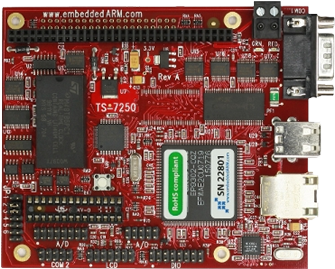
**Real Time Embedded Computing**

**Final Project:** Proposal



Done by:

Teekayu Klongtruajrok (**tkq66**)

**INTRODUCTION**

Automated machines could achieve many tasks that humans could not. However, as time can only brought about increase in entropy, the probability that the machine will break down can only, likewise, increase. There are two possible causes of why the machine broke down: the material deteriorates which makes the process implicit, or an external force acts upon the machine involuntarily, which makes the process explicit. The first case could be accounted for via regular maintenance whilst the second case could be divided into physical interactions via electrodynamics or Newtonian mechanics. Whichever that interaction may be, the machine has no way to respond to it reactively. Take for example a laptop computer that had fell from a table top, due to the lack of self-awareness, it does not acknowledge the imminent threat to its existence. If it does have the ability to detect threat, however, the reasonable course of action it could take would be to either back up all unique data blocks to the server with lowest latency or safely shut down the HDD to assure that no extraneous damages came to the, already vulnerable, hardware. That is, of course, if the design philosophy of the machine does not change. With “self-awareness” entering the design subconscious, there would be extensions and modifications made to make full use of this mechanism, allowing for a more proactive design that contributes to the machine’s survival instinct. This is the primary motivation behind this project: to create a mechanism that would allow machines to be self-aware of its physical surroundings.

**OBJECTIVE**

The purpose of this project is to create a mechanism that would allow machines to be self-aware of its physical surroundings in order to be able to evaluate potential threat to its physical conditions, thus allowing for a prompt reaction. The goal is to create real time embedded system that constantly monitors the physical surroundings via a sensor bank, then evaluate the data in the background and if the incoming data set could be classified as dangerous, then override all current operations before outputting a signal through one of the I/O pins that is designated as the reaction pin.

There are two levels of desired practicality. The first, which should be able to be achieved in the short term, is the system’s ability to read in from the sensors and a module that is able to respond accordingly. The second level is an extension, making the most of the resources in the lab: the response on one of board would contribute to the hivemind knowledge across all boards, that is to say is one board detects a thread, communication would be made among all the boards to be wary of the threat.

**RELATED SYSTEMS**

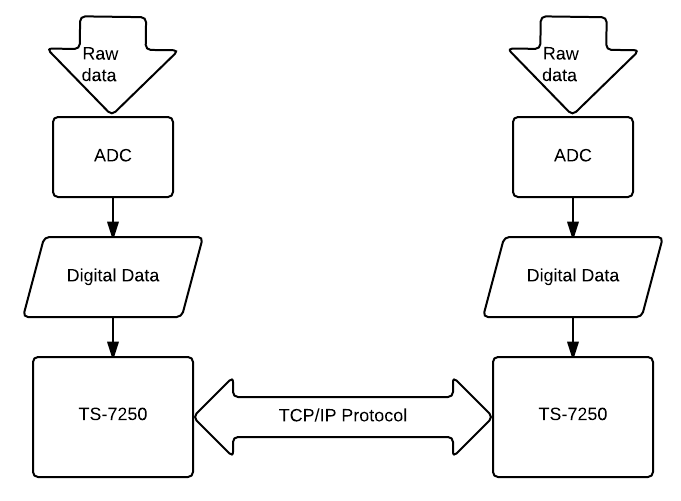
So far, the self-preservation mechanisms are tailored towards specific machines, for example creating a shockproof hard disks or black boxes in airplanes. All of those examples are reactive responses in a sense that they react after the time of impact. This is not the goal of this project. This project seeks to allow the machine to respond at delta time before impact.

**CONSTRAINTS AND SPECIFICATIONS**

The system would be applied onto at least two TS-7250 boards, each of which operates with a single core. It is assumed that there is a constant communication path between them, in this case, communication via WiFi. The success of the project is measured by the time displacement between the initial threat detection to the time reaction signal is achieved. If the time displacement is shorter than the duration of the threat, the project could be deemed a success. It is also assumed, for the sake of this prototype, that there are no resource intensive processes running at the time of testing. This would allow for the best-case results of the experiment, and further generalization in regards to the quality of the system could be done in future researches. For simplicity, only one sensor would be use for the sensor bank. Generalization for n-sensors (given that n is a positive integer), is trivial after a proof-of-concept for the system had already been done.

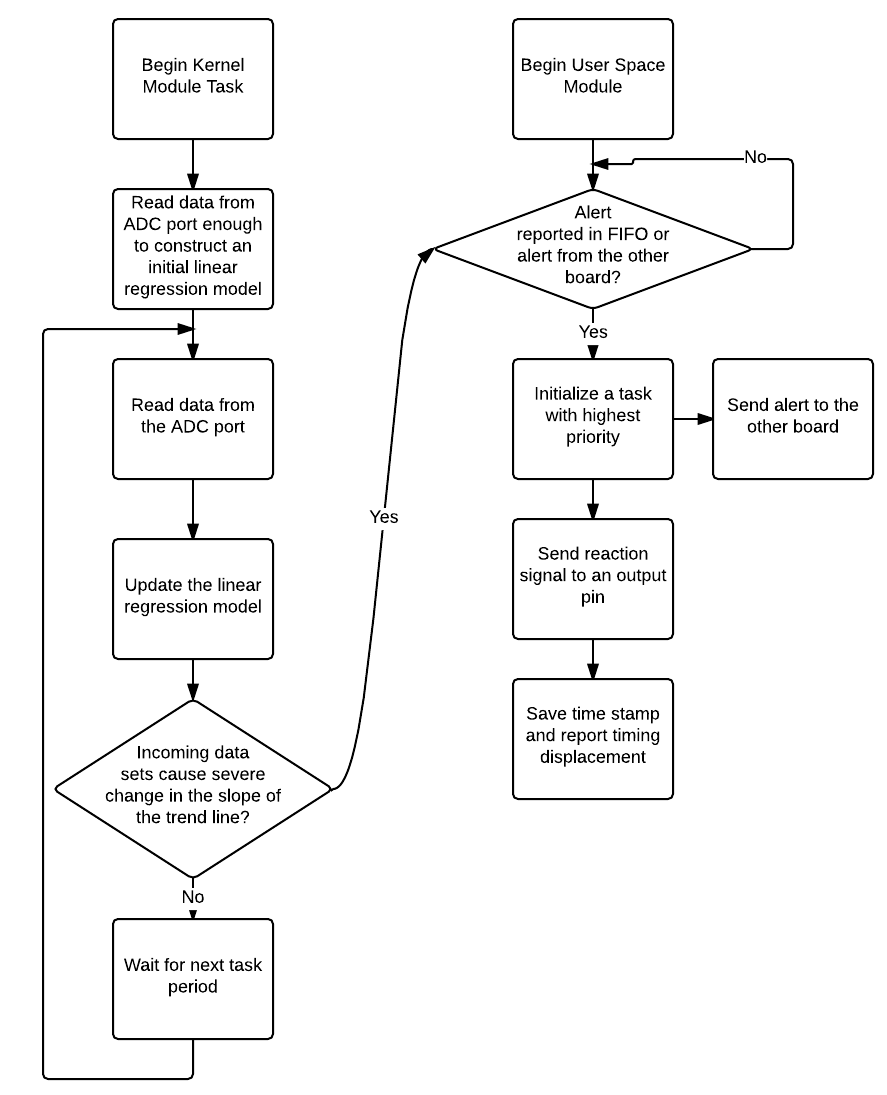
**METHODOLOGY**

The hardware component would consist of two TS-7250 boards and a hardware sensor. The sensor will feed raw data into the analog-to-digital converter on the TS-7250, which will then provide data that TS-7250 could understand. The data will be read via port mapping of the ADC output address. The communication with the second TS-7250 would be done via network communication.



*Figure 1: Hardware Flow Chart*

For the software, each TS-7250 would read in data from the ADC in the kernel module. The threat detection would be done via linear-regression model. When the system starts, there will be a calibration time waiting for the linear regression model to be constructed. When that is done, a real time task is constantly reading in new data from the ADC and update the linear regression model. If the slow had grown steeper than a certain limit , then we could conclude that the sensor is detecting an imminent physical contact, thus a threat had been detected. When the threat had been detected, the threat signal is pushed into the FIFO, which would be read by the User Space module. When the user space module detects the threat, the reactionary mechanism would then be carried out. A time stamp would be saved, a communication with the other board would be established, and a reaction signal would be outputted.

**

*Figure 2: Software Flow Chart*

**TIMELINE**

4/8/2015 – Hardware interface success, data could be read in from the ADC

4/15/2015 – Linear Regression algorithm successfully implemented

4/22/2015 – Kernel module successfully implemented, could report threat accurately

4/29/2015 – User space module implemented successfully, TCP/IP communication established, and time stamps recorded. Experiments to be performed.

**STRENGTHS AND LIMITATIONS**

The strength of this system is in its novel approach to machine resiliency. This project seek to influence the design philosophy of automata, allowing for a bright future of less insurance claim and less frustration over human errors, that might as well be attributed to acts of God in its ability to destroy technologies. The strength of this system is that it is a start in a right direction of the future of embedded system design, to create a one-size-fits implementation to protect all technologies. The limitation, of course, is that this is merely a prototype. Thus the integratability of this system is still in question. The lack of specific use-cases may detract future investors. However, with further developments, with this system becoming part of the standard operating system kernels and gaining an IEEE standard, this project would surely make a positive influence on the global community in its ability to achieve higher standard of living of consumer electronics. People would be happier with less broken electronics, billions of dollars would be saved from having to buy new materials, and the economy as well as the overall happiness rating of the world would improve. The United States would thus have less reliance upon China in manufacturing as people break less stuff, thus reducing the annual deficit, which would eventually lead to America fully paying off all its debt, thus becoming an independent financial and technological superpower once again.