FINAL PROJECT REPORT

CLAP DETECTION

DONE BY GROUP 8

CHAU HOANG TU

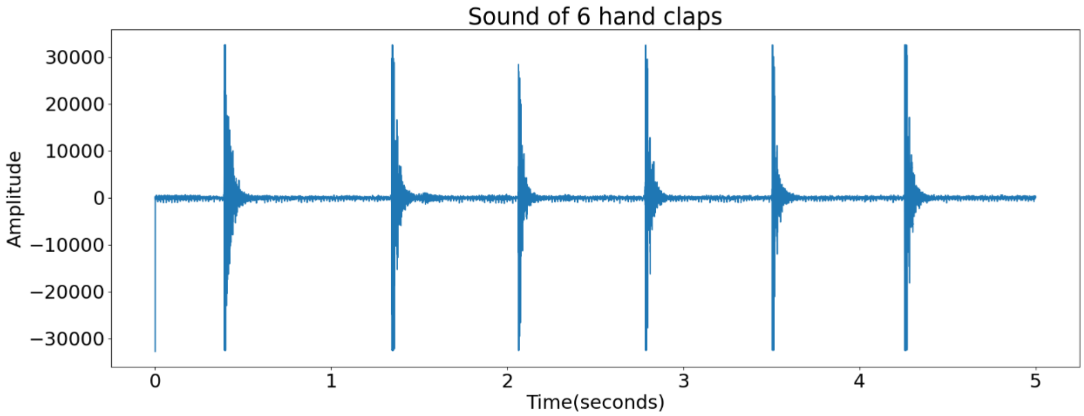
DAO MANH TUNG

TASK DISTRIBUTION

- Writing Fast Fourier Transform code: Dao Manh Tung.
- Writing report: Dao Manh Tung and Chau Hoang Tu
- Writing clap detection code: Chau Hoang Tu.
- Main Coordinator: Chau Hoang Tu.

INTRODUCTION

Our code can be used to be the solution for any application requiring claps to turn things on at our houses or even at companies as well. Things that can use this application are lights, TVs, fans or any other home facilities. This solution provides convenience, power efficiency and is the first step to obtaining smart home.

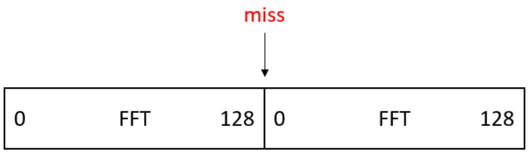


Algorithm:

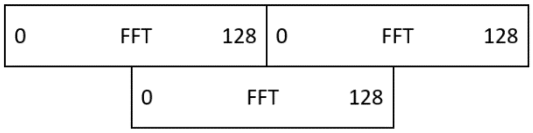
- FIrst step: sound is taken in 128 samples, with the frequency = 16000Hz, 16bit. Then, the absolute value of these samples is compared to a threshold value, to determine loud sound(above figure). If one of these values satisfies this condition, the flag of these samples will be true.
- Second step: if flag is true, FFT (Fast Fourier Transform) will be applied, and we find the index of peak value of FFT's calculation. If this index's value is equal to index of hand clap's maximum value, hand clap will be detected.

However, normal FFT cannot run on microcontrollers for real time applications. FFT call the cos and sin functions repeatedly, therefore we initialize array of cos and sin. It is certainly true that the time to access the array is always less than to call cos and sin functions. Time for FFT only consumes approximately 1ms (is measured by microcontrollers). Microcontrollers calculate FFT after 128 samples. Time for taking 128 samples consumes $128 \times \frac{1}{16000} = 8\ (ms)$, hence it is possible to be applied for clap detection in real time.

Note that: if we only take 128 samples for FFT, information which can be a clap will be missed.



Therefore, FFT must be applied for overlapping samples.



First step: The sound is recorded and store the value in array “data_second[]” and turn on the flag of this array. The data_second[] will be used to calculate the fast fourier transformation(FFT) when the index is 127.

```
while(1)
{
   SAMPLE =128;
   HALF_SAMPLE =64;
   if(time>INTERVAL_SAMPLE) //INTERVAL_SAMPLE: time to take 1 sample
   {
      if (index==count_sample )
      {
         /*if loop take enough 128 samples out loop */
         break;
      }

      */ reset time to 0 when take value from microphone */
      time =0;
      value_sensor= readValueSensor();
      /* store value of sensor in second array having 128 elements*/
      data_second[index]=value_sensor;
 if (abs(value_sensor)>AMP_THRESHOLD)
      {
         /* turn on flag*/
         flag_loud_second=1;
      }
//At this point, the value_sensor will be stored in the data_first[]. This array will store the value_sensor
between 2 loops. The first half of the array will store the value_sensor of the last half of loop 1 and the
second half of the array will store the value_sensor of the first half of loop 2. This array will be used to
calculate the FFT when the index is 63.
   if (index<HALF_SAMPLE)
      {


         /*store value of sensor in second half data_first */



         data_first[HALF_SAMPLE+index]=value_sensor;
         /*compare value of sensor with THRESHOLD to determine big loud */
         if (abs(value_sensor)>AMP_THRESHOLD)
         {
            /*turn on flag  */
```

```
                flag_loud_first =1;
            }
        }
        if (index>=HALF_SAMPLE)
        {
            /*compare value of sensor with THRESHOLD to determine big loud */
            if (abs(value_sensor)>AMP_THRESHOLD)
            {
                flag_loud_first=1;
            }

            /*store adc_val in first half data_re_first */
            data_first[index-HALF_SAMPLE]=value_sensor;

        }
```

Second step: If the flag for the is true, it will start calculating the FFT of the array and pick the largest value and then compare with the Frequency value of a hand clap. If they are equal then the light will switch its state.

```
        if (index==HALF_SAMPLE)
        {
            */ check big loud condition*/
            if (flag_loud_first==1)
            {
                */ calculate for fft data_first */
                fft(data_first);
                */ find index of the index of peak value of FFT’s calculation*/
                argmax_first=argmax_fft(data_first);
                if (argmax_first==FREQ_INDEX)
                {
                    */ when clap detect, toggle led */
                    change_LED_status();
                    }
                }
            }
            */reset flag */
            flag_loud_first=0;
        }
    index++;
    }
}
```

Reference:

https://www.geeksforgeeks.org/fast-fourier-transformation-poynomial-multiplication/

https://vnoi.info/wiki/translate/topcoder/Computational-Complexity-Section-1.md
https://vnoi.info/wiki/algo/trick/FFT.md

https://lloydrochester.com/post/c/example-fft/

https://pub.tik.ee.ethz.ch/students/2013-FS/GA-2013-03.pdf

https://www.eecs.yorku.ca/course_archive/2019-20/W/1011/labs/4/distinguishing-between-claps-and-other-noises.html