AVR Joystick

Generated by Doxygen 1.8.10

Sun Dec 6 2015 23:06:19

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 joystick.c File Reference

```
#include <avr/io.h>
#include <util/delay.h>
#include <stdio.h>
```

Macros

- #define F_CPU 16000000UL
- #define FOSC 16000000
- #define BAUD 9600
- #define MYUBRR FOSC/16/BAUD-1

Functions

- void USART_init (unsigned int ubrr)

  Initializing required variable.
- int USART_send (char c, FILE *stream)

  Send 8bit data.
- int USART_receive (FILE *stream)

  Receive 8bit sata.
- void init_ADC ()

  Initialize ADC.
- void scan_buttons (void)

  Initialize Digital Input buttons.
- int main ()

Variables

- int X_Analog
- int Y_Analog
- int B
- int A
- int F
- int G
- int E

- int D
- int C

2.1.1 Macro Definition Documentation

2.1.1.1 #define BAUD 9600

Baud Rate in bps. refer page 179 of 328p datasheet.

Definition at line 15 of file joystick.c.

2.1.1.2 #define F_CPU 16000000UL

Definition at line 11 of file joystick.c.

2.1.1.3 #define FOSC 16000000

Clock speed for UBRR calculation. refer page 179 of 328p datasheet.

Definition at line 14 of file joystick.c.

2.1.1.4 #define MYUBRR FOSC/16/BAUD-1

UBRR = (F_CPU/(16∗Baud))-1 for asynch USART page 179 328p datasheet. Baud rate 9600bps, assuming 16MHz clock UBRR0 becomes 0x0067

Definition at line 16 of file joystick.c.

2.1.2 Function Documentation

2.1.2.1 void init_ADC ( )

Initialize ADC.

ADMUX section 23.9.1 page 262 BIT 7 and BIT 6 >> AVCC with external capacitor at AREF pin REFS0 =0 and REFS1= 1 BIT 5 >> ADC Right Adjust Result ADLAR = 0 BIT 3:0 >> MUX3:0 0b0000 for channel 0

DIDR0 >> Digital Input Disable Register 0 section Section 23.9.4 page 265 - 266 Disable digital input buffer of ADC0 to save power consumption

ADSCRA ADC Control and Status Register A Section 23.9.2 page 263 -264 Bit 7 >> ADEN: ADC Enable =1 enable ADC Bit 6 >> ADSC: ADC Start Conversion =0 do not start conversion Bit 5 >> ADATE: ADC Auto Trigger Enable = 0 Disable trigger Bit 4 >> ADIF: ADC Interrupt Flag = 0 Bit 3 >> ADIE: ADC Interrupt Enable = 0 Disable ADC interrupt Bits 2:0 >> ADPS2:0: ADC Prescaler Select Bits 010 division factor = 4

Definition at line 77 of file joystick.c.

2.1.2.2 int main ( )

USART Sending And Recieving

Setting Digital PORTS as output

infinite loop

Enable Conversion For ADC0 PIN 0 or A0

Wait for conversion complete.

Store the value in X Analog

Disable Conversion

Change pin to ADC1 or A1

Enable Conversion again

Store the value in Y Analog

Scan for digital button input

Send All input information via USART

Reset Button State

Definition at line 153 of file joystick.c.

2.1.2.3 void scan_buttons (void)

Initialize Digital Input buttons.

D3(PD3)(E) Small Push Button E

D4(PD4)(D) Small Push Button D

D5(PD5)(C) Joystick Push Button C

D6(PD6)(B) Large Push Button B

D7(PD7)(A) Large Push Button A

D8(PB0)(F) Large Push Button F

D9(PB1)(G) Large Push Button GRead for input & change variable state

Definition at line 117 of file joystick.c.

2.1.2.4 void USART_init (unsigned int *ubrr*)

Initializing required variable.

Initialize USART for 8 bit data transmit no parity and 1 stop bit.

This is a code snippet from datasheet page 182

Parameters

ubrr	The UBRR value calculated in macro MYUBRR

See also

MYUBRR

Step 1. Set UCSR0C in Asynchronous mode, no parity, 1 stop bit, 8 data bits

Step 2. Set UCSR0A in Normal speed, disable multi-proc

Step 3. Load ubrr into UBRR0H and UBRR0L

Step 4. Enable Tx Rx and disable interrupt in UCSR0B

Definition at line 30 of file joystick.c.

2.1.2.5 int USART_receive (FILE * *stream*)

Receive 8bit sata.

This is a code snippet from datasheet page 187

Returns

Returns received data from UDR0

Step 1. Wait for Receive Complete Flag is high. Busy waiting

Step 2. Get and return received data from buffer

Definition at line 65 of file joystick.c.

2.1.2.6 int USART_send (char *c*, FILE * *stream*)

Send 8bit data.

This is a code snippet from datasheet page 184

Parameters

data	The 8 bit data to be sent

Step 1. Wait until UDRE0 flag is high. Busy Waiting

Step 2. Write char to UDR0 for transmission

Definition at line 50 of file joystick.c.

2.1.3 Variable Documentation

2.1.3.1 int A

Definition at line 17 of file joystick.c.

2.1.3.2 int B

Definition at line 17 of file joystick.c.

2.1.3.3 int C

Definition at line 17 of file joystick.c.

2.1.3.4 int D

Definition at line 17 of file joystick.c.

2.1.3.5 int E

Definition at line 17 of file joystick.c.

2.1.3.6 int F

Definition at line 17 of file joystick.c.

2.1.3.7 int G

Definition at line 17 of file joystick.c.

2.1.3.8 int X_Analog

Definition at line 17 of file joystick.c.

2.1.3.9 int Y_Analog

Definition at line 17 of file joystick.c.

Index