A blue path indicates each time the task yields This entire diagram applies inside the IOT Process Loop state. We must circulate execution back to the RTOS scheduler. through this loop continuously. Failure to arrive here means that the MQTT stack will not be serviced -- and you will not be able to send or receive any IOT messages. Keep in mind that some messages arrive spontaneously and there no way to predict their Process\_Loop() Flow Chart arrival, so when not doing something specific, we pass through this state perpetually. The more frequently we pass through this state, the more quickly our software will respond to incoming AWS IOT messages. For Subscriptions, we know how to retry failures. No mgttStatus == globalSubAckStatus == Entry from all previous states. MQTT ProcessLoop() BackoffAlgorithm\_GetNextBackoff() MQTTSucess? **MQTTSubAckFailure** Yes Nο AllJobsStep turns Neutral just after Shadow Subscribe Delta state Yes backoffAlgStatus Sub == BackoffAlgorithmSuccess AllJobsStep == ShadowStep == IOT ALL JOBS::Neutral IOT SHADOW::Idle We exhausted our Yes Subscription attempts Νo Allow the Shadow to finish This is the first time and the only time we run AllJobsStart because No opIOT = IOT OP::Disconnect: this registers the system for for DisconnectStep = IOT MQTT DISC::Start; Jobs notifications. blnSvstemRequestsShadowUpdate = false: blnSystemRequests Exit for processing opIOT = IOT OP::Shadow; ShadowUpdate ShadowStep = IOT SHADOW::Shadow Update Delta; True False This is the first time AWS processes may occasionally run concurrently, but we run for AllJobsGet. don't want to start them all simultaneously. We apply this logic chart to prioritize any pending actions in this order: blnAllJobs StartSubscriptions = false; blnAllJobs\_StartSubscriptions Shadow/Delta Update AllJobsStep = IOT ALL JOBS::Start; Run Jobs opIOT = IOT OP::All Jobs; True Download All Jobs Prioritize previously unfinished Jobs Retrieves the first Job False Starts/Runs that job Runs all pending Jobs Upon Startup, we manually trigger these actions but, after that point, all actions are triggered on events. This logic see subsection of blnJobRunning flow will continue to apply in part to coordinate actions **All Jobs Flow Chart** . False which are started from these events: Delta Update occurs for data/state changes New Job arrives from AWS True