

1 目的

H8 マイコンの割り込みについて理解を深め、割り込み制御を C 言語で記述する方法を学ぶ。本実験では、発展的な課題として「UFO ゲーム」作成に取り組み、キー入力処理と効果音処理を割り込みを用いて実現する。

2 原理

2.1 UFO ゲームの概要

本実験の課題である「UFO ゲーム」は、1980 年代にゲーム電卓としてカシオ計算機から発売された MG-880 の「デジタルインベーダー」が元となっている。

「UFO ゲーム」では、図 1 のようにキャラクタを使ってゲームフィールドを表現する。フィールドは、「:」によって分けられた照準フィールドと戦闘フィールドの 2 つから構成される。

照準フィールドには、敵を打ち落とすための数値が表示され、照準キーを押すたびに「... → 0 → 1 → 2 → ... → 8 → 9 → n → 0 → ...」のように順に変化する。戦闘フィールドには、フィールドの右側から左に向って敵が押し寄せてくる。押し寄せてくる速さは、ゲームが進行するにつれて徐々に速くなる。敵はインベーダーを表す数字の 0~ 9 と、UFO を示す n で表現されている。なお、インベーダーは進行するたびに戦闘フィールドの右端に乱数で発生し、UFO は特定の条件下で発生する。

敵の撃墜は、照準フィールドの値を敵を示す値と一致させて発射キーを押すことで行われるが、戦闘フィールドに複数の同じ値をもつ敵が存在する場合には最も照準フィールド寄りの敵 1 つのみが撃墜される。

撃墜された敵は消滅し、撃墜された敵よりも照準側にある敵は右側へ退く。敵の左端が「:」まで到達したときに占領されて、ゲーム終了となる。ゲームの完成見本として、UFO_game.mot が後述の実験用のディレクトリにあるので、各自で検証すること。

2.2 キー状態の判別

今回の実験課題では、キーを押すごとに照準の値を順に変化させなければならないため、キー状態の判別は重要な処理となる。一般に、機械的な接点をもつスイッチの特性として、チャタリング現象が生じる。チャタリング現象は、接点が切り替え時に振動することによって生じ、短い時間 (数 ms から数 10ms 程度) に ON と OFF を繰り返す現象である。その値をそのままマイコンで取り込むと、図 2 のように、一度だけ押したはずが複数回押したように検出されるため、チャタリングの除去が必要となる。

チャタリングの除去にはハードウェア処理による手法とソフトウェア処理による方法があるが、実験ボードのようにキーがマトリクスで実装されている場合はハードウェア処理ではコストがかかるため、一般的にはソフトウェア処理で行われることが多い。

チャタリング処理のアルゴリズムの原理は単純で、入力信号履歴を参照して過去一定期間内の値が全て同じ時に、その値を参照時の値とするものである。一定期間内の値が全て一致しない場合はキー状態が遷移中であると考え、アプリ側で適切に処理する必要がある。

2.3 リングバッファ

キー状態の判別には一定期間内の信号履歴が必要となるため、キーをセンシングするたびに値を保存しておく必要がある。

リングバッファはこのような用途によく用いられ、図3のように、通常の配列のような一次元状のバッファの最初と最後をつなぎ合わせることでリング状のバッファを形成する。すなわち、リングバッファの大きさが過去の履歴を保存しておく大きさとなる。

リングバッファのどの位置に最新のデータが入っているかを示すためにポインタを用意し、データの更新を行うたびにポインタも更新する。このとき、リングバッファの実現に配列を利用する場合、ポインタの更新にあたっては配列の最後の次は配列の最初になることに留意しなければならない。このことは、過去の履歴の参照の際にも必要となるので注意すること。

2.4 効果音の生成

効果音を発生させる方法はいくつかあるが、ここでは単純な一定音程で一定の期間の音を生成する方法について考える。一定音程の単音を発生させる最も簡単な方法は、図4のように、矩形波を生成させる方法である。矩形波の生成は、一定期間ごとに出力値を変化させることで実現できる。

このとき、単音のみの生成に専念するのであれば一定期間をムダ時間ループで実現すればよいが、本課題では音を生成することによってゲームの進行を妨げることができない。このため、割り込みを音程の半周期ごとにかけることで出力値の変化のタイミングをはかることで、ムダ時間の弊害を取り除くように工夫する。

また、効果音の長さは音長を割り込みの間隔で割ることで、指定時間となる割り込みの回数を知ることで制御を行う。

なお、効果音生成用の割り込み間隔は音程によって変化するため、他のキー処理等の常に一定間隔の割り込みと共用することはできない。このため、効果音生成用に専用のタイマを準備して割り込み制御を行うことが必要である。

3 使用機器

1. H8 マイコン
2. USB ケーブル
3. パーソナルコンピュータ

4 実験方法

本課題では、内蔵されているタイマ 2 つを割り込み発生源として使用する。タイマ 0 を効果音生成専用のタイマとして割り込み優先度をプライオリティ1(高優先度)として用いる。プライオリティ1の優先度を使用するためには、UI ビットの設定が必要であり、`USE_UI();` をプログラムの最初に行う必要がある。また、タイマ 0 をプライオリティ1に設定するためには、更に `timer_pri_set(ch, pri)` で設定を行う必要がある (ch:タイマのチャンネル番号、pri:プライオリティ値)。割り込みハンドラは `int_imia0()` であり、`sound.c` 内に記述する。

タイマ 1 は常に一定間隔 (1ms) で割り込みを発生し、キーのセンシング・敵の進行速度・時間待ち等の処理を行うものとする。割り込みハンドラは `int_imia1()` であり、`ufo.c` に記述する。

課題にあたっては、`/home/class/j3/jikken/kouki/no4` 以下に課題に必要なファイルを置いてあるので、必ず全てのファイルを各自でコピーして利用すること。

4.1 課題 1 キー読み込みの実装

コピーしたファイル (特に、`ufo.c`, `key.c`) をよく読み、`key.c` に記述されている各関数を完成させなさい。

`key_sense()` はタイマ 1 の割り込みごとに呼び出され、リングバッファにキーマトリクスの列ごとのスキャンデータを格納するように作成すること。リングバッファは `key.c` で大域変数として宣言されている `keybuf[p][r]` であり、`p` は参照ポインタ、`r` は列を表すものとする。また、割り込み処理をまたぐリングバッファのポインタは `keybufdp` を用いること。なお、キースキャンの方法等については、最後のページにある補足のキーマトリクスとセンシングを参照のこと。

`key_check(keynum)` はチャタリング除去を行った結果を返す。その戻り値は `key.c` で定義されている `KEYOFF`(指定キーは押されていない)、`KEYON`(指定キーが押されている)、`KEYTRANS`(指定キーは変化中)、`KEYNONE`(指定キーは存在しない) のいずれかを正しく返すものとする。また、チャタリング除

去で参照するバッファ上の範囲は、key.c で定義されている KEYCHKCOUNT の長さとする。課題ができれば、必ずその時点でコンパイル・実行して動作を確認し、プログラムと動作のチェックを受けること。チェックを受けずに先の課題を行うことは認めない。

4.2 課題 2 効果音生成の実装

コピーしたファイル (特に、ufo.c, sound.c) をよく読み、sound.c に記述されている各関数を完成させなさい。

sound_beep(hz, msec, vol) は引数 (hz:音程周波数で単位は [Hz], sec:音長で単位は [ms], vol:振幅で矩形波の D/A 上限値) から計算して、音程のための割り込み間隔・音長のための割り込み回数・矩形波の振幅を求めて必要な記述を加えること。

sound.c で宣言されている大域変数 timer0_count は割り込み回数を格納し、play_count は指定音長に対する割り込み回数を格納 da_amp は指定振幅を格納するように、それぞれ準備されている。-課題ができれば、必ずその時点でコンパイル・実行して動作を確認し、プログラムと動作のチェックを受けること、チェックを受けずに先の課題を行うことは認めない。

4.3 課題 3 UFO ゲームの発展

以下の課題に取り組み、UFO ゲームを更に発展させる。細部の仕様は適宜定めてよいが、レポートに記述すること。

・課題 3-1

自機数を導入し、占領された場合と、発射キーを押したときに照準値と同じ値の敵がいらない場合 (攻撃に失敗した場合) に自機数が減るようにして、自機数が 0 になったときにゲームオーバーとなるように変更する。

・課題 3-2

出現する敵数と発射できる弾数を制限し、敵を全部撃破したら面クリアとする。面が進むにつれて条件を厳しくし、占領されたときにゲームオーバーとなるように変更する。

5 実験結果

6 検討課題

・ 検討課題 1

ufo.c をよく読んで、どのようにして「今、キーが押された」かを検出しているかについて、説明しなさい。

1 ループ前のキーの状態を保持し、現在のキーの状態と比較することにより実現されている。1 ループ前に押されていない、かつ現在押されていればキーは今押されたことになる。

・ 検討課題 2

補足の図 5 キーマトリクス回路において、キースイッチと直列にダイオードが挿入されていないとどのような不具合が生ずるか、説明しなさい。

・ 検討課題 3

sound.c で実装したような割り込み処理による効果音生成において、効果音の音程が高くなると、どのような悪影響が生じてくるかについて検討せよ。

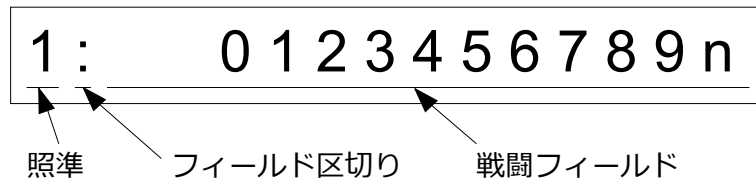


図 1: UFO ゲームのフィールド

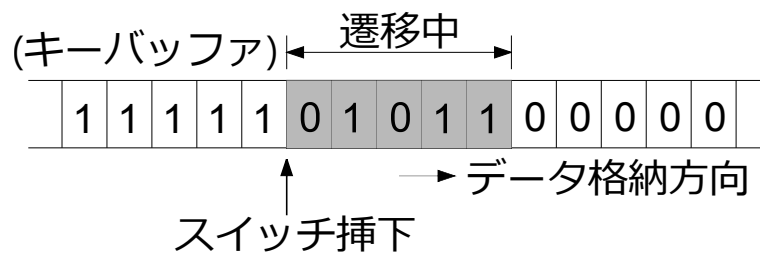


図 2: チャタリング発生時のデータバッファの様子

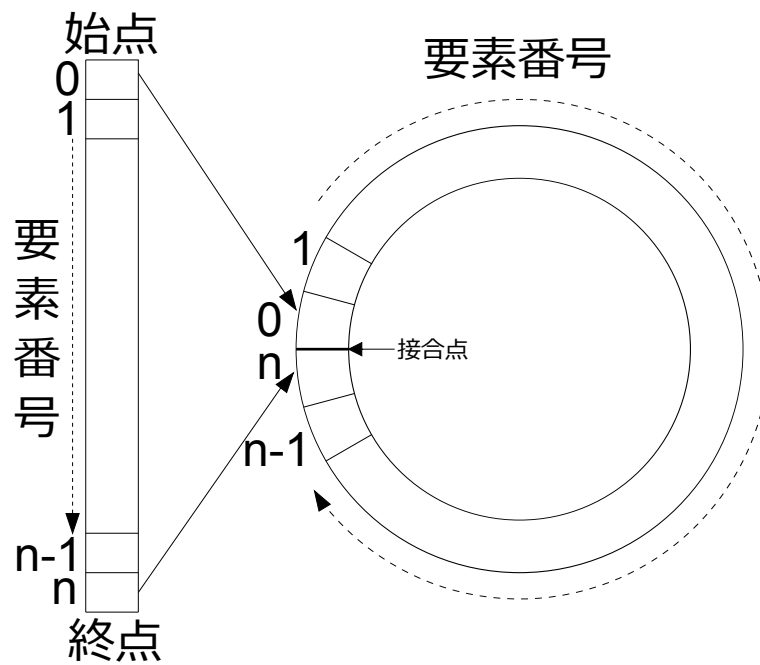
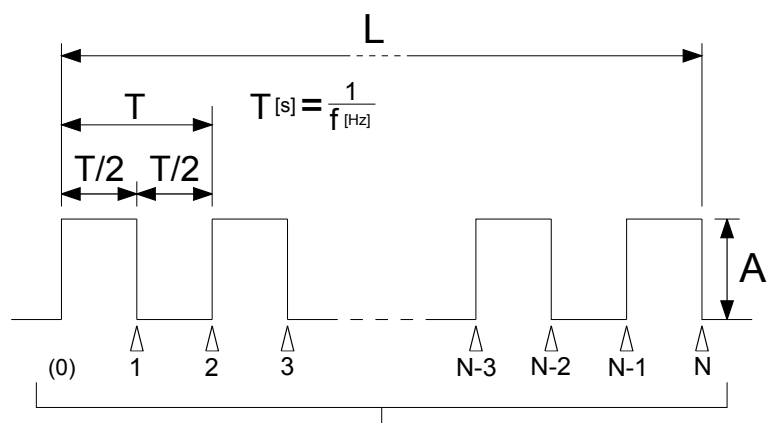


図 3: リングバッファの構成



割り込みタイミング(Δ)と回数 $N = \frac{L[s]}{T/2[s]}$
 (L:音長, f:音程周波数, N:割り込み回数, A:振幅)

図 4: 効果音の波形

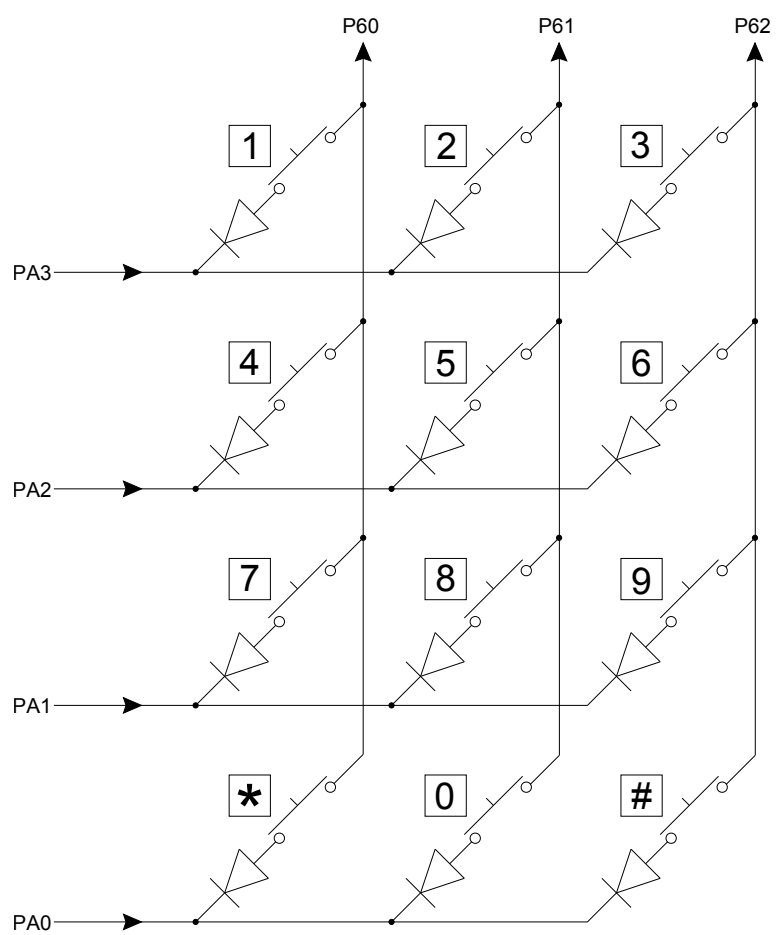


図 5: キーマトリクス回路図