# STATIC ARCHITECTURE

**Team Members:** Mohanad Fawzy

Marcelle Samir

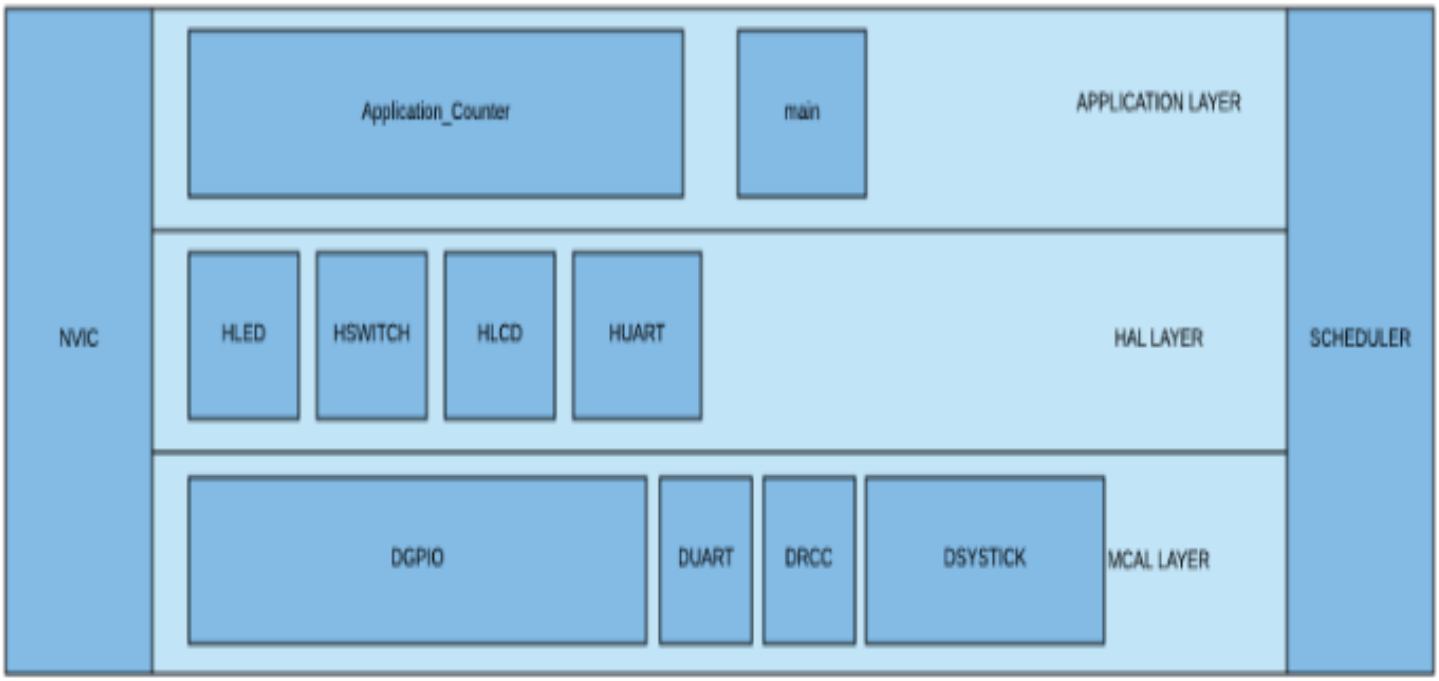
Mostafa Ramadan

Marina Medhat

Mostafa Nader

MARCH 23, 2020

## Static Architecture Diagram




| DRIVER | DRCC | DESCRIPTION |
|---|---|---|
| API's | uint_8t RCC_SetClkStatus (uint_32t Clk,uint_8t Status); | Enables the ready flags of either HSI, HSE or PLL |
| | uint_8t RCC_SetSystemClk (uint_32t Clk); | Sets HSI, HSE or PLL as the system Clock |
| | uint_8t RCC_SetPLLConfig (uint_32t Src, uint_8t Mul); | Configures the source to the PLL and the multiplier's Value |
| | uint_8t RCC_SetPriephralStatus (uint_32t Peripheral,uint_8t Status); | Enables the clock of any peripheral from AHB, APB1 or APB2 Bus |
| | uint_8t RCC_SetBusPrescale (uint_32t Bus,uint_8t Prescaler); | Set the prescaler value of either AHB, APB1 or APB2 Bus |
| | uint_8t RCC_GetBusClock (uint_32t Bus,uint_32t *Clk); | Gets the Clock system of either AHB, APB1 or APB2 Bus after prescaling |

| DRIVER | DGPIO | DESCRIPTION |
|---|---|---|
| API's | uint_8t GPIO_Config (GPIO_t * Pins ); | Configures one or multiple pins to one or multiple configuration modes at the same port |
| | uint_8t GPIO_Writee(Port_t *Port,uint_16t Pins ,uint_8t State); | Setting or clearing values to one or multiple pins at the same port |
| | uint_8t GPIO_ReadPort(Port_t *Port,uint_16t * Value); | Gets the value of an entire port |
| | uint_8t GPIO_ReadPin (Port_t *Port,uint_16t Pin,uint_8t * Value); | Gets the value of a single pin at a specific port |

| DRIVER | DSYSTICK | DESCRIPTION |
|---|---|---|
| API's | uint_8t SYSTICK_Init (void); | Initializes the interrupt mode and the prescaler of the systick timer |
| | uint_8t SYSTICK_Start (void); | Enables the systick timer to start counting |
| | uint_8t SYSTICK_Stop (void); | Disables the systick timer to stop counting |
| | uint_8t SYSTICK_SetCallback (systickcbf_t Cbf); | Call the function that the user sends when the handler executes |
| | uint_8t SYSTICK_SetTime (uint_32t Ttimeus,uint_32t Clk); | Receives time from the user in microseconds to star counting it |
| | uint_8t SYSTICK_SetPrescale (uint_32t Prescaler); | Sets the prescaler of the systick timer to 1 or 8 |

| DRIVER | DUART | DESCRIPTION |
|---|---|---|
| API's | uint_8t UART_Init(void); | Enables the UART and initializes the configuration parameters of the UART peripheral(Data length, parity bit, # of stop bits and baud rate) and enables the TC,RXNE interrupts |
| | uint_8t UART_Send(uint_8t *Buffer, uint_16t Length); | Send a stream of bytes on the transmission line |
| | uint_8t UART_Receive(uint_8t *Buffer, uint_16t Length); | Receives a stream of bytes on the receiving line |
| | uint_8t UART_Config(uint_32t BaudRate,uint_32t ParityBits,uint_32t DataSize,uint_32t StopBits); | Configures the configuration parameters of the UART peripheral(Data length, parity bit, # of stop bits and baud rate) |
| | uint_8t UART_SetTxCbf(TxCbf_t TxCbf); | Function called when the transmission handler is executed |
| | uint_8t UART_SetRxCbf(RxCbf_t RxCbf); | Function called when the receiving handler is executed |

| DRIVER | HLED | DESCRIPTION |
|---|---|---|
| API's | uint_8t HLED_Init(void); | Initializes the configurations of the LEDs according to the values in the configuration file |
| | uint_8t HLED_SetLedState(uint_8t Led_Number,uint_8t Led_State); | Setting or Clearing each Led according to the choice of the user in the configuration file |

| DRIVER | HSWITCH | DESCRIPTION |
|---|---|---|
| API's | uint_8t Switch_Init (void); | Initializes the configurations of the Switches according to the values in the configuration file |
| | uint_8t Switch_GetSwitchState(uint_8t SwitchNum,uint_8t * State); | Gets the values of the switch (High or Low ) after checking of five consecutive values |

| DRIVER | HLCD | DESCRIPTION |
|---|---|---|
| API's | uint_8t LCD_Init (void); | Initializes the LCD according to the Datasheet |
| | uint_8t LCD_WriteData (const uint_8t *data,uint_8t DataLength); | Writing a stream of data bytes on the LCD screen |
| | uint_8t LCD_ClearLCD (void); | Clears the LCD screen |

| DRIVER | HUART | DESCRIPTION |
|---|---|---|
| API's | uint_8t UART_Init(void); | Initializes the RX & TX Pins configurations and gets the clock's frequency |
| | uint_8t UART_Send(uint_8t *Buffer, uint_16t Length); | Calls the send function the UART Driver |
| | uint_8t UART_Receive(uint_8t *Buffer, uint_16t Length); | Calls the Receiving function the UART Driver |
| | `uint_8t HUART_Config(uint_32t BaudRate,uint_32t ParityBits,uint_32t DataSize,uint_32t StopBits);` | Send the Baud rate's **mantissa** and **fraction**, the **StopBits** configuration, the **ParityBit** configuration and the **DataLength** configuration to the MCAL driver to configure it |
| | `uint_8t HUART_SetTxCbf(TxCbf_t TxCbf);` | Calls the Transmitter call back function in the MCAL driver |
| | `uint_8t UART_SetRxCbf(RxCbf_t RxCbf);` | Calls the Receiver call back function in the MCAL driver |

| DRIVER | NVIC | DESCRIPTION |
|---|---|---|
| API's | uint_8t DNVIC_EnableIRQ(uint_8t IRQn); | Enables the bit of any peripheral from the 240 external interrupts |
| | uint_8t DNVIC_DisableIRQ(uint_8t IRQn); | Disables the bit of any peripheral from the 240 external interrupts |
| | uint_8t DNVIC_SetPendingIRQ (uint_8t IRQn); | Sets the Pending flag of any periphers from the 240 external interrupts by software |
| | uint_8t DNVIC_ClearPendingIRQ (uint_8t IRQn); | Clears the Pending flag of any periphers from the 240 external interrupts by software |
| | uint_8t DNVIC_GetPendingIRQ (uint_8t IRQn, uint_8t *Val); | Gets the Pending flag value of any periphers from the 240 external interrupts |
| | uint_8t DNVIC_GetActive (uint_8t IRQn, uint_8t *Val); | Gets the Active flag Value of any periphers from the 240 external interrupts |
| | uint_8t DNVIC_SetPriorityGrouping(uint_32t priority_grouping); | Divides the 4 bits available in ST into preemption and subgroup bits |
| | uint_8t DNVIC_SetPriority (uint_8t IRQn, uint_8t priority); | Sets the priority of any of the 240 external interrupt from 0->0x0F |
| | uint_8t DNVIC_GetPriority (uint_8t IRQn, uint_8t *priority); | Gets the priority value of any of the 240 external interrupt |
| | void DNVIC_voidDisableAllPeripherals(void); | Disables all the interrupts |
| | void DNVIC_voidEnableAllPeripherals(void); | Enables all the interrupts |
| | void DNVIC_voidDisableAllFaults(void); | Disables all the interrupts & Faults |
| | void DNVIC_voidEnableAllFaults(void); | Enables all the interrupts & Faults |
| | void DNVIC_voidSetBASEPRI(uint_8t priority); | Disable the interrupts of a specific priority or higher |

| DRIVER | SCHEDULER | DESCRIPTION |
|---|---|---|
| API's | uint_8t Sched_Init(void); | Initializes the systick timer with the tick time and initializes the tasks |
| | uint_8t Sched_Start(void); | Starts the systick timer |
| | void Sched_Suspend(void); | Suspends a running task |