GLOUSE - A Wearable Mouse Device

1st Rômulo Freire Férrer Filho
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
romulofffufc@gmail.com

2nd Marcus Vinícius Lucas Machado de Andrade
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
marcusviniciuslma@gmail.com

3rd Pablo Roberto Grisi
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
pablogrisi@gmail.com

4th Ricardo Jardel Nunes da Silveira
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Brasil
jardel@ufc.br

***Abstract*—Man machine interfaces are very important for breaking barriers between humans and computers. Mouse used to be the main way of interacting with the computer. Even today the mouse is one of the main way, but other ways have emerged, like touchscreen. The touchscreen revolution was definitive to include the elderly and children in the digital era. In this sense, a low-cost device called glouse is proposed, which is a fusion of glove and mouse, with the intention of further facilitating man's interaction with the computer, further breaking down existing barriers between man and machine. The implementation of glouse, based in the m-Cortex family processor-based device as well as accelerometers available in the market is shown in detail, as well as results obtained from search form with users, showing that in fact the device can facilitate interaction with the computer.**

***Index Terms*—Cortex M3, Human Interface Device, MPU6050, Accelerometer, Gyroscope, Glove Mouse**

I. Introduction

Since the creation of the first computers, the humanity seeks out to innovate the interactions between users and machines. One of these means of interactions is the mouse, presented for the first time in the 60's by Douglas Engelbart as an X-Y Position Indicator For A Display System. It's safe to affirm that the mouse democratized the use of computers by bringing a more intuitive way of perform commands: dragging and clicking. In the course of time, new models of mouse were being created to attend many needs. For example, mouses with high precision and a great time of response, for gamers. We also have ergonomic mouses, usually used by those who spends a lot of time working on the computer, for providing a greater comfort.

What if we could capture the movements of our hand and use them to movement a cursor, just like the mouse? We would have a device that fits to the user's hand, like a glove. This is the intent of Glouse. It's important to highlight that this application can be extended and improved to new solutions which the capture of hand's movement it's necessary, such as:

- medicine, to the performance of surgeries where great precision is needed;
- industry, in which the gestures of the users can be converted to the movement of some machine;
- games, where the gestures of the player control the actions of his character, leading to a more immersive experience;

II. Related Projects

There are related works in the field which are worth to be mentioned. For instance, Kurata [1] used computer vision techniques to develop a wearable input interface that performs mouse functions, the project was named "Hand-mouse". The Hand-mouse differs from our project, Glouse, by its more complex implementation and the need of a camera. Kataware [2] achieved the same goal by using a bend sensor to perform clicks. Another similar idea in the field of reproducing mouse interactions is the project of Raya et al [3], which uses sensors as a three-axis gyroscope, an accelerometer rand magnetometer to perform mouse functions through head movements. The sensors are located in a kind of helmet, and the project's main goal is to be a viable option for disabled people, mostly children affected by cerebral palsy.

III. Methodology

A. Hardware Platform

The Glouse was developed using the BluePill board which is based on the STM32F103C8 microprocessor [7], which is a chip from the STM32F103 family. This board is commonly called BluePill and it has become very popular for its compact size and low price, being available for 3 dollars on the internet.

STM32F10C38 incorporates an ARM Cortex-M3 high-performance core that operates at a 72 MHz frequency and has embedded flash memories up to 128 Kbytes. The model used in this project is equipped with 64 kB of flash memory and 20 kB of RAM. The board also allows many communications protocols to be used, such as SPI, USART and I^2C. The last one was used to communicate with the MPU6050 modules. Figure 1 shows a simplified Block Diagram containing the different devices and technologies used in this project.

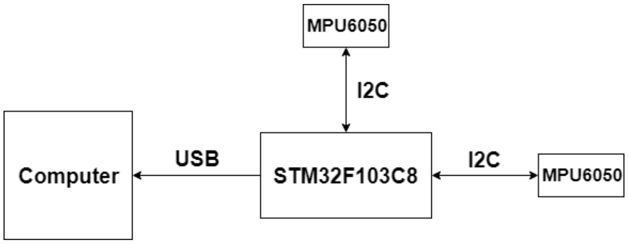

Fig. 1. Block Diagram for the Glouse System

The I^2C protocol developed by Phillips, consists in a multimaster Serial Bus used with low-speed peripherals. Its greatest advantage is reducing the amount of wires needed to connect devices. Using only two wires, SDA and SCL, we can connect up to 1023 slave devices to a single master. SCL is a single-direction clock wire that allows the master to send a clock signal to all slaves. SDA is a bidirectional data wire allowing the master and the slaves to send and receive data. BluePill has two I^2C ports, which allows us to connect up to 2046 slave devices. In order to facilitate the development, both I^2C ports where used to connect the Motion Sensors.

The MPU6050 chip is the responsible for obtaining the correct glove movement measures to be sent to the computer device driver mouse. It is composed by a combination of three smaller devices, a 3-axes gyroscope with a user-programmable fullscale range of $\pm 250,\ \pm 500,\ \pm 1000\ and \pm 2000°/sec$, a 3-axes accelerometer with a programmable full scale range of $\pm 2g,\ \pm 4g,\ \pm 8g\ and \pm 16g$ and a temperature sensor.

B. Implementation

Glouse uses the X and Y axes of the MPU6050s gyroscope to determine the correct movement of the mouse, it maps the x axis of the gyro to the y axis of the screen and the y axis of the gyro to the x axis of the screen. This inversion had to be done in order improve the user experience with the Glouse, also because the axes alignment on the module is different from the one on the screen. It also uses the other MPU6050 module's accelerometer to reproduce the button click movement, when the user moves the index finger down, the accelerometer detects this movement on the X axis, stops the mouse position in order to obtain more stability and perform the click event.

In order to transmit the data obtained by the Gyroscope to the computers mouse driver, the USB Human Interface Device Class (USB HID Class) library of the STM32 was used. It is a specification for computer peripherals that represents a device for human interaction. This library provides methods for sending HID's inputs, such as button click and mouse position. There are also other possible devices that can be emulated such as joysticks, touchscreens and others.

There are a few steps that had to be made in order to send the data to the mouse driver of the computer. First the data is captured with the MPU modules via I^2C. An external software library [5] was used in order to establish the I2C connection. This library provides methods to access the values of the sensors, such as *SD_MPU6050_ReadGyroscope()* and *SD_MPU6050_ReadAccelerometer()* and allow the programmer to easily build applications using the MPU6050 module. It is important to notice that the sensors are continuously streaming data to the BluePill board.

When the data arrives, a filter needs to be applied for smoothing the movement. This filter consists in reducing the value of both x and y axes to a interval that was defined after practical experiments. This is done by comparing the data received from the sensor to two auxiliary variables first set as zero, but updated every loop to the last value of axis x and y, so that the values sent to the computer are only increasing or decreasing the current position not setting a different one. The pseudo code for the filter can be seen in Figure 2. If this filter had not been applied, the user would not see the cursor moving on screen, it would only see the final position after moving it, as if the cursor were teleporting to a new position, not moving there.

Read value from sensor of the back of hand
if $|read\ value - previous\ value| > Moviment\_Constant$ **then**
  $normalize\ read\ value$
  $Moviment\_Value \leftarrow read\ value$
else
  $Moviment\_Value \leftarrow 0$
  $Read\ value\ from\ sensor\ of\ finger$
end if
if $read\_value\_finger < Click\_Constant$ **then**
  $Mouse\_Click \leftarrow true$
else
  $Mouse\_Click \leftarrow false$
  $Send\ Moviment\_Value\ and\ Mouse\_Click\ to\ mouse\ driver$
  $Previous\ value\ \leftarrow read\ value$
end if

Fig. 2. Filtering the data from the MPU6050

Another filter is applied to the index finger sensor. This filter is a simple comparison between the value read on the X axis of the accelerometer and a value determined during experiments. When the x-axis is smaller then this value, it means that the user has done a click movement, that is, he pointed his index finger down. Because of this, this prototype must have both MPU6050 sensors placed at different regions of the glove, if only one sensor is used or both are too close to each other, the movement of the hand that controls the cursor could be confused with the click movement.

After this filter is applied, the movement becomes much more smooth and stable, even though the board is still receiving new positions very fast. This way the user is able to control the direction and speed which the cursor will move around the screen.

When the processing is done, the output is either the button click event or the distance and direction that the cursor on the screen will be moved. This processed data is then sent to the Computer's mouse driver using the HID library above mentioned and processed as a normal input value by the computer's processor. The data flow described above can be seen in Figure 3.

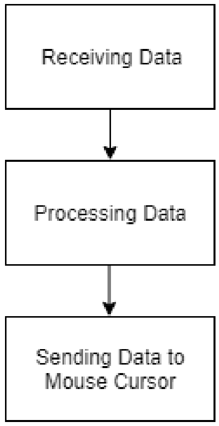

Fig. 3. Data Flow for the Glouse System

Two tools were mainly used to develop the Glouse, those are: STM32CubeMX® and Atollic® TrueSTUDIO®. CubeMX® provides the developer a simple graphical interface to configure and generate C language initialization code. In this project CubeMX® was used to set clock speed, I^2C ports and the USB HID Class. After the project is generated in CubeMX®, Atollic® TrueSTUDIO® is used to implement the main code. This IDE is based on Eclipse and as CubeMX® works with all STM32 boards and it is free of charges.

Regarding to prototype, one MPU6050 has been fixed in index finger, and another one has been fixed in back hand, as can be seen in figures 4 and 5. The MPU6050 fixed at hand back acts as and gyroscope and is connected to por B6 (SCL) and B7 (SDA) of Bluepill, while the MPU 6050 fixed in finger acts as an acceleremeter and is connected to ports B10 and B11 of Bluepill. Power source of system cames from USB 5V, and is converted in bluepill, by an fixed voltage regulator, for 3.3V. Clock source of STM32F103C8 microcontroler cames from an 8KHz crystal in bluepill and is internally multiplied for generating and 72KHz clock for clock of Cortex M3 core. Device is reseted at power on by an RC circuit, or can be reseted by the user trough a push buttonof bluepill.

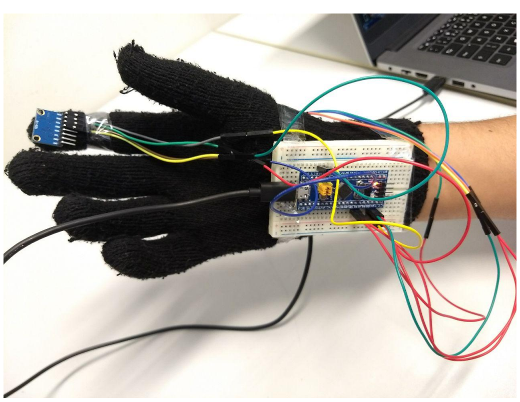

Fig. 4. Glouse front

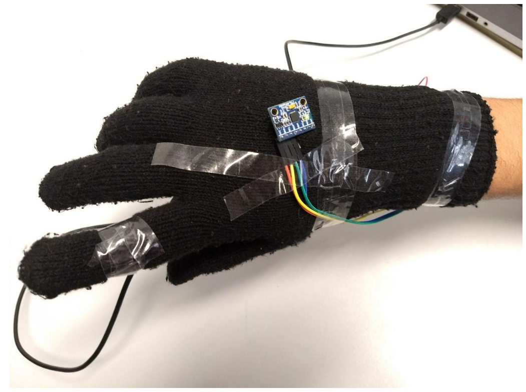

Fig. 5. Glouse back

IV. Evaluation

On the evaluation method, twelve users tested Glouse for a medium period of 15 minutes, in sequence they answered a brief form. In this form, the users evaluated Glouse with a grade from 1 to 5 on the following criteria:

- Mouse cursor movement smoothness;
- Mouse cursor time response;
- Click precision;
- Overall use.

Also, these two affirmations was presented to test users:

1) With due modifications, Glouse could replace the use of actual mouse model;
2) With due modifications, there are real applications which Glouse could be used.

To each one, the user should choose one of the following alternatives:

- Totally Disagree;
- Partially Disagree;
- Neutral;
- Partially Agree;
- Totally Agree.

Unfortunately there was no opportunity to expose the project and receive information from a large amount of people in a timely manner, therefore the statistics results presented here are not conclusive yet. But the real objective of this article session is to highlight the fact that Glouse can be extended to new solutions and show some applications thought by the test users as the difficulties faced by them. This information will help Glouse future development. Figure 6 contains the grades provided by users and Figure 7 shows concordance of test users to the first affirmation that, Glouse could replace the use of actual mouse model with some improvements.

After choosing an alternative to the first affirmation, if the test user fully or partially disagreed, he should write the reason that led to this choice. The main reason related is the movement necessary to move the cursor. To work it's much more efficient use the actual mouse, the complexity of movements necessary it's much inferior and less tiring.

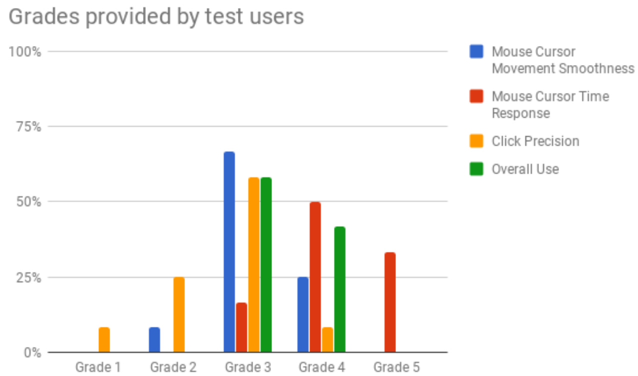

Fig. 6. Grades obtained through a form answered by test users.

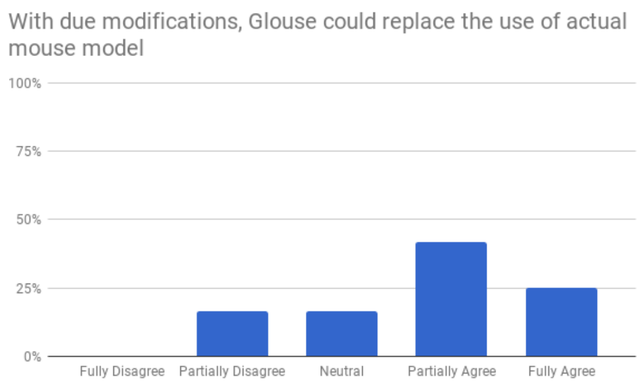

Fig. 7. Test user's response to the first affirmation.

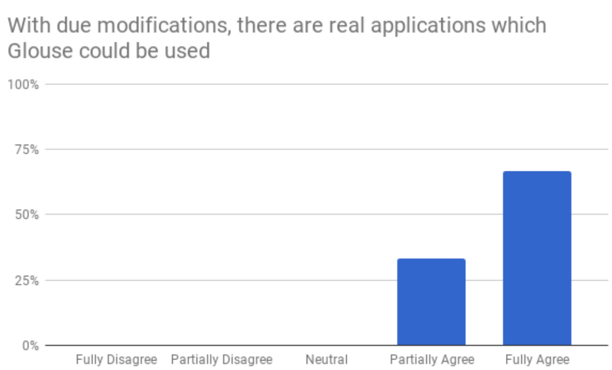

Fig. 8. Test user's response to the second affirmation.

Figure 8 shows concordance of test users to the second affirmation. As you can see, this data is in the path to prove the point: Glouse can be extended to new solutions. As the first affirmation, if the test user fully or partially disagreed, he should write the reason that led to this choice. But if the test user fully or partially agreed, he should write the applications he thought. Most people imagined a device like Glouse to use in slides presentations, with functions like: pass a slide, trigger an animation and play a video. Also there are some answers relating to games, where Glouse could provide more immersion and fun at playing. Lastly, some people also thought about medical applications, where a doctor could use Glouse to control a instrument of high precision in surgeries or another complicated procedure.

Mem. Type	Total	Used	Used (percentage)
RAM	20KB	4.14KB	20.7%
FLASH	64KB	16.16KB	25.3%

TABLE I
APPLICATION AND HAL LIBRARIES SIZE OF GLOUSE

Table I shows the memory size of application and all the necessary HAL (Hardware Abstraction layer) software libraries for Glouse generated by gcc compiler include in Atollic® TrueSTUDIO®. As can be seen, a small room percentage of STM32F103C8 is used, so it can be optimized for a smaller and cheaper device in future versions of Glouse.

V. CONCLUSIONS

Even with the fact that the prototype is very crude, the test users had a good overall experience, yet they still had some problems caused by the click precision, which is a very important feature of a mouse. Therefore, improving the click precision is absolutely necessary to a better Glouse. Future versions of Glouse should be wireless, besides making use of a better filtering for smoothing the movement, like Kalman filter, and fusion of the readings of the acceloremeter and gyroscopes, for increasing precision. With all these improvements, Glouse could really be extended to new solutions in many areas of applications.

In order to use a device such as Glouse to medical application, some other considerations must be made. Since we are speaking about human lives, this product must be more tested and the replacement of the Bluepill and MPU6050 boards for better quality ones should be considered.

In case anyone would like to reproduce it, the entire project is available for download at github [6].

REFERENCES

[1] T. Kurata, T. Okuma, M. Kourogi and K. Sakaue, "The Hand Mouse: GMM hand-color classification and mean shift tracking," Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Vancouver, BC, 2001, pp. 119-124.

[2] Kataware, Yogesh D., and U. L. Bombale. "A Wearable Wireless Device for Effective Human Computer Interaction." International Journal of Computer Applications 99.9 (2014): 9-14.

[3] R. Raya, J.O. Roa, E. Rocon, R. Ceres, J.L. Pons, Wearable inertial mouse for children with physical and cognitive impairments, Sensors and Actuators A: Physical, Volume 162, Issue 2, 2010, Pages 248-259, ISSN 0924-4247

[4] Go, Woo-Young, and Jong-Hwan Kim. "Flexible and Wearable Hand Exoskeleton and Its Application to Computer Mouse." Robot Intelligence Technology and Applications 2. Springer, Cham, 2014. 797-804.

[5] Darvishi, S., Library Software for MPU6050, https://github.com/sinadarvi/SD_HAL_MPU6050

[6] Source code for glouse, https://github.com/pablogrisi/projeto_micro

[7] Medium-density performance line ARM-based 32-bit MCU with 64 or 128 KB Flash datasheet, www.st.com/resource/en/datasheet/stm32f103c8.pdf .