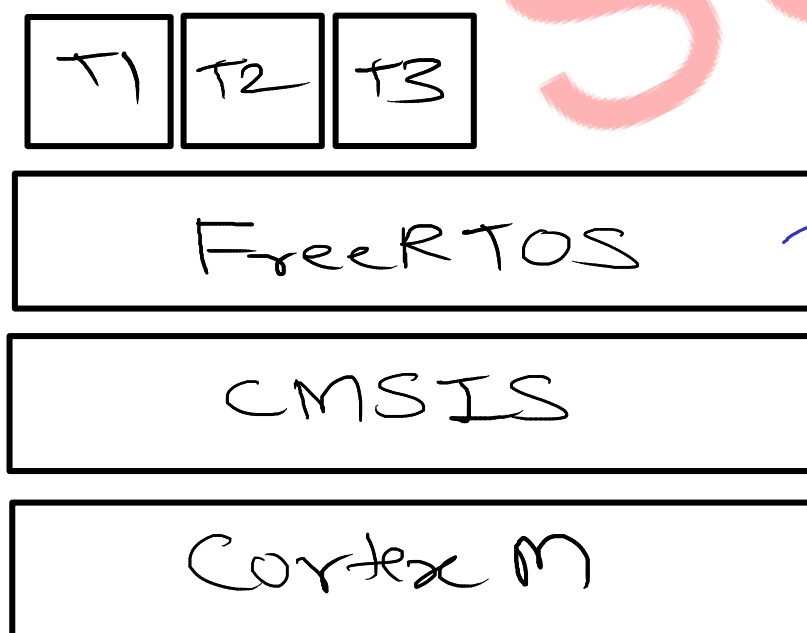
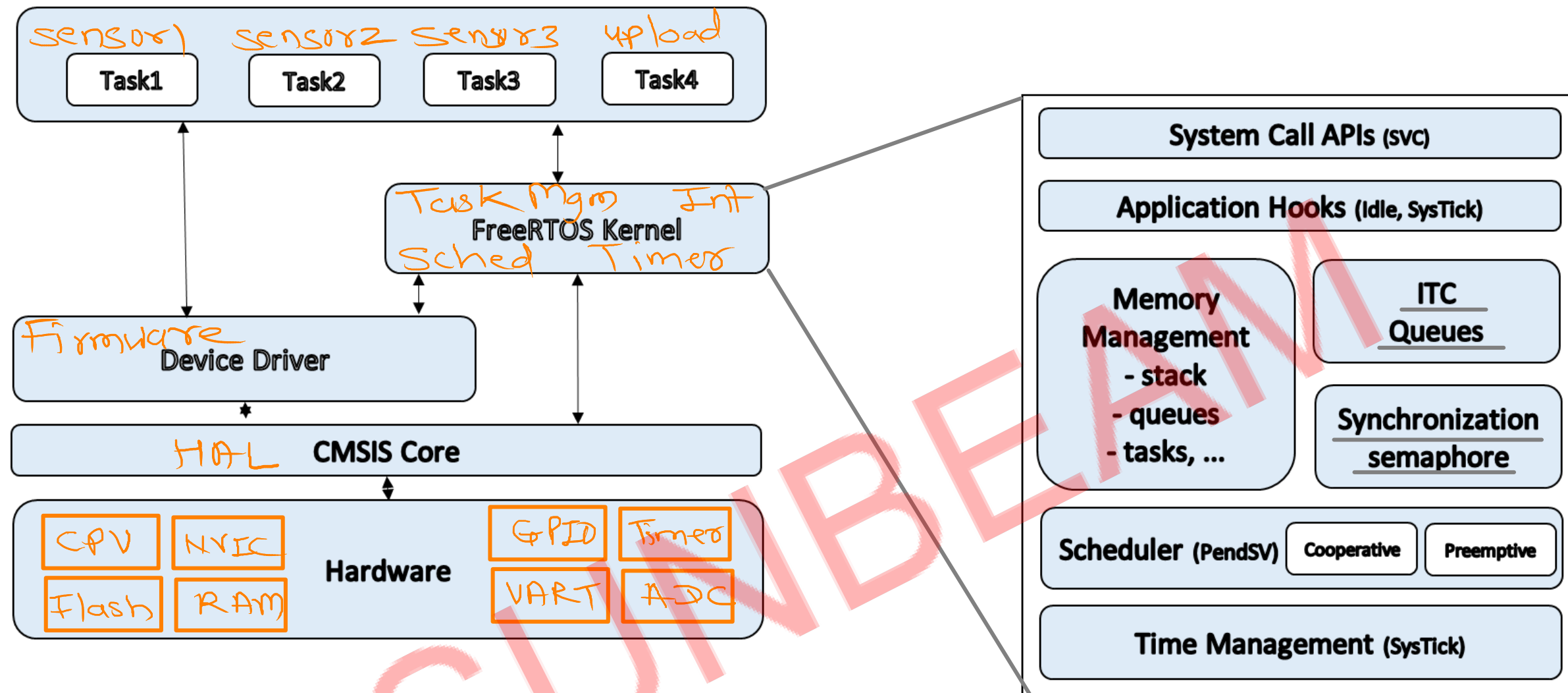


FreeRTOS Architecture

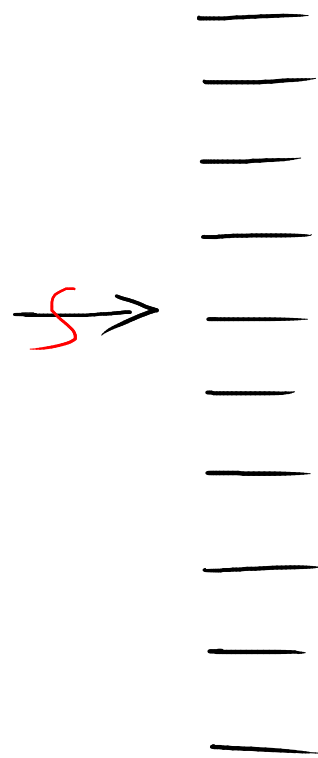


middleware
↳ OS - FreeRTOS
↳ USB stack
↳ Bluetooth stack

Interrupt Handling

Task1:

$s=0$



ISR()

$v(0)$

scheduler();

IntrTask:

$P(s)$



wid
~

IntrTask(void *prParam)

$s=0$

while(1)
~

$P(s)$;

led-onc();

delay(500);

led-offc();

\int TaskDelete();

ISR() {

$v(s)$

}



⑦

③

⑨

⑤

①

⑧

④

}

Synchronisation

1) Critical section

- code which can be executed by only one process at a time. if multiple processes execute critical section concurrently then result will be wrong.

postENTER_CRITICAL()
//critical section
≡
postEXIT_CRITICAL()

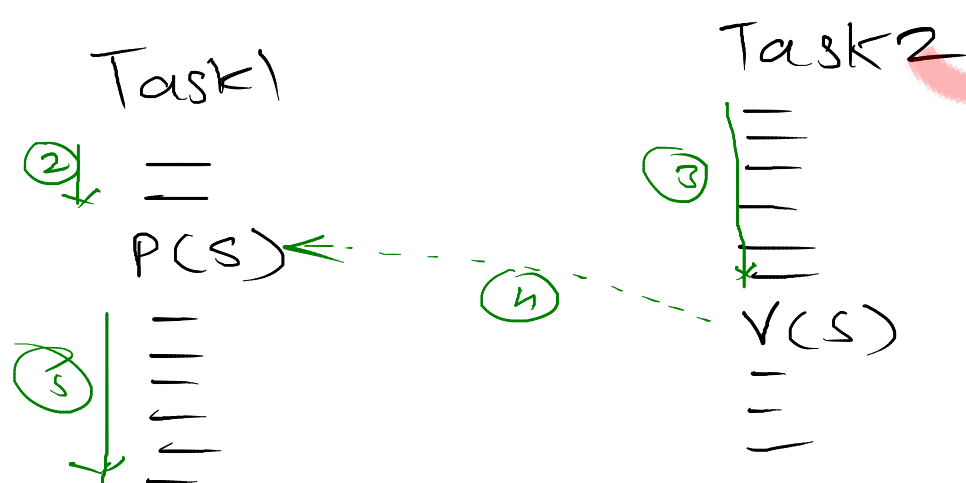
2) Semaphore

a) event/Flag

b) Binary Semaphore

c) Counting Semaphore

①
 $S=0$



$S=1$

Task 1
≡
P(S)
cnt++;
V(S)
≡

Task 2
≡
P(S)
cnt--;
V(S)
≡

$S=n$

Task 1
≡
P(S)
cnt++;
≡

Task 2
≡
cnt--;
V(S)
≡