Unipolar stepper motor driver using sequential logic

This tutorial will show how to make a driver for the unipolar stepper motor 28BYJ-48 (5V) with the ULN2003APG driver circuit. The aim of this driver is to control the motor with 2 pins of an Arduino. These 2 pins will be the direction of rotation and clock frequency which will be handled with a digital pin and one of the 3 timers of the board.

The movement that will be used is in full steps because the aim is to implement it as wheels of a differential robot.

The materials used for this project are:

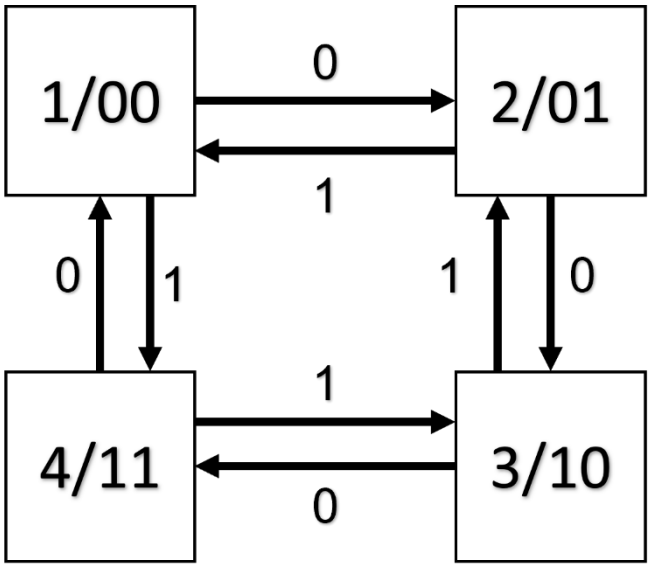
- 1 SN74LS76N Chip
- 2 HD74LS08P Chip
- 1 HD74LS02P Chip
- 1 Motor 28BYJ-48
- 1 Motor controller ULN-2003
- Arduino (To create the steps and direction)
- Protoboard
- Cable

The motor moves by changing the values on the 4 pins of the ULN2003 as shown in the next table:

Full-Step low torque

STEP	IN 1	IN 2	IN 3	IN 4
1	HIGH	LOW	LOW	LOW
2	LOW	HIGH	LOW	LOW
3	LOW	LOW	HIGH	LOW
5	LOW	LOW	LOW	HIGH

The sequential system must follow the next state diagram:



With the next state table:

	Input	
Current State	**X = 0**	**X = 1**
1	4	2
2	1	3
3	2	4
4	3	1

And the truth table with state transition is the next one

Input	Previous state		Next state		1 Flip-Flop		2 Flip-Flop	
X	a	b	A	B	J_a	K_a	J_b	k_b
0	0	0	0	1	0	x	1	x
0	0	1	1	0	1	x	x	1
0	1	0	1	1	x	0	1	x
0	1	1	0	0	x	1	x	1
1	0	0	1	1	1	x	1	x
1	0	1	0	0	0	x	x	1
1	1	0	0	1	x	1	1	x
1	1	1	1	0	x	0	x	1

The truth table was obtained using the state transition table of the JK flip-flop.

State Transition			
Previous state	Next state	Inputs needed	
q	**Q**	**J**	**K**
0	0	0	x
0	1	1	x
1	0	x	1

1	1	x	0

With the truth table the next logic functions are obtained

$$J_a = \overline{x}b + x\overline{b}$$

$$K_a = \overline{x}b + x\overline{b}$$

$$J_b = 1$$

$$K_b = 1$$

The flip flops just change between 4 states. The next truth table shows the change of pins with each state.

Input		Low torque			
A	B	In 1	In 2	In 3	In 4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	1	0	0	1

And the next combinatorial functions were obtained:

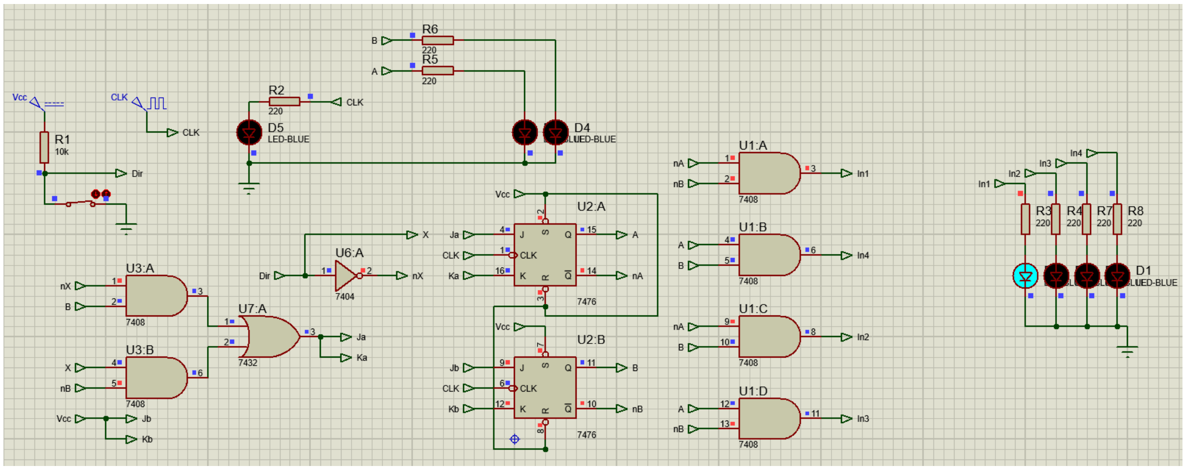
$$In\ 1 = \overline{A}\,\overline{B}$$

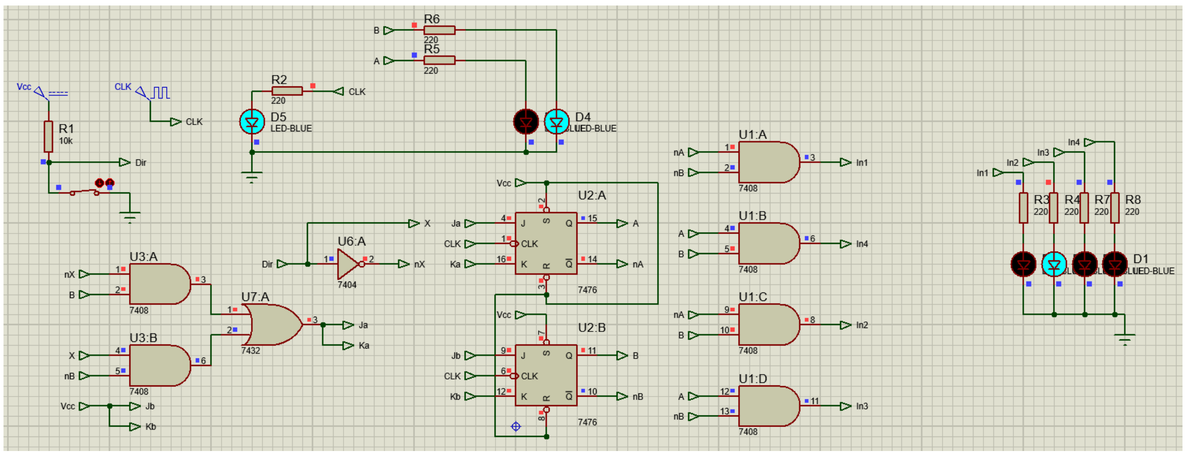
$$In\ 2 = \overline{A}B$$

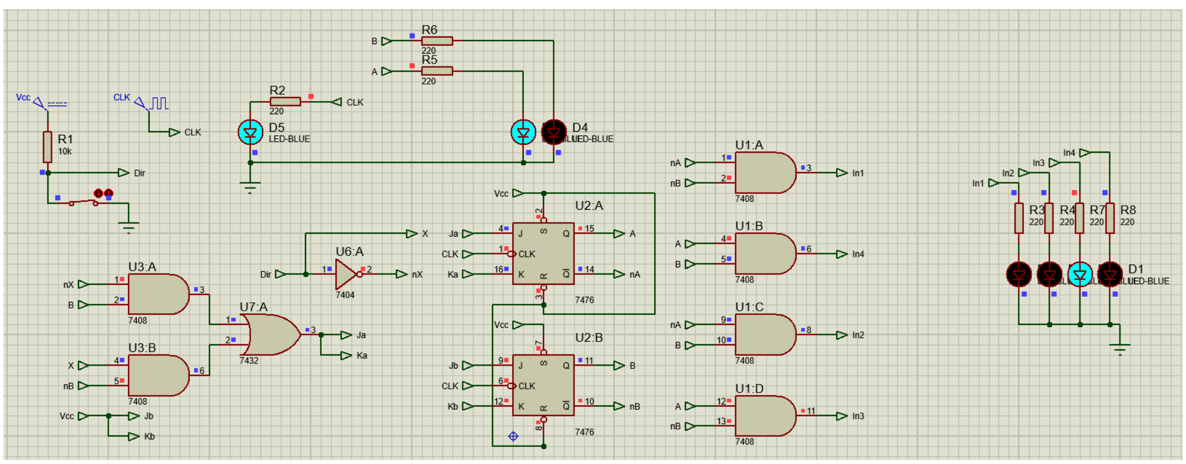
$$In\ 3 = A\overline{B}$$

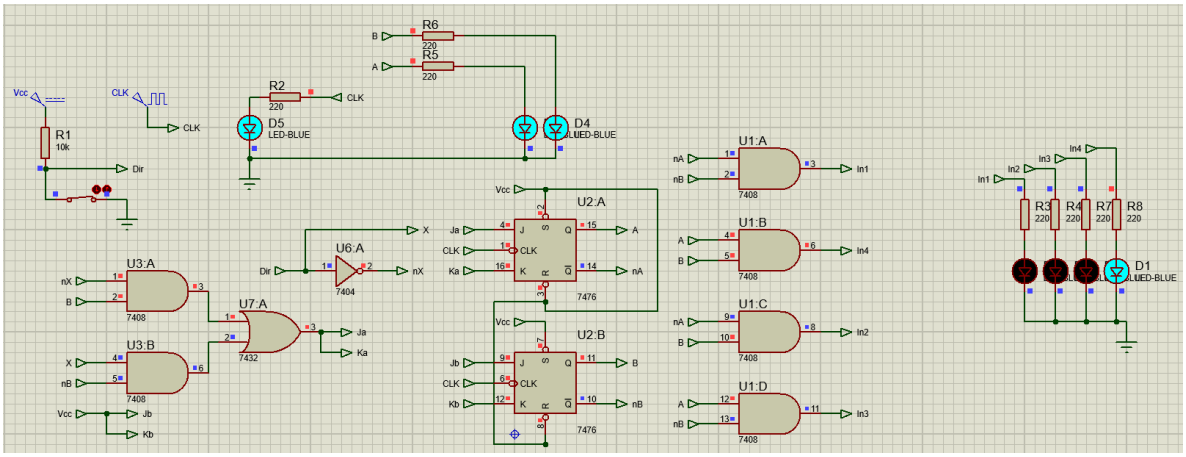
$$In\ 4 = AB$$

A simulation of the system was performed using proteus to check if the system is correct.

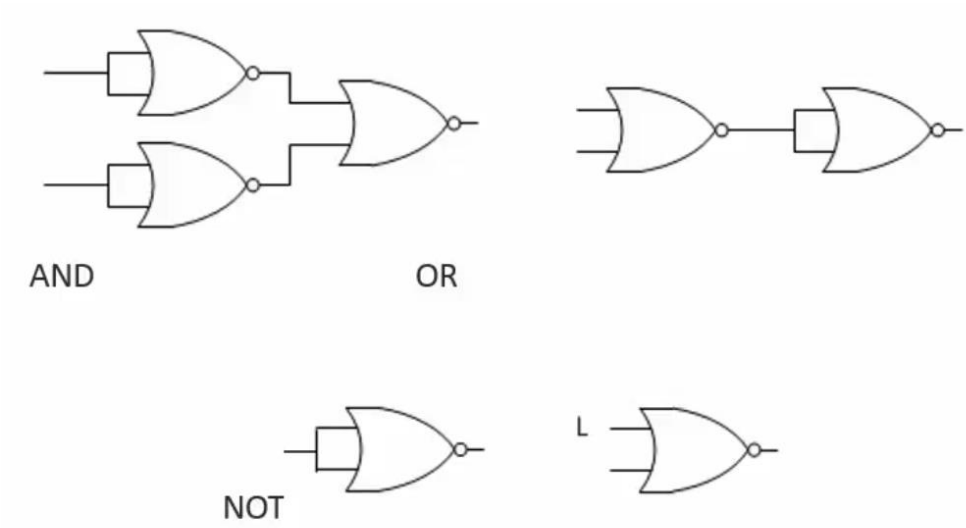

R6
220
R5
220
R2
220
D5
LED-BLUE
D4
R1
10k
Vcc
CLK
Dir
U1:A
U1:B
U1:C
U1:D
U2:A
U2:B
U3:A
U3:B
U6:A
U7:A
7408
7476
7404
7432
R3
R4
R7
R8
D1
In1
In2
In3
In4

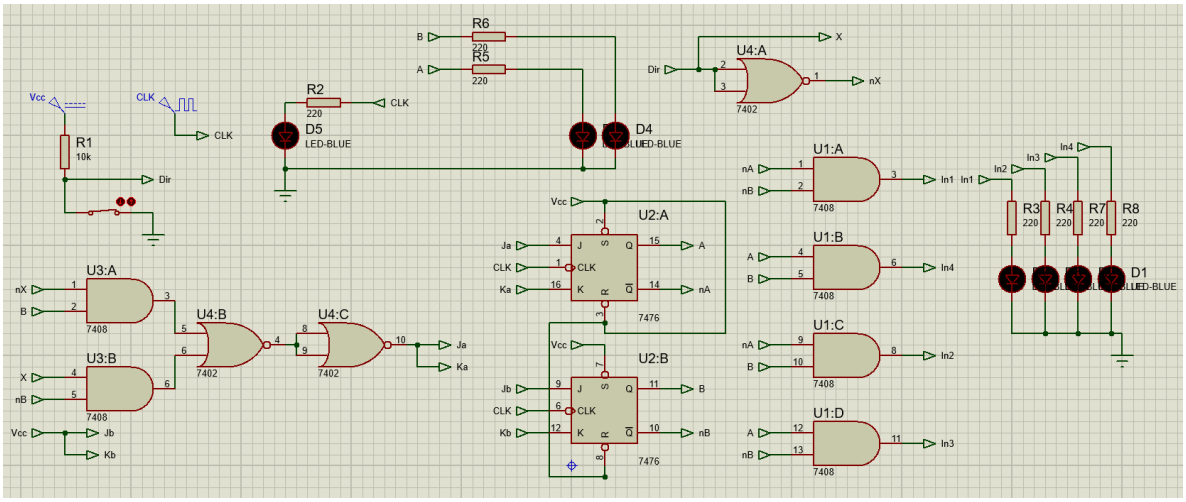

R6
220
R5
220
R2
220
D5
LED-BLUE
D4
R1
10k
Vcc
CLK
Dir
U1:A
U1:B
U1:C
U1:D
U2:A
U2:B
U3:A
U3:B
U6:A
U7:A
7408
7476
7404
7432
R3
R4
R7
R8
D1
In1
In2
In3
In4


R6
220
R5
220
R2
220
D5
LED-BLUE
D4
R1
10k
Vcc
CLK
Dir
U1:A
U1:B
U1:C
U1:D
U2:A
U2:B
U3:A
U3:B
U6:A
U7:A
7408
7476
7404
7432
R3
R4
R7
R8
D1
In1
In2
In3
In4

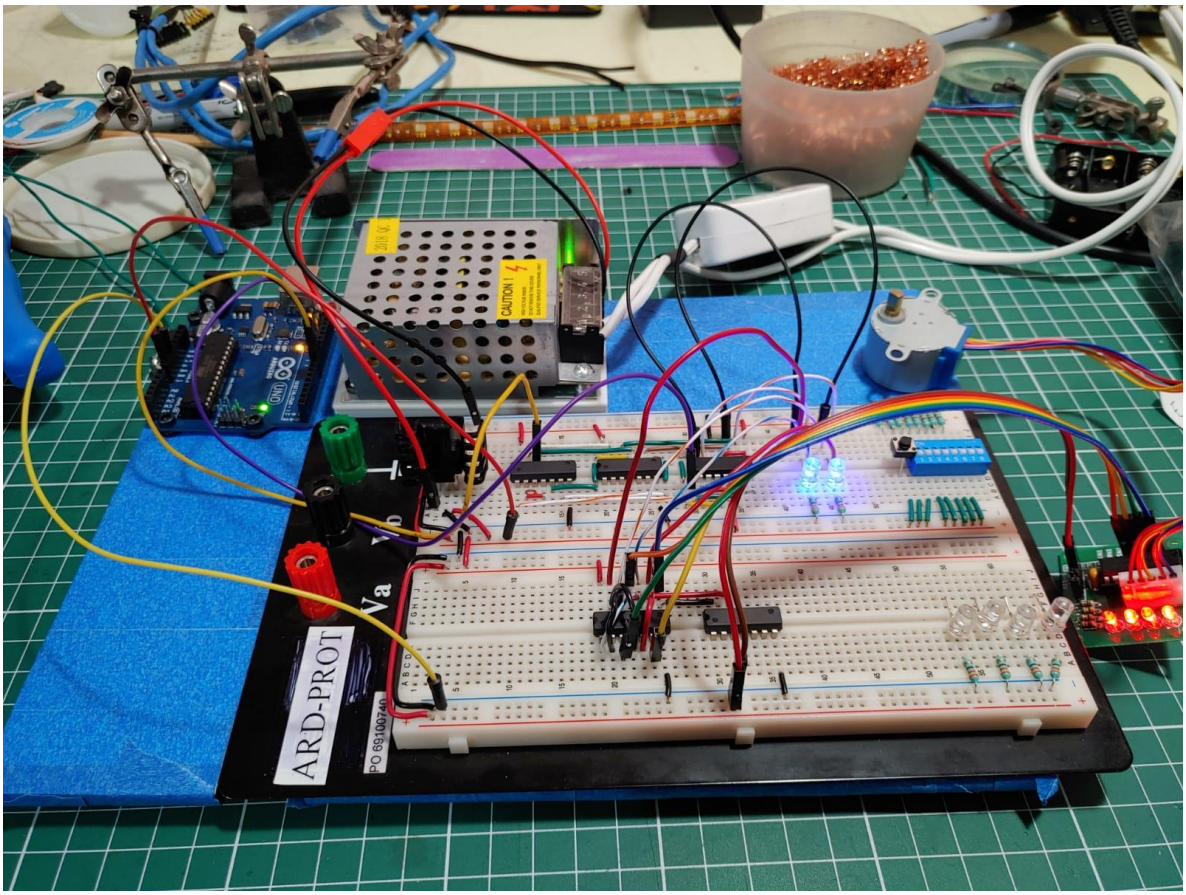


To maximize the usage of the gates and minimize the quantity of the chips, the NOT and OR chips were changed to a NOR with several configurations to make the required gates.





This is the physical implementation of the circuit.


ARD-PROT
CAUTION