光センサによる駒位置検出を用いたチェスの棋譜の自動記録システム

真壁　純矢*・竹下　大樹

Automatic Recording System of the Chess Score Sheet using Piece Position Detection with the Optical Sensor

Makabe JYUNYA* and Daiki TAKESHITA

（平成 26 年 10 月 9 日受理）

In an official competition, the chess players are required to record their score sheets themselves during the game itself. In this research, we aim at relieving the player's workload by building an automatic recording system for the chess score sheet. The chessboard which I used in this study detects the removal or placement of a piece in any of the squares through the optical sensor. Once the turn is played, at the push a button, the chessboard sends all the data—including the position and the number of transition states in the piece's presence and absence in the square —to the PC. Moreover, it includes the analysis of a player's play such as a promotion and a resign. The PC then analyses the received data and displays the chess score sheet of the current turn and the time limit in its GUI. This information is sent back to the chessboard, and its LCD displays it. In addition, if the movement is not right, the LCD displays its fact.

1 諸言

チェスは非常に古い歴史を持ち，全世界 160 ヶ国以上で楽しまれ，少なくとも 6 億 500 万人から 7 億人以上の競技人口を誇る．このように競技人口が多いチェスだが，誰もが試合の記録の仕方を知っているわけではなく，また，国際チェス連盟またはその下部組織が公認している試合では，プレイヤーは一手一手の駒の動きを棋譜として記録しなければならないという決まりがある[1][2].

本研究では駒位置を検出する上で，回路の小型化が可能であり，ヒステリシス（応差）が小さく，外乱光許容照度が大きいといったメリットがあるフォトリフレクタ（光センサ）を光変調型フォト IC と赤外線 LED を用いて構成し（図 1.1，図 1.2 を参照），駒位置の検出を行うことにした．本研究の目的は光センサと H8 マイコンをチェスボードに組み込むことでパソコンと通信し，作製したチェスボードから得られるデータを基に行う駒位置の追跡により棋譜の生成を行い，プレイヤーに対して有益な情報を示すことでチェスの支援をすることである.

本研究で用いたチェスボード及びに動作内容については，以降の章にて記述する.

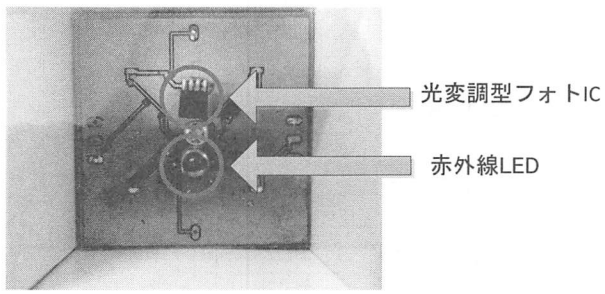


図 1.1 作製したフォトリフレクタ

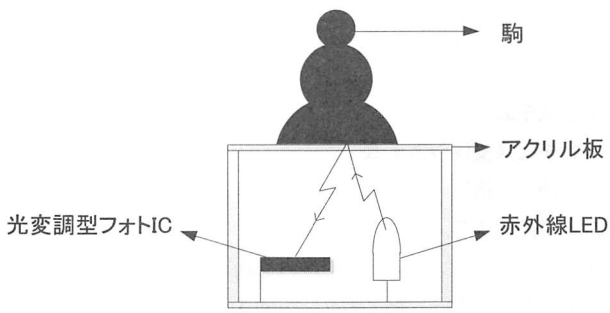


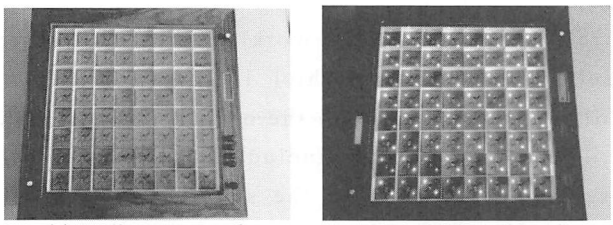
図 1.2 フォトリフレクタの動作の様子

2. チェスボード

*秋田高専専攻科学生

本研究で用いたチェスボードの電源を切った時と入れた時の様子を図2.1 (a), (b)に示し，外部の構成を図2.2に示す．このチェスボードでは明色と暗色の 2 色のマスの区別を白色と青色のLEDで行い，棋譜やタイマー等の情報表示はLCDで行った．また，駒位置情報の取得は8×8マス分の光センサの検出によって行い，リザイン，プロモーション，ターンの終了といったプレイヤーの意思表示はプッシュスイッチにて行えるようにした．

このチェスボードは LED, LCD, 光センサといった半導体素子を主として構成した．これらの半導体素子は軽量，長寿命，低消費電力，様々な大きさのものが現存するといった特徴を持つ．また，H8 マイコンを用いての制御方法について記述した資料は多く存在するため，システム開発が容易である．



(a) 電源を切った時　(b) 電源を入れた時

図2.1 本研究で使用したチェスボード

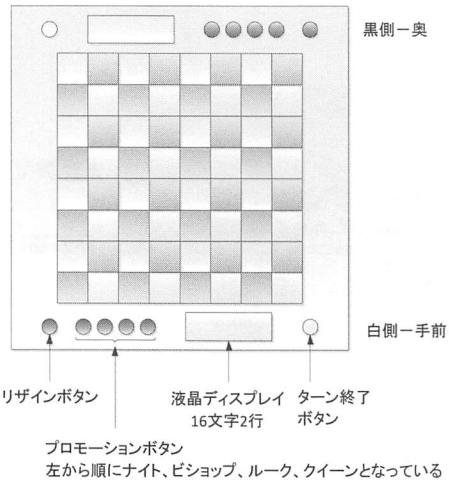


図2.2 チェスボードの外部の構成

3. システム

3.1. チェス支援システム

本研究で構築したチェス支援システムの動作及び支援内容を以下に記述する．

① まず，チェスボード内で駒の移動等のプレイヤーの操作が行われた後，ターン終了ボタンが押されることにより，ターン終了ボタンが押されるまでに行われたデータがPCへと転送する（図3.1を参照）．

② チェスボードから送られたデータを元にプレイヤーが行った操作を判断し，プレイヤーの操作が正しい場合，駒位置，棋譜，各プレイヤーの残り時間をGUIにて表示し(図3.2を参照)，LCDには棋譜と各プレイヤーの残り時間のみの表示を行う（図3.3を参照）．ここで，プレイヤーの操作がチェスのルールに準拠されていないと判断された場合はLCDにその旨を表示し，駒を元の位置に戻してから，再度操作を行ってもらう（図3.4を参照）．

③ ここまでで問題が無ければ，次の操作以降同じ動作を行う．

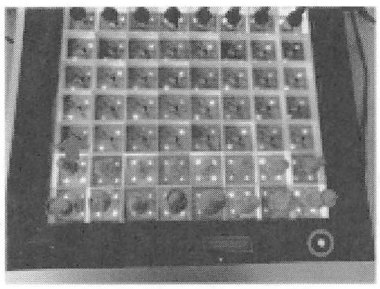


図3.1 プレイヤーの操作

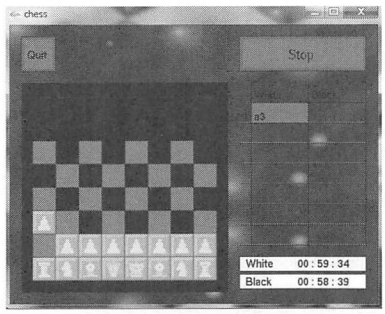


図3.2 GUI

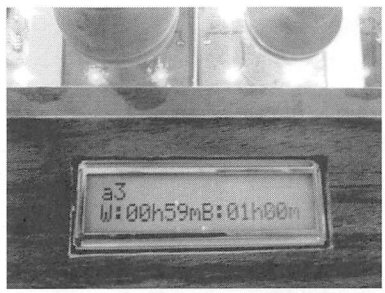


図3.3 エラーが無い時のLCDの様子

図3.4 エラーが有る時のLCDの様子

3.2. 通信データ

本研究のプログラムでは通信に扱うデータの型をchar型としたため，通信データは最低でも1バイトから扱うこととなる．マイコンからパソコンへの通信データを6バイト固定とし，パソコンからマイコンへの通信データを33バイト固定とした．この章では本研究におけるパソコンとマイコンの通信データについて記述する．

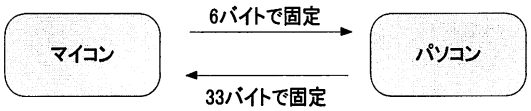


図3.5　パソコンとマイコンの通信データ

3.2.1. パソコンからマイコンへの通信データ

パソコンからマイコンへの33バイトの通信データは，以下のように扱った．なお，本研究で用いたLCDで扱える1文字あたりの縦と横の比は2対1であり，半角文字となっている．

(i)プレイヤーの操作の正誤情報:1バイト

プレイヤーの操作が正しいものであったかを示すデータ．正しいものであった場合にはデータを $(0)_{10}$ とし，誤りがあった場合には $(1)_{10}$ として扱った．本来は1ビットで十分であるが，最小のデータ量を1バイトとして通信データを扱ったため，1バイトとした．

(ii)16文字の文字列情報:16バイト

LCDの1行目に表示する文字列を扱うために用いるデータ．プレイヤーの操作が正しかった場合には棋譜情報，誤りがあった場合には駒の置き直しを指示する旨の内容を格納した．

(iii)16文字の文字列情報:16バイト

LCDの2行目に表示する文字列を扱うために用いるデータ．プレイヤーの操作が正しかった場合には各プレイヤーの残り時間，誤りがあった場合には駒の置き直しを指示する旨の内容を格納した．

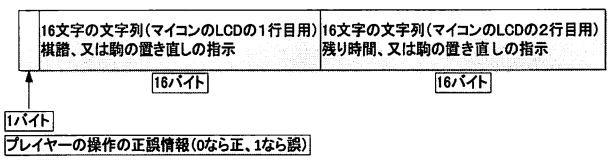


図3.6　パソコンからマイコンへの通信データの内容

3.2.2. マイコンからパソコンへの通信データ

マイコンからパソコンへの6バイトの通信データは光センサ検出結果とボタンの検出結果から基礎的なデータとして以下のようにまとめて扱った．

・白番か黒番の区別・状態遷移数:1バイト

(i)白番か黒番の区別:4ビット

白番か黒番かを示すデータ．白番と黒番の区別は白側のターン終了ボタンが押されたか黒側のターン終了ボタンが押されたかによって判別を行った．これは本来1ビットで十分であるが，後述の状態遷移のデータ数を格納するために用いる4ビットと併せて1バイト単位でデータを管理しているため，残りのデータ量である4ビットを利用する．なお，白側のボタンが押された場合にはデータを $(0)_{10}$ とし，黒側のボタンが押された場合には $(1)_{10}$ として扱った．

(ii)状態遷移数:4ビット

マス内で発生した状態遷移の回数（各マスに設けられた光センサが変化を検出した回数）を示すデータ．0回から4回までが，ルールに準拠している範囲内でのデータであるとして扱った．

・ファイル・ランク・マスの状態遷移:4バイト

ファイル・ランク・駒の状態遷移の1バイト分のデータを1セットとし，最大4バイト分のデータ数を扱った．

(i)ファイル:3ビット

状態遷移が発生したファイルのデータ．aからhまでの縦列の位置を示し，$(0)_{10}$ から $(7)_{10}$ までのデータとして扱った．

(ii)ランク:3ビット

状態遷移が発生したランクのデータ．1から8までの横列の位置を示し，$(0)_{10}$ から $(7)_{10}$ までのデータとして扱った．

(iii)マスの状態遷移:2ビット

マスの状態遷移（マス内の駒の有無の変化）を示すデータ．1つ目に動いた駒のデータを $(0)_{10}$，$(1)_{10}$ の有から無へ，無から有へのデータとして扱い，2つ目に動いた駒のデータを $(2)_{10}$，$(3)_{10}$ の有から無へ，無から有へのデータとして扱った．

・黒側・白側のボタンの状態:1バイト

(i)黒側のボタンの状態:4ビット

黒側の4種類のプロモーションボタンあるいはリザインボタンが押されたかを示すデータ．プロモーションボタンに関するデータはナイト，ビショップ，ルーク，クイーンの順に $(8)_{10}$，$(4)_{10}$，$(2)_{10}$，$(1)_{10}$ とし，リザインボタンに関するデータは $(15)_{10}$ として扱った．

(ii)白側のボタンの状態:4ビット

白側の4種類のプロモーションボタンあるいはリザインボタンが押されたかを示すデータ．黒側と同様にプロモーションボタンに関するデータはナイト，ビショップ，ルーク，クイーンの順に $(8)_{10}$，$(4)_{10}$，$(2)_{10}$，$(1)_{10}$ とし，リザインボタンに関するデータは $(15)_{10}$ として扱った．

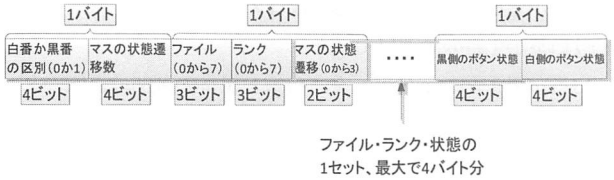


図3.7　マイコンからパソコンへの通信データの内容

ここで例として，図3.8のようにb5のポーンがa6のポーンを取った場合の通信データは各ビット10進数で表現すると図3.9のようになり，1バイトごとに2進数と16進数で表現すると図3.10のようになる．

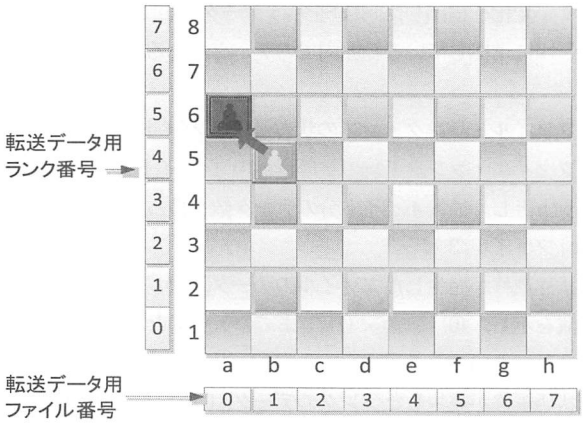


図3.8　b5のポーンがa6のポーンを取った場合の駒の動き

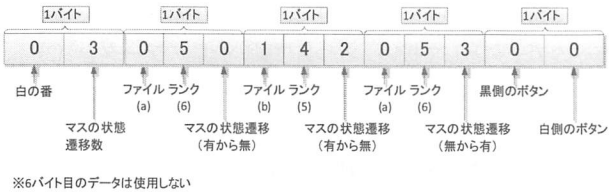


図3.9　各ビットの10進数で表現した通信データ

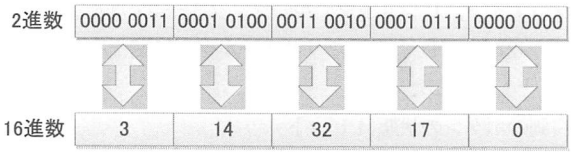
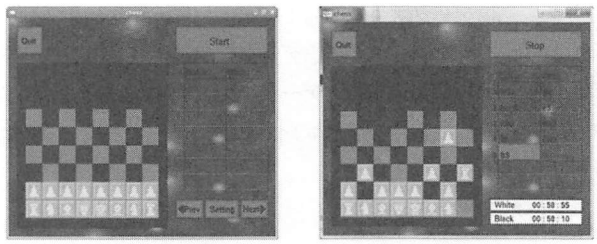


図3.10　1バイトごとに2進数と16進数で表現した通信データ

3.3. GUI

本研究ではチェスの支援として，移植性が良く，プログラミングが比較的簡単であるという特徴を持つOpenGL[3][4]を用いて作成したGUIにて，図3.11 (b)のように棋譜情報，駒位置情報，タイマー情報の表示を行った．設定画面では，タイマーの設定を行えるようにした（図3.12 を参照）．このシステムでは棋譜を付ける際に最も多く用いられている代数式表記法で記述を行う．



(a)　試合開始前　　(b)　試合開始後

図3.11　GUIのメイン画面

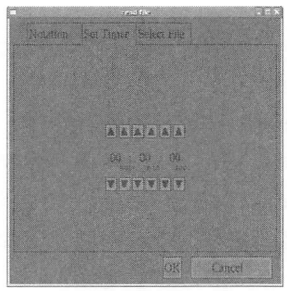


図3.12 設定画面

4. 光センサによる駒情報の取得

マスの状態遷移により，駒の情報を取得することが可能である．このシステムにおいてのマスの状態遷移数による駒の動作及びプレイヤーの判断は表4.1のようになる．なお，本研究ではシステムの基本的な動作に関わる部分を優先して構築したため，(i)スティルメイト，(ii)ドロー・オファー，(iii)デッド・ポジションといったドローの動作やプロモーションといった特殊動作への対応はまだ実現していない．また，プレイヤーが駒を持ち上げたマスと駒を置いたマスの遷移のみをデータとして扱ったため，図4.2のようにa1のルークがh1のルークを取得した場合では，a1とh1のマスのみが状態が遷移したとして扱うこととなる．現在は駒の移動の際に本来はマスの状態遷移に関わらない座標に関するデータを取得した場合はエラーとして扱っているが，通信データの1つ目の駒が1度目に持ち上げられた時のデータ，1つ目の駒が最後に置かれた時のデータ（2つ目の駒が持ち上げられる前のデータ），2つ目駒が1度目に持ち上げられた時のデータ，最後に行われた状態遷移のデータの4つデータから駒の動作を判断することにより，データをまとめるように試みている．

この章の4.1以降では取得する駒の動作情報について記述する．

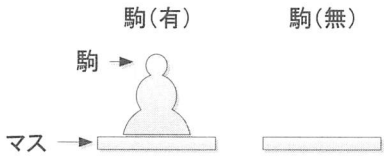


図4.1　マスの状態遷移の様子

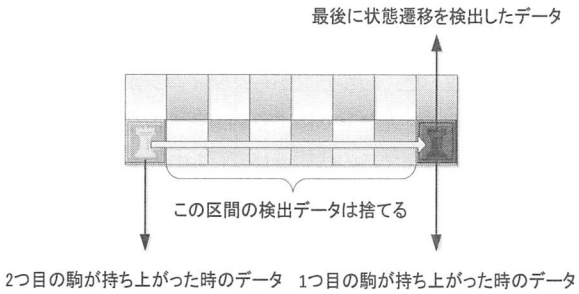


図4.2　a1のルークがh1のルークを取得した時の動作

表4.1 マスの状態遷移数による駒の動作及びプレイヤーの判断

状態遷移数(回)	動作の分類
0	(ⅰ)スティルメイト (ⅱ)ドロー・オファー (ⅲ)デッド・ポジション (ⅳ)リザイン
1	(ⅰ)ルール違反
2	(ⅰ)駒の移動 (ⅱ)駒の移動後のプロモーション
3	(ⅰ)相手の駒の取得 (ⅱ)相手の駒の取得後のプロモーション (ⅲ)アンパッサン
4	(ⅰ)キャスリング
5以上	(ⅰ)ルール違反

4.1. 駒の移動とプロモーション

マスの状態遷移数が 2 回の時の動作の一つとして(ⅰ)駒の移動が考えられる.(ⅱ)駒移動後のプロモーションであるかの判断は,駒位置とボタンのデータから行った.

① 駒が有ったマスから駒が無くなる.
② 駒が無かったマスに駒が有る状態になる.
③ プロモーションの条件を満たせば,プロモーションが行われる.

4.2. 相手の駒の取得とプロモーション

マスの状態遷移数が 3 回の時の動作の一つとして(ⅰ)相手の駒の取得が考えられ,駒の動かし方は相手の駒を先に持ち上げる場合と自分の駒を先に持ち上げる場合の 2 パターンであるとした.(ⅱ)相手の駒の取得後のプロモーションであるかの判断は駒位置のデータから行った.

① 駒が有ったマスから駒が無くなる.
② 駒が有ったマスから駒が無くなる.
③ 駒が無かったマスに駒が有る状態になる.
④ プロモーションの条件を満たせば,プロモーションが行われる.

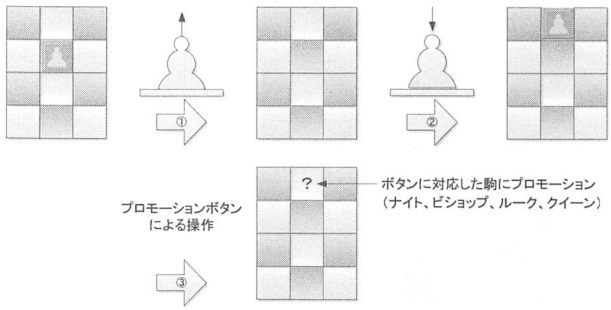


図4.3　駒の移動とプロモーションに関する状態遷移の様子

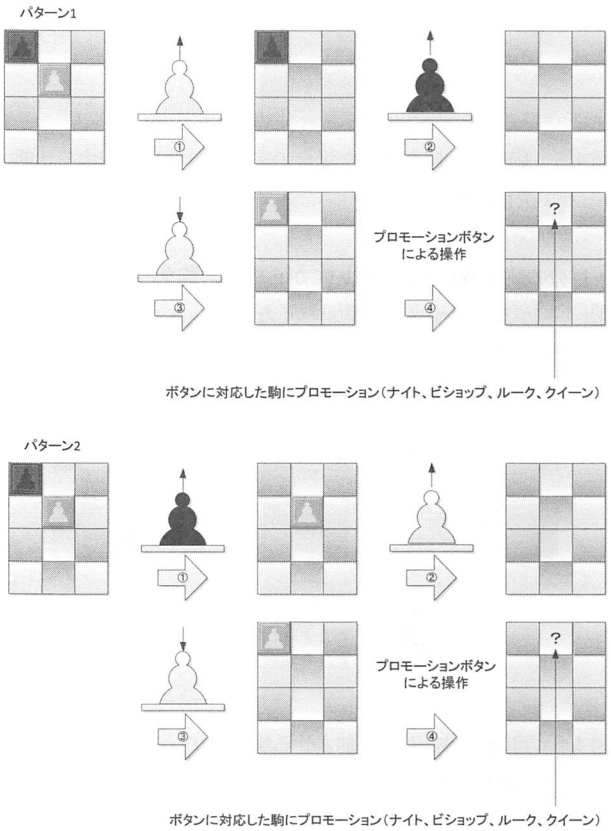


図4.4　駒の取得とプロモーションに関する状態遷移の様子

4.3. アンパッサン

マスの状態遷移数が 3 回の時の特殊動作として(ⅲ)アンパッサンが考えられ,駒の動かし方は先に相手と自分の駒を全部持ち上げてから駒を置いた場合 (2 パターン) と自分の駒を持ち上げて,置いてから相手の駒を持ち上げた場合とが考えられる.

・先に相手と自分の駒を全部持ち上げてから駒を置いた場合

① 駒が有ったマスから駒が無くなる.
② 駒が有ったマスから駒が無くなる.
③ 駒が無かったマスに駒が有る状態になる.

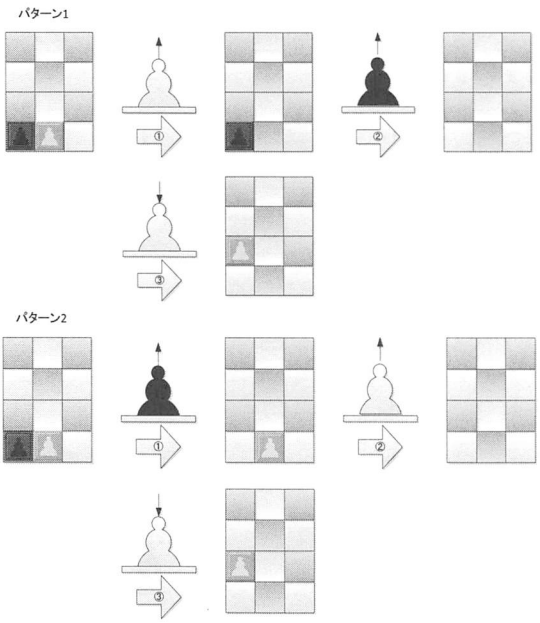


図4.5 駒を全部持ち上げた場合のアンパッサンに関する状態遷移の様子

・自分の駒を持ち上げ，置いてから相手の駒を持ち上げた場合

① 駒が有ったマスから駒が無くなる.
② 駒が無かったマスに駒が有る状態になる.
③ 駒が有ったマスから駒が無くなる.

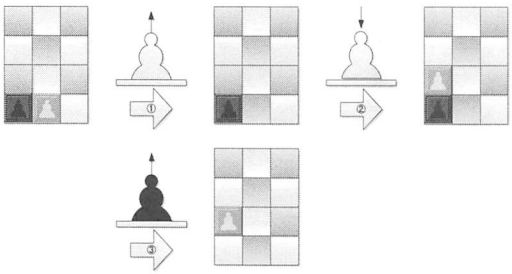


図4.6 自分の駒を先に移動させる場合のアンパッサンに関する状態遷移の様子

4.4. キャスリング

マスの状態遷移数が4回の時の特殊動作として(iii)キャスリングが考えられ，キングサイド・キャスリングとクイーンサイド・キャスリングの2パターンある．公式ルールではキャスリングの際の駒の移動手順は決められており，これに従った場合，移動する駒位置は異なるものの，マスの状態遷移はどちらも変わらず，以下のようになる.

① 駒が有ったマスから駒が無くなる.
② 駒が無かったマスに駒が有る状態になる.
③ 駒が有ったマスから駒が無くなる.
④ 駒が無かったマスに駒が有る状態になる.

5. 結論

本研究ではフォトリフレクタを作製し，H8マイコンと共にチェスボードに組み込むことでパソコンとの通信を行えるようにし，通信データを基とした基本的な駒位置の追跡と対応する棋譜の自動記録を達成した．しかしながら，マイコンからパソコンへの通信データを「4. 光センサによる駒情報の取得」で記述したように必要最低限なマスの状態遷移で行われたものとして扱ったため，ロバストな駒位置検出ができなくなってしまっていることや，ドロー，プロモーション等の一部の動作への対応がまだなこと等，改良すべきことがある．これらの問題を解決して，プレイヤーの負担を軽減させることができるシステムを構築していくことが今後の課題である.

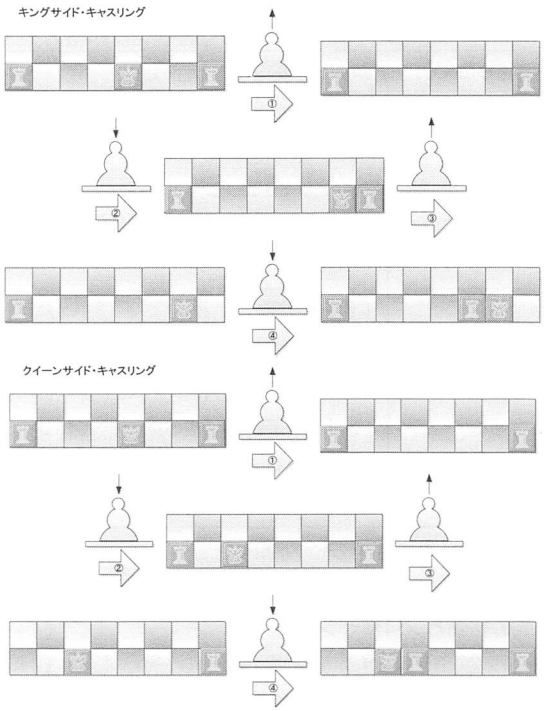


図4.7 キャスリングに関する状態遷移の様子

6. 参考文献

[1] JCA(日本チェス協会),
<http://www.jca-chess.com/>,H25.9.30
[2] CHESS MANIAC,<http://www.chessmaniac.com/>, H25.9.30
[3] 小本 真広, www.komoto.org,
<http://www.komoto.org/>,H25.9.30
[4] 和歌山大学システム工学部デザイン情報学科 床井浩平,
GLUTによる「手抜き」OpenGL入門,
<http://www.wakayama-u.ac.jp/~tokoi/opengl/libglut.html>
,H25.9.30