

mmculib Reference Manual

1.0.3

Generated by Doxygen 1.4.6

Fri Aug 24 14:44:40 2007

Contents

1	mmculib Hierarchical Index	1
1.1	mmculib Class Hierarchy	1
2	mmculib Data Structure Index	3
2.1	mmculib Data Structures	3
3	mmculib File Index	5
3.1	mmculib File List	5
4	mmculib Data Structure Documentation	9
4.1	biseq_obj_t Struct Reference	9
4.2	busart_dev_struct Struct Reference	10
4.3	button_cfg_t Struct Reference	11
4.4	button_private_t Struct Reference	12
4.5	buttons_private_t Struct Reference	13
4.6	chaser_private_t Struct Reference	14
4.7	cleds_private_t Struct Reference	15
4.8	flasher_obj_t Struct Reference	16
4.9	flasher_pattern_t Struct Reference	17
4.10	flasher_private_t Struct Reference	18
4.11	font_t Struct Reference	19
4.12	lcd_cfg_t Struct Reference	20
4.13	lcd_obj_t Struct Reference	21
4.14	led_cfg_t Struct Reference	22
4.15	lmatrix_port_t Struct Reference	23
4.16	lmatrix_private_t Struct Reference	24
4.17	mbuttons_private_t Struct Reference	25
4.18	mcleds_private_t Struct Reference	26
4.19	mcleds_state_t Struct Reference	27

4.20	mmelody_private_t Struct Reference	28
4.21	mpwm_channel_t Struct Reference	29
4.22	mpwm_obj_t Struct Reference	30
4.23	mtext_obj_t Struct Reference	31
4.24	mulwire_obj_t Struct Reference	32
4.25	mulwire_rom_code_t Union Reference	33
4.26	muxleds_cfg_t Struct Reference	34
4.27	muxleds_col_t Struct Reference	35
4.28	muxleds_obj_t Struct Reference	36
4.29	muxleds_row_t Struct Reference	37
4.30	nrf_cfg_t Struct Reference	38
4.31	nrf_config_bits_t Struct Reference	39
4.32	nrf_config_t Union Reference	41
4.33	nrf_obj_t Struct Reference	42
4.34	nrf_pins_t Struct Reference	43
4.35	pga_cfg_t Struct Reference	44
4.36	pga_private_t Struct Reference	45
4.37	piezo_cfg_t Struct Reference	46
4.38	rf_address_t Struct Reference	47
4.39	rf_node_t Struct Reference	48
4.40	rf_probe_t Struct Reference	49
4.41	ring_struct Struct Reference	50
4.42	scroller_obj_t Struct Reference	51
4.43	seq_obj_t Struct Reference	52
4.44	sflash_obj_t Struct Reference	53
4.45	spi_eeprom_cfg_t Struct Reference	54
4.46	spi_eeprom_private_t Struct Reference	55
4.47	spwm_obj_t Struct Reference	56
4.48	squeaker_private_t Struct Reference	57
4.49	stext_obj_t Struct Reference	59
4.50	ticker16_t Struct Reference	60
4.51	ticker8_t Struct Reference	61
4.52	ticker_t Struct Reference	62
4.53	time Struct Reference	63
4.54	tweeter_private_t Struct Reference	64
4.55	ulwire_enumerate_t Struct Reference	65

4.56	u1wire_obj_t Struct Reference	66
4.57	u1wire_rom_code_t Union Reference	67
4.58	u1wire_state_t Struct Reference	68
4.59	usart_dev_struct Struct Reference	69
5	mmculib File Documentation	71
5.1	ads7870.c File Reference	71
5.2	ads7870.h File Reference	74
5.3	ads8325.c File Reference	75
5.4	ads8327.c File Reference	76
5.5	ads8327.h File Reference	77
5.6	biseq.c File Reference	78
5.7	biseq.h File Reference	79
5.8	bits.h File Reference	81
5.9	busart.c File Reference	82
5.10	busart.h File Reference	84
5.11	button.c File Reference	86
5.12	button.h File Reference	88
5.13	buttons.c File Reference	91
5.14	buttons.h File Reference	92
5.15	chaser.c File Reference	94
5.16	chaser.h File Reference	95
5.17	cleds.c File Reference	97
5.18	cleds.h File Reference	98
5.19	colourmap.h File Reference	100
5.20	crc8541.c File Reference	102
5.21	crc8541.h File Reference	103
5.22	crc8541_test.c File Reference	104
5.23	ds18b20.c File Reference	105
5.24	ds18b20.h File Reference	107
5.25	ds2450.c File Reference	109
5.26	ds2450.h File Reference	111
5.27	dscrc16.c File Reference	112
5.28	dscrc16.h File Reference	113
5.29	dscrc8.c File Reference	114
5.30	dscrc8.h File Reference	115
5.31	flasher.c File Reference	116

5.32	flasher.h File Reference	117
5.33	flasher_tweak.c File Reference	119
5.34	flasher_tweak.h File Reference	120
5.35	font.c File Reference	121
5.36	font.h File Reference	122
5.37	isqrt16.c File Reference	123
5.38	isqrt32.c File Reference	124
5.39	lcd.c File Reference	125
5.40	lcd.h File Reference	128
5.41	led.c File Reference	130
5.42	led.h File Reference	131
5.43	led_flash.c File Reference	133
5.44	led_flash.h File Reference	134
5.45	lmatrix.c File Reference	135
5.46	lmatrix.h File Reference	136
5.47	mbuttons.c File Reference	138
5.48	mbuttons.h File Reference	139
5.49	mcleds.c File Reference	141
5.50	mcleds.h File Reference	142
5.51	mmelody.c File Reference	144
5.52	mmelody.h File Reference	146
5.53	mpwm.c File Reference	148
5.54	mpwm.h File Reference	149
5.55	mtext.c File Reference	150
5.56	mtext.h File Reference	151
5.57	mu1wire.c File Reference	153
5.58	mu1wire.h File Reference	155
5.59	muxleds.c File Reference	157
5.60	muxleds.h File Reference	158
5.61	nmea.c File Reference	160
5.62	nmea.h File Reference	161
5.63	nrf2401.c File Reference	162
5.64	nrf2401.h File Reference	166
5.65	nrf_config.h File Reference	171
5.66	pga.c File Reference	173
5.67	pga.h File Reference	176

5.68	piezo.c File Reference	179
5.69	piezo.h File Reference	180
5.70	piezo_beep.c File Reference	181
5.71	piezo_beep.h File Reference	182
5.72	rf.c File Reference	183
5.73	rf.h File Reference	185
5.74	rf_master.c File Reference	189
5.75	rf_slave.c File Reference	191
5.76	ring.c File Reference	192
5.77	ring.h File Reference	194
5.78	s_eeprom.c File Reference	196
5.79	s_eeprom.h File Reference	199
5.80	scroller.c File Reference	201
5.81	scroller.h File Reference	202
5.82	seq.c File Reference	204
5.83	seq.h File Reference	205
5.84	sflash.c File Reference	206
5.85	sflash.h File Reference	207
5.86	spi_adc.h File Reference	208
5.87	spwm.c File Reference	209
5.88	spwm.h File Reference	210
5.89	squeaker.c File Reference	211
5.90	squeaker.h File Reference	213
5.91	squeaker2.c File Reference	216
5.92	stext.c File Reference	218
5.93	stext.h File Reference	219
5.94	ticker.c File Reference	221
5.95	ticker.h File Reference	222
5.96	time.c File Reference	223
5.97	time.h File Reference	224
5.98	tweeter.c File Reference	226
5.99	tweeter.h File Reference	228
5.100	u1wire.c File Reference	230
5.101	u1wire.h File Reference	233
5.102	u1wire_debug.c File Reference	235
5.103	u1wire_discover.c File Reference	236

5.104u1wire_discover.h File Reference	237
5.105u1wire_enumerate.c File Reference	238
5.106u1wire_enumerate.h File Reference	239
5.107uint16toa.c File Reference	240
5.108uint16toa.h File Reference	241
5.109uint8toa.c File Reference	242
5.110uint8toa.h File Reference	243
5.111usart.c File Reference	244
5.112usart.h File Reference	246

Chapter 1

mmculib Hierarchical Index

1.1 mmculib Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

biseq_obj_t	9
busart_dev_struct	10
button_cfg_t	11
button_private_t	12
buttons_private_t	13
chaser_private_t	14
cleds_private_t	15
flasher_obj_t	16
flasher_pattern_t	17
flasher_private_t	18
font_t	19
lcd_cfg_t	20
lcd_obj_t	21
led_cfg_t	22
lmatrix_port_t	23
lmatrix_private_t	24
mbuttons_private_t	25
mcleds_private_t	26
mcleds_state_t	27
mmelody_private_t	28
mpwm_channel_t	29
mpwm_obj_t	30
mtext_obj_t	31
mulwire_obj_t	32
mulwire_rom_code_t	33
muxleds_cfg_t	34
muxleds_col_t	35
muxleds_obj_t	36
muxleds_row_t	37
nrf_cfg_t	38
nrf_config_bits_t	39
nrf_config_t	41
nrf_obj_t	42

nrf_pins_t	43
pga_cfg_t	44
pga_private_t	45
piezo_cfg_t	46
rf_address_t	47
rf_node_t	48
rf_probe_t	49
ring_struct	50
scroller_obj_t	51
seq_obj_t	52
sflash_obj_t	53
spi_eeprom_cfg_t	54
spi_eeprom_private_t	55
spwm_obj_t	56
squeaker_private_t	57
stext_obj_t	59
ticker16_t	60
ticker8_t	61
ticker_t	62
time	63
tweeter_private_t	64
ulwire_enumerate_t	65
ulwire_obj_t	66
ulwire_rom_code_t	67
ulwire_state_t	68
usart_dev_struct	69

Chapter 2

mmculib Data Structure Index

2.1 mmculib Data Structures

Here are the data structures with brief descriptions:

biseq_obj_t	9
busart_dev_struct	10
button_cfg_t	11
button_private_t	12
buttons_private_t	13
chaser_private_t	14
cleds_private_t	15
flasher_obj_t	16
flasher_pattern_t	17
flasher_private_t	18
font_t	19
lcd_cfg_t	20
lcd_obj_t	21
led_cfg_t	22
lmatrix_port_t	23
lmatrix_private_t	24
mbuttons_private_t	25
mcleds_private_t	26
mcleds_state_t	27
mmelody_private_t	28
mpwm_channel_t	29
mpwm_obj_t	30
mtext_obj_t	31
mulwire_obj_t	32
mulwire_rom_code_t	33
muxleds_cfg_t	34
muxleds_col_t	35
muxleds_obj_t	36
muxleds_row_t	37
nrf_cfg_t	38
nrf_config_bits_t	39
nrf_config_t	41
nrf_obj_t	42

nrf_pins_t	43
pga_cfg_t	44
pga_private_t	45
piezo_cfg_t	46
rf_address_t	47
rf_node_t	48
rf_probe_t	49
ring_struct	50
scroller_obj_t	51
seq_obj_t	52
sflash_obj_t	53
spi_eeprom_cfg_t	54
spi_eeprom_private_t	55
spwm_obj_t	56
squeaker_private_t	57
stext_obj_t	59
ticker16_t	60
ticker8_t	61
ticker_t	62
time	63
tweeter_private_t	64
ulwire_enumerate_t	65
ulwire_obj_t	66
ulwire_rom_code_t	67
ulwire_state_t	68
usart_dev_struct	69

Chapter 3

mmculib File Index

3.1 mmculib File List

Here is a list of all files with brief descriptions:

ads7870.c	71
ads7870.h (ADS7870 SPI ADC)	74
ads8325.c	75
ads8327.c	76
ads8327.h (ADS8327 SPI ADC)	77
biseq.c (Bidirectional sequencer)	78
biseq.h	79
bits.h	81
busart.c (Buffered USART implementation)	82
busart.h (Buffered USART interface)	84
button.c	86
button.h (Button polling and debouncing)	88
buttons.c	91
buttons.h (Multiple button polling and debouncing)	92
chaser.c	94
chaser.h	95
cleds.c	97
cleds.h	98
colourmap.h	100
crc8541.c	102
crc8541.h	103
crc8541_test.c	104
ds18b20.c	105
ds18b20.h	107
ds2450.c	109
ds2450.h	111
dscrc16.c	112
dscrc16.h	113
dscrc8.c	114
dscrc8.h	115
flasher.c	116
flasher.h	117
flasher_tweak.c	119

flasher_tweak.h	120
font.c	121
font.h	122
isqrt16.c	123
isqrt32.c	124
lcd.c	125
lcd.h	128
led.c	130
led.h	131
led_flash.c	133
led_flash.h (LED flashing routine)	134
lmatrix.c	135
lmatrix.h (This drives a multiplexed LED matrix. It only supports a single instance)	136
mbuttons.c	138
mbuttons.h	139
mcleds.c	141
mcleds.h	142
mmelody.c (Play simple melodies)	144
mmelody.h (Play simple melodies)	146
mpwm.c	148
mpwm.h	149
mtext.c (Moving text)	150
mtext.h (Moving text)	151
mulwire.c	153
mulwire.h	155
muxleds.c	157
muxleds.h	158
nmea.c	160
nmea.h	161
nrf2401.c	162
nrf2401.h	166
nrf_config.h	171
pga.c	173
pga.h	176
piezo.c	179
piezo.h	180
piezo_beep.c (Piezo beeping routines. Note these block)	181
piezo_beep.h	182
rf.c	183
rf.h	185
rf_master.c	189
rf_slave.c	191
ring.c (Ring buffer implementation)	192
ring.h (Ring buffer interface)	194
s_eeprom.c	196
s_eeprom.h	199
scroller.c	201
scroller.h (Image scroller)	202
seq.c	204
seq.h	205
sflash.c	206
sflash.h	207
spi_adc.h (SPI ADC)	208
spwm.c	209

spwm.h	210
squeaker.c (Play simple tunes with PWM)	211
squeaker.h (Play simple tunes with PWM)	213
squeaker2.c	216
stext.c (Sequenced text)	218
stext.h (Sequenced text)	219
ticker.c	221
ticker.h	222
time.c (Time routines)	223
time.h (Time routines)	224
tweeter.c (Generate PWM for a piezo tweeter)	226
tweeter.h (Generate PWM for a piezo tweeter)	228
ulwire.c (Low level routines to drive Dallas universal 1 wire bus. This only supports a single instance of a 1 wire bus)	230
ulwire.h	233
ulwire_debug.c	235
ulwire_discover.c	236
ulwire_discover.h (This discovers devices on a Dallas universal one wire bus. Note this has been superseded by ulwire_enumerate)	237
ulwire_enumerate.c	238
ulwire_enumerate.h (This discovers devices on a Dallas universal one wire bus)	239
uint16toa.c (16 bit unsigned int to ASCII conversion)	240
uint16toa.h (16 bit unsigned int to ASCII conversion)	241
uint8toa.c	242
uint8toa.h	243
usart.c (Unbuffered USART implementation)	244
usart.h (Unbuffered USART interface)	246

Chapter 4

mmculib Data Structure Documentation

4.1 `biseq_obj_t` Struct Reference

```
#include <biseq.h>
```

Data Fields

- `char * str`
- `uint8_t step`
- `int8_t dir`
- `biseq_mode_t mode`
- `int8_t(* callback)(void *data, char *str)`
- `void * callback_data`

4.1.1 Field Documentation

4.1.1.1 `int8_t(* biseq_obj_t::callback)(void *data, char *str)`

4.1.1.2 `void* biseq_obj_t::callback_data`

4.1.1.3 `int8_t biseq_obj_t::dir`

4.1.1.4 `biseq_mode_t biseq_obj_t::mode`

4.1.1.5 `uint8_t biseq_obj_t::step`

4.1.1.6 `char* biseq_obj_t::str`

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

- `biseq.h`

4.2 busart_dev_struct Struct Reference

Data Fields

- void(* [tx_irq_enable](#))(void)
- void(* [rx_irq_enable](#))(void)
- bool(* [tx_finished_p](#))(void)
- [ring_t](#) tx_ring
- [ring_t](#) rx_ring

4.2.1 Field Documentation

4.2.1.1 void(* [busart_dev_struct::rx_irq_enable](#))(void)

4.2.1.2 [ring_t](#) busart_dev_struct::rx_ring

4.2.1.3 bool(* [busart_dev_struct::tx_finished_p](#))(void)

4.2.1.4 void(* [busart_dev_struct::tx_irq_enable](#))(void)

4.2.1.5 [ring_t](#) busart_dev_struct::tx_ring

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

- [busart.c](#)

4.3 button_cfg_t Struct Reference

```
#include <button.h>
```

Data Fields

- port_t [port](#)
- port_mask_t [bitmask](#)

4.3.1 Field Documentation

4.3.1.1 port_mask_t [button_cfg_t::bitmask](#)

4.3.1.2 port_t [button_cfg_t::port](#)

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

- [button.h](#)

4.4 button_private_t Struct Reference

```
#include <button.h>
```

Data Fields

- [button_state_t](#) state
- const [button_cfg_t](#) * cfg
- [uint8_t](#) count
- [uint8_t](#) hold_count

4.4.1 Field Documentation

4.4.1.1 const [button_cfg_t](#)* [button_private_t::cfg](#)

4.4.1.2 [uint8_t](#) [button_private_t::count](#)

4.4.1.3 [uint8_t](#) [button_private_t::hold_count](#)

4.4.1.4 [button_state_t](#) [button_private_t::state](#)

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

- [button.h](#)

4.5 buttons_private_t Struct Reference

```
#include <buttons.h>
```

Data Fields

- [button_t buttons](#)
- [uint8_t num](#)

4.5.1 Detailed Description

Private multiple button structure.

Note:

These elements should be considered private so do not access them.

4.5.2 Field Documentation

4.5.2.1 [button_t buttons_private_t::buttons](#)

4.5.2.2 [uint8_t buttons_private_t::num](#)

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

- [buttons.h](#)

4.6 `chaser_private_t` Struct Reference

```
#include <chaser.h>
```

Data Fields

- `uint8_t` `flasher_num`
- `flasher_t` * `flashers`
- `chaser_sequence_t` `seq`
- `font_t` * `font`
- `uint8_t` `step`
- `int8_t` `dir`
- `chaser_mode_t` `mode`
- `flasher_pattern_t` * `on_pattern`
- `flasher_pattern_t` * `off_pattern`

4.6.1 Field Documentation

4.6.1.1 `int8_t` `chaser_private_t::dir`

4.6.1.2 `uint8_t` `chaser_private_t::flasher_num`

4.6.1.3 `flasher_t`* `chaser_private_t::flashers`

4.6.1.4 `font_t`* `chaser_private_t::font`

4.6.1.5 `chaser_mode_t` `chaser_private_t::mode`

4.6.1.6 `flasher_pattern_t`* `chaser_private_t::off_pattern`

4.6.1.7 `flasher_pattern_t`* `chaser_private_t::on_pattern`

4.6.1.8 `chaser_sequence_t` `chaser_private_t::seq`

4.6.1.9 `uint8_t` `chaser_private_t::step`

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

- `chaser.h`

4.7 cleds_private_t Struct Reference

```
#include <cleds.h>
```

Data Fields

- const [led_cfg_t](#) * [leds](#)
- const [led_cfg_t](#) * [row_config](#)
- [uint8_t](#) [rows_num](#)
- [uint8_t](#) [cols_num](#)
- [uint8_t](#) [row](#)

4.7.1 Field Documentation

4.7.1.1 [uint8_t cleds_private_t::cols_num](#)

4.7.1.2 [const led_cfg_t* cleds_private_t::leds](#)

4.7.1.3 [uint8_t cleds_private_t::row](#)

4.7.1.4 [const led_cfg_t* cleds_private_t::row_config](#)

4.7.1.5 [uint8_t cleds_private_t::rows_num](#)

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

- [cleds.h](#)

4.8 flasher_obj_t Struct Reference

```
#include <flasher.h>
```

Data Fields

- char [dummy](#) [sizeof([flasher_private_t](#))]

4.8.1 Field Documentation

4.8.1.1 char [flasher_obj_t::dummy](#)[sizeof([flasher_private_t](#))]

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

- [flasher.h](#)

4.9 flasher_pattern_t Struct Reference

```
#include <flasher.h>
```

Data Fields

- [uint8_t mod_period](#)
- [uint8_t mod_duty](#)
- [uint8_t flasher_period](#)
- [uint8_t flasher_duty](#)
- [uint8_t flashes](#)
- [uint8_t period](#)

4.9.1 Field Documentation

4.9.1.1 [uint8_t flasher_pattern_t::flasher_duty](#)

4.9.1.2 [uint8_t flasher_pattern_t::flasher_period](#)

4.9.1.3 [uint8_t flasher_pattern_t::flashes](#)

4.9.1.4 [uint8_t flasher_pattern_t::mod_duty](#)

4.9.1.5 [uint8_t flasher_pattern_t::mod_period](#)

4.9.1.6 [uint8_t flasher_pattern_t::period](#)

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

- [flasher.h](#)

4.10 `flasher_private_t` Struct Reference

```
#include <flasher.h>
```

Data Fields

- [flasher_pattern_t](#) * `pattern`
- [uint8_t](#) `mod_count`
- [uint8_t](#) `flasher_count`
- [uint8_t](#) `flashes_count`
- [uint8_t](#) `flasher_prescale`

4.10.1 Field Documentation

4.10.1.1 [uint8_t flasher_private_t::flasher_count](#)

4.10.1.2 [uint8_t flasher_private_t::flasher_prescale](#)

4.10.1.3 [uint8_t flasher_private_t::flashes_count](#)

4.10.1.4 [uint8_t flasher_private_t::mod_count](#)

4.10.1.5 [flasher_pattern_t* flasher_private_t::pattern](#)

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

- [flasher.h](#)

4.11 font_t Struct Reference

```
#include <font.h>
```

Data Fields

- uint8_t [pixels](#)
- uint8_t [offset](#)
- uint8_t [size](#)
- uint8_t [data](#) []

4.11.1 Field Documentation

4.11.1.1 uint8_t [font_t::data](#) []

4.11.1.2 uint8_t [font_t::offset](#)

4.11.1.3 uint8_t [font_t::pixels](#)

4.11.1.4 uint8_t [font_t::size](#)

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

- [font.h](#)

4.12 lcd_cfg_t Struct Reference

```
#include <lcd.h>
```

Data Fields

- port_t [data_port](#)
- port_bit_t [d_bit](#)
- port_t [e_port](#)
- port_bit_t [e_bit](#)
- port_t [rs_port](#)
- port_bit_t [rs_bit](#)

4.12.1 Field Documentation

4.12.1.1 port_bit_t [lcd_cfg_t::d_bit](#)

4.12.1.2 port_t [lcd_cfg_t::data_port](#)

4.12.1.3 port_bit_t [lcd_cfg_t::e_bit](#)

4.12.1.4 port_t [lcd_cfg_t::e_port](#)

4.12.1.5 port_bit_t [lcd_cfg_t::rs_bit](#)

4.12.1.6 port_t [lcd_cfg_t::rs_port](#)

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

- [lcd.h](#)

4.13 lcd_obj_t Struct Reference

```
#include <lcd.h>
```

Data Fields

- const [lcd_cfg_t](#) * [cfg](#)
- port_mask_t [e_mask](#)
- port_mask_t [rs_mask](#)
- uint8_t [data](#)

4.13.1 Field Documentation

4.13.1.1 const [lcd_cfg_t](#)* [lcd_obj_t::cfg](#)

4.13.1.2 uint8_t [lcd_obj_t::data](#)

4.13.1.3 port_mask_t [lcd_obj_t::e_mask](#)

4.13.1.4 port_mask_t [lcd_obj_t::rs_mask](#)

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

- [lcd.h](#)

4.14 led_cfg_t Struct Reference

```
#include <led.h>
```

Data Fields

- port_t [port](#)
- port_mask_t [bitmask](#)

4.14.1 Field Documentation

4.14.1.1 port_mask_t [led_cfg_t::bitmask](#)

4.14.1.2 port_t [led_cfg_t::port](#)

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

- [led.h](#)

4.15 lmatrix_port_t Struct Reference

```
#include <lmatrix.h>
```

Data Fields

- port_t [port](#)
- port_mask_t [bitmask](#)

4.15.1 Field Documentation

4.15.1.1 port_mask_t [lmatrix_port_t::bitmask](#)

4.15.1.2 port_t [lmatrix_port_t::port](#)

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

- [lmatrix.h](#)

4.16 `lmatrix_private_t` Struct Reference

```
#include <lmatrix.h>
```

Data Fields

- `lmatrix_port_t * col_port`
- `uint8_t col`
- `lmatrix_row_state_t state` [LMATRIX_COLS]

4.16.1 Field Documentation

4.16.1.1 `uint8_t lmatrix_private_t::col`

4.16.1.2 `lmatrix_port_t* lmatrix_private_t::col_port`

4.16.1.3 `lmatrix_row_state_t lmatrix_private_t::state`[LMATRIX_COLS]

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

- `lmatrix.h`

4.17 mbuttons_private_t Struct Reference

```
#include <mbuttons.h>
```

Data Fields

- [button_t](#) `buttons`
- `const` [button_cfg_t](#) `const * row_config`
- `uint8_t` `rows_num`
- `uint8_t` `cols_num`

4.17.1 Field Documentation

4.17.1.1 [button_t](#) `mbuttons_private_t::buttons`

4.17.1.2 `uint8_t` `mbuttons_private_t::cols_num`

4.17.1.3 `const` [button_cfg_t](#) `const* mbuttons_private_t::row_config`

4.17.1.4 `uint8_t` `mbuttons_private_t::rows_num`

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

- [mbuttons.h](#)

4.18 mcleds_private_t Struct Reference

```
#include <mcleds.h>
```

Data Fields

- [cleds_obj_t cleds](#)
- [colourmap_t * colourmap](#)
- [uint8_t colourmap_size](#)
- [ticker8_t primary_ticker](#)
- [mcleds_state_t * state](#)

4.18.1 Field Documentation

4.18.1.1 [cleds_obj_t mcleds_private_t::cleds](#)

4.18.1.2 [colourmap_t* mcleds_private_t::colourmap](#)

4.18.1.3 [uint8_t mcleds_private_t::colourmap_size](#)

4.18.1.4 [ticker8_t mcleds_private_t::primary_ticker](#)

4.18.1.5 [mcleds_state_t* mcleds_private_t::state](#)

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

- [mcleds.h](#)

4.19 mcleds_state_t Struct Reference

```
#include <mcleds.h>
```

Data Fields

- `uint8_t` [duty](#)

4.19.1 Field Documentation

4.19.1.1 `uint8_t mcleds_state_t::duty`

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

- [mcleds.h](#)

4.20 mmelody_private_t Struct Reference

```
#include <mmelody.h>
```

Data Fields

- [ticker_t](#) `ticker`
- `const char *` [cur](#)
- `const char *` [start](#)
- `const char *` [loop_start](#)
- `int8_t` [loop_count](#)
- `uint8_t` [note_fraction](#)
- [mmelody_speed_t](#) `speed`
- [mmelody_volume_t](#) `volume`
- `uint8_t` [octave](#)
- `void(*` [play_callback](#) `)(void *data, uint8_t note, uint8_t` [volume](#) `)`
- `void *` [play_callback_data](#)
- `uint16_t` [poll_rate](#)

4.20.1 Field Documentation

4.20.1.1 `const char*` [mmelody_private_t::cur](#)

4.20.1.2 `int8_t` [mmelody_private_t::loop_count](#)

4.20.1.3 `const char*` [mmelody_private_t::loop_start](#)

4.20.1.4 `uint8_t` [mmelody_private_t::note_fraction](#)

4.20.1.5 `uint8_t` [mmelody_private_t::octave](#)

4.20.1.6 `void(*` [mmelody_private_t::play_callback](#) `)(void *data, uint8_t note, uint8_t` [volume](#) `)`

4.20.1.7 `void*` [mmelody_private_t::play_callback_data](#)

4.20.1.8 `uint16_t` [mmelody_private_t::poll_rate](#)

4.20.1.9 [mmelody_speed_t](#) [mmelody_private_t::speed](#)

4.20.1.10 `const char*` [mmelody_private_t::start](#)

4.20.1.11 [ticker_t](#) [mmelody_private_t::ticker](#)

4.20.1.12 [mmelody_volume_t](#) [mmelody_private_t::volume](#)

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

- [mmelody.h](#)

4.21 mpwm_channel_t Struct Reference

```
#include <mpwm.h>
```

Data Fields

- `uint8_t` [duty](#)

4.21.1 Field Documentation

4.21.1.1 `uint8_t mpwm_channel_t::duty`

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

- [mpwm.h](#)

4.22 mpwm_obj_t Struct Reference

```
#include <mpwm.h>
```

Data Fields

- [uint8_t period](#)
- [uint8_t count](#)
- [mpwm_channel_t * channels](#)
- [uint8_t num_channels](#)

4.22.1 Field Documentation

4.22.1.1 [mpwm_channel_t* mpwm_obj_t::channels](#)

4.22.1.2 [uint8_t mpwm_obj_t::count](#)

4.22.1.3 [uint8_t mpwm_obj_t::num_channels](#)

4.22.1.4 [uint8_t mpwm_obj_t::period](#)

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

- [mpwm.h](#)

4.23 mtext_obj_t Struct Reference

```
#include <mtext.h>
```

Data Fields

- `const char *` [cur](#)
- `const char *` [start](#)
- `font_t *` [font](#)
- `uint8_t` [pixels](#)
- `uint8_t *` [image](#)
- `uint8_t *` [screen](#)
- `uint16_t` [poll_rate](#)
- `mtext_mode_t` [mode](#)
- `scroller_obj_t` [scroller](#)
- `uint8_t` [speed](#)
- `ticker_t` [ticker](#)

4.23.1 Field Documentation

4.23.1.1 `const char*` [mtext_obj_t::cur](#)

4.23.1.2 `font_t*` [mtext_obj_t::font](#)

4.23.1.3 `uint8_t*` [mtext_obj_t::image](#)

4.23.1.4 `mtext_mode_t` [mtext_obj_t::mode](#)

4.23.1.5 `uint8_t` [mtext_obj_t::pixels](#)

4.23.1.6 `uint16_t` [mtext_obj_t::poll_rate](#)

4.23.1.7 `uint8_t*` [mtext_obj_t::screen](#)

4.23.1.8 `scroller_obj_t` [mtext_obj_t::scroller](#)

4.23.1.9 `uint8_t` [mtext_obj_t::speed](#)

4.23.1.10 `const char*` [mtext_obj_t::start](#)

4.23.1.11 `ticker_t` [mtext_obj_t::ticker](#)

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

- [mtext.h](#)

4.24 mulwire_obj_t Struct Reference

```
#include <mulwire.h>
```

Data Fields

- [mulwire_rom_code_t rom_code](#)

4.24.1 Field Documentation

4.24.1.1 [mulwire_rom_code_t mulwire_obj_t::rom_code](#)

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

- [mulwire.h](#)

4.25 `mulwire_rom_code_t` Union Reference

```
#include <mulwire.h>
```

Data Fields

- struct {
 - uint8_t [family](#)
 - uint8_t [serial](#) [6]
 - uint8_t [crc](#)
- [fields](#)
- uint8_t [bytes](#) [8]

4.25.1 Field Documentation

4.25.1.1 uint8_t [mulwire_rom_code_t::bytes](#)[8]

4.25.1.2 uint8_t [mulwire_rom_code_t::crc](#)

4.25.1.3 uint8_t [mulwire_rom_code_t::family](#)

4.25.1.4 struct { ... } [mulwire_rom_code_t::fields](#)

4.25.1.5 uint8_t [mulwire_rom_code_t::serial](#)[6]

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

- [mulwire.h](#)

4.26 muxleds_cfg_t Struct Reference

```
#include <muxleds.h>
```

Data Fields

- port_t [port](#)
- uint8_t [bitmask](#)

4.26.1 Field Documentation

4.26.1.1 uint8_t [muxleds_cfg_t::bitmask](#)

4.26.1.2 port_t [muxleds_cfg_t::port](#)

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

- [muxleds.h](#)

4.27 muxleds_col_t Struct Reference

```
#include <muxleds.h>
```

Data Fields

- port_t [port](#)
- uint8_t [bitmask](#)
- uint8_t [row_state](#)

4.27.1 Field Documentation

4.27.1.1 uint8_t [muxleds_col_t::bitmask](#)

4.27.1.2 port_t [muxleds_col_t::port](#)

4.27.1.3 uint8_t [muxleds_col_t::row_state](#)

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

- [muxleds.h](#)

4.28 muxleds_obj_t Struct Reference

```
#include <muxleds.h>
```

Data Fields

- [muxleds_row_t rows](#) [MUXLEDS_ROWS_NUM]
- [muxleds_col_t cols](#) [MUXLEDS_COLS_NUM]
- [uint8_t col](#)
- [uint8_t row_on](#)
- [uint8_t rows_num](#)
- [uint8_t cols_num](#)

4.28.1 Field Documentation

4.28.1.1 [uint8_t muxleds_obj_t::col](#)

4.28.1.2 [muxleds_col_t muxleds_obj_t::cols](#)[MUXLEDS_COLS_NUM]

4.28.1.3 [uint8_t muxleds_obj_t::cols_num](#)

4.28.1.4 [uint8_t muxleds_obj_t::row_on](#)

4.28.1.5 [muxleds_row_t muxleds_obj_t::rows](#)[MUXLEDS_ROWS_NUM]

4.28.1.6 [uint8_t muxleds_obj_t::rows_num](#)

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

- [muxleds.h](#)

4.29 muxleds_row_t Struct Reference

```
#include <muxleds.h>
```

Data Fields

- port_t [port](#)
- uint8_t [bitmask](#)

4.29.1 Field Documentation

4.29.1.1 uint8_t [muxleds_row_t::bitmask](#)

4.29.1.2 port_t [muxleds_row_t::port](#)

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

- [muxleds.h](#)

4.30 nrf_cfg_t Struct Reference

```
#include <nrf2401.h>
```

Data Fields

- port_t [cs_port](#)
- port_bit_t [cs_bitno](#)
- port_t [ce_port](#)
- port_bit_t [ce_bitno](#)
- port_t [dr_port](#)
- port_bit_t [dr_bitno](#)

4.30.1 Field Documentation

4.30.1.1 port_bit_t [nrf_cfg_t::ce_bitno](#)

4.30.1.2 port_t [nrf_cfg_t::ce_port](#)

4.30.1.3 port_bit_t [nrf_cfg_t::cs_bitno](#)

4.30.1.4 port_t [nrf_cfg_t::cs_port](#)

4.30.1.5 port_bit_t [nrf_cfg_t::dr_bitno](#)

4.30.1.6 port_t [nrf_cfg_t::dr_port](#)

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

- [nrf2401.h](#)

4.31 nrf_config_bits_t Struct Reference

```
#include <nrf2401.h>
```

Data Fields

- uint8_t rxen:1
- uint8_t rx_ch_num:7
- uint8_t rf_pwr:2
- uint8_t xo_f:3
- uint8_t rfdr_sb:1
- uint8_t cm:1
- uint8_t rx2_en:1
- uint8_t crc_en:1
- uint8_t crc_l:1
- uint8_t addr_w:6
- rf_address_t addr_1
- rf_address_t addr_2
- uint8_t data1_w
- uint8_t data2_w

4.31.1 Field Documentation

4.31.1.1 [rf_address_t nrf_config_bits_t::addr_1](#)

4.31.1.2 [rf_address_t nrf_config_bits_t::addr_2](#)

4.31.1.3 [uint8_t nrf_config_bits_t::addr_w](#)

4.31.1.4 [uint8_t nrf_config_bits_t::cm](#)

4.31.1.5 [uint8_t nrf_config_bits_t::crc_en](#)

4.31.1.6 [uint8_t nrf_config_bits_t::crc_l](#)

4.31.1.7 [uint8_t nrf_config_bits_t::data1_w](#)

4.31.1.8 [uint8_t nrf_config_bits_t::data2_w](#)

4.31.1.9 [uint8_t nrf_config_bits_t::rf_pwr](#)

4.31.1.10 [uint8_t nrf_config_bits_t::rfd_r_sb](#)

4.31.1.11 [uint8_t nrf_config_bits_t::rx2_en](#)

4.31.1.12 [uint8_t nrf_config_bits_t::rx_ch_num](#)

4.31.1.13 [uint8_t nrf_config_bits_t::rxen](#)

4.31.1.14 [uint8_t nrf_config_bits_t::xo_f](#)

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

- [nrf2401.h](#)

4.32 nrf_config_t Union Reference

```
#include <nrf2401.h>
```

Data Fields

- [uint8_t bytes](#) [15]
- [nrf_config_bits_t bits](#)

4.32.1 Field Documentation

4.32.1.1 [nrf_config_bits_t nrf_config_t::bits](#)

4.32.1.2 [uint8_t nrf_config_t::bytes](#)[15]

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

- [nrf2401.h](#)

4.33 nrf_obj_t Struct Reference

```
#include <nrf2401.h>
```

Data Fields

- [nrf_pins_t pins](#)
- [nrf_config_t config](#)

4.33.1 Field Documentation

4.33.1.1 [nrf_config_t nrf_obj_t::config](#)

4.33.1.2 [nrf_pins_t nrf_obj_t::pins](#)

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

- [nrf2401.h](#)

4.34 nrf_pins_t Struct Reference

```
#include <nrf2401.h>
```

Data Fields

- [port_t cs_port](#)
- [uint8_t cs_bitmask](#)
- [port_t ce_port](#)
- [uint8_t ce_bitmask](#)
- [port_t dr_port](#)
- [uint8_t dr_bitmask](#)

4.34.1 Field Documentation

4.34.1.1 [uint8_t nrf_pins_t::ce_bitmask](#)

4.34.1.2 [port_t nrf_pins_t::ce_port](#)

4.34.1.3 [uint8_t nrf_pins_t::cs_bitmask](#)

4.34.1.4 [port_t nrf_pins_t::cs_port](#)

4.34.1.5 [uint8_t nrf_pins_t::dr_bitmask](#)

4.34.1.6 [port_t nrf_pins_t::dr_port](#)

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

- [nrf2401.h](#)

4.35 pga_cfg_t Struct Reference

```
#include <pga.h>
```

Data Fields

- port_t [cs_port](#)
- port_mask_t [cs_bitmask](#)

4.35.1 Field Documentation

4.35.1.1 port_mask_t [pga_cfg_t::cs_bitmask](#)

4.35.1.2 port_t [pga_cfg_t::cs_port](#)

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

- [pga.h](#)

4.36 pga_private_t Struct Reference

```
#include <pga.h>
```

Data Fields

- port_t [cs_port](#)
- port_mask_t [cs_bitmask](#)

4.36.1 Field Documentation

4.36.1.1 port_mask_t [pga_private_t::cs_bitmask](#)

4.36.1.2 port_t [pga_private_t::cs_port](#)

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

- [pga.h](#)

4.37 piezo_cfg_t Struct Reference

```
#include <piezo.h>
```

Data Fields

- port_t [port](#)
- port_mask_t [bitmask](#)

4.37.1 Field Documentation

4.37.1.1 port_mask_t [piezo_cfg_t::bitmask](#)

4.37.1.2 port_t [piezo_cfg_t::port](#)

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

- [piezo.h](#)

4.38 rf_address_t Struct Reference

```
#include <nrf2401.h>
```

Data Fields

- `uint8_t bytes` [5]

4.38.1 Field Documentation

4.38.1.1 `uint8_t rf_address_t::bytes`[5]

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

- `nrf2401.h`

4.39 rf_node_t Struct Reference

```
#include <rf.h>
```

Data Fields

- [rf_id_t id](#)
- [rf_channel_t channel](#)

4.39.1 Field Documentation

4.39.1.1 [rf_channel_t rf_node_t::channel](#)

4.39.1.2 [rf_id_t rf_node_t::id](#)

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

- [rf.h](#)

4.40 rf_probe_t Struct Reference

```
#include <rf.h>
```

Data Fields

- [rf_node_t](#) node
- [uint8_t](#) [device_id](#) [RF_DEVICE_ID_SIZE]

4.40.1 Field Documentation

4.40.1.1 [uint8_t](#) [rf_probe_t::device_id](#)[RF_DEVICE_ID_SIZE]

4.40.1.2 [rf_node_t](#) [rf_probe_t::node](#)

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

- [rf.h](#)

4.41 ring_struct Struct Reference

```
#include <ring.h>
```

Data Fields

- char * [in](#)
- char * [out](#)
- char * [top](#)
- char * [end](#)

4.41.1 Field Documentation

4.41.1.1 char* [ring_struct::end](#)

4.41.1.2 char* [ring_struct::in](#)

4.41.1.3 char* [ring_struct::out](#)

4.41.1.4 char* [ring_struct::top](#)

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

- [ring.h](#)

4.42 scroller_obj_t Struct Reference

```
#include <scroller.h>
```

Data Fields

- `uint8_t` [rows](#)
- `uint8_t` [cols](#)
- `uint8_t` [index](#)
- `scroller_dir_t` [dir](#)
- `bool` [running](#)

4.42.1 Field Documentation

4.42.1.1 `uint8_t` [scroller_obj_t::cols](#)

4.42.1.2 `scroller_dir_t` [scroller_obj_t::dir](#)

4.42.1.3 `uint8_t` [scroller_obj_t::index](#)

4.42.1.4 `uint8_t` [scroller_obj_t::rows](#)

4.42.1.5 `bool` [scroller_obj_t::running](#)

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

- [scroller.h](#)

4.43 seq_obj_t Struct Reference

```
#include <seq.h>
```

Data Fields

- const char * [str](#)
- const char * [cur](#)
- const char *(* [callback](#))(void *data, const char *[str](#))
- void * [callback_data](#)

4.43.1 Field Documentation

4.43.1.1 const char*(* [seq_obj_t::callback](#))(void *data, const char *[str](#))

4.43.1.2 void* [seq_obj_t::callback_data](#)

4.43.1.3 const char* [seq_obj_t::cur](#)

4.43.1.4 const char* [seq_obj_t::str](#)

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

- [seq.h](#)

4.44 sflash_obj_t Struct Reference

```
#include <sflash.h>
```

Data Fields

- [sflash_pattern_t pattern](#)
- [sflash_pattern_t current](#)

4.44.1 Field Documentation

4.44.1.1 [sflash_pattern_t sflash_obj_t::current](#)

4.44.1.2 [sflash_pattern_t sflash_obj_t::pattern](#)

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

- [sflash.h](#)

4.45 spi_eeprom_cfg_t Struct Reference

```
#include <s_eeprom.h>
```

Data Fields

- port_t [cs_port](#)
- port_bit_t [cs_bitno](#)

4.45.1 Field Documentation

4.45.1.1 port_bit_t [spi_eeprom_cfg_t::cs_bitno](#)

4.45.1.2 port_t [spi_eeprom_cfg_t::cs_port](#)

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

- [s_eeprom.h](#)

4.46 spi_eeprom_private_t Struct Reference

```
#include <s_eeprom.h>
```

Data Fields

- port_t [cs_port](#)
- port_mask_t [cs_bitmask](#)

4.46.1 Field Documentation

4.46.1.1 port_mask_t [spi_eeprom_private_t::cs_bitmask](#)

4.46.1.2 port_t [spi_eeprom_private_t::cs_port](#)

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

- [s_eeprom.h](#)

4.47 spwm_obj_t Struct Reference

```
#include <spwm.h>
```

Data Fields

- uint8_t [period](#)
- uint8_t [duty](#)
- uint8_t [count](#)

4.47.1 Field Documentation

4.47.1.1 uint8_t [spwm_obj_t::count](#)

4.47.1.2 uint8_t [spwm_obj_t::duty](#)

4.47.1.3 uint8_t [spwm_obj_t::period](#)

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

- [spwm.h](#)

4.48 squeaker_private_t Struct Reference

```
#include <squeaker.h>
```

Data Fields

- uint8_t [note_clock](#)
- uint8_t [note_period](#)
- uint8_t [note_duty](#)
- uint8_t [note_holdoff](#)
- const char * [start](#)
- const char * [cur](#)
- uint8_t [holdoff](#)
- uint16_t [poll_rate](#)
- const char * [loop_start](#)
- int8_t [loop_count](#)
- uint8_t [prescaler](#)
- uint8_t [note_fraction](#)
- [squeaker_speed_t](#) [speed](#)
- [squeaker_volume_t](#) [volume](#)
- [ticker8_t](#) [ticker](#)
- [squeaker_scale_t](#) * [scale_table](#)
- uint8_t [octave](#)

4.48.1 Field Documentation

- 4.48.1.1 `const char* squeaker_private_t::cur`
- 4.48.1.2 `uint8_t squeaker_private_t::holdoff`
- 4.48.1.3 `int8_t squeaker_private_t::loop_count`
- 4.48.1.4 `const char* squeaker_private_t::loop_start`
- 4.48.1.5 `uint8_t squeaker_private_t::note_clock`
- 4.48.1.6 `uint8_t squeaker_private_t::note_duty`
- 4.48.1.7 `uint8_t squeaker_private_t::note_fraction`
- 4.48.1.8 `uint8_t squeaker_private_t::note_holdoff`
- 4.48.1.9 `uint8_t squeaker_private_t::note_period`
- 4.48.1.10 `uint8_t squeaker_private_t::octave`
- 4.48.1.11 `uint16_t squeaker_private_t::poll_rate`
- 4.48.1.12 `uint8_t squeaker_private_t::prescaler`
- 4.48.1.13 `squeaker_scale_t* squeaker_private_t::scale_table`
- 4.48.1.14 `squeaker_speed_t squeaker_private_t::speed`
- 4.48.1.15 `const char* squeaker_private_t::start`
- 4.48.1.16 `ticker8_t squeaker_private_t::ticker`
- 4.48.1.17 `squeaker_volume_t squeaker_private_t::volume`

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

- [squeaker.h](#)

4.49 stext_obj_t Struct Reference

```
#include <stext.h>
```

Data Fields

- [seq_t seq](#)
- [font_t * font](#)
- [void\(* callback\)\(void *data, uint8_t pixel, bool val\)](#)
- [void * callback_data](#)
- [seq_obj_t seq_info](#)

4.49.1 Field Documentation

4.49.1.1 [void\(* stext_obj_t::callback\)\(void *data, uint8_t pixel, bool val\)](#)

4.49.1.2 [void* stext_obj_t::callback_data](#)

4.49.1.3 [font_t* stext_obj_t::font](#)

4.49.1.4 [seq_t stext_obj_t::seq](#)

4.49.1.5 [seq_obj_t stext_obj_t::seq_info](#)

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

- [stext.h](#)

4.50 ticker16_t Struct Reference

```
#include <ticker.h>
```

Data Fields

- uint16_t [period](#)
- uint16_t [clock](#)

4.50.1 Field Documentation

4.50.1.1 uint16_t [ticker16_t::clock](#)

4.50.1.2 uint16_t [ticker16_t::period](#)

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

- [ticker.h](#)

4.51 ticker8_t Struct Reference

```
#include <ticker.h>
```

Data Fields

- uint8_t [period](#)
- uint8_t [clock](#)

4.51.1 Field Documentation

4.51.1.1 uint8_t [ticker8_t::clock](#)

4.51.1.2 uint8_t [ticker8_t::period](#)

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

- [ticker.h](#)

4.52 ticker_t Struct Reference

```
#include <ticker.h>
```

Data Fields

- uint16_t [period](#)
- uint16_t [clock](#)

4.52.1 Field Documentation

4.52.1.1 uint16_t [ticker_t::clock](#)

4.52.1.2 uint16_t [ticker_t::period](#)

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

- [ticker.h](#)

4.53 time Struct Reference

```
#include <time.h>
```

Data Fields

- `uint16_t` [us_ticks](#)
- `uint16_t` [ms_ticks](#)

4.53.1 Field Documentation

4.53.1.1 `uint16_t` [time::ms_ticks](#)

4.53.1.2 `uint16_t` [time::us_ticks](#)

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

- [time.h](#)

4.54 `tweeter_private_t` Struct Reference

```
#include <tweeter.h>
```

Data Fields

- `uint8_t` [note_clock](#)
- `uint8_t` [note_period](#)
- `uint8_t` [note_duty](#)
- `uint16_t` [note_holdoff](#)
- `uint16_t` [poll_rate](#)
- `tweeter_scale_t * scale_table`

4.54.1 Field Documentation

4.54.1.1 `uint8_t` [tweeter_private_t::note_clock](#)

4.54.1.2 `uint8_t` [tweeter_private_t::note_duty](#)

4.54.1.3 `uint16_t` [tweeter_private_t::note_holdoff](#)

4.54.1.4 `uint8_t` [tweeter_private_t::note_period](#)

4.54.1.5 `uint16_t` [tweeter_private_t::poll_rate](#)

4.54.1.6 [tweeter_scale_t*](#) [tweeter_private_t::scale_table](#)

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

- [tweeter.h](#)

4.55 u1wire_enumerate_t Struct Reference

```
#include <u1wire_enumerate.h>
```

Data Fields

- [u1wire_obj_t device](#)
- [u1wire_state_t state](#)

4.55.1 Field Documentation

4.55.1.1 [u1wire_obj_t u1wire_enumerate_t::device](#)

4.55.1.2 [u1wire_state_t u1wire_enumerate_t::state](#)

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

- [u1wire_enumerate.h](#)

4.56 u1wire_obj_t Struct Reference

```
#include <ulwire.h>
```

Data Fields

- [u1wire_rom_code_t rom_code](#)

4.56.1 Field Documentation

4.56.1.1 [u1wire_rom_code_t u1wire_obj_t::rom_code](#)

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

- [ulwire.h](#)

4.57 u1wire_rom_code_t Union Reference

```
#include <ulwire.h>
```

Data Fields

- struct {
 - uint8_t [family](#)
 - uint8_t [serial](#) [6]
 - uint8_t [crc](#)
- [fields](#)
- uint8_t [bytes](#) [8]

4.57.1 Field Documentation

4.57.1.1 uint8_t [u1wire_rom_code_t::bytes](#)[8]

4.57.1.2 uint8_t [u1wire_rom_code_t::crc](#)

4.57.1.3 uint8_t [u1wire_rom_code_t::family](#)

4.57.1.4 struct { ... } [u1wire_rom_code_t::fields](#)

4.57.1.5 uint8_t [u1wire_rom_code_t::serial](#)[6]

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

- [ulwire.h](#)

4.58 u1wire_state_t Struct Reference

```
#include <ulwire_enumerate.h>
```

Data Fields

- [int8_t last_discrepancy](#)
- [int8_t last_device](#)
- [int8_t last_family_discrepancy](#)

4.58.1 Field Documentation

4.58.1.1 [int8_t u1wire_state_t::last_device](#)

4.58.1.2 [int8_t u1wire_state_t::last_discrepancy](#)

4.58.1.3 [int8_t u1wire_state_t::last_family_discrepancy](#)

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

- [u1wire_discover.c](#)
- [u1wire_enumerate.h](#)

4.59 usart_dev_struct Struct Reference

Data Fields

- `int8_t(* putc)(char ch)`
- `int8_t(* getc)(void)`
- `bool(* read_ready_p)(void)`
- `bool(* write_ready_p)(void)`
- `bool(* write_finished_p)(void)`

4.59.1 Field Documentation

4.59.1.1 `int8_t(* usart_dev_struct::getc)(void)`

4.59.1.2 `int8_t(* usart_dev_struct::putc)(char ch)`

4.59.1.3 `bool(* usart_dev_struct::read_ready_p)(void)`

4.59.1.4 `bool(* usart_dev_struct::write_finished_p)(void)`

4.59.1.5 `bool(* usart_dev_struct::write_ready_p)(void)`

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

- [usart.c](#)

Chapter 5

mmculib File Documentation

5.1 ads7870.c File Reference

```
#include "config.h"
#include "port.h"
#include "spi.h"
#include "spi_adc.h"
```

Enumerations

- enum { [ADS7870_CONVERT](#) = 0x80, [ADS7870_REG_READ](#) = 0x40, [ADS7870_REG_WRITE](#) = 0x00, [ADS7870_REG_16BIT](#) = 0x20 }
- enum [ads7870_register_t](#) {
[ADS7870_RESULTLO](#), [ADS7870_RESULTHI](#), [ADS7870_PGAVALID](#), [ADS7870_ADCTRL](#),
[ADS7870_GAINMUX](#), [ADS7870_DIGIOSTATE](#), [ADS7870_DIGIOCTRL](#), [ADS7870_REFOSC](#) }
- enum [ads7870_pga_gain_t](#) {
[ADS7870_PGA_GAIN_1](#) = 0, [ADS7870_PGA_GAIN_2](#) = 1, [ADS7870_PGA_GAIN_4](#) = 2,
[ADS7870_PGA_GAIN_5](#) = 3,
[ADS7870_PGA_GAIN_8](#) = 4, [ADS7870_PGA_GAIN_10](#) = 5, [ADS7870_PGA_GAIN_16](#) = 6,
[ADS7870_PGA_GAIN_20](#) = 7 }
- enum {
[ADS7870_REFOSC_OSCR](#) = BIT (5), [ADS7870_REFOSC_OSCE](#) = BIT (4), [ADS7870_-](#)
[REFOSC_REFE](#) = BIT (3), [ADS7870_REFOSC_BUFE](#) = BIT (2),
[ADS7870_REFOSC_R2V](#) = BIT (1), [ADS7870_REFOSC_RBE](#) = BIT (0) }
- enum { [ADS7870_GAINMUX_CNVBSY](#) = BIT (7) }

Functions

- static void [ads7870_chip_select](#) (void)
- static void [ads7870_chip_deselect](#) (void)
- static void [ads7870_reg_write](#) ([ads7870_register_t](#) reg, uint8_t val)
- static uint8_t [ads7870_reg_read](#) ([ads7870_register_t](#) reg)

- int16_t [ads7870_read](#) (void)
- static int8_t [ads7870_read_ready_p](#) (void)
- void [ads7870_channel_start](#) (uint8_t channel, [spi_adc_mode_t](#) mode)
- int16_t [ads7870_channel_convert](#) (uint8_t channel, [spi_adc_mode_t](#) mode)
- void [ads7870_init](#) (void)

5.1.1 Enumeration Type Documentation

5.1.1.1 anonymous enum

Enumerator:

ADS7870_CONVERT
ADS7870_REG_READ
ADS7870_REG_WRITE
ADS7870_REG_16BIT

5.1.1.2 anonymous enum

Enumerator:

ADS7870_REFOSC_OSCR
ADS7870_REFOSC_OSCE
ADS7870_REFOSC_REFE
ADS7870_REFOSC_BUFE
ADS7870_REFOSC_R2V
ADS7870_REFOSC_RBE

5.1.1.3 anonymous enum

Enumerator:

ADS7870_GAINMUX_CNVBSY

5.1.1.4 enum [ads7870_pga_gain_t](#)

Enumerator:

ADS7870_PGA_GAIN_1
ADS7870_PGA_GAIN_2
ADS7870_PGA_GAIN_4
ADS7870_PGA_GAIN_5
ADS7870_PGA_GAIN_8
ADS7870_PGA_GAIN_10
ADS7870_PGA_GAIN_16
ADS7870_PGA_GAIN_20

5.1.1.5 enum [ads7870_register_t](#)

Enumerator:

ADS7870_RESULTLO
ADS7870_RESULTHI
ADS7870_PGAVAILD
ADS7870_ADCTRL
ADS7870_GAINMUX
ADS7870_DIGIOSTATE
ADS7870_DIGIOCTRL
ADS7870_REFOSC

5.1.2 Function Documentation

5.1.2.1 `int16_t ads7870_channel_convert (uint8_t channel, spi_adc_mode_t mode)`

5.1.2.2 `void ads7870_channel_start (uint8_t channel, spi_adc_mode_t mode)`

5.1.2.3 `static void ads7870_chip_deselect (void) [static]`

5.1.2.4 `static void ads7870_chip_select (void) [static]`

5.1.2.5 `void ads7870_init (void)`

5.1.2.6 `int16_t ads7870_read (void)`

5.1.2.7 `static int8_t ads7870_read_ready_p (void) [static]`

5.1.2.8 `static uint8_t ads7870_reg_read (ads7870_register_t reg) [static]`

5.1.2.9 `static void ads7870_reg_write (ads7870_register_t reg, uint8_t val) [static]`

5.2 ads7870.h File Reference

ADS7870 SPI ADC.

```
#include "config.h"
```

Functions

- void [ads7870_init](#) (void)
- int16_t [ads7870_read](#) (void)
- int16_t [ads7870_convert](#) (void)
- void [ads7870_channel_start](#) (uint8_t channel, [spi_adc_mode_t](#) mode)
- int16_t [ads7870_channel_convert](#) (uint8_t channel, [spi_adc_mode_t](#) mode)

5.2.1 Detailed Description

ADS7870 SPI ADC.

Author:

M. P. Hayes, UCECE

Date:

09 August 2007

5.2.2 Function Documentation

5.2.2.1 int16_t [ads7870_channel_convert](#) (uint8_t *channel*, [spi_adc_mode_t](#) *mode*)

5.2.2.2 void [ads7870_channel_start](#) (uint8_t *channel*, [spi_adc_mode_t](#) *mode*)

5.2.2.3 int16_t [ads7870_convert](#) (void)

5.2.2.4 void [ads7870_init](#) (void)

5.2.2.5 int16_t [ads7870_read](#) (void)

5.3 ads8325.c File Reference

```
#include "config.h"
#include "port.h"
#include "spi.h"
```

Functions

- static void [ads8325_chip_select](#) (void)
- static void [ads8325_chip_deselect](#) (void)
- void [ads8325_init](#) (void)
- int16_t [ads8325_read](#) (void)

5.3.1 Function Documentation

5.3.1.1 static void [ads8325_chip_deselect](#) (void) [static]

5.3.1.2 static void [ads8325_chip_select](#) (void) [static]

5.3.1.3 void [ads8325_init](#) (void)

5.3.1.4 int16_t [ads8325_read](#) (void)

5.4 ads8327.c File Reference

```
#include "config.h"
#include "port.h"
#include "spi.h"
#include "spi_adc.h"
```

Functions

- static void [ads8327_chip_select](#) (void)
- static void [ads8327_chip_deselect](#) (void)
- static uint16_t [ads8327_command](#) (uint16_t val)
- void [ads8327_init](#) (void)
- int16_t [ads8327_read](#) (void)
- int16_t [ads8327_convert](#) (void)

5.4.1 Function Documentation

5.4.1.1 static void [ads8327_chip_deselect](#) (void) [static]

5.4.1.2 static void [ads8327_chip_select](#) (void) [static]

5.4.1.3 static uint16_t [ads8327_command](#) (uint16_t *val*) [static]

5.4.1.4 int16_t [ads8327_convert](#) (void)

5.4.1.5 void [ads8327_init](#) (void)

5.4.1.6 int16_t [ads8327_read](#) (void)

5.5 ads8327.h File Reference

ADS8327 SPI ADC.

```
#include "config.h"
```

Functions

- void [ads8327_init](#) (void)
- int16_t [ads8327_convert](#) (void)
- int16_t [ads8327_read](#) (void)

5.5.1 Detailed Description

ADS8327 SPI ADC.

Author:

M. P. Hayes, UCECE

Date:

09 August 2007

5.5.2 Function Documentation

5.5.2.1 int16_t ads8327_convert (void)

5.5.2.2 void ads8327_init (void)

5.5.2.3 int16_t ads8327_read (void)

5.6 biseq.c File Reference

Bidirectional sequencer.

```
#include <limits.h>
#include "biseq.h"
```

Functions

- [biseq_t biseq_init](#) ([biseq_obj_t](#) *dev, [int8_t](#)(*callback)(void *data, char *str), void *callback_data)
- void [biseq_set](#) ([biseq_t](#) biseq, char *str)
- char * [biseq_get](#) ([biseq_t](#) biseq)
- void [biseq_mode_set](#) ([biseq_t](#) biseq, [biseq_mode_t](#) mode)
- [biseq_mode_t](#) [biseq_mode_get](#) ([biseq_t](#) biseq)
- [int8_t](#) [biseq_update](#) ([biseq_t](#) biseq)

5.6.1 Detailed Description

Bidirectional sequencer.

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.6.2 Function Documentation

5.6.2.1 char* [biseq_get](#) ([biseq_t](#) biseq)

5.6.2.2 [biseq_t](#) [biseq_init](#) ([biseq_obj_t](#) * dev, [int8_t](#)(*)(void *data, char *str) callback, void * callback_data)

5.6.2.3 [biseq_mode_t](#) [biseq_mode_get](#) ([biseq_t](#) biseq)

5.6.2.4 void [biseq_mode_set](#) ([biseq_t](#) biseq, [biseq_mode_t](#) mode)

5.6.2.5 void [biseq_set](#) ([biseq_t](#) biseq, char * str)

5.6.2.6 [int8_t](#) [biseq_update](#) ([biseq_t](#) biseq)

5.7 biseq.h File Reference

```
#include "config.h"
```

Data Structures

- struct [biseq_obj_t](#)

Typedefs

- typedef biseq_struct * [biseq_t](#)

Enumerations

- enum [biseq_mode_t](#) { [BISEQ_MODE_NORMAL](#), [BISEQ_MODE_CYCLE](#), [BISEQ_MODE_NUM](#) }

Functions

- [biseq_t biseq_init](#) ([biseq_obj_t](#) *dev, int8_t(*callback)(void *data, char *str), void *callback_data)
- void [biseq_set](#) ([biseq_t](#) biseq, char *str)
- char * [biseq_get](#) ([biseq_t](#) biseq)
- void [biseq_mode_set](#) ([biseq_t](#) biseq, [biseq_mode_t](#) mode)
- [biseq_mode_t biseq_mode_get](#) ([biseq_t](#) biseq)
- int8_t [biseq_update](#) ([biseq_t](#) biseq)

5.7.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.7.2 Typedef Documentation

5.7.2.1 typedef struct biseq_struct* [biseq_t](#)

5.7.3 Enumeration Type Documentation

5.7.3.1 enum [biseq_mode_t](#)

Enumerator:

[BISEQ_MODE_NORMAL](#)

[BISEQ_MODE_CYCLE](#)

[BISEQ_MODE_NUM](#)

5.7.4 Function Documentation

5.7.4.1 `char* biseq_get (biseq_t biseq)`

5.7.4.2 `biseq_t biseq_init (biseq_obj_t * dev, int8_t(*) (void *data, char *str) callback, void * callback_data)`

5.7.4.3 `biseq_mode_t biseq_mode_get (biseq_t biseq)`

5.7.4.4 `void biseq_mode_set (biseq_t biseq, biseq_mode_t mode)`

5.7.4.5 `void biseq_set (biseq_t biseq, char * str)`

5.7.4.6 `int8_t biseq_update (biseq_t biseq)`

5.8 bits.h File Reference

Defines

- #define `BITS_MASK`(first, last) `((1 << ((last) + 1)) - (1 << (first)))`
- #define `BITS_CLR`(reg, first, last) `((reg) &= BITS_MASK (first, last))`
- #define `BITS_SET`(reg, first, last) `((reg) |= BITS_MASK (first, last))`
- #define `BITS_EXTRACT`(reg, first, last) `((reg) & BITS_MASK (first, last)) >> (first)`
- #define `BITS_INSERT`(reg, val, first, last)

5.8.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.8.2 Define Documentation

5.8.2.1 #define `BITS_CLR`(reg, first, last) `((reg) &= BITS_MASK (first, last))`

5.8.2.2 #define `BITS_EXTRACT`(reg, first, last) `((reg) & BITS_MASK (first, last)) >> (first)`

5.8.2.3 #define `BITS_INSERT`(reg, val, first, last)

Value:

```
(reg) = ((reg) & ~BITS_MASK (first, last)) \
        | ((val) & BITS_MASK (0, last - first)) << (first)
```

5.8.2.4 #define `BITS_MASK`(first, last) `((1 << ((last) + 1)) - (1 << (first)))`

5.8.2.5 #define `BITS_SET`(reg, first, last) `((reg) |= BITS_MASK (first, last))`

5.9 busart.c File Reference

Buffered USART implementation.

```
#include "ring.h"
#include "busart.h"
#include "peripherals.h"
#include <string.h>
```

Data Structures

- struct [busart_dev_struct](#)

Defines

- #define [BUSART0_ENABLE](#) (USART_NUM >= 1)
- #define [BUSART1_ENABLE](#) (USART_NUM >= 2)

Functions

- [busart_t busart_init](#) (uint8_t channel, uint16_t baud_divisor, char *tx_buffer, [ring_size_t](#) tx_size, char *rx_buffer, [ring_size_t](#) rx_size)
- [ring_size_t busart_write](#) ([busart_t](#) busart, const void *data, [ring_size_t](#) size)
- [ring_size_t busart_write_block](#) ([busart_t](#) busart, const void *data, [ring_size_t](#) size)
- [ring_size_t busart_read](#) ([busart_t](#) busart, void *data, [ring_size_t](#) size)
- [ring_size_t busart_read_block](#) ([busart_t](#) busart, void *data, [ring_size_t](#) size)
- [ring_size_t busart_read_num](#) ([busart_t](#) busart)
- [ring_size_t busart_write_num](#) ([busart_t](#) busart)
- bool [busart_read_ready_p](#) ([busart_t](#) busart)
- bool [busart_write_ready_p](#) ([busart_t](#) busart)
- bool [busart_write_finished_p](#) ([busart_t](#) busart)
- int8_t [busart_getc](#) ([busart_t](#) busart)
- int8_t [busart_putc](#) ([busart_t](#) busart, char ch)
- int8_t [busart_puts](#) ([busart_t](#) busart, const char *str)

5.9.1 Detailed Description

Buffered USART implementation.

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.9.2 Define Documentation

5.9.2.1 `#define BUSART0_ENABLE (USART_NUM >= 1)`

5.9.2.2 `#define BUSART1_ENABLE (USART_NUM >= 2)`

5.9.3 Function Documentation

5.9.3.1 `int8_t busart_getc (busart_t busart)`

5.9.3.2 `busart_t busart_init (uint8_t channel, uint16_t baud_divisor, char * tx_buffer, ring_size_t tx_size, char * rx_buffer, ring_size_t rx_size)`

5.9.3.3 `int8_t busart_putc (busart_t busart, char ch)`

5.9.3.4 `int8_t busart_puts (busart_t busart, const char * str)`

5.9.3.5 `ring_size_t busart_read (busart_t busart, void * data, ring_size_t size)`

5.9.3.6 `ring_size_t busart_read_block (busart_t busart, void * data, ring_size_t size)`

5.9.3.7 `ring_size_t busart_read_num (busart_t busart)`

5.9.3.8 `bool busart_read_ready_p (busart_t busart)`

5.9.3.9 `ring_size_t busart_write (busart_t busart, const void * data, ring_size_t size)`

5.9.3.10 `ring_size_t busart_write_block (busart_t busart, const void * data, ring_size_t size)`

5.9.3.11 `bool busart_write_finished_p (busart_t busart)`

5.9.3.12 `ring_size_t busart_write_num (busart_t busart)`

5.9.3.13 `bool busart_write_ready_p (busart_t busart)`

5.10 busart.h File Reference

Buffered USART interface.

```
#include "config.h"
#include "ring.h"
#include "usart0.h"
```

Defines

- #define [BUSART_BAUD_DIVISOR](#)(BAUD_RATE) USART0_BAUD_DIVISOR(BAUD_RATE)

Typedefs

- typedef [busart_dev_struct](#) [busart_dev_t](#)
- typedef [busart_dev_t](#) * [busart_t](#)

Functions

- [busart_t](#) [busart_init](#) (uint8_t channel, uint16_t baud_divisor, char *tx_buffer, [ring_size_t](#) tx_size, char *rx_buffer, [ring_size_t](#) rx_size)
- [ring_size_t](#) [busart_read](#) ([busart_t](#) busart, void *data, [ring_size_t](#) size)
- [ring_size_t](#) [busart_read_block](#) ([busart_t](#) busart, void *data, [ring_size_t](#) size)
- [ring_size_t](#) [busart_write](#) ([busart_t](#) busart, const void *data, [ring_size_t](#) size)
- [ring_size_t](#) [busart_write_block](#) ([busart_t](#) busart, const void *data, [ring_size_t](#) size)
- [ring_size_t](#) [busart_read_num](#) ([busart_t](#) busart)
- [ring_size_t](#) [busart_write_num](#) ([busart_t](#) busart)
- bool [busart_read_ready_p](#) ([busart_t](#) busart)
- bool [busart_write_ready_p](#) ([busart_t](#) busart)
- bool [busart_write_finished_p](#) ([busart_t](#) busart)
- int8_t [busart_getc](#) ([busart_t](#) busart)
- int8_t [busart_putc](#) ([busart_t](#) busart, char ch)
- int8_t [busart_puts](#) ([busart_t](#) busart, const char *str)

5.10.1 Detailed Description

Buffered USART interface.

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.10.2 Define Documentation

5.10.2.1 `#define BUSART_BAUD_DIVISOR(BAUD_RATE) USART0_BAUD_DIVISOR(BAUD_RATE)`

5.10.3 Typedef Documentation

5.10.3.1 `typedef struct busart_dev_struct busart_dev_t`

5.10.3.2 `typedef busart_dev_t* busart_t`

5.10.4 Function Documentation

5.10.4.1 `int8_t busart_getc (busart_t busart)`

5.10.4.2 `busart_t busart_init (uint8_t channel, uint16_t baud_divisor, char * tx_buffer, ring_size_t tx_size, char * rx_buffer, ring_size_t rx_size)`

5.10.4.3 `int8_t busart_putc (busart_t busart, char ch)`

5.10.4.4 `int8_t busart_puts (busart_t busart, const char * str)`

5.10.4.5 `ring_size_t busart_read (busart_t busart, void * data, ring_size_t size)`

5.10.4.6 `ring_size_t busart_read_block (busart_t busart, void * data, ring_size_t size)`

5.10.4.7 `ring_size_t busart_read_num (busart_t busart)`

5.10.4.8 `bool busart_read_ready_p (busart_t busart)`

5.10.4.9 `ring_size_t busart_write (busart_t busart, const void * data, ring_size_t size)`

5.10.4.10 `ring_size_t busart_write_block (busart_t busart, const void * data, ring_size_t size)`

5.10.4.11 `bool busart_write_finished_p (busart_t busart)`

5.10.4.12 `ring_size_t busart_write_num (busart_t busart)`

5.10.4.13 `bool busart_write_ready_p (busart_t busart)`

5.11 button.c File Reference

```
#include "button.h"
#include "delay.h"
```

Functions

- void [button_poll_count_set](#) (uint8_t poll_count)
- [button_t](#) [button_init](#) ([button_obj_t](#) *info, const [button_cfg_t](#) *cfg)
- [button_state_t](#) [button_debounce](#) ([button_t](#) button, bool pressed)
- [button_state_t](#) [button_poll](#) ([button_t](#) button)
- bool [button_held_p](#) ([button_t](#) button, uint8_t hold_count)
- bool [button_hold_released_p](#) ([button_t](#) button, uint8_t hold_count)

Variables

- static uint8_t [button_poll_count](#)

5.11.1 Detailed Description

Author:

M.P. Hayes

Date:

16 Feb 2003

Description: This debounces pushbuttons and switches using a polled wait-and-see method implemented with a state machine. It assumes that a pushed button gives a logic low on the input port.

5.11.2 Function Documentation

5.11.2.1 [button_state_t](#) [button_debounce](#) ([button_t](#) button, bool *pressed*)

Poll the specified button and return its debounced state.

5.11.2.2 bool [button_held_p](#) ([button_t](#) button, uint8_t *hold_count*)

Return true if button held for hold_count.

5.11.2.3 bool [button_hold_released_p](#) ([button_t](#) button, uint8_t *hold_count*)

Return true if button held for hold_count has been released.

5.11.2.4 [button_t](#) [button_init](#) ([button_obj_t](#) *info, const [button_cfg_t](#) *cfg)

Create a new button object.

5.11.2.5 `button_state_t` button_poll (`button_t` *button*)

Poll the specified button and return its debounced state.

5.11.2.6 `void` button_poll_count_set (`uint8_t` *poll_count*)

Set the number of polls required for the debounce period.

5.11.3 Variable Documentation**5.11.3.1** `uint8_t` `button_poll_count` [`static`]

5.12 button.h File Reference

Button polling and debouncing.

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [button_cfg_t](#)
- struct [button_private_t](#)

Defines

- #define [BUTTON_DEBOUNCE_RATE](#) (1000 / [BUTTON_DEBOUNCE_MS](#))
- #define [BUTTON_POLL_COUNT](#)([POLL_RATE](#)) (([POLL_RATE](#)) / [BUTTON_DEBOUNCE_RATE](#))
- #define [BUTTON_CFG](#)([PORT](#), [PORTBIT](#)) {([PORT](#)), [BIT](#) ([PORTBIT](#))}

Typedefs

- typedef [button_private_t](#) [button_obj_t](#)
- typedef [button_obj_t](#) * [button_t](#)

Enumerations

- enum { [BUTTON_DEBOUNCE_MS](#) = 50 }
- enum [button_state_t](#) { [BUTTON_STATE_UP](#), [BUTTON_STATE_DOWN](#), [BUTTON_STATE_PUSHED](#), [BUTTON_STATE_RELEASED](#) }

Functions

- void [button_poll_count_set](#) (uint8_t poll_count)
- [button_t](#) [button_init](#) ([button_obj_t](#) *info, const [button_cfg_t](#) *cfg)
- [button_state_t](#) [button_poll](#) ([button_t](#) button)
- bool [button_held_p](#) ([button_t](#) button, uint8_t hold_count)
- bool [button_hold_released_p](#) ([button_t](#) button, uint8_t hold_count)
- static [button_state_t](#) [button_state_get](#) ([button_t](#) button)
- static uint8_t [button_hold_count_get](#) ([button_t](#) button)
- static bool [button_pushed_p](#) ([button_t](#) button)
- static bool [button_released_p](#) ([button_t](#) button)
- static bool [button_down_p](#) ([button_t](#) button)
- static bool [button_pressed_p](#) ([button_t](#) button)

5.12.1 Detailed Description

Button polling and debouncing.

Author:

M. P. Hayes, UCECE

Date:

15 Feb 2003

5.12.2 Define Documentation

5.12.2.1 **#define** `BUTTON_CFG(PORT, PORTBIT)` `{(PORT), BIT (PORTBIT)}`

5.12.2.2 **#define** `BUTTON_DEBOUNCE_RATE` `(1000 / BUTTON_DEBOUNCE_MS)`

5.12.2.3 **#define** `BUTTON_POLL_COUNT(POLL_RATE)` `((POLL_RATE) /
BUTTON_DEBOUNCE_RATE)`

5.12.3 Typedef Documentation

5.12.3.1 **typedef** `button_private_t` `button_obj_t`

5.12.3.2 **typedef** `button_obj_t*` `button_t`

5.12.4 Enumeration Type Documentation

5.12.4.1 anonymous enum

Debounce period in milliseconds.

Enumerator:

`BUTTON_DEBOUNCE_MS`

5.12.4.2 **enum** `button_state_t`

Button states.

Enumerator:

`BUTTON_STATE_UP`

`BUTTON_STATE_DOWN`

`BUTTON_STATE_PUSHED`

`BUTTON_STATE_RELEASED`

5.12.5 Function Documentation

5.12.5.1 **static bool** `button_down_p` (`button_t` *button*) `[static]`

Return true if button down (pressed).

5.12.5.2 bool button_held_p ([button_t](#) *button*, uint8_t *hold_count*)

Return true if button held for *hold_count*.

5.12.5.3 static uint8_t button_hold_count_get ([button_t](#) *button*) [static]

Return duration button held (in polling periods).

5.12.5.4 bool button_hold_released_p ([button_t](#) *button*, uint8_t *hold_count*)

Return true if button held for *hold_count* has been released.

5.12.5.5 [button_t](#) button_init ([button_obj_t](#) * *info*, const [button_cfg_t](#) * *cfg*)

Create a new button object.

5.12.5.6 [button_state_t](#) button_poll ([button_t](#) *button*)

Poll the specified button and return its debounced state.

5.12.5.7 void button_poll_count_set (uint8_t *poll_count*)

Set the number of polls required for the debounce period.

5.12.5.8 static bool button_pressed_p ([button_t](#) *button*) [static]

Return true if button possibly pressed. Note this is not debounced. For a debounced version use `button_pushed_p`.

5.12.5.9 static bool button_pushed_p ([button_t](#) *button*) [static]

Return true if button pushed.

5.12.5.10 static bool button_released_p ([button_t](#) *button*) [static]

Return true if button released.

5.12.5.11 static [button_state_t](#) button_state_get ([button_t](#) *button*) [static]

Return button state.

5.13 buttons.c File Reference

```
#include "buttons.h"
```

Functions

- void `buttons_init` (`buttons_obj_t` *buttons, `button_obj_t` *button_objs, const `button_cfg_t` *config, `uint8_t` buttons_num, `uint8_t` poll_count)
- void `buttons_poll` (`buttons_t` buttons)
- bool `buttons_any_pushed_p` (`buttons_t` buttons)

5.13.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.13.2 Function Documentation

5.13.2.1 bool buttons_any_pushed_p (`buttons_t` buttons)

Return true if any buttons pushed.

5.13.2.2 void buttons_init (`buttons_obj_t` * buttons, `button_obj_t` * buttons_info, const `button_cfg_t` * config, `uint8_t` buttons_num, `uint8_t` poll_count)

Initialise multiple buttons.

5.13.2.3 void buttons_poll (`buttons_t` buttons)

Poll multiple buttons.

5.14 buttons.h File Reference

Multiple button polling and debouncing.

```
#include "button.h"
```

Data Structures

- struct [buttons_private_t](#)

Typedefs

- typedef [buttons_private_t](#) [buttons_obj_t](#)
- typedef [buttons_obj_t](#) * [buttons_t](#)

Functions

- void [buttons_init](#) ([buttons_obj_t](#) *buttons, [button_obj_t](#) *buttons_info, const [button_cfg_t](#) *config, uint8_t buttons_num, uint8_t poll_count)
- void [buttons_poll](#) ([buttons_t](#) buttons)
- bool [buttons_any_pushed_p](#) ([buttons_t](#) buttons)
- static bool [buttons_pushed_p](#) ([buttons_t](#) buttons, uint8_t id)
- static bool [buttons_released_p](#) ([buttons_t](#) buttons, uint8_t id)
- static bool [buttons_held_p](#) ([buttons_t](#) buttons, uint8_t id, uint8_t hold_count)
- static bool [buttons_hold_released_p](#) ([buttons_t](#) buttons, uint8_t id, uint8_t hold_count)

5.14.1 Detailed Description

Multiple button polling and debouncing.

Author:

M. P. Hayes, UCECE

Date:

17 Nov 2006

5.14.2 Typedef Documentation

5.14.2.1 typedef [buttons_private_t](#) [buttons_obj_t](#)

5.14.2.2 typedef [buttons_obj_t](#)* [buttons_t](#)

5.14.3 Function Documentation

5.14.3.1 bool [buttons_any_pushed_p](#) ([buttons_t](#) buttons)

Return true if any buttons pushed.

5.14.3.2 static bool buttons_held_p (buttons_t buttons, uint8_t id, uint8_t hold_count) [static]

Return true if selected button held for hold_count.

5.14.3.3 static bool buttons_hold_released_p (buttons_t buttons, uint8_t id, uint8_t hold_count) [static]

Return true if selected button held for hold_count is released.

5.14.3.4 void buttons_init (buttons_obj_t * buttons, button_obj_t * buttons_info, const button_cfg_t * config, uint8_t buttons_num, uint8_t poll_count)

Initialise multiple buttons.

5.14.3.5 void buttons_poll (buttons_t buttons)

Poll multiple buttons.

5.14.3.6 static bool buttons_pushed_p (buttons_t buttons, uint8_t id) [static]

Return true if selected button pushed.

5.14.3.7 static bool buttons_released_p (buttons_t buttons, uint8_t id) [static]

Return true if selected button released.

5.15 chaser.c File Reference

```
#include <limits.h>
#include "chaser.h"
#include "flasher.h"
#include "font.h"
```

Functions

- `chaser_t chaser_init` (`chaser_obj_t *dev`, `flasher_t *flashers`, `uint8_t flasher_num`)
- `void chaser_sequence_set` (`chaser_t chaser`, `chaser_sequence_t seq`)
- `void chaser_mode_set` (`chaser_t chaser`, `chaser_mode_t mode`)
- `static void chaser_pixel_set` (`void *data`, `uint8_t pixel`, `bool val`)
- `int8_t chaser_update` (`chaser_t chaser`)

5.15.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.15.2 Function Documentation

5.15.2.1 `chaser_t chaser_init` (`chaser_obj_t *dev`, `flasher_t *flashers`, `uint8_t flasher_num`)

5.15.2.2 `void chaser_mode_set` (`chaser_t chaser`, `chaser_mode_t mode`)

5.15.2.3 `static void chaser_pixel_set` (`void *data`, `uint8_t pixel`, `bool val`) [static]

5.15.2.4 `void chaser_sequence_set` (`chaser_t chaser`, `chaser_sequence_t seq`)

5.15.2.5 `int8_t chaser_update` (`chaser_t chaser`)

5.16 chaser.h File Reference

```
#include "config.h"
#include "flasher.h"
#include "font.h"
```

Data Structures

- struct [chaser_private_t](#)

Typedefs

- typedef char [chaser_font_index_t](#)
- typedef [chaser_font_index_t](#) * [chaser_sequence_t](#)
- typedef [chaser_private_t](#) [chaser_obj_t](#)
- typedef [chaser_obj_t](#) * [chaser_t](#)

Enumerations

- enum [chaser_mode_t](#) {
 [CHASER_MODE_NORMAL](#), [CHASER_MODE_CYCLE](#), [CHASER_MODE_INVERT](#),
 [CHASER_MODE_CYCLE_INVERT](#),
 [CHASER_MODE_NUM](#) }

Functions

- [chaser_t](#) [chaser_init](#) ([chaser_obj_t](#) *dev, [flasher_t](#) *flashers, uint8_t flasher_num)
- void [chaser_sequence_set](#) ([chaser_t](#) chaser, [chaser_sequence_t](#) seq)
- void [chaser_mode_set](#) ([chaser_t](#) chaser, [chaser_mode_t](#) mode)
- int8_t [chaser_update](#) ([chaser_t](#) chaser)
- static void [chaser_patterns_set](#) ([chaser_t](#) chaser, [flasher_pattern_t](#) *on_pattern, [flasher_pattern_t](#) *off_pattern)
- static [chaser_sequence_t](#) [chaser_sequence_get](#) ([chaser_t](#) chaser)
- static void [chaser_font_set](#) ([chaser_t](#) chaser, [font_t](#) *font)

5.16.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.16.2 Typedef Documentation

5.16.2.1 typedef char [chaser_font_index_t](#)

5.16.2.2 typedef [chaser_private_t](#) [chaser_obj_t](#)

5.16.2.3 typedef [chaser_font_index_t](#)* [chaser_sequence_t](#)

5.16.2.4 typedef [chaser_obj_t](#)* [chaser_t](#)

5.16.3 Enumeration Type Documentation

5.16.3.1 enum [chaser_mode_t](#)

Enumerator:

CHASER_MODE_NORMAL

CHASER_MODE_CYCLE

CHASER_MODE_INVERT

CHASER_MODE_CYCLE_INVERT

CHASER_MODE_NUM

5.16.4 Function Documentation

5.16.4.1 static void [chaser_font_set](#) ([chaser_t](#) *chaser*, [font_t](#)* *font*) [static]

5.16.4.2 [chaser_t](#) [chaser_init](#) ([chaser_obj_t](#)* *dev*, [flasher_t](#)* *flashers*, [uint8_t](#) *flasher_num*)

5.16.4.3 void [chaser_mode_set](#) ([chaser_t](#) *chaser*, [chaser_mode_t](#) *mode*)

5.16.4.4 static void [chaser_patterns_set](#) ([chaser_t](#) *chaser*, [flasher_pattern_t](#)* *on_pattern*, [flasher_pattern_t](#)* *off_pattern*) [static]

5.16.4.5 static [chaser_sequence_t](#) [chaser_sequence_get](#) ([chaser_t](#) *chaser*) [static]

5.16.4.6 void [chaser_sequence_set](#) ([chaser_t](#) *chaser*, [chaser_sequence_t](#) *seq*)

5.16.4.7 [int8_t](#) [chaser_update](#) ([chaser_t](#) *chaser*)

5.17 cleds.c File Reference

```
#include "cleds.h"
#include "port.h"
```

Functions

- void [cleds_init](#) ([cleds_obj_t](#) *cleds, const [led_cfg_t](#) *row_config, uint8_t rows_num, const [led_cfg_t](#) *col_config, uint8_t cols_num)
- uint8_t [cleds_common_set](#) ([cleds_t](#) cleds, uint8_t row)

5.17.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

29 June 2007

5.17.2 Function Documentation

5.17.2.1 uint8_t [cleds_common_set](#) ([cleds_t](#) cleds, uint8_t row)

5.17.2.2 void [cleds_init](#) ([cleds_obj_t](#) * cleds, const [led_cfg_t](#) * row_config, uint8_t rows_num, const [led_cfg_t](#) * col_config, uint8_t cols_num)

5.18 cleds.h File Reference

```
#include "config.h"
#include "port.h"
#include "led.h"
```

Data Structures

- struct [cleds_private_t](#)

Typedefs

- typedef [cleds_private_t](#) [cleds_obj_t](#)
- typedef [cleds_obj_t](#) * [cleds_t](#)

Functions

- void [cleds_init](#) ([cleds_obj_t](#) *cleds, const [led_cfg_t](#) *row_config, uint8_t rows_num, const [led_cfg_t](#) *col_config, uint8_t cols_num)
- uint8_t [cleds_common_set](#) ([cleds_t](#) cleds, uint8_t row)
- static void [cleds_set](#) ([cleds_t](#) cleds, uint8_t id, uint8_t val)
- static uint8_t [cleds_cols_num_get](#) ([cleds_t](#) cleds)
- static uint8_t [cleds_rows_num_get](#) ([cleds_t](#) cleds)
- static uint8_t [cleds_active_row_get](#) ([cleds_t](#) cleds)
- static uint8_t [cleds_common_cycle](#) ([cleds_t](#) cleds)

5.18.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

29 June 2007

5.18.2 Typedef Documentation

5.18.2.1 typedef [cleds_private_t](#) [cleds_obj_t](#)

5.18.2.2 typedef [cleds_obj_t](#)* [cleds_t](#)

5.18.3 Function Documentation

5.18.3.1 static uint8_t [cleds_active_row_get](#) ([cleds_t](#) *cleds*) [static]

5.18.3.2 static uint8_t [cleds_cols_num_get](#) ([cleds_t](#) *cleds*) [static]

5.18.3.3 static uint8_t [cleds_common_cycle](#) ([cleds_t](#) *cleds*) [static]

5.18.3.4 uint8_t [cleds_common_set](#) ([cleds_t](#) *cleds*, uint8_t *row*)

5.18.3.5 void [cleds_init](#) ([cleds_obj_t](#) * *cleds*, const [led_cfg_t](#) * *row_config*, uint8_t *rows_num*, const [led_cfg_t](#) * *col_config*, uint8_t *cols_num*)

5.18.3.6 static uint8_t [cleds_rows_num_get](#) ([cleds_t](#) *cleds*) [static]

5.18.3.7 static void [cleds_set](#) ([cleds_t](#) *cleds*, uint8_t *id*, uint8_t *val*) [static]

5.19 colourmap.h File Reference

```
#include "config.h"
```

Defines

- #define COLOURMAP_R_WEIGHT 1.0
- #define COLOURMAP_G_WEIGHT 1.0
- #define COLOURMAP_B_WEIGHT 1.0
- #define COLOURMAP_R(X) ((X) * COLOURMAP_R_WEIGHT * COLOURMAP_SCALE + 0.5)
- #define COLOURMAP_G(X) ((X) * COLOURMAP_G_WEIGHT * COLOURMAP_SCALE + 0.5)
- #define COLOURMAP_B(X) ((X) * COLOURMAP_B_WEIGHT * COLOURMAP_SCALE + 0.5)
- #define COLOURMAP_ENTRY(R, G, B) {COLOURMAP_R(R), COLOURMAP_G(G), COLOURMAP_B(B)},

Typedefs

- typedef uint8_t colourmap_elt_t
- typedef colourmap_elt_t colourmap_t [3]

5.19.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

3 July 2007

5.19.2 Define Documentation

5.19.2.1 `#define COLOURMAP_B(X) ((X) * COLOURMAP_B_WEIGHT * COLOURMAP_SCALE + 0.5)`

5.19.2.2 `#define COLOURMAP_B_WEIGHT 1.0`

5.19.2.3 `#define COLOURMAP_ENTRY(R, G, B) {COLOURMAP_R(R), COLOURMAP_G(G), COLOURMAP_B(B)},`

5.19.2.4 `#define COLOURMAP_G(X) ((X) * COLOURMAP_G_WEIGHT * COLOURMAP_SCALE + 0.5)`

5.19.2.5 `#define COLOURMAP_G_WEIGHT 1.0`

5.19.2.6 `#define COLOURMAP_R(X) ((X) * COLOURMAP_R_WEIGHT * COLOURMAP_SCALE + 0.5)`

5.19.2.7 `#define COLOURMAP_R_WEIGHT 1.0`

5.19.3 Typedef Documentation

5.19.3.1 `typedef uint8_t colourmap_elt_t`

5.19.3.2 `typedef colourmap_elt_t colourmap_t[3]`

5.20 crc8541.c File Reference

```
#include "crc8541.h"
```

Functions

- static `crc8_t` `crc8541_bit` (`crc8_t` *crc*, `uint8_t` *in*)
- `crc8_t` `crc8541_byte` (`crc8_t` *crc*, `uint8_t` *val*)
- `crc8_t` `crc8541` (`crc8_t` *crc*, `uint8_t` **bytes*, `uint8_t` *size*)

5.20.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.20.2 Function Documentation

5.20.2.1 `crc8_t` `crc8541` (`crc8_t` *crc*, `uint8_t` **bytes*, `uint8_t` *size*)

5.20.2.2 static `crc8_t` `crc8541_bit` (`crc8_t` *crc*, `uint8_t` *in*) [static]

5.20.2.3 `crc8_t` `crc8541_byte` (`crc8_t` *crc*, `uint8_t` *val*)

5.21 crc8541.h File Reference

```
#include "config.h"
```

Typedefs

- typedef uint8_t [crc8_t](#)

Functions

- [crc8_t](#) [crc8541_byte](#) ([crc8_t](#) crc, uint8_t val)
- [crc8_t](#) [crc8541](#) ([crc8_t](#) crc, uint8_t *bytes, uint8_t size)

5.21.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.21.2 Typedef Documentation

5.21.2.1 typedef uint8_t [crc8_t](#)

5.21.3 Function Documentation

5.21.3.1 [crc8_t](#) [crc8541](#) ([crc8_t](#) *crc*, uint8_t * *bytes*, uint8_t *size*)

5.21.3.2 [crc8_t](#) [crc8541_byte](#) ([crc8_t](#) *crc*, uint8_t *val*)

5.22 crc8541_test.c File Reference

```
#include <stdlib.h>
#include "crc8541.h"
```

Functions

- int `main` (int *argc*, char ***argv*)

5.22.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.22.2 Function Documentation

5.22.2.1 int `main` (int *argc*, char ** *argv*)

5.23 ds18b20.c File Reference

```
#include <stdio.h>
#include "ds18b20.h"
#include "ulwire.h"
#include "delay.h"
#include "dscrc8.h"
```

Defines

- #define [DS18B20_DEBUG](#) 0
- #define [DS18B20_CRC_CHECK](#) 0

Enumerations

- enum { [DS18B20_FAMILY_CODE](#) = 0x28, [DS1820_FAMILY_CODE](#) = 0x10, [DS18S20_FAMILY_CODE](#) = 0x10 }
- enum { [DS18B20_SCRATCHPAD_BYTES](#) = 9 }
- enum { [DS18B20_CONVERT_T](#) = 0x44, [DS18B20_READ_SCRATCHPAD](#) = 0xbe, [DS18B20_WRITE_SCRATCHPAD](#) = 0x4e }

Functions

- int8_t [ds18b20_temp_conversion_start](#) (ulwire_t dev)
- bool [ds18b20_temp_ready_p](#) (ulwire_t dev)
- int8_t [ds18b20_temp_read](#) (ulwire_t dev, ds18b20_temp_t *ptemp)
- bool [ds18b20_device_p](#) (ulwire_obj_t *dev)
- ulwire_t [ds18b20_init](#) (ulwire_obj_t *dev)

Variables

- static uint8_t [ds18b20_data](#) [DS18B20_SCRATCHPAD_BYTES]

5.23.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.23.2 Define Documentation

5.23.2.1 `#define DS18B20_CRC_CHECK 0`

5.23.2.2 `#define DS18B20_DEBUG 0`

5.23.3 Enumeration Type Documentation

5.23.3.1 anonymous enum

Enumerator:

DS18B20_FAMILY_CODE

DS1820_FAMILY_CODE

DS18S20_FAMILY_CODE

5.23.3.2 anonymous enum

Enumerator:

DS18B20_SCRATCHPAD_BYTES

5.23.3.3 anonymous enum

Enumerator:

DS18B20_CONVERT_T

DS18B20_READ_SCRATCHPAD

DS18B20_WRITE_SCRATCHPAD

5.23.4 Function Documentation

5.23.4.1 `bool ds18b20_device_p (u1wire_obj_t * dev)`

5.23.4.2 `u1wire_t ds18b20_init (u1wire_obj_t * dev)`

5.23.4.3 `int8_t ds18b20_temp_conversion_start (u1wire_t dev)`

5.23.4.4 `int8_t ds18b20_temp_read (u1wire_t dev, ds18b20_temp_t * ptemp)`

5.23.4.5 `bool ds18b20_temp_ready_p (u1wire_t dev)`

5.23.5 Variable Documentation

5.23.5.1 `uint8_t ds18b20_data[DS18B20_SCRATCHPAD_BYTES] [static]`

5.24 ds18b20.h File Reference

```
#include "config.h"
#include "ulwire.h"
```

Defines

- #define [DS18B20_TEMP_INT](#)(TEMP) (((TEMP) + DS18B20_COUNTS_PER_DEGREE / 2) >> DS18B20_BITS_PER_DEGREE)
- #define [DS18B20_TEMP_DOUBLE](#)(TEMP) ((TEMP) / (double)DS18B20_COUNTS_PER_DEGREE + 0.5)

Typedefs

- typedef int16_t [ds18b20_temp_t](#)

Enumerations

- enum { [DS18B20_BITS_PER_DEGREE](#) = 8 }
- enum { [DS18B20_COUNTS_PER_DEGREE](#) = (1 << 8) }
- enum { [DS18B20_TEMP_BITS](#) = 12, [DS1820_TEMP_BITS](#) = 9 }

Functions

- int8_t [ds18b20_temp_conversion_start](#) (ulwire_t dev)
- bool [ds18b20_temp_ready_p](#) (ulwire_t dev)
- int8_t [ds18b20_temp_read](#) (ulwire_t dev, [ds18b20_temp_t](#) *temp)
- bool [ds18b20_device_p](#) (ulwire_obj_t *dev)
- ulwire_t [ds18b20_init](#) (ulwire_obj_t *dev)

5.24.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.24.2 Define Documentation

5.24.2.1 `#define DS18B20_TEMP_DOUBLE(TEMP) ((TEMP) / (double)DS18B20_COUNTS_PER_DEGREE + 0.5)`

5.24.2.2 `#define DS18B20_TEMP_INT(TEMP) (((TEMP) + DS18B20_COUNTS_PER_DEGREE / 2) >> DS18B20_BITS_PER_DEGREE)`

5.24.3 Typedef Documentation

5.24.3.1 `typedef int16_t ds18b20_temp_t`

5.24.4 Enumeration Type Documentation

5.24.4.1 anonymous enum

Enumerator:

DS18B20_BITS_PER_DEGREE

5.24.4.2 anonymous enum

Enumerator:

DS18B20_COUNTS_PER_DEGREE

5.24.4.3 anonymous enum

Enumerator:

DS18B20_TEMP_BITS

DS1820_TEMP_BITS

5.24.5 Function Documentation

5.24.5.1 `bool ds18b20_device_p (u1wire_obj_t * dev)`

5.24.5.2 `u1wire_t ds18b20_init (u1wire_obj_t * dev)`

5.24.5.3 `int8_t ds18b20_temp_conversion_start (u1wire_t dev)`

5.24.5.4 `int8_t ds18b20_temp_read (u1wire_t dev, ds18b20_temp_t * temp)`

5.24.5.5 `bool ds18b20_temp_ready_p (u1wire_t dev)`

5.25 ds2450.c File Reference

```
#include <stdio.h>
#include "ds2450.h"
#include "dscrc16.h"
#include "ulwire.h"
#include "delay.h"
```

Defines

- `#define DS2450_DEBUG 1`
- `#define DS2450_CRC_CHECK 1`

Enumerations

- enum { `DS2450_FAMILY_CODE` = 0x20 }
- enum { `DS2450_MEMORY_BYTES` = 24 }
- enum { `DS2450_CONVERT` = 0x3c, `DS2450_READ_MEMORY` = 0xaa, `DS2450_WRITE_MEMORY` = 0x55 }

Functions

- `int8_t ds2450_adc_conversion_start (ulwire_t dev, uint8_t channel_mask)`
- `bool ds2450_adc_ready_p (ulwire_t dev)`
- `static int8_t ds2450_memory_read (ulwire_t dev, uint16_t addr, uint8_t *data, uint8_t size)`
- `static int8_t ds2450_memory_write (ulwire_t dev, uint16_t addr, uint8_t *data, uint8_t size)`
- `int8_t ds2450_adc_read (ulwire_t dev, uint8_t channel_mask, uint16_t *adc)`
- `bool ds2450_device_p (ulwire_obj_t *dev)`
- `void ds2450_debug (ulwire_t dev)`
- `ulwire_t ds2450_init (ulwire_obj_t *dev)`

5.25.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.25.2 Define Documentation

5.25.2.1 `#define DS2450_CRC_CHECK 1`

5.25.2.2 `#define DS2450_DEBUG 1`

5.25.3 Enumeration Type Documentation

5.25.3.1 anonymous enum

Enumerator:

DS2450_FAMILY_CODE

5.25.3.2 anonymous enum

Enumerator:

DS2450_MEMORY_BYTES

5.25.3.3 anonymous enum

Enumerator:

DS2450_CONVERT

DS2450_READ_MEMORY

DS2450_WRITE_MEMORY

5.25.4 Function Documentation

5.25.4.1 `int8_t ds2450_adc_conversion_start (u1wire_t dev, uint8_t channel_mask)`

5.25.4.2 `int8_t ds2450_adc_read (u1wire_t dev, uint8_t channel_mask, uint16_t * adc)`

5.25.4.3 `bool ds2450_adc_ready_p (u1wire_t dev)`

5.25.4.4 `void ds2450_debug (u1wire_t dev)`

5.25.4.5 `bool ds2450_device_p (u1wire_obj_t * dev)`

5.25.4.6 `u1wire_t ds2450_init (u1wire_obj_t * dev)`

5.25.4.7 `static int8_t ds2450_memory_read (u1wire_t dev, uint16_t addr, uint8_t * data, uint8_t size) [static]`

5.25.4.8 `static int8_t ds2450_memory_write (u1wire_t dev, uint16_t addr, uint8_t * data, uint8_t size) [static]`

5.26 ds2450.h File Reference

```
#include "config.h"
#include "ulwire.h"
```

Enumerations

- enum { [DS2450_CHANNELS_NUM](#) = 4 }

Functions

- int8_t [ds2450_adc_conversion_start](#) ([ulwire_t](#) dev, uint8_t channel_mask)
- bool [ds2450_adc_ready_p](#) ([ulwire_t](#) dev)
- int8_t [ds2450_adc_read](#) ([ulwire_t](#) dev, uint8_t channel_mask, uint16_t *adc)
- bool [ds2450_device_p](#) ([ulwire_obj_t](#) *dev)
- void [ds2450_debug](#) ([ulwire_t](#) dev)
- [ulwire_t](#) [ds2450_init](#) ([ulwire_obj_t](#) *dev)

5.26.1 Enumeration Type Documentation

5.26.1.1 anonymous enum

Enumerator:

[DS2450_CHANNELS_NUM](#)

5.26.2 Function Documentation

5.26.2.1 int8_t [ds2450_adc_conversion_start](#) ([ulwire_t](#) dev, uint8_t channel_mask)

5.26.2.2 int8_t [ds2450_adc_read](#) ([ulwire_t](#) dev, uint8_t channel_mask, uint16_t *adc)

5.26.2.3 bool [ds2450_adc_ready_p](#) ([ulwire_t](#) dev)

5.26.2.4 void [ds2450_debug](#) ([ulwire_t](#) dev)

5.26.2.5 bool [ds2450_device_p](#) ([ulwire_obj_t](#) *dev)

5.26.2.6 [ulwire_t](#) [ds2450_init](#) ([ulwire_obj_t](#) *dev)

5.27 dscrc16.c File Reference

```
#include "dscrc16.h"
```

Functions

- static `crc16_t dscrc16_bit` (`crc16_t` *crc*, `uint8_t` *in*)
- `crc16_t dscrc16_byte` (`crc16_t` *crc*, `uint8_t` *val*)
- `crc16_t dscrc16` (`crc16_t` *crc*, `void *`*bytes*, `uint8_t` *size*)

5.27.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.27.2 Function Documentation

5.27.2.1 `crc16_t dscrc16` (`crc16_t` *crc*, `void *`*bytes*, `uint8_t` *size*)

5.27.2.2 static `crc16_t dscrc16_bit` (`crc16_t` *crc*, `uint8_t` *in*) [static]

5.27.2.3 `crc16_t dscrc16_byte` (`crc16_t` *crc*, `uint8_t` *val*)

5.28 dscrc16.h File Reference

```
#include "config.h"
```

Typedefs

- typedef uint16_t [crc16_t](#)

Functions

- [crc16_t dscrc16_byte](#) ([crc16_t](#) crc, uint8_t val)
- [crc16_t dscrc16](#) ([crc16_t](#) crc, void *bytes, uint8_t size)

5.28.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.28.2 Typedef Documentation

5.28.2.1 typedef uint16_t [crc16_t](#)

5.28.3 Function Documentation

5.28.3.1 [crc16_t dscrc16](#) ([crc16_t](#) *crc*, void * *bytes*, uint8_t *size*)

5.28.3.2 [crc16_t dscrc16_byte](#) ([crc16_t](#) *crc*, uint8_t *val*)

5.29 dscrc8.c File Reference

```
#include "dscrc8.h"
```

Functions

- [crc8_t dscrc8_byte](#) ([crc8_t](#) crc, [uint8_t](#) data)
- [crc8_t dscrc8](#) ([crc8_t](#) crc, [uint8_t](#) *bytes, [uint8_t](#) size)

5.29.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.29.2 Function Documentation

5.29.2.1 [crc8_t dscrc8](#) ([crc8_t](#) *crc*, [uint8_t](#) * *bytes*, [uint8_t](#) *size*)

5.29.2.2 [crc8_t dscrc8_byte](#) ([crc8_t](#) *crc*, [uint8_t](#) *data*)

5.30 dscrc8.h File Reference

```
#include "config.h"
```

Typedefs

- typedef uint8_t [crc8_t](#)

Functions

- [crc8_t dscrc8_byte](#) ([crc8_t](#) crc, uint8_t val)
- [crc8_t dscrc8](#) ([crc8_t](#) crc, uint8_t *bytes, uint8_t size)

5.30.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.30.2 Typedef Documentation

5.30.2.1 typedef uint8_t [crc8_t](#)

5.30.3 Function Documentation

5.30.3.1 [crc8_t dscrc8](#) ([crc8_t](#) *crc*, uint8_t * *bytes*, uint8_t *size*)

5.30.3.2 [crc8_t dscrc8_byte](#) ([crc8_t](#) *crc*, uint8_t *val*)

5.31 flasher.c File Reference

```
#include "flasher.h"
```

Defines

- #define [FLASHER_TRANSPARENT](#)

Functions

- int8_t [flasher_pattern_set](#) ([flasher_t](#) flasher, [flasher_pattern_t](#) *pattern)
- [flasher_pattern_t](#) * [flasher_pattern_get](#) ([flasher_t](#) flasher)
- int8_t [flasher_phase_set](#) ([flasher_t](#) flasher, uint8_t phase)
- bool [flasher_update](#) ([flasher_t](#) flasher)
- [flasher_t](#) [flasher_init](#) ([flasher_obj_t](#) *flasher)

5.31.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

13 March 2005

5.31.2 Define Documentation

5.31.2.1 #define [FLASHER_TRANSPARENT](#)

5.31.3 Function Documentation

5.31.3.1 [flasher_t](#) [flasher_init](#) ([flasher_obj_t](#) **flasher*)

5.31.3.2 [flasher_pattern_t](#)* [flasher_pattern_get](#) ([flasher_t](#) *flasher*)

5.31.3.3 int8_t [flasher_pattern_set](#) ([flasher_t](#) *flasher*, [flasher_pattern_t](#) **pattern*)

5.31.3.4 int8_t [flasher_phase_set](#) ([flasher_t](#) *flasher*, uint8_t *phase*)

5.31.3.5 bool [flasher_update](#) ([flasher_t](#) *flasher*)

5.32 flasher.h File Reference

```
#include "config.h"
```

Data Structures

- struct [flasher_pattern_t](#)
- struct [flasher_private_t](#)
- struct [flasher_obj_t](#)

Defines

- #define [FLASHER_PRESCALE](#) 8
- #define [FLASHER_PATTERN](#)(POLL_RATE, MOD_FREQ, MOD_DUTY, FLASHER_PERIOD, FLASHER_DUTY, FLASHES, PERIOD)
- #define [FLASHER_ACTIVE_P](#)(FLASHER) ((([flasher_obj_t](#) *) (FLASHER)) → pattern != 0)
- #define [FLASHER_PATTERN_FLASHES_SET](#)(PATTERN, FLASHES) (PATTERN) → flashes = (FLASHES)

Typedefs

- typedef [flasher_obj_t](#) * [flasher_t](#)

Functions

- int8_t [flasher_pattern_set](#) ([flasher_t](#) flasher, [flasher_pattern_t](#) *pattern)
- [flasher_pattern_t](#) * [flasher_pattern_get](#) ([flasher_t](#) flasher)
- int8_t [flasher_phase_set](#) ([flasher_t](#) flasher, uint8_t phase)
- bool [flasher_update](#) ([flasher_t](#))
- [flasher_t](#) [flasher_init](#) ([flasher_obj_t](#) *info)

5.32.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

13 March 2005

5.32.2 Define Documentation

5.32.2.1 #define [FLASHER_ACTIVE_P](#)(FLASHER) ((([flasher_obj_t](#) *) (FLASHER)) → pattern != 0)

5.32.2.2 #define [FLASHER_PATTERN](#)(POLL_RATE, MOD_FREQ, MOD_DUTY, FLASHER_PERIOD, FLASHER_DUTY, FLASHES, PERIOD)

Value:

```
(POLL_RATE) / (double) (MOD_FREQ) + 0.5, \
(POLL_RATE) * (double) (MOD_DUTY) / (MOD_FREQ) / 100.0 + 0.5, \
(MOD_FREQ) * (FLASHER_PERIOD) / (double) FLASHER_PRESCALE + 0.5, \
(MOD_FREQ) * (FLASHER_PERIOD) * (FLASHER_DUTY) / 100.0 / FLASHER_PRESCALE + 0.5, \
(FLASHES), \
(PERIOD) / (double) (FLASHER_PERIOD) + 0.5
```

5.32.2.3 `#define FLASHER_PATTERN_FLASHES_SET(PATTERN, FLASHES) (PATTERN) → flashes = (FLASHES)`

5.32.2.4 `#define FLASHER_PRESCALE 8`

5.32.3 Typedef Documentation

5.32.3.1 `typedef flasher_obj_t* flasher_t`

5.32.4 Function Documentation

5.32.4.1 `flasher_t flasher_init (flasher_obj_t * info)`

5.32.4.2 `flasher_pattern_t* flasher_pattern_get (flasher_t flasher)`

5.32.4.3 `int8_t flasher_pattern_set (flasher_t flasher, flasher_pattern_t * pattern)`

5.32.4.4 `int8_t flasher_phase_set (flasher_t flasher, uint8_t phase)`

5.32.4.5 `bool flasher_update (flasher_t)`

5.33 flasher_tweak.c File Reference

```
#include "flasher.h"
```

Functions

- void [flasher_tweak_mod_duty](#) ([flasher_pattern_t](#) *pattern, uint8_t mod_duty)
- void [flasher_tweak_mod_freq](#) ([flasher_pattern_t](#) *pattern, uint16_t poll_freq, uint8_t mod_freq)

5.33.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.33.2 Function Documentation

5.33.2.1 void [flasher_tweak_mod_duty](#) ([flasher_pattern_t](#) * *pattern*, uint8_t *mod_duty*)

5.33.2.2 void [flasher_tweak_mod_freq](#) ([flasher_pattern_t](#) * *pattern*, uint16_t *poll_freq*, uint8_t *mod_freq*)

5.34 flasher_tweak.h File Reference

```
#include "flasher.h"
```

Functions

- void [flasher_tweak_mod_duty](#) ([flasher_pattern_t](#) *pattern, uint8_t mod_duty)
- void [flasher_tweak_mod_freq](#) ([flasher_pattern_t](#) *pattern, uint16_t poll_freq, uint8_t mod_freq)

5.34.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.34.2 Function Documentation

5.34.2.1 void [flasher_tweak_mod_duty](#) ([flasher_pattern_t](#) * *pattern*, uint8_t *mod_duty*)

5.34.2.2 void [flasher_tweak_mod_freq](#) ([flasher_pattern_t](#) * *pattern*, uint16_t *poll_freq*, uint8_t *mod_freq*)

5.35 font.c File Reference

```
#include "font.h"
```

Functions

- bool `font_display` (char *ch*, `font_t` **font*, void(*display)(void *data, uint8_t pixel, bool val), void *data)

5.35.1 Function Documentation

5.35.1.1 bool `font_display` (char *ch*, `font_t` **font*, void(*) (void *data, uint8_t pixel, bool val) *display*, void * *data*)

5.36 font.h File Reference

```
#include "config.h"
```

Data Structures

- struct [font_t](#)

Functions

- bool [font_display](#) (char *ch*, [font_t](#) **font*, void(*display)(void *data, uint8_t pixel, bool val), void *data)

5.36.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 March 2007

5.36.2 Function Documentation

5.36.2.1 bool [font_display](#) (char *ch*, [font_t](#) **font*, void(*)*(void *data, uint8_t pixel, bool val) display*, void **data*)

5.37 isqrt16.c File Reference

```
#include <limits.h>
#include <stdint.h>
```

Functions

- uint8_t [isqrt](#) (uint16_t val)

5.37.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.37.2 Function Documentation

5.37.2.1 uint8_t isqrt (uint16_t val)

5.38 isqrt32.c File Reference

```
#include <limits.h>
#include <stdint.h>
```

Functions

- uint16_t [isqrt32](#) (uint32_t val)

5.38.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.38.2 Function Documentation

5.38.2.1 uint16_t isqrt32 (uint32_t val)

5.39 lcd.c File Reference

```
#include "config.h"
#include "lcd.h"
#include "delay.h"
```

Defines

- #define [LCD_DEBUG](#) HOSTED
- #define [lcd_wait](#)(dev) DELAY_US (40 + 20)
- #define [lcd_data_set](#)(dev, data)
- #define [lcd_mode_control](#)(dev) port_pins_set_low (dev → cfg → rs_port, dev → rs_mask)
- #define [lcd_mode_data](#)(dev) port_pins_set_high (dev → cfg → rs_port, dev → rs_mask)
- #define [lcd_strobe](#)(dev)
- #define [lcd_write](#)(dev, data)

Functions

- void [lcd_putc](#) ([lcd_t](#) dev, char ch)
- void [lcd_puts](#) ([lcd_t](#) dev, const char *str)
- void [lcd_clear](#) ([lcd_t](#) dev)
- void [lcd_goto](#) ([lcd_t](#) dev, uint8_t row, uint8_t col)
- [lcd_t](#) [lcd_init](#) ([lcd_obj_t](#) *info, const [lcd_cfg_t](#) *cfg)

Variables

- static const uint8_t [lcd_init_data](#) []

5.39.1 Detailed Description

Author:

M.P. Hayes

Date:

16 Feb 2003

Description: Routines to interface to a Hitachi HD44780/KS0066 LCD controller in 4-bit mode with no readback.

5.39.2 Define Documentation

5.39.2.1 #define [lcd_data_set](#)(dev, data)

Value:

```
port_bus_write (dev->cfg->data_port, \
                dev->cfg->d_bit, dev->cfg->d_bit + 3, (data))
```

5.39.2.2 #define LCD_DEBUG HOSTED**5.39.2.3 #define lcd_mode_control(dev) port_pins_set_low (dev → cfg → rs_port, dev → rs_mask)****5.39.2.4 #define lcd_mode_data(dev) port_pins_set_high (dev → cfg → rs_port, dev → rs_mask)****5.39.2.5 #define lcd_strobe(dev)****Value:**

```

do
{
    port_pins_set_high (dev->cfg->e_port, dev->e_mask);
    DELAY_US (2);
    port_pins_set_low (dev->cfg->e_port, dev->e_mask);
}
while (0)

```

5.39.2.6 #define lcd_wait(dev) DELAY_US (40 + 20)**5.39.2.7 #define lcd_write(dev, data)****Value:**

```

do
{
    uint8_t _tmp = (data);

    /* Send MS nibble. */
    lcd_data_set (dev, _tmp >> 4);
    lcd_strobe (dev);
    /* Send LS nibble. */
    lcd_data_set (dev, _tmp);
    lcd_strobe (dev);
    lcd_wait (dev);
}
while (0)

```

5.39.3 Function Documentation**5.39.3.1 void lcd_clear (lcd_t dev)****5.39.3.2 void lcd_goto (lcd_t dev, uint8_t row, uint8_t col)****5.39.3.3 lcd_t lcd_init (lcd_obj_t * info, const lcd_cfg_t * cfg)****5.39.3.4 void lcd_putc (lcd_t dev, char ch)****5.39.3.5 void lcd_puts (lcd_t dev, const char * str)****5.39.4 Variable Documentation****5.39.4.1 const uint8_t lcd_init_data[] [static]****Initial value:**

```
{  
    LCD_FUNCTION | BIT (3) | BIT (2),  
  
    LCD_DISPLAY | BIT (2),  
  
    LCD_ENTRY_MODE | BIT (1),  
  
    LCD_CLEAR  
}
```

5.40 lcd.h File Reference

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [lcd_cfg_t](#)
- struct [lcd_obj_t](#)

Typedefs

- typedef [lcd_obj_t](#) * [lcd_t](#)

Enumerations

- enum {
 [LCD_CLEAR](#) = BIT (0), [LCD_HOME](#) = BIT (1), [LCD_ENTRY_MODE](#) = BIT (2), [LCD_-](#)
 [DISPLAY](#) = BIT (3),
 [LCD_SHIFT](#) = BIT (4), [LCD_FUNCTION](#) = BIT (5), [LCD_CG_RAM_ADDRESS](#) = BIT (6),
 [LCD_DD_RAM_ADDRESS](#) = BIT (7) }

Functions

- void [lcd_putc](#) ([lcd_t](#) dev, char)
- void [lcd_puts](#) ([lcd_t](#) dev, const char *)
- void [lcd_clear](#) ([lcd_t](#) dev)
- void [lcd_goto](#) ([lcd_t](#) dev, uint8_t row, uint8_t col)
- [lcd_t](#) [lcd_init](#) ([lcd_obj_t](#) *info, const [lcd_cfg_t](#) *cfg)

5.40.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 Feb 2003

5.40.2 Typedef Documentation

5.40.2.1 typedef [lcd_obj_t](#)* [lcd_t](#)

5.40.3 Enumeration Type Documentation

5.40.3.1 anonymous enum

Enumerator:

[LCD_CLEAR](#)

LCD_HOME
LCD_ENTRY_MODE
LCD_DISPLAY
LCD_SHIFT
LCD_FUNCTION
LCD_CG_RAM_ADDRESS
LCD_DD_RAM_ADDRESS

5.40.4 Function Documentation

5.40.4.1 void lcd_clear ([lcd_t](#) *dev*)

5.40.4.2 void lcd_goto ([lcd_t](#) *dev*, [uint8_t](#) *row*, [uint8_t](#) *col*)

5.40.4.3 [lcd_t](#) lcd_init ([lcd_obj_t](#) * *info*, const [lcd_cfg_t](#) * *cfg*)

5.40.4.4 void lcd_putc ([lcd_t](#) *dev*, char)

5.40.4.5 void lcd_puts ([lcd_t](#) *dev*, const char *)

5.41 led.c File Reference

```
#include "led.h"
```

Functions

- [led_t led_init](#) (const [led_cfg_t](#) *cfg)

5.41.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.41.2 Function Documentation

5.41.2.1 [led_t led_init](#) (const [led_cfg_t](#) * *cfg*)

5.42 led.h File Reference

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [led_cfg_t](#)

Defines

- #define [LED_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}

Typedefs

- typedef const [led_cfg_t](#) [led_obj_t](#)
- typedef [led_obj_t](#) * [led_t](#)

Functions

- static void [led_set](#) ([led_t](#) led, uint8_t state)
- static void [led_toggle](#) ([led_t](#) led)
- [led_t](#) [led_init](#) (const [led_cfg_t](#) *cfg)

5.42.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.42.2 Define Documentation

5.42.2.1 `#define LED_CFG(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}`

5.42.3 Typedef Documentation

5.42.3.1 `typedef const led_cfg_t led_obj_t`

5.42.3.2 `typedef led_obj_t* led_t`

5.42.4 Function Documentation

5.42.4.1 `led_t led_init (const led_cfg_t * cfg)`

5.42.4.2 `static void led_set (led_t led, uint8_t state)` `[static]`

5.42.4.3 `static void led_toggle (led_t led)` `[static]`

5.43 led_flash.c File Reference

```
#include "config.h"
#include "delay.h"
#include "led.h"
```

Functions

- void `led_flash` (`led_t` led, uint8_t blinks, uint8_t delayms)

5.43.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

10 Jan 2006

5.43.2 Function Documentation

5.43.2.1 void `led_flash` (`led_t` led, uint8_t *blinks*, uint8_t *delayms*)

5.44 led_flash.h File Reference

LED flashing routine.

```
#include "config.h"
#include "led.h"
```

Functions

- void `led_flash` (`led_t` led, uint8_t blinks, uint8_t delayms)

5.44.1 Detailed Description

LED flashing routine.

Author:

M. P. Hayes, UCECE

Date:

10 Jan 2006

5.44.2 Function Documentation

5.44.2.1 void `led_flash` (`led_t` led, uint8_t blinks, uint8_t delayms)

5.45 lmatrix.c File Reference

```
#include "lmatrix.h"
#include "port.h"
```

Defines

- #define [LMATRIX_TRANSPARENT](#)
- #define [ROWBIT\(N\)](#) BIT(N)

Functions

- void [lmatrix_update](#) ([lmatrix_t](#) lmatrix)
- void [lmatrix_set](#) ([lmatrix_t](#) lmatrix, uint8_t row, uint8_t col, bool val)
- void [lmatrix_write](#) ([lmatrix_t](#) lmatrix, uint8_t *screen, uint8_t *map)
- [lmatrix_t](#) [lmatrix_init](#) ([lmatrix_obj_t](#) *lmatrix)

Variables

- static [lmatrix_port_t](#) col_ports []

5.45.1 Define Documentation

5.45.1.1 #define [LMATRIX_TRANSPARENT](#)

5.45.1.2 #define [ROWBIT\(N\)](#) BIT(N)

5.45.2 Function Documentation

5.45.2.1 [lmatrix_t](#) [lmatrix_init](#) ([lmatrix_obj_t](#) * lmatrix)

5.45.2.2 void [lmatrix_set](#) ([lmatrix_t](#) lmatrix, uint8_t row, uint8_t col, bool val)

5.45.2.3 void [lmatrix_update](#) ([lmatrix_t](#) lmatrix)

5.45.2.4 void [lmatrix_write](#) ([lmatrix_t](#) lmatrix, uint8_t * screen, uint8_t * map)

5.45.3 Variable Documentation

5.45.3.1 [lmatrix_port_t](#) col_ports[] [static]

Initial value:

```
{
    {LMATRIX_COL1_PORT, BIT (LMATRIX_COL1_BIT)},
    {LMATRIX_COL2_PORT, BIT (LMATRIX_COL2_BIT)},
    {LMATRIX_COL3_PORT, BIT (LMATRIX_COL3_BIT)},
    {LMATRIX_COL4_PORT, BIT (LMATRIX_COL4_BIT)},
    {LMATRIX_COL5_PORT, BIT (LMATRIX_COL5_BIT)}
}
```

5.46 lmatrix.h File Reference

This drives a multiplexed LED matrix. It only supports a single instance.

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [lmatrix_port_t](#)
- struct [lmatrix_private_t](#)

Typedefs

- typedef uint8_t [lmatrix_row_state_t](#)
- typedef [lmatrix_private_t](#) [lmatrix_obj_t](#)
- typedef [lmatrix_obj_t](#) * [lmatrix_t](#)

Enumerations

- enum { [LMATRIX_PIXELS](#) = [LMATRIX_ROWS](#) * [LMATRIX_COLS](#) }

Functions

- void [lmatrix_set](#) ([lmatrix_t](#) lmatrix, uint8_t row, uint8_t col, bool val)
- void [lmatrix_write](#) ([lmatrix_t](#) lmatrix, uint8_t *screen, uint8_t *map)
- void [lmatrix_update](#) ([lmatrix_t](#) lmatrix)
- [lmatrix_t](#) [lmatrix_init](#) ([lmatrix_obj_t](#) *info)

5.46.1 Detailed Description

This drives a multiplexed LED matrix. It only supports a single instance.

Author:

M. P. Hayes, UCECE

Date:

28 March 2007

5.46.2 Typedef Documentation

5.46.2.1 typedef [lmatrix_private_t](#) [lmatrix_obj_t](#)

5.46.2.2 typedef uint8_t [lmatrix_row_state_t](#)

5.46.2.3 typedef [lmatrix_obj_t](#)* [lmatrix_t](#)

5.46.3 Enumeration Type Documentation

5.46.3.1 anonymous enum

Enumerator:

LMATRIX_PIXELS

5.46.4 Function Documentation

5.46.4.1 [lmatrix_t](#) [lmatrix_init](#) ([lmatrix_obj_t](#) * *info*)

5.46.4.2 void [lmatrix_set](#) ([lmatrix_t](#) *lmatrix*, uint8_t *row*, uint8_t *col*, bool *val*)

5.46.4.3 void [lmatrix_update](#) ([lmatrix_t](#) *lmatrix*)

5.46.4.4 void [lmatrix_write](#) ([lmatrix_t](#) *lmatrix*, uint8_t * *screen*, uint8_t * *map*)

5.47 mbuttons.c File Reference

```
#include "mbuttons.h"  
#include "port.h"
```

Functions

- void [mbuttons_init](#) ([mbuttons_obj_t](#) *mbuttons, [button_obj_t](#) *mbutton_objs, const [button_cfg_t](#) *row_config, uint8_t rows_num, const [button_cfg_t](#) *col_config, uint8_t cols_num, uint8_t poll_count)
- void [mbuttons_poll](#) ([mbuttons_t](#) mbuttons)
- bool [mbuttons_any_state_p](#) ([mbuttons_t](#) mbuttons, [button_state_t](#) state)
- void [mbuttons_wakeup_init](#) ([mbuttons_t](#) mbuttons)

5.47.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

29 June 2007

5.47.2 Function Documentation

5.47.2.1 bool [mbuttons_any_state_p](#) ([mbuttons_t](#) mbuttons, [button_state_t](#) state)

5.47.2.2 void [mbuttons_init](#) ([mbuttons_obj_t](#) *mbuttons, [button_obj_t](#) *mbutton_objs, const [button_cfg_t](#) *row_config, uint8_t rows_num, const [button_cfg_t](#) *col_config, uint8_t cols_num, uint8_t poll_count)

5.47.2.3 void [mbuttons_poll](#) ([mbuttons_t](#) mbuttons)

5.47.2.4 void [mbuttons_wakeup_init](#) ([mbuttons_t](#) mbuttons)

5.48 mbuttons.h File Reference

```
#include "button.h"
```

Data Structures

- struct [mbuttons_private_t](#)

Typedefs

- typedef [mbuttons_private_t](#) [mbuttons_obj_t](#)
- typedef [mbuttons_obj_t](#) * [mbuttons_t](#)

Functions

- void [mbuttons_init](#) ([mbuttons_obj_t](#) *mbuttons, [button_obj_t](#) *mbuttons_info, const [button_cfg_t](#) *row_config, uint8_t rows_num, const [button_cfg_t](#) *col_config, uint8_t cols_num, uint8_t poll_count)
- void [mbuttons_poll](#) ([mbuttons_t](#) mbuttons)
- bool [mbuttons_any_state_p](#) ([mbuttons_t](#) mbuttons, [button_state_t](#) state)
- static bool [mbuttons_any_pushed_p](#) ([mbuttons_t](#) mbuttons)
- bool [mbuttons_any_down_p](#) ([mbuttons_t](#) mbuttons)
- void [mbuttons_wakeup_init](#) ([mbuttons_t](#) mbuttons)
- static bool [mbuttons_pushed_p](#) ([mbuttons_t](#) mbuttons, uint8_t id)
- static bool [mbuttons_released_p](#) ([mbuttons_t](#) mbuttons, uint8_t id)
- static bool [mbuttons_held_p](#) ([mbuttons_t](#) mbuttons, uint8_t id, uint8_t hold_time)
- static bool [mbuttons_hold_released_p](#) ([mbuttons_t](#) mbuttons, uint8_t id, uint8_t hold_time)
- static bool [mbuttons_wakeup_p](#) ([mbuttons_t](#) mbuttons)

5.48.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

29 June 2007

5.48.2 Typedef Documentation

5.48.2.1 typedef [mbuttons_private_t](#) [mbuttons_obj_t](#)

5.48.2.2 typedef [mbuttons_obj_t](#)* [mbuttons_t](#)

5.48.3 Function Documentation

5.48.3.1 bool [mbuttons_any_down_p](#) ([mbuttons_t](#) *mbuttons*)

5.48.3.2 static bool [mbuttons_any_pushed_p](#) ([mbuttons_t](#) *mbuttons*) [static]

5.48.3.3 bool [mbuttons_any_state_p](#) ([mbuttons_t](#) *mbuttons*, [button_state_t](#) *state*)

5.48.3.4 static bool [mbuttons_held_p](#) ([mbuttons_t](#) *mbuttons*, uint8_t *id*, uint8_t *hold_time*)
[static]

5.48.3.5 static bool [mbuttons_hold_released_p](#) ([mbuttons_t](#) *mbuttons*, uint8_t *id*, uint8_t *hold_time*) [static]

5.48.3.6 void [mbuttons_init](#) ([mbuttons_obj_t](#) * *mbuttons*, [button_obj_t](#) * *mbuttons_info*, const [button_cfg_t](#) * *row_config*, uint8_t *rows_num*, const [button_cfg_t](#) * *col_config*, uint8_t *cols_num*, uint8_t *poll_count*)

5.48.3.7 void [mbuttons_poll](#) ([mbuttons_t](#) *mbuttons*)

5.48.3.8 static bool [mbuttons_pushed_p](#) ([mbuttons_t](#) *mbuttons*, uint8_t *id*) [static]

5.48.3.9 static bool [mbuttons_released_p](#) ([mbuttons_t](#) *mbuttons*, uint8_t *id*) [static]

5.48.3.10 void [mbuttons_wakeup_init](#) ([mbuttons_t](#) *mbuttons*)

5.48.3.11 static bool [mbuttons_wakeup_p](#) ([mbuttons_t](#) *mbuttons*) [static]

5.49 mcleds.c File Reference

```
#include "mcleds.h"
```

Functions

- `mcleds_t mcleds_init` (`mcleds_obj_t` *mcleds, const `led_cfg_t` *row_config, uint8_t rows_num, const `led_cfg_t` *col_config, uint8_t cols_num, `colourmap_t` *colourmap, uint8_t colourmap_size, `mcleds_state_t` *state, uint8_t update_rate)
- `bool mcleds_update` (`mcleds_t` mcleds, uint8_t *screen)
- `void mcleds_enable` (`mcleds_t` mcleds, uint8_t row)
- `void mcleds_off` (`mcleds_t` mcleds)

5.49.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

3 July 2007

5.49.2 Function Documentation

5.49.2.1 `void mcleds_enable` (`mcleds_t` mcleds, uint8_t row)

5.49.2.2 `mcleds_t mcleds_init` (`mcleds_obj_t` *mcleds, const `led_cfg_t` *row_config, uint8_t rows_num, const `led_cfg_t` *col_config, uint8_t cols_num, `colourmap_t` *colourmap, uint8_t colourmap_size, `mcleds_state_t` *state, uint8_t update_rate)

5.49.2.3 `void mcleds_off` (`mcleds_t` mcleds)

5.49.2.4 `bool mcleds_update` (`mcleds_t` mcleds, uint8_t *screen)

5.50 mcleds.h File Reference

```
#include "config.h"
#include "cleds.h"
#include "colourmap.h"
#include "ticker.h"
```

Data Structures

- struct [mcleds_state_t](#)
- struct [mcleds_private_t](#)

Typedefs

- typedef [mcleds_private_t](#) [mcleds_obj_t](#)
- typedef [mcleds_obj_t](#) * [mcleds_t](#)

Functions

- [mcleds_t](#) [mcleds_init](#) ([mcleds_obj_t](#) *mcleds, const [led_cfg_t](#) *row_config, uint8_t rows_num, const [led_cfg_t](#) *col_config, uint8_t cols_num, [colourmap_t](#) *colourmap, uint8_t colourmap_size, [mcleds_state_t](#) *state, uint8_t update_rate)
- bool [mcleds_update](#) ([mcleds_t](#) mcleds, uint8_t *screen)
- void [mcleds_off](#) ([mcleds_t](#) mcleds)
- void [mcleds_enable](#) ([mcleds_t](#) mcleds, uint8_t row)
- static uint8_t [mcleds_disable](#) ([mcleds_t](#) mcleds)
- static void [mcleds_colourmap_set](#) ([mcleds_t](#) mcleds, [colourmap_t](#) *colourmap, uint8_t colourmap_size)

5.50.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

3 July 2007

5.50.2 Typedef Documentation

5.50.2.1 typedef [mcleds_private_t](#) [mcleds_obj_t](#)

5.50.2.2 typedef [mcleds_obj_t](#)* [mcleds_t](#)

5.50.3 Function Documentation

5.50.3.1 static void [mcleds_colourmap_set](#) ([mcleds_t](#) *mcleds*, [colourmap_t](#) * *colourmap*, uint8_t *colourmap_size*) [static]

5.50.3.2 static uint8_t [mcleds_disable](#) ([mcleds_t](#) *mcleds*) [static]

5.50.3.3 void [mcleds_enable](#) ([mcleds_t](#) *mcleds*, uint8_t *row*)

5.50.3.4 [mcleds_t](#) [mcleds_init](#) ([mcleds_obj_t](#) * *mcleds*, const [led_cfg_t](#) * *row_config*, uint8_t *rows_num*, const [led_cfg_t](#) * *col_config*, uint8_t *cols_num*, [colourmap_t](#) * *colourmap*, uint8_t *colourmap_size*, [mcleds_state_t](#) * *state*, uint8_t *update_rate*)

5.50.3.5 void [mcleds_off](#) ([mcleds_t](#) *mcleds*)

5.50.3.6 bool [mcleds_update](#) ([mcleds_t](#) *mcleds*, uint8_t * *screen*)

5.51 mmelody.c File Reference

Play simple melodies.

```
#include "mmelody.h"
```

Defines

- #define [MMELODY_TRANSPARENT](#) 1

Enumerations

- enum { [MMELODY_SCALE_SIZE](#) = 12 }

Functions

- static void [mmelody_ticker_set](#) ([mmelody_t](#) mmelody)
- static void [mmelody_note_play](#) ([mmelody_t](#) mmelody, [mmelody_note_t](#) note)
- static void [mmelody_note_fraction_set](#) ([mmelody_t](#) mmelody, [uint8_t](#) note_fraction)
- static [mmelody_note_t](#) [mmelody_char_to_note](#) ([uint8_t](#) ch)
- static const char * [mmelody_scan](#) ([mmelody_t](#) mmelody, const char *str)
- void [mmelody_play](#) ([mmelody_t](#) mmelody, const char *str)
- void [mmelody_speed_set](#) ([mmelody_t](#) mmelody, [mmelody_speed_t](#) speed)
- void [mmelody_volume_set](#) ([mmelody_t](#) mmelody, [mmelody_volume_t](#) volume)
- void [mmelody_update](#) ([mmelody_t](#) mmelody)
- [mmelody_t](#) [mmelody_init](#) ([mmelody_obj_t](#) *mmelody, [uint16_t](#) poll_rate, [mmelody_callback_t](#) play_callback, void *play_callback_data)

5.51.1 Detailed Description

Play simple melodies.

Author:

M. P. Hayes, UCECE

Date:

20 April 2007

5.51.2 Define Documentation

5.51.2.1 #define [MMELODY_TRANSPARENT](#) 1

5.51.3 Enumeration Type Documentation

5.51.3.1 anonymous enum

Enumerator:

[MMELODY_SCALE_SIZE](#)

5.51.4 Function Documentation

- 5.51.4.1 static [mmelody_note_t](#) mmelody_char_to_note (uint8_t *ch*) [static]
- 5.51.4.2 [mmelody_t](#) mmelody_init ([mmelody_obj_t](#) * *mmelody*, uint16_t *poll_rate*, [mmelody_callback_t](#) *play_callback*, void * *play_callback_data*)
- 5.51.4.3 static void mmelody_note_fraction_set ([mmelody_t](#) *mmelody*, uint8_t *note_fraction*) [static]
- 5.51.4.4 static void mmelody_note_play ([mmelody_t](#) *mmelody*, [mmelody_note_t](#) *note*) [static]
- 5.51.4.5 void mmelody_play ([mmelody_t](#) *mmelody*, const char * *str*)
- 5.51.4.6 static const char* mmelody_scan ([mmelody_t](#) *mmelody*, const char * *str*) [static]
- 5.51.4.7 void mmelody_speed_set ([mmelody_t](#) *mmelody*, [mmelody_speed_t](#) *speed*)
- 5.51.4.8 static void mmelody_ticker_set ([mmelody_t](#) *mmelody*) [static]
- 5.51.4.9 void mmelody_update ([mmelody_t](#) *mmelody*)
- 5.51.4.10 void mmelody_volume_set ([mmelody_t](#) *mmelody*, [mmelody_volume_t](#) *volume*)

5.52 mmelody.h File Reference

Play simple melodies.

```
#include "config.h"
#include "font.h"
#include "ticker.h"
```

Data Structures

- struct [mmelody_private_t](#)

Typedefs

- typedef uint8_t [mmelody_speed_t](#)
- typedef uint8_t [mmelody_scale_t](#)
- typedef uint8_t [mmelody_note_t](#)
- typedef uint8_t [mmelody_volume_t](#)
- typedef [mmelody_private_t](#) [mmelody_obj_t](#)
- typedef [mmelody_obj_t](#) * [mmelody_t](#)
- typedef void(* [mmelody_callback_t](#))(void *data, uint8_t note, uint8_t volume)

Enumerations

- enum { [MMELODY_OCTAVE_DEFAULT](#) = 4 }
- enum { [MMELODY_SPEED_DEFAULT](#) = 200 }

Functions

- [mmelody_t](#) [mmelody_init](#) ([mmelody_obj_t](#) *dev, uint16_t poll_rate, [mmelody_callback_t](#) play_callback, void *play_callback_data)
- void [mmelody_play](#) ([mmelody_t](#) mmelody, const char *str)
- void [mmelody_update](#) ([mmelody_t](#) mmelody)
- void [mmelody_speed_set](#) ([mmelody_t](#) mmelody, [mmelody_speed_t](#) speed)
- void [mmelody_volume_set](#) ([mmelody_t](#) mmelody, [mmelody_volume_t](#) volume)

5.52.1 Detailed Description

Play simple melodies.

Author:

M. P. Hayes, UCECE

Date:

20 April 2007

5.52.2 Typedef Documentation

5.52.2.1 typedef void(* [mmelody_callback_t](#))(void *data, uint8_t note, uint8_t volume)

5.52.2.2 typedef uint8_t [mmelody_note_t](#)

5.52.2.3 typedef [mmelody_private_t](#) [mmelody_obj_t](#)

5.52.2.4 typedef uint8_t [mmelody_scale_t](#)

5.52.2.5 typedef uint8_t [mmelody_speed_t](#)

5.52.2.6 typedef [mmelody_obj_t](#)* [mmelody_t](#)

5.52.2.7 typedef uint8_t [mmelody_volume_t](#)

5.52.3 Enumeration Type Documentation

5.52.3.1 anonymous enum

Enumerator:

MMELODY_OCTAVE_DEFAULT

5.52.3.2 anonymous enum

Enumerator:

MMELODY_SPEED_DEFAULT

5.52.4 Function Documentation

5.52.4.1 [mmelody_t](#) [mmelody_init](#) ([mmelody_obj_t](#) * dev, uint16_t poll_rate, [mmelody_callback_t](#) play_callback, void * play_callback_data)

5.52.4.2 void [mmelody_play](#) ([mmelody_t](#) mmelody, const char * str)

5.52.4.3 void [mmelody_speed_set](#) ([mmelody_t](#) mmelody, [mmelody_speed_t](#) speed)

5.52.4.4 void [mmelody_update](#) ([mmelody_t](#) mmelody)

5.52.4.5 void [mmelody_volume_set](#) ([mmelody_t](#) mmelody, [mmelody_volume_t](#) volume)

5.53 mpwm.c File Reference

```
#include "config.h"
#include "mpwm.h"
```

Functions

- void [mpwm_period_set](#) ([mpwm_t](#) mpwm, uint16_t period)
- void [mpwm_duty_set](#) ([mpwm_t](#) mpwm, uint8_t channel, uint16_t duty)
- bool [mpwm_update](#) ([mpwm_t](#) mpwm)
- [mpwm_t](#) [mpwm_init](#) ([mpwm_obj_t](#) *dev, [mpwm_channel_t](#) *channels, uint8_t num_channels)

5.53.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.53.2 Function Documentation

5.53.2.1 void [mpwm_duty_set](#) ([mpwm_t](#) mpwm, uint8_t *channel*, uint16_t *duty*)

5.53.2.2 [mpwm_t](#) [mpwm_init](#) ([mpwm_obj_t](#) * *dev*, [mpwm_channel_t](#) * *channels*, uint8_t *num_channels*)

5.53.2.3 void [mpwm_period_set](#) ([mpwm_t](#) mpwm, uint16_t *period*)

5.53.2.4 bool [mpwm_update](#) ([mpwm_t](#) mpwm)

5.54 mpwm.h File Reference

```
#include "config.h"
```

Data Structures

- struct [mpwm_channel_t](#)
- struct [mpwm_obj_t](#)

Typedefs

- typedef mpwm_struct * [mpwm_t](#)

Functions

- void [mpwm_period_set](#) ([mpwm_t](#) mpwm, uint16_t period)
- void [mpwm_duty_set](#) ([mpwm_t](#) mpwm, uint8_t channel, uint16_t duty)
- bool [mpwm_update](#) ([mpwm_t](#))
- [mpwm_t](#) [mpwm_init](#) ([mpwm_obj_t](#) *info, [mpwm_channel_t](#) *channels, uint8_t num_channels)

5.54.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.54.2 Typedef Documentation

5.54.2.1 typedef struct mpwm_struct* [mpwm_t](#)

5.54.3 Function Documentation

5.54.3.1 void [mpwm_duty_set](#) ([mpwm_t](#) mpwm, uint8_t channel, uint16_t duty)

5.54.3.2 [mpwm_t](#) [mpwm_init](#) ([mpwm_obj_t](#) * info, [mpwm_channel_t](#) * channels, uint8_t num_channels)

5.54.3.3 void [mpwm_period_set](#) ([mpwm_t](#) mpwm, uint16_t period)

5.54.3.4 bool [mpwm_update](#) ([mpwm_t](#))

5.55 mtext.c File Reference

Moving text.

```
#include "mtext.h"
#include "font.h"
#include <limits.h>
```

Functions

- static void [mtext_pixel_set](#) (void *data, uint8_t pixel, bool val)
- static void [mtext_display](#) ([mtext_t](#) mtext, char ch)
- static const char * [mtext_scan](#) ([mtext_t](#) mtext, const char *str)
- [mtext_t](#) [mtext_init](#) ([mtext_obj_t](#) *mtext, uint16_t poll_rate, [font_t](#) *font, uint8_t *image, uint8_t font_rows, uint8_t font_cols, uint8_t *screen, uint8_t rows, uint8_t cols)
- void [mtext_scroller_dir_set](#) ([mtext_t](#) mtext, [scroller_dir_t](#) dir)
- [scroller_dir_t](#) [mtext_scroller_dir_get](#) ([mtext_t](#) mtext)
- int8_t [mtext_update](#) ([mtext_t](#) mtext)
- void [mtext_speed_set](#) ([mtext_t](#) mtext, uint8_t speed)

5.55.1 Detailed Description

Moving text.

Author:

M. P. Hayes, UCECE

Date:

8 April 2007

5.55.2 Function Documentation

5.55.2.1 static void [mtext_display](#) ([mtext_t](#) mtext, char ch) [static]

5.55.2.2 [mtext_t](#) [mtext_init](#) ([mtext_obj_t](#) *mtext, uint16_t poll_rate, [font_t](#) *font, uint8_t *image, uint8_t font_rows, uint8_t font_cols, uint8_t *screen, uint8_t rows, uint8_t cols)

5.55.2.3 static void [mtext_pixel_set](#) (void *data, uint8_t pixel, bool val) [static]

5.55.2.4 static const char* [mtext_scan](#) ([mtext_t](#) mtext, const char *str) [static]

5.55.2.5 [scroller_dir_t](#) [mtext_scroller_dir_get](#) ([mtext_t](#) mtext)

5.55.2.6 void [mtext_scroller_dir_set](#) ([mtext_t](#) mtext, [scroller_dir_t](#) dir)

5.55.2.7 void [mtext_speed_set](#) ([mtext_t](#) mtext, uint8_t speed)

5.55.2.8 int8_t [mtext_update](#) ([mtext_t](#) mtext)

5.56 mtext.h File Reference

Moving text.

```
#include "config.h"
#include "font.h"
#include "scroller.h"
#include "ticker.h"
```

Data Structures

- struct [mtext_obj_t](#)

Typedefs

- typedef [mtext_obj_t](#) * [mtext_t](#)

Enumerations

- enum { [MTEXT_SPEED_SCALER](#) = 16 }
- enum [mtext_mode_t](#) { [MTEXT_MODE_REPLACE](#), [MTEXT_MODE_SCROLL](#) }

Functions

- [mtext_t mtext_init](#) ([mtext_obj_t](#) *dev, uint16_t poll_rate, [font_t](#) *font, uint8_t *image, uint8_t font_rows, uint8_t font_cols, uint8_t *screen, uint8_t rows, uint8_t cols)
- int8_t [mtext_update](#) ([mtext_t](#) mtext)
- void [mtext_scroller_dir_set](#) ([mtext_t](#) mtext, [scroller_dir_t](#) dir)
- [scroller_dir_t](#) [mtext_scroller_dir_get](#) ([mtext_t](#) mtext)
- void [mtext_speed_set](#) ([mtext_t](#) mtext, uint8_t speed)
- static void [mtext_set](#) ([mtext_t](#) mtext, const char *str)
- static const char * [mtext_get](#) ([mtext_t](#) mtext)
- static void [mtext_mode_set](#) ([mtext_t](#) mtext, [mtext_mode_t](#) mode)
- static [mtext_mode_t](#) [mtext_mode_get](#) ([mtext_t](#) mtext)

5.56.1 Detailed Description

Moving text.

Author:

M. P. Hayes, UCECE

Date:

8 April 2007

5.56.2 Typedef Documentation

5.56.2.1 typedef [mtext_obj_t](#)* [mtext_t](#)

5.56.3 Enumeration Type Documentation

5.56.3.1 anonymous enum

Enumerator:

MTEXT_SPEED_SCALER

5.56.3.2 enum [mtext_mode_t](#)

Enumerator:

MTEXT_MODE_REPLACE

MTEXT_MODE_SCROLL

5.56.4 Function Documentation

5.56.4.1 static const char* [mtext_get](#) ([mtext_t](#) *mtext*) [static]

5.56.4.2 [mtext_t](#) [mtext_init](#) ([mtext_obj_t](#) * *dev*, [uint16_t](#) *poll_rate*, [font_t](#) * *font*, [uint8_t](#) * *image*, [uint8_t](#) *font_rows*, [uint8_t](#) *font_cols*, [uint8_t](#) * *screen*, [uint8_t](#) *rows*, [uint8_t](#) *cols*)

5.56.4.3 static [mtext_mode_t](#) [mtext_mode_get](#) ([mtext_t](#) *mtext*) [static]

5.56.4.4 static void [mtext_mode_set](#) ([mtext_t](#) *mtext*, [mtext_mode_t](#) *mode*) [static]

5.56.4.5 [scroller_dir_t](#) [mtext_scroller_dir_get](#) ([mtext_t](#) *mtext*)

5.56.4.6 void [mtext_scroller_dir_set](#) ([mtext_t](#) *mtext*, [scroller_dir_t](#) *dir*)

5.56.4.7 static void [mtext_set](#) ([mtext_t](#) *mtext*, const char * *str*) [static]

5.56.4.8 void [mtext_speed_set](#) ([mtext_t](#) *mtext*, [uint8_t](#) *speed*)

5.56.4.9 [int8_t](#) [mtext_update](#) ([mtext_t](#) *mtext*)

5.57 mu1wire.c File Reference

```
#include <stdio.h>
#include "config.h"
#include "delay.h"
#include "ulwire.h"
#include "irq.h"
#include "port.h"
```

Defines

- #define [U1WIRE_DEBUG](#) 1
- #define [U1WIRE_RELEASE](#)() port_pin_config_pullup (U1WIRE_PORT, U1WIRE_BIT)
- #define [U1WIRE_DRIVE](#)()
- #define [U1WIRE_TEST](#)() port_pin_read (U1WIRE_PORT, U1WIRE_BIT)

Enumerations

- enum { [U1WIRE_READ_ROM](#) = 0x33, [U1WIRE_SKIP_ROM](#) = 0xcc, [U1WIRE_MATCH_ROM](#) = 0x55, [U1WIRE_RECALL](#) = 0xb8 }
- enum { [U1WIRE_DELAY_OFFSET](#) = 6, [U1WIRE_ADDR_BYTES](#) = 6 }

Functions

- int8_t [ulwire_reset](#) (void)
- void [ulwire_bit_write](#) (uint8_t value)
- void [ulwire_byte_write](#) (uint8_t value)
- uint8_t [ulwire_bit_read](#) (void)
- uint8_t [ulwire_byte_read](#) (void)
- int8_t [ulwire_rom_code_read](#) (ulwire_t dev)
- int8_t [ulwire_command](#) (ulwire_t dev, uint8_t command)
- int8_t [ulwire_broadcast](#) (uint8_t command)
- int8_t [ulwire_read](#) (ulwire_t dev, void *data, uint8_t bytes)
- int8_t [ulwire_write](#) (ulwire_t dev, void *data, uint8_t bytes)
- bool [ulwire_ready_p](#) (void)
- int8_t [ulwire_init](#) (ulwire_obj_t *devices, uint8_t devices_max)

5.57.1 Define Documentation

5.57.1.1 #define U1WIRE_DEBUG 1

5.57.1.2 #define U1WIRE_DRIVE()

Value:

```
do {port_pin_config_output (U1WIRE_PORT, U1WIRE_BIT); \
    port_pin_set_low (U1WIRE_PORT, U1WIRE_BIT);} while (0)
```

5.57.1.3 `#define U1WIRE_RELEASE() port_pin_config_pullup (U1WIRE_PORT, U1WIRE_BIT)`

5.57.1.4 `#define U1WIRE_TEST() port_pin_read (U1WIRE_PORT, U1WIRE_BIT)`

5.57.2 Enumeration Type Documentation

5.57.2.1 anonymous enum

Enumerator:

U1WIRE_READ_ROM
U1WIRE_SKIP_ROM
U1WIRE_MATCH_ROM
U1WIRE_RECALL

5.57.2.2 anonymous enum

Enumerator:

U1WIRE_DELAY_OFFSET
U1WIRE_ADDR_BYTES

5.57.3 Function Documentation

5.57.3.1 `uint8_t u1wire_bit_read (void)`

5.57.3.2 `void u1wire_bit_write (uint8_t value)`

5.57.3.3 `int8_t u1wire_broadcast (uint8_t command)`

5.57.3.4 `uint8_t u1wire_byte_read (void)`

5.57.3.5 `void u1wire_byte_write (uint8_t value)`

5.57.3.6 `int8_t u1wire_command (u1wire_t dev, uint8_t command)`

5.57.3.7 `int8_t u1wire_init (u1wire_obj_t * devices, uint8_t devices_max)`

5.57.3.8 `int8_t u1wire_read (u1wire_t dev, void * data, uint8_t bytes)`

5.57.3.9 `bool u1wire_ready_p (void)`

5.57.3.10 `int8_t u1wire_reset (void)`

5.57.3.11 `int8_t u1wire_rom_code_read (u1wire_t dev)`

5.57.3.12 `int8_t u1wire_write (u1wire_t dev, void * data, uint8_t bytes)`

5.58 mu1wire.h File Reference

```
#include "config.h"
```

Data Structures

- union [mu1wire_rom_code_t](#)
- struct [mu1wire_obj_t](#)

Typedefs

- typedef [mu1wire_obj_t](#) * [mu1wire_t](#)

Enumerations

- enum {
 [MUIWIRE_ERR_BUS_LOW](#) = 1, [MUIWIRE_ERR_BUS_STUCK](#) = 2, [MUIWIRE_ERR_PRESENCE_SHORT](#) = 3, [MUIWIRE_ERR_PRESENCE_LONG](#) = 4,
 [MUIWIRE_ERR_MULTIPLE_DEVICES](#) = 5, [MUIWIRE_ERR_BUS_HIGH](#) = 6 }

Functions

- bool [mu1wire_ready_p](#) (void)
- int8_t [mu1wire_reset](#) (void)
- void [mu1wire_bit_write](#) (uint8_t value)
- void [mu1wire_byte_write](#) (uint8_t value)
- uint8_t [mu1wire_bit_read](#) (void)
- uint8_t [mu1wire_byte_read](#) (void)
- int8_t [mu1wire_command](#) ([mu1wire_t](#) dev, uint8_t command)
- int8_t [mu1wire_broadcast](#) (uint8_t command)
- int8_t [mu1wire_read](#) ([mu1wire_t](#) dev, void *data, uint8_t size)
- int8_t [mu1wire_write](#) ([mu1wire_t](#) dev, void *data, uint8_t size)
- int8_t [mu1wire_init](#) ([mu1wire_obj_t](#) *devices, uint8_t devices_max)
- void [mu1wire_debug](#) ([mu1wire_t](#) dev)

5.58.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.58.2 Typedef Documentation

5.58.2.1 typedef [mulwire_obj_t](#)* [mulwire_t](#)

5.58.3 Enumeration Type Documentation

5.58.3.1 anonymous enum

Enumerator:

MUIWIRE_ERR_BUS_LOW
MUIWIRE_ERR_BUS_STUCK
MUIWIRE_ERR_PRESENCE_SHORT
MUIWIRE_ERR_PRESENCE_LONG
MUIWIRE_ERR_MULTIPLE_DEVICES
MUIWIRE_ERR_BUS_HIGH

5.58.4 Function Documentation

5.58.4.1 uint8_t [mulwire_bit_read](#) (void)

5.58.4.2 void [mulwire_bit_write](#) (uint8_t *value*)

5.58.4.3 int8_t [mulwire_broadcast](#) (uint8_t *command*)

5.58.4.4 uint8_t [mulwire_byte_read](#) (void)

5.58.4.5 void [mulwire_byte_write](#) (uint8_t *value*)

5.58.4.6 int8_t [mulwire_command](#) ([mulwire_t](#) *dev*, uint8_t *command*)

5.58.4.7 void [mulwire_debug](#) ([mulwire_t](#) *dev*)

5.58.4.8 int8_t [mulwire_init](#) ([mulwire_obj_t](#) * *devices*, uint8_t *devices_max*)

5.58.4.9 int8_t [mulwire_read](#) ([mulwire_t](#) *dev*, void * *data*, uint8_t *size*)

5.58.4.10 bool [mulwire_ready_p](#) (void)

5.58.4.11 int8_t [mulwire_reset](#) (void)

5.58.4.12 int8_t [mulwire_write](#) ([mulwire_t](#) *dev*, void * *data*, uint8_t *size*)

5.59 muxleds.c File Reference

```
#include "muxleds.h"
```

Defines

- #define [MUXLEDs_TRANSPARENT](#)

Functions

- void [muxleds_set](#) ([muxleds_t](#) muxleds, uint8_t bit, uint8_t val)
- void [muxleds_toggle](#) ([muxleds_t](#) muxleds, uint8_t bit)
- void [muxleds_update](#) ([muxleds_t](#) muxleds)
- [muxleds_t](#) [muxleds_init](#) ([muxleds_obj_t](#) *muxleds, const [muxleds_cfg_t](#) *row_cfg, uint8_t rows_num, const [muxleds_cfg_t](#) *col_cfg, uint8_t cols_num, uint8_t row_on, uint8_t col_on)

5.59.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.59.2 Define Documentation

5.59.2.1 #define MUXLEDs_TRANSPARENT

5.59.3 Function Documentation

5.59.3.1 [muxleds_t](#) [muxleds_init](#) ([muxleds_obj_t](#) * *muxleds*, const [muxleds_cfg_t](#) * *row_cfg*, uint8_t *rows_num*, const [muxleds_cfg_t](#) * *col_cfg*, uint8_t *cols_num*, uint8_t *row_on*, uint8_t *col_on*)

5.59.3.2 void [muxleds_set](#) ([muxleds_t](#) *muxleds*, uint8_t *bit*, uint8_t *val*)

5.59.3.3 void [muxleds_toggle](#) ([muxleds_t](#) *muxleds*, uint8_t *bit*)

5.59.3.4 void [muxleds_update](#) ([muxleds_t](#) *muxleds*)

5.60 muxleds.h File Reference

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [muxleds_cfg_t](#)
- struct [muxleds_row_t](#)
- struct [muxleds_col_t](#)
- struct [muxleds_obj_t](#)

Defines

- #define [MUXLED_ROWS_NUM](#) 8
- #define [MUXLED_COLS_NUM](#) 8
- #define [MUXLED_ROW_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}
- #define [MUXLED_COL_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}

Typedefs

- typedef muxleds_struct * [muxleds_t](#)

Functions

- void [muxleds_set](#) ([muxleds_t](#) muxleds, uint8_t bit, uint8_t val)
- void [muxleds_toggle](#) ([muxleds_t](#) muxleds, uint8_t bit)
- void [muxleds_update](#) ([muxleds_t](#) muxleds)
- [muxleds_t](#) [muxleds_init](#) ([muxleds_obj_t](#) *dev, const [muxleds_cfg_t](#) *row_cfg, uint8_t row_size, const [muxleds_cfg_t](#) *col_cfg, uint8_t col_size)

5.60.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.60.2 Define Documentation

5.60.2.1 `#define MUXLED_COL_CFG(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}`

5.60.2.2 `#define MUXLED_ROW_CFG(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}`

5.60.2.3 `#define MUXLEDS_COLS_NUM 8`

5.60.2.4 `#define MUXLEDS_ROWS_NUM 8`

5.60.3 Typedef Documentation

5.60.3.1 `typedef struct muxleds_struct* muxleds_t`

5.60.4 Function Documentation

5.60.4.1 `muxleds_t muxleds_init (muxleds_obj_t * dev, const muxleds_cfg_t * row_cfg, uint8_t row_size, const muxleds_cfg_t * col_cfg, uint8_t col_size)`

5.60.4.2 `void muxleds_set (muxleds_t muxleds, uint8_t bit, uint8_t val)`

5.60.4.3 `void muxleds_toggle (muxleds_t muxleds, uint8_t bit)`

5.60.4.4 `void muxleds_update (muxleds_t muxleds)`

5.61 nmea.c File Reference

```
#include <stdio.h>
#include "config.h"
```

Functions

- `uint8_t nmea_checksum` (const char *string)
- `void nmea_puts` (const char *string)

5.61.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.61.2 Function Documentation

5.61.2.1 `uint8_t nmea_checksum` (const char * *string*)

5.61.2.2 `void nmea_puts` (const char * *string*)

5.62 nmea.h File Reference

```
#include "config.h"
```

Defines

- #define [NMEA_BUFFER_SIZE](#) 80

Functions

- uint8_t [nmea_checksum](#) (const char *string)
- void [nmea_puts](#) (const char *string)

5.62.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.62.2 Define Documentation

5.62.2.1 #define NMEA_BUFFER_SIZE 80

5.62.3 Function Documentation

5.62.3.1 uint8_t nmea_checksum (const char * *string*)

5.62.3.2 void nmea_puts (const char * *string*)

5.63 nrf2401.c File Reference

```
#include "nrf2401.h"
#include "delay.h"
```

Defines

- #define [CHAR_BIT](#) 8
- #define [NRF_DATA_READY_P](#)(DEV) (port_pins_read ((DEV) → pins.dr_port, (DEV) → pins.dr_bitmask))
- #define [NRF_CONFIGURATION_REGISTER_SIZE](#) 15
- #define [NRF_CE_HIGH_SET](#)(DEV) (port_pins_set_high ((DEV) → pins.ce_port, (DEV) → pins.ce_bitmask))
- #define [NRF_CE_LOW_SET](#)(DEV) (port_pins_set_low ((DEV) → pins.ce_port, (DEV) → pins.ce_bitmask))
- #define [NRF_CS_HIGH_SET](#)(DEV) (port_pins_set_high ((DEV) → pins.cs_port, (DEV) → pins.cs_bitmask))
- #define [NRF_CS_LOW_SET](#)(DEV) (port_pins_set_low ((DEV) → pins.cs_port, (DEV) → pins.cs_bitmask))
- #define [NRF_FULL_CONFIGURE](#)(DEV) (nrf_configure ((DEV), NRF_CONFIGURATION_REGISTER_SIZE))

Enumerations

- enum { [NRF_CHANNEL_NUMBER_MAX](#) = 83 }
- enum { [NRF_CONFIG_DELAY_US](#) = 20 }
- enum { [NRF_LINE_TIME_ENABLE_US](#) = 10 }
- enum { [NRF_T_SB_ACTIVE](#) = 202 }

Functions

- [nrf_t nrf_init](#) (nrf_obj_t *dev, const nrf_cfg_t *cfg)
- [bool nrf_setup](#) (nrf_t rf, uint8_t payload_size)
- [void nrf_configure](#) (nrf_t rf, uint8_t size)
- [uint8_t nrf_receive](#) (nrf_t rf, uint8_t *data, uint8_t ms_to_wait)
- [uint8_t nrf_transmit](#) (nrf_t rf, [rf_address_t](#) *dst_address, uint8_t *data, uint8_t size)
- [void nrf_rf_dir_set](#) (nrf_t rf, uint8_t mode)
- [void nrf_channel_set](#) (nrf_t rf, uint8_t channel)
- [void nrf_rf_power_set](#) (nrf_t rf, uint8_t rf_power)
- [void nrf_xtal_freq_set](#) (nrf_t rf, uint8_t freq)
- [void nrf_data_rate_set](#) (nrf_t rf, uint8_t data_rate)
- [void nrf_comms_mode_set](#) (nrf_t rf, uint8_t mode)
- [void nrf_single_or_dual_channel_set](#) (nrf_t rf, uint8_t mode)
- [void nrf_crc_status_set](#) (nrf_t rf, uint8_t status)
- [void nrf_crc_length_set](#) (nrf_t rf, uint8_t length)
- [void nrf_address_length_set](#) (nrf_t rf, uint8_t length)
- [void nrf_address1_set](#) (nrf_t rf, [rf_address_t](#) *address)
- [void nrf_address2_set](#) (nrf_t rf, [rf_address_t](#) *address)
- [void nrf_payload_length1_set](#) (nrf_t rf, uint8_t length)

- void `nrf_payload_length2_set` (`nrf_t rf`, `uint8_t length`)
- void `nrf_rf_standby` (`nrf_t rf`)
- void `nrf_rf_enable` (`nrf_t rf`)
- `uint8_t nrf_payload_length1_get` (`nrf_t rf`)
- bool `nrf_data_ready_p` (`nrf_t rf`)

5.63.1 Detailed Description

Author:

Tony Culliford

Date:

7 December 2004

Description: Interface routines for the Nordic nRF2401 transceiver chip

5.63.2 Define Documentation

5.63.2.1 `#define CHAR_BIT 8`

5.63.2.2 `#define NRF_CE_HIGH_SET(DEV) (port_pins_set_high ((DEV) → pins.ce_port, (DEV) → pins.ce_bitmask))`

5.63.2.3 `#define NRF_CE_LOW_SET(DEV) (port_pins_set_low ((DEV) → pins.ce_port, (DEV) → pins.ce_bitmask))`

5.63.2.4 `#define NRF_CONFIGURATION_REGISTER_SIZE 15`

5.63.2.5 `#define NRF_CS_HIGH_SET(DEV) (port_pins_set_high ((DEV) → pins.cs_port, (DEV) → pins.cs_bitmask))`

5.63.2.6 `#define NRF_CS_LOW_SET(DEV) (port_pins_set_low ((DEV) → pins.cs_port, (DEV) → pins.cs_bitmask))`

5.63.2.7 `#define NRF_DATA_READY_P(DEV) (port_pins_read ((DEV) → pins.dr_port, (DEV) → pins.dr_bitmask))`

5.63.2.8 `#define NRF_FULL_CONFIGURE(DEV) (nrf_configure ((DEV), NRF_CONFIGURATION_REGISTER_SIZE))`

5.63.3 Enumeration Type Documentation

5.63.3.1 anonymous enum

Enumerator:

`NRF_CHANNEL_NUMBER_MAX`

5.63.3.2 anonymous enum

Enumerator:

`NRF_CONFIG_DELAY_US`

5.63.3.3 anonymous enum

Enumerator:

NRF_LINE_TIME_ENABLE_US

5.63.3.4 anonymous enum

Enumerator:

NRF_T_SB_ACTIVE

5.63.4 Function Documentation

- 5.63.4.1 void nrf_address1_set ([nrf_t rf](#), [rf_address_t](#) * *address*)
- 5.63.4.2 void nrf_address2_set ([nrf_t rf](#), [rf_address_t](#) * *address*)
- 5.63.4.3 void nrf_address_length_set ([nrf_t rf](#), [uint8_t](#) *length*)
- 5.63.4.4 void nrf_channel_set ([nrf_t rf](#), [uint8_t](#) *channel*)
- 5.63.4.5 void nrf_comms_mode_set ([nrf_t rf](#), [uint8_t](#) *mode*)
- 5.63.4.6 void nrf_configure ([nrf_t rf](#), [uint8_t](#) *size*)
- 5.63.4.7 void nrf_crc_length_set ([nrf_t rf](#), [uint8_t](#) *length*)
- 5.63.4.8 void nrf_crc_status_set ([nrf_t rf](#), [uint8_t](#) *status*)
- 5.63.4.9 void nrf_data_rate_set ([nrf_t rf](#), [uint8_t](#) *data_rate*)
- 5.63.4.10 bool nrf_data_ready_p ([nrf_t rf](#))
- 5.63.4.11 [nrf_t](#) nrf_init ([nrf_obj_t](#) * *dev*, const [nrf_cfg_t](#) * *cfg*)
- 5.63.4.12 [uint8_t](#) nrf_payload_length1_get ([nrf_t rf](#))
- 5.63.4.13 void nrf_payload_length1_set ([nrf_t rf](#), [uint8_t](#) *length*)
- 5.63.4.14 void nrf_payload_length2_set ([nrf_t rf](#), [uint8_t](#) *length*)
- 5.63.4.15 [uint8_t](#) nrf_receive ([nrf_t rf](#), [uint8_t](#) * *data*, [uint8_t](#) *ms_to_wait*)
- 5.63.4.16 void nrf_rf_dir_set ([nrf_t rf](#), [uint8_t](#) *mode*)
- 5.63.4.17 void nrf_rf_enable ([nrf_t rf](#))
- 5.63.4.18 void nrf_rf_power_set ([nrf_t rf](#), [uint8_t](#) *rf_power*)
- 5.63.4.19 void nrf_rf_standby ([nrf_t rf](#))
- 5.63.4.20 bool nrf_setup ([nrf_t rf](#), [uint8_t](#) *payload_size*)
- 5.63.4.21 void nrf_single_or_dual_channel_set ([nrf_t rf](#), [uint8_t](#) *mode*)
- 5.63.4.22 [uint8_t](#) nrf_transmit ([nrf_t rf](#), [rf_address_t](#) * *dst_address*, [uint8_t](#) * *data*, [uint8_t](#) *size*)
- 5.63.4.23 void nrf_xtal_freq_set ([nrf_t rf](#), [uint8_t](#) *freq*)

5.64 nrf2401.h File Reference

```
#include "time.h"
#include "spi.h"
#include "port.h"
#include "config.h"
```

Data Structures

- struct [rf_address_t](#)
- struct [nrf_config_bits_t](#)
- union [nrf_config_t](#)
- struct [nrf_cfg_t](#)
- struct [nrf_pins_t](#)
- struct [nrf_obj_t](#)

Defines

- #define [NRF_PAYLOAD_SIZE](#) 25
- #define [NRF_DEFAULT_ADDRESS](#) {0xFD, 0xCA, 0x7E, 0xA9, 0x52}

Typedefs

- typedef [nrf_obj_t](#) * [nrf_t](#)

Enumerations

- enum { [NRF_TX_MODE](#) = 0, [NRF_RX_MODE](#) = 1 }
- enum { [NRF_DIRECT](#) = 0, [NRF_SHOCKBURST](#) = 1 }
- enum { [NRF_CRC_8](#) = 0, [NRF_CRC_16](#) = 1 }
- enum { [NRF_CRC_DISABLED](#) = 0, [NRF_CRC_ENABLED](#) = 1 }
- enum { [NRF_RF_POWER_20](#), [NRF_RF_POWER_10](#), [NRF_RF_POWER_5](#), [NRF_RF_POWER_0](#) }
- enum { [NRF_SINGLE_CHANNEL](#) = 0, [NRF_DUAL_CHANNEL](#) = 1 }
- enum { [NRF_DATA_250K](#) = 0, [NRF_DATA_1M](#) = 1 }
- enum {
[NRF_XTAL_FREQ_4M](#), [NRF_XTAL_FREQ_8M](#), [NRF_XTAL_FREQ_12M](#), [NRF_XTAL_FREQ_16M](#),
[NRF_XTAL_FREQ_20M](#) }
- enum { [NRF_TIME_OUT_ACK_MS](#) = 10 }
- enum { [NRF_ACK_DELAY_US](#) = 600 }

Functions

- `nrf_t nrf_init` (`nrf_obj_t` *dev, const `nrf_cfg_t` *cfg)
- `bool nrf_setup` (`nrf_t` rf, `uint8_t` payload_size)
- `void nrf_configure` (`nrf_t` rf, `uint8_t` size)
- `uint8_t nrf_transmit` (`nrf_t` rf, `rf_address_t` *address, `uint8_t` *data, `uint8_t` size)
- `uint8_t nrf_receive` (`nrf_t` rf, `uint8_t` *data, `uint8_t` ms_to_wait)
- `void nrf_rf_dir_set` (`nrf_t` rf, `uint8_t` mode)
- `void nrf_channel_set` (`nrf_t` rf, `uint8_t` channel)
- `void nrf_rf_power_set` (`nrf_t` rf, `uint8_t` rf_power)
- `void nrf_xtal_freq_set` (`nrf_t` rf, `uint8_t` freq)
- `void nrf_data_rate_set` (`nrf_t` rf, `uint8_t` data_rate)
- `void nrf_comms_mode_set` (`nrf_t` rf, `uint8_t` mode)
- `void nrf_single_or_dual_channel_set` (`nrf_t` rf, `uint8_t` mode)
- `void nrf_crc_status_set` (`nrf_t` rf, `uint8_t` status)
- `void nrf_crc_length_set` (`nrf_t` rf, `uint8_t` length)
- `void nrf_address_length_set` (`nrf_t` rf, `uint8_t` length)
- `void nrf_address1_set` (`nrf_t` rf, `rf_address_t` *address)
- `void nrf_address2_set` (`nrf_t` rf, `rf_address_t` *address)
- `void nrf_payload_length1_set` (`nrf_t` rf, `uint8_t` length)
- `void nrf_payload_length2_set` (`nrf_t` rf, `uint8_t` length)
- `void nrf_rf_standby` (`nrf_t` rf)
- `void nrf_rf_enable` (`nrf_t` rf)
- `uint8_t nrf_payload_length1_get` (`nrf_t` rf)
- `bool nrf_data_ready_p` (`nrf_t` rf)

5.64.1 Detailed Description

Author:

Tony Culliford

Date:

7 December 2004

Description: Interface routines for the Nordic nRF2401 transceiver chip

5.64.2 Define Documentation

5.64.2.1 `#define NRF_DEFAULT_ADDRESS {0xFD, 0xCA, 0x7E, 0xA9, 0x52}`

5.64.2.2 `#define NRF_PAYLOAD_SIZE 25`

5.64.3 Typedef Documentation

5.64.3.1 `typedef nrf_obj_t* nrf_t`

5.64.4 Enumeration Type Documentation

5.64.4.1 `anonymous enum`

Enumerator:

NRF_TX_MODE

*NRF_RX_MODE***5.64.4.2 anonymous enum****Enumerator:***NRF_DIRECT**NRF_SHOCKBURST***5.64.4.3 anonymous enum****Enumerator:***NRF_CRC_8**NRF_CRC_16***5.64.4.4 anonymous enum****Enumerator:***NRF_CRC_DISABLED**NRF_CRC_ENABLED***5.64.4.5 anonymous enum****Enumerator:***NRF_RF_POWER_20**NRF_RF_POWER_10**NRF_RF_POWER_5**NRF_RF_POWER_0***5.64.4.6 anonymous enum****Enumerator:***NRF_SINGLE_CHANNEL**NRF_DUAL_CHANNEL***5.64.4.7 anonymous enum****Enumerator:***NRF_DATA_250K**NRF_DATA_1M***5.64.4.8 anonymous enum****Enumerator:***NRF_XTAL_FREQ_4M**RF_XTAL_FREQ_8M**NRF_XTAL_FREQ_12M**NRF_XTAL_FREQ_16M**NRF_XTAL_FREQ_20M*

5.64.4.9 anonymous enum

Enumerator:

NRF_TIME_OUT_ACK_MS

5.64.4.10 anonymous enum

Enumerator:

NRF_ACK_DELAY_US

5.64.5 Function Documentation

- 5.64.5.1 void nrf_address1_set ([nrf_t](#) *rf*, [rf_address_t](#) * *address*)
- 5.64.5.2 void nrf_address2_set ([nrf_t](#) *rf*, [rf_address_t](#) * *address*)
- 5.64.5.3 void nrf_address_length_set ([nrf_t](#) *rf*, [uint8_t](#) *length*)
- 5.64.5.4 void nrf_channel_set ([nrf_t](#) *rf*, [uint8_t](#) *channel*)
- 5.64.5.5 void nrf_comms_mode_set ([nrf_t](#) *rf*, [uint8_t](#) *mode*)
- 5.64.5.6 void nrf_configure ([nrf_t](#) *rf*, [uint8_t](#) *size*)
- 5.64.5.7 void nrf_crc_length_set ([nrf_t](#) *rf*, [uint8_t](#) *length*)
- 5.64.5.8 void nrf_crc_status_set ([nrf_t](#) *rf*, [uint8_t](#) *status*)
- 5.64.5.9 void nrf_data_rate_set ([nrf_t](#) *rf*, [uint8_t](#) *data_rate*)
- 5.64.5.10 bool nrf_data_ready_p ([nrf_t](#) *rf*)
- 5.64.5.11 [nrf_t](#) nrf_init ([nrf_obj_t](#) * *dev*, const [nrf_cfg_t](#) * *cfg*)
- 5.64.5.12 [uint8_t](#) nrf_payload_length1_get ([nrf_t](#) *rf*)
- 5.64.5.13 void nrf_payload_length1_set ([nrf_t](#) *rf*, [uint8_t](#) *length*)
- 5.64.5.14 void nrf_payload_length2_set ([nrf_t](#) *rf*, [uint8_t](#) *length*)
- 5.64.5.15 [uint8_t](#) nrf_receive ([nrf_t](#) *rf*, [uint8_t](#) * *data*, [uint8_t](#) *ms_to_wait*)
- 5.64.5.16 void nrf_rf_dir_set ([nrf_t](#) *rf*, [uint8_t](#) *mode*)
- 5.64.5.17 void nrf_rf_enable ([nrf_t](#) *rf*)
- 5.64.5.18 void nrf_rf_power_set ([nrf_t](#) *rf*, [uint8_t](#) *rf_power*)
- 5.64.5.19 void nrf_rf_standby ([nrf_t](#) *rf*)
- 5.64.5.20 bool nrf_setup ([nrf_t](#) *rf*, [uint8_t](#) *payload_size*)
- 5.64.5.21 void nrf_single_or_dual_channel_set ([nrf_t](#) *rf*, [uint8_t](#) *mode*)
- 5.64.5.22 [uint8_t](#) nrf_transmit ([nrf_t](#) *rf*, [rf_address_t](#) * *address*, [uint8_t](#) * *data*, [uint8_t](#) *size*)
- 5.64.5.23 void nrf_xtal_freq_set ([nrf_t](#) *rf*, [uint8_t](#) *freq*)

5.65 nrf_config.h File Reference

```
#include "nrf2401.h"
```

Defines

- #define [RF_RX_MODE_SET](#)(DEV) nrf_rf_dir_set (DEV, NRF_RX_MODE)
- #define [RF_TX_MODE_SET](#)(DEV) nrf_rf_dir_set (DEV, NRF_TX_MODE)
- #define [RF_DEVICE_ENABLE](#)(DEV) nrf_rf_enable (DEV)
- #define [RF_DEVICE_DISABLE](#)(DEV) nrf_rf_standby (DEV)
- #define [RF_DEVICE_CHANNEL_SET](#)(DEV, CHANNEL) nrf_channel_set (DEV, (CHANNEL))
- #define [RF_DEVICE_ADDRESS_SET](#)(DEV, ADDRESS) nrf_address1_set (DEV, (ADDRESS))
- #define [RF_TRANSMIT](#)(DEV, ADDRESS, DATA, SIZE) nrf_transmit (DEV, ADDRESS, DATA, SIZE)
- #define [RF_RECEIVE](#)(DEV, DATA, MS_TO_WAIT) nrf_receive (DEV, DATA, MS_TO_WAIT)
- #define [RF_INIT](#)(DEV, CFG) nrf_init (DEV, CFG)
- #define [RF_SETUP](#)(DEV, SIZE) nrf_setup (DEV, SIZE)
- #define [RF_DATA_READY_P](#)(DEV) nrf_data_ready_p (DEV)

Typedefs

- typedef [nrf_t](#) rf_t
- typedef [nrf_obj_t](#) rf_obj_t
- typedef [nrf_cfg_t](#) rf_cfg_t

5.65.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.65.2 Define Documentation

- 5.65.2.1 `#define RF_DATA_READY_P(DEV) nrf_data_ready_p (DEV)`
- 5.65.2.2 `#define RF_DEVICE_ADDRESS_SET(DEV, ADDRESS) nrf_address1_set (DEV, (ADDRESS))`
- 5.65.2.3 `#define RF_DEVICE_CHANNEL_SET(DEV, CHANNEL) nrf_channel_set (DEV, (CHANNEL))`
- 5.65.2.4 `#define RF_DEVICE_DISABLE(DEV) nrf_rf_standby (DEV)`
- 5.65.2.5 `#define RF_DEVICE_ENABLE(DEV) nrf_rf_enable (DEV)`
- 5.65.2.6 `#define RF_INIT(DEV, CFG) nrf_init (DEV, CFG)`
- 5.65.2.7 `#define RF_RECEIVE(DEV, DATA, MS_TO_WAIT) nrf_receive (DEV, DATA, MS_TO_WAIT)`
- 5.65.2.8 `#define RF_RX_MODE_SET(DEV) nrf_rf_dir_set (DEV, NRF_RX_MODE)`
- 5.65.2.9 `#define RF_SETUP(DEV, SIZE) nrf_setup (DEV, SIZE)`
- 5.65.2.10 `#define RF_TRANSMIT(DEV, ADDRESS, DATA, SIZE) nrf_transmit (DEV, ADDRESS, DATA, SIZE)`
- 5.65.2.11 `#define RF_TX_MODE_SET(DEV) nrf_rf_dir_set (DEV, NRF_TX_MODE)`

5.65.3 Typedef Documentation

- 5.65.3.1 `typedef nrf_cfg_t rf_cfg_t`
- 5.65.3.2 `typedef nrf_obj_t rf_obj_t`
- 5.65.3.3 `typedef nrf_t rf_t`

5.66 pga.c File Reference

```
#include "pga.h"
```

Defines

- #define [PGA_TRANSPARENT](#)
- #define [PGA_INSN_REGISTER_WRITE](#)(REG) ((PGA_INSN_WRITE << 5) | (REG))
- #define [PGA_INSN_GAIN_REGISTER_WRITE](#) [PGA_INSN_REGISTER_WRITE](#) ([PGA_GAIN_REGISTER](#))
- #define [PGA_INSN_CHANNEL_REGISTER_WRITE](#) [PGA_INSN_REGISTER_WRITE](#) ([PGA_CHANNEL_REGISTER](#))
- #define [PGA_PUTC](#)(val) [spi_putc](#)(val)
- #define [PGA_GETC](#)() [spi_getc](#)()
- #define [PGA_SELECT](#)(PGA) [port_pins_set_low](#) ((PGA) → [cs_port](#), (PGA) → [cs_bitmask](#))
- #define [PGA_DESELECT](#)(PGA) [port_pins_set_high](#) ((PGA) → [cs_port](#), (PGA) → [cs_bitmask](#))

Enumerations

- enum { [PGA_GAIN_REGISTER](#) = 0, [PGA_CHANNEL_REGISTER](#) = 1 }
- enum { [PGA_INSN_NOP](#) = 0, [PGA_INSN_SHUTDOWN](#) = 1, [PGA_INSN_WRITE](#) = 2 }

Functions

- static void [pga_send_command](#) ([pga_t](#) pga, uint8_t command_byte_1, uint8_t command_byte_2)
- void [pga_gain_set](#) ([pga_t](#) pga, [pga_gain_t](#) gain)
- void [pga_channel_set](#) ([pga_t](#) pga, [pga_channel_t](#) channel)
- void [pga_shutdown](#) ([pga_t](#) pga)
- void [pga_startup](#) ([pga_t](#) pga)
- void [pga_chip_select](#) ([pga_t](#) pga)
- void [pga_chip_deselect](#) ([pga_t](#) pga)
- [pga_t](#) [pga_init](#) ([pga_obj_t](#) *pga, const [pga_cfg_t](#) *cfg)

5.66.1 Detailed Description

Author:

Micahel Hayes / Tony Culliford

Date:

8 February 2005

Description: Interface routines for Microchip PGAs (MCP6S21)

5.66.2 Define Documentation

5.66.2.1 `#define PGA_DESELECT(PGA) port_pins_set_high ((PGA) → cs_port, (PGA) → cs_bitmask)`

5.66.2.2 `#define PGA_GETC() spi_getc()`

5.66.2.3 `#define PGA_INSN_CHANNEL_REGISTER_WRITE PGA_INSN_REGISTER_WRITE (PGA_CHANNEL_REGISTER)`

5.66.2.4 `#define PGA_INSN_GAIN_REGISTER_WRITE PGA_INSN_REGISTER_WRITE (PGA_GAIN_REGISTER)`

5.66.2.5 `#define PGA_INSN_REGISTER_WRITE(REG) ((PGA_INSN_WRITE << 5) | (REG))`

5.66.2.6 `#define PGA_PUTC(val) spi_putc(val)`

5.66.2.7 `#define PGA_SELECT(PGA) port_pins_set_low ((PGA) → cs_port, (PGA) → cs_bitmask)`

5.66.2.8 `#define PGA_TRANSPARENT`

5.66.3 Enumeration Type Documentation

5.66.3.1 anonymous enum

Enumerator:

PGA_GAIN_REGISTER

PGA_CHANNEL_REGISTER

5.66.3.2 anonymous enum

Enumerator:

PGA_INSN_NOP

PGA_INSN_SHUTDOWN

PGA_INSN_WRITE

5.66.4 Function Documentation

5.66.4.1 void pga_channel_set ([pga_t](#) *pga*, [pga_channel_t](#) *channel*)

5.66.4.2 void pga_chip_deselect ([pga_t](#) *pga*)

5.66.4.3 void pga_chip_select ([pga_t](#) *pga*)

5.66.4.4 void pga_gain_set ([pga_t](#) *pga*, [pga_gain_t](#) *gain*)

5.66.4.5 [pga_t](#) pga_init ([pga_obj_t](#) * *pga*, const [pga_cfg_t](#) * *cfg*)

5.66.4.6 static void pga_send_command ([pga_t](#) *pga*, [uint8_t](#) *command_byte_1*, [uint8_t](#) *command_byte_2*) [static]

5.66.4.7 void pga_shutdown ([pga_t](#) *pga*)

5.66.4.8 void pga_startup ([pga_t](#) *pga*)

5.67 pga.h File Reference

```
#include "spi.h"
#include "config.h"
#include "port.h"
```

Data Structures

- struct [pga_cfg_t](#)
- struct [pga_private_t](#)

Defines

- #define [PGA_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}

Typedefs

- typedef [pga_private_t](#) [pga_obj_t](#)
- typedef [pga_obj_t](#) * [pga_t](#)

Enumerations

- enum [pga_channel_t](#) {
 [PGA_CHANNEL_0](#) = 0, [PGA_CHANNEL_1](#) = 1, [PGA_CHANNEL_2](#) = 2, [PGA_CHANNEL_3](#) = 3,
 [PGA_CHANNEL_4](#) = 4, [PGA_CHANNEL_5](#) = 5, [PGA_CHANNEL_6](#) = 6, [PGA_CHANNEL_7](#) = 7 }
• enum [pga_gain_t](#) {
 [PGA_GAIN_1](#) = 0, [PGA_GAIN_2](#) = 1, [PGA_GAIN_4](#) = 2, [PGA_GAIN_5](#) = 3,
 [PGA_GAIN_8](#) = 4, [PGA_GAIN_10](#) = 5, [PGA_GAIN_16](#) = 6, [PGA_GAIN_32](#) = 7 }

Functions

- void [pga_gain_set](#) ([pga_t](#) pga, [pga_gain_t](#) gain)
- void [pga_channel_set](#) ([pga_t](#) pga, [pga_channel_t](#) channel)
- void [pga_startup](#) ([pga_t](#) pga)
- void [pga_shutdown](#) ([pga_t](#) pga)
- void [pga_chip_select](#) ([pga_t](#) pga)
- void [pga_chip_deselect](#) ([pga_t](#) pga)
- [pga_t](#) [pga_init](#) ([pga_obj_t](#) *dev, const [pga_cfg_t](#) *cfg)

5.67.1 Detailed Description

Author:

Tony Culliford

Date:

8 February 2005

Description: Interface routines for Microchip PGAs (MCP6S2X)

5.67.2 Define Documentation

5.67.2.1 `#define PGA_CFG(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}`

5.67.3 Typedef Documentation

5.67.3.1 typedef `pga_private_t` `pga_obj_t`

5.67.3.2 typedef `pga_obj_t*` `pga_t`

5.67.4 Enumeration Type Documentation

5.67.4.1 enum `pga_channel_t`

Enumerator:

`PGA_CHANNEL_0`
`PGA_CHANNEL_1`
`PGA_CHANNEL_2`
`PGA_CHANNEL_3`
`PGA_CHANNEL_4`
`PGA_CHANNEL_5`
`PGA_CHANNEL_6`
`PGA_CHANNEL_7`

5.67.4.2 enum `pga_gain_t`

Enumerator:

`PGA_GAIN_1`
`PGA_GAIN_2`
`PGA_GAIN_4`
`PGA_GAIN_5`
`PGA_GAIN_8`
`PGA_GAIN_10`
`PGA_GAIN_16`
`PGA_GAIN_32`

5.67.5 Function Documentation

5.67.5.1 void `pga_channel_set` (`pga_t` *pga*, `pga_channel_t` *channel*)

5.67.5.2 void `pga_chip_deselect` (`pga_t` *pga*)

5.67.5.3 void `pga_chip_select` (`pga_t` *pga*)

5.67.5.4 void `pga_gain_set` (`pga_t` *pga*, `pga_gain_t` *gain*)

5.67.5.5 `pga_t` `pga_init` (`pga_obj_t` * *dev*, const `pga_cfg_t` * *cfg*)

5.67.5.6 void `pga_shutdown` (`pga_t` *pga*)

5.67.5.7 void `pga_startup` (`pga_t` *pga*)

5.68 piezo.c File Reference

```
#include "piezo.h"
```

Functions

- [piezo_t piezo_init](#) (const [piezo_cfg_t](#) *cfg)

5.68.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

12 March 2003

5.68.2 Function Documentation

5.68.2.1 [piezo_t piezo_init](#) (const [piezo_cfg_t](#) * *cfg*)

5.69 piezo.h File Reference

```
#include "config.h"
#include "delay.h"
#include "port.h"
```

Data Structures

- struct [piezo_cfg_t](#)

Defines

- #define [PIEZO_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}

Typedefs

- typedef const [piezo_cfg_t](#) [piezo_obj_t](#)
- typedef [piezo_obj_t](#) * [piezo_t](#)

Functions

- [piezo_t](#) [piezo_init](#) (const [piezo_cfg_t](#) *cfg)
- static void [piezo_set](#) ([piezo_t](#) piezo, uint8_t val)

5.69.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

12 March 2003

5.69.2 Define Documentation

5.69.2.1 #define [PIEZO_CFG](#)(PORT, PORTBIT) {(PORT), BIT (PORTBIT)}

5.69.3 Typedef Documentation

5.69.3.1 typedef const [piezo_cfg_t](#) [piezo_obj_t](#)

5.69.3.2 typedef [piezo_obj_t](#)* [piezo_t](#)

5.69.4 Function Documentation

5.69.4.1 [piezo_t](#) [piezo_init](#) (const [piezo_cfg_t](#) * *cfg*)

5.69.4.2 static void [piezo_set](#) ([piezo_t](#) *piezo*, uint8_t *val*) [static]

5.70 piezo_beep.c File Reference

Piezo beeping routines. Note these block.

```
#include "piezo.h"
```

Defines

- #define [PIEZO_SHORT_BEEP_TIME](#) 30
- #define [PIEZO_LONG_BEEP_TIME](#) 200
- #define [PIEZO_BEEP_PERIOD](#) 200

Functions

- void [piezo_beep](#) ([piezo_t](#) piezo, uint16_t duration)
- void [piezo_beep_short](#) ([piezo_t](#) piezo)
- void [piezo_beep_long](#) ([piezo_t](#) piezo)

5.70.1 Detailed Description

Piezo beeping routines. Note these block.

Author:

M. P. Hayes, UCECE

Date:

12 April 2007

5.70.2 Define Documentation

5.70.2.1 #define [PIEZO_BEEP_PERIOD](#) 200

5.70.2.2 #define [PIEZO_LONG_BEEP_TIME](#) 200

5.70.2.3 #define [PIEZO_SHORT_BEEP_TIME](#) 30

5.70.3 Function Documentation

5.70.3.1 void [piezo_beep](#) ([piezo_t](#) piezo, uint16_t duration)

5.70.3.2 void [piezo_beep_long](#) ([piezo_t](#) piezo)

5.70.3.3 void [piezo_beep_short](#) ([piezo_t](#) piezo)

5.71 piezo_beep.h File Reference

```
#include "piezo.h"
```

Functions

- void [piezo_beep](#) ([piezo_t](#) dev, uint16_t duration)
- void [piezo_beep_short](#) ([piezo_t](#) dev)
- void [piezo_beep_long](#) ([piezo_t](#) dev)

5.71.1 Function Documentation

5.71.1.1 void [piezo_beep](#) ([piezo_t](#) *dev*, uint16_t *duration*)

5.71.1.2 void [piezo_beep_long](#) ([piezo_t](#) *dev*)

5.71.1.3 void [piezo_beep_short](#) ([piezo_t](#) *dev*)

5.72 rf.c File Reference

```
#include <limits.h>
#include <string.h>
#include <stdio.h>
#include "rf.h"
#include "time.h"
#include "delay.h"
#include "port.h"
```

Defines

- #define [RF_DEBUG](#) 0
- #define [HIGH_BYTE](#)(x) ((x) >> 8)
- #define [LOW_BYTE](#)(x) ((x) & 0xFF)

Functions

- static void [rf_address_make](#) ([rf_address_t](#) *address, [rf_id_t](#) id)
- void [rf_node_make](#) ([rf_node_t](#) *node, [rf_id_t](#) id, uint8_t channel)
- void [rf_write_setup](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node)
- void [rf_standby](#) ([rf_t](#) rf)
- [rf_size_t](#) [rf_transmit](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, uint8_t *data, [rf_size_t](#) size)
- void [rf_read_enable](#) ([rf_t](#) rf)
- void [rf_read_setup](#) ([rf_t](#) rf, [rf_node_t](#) *node)
- [rf_size_t](#) [rf_receive](#) ([rf_t](#) rf, uint8_t *data, uint8_t ms_to_wait)
- [rf_size_t](#) [rf_write](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, uint8_t *data, [rf_size_t](#) size)
- [rf_size_t](#) [rf_read](#) ([rf_t](#) rf, uint8_t *data, [rf_size_t](#) size, uint8_t ms_to_wait)
- [rf_t](#) [rf_setup](#) ([rf_obj_t](#) *dev, [rf_cfg_t](#) *cfg)
- uint8_t [rf_acknowledge_wait](#) ([rf_t](#) rf, [rf_node_t](#) *node, uint8_t command, [time_t](#) *timestamp)
- void [rf_acknowledge](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, uint8_t command)
- [rf_t](#) [rf_init](#) ([rf_obj_t](#) *dev, [rf_cfg_t](#) *cfg)
- uint16_t [rf_write_data](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, uint8_t *data, uint16_t size)
- bool [rf_read_ready_p](#) ([rf_t](#) rf)
- [rf_size_t](#) [rf_command](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, [rf_cmd_t](#) command, void *data, [rf_size_t](#) data_size)
- void [rf_command_no_ack](#) ([rf_t](#) rf, [rf_node_t](#) *dst_node, [rf_cmd_t](#) command, void *data, [rf_size_t](#) data_size)

5.72.1 Detailed Description

Author:

Tony Culliford

Date:

8 December 2004

Description: Higher-level routines for interfacing to a wireless link.

5.72.2 Define Documentation

5.72.2.1 `#define HIGH_BYTE(x) ((x) >> 8)`

5.72.2.2 `#define LOW_BYTE(x) ((x) & 0xFF)`

5.72.2.3 `#define RF_DEBUG 0`

5.72.3 Function Documentation

5.72.3.1 `void rf_acknowledge (rf_t rf, rf_node_t * dst_node, uint8_t command)`

5.72.3.2 `uint8_t rf_acknowledge_wait (rf_t rf, rf_node_t * node, uint8_t command, time_t * timestamp)`

5.72.3.3 `static void rf_address_make (rf_address_t * address, rf_id_t id) [static]`

5.72.3.4 `rf_size_t rf_command (rf_t rf, rf_node_t * dst_node, rf_cmd_t command, void * data, rf_size_t data_size)`

5.72.3.5 `void rf_command_no_ack (rf_t rf, rf_node_t * dst_node, rf_cmd_t command, void * data, rf_size_t data_size)`

5.72.3.6 `rf_t rf_init (rf_obj_t * dev, rf_cfg_t * cfg)`

5.72.3.7 `void rf_node_make (rf_node_t * node, rf_id_t id, uint8_t channel)`

5.72.3.8 `rf_size_t rf_read (rf_t rf, uint8_t * data, rf_size_t size, uint8_t ms_to_wait)`

5.72.3.9 `void rf_read_enable (rf_t rf)`

5.72.3.10 `bool rf_read_ready_p (rf_t rf)`

5.72.3.11 `void rf_read_setup (rf_t rf, rf_node_t * node)`

5.72.3.12 `rf_size_t rf_receive (rf_t rf, uint8_t * data, uint8_t ms_to_wait)`

5.72.3.13 `rf_t rf_setup (rf_obj_t * dev, rf_cfg_t * cfg)`

5.72.3.14 `void rf_standby (rf_t rf)`

5.72.3.15 `rf_size_t rf_transmit (rf_t rf, rf_node_t * dst_node, uint8_t * data, rf_size_t size)`

5.72.3.16 `rf_size_t rf_write (rf_t rf, rf_node_t * dst_node, uint8_t * data, rf_size_t size)`

5.72.3.17 `uint16_t rf_write_data (rf_t rf, rf_node_t * dst_node, uint8_t * data, uint16_t size)`

5.72.3.18 `void rf_write_setup (rf_t rf, rf_node_t * dst_node)`

5.73 rf.h File Reference

```
#include "nrf_config.h"
```

Data Structures

- struct [rf_node_t](#)
- struct [rf_probe_t](#)

Defines

- #define [RF_PROBE_NOT_FOUND](#) 0xFD
- #define [RF_PAYLOAD_SIZE](#) NRF_PAYLOAD_SIZE
- #define [RF_RETRIES_MAX](#) 30

Typedefs

- typedef uint8_t [rf_channel_t](#)
- typedef uint8_t [rf_size_t](#)

Enumerations

- enum [rf_id_t](#) { [RF_BROADCAST_SLAVE_ID](#) = 0xFE, [RF_MASTER_ID](#) = 0xFF }
- enum { [RF_BROADCAST_CHANNEL](#) = 0 }
- enum { [RF_READ_WAIT_MS](#) = 2 }
- enum { [RF_WRITE_WAIT_MS](#) = 4 }
- enum { [RF_READY_WAIT_US](#) = 100 }
- enum { [RF_READY_WAIT2_US](#) = 300 }
- enum { [RF_DEVICE_ID_SIZE](#) = 8 }
- enum [rf_cmd_t](#) {
[RF_CMD_ACK](#) = 0x61, [RF_CMD_DATA_START](#), [RF_CMD_CHANNEL_SET](#), [RF_CMD_TIMESTAMP_REQ](#),
[RF_CMD_BROADCAST_MODE](#), [RF_CMD_BROADCAST_EXIT](#), [RF_CMD_DEVICE_ID_GET](#), [RF_CMD_DEVICE_ID_RESPONSE](#),
[RF_CMD_ENUMERATE_DEVICE](#), [RF_CMD_RESEND_PACKET](#), [RF_CMD_ALL_PACKETS_RECEIVED](#), [RF_CMD_SIZE](#) }

Functions

- [rf_address_t](#) * [rf_address_calc](#) ([rf_id_t](#) id)
- [rf_t](#) [rf_init](#) ([rf_obj_t](#) *rf, [rf_cfg_t](#) *cfg)
- uint8_t [rf_acknowledge_wait](#) ([rf_t](#) rf, [rf_node_t](#) *node, [rf_cmd_t](#) command, [time_t](#) *timestamp)
- void [rf_acknowledge](#) ([rf_t](#) rf, [rf_node_t](#) *node, [rf_cmd_t](#) command)
- uint8_t [rf_probes_enumerate](#) ([rf_t](#) rf, [rf_probe_t](#) *probes, uint8_t probes_max)
- [rf_id_t](#) [rf_enumeration_response](#) ([rf_t](#) rf, uint8_t *device_id, uint8_t *version)
- void [rf_read_enable](#) ([rf_t](#) rf)
- void [rf_write_setup](#) ([rf_t](#) rf, [rf_node_t](#) *node)

- `rf_size_t rf_write (rf_t rf, rf_node_t *node, uint8_t *data, rf_size_t size)`
- `uint16_t rf_write_data (rf_t rf, rf_node_t *node, uint8_t *data, uint16_t size)`
- `void rf_read_setup (rf_t rf, rf_node_t *node)`
- `bool rf_read_ready_p (rf_t rf)`
- `uint16_t rf_read_data (rf_t rf, rf_node_t *node, uint8_t *data, uint16_t size)`
- `rf_size_t rf_read (rf_t rf, uint8_t *data, rf_size_t size, uint8_t ms_to_wait)`
- `void rf_node_make (rf_node_t *node, rf_id_t id, rf_channel_t channel)`
- `rf_size_t rf_transmit (rf_t rf, rf_node_t *dst_node, uint8_t *data, rf_size_t size)`
- `rf_size_t rf_command (rf_t rf, rf_node_t *dst_node, rf_cmd_t command, void *data, rf_size_t data_size)`
- `void rf_command_no_ack (rf_t rf, rf_node_t *dst_node, rf_cmd_t command, void *data, rf_size_t data_size)`
- `void rf_standby (rf_t rf)`

5.73.1 Detailed Description

Author:

Tony Culliford / Michael Hayes

Date:

8 December 2004

Description: Higher-level routines for interfacing to a wireless link.

5.73.2 Define Documentation

5.73.2.1 `#define RF_PAYLOAD_SIZE NRF_PAYLOAD_SIZE`

5.73.2.2 `#define RF_PROBE_NOT_FOUND 0xFD`

5.73.2.3 `#define RF_RETRIES_MAX 30`

5.73.3 Typedef Documentation

5.73.3.1 `typedef uint8_t rf_channel_t`

5.73.3.2 `typedef uint8_t rf_size_t`

5.73.4 Enumeration Type Documentation

5.73.4.1 anonymous enum

Enumerator:

RF_BROADCAST_CHANNEL

5.73.4.2 anonymous enum

Enumerator:

RF_READ_WAIT_MS

5.73.4.3 anonymous enum

Enumerator:

RF_WRITE_WAIT_MS

5.73.4.4 anonymous enum

Enumerator:

RF_READY_WAIT_US

5.73.4.5 anonymous enum

Enumerator:

RF_READY_WAIT2_US

5.73.4.6 anonymous enum

Enumerator:

RF_DEVICE_ID_SIZE

5.73.4.7 enum [rf_cmd_t](#)

Enumerator:

RF_CMD_ACK

RF_CMD_DATA_START

RF_CMD_CHANNEL_SET

RF_CMD_TIMESTAMP_REQ

RF_CMD_BROADCAST_MODE

RF_CMD_BROADCAST_EXIT

RF_CMD_DEVICE_ID_GET

RF_CMD_DEVICE_ID_RESPONSE

RF_CMD_ENUMERATE_DEVICE

RF_CMD_RESEND_PACKET

RF_CMD_ALL_PACKETS_RECEIVED

RF_CMD_SIZE

5.73.4.8 enum [rf_id_t](#)

Enumerator:

RF_BROADCAST_SLAVE_ID

RF_MASTER_ID

5.73.5 Function Documentation

- 5.73.5.1 void rf_acknowledge ([rf_t](#) rf, [rf_node_t](#) * node, [rf_cmd_t](#) command)
- 5.73.5.2 uint8_t rf_acknowledge_wait ([rf_t](#) rf, [rf_node_t](#) * node, [rf_cmd_t](#) command, [time_t](#) * timestamp)
- 5.73.5.3 [rf_address_t](#)* rf_address_calc ([rf_id_t](#) id)
- 5.73.5.4 [rf_size_t](#) rf_command ([rf_t](#) rf, [rf_node_t](#) * dst_node, [rf_cmd_t](#) command, void * data, [rf_size_t](#) data_size)
- 5.73.5.5 void rf_command_no_ack ([rf_t](#) rf, [rf_node_t](#) * dst_node, [rf_cmd_t](#) command, void * data, [rf_size_t](#) data_size)
- 5.73.5.6 [rf_id_t](#) rf_enumeration_response ([rf_t](#) rf, uint8_t * device_id, uint8_t * version)
- 5.73.5.7 [rf_t](#) rf_init ([rf_obj_t](#) * rf, [rf_cfg_t](#) * cfg)
- 5.73.5.8 void rf_node_make ([rf_node_t](#) * node, [rf_id_t](#) id, [rf_channel_t](#) channel)
- 5.73.5.9 uint8_t rf_probes_enumerate ([rf_t](#) rf, [rf_probe_t](#) * probes, uint8_t probes_max)
- 5.73.5.10 [rf_size_t](#) rf_read ([rf_t](#) rf, uint8_t * data, [rf_size_t](#) size, uint8_t ms_to_wait)
- 5.73.5.11 uint16_t rf_read_data ([rf_t](#) rf, [rf_node_t](#) * node, uint8_t * data, uint16_t size)
- 5.73.5.12 void rf_read_enable ([rf_t](#) rf)
- 5.73.5.13 bool rf_read_ready_p ([rf_t](#) rf)
- 5.73.5.14 void rf_read_setup ([rf_t](#) rf, [rf_node_t](#) * node)
- 5.73.5.15 void rf_standby ([rf_t](#) rf)
- 5.73.5.16 [rf_size_t](#) rf_transmit ([rf_t](#) rf, [rf_node_t](#) * dst_node, uint8_t * data, [rf_size_t](#) size)
- 5.73.5.17 [rf_size_t](#) rf_write ([rf_t](#) rf, [rf_node_t](#) * node, uint8_t * data, [rf_size_t](#) size)
- 5.73.5.18 uint16_t rf_write_data ([rf_t](#) rf, [rf_node_t](#) * node, uint8_t * data, uint16_t size)
- 5.73.5.19 void rf_write_setup ([rf_t](#) rf, [rf_node_t](#) * node)

5.74 rf_master.c File Reference

```
#include <string.h>
#include "rf.h"
#include "delay.h"
#include "port.h"
#include "target.h"
```

Defines

- `#define RF_PROBE_RESPONSE_WAIT_MS 100`
- `#define RF_PROBE_RESPONSE_WAIT_NEXT_MS 5`
- `#define RF_ENUMERATE_REPEAT_NUM 3`
- `#define RF_UNACK_CMD_REPEAT_NUM 3`
- `#define RF_UNACK_CMD_REPEAT_DELAY_MS 2`

Functions

- `static uint8_t rf_probes_search (uint8_t *device_id, rf_probe_t *probes, uint8_t probes_num)`
- `uint8_t rf_probes_enumerate (rf_t rf, rf_probe_t *probes, uint8_t probes_max)`
- `uint16_t rf_read_data (rf_t rf, rf_node_t *node, uint8_t *data, uint16_t size)`

5.74.1 Detailed Description

Author:

MPH

Description: Stripped from original rf file wireless.c to handle rf functions only used by the master node.

5.74.2 Define Documentation

5.74.2.1 `#define RF_ENUMERATE_REPEAT_NUM 3`

5.74.2.2 `#define RF_PROBE_RESPONSE_WAIT_MS 100`

5.74.2.3 `#define RF_PROBE_RESPONSE_WAIT_NEXT_MS 5`

5.74.2.4 `#define RF_UNACK_CMD_REPEAT_DELAY_MS 2`

5.74.2.5 `#define RF_UNACK_CMD_REPEAT_NUM 3`

5.74.3 Function Documentation

5.74.3.1 `uint8_t rf_probes_enumerate (rf_t rf, rf_probe_t * probes, uint8_t probes_max)`

5.74.3.2 `static uint8_t rf_probes_search (uint8_t * device_id, rf_probe_t * probes, uint8_t probes_num) [static]`

5.74.3.3 `uint16_t rf_read_data (rf_t rf, rf_node_t * node, uint8_t * data, uint16_t size)`

5.75 rf_slave.c File Reference

```
#include <string.h>
#include "rf.h"
#include "delay.h"
```

Enumerations

- enum { [RF_PROBE_ENUMERATE_WAIT_MS](#) = 5 }

Functions

- [rf_id_t rf_enumeration_response](#) ([rf_t](#) rf, uint8_t *device_id, uint8_t *version)

5.75.1 Detailed Description

Author:

Michael Hayes

Date:

25 March 2005

Description: Slave only wireless routines.

5.75.2 Enumeration Type Documentation

5.75.2.1 anonymous enum

Enumerator:

[RF_PROBE_ENUMERATE_WAIT_MS](#)

5.75.3 Function Documentation

5.75.3.1 [rf_id_t rf_enumeration_response](#) ([rf_t](#) rf, uint8_t * device_id, uint8_t * version)

5.76 ring.c File Reference

Ring buffer implementation.

```
#include <string.h>
#include "ring.h"
```

Defines

- #define [RING_SIZE](#)(RING) ((RING) → end - (RING) → top)
- #define [RING_READ_NUM](#)(RING, TMP)
- #define [RING_WRITE_NUM](#)(RING, TMP) (RING_SIZE (RING) - RING_READ_NUM (RING, TMP) - 1)
- #define [RING_FULL_P](#)(RING, TMP) (RING_WRITE_NUM (RING, TMP) == 0)
- #define [RING_EMPTY_P](#)(RING) ((RING) → in == (RING) → out)

Functions

- [ring_size_t ring_empty_p](#) ([ring_t](#) *ring)
- [ring_size_t ring_read_num](#) ([ring_t](#) *ring)
- [ring_size_t ring_write_num](#) ([ring_t](#) *ring)
- [ring_size_t ring_init](#) ([ring_t](#) *ring, void *buffer, [ring_size_t](#) size)
- [ring_size_t ring_read](#) ([ring_t](#) *ring, void *buffer, [ring_size_t](#) num)
- [ring_size_t ring_write](#) ([ring_t](#) *ring, const void *buffer, [ring_size_t](#) num)

5.76.1 Detailed Description

Ring buffer implementation.

Author:

M. P. Hayes, UCECE

Date:

15 May 2000

5.76.2 Define Documentation

5.76.2.1 #define [RING_EMPTY_P](#)(RING) ((RING) → in == (RING) → out)

5.76.2.2 #define [RING_FULL_P](#)(RING, TMP) (RING_WRITE_NUM (RING, TMP) == 0)

5.76.2.3 #define [RING_READ_NUM](#)(RING, TMP)

Value:

```
((TMP) = ((RING)→in - (RING)→out)) < 0 \
? (TMP) + RING_SIZE (RING) : (TMP))
```


5.76.2.4 `#define RING_SIZE(RING) ((RING) → end - (RING) → top)`

5.76.2.5 `#define RING_WRITE_NUM(RING, TMP) (RING_SIZE (RING) - RING_READ_NUM (RING, TMP) - 1)`

5.76.3 Function Documentation

5.76.3.1 `ring_size_t ring_empty_p (ring_t * ring)`

5.76.3.2 `ring_size_t ring_init (ring_t * ring, void * buffer, ring_size_t size)`

5.76.3.3 `ring_size_t ring_read (ring_t * ring, void * buffer, ring_size_t num)`

5.76.3.4 `ring_size_t ring_read_num (ring_t * ring)`

5.76.3.5 `ring_size_t ring_write (ring_t * ring, const void * buffer, ring_size_t num)`

5.76.3.6 `ring_size_t ring_write_num (ring_t * ring)`

5.77 ring.h File Reference

Ring buffer interface.

```
#include "config.h"
```

Data Structures

- struct [ring_struct](#)

Typedefs

- typedef uint8_t [ring_size_t](#)
- typedef [ring_struct](#) [ring_t](#)

Functions

- bool [ring_empty_p](#) ([ring_t](#) *ring)
- [ring_size_t](#) [ring_read](#) ([ring_t](#) *ring, void *buffer, [ring_size_t](#) num)
- [ring_size_t](#) [ring_write](#) ([ring_t](#) *ring, const void *buffer, [ring_size_t](#) num)
- [ring_size_t](#) [ring_init](#) ([ring_t](#) *ring, void *buffer, [ring_size_t](#) size)
- [ring_size_t](#) [ring_read_num](#) ([ring_t](#) *ring)
- [ring_size_t](#) [ring_write_num](#) ([ring_t](#) *ring)

5.77.1 Detailed Description

Ring buffer interface.

Author:

M. P. Hayes, UCECE

Date:

15 May 2000

5.77.2 Typedef Documentation

5.77.2.1 typedef uint8_t [ring_size_t](#)

5.77.2.2 typedef struct [ring_struct](#) [ring_t](#)

5.77.3 Function Documentation

5.77.3.1 bool [ring_empty_p](#) ([ring_t](#) * *ring*)

5.77.3.2 [ring_size_t](#) [ring_init](#) ([ring_t](#) * *ring*, void * *buffer*, [ring_size_t](#) *size*)

5.77.3.3 [ring_size_t](#) [ring_read](#) ([ring_t](#) * *ring*, void * *buffer*, [ring_size_t](#) *num*)

5.77.3.4 [ring_size_t](#) [ring_read_num](#) ([ring_t](#) * *ring*)

5.77.3.5 [ring_size_t](#) [ring_write](#) ([ring_t](#) * *ring*, const void * *buffer*, [ring_size_t](#) *num*)

5.77.3.6 [ring_size_t](#) [ring_write_num](#) ([ring_t](#) * *ring*)

5.78 s_eeprom.c File Reference

```
#include "s_eeprom.h"
#include "spi.h"
#include "delay.h"
#include <stdio.h>
```

Defines

- #define [SPI_EEPROM_TRANSPARENT](#)
- #define [SPI_EEPROM_PUTC\(val\)](#) spi_putc(val)
- #define [SPI_EEPROM_GETC\(\)](#) spi_getc()
- #define [SPI_EEPROM_OP\(op\)](#) SPI_EEPROM_PUTC (op)
- #define [SPI_EEPROM_ENABLE\(DEV\)](#) port_pins_set_low ((DEV) → cs_port, (DEV) → cs_bitmask)
- #define [SPI_EEPROM_DISABLE\(DEV\)](#) port_pins_set_high ((DEV) → cs_port, (DEV) → cs_bitmask)

Enumerations

- enum {
[SPI_EEPROM_OP_WRSR](#) = 1, [SPI_EEPROM_OP_WRITE](#) = 2, [SPI_EEPROM_OP_READ](#) = 3,
[SPI_EEPROM_OP_WRDI](#) = 4,
[SPI_EEPROM_OP_RDSR](#) = 5, [SPI_EEPROM_OP_WREN](#) = 6 }
- enum {
[SPI_EEPROM_WPEN](#) = BIT (7), [SPI_EEPROM_BP1](#) = BIT (3), [SPI_EEPROM_BP0](#) = BIT (2),
[SPI_EEPROM_WEL](#) = BIT (1),
[SPI_EEPROM_WIP](#) = BIT (0) }

Functions

- [spi_eeprom_size_t spi_eeprom_read](#) ([spi_eeprom_t](#) eeprom, [spi_eeprom_addr_t](#) addr, void *buffer, [spi_eeprom_size_t](#) size)
- [spi_eeprom_size_t spi_eeprom_write](#) ([spi_eeprom_t](#) eeprom, [spi_eeprom_addr_t](#) addr, const void *buffer, [spi_eeprom_size_t](#) size)
- [uint8_t spi_eeprom_write_setup](#) ([spi_eeprom_t](#) eeprom, [spi_eeprom_addr_t](#) addr)
- [uint8_t spi_eeprom_disable](#) ([spi_eeprom_t](#) eeprom)
- static void [spi_eeprom_status_write](#) ([spi_eeprom_t](#) eeprom, [uint8_t](#) data)
- [spi_eeprom_t spi_eeprom_init](#) ([spi_eeprom_obj_t](#) *eeprom, const [spi_eeprom_cfg_t](#) *cfg)

5.78.1 Detailed Description

Author:

Michael Hayes

Date:

06/08/03

Description: Routines to read/write SPI EEPROM.

5.78.2 Define Documentation

5.78.2.1 `#define SPI_EEPROM_DISABLE(DEV) port_pins_set_high ((DEV) → cs_port, (DEV) → cs_bitmask)`

5.78.2.2 `#define SPI_EEPROM_ENABLE(DEV) port_pins_set_low ((DEV) → cs_port, (DEV) → cs_bitmask)`

5.78.2.3 `#define SPI_EEPROM_GETC() spi_getc()`

5.78.2.4 `#define SPI_EEPROM_OP(op) SPI_EEPROM_PUTC (op)`

5.78.2.5 `#define SPI_EEPROM_PUTC(val) spi_putc(val)`

5.78.2.6 `#define SPI_EEPROM_TRANSPARENT`

5.78.3 Enumeration Type Documentation

5.78.3.1 anonymous enum

Enumerator:

SPI_EEPROM_OP_WRSR

SPI_EEPROM_OP_WRITE

SPI_EEPROM_OP_READ

SPI_EEPROM_OP_WRDI

SPI_EEPROM_OP_RDSR

SPI_EEPROM_OP_WREN

5.78.3.2 anonymous enum

Enumerator:

SPI_EEPROM_WPEN

SPI_EEPROM_BP1

SPI_EEPROM_BP0

SPI_EEPROM_WEL

SPI_EEPROM_WIP

5.78.4 Function Documentation

5.78.4.1 `uint8_t spi_eeprom_disable (spi_eeprom_t eeprom)`

5.78.4.2 `spi_eeprom_t spi_eeprom_init (spi_eeprom_obj_t * eeprom, const spi_eeprom_cfg_t * cfg)`

5.78.4.3 `spi_eeprom_size_t spi_eeprom_read (spi_eeprom_t eeprom, spi_eeprom_addr_t addr, void * buffer, spi_eeprom_size_t size)`

5.78.4.4 `static void spi_eeprom_status_write (spi_eeprom_t eeprom, uint8_t data)` `[static]`

5.78.4.5 `spi_eeprom_size_t spi_eeprom_write (spi_eeprom_t eeprom, spi_eeprom_addr_t addr, const void * buffer, spi_eeprom_size_t size)`

5.78.4.6 `uint8_t spi_eeprom_write_setup (spi_eeprom_t eeprom, spi_eeprom_addr_t addr)`

5.79 s_eeprom.h File Reference

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [spi_eeprom_cfg_t](#)
- struct [spi_eeprom_private_t](#)

Typedefs

- typedef [spi_eeprom_private_t](#) [spi_eeprom_obj_t](#)
- typedef [spi_eeprom_obj_t](#) * [spi_eeprom_t](#)
- typedef uint16_t [spi_eeprom_addr_t](#)
- typedef uint16_t [spi_eeprom_size_t](#)

Functions

- [spi_eeprom_t](#) [spi_eeprom_init](#) ([spi_eeprom_obj_t](#) *dev, const [spi_eeprom_cfg_t](#) *cfg)
- [spi_eeprom_size_t](#) [spi_eeprom_read](#) ([spi_eeprom_t](#) dev, [spi_eeprom_addr_t](#) addr, void *buffer, [spi_eeprom_size_t](#) size)
- [spi_eeprom_size_t](#) [spi_eeprom_write](#) ([spi_eeprom_t](#) dev, [spi_eeprom_addr_t](#) addr, const void *buffer, [spi_eeprom_size_t](#) size)
- uint8_t [spi_eeprom_write_setup](#) ([spi_eeprom_t](#) dev, [spi_eeprom_addr_t](#) addr)
- uint8_t [spi_eeprom_disable](#) ([spi_eeprom_t](#) dev)

5.79.1 Typedef Documentation

5.79.1.1 typedef uint16_t [spi_eeprom_addr_t](#)

5.79.1.2 typedef [spi_eeprom_private_t](#) [spi_eeprom_obj_t](#)

5.79.1.3 typedef uint16_t [spi_eeprom_size_t](#)

5.79.1.4 typedef [spi_eeprom_obj_t](#)* [spi_eeprom_t](#)

5.79.2 Function Documentation

5.79.2.1 uint8_t [spi_eeprom_disable](#) ([spi_eeprom_t](#) *dev*)

5.79.2.2 [spi_eeprom_t](#) [spi_eeprom_init](#) ([spi_eeprom_obj_t](#) * *dev*, const [spi_eeprom_cfg_t](#) * *cfg*)

5.79.2.3 [spi_eeprom_size_t](#) [spi_eeprom_read](#) ([spi_eeprom_t](#) *dev*, [spi_eeprom_addr_t](#) *addr*, void * *buffer*, [spi_eeprom_size_t](#) *size*)

5.79.2.4 [spi_eeprom_size_t](#) [spi_eeprom_write](#) ([spi_eeprom_t](#) *dev*, [spi_eeprom_addr_t](#) *addr*, const void * *buffer*, [spi_eeprom_size_t](#) *size*)

5.79.2.5 uint8_t [spi_eeprom_write_setup](#) ([spi_eeprom_t](#) *dev*, [spi_eeprom_addr_t](#) *addr*)

5.80 scroller.c File Reference

```
#include "scroller.h"  
#include <string.h>
```

Functions

- [scroller_t](#) [scroller_init](#) ([scroller_t](#) *scroller*, int *rows*, int *cols*, [scroller_dir_t](#) *dir*)
- int8_t [scroller_update](#) ([scroller_t](#) *scroller*, uint8_t **image*, uint8_t **screen*)
- void [scroller_start](#) ([scroller_t](#) *scroller*, uint8_t **image*, uint8_t **screen*)
- uint8_t [scroller_speed_scale_get](#) ([scroller_t](#) *scroller*)

5.80.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.80.2 Function Documentation

5.80.2.1 [scroller_t](#) [scroller_init](#) ([scroller_t](#) *scroller*, int *rows*, int *cols*, [scroller_dir_t](#) *dir*)

5.80.2.2 uint8_t [scroller_speed_scale_get](#) ([scroller_t](#) *scroller*)

5.80.2.3 void [scroller_start](#) ([scroller_t](#) *scroller*, uint8_t * *image*, uint8_t * *screen*)

5.80.2.4 int8_t [scroller_update](#) ([scroller_t](#) *scroller*, uint8_t * *image*, uint8_t * *screen*)

5.81 scroller.h File Reference

Image scroller.

```
#include "config.h"
```

Data Structures

- struct [scroller_obj_t](#)

Typedefs

- typedef [scroller_obj_t](#) * [scroller_t](#)

Enumerations

- enum [scroller_dir_t](#) {
 [SCROLLER_OFF](#), [SCROLLER_LEFT](#), [SCROLLER_RIGHT](#), [SCROLLER_DOWN](#),
 [SCROLLER_UP](#) }

Functions

- [scroller_t](#) [scroller_init](#) ([scroller_t](#) scroller, int rows, int cols, [scroller_dir_t](#) dir)
- int8_t [scroller_update](#) ([scroller_t](#) scroller, uint8_t *image, uint8_t *screen)
- void [scroller_start](#) ([scroller_t](#) scroller, uint8_t *image, uint8_t *screen)
- uint8_t [scroller_speed_scale_get](#) ([scroller_t](#) scroller)
- static void [scroller_stop](#) ([scroller_t](#) scroller)
- static void [scroller_dir_set](#) ([scroller_t](#) scroller, [scroller_dir_t](#) dir)
- static [scroller_dir_t](#) [scroller_dir_get](#) ([scroller_t](#) scroller)

5.81.1 Detailed Description

Image scroller.

Author:

M. P. Hayes, UCECE

Date:

7 April 2007

5.81.2 Typedef Documentation

5.81.2.1 typedef [scroller_obj_t](#)* [scroller_t](#)

5.81.3 Enumeration Type Documentation

5.81.3.1 enum [scroller_dir_t](#)

Enumerator:

[SCROLLER_OFF](#)

SCROLLER_LEFT
SCROLLER_RIGHT
SCROLLER_DOWN
SCROLLER_UP

5.81.4 Function Documentation

- 5.81.4.1 static [scroller_dir_t](#) scroller_dir_get ([scroller_t](#) *scroller*) [static]
- 5.81.4.2 static void scroller_dir_set ([scroller_t](#) *scroller*, [scroller_dir_t](#) *dir*) [static]
- 5.81.4.3 [scroller_t](#) scroller_init ([scroller_t](#) *scroller*, int *rows*, int *cols*, [scroller_dir_t](#) *dir*)
- 5.81.4.4 uint8_t scroller_speed_scale_get ([scroller_t](#) *scroller*)
- 5.81.4.5 void scroller_start ([scroller_t](#) *scroller*, uint8_t * *image*, uint8_t * *screen*)
- 5.81.4.6 static void scroller_stop ([scroller_t](#) *scroller*) [static]
- 5.81.4.7 int8_t scroller_update ([scroller_t](#) *scroller*, uint8_t * *image*, uint8_t * *screen*)

5.82 seq.c File Reference

```
#include <limits.h>
#include "seq.h"
```

Functions

- `seq_t seq_init (seq_obj_t *seq, const char *(*callback)(void *data, const char *str), void *callback_data)`
- `int8_t seq_update (seq_t seq)`

5.82.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.82.2 Function Documentation

5.82.2.1 `seq_t seq_init (seq_obj_t *seq, const char *(*)(void *data, const char *str) callback, void *callback_data)`

5.82.2.2 `int8_t seq_update (seq_t seq)`

5.83 seq.h File Reference

```
#include "config.h"
```

Data Structures

- struct [seq_obj_t](#)

Typedefs

- typedef [seq_obj_t](#) * [seq_t](#)

Functions

- [seq_t](#) [seq_init](#) ([seq_obj_t](#) *dev, const char *(*callback)(void *data, const char *str), void *callback_data)
- static void [seq_set](#) ([seq_t](#) seq, const char *str)
- static const char * [seq_get](#) ([seq_t](#) seq)
- int8_t [seq_update](#) ([seq_t](#) seq)

5.83.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

1 April 2007

5.83.2 Typedef Documentation

5.83.2.1 typedef [seq_obj_t](#)* [seq_t](#)

5.83.3 Function Documentation

5.83.3.1 static const char* [seq_get](#) ([seq_t](#) seq) [static]

5.83.3.2 [seq_t](#) [seq_init](#) ([seq_obj_t](#) * dev, const char *(*)(void *data, const char *str) *callback*, void * *callback_data*)

5.83.3.3 static void [seq_set](#) ([seq_t](#) seq, const char * *str*) [static]

5.83.3.4 int8_t [seq_update](#) ([seq_t](#) seq)

5.84 sflash.c File Reference

```
#include "sflash.h"
```

Functions

- bool [sflash_update](#) ([sflash_t](#) sflash)

5.84.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

2 July 2007

5.84.2 Function Documentation

5.84.2.1 bool sflash_update ([sflash_t](#) *sflash*)

5.85 sflash.h File Reference

```
#include "config.h"
```

Data Structures

- struct [sflash_obj_t](#)

Typedefs

- typedef uint16_t [sflash_pattern_t](#)
- typedef [sflash_obj_t](#) * [sflash_t](#)

Functions

- static void [sflash_pattern_set](#) ([sflash_t](#) sflash, [sflash_pattern_t](#) pattern, [sflash_pattern_t](#) initial)
- static [sflash_pattern_t](#) [sflash_pattern_get](#) ([sflash_t](#) sflash)
- bool [sflash_update](#) ([sflash_t](#))

5.85.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

2 July 2007

5.85.2 Typedef Documentation

5.85.2.1 typedef uint16_t [sflash_pattern_t](#)

5.85.2.2 typedef [sflash_obj_t](#)* [sflash_t](#)

5.85.3 Function Documentation

5.85.3.1 static [sflash_pattern_t](#) [sflash_pattern_get](#) ([sflash_t](#) *sflash*) [static]

5.85.3.2 static void [sflash_pattern_set](#) ([sflash_t](#) *sflash*, [sflash_pattern_t](#) *pattern*, [sflash_pattern_t](#) *initial*) [static]

5.85.3.3 bool [sflash_update](#) ([sflash_t](#))

5.86 spi_adc.h File Reference

SPI ADC.

```
#include "config.h"
```

Enumerations

- enum [spi_adc_mode_t](#) { [SPI_ADC_MODE_SINGLE_ENDED](#), [SPI_ADC_MODE_DIFFERENTIAL](#), [SPI_ADC_MODE_DIFFERENTIAL_INVERTED](#) }

5.86.1 Detailed Description

SPI ADC.

Author:

M. P. Hayes, UCECE

Date:

09 August 2007

5.86.2 Enumeration Type Documentation

5.86.2.1 enum [spi_adc_mode_t](#)

Enumerator:

[SPI_ADC_MODE_SINGLE_ENDED](#)

[SPI_ADC_MODE_DIFFERENTIAL](#)

[SPI_ADC_MODE_DIFFERENTIAL_INVERTED](#)

5.87 spwm.c File Reference

```
#include "config.h"
#include "spwm.h"
```

Functions

- void [spwm_period_set](#) ([spwm_t](#) spwm, uint16_t period)
- void [spwm_duty_set](#) ([spwm_t](#) spwm, uint16_t duty)
- bool [spwm_update](#) ([spwm_t](#) spwm)
- [spwm_t](#) [spwm_init](#) ([spwm_obj_t](#) *dev)

5.87.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

13 March 2005

5.87.2 Function Documentation

5.87.2.1 void [spwm_duty_set](#) ([spwm_t](#) spwm, uint16_t duty)

5.87.2.2 [spwm_t](#) [spwm_init](#) ([spwm_obj_t](#) * dev)

5.87.2.3 void [spwm_period_set](#) ([spwm_t](#) spwm, uint16_t period)

5.87.2.4 bool [spwm_update](#) ([spwm_t](#) spwm)

5.88 spwm.h File Reference

```
#include "config.h"
#include "port.h"
```

Data Structures

- struct [spwm_obj_t](#)

Typedefs

- typedef spwm_struct * [spwm_t](#)

Functions

- void [spwm_period_set](#) ([spwm_t](#) spwm, uint16_t period)
- void [spwm_duty_set](#) ([spwm_t](#) spwm, uint16_t duty)
- bool [spwm_update](#) ([spwm_t](#))
- [spwm_t](#) [spwm_init](#) ([spwm_obj_t](#) *info)

5.88.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

13 March 2005

5.88.2 Typedef Documentation

5.88.2.1 typedef struct spwm_struct* [spwm_t](#)

5.88.3 Function Documentation

5.88.3.1 void [spwm_duty_set](#) ([spwm_t](#) spwm, uint16_t duty)

5.88.3.2 [spwm_t](#) [spwm_init](#) ([spwm_obj_t](#) * info)

5.88.3.3 void [spwm_period_set](#) ([spwm_t](#) spwm, uint16_t period)

5.88.3.4 bool [spwm_update](#) ([spwm_t](#))

5.89 squeaker.c File Reference

Play simple tunes with PWM.

```
#include "squeaker.h"
```

Defines

- #define [SQUEAKER_TRANSPARENT](#) 1
- #define [SQUEAKER_HOLDOFF_TIME](#) 50e-3

Enumerations

- enum { [SQUEAKER_PRESCALER](#) = 256 }
- enum { [SQUEAKER_SCALE_SIZE](#) = 12 }

Functions

- static void [squeaker_ticker_set](#) ([squeaker_t](#) squeaker)
- static void [squeaker_note_fraction_set](#) ([squeaker_t](#) squeaker, [uint8_t](#) note_fraction)
- void [squeaker_note_set](#) ([squeaker_t](#) squeaker, [squeaker_period_t](#) period, [squeaker_period_t](#) duty)
- static void [squeaker_rest_play](#) ([squeaker_t](#) squeaker)
- static void [squeaker_note_play](#) ([squeaker_t](#) squeaker, [squeaker_note_t](#) note)
- static [squeaker_note_t](#) [squeaker_char_to_note](#) ([uint8_t](#) ch)
- static const char * [squeaker_scan](#) ([squeaker_t](#) squeaker, const char *str)
- void [squeaker_play](#) ([squeaker_t](#) squeaker, const char *str)
- void [squeaker_speed_set](#) ([squeaker_t](#) squeaker, [squeaker_speed_t](#) speed)
- void [squeaker_volume_set](#) ([squeaker_t](#) squeaker, [squeaker_volume_t](#) volume)
- [int8_t](#) [squeaker_update](#) ([squeaker_t](#) squeaker)
- [squeaker_t](#) [squeaker_init](#) ([squeaker_obj_t](#) *squeaker, [uint16_t](#) poll_rate, [squeaker_scale_t](#) *scale_table)

5.89.1 Detailed Description

Play simple tunes with PWM.

Author:

M. P. Hayes, UCECE

Date:

14 April 2007

5.89.2 Define Documentation

5.89.2.1 `#define SQUEAKER_HOLDOFF_TIME 50e-3`

5.89.2.2 `#define SQUEAKER_TRANSPARENT 1`

5.89.3 Enumeration Type Documentation

5.89.3.1 anonymous enum

Enumerator:

SQUEAKER_PRESCALER

5.89.3.2 anonymous enum

Enumerator:

SQUEAKER_SCALE_SIZE

5.89.4 Function Documentation

5.89.4.1 static [squeaker_note_t](#) `squeaker_char_to_note (uint8_t ch)` [static]

5.89.4.2 [squeaker_t](#) `squeaker_init (squeaker_obj_t * squeaker, uint16_t poll_rate, squeaker_scale_t * scale_table)`

5.89.4.3 static void `squeaker_note_fraction_set (squeaker_t squeaker, uint8_t note_fraction)` [static]

5.89.4.4 static void `squeaker_note_play (squeaker_t squeaker, squeaker_note_t note)` [static]

5.89.4.5 void `squeaker_note_set (squeaker_t squeaker, squeaker_period_t period, squeaker_period_t duty)`

5.89.4.6 void `squeaker_play (squeaker_t squeaker, const char * str)`

5.89.4.7 static void `squeaker_rest_play (squeaker_t squeaker)` [static]

5.89.4.8 static const char* `squeaker_scan (squeaker_t squeaker, const char * str)` [static]

5.89.4.9 void `squeaker_speed_set (squeaker_t squeaker, squeaker_speed_t speed)`

5.89.4.10 static void `squeaker_ticker_set (squeaker_t squeaker)` [static]

5.89.4.11 int8_t `squeaker_update (squeaker_t squeaker)`

5.89.4.12 void `squeaker_volume_set (squeaker_t squeaker, squeaker_volume_t volume)`

5.90 squeaker.h File Reference

Play simple tunes with PWM.

```
#include "config.h"
#include "font.h"
#include "ticker.h"
```

Data Structures

- struct [squeaker_private_t](#)

Defines

- #define [SQUEAKER_DIVISOR](#)(POLL_RATE, FREQ) (POLL_RATE / FREQ + 0.5)
- #define [SQUEAKER_SCALE_TABLE](#)(POLL_RATE)

Typedefs

- typedef uint8_t [squeaker_speed_t](#)
- typedef uint8_t [squeaker_scale_t](#)
- typedef uint8_t [squeaker_note_t](#)
- typedef uint8_t [squeaker_duration_t](#)
- typedef uint8_t [squeaker_period_t](#)
- typedef uint8_t [squeaker_volume_t](#)
- typedef [squeaker_private_t](#) [squeaker_obj_t](#)
- typedef [squeaker_obj_t](#) * [squeaker_t](#)

Enumerations

- enum { [SQUEAKER_OCTAVE_DEFAULT](#) = 4 }
- enum { [SQUEAKER_SPEED_DEFAULT](#) = 200 }
- enum { [SQUEAKER_NOTE_MIN](#) = 40 }

Functions

- [squeaker_t](#) [squeaker_init](#) ([squeaker_obj_t](#) *dev, uint16_t poll_rate, [squeaker_scale_t](#) *scale_table)
- void [squeaker_play](#) ([squeaker_t](#) squeaker, const char *str)
- int8_t [squeaker_update](#) ([squeaker_t](#) squeaker)
- void [squeaker_speed_set](#) ([squeaker_t](#) squeaker, [squeaker_speed_t](#) speed)
- void [squeaker_volume_set](#) ([squeaker_t](#) squeaker, [squeaker_volume_t](#) volume)

5.90.1 Detailed Description

Play simple tunes with PWM.

Author:

M. P. Hayes, UCECE

Date:

14 April 2007

5.90.2 Define Documentation

5.90.2.1 `#define SQUEAKER_DIVISOR(POLL_RATE, FREQ) (POLL_RATE / FREQ + 0.5)`

5.90.2.2 `#define SQUEAKER_SCALE_TABLE(POLL_RATE)`

Value:

```
{SQUEAKER_DIVISOR (POLL_RATE, 82.41), \
  SQUEAKER_DIVISOR (POLL_RATE, 87.31), \
  SQUEAKER_DIVISOR (POLL_RATE, 92.50), \
  SQUEAKER_DIVISOR (POLL_RATE, 98.00), \
  SQUEAKER_DIVISOR (POLL_RATE, 103.83), \
  SQUEAKER_DIVISOR (POLL_RATE, 110.0), \
  SQUEAKER_DIVISOR (POLL_RATE, 116.54), \
  SQUEAKER_DIVISOR (POLL_RATE, 123.47), \
  SQUEAKER_DIVISOR (POLL_RATE, 130.81), \
  SQUEAKER_DIVISOR (POLL_RATE, 138.59), \
  SQUEAKER_DIVISOR (POLL_RATE, 146.83), \
  SQUEAKER_DIVISOR (POLL_RATE, 155.56)}
```

5.90.3 Typedef Documentation

5.90.3.1 `typedef uint8_t squeaker_duration_t`

5.90.3.2 `typedef uint8_t squeaker_note_t`

5.90.3.3 `typedef squeaker_private_t squeaker_obj_t`

5.90.3.4 `typedef uint8_t squeaker_period_t`

5.90.3.5 `typedef uint8_t squeaker_scale_t`

5.90.3.6 `typedef uint8_t squeaker_speed_t`

5.90.3.7 `typedef squeaker_obj_t* squeaker_t`

5.90.3.8 `typedef uint8_t squeaker_volume_t`

5.90.4 Enumeration Type Documentation

5.90.4.1 anonymous enum

Enumerator:

SQUEAKER_OCTAVE_DEFAULT

5.90.4.2 anonymous enum

Enumerator:

SQUEAKER_SPEED_DEFAULT

5.90.4.3 anonymous enum

Enumerator:

SQUEAKER_NOTE_MIN

5.90.5 Function Documentation

5.90.5.1 `squeaker_t` `squeaker_init` (`squeaker_obj_t` * *dev*, `uint16_t` *poll_rate*, `squeaker_scale_t` * *scale_table*)

5.90.5.2 `void` `squeaker_play` (`squeaker_t` *squeaker*, `const char` * *str*)

5.90.5.3 `void` `squeaker_speed_set` (`squeaker_t` *squeaker*, `squeaker_speed_t` *speed*)

5.90.5.4 `int8_t` `squeaker_update` (`squeaker_t` *squeaker*)

5.90.5.5 `void` `squeaker_volume_set` (`squeaker_t` *squeaker*, `squeaker_volume_t` *volume*)

5.91 squeaker2.c File Reference

```
#include "squeaker.h"
```

Defines

- #define [SQUEAKER_TRANSPARENT](#) 1
- #define [SQUEAKER_HOLDOFF_TIME](#) 50e-3

Enumerations

- enum { [SQUEAKER_PRESCALER](#) = 256 }
- enum { [SQUEAKER_SCALE_SIZE](#) = 12 }

Functions

- static void [squeaker_ticker_set](#) ([squeaker_t](#) *squeaker)
- static void [squeaker_note_fraction_set](#) ([squeaker_t](#) *squeaker, [uint8_t](#) note_fraction)
- void [squeaker_note_set](#) ([squeaker_t](#) *squeaker, [squeaker_period_t](#) period, [squeaker_period_t](#) duty)
- static void [squeaker_rest_play](#) ([squeaker_t](#) *squeaker)
- static void [squeaker_note_play](#) ([squeaker_t](#) *squeaker, [squeaker_note_t](#) note)
- static [squeaker_note_t](#) [squeaker_char_to_note](#) ([uint8_t](#) ch)
- static const char * [squeaker_scan](#) ([squeaker_t](#) *squeaker, const char *str)
- void [squeaker_play](#) ([squeaker_t](#) *squeaker, const char *str)
- void [squeaker_speed_set](#) ([squeaker_t](#) *squeaker, [squeaker_speed_t](#) speed)
- void [squeaker_volume_set](#) ([squeaker_t](#) *squeaker, [squeaker_volume_t](#) volume)
- [int8_t](#) [squeaker_update](#) ([squeaker_t](#) *squeaker)
- [squeaker_t](#) [squeaker_init](#) ([squeaker_obj_t](#) *squeaker, [uint16_t](#) poll_rate, [squeaker_scale_t](#) *scale_table)

5.91.1 Define Documentation

5.91.1.1 #define [SQUEAKER_HOLDOFF_TIME](#) 50e-3

5.91.1.2 #define [SQUEAKER_TRANSPARENT](#) 1

5.91.2 Enumeration Type Documentation

5.91.2.1 anonymous enum

Enumerator:

[SQUEAKER_PRESCALER](#)

5.91.2.2 anonymous enum

Enumerator:

[SQUEAKER_SCALE_SIZE](#)

5.91.3 Function Documentation

- 5.91.3.1 static [squeaker_note_t](#) squeaker_char_to_note (uint8_t *ch*) [static]
- 5.91.3.2 [squeaker_t](#) squeaker_init ([squeaker_obj_t](#) * *squeaker*, uint16_t *poll_rate*, [squeaker_scale_t](#) * *scale_table*)
- 5.91.3.3 static void squeaker_note_fraction_set ([squeaker_t](#) * *squeaker*, uint8_t *note_fraction*) [static]
- 5.91.3.4 static void squeaker_note_play ([squeaker_t](#) * *squeaker*, [squeaker_note_t](#) *note*) [static]
- 5.91.3.5 void squeaker_note_set ([squeaker_t](#) * *squeaker*, [squeaker_period_t](#) *period*, [squeaker_period_t](#) *duty*)
- 5.91.3.6 void squeaker_play ([squeaker_t](#) * *squeaker*, const char * *str*)
- 5.91.3.7 static void squeaker_rest_play ([squeaker_t](#) * *squeaker*) [static]
- 5.91.3.8 static const char* squeaker_scan ([squeaker_t](#) * *squeaker*, const char * *str*) [static]
- 5.91.3.9 void squeaker_speed_set ([squeaker_t](#) * *squeaker*, [squeaker_speed_t](#) *speed*)
- 5.91.3.10 static void squeaker_ticker_set ([squeaker_t](#) * *squeaker*) [static]
- 5.91.3.11 int8_t squeaker_update ([squeaker_t](#) * *squeaker*)
- 5.91.3.12 void squeaker_volume_set ([squeaker_t](#) * *squeaker*, [squeaker_volume_t](#) *volume*)

5.92 stext.c File Reference

Sequenced text.

```
#include "stext.h"
#include <limits.h>
```

Functions

- static const char * [stext_display](#) (void *data, const char *str)
- [stext_t](#) [stext_init](#) ([stext_obj_t](#) *stext, [font_t](#) *font, void(*callback)(void *data, uint8_t pixel, bool val), void *callback_data)

5.92.1 Detailed Description

Sequenced text.

Author:

M. P. Hayes, UCECE

Date:

2 April 2007

5.92.2 Function Documentation

5.92.2.1 static const char* [stext_display](#) (void * *data*, const char * *str*) [static]

5.92.2.2 [stext_t](#) [stext_init](#) ([stext_obj_t](#) * *stext*, [font_t](#) * *font*, void(*) (void *data, uint8_t pixel, bool val) *callback*, void * *callback_data*)

5.93 stext.h File Reference

Sequenced text.

```
#include "config.h"
#include "font.h"
#include "seq.h"
```

Data Structures

- struct [stext_obj_t](#)

Typedefs

- typedef [stext_obj_t](#) * [stext_t](#)

Enumerations

- enum [stext_mode_t](#) { [STEXT_MODE_NORMAL](#), [STEXT_MODE_CYCLE](#), [STEXT_MODE_NUM](#) }

Functions

- [stext_t stext_init](#) ([stext_obj_t](#) *dev, [font_t](#) *font, void(*callback)(void *data, uint8_t pixel, bool val), void *callback_data)
- static void [stext_set](#) ([stext_t](#) stext, const char *str)
- static const char * [stext_get](#) ([stext_t](#) stext)
- static int8_t [stext_update](#) ([stext_t](#) stext)

5.93.1 Detailed Description

Sequenced text.

Author:

M. P. Hayes, UCECE

Date:

2 April 2007

5.93.2 Typedef Documentation

5.93.2.1 typedef [stext_obj_t](#)* [stext_t](#)

5.93.3 Enumeration Type Documentation

5.93.3.1 enum [stext_mode_t](#)

Enumerator:

[STEXT_MODE_NORMAL](#)

STEXT_MODE_CYCLE

STEXT_MODE_NUM

5.93.4 Function Documentation

5.93.4.1 static const char* stext_get ([stext_t](#) *stext*) [static]

5.93.4.2 [stext_t](#) stext_init ([stext_obj_t](#) * *dev*, [font_t](#) * *font*, void(*) (void **data*, uint8_t *pixel*, bool *val*) *callback*, void * *callback_data*)

5.93.4.3 static void stext_set ([stext_t](#) *stext*, const char * *str*) [static]

5.93.4.4 static int8_t stext_update ([stext_t](#) *stext*) [static]

5.94 ticker.c File Reference

5.94.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

2 April 2007

5.95 ticker.h File Reference

```
#include "config.h"
```

Data Structures

- struct [ticker_t](#)
- struct [ticker16_t](#)
- struct [ticker8_t](#)

Defines

- #define [TICKER_INIT](#)(DEV, PERIOD)
- #define [TICKER_UPDATE](#)(DEV) $(\neg(\text{DEV}) \rightarrow \text{clock} ? 0 : ((\text{DEV}) \rightarrow \text{clock} = (\text{DEV}) \rightarrow \text{period}))$
- #define [TICKER_START](#)(DEV) $(\text{DEV}) \rightarrow \text{clock} = (\text{DEV}) \rightarrow \text{period};$

5.95.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

2 April 2007

5.95.2 Define Documentation

5.95.2.1 #define TICKER_INIT(DEV, PERIOD)

Value:

```
(DEV)->period = (PERIOD);          \  
    (DEV)->clock = (DEV)->period;
```

5.95.2.2 #define TICKER_START(DEV) $(\text{DEV}) \rightarrow \text{clock} = (\text{DEV}) \rightarrow \text{period};$

5.95.2.3 #define TICKER_UPDATE(DEV) $(\neg(\text{DEV}) \rightarrow \text{clock} ? 0 : ((\text{DEV}) \rightarrow \text{clock} = (\text{DEV}) \rightarrow \text{period}))$

5.96 time.c File Reference

Time routines.

```
#include "time.h"
#include "delay.h"
```

Functions

- void [time_init](#) (void)
- void [time_delay_us](#) (uint16_t us)
- [time_t](#) [time_rf_timestamp_get](#) (void)
- [time_t](#) [time_current_time_get](#) (void)
- uint32_t [time_time2int](#) ([time_t](#) time)
- void [time_irq_enable](#) (void)
- void [time_irq_disable](#) (void)

Variables

- static volatile uint16_t [ms_ticks](#) = 0

5.96.1 Detailed Description

Time routines.

Author:

Ian Downes, UCECE

Date:

12 January 2005

5.96.2 Function Documentation

5.96.2.1 [time_t](#) [time_current_time_get](#) (void)

5.96.2.2 void [time_delay_us](#) (uint16_t *us*)

5.96.2.3 void [time_init](#) (void)

5.96.2.4 void [time_irq_disable](#) (void)

5.96.2.5 void [time_irq_enable](#) (void)

5.96.2.6 [time_t](#) [time_rf_timestamp_get](#) (void)

5.96.2.7 uint32_t [time_time2int](#) ([time_t](#) *time*)

5.96.3 Variable Documentation

5.96.3.1 volatile uint16_t [ms_ticks](#) = 0 [static]

5.97 time.h File Reference

Time routines.

```
#include "config.h"
#include "irq.h"
```

Data Structures

- struct [time](#)

Defines

- #define [TOPCNT](#) (64000 - 1)
- #define [DELAY_MIN](#) 10
- #define [DELAY_MAX](#) 3999

Typedefs

- typedef [time](#) [time_t](#)

Enumerations

- enum [delay_ret_t](#) {
 [ERRD](#), [USTICKS](#), [TIMEOUT](#), [COMMS_INT](#),
 [OTHER_INT](#) }

Functions

- void [time_init](#) (void)
- [time_t](#) [time_rf_timestamp_get](#) (void)
- [time_t](#) [time_current_time_get](#) (void)
- void [time_delay_us](#) (uint16_t us)
- uint32_t [time_time2int](#) ([time_t](#) time)
- void [time_irq_enable](#) (void)
- void [time_irq_disable](#) (void)

5.97.1 Detailed Description

Time routines.

Author:

Ian Downes, UCECE

Date:

12 January 2005

5.97.2 Define Documentation

5.97.2.1 `#define DELAY_MAX 3999`

5.97.2.2 `#define DELAY_MIN 10`

5.97.2.3 `#define TOPCNT (64000 - 1)`

5.97.3 Typedef Documentation

5.97.3.1 typedef struct [time](#) [time_t](#)

5.97.4 Enumeration Type Documentation

5.97.4.1 enum [delay_ret_t](#)

Enumerator:

ERRD

USTICKS

TIMEOUT

COMMS_INT

OTHER_INT

5.97.5 Function Documentation

5.97.5.1 [time_t](#) time_current_time_get (void)

5.97.5.2 void time_delay_us (uint16_t *us*)

5.97.5.3 void time_init (void)

5.97.5.4 void time_irq_disable (void)

5.97.5.5 void time_irq_enable (void)

5.97.5.6 [time_t](#) time_rf_timestamp_get (void)

5.97.5.7 uint32_t time_time2int ([time_t](#) *time*)

5.98 tweeter.c File Reference

Generate PWM for a piezo tweeter.

```
#include "tweeter.h"
```

Defines

- `#define TWEETER_TRANSPARENT 1`
- `#define TWEETER_HOLDOFF_TIME 50e-3`

Enumerations

- `enum { TWEETER_SCALE_SIZE = 12 }`

Functions

- `void tweeter_note_set (tweeter_t tweeter, tweeter_period_t period, tweeter_period_t duty)`
- `void tweeter_note_play (tweeter_t tweeter, tweeter_note_t note, uint8_t velocity)`
- `int8_t tweeter_update (tweeter_t tweeter)`
- `tweeter_t tweeter_init (tweeter_obj_t *tweeter, uint16_t poll_rate, tweeter_scale_t *scale_table)`

5.98.1 Detailed Description

Generate PWM for a piezo tweeter.

Author:

M. P. Hayes, UCECE

Date:

20 April 2007

5.98.2 Define Documentation

5.98.2.1 `#define TWEETER_HOLDOFF_TIME 50e-3`

5.98.2.2 `#define TWEETER_TRANSPARENT 1`

5.98.3 Enumeration Type Documentation

5.98.3.1 anonymous enum

Enumerator:

TWEETER_SCALE_SIZE

5.98.4 Function Documentation

5.98.4.1 `tweeter_t` `tweeter_init` (`tweeter_obj_t` * *tweeter*, `uint16_t` *poll_rate*, `tweeter_scale_t` * *scale_table*)

5.98.4.2 `void` `tweeter_note_play` (`tweeter_t` *tweeter*, `tweeter_note_t` *note*, `uint8_t` *velocity*)

5.98.4.3 `void` `tweeter_note_set` (`tweeter_t` *tweeter*, `tweeter_period_t` *period*, `tweeter_period_t` *duty*)

5.98.4.4 `int8_t` `tweeter_update` (`tweeter_t` *tweeter*)

5.99 tweeter.h File Reference

Generate PWM for a piezo tweeter.

```
#include "config.h"
#include "font.h"
#include "ticker.h"
```

Data Structures

- struct [tweeter_private_t](#)

Defines

- #define [TWEETER_DIVISOR](#)(POLL_RATE, FREQ) (POLL_RATE / FREQ + 0.5)
- #define [TWEETER_SCALE_TABLE](#)(POLL_RATE)

Typedefs

- typedef uint8_t [tweeter_note_t](#)
- typedef uint8_t [tweeter_duration_t](#)
- typedef uint8_t [tweeter_period_t](#)
- typedef uint8_t [tweeter_velocity_t](#)
- typedef uint8_t [tweeter_scale_t](#)
- typedef [tweeter_private_t](#) [tweeter_obj_t](#)
- typedef [tweeter_obj_t](#) * [tweeter_t](#)

Enumerations

- enum { [TWEETER_NOTE_MIN](#) = 40 }

Functions

- int8_t [tweeter_update](#) ([tweeter_t](#) tweeter)
- void [tweeter_note_play](#) ([tweeter_t](#) tweeter, [tweeter_note_t](#) note, uint8_t velocity)
- [tweeter_t](#) [tweeter_init](#) ([tweeter_obj_t](#) *dev, uint16_t poll_rate, [tweeter_scale_t](#) *scale_table)

5.99.1 Detailed Description

Generate PWM for a piezo tweeter.

Author:

M. P. Hayes, UCECE

Date:

20 April 2007

5.99.2 Define Documentation

5.99.2.1 **#define** TWEETER_DIVISOR(POLL_RATE, FREQ) (POLL_RATE / FREQ + 0.5)

5.99.2.2 **#define** TWEETER_SCALE_TABLE(POLL_RATE)

Value:

```
{TWEETER_DIVISOR (POLL_RATE, 82.41), \
  TWEETER_DIVISOR (POLL_RATE, 87.31), \
  TWEETER_DIVISOR (POLL_RATE, 92.50), \
  TWEETER_DIVISOR (POLL_RATE, 98.00), \
  TWEETER_DIVISOR (POLL_RATE, 103.83), \
  TWEETER_DIVISOR (POLL_RATE, 110.0), \
  TWEETER_DIVISOR (POLL_RATE, 116.54), \
  TWEETER_DIVISOR (POLL_RATE, 123.47), \
  TWEETER_DIVISOR (POLL_RATE, 130.81), \
  TWEETER_DIVISOR (POLL_RATE, 138.59), \
  TWEETER_DIVISOR (POLL_RATE, 146.83), \
  TWEETER_DIVISOR (POLL_RATE, 155.56) }
```

5.99.3 Typedef Documentation

5.99.3.1 **typedef** uint8_t [tweeter_duration_t](#)

5.99.3.2 **typedef** uint8_t [tweeter_note_t](#)

5.99.3.3 **typedef** [tweeter_private_t](#) [tweeter_obj_t](#)

5.99.3.4 **typedef** uint8_t [tweeter_period_t](#)

5.99.3.5 **typedef** uint8_t [tweeter_scale_t](#)

5.99.3.6 **typedef** [tweeter_obj_t](#)* [tweeter_t](#)

5.99.3.7 **typedef** uint8_t [tweeter_velocity_t](#)

5.99.4 Enumeration Type Documentation

5.99.4.1 anonymous enum

Enumerator:

TWEETER_NOTE_MIN

5.99.5 Function Documentation

5.99.5.1 [tweeter_t](#) [tweeter_init](#) ([tweeter_obj_t](#) * *dev*, uint16_t *poll_rate*, [tweeter_scale_t](#) * *scale_table*)

5.99.5.2 void [tweeter_note_play](#) ([tweeter_t](#) *tweeter*, [tweeter_note_t](#) *note*, uint8_t *velocity*)

5.99.5.3 int8_t [tweeter_update](#) ([tweeter_t](#) *tweeter*)

5.100 u1wire.c File Reference

Low level routines to drive Dallas universal 1 wire bus. This only supports a single instance of a 1 wire bus.

```
#include <stdio.h>
#include "config.h"
#include "delay.h"
#include "u1wire.h"
#include "irq.h"
#include "port.h"
```

Defines

- #define [U1WIRE_DEBUG](#) 1
- #define [U1WIRE_RELEASE](#)() port_pin_config_pullup (U1WIRE_PORT, U1WIRE_BIT)
- #define [U1WIRE_DRIVE](#)()
- #define [U1WIRE_TEST](#)() port_pin_read (U1WIRE_PORT, U1WIRE_BIT)

Enumerations

- enum { [U1WIRE_READ_ROM](#) = 0x33, [U1WIRE_SKIP_ROM](#) = 0xcc, [U1WIRE_MATCH_ROM](#) = 0x55, [U1WIRE_RECALL](#) = 0xb8 }
- enum { [U1WIRE_DELAY_OFFSET](#) = 6, [U1WIRE_ADDR_BYTES](#) = 6 }

Functions

- int8_t [u1wire_reset](#) (void)
- void [u1wire_bit_write](#) (uint8_t value)
- void [u1wire_byte_write](#) (uint8_t value)
- uint8_t [u1wire_bit_read](#) (void)
- uint8_t [u1wire_byte_read](#) (void)
- int8_t [u1wire_rom_code_read](#) ([u1wire_t](#) dev)
- int8_t [u1wire_command](#) ([u1wire_t](#) dev, uint8_t command)
- int8_t [u1wire_broadcast](#) (uint8_t command)
- int8_t [u1wire_read](#) (void *data, uint8_t bytes)
- int8_t [u1wire_write](#) (void *data, uint8_t bytes)
- bool [u1wire_ready_p](#) (void)
- int8_t [u1wire_init](#) ([u1wire_obj_t](#) *devices, uint8_t devices_max)

5.100.1 Detailed Description

Low level routines to drive Dallas universal 1 wire bus. This only supports a single instance of a 1 wire bus.

Author:

M. P. Hayes

Date:

16 May 20002

5.100.2 Define Documentation**5.100.2.1 #define U1WIRE_DEBUG 1****5.100.2.2 #define U1WIRE_DRIVE()****Value:**

```
do {port_pin_config_output (U1WIRE_PORT, U1WIRE_BIT); \
    port_pin_set_low (U1WIRE_PORT, U1WIRE_BIT);} while (0)
```

5.100.2.3 #define U1WIRE_RELEASE() port_pin_config_pullup (U1WIRE_PORT, U1WIRE_BIT)**5.100.2.4 #define U1WIRE_TEST() port_pin_read (U1WIRE_PORT, U1WIRE_BIT)****5.100.3 Enumeration Type Documentation****5.100.3.1 anonymous enum****Enumerator:***U1WIRE_READ_ROM**U1WIRE_SKIP_ROM**U1WIRE_MATCH_ROM**U1WIRE_RECALL***5.100.3.2 anonymous enum****Enumerator:***U1WIRE_DELAY_OFFSET**U1WIRE_ADDR_BYTES*

5.100.4 Function Documentation

5.100.4.1 `uint8_t u1wire_bit_read (void)`

5.100.4.2 `void u1wire_bit_write (uint8_t value)`

5.100.4.3 `int8_t u1wire_broadcast (uint8_t command)`

5.100.4.4 `uint8_t u1wire_byte_read (void)`

5.100.4.5 `void u1wire_byte_write (uint8_t value)`

5.100.4.6 `int8_t u1wire_command (u1wire_t dev, uint8_t command)`

5.100.4.7 `int8_t u1wire_init (u1wire_obj_t * devices, uint8_t devices_max)`

5.100.4.8 `int8_t u1wire_read (void * data, uint8_t bytes)`

5.100.4.9 `bool u1wire_ready_p (void)`

5.100.4.10 `int8_t u1wire_reset (void)`

5.100.4.11 `int8_t u1wire_rom_code_read (u1wire_t dev)`

5.100.4.12 `int8_t u1wire_write (void * data, uint8_t bytes)`

5.101 u1wire.h File Reference

```
#include "config.h"
```

Data Structures

- union [ulwire_rom_code_t](#)
- struct [ulwire_obj_t](#)

Typedefs

- typedef [ulwire_obj_t](#) * [ulwire_t](#)

Enumerations

- enum {
 [U1WIRE_ERR_BUS_LOW](#) = 1, [U1WIRE_ERR_BUS_STUCK](#) = 2, [U1WIRE_ERR_PRESENCE_SHORT](#) = 3, [U1WIRE_ERR_PRESENCE_LONG](#) = 4,
 [U1WIRE_ERR_MULTIPLE_DEVICES](#) = 5, [U1WIRE_ERR_BUS_HIGH](#) = 6 }

Functions

- bool [ulwire_ready_p](#) (void)
- int8_t [ulwire_reset](#) (void)
- void [ulwire_bit_write](#) (uint8_t value)
- void [ulwire_byte_write](#) (uint8_t value)
- uint8_t [ulwire_bit_read](#) (void)
- uint8_t [ulwire_byte_read](#) (void)
- int8_t [ulwire_command](#) ([ulwire_t](#) dev, uint8_t command)
- int8_t [ulwire_broadcast](#) (uint8_t command)
- int8_t [ulwire_read](#) (void *data, uint8_t size)
- int8_t [ulwire_write](#) (void *data, uint8_t size)
- int8_t [ulwire_init](#) ([ulwire_obj_t](#) *devices, uint8_t devices_max)
- void [ulwire_debug](#) ([ulwire_t](#) dev)

5.101.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

08 June 2002

5.101.2 Typedef Documentation

5.101.2.1 typedef [u1wire_obj_t](#)* [u1wire_t](#)

5.101.3 Enumeration Type Documentation

5.101.3.1 anonymous enum

Enumerator:

UIWIRE_ERR_BUS_LOW

UIWIRE_ERR_BUS_STUCK

UIWIRE_ERR_PRESENCE_SHORT

UIWIRE_ERR_PRESENCE_LONG

UIWIRE_ERR_MULTIPLE_DEVICES

UIWIRE_ERR_BUS_HIGH

5.101.4 Function Documentation

5.101.4.1 `uint8_t u1wire_bit_read (void)`

5.101.4.2 `void u1wire_bit_write (uint8_t value)`

5.101.4.3 `int8_t u1wire_broadcast (uint8_t command)`

5.101.4.4 `uint8_t u1wire_byte_read (void)`

5.101.4.5 `void u1wire_byte_write (uint8_t value)`

5.101.4.6 `int8_t u1wire_command (u1wire_t dev, uint8_t command)`

5.101.4.7 `void u1wire_debug (u1wire_t dev)`

5.101.4.8 `int8_t u1wire_init (u1wire_obj_t * devices, uint8_t devices_max)`

5.101.4.9 `int8_t u1wire_read (void * data, uint8_t size)`

5.101.4.10 `bool u1wire_ready_p (void)`

5.101.4.11 `int8_t u1wire_reset (void)`

5.101.4.12 `int8_t u1wire_write (void * data, uint8_t size)`

5.102 u1wire_debug.c File Reference

```
#include "u1wire.h"  
#include <stdio.h>
```

Defines

- #define [U1WIRE_DEBUG](#) 1

Functions

- void [u1wire_debug](#) ([u1wire_t](#) dev)

5.102.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.102.2 Define Documentation

5.102.2.1 #define [U1WIRE_DEBUG](#) 1

5.102.3 Function Documentation

5.102.3.1 void [u1wire_debug](#) ([u1wire_t](#) dev)

5.103 u1wire_discover.c File Reference

```
#include "dscrc8.h"
#include "u1wire.h"
```

Data Structures

- struct [u1wire_state_t](#)

Enumerations

- enum { [UIWIRE_SEARCH](#) = 0xf0 }

Functions

- static bool [u1wire_search](#) ([u1wire_state_t](#) *state, [u1wire_rom_code_t](#) *rom_code)
- int8_t [u1wire_discover](#) ([u1wire_obj_t](#) *devices, uint8_t devices_max)

5.103.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.103.2 Enumeration Type Documentation

5.103.2.1 anonymous enum

Enumerator:

[UIWIRE_SEARCH](#)

5.103.3 Function Documentation

5.103.3.1 int8_t u1wire_discover ([u1wire_obj_t](#) * devices, uint8_t devices_max)

5.103.3.2 static bool u1wire_search ([u1wire_state_t](#) * state, [u1wire_rom_code_t](#) * rom_code) [static]

5.104 u1wire_discover.h File Reference

This discovers devices on a Dallas universal one wire bus. Note this has been superseded by u1wire_enumerate.

```
#include "u1wire.h"
```

Functions

- `int8_t u1wire_discover(u1wire_obj_t *devices, uint8_t devices_max)`

5.104.1 Detailed Description

This discovers devices on a Dallas universal one wire bus. Note this has been superseded by u1wire_enumerate.

Author:

M. P. Hayes, UCECE

Date:

24 February 2005

5.104.2 Function Documentation

5.104.2.1 `int8_t u1wire_discover(u1wire_obj_t * devices, uint8_t devices_max)`

5.105 ulwire_enumerate.c File Reference

```
#include "dscrc8.h"
#include "ulwire.h"
#include "ulwire_enumerate.h"
```

Enumerations

- enum { [UIWIRE_SEARCH](#) = 0xf0 }

Functions

- static bool [ulwire_search](#) ([ulwire_state_t](#) *state, [ulwire_rom_code_t](#) *rom_code)
- [ulwire_obj_t](#) * [ulwire_enumerate_next](#) ([ulwire_enumerate_t](#) *info)
- [ulwire_obj_t](#) * [ulwire_enumerate](#) ([ulwire_enumerate_t](#) *info)

5.105.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.105.2 Enumeration Type Documentation

5.105.2.1 anonymous enum

Enumerator:

[UIWIRE_SEARCH](#)

5.105.3 Function Documentation

5.105.3.1 [ulwire_obj_t](#)* [ulwire_enumerate](#) ([ulwire_enumerate_t](#) * *info*)

5.105.3.2 [ulwire_obj_t](#)* [ulwire_enumerate_next](#) ([ulwire_enumerate_t](#) * *info*)

5.105.3.3 static bool [ulwire_search](#) ([ulwire_state_t](#) * *state*, [ulwire_rom_code_t](#) * *rom_code*)
[static]

5.106 u1wire_enumerate.h File Reference

This discovers devices on a Dallas universal one wire bus.

```
#include "u1wire.h"
```

Data Structures

- struct [u1wire_state_t](#)
- struct [u1wire_enumerate_t](#)

Functions

- [u1wire_obj_t](#) * [u1wire_enumerate](#) ([u1wire_enumerate_t](#) *info)
- [u1wire_obj_t](#) * [u1wire_enumerate_next](#) ([u1wire_enumerate_t](#) *info)

5.106.1 Detailed Description

This discovers devices on a Dallas universal one wire bus.

Author:

M. P. Hayes, UCECE

Date:

24 February 2005

5.106.2 Function Documentation

5.106.2.1 [u1wire_obj_t](#)* [u1wire_enumerate](#) ([u1wire_enumerate_t](#) * *info*)

5.106.2.2 [u1wire_obj_t](#)* [u1wire_enumerate_next](#) ([u1wire_enumerate_t](#) * *info*)

5.107 uint16toa.c File Reference

16 bit unsigned int to ASCII conversion.

```
#include "config.h"
```

Functions

- void `uint16toa` (uint16_t num, char *str, bool leading_zeroes)

5.107.1 Detailed Description

16 bit unsigned int to ASCII conversion.

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.107.2 Function Documentation

5.107.2.1 void `uint16toa` (uint16_t *num*, char * *str*, bool *leading_zeroes*)

Convert 16 bit unsigned integer to ASCII.

5.108 uint16toa.h File Reference

16 bit unsigned int to ASCII conversion.

```
#include "config.h"
```

Functions

- void [uint16toa](#) (uint16_t num, char *str, bool leading_zeroes)

5.108.1 Detailed Description

16 bit unsigned int to ASCII conversion.

Author:

M. P. Hayes, UCECE

Date:

21 Nov 2006

5.108.2 Function Documentation

5.108.2.1 void [uint16toa](#) (uint16_t *num*, char * *str*, bool *leading_zeroes*)

Convert 16 bit unsigned integer to ASCII.

5.109 uint8toa.c File Reference

```
#include "config.h"
```

Functions

- void [uint8toa](#) (uint8_t num, char *str, bool leading_zeroes)

5.109.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

15 May 2007

5.109.2 Function Documentation

5.109.2.1 void [uint8toa](#) (uint8_t *num*, char * *str*, bool *leading_zeroes*)

5.110 uint8toa.h File Reference

```
#include "config.h"
```

Functions

- void [uint8toa](#) (uint8_t num, char *str, bool leading_zeroes)

5.110.1 Detailed Description

Author:

M. P. Hayes, UCECE

Date:

21 Nov 2006

5.110.2 Function Documentation

5.110.2.1 void [uint8toa](#) (uint8_t *num*, char * *str*, bool *leading_zeroes*)

5.111 usart.c File Reference

Unbuffered USART implementation.

```
#include "usart.h"
#include "peripherals.h"
```

Data Structures

- struct [usart_dev_struct](#)

Defines

- #define [USART0_ENABLE](#) (USART_NUM >= 1)
- #define [USART1_ENABLE](#) (USART_NUM >= 2)

Functions

- [usart_t usart_init](#) (uint8_t channel, uint16_t baud_divisor)
- bool [usart_read_ready_p](#) (usart_t usart)
- bool [usart_write_ready_p](#) (usart_t usart)
- bool [usart_write_finished_p](#) (usart_t usart)
- int8_t [usart_getc](#) (usart_t usart)
- int8_t [usart_putc](#) (usart_t usart, char ch)
- int8_t [usart_puts](#) (usart_t usart, const char *str)

5.111.1 Detailed Description

Unbuffered USART implementation.

Author:

M. P. Hayes, UCECE

Date:

21 June 2007

5.111.2 Define Documentation

5.111.2.1 `#define USART0_ENABLE (USART_NUM >= 1)`

5.111.2.2 `#define USART1_ENABLE (USART_NUM >= 2)`

5.111.3 Function Documentation

5.111.3.1 `int8_t usart_getc (usart_t usart)`

5.111.3.2 `usart_t usart_init (uint8_t channel, uint16_t baud_divisor)`

5.111.3.3 `int8_t usart_putc (usart_t usart, char ch)`

5.111.3.4 `int8_t usart_puts (usart_t usart, const char * str)`

5.111.3.5 `bool usart_read_ready_p (usart_t usart)`

5.111.3.6 `bool usart_write_finished_p (usart_t usart)`

5.111.3.7 `bool usart_write_ready_p (usart_t usart)`

5.112 usart.h File Reference

Unbuffered USART interface.

```
#include "config.h"
#include "usart0.h"
```

Defines

- #define [USART_BAUD_DIVISOR](#)(BAUD_RATE) USART0_BAUD_DIVISOR(BAUD_RATE)

Typedefs

- typedef [usart_dev_struct](#) [usart_dev_t](#)
- typedef [usart_dev_t](#) * [usart_t](#)

Functions

- [usart_t](#) [usart_init](#) (uint8_t channel, uint16_t baud_divisor)
- bool [usart_read_ready_p](#) ([usart_t](#) usart)
- bool [usart_write_ready_p](#) ([usart_t](#) usart)
- int8_t [usart_getc](#) ([usart_t](#) usart)
- int8_t [usart_putc](#) ([usart_t](#) usart, char ch)
- int8_t [usart_puts](#) ([usart_t](#) usart, const char *str)

5.112.1 Detailed Description

Unbuffered USART interface.

Author:

M. P. Hayes, UCECE

Date:

21 June 2007

5.112.2 Define Documentation

5.112.2.1 `#define USART_BAUD_DIVISOR(BAUD_RATE) USART0_BAUD_DIVISOR(BAUD_RATE)`

5.112.3 Typedef Documentation

5.112.3.1 `typedef struct usart_dev_struct usart_dev_t`

5.112.3.2 `typedef usart_dev_t* usart_t`

5.112.4 Function Documentation

5.112.4.1 `int8_t usart_getc (usart_t usart)`

5.112.4.2 `usart_t usart_init (uint8_t channel, uint16_t baud_divisor)`

5.112.4.3 `int8_t usart_putc (usart_t usart, char ch)`

5.112.4.4 `int8_t usart_puts (usart_t usart, const char * str)`

5.112.4.5 `bool usart_read_ready_p (usart_t usart)`

5.112.4.6 `bool usart_write_ready_p (usart_t usart)`

Index

addr_1
 nrf_config_bits_t, [40](#)
addr_2
 nrf_config_bits_t, [40](#)
addr_w
 nrf_config_bits_t, [40](#)
ads7870.c, [71](#)
 ADS7870_ADCTRL, [73](#)
 ads7870_channel_convert, [73](#)
 ads7870_channel_start, [73](#)
 ads7870_chip_deselect, [73](#)
 ads7870_chip_select, [73](#)
 ADS7870_CONVERT, [72](#)
 ADS7870_DIGIOCTRL, [73](#)
 ADS7870_DIGIOSTATE, [73](#)
 ADS7870_GAINMUX, [73](#)
 ADS7870_GAINMUX_CNVBSY, [72](#)
 ads7870_init, [73](#)
 ADS7870_PGA_GAIN_1, [72](#)
 ADS7870_PGA_GAIN_10, [72](#)
 ADS7870_PGA_GAIN_16, [72](#)
 ADS7870_PGA_GAIN_2, [72](#)
 ADS7870_PGA_GAIN_20, [72](#)
 ADS7870_PGA_GAIN_4, [72](#)
 ADS7870_PGA_GAIN_5, [72](#)
 ADS7870_PGA_GAIN_8, [72](#)
 ads7870_pga_gain_t, [72](#)
 ADS7870_PGAVALID, [73](#)
 ads7870_read, [73](#)
 ads7870_read_ready_p, [73](#)
 ADS7870_REFOSC, [73](#)
 ADS7870_REFOSC_BUFE, [72](#)
 ADS7870_REFOSC_OSCE, [72](#)
 ADS7870_REFOSC_OSCR, [72](#)
 ADS7870_REFOSC_R2V, [72](#)
 ADS7870_REFOSC_RBE, [72](#)
 ADS7870_REFOSC_REFE, [72](#)
 ADS7870_REG_16BIT, [72](#)
 ADS7870_REG_READ, [72](#)
 ads7870_reg_read, [73](#)
 ADS7870_REG_WRITE, [72](#)
 ads7870_reg_write, [73](#)
 ads7870_register_t, [72](#)
 ADS7870_RESULTHI, [73](#)
 ADS7870_RESULTLO, [73](#)
 ads7870.h, [74](#)
 ads7870_channel_convert, [74](#)
 ads7870_channel_start, [74](#)
 ads7870_convert, [74](#)
 ads7870_init, [74](#)
 ads7870_read, [74](#)
 ADS7870_ADCTRL
 ads7870.c, [73](#)
 ads7870_channel_convert
 ads7870.c, [73](#)
 ads7870.h, [74](#)
 ads7870_channel_start
 ads7870.c, [73](#)
 ads7870.h, [74](#)
 ads7870_chip_deselect
 ads7870.c, [73](#)
 ads7870_chip_select
 ads7870.c, [73](#)
 ADS7870_CONVERT
 ads7870.c, [72](#)
 ads7870_convert
 ads7870.h, [74](#)
 ADS7870_DIGIOCTRL
 ads7870.c, [73](#)
 ADS7870_DIGIOSTATE
 ads7870.c, [73](#)
 ADS7870_GAINMUX
 ads7870.c, [73](#)
 ADS7870_GAINMUX_CNVBSY
 ads7870.c, [72](#)
 ads7870_init
 ads7870.c, [73](#)
 ads7870.h, [74](#)
 ADS7870_PGA_GAIN_1
 ads7870.c, [72](#)
 ADS7870_PGA_GAIN_10
 ads7870.c, [72](#)
 ADS7870_PGA_GAIN_16
 ads7870.c, [72](#)
 ADS7870_PGA_GAIN_2
 ads7870.c, [72](#)
 ADS7870_PGA_GAIN_20
 ads7870.c, [72](#)
 ADS7870_PGA_GAIN_4
 ads7870.c, [72](#)

- ADS7870_PGA_GAIN_5
 - ads7870.c, [72](#)
- ADS7870_PGA_GAIN_8
 - ads7870.c, [72](#)
- ads7870_pga_gain_t
 - ads7870.c, [72](#)
- ADS7870_PGAVAILD
 - ads7870.c, [73](#)
- ads7870_read
 - ads7870.c, [73](#)
 - ads7870.h, [74](#)
- ads7870_read_ready_p
 - ads7870.c, [73](#)
- ADS7870_REFOSC
 - ads7870.c, [73](#)
- ADS7870_REFOSC_BUFE
 - ads7870.c, [72](#)
- ADS7870_REFOSC_OSCE
 - ads7870.c, [72](#)
- ADS7870_REFOSC_OSCR
 - ads7870.c, [72](#)
- ADS7870_REFOSC_R2V
 - ads7870.c, [72](#)
- ADS7870_REFOSC_RBE
 - ads7870.c, [72](#)
- ADS7870_REFOSC_REFE
 - ads7870.c, [72](#)
- ADS7870_REG_16BIT
 - ads7870.c, [72](#)
- ADS7870_REG_READ
 - ads7870.c, [72](#)
- ads7870_reg_read
 - ads7870.c, [73](#)
- ADS7870_REG_WRITE
 - ads7870.c, [72](#)
- ads7870_reg_write
 - ads7870.c, [73](#)
- ads7870_register_t
 - ads7870.c, [72](#)
- ADS7870_RESULTHI
 - ads7870.c, [73](#)
- ADS7870_RESULTLO
 - ads7870.c, [73](#)
- ads8325.c, [75](#)
 - ads8325_chip_deselect, [75](#)
 - ads8325_chip_select, [75](#)
 - ads8325_init, [75](#)
 - ads8325_read, [75](#)
- ads8325_chip_deselect
 - ads8325.c, [75](#)
- ads8325_chip_select
 - ads8325.c, [75](#)
- ads8325_init
 - ads8325.c, [75](#)
- ads8325_read
 - ads8325.c, [75](#)
- ads8325_read
 - ads8325.c, [75](#)
- ads8327.c, [76](#)
 - ads8327_chip_deselect, [76](#)
 - ads8327_chip_select, [76](#)
 - ads8327_command, [76](#)
 - ads8327_convert, [76](#)
 - ads8327_init, [76](#)
 - ads8327_read, [76](#)
- ads8327.h, [77](#)
 - ads8327_convert, [77](#)
 - ads8327_init, [77](#)
 - ads8327_read, [77](#)
- ads8327_chip_deselect
 - ads8327.c, [76](#)
- ads8327_chip_select
 - ads8327.c, [76](#)
- ads8327_command
 - ads8327.c, [76](#)
- ads8327_convert
 - ads8327.c, [76](#)
 - ads8327.h, [77](#)
- ads8327_init
 - ads8327.c, [76](#)
 - ads8327.h, [77](#)
- ads8327_read
 - ads8327.c, [76](#)
 - ads8327.h, [77](#)
- biseq.c, [78](#)
 - biseq_get, [78](#)
 - biseq_init, [78](#)
 - biseq_mode_get, [78](#)
 - biseq_mode_set, [78](#)
 - biseq_set, [78](#)
 - biseq_update, [78](#)
- biseq.h, [79](#)
 - biseq_get, [80](#)
 - biseq_init, [80](#)
 - BISEQ_MODE_CYCLE, [79](#)
 - biseq_mode_get, [80](#)
 - BISEQ_MODE_NORMAL, [79](#)
 - BISEQ_MODE_NUM, [79](#)
 - biseq_mode_set, [80](#)
 - biseq_mode_t, [79](#)
 - biseq_set, [80](#)
 - biseq_t, [79](#)
 - biseq_update, [80](#)
- biseq_get
 - biseq.c, [78](#)
 - biseq.h, [80](#)
- biseq_init
 - biseq.c, [78](#)
 - biseq.h, [80](#)

- BISEQ_MODE_CYCLE
 - biseq.h, 79
- biseq_mode_get
 - biseq.c, 78
 - biseq.h, 80
- BISEQ_MODE_NORMAL
 - biseq.h, 79
- BISEQ_MODE_NUM
 - biseq.h, 79
- biseq_mode_set
 - biseq.c, 78
 - biseq.h, 80
- biseq_mode_t
 - biseq.h, 79
- biseq_obj_t, 9
 - callback, 9
 - callback_data, 9
 - dir, 9
 - mode, 9
 - step, 9
 - str, 9
- biseq_set
 - biseq.c, 78
 - biseq.h, 80
- biseq_t
 - biseq.h, 79
- biseq_update
 - biseq.c, 78
 - biseq.h, 80
- bitmask
 - button_cfg_t, 11
 - led_cfg_t, 22
 - lmatrix_port_t, 23
 - muxleds_cfg_t, 34
 - muxleds_col_t, 35
 - muxleds_row_t, 37
 - piezo_cfg_t, 46
- bits
 - nrf_config_t, 41
- bits.h, 81
 - BITS_CLR, 81
 - BITS_EXTRACT, 81
 - BITS_INSERT, 81
 - BITS_MASK, 81
 - BITS_SET, 81
- BITS_CLR
 - bits.h, 81
- BITS_EXTRACT
 - bits.h, 81
- BITS_INSERT
 - bits.h, 81
- BITS_MASK
 - bits.h, 81
- BITS_SET
 - bits.h, 81
- busart.c, 82
 - BUSART0_ENABLE, 83
 - BUSART1_ENABLE, 83
 - busart_getc, 83
 - busart_init, 83
 - busart_putc, 83
 - busart_puts, 83
 - busart_read, 83
 - busart_read_block, 83
 - busart_read_num, 83
 - busart_read_ready_p, 83
 - busart_write, 83
 - busart_write_block, 83
 - busart_write_finished_p, 83
 - busart_write_num, 83
 - busart_write_ready_p, 83
- busart.h, 84
 - BUSART_BAUD_DIVISOR, 85
 - busart_dev_t, 85
 - busart_getc, 85
 - busart_init, 85
 - busart_putc, 85
 - busart_puts, 85
 - busart_read, 85
 - busart_read_block, 85
 - busart_read_num, 85
 - busart_read_ready_p, 85
 - busart_t, 85
 - busart_write, 85
 - busart_write_block, 85
 - busart_write_finished_p, 85
 - busart_write_num, 85
 - busart_write_ready_p, 85
- BUSART0_ENABLE
 - busart.c, 83
- BUSART1_ENABLE
 - busart.c, 83
- BUSART_BAUD_DIVISOR
 - busart.h, 85
- busart_dev_struct, 10
 - rx_irq_enable, 10
 - rx_ring, 10
 - tx_finished_p, 10
 - tx_irq_enable, 10
 - tx_ring, 10
- busart_dev_t
 - busart.h, 85
- busart_getc
 - busart.c, 83
 - busart.h, 85
- busart_init
 - busart.c, 83
 - busart.h, 85

- busart_putc
 - busart.c, 83
 - busart.h, 85
- busart_puts
 - busart.c, 83
 - busart.h, 85
- busart_read
 - busart.c, 83
 - busart.h, 85
- busart_read_block
 - busart.c, 83
 - busart.h, 85
- busart_read_num
 - busart.c, 83
 - busart.h, 85
- busart_read_ready_p
 - busart.c, 83
 - busart.h, 85
- busart_t
 - busart.h, 85
- busart_write
 - busart.c, 83
 - busart.h, 85
- busart_write_block
 - busart.c, 83
 - busart.h, 85
- busart_write_finished_p
 - busart.c, 83
 - busart.h, 85
- busart_write_num
 - busart.c, 83
 - busart.h, 85
- busart_write_ready_p
 - busart.c, 83
 - busart.h, 85
- button.c, 86
 - button_debounce, 86
 - button_held_p, 86
 - button_hold_released_p, 86
 - button_init, 86
 - button_poll, 86
 - button_poll_count, 87
 - button_poll_count_set, 87
- button.h, 88
 - BUTTON_CFG, 89
 - BUTTON_DEBOUNCE_MS, 89
 - BUTTON_DEBOUNCE_RATE, 89
 - button_down_p, 89
 - button_held_p, 89
 - button_hold_count_get, 90
 - button_hold_released_p, 90
 - button_init, 90
 - button_obj_t, 89
 - button_poll, 90
- BUTTON_POLL_COUNT, 89
 - button_poll_count_set, 90
 - button_pressed_p, 90
 - button_pushed_p, 90
 - button_released_p, 90
 - BUTTON_STATE_DOWN, 89
 - button_state_get, 90
 - BUTTON_STATE_PUSHED, 89
 - BUTTON_STATE_RELEASED, 89
 - button_state_t, 89
 - BUTTON_STATE_UP, 89
 - button_t, 89
- BUTTON_CFG
 - button.h, 89
- button_cfg_t, 11
 - bitmask, 11
 - port, 11
- button_debounce
 - button.c, 86
- BUTTON_DEBOUNCE_MS
 - button.h, 89
- BUTTON_DEBOUNCE_RATE
 - button.h, 89
- button_down_p
 - button.h, 89
- button_held_p
 - button.c, 86
 - button.h, 89
- button_hold_count_get
 - button.h, 90
- button_hold_released_p
 - button.c, 86
 - button.h, 90
- button_init
 - button.c, 86
 - button.h, 90
- button_obj_t
 - button.h, 89
- button_poll
 - button.c, 86
 - button.h, 90
- BUTTON_POLL_COUNT
 - button.h, 89
- button_poll_count
 - button.c, 87
- button_poll_count_set
 - button.c, 87
 - button.h, 90
- button_pressed_p
 - button.h, 90
- button_private_t, 12
 - cfg, 12
 - count, 12
 - hold_count, 12

- state, 12
- button_pushed_p
 - button.h, 90
- button_released_p
 - button.h, 90
- BUTTON_STATE_DOWN
 - button.h, 89
- button_state_get
 - button.h, 90
- BUTTON_STATE_PUSHED
 - button.h, 89
- BUTTON_STATE_RELEASED
 - button.h, 89
- button_state_t
 - button.h, 89
- BUTTON_STATE_UP
 - button.h, 89
- button_t
 - button.h, 89
- buttons
 - buttons_private_t, 13
 - mbuttons_private_t, 25
- buttons.c, 91
 - buttons_any_pushed_p, 91
 - buttons_init, 91
 - buttons_poll, 91
- buttons.h, 92
 - buttons_any_pushed_p, 92
 - buttons_held_p, 92
 - buttons_hold_released_p, 93
 - buttons_init, 93
 - buttons_obj_t, 92
 - buttons_poll, 93
 - buttons_pushed_p, 93
 - buttons_released_p, 93
 - buttons_t, 92
- buttons_any_pushed_p
 - buttons.c, 91
 - buttons.h, 92
- buttons_held_p
 - buttons.h, 92
- buttons_hold_released_p
 - buttons.h, 93
- buttons_init
 - buttons.c, 91
 - buttons.h, 93
- buttons_obj_t
 - buttons.h, 92
- buttons_poll
 - buttons.c, 91
 - buttons.h, 93
- buttons_private_t, 13
 - buttons, 13
 - num, 13
- buttons_pushed_p
 - buttons.h, 93
- buttons_released_p
 - buttons.h, 93
- buttons_t
 - buttons.h, 92
- bytes
 - mulwire_rom_code_t, 33
 - nrf_config_t, 41
 - rf_address_t, 47
 - ulwire_rom_code_t, 67
- callback
 - biseq_obj_t, 9
 - seq_obj_t, 52
 - stext_obj_t, 59
- callback_data
 - biseq_obj_t, 9
 - seq_obj_t, 52
 - stext_obj_t, 59
- ce_bitmask
 - nrf_pins_t, 43
- ce_bitno
 - nrf_cfg_t, 38
- ce_port
 - nrf_cfg_t, 38
 - nrf_pins_t, 43
- cfg
 - button_private_t, 12
 - lcd_obj_t, 21
- channel
 - rf_node_t, 48
- channels
 - mpwm_obj_t, 30
- CHAR_BIT
 - nrf2401.c, 163
- chaser.c, 94
 - chaser_init, 94
 - chaser_mode_set, 94
 - chaser_pixel_set, 94
 - chaser_sequence_set, 94
 - chaser_update, 94
- chaser.h, 95
 - chaser_font_index_t, 96
 - chaser_font_set, 96
 - chaser_init, 96
 - CHASER_MODE_CYCLE, 96
 - CHASER_MODE_CYCLE_INVERT, 96
 - CHASER_MODE_INVERT, 96
 - CHASER_MODE_NORMAL, 96
 - CHASER_MODE_NUM, 96
 - chaser_mode_set, 96
 - chaser_mode_t, 96
 - chaser_obj_t, 96

- chaser_patterns_set, 96
- chaser_sequence_get, 96
- chaser_sequence_set, 96
- chaser_sequence_t, 96
- chaser_t, 96
- chaser_update, 96
- chaser_font_index_t
 - chaser.h, 96
- chaser_font_set
 - chaser.h, 96
- chaser_init
 - chaser.c, 94
 - chaser.h, 96
- CHASER_MODE_CYCLE
 - chaser.h, 96
- CHASER_MODE_CYCLE_INVERT
 - chaser.h, 96
- CHASER_MODE_INVERT
 - chaser.h, 96
- CHASER_MODE_NORMAL
 - chaser.h, 96
- CHASER_MODE_NUM
 - chaser.h, 96
- chaser_mode_set
 - chaser.c, 94
 - chaser.h, 96
- chaser_mode_t
 - chaser.h, 96
- chaser_obj_t
 - chaser.h, 96
- chaser_patterns_set
 - chaser.h, 96
- chaser_pixel_set
 - chaser.c, 94
- chaser_private_t, 14
 - dir, 14
 - flasher_num, 14
 - flashers, 14
 - font, 14
 - mode, 14
 - off_pattern, 14
 - on_pattern, 14
 - seq, 14
 - step, 14
- chaser_sequence_get
 - chaser.h, 96
- chaser_sequence_set
 - chaser.c, 94
 - chaser.h, 96
- chaser_sequence_t
 - chaser.h, 96
- chaser_t
 - chaser.h, 96
- chaser_update
 - chaser.c, 94
 - chaser.h, 96
- cleds
 - mcleds_private_t, 26
- cleds.c, 97
 - cleds_common_set, 97
 - cleds_init, 97
- cleds.h, 98
 - cleds_active_row_get, 99
 - cleds_cols_num_get, 99
 - cleds_common_cycle, 99
 - cleds_common_set, 99
 - cleds_init, 99
 - cleds_obj_t, 99
 - cleds_rows_num_get, 99
 - cleds_set, 99
 - cleds_t, 99
- cleds_active_row_get
 - cleds.h, 99
- cleds_cols_num_get
 - cleds.h, 99
- cleds_common_cycle
 - cleds.h, 99
- cleds_common_set
 - cleds.c, 97
 - cleds.h, 99
- cleds_init
 - cleds.c, 97
 - cleds.h, 99
- cleds_obj_t
 - cleds.h, 99
- cleds_private_t, 15
 - cols_num, 15
 - leds, 15
 - row, 15
 - row_config, 15
 - rows_num, 15
- cleds_rows_num_get
 - cleds.h, 99
- cleds_set
 - cleds.h, 99
- cleds_t
 - cleds.h, 99
- clock
 - ticker16_t, 60
 - ticker8_t, 61
 - ticker_t, 62
- cm
 - nrf_config_bits_t, 40
- col
 - lmatrix_private_t, 24
 - muxleds_obj_t, 36
- col_port
 - lmatrix_private_t, 24

- col_ports
 - lmatrix.c, 135
- colourmap
 - mcleds_private_t, 26
- colourmap.h, 100
 - COLOURMAP_B, 101
 - COLOURMAP_B_WEIGHT, 101
 - colourmap_elt_t, 101
 - COLOURMAP_ENTRY, 101
 - COLOURMAP_G, 101
 - COLOURMAP_G_WEIGHT, 101
 - COLOURMAP_R, 101
 - COLOURMAP_R_WEIGHT, 101
 - colourmap_t, 101
- COLOURMAP_B
 - colourmap.h, 101
- COLOURMAP_B_WEIGHT
 - colourmap.h, 101
- colourmap_elt_t
 - colourmap.h, 101
- COLOURMAP_ENTRY
 - colourmap.h, 101
- COLOURMAP_G
 - colourmap.h, 101
- COLOURMAP_G_WEIGHT
 - colourmap.h, 101
- COLOURMAP_R
 - colourmap.h, 101
- COLOURMAP_R_WEIGHT
 - colourmap.h, 101
- colourmap_size
 - mcleds_private_t, 26
- colourmap_t
 - colourmap.h, 101
- cols
 - muxleds_obj_t, 36
 - scroller_obj_t, 51
- cols_num
 - cleds_private_t, 15
 - mbuttons_private_t, 25
 - muxleds_obj_t, 36
- COMMS_INT
 - time.h, 225
- config
 - nrf_obj_t, 42
- count
 - button_private_t, 12
 - mpwm_obj_t, 30
 - spwm_obj_t, 56
- crc
 - mulwire_rom_code_t, 33
 - ulwire_rom_code_t, 67
- crc16_t
 - dscrc16.h, 113
- crc8541
 - crc8541.c, 102
 - crc8541.h, 103
- crc8541.c, 102
 - crc8541, 102
 - crc8541_bit, 102
 - crc8541_byte, 102
- crc8541.h, 103
 - crc8541, 103
 - crc8541_byte, 103
 - crc8_t, 103
- crc8541_bit
 - crc8541.c, 102
- crc8541_byte
 - crc8541.c, 102
 - crc8541.h, 103
- crc8541_test.c, 104
 - main, 104
- crc8_t
 - crc8541.h, 103
 - dscrc8.h, 115
- crc_en
 - nrf_config_bits_t, 40
- crc_l
 - nrf_config_bits_t, 40
- cs_bitmask
 - nrf_pins_t, 43
 - pga_cfg_t, 44
 - pga_private_t, 45
 - spi_eeprom_private_t, 55
- cs_bitno
 - nrf_cfg_t, 38
 - spi_eeprom_cfg_t, 54
- cs_port
 - nrf_cfg_t, 38
 - nrf_pins_t, 43
 - pga_cfg_t, 44
 - pga_private_t, 45
 - spi_eeprom_cfg_t, 54
 - spi_eeprom_private_t, 55
- cur
 - mmelody_private_t, 28
 - mtext_obj_t, 31
 - seq_obj_t, 52
 - squeaker_private_t, 58
- current
 - sflash_obj_t, 53
- d_bit
 - lcd_cfg_t, 20
- data
 - font_t, 19
 - lcd_obj_t, 21
- data1_w

- nrf_config_bits_t, 40
- data2_w
 - nrf_config_bits_t, 40
- data_port
 - lcd_cfg_t, 20
- DELAY_MAX
 - time.h, 225
- DELAY_MIN
 - time.h, 225
- delay_ret_t
 - time.h, 225
- device
 - u1wire_enumerate_t, 65
- device_id
 - rf_probe_t, 49
- dir
 - biseq_obj_t, 9
 - chaser_private_t, 14
 - scroller_obj_t, 51
- dr_bitmask
 - nrf_pins_t, 43
- dr_bitno
 - nrf_cfg_t, 38
- dr_port
 - nrf_cfg_t, 38
 - nrf_pins_t, 43
- DS1820_FAMILY_CODE
 - ds18b20.c, 106
- DS1820_TEMP_BITS
 - ds18b20.h, 108
- ds18b20.c, 105
 - DS1820_FAMILY_CODE, 106
 - DS18B20_CONVERT_T, 106
 - DS18B20_CRC_CHECK, 106
 - ds18b20_data, 106
 - DS18B20_DEBUG, 106
 - ds18b20_device_p, 106
 - DS18B20_FAMILY_CODE, 106
 - ds18b20_init, 106
 - DS18B20_READ_SCRATCHPAD, 106
 - DS18B20_SCRATCHPAD_BYTES, 106
 - ds18b20_temp_conversion_start, 106
 - ds18b20_temp_read, 106
 - ds18b20_temp_ready_p, 106
 - DS18B20_WRITE_SCRATCHPAD, 106
 - DS18S20_FAMILY_CODE, 106
- ds18b20.h, 107
 - DS1820_TEMP_BITS, 108
 - DS18B20_BITS_PER_DEGREE, 108
 - DS18B20_COUNTS_PER_DEGREE, 108
 - ds18b20_device_p, 108
 - ds18b20_init, 108
 - DS18B20_TEMP_BITS, 108
 - ds18b20_temp_conversion_start, 108
 - DS18B20_TEMP_DOUBLE, 108
 - DS18B20_TEMP_INT, 108
 - ds18b20_temp_read, 108
 - ds18b20_temp_ready_p, 108
 - ds18b20_temp_t, 108
 - DS18B20_BITS_PER_DEGREE
 - ds18b20.h, 108
 - DS18B20_CONVERT_T
 - ds18b20.c, 106
 - DS18B20_COUNTS_PER_DEGREE
 - ds18b20.h, 108
 - DS18B20_CRC_CHECK
 - ds18b20.c, 106
 - ds18b20_data
 - ds18b20.c, 106
 - DS18B20_DEBUG
 - ds18b20.c, 106
 - ds18b20_device_p
 - ds18b20.c, 106
 - ds18b20.h, 108
 - DS18B20_FAMILY_CODE
 - ds18b20.c, 106
 - ds18b20_init
 - ds18b20.c, 106
 - ds18b20.h, 108
 - DS18B20_READ_SCRATCHPAD
 - ds18b20.c, 106
 - DS18B20_SCRATCHPAD_BYTES
 - ds18b20.c, 106
 - DS18B20_TEMP_BITS
 - ds18b20.h, 108
 - ds18b20_temp_conversion_start
 - ds18b20.c, 106
 - ds18b20.h, 108
 - DS18B20_TEMP_DOUBLE
 - ds18b20.h, 108
 - DS18B20_TEMP_INT
 - ds18b20.h, 108
 - ds18b20_temp_read
 - ds18b20.c, 106
 - ds18b20.h, 108
 - ds18b20_temp_ready_p
 - ds18b20.c, 106
 - ds18b20.h, 108
 - ds18b20_temp_t
 - ds18b20.h, 108
 - DS18B20_WRITE_SCRATCHPAD
 - ds18b20.c, 106
 - DS18S20_FAMILY_CODE
 - ds18b20.c, 106
 - ds2450.c, 109
 - ds2450_adc_conversion_start, 110
 - ds2450_adc_read, 110
 - ds2450_adc_ready_p, 110

- DS2450_CONVERT, [110](#)
- DS2450_CRC_CHECK, [110](#)
- DS2450_DEBUG, [110](#)
- ds2450_debug, [110](#)
- ds2450_device_p, [110](#)
- DS2450_FAMILY_CODE, [110](#)
- ds2450_init, [110](#)
- DS2450_MEMORY_BYTES, [110](#)
- ds2450_memory_read, [110](#)
- ds2450_memory_write, [110](#)
- DS2450_READ_MEMORY, [110](#)
- DS2450_WRITE_MEMORY, [110](#)
- ds2450.h, [111](#)
- ds2450_adc_conversion_start, [111](#)
- ds2450_adc_read, [111](#)
- ds2450_adc_ready_p, [111](#)
- DS2450_CHANNELS_NUM, [111](#)
- ds2450_debug, [111](#)
- ds2450_device_p, [111](#)
- ds2450_init, [111](#)
- ds2450_adc_conversion_start
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- ds2450_adc_read
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- ds2450_adc_ready_p
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- DS2450_CHANNELS_NUM
 - ds2450.h, [111](#)
- DS2450_CONVERT
 - ds2450.c, [110](#)
- DS2450_CRC_CHECK
 - ds2450.c, [110](#)
- DS2450_DEBUG
 - ds2450.c, [110](#)
- ds2450_debug
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- ds2450_device_p
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- DS2450_FAMILY_CODE
 - ds2450.c, [110](#)
- ds2450_init
 - ds2450.c, [110](#)
 - ds2450.h, [111](#)
- DS2450_MEMORY_BYTES
 - ds2450.c, [110](#)
- ds2450_memory_read
 - ds2450.c, [110](#)
- ds2450_memory_write
 - ds2450.c, [110](#)
- DS2450_READ_MEMORY
 - ds2450.c, [110](#)
- DS2450_WRITE_MEMORY
 - ds2450.c, [110](#)
- dscrc16
 - dscrc16.c, [112](#)
 - dscrc16.h, [113](#)
- dscrc16.c, [112](#)
- dscrc16, [112](#)
- dscrc16_bit, [112](#)
- dscrc16_byte, [112](#)
- dscrc16.h, [113](#)
- crc16_t, [113](#)
- dscrc16, [113](#)
- dscrc16_byte, [113](#)
- dscrc16_bit
 - dscrc16.c, [112](#)
- dscrc16_byte
 - dscrc16.c, [112](#)
 - dscrc16.h, [113](#)
- dscrc8
 - dscrc8.c, [114](#)
 - dscrc8.h, [115](#)
- dscrc8.c, [114](#)
- dscrc8, [114](#)
- dscrc8_byte, [114](#)
- dscrc8.h, [115](#)
- crc8_t, [115](#)
- dscrc8, [115](#)
- dscrc8_byte, [115](#)
- dscrc8_byte
 - dscrc8.c, [114](#)
 - dscrc8.h, [115](#)
- dummy
 - flasher_obj_t, [16](#)
- duty
 - mcleds_state_t, [27](#)
 - mpwm_channel_t, [29](#)
 - spwm_obj_t, [56](#)
- e_bit
 - lcd_cfg_t, [20](#)
- e_mask
 - lcd_obj_t, [21](#)
- e_port
 - lcd_cfg_t, [20](#)
- end
 - ring_struct, [50](#)
- ERRD
 - time.h, [225](#)
- family
 - mulwire_rom_code_t, [33](#)
 - u1wire_rom_code_t, [67](#)

- fields
 - mulwire_rom_code_t, 33
 - u1wire_rom_code_t, 67
- flasher.c, 116
 - flasher_init, 116
 - flasher_pattern_get, 116
 - flasher_pattern_set, 116
 - flasher_phase_set, 116
 - FLASHER_TRANSPARENT, 116
 - flasher_update, 116
- flasher.h, 117
 - FLASHER_ACTIVE_P, 117
 - flasher_init, 118
 - FLASHER_PATTERN, 117
 - FLASHER_PATTERN_FLASHES_SET, 118
 - flasher_pattern_get, 118
 - flasher_pattern_set, 118
 - flasher_phase_set, 118
 - FLASHER_PRESCALE, 118
 - flasher_t, 118
 - flasher_update, 118
- FLASHER_ACTIVE_P
 - flasher.h, 117
- flasher_count
 - flasher_private_t, 18
- flasher_duty
 - flasher_pattern_t, 17
- flasher_init
 - flasher.c, 116
 - flasher.h, 118
- flasher_num
 - flasher_private_t, 14
- flasher_obj_t, 16
 - dummy, 16
- FLASHER_PATTERN
 - flasher.h, 117
- FLASHER_PATTERN_FLASHES_SET
 - flasher.h, 118
- flasher_pattern_get
 - flasher.c, 116
 - flasher.h, 118
- flasher_pattern_set
 - flasher.c, 116
 - flasher.h, 118
- flasher_pattern_t, 17
 - flasher_duty, 17
 - flasher_period, 17
 - flashes, 17
 - mod_duty, 17
 - mod_period, 17
 - period, 17
- flasher_period
 - flasher_pattern_t, 17
- flasher_phase_set
 - flasher.c, 116
 - flasher.h, 118
- FLASHER_PRESCALE
 - flasher.h, 118
- flasher_prescale
 - flasher_private_t, 18
- flasher_private_t, 18
 - flasher_count, 18
 - flasher_prescale, 18
 - flashes_count, 18
 - mod_count, 18
 - pattern, 18
- flasher_t
 - flasher.h, 118
- FLASHER_TRANSPARENT
 - flasher.c, 116
- flasher_tweak.c, 119
 - flasher_tweak_mod_duty, 119
 - flasher_tweak_mod_freq, 119
- flasher_tweak.h, 120
 - flasher_tweak_mod_duty, 120
 - flasher_tweak_mod_freq, 120
- flasher_tweak_mod_duty
 - flasher_tweak.c, 119
 - flasher_tweak.h, 120
- flasher_tweak_mod_freq
 - flasher_tweak.c, 119
 - flasher_tweak.h, 120
- flasher_update
 - flasher.c, 116
 - flasher.h, 118
- flashes
 - flasher_private_t, 14
- flashes
 - flasher_pattern_t, 17
- flashes_count
 - flasher_private_t, 18
- font
 - flasher_private_t, 14
 - mtext_obj_t, 31
 - stext_obj_t, 59
- font.c, 121
 - font_display, 121
- font.h, 122
 - font_display, 122
- font_display
 - font.c, 121
 - font.h, 122
- font_t, 19
 - data, 19
 - offset, 19
 - pixels, 19
 - size, 19

- getc
 - usart_dev_struct, 69
- HIGH_BYTE
 - rf.c, 184
- hold_count
 - button_private_t, 12
- holdoff
 - squeaker_private_t, 58
- id
 - rf_node_t, 48
- image
 - mtext_obj_t, 31
- in
 - ring_struct, 50
- index
 - scroller_obj_t, 51
- isqrt
 - isqrt16.c, 123
- isqrt16.c, 123
 - isqrt, 123
- isqrt32
 - isqrt32.c, 124
- isqrt32.c, 124
 - isqrt32, 124
- last_device
 - u1wire_state_t, 68
- last_discrepancy
 - u1wire_state_t, 68
- last_family_discrepancy
 - u1wire_state_t, 68
- lcd.c, 125
 - lcd_clear, 126
 - lcd_data_set, 125
 - LCD_DEBUG, 125
 - lcd_goto, 126
 - lcd_init, 126
 - lcd_init_data, 126
 - lcd_mode_control, 126
 - lcd_mode_data, 126
 - lcd_putc, 126
 - lcd_puts, 126
 - lcd_strobe, 126
 - lcd_wait, 126
 - lcd_write, 126
- lcd.h, 128
 - LCD_CG_RAM_ADDRESS, 129
 - LCD_CLEAR, 128
 - lcd_clear, 129
 - LCD_DD_RAM_ADDRESS, 129
 - LCD_DISPLAY, 129
 - LCD_ENTRY_MODE, 129
 - LCD_FUNCTION, 129
 - lcd_goto, 129
 - LCD_HOME, 128
 - lcd_init, 129
 - lcd_init_data, 129
 - lcd_mode_control, 129
 - lcd_mode_data, 129
 - lcd_obj_t, 21
 - cfg, 21
 - data, 21
 - e_mask, 21
 - rs_mask, 21
 - lcd_putc, 129
 - lcd_t, 128
 - lcd_cfg_t, 20
 - d_bit, 20
 - data_port, 20
 - e_bit, 20
 - e_port, 20
 - rs_bit, 20
 - rs_port, 20
 - LCD_CG_RAM_ADDRESS, 129
 - LCD_CLEAR, 128
 - lcd_clear, 129
 - lcd.c, 126
 - lcd.h, 129
 - lcd_data_set, 125
 - lcd.c, 125
 - LCD_DD_RAM_ADDRESS, 129
 - lcd.h, 129
 - LCD_DEBUG, 125
 - lcd.c, 125
 - LCD_DISPLAY, 129
 - lcd.h, 129
 - LCD_ENTRY_MODE, 129
 - lcd.h, 129
 - lcd_goto, 129
 - lcd.c, 126
 - lcd.h, 129
 - LCD_HOME, 128
 - lcd.h, 128
 - lcd_init, 129
 - lcd.c, 126
 - lcd.h, 129
 - lcd_init_data, 129
 - lcd.c, 126
 - lcd_mode_control, 129
 - lcd.c, 126
 - lcd_mode_data, 129
 - lcd.c, 126
 - lcd_obj_t, 21
 - cfg, 21
 - data, 21
 - e_mask, 21
 - rs_mask, 21
 - lcd_putc, 129

- lcd.c, 126
- lcd.h, 129
- lcd_puts
 - lcd.c, 126
 - lcd.h, 129
- LCD_SHIFT
 - lcd.h, 129
- lcd_strobe
 - lcd.c, 126
- lcd_t
 - lcd.h, 128
- lcd_wait
 - lcd.c, 126
- lcd_write
 - lcd.c, 126
- led.c, 130
 - led_init, 130
- led.h, 131
 - LED_CFG, 132
 - led_init, 132
 - led_obj_t, 132
 - led_set, 132
 - led_t, 132
 - led_toggle, 132
- LED_CFG
 - led.h, 132
- led_cfg_t, 22
 - bitmask, 22
 - port, 22
- led_flash
 - led_flash.c, 133
 - led_flash.h, 134
- led_flash.c, 133
 - led_flash, 133
- led_flash.h, 134
 - led_flash, 134
- led_init
 - led.c, 130
 - led.h, 132
- led_obj_t
 - led.h, 132
- led_set
 - led.h, 132
- led_t
 - led.h, 132
- led_toggle
 - led.h, 132
- leds
 - cleds_private_t, 15
- lmatrix.c, 135
 - col_ports, 135
 - lmatrix_init, 135
 - lmatrix_set, 135
 - LMATRIX_TRANSPARENT, 135
 - lmatrix_update, 135
 - lmatrix_write, 135
 - ROWBIT, 135
- lmatrix.h, 136
 - lmatrix_init, 137
 - lmatrix_obj_t, 137
 - LMATRIX_PIXELS, 137
 - lmatrix_row_state_t, 137
 - lmatrix_set, 137
 - lmatrix_t, 137
 - lmatrix_update, 137
 - lmatrix_write, 137
- lmatrix_init
 - lmatrix.c, 135
 - lmatrix.h, 137
- lmatrix_obj_t
 - lmatrix.h, 137
- LMATRIX_PIXELS
 - lmatrix.h, 137
- lmatrix_port_t, 23
 - bitmask, 23
 - port, 23
- lmatrix_private_t, 24
 - col, 24
 - col_port, 24
 - state, 24
- lmatrix_row_state_t
 - lmatrix.h, 137
- lmatrix_set
 - lmatrix.c, 135
 - lmatrix.h, 137
- lmatrix_t
 - lmatrix.h, 137
- LMATRIX_TRANSPARENT
 - lmatrix.c, 135
- lmatrix_update
 - lmatrix.c, 135
 - lmatrix.h, 137
- lmatrix_write
 - lmatrix.c, 135
 - lmatrix.h, 137
- loop_count
 - mmelody_private_t, 28
 - squeaker_private_t, 58
- loop_start
 - mmelody_private_t, 28
 - squeaker_private_t, 58
- LOW_BYTE
 - rf.c, 184
- main
 - crc8541_test.c, 104
- mbuttons.c, 138
 - mbuttons_any_state_p, 138

- mbuttons_init, 138
 - mbuttons_poll, 138
 - mbuttons_wakeup_init, 138
- mbuttons.h, 139
 - mbuttons_any_down_p, 140
 - mbuttons_any_pushed_p, 140
 - mbuttons_any_state_p, 140
 - mbuttons_held_p, 140
 - mbuttons_hold_released_p, 140
 - mbuttons_init, 140
 - mbuttons_obj_t, 140
 - mbuttons_poll, 140
 - mbuttons_pushed_p, 140
 - mbuttons_released_p, 140
 - mbuttons_t, 140
 - mbuttons_wakeup_init, 140
 - mbuttons_wakeup_p, 140
- mbuttons_any_down_p
 - mbuttons.h, 140
- mbuttons_any_pushed_p
 - mbuttons.h, 140
- mbuttons_any_state_p
 - mbuttons.c, 138
 - mbuttons.h, 140
- mbuttons_held_p
 - mbuttons.h, 140
- mbuttons_hold_released_p
 - mbuttons.h, 140
- mbuttons_init
 - mbuttons.c, 138
 - mbuttons.h, 140
- mbuttons_obj_t
 - mbuttons.h, 140
- mbuttons_poll
 - mbuttons.c, 138
 - mbuttons.h, 140
- mbuttons_private_t, 25
 - buttons, 25
 - cols_num, 25
 - row_config, 25
 - rows_num, 25
- mbuttons_pushed_p
 - mbuttons.h, 140
- mbuttons_released_p
 - mbuttons.h, 140
- mbuttons_t
 - mbuttons.h, 140
- mbuttons_wakeup_init
 - mbuttons.c, 138
 - mbuttons.h, 140
- mbuttons_wakeup_p
 - mbuttons.h, 140
- mcleds.c, 141
 - mcleds_enable, 141
- mcleds_init, 141
- mcleds_off, 141
- mcleds_update, 141
- mcleds.h, 142
 - mcleds_colourmap_set, 143
 - mcleds_disable, 143
 - mcleds_enable, 143
 - mcleds_init, 143
 - mcleds_obj_t, 143
 - mcleds_off, 143
 - mcleds_t, 143
 - mcleds_update, 143
- mcleds_colourmap_set
 - mcleds.h, 143
- mcleds_disable
 - mcleds.h, 143
- mcleds_enable
 - mcleds.c, 141
 - mcleds.h, 143
- mcleds_init
 - mcleds.c, 141
 - mcleds.h, 143
- mcleds_obj_t
 - mcleds.h, 143
- mcleds_off
 - mcleds.c, 141
 - mcleds.h, 143
- mcleds_private_t, 26
 - cleds, 26
 - colourmap, 26
 - colourmap_size, 26
 - primary_ticker, 26
 - state, 26
- mcleds_state_t, 27
 - duty, 27
- mcleds_t
 - mcleds.h, 143
- mcleds_update
 - mcleds.c, 141
 - mcleds.h, 143
- mmelody.c, 144
 - mmelody_char_to_note, 145
 - mmelody_init, 145
 - mmelody_note_fraction_set, 145
 - mmelody_note_play, 145
 - mmelody_play, 145
 - MMELODY_SCALE_SIZE, 144
 - mmelody_scan, 145
 - mmelody_speed_set, 145
 - mmelody_ticker_set, 145
 - MMELODY_TRANSPARENT, 144
 - mmelody_update, 145
 - mmelody_volume_set, 145
- mmelody.h, 146

- mmelody_callback_t, 147
- mmelody_init, 147
- mmelody_note_t, 147
- mmelody_obj_t, 147
- MMELODY_OCTAVE_DEFAULT, 147
- mmelody_play, 147
- mmelody_scale_t, 147
- MMELODY_SPEED_DEFAULT, 147
- mmelody_speed_set, 147
- mmelody_speed_t, 147
- mmelody_t, 147
- mmelody_update, 147
- mmelody_volume_set, 147
- mmelody_volume_t, 147
- mmelody_callback_t
 - mmelody.h, 147
- mmelody_char_to_note
 - mmelody.c, 145
- mmelody_init
 - mmelody.c, 145
 - mmelody.h, 147
- mmelody_note_fraction_set
 - mmelody.c, 145
- mmelody_note_play
 - mmelody.c, 145
- mmelody_note_t
 - mmelody.h, 147
- mmelody_obj_t
 - mmelody.h, 147
- MMELODY_OCTAVE_DEFAULT
 - mmelody.h, 147
- mmelody_play
 - mmelody.c, 145
 - mmelody.h, 147
- mmelody_private_t, 28
 - cur, 28
 - loop_count, 28
 - loop_start, 28
 - note_fraction, 28
 - octave, 28
 - play_callback, 28
 - play_callback_data, 28
 - poll_rate, 28
 - speed, 28
 - start, 28
 - ticker, 28
 - volume, 28
- MMELODY_SCALE_SIZE
 - mmelody.c, 144
- mmelody_scale_t
 - mmelody.h, 147
- mmelody_scan
 - mmelody.c, 145
- MMELODY_SPEED_DEFAULT
 - mmelody.h, 147
- mmelody_speed_set
 - mmelody.c, 145
 - mmelody.h, 147
- mmelody_speed_t
 - mmelody.h, 147
- mmelody_t
 - mmelody.h, 147
- mmelody_ticker_set
 - mmelody.c, 145
- MMELODY_TRANSPARENT
 - mmelody.c, 144
- mmelody_update
 - mmelody.c, 145
 - mmelody.h, 147
- mmelody_volume_set
 - mmelody.c, 145
 - mmelody.h, 147
- mmelody_volume_t
 - mmelody.h, 147
- mod_count
 - flasher_private_t, 18
- mod_duty
 - flasher_pattern_t, 17
- mod_period
 - flasher_pattern_t, 17
- mode
 - biseq_obj_t, 9
 - chaser_private_t, 14
 - mtext_obj_t, 31
- mpwm.c, 148
 - mpwm_duty_set, 148
 - mpwm_init, 148
 - mpwm_period_set, 148
 - mpwm_update, 148
- mpwm.h, 149
 - mpwm_duty_set, 149
 - mpwm_init, 149
 - mpwm_period_set, 149
 - mpwm_t, 149
 - mpwm_update, 149
- mpwm_channel_t, 29
 - duty, 29
- mpwm_duty_set
 - mpwm.c, 148
 - mpwm.h, 149
- mpwm_init
 - mpwm.c, 148
 - mpwm.h, 149
- mpwm_obj_t, 30
 - channels, 30
 - count, 30
 - num_channels, 30
 - period, 30

- mpwm_period_set
 - mpwm.c, 148
 - mpwm.h, 149
- mpwm_t
 - mpwm.h, 149
- mpwm_update
 - mpwm.c, 148
 - mpwm.h, 149
- ms_ticks
 - time, 63
 - time.c, 223
- mtext.c, 150
 - mtext_display, 150
 - mtext_init, 150
 - mtext_pixel_set, 150
 - mtext_scan, 150
 - mtext_scroller_dir_get, 150
 - mtext_scroller_dir_set, 150
 - mtext_speed_set, 150
 - mtext_update, 150
- mtext.h, 151
 - mtext_get, 152
 - mtext_init, 152
 - mtext_mode_get, 152
 - MTEXT_MODE_REPLACE, 152
 - MTEXT_MODE_SCROLL, 152
 - mtext_mode_set, 152
 - mtext_mode_t, 152
 - mtext_scroller_dir_get, 152
 - mtext_scroller_dir_set, 152
 - mtext_set, 152
 - MTEXT_SPEED_SCALER, 152
 - mtext_speed_set, 152
 - mtext_t, 152
 - mtext_update, 152
- mtext_display
 - mtext.c, 150
- mtext_get
 - mtext.h, 152
- mtext_init
 - mtext.c, 150
 - mtext.h, 152
- mtext_mode_get
 - mtext.h, 152
- MTEXT_MODE_REPLACE
 - mtext.h, 152
- MTEXT_MODE_SCROLL
 - mtext.h, 152
- mtext_mode_set
 - mtext.h, 152
- mtext_mode_t
 - mtext.h, 152
- mtext_obj_t, 31
 - cur, 31
 - font, 31
 - image, 31
 - mode, 31
 - pixels, 31
 - poll_rate, 31
 - screen, 31
 - scroller, 31
 - speed, 31
 - start, 31
 - ticker, 31
- mtext_pixel_set
 - mtext.c, 150
- mtext_scan
 - mtext.c, 150
- mtext_scroller_dir_get
 - mtext.c, 150
 - mtext.h, 152
- mtext_scroller_dir_set
 - mtext.c, 150
 - mtext.h, 152
- mtext_set
 - mtext.h, 152
- MTEXT_SPEED_SCALER
 - mtext.h, 152
- mtext_speed_set
 - mtext.c, 150
 - mtext.h, 152
- mtext_t
 - mtext.h, 152
- mtext_update
 - mtext.c, 150
 - mtext.h, 152
- mulwire.c, 153
 - U1WIRE_ADDR_BYTES, 154
 - ulwire_bit_read, 154
 - ulwire_bit_write, 154
 - ulwire_broadcast, 154
 - ulwire_byte_read, 154
 - ulwire_byte_write, 154
 - ulwire_command, 154
 - U1WIRE_DEBUG, 153
 - U1WIRE_DELAY_OFFSET, 154
 - U1WIRE_DRIVE, 153
 - ulwire_init, 154
 - U1WIRE_MATCH_ROM, 154
 - ulwire_read, 154
 - U1WIRE_READ_ROM, 154
 - ulwire_ready_p, 154
 - U1WIRE_RECALL, 154
 - U1WIRE_RELEASE, 153
 - ulwire_reset, 154
 - ulwire_rom_code_read, 154
 - U1WIRE_SKIP_ROM, 154
 - U1WIRE_TEST, 154

- ulwire_write, 154
- mulwire.h, 155
 - mulwire_bit_read, 156
 - mulwire_bit_write, 156
 - mulwire_broadcast, 156
 - mulwire_byte_read, 156
 - mulwire_byte_write, 156
 - mulwire_command, 156
 - mulwire_debug, 156
 - MU1WIRE_ERR_BUS_HIGH, 156
 - MU1WIRE_ERR_BUS_LOW, 156
 - MU1WIRE_ERR_BUS_STUCK, 156
 - MU1WIRE_ERR_MULTIPLE_DEVICES, 156
 - MU1WIRE_ERR_PRESENCE_LONG, 156
 - MU1WIRE_ERR_PRESENCE_SHORT, 156
 - mulwire_init, 156
 - mulwire_read, 156
 - mulwire_ready_p, 156
 - mulwire_reset, 156
 - mulwire_t, 156
 - mulwire_write, 156
- mulwire_bit_read
 - mulwire.h, 156
- mulwire_bit_write
 - mulwire.h, 156
- mulwire_broadcast
 - mulwire.h, 156
- mulwire_byte_read
 - mulwire.h, 156
- mulwire_byte_write
 - mulwire.h, 156
- mulwire_command
 - mulwire.h, 156
- mulwire_debug
 - mulwire.h, 156
- MU1WIRE_ERR_BUS_HIGH
 - mulwire.h, 156
- MU1WIRE_ERR_BUS_LOW
 - mulwire.h, 156
- MU1WIRE_ERR_BUS_STUCK
 - mulwire.h, 156
- MU1WIRE_ERR_MULTIPLE_DEVICES
 - mulwire.h, 156
- MU1WIRE_ERR_PRESENCE_LONG
 - mulwire.h, 156
- MU1WIRE_ERR_PRESENCE_SHORT
 - mulwire.h, 156
- mulwire_init
 - mulwire.h, 156
- mulwire_obj_t, 32
 - rom_code, 32
- mulwire_read
 - mulwire.h, 156
- mulwire_ready_p
 - mulwire.h, 156
- mulwire_reset
 - mulwire.h, 156
- mulwire_rom_code_t, 33
 - bytes, 33
 - crc, 33
 - family, 33
 - fields, 33
 - serial, 33
- mulwire_t
 - mulwire.h, 156
- mulwire_write
 - mulwire.h, 156
- MUXLED_COL_CFG
 - muxleds.h, 159
- MUXLED_ROW_CFG
 - muxleds.h, 159
- muxleds.c, 157
 - muxleds_init, 157
 - muxleds_set, 157
 - muxleds_toggle, 157
 - MUXLED_TRANSPARENT, 157
 - muxleds_update, 157
- muxleds.h, 158
 - MUXLED_COL_CFG, 159
 - MUXLED_ROW_CFG, 159
 - MUXLED_COLS_NUM, 159
 - muxleds_init, 159
 - MUXLED_ROWS_NUM, 159
 - muxleds_set, 159
 - muxleds_t, 159
 - muxleds_toggle, 159
 - muxleds_update, 159
- muxleds_cfg_t, 34
 - bitmask, 34
 - port, 34
- muxleds_col_t, 35
 - bitmask, 35
 - port, 35
 - row_state, 35
- MUXLED_COLS_NUM
 - muxleds.h, 159
- muxleds_init
 - muxleds.c, 157
 - muxleds.h, 159
- muxleds_obj_t, 36
 - col, 36
 - cols, 36
 - cols_num, 36
 - row_on, 36
 - rows, 36
 - rows_num, 36
- muxleds_row_t, 37

- bitmask, 37
- port, 37
- MUXLEDS_ROWS_NUM
 - muxleds.h, 159
- muxleds_set
 - muxleds.c, 157
 - muxleds.h, 159
- muxleds_t
 - muxleds.h, 159
- muxleds_toggle
 - muxleds.c, 157
 - muxleds.h, 159
- MUXLEDS_TRANSPARENT
 - muxleds.c, 157
- muxleds_update
 - muxleds.c, 157
 - muxleds.h, 159
- nmea.c, 160
 - nmea_checksum, 160
 - nmea_puts, 160
- nmea.h, 161
 - NMEA_BUFFER_SIZE, 161
 - nmea_checksum, 161
 - nmea_puts, 161
- NMEA_BUFFER_SIZE
 - nmea.h, 161
- nmea_checksum
 - nmea.c, 160
 - nmea.h, 161
- nmea_puts
 - nmea.c, 160
 - nmea.h, 161
- node
 - rf_probe_t, 49
- note_clock
 - squeaker_private_t, 58
 - tweeter_private_t, 64
- note_duty
 - squeaker_private_t, 58
 - tweeter_private_t, 64
- note_fraction
 - mmelody_private_t, 28
 - squeaker_private_t, 58
- note_holdoff
 - squeaker_private_t, 58
 - tweeter_private_t, 64
- note_period
 - squeaker_private_t, 58
 - tweeter_private_t, 64
- nrf2401.c, 162
 - CHAR_BIT, 163
 - nrf_address1_set, 165
 - nrf_address2_set, 165
 - nrf_address_length_set, 165
 - NRF_CE_HIGH_SET, 163
 - NRF_CE_LOW_SET, 163
 - NRF_CHANNEL_NUMBER_MAX, 163
 - nrf_channel_set, 165
 - nrf_comms_mode_set, 165
 - NRF_CONFIG_DELAY_US, 163
 - NRF_CONFIGURATION_REGISTER_SIZE, 163
 - nrf_configure, 165
 - nrf_crc_length_set, 165
 - nrf_crc_status_set, 165
 - NRF_CS_HIGH_SET, 163
 - NRF_CS_LOW_SET, 163
 - nrf_data_rate_set, 165
 - NRF_DATA_READY_P, 163
 - nrf_data_ready_p, 165
 - NRF_FULL_CONFIGURE, 163
 - nrf_init, 165
 - NRF_LINE_TIME_ENABLE_US, 164
 - nrf_payload_length1_get, 165
 - nrf_payload_length1_set, 165
 - nrf_payload_length2_set, 165
 - nrf_receive, 165
 - nrf_rf_dir_set, 165
 - nrf_rf_enable, 165
 - nrf_rf_power_set, 165
 - nrf_rf_standby, 165
 - nrf_setup, 165
 - nrf_single_or_dual_channel_set, 165
 - NRF_T_SB_ACTIVE, 164
 - nrf_transmit, 165
 - nrf_xtal_freq_set, 165
- nrf2401.h, 166
 - NRF_ACK_DELAY_US, 169
 - nrf_address1_set, 170
 - nrf_address2_set, 170
 - nrf_address_length_set, 170
 - nrf_channel_set, 170
 - nrf_comms_mode_set, 170
 - nrf_configure, 170
 - NRF_CRC_16, 168
 - NRF_CRC_8, 168
 - NRF_CRC_DISABLED, 168
 - NRF_CRC_ENABLED, 168
 - nrf_crc_length_set, 170
 - nrf_crc_status_set, 170
 - NRF_DATA_1M, 168
 - NRF_DATA_250K, 168
 - nrf_data_rate_set, 170
 - nrf_data_ready_p, 170
 - NRF_DEFAULT_ADDRESS, 167
 - NRF_DIRECT, 168
 - NRF_DUAL_CHANNEL, 168

- nrf_init, 170
- nrf_payload_length1_get, 170
- nrf_payload_length1_set, 170
- nrf_payload_length2_set, 170
- NRF_PAYLOAD_SIZE, 167
- nrf_receive, 170
- nrf_rf_dir_set, 170
- nrf_rf_enable, 170
- NRF_RF_POWER_0, 168
- NRF_RF_POWER_10, 168
- NRF_RF_POWER_20, 168
- NRF_RF_POWER_5, 168
- nrf_rf_power_set, 170
- nrf_rf_standby, 170
- NRF_RX_MODE, 167
- nrf_setup, 170
- NRF_SHOCKBURST, 168
- NRF_SINGLE_CHANNEL, 168
- nrf_single_or_dual_channel_set, 170
- nrf_t, 167
- NRF_TIME_OUT_ACK_MS, 169
- nrf_transmit, 170
- NRF_TX_MODE, 167
- NRF_XTAL_FREQ_12M, 168
- NRF_XTAL_FREQ_16M, 168
- NRF_XTAL_FREQ_20M, 168
- NRF_XTAL_FREQ_4M, 168
- nrf_xtal_freq_set, 170
- RF_XTAL_FREQ_8M, 168
- NRF_ACK_DELAY_US
 - nrf2401.h, 169
- nrf_address1_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_address2_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_address_length_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_CE_HIGH_SET
 - nrf2401.c, 163
- NRF_CE_LOW_SET
 - nrf2401.c, 163
- nrf_cfg_t, 38
 - ce_bitno, 38
 - ce_port, 38
 - cs_bitno, 38
 - cs_port, 38
 - dr_bitno, 38
 - dr_port, 38
- NRF_CHANNEL_NUMBER_MAX
 - nrf2401.c, 163
- nrf_channel_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_comms_mode_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_config.h, 171
 - rf_cfg_t, 172
 - RF_DATA_READY_P, 172
 - RF_DEVICE_ADDRESS_SET, 172
 - RF_DEVICE_CHANNEL_SET, 172
 - RF_DEVICE_DISABLE, 172
 - RF_DEVICE_ENABLE, 172
 - RF_INIT, 172
 - rf_obj_t, 172
 - RF_RECEIVE, 172
 - RF_RX_MODE_SET, 172
 - RF_SETUP, 172
 - rf_t, 172
 - RF_TRANSMIT, 172
 - RF_TX_MODE_SET, 172
- nrf_config_bits_t, 39
 - addr_1, 40
 - addr_2, 40
 - addr_w, 40
 - cm, 40
 - crc_en, 40
 - crc_l, 40
 - data1_w, 40
 - data2_w, 40
 - rf_pwr, 40
 - rfdr_sb, 40
 - rx2_en, 40
 - rx_ch_num, 40
 - rxen, 40
 - xo_f, 40
- NRF_CONFIG_DELAY_US
 - nrf2401.c, 163
- nrf_config_t, 41
 - bits, 41
 - bytes, 41
- NRF_CONFIGURATION_REGISTER_SIZE
 - nrf2401.c, 163
- nrf_configure
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_CRC_16
 - nrf2401.h, 168
- NRF_CRC_8
 - nrf2401.h, 168
- NRF_CRC_DISABLED
 - nrf2401.h, 168
- NRF_CRC_ENABLED
 - nrf2401.h, 168
- nrf_crc_length_set

- nrf2401.c, 165
- nrf2401.h, 170
- nrf_crc_status_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_CS_HIGH_SET
 - nrf2401.c, 163
- NRF_CS_LOW_SET
 - nrf2401.c, 163
- NRF_DATA_1M
 - nrf2401.h, 168
- NRF_DATA_250K
 - nrf2401.h, 168
- nrf_data_rate_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_DATA_READY_P
 - nrf2401.c, 163
- nrf_data_ready_p
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_DEFAULT_ADDRESS
 - nrf2401.h, 167
- NRF_DIRECT
 - nrf2401.h, 168
- NRF_DUAL_CHANNEL
 - nrf2401.h, 168
- NRF_FULL_CONFIGURE
 - nrf2401.c, 163
- nrf_init
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_LINE_TIME_ENABLE_US
 - nrf2401.c, 164
- nrf_obj_t, 42
 - config, 42
 - pins, 42
- nrf_payload_length1_get
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_payload_length1_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_payload_length2_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_PAYLOAD_SIZE
 - nrf2401.h, 167
- nrf_pins_t, 43
 - ce_bitmask, 43
 - ce_port, 43
 - cs_bitmask, 43
 - cs_port, 43
 - dr_bitmask, 43
 - dr_port, 43
- nrf_receive
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_rf_dir_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_rf_enable
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_RF_POWER_0
 - nrf2401.h, 168
- NRF_RF_POWER_10
 - nrf2401.h, 168
- NRF_RF_POWER_20
 - nrf2401.h, 168
- NRF_RF_POWER_5
 - nrf2401.h, 168
- nrf_rf_power_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_rf_standby
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_RX_MODE
 - nrf2401.h, 167
- nrf_setup
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_SHOCKBURST
 - nrf2401.h, 168
- NRF_SINGLE_CHANNEL
 - nrf2401.h, 168
- nrf_single_or_dual_channel_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- nrf_t
 - nrf2401.h, 167
- NRF_T_SB_ACTIVE
 - nrf2401.c, 164
- NRF_TIME_OUT_ACK_MS
 - nrf2401.h, 169
- nrf_transmit
 - nrf2401.c, 165
 - nrf2401.h, 170
- NRF_TX_MODE
 - nrf2401.h, 167
- NRF_XTAL_FREQ_12M
 - nrf2401.h, 168
- NRF_XTAL_FREQ_16M
 - nrf2401.h, 168
- NRF_XTAL_FREQ_20M
 - nrf2401.h, 168
- NRF_XTAL_FREQ_4M

- nrf2401.h, 168
- nrf_xtal_freq_set
 - nrf2401.c, 165
 - nrf2401.h, 170
- num
 - buttons_private_t, 13
- num_channels
 - mpwm_obj_t, 30
- octave
 - mmelody_private_t, 28
 - squeaker_private_t, 58
- off_pattern
 - chaser_private_t, 14
- offset
 - font_t, 19
- on_pattern
 - chaser_private_t, 14
- OTHER_INT
 - time.h, 225
- out
 - ring_struct, 50
- pattern
 - flasher_private_t, 18
 - sflash_obj_t, 53
- period
 - flasher_pattern_t, 17
 - mpwm_obj_t, 30
 - spwm_obj_t, 56
 - ticker16_t, 60
 - ticker8_t, 61
 - ticker_t, 62
- pga.c, 173
 - PGA_CHANNEL_REGISTER, 174
 - pga_channel_set, 175
 - pga_chip_deselect, 175
 - pga_chip_select, 175
 - PGA_DESELECT, 174
 - PGA_GAIN_REGISTER, 174
 - pga_gain_set, 175
 - PGA_GETC, 174
 - pga_init, 175
 - PGA_INSN_CHANNEL_REGISTER_WRITE, 174
 - PGA_INSN_GAIN_REGISTER_WRITE, 174
 - PGA_INSN_NOP, 174
 - PGA_INSN_REGISTER_WRITE, 174
 - PGA_INSN_SHUTDOWN, 174
 - PGA_INSN_WRITE, 174
 - PGA_PUTC, 174
 - PGA_SELECT, 174
 - pga_send_command, 175
 - pga_shutdown, 175
 - pga_startup, 175
 - PGA_TRANSPARENT, 174
- pga.h, 176
 - PGA_CFG, 177
 - PGA_CHANNEL_0, 177
 - PGA_CHANNEL_1, 177
 - PGA_CHANNEL_2, 177
 - PGA_CHANNEL_3, 177
 - PGA_CHANNEL_4, 177
 - PGA_CHANNEL_5, 177
 - PGA_CHANNEL_6, 177
 - PGA_CHANNEL_7, 177
 - pga_channel_set, 178
 - pga_channel_t, 177
 - pga_chip_deselect, 178
 - pga_chip_select, 178
 - PGA_GAIN_1, 177
 - PGA_GAIN_10, 177
 - PGA_GAIN_16, 177
 - PGA_GAIN_2, 177
 - PGA_GAIN_32, 177
 - PGA_GAIN_4, 177
 - PGA_GAIN_5, 177
 - PGA_GAIN_8, 177
 - pga_gain_set, 178
 - pga_gain_t, 177
 - pga_init, 178
 - pga_obj_t, 177
 - pga_shutdown, 178
 - pga_startup, 178
 - pga_t, 177
- PGA_CFG
 - pga.h, 177
- pga_cfg_t, 44
 - cs_bitmask, 44
 - cs_port, 44
- PGA_CHANNEL_0
 - pga.h, 177
- PGA_CHANNEL_1
 - pga.h, 177
- PGA_CHANNEL_2
 - pga.h, 177
- PGA_CHANNEL_3
 - pga.h, 177
- PGA_CHANNEL_4
 - pga.h, 177
- PGA_CHANNEL_5
 - pga.h, 177
- PGA_CHANNEL_6
 - pga.h, 177
- PGA_CHANNEL_7
 - pga.h, 177
- PGA_CHANNEL_REGISTER
 - pga.c, 174

- pga_channel_set
 - pga.c, 175
 - pga.h, 178
- pga_channel_t
 - pga.h, 177
- pga_chip_deselect
 - pga.c, 175
 - pga.h, 178
- pga_chip_select
 - pga.c, 175
 - pga.h, 178
- PGA_DESELECT
 - pga.c, 174
- PGA_GAIN_1
 - pga.h, 177
- PGA_GAIN_10
 - pga.h, 177
- PGA_GAIN_16
 - pga.h, 177
- PGA_GAIN_2
 - pga.h, 177
- PGA_GAIN_32
 - pga.h, 177
- PGA_GAIN_4
 - pga.h, 177
- PGA_GAIN_5
 - pga.h, 177
- PGA_GAIN_8
 - pga.h, 177
- PGA_GAIN_REGISTER
 - pga.c, 174
- pga_gain_set
 - pga.c, 175
 - pga.h, 178
- pga_gain_t
 - pga.h, 177
- PGA_GETC
 - pga.c, 174
- pga_init
 - pga.c, 175
 - pga.h, 178
- PGA_INSN_CHANNEL_REGISTER_WRITE
 - pga.c, 174
- PGA_INSN_GAIN_REGISTER_WRITE
 - pga.c, 174
- PGA_INSN_NOP
 - pga.c, 174
- PGA_INSN_REGISTER_WRITE
 - pga.c, 174
- PGA_INSN_SHUTDOWN
 - pga.c, 174
- PGA_INSN_WRITE
 - pga.c, 174
- pga_obj_t
 - pga.h, 177
- pga_private_t, 45
 - cs_bitmask, 45
 - cs_port, 45
- PGA_PUTC
 - pga.c, 174
- PGA_SELECT
 - pga.c, 174
- pga_send_command
 - pga.c, 175
- pga_shutdown
 - pga.c, 175
 - pga.h, 178
- pga_startup
 - pga.c, 175
 - pga.h, 178
- pga_t
 - pga.h, 177
- PGA_TRANSPARENT
 - pga.c, 174
- piezo.c, 179
 - piezo_init, 179
- piezo.h, 180
 - PIEZO_CFG, 180
 - piezo_init, 180
 - piezo_obj_t, 180
 - piezo_set, 180
 - piezo_t, 180
- piezo_beep
 - piezo_beep.c, 181
 - piezo_beep.h, 182
- piezo_beep.c, 181
 - piezo_beep, 181
 - piezo_beep_long, 181
 - PIEZO_BEEP_PERIOD, 181
 - piezo_beep_short, 181
 - PIEZO_LONG_BEEP_TIME, 181
 - PIEZO_SHORT_BEEP_TIME, 181
- piezo_beep.h, 182
 - piezo_beep, 182
 - piezo_beep_long, 182
 - piezo_beep_short, 182
- piezo_beep_long
 - piezo_beep.c, 181
 - piezo_beep.h, 182
- PIEZO_BEEP_PERIOD
 - piezo_beep.c, 181
- piezo_beep_short
 - piezo_beep.c, 181
 - piezo_beep.h, 182
- PIEZO_CFG
 - piezo.h, 180
- piezo_cfg_t, 46
 - bitmask, 46

- port, 46
- piezo_init
 - piezo.c, 179
 - piezo.h, 180
- PIEZO_LONG_BEEP_TIME
 - piezo_beep.c, 181
- piezo_obj_t
 - piezo.h, 180
- piezo_set
 - piezo.h, 180
- PIEZO_SHORT_BEEP_TIME
 - piezo_beep.c, 181
- piezo_t
 - piezo.h, 180
- pins
 - nrf_obj_t, 42
- pixels
 - font_t, 19
 - mtext_obj_t, 31
- play_callback
 - mmelody_private_t, 28
- play_callback_data
 - mmelody_private_t, 28
- poll_rate
 - mmelody_private_t, 28
 - mtext_obj_t, 31
 - squeaker_private_t, 58
 - tweeter_private_t, 64
- port
 - button_cfg_t, 11
 - led_cfg_t, 22
 - lmatrix_port_t, 23
 - muxleds_cfg_t, 34
 - muxleds_col_t, 35
 - muxleds_row_t, 37
 - piezo_cfg_t, 46
- prescaler
 - squeaker_private_t, 58
- primary_ticker
 - mcleds_private_t, 26
- putc
 - usart_dev_struct, 69
- read_ready_p
 - usart_dev_struct, 69
- rf.c, 183
 - HIGH_BYTE, 184
 - LOW_BYTE, 184
 - rf_acknowledge, 184
 - rf_acknowledge_wait, 184
 - rf_address_make, 184
 - rf_command, 184
 - rf_command_no_ack, 184
 - RF_DEBUG, 184
 - rf_init, 184
 - rf_node_make, 184
 - rf_read, 184
 - rf_read_enable, 184
 - rf_read_ready_p, 184
 - rf_read_setup, 184
 - rf_receive, 184
 - rf_setup, 184
 - rf_standby, 184
 - rf_transmit, 184
 - rf_write, 184
 - rf_write_data, 184
 - rf_write_setup, 184
- rf.h, 185
 - rf_acknowledge, 188
 - rf_acknowledge_wait, 188
 - rf_address_calc, 188
 - RF_BROADCAST_CHANNEL, 186
 - RF_BROADCAST_SLAVE_ID, 187
 - rf_channel_t, 186
 - RF_CMD_ACK, 187
 - RF_CMD_ALL_PACKETS_RECEIVED, 187
 - RF_CMD_BROADCAST_EXIT, 187
 - RF_CMD_BROADCAST_MODE, 187
 - RF_CMD_CHANNEL_SET, 187
 - RF_CMD_DATA_START, 187
 - RF_CMD_DEVICE_ID_GET, 187
 - RF_CMD_DEVICE_ID_RESPONSE, 187
 - RF_CMD_ENUMERATE_DEVICE, 187
 - RF_CMD_RESEND_PACKET, 187
 - RF_CMD_SIZE, 187
 - rf_cmd_t, 187
 - RF_CMD_TIMESTAMP_REQ, 187
 - rf_command, 188
 - rf_command_no_ack, 188
 - RF_DEVICE_ID_SIZE, 187
 - rf_enumeration_response, 188
 - rf_id_t, 187
 - rf_init, 188
 - RF_MASTER_ID, 187
 - rf_node_make, 188
 - RF_PAYLOAD_SIZE, 186
 - RF_PROBE_NOT_FOUND, 186
 - rf_probes_enumerate, 188
 - rf_read, 188
 - rf_read_data, 188
 - rf_read_enable, 188
 - rf_read_ready_p, 188
 - rf_read_setup, 188
 - RF_READ_WAIT_MS, 186
 - RF_READY_WAIT2_US, 187
 - RF_READY_WAIT_US, 187
 - RF_RETRIES_MAX, 186
 - rf_size_t, 186

- rf_standby, 188
- rf_transmit, 188
- rf_write, 188
- rf_write_data, 188
- rf_write_setup, 188
- RF_WRITE_WAIT_MS, 187
- rf_acknowledge
 - rf.c, 184
 - rf.h, 188
- rf_acknowledge_wait
 - rf.c, 184
 - rf.h, 188
- rf_address_calc
 - rf.h, 188
- rf_address_make
 - rf.c, 184
- rf_address_t, 47
 - bytes, 47
- RF_BROADCAST_CHANNEL
 - rf.h, 186
- RF_BROADCAST_SLAVE_ID
 - rf.h, 187
- rf_cfg_t
 - nrf_config.h, 172
- rf_channel_t
 - rf.h, 186
- RF_CMD_ACK
 - rf.h, 187
- RF_CMD_ALL_PACKETS_RECEIVED
 - rf.h, 187
- RF_CMD_BROADCAST_EXIT
 - rf.h, 187
- RF_CMD_BROADCAST_MODE
 - rf.h, 187
- RF_CMD_CHANNEL_SET
 - rf.h, 187
- RF_CMD_DATA_START
 - rf.h, 187
- RF_CMD_DEVICE_ID_GET
 - rf.h, 187
- RF_CMD_DEVICE_ID_RESPONSE
 - rf.h, 187
- RF_CMD_ENUMERATE_DEVICE
 - rf.h, 187
- RF_CMD_RESEND_PACKET
 - rf.h, 187
- RF_CMD_SIZE
 - rf.h, 187
- rf_cmd_t
 - rf.h, 187
- RF_CMD_TIMESTAMP_REQ
 - rf.h, 187
- rf_command
 - rf.c, 184
- rf.h, 188
- rf_command_no_ack
 - rf.c, 184
 - rf.h, 188
- RF_DATA_READY_P
 - nrf_config.h, 172
- RF_DEBUG
 - rf.c, 184
- RF_DEVICE_ADDRESS_SET
 - nrf_config.h, 172
- RF_DEVICE_CHANNEL_SET
 - nrf_config.h, 172
- RF_DEVICE_DISABLE
 - nrf_config.h, 172
- RF_DEVICE_ENABLE
 - nrf_config.h, 172
- RF_DEVICE_ID_SIZE
 - rf.h, 187
- RF_ENUMERATE_REPEAT_NUM
 - rf_master.c, 190
- rf_enumeration_response
 - rf.h, 188
 - rf_slave.c, 191
- rf_id_t
 - rf.h, 187
- RF_INIT
 - nrf_config.h, 172
- rf_init
 - rf.c, 184
 - rf.h, 188
- rf_master.c, 189
 - RF_ENUMERATE_REPEAT_NUM, 190
 - RF_PROBE_RESPONSE_WAIT_MS, 190
 - RF_PROBE_RESPONSE_WAIT_NEXT_MS, 190
 - rf_probes_enumerate, 190
 - rf_probes_search, 190
 - rf_read_data, 190
 - RF_UNACK_CMD_REPEAT_DELAY_MS, 190
 - RF_UNACK_CMD_REPEAT_NUM, 190
- RF_MASTER_ID
 - rf.h, 187
- rf_node_make
 - rf.c, 184
 - rf.h, 188
- rf_node_t, 48
 - channel, 48
 - id, 48
- rf_obj_t
 - nrf_config.h, 172
- RF_PAYLOAD_SIZE
 - rf.h, 186
- RF_PROBE_ENUMERATE_WAIT_MS

- rf_slave.c, [191](#)
- RF_PROBE_NOT_FOUND
 - rf.h, [186](#)
- RF_PROBE_RESPONSE_WAIT_MS
 - rf_master.c, [190](#)
- RF_PROBE_RESPONSE_WAIT_NEXT_MS
 - rf_master.c, [190](#)
- rf_probe_t, [49](#)
 - device_id, [49](#)
 - node, [49](#)
- rf_probes_enumerate
 - rf.h, [188](#)
 - rf_master.c, [190](#)
- rf_probes_search
 - rf_master.c, [190](#)
- rf_pwr
 - nrf_config_bits_t, [40](#)
- rf_read
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_read_data
 - rf.h, [188](#)
 - rf_master.c, [190](#)
- rf_read_enable
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_read_ready_p
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_read_setup
 - rf.c, [184](#)
 - rf.h, [188](#)
- RF_READ_WAIT_MS
 - rf.h, [186](#)
- RF_READY_WAIT2_US
 - rf.h, [187](#)
- RF_READY_WAIT_US
 - rf.h, [187](#)
- RF_RECEIVE
 - nrf_config.h, [172](#)
- rf_receive
 - rf.c, [184](#)
- RF_RETRIES_MAX
 - rf.h, [186](#)
- RF_RX_MODE_SET
 - nrf_config.h, [172](#)
- RF_SETUP
 - nrf_config.h, [172](#)
- rf_setup
 - rf.c, [184](#)
- rf_size_t
 - rf.h, [186](#)
- rf_slave.c
 - RF_PROBE_ENUMERATE_WAIT_MS, [191](#)
- rf_slave.c, [191](#)
 - rf_enumeration_response, [191](#)
- rf_standby
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_t
 - nrf_config.h, [172](#)
- RF_TRANSMIT
 - nrf_config.h, [172](#)
- rf_transmit
 - rf.c, [184](#)
 - rf.h, [188](#)
- RF_TX_MODE_SET
 - nrf_config.h, [172](#)
- RF_UNACK_CMD_REPEAT_DELAY_MS
 - rf_master.c, [190](#)
- RF_UNACK_CMD_REPEAT_NUM
 - rf_master.c, [190](#)
- rf_write
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_write_data
 - rf.c, [184](#)
 - rf.h, [188](#)
- rf_write_setup
 - rf.c, [184](#)
 - rf.h, [188](#)
- RF_WRITE_WAIT_MS
 - rf.h, [187](#)
- RF_XTAL_FREQ_8M
 - nrf2401.h, [168](#)
- rfdr_sb
 - nrf_config_bits_t, [40](#)
- ring.c, [192](#)
 - RING_EMPTY_P, [192](#)
 - ring_empty_p, [193](#)
 - RING_FULL_P, [192](#)
 - ring_init, [193](#)
 - ring_read, [193](#)
 - RING_READ_NUM, [192](#)
 - ring_read_num, [193](#)
 - RING_SIZE, [192](#)
 - ring_write, [193](#)
 - RING_WRITE_NUM, [193](#)
 - ring_write_num, [193](#)
- ring.h, [194](#)
 - ring_empty_p, [195](#)
 - ring_init, [195](#)
 - ring_read, [195](#)
 - ring_read_num, [195](#)
 - ring_size_t, [195](#)
 - ring_t, [195](#)
 - ring_write, [195](#)
 - ring_write_num, [195](#)

- RING_EMPTY_P
 - ring.c, [192](#)
- ring_empty_p
 - ring.c, [193](#)
 - ring.h, [195](#)
- RING_FULL_P
 - ring.c, [192](#)
- ring_init
 - ring.c, [193](#)
 - ring.h, [195](#)
- ring_read
 - ring.c, [193](#)
 - ring.h, [195](#)
- RING_READ_NUM
 - ring.c, [192](#)
- ring_read_num
 - ring.c, [193](#)
 - ring.h, [195](#)
- RING_SIZE
 - ring.c, [192](#)
- ring_size_t
 - ring.h, [195](#)
- ring_struct, [50](#)
 - end, [50](#)
 - in, [50](#)
 - out, [50](#)
 - top, [50](#)
- ring_t
 - ring.h, [195](#)
- ring_write
 - ring.c, [193](#)
 - ring.h, [195](#)
- RING_WRITE_NUM
 - ring.c, [193](#)
- ring_write_num
 - ring.c, [193](#)
 - ring.h, [195](#)
- rom_code
 - mulwire_obj_t, [32](#)
 - ulwire_obj_t, [66](#)
- row
 - cleds_private_t, [15](#)
- row_config
 - cleds_private_t, [15](#)
 - mbuttons_private_t, [25](#)
- row_on
 - muxleds_obj_t, [36](#)
- row_state
 - muxleds_col_t, [35](#)
- ROWBIT
 - lmatrix.c, [135](#)
- rows
 - muxleds_obj_t, [36](#)
 - scroller_obj_t, [51](#)
- rows_num
 - cleds_private_t, [15](#)
 - mbuttons_private_t, [25](#)
 - muxleds_obj_t, [36](#)
- rs_bit
 - lcd_cfg_t, [20](#)
- rs_mask
 - lcd_obj_t, [21](#)
- rs_port
 - lcd_cfg_t, [20](#)
- running
 - scroller_obj_t, [51](#)
- rx2_en
 - nrf_config_bits_t, [40](#)
- rx_ch_num
 - nrf_config_bits_t, [40](#)
- rx_irq_enable
 - busart_dev_struct, [10](#)
- rx_ring
 - busart_dev_struct, [10](#)
- rxen
 - nrf_config_bits_t, [40](#)
- s_eeprom.c
 - SPI_EEPROM_BP0, [197](#)
 - SPI_EEPROM_BP1, [197](#)
 - SPI_EEPROM_OP_RDSR, [197](#)
 - SPI_EEPROM_OP_READ, [197](#)
 - SPI_EEPROM_OP_WRDI, [197](#)
 - SPI_EEPROM_OP_WREN, [197](#)
 - SPI_EEPROM_OP_WRITE, [197](#)
 - SPI_EEPROM_OP_WRSR, [197](#)
 - SPI_EEPROM_WEL, [197](#)
 - SPI_EEPROM_WIP, [197](#)
 - SPI_EEPROM_WPEN, [197](#)
- s_eeprom.c, [196](#)
 - SPI_EEPROM_DISABLE, [197](#)
 - spi_eeprom_disable, [198](#)
 - SPI_EEPROM_ENABLE, [197](#)
 - SPI_EEPROM_GETC, [197](#)
 - spi_eeprom_init, [198](#)
 - SPI_EEPROM_OP, [197](#)
 - SPI_EEPROM_PUTC, [197](#)
 - spi_eeprom_read, [198](#)
 - spi_eeprom_status_write, [198](#)
 - SPI_EEPROM_TRANSPARENT, [197](#)
 - spi_eeprom_write, [198](#)
 - spi_eeprom_write_setup, [198](#)
- s_eeprom.h, [199](#)
 - spi_eeprom_addr_t, [200](#)
 - spi_eeprom_disable, [200](#)
 - spi_eeprom_init, [200](#)
 - spi_eeprom_obj_t, [200](#)
 - spi_eeprom_read, [200](#)

- [spi_eeprom_size_t](#), [200](#)
 - [spi_eeprom_t](#), [200](#)
 - [spi_eeprom_write](#), [200](#)
 - [spi_eeprom_write_setup](#), [200](#)
- [scale_table](#)
 - [squeaker_private_t](#), [58](#)
 - [tweeter_private_t](#), [64](#)
- [screen](#)
 - [mtext_obj_t](#), [31](#)
- [scroller](#)
 - [mtext_obj_t](#), [31](#)
- [scroller.c](#), [201](#)
 - [scroller_init](#), [201](#)
 - [scroller_speed_scale_get](#), [201](#)
 - [scroller_start](#), [201](#)
 - [scroller_update](#), [201](#)
- [scroller.h](#), [202](#)
 - [scroller_dir_get](#), [203](#)
 - [scroller_dir_set](#), [203](#)
 - [scroller_dir_t](#), [202](#)
 - [SCROLLER_DOWN](#), [203](#)
 - [scroller_init](#), [203](#)
 - [SCROLLER_LEFT](#), [202](#)
 - [SCROLLER_OFF](#), [202](#)
 - [SCROLLER_RIGHT](#), [203](#)
 - [scroller_speed_scale_get](#), [203](#)
 - [scroller_start](#), [203](#)
 - [scroller_stop](#), [203](#)
 - [scroller_t](#), [202](#)
 - [SCROLLER_UP](#), [203](#)
 - [scroller_update](#), [203](#)
- [scroller_dir_get](#)
 - [scroller.h](#), [203](#)
- [scroller_dir_set](#)
 - [scroller.h](#), [203](#)
- [scroller_dir_t](#)
 - [scroller.h](#), [202](#)
- [SCROLLER_DOWN](#)
 - [scroller.h](#), [203](#)
- [scroller_init](#)
 - [scroller.c](#), [201](#)
 - [scroller.h](#), [203](#)
- [SCROLLER_LEFT](#)
 - [scroller.h](#), [202](#)
- [scroller_obj_t](#), [51](#)
 - [cols](#), [51](#)
 - [dir](#), [51](#)
 - [index](#), [51](#)
 - [rows](#), [51](#)
 - [running](#), [51](#)
- [SCROLLER_OFF](#)
 - [scroller.h](#), [202](#)
- [SCROLLER_RIGHT](#)
 - [scroller.h](#), [203](#)
- [scroller_speed_scale_get](#)
 - [scroller.c](#), [201](#)
 - [scroller.h](#), [203](#)
- [scroller_start](#)
 - [scroller.c](#), [201](#)
 - [scroller.h](#), [203](#)
- [scroller_stop](#)
 - [scroller.h](#), [203](#)
- [scroller_t](#)
 - [scroller.h](#), [202](#)
- [SCROLLER_UP](#)
 - [scroller.h](#), [203](#)
- [scroller_update](#)
 - [scroller.c](#), [201](#)
 - [scroller.h](#), [203](#)
- [seq](#)
 - [chaser_private_t](#), [14](#)
 - [stext_obj_t](#), [59](#)
- [seq.c](#), [204](#)
 - [seq_init](#), [204](#)
 - [seq_update](#), [204](#)
- [seq.h](#), [205](#)
 - [seq_get](#), [205](#)
 - [seq_init](#), [205](#)
 - [seq_set](#), [205](#)
 - [seq_t](#), [205](#)
 - [seq_update](#), [205](#)
- [seq_get](#)
 - [seq.h](#), [205](#)
- [seq_info](#)
 - [stext_obj_t](#), [59](#)
- [seq_init](#)
 - [seq.c](#), [204](#)
 - [seq.h](#), [205](#)
- [seq_obj_t](#), [52](#)
 - [callback](#), [52](#)
 - [callback_data](#), [52](#)
 - [cur](#), [52](#)
 - [str](#), [52](#)
- [seq_set](#)
 - [seq.h](#), [205](#)
- [seq_t](#)
 - [seq.h](#), [205](#)
- [seq_update](#)
 - [seq.c](#), [204](#)
 - [seq.h](#), [205](#)
- [serial](#)
 - [mulwire_rom_code_t](#), [33](#)
 - [u1wire_rom_code_t](#), [67](#)
- [sflash.c](#), [206](#)
 - [sflash_update](#), [206](#)
- [sflash.h](#), [207](#)
 - [sflash_pattern_get](#), [207](#)
 - [sflash_pattern_set](#), [207](#)

- sflash_pattern_t, 207
- sflash_t, 207
- sflash_update, 207
- sflash_obj_t, 53
 - current, 53
 - pattern, 53
- sflash_pattern_get
 - sflash.h, 207
- sflash_pattern_set
 - sflash.h, 207
- sflash_pattern_t
 - sflash.h, 207
- sflash_t
 - sflash.h, 207
- sflash_update
 - sflash.c, 206
 - sflash.h, 207
- size
 - font_t, 19
- speed
 - mmelody_private_t, 28
 - mtext_obj_t, 31
 - squeaker_private_t, 58
- spi_adc.h
 - SPI_ADC_MODE_DIFFERENTIAL, 208
 - SPI_ADC_MODE_DIFFERENTIAL_-INVERTED, 208
 - SPI_ADC_MODE_SINGLE_ENDED, 208
- spi_adc.h, 208
 - spi_adc_mode_t, 208
- SPI_ADC_MODE_DIFFERENTIAL
 - spi_adc.h, 208
- SPI_ADC_MODE_DIFFERENTIAL_INVERTED
 - spi_adc.h, 208
- SPI_ADC_MODE_SINGLE_ENDED
 - spi_adc.h, 208
- spi_adc_mode_t
 - spi_adc.h, 208
- spi_eeprom_addr_t
 - s_eeprom.h, 200
- SPI_EEPROM_BP0
 - s_eeprom.c, 197
- SPI_EEPROM_BP1
 - s_eeprom.c, 197
- spi_eeprom_cfg_t, 54
 - cs_bitno, 54
 - cs_port, 54
- SPI_EEPROM_DISABLE
 - s_eeprom.c, 197
- spi_eeprom_disable
 - s_eeprom.c, 198
 - s_eeprom.h, 200
- SPI_EEPROM_ENABLE
 - s_eeprom.c, 197
- SPI_EEPROM_GETC
 - s_eeprom.c, 197
- spi_eeprom_init
 - s_eeprom.c, 198
 - s_eeprom.h, 200
- spi_eeprom_obj_t
 - s_eeprom.h, 200
- SPI_EEPROM_OP
 - s_eeprom.c, 197
- SPI_EEPROM_OP_RDSR
 - s_eeprom.c, 197
- SPI_EEPROM_OP_READ
 - s_eeprom.c, 197
- SPI_EEPROM_OP_WRDI
 - s_eeprom.c, 197
- SPI_EEPROM_OP_WREN
 - s_eeprom.c, 197
- SPI_EEPROM_OP_WRITE
 - s_eeprom.c, 197
- SPI_EEPROM_OP_WRSR
 - s_eeprom.c, 197
- spi_eeprom_private_t, 55
 - cs_bitmask, 55
 - cs_port, 55
- SPI_EEPROM_PUTC
 - s_eeprom.c, 197
- spi_eeprom_read
 - s_eeprom.c, 198
 - s_eeprom.h, 200
- spi_eeprom_size_t
 - s_eeprom.h, 200
- spi_eeprom_status_write
 - s_eeprom.c, 198
- spi_eeprom_t
 - s_eeprom.h, 200
- SPI_EEPROM_TRANSPARENT
 - s_eeprom.c, 197
- SPI_EEPROM_WEL
 - s_eeprom.c, 197
- SPI_EEPROM_WIP
 - s_eeprom.c, 197
- SPI_EEPROM_WPEN
 - s_eeprom.c, 197
- spi_eeprom_write
 - s_eeprom.c, 198
 - s_eeprom.h, 200
- spi_eeprom_write_setup
 - s_eeprom.c, 198
 - s_eeprom.h, 200
- spwm.c, 209
 - spwm_duty_set, 209
 - spwm_init, 209
 - spwm_period_set, 209
 - spwm_update, 209

- spwm.h, 210
 - spwm_duty_set, 210
 - spwm_init, 210
 - spwm_period_set, 210
 - spwm_t, 210
 - spwm_update, 210
- spwm_duty_set
 - spwm.c, 209
 - spwm.h, 210
- spwm_init
 - spwm.c, 209
 - spwm.h, 210
- spwm_obj_t, 56
 - count, 56
 - duty, 56
 - period, 56
- spwm_period_set
 - spwm.c, 209
 - spwm.h, 210
- spwm_t
 - spwm.h, 210
- spwm_update
 - spwm.c, 209
 - spwm.h, 210
- squeaker.c, 211
 - squeaker_char_to_note, 212
 - SQUEAKER_HOLDOFF_TIME, 212
 - squeaker_init, 212
 - squeaker_note_fraction_set, 212
 - squeaker_note_play, 212
 - squeaker_note_set, 212
 - squeaker_play, 212
 - SQUEAKER_PRESCALER, 212
 - squeaker_rest_play, 212
 - SQUEAKER_SCALE_SIZE, 212
 - squeaker_scan, 212
 - squeaker_speed_set, 212
 - squeaker_ticker_set, 212
 - SQUEAKER_TRANSPARENT, 212
 - squeaker_update, 212
 - squeaker_volume_set, 212
- squeaker.h, 213
 - SQUEAKER_DIVISOR, 214
 - squeaker_duration_t, 214
 - squeaker_init, 215
 - SQUEAKER_NOTE_MIN, 215
 - squeaker_note_t, 214
 - squeaker_obj_t, 214
 - SQUEAKER_OCTAVE_DEFAULT, 214
 - squeaker_period_t, 214
 - squeaker_play, 215
 - squeaker_scale_t, 214
 - SQUEAKER_SCALE_TABLE, 214
 - SQUEAKER_SPEED_DEFAULT, 215
 - squeaker_speed_set, 215
 - squeaker_speed_t, 214
 - squeaker_t, 214
 - squeaker_update, 215
 - squeaker_volume_set, 215
 - squeaker_volume_t, 214
- squeaker2.c, 216
 - squeaker_char_to_note, 217
 - SQUEAKER_HOLDOFF_TIME, 216
 - squeaker_init, 217
 - squeaker_note_fraction_set, 217
 - squeaker_note_play, 217
 - squeaker_note_set, 217
 - squeaker_play, 217
 - SQUEAKER_PRESCALER, 216
 - squeaker_rest_play, 217
 - SQUEAKER_SCALE_SIZE, 216
 - squeaker_scan, 217
 - squeaker_speed_set, 217
 - squeaker_ticker_set, 217
 - SQUEAKER_TRANSPARENT, 216
 - squeaker_update, 217
 - squeaker_volume_set, 217
- squeaker_char_to_note
 - squeaker.c, 212
 - squeaker2.c, 217
- SQUEAKER_DIVISOR
 - squeaker.h, 214
- squeaker_duration_t
 - squeaker.h, 214
- SQUEAKER_HOLDOFF_TIME
 - squeaker.c, 212
 - squeaker2.c, 216
- squeaker_init
 - squeaker.c, 212
 - squeaker.h, 215
 - squeaker2.c, 217
- squeaker_note_fraction_set
 - squeaker.c, 212
 - squeaker2.c, 217
- SQUEAKER_NOTE_MIN
 - squeaker.h, 215
- squeaker_note_play
 - squeaker.c, 212
 - squeaker2.c, 217
- squeaker_note_set
 - squeaker.c, 212
 - squeaker2.c, 217
- squeaker_note_t
 - squeaker.h, 214
- squeaker_obj_t
 - squeaker.h, 214
- SQUEAKER_OCTAVE_DEFAULT
 - squeaker.h, 214

- squeaker_period_t
 - squeaker.h, 214
- squeaker_play
 - squeaker.c, 212
 - squeaker.h, 215
 - squeaker2.c, 217
- SQUEAKER_PRESCALER
 - squeaker.c, 212
 - squeaker2.c, 216
- squeaker_private_t, 57
 - cur, 58
 - holdoff, 58
 - loop_count, 58
 - loop_start, 58
 - note_clock, 58
 - note_duty, 58
 - note_fraction, 58
 - note_holdoff, 58
 - note_period, 58
 - octave, 58
 - poll_rate, 58
 - prescaler, 58
 - scale_table, 58
 - speed, 58
 - start, 58
 - ticker, 58
 - volume, 58
- squeaker_rest_play
 - squeaker.c, 212
 - squeaker2.c, 217
- SQUEAKER_SCALE_SIZE
 - squeaker.c, 212
 - squeaker2.c, 216
- squeaker_scale_t
 - squeaker.h, 214
- SQUEAKER_SCALE_TABLE
 - squeaker.h, 214
- squeaker_scan
 - squeaker.c, 212
 - squeaker2.c, 217
- SQUEAKER_SPEED_DEFAULT
 - squeaker.h, 215
- squeaker_speed_set
 - squeaker.c, 212
 - squeaker.h, 215
 - squeaker2.c, 217
- squeaker_speed_t
 - squeaker.h, 214
- squeaker_t
 - squeaker.h, 214
- squeaker_ticker_set
 - squeaker.c, 212
 - squeaker2.c, 217
- SQUEAKER_TRANSPARENT
 - squeaker.c, 212
 - squeaker2.c, 216
- squeaker_update
 - squeaker.c, 212
 - squeaker.h, 215
 - squeaker2.c, 217
- squeaker_volume_set
 - squeaker.c, 212
 - squeaker.h, 215
 - squeaker2.c, 217
- squeaker_volume_t
 - squeaker.h, 214
- start
 - mmelody_private_t, 28
 - mtext_obj_t, 31
 - squeaker_private_t, 58
- state
 - button_private_t, 12
 - lmatrix_private_t, 24
 - mcleds_private_t, 26
 - ulwire_enumerate_t, 65
- step
 - biseq_obj_t, 9
 - chaser_private_t, 14
- stext.c, 218
 - stext_display, 218
 - stext_init, 218
- stext.h, 219
 - stext_get, 220
 - stext_init, 220
 - STEXT_MODE_CYCLE, 219
 - STEXT_MODE_NORMAL, 219
 - STEXT_MODE_NUM, 220
 - stext_mode_t, 219
 - stext_set, 220
 - stext_t, 219
 - stext_update, 220
- stext_display
 - stext.c, 218
- stext_get
 - stext.h, 220
- stext_init
 - stext.c, 218
 - stext.h, 220
- STEXT_MODE_CYCLE
 - stext.h, 219
- STEXT_MODE_NORMAL
 - stext.h, 219
- STEXT_MODE_NUM
 - stext.h, 220
- stext_mode_t
 - stext.h, 219
- stext_obj_t, 59
 - callback, 59

- callback_data, 59
- font, 59
- seq, 59
- seq_info, 59
- stext_set
 - stext.h, 220
- stext_t
 - stext.h, 219
- stext_update
 - stext.h, 220
- str
 - biseq_obj_t, 9
 - seq_obj_t, 52
- ticker
 - mmelody_private_t, 28
 - mtext_obj_t, 31
 - squeaker_private_t, 58
- ticker.c, 221
- ticker.h, 222
 - TICKER_INIT, 222
 - TICKER_START, 222
 - TICKER_UPDATE, 222
- ticker16_t, 60
 - clock, 60
 - period, 60
- ticker8_t, 61
 - clock, 61
 - period, 61
- TICKER_INIT
 - ticker.h, 222
- TICKER_START
 - ticker.h, 222
- ticker_t, 62
 - clock, 62
 - period, 62
- TICKER_UPDATE
 - ticker.h, 222
- time, 63
 - ms_ticks, 63
 - us_ticks, 63
- time.c, 223
 - ms_ticks, 223
 - time_current_time_get, 223
 - time_delay_us, 223
 - time_init, 223
 - time_irq_disable, 223
 - time_irq_enable, 223
 - time_rf_timestamp_get, 223
 - time_time2int, 223
- time.h, 224
 - COMMS_INT, 225
 - DELAY_MAX, 225
 - DELAY_MIN, 225
 - delay_ret_t, 225
 - ERRD, 225
 - OTHER_INT, 225
 - time_current_time_get, 225
 - time_delay_us, 225
 - time_init, 225
 - time_irq_disable, 225
 - time_irq_enable, 225
 - time_rf_timestamp_get, 225
 - time_t, 225
 - time_time2int, 225
 - TIMEOUT, 225
 - TOPCNT, 225
 - USTICKS, 225
- time_current_time_get
 - time.c, 223
 - time.h, 225
- time_delay_us
 - time.c, 223
 - time.h, 225
- time_init
 - time.c, 223
 - time.h, 225
- time_irq_disable
 - time.c, 223
 - time.h, 225
- time_irq_enable
 - time.c, 223
 - time.h, 225
- time_rf_timestamp_get
 - time.c, 223
 - time.h, 225
- time_t
 - time.h, 225
- time_time2int
 - time.c, 223
 - time.h, 225
- TIMEOUT
 - time.h, 225
- top
 - ring_struct, 50
- TOPCNT
 - time.h, 225
- tweeter.c, 226
 - TWEETER_HOLDOFF_TIME, 226
 - tweeter_init, 227
 - tweeter_note_play, 227
 - tweeter_note_set, 227
 - TWEETER_SCALE_SIZE, 226
 - TWEETER_TRANSPARENT, 226
 - tweeter_update, 227
- tweeter.h, 228
 - TWEETER_DIVISOR, 229
 - tweeter_duration_t, 229

- tweeteer_init, 229
 - TWEETER_NOTE_MIN, 229
 - tweeteer_note_play, 229
 - tweeteer_note_t, 229
 - tweeteer_obj_t, 229
 - tweeteer_period_t, 229
 - tweeteer_scale_t, 229
 - TWEETER_SCALE_TABLE, 229
 - tweeteer_t, 229
 - tweeteer_update, 229
 - tweeteer_velocity_t, 229
- TWEETER_DIVISOR
 - tweeteer.h, 229
- tweeteer_duration_t
 - tweeteer.h, 229
- TWEETER_HOLDOFF_TIME
 - tweeteer.c, 226
- tweeteer_init
 - tweeteer.c, 227
 - tweeteer.h, 229
- TWEETER_NOTE_MIN
 - tweeteer.h, 229
- tweeteer_note_play
 - tweeteer.c, 227
 - tweeteer.h, 229
- tweeteer_note_set
 - tweeteer.c, 227
- tweeteer_note_t
 - tweeteer.h, 229
- tweeteer_obj_t
 - tweeteer.h, 229
- tweeteer_period_t
 - tweeteer.h, 229
- tweeteer_private_t, 64
 - note_clock, 64
 - note_duty, 64
 - note_holdoff, 64
 - note_period, 64
 - poll_rate, 64
 - scale_table, 64
- TWEETER_SCALE_SIZE
 - tweeteer.c, 226
- tweeteer_scale_t
 - tweeteer.h, 229
- TWEETER_SCALE_TABLE
 - tweeteer.h, 229
- tweeteer_t
 - tweeteer.h, 229
- TWEETER_TRANSPARENT
 - tweeteer.c, 226
- tweeteer_update
 - tweeteer.c, 227
 - tweeteer.h, 229
- tweeteer_velocity_t
 - tweeteer.h, 229
- tx_finished_p
 - busart_dev_struct, 10
- tx_irq_enable
 - busart_dev_struct, 10
- tx_ring
 - busart_dev_struct, 10
- u1wire.c, 230
 - U1WIRE_ADDR_BYTES, 231
 - u1wire_bit_read, 232
 - u1wire_bit_write, 232
 - u1wire_broadcast, 232
 - u1wire_byte_read, 232
 - u1wire_byte_write, 232
 - u1wire_command, 232
 - U1WIRE_DEBUG, 231
 - U1WIRE_DELAY_OFFSET, 231
 - U1WIRE_DRIVE, 231
 - u1wire_init, 232
 - U1WIRE_MATCH_ROM, 231
 - u1wire_read, 232
 - U1WIRE_READ_ROM, 231
 - u1wire_ready_p, 232
 - U1WIRE_RECALL, 231
 - U1WIRE_RELEASE, 231
 - u1wire_reset, 232
 - u1wire_rom_code_read, 232
 - U1WIRE_SKIP_ROM, 231
 - U1WIRE_TEST, 231
 - u1wire_write, 232
- u1wire.h, 233
 - u1wire_bit_read, 234
 - u1wire_bit_write, 234
 - u1wire_broadcast, 234
 - u1wire_byte_read, 234
 - u1wire_byte_write, 234
 - u1wire_command, 234
 - u1wire_debug, 234
 - U1WIRE_ERR_BUS_HIGH, 234
 - U1WIRE_ERR_BUS_LOW, 234
 - U1WIRE_ERR_BUS_STUCK, 234
 - U1WIRE_ERR_MULTIPLE_DEVICES, 234
 - U1WIRE_ERR_PRESENCE_LONG, 234
 - U1WIRE_ERR_PRESENCE_SHORT, 234
 - u1wire_init, 234
 - u1wire_read, 234
 - u1wire_ready_p, 234
 - u1wire_reset, 234
 - u1wire_t, 234
 - u1wire_write, 234
- U1WIRE_ADDR_BYTES
 - mulwire.c, 154
 - u1wire.c, 231

- ulwire_bit_read
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- ulwire_bit_write
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- ulwire_broadcast
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- ulwire_byte_read
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- ulwire_byte_write
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- ulwire_command
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- U1WIRE_DEBUG
 - mulwire.c, 153
 - ulwire.c, 231
 - ulwire_debug.c, 235
- ulwire_debug
 - ulwire.h, 234
 - ulwire_debug.c, 235
- ulwire_debug.c, 235
 - U1WIRE_DEBUG, 235
 - ulwire_debug, 235
- U1WIRE_DELAY_OFFSET
 - mulwire.c, 154
 - ulwire.c, 231
- ulwire_discover
 - ulwire_discover.c, 236
 - ulwire_discover.h, 237
- ulwire_discover.c
 - U1WIRE_SEARCH, 236
- ulwire_discover.c, 236
 - ulwire_discover, 236
 - ulwire_search, 236
- ulwire_discover.h, 237
 - ulwire_discover, 237
- U1WIRE_DRIVE
 - mulwire.c, 153
 - ulwire.c, 231
- ulwire_enumerate
 - ulwire_enumerate.c, 238
 - ulwire_enumerate.h, 239
- ulwire_enumerate.c
 - U1WIRE_SEARCH, 238
- ulwire_enumerate.c, 238
 - ulwire_enumerate, 238
 - ulwire_enumerate_next, 238
 - ulwire_search, 238
- ulwire_enumerate.h, 239
 - ulwire_enumerate, 239
 - ulwire_enumerate_next, 239
- ulwire_enumerate_next
 - ulwire_enumerate.c, 238
 - ulwire_enumerate.h, 239
- ulwire_enumerate_t, 65
 - device, 65
 - state, 65
- U1WIRE_ERR_BUS_HIGH
 - ulwire.h, 234
- U1WIRE_ERR_BUS_LOW
 - ulwire.h, 234
- U1WIRE_ERR_BUS_STUCK
 - ulwire.h, 234
- U1WIRE_ERR_MULTIPLE_DEVICES
 - ulwire.h, 234
- U1WIRE_ERR_PRESENCE_LONG
 - ulwire.h, 234
- U1WIRE_ERR_PRESENCE_SHORT
 - ulwire.h, 234
- ulwire_init
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- U1WIRE_MATCH_ROM
 - mulwire.c, 154
 - ulwire.c, 231
- ulwire_obj_t, 66
 - rom_code, 66
- ulwire_read
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- U1WIRE_READ_ROM
 - mulwire.c, 154
 - ulwire.c, 231
- ulwire_ready_p
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- U1WIRE_RECALL
 - mulwire.c, 154
 - ulwire.c, 231
- U1WIRE_RELEASE
 - mulwire.c, 153
 - ulwire.c, 231
- ulwire_reset
 - mulwire.c, 154

- ulwire.c, 232
- ulwire.h, 234
- ulwire_rom_code_read
 - mulwire.c, 154
 - ulwire.c, 232
- ulwire_rom_code_t, 67
 - bytes, 67
 - crc, 67
 - family, 67
 - fields, 67
 - serial, 67
- U1WIRE_SEARCH
 - ulwire_discover.c, 236
 - ulwire_enumerate.c, 238
- ulwire_search
 - ulwire_discover.c, 236
 - ulwire_enumerate.c, 238
- U1WIRE_SKIP_ROM
 - mulwire.c, 154
 - ulwire.c, 231
- ulwire_state_t, 68
 - last_device, 68
 - last_discrepancy, 68
 - last_family_discrepancy, 68
- ulwire_t
 - ulwire.h, 234
- U1WIRE_TEST
 - mulwire.c, 154
 - ulwire.c, 231
- ulwire_write
 - mulwire.c, 154
 - ulwire.c, 232
 - ulwire.h, 234
- uint16toa
 - uint16toa.c, 240
 - uint16toa.h, 241
- uint16toa.c, 240
- uint16toa, 240
- uint16toa.h, 241
- uint16toa, 241
- uint8toa
 - uint8toa.c, 242
 - uint8toa.h, 243
- uint8toa.c, 242
- uint8toa, 242
- uint8toa.h, 243
- uint8toa, 243
- us_ticks
 - time, 63
- usart.c, 244
 - USART0_ENABLE, 245
 - USART1_ENABLE, 245
 - usart_getc, 245
 - usart_init, 245
 - usart_putc, 245
 - usart_puts, 245
 - usart_read_ready_p, 245
 - usart_write_finished_p, 245
 - usart_write_ready_p, 245
- usart.h, 246
 - USART_BAUD_DIVISOR, 247
 - usart_dev_t, 247
 - usart_getc, 247
 - usart_init, 247
 - usart_putc, 247
 - usart_puts, 247
 - usart_read_ready_p, 247
 - usart_t, 247
 - usart_write_ready_p, 247
- USART0_ENABLE
 - usart.c, 245
- USART1_ENABLE
 - usart.c, 245
- USART_BAUD_DIVISOR
 - usart.h, 247
- usart_dev_struct, 69
 - getc, 69
 - putc, 69
 - read_ready_p, 69
 - write_finished_p, 69
 - write_ready_p, 69
- usart_dev_t
 - usart.h, 247
- usart_getc
 - usart.c, 245
 - usart.h, 247
- usart_init
 - usart.c, 245
 - usart.h, 247
- usart_putc
 - usart.c, 245
 - usart.h, 247
- usart_puts
 - usart.c, 245
 - usart.h, 247
- usart_read_ready_p
 - usart.c, 245
 - usart.h, 247
- usart_t
 - usart.h, 247
- usart_write_finished_p
 - usart.c, 245
- usart_write_ready_p
 - usart.c, 245
 - usart.h, 247
- USTICKS
 - time.h, 225

volume
 mmelody_private_t, [28](#)
 squeaker_private_t, [58](#)

write_finished_p
 usart_dev_struct, [69](#)

write_ready_p
 usart_dev_struct, [69](#)

xo_f
 nrf_config_bits_t, [40](#)