

LABORATORY 2(A):  
INTERFACING WIIMOTE

---

**Authors:**

Jeff C. Jensen (National Instruments)  
Trung N. Tran (National Instruments)

**Instructors:**

Edward A. Lee  
Sanjit A. Seshia

University of California, Berkeley  
EECS 149

February 2, 2011

# 1 Introduction

The Nintendo WiiMote is a handheld wireless game controller that measures acceleration and light intensity to determine motion and position, as well as momentary depressions of user buttons.

## 1.1 Goals

In this lab, you will:

1. Interface with the Nintendo WiiMote.
2. Send binary commands to the WiiMote to poll its accelerometer and set LED and rumble status.
3. Calibrate the accelerometer and display the results.

This lab should take 1 hour to complete.

## 1.2 Required Reading

Read the following material *in advance of your laboratory session*:

1. *Lee & Seshia* [1]
  - (a) §7.2.4 “Fixed-Point Numbers,” pp. 197-199.
2. *Technical Guide to the Wii Remote* (unofficial) [2]

## 1.3 Equipment

1. PC computer running Windows XP.
2. Nintendo WiiMote.
3. Bluetooth radio with standard Microsoft Bluetooth stack.
4. National Instruments LabVIEW 2010.

# 2 Equipment Introduction

The Nintendo WiiMote (Fig. 1) is part of the Nintendo Wii entertainment system, is a handheld wireless game controller that measures acceleration and light intensity to determine motion and position, as well as momentary depressions of user buttons. The three types of actuators on the Wii Remote are a speaker, LEDs, and a rumble unit that vibrates the device. While the Wii Remote is simple, popular, and widely available, its most compelling feature is that can serve as a conceptual bridge from computer-based to embedded-based measurement.

An onboard embedded processor (Fig. 2 on the remote contains an analog-to-digital converter (ADC) to sample the accelerometer and a radio to broadcast Bluetooth messages to a host. An online community of embedded systems hackers reverse-engineered and documented these messages [2]. Take note that measurements are not read as floating-point voltages, but instead as binary values output by the ADC on the remote.



Figure 1: Nintendo WiiMote

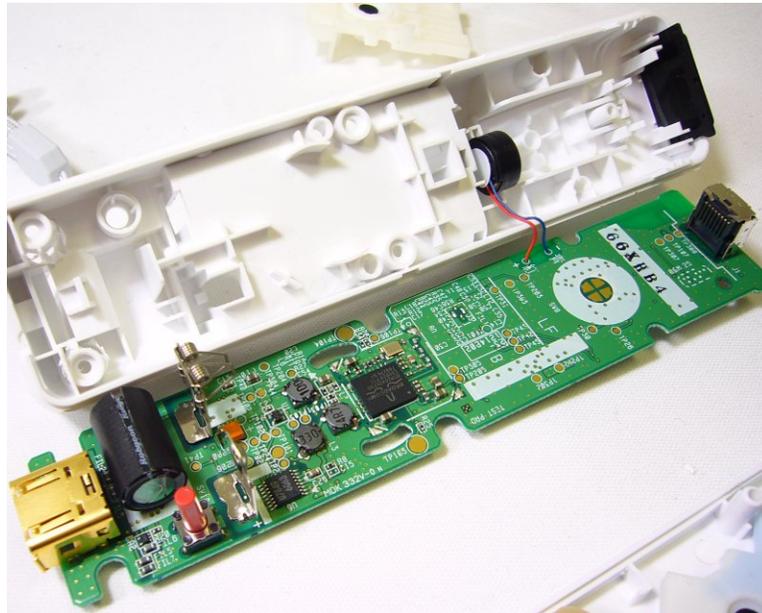


Figure 2: Nintendo WiiMote microcontroller exposed.

## **3 Procedure**

### **3.1 Load the Template Project**

Download the **lab2afiles.zip** from the course website to your computer, and extract it to the folder **U:\EECS149**. Open the LabVIEW project file (.lvproj) that has been extracted to this folder.

### **3.2 Configure Equipment**

From your project window, open **WiiMote Pair.vi**. This is a VI that will automatically pair your desktop computer to your WiiMote. You do not need to modify or view the block diagram.

When pairing the WiiMote, you must hold down both the 1 and 2 user buttons for the entire pairing process. Do not let go until the pairing VI has completed. Key in the MAC address of your WiiMote (written on the side), and run the VI to pair.

### **3.3 Review and Execute the Template Code**

From your project window, open the **Lab 02a VI**. This is the VI you will modify to complete this lab. The VI will periodically poll sensors, and allow you to transmit messages via Bluetooth.

### **3.4 Toggle User LEDs**

Modify the values for **Report ID** and **Payload** to toggle user LEDs.

### **3.5 Toggle Rumble**

Modify the values for **Report ID** and **Payload** to toggle the rumble.

### 3.6 Calibrate Accelerometer

You will see the raw ADC output from the WiiMote accelerometer. These values are for the X and Y axes of the accelerometer (Fig. 3). Modify the block diagram to display a new chart where the accelerometer values have been scaled to 1g.

Calibration parameters should be measured at runtime by pressing the appropriate record buttons when the WiiMote is at varied orientations. The X, Y 0g offset should record the value of each accelerometer axis when the WiiMote is positioned so that both axes are perpendicular to gravity (hence measure 0g). Extend the template VI to record each accelerometer axis when the WiiMote is positioned so that the axis is parallel to gravity (hence measuring +1g). Once the user has performed the calibration, your VI should display the accelerometer measurements scaled to  $[-1g, 1g]$ .

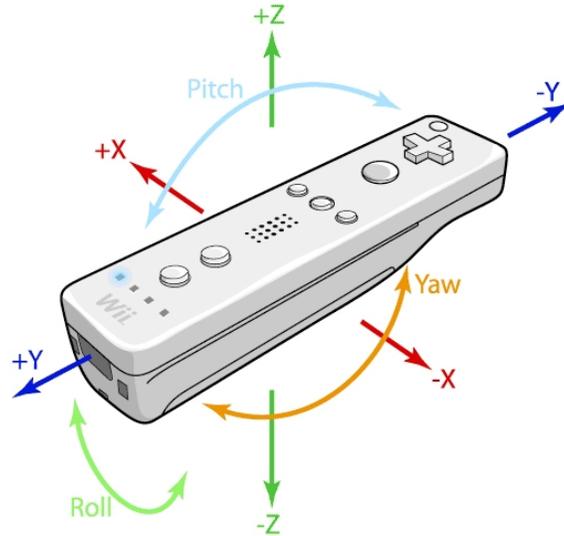


Figure 3: Nintendo WiiMote Axes

## 4 Acknowledgment

We thank the University of California, Berkeley, Department of EECS Instructional Support Group (ISG) and Electronics Support Group (ESG) for their continued efforts in supporting this lab.

## References

- [1] E.A. Lee and S.A. Seshia, *Introduction to Embedded Systems - A Cyber-Physical Systems Approach*, digital version 1.03. Berkeley, California, 2010. Available: [http://leeseshia.org/releases/LeeSeshia\\_DigitalV1\\_03.pdf](http://leeseshia.org/releases/LeeSeshia_DigitalV1_03.pdf). [Accessed January 25<sup>th</sup>, 2011].
- [2] Ian Rickard, “WiiMote” website. Available: [http://homepage.mac.com/ianrickard/wiimote/wiili\\_wimote.html](http://homepage.mac.com/ianrickard/wiimote/wiili_wimote.html). [Accessed January 30<sup>th</sup>, 2011].

## 5 Laboratory 2(a): Interfacing WiiMote Objectives and Checkout

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Lab Section: \_\_\_\_\_

1. (\_\_\_\_ / 5) Successfully pair with your WiiMote.
2. (\_\_\_\_ / 10) Change the value of the "Report ID" and "Payload" front panel controls to turn all four WiiMote LEDs on.
3. (\_\_\_\_ / 10) Change the value of the "Report ID" and "Payload" front panel controls to enable and disable the WiiMote rumble.
4. (\_\_\_\_ / 50) Using "Record" buttons, calibrate the WiiMote accelerometer to display values between  $[-1g, 1g]$ . You will need to use shift registers to save the value recorded when the "record" buttons are pressed. You can follow the design pattern shown for recording the X, Y 0g offset.  
Do not hard-code any scaling, offset, or other calibration parameters. All parameters should be recorded at runtime following user button presses.
5. (\_\_\_\_ / 25) Correctly calculate pitch and roll and display on the front panel.
6. (\_\_\_\_ / 0) BONUS (up to +5): Replace "Report ID" and "Payload" numeric controls with buttons for each of the user LEDs and for the rumble status. When the "Transmit" button is pressed, these controls should be read, and the appropriate report ID and payload should be transmitted to the WiiMote.

## 6 Lab Writeup Prompts

In your lab writeup, please address the following questions:

1. In the template VI you were provided, what was the function of the shift registers?
2. What are some advantages of dynamically calibrating a sensor, instead of simply recording calibration parameters once and hard-coding them in software?
3. What equation did you use to scale the raw accelerometer measurement to  $[-1g, 1g]$ ?
4. What did you like about this lab, and what do you think could be improved?

## 7 Laboratory 2(a): Interfacing WiiMote Prelab Exercises

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Lab Section: \_\_\_\_\_

1. What is the WiiMote Report ID and payload to turn on LED 3?

2. What is the WiiMote Report ID and payload to turn on the rumble?