/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Taylor Sullivan

Hands Free Computer Mouse Interface

Summer 2018

This is the code for the Feather Bluefruit 32U4

Based off the Example Code described below:

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>

#include "Adafruit\_BLE.h"

#include "Adafruit\_BluefruitLE\_SPI.h"

#include "Adafruit\_BluefruitLE\_UART.h"

#include "BluefruitConfig.h"

#if SOFTWARE\_SERIAL\_AVAILABLE

#include <SoftwareSerial.h>

#endif

/\*=========================================================================

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

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"

const int upButton = 2;

const int downButton = 3;

const int leftButton = 5;

const int rightButton = 6;

int range = 12; // output range of X or Y movement; affects movement speed

int responseDelay = 10;

/\*=========================================================================\*/

// 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)

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int dwellRightClickCounter = 0;

void setup(void)

{

pinMode(upButton, INPUT);

pinMode(downButton, INPUT);

pinMode(leftButton, INPUT);

pinMode(rightButton, INPUT);

digitalWrite(upButton, HIGH); // Pull button select pin high

digitalWrite(downButton, HIGH); // Pull button select pin high

digitalWrite(leftButton, HIGH); // Pull button select pin high

digitalWrite(rightButton, HIGH);

delay(1000); // short delay to let outputs settle

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 or computer's Bluetooth settings to pair your device"));

Serial.println(F("then open an application that accepts mouse input"));

Serial.println();

Serial.println();

dwellRightClickCounter = 1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*!

@brief Constantly poll for new command or response data

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void loop(void)

{

int upState = digitalRead(upButton);

int downState = digitalRead(downButton);

int rightState = digitalRead(rightButton);

int leftState = digitalRead(leftButton);

int xDistance = (leftState - rightState) \* range;

int yDistance = (upState - downState) \* range;

//Dwell Clicking

dwellRightClickCounter++;

//Serial.print("Dwell counter: ");

//Serial.println(dwellRightClickCounter);

if(dwellRightClickCounter == 600){ // if enought time elapsed, click mouse

ble.sendCommandCheckOK(F("AT+BleHidMouseButton=L,press"));

ble.sendCommandCheckOK(F("AT+BleHidMouseButton=0"));

Serial.println("click");

///Left Double-click code

delay(20);

ble.sendCommandCheckOK(F("AT+BleHidMouseButton=L,press"));

ble.sendCommandCheckOK(F("AT+BleHidMouseButton=0"));

Serial.println("click");

}

if ((xDistance != 0) || (yDistance != 0)) {

dwellRightClickCounter = 0;

ble.print(F("AT+BleHidMouseMove=")); ///move mouse code series

ble.print(xDistance);

ble.print(",");

ble.println(yDistance);

if( ble.waitForOK() )

{

Serial.println( F("OK!") );

}else

{

Serial.println( F("FAILED!") );

}

}

delay(responseDelay);

}

/\*

// Must include this Config File to run.

// name it BluefruitConfig.h

// COMMON SETTINGS

// ----------------------------------------------------------------------------------------------

// These settings are used in both SW UART, HW UART and SPI mode

// ----------------------------------------------------------------------------------------------

#define BUFSIZE 128 // Size of the read buffer for incoming data

#define VERBOSE\_MODE true // If set to 'true' enables debug output

// SOFTWARE UART SETTINGS

// ----------------------------------------------------------------------------------------------

// The following macros declare the pins that will be used for 'SW' serial.

// You should use this option if you are connecting the UART Friend to an UNO

// ----------------------------------------------------------------------------------------------

#define BLUEFRUIT\_SWUART\_RXD\_PIN 9 // Required for software serial!

#define BLUEFRUIT\_SWUART\_TXD\_PIN 10 // Required for software serial!

#define BLUEFRUIT\_UART\_CTS\_PIN 11 // Required for software serial!

#define BLUEFRUIT\_UART\_RTS\_PIN -1 // Optional, set to -1 if unused

// HARDWARE UART SETTINGS

// ----------------------------------------------------------------------------------------------

// The following macros declare the HW serial port you are using. Uncomment

// this line if you are connecting the BLE to Leonardo/Micro or Flora

// ----------------------------------------------------------------------------------------------

#ifdef Serial1 // this makes it not complain on compilation if there's no Serial1

#define BLUEFRUIT\_HWSERIAL\_NAME Serial1

#endif

// SHARED UART SETTINGS

// ----------------------------------------------------------------------------------------------

// The following sets the optional Mode pin, its recommended but not required

// ----------------------------------------------------------------------------------------------

#define BLUEFRUIT\_UART\_MODE\_PIN 12 // Set to -1 if unused

// HARDWARE SPI SETTINGS

// ----------------------------------------------------------------------------------------------

// The following macros declare the pins to use for HW SPI communication.

// SCK, MISO and MOSI should be connected to the HW SPI pins on the Uno, etc.

// This should be used with nRF51822 based Bluefruit LE modules that use SPI.

// ----------------------------------------------------------------------------------------------

#define BLUEFRUIT\_SPI\_CS 8

#define BLUEFRUIT\_SPI\_IRQ 7

#define BLUEFRUIT\_SPI\_RST 4 // Optional but recommended, set to -1 if unused

\*/