MAX32600 TMON Demonstration

April 3, 2015

1 Abstract

The MAX32600 has an internal temperature sensor. In this application, we will demonstrate how to set up the real-time clock to generate interrupts every one second; set up AFE(analog front-end) and ADC(analog-to-digital converter) to get ADC samples from internal temperature circuit; calculate the internal temperature based on eight measurements and display the temperature in Celsius on the terminal emulator through UART.

2 Requirements

- MAX32600B EvKit
- Olimex JTAG ARM-USB-TINY-H
- GNU ARM toolchain with newlib libc
- USB Full-size A to B cable
- PC or Workstation with USB and terminal emulator software

3 Setup

On the MAX32600 EvKit, the UART0 device can be configured to connect to an on-board FTDI USB to serial converter for easier connectivity to a standard PC. The onboard FTDI converter is powered from the PC, not the EvKit itself, therefore, the USB to serial connectivity is alway alive without power requirements from the EvKit.

- Connect your PC to the full-size USB-B port on the EvKit
- Depending on your operating system and drivers you may need to install the FTDI USB to serial driver.
- Open a serial terminal emulator program. On MS Windows, you may have or need to install one of several options including but not limited to; TeraTerm, PuTTY, RealTerm.
- Find the PC serial port. On MS Windows, the USB to serial converter typically enumerates itself to COM6 or higher.
- Set your terminal emulator program for the following serial options: BAUD => 115200; No parity; No flow control;
- Compile and load the application

4 Observation

Observe the text string on the terminal emulator. The internal temperature in Celsius should be updated once per second.

5 Source Code Overview

5.1 Drivers In Use

- Instruction Cache
- Clock Manager
- Power Manager
- IO Manager
- GPIO
- SysTick
- UART
- Real-Time Clock
- TMON
- AFE
- ADC

5.2 Interrupts Enabled

- Real-Time Clock

5.3 Code Operation

- Enable Instruction Cache
- Setup Clocks (Use external 8MHz crystal and set system clock select for 48MHz PLL output divided by 2 to get 24MHz system clock)
- Enable AFE power
- Set UART pin mapping
- Set UART configuration (baud rate and serial controls)
- Set ADC voltage reference to 1.5V
- Set DAC voltage reference to 1.5V
- Enable clock control for real-time clock interrupts
- Set pre-scale of real-time clock to divide input clock by 2$\hat{1}$2 (1Hz)
- Enable real-time clock
- Setup real-time clock interrupts to be triggered every second
- Wait for interrupts and get temperature every one second

5.4 Internal Four-Current Method

$$T_{MEAS} = \frac{V_{REF}}{2^{16}} * \frac{q * [NV_{BE3} + NV_{BE2} - NV_{BE1} - NV_{BE4}]}{nk \ln \frac{I_3}{I_1} * \frac{I_2}{I_4}}$$

Normalize results: for internal sensor, the internal value is inverted, so to offset and scale the NV_{BE} values:

$NV_{BE1} = 2 \times 2^{15} - V_{BE1}$
$NV_{BE2} = 2 \times 2^{15} - V_{BE2}$
$NV_{BE3} = 2 \times 2^{15} - V_{BE3}$
$NV_{BE4} = 2 \times 2^{15} - V_{BE4}$

Substituting for I_1, I_2, I_3 and I_4 in the denominator:

$$I_1 = \frac{V_{R1}}{R_{INT}}; \quad I_2 = \frac{V_{R2}}{R_{INT}}; \quad I_3 = \frac{V_{R3}}{R_{INT}}; \quad I_4 = \frac{V_{R4}}{R_{INT}}$$

Where:
T_{MEAS}: temperature in Kelvin
q: electron charge which is 1.60219×10^{19}
V_{BE1}: ADC reading with I1 as current source
V_{BE2}: ADC reading with I2 as current source
V_{BE3}: ADC reading with I3 as current source
V_{BE4}: ADC reading with I4 as current source
V_{REF}: ADC reference voltage (1.5V in this application)
n: diode ideality which is typically 1
k: Boltzmann's constant wich is 1.3807×10^{-23} Joules/Kelvin
I_1: current source low setting 4μA
I_2: current source high setting 60μA
I_3: current source high setting 64μA
I_4: current source high setting 120μA
V_{R1}: ADC reading with I_1 as current source
V_{R2}: ADC reading with I_2 as current source
V_{R3}: ADC reading with I_3 as current source
V_{R4}: ADC reading with I_4 as current source
2^{16}: Number of ADC steps for MAX32600 16-bit ADC

To convert the measured temperature in Kelvin to Celsius, use the following formula:
°C = K - 273.15