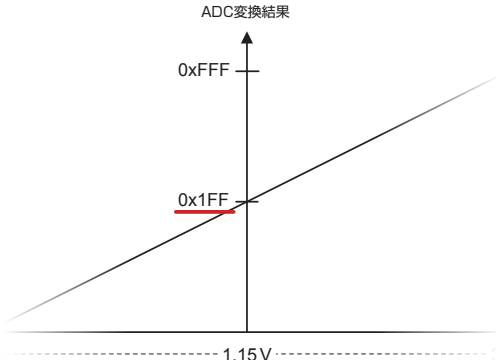
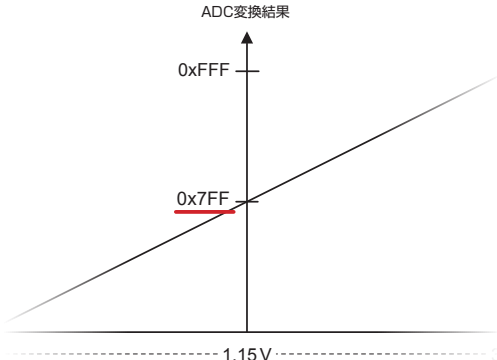


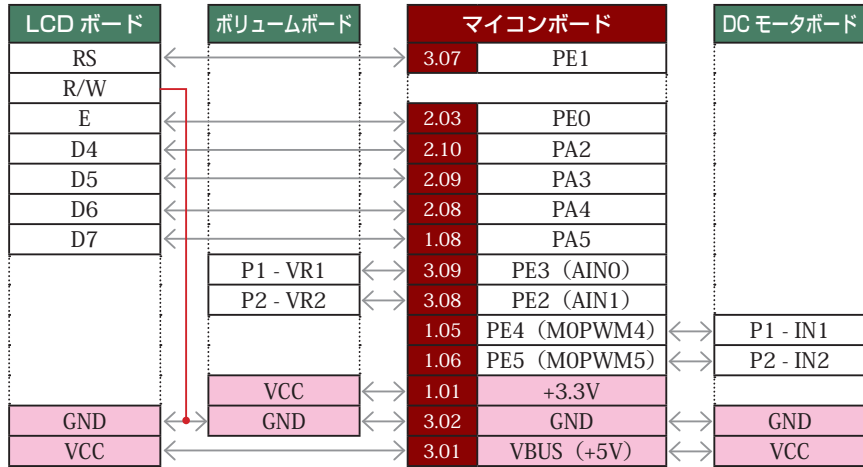
正誤表

キットで学ぶ! シリーズ No.07 ARM チャレンジャー入門編 Tiva C Series Cortex-M4 版

ページ	誤	正														
P20	<p>③ 「File system…」 をクリックします。</p>	<p>CCS6.2 以降は</p> <p>③ 「Browse…」 をクリックします。</p>														
P43	<p>最下部の注釈部分</p> <p>TivaWare に含まれる blinky.c について、 テキスト執筆時の ver. 2.1.0.12573 では、ライブラリ関数を使わないソースでしたが、 最新版の ver. 2.1.2.111 では、ライブラリ関数を使う記述に変更されています。 何卒ご了承ください。</p>															
P62	<p>タイマ構成表のタイマモジュールのビット数</p> <table border="1" data-bbox="340 846 605 1116"> <thead> <tr> <th>タイマモジュール</th> </tr> </thead> <tbody> <tr> <td>WTimer0 (16/32-Bit)</td> </tr> <tr> <td>WTimer1 (16/32-Bit)</td> </tr> <tr> <td>WTimer2 (16/32-Bit)</td> </tr> <tr> <td>WTimer3 (16/32-Bit)</td> </tr> <tr> <td>WTimer4 (16/32-Bit)</td> </tr> <tr> <td>WTimer5 (16/32-Bit)</td> </tr> </tbody> </table> <p>TM4C123GH6PM のタイマ構成</p>	タイマモジュール	WTimer0 (16/32-Bit)	WTimer1 (16/32-Bit)	WTimer2 (16/32-Bit)	WTimer3 (16/32-Bit)	WTimer4 (16/32-Bit)	WTimer5 (16/32-Bit)	<table border="1" data-bbox="875 846 1140 1116"> <thead> <tr> <th>タイマモジュール</th> </tr> </thead> <tbody> <tr> <td>WTimer0 (32/64-Bit)</td> </tr> <tr> <td>WTimer1 (32/64-Bit)</td> </tr> <tr> <td>WTimer2 (32/64-Bit)</td> </tr> <tr> <td>WTimer3 (32/64-Bit)</td> </tr> <tr> <td>WTimer4 (32/64-Bit)</td> </tr> <tr> <td>WTimer5 (32/64-Bit)</td> </tr> </tbody> </table> <p>TM4C123GH6PM のタイマ構成</p>	タイマモジュール	WTimer0 (32/64-Bit)	WTimer1 (32/64-Bit)	WTimer2 (32/64-Bit)	WTimer3 (32/64-Bit)	WTimer4 (32/64-Bit)	WTimer5 (32/64-Bit)
タイマモジュール																
WTimer0 (16/32-Bit)																
WTimer1 (16/32-Bit)																
WTimer2 (16/32-Bit)																
WTimer3 (16/32-Bit)																
WTimer4 (16/32-Bit)																
WTimer5 (16/32-Bit)																
タイマモジュール																
WTimer0 (32/64-Bit)																
WTimer1 (32/64-Bit)																
WTimer2 (32/64-Bit)																
WTimer3 (32/64-Bit)																
WTimer4 (32/64-Bit)																
WTimer5 (32/64-Bit)																
P70	<p>ADC 変換 中央値</p> 															

P94

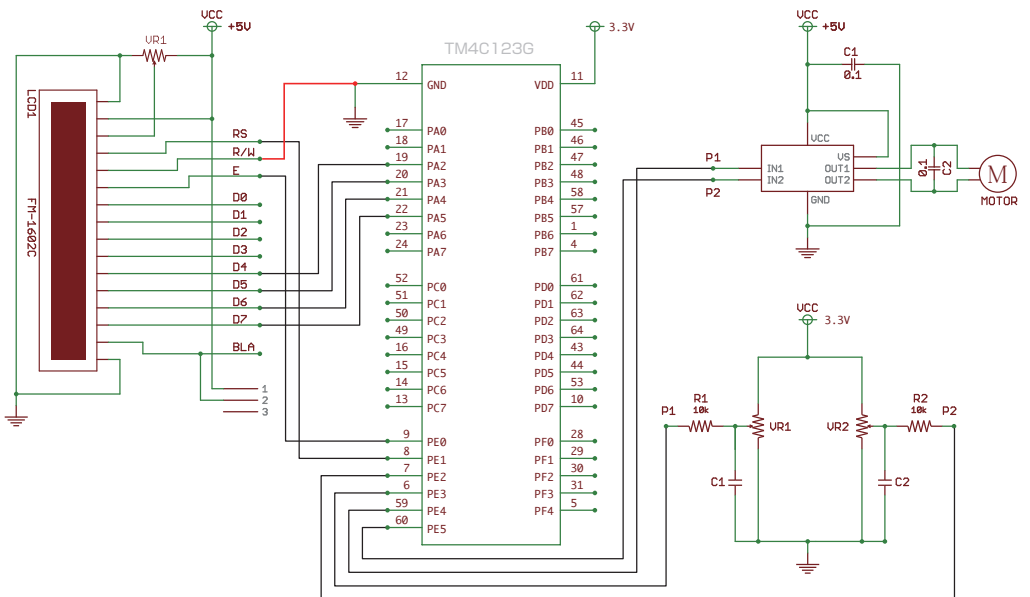
LCD ボードの R/W 端子は GND に接続されています。 下表の赤線追加。



P95

LCD ボードの R/W 端子は GND に接続されています。 下図の赤線追加。

回路図



ページ	正
<p>P98</p> <p>フローチャート 11-1</p> <p>0x30 を (ある時間間隔を空けて) 3 回設定するのは、LCD を初期化 (リセット) するためであり、本テキストで使用する LCD がそのような仕様になっています。リセット後に LCD は 8 ビット転送モードになりますが、0x30 を送る直接の目的は LCD のリセットのためになります。</p> <p>テキストにある「8 ビットモードに設定」という言葉は、やや誤解を招きやすいので、単に 0x30 を送るものごと理解ください。</p>	<pre> graph TD Start([LCD初期化 関数]) --> Init([lcd_init 処理開始]) Init --> RS[RS 信号を Low に設定 lcd_RS] RS --> Delay1[15ms 待ち DelayMS] Delay1 --> Transmit1[8 ビットモードに設定 (データビットに 0x30 を送る) lcd_transmit8] Transmit1 --> Delay2[5ms 待ち DelayMS] Delay2 --> Transmit2[再度 8 ビットモードに設定 (データビットに 0x30 を送る) lcd_transmit8] Transmit2 --> Delay3[1ms 待ち DelayMS] Delay3 --> Transmit3[再々度 8 ビットモードに設定 (データビットに 0x30 を送る) lcd_transmit8] Transmit3 --> Delay4[5ms 待ち DelayMS] Delay4 --> End[] </pre>
<p>P140</p>	<pre> 58 // : 使用する ADC <u>シーケンサ</u>の指定 59 ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0); 60 // : ADC <u>シーケンサ</u>の設定 61 ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 ADC_CTL_IE ADC_CTL_END); 62 // : ADC <u>シーケンサ</u>を有効にする 63 ADCSequenceEnable(ADC0_BASE, 3); </pre>