



Sound Assist

Adam Seubert, Alex Inskeep



Project goal

- To create an assistive listening device to increase the intelligibility of human speech in common media, such as movies and television.
- The device *must* increase the intelligibility of human speech.
- The device *should* be customizable to suit the needs of individual users.
- The device may include a wireless transmission system.



Design Overview

Analog Audio Stream

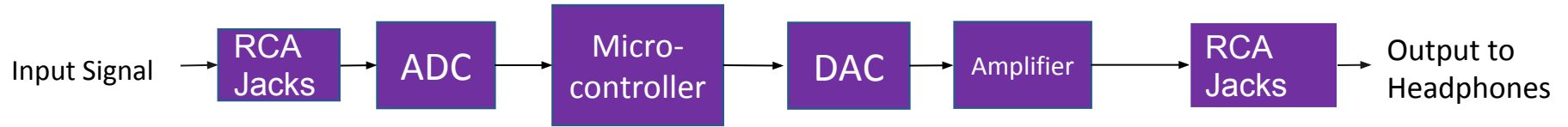


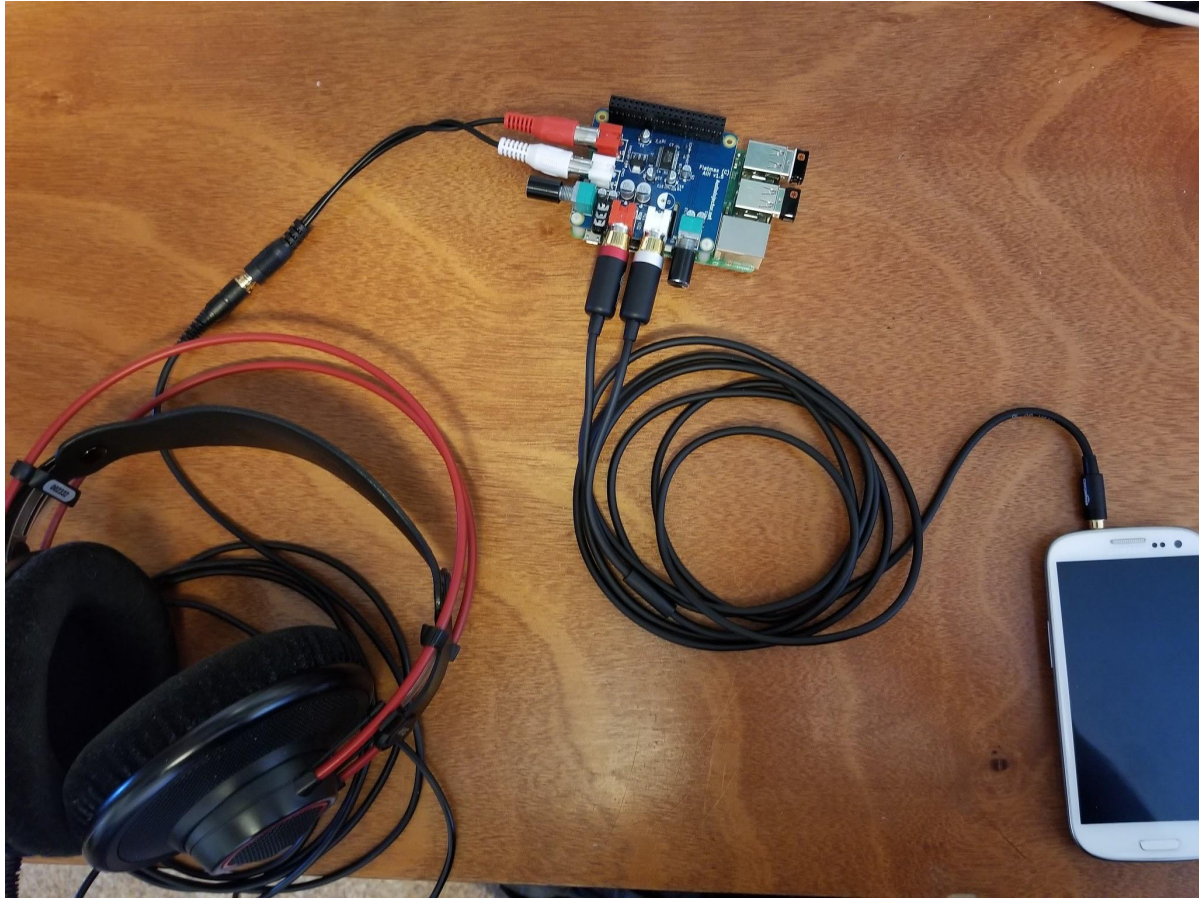
Sound Assist Device

Vocal Boosted Analog Audio Stream



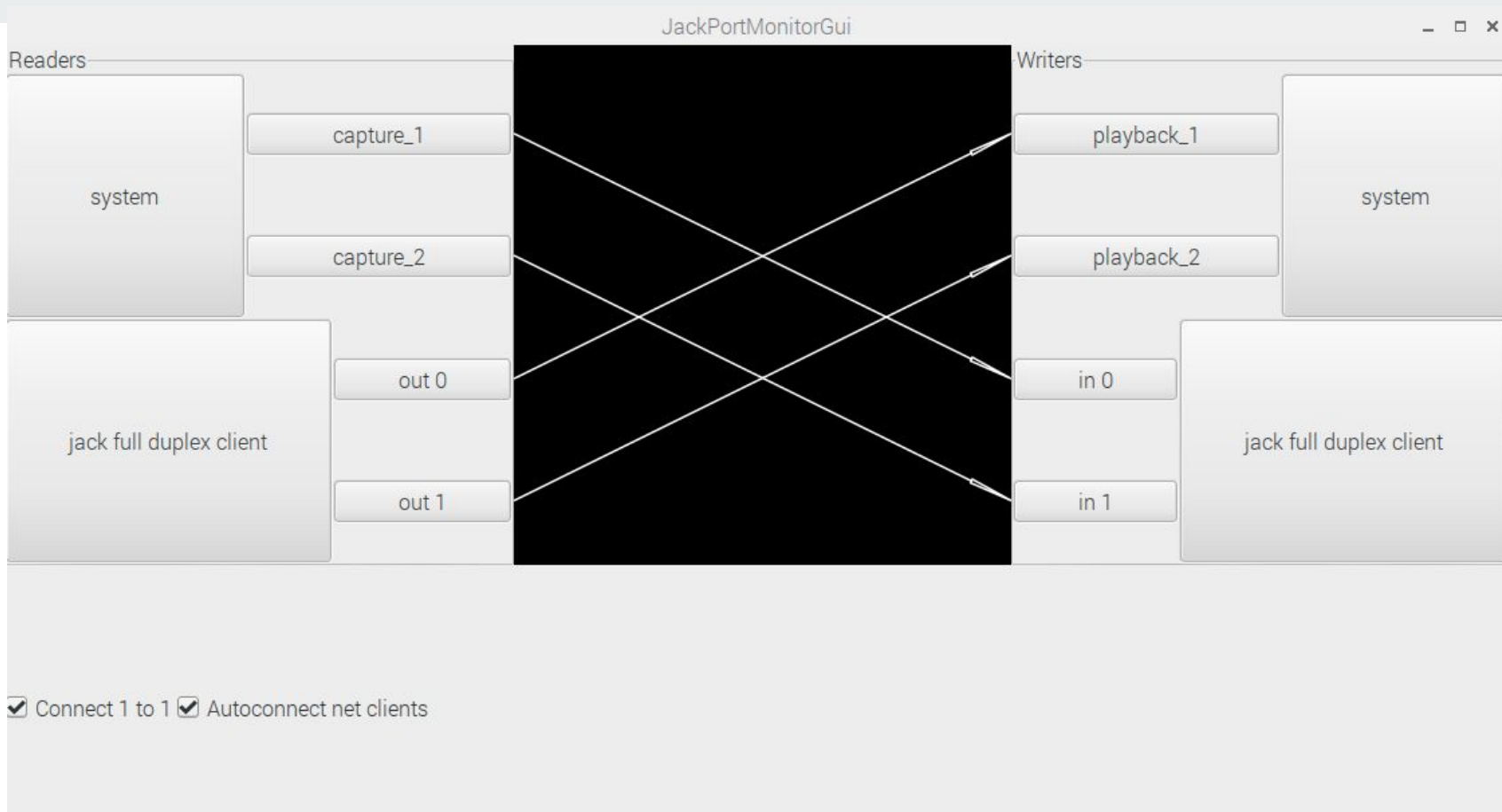
Hardware Design overview





Software Design Overview





```

1 #include <JackClient.H>
2 #include <iostream>
3 using namespace std;
4
5 #include <unistd.h> // for sleep
6
7 /** This jack client will play back what it captures
8  */
9 class JackFullDuplex : public JackClient {
10
11     // The Jack client callback - copy input to output
12     int processAudio(jack_nframes_t nframes) {
13         if (outputPorts.size()!=inputPorts.size()){
14             cout<<"Different input and output port count, don't know how to copy. In port cnt="<<inputPorts.size()<<" output port count="<<outputPor
15             return -1;
16         }
17
18         for (uint i=0; i<outputPorts.size(); i++) { // for each channel of audio, copy input to output
19             jack_default_audio_sample_t *out = ( jack_default_audio_sample_t* ) jack_port_get_buffer ( outputPorts[i], nframes ); // the output samp
20             jack_default_audio_sample_t *in = ( jack_default_audio_sample_t* ) jack_port_get_buffer ( inputPorts[i], nframes ); // the inputs sample
21             for (uint j=0; j<nframes; j++){ // rather then copying, you could do some signal processing here!
22                 //cout << "in: "<<in[j] << " out: " << in[j]*0.4<<endl;
23                 out[j]=in[j];
24             }
25         }
26         return 0;
27     }
28 };
29
30 int main(int argc, char *argv[]) {
31     JackFullDuplex jackClient; // init the full duplex jack client
32
33     // connect to the jack server
34     int res=jackClient.connect("jack full duplex client");
35     if (res!=0)
36         return JackDebug().evaluateError(res);
37
38     cout<<"Jack : sample rate set to : "<<jackClient.getSampleRate()<<" Hz"<<endl;
39     cout<<"Jack : block size set to : "<<jackClient.getBlockSize()<<" samples"<<endl;
40

```

le /home/pi/srdesProject/JackClient.C opened(17).

le /home/pi/srdesProject/JackClient.C saved.



End Product

- Plug and Play experience
 - Scripts start all programs on powerup
 - No additional connections or UI needed

- Compact and powerful solution with advancement options



Advanced options

- **Degenerate Unmixing Estimation Technique (DUET)** is a Blind Source Separation Algorithm for stereo audio.
- It can separate a stereo mixture into an arbitrary number of components
- It appears to remove about 60% of the background noise from various samples.
- Degrades audio quality some, but should be unnoticable once remixed with original signal.



Got Questions?