Etude 3: EmitEmo

CART 360 AUTUMN 2017

DUE: October 12th by 6:30pm

PUSH: To **your** CART360 Github ONLINE Repository in the ETUDES directory

WHAT:
1. Arduino File labeled <lastname,firstname>_etudeThree.ino
2. Video Documentation of the completed task requirements labeled <lastname,firstname>_etudeThreeVideo.<mov,mp4 ... >

ETUDE 3 DESCRIPTION:

Etude 3, a natural evolution of Etude 2. In Etude 2, you were required to assemble the circuit on your breadboard, correctly connect all respective I/O to your Arduino and load the provided Arduino framework, in which you added (programmed) the missing code functionality to arrive at the desired goal. Sound and State Change Control of your Arduino!

For Etude 3, the primary focus is to continue with State Change Detection along with the incorporation of Timers and use of Averaging Algorithms **(Running Average)**. The intention of Etude 3 is to showcase why Timers are a preferred option to using Delays (hardcoded wait periods) and why Averaging Algorithms (Average Filter, Buffered Average and Weighted Average) are necessary when dealing with your I/O data. **Refer to yesterday's lecture slides and in class demos for additional information and clarification on the above topics.**

The circuit for Etude 3, Fritzing Diagram is included below, is essentially the same as for Etude 2, except for one minor addition – Etude 3 incorporates a Photocell as an additional Analog Input. The input of the Photocell will be used to modulate (alter) your sound.

***In Etude 3, you are NOT REQUIRED to store (record) the modulated sound for use in your playback modes – the feature is beyond the scope of Etude 3.**

You will use:
The completed circuit from Etude 2, along with the inclusion of the Photocell and the Etude 3 Arduino Framework.

The outcome of the etude will extend the playable quality of the keyboard by:

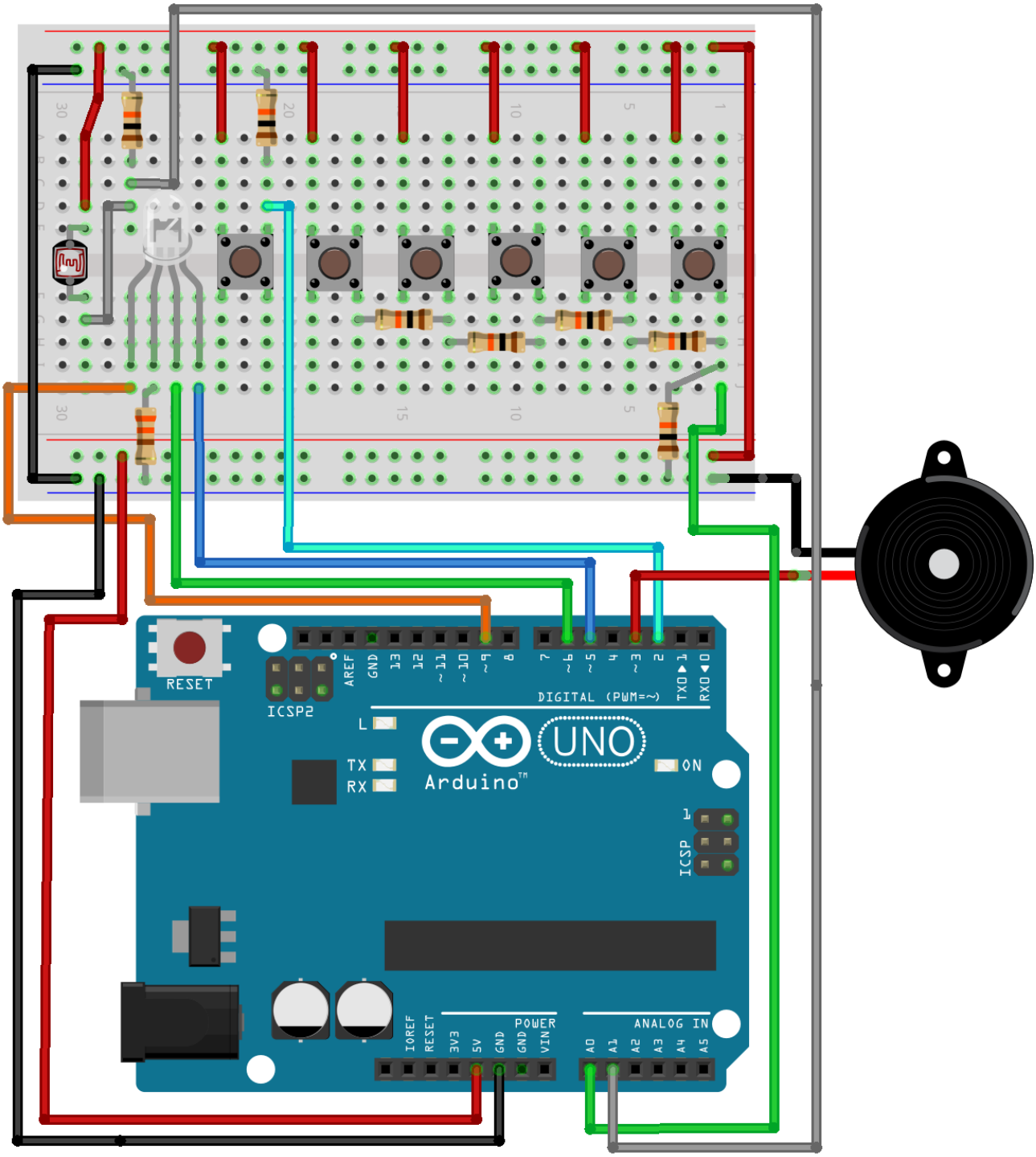
a) Modulating the sound of each note using the Photocell as modulator in both live mode and record mode **(* See Above)**
b) Capturing the length (time) of each note played in record mode
c) RGB LED in Etude 3 has Two Functions (Etude 2: Mode Selected and Etude 3:Note Intensity) – RGB LED intensity will change based on note intensity in Playback Mode
d) Incorporate Timers and calculate the Running Average for your I/O.

Carefully Read the Etude 3 Instructions in the provided Arduino Etude 3 Framework.

INSTRUCTIONS:

Step 1: Please build the circuit as **depicted in** the Fritzing Diagram – see below.
Step 2: Hook up your RGB Led to digital pins (acting as analog pins through PWM) 5,9,6
Step 3: Hook up the buttons connected to the resistor ladder to analog pin A0
Step 4: Hook up the Piezo Buzzer to pin 3 as an analog output
Step 5: Hook up the mode button to pin 2 as a digital input
Step 6: Hook up the Photocell to Analog Pin A1 as analog input
Step 7: Please make a **copy of the template code file supplied** and **follow** the instructions as specified in the provided Arduino Sketch.
Step 8: Please provide meaningful comments for any code that you write.
Step 9: Once completed, make a short video of your keyboard working in all 5 modes.
Step 10: Commit and Push new & modified **Arduino Sketch file and movie file** (labelled as specified above) in your ETUDES folder

NB: You cannot push a file > 100 MB, so be aware of your video's size (does not need to be HD, 4K)– rather keep it small and precise.


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino™
UNO
ON
ICSP
POWER
ANALOG IN
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5

fritzing

Resistor colour code