



VII CONGRESO ARGENTINO DE INGENIERÍA MECÁNICA

II CONGRESO ARGENTINO DE INGENIERÍA FERROVIARIA

15, 16 Y 17 DE SEPTIEMBRE DE 2021



FoDAMI

Universidad Tecnológica Nacional

VII Congreso Argentino de Ingeniería Mecánica y II Congreso Argentino de Ingeniería Ferroviaria : VII CAIM-II CAIFE / compilación de Nancy Quaranta ; Marta Caligaris ; Fernando Palmieri ; editado por Nancy Quaranta ; Marta Caligaris ; Fernando Palmieri. - 1a ed. - Ciudad Autónoma de Buenos Aires : Universidad Tecnológica Nacional ; San Nicolás de los Arroyos : Facultad Regional San Nicolás, 2022.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-950-42-0210-3

1. Ingeniería. I. Quaranta, Nancy, comp. II. Caligaris, Marta, comp. III. Palmieri, Fernando, comp. IV. Título.

CDD 620.007

ISBN 978-950-42-0210-3



AJEA (Actas de Jornadas y Eventos Académicos de UTN) no percibe fondos desde los congresos / jornadas, la publicación es gratuita y abierta para eventos realizados u organizados en unidades académicas de la UTN



Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

VII CONGRESO ARGENTINO DE INGENIERÍA MECÁNICA

II CONGRESO ARGENTINO DE INGENIERÍA FERROVIARIA

15, 16 Y 17 DE SEPTIEMBRE DE 2021

Facultad Regional San Nicolás (UTN)
Foro Docente del Área Mecánica de las Ingenierías (Fo.D.A.M.I.)



MOUSE PARA PERSONAS CON MOVILIDAD REDUCIDA

Msc.Ing. Eduardo Cirera ¹, Ing. Gerardo A. Bravo ², Ing. Martín Torres ³, Fernando Flores ⁴

1 Facultad de Ingeniería, Universidad Nacional del Nordeste,
Calle Las Heras N.º 727, Resistencia Chaco, Argentina,
ecirera@ing.unne.edu.ar, CP:3500

2 Facultad de Ingeniería, Universidad Nacional del Nordeste,
Calle Las Heras N.º 727, Resistencia Chaco, Argentina,
gbravo@ing.unne.edu.ar, CP:3500

3 Facultad de Ingeniería, Universidad Nacional del Nordeste,
Calle Las Heras N.º 727, Resistencia Chaco, Argentina,
martinalejandro885@gmail.com CP:3500

4 Facultad de Ingeniería, Universidad Nacional del Nordeste,
Calle Las Heras N.º 727, Resistencia Chaco, Argentina,
floresfernando422@gmail.com, CP:3409

RESUMEN

En este trabajo se presenta el diseño y construcción de un dispositivo auxiliar tipo mouse para personas con movilidad limitada o sin movilidad en miembros superiores. El trabajo fue realizado en base a un requerimiento específico, pero puede ser extrapolado a personas con problemáticas similares. Se ha desarrollado un dispositivo digital que permite el movimiento del puntero del Mouse de una PC o NOTEBOOK mediante el movimiento de la cabeza o parte del cuerpo con movilidad de la persona. La particularidad respecto de otros sistemas similares reside en la capacidad de hacer CLICK mediante soplos de intensidad configurables. Por lo cual, con la ayuda de un teclado emergente virtual, es posible reemplazar al teclado tradicional y también la utilización del Mouse. La electrónica fue desarrollada con herramientas Open Hardware y se tomaron de referencia fuentes de código Open Source. Actualmente, el prototipo está en uso, siendo probado y mejorado permanentemente y ha sido evaluado positivamente por personas con la problemática que se intenta atender. Como conclusión, en el marco de tareas de extensión, se ha logrado realizar un aporte social con gran potencial de crecimiento debido a la posibilidad de interacción con el entorno que es posible obtener.

Palabras Clave: headmouse; emulador de ratón; mouse auxiliar; accesibilidad.

1. INTRODUCCIÓN

El presente proyecto, mouse para personas con movilidad reducida, persigue el objetivo de facilitar el acceso a dispositivos de uso cotidiano, cualquier máquina que se valga de un mouse o ratón para su uso, a personas con movilidad reducida. El mismo reemplaza al mouse y teclado convencional en toda su funcionalidad y permite realizar movimientos del puntero del ratón con movimientos suaves al mover la cabeza. Por otro lado, los clics, izquierdo y derecho, con soplos de intensidad variable, ajustados por el usuario.

El dispositivo usa en su mayoría elementos de fácil adquisición y económicos, así como librerías de alto nivel en el código, buscando que el producto final sea de simple acceso y reproducción para cualquier persona que lo necesite. También cuenta con un software que permite ajustar la intensidad del soplido para el accionamiento de los clics, como así también deja al usuario deshabilitar estos últimos. Por otro lado, el dispositivo busca ser complemento de otras herramientas de accesibilidad que poseen los sistemas operativos, como el teclado virtual.

2.FUNCIONAMIENTO Y COMPONENTES

El funcionamiento del dispositivo puede entenderse como el reemplazo a un mouse convencional en el cual se realizan los movimientos del puntero con el movimiento del brazo y los clics con los dedos. Esto es reemplazado por movimientos suaves de la cabeza y soplos de intensidad ajustable para las acciones de clics derecho e izquierdo.

Para este objetivo se dispuso de una serie de componentes, hardware, firmware y software, descriptos a continuación.

2.1 Hardware

Como unidad de procesamiento del sistema propuesto se utiliza el microcontrolador Atmega32u4 de 8 bits [1], que presenta como característica principal soporte para la interfaz USB embebida en el mismo, por lo tanto, se usa la placa Arduino Micro, basada en este microcontrolador, de esta manera se pudo utilizar librerías de alto nivel y con gran soporte en la programación.

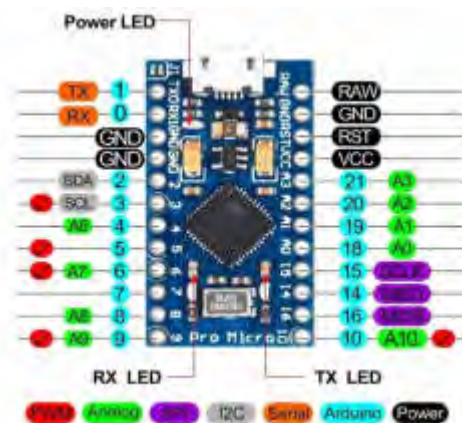


Figura 1: Pinout Arduino Micro

Para traducir los movimientos de la cabeza en movimientos del puntero se dispone de un giroscopio, de esta manera se logra cuantificar los movimientos en función de la velocidad angular de los mismos. Para esto se utiliza el módulo MPU6050 [2], que cuenta con giroscopio y acelerómetro incorporado, siendo su protocolo de comunicación el I2C.



Figura 2: MPU6050 y los ejes de rotación el giroscopio

Los clics son realizados por medio de una diferencia de presión estática generada con el soplido, para lograr la traducción de este proceso a magnitudes eléctricas se utiliza un sensor de presión cuya configuración es en puente de Wheatstone. A los valores del sensor se necesita amplificarlos, puesto que la señal brindada por el puente es diferencial. Para ello se ha utilizado el módulo HX711 [3] que posee un ADC ,Analog to Digital Converter, interno de 24 bits de resolución, que utiliza un protocolo serie no estandarizado.



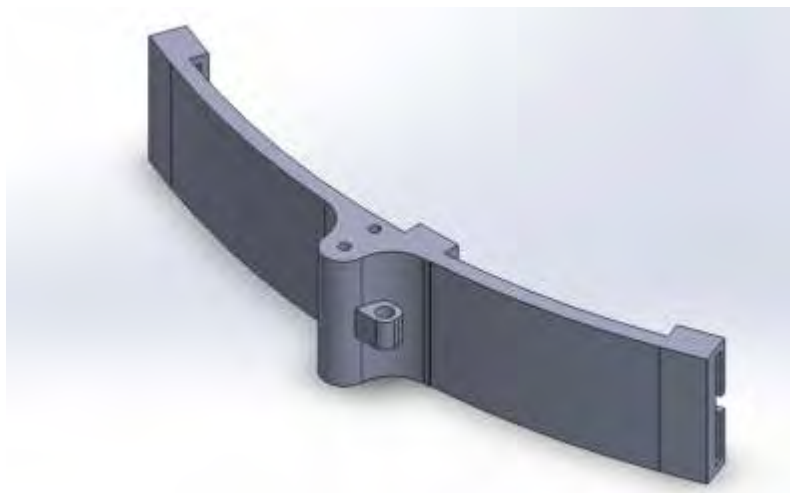
Figura 3: HX71



Figura 4: Vincha de minero en la que se montó la electrónica y boquilla.

La sujeción y contención de los componentes se realizó por medio de la adaptación de una vincha comercial, la cual puede brindar protección a la electrónica y comodidad al usuario.

Para direccionar el flujo de aire hacia el sensor, se diseñó un sistema que utiliza un soporte de fijación por el cual se vincula una manguera flexible desde el sensor de presión a la boquilla captadora. Dicha boquilla busca captar la mayor cantidad de presión de aire posible proveniente de un soplido, pero sin llegar a ser invasivo para el usuario, permitiendo, también, facilidades para poder intercambiarlo o sacarlo en pos de preservar una correcta higiene del mismo.



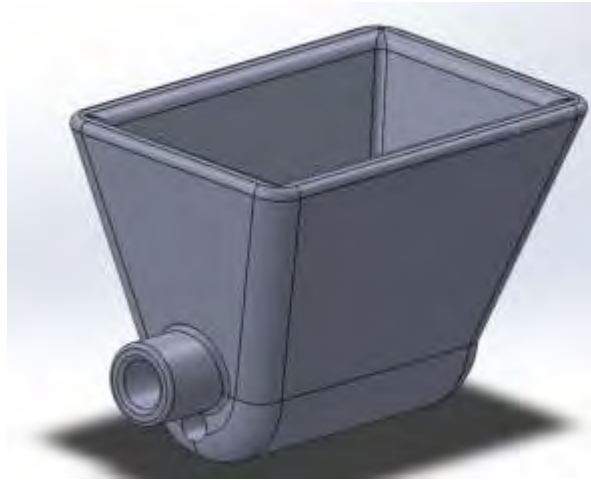


Figura 6: Boquilla captadora de aire para el accionamiento del clic

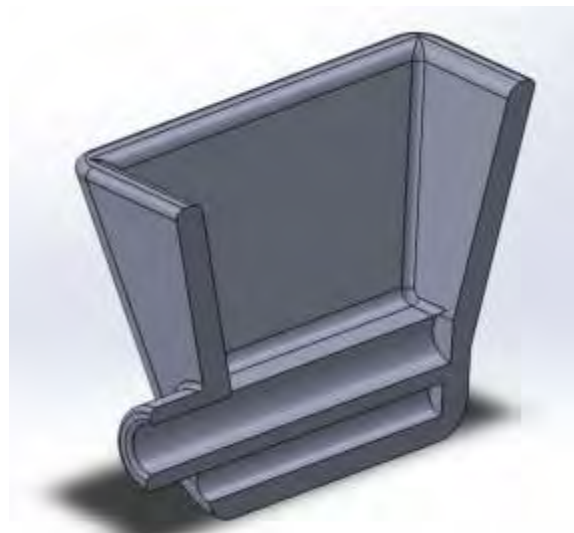


Figura 7: Boquilla captadora de aire en corte longitudinal, el cilindro de mayor diámetro es para la manguera de transmisión de presión.

2.2 Firmware y software

El dispositivo contiene un firmware de control, así como un software de configuración de valores, sobre los mismos se hablará en las siguientes subsecciones.

2.2.1 Firmware

Al ser el microcontrolador el Atmega32u4, este permite utilizar librerías de alto nivel que posee Arduino, incluyendo la librería Arduino en sí misma.

Para el manejo del MPU6050 se utilizó la librería MPU6050 de Jeff Rowberg [4], que a su vez necesita de las librerías I2Cdev [5], del mismo autor, y Wire de Arduino [6]. Con ellas se pueden obtener los valores del giroscopio del MPU6050, cada una cantidad de tiempo t , a los cuales se

les aplica una reducción de escala y corrección de offset. De esta manera se relacionan los movimientos de rotación sobre dos ejes del giroscopio con coordenadas x e y que serán las del puntero del ratón.

La gestión del clic se llevó a cabo por medio de un sensor de presión y el HX711, y de este último se obtienen los valores sin procesar por medio de la librería [Q2HX711 de Scott Russell](#) [7]. Esta librería permite obtener valores directos, sin procesar, del sensor de presión, entonces, en cada iteración del microcontrolador se realiza una comparación con el estado anterior de este sensor -usando de esta manera valores relativos- y es por ello que cuando el usuario genera una diferencia de presión, la misma se traduce en valores numéricos proporcional a la intensidad del soplido a nivel de código. Los mismos poseen variables umbrales, lo que permite poder controlar cuando se activa uno u otro y a su vez evitan que se solapen las funciones.

Por otra parte, los gestos soportados al día de hoy son el clic izquierdo para soplos leves, y para soplos de mayor intensidad el clic derecho, sobre los gestos elegidos y la opción de calibración se hablará más adelante en el apartado 3. PRUEBAS y REDISEÑO.

El acceso al ratón de la PC, notebook, netbook, etc, se realiza por medio de la librería Mouse de Arduino. La misma nos permite manipular el mouse en alto nivel realizando los movimientos y clics o gestos.

Por último, se implementó el uso de la librería EEPROM de Arduino [8] para guardar los valores de las variables umbral de accionamiento de los clics izquierdo y derecho, permitiendo reescribirlos cuando el usuario los modifica.

En resumen, el funcionamiento del firmware es el siguiente.

El mismo se inicia y pone en funcionamiento las comunicaciones necesarias, dejando un tiempo para la obtención y descarte de valores poco confiables de los sensores, y consulta a la EEPROM por los valores guardados. Luego, en el período de bucle continuo el microcontrolador consulta en cada iteración al giroscopio y al sensor de presión -por medio del amplificador- en búsqueda de cambios. Al encontrar un cambio en el giroscopio realiza la rutina de movimiento del mouse, y al encontrar un cambio en el sensor de presión realiza la rutina de los clics.

Por último, al firmware se lo presenta como una clase de C++ y su respectivo header para mayor orden, escalabilidad, y facilidades en las correcciones.

2.2.2 Software

Para la calibración de la intensidad umbral de los clics, y bloqueo del mouse, se proporciona un software creado en Java 8. El mismo posee una interfaz gráfica la cual se programó haciendo uso de la librería [Swing](#) de Java [9], la comunicación con el Arduino se logra mediante la librería

[PanamaHitek_Arduino](#) de Antony García González [10], y el bloqueo del ratón se logra con el uso de la clase [Robot](#) de Java [11].



Figura 8: Interfaz gráfica sin conexión - Interfaz gráfica con Arduino conectado

Teniendo el microcontrolador conectado se deberá iniciar el programa, el cual se conecta con el Arduino obteniendo del mismo los valores umbral de los clics izquierdo y derecho. Luego estos valores son mostrados en la interfaz gráfica y quedan a disposición del usuario para ser modificados. Cuando el usuario decide modificarlos sólo debe hacer clic sobre los botones en los costados y los valores cambiarán, actualizando los mismos en el Arduino.

Los valores varían, en el clic simple desde 5 hasta 30, y en el clic derecho desde 40 hasta 80.

El programa también permite la opción de bloquear completamente los clics, para ello el usuario deberá pasar el puntero del mouse por el bloque verde inferior y cuando este cambie a rojo los clics estarán inhabilitados, hasta que vuelva a pasar el puntero sobre el bloque rojo, cuando este cambie a verde los clics estarán habilitados. De esta manera, si el usuario decide utilizar una sensibilidad de clics tal que se active incluso cuando habla, puede bloquearlos y evitar que los mismos se activen.

3. PRUEBAS Y REDISEÑO

En esta etapa, con el prototipo diseñado y funcional, se contó con la colaboración de [Fernando Flores](#) quien se desempeña como artista plástico y es miembro de [Pintores con la Boca y el Pie](#). El se encuentra cuadripléjico, por lo que solo puede mover su cabeza, el diseño fue basado en su condición y fue el encargado de probar el dispositivo, es por eso que su ayuda fue de mucha importancia para sacar a la luz errores en los primeros prototipos, así como también fue de mucha importancia para la mejora del producto.

3.1 Primer prototipo

El primer prototipo entregado no poseía una interfaz gráfica de configuración de intensidad de los valores umbral de los clics, y estos a su vez eran, soplido leve clic simple o izquierdo, y soplido fuerte doble clic, es decir, no poseía la opción de clic derecho. La electrónica del mismo se encontraba en una pequeña caja plástica de la vincha, ubicada en la parte trasera de esta, de la cual se prolongaba el cable de conexión entre el microcontrolador y la computadora. Por lo tanto, el dispositivo estaba pensado para usarlo sentado.

3.1.1 Resultados del primer prototipo

De este primer prototipo, gracias al uso que le dio Fernando, se aprendió lo suficiente como para poder realizar un segundo prototipo mejorado.

En principio, el primer error se presentó en el diseño, ya que se lo pensó para usarlo en posición vertical, pero el usuario -Fernando- lo utilizaba, por una cuestión lógica, acostado, por esto la ficha micro usb del Arduino se desoldó quedando el aparato inutilizado. Por otro lado, Fernando hizo hincapié en la extrema sensibilidad del clic, el cual se accionaba incluso mientras hablaba, así como también nos manifestó la necesidad de poseer el acceso a la ventana de opciones que en el mouse convencional se presenta apretando el clic derecho.

A pesar de todo lo anterior Fernando manifestó que el aparato le resultaba muy cómodo, que había aprendido a utilizarlo con mucha facilidad y que estaba muy entusiasmado con la idea de seguir usándolo.

3.2 Segundo prototipo

En este prototipo se ubicó la electrónica en la parte frontal y se le brindó al usuario la primera versión del software, que sólo poseía la opción para bloquear el mouse. También se quitó la opción del doble clic con el soplido fuerte y se la reemplazó por el clic derecho.

3.2.1 Resultados del segundo prototipo

Esta versión mostró más resistencia que la anterior, pero luego de un tiempo de uso mostró la misma falla que el primer prototipo, es decir, se desoldó la ficha micro usb del Arduino. Por lo demás el usuario concluía en que estaba muy sensible el clic del mouse, y el problema no solo se manifestaba cuando hablaba sino en el uso cotidiano del mismo, ya que el clic se accionaba impidiendo el uso fluido del ratón.

3.3 Tercer prototipo

En este prototipo se tuvo en cuenta todo lo aprendido anteriormente, por lo que se buscó proteger la ficha micro usb del Arduino ubicándola de una manera que quede menos expuesta, así como también que sea menos sensible a posibles cargas por tensiones mecánicas producidas por tirones, movimientos bruscos, etc. Por otro lado, se complementó el dispositivo con un software

sobre el cual ya se comentó su función. Se agregó el uso de la EEPROM, y la comunicación entre Arduino y software.

3.3.1 Resultados del tercer prototipo

Este prototipo tuvo plena aceptación por parte de Fernando, el cual entendió fácilmente el uso del software y pudo calibrar la sensibilidad a su gusto, en los primeros prototipos estaba en 5 la sensibilidad, él según su conveniencia la dejó en 15, por otro lado, la nueva ubicación del microcontrolador contribuyó a la robustez del aparato ya que no se presentó la falla anterior.

Se destaca que al día de la fecha esta es la versión vigente y en uso.



Figura 9: Imagen lateral.



Figura 10: Imagen frontal.

4. CONCLUSIONES

Como conclusiones generales, se dice que el dispositivo posee tres virtudes fundamentales de todo desarrollo, es económico, sencillo, y útil. El mismo ayudó al usuario, en gran medida, en su uso cotidiano de la computadora, facilitándole dicha tarea, por lo que se cree que aporta un valor agregado real al desarrollo tecnológico en dispositivos para mejorar la accesibilidad.

Por otro lado se concluye, en base a la experiencia, que el desarrollo, al menos en aparatos relativos a la temática de accesibilidad, debe estar estrechamente guiado por la persona que sufre la problemática, es decir, es necesario y fundamental el trabajo en conjunto con la misma, ya que esta aporta una visión y un panorama que no posee quién desarrolla los aspectos técnicos del proyecto, y es muy sencillo cometer errores que se podrían evitar trabajando en colaboración con quién será el usuario final del dispositivo.

4.1 Mejoras a futuro

Como mejoras a futuro, queda en evidencia, que la primera en plantearse será la de lograr una versión inalámbrica del dispositivo, ya que la mayoría de los inconvenientes fueron por la necesidad de uso de un cable de conexión entre la computadora y el microcontrolador. Quedan por definir, el protocolo de transferencia de datos, y la alimentación del dispositivo.

Por otro lado, se analiza la opción de replantear el diseño sobre el que se monta la electrónica, buscando algo aún más robusta y adaptado a las necesidades del usuario.

Con respecto al software puede agregarse una opción para el control de la sensibilidad en el movimiento del mouse, y en el firmware la opción de corrección automática del offset del giroscopio.

En cuanto al hardware es necesario plantear el uso de otro sensor de presión de mayor disponibilidad en el mercado.

5. REFERENCIAS

- [1] Atmel®, ATmega16U4/ATmega32U4, 8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller, Atmel-7766J-USB-ATmega16U4/32U4-Datasheet_04/2016, 2016.
- [2] InvenSense®, MPU-6000 and MPU-6050 Product Specification Revision 3.4, datasheet. PS-MPU-6000A-00. Revision: 3.4, 2013.
- [3] AVIA SEMICONDUCTOR, 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales.

- [4] Jeff Rowberg, MPU6050, <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/MPU6050>, I2Cdev device library code is placed under the MIT license. Copyright (c) 2012 Jeff Rowberg.
- [5] Jeff Rowberg, I2Cdev, <https://github.com/jrowberg/i2cdevlib/tree/master/Arduino/I2Cdev>, I2Cdev device library code is placed under the MIT license. Copyright (c) 2013 Jeff Rowberg. Copyright (c) 2014 Marton Sebok. Copyright (c) 2015 Grégoire Surrel.
- [6] Arduino®, Wire, <https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/Wire>.
- [7] Scott Russell, <https://github.com/queuetue/Q2-HX711-Arduino-Library>.
- [8] Arduino®, <https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/EEPROM>.
- [9] Oracle, <https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html>.
- [10] Antony García González, Librería PanamaHitek_Arduino. Versión 3.2.0, https://github.com/PanamaHitek/PanamaHitek_Arduino.
- [11] Oracle, <https://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html>.