

Watering System

1.0.0

Generated by Doxygen 1.8.16

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 buffer.c File Reference	3
2.1.1 Detailed Description	4
2.1.2 Function Documentation	4
2.1.2.1 getAvg()	4
2.1.2.2 getCount()	5
2.1.2.3 getState()	5
2.1.2.4 initBuffer()	6
2.1.2.5 putVal()	6
2.1.2.6 setState()	6
2.2 display.c File Reference	7
2.2.1 Detailed Description	8
2.2.2 Function Documentation	8
2.2.2.1 initDisplay()	8
2.2.2.2 lcd_cmd()	9
2.2.2.3 lcd_init()	9
2.2.2.4 lcd_printChar()	10
2.2.2.5 lcd_printStr()	10
2.2.2.6 lcd_setCursor()	10
2.2.2.7 logDisplayMessage()	11
2.2.2.8 processDisplayMessages()	12
2.3 main.c File Reference	13
2.3.1 Detailed Description	13
2.3.2 Function Documentation	13
2.3.2.1 main()	14
2.3.2.2 setup()	14
2.4 sensor.c File Reference	15
2.4.1 Detailed Description	16
2.4.2 Function Documentation	16
2.4.2.1 initSensor()	16
2.4.2.2 LogSensorMessage()	16
2.4.2.3 processSensorMessages()	17
2.5 serial.c File Reference	17
2.5.1 Detailed Description	18
2.5.2 Function Documentation	18
2.5.2.1 getLogger()	18
2.5.2.2 initSerial()	19
2.5.2.3 processSerialMessages()	19
2.6 watersystem.c File Reference	20

2.6.1 Detailed Description	20
2.6.2 Function Documentation	20
2.6.2.1 initWaterSystem()	21
2.6.2.2 LogMessage()	21
2.6.2.3 ProcessMessages()	22
Index	23

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

buffer.c	Buffer class	3
buffer.h	??
defs.h	??
display.c	Display class	7
display.h	??
main.c	Main class	13
motor.c	??
motor.h	??
sensor.c	Sensor class	15
sensor.h	??
serial.c	Serial class	17
serial.h	??
support.h	??
time.c	??
time.h	??
watersystem.c	Watersystem class	20
watersystem.h	??

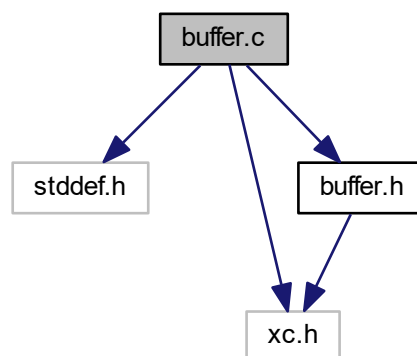
Chapter 2

File Documentation

2.1 buffer.c File Reference

Buffer class.

```
#include <stddef.h>
#include "xc.h"
#include "buffer.h"
Include dependency graph for buffer.c:
```



Macros

- `#define BUFFER_SIZE 100`

Functions

- void `putVal` (int newValue)
Puts a value.
- int `getAvg` ()
Gets the average.
- int `getCount` ()
Gets the count.
- int `getState` (void)
Gets the state.
- void `setState` (int level)
Sets a state.
- void `initBuffer` ()
Initializes the buffer.

Variables

- volatile int `theBuffer` [BUFFER_SIZE]
the buffer[buffer size]
- volatile int `bufferFront`
The buffer front.
- int `buffSize` = 0
Size of the buffer.
- int `state` = 0
The state.

2.1.1 Detailed Description

Buffer class.

2.1.2 Function Documentation

2.1.2.1 `getAvg()`

```
int getAvg ( )
```

Gets the average.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

The calculated average.

Definition at line 53 of file buffer.c.

2.1.2.2 getCount()

```
int getCount ( )
```

Gets the count.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

The count.

Definition at line 87 of file buffer.c.

2.1.2.3 getState()

```
int getState (
    void )
```

Gets the state.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/8/2019

Returns

The state.

Definition at line 103 of file buffer.c.

2.1.2.4 initBuffer()

```
void initBuffer ( )
```

Initializes the buffer.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Definition at line 131 of file buffer.c.

2.1.2.5 putVal()

```
void putVal (
    int newValue )
```

Puts a value.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>newValue</i>	The new value.
-----------------	----------------

Definition at line 33 of file buffer.c.

2.1.2.6 setState()

```
void setState (
    int level )
```

Sets a state.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/8/2019

Parameters

<i>level</i>	The level.
--------------	------------

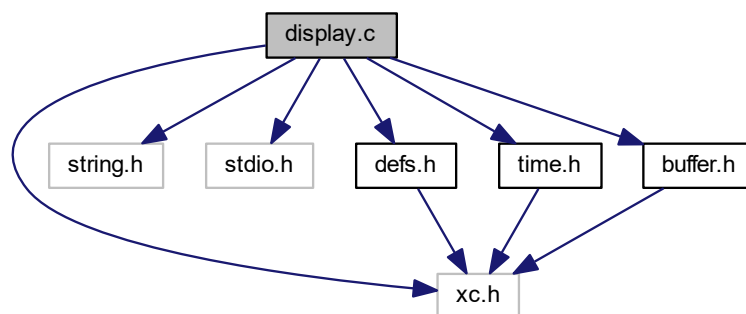
Definition at line 118 of file buffer.c.

2.2 display.c File Reference

Display class.

```
#include "xc.h"
#include <string.h>
#include <stdio.h>
#include "defs.h"
#include "time.h"
#include "buffer.h"
```

Include dependency graph for display.c:



Functions

- void [logDisplayMessage](#) (char *message)
Private functions.
- void [__attribute__](#) ((__interrupt__, __auto_psv__))
- void [lcd_cmd](#) (char command)
LCD command.
- void [lcd_init](#) (void)

- LCD initialize.*
 - void `lcd_setCursor` (char x, char y)
- LCD set cursor.*
 - void `lcd_printChar` (char myChar)
- LCD print character.*
 - void `lcd_printStr` (const char *s)
- LCD print string.*
 - int `initDisplay` (loggerCallback logger)
- Initializes the display.*
 - int `processDisplayMessages` ()
- Process the display messages.*

Variables

- loggerCallback `displayLogger` = NULL

2.2.1 Detailed Description

Display class.

2.2.2 Function Documentation

2.2.2.1 `initDisplay()`

```
int initDisplay (  
    loggerCallback logger )
```

Initializes the display.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>logger</i>	The logger.
---------------	-------------

Returns

An int.

Definition at line 237 of file display.c.

2.2.2.2 lcd_cmd()

```
void lcd_cmd (  
    char command )
```

LCD command.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>command</i>	The command.
----------------	--------------

Definition at line 57 of file display.c.

2.2.2.3 lcd_init()

```
void lcd_init (  
    void )
```

LCD initialize.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Definition at line 91 of file display.c.

Here is the call graph for this function:



2.2.2.4 lcd_printChar()

```
void lcd_printChar (
    char myChar )
```

LCD print character.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>myChar</i>	my character.
---------------	---------------

Definition at line 146 of file display.c.

2.2.2.5 lcd_printStr()

```
void lcd_printStr (
    const char * s )
```

LCD print string.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>s</i>	A char to process.
----------	--------------------

Definition at line 180 of file display.c.

2.2.2.6 lcd_setCursor()

```
void lcd_setCursor (
```

```
char x,  
char y )
```

LCD set cursor.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>x</i>	A char to process.
<i>y</i>	A char to process.

Definition at line 121 of file display.c.

Here is the call graph for this function:

**2.2.2.7 logDisplayMessage()**

```
void logDisplayMessage (  
    char * message )
```

Private functions.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<code>in, out</code>	<code>message</code>	If non-null, the message.
----------------------	----------------------	---------------------------

Definition at line 27 of file display.c.

2.2.2.8 processDisplayMessages()

```
int processDisplayMessages ( )
```

Process the display messages.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

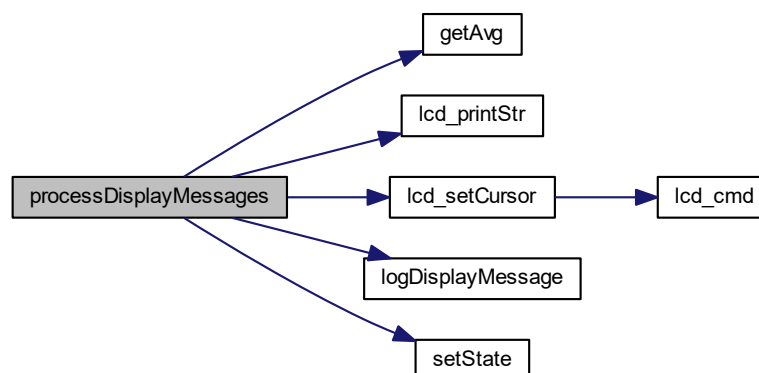
12/1/2019

Returns

An int.

Definition at line 276 of file display.c.

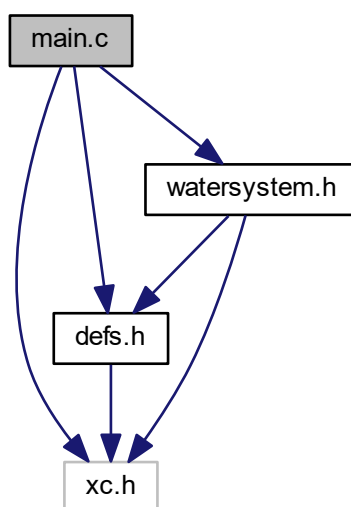
Here is the call graph for this function:



2.3 main.c File Reference

Main class.

```
#include "xc.h"  
#include "defs.h"  
#include "watersystem.h"  
Include dependency graph for main.c:
```



Functions

- void `setup` (void)
Setups this object.
- int `main` (void)
Main entry-point for this application.

2.3.1 Detailed Description

Main class.

2.3.2 Function Documentation

2.3.2.1 main()

```
int main (
    void )
```

Main entry-point for this application.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

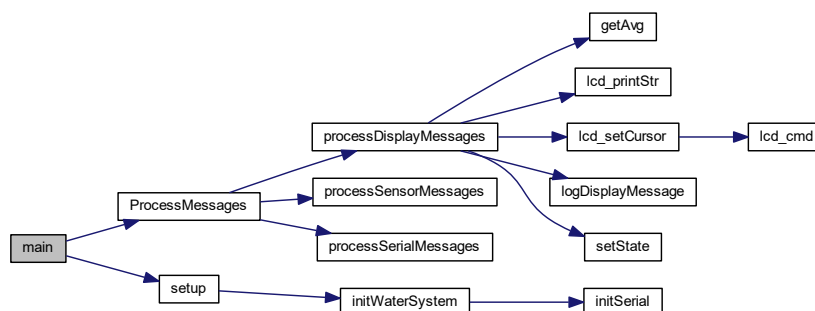
12/1/2019

Returns

Exit-code for the process - 0 for success, else an error code.

Definition at line 53 of file main.c.

Here is the call graph for this function:



2.3.2.2 setup()

```
void setup (
    void )
```

Setups this object.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Definition at line 34 of file main.c.

Here is the call graph for this function:

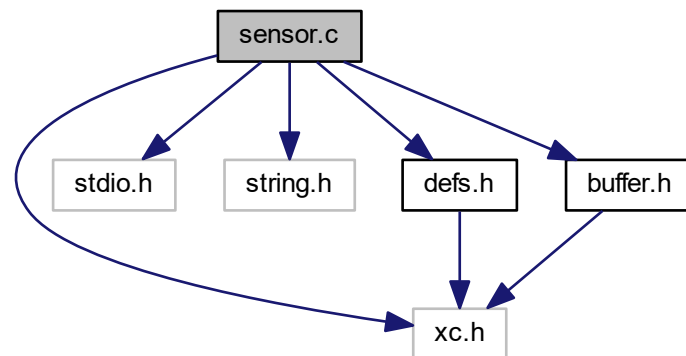


2.4 sensor.c File Reference

Sensor class.

```
#include "xc.h"
#include <stdio.h>
#include <string.h>
#include "defs.h"
#include "buffer.h"
```

Include dependency graph for sensor.c:



Functions

- void [LogSensorMessage](#) (char *message)
Logs sensor message.
- void [__attribute__](#) ((__interrupt__, __auto_psv__))
- int [initSensor](#) (loggerCallback logger)
Initializes the sensor.
- int [processSensorMessages](#) ()
Process the sensor messages.

Variables

- int `lowThreshold` = 6
The low threshold.
- int `highThreshold` = 9
The high threshold.
- loggerCallback `sensorLogger` = NULL

2.4.1 Detailed Description

Sensor class.

2.4.2 Function Documentation

2.4.2.1 `initSensor()`

```
int initSensor (  
    loggerCallback logger )
```

Initializes the sensor.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>logger</i>	The logger.
---------------	-------------

Returns

An int.

Definition at line 69 of file sensor.c.

2.4.2.2 `LogSensorMessage()`

```
void LogSensorMessage (  
    char * message )
```

Logs sensor message.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<i>in, out</i>	<i>message</i>	If non-null, the message.
----------------	----------------	---------------------------

Definition at line 34 of file sensor.c.

2.4.2.3 processSensorMessages()

```
int processSensorMessages ( )
```

Process the sensor messages.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

An int.

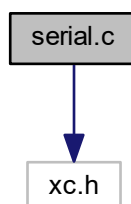
Definition at line 108 of file sensor.c.

2.5 serial.c File Reference

Serial class.

```
#include "xc.h"
```

Include dependency graph for serial.c:



Functions

- int `initSerial` (void)
Initializes the serial.
- void * `getLogger` (void)
Gets the logger.
- int `processSerialMessages` ()
Process the serial messages - Nothing to do here.

2.5.1 Detailed Description

Serial class.

2.5.2 Function Documentation

2.5.2.1 `getLogger()`

```
void * getLogger (  
    void )
```

Gets the logger.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

Null if it fails, else the logger.

Definition at line 90 of file serial.c.

2.5.2.2 initSerial()

```
int initSerial (
    void )
```

Initializes the serial.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

An int.

Definition at line 58 of file serial.c.

2.5.2.3 processSerialMessages()

```
int processSerialMessages ( )
```

Process the serial messages - Nothing to do here.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

An int.

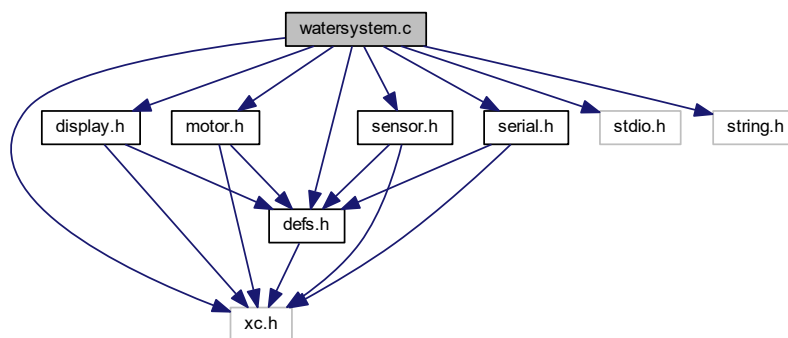
Definition at line 105 of file serial.c.

2.6 watersystem.c File Reference

Watersystem class.

```
#include "xc.h"
#include "display.h"
#include "motor.h"
#include "sensor.h"
#include "serial.h"
#include "defs.h"
#include "stdio.h"
#include "string.h"
```

Include dependency graph for watersystem.c:



Functions

- void [LogMessage](#) (char *message)
Logs a message.
- int [initWaterSystem](#) (void)
Initializes the water system.
- int [ProcessMessages](#) ()
This is the eternal pump for the Watersystem, can be used for the eternal loop.

Variables

- loggerCallback **loggerFunction** = NULL

2.6.1 Detailed Description

Watersystem class.

2.6.2 Function Documentation

2.6.2.1 initWaterSystem()

```
int initWaterSystem (  
    void )
```

Initializes the water system.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Returns

An int.

Definition at line 45 of file watersystem.c.

Here is the call graph for this function:



2.6.2.2 LogMessage()

```
void LogMessage (  
    char * message )
```

Logs a message.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Parameters

<code>in, out</code>	<code>message</code>	If non-null, the message.
----------------------	----------------------	---------------------------

Definition at line 29 of file watersystem.c.

2.6.2.3 ProcessMessages()

```
int ProcessMessages ( )
```

This is the eternal pump for the Watersystem, can be used for the eternal loop.

Author

Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

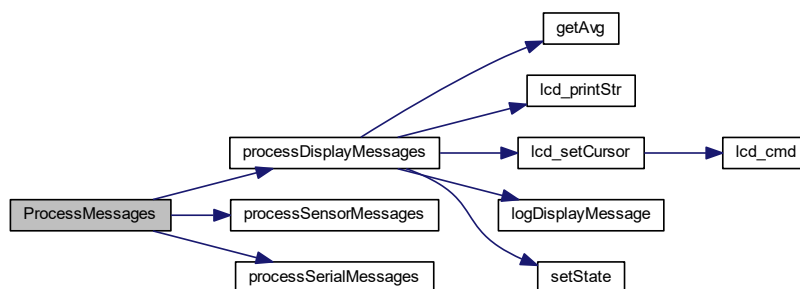
12/1/2019

Returns

An int.

Definition at line 81 of file watersystem.c.

Here is the call graph for this function:



Index

- buffer.c, [3](#)
 - getAvg, [4](#)
 - getCount, [4](#)
 - getState, [5](#)
 - initBuffer, [5](#)
 - putVal, [6](#)
 - setState, [6](#)
- display.c, [7](#)
 - initDisplay, [8](#)
 - lcd_cmd, [9](#)
 - lcd_init, [9](#)
 - lcd_printChar, [9](#)
 - lcd_printStr, [10](#)
 - lcd_setCursor, [10](#)
 - logDisplayMessage, [11](#)
 - processDisplayMessages, [12](#)
- getAvg
 - buffer.c, [4](#)
- getCount
 - buffer.c, [4](#)
- getLogger
 - serial.c, [18](#)
- getState
 - buffer.c, [5](#)
- initBuffer
 - buffer.c, [5](#)
- initDisplay
 - display.c, [8](#)
- initSensor
 - sensor.c, [16](#)
- initSerial
 - serial.c, [18](#)
- initWaterSystem
 - watersystem.c, [20](#)
- lcd_cmd
 - display.c, [9](#)
- lcd_init
 - display.c, [9](#)
- lcd_printChar
 - display.c, [9](#)
- lcd_printStr
 - display.c, [10](#)
- lcd_setCursor
 - display.c, [10](#)
- logDisplayMessage
 - display.c, [11](#)
- LogMessage
 - watersystem.c, [21](#)
- LogSensorMessage
 - sensor.c, [16](#)
- main
 - main.c, [13](#)
- main.c, [13](#)
 - main, [13](#)
 - setup, [14](#)
- processDisplayMessages
 - display.c, [12](#)
- ProcessMessages
 - watersystem.c, [22](#)
- processSensorMessages
 - sensor.c, [17](#)
- processSerialMessages
 - serial.c, [19](#)
- putVal
 - buffer.c, [6](#)
- sensor.c, [15](#)
 - initSensor, [16](#)
 - LogSensorMessage, [16](#)
 - processSensorMessages, [17](#)
- serial.c, [17](#)
 - getLogger, [18](#)
 - initSerial, [18](#)
 - processSerialMessages, [19](#)
- setState
 - buffer.c, [6](#)
- setup
 - main.c, [14](#)
- watersystem.c, [20](#)
 - initWaterSystem, [20](#)
 - LogMessage, [21](#)
 - ProcessMessages, [22](#)