

## Software Programming Assignment #2

Due: Mar 13, 2014 @ 6PM

---

*Note 1: This homework has to be done in teams of three (according to the registrar's records the # of students in the class is 27, a multiple of 3). Please use Piazza to find team members. As you form confirmed teams, please let me know by filling out the web form at <http://goo.gl/g18dZn>. I'll approve smaller teams of 2 only if the # of students in the class is actually not a multiple of 3, in which case I'd approve either one or two teams of 2 as appropriate. Teams of 2 won't have to do the event generation mode (and anything related to that mode).*

*Note 2: Your submission is in the form of a URL to a Dropbox folder whose contents should be organized as per the instructions at the end, and optionally URL to a video demonstration. Please submit via the web form at <http://goo.gl/LGpLvX>.*

---

The objective of this homework is two-fold: (i) To employ time synchronization concepts, (ii) To use Android in the context of embedded application. Besides mbed, of which your team would already have three, you would also need a BeagleBone Black (BBB) configured to run Android. If you don't have a BeagleBone Black, you can borrow one from me. You'd also need an 8GB or more SD Card and an ethernet cable.

1. Configure BeagleBone Black to run Android by following the instructions at <https://docs.google.com/document/d/1gXYd3piRO1zp4uT2v8VB4E6mTfgWeVUhMmKdziOxino> The document includes instructions to set things up so that you may use a VNC viewer application on your host (there are many VNC viewers available for various OS) to look at the Android display. Alternatively, you'd need a monitor with a HDMI input.
2. Your goal is to design a distributed event monitor and event generator system using two of the mbeds and the BBB. To do this, you'd need to develop a time synchronization strategy
3. Connect two of the mbeds, the BBB, and a host computer as follows:
  - o mbed1 connects to mbed2 via a serial link (UART)
  - o BBB connects to mbed1 via a serial link (UART)
  - o BBB connects to the host via a LAN

Additionally, for purposes of code download or debugging, you can connect the two mbeds to a host via USB. The figure at the end shows how the connectivity looks like.

4. **[20%] Event monitoring mode:** In this mode you need to monitor one external signal at each of the two mbeds in your system and assign events (0 to 1 transitions on a wire connected to a pin of your choice) timestamps according to a common time coordinate. The origin for your time coordinate should (roughly) correspond to the wall clock time. The quality of your design is defined by its performance on two metrics:
  - o difference in the timestamps reported by the two nodes for a common event
  - o difference in the timestamps reports for two events at the same mbed vs. the actual time difference
5. **[20%] Event generation mode:** In this mode you need to generate synchronized events on a GPIO pin of mbed1 and mbed2. Specifically, given a period T in milliseconds (ranging from 10 ms to 10000 ms), the three nodes should generate time synchronized square waves (i.e. 50% duty cycle) whose edges are aligned in time. The quality of your design is defined by its performance on two metrics:
  - o difference in occurrence time of same edge at the two mbeds as observed by an external observer

- difference between T and the externally observed occurrence time of two successive rising edges (or two successive falling edges) at the same mbed
6. **[20%] Frontend App:** You should write a simple Android app for BBB for controlling the two operation modes of your system.
- In the event monitoring mode the app should display the timestamps of the latest event on each of two mbeds, allow event timestamps to be logged into a file, and allow events to be sent to Xively. The app should allow separate on/off control for file logging and Xively logging for events, and allow configuring log file name, Xively feed id and Xively key (these parameters are common for all three nodes).
    - The events in the file should be logged such that each line is of the form *node\_id, timestamp* where *node\_id* is 1 for mbed1, and 2 for mbed2, and *timestamp* is in ISO8601 format (similar to Xively) with 8 digits in the fractional part of second (i.e. to 1/100th microsecond resolution).
    - When logging to Xively, in order to make visualization easier, give values of 0 and 1 to alternate events to the same node. Also use *mbed1* and *mbed2* as channel names.
  - In the event generation mode the application should let you control T (from 10 to 10000 ms) and also start/stop the event generation.
7. **[20%] Testing Mbed:** For the third mbed in your team (*mbed3*), write two simple programs for testing purposes. You'd connect the testing mbed to
- **Program 1** generates on a GPIO pin of your choice a square waveforms of 50% duty cycle with following controls: a configurable period T (from 0.1 ms to 10 s), and a control to start/stop the waveform generation. Make sure that your square waveform has a period which is as precisely T as possible - you may want to use the hardware counter/time to do this. You don't need to provide a fancy front end for this: the configuration could be done via a file which the program reads whenever the file is updated. You'd connect the output pin to the event inputs of *mbed1*, and *mbed2*.
  - **Program 2** monitors two GPIO pins on your choice, captures a high resolution timestamp according to a local clock whenever it sees a 0 to 1 event on a pin, and sends this information (i.e. pin and timestamp) to the host. You should come up with as high resolution a timestamp as you can as it would directly impact the quality of the performance you report, and so should use the event capture capabilities of mbed's counter.
8. **[20%] Submission instructions:** Create a Dropox folder with the following contents
- **Report:** At the top level of the folder should be a file *report.pdf* with your report. Include in your report a succinct description of your system structure and your time synchronization strategy, as well as a summary of the results from your performance tests.
  - **Code (source code and binary):** Submit source code and ready to run binaries for BBB, the three mbeds, and the host, along with a description of how to run it. Create five folders labelled *BBB*, *mbed1*, *mbed2*, *mbed3*, and *host*.
  - **Installation instructions for hardware and software:** Provide an installation instruction file *install.pdf* with details on hardware connectivity (specifically which pins you used and how to connect them), as well as software setup (how to run your software).
  - **Results of performance tests:** Include summary analysis of the results of following tests in your report, and provide raw data as spreadsheets inside a sub-folder names *results*.



