

BackligtCC int listIDPos; unsigned int host; struct host\_cpu\_load\_info currCpu; struct host cpu load info prevCpu; float powerScale; float timeActive; unsigned int kDVD1Level; \_Bool \_useDirectMap; unsigned int minReachableLoadIndex: float directMapSlope: float \_directMapIntercept; · (struct \_\_CFString \*)copyFieldCurrentValueForIndex:(int)arg1; (struct \_\_CFString \*)copyHeaderForIndex:(int)arg1; (int)numberOfFields; · (\_Bool)usesPID; · (void)refreshCurrentLoadingIndex; · (void)refreshCurrentLoadingIndexUsingPID; · (unsigned int)getUserUsage; · (void)defaultAction; (id)initWithCPU:(const char \*)arg1:(int)arg2:(struct \_\_CFDictionary \*)arg3:(float)arg4:(int)arg5;

CpuCC int listIDPos; unsigned int host; struct host\_cpu\_load\_info currCpu; struct host cpu load info prevCpu; float powerScale; float timeActive; unsigned int kDVD1Level; \_Bool \_useDirectMap; unsigned int \_minReachableLoadIndex; float directMapSlope; float directMapIntercept; (struct \_\_CFString \*)copyFieldCurrentValueForIndex:(int)arg1; (struct \_\_CFString \*)copyHeaderForIndex:(int)arg1; · (int)numberOfFields; (\_Bool)usesPID; · (void)refreshCurrentLoadingIndex; · (void)refreshCurrentLoadingIndexUsingPID; · (unsigned int)getUserUsage; · (void)defaultAction; (id)initWithCPU:(const char \*)arg1:(int)arg2:(struct \_\_CFDictionary \*)arg3:(float)arg4:(int)arg5;

CommonProduct \*cProduct; float hotspotTemperature; float THERMAL TRAP LOAD; float THERMAL\_TRAP\_SLEEP; float releaseHysteresis; int tState; struct ForcedThermalLevelStruct forcedThermalLevel[2]; \_Bool isEngagedWithHysteresis; - (struct \_\_CFString \*)copyFieldCurrentValueForIndex:(int)arg1; (struct \_\_CFString \*)copyHeaderForIndex:(int)arg1; (int)numberOfFields; (\_Bool)shouldForceThermalLevelForThreshold:(int)arg1; (\_Bool)hotspotIsEngaged; · (int)getThermalStateofHotspot; · (float)calculateTargetDifference; · (void)calculateControlEffort:(float)arg1;

· (id)initWithSC:(const char \*)arg1:(int)arg2:(id)arg3:(struct \_\_CFDictionary \*)arg4;

SupervisorControl

int listIDPos;
unsigned int gpuPowerTarget;
int dutyCyclingCrossOverThreshold;
float timeActive;
- (struct \_\_CFString \*)copyFieldCurrentValueForIndex:(int)arg1;
- (struct \_\_CFString \*)copyHeaderForIndex:(int)arg1;
- (int)numberOfFields;
- (void)refreshCurrentLoadingIndex;
- (void)defaultAction;
- (void)calculateMitigation;
- (id)initWithSOC:(const char \*)arg1:(int)arg2:(struct \_\_CFDictionary \*)arg3:(int)arg4:(int)arg5;

**PidComponent** 

CompoentControl

- (id)initWithCC:(const char \*)arg1:(const char \*)arg2:(int)arg3:(struct \_\_CFDictionary \*)arg4;

Bool allowCEOverride;

struct \_\_CFString \*nameofComponent;

float kp; float ki; float ts;

float integrator; float integratorMin; float integratorMax; float TARGET; int mitigationType;

int boundCheckLow; int boundCheckHigh;

\_Bool allowLIOverride;

unsigned int currentLoadingIndex;

unsigned int maxLoadingIndex;

unsigned int previous Value;

unsigned int releaseMaxLI;

unsigned int releaseRate;

(void)testLoadingIndexLevel;

(void)refreshCurrentLoadingIndex;

- (\_Bool)isEqualMType:(int)arg1;- (void)filteredMaxLoadingIndex;

\_Bool thermRelease;

double lastTime;

int loopTimer;

- (int)numberOfFields;

- (void)defaultAction;

(void)dealloc;

(\_Bool)isEqualMType:(int)arg1;

(void)calculateControlEffort:(float)arg1;

- (id)initPIDWith:(struct \_\_CFDictionary \*)arg1;

NSObject<OS\_dispatch\_queue> \*myCustomQueue;

- (struct \_\_CFString \*)copyFieldCurrentValueForIndex:(int)arg1;

NSObject<OS dispatch source> \*theTimer;

(struct \_\_CFString \*)copyHeaderForIndex:(int)arg1;

int controlEffort;

(void)dealloc;