lea.schoenberger [☺] tu-dortmund.de
mikail.yayla [☺] tu-dortmund.de

Exercises for the lecture
Cyber-Physical Systems Fundamentals
Summer Term 2018

Exercise Sheet 1

(5 Points)

Lab exercises starting from Monday, 28th May 2018

Hints: Please read the tutorial carefully and follow the instructions. If you have any questions, please contact your tutor.

Information about the lab computers

- In order to efficiently provide software required for various courses, the lab computers are equipped with several virtual machines. For the embedded systems lab exercises, we mostly use the Windows based VM *CPSF* (otherwise this will be indicated on the particular exercise sheet).
- Instead of the displayed Control + Alt + Del, press Control (right) + Del to log on.
- The provided account will be used for all virtual machines.
- The exercise sheets as well as required files can be found on the lab server under the path `\\pdc\cpsf`.
- The lab computers can access the university's network only. To browse the web, configure the Firefox settings as follows: Under Advanced → Network → Connections → Settings choose "Manual proxy configuration" and enter `proxy.cs.tu-dortmund.de` with port `3128` as "HTTP Proxy". Furthermore, check the box "Use this proxy server for all protocols".

1.1 Statechart Tutorial (5 Points)

1. Copy the zip file *tut_tpl.zip* under "\\pdc\cpsf\" to the desktop and extract it.
2. Change to the subdirectory *vs* and start *visualSTATE.vnw*.
3. Choose *Modeling* in the program window (Figure 1) or press F7.

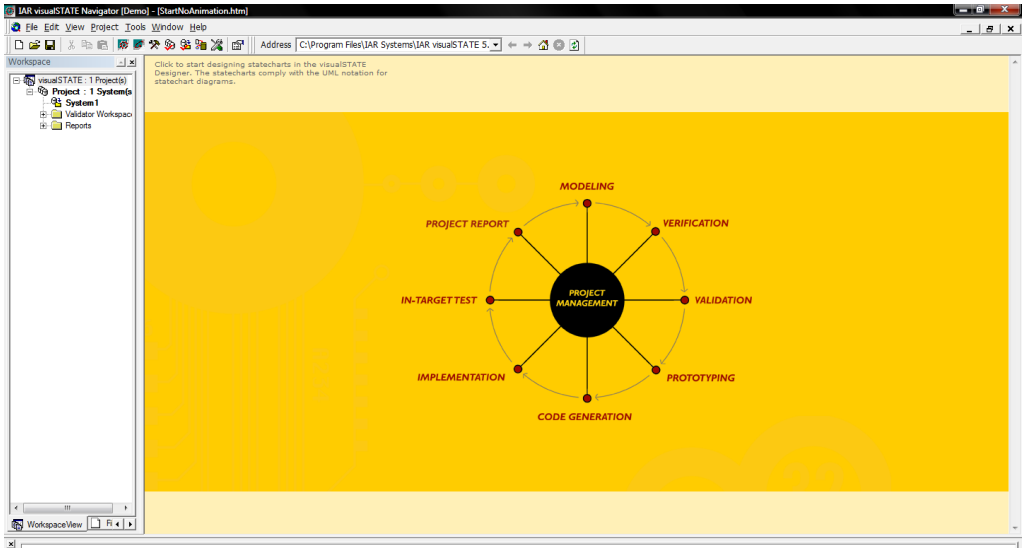

Figure 1: home screen

4. Now you can draw a diagram making use of the StateChart elements like states or transitions on the left side. To position these elements, please click on the the respective element and place it in the window using the left mouse button. To switch back to the pointer mode, press the right mouse button (cf. Figure 2).

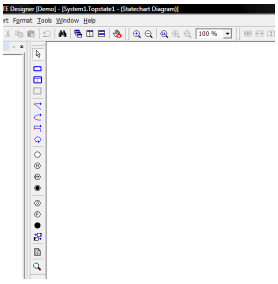

Figure 2: visualState Designer

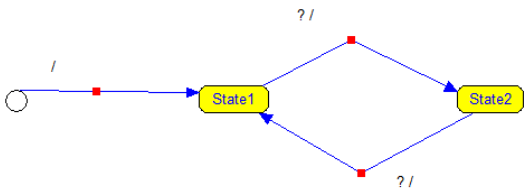

Figure 3: A simple StateChart example

5. Place the states and transitions as evident in Figure 3. Please note that it is necessary to define a start state.

6. To create a transition, double click on *?/*. Thereupon, a dialogue window opens (cf. Figure 4), where several properties can be modified. Double click on *Key0Press* to activate the transition at a push of the first button.

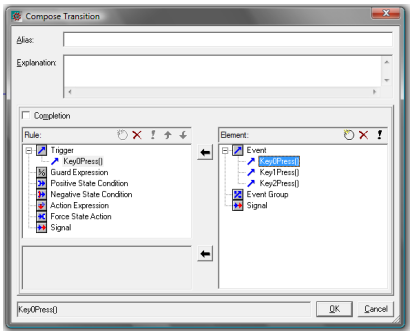

Figure 4: Transition dialogue window

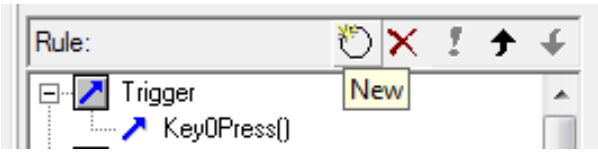

Figure 5: Create a new variable

7. To add a second action to the transition, click on *Action Expression* and subsequently on *writeLine0(VS_INT number):VS_VOID* within the dialogue window. By this means, you assigned the transition *writeLine0*, which displays a number in the first row of the LCD.

8. Then, a new variable shall be created. For this purpose, click on the *New* symbol in the *Rule* list (cf. Figure 5 while *Action Expression* is still selected.

9. Select the list entry *Internal Variable* in the right column and click on the *New* symbol. As a consequence, a new dialogue window appears where you can specify the variable's properties (cf. Figure 6).

10. Enter *zahl* as the variable's name and press *OK*.

11. To assign a meaning to the formula you defined under *Action Expression*, mark *X=?*, click on the text box in the left column and enter: *zahl = 1*.

12. Finally, the number should be displayed on the LCD. To make this possible, double click on *writeLine(?)*, choose *zahl* as a parameter in the dialogue popping up and close the dialogue window by clicking on *OK*. Hence, when the transition fires, the function *writeLine(zahl)* is called and therefore the respective number is displayed on the LCD.

13. Please complete the diagram as shown in Figure 7.

14. Save the file and close the current program window.

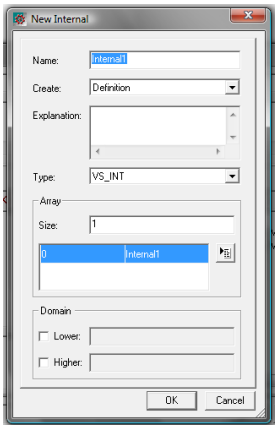

Figure 6: Create new internal variable

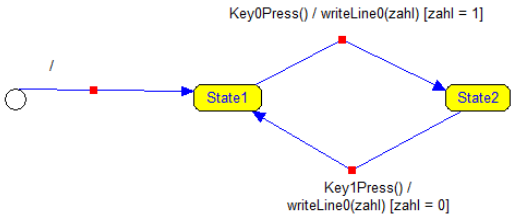

Figure 7: Complete StateChart example

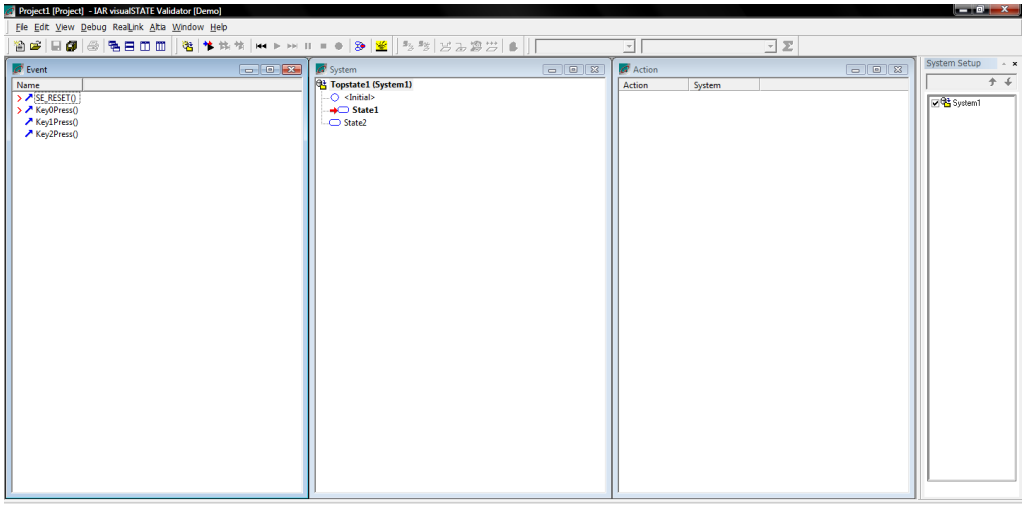

Figure 8: visualState validator

15. In the main application, click on *Validation* or, alternatively, press F8, to open the StateChart simulation mode (cf. Figure 8).

16. In the column *Event* you can find a list of you StateChart's events. By clicking on one of these events, it will be executed. The current states are highlighted in the column *System*, whereas the actions are listed in the last column.

17. If your system satisfies all requirements, close the validation window.

18. To automatically generate code from your StateChart, click on *Code Generation* in the main window (cf. Figure 1) or press F9. In the output window at the bottom of the screen, you can verify if your project has been translated properly.

19. If this is the case, start the program *IAR Embedded Workbench*, which can be found via the system search (click on the windows button and type the program name).

20. Choose *open existing workspace* and change to the project folder. In the subfolder *ew* a project *compiler* is provided.

21. Click on *Project* → *Make* or press F7 to compile the project.

22. Start the program *Flash Magic* that can be found via the system search.

23. Configure the properties according to Figure 9, whereat the user interface can differ depending on the version. LPC2103 may be found unter “ARM7”. Important settings: Board LPC2103, COM Port 1, Interface NXP ICP Bridge, Oscillator 16MHz.

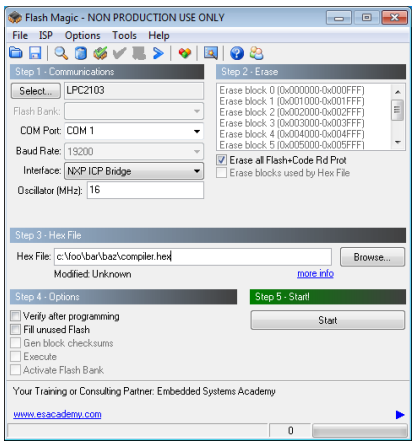

Figure 9: Flash Magic

24. Choose *compiler.hex* in the folder *Projekt/ew/Debug/Exe* as hex file. As soon as you click on *start*, your program will be transferred to the μController (cf. Figure 10).

Figure 10: Photo of a μController. Please simulate your StateChart before you test it on the hardware. Legend: 1. reset, 2. writeLine0(), 3. writeLine1(), 4. Key0Press, 5. Key1Press, 6. Key2Press

General information: An overview about the exercise sessions as well as further information can be found on `https://ls12-www.cs.tu-dortmund.de/daes/en/lehre/courses/sommersemester-2018/cyber-physical-system-fundamentals-ss-2018.html`. The exercise sheets will usually be published on the course website on Mondays and will be solved during the respective exercise sessions. The exercises are divided into two parts, in each of which at least 50% of the points must be achieved in order to receive the exam admission.