Arm R Series
RTOS
Hexacopter Flight Control

Overview: We will assume the reader has read the Arm M series report. As in that report, we will give the application a quick overview and generate some basic requirements. Based on this we will pick a processor and a software architecture.

Application: We will look at a flight control system for a hexacopter to deliver packages in an urban environment. Similar to the ABS system, we need to read several analog signals and acuate several motors. Moreover it needs internet connectivity in order to navigate the urban environment and possibly a GPS receiver to get its location. Thus we can generate some requirements:

1. Ethernet (to provide interface to a cellular modem or some similar connectivity device so the quadcopter can connect to the web).
2. 6 PWM modulators
3. Digital signal processing / ADCs to read accelerometer data
4. Floating point arithmetic
5. Must analyze accelerometer data and acuate motors sufficiently fast as to control the vehicle. This is a hard time requirement as there exists some maximum acceptable latency such that if not met, then vehicle could crash and cause physical harm (its operating in a city)
6. Must provide minimal layer 3 networking services. I.e. it must respond to pings and other layer ICMP messages
7. Must provide flight control service
8. Must provide location ping service
9. Must provide its current state in the delivery i.e. has it been delivered, is it coming back to the distribution center.
10. Must accept and provided delivery queue services i.e. I should be able to add deliveries to the vehicle and query to see what its going to deliver next.

Processor: The arm R series seems to be a good fit for this application. We need an order of magnitude more memory and compute power than the M series can give us. However we also have one key requirement that requires hard real time response times. That is the flight control service. Moreover many R series boards such as the TMS570LC4357 provide PWM modulators, ethernet, GPIO, and DSP modules. Thus the R series is a good choice that gives us the compute power we need to host all these complicated services with the low latency and determinism we need, all with the peripherals we also need.

Software Architecture: Given the number of services and especially the networking services we need, a cyclic executive is off the table. Thus to choose between an RTOS and a real time linux, we need to look at 2 things. 1) will Linux give the deterministic latency we need and 2) if we use an RTOS will we have to implement all sorts of drivers and services that Linux already implements. And in this case I believe an RTOS wins. For one the flight service is ultimately the number one requirement. We cannot have our drones falling out of the sky in the middle of a city. Moreover, we do not need any niche or services that we get for free on Linux. Most RTOS packages should include ethernet and IP layer services. Thus an RTOS is the right choice for this application.