**Project Based Learning Task 2**

**Design of a Robot Line Follower using Sequential circuits**

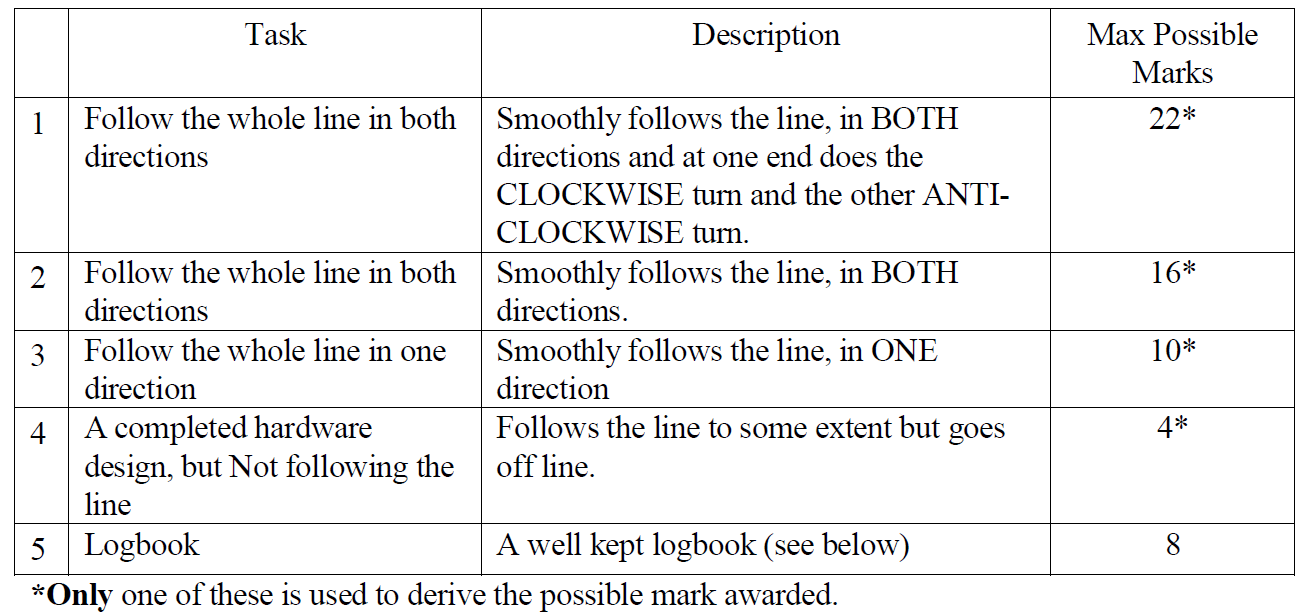
*26th of January, 2017*

**Objective:**

Design and implement a robot line follower using state machines and sequential circuit techniques to ***smoothly*** follow the line in ***both*** directions.

* At one end does the CLOCKWISE turn and the other ANTI-CLOCKWISE turn. At each end of the line the robot should turn around and follow the line again.
* The circuit designed is to be put on a wire-wrap board with room for IC sockets and attached to the robot.

**Assessment:**

This task is worth 30% of the coursework mark.

**Deadline:** 23rd February 2017

**Planning**

1. **Inputs:**We have got three inputs coming from our sensors and those would be the eight possibilities that we should consider.

|  |  |  |  |
| --- | --- | --- | --- |
| left | middle | right | **position** |
| 0 | 0 | 0 | not on the line |
| 0 | 0 | 1 | too much to the right |
| 0 | 1 | 1 | a little bit to the right |
| 0 | 1 | 0 | in the middle |
| 1 | 1 | 0 | a little bit to the left |
| 1 | 0 | 0 | too much to the left |
| 1 | 0 | 1 | left and right sensor on |
| 1 | 1 | 1 | all sensors on |

**\*The last two** options are never going to occur, so we are going to put them as “don’t care” in our state diagram.

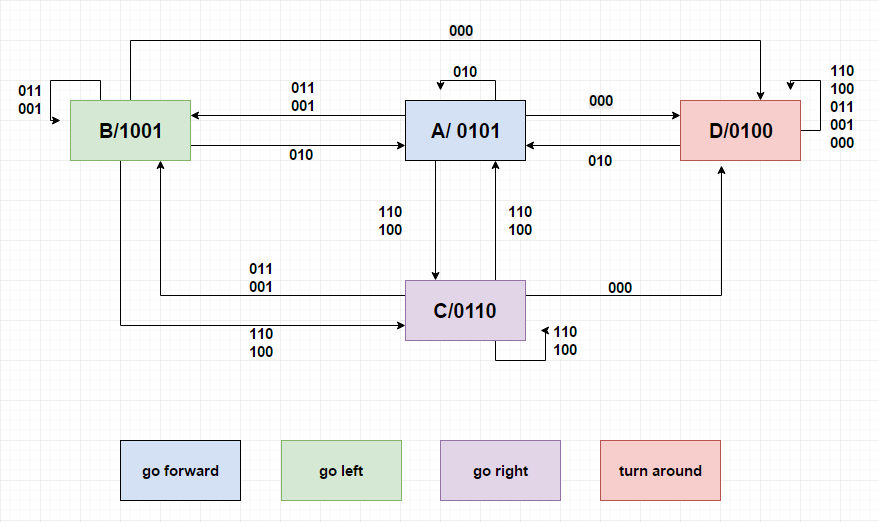
1. **Outputs:**

We have got two motors on our robot and we enable them by putting 0 to the enable bit. The direction is determined by the direction bit – ’1’ to go forward and ’0’ to go backwards.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Enable  Left | Direction Left | Enable Right | Direction Right | **Movement** |
| 0 | 1 | 0 | 1 | Forward |
| 1 | 0 | 0 | 1 | Hard right |
| 0 | 1 | 1 | 0 | Hard left |
| 0 | 0 | 0 | 1 | Turns around (clockwise) |
| 0 | 1 | 0 | 0 | Turns around  (anti-clockwise) |

**\***In this first attempt we are going to try to make the robot follow the line and only turn clockwise at the end of it.

1. **State diagram**



1. **State table**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Previous state** | **Next state** | | | | | | | | **Output** |
| 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| A | D | B | B | A | C | x | x | C | 0101 |
| B | D | B | B | A | C | x | x | C | 1001 |
| C | D | B | B | A | C | x | x | C | 0110 |
| D | D | B | B | A | C | x | x | C | 0100 |

**6. Transition table**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Previous state** | | **Next state** | | | | | | | | **Output** |
| 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| A | 00 | 10 | 01 | 01 | 00 | 11 | x | x | 11 | 0101 |
| B | 01 | 10 | 01 | 01 | 00 | 11 | x | x | 11 | 1001 |
| C | 11 | 10 | 01 | 01 | 00 | 11 | x | x | 11 | 0110 |
| D | 10 | 10 | 01 | 01 | 00 | 11 | x | x | 11 | 0100 |

**70. Excitation map  
 -** first flip flop:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Previous state** | **Next state** | | | | | | | | **Output** |
| 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 1- | 0- | 0- | 0- | 1- | x | x | 1- | 0101 |
| 01 | 1- | 0- | 0- | 0- | 1- | x | x | 1- | 1001 |
| 11 | -0 | -1 | -1 | -1 | -0 | x | x | -0 | 0110 |
| 10 | -0 | -1 | -1 | -1 | -0 | x | x | -0 | 0100 |

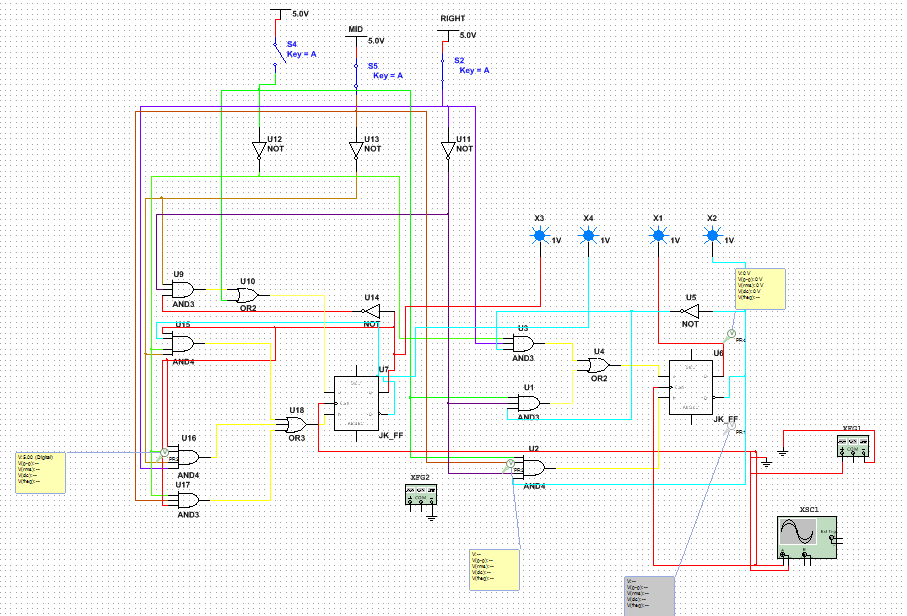
* second flip flop:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Previous state** | **Next state** | | | | | | | | **Output** |
| 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 0- | 1- | 1- | 0- | 1- | x | x | 1- | 0101 |
| 01 | -1 | -0 | -0 | -1 | -0 | x | x | -0 | 1001 |
| 11 | -1 | -0 | -0 | -1 | -0 | x | x | -0 | 0110 |
| 10 | 0- | 1- | 1- | 0- | 1- | x | x | 1- | 0100 |

**Simulation**

*2nd of February, 2017*

* Now that we have created our circuit we are going to recreate it in ***Multisim*** and see if it works.
* We do this in order to test it without having to wire it up first, since it would be a lot easier to find any mistakes made in the simulation rather than the real circuit.

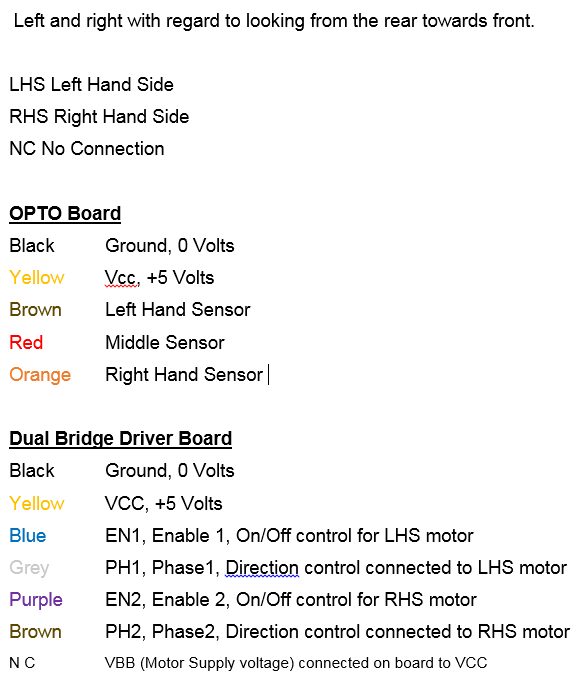
**Expectations:**We should be able to see easily if the output is as expected for the given inputs that we have. If our simulation is successful, we will be able to proceed with wiring up our circuit on the actual board.

**Results:**Everything works as expected for all inputs.  
  
**Modifications needed:**Circuit is way too big for the board that we have. We need to go back to our maps and reduce the expression as much as possible since we only have 6 chip sockets.

**Wiring up the circuit**

*9th of February, 2017*

**Objective:**

* Now that we have a working simulation, we can proceed with wiring up the circuit on our board.
* We’ve reduced the number of gates significantly, in order to be able to fit it in the sockets we have available.
* This is what we have to consider:  
    
  

**Testing**

*10th of February, 2017*

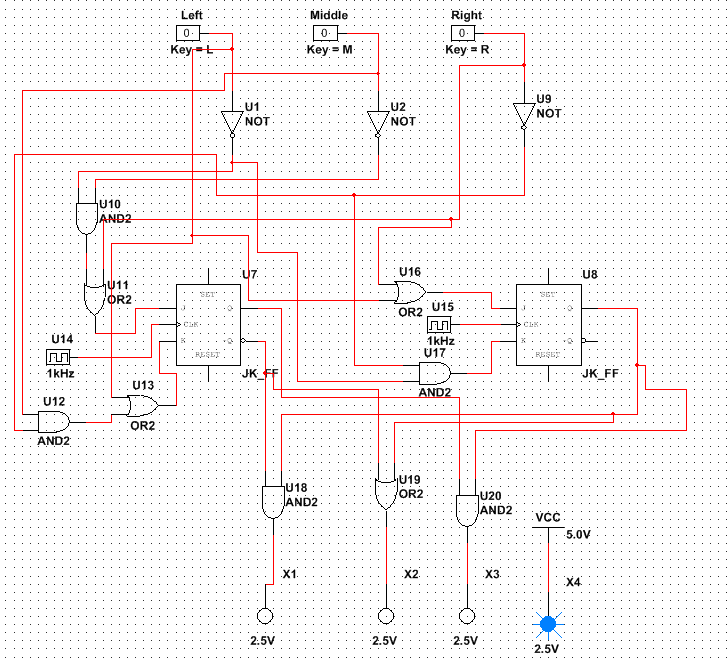
**Expected results:**

* We are finally going to test our robot that we have wired up on your circuit board.
* Our robot should follow the line and turn clockwise at the end of the turn.
* Adjustments might be needed in terms of clock speed.

**Checklist:**

* Does the robot follow the line?
* Does it move and turn smoothly?
* Does it go outside of the line?
* Does it turn around at the end of the line?

**\***The circuit that we used is the same as before:



**Actual results:**

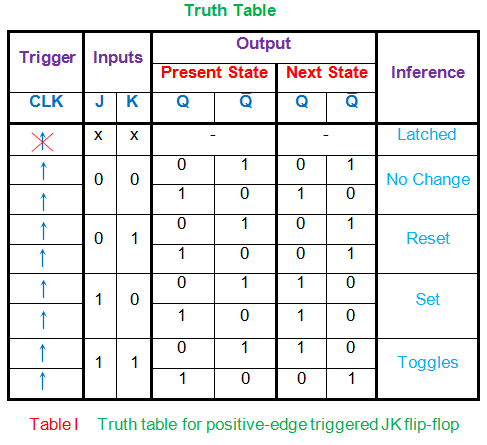
* **Does the robot follow the line?**   
   - Yes, the robot follows the line.
* **Does it move and turn smoothly?**   
   -It runs fairly smoothly.
* **Does it go outside of the line?**  
   - It rarely or almost never goes outside of the line, only on very sharp turns.
* **Does it turn around at the end of the line?**   
   - It does turn around at the end of the line, but only clockwise, as we have planned it like this for now.

**Modifications needed:  
*The speed of the clock*** needs to be adjusted just slightly in order for our robot to run smoothly. We still have to implement a way for our robot to turn clockwise *and* anti-clockwise at the end of the line.

**Clockwise and anti-clockwise turn**

*16th of February, 2017*

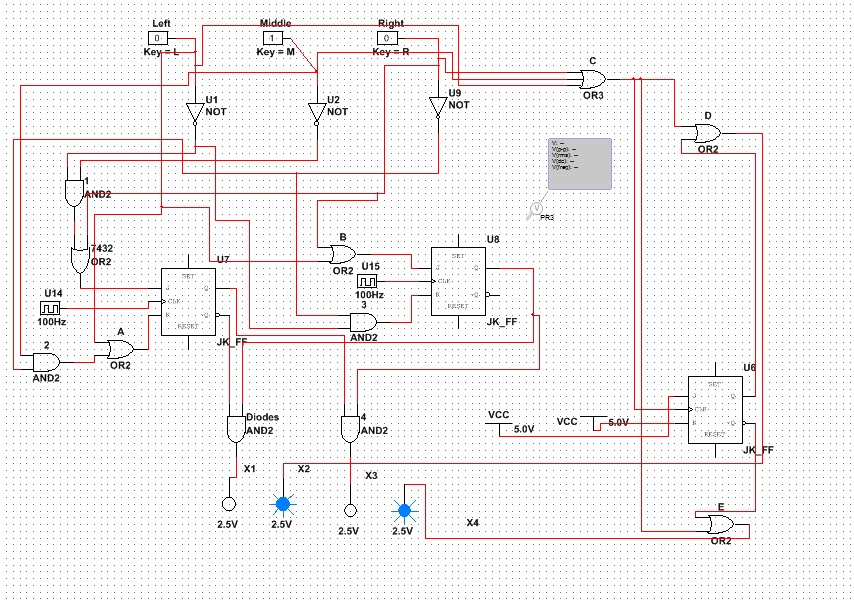
* We figured out the logic for the second turn by realising that we only need to trigger the turn when the 000 sensor position is hit.
* We thought it would be a good idea to use this to trigger a toggle change of a third flip flop to invert the robots turn every time it hits an end of line position.
* We set both of the flip flips J and K position to a constant five volts which enables the flip flop to toggle every time the 000 position hits, there for inverting the output.



Source: <http://www.electrical4u.com/jk-flip-flop/>

**Clockwise and anti-clockwise turn  
Simulating and Testing**

*17th of February, 2017*



**Expected results:**

* The simulation worked correctly, so the circuit was built on the line following robot itself.
* Our robot should now follow the line and turn clockwise at the end of the line and then anti-clockwise at the other end.
* Adjustments might be needed in terms of clock speed.

**Checklist:**

* Does the robot follow the line?
* Does it move and turn smoothly?
* Does it go outside of the line?
* Does it turn around at the end of the line?

**Actual results:**

* After building, the first test incurred an error at the end of line position which prevented the robot from turning, and instead pivoting from left to right at about a 20 degree angle each side. This instantly show us that our third flip flop was wired to the clock instead of the output of the or gate of the three sensors, so that it causes a change at only the 000 state.
* We rewired this, and the robot starting functioning correctly, giving us two way directional turns.
* **Does the robot follow the line?**   
   - Yes, the robot follows the line.
* **Does it move and turn smoothly?**   
   -It runs fairly smoothly, but less than it did with the previous circuit.
* **Does it go outside of the line?**  
   - It rarely or almost never goes outside of the line, only on very sharp turns.
* **Does it turn around at the end of the line?**   
   - It does turn around at the end of the line, both clockwise and anti-clockwise.

Table of Contents

**Planning46**

Inputs and outputs46

State diagram and State table47

Transition table and Excitation maps48  
Additional work49

**Single turn53**

Simulating with Multisim53

Reducing number of chips54  
Simulating the new circuit67

Wiring up71

Testing the robot on the line73

**Clockwise and Anti-clockwise turn75**

Planning75

Simulating and Testing the turns76

**Overview77**

Final circuit diagram77

Evaluation78