Using Mobile Phone as the Mouse of Your PC

苟铭浩 周潘正 李江彤 刘睿豪 张沛晗

2018.05

Group Members

苟铭浩	515020910069
周潘正	515111910114
李江彤	515110910030
刘睿豪	515021910266
张沛晗	515020910141

Motivation

Mouses are an important auxiliary device for laptops, just like laptops are for our lives. We carry a laptop almost everywhere we go, in most cases, with a mouse. However, it seems quite uncomfortable to have the mouse, something that is not flat and thin, in the bag. And almost everyone had the annoying experience of being forced to use the awkward touchpad when leaving the mouse behind.

Things would be much better if we could use our mobile phones as mouses. A smart phone is obviously more convenient to take, and more unlikely to be forgotten. More importantly, smart phone has everything it takes to be a mouse: the Bluetooth module as the data exchanger, the touch screen as multifunctional buttons and the accelerometer as the motion monitor.

Thus, inspired by such an idea, our goal is to develop a tool that transforms a mobile phone into a Bluetooth mouse for the PC. The project includes an Android App on the mobile side and a Python program on the PC side.

Related Work

Some applications can be found on google play and app store. For example, an app called Monect can transform a phone into not only a mouse or keyboard, but also a joystick,

gamepad, media controller and much more. To control a computer by mobile phone, first of all, two components need to be downloaded, the Android app and the desktop server. The app can communicate with your computer using Wi-Fi or Bluetooth. To use the app as a mouse, you can choose touchpad mode. Touchpad mode turns your phone into a laptop-style trackpad complete with a scroll bar to the right. Other common trackpad actions like a single tap to click and a two-finger drag to scroll up and down will also work.

Many other apps have similar functions and they share some traits in common. First, they are heavily weighted because they include so many functions that lots of using modes are supported. Second and the most importantly, the way they serve as a mouse is not the way actually a mouse works. The phone mimics the function of a laptop touchpad rather than moving around, which a mouse acts. So more accurately, the app turns a phone into a complicated computer controller, not exactly a mouse. Third, according to users' feedback, the connection is not reliable. The mobile phone fails to connect to the computer or the connection fails from time to time.

Based on the discussion above, a light-weighted, reliable-connected app is desired, which turns a phone exactly into a mouse, that is, function as a mouse and move as a mouse.

Technical Issues & Our Contributions

We use the accelerometer in the mobile phone with Android operating system. We use API provided by the operating system to read the accelerometer data.

Problem in Calculating Velocity of the Mouse

In theory, the integration of acceleration is velocity. However, this may not make sense in practice. There is some bias and error with the sensor and the stimulation of small error in acceleration may lead to tremendous error in velocity. In addition, the sum of discrete sampling of acceleration is different from the integration of continuous acceleration. There is also some error caused by discrete sampling. As a result, the common way to calculate velocity has great problem and needs to be improved.

Solutions

In order to solve the problem above, we proposed several solutions:

- Kalman filter is applied on the acceleration data.
- Calibration is applied before using the system to decrease caused by the sensor bias.
- Attenuation of velocity is exerted to avoid divergency of velocity.
- Activation function is used to alleviate the backlash.

Kalman filter

Kalman filter is usually used in dealing with inertial sensor data to remove noise. It's based on the prediction of the next state.

```
ax, xVar = kalmanFilter(ax,xPreData,xVar,Q,R)
ay, yVar = kalmanFilter(ay,yPreData,yVar,Q,R)
```

Calibration

Before using the system, the user should put the mobile phone in the working surface. Then the system set the zero of acceleration to the mean value of static sensor data. After this step, the effect of inclination of the surface and the sensor bias is compensated.

```
ax = x0int / 32.0 * 9.8 - axave
ay = y0int / 32.0 * 9.8 – ayave
```

Attenuation of Velocity

We let the velocity damps with a constant rate so that the velocity will convergent even the integration of acceleration divergent.

```
# damp the speed
vx = vx * dampRate
vy = vy * dampRate
```

Activation function

There will be backlash when stop moving the mouse because of the huge reverse acceleration. It always causes the cursor to move backward which is not expected.

Therefore, we designed an activation function to change the effect of acceleration on the change of velocity. When the velocity is very small, the cursor is static and there is no problem of backlash. As a result, the acceleration should have great effect on the change of velocity. On the contrary, when the cursor is moving fast, huge acceleration is usually cause by stopping the mouse. So, the acceleration shouldn't have a big enough impact on velocity or there will be backlash. In our design, the change of velocity is decided by several factors as shown below:

```
dvx = np.exp(-abs(vx)) * ax * v
dvy = np.exp(-abs(vy)) * ay * v
```

dv is the change of velocity, np.exp is the exponential function provided by NumPy package in python, a is the acceleration and v is a constant to magnify the relative speed which is specified by the user.

As we can see, dv decreases with the increase of the absolute value of v.

Communication Protocol

Hardware: The phone establishes a Bluetooth with the HC-05 Bluetooth chip. Then the chip is connected to USBtoTTL converter while 4 wires. The USBtoTTL converter is inserted into the USB port in the computer. As a result, the data is converted to serial communication byte in the computer.

Encoding: The data to be transmitted can be divided into Three groups. - Acceleration data. - Mouse click data. - Begin, stop and check byte.

We established the following encoding method for these data.

1. X Axis Acceleration Data High Byte(XH)

   $XT = g_x / 9.8 * 32$

   $XH = 128 + floor(XT)$

g_x can be both positive and negative so we add 128 to the original value. According to the expression, the absolute value for g_x should be no more than 4g, or overflow will happen.

2. X Axis Acceleration Data Low Byte(XL)

   XL = floor(256 * (XT - floor(XT)))

3. Y Axis Acceleration Data High Byte(YH)

   YT = g_y / 9.8 * 32

   YH = 128 + floor(YT)

4. Y Axis Acceleration Data Low Byte(YL)

   YL = floor(256 * (YT - floor(YT)))

5. Z Axis Acceleration Data High Byte(ZH)

   ZT = g_z / 9.8 * 32

   ZH = 128 + floor(ZT)

6. Z Axis Acceleration Data Low Byte(ZL)

   ZL = floor(256 * (ZT - floor(ZT)))

7. Left Button Down(LD)

   LD = 0x00

8. Left Button Up(LU)

   LU = 0x01

9. Right Button Down(RD)

   RD = 0xFF

10. Right Button Up(RU)

RU = 0xFE

11. Begin Acceleration Data Transmission(BT)

    BT = 0x16

12. Finish X Axis Data Transmission(FX)

    FX = 0x17

13. Finish Y Axis Data Transmission(FY)

    FY = 0x18

14. Finish Z Axis Data Transmission(FZ)

    FZ = 0x19

Encoding Method with Check: There might be error in transmission. So, we apply Check byte to minimize the error. The Byte sequence is shown below.

1. Acceleration Data

   | BT | XH | XL | FX | YH | YL | FY | ZH | ZL | FZ |

2. Left Button Down

   | LD | LD |

3. Left Button Up

   | LU | LU |

4. Right Button Down

   | RD | RD |

5. Right Button Up

   | RU | RU |

For acceleration data transmission, we can check the FX, FY and FZ to check if the sequence is complete. For click, the acceleration data can also be the value of LD, LU, RD and RU. So, we send two bytes to check if it is really click signal although there is some remote chance that the acceleration data have two consecutive LD, LU, RD or RU. Besides, we set LD, LU, RD and RU to 0x00, 0x01, 0xFF and 0xFE respectively because they are the most improbable bytes to appear in acceleration data.

Encoding Method without Check: The encoding method with check minimize the possibility of occurrence of error. But it consumes to much time, either. This reduces the transmission rate and leads to bad performance of the system, we also introduce an encoding method without check. This method reduces the bytes to be transmitted and accelerates the transmission. In our test, we haven't find any error till now even if we didn't check the byte.

1. Acceleration Data

   | BT | XH | YH |

2. Left Button Down

   | LD |

3. Left Button Up

   | LU |

4. Right Button Down

   | RD |

5. Right Button Up

   | RU |

This is the fewest bytes to be transmitted or the data will be incomplete. You can see that we do not transmit the z axis data because we don't quiet need it. The low byte for x axis

data and y axis data are also ignored. This increases the transmission speed while decreases the accuracy. In our test, the decrease of accuracy doesn't lead to a major problem.

Running Requirement

Mobile Phone:

- Phone with Android OS is required. (tested on EMUI 8.0.0(Huawei))
- Phone with Bluetooth 4 is required, or it cannot communicate with computer

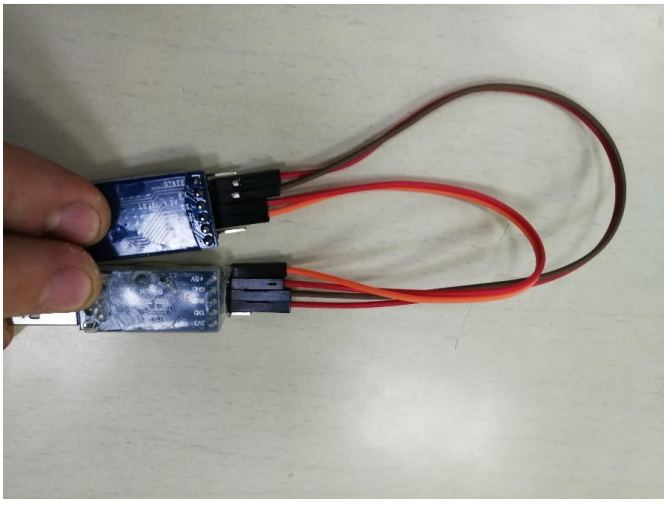
Computer:

- PC with Windows OS is required. (tested on Windows 10 Pro)
- PC with python environment is required. (tested on python 3.5)
- Python package NumPy, serial, win32api and win32con are required. (tested using Anaconda 5.1.0(Python 3.5))

Setup

Mobile Phone: Install the csNetworkNewbeta.apk APP on your Android phone and open the Bluetooth on system setting.

Computer: Insert an external Bluetooth in your USB port. In the test, we use HC-05 Bluetooth module and a ttl serial port to usb converter. The connection is shown in the picture below.



HC-05---------TTL to USB converter

5V ---------- 5V

GND ---------- GND

TX ---------- RX

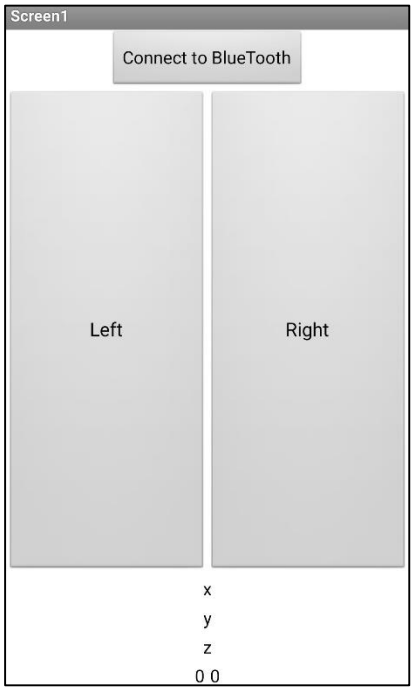
RX ---------- TX

Set the path to the root folder that contains driver.py

Test and Run

Open the "Xiao" APP and click the "Connect to BlueTooth" button then choose the Bluetooth device connected to your PC.

Note that this step must be done before running the python written driver program.



On the computer side, run the command as follows:

```
python driver.py (serial port number) [(cursor move speed)] [(threshhold for moving)] [(damp rate)] [(Q)] [(R)]
```

We can just give the first few parameters that is included in '[' and ']' and the parameters will be set the default value. Some of the few examples are given below:

```
python driver.py COM3 20
python driver.py COM3 20 0.4
python driver.py COM3 20 0.4 0.8
python driver.py COM3 20 0.4 0.8 0.1
python driver.py COM3 20 0.4 0.8 0.1 0.25
```

These values are also the default value and as a result, these commands have the same effect.

Future Work

1. Improve the accuracy of the mouse
2. Rewrite a Android using Java so that we can do the calculation on the mobile phone and no driver needed to be installed on computer
3. Try to adapt the communication protocol to the HID standard so that we don’t need an external Bluetooth receiver