*The following section will outline the hardware required to run the critical systems computer, the software architecture within its operating system and the specifications according to which applications in the Critical Systems Computer will be run.*

### LPC2468 Microcontroller

\*Why we chose LPC2468

ARM7

External Memory – (EMC)

I/O

\*Relevant feature sets

\*intention with design

### Memory

### The FreeRTOS and Software Architecture

For the sake of robustness, BLUEsat Software has been designed around the idea of static Operating System images. That is to say that for a given ‘image’, the satellite will function in a predetermined and unchanging manner according to a predefined set of stimuli. In order to change any of the core functions or procedures on the satellite, a new Operating System image needs to be ‘burnt’ or uploaded onto the satellite.

BLUEsat will be running a simple, lightweight, open-source operating system called the [Free Real Time Operating System (FreeRTOS)](http://en.wikipedia.org/wiki/FreeRTOS). The FreeRTOS itself provides no existing services other than a kernel.

Each peripheral hardware device running on BLUEsat will be controlled by a set of software drivers. Drivers abstract commands of devices and subsystems on the satellite (for example, polling data from the Telemetry system). They make up the basic interface and abstraction between the Critical Systems Computer and the rest of the hardware on the satellite.

In turn, these drivers are called upon according to a set of instructions defined by Applications. BLUEsat Applications will be written in order to address specific procedures required for particular operations that the satellite will need to do. For example, the Telemetry Application will need to poll data, raise flags on seemingly malfunctioning systems, compress this information within memory and pass it onto the communications system when commanded by Groundstation to do so.

The specifications for the functionality of each Driver and Application are included in this report specific to the subsystem that it addresses.

The software architecture has been designed around a central “command task”. The command task executes applications and exchanges data between them according to the current operating mode of the satellite. The priority with which certain applications are executed will be defined by either the operating mode of the satellite or direct commands from the Groundstation. This operating mode will be defined by a status manager which processes DTMF commands sent from the Groundstation by a BLUEsat administrator.

The full list of Operating Modes and what each mode entails is detailed below. A full list of the current list of DTMF commands is also given below.

\*add figure and flow diagram

### DTMF Commands

\*NEED TO DEFINE

### Operating Modes

\*NEED TO DEFINE