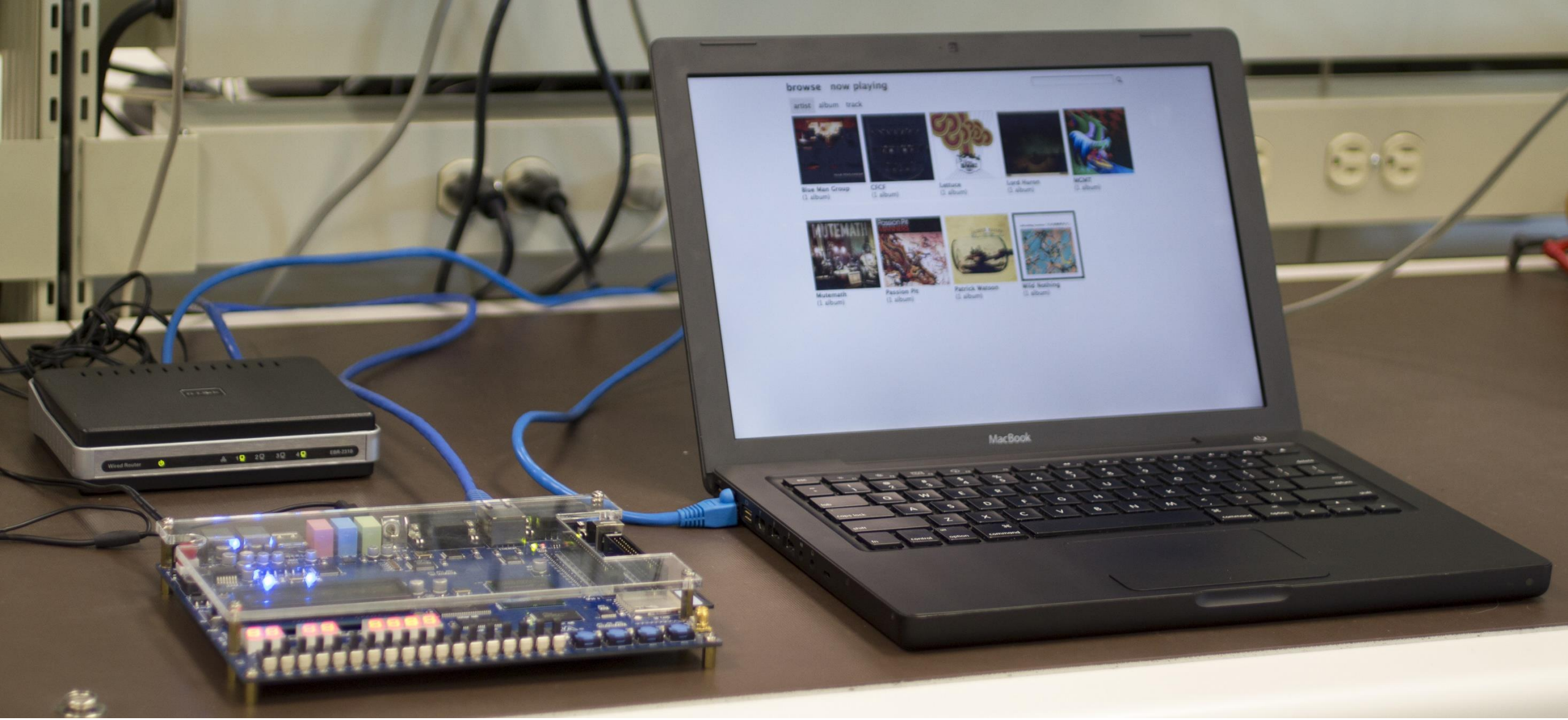


Network Controllable MP3 Player

BRADY THORNTON & JASON BROWN (GROUP 12)

Goal

A **user-friendly** MP3 player that can be controlled from any computer in your home.

How?

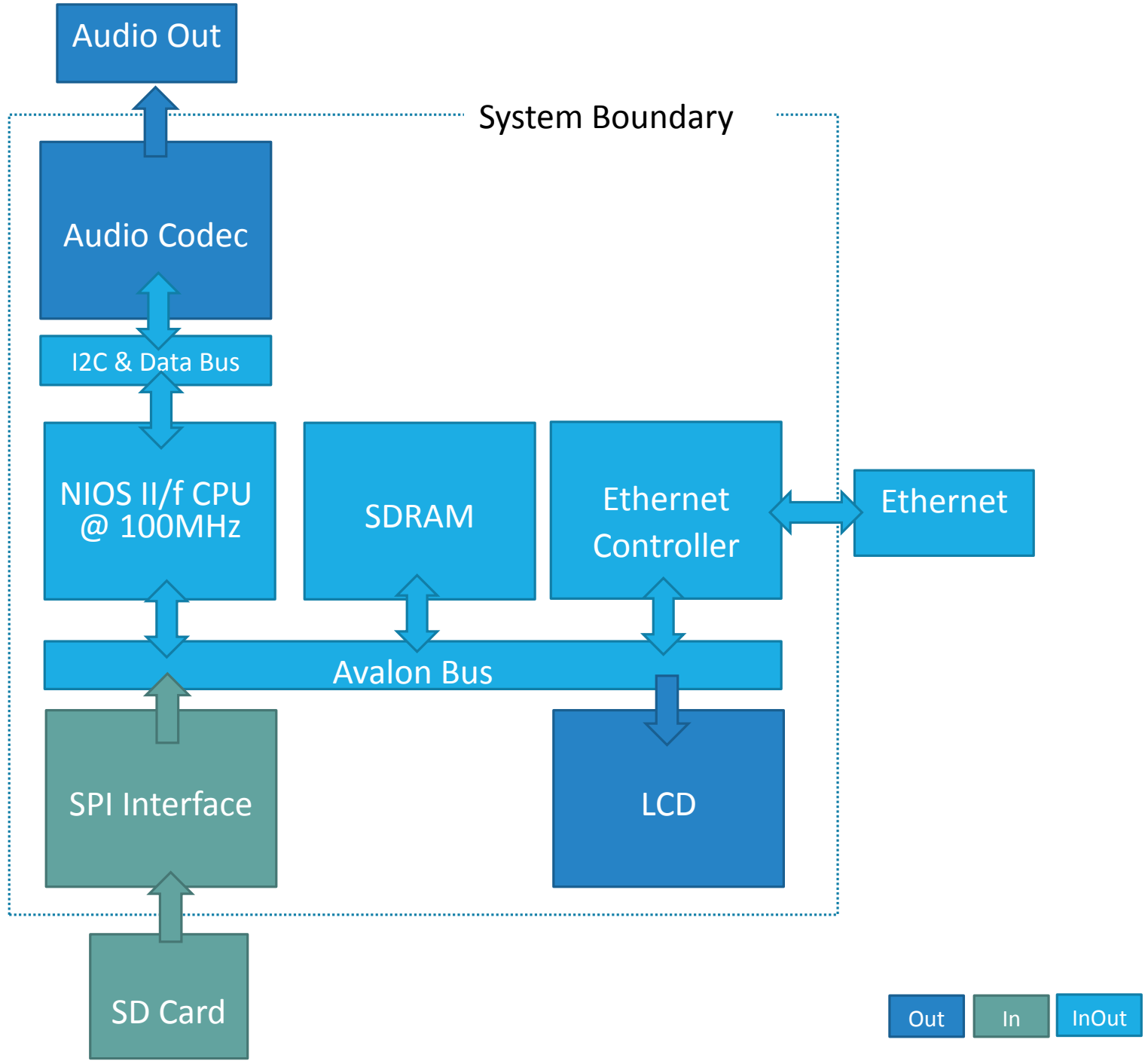
Music playback:	Decode and play MP3 files stored on an SD card.
Network Control:	Integrate a web server and client API for controlling playback.
User-Friendliness:	Design a user interface to bring these components together.

Motivation

- Practical and appealing to technical and non-technical individuals alike
- Well-defined subject matter with a clear end goal
- Interesting design challenges with streaming data, multitasking, and client-server architecture

Hardware Design

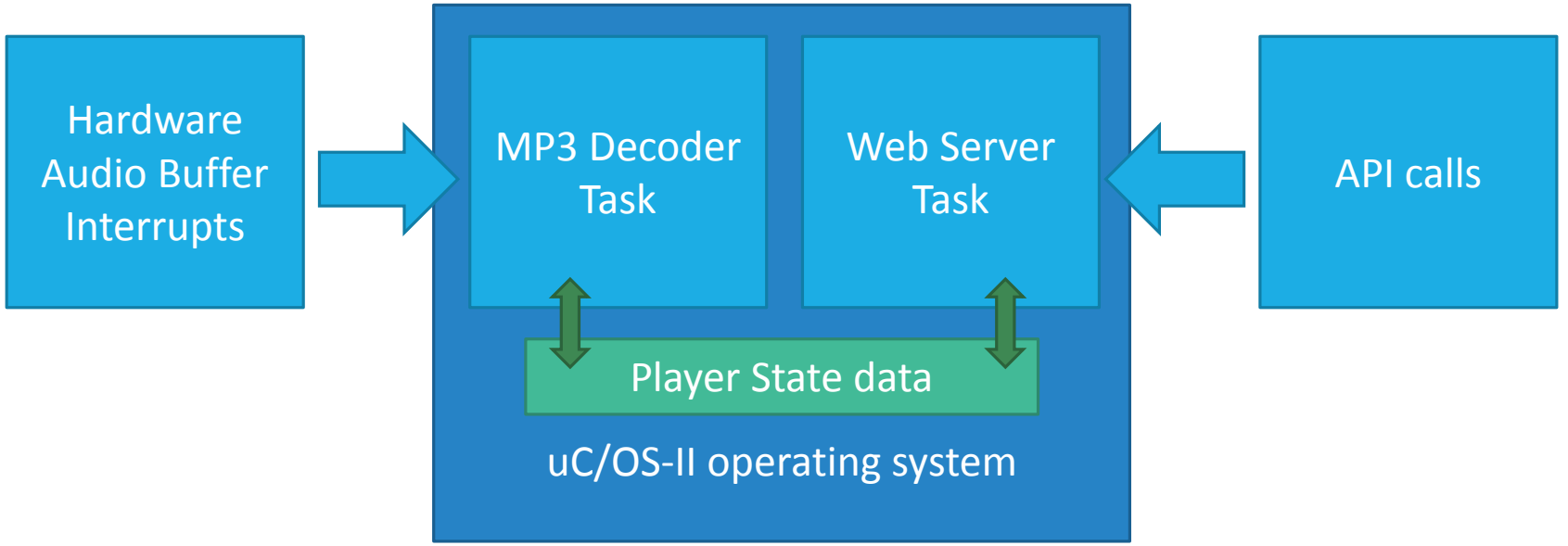
- Leverages built-in Altera DE2 components with some interfacing glue in the FPGA
- Use of open IP cores wherever possible (SD IP core, audio codec)
- Communication between blocks occurs on the Avalon bus



High-Level Features We're Proud Of

- We wrote a lightweight ID3 parsing library using the ID3v1/v2 and MPEG Layer III encoding specifications.
- Hot-swapping SD cards during playback is supported, with asynchronous client updating.
- Multiple web clients are supported and updated in real time.
- Clients connected to the internet will automatically download artist images, album covers, artist biographies, and recommended artists.

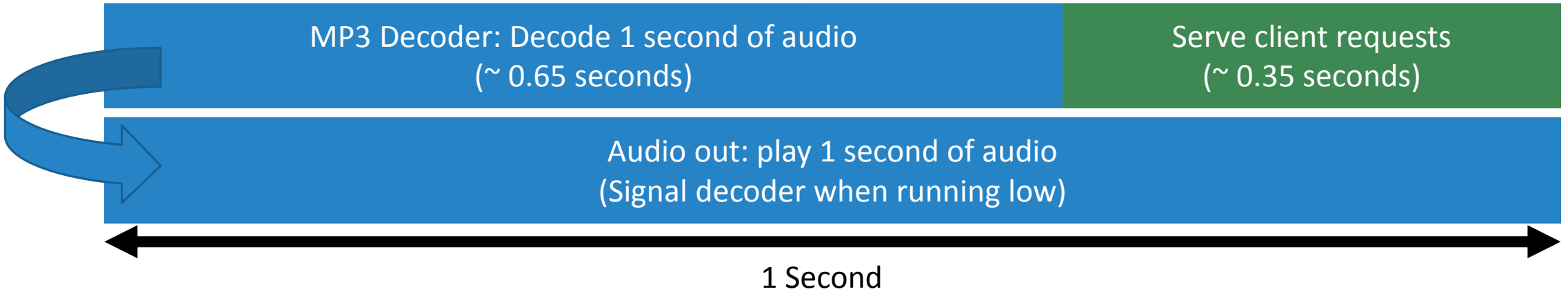
Software Overview



Multitasking Design

- We exploit interrupts on the audio codec's FIFO buffer of output samples to perform real-time task switching.
- When the FIFO runs low on data (25% full), the interrupt fires and its ISR posts to a binary semaphore, dispatching further MP3 decoding.
- Once full, the decoder task pends on the semaphore, yielding the CPU to the web server task.

Multitasking Performance



Web Task & Client-Server Architecture

- The web server task parses API calls and updates the server side state if a control request has been made.
- Player state is a C structure that's semaphore-protected to ensure there are no race conditions or other undesirable behavior during state changes.
- The player task checks the state during buffer re-loading and between tracks to control the output of audio samples.

Web Task & Client-Server Architecture

- Client initially loads web application using HTTP GET requests.
- Status checks are done once per second (per client). Control API requests are instantaneous.
- The client makes API calls using jQuery’s AJAX methods. Exchanged data is JSON formatted.

User Interface Design

- In industry, a user interface can mean the difference between a product's success or its failure.
- Our goal: a pleasant, intuitive, and responsive UI.
- Designed and coded from the ground up.

Demo

WEB APPLICATION