CSM213A/EE202A - Embedded Systems [Fall '14]

Assignment Part B: Programming [100 points] (Ver. 1.0)

Due: Wednesday December 17, 5PM

Instructions:

This assignment is to be done in self-formed groups of 3-4 students using Piazza as the vehicle to form teams. Submit your work as a .zip archive containing a folder with a PDF describing your work and any supporting files (e.g. code, spreadsheet etc.). Only one person in a team should submit. Additionally, one or more team members would need to demo the running system during the finals week.

Assignment Description:

The objective of this assignment is to create a prototypical IoT system following the Device-Gateway-Cloud model and with the functionality described below. You are allowed to use any software technologies of your choice as long as you meet the requirements laid out below.

System Configuration

Your system should have two mbeds acting as devices in the embedded layer. They should communicate both via a wired serial (UART, I2C, USB) or wireless (anything that you have access to) link to a gateway which should be a Linux-class embedded board such as Raspberry Pi or Beagle Bone Black (I have a few RPi and BBB to check out) or perhaps even a laptop (but peformance will likely be poorer). No other communication is allowed between the mbeds or between the mbed and the gateway. Finally your system should have a backend cloud component that would provide a user interface. The backend cloud component must run on a networked machine different from the gateway (you may chose to run it as a VM on one of the many cloud services).

Functionality Requirement: Time-synchronization

Establish a common synchronized time between the two mbeds via appropriate message exchange with the gateway, as well as learn the offset from the real-world time maintained at the gateway (which you should set to use NTP). As a consequence the two mbeds should be able to do synchronized actions and also timestamp events in terms of real-world time.

To demonstrate successful time synchronization:

1. Display the minutes and seconds of mbed's time in MMSS format on the LCD display of each mbed.
2. Generate on each mbed on a pin number of of your choice a signal consisting of a 100 ms wide pulse every 1 second with the rising edge of the pulse occurring at the 1 second boundary according to mbed's time.

The quality of your time synchronization will be assessed by three metrics:

- The time interval between the rising edges of the 1 pps signal at the two mbeds
- Synchronization between MMSS values at the two mbeds and a NTP-synced laptop.

Functionality Requirement: Sensing

Each of the mbed should:

1. Monitor asynchronous 0-to-1 digital events (i.e. change from 0 to 1) on an input pin of you choice (you must clearly specify which pin you use). Consider using timer hardware to get accurate timestamps of events.
2. Capture measurements from the light sensor at 1Hz rate continuously.
3. Capture measurements from Channel # 0 of ADC. The user interface should allow for the sampling to be dynamically turned on/off and the rate to be dynamically adjusted from {100 Hz, 10 Hz, 1 Hz}.

The collected waveforms and events should be sent to the backend where they should be visualizable in real-time as *value vs. time* plot, as well as stored for later access. I strongly recommend using one of the many services available for this (some were described in the student presentations). Remember, serial link is not a high rate link and so you should use it efficiently such by sending values to the gateway in binary format instead of ASCII.

To test the accuracy of your timestamps, give the two mbeds asynchronous digital events at a precise separation, and compare the timestamps reported for the events at the backend.

Functionality Requirement: Actuation

Each of the mbed sensor should be capable of being tasked for action as follows:

1. Set an output pin of you choice to 1/0 (you must clearly specify which pin you use). For testing purposes, you may also want to turn LED1 on/off in parallel.
2. Turn on or off the generation on the Analog Out pin of a smooth sine wave with specified frequency {1000 Hz, 100 Hz, 10 Hz, 1 Hz} and maximum amplitude. You may use this waveform to test #3 in sensing above.

Note

All the preceding functionalities must be active simultaneously.