

# Virandola

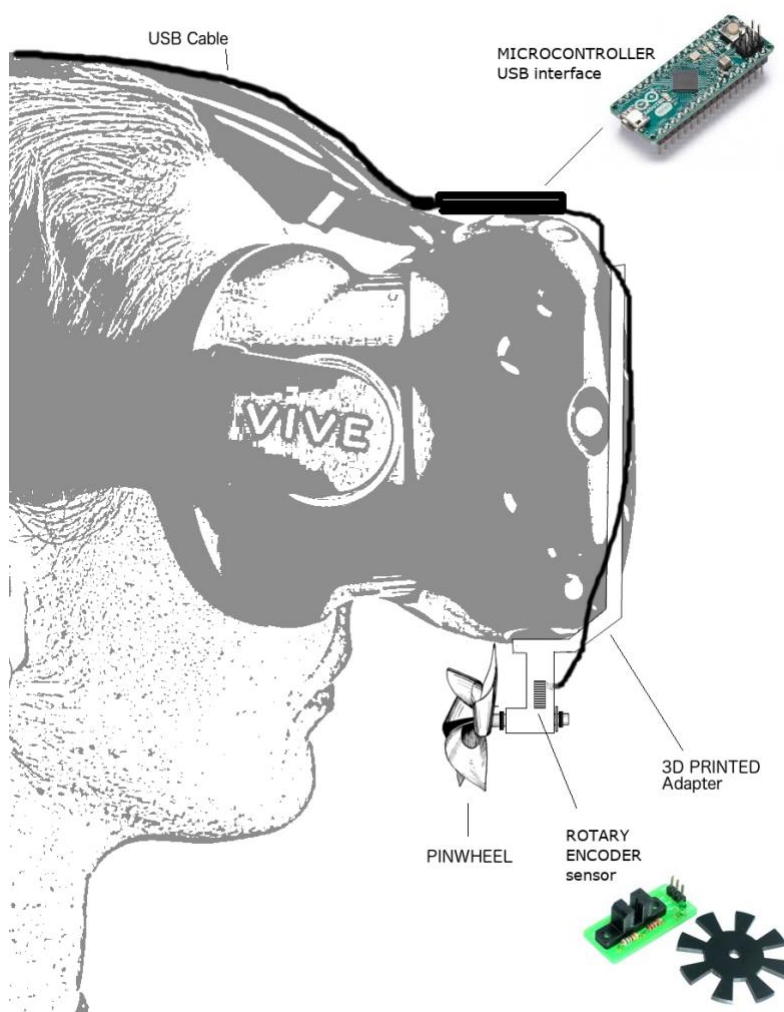
Virtual Reality Wind interface

By Bruno Herbelin

## Intro

The Virandola ('girandola' means windmill in Italian) is a little device mounted on a VR headset that captures the blast of air produced by the person when breathing out.

## Components



### 1) Pinwheel

Made with paper or taken from a toy, the pinwheel turns when the user blows on it.

### 2) Rotary encoder

The rotation of the pinwheel is converted into an electronic signal; with an optical encoder, there is no friction and the wheel can turn freely.

### 3) Microcontroller

A tiny electronic computer reads the signal of the encoder to calculate the rotation speed and communicate with USB to a computer.

### 4) Adapter for VR helmet

The pinwheel is placed in front of the mouth of a person wearing a head-mounted display (e.g. HTC Vive).

## Files provided

- **Virandola.pdf** : this manual
- **Teensy virandola.zip** : contains the virandola.ino.hex for the microcontroller
- **3d print virandola.zip** : contains the STL files for the 3D printer

# Electronics

## Cytron Simple Rotary Encoder Kit

Product Code RB-Cyt-39 by Cytron Technologies, <https://www.cytron.io/>

Cost ~ 10\$

E.g. Purchase on <https://www.robotshop.com/en/cytron-simple-rotary-encoder-kit.html>



## Teensy LC USB Microcontroller Development Board

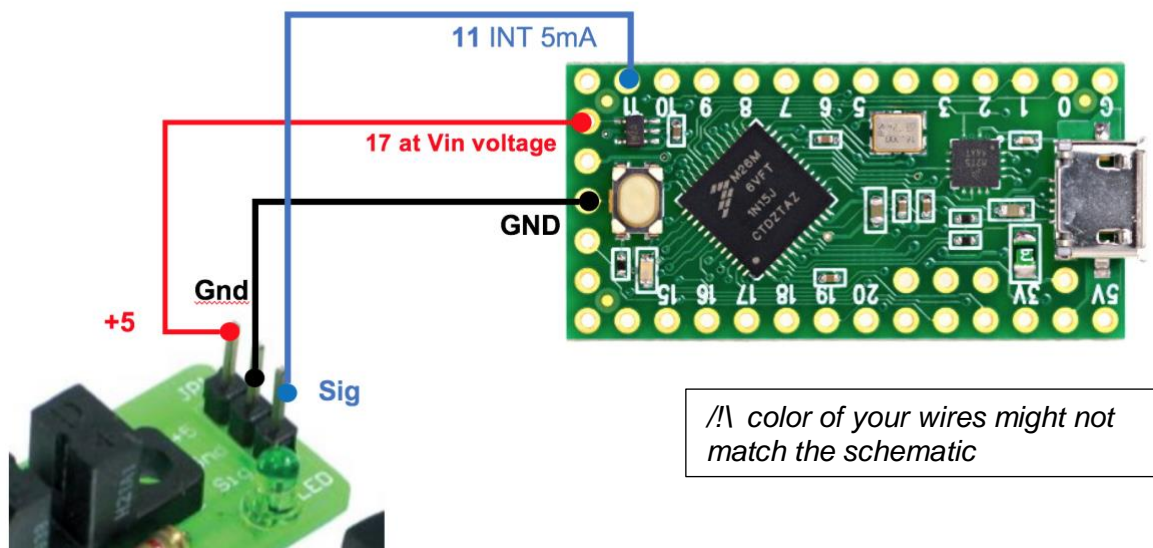
Teensy LC by PJRC <https://www.pjrc.com/>

Cost ~ 12\$

E.g. Purchase on <https://www.robotshop.com/en/teensy-lc-usb-microcontroller-development-board.html>



## Wiring



**In practice:** plug the wires provided with the Rotary Encoder and solder the other end to the Teensy LC board by respecting the schematic above.

## Microcontroller programming



Go to <https://www.pjrc.com/teensy/loader.html> and launch the Teensy loader (exists for Windows, MacOS and Linux).

Follow instructions by **opening the file 'virandola.ino.hex'** instead of 'blink\_fast.hex' given for the example (i.e. choose "Open HEX File" and open virandola.ino.hex). Press the button on the Teensy board when requested.

*Alternatively; see Annex 1 to program the Teensy yourself, or Annex 2 if you are using an Arduino Micro instead of a Teensy.*

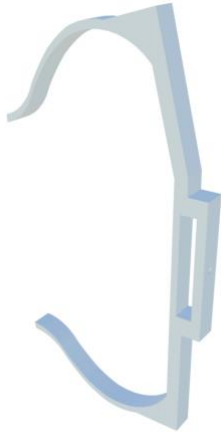
## Device assembly

### HMD mount

Using a 3D printer, create the different pieces of the device:

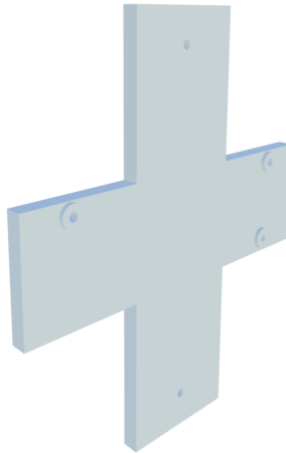
#### 2 X 'C' shape

courbe\_vive\_V2.stl



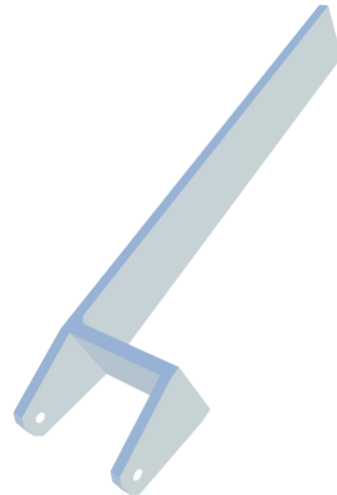
#### 1 X '+' shape

fixation\_webcam\_V2.stl



#### 1 X 'h' shape

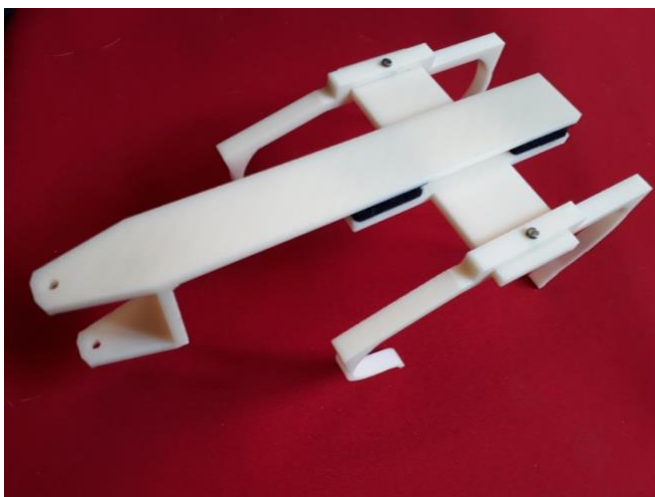
virandolaHMDfix.stl



The two 'C' shape pieces adjust to the shape of an HTC VIVE and are assembled together with the '+' shape.

Original design; Eric BOULAT;

<https://www.thingiverse.com/thing:1806594>

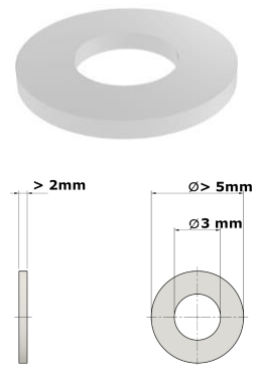


Fix the 'h' shape on top of the '+' shape using VELCRO (this way you can easily detach the Virandola from its HMD fixation).

## Pinwheel holder

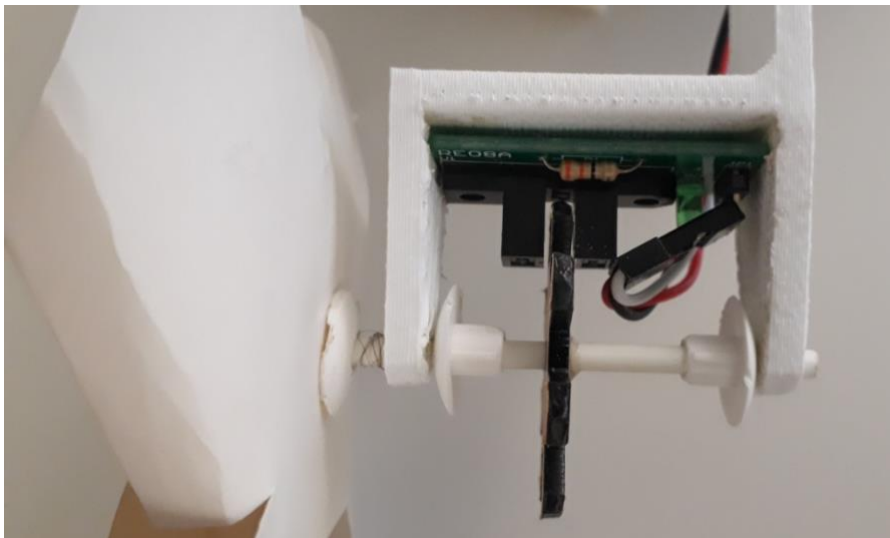
What you need from a DIY shop:

- 5 to 7cm long round plastic rod bar, diameter 3mm (any plastic or polymer material, e.g. ABS, Nylon or Acrylic)
- Thick plastic spacer washer (e.g. Nylon material) internal diameter 3mm, at least 2mm thick
- Super glue



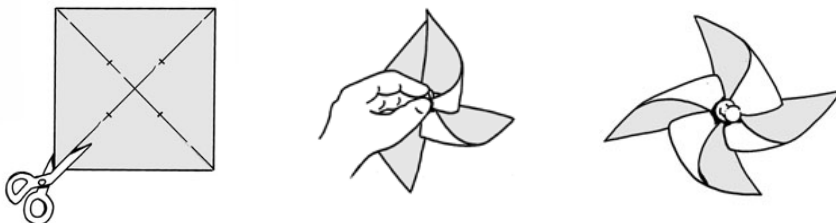
### Assembly

- 1) Place the rotary encoder board in the 3D printed 'h' shape
- 2) Insert a round plastic rod bar in the holes of the 'h' shape
- 3) Place the wheel of the rotary encoder and the washers on the rod bar
- 4) Align the wheel with the encoder board and glue the wheel and the washers on the rod; be careful to make sure the wheel can turn freely



## Pinwheel

Create a pinwheel from a piece of paper (instructions on <https://www.wikihow.com/Make-a-Pinwheel> or <http://www.leslietryon.com/3dcolorcutout/makepinw/makepinwheel.html> ) or find one on a kid's toy.

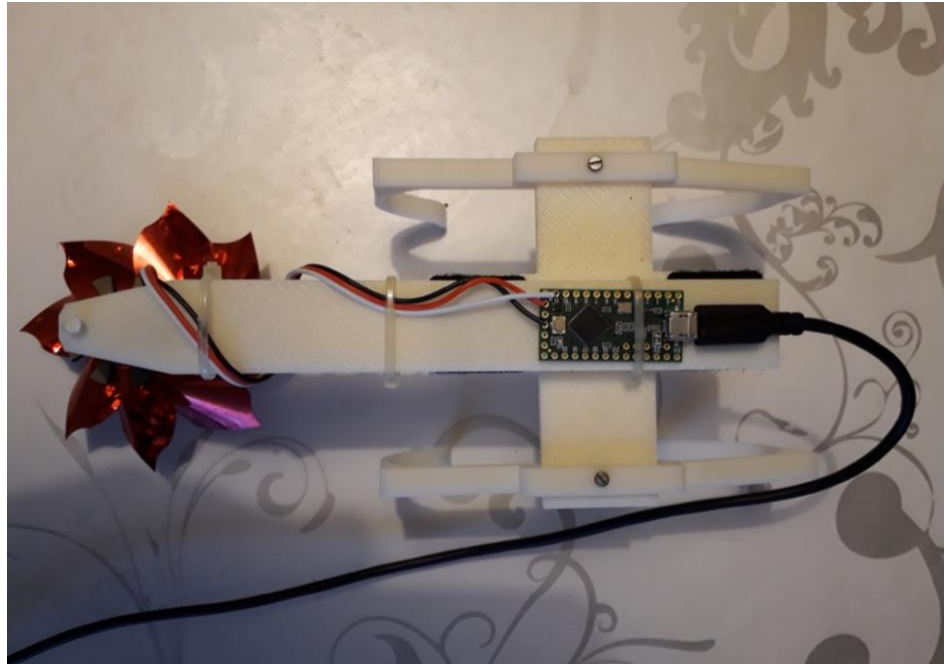
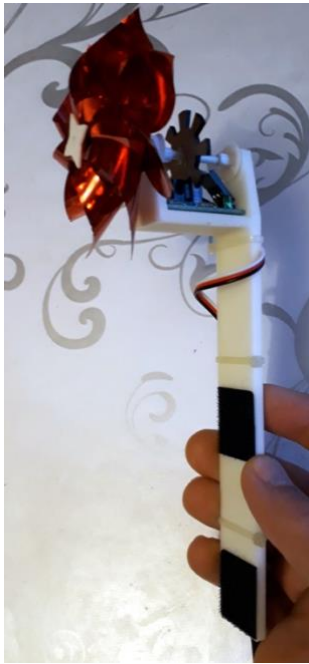


Glue the pinwheel on the rod and fix it with two washers.



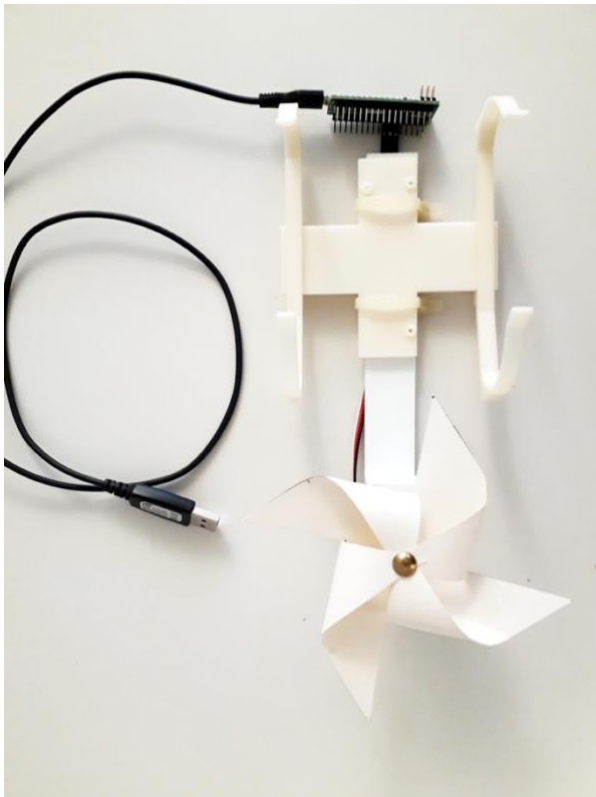
## Final device integration

Fix the Microcontroller board on the 'h' shape using



First prototype, with an Arduino Micro and a paper wheel.

Second prototype, with Teensy LC and a plastic wheel.



## Unity 3D integration

You need to create a custom `MonoBehavior` class to read the serial Input sent by the device though USB. You can read the data in the monobehavior's `Update()` function, and perform the initialization in the monobehavior's `Start()` function.

### Important steps

Import the Input / Output module

```
using System.IO.Ports;
```

Create a serial port object

```
SerialPort sp = new SerialPort("COM3", 9600);
```

Open the serial port

```
sp.Open();
```

Read data

```
sp.ReadLine();
```

### Example code

```
using UnityEngine;
using System.Collections;
using System.IO.Ports;

public class Virandola
    : MonoBehaviour {

    SerialPort sp = new SerialPort("COM3", 9600);

    void Start() {
        sp.Open();
        sp.ReadTimeout = 1;
    }

    void Update()
    {
        try{
            print(sp.ReadLine());
        }
        catch(System.Exception){
        }
    }
}
```

### Online tutorials

Sending Data from Arduino Uno to Unity

<https://answers.unity.com/questions/1069326/sending-data-from-arduino-uno-to-unity.html>

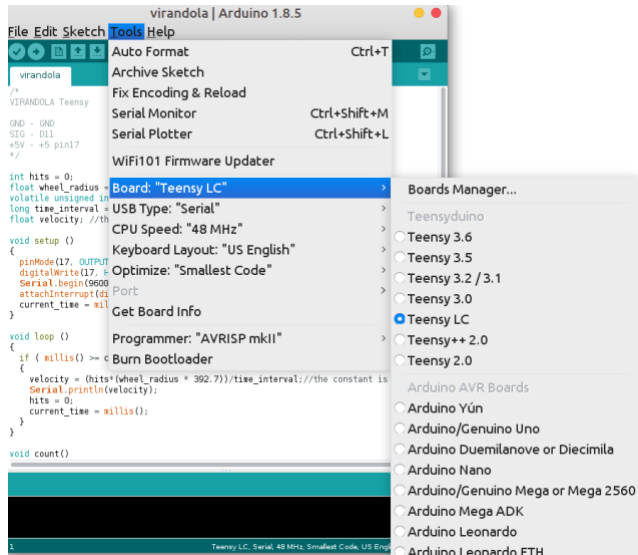
Transform Unity Object with Arduino

<https://answers.unity.com/questions/1561803/transform-unity-object-with-arduino.html>

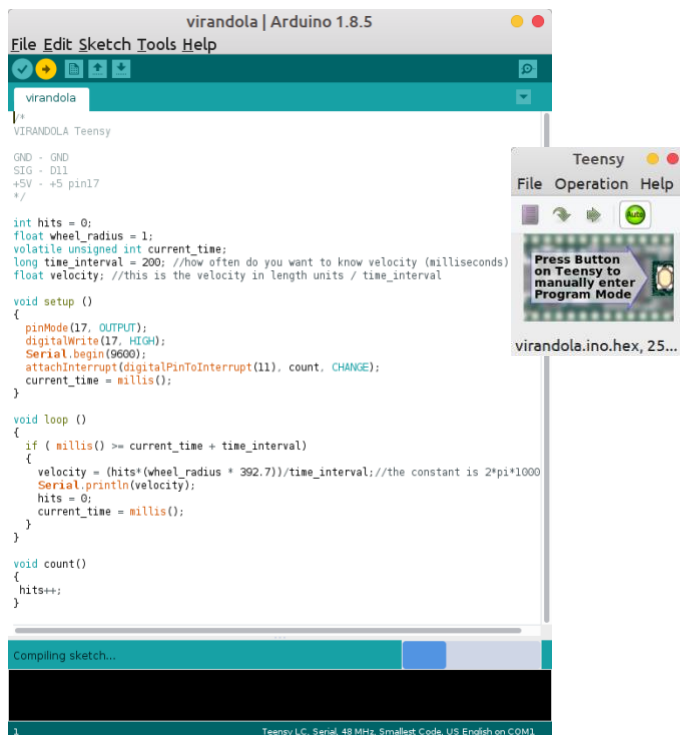
## Annex 1 – Programming the Teensy LC by yourself

Go to [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html) and download and install the **Teensyduino v1.45** modules for Arduino IDE following the instructions.

Run the Arduino IDE and select the 'Teensy LC' Board from the Tools menu



Enter source code (see below) and upload it to the board (arrow button). Follow instructions and press the button on the Teensy board when requested.



## Source code

```
/*
VIRANDOLA TEENSY
GND - GND
SIG - D11
+5V - +5 Pin17
*/

int hits = 0;
float wheel_radius = 1;
volatile unsigned int current_time;
//how often do you want to know velocity (milliseconds)
long time_interval = 200;
//this is the velocity in length units / time_interval
float velocity;

void setup ()
{
    pinMode(17, OUTPUT);
    digitalWrite(17, HIGH);

    Serial.begin(9600);
    attachInterrupt(digitalPinToInterrupt(11), count, CHANGE);
    current_time = millis();
}

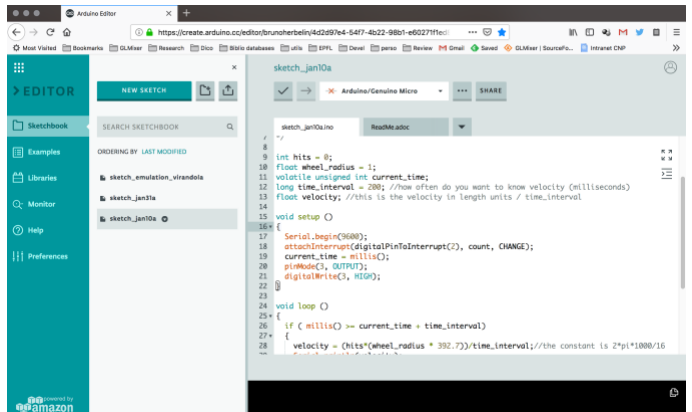
void loop ()
{
    if ( millis() >= current_time + time_interval)
    {
        // the constant is 2*pi*1000/16
        velocity = (hits*(wheel_radius * 392.7))/time_interval;
        Serial.println(velocity);
        hits = 0;
        current_time = millis();
    }
}

void count()
{
    hits++;
}
```



## Annex 2 – Using an Arduino Micro

- 1) Go to <https://create.arduino.cc/> and enter the Arduino Web Editor
- 2) Plug the Arduino Micro
- 3) create a new sketch
- 4) Select the Arduino/Genuino Micro board
- 5) Paste the code (see below) in the editor
- 6) Upload the code



### Source code

```
/*
 * VIRANDOLA
 *
 * GND - GND
 * SIG - D2
 * +5V - D3
 */

int hits = 0;
float wheel_radius = 1;
volatile unsigned int current_time;
//how often do you want to know velocity (milliseconds)
long time_interval = 200;
//this is the velocity in length units / time_interval
float velocity;

void setup ()
{
  Serial.begin(9600);
  attachInterrupt(digitalPinToInterrupt(2), count, CHANGE);
  current_time = millis();
  pinMode(3, OUTPUT);
  digitalWrite(3, HIGH);
}

void loop ()
{
  if ( millis() >= current_time + time_interval)
  {
    // the constant is 2*pi*1000/16
    velocity = (hits*(wheel_radius * 392.7))/time_interval;
    Serial.println(velocity);
    hits = 0;
    current_time = millis();
  }
}

void count()
{
  hits++;
}
```

## Annex 3 – Photo Album

Visit <https://photos.app.goo.gl/x870mh9C6JfmlUnf2> for more photos and videos.

### Virandola - Virtual Reality Swirl Interface

Jan 22–Feb 1

