Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

Division of Electronics and Communication Engineering
2024-2025 (EVEN SEM)

REPORT

for

SYSTEM VERILOG FOR FUNCTIONAL VERIFICATION

Title of the Report: Real-Time Motion Tracking Using MPU6050 and I2C LCD Display

A report submitted by

Name of the Student	***ANNIE REACHEL.A, KEVIN PHILIP ABRAHAM***
Register Number	**URK22EC1051, UTK22EC1001**
Subject Name	***System Verilog for functional verification***
Subject Code	***19EC2018***
Date of Report submission	***28/03/2025***

Total marks: _____/ 10 Marks

Signature of Faculty with date:

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE NO.
1.	INTRODUCTION	3
2.	IMPLEMENTATION	4
3.	WORKING	7
4.	OUTPUT	9
5.	CONCLUSION	10

CHAPTER 1

INTRODUCTION

This project focuses on the interfacing of an MPU6050 sensor and an I2C LCD display with Aries IOT V2.0 to measure and display real-time acceleration values along the X, Y, and Z axes. The MPU6050 is a compact motion-tracking device that integrates a 3-axis accelerometer and a 3-axis gyroscope, making it widely used in applications requiring motion sensing, orientation tracking, and stability control. The sensor communicates with the microcontroller using the I2C (Inter-Integrated Circuit) protocol, which simplifies connections and ensures efficient data transmission.

The raw data from the sensor is processed to calculate the acceleration values in gravitational units (g) and displayed on a 16x2 LCD screen. The clear and continuous display of these values enables easy monitoring of real-time motion. This setup is commonly used in various fields, including robotics, wearable devices, gaming, and motion-based control systems, providing accurate and reliable motion detection. Additionally, the simplicity of the I2C interface makes it a cost-effective solution for applications requiring real-time motion analysis.

CHAPTER 2

IMPLEMENTATION

CODE:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
TwoWire Wire(1);
LiquidCrystal_I2C lcd(0x3F, 16, 2);
const int MPU_addr = 0x68;

void setup() {
  lcd. init();
  lcd.backlight();
  Serial.begin(115200);
  Serial.println("Initializing MPU and
LCD...");
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0x00);
  Wire.endTransmission(true);
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x1B);
  Wire.write(0x10);
  Wire.endTransmission(true);
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x1C);
  Wire.write(0x00);
  Wire.endTransmission(true);
  Serial.println("Initialization Complete.");
}
void loop() {
  float XAxisFinal, YAxisFinal,
ZAxisFinal;
  XAxisFinal = readAxis(0x3B);
  YAxisFinal = readAxis(0x3D);
  ZAxisFinal = readAxis(0x3F);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("X:"); lcd.print(XAxisFinal);
lcd.print("g");
  lcd.setCursor(0, 1);
  lcd.print("Y:"); lcd.print(YAxisFinal);
lcd.print("g Z:");
  lcd.print(ZAxisFinal); lcd.print("g");
  Serial.print("X Axis = ");
Serial.print(XAxisFinal);
Serial.println("g");
  Serial.print("Y Axis = ");
Serial.print(YAxisFinal);
Serial.println("g");
  Serial.print("Z Axis = ");
Serial.print(ZAxisFinal);
Serial.println("g");
  delay(1000);
}
float readAxis(byte registerAddress) {
  int16_t axisFull;
  Wire.beginTransmission(MPU_addr);
  Wire.write(registerAddress);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr, 2, true);
  axisFull = Wire.read() << 8 | Wire.read();
  return (float)axisFull / 16384.0;  }
```

CHATPER 3

WORKING

Working of the MPU6050 and I2C LCD Display Project

- This project involves interfacing an **MPU6050** accelerometer and gyroscope sensor with a microcontroller (e.g., Arduino) using the **I2C (Inter-Integrated Circuit)** communication protocol. The goal is to measure the real-time acceleration values along the X, Y, and Z axes and display them on a **16x2 I2C LCD display**.

1.Initialization Phase

Wire Library and I2C Setup:

- ➢ The code uses the Wire library to handle I2C communication.
- ➢ A TwoWire object is created for communication.
- ➢ The LCD is initialized using the LiquidCrystal_I2C library with the address 0x3F (commonly used for I2C LCD modules).

MPU6050 Initialization:

- ➢ The MPU6050 sensor is powered on by writing to its **Power Management Register (0x6B)**.
- ➢ The sensor's gyroscope is set to ±1000°/s by writing to the **Gyroscope Configuration Register (0x1B)**.
- ➢ The accelerometer is configured with a sensitivity of ±2g by writing to the **Accelerometer Configuration Register (0x1C)**.

LCD Initialization:

- ➢ The LCD is cleared, backlight is turned on, and a message is printed using the lcd.init() and lcd.backlight() functions.

2. Reading Accelerometer Data

- The function readAxis() reads the data from the MPU6050 sensor using I2C communication.
- The sensor provides 16-bit raw data for each axis using the following registers:
- **0x3B** for X-axis
- **0x3D** for Y-axis
- **0x3F** for Z-axis
- Using Wire.requestFrom(), the microcontroller reads two bytes of data for each axis. These bytes are combined using bitwise operations (<< 8 |) to form a 16-bit signed integer value.
- The value is then converted to acceleration in terms of **g (gravitational units)** using the formula:

$$\text{Acceleration (g)} = \frac{\text{Raw Data}}{16384.0}$$

- Since the MPU6050 has a sensitivity of ±2g, the divisor 16384.0 is used.

3. Displaying the Data

- The real-time X, Y, and Z-axis acceleration values are displayed on the LCD using the lcd.print() function.
- The LCD has two rows with 16 columns, so the X-axis value is displayed on the first row and both Y and Z-axis values on the second row.
- The data is also printed to the Serial Monitor using Serial.print() for real-time debugging or monitoring.

4. Delay and Continuous Reading

- The delay(1000) function introduces a one-second delay, ensuring that the sensor data is updated every second.
- The loop then repeats, continuously reading and displaying new acceleration values.

CHATPER 4

OUTPUT

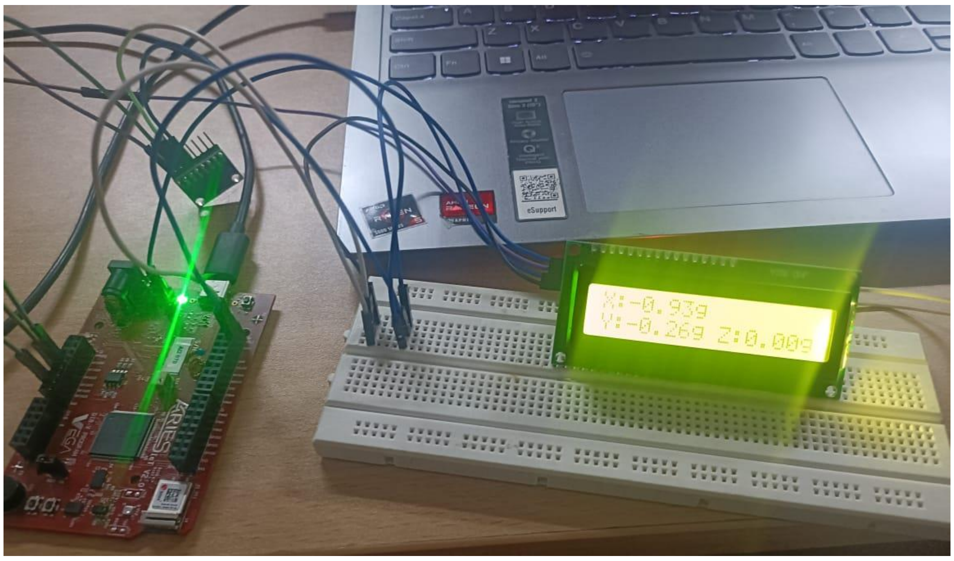

Fig:MPU6050 values when idle.

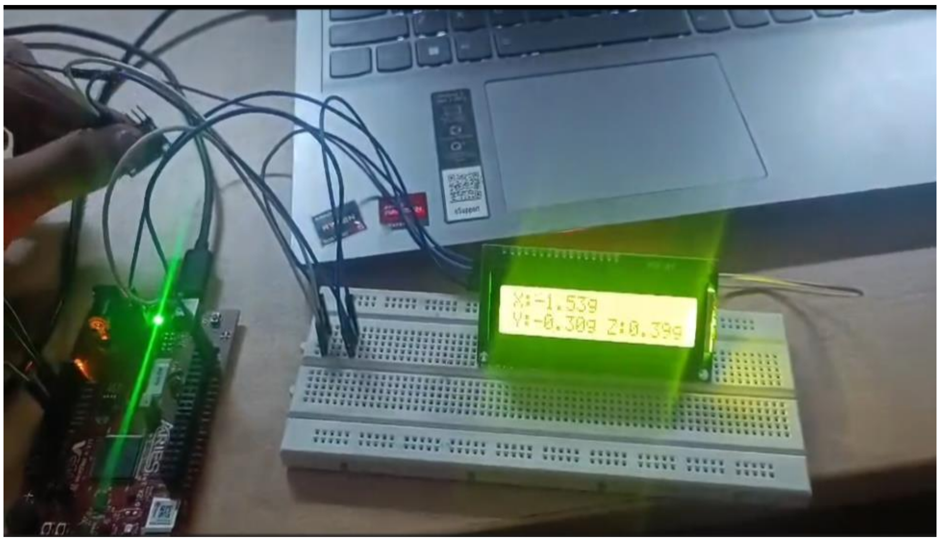

Fig: MPU6050 values during movement.

CHATPER 5
CONCLUSION

This project successfully demonstrates the real-time measurement and display of acceleration data from an MPU6050 sensor using an I2C LCD, effectively interfaced via the Aries IOT V2.0 platform. The implementation utilized the I2C protocol, processing raw sensor data into gravitational units for clear visualization on the LCD and serial monitor. The Aries IOT V2.0's robust architecture and efficient handling of I2C communication facilitated seamless integration, showcasing its capability in embedded motion-sensing applications.

The successful integration of the MPU6050 and I2C LCD demonstrates the platform’s capacity to support diverse sensor interfaces and data processing needs, making it a powerful tool for developing applications in robotics, wearables, and industrial automation. By leveraging the Aries IOT V2.0's features, this project underscores its potential to streamline the development of motion-sensing systems and accelerate the deployment of IoT solutions that rely on accurate and real-time motion data.