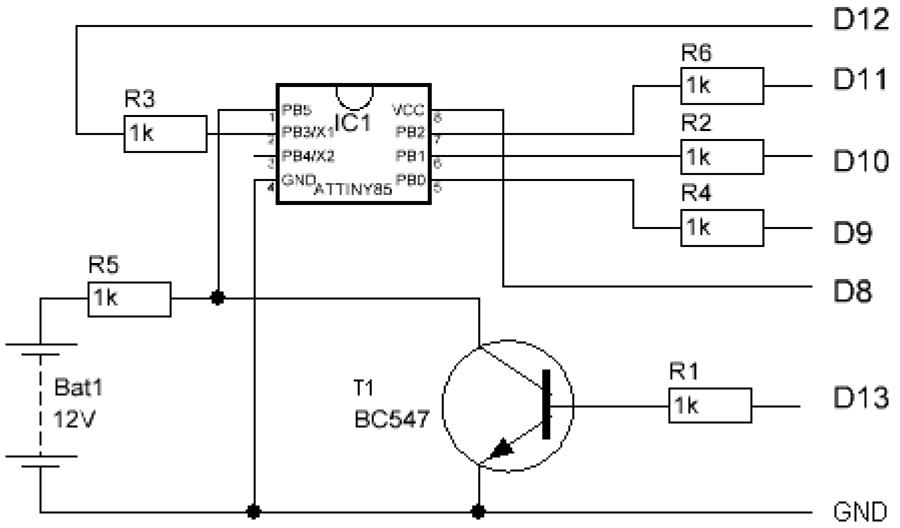
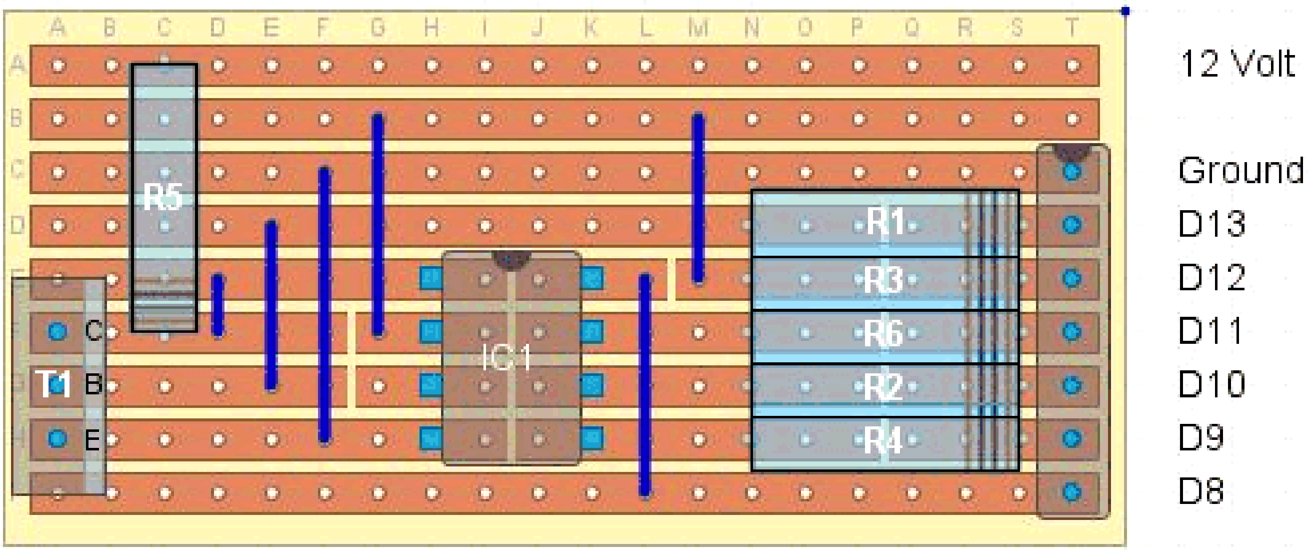
ATTiny Fuse Reset

When working with the Atmel's ATTiny series of microcontrollers they can ended-up getting "bricked" by setting a fuse value to something that disables further reprogramming. The way out of this impasse is to use a High Voltage programmer.

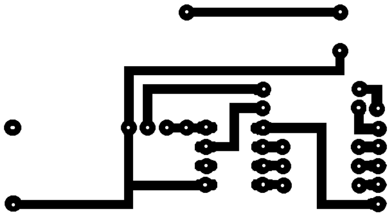
One of these can be made using a few 1kΩ resistors, one NPN transistor (BC337, BC547 or similar), a 12 volt power source (8 AA batteries, A23 battery, bench power supply or whatever) and an Arduino.



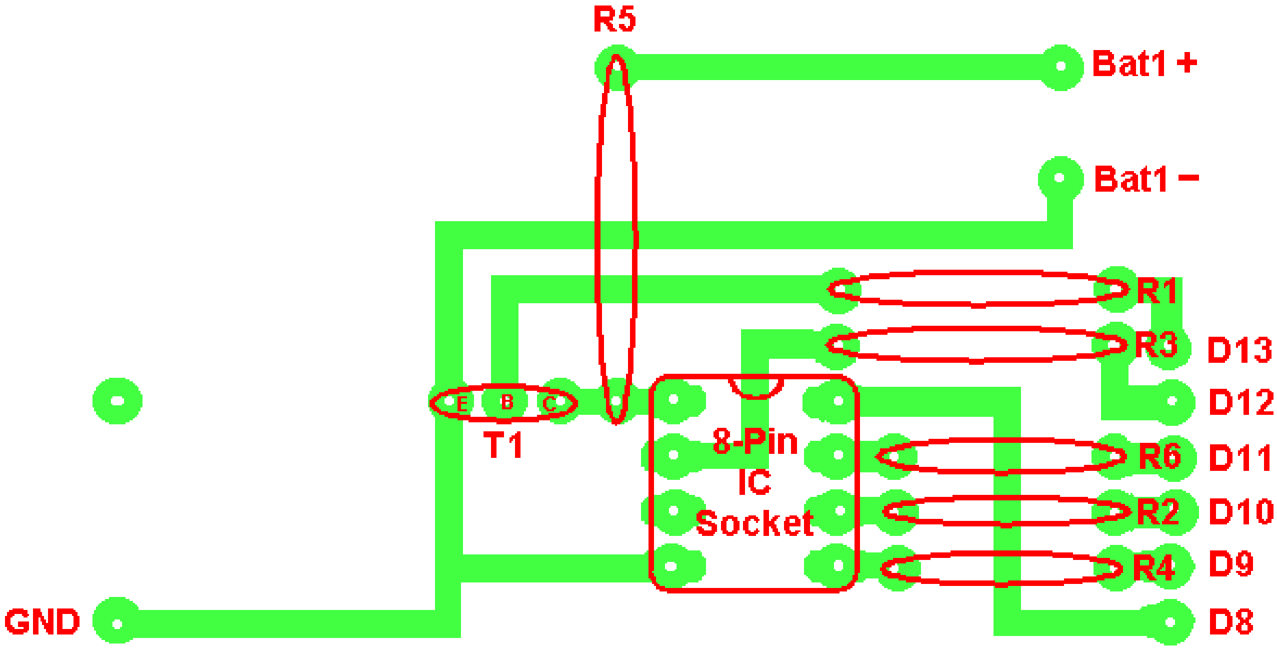
A PCB design appears below but it's simple enough to build on Veroboard:—



Connect this board to the Arduino and whatever you are going to use to supply the 12 volts needed. Next, load this sketch (it's also below in case the link doesn't work), plug in an ATTiny device that needs its fuses fixed (see list of supported devices in the source code) and open the Serial Monitor from the `Tools` menu in the Arduino IDE and select 19200 baud as the serial rate. Then, send a character to the Arduino using the Serial Monitor's "send" button to trigger the fuse reset cycle. The code will print out the device signature and before and after values of the fuses and, that it. You should now have a functional ATTiny!



PCB Artword



PCB Layout

```
// AVR High-voltage Serial Fuse Reprogrammer
// Adapted from code and design by Paul Willoughby 03/20/2010
//   http://www.rickety.us/2010/03/arduino-avr-high-voltage-serial-programmer/
//
// Fuse Calc:
//   http://www.engbedded.com/fusecalc/

#define  RST     13    // Output to level shifter for !RESET from transistor
#define  SCI     12    // Target Clock Input
#define  SDO     11    // Target Data Output
#define  SII     10    // Target Instruction Input
#define  SDI      9    // Target Data Input
#define  VCC      8    // Target VCC

#define  HFUSE  0x747C
#define  LFUSE  0x646C
#define  EFUSE  0x666E

// Define ATTiny series signatures
#define  ATTINY13   0x9007  // L: 0x6A, H: 0xFF             8 pin
#define  ATTINY24   0x910B  // L: 0x62, H: 0xDF, E: 0xFF   14 pin
#define  ATTINY25   0x9108  // L: 0x62, H: 0xDF, E: 0xFF    8 pin
#define  ATTINY44   0x9207  // L: 0x62, H: 0xDF, E: 0xFFF  14 pin
#define  ATTINY45   0x9206  // L: 0x62, H: 0xDF, E: 0xFF    8 pin
#define  ATTINY84   0x930C  // L: 0x62, H: 0xDF, E: 0xFFF  14 pin
#define  ATTINY85   0x930B  // L: 0x62, H: 0xDF, E: 0xFF    8 pin

void setup() {
  pinMode(VCC, OUTPUT);
  pinMode(RST, OUTPUT);
  pinMode(SDI, OUTPUT);
  pinMode(SII, OUTPUT);
  pinMode(SCI, OUTPUT);
  pinMode(SDO, OUTPUT);     // Configured as input when in programming mode
  digitalWrite(RST, HIGH);  // Level shifter is inverting, this shuts off 12V
  Serial.begin(19200);
}

void loop() {
   if (Serial.available() > 0) {
    Serial.print("Send a character to start");
    Serial.read();
    pinMode(SDO, OUTPUT);     // Set SDO to output
    digitalWrite(SDI, LOW);
    digitalWrite(SII, LOW);
    digitalWrite(SDO, LOW);
    digitalWrite(RST, HIGH);  // 12v Off
    digitalWrite(VCC, HIGH);  // Vcc On
    delayMicroseconds(20);
    digitalWrite(RST, LOW);   // 12v On
    delayMicroseconds(10);
    pinMode(SDO, INPUT);      // Set SDO to input
    delayMicroseconds(300);
    unsigned int sig = readSignature();
    Serial.print("Signature is: ");
    Serial.println(sig, HEX);
    Serial.print("Fuses were: ");
    readFuses();
    if (sig == ATTINY13) {
      writeFuse(LFUSE, 0x6A);
      writeFuse(HFUSE, 0xFF);
    } else if (sig == ATTINY24 || sig == ATTINY44 || sig == ATTINY84 ||
               sig == ATTINY25 || sig == ATTINY45 || sig == ATTINY85) {
      writeFuse(LFUSE, 0x62);
      writeFuse(HFUSE, 0xDF);
      writeFuse(EFUSE, 0xFF);
    }
    Serial.print("Fuses are now: ");
    readFuses();
    digitalWrite(SCI, LOW);
    digitalWrite(VCC, LOW);    // Vcc Off
    digitalWrite(RST, HIGH);   // 12v Off
    Serial.print("Finished.");
  }
}

byte shiftOut (byte val1, byte val2) {
  int inBits = 0;
  //Wait until SDO goes high
  while (!digitalRead(SDO))
    ;
  unsigned int dout = (unsigned int) val1 << 2;
  unsigned int iout = (unsigned int) val2 << 2;
  for (int ii = 10; ii >= 0; ii--)  {
    digitalWrite(SDI, !!(dout & (1 << ii)));
    digitalWrite(SII, !!(iout & (1 << ii)));
    inBits <<= 1;
    inBits |= digitalRead(SDO);
    digitalWrite(SCI, HIGH);
    digitalWrite(SCI, LOW);
  }
  return inBits >> 2;
}

void writeFuse (unsigned int fuse, byte val) {
  shiftOut(0x40, 0x4C);
  shiftOut( val, 0x2C);
  shiftOut(0x00, (byte) (fuse >> 8));
  shiftOut(0x00, (byte) fuse);
}

void readFuses () {
  byte val;
        shiftOut(0x04, 0x4C);  // LFuse
        shiftOut(0x00, 0x68);
  val = shiftOut(0x00, 0x6C);
  Serial.print("LFuse: ");
  Serial.print(val, HEX);
        shiftOut(0x04, 0x4C);  // HFuse
        shiftOut(0x00, 0x7A);
  val = shiftOut(0x00, 0x7E);
  Serial.print(", HFuse: ");
  Serial.print(val, HEX);
        shiftOut(0x04, 0x4C);  // EFuse
        shiftOut(0x00, 0x6A);
  val = shiftOut(0x00, 0x6E);
  Serial.print(", EFuse: ");
  Serial.println(val, HEX);
}

unsigned int readSignature () {
  unsigned int sig = 0;
  byte val;
  for (int ii = 1; ii < 3; ii++) {
          shiftOut(0x08, 0x4C);
          shiftOut(  ii, 0x0C);
          shiftOut(0x00, 0x68);
    val = shiftOut(0x00, 0x6C);
    sig = (sig << 8) + val;
  }
  return sig;
}
```