

BLE HID Mouse

1.0

Features

- BLE HID over GATT Profile in HID Device Role operation
- Simulates mouse moving in square clockwise position
- Simulates battery charging
- DeepSleep mode demonstration
- Reporting workflow status through UART
- LED status indication

General Description

This project demonstrates the mouse movement and button click HID reports in the boot and protocol mode. The example also demonstrates handling the suspend event from the central device and enters the low power mode when suspended.

Development Kit Configuration

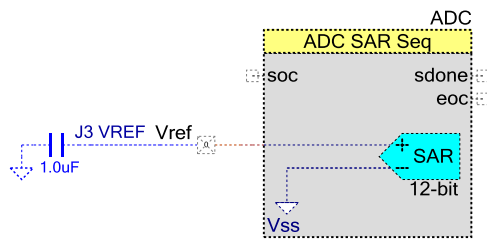
The device has the following configuration:

- The UART RX pin is connected to port 1 pin 4.
- The UART TX pin is connected to port 1 pin 5.
- A mechanical button (port 2 pin 7) is used for a mouse click, to wake up the device, start re-advertising.
- The green LED (port 3 pin 6) is used to indicate an advertising state.
- The red LED (port 2 pin 6) is used to indicate a BLE disconnection state.
- The blue LED (port 3 pin 7) is used to indicate simulation events.
- Connect the CY8CKIT-042 BLE Pioneer Kit board to the PC using a USB cable.
- Launch any of the RS-232 terminal applications on the PC and configure it to use the 'KitProg USB-UART' port with the speed of 115200 bps.
- Build the project and program the hex file into the CY8CKIT-042 BLE Pioneer Kit.
- Observe the results on the terminal application.

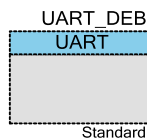
Project Configuration

The example project consists of the following components: BLE, UART, digital output pin, and digital input pin. ADC is not used by default. Battery voltage measurement could be enabled in bas.h header file. The UART is used for transmitting the debug information. The output pins are used to reflect the line signal output on the LED. The input pin is configured to the resistive pull up mode and is used as a mouse click button. The top design schematic is shown in **Figure 1**.

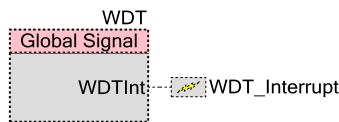
BLE HID Mouse Example Project



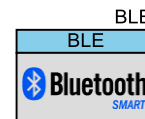
ADC measures battery voltage.



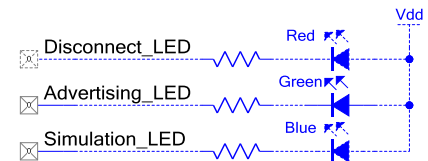
UART is used for transmitting the debug information.



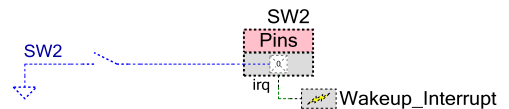
WDT is used as a generic timer for mouse event simulation.



BLE component is configured as HID device and simulates mouse moving in a square clockwise direction.



The red LED is used to indicate that the device is disconnected. The green LED is used to indicate that the device is advertising. The blue LED is used to indicate an events simulation.



The button is used to wake the device up from hibernate mode and a mouse click.

Figure 1. Top design schematic

The BLE component is configured as HID over the GATT Profile in the HID Device role (GATT Server). The HID Device has one instance of the HID Service, Battery Service, Device Information Service, and Scan Parameters Service.

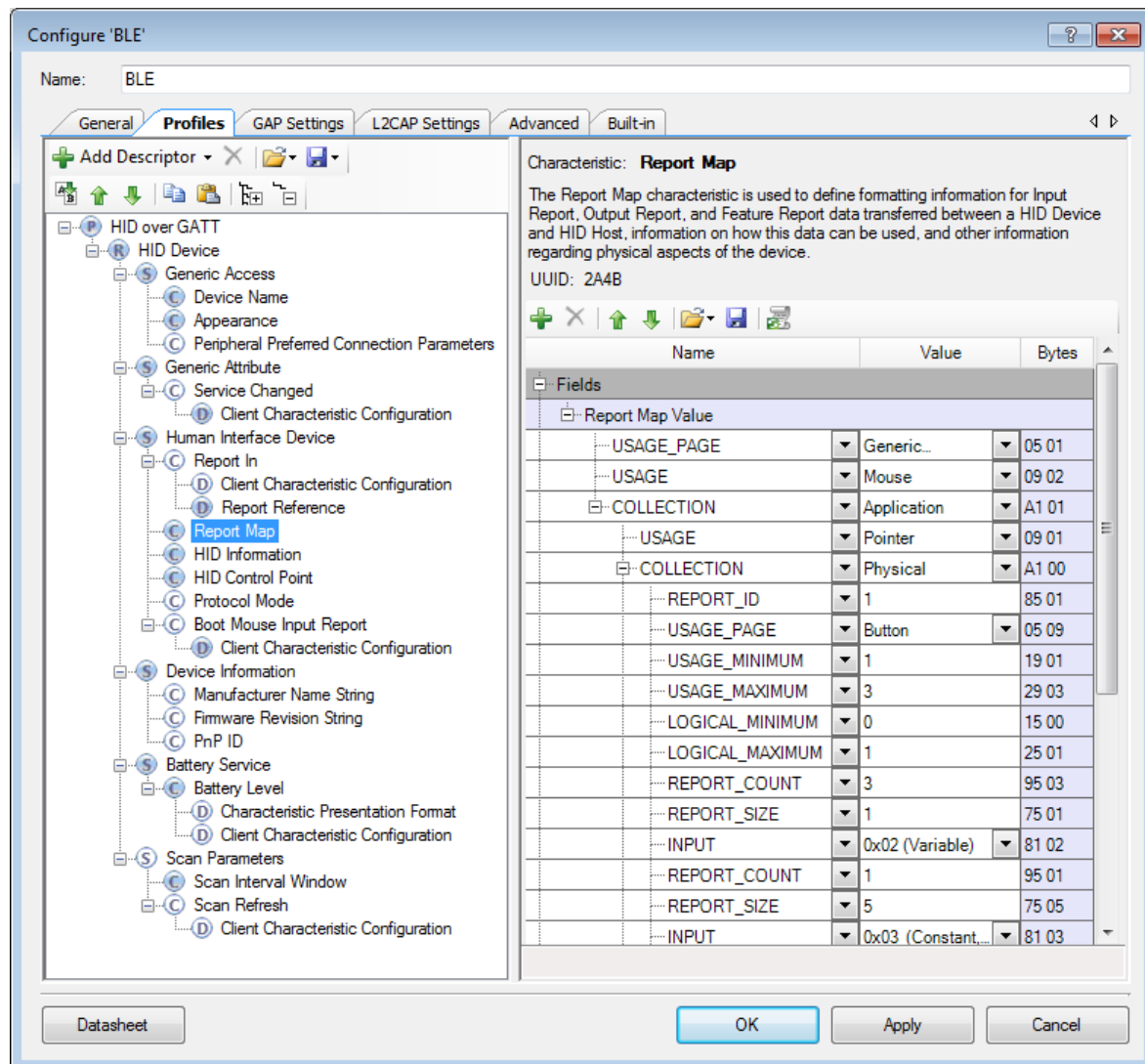


Figure 2. GATT settings

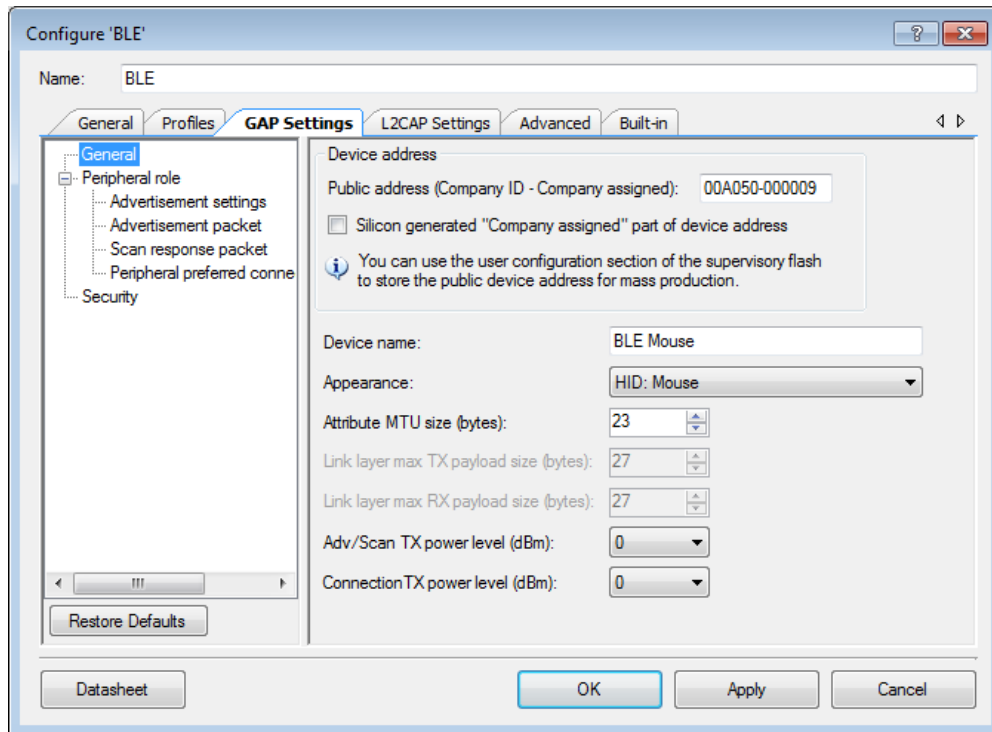


Figure 3. GAP settings

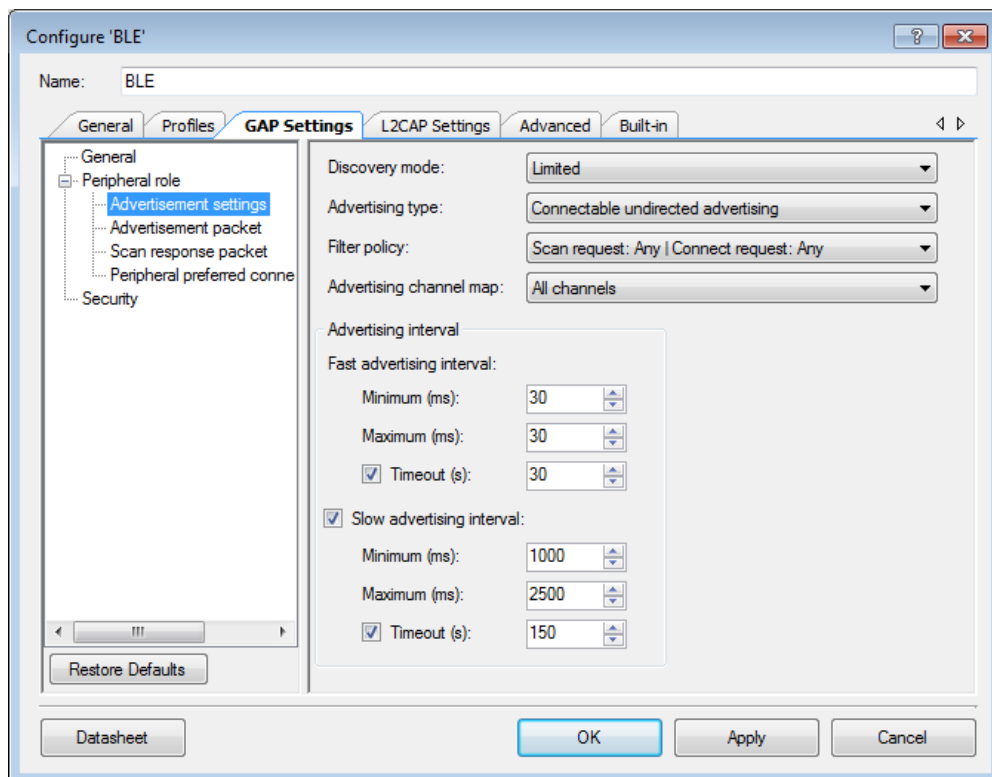


Figure 4. GAP settings -> Advertisement settings

Configure 'BLE'

Name: BLE

General Profiles **GAP Settings** L2CAP Settings Advanced Built-in

General

- Peripheral role
 - Advertisement settings
 - Advertisement packet**
 - Scan response packet
 - Peripheral preferred connection
 - Security

Advertisement data settings:

Name	Value
<input checked="" type="checkbox"/> Flags	
<input checked="" type="checkbox"/> Limited discoverable mode	
<input checked="" type="checkbox"/> BR/EDR not supported	
<input checked="" type="checkbox"/> Local Name	
Local name	Complete
<input type="checkbox"/> TX Power Level	
<input checked="" type="checkbox"/> Slave Connection Interval Range	
<input checked="" type="checkbox"/> Minimum	
Value (ms)	7.5
<input checked="" type="checkbox"/> Maximum	
Value (ms)	15
<input checked="" type="checkbox"/> Service UUID	
<input checked="" type="checkbox"/> Human Interface Device	
<input type="checkbox"/> Device Information	
<input checked="" type="checkbox"/> Battery Service	
<input type="checkbox"/> Scan Parameters	
<input type="checkbox"/> Service Solicitation	
<input type="checkbox"/> Service Data	
<input type="checkbox"/> Service Manager TK Value	
<input checked="" type="checkbox"/> Appearance	
Data	HID: Mouse
<input type="checkbox"/> Public Target Address	
<input type="checkbox"/> Random Target Address	
<input type="checkbox"/> Advertising Interval	
<input type="checkbox"/> LE Bluetooth Device Address	
<input type="checkbox"/> LE Role	
<input type="checkbox"/> Manufacturer Specific Data	

Advertisement packet:

Description	Value	Index
AD Data 1: <<Flags>>		
Length	0x02	[0]
<<Flags>>	0x01	[1]
BR/EDR not supported Limited discoverable mode	0x05	[2]
AD Data 2: <<Local Name>>		
Length	0x0A	[3]
<<Local Name>>	0x09	[4]
'B'	0x42	[5]
'L'	0x4C	[6]
'E'	0x45	[7]
' '	0x20	[8]
'M'	0x4D	[9]
'o'	0x6F	[10]
'u'	0x75	[11]
's'	0x73	[12]
'e'	0x65	[13]
AD Data 3: <<Slave Connection Interval Range>>		
Length	0x05	[14]
<<Slave Connection Interval Range>>	0x12	[15]
Minimum : 7.5 ms		
[0]	0x06	[16]
[1]	0x00	[17]
Maximum : 15 ms		
[0]	0x0C	[18]
[1]	0x00	[19]
AD Data 4: <<More 16-bit UUIDs available>>		
Length	0x05	[20]
<<More 16-bit UUIDs available>>	0x02	[21]

Restore Defaults

Datasheet

OK Apply Cancel

Figure 5. GAP settings -> Advertisement packet

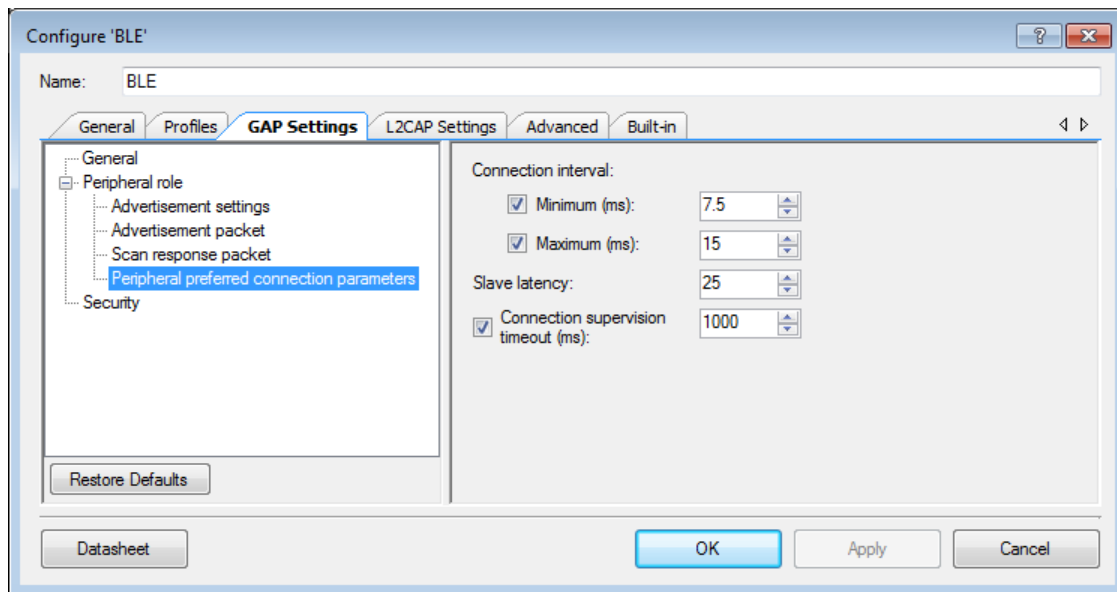


Figure 6. GAP Settings -> Peripheral preferred connection parameters

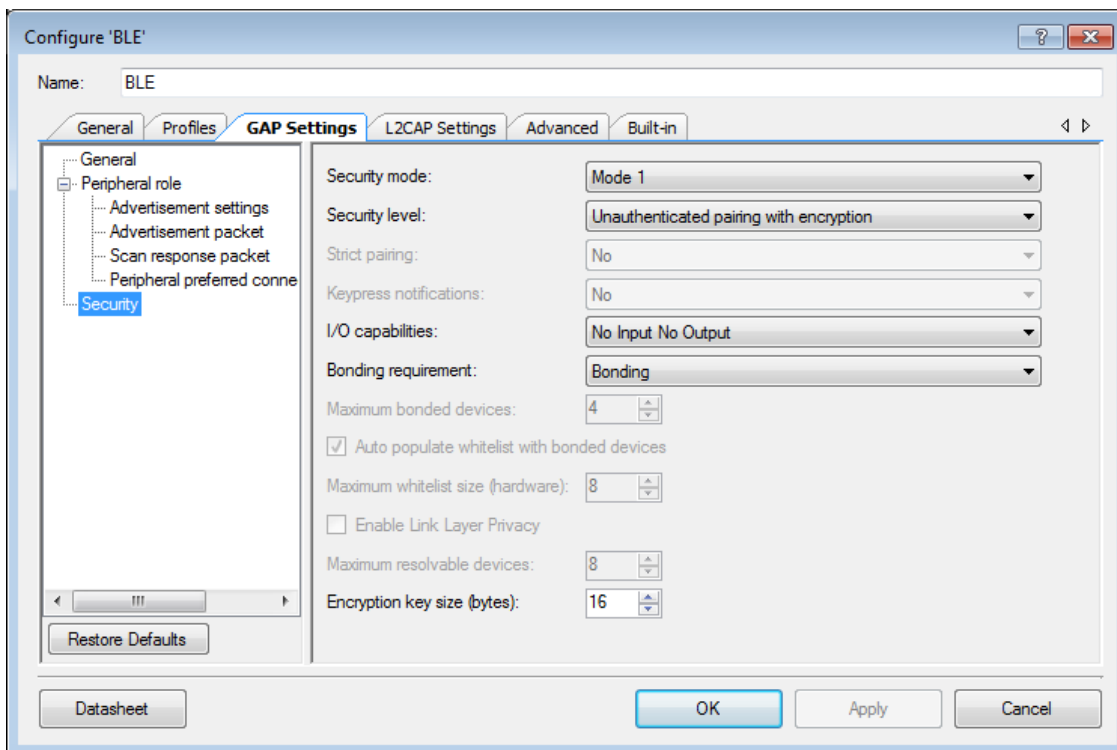


Figure 7. Security settings

Project Description

The project demonstrates the core functionality of the BLE component configured as a HID Device.

Right after the device is started it performs the BLE component initialization as well as initialization of the UART components. Four callback functions are required in this project for BLE operation. One callback function (AppCallBack()) is required to receive generic events from BLE Stack and the others (HidsCallBack(), BasCallBack(), ScpsCallBack()) are required for receiving events from the services. The component has also buried a call to CyBle_GappStartAdvertisement() on an execution of which the device will start advertising with the advertisement packet shown in **Figure 5**. On advertisement timeout, the system enters into the hibernate mode. Press the mechanical button on CY8CKIT-042 BLE (SW2) to wake up the system and start re-advertising. BLE subsystem and CPU enters into low power Deep-Sleep mode between connection and advertising intervals. BLE subsystem automatically wakes up to maintain connection and advertising data transfer.

To indicate that the device is advertising, the green LED is blinking. The red LED is turned on after disconnection to indicate that no Client is connected to the device. When a Client has connected successfully, both red and green LEDs are turned off. When a Client enables notifications, blinking of the blue LED indicates a simulated data transfer from the HID device to the Host.

You can connect the HID Device to Windows 8. Windows 7 and older OS don't have HOGP drivers. Make sure that a PC with Windows 8 has Bluetooth 4.0 installed. To connect to the HID Device, click on "Add a device" in the Devices and Printers window of the Control Panel. Select "BLE Mouse" device and click the Next button. Setup will automatically install the necessary files in the system and the mouse pointer will be moving in a square clockwise direction.

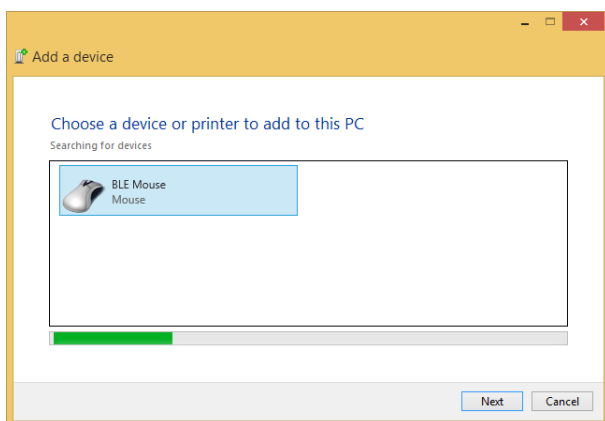


Figure 8. Windows 8 PC connecting to BLE Mouse

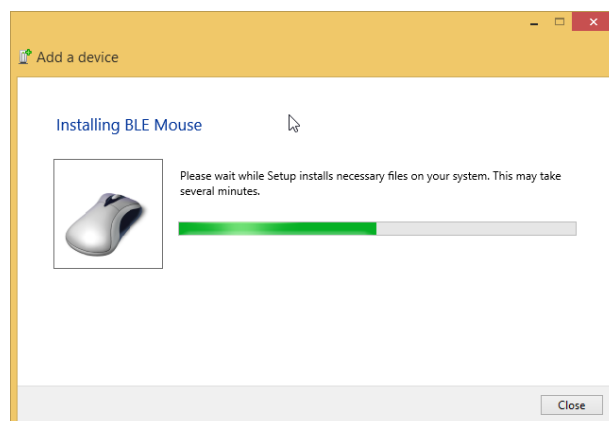


Figure 9. When device drivers are installed, mouse cursor starts moving

Also, you can connect the HID Device to the Android device with Bluetooth 4.0 support. To do this, go to your phone's Bluetooth settings and pair it with CY8CKIT-042 BLE device (it should be recognized as BLE Mouse).

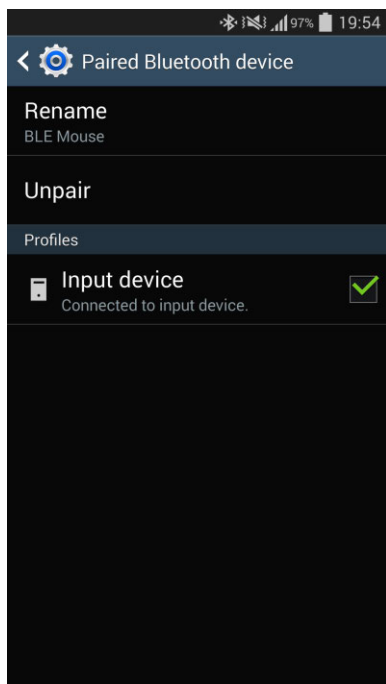


Figure 10. Android device recognizes BLE Mouse as input device

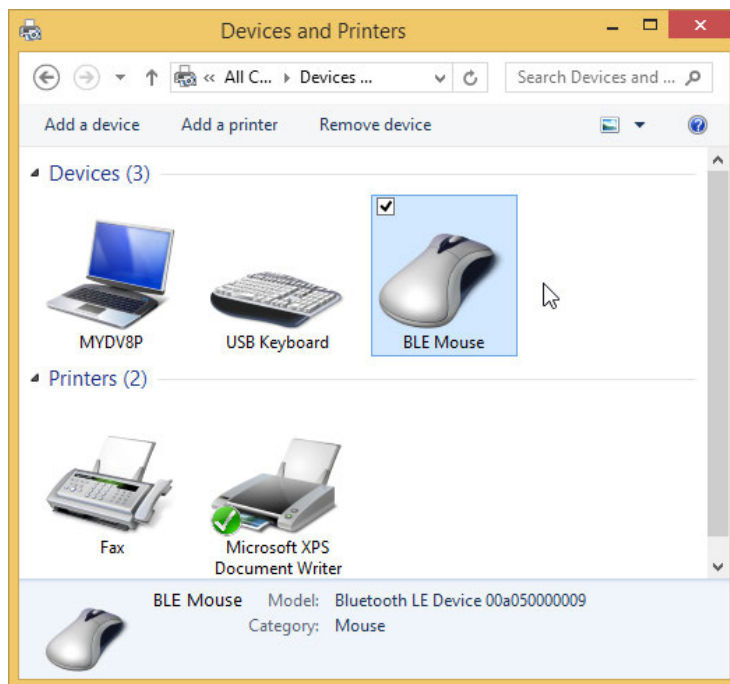


Figure 11. Windows 8 PC recognizes BLE mouse as input device

Additionally, this project implements Battery Level Service. By default, the battery level is simulated and is changed from 2 to 20 percent. To enable battery level measurement, set `BAS_MEASURE_ENABLE` to 1, `BAS_SIMULATE_ENABLE` to 0 and connect J2 pin P3[0] to J3 pin VREF. For instructions on how to use this service, refer to BLE_Battery_Level example project datasheet.

Expected Results

After pairing with peer device (Windows 8 PC, Android phone or tablet), the mouse pointer will be moving in a square clockwise direction.

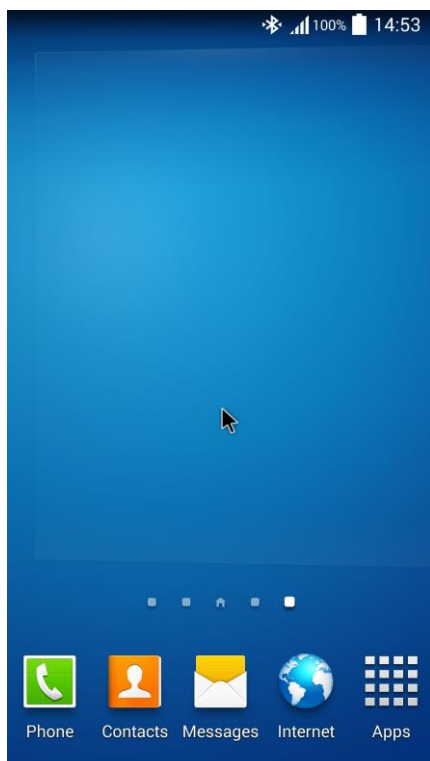


Figure 12. Initially, mouse cursor is placed at the center of the screen

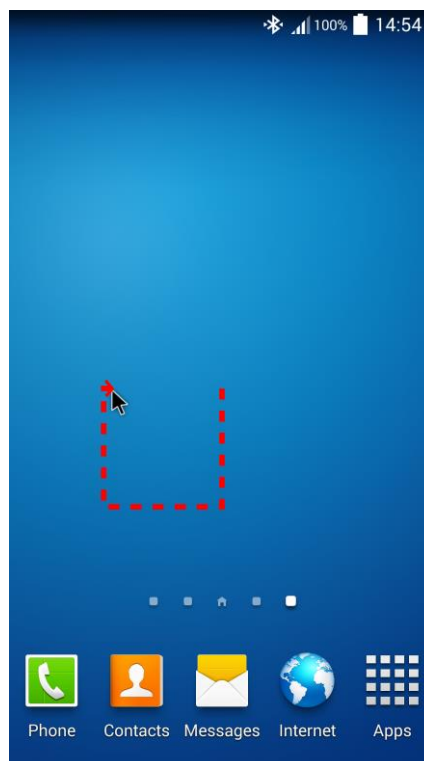


Figure 13. Mouse cursor is moving in a square clockwise direction

© Cypress Semiconductor Corporation, 2009-2015. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC® is a registered trademark, and PSoC Creator™ and Programmable System-on-Chip™ are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.