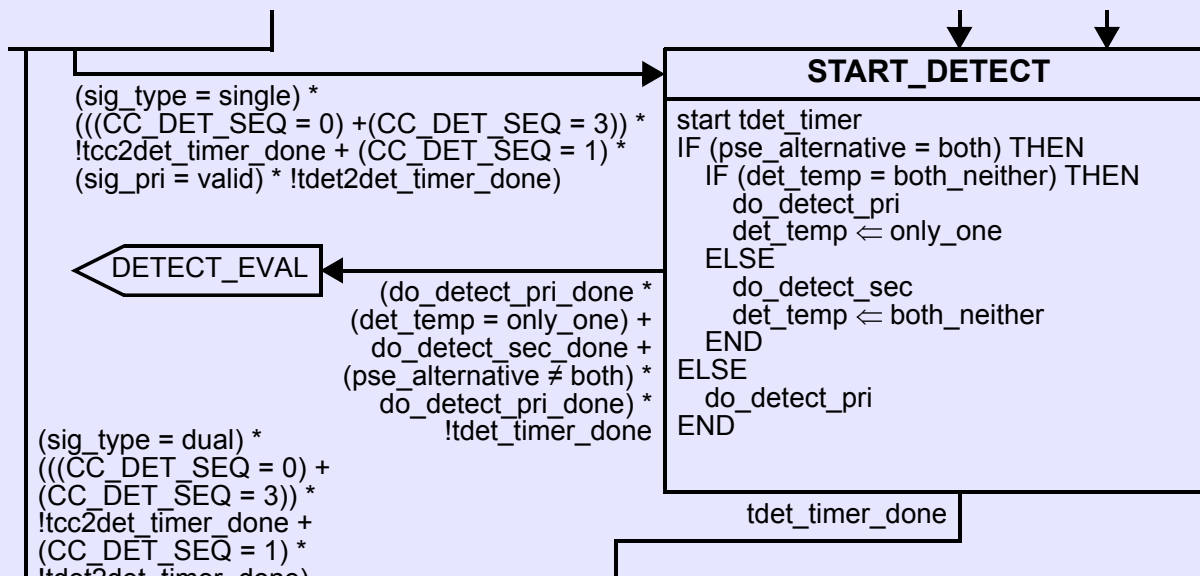


START_DETECT fix for D2.3 v100

Info (not part of baseline)

This baseline addresses an issue specific to the START_DETECT state and how 'function_done' is evaluated.



The arc from START_DETECT to DETECT_EVAL checks both **do_detect_pri_done** and **do_detect_sec_done**. Only one of the corresponding functions will have been called by the START_DETECT state, depending on the value of pse_alternative and det_temp.

What is the value of the do_detect_(pri|sec)_done, which hasn't been called ?

- If this is the first round through START_DETECT, the function has never been called. While not 100% clear cut per our current state diagram rules, the function would be considered "done". Even if we were to adopt text that said "Functions that have never been called evaluate as not done", this would not solve the issue, see next item.
- If the state diagram has had a prior run through START_DETECT and called the function, it would still be considered "done", even though it hasn't been called this time around.

Regardless of interpretation of the rules, this doesn't work.

Solution: we need to avoid checking the 'doneness' of a function that hasn't been called. Only START_DETECT has this issue.

145.2.5.7 State diagrams

Replace the arc logic from START_DETECT to DETECT_EVAL as follows:

```

!tdet_timer_done * (
(do_detect_pri_done * ((det_temp = only_one) + (pse_alternative ≠ both))) +
(do_detect_sec_done * (pse_alternative = both) * (det_temp = both_neither))
)
    
```

Info (not part of baseline)

Original arc logic in the same format:

```

!tdet_timer_done * (
do_detect_pri_done * (det_temp = only_one) +
do_detect_pri_done * (pse_alternative ≠ both) +
do_detect_sec_done
)
    
```