LAB2

Program the on chip PWM

Outline

- Learn to setup the on-chip PWM
- Implement it in Assembly
- Test the result by Oscilloscope

Setup the on-chip PWM

- Read the LPC2103 user manual carefully.
- We will use the channel 1 of the timer 3 to control and output a PWM wave. (some channel doesn't have a output pin on ARMmite pro board)
- There are four timers in LPC2103. Each timer has several channels. Each channels has a match register. In the lab, we use **timer 3's channel 1** to create PWM.
- Match Register values are continuously compared with timer counter. When equal, a interrupt will be triggered.
- We can specify which action will be done if a interrupt happen.
- The prescale register serve as a devisor of the timer counter. If set to 0, timer counter increments at every clock cycle. If set to n, timer counter increments at 1/n rate.
- If a channel is set to **PWM mode**, when the channel's match matches the timer counter, the **PWM output goes HIGH**.
- At the **beginning of each PWM cycle** (when Timer counter is 0), PWM output goes **LOW**.

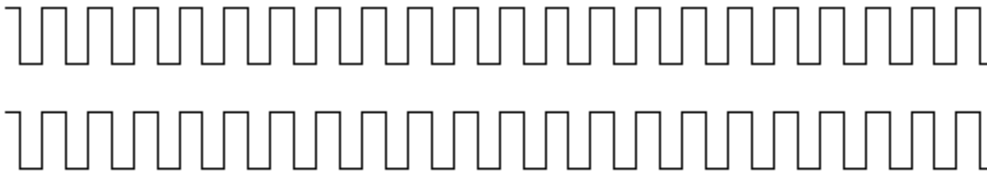
Setup the on-chip PWM (Cont.)

- 1) Setup Pin Function Select Register (PINSEL0, PINSEL1)
  - Select P0.0 as output of MAT3.1 (**Timer 3's channel 1**)
  - Leave other pins as GPIO
- 2) Reset and enable the timer 3 Counter Register (T3TCR)
  - Set bit 0 for counter enable
  - Set bit 1 for counter reset
- 3) Reset the Interrupt Register of timer 3 (T3IR)
  - Set bit 1 to reset the interrupt generated by Match Register 1 (MR1)
  - Interrupt Register can be written to clear interrupts, and read to check pending interrupt.
- 4) Set timer 3's Prescale Register (T3PR)
  - Set T3PR to 0, so timer 3 counter increments at every clock cycle.
  - If you set T3PR to n, the timer 3 counter increments at 1/n rate.

Setup the on-chip PWM (Cont.)

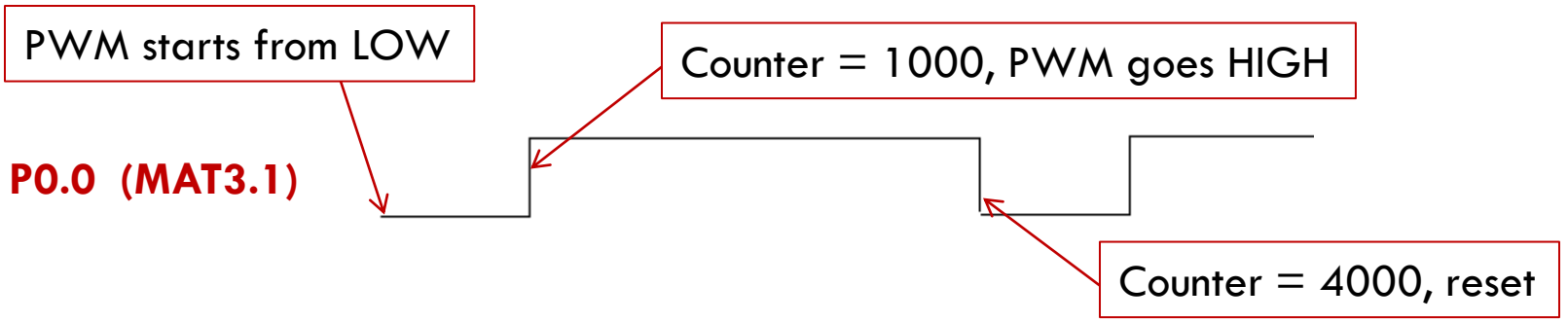
- 5) Setup PWM Control Register of timer 3 (T3PWMCON or PWM3CON)
  - Set bit 1 to **enable PWM for timer 3 channel 1**.
- 6) Specify action by Setting Match Control Register of timer 3 (T3MCR)
  - Set bit 10: the **timer 3 counter will be** reset at **interrupt triggered by match register 3**.
- 7) Set Match Register 1 of timer 3 (T3MR1)
  - Set T3MR1 to 1000, when it matches timer 3 counter, PWM output goes HIGH.
  - Remember channel 1 is set to PWM mode, and MR1 controls its output.
- 8) Set Match Register 3 of timer 3 (T3MR3)
  - Set T3MR3 to 4000, so timer interrupt will be triggered every 4000 cycles.
  - Remember we choose to reset the timer counter when interrupt triggered by MR3.
- 9) Clear the timer reset bit and thus generate PWM wave
  - Set bit 1 T3TCR to 0. (clear timer counter reset bit)

PWM Wave with 25% Duty Cycle



T3MR1 = 1000 & T3MR3 = 4000
Timer 3 channel 1 is in PWM mode &
Timer 3 channel 3 is not in PWM mode

When MR1 matches timer counter, PWM output goes HIGH. When MR3 matches, counter will be reset to 0, starting a new PWM cycle.



Implement by Assembly

```
            .text
            .equ PINSEL0, 0xE002C000        /* Check LPC2103 user manual, */
            .equ PINSEL1, 0xE002C004        /* and lpc2103_k9supd.h to find */
            .equ T3TCR, 0xE0074004          /* memory mapped I/O locations. */
            .equ T3IR, 0xE0074000           /* Define them as register name, */
            .equ T3PR, 0xE007400C           /* so the code could be more readable. */
            .equ T3PWMCON, 0xE0074074
            .equ T3MCR, 0xE0074014
            .equ T3MR1, 0xE007401C
            .equ T3MR3, 0xE0074024
            .equ T3TCR, 0xE0074004
_start:     .global _start
            ……
            ……
            ……
main:       ……                          /* Your application code starts from here */
```

Implement by Assembly

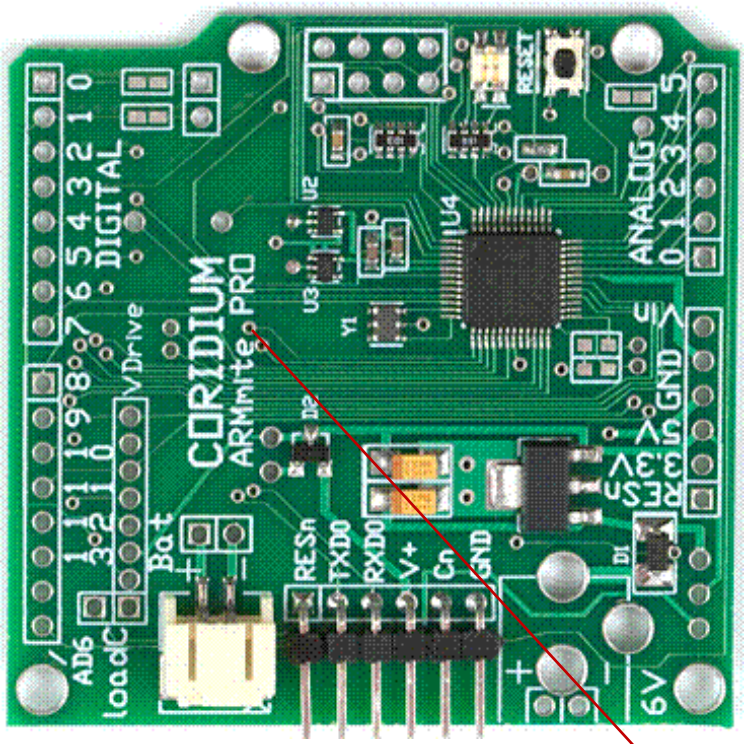
```
main:
                ldr     r2, = PINSEL0       /* 1) Setup Pin function select registers */
PINSEL0 = 0x02  mov     r3, #0x00000002
                str     r3, [r2, #0]

                ldr     r2, = PINSEL1
PINSEL1 = 0x00  mov     r3, #0x00000000
                str     r3, [r2, #0]

                ldr     r2, = T3TCR         /* 2) Reset and enable timer 3 counter */
T3TCR = 0x03    mov     r3, #0x00000003
                str     r3, [r2, #0]

                ldr     r2, = T3IR          /* 3) Reset the interrupt register of timer 3 */
T3IR = 0x02     mov     r3, #0x00000002
                str     r3, [r2, #0]

                dr      r2, = T3PR          /* 4) Set timer 3's prescale register */
T3PR = 0x00     mov     r3, #0x00000000
                str     r3, [r2, #0]
```

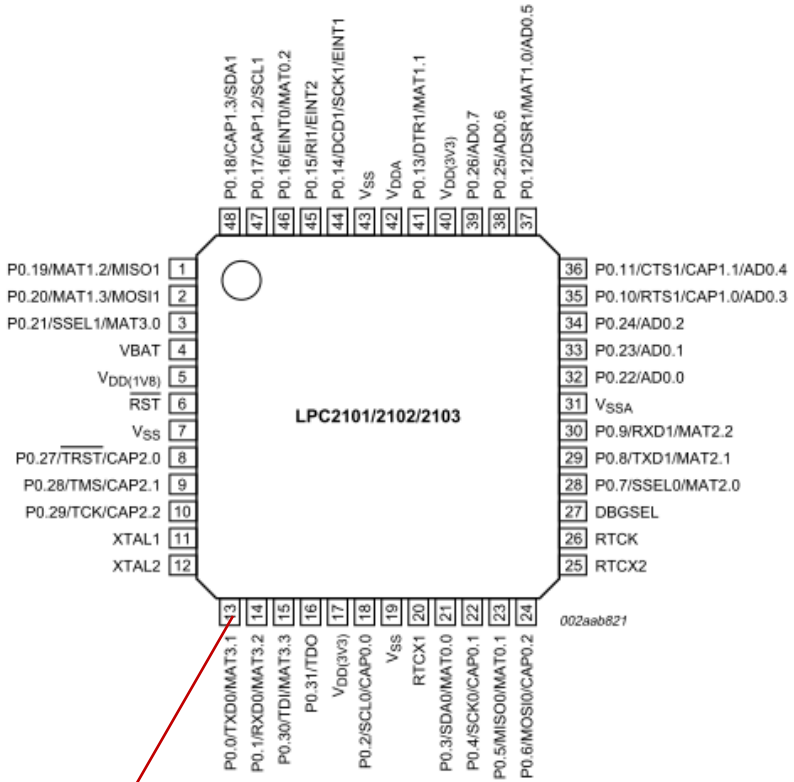
Implement by Assembly (Cont.)

```
                ldr     r2, = T3PWMCON      /* 5) Setup PWM Control Register of timer 3 */
T3PWMCON = 0x02 mov     r3, #0x00000002
                str     r3, [r2, #0]

                ldr     r2, = T3MCR         /* 6) Specify action by Setting */
T3MCR = 0x400   ldr     r3, = 0x00000400    /* Match Control Register of timer 3 */
                str     r3, [r2, #0]

                ldr     r2, = T3MR1         /* 7)Set match register 1 of timer 3 */
T3MR1 = 1000    ldr     r3, = 1000
                str     r3, [r2, #0]

                ldr     r2, = T3MR3         /* 8) Set match register 3 of timer 3 */
T3MR3 = 4000    ldr     r3, = 4000
                str     r3, [r2, #0]

                ldr     r2, = T3TCR         /* 9) Clear the timer reset bit, */
T3TCR = 0x01    mov     r3, #0x01           /* and thus generate PWM wave */
                str     r3, [r2, #0]
```

Test the Result by Oscilloscope

- Now we are going to run the program, and use Oscilloscope to see if the output is really a PWM wave with 25% duty cycle.
- Look at the tiny golden line on the board, you can find the output of P0.0

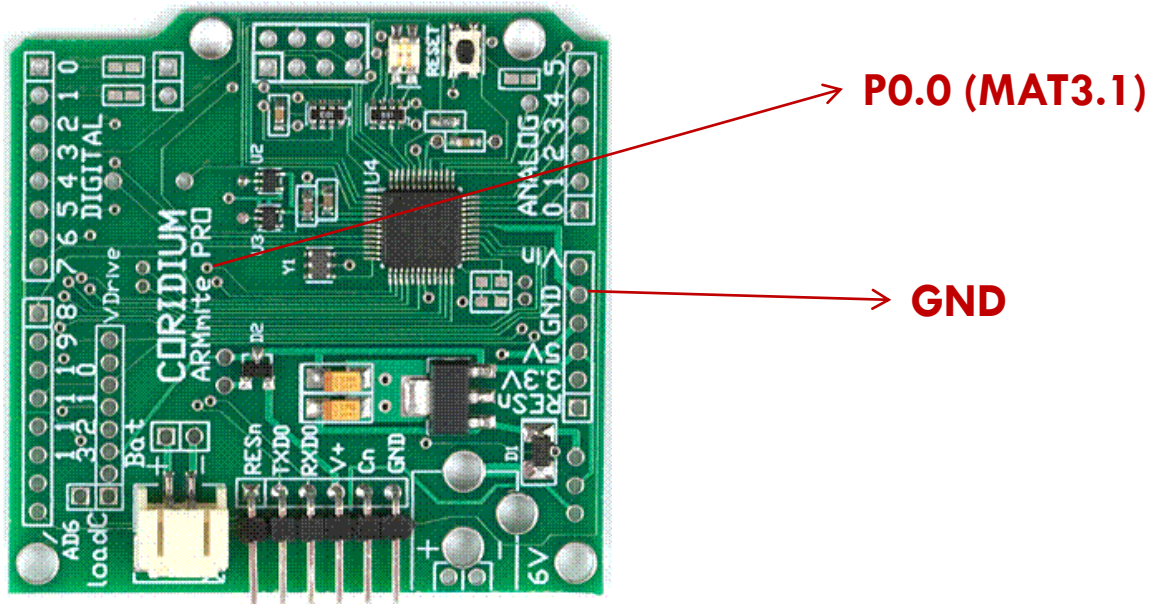




P0.0 (MAT3.1)

Test the Result by Oscilloscope

- Attach the board to your laptop, then run your program by MakeItC.
- Connect the ground of Oscilloscope to GND of the board.
- Probe the P0.0, and you should see a PWM wave with 25% duty cycle.
- No need additional power supply.



Appendix – C program

```
#include <arch\philips\lpc2103_k9supd.h>
#include "coridium.h“
int main(void)
{
        PINSEL0=0x02;           /* 1) Setup Pin function select registers */
        PINSEL1=0x00;
        T3TCR=0x03;             /* 2) Reset and enable timer 3 counter */
        T3IR=0x02;              /* 3) Reset the interrupt register of timer 3 */
        T3PR=0x00;              /* 4) Set timer 3's prescale register */
        T3PWMCON=0x02;          /* 5) Setup PWM Control Register of timer 3 */
        T3MCR=(1<<10);          /* 6) Specify action by Setting Match Control Register */
        T3MR1=1000;             /* 7)Set match register 1 of timer 3 */
        T3MR3=4000;             /* 8) Set match register 3 of timer 3 */
        T3TCR=0x01;             /* 9) Clear the timer reset bit, generating PWM wave*/
        while(1);
        return(0);
}
```