

```

1  /*****
2  /* Hochschule fuer Technik und Wirtschaft
3  /* Fakultät fuer Ingenieurwissenschaften
4  /* Labor fuer Eingebettete Systeme
5  /* Mikroprozessortechnik
6  /*****
7  /*
8  /* C_Uebung.h: Header-Datei für die Implementierungsdatei
9  /*****
10
11 #ifndef C_UEBUNG_H
12 #define C_UEBUNG_H
13
14 #include <LPC21xx.H> // LPC21xx Definitionen
15
16 /* ===== Konstanten ===== */
17
18 // UART-Konstanten
19 #define SEGMENT 0x1FC0000 // BCD-Segmentdefinition
20 #define PCLOCK 12500000 // Peripherie-Clock in Hz
21 #define BAUDRATE 19200 // Baudrate der seriellen Schnittstelle
22 #define DLAB_BIT 128 // DLAB-Bit setzen (7. Bit)
23 #define UART_PINSEL_CONFIG 0x50000 // P0.8 = TxD1, P0.9 = RxD1
24 #define UART_READY_BIT 0x20 // Ready-Bit im UART
25 #define UART_FIFO_ENABLE 0x07 // FIFO aktivieren und zurücksetzen
26 #define UART_RX_READY 0x01 // RX-Ready-Bit
27 #define HEX_DIGIT_LIMIT 8 // Maximale Länge der Hex-Eingabe
28 #define CR_LF "\r\n" // Carriage Return + Line Feed
29 #define INVALID_HEX_VALUE 0xFFFFFFFF // Ungültiger Hex-Wert
30
31 /* ===== Globale Variablen ===== */
32
33 unsigned int sek = 0; // Sekunden-Zähler
34
35 unsigned long bcd[10] = {
36     0x0FC0000, 0x180000, 0x16C0000, 0x13C0000, 0x1980000,
37     0x1B40000, 0x1F40000, 0x1C0000, 0x1FC0000, 0x1BC0000
38 };
39
40 static const unsigned int baudrates[] = {
41     110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600
42 };
43
44 /* ===== Funktionsprototypen ===== */
45
46 // GPIO- und Schalterzustands-Funktionen
47 unsigned int readInputBCD(void); // BCD-Eingang lesen
48 unsigned int readSwitchState1(void); // Zustand von Switch 1 lesen
49 unsigned int readSwitchState2(void); // Zustand von Switch 2 lesen
50 unsigned int readSwitchState3(void); // Zustand von Switch 3 lesen
51
52 // Initialisierungsfunktionen
53 unsigned int initBaudrate(void); // Baudrate initialisieren
54 void initTimer(void); // Timer initialisieren
55 void initExIn(void); // Externen Interrupt initialisieren
56 void initSeg(void); // 7-Segment Initialisieren
57
58 // Interrupt-Handler
59 void myEXTINT(void) __irq; // Externer Interrupt-Handler
60 void T0isr(void) __irq; // Timer-Interrupt-Handler
61
62 // UART-Funktionen
63 void uartInit(unsigned int baudRate, unsigned int dataBits, unsigned int stopBits, unsigned int
paritySelect, unsigned int parityEnable); // UART initialisieren
64 void uartSendChar(char data); // Zeichen über UART senden
65 void uartSendString(char* str); // String über UART senden
66 char uartReadChar(void); // Zeichen über UART empfangen
67
68 #endif /* C_UEBUNG_H */
69

```