

 main ▾

shshjmakerspace

30 seconds ago 

..



arduino-to-unity

9 minutes ago



led-on-off

9 minutes ago



Demo.gif

9 minutes ago



README.md

30 seconds ago

› Aridity

› Setup

- It is better to install aridity from git (download and import it as a custom package from assets tab) and not from asset store.
- On my PC it has some problems with COM ports but on another PC worked pretty well.
- Main git repo is --- > [HERE](#)
- LED on/off Tutorial ---> [HERE](#)
- Guide to setup a project (You need to follow it before reading the rest of this document) - -- > [HERE](#)
- Note: when you want to upload .ino code to Arduino, your unity project should not be running.

After Following the setup guide (above link) we will compile and upload below code to the Arduino. In this code arduino is sening an integer counter (we can send string message as well) to unity. On the other side, in unity Message Listener (OnMessageArrived() method) receives the message and we store it in a public variable to be able to use it if we want.

Arduino code...

```
// arduino-to-unity/arduino-to-unity.ino
```

```

unsigned long last_time = 0;
int i;
void setup(){
    Serial.begin(9600);
    i = 0;
}

void loop(){

    if(last_time < millis() - 2000){
        Serial.println(i);
        last_time = millis();
        i++;
    }
}

```

Unity Ardity Message Listener...

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MyMessageListener : MonoBehaviour {

    // Use this for initialization

    public string MyMSG;
    void Start () {
    }

    // Update is called once per frame
    void Update () {

    }

    // Invoked when a line of data is received from the serial device.
    void OnMessageArrived(string msg)
    {
        Debug.Log("Arrived: " + msg);
        MyMSG = msg
    }

    // Invoked when a connect/disconnect event occurs. The parameter 'success'
    // will be 'true' upon connection, and 'false' upon disconnection or
    // failure to connect.
    void OnConnectionEvent(bool success)
    {
        Debug.Log(success ? "Device connected" : "Device disconnected");
    }
}

```

```
}
```

LED on-off Using keyboard (Digital input/output)

In this section we use demo scenes available in Ardity. First we read a char from keyboard (A or Z) and send it to Arduino. Then based on the value, we change the LED status (on or off). At the same time Arduino is sending the led status to unity every 2 seconds.

In Unity, select "Ardity/Scenes/DemoScene_UserPoll_ReadWrite.unity" and then after running the project Press 'A' or 'Z'. Now in unity console you will see a message that says "Sending A" or "Sending Z"

Note: Here, the most important script is "Ardity/Scripts/Samples/SampleUserPolling_ReadWrite.cs" that you can see in the below section. As you see, in the update() method we try to read a char from keyboard and send it over serial port.

```
using UnityEngine;
using System.Collections;

/**
 * Sample for reading using polling by yourself, and writing too.
 */
public class SampleUserPolling_ReadWrite : MonoBehaviour
{
    public SerialController serialController;

    // Initialization
    void Start()
    {
        serialController = GameObject.Find("SerialController").GetComponent<SerialCo

        Debug.Log("Press A or Z to execute some actions");
    }

    // Executed each frame
    void Update()
    {
        //-----
        // Send data
        //-----

        // If you press one of these keys send it to the serial device. A
        // sample serial device that accepts this input is given in the README.
        if (Input.GetKeyDown(KeyCode.A))
        {
```

```

        Debug.Log("Sending A");
        serialController.SendSerialMessage("A");
    }

    if (Input.GetKeyDown(KeyCode.Z))
    {
        Debug.Log("Sending Z");
        serialController.SendSerialMessage("Z");
    }

    //-----
    // Receive data
    //-----

    string message = serialController.ReadSerialMessage();

    if (message == null)
        return;

    // Check if the message is plain data or a connect/disconnect event.
    if (ReferenceEquals(message, SerialController.SERIAL_DEVICE_CONNECTED))
        Debug.Log("Connection established");
    else if (ReferenceEquals(message, SerialController.SERIAL_DEVICE_DISCONNECTED))
        Debug.Log("Connection attempt failed or disconnection detected");
    else
        Debug.Log("Message arrived: " + message);
    }
}

```

Below is the arduino code for this section. After reading from the serial port, if the value is 'A' the LED will turn on and if it is 'Z' the LED will turn off. Also, at the end of loop, we are sending back the LED state to Unity.

```

// led-on-off.led-on-off.ino

int ledPin = 13;
bool ledState = false;

void setup(){
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}

void loop(){
    if(Serial.available()){

```

```

char c = Serial.read();
if(c){

    if(c=='A'){
        digitalWrite(ledPin,HIGH);
        ledState = true;
    }
    else if(c=='Z'){
        digitalWrite(ledPin,LOW);
        ledState = false;
    }
}

}

if(ledState){
    Serial.println("led is ON");
}
else{
    Serial.println("led is OFF");
}
delay(1000);

}

```

Demo...

