Week 3 Assignment 2

Created	@December 16, 2024 5:23 PM
Class	RoboDive

2.2) MPU 6050

CODE

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu; // Create an object for the MPU6050

void setup()
{
  Serial.begin(115200);

  Serial.println("Initialize MPU6050");
  while (!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_2G))
  {
    Serial.println("Could not find a valid MPU6050 sensor, checl
    delay(500);
  }

  mpu.calibrateGyro(); // Calibrate the gyroscope
  mpu.setThreshold(3); // Set a threshold to filter noise
  checkSettings();
}
```

```
// Function to display sensor settings
void checkSettings()
{
  Serial.println();

  Serial.print(" * Sleep Mode:        ");
  Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

  Serial.print(" * Clock Source:      ");
  switch (mpu.getClockSource())
  {
    case MPU6050_CLOCK_KEEP_RESET: Serial.println("Stops the clo
    case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with
    case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with
    case MPU6050_CLOCK_PLL_ZGYRO: Serial.println("PLL with Z axi
    case MPU6050_CLOCK_PLL_YGYRO: Serial.println("PLL with Y axi
    case MPU6050_CLOCK_PLL_XGYRO: Serial.println("PLL with X axi
    case MPU6050_CLOCK_INTERNAL_8MHZ: Serial.println("Internal 8
  }

  Serial.print(" * Gyroscope:         ");
  switch (mpu.getScale())
  {
    case MPU6050_SCALE_2000DPS: Serial.println("2000 dps"); brea
    case MPU6050_SCALE_1000DPS: Serial.println("1000 dps"); brea
    case MPU6050_SCALE_500DPS: Serial.println("500 dps"); break;
    case MPU6050_SCALE_250DPS: Serial.println("250 dps"); break;
  }

  Serial.print(" * Gyroscope offsets: ");
  Serial.print(mpu.getGyroOffsetX());
  Serial.print(" / ");
  Serial.print(mpu.getGyroOffsetY());
  Serial.print(" / ");
  Serial.println(mpu.getGyroOffsetZ());
```

```
  Serial.println();
}

void loop()
{
  Vector rawAccel = mpu.readRawAccel();
  float ax = rawAccel.XAxis;
  float ay = rawAccel.YAxis;
  float az = rawAccel.ZAxis;

  float tiltX = atan2(ay, az) * 180 / PI; // Angle about X-axis
  float tiltY = atan2(ax, az) * 180 / PI; // Angle about Y-axis

  Serial.print("Tilt Angle (degrees) - X: ");
  Serial.print(tiltX);
  Serial.print(" | Y: ");
  Serial.print(tiltY);
  Serial.println();

  delay(10); // Delay for readability
}
```

Method used

The method I used in the code is to find the angular displacement by taking the arc tan of the angle using the accelerations in the respective axes and converting the radian angle obtained into degrees and printing the output.

the a tan2(x,y) function is used for this purpose so it gives the accurate angle with respect to quadrants.