4th Year Project: Soldering Machine

James Glanville

11th June 2013

1 Existing Solutions

There are a number of hobbyist solutions to deal with SMD parts:

- Solder paste + heat:
  - Solder stencils: Expensive setup costs, very quick to use. Not suitable for this project due to cost.
  - Manual solder paste application: Mostly expensive dispenser, or with a syringe. Requires high accuracy while placing paste and is time consuming.
  - Hot air gun: Can be relatively difficult to get the right temperature profile.
  - Converted toaster over: seems quite common.
- Soldering by hand:
  - Dragging a bead of solder along pins, then cleaning up with solder sucker/wick. This requires a lot of feedback, which seems difficult to automate.
  - Some devices (SOIC?) have slightly bendy pins. A small amount of downwards pressure on the pin onto a tinned pad can work, but this method will not work for leadless packages.

2 Project Idea

I have decided to approach the project by building a machine which can place solder paste accurately, and then place the surface mount devices on top. It should achieve this with minimal human interaction. The machine should also be able to drill holes for through-hole devices, as well as be equiped with a milling bit for isolation routing through copper-clad pcbs.

When finished, the machine should be able to be connected to a computer via a USB port. A custom GUI will load the necessary design files (for example the holes to be drilled, pads to be soldered), generate the necessary GCODE, and send it to the printer.

3 Requirements

3.1 X/Y axes

A device pitch of 0.4mm (distance from centre of one pin to the next) is on the lower end of SMD ICs. Using solder paste, devices will shift slightly upon heating to self-align. A requirement is that the device should be placed to a small fraction of its pitch, so there is no danger of incorrect connections. I have decided to aim for an accuracy of +-0.05mm in placement. This should be sufficient, but may be difficult to achieve in practice.

3.2 Board size

A maximum board size of 100x100mm would be sufficient for the vast majority of boards. I have chosen 100x100mm because that is the maximum board size using the free version of EAGLE, a popular design package. As a result, there are a lot of designs that use the maximum size, and it would be useful if this project allowed those designs to be populated.

4 Design

4.1 X/Y axes

The X/Y axes have the same requirements: +-0.05mm accuracy. There is the choice between moving the various tools over a stationary bed, or keeping the tools stationary and moving the bed. I have decided to move only the bed, because it means that the pick and place functionality does not have to move the parts, simply lift and lower them. This should reduce the risk of the parts shifting on the needle, or falling off. It also allows for the use of more tools since weight is no longer a concern.

The bed will need travel distances of at least 100mm in both directions so that the entire bed can be accessed by tools. However, there should also be space to place devices before they are automatically placed. This should need no more than 20-30mm in one axis direction only.

Possible driving mechanisms:

- Toothed belts + pulleys + stepper motors: Simple, as used in reprap 3d printers. Can be run open loop very easily. The toothed belts stretch under load, and for the milling/drilling functionality are likely to deflect too much under loading. Requires: stepper motor + driver, belt, pulley.
- Threaded rod + stepper motors: Cheap, slower but probably fast enough. M3-5 easy to couple to motor shafts (http://www.thingiverse.com/thing:9622).
- Closed loop: dc motors + feedback. Linear potentiometers: relatively expensive, and potential issues with electrical noise. Rotary encoders: cheap, accurate (m4 pitch is 0.5mm, not much travel/turn, might only need 10 slot encoder)

I have decided to utilise M6 threaded rod coupled to small stepper motors.

Linear slides:

- Drawer slides: cheap, surprisingly high precision.
- 3d printed bushings + steel rod. If 3d printer existance assumed, then very cheap solution.
- LM8UUs (or smaller): About 50p each, relatively simple to use.

I have decided to use LM8UU bearings with steel rod.

Motor selection:

Having chosen to use small stepper motors to control the axes, the choice of stepper motor is now important. I had some NEMA17 motors, which claim a 2.2Kg/cm holding torque. I decided to run some tests to see what the maximum speed I could run them at without skipping steps, and found a result of 8mm/s. This is sufficient for use in the project (where speed is not a great concern), but I felt it would not be beneficial to use a smaller motor.

4.2 Z axis

The Z axis will not require as much precision as the X and Y. Potential mechanisms:

- Micro servos: cheap (2 on ebay). Require no drivers, and simple to drive. Z axis potentially does not need to be linear (UP/DOWN only?) so rotary->linear mechanism simpler. If not linear, then hinging out of the way is a cheaper mechanism than slides.

4.3 Part placement

- Manual placement. Much more of a problem for a large number of SMD resistors/capacitors etc, than larger components.
- Vacuum "tweezers". Mechanically relatively simple, but problems of intelligent control, as well as well as storing the components before placement. As an experiment, I tried using an aquarium pump in reverse (by attaching the silicone tubing to the inlet instead of the outlet) connected to a fairly large diameter needle (ID: Xmm).

4.4 Heated Bed

Soldering to a hot board is easier (smaller temperature difference). This is now irrelevant as I shall not be soldering pin-by-pin.

4.5 Flux application

Solder paste includes flux, so this does not need an extra step. This is a useful consequence of using solder paste as opposed to solder, where it would involve an additional placement step

4.6 Drilling and isolation routing

Isolation routing is a method to mill traces from copper-clad pcb. A conical engraving bit is used to cut very narrow tracks through the copper, to leave disconnected pads as required.

Drilling holes in pcbs is a time-consuming process to do by hand. If this machine could drill the holes automatically, it would reduce the total time taken significantly. A small brushless motor with a collet would be enough. The biggest challenge will be mechanically coupling the motor to the drill bit. This is not intrinsically difficult, but it should be made in a way that is accessable to people without the use of a lathe (?)

A 25 Amp ESC (Turnigy plush 25A, 9) powering a small brushless motor (Turnigy 2217 20turn 860kv 22A outrunner, 10) is capable of milling steel easily, so smaller, cheaper motors should be fine for routing/drilling copper.

The imodela cnc appears to use a small ball bearing for the shaft, with custom machined shafts. The milling bits fit inside the shaft, and are secured with a single grub screw. This seems to work, though it would need a very carefully machined shaft so that axial misalignment did not shear the end of the milling bit.

An alternative design would be to use a low cost rotary tool, which already has the bearings, collet and spindle motor.

4.7 Soldering iron

Designing a mount for a cheap and widely available soldering iron may well have been the simplest approach. However, using solder paste, this part of the design is not needed.

4.8 Melting solder paste

Once the board has been pasted and the parts deposited, the solder paste needs to be heated up so that it will melt and form good connections. There are a few possibilities for this process:

- Oven: a small over can be used to heat the entire board (see: http://www.openhardware.net/Misc_Stuff/T This appears to work well, however the oven should not be used for food after that, since the components of solder paste are toxic. As a result, it can be an expensive method since it requires dedicated, expensive hardware.
- Hand-held hot air gun: A small hot air gun can be used to gradually melt the solder for the components. There is a requirement for the hot air gun that it emits sufficiently hot air at a sufficiently low velocity that the solder will melt, but also that the devices will not shift under the force. I use a hot air attachment with a butane-powered soldering iron which works nicely. This does require the most human interaction (since it is less likely that the flow of hot air will cover the board uniformly, but it is not a particularly precise process.

- Hot-plate. A small hot-plate can be used to heat the board from underneath (see http://www.hobbytronics smd-soldering). An advantage compared to the oven method is that a piece of thermally conductive metal can be used between a hotplate and the board, so that the hot plate will not become food-unsafe.

The choice of method to heat the solder paste may well depend on the end-user - one solution is not clearly superior to the others. During the course of this project I am likely to use the hand-held hot air gun since I possess the equipment.

4.9 Electronics/Firmware

I have decided to use an avr chip to translate GCODE to stepper motor commands for the following reasons:

- Cost: an atmega328 is 2-3, a usb-serial chip is ¡2, which is not much.
- Mature open source firmware (GRBL, https://github.com/grbl/grbl) is available for gcode->stepper conversion, which leaves more time to focus on other parts of this project.

On the host pc, custom software will have to be written to translate the design into a gcode file to be sent to the machine. Usefully however, the machine's firmware will not need to be written or modified.

5 Host computer software

Software will have to be written to translate board cad data to gcode. Some of this already exists in the public domain, some will have to be made as part of this project.

Isolation milling and drilling: pcb2gcode (https://github.com/festlv/pcb2gcode-metric). Takes gerber designs and generates isolation routing/drilling gcode. It is also capable of voronai isolation (functionally equivalent routing that cuts the least amount of material). The only configuration will be to generate a millproject file with the correct cutting depths.

Solder paste placement: No freely available software (as far as I am aware).

Device placement: No freely available software (as far as I am aware).

The solder paste placement should be relatively straightforward. The program will simply need to find the locations of the pads, calculate their area, and then apply a suitable amount of paste to them. For ICs, the paste will not need to be separately applied to each pin (due to the high surface tension of molten solder), so some method of detecting ICs on the board layout will be needed.

The code to place devices will be the most challenging. However, provided there is a sensible area in which to hand-deposit the devices, it should not be too hard to calculate how to transport them to the required location.

In my opinion, the most challenging part of writing the conversion code will simply be to extract the necessary data from the cad files. Once this has been obtained, the gcode should be easy to create.

6 Tools

I have a small mill, a 3d printer and a lathe. It may be worth considering that a lot of people/schools have access to a laser cutter and perhaps mill/lathe, so if possible all custom components could be 3d printed or lasercut.

7 Solder paste extrusion

I have decided to use the design here: http://www.thingiverse.com/thing:20733

The extruder works by applying pressure to the top of the solder paste syringe. Pressure is applied by tightening a pulley made of T5 timing belt with a geared stepper motor. I chose this design because the extruder is compact, and the design allows for fast reduction of pressure, important for clean extrusion and to avoid the solder paste dribbling.

To calibrate the extruder, two parameters are needed: steps/volume extruded and retraction steps. The first is easy to calculate:

(All dimensions in mm,mm^2,mm^3)

$$\begin{aligned}
Volumeextrudedperstep &= (cross - sectionalareaofsyringe * T5pitch * numberofteethofT5pulley) && (1)\\
&/(numberofmicrosteps/step * steps/rotationofmotor * gearreductionratio) && (2)\\
&= (66.1276 * 5 * 10)/(16 * 200 * 12) && (3)\\
&= 0.0861mm^3 && (4)\\
& && (5)
\end{aligned}$$

$$\begin{aligned}
Steps/mm^3 &= 1/volumeextrudedperstep && (6)\\
&= 1/0.0861 && (7)\\
&= 11.61 && (8)\\
& && (9)
\end{aligned}$$

The retraction steps must be experimentally determined. The choice is a tradeoff between minimising solder paste dribble (large number of retraction steps) and reducing total process time (small number of retraction steps). It must be enough that negligible pressure is exerted on the syringe when retracted.

Some method to ”prime” the extruder will be needed. Assuming the extruder is initially hand-tightened so that the belt has little slack, a possible method is to repeatedly advance the extruder and then retract by a slightly smaller amount, until the user notifies the machine that solder paste has been extruded. At this point, the extruder should pause after retracting. The solder paste deposition process can then commence since the exact point of extrusion has been established. Assuming that x is the required granularity of calibration (the maximum amount of solder paste that will be wasted), and y is the retraction distance, the following example gcode will carry out this calibration:

```
//values in braces {} are to be evaluated on the host

//Start loop here:
G92 E{y} //Set extruder position in firmware to be y
G1 E{x+y} //Extrude x amount of paste
G1 E{y} //Retract
//if user has not indicated extrusion then loop again

//when the user has indicated extrusion:
G92 E{y}
```

8 Managing devices prior to placement

There will need to be a way to place the devices in a known position, so they can then be picked up and placed in their final locations. Commercial pick and place machines take the parts directly from the reels they are supplied in. While this is optimal for that application in that it is a very fast way to operate, the cost of reel dispensers is prohibitive for this machine. Instead, the parts will have to be roughly placed by hand. There are a number of ways this could be done:

- Embossed outlines of all necessary device footprints.
- Rectangular embossed outline, devices to be located in one corner.

I have decided that the latter will be easiest to implement. A rectangular hole will be milled into the bed. Each part can then be manually placed inside, then pushed to one corner. A consistent system will be needed to ensure correct orientation. I have decided to use the following set of rules:

- ICs will be placed with pin1 touching the datum corner.
- Devices will always be placed with the longest side (if applicable) touching the long side of the rectangle.
- Devices such as mosfets will be placed with the corner between the longest side and side with most pins touching the datum corner.

- Polar 2-lead devices with always have the anode nearest the datum corner.

Given the current component to be placed, and the device footprint, it will be trivial to find the coordinates of the centre of the part. Some consideration will have to be given as to how to determine the centre of mass of parts which are not symmetrical in two directions (eg mosfets).

9 Vacuum Placing

9.1 Prototype

To test how effectively small parts could be placed with a vacuum, I made a simple test setup:

- 12V diaphragm air pump (8.89 - ebay)
- 6mm OD/4mm ID silicone tube (2.69/m - ebay)
- various needles (need more info here)
- MOSFET
- 330Ω resistor

Figure 1: Pump with hose and needle.

The pump draws 200mA at 12V, which is easily controlled with a MOSFET. Using a 1.8mm ID needle, SOIC-20 parts were picked up, and had significant friction against the needle - essential to avoid slipping or rotating. It was important to place the needle as centrally as possible on the part to avoid generating any imbalance that allowed the part to fall.

9.2 Conclusions

A vacuum needle solution is workable, and a good choice for this project. Things that need to be considered for use in the final project:

- The pump takes a certain time to generate a vacuum because of the relatively large internal volume of the pump and silicon tube. It will be important to measure this, and add a margin of error so that full suction is applied to each device before any movement takes place. Similarly, when turning off the pump, the pressure must return to atmospheric before the device can be considered to have been placed succesfully. In testing, the pump I used took less than a second to leak sufficiently for this to occur, but if another pump were to be used, it may be necessary to place a tiny hole in the air line so that pressure drops quickly.
- It is important to place the needle very near to the centre of mass of the device. The majority of devices will be symmetrical, which will make this easier.
- The needle must be able to be raised and lowered to the correct heights. It may be possible to push the needle down with a weak spring, with a micro servo that raises it - this would remove any need to know the exact height of each device. This will need to be tested.

Figure 2: SMD resistor being picked up.

10 Measuring Accuracy

It is important to measure the accuracy of the axes to ensure that they can reach the specified +-0.05mm. This will need to be measured at all points on the bed (as the accuracy will worsen when the bed is central due to increased rod deflection.). Of particular concern will be the deflection under the load of the isolation routing bit. The solder paste and device placement will not involve any significant force. The drilling will generate a force, but the tolerance required of a 1mm hole is not nearly as demanding as that of the isolation routing cutter. I have not yet measured the exact forces exerted by the routing bit, but I estimate that they are 5-10N. This means that the anti-backlash nut will need to exert the same or greater force to ensure that the board does not move. This in turn places extra resistance on the motor movement.