**Adding a potassium M conductance**

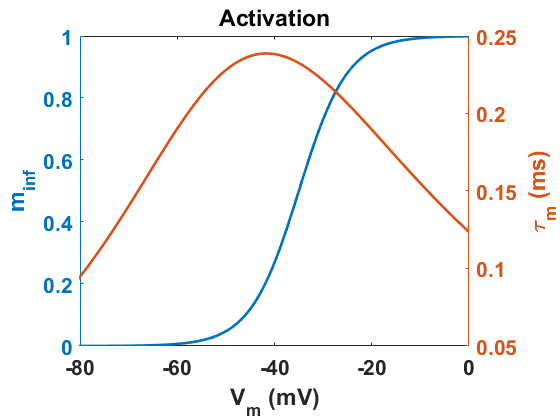
**I. M conductance**

We wish to add a potassium M conductance to the dynamic clamp system. The conductance we have chosen is one that appears in Fransen et al., J. Neurosci. (2002). The conductance is modeled in the Hodgkin-Huxley way by a single activation gate:

Here, represents the fraction of open M channels at an instant in time . Its steady-state value depends on membrane potential :

So does its time constant:

This is what they look like:



The current this conductances passes is given by:

where is the maximal M condutance (in nS) and is the potassium reversal potential (-80 mV).

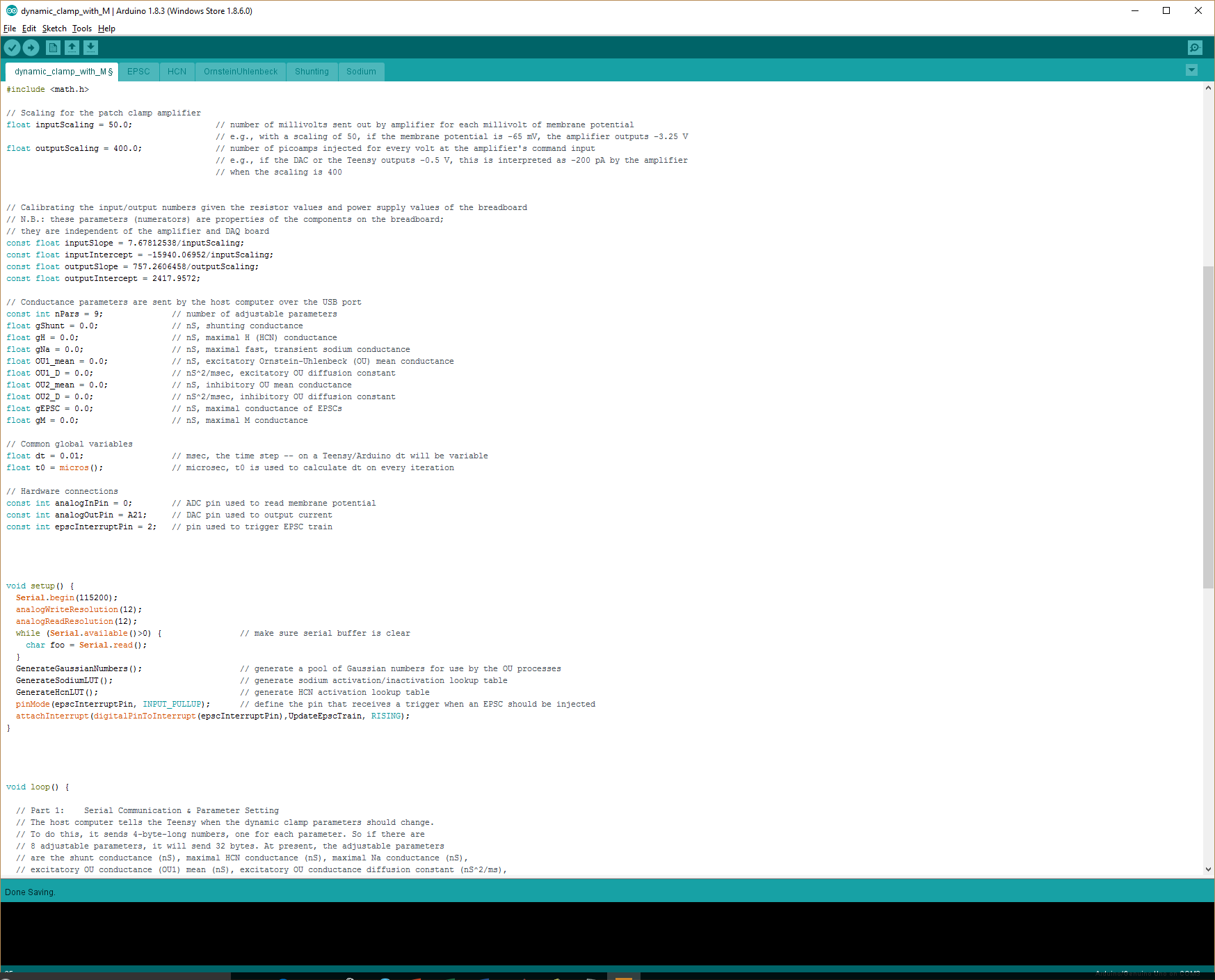
**II. Modifying the Arduino program**

To add this conductance to the Arduino program, we have to do two things: (1) add references to the M conductance to the main file *dynamic\_clamp.ino*, and (2) add a tabbed file defining the conductance.

Open the dynamic clamp program in Arduino.

*Main file*

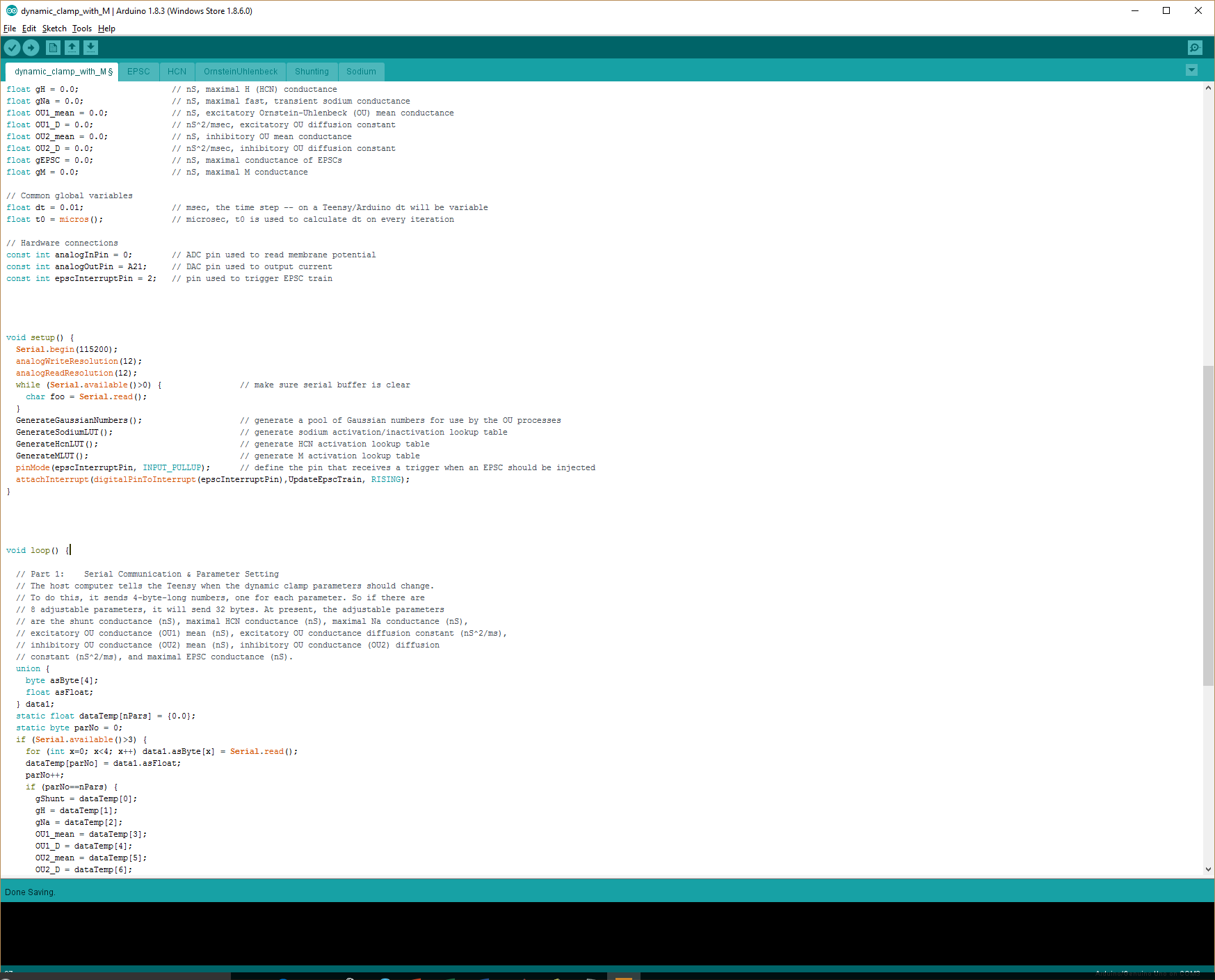
(1) Add the maximal M conductance to the list of global variables in the first section.



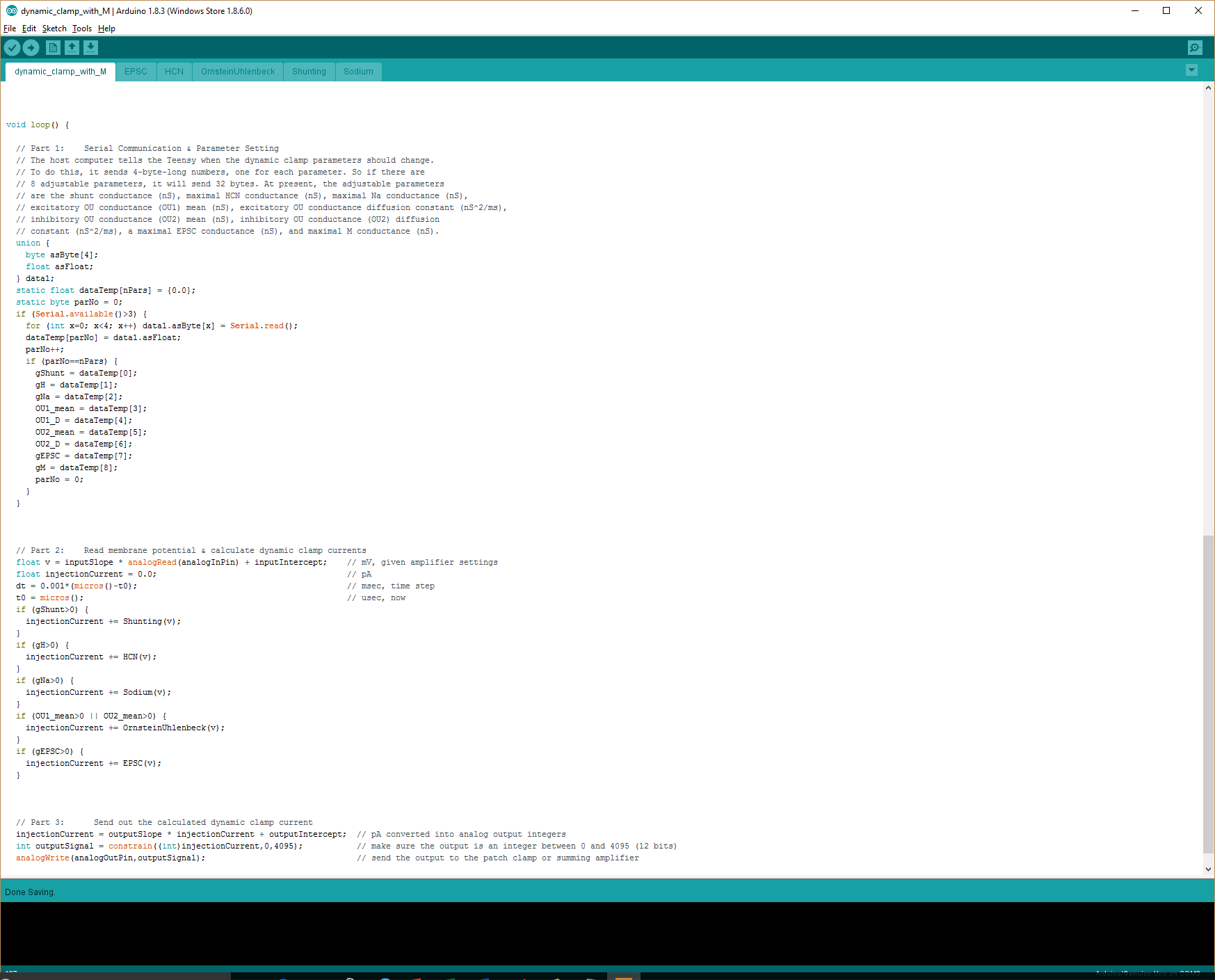
Note that we both changed the number of parameters (nPars) to 9 and added the last line declaring and setting its initial value to 0.0

(2) In the *setup()* function, add a call to a function *GenerateMLUT()* that will create lookup tables of values of and for a wide range of membrane potentials. Rather than doing the calculations at every time step, we will simply look up the appropriate values for the measured , thus allowing the program to execute more quickly.

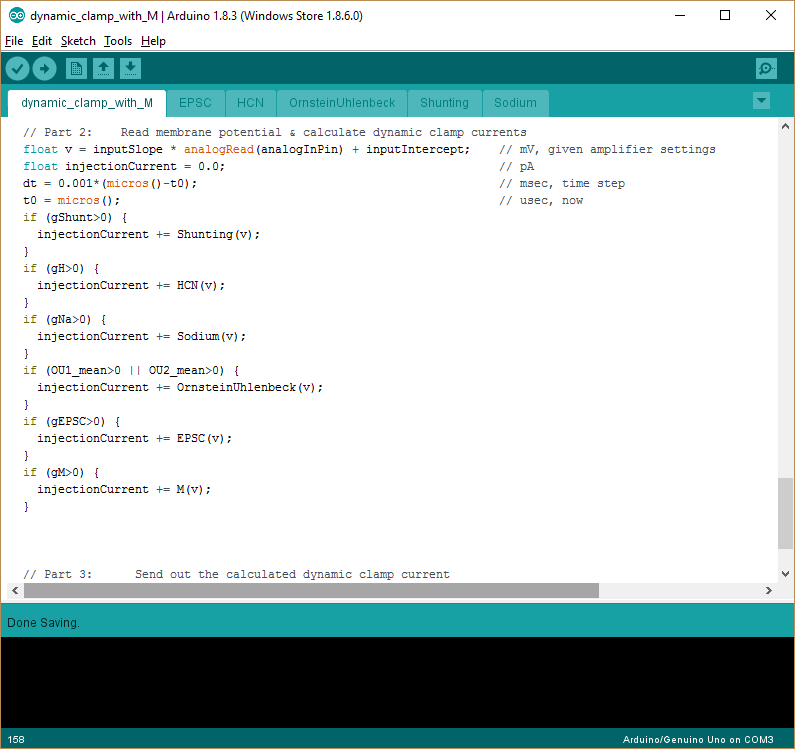
The function *GenerateMLUT()* will be defined in the tabbed file we will create shortly.



(3) Add the maximal M conductance to the list of parameters the Teensy receives over the USB port from the Processing sketch.

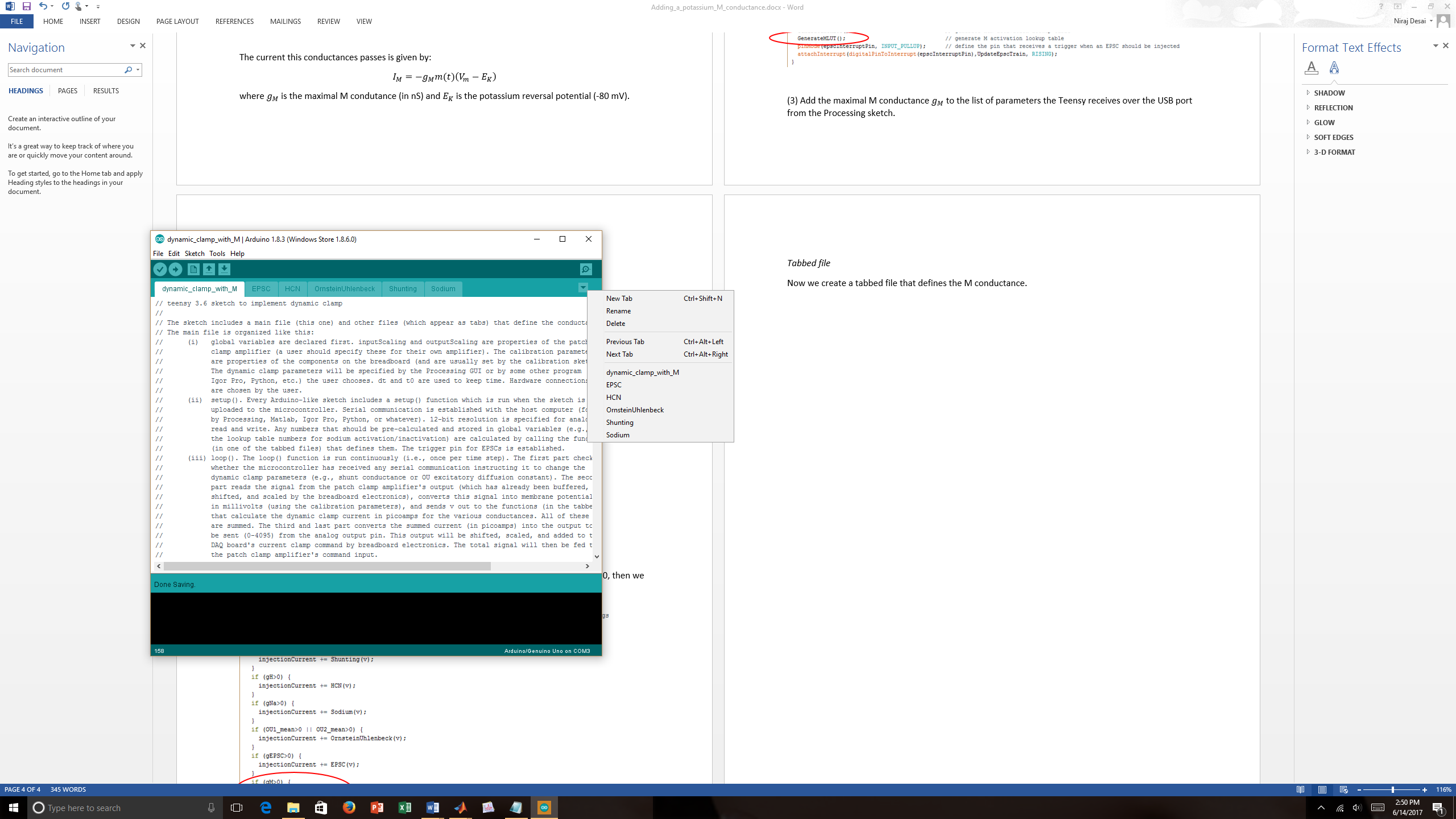


(4) If >0, have the simulated M current added to the other simulated currents. If it equals 0, then we don’t bother since the current will simply be zero.



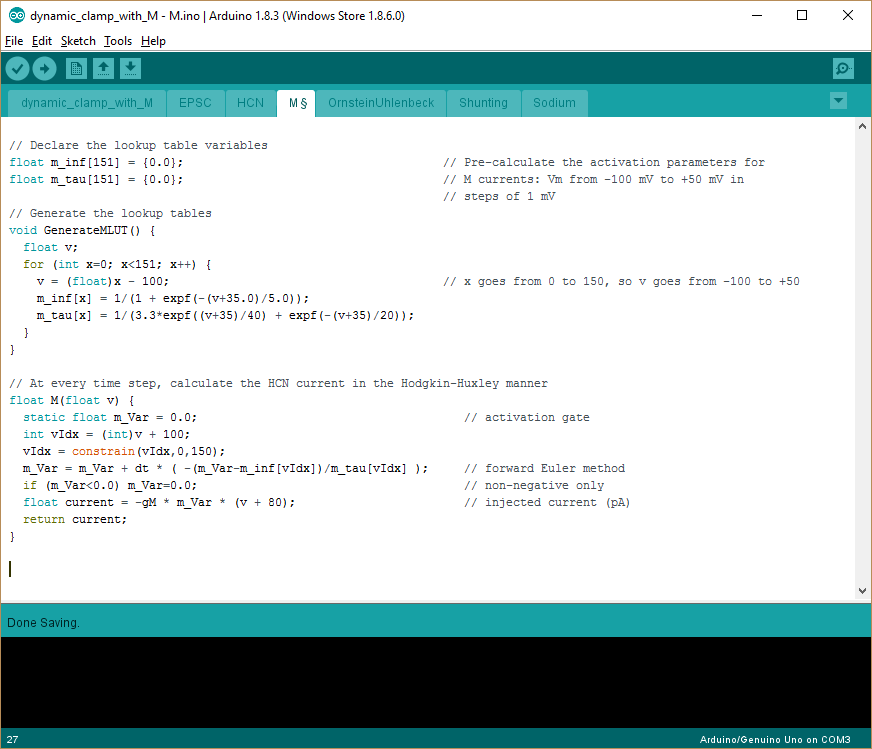
*Tabbed file*

Now we create a tabbed file that defines the M conductance.



(1) Create a new file. In the Arduino window, look for the downward-facing arrow (indicated by the red arrow). Click it and select “New Tab”. You will then be asked to name the new tabbed file – just call it “M”.

(2) In that M file, enter the three parts shown below: (i) array declaration in which the lookup table values of and will be stored; (ii) the function *GenerateMLUT()* that generates the lookup tables (for membrane potentials between -100 mV and +50 mV); and (iii) the function that numerically integrates and calculates the M current at every time step.



A few things to note. (1) We use the function *expf()* in preference to *exp()* because the former instructs the compiler to make use of Teensy’s floating point unit (FPU), thus speeding up the calculations. (2) We use the built-in function *constrain()* and an *if* statement to make sure vIdx and m\_Var stay within bounds. These lines aren’t strictly necessary because an out-of-bounds situation may never arise, but better safe than sorry. (3) We use the forward Euler method to integrate , which is called *m\_Var* in the code. The time step is short enough (compared to the kinetics of the M conductance) that this is acceptable.

Finally, save the modified program and upload it to the microcontroller by pressing the right-facing arrow at upper left.

**III. Modifying the Processing program**

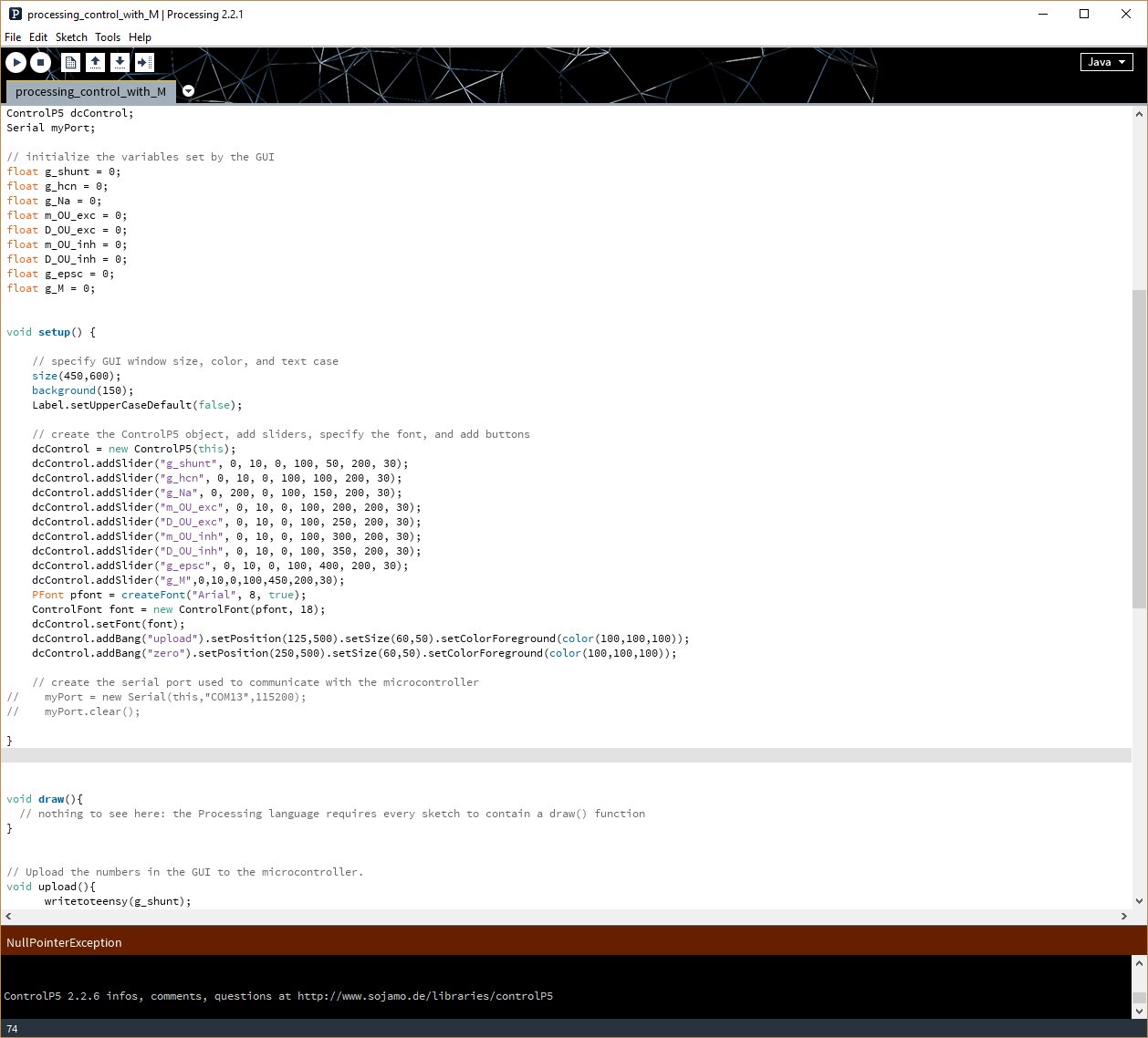
To modify the Processing program (*processing\_control.pde*) so that it will specify the maximal M conductance in addition to the other parameters it already specifies, we need only add four lines.

(1) Add to the list of variables declared when the GUI opens.

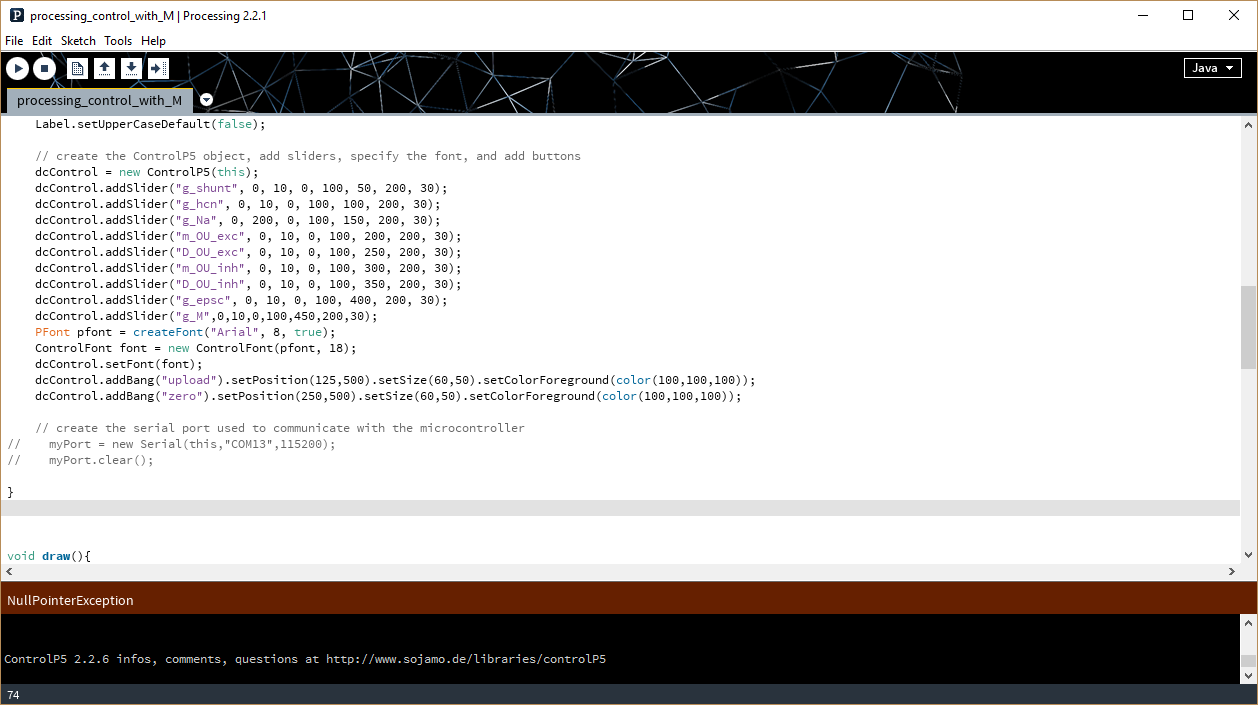
(2) Add a slider with which to specify the value of .

(3) Add a statement to write the value of to the Teensy microcontroller through the USB port.

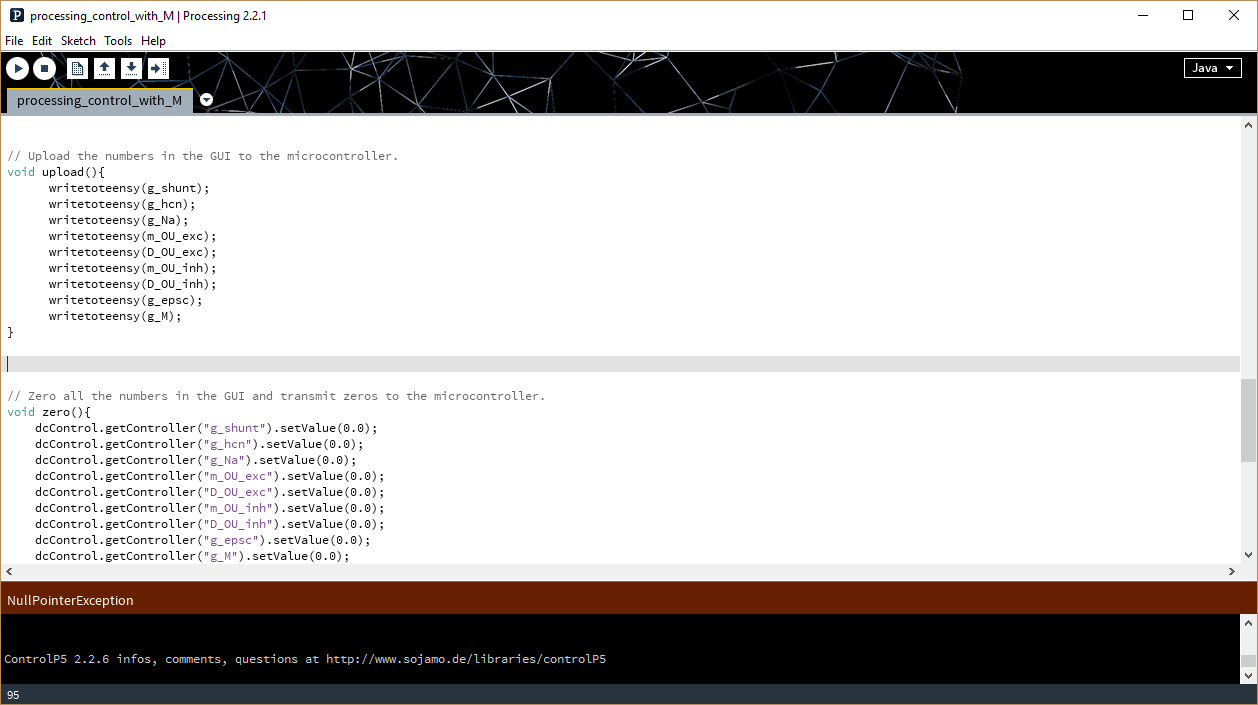
(4) Add to the list of variables zeroed out when the user presses the ***Zero*** button.



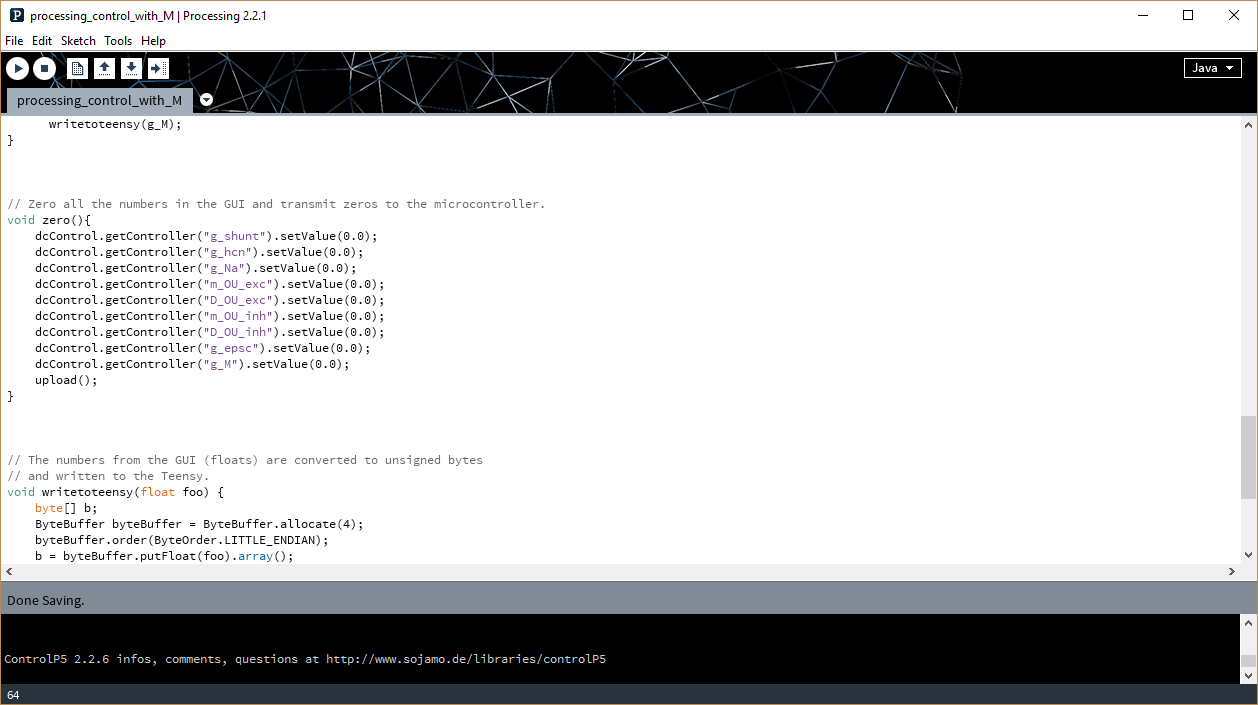
**1**



**2**



**3**



**4**

Save the changes and then run the Processing program. It should open up a GUI window that looks like this:

Notice that a slider for the M conductance has been added to the end. It can be used just as the others are.

The completed Arduino and Processing programs, with the changes specified in this document, can be found in the folders called *dynamic\_clamp\_with\_M* and *processing\_control\_with­\_M*.