CAN APIs:

Function Name	CAN_Init(void)		
Arguments	Inputs	NA	
		description	
	Outputs	NA	
		description	
	Inputs/Outputs	NA	
		description	
Return	E_OK	0	
	E_NOK	1	
Description	Call this API to Initialize CAN Module		

Function Name	CAN_SendBytes(Uint8 Byte)		
Arguments	Inputs	Byte	Uint8 (Char)
		Description: Byte Which Sent by CAN	
	Outputs	NA	
		description	
	Inputs/Outputs	NA	
		description	
Return	E_OK	0	
	E_NOK	1	
Description	Call this API to send Byte Using CAN Module		

Function Name	CAN_RecieveBytes(Uint8* Byte)		
Arguments	Inputs	NA	
		Description:	
	Outputs	Byte	Uint8*
		Description: Received Byte Store in Byte	
	Inputs/Outputs	NA	
		Description:	
Return	E_OK	0	
	E_NOK	1	
Description	Call this API to Receive Byte Using CAN Module		

DIO APIs:

Function Name	void Dio_WriteChannel (enu_Dio_ChannelType ChannelId, STD_LevelType Level)		
Arguments	Inputs	ChannelId	enu_Dio_ChannelType (enum)
		Level	STD_LevelType (enum)
		Description: Choose which port do u want and choose which level do u need (STD_low or STD_high)	
	Outputs	NA	
		description	
	Inputs/Outputs	NA	
		description	
Return	NA		
Description	Call this API to write on port and bin you choose		

<table>
<tr><th>Function Name</th><th colspan="3">STD_LevelType Dio_ReadChannel(enu_Dio_ChannelType ChannelId)</th></tr>
<tr><td rowspan="5">Arguments</td><td rowspan="2">Inputs</td><td>ChannelId</td><td>enu_Dio_ChannelType (enum)</td></tr>
<tr><td colspan="2">Description: Choose which port do you need to see its state</td></tr>
<tr><td rowspan="2">Outputs</td><td>NA</td><td></td></tr>
<tr><td colspan="2">Description:</td></tr>
<tr><td rowspan="2">Inputs/Outputs</td><td>NA</td><td></td></tr>
<tr><td colspan="2">Description:</td></tr>
<tr><td>Return</td><td>STD_LevelType</td><td colspan="2">Pin State</td></tr>
<tr><td>Description</td><td colspan="3">Call this API to Read state from port and bin you choose</td></tr>
</table>