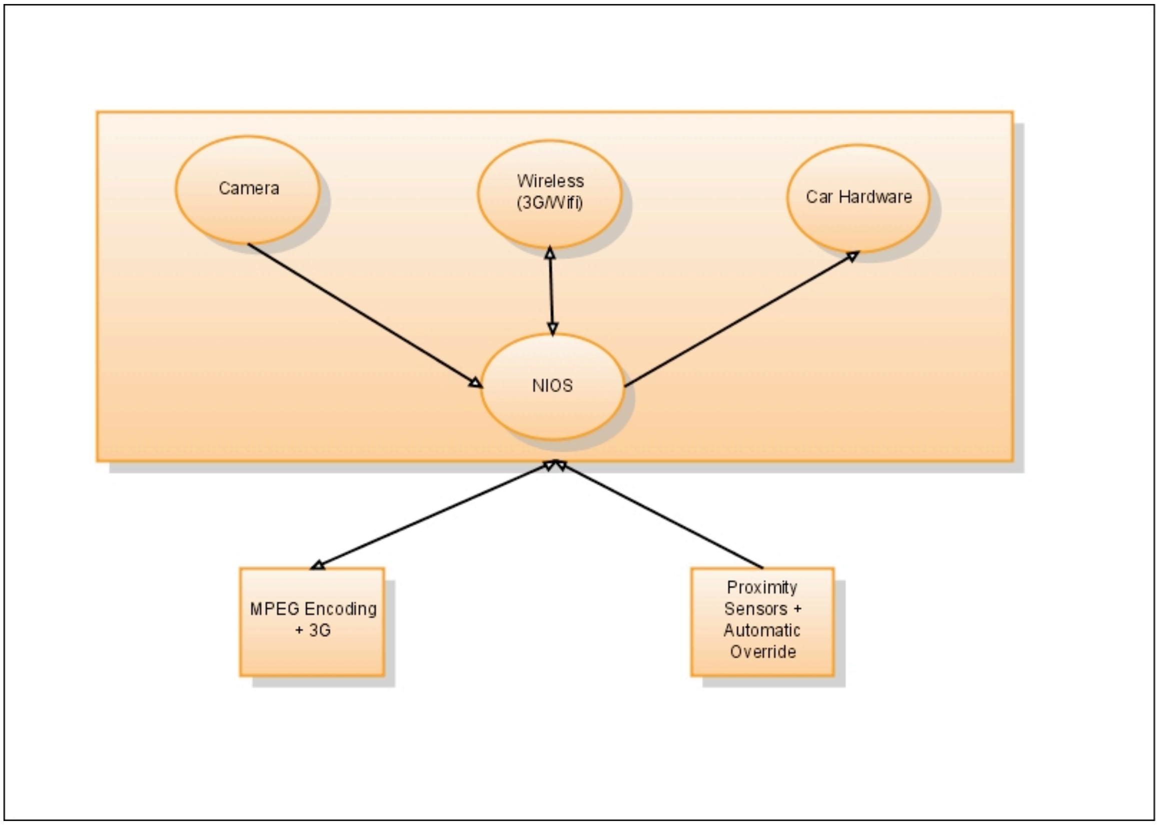
Project Proposal

Thomas Chau, Ben Sack, Peter Tsonev

CSEE 4840 – Prof. Stephen Edwards

March 2, 2009

The diagram below summarizes our project plan. The vanilla stage is represented in the big rectangle and the fancy stage would be one of the smaller rectangles.



We plan to implement control functionality on an RC car using an FPGA board mounted on the car. The diagram contains everything that will be on the car end. The human end is not included above, but it will probably consist of a PC sending commands to the board on the car and displaying video from the car. We can use another FPGA board instead of the PC, but the gist of the project is the functionality of the diagram on the car.

The big rectangle is the base of the project. The NIOS processor will receive input from the camera and send it to the human end via a WiFi connection. Control signals from the human end will arrive through the same wireless interface to be processed by NIOS and fed to the car hardware. This concludes the kernel of the project.

Since this might not be fancy enough, we will also implement the functionality of **one** of the two smaller rectangles (see figure). The MPEG Encoding would consist of both hardware and software implementation and will process the camera input so that the compressed data will consume much less bandwidth on the wireless channel. This might enable us to use a 3G wireless interface instead of WiFi and have the car controlled via a cell phone network. Manhattan has very good 3G coverage, and therefore, this branch of the project is known under the code name *2 Fast, 2 Furious: the New York Drift.*

If we choose to go for the other branch (the rightmost small rectangle), there will be proximity sensors mounted on the car that will send data about the environment to NIOS. Software, then, can use those signals to figure out if the car is too close to a wall or if it is approaching an obstacle too fast and automatically override human control to correct the course. The RC car we have is going reasonably fast and it is tough to make a sharp turn in a dorm hallway without hitting the wall. Thus, the proximity sensors can provide assisted high-speed driving.

First things first:

1) Reverse-engineer the car hardware with the scope
2) Research proximity sensors, MPEG encoding, and wireless cards.
3) Weigh costs, difficulty of implementation, and time to choose one of the two branches.
4) Mount the FPGA on the car and power it up without frying it.

NOTE: Our design requires us to purchase some of the components (wireless card, sensors, camera) and we need to know if we have a small budget or some other means to get some of this equipment.