Lab 4: Timer in Periodic & PWM Modes

… as well as setting up a ADC Channel

Lab 4: Timers

Part 1: **Periodic Mode & ADC Channel** – Generate a waveform with frequency dependent on an analog voltage input.

- Verify that system clock speed is 80 MHz.
- Program AIN8 as an ADC channel input to the LaunchPad. *[see slides 3 -8]*
- AIN8 is connected to the potentiometer. It reads the voltage output from the potentiometer, which can vary from 0V to 3.3V.
- Program Timer 0A to be in 16-bit, Periodic, Count-Down mode. Enable Time-Out interrupts.
- Upon each Time-Out interrupt of Timer 0A, toggle the output of **GPIO PB6**. Thus PB6 would output a square wave with 50% duty-cycle. PB6 should have a default logic level of '0'.
- Vary the Time-Out interval based on the value of AIN8 reading. That is, as the potentiometer turns, the frequency of PB6 output should change accordingly. Scale the output frequency to the range shown in the table. Assume a linear correlation. Verify the output waveform frequency shifts using the Analog Discovery 2.
- Pressing SW2 should disable Timer 0A and there should be no output waveform at PB6. (TAEN bit).
- Pressing SW1 should resume waveform output.

AIN8 Voltage	PB6 Output Frequency
0V	10 – 15 kHz
3.3V	75 – 85 kHz

Part 2: **PWM Mode** – Generate a PWM waveform to vary LED intensity.

- The Blue LED is connected to **GPIO PF2** which is also the Timer 1A output (T1CCP0) when set to its Alternate Function.
- Set T1CCP0 to output a PWM waveform. Vary the duty cycle form 10% to 90% such that the intensity of the Blue LED cycles from low (10%) to high (90%) intensity and then high intensity to low intensity. Repeat the cycle forever. Check the changes to duty cycle using the Analog Discovery 2.
- Select a waveform frequency such that the LED changes intensity in a gradual manner.

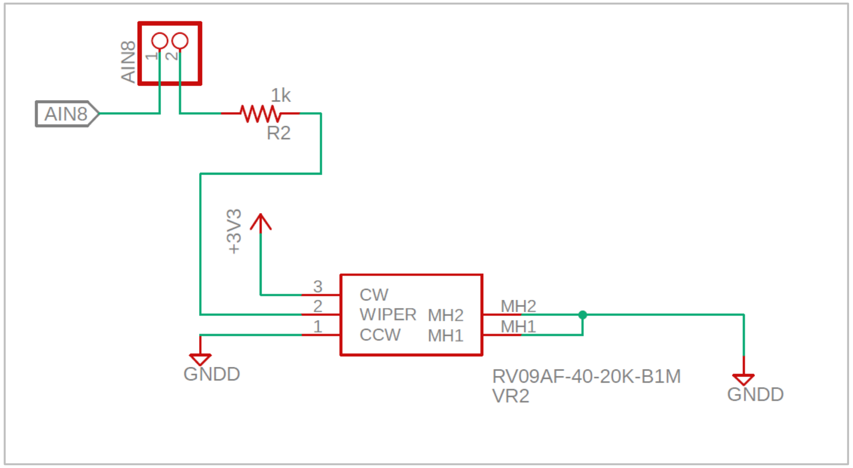
ADC: Setting up an ADC Channel

As part of Lab 4 exercise, we setup one ADC channel to the Tiva LaunchPad.

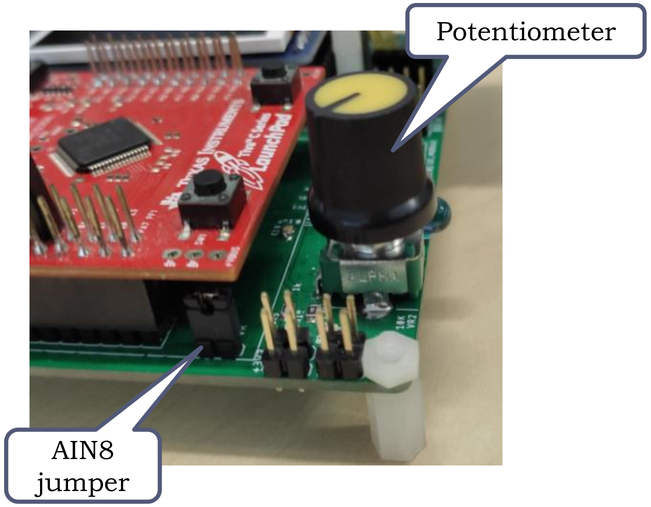
The LaunchPad has a total of 12 possible ADC channels – AIN0 to AIN11.

All the channels are multiplexed with GPIO pins.

We select ADC function through setting the appropriate AMSEL register bits.



- Potentiometer on base board is connected to **AIN8** which is mapped to GPIO PE5.
  - Note: Jumper AIN8 on the base board needs to be connected for AIN8 signal to be through.
- Potentiometer output ranges from voltage 0V to 3.3V. This output is fed into AIN8 as an analog voltage.
- 12-bit ADC of the LaunchPad converts analog values to digital in the range 0x0000 to 0xFFFF (0 to 4095), where 0x0000 corresponds to 0V & 0xFFFF to 3.3V.



Potentiometer

AIN8 jumper

Setting up ADC Channel (AIN8)

```
/*****  file: Hal.h     ******/
#define PE_ADC_AIN8   5U

/* ADC  */
#define ADC0_SS0_Start() ((ADC0->PSSI |= ADC_PSSI_SS0) )
#define ADC0_BUSY() ((ADC0->ACTSS & ADC_ACTSS_BUSY)>>16 == 0x01? TRUE : FALSE)
#define ADC0_GET_FIFO() (ADC0->SSFIFO0)
```

```
/*****  file: Hal.c     ******/
void Port_Init( void )
{
   SYSCTL->RCGCGPIO |= SYSCTL_RCGCGPIO_R4| /* enable Port E clock */
   while( 0 == (SYSCTL->PRGPIO & SYSCTL_PRGPIO_R4)){};     /* port E */

   SYSCTL->RCGCADC |= SYSCTL_RCGCADC_R0; /* Enable Clock for ADC0 */
   while( 0 == (SYSCTL->PRADC & SYSCTL_PRADC_R0) ); /* Wait for clock to be ready */

   /* enable analog function for AIN8  */
   GPIOE->DEN &= ~BIT(PE_ADC_AIN8);
   GPIOE->AMSEL |= BIT(PE_ADC_AIN8);
```

Setting up ADC Channel (AIN8)

```
/*****  file: main.c     ******/
#define ADC_UPDATE_MS 20U // update ADC every 20 ms
volatile int      g_nAdcUpdate = ADC_UPDATE_MS;
volatile BOOL     g_bAdcUpdate = FALSE;
volatile uint16_t AIN8_Result;

void main_AdcInit( void );

int main()
{
   main_AdcInit();
   for(;;)
   {
      /* ADC Update */
      if(g_bAdcUpdate != FALSE && ADC0_BUSY() == 0)
      {
          g_bAdcUpdate = FALSE;
          AIN8_Result = ADC0_GET_FIFO(); /* Read measurements */
          ADC0_SS0_Start(); /* Start next measurement */
       }
    }
}
```

Setting up ADC Channel (AIN8)

```
/*****  file: main.c     ******/

/* Set SS0 for AIN8 */
void main_AdcInit( )
{
   ADC0->ACTSS &= ~ADC_ACTSS_ASEN0; /* Disable ADC */
   ADC0->EMUX |= ADC_EMUX_EM0_PROCESSOR; /* Set Processer as trigger */
   ADC0->SSMUX0 |= (0x08 << ADC_SSMUX0_MUX0_S); /* Connect AIN8 to SS0 */
   ADC0->SSCTL0 |= ADC_SSCTL0_END0; /* 1st sample is End of Seq */
   ADC0->SAC |= ADC_SAC_AVG_32X; /* Set hardware averaging to 32 times */
   ADC0->ACTSS |= ADC_ACTSS_ASEN0; /* Enable ADC */
   ADC0_SS0_Start();  /* Start first measurement */
}
```

Setting up ADC Channel (AIN8)

```
/*****  file: main.c     ******/

void SysTick_Handler( void )
{
   if( 0 != g_nAdcUpdate )
   {
      g_nAdcUpdate--;
      if( 0 == g_nAdcUpdate )
      {
          g_bAdcUpdate = TRUE;
          g_nAdcUpdate = ADC_UPDATE_MS;
      }
   }
```

Setting up ADC Channel (AIN8)

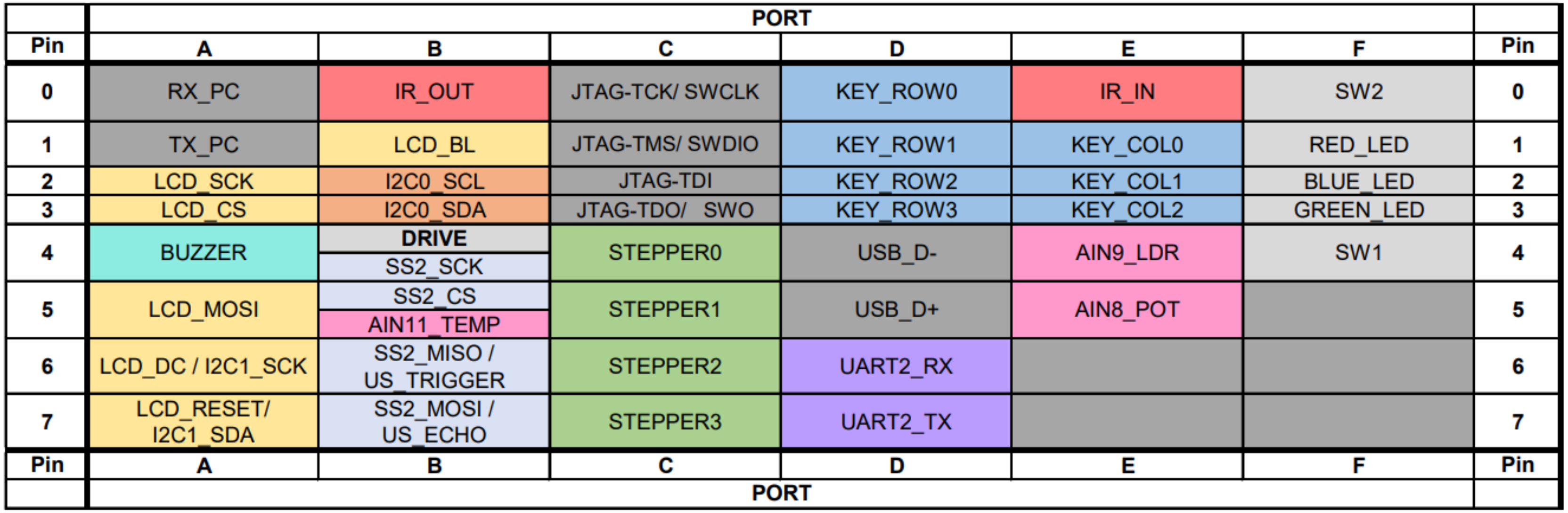
```
/*****  file: main.c     ******/

void SysTick_Handler( void )
{
   if( 0 != g_nAdcUpdate )
   {
      g_nAdcUpdate--;
      if( 0 == g_nAdcUpdate )
      {
          g_bAdcUpdate = TRUE;
          g_nAdcUpdate = ADC_UPDATE_MS;
      }
   }
```

- ADC value read from the potentiometer (AIN8) is stored as a 12-bit number in variable `AIN8_Result`. It can range from 0 to 4095.
- In KEIL μVison, add `AIN8_Result` to the Debug Watch window to verify that it is reflecting the correct result as you turn the potentiometer.

Lab 4 Submission

- Submit the following:
  - Zip file of your µVision KEIL program directory.
- Submission Dateline:
  - Sunday, 12 Jun 22, 23:59 hrs.
  - Upload Zipped folder to Moodle

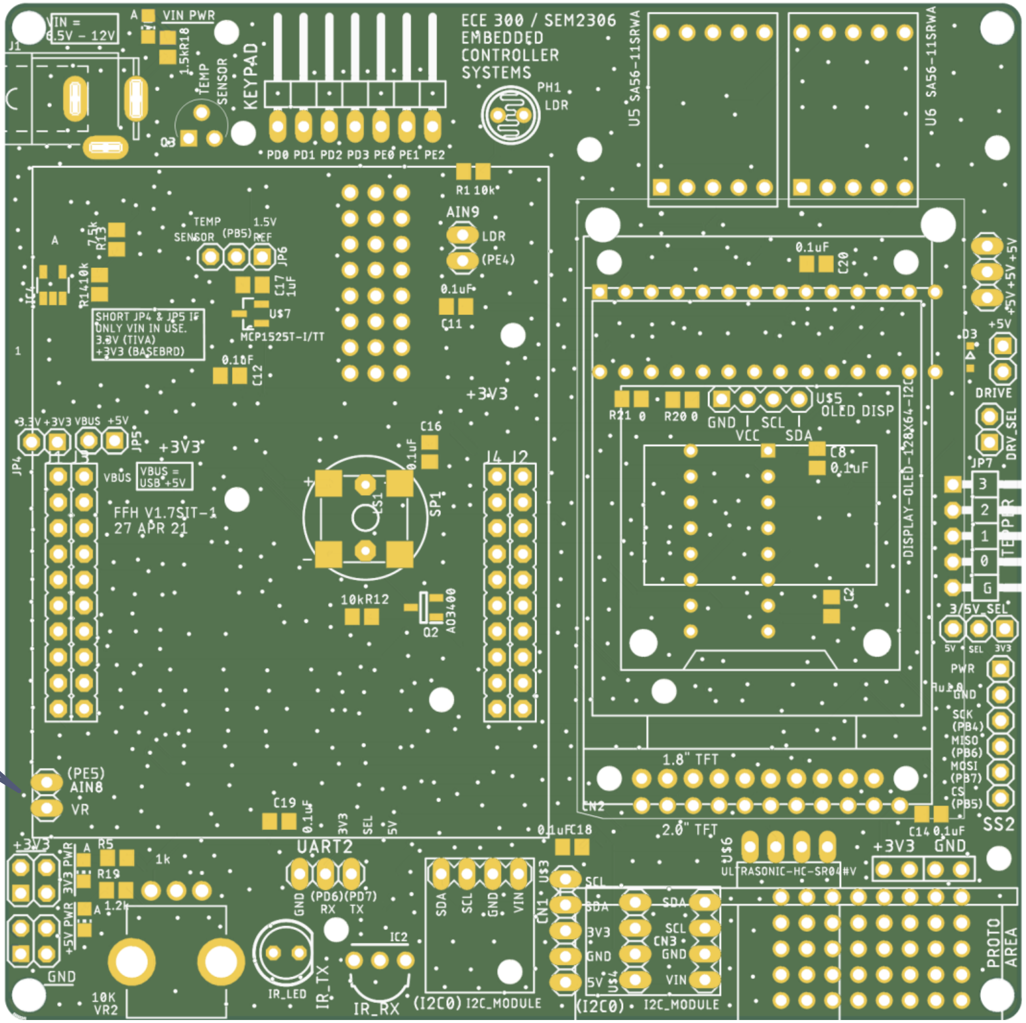
Port Assignment (Tiva Base Board)



	PORT						
Pin	A	B	C	D	E	F	Pin
0	RX_PC	IR_OUT	JTAG-TCK/ SWCLK	KEY_ROW0	IR_IN	SW2	0
1	TX_PC	LCD_BL	JTAG-TMS/ SWDIO	KEY_ROW1	KEY_COL0	RED_LED	1
2	LCD_SCK	I2C0_SCL	JTAG-TDI	KEY_ROW2	KEY_COL1	BLUE_LED	2
3	LCD_CS	I2C0_SDA	JTAG-TDO/ SWO	KEY_ROW3	KEY_COL2	GREEN_LED	3
4	BUZZER	**DRIVE** SS2_SCK	STEPPER0	USB_D-	AIN9_LDR	SW1	4
5	LCD_MOSI	SS2_CS AIN11_TEMP	STEPPER1	USB_D+	AIN8_POT		5
6	LCD_DC / I2C1_SCK	SS2_MISO / US_TRIGGER	STEPPER2	UART2_RX			6
7	LCD_RESET/ I2C1_SDA	SS2_MOSI / US_ECHO	STEPPER3	UART2_TX			7
Pin	A	B	C	D	E	F	Pin
	PORT						

Keep this for reference.

Tiva Base Board Layout - Top

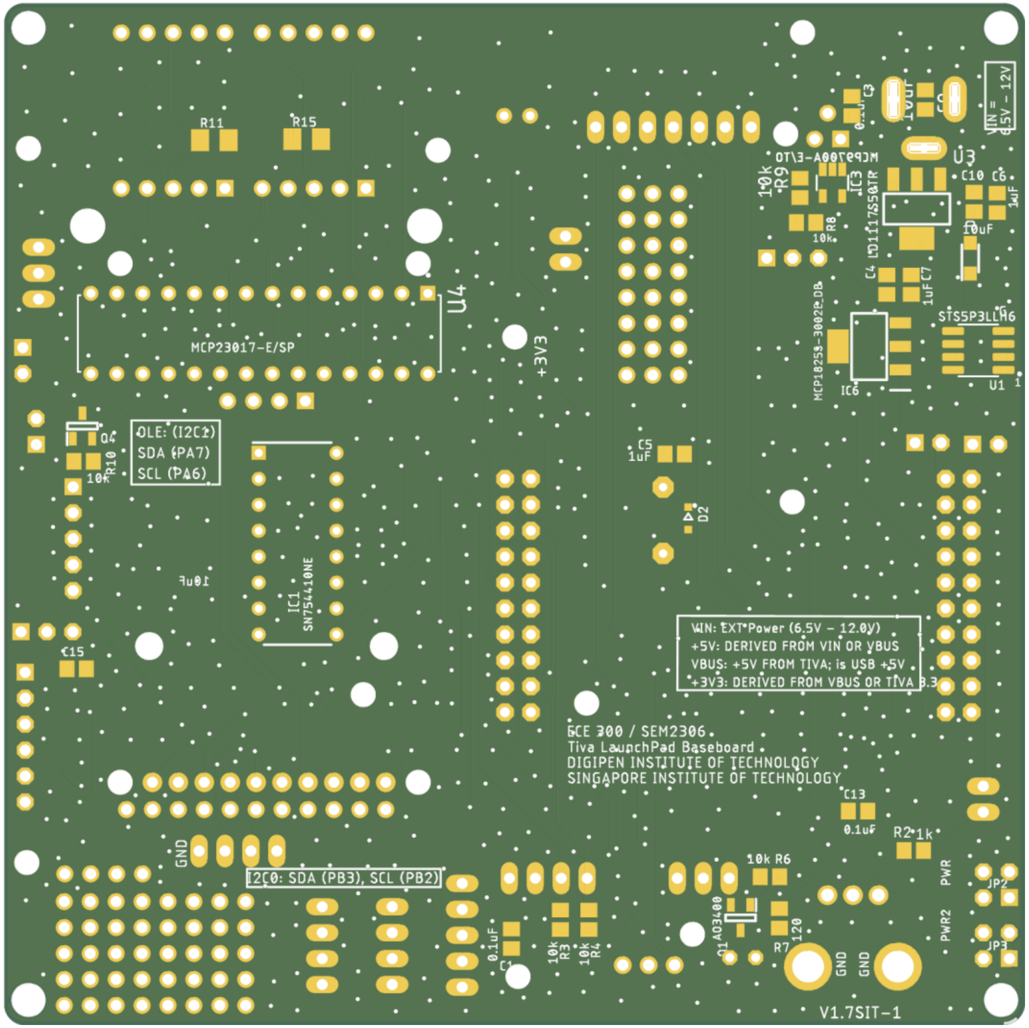


AIN8 jumper

TOP layout of base board.

Keep this for reference.

Tiva Base Board Layout - Bottom



Bottom layout of base board.

Keep this for reference.