EGYPFWD Initiative

Advanced Embedded Systems Nanodegree,

Embedded Software Design Masterclass by SPRINTS Egypt.

Automotive Door Control System Static Design

A Graduation Project submitted in partial Fulfillment of

Embedded Software Design Masterclass.

Prepared by

Yahia Emad Ahmed Derbala

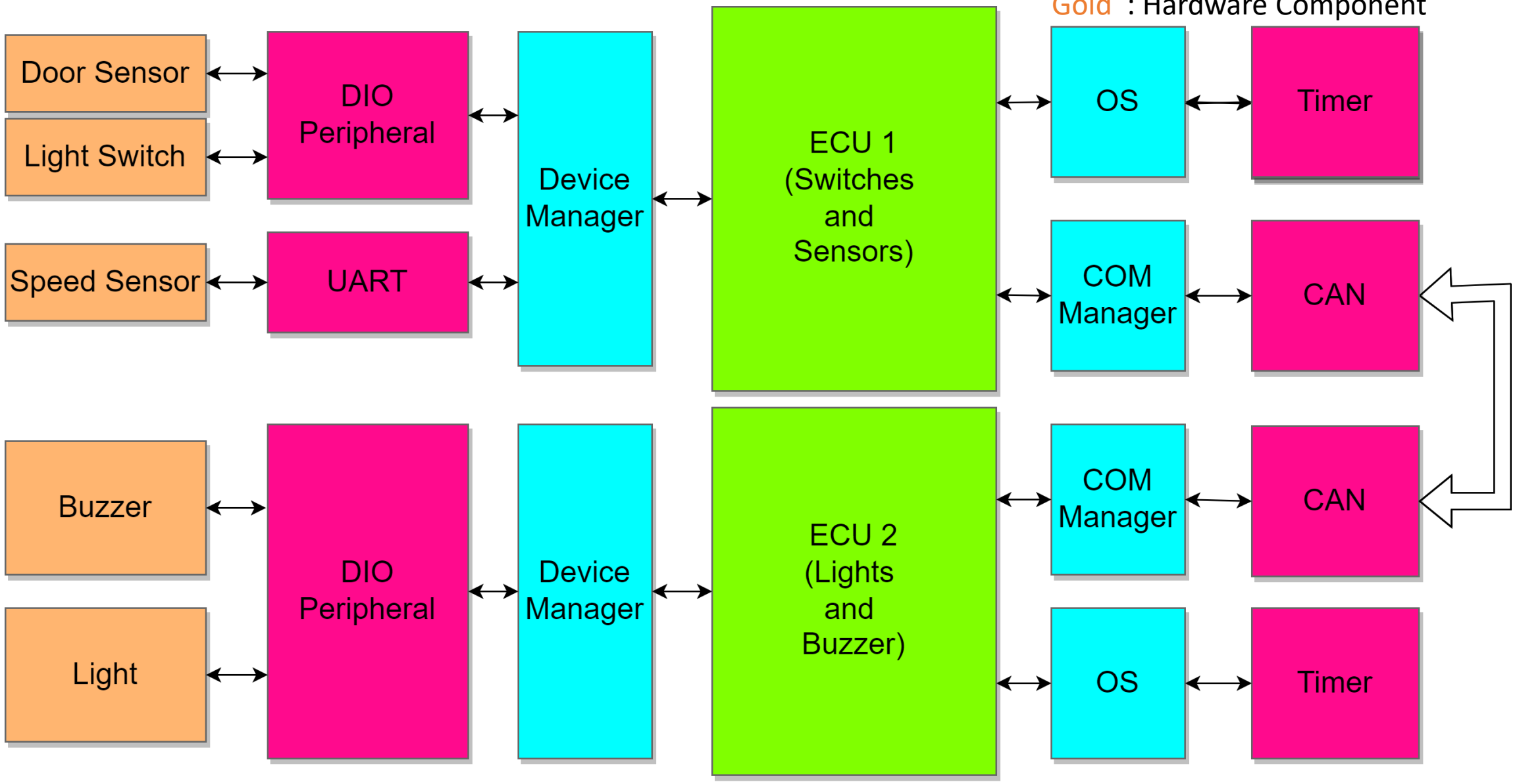
yahderbala@gmail.com

LinkedIn

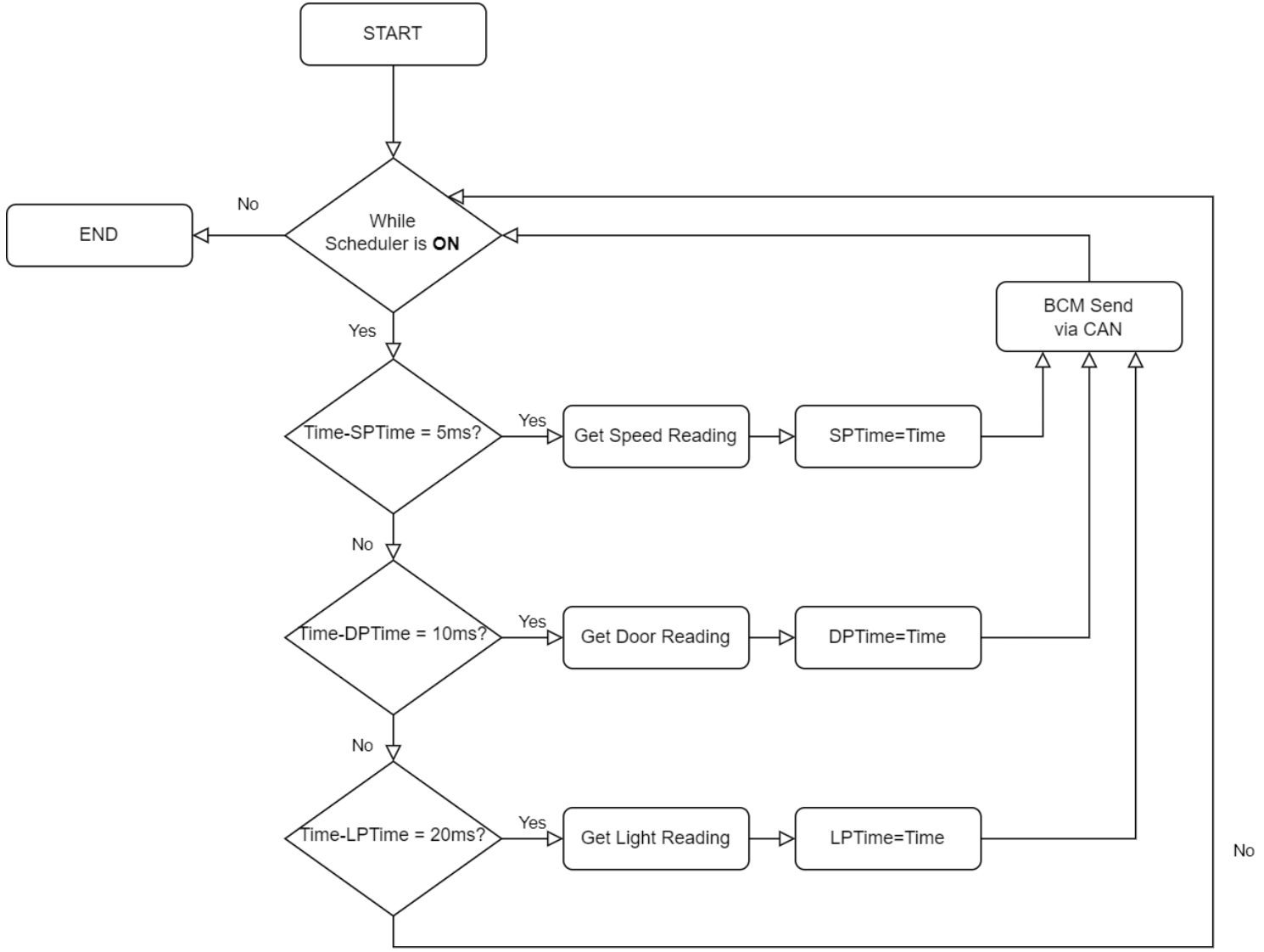
Github

System Schematic Diagram

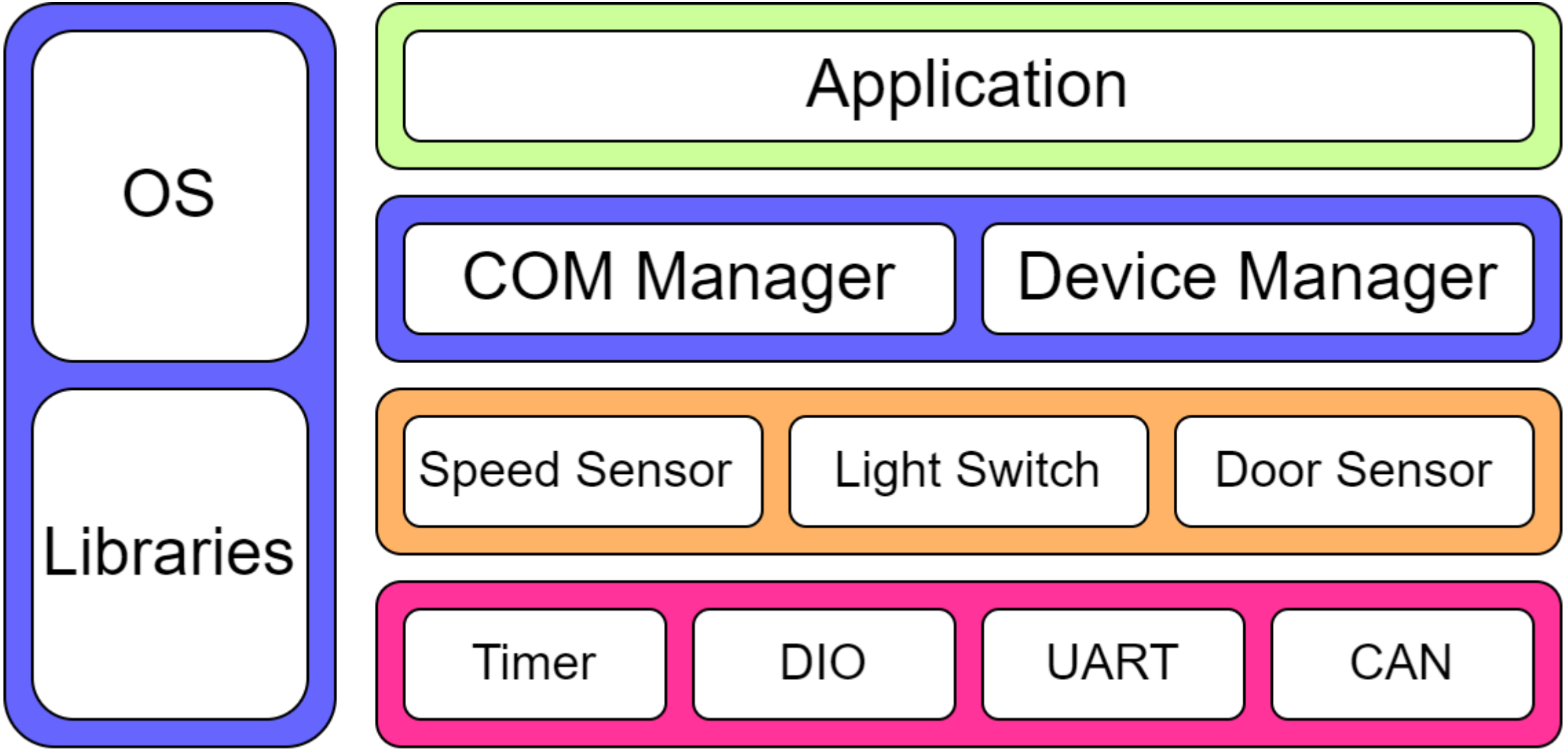
Green: Microcontroller
Blue : Software Component
Pink : Peripheral
Gold : Hardware Component



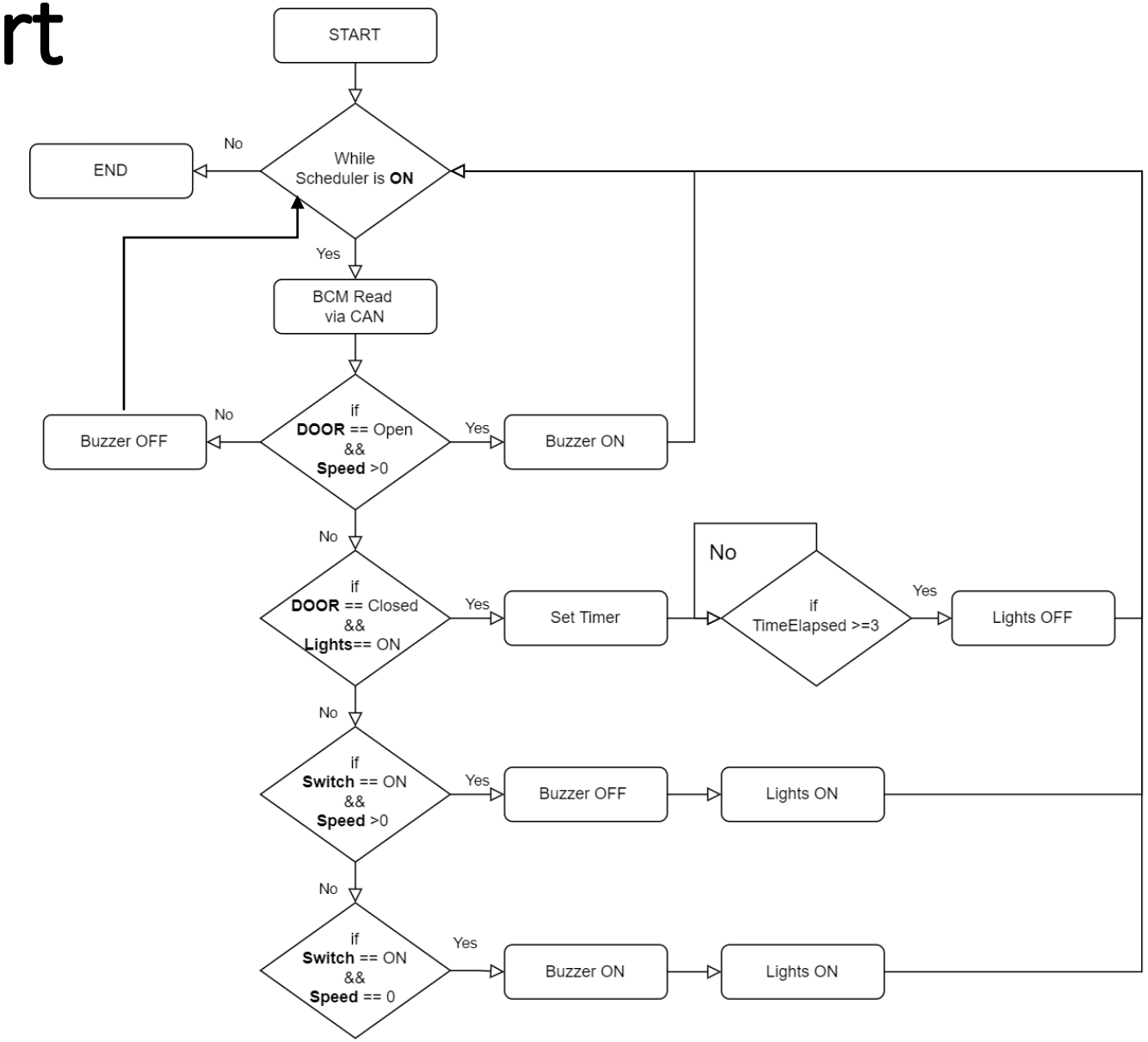
ECU 1 Flowchart



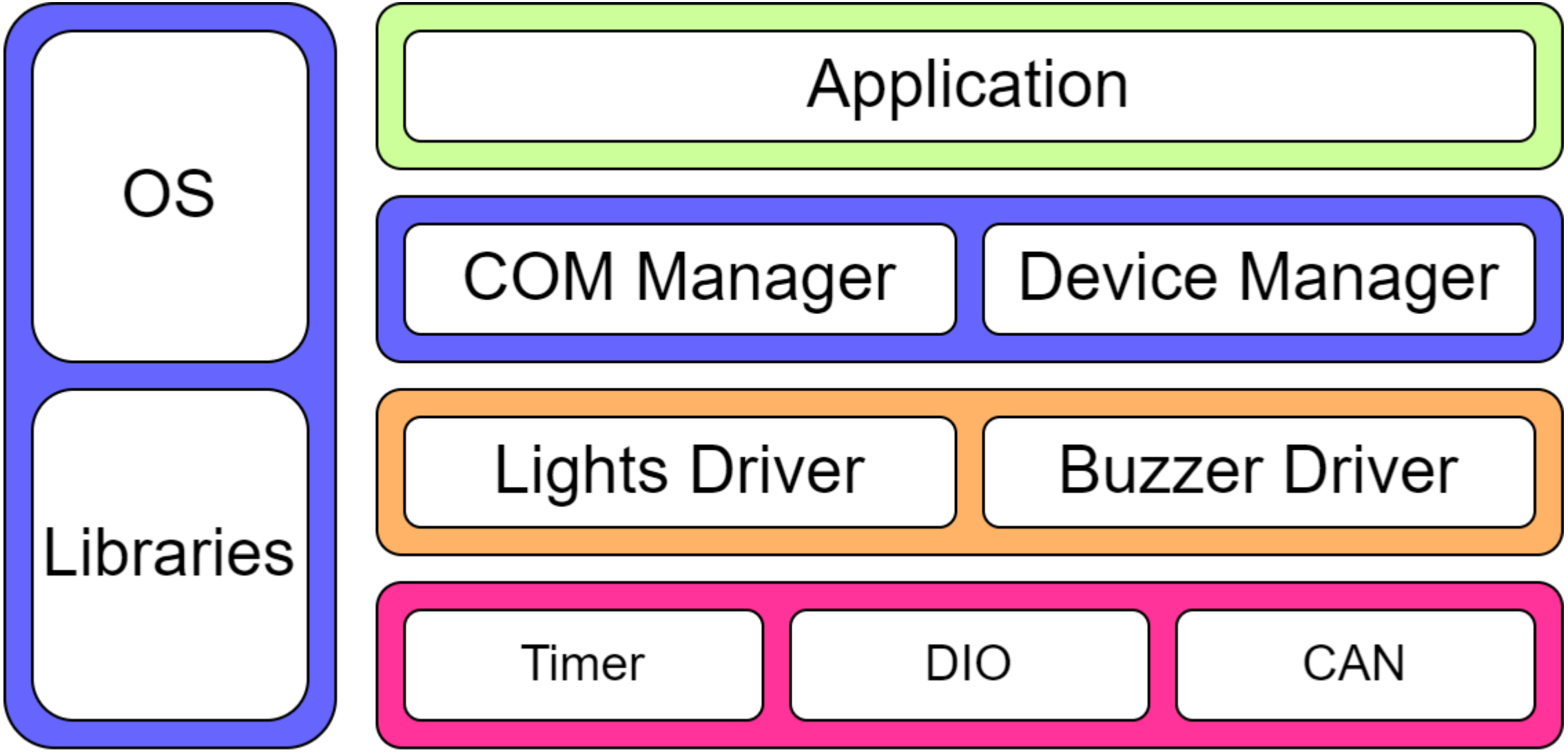
Static Design of ECU 1



ECU 2 Flowchart



Static Design of ECU 2



ECU 1&2 APIs : Timer

Typedef	Struct TimerCfg_t	Range
Enum	Timer_Mode	Mode_16→Mode_32→Mode_64
Enum	Timer_Number	Timer_0 → Timer_24 (**depends on mode**)
Enum	Prescaler	Prescaler_8→Prescaler_16
int	Preload	$0 \rightarrow 2^{Timer\_Mode}$
Typedef	**enum ACK_t**	**ACK_OK | ACK_ERROR**

ECU 1&2 APIs : Timer

API Name	ACK_t xTimer_Init(TimerCfg_t Timer)
Description	Initialize timer peripheral with provided config
Input Parameters	TimerCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : Timer

API Name	ACK_t vTimer_Start(TimerCfg_t Timer)
Description	Start timer
Input Parameters	TimerCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : Timer

API Name	ACK_t vTimer_Stop(TimerCfg_t Timer)
Description	Stop timer
Input Parameters	TimerCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : DIO

Typedef	Struct PinCfg_t	RANGE
Enum	Pin_Number	Pin_0 → Pin_43
Enum	Pin_Direction	Pin_Input→ Pin_Output
Enum	Pin_Mode	Pin_High→Pin_Low
Enum	Pin_Special_Function	Pin_*FunctionName*
Typedef	**Enum** Pin_Status	RANGE
Description	Pin reading value	Pin_High→Pin_Low
Typedef	**enum ACK_t**	**ACK_OK | ACK_ERROR**

ECU 1&2 APIs : DIO

API Name	ACK_t xDio_Init(PinCfg_t Pins[])
Description	Initialize dio peripheral with provided configs
Input Parameters	PinCfg_t[“Number of pins”]
Output Parameters	ACK_t

ECU 1&2 APIs : DIO

API Name	ACK_t xDio_Set(PinCfg_t Pin)
Description	Set DIO pin high
Input Parameters	PinCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : DIO

API Name	ACK_t xDio_Clear(PinCfg_t Pin)
Description	Un-set DIO pin to low
Input Parameters	PinCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : DIO

API Name	ACK_t xDio_Get(PinCfg_t Pin)
Description	Get DIO pin status
Input Parameters	PinCfg_t
Output Parameters	Pin_Status

ECU 1&2 APIs : UART

Typedef	Struct UartCfg_t	RANGE
Enum	UART_Mode	UART_HW→UART_SW
Enum	Baudrate	9600→115200
Enum	RX_Pin	Pin_0→Pin_43
Enum	TX_Pin	
Typedef	Char UART_Msg[]	
Description	Message string to be used for sending and receiving	
Typedef	enum ACK_t	ACK_OK | ACK_ERROR

ECU 1&2 APIs : UART

API Name	ACK_t xUart_Init(UartCfg_t UART)
Description	Initialize UART peripheral with config
Input Parameters	UartCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : UART

API Name	ACK_t xUart_Send(UartCfg_t UART)
Description	Send from UART peripheral
Input Parameters	UartCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : UART

API Name	ACK_t xUart_Receive(UartCfg_t UART)
Description	Receive from UART peripheral
Input Parameters	UartCfg_t
Output Parameters	UART_Msg

ECU 1&2 APIs : CAN

Typedef	Struct CANCfg_t	RANGE
Enum	CAN_Mode	CAN_Mode1→CAN_ModeX
Enum	Baudrate	9600→115200

Typedef	Char CAN_Msg[]
Description	Message string to be used for sending and receiving

Typedef	enum ACK_t	ACK_OK	ACK_ERROR

ECU 1&2 APIs : CAN

API Name	ACK_t xCAN_Init(CANCfg_t CAN)
Description	Initialize CAN peripheral with config
Input Parameters	CANCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : CAN

API Name	ACK_t xCAN_Send(CANCfg_t CAN)
Description	Send data over CANBUS
Input Parameters	CANCfg_t
Output Parameters	ACK_t

ECU 1&2 APIs : CAN

API Name	CAN_Msg **xCAN_Receive(**CANCfg_t **CAN)**
Description	Receive data over CANBUS
Input Parameters	CANCfg_t
Output Parameters	CAN_Msg

ECU 1&2 APIs : COM Manager

Typedef	Struct COM_Device_t	RANGE
Enum	COM_Protocol	CAN / UART / I2C
Enum	Channel	Channel_0→Channel_3
char	COM_Msg[]	
Typedef	**enum ACK_t**	**ACK_OK | ACK_ERROR**

ECU 1&2 APIs : COM Manager

API Name	ACK_t xCom_Init(COM_Device_t Manager)
Description	Initialize a Communication manager
Input Parameters	COM_Device_t
Output Parameters	ACK_t

ECU 1&2 APIs : COM Manager

API Name	ACK_t xCom_Send(COM_Device_t Manager)
Description	Send message using communication manager
Input Parameters	COM_Device_t
Output Parameters	ACK_t

ECU 1&2 APIs : COM Manager

API Name	ACK_t xCom_Receive(COM_Device_t Manager)
Description	Receive message using communication manager
Input Parameters	COM_Device_t
Output Parameters	COM_Device_t

ECU 1&2 APIs : Device Manager

Typedef	Struct Device_t	Range
Enum	Device_ID	Device_0→Device_2 for ECU1 Device_0→Device 1 for ECU 2
Enum	Interface_Type	UART / DIO
char	Device_Msg[]	
Typedef	**enum ACK_t**	**ACK_OK	ACK_ERROR**

ECU 1&2 APIs : Device Manager

API Name	ACK_t xDevice_Init(Device_t Device)
Description	Initialize device manager
Input Parameters	Device_t
Output Parameters	ACK_t

ECU 1&2 APIs : Device Manager

API Name	ACK_t xDevice_Receive(Device_t Device)
Description	Receive device reading using device manager
Input Parameters	Device_t
Output Parameters	Device_t

ECU 1 APIs : Speed Sensor

Typedef	Struct SpeedCfg_t	Range
Enum	Sensor_Type	Type_UART or Type_I2C
Enum	TX_Pin	Pin_0→Pin_43
Enum	RX_Pin	

Typedef	Char Speed_Msg[]	
Description	Message string to be used for sending and receiving	
Typedef	**enum ACK_t**	**ACK_OK	ACK_ERROR**

ECU 1 APIs : Speed Sensor

API Name	ACK_t vSpeed_Init(SpeedCfg_t Speed)
Description	Initialize a speed sensor with the given config
Input Parameters	SpeedCfg_t
Output Parameters	ACK_t

ECU 1 APIs : Speed Sensor

API Name	Speed_Msg xSpeed_GetReading(SpeedCfg_t Speed)
Description	Get speed reading from sensor
Input Parameters	SpeedCfg_t
Output Parameters	Speed_Msg

ECU 1 APIs : Light Sensor

Typedef	**Struct LightCfg_t**	**Range**
Enum	Sensor_Type	Type_UART / Type_DIO
Enum	Pin	Pin_0→Pin_43

Typedef	**Enum** Light_Status	
Description	Light sensor reading	
Typedef	**enum ACK_t**	**ACK_OK	ACK_ERROR**

ECU 1 APIs : Light Sensor

API Name	ACK_t xLight_Init(LightCfg_t Light)
Description	Initialize a light sensor with the given config
Input Parameters	LightCfg_t
Output Parameters	ACK_t

ECU 1 APIs : Light Sensor

API Name	Light_Status xLight_Read(LightCfg_t Light)
Description	Read a light sensor
Input Parameters	LightCfg_t
Output Parameters	Light_Status

ECU 1 APIs : Door Sensor

Typedef	Struct DoorCfg_t	Range
Enum	Sensor_Type	Type_UART / Type_DIO
Enum	Pin	Pin_0→Pin_43

Typedef	Enum Door_Status		
Description	Door sensor reading		
Typedef	**enum ACK_t**	**ACK_OK**	**ACK_ERROR**

ECU 1 APIs : Door Sensor

API Name	ACK_t xDoor_Init(DoorCfg_t Door)
Description	Initialize a Door sensor with the given config
Input Parameters	DoorCfg_t
Output Parameters	ACK_t

ECU 1 APIs : Door Sensor

API Name	Door_Status xDoor_Read(DoorCfg_t Door)
Description	Read a Door sensor
Input Parameters	DoorCfg_t
Output Parameters	Door_Status

ECU 2 APIs : Light Actuator

Typedef	Struct LightCfg_t	Range
Enum	Actuator_Type	Type_UART / Type_DIO
Enum	Pin	Pin_0→Pin_43

Typedef	enum ACK_t	ACK_OK | ACK_ERROR

ECU 2 APIs : Light Sensor

API Name	ACK_t xLight_Init(LightCfg_t LED)
Description	Initialize a Light actuator with the given config
Input Parameters	LightCfg_t
Output Parameters	ACK_t

ECU 2 APIs : Light Actuator

API Name	ACK_t xLight_SetAction(LightCfg_t LED)
Description	Set actuator action
Input Parameters	LightCfg_t
Output Parameters	ACK_t

ECU 2 APIs : Buzzer Actuator

Typedef	Struct BuzzerCfg_t	RANGE
Enum	Actuator_Type	Type_UART / Type_DIO
Enum	Pin	Pin_0→Pin_43

Typedef	enum ACK_t	ACK_OK | ACK_ERROR

ECU 2 APIs : Buzzer Sensor

API Name	ACK_t Buzzer_Init(BuzzerCfg_t Buzzer)
Description	Initialize a Buzzer actuator with the given config
Input Parameters	BuzzerCfg_t
Output Parameters	ACK_t

ECU 2 APIs : Buzzer Actuator

API Name	ACK_t xBuzzer_SetAction(BuzzerCfg_t Buzzer)
Description	Set actuator action
Input Parameters	BuzzerCfg_t
Output Parameters	ACK_t