Pick & Place Using the Silhouette Curio

Introduction

This document explains the hardware, software, and procedure to use the Silhouette Curio as a pick and place machine 'assistant' for surface mount assembly into printed circuit boards. The setup requires:

a) Silhouette Curio craft cutter with a deep cut blade and a pen holder.
b) A vacuum pump assembly with a controller.
c) A pick-up tool made using a 0.5 mm mechanical pencil.
d) A computer running Windows or Linux with Python 3 installed as well as two USB ports available. A driver needs to be installed on Windows in order to communicate with the Silhouette Curio.
e) Materials to build a foam gig to hold the carrier tape and PCBs. These include: a cutting mat, 2 mm craft foam, double side Duck tape or similar, thin double sided tape, and a Sharpie marker.

Each of the points above is explained in the sections that follow.

During the development of this simple pick and place setup the author was able to successfully assemble more than 400 small boards. The setup has the following features and limitations:

a) It can pick and place between 10 and 12 parts per minute.
b) It can pick and place 0805 components. It may be able to pick and place smaller components but there were not designs and parts available to test this during development.
c) One important limitation of this setup is the 5 mm clearance allowed by the Silhouette Curio. In theory this will limit the height of any component to about 2.5 mm.
d) Although the setup doesn't include the ability to rotate components, pre-rotated components in any angle can be setup in the craft foam gig.
e) PCB size and the number of PCBs per job are limited by the size of the Curio tray. There are two trays sizes available for the silhouette Curio. The small tray has a usable area of 21.6 cm x 15.3 cm, while the big tray has a usable area of 21.6 cm x 30.6 cm. Although support for the big tray is in the Python software, it has not been tested yet.
f) Only rectangular PCBs are supported so far. This may change with future versions of the software.
g) This pick and place setup is 'blind': doesn't include computer vision. For this reason small errors may occur during pick and place. These errors must be fixed before reflow soldering the PCBs. Think of this pick and place setup as an 'assistant' more than a fully automated pick and place system.

License

Vacuum Pump and Controller

The vacuum pump and controller are used to pick up the components from the cut tape and hold them until placed in the PCB by the Silhouette Curio. The vacuum pump controller may use an Arduino board and is controlled via the serial port.

Parts and Materials

1) One 12V power adapter. Any DC 12V power adapter capable of delivering about 2.0A with a barrel connector that matches the Arduino board should do.

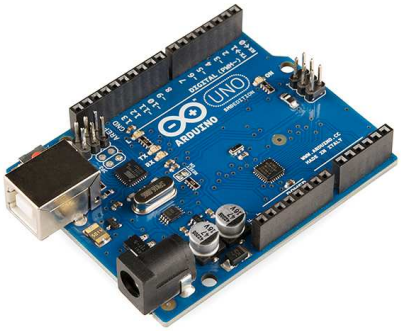
2) One 12V vacuum pump. I tested successfully both the “AIRPRO Model D2028B” and the “Micro Air Pump SC5002PM”. Either one will do. I bought both pumps at my local electronics store, but they are available in several places over the internet, including Amazon. Another place that sells one of these vacuum pumps is Sparkfun; the part number is ROB-10398.

3) One 12 Solenoid valve, normally closed. The one I used is the “Solenoid Valve 12VDC 34 Ohms 4PSI” from Jameco. It is very similar to the "Keurig 12V Pneumatic Solenoid Air Valve 4 PSI SH-V0829 normally closed" that shows up in Amazon and eBay. I also tested some Keuring valve replacements from Amazon and they work fine as well.

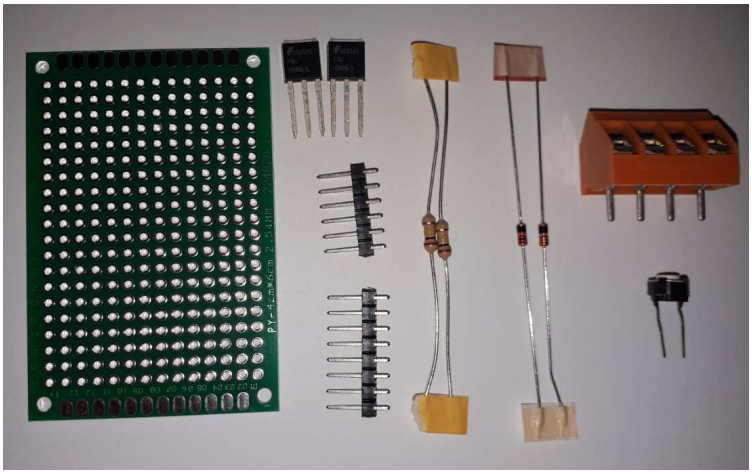
4) Vinyl tubing. The one I used is “Clear Vinyl Tubing, 0.17 Inch Inside Diameter X 1/4 Inch Outside Diameter” from Home Depot. Around two meters should be enough.

5) A couple of ‘T’s for drip irrigation. The ‘T’s should work with the tubing above. Amazon seems to carry them. One ‘T’ is need for the pick-up tool.

6) Arduino Uno. I don’t remember where I got mine. It is very old. I am not a big fan of the Arduino boards, but on the other hand I am aware that it is very common and easy to find, so I am using it in this project.



7) MOSFET driver for Arduino. My implementation requires: 1 x prototype board, 2 x N-MOSFET (rated at 4A or more, I used the FQU13N06L), 2 x small diodes (I used the 1N4148), 2 x 100kΩ resistors, 1 x push button, 1 x 6 pin male header connector, 1 x 8 pin male header connector, and 1 x 4 terminal block connector. This is how my parts looked like before assembling the driver board:



You can probably use the “SparkFun Power Driver Shield Kit” with very simple modifications to the code I wrote as they used different pins to connect to the gates of the 12V MOSFETs. The kit is available here:

https://www.sparkfun.com/products/10618

Another possible board you can use to control the vacuum pump and solenoid valve is the Velleman “Motor and Power Shield for Arduino”. As with the previous board, the provided sketch will need to be modified to account for the different Arduino board pins used.

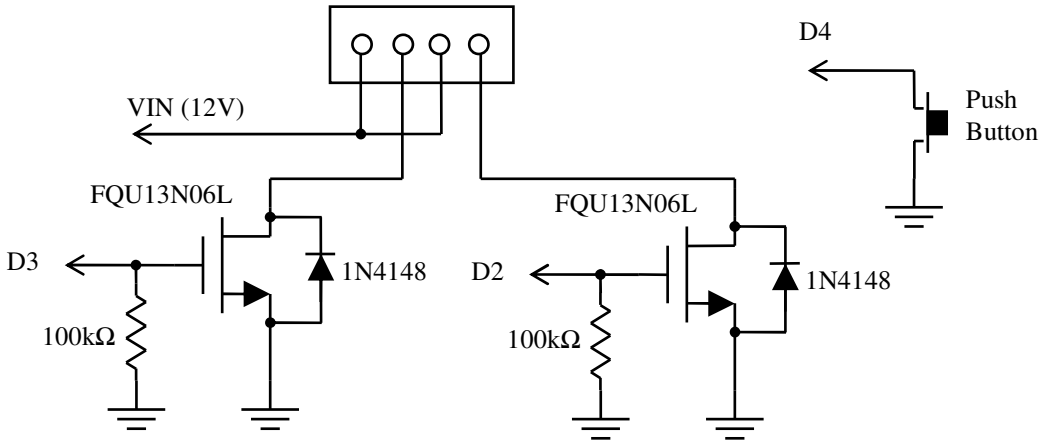
https://www.velleman.eu/products/view/?id=412174

8) Wire, both solid and stranded. The solid wire is for the prototype board. The stranded wire is for the vacuum pump and solenoid valve.

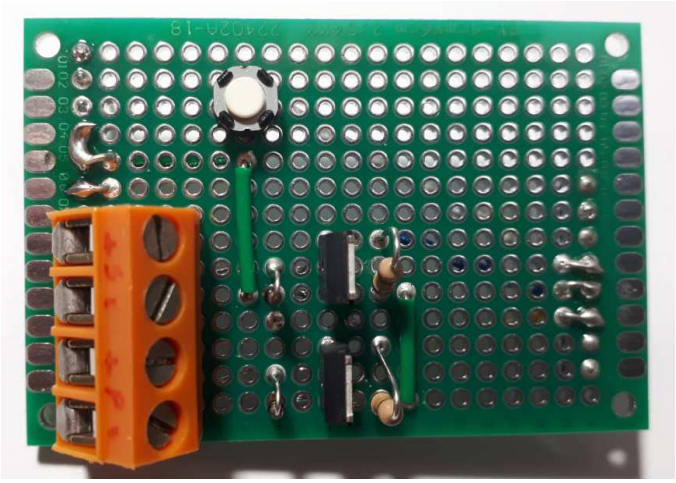
9) Solder, soldering iron, tools, etc.

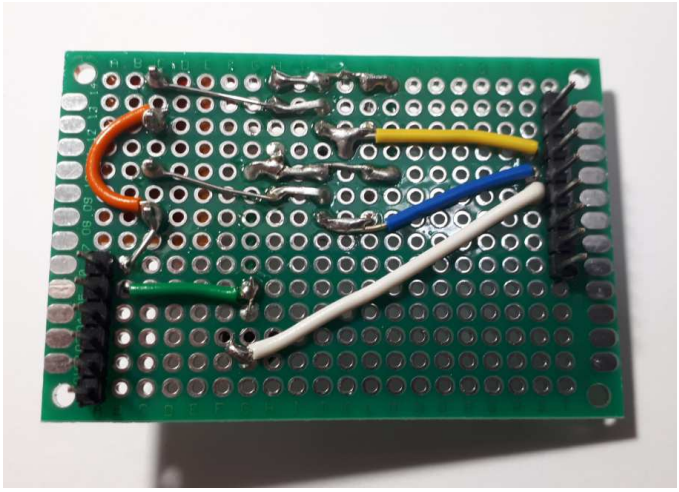
Assembling a Vacuum/Valve Driver board

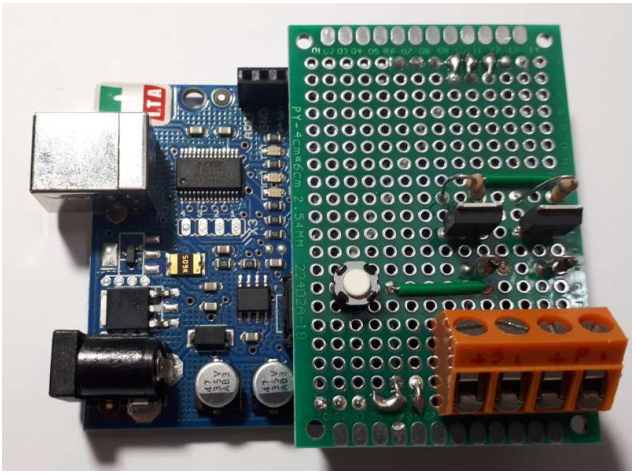
The driver board is used to turn on/off both the vacuum pump and the solenoid valve with an Arduino board via commands received from the serial port. This is the schematic diagram of the driver board:



The pictures below show the completed driver board: top side, bottom side, and the driver board attached to the Arduino board.







Arduino Sketch for the Vacuum Pump and Solenoid Valve Controller

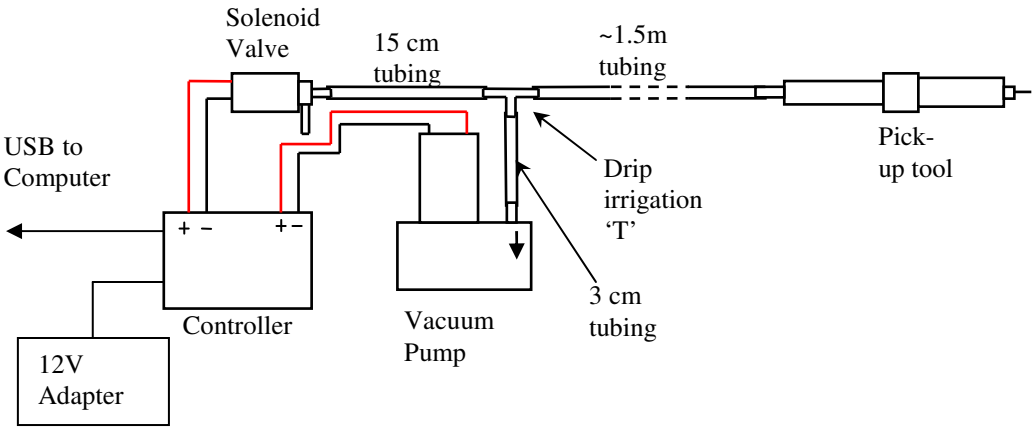
Shown below is the source code for the controller. Compile and load it to the Arduino board.

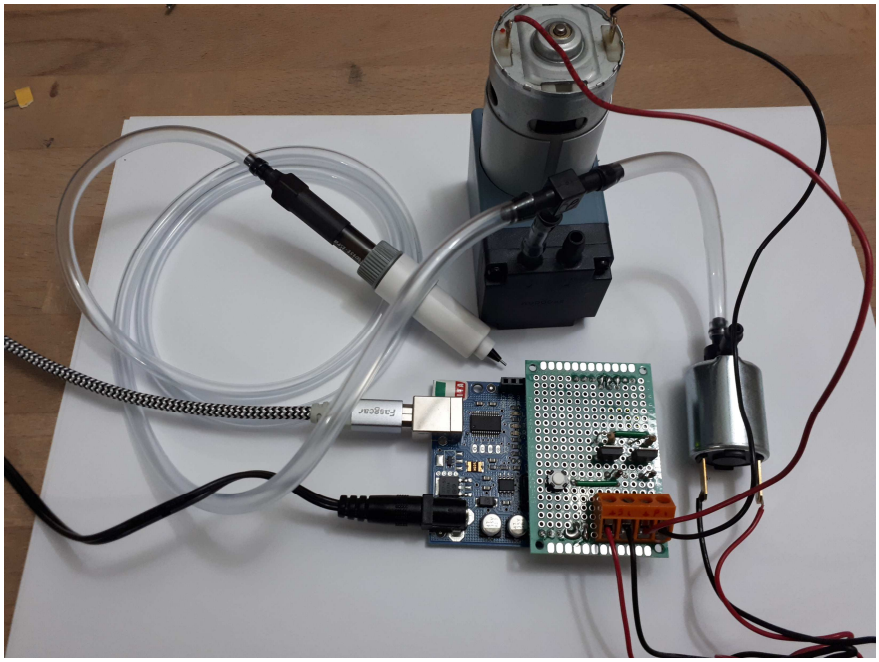
```
#define Vacuum 2
#define Valve 3
#define PushButton 4

int c = 0; // for incoming serial data
int ton=0;
int toff=0;
String incomingString;

void setup()
{
    Serial.begin(115200); // opens serial port, sets data rate to 115200 bps

    pinMode(Vacuum, OUTPUT);    // sets the digital pin 2 as output
    pinMode(Valve, OUTPUT);    // sets the digital pin 3 as output
    pinMode(PushButton, INPUT_PULLUP); // To manually turn the vacuum pump on

    digitalWrite(Vacuum, LOW); // Vacuum pump off
    digitalWrite(Valve, LOW); // Vent valve closed
}

void Pump (int value)
{
    if(value==1)
    {
        digitalWrite(Vacuum, HIGH); // Vacuum pump on
        digitalWrite(Valve, LOW); // Vent valve is closed
    }
    else
    {
        digitalWrite(Vacuum, LOW); // Vacuum pump off
        digitalWrite(Valve, HIGH); // Vent valve is open
        delay(250); // Wait for pressure to equalize
        digitalWrite(Valve, LOW); // Vent valve is closed
    }
}

void loop()
{
   if (Serial.available() > 0)
   {
       c = Serial.read(); // read the incoming byte

       if(c=='0') // Vacuum off
       {
           if(toff!=0) delay(toff);
           Pump(0);
       }
       else if (c=='1') // Vacuum on
       {
           if(ton!=0) delay(ton);
           Pump(1);
       }
       else if (c=='t') // Delayed action 'off' time in ms
       {
           incomingString = Serial.readString();
           toff=incomingString.toInt();
       }
       else if (c=='T') // Delayed action 'on' time in ms
       {
           incomingString = Serial.readString();
           ton=incomingString.toInt();
```

```
        }
        else if ((c=='I') || (c=='i')) // Identify command
        {
            Serial.write('P');
        }
    }

    if(digitalRead(PushButton)==LOW) // External push-button pressed?
    {
        delay(50); // Debounce time
        if (digitalRead(PushButton)==LOW)
        {
            Pump(1);
            while(1)
            {
                if(digitalRead(PushButton)==HIGH)
                {
                    delay(50); // Debounce time
                    if(digitalRead(PushButton)==HIGH) break;
                }
            }
            Pump(0);
         }
    }
}
```

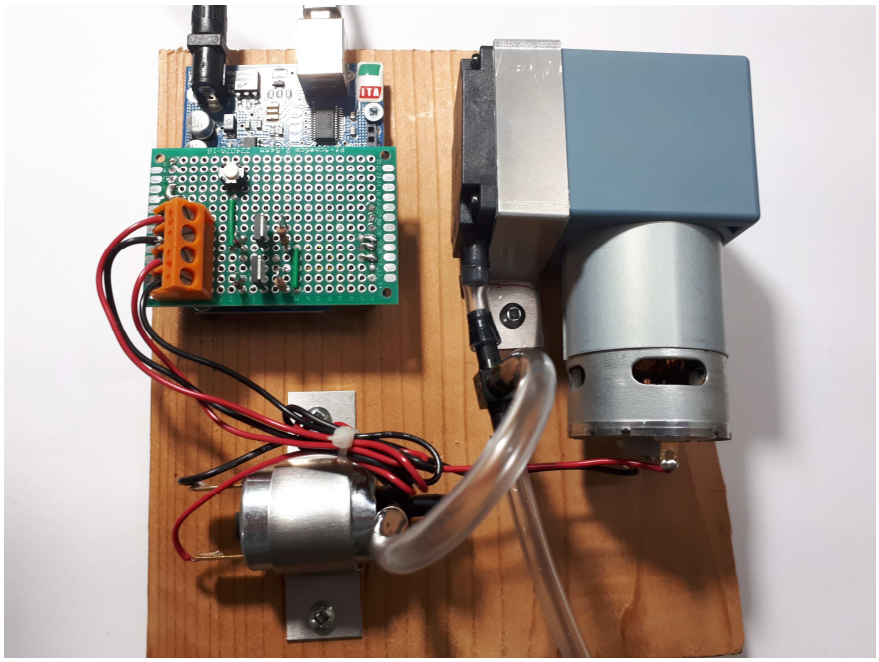
Assembling the Vacuum Pump Subsystem

Below is a diagram of the vacuum pump subsystem followed by some pictures:





I highly recommend attaching the parts from the picture above to some sort of base, as the vacuum pump vibrates quite a bit when it is on. I used a scrap piece of wood as shown below.



To test the vacuum, apply 12V power to the Arduino board and press the push button. The vacuum pump should start running. When the push button is released the solenoid valve should activate briefly in order to equalize the pressure at the tip of the pick up tool with the atmosphere releasing the picked part. Without the valve it may take a long time for the part to be released, resulting in placement errors.

Preparing the Pick-Up Tool

To make the pick-up tool, these are the materials used:

1) Silhouette Pen Holder, available both at Michael's and Amazon.

2) One 0.5 mm mechanical pencil with non-retractable tip. The ones with the plastic tips seem to work better than the ones with the metal tip. I think this is because they are wider and provide more friction when parts are picked, preventing the parts from moving around. I got my pencil at the local dollar store.

3) One 'T' for drip irrigation. An 'I' should work better, but I didn't have any.

4) Hot glue gun.

5) A piece of heat-shrink that fits both the 'T' and the mechanical pencil.

6) Lighter and fine grid sand paper.

Here is a picture of all the parts I used:

Pick-up Tool Assembly Procedure

To make the pick-up tool we need to take apart the mechanical pencil. We just need the tip attached to the barrel, so we should get rid of all the mechanical parts. This is step 1:

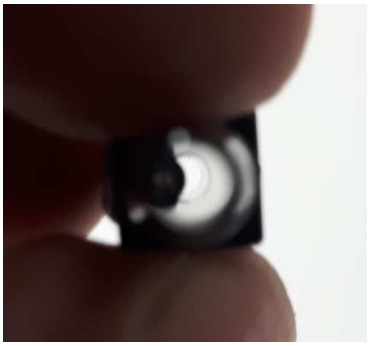
We follow with step 2 that requires a bit of brute force to remove the internal mechanism at the tip of the mechanical pencil:

We just need the tip and the shaft. The shaft is a bit long so cut it so that the length of tip plus shaft is around 75 mm:

Now cut the ‘T’ as show in the picture below.

Seal the side hole with hot glue.

Make sure that the glue is not blocking the inner air path.

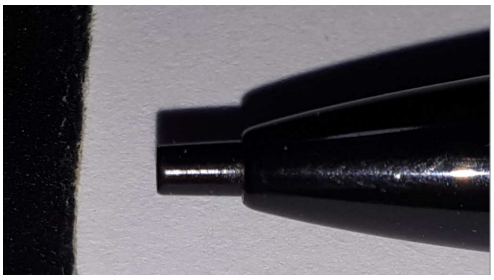


Add hot glue all around one end of the ‘I’ and insert it in the barrel of the mechanical pencil. Wait for it to cool down.

Finally, cover with the heat shrink and heat it carefully using a lighter so it shrinks tightly all around. I tried with a heat gun as well, but the heat gun tends to melt the barrel of the pencil. The heat from the lighter seems to be more 'focused'.

The tip of the pencil needs to be as flat as possible to improve suction of SMT parts. Use fine grid sand paper or a fine file to go from this:

To this:



(Light pressure against the sand paper seems to work well. If you rotate the pencil while sanding and use a circular motion, you should get a fairly flat and uniform surface.)

To complete the pick up tool, we use the Silhouette pen holder by following the instructions in the package. There are three collets in the package for three different size pens/pencils. The mechanical pencil I used requires the smallest one. The picture below shows the pick-up tool attached to the tubing from the vacuum subsystem ready to be used with the silhouette Curio.

As mentioned before, a mechanical pencil with a fixed metal tip can be also used. These pencils don't work well for some parts like ceramic capacitors and ICs as they don't provide a strong grip of the picked part. Here is a picture of the metal tip pick up tool I tested in case you want to try also.

The assembly instructions for the metal tip mechanical pencil are the same as for the one with the plastic tip. The mechanical pencil I used is the "Pentel Twist-Erase 0.5 mm":

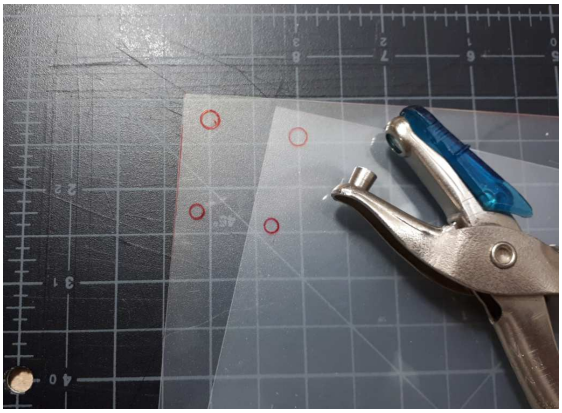
Making Pick and Place Mats

The cutting mat that comes with the Silhouette Curio is not the best option to setup pick and place jobs. The glue is not very sticky and the mat can be easily cut through if errors are made. Also, it is quite expensive and there is only one included with a new Silhouette Curio. Fortunately it is very easy to make pick and place mats for the Silhouette Curio. All we need is an inexpensive chopping mat from the dollar store and the original Curio cutting mat as a template. The mat I started with is the one in the picture below. It was around 1.50$ at my local dollar store.

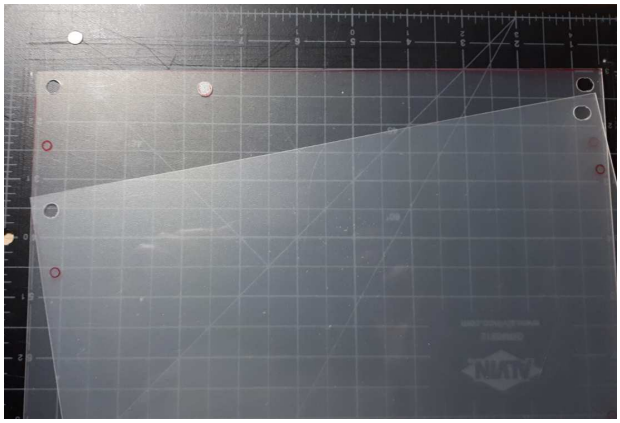
Use the cutting mat provided with the Silhouette Curio as a template to trace the required cuts into the chopping mat. There was enough space to make two pick and place mats out of one chopping mat:

The chopping mat I got was easy to cut with an X-acto knife and a ruler:

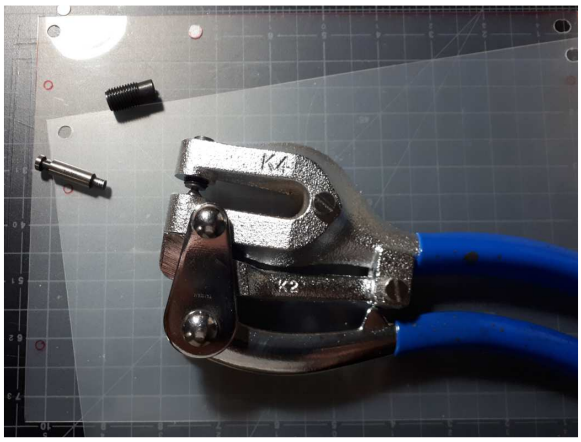
We also need to make the holes. Once again use the cutting mat that comes with the Curio to trace the holes:



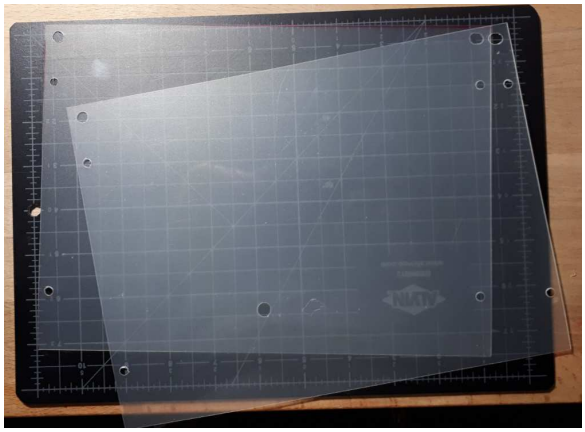
The big hole is the size of an standard paper hole puncher, so that is what I used to make the holes. Notice that the top right hole is not circular; a good approximation to the desired shape can be achieved by using the hole puncher twice.



The smaller holes are 3/16 of an inch diameter. They are not as crucial for accuracy as the big holes, so it would be fine to cut them using the X-acto knife. I used my sheet metal hole punch instead:



And this is the result, two pick and place job mats:

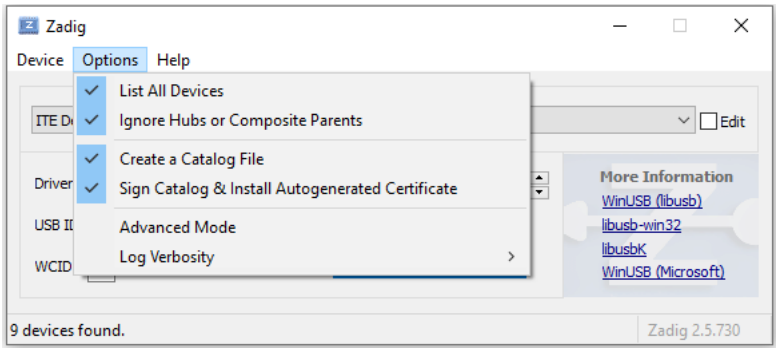


It is very important that these mats fit perfectly into the Curio tray as any looseness will result in pick and place errors.

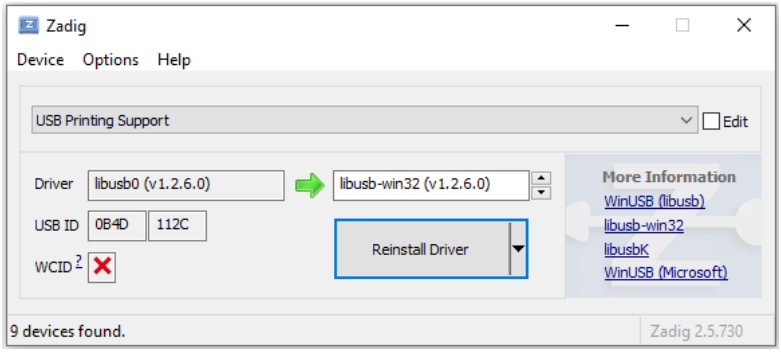
Installing the 'libusb' Driver on Windows

In order to communicate the Python program with the Silhouette Curio, a supported device driver must be installed. Power up the Silhouette Curio and connect it with the USB cable to the computer.

Download the newest Zadig USB driver installer from https://zadig.akeo.ie/. Run the program. Go to menu 'Options' and select 'List All Devices':



Look for USB Printing Support in the dropdown list. Ensure USB ID is: 0B4D (Graftek America) and the DEVICE ID is: 112C (Curio):



Select the driver "libusb-win32 (v1.2.6.0)" which installs a libusb0-Port for Windows. Click replace driver. (In my computer it shows 'Reinstall Driver' because I installed the driver already as you can see in the 'Driver' box to the left)

Installing Python

In order to run the Pick & Place program, Python 3.x is required with the packages 'pyusb', 'pillow', and 'pyserial'. Other packages may be needed as well if they don't come pre-installed with your Python 3 distribution. Optionally you can download a

Python distribution which comes with all the required packages installed. The one I used is WinPython. It can be downloaded from:

https://winpython.github.io/

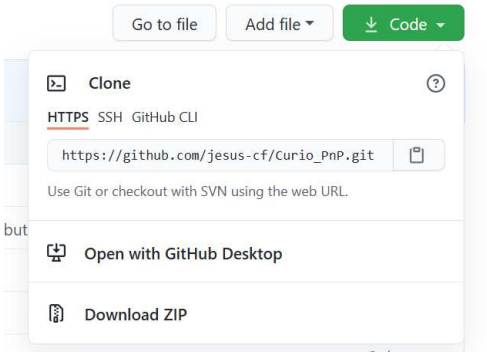
The distribution used at the time of writing this guide was "*WinPython64-3.8.1.0cod*". If you install WinPython, remember to run the program "*WinPython Control Panel.exe*" as an administrator, click the 'Advanced' tab, and then "register distribution" so files with the '*.py*' and '*.pyw*' extensions are associated with the installed distribution of Python.

Download the Pick and Place Program

To setup pick and place jobs, create job mats, and proceed with pick and place, we use the program "PnP_Curio". This pick and place editor program can be downloaded from Github:

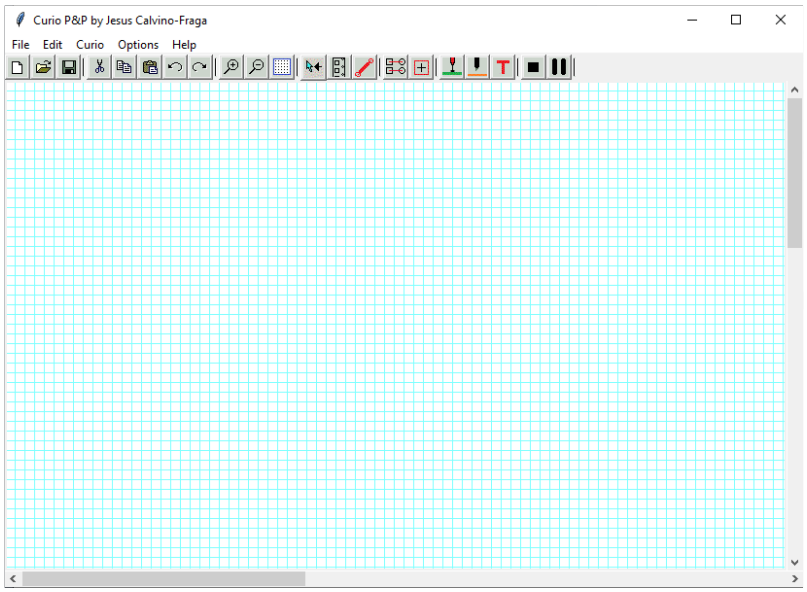
https://github.com/jesus-cf/Curio_PnP

You can use Git to copy the code to your computer or download a compressed file:



Running the Pick and Place Program

Double click the file ‘PNP_Curio.py’. A window similar to the one below shows up (the colors may be different, but they can be configured in the ‘Options’ menu):



Setting a Pick and Place Job

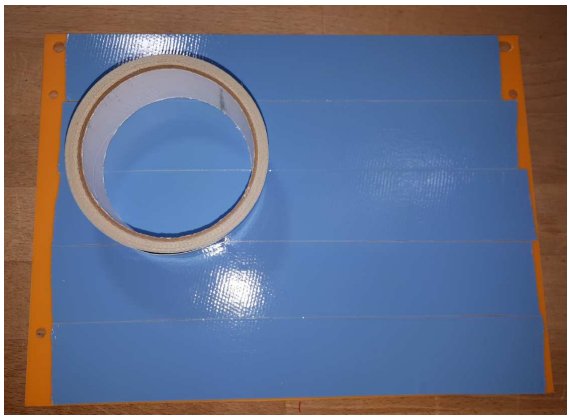
Making a Craft Foam Job Gig

To achieve acceptable accuracy for the pick and place job, both the printed circuit boards (PCBs) and surface mount devices (SMDs) carrier tape must be placed accurately on the tray Silhouette Curio. This can be achieved by using a foam gig. To make a foam gig we need double sided duck tape, regular double sided tape, 2 mm craft foam, and optionally a Sharpie marker to write the name of the components in the foam.

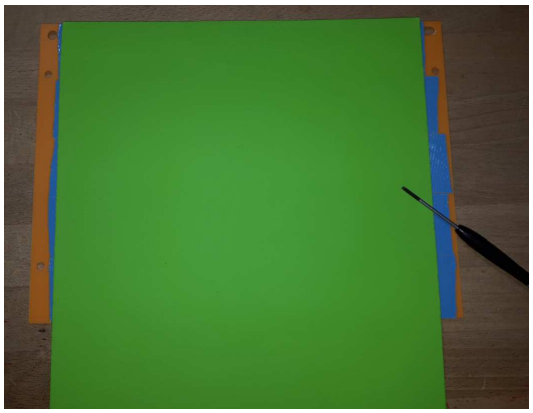
We start with the double sided duck tape and the job mat we made earlier using a chopping mat:

This is the double sided Duck tape I used. Other double sided tapes may also work. I bought the double sided Duck tape at my local hardware store, but it is available from Amazon as well.

Cover the job mat with the double sided duck tape. Make sure the Duck tape gets near the edges. The excess is removed in the next steps.

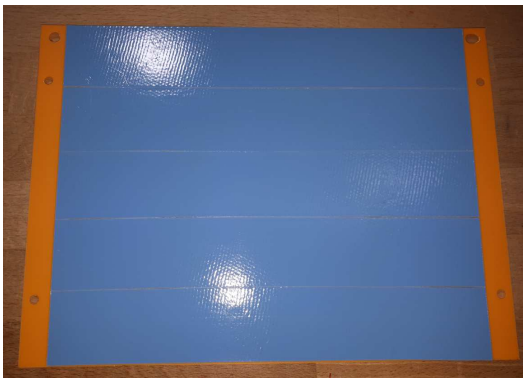


Place the 2 mm craft foam in the middle of the mat. Using a small screw driver or other pointy object, lightly score the surface of the Duck tape using the sides of the craft foam as a guide:



Use a sharp knife and a ruler to cut the excess Duck tape from the edges:

Remove the excess Duck tape from the sides:

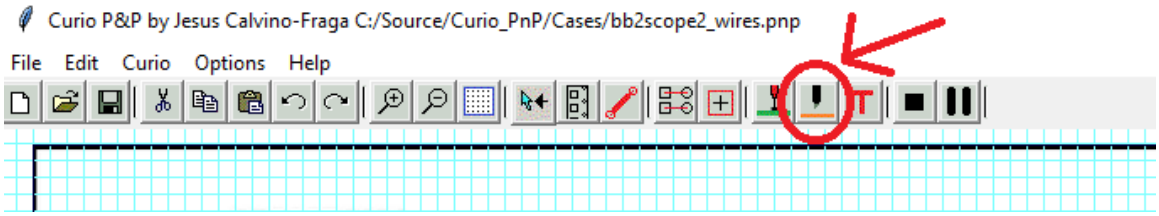


For the next step we need the 2 x 2mm stacking spacers that come with the Curio, if the stacking spacers are not used, the cut blade will not be able to reach the job mat.

Stack the 2 spacers and the job mat and place them on the Curio base:

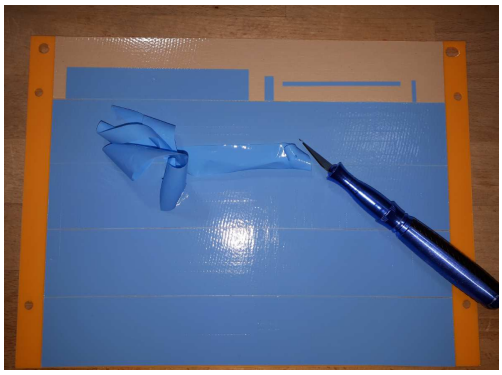
Set the standard blade that comes with the Curio to a depth of ‘1’. The blade depth set to ‘1’ should be enough to cut the blue backing of the double sided Duck tape:

Place the blade in the Curio, secure it, and press the ‘cut gig’ button in the PNP_Curio program.

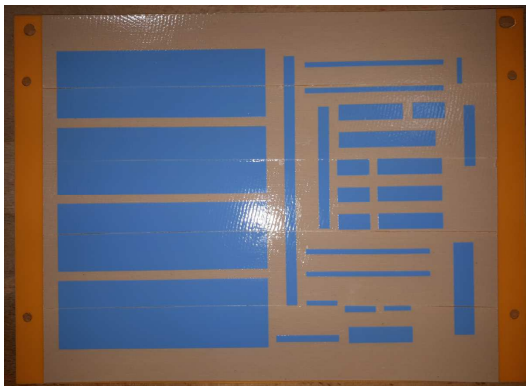


It is hard to see in the picture below, but a rectangle for each PCB and carrier tape has been cut in the blue backing of the Duck tape:

Weed the excess backing off the Duck tape, exposing the sticky part. A sharp knife or tweezers are useful when completing this step:



There should be Duck tape backing remaining only where the foam cutouts are going to be. If we don't leave this Duck tape backings, the foam cutouts will be extremely hard to remove.



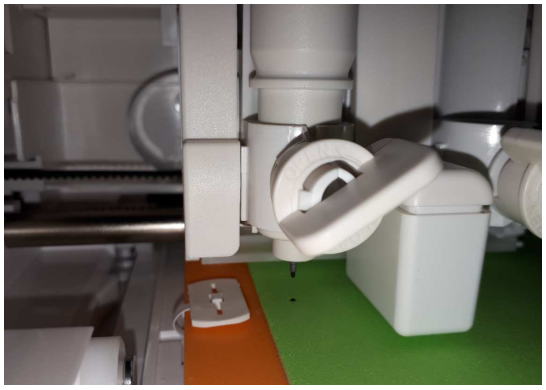
Place the craft foam. It should stick pretty well to the Duck tape:

Cut the excess foam from the top and bottom:

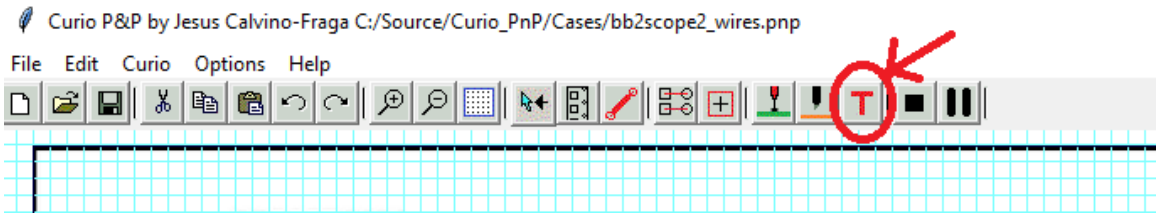
This step is optional, but it looks very good when done with the Curio. The names of the pieces of cut carrier tape can be written on the craft foam. This turns out to be very useful for reloading the foam gig if we are going to use it for several pick and place jobs. We need a Sharpie marker and a pen holder:

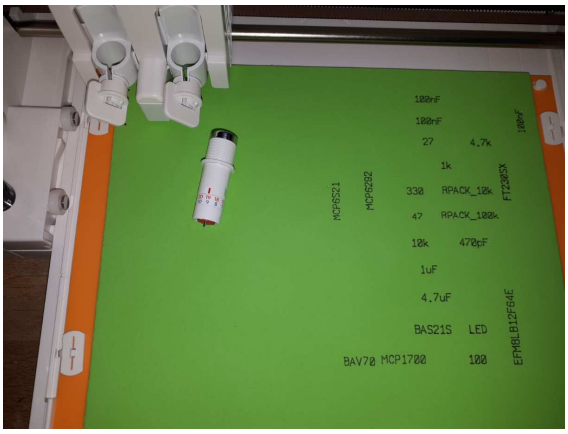
Put the Sharpie in the pen holder following the instructions in the pen holder box. Now place the pen holder in the Curio but don't secure it yet. Lower the tool holder all the way down and leave the Sharpie tip to gently rest on the craft foam. At this point, secure the pen holder to the tool holder. We need to do this otherwise the Sharpie will dig into the foam damaging both the foam and the Sharpie tip:

After we release the tool holder, the Sharpie goes up about 5 mm:



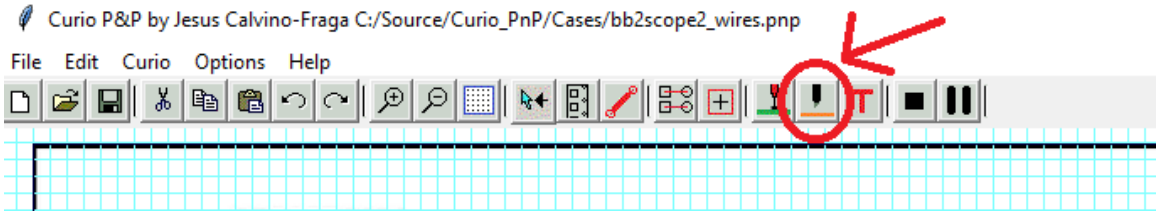
From the PNP_Curio program send the text to the Silhouette Curio. Any visible text is sent to the Curio and written in the foam with the Sharpie marker.

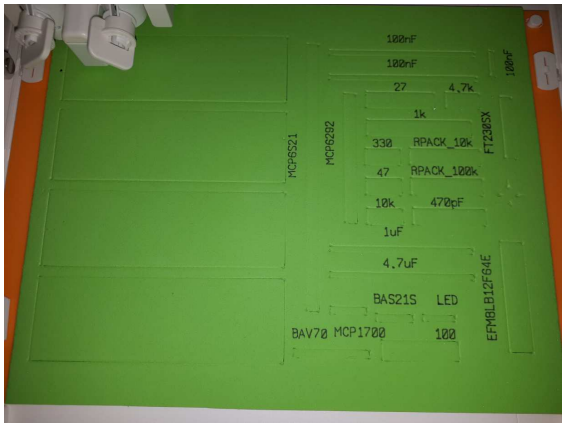




Now we need to cut the placeholder holes for the PCBs and SMD's carrier tape. We use the 'Deep Cut Blade' set at 19. The job mat is stacked on the Curio base with two 2-mm spacers as before:

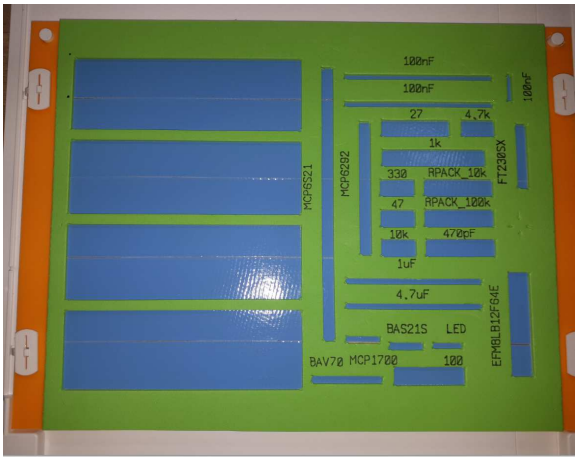
From the PNP_Curio program, click the 'cut gig' button. The place holder holes for the PCBs and carrier tape are cut into the foam.





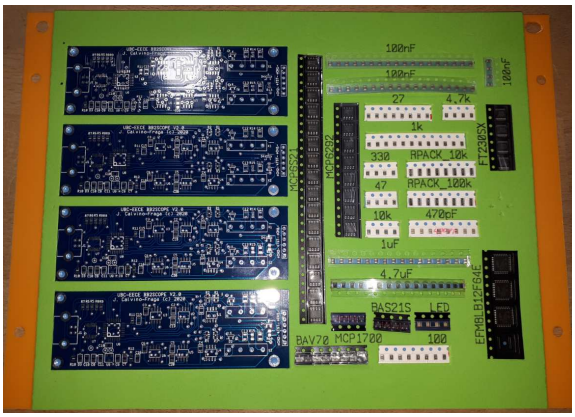
Normally one pass is enough to cut all the place holder holes. Try to remove one of the cut-outs. If it doesn't come out easily, you can try sending the cuts again to the Curio. There is always a little bit of uncut foam at the corners, but the cut-outs break easily if you pull them from the middle using tweezers as shown in the picture.

After removing all the cut-outs, this is how the pick and place gig looks like.

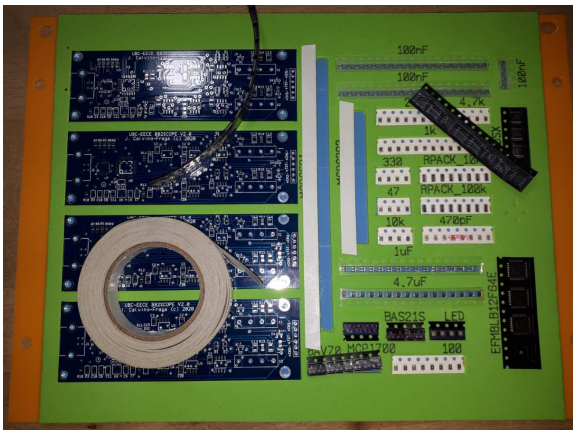


The cut-outs should all be cleanly removed as shown below.

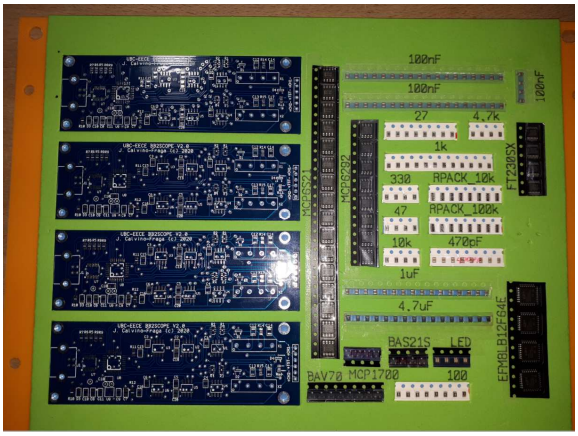
Every time I make a new gig I like to test that all the PCBs and carrier tape fit properly with the correct orientation before adding the double sided tape to hold the plastic carrier tape:



To hold in place plastic carrier tape, apply standard double-sided sticky tape to where the sprocket holes rest in the craft foam. I used 7 mm wide double side sticky tape from the dollar store as shown in the picture below. Neither the paper carrier tape nor PCBs require double sided tape to remain in place.



The final result looks like this. The foam gig is ready for a pick and place job.



Pick and Place

asdf