

# Arduino driver library for Decawave DW1000

Generated by Doxygen 1.8.6

Sun Jul 5 2015 01:18:38



# Contents

- 1 Data Structure Index** **1**
- 1.1 Data Structures 1
  
- 2 File Index** **3**
- 2.1 File List 3
  
- 3 Data Structure Documentation** **5**
- 3.1 DW1000Class Class Reference 5
- 3.1.1 Member Function Documentation 7
- 3.1.1.1 attachReceivedHandler 7
- 3.1.1.2 attachReceiveErrorHandler 7
- 3.1.1.3 attachReceiveTimeoutHandler 7
- 3.1.1.4 attachReceiveTimestampAvailableHandler 7
- 3.1.1.5 attachSentHandler 7
- 3.1.1.6 begin 7
- 3.1.1.7 begin 7
- 3.1.1.8 commitConfiguration 8
- 3.1.1.9 enableMode 8
- 3.1.1.10 end 8
- 3.1.1.11 getData 8
- 3.1.1.12 getData 8
- 3.1.1.13 getDataLength 8
- 3.1.1.14 getPrettyBytes 8
- 3.1.1.15 getPrettyBytes 8
- 3.1.1.16 getPrintableDeviceIdentifier 8
- 3.1.1.17 getPrintableDeviceMode 8
- 3.1.1.18 getPrintableExtendedUniquelIdentifier 8
- 3.1.1.19 getPrintableNetworkIdAndShortAddress 9
- 3.1.1.20 getReceiveTimestamp 10
- 3.1.1.21 getReceiveTimestamp 10
- 3.1.1.22 getSystemTimestamp 10
- 3.1.1.23 getSystemTimestamp 10

3.1.1.24	getTransmitTimestamp	10
3.1.1.25	getTransmitTimestamp	10
3.1.1.26	idle	10
3.1.1.27	interruptOnAutomaticAcknowledgeTrigger	10
3.1.1.28	interruptOnReceived	10
3.1.1.29	interruptOnReceiveError	10
3.1.1.30	interruptOnReceiveTimeout	10
3.1.1.31	interruptOnReceiveTimestampAvailable	10
3.1.1.32	interruptOnSent	10
3.1.1.33	newConfiguration	10
3.1.1.34	newReceive	10
3.1.1.35	newTransmit	10
3.1.1.36	receivePermanently	10
3.1.1.37	reset	10
3.1.1.38	select	10
3.1.1.39	setChannel	11
3.1.1.40	setData	11
3.1.1.41	setData	11
3.1.1.42	setDataRate	11
3.1.1.43	setDefault	11
3.1.1.44	setDelay	11
3.1.1.45	setDeviceAddress	11
3.1.1.46	setInterruptPolarity	11
3.1.1.47	setNetworkId	11
3.1.1.48	setPreambleCode	11
3.1.1.49	setPreambleLength	11
3.1.1.50	setPulseFrequency	12
3.1.1.51	setReceiverAutoReenable	12
3.1.1.52	softReset	12
3.1.1.53	startReceive	12
3.1.1.54	startTransmit	12
3.1.1.55	suppressFrameCheck	12
3.1.1.56	tune	12
3.1.2	Field Documentation	12
3.1.2.1	CHANNEL_1	12
3.1.2.2	CHANNEL_2	12
3.1.2.3	CHANNEL_3	12
3.1.2.4	CHANNEL_4	13
3.1.2.5	CHANNEL_5	13
3.1.2.6	CHANNEL_7	13

3.1.2.7	FRAME_LENGTH_EXTENDED	13
3.1.2.8	FRAME_LENGTH_NORMAL	13
3.1.2.9	MODE_LONGDATA_FAST_ACCURACY	13
3.1.2.10	MODE_LONGDATA_FAST_LOWPOWER	13
3.1.2.11	MODE_LONGDATA_RANGE_ACCURACY	13
3.1.2.12	MODE_LONGDATA_RANGE_LOWPOWER	13
3.1.2.13	MODE_SHORTDATA_FAST_ACCURACY	13
3.1.2.14	MODE_SHORTDATA_FAST_LOWPOWER	13
3.1.2.15	PAC_SIZE_16	13
3.1.2.16	PAC_SIZE_32	13
3.1.2.17	PAC_SIZE_64	13
3.1.2.18	PAC_SIZE_8	13
3.1.2.19	PREAMBLE_CODE_16MHZ_1	13
3.1.2.20	PREAMBLE_CODE_16MHZ_2	13
3.1.2.21	PREAMBLE_CODE_16MHZ_3	13
3.1.2.22	PREAMBLE_CODE_16MHZ_4	13
3.1.2.23	PREAMBLE_CODE_16MHZ_5	13
3.1.2.24	PREAMBLE_CODE_16MHZ_6	13
3.1.2.25	PREAMBLE_CODE_16MHZ_7	13
3.1.2.26	PREAMBLE_CODE_16MHZ_8	13
3.1.2.27	PREAMBLE_CODE_64MHZ_10	13
3.1.2.28	PREAMBLE_CODE_64MHZ_11	13
3.1.2.29	PREAMBLE_CODE_64MHZ_12	14
3.1.2.30	PREAMBLE_CODE_64MHZ_17	14
3.1.2.31	PREAMBLE_CODE_64MHZ_18	14
3.1.2.32	PREAMBLE_CODE_64MHZ_19	14
3.1.2.33	PREAMBLE_CODE_64MHZ_20	14
3.1.2.34	PREAMBLE_CODE_64MHZ_9	14
3.1.2.35	TRX_RATE_110KBPS	14
3.1.2.36	TRX_RATE_6800KBPS	14
3.1.2.37	TRX_RATE_850KBPS	14
3.1.2.38	TX_PREAMBLE_LEN_1024	14
3.1.2.39	TX_PREAMBLE_LEN_128	14
3.1.2.40	TX_PREAMBLE_LEN_1536	14
3.1.2.41	TX_PREAMBLE_LEN_2048	14
3.1.2.42	TX_PREAMBLE_LEN_256	14
3.1.2.43	TX_PREAMBLE_LEN_4096	14
3.1.2.44	TX_PREAMBLE_LEN_512	14
3.1.2.45	TX_PREAMBLE_LEN_64	14
3.1.2.46	TX_PULSE_FREQ_16MHZ	14

3.1.2.47	TX_PULSE_FREQ_64MHZ	14
3.2	DW1000Time Class Reference	14
3.2.1	Constructor & Destructor Documentation	15
3.2.1.1	DW1000Time	15
3.2.1.2	DW1000Time	15
3.2.1.3	DW1000Time	15
3.2.1.4	DW1000Time	15
3.2.1.5	DW1000Time	15
3.2.1.6	DW1000Time	15
3.2.1.7	~DW1000Time	15
3.2.2	Member Function Documentation	15
3.2.2.1	getAsBytes	15
3.2.2.2	getAsFloat	15
3.2.2.3	getAsMeters	16
3.2.2.4	getTimestamp	16
3.2.2.5	getTimestamp	16
3.2.2.6	operator!=	16
3.2.2.7	operator*	16
3.2.2.8	operator*= operator+=	16
3.2.2.9	operator+	16
3.2.2.10	operator+=	16
3.2.2.11	operator-	16
3.2.2.12	operator-=	16
3.2.2.13	operator/	16
3.2.2.14	operator/=	16
3.2.2.15	operator=	16
3.2.2.16	operator==	16
3.2.2.17	setTime	16
3.2.2.18	setTime	16
3.2.2.19	setTimestamp	16
3.2.2.20	setTimestamp	16
3.2.3	Field Documentation	16
3.2.3.1	MICROSECONDS	16
3.2.3.2	MILLISECONDS	16
3.2.3.3	NANOSECONDS	16
3.2.3.4	SECONDS	16
3.2.3.5	TIME_OVERFLOW	16
<b>4</b>	<b>File Documentation</b>	<b>17</b>
4.1	/home/thomas/git/arduino-dw1000/DW1000/DW1000.cpp File Reference	17

4.1.1	Variable Documentation	17
4.1.1.1	DW1000	17
4.2	/home/thomas/git/arduino-dw1000/DW1000/DW1000.h File Reference	17
4.2.1	Macro Definition Documentation	20
4.2.1.1	AAT_BIT	20
4.2.1.2	AGC_TUNE	20
4.2.1.3	AGC_TUNE1_SUB	20
4.2.1.4	AGC_TUNE2_SUB	20
4.2.1.5	AGC_TUNE3_SUB	20
4.2.1.6	CHAN_CTRL	20
4.2.1.7	DEV_ID	20
4.2.1.8	DIS_DRXB_BIT	20
4.2.1.9	DRX_TUNE	20
4.2.1.10	DRX_TUNE0b_SUB	20
4.2.1.11	DRX_TUNE1a_SUB	20
4.2.1.12	DRX_TUNE1b_SUB	20
4.2.1.13	DRX_TUNE2_SUB	20
4.2.1.14	DRX_TUNE4H_SUB	20
4.2.1.15	DX_TIME	20
4.2.1.16	EUI	20
4.2.1.17	FFEN_BIT	20
4.2.1.18	FS_CTRL	20
4.2.1.19	FS_PLLCFG_SUB	20
4.2.1.20	FS_PLLTUNE_SUB	21
4.2.1.21	HIRQ_POL_BIT	21
4.2.1.22	IDLE_MODE	21
4.2.1.23	JUNK	21
4.2.1.24	LDE_CFG1_SUB	21
4.2.1.25	LDE_CFG2_SUB	21
4.2.1.26	LDE_IF	21
4.2.1.27	LDE_REPC_SUB	21
4.2.1.28	LDE_RXANTD_SUB	21
4.2.1.29	LDEDONE_BIT	21
4.2.1.30	LDEERR_BIT	21
4.2.1.31	LEN_AGC_TUNE1	21
4.2.1.32	LEN_AGC_TUNE2	21
4.2.1.33	LEN_AGC_TUNE3	21
4.2.1.34	LEN_CHAN_CTRL	21
4.2.1.35	LEN_DEV_ID	21
4.2.1.36	LEN_DRX_TUNE0b	21

4.2.1.37	LEN_DRX_TUNE1a	21
4.2.1.38	LEN_DRX_TUNE1b	21
4.2.1.39	LEN_DRX_TUNE2	21
4.2.1.40	LEN_DRX_TUNE4H	21
4.2.1.41	LEN_DX_TIME	21
4.2.1.42	LEN_EUI	21
4.2.1.43	LEN_EXT_UWB_FRAMES	21
4.2.1.44	LEN_FS_PLLCFG	21
4.2.1.45	LEN_FS_PLLTUNE	21
4.2.1.46	LEN_LDE_CFG1	21
4.2.1.47	LEN_LDE_CFG2	21
4.2.1.48	LEN_LDE_REPC	22
4.2.1.49	LEN_LDE_RXANTD	22
4.2.1.50	LEN_OTP_ADDR	22
4.2.1.51	LEN_OTP_CTRL	22
4.2.1.52	LEN_OTP_RDAT	22
4.2.1.53	LEN_PANADR	22
4.2.1.54	LEN_PHR_MODE_SUB	22
4.2.1.55	LEN_PMSC_CTRL0	22
4.2.1.56	LEN_RF_RXCTRLH	22
4.2.1.57	LEN_RF_TXCTRL	22
4.2.1.58	LEN_RX_BUFFER	22
4.2.1.59	LEN_RX_FINFO	22
4.2.1.60	LEN_RX_STAMP	22
4.2.1.61	LEN_RX_TIME	22
4.2.1.62	LEN_STAMP	22
4.2.1.63	LEN_SYS_CFG	22
4.2.1.64	LEN_SYS_CTRL	22
4.2.1.65	LEN_SYS_MASK	22
4.2.1.66	LEN_SYS_STATUS	22
4.2.1.67	LEN_SYS_TIME	22
4.2.1.68	LEN_TC_PGDELAY	22
4.2.1.69	LEN_TX_ANTD	22
4.2.1.70	LEN_TX_BUFFER	22
4.2.1.71	LEN_TX_FCTRL	22
4.2.1.72	LEN_TX_POWER	22
4.2.1.73	LEN_TX_STAMP	22
4.2.1.74	LEN_TX_TIME	22
4.2.1.75	LEN_UWB_FRAMES	22
4.2.1.76	NO_SUB	23



4.2.1.77	OTP_ADDR_SUB	23
4.2.1.78	OTP_CTRL_SUB	23
4.2.1.79	OTP_IF	23
4.2.1.80	OTP_RDAT_SUB	23
4.2.1.81	PANADR	23
4.2.1.82	PHR_MODE_SUB	23
4.2.1.83	PMSC	23
4.2.1.84	PMSC_CTRL0_SUB	23
4.2.1.85	RF_CONF	23
4.2.1.86	RF_RXCTRLH_SUB	23
4.2.1.87	RF_TXCTRL_SUB	23
4.2.1.88	RX_BUFFER	23
4.2.1.89	RX_FINFO	23
4.2.1.90	RX_MODE	23
4.2.1.91	RX_STAMP_SUB	23
4.2.1.92	RX_TIME	23
4.2.1.93	RXAUTR_BIT	23
4.2.1.94	RXDFR_BIT	23
4.2.1.95	RXDLYS_BIT	23
4.2.1.96	RXENAB_BIT	23
4.2.1.97	RXFCE_BIT	23
4.2.1.98	RXFCG_BIT	23
4.2.1.99	RXM110K_BIT	23
4.2.1.100	RXPHE_BIT	23
4.2.1.101	RXRFSL_BIT	23
4.2.1.102	RXRFTO_BIT	23
4.2.1.103	SFCST_BIT	23
4.2.1.104	SYS_CFG	24
4.2.1.105	SYS_CTRL	24
4.2.1.106	SYS_MASK	24
4.2.1.107	SYS_STATUS	24
4.2.1.108	SYS_TIME	24
4.2.1.109	TC_PGDELAY_SUB	24
4.2.1.110	TIME_RES	24
4.2.1.111	TIME_RES_INV	24
4.2.1.112	TRXOFF_BIT	24
4.2.1.113	TX_ANTD	24
4.2.1.114	TX_BUFFER	24
4.2.1.115	TX_CAL	24
4.2.1.116	TX_FCTRL	24

---

4.2.1.117 TX_MODE . . . . .	24
4.2.1.118 TX_POWER . . . . .	24
4.2.1.119 TX_STAMP_SUB . . . . .	24
4.2.1.120 TX_TIME . . . . .	24
4.2.1.121 TXDLYS_BIT . . . . .	24
4.2.1.122 TXFRB_BIT . . . . .	24
4.2.1.123 TXFRS_BIT . . . . .	24
4.2.1.124 TXPHS_BIT . . . . .	24
4.2.1.125 TXPRS_BIT . . . . .	24
4.2.1.126 TXSTRT_BIT . . . . .	24
4.2.1.127 WAIT4RESP_BIT . . . . .	24
4.2.2 Variable Documentation . . . . .	24
4.2.2.1 DW1000 . . . . .	24
4.3 /home/thomas/git/arduino-dw1000/DW1000/DW1000Time.cpp File Reference . . . . .	24
4.4 /home/thomas/git/arduino-dw1000/DW1000/DW1000Time.h File Reference . . . . .	25
4.4.1 Macro Definition Documentation . . . . .	25
4.4.1.1 DISTANCE_OF_RADIO . . . . .	25
4.4.1.2 LEN_STAMP . . . . .	25
4.4.1.3 TIME_RES . . . . .	25
4.4.1.4 TIME_RES_INV . . . . .	25
<b>Index</b>	<b>26</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">DW1000Class</a> . . . . .	5
<a href="#">DW1000Time</a> . . . . .	14



# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">/home/thomas/git/arduino-dw1000/DW1000/DW1000.cpp</a>	17
<a href="#">/home/thomas/git/arduino-dw1000/DW1000/DW1000.h</a>	17
<a href="#">/home/thomas/git/arduino-dw1000/DW1000/DW1000Time.cpp</a>	24
<a href="#">/home/thomas/git/arduino-dw1000/DW1000/DW1000Time.h</a>	25



## Chapter 3

# Data Structure Documentation

### 3.1 DW1000Class Class Reference

```
#include <DW1000.h>
```

#### Static Public Member Functions

- static void [begin](#) (int irq, int rst)
- static void [begin](#) (int irq)
- static void [select](#) (int ss)
- static void [end](#) ()
- static void [reset](#) ()
- static void [softReset](#) ()
- static void [getPrintableDeviceIdentifier](#) (char msgBuffer[])
- static void [getPrintableExtendedUniqueIdentifier](#) (char msgBuffer[])
- static void [getPrintableNetworkIdAndShortAddress](#) (char msgBuffer[])
- static void [getPrintableDeviceMode](#) (char msgBuffer[])
- static void [setNetworkId](#) (unsigned int val)
- static void [setDeviceAddress](#) (unsigned int val)
- static void [setReceiverAutoReenable](#) (boolean val)
- static void [setInterruptPolarity](#) (boolean val)
- static void [suppressFrameCheck](#) (boolean val)
- static void [setDataRate](#) (byte rate)
- static void [setPulseFrequency](#) (byte freq)
- static void [setPreambleLength](#) (byte prealen)
- static void [setChannel](#) (byte channel)
- static void [setPreambleCode](#) (byte preacode)
- static [DW1000Time](#) [setDelay](#) (const [DW1000Time](#) &delay)
- static void [receivePermanently](#) (boolean val)
- static void [setData](#) (byte data[], unsigned int n)
- static void [setData](#) (const String &data)
- static void [getData](#) (byte data[], unsigned int n)
- static void [getData](#) (String &data)
- static unsigned int [getDataLength](#) ()
- static void [getTransmitTimestamp](#) ([DW1000Time](#) &time)
- static void [getReceiveTimestamp](#) ([DW1000Time](#) &time)
- static void [getSystemTimestamp](#) ([DW1000Time](#) &time)
- static void [getTransmitTimestamp](#) (byte data[])
- static void [getReceiveTimestamp](#) (byte data[])

- static void [getSystemTimestamp](#) (byte data[])
- static void [interruptOnSent](#) (boolean val)
- static void [interruptOnReceived](#) (boolean val)
- static void [interruptOnReceiveError](#) (boolean val)
- static void [interruptOnReceiveTimeout](#) (boolean val)
- static void [interruptOnReceiveTimestampAvailable](#) (boolean val)
- static void [interruptOnAutomaticAcknowledgeTrigger](#) (boolean val)
- static void [attachSentHandler](#) (void(\*handleSent)(void))
- static void [attachReceivedHandler](#) (void(\*handleReceived)(void))
- static void [attachReceiveErrorHandler](#) (void(\*handleReceiveError)(void))
- static void [attachReceiveTimeoutHandler](#) (void(\*handleReceiveTimeout)(void))
- static void [attachReceiveTimestampAvailableHandler](#) (void(\*handleReceiveTimestampAvailable)(void))
- static void [idle](#) ()
- static void [newConfiguration](#) ()
- static void [commitConfiguration](#) ()
- static void [newReceive](#) ()
- static void [startReceive](#) ()
- static void [newTransmit](#) ()
- static void [startTransmit](#) ()
- static void [tune](#) ()
- static void [enableMode](#) (const byte mode[])
- static void [setDefault](#) ()
- static void [getPrettyBytes](#) (byte cmd, word offset, char msgBuffer[], unsigned int n)
- static void [getPrettyBytes](#) (byte data[], char msgBuffer[], unsigned int n)

### Static Public Attributes

- static const byte [TRX\\_RATE\\_110KBPS](#) = 0x00
- static const byte [TRX\\_RATE\\_850KBPS](#) = 0x01
- static const byte [TRX\\_RATE\\_6800KBPS](#) = 0x02
- static const byte [TX\\_PULSE\\_FREQ\\_16MHZ](#) = 0x01
- static const byte [TX\\_PULSE\\_FREQ\\_64MHZ](#) = 0x02
- static const byte [TX\\_PREAMBLE\\_LEN\\_64](#) = 0x01
- static const byte [TX\\_PREAMBLE\\_LEN\\_128](#) = 0x05
- static const byte [TX\\_PREAMBLE\\_LEN\\_256](#) = 0x09
- static const byte [TX\\_PREAMBLE\\_LEN\\_512](#) = 0x0D
- static const byte [TX\\_PREAMBLE\\_LEN\\_1024](#) = 0x02
- static const byte [TX\\_PREAMBLE\\_LEN\\_1536](#) = 0x06
- static const byte [TX\\_PREAMBLE\\_LEN\\_2048](#) = 0x0A
- static const byte [TX\\_PREAMBLE\\_LEN\\_4096](#) = 0x03
- static const byte [PAC\\_SIZE\\_8](#) = 8
- static const byte [PAC\\_SIZE\\_16](#) = 16
- static const byte [PAC\\_SIZE\\_32](#) = 32
- static const byte [PAC\\_SIZE\\_64](#) = 64
- static const byte [CHANNEL\\_1](#) = 1
- static const byte [CHANNEL\\_2](#) = 2
- static const byte [CHANNEL\\_3](#) = 3
- static const byte [CHANNEL\\_4](#) = 4
- static const byte [CHANNEL\\_5](#) = 5
- static const byte [CHANNEL\\_7](#) = 7
- static const byte [PREAMBLE\\_CODE\\_16MHZ\\_1](#) = 1
- static const byte [PREAMBLE\\_CODE\\_16MHZ\\_2](#) = 2
- static const byte [PREAMBLE\\_CODE\\_16MHZ\\_3](#) = 3
- static const byte [PREAMBLE\\_CODE\\_16MHZ\\_4](#) = 4



- static const byte `PREAMBLE_CODE_16MHZ_5` = 5
- static const byte `PREAMBLE_CODE_16MHZ_6` = 6
- static const byte `PREAMBLE_CODE_16MHZ_7` = 7
- static const byte `PREAMBLE_CODE_16MHZ_8` = 8
- static const byte `PREAMBLE_CODE_64MHZ_9` = 9
- static const byte `PREAMBLE_CODE_64MHZ_10` = 10
- static const byte `PREAMBLE_CODE_64MHZ_11` = 11
- static const byte `PREAMBLE_CODE_64MHZ_12` = 12
- static const byte `PREAMBLE_CODE_64MHZ_17` = 17
- static const byte `PREAMBLE_CODE_64MHZ_18` = 18
- static const byte `PREAMBLE_CODE_64MHZ_19` = 19
- static const byte `PREAMBLE_CODE_64MHZ_20` = 20
- static const byte `FRAME_LENGTH_NORMAL` = 0x00
- static const byte `FRAME_LENGTH_EXTENDED` = 0x03
- static const byte `MODE_LONGDATA_RANGE_LOWPOWER` [] = {`TRX_RATE_110KBPS`, `TX_PULSE_FREQ_16MHZ`, `TX_PREAMBLE_LEN_1024`}
- static const byte `MODE_SHORTDATA_FAST_LOWPOWER` [] = {`TRX_RATE_6800KBPS`, `TX_PULSE_FREQ_16MHZ`, `TX_PREAMBLE_LEN_128`}
- static const byte `MODE_LONGDATA_FAST_LOWPOWER` [] = {`TRX_RATE_6800KBPS`, `TX_PULSE_FREQ_16MHZ`, `TX_PREAMBLE_LEN_1024`}
- static const byte `MODE_SHORTDATA_FAST_ACCURACY` [] = {`TRX_RATE_6800KBPS`, `TX_PULSE_FREQ_64MHZ`, `TX_PREAMBLE_LEN_128`}
- static const byte `MODE_LONGDATA_FAST_ACCURACY` [] = {`TRX_RATE_6800KBPS`, `TX_PULSE_FREQ_64MHZ`, `TX_PREAMBLE_LEN_1024`}
- static const byte `MODE_LONGDATA_RANGE_ACCURACY` [] = {`TRX_RATE_110KBPS`, `TX_PULSE_FREQ_64MHZ`, `TX_PREAMBLE_LEN_1024`}

### 3.1.1 Member Function Documentation

3.1.1.1 `static void DW1000Class::attachReceivedHandler ( void(*)(void) handleReceived )` [inline],[static]

3.1.1.2 `static void DW1000Class::attachReceiveErrorHandler ( void(*)(void) handleReceiveError )` [inline],[static]

3.1.1.3 `static void DW1000Class::attachReceiveTimeoutHandler ( void(*)(void) handleReceiveTimeout )` [inline],[static]

3.1.1.4 `static void DW1000Class::attachReceiveTimestampAvailableHandler ( void(*)(void) handleReceiveTimestampAvailable )` [inline],[static]

3.1.1.5 `static void DW1000Class::attachSentHandler ( void(*)(void) handleSent )` [inline],[static]

3.1.1.6 `void DW1000Class::begin ( int irq, int rst )` [static]

Initiates and starts a sessions with one or more DW1000.

#### Parameters

<code>in</code>	<code><i>irq</i></code>	The interrupt line/pin that connects the Arduino.
<code>in</code>	<code><i>rst</i></code>	The reset line/pin for hard resets of ICs that connect to the Arduino.

3.1.1.7 `void DW1000Class::begin ( int irq )` [static]

Initiates and starts a sessions with one or more DW1000. Soft resets (i.e. command triggered) are used and it is assumed that no reset line is wired.

## Parameters

in	<i>irq</i>	The interrupt line/pin that connects the Arduino.
----	------------	---

3.1.1.8 void DW1000Class::commitConfiguration ( ) [static]

3.1.1.9 void DW1000Class::enableMode ( const byte *mode[]* ) [static]

3.1.1.10 void DW1000Class::end ( ) [static]

Tells the driver library that no communication to a DW1000 will be required anymore. This basically just frees SPI and the previously used pins.

3.1.1.11 void DW1000Class::getData ( byte *data[]*, unsigned int *n* ) [static]

3.1.1.12 void DW1000Class::getData ( String & *data* ) [static]

3.1.1.13 unsigned int DW1000Class::getDataLength ( ) [static]

3.1.1.14 void DW1000Class::getPrettyBytes ( byte *cmd*, word *offset*, char *msgBuffer[]*, unsigned int *n* ) [static]

3.1.1.15 void DW1000Class::getPrettyBytes ( byte *data[]*, char *msgBuffer[]*, unsigned int *n* ) [static]

3.1.1.16 void DW1000Class::getPrintableDeviceIdentifier ( char *msgBuffer[]* ) [static]

Generates a String representation of the device identifier of the chip. That usually are the letters "DECA" plus the version and revision numbers of the chip.

## Parameters

out	<i>msgBuffer</i>	The String buffer to be filled with printable device information. Provide 128 bytes, this should be sufficient.
-----	------------------	---

3.1.1.17 void DW1000Class::getPrintableDeviceMode ( char *msgBuffer[]* ) [static]

Generates a String representation of the main operational settings of the chip. This includes data rate, pulse repetition frequency, preamble and channel settings.

## Parameters

out	<i>msgBuffer</i>	The String buffer to be filled with printable device information. Provide 128 bytes, this should be sufficient.
-----	------------------	---

3.1.1.18 void DW1000Class::getPrintableExtendedUniqueIdentifier ( char *msgBuffer[]* ) [static]

Generates a String representation of the extended unique identifier (EUI) of the chip.

## Parameters

out	<i>msgBuffer</i>	The String buffer to be filled with printable device information. Provide 128 bytes, this should be sufficient.
-----	------------------	---

3.1.1.19 void DW1000Class::getPrintableNetworkIdAndShortAddress ( char *msgBuffer*[] ) [static]

Generates a String representation of the short address and network identifier currently defined for the respective chip.

## Parameters

out	<i>msgBuffer</i>	The String buffer to be filled with printable device information. Provide 128 bytes, this should be sufficient.
-----	------------------	---

- 3.1.1.20 void DW1000Class::getReceiveTimestamp ( DW1000Time & time ) [static]
- 3.1.1.21 void DW1000Class::getReceiveTimestamp ( byte data[] ) [static]
- 3.1.1.22 void DW1000Class::getSystemTimestamp ( DW1000Time & time ) [static]
- 3.1.1.23 void DW1000Class::getSystemTimestamp ( byte data[] ) [static]
- 3.1.1.24 void DW1000Class::getTransmitTimestamp ( DW1000Time & time ) [static]
- 3.1.1.25 void DW1000Class::getTransmitTimestamp ( byte data[] ) [static]
- 3.1.1.26 void DW1000Class::idle ( ) [static]
- 3.1.1.27 void DW1000Class::interruptOnAutomaticAcknowledgeTrigger ( boolean val ) [static]
- 3.1.1.28 void DW1000Class::interruptOnReceived ( boolean val ) [static]
- 3.1.1.29 void DW1000Class::interruptOnReceiveError ( boolean val ) [static]
- 3.1.1.30 void DW1000Class::interruptOnReceiveTimeout ( boolean val ) [static]
- 3.1.1.31 void DW1000Class::interruptOnReceiveTimestampAvailable ( boolean val ) [static]
- 3.1.1.32 void DW1000Class::interruptOnSent ( boolean val ) [static]
- 3.1.1.33 void DW1000Class::newConfiguration ( ) [static]
- 3.1.1.34 void DW1000Class::newReceive ( ) [static]
- 3.1.1.35 void DW1000Class::newTransmit ( ) [static]
- 3.1.1.36 void DW1000Class::receivePermanently ( boolean val ) [static]
- 3.1.1.37 void DW1000Class::reset ( ) [static]

Resets all connected or the currently selected DW1000 chip. A hard reset of all chips is preferred, although a soft reset of the currently selected one is executed if no reset pin has been specified (when using `begin(int)`, instead of `begin(int, int)`).

- 3.1.1.38 void DW1000Class::select ( int ss ) [static]

Selects a specific DW1000 chip for communication. In case of a single DW1000 chip in use this call only needs to be done once at start up, but is still mandatory.

## Parameters

in	ss	The chip select line/pin that connects the to-be-selected chip with the Arduino.
----	----	--

3.1.1.39 void DW1000Class::setChannel ( byte *channel* ) [static]

3.1.1.40 void DW1000Class::setData ( byte *data*[], unsigned int *n* ) [static]

3.1.1.41 void DW1000Class::setData ( const String & *data* ) [static]

3.1.1.42 void DW1000Class::setDataRate ( byte *rate* ) [static]

Specifies the data transmission rate of the DW1000 chip. One of the values TRX\_RATE\_110KBPS (i.e. 110 kb/s) TRX\_RATE\_850KBPS (i.e. 850 kb/s) TRX\_RATE\_6800KBPS (i.e. 6.8 Mb/s) has to be provided.

See [setDefaultts\(\)](#) and [enableMode\(\)](#) for additional information on data rate settings.

Parameters

in	<i>rate</i>	The data transmission rate, encoded by the above defined constants.
----	-------------	---

3.1.1.43 void DW1000Class::setDefaultts ( ) [static]

3.1.1.44 DW1000Time DW1000Class::setDelay ( const DW1000Time & *delay* ) [static]

3.1.1.45 void DW1000Class::setDeviceAddress ( unsigned int *val* ) [static]

(Re-)set the device address (i.e. short address) for the currently selected chip. This setting is important for certain MAC address filtering rules.

Parameters

in	<i>val</i>	An arbitrary numeric device address.
----	------------	--------------------------------------

3.1.1.46 void DW1000Class::setInterruptPolarity ( boolean *val* ) [static]

Specifies the interrupt polarity of the DW1000 chip.

As part of [setDefaultts\(\)](#) if the device is in idle mode, interrupt polarity is set to active high.

Parameters

in	<i>val</i>	true for active high interrupts, false for active low interrupts.
----	------------	---

3.1.1.47 void DW1000Class::setNetworkId ( unsigned int *val* ) [static]

(Re-)set the network identifier which the selected chip should be associated with. This setting is important for certain MAC address filtering rules.

Parameters

in	<i>val</i>	An arbitrary numeric network identifier.
----	------------	--

3.1.1.48 void DW1000Class::setPreambleCode ( byte *preacode* ) [static]

3.1.1.49 void DW1000Class::setPreambleLength ( byte *prealen* ) [static]

### 3.1.1.50 void DW1000Class::setPulseFrequency ( byte *freq* ) [static]

Specifies the pulse repetition frequency (PRF) of data transmissions with the DW1000. Either TX\_PULSE\_FREQ\_16MHZ (i.e. 16 MHz) TX\_PULSE\_FREQ\_64MHZ (i.e. 64 MHz) has to be chosen.

Note that the 16 MHz setting is more power efficient, while the 64 MHz setting requires more power, but also delivers slightly better transmission performance (i.e. on communication range and timestamp accuracy) (see DWM1000 User Manual, section 9.3).

See [setDefaultts\(\)](#) and [enableMode\(\)](#) for additional information on PRF settings.

#### Parameters

<i>in</i>	<i>freq</i>	The PRF, encoded by the above defined constants.
-----------	-------------	--

### 3.1.1.51 void DW1000Class::setReceiverAutoReenable ( boolean *val* ) [static]

Specifies whether the DW1000 chip should, again, turn on its receiver in case that the last reception failed.

This setting is enabled as part of [setDefaultts\(\)](#) if the device is in idle mode.

#### Parameters

<i>in</i>	<i>val</i>	true to enable, false to disable receiver auto-reenable.
-----------	------------	--

### 3.1.1.52 void DW1000Class::softReset ( ) [static]

Resets the currently selected DW1000 chip programmatically (via corresponding commands).

### 3.1.1.53 void DW1000Class::startReceive ( ) [static]

### 3.1.1.54 void DW1000Class::startTransmit ( ) [static]

### 3.1.1.55 void DW1000Class::suppressFrameCheck ( boolean *val* ) [static]

Specifies whether to suppress any frame check measures while sending or receiving messages. If suppressed, no 2-byte checksum is appended to the message before sending and this checksum is also not expected at receiver side. Note that when suppressing frame checks, the error event handler (attached via [attachReceiveErrorHandler\(\)](#)) will not be triggered if received data is corrupted.

Frame checks are enabled as part of [setDefaultts\(\)](#) if the device is in idle mode.

#### Parameters

<i>in</i>	<i>val</i>	true to suppress frame check on sender and receiver side, false otherwise.
-----------	------------	--

### 3.1.1.56 void DW1000Class::tune ( ) [static]

## 3.1.2 Field Documentation

### 3.1.2.1 const byte DW1000Class::CHANNEL\_1 = 1 [static]

### 3.1.2.2 const byte DW1000Class::CHANNEL\_2 = 2 [static]

### 3.1.2.3 const byte DW1000Class::CHANNEL\_3 = 3 [static]

- 3.1.2.4 const byte DW1000Class::CHANNEL\_4 = 4 [static]
- 3.1.2.5 const byte DW1000Class::CHANNEL\_5 = 5 [static]
- 3.1.2.6 const byte DW1000Class::CHANNEL\_7 = 7 [static]
- 3.1.2.7 const byte DW1000Class::FRAME\_LENGTH\_EXTENDED = 0x03 [static]
- 3.1.2.8 const byte DW1000Class::FRAME\_LENGTH\_NORMAL = 0x00 [static]
- 3.1.2.9 const byte DW1000Class::MODE\_LONGDATA\_FAST\_ACCURACY = {TRX\_RATE\_6800KBPS, TX\_PULSE\_FREQ\_64MHZ, TX\_PREAMBLE\_LEN\_1024} [static]
- 3.1.2.10 const byte DW1000Class::MODE\_LONGDATA\_FAST\_LOWPOWER = {TRX\_RATE\_6800KBPS, TX\_PULSE\_FREQ\_16MHZ, TX\_PREAMBLE\_LEN\_1024} [static]
- 3.1.2.11 const byte DW1000Class::MODE\_LONGDATA\_RANGE\_ACCURACY = {TRX\_RATE\_110KBPS, TX\_PULSE\_FREQ\_64MHZ, TX\_PREAMBLE\_LEN\_1024} [static]
- 3.1.2.12 const byte DW1000Class::MODE\_LONGDATA\_RANGE\_LOWPOWER = {TRX\_RATE\_110KBPS, TX\_PULSE\_FREQ\_16MHZ, TX\_PREAMBLE\_LEN\_1024} [static]
- 3.1.2.13 const byte DW1000Class::MODE\_SHORTDATA\_FAST\_ACCURACY = {TRX\_RATE\_6800KBPS, TX\_PULSE\_FREQ\_64MHZ, TX\_PREAMBLE\_LEN\_128} [static]
- 3.1.2.14 const byte DW1000Class::MODE\_SHORTDATA\_FAST\_LOWPOWER = {TRX\_RATE\_6800KBPS, TX\_PULSE\_FREQ\_16MHZ, TX\_PREAMBLE\_LEN\_128} [static]
- 3.1.2.15 const byte DW1000Class::PAC\_SIZE\_16 = 16 [static]
- 3.1.2.16 const byte DW1000Class::PAC\_SIZE\_32 = 32 [static]
- 3.1.2.17 const byte DW1000Class::PAC\_SIZE\_64 = 64 [static]
- 3.1.2.18 const byte DW1000Class::PAC\_SIZE\_8 = 8 [static]
- 3.1.2.19 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_1 = 1 [static]
- 3.1.2.20 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_2 = 2 [static]
- 3.1.2.21 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_3 = 3 [static]
- 3.1.2.22 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_4 = 4 [static]
- 3.1.2.23 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_5 = 5 [static]
- 3.1.2.24 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_6 = 6 [static]
- 3.1.2.25 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_7 = 7 [static]
- 3.1.2.26 const byte DW1000Class::PREAMBLE\_CODE\_16MHZ\_8 = 8 [static]
- 3.1.2.27 const byte DW1000Class::PREAMBLE\_CODE\_64MHZ\_10 = 10 [static]
- 3.1.2.28 const byte DW1000Class::PREAMBLE\_CODE\_64MHZ\_11 = 11 [static]

- 3.1.2.29 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_12 = 12` [static]
- 3.1.2.30 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_17 = 17` [static]
- 3.1.2.31 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_18 = 18` [static]
- 3.1.2.32 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_19 = 19` [static]
- 3.1.2.33 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_20 = 20` [static]
- 3.1.2.34 `const byte DW1000Class::PREAMBLE_CODE_64MHZ_9 = 9` [static]
- 3.1.2.35 `const byte DW1000Class::TRX_RATE_110KBPS = 0x00` [static]
- 3.1.2.36 `const byte DW1000Class::TRX_RATE_6800KBPS = 0x02` [static]
- 3.1.2.37 `const byte DW1000Class::TRX_RATE_850KBPS = 0x01` [static]
- 3.1.2.38 `const byte DW1000Class::TX_PREAMBLE_LEN_1024 = 0x02` [static]
- 3.1.2.39 `const byte DW1000Class::TX_PREAMBLE_LEN_128 = 0x05` [static]
- 3.1.2.40 `const byte DW1000Class::TX_PREAMBLE_LEN_1536 = 0x06` [static]
- 3.1.2.41 `const byte DW1000Class::TX_PREAMBLE_LEN_2048 = 0x0A` [static]
- 3.1.2.42 `const byte DW1000Class::TX_PREAMBLE_LEN_256 = 0x09` [static]
- 3.1.2.43 `const byte DW1000Class::TX_PREAMBLE_LEN_4096 = 0x03` [static]
- 3.1.2.44 `const byte DW1000Class::TX_PREAMBLE_LEN_512 = 0x0D` [static]
- 3.1.2.45 `const byte DW1000Class::TX_PREAMBLE_LEN_64 = 0x01` [static]
- 3.1.2.46 `const byte DW1000Class::TX_PULSE_FREQ_16MHZ = 0x01` [static]
- 3.1.2.47 `const byte DW1000Class::TX_PULSE_FREQ_64MHZ = 0x02` [static]

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

- [/home/thomas/git/arduino-dw1000/DW1000/DW1000.h](#)
- [/home/thomas/git/arduino-dw1000/DW1000/DW1000.cpp](#)

## 3.2 DW1000Time Class Reference

```
#include <DW1000Time.h>
```

### Public Member Functions

- [DW1000Time](#) ()
- [DW1000Time](#) (long long int time)
- [DW1000Time](#) (float timeUs)
- [DW1000Time](#) (byte data[])
- [DW1000Time](#) (long value, float factorUs)



- [DW1000Time](#) (const [DW1000Time](#) &copy)
- [~DW1000Time](#) ( )
- void [setTime](#) (float timeUs)
- void [setTime](#) (long value, float factorUs)
- float [getAsFloat](#) ( ) const
- void [getAsBytes](#) (byte data[]) const
- float [getAsMeters](#) ( ) const
- void [getTimestamp](#) (byte data[]) const
- long long int [getTimestamp](#) ( ) const
- void [setTimestamp](#) (byte data[])
- void [setTimestamp](#) (const [DW1000Time](#) &copy)
- [DW1000Time](#) & [operator=](#) (const [DW1000Time](#) &assign)
- [DW1000Time](#) & [operator+=](#) (const [DW1000Time](#) &add)
- const [DW1000Time](#) [operator+](#) (const [DW1000Time](#) &add) const
- [DW1000Time](#) & [operator-=](#) (const [DW1000Time](#) &sub)
- const [DW1000Time](#) [operator-](#) (const [DW1000Time](#) &sub) const
- [DW1000Time](#) & [operator\\*=\[operator\\\*=" \\(float factor\\)\]\(#\)](#)
- const [DW1000Time](#) [operator\\*\[operator\\\*=" \\(float factor\\) const\]\(#\)](#)
- [DW1000Time](#) & [operator/= \(float factor\)](#)
- const [DW1000Time](#) [operator/ \(float factor\) const](#)
- boolean [operator==](#) (const [DW1000Time](#) &cmp) const
- boolean [operator!=](#) (const [DW1000Time](#) &cmp) const

### Static Public Attributes

- static const float [SECONDS](#) = 1e6
- static const float [MILLISECONDS](#) = 1e3
- static const float [MICROSECONDS](#) = 1
- static const float [NANOSECONDS](#) = 1e-3
- static const long long unsigned int [TIME\\_OVERFLOW](#) = 1099511627776

### 3.2.1 Constructor & Destructor Documentation

- 3.2.1.1 [DW1000Time::DW1000Time](#) ( )
- 3.2.1.2 [DW1000Time::DW1000Time](#) ( long long int *time* )
- 3.2.1.3 [DW1000Time::DW1000Time](#) ( float *timeUs* )
- 3.2.1.4 [DW1000Time::DW1000Time](#) ( byte *data[]* )
- 3.2.1.5 [DW1000Time::DW1000Time](#) ( long *value*, float *factorUs* )
- 3.2.1.6 [DW1000Time::DW1000Time](#) ( const [DW1000Time](#) & *copy* )
- 3.2.1.7 [DW1000Time::~DW1000Time](#) ( )

### 3.2.2 Member Function Documentation

- 3.2.2.1 void [DW1000Time::getAsBytes](#) ( byte *data[]* ) const
- 3.2.2.2 float [DW1000Time::getAsFloat](#) ( ) const

- 3.2.2.3 `float DW1000Time::getAsMeters ( ) const`
- 3.2.2.4 `void DW1000Time::getTimestamp ( byte data[ ] ) const`
- 3.2.2.5 `long long int DW1000Time::getTimestamp ( ) const`
- 3.2.2.6 `boolean DW1000Time::operator!= ( const DW1000Time & cmp ) const`
- 3.2.2.7 `const DW1000Time DW1000Time::operator* ( float factor ) const`
- 3.2.2.8 `DW1000Time & DW1000Time::operator*= ( float factor )`
- 3.2.2.9 `const DW1000Time DW1000Time::operator+ ( const DW1000Time & add ) const`
- 3.2.2.10 `DW1000Time & DW1000Time::operator+= ( const DW1000Time & add )`
- 3.2.2.11 `const DW1000Time DW1000Time::operator- ( const DW1000Time & sub ) const`
- 3.2.2.12 `DW1000Time & DW1000Time::operator-= ( const DW1000Time & sub )`
- 3.2.2.13 `const DW1000Time DW1000Time::operator/ ( float factor ) const`
- 3.2.2.14 `DW1000Time & DW1000Time::operator/= ( float factor )`
- 3.2.2.15 `DW1000Time & DW1000Time::operator= ( const DW1000Time & assign )`
- 3.2.2.16 `boolean DW1000Time::operator== ( const DW1000Time & cmp ) const`
- 3.2.2.17 `void DW1000Time::setTime ( float timeUs )`
- 3.2.2.18 `void DW1000Time::setTime ( long value, float factorUs )`
- 3.2.2.19 `void DW1000Time::setTimestamp ( byte data[ ] )`
- 3.2.2.20 `void DW1000Time::setTimestamp ( const DW1000Time & copy )`

### 3.2.3 Field Documentation

- 3.2.3.1 `const float DW1000Time::MICROSECONDS = 1 [static]`
- 3.2.3.2 `const float DW1000Time::MILLISECONDS = 1e3 [static]`
- 3.2.3.3 `const float DW1000Time::NANOSECONDS = 1e-3 [static]`
- 3.2.3.4 `const float DW1000Time::SECONDS = 1e6 [static]`
- 3.2.3.5 `const long long unsigned int DW1000Time::TIME_OVERFLOW = 1099511627776 [static]`

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

- </home/thomas/git/arduino-dw1000/DW1000/DW1000Time.h>
- </home/thomas/git/arduino-dw1000/DW1000/DW1000Time.cpp>

## Chapter 4

# File Documentation

### 4.1 /home/thomas/git/arduino-dw1000/DW1000/DW1000.cpp File Reference

```
#include "pins_arduino.h"  
#include "DW1000.h"
```

#### Variables

- [DW1000Class DW1000](#)

#### 4.1.1 Variable Documentation

##### 4.1.1.1 DW1000Class DW1000

### 4.2 /home/thomas/git/arduino-dw1000/DW1000/DW1000.h File Reference

```
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
#include <Arduino.h>  
#include "DW1000Time.h"  
#include "../SPI/SPI.h"
```

#### Data Structures

- class [DW1000Class](#)

#### Macros

- #define [TIME\\_RES](#) 0.000015650040064103
- #define [TIME\\_RES\\_INV](#) 63897.6
- #define [LEN\\_STAMP](#) 5
- #define [IDLE\\_MODE](#) 0x00
- #define [RX\\_MODE](#) 0x01
- #define [TX\\_MODE](#) 0x02
- #define [JUNK](#) 0x00

- #define NO\_SUB 0x00
- #define DEV\_ID 0x00
- #define LEN\_DEV\_ID 4
- #define EUI 0x01
- #define LEN\_EUI 8
- #define PANADR 0x03
- #define LEN\_PANADR 4
- #define SYS\_CFG 0x04
- #define LEN\_SYS\_CFG 4
- #define FFEN\_BIT 0
- #define DIS\_DRXB\_BIT 12
- #define HIRQ\_POL\_BIT 9
- #define RXAUTR\_BIT 29
- #define PHR\_MODE\_SUB 16
- #define LEN\_PHR\_MODE\_SUB 2
- #define RXM110K\_BIT 22
- #define SYS\_CTRL 0x0D
- #define LEN\_SYS\_CTRL 4
- #define SFCST\_BIT 0
- #define TXSTRT\_BIT 1
- #define TXDLYS\_BIT 2
- #define TRXOFF\_BIT 6
- #define WAIT4RESP\_BIT 7
- #define RXENAB\_BIT 8
- #define RXDLYS\_BIT 9
- #define SYS\_STATUS 0x0F
- #define LEN\_SYS\_STATUS 5
- #define AAT\_BIT 3
- #define TXFRB\_BIT 4
- #define TXPRS\_BIT 5
- #define TXPHS\_BIT 6
- #define TXFRS\_BIT 7
- #define LDEDONE\_BIT 10
- #define RXPHE\_BIT 12
- #define RXDFR\_BIT 13
- #define RXFCG\_BIT 14
- #define RXFCE\_BIT 15
- #define RXRFSL\_BIT 16
- #define RXRFTO\_BIT 17
- #define LDEERR\_BIT 18
- #define SYS\_MASK 0x0E
- #define LEN\_SYS\_MASK 4
- #define SYS\_TIME 0x06
- #define LEN\_SYS\_TIME LEN\_STAMP
- #define RX\_TIME 0x15
- #define LEN\_RX\_TIME 14
- #define RX\_STAMP\_SUB 0
- #define LEN\_RX\_STAMP LEN\_STAMP
- #define TX\_TIME 0x17
- #define LEN\_TX\_TIME 10
- #define TX\_STAMP\_SUB 0
- #define LEN\_TX\_STAMP LEN\_STAMP
- #define DX\_TIME 0x0A
- #define LEN\_DX\_TIME LEN\_STAMP
- #define TX\_BUFFER 0x09

- #define `LEN_TX_BUFFER` 1024
- #define `LEN_UWB_FRAMES` 127
- #define `LEN_EXT_UWB_FRAMES` 1023
- #define `RX_FINFO` 0x10
- #define `LEN_RX_FINFO` 4
- #define `RX_BUFFER` 0x11
- #define `LEN_RX_BUFFER` 1024
- #define `TX_FCTRL` 0x08
- #define `LEN_TX_FCTRL` 5
- #define `CHAN_CTRL` 0x1F
- #define `LEN_CHAN_CTRL` 4
- #define `OTP_IF` 0x2D
- #define `OTP_ADDR_SUB` 0x04
- #define `OTP_CTRL_SUB` 0x06
- #define `OTP_RDAT_SUB` 0x0A
- #define `LEN_OTP_ADDR` 2
- #define `LEN_OTP_CTRL` 2
- #define `LEN_OTP_RDAT` 4
- #define `AGC_TUNE` 0x23
- #define `AGC_TUNE1_SUB` 0x04
- #define `AGC_TUNE2_SUB` 0x0C
- #define `AGC_TUNE3_SUB` 0x12
- #define `LEN_AGC_TUNE1` 2
- #define `LEN_AGC_TUNE2` 4
- #define `LEN_AGC_TUNE3` 2
- #define `DRX_TUNE` 0x27
- #define `DRX_TUNE0b_SUB` 0x02
- #define `DRX_TUNE1a_SUB` 0x04
- #define `DRX_TUNE1b_SUB` 0x06
- #define `DRX_TUNE2_SUB` 0x08
- #define `DRX_TUNE4H_SUB` 0x26
- #define `LEN_DRX_TUNE0b` 2
- #define `LEN_DRX_TUNE1a` 2
- #define `LEN_DRX_TUNE1b` 2
- #define `LEN_DRX_TUNE2` 4
- #define `LEN_DRX_TUNE4H` 2
- #define `LDE_IF` 0x2E
- #define `LDE_CFG1_SUB` 0x0806
- #define `LDE_RXANTD_SUB` 0x1804
- #define `LDE_CFG2_SUB` 0x1806
- #define `LDE_REPC_SUB` 0x2804
- #define `LEN_LDE_CFG1` 1
- #define `LEN_LDE_CFG2` 2
- #define `LEN_LDE_REPC` 2
- #define `LEN_LDE_RXANTD` 2
- #define `TX_POWER` 0x1E
- #define `LEN_TX_POWER` 4
- #define `RF_CONF` 0x28
- #define `RF_RXCTRLH_SUB` 0x0B
- #define `RF_TXCTRL_SUB` 0x0C
- #define `LEN_RF_RXCTRLH` 1
- #define `LEN_RF_TXCTRL` 4
- #define `TX_CAL` 0x2A
- #define `TC_PGDELAY_SUB` 0x0B
- #define `LEN_TC_PGDELAY` 1

- `#define FS_CTRL 0x2B`
- `#define FS_PLLCFG_SUB 0x07`
- `#define FS_PLLTUNE_SUB 0x0B`
- `#define LEN_FS_PLLCFG 4`
- `#define LEN_FS_PLLTUNE 1`
- `#define PMSC 0x36`
- `#define PMSC_CTRL0_SUB 0x00`
- `#define LEN_PMSC_CTRL0 4`
- `#define TX_ANTD 0x18`
- `#define LEN_TX_ANTD 2`

## Variables

- `DW1000Class DW1000`

## 4.2.1 Macro Definition Documentation

4.2.1.1 `#define AAT_BIT 3`

4.2.1.2 `#define AGC_TUNE 0x23`

4.2.1.3 `#define AGC_TUNE1_SUB 0x04`

4.2.1.4 `#define AGC_TUNE2_SUB 0x0C`

4.2.1.5 `#define AGC_TUNE3_SUB 0x12`

4.2.1.6 `#define CHAN_CTRL 0x1F`

4.2.1.7 `#define DEV_ID 0x00`

4.2.1.8 `#define DIS_DRXB_BIT 12`

4.2.1.9 `#define DRX_TUNE 0x27`

4.2.1.10 `#define DRX_TUNE0b_SUB 0x02`

4.2.1.11 `#define DRX_TUNE1a_SUB 0x04`

4.2.1.12 `#define DRX_TUNE1b_SUB 0x06`

4.2.1.13 `#define DRX_TUNE2_SUB 0x08`

4.2.1.14 `#define DRX_TUNE4H_SUB 0x26`

4.2.1.15 `#define DX_TIME 0x0A`

4.2.1.16 `#define EUI 0x01`

4.2.1.17 `#define FFEN_BIT 0`

4.2.1.18 `#define FS_CTRL 0x2B`

4.2.1.19 `#define FS_PLLCFG_SUB 0x07`

4.2.1.20 `#define FS_PLLTUNE_SUB 0x0B`

4.2.1.21 `#define HIRQ_POL_BIT 9`

4.2.1.22 `#define IDLE_MODE 0x00`

4.2.1.23 `#define JUNK 0x00`

4.2.1.24 `#define LDE_CFG1_SUB 0x0806`

4.2.1.25 `#define LDE_CFG2_SUB 0x1806`

4.2.1.26 `#define LDE_IF 0x2E`

4.2.1.27 `#define LDE_REPC_SUB 0x2804`

4.2.1.28 `#define LDE_RXANTD_SUB 0x1804`

4.2.1.29 `#define LDEDONE_BIT 10`

4.2.1.30 `#define LDEERR_BIT 18`

4.2.1.31 `#define LEN_AGC_TUNE1 2`

4.2.1.32 `#define LEN_AGC_TUNE2 4`

4.2.1.33 `#define LEN_AGC_TUNE3 2`

4.2.1.34 `#define LEN_CHAN_CTRL 4`

4.2.1.35 `#define LEN_DEV_ID 4`

4.2.1.36 `#define LEN_DRX_TUNE0b 2`

4.2.1.37 `#define LEN_DRX_TUNE1a 2`

4.2.1.38 `#define LEN_DRX_TUNE1b 2`

4.2.1.39 `#define LEN_DRX_TUNE2 4`

4.2.1.40 `#define LEN_DRX_TUNE4H 2`

4.2.1.41 `#define LEN_DX_TIME LEN_STAMP`

4.2.1.42 `#define LEN_EUI 8`

4.2.1.43 `#define LEN_EXT_UWB_FRAMES 1023`

4.2.1.44 `#define LEN_FS_PLLCFG 4`

4.2.1.45 `#define LEN_FS_PLLTUNE 1`

4.2.1.46 `#define LEN_LDE_CFG1 1`

4.2.1.47 `#define LEN_LDE_CFG2 2`

4.2.1.48 `#define LEN_LDE_REPC 2`

4.2.1.49 `#define LEN_LDE_RXANTD 2`

4.2.1.50 `#define LEN_OTP_ADDR 2`

4.2.1.51 `#define LEN_OTP_CTRL 2`

4.2.1.52 `#define LEN_OTP_RDAT 4`

4.2.1.53 `#define LEN_PANADR 4`

4.2.1.54 `#define LEN_PHR_MODE_SUB 2`

4.2.1.55 `#define LEN_PMSC_CTRL0 4`

4.2.1.56 `#define LEN_RF_RXCTRLH 1`

4.2.1.57 `#define LEN_RF_TXCTRL 4`

4.2.1.58 `#define LEN_RX_BUFFER 1024`

4.2.1.59 `#define LEN_RX_FINFO 4`

4.2.1.60 `#define LEN_RX_STAMP LEN_STAMP`

4.2.1.61 `#define LEN_RX_TIME 14`

4.2.1.62 `#define LEN_STAMP 5`

4.2.1.63 `#define LEN_SYS_CFG 4`

4.2.1.64 `#define LEN_SYS_CTRL 4`

4.2.1.65 `#define LEN_SYS_MASK 4`

4.2.1.66 `#define LEN_SYS_STATUS 5`

4.2.1.67 `#define LEN_SYS_TIME LEN_STAMP`

4.2.1.68 `#define LEN_TC_PGDELAY 1`

4.2.1.69 `#define LEN_TX_ANTD 2`

4.2.1.70 `#define LEN_TX_BUFFER 1024`

4.2.1.71 `#define LEN_TX_FCTRL 5`

4.2.1.72 `#define LEN_TX_POWER 4`

4.2.1.73 `#define LEN_TX_STAMP LEN_STAMP`

4.2.1.74 `#define LEN_TX_TIME 10`

4.2.1.75 `#define LEN_UWB_FRAMES 127`



4.2.1.76 #define NO\_SUB 0x00

4.2.1.77 #define OTP\_ADDR\_SUB 0x04

4.2.1.78 #define OTP\_CTRL\_SUB 0x06

4.2.1.79 #define OTP\_IF 0x2D

4.2.1.80 #define OTP\_RDAT\_SUB 0x0A

4.2.1.81 #define PANADR 0x03

4.2.1.82 #define PHR\_MODE\_SUB 16

4.2.1.83 #define PMSC 0x36

4.2.1.84 #define PMSC\_CTRL0\_SUB 0x00

4.2.1.85 #define RF\_CONF 0x28

4.2.1.86 #define RF\_RXCTRLH\_SUB 0x0B

4.2.1.87 #define RF\_TXCTRL\_SUB 0x0C

4.2.1.88 #define RX\_BUFFER 0x11

4.2.1.89 #define RX\_FINFO 0x10

4.2.1.90 #define RX\_MODE 0x01

4.2.1.91 #define RX\_STAMP\_SUB 0

4.2.1.92 #define RX\_TIME 0x15

4.2.1.93 #define RXAUTR\_BIT 29

4.2.1.94 #define RXDFR\_BIT 13

4.2.1.95 #define RXDLYS\_BIT 9

4.2.1.96 #define RXENAB\_BIT 8

4.2.1.97 #define RXFCE\_BIT 15

4.2.1.98 #define RXFCG\_BIT 14

4.2.1.99 #define RXM110K\_BIT 22

4.2.1.100 #define RXPHE\_BIT 12

4.2.1.101 #define RXRFSL\_BIT 16

4.2.1.102 #define RXRFTO\_BIT 17

4.2.1.103 #define SFCST\_BIT 0

```
4.2.1.104 #define SYS_CFG 0x04
4.2.1.105 #define SYS_CTRL 0x0D
4.2.1.106 #define SYS_MASK 0x0E
4.2.1.107 #define SYS_STATUS 0x0F
4.2.1.108 #define SYS_TIME 0x06
4.2.1.109 #define TC_PGDELAY_SUB 0x0B
4.2.1.110 #define TIME_RES 0.000015650040064103
4.2.1.111 #define TIME_RES_INV 63897.6
4.2.1.112 #define TRXOFF_BIT 6
4.2.1.113 #define TX_ANTD 0x18
4.2.1.114 #define TX_BUFFER 0x09
4.2.1.115 #define TX_CAL 0x2A
4.2.1.116 #define TX_FCTRL 0x08
4.2.1.117 #define TX_MODE 0x02
4.2.1.118 #define TX_POWER 0x1E
4.2.1.119 #define TX_STAMP_SUB 0
4.2.1.120 #define TX_TIME 0x17
4.2.1.121 #define TXDLYS_BIT 2
4.2.1.122 #define TXFRB_BIT 4
4.2.1.123 #define TXFRS_BIT 7
4.2.1.124 #define TXPHS_BIT 6
4.2.1.125 #define TXPRS_BIT 5
4.2.1.126 #define TXSTRT_BIT 1
4.2.1.127 #define WAIT4RESP_BIT 7
```

## 4.2.2 Variable Documentation

4.2.2.1 DW1000Class DW1000

## 4.3 /home/thomas/git/arduino-dw1000/DW1000/DW1000Time.cpp File Reference

```
#include "DW1000Time.h"
```

## 4.4 /home/thomas/git/arduino-dw1000/DW1000/DW1000Time.h File Reference

```
#include <Arduino.h>
```

### Data Structures

- class [DW1000Time](#)

### Macros

- #define [TIME\\_RES](#) 0.000015650040064103f
- #define [TIME\\_RES\\_INV](#) 63897.6f
- #define [DISTANCE\\_OF\\_RADIO](#) 0.0046917639786159f
- #define [LEN\\_STAMP](#) 5

#### 4.4.1 Macro Definition Documentation

4.4.1.1 #define [DISTANCE\\_OF\\_RADIO](#) 0.0046917639786159f

4.4.1.2 #define [LEN\\_STAMP](#) 5

4.4.1.3 #define [TIME\\_RES](#) 0.000015650040064103f

4.4.1.4 #define [TIME\\_RES\\_INV](#) 63897.6f

# Index

- ~DW1000Time
  - DW1000Time, [15](#)
- /home/thomas/git/arduino-dw1000/DW1000/DW1000.-  
cpp, [17](#)
- /home/thomas/git/arduino-dw1000/DW1000/DW1000.h,  
[17](#)
- /home/thomas/git/arduino-dw1000/DW1000/DW1000-  
Time.cpp, [24](#)
- /home/thomas/git/arduino-dw1000/DW1000/DW1000-  
Time.h, [25](#)
  
- AAT\_BIT
  - DW1000.h, [20](#)
- AGC\_TUNE
  - DW1000.h, [20](#)
- AGC\_TUNE1\_SUB
  - DW1000.h, [20](#)
- AGC\_TUNE2\_SUB
  - DW1000.h, [20](#)
- AGC\_TUNE3\_SUB
  - DW1000.h, [20](#)
- attachReceiveErrorHandler
  - DW1000Class, [7](#)
- attachReceiveTimeoutHandler
  - DW1000Class, [7](#)
- attachReceiveTimestampAvailableHandler
  - DW1000Class, [7](#)
- attachReceivedHandler
  - DW1000Class, [7](#)
- attachSentHandler
  - DW1000Class, [7](#)
  
- begin
  - DW1000Class, [7](#)
  
- CHAN\_CTRL
  - DW1000.h, [20](#)
- CHANNEL\_1
  - DW1000Class, [12](#)
- CHANNEL\_2
  - DW1000Class, [12](#)
- CHANNEL\_3
  - DW1000Class, [12](#)
- CHANNEL\_4
  - DW1000Class, [12](#)
- CHANNEL\_5
  - DW1000Class, [13](#)
- CHANNEL\_7
  - DW1000Class, [13](#)
- commitConfiguration
  - DW1000Class, [8](#)
  
- DEV\_ID
  - DW1000.h, [20](#)
- DIS\_DRXB\_BIT
  - DW1000.h, [20](#)
- DISTANCE\_OF\_RADIO
  - DW1000Time.h, [25](#)
- DRX\_TUNE
  - DW1000.h, [20](#)
- DRX\_TUNE0b\_SUB
  - DW1000.h, [20](#)
- DRX\_TUNE1a\_SUB
  - DW1000.h, [20](#)
- DRX\_TUNE1b\_SUB
  - DW1000.h, [20](#)
- DRX\_TUNE2\_SUB
  - DW1000.h, [20](#)
- DRX\_TUNE4H\_SUB
  - DW1000.h, [20](#)
  
- DW1000
  - DW1000.cpp, [17](#)
  - DW1000.h, [24](#)
- DW1000.cpp
  - DW1000, [17](#)
- DW1000.h
  - AAT\_BIT, [20](#)
  - AGC\_TUNE, [20](#)
  - AGC\_TUNE1\_SUB, [20](#)
  - AGC\_TUNE2\_SUB, [20](#)
  - AGC\_TUNE3\_SUB, [20](#)
  - CHAN\_CTRL, [20](#)
  - DEV\_ID, [20](#)
  - DIS\_DRXB\_BIT, [20](#)
  - DRX\_TUNE, [20](#)
  - DRX\_TUNE0b\_SUB, [20](#)
  - DRX\_TUNE1a\_SUB, [20](#)
  - DRX\_TUNE1b\_SUB, [20](#)
  - DRX\_TUNE2\_SUB, [20](#)
  - DRX\_TUNE4H\_SUB, [20](#)
  - DW1000, [24](#)
  - DX\_TIME, [20](#)
  - EUI, [20](#)
  - FFEN\_BIT, [20](#)
  - FS\_CTRL, [20](#)
  - FS\_PLLCFG\_SUB, [20](#)
  - FS\_PLLTUNE\_SUB, [20](#)
  - HIRQ\_POL\_BIT, [21](#)
  - IDLE\_MODE, [21](#)
  - JUNK, [21](#)

LDE\_CFG1\_SUB, 21  
LDE\_CFG2\_SUB, 21  
LDE\_IF, 21  
LDE\_REPC\_SUB, 21  
LDE\_RXANTD\_SUB, 21  
LDEDONE\_BIT, 21  
LDEERR\_BIT, 21  
LEN\_AGC\_TUNE1, 21  
LEN\_AGC\_TUNE2, 21  
LEN\_AGC\_TUNE3, 21  
LEN\_CHAN\_CTRL, 21  
LEN\_DEV\_ID, 21  
LEN\_DRX\_TUNE0b, 21  
LEN\_DRX\_TUNE1a, 21  
LEN\_DRX\_TUNE1b, 21  
LEN\_DRX\_TUNE2, 21  
LEN\_DRX\_TUNE4H, 21  
LEN\_DX\_TIME, 21  
LEN\_EUI, 21  
LEN\_FS\_PLLCFG, 21  
LEN\_FS\_PLLTUNE, 21  
LEN\_LDE\_CFG1, 21  
LEN\_LDE\_CFG2, 21  
LEN\_LDE\_REPC, 21  
LEN\_LDE\_RXANTD, 22  
LEN\_OTP\_ADDR, 22  
LEN\_OTP\_CTRL, 22  
LEN\_OTP\_RDAT, 22  
LEN\_PANADR, 22  
LEN\_PHR\_MODE\_SUB, 22  
LEN\_PMSC\_CTRL0, 22  
LEN\_RF\_RXCTRLH, 22  
LEN\_RF\_TXCTRL, 22  
LEN\_RX\_BUFFER, 22  
LEN\_RX\_FINFO, 22  
LEN\_RX\_STAMP, 22  
LEN\_RX\_TIME, 22  
LEN\_STAMP, 22  
LEN\_SYS\_CFG, 22  
LEN\_SYS\_CTRL, 22  
LEN\_SYS\_MASK, 22  
LEN\_SYS\_STATUS, 22  
LEN\_SYS\_TIME, 22  
LEN\_TC\_PGDELAY, 22  
LEN\_TX\_ANTD, 22  
LEN\_TX\_BUFFER, 22  
LEN\_TX\_FCTRL, 22  
LEN\_TX\_POWER, 22  
LEN\_TX\_STAMP, 22  
LEN\_TX\_TIME, 22  
LEN\_UWB\_FRAMES, 22  
NO\_SUB, 22  
OTP\_ADDR\_SUB, 23  
OTP\_CTRL\_SUB, 23  
OTP\_IF, 23  
OTP\_RDAT\_SUB, 23  
PANADR, 23  
PHR\_MODE\_SUB, 23  
PMSC, 23  
PMSC\_CTRL0\_SUB, 23  
RF\_CONF, 23  
RF\_RXCTRLH\_SUB, 23  
RF\_TXCTRL\_SUB, 23  
RX\_BUFFER, 23  
RX\_FINFO, 23  
RX\_MODE, 23  
RX\_STAMP\_SUB, 23  
RX\_TIME, 23  
RXAUTR\_BIT, 23  
RXDFR\_BIT, 23  
RXDLYS\_BIT, 23  
RXENAB\_BIT, 23  
RXFCE\_BIT, 23  
RXFCG\_BIT, 23  
RXM110K\_BIT, 23  
RXPHE\_BIT, 23  
RXRFSL\_BIT, 23  
RXRFTO\_BIT, 23  
SFCST\_BIT, 23  
SYS\_CFG, 23  
SYS\_CTRL, 24  
SYS\_MASK, 24  
SYS\_STATUS, 24  
SYS\_TIME, 24  
TC\_PGDELAY\_SUB, 24  
TIME\_RES, 24  
TIME\_RES\_INV, 24  
TRXOFF\_BIT, 24  
TX\_ANTD, 24  
TX\_BUFFER, 24  
TX\_CAL, 24  
TX\_FCTRL, 24  
TX\_MODE, 24  
TX\_POWER, 24  
TX\_STAMP\_SUB, 24  
TX\_TIME, 24  
TXDLYS\_BIT, 24  
TXFRB\_BIT, 24  
TXFRS\_BIT, 24  
TXPHS\_BIT, 24  
TXPRS\_BIT, 24  
TXSTRT\_BIT, 24  
WAIT4RESP\_BIT, 24  
DW1000Class, 5  
attachReceiveErrorHandler, 7  
attachReceiveTimeoutHandler, 7  
attachReceiveTimestampAvailableHandler, 7  
attachReceivedHandler, 7  
attachSentHandler, 7  
begin, 7  
CHANNEL\_1, 12  
CHANNEL\_2, 12  
CHANNEL\_3, 12  
CHANNEL\_4, 12  
CHANNEL\_5, 13  
CHANNEL\_7, 13

- commitConfiguration, 8
- enableMode, 8
- end, 8
- getData, 8
- getDataLength, 8
- getPrettyBytes, 8
- getPrintableDeviceIdentifier, 8
- getPrintableDeviceMode, 8
- getPrintableExtendedUniqueIdentifier, 8
- getPrintableNetworkIdAndShortAddress, 8
- getReceiveTimestamp, 10
- getSystemTimestamp, 10
- getTransmitTimestamp, 10
- idle, 10
- interruptOnAutomaticAcknowledgeTrigger, 10
- interruptOnReceiveError, 10
- interruptOnReceiveTimeout, 10
- interruptOnReceiveTimestampAvailable, 10
- interruptOnReceived, 10
- interruptOnSent, 10
- newConfiguration, 10
- newReceive, 10
- newTransmit, 10
- PAC\_SIZE\_16, 13
- PAC\_SIZE\_32, 13
- PAC\_SIZE\_64, 13
- PAC\_SIZE\_8, 13
- receivePermanently, 10
- reset, 10
- select, 10
- setChannel, 11
- setData, 11
- setDataRate, 11
- setDefaults, 11
- setDelay, 11
- setDeviceAddress, 11
- setInterruptPolarity, 11
- setNetworkId, 11
- setPreambleCode, 11
- setPreambleLength, 11
- setPulseFrequency, 11
- setReceiverAutoReenable, 12
- softReset, 12
- startReceive, 12
- startTransmit, 12
- suppressFrameCheck, 12
- TRX\_RATE\_110KBPS, 14
- TRX\_RATE\_6800KBPS, 14
- TRX\_RATE\_850KBPS, 14
- TX\_PREAMBLE\_LEN\_1024, 14
- TX\_PREAMBLE\_LEN\_128, 14
- TX\_PREAMBLE\_LEN\_1536, 14
- TX\_PREAMBLE\_LEN\_2048, 14
- TX\_PREAMBLE\_LEN\_256, 14
- TX\_PREAMBLE\_LEN\_4096, 14
- TX\_PREAMBLE\_LEN\_512, 14
- TX\_PREAMBLE\_LEN\_64, 14
- TX\_PULSE\_FREQ\_16MHZ, 14
- TX\_PULSE\_FREQ\_64MHZ, 14
- tune, 12
- DW1000Time, 14
  - ~DW1000Time, 15
  - DW1000Time, 15
  - DW1000Time, 15
  - getAsBytes, 15
  - getAsFloat, 15
  - getAsMeters, 15
  - getTimestamp, 16
  - MICROSECONDS, 16
  - MILLISECONDS, 16
  - NANOSECONDS, 16
  - operator\*, 16
  - operator\*=: 16
  - operator+, 16
  - operator+=: 16
  - operator-, 16
  - operator-=, 16
  - operator/, 16
  - operator/=, 16
  - operator=, 16
  - operator==, 16
  - SECONDS, 16
  - setTime, 16
  - setTimestamp, 16
  - TIME\_OVERFLOW, 16
- DW1000Time.h
  - DISTANCE\_OF\_RADIO, 25
  - LEN\_STAMP, 25
  - TIME\_RES, 25
  - TIME\_RES\_INV, 25
- DX\_TIME
  - DW1000.h, 20
- EUI
  - DW1000.h, 20
- enableMode
  - DW1000Class, 8
- end
  - DW1000Class, 8
- FFEN\_BIT
  - DW1000.h, 20
- FRAME\_LENGTH\_NORMAL
  - DW1000Class, 13
- FS\_CTRL
  - DW1000.h, 20
- FS\_PLLCFG\_SUB
  - DW1000.h, 20
- FS\_PLLTUNE\_SUB
  - DW1000.h, 20
- getAsBytes
  - DW1000Time, 15
- getAsFloat
  - DW1000Time, 15
- getAsMeters
  - DW1000Time, 15

getData  
     DW1000Class, 8  
 getDataLength  
     DW1000Class, 8  
 getPrettyBytes  
     DW1000Class, 8  
 getPrintableDeviceIdentifier  
     DW1000Class, 8  
 getPrintableDeviceMode  
     DW1000Class, 8  
 getPrintableExtendedUniquelIdentifier  
     DW1000Class, 8  
 getPrintableNetworkIdAndShortAddress  
     DW1000Class, 8  
 getReceiveTimestamp  
     DW1000Class, 10  
 getSystemTimestamp  
     DW1000Class, 10  
 getTimestamp  
     DW1000Time, 16  
 getTransmitTimestamp  
     DW1000Class, 10  
  
 HIRQ\_POL\_BIT  
     DW1000.h, 21  
  
 IDLE\_MODE  
     DW1000.h, 21  
 idle  
     DW1000Class, 10  
 interruptOnAutomaticAcknowledgeTrigger  
     DW1000Class, 10  
 interruptOnReceiveError  
     DW1000Class, 10  
 interruptOnReceiveTimeout  
     DW1000Class, 10  
 interruptOnReceiveTimestampAvailable  
     DW1000Class, 10  
 interruptOnReceived  
     DW1000Class, 10  
 interruptOnSent  
     DW1000Class, 10  
  
 JUNK  
     DW1000.h, 21  
  
 LDE\_CFG1\_SUB  
     DW1000.h, 21  
 LDE\_CFG2\_SUB  
     DW1000.h, 21  
 LDE\_IF  
     DW1000.h, 21  
 LDE\_REPC\_SUB  
     DW1000.h, 21  
 LDE\_RXANTD\_SUB  
     DW1000.h, 21  
 LDEDONE\_BIT  
     DW1000.h, 21  
 LDEERR\_BIT  
     DW1000.h, 21  
  
 DW1000.h, 21  
 LEN\_AGC\_TUNE1  
     DW1000.h, 21  
 LEN\_AGC\_TUNE2  
     DW1000.h, 21  
 LEN\_AGC\_TUNE3  
     DW1000.h, 21  
 LEN\_CHAN\_CTRL  
     DW1000.h, 21  
 LEN\_DEV\_ID  
     DW1000.h, 21  
 LEN\_DRX\_TUNE0b  
     DW1000.h, 21  
 LEN\_DRX\_TUNE1a  
     DW1000.h, 21  
 LEN\_DRX\_TUNE1b  
     DW1000.h, 21  
 LEN\_DRX\_TUNE2  
     DW1000.h, 21  
 LEN\_DRX\_TUNE4H  
     DW1000.h, 21  
 LEN\_DX\_TIME  
     DW1000.h, 21  
 LEN\_EUI  
     DW1000.h, 21  
 LEN\_EXT\_UWB\_FRAMES  
     DW1000.h, 21  
 LEN\_FS\_PLLCFG  
     DW1000.h, 21  
 LEN\_FS\_PLLTUNE  
     DW1000.h, 21  
 LEN\_LDE\_CFG1  
     DW1000.h, 21  
 LEN\_LDE\_CFG2  
     DW1000.h, 21  
 LEN\_LDE\_REPC  
     DW1000.h, 21  
 LEN\_LDE\_RXANTD  
     DW1000.h, 22  
 LEN\_OTP\_ADDR  
     DW1000.h, 22  
 LEN\_OTP\_CTRL  
     DW1000.h, 22  
 LEN\_OTP\_RDAT  
     DW1000.h, 22  
 LEN\_PANADR  
     DW1000.h, 22  
 LEN\_PHR\_MODE\_SUB  
     DW1000.h, 22  
 LEN\_PMSC\_CTRL0  
     DW1000.h, 22  
 LEN\_RF\_RXCTRLH  
     DW1000.h, 22  
 LEN\_RF\_TXCTRL  
     DW1000.h, 22  
 LEN\_RX\_BUFFER  
     DW1000.h, 22  
 LEN\_RX\_FINFO

- DW1000.h, [22](#)
- LEN\_RX\_STAMP
  - DW1000.h, [22](#)
- LEN\_RX\_TIME
  - DW1000.h, [22](#)
- LEN\_STAMP
  - DW1000.h, [22](#)
  - DW1000Time.h, [25](#)
- LEN\_SYS\_CFG
  - DW1000.h, [22](#)
- LEN\_SYS\_CTRL
  - DW1000.h, [22](#)
- LEN\_SYS\_MASK
  - DW1000.h, [22](#)
- LEN\_SYS\_STATUS
  - DW1000.h, [22](#)
- LEN\_SYS\_TIME
  - DW1000.h, [22](#)
- LEN\_TC\_PGDELAY
  - DW1000.h, [22](#)
- LEN\_TX\_ANTD
  - DW1000.h, [22](#)
- LEN\_TX\_BUFFER
  - DW1000.h, [22](#)
- LEN\_TX\_FCTRL
  - DW1000.h, [22](#)
- LEN\_TX\_POWER
  - DW1000.h, [22](#)
- LEN\_TX\_STAMP
  - DW1000.h, [22](#)
- LEN\_TX\_TIME
  - DW1000.h, [22](#)
- LEN\_UWB\_FRAMES
  - DW1000.h, [22](#)
  
- MICROSECONDS
  - DW1000Time, [16](#)
- MILLISECONDS
  - DW1000Time, [16](#)
  
- NANOSECONDS
  - DW1000Time, [16](#)
- NO\_SUB
  - DW1000.h, [22](#)
- newConfiguration
  - DW1000Class, [10](#)
- newReceive
  - DW1000Class, [10](#)
- newTransmit
  - DW1000Class, [10](#)
  
- OTP\_ADDR\_SUB
  - DW1000.h, [23](#)
- OTP\_CTRL\_SUB
  - DW1000.h, [23](#)
- OTP\_IF
  - DW1000.h, [23](#)
- OTP\_RDAT\_SUB
  - DW1000.h, [23](#)
  
- operator\*
  - DW1000Time, [16](#)
- operator\*=
  - DW1000Time, [16](#)
- operator+
  - DW1000Time, [16](#)
- operator+=
  - DW1000Time, [16](#)
- operator-
  - DW1000Time, [16](#)
- operator-=
  - DW1000Time, [16](#)
- operator/
  - DW1000Time, [16](#)
- operator/=
  - DW1000Time, [16](#)
- operator=
  - DW1000Time, [16](#)
- operator==
  - DW1000Time, [16](#)
  
- PAC\_SIZE\_16
  - DW1000Class, [13](#)
- PAC\_SIZE\_32
  - DW1000Class, [13](#)
- PAC\_SIZE\_64
  - DW1000Class, [13](#)
- PAC\_SIZE\_8
  - DW1000Class, [13](#)
- PANADR
  - DW1000.h, [23](#)
- PHR\_MODE\_SUB
  - DW1000.h, [23](#)
- PMSC
  - DW1000.h, [23](#)
- PMSC\_CTRL0\_SUB
  - DW1000.h, [23](#)
- PREAMBLE\_CODE\_16MHZ\_1
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_2
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_3
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_4
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_5
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_6
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_7
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_16MHZ\_8
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_64MHZ\_10
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_64MHZ\_11
  - DW1000Class, [13](#)
- PREAMBLE\_CODE\_64MHZ\_12
  - DW1000Class, [13](#)



PREAMBLE\_CODE\_64MHZ\_17  
     DW1000Class, 14  
 PREAMBLE\_CODE\_64MHZ\_18  
     DW1000Class, 14  
 PREAMBLE\_CODE\_64MHZ\_19  
     DW1000Class, 14  
 PREAMBLE\_CODE\_64MHZ\_20  
     DW1000Class, 14  
 PREAMBLE\_CODE\_64MHZ\_9  
     DW1000Class, 14  
  
 RF\_CONF  
     DW1000.h, 23  
 RF\_RXCTRLH\_SUB  
     DW1000.h, 23  
 RF\_TXCTRL\_SUB  
     DW1000.h, 23  
 RX\_BUFFER  
     DW1000.h, 23  
 RX\_FINFO  
     DW1000.h, 23  
 RX\_MODE  
     DW1000.h, 23  
 RX\_STAMP\_SUB  
     DW1000.h, 23  
 RX\_TIME  
     DW1000.h, 23  
 RXAUTR\_BIT  
     DW1000.h, 23  
 RXDFR\_BIT  
     DW1000.h, 23  
 RXDLYS\_BIT  
     DW1000.h, 23  
 RXENAB\_BIT  
     DW1000.h, 23  
 RXFCE\_BIT  
     DW1000.h, 23  
 RXFCG\_BIT  
     DW1000.h, 23  
 RXM110K\_BIT  
     DW1000.h, 23  
 RXPHE\_BIT  
     DW1000.h, 23  
 RXRFSL\_BIT  
     DW1000.h, 23  
 RXRFTO\_BIT  
     DW1000.h, 23  
 receivePermanently  
     DW1000Class, 10  
 reset  
     DW1000Class, 10  
  
 SECONDS  
     DW1000Time, 16  
 SFCST\_BIT  
     DW1000.h, 23  
 SYS\_CFG  
     DW1000.h, 23  
 SYS\_CTRL  
     DW1000.h, 24  
     SYS\_MASK  
         DW1000.h, 24  
     SYS\_STATUS  
         DW1000.h, 24  
     SYS\_TIME  
         DW1000.h, 24  
     select  
         DW1000Class, 10  
     setChannel  
         DW1000Class, 11  
     setData  
         DW1000Class, 11  
     setDataRate  
         DW1000Class, 11  
     setDefault  
         DW1000Class, 11  
     setDelay  
         DW1000Class, 11  
     setDeviceAddress  
         DW1000Class, 11  
     setInterruptPolarity  
         DW1000Class, 11  
     setNetworkId  
         DW1000Class, 11  
     setPreambleCode  
         DW1000Class, 11  
     setPreambleLength  
         DW1000Class, 11  
     setPulseFrequency  
         DW1000Class, 11  
     setReceiverAutoReenable  
         DW1000Class, 12  
     setTime  
         DW1000Time, 16  
     setTimestamp  
         DW1000Time, 16  
     softReset  
         DW1000Class, 12  
     startReceive  
         DW1000Class, 12  
     startTransmit  
         DW1000Class, 12  
     suppressFrameCheck  
         DW1000Class, 12  
  
 TC\_PGDELAY\_SUB  
     DW1000.h, 24  
 TIME\_OVERFLOW  
     DW1000Time, 16  
 TIME\_RES  
     DW1000.h, 24  
     DW1000Time.h, 25  
 TIME\_RES\_INV  
     DW1000.h, 24  
     DW1000Time.h, 25  
 TRX\_RATE\_110KBPS  
     DW1000Class, 14  
 TRX\_RATE\_6800KBPS

DW1000Class, [14](#)  
TRX\_RATE\_850KBPS  
  DW1000Class, [14](#)  
TRXOFF\_BIT  
  DW1000.h, [24](#)  
TX\_ANTD  
  DW1000.h, [24](#)  
TX\_BUFFER  
  DW1000.h, [24](#)  
TX\_CAL  
  DW1000.h, [24](#)  
TX\_FCTRL  
  DW1000.h, [24](#)  
TX\_MODE  
  DW1000.h, [24](#)  
TX\_POWER  
  DW1000.h, [24](#)  
TX\_PREAMBLE\_LEN\_1024  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_128  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_1536  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_2048  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_256  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_4096  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_512  
  DW1000Class, [14](#)  
TX\_PREAMBLE\_LEN\_64  
  DW1000Class, [14](#)  
TX\_PULSE\_FREQ\_16MHZ  
  DW1000Class, [14](#)  
TX\_PULSE\_FREQ\_64MHZ  
  DW1000Class, [14](#)  
TX\_STAMP\_SUB  
  DW1000.h, [24](#)  
TX\_TIME  
  DW1000.h, [24](#)  
TXDLYS\_BIT  
  DW1000.h, [24](#)  
TXFRB\_BIT  
  DW1000.h, [24](#)  
TXFRS\_BIT  
  DW1000.h, [24](#)  
TXPHS\_BIT  
  DW1000.h, [24](#)  
TXPRS\_BIT  
  DW1000.h, [24](#)  
TXSTRT\_BIT  
  DW1000.h, [24](#)  
tune  
  DW1000Class, [12](#)  
  
WAIT4RESP\_BIT  
  DW1000.h, [24](#)