STATIC DESIGN

EGFWD ADVANCED EMBEDDED SYSTEMS SCHOLERSHIP

EMBEDDED SW DESIGN PROJECT

Prepared by

MOHAMMED RASHAD

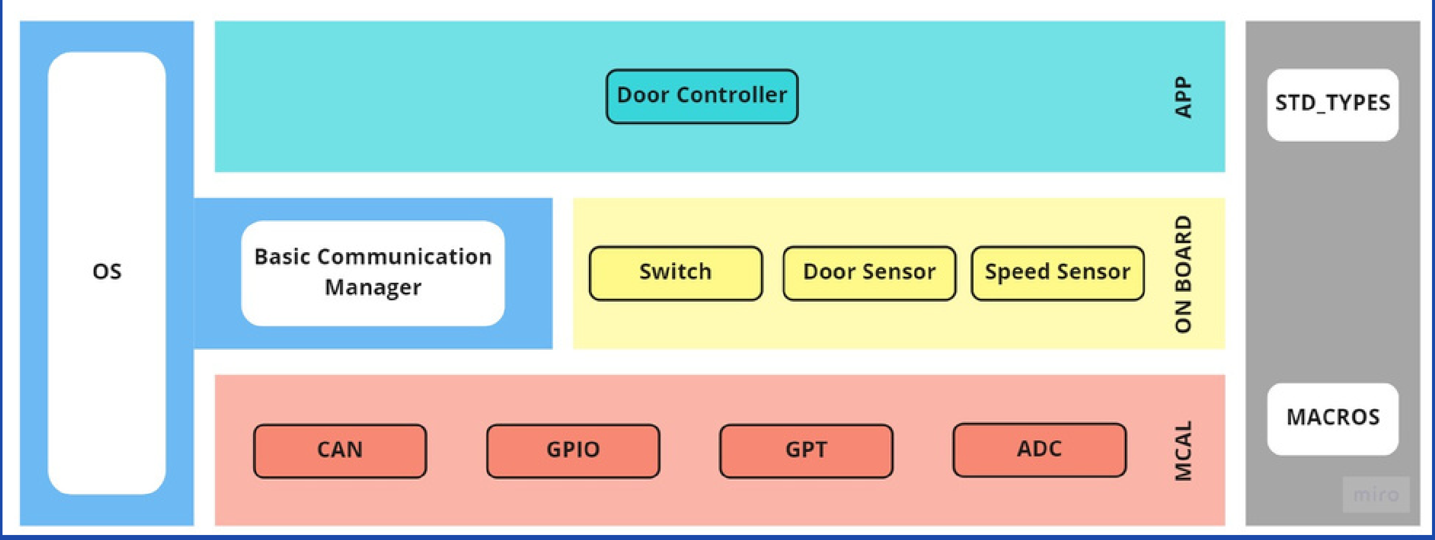
Ministry of Communications and Information Technology

egypt fwd

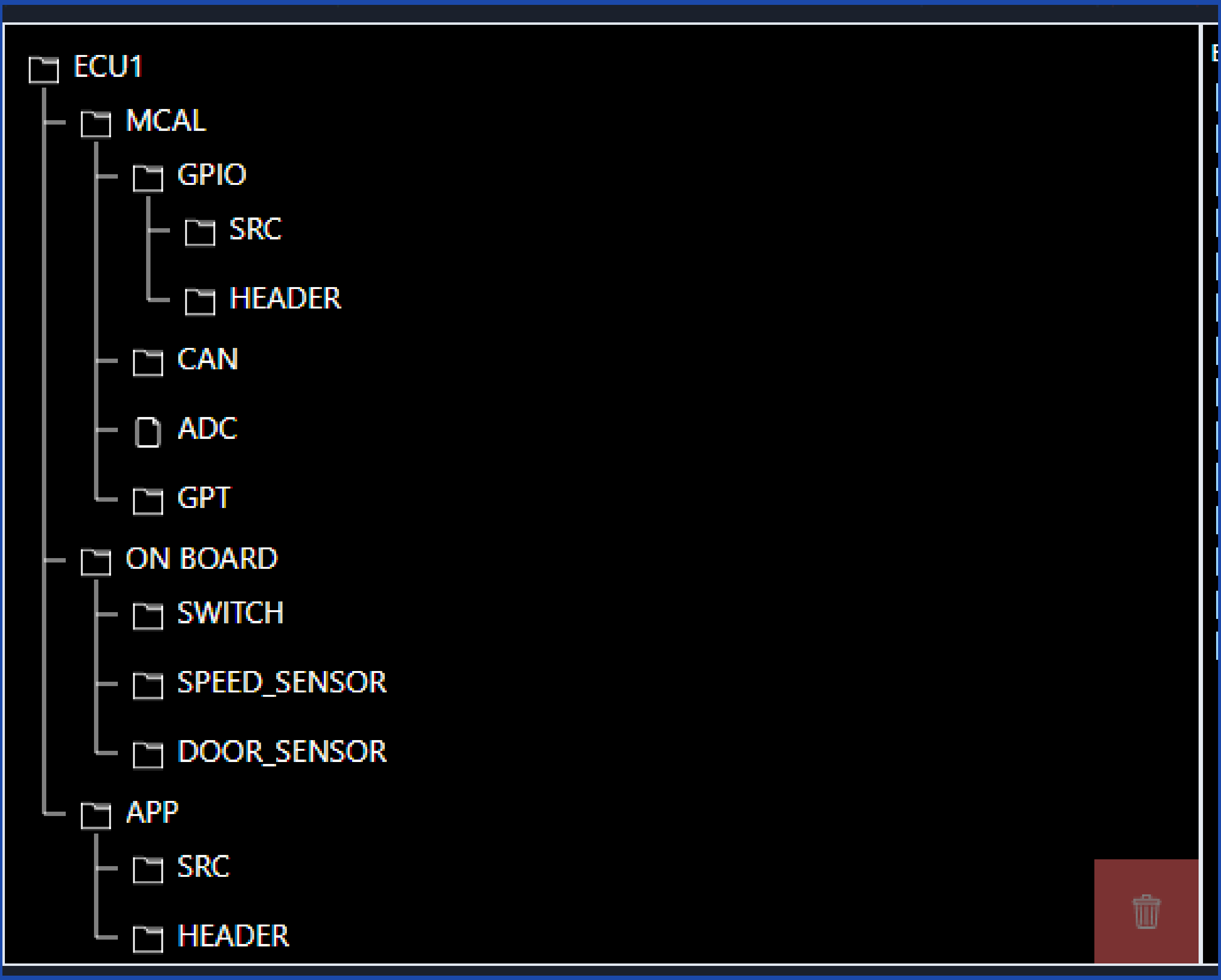
مستقبلنا رقمي

LAYERED ARCHITECTURE FOR ECU1

Layers Design



FOLDER STRUCTURE



```
ECU1
├─ MCAL
│  ├─ GPIO
│  │  ├─ SRC
│  │  └─ HEADER
│  ├─ CAN
│  ├─ ADC
│  └─ GPT
├─ ON BOARD
│  ├─ SWITCH
│  ├─ SPEED_SENSOR
│  └─ DOOR_SENSOR
└─ APP
   ├─ SRC
   └─ HEADER
```

CAN

API TYPES	- CAN_CONFIG - DATA
API FUNCTIONS	- void CAN_INIT (CAN_CONFIG * CAN_PTR) - void CAN_READ (DATA * DATA_PTR) - void CAN_WRITE (DATA * DATA_PTR)

CAN_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of CAN communication which will be passed to init function then .

DATA

TYPE	uint8 array
RANGE	-
DESCRIPTION	Array of bits holding data received or sent over CAN protocol

CAN_INIT

ARGUMENTS	• CAN_CONFIG * CAN_PTR pointer to connection post-build configurations
RETURN	void
DESCRIPTION	This function Initializes CAN communication

CAN_READ

ARGUMENTS	• DATA * DATA_PTR pointer to received data array
RETURN	void
DESCRIPTION	This function will receive data and store it in the array which is passed to it .

CAN_WRITE

ARGUMENTS	• DATA * DATA_PTR pointer to data array to be sent
RETURN	void
DESCRIPTION	This function will get the data array passed to it and send it over CAN .

GPIO

API TYPES	• GPIO_CONFIG
API FUNCTIONS	• void GPIO_PINDIR (GPIO_CONFIG * PIN_PTR) • uint8 GPIO_READ (uint8 PIN , uint8 PORT) • void GPIO_WRITE (uint8 PIN , uint8 PORT , uint8 LEVEL)

GPIO_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of GPIO pin which will be passed to direction function then .

GPIO_PINDIR

ARGUMENTS	• GPIO_CONFIG * PIN_PTR pointer to pin post-build configurations
RETURN	void
DESCRIPTION	This function Initializes pin direction and configurations

GPIO_READ

ARGUMENTS	• uint8 PIN pin number • uint8 PORT port number
RETURN	uint8 type level either 0 for LOW or 1 for HIGH
DESCRIPTION	This function will read pin level.

GPIO_WRITE

ARGUMENTS	• uint8 PIN pin number • uint8 PORT port number • uint8 LEVEL pin level
RETURN	void
DESCRIPTION	This function will set given pin level.

GPT

API TYPES

- GPT_CONFIG

API FUNCTIONS

- void GPT_INIT (GPT_CONFIG * GPT_PTR)
- void GPT_DELAYMS (uint32 delay , uint8 TIMER)
- void GPT_DELAYCB (void (*CB_FUNC) (void) , uint32 delay , uint8 TIMER)

GPT_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of GPT channel which will be passed to init function then .

GPT_INIT

ARGUMENTS

- GPT_CONFIG *GPT_PTR
  pointer to timer channel post-build configurations

RETURN

void

DESCRIPTION

This function Initializes GPT channel by given configurations

GPT_DELAYMS

ARGUMENTS

- uint32 delay
  delay interval in ms
- uint8 TIMER
  timer ID

RETURN

void

DESCRIPTION

This function takes delay interval in ms and converts it to ticks then load it to the timer in order to block CPU for that interval

GPT_DELAYCB

ARGUMENTS

- uint32 delay
  delay interval in ms
- void (*CB_FUNC) (void)
  pointer to user defined callback function
- uint8 TIMER
  timer ID

RETURN

void

DESCRIPTION

This function blocks CPU for given interval in ms then it calls a callback function defined by user

ADC

API TYPES	- ADC_CONFIG
API FUNCTIONS	- void ADC_INIT (ADC_CONFIG * ADC_PTR) - uint32 ADC_VALUE (uint8 ADC_CHANNEL)

ADC_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of ADC channel which will be passed to init function then .

ADC_INIT

ARGUMENTS	• ADC_CONFIG *ADC_PTR pointer to timer channel post-build configurations
RETURN	void
DESCRIPTION	This function Initializes ADC channel by given configurations

ADC_VALUE

ARGUMENTS	• uint8 ADC_CHANNEL ADC channel ID
RETURN	uint 32 value
DESCRIPTION	this function returns value of ADC at the end of conversion

SWITCH

API TYPES	--
API FUNCTIONS	• void SWITCH_INIT (uint8 PIN , uint8 PORT) • uint8 SWITCH_STATE (uint8 PIN , uint8 PORT)
INCLUDES	• GPIO

SWITCH_INIT

ARGUMENTS	• uint8 PIN pin number connected to the switch • uint8 PORT port id of pin connected to the switch
RETURN	void
DESCRIPTION	This function Initializes switch using GPIO module

SWITCH_STATE

ARGUMENTS	• uint8 PIN pin number connected to the switch • uint8 PORT port id of pin connected to the switch
RETURN	uint8 Level
DESCRIPTION	this function returns value of switch ON or OFF states

DOOR SENSOR

API TYPES	--
API FUNCTIONS	• void DSENSOR_INIT (uint8 PIN , uint8 PORT) • uint8 DSENSOR_STATE (uint8 PIN , uint8 PORT)
INCLUDES	• GPIO

DSENSOR_INIT

ARGUMENTS

- uint8 PIN
  pin number connected to the sensor
- uint8 PORT
  port id of pin connected to the sensor

RETURN

void

DESCRIPTION

This function Initializes sensor using GPIO module

DSENSOR_STATE

ARGUMENTS

- uint8 PIN
  pin number connected to the sensor
- uint8 PORT
  port id of pin connected to the sensor

RETURN

uint8 Level

DESCRIPTION

this function returns value of sensor HIGH or LOW states

SPEED SENSOR

API TYPES	--
API FUNCTIONS	• void SSENSOR_INIT (uint8 CHANNEL) • uint32 SSENSOR_VALUE (uint8 CHANNEL)
INCLUDES	• ADC

SSENSOR_INIT

ARGUMENTS

- uint8 CHANNEL
  ADC channel id connected to the sensor

RETURN

void

DESCRIPTION

This function Initializes sensor using ADC module

SSENSOR_VALUE

ARGUMENTS

- uint8 CHANNEL
  ADC channel id connected to the sensor

RETURN

uint32 value

DESCRIPTION

this function returns value of sensor reading using ADC

BCM

API TYPES	• **COMM_CONFIG** • **DATA**
API FUNCTIONS	• void COMM_INIT (COMM_CONFIG* COMM) • void SEND (DATA* DATA_ARR , uint8 COMM_ID) • void RECEIVE (DATA* DATA_ARR , uint8 COMM_ID)
INCLUDES	• CAN

COMM_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of communication protocol which will be passed to init function then .

DATA

TYPE	uint8 array
RANGE	-
DESCRIPTION	Array of bits holding data received or sent

COMM_INIT

ARGUMENTS

- COMM_CONFIG * COMM
  pointer to communication post-build configurations

RETURN

void

DESCRIPTION

This function Initializes communication

SEND

ARGUMENTS

- DATA * DATA_ARR
  pointer to data array to be sent
- uint8 COMM_ID
  communication ID

RETURN

void

DESCRIPTION

This function will send the data given to it over specified communication

RECEIVE

ARGUMENTS

- DATA * DATA_ARR
  pointer to data array to store received data
- uint8 COMM_ID
  communication ID

RETURN

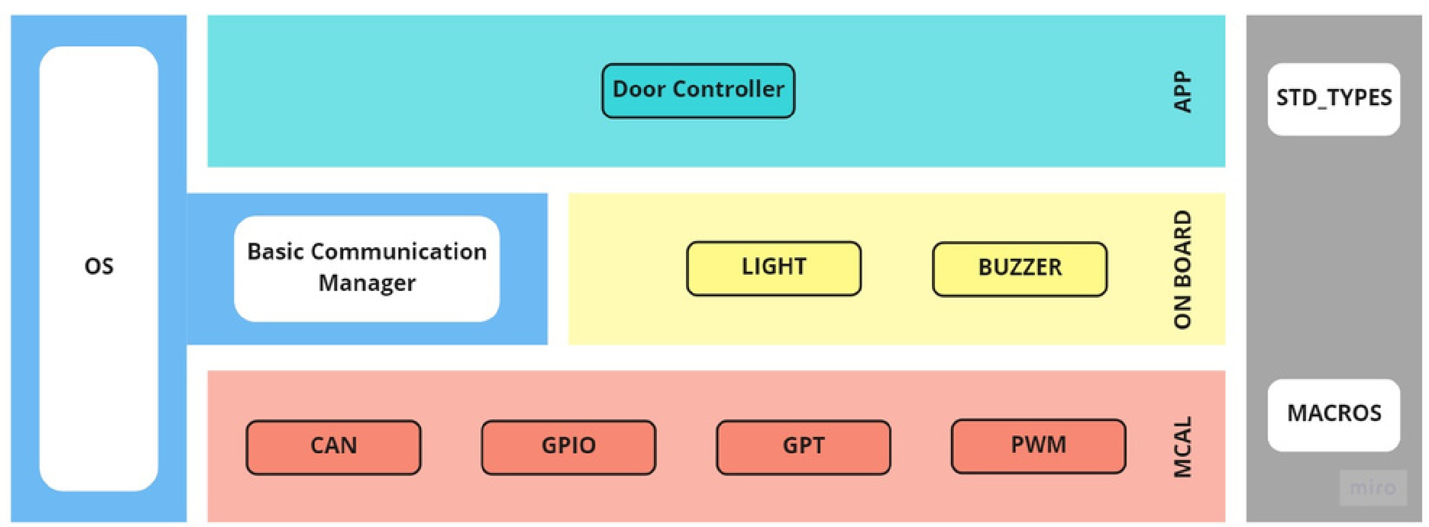
void

DESCRIPTION

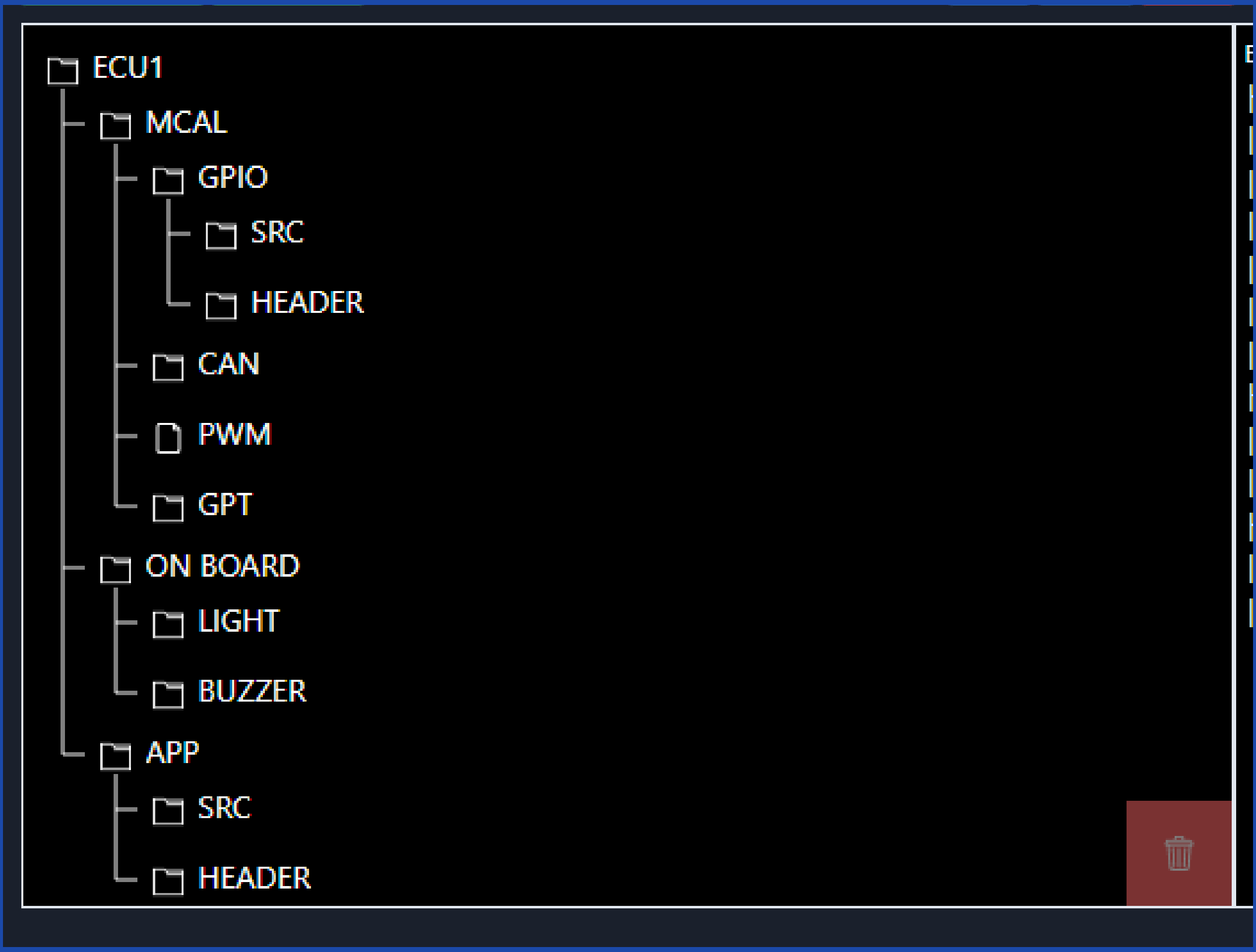
This function will receive the data from specified communication and store it in the array

LAYERED ARCHITECTURE FOR ECU2

Layers Design



FOLDER STRUCTURE



```
ECU1
├─ MCAL
│  ├─ GPIO
│  │  ├─ SRC
│  │  └─ HEADER
│  ├─ CAN
│  ├─ PWM
│  └─ GPT
├─ ON BOARD
│  ├─ LIGHT
│  └─ BUZZER
└─ APP
   ├─ SRC
   └─ HEADER
```

CAN

API TYPES	- CAN_CONFIG - DATA
API FUNCTIONS	- void CAN_INIT (CAN_CONFIG * CAN_PTR) - void CAN_READ (DATA * DATA_PTR) - void CAN_WRITE (DATA * DATA_PTR)

CAN_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of CAN communication which will be passed to init function then .

DATA

TYPE	uint8 array
RANGE	-
DESCRIPTION	Array of bits holding data received or sent over CAN protocol

CAN_INIT

ARGUMENTS	• CAN_CONFIG * CAN_PTR pointer to connection post-build configurations
RETURN	void
DESCRIPTION	This function Initializes CAN communication

CAN_READ

ARGUMENTS	• DATA * DATA_PTR pointer to received data array
RETURN	void
DESCRIPTION	This function will receive data and store it in the array which is passed to it .

CAN_WRITE

ARGUMENTS	• DATA * DATA_PTR pointer to data array to be sent
RETURN	void
DESCRIPTION	This function will get the data array passed to it and send it over CAN .

GPIO

API TYPES	- GPIO_CONFIG
API FUNCTIONS	- void GPIO_PINDIR (GPIO_CONFIG * PIN_PTR) - uint8 GPIO_READ (uint8 PIN , uint8 PORT) - void GPIO_WRITE (uint8 PIN , uint8 PORT , uint8 LEVEL)

GPIO_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of GPIO pin which will be passed to direction function then .

GPIO_PINDIR

ARGUMENTS

- GPIO_CONFIG * PIN_PTR
  pointer to pin post-build configurations

RETURN

void

DESCRIPTION

This function Initializes pin direction and configurations

GPIO_READ

ARGUMENTS

- uint8 PIN
  pin number
- uint8 PORT
  port number

RETURN

uint8 type level
either 0 for LOW or 1 for HIGH

DESCRIPTION

This function will read pin level.

GPIO_WRITE

ARGUMENTS

- uint8 PIN
  pin number
- uint8 PORT
  port number
- uint8 LEVEL
  pin level

RETURN

void

DESCRIPTION

This function will set given pin level.

GPT

API TYPES	- **GPT_CONFIG**
API FUNCTIONS	- void GPT_INIT (GPT_CONFIG * GPT_PTR) - void GPT_DELAYMS (uint32 delay , uint8 TIMER) - void GPT_DELAYCB (void (*CB_FUNC) (void) , uint32 delay , uint8 TIMER)

GPT_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of GPT channel which will be passed to init function then .

GPT_INIT

ARGUMENTS

- GPT_CONFIG *GPT_PTR
  pointer to timer channel post-build configurations

RETURN

void

DESCRIPTION

This function Initializes GPT channel by given configurations

GPT_DELAYMS

ARGUMENTS

- uint32 delay
  delay interval in ms
- uint8 TIMER
  timer ID

RETURN

void

DESCRIPTION

This function takes delay interval in ms and converts it to ticks then load it to the timer in order to block CPU for that interval

GPT_DELAYCB

ARGUMENTS

- uint32 delay
  delay interval in ms
- void (*CB_FUNC) (void)
  pointer to user defined callback function
- uint8 TIMER
  timer ID

RETURN

void

DESCRIPTION

This function blocks CPU for given interval in ms then it calls a callback function defined by user

PWM

API TYPES	- **PWM_CONFIG**
API FUNCTIONS	- void PWM_INIT (PWM_CONFIG * PWM_PTR , uint8 DUTY)

PWM_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of PWM which will be passed to init function then .

PWM_INIT

ARGUMENTS	• PWM_CONFIG *PWM_PTR pointer to PWM post-build configurations • uint8 DUTY Duty cycle
RETURN	void
DESCRIPTION	This function Initializes PWM by given configurations and duty cycle and starts it

LIGHT

API TYPES	--
API FUNCTIONS	• void LIGHT_INIT (uint8 PIN , uint8 PORT) • void LIGHT_ON (uint8 PIN , uint8 PORT) • void LIGHT_OFF (uint8 PIN , uint8 PORT)
INCLUDES	• GPIO

LIGHT_INIT

ARGUMENTS	• uint8 PIN pin number connected to the light • uint8 PORT port id of pin connected to the light
RETURN	void
DESCRIPTION	This function Initializes light using GPIO module

LIGHT_ON

ARGUMENTS	• uint8 PIN pin number connected to the light • uint8 PORT port id of pin connected to the light
RETURN	void
DESCRIPTION	this function turns light ON

LIGHT_OFF

ARGUMENTS	• uint8 PIN pin number connected to the light • uint8 PORT port id of pin connected to the light
RETURN	void
DESCRIPTION	this function switches light OFF

BUZZER

API TYPES

--

API FUNCTIONS

- void BUZZER_INIT (uint8 CHANNEL , uint8 DUTY)
- void BUZZER_ON (uint8 CHANNEL , uint8 DUTY)
- void BUZZER_OFF (uint8 CHANNEL)

INCLUDES

- PWM

BUZZER_INIT

ARGUMENTS

- uint8 CHANNEL
  PWM channel connected to the buzzer
- uint8 DUTY
  duty cycle of buzzer

RETURN

void

DESCRIPTION

This function Initializes buzzer using PWM module

BUZZER_ON

ARGUMENTS	• uint8 CHANNEL PWM channel connected to the buzzer • uint8 DUTY duty cycle of buzzer
RETURN	void
DESCRIPTION	this function turns buzzer ON

BUZZER_OFF

ARGUMENTS	• uint8 CHANNEL PWM channel connected to the buzzer
RETURN	void
DESCRIPTION	this function switches buzzer OFF

BCM

API TYPES	• COMM_CONFIG • DATA
API FUNCTIONS	• void COMM_INIT (COMM_CONFIG* COMM) • void SEND (DATA* DATA_ARR , uint8 COMM_ID) • void RECEIVE (DATA* DATA_ARR , uint8 COMM_ID)
INCLUDES	• CAN

COMM_CONFIG

TYPE	STRUCTURE
RANGE	-
DESCRIPTION	This structure will post-build configurable parameters of communication protocol which will be passed to init function then .

DATA

TYPE	uint8 array
RANGE	-
DESCRIPTION	Array of bits holding data received or sent

COMM_INIT

ARGUMENTS

- COMM_CONFIG * COMM
  pointer to communication post-build configurations

RETURN

void

DESCRIPTION

This function Initializes communication

SEND

ARGUMENTS

- DATA * DATA_ARR
  pointer to data array to be sent
- uint8 COMM_ID
  communication ID

RETURN

void

DESCRIPTION

This function will send the data given to it over specified communication

RECEIVE

ARGUMENTS

- DATA * DATA_ARR
  pointer to data array to store received data
- uint8 COMM_ID
  communication ID

RETURN

void

DESCRIPTION

This function will receive the data from specified communication and store it in the array