



## PREAMBLE

This robot solves the 3x3x3 [Rubik's Cube®](#).

I started to think about this problem in about August of 2000. In Jan 2001 fellow Mindstorms forums user 'agiecco' announced his intention to work on a robotic solution and, simultaneously, I saw that Rubik's Cubes were on sale at [www.target.com](http://www.target.com). So I bought a couple of cubes and started getting down to business...

I produced a 'late beta' version in mid-April 2001 that was a little clunky. The final version (presented here) is smooth and fairly reliable.

## MECHANICS

To achieve a cube solution, you must be able to rotate the whole cube by  $90^\circ$  in two orthogonal axes, *and* be able to turn a face by  $90^\circ$  relative to the rest of the cube.

I opted to use the left and right grabbers to turn the faces; the yellow grabber can rotate the left face by an  $90^\circ$ ; the green grabber can rotate the right face by  $90^\circ$ ; or they can both turn simultaneously while the bottom grab is open to rotate the whole cube through  $90^\circ$ .

The bottom grabber holds the center 'slice' of the cube when the left or right grabber is turning a single face, and also provides a  $90^\circ$  turn for rotating the whole cube.



The tricky part is to re-orient the cube between moves to present the next face to be turned by the left or right grabber.



## PROBLEMS SOLVED

### 1. The cube faces are generally too stiff for LEGO elements to turn

This problem was solved by a tip I found on Lars Petrus's [Speed Cubing page](#): lubricate the cube with silicone spray lubricant. I got an aerosol can of *LubriMatic Heavy Duty Silicone Lubricant* from my local ACE hardware store and squirted some inside the cube. After wiping off the excess spray you must rotate the cube faces for about 30 minutes to prevent the solvent in the spray dissolving the cube body and sticking it solid. Once you have done this, the result is a fairly slick cube.



However, I found that some of the faces still got stuck because the springs inside the cube were too stiff, so I pushed some wedges (actually, plastic knives) in between the cube slices and left them over night to force the springs inside the cube to loosen up. After this treatment the cubes handled very well.

### 2. Getting enough torque

Even with a treated cube, getting enough torque to turn the cube faces was going to be a problem. I remembered the system that Jin Sato used on the thigh joints of [MIBO](#) -- worm gear to the outer 56t ring of the large Technic turntable. This gives torque to spare for turning the faces of a treated cube.

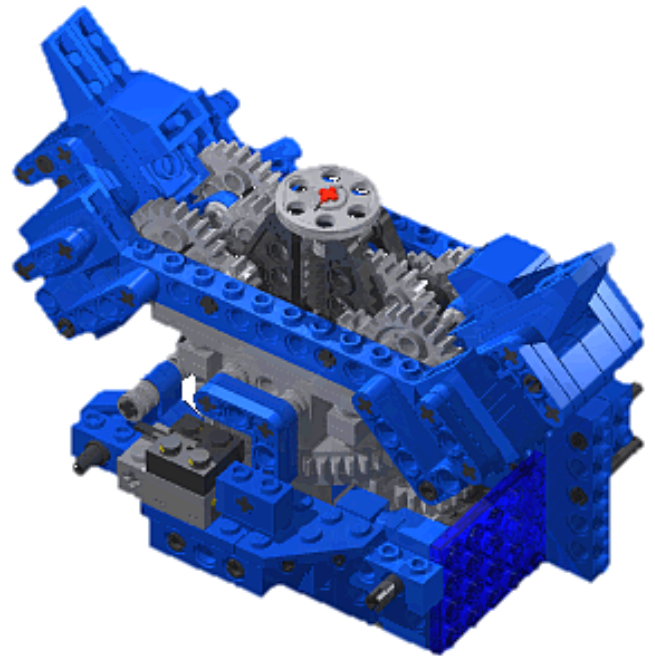
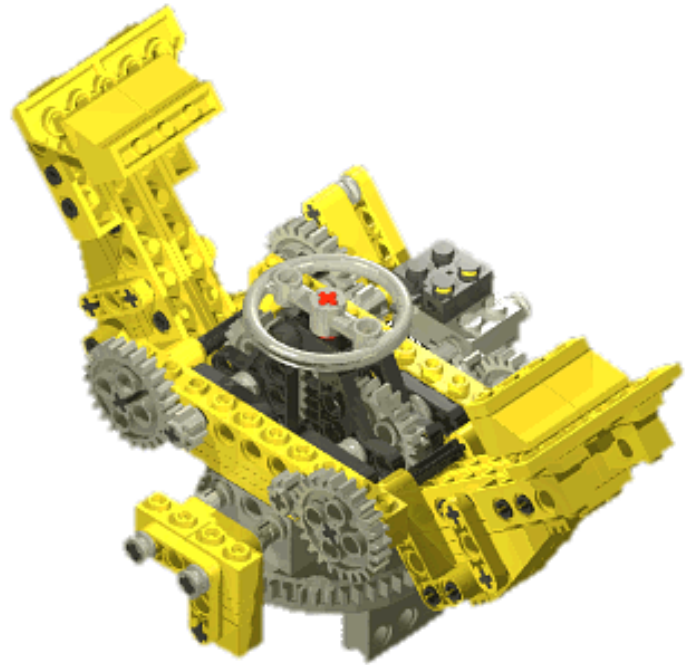
### 3. LEGO grabbers don't grip strongly enough

My early attempts at building a cube solver

were all stymied by grips that slipped. The worm-56t gave enough torque to turn, but the fingers couldn't hang on and the grip was simply pried apart as the grabber rotated around the stationary cube face. I thought about changing the device's name to *ButterFingers*.

I rebuilt the left and right grabbers six times (and the bottom grabber four times) trying elastic bands, Technic shocks, and pneumatics, before I came up with an adequate grip mechanism. In the present version, an axle runs from a motor through the center of the large Technic turntable to a worm screw. The worm screw turns two 24t gears mounted either side of the worm inside the body of the grabber. Each end of the 24t axles terminates with an 8t gear outside the body of the grabber, and these 8ts engage with 24t-s on either end of the axles which carry the grabber's fingers. This system can be strained quite tight without risk of gear slippage, and also allows the large turntables to rotate 180° without any significant loss of grip.

For the bottom grabber I had to use a slightly different arrangement (same gear combinations) because the fingers of the green and yellow grabs kept catching on the external 24ts of the bottom grab. Eventually I managed to work out how to mount all the gears internally in the 4-stud width of large Technic turntable.



#### 4. Precision of movement

Having solved these problems, there was still the problem of 'slop' or 'gear lash' in the left and right drive trains.

Most of this was absorbed in the beta version by putting rotation sensors on the worm drive axles that rotate the grabs. However, the worm screws are a *\*tiny\** bit too short to fit snugly -- they travel a little when the motor direction is reversed. To cure this I tried a [suggestion from John Barnes](#) and cut thin shims out of the plastic insert tray from inside a LEGO box. Two shims on each drive axle fixed the worm gear nicely in position so that the gear lash (although still detectable) was *\*nearly\** within the tolerance of the cube for repeated turning.

However, the rotation sensors for the LEFT and RIGHT grabbers occasionally lost track of their position and had to be manually tweaked during a solution. There was also the problem that the rotation sensors were on the same axle as the worm screw turning the turntable. This meant that when the cube was a little stiff, even if the worm screw had performed the correct number of rotations to turn the cube face  $90^\circ$ , the LEGO pieces of the grabber had enough flex that the grabber was slightly twisted and the face did not make it all the way around to the  $90^\circ$  point.

Therefore, I abandoned rotation sensors and put two touch sensors at the limits of the quarter turn of the turntable (similar to the bottom grabber). I built a 'toucher' attached to the rotating part of the large turntable, and this seemed to compensate better for the twisting of the other LEGO elements of the grabber during stiff turns.

The disadvantage of the touch sensor approach, of course, is that the grabbers can no longer make a full  $180^\circ$  turn, so there is more time taken repositioning the side grabbers. The robot averages one face rotation every 30 seconds (i.e., a 20 rotation solution takes about 10 minutes). The final problem is making sure that the faces of the cube are kept in orthogonal alignment. A standard Rubik's Cube has side dimensions \*just\* larger than 7 LEGO studs. Fortunately there is enough flex in the joints of LEGO Technic to absorb the tiny additional dimension. Each grabber arm is fitted with reverse slopes that force the cube into the correct orthogonal alignment as the grabbers close.



## 5. Establishing the initial (unsolved) state of the cube

The longest part of this project involved writing the color-recognition software. I downloaded the Logitech Quick Cam SDK from the [Logitech Developer's](#) site (the LEGO Vision Cam is a repackaged Logitech Quick Cam) and used VB5 to write a fairly decent program (click the Code link for source). The color recognition is fairly robust (about one error every two cubes when well-calibrated), but not perfect, so I incorporated a feature that requires you to confirm that each face has been correctly scanned (and, optionally, allows you to correct the input manually) before it scans the next face.

The software requires calibration with a solved cube under the particular lighting conditions, and it is quite finicky about changes in lighting conditions. I also left in the earlier manual input option so that you can get a solved cube for calibration, or in case anyone who doesn't have a Vision Cam wants to try this.

Briefly, the software sends a message to the top RCX asking it to present one face of the cube to the video camera. The computer captures a frame from the video camera, and scans a 50x50 pixel area of each color patch to find the median red, green and blue (RGB) color values for each color patch on the face. The RGB values are converted to CIE X Y y coordinates, and then the CIE values are trigonometrically compared to the calibration values to find the closest match. The computer then asks the robot to show it the next face, and the process is repeated until all the faces have been scanned.



## 6. General solution to the Rubik's cube

There are any many general solutions to the  $3^3$  Rubik's Cube on the internet (see [here](#) for a partial list). However, most of these produce a sequence of moves involving 50 or 60 face rotations. Given that CubeSolver moves quite slowly (averages about 30 seconds between face rotations), I wanted a relatively short sequence of moves. Fortunately, I found some [C source code by Mike Reid](#) on the internet which implements Herbert Kociemba's solution method: it provides short solutions (< 40 face rotations). I ported this code to Microsoft Visual C++ and recompiled it as an OCX for use in Visual Basic. I must say that Mike did a very decent job with his code -- move sequences are generated quickly, with a length usually about 20 moves or less.

[Home](#) : [CubeSolver](#) : [code](#) [instructions](#)  
[video](#)

[FAQ](#)

LEGO®, Vision Command, and RCX are trademarks of the LEGO Group of companies which does not sponsor, authorize or endorse this site. Read the full [heavy latin](#) from LEGO for more details. Visual Basic is a trademark of the Microsoft Corporation. Rubik® and Rubik's Cube® are registered trademarks of Seven Towns Limited which does not sponsor, authorize or endorse this site.

Text, graphics, and downloads from <http://jpbrown.i8.com> copyright © [JP Brown](#), 2001-2003. All rights reserved.

822902