Remote Pong Game

Team: Fengyi Song, Mingrui Liu, Wanding Li, Junchao Zhang
Project Overview

• Remote Pong Game
  – Hardware: Image Display, Audio Effects
  – Software: Game Logic, Device Driver
  – Peripheral Controller: Wii remote controller
Project Overview
Audio Effects in Hardware

• Operate at audio clock 11.3 Mhz( PLL )
• Use I2C protocol to configure codec SSM 2603
• Store audio samples in the on chip ROM
• Two clocks derived from audio clock
  - Channel clock: Send sample on one channel at a time.
  - Bit clock: Send one bit of each sample.
Audio Block Diagram

Audio Effects

Audio ROM

data/16

address

Audio Codec SSM 2603

Samples/16

I2C Configuration

action signal

parameters

Speaker
Game UI Display
Game Over

two 1-bit logic: zone & data (1/0)
Software Process
Wii Controller

- Connected with Socket using Bluetooth
- Dongle as position sensor light
- Software preparation
  - Libwiimote (C-library)
  - Linux Device Driver: BlueZ, libwiimote-dev
  - Scale the screen size from 1784*1272 to 512*480 by using wiimote open source API
Game Logic Controller

• Game status:
  On, score, life, paddle coord, Ball coords, over

• Pthread:
  Updating paddle coords

• Check boarder:
  Check ball and brick

Game logic API

Write_ball();
Write_on();
Write_life();
Write_paddle();
Write_over();
Write_score();
Pthread_function() {
    mutex_lock;
    updating paddle coords;
    mutex_unlock;
}
Main() {
    pthread();
    while(1) {
        updating game status;
        check boarder;
    }
}
Device Driver

• ioctl calls to write positions \((x, y)\) of ball, paddle, scores, remaining lives, game_over and brick on and off.

• VGA memory: 3-bit address, 32-bit words
Experiences and Issues

• Image Display
  – Complex display logic (sample duplicating)

• Audio Effects
  – Synchronization between software and hardware

• Game Logic
  – Multithreaded and synchronization (mutex)
  – Missing libraries(Bluez)