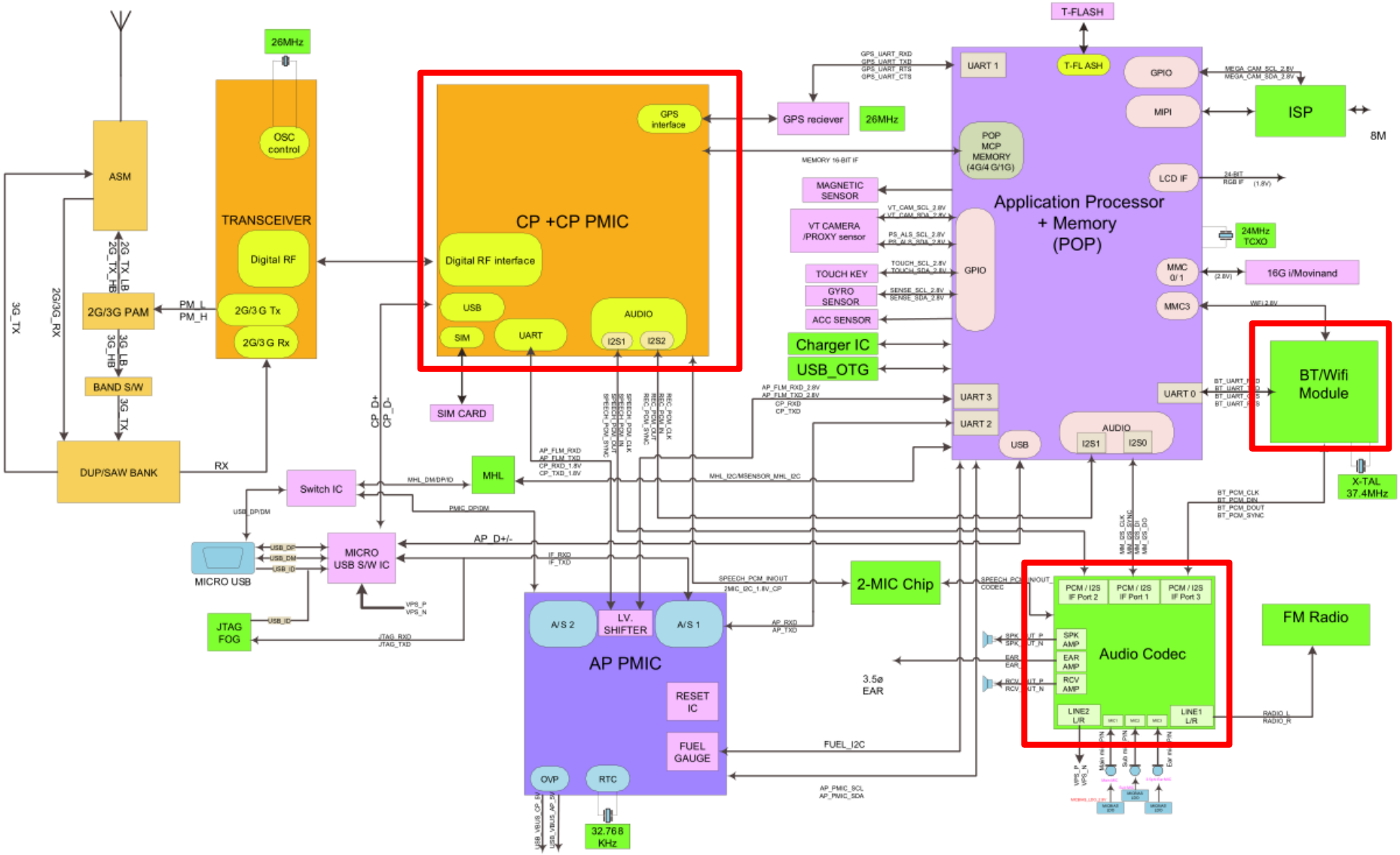
Introduction to Android Sound Sensor

CS 436 Software Development on Mobile

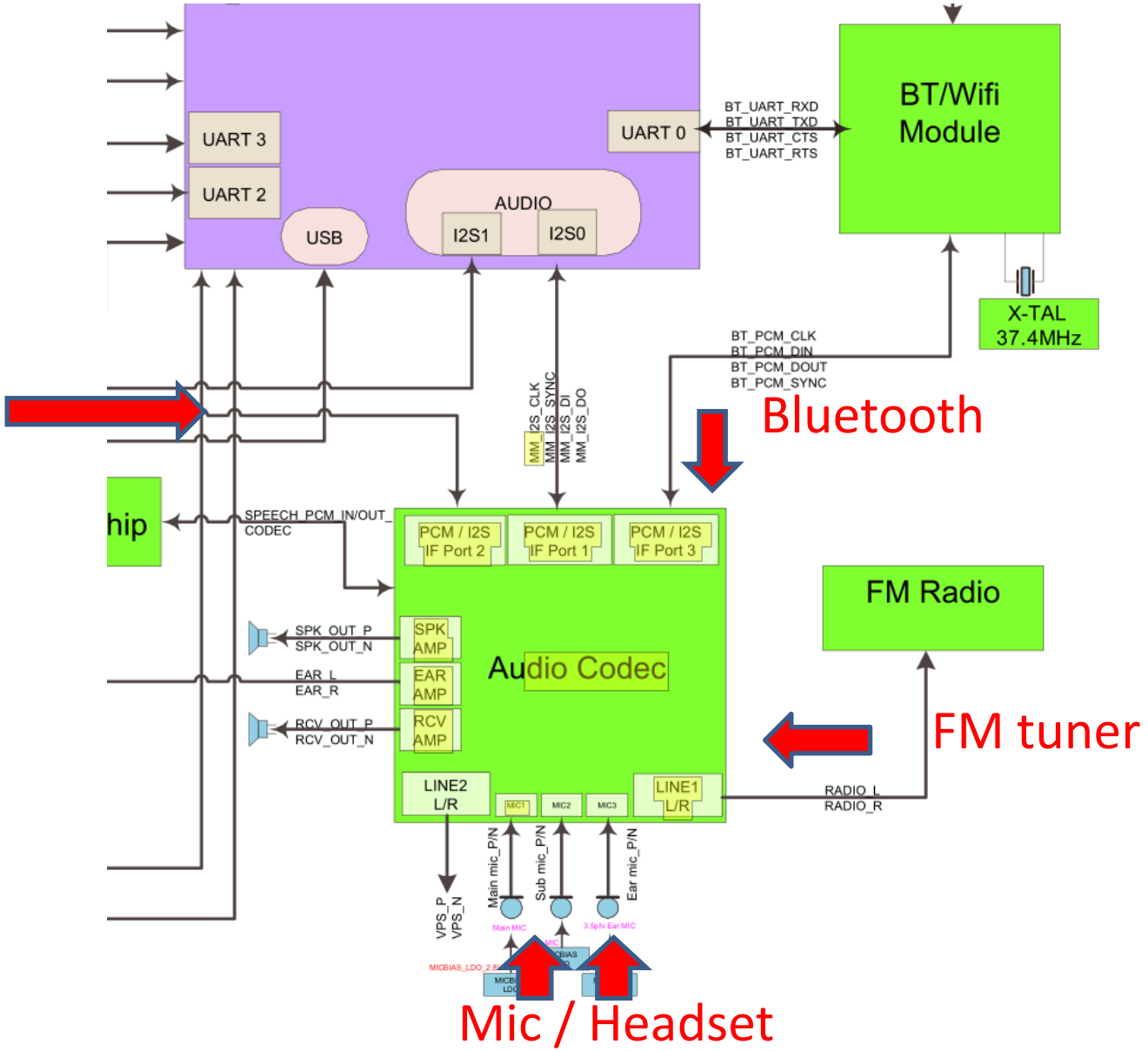
Dr.Paween Khoenkaw

Department of Computer Science
Maejo University

Sound recorder

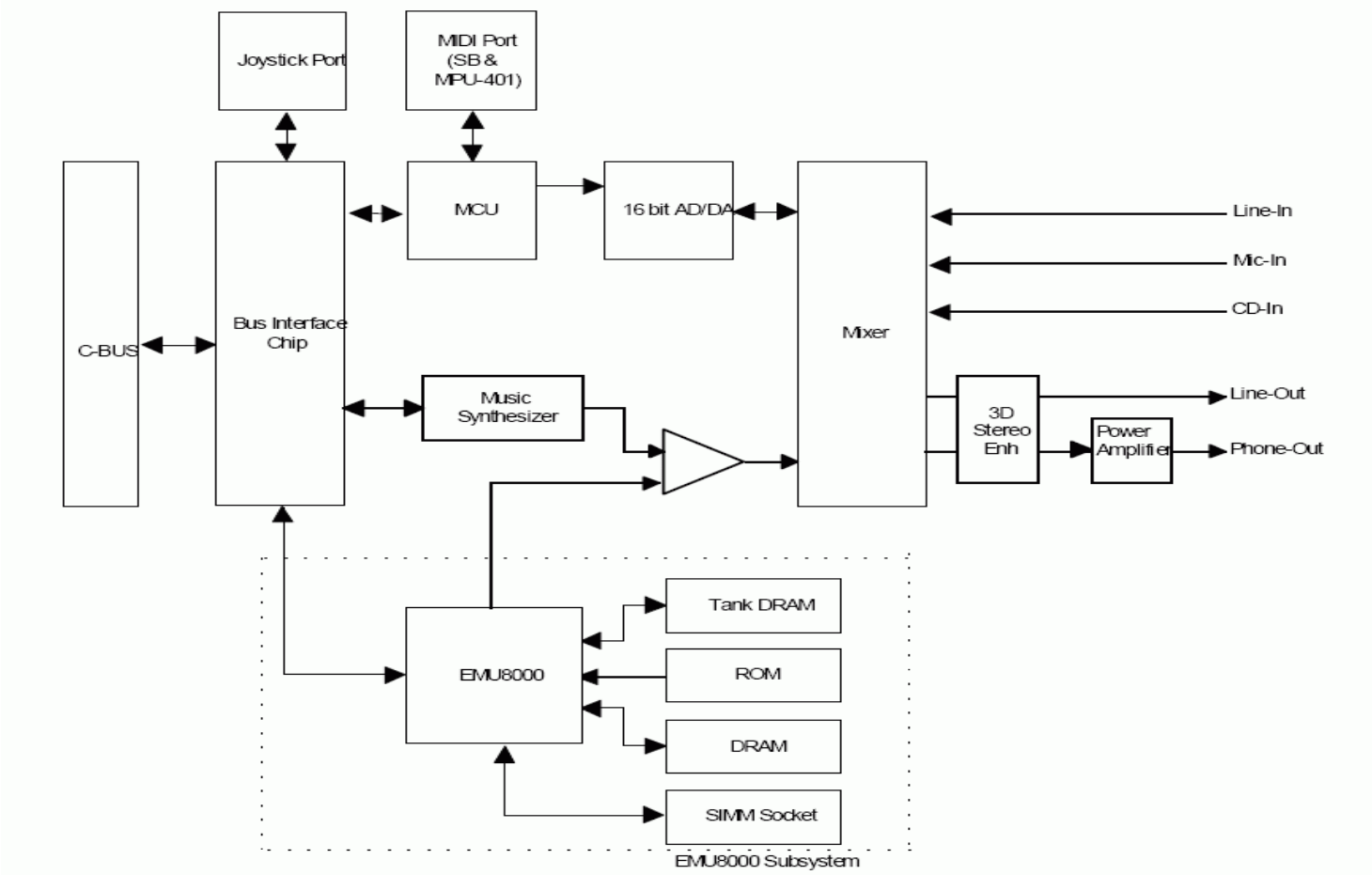


Sound recorder



Sound recorder

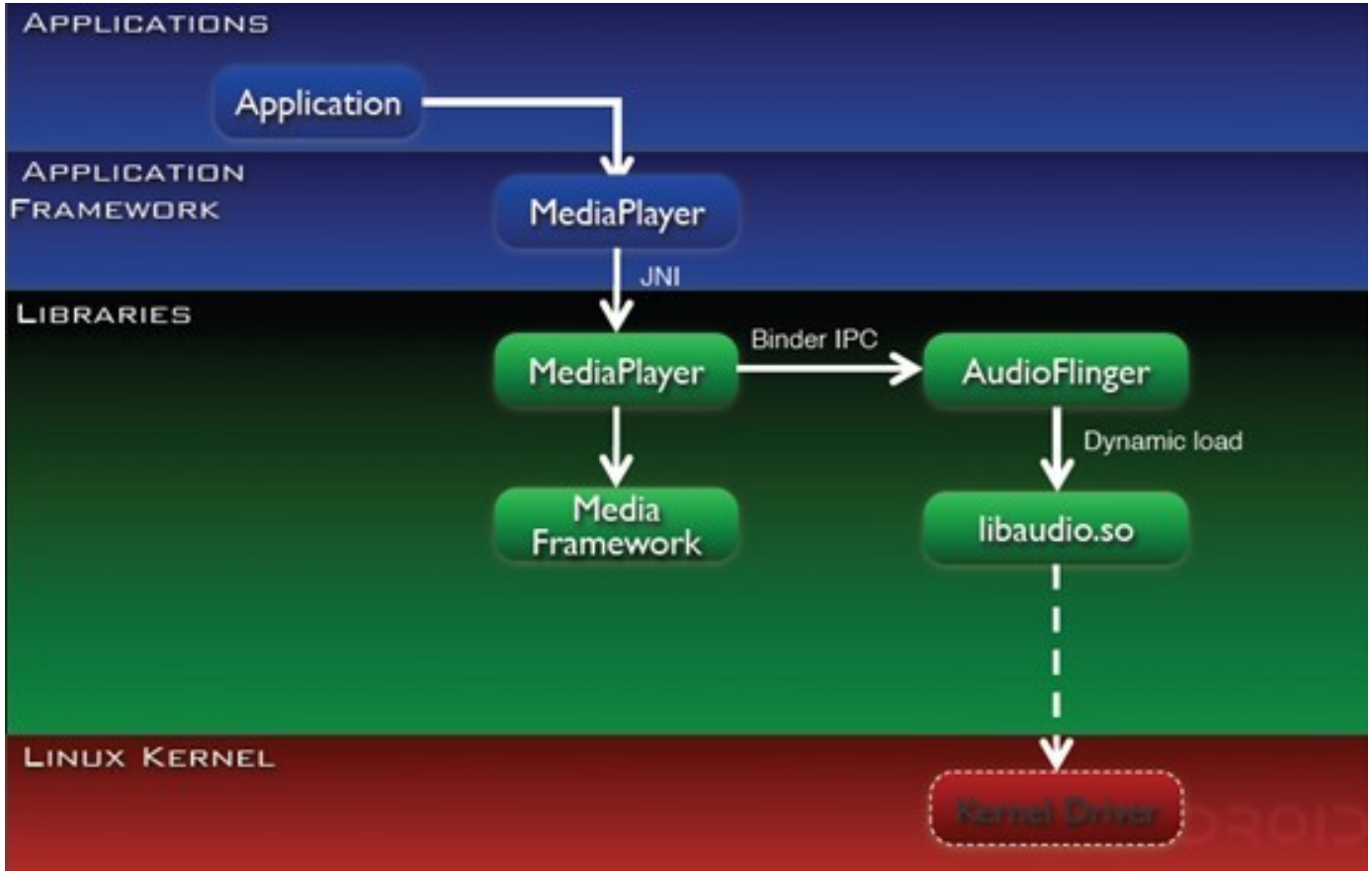
Compare with PC's Sound card



SoundBlaster AWE32 architecture

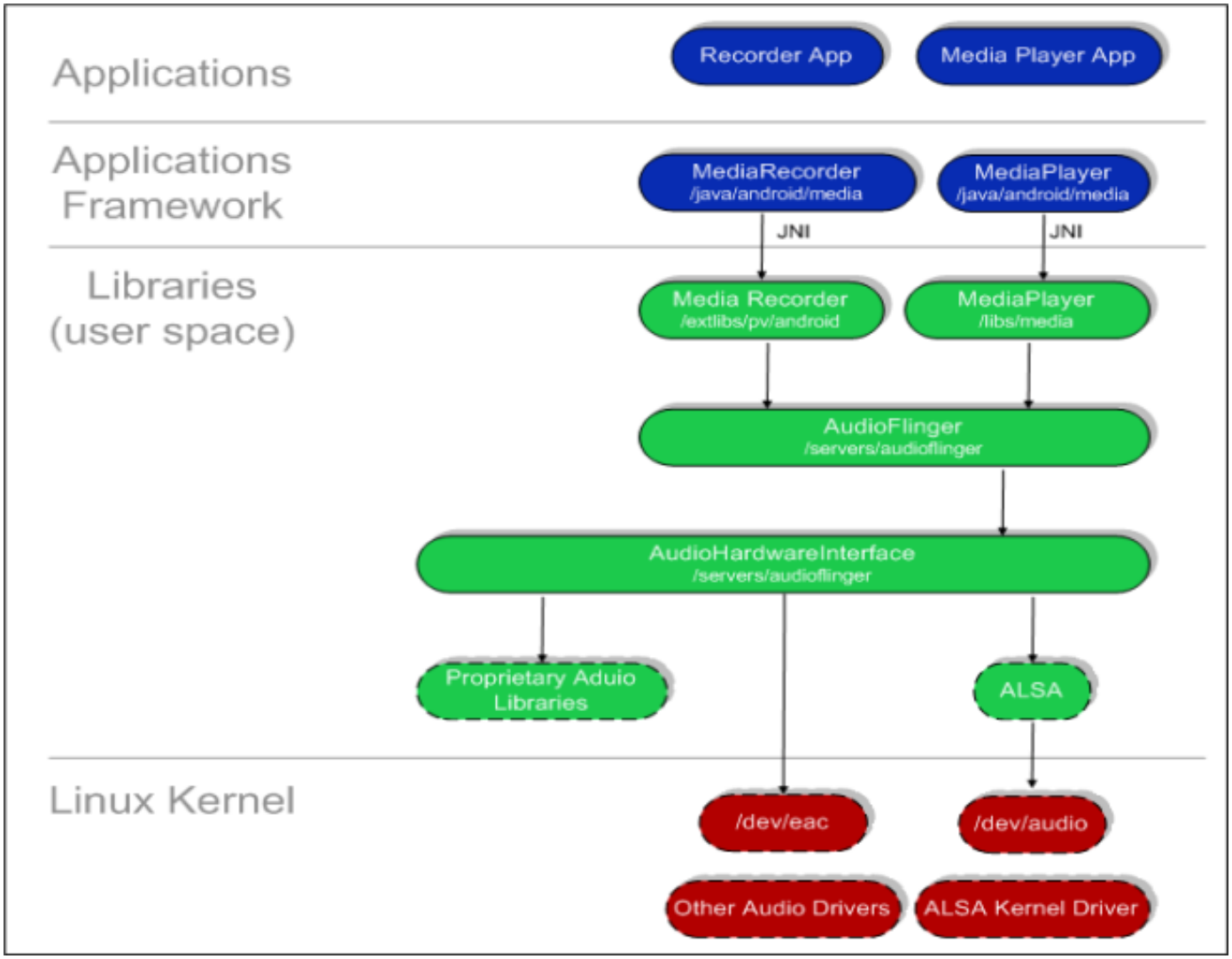
Sound recorder

Android Audio framework



Sound recorder

Android Audio framework



Sound recorder

Sound recording application

1) Record compressed audio to file (SDCARD)
2) Record PCM audio sample to memory

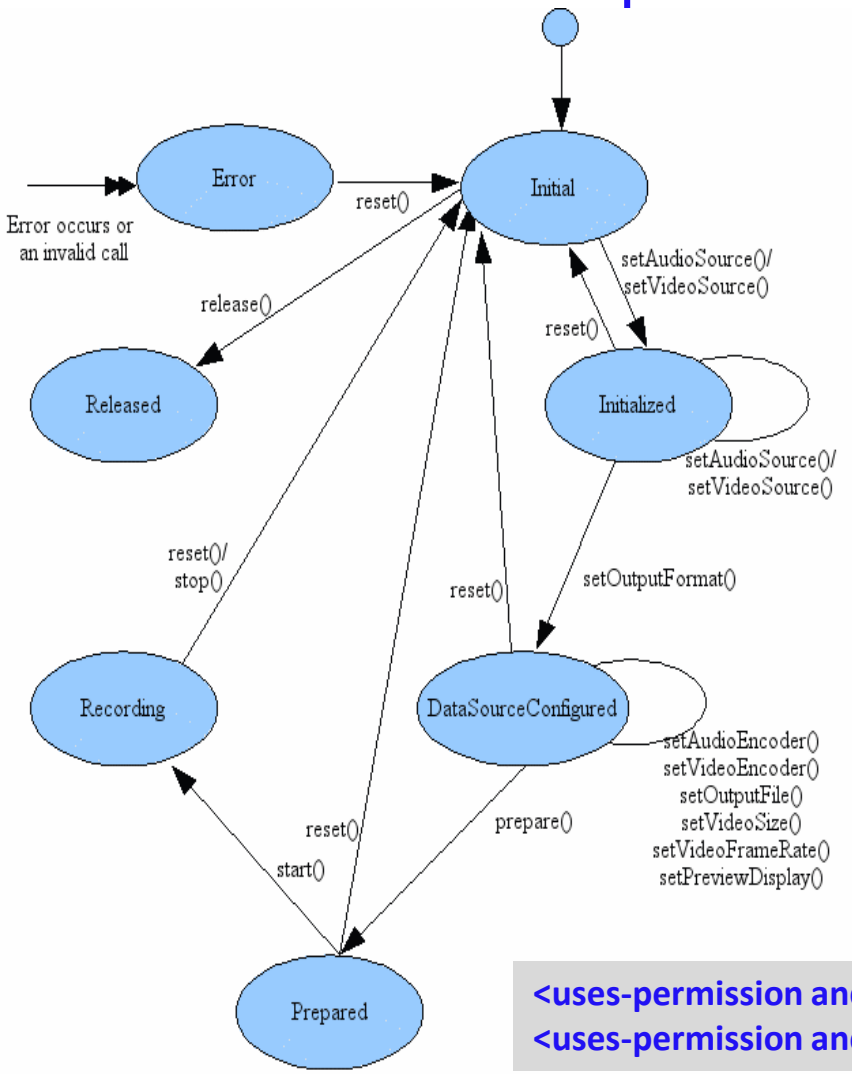
Sound recorder

Record compressed audio to file (SDCARD)

Step 1) Ask permission to record audio and write file
Step 2) Set record source
Step 3) Set output format
Step 4) Set audio encoder (codec)
Step 5) Set output file
Step 6) Start recording

Sound recorder

Record compressed audio to file (SDCARD)



MediaRecorder state diagram

```
final MediaRecorder audioRecorder = new MediaRecorder();
 audioRecorder.setAudioSource(MediaRecorder.AudioSource.DEFAULT);

audioRecorder.setOutputFormat(MediaRecorder.OutputFormat.DEFAULT);
 audioRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);
audioRecorder.setOutputFile(fname);

audioRecorder.prepare();
 audioRecorder.start();

 audioRecorder.stop();
 audioRecorder.release();
```

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Sound recorder

Native audio codec supported

Format / Codec	Encoder	Decoder
AAC LC	•	•
HE-AACv1 (AAC+)	• (Android 4.1+)	•
HE-AACv2 (enhanced AAC+)		•
AAC ELD (enhanced low delay AAC)	• (Android 4.1+)	• (Android 4.1+)
AMR-NB	•	•
AMR-WB	•	•
FLAC		• (Android 3.1+)
MP3		•
MIDI		•
Vorbis		•
PCM/WAVE	• (Android 4.1+)	•

Sound recorder

!!Warning Audio recording is not a shareable resource!!

Sound recorder

Native codec that supported by Android

-AAC
-AMR NB
-AMR WB

Sound recorder

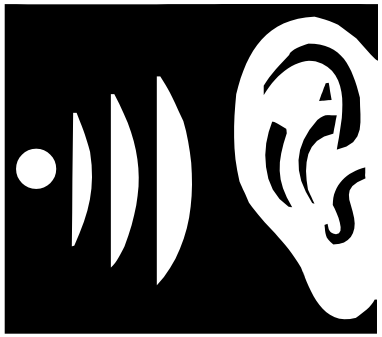
Advanced Audio Coding (AAC)

-Designed to be the successor of the MP3 format
-Lossy compression
-Perceptual noise shaping
-Based on characteristics of the human ear

Sound recorder

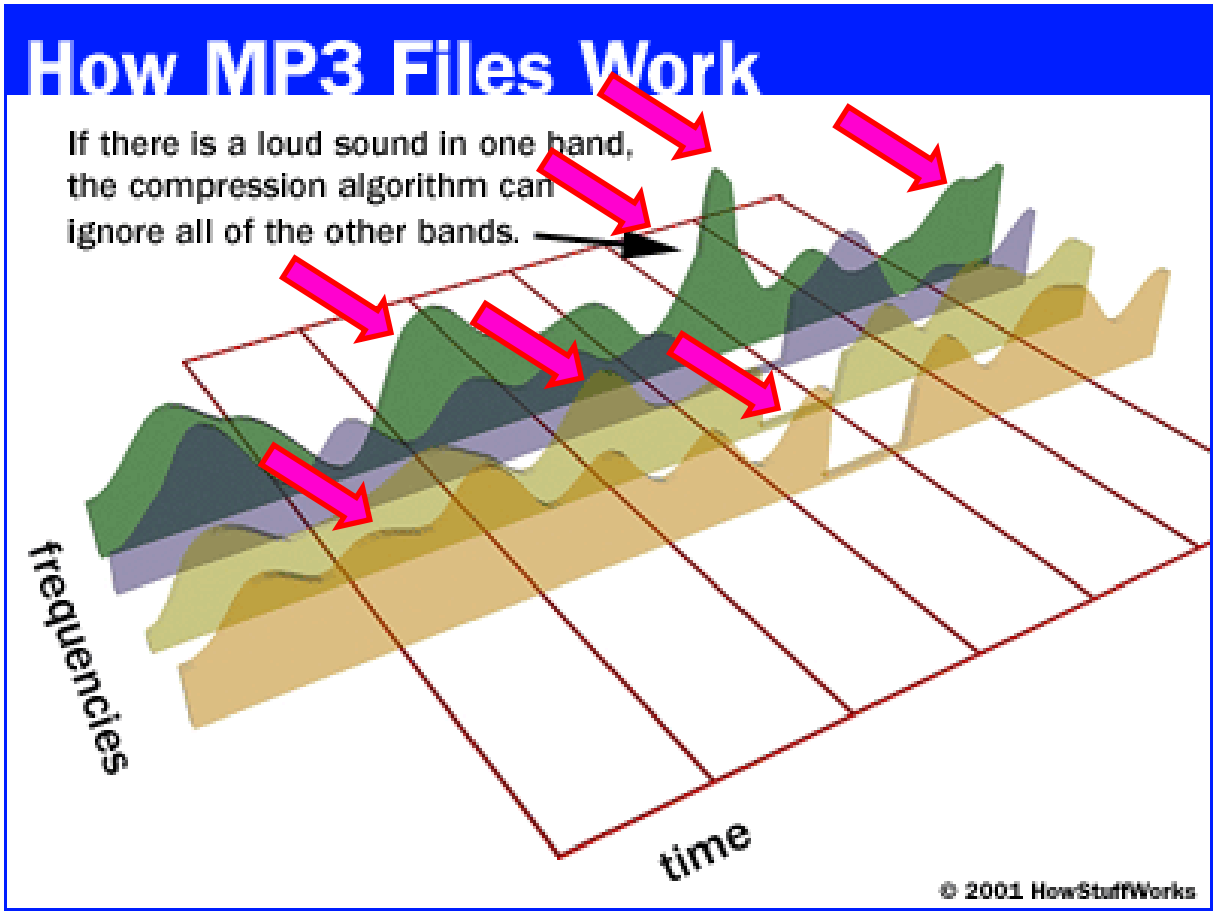
Advanced Audio Coding (AAC)

1) There are certain sounds that the human ear cannot hear.
2)There are certain sounds that the human ear hears much better than others.
3)If there are two sounds playing simultaneously, we hear the louder one but cannot hear the softer one.



Sound recorder

Advanced Audio Coding (AAC)



frequency-masking and time-masking

A tone could be rendered inaudible by another tone of lower frequency

Sound recorder

Advanced Audio Coding (AAC)

AAC's improvements over MP3

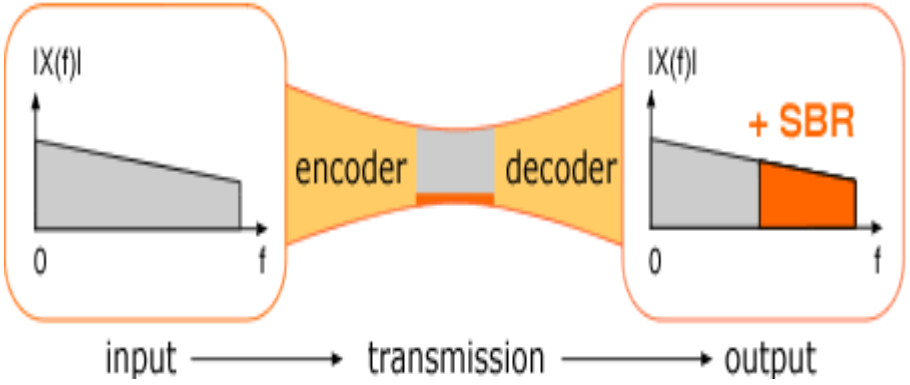
-Use wider sampling window

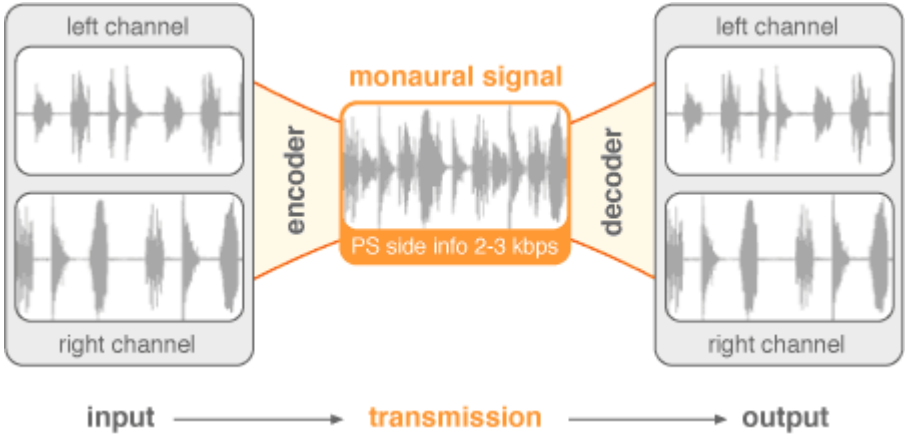
-Better encode both stationary signals and transient signals

-SBR Technique (Side Band Replication)

-Parametric Stereo Technique (PS)

-Encode our sounds to 64Kbit/s with the same quality of a 128Kbit/s encoded MP3





http://sonnati.wordpress.com/2008/01/18/183/

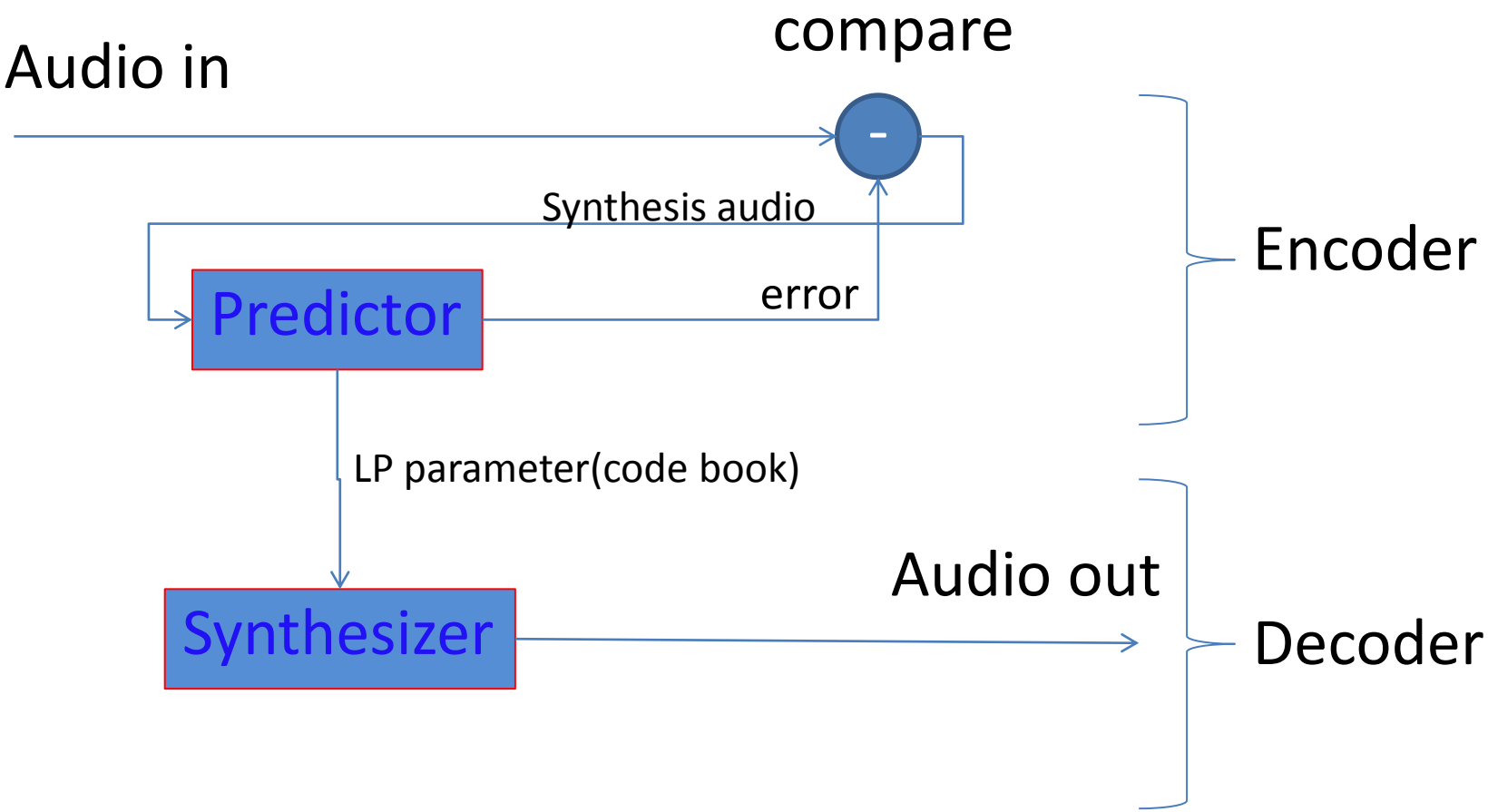
Sound recorder

Adaptive Multi-Rate audio codec(AMR)

-Algebraic code-excited linear prediction (ACELP)
-Multi-rate Narrow band
-Multi-rate Wide band
-Optimize for speech

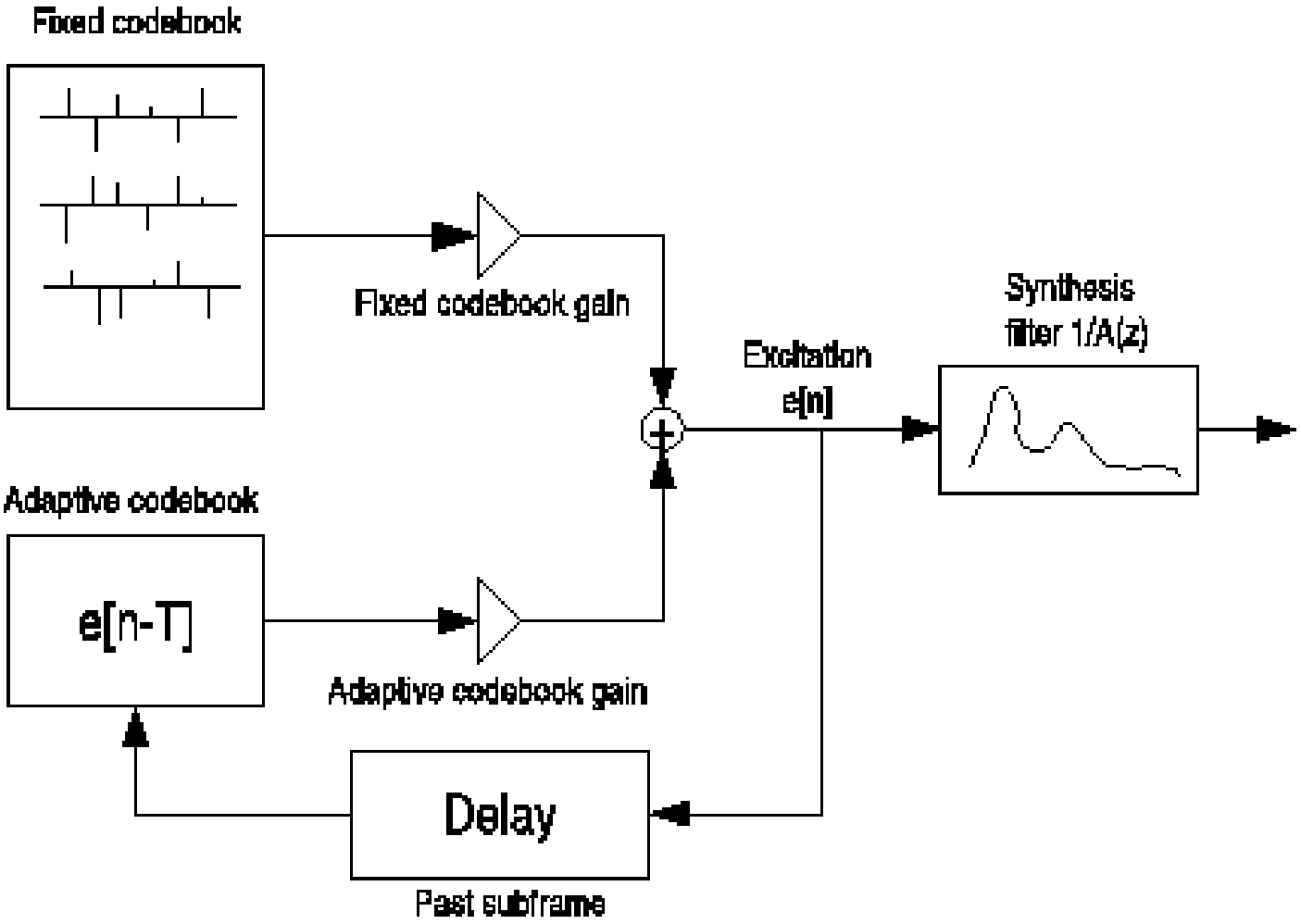
Sound recorder

Linear predictive coding (LPC)



Sound recorder

Code Excited Linear Prediction(CELP)



http://www.speex.org/docs/manual/speex-manual/node9.html

Sound recorder

Adaptive Multi-Rate audio codec(AMR)

-Algebraic code-excited linear prediction (ACELP)
-Discontinuous transmission (DTX)
-Voice activity detection (VAD)
-Comfort noise (or comfort tone)

Sound recorder

Adaptive Multi-Rate audio codec(AMR)

Mode	Bitrate (kbit/s)	Channel	Compatible with
AMR_12.20	12.20	FR	ETSI GSM enhanced full rate
AMR_10.20	10.20	FR	
AMR_7.95	7.95	FR/HR	
AMR_7.40	7.40	FR/HR	TIA/EIA IS-641 TDMA enhanced full rate
AMR_6.70	6.70	FR/HR	ARIB 6.7 kbit/s enhanced full rate
AMR_5.90	5.90	FR/HR	
AMR_5.15	5.15	FR/HR	
AMR_4.75	4.75	FR/HR	
AMR_SID	1.80	FR/HR	

FR=full rate channel, HR=half rate channel

Sound recorder

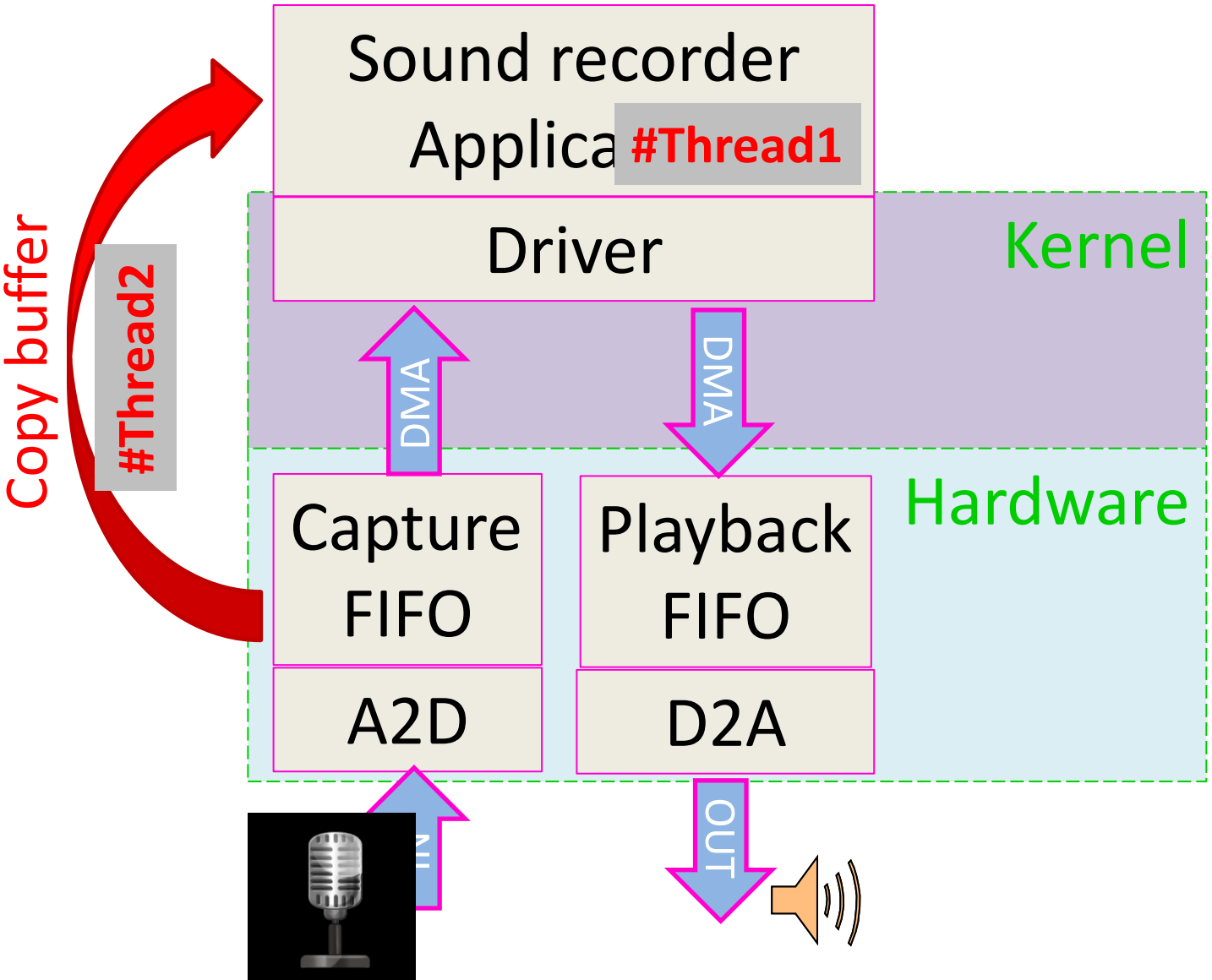
Conclusion

-AAC is codec for music and movie
-AMR is codec for speech recording
-You cannot record in MP3, WMA and OGG format
-Jellybean support recording in WAV format

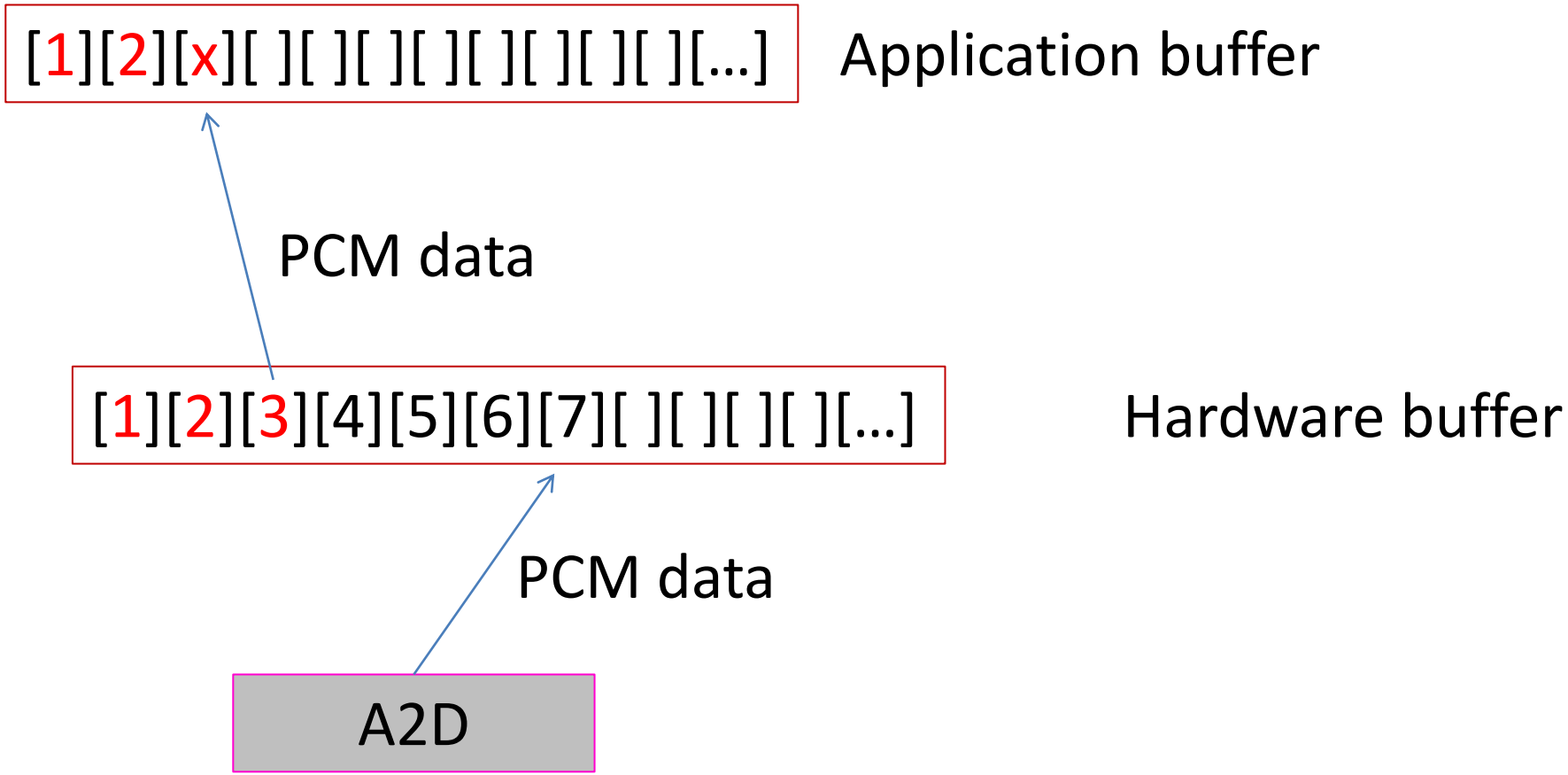
Sound recorder

Record PCM audio sample to memory

Sound recorder



Sound recorder

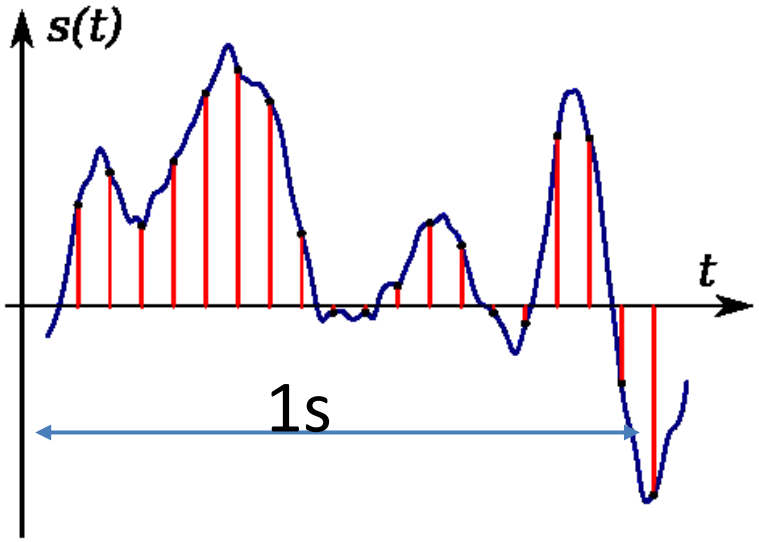


Sound recorder

What is PCM?

Sound recorder

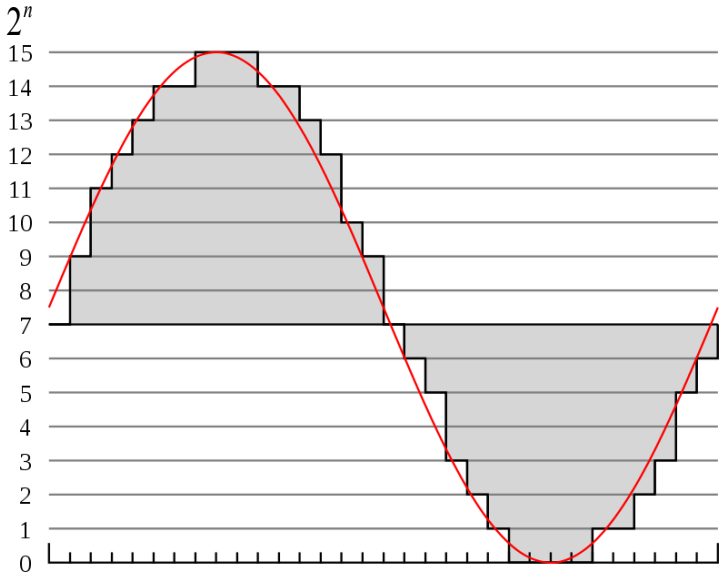
Sampling

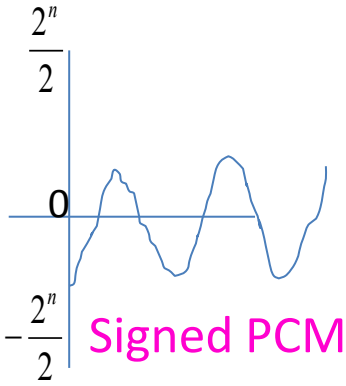


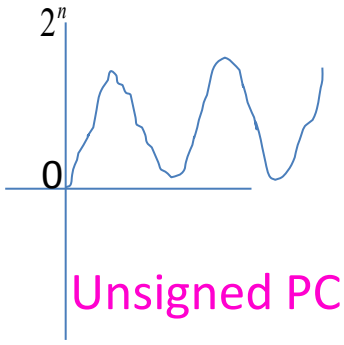
$$\text{Sampling rate} = \frac{1}{\Delta t}$$

Sampling rate = number of sample per second

Quantize







Sound recorder

How many buffer need to record 44.1kHz unsigned16bit mono for 20ms ?

20ms need 44100 x 0.02 =882 samples

882 sample need 882 x 16 x 1 =14112 bits =1764 bytes

The answer is 1764 bytes

Sound recorder

How to get recording buffer

Step 1) Create empty buffer
Step 2) Select audio source
Step 3) Initial AudioRecorder object
Step 4) Start recording
Step 5) Loop to get audio data
Step 6) Stop recording and release resource

Sound recorder

Initial configuration for AudioRecorder object

```
public AudioRecord audioRecord;
public int mSamplesRead; //how many samples read
public int buffersizebytes;
public int buflen;
public int channelConfiguration = AudioFormat.CHANNEL_IN_MONO;
public int audioEncoding = AudioFormat.ENCODING_PCM_16BIT;
public static short[] buffer; //+-32767
public static final int SAMPPERSEC = 22050; //samp per sec 8000, 11025, 22050 44100 or 48000
Handler h;
```

!!Not all device support all sample rate and all encoding!!

Sound recorder

Get data from buffer

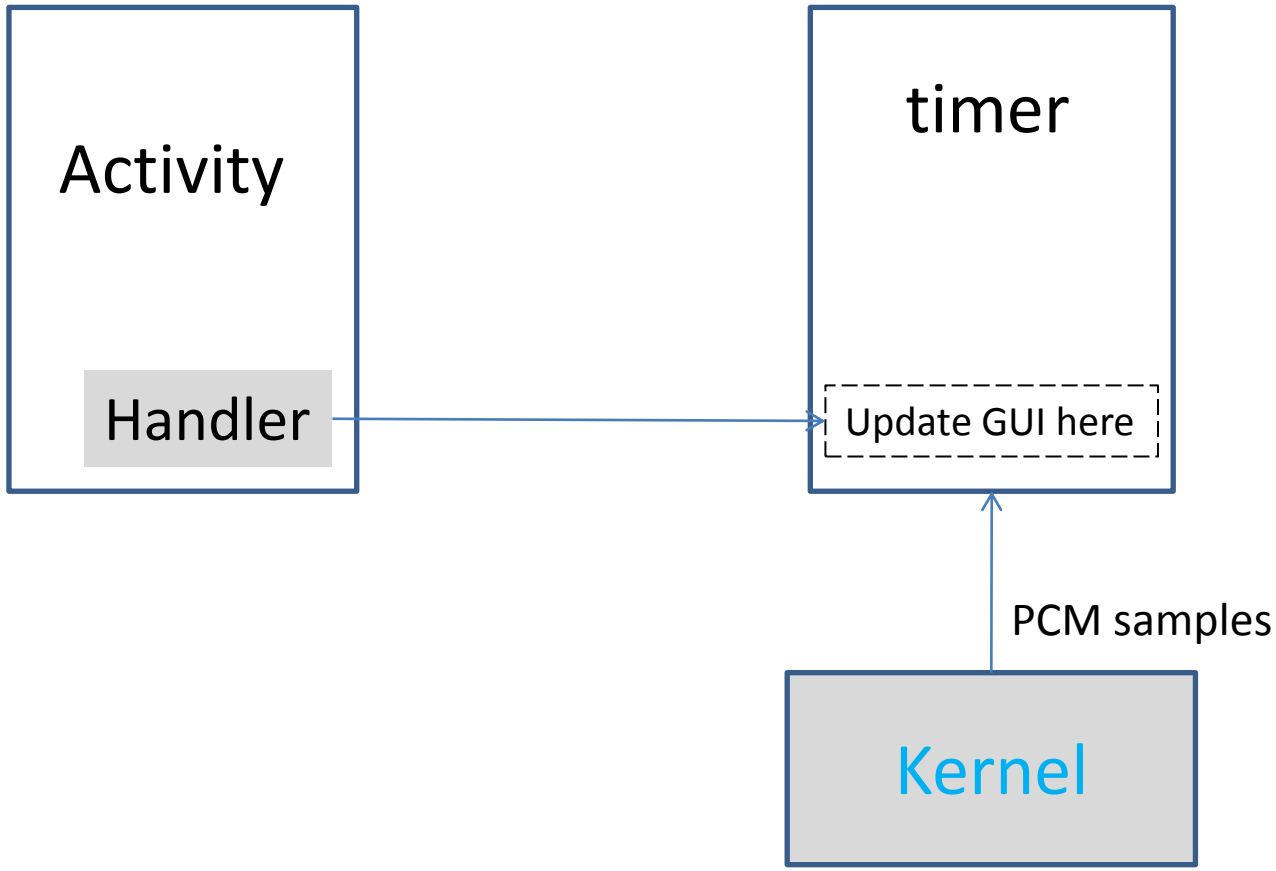
```
mSamplesRead = audioRecord.read(buffer, 0, buffersizebytes);
```

This method must run in separate thread

The easiest way is to implement in timer class

Sound recorder

Update GUI thread



Asynchronous call is required!

Sound recorder

Update GUI thread

```
protected void onCreate(Bundle savedInstanceState) {
                        . . . . . .
                        h=new Handler();
                        }
```

```
h.post(UpdateGui);
```

```
private Runnable UpdateGui=new Runnable()
        {
                @Override
                public void run() {
                ……..
                }
        };
```

Sound recorder

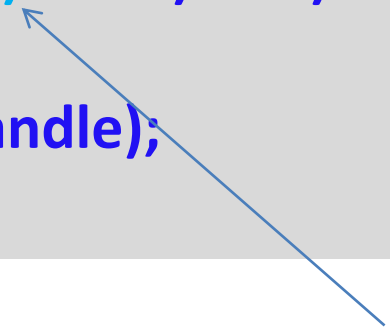
Birthday candle algorithm

1) Play Video
2) Start audio capture
3) Get audio buffer
4) IF Average(buffer) > Threshold then
   Stop Video
   END IF
4) Goto 2

Sound recorder

Birthday candle algorithm

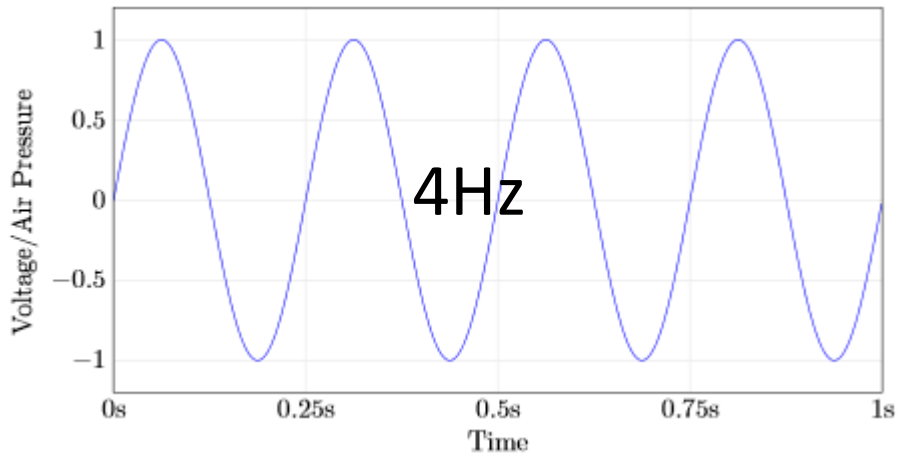
```
mSamplesRead = audioRecord.read(buffer, 0, buffersizebytes);
for(int i=0;i<(buffer.length);i++)
                        {
                              temp=temp+Math.abs(buffer[i]);
                        }

 if((((float)(temp/buffer.length)/32768f)*100f)>10f)
                        {
                              h.post(update_candle);
                        }
```

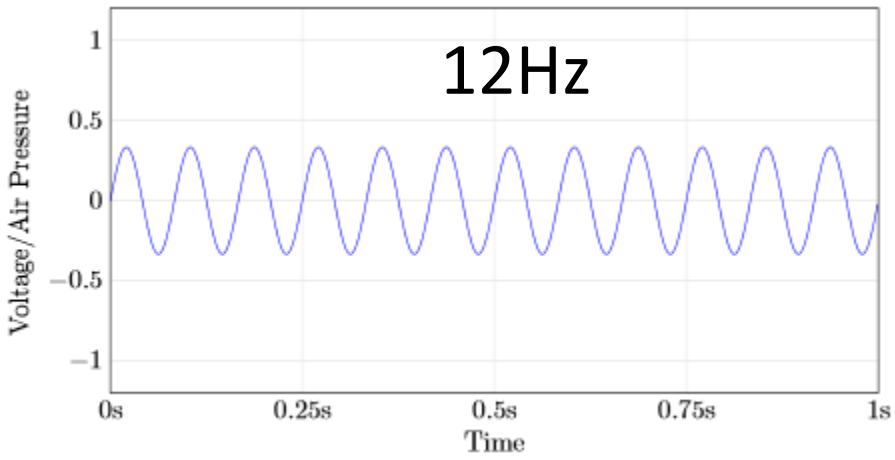


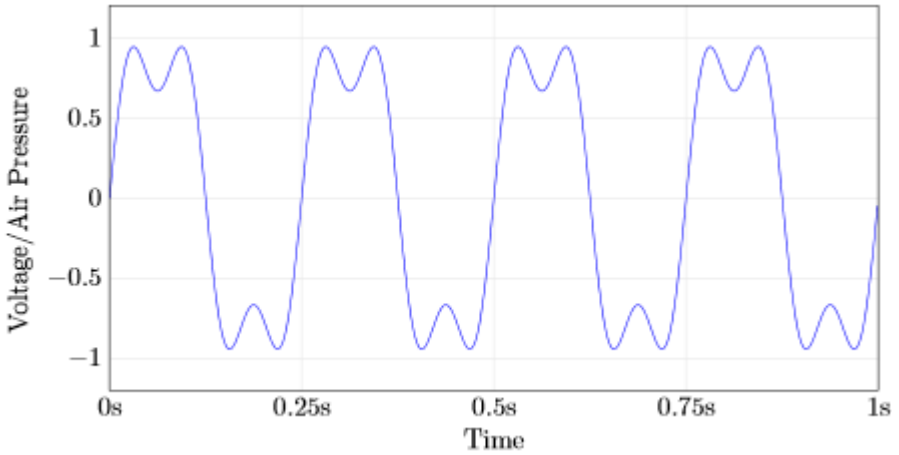
Sound recorder

Fast Fourier transform (FFT)

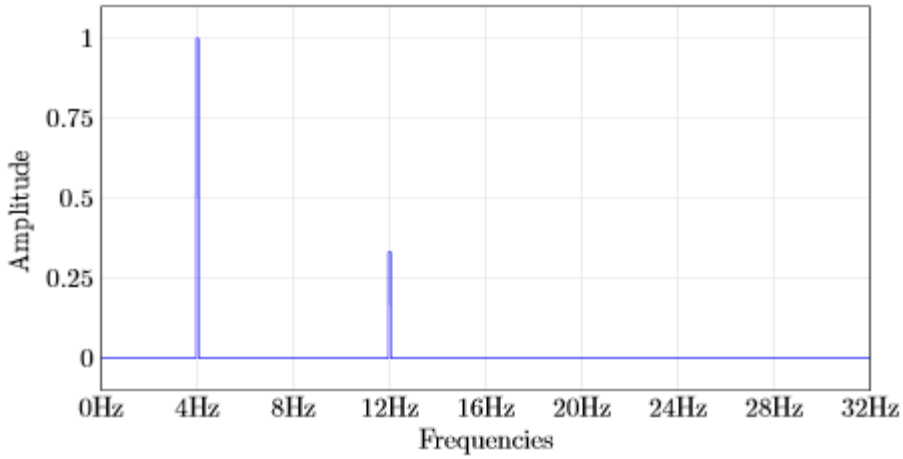
Sound recorder







4Hz+12Hz

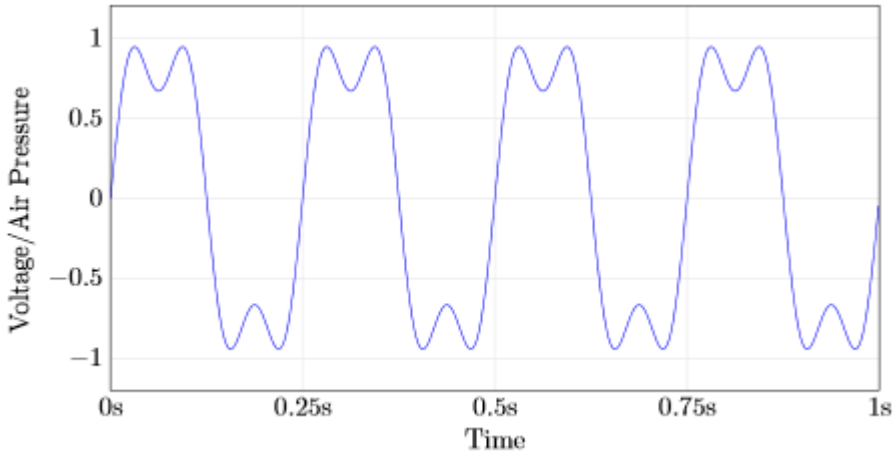


Sound recorder

Fourier's formula using sines and cosines

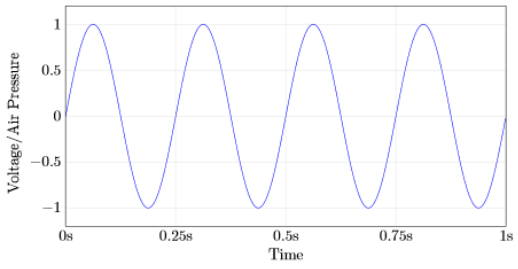
$$(S_N f)(x) = \frac{a_0}{2} + \sum_{n=1}^{N} [a_n \cos(nx) + b_n \sin(nx)], \quad N \geq 0.$$

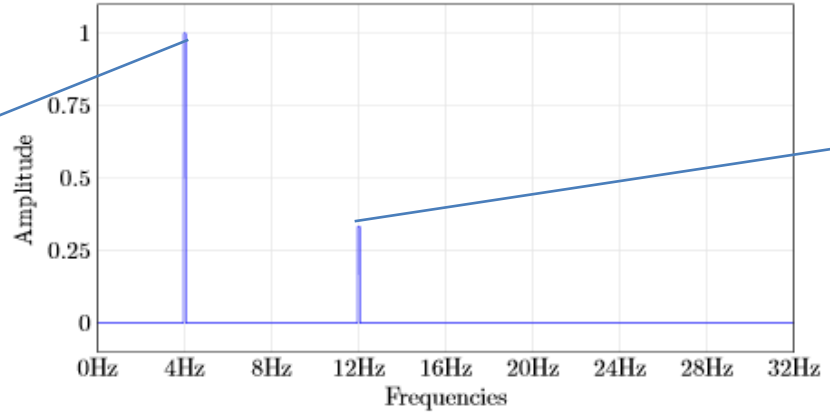
Sound recorder

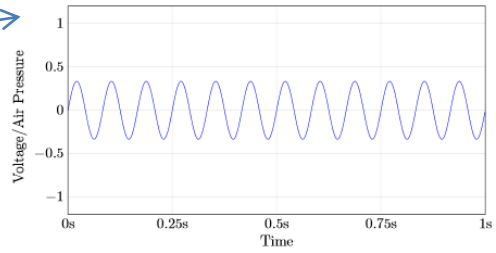


Discrete Fourier transform

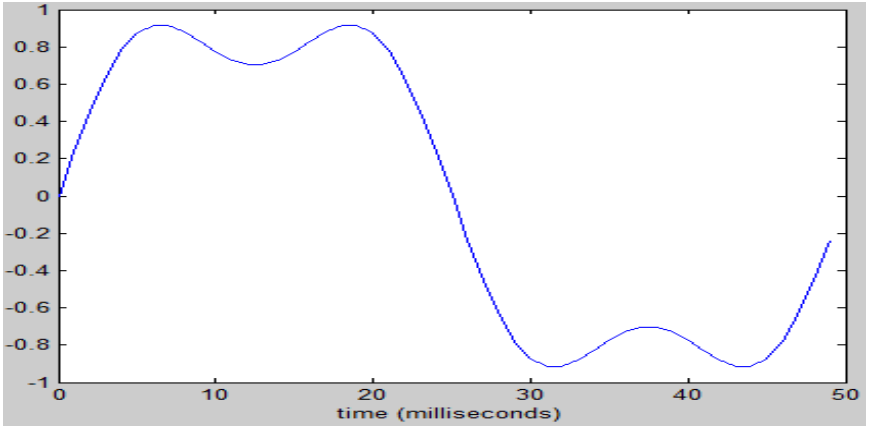
$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k \frac{n}{N}} \qquad k = 0, \ldots, N-1.$$





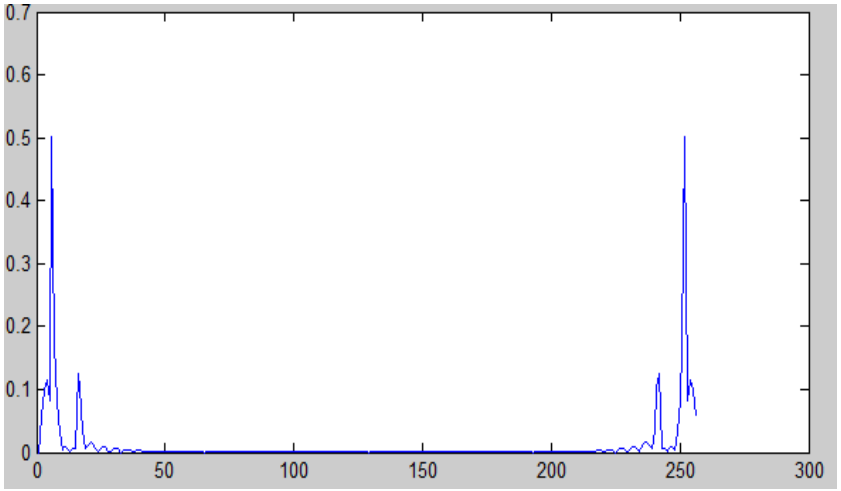


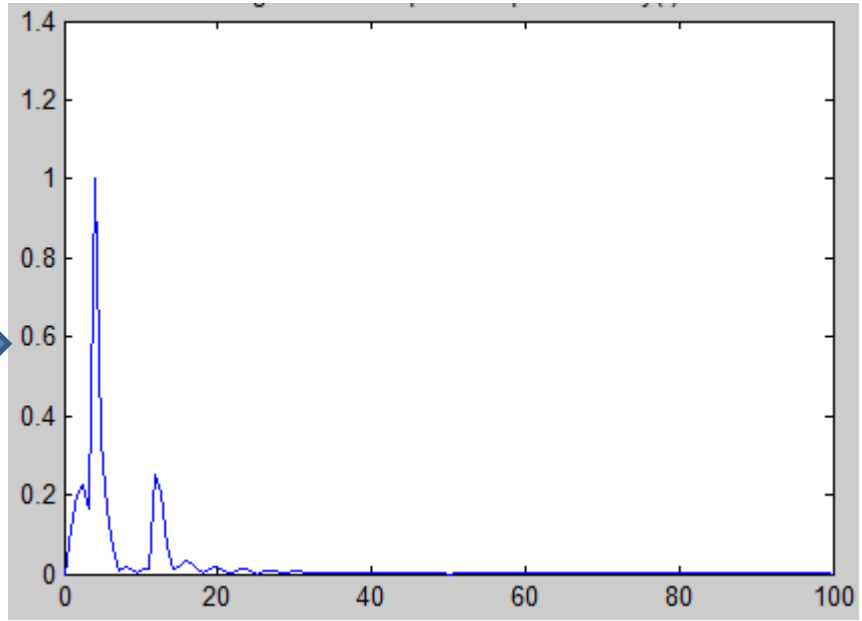
Sound recorder



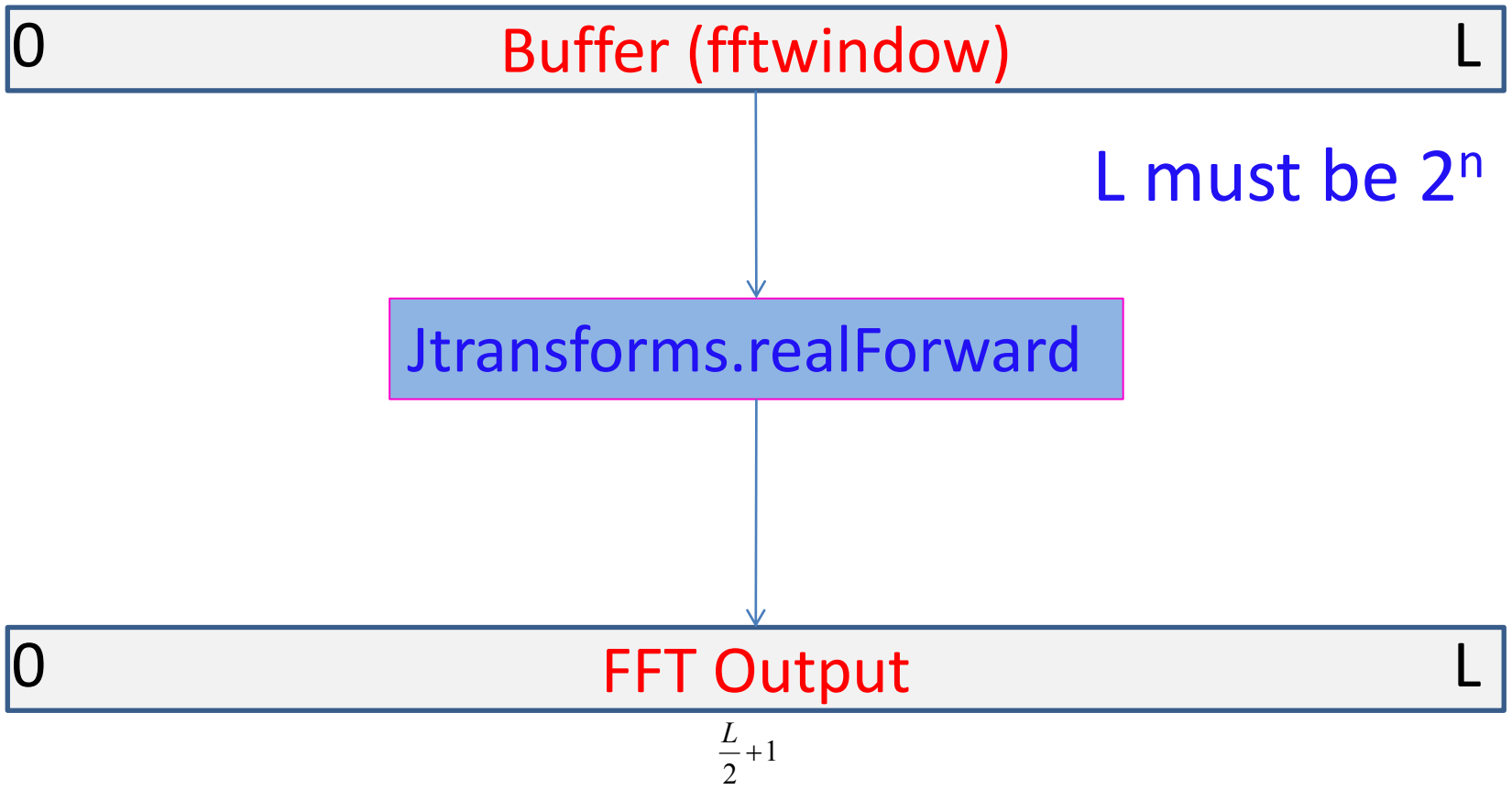
Jtransforms library

Fast Fourier transform





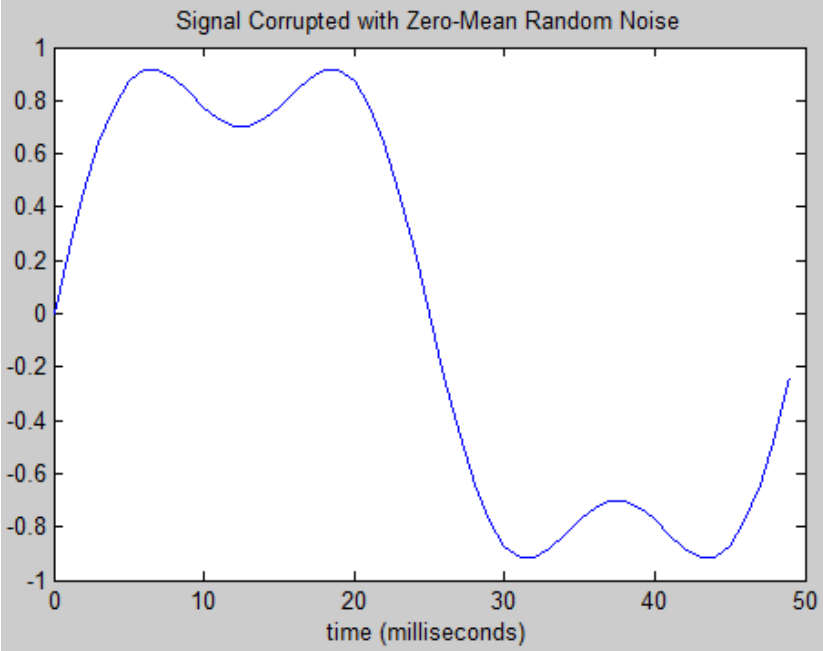
Sound recorder

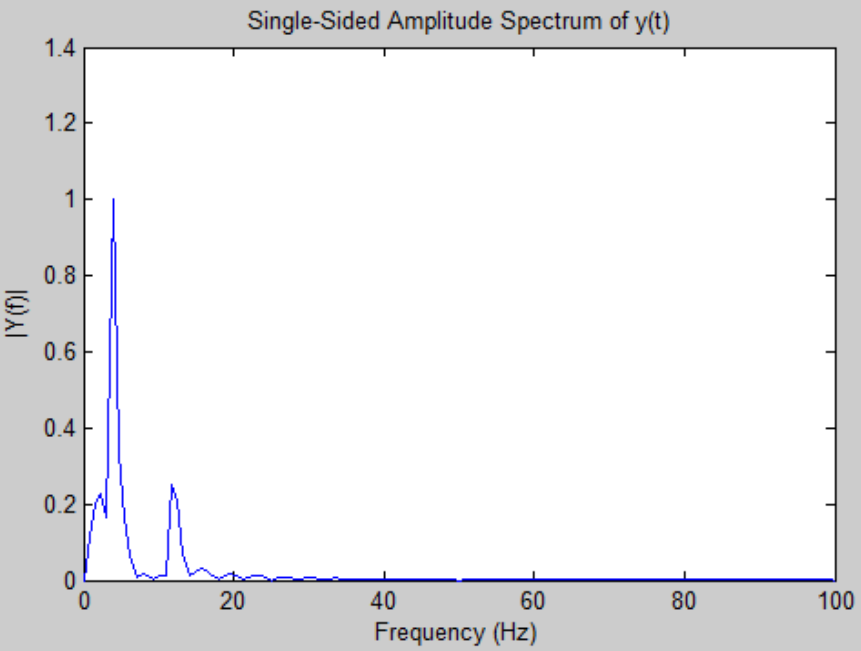


$$Amplitude = 2\left|fft[1:\frac{L}{2}+1]\right|$$

$$Frequency = \frac{SamplingRate}{2L}$$

Sound recorder





Sound recorder

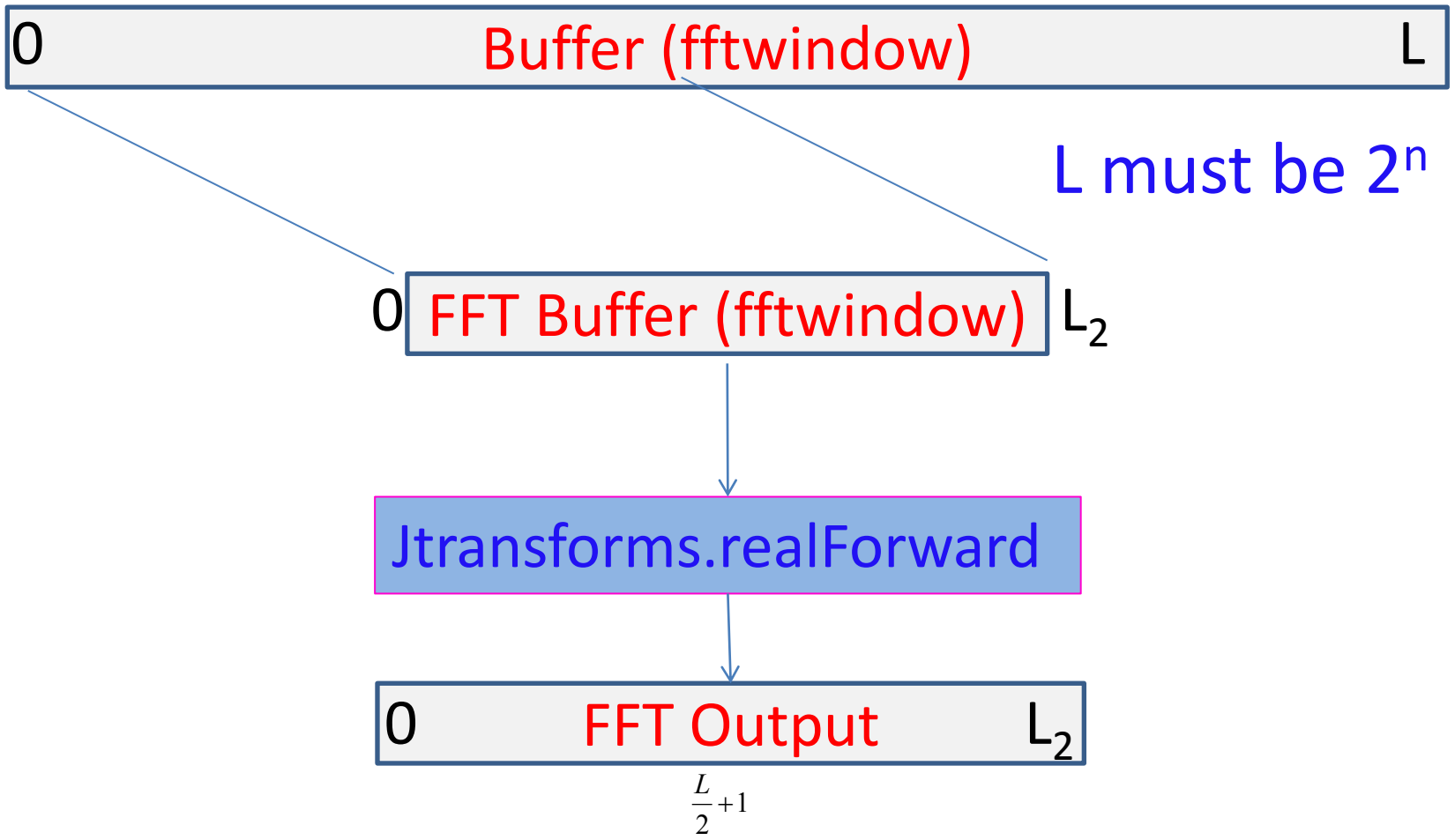
But minimum buffer size of **44.1k** 16bit mono is **8192** bytes

After do FFT we get result **4097** elements every 185ms

AChartEngine cannot handle that information

We will cheat!

Sound recorder



$$Amplitude = 2\left|fft[1:\frac{L_2}{2}+1]\right| \qquad Frequency = \frac{SamplingRate \div FFTWindow}{2}$$

Thank you ☺