

Motoransteuerung 8051

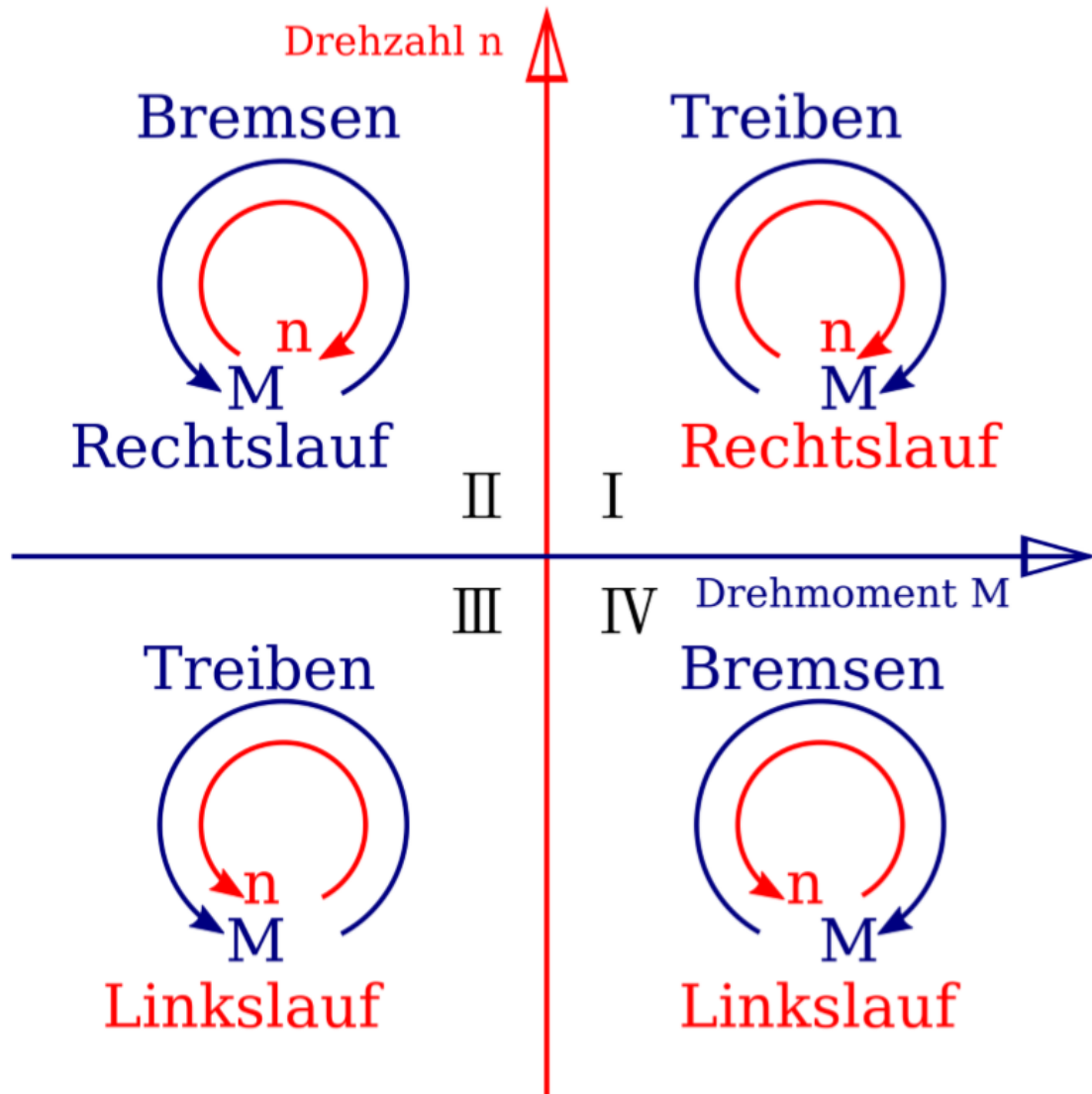
Lukas Heiligenbrunner

Rene Fischer



Aufgabenstellung:

Programmieren einer Motorsteuerung für den 8051 in C.



Funktionen:

- Links drehen 100%
- Rechts drehen 100%
- Auslaufen lassen
- Hart stoppen
- Drehzahl steuern (0-100%)

Die Grundsätzliche Ansteuerung der Motoren war gegeben, nur das tatsächliche Eingabe in die Serielle Schnittstelle und die PWM Steuerung ist zu Programmieren.

Durchführung:

1. Programmierung und Testung der Seriellen Schnittstelle.
 - a. Hinzufügen der Timer0 InterruptServiceRoutine
 - b. Testen des Empfangens einzelner Buchstaben
2. Hinzufügen einer Detektion welcher Buchstabe eingetippt wurde
 - a. Bei r → einschalten der Motorstufe; links auf 0 und rechts auf 1
 - b. Bei l → einschalten der Motorstufe; rechts auf 0 und links auf 1
 - c. Bei s → hartes stoppen der Motorstufe: Motorstufe auf 1; rechts auf 0; links auf 0
 - d. Bei p → stoppen der Motorstufe: Motorstufe auf 0; rechts auf 0; links auf 0
3. Einfügen eines Menüs, in dem man den Duty Cycle des PWMs bestimmen kann (drücken des Buchstaben d)
 - a. Einlesen von 3 Zeichen und anschließende Umwandlung in einen Integer
 - b. Außer folgende Zeichen:
 - i. C → schließen des Menüs und zurück zum Hauptmenu
 - ii. A → wechseln des Modus in Automatik (PWM Steuerung)
 - iii. R → rechtslauf in der PWM Steuerung
 - iv. L → linkslauf in der PWM Steuerung
 - c. In Timer Interrupt Service Routine bis hundert zählen und vom Wert 0 – Duty Cycle Motor einschalten und von Duty Cycle – 100 Ausschalten → dadurch entsteht ein PWM Signal
4. Programmieren von Funktionen zum Ausgeben von char bzw. Strings auf die Serielle Schnittstelle

Aussehen der Ausgabe:

```
[l] --> links laufen 100%
[r] --> rechts laufen 100%
[p] --> auslaufen lassen
[s] --> hart stoppen
[d] --> duty cycle optionen
You are in speed setup
[c]--> Exit
[a] --> enable automatic mode
[r] --> rotate right
[l] -->rotate left
Please enter speed:
aset to auto mode...
050You selected: 05cexiting from menu
[l] --> links laufen 100%
[r] --> rechts laufen 100%
[p] --> auslaufen lassen
[s] --> hart stoppen
[d] --> duty cycle optionenMotor auslaufen lassen
```

Aufgetretene Fehler und Debugging:

- Enter Erkennung im Simulator anders als auf der tatsächlichen Hardware mit Toolstick Terminal
- While Schleife hat nicht funktioniert, Argumentsvariablen sollten nicht verändert werden.

Code:

Main.c

```
//-----  
-----  
  
// Main.c  
  
//-----  
-----  
  
// HTBLA Steyr - Elektronik und Technische Informatik  
//  
// Author: Felix Hutsteiner  
// Version: 1.0  
//  
// Changes:  
// 5.2.2014 created  
//  
//  
// Target: C8051F330 Silabs  
// Tool chain: Keil µVision 4  
//  
// Program Description:  
//  
// This program is a template for c51 main modules  
//  
  
//-----  
-----  
  
// Includes  
  
//-----  
-----  
  
#include <C8051F330.h> // SFR declaration  
#include <stdio.h>  
  
//-----  
-----  
  
// Global Constants  
  
//-----  
-----  
  
//-----  
-----
```

```

// Global Variables
//-----

sbit motor_r = P0^0; // 1: Motoranschluss für Rechtslauf an +UB, 0: ...
Masse

sbit motor_l = P0^1; // 1: Motoranschluss für Linkslauf an +UB, 0: ...
Masse

sbit fb_inh = P1^2; // 1: Motorbrücke aktiviert, 0: Motorbrücke gesperrt

// Achtung: für Rechtslauf muss motor_l auf 0 sein, und umgekehrt
!!!!!!!!!

//-----

// Function Prototypes
//-----

extern void init_device();
extern void writeSerial(char out);
extern void string_Print(char mystring[]);

//-----

// main() Routine
//-----

void main (void) small {

PCA0MD &= 0xbf; // Watchdogtimer abschalten

init_device(); // Init laut Config-Tool

EA=1; //alle interrupts aktivieren
ET0=1; //enable Timer 0
ES0=1;
TR0=1; //TimerRunflag setzen um den Timer zu starten
TI0=1; //Timer Interrupt aktivieren

string_Print("[l] --> links laufen 100%\n[r] --> rechts laufen 100% \n[p]
--> auslaufen lassen\n[s] --> hart stoppen\n[d] --> duty cicle optionen");

while (1) {

```

```

// main loop code
}
}

//-----
// End Of File
//-----

```

SUB.c

```

//-----
// template.c
//-----

// HTBLA Steyr - Elektronik und Technische Informatik
//
// Author: Felix Hutsteiner
// Version: 1.0
//
// Changes:
// 5.2.2014 created
//
//
// Target: C8051F330 Silabs
// Tool chain: Keil µVision 4
//
// Program Description:
// This program is a template for c51 subfunction modules
//

//-----
// Includes
//-----

#include <C8051F330.h> // SFR declaration
#include <stdio.h>

```

```

//-----
// Global Constants
//-----

#define PWM_period 100

//-----
// Global Variables
//-----

sbit motor_r = P0^0; // 1: Motoranschluss für Rechtslauf an +UB, 0: ...
Masse

sbit motor_l = P0^1; // 1: Motoranschluss für Linkslauf an +UB, 0: ...
Masse

sbit fb_inh = P1^2; // 1: Motorbrücke aktiviert, 0: Motorbrücke gesperrt

char uart_in;

char percentchar[3];

int setmenu = 0;
int currpercent = 0;
int temp = 0;

int tempint = 0;

int manual = 1;
int direction = 0; // 0 --> right 1 --> left

// Achtung: für Rechtslauf muss motor_l auf 0 sein, und umgekehrt
!!!!!!!!!!

//-----
// Function Prototypes
//-----

void writeSerial(char out);
void T0_interrupt (void);
void serial_interrupt (void);

```

```

void string_Print(char mystring[]);
void writeInt(int out);

//-----
// Functions
//-----

void writeSerial (char out){
while(!TI0);
SBUF0=out;
TI0=0;
}

void writeInt(int out){
tempint = out;
if(tempint == 0)
{
writeSerial(0+ '0');
}
while(tempint > 0){
writeSerial((tempint%10) + '0');
tempint = tempint/10;
}
}

void string_Print(char mystring[])
{
int i =0;
for(i=0;mystring[i] != 0 ;i++)
{
writeSerial(mystring[i]);
}
}

void T0_interrupt (void) interrupt 1 using 1 {
if(manual == 0)
{
if(temp <= currpercent && temp != currpercent!=0)
{

```



```

if(!direction)
{
fb_inh = 1;
motor_l = 0;
motor_r = 1;
}else{
fb_inh = 1;
motor_r = 0;
motor_l = 1;
}
}else{
fb_inh = 0;
motor_r = 0;
motor_l = 0;
}
}

if(temp >= 100)
{
temp=0;
}else{
temp++;
}
}

void serial_interrupt (void) interrupt 4 using 1 {
if(RI0){
uart_in = SBUF0;
RI0 = 0;
//writeSerial(uart_in);
if(setmenu == 1)
{
writeSerial(uart_in);

if(uart_in == 'c')
{
string_Print("exiting from menu\n");
string_Print("[l] --> links laufen 100%\n[r] --> rechts laufen 100% \n[p]
--> auslaufen lassen\n[s] --> hart stoppen\n[d] --> duty cicle optionen");

```

```

setmenu = 0;
}else if(uart_in == 'a'){
string_Print("set to auto mode...\n");
manual = 0;
}else if(uart_in == 'r'){
string_Print("running right...\n");
direction = 0;
}else if(uart_in == 'l'){
string_Print("running left...\n");
direction = 1;
}else{
if(percentchar[0] == 0)
{
percentchar[0] = uart_in;
} else if(percentchar[1] == 0){
percentchar[1] = uart_in;
} else{
percentchar[2] = uart_in;
currpercent = (percentchar[0]-'0')*100+(percentchar[1]-
'0')*10+(percentchar[2]-'0');
percentchar[0]=0;
percentchar[1]=0;
percentchar[2]=0;

if(currpercent > 100)
{
currpercent = 100;
}

string_Print("You selected: ");
writeInt(currpercent);

}
}
}else{
if(uart_in == 'l')
{
string_Print("links laufen\n");
manual =1;

```

```

fb_inh = 1;
motor_r = 0;
motor_l = 1;

}else if(uart_in == 'r')
{
string_Print("rechts laufen\n");
manual =1;

fb_inh = 1;
motor_l = 0;
motor_r = 1;
}else if(uart_in == 's')
{
string_Print("STOP Motor\n");
manual =1;

fb_inh = 1;
motor_r = 0;
motor_l = 0;
}else if(uart_in == 'p')
{
string_Print("Motor auslaufen lassen\n");
manual =1;

fb_inh = 0;
motor_r = 0;
motor_l = 0;
}else if(uart_in == 'd')
{
string_Print("\nYou are in speed setup \n[c]--> Exit \n[a] --> enable
automatic mode \n[r] --> rotate right\n[l] -->rotate left\nPlease enter
speed:\n");
setmenu = 1;
}
}

}
}

```

```
//-----  
-----  
  
// End Of File  
//-----  
-----
```