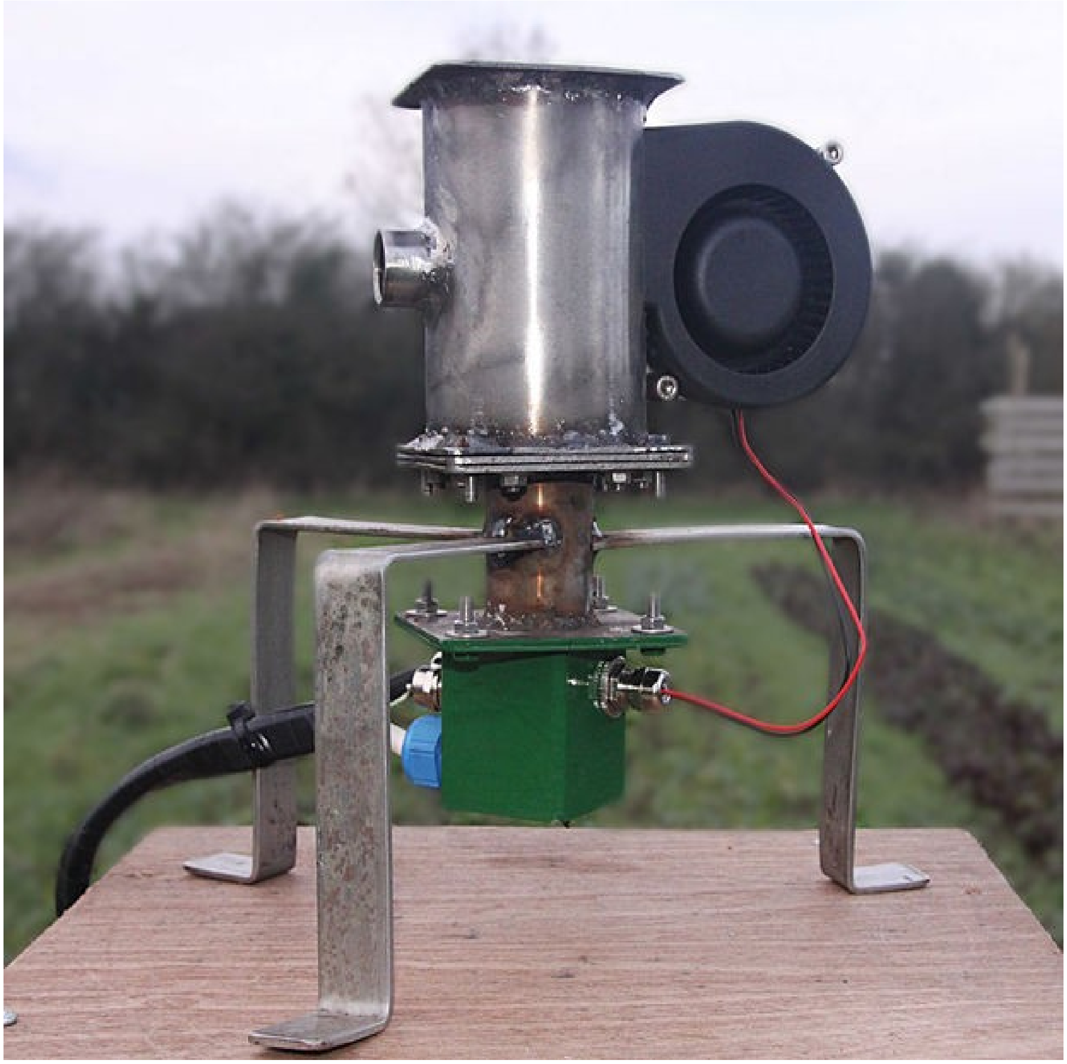
Ардуино контролиран „Фумагинатор на Оксалова киселина“

Автор: http://www.instructables.com/member/Tecwyn+Twmffat/



Within the relative chaos of my domestic arrangements, I have one small oasis of perfect order. One thing that I can call 'organised' and 'fully functional'. One thing that I can touch and know that it is reliable and ready for action at a moments notice - and that is my wonderful home made Hackable Arduino Prototyping Board.

This vaporisation project was initially envisaged to be a simple fumigation machine for beehives, but very quickly got completely out of hand and developed into so much more than that. Indeed, people are now flocking from all four corners of my island to witness the amazingly performance of this machine as it sits in the centre of my living room vaporising all manner of exotic substances, including frankincense and cheese.

Back to December ...... It is Winter 2015 and Sussex university has just published a report that concludes that vaporising oxalic acid is the best way to treat honey bees for the varroa mite. I must therefore build my vaporising machine immediately. There can be no delay .... I wonder what else I can vaporise?

The varroa is a tiny 'mite' that lives on honey bees and in their brood cells, sucking the bee's blood for nutrition. It is not particularly harmful in itself, but it does carry some very nasty diseases which are very harmful to the bees and can cause complete destruction of the colony if kept unchecked. I must act now to protect my bees.

The treatment for varroa must be performed in the coldest part of Winter, when the amount of brood cells in the hive is at it's minimum as the mites can hide in these cells and completely avoid the oxalic acid vapour. The report was only released on the 17th of December and so the pressure is on for me to build my fumigation machine before the bees start to produce more brood in the early spring.

Fortunately, I have a whole load of electronic bits and pieces left over from previous projects, so with a bit of hunting around in my untidy workshop and unsoldering of a few components here and there I was able to build this project in time for the coldest part of Winter to arrive.

Apart from my illustrious hackable prototyping board, this project was very much designed as I went along and I made no drawings or sketches before hand. I drew the main vaporisation chamber in Fusion 360 as this was the most esoteric part of the design. Much of the metal tube and plate was picked up off the workshop floor or found in dirty rat infested boxes leftover from months/years/decades gone by.

I am so pleased with the end result that the machine now doubles up as a coffee table in my lounge. Occasionally, if I am 'in the mood', I press the 'on' button and vaporise something, sometimes just for fun and sometimes to fill the air with exotic perfumes. Surely, this must be the most ridiculously over engineered air freshener ever built?

Step 1: Introduction

This Instructable actually has three major parts to it:

1. Demonstrating the features of the prototyping board by doing an actual project on it.
2. Using and coding an Arduino to control a potentially hazardous process and produce warning sounds.

3. Building an electrical heating device to vaporise and inject chemicals into a beehive to kill varroa mites.

Most likely, specific parts of this projects will be useful to people, like how to use resistance wire to build your own heater and how to control a hazardous process using an Arduino. But most useful of all is the prototyping board and, yes, the files are included for building it yourself - you just send the files to Hong Kong with a few dozens of dollars and they post you back a stack of boards. The boards have even been thoroughly tested and debugged so you know they will be perfect.

Step 2: Control of Hazardous Processes

The photo above shows a beehive being fumigated with oxalic acid using the traditional electrical heating pan inserted into the beehive entrance. Unless the beekeeper is wearing the proper mask, he will be exposed to very toxic acid fumes.

The question everybody will be asking is 'Could we not have a very long cable to the heating pan device?' and 'Why go to all the trouble of making a controller?' also 'Why not just have a battery and a big switch to operate the device?'

All of these questions are very worthwhile. Oxalic fumes are toxic to human beings and yes, we could have a long cable, a switch and a battery and hide in an underground bunker whilst operating the device remotely. Or we could use the existing products in the market place, which inevitably expose the beekeeper to nasty oxalic acid fumes.

As engineers it is our jobs to ask lots of awkward questions and look at all the potential problems in minute detail. Believe me, this will drive your boss crazy and he will soon start shouting obscenities in your general direction.

If we look at this device in particular, we can torment our imaginary boss with the following questions and statements:

"Brian, we need to be able prevent the beekeeper from being exposed to the oxalic acid fumes"

"OK Tecwyn, we'll put the device on a long cable with a big switch and hide in a bunker"

"Errrrr ..... Well ..... It's not quite as simple as that. We also need to be able to control the temperature at which we heat the powder"

To which Brian answers: "You engineers just love to find problems with everything. What's the matter with you guys?"

"Errrrr ..... Yes ...... But that's my job isn't it?"

"No" says Brian, "Your job is to make this device as quickly and cheaply as possible."

"We also need to make sure the device is cold before the beekeeper tries to put more powder in it" I say, nervously.

By this time Brian, my imaginary boss, is reaching for a large bottle of whiskey in his desk draw.

"We also need warning lights or sounds to indicate that the device is producing fumes".

Brian replies: "Well you've just succeeded in turning a cheap $50 product into a something ten times as expensive. Are you trying to bankrupt the company?"

"No" I reply, "We have to make sure that the device does it's job properly and does not harm the beekeeper".

Brian now drinks the whiskey, reaches back in his desk draw for his pump action shotgun and I run out of the building as fast as possible with shots being fired behind me.

Personally, I can not give my imaginary boss the satisfaction of seeing me take short cuts and produce a second rate fumigation machine with no proper control system, so I MUST, I simply MUST do it properly or not at all!

Eventually, I calm down and write a short list of all the desirable functions of my controller, wondering if I've still got an imaginary job to go to in the morning:

1. It should warn the beekeeper that fumes are about to be produced so that he can either walk away to a safe distance or put on a suitable mask.
2. It should control the temperature at which the oxalic acid powder is heated to so that it is not 'overcooked' or 'undercooked'.
3. It should tell the beekeeper if the device is still hot so that he does not try to add more powder too soon.
4. It should reassure the beekeeper that everything is working properly, or if there is a fault somewhere.

Then looking at the device itself:

1. It must have a fan to blow the fumes into the beehive.
2. The heating coil must produce enough heat to vaporise the powder reasonably quickly.
3. The heating coil should have a reasonable life expectancy, so should not be operated at too high a power.

4. All electrical cables must be tightly secured.
5. It must be properly tested to check for short circuits and leaks.

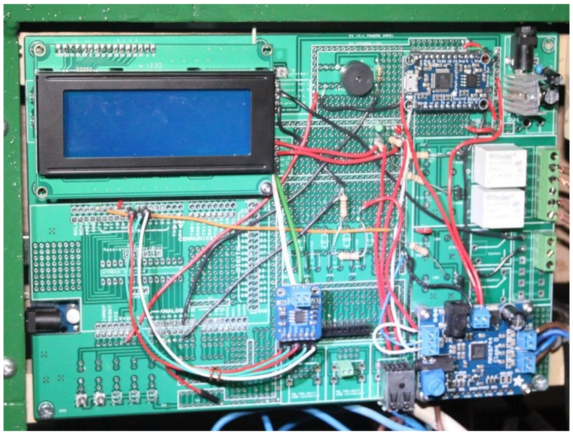
Hopefully Brian will be in a better mood tomorrow.

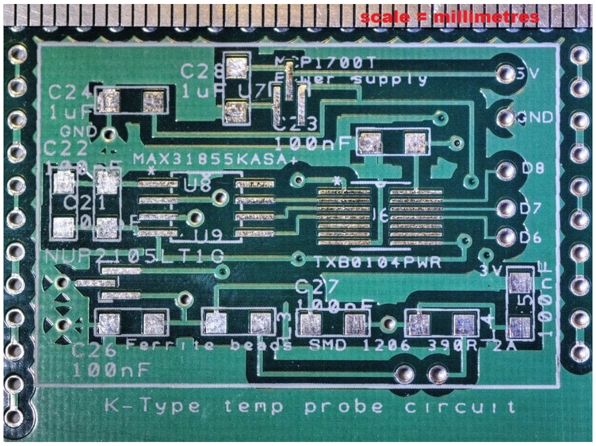
As far as the Arduino goes, the code to warn the beekeeper that the temperature of the pot is too high for refilling straight away would look something like this:

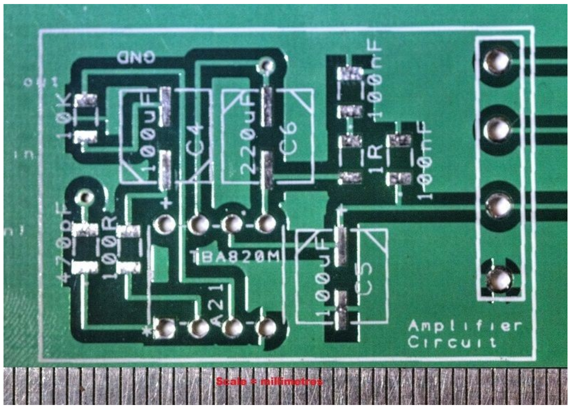
```
while(heat > 100) {
heat = thermocouple.readThermocouple(CELSIUS);
lcd.clear();
lcd.setCursor(0,0); lcd.print("Please wait while");
lcd.setCursor(0,1); lcd.print("the device cools");
lcd.setCursor(0,2); lcd.print("to below 100");lcd.print((char)223);lcd.print("C");
lcd.setCursor(0,3);lcd.print("Probe Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C.");
tone(A2, 300, 100); // plays a 300 hz sound for 100ms
delay(3000);
}
```

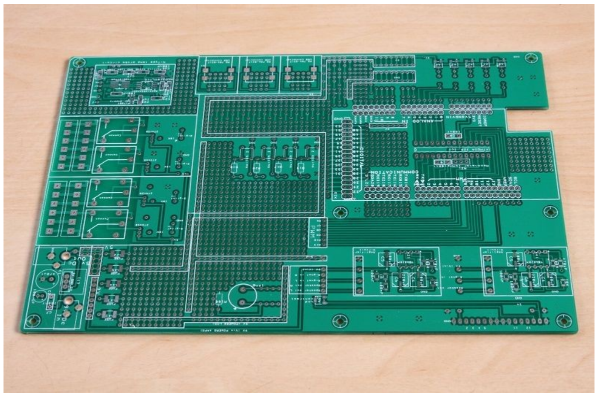
In this bit of code, the K type temperature sensor is being read and it's value is declared to be 'heat'. The LCD screen is then cleared and a message is printed to tell the beekeeper that the pot is too hot and that he must wait until it cools down. An alarm noise is broadcast every 3 seconds as the code loops. The controller will stay in this 'while' loop until the conditions of the loop are met ie. that the 'heat' is no longer greater than 100 ºC.

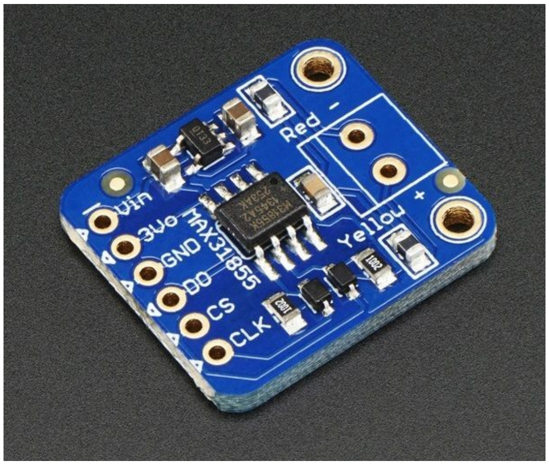
Step 3: The Arduino Hackable Prototyping Board

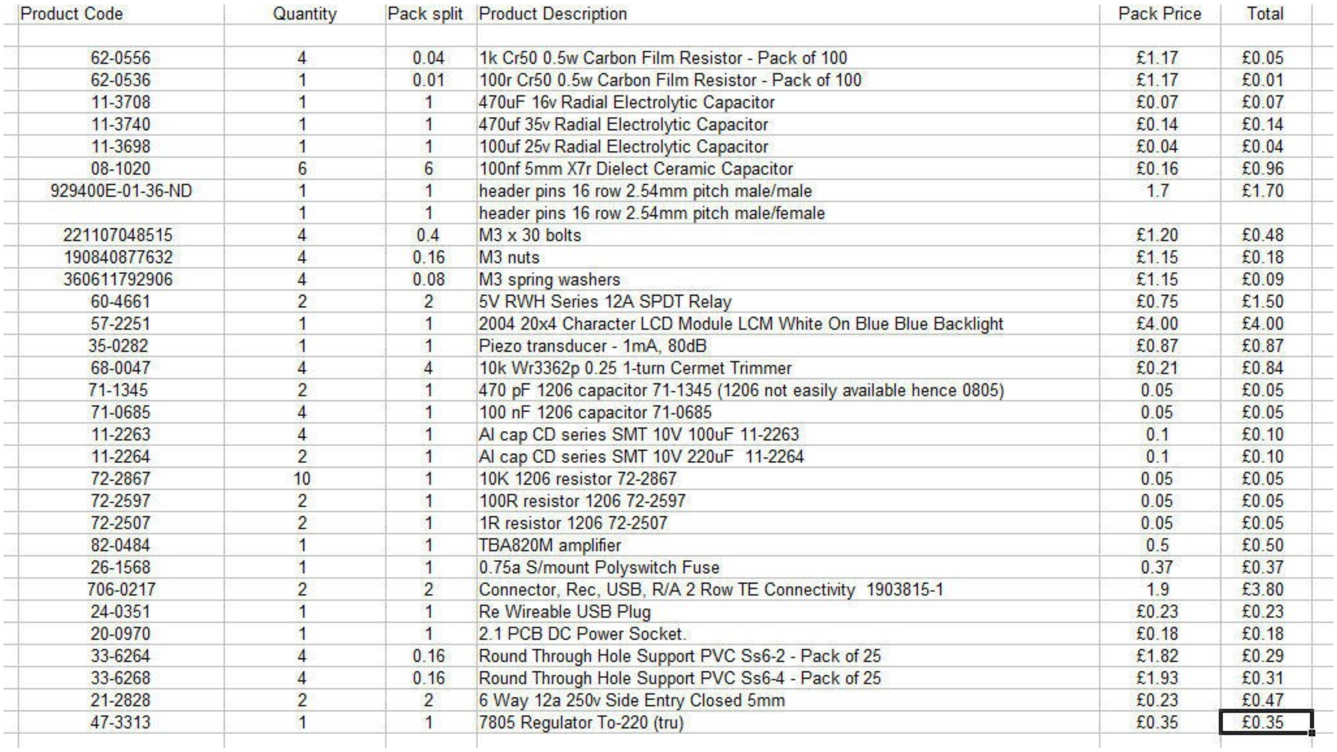












Product Code	Quantity	Pack split	Product Description	Pack Price	Total
62-0556	4	0.04	1k Cr50 0.5w Carbon Film Resistor - Pack of 100	£1.17	£0.05
62-0536	1	0.01	100r Cr50 0.5w Carbon Film Resistor - Pack of 100	£1.17	£0.01
11-3708	1	1	470uF 16v Radial Electrolytic Capacitor	£0.07	£0.07
11-3740	1	1	470uf 35v Radial Electrolytic Capacitor	£0.14	£0.14
11-3698	1	1	100uf 25v Radial Electrolytic Capacitor	£0.04	£0.04
08-1020	6	6	100nf 5mm X7r Dielect Ceramic Capacitor	£0.16	£0.96
929400E-01-36-ND	1	1	header pins 16 row 2.54mm pitch male/male	1.7	£1.70
	1	1	header pins 16 row 2.54mm pitch male/female		
221107048515	4	0.4	M3 x 30 bolts	£1.20	£0.48
190840877632	4	0.16	M3 nuts	£1.15	£0.18
360611792906	4	0.08	M3 spring washers	£1.15	£0.09
60-4661	2	2	5V RWH Series 12A SPDT Relay	£0.75	£1.50
57-2251	1	1	2004 20x4 Character LCD Module LCM White On Blue Blue Backlight	£4.00	£4.00
35-0282	1	1	Piezo transducer - 1mA, 80dB	£0.87	£0.87
68-0047	4	4	10k Wr3362p 0.25 1-turn Cermet Trimmer	£0.21	£0.84
71-1345	2	1	470 pF 1206 capacitor 71-1345 (1206 not easily available hence 0805)	0.05	£0.05
71-0685	4	1	100 nF 1206 capacitor 71-0685	0.05	£0.05
11-2263	4	1	Al cap CD series SMT 10V 100uF 11-2263	0.1	£0.10
11-2264	2	1	Al cap CD series SMT 10V 220uF 11-2264	0.1	£0.10
72-2867	10	1	10K 1206 resistor 72-2867	0.05	£0.05
72-2597	2	1	100R resistor 1206 72-2597	0.05	£0.05
72-2507	2	1	1R resistor 1206 72-2507	0.05	£0.05
82-0484	1	1	TBA820M amplifier	0.5	£0.50
26-1568	1	1	0.75a S/mount Polyswitch Fuse	0.37	£0.37
706-0217	2	2	Connector, Rec, USB, R/A 2 Row TE Connectivity 1903815-1	1.9	£3.80
24-0351	1	1	Re Wireable USB Plug	£0.23	£0.23
20-0970	1	1	2.1 PCB DC Power Socket.	£0.18	£0.18
33-6264	4	0.16	Round Through Hole Support PVC Ss6-2 - Pack of 25	£1.82	£0.29
33-6268	4	0.16	Round Through Hole Support PVC Ss6-4 - Pack of 25	£1.93	£0.31
21-2828	2	2	6 Way 12a 250v Side Entry Closed 5mm	£0.23	£0.47
47-3313	1	1	7805 Regulator To-220 (tru)	£0.35	£0.35

- Originally, the prototyping board was born as a result of building a device for detecting methanol vapour around biodiesel production tanks. Basically, I got fed up with continually having to redesign one small circuit board for one small project so decided to make the board four times bigger and put all the features I could possibly need for all my current and future projects into it. The features are listed here:

- Detachable 4 x 20 LCD screen.
- 10 amp relays x 4.
- USB sockets x 6.
- Detachable Arduino Due or Uno.
- Ability to attach 3rd part shields direct to the Arduino.
- Optional hard wired detachable ATMEGA 328 chip.
- Audio buzzer.

- Audio amplifier circuits for high powered external speakers.
- K - type thermocouple circuit for high temperature probes.
- DC power regulator circuit.
- Potentiometers x 4.
- Push button switches x 3.
- Various arrangements of empty pins in groups of 2,3 and 4 for general hacking.
- 5V, 9V and ground busses travelling all around the board.
- Mounting holes for bolting into a box.
-
- Double sided, gold plated board, with very durable pin sockets for re-working (The board MUST be double sided).

Be warned, the board does use some very small surface mounted components (SMC), but the EEVblog video tutorial shows exactly how to do this. The advantages of using SMCs is that we can use the most up to date components and they save a huge amount of space so more of the board can be used for other things.

Obviously, we could just buy a Thermocouple Amplifier MAX31855 breakout board from adafruit, or do what I do, which is to start of with one of these break out boards to test that my code works properly and then upgrade the board by hard soldering all the components on directly. The advantage with hard soldering is that the circuit is a lot more robust and resistant to vibration and general abuse out there in the real world. One thing to remember is that the MAX31855 chip has an internal reference temperature sensor and the chip does get slightly warm during use. Also, it needs to be positioned well away from any obvious heat sources on the board or otherwise, such as the LCD screen or the power regulators.

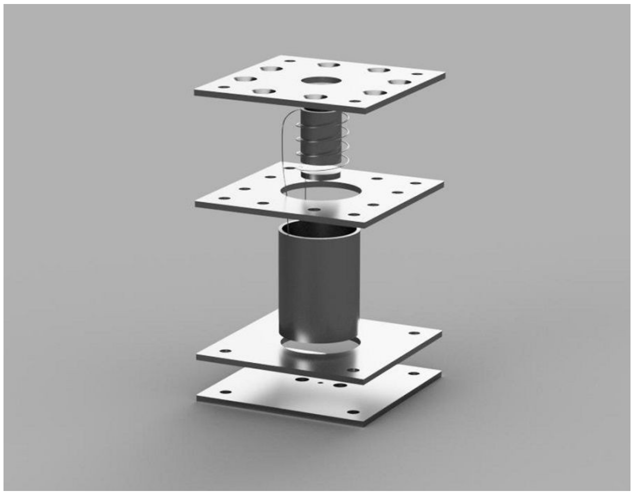
Other things to be aware of are things like digital noise - is the coding too complicated? I had this problem with the 'one wire' system of temperature measurement - it gave too many errors.

It's important that our thermocouple circuit does not fail due to vibration etc. and so too our audio amplifiers. These circuits are giving us temperature readings and audio warnings about the status of the device and so are important safety features.

It is also important to me that the board is easy to use so, for this reason, I have annotated every single component with it's manufacturer's reference number or actual value so that when I come back to it a year or so later I can just read the text and solder in the components without having lots of other files or pages to read. The information is all on the board right in front of me.

If you actually want your own prototyping board click here: http://www.freewebstore.org/goat-industries/

Step 4: Construction of the Vaporisation Device



Basically, this device works by putting 60 watts of power through a small home made heating coil wrapped around a small stainless steel pot. The oxalic acid is measured into the pot, the electricity is turned on and a blower pumps the fumes into the beehive. This device is electrically connected to the Arduino based control system so that the temperature can be monitored and the blower switched on at the right time. All we need is the temperature in the pot to reach the vaporisation (sublimation) temperature of oxalic acid point of oxalic acid: 175 degrees C.

Additionally, the coil is insulated from the pot by one wrap of ceramic fibre. A second layer of fibre then goes around the outside of the coil and the whole assembly goes inside another, bigger, stainless steel tube or 'shroud' to protect it from the elements.

Essential dimensions:

- Square plates: 86 x 86 x 3mm
- Stainless pot: OD 19, ID 16 x 25mm long
- Stainless shroud: OD 38, ID 34 x 80mm long
- 24 gauge kanthal 01 x 300mm long on 12 volt battery
- Bolts (not shown): 5mm

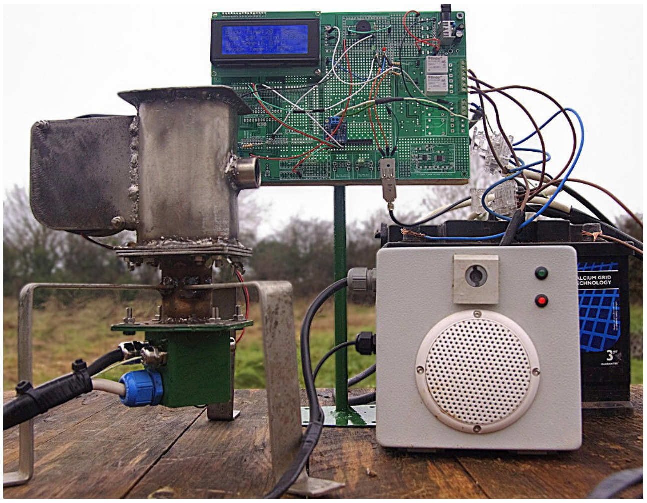
The power of the heater is given by the type of wire and it's length. When I assembled this heater I tested the amps going through the wire with a clamp meter and 12v car battery to check that there were no short circuits between the wire and the stainless components and also between the individual coils of wire touching each other.

The coil wires get attached to insulated post terminals (not shown) which screw into the bottom most plate and the thermocouple wire (last photo) pushes in through a small 3mm hole in the centre of the bottom plate up as close to the underside of the vaporisation pot as possible, but not touching it.

Winding the coils around the stainless pot is quite fiddly as the wires are very springy. With some practise, the wires can be held in the desired position and blasted with a bit of current to make them 'set' in the right place.

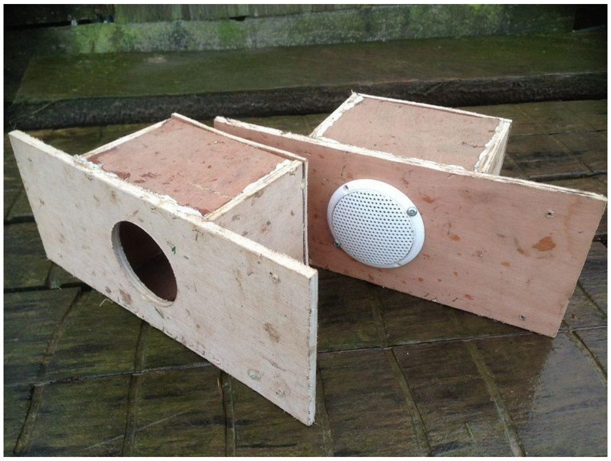
Step 5: Testing the Device with a Piece of Cheese

Виж видеото във папката „Videos“ или https://www.youtube.com/watch?v=3sYQ6gubpH0



A piece of cheese was vaporised at 180 degrees C to test the beehive fumigation machine.

Step 6: Making a Cabinet for the Machine



Not content with producing a mere prototype I decided that the cheese test was so successful that the project should go one giant leap forwards. This involved adapted the old beehive stand on the left into a new vaporiser stand which would house the car battery on the base and have the circuit board in the front.

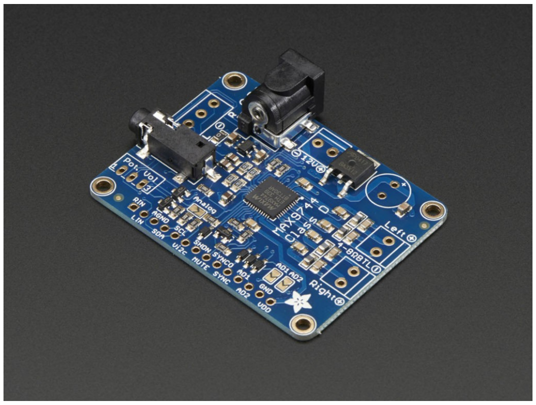
I was so excited about my sound recordings from the NASA website that I concluded that the device needed it's own 40 watt sound system to broadcast the warning noises at high volume so that the beekeeper took them a little bit more seriously. The cabinet would therefore be fitted with speakers and an additional audio amplifier board fitted to the prototype board, if I can find space for it?

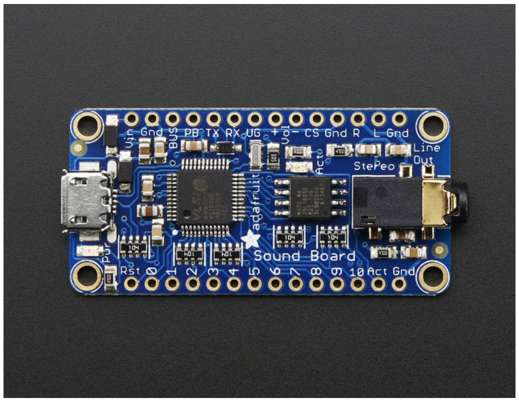
To get the most out of the speakers, they needed to be fitted in their own little plywood boxes and screwed into the cabinet.

The cabinet might, at first glance, seem a bit over the top but the electrical power draw when everything is running is quite high and so an old car battery was mounted in the bottom floor to provide enough energy for fumigating at least 13 beehives in one session.

Step 7: Warning Noises

Виж видеото от папката „Videos“ или от: https://www.youtube.com/watch?v=EgsLC20ieco





In a project like this, we can never have enough warning sounds, whether they be beeps, sirens or other special effects borrowed from the NASA website.

Some really cool beeping noises can be made on the arduino uno using the following code that I wrote myself:

```
// make beeping noises at increasing frequencies depending on temperature:
freq = startfrequency;
beeps = 0;
beepBeepThreshold = beepthreshold;
while(beeps < numberofbeepsmax){
if(heat > beepBeepThreshold)
tone(A2, freq, 9); // play a note from pin 2 at freq for 9ms
beeps++;
beepBeepThreshold = beepBeepThreshold + 10;
freq = freq + 100;
delay(10);
}
```

Earlier on in the code, the K type temperature sensor was read and it's value is declared to be 'heat'. The arduino now produces a series of beeping noises of increasing frequency, based on the temperature reading. When the device gets hot, a dramatic rapid series of loud beeps of increasing pitch are produced every second to indicate how hot it is without having to look at a screen.

Unfortunately, the 'tone' function does not seem to work on an Arduino Due.

For added drama, I thought that a second world war air raid siren and some NASA rocket launch audio might be fun. This was cobbled together in 'Fruity Loops' and windows movie maker (see video).

I added a couple of audio boards to the project, one, the Adafruit Stereo 20W Class D Audio Amplifier - MAX9744 and two, the Adafruit Audio FX Sound Board - WAV/OGG Trigger with 16MB Flash. I wanted my home made bleeps and the NASA sound effects to be REALLY LOUD! Strangely, the bleeps ended up being much louder than the audio files, even though they were put through different channels of the same amp. I can only think that this is because the bleeps are pure sine waves?

Now the vaporiser is ready for a final testing with oxalic acid and all the warning noises and vaporisation countdown.

Step 8: Arduino Code

```
#include "MAX31855.h"
#include "LiquidCrystal.h"

#include "Adafruit_MAX31855.h"

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

const int turnonAudioboard = 13;

const int audioTrigger0 = A0;

const int audioTrigger1 = A1;

const int audioTrigger2 = A2;

const int audioTrigger3 = A3;

const int audioTrigger4 = A4;

const int fanRelay = 10; //

const int heaterRelay = 9; // controls the heaters

const int countthreshold =100; //resets the max and min values every 100 seconds

const int startfrequency = 300; // set the start frequency of the beeps

const int numberofbeepsmax = 50; // set the maximumnumber of beeps possible

int fancountdown =0;

int i;

int count =5;

int freq = 0;
int beeps = 0;
int beepthreshold = -10; // The value above which beeps start to be sounded
int
beepBeepThreshold = 0;
int audioCount0 =0;
int audioCount1 =0;
int audioCount2 =0;
int audioCount3 =0;
int audioCount4 =0;
int alarmpotvalue = 500;
int tempsetpoint =160;
int blowersetpoint = 20;
int bloweron = 0;
int zeropotvalue =0;
int thermocoupleSOPin = 8;
int thermocoupleCSPin = 7;
int thermocoupleCLKPin = 6;
// temperature variables:
double setpoint;
double heat;
double output;
double junction;
MAX31855 thermocouple(thermocoupleSOPin,thermocoupleCSPin, thermocoupleCLKPin);
void setup()
{
```

```
pinMode(audioTrigger0, OUTPUT);
digitalWrite(audioTrigger0,HIGH); // Low triggers the audio sound on the FX board
pinMode(audioTrigger1, OUTPUT);
digitalWrite(audioTrigger1,HIGH); // Low triggers the audio sound on the FX board
pinMode(audioTrigger2, OUTPUT);
digitalWrite(audioTrigger2,HIGH); // Low triggers the audio sound on the FX board
pinMode(audioTrigger3, OUTPUT);
digitalWrite(audioTrigger3,HIGH); // Low triggers the audio sound on the FX board
pinMode(audioTrigger4, OUTPUT);
digitalWrite(audioTrigger4,HIGH); // Low triggers the audio sound on the FX board
delay(500);
pinMode(turnonAudioboard, OUTPUT);
digitalWrite(turnonAudioboard, HIGH);
pinMode(fanRelay, OUTPUT); // initialize the fan relay pin as an output
pinMode(heaterRelay, OUTPUT);
digitalWrite(fanRelay, LOW); // turn fan relay off.
digitalWrite(heaterRelay, LOW);
tone(A5, 400, 100); // plays a 400 hz sound for 100ms

lcd.begin(20,4); // set the lcd dimension

lcd.clear(); // LCD screen clear

lcd.setCursor(2,0); lcd.print("Varroa Blaster");
delay(2000);
tone(A5, 400, 100); // plays a 400 hz sound for 100ms
```

```
lcd.setCursor(2,1); lcd.print("Goat Industries");


delay(2000);

noTone (A5);

// Read current temperature

heat = thermocouple.readThermocouple(CELSIUS);

heat = heat-3; //
adjustment for boiling point of water

// If thermocouple problem detected

if((heat == FAULT_OPEN) || (heat == FAULT_SHORT_GND) ||
(heat == FAULT_SHORT_VCC))

delay(500);

delay(500);

// If the pot is too hot:

while(heat >
100) {

heat = thermocouple.readThermocouple(CELSIUS);

heat =
heat-3;

lcd.clear();


lcd.setCursor(0,0); lcd.print("Please wait while");
lcd.setCursor(0,1); lcd.print("the device cools");
lcd.setCursor(0,2); lcd.print("to below 100");lcd.print((char)223);lcd.print("C");
lcd.setCursor(0,3);lcd.print("Probe
Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C.");

tone(A5, 300,
100); // plays a 300 hz sound for 100ms

delay(3000);
}

if (heat<90) //plays audio0 when temperature (heat) is les than 90 degrees C.

{

digitalWrite(audioTrigger0,LOW); // Low
triggers the audio sound on the FX board

delay(500);

digitalWrite(audioTrigger0,HIGH);

}

else{}


while(count>0){

lcd.clear();


lcd.setCursor(4,0); lcd.print("Starting");
lcd.setCursor(2,1); lcd.print("vapourisation");
lcd.setCursor(0,3); lcd.print("Count:"); lcd.setCursor(6,3);
lcd.print(count); lcd.setCursor(8,3);
lcd.print("Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C");

tone(A5, 400,100);

count = count -1;

delay(1000);

noTone (A5);

}

}

void loop()

{

// Read current temperature

heat = thermocouple.readThermocouple(CELSIUS);
heat = heat-3;

junction = thermocouple.readJunction(CELSIUS);

// If thermocouple problem detected

if((heat == FAULT_OPEN) || (heat == FAULT_SHORT_GND) ||

(heat == FAULT_SHORT_VCC))

blowersetpoint =20;

//heat = 17; // temporary line for debugging

lcd.clear();

lcd.setCursor(0,0); lcd.print("Blow Time:"); lcd.setCursor(10,0);lcd.print(blowersetpoint,DEC);
lcd.setCursor(14,0);lcd.print("Secs.");

lcd.setCursor(0,1); lcd.print("Temp set point:"); lcd.setCursor(15,1);
lcd.print(tempsetpoint,DEC);lcd.setCursor(18,1);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,2);lcd.print("Int.Temp:");lcd.print(junction);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,3);lcd.print("Probe Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C");

if
((heat>100)&&(heat<105)&&(audioCount1<1)) //plays audio1 when temperature (heat) gets to 100
degrees C.

{

audioCount1 = 1;

digitalWrite(audioTrigger1,LOW); // Low
triggers the audio sound on the FX board

delay(500); // IMPORTANT This must be at least 500!

digitalWrite(audioTrigger1,HIGH);

}

else{}

if
((heat>130)&&(heat<135)&&(audioCount2<1)) //plays audio2 when temperature (heat) gets to 130
degrees C.
```

```
{

audioCount2 = 1;

digitalWrite(audioTrigger2,LOW); // Low triggers the audio sound on the FX board

delay(500);

digitalWrite(audioTrigger2,HIGH);

}

else{}

if
((heat>150)&&(heat<155)&&(audioCount3<1)) //plays audio3 when temperature (heat) gets to 150
degrees C.

{

audioCount3 = 1;

digitalWrite(audioTrigger3,LOW); // Low triggers the audio sound on the FX board

delay(500);

digitalWrite(audioTrigger3,HIGH);

}

else{}

// make beeping noises at increasing frequencies

freq =
startfrequency;

beeps = 0;

beepBeepThreshold
= beepthreshold;

while(beeps < numberofbeepsmax){

if(heat >beepBeepThreshold)

tone(A5, freq,9); // play a note from pin 1 at freq for 9ms

beeps++;
```

```
beepBeepThreshold
= beepBeepThreshold + 10;

freq = freq +
200;

delay(10);

}

noTone (A5);

if (heat

}

else
{digitalWrite(heaterRelay, LOW);bloweron = 1;

}

fancountdown =blowersetpoint;

delay(1000);

//Now the temperature has been reached and the fan must be turned on.

// Start the audio first:

if (bloweron>0){

digitalWrite(audioTrigger4,LOW); // Low triggers the audio sound on the FX board

delay(500);

digitalWrite(audioTrigger4,HIGH);

delay(14000);

}

else{}

//Turn on the fan:

while(bloweron >0){ // bloweron has the staus of '1' and so this loop is activated:

digitalWrite(fanRelay, HIGH);
```

```
if (heat

}

else
{digitalWrite(heaterRelay, LOW);

}

// Read current temperature

heat = thermocouple.readThermocouple(CELSIUS);

heat = heat-3;

junction = thermocouple.readJunction(CELSIUS);

// If thermocouple problem detected

if((heat == FAULT_OPEN) || (heat == FAULT_SHORT_GND) ||

(heat == FAULT_SHORT_VCC))

tone(A5, 1500, 100); // plays a 1500 hz sound for 100ms

delay(500);

tone(A5, 1500,100); // plays a 1500 hz sound for 100ms

delay(500);

fancountdown=fancountdown -1;

lcd.clear();

lcd.setCursor(0,0);
lcd.print("Blowing:");
lcd.setCursor(10,0);lcd.print(fancountdown,DEC);
lcd.setCursor(14,0);lcd.print("Secs.");

lcd.setCursor(0,1);
lcd.print("Temp set point:"); lcd.setCursor(15,1);
lcd.print(tempsetpoint,DEC);lcd.setCursor(18,1);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,2);lcd.print("Int.Temp:");lcd.print(junction);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,3);lcd.print("Probe
Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C");
```

```
noTone (A5);

if (fancountdown< 1){

// This creates a 'Finished' loop

while (fancountdown <1){

lcd.clear();

// Read current temperature

heat = thermocouple.readThermocouple(CELSIUS);

heat = heat-3;

junction = thermocouple.readJunction(CELSIUS);

// If thermocouple problem detected

if((heat == FAULT_OPEN) || (heat == FAULT_SHORT_GND) ||

(heat == FAULT_SHORT_VCC))

lcd.setCursor(0,0); lcd.print("Finished");
lcd.setCursor(10,0);lcd.print(fancountdown,DEC);
lcd.setCursor(13,0);lcd.print("Secs.");

lcd.setCursor(0,1); lcd.print("Temp set point:"); lcd.setCursor(15,1);
lcd.print(tempsetpoint,DEC);lcd.setCursor(18,1);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,2);lcd.print("Int.Temp:");lcd.print(junction);lcd.print((char)223);lcd.print("C");

lcd.setCursor(0,3);lcd.print("Probe Temp:");lcd.print(heat);lcd.print((char)223);lcd.print("C");

digitalWrite(heaterRelay, LOW);

digitalWrite(fanRelay, LOW);

delay(1000);

}

}

else{}
```

}

}

Step 9: Final Testing with Oxalic Acid

Виж видеото в папката „Videos“ или от https://www.youtube.com/watch?v=qG71a01LKJk

Testing the device with cheese was really easy as we could hear the cheese sizzling and then smelt a very strong cheesy smell after it had been vaporised. Testing with the oxalic acid was an entirely different matter as the vapour is invisible and to smell it would damage our lungs through inhalation of acid. Previous tests with the acid showed fumes coming out of the fan just before the fan automatically switched on, so we knew we had got the temperature / fan thing right.

Eventually, after some very extensive testing (see video above), we got down into the muddy apiary and fumigated our bees.