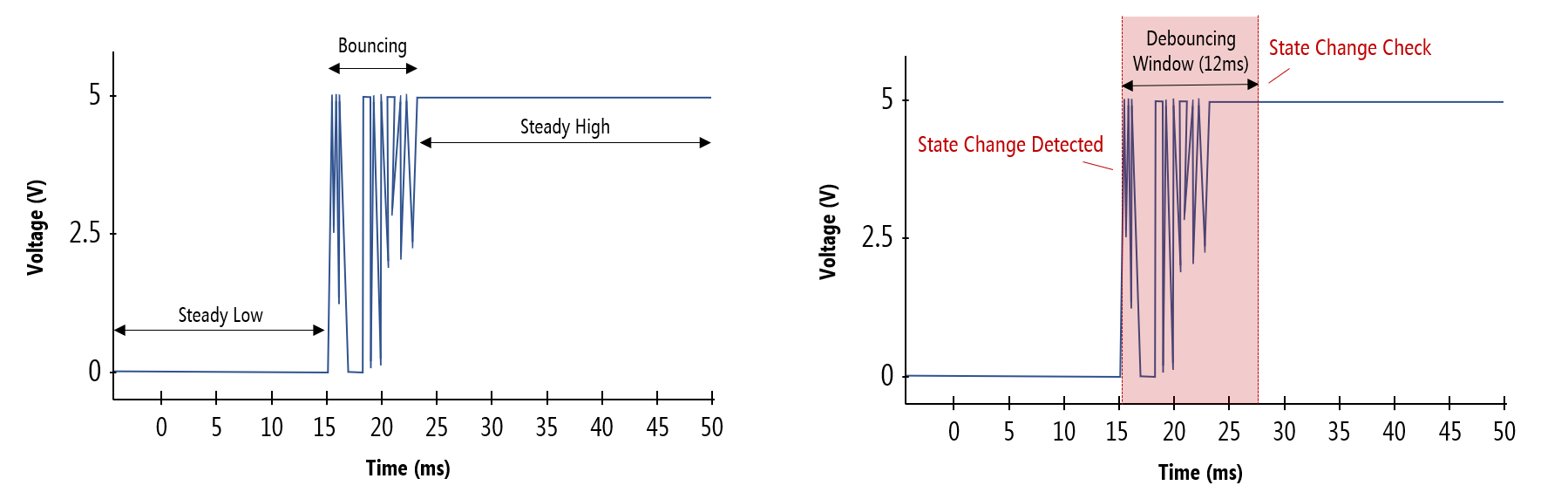
Opdracht 1:  
A screenshot of a computer program

Description automatically generated

Opdracht 2 C



Debouncing: doordat we met elektriciteit werken, komen we met natuurkunde in aanraking. Bij een knop die het signaal afgeeft voor hoog of laag moet worden, ontstaat een ruis op de lijn. Die ruis zien we aangegeven in de afbeeldingen hierboven. De oplossing wordt al gegeven in de tweede afbeelding. Omdat we niet willen dat een signaal binnen een paar milliseconden tientallen keer aan en uit gaan, moeten we hier een oplossing voor vinden. De oplossing om bij te houden wanneer de staat van het signaal veranderd en dit een paar milliseconden later nogmaals te controleren voordat de staat daadwerkelijk veranderd wordt.

Opdracht 4

1. What is the initial state of a binary semaphore? Empty or given?

Hij is leeg. Dit betekent dat de semafoor beschikbaar is en kan worden ingenomen door een taak

1. What is the initial state of a counting semaphore?

Dit hangt af van de waarde die je op het begin verteld aan de code. Stel de waarde is 3 kunnen er maximaal 3 taken de semifoor innemen voordat verdere aanvragen worden geblokkeerd.

1. How does the counting semaphore prevent a deadlock in the dining philosophers

problem? Als je voorbeeld heb je allemaal mensen aan een eettafel. Deze mensen willen allemaal eten met een vork. Wat je wilt vermijden is dat elk persoon een vork vasthoud en wacht op de andere, wat resulteer in een circulair wachtpatroon. Dit kan je voorkomen om een controle ervoor te doen dat mensen pas mogen eten als alles op hun bord ligt.