Not Just another SMD Reflow Oven

Soldering PCB boards on Toaster Ovens is old stuff :-)

A "radical" approach for reflow soldering. A real bread toaster with a very smart controller based on the LM3S102.

Abstract

Project LM1785

Motivation

I have been using a toaster oven for more than five years on small production runs and have made quite a few thousands boards with it without any problems ! As many of us I spend a lot of time working at home and traveling to see clients in other states. Having a small and portable option for being able to work far from the office or having more space at home is the main motivation for this project.

Ok... Ok... I admit it :-) It is just really great fun to see the boards being soldered in the Toaster! The performance is incredible and exceeded all my expectations.

Objectives

- The controller should be able to give the same performance as huge 7 zones ovens except on board size and speed of soldering.
- Size is not negotiable either. It should be small and easy to transport.
- User Interface should be nice.
- NRE costs should be kept at the minimum possible as this is never intended to be mass produced.

Hardware Development Rationale

The choice of the LM3S102 was an easy one. It is small, inexpensive, free tools are available, C friendly for such a small CPU and most important for this project; it keeps NRE costs minimum because it has plenty of resources even at the small price. It is a great 8 bit “killer”, especially for small to medium volume projects.

For the user interface the decision was to use a Nokia 3310 graphical display because it is cheap, readily available and very easy to interface. Buttons for starting the cycle were also included and they make the board more versatile for future uses or expansion.

A USB interface was also added to make development and temperature profiling easier.

Temperature sensing is made using a cheap type “K” thermocouple and a MAX6675 interface chip. The thermocouple was attached to a PCB to help measuring the temperature as close as possible to the components and solder paste. Measuring air temperature inside the toaster would not help

much. Precision and durability are great and response time is more than good enough for the application. Once more this decision was heavily driven by the objective of keeping NRE cost at a minimum. Using opamps, cold junction compensation and AD convertes would not make any sense for a small series project.

The power section is galvanically isolated using one AC optocoupler for detecting the zero crossings and a optoisolated Triac Driver. The Triac is a snubberless 12 amps Triac. At the power levels involved the zero crossing activation may seem like overkill but the pulse skipping technique used keeps switching noise levels down, stresses less the switching element and the heaters and allows for much better temperature control.

Software Development Rationale

Keil IDE and Luminary driver library allows a very quick learning curve. I had the first program up and running in less than an hour! The easy of use for making complex programs and the power of the CPU that allows you to use C exclusively reduced development time dramatically when compared to the usual 8 bit's CPU's were we need to be looking all the time if the resources are going to be exausthed.

All the timing critical functions of the program are done inside timer interrupts. The "main loop" just takes care of the "decisions" that are made every quarter second and user interfacing.

The Pulse skipping Triac activation and zero crossing control is all done inside a 100uS interrupt.

In this first version only one button is used to start the soldering cycle.

A Delphi 3 software was also developed to allow easier visualization of the reflow profile performance.

Temperature Control Strategy

Kester R276 recommend reflow profile was used as the basis for the control strategy. The toaster has too much power and a simple on-off control is not enough for achieving good results. Instead of that the activation of the heaters is done at zero crossing with pulse skipping proportional to the power needed at each stage of the soldering process. An Initial preheat stage, that brings the board to 80c, was added to the standard profile to allow for the stabilization of the temperature in the toaster and PCB before attempting to follow the profile. This is necessary to diminish the effects of

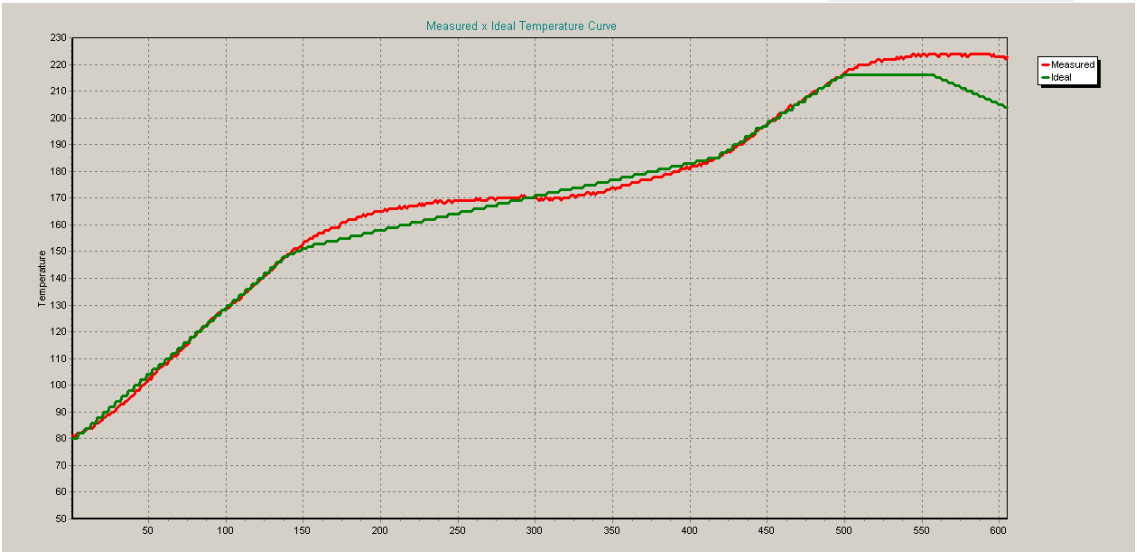
the thermal inertia of the system. After this first phase the usual preheat, soaking and reflow profiles are respected. The results are outstanding and even better than many commercial reflow ovens that I have seen, especially when compared to small 3 zones ovens.

Conclusions

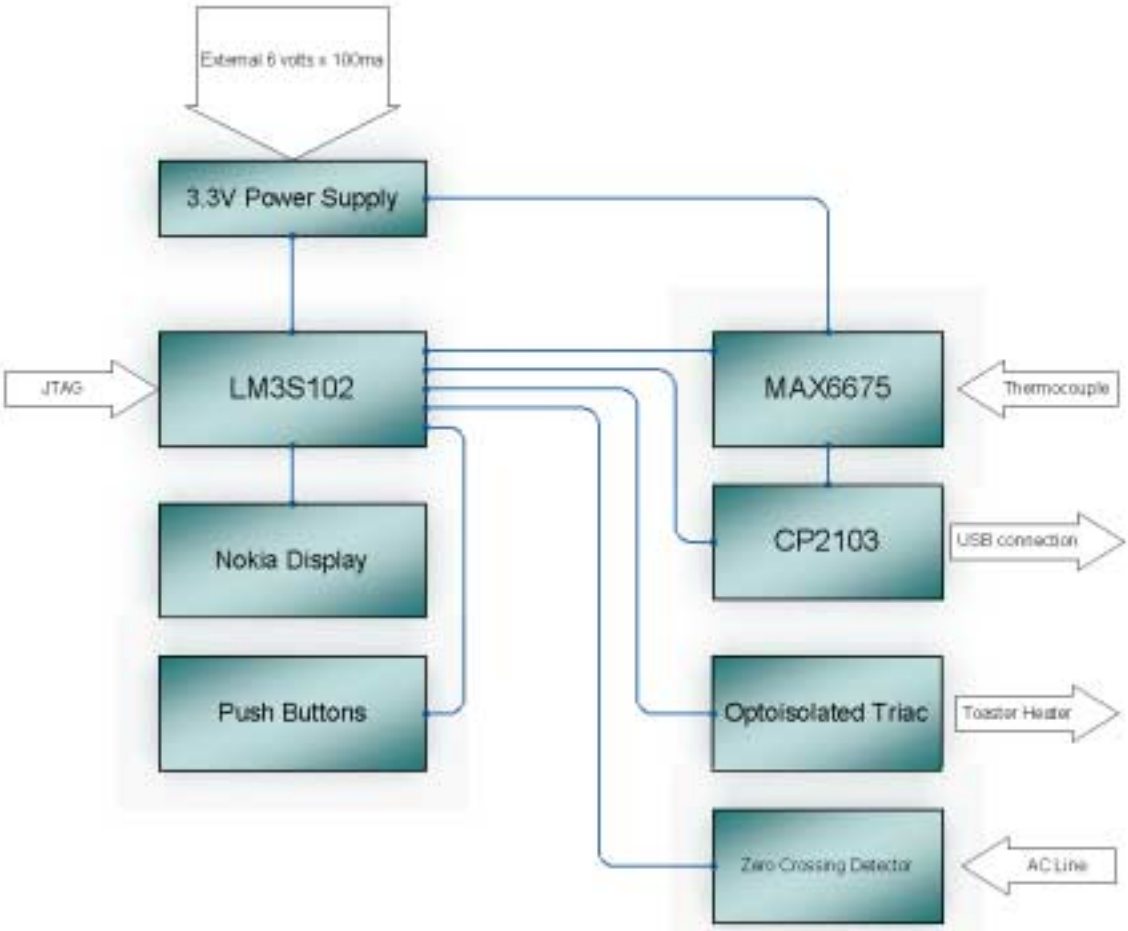
The results could not be any better! The cost to develop the project were minimal using Luminary Micro LM3S102 and Keil IDE, temperature profile is at least as good as a 7 zones reflow oven, the graphical displays allows all process information to be available at a quick glance, boards come out nicely soldered in a few minutes and the pleasure to do something in a way I have never seen done before is priceless. :-)

Project Photo

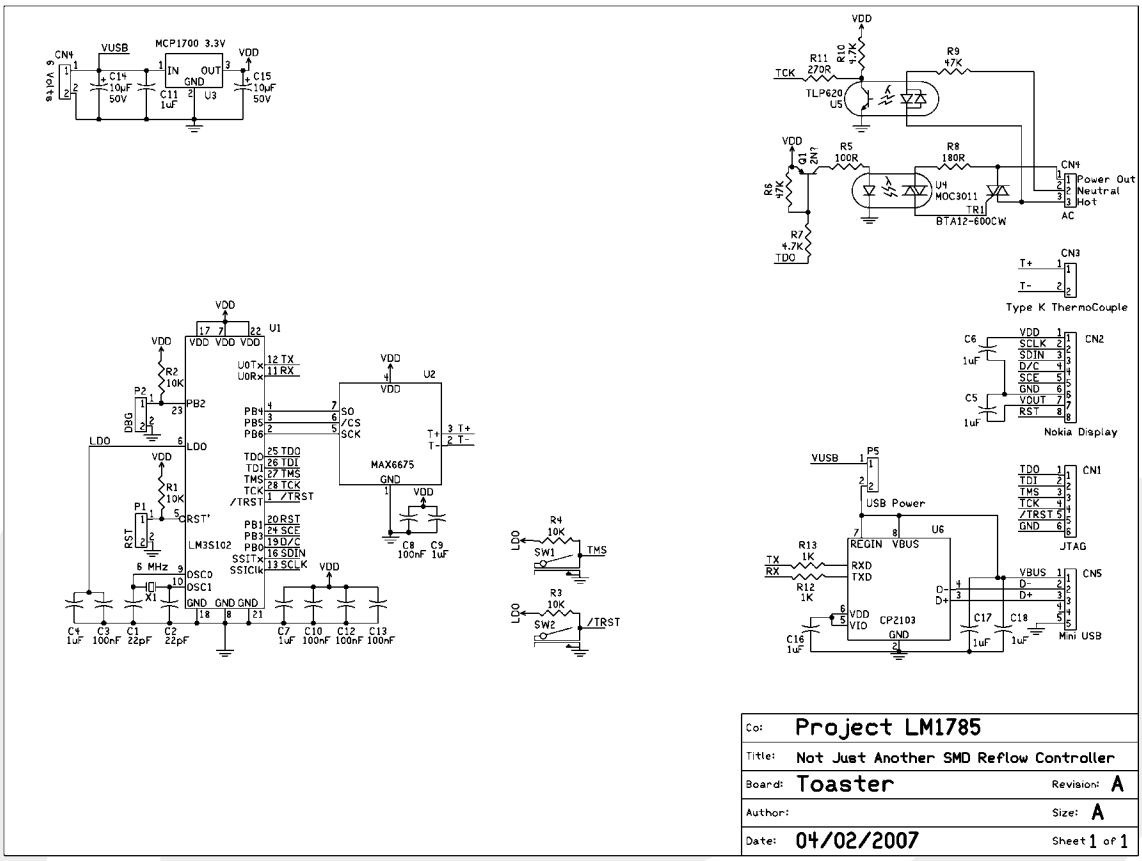
Real Temperature Profile Graphics



Block Diagram



Schematics


MCP1700 3.3V
VUSB
VDD
LM3S102
MAX6675
TLP620
MOC3011
BTA12-600CW
Power Out
Neutral
Hot
AC
Type K ThermoCouple
Nokia Display
JTAG
Mini USB
USB Power
CP2103
Project LM1785
Not Just Another SMD Reflow Controller
Toaster
Author:
Date: 04/02/2007
Revision: A
Size: A
Sheet 1 of 1

Sample of Code

```
//*****************************************************************************
// The interrupt handler for timer interrupt that controls triac driving
//*****************************************************************************
void Timer0IntHandler(void)
{
   TimerIntClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);  // Clear pending Interrupt signal

   // During the Zero crossing do not do anything, wait for it to end
   // The zero crossing pulse takes about 800 uS

   if ((GPIOPinRead(GPIO_PORTC_BASE, Zcross)))     // Test if inside Zero crossing
   {
       ZcrossActive = 1 ;
   }
   else
   {
       // Counts Zero Crossings for the pulse skipping
       if (ZcrossActive)     // Executes once at the end of the zero crossing pulse
       {
              ZcrossActive = 0 ;                   // Prepares for Next Zero Crossing
              ZcrossCount++ ;                      // Increments Cycles
              if (ZcrossCount > 4)                 // If 5 counts zero counter
                  ZcrossCount = 0 ;
              TriacOnTimer = 0 ;                   // Triac is not fired yet
       }
       if (ToasterOn)                              // If control wants more heat
       {
              if (ZcrossCount < OnCycles)          // Activates for OnCycles, limits the
              {                                    // power at each stage
                                                   // by skipping AC cycles
                 TriacOnTimer++ ;
                 if (TriacOnTimer < 20)            // Turn on the triac gate for 2 ms
                     GPIOPinWrite(GPIO_PORTC_BASE, Out1, 0) ;    // Turn on the Heater
                 else
                     GPIOPinWrite(GPIO_PORTC_BASE, Out1, Out1) ; // Off gate current
              }
       }
       else
       {
              GPIOPinWrite(GPIO_PORTC_BASE, Out1, Out1) ;        // Turn Off the Heater
       }

   }

}
```