Reinvented Radio

Thomas Schenk

Contents

[Introduction 3](#_Toc156165026)

[Problem 4](#_Toc156165027)

[Solution 4](#_Toc156165028)

[Planning 4](#_Toc156165029)

[Progress 5](#_Toc156165030)

[First thoughts 5](#_Toc156165031)

[Hardware Research 6](#_Toc156165032)

[Product 7](#_Toc156165033)

[Hardware Setup 7](#_Toc156165034)

[Small POC 8](#_Toc156165035)

[Current program 8](#_Toc156165036)

[Demo 9](#_Toc156165037)

[Feasibility of planning 10](#_Toc156165038)

[Reflection 11](#_Toc156165039)

## Introduction

We live in a world where cultural exploration and personal preference is easier than ever to explore, one of the leading industries in which this happens; is the music industry. Never have there been such a broad selection of music genres. Some are niche, some are mainstream. There is a genre for everyone.

Almost hand in hand with the revolutionary rise of the music industry, are the devices on which to listen to this music. Starting out with the big and bulky stereo receivers, all the way to the present day, where more artistic designs of speakers are becoming mainstream.

However, buried in all these new, artistic and creative music devices; another, far bigger, genre of audio devices is slowly rising. Retro stereo systems.

Audiophiles, name for audio-enthusiasts, have been hoarding old radios, stereos, record players, walkman’s and every other pre 1985 audio device known to man. Not only for their arguably pretty exteriors, but supposedly their guts are of sublime quality.

I had to see for myself, and I decided to give it a try; I dug up an old stereo receiver of my grandpa and have been listening to some music for the last month.

While the music is especially great and sounds very “full”. There usually is a plethora of knobs and buttons which can be used to further tune the output audio a bit.

Nothing really screams: “I am an old piece of hardware, and I am outdated”, except for getting an input signal.

## Problem

While being great, old audio hardware usually is very limited when it comes to audio input. You always have some sort of wired audio input, aux, 3.5mm, 2.5mm; some even use coax.

Our phones, if relatively new, usually don’t support wired audio output anymore. We have Bluetooth. This means we cannot connect our phone to our amazing audio system.

## Solution

Integrate a small, embedded device which adds a wireless functionality to older audio hardware. This functionality must be compatible with (my) phones.

## Planning

22 dec: first prototype with Bluetooth connection working; check if hardware is still needed in the same way; maybe buy extra?

29 dec: revise prototype, integrate possible new hardware

5 jan: finish testing prototype, especially connectivity etc; take a look at possible equalizer configurations with potentiometers (preprocessing signal on receiver side)

13 jan: Integrate preprocessing with prototype

20 jan: finished product.

## Progress

This chapter contains the progress to the solution presented at the start of this document in a chronological order.

### First thoughts

When starting a project, naturally most people have already thought about how to go to their end goal. Usually this thought pattern is either somewhat right; or downright wrong. Though we will see how that works for us.

Firstly, hardware. What are we going to use, why; and most importantly: where do we get it?

Very generally, we need two pieces of hardware:

* A means of receiving and outputting a wireless audio signal (Some integrated circuit board)
* An already working audio system.

Secondly, phasing. How is this project going to be divided? Naturally, a solution isn’t found on day one; and the road towards the solution isn’t linear. Therefore I would like to introduce a phasing to which I can adhere my progress

The phasing would consist of the following phases:

1. Hardware Research
2. Product
3. Demo
4. Reflection

I will adhere to these phases as individual chapters in the rest of this document, this keeps its tidy and very clear.

## Hardware Research

The research for hardware was meant to provide more insight into which parts are needed to build an audio transceiver.

Concluding from this research is that we need the following pieces of hardware:

1. Microcontroller (ESP32)
2. DAC (Digital to analog converter)
3. Actuator (either headphones or speakers)

According to these finds, a DAC was ordered since the ISSD did not have one. Sadly this did not go without failure, it turned out to be a limited/fake DAC. This meant I had to re-order a different DAC. Which took out a large chunk of my development time.

Finally, the new DAC arrived. Which is of the following type.  
  
A blue circuit board with different colored ports

Description automatically generated

The PCM5102MK is a DAC with LR line and AUX outputs, essentially proving two of the most common hardware interfaces of old audio equipment.

The reason we upgrade to an external DAC, is due to its resolution. The resolution of most built in DAC’s of microcontrollers, is only 8 bits. Just like our ESP32. This one however, is 32 bits. Which means that it can convey audio related information much more precisely. Resulting in a higher quality output.

## Product

After getting all the hardware needed, we can write the project code. To aid us in this, we will use a package called “Audio-Tools”, which enables us to use a lot of audio functionality on the normal Arduino platform. This library was originally even made for the esp32, but was later extended to include the whole lineage of Arduino.

The package is comparable to the Teensy audio capabilities, providing sinks, endpoints, streams, effects and way more.

### Hardware Setup

To get our DAC working properly with our ESP32, there are some connections we need to make. Below is the schematic for these connections.A diagram of a device

Description automatically generated

### Small POC

To get started, I made a small POC to create an endpoint to which I could connect via Bluetooth. The music transmitted from my phone would then we processed by the external DAC, into my earphones.

The code for this was very minimal, only needing one line in the setup.

a2dp\_sink.start("a2dp-i2s");

This worked wonderfully! And already proved to me that the core idea of this product was already sort of reached! Though, of course this is not enough; and we can make this a bit more fancy.

### Current program

After having upgraded some parts of my code, it now behaves a bit more elegantly. While simultaneously making space for future improvements. Most of these improvements are entryways to utilize effects in the future.

#include "BluetoothA2DPSink.h"

#include "AudioTools.h"

#define BCK\_PIN 26

#define WS\_PIN 25

#define DATA\_PIN 22

BluetoothA2DPSink a2dp\_sink;

I2SStream i2s;

AudioEffectStream effects(i2s);

ADSRGain adsr(0.0001,0.0001, 0.9 , 0.0002);

Fuzz fuzz(10, 300);

StreamCopy copier (i2s, effects);

Distortion diss(9000, 10000);

PitchShift shifter (2, 1000);

// Write data to I2S

void read\_data\_stream(const uint8\_t \*data, uint32\_t length) {

  effects.write(data, length);

}

void setup() {

  Serial.begin(115200);

  // register callback

  a2dp\_sink.set\_stream\_reader(read\_data\_stream, false);

  // Start Bluetooth Audio Receiver

  a2dp\_sink.set\_auto\_reconnect(false);

  a2dp\_sink.start("a2dp-i2s");

  // setup output

  auto cfg = i2s.defaultConfig();

  cfg.pin\_bck = BCK\_PIN;

  cfg.pin\_ws = WS\_PIN;

  cfg.pin\_data = DATA\_PIN;

  i2s.begin(cfg);

  copier.begin();

  effects.begin(cfg);

  effects.addEffect(shifter);

}

void loop() {

  copier.copy();

 }

The biggest change is that instead of using a dedicated stream for the Bluetooth communication, we read its input to a stream. This means that we can use all the capabilities a stream gives us (dynamically adjusting parameters, effects, filters etc) without giving up the better implementation of the ‘Bluetooth sink’ class.

I already inputted various effects as templates to use, these need to be added to the “effects” stream to be viable. Currently the pitch shifter is attached, which shifts the pitch. This makes listening to music pretty funny.

## Demo

To demonstrate how this device can be used, I hooked it up to my own radio system which does not have a Bluetooth module inside of it. The wireless capabilities shown, are all made by myself.

*This video is posted in the same folder under the name bluetooth\_demo.mp4.*

## Feasibility of planning

In order to validate my planning, I want to look at it again and provide insights into what stage I was in at those specific times.

To recall the planning:

22 dec: first prototype with Bluetooth connection working; check if hardware is still needed in the same way; maybe buy extra?

29 dec: revise prototype, integrate possible new hardware

5 jan: finish testing prototype, especially connectivity etc; take a look at possible equalizer configurations with potentiometers (preprocessing signal on receiver side)

13 jan: Integrate preprocessing with prototype

20 jan: finished product.

22 dec:  
I had the first prototype working, I did this with the onboard DAC, which according to my research had told me: sounded horrible. Like I predicted when making the planning: check if new hardware is needed. This is the moment I ordered the first DAC.

29 dec:  
The DAC had arrived, and while trying to integrate it; I immediately ran into a bunch of assorted problems. I was missing pins, I had a terrible buzz and overall the DAC didn’t seem to respond to what I did. I found out I had a faulty DAC, and ordered a new one.

5 jan:

Due to the holidays, much development time was lost. Especially since I was still in a phase which I did not account being in. The new DAC arrived, and I tried to make progress implementing its usability.

13 jan:  
I now had a working prototype which could connect to a bluetooth source and play music with no bugs which I discovered!

20 jan:  
*Cannot say anything*

## Reflection

Reflecting back on this idea and execution, I am fairly happy with the unique idea and the usecase it will have in my personal life. Since I will continue using and developing this product for a long time!

What I am a bit less happy about is the execution and how my planning was completely overthrown by a series of unfortunate events. First and foremost, the wrong DAC. Like mentioned before, I received a “DAC” which was so limited in its use, that it couldn’t be called a DAC. I therefore had to reconfigure what DAC I would use, and order a whole new one.

Apart from that, the last time I had to focus on this project; I got extremely sick. Already feeling rushed by the previous problem, this obviously wasn’t great.

Apart from that, I think it is a fun project!