*Metronome Program Details*

**About:**

This document is an update on the Project after a programming solution has been formulated.

*You can view Repository at:* <https://github.com/AbigailHerron/MetronomeProject>

*Our team Trello for this Project is:* <https://trello.com/b/inEctDv3/iot-project>

**Technical Difficulties Faced:**

* Understanding C++ and converting the C# logic flow was trickier than anticipated, however I am 70% confident that a possible solution has been found.
* Was unsure whether to use Clock() function or the Millis() function, however I decided on the Millis() as it is measured in milliseconds, the same as delay() whereas the Clock() is measured in ticks and would need the CLOCK\_PER\_SEC conversion to be incorporated to the solution, thus making it more complicated than necessary.
* Have yet to test the solution with the Arduino Yun, so I can’t be certain that it works.
* Was unable to add other team members as contributors to GitHub repository.

**Research Links:**

*Understanding the Clock()* - <http://www.cplusplus.com/reference/ctime/clock/>

*Understanding basic Arduino Functions* - <https://www.arduino.cc/reference/en/#functions>

**The Proposed Solution:**





**Possible Additions to Project:**

* Added use of a button to record beats whilst button is pressed, avoiding the need to reset the program every time you want to change to a different tempo
* Change the LED pin from Digital to Analog and have the first beat in the tempo a brighter burst than the following four based on a 4/4 time signature
* Added timings to represent different time signatures (e.g. 4/4, 2/4, 3/6, etc) and have the program ask if one of these set time signatures is desired (prompt should appear when the custom beat is within a few milliseconds of standard time signature)