**PROTOCOLO**

***Versión 1.0***

**Elaboro: María Fernanda Tarquino Triana**

# OBJETIVO

Mostrar el procedimiento a seguir para rotar un cubo creado en Unity mediante el uso de un sensor IMU.

# ALCANCE

Proporcionar los paso a paso necesarios a seguir para poder controlar la rotación de un cubo creado en Unity mediante el uso de un MPU6050 y una ESP32 conectada al computador.

# INSTALACIÓN DE SOFTWARE

1. PlatformIO IDE
   1. Descargar Visual Studio Code (VSCode) <https://code.visualstudio.com>
   2. Abrir VSCode e instalar la extensión PlatformIO IDE desde Extensions buscándola y darle a Install.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 1: Pestaña Extensions.

1. Unity
   1. Descargar Unity <https://unity.com/download>
   2. Crear cuenta de Unity <https://id.unity.com/en/conversations/279df4f8-cce6-4597-96fa-41d4018505fa019f>
2. **PLATFORMIO**
3. Entrar a PlatformIO desde la pestaña ubicada en la parte izquierda dando clic al ícono. Este se demorará un poco en cargar y aparecerá la pestaña Home.

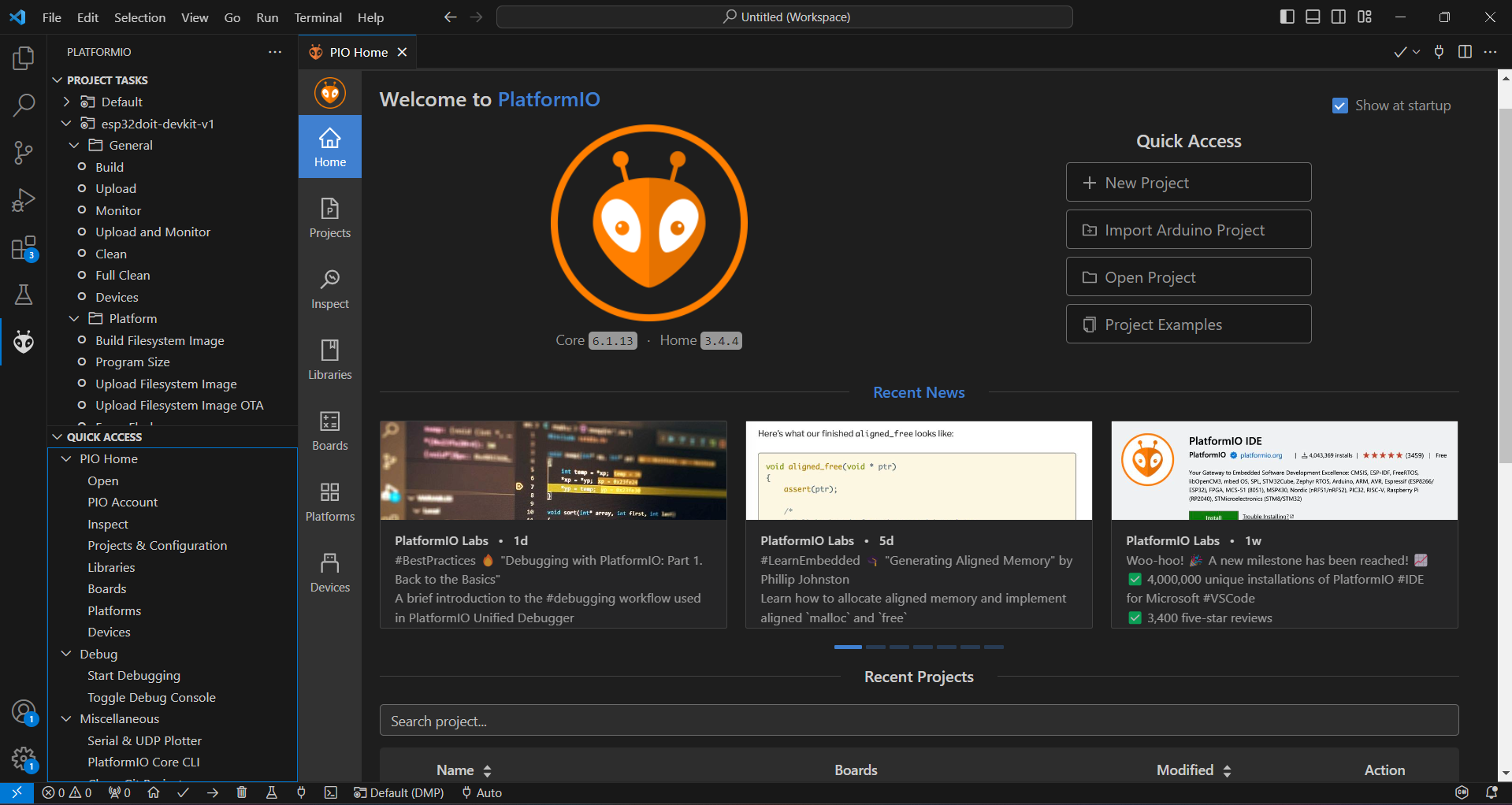


Figura 2: Página Home PlatformIO.

1. Para crear un proyecto se da click en New Project, aparecerá una ventana en la que se le pondrá el nombre al proyecto, se escogerá la board con la que se va a trabajar que en este caso la ESP32 aparece como DOIT ESP32 DEVKIT V1 y el Framework que será Arduino Framework. Una vez con todo esto se le da clic a Finish.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 3: Ventana creación del proyecto PlatformIO.

1. Una vez se termine de crear el proyecto se va a cargar la librería necesaria para poder trabajar con el MPU6050. Se va otra vez a PlatformIO, en Quick Access se le da a Libraries, se busca la librería “I2Cdevlib-MPU6050” y se da click en esta.

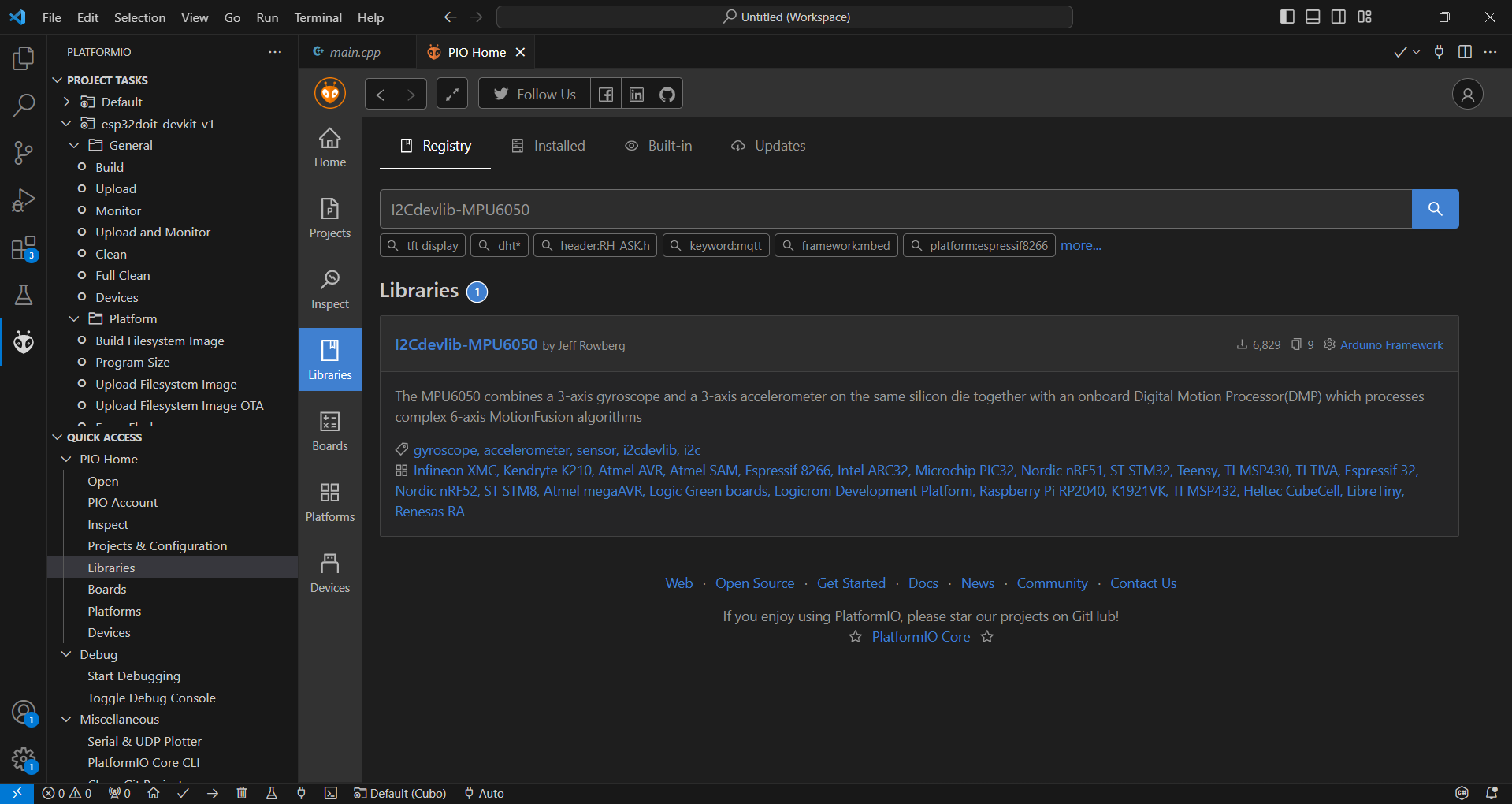


Figura 4: Página Libraries PlatformIO.

1. Para agregar la librería al proyecto se le da a Add to Project, aparecerá una ventana en la cual se escoge el proyecto al que se le va a agregar la librería (en este caso se trabaja en el proyecto con nombre Cubo) y por último se le da a Add.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 5: Página librería MPU6050.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 6: Ventana agregar librería al proyecto.

1. En la misma página de la librería del MPU6050 se busca en Examples el código que se llama “MPU6050\_DMP6\_using\_DMP\_V6.12 y se copia todo el código.

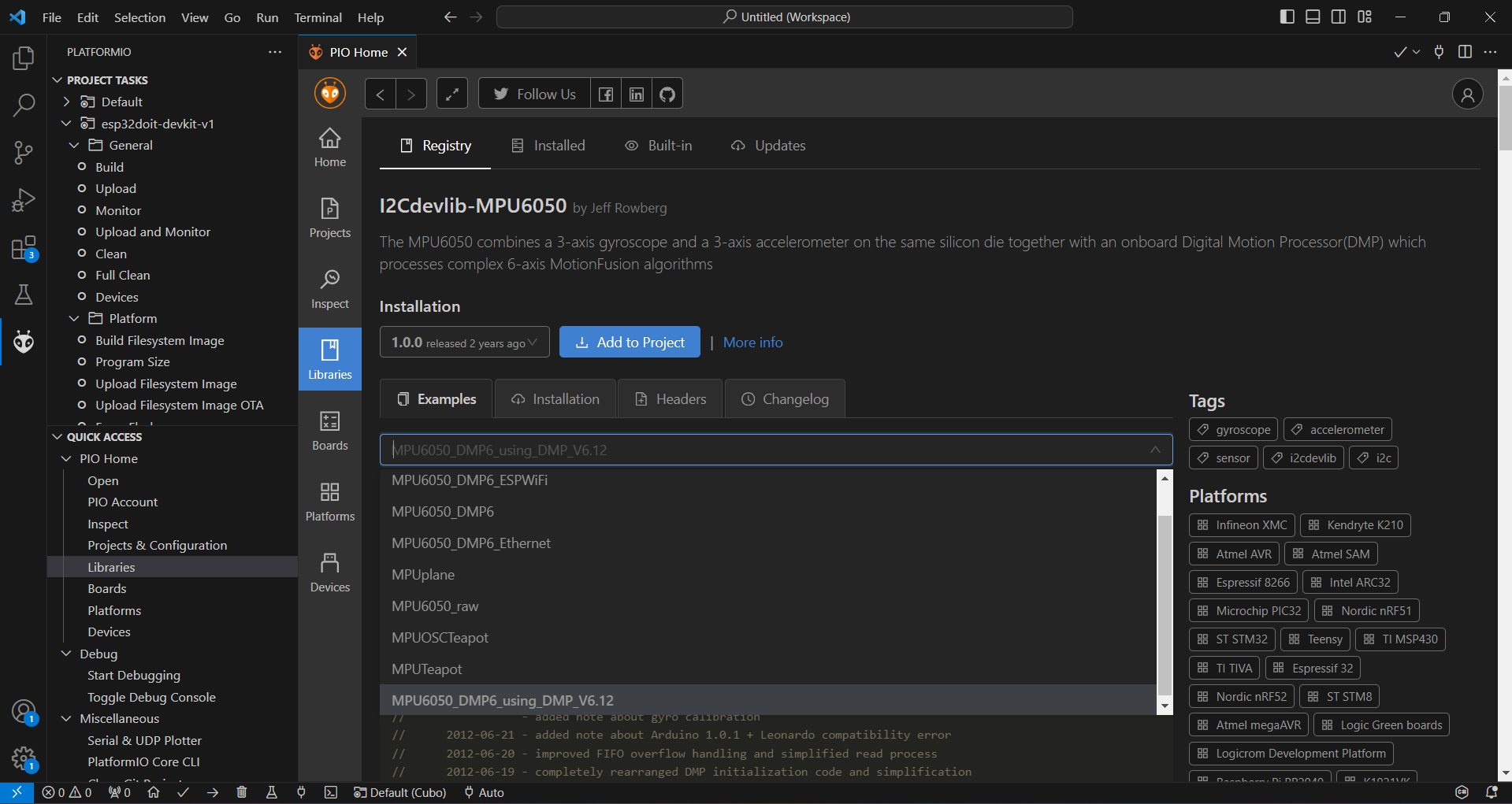


Figura 7: Ventana librería con los ejemplos.

1. Ahora en el panel de la izquierda de VSC se va a Explorer, en la carpeta del proyecto se busca el main.cpp que se encuentra desglosando src y se pega el código.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 8: Ventana main.cpp ya con el código de ejemplo.

1. Se realizan los siguientes cambios al código:
   1. Descomentar “#define OUTPUT\_READABLE\_QUATERNION” y comentar “#define OUTPUT\_READABLE\_YAWPITCHROLL”
   2. Cambiar el INTERRUPT\_PIN 2 a INTERRUPT\_PIN 4 y cambiar LED\_PIN 13 a LED\_PIN 2
   3. En el loop en la parte de OUTPUT\_READABLE\_QUATERNION cambiar los Serial.print() y se agrega un delay de 50 milisegundos para que se vea de la siguiente forma:

Texto

Descripción generada automáticamente

Figura 9: Parte del código en loop de la impresión de datos.

1. Dado que se tiene un Serial.begin(115200) es necesario modificar el platform.ini, dándole click y agregando al final la línea “monitor\_speed = 115200”.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 10: Página platform.ini del proyecto.

1. Se guardan todos los cambios, se conecta la ESP32 y para poder subir el código hay que ir a PlatformIO en la parte de Project Task y en la sección de General darle a Upload and Monitor (mientras se sube el código revisar el mensaje en la Terminal en el que se menciona a que COM se encuentra conectada la ESP32).

Texto

Descripción generada automáticamente con confianza media

Figura 11: Sección Project Tasks PlatformIO.

1. Cuando termine de cargar el código va a pedir que se mande algún carácter para seguir con el proceso. La primera fila de datos luego de que se inicialice el DMP serán los datos de offset con el cuál calibrar el MPU6050

Imagen que contiene Logotipo

Descripción generada automáticamente

Figura 12: Valores de los offset mostrados en Terminal.

1. Para calibrar el MPU de acuerdo con estos datos hay que tener en cuenta que están en el orden del acelerómetro de x,y,z y luego los giroscopios de x,y,z (esto se puede repetir las veces que se quieran para calibrar el sensor). Las funciones donde se modifican los offset se encuentran en el setup y una vez corregidos se verían así de acuerdo con los datos de antes:

Texto

Descripción generada automáticamente

Figura 13: Parte del código en setup de los offsets.

1. Una vez calibrado es necesario dejar lo más limpio posible la transmisión de datos por lo cual se va a comentar el código en la parte del setup tal que no sea necesario ingresar algún carácter para empezar a obtener datos y tengamos únicamente los quaterniones enviados por el sensor en la terminal. Por esto se van a comentar tanto los Serial.print() como los while desde la línea 188 hasta la línea 233 de la siguiente forma:

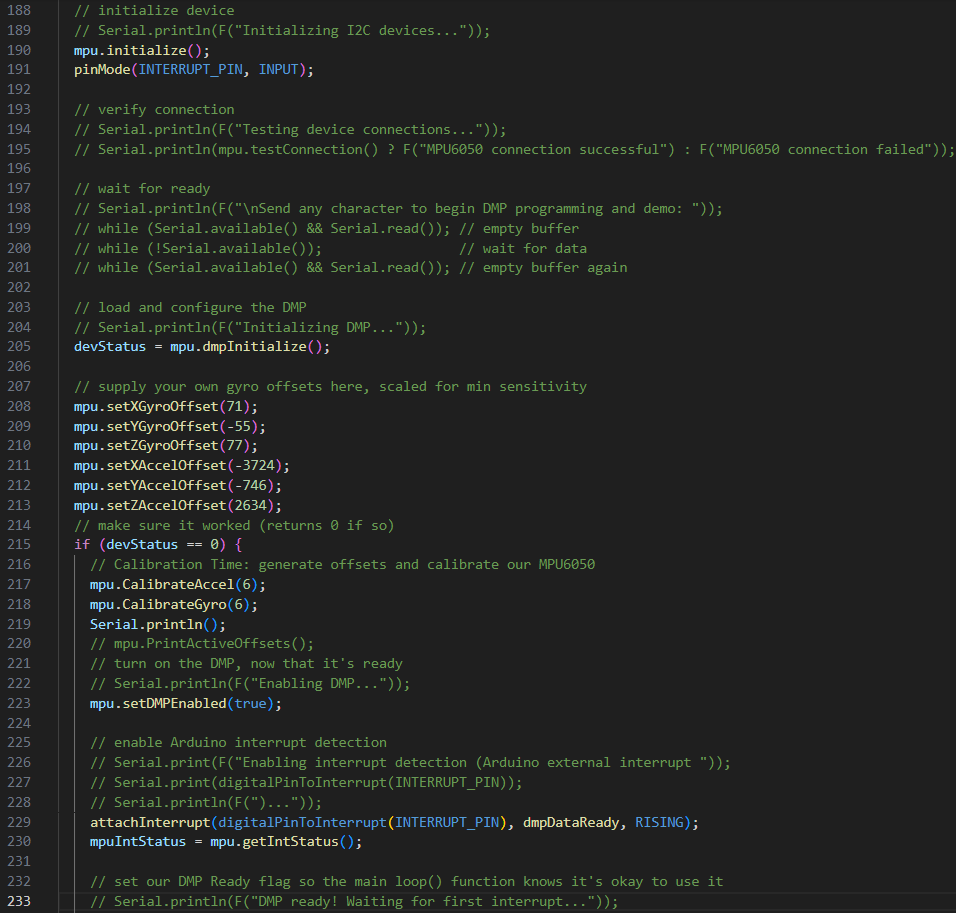


Figura 14: Modificación código documentando en el setup.

1. Se vuelve a cargar el código a la ESP32 y se verifica el funcionamiento del sensor.

# UNITY

1. Abrir Unity, iniciar sesión, ir a la sección Installs y en Install Editor seleccionar la versión 2022.3.19f1.

Captura de pantalla de un celular

Descripción generada automáticamente

Figura 15: Ventana Installs de Unity.

Captura de pantalla de un celular

Descripción generada automáticamente

Figura 16: Ventana instalación de versión del editor.

1. Una vez terminada la instalación de la versión ir a Projects y crear un nuevo proyecto en New Project.

Captura de pantalla de un celular

Descripción generada automáticamente

Figura 17: Ventana Projects Unity.

1. Seleccionar el template 3D, asignarle un nombre (en caso de que ser necesario se le puede cambiar la ruta de guardado) y darle a Create project.

Captura de pantalla de un celular

Descripción generada automáticamente

Figura 18: Ventana creación del proyecto en Unity.

1. Una vez abierto agregamos las ventanas de Game y Console desde la opción de Window>General.
2. Creamos un cubo ya sea desde la pestaña o desde el panel Hierarchy. Se puede modificar las características del cubo desde el panel Inspector (escala, rotación y posición en Game)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 19: Opción 1 creación del cubo.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 20: Opción 1 creación del cubo.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 21: Inspector del cubo generado.

1. Al final del inspector darle click a Add Component, aparecerá una ventanita en la cual se buscará New Script, se le asignará un nombre al script y se le dará clic a Create an Add. Este script aparecerá creado en la sección de Project.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 22: Ventana creación Script.

1. Desde Project abrimos el script que se encuentra en Assets con doble click

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Figura 23: Sección Project.

1. Esto nos abrirá una ventana de Visual Studio Code en el cuál podremos programar el poder rotar el cubo. El código para poder rotar el cubo es el siguiente:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO.Ports;

public class RotarCubo : MonoBehaviour

{

    // Start is called before the first frame update

    SerialPort serialPort=new SerialPort("COM5",115200); //Cambiar el número del COM de acuerdo al observado al cargar el código en la ESP32

    public string data; //Variable para leer los datos enviados de la ESP32

    float w,x,y,z; //Variabbles para almacenar los datos correspondientes de los quaterniones

    void Start()

    {

        // Se inicia la comunicación del puerto y se establece un tiempo de espera de envío de datos

        serialPort.Open();

        serialPort.ReadTimeout=60;

    }

    // Update is called once per frame

    void Update()

    {

        try{

            if(serialPort.IsOpen){

                //Leer por líneas los datos recibidos

                print(serialPort.ReadLine());

                data=serialPort.ReadLine();

                //Obtener un string con diferentes posiciones separando la data por ","

                string[] datos=data.Split(",");

                //Convertir los datos correspondientes a cada variable del quaternion en float

                w=float.Parse(datos[0]);

                x=float.Parse(datos[1]);

                y=float.Parse(datos[2]);

                z=float.Parse(datos[3]);

                //Generar la rotación del objeto creando un quaternion usando los datos obtenidos

                transform.rotation= new Quaternion(z,-x,-y,w);

            }

        } catch (System.Exception ex){

            ex=new System.Exception();

        }

    }

}

***Nota:***

* El formato de creación del Quaternion es (x,y,z,w) pero para este caso se cambió el orden de los ejes dado que la referencia con la cual se trabajó respecto al sensor y que los ejes del sensor no están orientados de la misma forma que el Unity se obtuvo ese formato (z,-x,-y).
* Para saber como ordenar los ejes de acuerdo con los datos que nos envía el MPU6050 se debe ir probando eje por eje hasta observar que el movimiento del cubo tenga sentido respecto al movimiento que uno está realizando. En el código para poder calibrar esto se dejará el eje a probar y el w con el valor recibido por el sensor mientras los otros 2 en 0 e ir cambiando hasta encontrar el orden adecuado. Ejemplo para probar el eje x serían las siguientes 3 pruebas “new Quaternion(x,0,0,w)”, “new Quaternion(0,x,0,w)”, “new Quaternion(0,0,x,w)”.
* En el caso de encontrar el orden correcto pero el sentido de giro es contrario a lo que se realiza solo se debe colocar un “-“ al eje que se encuentre mal.

1. Volver a Unity, conectar la ESP32 al mismo puerto donde se conectó cuando se subió el código y darle al botón de Play ubicado en el centro de la parte superior.

***Posible error:***

En caso de obtener el error “error CS0246: The type or namespace name 'SerialPort' could not be found (are you missing a using directive or an assembly reference?)” ir a Edit > Project Settings. Esto abrirá una Ventana en la cual buscaremos la sección Player y dentro de Player la configuración que dice “Api Compatibility Level” en donde se cambiará a la opción “.NET framework”.

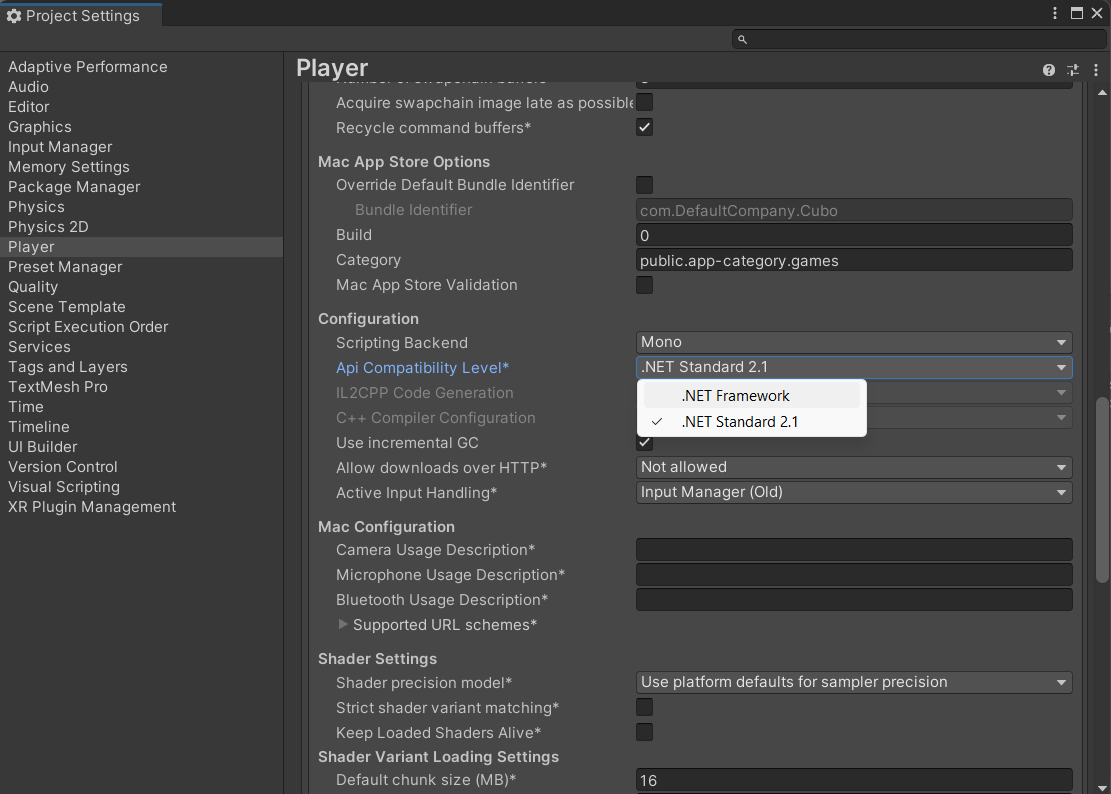


Figura 24: Ventana Project Settings.

# CONTROL DE CAMBIOS

|  |  |  |  |
| --- | --- | --- | --- |
| **DESCRIPCIÓN DEL CAMBIO** | **FECHA** | **VERSIÓN** | **APROBADO POR** |
|  |  |  |  |