Marmara University
Faculty of Engineering

EE2003 Digital Design Project Report

Obstacle Detection Module

	Dept		Name	Surname	Task Distribution
1.	EEE		Canberk	Suner	System Architecture & VHDL Coding
2.	EEE		Ömer Faruk	Kocaoğlu	Simulation & Verification
3.	EEE		Resul	Argın	Hardware Integration & Testing
4.	EEE		Yusuf	Polat	Documentation & Presentation

Table of Contents

1. LIST OF FIGURES AND TABLES 3

1.1 List of Figures 3

1.2 List of Tables 3

2. PROJECT OVERVİEW 3

3. SYSTEM ARCHITECTURE AND TECHNICAL SPECIFICATIONS 3

3.1. Sensor Interface and Distance Calculation Logic 3

3.2. Communication Protocols and Display Logic 4

3.3. Control System and Wiring Configuration 4

3.4. Simulation and Verification 5

4. WIRING DIAGRAM 7

4.1. Hardware İmplamentation Notes 7

5. FSM OF THE PROJECT 8

5.1. System Power Mode FSM (Push-Button Control) 8

5.2. State Definitions and Output Management 8

5.3. State Transition and Output Table 9

6. DETAILED PROJECT EXPLANATION VIDEO 10

1. LIST OF FIGURES AND TABLES

1.1 List of Figures

Figure 1: *Obejct is on Warning State*

Figure 2: *Obejct is on Danger State*

Figure 3: *Obejct is on Safe State*

Figure 4: *Wiring Diagram of Project*

1.2 List of Tables

Table 1: State Table of Project

2. PROJECT OVERVİEW

Trying to reverse park a vehicle into a narrow garage in a poorly lit area with limited visibility is generally a difficult and stressful task for anyone. The driver cannot clearly see the low concrete barrier on the rear wall in the mirrors. Without an assistance system, the driver has to estimate the distance, which increases the risk of collision.

This project was developed to design and implement a digital parking sensor system using an FPGA card to assist drivers in tight spaces or under limited visibility conditions. The system will measure the distance between the vehicle and obstacles using ultrasonic sensors and provide various alerts accordingly.

The primary goal is to enhance safety by providing real-time feedback. The system will prevent collisions by alerting the user through visual indicators, audible warnings, and a digital distance display.

3. SYSTEM ARCHITECTURE AND TECHNICAL SPECIFICATIONS

This section details the hardware components, VHDL logic implementation, communication protocols, and pin configurations used in the Obstacle Detection Module.

3.1. Sensor Interface and Distance Calculation Logic

The system uses the HC-SR04 ultrasonic sensor, interfaced via the Basys 3 PMOD JA port[1]. The distance calculation is performed within the VHDL architecture using high-precision counters based on the speed of sound.

- **Operating Principle:** The sensor emits an ultrasonic pulse upon receiving a trigger signal. The duration of the returning echo pulse is measured by the FPGA.
- **Mathematical Model:**
  - ***System Clock:*** 100 MHz (10 ns period).
  - ***Physics:*** Sound waves take approximately $58 \mu s$ to travel 1 cm (round trip). This corresponds to 5800 clock cycles at 100 MHz.

    - ***Implementation:*** To achieve high precision, the VHDL code uses a pre-scaler constant C_CLKS_PER_UNIT = 58. The distance counter increments every 58 clock cycles. This design choice results in a measurement resolution of **0.01 cm (0.1 mm)**, allowing for precise integer arithmetic without floating-point processing.

- **Voltage Levels:** Since the HC-SR04 operates at 5V while the Basys 3 FPGA input pins accept 3.3V, the sensor's echo signal is voltage-divided or directly compatible via the current-limiting resistors inherent in PMOD ports, ensuring safe logic levels.

3.2. Communication Protocols and Display Logic

The system implements custom state machines to drive the visual output modules, ensuring synchronized feedback.

- **LCD Interface (4-Bit Parallel Mode):**
    - To optimize FPGA pin usage, the LCD is connected to the **PMOD JB** header and driven in **4-bit mode**.
    - The VHDL Finite State Machine (FSM) splits 8-bit ASCII characters into two 4-bit "nibbles" (Upper and Lower). These are transmitted sequentially via pins lcd_d[7:4], controlled by the lcd_rs (Register Select) and lcd_en (Enable) signals.
    - ***Protocol Flow:*** Set Data -> Pulse Enable -> Wait -> Send Next Nibble. This allows the text "Mesafe: XX.XX cm" to be displayed dynamically.
- **7-Segment Display (Time-Multiplexing):**
    - The onboard 4-digit 7-segment display shows the distance in centimeters.
    - Since all digits share common cathode lines, the system uses a **Time-Multiplexing** technique. A refresh counter (refresh_counter) cycles through the anodes (an[3:0]) at a high frequency (>60 Hz), making all digits appear simultaneously lit to the human eye due to persistence of vision.

3.3. Control System and Wiring Configuration

The project is controlled by a Moore-type Finite State Machine managed by a single push-button input on pin L3.

- **Single-Button Mode Control:**

The system utilizes a single button with a software-based debounce filter (5ms delay) to cycle through four operational power modes:

1. **Standby (Mode 0):** System OFF.
2. **Full Active (Mode 1):** Both LCD and 7-Segment displays are ON.
3. **LCD Only (Mode 2):** 7-Segment is disabled to save power.
4. **7-Segment Only (Mode 3):** LCD is disabled.

- **Peripheral Wiring (XDC Mapping):**
  - **Sensor & Alerts (PMOD JA):** The Trigger (N17), Echo (P18), Buzzer (M2), and RGB Status LEDs (Green M1, Red K3, Yellow M3) are mapped to the JA header.
  - **LCD (PMOD JB):** The LCD control and data lines are mapped to the JB header (J1 through G3).
  - **Power:** External components are powered via a regulated 5V source (represented by Arduino in diagrams) sharing a Common Ground with the FPGA to ensure signal integrity.

3.4. Simulation and Verification

The VHDL code was verified using Vivado's logic simulator before hardware synthesis.

- **Distance Logic Verification:** The simulation waveform confirmed that the sonic_dist counter increments correctly every 58 clock cycles when the echo signal is high.
- **FSM Transitions:** The button debounce logic and state transitions (Mode 0 to Mode 3) were validated to ensure no false triggers occurred due to mechanical switch bounce.

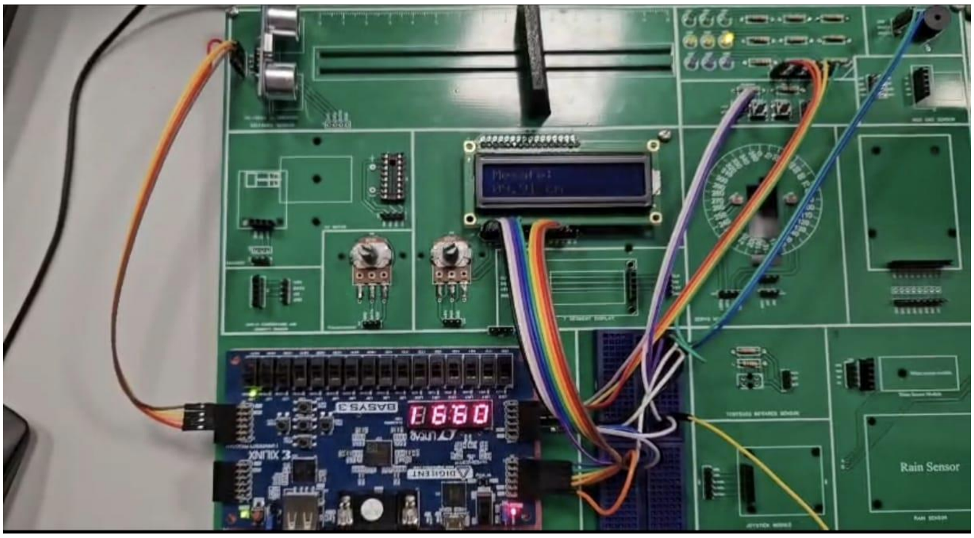

Figure 1: Obejct is on Warning State

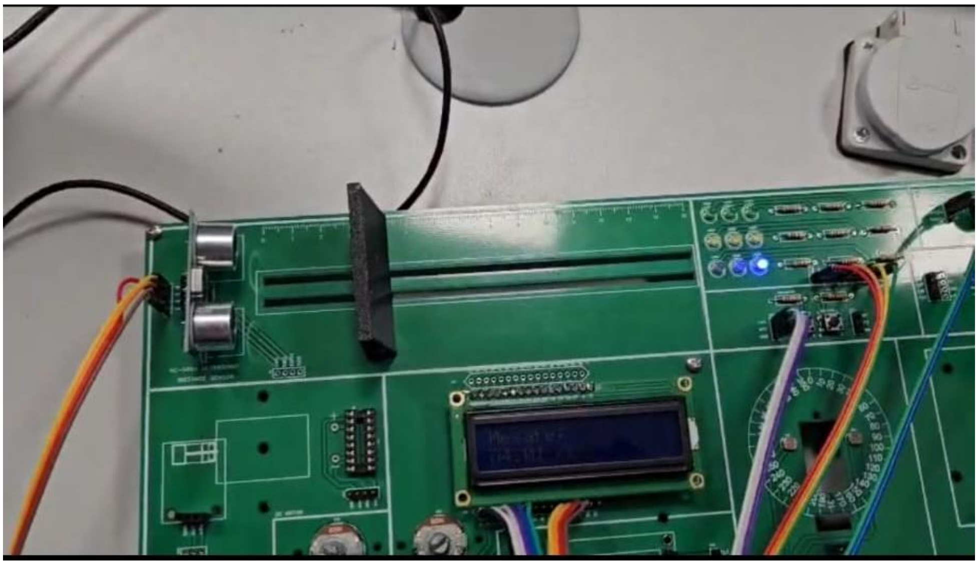

Figure 2: Obejct is on Danger State

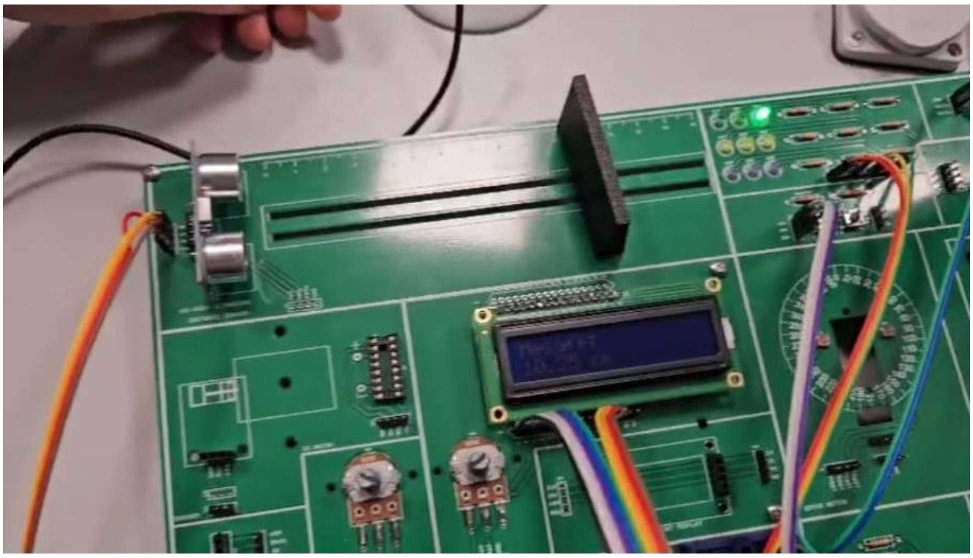

Figure 3: Obejct is on Safe State

4. WIRING DIAGRAM

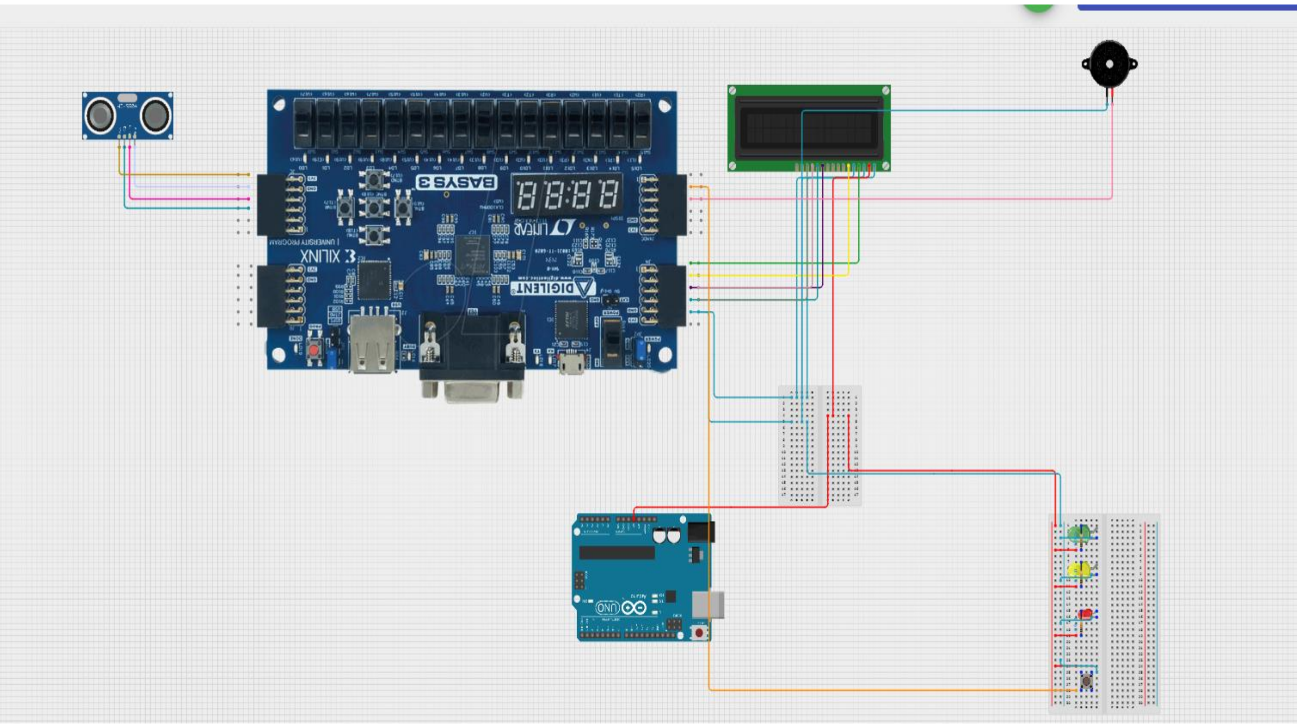

Figure 4: Wiring Diagram of Project

4.1. Hardware İmplamentation Notes

- **Power Supply (5V Source):** Although the system design typically utilizes power from an STM board, an Arduino Uno is used in this diagram as a representative 5V power source for the external components.
- **PCB Representation:** For clarity in this diagram, the LEDs and the push-button are shown on a breadboard; however, in the physical implementation, these components are integrated onto a custom PCB.
- **Common Ground and VCC:** To ensure signal integrity and stable operation, all modules—including the LCD, the buzzer, and the LED network—share a common VCC (5V) line and a common Ground (GND).

5. FSM OF THE PROJECT

The project is governed by a Moore-type Finite State Machine (FSM) that manages both the system's operational modes via a push-button and the distance-based alert logic. In a Moore machine, the outputs (display states and alert levels) are determined solely by the current state of the system.

5.1. System Power Mode FSM (Push-Button Control)

The top-level controller manages the active state of the hardware interfaces through a single push-button input. It cycles through four distinct operational modes with each press:

- **Mode 1 (Full Operation):** All systems, including the HC-SR04 sensor, LEDs, Buzzer, 7-Segment display, and LCD, are fully functional.
- **Mode 2 (7-Segment Power Save):** The 7-segment display is disabled to reduce power consumption while all other modules (LCD, sensor, and alerts) remain active.
- **Mode 3 (LCD Power Save):** The LCD screen is disabled, while the 7-segment display and alert systems continue to provide real-time feedback.
- **Mode 4 (Standby/OFF):** All outputs and displays are turned off, placing the system in a low-power standby state.

5.2. State Definitions and Output Management

The FSM continuously receives the distance data in centimeters and updates the following outputs accordingly:

- **st_SAFE (Safe State):** Distance is between 10 cm and 15 cm.
  - ***Displays:*** The real-time distance is shown in centimeters on both the 7-segment and LCD screens.
  - ***LED:*** Green LED is constant ON.
  - ***Buzzer:*** Silent (OFF).
- **st_WARNING (Warning State):** Distance is between 5 cm and 10 cm.
  - ***Displays:*** The real-time distance is shown in centimeters on both screens.
  - ***LED:*** Yellow LED blinks at a low frequency (1 Hz).
  - ***Buzzer:*** Active at low frequency.
- **st_DANGER (Danger State):** Distance is between 0 cm and 5 cm.
  - ***Displays:*** The real-time distance is shown in centimeters on both screens.
  - ***LED:*** Red LED blinks at a high frequency (4 Hz).
  - ***Buzzer:*** Active at high frequency.

5.3. State Transition and Output Table

Current State	**Condition (Distance D)**	**Next State**	**Output Behavior**	**Display Output**
SAFE	D > 10	SAFE	Green LED ON	Distance (cm)
SAFE	5 < D ≤ 10	WARNING	Transition to Warning	Distance (cm)
WARNING	5 < D ≤ 10	WARNING	Yellow LED Blink, Buzzer Low Freq	Distance (cm)
WARNING	0 ≤ D ≤ 5	DANGER	Transition to Danger	Distance (cm)
WARNING	D > 10	SAFE	Transition to Safe	Distance (cm)
DANGER	0 ≤ D ≤ 5	DANGER	Red LED Blink, Buzzer High Freq	Distance (cm)
DANGER	D > 5	WARNING	Transition to Warning	Distance (cm)

Table 1: State Table of Project

6. DETAILED PROJECT EXPLANATION VIDEO

https://www.youtube.com/watch?v=HMlnp5bxW-0