

Design doc:

Schedule section is a wild guess. Break into general tasks and figure out what's parallelizable, assign to group members

Possible Risks and Risk Management is the most important section. Most projects depend on unforeseen risks. To manage these: "get your hello worlds working"

Get ~3 Raspberry Pis networking the way you want, get camera data going through the network. Do this *this semester*.

Connecting nodes to campus network for updates/software management, taking them off the network for ad-hoc work creates a lot of downtime. Should they be left in the senior design lab, etc.? Should we use a firewall...?

What is the process for updating software for all the nodes? Repetitive processes are a risk (development environment is a risk)

Could connect a Raspberry Pi to a development PC; use a switch to connect multiple Pis

How do we order parts? See slides from Lee Harker. Give ETG a bill of materials, CC Dr. D. **Get 3-4 Pis to begin with (1 sensor node, 1-2 forwarding nodes, 1 network master)**

There is a raspberry pi case with a hole that has a holder for the PiCamera.

For each functional block: Break down into a few milestones, make a guess of how much time each will take. These are guesses.

Get development environment set up too.

ZeroMQ may be helpful as a messaging system. You have to deal with the fact that you don't know what state the other Pis are in. ZeroMQ can deal with this automatically if nodes crash and reboot...

When we order network interfaces – be careful we get something we can actually use with a Pi.

The onboard WiFi on the Pi 3 does not support mesh.

When each Pi boots, figure out: what am I? (relay node, camera node, network master)

Look for cases with battery, etc. Buy off the shelf as much as possible.

Repo licensing: One of the GPLs or SDs. GPL might be an issue if we needed to include a driver in the project. The university won't care

For tasks, make a dependency graph. What does each task depend upon? This determines which order we do the tasks in / which are parallelizable.

Hello world on Pi: Boot up, see camera stuff from base node

From a software engineering perspective, there's a huge difference between this and other projects. We're managing OS changes...; trying to get stuff to work. We're probably not going to check in the entire OS (maybe *etc*). How do people typically manage these configurations in a git repo? Take/keep notes of how we are doing this. Use the Wiki in GitLab?