Arduino Ethernet Shield/USB Interface Project

Version 0.1.1

Generated by Doxygen 1.6.1

Sun Dec 5 09:39:33 2010

Contents

1 Directory Hierarchy 1

1.1 Directories 1

2 Class Index 3

2.1 Class List 3

3 Directory Documentation 5

3.1 Arduino/ Directory Reference 5

3.2 Arduino/loop_cpp/include/ Directory Reference 6

3.3 Arduino/loop_cpp/ Directory Reference 7

3.4 Arduino/loop_cpp/src/ Directory Reference 8

4 Class Documentation 9

4.1 Arduino Class Reference 9

4.1.1 Detailed Description 10

4.1.2 Constructor & Destructor Documentation 10

4.1.2.1 Arduino 10

4.1.2.2 Arduino 10

4.1.2.3 Arduino 11

4.1.2.4 ∼Arduino 11

4.1.3 Member Function Documentation 11

4.1.3.1 connect 11

4.1.3.2 flipBit 12

4.1.3.3 getIpAddress 12

4.1.3.4 getMacAddress 12

4.1.3.5 init_ethernet 12

4.1.3.6 operator= 13

4.1.3.7 reconnect 13

4.1.3.8 serialConnect 13

4.1.3.9 serialPrint 13
4.1.3.10 serialWriteToFile 14
4.1.3.11 setIpAddress 14
4.1.3.12 setMacAddress 14
4.2 StartIt Class Reference 15
4.2.1 Detailed Description 15
4.2.2 Constructor & Destructor Documentation 15
4.2.2.1 StartIt 15
4.2.2.2 ∼StartIt 15
4.2.3 Member Function Documentation 15
4.2.3.1 run_it 15

Chapter 1

Directory Hierarchy

1.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Arduino 5
loop_cpp 7
include 6
src 8

Chapter 2

Class Index

2.1 Class List

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

Arduino 9
StartIt 15

Chapter 3

Directory Documentation

3.1 Arduino/ Directory Reference

Directories

- directory loop_cpp

3.2 Arduino/loop_cpp/include/ Directory Reference

Files

- file **arduino.h**
- file **StartIt.h**

3.3 Arduino/loop_cpp/ Directory Reference

Directories

- directory include
- directory src

3.4 Arduino/loop_cpp/src/ Directory Reference

Files

- file **arduino.cpp**
- file **loop_cpp.pde**
- file **StartIt.cpp**

Chapter 4

Class Documentation

4.1 Arduino Class Reference

Public Member Functions

- Arduino ()

  CONSTRUCTOR: Arduino::Arduino(); Class default constructor.

- Arduino (byte mac[6], byte ip[4])

  CONSTRUCTOR: Arduino::Arduino(byte mac [6], byte ip [4]); Class ethernet style constructor.

- Arduino (const Arduino &rhs)

  COPY CONSTRUCTOR: Arduino::Arduino(const Arduino &rhs); Class copy constructor.

- Arduino operator= (const Arduino &rhs)

  ASSIGNMENT OPERATOR: Arduino::operator=Arduino(const Arduino rhs); Class assignment operator.

- ∼Arduino ()

  DESTRUCTOR: Arduino::∼Arduino(); Class default destructor.

- void init_ethernet ()

  METHOD: Arduino::init_ethernet(); Default ether net mode initializer.

- void flipBit (int pin)

  METHOD: Arduino::flipBit(int pin); Default ether net mode initializer.

- const byte ∗ getMacAddress ()

  METHOD: Arduino: getMacAddress(); Getter for this arduino's MAC address.

- const byte ∗ getIpAddress ()

  METHOD: Arduino::getIpAddress(); Getter for this arduino's IP [IEEE 802.11 IPv4] address;.

- void setMacAddress (byte ∗)

  METHOD: Arduino: setMacAddress(); setter for this arduino's MAC address.

- void setIpAddress (byte ∗)

  METHOD: Arduino::setIpAddress(); Setter for this arduino's IP [IEEE 802.11 IPv4] address;.

- void reconnect ()

  METHOD: Arduino::reconnect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

- void connect ()

  METHOD: Arduino::connect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

- void serialConnect ()

  METHOD: Arduino::serialConnect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

- void serialWriteToFile (const char ∗, char ∗)

  METHOD: Arduino::serialWriteToFile(const char∗, char∗).

- void serialPrint (const char ∗)

  METHOD: Arduino::serialPrint(const char∗); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

4.1.1 Detailed Description

Definition at line 35 of file arduino.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Arduino::Arduino ()

CONSTRUCTOR: Arduino::Arduino(); Class default constructor. CONSTRUCTOR: Arduino::Arduino() Class default constructor.

Returns:

None

Parameters:

← ***None***

← ***None***

Definition at line 23 of file arduino.cpp.

4.1.2.2 Arduino::Arduino (byte *mac*[6], byte *ip*[4])

CONSTRUCTOR: Arduino::Arduino(byte mac [6], byte ip [4]); Class ethernet style constructor.

Returns:

None

Parameters:

← ***byte*** mac[6] (MAC address)

← ***byte*** ip[4] (IP address [IEEE 802.11 IPv4])

← ***byte*** mac[6] (mac address)

← ***byte*** ip[4] (ip address [IEEE 802.11 IPv4])

Definition at line 38 of file arduino.cpp.

4.1.2.3 Arduino::Arduino (const Arduino & *rhs*)

COPY CONSTRUCTOR: Arduino::Arduino(const Arduino &rhs); Class copy constructor. Class copy constructor. Effectively copies over all private member variables from object rhs.

Returns:

None

Parameters:

← ***const*** Arduino rhs - Reference to the right hand side of the equation

← ***const*** Arduino rhs - Reference to the right hand side of the equation

Definition at line 68 of file arduino.cpp.

4.1.2.4 Arduino::∼Arduino ()

DESTRUCTOR: Arduino::∼Arduino(); Class default destructor. DESTRUCTOR: Arduino::∼Arduino() Class default destructor.

Returns:

None

Parameters:

← ***None***

← ***None***

Definition at line 135 of file arduino.cpp.

4.1.3 Member Function Documentation

4.1.3.1 void Arduino::connect ()

METHOD: Arduino::connect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class. METHOD: Arduino::connect() Connects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

Definition at line 285 of file arduino.cpp.

4.1.3.2 void Arduino::flipBit (int *pin*)

METHOD: Arduino::flipBit(int pin); Default ether net mode initializer. METHOD: Arduino::flipBit(int pin) Default ether net mode initializer.

Returns:

None

Parameters:

← ***int*** pin

← ***int*** pin

Definition at line 185 of file arduino.cpp.

4.1.3.3 const byte ∗ Arduino::getIpAddress ()

METHOD: Arduino::getIpAddress(); Getter for this arduino's IP [IEEE 802.11 IPv4] address;. METHOD: Arduino::getIpAddress() Getter for this arduino's IP [IEEE 802.11 IPv4] address. Returns a pointer to the first element of a byte array in memory (casted as a void ∗).

Returns:

const byte∗

Definition at line 215 of file arduino.cpp.

4.1.3.4 const byte ∗ Arduino::getMacAddress ()

METHOD: Arduino: getMacAddress(); Getter for this arduino's MAC address. METHOD: Arduino: getMacAddress() Getter for this arduino's MAC address. Returns a pointer to the first element of a byte array in memory (casted as a void ∗).

Returns:

const byte∗

Definition at line 200 of file arduino.cpp.

4.1.3.5 void Arduino::init_ethernet ()

METHOD: Arduino::init_ethernet(); Default ether net mode initializer. METHOD: Arduino::init_ethernet() Default ether net mode initializer.

Returns:

None

Parameters:

← ***None***

← ***None***

Definition at line 154 of file arduino.cpp.

4.1.3.6 Arduino::Arduino Arduino::operator= (const Arduino & *rhs*)

ASSIGNMENT OPERATOR: Arduino::operator=Arduino(const Arduino rhs); Class assignment operator. ASSIGNMENT OPERATOR: Arduino::operator=Arduino(const Arduino rhs) Class assignment operator.

Returns:

Arduino lhs - Reference to the left hand side of the equation

Parameters:

← ***const*** Arduino rhs - Reference to the right hand side of the equation

Returns:

Arduino ∗lhsPtr - Reference to the left hand side of the equation

Parameters:

← ***const*** Arduino rhs - Reference to the right hand side of the equation

Definition at line 101 of file arduino.cpp.

4.1.3.7 void Arduino::reconnect ()

METHOD: Arduino::reconnect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class. METHOD: Arduino::reconnect() Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

Definition at line 264 of file arduino.cpp.

4.1.3.8 void Arduino::serialConnect ()

METHOD: Arduino::serialConnect(); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class. METHOD: Arduino::serialConnect() Connects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

Definition at line 309 of file arduino.cpp.

4.1.3.9 void Arduino::serialPrint (const char ∗ *txt*)

METHOD: Arduino::serialPrint(const char∗); Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class. METHOD: Arduino::serialPrint(const char∗) Connects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

Parameters:

← ***const*** char∗ txt

Definition at line 324 of file arduino.cpp.

4.1.3.10 void Arduino::serialWriteToFile (const char ∗ *inTxt*, char ∗ *outFile*)

METHOD: Arduino::serialWriteToFile(const char∗, char∗). METHOD: Arduino::serialWriteToFile(const char∗ inTxt, char∗ outFile).

Parameters:

← ***const*** char∗ inTxt

→ ***char***∗ outFileLoc Reconnects with the Arduino's current MAC address and IPv4 Address. This Arduino's IP and MAC address are stored as private member variables of this class.

Definition at line 340 of file arduino.cpp.

4.1.3.11 void Arduino::setIpAddress (byte ∗ *addrPtr*)

METHOD: Arduino::setIpAddress(); Setter for this arduino's IP [IEEE 802.11 IPv4] address;. METHOD: Arduino::setIpAddress() Setter for this arduino's IP [IEEE 802.11 IPv4] address. Changes a private member variable within this Arduino object. Attempts to reconnect.

Parameters:

← ***byte***∗ addrPtr

Definition at line 247 of file arduino.cpp.

4.1.3.12 void Arduino::setMacAddress (byte ∗ *addrPtr*)

METHOD: Arduino: setMacAddress(); setter for this arduino's MAC address. METHOD: Arduino: setMacAddress() Setter for this arduino's MAC address. Changes a private member variable within this Arduino object. Attempts to reconnect.

Parameters:

← ***byte***∗ addrPtr

Definition at line 230 of file arduino.cpp.

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

- arduino.h
- arduino.cpp

4.2 StartIt Class Reference

Public Member Functions

- StartIt (Arduino obj)

  CONSTRUCTOR: StartIt() Class default constructor.

- ∼StartIt ()

  DESTRUCTOR: ∼StartIt() Class destructor.

- void run_it ()
- void **setDateTime** (char ∗inTxt)

4.2.1 Detailed Description

Definition at line 24 of file StartIt.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 StartIt::StartIt (Arduino *obj*)

CONSTRUCTOR: StartIt() Class default constructor.

Parameters:

← ***Arduino*** obj

Definition at line 21 of file StartIt.cpp.

4.2.2.2 StartIt::∼StartIt ()

DESTRUCTOR: ∼StartIt() Class destructor.

Definition at line 65 of file StartIt.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void StartIt::run_it ()

METHOD: run_it() Runs the main code we wish to execute on the microcontroller.

Definition at line 33 of file StartIt.cpp.

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

- StartIt.h
- StartIt.cpp

Index

~Arduino
    Arduino, 11
~StartIt
    StartIt, 15

Arduino, 9
    ~Arduino, 11
    Arduino, 10, 11
    connect, 11
    flipBit, 11
    getIpAddress, 12
    getMacAddress, 12
    init_ethernet, 12
    operator=, 12
    reconnect, 13
    serialConnect, 13
    serialPrint, 13
    serialWriteToFile, 13
    setIpAddress, 14
    setMacAddress, 14
Arduino/ Directory Reference, 5
Arduino/loop_cpp/ Directory Reference, 7
Arduino/loop_cpp/include/ Directory Reference, 6
Arduino/loop_cpp/src/ Directory Reference, 8

connect
    Arduino, 11

flipBit
    Arduino, 11

getIpAddress
    Arduino, 12
getMacAddress
    Arduino, 12

init_ethernet
    Arduino, 12

operator=
    Arduino, 12

reconnect
    Arduino, 13
run_it
    StartIt, 15

serialConnect
    Arduino, 13
serialPrint
    Arduino, 13
serialWriteToFile
    Arduino, 13
setIpAddress
    Arduino, 14
setMacAddress
    Arduino, 14
StartIt, 15
    ~StartIt, 15
    run_it, 15
    StartIt, 15