IJCSN International Journal of Computer Science and Network, Volume 5, Issue 5, October 2016
ISSN (Online) : 2277-5420 www.IJCSN.org
Impact Factor: 1.02

Magnetic Manipulation Based Arduino Assisted 8 Queens Puzzle Solver

[1] Shimon Johnson, [2] Srujan Shetty, [3] Aaron Philip, [4] Nitin Varghese, [5] Amroz Kamal

[1,2,3,4] Department of Computer Science, Mumbai University,
Mumbai, Maharashtra, 400703, India

[5] Associate Professor, Mumbai University, Department of Computer Science,
Fr. Conceicao Rodrigues Institute of Technology,
Mumbai, Maharashtra, 400703, India

Abstract - The N-Queens problem is a well-known NP-Hard problem originally proposed by the chess player Max Bezzel and introduced in 1850 by Carl Gauss. The n-Queen problem is basically a generalized form of 8-Queen problem. In 8-Queen problem, the goal is to place 8 queens such that no queen can kill the other using standard chess queen moves. So, in this paper, the proposed solution will be applied to 8-Queen problem by making use of Artificial Intelligence (AI). The solution can very easily be extended to the generalized form of the problem for large values of `n'. Empirical observations of smaller-size problems show that the number of solutions increases exponentially with increasing n. Search based algorithms have been developed to generate all possible solution sets for a given nXn board. In practice, however, these approaches provide a very limited class of solutions for large size boards.

Keywords - *Reed Switches, Magnetic Manipulation, Arduino Mega, Arduino Uno, Rack Gears, Servo Motor, Drawer Bearings.*

1. Introduction

The N-Queens problem is a classical NP-Hard combinatorial problem in the AI search area. We are particularly interested in the N-Queens problem since it is a relatively simple yet nontrivial case study and test bed in which to explore general issues of designing efficient AI search algorithms and predicting their performance. Due to the exponential growth of the search load in the n-queens problem, even very efficient AI search algorithms can only handle the complexity for about 100-queens .There has been little progress in exploring the N-Queens problem for larger sizes during the last decade. We propose on developing an algorithm that can calculate best possible states of the all positions of the queens on the chess board on basis of the initial arrangement of the 8 queens placed by the user on the chess board and then seamlessly calculating the safe positions of the 8 queens. The 8 queens will then be moved into these safe positions by physically moving them on the chess board using a specially developed robotic mechanism placed under the chess board.

The board mechanism will use magnetic manipulation technique to locate a queen on a chess board and the strength of the relative magnetism to move a queen piece from one square piece to another square, these movements will be according to the legal moves of a Queen in Chess.

2. Mechanical Design

The mechanical system that we propose on developing would be supplied with information about the safe positions and will accordingly move the Queens into positions as per the initial input provided by the user. We aim at developing a mechanism that will freely aid in moving the Queens without making any actual physical contact with them whatsoever, but by making use of magnets, Queens will be positioned on their appropriate safe positions.

The System in Totality would consist of 2 stepper motors moving in both X and Y directions on Rack Gears aligned with respect to the Drawer Bearings, a Servo motor to raise and lower the magnet, an Arduino Uno primarily to control all motor movements, an Arduino Mega multiplexed with all Reed Switches placed under every square grid on the board. The Mechanical system works like any other X-Y table, with a few key differences. First,

the X axis has an extra servo attached to it, which raises and lowers a magnet. The magnet is attracted to pieces on the chess board above, allowing them to move. Second, embedded in the board are 64 magnetically activated reed switches, allowing the Arduino to know the location of each piece.

The Mechanism would slide over Drawer Bearings with the help of stepper motors with enough torque to aid in movement of the magnet on both the X and the Y axis.
On the Y Axis, each rail is connected with the wooden structure, so that the X Axis may sit atop it. Also on the X Axis is the servo that raises and lowers the powerful magnet, so that it may position itself before moving pieces.

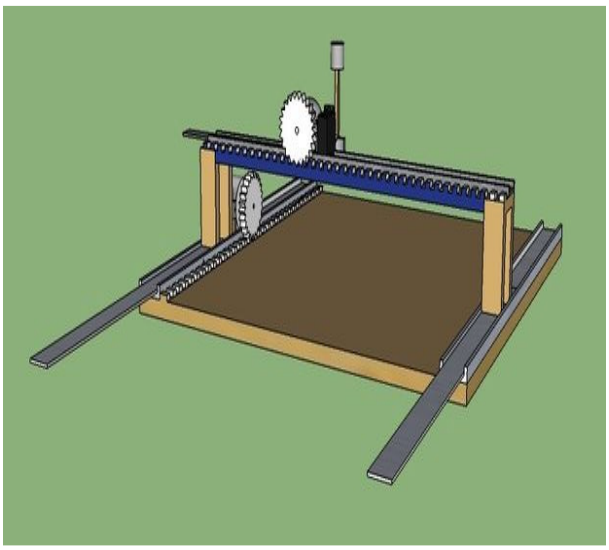

Fig. 1: Design Schematic:-Front View

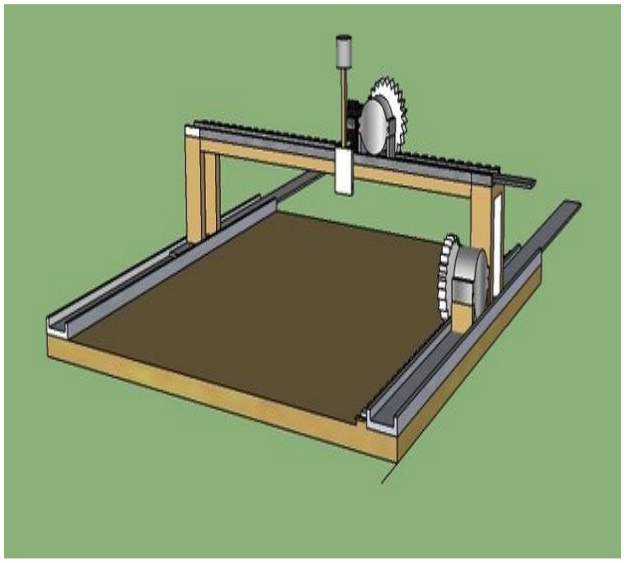

Fig. 2: Design Schematic:-Rear View

3. Electronics

Under each square grid on the chess board Magnetic Reed switches will be soldered such that the positions of the placed Queens are identified by the system and then the algorithm starts processing to return out results. Each square grind on the board has a co-ordinate number assigned to it and according to its square number moves can be defined to the Bot to physically move the queens into their safe positions.

All motor movements will be controlled using an Arduino Uno which will solely be responsible for all movements Made on the Chess Board and an Arduino Mega will be used to keep track of the locations of the queens by hooking up all the 64 reed switches to it via a Multiplexer.

Base 10

7	15	23	31	39	47	55	63
6	14	22	30	38	46	54	62
5	13	21	29	37	45	53	61
4	12	20	28	36	44	52	60
3	11	19	27	35	43	51	59
2	10	18	26	34	42	50	58
1	9	17	25	33	41	49	57
0	8	16	24	32	40	48	56

Fig. 3: Co-ordinates (Base 10)

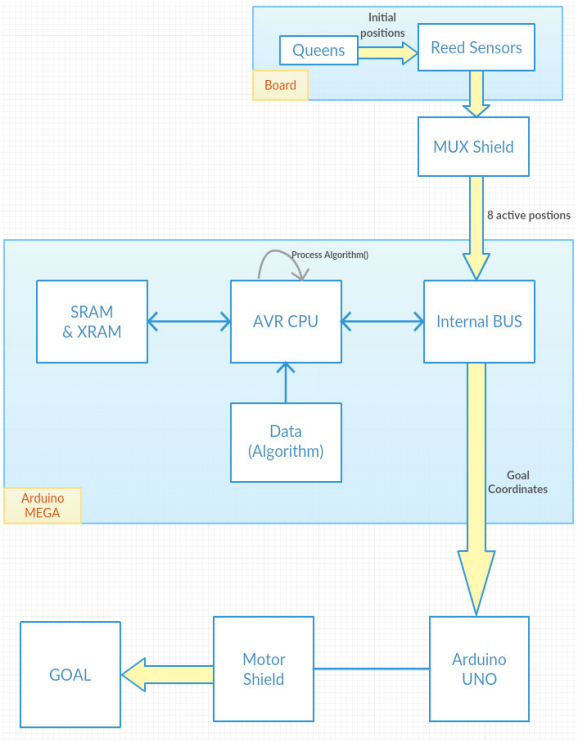

Fig. 4: Block Diagram

This diagram primarily shows an over view of the entire system. The input is provided to the system which is sensed using Magnetic Reed Switches, which are connected to the Arduino Mega via the Mux shield with 8 active positions. The AVR CPU present in the Arduino Mega Runs the developed algorithm to finally produce the Goal coordinates which are then sent to the Arduino Uno which controls the motors via the motor shield to move the queens into The Safe positions.

4. Programming

Our program will provide the system with an ability to think in order to find the goal state such that no queens attacks each other on the chess board by the legal chess moves.

5. Formulation

State description: Any arrangement of 1 to 8 queens on the board.

Initial state: User defined arrangement of all 8 queens on the board.

Successor function: Place an attacked queen on an unattacked square.

Goal state: 8 queens on the board (none attacked).

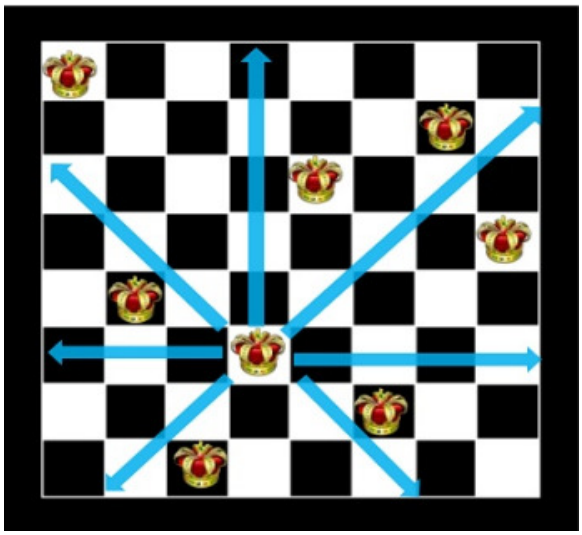

Fig. 5: Acceptable Solution

6. Algorithm

```
Place_Queen (row)
  FOR column= 0 to total_squares
      IF (column==available and LeftDiagonal
      [row+column] == available and RightDiagonal [row-
      column] == available) then
         positionInRow [row] = column;
         Columns [column] =NOT available;
         RightDiagonal [row-column] =NOT available;
         IF row < squares-1 then
                  CALL Place_Queen (row+1);
         END IF
         ELSE then
                  PRINT ("Solution Found");
         END ELSE
         Columns [column] = available;
         LeftDiagonal [row+column] = available;
         RightDiagonal[row-column] = available;
      END IF
  END FOR
END Place_Queen
```

The basic purpose of microcontroller is to receive inputs from reed switches on the chess board, send and receive information to/from the computer terminal and give instructions to motor driver for the motor movement control.

7. Conclusion

It is expected that in the near future every industrial sector will be fully automated which requires the system to have

the ability to think itself. We expect this might be a big step towards achieving that goal as our system is able to place the queens in non-conflicting positions.

8. Future Scope

Once we are able to make our system work flawlessly, we will be looking forward to develop a system with much higher ability that will work towards human development. As human needs keeps on rising on day to day basis, we have a wide scope to build a system that will solve those problems.

References

[1] Wheland, Norman D. (October 1978). "A Computer Chess Tutorial". BYTE. p. 168. Retrieved 17 October 2013.

[2] "Linear Congruence Equations for The Solutions of The N-Queens Problem" (By Erbas, M.M. Tanik, Z. Aliyazicioglu)

[3] "A Circulant Matrix Based Approach to Storage Schemes for Parallel Memory Systems" (By C. Erbas, M.M. Tanik, V.S.S. Nair)

[4] "A Graph Model for Deadlock Prevention, Ph.D. Thesis, Texas A & M University, 1978" (By M.M. Tanik)

[5] "A Survey of Known Results and Research Areas for N-Queens" (By Bell, Jordan; Stevens, Brett)