

HemOS Reference Manual

0.1

Generated by Doxygen 1.3.3

Thu Sep 25 10:18:05 2003

Contents

1	HemOS File Index	1
2	HemOS File Documentation	1

1 HemOS File Index

1.1 HemOS File List

Here is a list of all documented files with brief descriptions:

hemavr.c (HemAVR module software)	1
hemisson.h (HemAVR-to-Hemisson communication over serial link)	2
i2c.h (TWI interface)	4
uart0.h	??
uart1.h	??

2 HemOS File Documentation

2.1 hemavr.c File Reference

HemAVR module software.

```
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <avr/interrupt.h>
#include <avr/signal.h>
#include <stdio.h>
#include "uart0.h"
#include "hemavr.h"
```

Functions

- void hemavr_init (void)

Variables

- `const char s_startup[]` **PROGMEM** = "HemOS 0.1 – HemAVR REV A
CPU Module\r\n"

2.1.1 Detailed Description

HemAVR module software.

After initialization, this module will send a boot-message to UART0 as a sign of life.

Version:

0.1

Author:

Ingo Frank <if@elektronengehirn.net>

Note:

Look at the file *COPYRIGHT* for redistribution information—*ROBOTICS* for implementation issues.

2.1.2 Function Documentation

2.1.2.1 void hemavr_init (void)

Initialization of HemAVR module.

clocked internally by the prescaler => $14745600 / 256 = 57600$ Hz 256 ticks in
 $256 / 57600$ Hz = 4.44 ms => 57 (OCR2) 57600 Hz / 256 = 222.65625 Hz =
4.44 ms => $256 / 4.44 = 57$ (OCR2)

- Use UART0 as stdin resp. stdout/stderr for communication with PC.
- Initialize TWI (I2C) master mode.
- Print a startup message to stdout.

initialize UART0 (link to PC)

initialize stdin, stdout, stderr

initialize timer2 – used for multithreading scheduler

initialize i2c busmaster

2.2 hemisson.h File Reference

HemAVR-to-Hemisson communication over serial link.

Defines

- #define **XTAL** 14745600
Frequency of the quartz-oscillator on the HemAVR board.
- #define **BAUDRATE1** 115200
Baudrate of serial link to the robot.
- #define **BAUD1** XTAL / (16 * BAUDRATE1) - 1
- #define **FRONT** 2
- #define **FRONTLEFT** 1
- #define **FRONTRIGHT** 0
- #define **LEFT** 4
- #define **RIGHT** 5
- #define **REAR** 3
- #define **GROUNDLEFT** 6
- #define **GROUNDRIGHT** 7
- #define **FRONTZONE** 0
- #define **REARZONE** 1
- #define **GROUNDZONE** 2

Functions

- void **hemisson_init** (void)
Initialize i/o over UART1.
- unsigned char **hemisson_get_brightness** (unsigned char sensor)
Get value for brightness from infrared sensor.
- unsigned char **hemisson_get_proximity** (unsigned char sensor)
Get value for proximity from infrared sensor.
- void **hemisson_set_speed** (int left, int right)
Set values for differential drive.

2.2.1 Detailed Description

HemAVR-to-Hemisson communication over serial link.

Version:

0.1

Author:

Ingo Frank <if@elektronengehirn.net>

Note:

Look at the file *COPYRIGHT* for redistribution information—*ROBOTICS* for implementation issues.

2.2.2 Function Documentation

2.2.2.1 unsigned char hemisson_get_brightness (unsigned char *sensor*)

Get value for brightness from infrared sensor.

Parameters:

sensor The number of infrared sensor.

Returns:

A value 0–255 (very bright – very dark).

2.2.2.2 unsigned char hemisson_get_proximity (unsigned char *sensor*)

Get value for proximity from infrared sensor.

Parameters:

sensor The number of infrared sensor.

Returns:

A value 0–255 (very far – very close).

2.2.2.3 void hemisson_set_speed (int *left*, int *right*)

Set values for differential drive.

Speed can be set from -9 to 9 for both motors:

- 0,0 will stop the robot,
- -1,1 will start a slow left-rotation and
- 9,9 will move it fast forward.

Parameters:

left speed for left motor.

right speed for right motor.

2.3 i2c.h File Reference

TWI interface.

Defines

- #define **I2C_TIMEOUT** 1000
- #define **I2C_ACK** 1
- #define **I2C_NACK** 0

Functions

- void **i2c_start** (void)
- int **i2c_read** (unsigned char *ack*)
- void **i2c_write** (int *data*)
- void **i2c_stop** (void)

2.3.1 Detailed Description

TWI interface.

Version:

0.1

Author:

Ingo Frank <if@elektronengehirn.net>

Note:

Look at the file *COPYRIGHT* for redistribution information—*ROBOTICS* for implementation issues.

2.3.2 Function Documentation

2.3.2.1 int i2c_read (unsigned char *ack*)

Read a byte.

Parameters:

ack Acknowledge.

Returns:

Received byte.

2.3.2.2 void i2c_start (void)

Start a new connection.

2.3.2.3 void i2c_stop (void)

Close the connection.

2.3.2.4 void i2c_write (int *data*)

Write a byte.

Parameters:

data The byte to be transmitted.

Index

- hemavr.c, 1
 - hemavr_init, 2
- hemavr_init
 - hemavr.c, 2
- hemisson.h, 2
 - hemisson_get_brightness, 4
 - hemisson_get_proximity, 4
 - hemisson_set_speed, 4
- hemisson_get_brightness
 - hemisson.h, 4
- hemisson_get_proximity
 - hemisson.h, 4
- hemisson_set_speed
 - hemisson.h, 4

- i2c.h, 4
 - i2c_read, 5
 - i2c_start, 5
 - i2c_stop, 5
 - i2c_write, 5
- i2c_read
 - i2c.h, 5
- i2c_start
 - i2c.h, 5
- i2c_stop
 - i2c.h, 5
- i2c_write
 - i2c.h, 5