Vehicle door handle control unit

The project aims to implement a vehicle door handle control unit that supports an anti-theft locking capability.

The system consists of the following components:

- A microcontroller unit.
- Two push buttons, one to simulate the handle lock/unlock button and the other one to simulate the door lock/unlock.
- Three LEDs, one to simulate the vehicle lock, one to simulate the hazard lights and one for the vehicle ambient light.
- All the push buttons are connected as active low.
- All the LEDs are connected as active high.

The system shall be implemented to handle the following use cases:

Use case	Initial state	inputs	Resultant System state	actions:
default state	System is powered off. Inputs:	System is powered on, no button is pressed, the switch is in the on state.	Vehicle door is locked.	The three LEDs are off.
door unlock	Vehicle door is locked.	Door handle button is pressed.	Vehicle door is unlocked but it is closed.	Vehicle lock LED is on. Hazard LED is blinking one time 0.5 second high and 0.5 second low for each blink, Ambient light LED is on for 2 seconds then off.
door is open	Vehicle door is unlocked and the door is closed.	Door unlock button is pressed.	Vehicle door is unlocked and the door is open.	Ambient light LED is on.

anti theft vehicle lock.	Vehicle door is unlocked and the door is closed.	No buttons pressed for 10 seconds.	Vehicle door is locked	Vehicle lock LED is off. Hazard LED is blinking two times 0.5 second high and 0.5 second low for each blink, Ambient light LED is off.
Closing the door	Vehicle door is unlocked and the door is open.	Door lock button is pressed	Vehicle door is unlocked and the door is closed	Vehicle lock LED is off. Hazard LED is off. Ambient light LED is on for one second and then is off.
Locking the door	Vehicle door is unlocked and the door is closed	Door handle lock button is pressed	Vehicle door is locked and the door is closed	Vehicle lock LED is off. Hazard LED is blinking two times 0.5 second high and 0.5 second low for each blink, Ambient light LED is off.

Detailed Deliverables:

You are required to implement a *GPIO driver* that supports the following functions:

1- void GPIO_Init(void)
- A function to initialize the GPIO registers with the needed initial values to support the connected hardware actions.

2- void GPIO_WritePinValue(unsigned char PinId, unsigned char PinData)
- A function to set a value (1 or 0) to a specific pin.

3- unsigned char GPIO_ReadPinState(unsigned char PinId)
- A function to return (1) if a falling edge is detected on the pin and (0) otherwise.

The GPIO driver shall be used to read the buttons' state and drive the LEDs' outputs.

You are required to implement a *GPT driver* that supports the following functions:
1- void GPT_Init(void)
   - A function to initialize the GPT registers with the needed initial values to support the needed timing actions.
2- void GPT_StartTimer(unsigned long int OverFlowTicks)
   - A function to request the GPT to start and send its number of tickets before timer overflow and stop.
3- unsigned char GPT_CheckTimeIsElapsed(void)
   - A function to return (1) if an overflow occurred after the last call of GPT_StartTimer and (0) if no overflow occurred or GPT_StartTimer is not called from the last read.
4- unsigned long int GPT_GetElapsedTime(void)
   - A function to return number of elapsed ticks from the last call of the GPT_StartTimer, 0 if it is not called and 0xffffffff if an overflow occurred.
5- unsigned long int GPT_GetRemainingTime(void)
   - A function to return number of remaining ticks till the overflow ticks passed to GPT_StartTimer, 0xffffffff if GPT_startTime is not called, 0 if an overflow occurred.

The GPT driver shall be used to handle all the time actions in the project. **(No delay functions will be used, and no hardware wait loops).**

You are required to use the implemented GPIO and GPT drivers to implement the different system use cases.

You are required to deliver the implementation source code, the simulation file and video showing that the code is running and explaining what is done in the project.

This project will be per team. Each team should have a minimum of 2 members and a maximum of 4 members.

Deadline: Day of embedded final exam at 11:59 pm.