Table of Contents

Context_View diagram

Use Case diagram in package 'Requirements'

Context_View
Version 1.0
Thomas Batt created on 09.03.2016. Last modified 13.04.2022

Name: Context_View
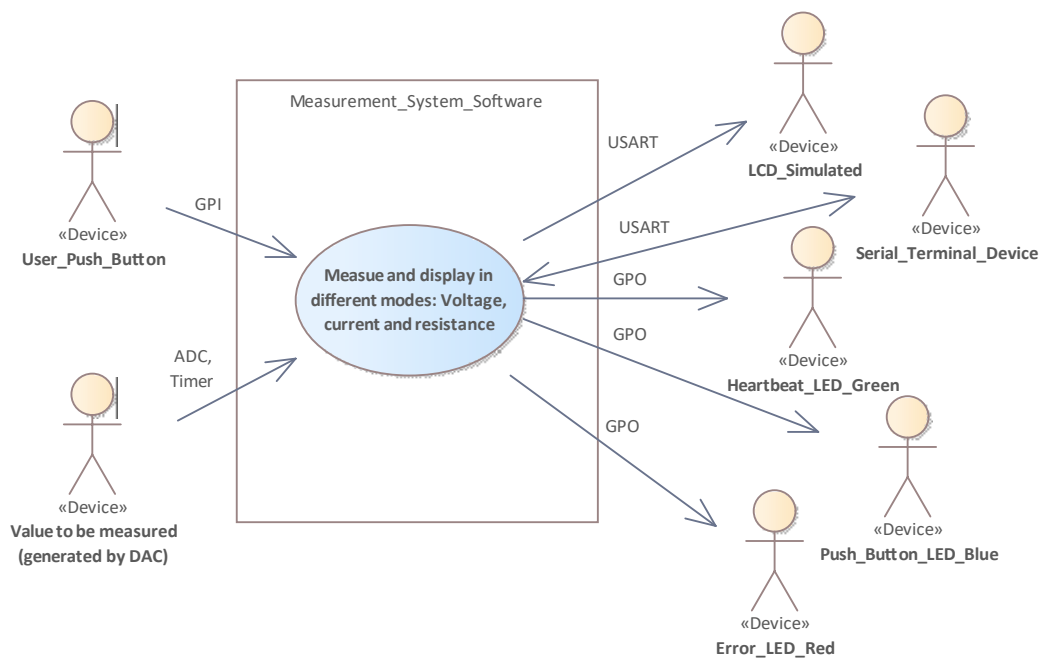Package: Requirements
Version: 1.0
Author: Thomas Batt

Figure 1: Context_View

Functional-Requirements_View diagram

Use Case diagram in package 'Requirements'

Functional-Requirements_View
Version 1.0
Thomas Batt created on 09.03.2016. Last modified 13.04.2022

Name: Functional-Requirements_View
Package: Requirements
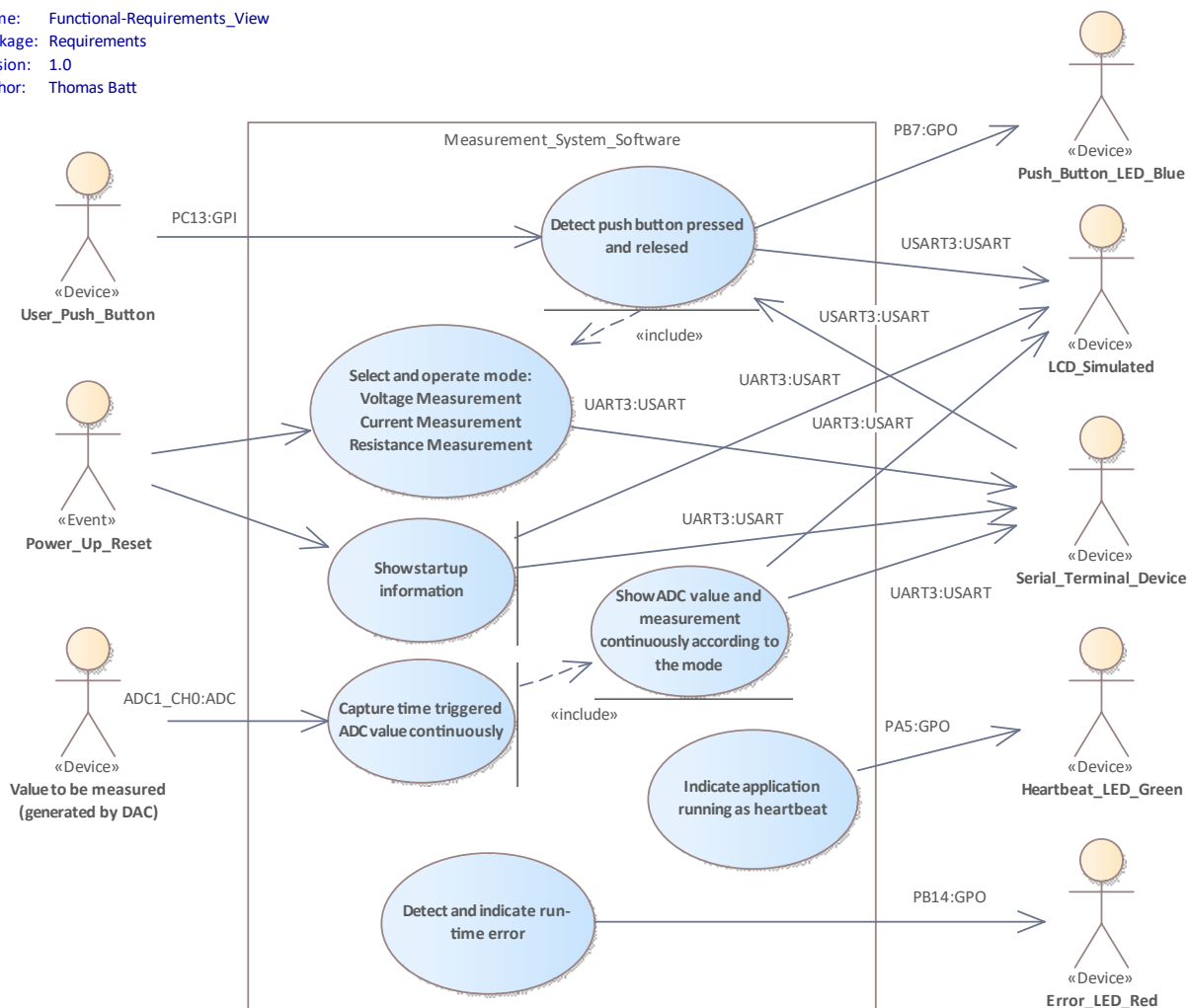Version: 1.0
Author: Thomas Batt

Measurement_System_Software

PB7:GPO

«Device»
Push_Button_LED_Blue

PC13:GPI

Detect push button pressed
and relesed

«Device»
User_Push_Button

«include»

USART3:USART

USART3:USART

«Device»
LCD_Simulated

UART3:USART

Select and operate mode:
Voltage Measurement
Current Measurement
Resistance Measurement

UART3:USART

UART3:USART

«Event»
Power_Up_Reset

Show startup
information

Show ADC value and
measurement
continuously according to
the mode

UART3:USART

«Device»
Serial_Terminal_Device

ADC1_CH0:ADC

Capture time triggered
ADC value continuously

«include»

Indicate application
running as heartbeat

PA5:GPO

«Device»
Heartbeat_LED_Green

«Device»
Value to be measured
(generated by DAC)

Detect and indicate run-
time error

PB14:GPO

«Device»
Error_LED_Red

Figure 2: Functional-Requirements_View

Task_Management diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Task_Management
Version 1.0
Thomas Batt created on 19.02.2016. Last modified 09.04.2020

Name: Task_Management
Package: Software_Run-Time-Architectures
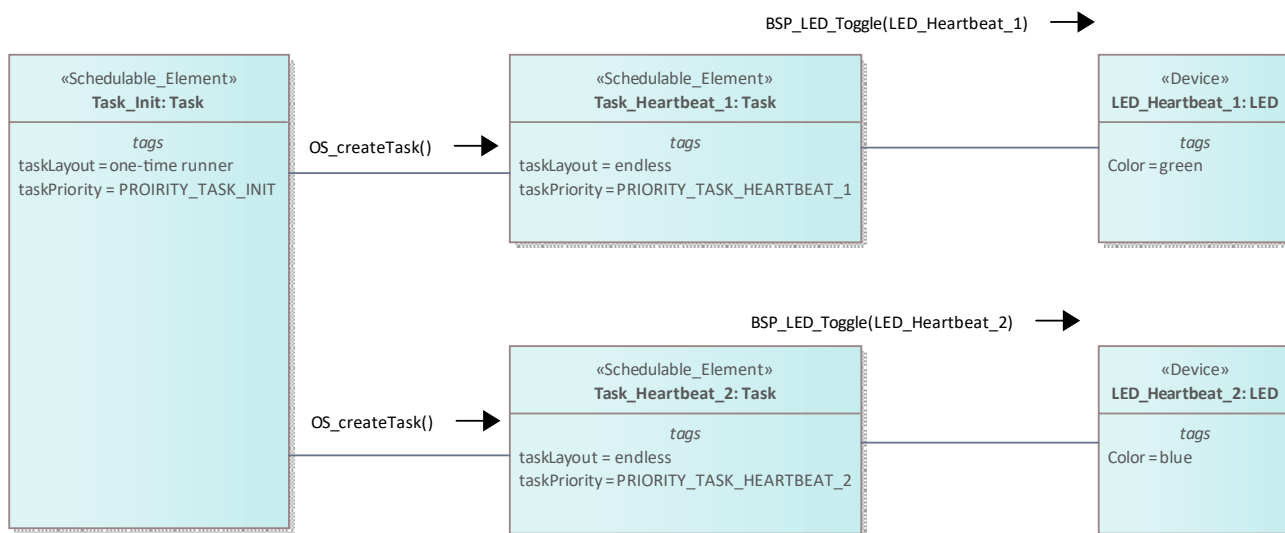Version: 1.0
Author: Thomas Batt

BSP_LED_Toggle(LED_Heartbeat_1) ➡

«Schedulable_Element» **Task_Init: Task**
tags
taskLayout = one-time runner
taskPriority = PROIRITY_TASK_INIT

OS_createTask() ➡

«Schedulable_Element» **Task_Heartbeat_1: Task**
tags
taskLayout = endless
taskPriority = PRIORITY_TASK_HEARTBEAT_1

«Device» **LED_Heartbeat_1: LED**
tags
Color = green

BSP_LED_Toggle(LED_Heartbeat_2) ➡

OS_createTask() ➡

«Schedulable_Element» **Task_Heartbeat_2: Task**
tags
taskLayout = endless
taskPriority = PRIORITY_TASK_HEARTBEAT_2

«Device» **LED_Heartbeat_2: LED**
tags
Color = blue

Figure 3: Task_Management

Synchronization_and_Interrupt_Management diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Synchronization_and_Interrupt_Management
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Synchronization_and_Interrupt_Management
Package: Software_Run-Time-Architectures
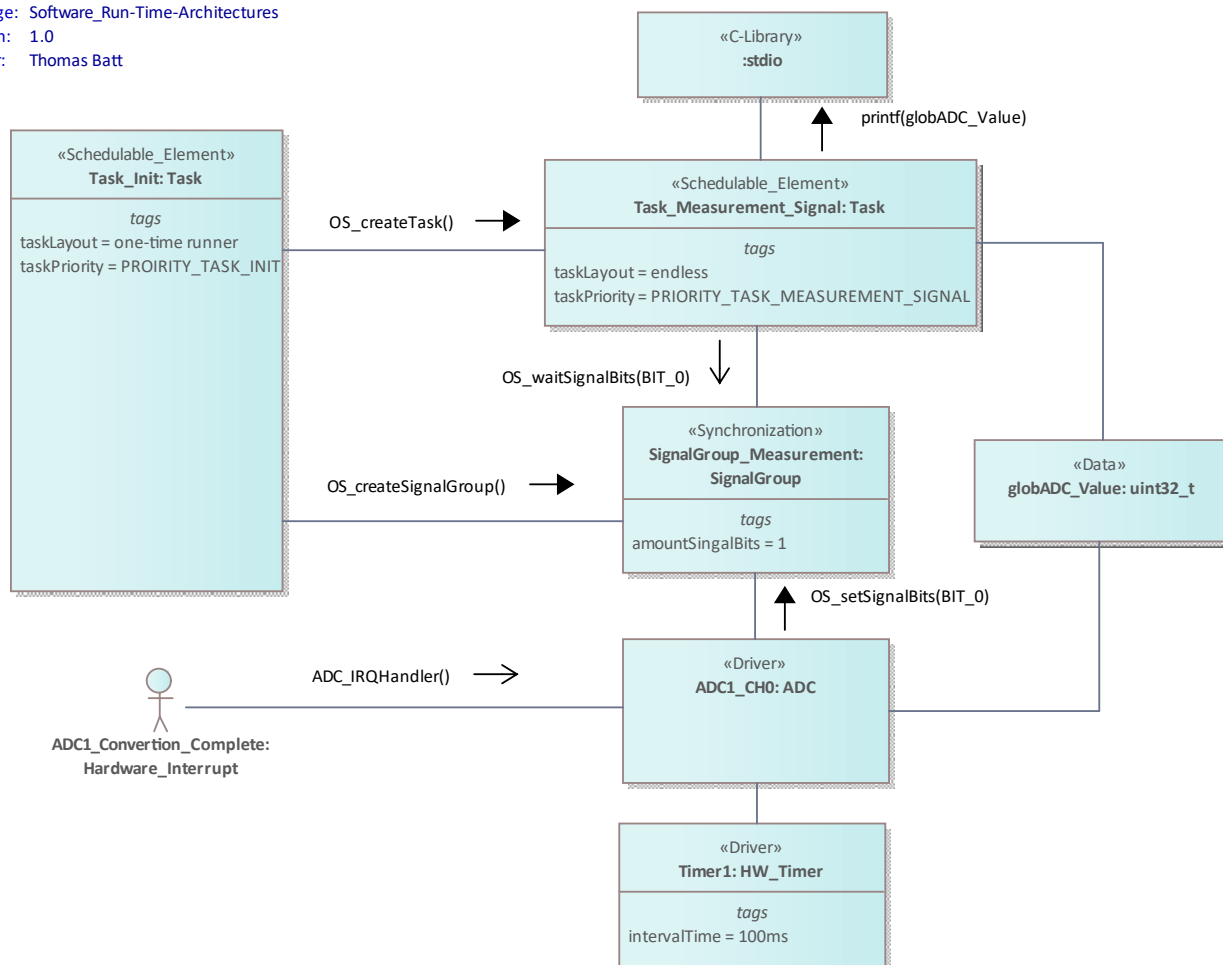Version: 1.0
Author: Thomas Batt

«C-Library»
:stdio

printf(globADC_Value)

«Schedulable_Element»
Task_Init: Task

tags
taskLayout = one-time runner
taskPriority = PROIRITY_TASK_INIT

OS_createTask()

«Schedulable_Element»
Task_Measurement_Signal: Task

tags
taskLayout = endless
taskPriority = PRIORITY_TASK_MEASUREMENT_SIGNAL

OS_waitSignalBits(BIT_0)

«Synchronization»
SignalGroup_Measurement: SignalGroup

tags
amountSingalBits = 1

OS_createSignalGroup()

«Data»
globADC_Value: uint32_t

OS_setSignalBits(BIT_0)

ADC_IRQHandler()

«Driver»
ADC1_CH0: ADC

ADC1_Convertion_Complete: Hardware_Interrupt

«Driver»
Timer1: HW_Timer

tags
intervalTime = 100ms

Figure 4: Synchronization_and_Interrupt_Management

Communication diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Communication
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Communication
Package: Software_Run-Time-Architectures
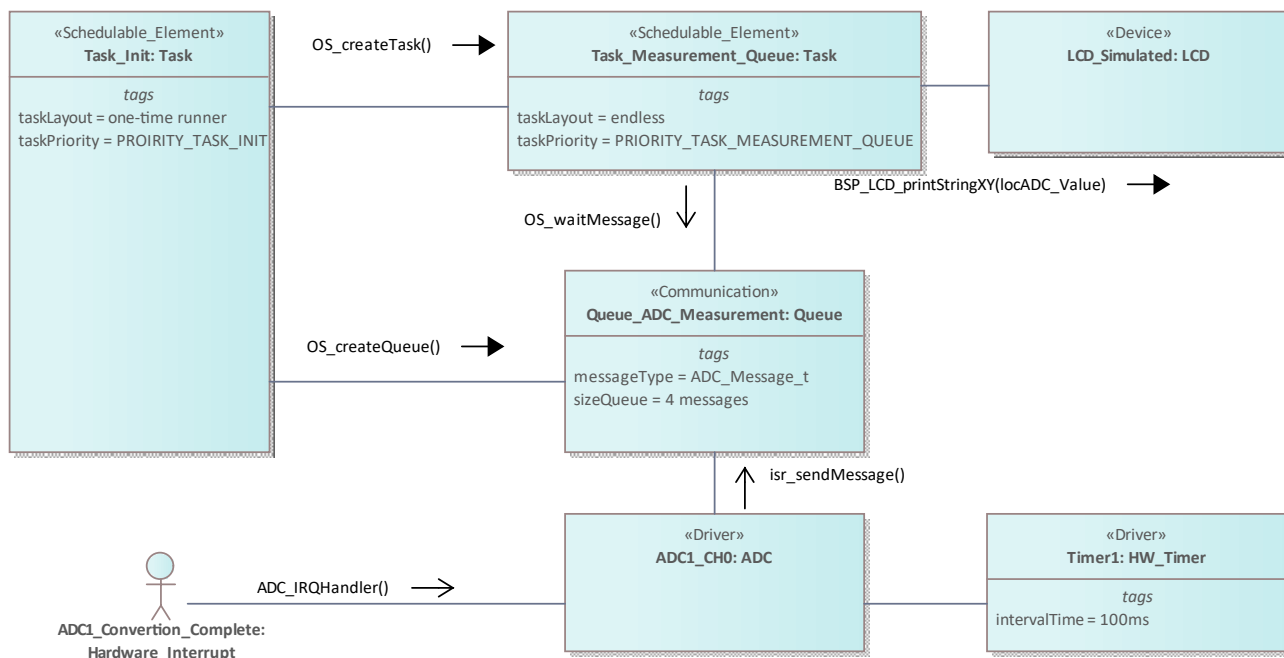Version: 1.0
Author: Thomas Batt

Figure 5: Communication

Time_Management_1 diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Time_Management_1
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Time_Management_1
Package: Software_Run-Time-Architectures
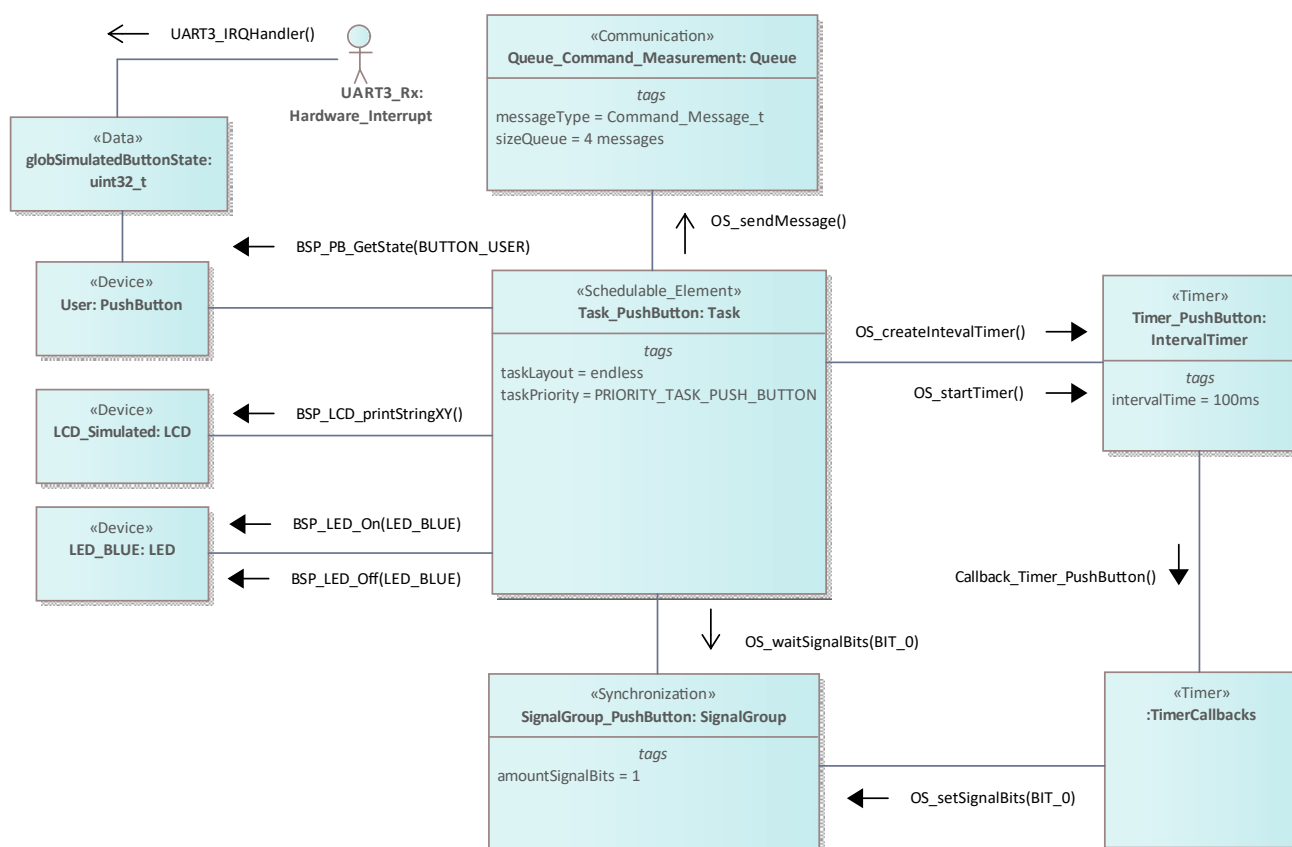Version: 1.0
Author: Thomas Batt

Figure 6: Time_Management_1

Time_Management_2 diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Time_Management_2
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Time_Management_2
Package: Software_Run-Time-Architectures
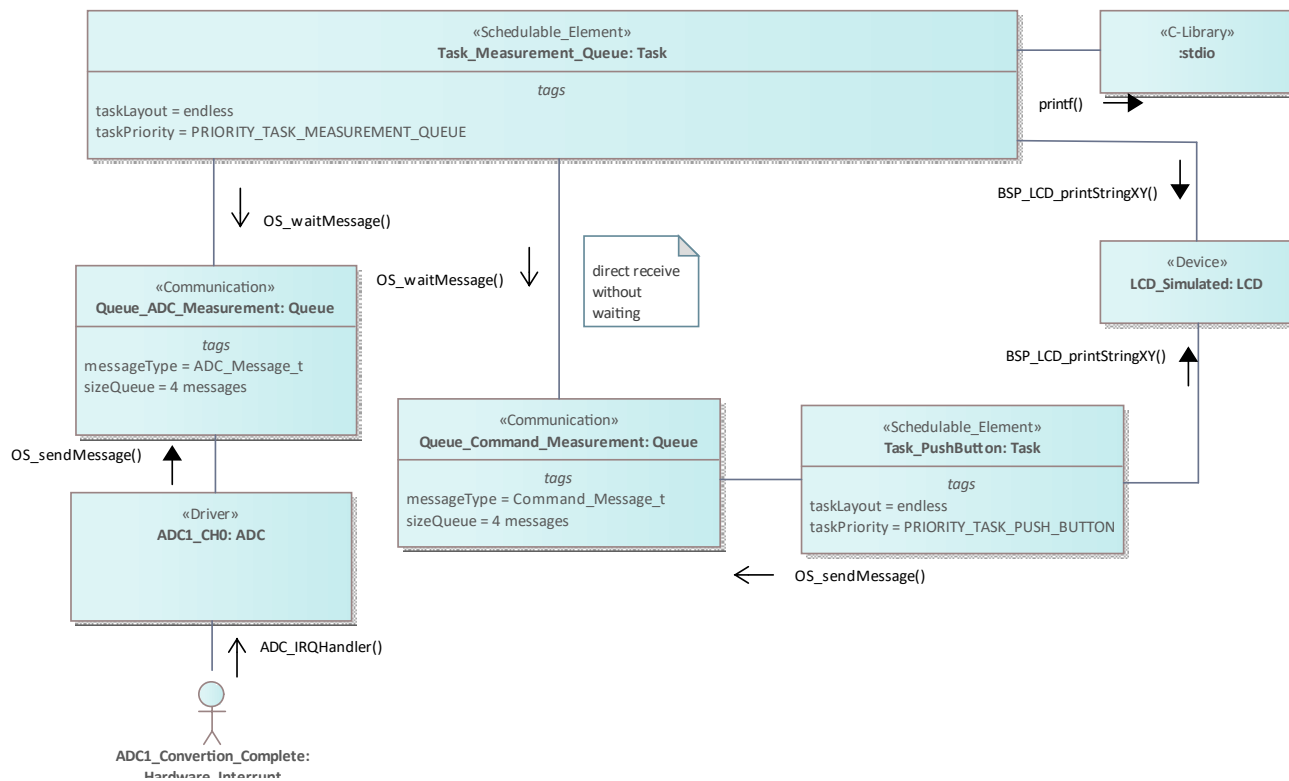Version: 1.0
Author: Thomas Batt

Figure 7: Time_Management_2

Resource_Management diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Resource_Management
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Resource_Management
Package: Software_Run-Time-Architectures
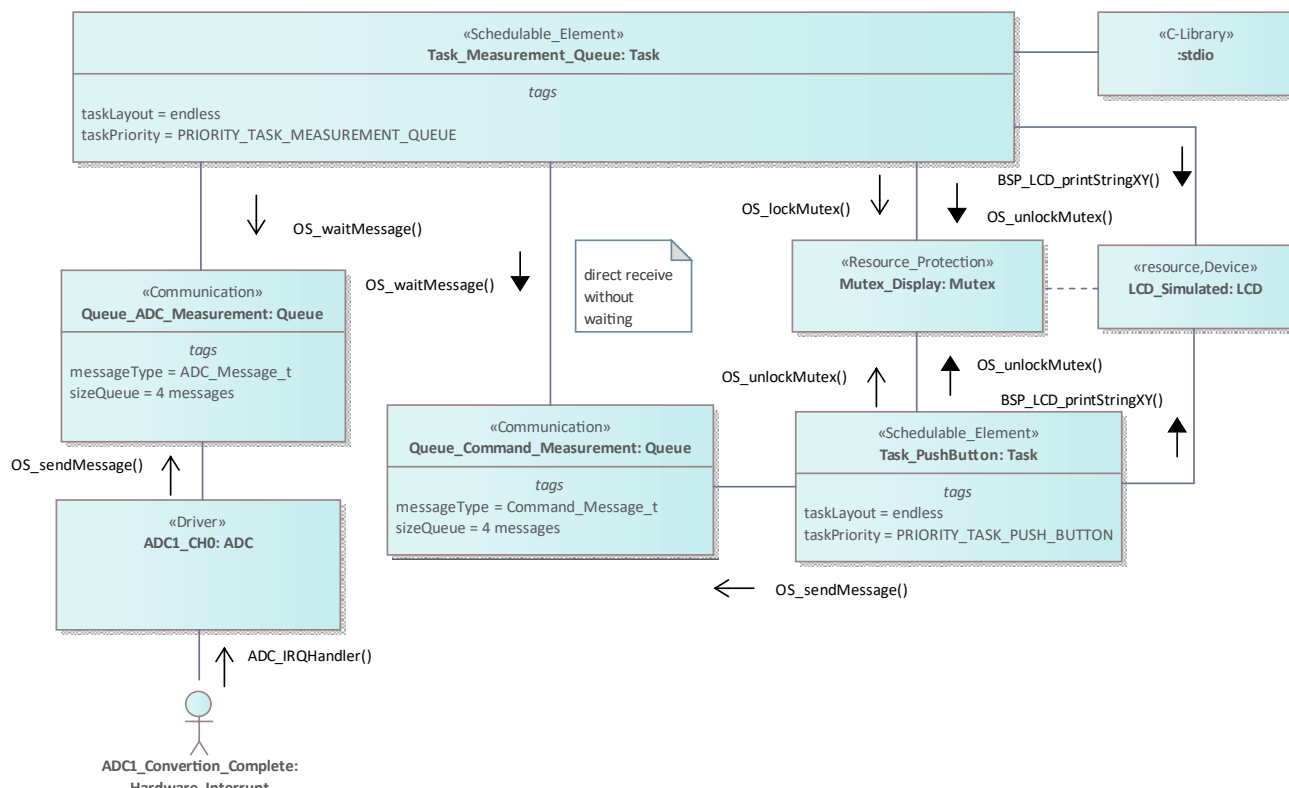Version: 1.0
Author: Thomas Batt

Figure 8: Resource_Management

Init_Time_and_Resource_Management diagram

Communication diagram in package 'Software_Run-Time-Architectures'

Init_Time_and_Resource_Management
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 13.04.2022

Name: Init_Time_and_Resource_Management
Package: Software_Run-Time-Architectures
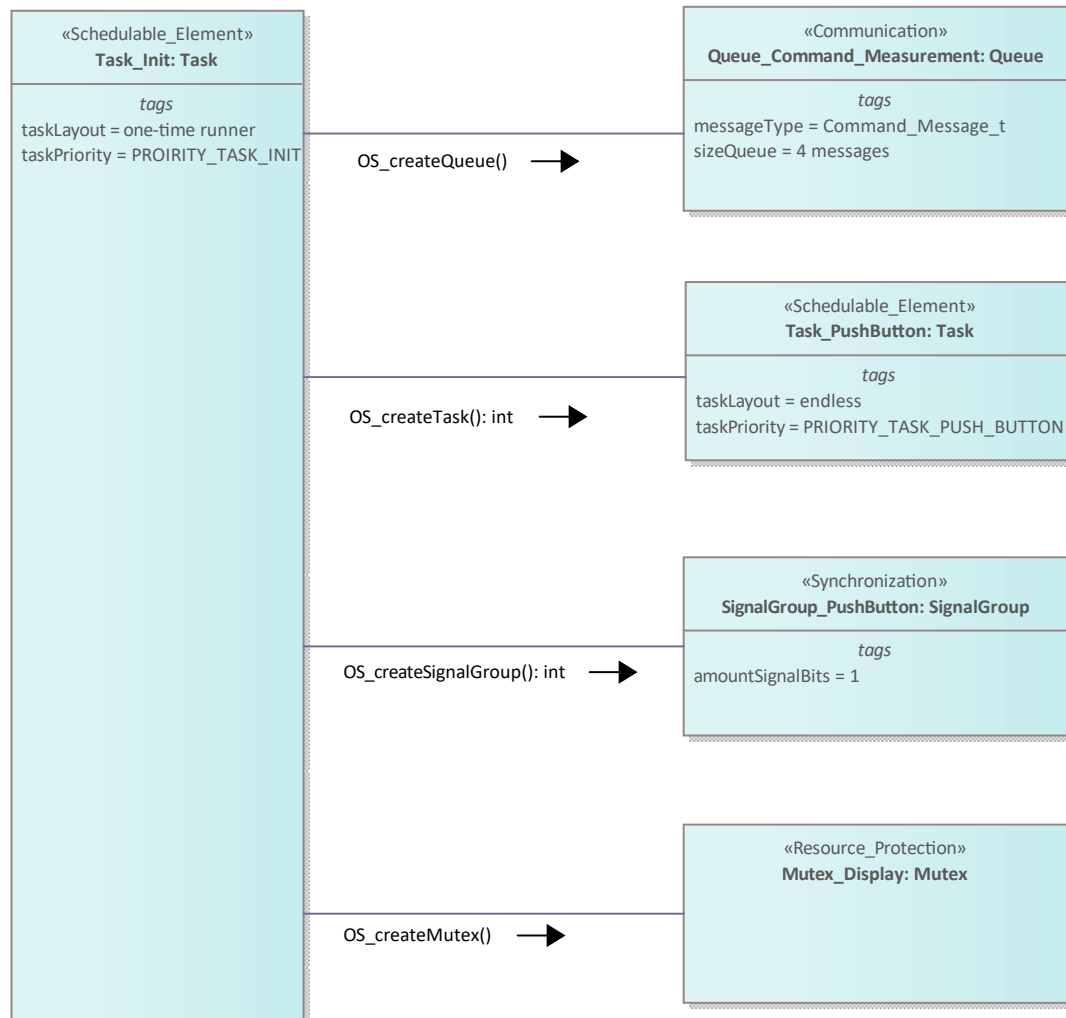Version: 1.0
Author: Thomas Batt

Figure 9: Init_Time_and_Resource_Management

OSAL diagram

Class diagram in package 'OSAL'

OSAL
Version 1.0
Thomas Batt created on 19.02.2016. Last modified 13.04.2022

Name: OSAL
Package: «Software_Layer» OSAL
Version: 1.0
Author: Thomas Batt

«Task» **Task**
+ OS_createTask(): int
+ OS_deleteTask(): int
+ OS_setPriorityTask(): int
+ OS_getPriorityTask(): int
+ OS_getID_Task(): int
+ *functionTask(): int*

«SignalGroup» **SignalGroup**
+ OS_createSignalGroup(): int
+ OS_deleteSignalGroup(): int
+ OS_setSignalBits(): int
+ OS_waitSignalBits(): int

«MessageQueue» **Queue**
+ OS_createQueue(): int
+ OS_deleteQueue(): int
+ OS_sendMessage(): int
+ OS_waitMessage(): int
+ isr_sendMessage(): int

«Timer» **IntervalTimer**
+ OS_createIntevalTimer(): int
+ OS_deleteIntervalTimer(): int
+ OS_startTimer(): int
+ OS_StopTimer(): int

«Resource_Protection» **Mutex**
+ OS_createMutex(): int
+ OS_deleteMutex(): int
+ OS_lockMutex(): int
+ OS_unlockMutex(): int

«Timer» **TimerCallbacks**
+ Callback_Timer_PushButton()

Figure 10: OSAL

Hardware diagram

Class diagram in package 'Hardware'

Hardware
Version 1.0
Thomas Batt created on 26.02.2016. Last modified 14.04.2020

Name: Hardware
Package: «Software_Layer» Hardware
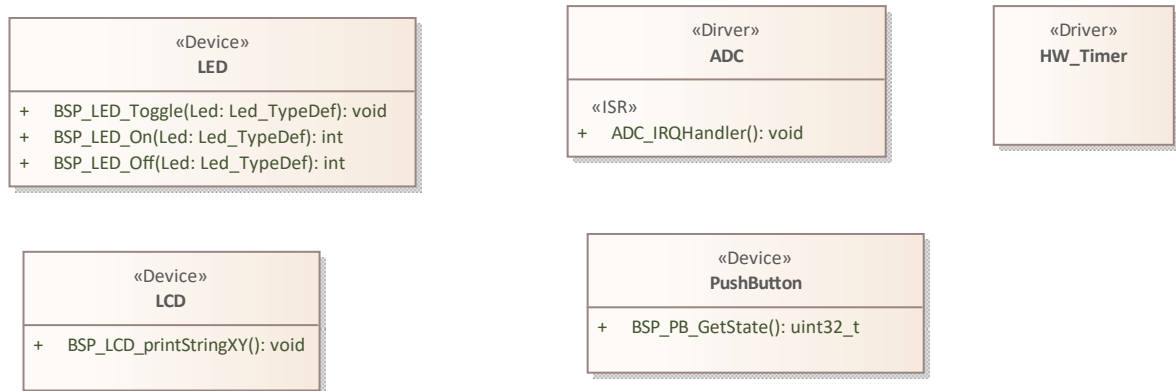Version: 1.0
Author: Thomas Batt

«Device»
LED

+   BSP_LED_Toggle(Led: Led_TypeDef): void
+   BSP_LED_On(Led: Led_TypeDef): int
+   BSP_LED_Off(Led: Led_TypeDef): int

«Dirver»
ADC

«ISR»
+   ADC_IRQHandler(): void

«Driver»
HW_Timer

«Device»
LCD

+   BSP_LCD_printStringXY(): void

«Device»
PushButton

+   BSP_PB_GetState(): uint32_t

Figure 11: Hardware