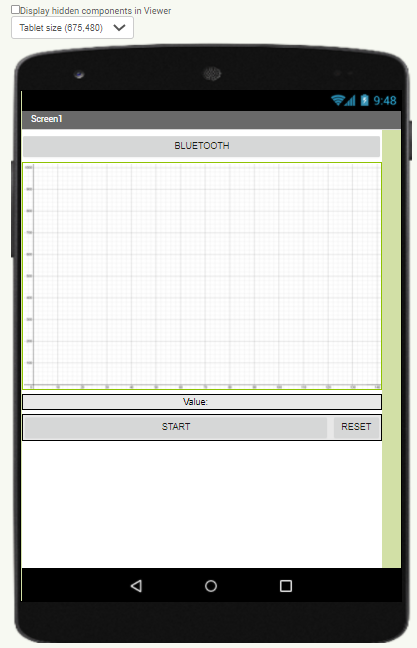
**Activity Part 4 Worksheet: Scrolling Graph**



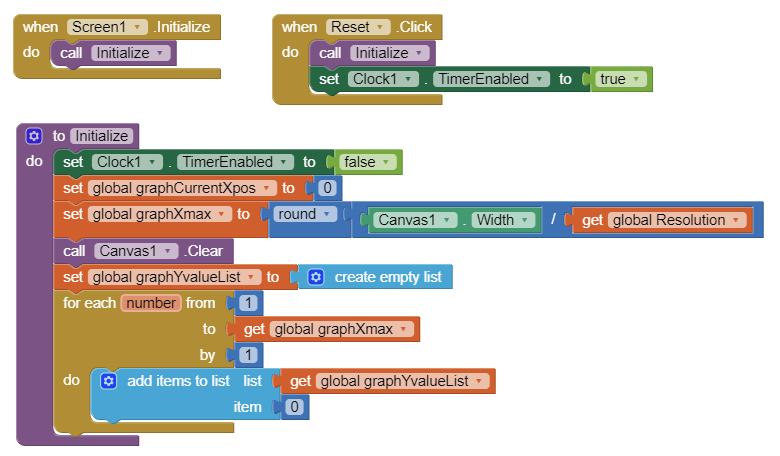
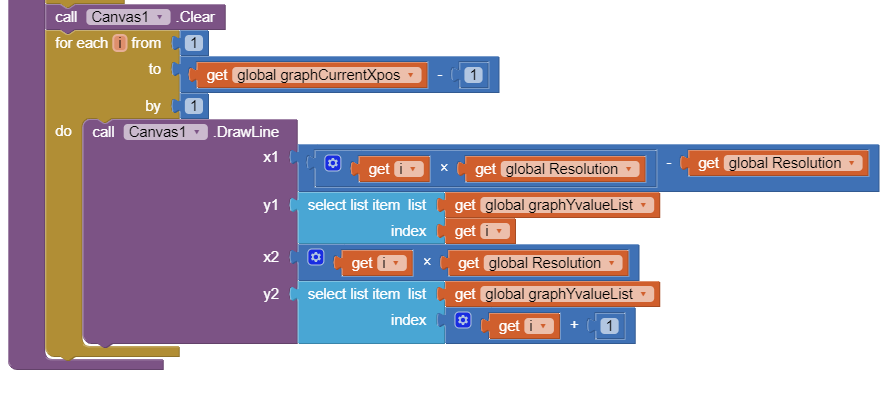
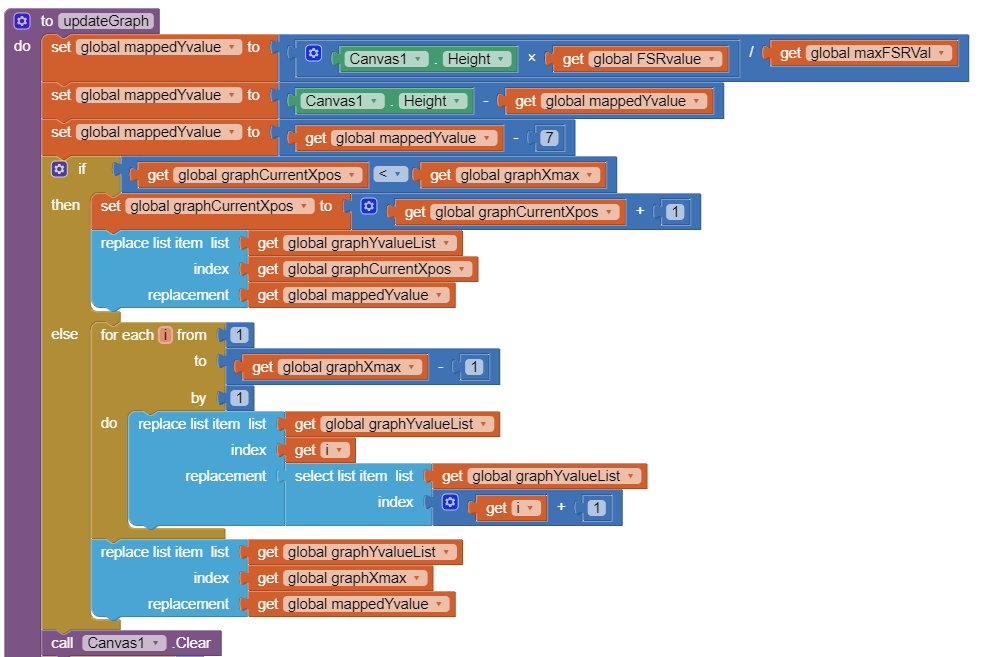
**Additional Equipment Needed:** None

**Procedure:**

In the previous activity you added the code and user interface to display the force sensor values. In this last activity you will add a scrolling graph. MIT App Inventor is not meant to display real-time data. However, with the Arduino handling any buffering, you will still be able to chart the data with a small delay.

1. Use Desmos and Snipping Tools to create a grid. Or find a free image online. Save it as a .PNG or.JPG format.
2. On the Design tab of MIT App Inventor 2, add a canvas that fill the parent width. You can try different settings for the height. The example shown uses 300 pixels, intended for a tablet. Add a horizontal arrangement with a START and RESET button. You should already have a START/PAUSE button from the previous activity and will only need to add the RESET button..
3. You will need to initialize seven different variables, one of which you should already have from the previous code.

|  |  |
| --- | --- |
| Variable Name and Initial Value | Description |
|  | Holds the digital value of the sensor from 0‒1023; or the voltage value from 0‒5000 mV. Initialize the value to 0 in either case. |
|  | The horizontal distance between sensor reading plot points. The recommended value is 5 pixels. |
|  | The raw sensor value proportioned to the vertical height of the canvas. Initialize the value to 0. |
|  | The maximum x-value that can be plotted which is derived from the canvas width and the resolution. |
|  | Keeps track of the most recent horizontal pixel value on the graph. |
|  | A list of all of the mapped sensor values (the y-values). |
|  | The maximum y-value that corresponds to the height of the canvas. If plotting, digital values, the maximum will be 1023. If plotting voltage values, the maximum will be 5000 (mv). |

1. When the screen initializes or when the RESET button is clicked, you need to control the clock (disabled upon initialize but enabled for RESET) as well as set the beginning x-value to 0, calculate the maximum x-value. You also want to clear the canvas, create an empty list of y-values and then fill them with 0s to start.
2. Create an UpdateGraph function which you will need to call from the Clock set of blocks. 

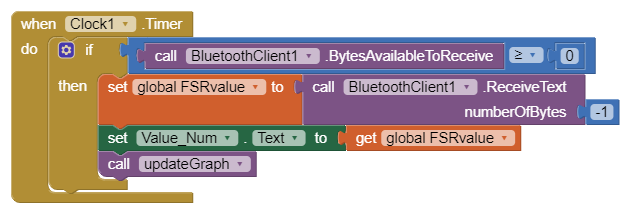
For each consecutive value, plot the sensor values from one to the next until the list is exhausted.

If the maximum value is reached shift all of the values to the left resulting in the graph scrolling left.

Clear the canvas with each new sensor value.

As long as the current x value is less than max, increase by one. Use this value as the index for the list of sensor values. Replace the 0 placeholder in the list with the sensor value.

Map the sensor value proportionally to the canvas height and subtract it from the height since the x-values increase to the right while the y-values increase going down. Then adjust a few pixels to account for the axis labels on the image.

1. Call the UpdateGraph function from the Clock blocks.  
   
2. Upload the new code to the Android tablet and test. Use your finger to simulate a labor curve as shown. Test the scrolling function as well as the START/PAUSE and RESET buttons.