Wii Nunchuk Interface

The Wii Nunchuk uses a proprietary connector with six pins. The connections are as follows.

1 3 5 2 4 6	1 = +3V (3V recommended but works at 5V)	2 = Clock
	3 = N/A	4 = N/A
	5 = Data	6 = Gnd

Figure 1: Wii Nunchuk Pinout

To communicate with the Nunchuk, we must send a handshake signal. If you are using a **black** Wii Nunchuk, send 2 bytes `0xF0, 0x55` to initialize the first register and `0xFB, 0x00` to initialize the second register of the Nunchuk. On a **white** Wii Nunchuk, send `0x40, 0x00` followed by `0x00`. The I^2C address of both Wii Nunchuks is `0x52`. The frequency used to communicate with the Wii Nunchuk is 100KHz.

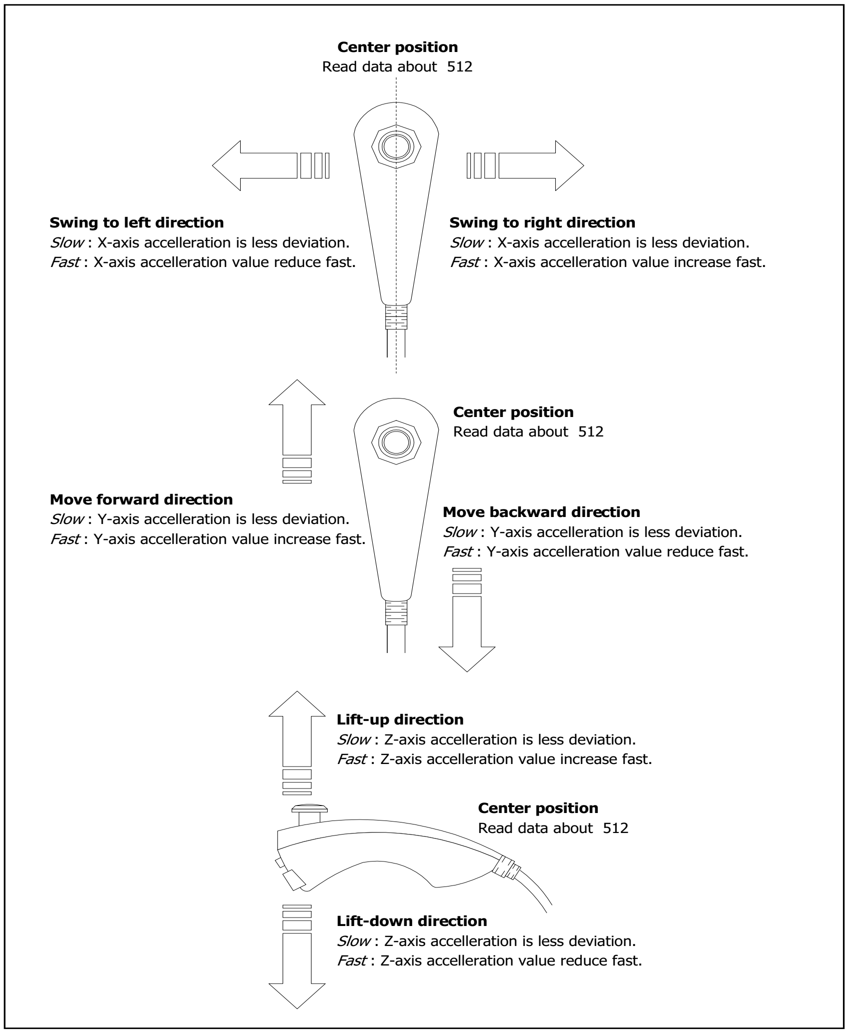

Figure 2: Wii Nunchuk Physical Operation

I^{2}C Bus: Inter-Integrated Circuits

The I^{2}C (Inter-IC) bus is a bi-directional two-wire serial bus that provides a communication link between integrated circuits (ICs). Phillips introduced the I^{2}C bus 20 years ago for mass-produced items such as televisions, VCRs, and audio equipment. Today, I^{2}C is the de-facto solution for embedded applications. It's also known as the "two-wire interface" (TWI), due to the fact that it uses just two wires: SDA (serial data) and SCL (serial clock). A typical I^{2}C configuration consists of one master device and any number of slave devices sharing the bus. The master initiates all transactions with the slaves. In this lab, the Intel Atom motherboard acts as the master device and the Wii Nunchuk is the slave device.

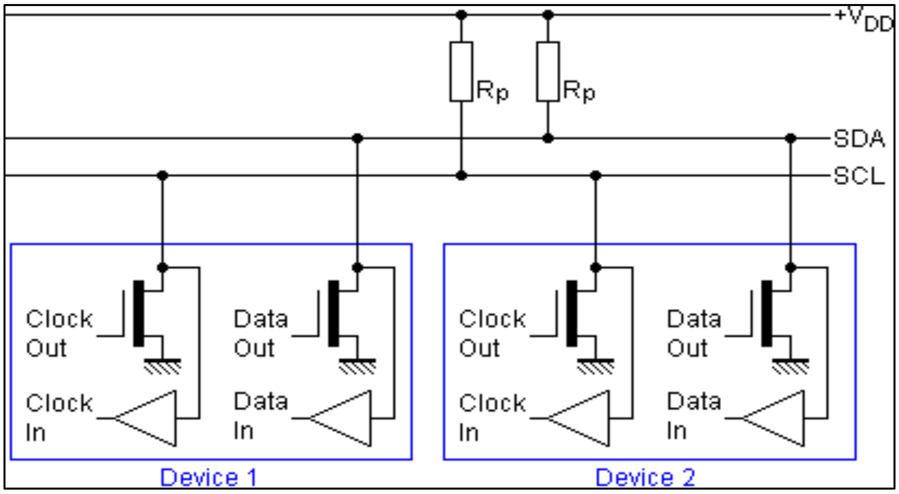

Figure 3: I²C Device Circuit

Another name you will hear used in connection with I²C is SMBus (System Management Bus). The SMBus protocol was developed by Intel in 2005 and is based on I²C. SMBus defines a stricter range of operating limits, including clock speed, timing and data formats, compared to I²C. Many I²C devices have, in turn, incorporated policies from SMBus and both kinds of devices can often be used on the same bus.

Wii Nunchuk Output

The Wii Nunchuk is a slave I^2C bus device that outputs six bytes of data as follows.

Data byte receive								Address
Joystick X								0x00
Joystick Y								0x01
Accelerometer X (bit 9 to bit 2 for 10-bit resolution)								0x02
Accelerometer Y (bit 9 to bit 2 for 10-bit resolution)								0x03
Accelerometer Z (bit 9 to bit 2 for 10-bit resolution)								0x04
Accel. Z bit 1	Accel. Z bit 0	Accel. Y bit 1	Accel. Y bit 0	Accel. X bit 1	Accel. X bit 0	C-button	Z-button	0x05

Byte 0x00 : X-axis data of the joystick

Byte 0x01 : Y-axis data of the joystick

Byte 0x02 : X-axis data of the accellerometer sensor

Byte 0x03 : Y-axis data of the accellerometer sensor

Byte 0x04 : Z-axis data of the accellerometer sensor

Byte 0x05 : bit 0 as Z button status - 0 = pressed and 1 = release

bit 1 as C button status - 0 = pressed and 1 = release

bit 2 and 3 as 2 lower bit of X-axis data of the accellerometer sensor

bit 4 and 5 as 2 lower bit of Y-axis data of the accellerometer sensor

bit 6 and 7 as 2 lower bit of Z-axis data of the accellerometer sensor

Figure 3: Wii Nunchuk I^2C Output

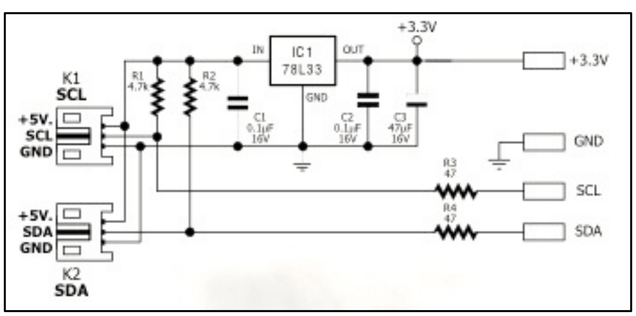

Figure 4: Nunchuk Interface Board Circuit