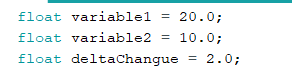
Proceso de transmisión y recepecion

1. Se crean las variables en Arduino



Texto, Carta

Descripción generada automáticamente

1. Verificamos si hay datos en el puerto, luego comprobamos si el dato corresponde a la tecla 1; si es así, enviamos todas las variables al buffer mediante un arreglo.Imagen que contiene Texto

   Descripción generada automáticamente
2. En Unity creamos variables locales para almacenar los valores traídos de Arduino

Texto

Descripción generada automáticamente

1. En void update verificamos si se presiono la tecla S, si es así, enviamos el valor 0x31 que representa al numero 1 que es lo que desencadena el envió de datos desde Arduino, creamos un arreglo de bytes de 28 posiciones llamado buffer donde almacenaremos todos los datos que recibimos EN ORDEN



Lo convertimos a Single para ver su representación flotante y posteriormente los convertimos a string para visualizarlos en texto, además vaciamos las variables de texto cada vez que se presiona S para asegurarnos que los datos no se acumulen

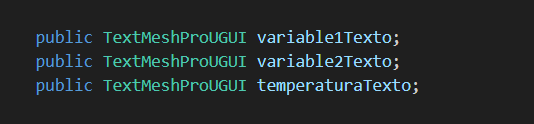
Manejo interfaz grafica

1. Cargamos las bibliotecas de TextMeshPro

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza media

1. Creamos variables de texto para todas las variables, en estas variables almacenaremos los valores recibidos convertidos a String



1. Texto

   Descripción generada automáticamente
2. En unity creamos objetos de texto y posteriormente los asignamos en el inspector a las variables de texto

Texto

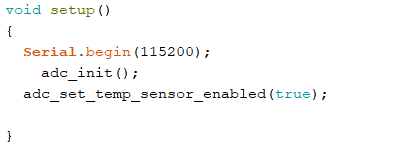
Descripción generada automáticamente

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Construcción y funcionamiento del micro

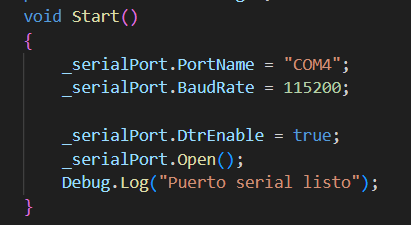
1. En Arduino establecemos la velocidad de ejecución, además establecemos la lectura de el pin4 encargado de la temperatura



1. En Unity creamos un objeto \_SerialPort de la clase serialPort mediante un método constructor



1. En void start establecemos los atributos de \_SerialPort, es importante que la velocidad establecida en Arduino sea la misma velocidad establecida en unity , además estableceremos otros valores, como el nombre del puerto, y si este está activado



1. Se usa la función OnDestroy para asegurarse que la comunicación entre Arduino y Unity se corte si el objeto al que esta adjutando el script es eliminado, evitando errores y congelamientos

Texto

Descripción generada automáticamente-