Here is a parts list for the main components of the Skittles Sorting Machine:

Epoxy: http://www.abatron.com/shop-online/woodepox.html
Camera: http://parallax.com/product/28302
Camera cable adapter: http://parallax.com/product/28325
BASIC Stamp micro controller: http://parallax.com/product/bs2-ic
BASIC Stamp board w/ Servo driver: http://parallax.com/product/28850
BASIC Stamp Editor Program Download: http://parallax.com/downloads/basic-stamp-editor-software
Power supply: http://parallax.com/product/750-00009
Arm Servo: http://parallax.com/product/900-00005
Wheel and Hopper Servos (2x): http://parallax.com/product/900-00008
Funnel: http://www.homedepot.com/p/Perky-Pet-Magnolia-Top-Fill-Hummingbird-Feeder-with-Free-Nectar-120TFN/202692316
IR LED/Receiver: http://www.radioshack.com/radioshack-infrared-led-emitter-and-detector/2760142.html#.VIHUy1a4mlI

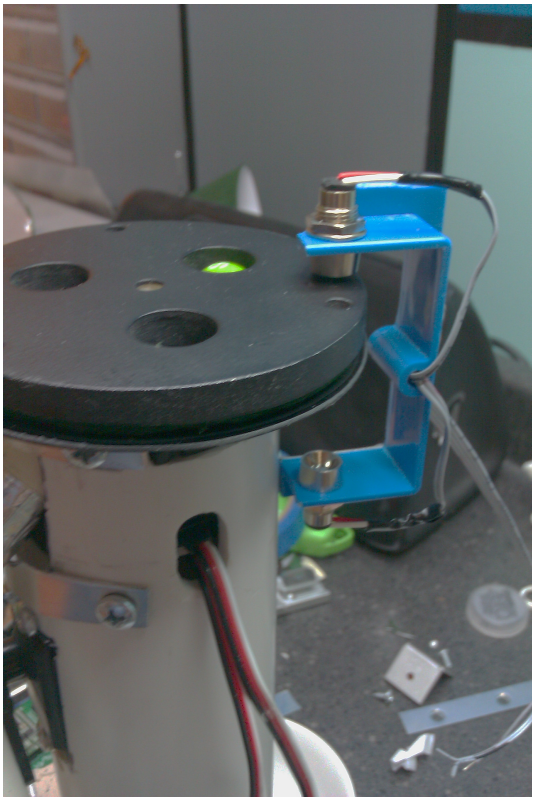
The epoxy makes up the outer case, inner covers of the micro controller, hopper floor, hopper blade, turnstile, and the dispenser tubes. The epoxy takes practice to work with. If you use this one, I recommend the Quart size. It comes with 2 quarts.
The funnel is a modified hummingbird feeder. (I cut the bottom off).
The center part and all of the tubing is PVC.
The base is wood and the bowls are ceramic.
The IR LED/Receiver looks for the 3 small holes in the turnstile (rotating disk) to stop it when it is in the right place.
The arm is made from part of an old telescope, but thin-walled PVC should work. It is attached to the servo (the one I called Arm Servo above)
The continuous rotation servos power the turnstile (candy wheel) and hopper blade, which agitates the candy in the funnel to keep them going into the tube and down to the turnstile.

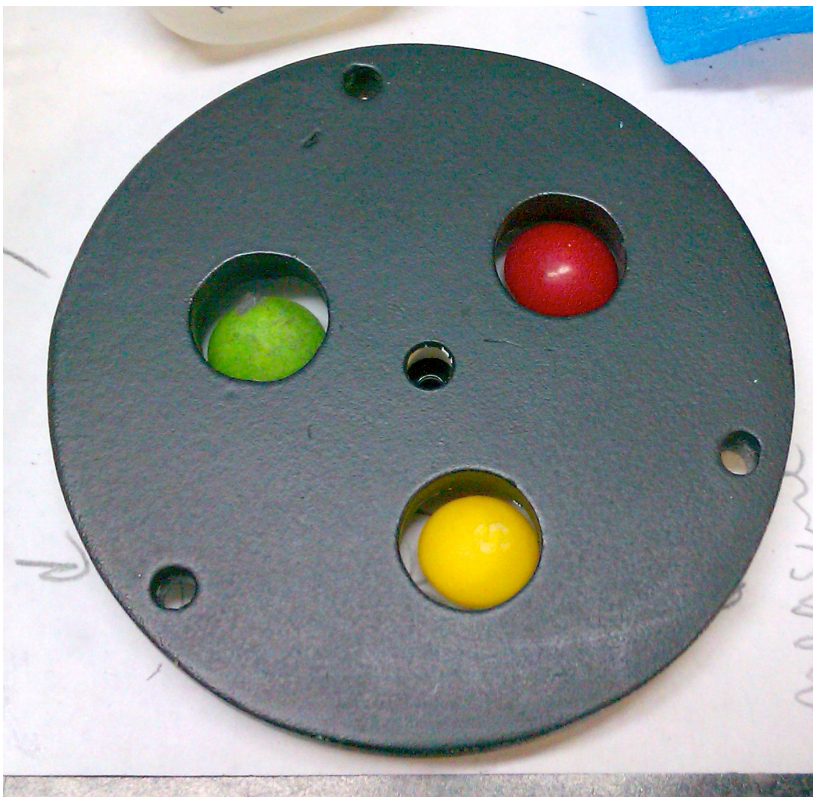
Critical things:

Care must be taken on the inner diameter of the tube the skittles go through at the bottom of the hopper (see below). If it is too wide, 2 skittles fall in sideways and get stuck. If it is too narrow, larger skittles will get stuck. I used a thin plastic sheet to help. I rolled it up inside the tube to make it the correct width. (Shown on right)

The new epoxy hopper blade is shown above on the left. The old one (center, right) was metal and poorly shaped; it didn't work well.

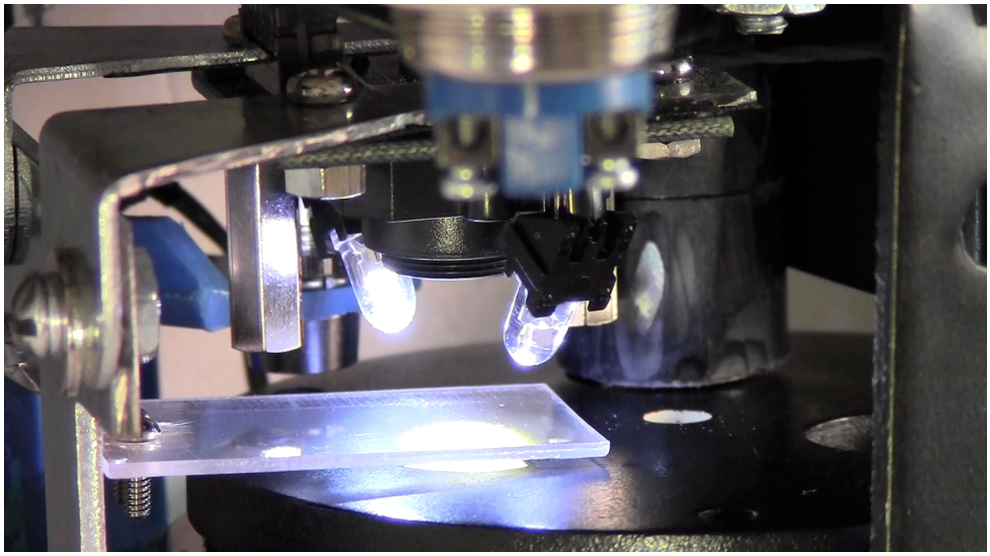
Here is the turnstile. It's made from epoxy and mounted on a continuous rotation servo that's in the PVC body. The cables for it and the arm servo can be seen coming out of the side of the PVC below. There is a support plate below it made from black matte plastic. The turnstile rides on this with felt that is adhered to the bottom of the turnstile. This takes stress off the servo; it should only rotate the disk, not support it.



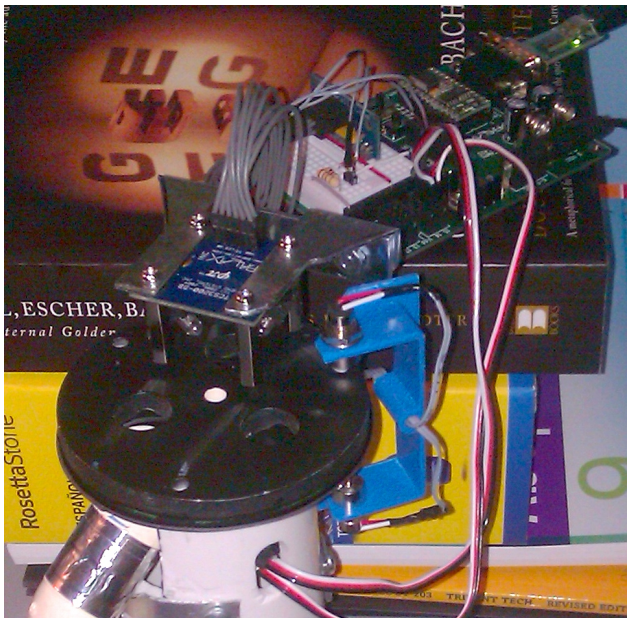


The wheel and interior parts are painted matte black for a reason. It is required to work right. Any other color (or white) will saturate the sensor.
There is a problem of specular reflection and also what I call the "S problem" which is the fact that there's a white "S" on one side of a Skittle (similarly a logo appears on one side of an m&m) which tricks the sensor. The sensor gives back Red, Blue, Green values from 0 to 255, which you must use to determine if a candy is a particular color. This is done by first building the machine, programming it to move and stop in the right place, and to display the RGB values on an output screen. This is quite simple on a BASIC Stamp, using the DEBUG function and is done in the code provided on my website. The "S problem" is that a white S gives significantly higher readings in the color spectrum. What's worse is that depending on the color of the candy and how well the "S" was printed on it, this change in the reading varies. In short, it is an unknown. The specular reflection I mentioned is that these candies start out quite shiny. If you handle them awhile, they become dull (as seen above on the right…yuck!) The camera will see them quite differently in both cases. Since it uses a white LED to illuminate them, in the shiny case it sees a white reflection from the LEDs on the camera, increasing the readings in a way that *strongly* depends on the orientation of the candy. So if the candy is tilted up a little in the turnstile hole

way or laying flat. This is another unknown, and it can be surprising how such a good sensor (and it is very good in my opinion) can get such drastically different readings for the same piece of candy. With both of these problems in hand, even if you can stop the turnstile in *exactly* the same place each time, the same piece of candy will still get different readings. The worst part is that the color space for two (or more) candies will overlap. This means yellows can look orange, reds can look purple, etc. If left unresolved, these problems make reliable color sorting with this machine impossible.
To (almost completely) alleviate the problems mentioned above, I came up with the solution of a diffuser lens over the hole below the camera, shown below.



It is small piece of glare-resistant Plexiglas. It looks slightly frosted and serves to average out, or diffuse, the light from the candy. It also helps keep the specular reflection constant, since the camera looks at it, not the candy, and it is always orientated the same way. There is a small strip of black felt at the end to avoid scratching the turnstile. It was implemented in the 3rd revision of the machine.



Here is a picture of how the color sensor is mounted. Notice the 3 holes for candy stop in the same place every time. One hole is under the feeding tube, the other hole is under the camera, and the 3rd hole is over the chute. The blue bracket holds the IR LED and Phototransistor.
If you watch the older videos you can see the insides better. The latest version of the machine hides all this stuff.

See my website for other pictures.

The source code is available on my website as well. This code allows for the sorting of m&ms also, and can be modified through cut/paste to include other candy provided you get the RGB values from the debugger on the BASIC Stamp. The code has the debugging already in it and you only need to run the debugger in the programming environment. This requires more inputs to the controller to determine which kind of candy will be sorted. I don't detail that selector circuit here, but it's fairly simple. The only caveat is that there are limited inputs to the BASIC Stamp and to allow 4 inputs, the code is written with a binary code (3 inputs allow for up to 8 selections, 4 of which are assigned in code)
As built, with the source code provided, it will sort Skittles only. No selection is the same thing in the code as selecting Skittles. You will only need to leave Pins 6,7, & 8 disconnected, or grounded low. If you want to use these pins for other things, reassign them in the I/O definitions and delete the part of the initial loop that defines the "select" variable. If you get stuck, email me and I'll try my best to help.

-Brian Egenriether