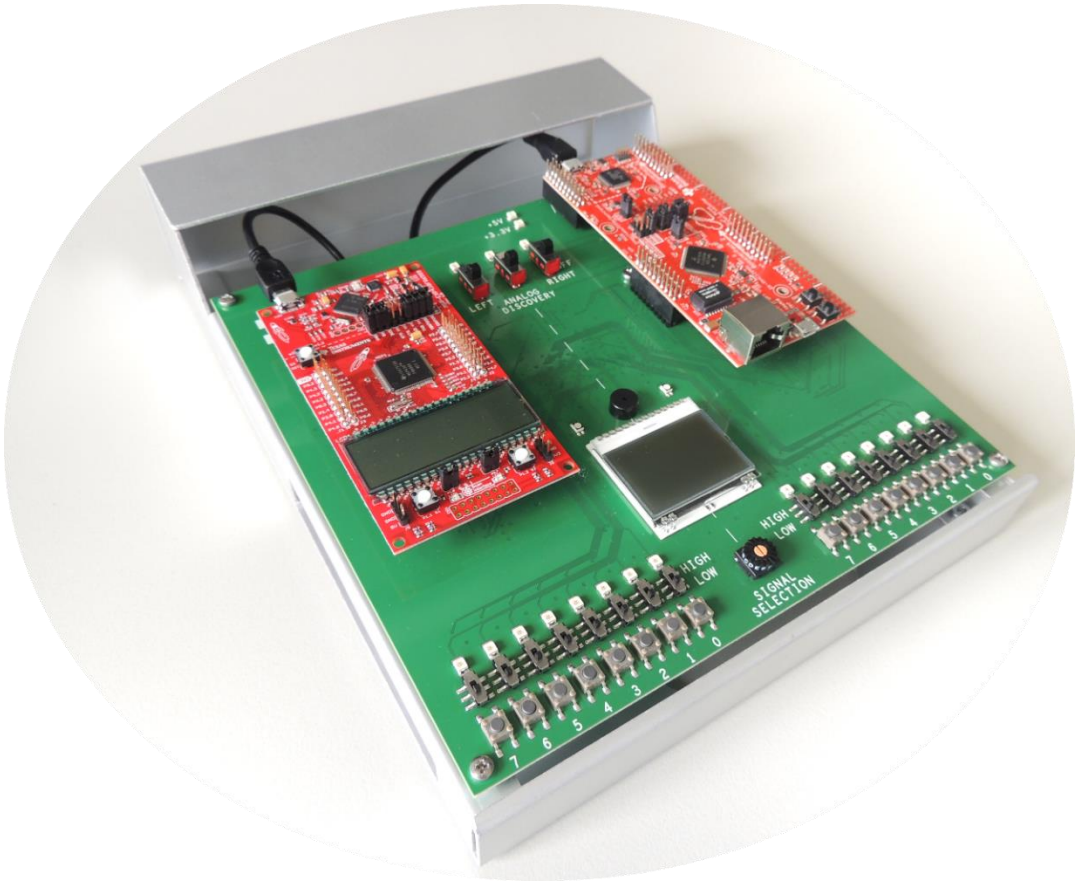
OST Ostschweizer Fachhochschule

ESP_Readme

Version 1.51

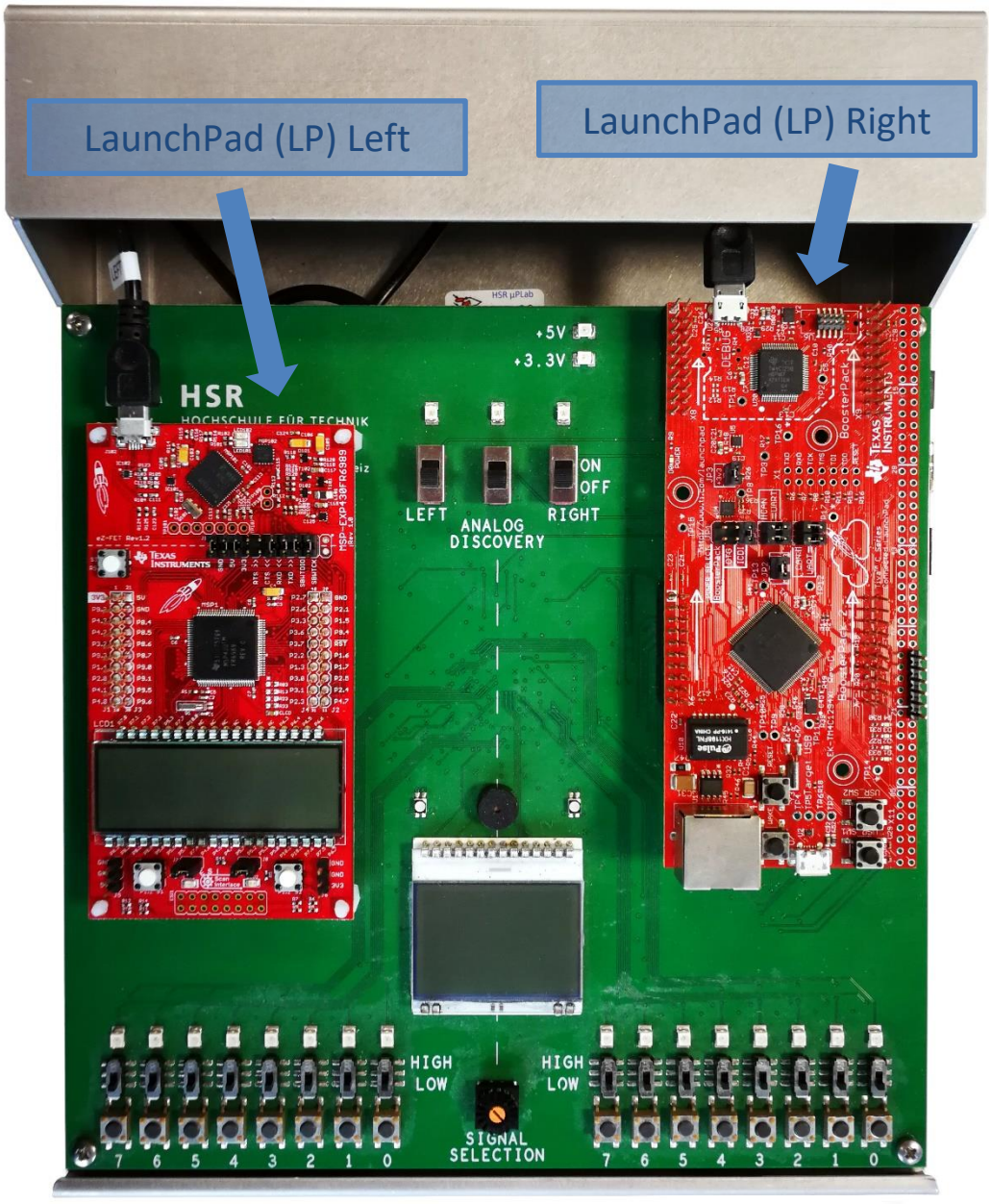
Inhaltsverzeichnis



1. Systembeschreibung

1.1. Bezeichnungen

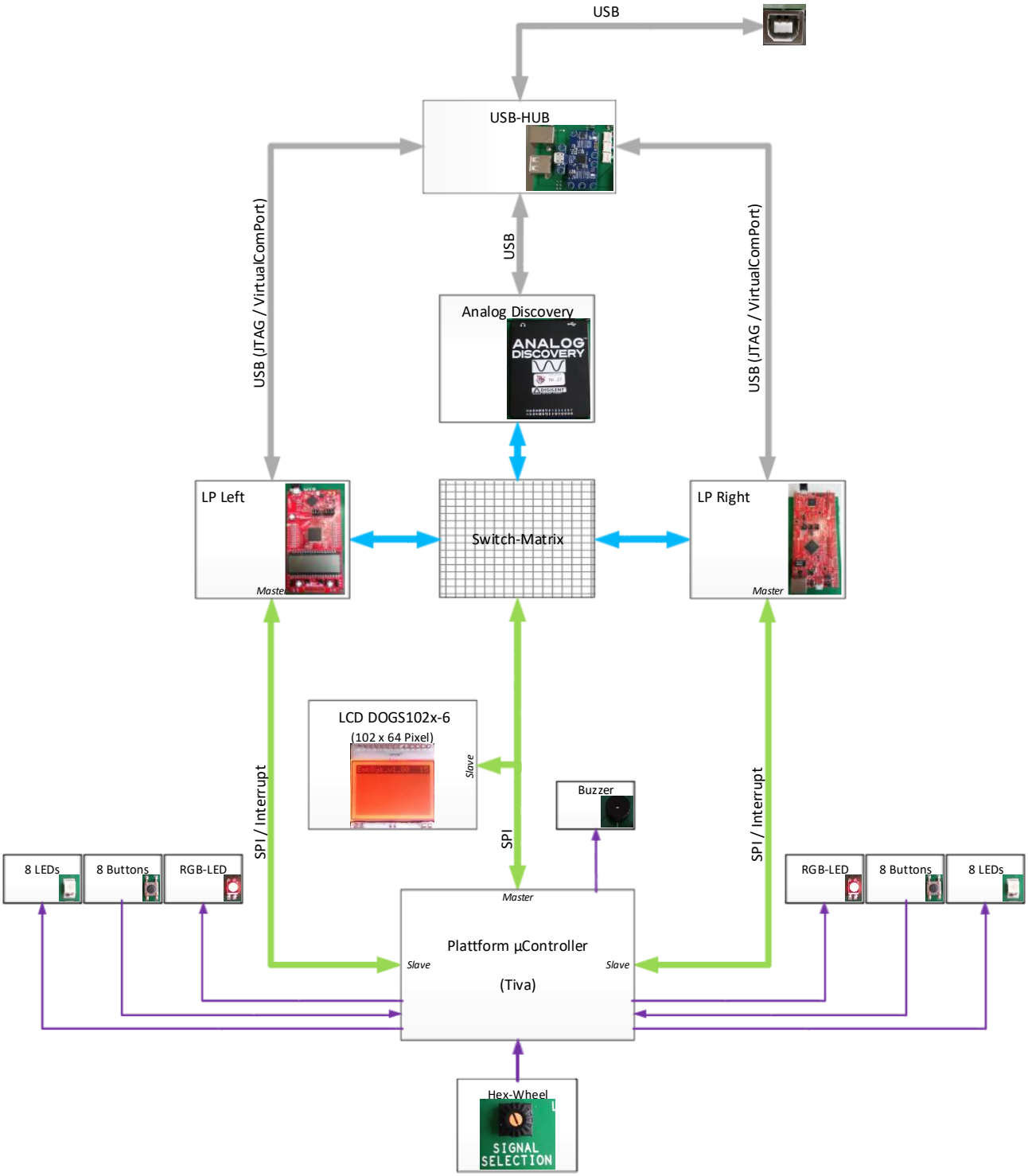
ESP	Embedded System Platform
LP	LaunchPad
AD	AnalogDiscovery



15 - 8 7 - 0

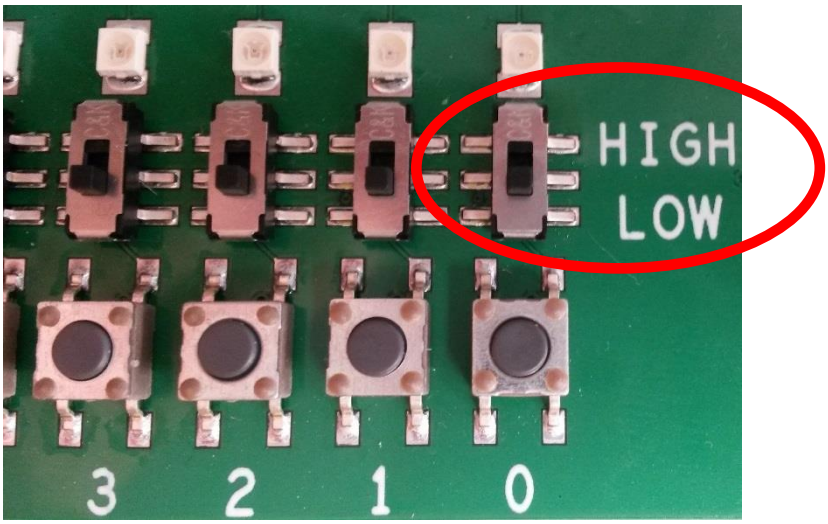
LED / Button / Switches

1.2. Blockschaltbild

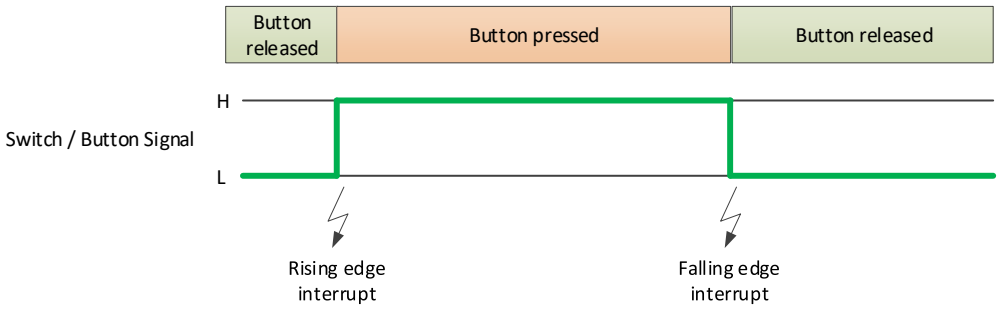


1.3. Details zu den Button / Switches

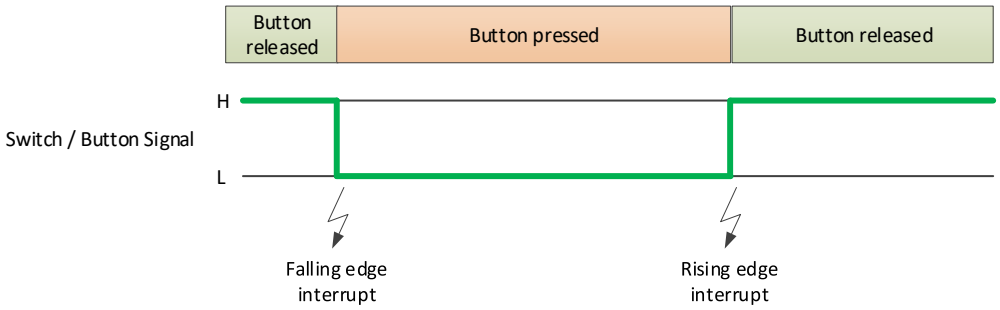
Mithilfe der Schiebeschalter (Switches) kann der Signalruhepegel vorgegeben werden. Die untenstehenden Signalverlaufsbilder stellen die beiden Varianten dar.



Switch-Position: LOW



Switch-Position: HIGH



2. ESP API's

2.1. MSP430FR6989

```
#include "ESP_MSPEXP430FR6989.h"          //MSP430FR6989 Embedded System Platform API
```

2.2. EKTM4C1294XL

```
#include "ESP_EKTM4C1294XL.h"             //EKTM4C1294XL Embedded System Platform API
```

2.3. Header

uint32_t ***esp_init*** *(void);*
uint32_t ***esp_sysCtlClockGet*** *(void);*

bool ***esp_buzzerEna*** *(bool enable);*
void ***esp_buzzerFreq*** *(uint16_t frequency);*

void ***esp_lcdClr*** *(void);*
void ***esp_lcdDrawVLine*** *(uint8_t x0, uint8_t y0, uint8_t h, bool opaque);*
void ***esp_lcdDrawHLine*** *(uint8_t x0, uint8_t y0, uint8_t w, bool opaque);*
void ***esp_lcdDrawRect*** *(uint8_t x0, uint8_t y0, uint8_t w, uint8_t h, bool opaque);*
void ***esp_lcdFillRect*** *(uint8_t x0, uint8_t y0, uint8_t w, uint8_t h, bool opaque);*
uint8_t ***esp_lcdTxt*** *(uint8_t pos, uint8_t line, char* txt);*
void ***esp_lcdBLight*** *(uint8_t red, uint8_t green);*

void ***esp_dbgOutInit*** *(uint8_t number, bool updateLED);*
void ***esp_dbgOutSet*** *(uint8_t number, bool updateLED);*
void ***esp_dbgOutClear*** *(uint8_t number, bool updateLED);*
void ***esp_dbgOutToggle*** *(uint8_t number, bool updateLED);*

uint16_t ***esp_bt*** *(void);*
void ***esp_btIntSetup****(uint8_t intNr, esIntCfg_t config, pFctHandler pCallbackFct);*
void ***esp_btIntEna*** *(uint16_t mask, uint16_t ena);*

void ***esp_led*** *(uint16_t mask, uint16_t value);*
void ***esp_rgbLed*** *(uint8_t red, uint8_t green, uint8_t blue, bool leftSide);*

2.4. Beschreibungen

SYSTEM

uint32_t ***esp_init*** (void);
Wrapper function to initialize the embedded system platform.
param -
return Version (Hex Coded) of this library. eg. 0x01012010 --> V01.012.010

uint32_t ***esp_sysCtlClockGet*** (void);
Wrapper function to get the current system-clock in Hz.
param -
return Current system-clock in Hz.

BUZZER

bool ***esp_buzzerEna*** (bool enable);
Enable the buzzer of the embedded system platform.
param enable If true, then enable, otherwise disable.
return True if the buzzer is enabled, which also depends to the powerUp-Check
Hinweis:
Damit der Buzzer überhaupt aktiviert wird, muss vor und während dem PowerUp der ESP-Hardware der Taster8 und der Taster7 gedrückt werden. Nachdem die Stromversorgung der ESP-Hardware unterbrochen wurde, muss dies beim nächsten PowerUp erneut gemacht werden!

void ***esp_buzzerFreq*** (uint16_t frequency);
Set the frequency of the buzzer in Hz.
param frequency The to be set frequency in Hz.
return -

LCD

void ***esp_lcdClr***(void);
Clear the complete LCD of the embedded system platform.
param -
return -

void ***esp_lcdDrawVLine*** (uint8_t x0, uint8_t y0, uint8_t h, bool opaque);
Paint a vertical line on a specific position with a defined length on the platform LCD.
param x0 x-Position (0..101)
param y0 y-Position (0..63)
param h height (1..64)
param opaque If true: vertical line is painted, otherwise cleared
return -

void ***esp_lcdDrawHLine*** (uint8_t x0, uint8_t y0, uint8_t w, bool opaque);
Paint a horizontal line on a specific position with a defined width on the platform LCD.
param x0 x-Position (0..101)
param y0 y-Position (0..63)
param w width (1..102)
param opaque If true: horizontal line is painted, otherwise cleared
return -

void ***esp_lcdDrawRect*** (uint8_t x0, uint8_t y0, uint8_t w, uint8_t h, bool opaque);
Paint a rectangle (only outlines) on a specific position on the platform LCD.
param x0 x-Position (0..101)
param y0 y-Position (0..63)
param w width (1..102)
param h height (1..64)
param opaque If true: outlines are painted, otherwise outlines are cleared
return -

LCD

void ***esp_lcdFillRect*** (uint8_t x0, uint8_t y0, uint8_t w, uint8_t h, bool opaque);
Paint a filled rectangle on a specific position on the platform LCD.
param x0 x-Position (0..101)
param y0 y-Position (0..63)
param w width (1..102)
param h height (1..64)
param opaque If true: rectangle is filled, otherwise area is cleared
return -

uint8_t ***esp_lcdTxt*** (uint8_t pos, uint8_t line, char* txt);
Write a text on a single line on the embedded system platform LCD.
note Text is truncated, if not possible to completely display on the selected line.
param pos Position of the begining (x) 0..15 (left to right)
param line Position of the begining (y) 0..5 (top to down)
param txt Pointer to the to be displayed text.
return The number of characters that are written on the LCD.

void ***esp_lcdBLight*** (uint8_t red, uint8_t green);
Set the LCD-Backlight color.
param red The duty of the pwm for the red Backlight
param green The duty of the pwm for the green Backlight
return -

DEBUG

void ***esp_dbgOutInit*** (uint8_t number, bool updateLED);
Init one of the possible digital output-signals used for debug purpose and set value to zero.
By selecting/using correct signal setting on ESP, signals can be measured by Analog Discovery unit.
note For left LaunchPad select setting 4
note For right LaunchPad select setting 6
param number A number between 0 and 15
param updateLED If true, also clear the ESP-LED (over SPI)
return -

void ***esp_dbgOutSet*** (uint8_t number, bool updateLED);
Set one of the possible digital output-signals used for debug purpose.
param number A number between 0 and 15
param updateLED If true, also set the ESP-LED (over SPI)
return -

void ***esp_dbgOutClear*** (uint8_t number, bool updateLED);
Clear one of the possible digital output-signals used for debug purpose.
param number A number between 0 and 15
param updateLED If true, also update the ESP-LED (over SPI)
return -

void ***esp_dbgOutToggle*** (uint8_t number, bool updateLED);
Toggle one of the possible digital output-signals used for debug purpose.
param number A number between 0 and 15
param updateLED If true, also update the ESP-LED (over SPI)
return -

BUTTON / SWITCHES

uint16_t ***esp_bt*** (void);
Get the value of all buttons from the embedded system platform.
param -
return The bitwise value of all buttons.

void ***esp_btIntSetup*** (uint8_t intNr, espIntCfg_t config, pEsespLib_pFctHandler pCallbackFct);
Configure a single button interrupt.
param intNr A number between 0 and 15
param config Configuration of interrupt-event (rising, falling or both edges)
param pCallbackFct Pointer to the callback-function if interrupt occurred
return -

BUTTON / SWITCHES	void ***esp_btIntEna*** (uint16_t mask, uint16_t ena); Enable interrupts on the left Buttons. Only the masked interrupt-bits are changed. param mask The to be changed interrupts (bitwise) param ena Bitwise enable value (a one means enabled, otherwise disabled) return -
LED	void ***esp_led*** (uint16_t mask, uint16_t value); Set the value of the complete led-row on the embedded system platform. param mask The to be updated led (bitwise-mask) param value The value to be set on the led (depending on mask)! return -
	void ***esp_rgbLed*** (uint8_t red, uint8_t green, uint8_t blue, bool leftSide); Set one of the RGB-LEDs with a specific color setting. param red The duty of the pwm for the red-led param green The duty of the pwm for the green-led param blue The duty of the pwm for the blue-led param leftSide If true, the RGB-LED on the left side is set, otherwise right return -

3. MSP430FR6989 API's

3.1. LCD-API

```
#include "LP_LCD_MSPEXP430FR6989.h"          //MSP430FR6989 Launchpad-LCD API
```

3.1.1. Header

uint32_t ***lp_lcdInit*** *(void);*

void ***lp_lcdClrAll*** *(void);*

void ***lp_lcdClrTxt*** *(void);*

void ***lp_lcdSpecSymb*** *(lcdSymb_t symb, bool_t on);*

void ***lp_lcdTxtWrite*** *(char* txt);*

void ***lp_lcdHex8Bit*** *(uint8_t value);*

void ***lp_lcdHex16Bit*** *(uint16_t value);*

void ***lp_lcdInt*** *(int value);*

void ***lp_lcdFloat*** *(float value);*

void ***lp_lcdTime*** *(uint8_t hour, uint8_t min, uint8_t sec);*

3.1.2. Beschreibungen

LCD

uint32_t ***lp_lcdInit*** (void) ;
Wrapper function to initialize the LCD of the MSP-EXP430FR6989 Launchpad.
param -
return Version (Hex Coded) of this library. e.g. 0x01012010 --> V01.012.010

void ***lp_lcdClrAll*** (void) ;
Clear the complete LCD of the MSP-EXP430FR6989 Launchpad.
param -
return -

void ***lp_lcdClrTxt*** (void);
Clears only the text area of the LCD of the MSP-EXP430FR6989 Launchpad.
param -
return -

void ***lp_lcdSpecSymb*** (lcdSymb_t symb, bool_t on);
Turn ON/OFF a specific symbol of the LCD of the MSP-EXP430FR6989 Launchpad.
param symb One of the possible symbols from enumeration
param on If true, then display it, otherwise clear it.
return -

void ***lp_lcdTxtWrite*** (char* txt);
Write a short text to the display.
param txt Pointer to the to be displayed text.
return -

void ***lp_lcdHex8Bit*** (uint8_t value);
Write a 8 Bit Hex Value to the middle of the LCD (2 Character)
param value 8Bit hexadecimal value
return -

void ***lp_lcdHex16Bit*** (uint16_t value);
Write a 16 Bit Hex Value to the middle of the LCD (4 Character)
param value 16Bit hexadecimal value
return -

void ***lp_lcdInt*** (int value);
Write a 16bit integer value to the LCD
param value 16bit integer value.
return -

void ***lp_lcdFloat*** (float value);
Write a float value to the LCD.
note If the value is out of the to be displayed range, the Note-Symbol (!) is activated on the display.
param value The float value to be displayed.
return -

void ***lp_lcdTime*** *(uint8_t hour, uint8_t min, uint8_t sec);*
Write a time information to the LCD.
param hour to be displayed
param min to be displayed
param sec to be displayed
return -