

# Using Your Mobile Phone as the Mouse to Control Your Computer

---

## Developer

- [Minghao Gou](#), School of Mechanical Engineering, Shanghai Jiao Tong University
- Panzheng Zhou, School of Chemistry, Shanghai Jiao Tong University
- [Jiangtong Li](#), Zhiyuan College, Shanghai Jiao Tong University
- Ruihao liu, School of Biomedical Engineering, Shanghai Jiao Tong University
- [Peihan Zhang](#), School of Mechanical Engineering, Shanghai Jiao Tong University

## How It Works

### Acceleration data

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 design 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,  $\text{np.exp}$  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)**

$$XT = g_x / 9.8 * 32$$

$$XH = 128 + \text{floor}(XT)$$

$g_x$  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)**

$$XL = \text{floor}(256 * (XT - \text{floor}(XT)))$$

### **3. Y Axis Acceleration Data High Byte(YH)**

$$YT = g_y / 9.8 * 32$$

$$YH = 128 + \text{floor}(YT)$$

### **4. Y Axis Acceleration Data Low Byte(YL)**

$$YL = \text{floor}(256 * (YT - \text{floor}(YT)))$$

### **5. Z Axis Acceleration Data High Byte(ZH)**

$$ZT = g_z / 9.8 * 32$$

$$ZH = 128 + \text{floor}(ZT)$$

### **6. Z Axis Acceleration Data Low Byte(ZL)**

$$ZL = \text{floor}(256 * (ZT - \text{floor}(ZT)))$$

### **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 minimizes the possibility of occurrence of error. But it consumes too 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 found 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 ingored. This increases the transmission speed while decreases the accurate. 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 a 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

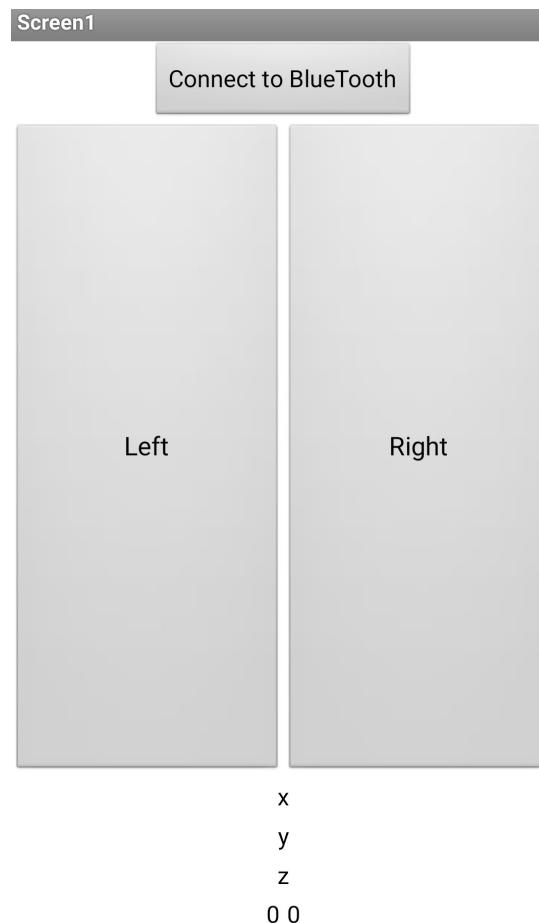
Set the path to the root folder that contains driver.py

## Run

### Phone

Open the "Xiao" APP and click the "Connect to BlueTooth" button then chooee the bluetooth device conneted to your PC.

**We should pay attention that this step must be done before run the python written driver program.**



## Computer

Run the command as follows:

```
python driver.py (serial port number) [(cursor move speed)] [(threshold 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.

## Calibration

Make sure that the xiao APP is running without lock the screen when the program begins and put the mobile phone on the surface where you want to use this program.  
The program will print some of the information and after that it's available to use the mouse to control your computer.