

**CSEE 4840**

Spring 2012

Embedded System: Project Presentation:  
**The Awesome Guitar Game**

Avijit Singh Wasu

Laurent Charignon

{asw2156,lc2817}@columbia.edu

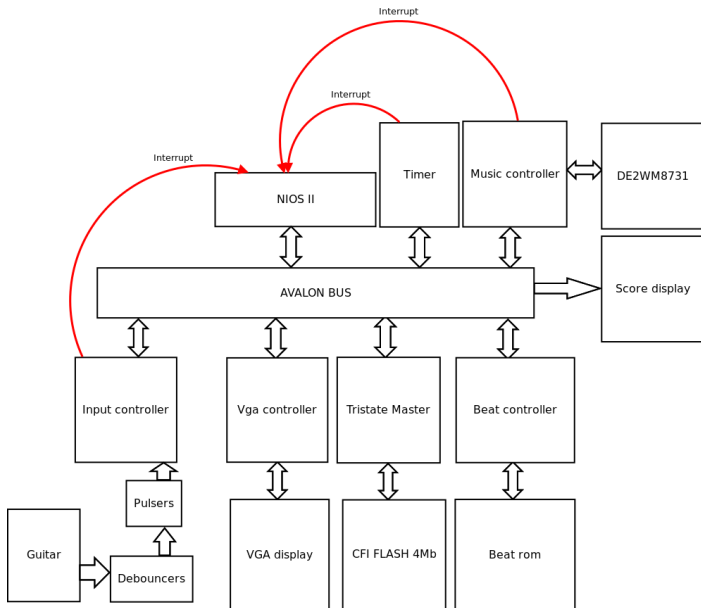
# Overview

- ▶ We created a clone of "Guitar Hero" on a FPGA
- ▶ We use with a Playstation II controller that we have customized
- ▶ We display the falling beats of the song on a VGA display
- ▶ We play the music at the same time

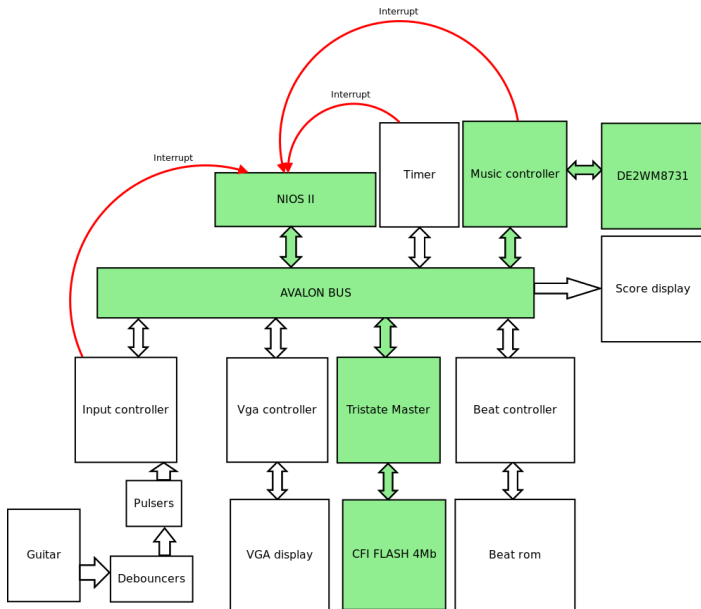
# Today's presentation

1. The overall architecture
2. A two-month adventure!
3. Lessons learned

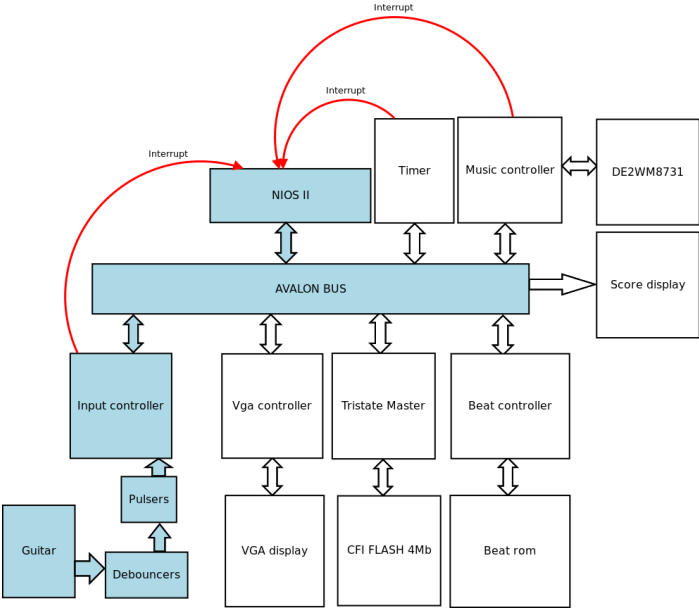
# Architecture



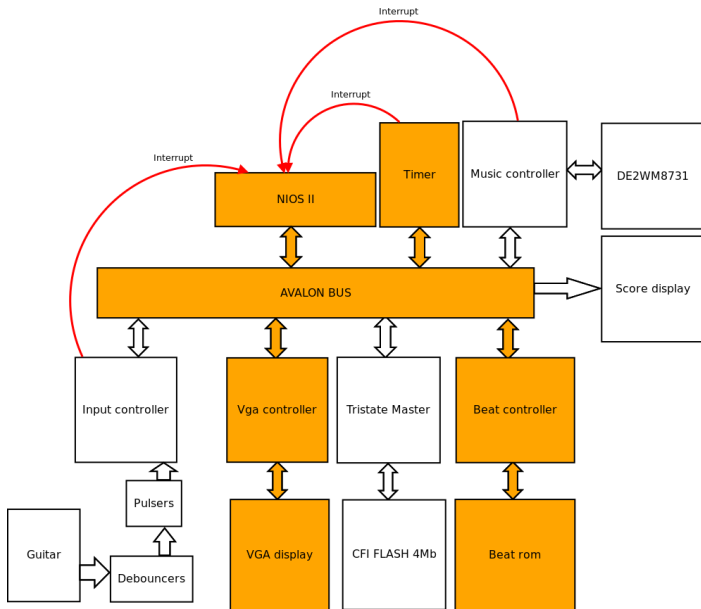
# Playing the music



# Getting the input from the guitar



# Display the falling beats at the right time!



## Software part

Everything is managed by the NIOS processor, everything goes through it. Our software part is made of:

- ▶ A set of callbacks for interruptions
- ▶ An initialization phase which triggers the interruptions and initialize values
- ▶ An infinite loop that executes various operations



## A two-month adventure

- We have well divided the project from the start, thus allowing us to
- Be on several fronts at the same time: parallelize the work was crucial
  - Reuse some components, repeat some patterns

# Timeline month 1

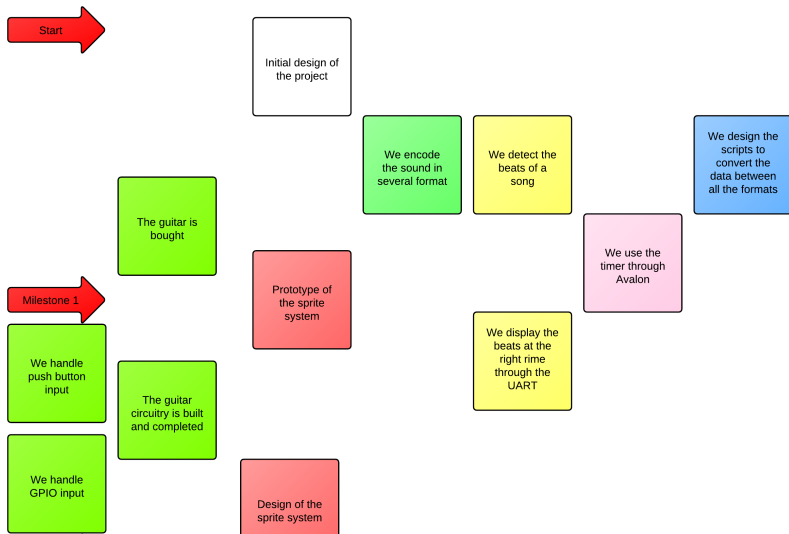


Figure: Milestone 1 and 2

# Building the guitar 1/3



Figure: Make the beats fall

## Building the guitar 2/3

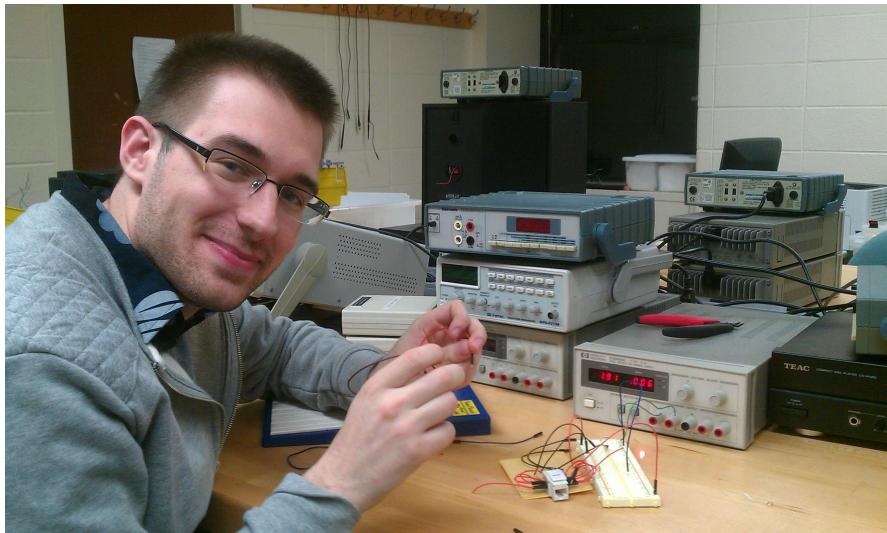


Figure: Make the beats fall

## Building the guitar 3/3

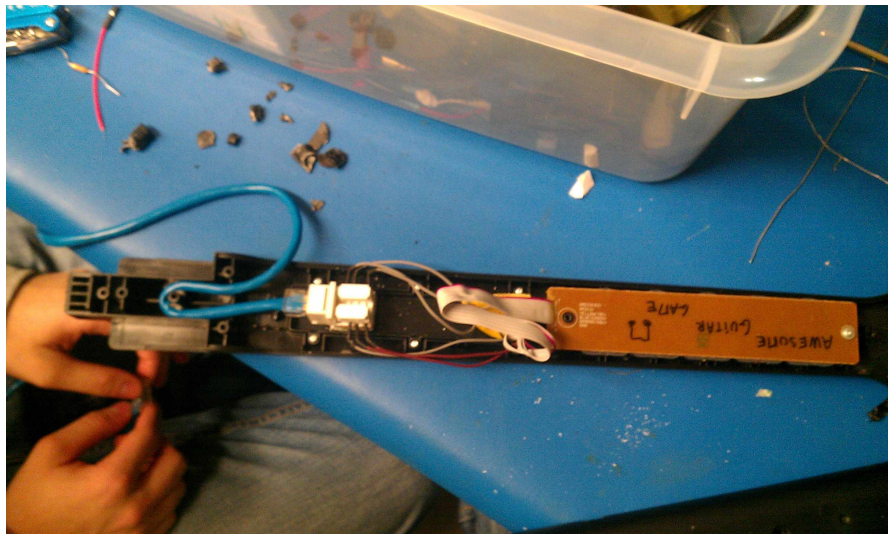


Figure: Make the beats fall

# Timeline month 1

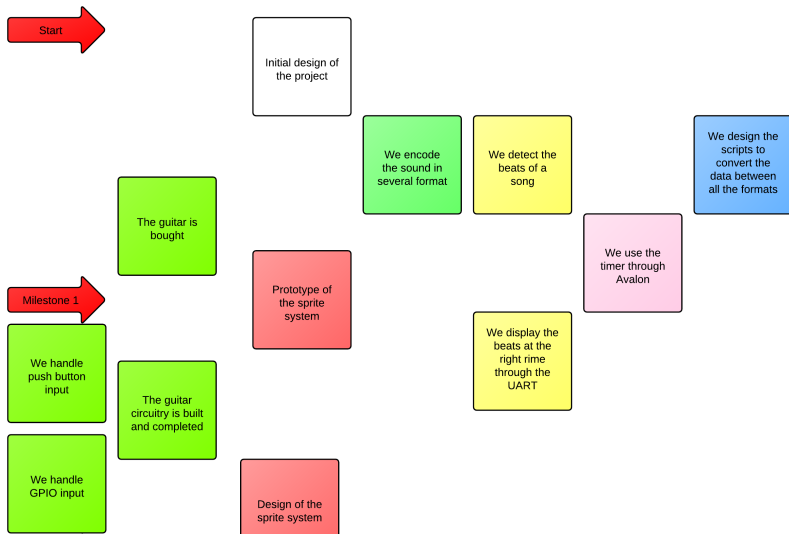


Figure: Milestone 1 and 2

# Timeline month 2

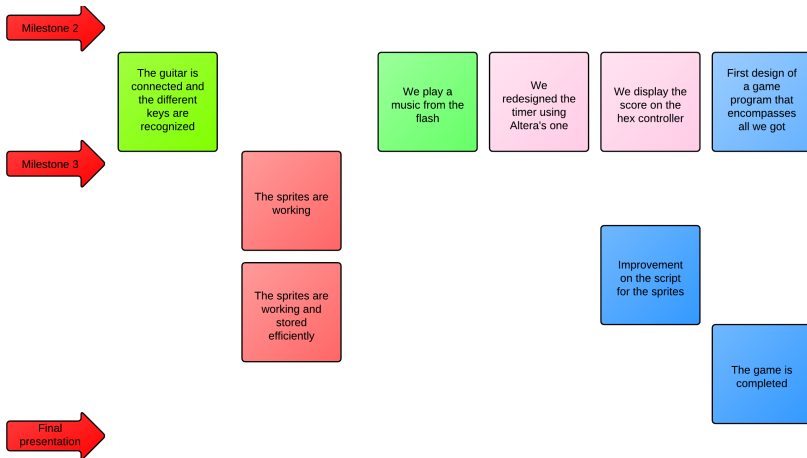
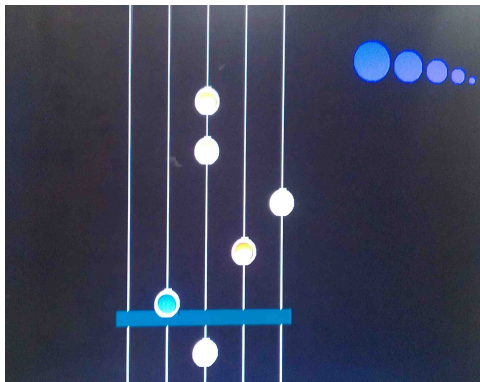


Figure: Milestone 3 and final

## The sprites being displayed



**Figure:** We display the score on the board and an easily readable score on the screen



# Timeline month 2

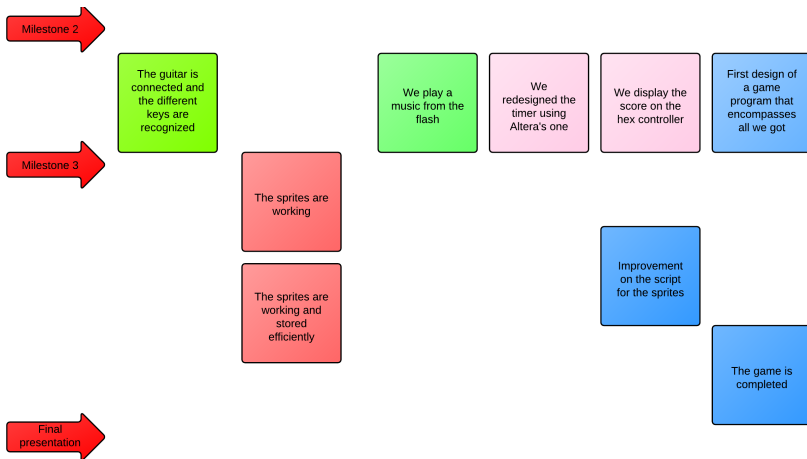


Figure: Milestone 3 and final

## Lessons learnt

- How to work **really a lot**
- Overcome **technical and practical** challenges: we have greatly improved our problem solving skills
- Don't hesitate to try things and refactor or rebuild a lot to have more maintainable and efficient code
- How to work in a team and make code that integrates well in a big architecture
- And so many other things...

Thank you for your attention