AVR-Libc

Übersicht

Hochschule Offenburg
University of Applied Sciences

- AVR-Libc
- Datentypen
- Dynamische Speicherbelegung
- Stringwandlung
- Register Definitionen
- Warteschleifen
- Daten im Programmspeicher
- EEPROM

AVR-Libc

- Standard C Bibliothek für AVR
- Register Definitionen
- Routinen für die Verwendung bestimmter µC Funktionen
- Hilfsfunktionen
- http://www.nongnu.org/avr-libc/

Datentypen

- Integer mit definierter Größe
  - int8_t, uint8_t ... uint64_t
- Werden im Header stdint.h definiert
  - typedef unsigned char uint8_t

Dynamische Speicherbelegung

Hochschule Offenburg
University of Applied Sciences

- Heap befindet sich im SRAM
- Funktionen: malloc, calloc, realloc und free
- Header: stdlib.h
- Beispiel:

```
#include <stdlib.h>

int main(void)
{
   uint8_t* a = malloc(10*sizeof(uint8_t));
   ...
   free (a);
   return 0;
}
```

Stringwandlung

- char * itoa (int __val, char *__s, int __radix)
- char * dtostrf (double __val, signed char __width,
  unsigned char __prec, char *__s)
- Header: stdlib.h
- Beispiel

```
#include <stdlib.h>

   uint8_t a = 27;
   char buffer[4];

   itoa( a, buffer, 10)
```

Register Definitionen

- Definitionen für Register und Bits
  - #define PORTA _SFR_IO8(0x1B),
  - #define PA4 4
- Makro _BV: #define _BV(bit) (1 << (bit))
- Header: avr/io.h
- Beispiel

```
#include <stdlib.h>
#include <avr/io.h>
...
//Bit 4 von Port A setzen
PORTA |= _BV(PA4);
//Bit 4 und 5 löschen
PORTA &= ~(_BV(PA4) | _BV(PA5));
...
```

Warteschleifen

- void _delay_ms (double __ms)
  - max. 262.14 ms / F_CPU in MHz
- void _delay_us (double __us)
  - 768 us / F_CPU in MHz
- Header: util/delay.h
- Taktfrequenz muss bekannt sein

```
#define F_CPU 16000000UL
#include <util/delay.h>
...
_delay_ms(10); // 10ms warten
...
```

Daten im Programmspeicher

Hochschule Offenburg
University of Applied Sciences

- Header: avr/pgmspace.h
- Funktionen:
  - pgm_read_byte(address_short)
  - void * memcpy_P (void * dest, PGM_VOID_P src, size_t n)
  - ...
- Mit dem Attribut PROGMEM können Globale Variablen im Programmspeicher gesichert werden.

Daten im Programmspeicher (2)

Hochschule Offenburg
University of Applied Sciences

- Beispiel

```
#include <avr/pgmspace.h>

//Daten werden im SRAM gespeichert
uint8_t data1[] = {0,1,2,3,4};

//Daten werden im Flash gespeichert
uint8_t data2[] PROGMEM = {5,6,7,8,9};



int main(void)
{
    ...
    uint8_t a= data1[2];
    uint8_t a= pgm_read_byte(&(data2[2]));
    ...
}
```

EEPROM

Hochschule Offenburg
University of Applied Sciences

- Header: avr/eeprom.h
- Funktionen
  - uint8_t eeprom_read_byte (const uint8_t *addr)
  - void eeprom_write_byte (uint8_t *addr, uint8_t value)
- Attribut: EEMEM

```
#include <avr/eeprom.h>

uint8_t data3[] EEMEM ={ 10, 11, 12, 13, 14};
...
if(eeprom_is_ready())
   int8_t = eeprom_read_byte(data3+2);
```