

## Project Parsnip: Hardware

Generated by Doxygen 1.9.4

<b>1 README</b>	<b>1</b>
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy . . . . .	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List . . . . .	5
<b>4 File Index</b>	<b>7</b>
4.1 File List . . . . .	7
<b>5 Class Documentation</b>	<b>9</b>
5.1 Device Class Reference . . . . .	9
5.2 DeviceESP32 Class Reference . . . . .	10
5.3 DeviceESP8266 Class Reference . . . . .	10
5.4 DeviceServerInterface Class Reference . . . . .	10
5.5 EEPROM_CONFIG_t Struct Reference . . . . .	11
5.6 LoraSensor Class Reference . . . . .	11
5.6.1 Member Function Documentation . . . . .	11
5.6.1.1 getId() . . . . .	11
5.6.1.2 read() . . . . .	12
5.7 Sensor Class Reference . . . . .	12
5.8 TemperatureSensorDHT Class Reference . . . . .	12
5.8.1 Member Function Documentation . . . . .	13
5.8.1.1 getId() . . . . .	13
5.8.1.2 read() . . . . .	13
5.9 Transmitter Class Reference . . . . .	13
<b>6 File Documentation</b>	<b>15</b>
6.1 Device.h . . . . .	15
6.2 DeviceServerInterface.h . . . . .	15
6.3 LoraSensor.h . . . . .	16
6.4 Sensor.h . . . . .	16
6.5 TemperatureSensorDHT.h . . . . .	16
6.6 Transmitter.h . . . . .	17
6.7 DeviceESP32.h . . . . .	17
6.8 DeviceESP8266.h . . . . .	18
6.9 TransmitterESP32.h . . . . .	18
6.10 TransmitterESP8266.h . . . . .	18
<b>Index</b>	<b>19</b>

# Chapter 1

## README

Install PlatformIO (macOS/Linux): <https://docs.platformio.org/en/stable/core/installation/methods/html#super-quick-macos-linux>

Homebrew: brew install platformio

Windows install: <https://docs.platformio.org/en/stable/core/installation/methods/installer-s/html#local-download-macos-linux-windows>

Install PlatformIO shell commands: <https://docs.platformio.org/en/stable/core/installation/shell-c/html#unix-and-unix-like>

Supported Platforms: ESP32 ESP8266

Supported Sensors: DHT11 Temperature & Humidity DHT21 Temperature & Humidity DHT22 Temperature & Humidity

To compile code, go into a esp32 or esp8266 directory and run: pio run

To compile and upload code to a device: pio run --target upload Note: You may have to change the platformio.ini file and change the upload\_port and monitor\_port to the correct port.

To monitor the device (device must be plugged into the port) run: pio device monitor

## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Device . . . . .	9
DeviceESP32 . . . . .	10
DeviceESP8266 . . . . .	10
DeviceServerInterface . . . . .	10
EEPROM_CONFIG_t . . . . .	11
Sensor . . . . .	12
LoraSensor . . . . .	11
TemperatureSensorDHT . . . . .	12
Transmitter . . . . .	13

## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Device</a>	9
<a href="#">DeviceESP32</a>	10
<a href="#">DeviceESP8266</a>	10
<a href="#">DeviceServerInterface</a>	10
<a href="#">EEPROM_CONFIG_t</a>	11
<a href="#">LoraSensor</a>	11
<a href="#">Sensor</a>	12
<a href="#">TemperatureSensorDHT</a>	12
<a href="#">Transmitter</a>	13

## Chapter 4

# File Index

### 4.1 File List

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

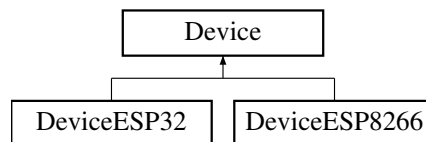
lib/deviceBase/ <a href="#">Device.h</a> . . . . .	15
lib/deviceServerInterface/ <a href="#">DeviceServerInterface.h</a> . . . . .	15
lib/lora/ <a href="#">LoraSensor.h</a> . . . . .	16
lib/sensorBase/ <a href="#">Sensor.h</a> . . . . .	16
lib/temperatureSensors/ <a href="#">TemperatureSensorDHT.h</a> . . . . .	16
lib/transmitterBase/ <a href="#">Transmitter.h</a> . . . . .	17
receiver/esp32/src/ <a href="#">DeviceESP32.h</a> . . . . .	17
receiver/esp8266/src/ <a href="#">DeviceESP8266.h</a> . . . . .	18
transmitter/esp32/src/ <a href="#">TransmitterESP32.h</a> . . . . .	18
transmitter/esp8266/src/ <a href="#">TransmitterESP8266.h</a> . . . . .	18

## Chapter 5

# Class Documentation

### 5.1 Device Class Reference

Inheritance diagram for Device:



#### Public Member Functions

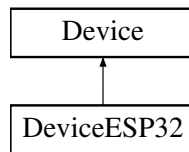
- **Device** (String deviceServerAddress="https://parsnipbackend.azurewebsites.net")
- AutoConnect & **getPortal** ()
- std::map< std::string, float > **readSensors** ()
- void **addSensor** ([Sensor](#) \*sensor)
- void **removeSensor** (int id)
- void **handleClientRequest** ()
- void **beginServer** ()
- void **sendSensorData** ()
- String **onHandleAuthToken** (AutoConnectAux &page, PageArgument &args)
- String **onLoadAuthPage** (AutoConnectAux &page, PageArgument &args)
- String **getAuthenticationToken** ()

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

- lib/deviceBase/Device.h
- lib/deviceBase/Device.cpp

## 5.2 DeviceESP32 Class Reference

Inheritance diagram for DeviceESP32:



### Public Member Functions

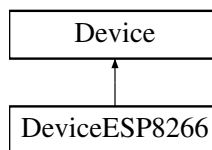
- **DeviceESP32** (String deviceServerAddress="https://parsnipbackend.azurewebsites.net")

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

- receiver/esp32/src/DeviceESP32.h
- receiver/esp32/src/DeviceESP32.cpp

## 5.3 DeviceESP8266 Class Reference

Inheritance diagram for DeviceESP8266:



### Public Member Functions

- **DeviceESP8266** (String deviceServerAddress="https://parsnipbackend.azurewebsites.net")

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

- receiver/esp8266/src/DeviceESP8266.h
- receiver/esp8266/src/DeviceESP8266.cpp

## 5.4 DeviceServerInterface Class Reference

### Public Member Functions

- **DeviceServerInterface** (String baseUrl)
- void **setAuthenticationToken** (String authToken)
- int **sendPlantData** (std::map< std::string, float > sensorReadings)
- String **getAuthenticationToken** ()
- String **getDeviceSensorIds** ()
- void **setHttpUrl** (String url)
- void **setAuthHeader** ()

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

- lib/deviceServerInterface/DeviceServerInterface.h
- lib/deviceServerInterface/DeviceServerInterface.cpp



## 5.5 EEPROM\_CONFIG\_t Struct Reference

### Public Attributes

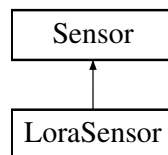
- char **token** [257]

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

- lib/deviceBase/Device.cpp

## 5.6 LoraSensor Class Reference

Inheritance diagram for LoraSensor:



### Public Member Functions

- **LoraSensor** (int id)
- int **getId** () const override
- std::map< std::string, float > **read** () const override
- float **getTemperature** () const
- float **getHumidity** () const
- float **getHeatIndex** () const
- float **getHeatIndexC** () const
- std::tuple< float, float > **getTemperatureAndHumidity** () const

### 5.6.1 Member Function Documentation

#### 5.6.1.1 getId()

```
int LoraSensor::getId ( ) const [override], [virtual]
```

Implements [Sensor](#).

### 5.6.1.2 read()

```
std::map< std::string, float > LoraSensor::read ( ) const [override], [virtual]
```

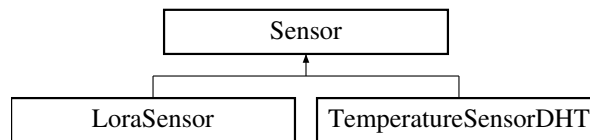
Implements [Sensor](#).

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

- lib/lora/LoraSensor.h
- lib/lora/LoraSensor.cpp

## 5.7 Sensor Class Reference

Inheritance diagram for Sensor:



### Public Member Functions

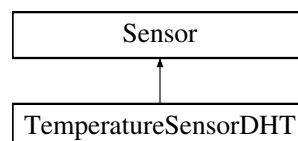
- virtual int **getId** ( ) const =0
- virtual std::map< std::string, float > **read** ( ) const =0

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

- lib/sensorBase/Sensor.h

## 5.8 TemperatureSensorDHT Class Reference

Inheritance diagram for TemperatureSensorDHT:



### Public Member Functions

- **TemperatureSensorDHT** (int id, int temperatureSensorPin, u\_int8\_t dhtType)
- int **getId** ( ) const override
- std::map< std::string, float > **read** ( ) const override
- float **getTemperature** ( ) const
- float **getHumidity** ( ) const
- float **getHeatIndex** ( ) const
- float **getHeatIndexC** ( ) const
- std::tuple< float, float > **getTemperatureAndHumidity** ( ) const

## 5.8.1 Member Function Documentation

### 5.8.1.1 getId()

```
int TemperatureSensorDHT::getId ( ) const [override], [virtual]
```

Implements [Sensor](#).

### 5.8.1.2 read()

```
std::map< std::string, float > TemperatureSensorDHT::read ( ) const [override], [virtual]
```

Implements [Sensor](#).

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

- lib/temperatureSensors/TemperatureSensorDHT.h
- lib/temperatureSensors/TemperatureSensorDHT.cpp

## 5.9 Transmitter Class Reference

### Public Member Functions

- **Transmitter** (String deviceServerAddress="https://parsnipbackend.azurewebsites.net")
- std::map< std::string, float > **readSensors** ()
- void **addSensor** ([Sensor](#) \*sensor)
- void **removeSensor** (int id)
- void **sendSensorData** ()

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

- lib/transmitterBase/Transmitter.h
- lib/transmitterBase/Transmitter.cpp

## Chapter 6

# File Documentation

### 6.1 Device.h

```
1 #ifndef Device_h
2 #define Device_h
3
4 #include "Arduino.h"
5
6 #include <vector>
7 #include <map>
8 #include <string>
9
10 #include "Sensor.h"
11 #include "DeviceServerInterface.h"
12
13 #include <EEPROM.h>
14 #include <AutoConnect.h>
15
16 class Device
17 {
18 public:
19     Device(String deviceServerAddress = "https://parsnipbackend.azurewebsites.net");
20
21     // Concrete methods
22     AutoConnect &getPortal();
23     std::map<std::string, float> readSensors();
24     void addSensor(Sensor *sensor);
25     void removeSensor(int id);
26     void handleClientRequest();
27     void beginServer();
28     void sendSensorData();
29
30     String onHandleAuthToken(AutoConnectAux &page, PageArgument &args);
31     String onLoadAuthPage(AutoConnectAux &page, PageArgument &args);
32
33     String getAuthenticationToken();
34
35 private:
36     WebServer server;
37     AutoConnect Portal;
38     std::vector<Sensor *> sensors_;
39     DeviceServerInterface *deviceServerInterface;
40 };
41
42 #endif
```

### 6.2 DeviceServerInterface.h

```
1 #ifdef ESP32
2 #include <HTTPClient.h>
3 #endif
4
5 #ifdef ESP8266
6 #include <ESP8266HTTPClient.h>
7 #endif
8
9 #include <map>
```

```

10 #include <string>
11
12 class DeviceServerInterface
13 {
14 public:
15     DeviceServerInterface(String baseUrl);
16     void setAuthenticationToken(String authToken);
17     int sendPlantData(std::map<std::string, float> sensorReadings);
18     String getAuthenticationToken();
19     String getDeviceSensorIds();
20     void setHttpUrl(String url);
21     void setAuthHeader();
22
23 private:
24     HTTPClient http;
25     String baseUrl;
26     String authToken;
27 };

```

## 6.3 LoraSensor.h

```

1 #ifndef LoraSensor_h
2 #define LoraSensor_h
3
4 #include "../Sensor.h"
5
6 // #include <LoRa.h>
7
8 #include <map>
9 #include <tuple>
10
11 class LoraSensor : public Sensor
12 {
13 public:
14     LoraSensor(int id);
15     int getId() const override;
16     std::map<std::string, float> read() const override;
17
18     float getTemperature() const;
19     float getHumidity() const;
20     float getHeatIndex() const;
21     float getHeatIndexC() const;
22     std::tuple<float, float> getTemperatureAndHumidity() const;
23
24 private:
25     int id_;
26 };
27
28 #endif

```

## 6.4 Sensor.h

```

1 #ifndef Sensor_h
2 #define Sensor_h
3
4 #include <map>
5 #include <string>
6
7 class Sensor
8 {
9 public:
10     virtual int getId() const = 0;
11     virtual std::map<std::string, float> read() const = 0;
12
13 private:
14 };
15
16 #endif

```

## 6.5 TemperatureSensorDHT.h

```

1 #ifndef TemperatureSensorDHT_h
2 #define TemperatureSensorDHT_h
3
4 #include "Sensor.h"

```

```

5 #include "DHT.h"
6
7 #include <tuple>
8
9 class TemperatureSensorDHT : public Sensor
10 {
11 public:
12     TemperatureSensorDHT(int id, int temperatureSensorPin, u_int8_t dhtType);
13     int getId() const override;
14     std::map<std::string, float> read() const override;
15
16     float getTemperature() const;
17     float getHumidity() const;
18     float getHeatIndex() const;
19     float getHeatIndexC() const;
20     std::tuple<float, float> getTemperatureAndHumidity() const;
21
22 private:
23     DHT *dht;
24     u_int8_t dhtType;
25     int temperatureSensorPin;
26     int id_;
27     bool sensorWorking;
28 };
29
30 #endif

```

## 6.6 Transmitter.h

```

1 #ifndef Transmitter_h
2 #define Transmitter_h
3
4 #include "Arduino.h"
5
6 #include <vector>
7 #include <map>
8 #include <string>
9
10 #include "Sensor.h"
11
12 class Transmitter
13 {
14 public:
15     Transmitter(String deviceServerAddress = "https://parsnipbackend.azurewebsites.net");
16
17     std::map<std::string, float> readSensors();
18     void addSensor(Sensor *sensor);
19     void removeSensor(int id);
20     void sendSensorData();
21
22 private:
23     std::vector<Sensor *> sensors_;
24 };
25
26 #endif

```

## 6.7 DeviceESP32.h

```

1 #ifndef DeviceESP32_h
2 #define DeviceESP32_h
3
4 #include "Device.h"
5
6 #include "WiFi.h"
7 #include "WebServer.h"
8
9 class DeviceESP32 : public Device
10 {
11 public:
12     DeviceESP32(String deviceServerAddress = "https://parsnipbackend.azurewebsites.net");
13
14 private:
15 };
16
17 #endif

```

## 6.8 DeviceESP8266.h

```
1 #ifndef DeviceESP8266_h
2 #define DeviceESP8266_h
3
4 #include "Device.h"
5
6 #include <ESP8266WiFi.h>
7 #include <ESP8266WebServer.h>
8 #include <WiFiClient.h>
9
10 class DeviceESP8266 : public Device
11 {
12 public:
13     DeviceESP8266(String deviceServerAddress = "https://parsnipbackend.azurewebsites.net");
14
15 private:
16 };
17
18 #endif
```

## 6.9 TransmitterESP32.h

```
1 #ifndef TRANSMITTER_ESP32_H
2 #define TRANSMITTER_ESP32_H
3
4 #include "Transmitter.h"
5
6 class TransmittterESP32() : Transmitter() {}
7
8 #endif
```

## 6.10 TransmitterESP8266.h

```
1 #ifndef TRANSMITTER_ESP8266_H
2 #define TRANSMITTER_ESP8266_H
3
4 #include "Transmitter.h"
5
6 class TransmittterESP8266() : Transmitter() {}
7
8 #endif
```

# Index

- Device, [9](#)
- DeviceESP32, [10](#)
- DeviceESP8266, [10](#)
- DeviceServerInterface, [10](#)
  
- EEPROM\_CONFIG\_t, [11](#)
  
- getId
  - LoraSensor, [11](#)
  - TemperatureSensorDHT, [13](#)
  
- lib/deviceBase/Device.h, [15](#)
- lib/deviceServerInterface/DeviceServerInterface.h, [15](#)
- lib/lora/LoraSensor.h, [16](#)
- lib/sensorBase/Sensor.h, [16](#)
- lib/temperatureSensors/TemperatureSensorDHT.h, [16](#)
- lib/transmitterBase/Transmitter.h, [17](#)
- LoraSensor, [11](#)
  - getId, [11](#)
  - read, [11](#)
  
- read
  - LoraSensor, [11](#)
  - TemperatureSensorDHT, [13](#)
- receiver/esp32/src/DeviceESP32.h, [17](#)
- receiver/esp8266/src/DeviceESP8266.h, [18](#)
  
- Sensor, [12](#)
  
- TemperatureSensorDHT, [12](#)
  - getId, [13](#)
  - read, [13](#)
- Transmitter, [13](#)
- transmitter/esp32/src/TransmitterESP32.h, [18](#)
- transmitter/esp8266/src/TransmitterESP8266.h, [18](#)