Diagram: Architecture_Overview

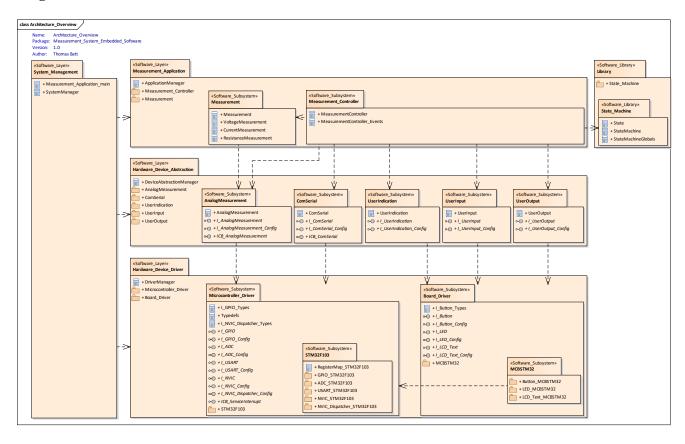


Diagram: Hardware_Device_Abstraction

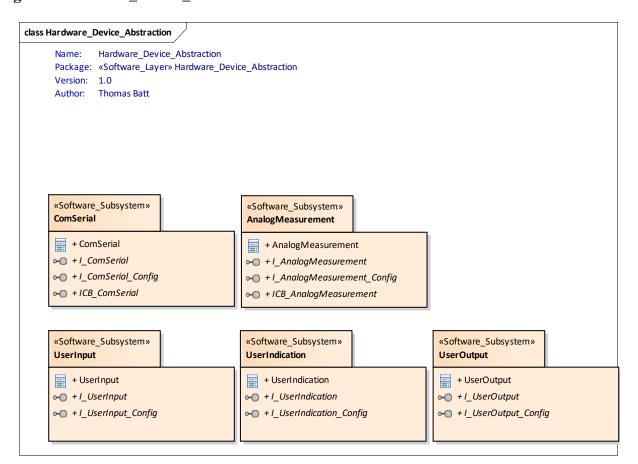


Diagram: Hardware_Device_Abstraction_Manager

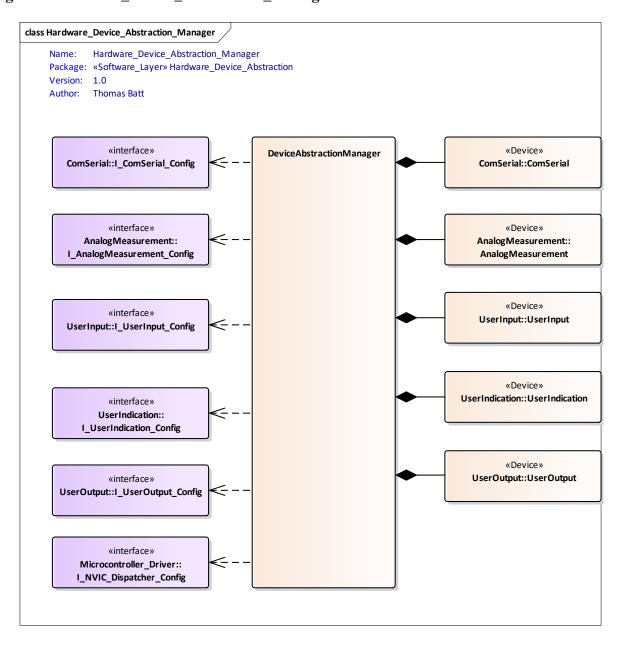


Diagram: Hardware_Device_Abstraction_Manager_extended

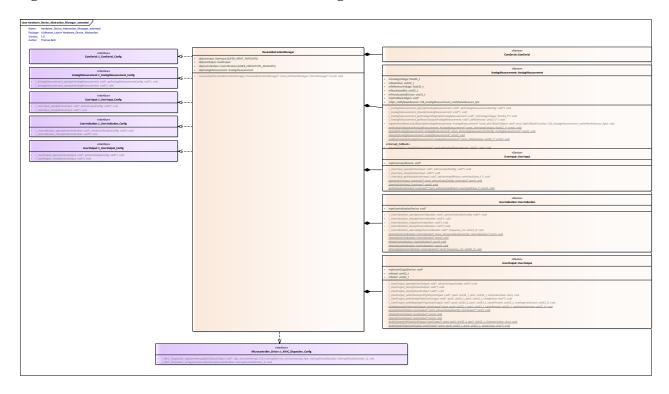


Diagram: ComSerial

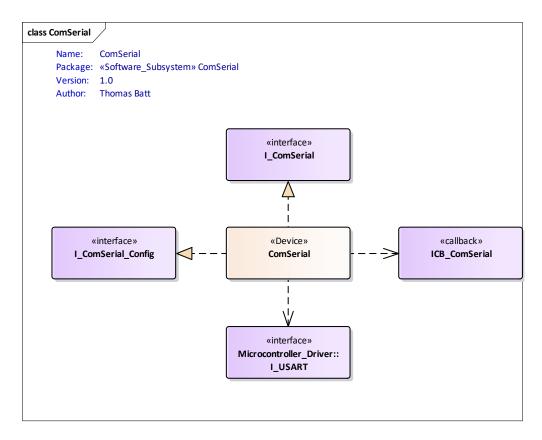


Diagram: UserOutput

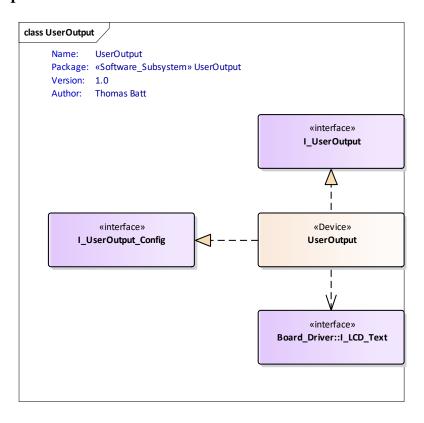


Diagram: UserOutput_extended

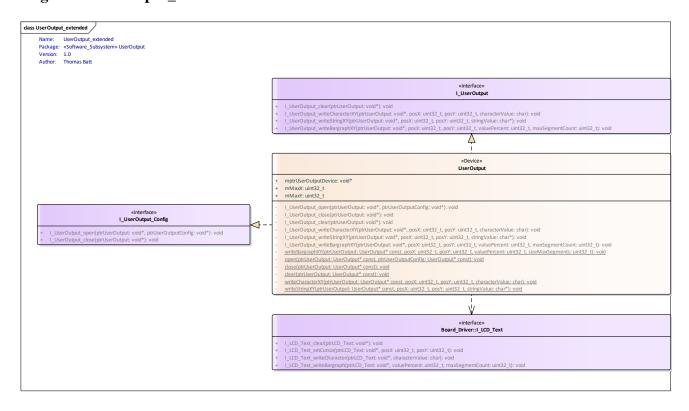


Diagram: UserIndication

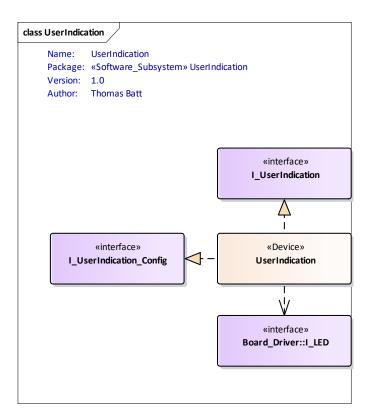


Diagram: UserIndication_extended

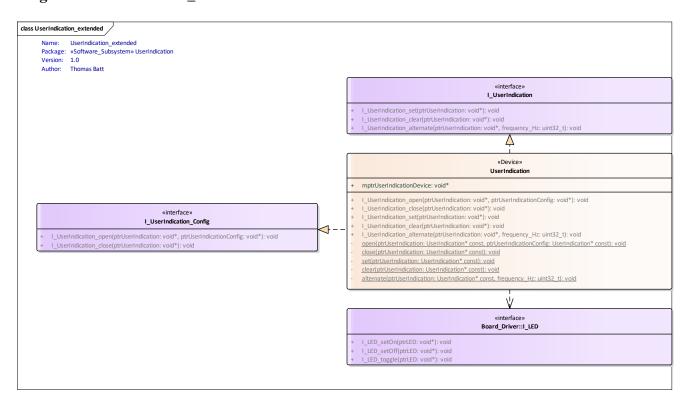


Diagram: UserInput

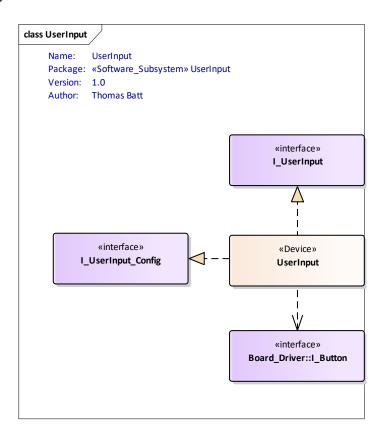


Diagram: UserInput_extended

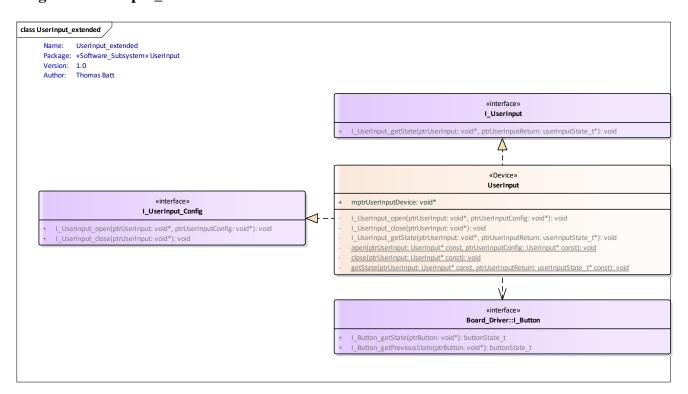
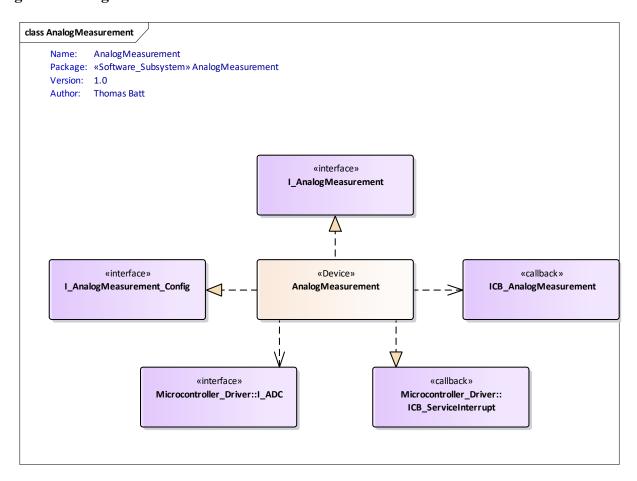


Diagram: AnalogMeasurement



${\bf Diagram: Analog Measurement_extended}$

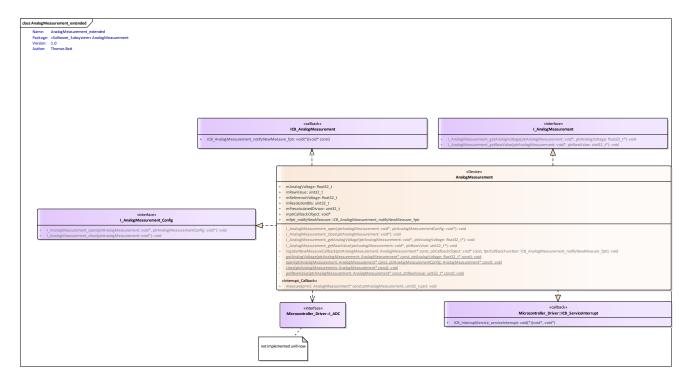


Diagram: Hardware_Device_Driver

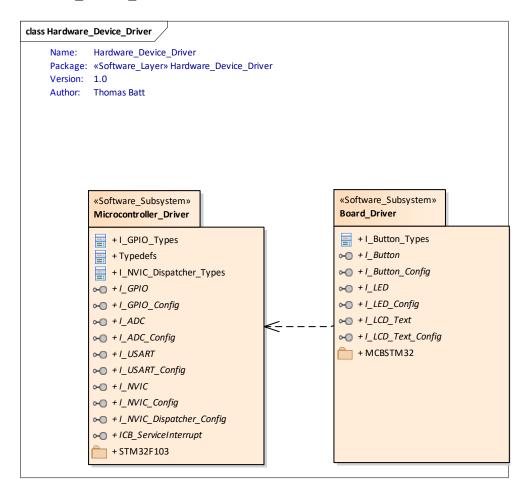


Diagram: Hardware_Device_Driver_Manager

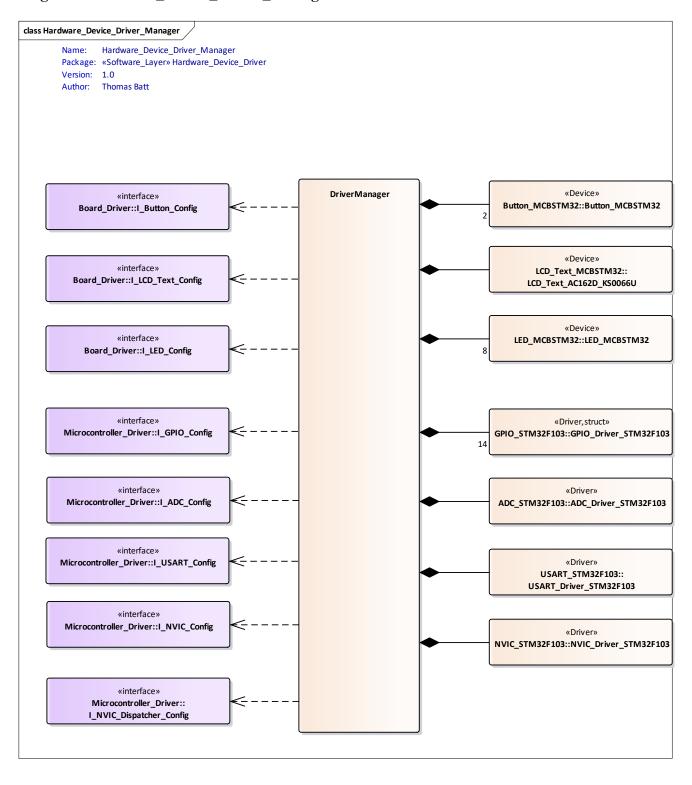


Diagram: Hardware_Device_Driver_Manager_extended

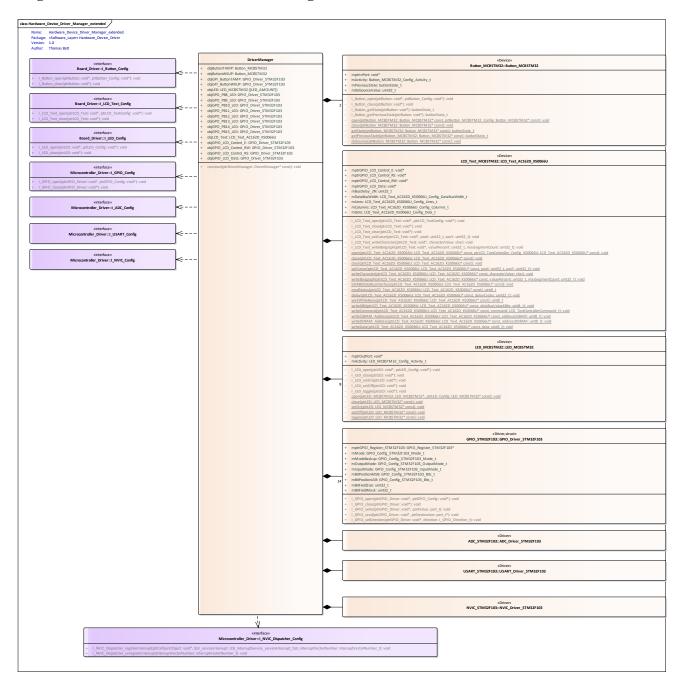


Diagram: Microcontroller_Driver

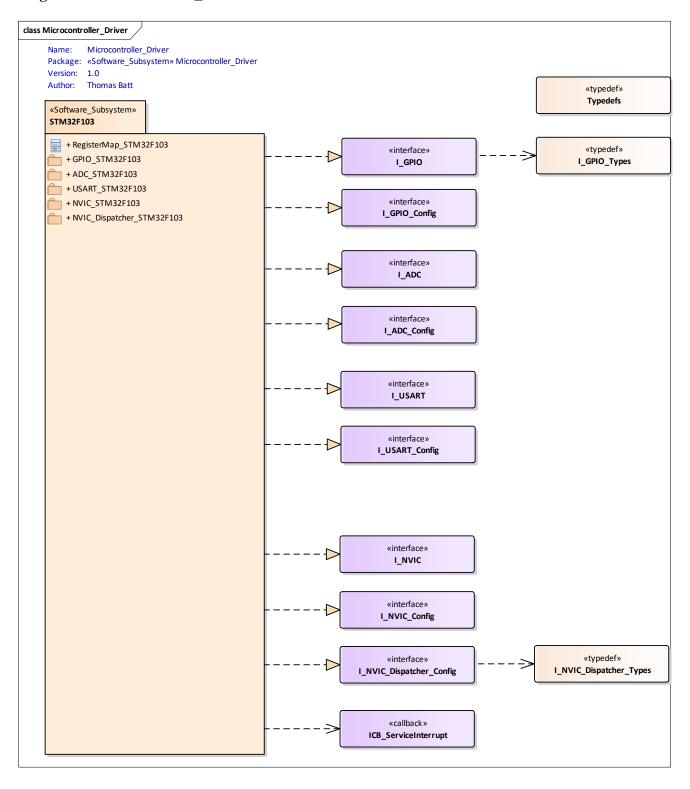


Diagram: GPIO_STM32F103

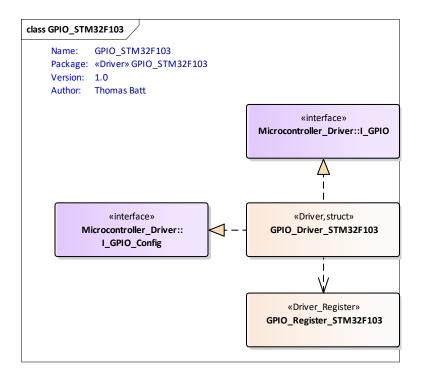


Diagram: GPIO_STM32F103_extended

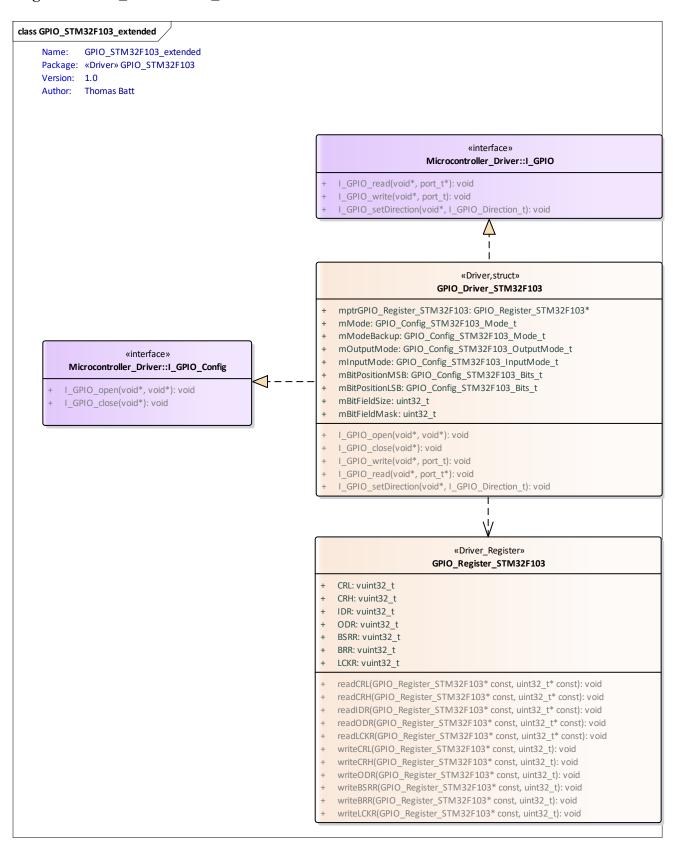


Diagram: ADC_STM32F103

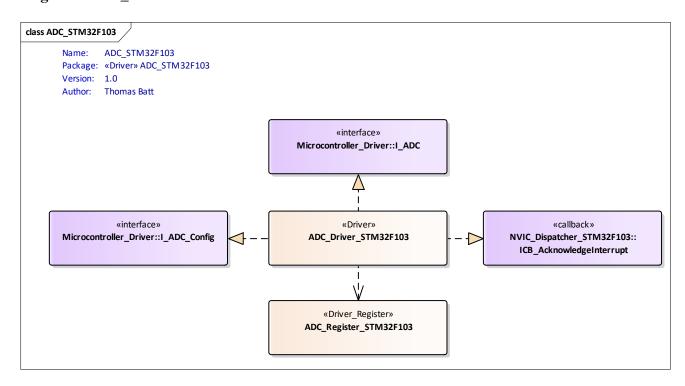


Diagram: USART_STM32F103

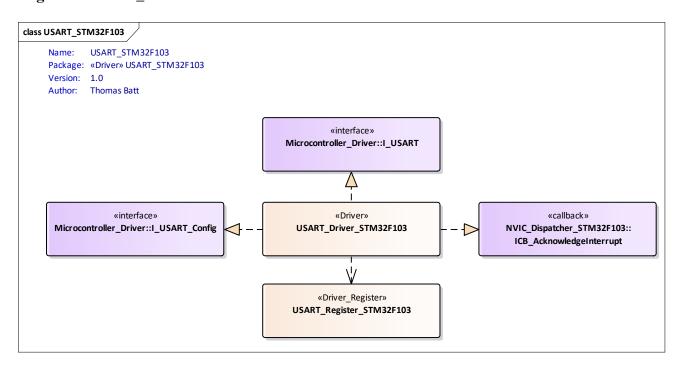


Diagram: InterruptController_STM32F103

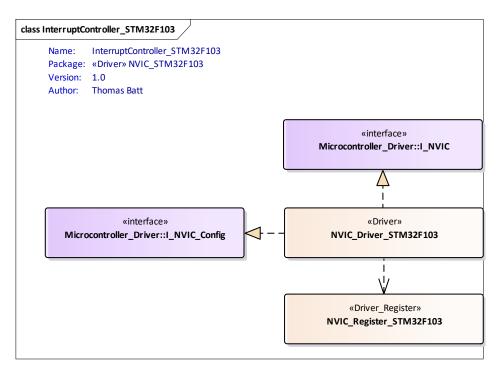


Diagram: NVIC_Dispatcher_STM32F103

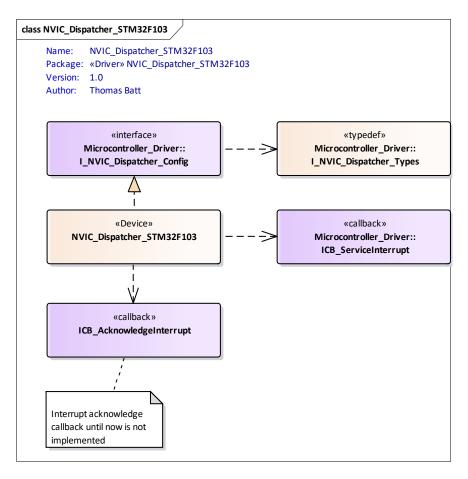


Diagram: NVIC_Dispatcher_STM32F103_extended

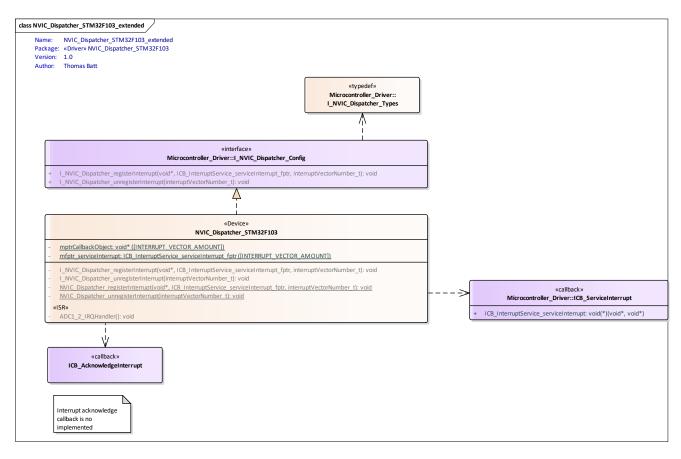


Diagram: Board_Driver

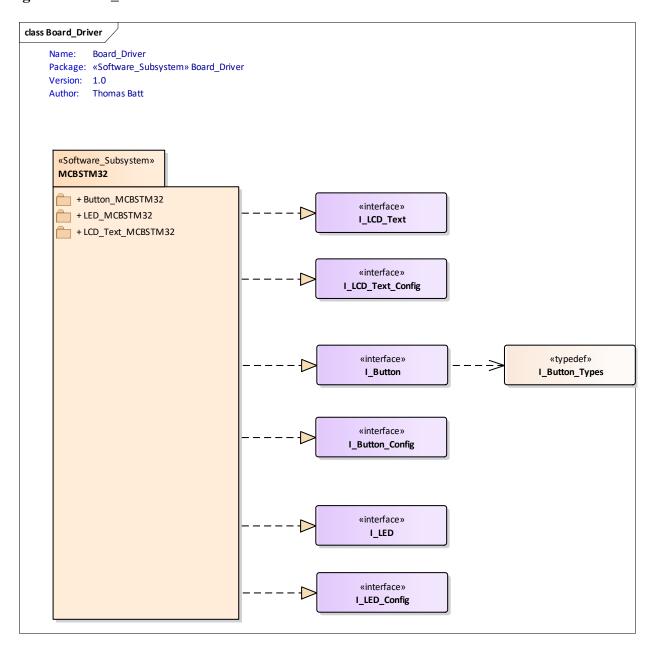


Diagram: LCD_Text_MCBSTM32

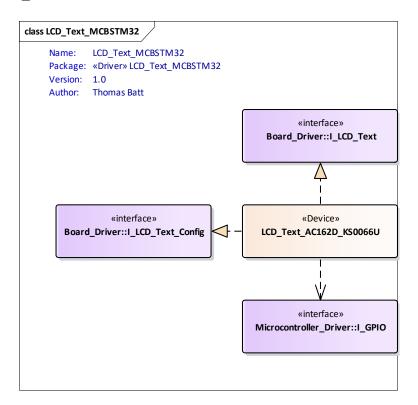


Diagram: LCD_Text_MCBSTM32_extended

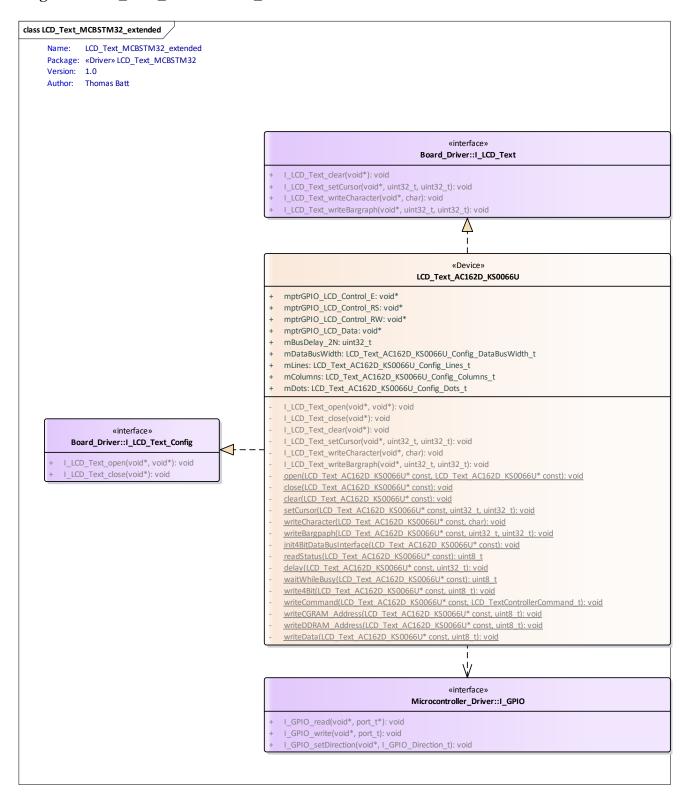


Diagram: LED_MCBSTM32

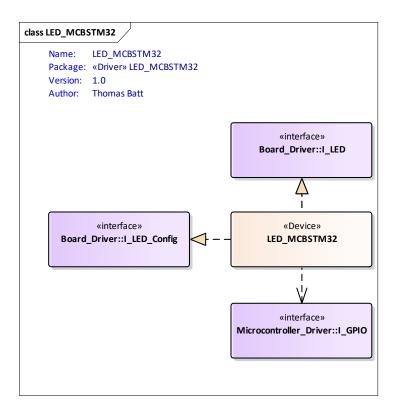


Diagram: LED_MCBSTM32_extended

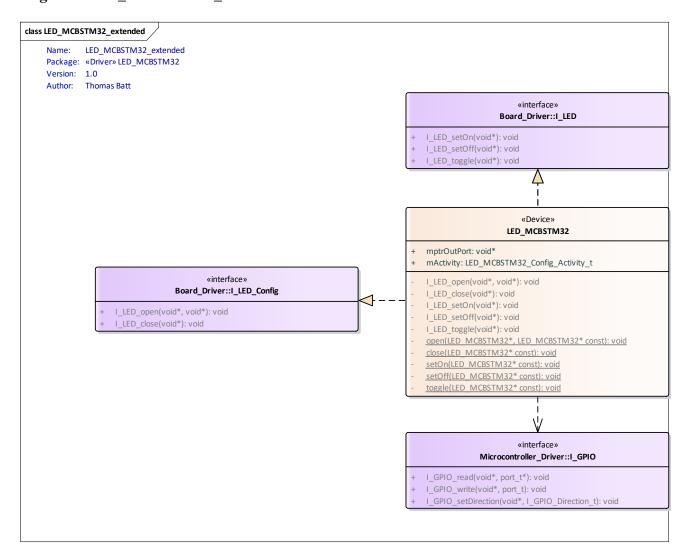


Diagram: Button_MCBSTM32

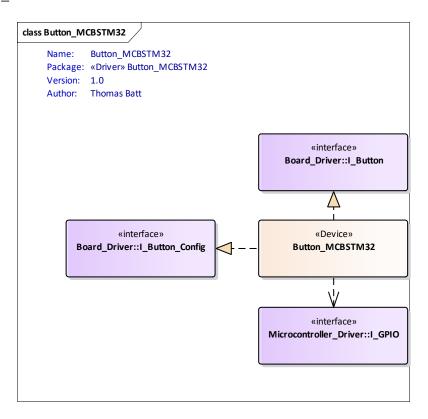


Diagram: Button_MCBSTM32_extended

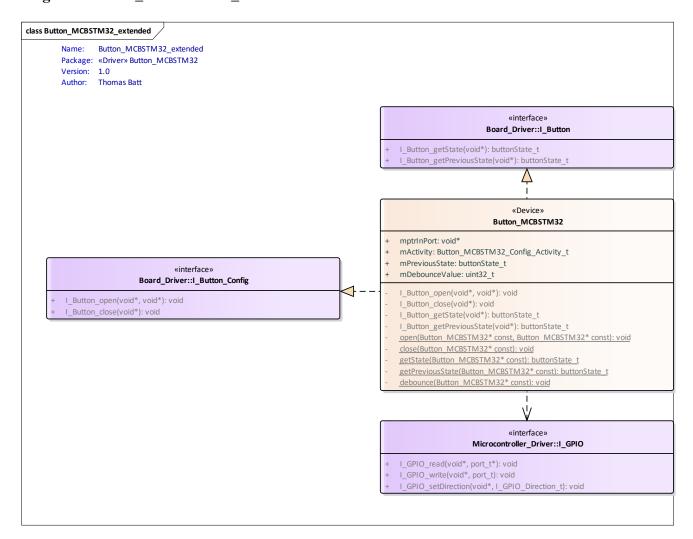


Diagram: System_Management

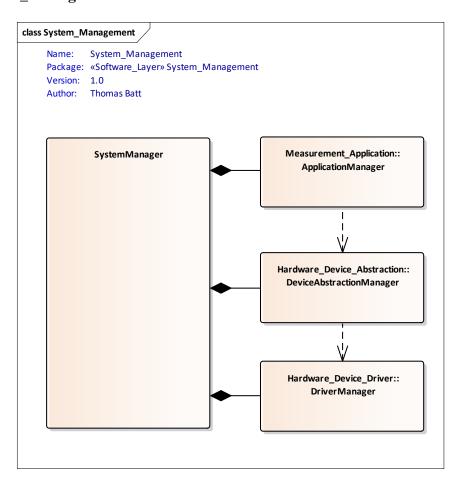


Diagram: System_Management_extended

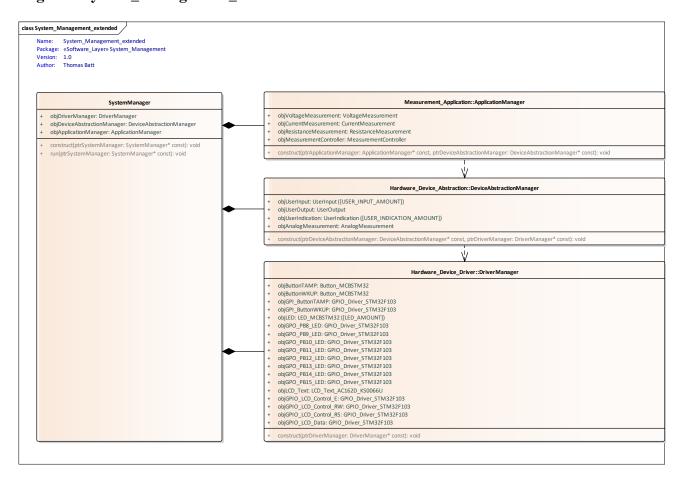


Diagram: Measurement_Application

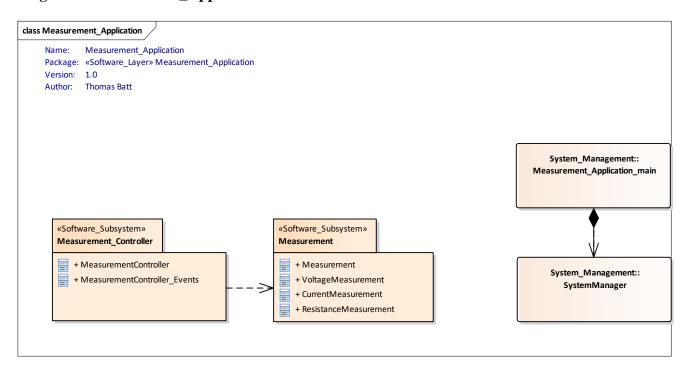
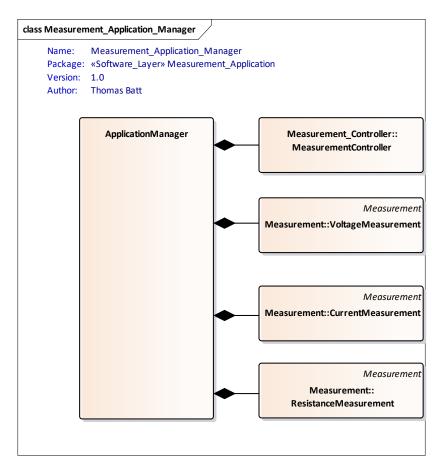


Diagram: Measurement_Application_Manager



${\bf Diagram: Measurement_Application_Manager_extended}$

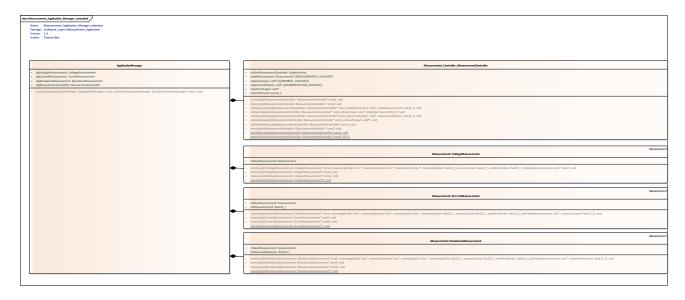


Diagram: Measurement

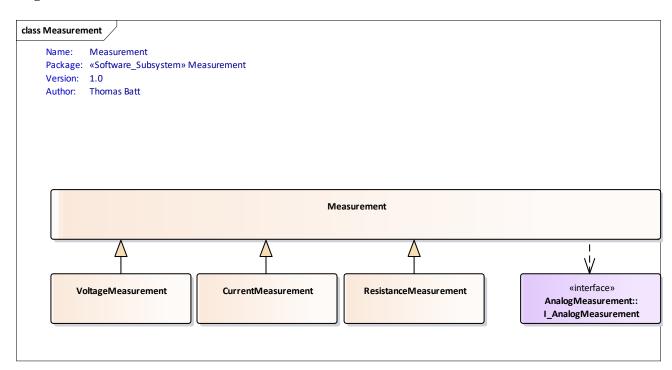


Diagram: Measurement_extended

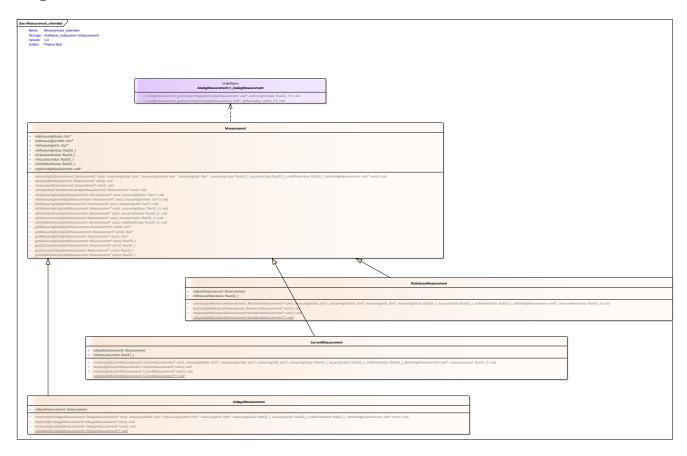


Diagram: Measurement_Controller

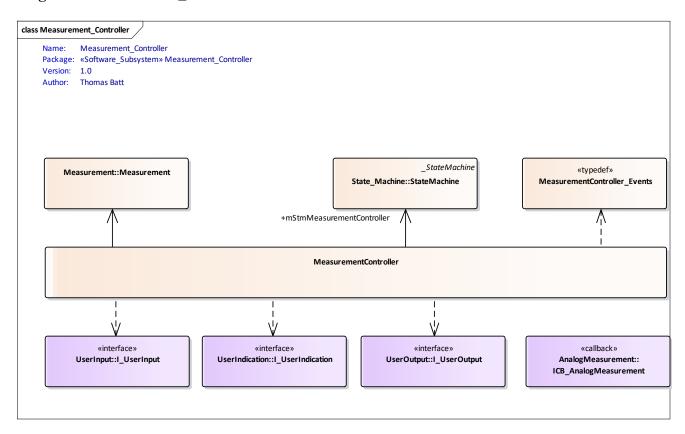


Diagram: Measurement_Controller_extended

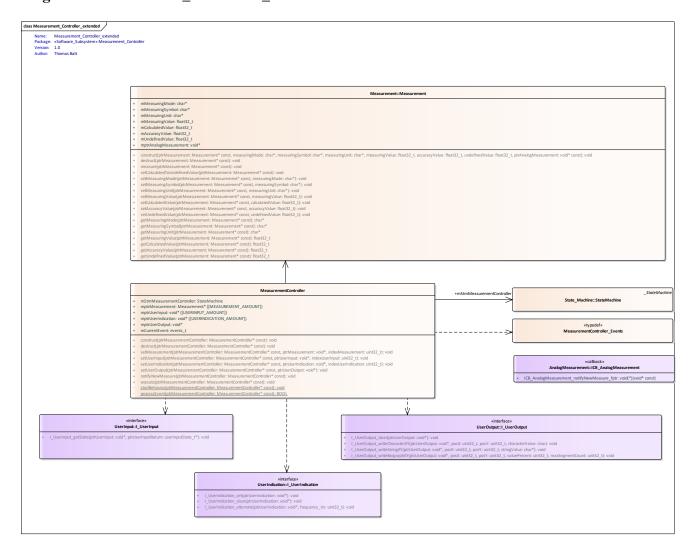


Diagram: StateMachine_MeasurementController

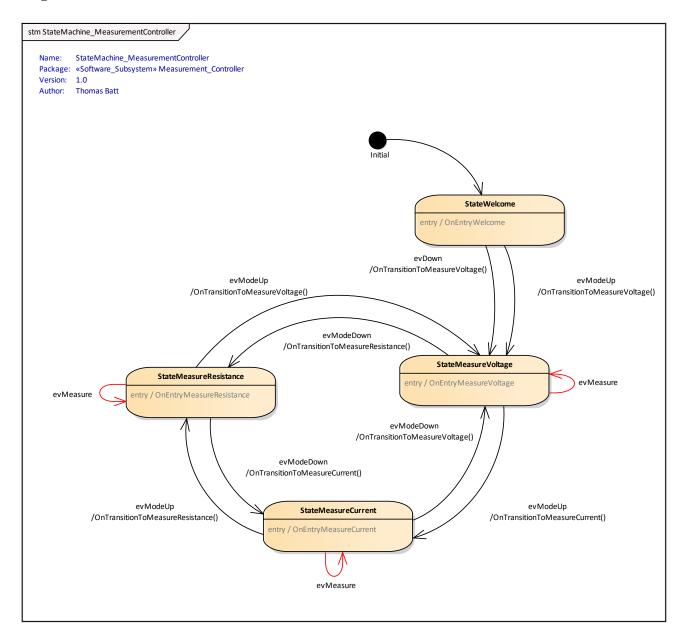


Diagram: Library

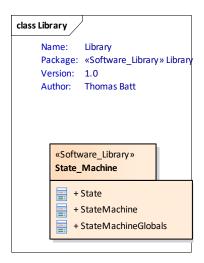


Diagram: State_Machine

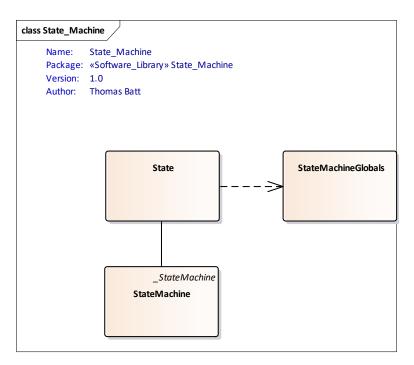


Diagram: Exercise[1.1]

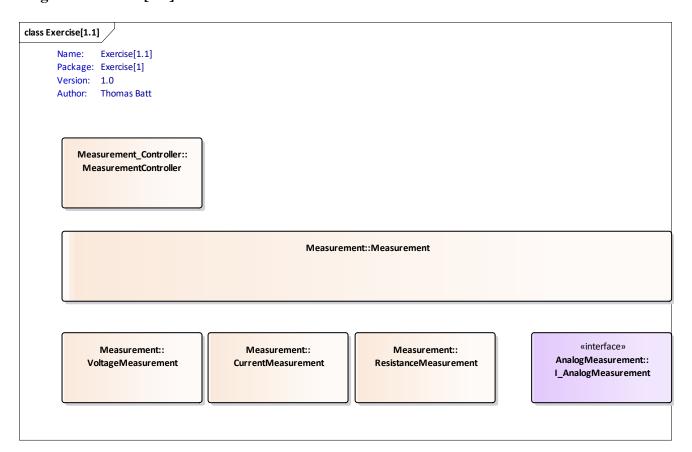


Diagram: Exercise[1.2]

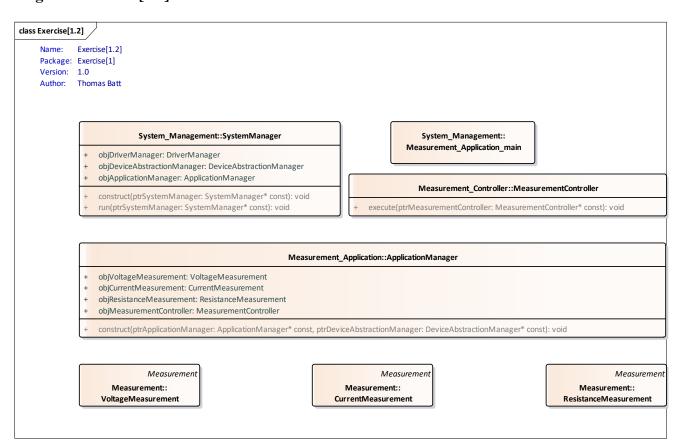


Diagram: Exercise[1.3]

class Exercise[1.3]

Name: Exercise[1.3]
Package: Exercise[1]
Version: 1.0
Author: Thomas Batt

Diagram: Exercise[2.1]

class Exercise[2.1]

Exercise[2.1] Name: Package: Exercise[2] Version: 1.0 Author: **Thomas Batt**

Measurement_Controller::MeasurementController

- + mStmMeasurementController: StateMachine
- mptrMeasurement: Measurement* ([MEASUREMENT_AMOUNT])
- + mptrUserInput: void* ([USERINPUT_AMOUNT])
- + mptrUserIndication: void* ([USERINDICATION_AMOUNT])
- mptrUserOutput: void*
- + mCurrentEvent: events_t
- + construct(MeasurementController* const): void
- destruct(MeasurementController* const): void
- setMeasurement(MeasurementController* const, void*, uint32_t): void
- + setUserInput(MeasurementController* const, void*, uint32_t): void
- + setUserIndication(MeasurementController* const, void*, uint32_t): void
- + setUserOutput(MeasurementController* const, void*): void
- + notifyNewMeasure(MeasurementController* const): void
- execute(MeasurementController* const): void
- <u>startBehavior(MeasurementController* const): void</u>

«typedef» Measurement_Controller:: MeasurementController_Events

«Software_Library» State_Machine

+ State

+ StateMachine

+ StateMachineGlobals

(from Library)

Measurement::Measurement

«interface» UserInput::I_UserInput

«interface» UserOutput::I_UserOutput

«interface» AnalogMeasurement:: I_AnalogMeasurement

«interface» UserIndication::I_UserIndication

Diagram: Exercise[2.2]

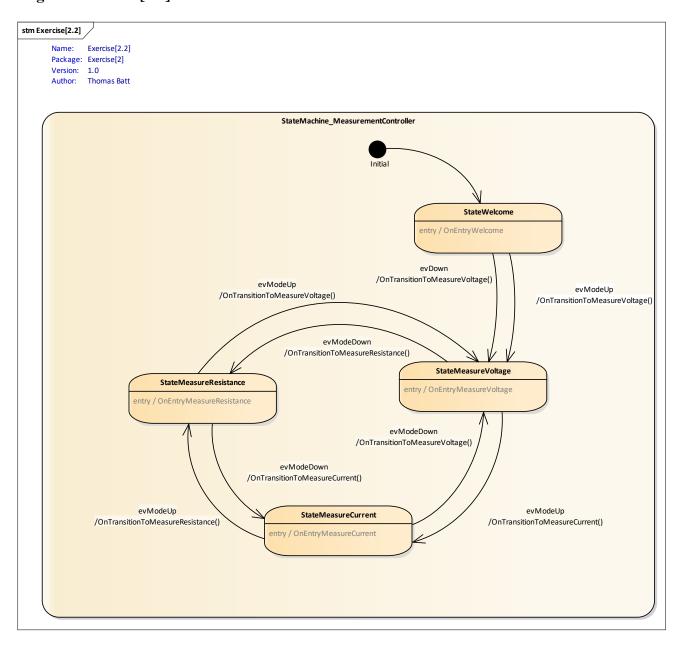


Diagram: Exercise[3.1]

class Exercise[3.1] Exercise[3.1] Package: Exercise[3] Version: 1.0 Author: Thomas Batt Measurement_Controller::MeasurementController notifyNewMeasure(MeasurementController* const): void «interface» «interface» «callback» AnalogMeasurement:: AnalogMeasurement:: $Analog Measurement :: ICB_Analog Measurement\\$ I_AnalogMeasurement_Config I_AnalogMeasurement ICB_AnalogMeasurement_notifyNewMeasure_fptr: void(*)(void* const) AnalogMeasurement::AnalogMeasurement registerNewMeasureCallback(AnalogMeasurement* const, void* const, ICB_AnalogMeasurement_notifyNewMeasure_fptr): void «Interrupt_Callback» measure(AnalogMeasurement* const ptrAnalogMeasurement, uint32_t par): void «interface» «callback» Microcontroller_Driver::I_ADC Microcontroller_Driver:: ICB_ServiceInterrupt

Diagram: Exercise[3.2]

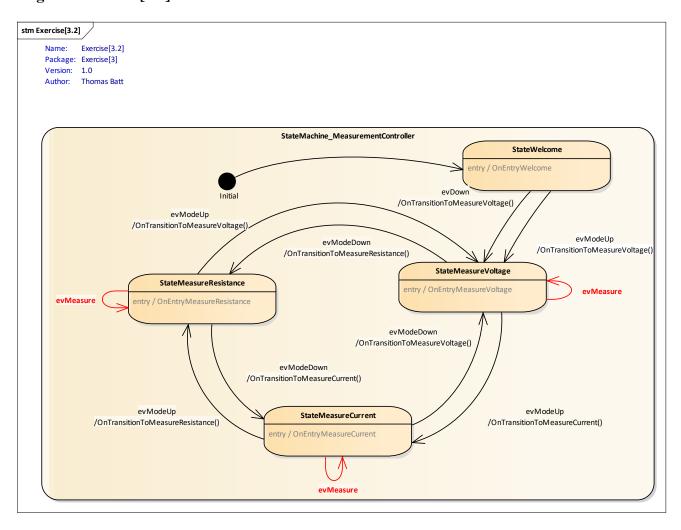


Diagram: Exercise[3.2]

class Exercise[3.2]

Name: Exercise[3.2]
Package: Exercise[3]
Version: 1.0
Author: Thomas Batt

Measurement_Application::ApplicationManager

- + objVoltageMeasurement: VoltageMeasurement
- + objCurrentMeasurement: CurrentMeasurement
- + objResistanceMeasurement: ResistanceMeasurement
- + objMeasurementController: MeasurementController
- + construct(ApplicationManager* const, DeviceAbstractionManager* const): void

Hardware_Device_Abstraction::DeviceAbstractionManager

- + objUserInput: UserInput ([USER_INPUT_AMOUNT])
- + objUserOutput: UserOutput
- + objUserIndication: UserIndication ([USER_INDICATION_AMOUNT])
- + objAnalogMeasurement: AnalogMeasurement
- + construct(DeviceAbstractionManager* const, DriverManager* const): void

Measurement_Controller::MeasurementController

+ notifyNewMeasure(MeasurementController* const): void

«Device»

AnalogMeasurement::AnalogMeasurement

+ registerNewMeasureCallback(AnalogMeasurement* const, void* const, ICB_AnalogMeasurement_notifyNewMeasure_fptr): void

«Interrupt_Callback»

+ measure(AnalogMeasurement* const ptrAnalogMeasurement, uint32_t par): void

Diagram: Exercise[4.1]

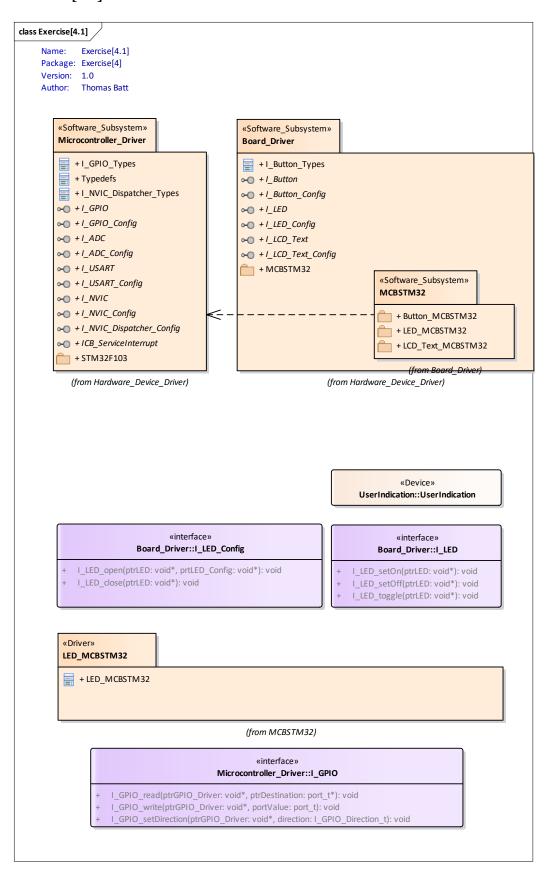


Diagram: Exercise[4.2]

