

Security System (Intruder Alarm)

Generated by Doxygen 1.8.14

Contents

1	Security System (Intruder Alarm)	1
2	Module Index	3
2.1	Modules	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	GPIO_Controller	9
5.1.1	Detailed Description	9
5.2	GPIO_Controller_Type_Definitions	10
5.2.1	Detailed Description	10
5.2.2	Typedef Documentation	10
5.2.2.1	GPIO_PIN	10
5.3	GPIO_Controller_public_functions	11
5.3.1	Detailed Description	11
5.3.2	Function Documentation	11
5.3.2.1	delayMicro()	11
5.3.2.2	digitalRead()	12
5.3.2.3	digitalWrite()	12
5.3.2.4	enableClocksGPIO()	12

5.3.2.5	<code>getPin()</code>	13
5.3.2.6	<code>initGPIO()</code>	13
5.4	<code>LedBuzzerButton_Controller</code>	15
5.4.1	Detailed Description	15
5.5	<code>LedBuzzerButton_Controller_public_functions</code>	16
5.5.1	Detailed Description	16
5.5.2	Function Documentation	16
5.5.2.1	<code>initButton()</code>	16
5.5.2.2	<code>initBuzzer()</code>	17
5.5.2.3	<code>initLed()</code>	17
5.5.2.4	<code>startAlarm()</code>	17
5.5.2.5	<code>startLights()</code>	18
5.5.2.6	<code>stopAlarm()</code>	18
5.6	<code>Motor_Controller</code>	19
5.6.1	Detailed Description	19
5.7	<code>Motor_Controller_public_functions</code>	20
5.7.1	Detailed Description	20
5.7.2	Function Documentation	20
5.7.2.1	<code>initMotor()</code>	20
5.7.2.2	<code>moveDoor()</code>	20
5.8	<code>PIR_Controller</code>	22
5.8.1	Detailed Description	22
5.9	<code>PIR_Controller_public_functions</code>	23
5.9.1	Detailed Description	23
5.9.2	Function Documentation	23
5.9.2.1	<code>initPir()</code>	23
5.10	<code>TouchInput_Controller</code>	24
5.10.1	Detailed Description	24
5.11	<code>TouchInput_Controller_public_functions</code>	25
5.11.1	Detailed Description	25

5.11.2	Function Documentation	25
5.11.2.1	addKey()	25
5.11.2.2	drawKeypad()	25
5.11.2.3	getTouch()	26
5.11.2.4	initGLCD()	26
5.12	EXTI_IRQ_handler	27
5.12.1	Detailed Description	27
5.12.2	Function Documentation	27
5.12.2.1	EXTI0_IRQHandler()	27
5.12.2.2	EXTI15_10_IRQHandler()	27
5.12.2.3	EXTI4_IRQHandler()	28
5.12.2.4	HAL_GPIO_EXTI_Callback()	28
5.13	MAIN	29
5.13.1	Detailed Description	29
5.14	TouchInput_Controller_private_variables	30
5.14.1	Detailed Description	30
5.15	TouchInput_Controller_private_functions	31
5.15.1	Detailed Description	31
5.15.2	Function Documentation	31
5.15.2.1	addKey()	31
5.15.2.2	clearInput()	32
5.15.2.3	drawKeypad()	32
5.15.2.4	getPressedKey()	32
5.15.2.5	getTouch()	33
5.15.2.6	initGLCD()	33
5.15.2.7	rightPass()	34
5.15.2.8	wrongPass()	34
5.16	CMSIS	35
5.16.1	Detailed Description	35
5.17	Stm32f7xx_system	36

5.17.1 Detailed Description	36
5.18 STM32F7xx_System_Private_Includes	37
5.18.1 Detailed Description	37
5.18.2 Macro Definition Documentation	37
5.18.2.1 HSE_VALUE	37
5.18.2.2 HSI_VALUE	37
5.19 STM32F7xx_System_Private_TypesDefinitions	38
5.20 STM32F7xx_System_Private_Defines	39
5.20.1 Detailed Description	39
5.20.2 Macro Definition Documentation	39
5.20.2.1 VECT_TAB_OFFSET	39
5.21 STM32F7xx_System_Private_Macros	40
5.22 STM32F7xx_System_Private_Variables	41
5.22.1 Detailed Description	41
5.23 STM32F7xx_System_Private_FunctionPrototypes	42
5.24 STM32F7xx_System_Private_Functions	43
5.24.1 Detailed Description	43
5.24.2 Function Documentation	43
5.24.2.1 SystemCoreClockUpdate()	43
5.24.2.2 SystemInit()	44
6 Data Structure Documentation	45
6.1 _GPIO_PIN Struct Reference	45
6.1.1 Detailed Description	45

7 File Documentation	47
7.1 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/defines.h File Reference	47
7.1.1 Detailed Description	47
7.2 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/gpioController.h File Reference	48
7.2.1 Detailed Description	48
7.3 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/LedBuzzerController.h File Reference	49
7.3.1 Detailed Description	49
7.4 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/motorController.h File Reference	49
7.4.1 Detailed Description	50
7.5 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/pirController.h File Reference	50
7.5.1 Detailed Description	50
7.6 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/include/touchInputController.h File Reference	51
7.6.1 Detailed Description	51
7.7 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↔ Security_System_CW/RTE/Device/STM32F746NGHx/stm32f7xx_hal_conf.h File Reference	52
7.7.1 Detailed Description	53
7.7.2 Macro Definition Documentation	54
7.7.2.1 EXTERNAL_CLOCK_VALUE	54
7.7.2.2 HSE_STARTUP_TIMEOUT	54
7.7.2.3 HSE_VALUE	55
7.7.2.4 HSI_VALUE	55
7.7.2.5 LSE_STARTUP_TIMEOUT	55
7.7.2.6 LSE_VALUE	55
7.7.2.7 LSI_VALUE	55
7.7.2.8 PHY_AUTONEGO_COMPLETE	55
7.7.2.9 PHY_AUTONEGOTIATION	56
7.7.2.10 PHY_BCR	56

7.7.2.11	PHY_BSR	56
7.7.2.12	PHY_DUPLEX_STATUS	56
7.7.2.13	PHY_FULLDUPLEX_100M	56
7.7.2.14	PHY_FULLDUPLEX_10M	56
7.7.2.15	PHY_HALFDUPLEX_100M	56
7.7.2.16	PHY_HALFDUPLEX_10M	56
7.7.2.17	PHY_ISOLATE	57
7.7.2.18	PHY_JABBER_DETECTION	57
7.7.2.19	PHY_LINK_INTERRUPT	57
7.7.2.20	PHY_LINK_STATUS	57
7.7.2.21	PHY_LINKED_STATUS	57
7.7.2.22	PHY_LOOPBACK	57
7.7.2.23	PHY_MICR	57
7.7.2.24	PHY_MICR_INT_EN	57
7.7.2.25	PHY_MICR_INT_OE	58
7.7.2.26	PHY_MISR	58
7.7.2.27	PHY_MISR_LINK_INT_EN	58
7.7.2.28	PHY_POWERDOWN	58
7.7.2.29	PHY_RESET	58
7.7.2.30	PHY_RESTART_AUTONEGOTIATION	58
7.7.2.31	PHY_SPEED_STATUS	58
7.7.2.32	PHY_SR	58
7.7.2.33	TICK_INT_PRIORITY	59
7.7.2.34	VDD_VALUE	59
7.8	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/RTE/Device/STM32F746NGHx/system_stm32f7xx.c File Reference	59
7.8.1	Detailed Description	60
7.9	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/gpioController.c File Reference	61
7.9.1	Detailed Description	61
7.10	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/LedBuzzerController.c File Reference	61
7.10.1	Detailed Description	62
7.11	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/main.c File Reference	62
7.11.1	Detailed Description	63
7.12	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/motorController.c File Reference	63
7.12.1	Detailed Description	63
7.13	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/pirController.c File Reference	64
7.13.1	Detailed Description	64
7.14	C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/touchInputController.c File Reference	64
7.14.1	Detailed Description	65

Chapter 1

Security System (Intruder Alarm)

Author

Tiberiu Simion Voicu, loakim loakim

Our system models a security system. Initially the system shows a welcome message on screen. When the button is pressed the PIR sensor is initialized and when it detects someone it turns on the lights and prompts for a password. If a wrong password is inserted then the alarm starts and prompts for a password again. When the correct password is inserted the door opens.

Our system when initialized waits for an interrupt to be activated from pressing the button and then when the PIR sensor scans someone it sends an interrupt to ask for a password. Moreover every touch interaction is processed through interrupts.

Hardware used:

- STM32f7 - Discovery Board
- PIR sensor (2pts)
- 2 LEDs (1pt)
- Buzzer (1pt)
- DC motor (2pts)
- Button (1pt)
- Touchscreen (1pt)
- GLCD (1pt)
- Interrupt (2pts)
- Total of points: 11

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

GPIO_Controller	9
GPIO_Controller_Type_Definitions	10
GPIO_Controller_public_functions	11
LedBuzzerButton_Controller	15
LedBuzzerButton_Controller_public_functions	16
Motor_Controller	19
Motor_Controller_public_functions	20
PIR_Controller	22
PIR_Controller_public_functions	23
TouchInput_Controller	24
TouchInput_Controller_public_functions	25
TouchInput_Controller_private_variables	30
TouchInput_Controller_private_functions	31
EXTI_IRQ_handler	27
MAIN	29
CMSIS	35
Stm32f7xx_system	36
STM32F7xx_System_Private_Includes	37
STM32F7xx_System_Private_TypesDefinitions	38
STM32F7xx_System_Private_Defines	39
STM32F7xx_System_Private_Macros	40
STM32F7xx_System_Private_Variables	41
STM32F7xx_System_Private_FunctionPrototypes	42
STM32F7xx_System_Private_Functions	43

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

_GPIO_PIN	Typdef struct to include GPIO port and pin	45
---------------------------	--	----

Chapter 4

File Index

4.1 File List

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

C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ defines.h This file contains definitions of variables used throughout the project	47
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ gpioController.h This file defines all the digital arduino pins and provides definitions of the functions used to control the GPIO pins	48
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ LedBuzzerController.h This file provide definitions of the functions used to control the control the LEDs, buzzer and button	49
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ motorController.h This file defines the functions used to control the motor	49
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ pirController.h This file defines the functions used to control the pir sensor	50
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ sdController.h	??
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/ touchInputController.h This file defines the functions used to control the touchscreen and keypad	51
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/RTE/ RTE_Components.h	??
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/RTE/CMSIS_Driver/ SPI_MultiSlave_Config.h	??
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/RTE/Device/STM32F746NGHx/ RTE_Device.h	??
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/RTE/Device/STM32F746NGHx/ stm32f7xx_hal_conf.h HAL configuration file	52
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/RTE/Device/STM32F746NGHx/ system_stm32f7xx.c CMSIS Cortex-M7 Device Peripheral Access Layer System Source File	59

C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ gpioController.c This file provides implementation of the functions used to control the GPIO pins	61
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ LedBuzzerController.c This file provide implementation of the functions used to control the control the LEDs, buzzer and button	61
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ main.c Main file used to initialise all the functions to handle the functionality of the project	62
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ motorController.c This file provides implementation of the functions used to control the motor	63
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ pirController.c This file provides implementation of the functions used to control the pir sensor	64
C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/src/ touchInputController.c This file provides implementation of the functions used to control the touchscreen and keypad .	64

Chapter 5

Module Documentation

5.1 GPIO_Controller

functions to control GPIOs

Modules

- [GPIO_Controller_Type_Definitions](#)
type definitions used in the functions to control GPIOs
- [GPIO_Controller_public_functions](#)
functions to control GPIOs

5.1.1 Detailed Description

functions to control GPIOs

5.2 GPIO_Controller_Type_Definitions

type definitions used in the functions to control GPIOs

Data Structures

- struct [_GPIO_PIN](#)

Typdef struct to include GPIO port and pin.

Typedefs

- typedef struct [_GPIO_PIN](#) [GPIO_PIN](#)

Typdef struct to include GPIO port and pin.

5.2.1 Detailed Description

type definitions used in the functions to control GPIOs

5.2.2 Typedef Documentation

5.2.2.1 GPIO_PIN

```
typedef struct \_GPIO\_PIN GPIO\_PIN
```

Typdef struct to include GPIO port and pin.

Parameters

<i>GPIO_TypeDef</i>	GPIO Port
<i>uint16_t</i>	GPIO Pin

Return values

<i>GPIO_PIN</i>	
-----------------	--

5.3 GPIO_Controller_public_functions

functions to control GPIOs

Functions

- void `enableClocksGPIO` (void)
Enable GPIO clocks.
- `GPIO_PIN` `getPin` (uint8_t dIndex)
Get gpio pin.
- void `initGPIO` (uint32_t dIndex, uint32_t mode, uint32_t pull, uint32_t speed, uint32_t alternate)
Initiate given GPIO pin based on provided parameters.
- void `digitalWrite` (uint8_t dIndex, uint8_t set)
Set gpio pin 1 or 0 if it's set in output mode.
- uint8_t `digitalRead` (uint8_t dIndex)
Read the value of a GPIO pin if it's set in input mode.
- void `delayMicro` (uint64_t micros)
To be used for small delays needed in interrupts.

5.3.1 Detailed Description

functions to control GPIOs

5.3.2 Function Documentation

5.3.2.1 `delayMicro()`

```
void delayMicro (
    uint64_t micros )
```

To be used for small delays needed in interrupts.

hal one is based on clocks which are paused when the context changes

Parameters

<i>micros</i>	Approximate time in microsec
---------------	------------------------------

Return values

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

5.3.2.2 digitalRead()

```
uint8_t digitalRead (
    uint8_t dIndex )
```

Read the value of a GPIO pin if it's set in input mode.

Parameters

<i>dIndex</i>	index from gpio pins array
---------------	----------------------------

Return values

<i>uint8_t</i>	0 or 1 based on value of GPIO pin
----------------	-----------------------------------

5.3.2.3 digitalWrite()

```
void digitalWrite (
    uint8_t dIndex,
    uint8_t set )
```

Set gpio pin 1 or 0 if it's set in output mode.

Parameters

<i>dIndex</i>	index from gpio pins array
<i>set</i>	set gpio as either 0 or 1

Return values

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

5.3.2.4 enableClocksGPIO()

```
void enableClocksGPIO (
    void )
```

Enable GPIO clocks.

Parameters

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

Return values

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

5.3.2.5 getPin()

```
GPIO_PIN getPin (
    uint8_t dIndex )
```

Get gpio pin.

Parameters

<i>dIndex</i>	index from gpio pins array
---------------	----------------------------

Return values

<i>GPIO_PIN</i>	the GPIO pin belonging to dIndex number on the board
-----------------	--

5.3.2.6 initGPIO()

```
void initGPIO (
    uint32_t dIndex,
    uint32_t mode,
    uint32_t pull,
    uint32_t speed,
    uint32_t alternate )
```

Initiate given GPIO pin based on provided parameters.

Note

Uses hal gpio library

Parameters

<i>dIndex</i>	index from gpio pins array
<i>mode</i>	mode of GPIO from mode typedef in hal gpio lib
<i>pull</i>	pull of GPIO from pull typedef in hal gpio lib
<i>speed</i>	speed of GPIO from speed typedef in hal gpio lib
<i>alternate</i>	optional parameter only used if mode is alternate(alternate typedef in gpio lib)

Return values

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

5.4 LedBuzzerButton_Controller

functions to control LEDs, buzzer and button

Modules

- [LedBuzzerButton_Controller_public_functions](#)
functions to control LEDs, buzzer and button

5.4.1 Detailed Description

functions to control LEDs, buzzer and button

5.5 LedBuzzerButton_Controller_public_functions

functions to control LEDs, buzzer and button

Functions

- void `initBuzzer` (int8_t Dindex)
Initiate buzzer.
- void `initLed` (int8_t Dindex)
Initiate Led.
- void `initButton` (void)
Initiate button and interrupt.
- void `startAlarm` (void)
Turn buzzer on.
- void `stopAlarm` (void)
Turn buzzer and Leds Off.
- void `startLights` (void)
Turn the 2 Leds on.

5.5.1 Detailed Description

functions to control LEDs, buzzer and button

5.5.2 Function Documentation

5.5.2.1 `initButton()`

```
void initButton (
    void )
```

Initiate button and interrupt.

Parameters

None	
------	--

Return values

None	
------	--

5.5.2.2 initBuzzer()

```
void initBuzzer (
    int8_t Dindex )
```

Initiate buzzer.

Parameters

<i>dIndex</i>	
---------------	--

Return values

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

5.5.2.3 initLed()

```
void initLed (
    int8_t Dindex )
```

Initiate Led.

Parameters

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

Return values

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

5.5.2.4 startAlarm()

```
void startAlarm (
    void )
```

Turn buzzer on.

Parameters

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

Return values

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

5.5.2.5 startLights()

```
void startLights (
    void )
```

Turn the 2 Leds on.

Parameters

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

Return values

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

5.5.2.6 stopAlarm()

```
void stopAlarm (
    void )
```

Turn buzzer and Leds Off.

Parameters

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

Return values

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

5.6 Motor_Controller

functions to control the motor

Modules

- [Motor_Controller_public_functions](#)
functions to control the motor

5.6.1 Detailed Description

functions to control the motor

5.7 Motor_Controller_public_functions

functions to control the motor

Functions

- void `initMotor` (int8_t dIndex)
Initiate Motor.
- void `moveDoor` (void)
Turn on motor to move the door 180 degrees.

5.7.1 Detailed Description

functions to control the motor

5.7.2 Function Documentation

5.7.2.1 initMotor()

```
void initMotor (
    int8_t dIndex )
```

Initiate Motor.

Parameters

<i>dIndex</i>	index belonging to GPIO Pin on board to be used for controlling the motor
---------------	---

Return values

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

5.7.2.2 moveDoor()

```
void moveDoor (
    void )
```

Turn on motor to move the door 180 degrees.

Parameters

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

Return values

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

5.8 PIR_Controller

functions to control the PIR sensor

Modules

- [PIR_Controller_public_functions](#)
functions to control the PIR sensor

5.8.1 Detailed Description

functions to control the PIR sensor

5.9 PIR_Controller_public_functions

functions to control the PIR sensor

Functions

- void `initPir` (void)
Initiate Passive Infrared Sensor and interrupt.

5.9.1 Detailed Description

functions to control the PIR sensor

5.9.2 Function Documentation

5.9.2.1 `initPir()`

```
void initPir (  
    void )
```

Initiate Passive Infrared Sensor and interrupt.

Parameters

None	
------	--

Return values

None	
------	--

5.10 TouchInput_Controller

functions to control the touchscreen and keypad

Modules

- [TouchInput_Controller_public_functions](#)
functions to control the touchscreen and keypad
- [TouchInput_Controller_private_variables](#)
variables used in functions to control the touchscreen and keypad
- [TouchInput_Controller_private_functions](#)
functions to control the touchscreen and keypad

5.10.1 Detailed Description

functions to control the touchscreen and keypad

5.11 TouchInput_Controller_public_functions

functions to control the touchscreen and keypad

Functions

- void `initGLCD` (void)
Initiate GLCD and Touch.
- void `drawKeypad` (void)
Draw keypad on GLCD.
- char `getTouch` (void)
Get the symbol resulted from touching a certain area of the display.
- void `addKey` (char key)
Add a key to password to be checked whether it's correct.

5.11.1 Detailed Description

functions to control the touchscreen and keypad

5.11.2 Function Documentation

5.11.2.1 `addKey()`

```
void addKey (
    char key )
```

Add a key to password to be checked whether it's correct.

Parameters

Key	the char that was selected
-----	----------------------------

Return values

None	
------	--

5.11.2.2 `drawKeypad()`

```
void drawKeypad (
    void )
```

Draw keypad on GLCD.

Parameters

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

Return values

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

5.11.2.3 getTouch()

```
char getTouch (  
                void )
```

Get the symbol resulted from touching a certain area of the display.

Parameters

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

Return values

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

5.11.2.4 initGLCD()

```
void initGLCD (  
                void )
```

Initiate GLCD and Touch.

Parameters

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

Return values

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

5.12 EXTI_IRQ_handler

functions to manage the interrupts used for the control of the system

Functions

- void [HAL_GPIO_EXTI_Callback](#) (uint16_t GPIO_Pin)
Initiate Motor.
- void [EXTI0_IRQHandler](#) (void)
Handle line 0 interrupt.
- void [EXTI4_IRQHandler](#) ()
Handle Interrupts on line 4 for Touch input.
- void [EXTI15_10_IRQHandler](#) (void)
Handle Interrupts on line 15-10 for Touch input.

5.12.1 Detailed Description

functions to manage the interrupts used for the control of the system

5.12.2 Function Documentation

5.12.2.1 EXTI0_IRQHandler()

```
void EXTI0_IRQHandler (
    void )
```

Handle line 0 interrupt.

Parameters

None	
------	--

Return values

None	
------	--

5.12.2.2 EXTI15_10_IRQHandler()

```
void EXTI15_10_IRQHandler (
    void )
```

Handle Interrupts on line 15-10 for Touch input.

Parameters

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

Return values

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

5.12.2.3 EXTI4_IRQHandler()

```
void EXTI4_IRQHandler ( )
```

Handle Interrupts on line 4 for Touch input.

Parameters

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

Return values

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

5.12.2.4 HAL_GPIO_EXTI_Callback()

```
void HAL_GPIO_EXTI_Callback (
    uint16_t GPIO_Pin )
```

Initiate Motor.

Parameters

<i>dIndex</i>	index belonging to GPIO Pin on board to be used for controlling the motor
---------------	---

Return values

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

5.13 MAIN

functions to handle the functionality of the project

Functions

- int **main** (void)

5.13.1 Detailed Description

functions to handle the functionality of the project

5.14 TouchInput_Controller_private_variables

variables used in functions to control the touchscreen and keypad

Variables

- uint16_t **box_x** = 50
- uint16_t **box_y** = 30
- uint16_t **char_x** = 15
- uint16_t **char_y** = 15
- uint16_t **w** = 50
- uint16_t **h** = 50
- char **passInput** [5]
- int **inputLen** = 0

5.14.1 Detailed Description

variables used in functions to control the touchscreen and keypad

5.15 TouchInput_Controller_private_functions

functions to control the touchscreen and keypad

Functions

- char `getPressedKey` (uint16_t x, uint16_t y)
Provide x and y coordinate on.
- void `wrongPass` (void)
Handles inserting wrong password and starts alarm.
- void `rightPass` (void)
Handles inserting right password and stops alarm.
- void `clearInput` (void)
Resets the input password string.
- void `initGLCD` ()
Initiate GLCD and Touch.
- void `drawKeypad` (void)
Draw keypad on GLCD.
- char `getTouch` ()
Get the symbol resulted from touching a certain area of the display.
- void `addKey` (char key)
Add a key to password to be checked whether it's correct.

5.15.1 Detailed Description

functions to control the touchscreen and keypad

5.15.2 Function Documentation

5.15.2.1 addKey()

```
void addKey (
    char key )
```

Add a key to password to be checked whether it's correct.

Parameters

Key	the char that was selected
-----	----------------------------

Return values

None	
------	--

5.15.2.2 clearInput()

```
void clearInput (
    void )
```

Resets the input password string.

Parameters

None	
------	--

Return values

None	
------	--

5.15.2.3 drawKeypad()

```
void drawKeypad (
    void )
```

Draw keypad on GLCD.

Parameters

None	
------	--

Return values

None	
------	--

5.15.2.4 getPressedKey()

```
char getPressedKey (
    uint16_t x,
    uint16_t y )
```

Provide x and y coordinate on.

touchscreen in order to get pressed key

Parameters

<i>x</i>	x coordinate on GLCD
<i>y</i>	y coordinate on GLCD

Return values

<i>char</i>	the symbol which was pressed on the touchscreen
-------------	---

5.15.2.5 getTouch()

```
char getTouch (
                void )
```

Get the symbol resulted from touching a certain area of the display.

Parameters

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

Return values

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

5.15.2.6 initGLCD()

```
void initGLCD (
                void )
```

Initiate GLCD and Touch.

Parameters

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

Return values

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

5.15.2.7 rightPass()

```
void rightPass (  
    void )
```

Handles inserting right password and stops alarm.

Parameters

None	
------	--

Return values

None	
------	--

5.15.2.8 wrongPass()

```
void wrongPass (  
    void )
```

Handles inserting wrong password and starts alarm.

Parameters

None	
------	--

Return values

None	
------	--

5.16 CMSIS

Modules

- [Stm32f7xx_system](#)

5.16.1 Detailed Description

5.17 Stm32f7xx_system

Modules

- [STM32F7xx_System_Private_Includes](#)
- [STM32F7xx_System_Private_TypesDefinitions](#)
- [STM32F7xx_System_Private_Defines](#)
- [STM32F7xx_System_Private_Macros](#)
- [STM32F7xx_System_Private_Variables](#)
- [STM32F7xx_System_Private_FunctionPrototypes](#)
- [STM32F7xx_System_Private_Functions](#)

5.17.1 Detailed Description

5.18 STM32F7xx_System_Private_Includes

Macros

- `#define HSE_VALUE ((uint32_t)25000000)`
- `#define HSI_VALUE ((uint32_t)16000000)`

5.18.1 Detailed Description

5.18.2 Macro Definition Documentation

5.18.2.1 HSE_VALUE

```
#define HSE_VALUE ((uint32_t)25000000)
```

Default value of the External oscillator in Hz

5.18.2.2 HSI_VALUE

```
#define HSI_VALUE ((uint32_t)16000000)
```

Value of the Internal oscillator in Hz

5.19 STM32F7xx_System_Private_TypesDefinitions

5.20 STM32F7xx_System_Private_Defines

Macros

- #define [VECT_TAB_OFFSET](#) 0x00

5.20.1 Detailed Description

5.20.2 Macro Definition Documentation

5.20.2.1 VECT_TAB_OFFSET

```
#define VECT_TAB_OFFSET 0x00
```

< Uncomment the following line if you need to relocate your vector Table in Internal SRAM. Vector Table base offset field. This value must be a multiple of 0x200.

5.21 STM32F7xx_System_Private_Macros

5.22 STM32F7xx_System_Private_Variables

Variables

- uint32_t **SystemCoreClock** = 16000000
- const uint8_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

5.22.1 Detailed Description

5.23 STM32F7xx_System_Private_FunctionPrototypes

5.24 STM32F7xx_System_Private_Functions

Functions

- void [SystemInit](#) (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void [SystemCoreClockUpdate](#) (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

5.24.1 Detailed Description

5.24.2 Function Documentation

5.24.2.1 SystemCoreClockUpdate()

```
void SystemCoreClockUpdate (  
    void )
```

Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Note

Each time the core clock (HCLK) changes, this function must be called to update SystemCoreClock variable value. Otherwise, any configuration based on this variable will be incorrect.

- The system frequency computed by this function is not the real frequency in the chip. It is calculated based on the predefined constant and the selected clock source:

- If SYSCLK source is HSI, SystemCoreClock will contain the [HSI_VALUE\(*\)](#)
- If SYSCLK source is HSE, SystemCoreClock will contain the [HSE_VALUE\(**\)](#)
- If SYSCLK source is PLL, SystemCoreClock will contain the [HSE_VALUE\(**\)](#) or [HSI_VALUE\(*\)](#) multiplied/divided by the PLL factors.

(*) HSI_VALUE is a constant defined in [stm32f7xx_hal_conf.h](#) file (default value 16 MHz) but the real value may vary depending on the variations in voltage and temperature.

(**) HSE_VALUE is a constant defined in [stm32f7xx_hal_conf.h](#) file (default value 25 MHz), user has to ensure that HSE_VALUE is same as the real frequency of the crystal used. Otherwise, this function may have wrong result.

- The result of this function could be not correct when using fractional value for HSE crystal.

Parameters

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

Return values

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

5.24.2.2 SystemInit()

```
void SystemInit (
    void )
```

Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.

Parameters

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

Return values

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

Chapter 6

Data Structure Documentation

6.1 _GPIO_PIN Struct Reference

Typdef struct to include GPIO port and pin.

```
#include <gpioController.h>
```

Data Fields

- GPIO_TypeDef * **port**
- uint16_t **pin**

6.1.1 Detailed Description

Typdef struct to include GPIO port and pin.

Parameters

<i>GPIO_TypeDef</i>	GPIO Port
<i>uint16_t</i>	GPIO Pin

Return values

<i>GPIO_PIN</i>	
-----------------	--

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

- C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/↵ Security_System_CW/include/[gpioController.h](#)

Chapter 7

File Documentation

7.1 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/defines.h File Reference

This file contains definitions of variables used throughout the project.

Macros

- `#define buzzer_GPIO_Num 0`
- `#define led1_GPIO_Num 1`
- `#define led2_GPIO_Num 2`
- `#define button_GPIO_Num 3`
- `#define motor_GPIO_Num 4`
- `#define pir_GPIO_Num 5`

7.1.1 Detailed Description

This file contains definitions of variables used throughout the project.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

27 April 2018

7.2 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/gpioController.h File Reference

This file defines all the digital arduino pins and provides definitions of the functions used to control the GPIO pins.

```
#include "stm32f7xx_hal_gpio.h"
#include "stm32f7xx_hal.h"
```

Data Structures

- struct [_GPIO_PIN](#)
Typdef struct to include GPIO port and pin.

Typedefs

- typedef struct [_GPIO_PIN](#) [GPIO_PIN](#)
Typdef struct to include GPIO port and pin.

Functions

- void [enableClocksGPIO](#) (void)
Enable GPIO clocks.
- [GPIO_PIN](#) [getPin](#) (uint8_t dIndex)
Get gpio pin.
- void [initGPIO](#) (uint32_t dIndex, uint32_t mode, uint32_t pull, uint32_t speed, uint32_t alternate)
Initiate given GPIO pin based on provided parameters.
- void [digitalWrite](#) (uint8_t dIndex, uint8_t set)
Set gpio pin 1 or 0 if it's set in output mode.
- uint8_t [digitalRead](#) (uint8_t dIndex)
Read the value of a GPIO pin if it's set in input mode.
- void [delayMicro](#) (uint64_t micros)
To be used for small delays needed in interrupts.

7.2.1 Detailed Description

This file defines all the digital arduino pins and provides definitions of the functions used to control the GPIO pins.

Author

Tiberiu Simion Voicu, loakim loakim

Version

V1.0.0

Date

17 April 2018

7.3 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/LedBuzzerController.h File Reference

This file provide definitions of the functions used to control the control the LEDs, buzzer and button.

```
#include "gpioController.h"
#include "defines.h"
```

Functions

- void [initBuzzer](#) (int8_t Dindex)
Initiate buzzer.
- void [initLed](#) (int8_t Dindex)
Initiate Led.
- void [initButton](#) (void)
Initiate button and interrupt.
- void [startAlarm](#) (void)
Turn buzzer on.
- void [stopAlarm](#) (void)
Turn buzzer and Leds Off.
- void [startLights](#) (void)
Turn the 2 Leds on.

7.3.1 Detailed Description

This file provide definitions of the functions used to control the control the LEDs, buzzer and button.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.4 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/motorController.h File Reference

This file defines the functions used to control the motor.

```
#include "gpioController.h"
#include "defines.h"
```

Functions

- void `initMotor` (int8_t dIndex)
Initiate Motor.
- void `moveDoor` (void)
Turn on motor to move the door 180 degrees.

7.4.1 Detailed Description

This file defines the functions used to control the motor.

Author

Tiberiu Simion Voicu, loakim loakim

Version

V1.0.0

Date

17 April 2018

7.5 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/pirController.h File Reference

This file defines the functions used to control the pir sensor.

```
#include "defines.h"
```

Functions

- void `initPir` (void)
Initiate Passive Infrared Sensor and interrupt.

7.5.1 Detailed Description

This file defines the functions used to control the pir sensor.

Author

Tiberiu Simion Voicu, loakim loakim

Version

V1.0.0

Date

17 April 2018

7.6 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/include/touchInputController.h File Reference

This file defines the functions used to control the touchscreen and keypad.

```
#include "GLCD_Config.h"
#include "Board_GLCD.h"
#include "Board_Touch.h"
#include "pirController.h"
#include "gpioController.h"
#include "LedBuzzerController.h"
#include "motorController.h"
#include "stm32746g_discovery_ts.h"
```

Macros

- `#define pass1 "1234"`

Functions

- void `initGLCD` (void)
Initiate GLCD and Touch.
- void `drawKeypad` (void)
Draw keypad on GLCD.
- char `getTouch` (void)
Get the symbol resulted from touching a certain area of the display.
- void `addKey` (char key)
Add a key to password to be checked whether it's correct.

Variables

- `GLCD_FONT GLCD_Font_16x24`

7.6.1 Detailed Description

This file defines the functions used to control the touchscreen and keypad.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.7 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/RTE/Device/STM32F746NGHx/stm32f7xx_hal_conf.h File Reference

HAL configuration file.

Macros

- #define **HSE_VALUE** 25000000U
This is the list of modules to be used in the HAL driver.
- #define **HSE_STARTUP_TIMEOUT** 100U
- #define **HSI_VALUE** 16000000U
Internal High Speed oscillator (HSI) value. This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL).
- #define **LSI_VALUE** 32000U
Internal Low Speed oscillator (LSI) value.
- #define **LSE_VALUE** 32768U
External Low Speed oscillator (LSE) value.
- #define **LSE_STARTUP_TIMEOUT** 5000U
- #define **EXTERNAL_CLOCK_VALUE** 12288000U
External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.
- #define **VDD_VALUE** 3300U
This is the HAL system configuration section.
- #define **TICK_INT_PRIORITY** 0x0FU
- #define **USE_RTOS** 0U
- #define **PREFETCH_ENABLE** 1U
- #define **ART_ACCELERATOR_ENABLE** 1U /* To enable instruction cache and prefetch */
- #define **MAC_ADDR0** 2U
Uncomment the line below to expanse the "assert_param" macro in the HAL drivers code.
- #define **MAC_ADDR1** 0U
- #define **MAC_ADDR2** 0U
- #define **MAC_ADDR3** 0U
- #define **MAC_ADDR4** 0U
- #define **MAC_ADDR5** 0U
- #define **ETH_RX_BUF_SIZE** ETH_MAX_PACKET_SIZE /* buffer size for receive */
- #define **ETH_TX_BUF_SIZE** ETH_MAX_PACKET_SIZE /* buffer size for transmit */
- #define **ETH_RXBUFNB** 4U /* 4 Rx buffers of size ETH_RX_BUF_SIZE */
- #define **ETH_TXBUFNB** 4U /* 4 Tx buffers of size ETH_TX_BUF_SIZE */
- #define **DP83848_PHY_ADDRESS** 0x01U
- #define **PHY_RESET_DELAY** 0x000000FFU
- #define **PHY_CONFIG_DELAY** 0x00000FFFU
- #define **PHY_READ_TO** 0x0000FFFFU
- #define **PHY_WRITE_TO** 0x0000FFFFU
- #define **PHY_BCR** ((uint16_t)0x00U)
- #define **PHY_BSR** ((uint16_t)0x01U)
- #define **PHY_RESET** ((uint16_t)0x8000U)
- #define **PHY_LOOPBACK** ((uint16_t)0x4000U)
- #define **PHY_FULLDUPLEX_100M** ((uint16_t)0x2100U)
- #define **PHY_HALFDUPLEX_100M** ((uint16_t)0x2000U)
- #define **PHY_FULLDUPLEX_10M** ((uint16_t)0x0100U)

- #define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
- #define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
- #define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
- #define PHY_POWERDOWN ((uint16_t)0x0800U)
- #define PHY_ISOLATE ((uint16_t)0x0400U)
- #define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
- #define PHY_LINKED_STATUS ((uint16_t)0x0004U)
- #define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
- #define PHY_SR ((uint16_t)0x10U)
- #define PHY_MICR ((uint16_t)0x11U)
- #define PHY_MISR ((uint16_t)0x12U)
- #define PHY_LINK_STATUS ((uint16_t)0x0001U)
- #define PHY_SPEED_STATUS ((uint16_t)0x0002U)
- #define PHY_DUPLEX_STATUS ((uint16_t)0x0004U)
- #define PHY_MICR_INT_EN ((uint16_t)0x0002U)
- #define PHY_MICR_INT_OE ((uint16_t)0x0001U)
- #define PHY_MISR_LINK_INT_EN ((uint16_t)0x0020U)
- #define PHY_LINK_INTERRUPT ((uint16_t)0x2000U)
- #define USE_SPI_CRC 1U
- #define assert_param(expr) ((void)0)

Include module's header file.

7.7.1 Detailed Description

HAL configuration file.

Author

MCD Application Team

Version

V1.2.0 modified by ARM

Date

23-September-2016

Attention

© COPYRIGHT(c) 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7.7.2 Macro Definition Documentation

7.7.2.1 EXTERNAL_CLOCK_VALUE

```
#define EXTERNAL_CLOCK_VALUE 12288000U
```

External clock source for I2S peripheral This value is used by the I2S HAL module to compute the I2S clock source frequency, this source is inserted directly through I2S_CKIN pad.

Value of the Internal oscillator in Hz

7.7.2.2 HSE_STARTUP_TIMEOUT

```
#define HSE_STARTUP_TIMEOUT 100U
```

Time out for HSE start up, in ms

7.7.2.3 HSE_VALUE

```
#define HSE_VALUE 25000000U
```

This is the list of modules to be used in the HAL driver.

Adjust the value of External High Speed oscillator (HSE) used in your application. This value is used by the RCC HAL module to compute the system frequency (when HSE is used as system clock source, directly or through the PLL). Value of the External oscillator in Hz

7.7.2.4 HSI_VALUE

```
#define HSI_VALUE 16000000U
```

Internal High Speed oscillator (HSI) value. This value is used by the RCC HAL module to compute the system frequency (when HSI is used as system clock source, directly or through the PLL).

Value of the Internal oscillator in Hz

7.7.2.5 LSE_STARTUP_TIMEOUT

```
#define LSE_STARTUP_TIMEOUT 5000U
```

Time out for LSE start up, in ms

7.7.2.6 LSE_VALUE

```
#define LSE_VALUE 32768U
```

External Low Speed oscillator (LSE) value.

< Value of the Internal Low Speed oscillator in Hz The real value may vary depending on the variations in voltage and temperature. Value of the External Low Speed oscillator in Hz

7.7.2.7 LSI_VALUE

```
#define LSI_VALUE 32000U
```

Internal Low Speed oscillator (LSI) value.

LSI Typical Value in Hz

7.7.2.8 PHY_AUTONEGO_COMPLETE

```
#define PHY_AUTONEGO_COMPLETE ((uint16_t)0x0020U)
```

Auto-Negotiation process completed

7.7.2.9 PHY_AUTONEGOTIATION

```
#define PHY_AUTONEGOTIATION ((uint16_t)0x1000U)
```

Enable auto-negotiation function

7.7.2.10 PHY_BCR

```
#define PHY_BCR ((uint16_t)0x00U)
```

Transceiver Basic Control Register

7.7.2.11 PHY_BSR

```
#define PHY_BSR ((uint16_t)0x01U)
```

Transceiver Basic Status Register

7.7.2.12 PHY_DUPLEX_STATUS

```
#define PHY_DUPLEX_STATUS ((uint16_t)0x0004U)
```

PHY Duplex mask

7.7.2.13 PHY_FULLDUPLEX_100M

```
#define PHY_FULLDUPLEX_100M ((uint16_t)0x2100U)
```

Set the full-duplex mode at 100 Mb/s

7.7.2.14 PHY_FULLDUPLEX_10M

```
#define PHY_FULLDUPLEX_10M ((uint16_t)0x0100U)
```

Set the full-duplex mode at 10 Mb/s

7.7.2.15 PHY_HALFDUPLEX_100M

```
#define PHY_HALFDUPLEX_100M ((uint16_t)0x2000U)
```

Set the half-duplex mode at 100 Mb/s

7.7.2.16 PHY_HALFDUPLEX_10M

```
#define PHY_HALFDUPLEX_10M ((uint16_t)0x0000U)
```

Set the half-duplex mode at 10 Mb/s

7.7.2.17 PHY_ISOLATE

```
#define PHY_ISOLATE ((uint16_t)0x0400U)
```

Isolate PHY from MII

7.7.2.18 PHY_JABBER_DETECTION

```
#define PHY_JABBER_DETECTION ((uint16_t)0x0002U)
```

Jabber condition detected

7.7.2.19 PHY_LINK_INTERRUPT

```
#define PHY_LINK_INTERRUPT ((uint16_t)0x2000U)
```

PHY link status interrupt mask

7.7.2.20 PHY_LINK_STATUS

```
#define PHY_LINK_STATUS ((uint16_t)0x0001U)
```

PHY Link mask

7.7.2.21 PHY_LINKED_STATUS

```
#define PHY_LINKED_STATUS ((uint16_t)0x0004U)
```

Valid link established

7.7.2.22 PHY_LOOPBACK

```
#define PHY_LOOPBACK ((uint16_t)0x4000U)
```

Select loop-back mode

7.7.2.23 PHY_MICR

```
#define PHY_MICR ((uint16_t)0x11U)
```

MI Interrupt Control Register

7.7.2.24 PHY_MICR_INT_EN

```
#define PHY_MICR_INT_EN ((uint16_t)0x0002U)
```

PHY Enable interrupts

7.7.2.25 PHY_MICR_INT_OE

```
#define PHY_MICR_INT_OE ((uint16_t)0x0001U)
```

PHY Enable output interrupt events

7.7.2.26 PHY_MISR

```
#define PHY_MISR ((uint16_t)0x12U)
```

MII Interrupt Status and Misc. Control Register

7.7.2.27 PHY_MISR_LINK_INT_EN

```
#define PHY_MISR_LINK_INT_EN ((uint16_t)0x0020U)
```

Enable Interrupt on change of link status

7.7.2.28 PHY_POWERDOWN

```
#define PHY_POWERDOWN ((uint16_t)0x0800U)
```

Select the power down mode

7.7.2.29 PHY_RESET

```
#define PHY_RESET ((uint16_t)0x8000U)
```

PHY Reset

7.7.2.30 PHY_RESTART_AUTONEGOTIATION

```
#define PHY_RESTART_AUTONEGOTIATION ((uint16_t)0x0200U)
```

Restart auto-negotiation function

7.7.2.31 PHY_SPEED_STATUS

```
#define PHY_SPEED_STATUS ((uint16_t)0x0002U)
```

PHY Speed mask

7.7.2.32 PHY_SR

```
#define PHY_SR ((uint16_t)0x10U)
```

PHY status register Offset

7.7.2.33 TICK_INT_PRIORITY

```
#define TICK_INT_PRIORITY 0x0FU
```

tick interrupt priority

7.7.2.34 VDD_VALUE

```
#define VDD_VALUE 3300U
```

This is the HAL system configuration section.

Value of VDD in mv

7.8 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/RTE/Device/STM32F746NGHx/system_stm32f7xx.c File Reference

CMSIS Cortex-M7 Device Peripheral Access Layer System Source File.

```
#include "stm32f7xx.h"
```

Macros

- #define **HSE_VALUE** ((uint32_t)25000000)
- #define **HSI_VALUE** ((uint32_t)16000000)
- #define **VECT_TAB_OFFSET** 0x00

Functions

- void **SystemInit** (void)
Setup the microcontroller system Initialize the Embedded Flash Interface, the PLL and update the SystemFrequency variable.
- void **SystemCoreClockUpdate** (void)
Update SystemCoreClock variable according to Clock Register Values. The SystemCoreClock variable contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.

Variables

- uint32_t **SystemCoreClock** = 16000000
- const uint8_t **AHBPrescTable** [16] = {0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 3, 4, 6, 7, 8, 9}
- const uint8_t **APBPrescTable** [8] = {0, 0, 0, 0, 1, 2, 3, 4}

7.8.1 Detailed Description

CMSIS Cortex-M7 Device Peripheral Access Layer System Source File.

Author

MCD Application Team

Version

V1.2.0

Date

30-December-2016 This file provides two functions and one global variable to be called from user application:

- [SystemInit\(\)](#): This function is called at startup just after reset and before branch to main program. This call is made inside the "startup_stm32f7xx.s" file.
- SystemCoreClock variable: Contains the core clock (HCLK), it can be used by the user application to setup the SysTick timer or configure other parameters.
- [SystemCoreClockUpdate\(\)](#): Updates the variable SystemCoreClock and must be called whenever the core clock is changed during program execution.

Attention

© COPYRIGHT 2016 STMicroelectronics

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of STMicroelectronics nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

7.9 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/gpioController.c File Reference

This file provides implementation of the functions used to control the GPIO pins.

```
#include "gpioController.h"
```

Functions

- **GPIO_PIN** `getPin` (uint8_t dIndex)
Get gpio pin.
- void **enableClocksGPIO** (void)
Enable GPIO clocks.
- void **initGPIO** (uint32_t Dindex, uint32_t mode, uint32_t pull, uint32_t speed, uint32_t alternate)
Initiate given GPIO pin based on provided parameters.
- void **digitalWrite** (uint8_t num, uint8_t set)
Set gpio pin 1 or 0 if it's set in output mode.
- void **init_button** (int8_t Dindex)
- uint8_t **digitalRead** (uint8_t Dindex)
Read the value of a GPIO pin if it's set in input mode.
- void **delayMicro** (uint64_t micros)
To be used for small delays needed in interrupts.

7.9.1 Detailed Description

This file provides implementation of the functions used to control the GPIO pins.

Author

Tiberiu Simion Voicu, loakim loakim

Version

V1.0.0

Date

17 April 2018

7.10 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/LedBuzzerController.c File Reference

This file provide implementation of the functions used to control the control the LEDs, buzzer and button.

```
#include "LedBuzzerController.h"
```

Functions

- void `initBuzzer` (int8_t Dindex)
Initiate buzzer.
- void `initLed` (int8_t Dindex)
Initiate Led.
- void `initButton` ()
Initiate button and interrupt.
- void `startLights` ()
Turn the 2 Leds on.
- void `startAlarm` ()
Turn buzzer on.
- void `stopAlarm` ()
Turn buzzer and Leds Off.

7.10.1 Detailed Description

This file provide implementation of the functions used to control the control the LEDs, buzzer and button.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.11 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/main.c File Reference

main file used to initialise all the functions to handle the functionality of the project

```
#include "stm32746g_discovery_sdram.h"
#include "pirController.h"
#include "LedBuzzerController.h"
#include "touchInputController.h"
#include "motorController.h"
```

Functions

- int `main` (void)

7.11.1 Detailed Description

main file used to initialise all the functions to handle the functionality of the project

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.12 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/motorController.c File Reference

This file provides implementation of the functions used to control the motor.

```
#include "motorController.h"
```

Functions

- void `initMotor` (int8_t Dindex)
Initiate Motor.
- void `moveDoor` ()
Turn on motor to move the door 180 degrees.

7.12.1 Detailed Description

This file provides implementation of the functions used to control the motor.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.13 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/pirController.c File Reference

This file provides implementation of the functions used to control the pir sensor.

```
#include "gpioController.h"
#include "pirController.h"
```

Functions

- void `initPir` ()
Initiate Passive Infrared Sensor and interrupt.

7.13.1 Detailed Description

This file provides implementation of the functions used to control the pir sensor.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

7.14 C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded_Systems/coursework/Security-System-embedded/Security_System_CW/src/touchInputController.c File Reference

This file provides implementation of the functions used to control the touchscreen and keypad.

```
#include "touchInputController.h"
```

Functions

- char `getPressedKey` (uint16_t x, uint16_t y)
Provide x and y coordinate on.
- void `wrongPass` (void)
Handles inserting wrong password and starts alarm.
- void `rightPass` (void)
Handles inserting right password and stops alarm.
- void `clearInput` (void)
Resets the input password string.
- void `initGLCD` ()
Initiate GLCD and Touch.
- void `drawKeypad` (void)
Draw keypad on GLCD.
- char `getTouch` ()
Get the symbol resulted from touching a certain area of the display.
- void `addKey` (char key)
Add a key to password to be checked whether it's correct.

Variables

- uint16_t **box_x** = 50
- uint16_t **box_y** = 30
- uint16_t **char_x** = 15
- uint16_t **char_y** = 15
- uint16_t **w** = 50
- uint16_t **h** = 50
- char **passInput** [5]
- int **inputLen** = 0

7.14.1 Detailed Description

This file provides implementation of the functions used to control the touchscreen and keypad.

Author

Tiberiu Simion Voicu, Ioakim Ioakim

Version

V1.0.0

Date

17 April 2018

Index

- [_GPIO_PIN, 45](#)
- [addKey](#)
 - [TouchInput_Controller_private_functions, 31](#)
 - [TouchInput_Controller_public_functions, 25](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/RTE/↔Device/STM32F746NGHx/stm32f7xx_hal↔_conf.h, 52](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/RT↔E/Device/STM32F746NGHx/system_↔stm32f7xx.c, 59](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/↔LedBuzzerController.h, 49](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/defined↔h, 47](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/gpio↔Controller.h, 48](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/motor↔Controller.h, 49](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/pir↔Controller.h, 50](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/include/touch↔InputController.h, 51](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/Led↔BuzzerController.c, 61](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/gpio↔Controller.c, 61](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/main.c, 62](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/motor↔Controller.c, 63](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/pir↔Controller.c, 64](#)
- [C:/Users/ioaki/Dropbox/UEA/year3/1-Embedded↔_Systems/coursework/Security-System-embedded/Security_System_CW/src/touch↔InputController.c, 64](#)
- [CMSIS, 35](#)
- [clearInput](#)
 - [TouchInput_Controller_private_functions, 32](#)
- [delayMicro](#)
 - [GPIO_Controller_public_functions, 11](#)
- [digitalRead](#)
 - [GPIO_Controller_public_functions, 11](#)
- [digitalWrite](#)
 - [GPIO_Controller_public_functions, 12](#)
- [drawKeypad](#)
 - [TouchInput_Controller_private_functions, 32](#)
 - [TouchInput_Controller_public_functions, 25](#)
- [EXTERNAL_CLOCK_VALUE](#)
 - [stm32f7xx_hal_conf.h, 54](#)
- [EXTI0_IRQHandler](#)
 - [EXTI_IRQ_handler, 27](#)
- [EXTI15_10_IRQHandler](#)
 - [EXTI_IRQ_handler, 27](#)
- [EXTI4_IRQHandler](#)
 - [EXTI_IRQ_handler, 28](#)
- [EXTI_IRQ_handler, 27](#)
- [EXTI0_IRQHandler, 27](#)
- [EXTI15_10_IRQHandler, 27](#)
- [EXTI4_IRQHandler, 28](#)
- [HAL_GPIO_EXTI_Callback, 28](#)
- [enableClocksGPIO](#)
 - [GPIO_Controller_public_functions, 12](#)
- [GPIO_Controller, 9](#)
- [GPIO_Controller_Type_Definitions, 10](#)
- [GPIO_PIN, 10](#)
- [GPIO_Controller_public_functions, 11](#)
 - [delayMicro, 11](#)
 - [digitalRead, 11](#)

- digitalWrite, [12](#)
- enableClocksGPIO, [12](#)
- getPin, [13](#)
- initGPIO, [13](#)
- GPIO_PIN
 - GPIO_Controller_Type_Definitions, [10](#)
- getPin
 - GPIO_Controller_public_functions, [13](#)
- getPressedKey
 - TouchInput_Controller_private_functions, [32](#)
- getTouch
 - TouchInput_Controller_private_functions, [33](#)
 - TouchInput_Controller_public_functions, [26](#)
- HAL_GPIO_EXTI_Callback
 - EXTI_IRQ_handler, [28](#)
- HSE_STARTUP_TIMEOUT
 - stm32f7xx_hal_conf.h, [54](#)
- HSE_VALUE
 - STM32F7xx_System_Private_Includes, [37](#)
 - stm32f7xx_hal_conf.h, [54](#)
- HSI_VALUE
 - STM32F7xx_System_Private_Includes, [37](#)
 - stm32f7xx_hal_conf.h, [55](#)
- initButton
 - LedBuzzerButton_Controller_public_functions, [16](#)
- initBuzzer
 - LedBuzzerButton_Controller_public_functions, [16](#)
- initGLCD
 - TouchInput_Controller_private_functions, [33](#)
 - TouchInput_Controller_public_functions, [26](#)
- initGPIO
 - GPIO_Controller_public_functions, [13](#)
- initLed
 - LedBuzzerButton_Controller_public_functions, [17](#)
- initMotor
 - Motor_Controller_public_functions, [20](#)
- initPir
 - PIR_Controller_public_functions, [23](#)
- LSE_STARTUP_TIMEOUT
 - stm32f7xx_hal_conf.h, [55](#)
- LSE_VALUE
 - stm32f7xx_hal_conf.h, [55](#)
- LSI_VALUE
 - stm32f7xx_hal_conf.h, [55](#)
- LedBuzzerButton_Controller, [15](#)
- LedBuzzerButton_Controller_public_functions, [16](#)
 - initButton, [16](#)
 - initBuzzer, [16](#)
 - initLed, [17](#)
 - startAlarm, [17](#)
 - startLights, [18](#)
 - stopAlarm, [18](#)
- MAIN, [29](#)
- Motor_Controller, [19](#)
- Motor_Controller_public_functions, [20](#)
 - initMotor, [20](#)
 - moveDoor, [20](#)
- moveDoor
 - Motor_Controller_public_functions, [20](#)
- PHY_AUTONEGO_COMPLETE
 - stm32f7xx_hal_conf.h, [55](#)
- PHY_AUTONEGOTIATION
 - stm32f7xx_hal_conf.h, [55](#)
- PHY_BCR
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_BSR
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_DUPLEX_STATUS
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_FULLDUPLEX_100M
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_FULLDUPLEX_10M
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_HALFDUPLEX_100M
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_HALFDUPLEX_10M
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_ISOLATE
 - stm32f7xx_hal_conf.h, [56](#)
- PHY_JABBER_DETECTION
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_LINK_INTERRUPT
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_LINK_STATUS
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_LINKED_STATUS
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_LOOPBACK
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_MICR_INT_EN
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_MICR_INT_OE
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_MICR
 - stm32f7xx_hal_conf.h, [57](#)
- PHY_MISR_LINK_INT_EN
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_MISR
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_POWERDOWN
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_RESET
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_RESTART_AUTONEGOTIATION
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_SPEED_STATUS
 - stm32f7xx_hal_conf.h, [58](#)
- PHY_SR
 - stm32f7xx_hal_conf.h, [58](#)
- PIR_Controller, [22](#)
- PIR_Controller_public_functions, [23](#)
 - initPir, [23](#)

- rightPass
 - TouchInput_Controller_private_functions, 33
- STM32F7xx_System_Private_Defines, 39
 - VECT_TAB_OFFSET, 39
- STM32F7xx_System_Private_FunctionPrototypes, 42
- STM32F7xx_System_Private_Functions, 43
 - SystemCoreClockUpdate, 43
 - SystemInit, 44
- STM32F7xx_System_Private_Includes, 37
 - HSE_VALUE, 37
 - HSI_VALUE, 37
- STM32F7xx_System_Private_Macros, 40
- STM32F7xx_System_Private_TypesDefinitions, 38
- STM32F7xx_System_Private_Variables, 41
- startAlarm
 - LedBuzzerButton_Controller_public_functions, 17
- startLights
 - LedBuzzerButton_Controller_public_functions, 18
- stm32f7xx_hal_conf.h
 - EXTERNAL_CLOCK_VALUE, 54
 - HSE_STARTUP_TIMEOUT, 54
 - HSE_VALUE, 54
 - HSI_VALUE, 55
 - LSE_STARTUP_TIMEOUT, 55
 - LSE_VALUE, 55
 - LSI_VALUE, 55
 - PHY_AUTONEGO_COMPLETE, 55
 - PHY_AUTONEGOTIATION, 55
 - PHY_BCR, 56
 - PHY_BSR, 56
 - PHY_DUPLEX_STATUS, 56
 - PHY_FULLDUPLEX_100M, 56
 - PHY_FULLDUPLEX_10M, 56
 - PHY_HALFDUPLEX_100M, 56
 - PHY_HALFDUPLEX_10M, 56
 - PHY_ISOLATE, 56
 - PHY_JABBER_DETECTION, 57
 - PHY_LINK_INTERRUPT, 57
 - PHY_LINK_STATUS, 57
 - PHY_LINKED_STATUS, 57
 - PHY_LOOPBACK, 57
 - PHY_MICR_INT_EN, 57
 - PHY_MICR_INT_OE, 57
 - PHY_MICR, 57
 - PHY_MISR_LINK_INT_EN, 58
 - PHY_MISR, 58
 - PHY_POWERDOWN, 58
 - PHY_RESET, 58
 - PHY_RESTART_AUTONEGOTIATION, 58
 - PHY_SPEED_STATUS, 58
 - PHY_SR, 58
 - TICK_INT_PRIORITY, 58
 - VDD_VALUE, 59
- Stm32f7xx_system, 36
- stopAlarm
 - LedBuzzerButton_Controller_public_functions, 18
- SystemCoreClockUpdate
 - STM32F7xx_System_Private_Functions, 43
- SystemInit
 - STM32F7xx_System_Private_Functions, 44
- TICK_INT_PRIORITY
 - stm32f7xx_hal_conf.h, 58
- TouchInput_Controller, 24
- TouchInput_Controller_private_functions, 31
 - addKey, 31
 - clearInput, 32
 - drawKeypad, 32
 - getPressedKey, 32
 - getTouch, 33
 - initGLCD, 33
 - rightPass, 33
 - wrongPass, 34
- TouchInput_Controller_private_variables, 30
- TouchInput_Controller_public_functions, 25
 - addKey, 25
 - drawKeypad, 25
 - getTouch, 26
 - initGLCD, 26
- VDD_VALUE
 - stm32f7xx_hal_conf.h, 59
- VECT_TAB_OFFSET
 - STM32F7xx_System_Private_Defines, 39
- wrongPass
 - TouchInput_Controller_private_functions, 34