Little-Wire

Generated by Doxygen 1.8.3.1

Thu May 2 2013 01:43:07

Contents

Chapter 1

Introduction

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Chapter 3

Module Documentation

3.1 General

General library functions.

Typedefs

- typedef usb_dev_handle **littleWire**

Functions

- littleWire ∗ littleWire_connect ()
- unsigned char readFirmwareVersion (littleWire ∗lwHandle)
- int customMessage (littleWire ∗lwHandle, unsigned char ∗receiveBuffer, unsigned char command, unsigned char d1, unsigned char d2, unsigned char d3, unsigned char d4)
- int littleWire_error ()
- char ∗ littleWire_errorName ()

3.1.1 Detailed Description

General library functions.

3.1.2 Function Documentation

3.1.2.1 int customMessage (littleWire ∗ *lwHandle,* unsigned char ∗ *receiveBuffer,* unsigned char *command,* unsigned char *d1,* unsigned char *d2,* unsigned char *d3,* unsigned char *d4*)

Sends a custom message to the device.

Useful when developing new features in the firmware.

Parameters

receiveBuffer	Returned data buffer
command	Firmware command
d1	data[0] for the command
d2	data[1] for the command
d3	data[2] for the command
d4	data[3] for the command

Returns

status

Definition at line 273 of file littleWire.c.

3.1.2.2 littleWire* littleWire_connect ( )

Tries to connect to the device.

Parameters

(none)	

Returns

littleWire pointer for healthy connection, NULL for a failed trial.

Definition at line 60 of file littleWire.c.

3.1.2.3 int littleWire_error ( )

Returns the numeric value of the status of the last communication attempt

Parameters

(none)	

Returns

Numeric value of the status of the last communication attempt

Definition at line 427 of file littleWire.c.

3.1.2.4 char* littleWire_errorName ( )

Returns the string version of the last communication attempt status if there was an error

Parameters

(none)	

Returns

String version of the last communication attempt status if there was an error

Definition at line 432 of file littleWire.c.

3.1.2.5 unsigned char readFirmwareVersion (littleWire * *lwHandle*)

Reads the firmware version of the Little Wire

Format: 0xXY => X: Primary version Y: Minor version

Parameters

(none)	

Returns

Firmware version

Definition at line 70 of file littleWire.c.

3.2 GPIO

GPIO library functions with Arduino-like syntax.

Functions

- void digitalWrite (littleWire *lwHandle, unsigned char pin, unsigned char state)
- void pinMode (littleWire *lwHandle, unsigned char pin, unsigned char mode)
- unsigned char digitalRead (littleWire *lwHandle, unsigned char pin)
- void internalPullup (littleWire *lwHandle, unsigned char pin, unsigned char state)

3.2.1 Detailed Description

GPIO library functions with Arduino-like syntax.

3.2.2 Function Documentation

3.2.2.1 unsigned char digitalRead (littleWire * *lwHandle,* unsigned char *pin*)

Read pin value

Parameters

lwHandle	littleWire device pointer
pin	Pin name (**PIN1** , **PIN2** , **PIN3** or **PIN4**)

Returns

Pin state (**HIGH** or **LOW**)

Definition at line 95 of file littleWire.c.

3.2.2.2 void digitalWrite (littleWire * *lwHandle,* unsigned char *pin,* unsigned char *state*)

Set pin value

Parameters

lwHandle	littleWire device pointer
pin	Pin name (**PIN1** , **PIN2** , **PIN3** or **PIN4**)
state	Pin state (**HIGH** or **LOW**)

Returns

(none)

Definition at line 77 of file littleWire.c.

3.2.2.3 void internalPullup (littleWire * *lwHandle,* unsigned char *pin,* unsigned char *state*)

Sets the state of the internal pullup resistor.

Call this function after you assign the pin as an input.

Parameters

lwHandle	littleWire device pointer
pin	Pin name (**PIN1** , **PIN2** , **PIN3** or **PIN4**) state (**ENABLE** or **DISABLE**)

Returns

(none)

Definition at line 102 of file littleWire.c.

3.2.2.4 void pinMode (littleWire * *lwHandle,* unsigned char *pin,* unsigned char *mode*)

Set pin as input/output

Parameters

lwHandle	littleWire device pointer
pin	Pin name (**PIN1** , **PIN2** , **PIN3** or **PIN4**)
mode	Mode of pin (**INPUT** or **OUTPUT**)

Returns

(none)

Definition at line 86 of file littleWire.c.

3.3 ADC

Analog to digital converter functions.

Functions

- void analog_init (littleWire *lwHandle, unsigned char voltageRef)
- unsigned int analogRead (littleWire *lwHandle, unsigned char channel)

3.3.1 Detailed Description

Analog to digital converter functions.

3.3.2 Function Documentation

3.3.2.1 void analog_init (littleWire * *lwHandle,* unsigned char *voltageRef*)

Initialize the analog module. VREF_VCC is the standard voltage coming from the USB plug

Others are the Attiny's internal voltage references.

Parameters

lwHandle	littleWire device pointer
voltageRef	(**VREF_VCC** , **VREF_110mV** or **VREF_2560mV**)

Returns

(none)

Definition at line 111 of file littleWire.c.

3.3.2.2 unsigned int analogRead (littleWire * *lwHandle,* unsigned char *channel*)

Read analog voltage. Analog voltage reading from ADC_PIN3 isn't advised (it is a bit noisy) but supported. Use it at your own risk.

For more about internal temperature sensor, look at the Attiny85 datasheet.

Parameters

lwHandle	littleWire device pointer
channel	Source of ADC reading (**ADC_PIN2** , **ADC_PIN3** or **ADC_TEMP_SENS**)

Returns

10 bit ADC result

Definition at line 116 of file littleWire.c.

3.4 PWM

Pulse width modulation functions.

Functions

- void pwm_init (littleWire *lwHandle)
- void pwm_stop (littleWire *lwHandle)
- void pwm_updateCompare (littleWire *lwHandle, unsigned char channelA, unsigned char channelB)
- void pwm_updatePrescaler (littleWire *lwHandle, unsigned int value)

3.4.1 Detailed Description

Pulse width modulation functions.

3.4.2 Function Documentation

3.4.2.1 void pwm_init (littleWire * *lwHandle*)

Initialize the PWM module on the Little-Wire

Parameters

lwHandle	littleWire device pointer

Returns

(none)

Definition at line 123 of file littleWire.c.

3.4.2.2 void pwm_stop (littleWire * *lwHandle*)

Stop the PWM module on the Little-Wire

Parameters

lwHandle	littleWire device pointer

Returns

(none)

Definition at line 128 of file littleWire.c.

3.4.2.3 void pwm_updateCompare (littleWire * *lwHandle,* unsigned char *channelA,* unsigned char *channelB*)

Update the compare values of the PWM output pins. Resolution is 8 bit.

Parameters

lwHandle	littleWire device pointer
channelA	Compare value of **PWMA** pin
channelB	Compare value of **PWMB** pin

Returns

(none)

Definition at line 133 of file littleWire.c.

3.4.2.4 void pwm_updatePrescaler (littleWire * *lwHandle,* unsigned int *value*)

Update the prescaler of the PWM module. Adjust this value according to your need for speed in PWM output. Default is 1024. Lower prescale means higher frequency PWM output.

Parameters

lwHandle	littleWire device pointer
value	Presecaler value (**1024**, **256**, **64**, **8** or **1**)

Returns

(none)

Definition at line 138 of file littleWire.c.

3.5 SPI

Serial peripheral interface functions.

Functions

- void spi_init (littleWire *lwHandle)
- void spi_sendMessage (littleWire *lwHandle, unsigned char *sendBuffer, unsigned char *inputBuffer, unsigned char length, unsigned char mode)
- unsigned char debugSpi (littleWire *lwHandle, unsigned char message)
- void spi_updateDelay (littleWire *lwHandle, unsigned int duration)

3.5.1 Detailed Description

Serial peripheral interface functions.

3.5.2 Function Documentation

3.5.2.1 unsigned char debugSpi (littleWire * *lwHandle,* unsigned char *message*)

Send one byte SPI message over MOSI pin. Slightly slower than the actual one.

There isn't any chip select control involved. Useful for debug console app

Parameters

lwHandle	littleWire device pointer
message	Message to send

Returns

Received SPI message

Definition at line 176 of file littleWire.c.

3.5.2.2 void spi_init (littleWire * *lwHandle*)

Initialize the SPI module on the Little-Wire

Parameters

lwHandle	littleWire device pointer

Returns

(none)

Definition at line 160 of file littleWire.c.

3.5.2.3 void spi_sendMessage (littleWire * *lwHandle,* unsigned char * *sendBuffer,* unsigned char * *inputBuffer,* unsigned char *length,* unsigned char *mode*)

Send SPI message(s). SPI Mode is 0.

Parameters

lwHandle	littleWire device pointer
sendBuffer	Message array to send
inputBuffer	Returned answer message
length	Message length - maximum 4
mode	**AUTO_CS** or **MANUAL_CS**

Returns

(none)

Definition at line 165 of file littleWire.c.

3.5.2.4 void spi_updateDelay (littleWire * *lwHandle,* unsigned int *duration*)

Change the SPI message frequency by adjusting delay duration. By default, Little-Wire sends the SPI messages with maximum speed.

If your hardware can't catch up with the speed, increase the duration value to lower the SPI speed.

Parameters

lwHandle	littleWire device pointer
duration	Amount of delay.

Returns

(none)

Definition at line 183 of file littleWire.c.

3.6 I2C

Inter IC communication functions.

Functions

- void i2c_init (littleWire *lwHandle)
- unsigned char i2c_start (littleWire *lwHandle, unsigned char address7bit, unsigned char direction)
- void i2c_write (littleWire *lwHandle, unsigned char *sendBuffer, unsigned char length, unsigned char endWithStop)
- void i2c_read (littleWire *lwHandle, unsigned char *readBuffer, unsigned char length, unsigned char endWithStop)
- void i2c_updateDelay (littleWire *lwHandle, unsigned int duration)

3.6.1 Detailed Description

Inter IC communication functions.

3.6.2 Function Documentation

3.6.2.1 void i2c_init (littleWire * *lwHandle*)

Initialize the I2C module on the Little-Wire

Returns

(none)

Definition at line 188 of file littleWire.c.

3.6.2.2 void i2c_read (littleWire * *lwHandle,* unsigned char * *readBuffer,* unsigned char *length,* unsigned char *endWithStop*)

Read byte(s) over i2c bus

Parameters

lwHandle	littleWire device pointer
readBuffer	Returned message array
length	Message length -> Max = 8
endWithStop	Should we send a STOP condition after this buffer? (**END_WITH_STOP** or **NO_STOP**)

Returns

(none)

Definition at line 209 of file littleWire.c.

3.6.2.3 unsigned char i2c_start (littleWire * *lwHandle,* unsigned char *address7bit,* unsigned char *direction*)

Start the i2c communication

Parameters

lwHandle	littleWire device pointer
address	7 bit slave address.
direction	(**READ** or **WRITE**)

Returns

1 if received ACK

Definition at line 193 of file littleWire.c.

3.6.2.4 void i2c_updateDelay (littleWire * *lwHandle,* unsigned int *duration*)

Update i2c signal delay amount. Tune if neccessary to fit your requirements.

Parameters

lwHandle	littleWire device pointer
duration	Delay amount

Returns

(none)

Definition at line 224 of file littleWire.c.

3.6.2.5 void i2c_write (littleWire * *lwHandle,* unsigned char * *sendBuffer,* unsigned char *length,* unsigned char *endWithStop*)

Send byte(s) over i2c bus

Parameters

lwHandle	littleWire device pointer
sendBuffer	Message array to send
length	Message length -> Max = 4
endWithStop	Should we send a STOP condition after this buffer? (**END_WITH_STOP** or **NO_STOP**)

Returns

(none)

Definition at line 204 of file littleWire.c.

3.7 Onewire

Onewire functions.

Functions

- void onewire_sendBit (littleWire *lwHandle, unsigned char bitValue)
- void onewire_writeByte (littleWire *lwHandle, unsigned char messageToSend)
- unsigned char onewire_readByte (littleWire *lwHandle)
- unsigned char onewire_readBit (littleWire *lwHandle)
- unsigned char onewire_resetPulse (littleWire *lwHandle)
- int onewire_firstAddress (littleWire *lwHandle)
- int onewire_nextAddress (littleWire *lwHandle)

3.7.1 Detailed Description

Onewire functions.

3.7.2 Function Documentation

3.7.2.1 int onewire_firstAddress (littleWire * *lwHandle*)

Start searching for device address on the onewire bus.

Read the 8 byte address from **ROM_NO** array

Parameters

lwHandle	littleWire device pointer

Returns

Nonzero if any device found

Definition at line 415 of file littleWire.c.

3.7.2.2 int onewire_nextAddress (littleWire * *lwHandle*)

Try to find the next adress on the onewire bus.

Read the 8 byte address from **ROM_NO** array

Parameters

lwHandle	littleWire device pointer

Returns

Nonzero if any new device found

Definition at line 296 of file littleWire.c.

3.7.2.3 unsigned char onewire_readBit (littleWire * *lwHandle*)

Read a single bit over onewire bus

Parameters

lwHandle	littleWire device pointer

Returns

Read bit (**1** or **0**)

Definition at line 248 of file littleWire.c.

3.7.2.4 unsigned char onewire_readByte (littleWire * *lwHandle*)

Read a byte over onewire bus.

Parameters

lwHandle	littleWire device pointer

Returns

Read byte

Definition at line 240 of file littleWire.c.

3.7.2.5 unsigned char onewire_resetPulse (littleWire * *lwHandle*)

Send a reset pulse over onewire bus

Parameters

lwHandle	littleWire device pointer

Returns

Nonzero if any device presents on the bus

Definition at line 255 of file littleWire.c.

3.7.2.6 void onewire_sendBit (littleWire * *lwHandle,* unsigned char *bitValue*)

Send a single bit over onewire bus.

Parameters

lwHandle	littleWire device pointer
bitValue	**1** or **0**

Returns

(none)

Definition at line 229 of file littleWire.c.

3.7.2.7 void onewire_writeByte (littleWire * *lwHandle,* unsigned char *messageToSend*)

Send a byte over onewire bus.

Parameters

lwHandle	littleWire device pointer
messageToSend	Message to send

Returns

(none)

Definition at line 234 of file littleWire.c.

3.8 SOFT_PWM

Software PWM functions. Designed to be used with RGB LEDs.

Functions

- void softPWM_state (littleWire *lwHandle, unsigned char state)
- void softPWM_write (littleWire *lwHandle, unsigned char ch1, unsigned char ch2, unsigned char ch3)

3.8.1 Detailed Description

Software PWM functions. Designed to be used with RGB LEDs.

3.8.2 Function Documentation

3.8.2.1 void softPWM_state (littleWire * *lwHandle,* unsigned char *state*)

Sets the state of the softPWM module

Parameters

lwHandle	littleWire device pointer
state	State of the softPWM module (**ENABLE** or **DISABLE**)

Returns

(none)

Definition at line 263 of file littleWire.c.

3.8.2.2 void softPWM_write (littleWire * *lwHandle,* unsigned char *ch1,* unsigned char *ch2,* unsigned char *ch3*)

Updates the values of softPWM modules

Parameters

lwHandle	littleWire device pointer
ch1	Value of channel 1 - **PIN4**
ch2	Value of channel 2 - **PIN1**
ch3	Value of channel 3 - **PIN2**

Returns

(none)

Definition at line 268 of file littleWire.c.

3.9 Servo

Servo functions. Higher level access to PWM module.

Functions

- void servo_init (littleWire *lwHandle)
- void servo_updateLocation (littleWire *lwHandle, unsigned char locationChannelA, unsigned char locationChannelB)

3.9.1 Detailed Description

Servo functions. Higher level access to PWM module.

3.9.2 Function Documentation

3.9.2.1 void servo_init (littleWire * *lwHandle*)

Initialize the PWM module on the Little-Wire with the Servo special settings.

Parameters

lwHandle	littleWire device pointer

Returns

(none)

Definition at line 39 of file littleWire_servo.c.

3.9.2.2 void servo_updateLocation (littleWire * *lwHandle*, unsigned char *locationChannelA*, unsigned char *locationChannelB*)

Update servo locations

Parameters

lwHandle	littleWire device pointer
locationChannelA	Location of servo connected to channel A (in degrees)
locationChannelB	Location of servo connected to channel B (in degrees)

Returns

(none)

Definition at line 52 of file littleWire_servo.c.

Index