Task 7

Fan Controller I

“What dreadful hot weather we have! It keeps me in a continual state of inelegance.”
– Jane Austen (1775-1817)

Embedded systems applications today have to control DC devices such as motors, fans, etc. In this task, you will use the virtual MCU developed in previous weeks to control the rotational speed of a DC fan.

7.1 Fan speed control

Develop a suitable interface circuit to allow your MCU hardware to (programatically) control the high-current, high-voltage to a DC fan. Be sure that your circuit is safe under all operating conditions for the fan and the MCU hardware. Control the speed of your fan via PWM from your virtual MCU. Provide a schematic for your interface circuit and justify all design decisions.

7.2 Tachometer sensor

Rotational speed, acceleration, and jerk of DC motors are unknown without feedback sensors. Your DC fan provides a tachometer feedback to the controlling hardware. Develop the appropriate interface circuitry to connect your DC motor tach sensor.

Be careful here! You can easily destroy the electronics of your DC fan and/or your MCU with the voltages and currents involved in this task.

Connect the DC fan to the input capture peripheral of your virtual peripheral so you can determine the rotational speed of your DC fan. Be sure that your tachometer readings are accurate and truly represent the rotational speed of the fan.
7.3 Tachometer verification

Design an independent method of measuring the fan’s rotational speed to verify the tach sensor output. Explain your method and include your circuit schematic.

7.4 Performance metrics

Develop a method by which you determine the following for your virtual MCU-MCU-fan setup:

- maximum rotational speed, \( R_{\text{max}} \), of your DC fan in RPM
- minimum rotational speed, \( R_{\text{min}} \), of your DC fan
- the rotational speeds at 10% increments of PWM: \( R_{10}, R_{20}, R_{30}, ..., R_{100} \)
- verify the twelve speeds above with your tach output and your independent method

7.5 Open-loop fan speed control

Write an virtual MCU application that does the following:

- During the program’s operation \( (t \geq 0) \), write the fan’s rotational speed (from the fan’s tachometer sensor) to a log file every second where \( r_k \) is the rotational speed (in RPM) at the \( k \)-th second
- Starting from a stopped fan \( (t = 0) \), drive the fan at \( R_{20} \) for \( 0 \leq t < 10 \text{ s} \)
- drive the fan at \( R_{50} \) for \( 10 \text{ s} \leq t < 20 \text{ s} \)
- drive the fan at \( R_{80} \) for \( 20 \text{ s} \leq t < 40 \text{ s} \)
- drive the fan at \( R_{\text{min}} \) for \( 40 \text{ s} \leq t < 60 \text{ s} \)
- drive the fan at \( R_{100} \) for \( 60 \text{ s} \leq t < 80 \text{ s} \)
- drive the fan at \( R_{30} \) for \( 80 \text{ s} \leq t < 100 \text{ s} \)
- drive the fan at \( R_{\text{max}} \) for \( 100 \text{ s} \leq t < 120 \text{ s} \)
- turn off the fan at \( t = 120 \text{ s} \) and continue to record the fan’s rotational speed every second until \( t = 180 \text{ s} \)

Using the sampled fan speeds in your log file, compute the following RMS error metric:

\[
E_{\text{RMS}} = \sqrt{\frac{\sum_{n=0}^{180} \left( \frac{r_k - R_k}{R_k} \right)^2}{180}}
\]

where \( r_k \) is the fan’s speed at \( t = k \) seconds and \( R_k \) is the fan’s target speed at \( t = k \).
7.6 Check Off

Demonstrate the following things to the TA:

1. Operation of your virtual MCU controlling the fan at the 10% intervals, $R_{\text{min}}$, and $R_{\text{max}}$.

2. Verification of your fan tachometer signal measurements via your independent method.

3. Fan speed control and measurement over the 180 s test cycle and results for $E_{\text{RMS}}$.

7.7 Submission

1. Commit your project to your BitBucket/Mercurial repository to your “trunk” folder in accordance with the procedure prescribed by the TA. Your BitBucket/Mercurial repository should contain all files required to produce your “build”, including the appropriate build file(s).

2. A “tag” release in your BitBucket/Mercurial repository called task7 that can be recalled at any time to build this task’s deliverables.

Each team should submit to their team repository the following files:\footnote{Files should be possess the same licensing header as the other ESOS source code files and be fully commented.}:

- The Python vMCU application task7_fan10.py

- The Python vMCU application task7_fan180.py

- Schematics of connection of the hardware to the fan for speed control and tachometer feedback

- Any unit test or test bench files used to verify your code created for this task
### CHECK-OFF

<table>
<thead>
<tr>
<th>% duty cycle</th>
<th>Speed (RPM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{\text{min}}$</td>
<td></td>
</tr>
<tr>
<td>$R_{10}$</td>
<td>10</td>
</tr>
<tr>
<td>$R_{20}$</td>
<td>20</td>
</tr>
<tr>
<td>$R_{30}$</td>
<td>30</td>
</tr>
<tr>
<td>$R_{40}$</td>
<td>40</td>
</tr>
<tr>
<td>$R_{50}$</td>
<td>50</td>
</tr>
<tr>
<td>$R_{60}$</td>
<td>60</td>
</tr>
<tr>
<td>$R_{70}$</td>
<td>70</td>
</tr>
<tr>
<td>$R_{80}$</td>
<td>80</td>
</tr>
<tr>
<td>$R_{90}$</td>
<td>90</td>
</tr>
<tr>
<td>$R_{100}$</td>
<td>100</td>
</tr>
<tr>
<td>$R_{\text{max}}$</td>
<td>$E_{RMS}$</td>
</tr>
</tbody>
</table>