

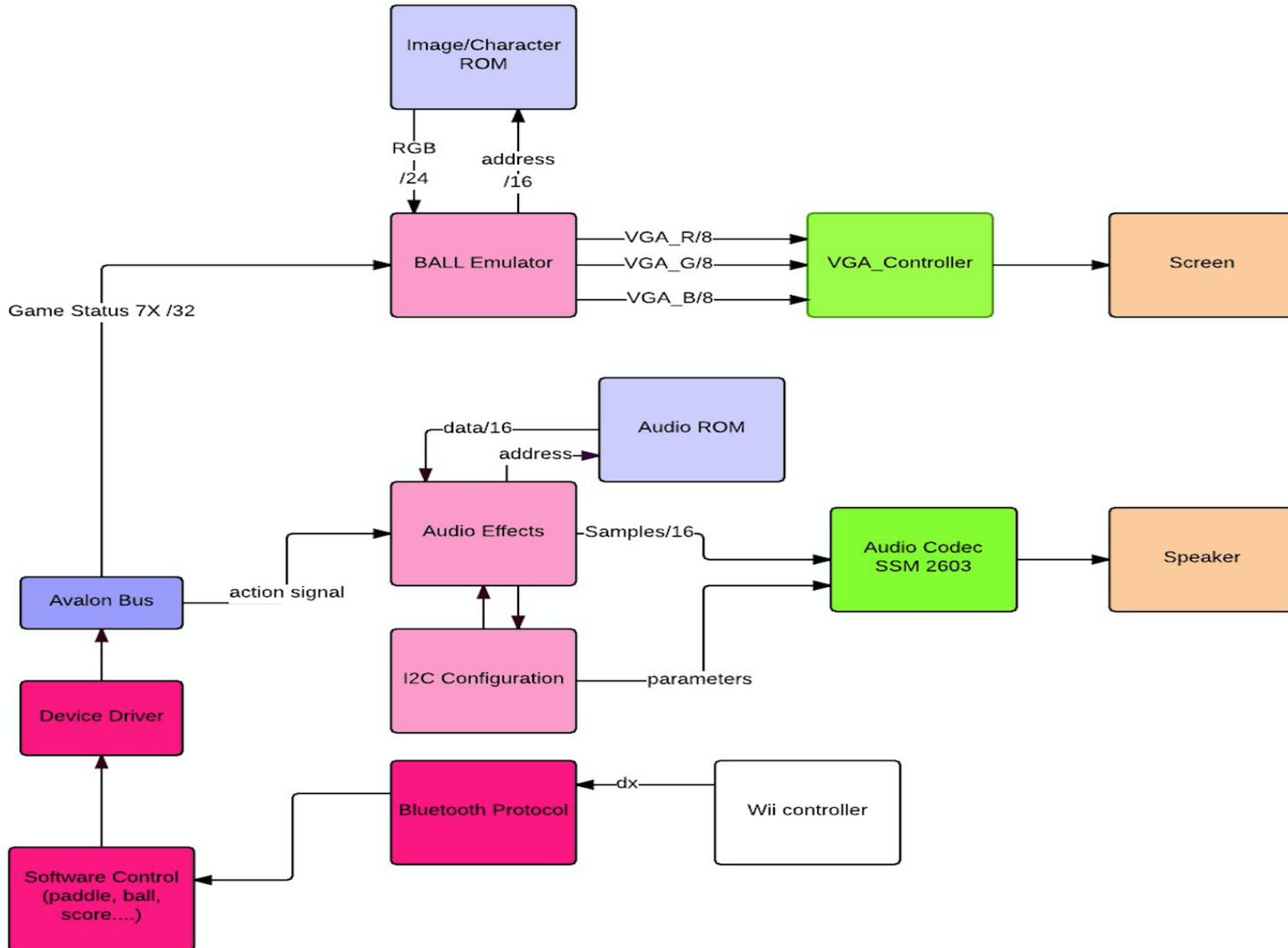
# Remote Pong Game

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



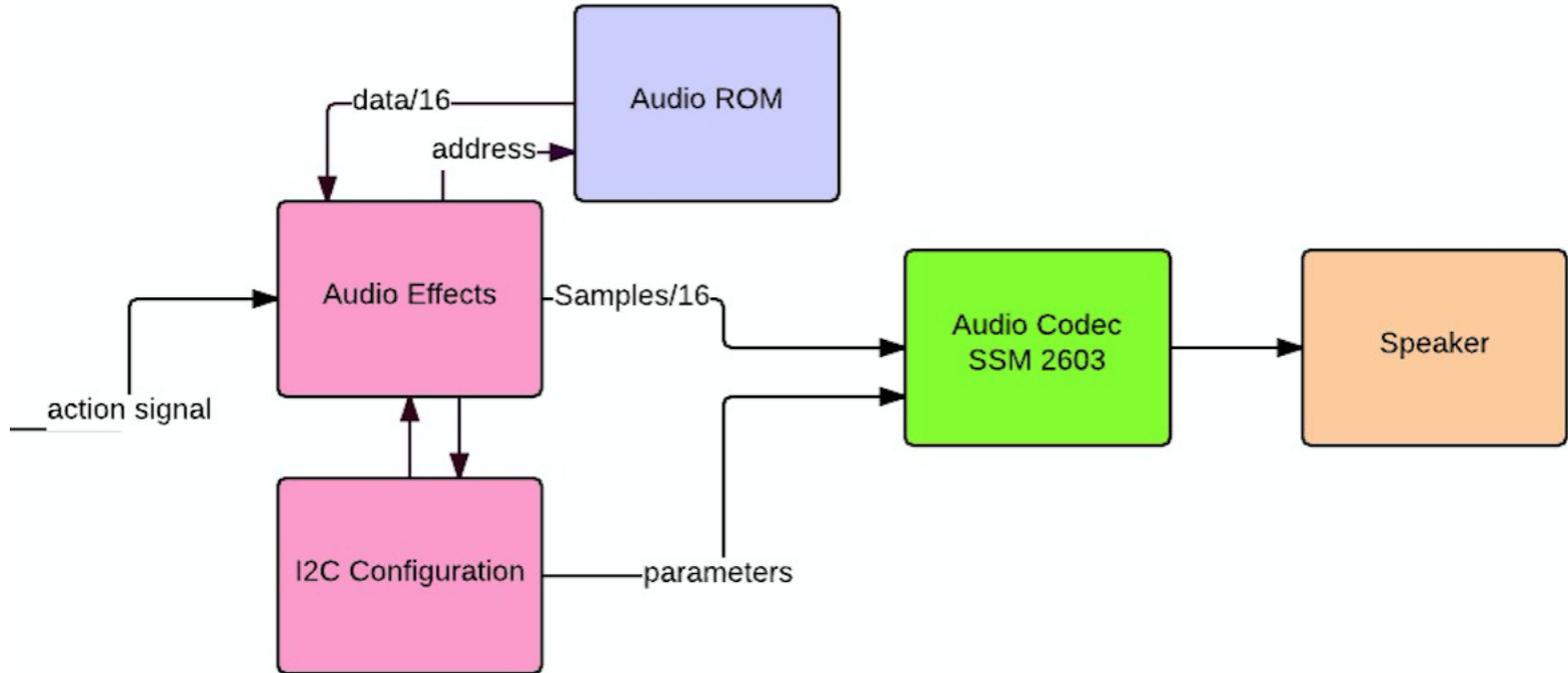
# Audio Effects in Hardware

- 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



# Game UI Display



sprite1

sprite2

sprite3

sprite4

sprite5

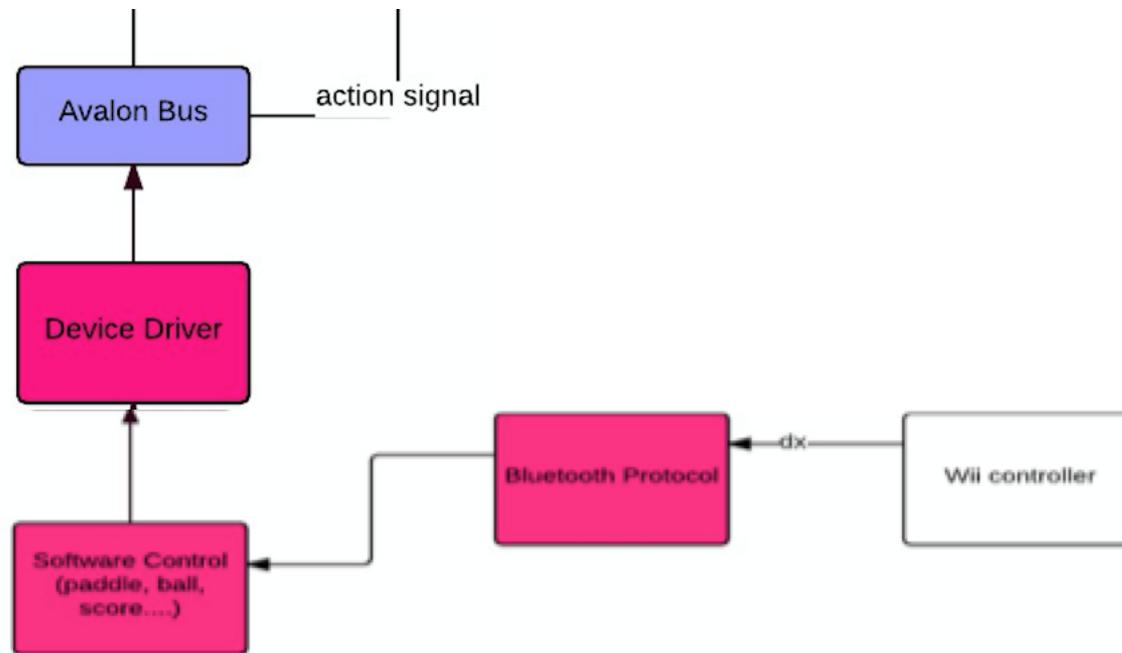
sprite6

# Game Over



two 1-bit  
logic:  
zone & data  
(1/0)

# Software Process



# Wii Controller

- Connected with Sockit 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 logic API

```
Write_ball();
Wirte_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;
    }
}
```

- Game status:

On,score,life,paddle coord,  
Ball coords,over

- Pthread:

Updating paddle coords

- Check boarder:

Check ball and brick

# 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)