Outdoor Module Datasheet

Author: Łukasz Wolf

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

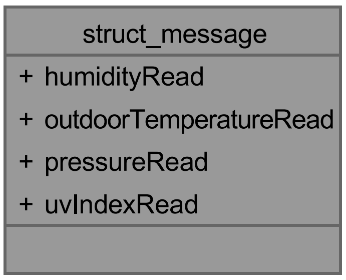
Chapter 3

Class Documentation

3.1 struct_message Struct Reference

Data structure for ESP-NOW transmission.

Collaboration diagram for struct_message:



Public Attributes

- uint8_t humidityRead

  Relative humidity (%)
- int16_t outdoorTemperatureRead

  Temperature ∗ 10 (e.g. 255 = 25.5°C)
- uint16_t pressureRead

  Atmospheric pressure (hPa)
- uint8_t uvIndexRead

  UV Raw Value (Clamped to 255)

3.1.1 Detailed Description

Data structure for ESP-NOW transmission.

Warning

Must match the receiver's structure exactly (including padding).

3.1.2 Member Data Documentation

3.1.2.1 humidityRead

`uint8_t struct_message::humidityRead`

Relative humidity (%)

3.1.2.2 outdoorTemperatureRead

`int16_t struct_message::outdoorTemperatureRead`

Temperature $*$ 10 (e.g. 255 = 25.5°C)

3.1.2.3 pressureRead

`uint16_t struct_message::pressureRead`

Atmospheric pressure (hPa)

3.1.2.4 uvIndexRead

`uint8_t struct_message::uvIndexRead`

UV Raw Value (Clamped to 255)

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

- src/main.cpp

Chapter 4

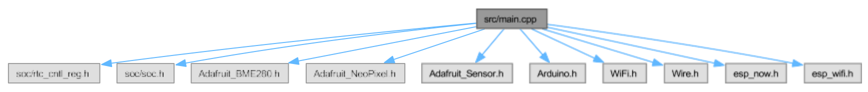
File Documentation

4.1 src/main.cpp File Reference

ESP32 Sensor Node (BME280 + UV) transmitting via ESP-NOW.

```
#include "soc/rtc_cntl_reg.h"
#include "soc/soc.h"
#include <Adafruit_BME280.h>
#include <Adafruit_NeoPixel.h>
#include <Adafruit_Sensor.h>
#include <Arduino.h>
#include <WiFi.h>
#include <Wire.h>
#include <esp_now.h>
#include <esp_wifi.h>
```

Include dependency graph for main.cpp:



Classes

- struct struct_message

    Data structure for ESP-NOW transmission.

Typedefs

- typedef struct struct_message struct_message

Functions

- Adafruit_NeoPixel pixel (1, NEOPIXEL_PIN, NEO_GRB+NEO_KHZ800)

  NeoPixel instance.
- void OnDataSent (const uint8_t ∗mac_addr, esp_now_send_status_t status)

  ESP-NOW send callback function.
- void setupEspNow ()

  Initializes ESP-NOW and registers the peer.
- void setEspNowChannel (uint8_t ch)

  Changes the WiFi channel.
- void fillMeasurement ()

  Reads sensors and populates the `myData` structure.
- bool trySendOnChannel (uint8_t channel)

  Attempts to send data on a specific WiFi channel using "Burst Mode".
- void goToDeepSleep ()

  Prepares hardware for sleep and enters Deep Sleep.
- void setup ()

  Main setup routine.
- void loop ()

Variables

- const int SDA_PIN = 20

  I2C SDA Pin.
- const int SCL_PIN = 10

  I2C SCL Pin.
- const int NEOPIXEL_PIN = 5

  WS2812B NeoPixel control pin.
- const int UV_SENSOR_PIN = 1

  Analog pin for UV sensor.
- const uint8_t BMP_ADDR = 0x76

  I2C address for BME280 sensor.
- const uint64_t SLEEP_TIME_SECONDS = 60

  Time to sleep between measurements in seconds.
- const unsigned long MAX_RETRY_TIME_MS = 20000

  Maximum time allowed to try finding a receiver (ms)
- const uint8_t MAX_WIFI_CHANNEL = 13

  Highest allowed WiFi channel.
- uint8_t broadcastAddress [ ] = {0xf4, 0x65, 0x0b, 0xe9, 0x77, 0x78}

  Target MAC address (Broadcast).
- Adafruit_BME280 bme

  BME280 sensor instance.
- bool bmpOk = false

  Flag indicating if BME280 initialized successfully.
- RTC_DATA_ATTR uint8_t savedChannel = 1

  Last successful WiFi channel.
- struct_message telemetryData
- esp_now_peer_info_t peerInfo
- volatile bool transmissionFinished = false
- volatile bool transmissionSuccess = false

4.1.1 Detailed Description

ESP32 Sensor Node (BME280 + UV) transmitting via ESP-NOW.

- This program reads data from BME280 (I2C) and an analog UV sensor, then broadcasts the data using ESP-NOW with channel scanning capability. Ideally suited for battery-powered operation using Deep Sleep.

  Author

- Łukasz Wolf

  Date

  2023-10-27

4.1.2 Typedef Documentation

4.1.2.1 struct_message

```
typedef struct struct_message struct_message
```

4.1.3 Function Documentation

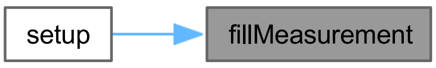
4.1.3.1 fillMeasurement()

```
void fillMeasurement ()
```

Reads sensors and populates the `myData` structure.

- Reads Temperature, Humidity, Pressure from BME280. Reads UV raw value from Analog Pin. Prints debug info to Serial.

Here is the caller graph for this function:

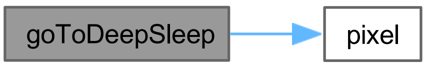


4.1.3.2 goToDeepSleep()

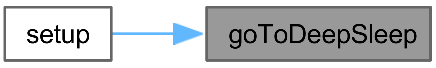
```
void goToDeepSleep ()
```

Prepares hardware for sleep and enters Deep Sleep.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.3.3 loop()

```
void loop ()
```

4.1.3.4 OnDataSent()

```
void OnDataSent (
            const uint8_t * mac_addr,
            esp_now_send_status_t status)
```

ESP-NOW send callback function.

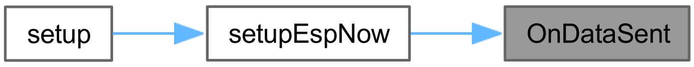
- Triggered when data is sent. Updates status flags.

Parameters

mac_addr	Destination MAC address.
status	Status of transmission (ESP_NOW_SEND_SUCCESS or FAIL).

-

Here is the caller graph for this function:

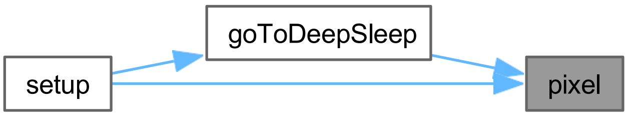


4.1.3.5 pixel()

```
Adafruit_NeoPixel pixel (
            1 ,
            NEOPIXEL_PIN ,
            NEO_GRB+ NEO_KHZ800)
```

NeoPixel instance.

Here is the caller graph for this function:



4.1.3.6 setEspNowChannel()

```
void setEspNowChannel (
            uint8_t ch)
```

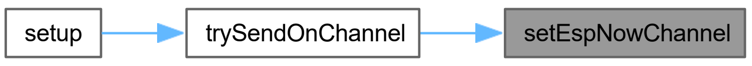
Changes the WiFi channel.

Parameters

	ch	Channel number (1-13).

-

Here is the caller graph for this function:



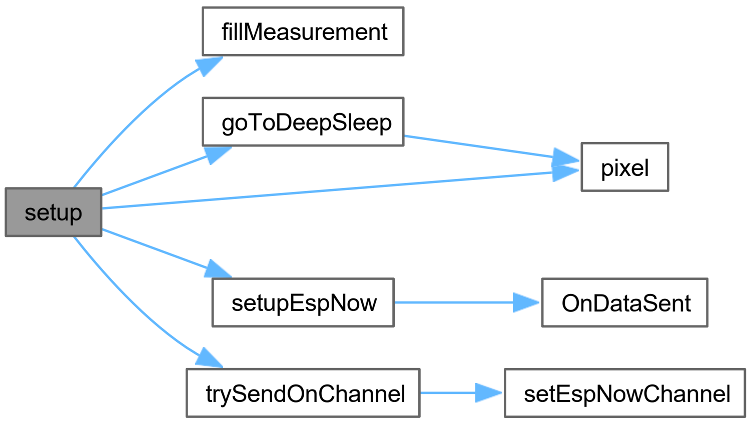
4.1.3.7 setup()

```
void setup ()
```

Main setup routine.

- Runs once per wake-up cycle.

Here is the call graph for this function:



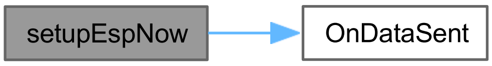
4.1.3.8 setupEspNow()

```
void setupEspNow ()
```

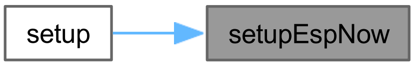
Initializes ESP-NOW and registers the peer.

- Sets WiFi mode to Station, disconnects from APs, and adds the broadcast peer.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.3.9 trySendOnChannel()

```
bool trySendOnChannel (
             uint8_t channel)
```

Attempts to send data on a specific WiFi channel using "Burst Mode".

- Sends up to 5 packets rapidly to increase the chance of delivery if the receiver is briefly busy.

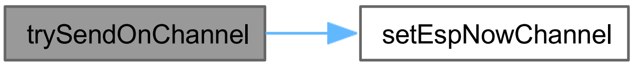
Parameters

	channel	The WiFi channel to transmit on.

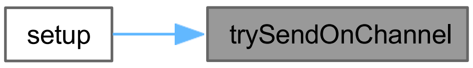
Returns

- true if ACK received (transmission successful).
  false if all attempts failed.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.4 Variable Documentation

4.1.4.1 bme

```
Adafruit_BME280 bme
```

BME280 sensor instance.

4.1.4.2 BMP_ADDR

```
const uint8_t BMP_ADDR = 0x76
```

I2C address for BME280 sensor.

4.1.4.3 bmpOk

```
bool bmpOk = false
```

Flag indicating if BME280 initialized successfully.

4.1.4.4 broadcastAddress

```
uint8_t broadcastAddress[] = {0xf4, 0x65, 0x0b, 0xe9, 0x77, 0x78}
```

Target MAC address (Broadcast).

*

Note

Specific address used: F4:65:0B:E9:77:78

4.1.4.5 MAX_RETRY_TIME_MS

```
const unsigned long MAX_RETRY_TIME_MS = 20000
```

Maximum time allowed to try finding a receiver (ms)

4.1.4.6 MAX_WIFI_CHANNEL

```
const uint8_t MAX_WIFI_CHANNEL = 13
```

Highest allowed WiFi channel.

4.1.4.7 telemetryData

```
struct_message telemetryData
```

4.1.4.8 NEOPIXEL_PIN

```
const int NEOPIXEL_PIN = 5
```

WS2812B NeoPixel control pin.

4.1.4.9 peerInfo

```
esp_now_peer_info_t peerInfo
```

4.1.4.10 savedChannel

```
RTC_DATA_ATTR uint8_t savedChannel = 1
```

Last successful WiFi channel.

*

Note

Stored in RTC memory to survive Deep Sleep.

4.1.4.11 SCL_PIN

```
const int SCL_PIN = 10
```

I2C SCL Pin.

4.1.4.12 SDA_PIN

```
const int SDA_PIN = 20
```

I2C SDA Pin.

4.1.4.13 SLEEP_TIME_SECONDS

```
const uint64_t SLEEP_TIME_SECONDS = 60
```

Time to sleep between measurements in seconds.

4.1.4.14 transmissionFinished

```
volatile bool transmissionFinished = false
```

4.1.4.15 transmissionSuccess

```
volatile bool transmissionSuccess = false
```

4.1.4.16 UV_SENSOR_PIN

```
const int UV_SENSOR_PIN = 1
```

Analog pin for UV sensor.