4th Year Project: Soldering Machine

James Glanville

11th June 2013

Contents

1 Introduction

The aim of this project is to develop ways to assist electronics hobbyists in working with SMD electronic components and circuit boards. SMD (Surface Mount Devices) are electronic components which are designed to be soldered to PCBs (Printed Circuit Boards) without requiring holes to be drilled in the board. Compared to through-hole devices, SMD parts are usually smaller and cheaper and as a result are becoming ubiquitous in many applications. Even when hobbyist users do not require the size or cost advantages, many modern devices are only available in SMD packages. Using SMD parts require smaller board features (A manufacturing challenge) as well as more advanced soldering techniques. For this project, a machine is envisioned which solves some or all of the challenges involved to ease the process of SMD board development.

2 Existing Solutions

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

2.1 Hand soldering

Despite the increased complexity, many SMD parts can be soldered by hand. There are two main methodologies for this process:

- The pins are soldered individually. This is mainly suitable for larger SMD parts. A steady hand is required, as well as good eyesight. Binocular microscopes are often used to assist this process.
- The process of "drag soldering" is used. The device is placed on the board, and a small amount of solder is placed on the tip of a soldering iron. The iron is slowly "dragged" along the pins. If done correctly, surface tension effects cause the majority of the solder to remain on the tip of the iron, while the necessary electrical connections are soldered.

These methods are imprecise, and will often neccesitate the use of solder wick or a solder sucker to remove solder bridges. More importantly, they are wholly unsuitable for use with "leadless" packages. Instead of metal pins, these devices only have metal pads on their bases. Without exposed metal when the device is placed, it is not possible to solder the connections by hand.

2.2 Solder paste

Solder paste consists of powdered metal solder suspended in flux. It is used commercially to manufacture boards with SMD compenents. The solder paste is applied to the board where connections are to be made. The devices are then placed on top (the paste has slight adhesive properties which limits undesirable component movement). Finally the paste is heated either directly or indirectly which causes it to melt and form solder joints. The paste is placed in a number of ways:

- Solder stencils: A thin stencil is manufactured from plastic or metal. It has holes which correspond to locations on the board that require solder. Solder paste is then roughly applied to the top of the stencil before being scraped off. This forces controlled volumes of paste into the holes. This a very quick process, and suitable for production quantities, but has the expensive setup costs of stencil manufacture.
- Solder paste can be applied by hand using a syringe. However, this requires high accuracy while placing paste and is time consuming.
- Automated solder paste extrusion uses a CNC (Computer Numerical Control) platform to manoeuvre a solder paste extruder around the board. Small amounts of paste are extruded where necessary. This is not the fastest method of manufacture, but is suitable when only small numbers of boards are required.

Once the solder paste has been placed and the devices placed on top, it must be heated to form solder connections. This can be achieved in several ways:

- A hot air gun can be used manually to locally heat the required areas of the board. Hot air guns are inexpensive, but the process requires care so as not to overheat any components. It can also be relatively difficult to get the correct temperature profile - some solder joints may be overheated while some may not fully form.
- Entire board heating involves placing the board in an oven which is heated to the correct temperature. This can vary from large commercial equipment to the use of a converted toaster oven.

3 Project Idea

I have decided to approach the project by building a machine which can extrude 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.

4 Requirements

4.1 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.

4.2 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.

5 Design

5.1 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.

5.2 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.

5.3 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).

5.4 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.

5.5 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

5.6 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.

5.7 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.

5.8 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 oven 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.

5.9 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.

6 Improved designs

6.1 Solder paste extruder

The solder paste extruder design that was initially used was unnecessarily bulky and complex. This made it both harder to fit to the tool mounting mechanism, and increased the distance between the mount and the tool, amplifying any wobble in the mount. It was therefore deemed necessary to redesign it, optimising for size. Operation and performance is similar between all designs however.

6.1.1 Paste extruder using bevel gears

The first attempt to reduce the size of the extruder used bevel gears to shrink the size of the gearbox.

6.1.2 Paste extruder using worm gears

I realised that using worm gears would give the same gear ratio in a smaller gearbox. The improved design is as small as possible and performs as well as the much larger extruder I started with.

7 Serial interface

For simplicity, the communication between the PC and the electronics board is implemented as a serial link. An arbitrary baud rate of 57600 has been chosen, this is sufficiently fast so that the electronics board is not slowed by waiting for instructions (as the firmware buffers incoming commands in a small buffer in RAM), and sufficiently slow that the error rate is assumed to be zero.

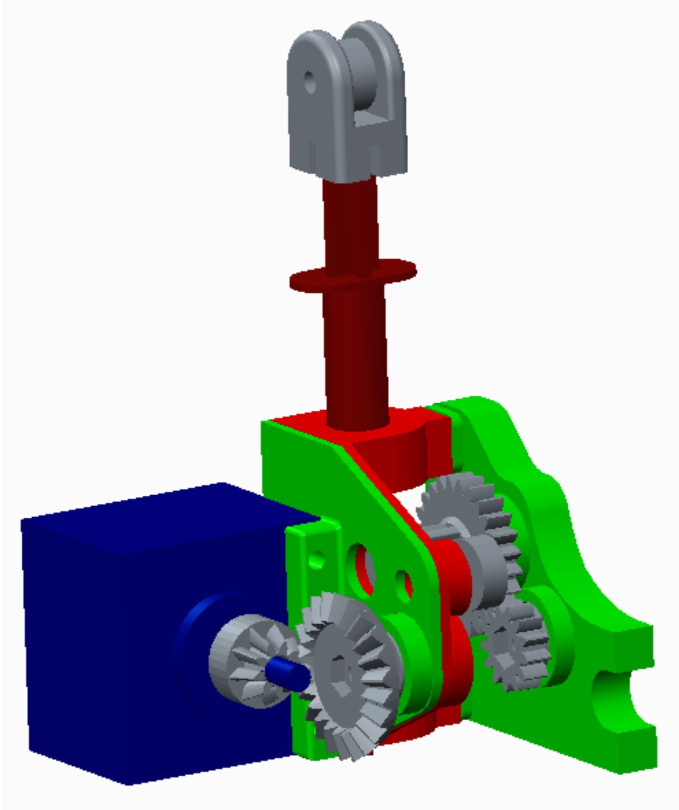


Figure 1: Placeholder image of bevel extruder

7.1 USB-serial cable

A usb-serial cable based on the FTDI ft232r chipset can be plugged into the serial header on the electronics board. This is a cheap (¡5) device that is a complete solution. Driver support for the FT232R is available for all platforms that have been considered (Windows/OSX/Linux/Android).

7.2 Bluetooth serial link

As an alternative to a USB-serial cable, a bluetooth link can be used. Depending on the end user, this may be more convenient. The HC-05 bluetooth module was chosen because of its low cost (3.50) and simplicity. The board requires a 3.3V supply, which is not present on the electronics board. I made a small breakout board for the module, including an AMS1117-3.3 3.3V linear regulator (0.10) and link status LEDs. The 3.3V TX output from the module is sufficient for the AVR despite its higher 5V supply. To avoid damage to the bluetooth module, a potential divider consisting of two 1k resistors is used. This 2.5V output is again sufficient for the bluetooth module.

To configure the baud rate for the bluetooth module, it was connected to a pc using a usb-serial cable. The bluetooth module is powered up whilst the "KEY" pin is held high (3.3V). This starts the module in AT configuration mode. The baud rate was then set by sending the

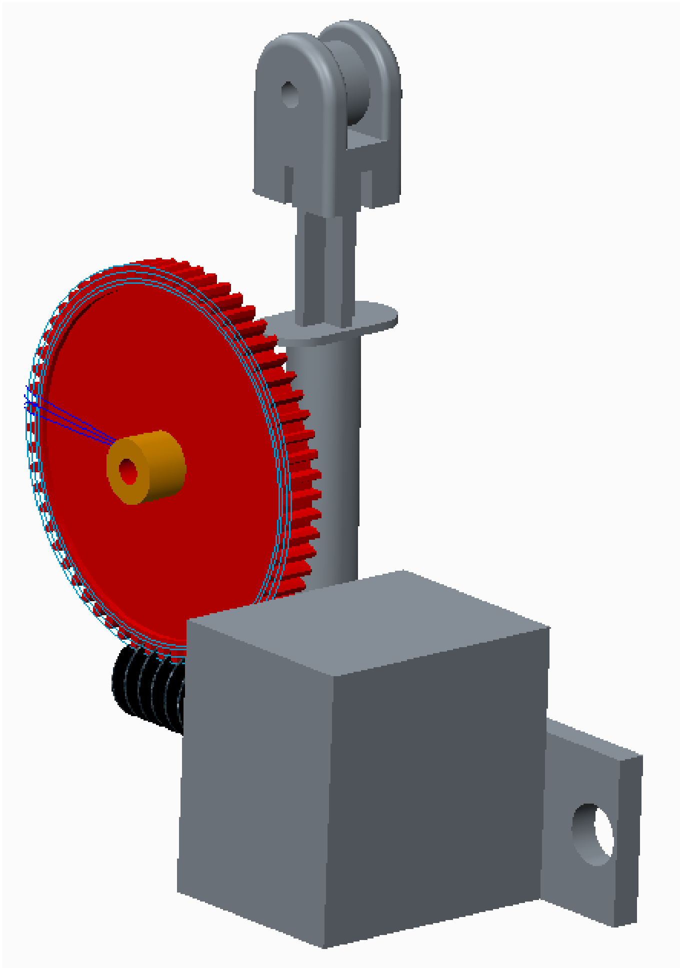


Figure 2: Placeholder image of worm extruder

string "AT+UART=57600,1,0" at 38400 baud. This configuration is only required once, the module stores all configuration in non-volatile storage.

8 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).

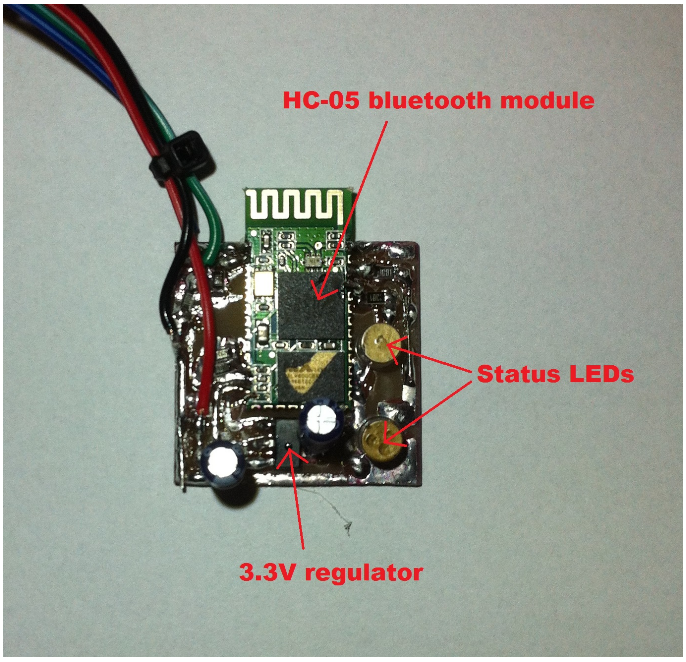


Figure 3: Bluetooth module mounted on breakout board.

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.

9 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.

10 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}
```

11 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).

12 Vacuum Placing

12.1 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 4: 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.

12.2 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 5: SMD resistor being picked up.

13 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.

14 Hot air tool

Ideally the heating of the solder paste should be carried out automatically by the machine. This section describes a prototype hot air tool.

A hot air tool used for soldering has the following requirements:

- Ability to heat solder paste up to 240 °C
- Slow, steady airflow
- Controllable temperature (ie not open-loop)

Calculations:

Assumptions: Specific heat capacity of dry air is not significantly affected by temperature for estimates of power required. 50% of energy into heater is used to heat hot air stream (other radiative losses)

Flow rate of air pump = 18 cm^3 / s Specific heat capacity of dry air = 1.006 J/gK Delta T = 240 - 20 = 220K Density of air = 0.001225 g / cm^3 Power required to heat air (ideal) = (18*0.001225)*220*1.006 = 4.88W Power required to heat air (assuming losses) = 4.88*2 = 9.76W

For simplicity, it is assumed that the heater requires 10W of input power.

The design is based on a small coil of nichrome resistance wire as a heating element. Calculations must be made to find the size of wire reqired:

Size of coil (Chosen to be similar in diameter to the air tube with 6mm OD/4mm ID)

Diameter = 5mm Length = 12mm Wire spacing = 2mm Angle of wire = tan^{-1} (2/(5* pi)) = 7.26 ^{0} Turns of wire = 12/2 + 1 = 7 Length of wire = 7 * sqrt ((5 * pi $)^2$ * 2^2) = 110.84mm

Resistivity of nichrome at room temperature = 1.0 * 10E-6 ohm m Power supply voltage = 12V Current draw = 10 / 12 = 0.83 A Resistance required = 12^2 / 10 = 14.4 ohms Cross sectional area of wire = 1.0 * 10^{-6} * 110.84 * 10^{-3} / 14.4 = 7.70 * 10^{-9} m^2 = 7.70 * 10^{-3} mm^2 Diameter of wire = sqrt(7.70E-3/pi) * 2 = 0.099mm

This is the minimum diameter of wire. Next the maximum diameter of wire is calculated, assuming a maximum current draw of 10A.

Resistance required = 12 / 10 = 1.2 ohms Cross sectional area of wire = 7.70 * 10^{-3} * (14.4/1.2) = 0.092 mm ^{2} Diameter of wire = sqrt(0.092/pi) * 2 = 0.34mm

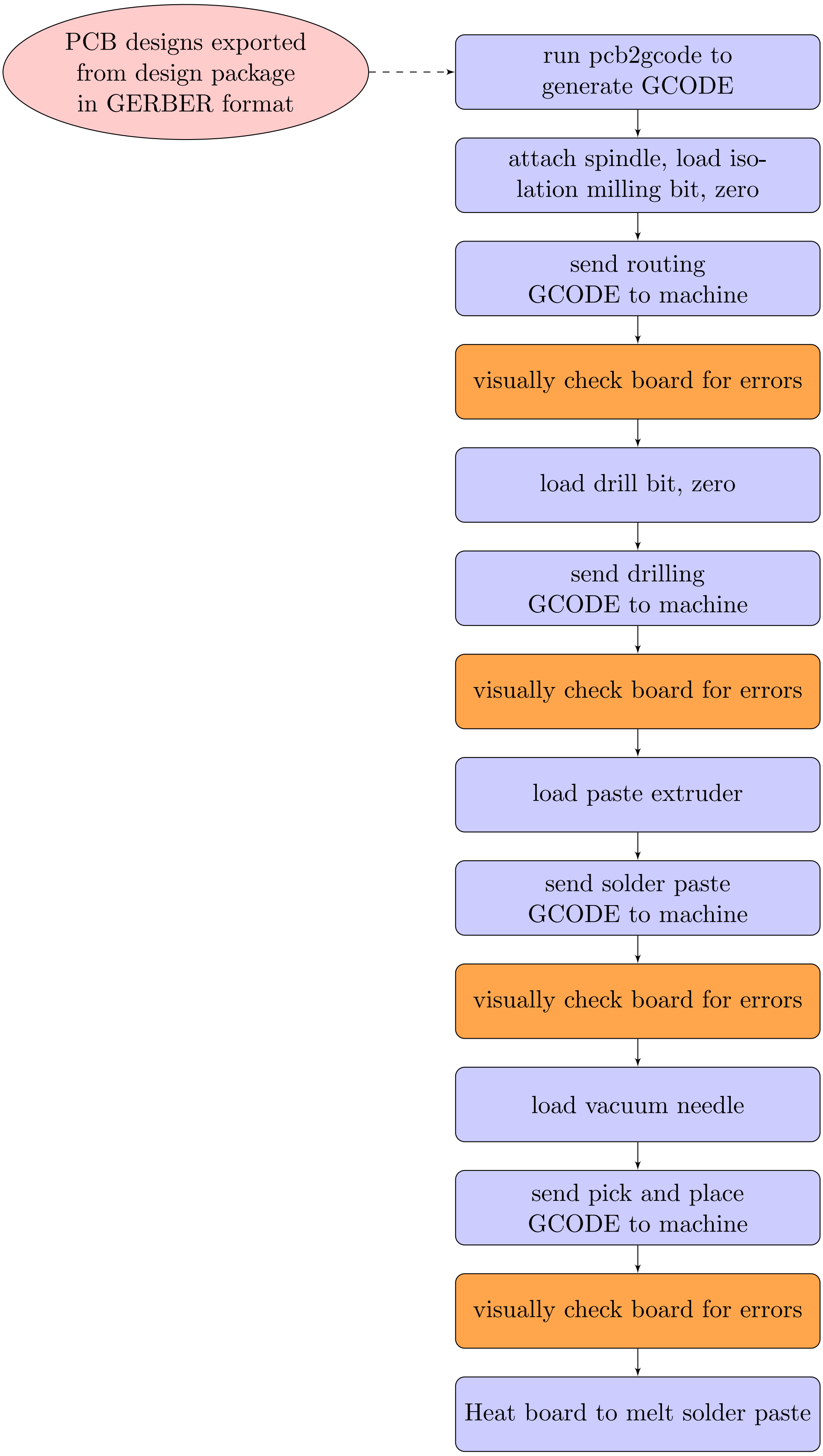
Therefore wire diameter is between 0.099mm and 0.34mm The most commonly available suitable wire is sold as 30 SWG, equivalent to 0.315mm diameter.

Resistance of 110.84mm of 0.315mm diameter nichrome wire:

$$\text{Resistance} = \frac{\rho L}{A} = \frac{1.0 * 10^-6 * 110.84 * 10^-3}{\pi * (\frac{0.315 * 10^-3}{2})^2} = 1.42\Omega$$

$$\text{Maximum current draw} = \frac{\text{V}}{\text{R}} = \frac{12}{1.42} = 8.45\text{A}$$

15 Operating Steps



16 Zeroing the axes

When the machine is used it is necessary to ”zero” the axes to a known datum point on the board. This is particularly important when swapping tools, as there may be a small amount of slop in the tool mounts which will vary between tools.

16.1 X and Y axes

In order to zero the machine in the X and Y directions, an ”L-shaped” piece of metal has been added to the bed (insert pictures/diagrams). All of the tools are metal, and therefore conductive. The axes are first manually moved to be near-zero. The tips of the tools are all electrically connected to an input pin on the AVR, with a weak (10K) pullup to 5V. The ”L-shaped” piece of metal is connected to ground. When instructed, the firmware slowly moves the X or Y axis in the negative direction until a contact is made, and the input goes LOW. At this point, the exact position of the tool is known. For convenience it is important that the location of the centre of the tool is known, this can simply be calculated as the diameters of all of the tools are known and fixed.

16.2 Z axis

Slightly different mechanisms are used for the zeroing of the Z axis when used for different tools:

16.2.1 Vacuum needle

A small microswitch is part of the vacuum needle assembly. It is triggered when the needle is touching either the bed or the top of a component. As a result, zeroing can be carried out simply by moving the tool downwards until it is triggered.

16.2.2 All other tools

The other tools are zeroing electrically using the same mechanism as for the X and Y axes. The tools are lowered until contact is made between the PCB and the tool. A temporary contact must be made from ground to the copper pcb, this is achieved with an alligator clip.

17 Future Work

This section contains envisioned improvements to the machine that were not implemented due to time constraints.

17.1 Double-sided boards

It is common to use double sided PCBs in order to achieve more compact circuits. The machine will not currently have the ability to process such boards, although this would be possible in a later version.

17.1.1 Challenges

There are a number of challenges associated with working with dual sided boards:

1. Clamping the PCB to the bed of the machine is more complex with the second side of a double sided board. The components that have already been populated will not allow the board to sit flat and level.

2. The PCB material is a moderately good conductor of heat. When the second side of the board is heated to melt the solder paste, there is a danger that components on the bottom will be loosened and fall off.

3. Both sides of the board must be precisely aligned so that any through-holes or vias line up.

4. Any vias that are required will add considerable complexity. Commercially they are manufactured in a multi-stage chemical process which is not feasible for the hobbyist.

17.1.2 Envisioned solutions

1. As isolation routing places the greatest stresses on the board mounts, it is unlikely that the reverse side of the board can be manufactured after the first side has been populated with components. Therefore the isolation routing should be carried out on both sides first. Any holes through the board should also be drilled, and cutting the board out also (leaving tabs to hold it in place). When this has been done, the rest of the manufacturing process can be continued with less secure mounting from the board edges.

2. Melting solder on one side of the board while leaving the other side solid will require active cooling of the bottom of the board. Commercially components are often glued so that all the solder paste can be reflowed in one process, but the application of adhesive adds a complex extra process to the machine which should be avoided.

3. Alignment may be achieved by the use of a small USB microscope. The user would be able to see an image of the board, and select datum points (for example the center of a pad). Once the board was flipped, the same datum points could be located. From this, a transformation matrix can simply be computed and applied to the coordinates of the GCODE.

4. A reasonable substitute for vias is to drill a hole through the board, place a short piece of wire into it and solder both sides to the copper. It is likely that the only practical way to achieve this is manually once the board has been manufactured. The ability to perform

light milling operations with the machine could be used to mill the ends of the wire much closer to the surface of the PCB than could be achieved with wire cutters. In this way it is likely that thermal vias under components could be achieved (Vias used for the purpose of heatsinking are placed under the thermal tab of an SMD component to conduct heat to copper on the reverse side of the board).

Appendices

A Servo tester

The ESC used to control the brushless spindle motor is controlled with standard "servo" control. This consists of variable length pulses sent approximately 20ms apart. The length of pulses corresponds to the position of the servo, or in this case the desired operation speed of the brushless motor. The pulses are specified to be 1-2ms, which correspond to the minimum and maximum desired speeds.

During testing it because apparent that it would be useful to have manual control of spindle speed. I designed a simple circuit to provide the necessary pulses.

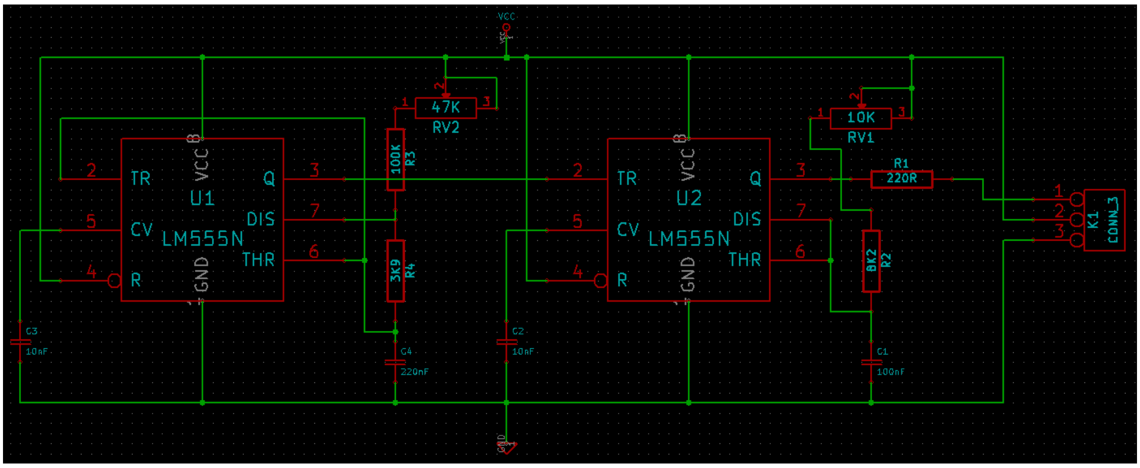


Figure 6: Servo tester schematic.

The circuit works by using one 555 timer to generate a square wave of approximately 50Hz. This triggers a second 555 timer, which operates in monostable mode, with a pulse length that is dependent on the position of a potentiometer. The 50Hz wave satisfies the condition of pulses that should be sent approximately every 20ms, and the pulses are varied between 1-2ms. The speed controllers contain a 5V linear regulator which is connected to one pin of the servo connection. This powers the 555 timers so that no other power supply is required for operation. The circuit performed as expected and has been useful in project development.

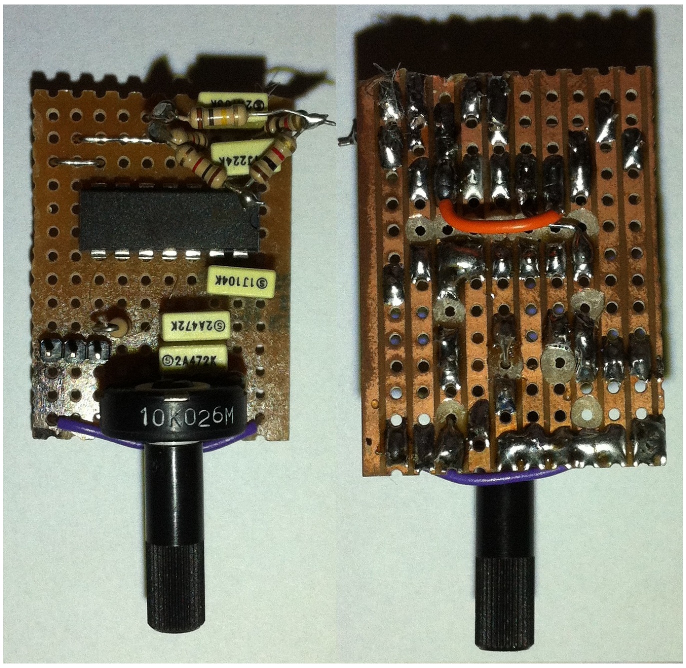


Figure 7: Servo tester board (front and back).

B Calliper serial output modification

During the testing of accuracy it was necessary to take a large number of positional measurements in all directions. To speed up this process, I modified an inexpensive pair of callipers to output a serial data stream which could be logged against the position according to the firmware. Instructions located at http://www.robotroom.com/Caliper-Digital-Data-Port.html

The positional data can be read from four traces on the calliper internal PCB. For convenience, I brought out the +V, ground, data and clock lines on to four 2.54" pin headers. As the callipers are powered by a 1.5V battery, NPN transistors are used to amplify the signals to 5V. An ATTINY2313 decodes the data stream and converts it to a serial stream, sent via a USB-serial cable to the PC.

Figure 8: Modified callipers (Data port highlighted in red).

C SMD component types

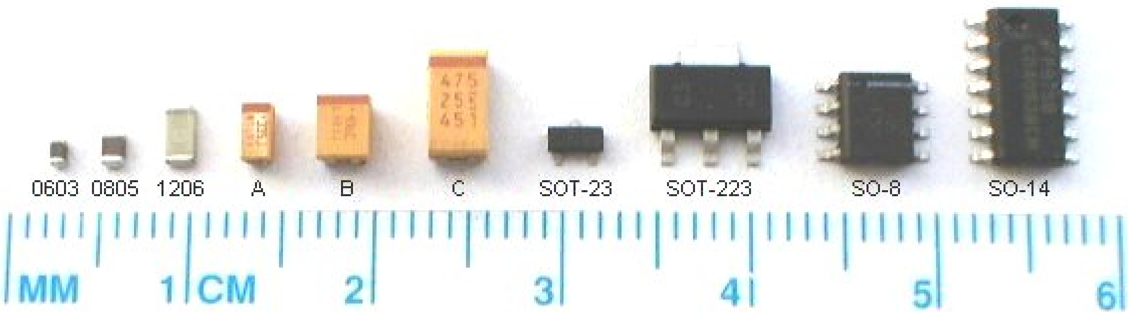


Figure 9: Example SMD components (From fpga4fun.com)

A large variety of SMD types and sizes are available. A selection is described here (from wikipedia, cite?)

C.1 Two-terminal passive components (mostly resistors and capacitors)

Package types are given a four digit code, which describes the length and width in mm.

Example:

0603 is a 0.6 x 0.3mm rectangular package.

C.2 SOT: Small-Outline Transistor

Despite the name, these devices are not limited to single transistors, other types (eg linear regulators) are common in the same packages.

Example:

SOT-223 is a 6.7x3.7x1.8mm body, three terminals plus a heat transfer pad.

C.3 SOIC: Small-Outline Integrated Circuit

Dual-in-line devices with 8 or more gull-wing lead form pins, spacing of 1.27mm

D Building pcb2gcode from source

The pcb2gcode source should first be obtained:

```
git clone git@github.com:JamesGlanville/pcb2gcode-metric.git
```

D.1 Compilation in Linux

For these instructions the use of Ubuntu/Debian is assumed, although other distributions should be similar.

To install prerequisites:

```
sudo apt-get install build-essential automake autoconf libtool libboost-all
    -dev libgtkmm-2.4-dev gerbv
```

To compile pcb2gcode:

```
cd pcb2gcode-metric
./git-build.sh
```

(Optional) To install pcb2gcode:

```
sudo make install
```

D.2 Compilation in Windows

Installing prerequisites is more complex in Windows:

D.2.1 Visual Studio

Install Visual Studio 2012 Professional from:

```
www.visualstudio.com
```

D.2.2 GTK+ development files

Download:

```
http://ftp.gnome.org/pub/gnome/binaries/win32/gtk+/2.24/gtk+-bundle_2
    .24.10-20120208_win32.zip
```

and extract to pcb2gcode-metric folder.

D.2.3 Gerbv

download gerbv from http://sourceforge.net/projects/gerbv/ gerbvinst-2.6.1.exe and install TODO: add instructions for generating .lib file

D.2.4 Boost libraries

```
download boost_1_55_0.zip from http://sourceforge.net/projects/boost/
```

unzip contents to c:
boost155 open VS2012 x86 Native Tools Command Prompt, from C:
ProgramData
Microsoft
Windows
Start Menu
Programs
Microsoft Visual Studio 2012
Visual Studio Tools AS ADMINISTRATOR cd c:
boost155 bootstrap.bat bjam.exe

D.2.5 GTKMM

```
download http://ftp.gnome.org/pub/GNOME/binaries/win32/gtkmm/2.22/gtkmm-
    win32-devel-2.22.0-2.exe
```

from http://ftp.gnome.org/pub/GNOME/binaries/win32/gtkmm/2.22/ and install.

D.2.6 gerbv?

```
download gerbv-2.6.1.tar.gz from http://sourceforge.net/projects/gerbv/
    files/gerbv/gerbv-2.6.1/
```

and extract to c:
gerbv-2.6.1

hack appwindow.c create config.h create demo-config.h