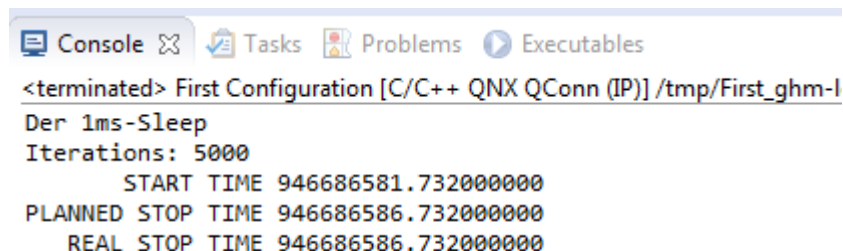


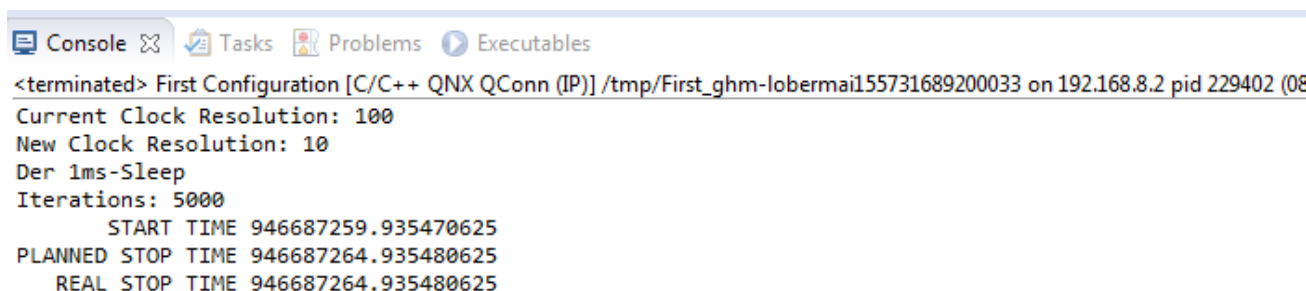
2. Aufgabe: Fester Zeittakt

- a) Der Takt von 1ms wird durch eine while-Schleife realisiert, die so lange ausgeführt wird, bis die eingegebene Zeit gewartet wurde. Um die vergangene Zeit zu messen, wird die Zeit zu Beginn des Vorgangs gemessen, diese speichern wir in der Variable *starttime*. Die Variable *deadline* wird verwendet, um die Zeit in den eingegebenen Sekunden festzulegen. *Current* wird benötigt, um die aktuelle Zeit mit der *deadline*-Zeit zu vergleichen und die Schleife abubrechen. In *aftertime* wird nach dem Sleep berechnet, was die aktuelle Zeit nach dem Step sein müsste. In der Schleife muss außerdem berechnet werden, wenn die Anzahl der Nanosekunden eine Sekunde erreicht hat, um die Sekunden korrekt hochzuzählen. Zuletzt gibt es noch eine Laufvariable, die die Schleifendurchläufe zählt.
- So wird in der Schleife bis zu der geforderten Zeit gewartet. Am Ende kann dann die *starttime* mit der erwarteten *deadline* und der *aftertime* verglichen werden. *Deadline* und *aftertime* sollten hierbei dieselbe Zeit enthalten und die Differenz die anfänglich geforderten Sekunden ergeben. Bei einer Wartezeit von fünf Sekunden enthält die Laufvariable am Ende 5000, dies entspricht genau fünf Sekunden, wenn mit jedem der 5000 Schleifendurchläufe 1ms gewartet wird



```
<terminated> First Configuration [C/C++ QNX QConn (IP)] /tmp/First_ghm-l
Der 1ms-Sleep
Iterations: 5000
      START TIME 946686581.732000000
PLANNED STOP TIME 946686586.732000000
      REAL STOP TIME 946686586.732000000
```

- b) Die Funktion *changeSystemTick* soll die aktuelle Auflösung der Systemuhr ermitteln, dies wird durch ein *_clockperiod*-Struct realisiert, das einmal für die aktuellen Ticks und einmal für die veränderten Ticks benötigt wird. *ClockPeriod* in Zeile 23 speichert die aktuellen Auflösung der Systemuhr in *clockRes*, während *ClockPeriod* in Zeile 33 die neuen Auflösung setzt.
- Um die neuen Ticks dynamisch anpassen zu können, werden diese im Methodenaufwurf in *clockRes_new* gespeichert, bevor diese Variable im zweiten *ClockPeriod*-Aufruf gesetzt wird.
- In der Main wird dann die Funktion aufgerufen, der geringste Wert, auf den die Systemuhr gesetzt werden kann, sind 10 Mikrosekunden. Mit diesem Wert wird der 1ms-Sleep allerdings immer noch korrekt ausgeführt, er hat also keinen Einfluss auf unsere Ausführung.



```
<terminated> First Configuration [C/C++ QNX QConn (IP)] /tmp/First_ghm-lobermail155731689200033 on 192.168.8.2 pid 229402 (08
Current Clock Resolution: 100
New Clock Resolution: 10
Der 1ms-Sleep
Iterations: 5000
      START TIME 946687259.935470625
PLANNED STOP TIME 946687264.935480625
      REAL STOP TIME 946687264.935480625
```