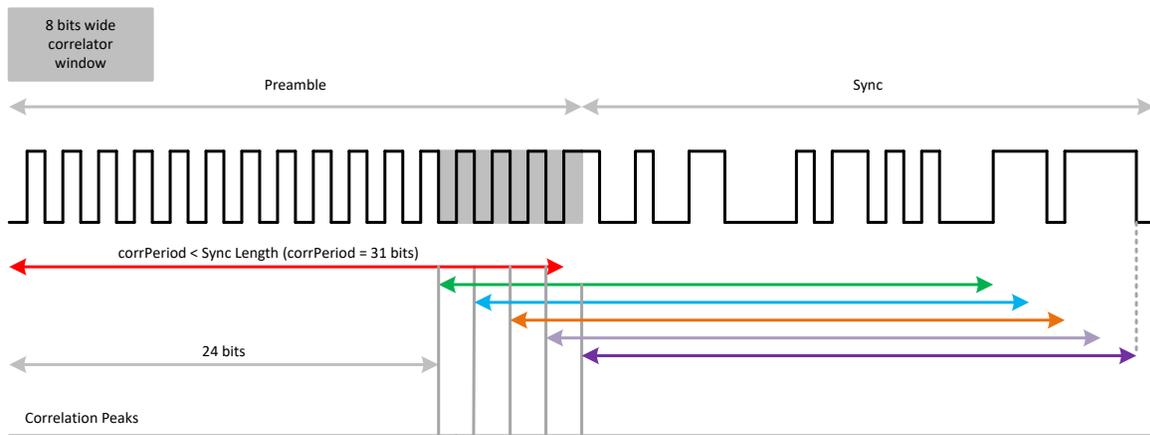


Assume the following case where CorrPeriod is shorter than the sync length:

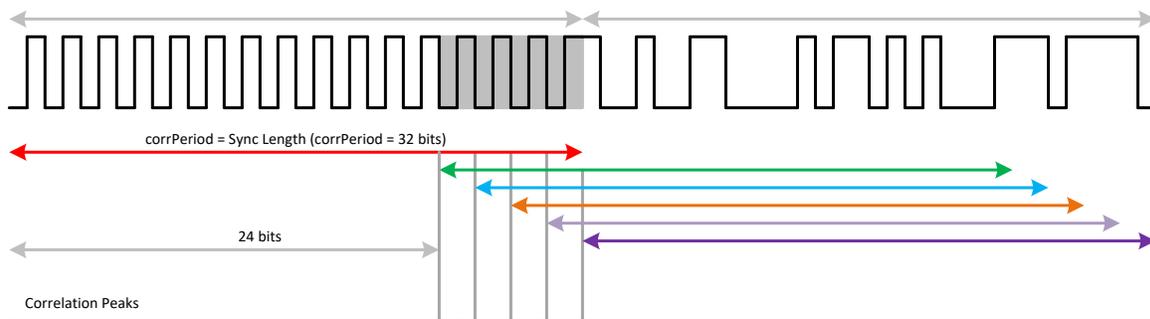
corrConfig.numCorrBusy = 1



You will get the first correlation peak 24 bits into your **first** corrPeriod.

After a correlation peak, the corrPeriod is restarted and two bits into the **second** corrPeriod you get the next peak. It continues like this for the **third**, **fourth** and **fifth** corrPeriod. Within the **sixth** corrPeriod there are no correlation peaks, and at the end of the period the Corr State will go from BUSY to IDLE. If csConf.idleOp = 1, the command will end straight away if IDLE is reported.

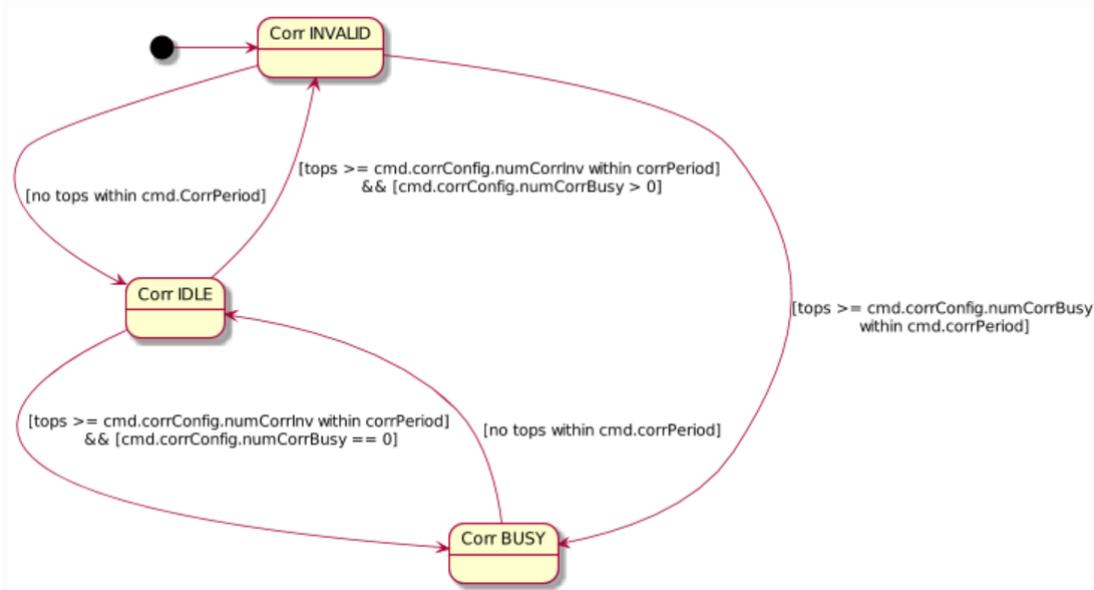
If, instead, the corrPeriod is the same length as the sync length you will have the following scenario:



The 5 correlation tops will happen at the same place, but since the corrPeriod is just as long as the sync word the radio will detect sync before/at the same time as the channel is reported BUSY. When sync word is detected, the radio will receive the packet.

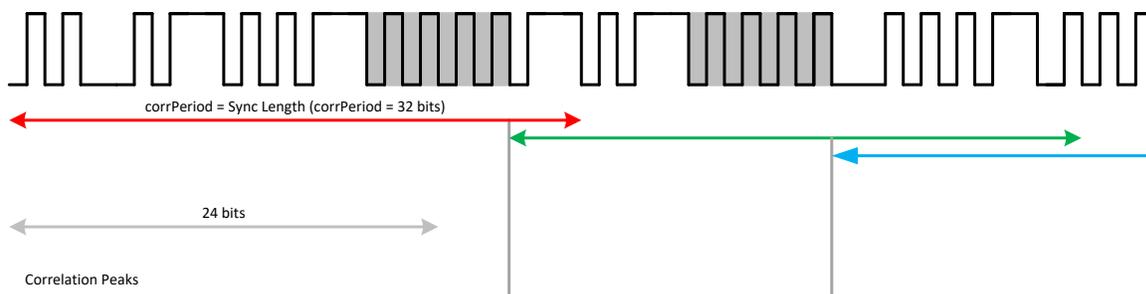
This is the reason for why corrPeriod should be longer than the sync word.

Assume the following diagram describing how the correlation state is updated:



To avoid staying awake too long because of noise you want to make it harder to get to the Corr BUSY state. The way to do this is to increase `corrConfig.numCorrBusy`.

The following figure shows a scenario where `corrPeriod = 1` and you have noise that contains preamble:



Here the corr State will be BUSY after the **first** `corrPeriod` and you will continue for the **second** and **third** period (until you reach the end trigger of the sniff command)

If you increased `corrConfig.numCorrBusy` from 1 to 2, the corr State would be set to IDLE after the first `corrPeriod` as this period only had 1 correlation peak.

The RX Sniff command does not normally take a decision on the presence of a valid preamble until the end of the `corrPeriod` and as previously shown the `corrPeriod` has to be wide enough to fit the sync word. With the carrier sense end time, it is possible to force an early check. The earliest we can force this check is when we know that we would have gotten the number of correlation peaks set by `corrConfig.numCorrBusy` in the presence of preamble. This means that if `corrConfig.numCorrBusy = 2` this would be after 24 bits (first peak) + 2 bits (second peak).

Please note that margins should be added