Watering System

1.0.0

Generated by Doxygen 1.8.16

Chapter 1

File Index

1.1 File List

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

Chapter 2

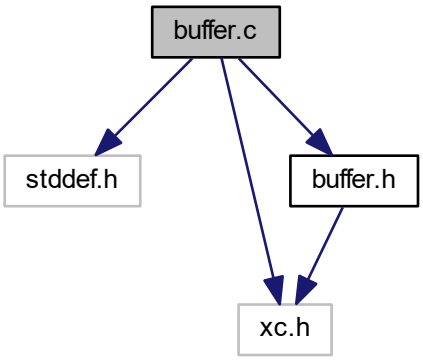
File Documentation

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

newValue	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

level	The level.

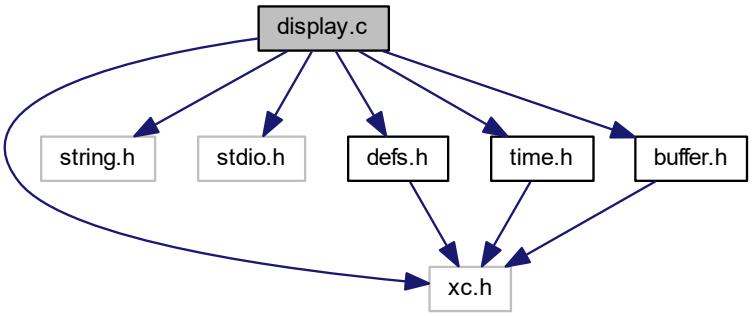
Definition at line 118 of file buffer.c.

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

logger	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

command	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

myChar	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

s	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

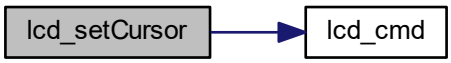
12/1/2019

Parameters

x	A char to process.
y	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

in,out	*message*	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

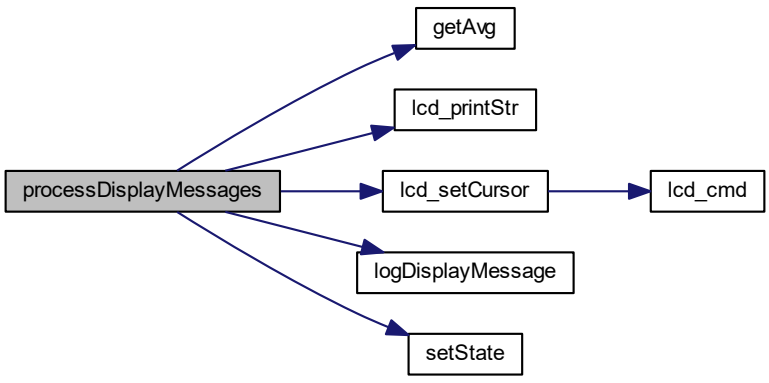
12/1/2019

Returns

An int.

Definition at line 276 of file display.c.

Here is the call graph for this function:

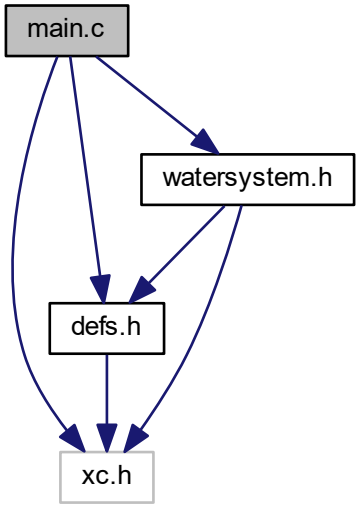


2.3 main.c File Reference

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

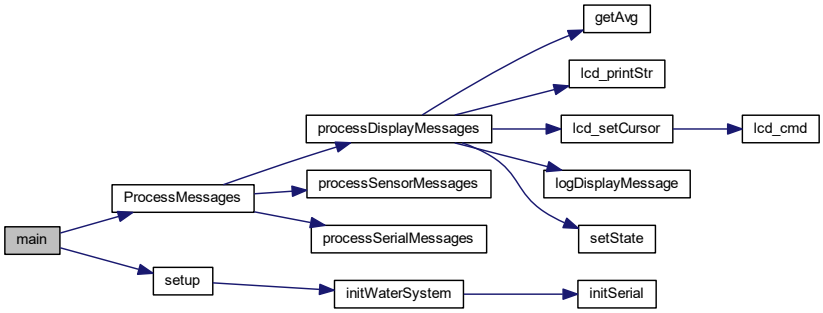
12/1/2019

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

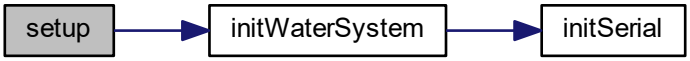
Tyler Krussow, Alex Lema, Darnell Otterson, and Jacob Wenthe

Date

12/1/2019

Definition at line 34 of file main.c.

Here is the call graph for this function:

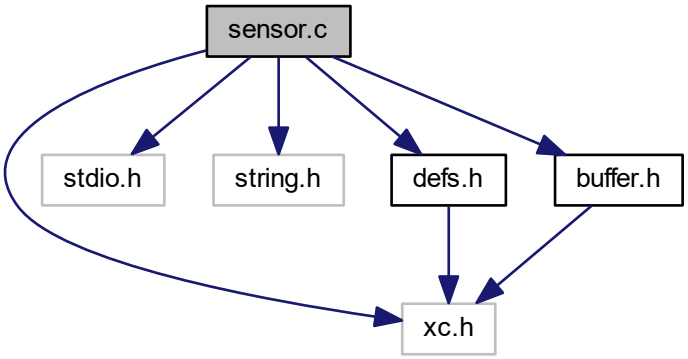


2.4 sensor.c File Reference

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 t hreshold.
- 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

logger	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

in,out	*message*	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.

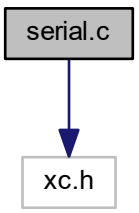
Definition at line 108 of file sensor.c.

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.

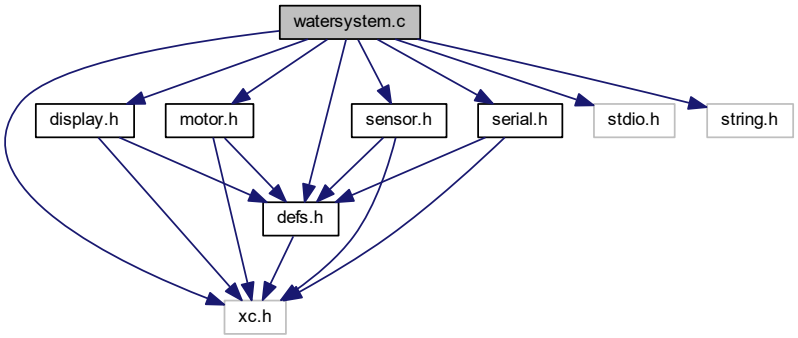
Definition at line 105 of file serial.c.

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

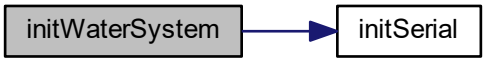
12/1/2019

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

in,out	*message*	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

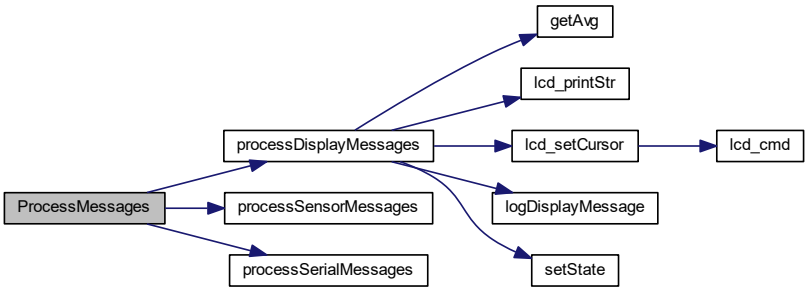
12/1/2019

Returns

An int.

Definition at line 81 of file watersystem.c.

Here is the call graph for this function:



Index