# Physical & electronic implementation

## Preface

The human interface of this device is a glove, provided with an accelerometer to detect the hand rotation of the user and metal tips on the thumb and index fingers.

By pinching the two fingers together, the Arduino board enables the serial communication via USB. If the fingers are not touching, an open circuit is detected and every rotation or movement is ignored.

## Accelerometer

The accelerometer is mounted on a PCB from < NOME AZIENDA >. It uses the MMA7361 IC.

This kind of sensor is a capacitive micromachined accelerometer, which means that internally there are multiple membranes and their relative movement alters the devices’ capacity. The capacity change leads to a variation in the output voltage seen at the three X,Y and Z pins.

The voltage value can oscillate in a range between 0 and 3,3 volts.

The MMA7361 IC can detect accelerations of ±1.5Gs or ±6Gs, based on the mode it is set to by grounding a pin. For this experiment, it was set to ±1.5Gs sensibility.

Using this resolution, an acceleration of 1G leads to a voltage variation of 800mV. The voltage variation follows the acceleration direction, sign included.

## Arduino board

The Arduino board used is a MKR WiFi 1010. It has a 12-bit DAC, but the Arduino IDE uses 10-bit resolution by default, giving 1023 steps. This leads to a sensibility of 4,9mV for each increment.

When resting, the accelerometer detects 1G on the positive Z axis, leading to an output voltage of 2,45V. The ADC is supposed to read 501 / 1023.

The other two axis are not being subjected to any acceleration, leading to an ADC value of 338 / 1023.

## Algorithm

At startup, there is a calibration phase where the hypothetical values of 501 and 338 are overwritten by the real module values, to compensate for the user’s hand resting position and manufacturing variability of the accelerometer.

Every axis voltage is read continuously and mathematically low pass filtered by means of a moving average with a window size of 20 samples.

Three ‘deltas’ are computed to know how big is the rotation with respect to the resting position of the calibration phase.

If while pinching the index and the thumb fingers, a threshold is passed, be it a positive or negative variation, the program looks into a map to compute what value must be printed on the serial port.  
If the rotation affects more than one axis over the set threshold, a decision is made, based on which axis is subjected to the biggest variation. This avoids changing two parameters at once if the hand is slightly tilted into another axis.

Having the output values mapped to the deltas allows to linearize the behaviour of the device.

If the Z axis has the biggest value of the three, the glove is assumed to be in the resting position and no message is sent.