Ex:No:11	**Implement I2C communication protocol between Raspberry Pi Pico and a peripheral device**
Date:	

Objective:

To Implement I2C communication protocol between Raspberry Pi Pico and MPU6050 IMU sensor

Components Required:

S.No	Component	Specification
1	Raspberry Pi Pico	Microcontroller board (RP2040)
2	MPU6050 Module	3-axis accelerometer and gyroscope
3	Jumper Wires	Male-to-female/female-to-female
4	Breadboard	For making temporary connections
5	Micro USB Cable	For flashing and serial communication
6	PC/Laptop	For code development and flashing

Circuit Connections Required:

Pico Pin	MPU6050 Pin	Description
GPIO 4 -	SDA	I2C Data
GPIO 5 -	SCL	I2C Clock
3.3V -	VCC	Power
GND -	GND	Ground

Program:

MPU6050 Header file:

```
#ifndef MPU6050_H
#define MPU6050_H

#include "hardware/i2c.h"

// MPU6050 default I2C address
#define MPU6050_ADDR            0x68

// MPU6050 register addresses
#define MPU6050_REG_PWR_MGMT_1   0x6B
#define MPU6050_REG_WHO_AM_I     0x75
#define MPU6050_REG_ACCEL_XOUT_H 0x3B

// Sensitivity scale factor for ±2g (default)
#define MPU6050_ACCEL_SCALE      16384.0

// I2C port used
#define MPU6050_I2C_PORT i2c0

// Function to initialize MPU6050
```

```
void mpu6050_init(void);

// Function to read raw accelerometer data
void mpu6050_read_accel(float *x, float *y, float *z);

#endif
```

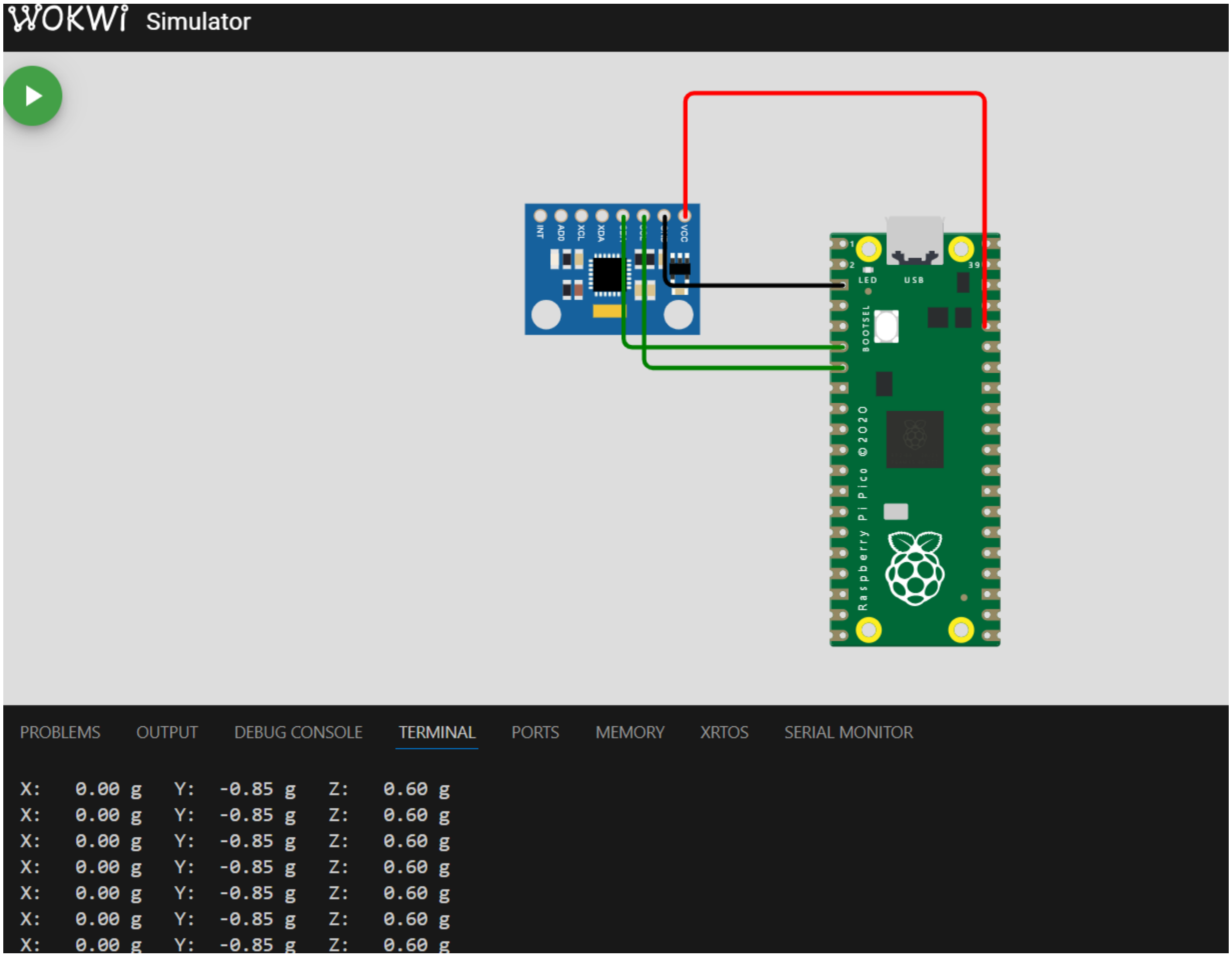
isquared.c

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/i2c.h"

#define MPU6050_ADDR 0x68
#define I2C_PORT i2c0

void mpu6050_init(void) {
    sleep_ms(1000); // Allow MPU6050 to power up

    // WHO_AM_I check
    uint8_t who_am_i_reg = 0x75;
    uint8_t who_am_i = 0;
    i2c_write_blocking(I2C_PORT, MPU6050_ADDR, &who_am_i_reg, 1, true);
    i2c_read_blocking(I2C_PORT, MPU6050_ADDR, &who_am_i, 1, false);

    if (who_am_i != 0x68) {
        while (1) {
            printf("MPU6050 not found! WHO_AM_I = 0x%02X\n", who_am_i);
            sleep_ms(2000);
        }
    }

    // Wake up device (write 0 to PWR_MGMT_1)
    uint8_t init_data[] = {0x6B, 0x00};
    i2c_write_blocking(I2C_PORT, MPU6050_ADDR, init_data, 2, false);
    printf("MPU6050 initialized successfully\n");
}

int main(void) {
    stdio_init_all();
    sleep_ms(1000); // Wait for USB serial to connect

    i2c_init(I2C_PORT, 400000);
    gpio_set_function(4, GPIO_FUNC_I2C); // SDA
    gpio_set_function(5, GPIO_FUNC_I2C); // SCL
    gpio_pull_up(4);
    gpio_pull_up(5);

    mpu6050_init();

    uint8_t reg = 0x3B; // Start of accelerometer data
    uint8_t accel_data[6];
    int16_t accelX, accelY, accelZ;
    float f_accelX, f_accelY, f_accelZ;

    while (1) {
        i2c_write_blocking(I2C_PORT, MPU6050_ADDR, &reg, 1, true);
        i2c_read_blocking(I2C_PORT, MPU6050_ADDR, accel_data, 6, false);

        accelX = (accel_data[0] << 8) | accel_data[1];
        accelY = (accel_data[2] << 8) | accel_data[3];
        accelZ = (accel_data[4] << 8) | accel_data[5];

        f_accelX = accelX / 16384.0;
        f_accelY = accelY / 16384.0;
        f_accelZ = accelZ / 16384.0;

        printf("X: %6.2f g   Y: %6.2f g   Z: %6.2f g\n", f_accelX, f_accelY, f_accelZ);
        sleep_ms(300);
    }
}
```

main.c

```
#include <stdio.h>
#include "pico/stdlib.h"
#include "hardware/i2c.h"
#include "mpu6050.h"

int main(void) {
    stdio_init_all();

    // Initialize I2C on GPIO4 (SDA) and GPIO5 (SCL)
    i2c_init(MPU6050_I2C_PORT, 400000);
    gpio_set_function(4, GPIO_FUNC_I2C);
    gpio_set_function(5, GPIO_FUNC_I2C);
    gpio_pull_up(4);
    gpio_pull_up(5);

    // Initialize MPU6050
    mpu6050_init();
    printf("MPU6050 initialized successfully.\n");

    float accelX, accelY, accelZ;

    while (1) {
        mpu6050_read_accel(&accelX, &accelY, &accelZ);

        printf("Accel X: %6.2f g | Y: %6.2f g | Z: %6.2f g\n", accelX, accelY, accelZ);
        sleep_ms(500);
    }

    return 0;
}
```

OUTPUT:



```
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
X:   0.00 g   Y:  -0.85 g   Z:   0.60 g
```

INFERENCE:

- This project successfully demonstrates **I2C communication** between the **Raspberry Pi Pico** and **MPU6050**.
- Real-time **acceleration data** is acquired, converted to physical units, and monitored over USB.
- It can be extended for applications like fall detection, motion sensing, tilt measurement, etc.

RESULT:

- The MPU6050 sensor starts reading raw accelerometer values.
- These values are scaled into **g-force** values and printed via USB serial.