Using Guitar Hero Controller (Wii) on BeagleBone

Wiiuse

Wiiuse allows use of wiimotes and numerous wiimote extensions (nunchuck, guitar, wheel, etc.) for use in your C projects. It handles bluetooth connections behind the scenes so long as you have the correct dependencies installed.

Parts List

(1x) Wii Remote

(1x) Guitar from *Guitar Hero* 3 (The guitar from World Tour is not supported by Wiiuse)

(1x) Bluetooth adapter

- We found this bluetooth adapter works seamlessly with the BBG:
  - https://www.amazon.ca/dp/B00DJ83070?psc=1&ref=ppx_yo2ov_dt_b_product_details

Installation

Host

Package Installation:

```
(host)$ sudo-apt get install libbluetooth-dev // needed by Wiiuse
(host)$ sudo-apt get install cmake // needed to compile Wiiuse manually
```

Wiiuse Library Installation:

- Download the latest release (.zip) from here
- Unzip the folder
  - Store the `wiiuse-x.xx.x` folder anywhere on the host
  - Copy the `wiiuse-x.xx.x` folder from the host to the shared NFS (to install on the BeagleBone later)
- Install on the host
  - (In-depth instructions found in the repo)

```
// Change directories to the wiiuse folder from the .zip
(host)$ cd wiiuse-x.xx.x
(host)$ mkdir build
(host)$ cd build
(host)$ cmake ..

// After cmake is finished
(host)$ make
```

```
// After make is finished
(host)$ make install
```

Create a directory in your shared NFS to store `.so` files from the BeagleBone

```
(host)$ mkdir ~/cmpt433/public/BeagleBone_Libs
(host)$ chmod a+rw ~/cmpt433/public/BeagleBone_Libs
```

BeagleBone

Package Installation:

```
(bbg)$ sudo-apt get install libbluetooth-dev // needed by Wiiuse
(bbg)$ sudo-apt get install cmake // needed to compile Wiiuse manually
(bbg)$ sudo-apt get install bluez // needed to detect bluetooth devices
```

Wiiuse Library Installaton:

- Copy the `wiiuse-x.xx.x` folder from the NFS to the BeagleBone directly

```
// In your shared NFS, copy the wiiuse folder to the Beaglebone (/mnt/remote/...)
(bbg)$ cp -R wiiuse-x.xx.x /home/debian/
(bbg)$ cd /home/debian/

// Follow the previously mentioned installation steps from the host
(bbg)$ cd wiiuse-x.xx.x
(bbg)$ mkdir build
(bbg)$ cd build
(bbg)$ cmake ..

// After cmake is finished
(bbg)$ make

// After make is finished
(bbg)$ make install

// .so files should be placed in your /usr/local/lib/ folder
```

```
// Copy .so files onto the shared NFS folder
(bbg)$ cp /usr/local/lib/libwiiuse.so /mnt/remote/BeagleBone_Libs
(bbg)$ cp /usr/lib/arm-linux-gnueabihf/libbluetooth.so /mnt/remote/BeagleBone_Libs
```

Include in C Project

```
// On your C file:
#include <wiiuse.h>

// On your makefile:
LFLAGS = -L$(HOME)/cmpt433/public/BeagleBone_Libs

app:
  $(CC_C) $(CFLAGS) your_program.c -o $(OUTDIR)/$(TARGET) $(LFLAGS) -lwiiuse -lbluetooth
```

Connecting your Wii Remote & Button Mapping

Wiiuse library usage (basic code adapted from wiiuse Example by Michael Laforest)

Create and initialize Wiimote structs

```
int maxWiimotes = 4;
int connectionWaitTime = 5;
wiimote** wiimotes;
wiimotes = wiiuse_init(maxWiimotes);
int wiimotesFound = wiiuse_find(wiimotes, maxWiimotes, connectionWaitTime);
int wiimotesConnected = wiiuse_connect(wiimotes, maxWiimotes);
```

- Press sync (under the battery cover) or 1+2 on the wiimote to pair.

Enable LEDs and disable motion/IR sensing

```
wiiuse_set_leds(wiimotes[0], WIIMOTE_LED_1);
wiiuse_motion_sensing(wiimotes[0], 0);
wiiuse_set_ir(wiimotes[0], 0);
```

```
if(WIIMOTE_IS_CONNECTED(wm[i]))
```

Poll Wiimotes

```
if(wiiuse_poll(wiimotes, maxWiimotes) != 0 ){
  //handle events
}
```

- wiiuse_poll returns the number of wiimotes which had an event occur since the last call. In addition, each wiimote with an event sets wiimote→event to represent the type of event which occurred. All input is type WIIUSE_EVENT.

Handle Events

```
if (wiimote->exp.type == EXP_GUITAR_HERO_3) {
    struct guitar_hero_3_t* gh3 = (guitar_hero_3_t*)&wm->exp.gh3;

    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_STRUM_UP)) {
      printf("Guitar: Strum Up pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_STRUM_DOWN)) {
      printf("Guitar: Strum Down pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_GREEN)) {
      printf("Guitar: Green pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_RED)) {
      printf("Guitar: Red pressed\n")
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_YELLOW)) {
      printf("Guitar: Yellow pressed\n")
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_BLUE)) {
      printf("Guitar: Blue pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_ORANGE)) {
```

```
      printf("Guitar: Orange pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_PLUS)) {
      printf("Guitar: Plus pressed\n");
    }
    if (IS_PRESSED(gh3, GUITAR_HERO_3_BUTTON_MINUS)) {
      printf("Guitar: Minus pressed\n");
    }

    printf("Guitar whammy bar:          %f\n", gh3->whammy_bar);
    printf("Guitar joystick angle:      %f\n", gh3->js.ang);
    printf("Guitar joystick magnitude:  %f\n", gh3->js.mag);
  }
```

- IS_PRESSED detects if a button is pressed or held.
- IS_JUST_PRESSED detects if a button was pressed in this specific event.
- IS_HELD detects if the button is held down, and will NOT detect the button being pressed in the current event.

Troubleshooting

Error while loading shared libraries: libwiiuse.so: cannot open shared object file: No such file or directory

If you get this error when running your project, double check the libraries that are linked to your program.

```
(bbg)$ ldd ./your_program_name

// Output
linux-vdso.so.1 (0xbeee7000)
libsmf.so.0 => /usr/lib/arm-linux-gnueabihf/libsmf.so.0 (0xb6ef3000)
libwiiuse.so => not found
libpthread.so.0 => /lib/arm-linux-gnueabihf/libpthread.so.0 (0xb6ecd000)
libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6dce000)
libglib-2.0.so.0 => /usr/lib/arm-linux-gnueabihf/libglib-2.0.so.0 (0xb6cdf000)
libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6 (0xb6c77000)
/lib/ld-linux-armhf.so.3 (0xb6f23000)
libpcre.so.3 => /lib/arm-linux-gnueabihf/libpcre.so.3 (0xb6c19000)
```

Solution:

- Link the `libwiiuse.so` found in `/mnt/remote/BeagleBone_Libs` permanently on the BeagleBone

```
(bbg)$ echo "/mnt/remote/BeagleBone_Libs" | sudo tee /etc/ld.so.conf
(bbg)$ sudo ldconfg
```

- Check your program again to see if it was successfully linked

```
(bbg)$ ldd ./your_program_name

  linux-vdso.so.1 (0xbed70000)
  libsmf.so.0 => /mnt/remote/BeagleHero_Libs/libsmf.so.0 (0xb6f2b000)
  libwiiuse.so => /mnt/remote/BeagleHero_Libs/libwiiuse.so (0xb6f10000)
  libpthread.so.0 => /lib/arm-linux-gnueabihf/libpthread.so.0 (0xb6eea000)
  libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6deb000)
  libglib-2.0.so.0 => /mnt/remote/BeagleHero_Libs/libglib-2.0.so.0 (0xb6cfc000)
  libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6 (0xb6c94000)
  /lib/ld-linux-armhf.so.3 (0xb6f5b000)
  libbluetooth.so.3 => /mnt/remote/BeagleHero_Libs/libbluetooth.so.3 (0xb6c5f000)
  libpcre.so.3 => /mnt/remote/BeagleHero_Libs/libpcre.so.3 (0xb6c01000)
```

References

- https://github.com/wiiuse/wiiuse
- https://www.baeldung.com/linux/solve-shared-object-error (Troubleshooting section)

Authors

By: Denzel Tjokroardi, Para Upton, Joseph Nevado, Daniel Olivera