MCTX3420 Team 4: Progress Report #8

Sam Moore, Rowan Heinrich, Callum Schofield, James Rosher, Justin Kruger, Jeremy Tan

Work Done:

1. Reading/Writing using real pins on BeagleBone
    a. Discovered that the linux kernel on our original image didn't support PWM; recompiled kernel
    b. Documentation on BeagleBone is contradictory, out of date and often incorrect. Got GPIO, PWM and ADC pin control, but took much longer than initially expected.
2. Estimated some sampling rates
    a. Since our software runs under an operating system, process/thread context switching means the sample rate is not constant.
    b. Did some analysis using software timestamps (resolution 1us).
        i. Without ADC_Read - Average sample rate is 18KHz
        ii. With ADC_Read - Average sample rate is 100Hz with a lot of variation - Figure 1.
3. Control of real pins using server API (set GPIO and PWM, read from ADC)
    a. EG: http://server/api/actuators?id=1&set=0 - Turn GPIO1_16 off (set=1 to turn on)
4. Collaboration with Electronics Team
    a. Setup our software on the Electronics' BeagleBone. However we need to help replace their kernel before PWM can be used. (See 1a).
5. Tested image related code (OpenCV) on BeagleBone
    a. May need to reconsider using FastCGI for image streaming, as it slows down the rest of the API
    b. Discovered why image stream was freezing software; FastCGI thread was running the image capture code. Possibly due to a dodgy webcam this was hanging, which made the software unresponsive. Moving image capture code to a separate thread or process will stop the software being unresponsive.
6. GUI Design
    a. reformatted gui and included :
        i. http basic login username and password, password protected (access denied when incorrect login is used)
        ii. camera stream to web page - division where stream.html is constantly reloading the page (refreshing) showing the image from the camera
        iii. plot of data - strain against time using javascript flot library
        iv. basic error and warning messages

Work TODO:

1. GUI Design
    a. Integrate gui designs.
    b. Begin adding controls which link to the server to control beaglebone.
    c. Continue implementing sanity checks for the gui.
    d. Include an actuator vs stress graph.
2. Collaboration with Electronics team
    a. Need to replace linux kernel on Electronics' BeagleBone so that they can test PWM
    b. Need to give Electronics team a detailed description of how to start the server and use the API
        i. It is currently only possible to control hardware directly by typing the correct URLs into a web browser. We should provide the syntax and API description.
        ii. Eventually we can provide the GUI, but currently many features are unimplemented
    c. Begin adding more GPIO/PWM control and ADC reading of the sensors as to the software , as specified by Electronics
3. Collaboration with Sensors Team
    a. Sensors has asked if the Logitech C165 is acceptable; we need to respond
    b. Before we implement any dilatometer image processing, we should get some realistic test images
        i. We also need confirmation we are definitely using the dilatometer; last week we wrote an interferometer test algorithm as requested only to find out it wasn't needed.

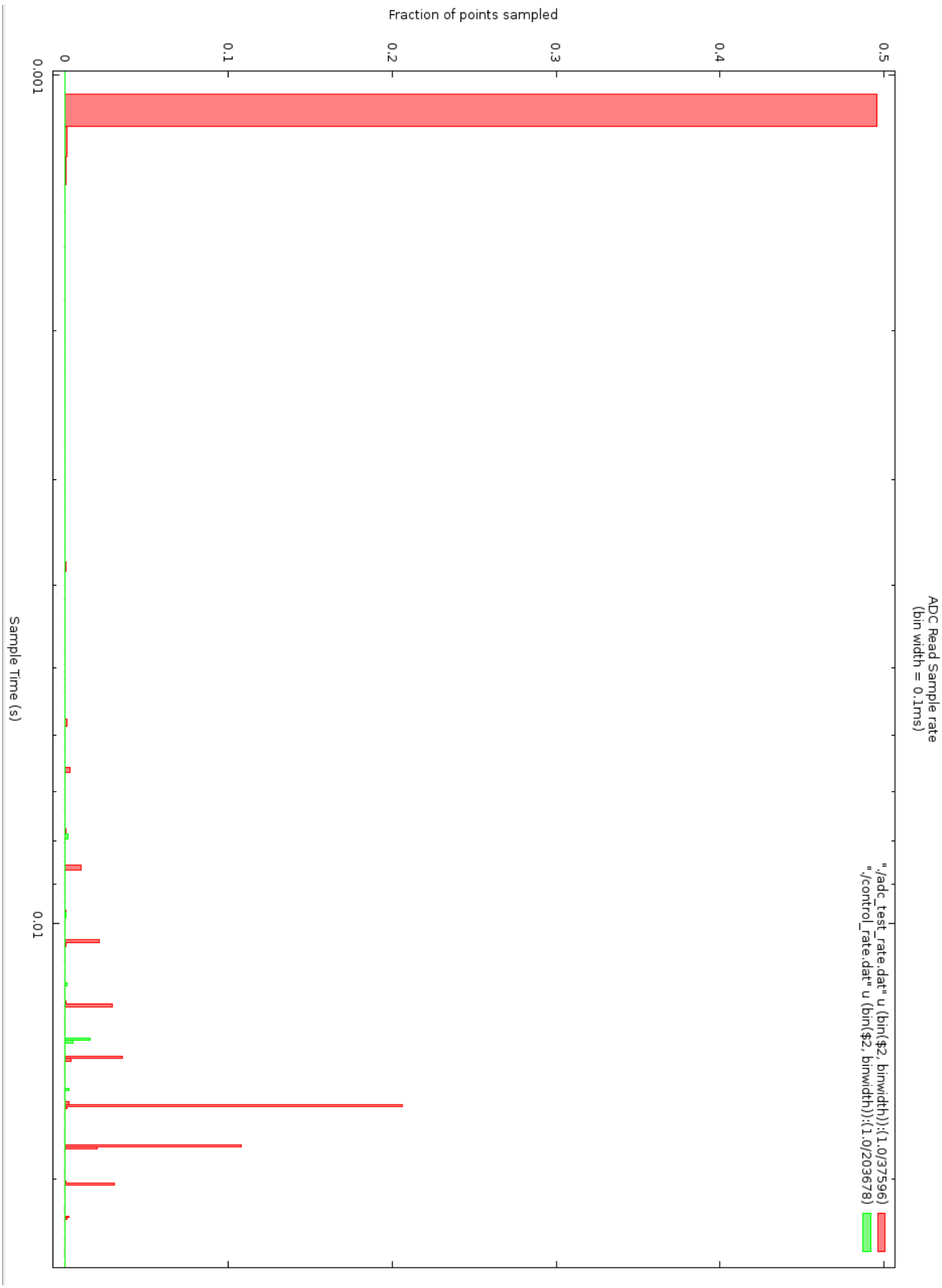

Figure 1: Histogram of sample times using ADC_Read. Note the logarithmic x axis.