Small Operating System Design

Prepared By:

Moustafa Abdelrahim

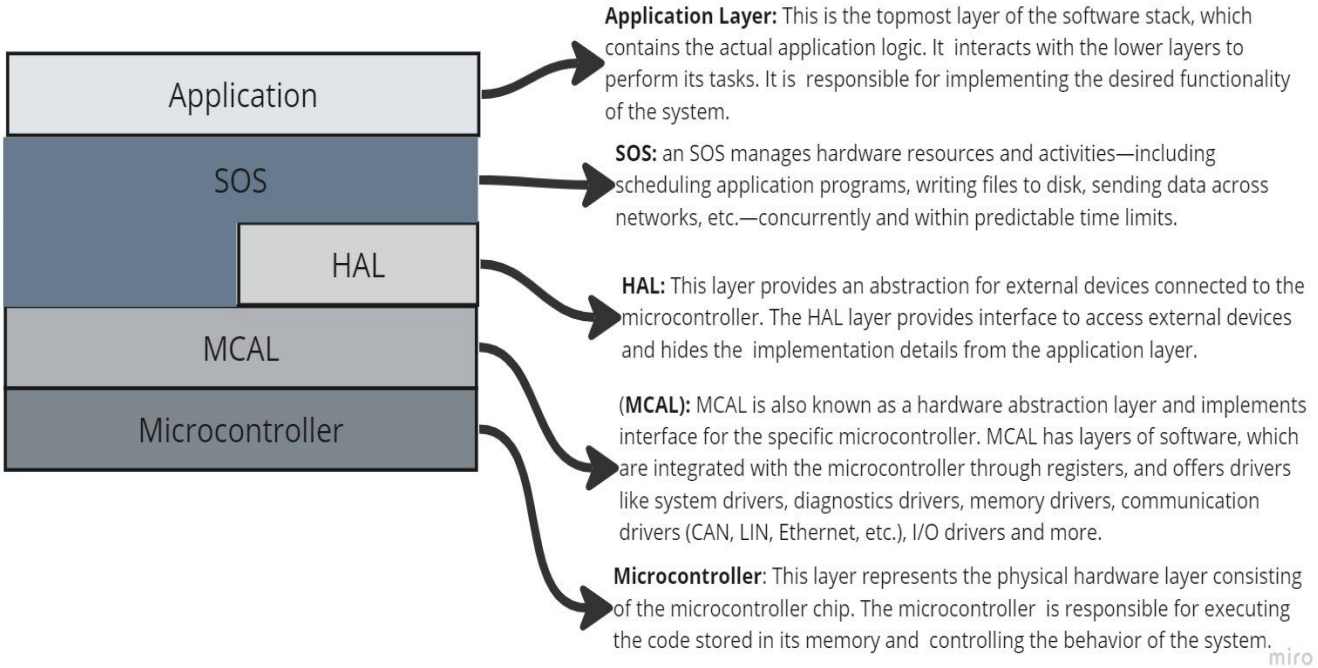
Table Of Content:

1. Introduction:

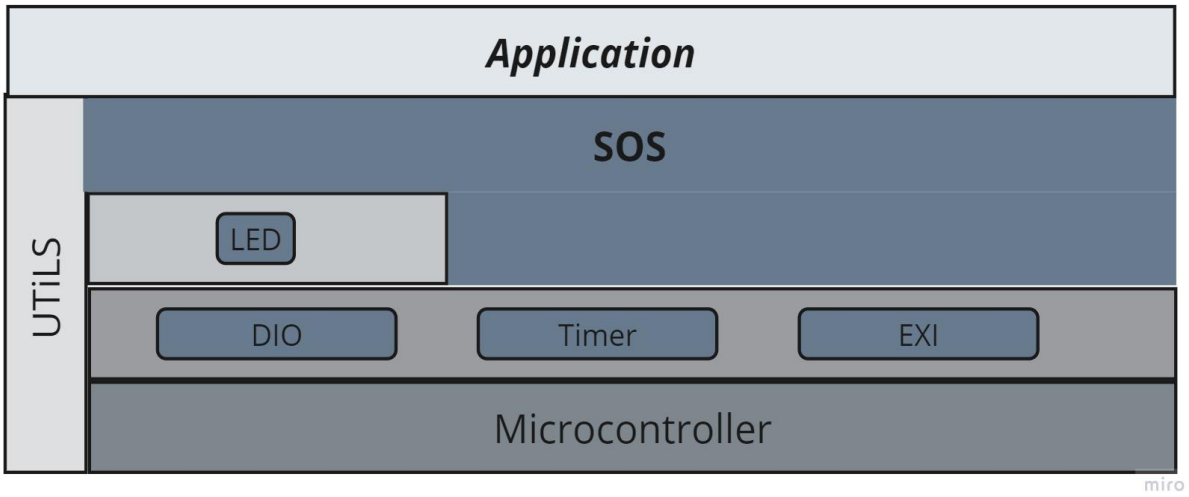
A Small operating system is a software which handles the computer's functionality like scheduling, input/output operation, resource allocation, file system manipulation, etc. and it acts as an interface between the user and hardware. This Module is designed to execute multi tasks based upon their priority and periodicity in a non preemptive manner

2. High Level Design

2.1. Layered Architecture:



2.2. Modules Description:



-**DIO** Module Provides Low Level Hardware Access To the MCU

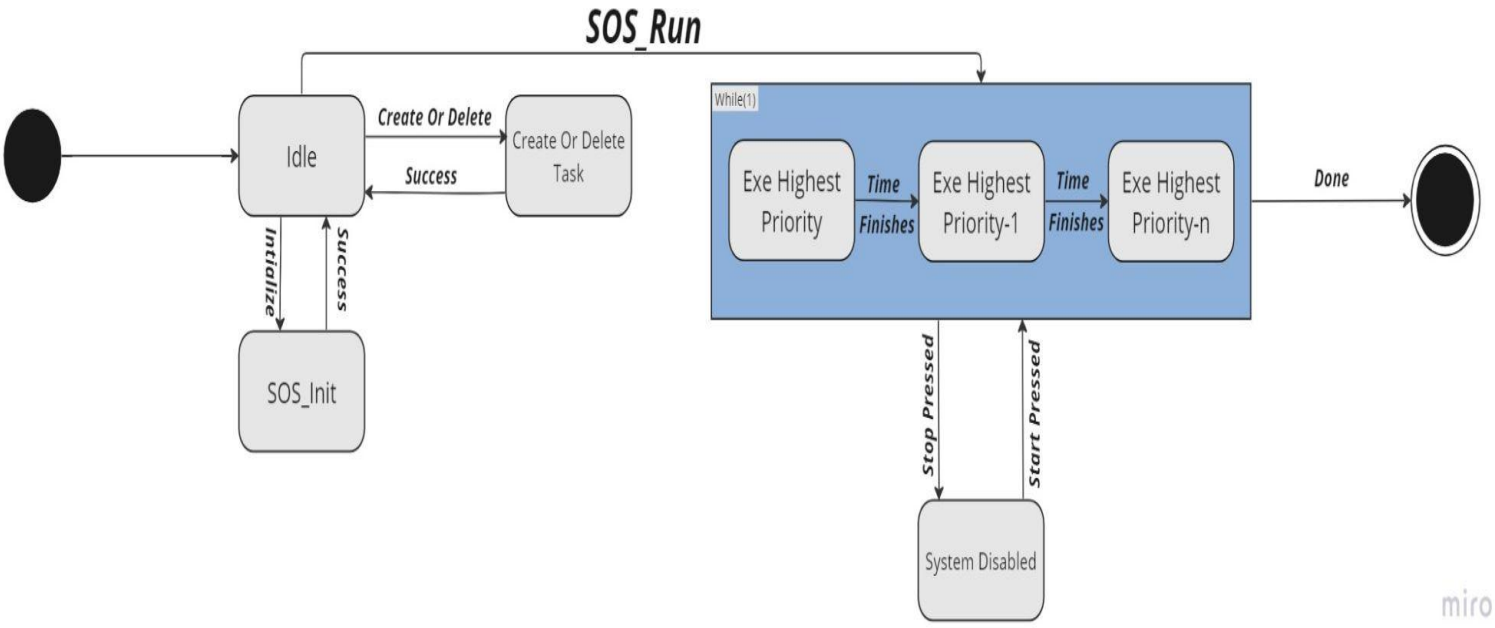
-**Timer** : The Timer driver is responsible for setting up and controlling the timers of the microcontroller.

-**EXI** *The Interrupt driver is responsible for setting up and* controlling the interrupts of the microcontroller

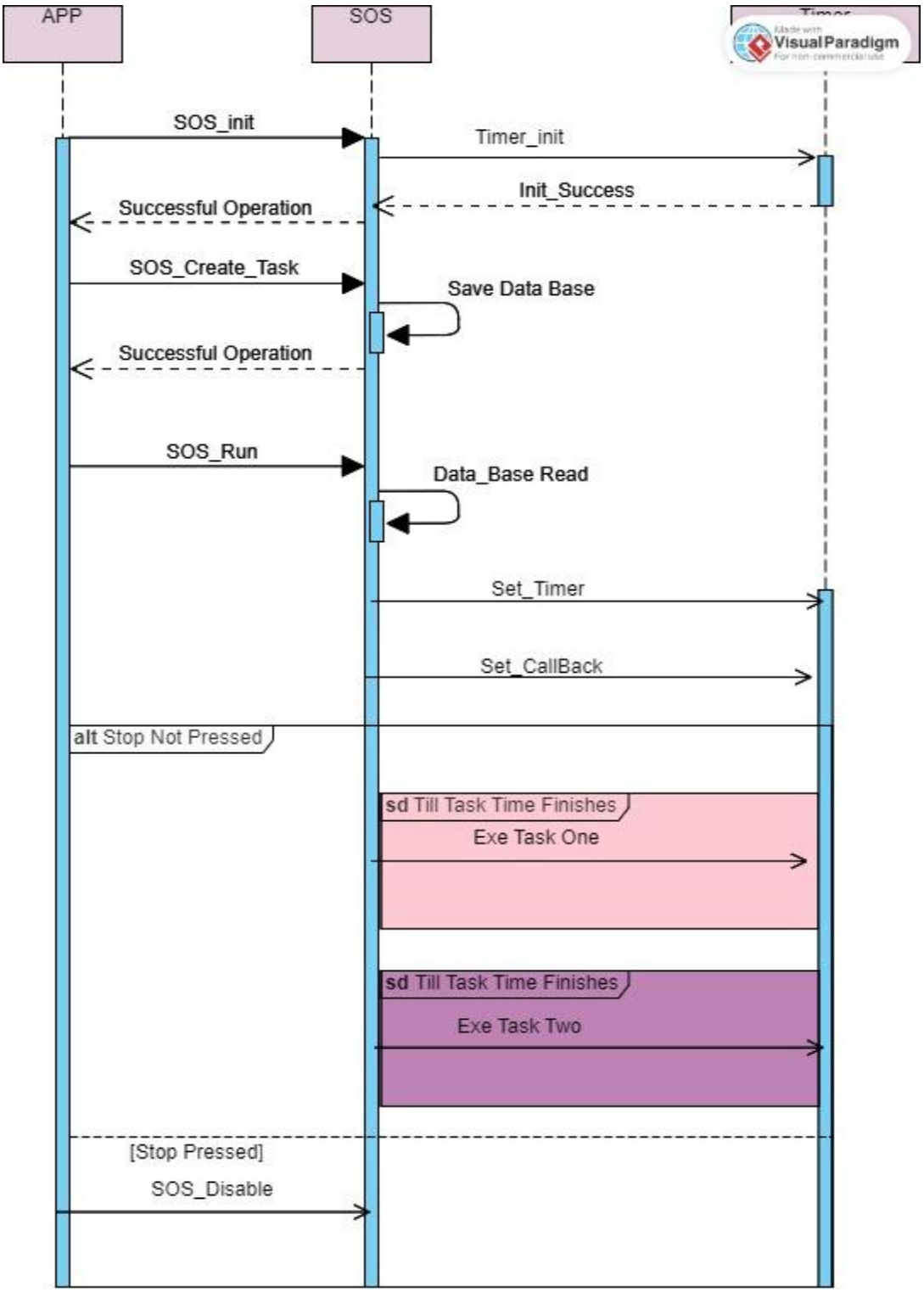
-**LED** Modules Provides access To The Hardware LED For an On/Off States

miro

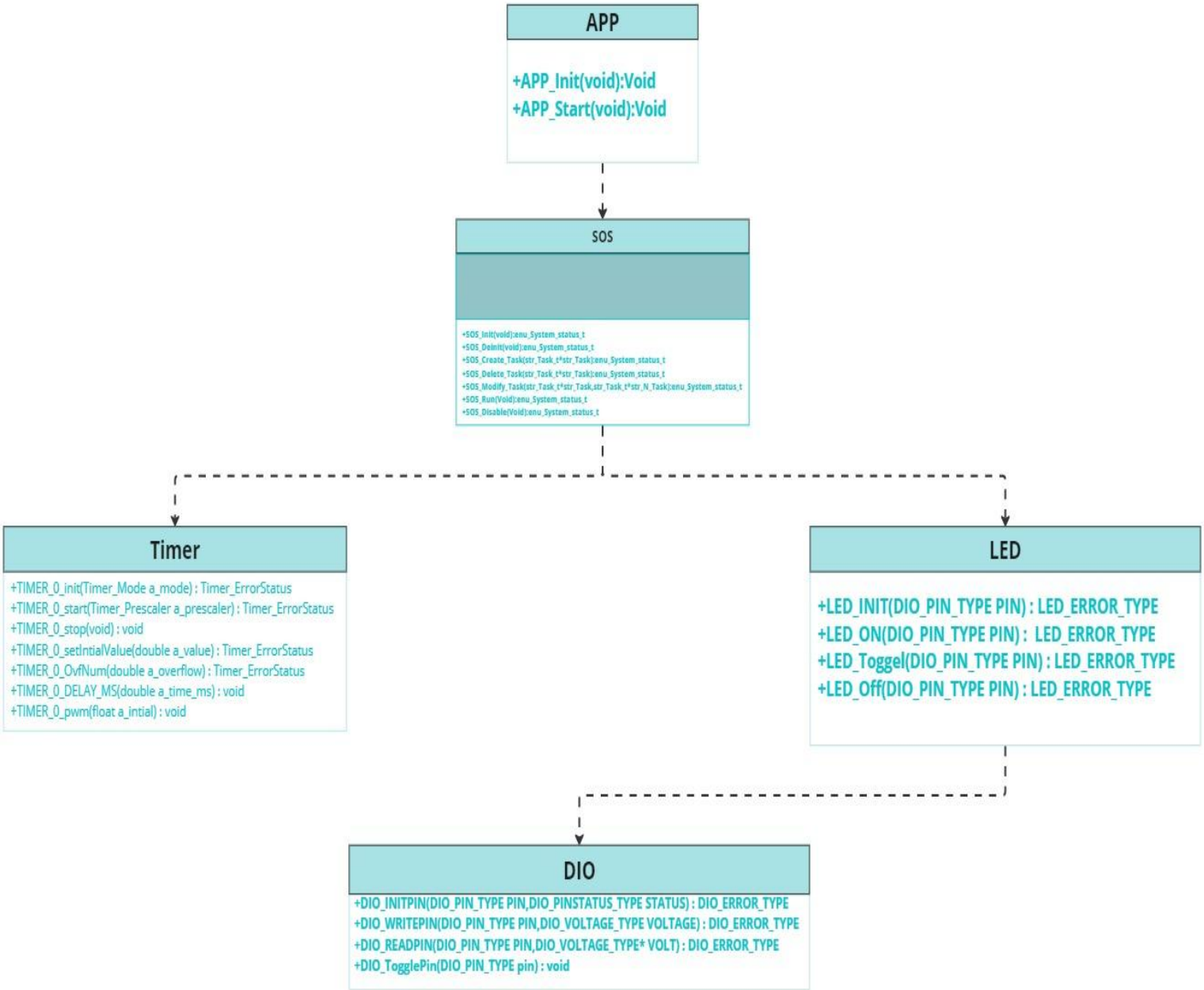
3. State Machine:

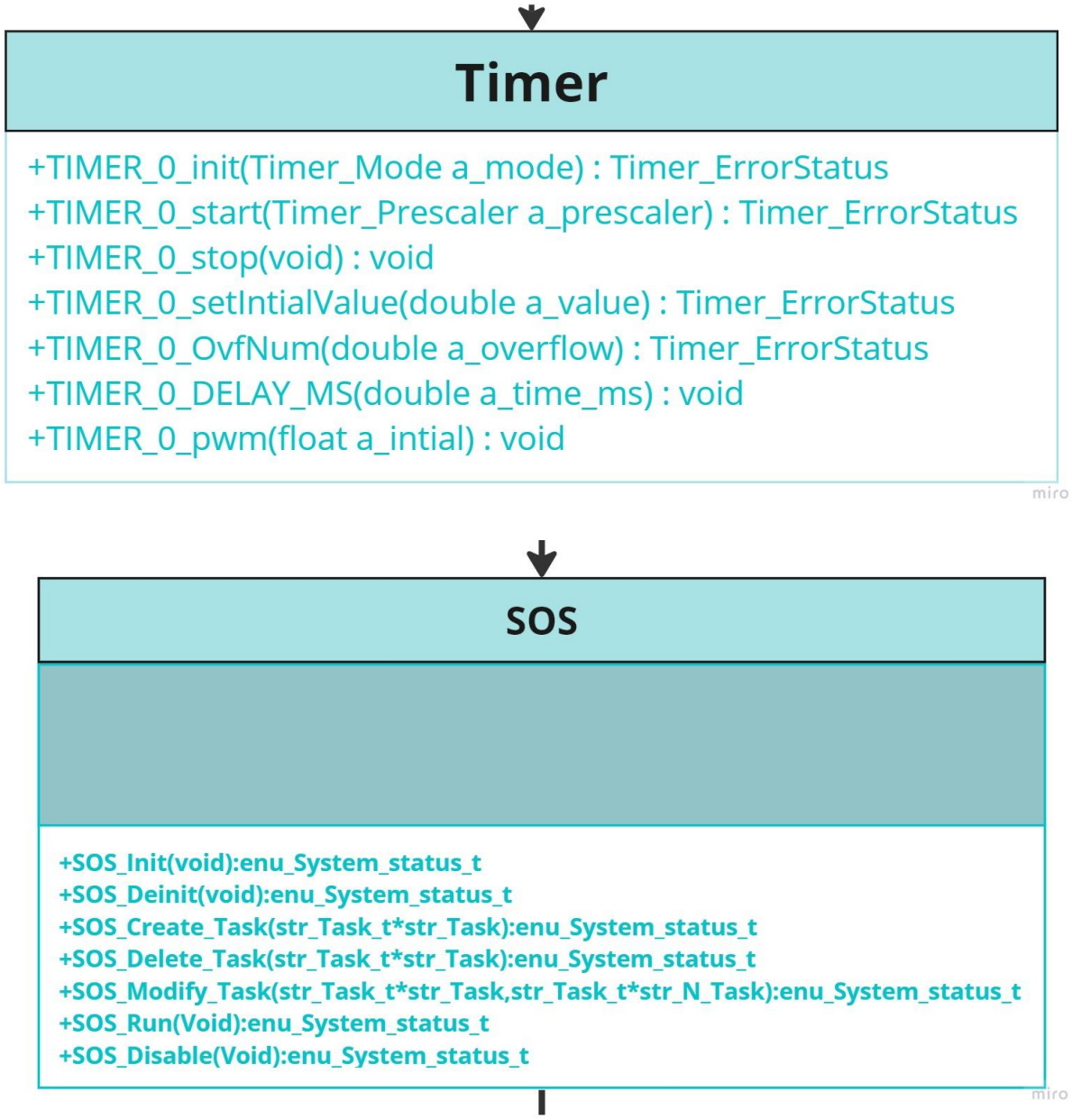


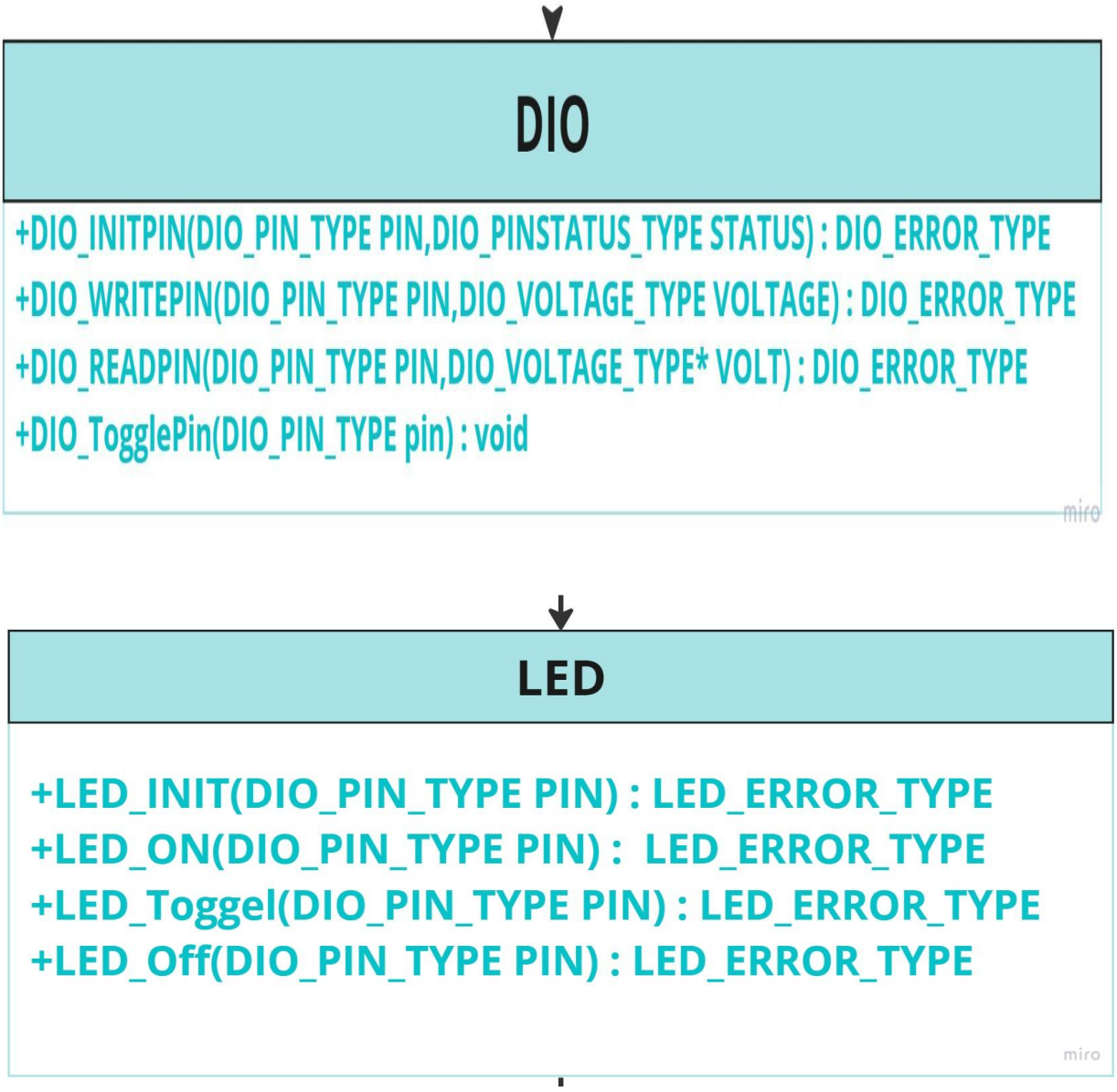
4. Sequence Diagram:



5. Cass Diagram:




Timer
+TIMER_0_init(Timer_Mode a_mode) : Timer_ErrorStatus
+TIMER_0_start(Timer_Prescaler a_prescaler) : Timer_ErrorStatus
+TIMER_0_stop(void) : void
+TIMER_0_setIntialValue(double a_value) : Timer_ErrorStatus
+TIMER_0_OvfNum(double a_overflow) : Timer_ErrorStatus
+TIMER_0_DELAY_MS(double a_time_ms) : void
+TIMER_0_pwm(float a_intial) : void
miro
SOS
+SOS_Init(void):enu_System_status_t
+SOS_Deinit(void):enu_System_status_t
+SOS_Create_Task(str_Task_t*str_Task):enu_System_status_t
+SOS_Delete_Task(str_Task_t*str_Task):enu_System_status_t
+SOS_Modify_Task(str_Task_t*str_Task,str_Task_t*str_N_Task):enu_System_status_t
+SOS_Run(Void):enu_System_status_t
+SOS_Disable(Void):enu_System_status_t
miro


DIO
+DIO_INITPIN(DIO_PIN_TYPE PIN,DIO_PINSTATUS_TYPE STATUS) : DIO_ERROR_TYPE
+DIO_WRITEPIN(DIO_PIN_TYPE PIN,DIO_VOLTAGE_TYPE VOLTAGE) : DIO_ERROR_TYPE
+DIO_READPIN(DIO_PIN_TYPE PIN,DIO_VOLTAGE_TYPE* VOLT) : DIO_ERROR_TYPE
+DIO_TogglePin(DIO_PIN_TYPE pin) : void
miro
LED
+LED_INIT(DIO_PIN_TYPE PIN) : LED_ERROR_TYPE
+LED_ON(DIO_PIN_TYPE PIN) : LED_ERROR_TYPE
+LED_Toggel(DIO_PIN_TYPE PIN) : LED_ERROR_TYPE
+LED_Off(DIO_PIN_TYPE PIN) : LED_ERROR_TYPE
miro