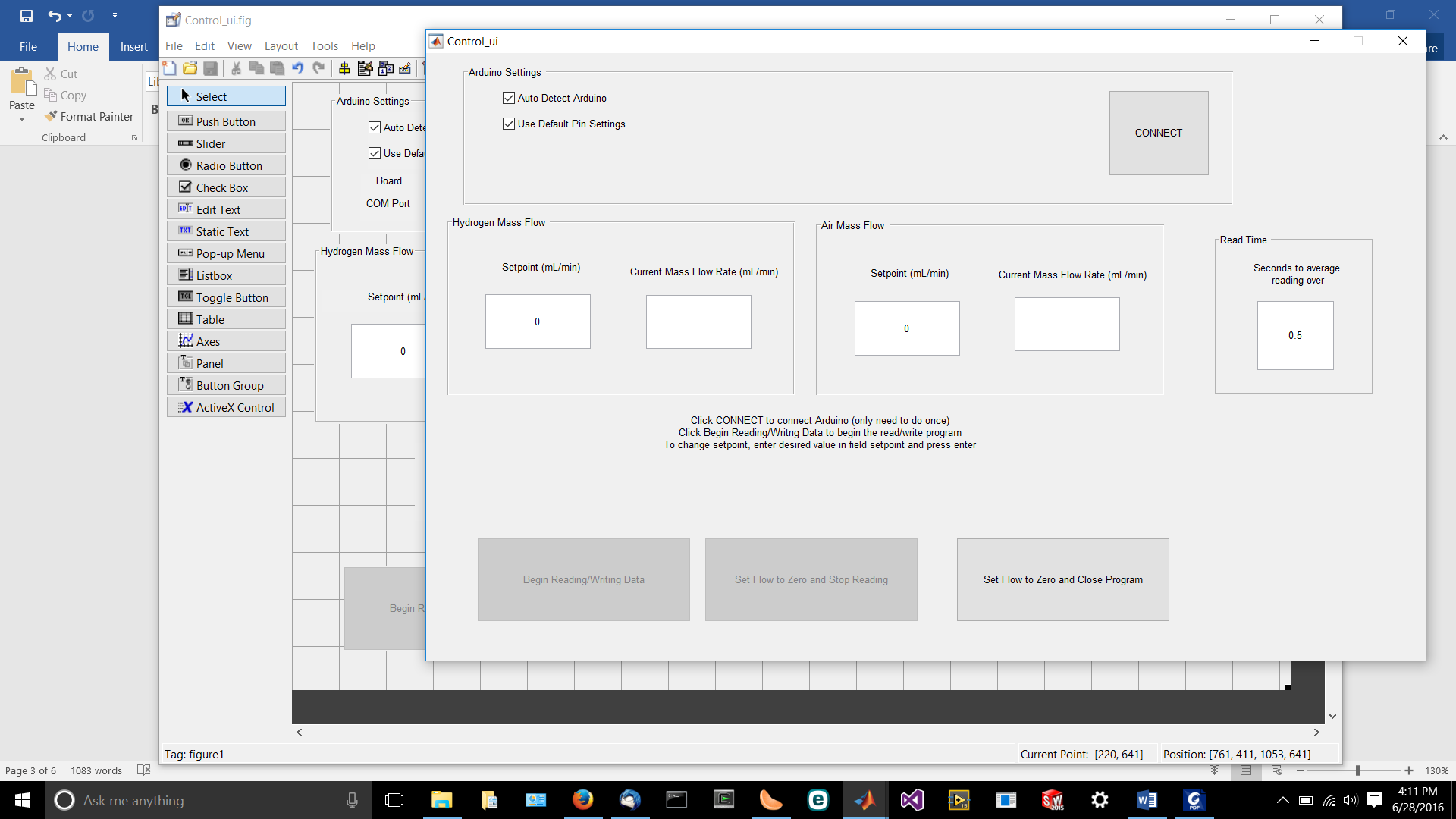


Figure 1. User Interface

# 2. Flow Control

## 2.1 Setting Flow Rate

Press the Begin Reading/Writing Data button to begin reading data from and writing data to the mass flow controllers

For each gas, the user interface displays a setpoint and the current mass flow rate in mL/min. To change the setpoint, edit the appropriate box and hit enter on your keyboard.

## 2.2 Dynamic Averaging

The program reads the mass flow data from the controller over a specified period of time and performs a dynamic average which it then displays to the screen and writes to the data file. By default, that period is 0.5 seconds. In order to change this, a new value must be added **before** the read/write button is pressed.

## 2.3 Data Logging

The program logs data into an Excel sheet that is auto generated when the program is started with the file name Flow\_Data\_date\_time.xlsx. Once per specified time period (see 2.2), the program adds the current mass flow rate and the setpoint for each controller along with a time stamp to the Excel sheet. Note: this functionality will not work if Microsoft Excel is not installed on the computer.

# 3. Hardware Setup

## 3.1 Power Supply

The power supply converts 120 VAC into 12 VDC which powers both mass flow controllers and the Arduino. It is calibrated to output exactly 12V. Calibration can be done using the +V ADJ screw located to the right of the screw terminals.

## 3.2 Arduino Mega

This system is controlled by an Arduino Mega 2560. While Arduinos can be run directly off of the power they get from the USB connection, they sometimes will not have enough power to drive all of their pins to the specified levels. In order to avoid this, the Arduino is powered with 12V from the power supply. The Arduino is equipped with an on-board regulator which converts the 12V down to 5V.

## 3.3 Breadboard

The mass flow controllers accept control in the form of a 0-5 VDC analog signal. In order to achieve this, the PWM signal from the Arduino must be sent through a low pass filter. Since the signal should be very stable and a fast response is not required, the filter uses a 100 µF capacitor and a 1 KΩ resistor wired as shown below.

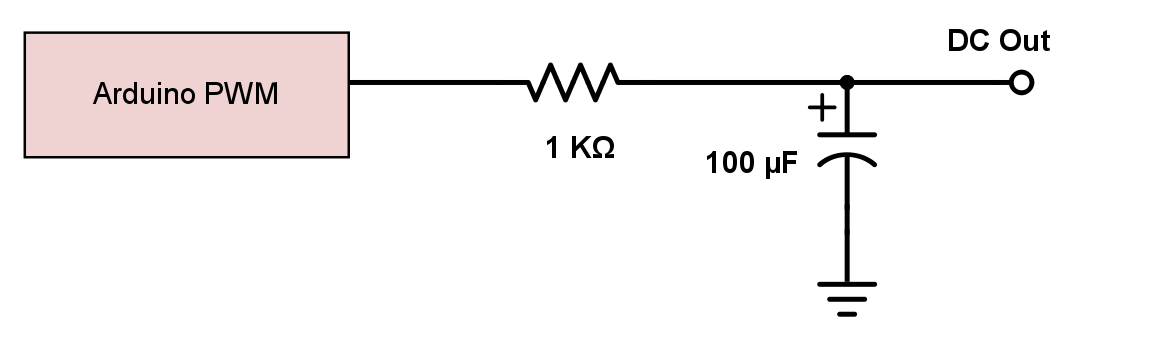


Figure 2. Low Pass Filter

## 3.4 Mass Flow Controllers

Each mass flow controller runs off of a 15 pin D connector. The pin connections are listed in the table below; all unlisted pins are not connected. For a more detailed pinout list, please refer to the Mass Flow Controller Operating Manual. For a more detailed schematic, refer to the appendix.

|  |  |  |  |
| --- | --- | --- | --- |
| Pin Number | Pin Function | Hydrogen GFC Connection | Air GFC Connection |
| 1 | 0-5 VDC Flow Signal Common | Arduino GND  (Green) | Arduino GND  (Green) |
| 2 | 0-5 VDC Flow Signal Output | Arduino Pin A0  (Orange) | Arduino Pin A1  (Red) |
| 5 | Common, Power Supply | Power Supply -V  (Black) | Power Supply -V  (Black) |
| 7 | +12 VDC Power Supply | Power Supply +V  (Red) | Power Supply +V  (Red) |
| 8 | Remote Setpoint Input | Arduino Pin 2 through Low Pass Filter  (White) | Arduino Pin 3 through Low Pass Filter  (Blue) |
| 10 | Remote Setpoint Common | Arduino GND  (Blue) | Arduino GND  (Green) |

Figure 3. Pin Diagram

# 4. Software

This provides a brief overview of the control code. For more detail, please refer to the comments in the code itself.

## 4.1 Data Logging

The file name for the logged data is generated on line 67 in the function Control\_ui\_OpeningFcn. To change the output file name or location, edit this line.

## 4.2 Control System Functions

### 4.2.1 Arduino Setup

Initializes connection with Arduino and sets pin modes. To edit pin assignments, run the program and uncheck the box “Use Default Pin Settings.” This will make the pin assignments editable from the UI.

### 4.2.2 Arduino Control

This is the main loop that runs continuously to relay information between the Arduino and the UI.

### 4.2.3 Average Flow

Collects the flow data for a specified period of time and averages it.

### 4.2.4 Write Data

Writes the data to the log file.

### 4.2.5 Proportional Control

Implements proportional control.

## 4.3 Callbacks

These functions create and execute the UI objects. They are mostly auto-generated and all they do is call the Control System Functions.

## 4.4 User Interface Figure

The Matlab file Control\_ui.fig contains the control UI figure. To edit, open it using GUIDE by type guide in the Matlab Command Window.

# Appendix

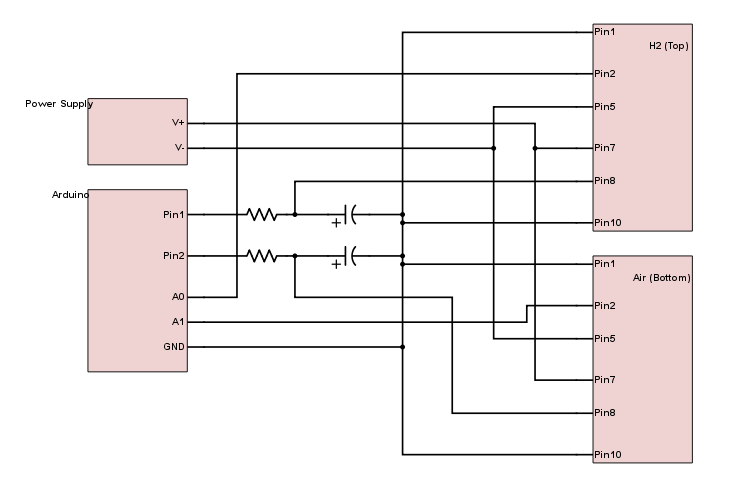


Figure . Full Wiring Diagram