STEP BY STEP PROCEDURE

Step 1: Set Up the Project

1. **Launch Quartus Prime**
    - Open Quartus Prime (v16.0 or later).
    - Go to **File > New Project Wizard**.
2. **Create a New Project**
    - Specify a **project directory** (avoid spaces in the path).
    - Name the project (e.g., SmartParkingSystem).
    - Select **Empty Project**.
3. **Select FPGA Device**
    - Choose:
        - **Family**: MAX 10
        - **Device**: 10M50DAF484C7G (matches DE10-Lite).
4. **Add Design Files**
    - Copy the SmartParkingSystem.v code into a new file:
        - **File > New > Verilog HDL File**, paste the code, and save as SmartParkingSystem.v.
    - Add the file to the project:
        - **Project > Add/Remove Files in Project**, then select the .v file.

Step 2: Assign Pin Locations

1. **Open the Pin Planner**
    - Go to **Assignments > Pin Planner**.
    - Enter the pin assignments from the table below (or import the .qsf file if you created one).

2. **Manual Pin Assignment**

Signal	FPGA Pin	Direction	I/O Standard
clk	PIN_P11	Input	3.3-V LVTTL
reset_n	PIN_B8	Input	3.3-V SCHMITT TRIGGER
echo	PIN_V10	Input	3.3-V LVTTL
trigger	PIN_W10	Output	3.3-V LVTTL
ir_sensors[0]	PIN_V9	Input	3.3-V LVTTL
ir_sensors[1]	PIN_W9	Input	3.3-V LVTTL
ir_sensors[2]	PIN_V8	Input	3.3-V LVTTL
ir_sensors[3]	PIN_W8	Input	3.3-V LVTTL
servo_pwm	PIN_V7	Output	3.3-V LVTTL
red_led	PIN_A8	Output	3.3-V LVTTL
green_led	PIN_A9	Output	3.3-V LVTTL

3. **Save Assignments**
    - Go to **Assignments > Save Assignment** (or press Ctrl+S).

Step 3: Compile the Design

1. **Start Compilation**
    - Click the **Start Compilation** button (blue triangle) or go to **Processing > Start Compilation**.
    - Wait for the compilation to complete (check the **Messages** window for errors).
2. **Fix Errors (if any)**

- Common issues:
    - Missing pin assignments.
    - Syntax errors in Verilog code.
    - Incorrect device selection.

Step 4: Program the FPGA

1. **Connect the DE10-Lite**
    - Plug in the USB cable (to the **USB Blaster** port on the board).
    - Power on the board (via USB or external 5V supply).
2. **Open the Programmer**
    - Go to **Tools > Programmer**.
    - Ensure the **USB-Blaster** is selected under **Hardware Setup**.
3. **Add the** .sof **File**
    - Click **Add File**, navigate to the project's output directory (output_files/), and select SmartParkingSystem.sof.
    - Check the **Program/Configure** box.
4. **Program the Board**
    - Click **Start**. The FPGA will now run your design.

Step 5: Test the System

1. **Hardware Setup**
    - Connect:
        - Ultrasonic sensor to GPIO_0 (echo) and GPIO_1 (trigger).
        - IR sensors to GPIO_2 to GPIO_5.
        - Servo to GPIO_6 (PWM) and VCC5 (power).
2. **Verify Operation**
    - **LEDs**:
        - LEDR0 (red) = No slots available.

        - LEDR1 (green) = Slots available.
    - **Servo**: Should rotate to open/close the gate when a vehicle is detected.
    - **Serial Monitor**: Use a UART terminal (optional) for debug prints if added to the code.