Anlogic Digital Conversor

Generated by Doxygen 1.9.1

1 Class Index		1
1.1 Class List	 	1
2 File Index		3
2.1 File List	 	3
3 Class Documentation		5
3.1 adc_chan Struct Reference	 	5
3.1.1 Detailed Description	 	6
3.2 adc_ctrl_t Struct Reference	 	6
3.2.1 Detailed Description	 	6
3.3 adc_seqX_ctrl Struct Reference	 	7
3.3.1 Detailed Description	 	7
3.4 adc_seqX_gdat Struct Reference	 	7
3.4.1 Detailed Description	 	8
3.5 adc_thr Struct Reference	 	8
3.5.1 Detailed Description	 	8
4 File Documentation		9
4.1 inc/ADC_FW.h File Reference	 	9
4.1.1 Detailed Description	 	11
4.1.2 Function Documentation	 	11
4.1.2.1 ADC_Disable()	 	11
4.1.2.2 ADC_Enable()	 	12
4.1.2.3 ADC_Init()	 	12
4.1.2.4 ADC_Power()	 	14
4.2 inc/Aplication.h File Reference	 	15
4.2.1 Detailed Description	 	15
4.2.2 Function Documentation	 	15
4.2.2.1 GPIO_Init()	 	16
4.2.2.2 LPC_Init()	 	16
4.3 inc/Disp7Seg_FW.h File Reference	 	17
4.3.1 Detailed Description	 	18
4.3.2 Function Documentation	 	18
4.3.2.1 DISP7SEG_Init()	 	18
4.3.2.2 DISP_Sweep()	 	19
4.4 inc/Disp7Seg_SW.h File Reference	 	20
4.4.1 Detailed Description	 	20
4.4.2 Function Documentation	 	21
4.4.2.1 Display()	 	21
4.5 inc/GPIO_FW.h File Reference	 	21
4.5.1 Detailed Description	 	24
4.5.2 Function Documentation	 	24

4.5.2.1 GetOFFSET()	 24
4.5.2.2 GPIO_ClearOUT()	 25
4.5.2.3 GPIO_Debounce()	 25
4.5.2.4 GPIO_DebounceUserKEY()	 26
4.5.2.5 GPIO_Disable()	 27
4.5.2.6 GPIO_Enable()	 27
4.5.2.7 GPIO_GetPIN()	 28
4.5.2.8 GPIO_SetDIR()	 29
4.5.2.9 GPIO_SetModeCLKDIV()	 29
4.5.2.10 GPIO_SetModeDAC()	 30
4.5.2.11 GPIO_SetModeFILTER()	 31
4.5.2.12 GPIO_SetModeHYS()	 32
4.5.2.13 GPIO_SetModel2C()	 32
4.5.2.14 GPIO_SetModeINPUT()	 33
4.5.2.15 GPIO_SetModeINV()	 34
4.5.2.16 GPIO_SetModeOD()	 34
4.5.2.17 GPIO_SetOUT()	 35
4.5.2.18 GPIO_SetPIN()	 36
4.5.2.19 GPIO_ToogleOUT()	 36
4.5.2.20 IOCONDisable()	 37
4.5.2.21 IOCONEnable()	 37
4.6 inc/GPIO_SW.h File Reference	 38
4.6.1 Detailed Description	 38
4.6.2 Function Documentation	 38
4.6.2.1 GetInput()	 39
4.6.2.2 GetUserKEY()	 39
4.7 inc/LPC845.h File Reference	 40
4.7.1 Detailed Description	 40
4.8 inc/SwitchMatrix_FW.h File Reference	 41
4.8.1 Detailed Description	 42
4.8.2 Enumeration Type Documentation	 42
4.8.2.1 anonymous enum	 42
4.8.2.2 anonymous enum	 42
4.8.2.3 anonymous enum	 43
4.8.2.4 anonymous enum	 43
4.8.3 Function Documentation	 44
4.8.3.1 SWM()	 44
4.8.3.2 SWM_Disable()	 45
4.8.3.3 SWM_Enable()	 45
4.8.3.4 SWM_PinEnable()	 46
4.9 inc/SYSCON_FW.h File Reference	 46
4.9.1 Detailed Description	 49

4.9.2 Function Documentation	49
4.9.2.1 BoardClockRUN()	49
4.10 inc/SysTick_FW.h File Reference	50
4.10.1 Detailed Description	51
4.10.2 Function Documentation	51
4.10.2.1 SysTick_Init()	51
4.10.2.2 SysTick_Off()	52
4.10.2.3 SysTick_Set()	52
4.11 source/05-ConversionADC.c File Reference	53
4.11.1 Detailed Description	53
4.11.2 Function Documentation	54
4.11.2.1 main()	54
4.11.3 Variable Documentation	54
4.11.3.1 tick	55
4.12 source/ADC_FW.c File Reference	55
4.12.1 Detailed Description	55
4.12.2 Function Documentation	56
4.12.2.1 ADC0_SEQA_IRQHandler()	56
4.12.2.2 ADC_Disable()	57
4.12.2.3 ADC_Enable()	57
4.12.2.4 ADC_Init()	58
4.12.2.5 ADC_Power()	59
4.13 source/Aplication.c File Reference	
4.13.1 Detailed Description	60
4.13.2 Function Documentation	61
4.13.2.1 GPIO_Init()	61
4.13.2.2 LPC_Init()	61
4.13.3 Variable Documentation	62
4.13.3.1 tick	62
4.14 source/Disp7Seg_FW.c File Reference	62
4.14.1 Detailed Description	63
4.14.2 Function Documentation	63
4.14.2.1 DISP7SEG_Init()	63
4.14.2.2 DISP_Sweep()	64
4.15 source/Disp7Seg_SW.c File Reference	65
4.15.1 Detailed Description	65
4.15.2 Function Documentation	66
4.15.2.1 Display()	66
4.15.3 Variable Documentation	66
4.15.3.1 Digits_to_BCD7seg	67
4.16 source/GPIO_FW.c File Reference	67
4.16.1 Detailed Description	68

4.16.2 Function Documentation	68
4.16.2.1 GetOFFSET()	68
4.16.2.2 GPIO_ClearOUT()	69
4.16.2.3 GPIO_Debounce()	70
4.16.2.4 GPIO_DebounceUserKEY()	70
4.16.2.5 GPIO_Disable()	71
4.16.2.6 GPIO_Enable()	72
4.16.2.7 GPIO_GetPIN()	72
4.16.2.8 GPIO_SetDIR()	73
4.16.2.9 GPIO_SetModeCLKDIV()	74
4.16.2.10 GPIO_SetModeDAC()	74
4.16.2.11 GPIO_SetModeFILTER()	75
4.16.2.12 GPIO_SetModeHYS()	76
4.16.2.13 GPIO_SetModel2C()	76
4.16.2.14 GPIO_SetModeINPUT()	77
4.16.2.15 GPIO_SetModeINV()	78
4.16.2.16 GPIO_SetModeOD()	79
4.16.2.17 GPIO_SetOUT()	79
4.16.2.18 GPIO_SetPIN()	80
4.16.2.19 GPIO_ToogleOUT()	81
4.16.2.20 IOCONDisable()	81
4.16.2.21 IOCONEnable()	82
4.16.3 Variable Documentation	82
4.16.3.1 offset	83
4.17 source/GPIO_SW.c File Reference	83
4.17.1 Detailed Description	83
4.17.2 Function Documentation	83
4.17.2.1 GetInput()	84
4.17.2.2 GetUserKEY()	84
4.18 source/mtb.c File Reference	85
4.18.1 Detailed Description	85
4.19 source/SwitchMatrix_FW.c File Reference	86
4.19.1 Detailed Description	86
4.19.2 Function Documentation	86
4.19.2.1 SWM()	86
4.19.2.2 SWM_Disable()	87
4.19.2.3 SWM_Enable()	88
4.19.2.4 SWM_PinEnable()	88
4.20 source/SYSCON_FW.c File Reference	89
4.20.1 Detailed Description	89
4.20.2 Function Documentation	90
4.20.2.1 BoardClockRUN()	90

erence																					. 9	0
																					. 9	1
n																					. 9	1
idler()																					. 9	1
)																					. 9	2
)																					. 9	3
)																					. 9	3
n																					. 9	4
																					. 9	4
																					a	5
(on	erence 9																				

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

adc_chan	5
$adc_ctrl_t \ \ldots \ldots \ldots \ldots \ldots \ldots \ldots$	6
adc_seqX_ctrl	7
adc_seqX_gdat	7
ade thr	۵

2 Class Index

Chapter 2

File Index

2.1 File List

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

inc/ADC_FW.h	
: Firmware functions ADC	9
inc/Aplication.h	
: Functions used in main	15
inc/Disp7Seg_FW.h	
: Firmware functions for DISP7SEG	17
inc/Disp7Seg_SW.h	
: Software functions for DISP7SEG	20
inc/GPIO_FW.h	
: Firmware functions for GPIO	21
inc/GPIO_SW.h	
: Software functions for GPIO	38
inc/LPC845.h	
: Declarations for type of data	40
inc/SwitchMatrix_FW.h	
: Firmware functions for SWM	41
inc/SYSCON_FW.h	
: Firmware functions for SYSCON	46
inc/SysTick_FW.h	
: Firmware functions for SysTick	50
source/05-ConversionADC.c	
: Firmware functions ADC	53
source/ADC_FW.c	
: Firmware functions ADC	55
source/Aplication.c : Functions used in main	00
	60
source/Disp7Seg_FW.c : Firmware functions for DISP7SEG	00
	62
source/Disp7Seg_SW.c : Software functions for DISP7SEG	G E
source/GPIO FW.c	65
: Firmware functions for GPIO	67
source/GPIO SW.c	07
: Software functions for GPIO	83
source/mtb.c	03
MTB initialization file	85
WITE HAUGHZUUTT HIC	U.

4 File Index

source/semihost_hardfault.c	??
source/SwitchMatrix_FW.c	
: Firmware functions for SWM	86
source/SYSCON_FW.c	
: Firmware functions for SYSCON	89
source/SysTick_FW.c	
: Firmware functions for SysTick	90

Chapter 3

Class Documentation

3.1 adc_chan Struct Reference

Public Attributes

```
union {
   _RW uint32_t _CHAN_THRSEL
 struct {
      RW uint32_t _CH0_THRSEL:1
      Compare against THR.
      RW uint32_t _CH1_THRSEL:1
      Compare against THR.
      _RW uint32_t _CH2_THRSEL:1
      Compare against THR.
      _RW uint32_t _CH3_THRSEL:1
      Compare against THR.
      _RW uint32_t _CH4_THRSEL:1
      Compare against THR.
      RW uint32 t CH5 THRSEL:1
      Compare against THR.
      RW uint32_t _CH6_THRSEL:1
      Compare against THR.
      _RW uint32_t _CH7_THRSEL:1
      Compare against THR.
      _RW uint32_t _CH8_THRSEL:1
      Compare against THR.
      RW uint32_t _CH9_THRSEL:1
      Compare against THR.
      RW uint32_t _CH10_THRSEL:1
      Compare against THR.
      RW uint32 t CH11 THRSEL:1
      Compare against THR.
      _RW uint32_t _RESERVED:20
};
```

6 Class Documentation

3.1.1 Detailed Description

Definition at line 145 of file ADC_FW.h.

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

• inc/ADC_FW.h

3.2 adc ctrl t Struct Reference

Public Attributes

```
union {
  __RW uint32_t _CTRL
    < Union between CTRL and bit field; They're overlapped
    __RW uint32_t _CLKDIV:8
      Clock divided by this + 1 to produce sampling clock <= 30MHz.
     _RW uint32_t _ASYNCMODE:1
      Asyncronous operation mode.
    __RW uint32_t _RESERVED_0:1
    __RW uint32_t _LPWRMODE:1
      Power down ADC while is not used.
    __RW uint32_t _RESERVED_1:19
     _RW uint32_t _CALMODE:1
      Self calibration.
     _RW uint32_t _RESERVED_2:1
 }
};
```

3.2.1 Detailed Description

< Struct for handling adc configuration

Definition at line 56 of file ADC_FW.h.

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

• inc/ADC FW.h

3.3 adc seqX ctrl Struct Reference

Public Attributes

```
union {
    _RW uint32_t _SEQx_CTRL
  struct {
    __RW uint32_t _CHANNELS:12
      Select which channel will be sampled.
    RW uint32 t TRIGGER:3
      Select which HW trigger will start convertion.
      RW uint32_t _RESERVED_0:3
      RW uint32_t _TRIGPOL:1
      Polarity of the input trigger.
      _RW uint32_t _SYNCBYPASS:1
      Byspass syncronization FF, so is slower.
      RW uint32_t _TSAMP:5
      RW uint32 t RESERVED 1:1
      _RW uint32_t _START:1
      Launch one pass.
      RW uint32 t BURST:1
      Sequence continuosly converted.
      _RW uint32_t _SINGLESTEP:1
      When start in 1 this converts only the next channel.
      RW uint32 t LOWPRIO:1
      Set priority for sequence A.
      _RW uint32_t _MODE:1
      Read global data or individual channel.
      RW uint32_t _SEQx_ENA:1
      Enable sequence.
};
```

3.3.1 Detailed Description

Definition at line 74 of file ADC_FW.h.

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

· inc/ADC FW.h

3.4 adc_seqX_gdat Struct Reference

Public Attributes

8 Class Documentation

```
union {
   __RW uint32_t _SEQx_GDAT
  struct {
    __RW uint32_t _RESERVED_0:4
    __RW uint32_t _RESULT:12
      12 bit A/D convertion
      _RW uint32_t _THCMPRANGE:2
      Compare the result with thrn_low and thrn_high.
      RW uint32 t THCMPCROSS:2
      Indicates a crossing of the threshold.
      _RW uint32_t _RESERVED_1:6
    __RW uint32_t _CHN:4
      Indicates the channel converted.
      RW uint32 t OVERRUN:1
      If a new convertion was loaded and the previous was not read.
      _RW uint32_t _DATAVALID:1
      There's a new result.
 }
};
```

3.4.1 Detailed Description

Definition at line 111 of file ADC_FW.h.

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

• inc/ADC_FW.h

3.5 adc thr Struct Reference

Public Attributes

```
union {
    __RW uint32_t _THRn_LH
    struct {
     __RW uint32_t _RESERVED_0:4
     __RW uint32_t _THR:12
     _12bits for compare
    __RW uint32_t _RESERVED_1:16
    }
};
```

3.5.1 Detailed Description

Definition at line 132 of file ADC_FW.h.

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

• inc/ADC_FW.h

Chapter 4

File Documentation

inc/ADC FW.h File Reference 4.1

: Firmware functions ADC

Classes

- struct adc_ctrl_t
- struct adc_seqX_ctrl
- struct adc_seqX_gdat
- struct adc_thr
- struct adc_chan

Macros

- #define MASK_ADC_SYSCON 4
- #define ADC_SYSAHB 24
- #define ADC_0 PORT0,7,14

POT1 on board; 14 is the bit in PINENABLE.

- #define **ADC_ADD** ((__RW uint32_t *) 0x4001C000UL)
- #define ADC CTRL ADC ADD[0]
- #define _ADC_SEQA_CTRL ADC_ADD[2]
- #define _ADC_SEQB_CTRL ADC_ADD[3]
- #define ADC SEQA GDAT ADC ADD[4]
- #define _ADC_SEQB_GDAT ADC_ADD[5]
- #define _ADC_DAT0 ADC_ADD[8]
- #define ADC DAT1 ADC ADD[9]
- #define ADC DAT2 ADC ADD[10]
- #define _ADC_DAT3 ADC_ADD[11]
- #define _ADC_DAT4 ADC_ADD[12]
- #define _ADC_DAT5 ADC_ADD[13]
- #define _ADC_DAT6 ADC_ADD[14]
- #define _ADC_DAT7 ADC_ADD[15]
- #define _ADC_DAT8 ADC_ADD[16]
- #define _ADC_DAT9 ADC_ADD[17]
- #define _ADC_DAT10 ADC_ADD[18]

```
    #define _ADC_DAT11 ADC_ADD[19]

    #define _ADC_THR0_LOW ADC_ADD[20]

    #define ADC THR1 LOW ADC ADD[21]

• #define ADC THR0 HIGH ADC ADD[22]

    #define _ADC_THR1_HIGH_ ADC_ADD[23]

    #define _ADC_CHAN_THRSEL ADC_ADD[24]

    #define ADC_INTEN ADC_ADD[25]

    #define ADC_FLAGS ADC ADD[26]

• #define ADC TRM ADC ADD[27]

    #define ADC CTRL ( ( RW adc ctrl t *) 0x4001C000UL)

     Pointer to a struct in that memory.
#define ADC_SEQA_CTRL ( (__RW adc_seqX_ctrl *) 0x4001C008UL)
     Pointer to a struct in that memory.

    #define ADC SEQB CTRL ( ( RW adc seqX ctrl *) 0x4001C00CUL)

     Pointer to a struct in that memory.

    #define ADC SEQA GDAT ( ( RW adc seqX gdat *) 0x4001C010UL)

     Pointer to a struct in that memory.

    #define ADC SEQB GDAT (( RW adc seqX gdat *) 0x4001C014UL)

     Pointer to a struct in that memory.
#define ADC_DAT0 ( ( __RW adc_seqX_gdat *) 0x4001C020UL)
     Pointer to a struct in that memory.
#define ADC_DAT1 ( (__RW adc_seqX_gdat *) 0x4001C024UL)
     Pointer to a struct in that memory.

    #define ADC_DAT2 ( ( __RW adc_seqX_gdat *) 0x4001C028UL)

     Pointer to a struct in that memory.
#define ADC_DAT3 ( ( __RW adc_seqX_gdat *) 0x4001C02CUL)
     Pointer to a struct in that memory.

    #define ADC DAT4 (( RW adc seqX gdat *) 0x4001C030UL)

     Pointer to a struct in that memory.
#define ADC_DAT5 ( ( __RW adc_seqX_gdat *) 0x4001C034UL)
     Pointer to a struct in that memory.

    #define ADC_DAT6 ( ( __RW adc_seqX_gdat *) 0x4001C038UL)

     Pointer to a struct in that memory.
#define ADC_DAT7 ( ( __RW adc_seqX_gdat *) 0x4001C03CUL)
     Pointer to a struct in that memory.
#define ADC_DAT8 ( ( __RW adc_seqX_gdat *) 0x4001C040UL)
     Pointer to a struct in that memory.

    #define ADC_DAT9 ( ( __RW adc_seqX_gdat *) 0x4001C044UL)

     Pointer to a struct in that memory.

    #define ADC_DAT10 ( (__RW adc_seqX_gdat *) 0x4001C048UL)

     Pointer to a struct in that memory.
#define ADC_DAT11 ( ( __RW adc_seqX gdat *) 0x4001C04CUL)
     Pointer to a struct in that memory.

    #define ADC_THR0_LOW ( ( __RW adc_thr *) 0x4001C050UL)

     Pointer to a struct in that memory.

    #define ADC THR1 LOW (( RW adc thr *) 0x4001C054UL)

     Pointer to a struct in that memory.

    #define ADC_THR0_HIGH ( ( __RW adc_thr *) 0x4001C058UL)

     Pointer to a struct in that memory.

    #define ADC THR1 HIGH (( RW adc thr *) 0x4001C05CUL)
```

Pointer to a struct in that memory.

```
    #define ADC_CHAN_THRSEL ( ( __RW adc_chan *) 0x4001C060UL)
        Pointer to a struct in that memory.
    #define MASK_SEQA_INTEN 1<<0
        Interrupt after each conv.</li>
    #define MASK_SEQB_INTEN 1<<1
        Interrupt after each conv.</li>
    #define MASK_ISE_ADC_SEQA 1<<16
        Enable Interrupt NVIC.</li>
    #define MASK_ISE_ADC_SEQB 1<<17</li>
```

Enable Interrupt NVIC.

Functions

```
    void ADC_Init (uint8_t port, uint8_t pin, uint8_t ena)
        : Initialize ADC on a pin
        void ADC_Power (void)
            : Power ADC

    void ADC_Enable (void)
            : Enable clock in ADC

    void ADC_Disable (void)
            : Disable clock in ADC
```

4.1.1 Detailed Description

```
: Firmware functions ADC
```

: 12 bits convertion

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

4.1.2 Function Documentation

4.1.2.1 ADC_Disable()

```
void ADC_Disable (
void )

: Disable clock in ADC

:
Author

: Tobias Bavasso Piizzi

Date
```

: 08/01/2021

Parameters

[in] void

Returns

: void

Definition at line 90 of file ADC_FW.c.

4.1.2.2 ADC_Enable()

```
void ADC_Enable (
    void )
```

: Enable clock in ADC

:

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

[in] void

Returns

: void

Definition at line 77 of file ADC_FW.c.

```
77 {
78 SYSAHBCLKCTRL0|= (1«ADC_SYSAHB);
79 }
```

4.1.2.3 ADC_Init()

```
uint8_t pin,
uint8_t ena )
```

: Initialize ADC on a pin

: Continuos conversion of POTE in board

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

	[in] uint8_t port: PORT0,PORT1	
	[in] uint8_t pin: 0,31	
Ì	[in] uint8 t en: bit to enable in PINENABLE (page 143	UM)

Returns

: void

- < Enable CLOCK in SYSAHB
- < Enable service interrupt
- < Interrupt after conversion finish
- < Enable Switch Matrix
- < Enable pin in SWN as AnalogInput
- < Disable Switch Matrix
- < Power in SYSCON
- < Div = 0
- < Sync
- < OFF
- < OFF
- < Sample CH0
- < No hardware trigger
- < Positive trigger
- < Enable sync

< Individual end of conversion

< Start, enable set on the same line first time

```
Definition at line 23 of file ADC_FW.c.
24
           ADC_Enable();
ISER0|= MASK_ISE_ADC_SEQA;
ADC_INTEN|= MASK_SEQA_INTEN;
25
26
27
28
           SWM_Enable();
29
           SWM_PinEnable(port, pin, ena);
           SWM_Disable();
30
           ADC_Power();
31
32
33
34
           ADC_CTRL->_CLKDIV = 0x00;
ADC_CTRL->_ASYNCMODE = 0;
ADC_CTRL->_LPWRMODE = 0;
ADC_CTRL->_CALMODE = 0;
35
36
37
38
39
40
           ADC_SEQA_CTRL->_CHANNELS
                                                               = 0x01;
           ADC_SEQA_CTRL->_TRIGGER
ADC_SEQA_CTRL->_TRIGGER
ADC_SEQA_CTRL->_TRIGFOL
ADC_SEQA_CTRL->_SYNCBYPASS
ADC_SEQA_CTRL->_TSAMP
41
                                                               = 0x00;
42
                                                               = 0x1;
                                                                 = 0x0;
43
44
                                                               = 0x00;
           ADC_SEQA_CTRL->_START
                                                               = 0;
```

= 0; = 0x0;

= 0x0;

4.1.2.4 ADC_Power()

ADC_SEQA_CTRL->_BURST

ADC_SEQA_CTRL->_SINGLESTEP

ADC_SEQA_CTRL->_LOWPRIO = 0x0
ADC_SEQA_CTRL->_MODE = 0;
ADC_SEQA_CTRL->_SEQX_ENA = 0;
_ADC_SEQA_CTRL |= ((0b100001) « 26);

```
void ADC_Power (
            void )
: Power ADC
```

45 46

47

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

[in] void

Returns

: void

Definition at line 63 of file ADC_FW.c.

```
63 {
64 PDRUNCFG&= (~(1 « MASK_ADC_SYSCON));
65 66 }
```

4.2 inc/Aplication.h File Reference

: Functions used in main

```
#include "LPC845.h"
#include "GPIO_FW.h"
#include "GPIO_SW.h"
#include "SwitchMatrix_FW.h"
#include "SYSCON_FW.h"
#include "SysTick_FW.h"
#include "Disp7Seg_FW.h"
#include "Disp7Seg_SW.h"
#include "ADC_FW.h"
```

Functions

```
    void LPC_Init (void)

            Initialize the board

    void GPIO_Init (void)

            Initialize the GPIO
```

4.2.1 Detailed Description

```
: Functions used in main
:
Author
: Tobias Bavasso Piizzi
Date
```

: 04/01/2021

4.2.2 Function Documentation

4.2.2.1 GPIO_Init()

```
void GPIO_Init (
     void )
```

: Initialize the GPIO

: It depends on each proyect

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 35 of file Aplication.c.

```
35 GPIO_SetDIR(UserKEY, INPUT);
36 GPIO_SetDIR(LedGREEN, OUTPUT);
38 GPIO_SetDIR(LedBLUE, OUTPUT);
39
40 GPIO_SetPIN(LedGREEN, LED_OFF);
41 GPIO_SetPIN(LedBLUE, LED_OFF);
42 }
```

4.2.2.2 LPC_Init()

: Initialize the board

: It depends on each proyect

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] void
```

Returns

: void

Definition at line 19 of file Aplication.c.

4.3 inc/Disp7Seg_FW.h File Reference

: Firmware functions for DISP7SEG

Macros

```
    #define SEG_A PORT0,21
        Pin to connect segA.

    #define SEG_B PORT0,22
```

Pin to connect segB.

• #define SEG_C PORT0,16

Pin to connect segC.

#define SEG_D PORT0,17

Pin to connect segD.

• #define SEG_E PORT0,18

Pin to connect segE.

• #define SEG_F PORT0,20

Pin to connect segF.

• #define SEG_G PORT0,19

Pin to connect segG.

• #define SEG_DP PORT0,23

Pin to connect segDP.

• #define TR_D1 PORT0,0

Pin to connect transistor DISP1.

#define TR_D0 PORT0,1

Pin to connect transistor DISP0.

• #define DIGITS 2

Number of displays.

- #define DIGIT_0 0
- #define DIGIT_1 1

Functions

```
    void DISP7SEG_Init (void)

            Set pins for display as out

    void DISP_Sweep (void)

            Refresh the display 7Seg (2 Disp)
```

4.3.1 Detailed Description

```
: Firmware functions for DISP7SEG
:
Author
: Tobias Bavasso Piizzi
Date
```

4.3.2 Function Documentation

4.3.2.1 DISP7SEG_Init()

: 07/01/2021

```
void DISP7SEG_Init (
void )

: Set pins for display as out

:

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021
```

[in] void

Returns

: void

```
Definition at line 19 of file Disp7Seg_FW.c.
```

```
20
         GPIO_SetDIR(SEG_A, OUTPUT);
         GPIO_SetDIR(SEG_B, OUTPUT);
22
         GPIO_SetDIR(SEG_C, OUTPUT);
23
         GPIO_SetDIR(SEG_D, OUTPUT);
24
25
         GPIO_SetDIR(SEG_E, OUTPUT);
        GPIO_SetDIR(SEG_F, OUTPUT);
GPIO_SetDIR(SEG_G, OUTPUT);
GPIO_SetDIR(TR_D0, OUTPUT);
26
        GPIO_SetDIR(TR_D1, OUTPUT);
         GPIO_ClearOUT(SEG_A);
30
        GPIO_ClearOUT(SEG_B);
GPIO_ClearOUT(SEG_C);
GPIO_ClearOUT(SEG_D);
31
32
33
         GPIO_ClearOUT(SEG_E);
35
         GPIO_ClearOUT(SEG_F);
36
         GPIO_ClearOUT(SEG_G);
37
         GPIO_ClearOUT(TR_D0);
         GPIO_ClearOUT(TR_D1);
38
39 }
```

4.3.2.2 DISP_Sweep()

```
void DISP_Sweep (
     void )
```

: Refresh the display 7Seg (2 Disp)

: Is necessary to be used in SysTick_Handler

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

Parameters

```
[in] void
```

Returns

: void

- < Number of disp
- < Turn off transistor
- < Turn off transistor

- < Next time sweep other disp
- < Reset the digits

```
Definition at line 51 of file Disp7Seg_FW.c.
```

```
uint8_t aux;
       static uint8_t digit = 0;
54
       GPIO_ClearOUT(TR_D0);
GPIO_ClearOUT(TR_D1);
55
56
57
       aux = buff_Disp7[digit];
59
       60
       GPIO_SetPIN( SEG_B, ((aux » 1) & (uint8_t) 0x01));
61
       GPIO_SetPIN( SEG_C, ((aux » 2) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_D, ((aux » 3) & (uint8_t) 0x01));
62
63
       GPIO_SetPIN( SEG_E, ((aux » 4) & (uint8_t) 0x01));
       GPIO_SetPIN( SEG_F, ((aux » 5) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_G, ((aux » 6) & (uint8_t) 0x01));
       GPIO_SetPIN( SEG_DP, ((aux » 7) & (uint8_t) 0x01));
67
68
       switch (digit) {
69
       case DIGIT_0:
         GPIO_SetOUT(TR_D0);
73
       case DIGIT_1:
       GPIO_SetOUT(TR_D1);
74
75
           break:
76
       default:
       digit = 0;
           GPIO_SetOUT(TR_D0);
78
79
           break;
80
      }
81
       digit++;
82
       digit %= DIGITS;
84
85 }
```

4.4 inc/Disp7Seg_SW.h File Reference

: Software functions for DISP7SEG

Functions

```
void Display (uint8_t val): Writes on Disp7Seg
```

4.4.1 Detailed Description

: Software functions for DISP7SEG

:

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

4.4.2 Function Documentation

4.4.2.1 Display()

: Writes on Disp7Seg

: High lever of layers

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

Parameters

```
[in] uint8_t val: 0 to 99
```

Returns

: void

< Disable SysTick INT

< Enable SysTick INT

Definition at line 38 of file Disp7Seg_SW.c.

```
38
         uint8_t i;
uint8_t auxDisp[DIGITS];
39
         for (i = 0; i < DIGITS; i++) {
   auxDisp[i] = Digits_to_BCD7seg[val % 10];</pre>
43
44
               val /= 10;
4.5
         for (i = 0; i < DIGITS; i++) {</pre>
46
              SYSTICK_INT_DIS;
buff_Disp7[i] = auxDisp[i];
SYSTICK_INT_EN;
48
49
50
51
52 }
```

4.5 inc/GPIO_FW.h File Reference

: Firmware functions for GPIO

Macros

```
• #define PORT0 0
• #define PORT1 1
• #define LedGREEN PORT1, 0
     Led green in board.

    #define LedBLUE PORT1, 1

     Led blue in board.

    #define LedRED PORT1, 2

     Led red in board.

    #define UserKEY PORT0, 4

     Key in board.
• #define INPUT 0
• #define OUTPUT 1
• #define LOW 0

    #define HIGH 1

• #define ACT_HIGH 1
• #define ACT_LOW 0

    #define LED ON 0

     The led are active low.

    #define LED_OFF 1

     The led are active low.
• #define BOUNCE 10
     Times to check the bounce.

    #define SYSAHBCLKCTRL ( ( __RW uint32_t *) 0x40048080UL)

    #define SYSAHBCLKCTRL[0]

• #define SYSAHBCLKCTRL1 SYSAHBCLKCTRL[1]

    #define GPIO_PBYTE ( ( RW uint8 t *) 0xA000000UL)

    #define GPIO_PWORD ( ( __RW uint32_t *) 0xA0001000UL)

    #define GPIO_DIRP ( ( __RW uint32_t *) 0xA0002000UL)

    #define GPIO_PORT ( ( __RW uint32_t *) 0xA0002100UL)

    #define GPIO_SETP ( ( __RW uint32_t *) 0xA0002200UL)

    #define GPIO_CLRP ( ( __RW uint32_t *) 0xA0002280UL)

    #define GPIO_NOTP ( ( __RW uint32_t *) 0xA0002300UL)

    #define NO PULL UP DOWN 0x00

• #define PULL_DOWN 0x01
• #define PULL UP 0x02
• #define REPEATER 0x03

    #define HYS EN 0x01

    #define HYS DIS 0x00

    #define INV_INPUT 0x01

    #define NOT_INV_INPUT 0x00

    #define OD_EN 0x01

    #define OD DIS 0x00

• #define BYPASS FILTER 0x00
• #define CLK1 FILTER 0x01
• #define CLK2 FILTER 0x02
• #define CLK3 FILTER 0x03

    #define IOCONCLKDIV0 0x00

• #define IOCONCLKDIV1 0x01
• #define IOCONCLKDIV2 0x02
  #define IOCONCLKDIV3 0x03
```

#define IOCONCLKDIV4 0x04#define IOCONCLKDIV5 0x05

- #define IOCONCLKDIV6 0x06
- #define DAC_EN 0x01
- #define DAC DIS 0x00
- #define STD MODE 0x00
- #define STD_GPIO 0x01
- #define FAST MODE 0x02
- #define IOCON_ ((__RW uint32_t *) 0x40044000UL)

Functions

```
• void GPIO_Enable (void)
```

: Enable GPIO0 and GPIO1

• void GPIO_Disable (void)

: Disable GPIO0 and GPIO1

void GPIO_SetDIR (uint8_t port, uint8_t pin, uint8_t dir)

: Choose GPIO as Input/Output

void GPIO_SetPIN (uint8_t port, uint8_t pin, uint8_t state)

: Choose GPIO's output state

uint8_t GPIO_GetPIN (uint8_t port, uint8_t pin, uint8_t state)

: Return GPIO's input state

void GPIO SetOUT (uint8 t port, uint8 t pin)

: Put GPIO's out to 1

void GPIO_ClearOUT (uint8_t port, uint8_t pin)

: Put GPIO's out to 0

void GPIO_ToogleOUT (uint8_t port, uint8_t pin)

: Invert GPIO's out

· void GPIO_DebounceUserKEY (void)

: Firmware debounce for user key in board

void GPIO_Debounce (uint8_t port, uint8_t pin, uint8_t state)

: Firmware debounce for a GPIO

void IOCONEnable (void)

: Enable IOCON

void IOCONDisable (void)

: Disable IOCON

• uint8_t GetOFFSET (uint8_t port, uint8_t pin)

: Usefull for SetMode functions

void GPIO_SetModeINPUT (uint8_t port, uint8_t pin, uint8_t mode)

: on-chip pull-up/pull-down resistor

void GPIO_SetModeHYS (uint8_t port, uint8_t pin, uint8_t mode)

: Hysteresis

void GPIO_SetModeINV (uint8_t port, uint8_t pin, uint8_t mode)

: Invert input

void GPIO_SetModeOD (uint8_t port, uint8_t pin, uint8_t mode)

: Open drain

void GPIO_SetModeFILTER (uint8_t port, uint8_t pin, uint8_t mode)

: Digital filter sample mode

void GPIO_SetModeCLKDIV (uint8_t port, uint8_t pin, uint8_t mode)

: Select peripheral clock divider for input filter sampling clock

void GPIO_SetModeDAC (uint8_t port, uint8_t pin, uint8_t mode)

: Selects DAC mode

void GPIO_SetModel2C (uint8_t port, uint8_t pin, uint8_t mode)

: Selects I2C mode

4.5.1 Detailed Description

: Firmware functions for GPIO

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.5.2 Function Documentation

4.5.2.1 GetOFFSET()

: Usefull for SetMode functions

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 231 of file GPIO_FW.c.

4.5.2.2 GPIO_ClearOUT()

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 113 of file GPIO_FW.c.

{

4.5.2.3 GPIO_Debounce()

: Firmware debounce for a GPIO

: Use in SysTick_Handler or in some timer interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

	[in] uint8_t port : PORT0,PORT1
	[in] uint8_t pin: 0,31
	[in] uint8_t state : ACT_LOW,ACT_HIGH

Returns

: void

Definition at line 169 of file GPIO_FW.c.

```
169
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              {
                                                             170
 171
172
173
                                                          if (GPIO_GetPIN(port, pin, state)) // The key is pushed?
    j = 0x01; //Something is happening, the key is been pushed
  174
  175
                                                          if (buff_In ^ j) { // If the key is pushed while q != BOUNCE
  176
                                                                                     define continuous continuous
  177
                                                                                                                                                                                                                                                                                                    // I change the buffer
 178
179
  180
  181
                                                         } else
  183
                                                                                     q = 0;
184 }
```

4.5.2.4 GPIO_DebounceUserKEY()

: Firmware debounce for user key in board

: Use in SysTick_Handler or in some timer interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in]

Returns

: void

Definition at line 141 of file GPIO_FW.c.

```
141
142
          static uint8_t q = 0; //Quantity of bounces
                                       //It captures changes
143
         uint8_t j = 0;
144
         if (GPIO_GetPIN(UserKEY, ACT_LOW)) // The key is pushed?
    j = 0x01; //Something is happening, the key is been pushed
145
           j = 0x01;
146
147
         if (buff_UserKEY ^ j) {
                                                 // If the key is pushed while q != BOUNCE // I change the buffer
148
             q++;
if (q == BOUNCE) {
    q = 0;
'...ff UserKEY ^=
149
150
151
                   buff_UserKEY ^= 0x01;
152
153
154
         } else
155
            q = 0;
156 }
```

4.5.2.5 GPIO_Disable()

```
void GPIO_Disable (
     void )
```

: Disable GPIO0 and GPIO1

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 32 of file GPIO_FW.c.

4.5.2.6 GPIO_Enable()

```
void GPIO_Enable (
     void )
```

: Enable GPIO0 and GPIO1

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] void
```

Returns

: void

Definition at line 19 of file GPIO_FW.c.

4.5.2.7 GPIO_GetPIN()

: Return GPIO's input state

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin: 0,31
[in] uint8_t STATE : ACT_LOW,ACT_HIGH

Returns

```
: uint8_t : 1 pin == [state] , 0 pin != [state]
```

{

Definition at line 81 of file GPIO_FW.c.

4.5.2.8 GPIO_SetDIR()

: Choose GPIO as Input/Output

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

	[in] uint8_t port : PORT0,PORT1
	[in] uint8_t pin: 0,31
	[in] uint8_t dir : INPUT,OUTPUT

Returns

: void

Definition at line 48 of file GPIO_FW.c. 48

4.5.2.9 GPIO_SetModeCLKDIV()

```
uint8_t pin,
uint8_t mode )
```

: Select peripheral clock divider for input filter sampling clock

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: IOCONCLKDIV0 to IOCONCLKDIV6

Returns

: void

```
Definition at line 338 of file GPIO FW.c.
```

```
339     uint8_t offset;
340     offset = GetOFFSET(port, pin);
341     IOCON_[offset] &= (~(0x07 & 13));
342     IOCON_[offset] |= (mode & 13);
343 }
```

4.5.2.10 GPIO_SetModeDAC()

: Selects DAC mode

:

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: DAC_EN,DAC_DIS
```

Returns

: void

Definition at line 356 of file GPIO_FW.c.

4.5.2.11 GPIO SetModeFILTER()

: Digital filter sample mode

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:

BYPASS_FILTER,CLK1_FILTER,CLK2_FILTER,CLK3_FILTER
```

Returns

: void

Definition at line 320 of file GPIO_FW.c.

4.5.2.12 GPIO_SetModeHYS()

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:HYS_EN,HYS_DIS
```

Returns

: void

```
Definition at line 266 of file GPIO_FW.c.
```

4.5.2.13 GPIO_SetModel2C()

: Selects I2C mode

•

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:STD_MODE,STD_GPIO,FAST_MODE
```

Returns

: void

Definition at line 374 of file GPIO_FW.c.

```
374
375     uint8_t offset;
376     offset = GetOFFSET(port, pin);
377     IOCON_[offset] &= (~(0x03 « 8));
378     IOCON_[offset] |= (mode « 8);
379 }
```

4.5.2.14 GPIO_SetModeINPUT()

: on-chip pull-up/pull-down resistor

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:NO_PULL_UP_DOWN,PULL_DOWN,PULL_UP,REPEATER
```

Returns

: void

Definition at line 248 of file GPIO_FW.c.

4.5.2.15 GPIO_SetModelNV()

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: INV_INPUT,NOT_INV_INPUT
```

Returns

: void

Definition at line 284 of file GPIO_FW.c.

4.5.2.16 GPIO_SetModeOD()

: Open drain

:

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: OD_EN,OD_DIS
```

Returns

: void

Definition at line 302 of file GPIO_FW.c.

4.5.2.17 GPIO SetOUT()

: Put GPIO's out to 1

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 99 of file GPIO_FW.c.

```
99
100 GPIO_SETP[port] |= (1 « pin);
101 }
```

4.5.2.18 GPIO_SetPIN()

: Choose GPIO's output state

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin: 0,31
[in] uint8_t state : LOW,HIGH

Returns

: void

Definition at line 64 of file GPIO_FW.c.

```
65 port = port * 32 + pin;
66 GPIO_PBYTE[port] &= (~1);
67 GPIO_PBYTE[port] |= state;
68 }
```

4.5.2.19 GPIO_ToogleOUT()

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

```
Definition at line 127 of file GPIO_FW.c.
```

4.5.2.20 IOCONDisable()

```
void IOCONDisable ( \mbox{void} \quad \mbox{)}
```

: Disable IOCON

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in]

Returns

: void

Definition at line 208 of file GPIO_FW.c.

4.5.2.21 IOCONEnable()

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters



Returns

: void

Definition at line 195 of file GPIO_FW.c.

4.6 inc/GPIO_SW.h File Reference

: Software functions for GPIO

Functions

- uint8_t GetUserKEY (void)
 - : State of the user key in board
- uint8_t GetInput (void)
 - : State of the input

4.6.1 Detailed Description

: Software functions for GPIO

: These are functions in a higher layer of abstraction

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.6.2 Function Documentation

4.6.2.1 GetInput()

```
uint8_t GetInput (
     void )
```

: State of the input

: Is necessary using GPIO_Debounce

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] void
```

Returns

: uint8_t 1 if input pressed, 0 if input pressed

Definition at line 48 of file GPIO_SW.c.

4.6.2.2 GetUserKEY()

: State of the user key in board

: Is necessary using GPIO_DebounceUserKEY

Author

: Tobias Bavasso Piizzi

Date

Parameters

[in] void

Returns

: uint8_t 1 if user key pressed, 0 if user key not

Definition at line 21 of file GPIO_SW.c.

```
static uint8_t buff_before = 0x00;
23
       if ( buff_UserKEY == 0x01 && buff_before == 0x00 ){
25
          buff_before = 0x01;
26
          return (1);
27
      else if ( buff_UserKEY == 0x01 && buff_before == 0x01 )
      else if ( buff_UserKEY == 0x00 && buff_before == 0x01 ) {
30
       buff_before = 0x00;
31
32
          return (0);
33
      else
          return (0);
36 }
```

4.7 inc/LPC845.h File Reference

: Declarations for type of data

Macros

- #define __R volatile const
- #define __W volatile
- #define __RW volatile
- #define _ISER ((__RW uint32_t *) 0xE000E100UL)
- #define ISER0 _ISER[0]

Typedefs

- typedef unsigned int uint32_t
- typedef unsigned short uint16_t
- typedef unsigned char uint8_t

4.7.1 Detailed Description

: Declarations for type of data

: Only contains macros

Author

: Tobias Bavasso Piizzi

Date

4.8 inc/SwitchMatrix FW.h File Reference

: Firmware functions for SWM

Macros

```
    #define PINASSIGN ( (__RW uint32_t *) 0x4000C000UL)
    #define PINENABLE ( (__RW uint32_t *) 0x4000C1C0UL)
```

Enumerations

```
enum { BYTE0 , BYTE1 , BYTE2 , BYTE3 }
enum {
 UO_TXD , UO_SCLK , U1_CTS , U2_RTS ,
 SPI0_MOSI, SPI0_SSEL2, SPI1_MISO, SCT_IN1,
 SCT OUT1, SCT OUT5, I2C2 SDA, COMP0 OUT,
 UART3 RXD, UART4 SCLK, TO MAT3 }
enum {
 U0 RXD, U1 TXD, U0 SCLK, U2 CTS,
 SPI0 MISO, SPI0 SSEL3, SPI1 SSEL0, SCT IN2,
 SCT_OUT2, SCT_OUT6, I2C2_SCL, CLKOUT,
 UART3_SCLK, T0_MAT0, T0_CAP0}
• enum {
 UO RTS, U1 RXD, U2 TXD, U2 SCLK,
 SPIO SSELO, SPI1 SCK, SPI1 SSEL1, SCT IN3,
 SCT_OUT3, I2C1_SDA, I2C3_SDA, GPIO_INT_BMAT,
 UART4_TXD, T0_MAT1, T0_CAP1}
• enum {
 UO CTS, U1 RTS, UO RXD, SPIO SCK,
 SPI0_SSEL1, SPI1_MOSI, SCT0_IN0, SCT_OUT0,
 SCT_OUT4, I2C1_SCL, I2C3_SCL, UART3_TXD,
 UART4_RXD, T0_MAT2, T0_CAP2}
enum {
 ADC 0, ADC 1, ADC 2, ADC 3,
 ADC 4, ADC 5, ADC 6, ADC 7,
 ADC_8, ADC_9, ADC_10, ADC_11,
 DACOUTO, DACOUT1, CAPT_X0, CAPT_X1,
 CAPT_X2, CAPT_X3}
enum {
 CAPT_X4, CAPT_X5, CAPT_X6, CAPT_X7,
 CAPT_X8, CAPT_YL, CAPT_YH}
```

Functions

```
    void SWM (uint8_t port, uint8_t pin, uint8_t assign, uint8_t byte)

            Assign movable functions for pin

    void SWM_PinEnable (uint8_t port, uint8_t pin, uint8_t ena)

            Enable pin works as value passed in ena

    void SWM_Enable (void)

            Enable SWM

    void SWM_Disable (void)

            Disable SWM
```

4.8.1 Detailed Description

: Firmware functions for SWM

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.8.2 Enumeration Type Documentation

4.8.2.1 anonymous enum

anonymous enum

Enumerator

UO_TXD | Possible assign.

Definition at line 38 of file SwitchMatrix_FW.h.

```
38
39
                UO_TXD,
               UO_SCLK,
U1_CTS,
U2_RTS,
40
41
42
              SPIO_MOSI,
SPIO_SSEL2,
SPII_MISO,
SCT_IN1,
SCT_OUTI,
43
44
45
46
48
               SCT_OUT5,
               I2C2_SDA,
COMP0_OUT,
UART3_RXD,
UART4_SCLK,
T0_MAT3
49
50
51
52
53
54 };
```

4.8.2.2 anonymous enum

anonymous enum

Enumerator

U0_RXD Possible assign.

Definition at line 56 of file SwitchMatrix_FW.h.

```
56
57
         {
U0_RXD,
U1_TXD,
58
         U0_SCLK,
U2_CTS,
SPI0_MISO,
59
60
61
          SPIO_SSEL3,
63
         SPI1_SSEL0,
         SCT_IN2,
SCT_OUT2,
SCT_OUT6,
64
65
66
          I2C2_SCL,
68
          CLKOUT,
          UART3_SCLK,
69
70
71
          TO_MATO,
          T0_CAP0
72 };
```

4.8.2.3 anonymous enum

anonymous enum

Enumerator

UO RTS Possible assign.

Definition at line 74 of file SwitchMatrix_FW.h.

```
74
75
        UO_RTS,
76
        U1_RXD,
77
        U2_TXD,
78
       U2_SCLK,
SPI0_SSEL0,
79
80
        SPI1_SCK,
81
        SPI1_SSEL1,
        SCT_IN3,
83
        SCT_OUT3,
84
       I2C1_SDA,
        I2C3_SDA,
GPIO_INT_BMAT,
85
86
        UART4_TXD,
88
        TO_MAT1,
89
        T0_CAP1
90 };
```

4.8.2.4 anonymous enum

anonymous enum

Enumerator

UO_CTS Possible assign.

Definition at line 92 of file SwitchMatrix_FW.h.

```
98 SPI1_MOSI,
99 SCT0_INO,
100 SCT_OUTO,
101 SCT_OUT4,
102 I2C1_SCL,
103 I2C3_SCL,
104 UART3_TXD,
105 UART4_RXD,
106 T0_MAT2,
107 T0_CAP2
108 };
```

4.8.3 Function Documentation

4.8.3.1 SWM()

: Assign movable functions for pin

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin: 0,31
[in] uint8_t assign :
[in] uint8_t byte : BYTE0,BYTE1,BYTE2,BYTE3

Returns

: void

Definition at line 22 of file SwitchMatrix_FW.c.

4.8.3.2 SWM_Disable()

```
void SWM_Disable (
void )

: Disable SWM

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters
```

Returns

: void

[in] void

Definition at line 67 of file SwitchMatrix_FW.c.

4.8.3.3 SWM_Enable()

```
void SWM_Enable (
     void )
: Enable SWM
```

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 54 of file SwitchMatrix_FW.c.

```
55 SYSAHBCLKCTRL0|= (1«7);
56 }
```

4.8.3.4 SWM_PinEnable()

: Enable pin works as value passed in ena

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
[in] uint8_t ena : READ Page 143 UserManual. There are multiple choices
```

Returns

: void

Definition at line 38 of file SwitchMatrix_FW.c.

4.9 inc/SYSCON_FW.h File Reference

: Firmware functions for SYSCON

Macros

- #define SYSCON_ADD ((__RW uint32_t *) 0x40048000UL)
- #define SYSMEMREMAP SYSCON ADD [0]
- #define SYSPLLCTRL SYSCON ADD [2]
- #define SYSPLLSTAT SYSCON ADD [3]
- #define SYSOSCCTRL SYSCON ADD [8]
- #define WDTOSCCTRL SYSCON ADD [9]
- #define FROOSCCTRL SYSCON_ADD [10]
- #define FRODIRECTCLKUEN SYSCON ADD [12]
- #define SYSRSTSTAT SYSCON_ADD [14]
- #define SYSPLLCLKSEL SYSCON ADD [16]
- #define SYSPLLCLKUEN SYSCON ADD [17]
- #define MAINCLKPLLSEL SYSCON ADD [18]
- #define MAINCLKPLLUEN SYSCON ADD [19]
- #define MAINCLKSEL SYSCON_ADD [20]
- #define MAINCLKUEN SYSCON ADD [21]
- #define SYSAHBCLKDIV SYSCON ADD [22]
- #define CAPTCLKSEL SYSCON ADD [24]
- #define ADCCLKSEL SYSCON_ADD [25]
- #define ADCCLKDIV SYSCON_ADD [26]
- #define SCTCLKSEL SYSCON_ADD [27]
- #define SCTCLKDIV SYSCON ADD [28]
- #define EXTCLKSEL SYSCON_ADD [29]
- #define _SYSAHBCLKCTRL0 SYSCON ADD [32]
- #define SYSAHBCLKCTRL1 SYSCON ADD [33]
- #define PRESETCTRL0 SYSCON_ADD [34]
- #define PRESETCTRL1 SYSCON ADD [35]
- #define UARTOCLKSEL SYSCON ADD [36]
- #define UART1CLKSEL SYSCON_ADD [37]
- #define UART2CLKSEL SYSCON_ADD [38]
- #define UART3CLKSEL SYSCON_ADD [39]
- #define UART4CLKSEL SYSCON_ADD [40]
- #define I2C0CLKSEL SYSCON_ADD [41]
- #define I2C1CLKSEL SYSCON_ADD [42]
- #define I2C2CLKSEL SYSCON_ADD [43]
- #define I2C3CLKSEL SYSCON_ADD [44]
- #define SPIOCLKSEL SYSCON_ADD [45]
- #define SPI1CLKSEL SYSCON_ADD [46]
- #define FRG0DIV SYSCON ADD [52]
- #define FRG0MULT SYSCON_ADD [53]
- #define FRG0CLKSEL SYSCON_ADD [54]
- #define FRG1DIV SYSCON_ADD [56]
- #define FRG1MULT SYSCON_ADD [57]
- #define FRG1CLKSEL SYSCON ADD [58]
- #define CLKOUTSEL SYSCON ADD [60]
- #define CLKOUTDIV SYSCON ADD [61]
- #define EXTTRACECMD SYSCON ADD [63]
- #define PIOPORCAPO SYSCON ADD [64]
- #define PIOPORCAP1 SYSCON_ADD [65]
- #define _IOCONCLKDIV6 SYSCON_ADD [77]
- #define _IOCONCLKDIV5 SYSCON_ADD [78]
- #define _IOCONCLKDIV4 SYSCON_ADD [79]
- #define IOCONCLKDIV3 SYSCON ADD [80]
- #define _IOCONCLKDIV2 SYSCON_ADD [81]

- #define IOCONCLKDIV1 SYSCON ADD [82]
- #define _IOCONCLKDIV0 SYSCON ADD [83]
- #define BODCTRL SYSCON ADD [84]
- #define SYSTCKCAL SYSCON ADD [85]
- #define IRQLATENCY SYSCON ADD [92]
- #define NMISRC SYSCON ADD [93]
- #define PINTSEL0 SYSCON ADD [94]
- #define PINTSEL1 SYSCON_ADD [95]
- #define PINTSEL2 SYSCON ADD [96]
- #define PINTSEL3 SYSCON ADD [97]
- #define PINTSEL4 SYSCON_ADD [98]
- #define PINTSEL5 SYSCON ADD [99]
- #define PINTSEL6 SYSCON ADD [100]
- #define PINTSEL7 SYSCON ADD [101]
- #define STARTERPO SYSCON ADD [129]
- #define STARTERP1 SYSCON ADD [133]
- #define PDSLEEPCFG SYSCON_ADD [140]
- #define PDAWAKECFG SYSCON_ADD [141]
- #define PDRUNCFG SYSCON ADD [142]
- #define **DEVICE_ID** SYSCON ADD [254]
- #define CLOCK_FRO_SETTING_API_ROM_ADDRESS 0x0F0026F5U
- #define F30MHz 30000U
- #define FRO_OUT_PowerDown 1
- #define FRO PD 2
- #define SYSCON FROOSCCTRL FRO DIRECT MASK (0x20000U)
- #define SYSCON_FROOSCCTRL_FRO_DIRECT_SHIFT (17U)
- #define kCLOCK_FroSrcFroOsc 1U << SYSCON_FROOSCCTRL_FRO_DIRECT_SHIFT
- #define kPDRUNCFG PD SYSOSC 0x20
- #define CLK_FROM_SYS_OSC 0x00
- #define FREQ30MHz 30000000U
- #define CLK_SYS_PLLSRCFRODIV 0x03
- #define CLOCK FAIM BASE 0x50010000U
- #define SYSPLL_MIN_FCCO_FREQ_HZ 156000000U
- #define SYSCON_SYSPLLCTRL_MSEL_MASK 0x1FU
- #define SYSCON_SYSPLLCTRL_MSEL_SHIFT (0U)
- #define SYSCON_SYSPLLCTRL_PSEL_MASK 0x60U
- #define SYSCON SYSPLLCTRL PSEL SHIFT (5U)
- #define SYSCON_SYSPLLCTRL_MSEL(x) (((uint32_t)(((uint32_t)(x)) << SYSCON_SYSPLLCTRL_←
 MSEL SHIFT)) & SYSCON SYSPLLCTRL MSEL MASK)
- #define SYSCON_SYSPLLCTRL_PSEL(x) (((uint32_t)(((uint32_t)(x)) << SYSCON_SYSPLLCTRL_↔
 PSEL_SHIFT)) & SYSCON_SYSPLLCTRL_PSEL_MASK)
- #define **CLK_MAIN_CLK_MUX_GET_MUX**(x) ((uint32_t)(x) & 0xFFU)
- #define CLK MAIN CLK MUX GET PRE MUX(x) (((uint32 t)(x) >> 8U) & 0xFFU)
- #define SYSCON MAINCLKSEL SEL MASK 0x03U
- #define SYSCON_MAINCLKSEL_SEL_SHIFT (0U)
- #define SYSCON MAINCLKPLLSEL SEL MASK (0x3U)
- #define SYSCON MAINCLKPLLSEL SEL SHIFT (0U)
- #define SYSCON_MAINCLKPLLSEL_SEL(x) (((uint32_t)(((uint32_t)(x)) << SYSCON_MAINCLKPLLSEL ← SEL SHIFT)) & SYSCON MAINCLKPLLSEL SEL MASK)
- #define kCLOCK MainClkSrcFro 0
- #define SYSCON_SYSAHBCLKDIV_DIV(x) (((uint32_t)(((uint32_t)(x)) << SYSCON_SYSAHBCLKDIV_

 DIV SHIFT)) & SYSCON SYSAHBCLKDIV DIV MASK)
- #define SYSCON SYSAHBCLKDIV DIV MASK 0xFFU
- #define SYSCON SYSAHBCLKDIV DIV SHIFT (0U)

Functions

- void BoardClockRUN ()
 - : Runs clock at 30MHz
- void ClockSetFroOscFREQ (uint32 t freq)
- void PowerDisablePD (uint8_t en)
- void CLOCK_SetFroOutClkSrc (uint32_t src)
- void CLOCK_Select (uint8_t sel)
- void CLOCK_InitSystemPII (uint32_t freq, uint8_t src)
- uint32_t CLOCK_GetSystemPLLInClockRate (void)
- uint32_t CLOCK_GetFroFreq (void)
- uint32_t FindSyestemPIIPsel (uint32_t outFreq)
- void CLOCK_SetMainClkSrc (uint32_t src)
- void CLOCK_SetCoreSysClkDiv (uint32_t value)

4.9.1 Detailed Description

: Firmware functions for SYSCON

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.9.2 Function Documentation

4.9.2.1 BoardClockRUN()

```
void BoardClockRUN (
     void )
```

: Runs clock at 30MHz

: Select clock from fro

Author

: Tobias Bavasso Piizzi

Date

Parameters

[in] void

Returns

: void

Definition at line 19 of file SYSCON_FW.c.

```
19 {
20    PowerDisablePD(FRO_OUT_PowerDown);
21    PowerDisablePD(FRO_PD);
22    ClockSetFroOscfREQ(F30MHz);
23    CLOCK_SetFroOutClkSrc(kCLOCK_FroSrcFroOsc);
24    PowerDisablePD(kPDRUNCFG_PD_SYSOSC);
25    CLOCK_Select(CLK_FROM_SYS_OSC);
26    CLOCK_InitSystemPl1(FREQ30MHz, CLK_SYS_PLLSRCFRODIV);
27    CLOCK_SetMainClkSrc(kCLOCK_MainClkSrcFro);
28    CLOCK_SetCoreSysClkDiv(lU);
```

4.10 inc/SysTick_FW.h File Reference

: Firmware functions for SysTick

Macros

- #define TICK OUT 1S 100
 - Systick interrupt each 1 second.
- #define **SysTick**_ ((__RW uint32_t *) 0xE000E000UL)
- #define SYST_CSR SysTick_[4]
- #define SYST_RVR SysTick_[5]
- #define SYST_CVR SysTick [6]
- #define SYST_CALIB SysTick_[7]
- #define SYSTICK_ENABLE_INTERRUPT_CLK 0x07
- #define SYSTICK DISABLE 0x00
- #define **SYSTICK_INT_DIS** SYST_CSR &= \sim 0x02;
- #define SYSTICK_INT_EN SYST_CSR = SYSTICK_ENABLE_INTERRUPT_CLK;
- #define FRE30MHz 30000U

Functions

- void SysTick_Init (void)
 - : Initialize the systick
- void SysTick_Off (void)
 - : Stops the systick
- void SysTick_Set (uint32_t freq)
 - : Set the counter as freq* 10mS -1

4.10.1 Detailed Description

: Firmware functions for SysTick

: Used for 30 MHz

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.10.2 Function Documentation

4.10.2.1 SysTick_Init()

```
void SysTick_Init (
            void )
```

: Initialize the systick

: Enable SysTick, enable interrupt and set the counter

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 19 of file SysTick_FW.c.

```
SysTick_Set(FRE30MHz);
SYST_CSR = SYSTICK_ENABLE_INTERRUPT_CLK;
SYST_CVR = 0;
20
21
23 }
```

4.10.2.2 SysTick_Off()

```
void SysTick_Off (
     void )
```

: Stops the systick

: disable SysTick, disable interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 34 of file SysTick_FW.c.

```
34 {
35     SYST_CSR = SYSTICK_DISABLE;
```

4.10.2.3 SysTick_Set()

: Set the counter as freq*10mS -1

: Always use at 30MHz

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint32_t freq: FRE30MHz
```

Returns

: void

```
Definition at line 47 of file SysTick_FW.c.
```

4.11 source/05-ConversionADC.c File Reference

```
: Firmware functions ADC
```

```
#include "Aplication.h"
```

Functions

• int main (void)

: Main Function

Variables

```
• uint32_t tick = 0
```

Var for SysTick_Handler.

• uint32_t conv = 0

Var for ADC.

4.11.1 Detailed Description

: Firmware functions ADC

: 12 bits convertion

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

4.11.2 Function Documentation

4.11.2.1 main()

```
:int main (
     void )
```

: Main Function

: initialize the system and stay in the while

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

[in] void

Returns

: int

Definition at line 23 of file 05-ConversionADC.c.

```
uint8_t state = 1;
24
             LPC_Init();
ADC_Init(ADC_0);
25
26
27
28
              while(1) {
29
                       if(state == 0) {
   if ( conv > (0xFFF/2)) {
      state ++;
      GPIO_SetPIN(LedGREEN, LED_OFF);
      GPIO_SetPIN(LedBLUE, LED_ON);
}
30
31
32
33
34
35
36
                       else if (state == 1) {
   if ( conv <= (0xFFF/2)) {
     state --;
     GPIO_SetPIN(LedGREEN, LED_ON);
     GPIO_SetPIN(LedBLUE, LED_OFF);
}</pre>
37
38
39
40
41
42
43
44
45
46
               return 0 ;
47 }
```

4.11.3 Variable Documentation

4.11.3.1 tick

```
uint32\_t tick = 0
```

Var for SysTick_Handler.

Declared in main.

Definition at line 20 of file 05-ConversionADC.c.

4.12 source/ADC_FW.c File Reference

```
: Firmware functions ADC
```

```
#include "Aplication.h"
```

Functions

- void ADC_Init (uint8_t port, uint8_t pin, uint8_t ena)
 - : Initialize ADC on a pin
- void ADC_Power (void)
 - : Power ADC
- void ADC_Enable (void)
 - : Enable clock in ADC
- void ADC_Disable (void)
 - : Disable clock in ADC
- void ADC0_SEQA_IRQHandler (void)
 - : Interruption for ADC

Variables

uint32_t tick

Var for SysTick_Handler.

· uint32_t conv

Var for ADC.

4.12.1 Detailed Description

: Firmware functions ADC

: 12 bits convertion

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

4.12.2 Function Documentation

4.12.2.1 ADC0_SEQA_IRQHandler()

: Interruption for ADC

: Interrupt when some channel finishes its conversion

Author

: Tobias Bavasso Piizzi

Date

: 10/01/2021

Parameters

[in] void

Returns

: void

- < Clean flags
- < Read global data
- < Make an average
- < Start a new conversion

Definition at line 104 of file ADC_FW.c.

```
static uint8_t i = 0;
static uint32_t sum = 0;
106
107
108
         (void) _ADC_SEQA_GDAT;
109
110
         sum += ADC_SEQA_GDAT->_RESULT;
111
         if('i == 0xFF') {
112
             conv = sum/i;
i = 0;
113
114
115
                      = 0;
              sum
116
117
         ADC_SEQA_CTRL->_START = 1;
118 }
```

4.12.2.2 ADC_Disable()

```
:void ADC_Disable (
            void )
: Disable clock in ADC
Author
    : Tobias Bavasso Piizzi
Date
    : 08/01/2021
```

Parameters

[in] void

Returns

: void

4.12.2.3 ADC_Enable()

```
:void ADC_Enable (
         void )
```

: Enable clock in ADC

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

[in] void

Returns

: void

Definition at line 77 of file ADC_FW.c.

```
77 {
78 SYSAHBCLKCTRL0|= (1«ADC_SYSAHB);
79 }
```

4.12.2.4 ADC_Init()

: Initialize ADC on a pin

: Continuos conversion of POTE in board

Author

: Tobias Bavasso Piizzi

Date

: 08/01/2021

Parameters

	[in] uint8_t port: PORT0,PORT1
	[in] uint8_t pin: 0,31
Ī	[in] uint8_t en: bit to enable in PINENABLE (page 143 UM)

Returns

: void

- < Enable CLOCK in SYSAHB
- < Enable service interrupt
- < Interrupt after conversion finish
- < Enable Switch Matrix
- < Enable pin in SWN as AnalogInput
- < Disable Switch Matrix
- < Power in SYSCON

```
< Div = 0
```

< Sync

< OFF

< OFF

- < Sample CH0
- < No hardware trigger
- < Positive trigger
- < Enable sync
- < Individual end of conversion
- < Start, enable set on the same line first time

Definition at line 23 of file ADC_FW.c.

```
24
         ADC_Enable();
ISER0|= MASK_ISE_ADC_SEQA;
ADC_INTEN|= MASK_SEQA_INTEN;
26
27
2.8
         SWM_Enable();
29
         SWM_PinEnable(port, pin, ena);
30
         SWM_Disable();
31
         ADC_Power();
32
33
34
        ADC_CTRL->_CLKDIV = 0x00;
ADC_CTRL->_ASYNCMODE = 0;
ADC_CTRL->_LPWRMODE = 0;
35
36
38
         ADC_CTRL->_CALMODE = 0;
39
         ADC_SEQA_CTRL->_CHANNELS
                                                   = 0x01;
40
         ADC_SEQA_CTRL->_TRIGGER
ADC_SEQA_CTRL->_TRIGPOL
                                                   = 0x00;
41
                                                   = 0x1;
42
43
         ADC_SEQA_CTRL->_SYNCBYPASS
         ADC_SEQA_CTRL->_TSAMP
ADC_SEQA_CTRL->_START
                                                   = 0x00;
45
                                                   = 0;
        ADC_SEQA_CIRL->_BURST
ADC_SEQA_CTRL->_SINGLESTEP
ADC_SEQA_CTRL->_LOWPRIO
                                                  = 0;
46
47
                                                         = 0x0;
                                                   = 0x0;
48
49
        ADC_SEQA_CTRL->_MODE
                                                  = 0;
50
        ADC_SEQA_CTRL->_SEQx_ENA
51
         _ADC_SEQA_CTRL |= ((0b100001) « 26);
52 }
```

4.12.2.5 ADC_Power()

```
:void ADC_Power (
     void )
: Power ADC
:
```

: Tobias Bavasso Piizzi

Date

Author

: 08/01/2021

Parameters

[in] void

Returns

: void

Definition at line 63 of file ADC_FW.c.

```
63 {
64 PDRUNCFG&= (~(1 « MASK_ADC_SYSCON));
65
66 }
```

4.13 source/Aplication.c File Reference

```
: Functions used in main
```

```
#include "Aplication.h"
```

Functions

```
• void LPC_Init (void)
```

: Initialize the board

• void GPIO_Init (void)

: Initialize the GPIO

Variables

• uint32_t tick

Declared in main.

4.13.1 Detailed Description

: Functions used in main

.

Author

: Tobias Bavasso Piizzi

Date

4.13.2 Function Documentation

4.13.2.1 GPIO_Init()

: Initialize the GPIO

: It depends on each proyect

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 35 of file Aplication.c.

```
35 {
36 GPIO_SetDIR(UserKEY, INPUT);
37 GPIO_SetDIR(LedGREEN, OUTPUT);
38 GPIO_SetDIR(LedBLUE, OUTPUT);
39
40 GPIO_SetPIN(LedGREEN, LED_OFF);
41 GPIO_SetPIN(LedBLUE, LED_OFF);
```

4.13.2.2 LPC_Init()

: Initialize the board

: It depends on each proyect

Author

: Tobias Bavasso Piizzi

Date

Parameters

[in] void

Returns

: void

Definition at line 19 of file Aplication.c.

```
19 {
20 GPIO_Enable();
21 BoardClockRUN();
22 SysTick_Init();
23 GPIO_Init();
```

4.13.3 Variable Documentation

4.13.3.1 tick

```
uint32_t tick [extern]
```

Declared in main.

Declared in main.

Definition at line 20 of file 05-ConversionADC.c.

4.14 source/Disp7Seg_FW.c File Reference

```
: Firmware functions for DISP7SEG
```

```
#include "Aplication.h"
```

Functions

```
• void DISP7SEG_Init (void)
```

: Set pins for display as out

void DISP_Sweep (void)

: Refresh the display 7Seg (2 Disp)

Variables

```
    __RW uint8_t buff_Disp7 []
    Display buffer.
```

4.14.1 Detailed Description

```
: Firmware functions for DISP7SEG
:
Author
: Tobias Bavasso Piizzi

Date
: 07/01/2021
```

4.14.2 Function Documentation

4.14.2.1 DISP7SEG_Init()

```
:void DISP7SEG_Init (void)

: Set pins for display as out

:

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

Parameters

[in] void
```

Returns

: void

Definition at line 19 of file Disp7Seg_FW.c.

```
19 {
20 GPIO_SetDIR(SEG_A, OUTPUT);
21 GPIO_SetDIR(SEG_B, OUTPUT);
22 GPIO_SetDIR(SEG_C, OUTPUT);
23 GPIO_SetDIR(SEG_D, OUTPUT);
24 GPIO_SetDIR(SEG_E, OUTPUT);
25 GPIO_SetDIR(SEG_F, OUTPUT);
```

```
26
         GPIO_SetDIR(SEG_G, OUTPUT);
        GPIO_SetDIR(TR_D0, OUTPUT);
GPIO_SetDIR(TR_D1, OUTPUT);
28
29
         GPIO_ClearOUT(SEG_A);
GPIO_ClearOUT(SEG_B);
30
31
32
         GPIO_ClearOUT(SEG_C);
33
         GPIO_ClearOUT(SEG_D);
34
         GPIO_ClearOUT(SEG_E);
35
36
         GPIO_ClearOUT(SEG_F);
GPIO_ClearOUT(SEG_G);
         GPIO_ClearOUT(TR_D0);
37
38
         GPIO_ClearOUT(TR_D1);
```

4.14.2.2 DISP_Sweep()

: Refresh the display 7Seg (2 Disp)

: Is necessary to be used in SysTick_Handler

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

Parameters

```
[in] void
```

Returns

: void

- < Number of disp
- < Turn off transistor
- < Turn off transistor
- < Next time sweep other disp
- < Reset the digits

Definition at line 51 of file Disp7Seg_FW.c.

```
51 {
52     uint8_t aux;
53     static uint8_t digit = 0;
54
55     GPIO_ClearOUT(TR_D0);
56     GPIO_ClearOUT(TR_D1);
```

```
aux = buff_Disp7[digit];
59
         GPIO_SetPIN( SEG_A, ((aux » 0) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_B, ((aux » 1) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_C, ((aux » 2) & (uint8_t) 0x01));
60
61
          GPIO_SetPIN( SEG_D, ((aux » 3) & (uint8_t) 0x01));
63
          GPIO_SetPIN( SEG_E, ((aux » 4) & (uint8_t) 0x01));
        GPIO_SetPIN( SEG_F, ((aux » 5) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_G, ((aux » 6) & (uint8_t) 0x01));
GPIO_SetPIN( SEG_DP, ((aux » 7) & (uint8_t) 0x01));
65
66
67
68
69
         switch (digit) {
70
       case DIGIT_0:
          GPIO_SetOUT(TR_D0);
71
72
73
         case DIGIT 1:
          GPIO_SetOUT(TR_D1);
74
75
               break;
76
         default:
              digit = 0;
78
               GPIO_SetOUT(TR_D0);
79
               break;
80
81
         digit++;
83
         digit %= DIGITS;
84
85 }
```

4.15 source/Disp7Seg_SW.c File Reference

```
: Software functions for DISP7SEG
```

```
#include "Aplication.h"
```

Functions

```
    void Display (uint8_t val)
```

: Writes on Disp7Seg

Variables

```
    __RW uint8_t buff_Disp7 [DIGITS]
    Buffer de display.
```

```
uint8_t Digits_to_BCD7seg []
```

__RW uint8_t tick_Disp7

4.15.1 Detailed Description

```
: Software functions for DISP7SEG
```

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

4.15.2 Function Documentation

4.15.2.1 Display()

: Writes on Disp7Seg

: High lever of layers

Author

: Tobias Bavasso Piizzi

Date

: 07/01/2021

Parameters

```
[in] uint8_t val: 0 to 99
```

Returns

: void

< Disable SysTick INT

< Enable SysTick INT

Definition at line 38 of file Disp7Seg_SW.c.

4.15.3 Variable Documentation

4.15.3.1 Digits_to_BCD7seg

```
uint8_t Digits_to_BCD7seg[]
Initial value:
= { 0x3f, 0x06, 0x5B, 0x4f, 0x66, 0x6D, 0x7C, 0x07, 0x7f, 0x67 }
```

Tabla de conversion bcd a 7 segmentos Codigo bcd a b c d e f g dp 0 1 1 1 1 1 1 1 0 0 1 0 1 1 0 0 0 0 2 1 1 0 1 1 0 1 3 1 1 1 1 1 0 0 1 4 0 1 1 0 0 1 1 5 1 0 1 1 0 1 1 6 0 0 1 1 1 1 1 7 1 1 1 0 0 0 0 8 1 1 1 1 1 1 1 9 1 1 1 0 0 1 1

Definition at line 26 of file Disp7Seg_SW.c.

4.16 source/GPIO_FW.c File Reference

```
: Firmware functions for GPIO #include "Aplication.h"
```

Functions

```
• void GPIO_Enable (void)
     : Enable GPIO0 and GPIO1

    void GPIO_Disable (void)

     : Disable GPIO0 and GPIO1

    void GPIO_SetDIR (uint8_t port, uint8_t pin, uint8_t dir)

     : Choose GPIO as Input/Output

    void GPIO_SetPIN (uint8_t port, uint8_t pin, uint8_t state)

     : Choose GPIO's output state

    uint8 t GPIO GetPIN (uint8 t port, uint8 t pin, uint8 t state)

     : Return GPIO's input state

    void GPIO_SetOUT (uint8_t port, uint8_t pin)

     : Put GPIO's out to 1

    void GPIO_ClearOUT (uint8_t port, uint8_t pin)

     : Put GPIO's out to 0

    void GPIO_ToogleOUT (uint8_t port, uint8_t pin)

     : Invert GPIO's out

    void GPIO_DebounceUserKEY (void)

     : Firmware debounce for user key in board
• void GPIO Debounce (uint8 t port, uint8 t pin, uint8 t state)
     : Firmware debounce for a GPIO

    void IOCONEnable (void)

     : Enable IOCON
• void IOCONDisable (void)
     : Disable IOCON

    uint8 t GetOFFSET (uint8 t port, uint8 t pin)

     : Usefull for SetMode functions

    void GPIO SetModeINPUT (uint8 t port, uint8 t pin, uint8 t mode)
```

: on-chip pull-up/pull-down resistor

```
    void GPIO_SetModeHYS (uint8_t port, uint8_t pin, uint8_t mode)

            Hysteresis

    void GPIO_SetModeINV (uint8_t port, uint8_t pin, uint8_t mode)

            Invert input

    void GPIO_SetModeOD (uint8_t port, uint8_t pin, uint8_t mode)

            Open drain

    void GPIO_SetModeFILTER (uint8_t port, uint8_t pin, uint8_t mode)

            Digital filter sample mode

    void GPIO_SetModeCLKDIV (uint8_t port, uint8_t pin, uint8_t mode)

            Select peripheral clock divider for input filter sampling clock

    void GPIO_SetModeDAC (uint8_t port, uint8_t pin, uint8_t mode)

            Selects DAC mode

    void GPIO_SetModel2C (uint8_t port, uint8_t pin, uint8_t mode)

            Selects I2C mode
```

Variables

```
    __RW uint8_t buff_UserKEY = 0
    __RW uint8_t buff_In = 0
    uint8_t offset []
```

4.16.1 Detailed Description

```
: Firmware functions for GPIO
:
Author
: Tobias Bavasso Piizzi

Date
: 04/01/2021
```

4.16.2 Function Documentation

4.16.2.1 GetOFFSET()

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 231 of file GPIO FW.c.

4.16.2.2 GPIO_ClearOUT()

: Put GPIO's out to 0

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 113 of file GPIO_FW.c.

```
113
114 GPIO_CLRP[port] |= (1 « pin);
115 }
```

4.16.2.3 GPIO_Debounce()

: Firmware debounce for a GPIO

: Use in SysTick_Handler or in some timer interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

		[in] uint8_t port : PORT0,PORT1
		[in] uint8_t pin: 0,31
[in] uint8_t state : ACT_LOW,A		[in] uint8_t state : ACT_LOW,ACT_HIGH

Returns

: void

Definition at line 169 of file GPIO_FW.c.

```
169
170
       171
172
173
       if (GPIO_GetPIN(port, pin, state))
                                        // The key is pushed?
          j = 0 \times 01;
174
                                //Something is happening, the key is been pushed
175
176
177
                                // If the key is pushed while q != BOUNCE
       if (buff_In ^ j) {
          q++;
if (q == BOUNCE) {
                                    // I change the buffer
178
179
              q = 0;
              buff_In ^= 0x01;
181
182
       } else
          q = 0;
183
184 }
```

4.16.2.4 GPIO_DebounceUserKEY()

: Firmware debounce for user key in board

: Use in SysTick_Handler or in some timer interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters



Returns

: void

Definition at line 141 of file GPIO_FW.c.

```
141
        142
143
        uint8_t j = 0;
144
       if (GPIO_GetPIN(UserKEY, ACT_LOW))
                                             // The key is pushed?
145
           j = 0x01;
146
                                  //Something is happening, the key is been pushed
147
                                         // If the key is pushed while q != BOUNCE // I change the buffer
148
        if (buff_UserKEY ^ j) {
        if (put___
    q++;
    if (q == BOUNCE) {
        q = 0;
        '...ff UserKEY ^=
149
150
151
               buff_UserKEY ^= 0x01;
152
153
154
      } else
          q = 0;
155
156 }
```

4.16.2.5 GPIO_Disable()

: Disable GPIO0 and GPIO1

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

```
Returns
```

: void

```
Definition at line 32 of file GPIO_FW.c. ^{32} { ^{33} SYSAHBCLKCTRL0&= (~(1«6));
           SYSAHBCLKCTRL0&= (~(1«6));
SYSAHBCLKCTRL0 &= (~(1«20));
```

4.16.2.6 GPIO_Enable()

```
:void GPIO_Enable (
           void )
```

: Enable GPIO0 and GPIO1

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] void
```

Returns

: void

Definition at line 19 of file GPIO_FW.c.

4.16.2.7 GPIO_GetPIN()

```
:uint8_t GPIO_GetPIN (
            uint8_t port,
            uint8_t pin,
            uint8_t dir )
```

: Return GPIO's input state

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

		[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin: 0,31		[in] uint8_t pin: 0,31
		[in] uint8_t STATE : ACT_LOW,ACT_HIGH

Returns

```
: uint8_t : 1 pin == [state] , 0 pin != [state]
```

Definition at line 81 of file GPIO_FW.c.

```
81
82     port = port * 32 + pin;
83     if ( GPIO_PBYTE[port] == state)
84         return 1;
85     else
86         return 0;
87 }
```

4.16.2.8 GPIO_SetDIR()

: Choose GPIO as Input/Output

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

	[in] uint8_t port : PORT0,PORT				
	[in] uint8_t pin: 0,31				
	[in] uint8_t dir : INPUT,OUTPUT				

```
Returns
```

: void

```
Definition at line 48 of file GPIO_FW.c.
```

4.16.2.9 GPIO_SetModeCLKDIV()

: Select peripheral clock divider for input filter sampling clock

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: IOCONCLKDIV0 to IOCONCLKDIV6
```

{

Returns

: void

Definition at line 338 of file GPIO_FW.c.

4.16.2.10 GPIO_SetModeDAC()

: Selects DAC mode

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: DAC_EN,DAC_DIS
```

{

Returns

: void

Definition at line 356 of file GPIO_FW.c.

4.16.2.11 GPIO_SetModeFILTER()

: Digital filter sample mode

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: BYPASS_FILTER,CLK1_FILTER,CLK2_FILTER,CLK3_FILTER
```

{

Returns

: void

Definition at line 320 of file GPIO_FW.c.

4.16.2.12 GPIO_SetModeHYS()

: Hysteresis

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:HYS_EN,HYS_DIS
```

Returns

: void

Definition at line 266 of file GPIO_FW.c.

4.16.2.13 GPIO_SetModel2C()

```
uint8_t pin,
uint8_t mode )
: Selects I2C mode
:
```

_

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:STD_MODE,STD_GPIO,FAST_MODE
```

Returns

: void

```
Definition at line 374 of file GPIO FW.c.
```

4.16.2.14 GPIO_SetModeINPUT()

: on-chip pull-up/pull-down resistor

:

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode:NO_PULL_UP_DOWN,PULL_DOWN,PULL_UP,REPEATER
```

Returns

: void

Definition at line 248 of file GPIO_FW.c.

4.16.2.15 GPIO_SetModelNV()

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1: [in] uint8_t pin: 0,31: [in] uint8_t mode: INV_INPUT,NOT_INV_INPUT
```

Returns

: void

Definition at line 284 of file GPIO_FW.c.

4.16.2.16 GPIO_SetModeOD()

Date

: 04/01/2021

Parameters

```
[in] uint8_t port: PORT0,PORT1 : [in] uint8_t pin: 0,31 : [in] uint8_t mode: OD_EN,OD_DIS
```

{

Returns

: void

```
Definition at line 302 of file GPIO_FW.c.
```

4.16.2.17 GPIO_SetOUT()

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 99 of file GPIO_FW.c.

```
99
100 GPIO_SETP[port] |= (1 « pin);
101 }
```

4.16.2.18 GPIO_SetPIN()

: Choose GPIO's output state

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] uint8_t port : PORT0,PORT1			
[in] uint8_t pin: 0,31			
[in] uint8_t state : LOW,HIGH			

Returns

: void

Definition at line 64 of file GPIO_FW.c.

```
64
65 port = port * 32 + pin;
66 GPIO_PBYTE[port] &= (~1);
67 GPIO_PBYTE[port] |= state;
68 }
```

4.16.2.19 GPIO_ToogleOUT()

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] uint8_t port : PORT0,PORT1
[in] uint8_t pin : 0,31
```

Returns

: void

Definition at line 127 of file GPIO_FW.c.

4.16.2.20 IOCONDisable()

: Disable IOCON

:

Author

: Tobias Bavasso Piizzi

Date

Parameters [in] Returns : void Definition at line 208 of file GPIO_FW.c. 4.16.2.21 IOCONEnable() :void IOCONEnable (void) : Enable IOCON Author : Tobias Bavasso Piizzi Date : 04/01/2021 **Parameters** [in] Returns

4.16.3 Variable Documentation

Definition at line 195 of file GPIO_FW.c.

: void

4.16.3.1 offset

```
uint8_t offset[]
```

Initial value:

```
= { 0x044, 0x02C, 0x018, 0x014, 0x010, 0x00C, 0x040, 0x03C, 0x038, 0x034, 0x020, 0x01C, 0x008, 0x004, 0x048, 0x028, 0x024, 0x000, 0x078, 0x074, 0x070, 0x06C, 0x068, 0x064, 0x060, 0x05C, 0x058, 0x054, 0x050, 0x068, 0x06C, 0x08C, 0x090, 0x094, 0x098, 0x0A4, 0x0A8, 0x0AC, 0x0B8, 0x0C4, 0x0D8, 0x0B4, 0x0B4, 0x0BC, 0x0B0, 0x0B0, 0x0B4, 0x0BC, 0x0BC, 0x0B0, 0x0B4, 0x0BC, 0x0BC, 0x0B0, 0x0B4, 0x0BC, 0x0BC
```

Definition at line 214 of file GPIO_FW.c.

4.17 source/GPIO SW.c File Reference

```
: Software functions for GPIO
```

```
#include "Aplication.h"
```

Functions

- uint8 t GetUserKEY (void)
 - : State of the user key in board
- uint8_t GetInput (void)
 - : State of the input

Variables

- uint8_t buff_UserKEY
- uint8_t buff_In

4.17.1 Detailed Description

: Software functions for GPIO

: These functions avoid bouncing. Both must be used w/ GPIO_DebounceUserKEY or GPIO_Debounce

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.17.2 Function Documentation

4.17.2.1 GetInput()

: State of the input

: Is necessary using GPIO_Debounce

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

```
[in] void
```

Returns

: uint8_t 1 if input pressed, 0 if input pressed

Definition at line 48 of file GPIO_SW.c.

4.17.2.2 GetUserKEY()

: State of the user key in board

: Is necessary using GPIO_DebounceUserKEY

Author

: Tobias Bavasso Piizzi

Date

Parameters

[in] void

Returns

: uint8_t 1 if user key pressed, 0 if user key not

Definition at line 21 of file GPIO_SW.c.

```
static uint8_t buff_before = 0x00;
22
23
       if ( buff_UserKEY == 0x01 && buff_before == 0x00 ){
2.4
25
           buff_before = 0 \times 01;
26
           return (1);
28
       else if ( buff_UserKEY == 0x01 && buff_before == 0x01 )
2.9
           return (0);
       else if ( buff_UserKEY == 0x00 && buff_before == 0x01 ) {
30
31
           buff_before = 0x00;
32
           return (0);
35
           return (0);
36 1
```

4.18 source/mtb.c File Reference

MTB initialization file.

```
#include <cr_mtb_buffer.h>
```

Macros

• #define MTB BUFFER SIZE 128

Functions

__CR_MTB_BUFFER (__MTB_BUFFER_SIZE)

4.18.1 Detailed Description

MTB initialization file.

Symbols controlling behavior of this code... __MTB_DISABLE If this symbol is defined, then the buffer array for the MTB will not be created.

__MTB_BUFFER_SIZE Symbol specifying the sizer of the buffer array for the MTB. This must be a power of 2 in size, and fit into the available RAM. The MTB buffer will also be aligned to its 'size' boundary and be placed at the start of a RAM bank (which should ensure minimal or zero padding due to alignment).

__MTB_RAM_BANK Allows MTB Buffer to be placed into specific RAM bank. When this is not defined, the "default" (first if there are several) RAM bank is used.

4.19 source/SwitchMatrix_FW.c File Reference

```
: Firmware functions for SWM #include "Aplication.h"
```

Functions

```
    void SWM (uint8_t port, uint8_t pin, uint8_t assign, uint8_t byte)

            : Assign movable functions for pin

    void SWM_PinEnable (uint8_t port, uint8_t pin, uint8_t ena)

            : Enable pin works as value passed in ena

    void SWM_Enable (void)

            : Enable SWM

    void SWM_Disable (void)

            : Disable SWM
```

4.19.1 Detailed Description

```
: Firmware functions for SWM
:
Author
: Tobias Bavasso Piizzi
Date
```

: 04/01/2021

4.19.2 Function Documentation

4.19.2.1 SWM()

Parameters

	[in] uint8_t port : PORT0,PORT1
	[in] uint8_t pin: 0,31
	[in] uint8_t assign :
	[in] uint8_t byte : BYTE0,BYTE1,BYTE2,BYTE3

Returns

: void

Definition at line 22 of file SwitchMatrix_FW.c.

4.19.2.2 SWM_Disable()

```
:void SWM_Disable ( void )
```

: Disable SWM

:

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 67 of file SwitchMatrix_FW.c.

```
67 {
68 SYSAHBCLKCTRL0&= (~(1«7));
69 }
```

4.19.2.3 SWM_Enable()

```
    :void SWM_Enable (
        void )
    : Enable SWM
    :
    Author
    : Tobias Bavasso Piizzi
```

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 54 of file SwitchMatrix_FW.c.

4.19.2.4 SWM_PinEnable()

: Enable pin works as value passed in ena

:

Author

: Tobias Bavasso Piizzi

Date

Parameters

		[in] uint8_t port : PORT0,PORT1	
ſ		[in] uint8_t pin: 0,31	
[in] uint8_t ena : READ Page 143 UserManual. There are multiple choice			

Returns

: void

Definition at line 38 of file SwitchMatrix_FW.c.

4.20 source/SYSCON_FW.c File Reference

: Firmware functions for SYSCON

```
#include "Aplication.h"
```

Functions

- void BoardClockRUN (void)
 - : Runs clock at 30MHz
- void ClockSetFroOscFREQ (uint32_t freq)
- void PowerDisablePD (uint8_t en)
- void CLOCK_SetFroOutClkSrc (uint32_t src)
- void CLOCK_Select (uint8_t sel)
- void CLOCK_InitSystemPII (uint32_t freq, uint8_t src)
- uint32_t CLOCK_GetSystemPLLInClockRate (void)
- uint32_t CLOCK_GetFroFreq (void)
- uint32_t FindSyestemPIIPsel (uint32_t outFreq)
- void CLOCK_SetMainClkSrc (uint32_t src)
- void CLOCK_SetCoreSysClkDiv (uint32_t value)

4.20.1 Detailed Description

: Firmware functions for SYSCON

: Only starts the board at 30MHz

Author

: Tobias Bavasso Piizzi

Date

4.20.2 Function Documentation

4.20.2.1 BoardClockRUN()

: Runs clock at 30MHz

: Select clock from fro

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 19 of file SYSCON_FW.c.

```
19 {
20    PowerDisablePD(FRO_OUT_PowerDown);
21    PowerDisablePD(FRO_PD);
22    ClockSetFroOscFREQ(F30MHz);
23    CLOCK_SetFroOutClkSrc(kCLOCK_FroSrcFroOsc);
24    PowerDisablePD(kPDRUNCFG_PD_SYSOSC);
25    CLOCK_Select(CLK_FROM_SYS_OSC);
26    CLOCK_InitSystemPl1(FREQ30MHz, CLK_SYS_PLLSRCFRODIV);
27    CLOCK_SetMainClkSrc(kCLOCK_MainClkSrcFro);
28    CLOCK_SetCoreSysClkDiv(1U);
29 }
```

4.21 source/SysTick_FW.c File Reference

```
: Firmware functions for SysTick
```

```
#include "Aplication.h"
```

Functions

```
    void SysTick_Init (void)

            Initialize the systick

    void SysTick_Off (void)

            Stops the systick

    void SysTick_Set (uint32_t freq)

            Set the counter as freq*10mS-1

    void SysTick_Handler (void)

            Interrupt each 10mS
```

Variables

uint32_t tick
 Declared in main.

4.21.1 Detailed Description

: Firmware functions for SysTick

: Only develop for 30MHz

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

4.21.2 Function Documentation

4.21.2.1 SysTick_Handler()

Date

Parameters

[in] void

Returns

: void

Definition at line 61 of file SysTick_FW.c.

```
61 {
62 63 if (tick >= 0U)
64 tick--;
65 66 67 68 }
```

4.21.2.2 SysTick_Init()

: Initialize the systick

: Enable SysTick, enable interrupt and set the counter

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 19 of file SysTick_FW.c.

```
19 SysTick_Set(FRE30MHz);
21 SYST_CSR = SYSTICK_ENABLE_INTERRUPT_CLK;
22 SYST_CVR = 0;
23 }
```

4.21.2.3 SysTick_Off()

```
: SysTick_Off ( void )
```

: Stops the systick

: disable SysTick, disable interrupt

Author

: Tobias Bavasso Piizzi

Date

: 04/01/2021

Parameters

[in] void

Returns

: void

Definition at line 34 of file SysTick_FW.c.

```
34 {
35 SYST_CSR = SYSTICK_DISABLE;
36 }
```

4.21.2.4 SysTick_Set()

: Set the counter as freq*10mS -1

: Always use at 30MHz

Author

: Tobias Bavasso Piizzi

Date

Parameters

```
[in] uint32_t freq: FRE30MHz
```

Returns

: void

Definition at line 47 of file SysTick_FW.c.

```
47
48 SYST_RVR = freq*10 - 1; // 30MHz*10mS-1
49 }
```

4.21.3 Variable Documentation

4.21.3.1 tick

```
uint32_t tick [extern]
```

Declared in main.

Declared in main.

Definition at line 20 of file 05-ConversionADC.c.

Index

05.0	DIOD O
05-ConversionADC.c	DISP_Sweep, 64
main, 54	Disp7Seg_FW.h
tick, 54	DISP7SEG_Init, 18
ADOS 0504 IDOU #	DISP_Sweep, 19
ADC0_SEQA_IRQHandler	DISP7SEG_Init
ADC_FW.c, 56	Disp7Seg_FW.c, 63
adc_chan, 5	Disp7Seg_FW.h, 18
adc_ctrl_t, 6	Disp7Seg_SW.c
ADC_Disable	Digits_to_BCD7seg, 66
ADC_FW.c, 56	Display, 66
ADC_FW.h, 11	Disp7Seg_SW.h
ADC_Enable	Display, 21
ADC_FW.c, 57	DISP_Sweep
ADC_FW.h, 12	Disp7Seg_FW.c, 64
ADC FW.c	Disp7Seg_FW.h, 19
ADC0_SEQA_IRQHandler, 56	Display
ADC Disable, 56	Disp7Seg SW.c, 66
ADC Enable, 57	
ADC_Init, 58	Disp7Seg_SW.h, 21
ADC_Power, 59	GetInput
ADC_FW.h	GPIO_SW.c, 83
ADC_Disable, 11	GPIO_SW.h, 38
ADC_Enable, 12	GetOFFSET
ADC_Init, 12	GPIO_FW.c, 68
ADC_Power, 14	GPIO_FW.h, 24
ADC_Init	GetUserKEY
ADC_FW.c, 58	GPIO_SW.c, 84
ADC_FW.h, 12	GPIO_SW.h, 39
ADC_Power	GPIO_ClearOUT
ADC_FW.c, 59	GPIO_FW.c, 69
ADC_FW.h, 14	GPIO_FW.h, 24
adc_seqX_ctrl, 7	GPIO_Debounce
adc_seqX_gdat, 7	GPIO_FW.c, 69
adc_thr, 8	GPIO_FW.h, 25
Aplication.c	GPIO DebounceUserKEY
GPIO_Init, 61	GPIO_FW.c, 70
LPC_Init, 61	GPIO_FW.h, 26
tick, 62	GPIO_Disable
Aplication.h	GPIO_FW.c, 71
GPIO_Init, 15	GPIO FW.h, 27
LPC Init, 16	GPIO_Enable
Li 0_ iiiit, 10	GPIO_FW.c, 72
BoardClockRUN	
SYSCON FW.c, 90	GPIO_FW.h, 27
SYSCON_FW.h, 49	GPIO_FW.c
01000N_1 VV.II, +3	GetOFFSET, 68
Digits to BCD7seg	GPIO_ClearOUT, 69
Disp7Seg SW.c, 66	GPIO_Debounce, 69
Disp7Seg_SW.c, 00 Disp7Seg FW.c	GPIO_DebounceUserKEY, 70
DISP7SEG_Init, 63	GPIO_Disable, 71
5.01 / OEG_IIII, 00	GPIO Enable 72

96 INDEX

GPIO_GetPIN, 72	GPIO_FW.h, 31
GPIO SetDIR, 73	GPIO SetModel2C
GPIO_SetModeCLKDIV, 74	GPIO FW.c, 76
GPIO_SetModeDAC, 74	GPIO FW.h, 32
GPIO SetModeFILTER, 75	GPIO SetModeINPUT
	_
GPIO_SetModeHYS, 76	GPIO_FW.c, 77
GPIO_SetModel2C, 76	GPIO_FW.h, 33
GPIO_SetModeINPUT, 77	GPIO_SetModeINV
GPIO_SetModeINV, 78	GPIO_FW.c, 78
GPIO SetModeOD, 78	GPIO FW.h, 33
GPIO SetOUT, 79	GPIO SetModeOD
GPIO_SetPIN, 80	GPIO FW.c, 78
GPIO_ToogleOUT, 80	GPIO FW.h, 34
_ ·	GPIO_SetOUT
IOCONDisable, 81	
IOCONEnable, 82	GPIO_FW.c, 79
offset, 82	GPIO_FW.h, 35
GPIO_FW.h	GPIO_SetPIN
GetOFFSET, 24	GPIO_FW.c, 80
GPIO_ClearOUT, 24	GPIO FW.h, 35
GPIO Debounce, 25	GPIO SW.c
GPIO DebounceUserKEY, 26	GetInput, 83
GPIO Disable, 27	GetUserKEY, 84
-	
GPIO_Enable, 27	GPIO_SW.h
GPIO_GetPIN, 28	GetInput, 38
GPIO_SetDIR, 29	GetUserKEY, 39
GPIO_SetModeCLKDIV, 29	GPIO_ToogleOUT
GPIO_SetModeDAC, 30	GPIO_FW.c, 80
GPIO SetModeFILTER, 31	GPIO FW.h, 36
GPIO SetModeHYS, 31	<u> </u>
GPIO SetModel2C, 32	inc/ADC_FW.h, 9
GPIO_SetModeINPUT, 33	inc/Aplication.h, 15
	inc/Disp7Seg_FW.h, 17
GPIO_SetModeINV, 33	
GPIO_SetModeOD, 34	inc/Disp7Seg_SW.h, 20
GPIO_SetOUT, 35	inc/GPIO_FW.h, 21
GPIO_SetPIN, 35	inc/GPIO_SW.h, 38
GPIO_ToogleOUT, 36	inc/LPC845.h, 40
IOCONDisable, 37	inc/SwitchMatrix_FW.h, 41
IOCONEnable, 37	inc/SYSCON_FW.h, 46
GPIO GetPIN	inc/SysTick_FW.h, 50
GPIO FW.c, 72	IOCONDisable
— · · · · · · · · · · · · · · · · · · ·	GPIO FW.c, 81
GPIO_FW.h, 28	GPIO FW.h, 37
GPIO_Init	- · · · ·
Aplication.c, 61	IOCONEnable
Aplication.h, 15	GPIO_FW.c, 82
GPIO_SetDIR	GPIO_FW.h, 37
GPIO_FW.c, 73	
GPIO FW.h, 29	LPC_Init
GPIO SetModeCLKDIV	Aplication.c, 61
GPIO_FW.c, 74	Aplication.h, 16
	•
GPIO_FW.h, 29	main
GPIO_SetModeDAC	05-ConversionADC.c, 54
GPIO_FW.c, 74	22 22
GPIO_FW.h, 30	offset
GPIO_SetModeFILTER	GPIO FW.c, 82
	GI 10_1 VV.0, 02
GPIO FW.h, 31	source/05-ConversionADC a 52
GPIO_SetModeHYS	source/05-ConversionADC.c, 53
	source/ADC_FW.c, 55
GPIO_FW.c, 76	source/Aplication.c, 60

INDEX 97

source/Disp7Seg_FW.c, 62		SysTick_FW.h, 52
source/Disp7Seg_SW.c, 65		
source/GPIO_FW.c, 67	tick	
source/GPIO_SW.c, 83		05-ConversionADC.c, 54
source/mtb.c, 85		Aplication.c, 62
source/SwitchMatrix_FW.c, 86		SysTick_FW.c, 94
source/SYSCON_FW.c, 89		
source/SysTick_FW.c, 90	U0_	RXD
SwitchMatrix FW.c		SwitchMatrix_FW.h, 42
SWM, 86	UO_	_CTS
SWM_Disable, 87		SwitchMatrix_FW.h, 43
SWM_Enable, 87	UO_	_RTS
SWM PinEnable, 88		SwitchMatrix_FW.h, 43
SwitchMatrix FW.h	UO	_TXD
SWM, 44		SwitchMatrix_FW.h, 42
SWM_Disable, 44		
SWM_Enable, 45		
SWM PinEnable, 46		
U0 RXD, 42		
UO_CTS, 43		
UO_RTS, 43		
UO_TXD, 42		
SWM		
SwitchMatrix_FW.c, 86		
SwitchMatrix_FW.h, 44		
SWM_Disable		
SwitchMatrix_FW.c, 87		
SwitchMatrix_FW.h, 44		
SWM_Enable		
SwitchMatrix_FW.c, 87		
SwitchMatrix_FW.h, 45		
SWM_PinEnable		
SwitchMatrix_FW.c, 88		
SwitchMatrix_FW.h, 46		
SYSCON_FW.c		
BoardClockRUN, 90		
SYSCON_FW.h		
BoardClockRUN, 49		
SysTick_FW.c		
SysTick_Handler, 91		
SysTick_Init, 92		
SysTick_Off, 92		
SysTick_Set, 93		
tick, 94		
SysTick_FW.h		
SysTick_Init, 51		
SysTick_Off, 51		
SysTick_Set, 52		
SysTick_Handler		
SysTick_FW.c, 91		
SysTick_Init		
SysTick_FW.c, 92		
SysTick_FW.h, 51		
SysTick_Off		
SysTick_FW.c, 92		
SysTick_FW.h, 51		
SysTick_Set		
SysTick_FW.c, 93		