Rex Odomero Oghenerobo

Technical Report Embedded Systems Projects

CENTRIA UNIVERSITY OF APPLIED SCIENCES  
March 2025

**Table of Contents**

1. **I**ntroduction
2. Project 1: Multi-Sensor Monitoring System
   * 2.1 Hardware Setup
   * 2.2 Code Implementation
   * 2.3 Challenges & Solutions
3. Project 2: 74HC595 Shift Register with 8 LEDs
   * 3.1 Hardware Setup
   * 3.2 Code Implementation
   * 3.3 Challenges & Solutions
4. Comparative Analysis
5. Conclusion
6. References

**1. Introduction**

This report consolidates two embedded systems projects:

* Project 1: A multi-sensor system (HC-SR04, PIR, LED, pushbutton) for distance and motion monitoring.
* Project 2: An 8-LED control system using a 74HC595 shift register to minimize I/O pins.

Both projects were simulated in Tinkercad® and implemented with Arduino Uno.

**2. Project 1: Multi-Sensor Monitoring System**

**2.1 Hardware Setup**

Components:

A screenshot of a computer

Description automatically generated

* Arduino Uno, HC-SR04 (distance), PIR (motion), LED (visual feedback), pushbutton (mode control).

**Circuit Diagram:** A screenshot of a computer

Description automatically generated

**Improvements:**

* Added capacitor (0.1µF) to PIR sensor to reduce noise.

**2.2 Code Implementation**

Key Features:

* Distance measurement (getDistance() function).
* Motion-triggered LED alerts.
* Button-activated calibration mode.

**Code Snippet:**

int LedR = 5; // The LED is assigned to digital pin 5

int Nappi = 7; // The push button is assigned to digital pin 7

int PIR = 2; // The PIR motion sensor is assigned to digital pin 2

void setup() {

pinMode(LedR, OUTPUT); // The LED pin is configured as an OUTPUT

pinMode(Nappi, INPUT); // The Button pin is configured as an INPUT

pinMode(PIR, INPUT); // The PIR pin is configured as an INPUT

}

void loop() {

if (digitalRead(PIR)) { // Motion Detector

digitalWrite(LedR, HIGH); // Turns the LED ON

while(digitalRead(PIR)); //The LED is kept ON until motion stops

delay(4000); // The LED stays ON 4 seconds after there is no more motion

}

if (!digitalRead(Nappi) && digitalRead(LedR)) { // Checks whether when the button is pressed the LED is ON

digitalWrite(LedR, LOW); // The LED is Turned OFF

delay(1000); // Ignores delay

while(digitalRead(Nappi)); // Waits until button is released

delay(20); // Slight delay after release

}

if (!digitalRead(Nappi) && !digitalRead(LedR)) { // Checks whether when the button is pressed the LED is ON

digitalWrite(LedR, HIGH); //Turns the LED ON

delay(1000); // Ignores delay

while(digitalRead(Nappi)); //Waits until button is released

delay(20); //Little delay after release

}

}

A computer screen shot of a computer

Description automatically generated

**2.3 Challenges & Solutions**

| Challenge | Solution |
| --- | --- |
| PIR false triggers | Added capacitor and delay(500ms) |
| Button debouncing | Used delay() (library recommended for robustness) |

**3. Project 2: 74HC595 Shift Register with 8 LEDs**

**3.1 Hardware Setup**

Components:

A screenshot of a computer

Description automatically generated

* Arduino Uno, 74HC595 shift register, 8 LEDs (with 220Ω resistors), jumper wires.

**Circuit Diagram:**  
A computer screen shot of a computer

Description automatically generated

**Key Connections:**

| 74HC595 Pin | Arduino Pin | Purpose |
| --- | --- | --- |
| DS (14) | 4 | Serial data input |
| SHCP (11) | 5 | Clock pulse |
| STCP (12) | 6 | Latch |

Power:

* VCC: 5V | GND: Common ground.

**3.2 Code Implementation**

Key Logic:

* Used shiftOut() to send data serially to the shift register.
* Enabled control of 8 LEDs with only 3 Arduino pins.

**Code Snippet:**

int trigPin = 11;

int echoPin = 12;

long viive;

long cm;

void setup() {

Serial.begin(9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

}

void loop() {

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

pinMode(echoPin, INPaUT);

viive = pulseIn(echoPin, HIGH);

cm = (viive/2) / 29.1;

Serial.print(cm);

Serial.print(" cm");

Serial.println();

delay(1000);

}

**Serial Monitor Interaction:**  
A screen shot of a computer

Description automatically generated

**3.3 Challenges & Solutions**

| Challenge | Solution |
| --- | --- |
| LED flickering | Added latch delay (100µs) |
| Incorrect LED sequence | Verified bit order (LSBFIRST) |

**4. Comparative Analysis**

| Aspect | Project 1 (Sensors) | Project 2 (Shift Register) |
| --- | --- | --- |
| I/O Efficiency | Uses multiple pins | Saves pins (3 vs. 8) |
| Complexity | Moderate (sensor fusion) | Low (serial data control) |
| Applications | Security systems | Display panels |

**5. Conclusion**

* Project 1 demonstrated robust multi-sensor integration with real-time feedback.
* Project 2 showcased efficient I/O management using a shift register.
* Recommendation: Explore combining both projects (e.g., shift register for sensor status LEDs).

**6. References**

1. Arduino Documentation. (2025). shiftOut() function.
2. HC-SR04 Datasheet.