Safety-Critical Systems 4:
Engineering of Embedded Software Systems

AG Betriebssysteme, FB 3
Jan Bredereke

WiSe 2002/03

issued: Nov. 12, 2002
due: Nov. 28, 2002

Assignment 4

Bicycle Odometer – Software Implementation

We continue with our simple bicycle odometer from the last assignments, again. We now implement the controller software.

The implementation language is C. The basic real-time operating system provides us with the following header file. It contains all the variables and functions we can use. No standard header files or library functions may be used. (We have limited ROM space in the controller.)

```
/*
Safety-Critical Systems 4: Engineering of Embedded Software Systems
WiSe 2002/03
bicycle odometer: interface software - operating system and hardware
Jan Bredereke
*/

/* application functions called by the operating system */
void rtinit(void); /* called once at power-up; before rtmain() */
void rtmain(void); /* called exactly every 1 ms */

/* number of output display digits */
#define DIGIT_NUM 5

/* input and output variables, as described in the documentation */
unsigned char inByte; /* sensor and button input bits */
unsigned char modeOut; /* mode indicator output bits */
unsigned char numberOut[DIGIT_NUM]; /* number display digits */

/* Bit masks for the input variables: */
#define INBYTE_SENSOR ((unsigned char) 1)
#define INBYTE_BUTTON ((unsigned char) 2)
#define MODE_IND_KMH ((unsigned char) 1)
#define MODE_IND_KMTOTAL ((unsigned char) 2)
#define MODE_IND_KMTRIP ((unsigned char) 4)

/* value for blank digit */
#define D_BLANK ((unsigned char) 15)
```

This header file is also available for download on the lecture's Web site.

Implement the software requirements specification. In order to make the results of the solutions comparable, you should use as a base the software requirements specification on which we agree on Thursday Nov. 14 in the seminar. (It will also be published on the lecture's Web site.)

Additionally, write a software testing environment which allows to test your program on a standard desktop computer. Perform tests and document them.

Submit all relevant files of your solution by email to `brederek@tzi.de`.

Bonus assignment: implement the hardware and the basic operating system, too, then ask Prof. Jan Peleska for his RT-Tester tool, and perform real-time hardware-in-the-loop tests. `:-)`