

Autonomous Foosball Table

by [eski](#) on December 24, 2007

Table of Contents

Autonomous Foosball Table	1
Intro: Autonomous Foosball Table	2
Step 1: Assembling the motor control board	3
Step 2: Servo Motor Assembly	4
Step 3: Software	5
Step 4: Webcam mount	6
Step 5: Conclusion	6
Related Instructables	7
Comments	7

Intro: Autonomous Foosball Table

The main project goal was to complete a working prototype for an Autonomous Foosball Table (AFT), where a human player faces a robotic opponent. From the human perspective of the game, the foosball table is very similar to a regular table. The player(s) on the human side are controlled via a series of four handles that can be moved in and out and rotated to move the players linearly across the playing field and to kick the ball towards the opponent's goal. The autonomous side consists of:

- > Eight servo motors used to manipulate the handles of the foosball table
- > A microcontroller to activate the servo motors and communicate with the computer
- > An over-head mounted webcam to track the ball and players
- > A computer to process the webcam images, implement artificial intelligence, and communicate with the microcontroller

Budget constraints for the prototype slowed the project some and kept its functionality to a minimum. Proper motors to move the players at a competitive speed were found to be very expensive, so lower-end servos had to be used.

While this particular implementation was limited by cost and time, a larger gear ratio would yield a faster playing robot, although doing so would cost more than the \$500 base price (price without power supply & computer).

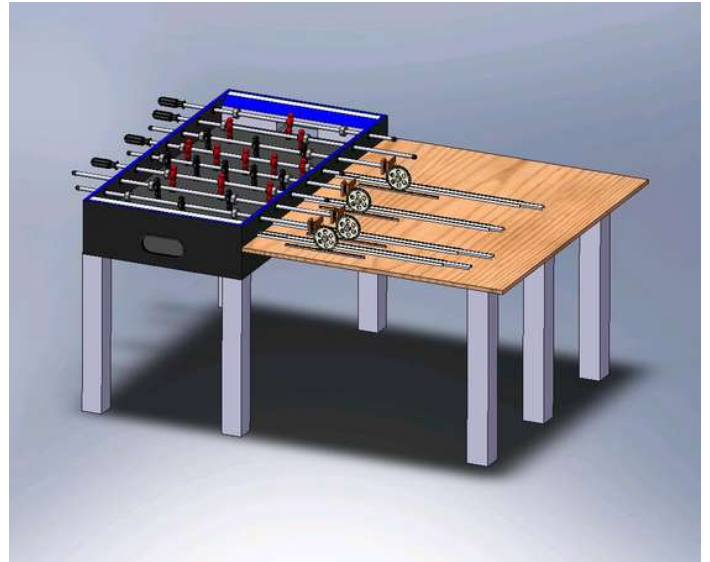
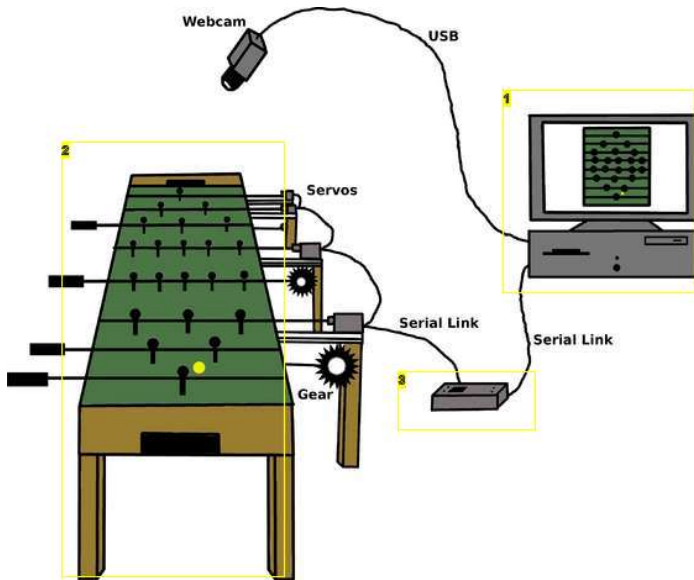


Image Notes

1. Image processing / AI PC
2. Standard foosball table
3. Motor control board

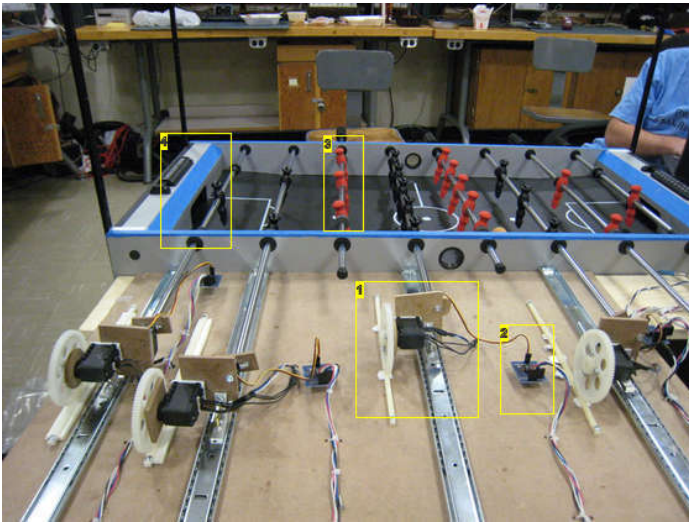


Image Notes

1. Servo assembly
2. PWM control board
3. Human controlled players painted red for higher contrast
4. Blue painters tape is used to create a visual boundary between the outside world and the table

Step 1: Assembling the motor control board

The attached images are a full circuit schematic as well as a picture of the final product for the motor control board. All of these required parts may be purchased at most major online electronics stores (including [Digi-Key](#) and [Mouser](#)). As a side note, all of the parts used here were through hole, and thus, the parts may be assembled on a protoboard/breadboard, or using the attached PCB design. A much smaller package could be created by using a number of surface mount parts.

When we implemented, the design, we split the motor controls into 2 circuits, though there is no advantage to doing so other than any particular cabling scheme used. The small blue board implements the PWM control circuitry, which is basically just a clocked PIC-12F with some specialized code.

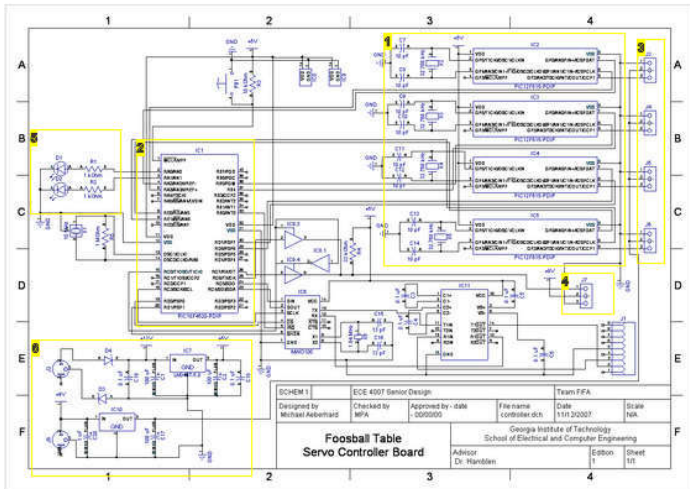


Image Notes

1. PWM servo control modules
2. Master controller (PIC-18F) interprets the data coming from the PC and issues serial communications to the high-torque servos as well as a bitwise control scheme for the PWM servos
3. Connectors to the PWM servos
4. Connector to the high-torque servos (all 4 share a single serial line)
5. LED indicators
6. Voltage regulation from the power supply

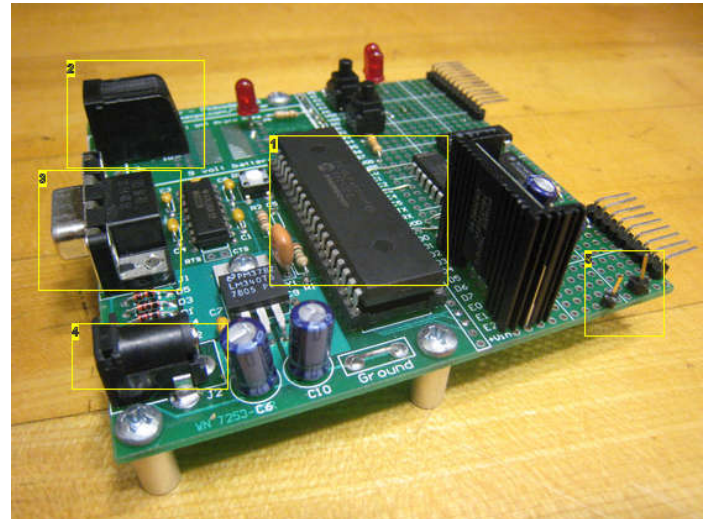


Image Notes

1. Main microcontroller (PIC-18F)
2. Standard phone jack connector used for PIC-18F programming
3. RS-232 physical connector for serial data communications between PC and motor control board
4. Power supply connector
5. 9.6V input for servo power

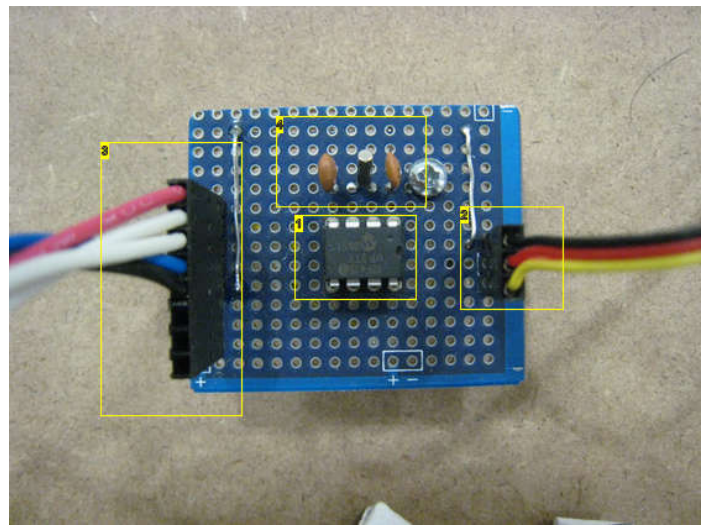
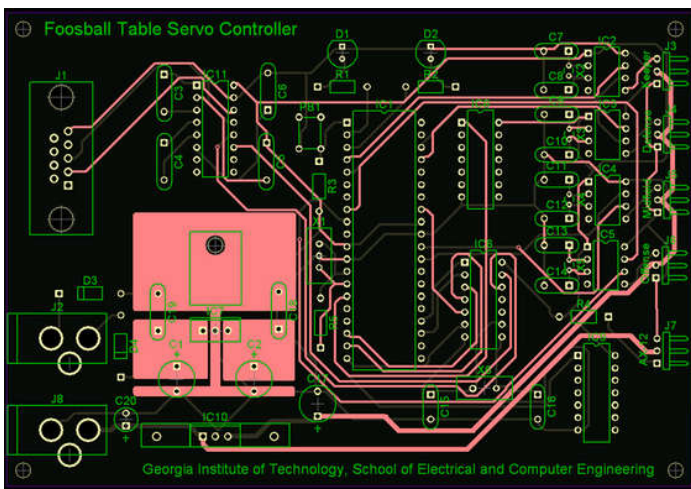


Image Notes

1. PIC-12F
2. Outward connector to servo
3. Inbound connector from master motor control board
4. Simple oscillator used by the PIC-12F to determine timing

Step 2: Servo Motor Assembly

Two different types of servos are used. First, the lateral motion is controlled by a group of four high-torque servos: Robotis Dynamixel Tribotix AX-12. These four run on a single serial line and provide amazing functionality. The high torque enables these servos to be geared in such a way that provides a high tangential velocity for lateral movement. We were able to find a set of 3.5 inch gears and tracks to go with them from Grainger at a cost of about \$10 for each. The servos provide torque overload protection, an individual servo addressing scheme, quick communications, internal temperature monitoring, two-way communications, etc. The downside of these servos is that they're expensive and not very fast (though gearing helps them). So in order to get faster motion for kicking, Hitec HS-81s are used. The HS-81s are relatively inexpensive, have a decently fast angular velocity, and are easy to interface (standard PWM). The HS-81s only rotate 90 degrees, however (though it is possible -- and not recommended -- to attempt to modify them to 180 degrees). Additionally, they have internal nylon gears that strip easily if you attempt to modify the servo. It would be worth the money to find a 180 degree rotating servo that has this type of angular velocity.

The entire system is tied together with pieces of medium-density fiberboard (MDF) and high-density fiberboard (HDF). This was chosen for its low cost (~\$5 for a 6'x4' sheet), ease of cutting, and ability to interface with virtually any surface. A more permanent solution would be to machine aluminum brackets to hold everything together. The screws holding the PWM servos in place are standard machine screws (#10s) with hex nuts holding them from the other side. 1mm metric machine screws, about 3/4" in length, hold the AX-12 into the MDF that connects the two servos together. A double-action drawer track holds the entire assembly down and in-line with the track.

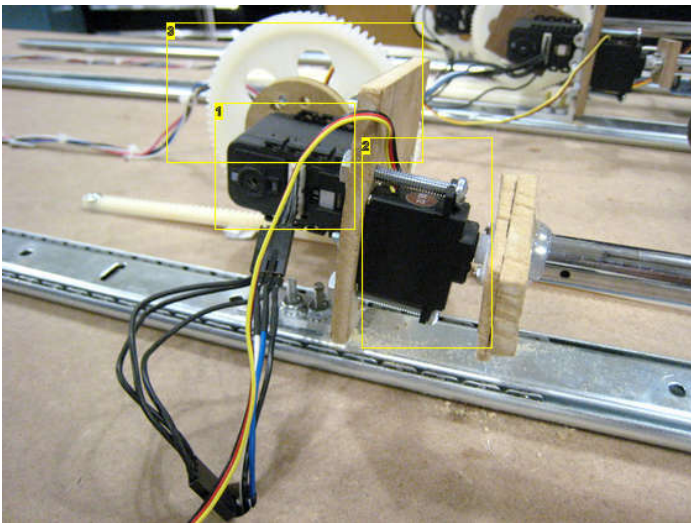


Image Notes

1. AX-12 servo motor (high torque)
2. HS-81 PWM servo (high speed)
3. 3.5 inch gear

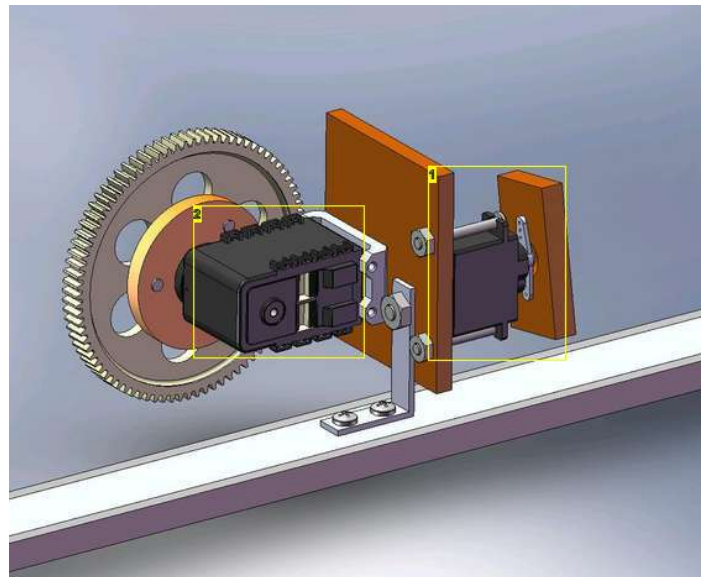


Image Notes

1. HS-81 PWM servo motor (high speed)
2. AX-12 servo motor (high torque)

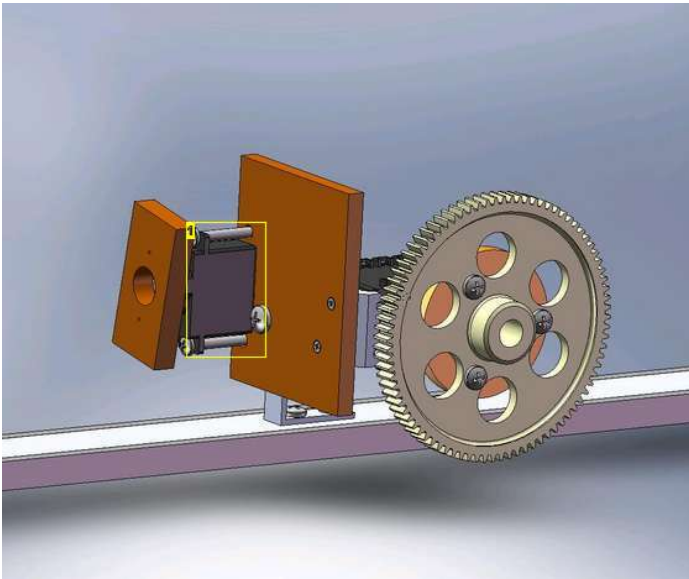


Image Notes

1. HS-81 PWM servo motor (high speed)

Step 3: Software

The last step is to install all of the software used on the machine. This consists of a few individual pieces of code:

- > The code run on the image processing PC
- > The code run on the PIC-18F microcontroller
- > The code run on each of the PIC-12F microcontrollers

There are two prerequisites to install on the image processing PC. The image processing is done through the Java Media Framework (JMF), which is available through Sun [here](#) . Also available through Sun, the Java Communications API is used to communicate to the motor control board, across the serial port on the computer. The beauty of using Java is that it "should" run on any operating system, though we used *Ubuntu* , a linux distribution. Contrary to popular opinion, the processing speed in Java isn't too bad, especially in basic looping (which vision analysis uses quite a bit).

As seen in the screenshot, both the ball and the opponent players are tracked on each frame update. In addition, the outline of the table is located visually, which is why blue painters tape was used to create a visual outline. Goals are registered when the computer cannot locate the ball for 10 consecutive frames, typically indicating the ball fell into the goal, off the playing surface. When this happens, the software initiates a sound-byte to either cheer itself or boo the opponent, depending on the direction of the goal. A better system, though we did not have the time to implement it, would be to use a simple infrared emitter/sensor pair to detect the ball falling into the goal.

All of the software used in this project is available in a single zip file, [here](#) . To compile the Java code, use the **javac** command. The PIC-18F and PIC-12F code is distributed with Microchip's MPLAB software.



Step 4: Webcam mount

A Philips SPC-900NC webcam was used, though it is **not recommended**. The specs for this camera were falsified either by the engineering or sales staff at Philips. Instead, any cheap webcam would do, so long as it is supported by the operating system. For more information regarding the use of webcams under linux, check out [this page](#).

We measured the distance required by the webcam's focal length to fit the entire foosball table in the frame. For this camera model, that number turned out to be just over 5 feet. We used shelving racks available from any major hardware store to build a mount for the camera. The shelving racks extend upward from each of the four corners of the table and are cross-braced by angled aluminum brackets. It is very important that the camera is centered and has no angular rotation, since the software assumes the x- and y-axis are aligned to the table.

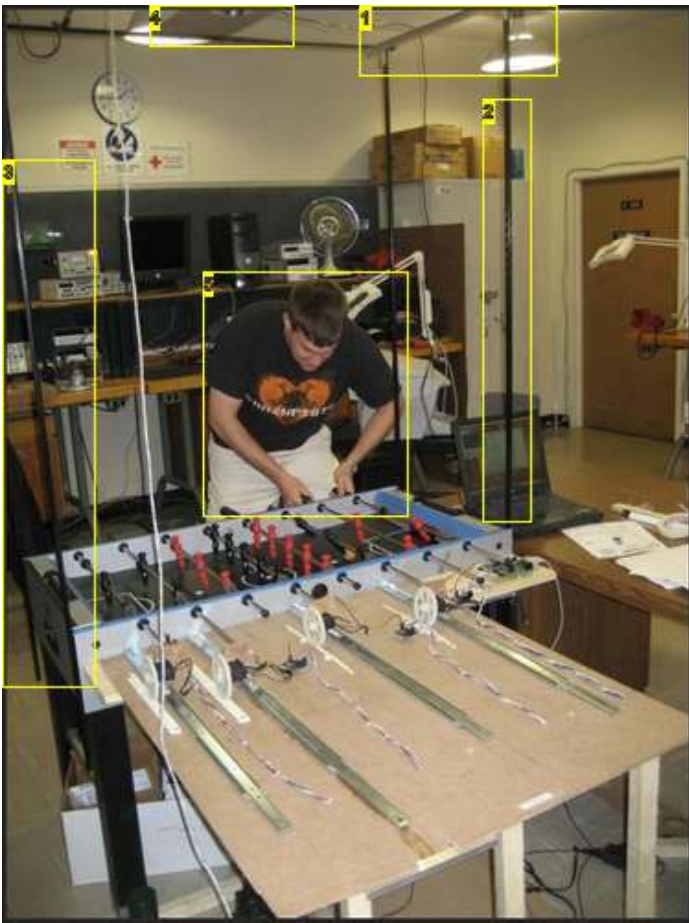


Image Notes

1. Cross-brace for shelving poles
2. Shelving pole
3. Shelving pole
4. Webcam
5. Unworthy human opponent

Step 5: Conclusion

All of the related project files can be downloaded at [this site](#). A backup of the majority of the site content can be found [here](#), at my personal web host. This includes the final report, which has a marketing analysis as well as things we would change, our original goals and a list of what specs were actually attained. The project is NOT meant to be the most competitive player in the world. It is a good tool to show more of the steps used in designing such a beast as well as a decent prototype of this type of robot built for an incredibly low cost. There are other such robots in the world, and certainly, many of them would "beat" this robot. This project was designed by a group of four electrical/computer engineers at Georgia Tech as a senior design project. No help was received by any mechanical engineers and no third-party funding was used. It was a great learning process for us all and a decent use of senior design course time.

I would like to thank

> **Dr. James Hamblen**, our section advisor, for his continuous help in technical strategies

> **Dr. Jennifer Michaels**, the lead professor, for not discouraging us from attempting a more ambitious project

> **James Steinberg and Edgar Jones**, the senior design lab administrators, for constant help in ordering parts, troubleshooting, and the finding the "cool stuff" to throw into the project at low cost and high functionality

> And of course, the other three members of my team, of which, none of this would have been possible: **Michael Aeberhard, Evan Tarr, and Nardis Walker**.

Related Instructables



Build an autonomous Wall-E Robot by djsures



BiPed robot V-3 by alex.v



Make a scary scarab robot by djsures



Inexpensive Autonomous Humanoid Robot (Photos) by chrisvo



DIY Mod an Omnibot 80's Robot with Voice, Camera, Servos, Bluetooth by djsures



Low Cost Hobby Servo XY Table by CarlS

Comments

40 comments [Add Comment](#)



nuckthebuck says:

Apr 15, 2011. 9:07 PM [REPLY](#)

I always loved autonomous objects...especially competitive ones. some other organizations that have done autonomous objects are FIRST ROBOTICS CHAMPIONSHIPS, and DARPA. check these out and fall in love... or look at the popsci.com for amazing news about technology. i am a electrical engineer from the first robotics championships. currently a student soon will mentor other students in programming, wiring, and electronics. and I hope to join the DARPA teams someday



jerkey says:

Dec 11, 2010. 6:12 AM [REPLY](#)

This is a really cool instructable. Thank you for showing people how to do something so advanced. With understanding of modern technology, anything is possible and we can begin solving the problems of the world!



Shut Up Now says:

Mar 4, 2008. 5:05 PM [REPLY](#)

very nice work. this is a proof of concept and someday you may be able to use a system that could beat a descent human opponent. maybe strong magnets placed in a certain order of polarity at set distances inside the poles of the foosball table could be paired with a computer controlled electromagnet in order to move the poles. ur reply to this will probabably make this idea look stupid.. oh well



Shut Up Now says:

Mar 4, 2008. 5:07 PM [REPLY](#)

edit: in my above post it says computer controlled magnet, by that i mean a set of magnets for each pole.



jerkey says:

Dec 11, 2010. 6:09 AM [REPLY](#)

no there are only magnets with both poles, north and south. unipolar magnets have so far not been discovered.



dnicky2288 says:

Mar 22, 2009. 10:29 PM [REPLY](#)

Let me be the first to say that your project is so cool! I am quite a good programmer myself, but this is kind of advanced which is very interesting to me. I have a web cam, had c++ Programming before, and have a Foosball table. I am complete! I can make my own one now! Thanks for the help. I'm sure "Eric Wilhelm" will love to hear about this. I will spread the word for you. Again this is really awesome!



iansilv says:

Jan 25, 2009. 4:56 PM [REPLY](#)

Could the author elaborate on what gears / motors / other parts could be improved on if the budget fo rthe project could be exceeded? For example,w hat model of motor to use for a faster player movement, or what model of server will do the full 180 degree kick?



ander says:

Jan 21, 2009. 11:39 PM [REPLY](#)

My gosh, this is great—and how generous of you to go to the considerable trouble to make an Instructable so you could share it with everybody. To those of you complaining that the AI can be improved—well, build one, improve it, and post about it here! (How do you spell "collaboration"?)



learningmasters says:

Sep 9, 2008. 4:09 PM [REPLY](#)

can you make it track a person???



ultrauber says:

Aug 21, 2008. 1:45 PM [REPLY](#)

That's quite a mind-boggling schematic, especially since I can't even solder.



YakAttack says:

Jul 11, 2008. 7:34 PM [REPLY](#)

Excellent job guys! Vision with mechanics always presents extra trouble! I worked on a similiar project myself with some classmates in college: RoboPoolTable



Q branch says:
nice!!! but it seems to lag a little.

Jul 4, 2008. 2:35 PM [REPLY](#)



onlyonebowman says:
Great Job

Jun 4, 2008. 5:58 PM [REPLY](#)



jtkawasaki8 says:
i give u major props for makeing that

May 18, 2008. 8:24 PM [REPLY](#)



=SMART= says:
wow looks very complicated but it is definitely worth it coz it sooo cool!

Apr 26, 2008. 12:26 PM [REPLY](#)



etarr says:
I think there's a typo on this page. You said something about thanking the other 'three' members of your team.

Jan 11, 2008. 5:45 AM [REPLY](#)



peasant54 says:
"> And of course, the other three members of my team, of which, none of this would have been possible: Michael Aeberhard, Evan Tarr, and Nardis Walker." No typo. the three members are listed clearly, I don't get what you mean. The three first thanked persons were not in the team.

Apr 22, 2008. 4:59 AM [REPLY](#)



eski says:
etarr was one of the members on the team (evan tarr). his comment was a joke, as one of the members did a bit less than the others ;)

Apr 22, 2008. 2:42 PM [REPLY](#)



biolethal says:
Uhh... what happened to the video? It says that the video isnt available anymore.

Mar 18, 2008. 5:54 PM [REPLY](#)



aki12345 says:
can that program be used to track anything? if yes, how do I compile it, or how do I use it? I don't want the output part, just the tracking

Mar 1, 2008. 8:31 AM [REPLY](#)



awesome says:
This is impressive!

Feb 28, 2008. 11:14 AM [REPLY](#)



fukenofooser says:
Very nice work! Getting a robot to even hit a ball is impressive. I've seen many people that couldn't do that! But, being a Foos junkie, I'm pretty sure any players could skunk it 9-0. For a real tracking system vision wouldn't work, real shots can not be seen, you have to know where it has to go and race it or block the long leaving the short or middle wide open. But, as I said, Impressive work!

Feb 28, 2008. 10:32 AM [REPLY](#)



golfball says:
Cool stuff the amount of work you have put in to it shows

Jan 24, 2008. 1:48 AM [REPLY](#)



furcino says:
Very nice of you to release the source, I look forward to reading it. Well, I understand that it can't work too fast with a just a webcam locating the ball. Most of these use infrared.

Jan 20, 2008. 10:12 AM [REPLY](#)



orangeontoast says:
what does this have to do with a laser?

Jan 17, 2008. 12:48 PM [REPLY](#)



Charl says:
Well done- Look awesome, Could be a bit smother with a faster camera. :-)

Jan 11, 2008. 7:51 AM [REPLY](#)



andrew101 says:
sweet

Jan 10, 2008. 7:35 PM [REPLY](#)



KentsOkay says:
Awesome! Now if only I had the stuff to build it...

Jan 7, 2008. 3:48 PM [REPLY](#)



Doctor What says:
Looks fun, but I don't have the money for such a project, maybe when I am slightly older, and have sustainable employment (and my parents don't make me put my money earned in the bank).

Jan 1, 2008. 6:29 PM [REPLY](#)



eski says:
@kkinney:
The robot is very limited by the mechanical design, but it's still pretty intimidating playing against it. It does a pretty good job defensively, as the software attempts to cut off the angle to the goal. Unfortunately, the offensive capabilities are rather unimpressive. The kicks are a little too slow and inaccurate.

@jcampbell:
As exciting as robot-on-robot action sounds, it would be pretty silly to have 2 of the same robots playing against each other. I think it would be interesting, however, to make a competition like *robocup*, but with foosball

Dec 28, 2007. 12:06 PM [REPLY](#)



kkinney says:
Spiff! How well does it do against a moderate carbon based player?

Dec 27, 2007. 6:11 PM [REPLY](#)



jcampbell says:
So when do you build a second unit so two machines can play against each other? Sounds like a setting for some fun research...

Dec 27, 2007. 12:58 PM [REPLY](#)



Bor says:
Is there a way to run the software on a Mac?

Dec 27, 2007. 6:38 AM [REPLY](#)



eski says:
The software is written in Java, which is operating system independent. However, it utilizes the Java Media Framework and the Java Serial Communications API. Anytime you use something like that for hardware controls, it tends to get more OS-dependent. We didn't test the software on a windows PC or a Mac because we had no reason to. Some software changes may be necessary for a Mac implementation, though they are Unix-based, so it may not be too significant.

Dec 27, 2007. 10:59 AM [REPLY](#)



eski says:
We came up with the idea in the first week or so of the semester and announced our intentions by the first of September. We did a demo of the final project December 8. So a about 3 months from concept to implementation with a 4 person team. There was a lllloooottt of writing for the course though, so I would say only about 70% of our time was doing technical stuff.

Dec 26, 2007. 9:02 AM [REPLY](#)



sardines454 says:
that's incredible. how long did it take you to come up with idea and make it?

Dec 26, 2007. 5:47 AM [REPLY](#)



joejoerowley says:
I heard about something like this at an MSOE camp this summer. Really Cool!

Dec 25, 2007. 12:42 PM [REPLY](#)



Bor says:
it's awesome!

Dec 25, 2007. 7:16 AM [REPLY](#)



GorillazMiko says:
Awesome, looks amazingly difficult..

Dec 24, 2007. 11:10 PM [REPLY](#)



Weissensteinburg says:
This is awesome - Great job!

Dec 24, 2007. 10:33 PM [REPLY](#)