* Race conditions with initialising a task handler before the task is defined
* Fastest cycle time = 0.02 = 20us with no code on C6 single core
* This code tricked me into thinking that my watchdog didn’t work lol, thought that it still exceeded after x time when in reality I turned it off after x time

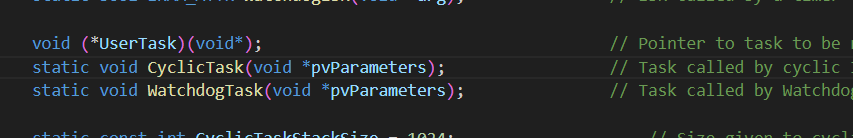
A screen shot of a computer program

Description automatically generated

* Fastest speeds:  
    
  A screen shot of a computer

  Description automatically generated
* Fastest speed with wifi = 1ms, wdog 0.8ms. Not any more lol. Fastest speed will be application specific. This was with a single core though.
* When switching to S3 chip, tasks EspNowTask and UdpProcessingTask need more stack size to work. Unsure why this is for now, but not a major issue. Could be due to different core architectures processing commands differently, and different storage mechanisms in each
* C6:  
  A screen shot of a computer

  AI-generated content may be incorrect.
* S3:  
  A screen shot of a computer

  AI-generated content may be incorrect.
* IRAM usage on S3 is worryingly high. Research online shows that this is expected.
* IsAccessPoint needs to be constexpr to init at compile time, else the app\_main breaks by running the timer code twice.
* Defining the void user task before the cyclic task crashes the system  
    
    
  In fact, it must be declared at the bottom of the private area.  
  A screenshot of a computer program

  AI-generated content may be incorrect.  
    
  Further testing shows that it must be declared after volatile variables. Idk man
* Longest run:  
  A screen shot of a computer

  AI-generated content may be incorrect.