Schlosssteuerung

Generated by Doxygen 1.9.0

Chapter 1

Lock controller

This is the documentation page for a lock controller with a pin and puk. The whole project is tested on a arduino uno rev3 with an ATmega328P.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 keypad.c File Reference

This file contains all functions for the key detection.

```
#include <stdbool.h>
#include <avr/io.h>
#include <util/delay.h>
#include "led.h"
```

Functions

- void setup_keypad (void)

  Initiates the kepad for use.
- char keypad (void)

  reads the pressed key and returns pressed value

Variables

- int h =0

  Variables used in for loops.
- int v =0

  Variables used in for loops.
- const int rows =4

  Number of rows of keypad.
- const int columns =4

  Number of columnss of keypad
- const char Output [4] ={PORTD2,PORTD3,PORTD4,PORTD5}

  Array of pins used as output for rows of keypad.
- const char Input [4] ={PORTD6,PORTD7,PORTB0,PORTB1}

  Array of pins used as input for columnss of keypad.
- const char keys [4][4]

  Array representing the values on the Keypad.

3.1.1 Detailed Description

This file contains all functions for the key detection.

Author

Sebastian Pötter

Date

1 Jan 2021

This file controlles the keyinput and return of the pressed key

3.1.2 Function Documentation

3.1.2.1 keypad()

```
char keypad (
             void  )
```

reads the pressed key and returns pressed value

test if a button is pressed then test which button it is. If colomn Pin changed to Low button is pressed. By setting one row after another to High check if the Low Pin changes to High.

Returns

Char of the pressed button

Definition at line 78 of file keypad.c.

3.1.2.2 setup_keypad()

```
void setup_keypad (
             void  )
```

Initiates the kepad for use.

Set poin modes for the kepad, rows as Output, colomns as Input and all set to High Set Port 2,3,4,5 as Outpt

Set Port 6,7,8,9 as Input and High

Definition at line 48 of file keypad.c.

3.1.3 Variable Documentation

3.1.3.1 columns

```
const int columns =4
```

Number of columnss of keypad

Definition at line 27 of file keypad.c.

3.1.3.2 h

```
int h =0
```

Variables used in for loops.

Definition at line 20 of file keypad.c.

3.1.3.3 Input

```
const char Input[4] ={PORTD6,PORTD7,PORTB0,PORTB1}
```

Array of pins used as input for columnss of keypad.

Definition at line 31 of file keypad.c.

3.1.3.4 keys

```
const char keys[4][4]
```

Initial value:

```
= {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
}
```

Array representing the values on the Keypad.

Definition at line 33 of file keypad.c.

3.1.3.5 Output

```
const char Output[4] ={PORTD2,PORTD3,PORTD4,PORTD5}
```

Array of pins used as output for rows of keypad.

Definition at line 29 of file keypad.c.

3.1.3.6 rows

```
const int rows =4
```

Number of rows of keypad.

Definition at line 25 of file keypad.c.

3.1.3.7 v

```
int v =0
```

Variables used in for loops.

Definition at line 22 of file keypad.c.

3.2 keypad.h File Reference

Header file for keypad.c.

Functions

- char keypad (void)
  reads the pressed key and returns pressed value
- void setup_keypad (void)
  Initiates the kepad for use.

3.2.1 Detailed Description

Header file for keypad.c.

Author

Sebastian Pötter

Date

1 Jan 2021

This file controlles the keyinput and return of the pressed key

3.2.2 Function Documentation

3.2.2.1 keypad()

```
char keypad (
             void  )
```

reads the pressed key and returns pressed value

test if a button is pressed then test which button it is. If colomn Pin changed to Low button is pressed. By setting one row after another to High check if the Low Pin changes to High.

Returns

Char of the pressed button

Definition at line 78 of file keypad.c.

3.2.2.2 setup_keypad()

```
void setup_keypad (
             void  )
```

Initiates the kepad for use.

Set poin modes for the kepad, rows as Output, colomns as Input and all set to High Set Port 2,3,4,5 as Outpt

Set Port 6,7,8,9 as Input and High

Definition at line 48 of file keypad.c.

3.3 led.c File Reference

This is the file to controll the rgb led.

```
#include <stdbool.h>
#include <avr/io.h>
```

Functions

- void setup_LED (void)

  Initiates the LED for use.
- void setLED (bool redValue, bool greenValue, bool blueValue)

  Aktivates LED colors depending on paramters.

3.3.1 Detailed Description

This is the file to controll the rgb led.

Author

Markus Reinhold

Date

1 Jan 2021

3.3.2 Function Documentation

3.3.2.1 setLED()

```
void setLED (
             bool redValue,
             bool greenValue,
             bool blueValue )
```

Aktivates LED colors depending on paramters.

Set pin modes for the red, green and blue in the LED and init with all on. Aktivate LED's depending on the Parameters.

Parameters

redValue	bool paramter, if true the red LED will be turned on
greenValue	bool paramter, if true the green LED will be turned on
blueValue	bool paramter, if true the blue LED will be turned on

Definition at line 45 of file led.c.

3.3.2.2 setup_LED()

```
void setup_LED (
             void  )
```

Initiates the LED for use.

Set pin modes for the red, green and blue in the LED and init with all on

Definition at line 22 of file led.c.

3.4 led.h File Reference

Header file for led.c.

```
#include <stdbool.h>
```

Functions

- void setLED (bool redValue, bool greenValue, bool blueValue)

  Aktivates LED colors depending on paramters.
- void setup_LED (void)

  Initiates the LED for use.

3.4.1 Detailed Description

Header file for led.c.

Author

Markus Reinhold

Date

1 Jan 2021

3.4.2 Function Documentation

3.4.2.1 setLED()

```
void setLED (
            bool redValue,
            bool greenValue,
            bool blueValue )
```

Aktivates LED colors depending on paramters.

Set pin modes for the red, green and blue in the LED and init with all on. Aktivate LED's depending on the Parameters.

Parameters

redValue	bool paramter, if true the red LED will be turned on
greenValue	bool paramter, if true the green LED will be turned on
blueValue	bool paramter, if true the blue LED will be turned on

Definition at line 45 of file led.c.

3.4.2.2 setup_LED()

```
void setup_LED (
            void  )
```

Initiates the LED for use.

Set pin modes for the red, green and blue in the LED and init with all on

Definition at line 22 of file led.c.

3.5 main.c File Reference

This is the main file of the arduino project.

```
#include "keypad.h"
#include "led.h"
#include "states.h"
#include "millis.h"
#include <stdio.h>
#include <avr/io.h>
#include <util/atomic.h>
#include <avr/interrupt.h>
#include <util/delay.h>
```

Macros

- #define MS_DELAY 3000

Functions

- int main (void)

  Main function.

Variables

- const unsigned long period = 50

  used to make non-blocking delay
- unsigned long kdelay =0

  variable used in non-blocking delay

3.5.1 Detailed Description

This is the main file of the arduino project.

Author

Markus Reinhold

Date

1 Jan 2021

This file set up the keypad-pin and led-pin and controll the state machine

3.5.2 Macro Definition Documentation

3.5.2.1 MS_DELAY

```
#define MS_DELAY 3000
```

Definition at line 29 of file main.c.

3.5.3 Function Documentation

3.5.3.1 main()

```
int main (
            void  )
```

Main function.

Here is the programme entry point where the setup and the state machine stats

Definition at line 41 of file main.c.

3.5.4 Variable Documentation

3.5.4.1 kdelay

```
unsigned long kdelay =0
```

variable used in non-blocking delay

Definition at line 34 of file main.c.

3.5.4.2 period

```
const unsigned long period = 50
```

used to make non-blocking delay

Definition at line 32 of file main.c.

3.6 millis.c File Reference

Functions for the delay for the key press detection.

```
#include <avr/io.h>
#include <util/atomic.h>
#include <avr/interrupt.h>
```

Functions

- ISR (TIMER1_COMPA_vect)
- void init_millis (unsigned long f_cpu)
  Initiates the millis for use.
- unsigned long millis ()
  Used to get millis time.

Variables

- volatile unsigned long timer1_millis

3.6.1 Detailed Description

Functions for the delay for the key press detection.

Author

Markus Reinhold

Date

1 Jan 2021

3.6.2 Function Documentation

3.6.2.1 init_millis()

```
void init_millis (
            unsigned long f_cpu )
```

Initiates the millis for use.

Sets Timer Counter Register and Output Compare Register

Definition at line 35 of file millis.c.

3.6.2.2 ISR()

```
ISR (
            TIMER1_COMPA_vect  )
```

Definition at line 24 of file millis.c.

3.6.2.3 millis()

```
unsigned long millis (
            void  )
```

Used to get millis time.

returns the milliseconds elapsed since the program was started

Returns

millis time

Definition at line 58 of file millis.c.

3.6.3 Variable Documentation

3.6.3.1 timer1_millis

```
volatile unsigned long timer1_millis
```

Definition at line 21 of file millis.c.

3.7 millis.h File Reference

Header file for millis.c.

Functions

- unsigned long millis (void)
  Used to get millis time.
- void init_millis (unsigned long)
  Initiates the millis for use.

3.7.1 Detailed Description

Header file for millis.c.

Author

Markus Reinhold

Date

1 Jan 2021

3.7.2 Function Documentation

3.7.2.1 init_millis()

```
void init_millis (
             unsigned long f_cpu )
```

Initiates the millis for use.

Sets Timer Counter Register and Output Compare Register

Definition at line 35 of file millis.c.

3.7.2.2 millis()

```
unsigned long millis (
             void  )
```

Used to get millis time.

returns the milliseconds elapsed since the program was started

Returns

millis time

Definition at line 58 of file millis.c.

3.8 states.c File Reference

Controlls the state machine of the lock.

```
#include <stdio.h>
#include <stdbool.h>
#include <util/delay.h>
#include <avr/io.h>
#include "states.h"
#include "led.h"
#include "keypad.h"
```

Macros

- #define INIT 0
- #define KEY 1
- #define OPEN 2
- #define CHANGEKEY 3
- #define NEWKEY 4
- #define LOCKED 5

Functions

- void stateM (void)

  States of lock funktionality.

Variables

- int error = 0

  after 4 wrong keys => locked
- int state = INIT
- int stateOld = INIT
- int pos = 0

  current position of key
- char pin [4] = "****"

  init pin
- char puck [6] = "#12345"

  init puk
- int timeout = 0
- int openTimeout = 250

  time out for open state

3.8.1 Detailed Description

Controlls the state machine of the lock.

Author

Lukas Brüggemann

Date

1 Jan 2021

This file contains the state machine of the lock

3.8.2 Macro Definition Documentation

3.8.2.1 CHANGEKEY

```
#define CHANGEKEY 3
```

Definition at line 25 of file states.c.

3.8.2.2 INIT

```
#define INIT 0
```

Definition at line 22 of file states.c.

3.8.2.3 KEY

```
#define KEY 1
```

Definition at line 23 of file states.c.

3.8.2.4 LOCKED

```
#define LOCKED 5
```

Definition at line 27 of file states.c.

3.8.2.5 NEWKEY

```
#define NEWKEY 4
```

Definition at line 26 of file states.c.

3.8.2.6 OPEN

```
#define OPEN 2
```

Definition at line 24 of file states.c.

3.8.3 Function Documentation

3.8.3.1 stateM()

```
void stateM (
            void  )
```

States of lock funktionality.

Reads pressed button and depending on current state and pressed button to handle locks. Depending on state of the lock the indicator LED is set.

Definition at line 67 of file states.c.

3.8.4 Variable Documentation

3.8.4.1 error

```
int error = 0
```

after 4 wrong keys => locked

Definition at line 31 of file states.c.

3.8.4.2 openTimeout

```
int openTimeout = 250
```

time out for open state

Definition at line 42 of file states.c.

3.8.4.3 pin

```
char pin[4] = "****"
```

init pin

Definition at line 37 of file states.c.

3.8.4.4 pos

```
int pos = 0
```

current position of key

Definition at line 35 of file states.c.

3.8.4.5 puck

```
char puck[6] = "#12345"
```

init puk

Definition at line 39 of file states.c.

3.8.4.6 state

```
int state = INIT
```

Definition at line 32 of file states.c.

3.8.4.7 stateOld

```
int stateOld = INIT
```

Definition at line 33 of file states.c.

3.8.4.8 timeout

```
int timeout = 0
```

Definition at line 40 of file states.c.

3.9 states.h File Reference

header of the states.c

Functions

- void stateM (void)

  States of lock funktionality.

3.9.1 Detailed Description

header of the states.c

Author

Lukas Brüggemann

Date

1 Jan 2021

3.9.2 Function Documentation

3.9.2.1 stateM()

```
void stateM (
             void  )
```

States of lock funktionality.

Reads pressed button and depending on current state and pressed button to handle locks. Depending on state of the lock the indicator LED is set.

Definition at line 67 of file states.c.

Index