How to Perform Real-Time Processing on the Raspberry Pi

Steven Doran
SCALE 13X

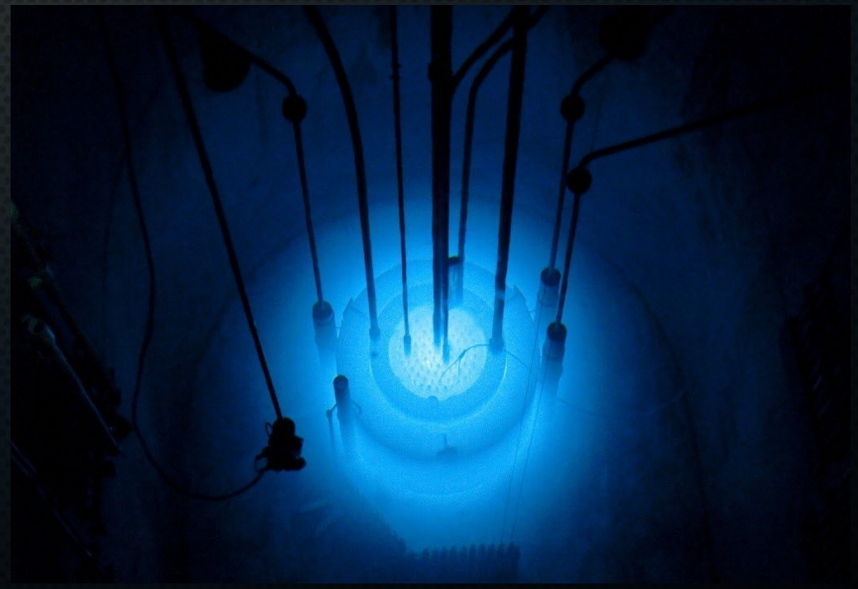
Outline

- What is Real-Time?
- What is the Raspberry Pi?
- Can the Raspberry Pi handle Real-Time (And why would you want to?
- Why would you want to use the Raspberry Pi for Real-Time?
- Example Project
- Questions

What is Real-Time?

An application that requires a guaranteed response within a strict timing constraints

Some Examples of Real-Time



Soft Real Time vs. Hard Real-Time

- Hard Real-Time – If the event is not processed in a strict timing window then bad things will happen

- Soft Real-Time – If the event is not processed in a not as strict timing window then the system may degrade but its not as bad

Hard Real-Time Example

Hard Real-Time Example (Gone Wrong)

Soft Real-Time Example

What is Real-Time is NOT!

#1 Software that runs fast

What is Real-Time is NOT!

Software that runs fast

Example:

A processing Thread w/ a execution requirement of 2.5ms, but the system can only supports 5.0 ms

What is Real-Time is NOT!

#1 Software that runs fast

Example:

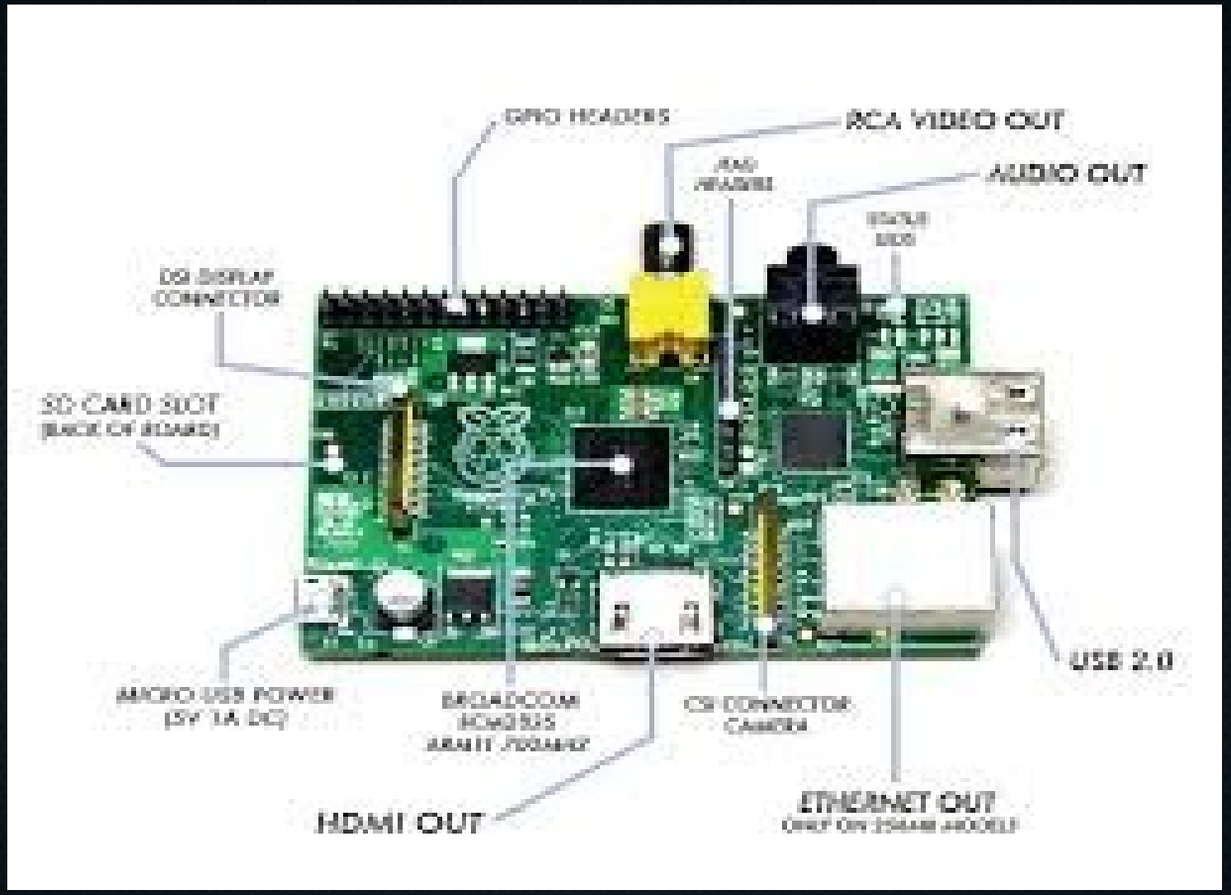
A processing Thread w/ a execution requirement of 2.5ms, but the system can only supports 5.0 ms

#2 Just because it is embedded

Real-Time Metrics

- Interrupt Latency – Time to process an interrupt
- Scheduling Latency – Time for the OS Scheduler to start a processing task (Nominal and Worst-case)

What is a Raspberry Pi?



The Raspberry Pi is a small computer w/ 700 Mhz Single Core Atom and up to 512 Mb memory all for $25

Raspberry Pi (The Good)

- It's cheap
- It is easy hacked (Easily Configurable)
- It comes with GPIO connectors for device input / output
- It supports 1080p video

Raspberry Pi Projects

Raspberry Pi (The Bad)

- No Real-Time Clock
  - Expects device to be always connected to the internet
  - Cannot generate deterministic timing pulses to control things like DC motors
- RAM cannot be expanded
- The built-in Network Interface is really a USB device and shares the same bus on any USB device connected to it
- Power issues especially with the first hardware version

Can the Raspberry Pi handle Real-Time (And why would you want to?)

- Kinda with allot of tweaking
  - Current Linux / BSD kernels for the Raspberry Pi do not support real-time
  - Standard Linux or BSD install includes services that generate lots of overhead
  - No real-time clock
- Because it is cheap and flexible

A Sample Raspberry Pi Project that benefits from Real-Time – Nerf Tank

- Let's say we want to have a Raspberry Pi control a deadly Nerf Tank
  - Pi needs to detect the baddy
  - Pi needs to move the turret to aim the Nerf gun at the baddy
  - Pi needs to fire the deadly Nerf projectile at the baddy (if in range)
  - Pi needs to make sure the tank does not collide with anything while it is doing the above steps

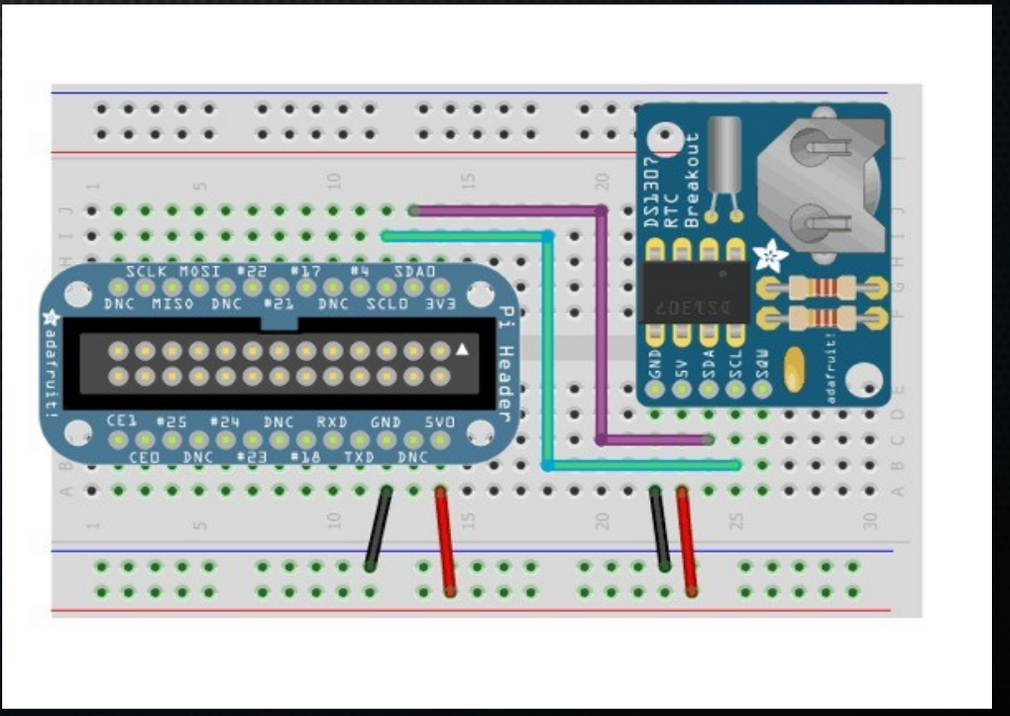
  Pi needs to do all of the above at the same time (Multi-threaded)

Nerf Tank Real-Time Problem #1 – No Real-Time Clock

Problem Statement

- Nerf Tank will have timing requirements and will not be always (if ever) connected to the Internet

- Solution
  - Add a Real-Time clock



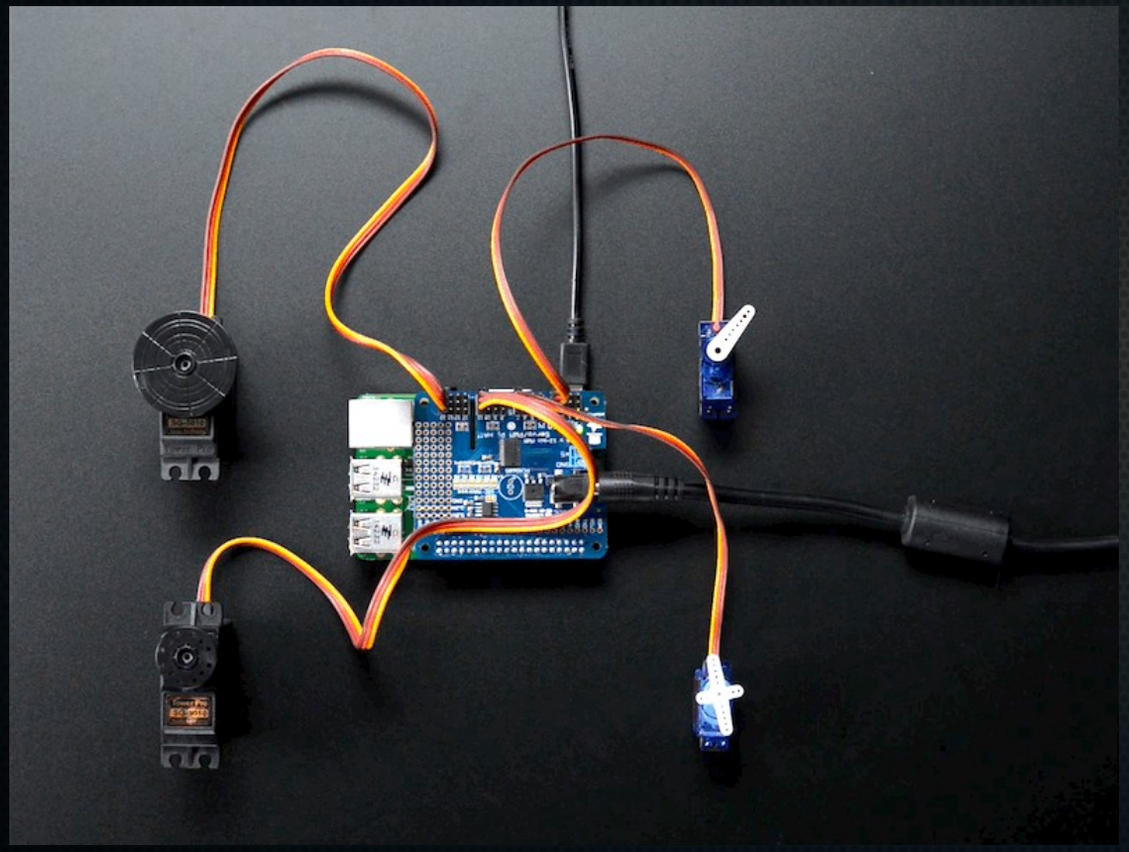
Nerf Tank Real-Time Problem #2 – Pulse Generation

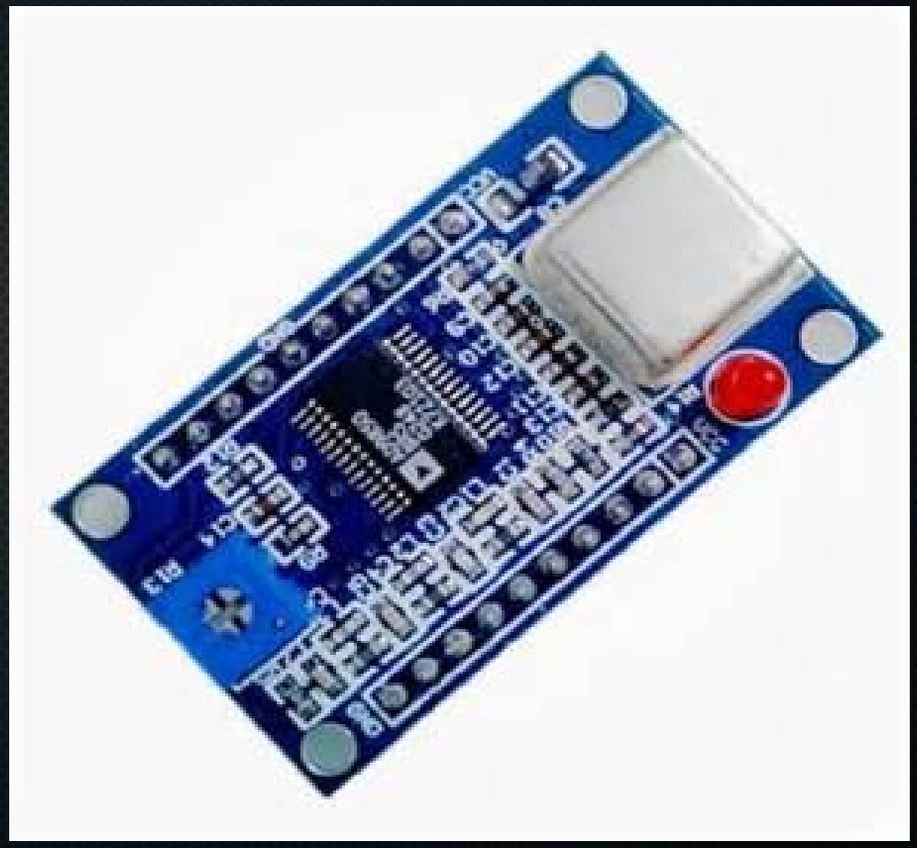
Problem Statement

- Even with a Real-time clock the Raspberry Pi will not generate the deterministic timing pulses we need to move the drive motors and turret with accuracy

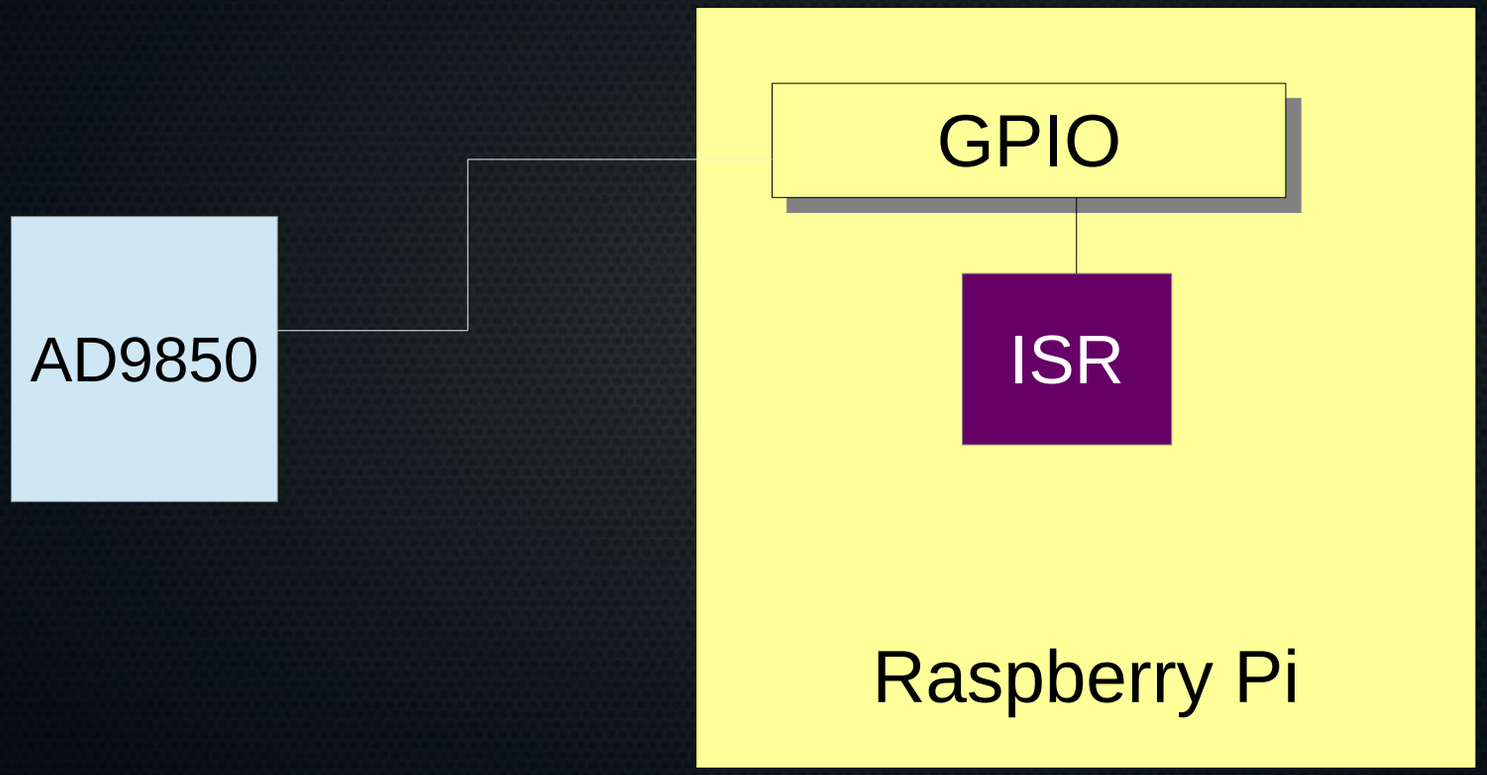
- Solutions
  - Connect a AD9850 Pulse generator to the Raspberry Pi
  - Connect a AdaFruit Servo HAT to the Raspberry Pi

Nerf Tank Real-Time Problem #2 – Pulse Generation





AD9850 Pulse Generator connected to the Raspberry Pi



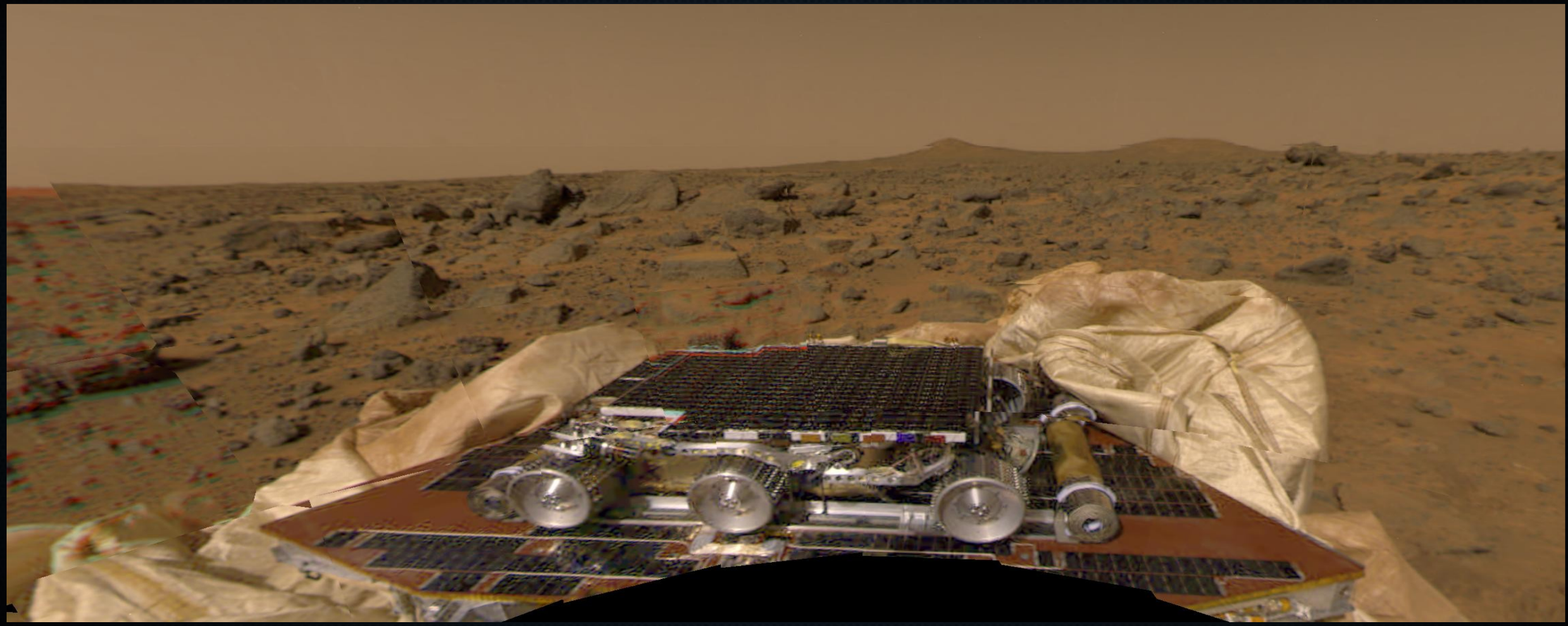
Nerf Tank Real-Time Problem #3 – Multi-Threaded Applications

Problem Statement

- The software that will control the Nerf Tank will require multiple processing threads with unique timing and priorities that the *stock* Raspberry Pi distribution kernels cannot guarantee
- Processing tasks share resources

- Solutions
  - Use RTEMS
  - Build your own real-time Linux kernel from source
    - Who uses *make menuconfig* anymore?
  - Remove all unused services

Fun Example of Priority Inversion



Real Time Executive for Multiprocessor Systems (RTEMS) for Raspberry Pi

- Open Source Real Time Operating System (RTOS)
- Used in Spaceflight, medical, and other real-time embedded applications
- Ported to the Raspberry Pi *(w/ some limitations)*

Nerf Tank Real-Time Problem #4 – Maximum CPU Performance

Problem Statement

- The Raspberry Pi's CPU is not set at its maximum performance using default kernels

- Solutions
  - Overclock the CPU
    - Using RaspBian
      - Raspi-config
    - Remove all unneeded services

Summary

- We discussed what Real-Time means
- We described the Raspberry Pi w/ its Pro's and Con's
- We used a Nerf Tank example on how to tweak the Raspberry Pi to perform real-time processing

Questions