firmread / hidmouse_diff.ino

Last active 2 minutes ago

Revisions

firmread revised this gist 2 minutes ago.

1 changed file with **110 additions** and **54 deletions**.

164 hidmouse_diff.ino


```
...  @@ -1,17 +1,17 @@
 1   /*************************************************************************
 2  - This is an example for our nRF51822 based Bluefruit LE modules
 3
 4  - Pick one up today in the adafruit shop!
 5
 6  - Adafruit invests time and resources providing this open source code,
 7  - please support Adafruit and open-source hardware by purchasing
 8  - products from Adafruit!
 9
10  - MIT license, check LICENSE for more information
11  - All text above, and the splash screen below must be included in
12  - any redistribution
13   *************************************************************************
14  -/*
15     This example shows how to send HID (keyboard/mouse/etc) data via BLE
16     Note that not all devices support BLE Mouse!
17     - OSX, Windows 10 both work
...  @@ -20,50 +20,57 @@
20   */
21
22   #include <Arduino.h>
23
23   #include <SPI.h>
24   #if not defined (_VARIANT_ARDUINO_DUE_X_)
25  -  #include <SoftwareSerial.h>
26   #endif
27
28   #include "Adafruit_BLE.h"
29   //#include "Adafruit_BluefruitLE_SPI.h"
30   #include "Adafruit_BluefruitLE_UART.h"
31  -
32   #include "BluefruitConfig.h"
33
34   /*=========================================================================
35       APPLICATION SETTINGS
36
37  -    FACTORYRESET_ENABLE       Perform a factory reset when running this sketch
38  -
39  -                              Enabling this will put your Bluefruit LE module
40                                 in a 'known good' state and clear any config
41                                 data set in previous sketches or projects, so
42  -                              running this at least once is a good idea.
43  -
44  -                              When deploying your project, however, you will
45                                 want to disable factory reset by setting this
46                                 value to 0.  If you are making changes to your
47  -                              Bluefruit LE device via AT commands, and those
48                                 changes aren't persisting across resets, this
49                                 is the reason why.  Factory reset will erase
50                                 the non-volatile memory where config data is
51                                 stored, setting it back to factory default
52                                 values.
53  -
54  -                              Some sketches that require you to bond to a
55                                 central device (HID mouse, keyboard, etc.)
56                                 won't work at all with this feature enabled
57                                 since the factory reset will clear all of the
58                                 bonding data stored on the chip, meaning the
59                                 central device won't be able to reconnect.
60       MINIMUM_FIRMWARE_VERSION  Minimum firmware version to have some new features
61       -------------------------------------------------------------------------*/
62  -    #define FACTORYRESET_ENABLE         0
63  -    #define MINIMUM_FIRMWARE_VERSION    "0.6.6"
64   /*=========================================================================*/
65
66  - Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
67
68   // A small helper
69   void error(const __FlashStringHelper*err) {
...  @@ -76,14 +83,21 @@ void error(const __FlashStringHelper*err) {
76       @brief  Sets up the HW an the BLE module (this function is called
77               automatically on startup)
78
79  -/**************************************************************************/
80   void setup(void)
81   {
82     while (!Serial);  // required for Flora & Micro
83     delay(500);
84
85     Serial.begin(115200);
86     Serial.println(F("Adafruit Bluefruit HID Mouse Example"));
87  -
88     Serial.println(F("---------------------------------------"));
89
90     /* Initialise the module */
...  @@ -99,7 +113,7 @@ void setup(void)
99     {
100      /* Perform a factory reset to make sure everything is in a known state */
101      Serial.println(F("Performing a factory reset: "));
102 -    if ( ! ble.factoryReset() ){
103        error(F("Couldn't factory reset"));
```

```
 1   /*************************************************************************
 2  + This is an example for our nRF51822 based Bluefruit LE modules
 3
 4  + Pick one up today in the adafruit shop!
 5
 6  + Adafruit invests time and resources providing this open source code,
 7  + please support Adafruit and open-source hardware by purchasing
 8  + products from Adafruit!
 9
10  + MIT license, check LICENSE for more information
11  + All text above, and the splash screen below must be included in
12  + any redistribution
13   *************************************************************************
14  +/*
15     This example shows how to send HID (keyboard/mouse/etc) data via BLE
16     Note that not all devices support BLE Mouse!
17     - OSX, Windows 10 both work
20   */
21
22   #include <Arduino.h>
23  +#include <Wire.h>
24   #include <SPI.h>
25   #if not defined (_VARIANT_ARDUINO_DUE_X_)
26  +#include <SoftwareSerial.h>
27   #endif
28
29   #include "Adafruit_BLE.h"
30   //#include "Adafruit_BluefruitLE_SPI.h"
31   #include "Adafruit_BluefruitLE_UART.h"
32   #include "BluefruitConfig.h"
33
34  +//int counter = 0;
35  +const int MPU = 0x68; // I2C address of the MPU-6050
36  +//int16_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
37  +int8_t AcX, AcY, AcZ, Tmp, GyX, GyY, GyZ;
38  +
39  +boolean bDidType = false;
40  +
41   /*=========================================================================
42       APPLICATION SETTINGS
43
44  +    FACTORYRESET_ENABLE       Perform a factory reset when running this sketch
45  +
46  +                              Enabling this will put your Bluefruit LE module
47                                 in a 'known good' state and clear any config
48                                 data set in previous sketches or projects, so
49  +                              running this at least once is a good idea.
50  +
51  +                              When deploying your project, however, you will
52                                 want to disable factory reset by setting this
53                                 value to 0.  If you are making changes to your
54  +                              Bluefruit LE device via AT commands, and those
55                                 changes aren't persisting across resets, this
56                                 is the reason why.  Factory reset will erase
57                                 the non-volatile memory where config data is
58                                 stored, setting it back to factory default
59                                 values.
60  +
61  +                              Some sketches that require you to bond to a
62                                 central device (HID mouse, keyboard, etc.)
63                                 won't work at all with this feature enabled
64                                 since the factory reset will clear all of the
65                                 bonding data stored on the chip, meaning the
66                                 central device won't be able to reconnect.
67       MINIMUM_FIRMWARE_VERSION  Minimum firmware version to have some new features
68       -------------------------------------------------------------------------*/
69  +#define FACTORYRESET_ENABLE         0
70  +#define MINIMUM_FIRMWARE_VERSION    "0.6.6"
71   /*=========================================================================*/
72
73  +Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
74
75   // A small helper
76   void error(const __FlashStringHelper*err) {
83       @brief  Sets up the HW an the BLE module (this function is called
84               automatically on startup)
85
86  +/**************************************************************************/
87   void setup(void)
88   {
89     while (!Serial);  // required for Flora & Micro
90     delay(500);
91  +
92  +  Wire.begin();
93  +  Wire.beginTransmission(MPU);
94  +  Wire.write(0x6B);  // PWR_MGMT_1 register
95  +  Wire.write(0);     // set to zero (wakes up the MPU-6050)
96  +  Wire.endTransmission(true);
97  +
98     Serial.begin(115200);
99     Serial.println(F("Adafruit Bluefruit HID Mouse Example"));
100 +
101    Serial.println(F("---------------------------------------"));
102
103    /* Initialise the module */
113    {
114      /* Perform a factory reset to make sure everything is in a known state */
115      Serial.println(F("Performing a factory reset: "));
116 +    if ( ! ble.factoryReset() ){
117        error(F("Couldn't factory reset"));
```

```
    /* Perform a factory reset to make sure everything is in a known state */
    Serial.println(F("Performing a factory reset: "));
-   if ( ! ble.factoryReset() ){
+   if ( ! ble.factoryReset() ){
      error(F("Couldn't factory reset"));
    }
  }
...
@@ -111,6 +125,12 @@ void setup(void)
  /* Print Bluefruit information */
  ble.info();

+ /* Change the device name to make it easier to find */
+ Serial.println(F("Setting device name to 'Bluefruit clickKick': "));
+ if (! ble.sendCommandCheckOK(F( "AT+GAPDEVNAME=Bluefruit clickKick" )) ) {
+   error(F("Could not set device name?"));
+ }
+
  // This demo only available for firmware from 0.6.6
  if ( !ble.isVersionAtLeast(MINIMUM_FIRMWARE_VERSION) )
  {
...
@@ -147,41 +167,76 @@ void setup(void)
     @brief  Constantly poll for new command or response data
 */
 /**************************************************************************/
-void loop(void)
-{
-  Serial.println(F("x,y = "));
+void loop(void) {
+
+ //-----Notification Input from Sensor
+ Wire.beginTransmission(MPU);
+ Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
+ Wire.endTransmission(false);
+ Wire.requestFrom(MPU, 14, true); // request a total of 14 registers
+ AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
+ AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
+ AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
+ Tmp = Wire.read() << 8 | Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
+ GyX = Wire.read() << 8 | Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
+ GyY = Wire.read() << 8 | Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
+ GyZ = Wire.read() << 8 | Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
+
+ Serial.print(" | AcX = "); Serial.print(AcX);
+ Serial.print(" | AcY = "); Serial.print(AcY);
+ Serial.print(" | AcZ = "); Serial.print(AcZ);
+ Serial.print(" | Tmp = "); Serial.print(Tmp / 340.00 + 36.53); //equation for temperature in degrees C from datasheet
+ Serial.print(" | GyX = "); Serial.print(GyX);
+ Serial.print(" | GyY = "); Serial.print(GyY);
+ Serial.print(" | GyZ = "); Serial.println(GyZ);
+
+ //Serial.println(F("x,y = "));

  // Check for user input and echo it back if anything was found
- char input[BUFSIZE+1];
- getUserInput(input, BUFSIZE);
+ char input[BUFSIZE + 1];
+ //getUserInput(input, BUFSIZE);

  Serial.println(input);

+ /* ----------Mouse Action
+    BLE Services for Command mode
+    https://learn.adafruit.com/introducing-adafruit-ble-bluetooth-low-energy-friend/ble-services#at-plus-blehidmousemove
+ */
  // Press (and hold) the Left mouse's button
  if ( ble.sendCommandCheckOK(F("AT+BleHidMouseButton=L,press")) )
  {
    // delay a bit
    delay(250);

    // Mouse moves according to the user's input
-   ble.print(F("AT+BleHidMouseMove="));
-   ble.println(input);
+   // Parameter: X Ticks (+/-), Y Ticks (+/-), Scroll Wheel (+/-), Pan Wheel (+/-)

-   if( ble.waitForOK() )
-   {
+   //Right
+   if (AcY > 0) {
+     ble.print(F("AT+BleHidMouseMove= 2,,,"));
+     ble.println(input);
+   }
+
+   //Up
+   if(input){
+     ble.print(F("AT+BleHidMouseMove= ,-2,,"));
+     ble.println(input);
+     }
+
+   if ( ble.waitForOK() ){
      Serial.println( F("OK!") );
-   }else
+   }else
    {
      Serial.println( F("FAILED!") );
    }

    // Way for user to release left button
-   Serial.println( F("Enter anything to release Left Button") );
-   getUserInput(input, BUFSIZE);
+   Serial.println(F("Enter anything to release Left Button") );
+//   getUserInput(input, BUFSIZE);

    // Release the Left mouse's button
    ble.sendCommandCheckOK(F("AT+BleHidMouseButton=0"));
- }else
+ }else
  {
    // Failed, probably pairing is not complete yet
    Serial.println( F("Please make sure Bluefruit is paired and try again") );
...
@@ -193,18 +248,19 @@ void loop(void)
     @brief  Checks for user input (via the Serial Monitor)
 */
 /**************************************************************************/
-void getUserInput(char buffer[], uint8_t maxSize)
-{
-  memset(buffer, 0, maxSize);
-  while( Serial.available() == 0 ) {
-    delay(1);
-  }
-
-  uint8_t count=0;
-
-  do
-  {
-    count += Serial.readBytes(buffer+count, maxSize);
-    delay(2);
-  } while( (count < maxSize) && !(Serial.available() == 0) );
-}
+//void getUserInput(char buffer[], uint8_t maxSize)
+//{
+//  // Sets the first num bytes of the block of memory pointed
+//  memset(buffer, 0, maxSize);
+//  while ( Serial.available() == 0 ) {
+//    delay(1);
+//  }
+//
+//  uint8_t count = 0;
+//
+//  do
+//  {
+//    count += Serial.readBytes(buffer + count, maxSize);
+//    delay(2);
+//  } while ( (count < maxSize) && !(Serial.available() == 0) );
+//}
```

firmread revised this gist 4 minutes ago.

1 changed file with 6 additions and 25 deletions.

31 ■■■■□ hidmouse_diff.ino — View

```
...
@@ -10,8 +10,7 @@
  MIT license, check LICENSE for more information
  All text above, and the splash screen below must be included in
  any redistribution
-**************************************************************************/
+**************************************************************************
```

```
  MIT license, check LICENSE for more information
  All text above, and the splash screen below must be included in
  any redistribution
-**********************************************************************/
-
 /*
   This example shows how to send HID (keyboard/mouse/etc) data via BLE
   Note that not all devices support BLE Mouse!
@@ -22,12 +21,12 @@

 #include <Arduino.h>
 #include <SPI.h>
-#if not defined (_VARIANT_ARDUINO_DUE_X_) && not defined(ARDUINO_ARCH_SAMD)
+#if not defined (_VARIANT_ARDUINO_DUE_X_)
   #include <SoftwareSerial.h>
 #endif

 #include "Adafruit_BLE.h"
-#include "Adafruit_BluefruitLE_SPI.h"
+//#include "Adafruit_BluefruitLE_SPI.h"
 #include "Adafruit_BluefruitLE_UART.h"

 #include "BluefruitConfig.h"
@@ -64,25 +63,7 @@
     #define MINIMUM_FIRMWARE_VERSION    "0.6.6"
 /*=========================================================================*/

-
-// Create the bluefruit object, either software serial...uncomment these lines
-/*
-SoftwareSerial bluefruitSS = SoftwareSerial(BLUEFRUIT_SWUART_TXD_PIN,
BLUEFRUIT_SWUART_RXD_PIN);
-
-Adafruit_BluefruitLE_UART ble(bluefruitSS, BLUEFRUIT_UART_MODE_PIN,
-                      BLUEFRUIT_UART_CTS_PIN, BLUEFRUIT_UART_RTS_PIN);
-*/
-
-/* ...or hardware serial, which does not need the RTS/CTS pins. Uncomment this line */
-// Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
-
-/* ...hardware SPI, using SCK/MOSI/MISO hardware SPI pins and then user selected CS/IRQ/RST */
-Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS, BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);
-
-/* ...software SPI, using SCK/MOSI/MISO user-defined SPI pins and then user selected CS/IRQ/RST */
-//Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_SCK, BLUEFRUIT_SPI_MISO,
-//                             BLUEFRUIT_SPI_MOSI, BLUEFRUIT_SPI_CS,
-//                             BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);
+Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);

 // A small helper
 void error(const __FlashStringHelper*err) {
@@ -94,15 +75,15 @@ void error(const __FlashStringHelper*err) {
 /*!
     @brief  Sets up the HW an the BLE module (this function is called
             automatically on startup)
-*/
+*/
 /**************************************************************************/
 void setup(void)
 {
   while (!Serial);  // required for Flora & Micro
   delay(500);
-
+
   Serial.begin(115200);
   Serial.println(F("Adafruit Bluefruit HID Mouse Example"));

   Serial.println(F("---------------------------------------"));

   /* Initialise the module */
```

firmread created this gist 5 minutes ago.

229 ■■■■■ hidmouse_diff.ino View

```
@@ -0,0 +1,229 @@
+/*********************************************************************
+ This is an example for our nRF51822 based Bluefruit LE modules
+
+ Pick one up today in the adafruit shop!
+
+ Adafruit invests time and resources providing this open source code,
+ please support Adafruit and open-source hardware by purchasing
+ products from Adafruit!
+
+ MIT license, check LICENSE for more information
+ All text above, and the splash screen below must be included in
+ any redistribution
+*********************************************************************/
+
+/*
+  This example shows how to send HID (keyboard/mouse/etc) data via BLE
+  Note that not all devices support BLE Mouse!
+  - OSX, Windows 10 both work
+  - Android has limited support
+  - iOS completely ignores mouse
+*/
+
+#include <Arduino.h>
+#include <SPI.h>
+#if not defined (_VARIANT_ARDUINO_DUE_X_) && not defined(ARDUINO_ARCH_SAMD)
+  #include <SoftwareSerial.h>
+#endif
+
+#include "Adafruit_BLE.h"
+#include "Adafruit_BluefruitLE_SPI.h"
+#include "Adafruit_BluefruitLE_UART.h"
+
+#include "BluefruitConfig.h"
+
+/*=========================================================================
+    APPLICATION SETTINGS
+
+    FACTORYRESET_ENABLE       Perform a factory reset when running this sketch
+
+                              Enabling this will put your Bluefruit LE module
+                              in a 'known good' state and clear any config
+                              data set in previous sketches or projects, so
+                              running this at least once is a good idea.
+
+                              When deploying your project, however, you will
+                              want to disable factory reset by setting this
+                              value to 0.  If you are making changes to your
+                              Bluefruit LE device via AT commands, and those
+                              changes aren't persisting across resets, this
+                              is the reason why.  Factory reset will erase
+                              the non-volatile memory where config data is
+                              stored, setting it back to factory default
+                              values.
+
+                              Some sketches that require you to bond to a
+                              central device (HID mouse, keyboard, etc.)
+                              won't work at all with this feature enabled
```

```
+                            Some sketches that require you to bond to a
+                            central device (HID mouse, keyboard, etc.)
+                            won't work at all with this feature enabled
+                            since the factory reset will clear all of the
+                            bonding data stored on the chip, meaning the
+                            central device won't be able to reconnect.
+    MINIMUM_FIRMWARE_VERSION  Minimum firmware version to have some new features
+    ----------------------------------------------------------------------*/
+    #define FACTORYRESET_ENABLE         0
+    #define MINIMUM_FIRMWARE_VERSION    "0.6.6"
+/*=========================================================================*/
+
+
+// Create the bluefruit object, either software serial...uncomment these lines
+/*
+SoftwareSerial bluefruitSS = SoftwareSerial(BLUEFRUIT_SWUART_TXD_PIN,
BLUEFRUIT_SWUART_RXD_PIN);
+
+Adafruit_BluefruitLE_UART ble(bluefruitSS, BLUEFRUIT_UART_MODE_PIN,
+                      BLUEFRUIT_UART_CTS_PIN, BLUEFRUIT_UART_RTS_PIN);
+*/
+
+/* ...or hardware serial, which does not need the RTS/CTS pins. Uncomment this line */
+// Adafruit_BluefruitLE_UART ble(BLUEFRUIT_HWSERIAL_NAME, BLUEFRUIT_UART_MODE_PIN);
+
+/* ...hardware SPI, using SCK/MOSI/MISO hardware SPI pins and then user selected CS/IRQ/RST */
+Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS, BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);
+
+/* ...software SPI, using SCK/MOSI/MISO user-defined SPI pins and then user selected CS/IRQ/RST
*/
+//Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_SCK, BLUEFRUIT_SPI_MISO,
+//                             BLUEFRUIT_SPI_MOSI, BLUEFRUIT_SPI_CS,
+//                             BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);
+
+// A small helper
+void error(const __FlashStringHelper*err) {
+  Serial.println(err);
+  while (1);
+}
+
+/**************************************************************************/
+/*!
+    @brief  Sets up the HW an the BLE module (this function is called
+            automatically on startup)
+*/
+/**************************************************************************/
+void setup(void)
+{
+  while (!Serial);  // required for Flora & Micro
+  delay(500);
+
+  Serial.begin(115200);
+  Serial.println(F("Adafruit Bluefruit HID Mouse Example"));
+  Serial.println(F("---------------------------------------"));
+
+  /* Initialise the module */
+  Serial.print(F("Initialising the Bluefruit LE module: "));
+
+  if ( !ble.begin(VERBOSE_MODE) )
+  {
+    error(F("Couldn't find Bluefruit, make sure it's in CoMmanD mode & check wiring?"));
+  }
+  Serial.println( F("OK!") );
+
+  if ( FACTORYRESET_ENABLE )
+  {
+    /* Perform a factory reset to make sure everything is in a known state */
+    Serial.println(F("Performing a factory reset: "));
+    if ( ! ble.factoryReset() ){
+      error(F("Couldn't factory reset"));
+    }
+  }
+
+  /* Disable command echo from Bluefruit */
+  ble.echo(false);
+
+  Serial.println("Requesting Bluefruit info:");
+  /* Print Bluefruit information */
+  ble.info();
+
+  // This demo only available for firmware from 0.6.6
+  if ( !ble.isVersionAtLeast(MINIMUM_FIRMWARE_VERSION) )
+  {
+    error(F("This sketch requires firmware version " MINIMUM_FIRMWARE_VERSION " or higher!"));
+  }
+
+  /* Enable HID Service (including Mouse) */
+  Serial.println(F("Enable HID Service (including Mouse): "));
+  if (! ble.sendCommandCheckOK(F( "AT+BleHIDEn=On"  ))) {
+    error(F("Failed to enable HID (firmware >=0.6.6?)"));
+  }
+
+  /* Add or remove service requires a reset */
+  Serial.println(F("Performing a SW reset (service changes require a reset): "));
+  if (! ble.reset() ) {
+    error(F("Could not reset??"));
+  }
+
+  Serial.println();
+  Serial.println(F("Go to your phone's Bluetooth settings to pair your device"));
+  Serial.println(F("then open an application that accepts mouse input"));
+  Serial.println();
+
+  Serial.println(F("The example will try to draw a rectangle using the left mouse button with
your input"));
+  Serial.println(F("Parameters are a pair of 8-bit signed numbers (x,y) e.g:"));
+  Serial.println(F("  100,100  : draw toward bottom right corner"));
+  Serial.println(F("  -100,-100: draw toward top left corner"));
+
+  Serial.println();
+}
+
+/**************************************************************************/
+/*!
+    @brief  Constantly poll for new command or response data
+*/
+/**************************************************************************/
+void loop(void)
+{
+  Serial.println(F("x,y = "));
+
+  // Check for user input and echo it back if anything was found
+  char input[BUFSIZE+1];
+  getUserInput(input, BUFSIZE);
+
+  Serial.println(input);
+
+  // Press (and hold) the Left mouse's button
+  if ( ble.sendCommandCheckOK(F("AT+BleHidMouseButton=L,press")) )
+  {
+    // delay a bit
```

```
+  // Press (and hold) the Left mouse's button
+  if ( ble.sendCommandCheckOK(F("AT+BleHidMouseButton=L,press")) )
+  {
+    // delay a bit
+    delay(250);
+
+    // Mouse moves according to the user's input
+    ble.print(F("AT+BleHidMouseMove="));
+    ble.println(input);
+
+    if( ble.waitForOK() )
+    {
+      Serial.println( F("OK!") );
+    }else
+    {
+      Serial.println( F("FAILED!") );
+    }
+
+    // Way for user to release left button
+    Serial.println( F("Enter anything to release Left Button") );
+    getUserInput(input, BUFSIZE);
+
+    // Release the Left mouse's button
+    ble.sendCommandCheckOK(F("AT+BleHidMouseButton=0"));
+  }else
+  {
+    // Failed, probably pairing is not complete yet
+    Serial.println( F("Please make sure Bluefruit is paired and try again") );
+  }
+}
+
+/**************************************************************************/
+/*!
+    @brief  Checks for user input (via the Serial Monitor)
+*/
+/**************************************************************************/
+void getUserInput(char buffer[], uint8_t maxSize)
+{
+  memset(buffer, 0, maxSize);
+  while( Serial.available() == 0 ) {
+    delay(1);
+  }
+
+  uint8_t count=0;
+
+  do
+  {
+    count += Serial.readBytes(buffer+count, maxSize);
+    delay(2);
+  } while( (count < maxSize) && !(Serial.available() == 0) );
+}⊘↵
```