Static Design

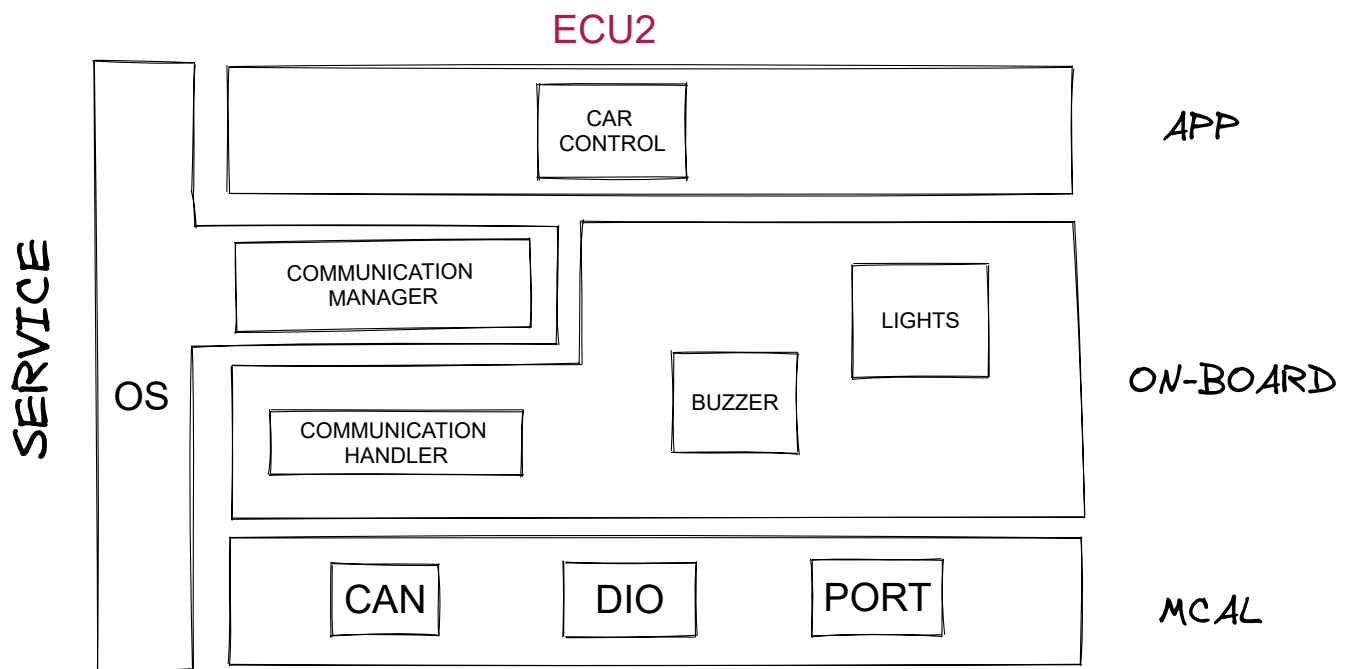## ECU2

Layered Architecture and Modules.

ECU2

- since I'm not sending or receiving data from any external hardware component that will be in the on-board layer, I can neglect the communication handler in this case
- Also, I only have on communication manager so there is no need to add the middleware layer
- Note: Vehicles use Hall effect sensors to determine speed. All a Hall effect sensor does is detect the presence of a magnetic field. So a magnet is mounted to whatever is spinning and the H.E. sensor is mounted to a stationary point. Every time the wheel makes a makes a revolution the magnet passes by the H.E. sensor and then the microprocessor does the math to determine the RPM which is then translated into MPH.

APIs details and Typedefs

CAN

1. APIs

Name	Can_Init
Syntax	`void Can_Init(const Can_ConfigType *config)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	Config
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Initialize the module

Name	Can_DeInit
Syntax	`void Can_DeInit(void)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Deinitialize the module

Name	Can_Read
Syntax	`void Can_Read(void)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Reads the data sent by ECU1 through the CAN protocol

Name	Can_ConfigType
Type	Struct
Description	A struct that holds all of the peripheral initialization

Port

1. APIs

Name	Port_Init
Syntax	`void Port_Init(Port_ConfigType *config)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	config
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Initialize the module

Name	Port_DeInit
Syntax	`void Port_DeInit(void)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Deinitialize the module

2. Typedefs

Name	Can_ConfigType
Type	Struct
Description	A struct that holds all of the peripheral initialization

DIO

1. APIs

Name	Dio_ReadChannel
Syntax	`Dio_LevelType Dio_ReadChannel(Dio_ChannelType channelId)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId
Parameters (inout)	None
Parameters (out)	None
Return Value	Dio_LevelType
Description	Read a specific MCU pin state

Name	Dio_WriteChannel
Syntax	`void Dio_WriteChannel(Dio_ChannelType channelId, Dio_LevelType level)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	channelId, level
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Set a specific pin level either to high or low

2. Typedefs

Name	Dio_ChannelType
Type	uint8
Description	This type identifies which pin on the MCU we need to write to or read from

Name	Dio_LevelType
Type	uint8
Description	This type identifies The state of the pin either high or low

Communication Manger

1. APIs

Name	commMgr_Read
Syntax	`void commMgr_Read(u8 Id, u8 Data)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	Id, Data
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	Chooses which bus or peripheral through which it reads the data

Lights

1. APIs

Name	lights_SetState
Syntax	`void lights_SetState(u8 Id, u8 level)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	Id, level

Name	lights_SetState
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	It sets the light determined by the provided ID to either on or off

Buzzer

1. APIs

Name	buzzer_SetState
Syntax	`void buzzer_SetState(u8 level)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	level
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	It sets the buzzer to either on or off

Car Control

1. APIs

Name	carCtrl_UpdateState
Syntax	`void carCtrl_UpdateState(void)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None

Name	carCtrl_UpdateState
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	A function That updates the internal car control module state with the newly received state from ECU1.