"How to" build custom micro controller projects

Texas Instruments MSP430 Microcontrollers

Ross Smith
PhD candidate

Wearable Computer Lab
University of South Australia

Overview

Design/Building Process

- Why we use microcontrollers
- Circuit Design
- PCB Layout Software
- Building PCBs
- MSPGCC/Debugging
- Code Examples - simple
- Past Projects
- Problems

Why we use Microcontrollers

- Prototypes
- Small – Light weight
- Low power – MSP430x
- Wearable Computing
- Customisable for each problem domain
- Cheap – after setup costs

MSP430F1232 - Specs

Yes there are others... this one is low power

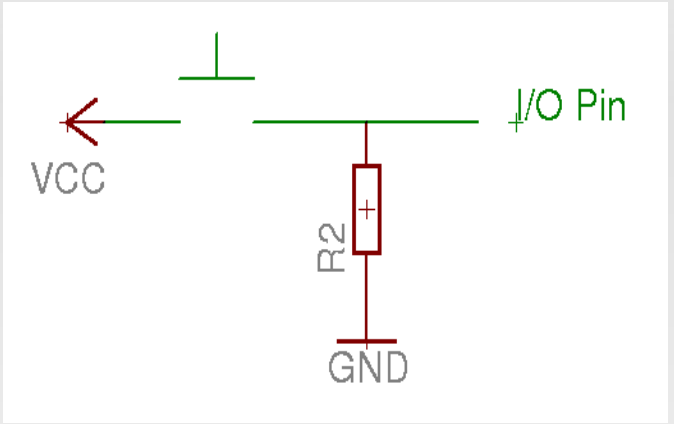
- Ultra Low Power Consumption 1.8-3.6 V
- 200uA in active mode (3.3V)
- .1uA with RAM retention (3.3V)
- 16-Bit RISC
- ADC 10bit 200 ksps
- UART Serial RS232
- 8KB Flash Memory
- 256 bytes RAM
- 22 GPIO pins
- http://en.wikipedia.org/wiki/MSP430

Circuit Design

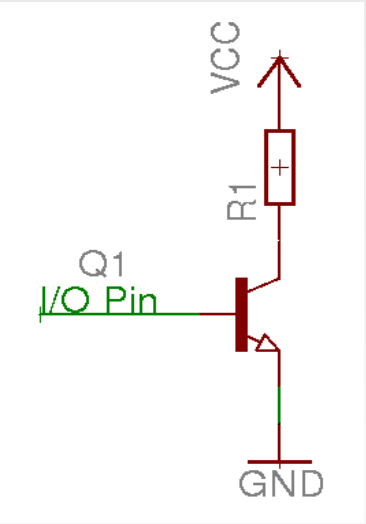
- I am not an electrical engineer
- There are many simple digital circuits DIY
- I/O pin access
- RS232 (Bluetooth)
- ADC

Circuit Design – I/O Pins

Input
- Buttons / communications
- Generally require a pull-up resistor

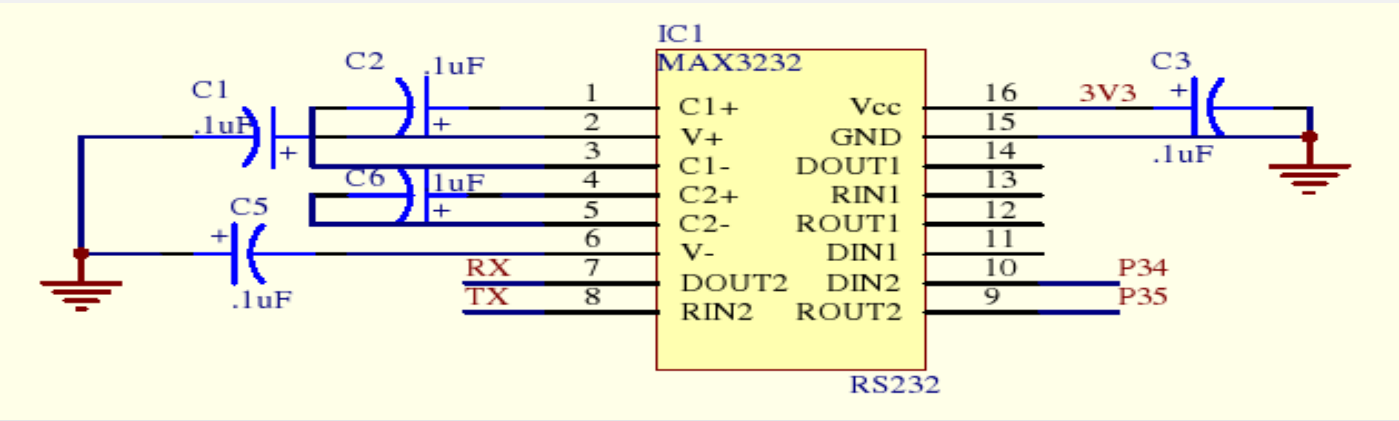


Output
- Only for very small currents
- Driving low current LEDs
- Use a transistor if larger currents are required



Circuit Design - RS232

- Level shift required if talking to PC
- Often not needed when talking to other chips
- Max232 or other varieties MAX3232 etc.



Circuit Design - ADC

- Choose a voltage level to work with 0-1.5 1.5-3
- Attach source to ADC pin
- Mainly a software problem from here (later on)
- Eg: Battery monitoring / Analogue position

Circuit Design - Notes

- SMD parts save space / but not too small
- SOP 50mil pin-pin OK
- SSOP 20mil pin-pin hard & magnification required
- SMD resistors/capacitors and diodes OK

PCB Software

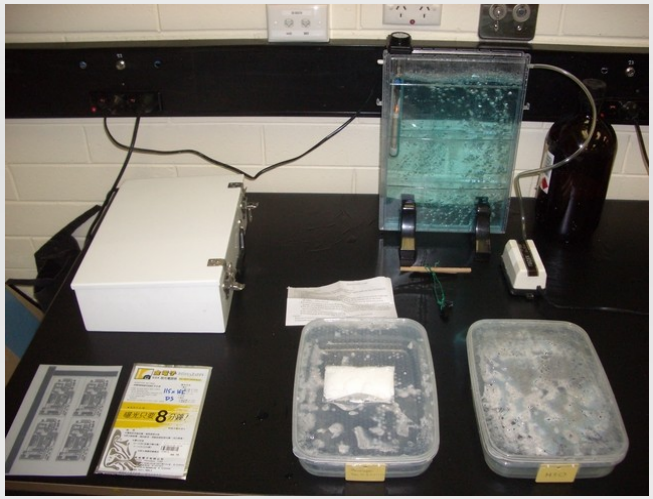
- Requirements – Schematic Layout and PCB layout
- Want to print transparent positives of PCB

- Protel – Used in the past
- Eagle – swapping to this*
- PCB – Recently trying out have to use third party schematic editor gEDA or xcircuit

PCB Software – Eagle

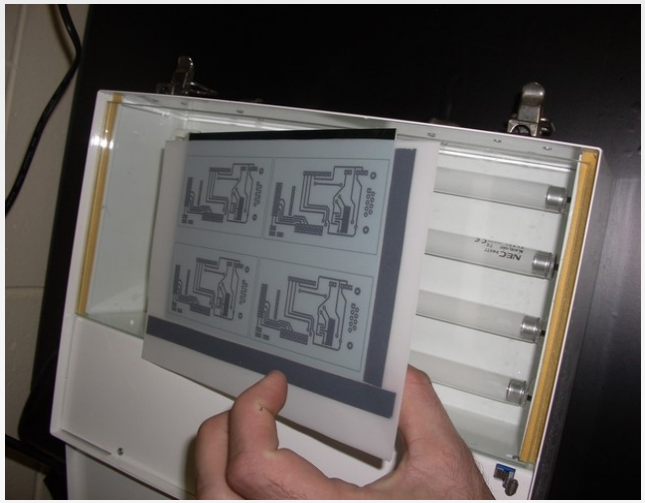
- Additional lib required for MSP430s
- Can make custom parts
- Freeware
- Limited Board size 100 x 80 mm
- Maximum double layer boards
- Only one schematic

Construction - Etching



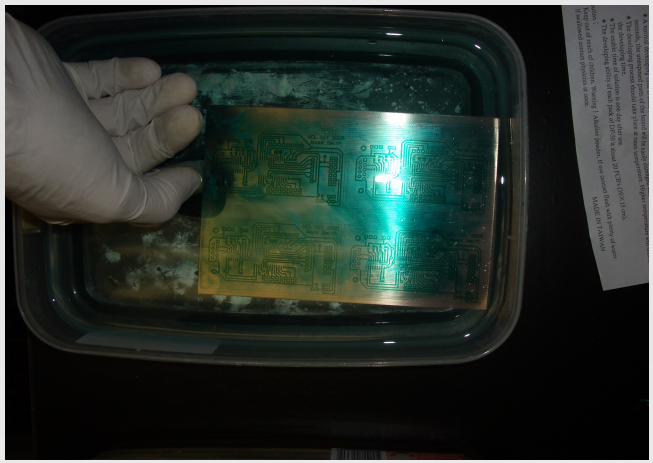
- Set up a clean work area
- Prepare transparency
- Double sided example
- Two transparencies using double sided tape
- Print on the side touching the PCB

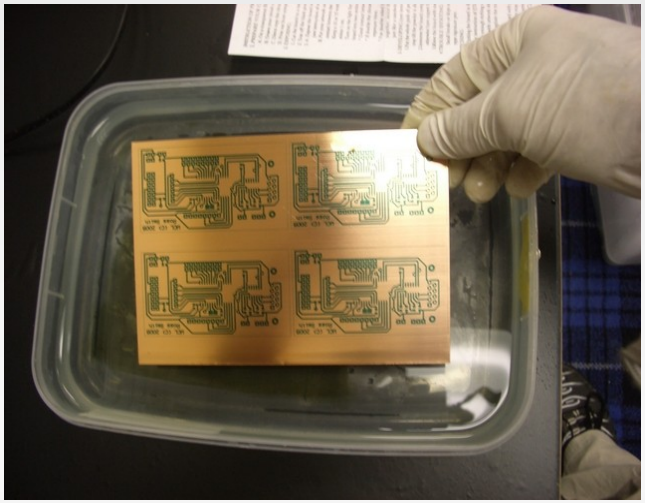
Construction



- Remove the protective plastic
- Expose using UV light box 6m30s per side

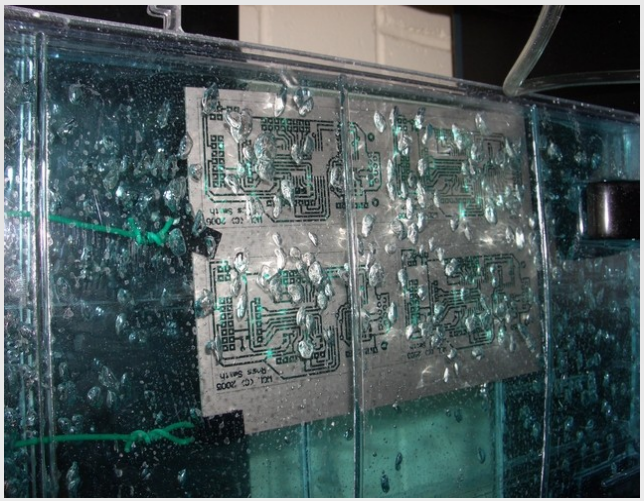
Construction

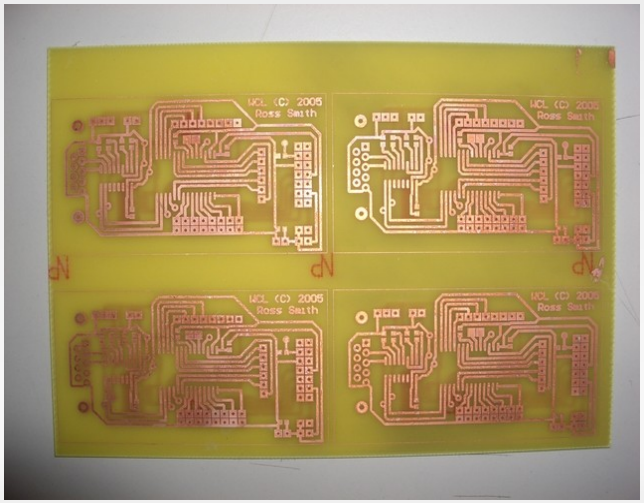




- Develop using DP-50 (Alkaline powder)
- Rinse in water before etching

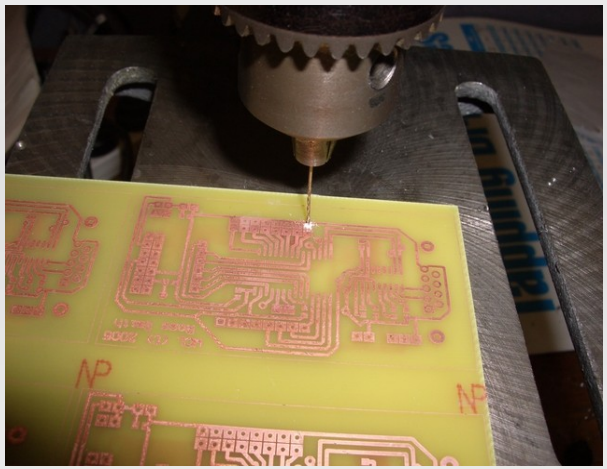
Construction





- Etch using Ammonium Persulphate NH4 (ferric chloride FeCl3)
- Common Alternative
- Remains transparent while etching
- Watch for the edges of the board to disappear

Construction



- Hand Drill holes
- CNC Mill if you have it
- Solder components
- Add vias

Acquiring Parts – Free Samples

Texas instruments – go to http://ti.com
- MSP430's
- Voltage Regulators
- Level shifting chips

Good for large quantities

Maxim - http://www.maxim-ic.com

Level shifting chips

National Semiconductors - http://www.national.com
- 3A H-Bridge chips - LMD18200

Programming MSP430s

- MSPGCC - http://mspgcc.sourceforge.net
- Provided notes I used while compiling and using mspgcc http://r-smith.net/lca2007
- Extended commands with GDB

- Code Composer Essentials (Windows) eclipse based environment

- JTAG programming / runtime debugging

Debugging - MSPGDB-proxy

- erase – target flash memory
- puc – reset over jtag using Power Up Clear
- reset – hardware reset
- identify – the msp device information
- jtag - Report how JTAG is to be handled when a program is running
- vcc – report/define the current voltage
- dump – read out target registers

monitor erase all
load myProgram

Debugging

JTAG – runtime debugging
Serial – console debugging
LEDS – watch for the flashing lights !!

Code Examples

Controlling IO Pins (Flash P1.0)

```
P1DIR = 0x01
int main(){
while(true)
  P1OUT = P1OUT ^ 0x01
    delay(1000)
}
```

Code Examples – ADC

```
  /*Disable the ADC to change the
input channel*/
  ADC10CTL0 &= ~ENC;

  /*Select the input channel P2.1*/
  ADC10CTL1=INCH_11;

  /*Re-enable the ADC*/
  ADC10CTL0 |= ENC;
```

Code Examples - ADC

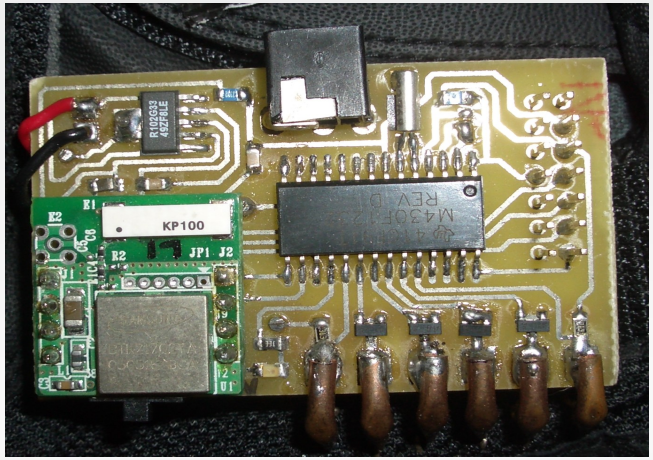
```
  /*Request a sample from the ADC, this
takes 6 clock cycles*/
  ADC10CTL0 |= ADC10SC;

  /*Go into low power mode while we wait
for a result the interrupt will wake up up
when it is ready*/
 // _BIS_SR(CPUOFF + GIE); or ADC10BUSY?

 while (ADC10CTL1 & ADC10BUSY);
  return ADC10MEM;
```

Past Projects

- Tinmith Glove Controller
- Detects button presses of fingers to thumb/palm
- Li Po Battery powered
- Monitors battery state
- Bluetooth communications
- Conductive fabric for wires in gloves
- Conductive material on finger tips for switch pads



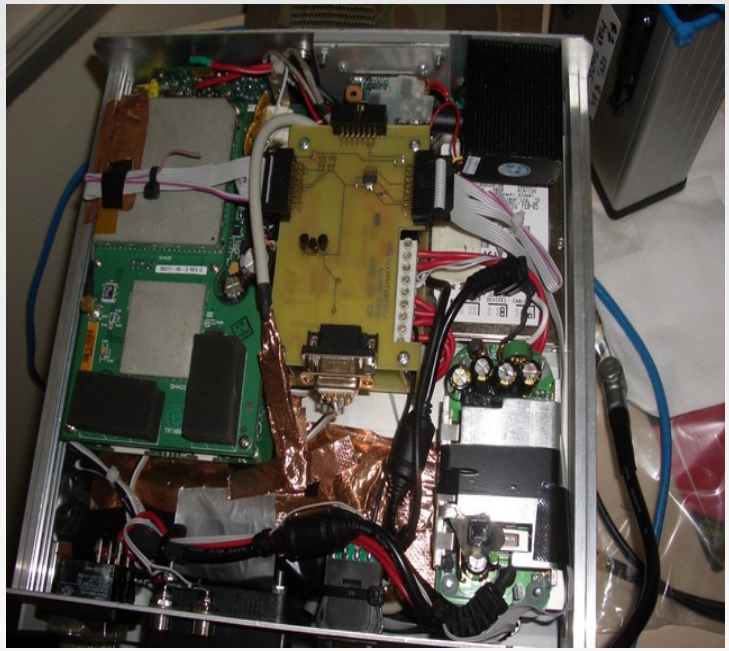
Past/Current Projects – Timnith Backpack

- Controls simple tasks but makes our life easy :)
- Uses MSP430 to init hardware
- When power is applied it:
- pulses laptop, GPS and HMD to turn on

Past/Current Projects – Tinmith Backpack

Accepts RS232 (USB) commands allowing:

- GPS reset
- Temperature reading
- Encoder allows different configurations on start up (0-9)
- Software controllable status lights
- Laptop On/Off (Cut own throat)



Interesting Problems / Limitations

- RS232 heat issues using DCO
- Avoid using Jtag pins for I/O (they are used for runtime debugging)
- 8Mhz can be a limitation depending on workload

Questions ?

Links:

http://r-smith.net/lca2007 My notes
http://ti.com - Samples
http://national.com - Samples
http://maxim-ic.com - Samples
http://www.cadsoft.de/freeware.htm - Eagle
http://sourceforge.net/projects/pcb - PCB
http://www.geda.seul.org - gEDA schematic
http://opencircuitdesign.com/xcircuit

Cost of Parts

UV Box - $300
Kinsten PCB - $7 - $30
Etchant (Ammonium Persulphate 2.5Kg) - $24
Etching Tank / Heater / Pump - $108
Developer - $3.30
Soldering Iron - $30
Drill - $60