

Haxee_ESP32

v0.2

Generované programom Doxygen 1.9.7

1 Register tried	1
1.1 Zoznam tried	1
2 Register súborov	3
2.1 Zoznam súborov	3
3 Dokumentácia tried	5
3.1 Dokumentácia triedy Haxee_ESP32	5
3.2 Dokumentácia triedy Haxee_ESP32_LEDStripe	6
3.2.1 Dokumentácia k metódam	6
3.2.1.1 lightError()	6
3.2.1.2 lightInfo()	7
3.2.1.3 lightSuccess()	7
3.2.1.4 setError()	7
3.2.1.5 setInfo()	7
3.2.1.6 setSuccess()	8
3.2.1.7 setup()	8
4 Dokumentácia súborov	9
4.1 Haxee_ESP32.h	9
4.2 Haxee_ESP32_LEDStripe.h	9
4.3 Haxee_ESP32_MFRC522.h	10
4.4 Haxee_ESP32_MQTT.h	10
Register	11

Kapitola 1

Register tried

1.1 Zoznam tried

Nasledujúci zoznam obsahuje predovšetkým identifikáciu tried, ale nachádzajú sa tu i ďalšie netriviálne prvky, ako sú štruktúry (struct), uniony (union) a rozhrania (interface). V zozname sú uvedené ich stručné popisy:

Haxee_ESP32	5
Haxee_ESP32_LEDStripe	6

Kapitola 2

Register súborov

2.1 Zoznam súborov

Tu nájdete zoznam všetkých dokumentovaných súborov so stručnými popismi:

Haxee_ESP32.h	9
Haxee_ESP32_LEDStripe.h	9
Haxee_ESP32_MFRC522.h	10
Haxee_ESP32_MQTT.h	10

Kapitola 3

Dokumentácia tried

3.1 Dokumentácia triedy Haxee_ESP32

Verejné typy

- enum **MessageType** { **Error** , **Success** , **Info** }

Verejné metódy

- **Haxee_ESP32** (String ssid, String password, String mqtt_server, int mqtt_port, String t)
- bool **setup** ()
- String **readCard** ()
- String **formatMessage** (String cardID)
- void **setupMessageColor** (MessageType type, int r, int g, int b)
- void **publish** (String topic, String text)
- void **reconnect** ()
- bool **clientConnected** ()
- void **clientLoop** ()
- void **timeLoop** ()
- void **lightLed** (MessageType type)
- void **error** ()

Statické verejné metódy

- static void **callback** (char *topic, byte *message, unsigned int length)

Verejné atribúty

- String **topic** = ""

Privátne metódy

- bool **setup_wifi** ()

Privátne atribúty

- `int _success [3] = {0, 255, 0}`
- `int _info [3] = {0, 0, 255}`
- `int _error [3] = {255, 0, 0}`
- `char _ssid [100]`
- `char _password [100]`
- `char _mqtt_server [100]`
- `int _mqtt_port = 1883`

Dokumentácia pre túto triedu bola generovaná z nasledujúcich súborov:

- `Haxee_ESP32.h`
- `Haxee_ESP32.cpp`

3.2 Dokumentácia triedy `Haxee_ESP32_LEDStripe`

Verejné metódy

- **`Haxee_ESP32_LEDStripe ()`**
Konštruktor triedy `Haxee_ESP32_LEDStripe`.
- **`bool setup ()`**
Nastavenie modulu LED pásika.
- **`void lightError ()`**
Rozsvietenie LED pásika na farbu značiacu chybu.
- **`void lightSuccess ()`**
Rozsvietenie LED pásika na farbu značiacu úspešný úkon.
- **`void lightInfo ()`**
Rozsvietenie LED pásika na základnú / informačnú farbu.
- **`void setError (int r, int g, int b)`**
Nastavenie farby pre chybu.
- **`void setSuccess (int r, int g, int b)`**
Nastavenie farby pre úspešný úkon.
- **`void setInfo (int r, int g, int b)`**
Nastavenie informačnej farby.

Privátne atribúty

- `int _success [3] = {0, 255, 0}`
- `int _info [3] = {0, 0, 255}`
- `int _error [3] = {255, 0, 0}`

3.2.1 Dokumentácia k metódam

3.2.1.1 `lightError()`

```
void Haxee_ESP32_LEDStripe::lightError ( )
```

Rozsvietenie LED pásika na farbu značiacu chybu.

Funkcia rozsvieti LED pásik na farbu, ktorá bola nastavená ako chybová farba

3.2.1.2 lightInfo()

```
void Haxee_ESP32_LEDStripe::lightInfo ( )
```

Rozsvietenie LED pásika na základnú / informačnú farbu.

Funkcia rozsvieti LED pásik na farbu, ktorá bola nastavená ako informačná farba

3.2.1.3 lightSuccess()

```
void Haxee_ESP32_LEDStripe::lightSuccess ( )
```

Rozsvietenie LED pásika na farbu značiacu úspešný úkon.

Funkcia rozsvieti LED pásik na farbu, ktorá bola nastavená ako úspešná farba

3.2.1.4 setError()

```
void Haxee_ESP32_LEDStripe::setError (
    int r,
    int g,
    int b )
```

Nastavenie farby pre chybu.

Funkcia nastaví nový RGB kód pre chybovú farbu

Parametre

<i>r</i>	Prvá (RED) hodnota z RGB kódu
<i>g</i>	Druhá (GREEN) hodnota z RGB kódu
<i>b</i>	Tretia (BLUE) hodnota z RGB kódu

3.2.1.5 setInfo()

```
void Haxee_ESP32_LEDStripe::setInfo (
    int r,
    int g,
    int b )
```

Nastavenie informačnej farby.

Funkcia nastaví nový RGB kód pre informačnú farbu

Parametre

<i>r</i>	Prvá (RED) hodnota z RGB kódu
<i>g</i>	Druhá (GREEN) hodnota z RGB kódu
<i>b</i>	Tretia (BLUE) hodnota z RGB kódu

3.2.1.6 setSuccess()

```
void Haxee_ESP32_LEDStripe::setSuccess (
    int r,
    int g,
    int b )
```

Nastavenie farby pre úspešný úkon.

Funkcia nastaví nový RGB kód pre úspešnú farbu

Parametre

<i>r</i>	Prvá (RED) hodnota z RGB kódu
<i>g</i>	Druhá (GREEN) hodnota z RGB kódu
<i>b</i>	Tretia (BLUE) hodnota z RGB kódu

3.2.1.7 setup()

```
bool Haxee_ESP32_LEDStripe::setup ( )
```

Nastavenie modulu LED pásika.

Návratová hodnota

vracia informáciu o tom, či nastavenie bolo úspešné alebo nie

Dokumentácia pre túto triedu bola generovaná z nasledujúcich súborov:

- Haxee_ESP32_LEDStripe.h
- Haxee_ESP32_LEDStripe.cpp

Kapitola 4

Dokumentácia súborov

4.1 Haxee_ESP32.h

```
00001 #ifndef Haxee_ESP32_h
00002 #define Haxee_ESP32_h
00003
00004 #include "Arduino.h"
00005
00006 class Haxee_ESP32
00007 {
00008 private:
00009     int _success[3] = {0, 255, 0};
00010
00011     int _info[3] = {0, 0, 255};
00012
00013     int _error[3] = {255, 0, 0};
00014
00015
00016     char _ssid[100];
00017     char _password[100];
00018     char _mqtt_server[100];
00019     int _mqtt_port = 1883;
00020
00021     bool setup_wifi();
00022
00023 public:
00024     Haxee_ESP32(String ssid, String password, String mqtt_server, int mqtt_port, String t);
00025     Haxee_ESP32();
00026
00027     enum MessageType { Error, Success, Info };
00028
00029     String topic = "";
00030
00031     bool setup();
00032     String readCard();
00033     String formatMessage(String cardID);
00034
00035     void setupMessageColor(MessageType type, int r, int g, int b);
00036
00037     void publish(String topic, String text);
00038     static void callback(char* topic, byte* message, unsigned int length);
00039     void reconnect();
00040     bool clientConnected();
00041     void clientLoop();
00042     void timeLoop();
00043     void lightLed(MessageType type);
00044
00045     void error();
00046 };
00047
00048 #endif
```

4.2 Haxee_ESP32_LEDStripe.h

```
00001 /*****
00002  * @file Haxee_ESP32_LEDStripe.h
00003  *
```

```
00004  * @brief Knižnica na ovládanie LED pásika
00005  *****/
00006  #ifndef Haxee_ESP32_LEDStripe_h
00007  #define Haxee_ESP32_LEDStripe_h
00008
00009  #include "Arduino.h"
00010
00011  class Haxee_ESP32_LEDStripe
00012  {
00013  private:
00014      int _success[3] = {0, 255, 0};
00015
00016      int _info[3] = {0, 0, 255};
00017
00018      int _error[3] = {255, 0, 0};
00019
00020  public:
00024      Haxee_ESP32_LEDStripe();
00025
00031      bool setup();
00032
00038      void lightError();
00039
00045      void lightSuccess();
00046
00052      void lightInfo();
00053
00063      void setError(int r, int g, int b);
00064
00074      void setSuccess(int r, int g, int b);
00075
00085      void setInfo(int r, int g, int b);
00086  };
00087
00088  #endif
```

4.3 Haxee_ESP32_MFRC522.h

4.4 Haxee_ESP32_MQTT.h

Register

Haxee_ESP32, [5](#)
Haxee_ESP32_LEDStripe, [6](#)
 lightError, [6](#)
 lightInfo, [6](#)
 lightSuccess, [7](#)
 setError, [7](#)
 setInfo, [7](#)
 setSuccess, [8](#)
 setup, [8](#)

lightError
 Haxee_ESP32_LEDStripe, [6](#)
lightInfo
 Haxee_ESP32_LEDStripe, [6](#)
lightSuccess
 Haxee_ESP32_LEDStripe, [7](#)

setError
 Haxee_ESP32_LEDStripe, [7](#)
setInfo
 Haxee_ESP32_LEDStripe, [7](#)
setSuccess
 Haxee_ESP32_LEDStripe, [8](#)
setup
 Haxee_ESP32_LEDStripe, [8](#)