AqaServer

Generated by Doxygen 1.8.7

Wed May 4 2016 09:29:31

Contents

Chapter 1

Main Page

Small app for Arduino Uno for aquarium management.

It's suppose to manage:

- water level,
- temperature in tank,
- time to sleep/morning,
- feeding.

Scheme

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Tmr Class Reference

Timer class for containg timers like wake time or sleep time.

```
#include <timers.h>
```

Public Attributes

- byte h

  Hour of timer.
- byte m

  Minute of timer.
- byte done

  Flag if time was set.

4.1.1 Detailed Description

Timer class for containg timers like wake time or sleep time.

Definition at line 9 of file timers.h.

4.1.2 Member Data Documentation

4.1.2.1 byte Tmr::done

Flag if time was set.

Definition at line 12 of file timers.h.

4.1.2.2 byte Tmr::h

Hour of timer.

Definition at line 10 of file timers.h.

4.1.2.3 byte Tmr::m

Minute of timer.

Definition at line 11 of file timers.h.

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

- timers.h

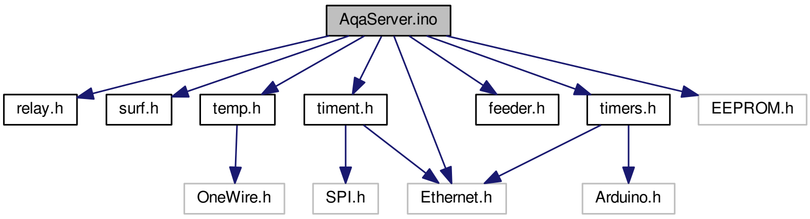
Chapter 5

File Documentation

5.1 AqaServer.ino File Reference

```
#include "relay.h"
#include "surf.h"
#include "temp.h"
#include "timent.h"
#include "feeder.h"
#include "timers.h"
#include <EEPROM.h>
#include <Ethernet.h>
```

Include dependency graph for AqaServer.ino:



Macros

- #define BFS 10

  Size of input buffer.

Functions

- EthernetServer server (9011)

  Server instance for port 9011.
- void setup ()

  Setting up all the inputs/outputs.
- void loop ()

  Looping the Arduino.

Variables

- byte gateway [] = { 10, 0, 0, 138 }

  IP adress of gateway.
- byte subnet [] = { 255, 255, 255, 0}

  Network mask adress of Arduino.

5.1.1 Macro Definition Documentation

5.1.1.1 #define BFS 10

Size of input buffer.

Definition at line 31 of file AqaServer.ino.

5.1.2 Function Documentation

5.1.2.1 loop ( )

Looping the Arduino.

Definition at line 73 of file AqaServer.ino.

5.1.2.2 EthernetServer server (9011)

Server instance for port 9011.

5.1.2.3 setup ( )

Setting up all the inputs/outputs.

Definition at line 48 of file AqaServer.ino.

5.1.3 Variable Documentation

5.1.3.1 byte gateway[ ] = { 10, 0, 0, 138 }

IP adress of gateway.

Definition at line 39 of file AqaServer.ino.

5.1.3.2 byte subnet[ ] = { 255, 255, 255, 0}

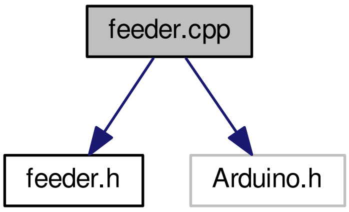
Network mask adress of Arduino.

Definition at line 40 of file AqaServer.ino.

5.2 feeder.cpp File Reference

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

Include dependency graph for feeder.cpp:



Functions

- void initFeed ()

  Step motor initiation.

- void turnAround ()

  Turning step motor one time around.

5.2.1 Function Documentation

5.2.1.1 initFeed ( )

Step motor initiation.

Definition at line 8 of file feeder.cpp.

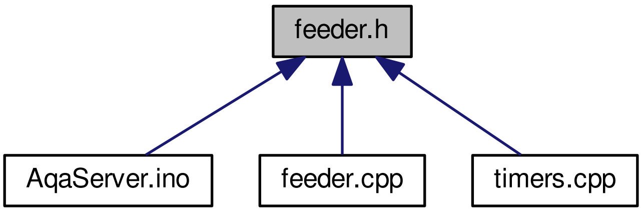
5.2.1.2 turnAround ( )

Turning step motor one time around.

Definition at line 16 of file feeder.cpp.

5.3 feeder.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define motorPin1 6

  Pin number of step motor.
- #define motorPin2 7

  Pin number of step motor.
- #define motorPin3 8

  Pin number of step motor.
- #define motorPin4 9

  Pin number of step motor.
- #define delayTime 10

  Time delay at turning.

Functions

- void initFeed ()

  Step motor initiation.
- void turnAround ()

  Turning step motor one time around.

5.3.1 Macro Definition Documentation

5.3.1.1 #define delayTime 10

Time delay at turning.

Definition at line 7 of file feeder.h.

5.3.1.2 #define motorPin1 6

Pin number of step motor.

Definition at line 3 of file feeder.h.

5.3.1.3 #define motorPin2 7

Pin number of step motor.

Definition at line 4 of file feeder.h.

5.3.1.4 #define motorPin3 8

Pin number of step motor.

Definition at line 5 of file feeder.h.

5.3.1.5 #define motorPin4 9

Pin number of step motor.

Definition at line 6 of file feeder.h.

5.3.2 Function Documentation

5.3.2.1 void initFeed ( )

Step motor initiation.

Definition at line 8 of file feeder.cpp.

5.3.2.2 void turnAround ( )

Turning step motor one time around.

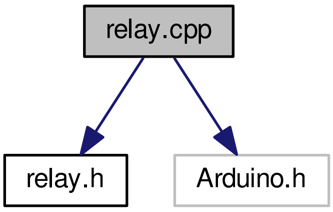
Definition at line 16 of file feeder.cpp.

5.4 README.md File Reference

5.5 relay.cpp File Reference

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

Include dependency graph for relay.cpp:



Functions

- void initRelays ()

  Relay initiation.
- void switchRelay (int x)

  Switch relay given by x to different state.
- void switchSet (int x, int state)

  Set the realy state by given x.
- void switchOn (int x)

  Switch off relay state by the given x.
- void switchOff (int x)

  Switch off relay state by the given x.
- int switchState (int x)

  Getter for relay state by the given x.

Variables

- int relays_stats [RELAY_CNT]

  Relay status array.

5.5.1 Function Documentation

5.5.1.1 initRelays ( )

Relay initiation.

Definition at line 9 of file relay.cpp.

5.5.1.2 switchOff (int *x*)

Switch off relay state by the given x.

Parameters

x	Number of relay.

Definition at line 32 of file relay.cpp.

5.5.1.3 switchOn (int *x*)

Switch off relay state by the given x.

Parameters

x	Number of relay.

Definition at line 26 of file relay.cpp.

5.5.1.4 switchRelay (int *x*)

Switch relay given by x to different state.

Parameters

x	Number of relay

Definition at line 18 of file relay.cpp.

5.5.1.5 switchSet (int *x*, int *state*)

Set the realy state by given x.

Parameters

x	Number of relay.
state	New state of relay.

Definition at line 22 of file relay.cpp.

5.5.1.6 switchState (int *x*)

Getter for relay state by the given x.

Parameters

x	Number of relay.

Returns

State of relay.

Definition at line 38 of file relay.cpp.

5.5.2 Variable Documentation

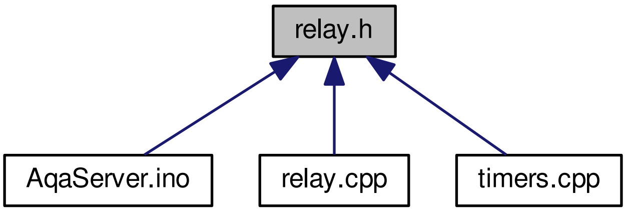
5.5.2.1 int relays_stats[RELAY_CNT]

Relay status array.

Definition at line 4 of file relay.cpp.

5.6 relay.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define RELAY_CNT 3

  Number of relays.
- #define LIGHTS 3

  Light pin relay.
- #define FILTER 4

  Filter pin relay.
- #define HEATER 5

  Heater pin relay.

Functions

- void initRelays ()

  Relay initiation.
- void switchRelay (int x)

  Switch relay given by x to different state.
- void switchSet (int x, int state)

  Set the realy state by given x.
- void switchOn (int x)

  Switch off relay state by the given x.
- void switchOff (int x)

  Switch off relay state by the given x.
- int switchState (int x)

  Getter for relay state by the given x.

5.6.1 Macro Definition Documentation

5.6.1.1 #define FILTER 4

Filter pin relay.

Definition at line 5 of file relay.h.

5.6.1.2 #define HEATER 5

Heater pin relay.

Definition at line 6 of file relay.h.

5.6.1.3 #define LIGHTS 3

Light pin relay.

Definition at line 4 of file relay.h.

5.6.1.4 #define RELAY_CNT 3

Number of relays.

Definition at line 3 of file relay.h.

5.6.2 Function Documentation

5.6.2.1 void initRelays ( )

Relay initiation.

Definition at line 9 of file relay.cpp.

5.6.2.2 void switchOff (int *x*)

Switch off relay state by the given x.

Parameters

x	Number of relay.

Definition at line 32 of file relay.cpp.

5.6.2.3 void switchOn (int *x*)

Switch off relay state by the given x.

Parameters

x	Number of relay.

Definition at line 26 of file relay.cpp.

5.6.2.4 void switchRelay (int *x*)

Switch relay given by x to different state.

Parameters

x	Number of relay

Definition at line 18 of file relay.cpp.

5.6.2.5 void switchSet (int *x*, int *state*)

Set the realy state by given x.

Parameters

x	Number of relay.
state	New state of relay.

Definition at line 22 of file relay.cpp.

5.6.2.6 int switchState (int *x*)

Getter for relay state by the given x.

Parameters

x	Number of relay.

Returns

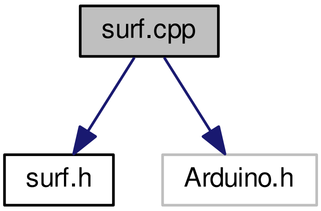
State of relay.

Definition at line 38 of file relay.cpp.

5.7 surf.cpp File Reference

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

Include dependency graph for surf.cpp:



Functions

- void initSurface ()

  Relay initiation.
- int getSurface ()

  Relay status getter.

5.7.1 Function Documentation

5.7.1.1 getSurface ( )

Relay status getter.

Returns

The state of switch.

Definition at line 11 of file surf.cpp.

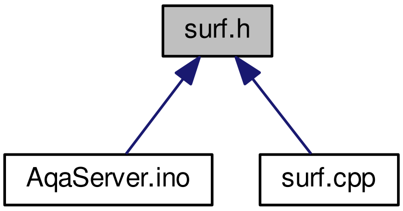
5.7.1.2 initSurface ( )

Relay initiation.

Definition at line 7 of file surf.cpp.

5.8 surf.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define SWITCH_SUR A0

  Surface switch input pin.

Functions

- void initSurface ()

  Relay initiation.
- int getSurface ()

  Relay status getter.

5.8.1 Macro Definition Documentation

5.8.1.1 #define SWITCH_SUR A0

Surface switch input pin.

Definition at line 3 of file surf.h.

5.8.2 Function Documentation

5.8.2.1 int getSurface ( )

Relay status getter.

Returns

The state of switch.

Definition at line 11 of file surf.cpp.

5.8.2.2 void initSurface ( )

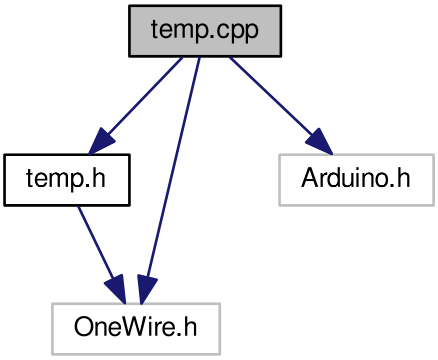
Relay initiation.

Definition at line 7 of file surf.cpp.

5.9 temp.cpp File Reference

```
#include "temp.h"
#include <OneWire.h>
#include <Arduino.h>
```

Include dependency graph for temp.cpp:



Functions

- OneWire ds (DS18S20_Pin)
- float getTemp ()

  Temperature getter.

Variables

- int DS18S20_Pin = 2

  Pin of RTC.

5.9.1 Function Documentation

5.9.1.1 OneWire ds (DS18S20_Pin)

5.9.1.2 getTemp ( )

Temperature getter.

Returns

Temperature of aqarium.

Definition at line 11 of file temp.cpp.

5.9.2 Variable Documentation

5.9.2.1 int DS18S20_Pin = 2

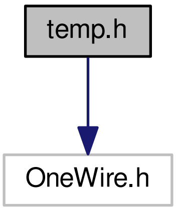
Pin of RTC.

Definition at line 5 of file temp.cpp.

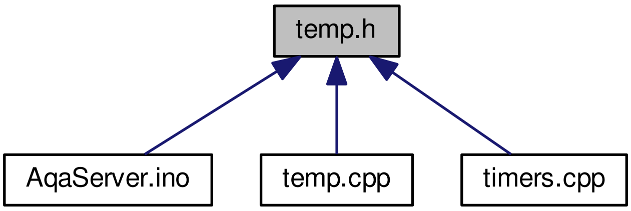
5.10 temp.h File Reference

```
#include <OneWire.h>
```

Include dependency graph for temp.h:



This graph shows which files directly or indirectly include this file:



Functions

- float getTemp ()

  Temperature getter.

5.10.1 Function Documentation

5.10.1.1 float getTemp ( )

Temperature getter.

Returns

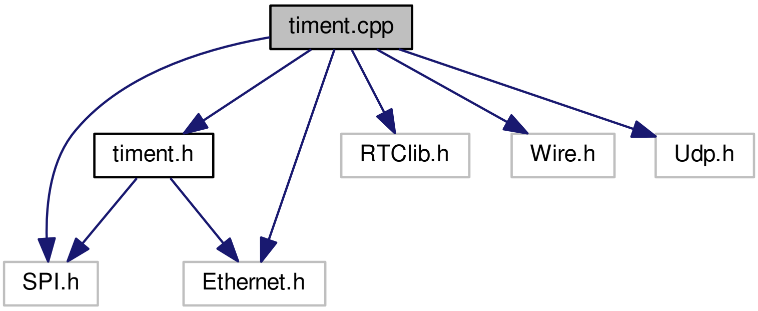
Temperature of aqarium.

Definition at line 11 of file temp.cpp.

5.11 timent.cpp File Reference

```
#include "timent.h"
#include <SPI.h>
#include <Ethernet.h>
#include <RTClib.h>
#include "Wire.h"
#include <Udp.h>
```

Include dependency graph for timent.cpp:



Macros

- #define DS3231_I2C_ADDRESS 0x68

Functions

- void initTiment ()

  Init of RTC.
- byte decToBcd (byte val)

  Decadic format to binary coded decimal.
- byte bcdToDec (byte val)

  Binary coded decimal to decadic format.
- void setDS3231time (byte second, byte minute, byte hour, byte dayOfWeek, byte dayOfMonth, byte month, byte year)

  Setter for RTC.

- void readDS3231MinutesHoures (byte ∗minute, byte ∗hour)
- void readDS3231time (byte ∗second, byte ∗minute, byte ∗hour, byte ∗dayOfWeek, byte ∗dayOfMonth, byte ∗month, byte ∗year)
- void displayHttpTime (EthernetClient ∗client)

  Response for ethernet connection for time values.
- void displayTime ()

  Displays time on the console.
- unsigned long sendNTPpacket (byte ∗address)

  Sender of NTP request to NTP server given by address.
- boolean setTime ()

  Set time by NTP server to RTC.

Variables

- byte timeServer [ ] = {10, 0, 0, 4}

  Remote NTP server.
- unsigned int localPort = 31011

  Local TCP connection port.
- const int NTP_PACKET_SIZE = 48
- byte pb [NTP_PACKET_SIZE]
- EthernetUDP Udp

5.11.1 Macro Definition Documentation

5.11.1.1 #define DS3231_I2C_ADDRESS 0x68

Definition at line 6 of file timent.cpp.

5.11.2 Function Documentation

5.11.2.1 bcdToDec (byte *val*)

Binary coded decimal to decadic format.

Parameters

val	Binary coded number.

Returns

Decimal value.

Definition at line 32 of file timent.cpp.

5.11.2.2 decToBcd (byte *val*)

Decadic format to binary coded decimal.

Parameters

val	Decadic number

Returns

Binary coded decimal value.

Definition at line 27 of file timent.cpp.

5.11.2.3 displayHttpTime (EthernetClient ∗ *client*)

Response for ethernet connection for time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 75 of file timent.cpp.

5.11.2.4 displayTime ( )

Displays time on the console.

Definition at line 124 of file timent.cpp.

5.11.2.5 initTiment ( )

Init of RTC.

Definition at line 23 of file timent.cpp.

5.11.2.6 void readDS3231MinutesHoures (byte ∗ *minute,* byte ∗ *hour*)

Definition at line 49 of file timent.cpp.

5.11.2.7 void readDS3231time (byte ∗ *second,* byte ∗ *minute,* byte ∗ *hour,* byte ∗ *dayOfWeek,* byte ∗ *dayOfMonth,* byte ∗ *month,* byte ∗ *year*)

Definition at line 60 of file timent.cpp.

5.11.2.8 sendNTPpacket (byte ∗ *address*)

Sender of NTP request to NTP server given by address.

Parameters

address	Address of NTP server.

Returns

NTP time stamp.

Definition at line 174 of file timent.cpp.

5.11.2.9 setDS3231time (byte *second,* byte *minute,* byte *hour,* byte *dayOfWeek,* byte *dayOfMonth,* byte *month,* byte *year*)

Setter for RTC.

Parameters

second	Second of time.
minute	Minute of time.
hour	Hours of time
dayOfWeek	Day of the week.
dayOfMonth	Day of the month.
month	Current month.
year	Current year.

Definition at line 35 of file timent.cpp.

5.11.2.10 setTime ( )

Set time by NTP server to RTC.

Returns

Status of time setting.

Definition at line 194 of file timent.cpp.

5.11.3 Variable Documentation

5.11.3.1 unsigned int localPort = 31011

Local TCP connection port.

Definition at line 14 of file timent.cpp.

5.11.3.2 const int NTP_PACKET_SIZE = 48

Definition at line 16 of file timent.cpp.

5.11.3.3 byte pb[NTP_PACKET_SIZE]

Definition at line 17 of file timent.cpp.

5.11.3.4 byte timeServer[ ] = {10, 0, 0, 4}

Remote NTP server.

Definition at line 13 of file timent.cpp.

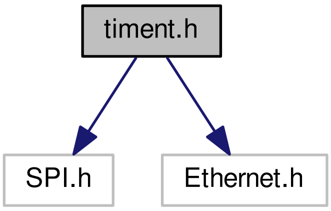
5.11.3.5 EthernetUDP Udp

Definition at line 18 of file timent.cpp.

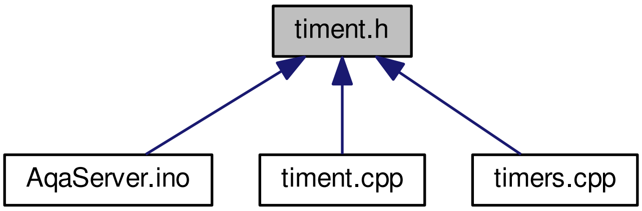
5.12 timent.h File Reference

```
#include <SPI.h>
#include <Ethernet.h>
```

Include dependency graph for timent.h:



This graph shows which files directly or indirectly include this file:



Functions

- void initTiment ()

  Init of RTC.
- byte decToBcd (byte val)

  Decadic format to binary coded decimal.
- byte bcdToDec (byte val)

  Binary coded decimal to decadic format.
- void setDS3231time (byte second, byte minute, byte hour, byte dayOfWeek, byte dayOfMonth, byte month, byte year)

  Setter for RTC.
- void readDS3231MinutesHoures (byte ∗minute, byte ∗hour)
- void readDS3231time (byte ∗second, byte ∗minute, byte ∗hour, byte ∗dayOfWeek, byte ∗dayOfMonth, byte ∗month, byte ∗year)
- void displayHttpTime (EthernetClient ∗client)

  Response for ethernet connection for time values.
- void displayTime ()

  Displays time on the console.
- unsigned long sendNTPpacket (byte ∗address)

  Sender of NTP request to NTP server given by address.
- boolean setTime ()

  Set time by NTP server to RTC.

5.12.1 Function Documentation

5.12.1.1 byte bcdToDec (byte *val*)

Binary coded decimal to decadic format.

Parameters

val	Binary coded number.

Returns

Decimal value.

Definition at line 32 of file timent.cpp.

5.12.1.2 byte decToBcd (byte *val*)

Decadic format to binary coded decimal.

Parameters

val	Decadic number

Returns

Binary coded decimal value.

Definition at line 27 of file timent.cpp.

5.12.1.3 void displayHttpTime (EthernetClient ∗ *client*)

Response for ethernet connection for time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 75 of file timent.cpp.

5.12.1.4 void displayTime ( )

Displays time on the console.

Definition at line 124 of file timent.cpp.

5.12.1.5 void initTiment ( )

Init of RTC.

Definition at line 23 of file timent.cpp.

5.12.1.6 void readDS3231MinutesHoures (byte ∗ *minute,* byte ∗ *hour*)

Definition at line 49 of file timent.cpp.

5.12.1.7 void readDS3231time (byte ∗ *second,* byte ∗ *minute,* byte ∗ *hour,* byte ∗ *dayOfWeek,* byte ∗ *dayOfMonth,* byte ∗ *month,* byte ∗ *year*)

Definition at line 60 of file timent.cpp.

5.12.1.8 unsigned long sendNTPpacket (byte ∗ *address*)

Sender of NTP request to NTP server given by address.

Parameters

address	Address of NTP server.

Returns

NTP time stamp.

Definition at line 174 of file timent.cpp.

5.12.1.9 void setDS3231time (byte *second,* byte *minute,* byte *hour,* byte *dayOfWeek,* byte *dayOfMonth,* byte *month,* byte *year*)

Setter for RTC.

Parameters

second	Second of time.
minute	Minute of time.
hour	Hours of time
dayOfWeek	Day of the week.
dayOfMonth	Day of the month.
month	Current month.
year	Current year.

Definition at line 35 of file timent.cpp.

5.12.1.10 boolean setTime ( )

Set time by NTP server to RTC.

Returns

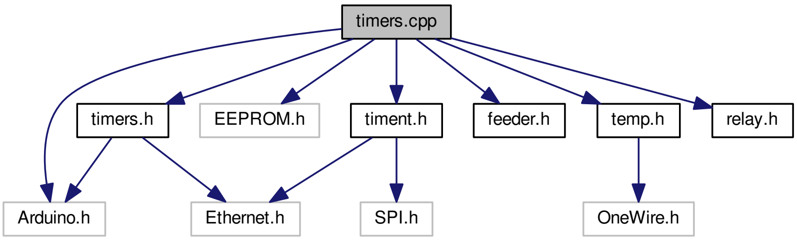
Status of time setting.

Definition at line 194 of file timent.cpp.

5.13 timers.cpp File Reference

```
#include "timers.h"
#include <Arduino.h>
#include <EEPROM.h>
#include "timent.h"
#include "feeder.h"
#include "temp.h"
#include "relay.h"
```

Include dependency graph for timers.cpp:



Functions

- void setTimers ()

  Read RTC minute & hour values.
- void checkTempFrMin (int heater)

  Check temperature in intervals.
- void feedFish ()

  Feeds fish in aquarium.
- void readKonstVal ()

  Read EEPROM values and set time time values for aqarium.
- bool setTempr (char ∗input)

  Set wish temperature of aquarium.
- void checkTemp (int heater)

  Check temperature.
- boolean setEETime (char ∗input)

  Setup of time in EEPROM.
- unsigned int getWishTemp ()

  Getter for wish temperature in tank.
- void setupTimes (int lights, int filter)

  Seting up pins by current time.
- void displayHttpWakeTime (EthernetClient ∗client)

  Response for ethernet connection for wake time values.
- void displayHttpSleepTime (EthernetClient ∗client)

  Response for ethernet connection for sleep time values.
- void displayHttpFeedTime (EthernetClient ∗client)

  Response for ethernet connection for feep time values.

Variables

- Tmr wake

  Aquarium wake time.
- Tmr endt

  Aquarium sleep time.
- Tmr feedt

  Aquarium feed time.
- int tankTmpr

Aquarium wish temperature.

- byte minute

  Current aquarium minute time.
- byte hour

  Current aquarium minute time.
- boolean checktemp = true

5.13.1 Function Documentation

5.13.1.1 checkTemp (int *heater*)

Check temperature.

Parameters

heater	Heater pin number.

Definition at line 79 of file timers.cpp.

5.13.1.2 checkTempFrMin (int *heater*)

Check temperature in intervals.

Parameters

heater	Heater pin number.

Definition at line 26 of file timers.cpp.

5.13.1.3 displayHttpFeedTime (EthernetClient ∗ *client*)

Response for ethernet connection for feep time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 151 of file timers.cpp.

5.13.1.4 displayHttpSleepTime (EthernetClient ∗ *client*)

Response for ethernet connection for sleep time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 144 of file timers.cpp.

5.13.1.5 displayHttpWakeTime (EthernetClient ∗ *client*)

Response for ethernet connection for wake time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 137 of file timers.cpp.

5.13.1.6 feedFish ( )

Feeds fish in aquarium.

Definition at line 35 of file timers.cpp.

5.13.1.7 getWishTemp ( )

Getter for wish temperature in tank.

Returns

Wish temperature.

Definition at line 119 of file timers.cpp.

5.13.1.8 readKonstVal ( )

Read EEPROM values and set time time values for aqarium.

Definition at line 44 of file timers.cpp.

5.13.1.9 setEETime (char ∗ *input*)

Setup of time in EEPROM.

Parameters

input	Time to store in EEPROM.

Returns

Statues of stored values.

Definition at line 87 of file timers.cpp.

5.13.1.10 setTempr (char ∗ *input*)

Set wish temperature of aquarium.

Parameters

input	Input of readed chars

Returns

Statues of written data.

Definition at line 72 of file timers.cpp.

5.13.1.11 setTimers ( )

Read RTC minute & hour values.

Definition at line 22 of file timers.cpp.

5.13.1.12 setupTimes (int *lights*, int *filter*)

Seting up pins by current time.

Parameters

lights	Lights pin value.
filter	Filter pin value.

Definition at line 123 of file timers.cpp.

5.13.2 Variable Documentation

5.13.2.1 boolean checktemp = true

Definition at line 17 of file timers.cpp.

5.13.2.2 Tmr endt

Aquarium sleep time.

Definition at line 10 of file timers.cpp.

5.13.2.3 Tmr feedt

Aquarium feed time.

Definition at line 11 of file timers.cpp.

5.13.2.4 byte hour

Current aquarium minute time.

Definition at line 15 of file timers.cpp.

5.13.2.5 byte minute

Current aquarium minute time.

Definition at line 14 of file timers.cpp.

5.13.2.6 int tankTmpr

Aquarium wish temperature.

Definition at line 12 of file timers.cpp.

5.13.2.7 Tmr wake

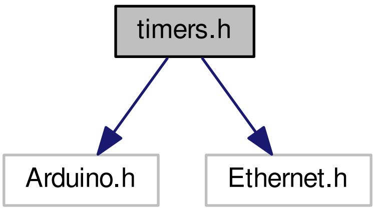
Aquarium wake time.

Definition at line 9 of file timers.cpp.

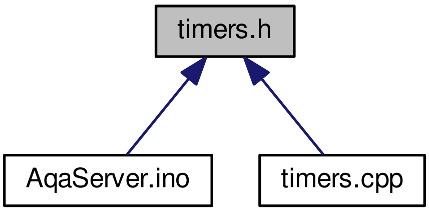
5.14 timers.h File Reference

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

Include dependency graph for timers.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Tmr

  Timer class for containg timers like wake time or sleep time.

Functions

- void setTimers ()

  Read RTC minute & hour values.
- void readKonstVal ()

  Read EEPROM values and set time time values for aqarium.
- bool setTempr (char ∗input)

  Set wish temperature of aquarium.
- void feedFish ()

  Feeds fish in aquarium.
- void setupTimes (int lights, int filter)

Seting up pins by current time.
- unsigned int getWishTemp ()

  Getter for wish temperature in tank.
- void checkTempFrMin (int heater)

  Check temperature in intervals.
- void checkTemp (int heater)

  Check temperature.
- boolean setEETime (char ∗input)

  Setup of time in EEPROM.
- void displayHttpWakeTime (EthernetClient ∗client)

  Response for ethernet connection for wake time values.
- void displayHttpFeedTime (EthernetClient ∗client)

  Response for ethernet connection for feep time values.
- void displayHttpSleepTime (EthernetClient ∗client)

  Response for ethernet connection for sleep time values.

5.14.1 Function Documentation

5.14.1.1 void checkTemp (int *heater*)

Check temperature.

Parameters

heater	Heater pin number.

Definition at line 79 of file timers.cpp.

5.14.1.2 void checkTempFrMin (int *heater*)

Check temperature in intervals.

Parameters

heater	Heater pin number.

Definition at line 26 of file timers.cpp.

5.14.1.3 void displayHttpFeedTime (EthernetClient ∗ *client*)

Response for ethernet connection for feep time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 151 of file timers.cpp.

5.14.1.4 void displayHttpSleepTime (EthernetClient ∗ *client*)

Response for ethernet connection for sleep time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 144 of file timers.cpp.

5.14.1.5 void displayHttpWakeTime (EthernetClient ∗ *client*)

Response for ethernet connection for wake time values.

Parameters

client	Ethernet client to send the message to.

Definition at line 137 of file timers.cpp.

5.14.1.6 void feedFish ( )

Feeds fish in aquarium.

Definition at line 35 of file timers.cpp.

5.14.1.7 unsigned int getWishTemp ( )

Getter for wish temperature in tank.

Returns

Wish temperature.

Definition at line 119 of file timers.cpp.

5.14.1.8 void readKonstVal ( )

Read EEPROM values and set time time values for aqarium.

Definition at line 44 of file timers.cpp.

5.14.1.9 boolean setEETime (char ∗ *input*)

Setup of time in EEPROM.

Parameters

input	Time to store in EEPROM.

Returns

Statues of stored values.

Definition at line 87 of file timers.cpp.

5.14.1.10 bool setTempr (char ∗ *input*)

Set wish temperature of aquarium.

Parameters

input	Input of readed chars

Returns

Statues of written data.

Definition at line 72 of file timers.cpp.

5.14.1.11 void setTimers ( )

Read RTC minute & hour values.

Definition at line 22 of file timers.cpp.

5.14.1.12 void setupTimes (int *lights*, int *filter*)

Seting up pins by current time.

Parameters

lights	Lights pin value.
filter	Filter pin value.

Definition at line 123 of file timers.cpp.

Index