MODULE *Microcontroller*

EXTENDS *Naturals*

VARIABLES *value*, *state*, *noOfPulses*, *period*, *count*, *interrupt*, *input*

$$
\begin{aligned}
\mathit{MicrocontrollerInvariant} \triangleq\ & \land\ \mathit{value} = \{\text{“Valid”},\ \text{“Invalid”}\} \\
& \land\ \mathit{state} = \{\text{“Off”},\ \text{“On”}\} \\
& \land\ \mathit{noOfPulses} \in (1\,..\,100) \\
& \land\ \mathit{period} \in (1\,..\,100) \\
& \land\ \mathit{count} \in (1\,..\,100) \\
& \land\ \mathit{interrupt} = \{\text{“None”},\ \text{“Overflow”},\ \text{“Empty”}\} \\
& \land\ \mathit{input} \in (1\,..\,100)
\end{aligned}
$$

$$
\begin{aligned}
\mathit{StorePWM}(n) \triangleq\ & \land\ \mathit{value} = \text{“Invalid”} \\
& \land\ \mathit{state} = \text{“On”} \\
& \land\ \mathit{noOfPulses} = n \\
& \land\ \mathit{value}' = \text{“Valid”}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{InvalidatePWM} \triangleq\ & \land\ \mathit{value} = \text{“Valid”} \\
& \land\ \mathit{value}' = \text{“Invalid”}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{SetPeriod}(n) \triangleq\ & \land\ \mathit{state} = \text{“On”} \\
& \land\ \mathit{period}' = n
\end{aligned}
$$

$$
\mathit{CheckOverflow} \triangleq\ \land\ \mathit{interrupt} = \text{“Overflow”}
$$

$$
\begin{aligned}
\mathit{Increment} \triangleq\ & \land\ \mathit{state} = \text{“On”} \\
& \land\ \text{IF}\ \mathit{count} = \mathit{period}\ \text{THEN} \\
& \quad \land\ \mathit{interrupt}' = \text{“Overflow”} \\
& \quad \land\ \mathit{count}' = 0 \\
& \quad \text{ELSE} \\
& \quad \land\ \mathit{count}' = \mathit{count} + 1
\end{aligned}
$$

$$
\begin{aligned}
\mathit{PowerOn} \triangleq\ & \land\ \mathit{state} = \text{“off”} \\
& \land\ \mathit{state}' = \text{“on”}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{PowerOff} \triangleq\ & \land\ \mathit{state} = \text{“on”} \\
& \land\ \mathit{state}' = \text{“off”}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{PWMactions} &\triangleq \mathit{StorePWM}(\mathit{input}) \lor \mathit{InvalidatePWM} \\
\mathit{TCactions} &\triangleq \mathit{Increment} \lor \mathit{SetPeriod}(\mathit{input}) \\
\mathit{GPIOactions} &\triangleq \mathit{PowerOff} \lor \mathit{PowerOn}
\end{aligned}
$$

$$
\begin{array}{l}
\textit{Next} \triangleq \lor\ \textit{PWMactions} \\
\qquad\qquad \lor\ \textit{TCactions} \\
\qquad\qquad \lor\ \textit{GPIOactions}
\end{array}
$$

$$
\begin{array}{l}
\textit{InitializeMicrocontroller} \triangleq \land\ \textit{MicrocontrollerInvariant} \\
\qquad\qquad\qquad\qquad\qquad\quad \land\ \textit{state} = \text{“off”} \\
\qquad\qquad\qquad\qquad\qquad\quad \land\ \textit{count} = 0 \\
\qquad\qquad\qquad\qquad\qquad\quad \land\ \textit{period} = 0 \\
\qquad\qquad\qquad\qquad\qquad\quad \land\ \textit{noOfPulses} = 0 \\
\qquad\qquad\qquad\qquad\qquad\quad \land\ \textit{input} = 0
\end{array}
$$

$$\textit{MicrocontrollerSpec} \triangleq \textit{InitializeMicrocontroller} \land \Box \textit{Next}$$

THEOREM $\textit{MicrocontrollerSpec} \Rightarrow \Box \textit{MicrocontrollerInvariant}$