```c
/*---------------------/*-------------------------------------------------------------------------------
 Main Program

 Eisenbahnsteuerung made by: Andreas Lutters / Achim Walder

 https://wolles-elektronikkiste.de/atmel-studio-7-ein-einstieg

-----------------------------------------------------------------------------*/

#include <avr/io.h>
#include "config.h"
#include "usart.h"
#include "timer.h"
#include <util/delay.h>
#include "cmd.h"
#include <avr/eeprom.h>
// A.Fischer ==> sei() kann in main.c durch Hinzufügen der folgenden Header-Datei deklariert werden:
#include <util/atomic.h>

//$ simulavr -g -d atmega644p

//-----------------------------------------------------------------------------
//Hier startet das Hauptprogramm
int main(void)
	{
	unsigned int	USB_Counter;
	unsigned char	Fahrstufe_Nr;
	unsigned int	spalte;
	unsigned long	ProgrammStartDelay;		// ProgrammStartDelay - Verzögerung für Programmstart
	unsigned char	M_Block_Nr;		// Main-Block-Nummer [1..32] [Datentype Byte]
	unsigned int	i1;		// lokale Zähler
	unsigned int	Sig_Nr;		// lokale Zähler für Signale
//	unsigned TestLed;		// TestLed für Steuerregister
	
	//#################################################################################################################
	// intitialisierung von Variablen
	//#################################################################################################################

	MC_Responce_Modus = 1;		// MC-RückmeldungModus = 0 keine Rückmeldung
	MC_Betrieb_Modus = 0;		// MC-BetriebsModus = 0
	MC_Test_Modus = 0;		// MC-TestModus = 0
	Bahn_Automatik_Modus = 0;		// MC-AutomatikModus = 0
	Bahn_Automatik_HS_Modus = 0;		// MC-AutomatikModus = 0
	Bahn_Automatik_SB_Modus = 0;		// MC-AutomatikModus = 0
	Bahn_Automatik_NS_Modus = 0;		// MC-AutomatikModus = 0
	Bahn_Automatik_GB_Modus = 0;		// MC-AutomatikModus = 0
	Bahn_Automatik_St_Modus = 0;		// MC-AutomatikModus = 0
	
	Peek = 0;		// Peek-Zähler für Timer
	Stufe = 0;
	zeile = 0;
	spalte = 0;
	
	BlockKarte = 0;		// BlockKarten-Zähler
	BK_Karten_Offset = 35;		// Start für Zähler
	Fahrstufen_MAX = 80;		// maximale Fahrstufe

	delay_groesse = 5;		//Achtung schnelle Variante!
	Delay_BS = 200;		// Delay_BS = Wartepause Default Wert für BST Routine
	
	DDRA = OUTA;		// Konfiguration der
```

```
Ausgänge bzw. Eingänge definition erfolgt in der config.h
    DDRC = OUTC;                                    // Konfiguration der Ausgänge bzw. Eingänge definition erfolgt in der config.h
    DDRD = OUTD;                                    // Konfiguration der Ausgänge bzw. Eingänge definition erfolgt in der config.h
    
    Main_Counter = 0;                               // Fahrstufen-Delay-Counter = 0

    USB_Counter = 0;                                // USB-Counter = 0

    for( i1 = 0; i1 < Fahrstufen_MAX; i1++)         // Initialisiere alle FahrStufen-Array = "0"
    {
    FahrStufe[i1][0] = 0;
    FahrStufe[i1][1] = 0;
    FahrStufe[i1][2] = 0;
    FahrStufe[i1][3] = 0;
    }
    
    for( M_Block_Nr = 0; M_Block_Nr <= 32; M_Block_Nr++)    // Initialisiere alle Block-Werte = "0"
    {
    Block_FS_max_Register[M_Block_Nr]       = 0;
    Block_FS_min_Register[M_Block_Nr]       = 0;
    Block_FS_soll_Register[M_Block_Nr]      = 0;
    Block_FS_Delay1_Register[M_Block_Nr]    = 0;
    Block_FS_Delay2_Register[M_Block_Nr]    = 0;
    Block_Delay_Register[M_Block_Nr]        = 0;
    }
    
    
    for( Sig_Nr = 0; Sig_Nr < 280; Sig_Nr++)
    {
    Signal[Sig_Nr] = 0;
    }
    
    usart_init(BAUDRATE);                           // setup the UART
    
    
    //#################################################################################################################################
    // Programmstart
    
    //#################################################################################################################################
    //Timer starten
    timer_init();
    
    for(ProgrammStartDelay = 0; ProgrammStartDelay < 1000000; ProgrammStartDelay++)
    {asm("nop");};
    
    sei();                                          // Globale Interrupts einschalten
    timer_speed = Normale_Geschwindigkeit;          // Achtung war 30000!!! bm 4

    set_MC_LED();                                   // Test für MC_LED_Test + MC_INIT

    
    //#################################################################################################################################
    // Hauptschleife
    
    //#################################################################################################################################
    while(1)
    {
    
        //-------------------------------------------------------------------------------------------------------------------------------
    // USB Schnittstelle überwachen
    
```

```
        //-------------------------------------------------------------------------------------------
117     if (usart_status.usart_ready)
118     {
119     if(extract_cmd(&usart_rx_buffer[0]))
120     {
121     asm("nop");
122     //if (MC_Responce_Modus == 1)
123     //usart_write("  ==> USB ok !\r\n");
124     }
125     else
126     {
127     usart_write("\r\n ==> [MAIN.C] USB-Error: %s\r\n", &usart_rx_buffer[0]);
128     }
129     usart_status.usart_ready = 0;
130     }
131     
132     
        //#########################################################################################
133     //
134     // Betriebsmodus 4 ==> Impulsbreitensteuerung
135     //
136     
        //#########################################################################################
137     if (MC_Betrieb_Modus == 4)
138     {
139     if (Main_Counter > 30 )// Zähler für Geschwindigkeitsänderungen
140     {
141     Main_Counter = 0;
142     };
143     Main_Counter = Main_Counter + 1;
144
145     
        //#########################################################################################
146     // Geschwindigkeit in allen Blocks verändern
147     
        //#########################################################################################
148     if (Main_Counter == 1)
149     {
150     for (M_Block_Nr = 1; M_Block_Nr <= 32; M_Block_Nr++)// für alle 32 Blockabschnitte
151     {
152
153     // Bl-Aktiv gleich "0" ==> alle Werte auf "0"
154     if ((Block_Status_Register[M_Block_Nr] == 0) || (Block_Aktiv_Register[M_Block_Nr] == 0))
155     {
156     Timer_Block_FS_Delay_Register[M_Block_Nr] = 0;// mit 0 laden
157     Timer_Block_FS_ist_Register[M_Block_Nr] = 0;// FS auf "0"
158     }
159
160     // Bl-Aktiv ungleich "0"
161     if ((Block_Status_Register[M_Block_Nr] == 1) && (Timer_Block_FS_Delay_Register[M_Block_Nr] == 0))// nur wenn Block = "GRÜN" und Timer = "0"
162     {
163     // Block - beschleunigen ----------------------------------
```

```
164 ————————————————————————————————if (Block_Aktiv_Register[M_Block_Nr] ==
                                    1) ————————————————————————————————————————————————————————————————————————————————————————————————————————//
                                    nur wenn Aktiv = "1"
165 ————————————————————————————————————————————————————————————————————————————
```

```
200         if (Main_Counter == 10)
201         {
202         for (M_Block_Nr = 1; M_Block_Nr <= 32;
                M_Block_Nr++)                                                   // Block-Nr [1..32] - Regelung
203         {
204         // Beschleunigen ----------------
205         for ( Fahrstufe_Nr = 1; Fahrstufe_Nr <
                Timer_Block_FS_ist_Register[M_Block_Nr] + 1; Fahrstufe_Nr++ )   // Fahrstufe +1
206         {
207         if ((M_Block_Nr >  0) && (M_Block_Nr < 9))
208         {
209         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][0] |=
                _BV(M_Block_Nr -  1);
210         };
211         if ((M_Block_Nr >  8) && (M_Block_Nr < 17))
212         {
213         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][1] |=
                _BV(M_Block_Nr -  9);
214         };
215         if ((M_Block_Nr > 16) && (M_Block_Nr < 25))
216         {
217         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][2] |=
                _BV(M_Block_Nr - 17);
218         };
219         if ((M_Block_Nr > 24) && (M_Block_Nr < 33))
220         {
221         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][3] |=
                _BV(M_Block_Nr - 25);
222         };
223         }
224         }
225         };
226
227         
                //#######################################################################
                ##########################################################################
                ######
228         // bremsen --- neue Geschwindigkeit in Fahrstufen_Array übertragen
229         
                //#######################################################################
                ##########################################################################
                ######
230         if (Main_Counter == 20)
231         {
232         for (M_Block_Nr = 1; M_Block_Nr <= 32;
                M_Block_Nr++)                                                   // Block-Nr [1..32] - Regelung
233         {
234         // Bremsen ----------------
235         for ( Fahrstufe_Nr = Timer_Block_FS_ist_Register[M_Block_Nr] +
                1; Fahrstufe_Nr < Fahrstufen_MAX; Fahrstufe_Nr++ )   // Fahrstufe
                -1
236         {
237         if ((M_Block_Nr >  0) && (M_Block_Nr < 9))
238         {
239         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][0] &=
                255-_BV(M_Block_Nr -  1);
240         };
241         if ((M_Block_Nr >  8) && (M_Block_Nr < 17))
242         {
243         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][1] &=
                255-_BV(M_Block_Nr -  9);
244         };
245         if ((M_Block_Nr > 16) && (M_Block_Nr < 25))
246         {
247         FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][2] &=
                255-_BV(M_Block_Nr - 17);
248         };
249         if ((M_Block_Nr > 24) && (M_Block_Nr < 33))
```

```
250     ————————————————————{
251     ————————————————————FahrStufe[Fahrstufen_MAX - Fahrstufe_Nr][3] &=
                             255-_BV(M_Block_Nr - 25);
252     ————————————————————};
253     ————————————————————}————
254     ————————————————}————
255     ————————————}
256     ————————}————
257
258     ————————}————————// Ende while (1)
259
260     ————return(0);
261     ————}
```