Korsel ATTiny2313

1

Generated by Doxygen 1.7.1

Fri Feb 4 2011 13:47:45

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 V24Command Struct Reference

Command struct consisting of header and data.

Data Fields

- unsigned char command
- char data

3.1.1 Detailed Description

Command struct consisting of header and data.

3.1.2 Field Documentation

3.1.2.1 unsigned char V24Command::command

3.1.2.2 char V24Command::data

The documentation for this struct was generated from the following file:

- /home/steffen/Dropbox/Bachelorarbeit/Programmierung/Korsel_attiny2313/src/Korsel_BT.c

Chapter 4

File Documentation

4.1 /home/steffen/Dropbox/Bachelorarbeit/Programmierung/Korsel_-attiny2313/src/Korsel_BT.c File Reference

```
#include <avr/io.h>
#include <avr/interrupt.h>
#include "v24_commands.h"
#include <util/delay.h>
```

Data Structures

- struct V24Command

  Command struct consisting of header and data.

Defines

- #define F_CPU 4000000UL

  Define CPU frequency.

- #define USART_BAUD_RATE 9600

  Debug Mode.

- #define V24_UBRR (F_CPU/(USART_BAUD_RATE∗8L)-1)

  Calculate and define UBRR.

- #define PWM_motor_left OCR1A
- #define PWM_motor_right OCR1B
- #define motor_right_direction PB1
- #define motor_left_direction PB2
- #define RXD PD0
- #define TXD PD1
- #define photo_sensor PD2

- #define button_top PD3
- #define button_left PD4
- #define button_right PD5
- #define button_middle PD6

Typedefs

- typedef struct V24Command V24COMMAND

  Command struct consisting of header and data.

Functions

- void init_io (void)

  initialize IO ports

- void delay (void)

  delay of 100 ms

- void pwm (void)

  Initialize motor outputs as PWM (pulse width modulation).

- void setup_interrupt (void)

  Initialize interrupts.

- void USART_Init ()

  Initialize USART for serial communication.

- void USART_Transmit (unsigned char data)

  Transmit a char on USART.

- void USART_Transmit_Command (V24COMMAND *command)

  Transmit a package consisting of a header and a value on USART.

- void Stop_Motor ()

  If a contact button is pressed this method is called Resets motor speed to 0 and sends a "button pressed" command.

- ISR (INT0_vect)

  ISR for photo sensor.

- ISR (USART_RX_vect)

  ISR for USART receive.

- int main (void)

  Main function.

Variables

- bool LEFT_MOTOR_SPEED_FORWARD_headerDetected = false

  true if last package received was the protocol header LEFT_MOTOR_SPEED_FORWARD

- bool LEFT_MOTOR_SPEED_BACKWARD_headerDetected = false

  true if last package received was the protocol header LEFT_MOTOR_SPEED_BACKWARD

- bool RIGHT_MOTOR_SPEED_FORWARD_headerDetected = false

  true if last package received was the protocol header RIGHT_MOTOR_SPEED_FORWARD

- bool RIGHT_MOTOR_SPEED_BACKWARD_headerDetected = false

  true if last package received was the protocol header RIGHT_MOTOR_SPEED_BACKWARD

4.1.1 Define Documentation

4.1.1.1 #define button_left PD4

4.1.1.2 #define button_middle PD6

4.1.1.3 #define button_right PD5

4.1.1.4 #define button_top PD3

4.1.1.5 #define F_CPU 4000000UL

Define CPU frequency.

4.1.1.6 #define motor_left_direction PB2

4.1.1.7 #define motor_right_direction PB1

4.1.1.8 #define photo_sensor PD2

4.1.1.9 #define PWM_motor_left OCR1A

4.1.1.10 #define PWM_motor_right OCR1B

4.1.1.11 #define RXD PD0

4.1.1.12 #define TXD PD1

4.1.1.13 #define USART_BAUD_RATE 9600

Debug Mode.

4.1.1.14 #define V24_UBRR (F_CPU/(USART_BAUD_RATE∗8L)-1)

Calculate and define UBRR.

4.1.2 Typedef Documentation

4.1.2.1 typedef struct V24Command V24COMMAND

Command struct consisting of header and data.

4.1.3 Function Documentation

4.1.3.1 void delay (void)

delay of 100 ms

4.1.3.2 void init_io (void)

initialize IO ports

4.1.3.3 ISR (INT0_vect)

ISR for photo sensor.

this ISR is called if the photo sensor state changes from white to black or from black to white

4.1.3.4 ISR (USART_RX_vect)

ISR for USART receive.

A command consists of two chars. The first one ist the header and the second one the speed value. The first time the ISR is called the char is read from the UDR. If a header was detected the headerDetected Flag is set. The next time the ISR is called the char read from UDR is a value for motor speed.

4.1.3.5 int main (void)

Main function.

Initializations and main loop

4.1.3.6 void pwm (void)

Initialize motor outputs as PWM (pulse width modulation).

4.1.3.7 void setup_interrupt (void)

Initialize interrupts.

4.1.3.8 void Stop_Motor ( )

If a contact button is pressed this method is called Resets motor speed to 0 and sends a "button pressed" command.

4.1.3.9 void USART_Init ( )

Initialize USART for serial communication.

no parity, 1 stop bit, char size 8

4.1.3.10 void USART_Transmit (unsigned char *data*)

Transmit a char on USART.

4.1.3.11 void USART_Transmit_Command (V24COMMAND * *command*)

Transmit a package consisting of a header and a value on USART.

4.1.4 Variable Documentation

4.1.4.1 bool LEFT_MOTOR_SPEED_BACKWARD_headerDetected = false

true if last package received was the protocol header LEFT_MOTOR_SPEED_BACKWARD

4.1.4.2 bool LEFT_MOTOR_SPEED_FORWARD_headerDetected = false

true if last package received was the protocol header LEFT_MOTOR_SPEED_FORWARD

4.1.4.3 bool RIGHT_MOTOR_SPEED_BACKWARD_headerDetected = false

true if last package received was the protocol header RIGHT_MOTOR_SPEED_BACKWARD

4.1.4.4 bool RIGHT_MOTOR_SPEED_FORWARD_headerDetected = false

true if last package received was the protocol header RIGHT_MOTOR_SPEED_FORWARD

4.2 /home/steffen/Dropbox/Bachelorarbeit/Programmierung/Korsel_attiny2313/src/v24_commands.h File Reference

Defines

- #define LEFT_MOTOR_SPEED_FORWARD 0x01
- #define LEFT_MOTOR_SPEED_BACKWARD 0x11
- #define RIGHT_MOTOR_SPEED_FORWARD 0x02
- #define RIGHT_MOTOR_SPEED_BACKWARD 0x12
- #define PHOTO_SENSOR 0x22
- #define BUTTON_PRESSED 0x33
- #define TOP_BUTTON_PRESSED 0x03
- #define LEFT_BUTTON_PRESSED 0x04
- #define RIGHT_BUTTON_PRESSED 0x05
- #define FRONT_BUTTON_PRESSED 0x06

4.2.1 Define Documentation

4.2.1.1 #define BUTTON_PRESSED 0x33

4.2.1.2 #define FRONT_BUTTON_PRESSED 0x06

4.2.1.3 #define LEFT_BUTTON_PRESSED 0x04

4.2.1.4 #define LEFT_MOTOR_SPEED_BACKWARD 0x11

4.2.1.5 #define LEFT_MOTOR_SPEED_FORWARD 0x01

4.2.1.6 #define PHOTO_SENSOR 0x22

4.2.1.7 #define RIGHT_BUTTON_PRESSED 0x05

4.2.1.8 #define RIGHT_MOTOR_SPEED_BACKWARD 0x12

4.2.1.9 #define RIGHT_MOTOR_SPEED_FORWARD 0x02

4.2.1.10 #define TOP_BUTTON_PRESSED 0x03

Index