Portable Game Console

Wygenerowano przez Doxygen 1.8.13

Spis treści

1 Dokumentacja struktur danych

1.1 Dokumentacja struktury Button

Pola danych

- char ∗ **string**
- uint16_t **textColor**
- uint16_t **buttonColor**
- uint8_t **x**
- uint16_t **y**
- uint8_t **width**
- uint16_t **height**
- Option **earlierOption**
- void(∗ **Display**)(struct Button ∗)

1.2 Dokumentacja struktury Element

Pola danych

- uint8_t **x**
- uint8_t **y**
- Color **color**
- Direction **newDirection**
- Direction **oldDirection**

1.3 Dokumentacja struktury TouchPoint

Pola danych

- uint8_t **x**
- uint16_t **y**
- uint32_t **xad**
- uint32_t **yad**
- uint8_t **status**

1.4 Dokumentacja struktury Vector

Pola danych

- int8_t **x**
- int8_t **y**

2 Dokumentacja plików

2.1 Dokumentacja pliku AccelerometerController.c

Funkcje

- void AccelerometerInit (void)
- int8_t AccelerometerGetAngle (Axis axis)
- Direction AccelerometerGetDirection (void)

2.1.1 Opis szczegółowy

Autor

Mateusz Chudy

2.1.2 Dokumentacja funkcji

2.1.2.1 AccelerometerGetAngle()

```
int8_t AccelerometerGetAngle (
            Axis axis )
```

Pobiera wartosc proporcjonalna do kata nachylenia do danej osi.

Parametry

axis	- indeks rejestru, pod ktorym znajduje sie wartosc dla danej osi

Zwraca

wartosc proporcjonalna do kata nachylenia do danej osi

2.1.2.2 AccelerometerGetDirection()

```
Direction AccelerometerGetDirection (
            void  )
```

Pobiera kierunek nachylenia.

Zwraca

kierunek nachylenia

2.2 Dokumentacja pliku AccelerometerController.h

Definicje

- #define **X0** 0
- #define **Y0** -5
- #define **WRITE** 0x3A
- #define **READ** 0x3B
- #define **ACCE_SENSITIVITY** 10

Wyliczenia

- enum Axis { **AXIS_X** = 0x29, **AXIS_Y** = 0x2B, **AXIS_Z** = 0x2D }
- enum Direction {
  CONST, **UP**, **DOWN**, **LEFT**,
  RIGHT }

Funkcje

- void AccelerometerInit (void)
- int8_t AccelerometerGetAngle (Axis axis)
- Direction AccelerometerGetDirection (void)

2.2.1 Opis szczegółowy

Autor

Mateusz Chudy

2.2.2 Dokumentacja funkcji

2.2.2.1 AccelerometerGetAngle()

```
int8_t AccelerometerGetAngle (
            Axis axis )
```

Pobiera wartosc proporcjonalna do kata nachylenia do danej osi.

Parametry

axis	- indeks rejestru, pod ktorym znajduje sie wartosc dla danej osi

Zwraca

wartosc proporcjonalna do kata nachylenia do danej osi

2.2.2.2 AccelerometerGetDirection()

```
Direction AccelerometerGetDirection (
            void  )
```

Pobiera kierunek nachylenia.

Zwraca

kierunek nachylenia

2.3 Dokumentacja pliku ApplicationMenu.c

Funkcje

- void ApplicationMenuShow (void)

2.3.1 Opis szczegółowy

Autor

Mateusz Chudy

2.4 Dokumentacja pliku ApplicationMenu.h

Definicje

- #define **APP_PAINT_BUTTON_X1** 0
- #define **APP_PAINT_BUTTON_Y1** 0
- #define **APP_PAINT_BUTTON_X2** 240
- #define **APP_PAINT_BUTTON_Y2** 40
- #define **APP_PAINT_BUTTON_COLOR** DARKBLUE
- #define **APP_PAINT_TEXT_COLOR** WHITE
- #define **APP_SPIRIT_LEVEL_BUTTON_X1** 0
- #define **APP_SPIRIT_LEVEL_BUTTON_Y1** 40
- #define **APP_SPIRIT_LEVEL_BUTTON_X2** 240
- #define **APP_SPIRIT_LEVEL_BUTTON_Y2** 80
- #define **APP_SPIRIT_LEVEL_BUTTON_COLOR** 0X01CA
- #define **APP_SPIRIT_LEVEL_TEXT_COLOR** WHITE

Funkcje

- void ApplicationMenuShow (void)

2.4.1 Opis szczegółowy

Autor

Mateusz Chudy

2.5 Dokumentacja pliku Button.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- struct Button ∗ ButtonCreate (char ∗string, Option option, uint16_t textColor, uint16_t buttonColor, uint8_t x, uint16_t y, uint8_t width, uint16_t height)
- struct Button ∗∗ ButtonDestroy (struct Button ∗button)
- void ButtonDisplay (struct Button ∗button)
- bool ButtonCheck (struct Button ∗button)

2.5.1 Opis szczegółowy

Autor

Mateusz Chudy

2.5.2 Dokumentacja funkcji

2.5.2.1 ButtonCheck()

```
bool ButtonCheck (
            struct Button * button )
```

Sprawdza czy przycisk zostal nacisniety.

Parametry

button	- wskaznik na strukture przycisku

Zwraca

wartosc logiczna oznaczajaca czy przycisk zostal wcisniety

2.5.2.2 ButtonCreate()

```
struct Button* ButtonCreate (
            char * string,
            Option option,
            uint16_t textColor,
            uint16_t buttonColor,
            uint8_t x,
```

```
            uint16_t y,
            uint8_t width,
            uint16_t height )
```

Tworzy nowy przycik.

Parametry

string	- napis na przycisku
option	- opcja z ktorej powstal przycisk
textColor	- kolor tekstu
buttonColor	- kolor przycisku
x	- wspolrzedna x przycisku
y	- wspolrzedna y przycisku
width	- szerokosc przycisku
height	- wysokosc przycisku

Zwraca

wskaznik na strukture nowego przycisku

2.5.2.3 ButtonDestroy()

```
struct Button** ButtonDestroy (
            struct Button * button )
```

Niszczy przycisk.

Parametry

button	- wskaznik na strukture przycisku

Zwraca

wskaznik na strukture przycisku

2.5.2.4 ButtonDisplay()

```
void ButtonDisplay (
            struct Button * button )
```

Wyswietla przycisk.

Parametry

button	- wskaznik na strukture przycisku

2.6 Dokumentacja pliku Button.h

Struktury danych

- struct Button

Funkcje

- struct Button ∗ ButtonCreate (char ∗string, Option option, uint16_t textColor, uint16_t buttonColor, uint8_t x, uint16_t y, uint8_t width, uint16_t height)
- struct Button ∗∗ ButtonDestroy (struct Button ∗button)
- void ButtonDisplay (struct Button ∗button)
- bool ButtonCheck (struct Button ∗button)

2.6.1 Opis szczegółowy

Autor

Mateusz Chudy

2.6.2 Dokumentacja funkcji

2.6.2.1 ButtonCheck()

```
bool ButtonCheck (
            struct Button * button )
```

Sprawdza czy przycisk zostal nacisniety.

Parametry

button	- wskaznik na strukture przycisku

Zwraca

wartosc logiczna oznaczajaca czy przycisk zostal wcisniety

2.6.2.2 ButtonCreate()

```
struct Button* ButtonCreate (
            char * string,
            Option option,
            uint16_t textColor,
            uint16_t buttonColor,
            uint8_t x,
```

```
            uint16_t y,
            uint8_t width,
            uint16_t height )
```

Tworzy nowy przycik.

Parametry

string	- napis na przycisku
option	- opcja z ktorej powstal przycisk
textColor	- kolor tekstu
buttonColor	- kolor przycisku
x	- wspolrzedna x przycisku
y	- wspolrzedna y przycisku
width	- szerokosc przycisku
height	- wysokosc przycisku

Zwraca

wskaznik na strukture nowego przycisku

2.6.2.3 ButtonDestroy()

```
struct Button** ButtonDestroy (
            struct Button * button )
```

Niszczy przycisk.

Parametry

button	- wskaznik na strukture przycisku

Zwraca

wskaznik na strukture przycisku

2.6.2.4 ButtonDisplay()

```
void ButtonDisplay (
            struct Button * button )
```

Wyswietla przycisk.

Parametry

button	- wskaznik na strukture przycisku

2.7 Dokumentacja pliku Desktop.c

Funkcje

- void DesktopCheckSelect (void)
- void DesktopShow (void)
- void DesktopInvokeShowFunction (Option option)

2.7.1 Opis szczegółowy

Autor

Mateusz Chudy

2.7.2 Dokumentacja funkcji

2.7.2.1 DesktopInvokeShowFunction()

```
void DesktopInvokeShowFunction (
            Option option )
```

Wywoluje funkcje uruchomienia wybranej opcji menu.

Parametry

option	- opcja ktora uruchamiamy

2.8 Dokumentacja pliku Desktop.h

Definicje

- #define **DESK_MENU_BUTTON_X1** 0
- #define **DESK_MENU_BUTTON_Y1** 280
- #define **DESK_MENU_BUTTON_X2** 240
- #define **DESK_MENU_BUTTON_Y2** 320
- #define **DESK_MENU_BUTTON_COLOR** DARKBLUE
- #define **DESK_MENU_TEXT_COLOR** WHITE
- #define **BACK_BUTTON_X1** 0
- #define **BACK_BUTTON_Y1** 280
- #define **BACK_BUTTON_X2** 240
- #define **BACK_BUTTON_Y2** 320
- #define **BACK_BUTTON_COLOR** RED
- #define **BACK_TEXT_COLOR** WHITE

Funkcje

- void DesktopCheckSelect (void)
- void DesktopShow (void)
- void DesktopInvokeShowFunction (Option option)

Zmienne

- struct Button ∗ buttonArray [LAST]

2.8.1 Opis szczegółowy

Autor

Mateusz Chudy

2.8.2 Dokumentacja funkcji

2.8.2.1 DesktopInvokeShowFunction()

```
void DesktopInvokeShowFunction (
             Option option )
```

Wywoluje funkcje uruchomienia wybranej opcji menu.

Parametry

option	- opcja ktora uruchamiamy

2.9 Dokumentacja pliku DisplayTexts.h

Definicje

- #define **WELCOME_TEXT** "WELCOME!"
- #define **MENU_TEXT** "Menu"
- #define **GAMES_TEXT** "Games"
- #define **APPLICATION_TEXT** "Application"
- #define **PING_PONG_TEXT** "Ping Pong"
- #define **SNAKE_TEXT** "Snake"
- #define **PAINT_TEXT** "Paint"
- #define **BACK_TEXT** "Back"
- #define **EXIT_TEXT** "Exit"
- #define **UNKNOWN_OPTION_TEXT** "Unknown option."
- #define **GAME_OVER_TEXT** "Game Over"
- #define **GAME_RETRY_QUESTION** "Do you wanna play again?"
- #define **YES_TEXT** "Yes"
- #define **NO_TEXT** "No"

2.9.1 Opis szczegółowy

Autor

Mateusz Chudy

2.10 Dokumentacja pliku Font.c

Funkcje

- void FontDisplayChar (char character, uint16_t colorFront, uint16_t colorBackground, uint8_t x, uint16_t y)
- void FontDisplayString (char ∗string, uint16_t colorFront, uint16_t colorBackground, uint8_t x, uint16_t y)

Zmienne

- const uint8_t ascii [ ] PROGMEM

2.10.1 Opis szczegółowy

Autor

Mateusz Chudy

2.10.2 Dokumentacja funkcji

2.10.2.1 FontDisplayChar()

```
void FontDisplayChar (
            char character,
            uint16_t colorFront,
            uint16_t colorBackground,
            uint8_t x,
            uint16_t y )
```

Wyswietla znak.

Parametry

character	- wyswietlany znak
colorFront	- kolor czcionki
colorBackground	- kolor tla
x	- wspolrzedna x lewego gornego konca znaku
y	- wspolrzedna y lewego gornego konca znaku

2.10.2.2 FontDisplayString()

```
void FontDisplayString (
            char * string,
            uint16_t colorFront,
            uint16_t colorBackground,
```

```
            uint8_t x,
            uint16_t y )
```

Wyswietla napis.

Parametry

string	- wyswietlany napis
colorFront	- kolor czcionki
colorBackground	- kolor tla
x	- wspolrzedna x lewego gornego konca napisu
y	- wspolrzedna y lewego gornego konca napisu

2.11 Dokumentacja pliku Font.h

Definicje

- #define **CHAR_MAX_X** 8
- #define **CHAR_MAX_Y** 16
- #define **CHAR_MASK** 0x80
- #define **STRING_MAX_X** 230
- #define **STRING_MAX_Y** 300

Funkcje

- void FontDisplayChar (char character, uint16_t colorFront, uint16_t colorBackground, uint8_t x, uint16_t y)
- void FontDisplayString (char ∗string, uint16_t colorFront, uint16_t colorBackground, uint8_t x, uint16_t y)

Zmienne

- const uint8_t ascii [ ]

2.11.1 Opis szczegółowy

Autor

Mateusz Chudy

2.11.2 Dokumentacja funkcji

2.11.2.1 FontDisplayChar()

```
void FontDisplayChar (
            char character,
            uint16_t colorFront,
            uint16_t colorBackground,
            uint8_t x,
            uint16_t y )
```

Wyswietla znak.

Parametry

character	- wyswietlany znak
colorFront	- kolor czcionki
colorBackground	- kolor tla
x	- wspolrzedna x lewego gornego konca znaku
y	- wspolrzedna y lewego gornego konca znaku

2.11.2.2 FontDisplayString()

```
void FontDisplayString (
             char * string,
             uint16_t colorFront,
             uint16_t colorBackground,
             uint8_t x,
             uint16_t y )
```

Wyswietla napis.

Parametry

string	- wyswietlany napis
colorFront	- kolor czcionki
colorBackground	- kolor tla
x	- wspolrzedna x lewego gornego konca napisu
y	- wspolrzedna y lewego gornego konca napisu

2.12 Dokumentacja pliku GamesMenu.c

Funkcje

- void GamesMenuShow (void)

2.12.1 Opis szczegółowy

Autor

Mateusz Chudy

2.13 Dokumentacja pliku GamesMenu.h

Definicje

- #define **GAMES_PONG_BUTTON_X1** 0
- #define **GAMES_PONG_BUTTON_Y1** 0
- #define **GAMES_PONG_BUTTON_X2** 240

- #define **GAMES_PONG_BUTTON_Y2** 40
- #define **GAMES_PONG_BUTTON_COLOR** DARKBLUE
- #define **GAMES_PONG_TEXT_COLOR** WHITE
- #define **GAMES_SNAKE_BUTTON_X1** 0
- #define **GAMES_SNAKE_BUTTON_Y1** 40
- #define **GAMES_SNAKE_BUTTON_X2** 240
- #define **GAMES_SNAKE_BUTTON_Y2** 80
- #define **GAMES_SNAKE_BUTTON_COLOR** 0X01CA
- #define **GAMES_SNAKE_TEXT_COLOR** WHITE

Funkcje

- void GamesMenuShow (void)

2.13.1 Opis szczegółowy

Autor

Mateusz Chudy

2.14 Dokumentacja pliku GlobalErrors.h

Wyliczenia

- enum ExecuteResult { **EXECUTE_OK** = 0, **OUT_OF_RANGE** = 1, **INVALID_PARAMETER** = 2 }

2.14.1 Opis szczegółowy

Autor

Mateusz Chudy

2.15 Dokumentacja pliku GraphicsPrimitives.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- void GraphicsClearScreen (uint16_t color)
- void GraphicsClearArea (uint16_t color, uint8_t x, uint16_t y, uint8_t width, uint16_t height)
- void GraphicsSetCursor (uint8_t x, uint16_t y)
- void GraphicsPutPixel (uint16_t color, uint8_t x, uint16_t y)
- void GraphicsDrawBigDot (uint16_t color, uint8_t x, uint16_t y)
- void GraphicsDrawCircle (uint16_t color, uint8_t x, uint16_t y, uint8_t r)
- ExecuteResult GraphicsDrawLine (uint16_t color, uint8_t x1, uint16_t y1, uint8_t x2, uint16_t y2)
- void GraphicsDrawSimpleLine (uint16_t color, uint8_t x1, uint16_t y1, uint8_t x2, uint16_t y2)
- void GraphicsDisplayImage (const uint8_t *image, uint8_t x, uint16_t y, uint8_t width, uint16_t height)

2.15.1 Opis szczegółowy

Autor

Mateusz Chudy

2.15.2 Dokumentacja funkcji

2.15.2.1 GraphicsClearArea()

```
void GraphicsClearArea (
            uint16_t color,
            uint8_t x,
            uint16_t y,
            uint8_t width,
            uint16_t height )
```

Zamalowuje obszar na zadany kolor.

Parametry

color	- kolor zamalowania obszaru
x	- wspolrzedna x lewego gornego konca obszaru
y	- wspolrzedna y lewego gornego konca obszaru
width	- szerokosc obszaru
height	- wysokosc obszaru

2.15.2.2 GraphicsClearScreen()

```
void GraphicsClearScreen (
            uint16_t color )
```

Czysci ekran na zadany kolor.

Parametry

color	- kolor czyszczenia ekranu

2.15.2.3 GraphicsDisplayImage()

```
void GraphicsDisplayImage (
            const uint8_t * image,
            uint8_t x,
            uint16_t y,
```

```
            uint8_t width,
            uint16_t height )
```

Wyswietla obrazek.

Parametry

image	- obrazek
x	- wspolrzedna x lewego gornego konca obrazka
y	- wspolrzedna y lewego gornego konca obrazka
width	- szerokosc obrazka
height	- wysokosc obrazka

2.15.2.4 GraphicsDrawBigDot()

```
void GraphicsDrawBigDot (
            uint16_t color,
            uint8_t x,
            uint16_t y )
```

Wyswietla kropke 3x3 piksele o zadanych wspolrzednych.

Parametry

color	- kolor kropki
x	- wspolrzedna x
y	- wspolrzedna y

2.15.2.5 GraphicsDrawCircle()

```
void GraphicsDrawCircle (
            uint16_t color,
            uint8_t x,
            uint16_t y,
            uint8_t r )
```

Wyswietla okrag o srodku w zadanych wspolrzednych.

Parametry

color	- kolor kropki
x	- wspolrzedna x
y	- wspolrzedna y
r	- promien okregu

2.15.2.6 GraphicsDrawLine()

```
ExecuteResult GraphicsDrawLine (
            uint16_t color,
            uint8_t x1,
            uint16_t y1,
            uint8_t x2,
            uint16_t y2 )
```

Wyswietla linie pomiedzy dwoma punktami.

Parametry

color	- kolor lini
x1	- wspolrzedna x pierwszego punktu
y1	- wspolrzedna y pierwszego punktu
x2	- wspolrzedna x drugiego punktu
y2	- wspolrzedna y drugiego punktu

Zwraca

rezultat operacji

2.15.2.7 GraphicsDrawSimpleLine()

```
void GraphicsDrawSimpleLine (
            uint16_t color,
            uint8_t x1,
            uint16_t y1,
            uint8_t x2,
            uint16_t y2 )
```

Wyswietla linie w poziomie lub w pionie.

Parametry

color	- kolor lini
x1	- wspolrzedna x pierwszego punktu
y1	- wspolrzedna y pierwszego punktu
x2	- wspolrzedna x drugiego punktu
y2	- wspolrzedna y drugiego punktu

2.15.2.8 GraphicsPutPixel()

```
void GraphicsPutPixel (
            uint16_t color,
```

```
            uint8_t x,
            uint16_t y )
```

Wyswietla punkt o zadanych wspolrzednych.

Parametry

color	- kolor punktu
x	- wspolrzedna x
y	- wspolrzedna y

2.15.2.9 GraphicsSetCursor()

```
void GraphicsSetCursor (
            uint8_t x,
            uint16_t y )
```

Ustawia kursor w punkcie o zadanych wspolrzednych.

Parametry

x	- wspolrzedna x
y	- wspolrzedna y

2.16 Dokumentacja pliku GraphicsPrimitives.h

Definicje

- #define **MIN_ALPHA_SIZE** -180
- #define **MAX_ALPHA_SIZE** 180

Funkcje

- void GraphicsClearScreen (uint16_t color)
- void GraphicsClearArea (uint16_t color, uint8_t x, uint16_t y, uint8_t width, uint16_t height)
- void GraphicsSetCursor (uint8_t x, uint16_t y)
- void GraphicsPutPixel (uint16_t color, uint8_t x, uint16_t y)
- void GraphicsDrawBigDot (uint16_t color, uint8_t x, uint16_t y)
- void GraphicsDrawCircle (uint16_t color, uint8_t x, uint16_t y, uint8_t r)
- ExecuteResult GraphicsDrawLine (uint16_t color, uint8_t x1, uint16_t y1, uint8_t x2, uint16_t y2)
- void GraphicsDrawSimpleLine (uint16_t color, uint8_t x1, uint16_t y1, uint8_t x2, uint16_t y2)
- __attribute__ ((__deprecated__)) void GraphicsDisplayImage(const uint8_t ∗image

Zmienne

- uint8_t **x**
- uint8_t uint16_t **y**
- uint8_t uint16_t uint8_t **width**
- uint8_t uint16_t uint8_t uint16_t **height**

2.16.1 Opis szczegółowy

Autor

Mateusz Chudy

2.16.2 Dokumentacja funkcji

2.16.2.1 GraphicsClearArea()

```
void GraphicsClearArea (
            uint16_t color,
            uint8_t x,
            uint16_t y,
            uint8_t width,
            uint16_t height )
```

Zamalowuje obszar na zadany kolor.

Parametry

color	- kolor zamalowania obszaru
x	- wspolrzedna x lewego gornego konca obszaru
y	- wspolrzedna y lewego gornego konca obszaru
width	- szerokosc obszaru
height	- wysokosc obszaru

2.16.2.2 GraphicsClearScreen()

```
void GraphicsClearScreen (
            uint16_t color )
```

Czysci ekran na zadany kolor.

Parametry

color	- kolor czyszczenia ekranu

2.16.2.3 GraphicsDrawBigDot()

```
void GraphicsDrawBigDot (
            uint16_t color,
            uint8_t x,
            uint16_t y )
```

Wyswietla kropke 3x3 piksele o zadanych wspolrzednych.

Parametry

color	- kolor kropki
x	- wspolrzedna x
y	- wspolrzedna y

2.16.2.4 GraphicsDrawCircle()

```
void GraphicsDrawCircle (
             uint16_t color,
             uint8_t x,
             uint16_t y,
             uint8_t r )
```

Wyswietla okrag o srodku w zadanych wspolrzednych.

Parametry

color	- kolor kropki
x	- wspolrzedna x
y	- wspolrzedna y
r	- promien okregu

2.16.2.5 GraphicsDrawLine()

```
ExecuteResult GraphicsDrawLine (
             uint16_t color,
             uint8_t x1,
             uint16_t y1,
             uint8_t x2,
             uint16_t y2 )
```

Wyswietla linie pomiedzy dwoma punktami.

Parametry

color	- kolor lini
x1	- wspolrzedna x pierwszego punktu
y1	- wspolrzedna y pierwszego punktu
x2	- wspolrzedna x drugiego punktu
y2	- wspolrzedna y drugiego punktu

Zwraca

rezultat operacji

2.16.2.6 GraphicsDrawSimpleLine()

```
void GraphicsDrawSimpleLine (
            uint16_t color,
            uint8_t x1,
            uint16_t y1,
            uint8_t x2,
            uint16_t y2 )
```

Wyswietla linie w poziomie lub w pionie.

Parametry

color	- kolor lini
x1	- wspolrzedna x pierwszego punktu
y1	- wspolrzedna y pierwszego punktu
x2	- wspolrzedna x drugiego punktu
y2	- wspolrzedna y drugiego punktu

2.16.2.7 GraphicsPutPixel()

```
void GraphicsPutPixel (
            uint16_t color,
            uint8_t x,
            uint16_t y )
```

Wyswietla punkt o zadanych wspolrzednych.

Parametry

color	- kolor punktu
x	- wspolrzedna x
y	- wspolrzedna y

2.16.2.8 GraphicsSetCursor()

```
void GraphicsSetCursor (
            uint8_t x,
            uint16_t y )
```

Ustawia kursor w punkcie o zadanych wspolrzednych.

Parametry

x	- wspolrzedna x
y	- wspolrzedna y

2.17 Dokumentacja pliku I2CController.c

Funkcje

- void I2CInit (void)
- void I2CStart (void)
- void I2CWriteByte (int8_t byte)
- int8_t I2CReadByte (void)
- int8_t I2CReadLastByte (void)
- void I2CStop (void)

2.17.1 Opis szczegółowy

Autor

Mateusz Chudy

2.17.2 Dokumentacja funkcji

2.17.2.1 I2CReadByte()

```
int8_t I2CReadByte (
            void  )
```

Odbiera bajt danych.

Zwraca

odebrany bajt danych

2.17.2.2 I2CReadLastByte()

```
int8_t I2CReadLastByte (
            void  )
```

Odbiera ostatni bajt danych.

Zwraca

odebrany ostatni bajt danych

2.17.2.3 I2CWriteByte()

```
void I2CWriteByte (
            int8_t byte )
```

Wysyla bajt danych.

Parametry

byte	- wysylany bajt danych

2.18 Dokumentacja pliku I2CController.h

Funkcje

- void I2CInit (void)
- void I2CStart (void)
- void I2CWriteByte (int8_t byte)
- int8_t I2CReadByte (void)
- int8_t I2CReadLastByte (void)
- void I2CStop (void)

2.18.1 Opis szczegółowy

Autor

Mateusz Chudy

2.18.2 Dokumentacja funkcji

2.18.2.1 I2CReadByte()

```
int8_t I2CReadByte (
            void  )
```

Odbiera bajt danych.

Zwraca

odebrany bajt danych

2.18.2.2 I2CReadLastByte()

```
int8_t I2CReadLastByte (
            void  )
```

Odbiera ostatni bajt danych.

Zwraca

odebrany ostatni bajt danych

2.18.2.3 I2CWriteByte()

```
void I2CWriteByte (
            int8_t byte )
```

Wysyla bajt danych.

Parametry

byte	- wysylany bajt danych

2.19 Dokumentacja pliku LCDController.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- void LCDPWMInit (void)
- void LCDWriteIndex (uint16_t index)
- void LCDWriteReg (uint16_t index, uint16_t data)
- void LCDWriteData (uint16_t data)
- void LCDInit (void)

2.19.1 Opis szczegółowy

Autor

Mateusz Chudy

2.19.2 Dokumentacja funkcji

2.19.2.1 LCDWriteData()

```
void LCDWriteData (
            uint16_t data )
```

Wpisuje dane.

Parametry

data	- wpisywane dane

2.19.2.2 LCDWriteIndex()

```
void LCDWriteIndex (
            uint16_t index )
```

Wpisuje do sterownika indeks rejestru.

Parametry

index	- indeks rejestru

2.19.2.3 LCDWriteReg()

```
void LCDWriteReg (
            uint16_t index,
            uint16_t data )
```

Wpisuje dane do rejestru.

Parametry

index	- indeks rejestru
data	- wpisywane dane

2.20 Dokumentacja pliku LCDController.h

Definicje

- #define **LCD_SIZE_X** 240
- #define **LCD_SIZE_Y** 320
- #define **INDEX_START_OSCILLATION** 0x000
- #define **INDEX_DRIVER_OUTPUT_CONTROL** 0x001
- #define **INDEX_CRYSTAL_DRIVE_INVERSION_CONTROL** 0x002
- #define **INDEX_ENTRY_MODE** 0x003
- #define **INDEX_MOVING_IMAGE_DISPLAY_CONTROL** 0x006
- #define **INDEX_DISPLAY_CONTROL1** 0x007
- #define **INDEX_DISPLAY_CONTROL2** 0x008
- #define **INDEX_DISPLAY_CONTROL3** 0x009
- #define **INDEX_DISPLAY_CONTROL4** 0x00B
- #define **INDEX_EXTERNAL_DISPLAY_INTERFACE_CONTROL1** 0x00C
- #define **INDEX_FRAME_CYCLE_ADJUSTMENT_CONTROL** 0x00D
- #define **INDEX_SOURCE_DRIVER_INTERFACE_CONTROL1** 0x012
- #define **INDEX_GATE_DRIVER_INTERFACE_CONTROL1** 0x013
- #define **INDEX_SOURCE_DRIVER_INTERFACE_CONTROL2** 0x018
- #define **INDEX_GATE_DRIVER_INTERFACE_CONTROL2** 0x019
- #define **INDEX_POWER_CONTROL1** 0x100
- #define **INDEX_POWER_CONTROL2** 0x101
- #define **INDEX_POWER_CONTROL3** 0x102
- #define **INDEX_POWER_CONTROL4** 0x103
- #define **INDEX_POWER_CONTROL5** 0x110
- #define **INDEX_POWER_CONTROL6** 0x111
- #define **INDEX_HORIZONTAL_RAM_ADDRESS_SET** 0x200
- #define **INDEX_VERTICAL_RAM_ADDRESS_SET** 0x201
- #define **INDEX_RAM_DATA_WRITE** 0x202
- #define **INDEX_RAM_WRITE_DATA_MASK1** 0x203

- #define **INDEX_RAM_WRITE_DATA_MASK2** 0x204
- #define **INDEX_HORIZONTAL_RAM_ADDRESS_START_POSITION** 0x210
- #define **INDEX_HORIZONTAL_RAM_ADDRESS_END_POSITION** 0x211
- #define **INDEX_VERTICAL_RAM_ADDRESS_START_POSITION** 0x212
- #define **INDEX_VERTICAL_RAM_ADDRESS_END_POSITION** 0x213
- #define **INDEX_MOVING_PICTURE_CONTROL_ADDRESS_START_POSITION1** 0x214
- #define **INDEX_MOVING_PICTURE_CONTROL_ADDRESS_END_POSITION1** 0x215
- #define **INDEX_MOVING_PICTURE_CONTROL_ADDRESS_START_POSITION2** 0x216
- #define **INDEX_MOVING_PICTURE_CONTROL_ADDRESS_END_POSITION2** 0x217
- #define **INDEX_GAMMA_CONTROL1** 0x300
- #define **INDEX_GAMMA_CONTROL2** 0x301
- #define **INDEX_GAMMA_CONTROL3** 0x302
- #define **INDEX_GAMMA_CONTROL4** 0x303
- #define **INDEX_GAMMA_CONTROL5** 0x304
- #define **INDEX_GAMMA_CONTROL6** 0x305
- #define **INDEX_GAMMA_CONTROL7** 0x306
- #define **INDEX_GAMMA_CONTROL8** 0x307
- #define **INDEX_GAMMA_CONTROL9** 0x308
- #define **INDEX_GAMMA_CONTROL10** 0x309
- #define **INDEX_LINE_NUMBER_CONTROL** 0x400
- #define **INDEX_SCREEN_CONTROL** 0x401
- #define **INDEX_FULL_SCREEN_IMAGE_RAM_AREA_START_LINE** 0x402
- #define **INDEX_FULL_SCREEN_IMAGE_RAM_AREA_END_LINE** 0x403
- #define **INDEX_VERTICAL_SCROLL_CONTROL** 0x404
- #define **LCD_RST_H**() PORTB |= (1 << PB0);
- #define **LCD_RST_L**() PORTB &= ~(1 << PB0);
- #define **LCD_RS_H**() PORTB |= (1 << PB1);
- #define **LCD_RS_L**() PORTB &= ~(1 << PB1);
- #define **LCD_CS_H**() PORTB |= (1 << PB2);
- #define **LCD_CS_L**() PORTB &= ~(1 << PB2);

Wyliczenia

- enum Color {
  BLACK = 0X0000, **CYAN** = 0x07FF, **PURPLE** = 0xF81F, **RED** = 0XF800,
  GREEN = 0X07E0, **BLUE** = 0X001F, **YELLOW** = 0XFFE0, **ORANGE** = 0XFC08,
  GRAY = 0X8430, **LGRAY** = 0XC618, **DARKGRAY** = 0X8410, **PORPO** = 0X801F,
  PINK = 0XF81F, **GRAYBLUE** = 0X5458, **LGRAYBLUE** = 0XA651, **DARKBLUE** = 0X01CF,
  LIGHTBLUE = 0X7D7C, **WHITE** = 0XFFFF }

Funkcje

- void LCDPWMInit (void)
- void LCDWriteIndex (uint16_t index)
- void LCDWriteReg (uint16_t index, uint16_t data)
- void LCDWriteData (uint16_t data)
- void LCDInit (void)

2.20.1 Opis szczegółowy

Autor

Mateusz Chudy

2.20.2 Dokumentacja funkcji

2.20.2.1 LCDWriteData()

```
void LCDWriteData (
            uint16_t data )
```

Wpisuje dane.

Parametry

data	- wpisywane dane

2.20.2.2 LCDWriteIndex()

```
void LCDWriteIndex (
            uint16_t index )
```

Wpisuje do sterownika indeks rejestru.

Parametry

index	- indeks rejestru

2.20.2.3 LCDWriteReg()

```
void LCDWriteReg (
            uint16_t index,
            uint16_t data )
```

Wpisuje dane do rejestru.

Parametry

index	- indeks rejestru
data	- wpisywane dane

2.21 Dokumentacja pliku Main.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- int main (void)

2.21.1 Opis szczegółowy

Autor

Mateusz Chudy

2.21.2 Dokumentacja funkcji

2.21.2.1 main()

```
int main (
            void  )
```

Glowna funkcja programu.

Zwraca

0

2.22 Dokumentacja pliku MainMenu.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- void MainMenuShow (void)

2.22.1 Opis szczegółowy

Autor

Mateusz Chudy

2.23 Dokumentacja pliku MainMenu.h

Definicje

- #define **MAIN_GAMES_BUTTON_X1** 0
- #define **MAIN_GAMES_BUTTON_Y1** 0
- #define **MAIN_GAMES_BUTTON_X2** 240
- #define **MAIN_GAMES_BUTTON_Y2** 40
- #define **MAIN_GAMES_BUTTON_COLOR** DARKBLUE
- #define **MAIN_GAMES_TEXT_COLOR** WHITE
- #define **MAIN_APP_BUTTON_X1** 0
- #define **MAIN_APP_BUTTON_Y1** 40
- #define **MAIN_APP_BUTTON_X2** 240
- #define **MAIN_APP_BUTTON_Y2** 80
- #define **MAIN_APP_BUTTON_COLOR** 0X01CA
- #define **MAIN_APP_TEXT_COLOR** WHITE

Funkcje

- void MainMenuShow (void)

2.23.1 Opis szczegółowy

Autor

Mateusz Chudy

2.24 Dokumentacja pliku MenuOptions.h

Wyliczenia

- enum Option {
  DESKTOP = 0, **MAIN**, **GAMES**, **APPLICATIONS**,
  PING_PONG, **SNAKE**, **PAINT**, **BACK**,
  YES, **NO**, **LAST** }

2.24.1 Opis szczegółowy

Autor

Mateusz Chudy

2.25 Dokumentacja pliku Paint.c

Funkcje

- void PaintInit (Color *buttonColors, struct Button *button)
- Color PaintCheckColorChange (Color *buttonColors, Color currentColor)
- void PaintShow (void)

2.25.1 Opis szczegółowy

Autor

Mateusz Chudy

2.25.2 Dokumentacja funkcji

2.25.2.1 PaintCheckColorChange()

```
Color PaintCheckColorChange (
            Color * buttonColors,
            Color currentColor )
```

Sprawdza zmiane koloru przez urzytkownika.

Parametry

buttonColors	- tablica dostepnych kolorow rysowania
currentColor	- obecny kolor rysowania

Zwraca

obecnie wybrany kolor

2.25.2.2 PaintInit()

```
void PaintInit (
            Color * buttonColors,
            struct Button * button )
```

Inicjalizuje aplikacje Paint.

Parametry

buttonColors	- tablica dostepnych kolorow rysowania
button	- wskaźnik na strukture reprezentujaca przycisk exit

2.26 Dokumentacja pliku Paint.h

Definicje

- #define **PAINT_BACKGROUND_COLOR** WHITE
- #define **PAINT_BUTTONS_SIZE** 30

- #define **PAINT_BUTTONS_SIZE** 30
- #define **PAINT_BUTTONS_MAX_X** 8
- #define **PAINT_EXIT_BUTTON_X** 5
- #define **PAINT_EXIT_BUTTON_Y** 285

Funkcje

- void PaintInit (Color ∗buttonColors, struct Button ∗button)
- Color PaintCheckColorChange (Color ∗buttonColors, Color currentColor)
- void PaintShow (void)

2.26.1 Opis szczegółowy

Autor

Mateusz Chudy

2.26.2 Dokumentacja funkcji

2.26.2.1 PaintCheckColorChange()

```
Color PaintCheckColorChange (
            Color * buttonColors,
            Color currentColor )
```

Sprawdza zmiane koloru przez urzytkownika.

Parametry

buttonColors	- tablica dostepnych kolorow rysowania
currentColor	- obecny kolor rysowania

Zwraca

obecnie wybrany kolor

2.26.2.2 PaintInit()

```
void PaintInit (
            Color * buttonColors,
            struct Button * button )
```

Inicjalizuje aplikacje Paint.

Parametry

buttonColors	- tablica dostepnych kolorow rysowania
button	- wskaźnik na strukture reprezentujaca przycisk exit

2.27 Dokumentacja pliku PingPong.c

Funkcje

- bool CalculateBallPosition (uint8_t ∗newX, uint16_t ∗newY, uint8_t lineX, Vector ∗moveVector)
- void CalculateLinePosition (uint8_t ∗newX)
- void PingPongPlay (void)

2.27.1 Opis szczegółowy

Autor

Mateusz Chudy

2.27.2 Dokumentacja funkcji

2.27.2.1 CalculateBallPosition()

```
bool CalculateBallPosition (
            uint8_t * newX,
            uint16_t * newY,
            uint8_t lineX,
            Vector * moveVector )
```

Oblicza nowe wspolrzedne pilki.

Parametry

newX	- nowa wartosc wspolrzednej x
newX	- nowa wartosc wspolrzednej y
lineX	- wspolrzedna x lini
moveVector	- wskaznik na wektor przesuniecia pilki

2.27.2.2 CalculateLinePosition()

```
void CalculateLinePosition (
            uint8_t * newX )
```

Oblicza nowe wspolrzedne lini.

Parametry

newX	- nowa wartosc wspolrzednej x lini

2.28 Dokumentacja pliku PingPong.h

Struktury danych

- struct Vector

Definicje

- #define **PONG_ACCE_SENSITIVITY** 10
- #define **PONG_DISP_SENSITIVITY** 1
- #define **PONG_Y_POS** 300
- #define **PONG_LINE_SIZE** 60
- #define **PONG_BALL_SPEED** 20
- #define **PONG_LINE_SPEED** 2
- #define **PONG_MIN_X** 0
- #define **PONG_MAX_X** 240
- #define **PONG_MIN_Y** 0
- #define **PONG_MAX_Y** 320

Funkcje

- bool CalculateBallPosition (uint8_t ∗newX, uint16_t ∗newY, uint8_t lineX, Vector ∗moveVector)
- void CalculateLinePosition (uint8_t ∗newX)
- void PingPongPlay (void)

2.28.1 Opis szczegółowy

Autor

Mateusz Chudy

2.28.2 Dokumentacja funkcji

2.28.2.1 CalculateBallPosition()

```
bool CalculateBallPosition (
            uint8_t * newX,
            uint16_t * newY,
            uint8_t lineX,
            Vector * moveVector )
```

Oblicza nowe wspolrzedne pilki.

Parametry

newX	- nowa wartosc wspolrzednej x
newX	- nowa wartosc wspolrzednej y
lineX	- wspolrzedna x lini
moveVector	- wskaznik na wektor przesuniecia pilki

2.28.2.2 CalculateLinePosition()

```
void CalculateLinePosition (
            uint8_t * newX )
```

Oblicza nowe wspolrzedne lini.

Parametry

newX	- nowa wartosc wspolrzednej x lini

2.29 Dokumentacja pliku Question.c

Funkcje

- void QuestionDisplay (char ∗string, Option yes, Option no)

2.29.1 Opis szczegółowy

Autor

Mateusz Chudy

2.29.2 Dokumentacja funkcji

2.29.2.1 QuestionDisplay()

```
void QuestionDisplay (
            char * string,
            Option yes,
            Option no )
```

Wyswietla pytanie i przyciski odpowiedzi.

Parametry

string	- napis z pytaniem
yes	- opcja do ktorej przechodzi gdy odpowiedz jest pozytywna
no	- opcja do ktorej przechodzi gdy odpowiedz jest negatywna

2.30 Dokumentacja pliku Question.h

Definicje

- #define **QUESTION_BACKGROUND_COLOR** GRAY
- #define **QUESTION_TEXT_COLOR** BLACK
- #define **QUESTION_Y_POS** 80
- #define **ANSWER_Y_POS** 160
- #define **YES_X_POS** 30
- #define **NO_X_POS** 150
- #define **YES_COLOR** GREEN
- #define **NO_COLOR** RED
- #define **QUEST_BUTTON_WIDTH** 60
- #define **QUEST_BUTTON_HEIGHT** 40

Funkcje

- void QuestionDisplay (char ∗string, Option yes, Option no)

2.30.1 Opis szczegółowy

Autor

Mateusz Chudy

2.30.2 Dokumentacja funkcji

2.30.2.1 QuestionDisplay()

```
void QuestionDisplay (
            char * string,
            Option yes,
            Option no )
```

Wyswietla pytanie i przyciski odpowiedzi.

Parametry

string	- napis z pytaniem
yes	- opcja do ktorej przechodzi gdy odpowiedz jest pozytywna
no	- opcja do ktorej przechodzi gdy odpowiedz jest negatywna

2.31 Dokumentacja pliku Snake.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- Element ∗∗ SnakeInit (Element ∗∗snakeElements, uint8_t ∗snakeSize, Element ∗food)
- void SnakePlay (void)

2.31.1 Opis szczegółowy

Autor

Mateusz Chudy

2.31.2 Dokumentacja funkcji

2.31.2.1 SnakeInit()

```
Element** SnakeInit (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Inicjalizuje gre Snake.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.32 Dokumentacja pliku Snake.h

Definicje

- #define **SNAKE_START_SIZE** 3
- #define **SNAKE_START_X** 6
- #define **SNAKE_START_Y** 0
- #define **SNAKE_MAX_X** 11
- #define **SNAKE_MAX_Y** 15
- #define **SNAKE_START_DERECTION** DOWN
- #define **SNAKE_BACKGROUND_COLOR** 0x1002
- #define **SNAKE_COLOR_FILTER** 0x4444
- #define **SNAKE_DELAY** 200

Funkcje

- Element ** SnakeInit (Element **snakeElements, uint8_t *snakeSize, Element *food)
- void SnakePlay (void)

2.32.1 Opis szczegółowy

Autor

Mateusz Chudy

2.32.2 Dokumentacja funkcji

2.32.2.1 SnakeInit()

```
Element** SnakeInit (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Inicjalizuje gre Snake.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.33 Dokumentacja pliku SnakeElements.c

Funkcje

- void SnakeDiplayElement (Element *element, bool clear)
- void SnakeNewFood (Element **snakeElements, uint8_t snakeSize, Element *food)
- Element ** SnakeElementAdd (Element **snakeElements, uint8_t *snakeSize, Color color, uint8_t x, uint8_t y, Direction direction)
- Element ** SnakeDestroy (Element **snakeElements, uint8_t *snakeSize)

2.33.1 Opis szczegółowy

Autor

Mateusz Chudy

2.33.2 Dokumentacja funkcji

2.33.2.1 SnakeDestroy()

```
Element** SnakeDestroy (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Niszczy tablice elementow wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.33.2.2 SnakeDiplayElement()

```
void SnakeDiplayElement (
            Element * element,
            bool clear )
```

Wyswietla element.

Parametry

element	- wskaznik na wyswietlany element
clear	- flaga oznaczajaca czy zamazujemy

2.33.2.3 SnakeElementAdd()

```
Element** SnakeElementAdd (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Color color,
            uint8_t x,
            uint8_t y,
            Direction direction )
```

Dodaje element do wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
color	- kolor elementu
x	- wspolrzedna x elementu
y	- wspolrzedna y elementu
direction	- kierunek ruchu elementu

Zwraca

wskaznik na tablice elementow weza

2.33.2.4 SnakeNewFood()

```
void SnakeNewFood (
            Element ** snakeElements,
            uint8_t snakeSize,
            Element * food )
```

Tworzy nowy pokarm i rysuje go.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- rozmiar weza
food	- wskaznik na element pokarmu

2.34 Dokumentacja pliku SnakeElements.h

Struktury danych

- struct Element

Definicje

- #define **ELEMENT_SIZE** 20
- #define **CHANGE_SIZE_FREQUENCY** 100

Funkcje

- void SnakeDiplayElement (Element ∗element, bool clear)
- void SnakeNewFood (Element ∗∗snakeElements, uint8_t snakeSize, Element ∗food)
- Element ∗∗ SnakeElementAdd (Element ∗∗snakeElements, uint8_t ∗snakeSize, Color color, uint8_t x, uint8↩ _t y, Direction direction)
- Element ∗∗ SnakeDestroy (Element ∗∗snakeElements, uint8_t ∗snakeSize)

2.34.1 Opis szczegółowy

Autor

Mateusz Chudy

2.34.2 Dokumentacja funkcji

2.34.2.1 SnakeDestroy()

```
Element** SnakeDestroy (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Niszczy tablice elementow wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.34.2.2 SnakeDiplayElement()

```
void SnakeDiplayElement (
            Element * element,
            bool clear )
```

Wyswietla element.

Parametry

element	- wskaznik na wyswietlany element
clear	- flaga oznaczajaca czy zamazujemy

2.34.2.3 SnakeElementAdd()

```
Element** SnakeElementAdd (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Color color,
```

```
            uint8_t x,
            uint8_t y,
            Direction direction )
```

Dodaje element do wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
color	- kolor elementu
x	- wspolrzedna x elementu
y	- wspolrzedna y elementu
direction	- kierunek ruchu elementu

Zwraca

wskaznik na tablice elementow weza

2.34.2.4 SnakeNewFood()

```
void SnakeNewFood (
            Element ** snakeElements,
            uint8_t snakeSize,
            Element * food )
```

Tworzy nowy pokarm i rysuje go.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- rozmiar weza
food	- wskaznik na element pokarmu

2.35 Dokumentacja pliku SnakeMove.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- uint8_t SnakeCheckCollision (Element ∗∗snakeElements, uint8_t snakeSize, uint8_t isFall)
- Element ∗∗ SnakeCheckChangeDirection (Element ∗∗snakeElements, uint8_t ∗snakeSize)
- Element ∗∗ SnakeCheckFood (Element ∗∗snakeElements, uint8_t ∗snakeSize, Element ∗food)
- Element ∗∗ SnakeUpdateDirections (Element ∗∗snakeElements, uint8_t ∗snakeSize)
- Element ∗∗ SnakeMove (Element ∗∗snakeElements, uint8_t ∗snakeSize, Element ∗food)

2.35.1 Opis szczegółowy

Autor

Mateusz Chudy

2.35.2 Dokumentacja funkcji

2.35.2.1 SnakeCheckChangeDirection()

```
Element** SnakeCheckChangeDirection (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Sprawdza i zmienia kierunek wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.35.2.2 SnakeCheckCollision()

```
uint8_t SnakeCheckCollision (
            Element ** snakeElements,
            uint8_t snakeSize,
            uint8_t isFall )
```

Sprawdza kolizje weza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- rozmiar weza
isFall	- flaga wskazujaca czy w grze obecne sa sciany

Zwraca

wartosc logiczna wskazujaca na kolizje lub jej brak

2.35.2.3 SnakeCheckFood()

```
Element** SnakeCheckFood (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Sprawdza czy waz natrafil na pokarm, jesli takrysuje nowy i zwieksza wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.35.2.4 SnakeMove()

```
Element** SnakeMove (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Przemieszcza wenza na ekranie.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.35.2.5 SnakeUpdateDirections()

```
Element** SnakeUpdateDirections (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Aktualizuje kierunki dla elementow wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.36 Dokumentacja pliku SnakeMove.h

Funkcje

- uint8_t SnakeCheckCollision (Element **snakeElements, uint8_t snakeSize, uint8_t isFall)
- Element ** SnakeCheckChangeDirection (Element **snakeElements, uint8_t *snakeSize)
- Element ** SnakeCheckFood (Element **snakeElements, uint8_t *snakeSize, Element *food)
- Element ** SnakeUpdateDirections (Element **snakeElements, uint8_t *snakeSize)
- Element ** SnakeMove (Element **snakeElements, uint8_t *snakeSize, Element *food)

2.36.1 Opis szczegółowy

Autor

Mateusz Chudy

2.36.2 Dokumentacja funkcji

2.36.2.1 SnakeCheckChangeDirection()

```
Element** SnakeCheckChangeDirection (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Sprawdza i zmienia kierunek wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.36.2.2 SnakeCheckCollision()

```
uint8_t SnakeCheckCollision (
            Element ** snakeElements,
            uint8_t snakeSize,
            uint8_t isFall )
```

Sprawdza kolizje weza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- rozmiar weza
isFall	- flaga wskazujaca czy w grze obecne sa sciany

Zwraca

wartosc logiczna wskazujaca na kolizje lub jej brak

2.36.2.3 SnakeCheckFood()

```
Element** SnakeCheckFood (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Sprawdza czy waz natrafil na pokarm, jesli takrysuje nowy i zwieksza wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.36.2.4 SnakeMove()

```
Element** SnakeMove (
            Element ** snakeElements,
            uint8_t * snakeSize,
            Element * food )
```

Przemieszcza wenza na ekranie.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza
food	- wskaznik na element pokarmu

Zwraca

wskaznik na tablice elementow weza

2.36.2.5 SnakeUpdateDirections()

```
Element** SnakeUpdateDirections (
            Element ** snakeElements,
            uint8_t * snakeSize )
```

Aktualizuje kierunki dla elementow wenza.

Parametry

snakeElements	- wskaznik na tablice elementow weza
snakeSize	- wskaznik na rozmiar weza

Zwraca

wskaznik na tablice elementow weza

2.37 Dokumentacja pliku SPIController.c

Funkcje

- void SPIInit (void)
- uint8_t SPICommunication (uint8_t data)

2.37.1 Opis szczegółowy

Autor

Mateusz Chudy

2.37.2 Dokumentacja funkcji

2.37.2.1 SPICommunication()

```
uint8_t SPICommunication (
            uint8_t data )
```

Wysyla / odbiera bajt danych z magistrali SPI.

Parametry

data	- wysylany bajt danych

Zwraca

odebrany bajt danych

2.38 Dokumentacja pliku SPIController.h

Funkcje

- void SPIInit (void)
- uint8_t SPICommunication (uint8_t data)

2.38.1 Opis szczegółowy

Autor

Mateusz Chudy

2.38.2 Dokumentacja funkcji

2.38.2.1 SPICommunication()

```
uint8_t SPICommunication (
            uint8_t data )
```

Wysyla / odbiera bajt danych z magistrali SPI.

Parametry

data	- wysylany bajt danych

Zwraca

odebrany bajt danych

2.39 Dokumentacja pliku ToolSet.c

Funkcje

- uint16_t ToolsRandom (uint16_t min, uint16_t max)
- void ToolsSwap (uint16_t *a, uint16_t *b)
- uint32_t ToolsPow (uint16_t m, uint16_t n)

2.39.1 Opis szczegółowy

Autor

Mateusz Chudy

2.39.2 Dokumentacja funkcji

2.39.2.1 ToolsPow()

```
uint32_t ToolsPow (
            uint16_t m,
            uint16_t n )
```

Realizuje potegowanie.

Parametry

m	- liczba
n	- potega

Zwraca

liczba m do potegi n

2.39.2.2 ToolsRandom()

```
uint16_t ToolsRandom (
            uint16_t min,
            uint16_t max )
```

Generuje liczbe losowa z zadanego przedzialu.

Parametry

min	- ograniczenie dolne przedzialu
max	- ograniczenie gorne przedzialu

Zwraca

wygenerowana liczba losowa

2.39.2.3 ToolsSwap()

```
void ToolsSwap (
            uint16_t * a,
            uint16_t * b )
```

Zamienia dwie zmienne wartosciami.

Parametry

a	- wskaznik na pierwsza zmienna
b	- wskaznik na druga zmienna

2.40 Dokumentacja pliku ToolSet.h

Funkcje

- uint16_t ToolsRandom (uint16_t min, uint16_t max)
- void ToolsSwap (uint16_t ∗a, uint16_t ∗b)
- uint32_t ToolsPow (uint16_t m, uint16_t n)

2.40.1 Opis szczegółowy

Autor

Mateusz Chudy

2.40.2 Dokumentacja funkcji

2.40.2.1 ToolsPow()

```
uint32_t ToolsPow (
            uint16_t m,
            uint16_t n )
```

Realizuje potegowanie.

Parametry

m	- liczba
n	- potega

Zwraca

liczba m do potegi n

2.40.2.2 ToolsRandom()

```
uint16_t ToolsRandom (
            uint16_t min,
            uint16_t max )
```

Generuje liczbe losowa z zadanego przedzialu.

Parametry

min	- ograniczenie dolne przedzialu
max	- ograniczenie gorne przedzialu

Zwraca

wygenerowana liczba losowa

2.40.2.3 ToolsSwap()

```
void ToolsSwap (
            uint16_t * a,
            uint16_t * b )
```

Zamienia dwie zmienne wartosciami.

Parametry

a	- wskaznik na pierwsza zmienna
b	- wskaznik na druga zmienna

2.41 Dokumentacja pliku TouchController.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- ISR (INT0_vect)
- void TouchInit (void)
- uint16_t TouchGetADC (uint8_t cmdCode)
- uint16_t TouchGetADCEx (uint8_t cmdCode)
- ExecuteResult TouchReadADS (uint16_t ∗xad, uint16_t ∗yad)
- ExecuteResult TouchReadADS2 (uint32_t ∗xad, uint32_t ∗yad)
- void TouchConvertAdToXy (void)
- ExecuteResult TouchReadOnce (void)
- ExecuteResult TouchReadContinue (void)

2.41.1 Opis szczegółowy

Autor

Mateusz Chudy

2.41.2 Dokumentacja funkcji

2.41.2.1 TouchGetADC()

```
uint16_t TouchGetADC (
            uint8_t cmdCode )
```

Pobiera ze sterownika pojedynczy wynik ADC.

Parametry

cmdCode	- komenda wskazujaca na os

Zwraca

wynik konwersji AD

2.41.2.2 TouchGetADCEx()

```
uint16_t TouchGetADCEx (
            uint8_t cmdCode )
```

Pobiera ze sterownika kilka razy wynik ADC i filtruje zwracana wartosc.

Parametry

cmdCode	- komenda wskazujaca na os

Zwraca

przefiltrowany wynik konwersji AD

2.41.2.3 TouchReadADS()

```
ExecuteResult TouchReadADS (
            uint16_t * xad,
            uint16_t * yad )
```

Pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y.

Parametry

xad	- przefiltrowany wynik konwersji AD dla osi x
yad	- przefiltrowany wynik konwersji AD dla osi y

Zwraca

rezultat operacji

2.41.2.4 TouchReadADS2()

```
ExecuteResult TouchReadADS2 (
            uint32_t * xad,
            uint32_t * yad )
```

Dwukrotnie pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y.

Wartosc obu odczytow dla porownania, w granicach dopuszczalnego bledu

Parametry

xad	- przefiltrowany wynik konwersji AD dla osi x
yad	- przefiltrowany wynik konwersji AD dla osi y

Zwraca

rezultat operacji

2.41.2.5 TouchReadContinue()

```
ExecuteResult TouchReadContinue (
            void  )
```

Pobiera wspolrzedne dotkniecia w sposob ciagly.

Zwraca

rezultat operacji

2.41.2.6 TouchReadOnce()

```
ExecuteResult TouchReadOnce (
            void  )
```

Pobiera jednokrotnie wspolrzedne dotkniecia.

Zwraca

rezultat operacji

2.42 Dokumentacja pliku TouchController.h

Struktury danych

- struct TouchPoint

Definicje

- #define **TOUCH_CS_H**() PORTD |= (1 << PD4);
- #define **TOUCH_CS_L**() PORTD &= ~(1 << PD4);
- #define **CMD_READ_X** 0xD0
- #define **CMD_READ_Y** 0x90
- #define **ERR_RANGE** 50
- #define **READ_TIMES** 10
- #define **LOST_VAL** 4
- #define **CORRECT_X** -20
- #define **CORRECT_Y** -2

Funkcje

- ISR (INT0_vect)
- void TouchInit (void)
- uint16_t TouchGetADC (uint8_t cmdCode)
- uint16_t TouchGetADCEx (uint8_t cmdCode)
- ExecuteResult TouchReadADS (uint16_t *xad, uint16_t *yad)
- ExecuteResult TouchReadADS2 (uint32_t *xad, uint32_t *yad)
- void TouchConvertAdToXy (void)
- ExecuteResult TouchReadOnce (void)
- ExecuteResult TouchReadContinue (void)

Zmienne

- TouchPoint touchPoint
- volatile uint8_t interruptFlag

2.42.1 Opis szczegółowy

Autor

Mateusz Chudy

2.42.2 Dokumentacja funkcji

2.42.2.1 TouchGetADC()

```
uint16_t TouchGetADC (
            uint8_t cmdCode )
```

Pobiera ze sterownika pojedynczy wynik ADC.

Parametry

cmdCode	- komenda wskazujaca na os

Zwraca

wynik konwersji AD

2.42.2.2 TouchGetADCEx()

```
uint16_t TouchGetADCEx (
            uint8_t cmdCode )
```

Pobiera ze sterownika kilka razy wynik ADC i filtruje zwracana wartosc.

Parametry

cmdCode	- komenda wskazujaca na os

Zwraca

przefiltrowany wynik konwersji AD

2.42.2.3 TouchReadADS()

```
ExecuteResult TouchReadADS (
            uint16_t * xad,
            uint16_t * yad )
```

Pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y.

Parametry

xad	- przefiltrowany wynik konwersji AD dla osi x
yad	- przefiltrowany wynik konwersji AD dla osi y

Zwraca

rezultat operacji

2.42.2.4 TouchReadADS2()

```
ExecuteResult TouchReadADS2 (
            uint32_t * xad,
            uint32_t * yad )
```

Dwukrotnie pobiera ze sterownika przefiltrowane wyniki konwersji AD dla osi x i y.

Wartosc obu odczytow dla porownania, w granicach dopuszczalnego bledu

Parametry

xad	- przefiltrowany wynik konwersji AD dla osi x
yad	- przefiltrowany wynik konwersji AD dla osi y

Zwraca

rezultat operacji

2.42.2.5 TouchReadContinue()

```
ExecuteResult TouchReadContinue (
            void  )
```

Pobiera wspolrzedne dotkniecia w sposob ciagly.

Zwraca

rezultat operacji

2.42.2.6 TouchReadOnce()

```
ExecuteResult TouchReadOnce (
            void  )
```

Pobiera jednokrotnie wspolrzedne dotkniecia.

Zwraca

rezultat operacji

2.43 Dokumentacja pliku Welcome.c

Definicje

- #define **F_CPU** 16000000UL

Funkcje

- void WelcomeShow (void)

2.43.1 Opis szczegółowy

Autor

Mateusz Chudy

2.44 Dokumentacja pliku Welcome.h

Definicje

- #define **WELCOME_X_POS** 88
- #define **WELCOME_Y_POS** 140
- #define **WELCOME_STRING_COLOR** WHITE
- #define **WELCOME_TIME** 2000

Funkcje

- void WelcomeShow (void)

2.44.1 Opis szczegółowy

Autor

Mateusz Chudy

Skorowidz