

Hudobný vizualizátor

V tomto dokumente prezentujem zapojenie, umožňujúce vizualizáciu hudby. Zapojenie využíva 8×32 LED maticu a Arduino. Zapojenie je pripojené ku prehrávaču 4-pinovým jack adaptérom. Pravý a ľavý kanál je zmiešaný, ku signálu je primiešaný jednosmerný prúd a signál je privedený do Arduina. Arduino je naprogramované aby na privedenom signále vykonávalo rýchlu fourierovú transformáciu a výsledok je vyvedený na LED maticu. Ku jack prívodu je pripojený druhé jack rozhranie umožňujúce výstup signálu.

Zapojenie

Aby neboli problémy s digitalizáciou signálu v Arduine, ku signálu je pripojený jednosmerný prúd tak, aby nulová hodnota signálu bola v strede možných hodnôt.

Výstup $3.3V$ je pripojený na ochranný rezistor. Potom sa tento prúd rozvetví a jedna vetva je pripojená na pin AREF na Arduine. Toto napätie je vonkajšia referencia. V druhej vetve je napätie dvoma rezistormi rozdelené a primiešané ku zvukovému signálu. Pri digitalizácií sa zvukový signál premení na hodnoty medzi 0 a 1023. Na tieto hodnoty je namapované napätie medzi 0 a AREF voltami. Keďže ku zvukovému signálu je privedená polovica napätia AREF, nula zvukového signálu sa po zmiešaní namapuje na hodnotu 512, takže zvukový signál sa de facto namapuje na hodnoty ± 512 .

V zapojení je použité jack rozhranie typu SRRT (socket, ring, ring, tip). Socket je uzemnenie, prvý ring je kontakt pre mikrofón. Druhý ring a tip zodpovedajú pravému a ľavému kanálu. Druhý ring a tip sú pred zmiešaním prevedené rezistorom a kondenzátorom, na ochranu pred privedeným DC zmiešavacím napätím. Pravý a ľavý kanál sú zmiešané, spolu s rozdeleným DC napätím. Tento signál je pripojený na A0 pin. Na Arduino je napojená LED matica. Tá má päť kontaktov. Napájací pin $5V$, zem a 3 piny ovládajúce maticu: DIN (digital in), CS (chip select) a CLK (clock).

Arduino

V Arduine prebieha niekoľko dejov: analóg-digitálna konverzia, rýchla Fourierova transformácia a odosielanie výstupu (Serial Peripheral Interface)

Analóg-digitálna konverzia je v Arduine vstavanou funkciou. Je však potrebné nastaviť hodnoty registrov ADCSRA a ADMUX. ADCSRA je potrebné nastaviť na hodnotu $0b11100101$. Toto nastaví konverter na free-running mód, čiže konverter bude bežať kontinuálne. Navyše sa týmto nastavením nastaví prescaler na 32. Arduino beží na frekvencii $16MHz$. Každá konverzia trvá 13 cyklov, čiže výsledne bude ADC bežať na frekvencii $38.46kHz$. ADC je nastavený aby používal vonkajšiu referenciu nastavením registra ADMUX na hodnotu $0b00000000$. Nyquistova veta nám hovorí, že takto

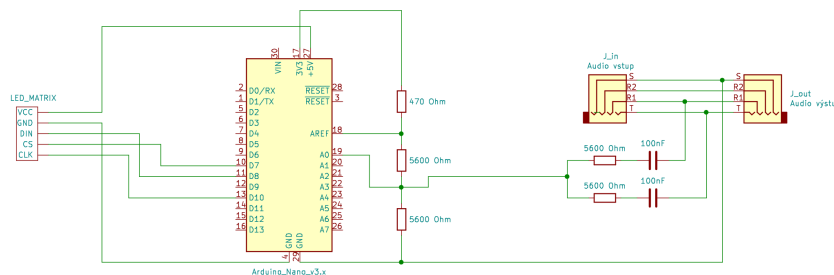
nastavený konverter bude vedieť vierohodne (bez aliasingu) digitalizovať signál s frekvenciou až do $19kHz$ (polovica zo vzorkovacej frekvencie).

Rýchla Fourierova transformácia má v Arduine samostatnú knižnicu. Signál je najskôr navzorkovaný, zo signálu je odobraná jednosmerná zložka, následne je aplikované Hammingovo okno a vzorka je transformovaná. Výstup je vhodne namapovaný a je zobrazený na LED matici.

Problémy

Najväčším problémom je kontakt na jacku. Jack adaptér ktorý používam nie je najvhodnejší na pripojenie na nepájivom poli a jednoducho stráca kontakt. Ďalší problém, ktorý sa vyskytol bolo nepresné rozdelenie napätia primiešaného do signálu. Predpokladal som presné rozdelenie na polovicu, no na LED matici sa stále zobrazoval, aj pri vypnutom zvuku, vrchol na najnižšom stĺpci. Tento problém bol odstránený, keď som v FFT knižnici našiel a použil funkciu `DCRemoval()`.

Schéma zapojenia



Zdroje

Inšpirácia na projekt aj zdroj veľkej väčšiny kódu bol <https://github.com/shajeebtm/Arduino-audio-spectrum-visualizer-analyzer/>

Zdroje pre ADC konverziu:

<https://onlinedocs.microchip.com/pr/GUID-80B1922D-872B-40C8-A8A5-0CBE009FD908-en-US-3/GUID-80858615-729B-4FC6-97DF-4755FEF75CEB.html>

<http://maxembedded.com/2011/06/the-adc-of-the-avr/>

Hudba vo videu: Grimes - World Princess part II