

```
#include <Servo.h>#include
```

```
char input = ' ';
```

```
int stepSize = 6;  
int stepSize2 = 2;
```

```
Servo thumbServo;  
Servo indexServo;  
Servo majeureServo;  
Servo ringfingerServo;
```

```
void setup() {  
  thumbServo.attach(2);  
  indexServo.attach(3);  
  majeureServo.attach(4);  
  ringfingerServo.attach(5);  
  Serial.begin(9600);  
  
}
```

```
void loop() {
```

```
  if(input == ' '){return;}  
  int tPos = thumbServo.read();  
  int iPos = indexServo.read();  
  int mPos = majeureServo.read();  
  int rPos = ringfingerServo.read();
```

```
  switch(input){  
    case 'c':  
      tPos += stepSize;  
      break;  
    case 'v':  
      tPos -= stepSize;  
      break;  
    case 'e':  
      iPos += stepSize;  
      break;  
    case 'd':  
      iPos -= stepSize;  
      break;  
    case 'z':  
      mPos += stepSize;  
      break;  
    case 's':  
      mPos -= stepSize;  
      break;
```

```

case 'a':
rPos += stepSize;
break;
case 'q':
rPos -= stepSize;
break;
case 'n':
tPos += stepSize2;
break;
case 'b':
tPos -= stepSize2;
break;
case 'i':
iPos += stepSize2;
break;
case 'k':
iPos -= stepSize2;
break;
case 'o':
mPos += stepSize2;
break;
case 'l':
mPos -= stepSize2;
break;
case 'p':
rPos += stepSize2;
break;
case 'm':
rPos -= stepSize2;
break;

}
tPos = constrain(tPos, 0, 180);
iPos = constrain(iPos, 0, 180);
mPos = constrain(mPos, 0, 180);
rPos = constrain(rPos, 0, 180);


if(tPos != thumbServo.read()){
Serial.println("writing tPos: " + (String) tPos);
thumbServo.write(tPos);
}
if(iPos != indexServo.read()){
Serial.println("writing iPos: " + (String) iPos);
indexServo.write(iPos);
}
if(mPos != majeureServo.read()){
Serial.println("writing mPos: " + (String) mPos);
majeureServo.write(mPos);
}
if(rPos != ringfingerServo.read()){

```

```
Serial.println("writing rPos: " + (String) rPos);  
ringfingerServo.write(rPos);  
}
```

```
input = ' ';  
}
```

```
/*  
SerialEvent occurs whenever a new data comes in the  
hardware serial RX. This routine is run between each  
time loop() runs, so using delay inside loop can delay  
response. Multiple bytes of data may be available.  
*/  
void serialEvent() {  
  while (Serial.available()) {  
    // get the new byte:  
    char inChar = (char)Serial.read();  
    input = inChar;  
  
  }  
}
```