## Servo Control Panel

The idea is to realize an instrument panel to control a servo and display the angle and sweep mode on a LCD screen. Although compact and cheap boards dedicated to servo testing do exist, I wanted to realize a device to control and display the flap deflection of a wing model in a little, didactic wind tunnel and I wanted to do it, as far as possible, with the items provided in the Arduino Starter Kit.

Usually, servo testers give the possibility to control the servo angle by rotating a knob, put the servo in the neutral position, and sweep automatically between the minimum and maximum angle positions. Since my objective is to provide a user interface for wing angle of attack variation or control surface deflection (Fig. 1), I do not need to put the servo in the neutral position (i.e. 90° if the servo rotates between 0° and 180°), but I need a switch to set a reference zero angle. In fact, the servo may be at any position while the wing (or control surface) angle relative to the wind is null. Also, I am interested to measure the deflection of the control surface relative to the reference zero. Finally, I want the servo to automatically sweep between two given angles around the reference zero at a certain angular velocity, which I want to control, although I am not interested in measuring it (I just want it to rotate faster or slower).

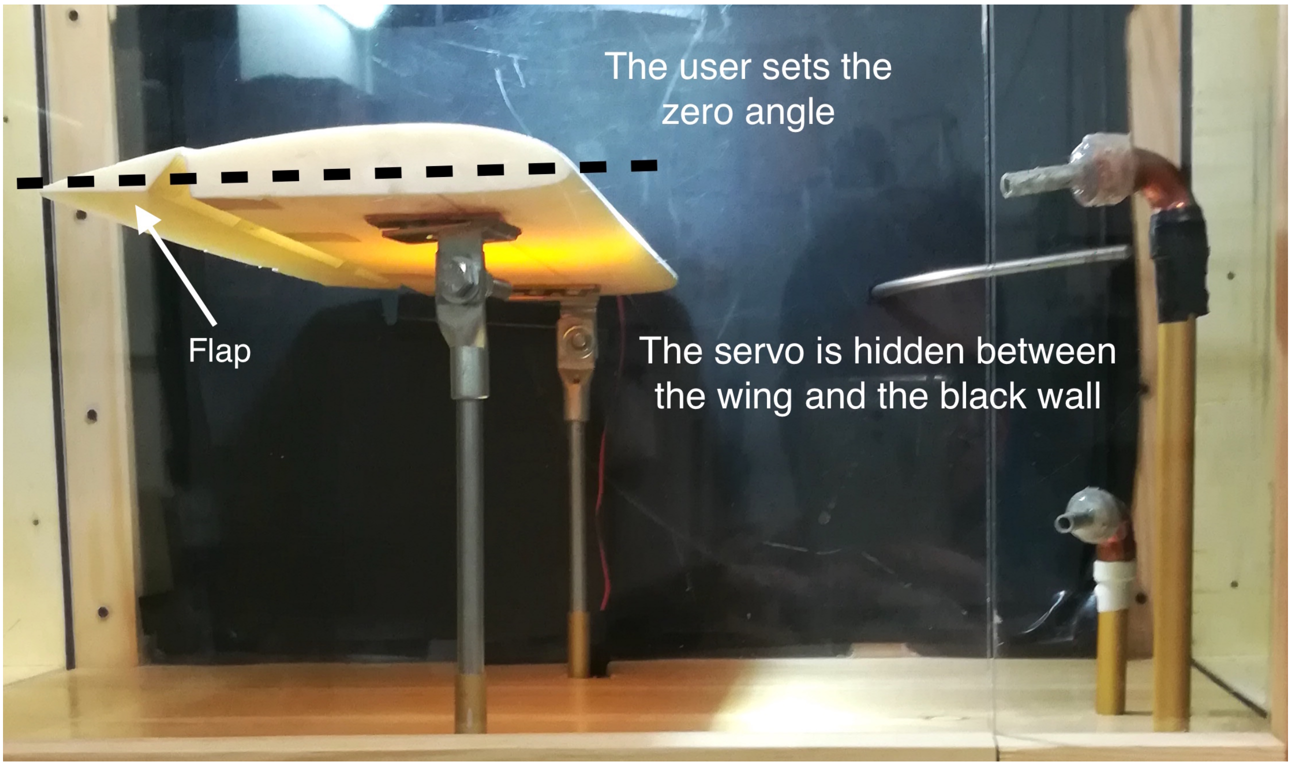
The user sets the zero angle
Flap
The servo is hidden between the wing and the black wall

Fig. 1 – A wing installation in a wind tunnel. This tutorial is about the control panel not shown in this picture, but the description of the apparatus explains the motivations of the control panel functionality.

In my case, I used a little servo with JR connector, I assumed that the flap angle is equal to the servo angle (this is a rough estimate), and the auto-sweep function has been set between -30° and +60° (positive values, means increase in lift force: flap rotates down, the aerodynamic force increases upwards.

The panel operates in two modes: manual control and auto sweep. The user interface provides three buttons, a knob, and a LCD screen. The display shows the servo angle in manual control mode and the limit angles (set in the Arduino code and not editable from the panel) in auto-sweep mode. The switch buttons serve as zeroing the reference angle, commuting between the manual control and auto-sweep mode, and changes the sign of the angle reported on the LCD display. In fact, since the servo rotation may be opposite to the desired convention, the last switch button gives the possibility to change the sign on the screen, but does not invert the rotation of the servo, which is linked to the rotation of the knob. In manual control mode, the servo knob controls the servo angle, which is shown on the LCD screen. Initially, the display shows the servo absolute angle, after the zeroing it shows the servo angle relative to that zero. In the auto-sweep mode, the servo knob controls the rotation rate by adjusting the delay between consecutive rotations (between 15 and 100 ms). Breadboard connections and interactive actions are shown in Fig. 2.

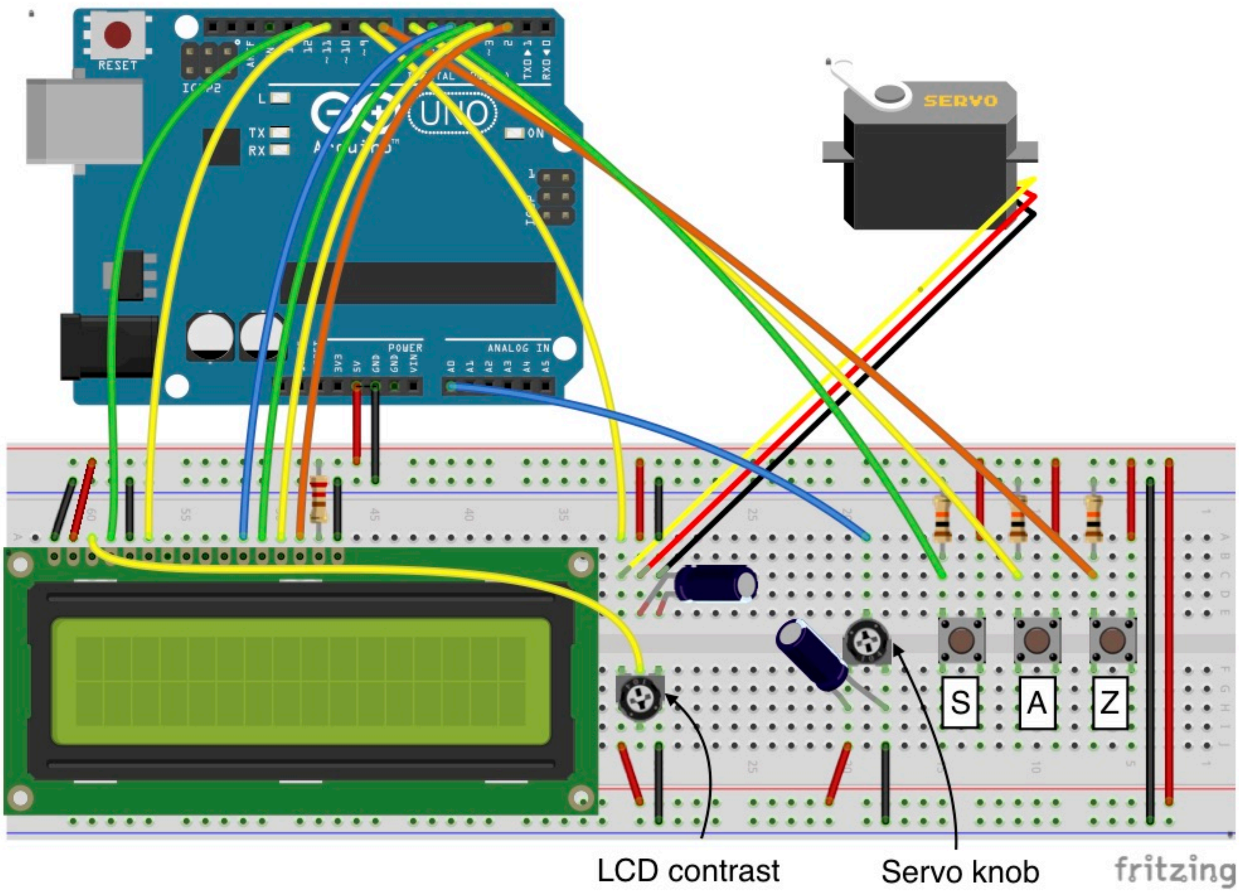

Fig. 2 – Breadboard connections for the Servo Control Panel. The S button serves to change the sign of the angle shown on the LCD screen. The A button commutes between manual control and auto-sweep. The Z button set the actual servo angle as zero. The servo knob controls the servo angle in manual mode and rotation rate in auto-sweep mode.

I have managed to use items from the Arduino Starter Kit except for the bigger breadboard. The panel shown in Fig. 3 has a slightly different arrangement from the breadboard view of Fig. 2 for assembly reasons. Consider that you may need an external power source to control the servo, according to the voltage and power consumption of your device. If the servo vibrates, check your connections, then try to add an external power source. You may need to rearrange some of the items on your breadboard.

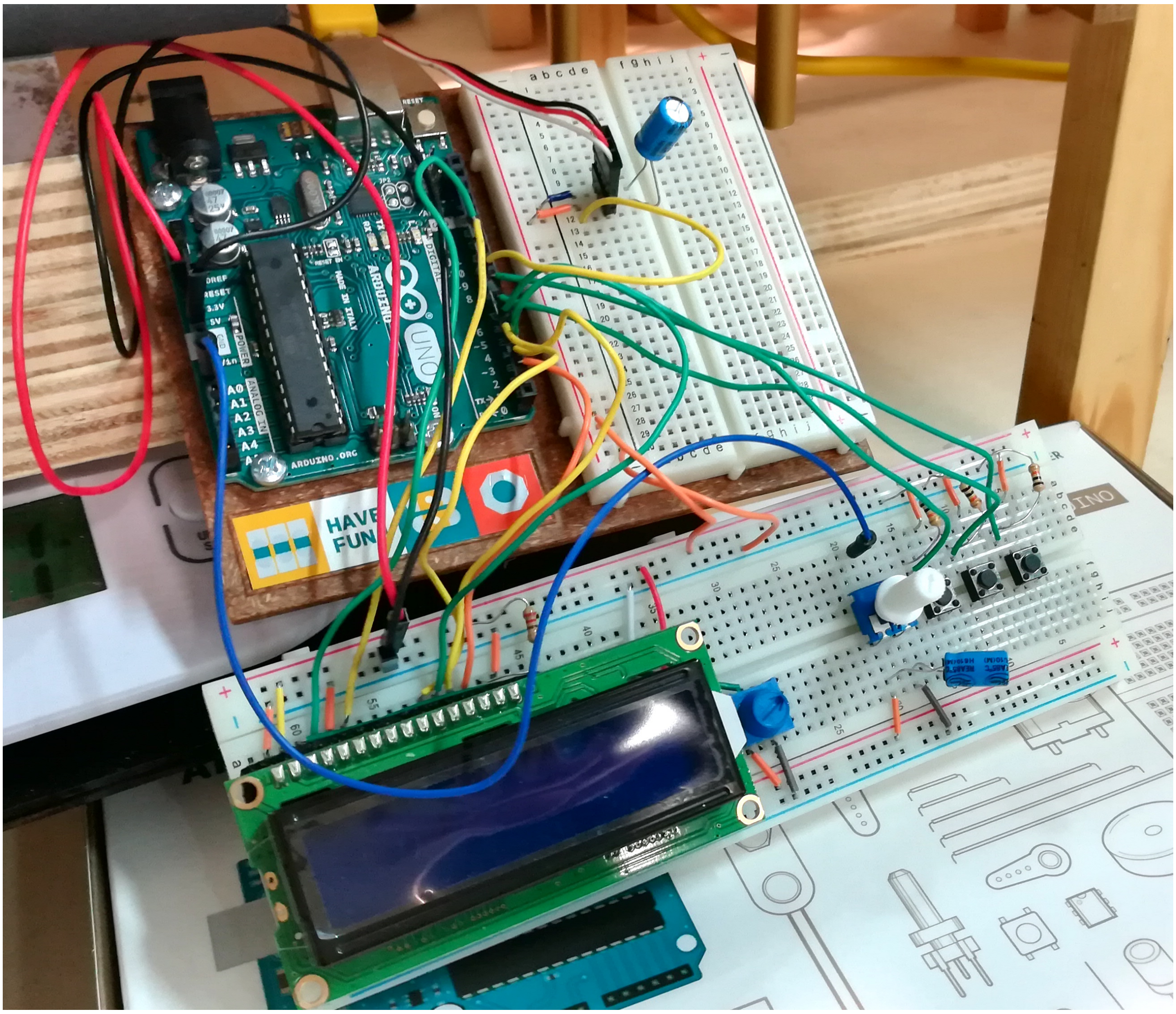

Fig. 3 – My arrangement of the Servo Control Panel.

Note: although incorrect, I have kept the original names for the code filename, variables, and display statements. Angle of attack, alpha, AOA are synonymous, defined as the angle between a reference line and the wind direction in the longitudinal plane. In my application, I have changed the deflection angle of a control surface, which also changes the angle of attack, by altering the local wind direction due to the pressure difference between the upper and lower wing surfaces. By AOA on the display I mean servo angular position, which is related to the control surface deflection by the linkage system. The variation of angle of attack and lift force due the control surface deflection is related to the control surface effectiveness.

# Bill of Materials: angleAttackControlPanel.fzz

/Users/danilo/Documents/Arduino/angleAttackControlPanel/angleAttackControlPanel.fzz

sabato, dicembre 9 2017, 11:53:40

## Assembly List

| Label | Part Type | Properties |
|---|---|---|
| Arduino UNO | Arduino Uno (Rev3) | tipo Arduino UNO (Rev3) |
| Autosweep ON/OFF Switch | Pushbutton | package [THT] |
| Autosweep Switch Pulldown | 10kΩ Resistor | tolerance ±5%; package THT; bands 4; resistenza 10kΩ; pin spacing 400 mil |
| LCD | LCD screen | tipo Character; pins 16 |
| LCD Contrast Knob | Trimmer Potentiometer | tipo Trimmer Potentiometer; package THT; maximum resistance 10kΩ; size Trimmer - 12mm; track Linear |
| R1 | 220Ω Resistor | tolerance ±5%; package THT; bands 4; resistenza 220Ω; pin spacing 400 mil |
| Servo | Basic Servo | |
| Servo Angle Knob | Trimmer Potentiometer | tipo Trimmer Potentiometer; package THT; maximum resistance 10kΩ; size Trimmer - 12mm; track Linear |
| Servo Capacitor1 | Electrolytic Capacitor | package 100 mil [THT, electrolytic]; capacitance 100µF; voltage 6.3V |
| Servo Knob Capacitor | Electrolytic Capacitor | package 100 mil [THT, electrolytic]; capacitance 100µF; voltage 6.3V |
| Sign Switch | Pushbutton | package [THT] |
| Sign Switch Pulldown | 10kΩ Resistor | tolerance ±5%; package THT; bands 4; resistenza 10kΩ; pin spacing 400 mil |
| Zero Switch | Pushbutton | package [THT] |
| Zero Switch Pulldown | 10kΩ Resistor | tolerance ±5%; package THT; bands 4; resistenza 10kΩ; pin spacing 400 mil |

## Shopping List

| Amount | Part Type | Properties |
|---|---|---|
| 1 | Arduino Uno (Rev3) | tipo Arduino UNO (Rev3) |
| 3 | Pushbutton | package [THT] |
| 3 | 10kΩ Resistor | tolerance ±5%; package THT; bands 4; resistenza 10kΩ; pin spacing 400 mil |
| 1 | LCD screen | tipo Character; pins 16 |
| 2 | Trimmer Potentiometer | tipo Trimmer Potentiometer; package THT; maximum resistance 10kΩ; size Trimmer - 12mm; track Linear |
| 1 | 220Ω Resistor | tolerance ±5%; package THT; bands 4; resistenza 220Ω; pin spacing 400 mil |

| Amount | Part Type | Properties |
|---|---|---|
| 1 | Basic Servo | |
| 2 | Electrolytic Capacitor | package 100 mil [THT, electrolytic]; capacitance 100µF; voltage 6.3V |

Exported with Fritzing 0.9.3- http://fritzing.org