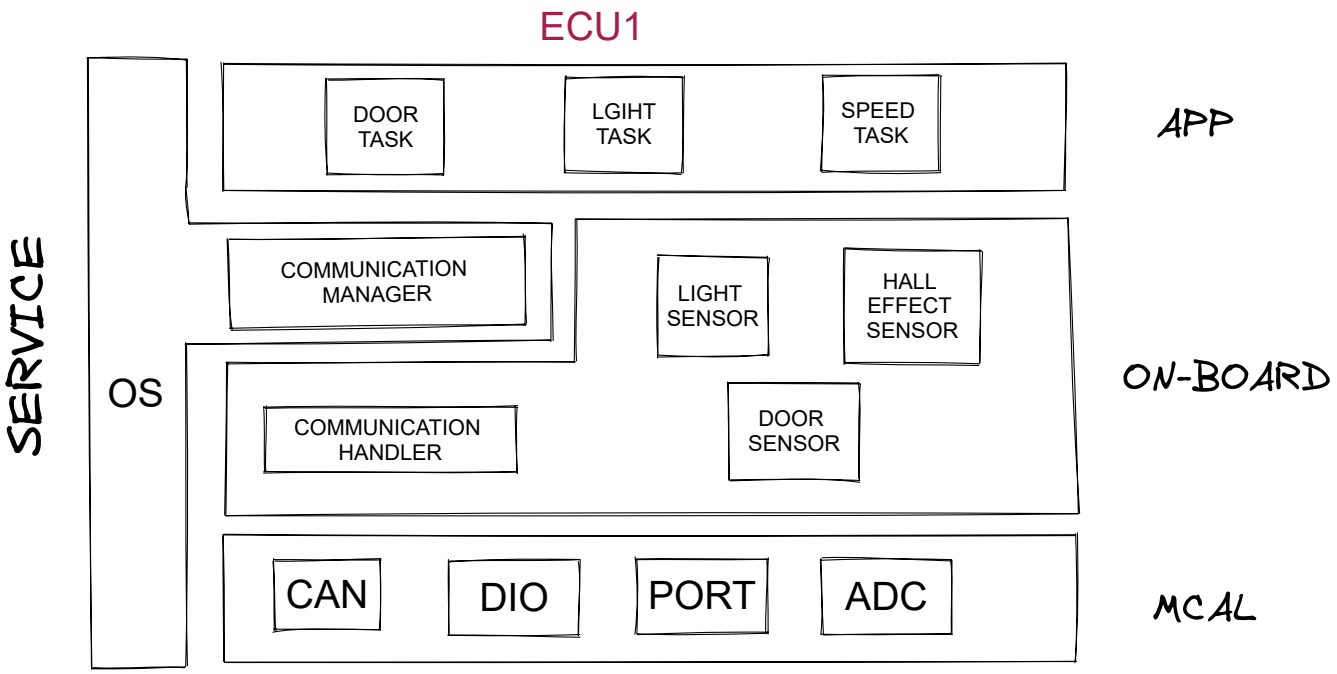
Static Design

ECU1

Layered Architecture and Modules.



- 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_PutString
Syntax	`void Can_PutString(const char *str)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	str
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	sends data through the CAN peripheral

2. Typedefs

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

---

ADC

1. APIs

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

Name	Adc_DeInit
Syntax	`void Adc_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	Adc_ReadData
Syntax	`u16 Adc_ReadData(u8 channel)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	channel
Parameters (inout)	None
Parameters (out)	None
Return Value	u16
Description	Get the current reading from the ADC

2. Typedefs

Name	uint8
Type	unsigned char
Description	A standard 8-bit value Type

Name	uint16
Type	unsigned short int
Description	A standard 16-bit value Type

Hall Effect Sensor

1. APIs

Name	Hes_GetValue
Syntax	`u16 Hes_GetValue(void)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None

Name	Hes_GetValue
Return Value	u16
Description	Gets the current value sensed by the Hall Effect Sensor

---

Door Sensor

1. APIs

Name	Ds_GetDoorState
Syntax	`Ds_StateType Ds_GetDoorState(void)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	Ds_StateType
Description	gets the current state of the door

Name	Ds_StateType
Type	enum
Description	This type identifies The state of the door either OPEN or CLOSED

Light Switch

1. APIs

Name	Ls_GetLightState
Syntax	`Ls_StateType Ls_GetLightState(void)`
Sync/Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None

Name	Ls_GetLightState
Parameters (inout)	None
Parameters (out)	None
Return Value	Ls_StateType
Description	Gets the current state of the light switch

Name	Ls_StateType
Type	enum
Description	This type identifies The state of the light switch either ON or OFF

Communication Manger

1. APIs

Name	commMgr_Send
Syntax	`void commMgr_Send(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 sends the data

---

Door Task

1. APIs

Name	task_SendDoorState
Syntax	`void task_SendDoorState(void)`
Sync/Async	Synchronous

Name	task_SendDoorState
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	A task that sends the door state through the can protocol to another ECU every 10ms

---

Light Task

1. APIs

Name	task_SendLightSwitchState
Syntax	`void task_SendLightSwitchState(void)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	A task that sends the light switch state through the can protocol to another ECU every 20ms

---

Speed Task

1. APIs

Name	task_SendSpeedState
Syntax	`void task_SendSpeedState(void)`
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in)	None
Parameters (inout)	None
Parameters (out)	None
Return Value	None
Description	A task that sends the speed state through the can protocol to another ECU every 5ms