# Brief status report

## Status update: as well as a few paragraphs about the related work. If you are working alone, I’d like to see at least 5.

## A list of references with the work cited.

## Can be email.

An short example of roughly what I expect:

Our research involves making an open-source low-powered video game that will run on an embedded Raspberry Pi board. Weaver[1] wrote a cross-platform assembly language game for the ARM platform but unlike us he did not characterize the power consumption while running. Mallow and Snap[2] look at optimizing a Ray-Tracing program on the Tegra2 ARM system. This is similar to our project, only they looked solely at ray-tracing applications and not video games.

[1] V. Weaver. "Tom Bombem: An ARM Implementation of the Classic DOS game." Proc. of the 4th Conference on Useless Video Games, p 10-18, May 1996.

[2] M. Mallow and G. Snap. "Optimizing Energy Consumption on the Tegra2." Journal of Embedded Programming, p11-19, Vol 1 Issue 15, June 2010.

My research is focus on the power/energy implications on GPU.

Raspberry pi B+ GPU:

Broadcom BCM2835, a System-on-Chip that contains an ARM1176JZFS with floating point, running at 700MHz, and a Videocore 4 GPU. The GPU provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode and is capable of 1Gpixel/s, 1.5Gtexel/s or 24 GFLOPs of general purpose compute.

Reference:

Power and performance analysis of GPU-Accelerated Systems

Abstract:

Our analysis on a real system discloses that system energy can be reduced by 28% retaining a decrease in performance within 1% by controlling the voltage and frequency levels of GPUs. We show that energy savings can be achieved when GPU core and memory clock frequencies are appropriately scaled considering the workload characteristics. Another interesting finding is that voltage and frequency scaling of CPUs is trivial for total system energy reduction, and even should not be applied in state-of-the-art GPU-accelerated systems.