

## Projekt SW

Generated by Doxygen 1.8.16



<b>1 Data Structure Index</b>	<b>1</b>
1.1 Data Structures . . . . .	1
<b>2 File Index</b>	<b>3</b>
2.1 File List . . . . .	3
<b>3 Data Structure Documentation</b>	<b>5</b>
3.1 controlPeripheralStruct Struct Reference . . . . .	5
3.1.1 Detailed Description . . . . .	5
3.1.2 Field Documentation . . . . .	5
3.1.2.1 huart . . . . .	5
3.1.2.2 pDataToRxhuart . . . . .	5
3.1.2.3 sizepDataToRxHuart . . . . .	6
3.2 decoderStructure Struct Reference . . . . .	6
3.2.1 Detailed Description . . . . .	6
3.2.2 Field Documentation . . . . .	6
3.2.2.1 command . . . . .	6
3.2.2.2 decodeValue . . . . .	6
3.2.2.3 isEndDecode . . . . .	7
3.2.2.4 positionOfDecodeCommand . . . . .	7
3.2.2.5 sizeTableOdRejectedCommands . . . . .	7
3.2.2.6 tableOfRejectedCommands . . . . .	7
<b>4 File Documentation</b>	<b>9</b>
4.1 Other/Inc/decoder.h File Reference . . . . .	9
4.1.1 Detailed Description . . . . .	10
4.1.2 Macro Definition Documentation . . . . .	10
4.1.2.1 COMMAND_AT . . . . .	10
4.1.2.2 COMMAND_INVALID . . . . .	10
4.1.2.3 COMMAND_OFALL . . . . .	10
4.1.2.4 COMMAND_OFF0 . . . . .	11
4.1.2.5 COMMAND_OFF1 . . . . .	11
4.1.2.6 COMMAND_OFF2 . . . . .	11
4.1.2.7 COMMAND_OFF3 . . . . .	11
4.1.2.8 COMMAND_OFF4 . . . . .	11
4.1.2.9 COMMAND_OFF5 . . . . .	11
4.1.2.10 COMMAND_OFF6 . . . . .	11
4.1.2.11 COMMAND_OFF7 . . . . .	11
4.1.2.12 COMMAND_ON0 . . . . .	12
4.1.2.13 COMMAND_ON1 . . . . .	12
4.1.2.14 COMMAND_ON2 . . . . .	12
4.1.2.15 COMMAND_ON3 . . . . .	12
4.1.2.16 COMMAND_ON4 . . . . .	12

4.1.2.17	COMMAND_ON5	12
4.1.2.18	COMMAND_ON6	12
4.1.2.19	COMMAND_ON7	12
4.1.2.20	COMMAND_ONALL	13
4.1.2.21	DECODE_ANSWER_ERROR	13
4.1.2.22	DECODE_ANSWER_INIT_MALLOC_ERROR	13
4.1.2.23	DECODE_ANSWER_OK	13
4.1.2.24	DECODE_CLEAR_STRUCT_RESET_REJECTED_TABLE	13
4.1.2.25	DECODE_REJECT_COMMAND_OFF	13
4.1.2.26	DECODE_REJECT_COMMAND_ON	13
4.1.2.27	DECODER_MAX_LENGTH_COMMAND	13
4.1.3	Function Documentation	13
4.1.3.1	decoderAddNextSign()	13
4.1.3.2	decoderClearStructure()	14
4.1.3.3	decoderGetDecodeValue()	14
4.1.3.4	decoderGetDecodingEffect()	15
4.1.3.5	decoderInitStructure()	15
4.1.3.6	decoderIsEndDecode()	15
4.1.3.7	decoderSendAnswer()	16
4.1.3.8	decoderShowInfo()	16
4.2	Other/Inc/generalModule.h File Reference	16
4.2.1	Detailed Description	18
4.2.2	Macro Definition Documentation	18
4.2.2.1	SIZE_BUFFER_TO_RECEIVE_FROM_UART_INTERRUPT	18
4.2.2.2	SIZE_XQUEUEUARTPCR_X	18
4.2.2.3	SIZE_XQUEUEUARTPCTX	18
4.2.3	Function Documentation	18
4.2.3.1	__io_putchar()	18
4.2.3.2	generalModuleInit()	19
4.2.3.3	generalModuleUartReceiveIt()	19
4.2.3.4	generalModuleUartRxInterrupt()	20
4.2.3.5	switchGPIO()	20
4.2.3.6	vTaskDecodeData()	20
4.2.3.7	vTaskGPIOController()	21
4.2.3.8	vTaskSendRespons()	21
4.2.4	Variable Documentation	21
4.2.4.1	decoderGeneralStructure	21
4.2.4.2	peripheralStruct	21
4.2.4.3	xQueueCommand	21
4.2.4.4	xQueueUartPCR_X	22
4.2.4.5	xQueueUartPCTX	22
4.2.4.6	xTaskDecodeData	22

4.2.4.7 xTaskSendRespond . . . . .	22
4.2.4.8 xTaskWywołanie . . . . .	22
4.3 Other/Src/decoder.c File Reference . . . . .	22
4.3.1 Detailed Description . . . . .	23
4.3.2 Function Documentation . . . . .	23
4.3.2.1 decoderAddNextSign() . . . . .	23
4.3.2.2 decoderClearStructure() . . . . .	24
4.3.2.3 decoderGetDecodeValue() . . . . .	24
4.3.2.4 decoderGetDecodingEffect() . . . . .	24
4.3.2.5 decoderInitStructure() . . . . .	25
4.3.2.6 decoderIsEndDecode() . . . . .	25
4.3.2.7 decoderSendAnswer() . . . . .	25
4.3.2.8 decoderShowInfo() . . . . .	26
4.4 Other/Src/generalModule.c File Reference . . . . .	26
4.4.1 Detailed Description . . . . .	27
4.4.2 Function Documentation . . . . .	27
4.4.2.1 __io_putchar() . . . . .	27
4.4.2.2 generalModuleInit() . . . . .	27
4.4.2.3 generalModuleUartReceiveIt() . . . . .	28
4.4.2.4 generalModuleUartRxInterrupt() . . . . .	28
4.4.2.5 switchGPIO() . . . . .	29
4.4.2.6 vTaskDecodeData() . . . . .	29
4.4.2.7 vTaskGPIOController() . . . . .	29
4.4.2.8 vTaskSendRespons() . . . . .	29
<b>Index</b>	<b>31</b>



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<b>controlPeripheralStruct</b>	
Struktura kontrolujaca interfejs UART . . . . .	5
<b>decoderStructure</b>	
DecoderStructure glowna struktura dekodera . . . . .	6





## Chapter 2

# File Index

### 2.1 File List

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

Other/Inc/ <b>decoder.h</b>	
Plik naglowkowy dla modulu dekodera . . . . .	9
Other/Inc/ <b>generalModule.h</b>	
Plik naglowkowy modulu zawierajacy taski oraz kolejki . . . . .	16
Other/Src/ <b>decoder.c</b>	
Plik zrodlowy dekodera . . . . .	22
Other/Src/ <b>generalModule.c</b>	
Plik zrodlowy glownego modulu . . . . .	26



## Chapter 3

# Data Structure Documentation

### 3.1 controlPeripheralStruct Struct Reference

Struktura kontrolujaca interfejs UART.

```
#include <generalModule.h>
```

#### Data Fields

- UART\_HandleTypeDef \* **huart**
- uint8\_t \* **pDataToRxhuart**
- uint16\_t **sizepDataToRxHuart**

#### 3.1.1 Detailed Description

Struktura kontrolujaca interfejs UART.

#### 3.1.2 Field Documentation

##### 3.1.2.1 huart

```
UART_HandleTypeDef* controlPeripheralStruct::huart
```

Wskaźnik do struktury UART\_HandleTypeDef

##### 3.1.2.2 pDataToRxhuart

```
uint8_t* controlPeripheralStruct::pDataToRxhuart
```

Wskaźnik do bufora danych wykorzystywanych w przerwaniu odbiornika UART

### 3.1.2.3 sizepDataToRxHuart

```
uint16_t controlPeripheralStruct::sizepDataToRxHuart
```

Rozmiar bufora danych którego wskaźnikiem jest pDataToRxhuart

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

- Other/Inc/ **generalModule.h**

## 3.2 decoderStructure Struct Reference

**decoderStructure** (p. 6) glowna struktura dekodera

```
#include <decoder.h>
```

### Data Fields

- uint8\_t \* **tableOfRejectedCommands**
- uint8\_t **sizeTableOdRejectedCommands**
- uint32\_t **command** [ **DECODER\_MAX\_LENGTH\_COMMAND**]
- uint8\_t **positionOfDecodeCommand**
- uint8\_t **isEndDecode**
- uint8\_t **decodeValue**

### 3.2.1 Detailed Description

**decoderStructure** (p. 6) glowna struktura dekodera

### 3.2.2 Field Documentation

#### 3.2.2.1 command

```
uint32_t decoderStructure::command[ DECODER_MAX_LENGTH_COMMAND]
```

Tablica przechowująca komendy do odkodowania

#### 3.2.2.2 decodeValue

```
uint8_t decoderStructure::decodeValue
```

Zdekodowana wartosc

### 3.2.2.3 isEndDecode

```
uint8_t decoderStructure::isEndDecode
```

Zmienna informująca o zakończeniu dekodowania

### 3.2.2.4 positionOfDecodeCommand

```
uint8_t decoderStructure::positionOfDecodeCommand
```

Pozycja aktualnie odkodowywanej wartosci

### 3.2.2.5 sizeTableOfRejectedCommands

```
uint8_t decoderStructure::sizeTableOfRejectedCommands
```

Rozmiar tablicy odrzucanych komend jest sizeof(decoderArrayOfCommand)

### 3.2.2.6 tableOfRejectedCommands

```
uint8_t* decoderStructure::tableOfRejectedCommands
```

Wskaznik do tablicy ktora zawiera aktualnie odrzucone komendy

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

- Other/Inc/ **decoder.h**



## Chapter 4

# File Documentation

### 4.1 Other/Inc/decoder.h File Reference

Plik naglowkowy dla modulu dekodera.

#### Data Structures

- struct **decoderStructure**  
*decoderStructure* (p. 6) glowna struktura dekodera

#### Macros

- #define **DECODER\_MAX\_LENGTH\_COMMAND** 7
- #define **DECODE\_REJECT\_COMMAND\_ON** 1
- #define **DECODE\_REJECT\_COMMAND\_OFF** 0
- #define **DECODE\_ANSWER\_ERROR** 0x01
- #define **DECODE\_ANSWER\_OK** 0x02
- #define **DECODE\_ANSWER\_INIT\_MALLOC\_ERROR** 0x03
- #define **DECODE\_CLEAR\_STRUCT\_RESET\_REJECTED\_TABLE** 0x04
- #define **COMMAND\_ON0** 0x10
- #define **COMMAND\_ON1** 0x11
- #define **COMMAND\_ON2** 0x12
- #define **COMMAND\_ON3** 0x13
- #define **COMMAND\_ON4** 0x14
- #define **COMMAND\_ON5** 0x15
- #define **COMMAND\_ON6** 0x16
- #define **COMMAND\_ON7** 0x17
- #define **COMMAND\_OFF0** 0x20
- #define **COMMAND\_OFF1** 0x21
- #define **COMMAND\_OFF2** 0x22
- #define **COMMAND\_OFF3** 0x23
- #define **COMMAND\_OFF4** 0x24
- #define **COMMAND\_OFF5** 0x25
- #define **COMMAND\_OFF6** 0x26
- #define **COMMAND\_OFF7** 0x27
- #define **COMMAND\_ONALL** 0x30
- #define **COMMAND\_OFALL** 0x40
- #define **COMMAND\_AT** 0x50
- #define **COMMAND\_INVALID** 0x00

## Functions

- void **decoderInitStructure** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja inicjalizująca strukturę **decoderStructure** (p. 6).*
- void **decoderAddNextSign** ( **decoderStructure** \* **decoderGeneralStructure**, uint8\_t \*letter)  
*Funkcja dekodująca kolejny znak.*
- void **decoderSendAnswer** (uint8\_t what)  
*Funkcja Wyświetlająca stan dekodera Wykorzystywana tylko do testowania.*
- uint8\_t **decoderIsEndDecode** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja sprawdzająca czy zakończono proces dekodowania.*
- uint8\_t **decoderGetDecodeValue** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja zwracająca wartość zdekodowanej komendy.*
- void **decoderClearStructure** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja czyszcząca strukturę **decoderStructure** (p. 6) na potrzeby kolejnego dekodowania.*
- void **decoderShowInfo** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p. 6).*
- uint8\_t **decoderGetDecodingEffect** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p. 6).*

### 4.1.1 Detailed Description

Plik nagłówkowy dla modułu dekodera.

Author

Daniel Dunak

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 COMMAND\_AT

```
#define COMMAND_AT 0x50
```

Kod komendy AT

#### 4.1.2.2 COMMAND\_INVALID

```
#define COMMAND_INVALID 0x00
```

Kod nieznaney komendy

#### 4.1.2.3 COMMAND\_OFALL

```
#define COMMAND_OFALL 0x40
```

Kod komendy włączającej wszystkie diody



#### 4.1.2.4 COMMAND\_OFF0

```
#define COMMAND_OFF0 0x20
```

Kod komendy wylaczajacej diode nr.0

#### 4.1.2.5 COMMAND\_OFF1

```
#define COMMAND_OFF1 0x21
```

Kod komendy wylaczajacej diode nr.1

#### 4.1.2.6 COMMAND\_OFF2

```
#define COMMAND_OFF2 0x22
```

Kod komendy wylaczajacej diode nr.2

#### 4.1.2.7 COMMAND\_OFF3

```
#define COMMAND_OFF3 0x23
```

Kod komendy wylaczajacej diode nr.3

#### 4.1.2.8 COMMAND\_OFF4

```
#define COMMAND_OFF4 0x24
```

Kod komendy wylaczajacej diode nr.4

#### 4.1.2.9 COMMAND\_OFF5

```
#define COMMAND_OFF5 0x25
```

Kod komendy wylaczajacej diode nr.5

#### 4.1.2.10 COMMAND\_OFF6

```
#define COMMAND_OFF6 0x26
```

Kod komendy wylaczajacej diode nr.6

#### 4.1.2.11 COMMAND\_OFF7

```
#define COMMAND_OFF7 0x27
```

Kod komendy wylaczajacej diode nr.7

**4.1.2.12 COMMAND\_ON0**

```
#define COMMAND_ON0 0x10
```

Kod komendy włączającej diode nr.0

**4.1.2.13 COMMAND\_ON1**

```
#define COMMAND_ON1 0x11
```

Kod komendy włączającej diode nr.1

**4.1.2.14 COMMAND\_ON2**

```
#define COMMAND_ON2 0x12
```

Kod komendy włączającej diode nr.2

**4.1.2.15 COMMAND\_ON3**

```
#define COMMAND_ON3 0x13
```

Kod komendy włączającej diode nr.3

**4.1.2.16 COMMAND\_ON4**

```
#define COMMAND_ON4 0x14
```

Kod komendy włączającej diode nr.4

**4.1.2.17 COMMAND\_ON5**

```
#define COMMAND_ON5 0x15
```

Kod komendy włączającej diode nr.5

**4.1.2.18 COMMAND\_ON6**

```
#define COMMAND_ON6 0x16
```

Kod komendy włączającej diode nr.6

**4.1.2.19 COMMAND\_ON7**

```
#define COMMAND_ON7 0x17
```

Kod komendy włączającej diode nr.7

#### 4.1.2.20 COMMAND\_ONALL

```
#define COMMAND_ONALL 0x30
```

Kod komendy wylaczajacej wszystkie diody

#### 4.1.2.21 DECODE\_ANSWER\_ERROR

```
#define DECODE_ANSWER_ERROR 0x01
```

Wartosc oznaczajaca bladana komende po odkodowania

#### 4.1.2.22 DECODE\_ANSWER\_INIT\_MALLOCH\_ERROR

```
#define DECODE_ANSWER_INIT_MALLOCH_ERROR 0x03
```

Blad inicjalizacji przy uzyciu komendy malloc

#### 4.1.2.23 DECODE\_ANSWER\_OK

```
#define DECODE_ANSWER_OK 0x02
```

Wartosc oznaczajaca ze komenda zostala poprawnie rozpoznana

#### 4.1.2.24 DECODE\_CLEAR\_STRUCT\_RESET\_REJECTED\_TABLE

```
#define DECODE_CLEAR_STRUCT_RESET_REJECTED_TABLE 0x04
```

Wartosc oznaczajaca wymuszenie resetu tablicy dla dekodera

#### 4.1.2.25 DECODE\_REJECT\_COMMAND\_OFF

```
#define DECODE_REJECT_COMMAND_OFF 0
```

stan dla komendy która ma nie byc sprawdzana

#### 4.1.2.26 DECODE\_REJECT\_COMMAND\_ON

```
#define DECODE_REJECT_COMMAND_ON 1
```

stan dla komendy ktora ma byc sprawdzana

#### 4.1.2.27 DECODER\_MAX\_LENGTH\_COMMAND

```
#define DECODER_MAX_LENGTH_COMMAND 7
```

Maksymalny rozmiar rozkazu ktory moze zostac wyslany

### 4.1.3 Function Documentation

#### 4.1.3.1 decoderAddNextSign()

```
void decoderAddNextSign (
    decoderStructure * decoderGeneralStructure,
    uint8_t * letter )
```

Funkcja dekodujaca kolejny znak.

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
<i>letter</i>	wskaznik na kolejny znak do dekodowania

## Return values

<i>None</i>	
-------------	--

**4.1.3.2 decoderClearStructure()**

```
void decoderClearStructure (
    decoderStructure * decoderGeneralStructure )
```

Funkcja czyszczaca strukture **decoderStructure** (p. 6) na potrzeby kolejnego dekodowania.

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>None</i>	
-------------	--

**4.1.3.3 decoderGetDecodeValue()**

```
uint8_t decoderGetDecodeValue (
    decoderStructure * decoderGeneralStructure )
```

Funkcja zwracajaca wartosc zdekodowanej komendy.

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>uint8↔ _t</i>	zwraca wartosc komendy
----------------------	------------------------

#### 4.1.3.4 decoderGetDecodingEffect()

```
uint8_t decoderGetDecodingEffect (
    decoderStructure * decoderGeneralStructure )
```

Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p. 6).

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

##### Return values

<i>uint8_t</i>	dekoduje komende z tablicy char na wartosc liczbową
----------------	---

#### 4.1.3.5 decoderInitStructure()

```
void decoderInitStructure (
    decoderStructure * decoderGeneralStructure )
```

Funkcja inicalizująca strukturę **decoderStructure** (p. 6).

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

##### Return values

<i>None</i>	
-------------	--

#### 4.1.3.6 decoderIsEndDecode()

```
uint8_t decoderIsEndDecode (
    decoderStructure * decoderGeneralStructure )
```

Funkcja sprawdzająca czy zakończono proces dekodowania.

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>None</i>	
-------------	--

#### 4.1.3.7 decoderSendAnswer()

```
void decoderSendAnswer (
    uint8_t what )
```

Funkcja Wyświetlająca stan dekodera Wykorzystywana tylko do testowania.

## Parameters

<i>what</i>	wartosc
-------------	---------

## Return values

<i>None</i>	
-------------	--

#### 4.1.3.8 decoderShowInfo()

```
void decoderShowInfo (
    decoderStructure * decoderGeneralStructure )
```

Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p.6).

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p.6)
--------------------------------	---

## Return values

<i>None</i>	
-------------	--

## 4.2 Other/inc/generalModule.h File Reference

Plik naglowkowy modulu zawierajacy taski oraz kolejki.

```
#include "decoder.h"
#include "cmsis_os.h"
```

## Data Structures

- struct **controlPeripheralStruct**  
*Struktura kontrolująca interfejs UART.*

## Macros

- #define **SIZE\_BUFFER\_TO\_RECEIVE\_FROM\_UART\_INTERRUPT** 1
- #define **SIZE\_XQUEUEUARTPCR** 20
- #define **SIZE\_XQUEUEUARTPCTX** 150

## Functions

- void **vTaskGPIOController** ()  
*Task do kontroli wejsc/wyjsc GPIO mikrokontrolera.*
- void **vTaskDecodeData** ()  
*Task do dekodowania danych.*
- void **vTaskSendRespons** ()  
*Task do odpowiedzi po interfejsie UART.*
- void **generalModuleInit** ( **controlPeripheralStruct** \* **peripheralStruct**, UART\_HandleTypeDef \*huart)  
*Funkcja inicjalizująca wszystkie taski, kolejki oraz peryferia.*
- int **\_\_io\_putchar** (int ch)  
*Funkcja zastępująca wewnątrz funkcje \_\_io\_putchar wywoływana przez printf.*
- void **generalModuleUartRxInterrupt** ( **controlPeripheralStruct** \* **peripheralStruct**)  
*Funkcja dla przerwania od układu UART.*
- void **generalModuleUartReceiveIt** ( **controlPeripheralStruct** \* **peripheralStruct**)  
*Funkcja załączenia układu przerwania UART.*
- void **switchGPIO** (uint8\_t rozkaz)  
*Funkcja zmieniająca stany I/O GPIO w zależności od odebranego rozkazu.*

## Variables

- **decoderStructure** **decoderGeneralStructure**
- **controlPeripheralStruct** **peripheralStruct**
- xQueueHandle **xQueueUartPCR**
- xTaskHandle **xTaskDecodeData**
- xTaskHandle **xTaskSendRespond**
- xQueueHandle **xQueueUartPCTX**
- xQueueHandle **xQueueCommand**
- xTaskHandle **xTaskWywołanie**

### 4.2.1 Detailed Description

Plik naglowkowy modulu zawierajacy taski oraz kolejki.

#### Author

Daniel Dunak, Michał Kuska

#### Attention

PC1 -> Dioda nr.0  
PC3 -> Dioda nr.1  
PA1 -> Dioda nr.2  
PA3 -> Dioda nr.3  
PA5 -> Dioda nr.4  
PA7 -> Dioda nr.5  
PC5 -> Dioda nr.6  
PB1 -> Dioda nr.7  
PC12 -> UART5\_TX  
PD2 -> UART5\_RX

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 SIZE\_BUFFER\_TO\_RECEIVE\_FROM\_UART\_INNTERUPT

```
#define SIZE_BUFFER_TO_RECEIVE_FROM_UART_INNTERUPT 1
```

Rozmiar buforu do odbierania danych z przerwania od odbiornika UART poprzez interfejs HAL

#### 4.2.2.2 SIZE\_XQUEUEUARTPCR

```
#define SIZE_XQUEUEUARTPCR 20
```

Rozmiar kolejki do dekodera komend

#### 4.2.2.3 SIZE\_XQUEUEUARTPCTX

```
#define SIZE_XQUEUEUARTPCTX 150
```

- Rozmiar kolejki do wysyłania danych poprzez interfejs UART

### 4.2.3 Function Documentation

#### 4.2.3.1 \_\_io\_putchar()

```
int __io_putchar (  
    int ch )
```

Funkcja zastępująca wewnątrz funkcje \_\_io\_putchar wywoływaną przez printf.



## Parameters

<i>ch</i>	reprezentacja liczba znaku char
-----------	---------------------------------

## Return values

<i>None</i>	
-------------	--

### 4.2.3.2 generalModuleInit()

```
void generalModuleInit (
    controlPeripheralStruct * peripheralStruct,
    UART_HandleTypeDef * huart )
```

Funkcja inicjalizująca wszystkie taski, kolejki oraz peryferia.

## Parameters

<i>peripheralStruct</i>	wskaznik do peripheralStruct
<i>huart</i>	wskaznik na strukture huart

## Return values

<i>None</i>	
-------------	--

### 4.2.3.3 generalModuleUartReceiveIt()

```
void generalModuleUartReceiveIt (
    controlPeripheralStruct * peripheralStruct )
```

Funkcja załączenia układu przerwania UART.

## Parameters

<i>peripheralStruct</i>	wskaznik do struktury <b>controlPeripheralStruct</b> (p. 5)
-------------------------	---

## Return values

<i>None</i>	
-------------	--

#### 4.2.3.4 generalModuleUartRxInterrupt()

```
void generalModuleUartRxInterrupt (
    controlPeripheralStruct * peripheralStruct )
```

Funkcja dla przerwania od układu UART.

##### Parameters

<i>peripheralStruct</i>	wskaźnik do struktury <b>controlPeripheralStruct</b> (p. 5)
-------------------------	---

##### Return values

<i>None</i>	
-------------	--

#### 4.2.3.5 switchGPIO()

```
void switchGPIO (
    uint8_t rozkaz )
```

Funkcja zmieniająca stany I/O GPIO w zależności od odebranego rozkazu.

##### Parameters

<i>rozkaz</i>	Komenda dekodera
---------------	------------------

##### Return values

<i>None</i>	
-------------	--

#### 4.2.3.6 vTaskDecodeData()

```
void vTaskDecodeData ( )
```

Task do dekodowania danych.

##### Return values

<i>None</i>	
-------------	--

#### 4.2.3.7 vTaskGPIOController()

```
void vTaskGPIOController ( )
```

Task do kontroli wejsc/wyjsc GPIO mikrokontrolera.

Return values

<i>None</i>	
-------------	--

#### 4.2.3.8 vTaskSendRespons()

```
void vTaskSendRespons ( )
```

Task do odpowiedzi po interfejsie UART.

Return values

<i>None</i>	
-------------	--

### 4.2.4 Variable Documentation

#### 4.2.4.1 decoderGeneralStructure

```
decoderStructure decoderGeneralStructure
```

Struktura dekodera

#### 4.2.4.2 peripheralStruct

```
controlPeripheralStruct peripheralStruct
```

Struktura kontrolera UART

#### 4.2.4.3 xQueueCommand

```
xQueueHandle xQueueCommand
```

Kolejka do przekazywania rozkazów wymagających zmiany Diod

#### 4.2.4.4 xQueueUartPCRx

xQueueHandle xQueueUartPCRx

Kolejki do dokodera

#### 4.2.4.5 xQueueUartPCTx

xQueueHandle xQueueUartPCTx

Kolejka do wysyłania odpowiedzi

#### 4.2.4.6 xTaskDecodeData

xTaskHandle xTaskDecodeData

Task dekodera

#### 4.2.4.7 xTaskSendRespond

xTaskHandle xTaskSendRespond

Task wysyłania odpowiedzi

#### 4.2.4.8 xTaskWywołanie

xTaskHandle xTaskWywołanie

Task do konfiguracji wyjści GPIO

### 4.3 Other/Src/decoder.c File Reference

Plik zrodlowy dekodera.

```
#include <stdio.h>
#include "stm32f4xx_hal.h"
#include <stdlib.h>
#include "decoder.h"
```

## Functions

- void **decoderInitStructure** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja inicalizująca strukture **decoderStructure** (p. 6).*
- void **decoderAddNextSign** ( **decoderStructure** \* **decoderGeneralStructure**, uint8\_t \*letter)  
*Funkcja dekodująca kolejny znak.*
- void **decoderSendAnswer** (uint8\_t what)  
*Funkcja Wyświetlająca stan dekodera Wykorzystywana tylko do testowania.*
- uint8\_t **decoderIsEndDecode** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja sprawdzająca czy zakończono proces dekodowania.*
- uint8\_t **decoderGetDecodeValue** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja zwracająca wartość zdekodowanej komendy.*
- void **decoderClearStructure** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja czyszcząca strukture **decoderStructure** (p. 6) na potrzeby kolejnego dekodowania.*
- void **decoderShowInfo** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p. 6).*
- uint8\_t **decoderGetDecodingEffect** ( **decoderStructure** \* **decoderGeneralStructure**)  
*Funkcja wyświetlająca informacje o strukturze **decoderStructure** (p. 6).*

### 4.3.1 Detailed Description

Plik źródłowy dekodera.

Author

Daniel Dunak, Michał Kuska

### 4.3.2 Function Documentation

#### 4.3.2.1 decoderAddNextSign()

```
void decoderAddNextSign (
    decoderStructure * decoderGeneralStructure,
    uint8_t * letter )
```

Funkcja dekodująca kolejny znak.

Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
<i>letter</i>	wskaznik na kolejny znak do dekodowania

Return values

<i>None</i>	
-------------	--

#### 4.3.2.2 decoderClearStructure()

```
void decoderClearStructure (
    decoderStructure * decoderGeneralStructure )
```

Funkcja czyszczaca strukture **decoderStructure** (p. 6) na potrzeby kolejnego dekodowania.

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

##### Return values

<i>None</i>	
-------------	--

#### 4.3.2.3 decoderGetDecodeValue()

```
uint8_t decoderGetDecodeValue (
    decoderStructure * decoderGeneralStructure )
```

Funkcja zwracajaca wartosc zdekodowanej komendy.

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

##### Return values

<i>uint8_t</i>	zwraca wartosc komendy
----------------	------------------------

#### 4.3.2.4 decoderGetDecodingEffect()

```
uint8_t decoderGetDecodingEffect (
    decoderStructure * decoderGeneralStructure )
```

Funkcja wyswietlajaca informacje o strukturze **decoderStructure** (p. 6).

##### Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<code>uint8_t</code>	dekoduje komende z tablicy char na wartosc liczbową
----------------------	---

#### 4.3.2.5 decoderInitStructure()

```
void decoderInitStructure (
    decoderStructure * decoderGeneralStructure )
```

Funkcja inicalizująca strukture **decoderStructure** (p. 6).

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>None</i>	
-------------	--

#### 4.3.2.6 decoderIsEndDecode()

```
uint8_t decoderIsEndDecode (
    decoderStructure * decoderGeneralStructure )
```

Funkcja sprawdzająca czy zakonczono proces dekodowania.

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>None</i>	
-------------	--

#### 4.3.2.7 decoderSendAnswer()

```
void decoderSendAnswer (
    uint8_t what )
```

Funkcja Wyszylajaca stan dekodera Wykorzystywana tylko do testowania.

## Parameters

<i>what</i>	wartosc
-------------	---------

## Return values

<i>None</i>	
-------------	--

## 4.3.2.8 decoderShowInfo()

```
void decoderShowInfo (
    decoderStructure * decoderGeneralStructure )
```

Funkcja wyswietlajaca informacje o strukturze **decoderStructure** (p. 6).

## Parameters

<i>decoderGeneralStructure</i>	wskaznik do <b>decoderStructure</b> (p. 6)
--------------------------------	--

## Return values

<i>None</i>	
-------------	--

## 4.4 Other/Src/generalModule.c File Reference

Plik zrodlowy glownego modulu.

```
#include <stdio.h>
#include <stdlib.h>
#include "stm32f4xx_hal.h"
#include "generalModule.h"
```

## Functions

- void **generalModuleInit** ( **controlPeripheralStruct** \* **peripheralStruct**, UART\_HandleTypeDef \*huart)  
*Funkcja inicalizujaca wszystkie taski, kolejki oraz peryferia.*
- void **generalModuleUartRxInterrupt** ( **controlPeripheralStruct** \* **peripheralStruct**)  
*Funkcja dla przerwania od ukladu UART.*
- void **generalModuleUartReceiveIt** ( **controlPeripheralStruct** \* **peripheralStruct**)  
*Funkcja zalaczenia ukladu przerwania UART.*
- void **vTaskDecodeData** ()  
*Task do dekodowania danych.*
- void **vTaskGPIOController** ()



*Task do kontroli wejsc/wyjsc GPIO mikrokontrolera.*

- void **vTaskSendRespons** ()

*Task do odpowiedzi po interfejsie UART.*

- int **\_\_io\_putchar** (int ch)

*Funkcja zastępująca wewnątrz funkcje \_\_io\_putchar wywoływana przez printf.*

- void **switchGPIO** (uint8\_t rozkaz)

*Funkcja zmieniająca stany I/O GPIO w zależności od odebranego rozkazu.*

### 4.4.1 Detailed Description

Plik źródłowy głównego modułu.

Author

Daniel Dunak, Michał Kuska

### 4.4.2 Function Documentation

#### 4.4.2.1 \_\_io\_putchar()

```
int __io_putchar (
    int ch )
```

Funkcja zastępująca wewnątrz funkcje \_\_io\_putchar wywoływana przez printf.

Parameters

<i>ch</i>	reprezentacja liczba znaku char
-----------	---------------------------------

Return values

<i>None</i>	
-------------	--

#### 4.4.2.2 generalModuleInit()

```
void generalModuleInit (
    controlPeripheralStruct * peripheralStruct,
    UART_HandleTypeDef * huart )
```

Funkcja inicjalizująca wszystkie taski, kolejki oraz peryferia.

## Parameters

<i>peripheralStruct</i>	wskaznik do peripheralStruct
<i>huart</i>	wskaznik na strukture huart

## Return values

<i>None</i>	
-------------	--

**4.4.2.3 generalModuleUartReceiveIt()**

```
void generalModuleUartReceiveIt (
    controlPeripheralStruct * peripheralStruct )
```

Funkcja załączenia układu przerwania UART.

## Parameters

<i>peripheralStruct</i>	wskaznik do struktury <b>controlPeripheralStruct</b> (p. 5)
-------------------------	---

## Return values

<i>None</i>	
-------------	--

**4.4.2.4 generalModuleUartRxInterrupt()**

```
void generalModuleUartRxInterrupt (
    controlPeripheralStruct * peripheralStruct )
```

Funkcja dla przerwania od układu UART.

## Parameters

<i>peripheralStruct</i>	wskaznik do struktury <b>controlPeripheralStruct</b> (p. 5)
-------------------------	---

## Return values

<i>None</i>	
-------------	--

#### 4.4.2.5 switchGPIO()

```
void switchGPIO (
    uint8_t rozkaz )
```

Funkcja zmieniajaca stany I/O GPIO w zaleznosci od odebranego rozkazu.

##### Parameters

<i>rozkaz</i>	Komenda dekodera
---------------	------------------

##### Return values

<i>None</i>	
-------------	--

#### 4.4.2.6 vTaskDecodeData()

```
void vTaskDecodeData ( )
```

Task do dekodowania danych.

##### Return values

<i>None</i>	
-------------	--

#### 4.4.2.7 vTaskGPIOController()

```
void vTaskGPIOController ( )
```

Task do kontroli wejsc/wyjsc GPIO mikrokontrolera.

##### Return values

<i>None</i>	
-------------	--

#### 4.4.2.8 vTaskSendRespons()

```
void vTaskSendRespons ( )
```

Task do odpowiedzi po interfejsie UART.

Return values

<i>None</i>	
-------------	--

# Index

- `__io_putchar`
    - `generalModule.c`, 27
    - `generalModule.h`, 18
  - `command`
    - `decoderStructure`, 6
  - `COMMAND_AT`
    - `decoder.h`, 10
  - `COMMAND_INVALID`
    - `decoder.h`, 10
  - `COMMAND_OFALL`
    - `decoder.h`, 10
  - `COMMAND_OFF0`
    - `decoder.h`, 10
  - `COMMAND_OFF1`
    - `decoder.h`, 11
  - `COMMAND_OFF2`
    - `decoder.h`, 11
  - `COMMAND_OFF3`
    - `decoder.h`, 11
  - `COMMAND_OFF4`
    - `decoder.h`, 11
  - `COMMAND_OFF5`
    - `decoder.h`, 11
  - `COMMAND_OFF6`
    - `decoder.h`, 11
  - `COMMAND_OFF7`
    - `decoder.h`, 11
  - `COMMAND_ON0`
    - `decoder.h`, 11
  - `COMMAND_ON1`
    - `decoder.h`, 12
  - `COMMAND_ON2`
    - `decoder.h`, 12
  - `COMMAND_ON3`
    - `decoder.h`, 12
  - `COMMAND_ON4`
    - `decoder.h`, 12
  - `COMMAND_ON5`
    - `decoder.h`, 12
  - `COMMAND_ON6`
    - `decoder.h`, 12
  - `COMMAND_ON7`
    - `decoder.h`, 12
  - `COMMAND_ONALL`
    - `decoder.h`, 12
- `controlPeripheralStruct`, 5
    - `huart`, 5
    - `pDataToRxhuart`, 5
    - `sizepDataToRxHuart`, 5
  - `DECODE_ANSWER_ERROR`
    - `decoder.h`, 13
  - `DECODE_ANSWER_INIT_MALLOC_ERROR`
    - `decoder.h`, 13
  - `DECODE_ANSWER_OK`
    - `decoder.h`, 13
  - `DECODE_CLEAR_STRUCT_RESET_REJECTED_TABLE`
    - `decoder.h`, 13
  - `DECODE_REJECT_COMMAND_OFF`
    - `decoder.h`, 13
  - `DECODE_REJECT_COMMAND_ON`
    - `decoder.h`, 13
  - `decoder.c`
    - `decoderAddNextSign`, 23
    - `decoderClearStructure`, 24
    - `decoderGetDecodeValue`, 24
    - `decoderGetDecodingEffect`, 24
    - `decoderInitStructure`, 25
    - `decoderIsEndDecode`, 25
    - `decoderSendAnswer`, 25
    - `decoderShowInfo`, 26
  - `decoder.h`
    - `COMMAND_AT`, 10
    - `COMMAND_INVALID`, 10
    - `COMMAND_OFALL`, 10
    - `COMMAND_OFF0`, 10
    - `COMMAND_OFF1`, 11
    - `COMMAND_OFF2`, 11
    - `COMMAND_OFF3`, 11
    - `COMMAND_OFF4`, 11
    - `COMMAND_OFF5`, 11
    - `COMMAND_OFF6`, 11
    - `COMMAND_OFF7`, 11
    - `COMMAND_ON0`, 11
    - `COMMAND_ON1`, 12
    - `COMMAND_ON2`, 12
    - `COMMAND_ON3`, 12
    - `COMMAND_ON4`, 12
    - `COMMAND_ON5`, 12
    - `COMMAND_ON6`, 12
    - `COMMAND_ON7`, 12
    - `COMMAND_ONALL`, 12
    - `DECODE_ANSWER_ERROR`, 13
    - `DECODE_ANSWER_INIT_MALLOC_ERROR`, 13
    - `DECODE_ANSWER_OK`, 13
    - `DECODE_CLEAR_STRUCT_RESET_REJECTED_TABLE`, 13
    - `DECODE_REJECT_COMMAND_OFF`, 13
    - `DECODE_REJECT_COMMAND_ON`, 13

- DECODER\_MAX\_LENGTH\_COMMAND, 13
- decoderAddNextSign, 13
- decoderClearStructure, 14
- decoderGetDecodeValue, 14
- decoderGetDecodingEffect, 14
- decoderInitStructure, 15
- decoderIsEndDecode, 15
- decoderSendAnswer, 16
- decoderShowInfo, 16
- DECODER\_MAX\_LENGTH\_COMMAND
- decoder.h, 13
- decoderAddNextSign
  - decoder.c, 23
  - decoder.h, 13
- decoderClearStructure
  - decoder.c, 24
  - decoder.h, 14
- decoderGeneralStructure
  - generalModule.h, 21
- decoderGetDecodeValue
  - decoder.c, 24
  - decoder.h, 14
- decoderGetDecodingEffect
  - decoder.c, 24
  - decoder.h, 14
- decoderInitStructure
  - decoder.c, 25
  - decoder.h, 15
- decoderIsEndDecode
  - decoder.c, 25
  - decoder.h, 15
- decoderSendAnswer
  - decoder.c, 25
  - decoder.h, 16
- decoderShowInfo
  - decoder.c, 26
  - decoder.h, 16
- decoderStructure, 6
  - command, 6
  - decodeValue, 6
  - isEndDecode, 6
  - positionOfDecodeCommand, 7
  - sizeTableOfRejectedCommands, 7
  - tableOfRejectedCommands, 7
- decodeValue
  - decoderStructure, 6
- generalModule.c
  - \_\_io\_putchar, 27
  - generalModuleInit, 27
  - generalModuleUartReceiveIt, 28
  - generalModuleUartRxInterrupt, 28
  - switchGPIO, 28
  - vTaskDecodeData, 29
  - vTaskGPIOController, 29
  - vTaskSendRespons, 29
- generalModule.h
  - \_\_io\_putchar, 18
  - decoderGeneralStructure, 21
  - generalModuleInit, 19
  - generalModuleUartReceiveIt, 19
  - generalModuleUartRxInterrupt, 19
  - peripheralStruct, 21
  - SIZE\_BUFFER\_TO\_RECEIVE\_FROM\_UART\_ININTERUPT, 18
  - SIZE\_XQUEUEUARTPCR, 18
  - SIZE\_XQUEUEUARTPCTX, 18
  - switchGPIO, 20
  - vTaskDecodeData, 20
  - vTaskGPIOController, 20
  - vTaskSendRespons, 21
  - xQueueCommand, 21
  - xQueueUartPCR, 21
  - xQueueUartPCTX, 22
  - xTaskDecodeData, 22
  - xTaskSendRespond, 22
  - xTaskWywolanie, 22
- generalModuleInit
  - generalModule.c, 27
  - generalModule.h, 19
- generalModuleUartReceiveIt
  - generalModule.c, 28
  - generalModule.h, 19
- generalModuleUartRxInterrupt
  - generalModule.c, 28
  - generalModule.h, 19
- huart
  - controlPeripheralStruct, 5
- isEndDecode
  - decoderStructure, 6
- Other/Inc/decoder.h, 9
- Other/Inc/generalModule.h, 16
- Other/Src/decoder.c, 22
- Other/Src/generalModule.c, 26
- pDataToRxhuart
  - controlPeripheralStruct, 5
- peripheralStruct
  - generalModule.h, 21
- positionOfDecodeCommand
  - decoderStructure, 7
- SIZE\_BUFFER\_TO\_RECEIVE\_FROM\_UART\_ININTERUPT
  - generalModule.h, 18
- SIZE\_XQUEUEUARTPCR
  - generalModule.h, 18
- SIZE\_XQUEUEUARTPCTX
  - generalModule.h, 18
- sizepDataToRxHuart
  - controlPeripheralStruct, 5
- sizeTableOfRejectedCommands
  - decoderStructure, 7
- switchGPIO
  - generalModule.c, 28
  - generalModule.h, 20

tableOfRejectedCommands  
    decoderStructure, 7

vTaskDecodeData  
    generalModule.c, 29  
    generalModule.h, 20

vTaskGPIOController  
    generalModule.c, 29  
    generalModule.h, 20

vTaskSendRespons  
    generalModule.c, 29  
    generalModule.h, 21

xQueueCommand  
    generalModule.h, 21

xQueueUartPCRx  
    generalModule.h, 21

xQueueUartPCTx  
    generalModule.h, 22

xTaskDecodeData  
    generalModule.h, 22

xTaskSendRespond  
    generalModule.h, 22

xTaskWywołanie  
    generalModule.h, 22