

```

1  /*=====*/
/* Projekt      :   Olli - eine Holzkiste lernt laufen      */
/*                                                     */
/* Hardware      :   Infineon C509                        */
5 /*                                                     */
/* Dateiname     :   move_order.c                        */
/*                                                     */
/* Version       :   1.1 vom 06. Dezember 2003           */
/*                                                     */
10 /* Autoren      :   L. Kulf, J. Roos                    */
/*                                                     */
/* Datei-        :   Fahrbefehlsmanager für Automatikbetrieb */
/* beschreibung  :   Fahrbefehlsmanager für Automatikbetrieb */
15 /*                                                     */
/*                                                     */
/*=====*/

20 // -----
// Dateien einbinden
#include "move_order.h"
#include "main.h"
#include "seriell_olli_seruell.h"

25 // -----
// Variablendeklaration, globale Variablen
unsigned char move_order_pos = 0; // anzufahrende Position
extern _bit m_new_move_order; // Merker, neuer Fahrbefehl angekommen
30 extern unsigned char befehl_s0_in[5]; // in arraystelle [2]+[3] steht Positionsnummer

// -----
void check_move_order(void)
35 {
    // Überprüfen, ob neuer Fahrbefehl vorhanden ist
    if(m_new_move_order == true)
    {
        // Fahrbefehle auslesen und abspeichern

40        // Fahrbefehl für mechanische Position 1 auslesen
        if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 1))
            ||
            ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 1))
45            ||
            ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 1))
            ||
            ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 1)))
        {
50            move_order_pos=1;
        }

        // Fahrbefehl für mechanische Position 2 auslesen
55        if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 2))
            ||
            ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 2))
            ||
            ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 2))
60            ||
            ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 2)))
        {
            move_order_pos=2;
65        }

        // Fahrbefehl für mechanische Position 3 auslesen
        if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 3))
            ||
70            ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 3))
            ||
            ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 3))
            ||
            ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 3)))
75        {
            move_order_pos=3;
        }

        // Fahrbefehl für mechanische Position 4 auslesen
80        if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 4))
            ||
            ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 4))
            ||
85            ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 4))
            ||
            ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 4)))
        {

```

```

90         move_order_pos=4;
        }

    // Fahrbefehl für mechanische Position 5 auslesen
    if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 5))
        ||
        ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 5))
        ||
        ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 5))
        ||
100    ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 5)))

    {
        move_order_pos=5;
    }

105    // Fahrbefehl für mechanische Position 6 (Grundstellung) auslesen
    if (((befehl_s0_in[1] == 0) && (befehl_s0_in[2] == 6))
        ||
        ((befehl_s0_in[1] == 1) && (befehl_s0_in[2] == 6))
110    ||
        ((befehl_s0_in[1] == 2) && (befehl_s0_in[2] == 6))
        ||
        ((befehl_s0_in[1] == 3) && (befehl_s0_in[2] == 6)))

115    {
        move_order_pos=6;
    }
}

120 // -----
    // Dateiende

```