

Assistive Cursor Control Using Accelerometer

Sumanth M S¹, Amoolya R Bayari², Dr. Abhay Deshpande³

B.E^{1,2}, Assistant Professor³

Department of Electronics and Communication Engineering

R. V College of Engineering, Bangalore, Karnataka, India

Corresponding Author's email: amoolyabayari@gmail.com

Abstract

The Internet undergoes a fundamental transformation as billions of connected “things” surround us and embed themselves into the fabric of our everyday lives. However, this is only the beginning of true convergence between the realm of humans and that of machines, which materializes with the advent of connected machines worn by humans, or wearables. The resulting shift from the Internet of Things to the Internet of Wearable Things (IoWT) brings along a truly personalized user experience by capitalizing on the rich contextual information, which wearables produce more than any other today’s technology. In this paper, after reviewing the relevant contemporary background, we propose efficient means for the delegation of use applicable to a wearable device. Our efficient solutions facilitate contexts where one can use this device to control the cursor of a computer device without the presence of a physical mouse.

Keywords— *Internet of Wearable Things, wearable, delegation of use, assistive cursor control, cursor, personally identifiable information, data processing*

INTRODUCTION

1.1 Aim of the Project

To design a device which is able to sense hand gestures and convert them into a cursor motion on the screen of our computer and sense the joystick movement and convert them into the arrow key functionality.

1.2 Literature Review

In this project, a 3-axis accelerometer and a 3-axis Gyroscope (6 Degrees of Freedom) is used to control cursor movement. The accelerometer reads the tilt of each axis and outputs each as an analog voltage. The three ports of MCU are connected to the tilt outputs of the accelerometer respectively.

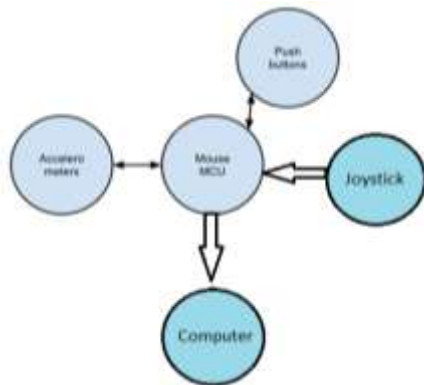


Fig: 1 General Block Diagram

Then, each of the voltage value of tilt is converted to digital numbers. These numbers are converted to final mouse movement values by noise detection and scaling. In order to provide accurate mouse-

motion and a user-friendly interface, several different types of post-processing are applied to the converted ADC values.

2. OBJECTIVE

2.1 Advantage

We have designed and built a computer pointing device with accelerometer based movement control. Implementation of this allows the user to wear a set of hardware and control a cursor through different hand orientations. Users can operate their computers with their hands in midair without the hassle of desks surfaces.

3. METHODOLOGY

Accelerometer sends the analog signals to the mouse MCU indicating the coordinates. The mouse MCU transmits the signal which is received by the receiver at the other end.

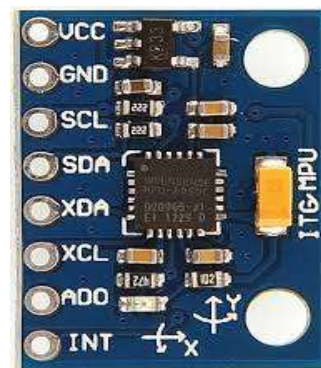


Fig: 2 Modern Device 3-Axis Accelerometer MPU 6050

4. LOGICAL STRUCTURE

At a high level, the design consists of two main parts: a glove and a base station. Operation of this device begins with the glove. A user wearing the glove can use hand tilt orientation and finger presses to operate the glove. The glove senses these user actions via two types of sensors: accelerometers and finger contact pads. After the glove's microcontroller processes the input data, it forwards a message a transceiver mounted on the glove unit. The transceiver then transmits this message wireless to a transceiver on the base station. The receiving transceiver forwards the message base station microcontroller. Finally, the microcontroller converts the message into a computer HID user friendly format and moves the computer cursor appropriately.

4.1 Hardware/Software tradeoffs

To interface this project with computers, it would be required to somehow implement a USB HID class device. During the planning stages of the design, we researched different ways implementing a USB HID and discovered that we could either proceed by doing this through software or hardware support. The MCU will read the signal and the gesture is interfaced with the computer through a USB cable.

Also, There is an additional Joystick to act as a keyboard which the functionality of the arrow keys. The joystick has two pins specific for the horizontal and the vertical movement control or the arrow keys.

The V_x and the V_y are the horizontal and vertical pins respectively. Hence this joystick is also interfaced with the computer with the help of Microcontroller Unit. The MCU reads the analog value from the joystick and converts it into digital values and hence the movement.

Thus came to the conclusion that the design would require an microcontroller in the base station in order to receive wireless messages and process the information. With this in mind, we decided it would be a more practical and simple to purchase a single hardware board that could perform both these functions. Almost too fittingly, we found a board that could perform the functions we needed and provided relevant example libraries, all within the budget.

4.2 Hardware Design and Implementation

As mentioned above, the physical design of the project includes two main parts: a glove unit and a base station. The glove unit's main purpose is to process information from sensors and transmit the user input to the base station. The base station's purpose is to receive the glove's sent data and forward it to the user's computer in a HID class friendly format. Each unit has its own unique microcontroller and respective peripherals, which are detailed more below.

5. PROCEDURE

Connect the 5V and GND of ATmega 32u4 to breadboard. Connect the GND and 5V of joystick in the same line as that of Leonardo as shown in the schematics. Connect the VRX to A1 (Analog Pin 1) and VRY to A0 of Leonardo. Connect the GND and VCC of MPU6050 in the railing as in schematics. Connect the SCL and SDA to SCL and SDA of ATmega32u4.

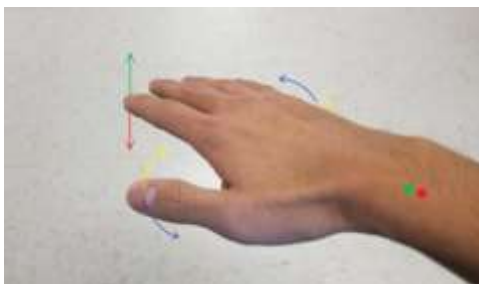


Fig: 3

Hand positioning for flat mode Blue rotation represents left movement, yellow is right,

green is up, and red is down (the dot represents pivoting above that point).



Fig: 4

Hand positioning for handshake mode Blue rotation represents left movement, yellow is right, green is up, and red is down (the dot represents pivoting above that point).

Although any starting tilt position is allowed (theoretically), we wanted to consider a special case: handshake position as shown in the picture below. The hand position is more ergonomic and allows the user easier hand rotation in both directions. In this case, instead of x and y axes, the y and z axes are read out of the accelerometer. This allows the user an up-down motion to change the y cursor position, which is consistent with that of flat position.

6. Axis Inversion

As one of the user operation options, our glove mouse allows the users to invert the

axes of mouse movement. When inverting DIP switches for both x and y axes are off, the cursor moves in normal direction. For example, left tilt will move the cursor to the left, and so on. If x-axis inverting is switched on, left tilting cursor movement towards the right

This function was implemented by setting a flag based on the switch status. If switch is off, the flag is set to be 1; if switch is on, the flag is set to be -1. The flag is then multiplied to the ADC sample. This option was added to give more freedom to users who have different directional preferences.

Gyro and accelerometer simulation



Fig: 5

Our final prototype is able to effectively control mouse movement with user hand gestures. The sensitivity of the motion is sensed by the device and hence the cursor

motion. Our final device is able to effectively carry out the arrow key functionality through a Joystick and a programmed microcontroller unit.

6.1 Initial Axes Calibration

When the move or scroll is enabled, the *calibrate ()* function is called to read the current tilt values. This function averages 1024 samples to get reliable offset values, which are used as the starting reference point. By subtracting those offsets from every ADC sample, the user can start at any position and use that hand orientation as the current reference axes. Once calibration is finished, the ADC samples are read in the ISR.

The corresponding offset value is subtracted from each sample. Then, each sample is zeroed if its absolute value is smaller than the threshold. We chose the threshold experimentally such that natural user hand quivering is treated as a noise instead of valid mouse movement. Because of the fact calibration should be executed before the program starts to read the ADC samples, the *cli()* and *sei()* (disable and enable the interrupt) functions were used to make this function atomic.

7. Future Work

Higher resolution cursor movement could be attractive for commercialization. Wireless communication could be established between the Microcontroller and the Sensor to making it completely wireless. Additional buttons can be added to carry out the functionality of left and right click and also scrolling.

RESULT

The final prototype is able to effectively control mouse movement with user hand gestures. The sensitivity of the motion is sensed by the device and hence the cursor motion. The final device is able to effectively carry out the arrow key functionality through a Joystick and a programmed microcontroller unit.

CONCLUSION

Primarily, we believe that a smaller wearable unit would be more practical for everyday use. The project is built with a small budget in mind and with easily usable, but not necessarily the smallest, parts. Additionally, higher resolution cursor movement could be attractive for commercialization. Wireless communication could be established between the Microcontroller and the Sensor to making it

completely wireless. Additional buttons can be added to carry out the functionality of left and right click and also scrolling.

REFERENCES

1. Abhinav Sharma, Mukesh Agarwal, Anima Sharma and Sachin Gupta. 20013. "Sixth sense Technology," in Proc. International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 1, pp. 277-282, April.
2. Saikat Basak and Arundhuti Chowdhury. 2014. "A Vision Interface Framework for Intuitive Gesture Recognition using Color based Blob Detection," in Proc. International Journal of Computer Applications, Vol. 90, No. 15, March.
3. Shiguo Lian, Wei Hu and Kai Wang. 2014. "Automatic User State Recognition for Hand Gesture Based Low-Cost Television Control System," IEEE Transactions on Consumer Electronics, Vol. 60, No. 1, pp. 107-115, February.
4. Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh and

- Mahmoud Alhelou. 2011. "Hand Gesture Recognition System," in Computer Engg Department, The University of Gaza.
5. G. Senthilkumar, K. Gopalakrishnan and V. Sathish Kumar. 2014. "Embedded Image Capturing System Using Raspberry Pi System," in Proc. International Journal of Emerging Trends & Technology in Computer Science. Vol. 3, pp. 213-215, April.