Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton

Man-Leong Chan
13th December, 2016

Real-time One to Many Synchronized Wireless Multimedia Entertainment System

Project Supervisor: Denis A Nicole
Second Examinor: Professor Kirk Martinez

A project progress report submitted for the award of
MEng Computer Science

Abstract

The purpose of this project is to build a household real-time synchronous wireless multimedia system(RT-SWMS) and potentially explore the limitation of such a system. Each client is modeled by a Raspberry Pi Zero attached to a portable speaker.

Contents

Acknowledgements

Many thanks to Dr. Klaus Peter Zauner and my supervisor Denis Nicole for their guidence in this project.

1 Introduction

The purpose of this project is to build a household real-time synchronous wireless multimedia system(RT-SWMS) and potentially explore the limitation of such a system. The system consist of many nodes in a network all connected by a home router using WiFi. Each of these node is modeled by a Raspberry Pi Zero connected to a portable speaker and listen to a multicast server in the same private network. However unlike Peercasting [**?**], which is multicasting via peer-to-peer technology, our system uses IP Multicast in a private network and therefore will follow the classic client-server model. Each node will return an acknoledgment back to the server upon receiving each datagram, such response time will be measured and the server will attempt to sychronize all nodes base on the delay in acknoledgement.

2 Background

2.1 Multicast

asd

2.2 Audio Quality

3 Implementation

Although this project is fundamentally a software base project, there are a few hardware issues need addressing.

3.1 Hardware

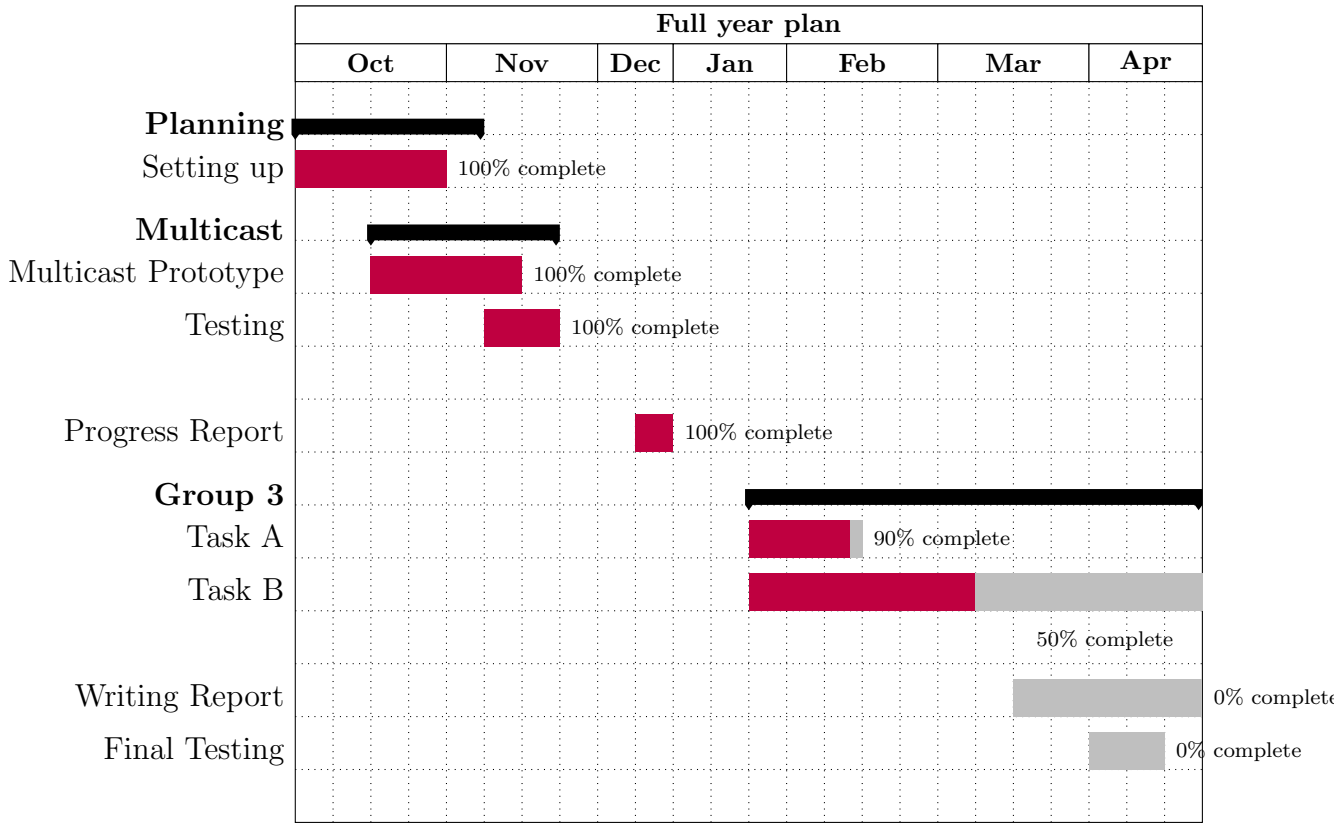
Each node is modeled by a Single-board Computer(SBC) due to its portability.

3.2 Audio

One of the problems of the full size Raspberry-Pi is that its analog output interface (which is not available in the Raspberry Pi Zero) produces noticeable amount of noise. Such noise is induced by the power supply and there is

no simple way of removing it. One of the solution is to utilize the high quality HDMI output and split the video and audio signal apart. However such solution requires an expensive HDMI video/audio splitter and would also reduce the portability of the SBC. Alternatively audio signals could also be produced by the GPIO pins on the Raspberry-Pi. However such audio output was tested to be much worst than the original analog output. Therefore a USB audio interface is used to produce much cleaner audio performance without compromising the SBC's portability.

4 Timeline



References

[1] J. Buford, H. Yu, and E. K. Lua, *P2P Networking and Applications (Morgan Kaufmann Series in Networking (Hardcover))*. Morgan Kaufmann, 2009.