ELEC254 – Fall 2008
Group 25 Project Report
Rubik’s Cube Solver

By

CHEN Yuncong
TANG Chengzhi
WANG Wei


**Introduction:**

Electrical & Electronic Engineering becomes more and more important in our daily life. The development of electronic technologies has changed our life greatly. Among various electronic technologies, the application of embedded system can be found everywhere now. The micro processors become cheaper and cheaper and their functions are stronger and stronger.

The Intel 8051 is a really classic micro controller. It is reliable and cheap and it can fulfill some simple tasks. We can write assembly codes to control it.

In this project, my partners and I plan to design a Rubik’s Cube Solver using the knowledge of Elec254. It has three main functions. First, it can solve the Rubik’s Cube automatically. Second, we can use the computer keyboard to rotate the Rubik’s Cube arbitrarily. Third, the Rubik’s Cube can rotate randomly by itself.

This system has 4 main parts which is stepper motor controller, LCD, communication component (DB 9 series port), color detection.

The process of solving the Rubik’s Cube is as followed. At the beginning, we wrote a program under Linux environment and used the web camera to detect the colors of the Rubik’s Cube, and generated the digitalized initial state of the Rubik’s Cube. Then, we sent the initial state signal to the micro controller 8051 by the series port. After that, we used 8051 to control six step motors to rotate the Rubik’s Cube. At the same time, the LCD shows the progress and some information about the system.
Overall description of the system

Main hardware
- Intel 8051
- Step motors
- LCD
- Max232 regulator
- DB 9 serial port
- Notebook computer
- Web camera
- Multiplexer
- LS293D Driver
- Breadboard
- Button
- Oscillator(11.059M Hz)

Main software
- Maneuver generation
- LCD display
- Motor controlling
- Serial port communication
- Color recognition
Hardware operation principle

Step one
At the beginning, the web camera is used to take one photo of each side of the Rubik’s Cube. We wrote a program under Linux environment. The color detection principle is that the Web-camera captures the picture by recording every pixel. And every pixel is recorded as RGB format (three basic colors: red, green and blue). The computer uses 8 bits to represent each basic color. In the program, we define a value range for each color of the Rubik’s Cube. When we capture the picture, we always put the Rubik’s Cube at the same position in front of the Web-camera. In the program, we pick 9 fixed points out of each picture, and compare them with the value range we have defined, thus we can know the colors of the Rubik’s Cube. We also designed error detection mechanism. The principle is that the total number of the cube with each color is 9. So in the program, the number of each color is counted, the total numbers should always be 9; if not, the PC will report the error.

Step two
After getting the color information of the Rubik’s Cube, we use the RS232 serial port to send the color information from the PC to the Intel 8051.

Step three
After receiving the color information, the Intel 8051 will generate the Rubik’s Cube solution strategy with the main program we have designed.

Step four
We use the Intel 8051 to control the six step motors to rotate the Rubik’s Cube. When we design the circuit, we chose to use four 1-
Multiplexer in order to reduce the controlling pins on Intel 8051. Every step motor should be controlled by 4 pin, without multiplexer, we have to use 4*6=24 pins; however, with the multiplexers, only seven pins are enough. (Three of them are used as the controlling pins of the multiplexers) There are three buttons in our circuit. One is the continue button. After pressing it once, the corresponding step motor will rotate 90 degrees (the Intel8051 and the multiplexers decide which step motor should work now). Considering that the machine is not precise enough, we designed a small adjustment system. The other two buttons are used to do small adjustments. The smallest adjustment is 1.8 degrees. One button is used to do clockwise adjustment and the other is used to do counter clockwise adjustment.

**LCD display**

At step one, we use the continue button to control the process of the picture capture. If the web-camera captures one picture, the PC will send a “d” to the Intel 8051, and then the “d” will be displayed on the LCD. In this way, we know that this picture capture step is done. Then we press the continue button to capture the next picture.

At step two, the PC will send the color information to the Intel8051 (six colors: w,y,r,g,b,o). The color information received by the 8051 will be displayed on the LCD. With the LCD, we can also know the progress and check whether the data is wrong.

At the beginning of step four, the LCD will display the total rotation steps that are needed. Then the LCD will display the rotation progress. Each rotation step is 90 degrees in clockwise direction. On the LCD, “f” represents that the front side rotates one step; “b” represents that the back side rotates one step; “l” represents that the left side rotates one step; “r” represents that the right side rotates one step; ”u” represents that the upper side rotates one step; “d” represents that the down side rotates one step.
Problems we met

1. In this project, the mechanism requires high precision, in order to make the mechanism more precise, a mechanic professional helped us produce the main structure of the device.
2. The Rubik’s Cube always got stuck, and we have spent a lot of time to adjust the mechanic device.
3. During the time we were doing the project, two LCD were burned; finally, we found the reason was that the current supplied was too big.
4. When we connected the RS232 serial port, we tried many times but still failed. Finally, we found that the oscillator (11.0592MHz) is important to necessary to make the baud rate of the Intel 8051 is the same as that of the PC. Another problem is that we chose to use a UBS to RS232 data wire; we got a data sheet of RS232 from internet; then we tried many times but still failed. The reason is that the configuration of the RS232 drive software is different from the tradition one, and the function of each pin is different. After many experiments, we managed to know the function of each pin.
Hardware Features

Stepped motor
It’s a kind of unipolar stepped motor. Four of the wires attached to are given in a serial order to make it drive.

LS293D
Each LS293D is used to drive a single motor.

Logic Symbol

Logic Diagrams
MAX232

MAX232 is used as USB-serial communication port for transmission use.
74HC4051

Each HC4051 serves as a multiplexer for choosing which motor to drive in each next step in order to rotate the Rubik’s cube.
Functional Diagram:
Schematic Diagram:

Hardware actual picture
Software Algorithm

We employ one of the cube solving algorithm – Fridrich’s System and make some modifications in order to fit the hardware requirement.

<table>
<thead>
<tr>
<th>Action description</th>
<th>Average number of moves</th>
<th>Time(average for our solver)</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place the four edges from the first layer</td>
<td>7</td>
<td>16 sec.</td>
<td></td>
</tr>
<tr>
<td>Place four blocks each consisting of one corner from the first layer and a corresponding edge from the second layer.</td>
<td>4*7=28</td>
<td>4*16=64 sec.</td>
<td></td>
</tr>
<tr>
<td>Simultaneously orient the corners and edges so that the last layer has the required color</td>
<td>9</td>
<td>24 sec.</td>
<td></td>
</tr>
<tr>
<td>Simultaneously permute the 8 cubes in the last layer without rotating corners or flipping edges</td>
<td>12</td>
<td>32 sec.</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>56</td>
<td>136 sec.</td>
<td></td>
</tr>
</tbody>
</table>
One of the unique features of this system is that the last layer is always solved using two algorithms of an average length of 9 and 12, which is very efficient. The average lengths are based on frequencies with which various orientations and permutations occur and on the length of algorithms for each position. Another interesting feature is that for the first two layers no lengthy algorithms are needed and you can use your intuition and utilize the specifics of the particular initial state and subsequent states of the cube.
## Cost List

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Sub-Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor suit</td>
<td>6</td>
<td>25</td>
<td>150</td>
</tr>
<tr>
<td>Cube container</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Rubik’s Cube</td>
<td>1</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>74H4051</td>
<td>3</td>
<td>7.5</td>
<td>30</td>
</tr>
<tr>
<td>ULA2803</td>
<td>4</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>LS293D</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>LCD</td>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>DB 9 serial port</td>
<td>1</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>DB 15 serial port</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MAX232 regulator</td>
<td>1</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Oscillator</td>
<td>1</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Switch</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Connector</td>
<td>10</td>
<td>1.6</td>
<td>16</td>
</tr>
<tr>
<td>Other Stuff</td>
<td></td>
<td></td>
<td>31.5</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>445</strong></td>
</tr>
</tbody>
</table>
Summary and Further Development

It took us quite a period to deal with the problems we faced, with the problems continue to appear along with our progress. We had a lot of fun dealing with the problems and to see every successful solution for each of these problems, with help of the knowledge we learned in class and outside the classes. We’ve stated some major problems we faced, whereas there were countless problems we actually had, and it’s hard to turn all those into paper statements. Practical deeds are much more vital than theoretical practice only.

Our inspiration comes from Rubik’s cube since two of us are cubic fan. We had fun in this inspiration, we had fun dealing with the problems using knowledge, and we had fun presenting our project to people to share the fun. We believe that this is only the beginning of our study on microprocessors, and we had felt more power to keep up after this admirable success for all of us.

We plan to combine the camera and the cube together, and to work on improving the algorithm to make it like a real commercial product. That way this cube solver may bring more fun to others.