

情報科学実験B 課題3レポート

氏名	山久保孝亮
所属	大阪大学基礎工学部情報科学科ソフトウェア科学コース
メールアドレス	u327468b@ecs.osaka-u.ac.jp
学籍番号	09B22084
提出日	2024 年 6 月 18 日
担当教員	松本真佑, 小南大智

1 システムの仕様

今回作成した課題 3E ではプッシュスイッチを開放した時にきらきら星を演奏する。また、プッシュスイッチを押下するまでは無音である。演奏が終了すると 1 秒間音を鳴らさず、その後再び演奏を再開することを繰り返す。また、すべての音と音の間に休止を挟むようにした。これによって同じ音が連続してしまった際に音の切れ目がわからなくなってしまうことを回避している。

2 ハードウェア回路

ハードウェア回路は以下の図 1 のとおりである。

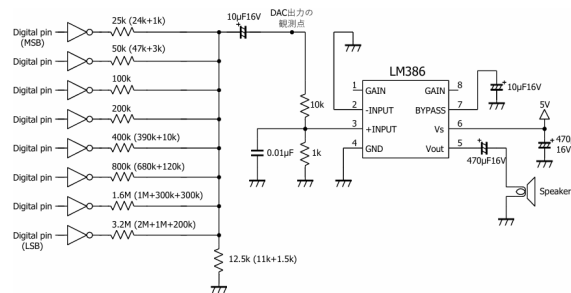


図 1: D/A コンバータとアンプの回路図

上図左部の Digital pin の LSB から MSB までを Arduino の D2 から D9 に差し込んだ。プッシュスイッチに関しては課題 1 のチャタリング防止回路を利用した。[1] チャタリング防止回路を採用した理由は、プッシュスイッチの開放のタイミングを正確に把握するためである。

3 制御プログラム

制御プログラムは setup 関数, loop 関数, ISR に大きく分けられる。

setup 関数

ここでは初期設定を行った。初期状態では音を鳴らさないようにしなければならないので、TIMSK1 の OCIE1A を 0 に設定することで割り込みが発生しないようにした。また、OCR1A が最初に鳴らす音である scale[0] に設定した。scale は演奏する曲の音階が入力された配列であり、休止の際の値は 655

loop 関数

ここでは音階と音を出力する長さの制御を行った。具体的には以下のフローチャートに従って処理が実行される。以下ではスイッチが解放されたときのループの処理とそれ以降のループの処理を記述する。スイッチが解放されると、offbutton 関数を最初に実行し、change というフラグを使用することで非トラッキング処理でスイッチの開放を検出する。offbutton 関数はグローバル変数として前回のループのボタンの状態を記憶して起き、直前は押していたが現在は押していないときにフラグを 1 にする関数である。また、フラグが立っていれば割り込みを有効にする。これによってスイッチを開放した時に音を鳴るようになる。次に millis 関数を使って現在の時間を取得し初期値が 0 である変数 now との差が length[0] の値以上かどうかを判定する。length はそれぞれの音の長さを格納する配列である。この条件式が満たされていない間は OCR1A が初期値の scale[0] に設定されるので 最初に指定している音階の音が鳴り続ける。条件式が満たされた場合、now の値がその時の millis 関数の値に設定され初期値が 0 の count の値が 1 インクリメントされ、OCR1A が初期値の scale[1] に設定される。それ以降のループ

では新たに設定された now と millis 関数の値の差が length[count] 以上になるかを判定して OCR1A を scale[count] に設定し, count をインクリメントしていく作業を count が 84 になるまで繰り返す. 84 というのは音階と休止をすべて合わせた数である. 単純な音階と最後の休止を実現するのであれば 43 個で十分であったが, システムの仕様で記述した通り, それぞれの音の間に短い休止をいれることで音の切れ目を表現するために採用した. 84 になれば OCR1A を scale[0] に設定することで繰り返し楽曲を演奏する機能を実現した.

4 課題 3A の理想値の計算

図 1 の D/A コンバータとアンプの回路図における左側の 8 個の抵抗は並列回路であるので, これらの合成抵抗 R の計算式は以下になる.

$$\frac{1}{R} = \frac{1}{25k} + \frac{1}{50k} + \frac{1}{100k} + \frac{1}{200k} + \frac{1}{400k} + \frac{1}{800k} + \frac{1}{1.6M} + \frac{1}{3.2M} = \frac{51}{640000} \quad (1)$$

即ち R の値は,

$$R = \frac{640000}{51} \quad (2)$$

また, 図 1 の $12.5k \Omega$ の抵抗は分圧回路と見なせるので [1], のこぎり波の最大電圧である V_{out} は,

$$V_{out} = \frac{R}{R + 12.5k} * 5 = \quad (3)$$

5 工夫点

今回私が工夫した点としては, 曲の演奏情報を配列に格納し, loop 関数の構造をシンプルにした点である. この工夫の利点としては, 配列の音階や音の長さの情報を変更してもプログラムが動作するので今回実装したきらきら星以外の曲を演奏しようとする場合配列を変更するだけでよいということである.

6 考察

今回の課題を通して私が考察したのは, Arduino を使って和音を出力することができないかということである. 和音について考えた理由は, 楽曲を演奏するにあたって和音を使用することで表現の範囲が広がるためである. 最初に私は複数のスピーカと fork() のようなものを使用し複数のスピーカで異なる音階を同期して鳴らすことで和音を実現できるのではないかと考えた.

7 感想

今回の課題ではハードウェアの問題なのか, ソフトウェアの問題なのかを見極めることが非常に難しかった. 問題を分割して個々の問題の状態を観察することができるのかを考える力は授業内の課題に限らず, 生きていく上でも非常に有用な能力であると感じた.

8 謝辞

動作確認や質問対応をしてくださった教授及び TA の皆様, 有難うございました. 引き続き課題 4 でも宜しくお願いします.

参考文献

- [1] 2024 年度情報科学実験 B 指導書