CMPE 443 PRINCIPLES OF EMBEDDED SYSTEMS DESIGN

LAB #005

“PLL Configuration, Interrupts and Timers”

Motivation

In this experiment, we study on changing the operating frequency of the microcontroller, interrupts and timers. In addition, ultrasonic sensors will be introduced. Ultrasonic sensors are used for distance measurement, i.e. it gives an idea about how close the objects are. They have a wide range of application in embedded systems as the self-parking mechanism of the cars, automatic sliding doors, automatic gates. So, in this experiment you will learn:

- configuring PLL for changing the increasing clock frequency for CPU
- the process of enabling and setting priority of interrupts
- calculating time from timer counter and clock frequency.
- programming output compare timer function to generate a pulse.
- programming input capture function to determine duration of a input pulse.
- determining the distance between an object and the ultrasonic sensor.

1) Problem Description

In this experiment, you will firstly change the CPU Clock frequency to 60 MHz and after the observation of this clock change, you will work with ultrasonic sensor. By using the ultrasonic sensor, you will detect obstacles and its distance to ultrasonic sensor. According to the distance value, you will turn on or off the LEDs which are on the LPC4088 board.

Note: When a question ends with ***(*)*** *notation, that means write on the code. When a question ends with* ***(?)*** *notation, that means write on the paper. When you see* ***(*?)****, the answer of this question should be written on the paper and code.*

2) Select the correct PLL (Phase Locked Loop) *0.5 pts*

Answer the following questions by using the PLL section of the LPC4088 user manual or the PLL slides:

- Which PLL is used for Peripheral Clock (PCLK) and CPU clock (CCLK)? ___________ *0,25 pts*
- What is the default oscillator for this PLL? ___________ *0,25 pts*

3) PLL Registers *2 pts*

PLL provides a faster clock than the source clock. In this experiment, you will change the CPU Clock Frequency. In order to setup PLL in LPC4088. You need to use PLLCFG, PLLCON, PLLFEED and PLLSTAT registers.

What is the address of these registers for the PLL you found. (Write the address in **PLL.h**)

- PLLCON **(*?)** ___________ *0,5 pts*
- PLLCFG **(*?)** ___________ *0,5 pts*
- PLLSTAT **(*?)** ___________ *0,5 pts*
- PLLFEED **(*?)** ___________ *0,5 pts*

4) PLL Calculations *1 pt*

For the PLL Input Clock, consider oscillator clock (*OSC_CLK*) on the LPC4088 Quick Start Board:

- What is the frequency of the *OSC_CLK* (Hint: Search for the crystal which is connected between X1 and X2 pins)? **(?)** ___________ *0,25 pts*
- What should be the value of the CLKSRCSEL to make the clock source for the PLL as the *OSC_CLK*? **(?)** ___________ *0,25 pts*
- For achieving a frequency around 120 MHz, what is PLLCFG value? **(?)** ___________ *0,25 pts*

- For achieving a frequency around 60 MHz, what is PLLCFG value? **(?)**

____ *0,25 pts*

5) PLL Change on LPC4088 *4 pts*

Firstly observe the code by loading it to board and the write the frequency of the blinking action of Red LED **(?)**

_____ *0,5 pts*

In the PLL.c file, write the codes for:

- Disable PLL **(*?)**

  _____ *0,5 pts*
- Feed PLL **(*?)**

  _____ *0,5 pts*
- Select the clock source as the OSC_CLK, (i.e. Write to CLKSRCSEL) **(*?)**

  ___________ *0,25 pts*
- Write to PLLCFG for 60MHz **(*?)**

  _____ *0,25 pts*
- Enable PLL **(*?)**

  _____ *0,5 pts*
- Check PLL and wait for the PLL to lock **(*?)**

  _____ *0,25 pts*
- Select the CPU clock as the same clock rate as Main PLL, (i.e. Write to CCLKSEL) **(*?)**

  ___________ *0,25 pts*
- Configure PCLK as PCLKDIV = 0x04. Therefore what is PCLK frequency? **(*?)**

  ___________ *0.5 pts*

Write the new frequency of the blinking action of Red LED **(?)**

_____ *0.5 pts*

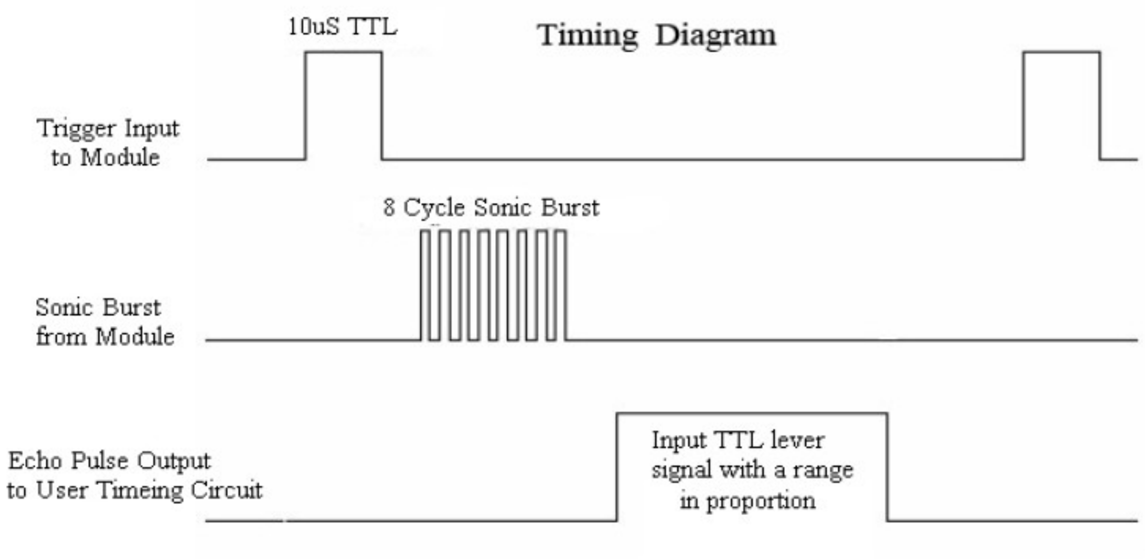
*Note: Before testing uncomment PLL_Change_Configuration function. After testing PLL, comment (//) the **PLL_Test** method which is in the **update** method.*

6) Ultrasonic Sensor *1 pt*

HC - SR04 Ultrasonic sensor (US) has 4 pins which are VCC, Trigger, Echo and GND. User manual of the HC - SR04 is given in the Files folder in Canvas.

In order to use HC - SR04 Ultrasonic sensor, you should give a trigger signal which is a square signal. After that ultrasonic will send 8 cycle burst of ultrasound at 40KHz and it will give **HIGH** value to its Echo pin. The Echo is a distance object that is pulse width and the range in proportion which means that:

- If obstacle goes away, pulse width on Echo will increase.
- If obstacle approaches, pulse width on Echo will decrease.

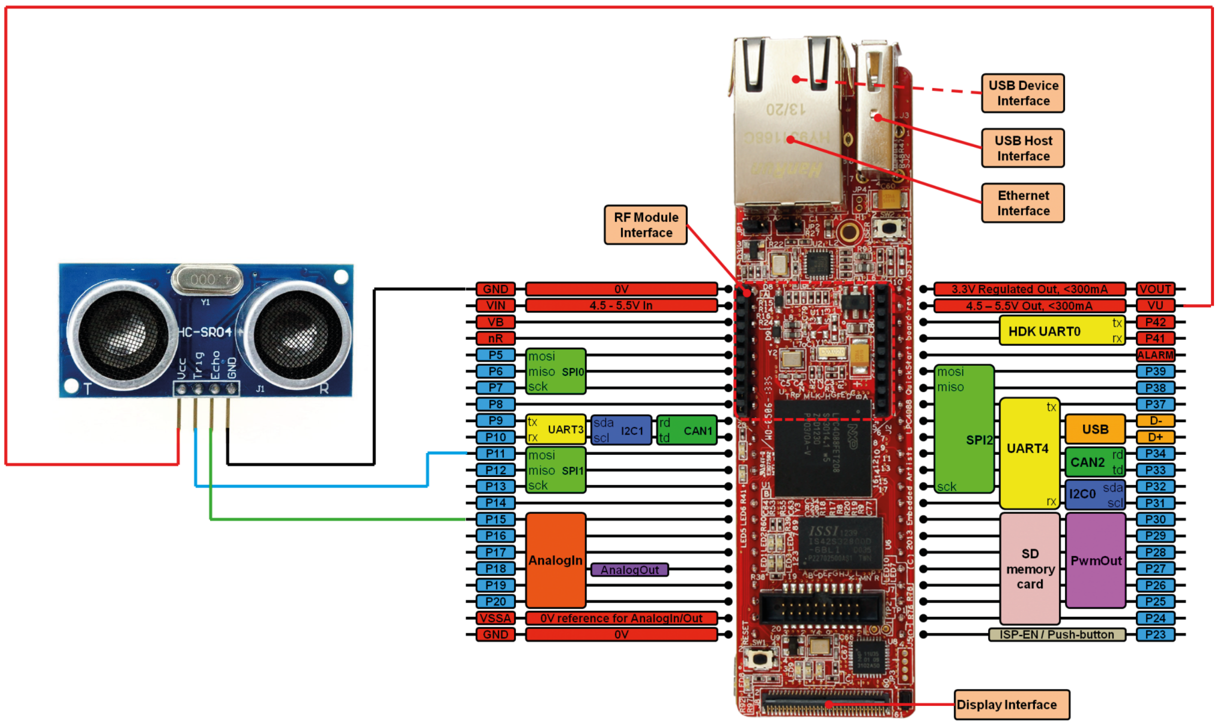

10uS TTL
Timing Diagram
Trigger Input to Module
8 Cycle Sonic Burst
Sonic Burst from Module
Echo Pulse Output to User Timeing Circuit
Input TTL lever signal with a range in proportion

- What is the min range of HC - SR04? **(?)** ____________ *0.25 pts*
- What is the max range of HC - SR04? **(?)** ____________ *0.25 pts*
- What is the min HIGH pulse width for triggering the HC - SR04? **(?)**
  ____________ *0.25 pts*
- What is the min suggested measurement cycle length for HC - SR04? **(?)**
  ____________ *0.25 pts*

7) Connecting Ultrasonic Sensor to LPC4088

Connect the HC - SR04 pins to LPC4088:

HC - SR04 Pins	LPC4088 Pins
Vcc	Vin
GND	GND
Trig	P11 (P0_9)
Echo	P15 (P0_23)



8) Initialize Ultrasonic Sensor *1.5 pts*

In order to initialize pins which are connected to Ultrasonic Sensor, you find the IOCON register addresses (**Ultrasonic.h**):

- IOCON register address of Trigger Pin **(*?)**

  ______ ______ *0.25 pts*

- IOCON register address of Echo Pin **(*?)**

  ______ ______ *0.25 pts*

Trigger Pin will be used as T2_MAT_3 and Echo Pin will be used as T3_CAP_0:

- For Trigger Pin, what should be value of FUNC field in IOCON register? **(?)** ____________ *0.25 pts*
- For Echo Pin, what should be value of FUNC field in IOCON register? **(?)** ____________ *0.25 pts*
- Write the necessary code for Trigger and Echo functionality into **Ultrasonic_Trigger_Timer_Init** and **Ultrasonic_Capture_Timer_Init** methods which are in **Ultrasonic.c** file. **(*?)** ____________ *0.5 pts*

9) Initialize Trigger Timer *4 pts*

In LPC4088, there are 4 Timer peripherals which are Timer0, Timer1, Timer2 and Timer3.

In order to use Timer, Timer should be turned on. In order to turn on any peripherals, Power Control for Peripherals register (PCONP) is used. When a peripheral control bit is 1 in PCONP, that peripheral is enabled.

- In **Ultrasonic_Trigger_Timer_Init (Ultrasonic.c)**, power up Timer2 **(*?)**
  ___________ *0.5 pts*

Prescale Counter register which is PC register of Timer is used for incrementing the value of the Timer Counter (TC). The Prescale Counter is incremented on every PCLK. When it reaches the value stored in the Prescale Register (PR), the Timer Counter is incremented and the Prescale Counter is reset on the next PCLK. This causes the Timer Counter to increment on every PCLK when PR = 0, every 2 PCLK when PR = 1, etc. (every PR+1 cycles of PCLK).

Timer Counters (TC) of Timer2 and Timer3 should be incremented at every 1 microsecond (1 second = 1000000 microsecond).

- What is the required number of PCLK cycles for 1 microsecond? **(?)**
  ___________ *0.5 pts*
- Change Timer2 PR Register value for 1 microsecond incrementing **(*?)**
  ___________ *0.5 pts*

Trigger pin will be used as External Match output for Timer2. External Match means that when a match occurs between the TC and MR, this bit can either toggle, go **LOW**, go **HIGH**, or do nothing. Trigger Pin is working as T2_MAT_3. Therefore when TC matches to MR3, it should toggled and its initial output value should be **HIGH**. In **Ultrasonic_Trigger_Timer_Init** method:

- What is the correct value of Timer2 EMR register for toggling output value of Trigger Pin when match occurs. (Initial output value of Trigger Pin is **HIGH**) **(*?)**
  ___________ *0.5 pts*

Trigger pin should give HIGH value 10 microsecond for activating the Ultrasonic sensor. In **Ulrasonic_Start_Trigger** method:

- Give correct value to MR3 Register for 10 microsecond **(*?)** ___________ *0.5 pts*

Timer2 interrupt is enabled but the conditions for getting interrupt is not set.

- Enable interrupt for **MR3** register. (**Ulrasonic_Start_Trigger**) **(*?)** ___________ *0.5 pts*

In order to use Timer2, you should remove reset on Timer2 (**Ulrasonic_Start_Trigger**):

- Remove the reset on counters of Timer2. **(*?)** ___________ *0.5 pts*
- Enable Timer2 Counter and Prescale Counter for counting. **(*?)** ___________ *0.5 pts*

10) Initialize Capture Timer *2.5 pts*

- In **Ultrasonic_Capture_Timer_Init (Ultrasonic.c)**, power up Timer3 **(*?)** ___________ *0.5 pts*
- Change Timer3 PR Register value for 1 microsecond incrementing **(*?)** ___________ *0.5 pts*

Echo pin will be used as capturing input. The Capture Control Register (CCR) is used to control whether one of the two Capture Registers (CR0 and CR1) is loaded with the value in the Timer Counter when the capture event occurs, and whether an interrupt is generated by the capture event. Echo Pin is working as T3_CAP_0 and also for calculating distance, falling edge time and rising edge time difference should be know. In **Ultrasonic_Capture_Timer_Init** method:

- Write the correct value for getting interrupt when rising edge occur **(*?)** ___________ *0.5 pts*
- Remove the reset on counters of Timer3. **(*?)** ___________ *0.5 pts*
- Enable Timer3 Counter and Prescale Counter for counting. **(*?)** ___________ *0.5 pts*

11)

Initialize Interrupts *4.5 pts*

Interrupt mechanism is used for events which need immediate attention. In our case, Ultrasonic sensor should work at the same frequency apart from main program and when distance value is changed immediately our program should know. Therefore Timer2 and Timer3 are used for sending Trigger Signal and getting the signal from Ultrasonic Sensor.

Every peripheral of the LPC4088 has a different type of external interrupt (See lecture slides for more details). For example, in Timer case, you can configure interrupts by using Match Control Register (MCR) or Capture Control Register (CCR) (Trigger Pin will use MCR for interrupt and Echo Pin will use CCR for interrupt) and you have to enable the related interrupts, clear pending interrupts and set the priority of the interrupts with NVIC functions found in CMSIS-Core API. You can access CMSIS-Core API functions in **cortex_m4.h**. Explanation is given in lecture slides.

External interrupts fall into the category of maskable interrupts. In this category, enabling individual interrupts is necessary but not sufficient for the activation of the interrupt mechanism. You have to remove the system mask. This is provided by **__enable_irq()** function of CMSIS-Core API.

After enabling interrupts, whenever an interrupt occurs, the **Handler** method of that component is called. For example, by default Timer2 interrupt handler is **TIMER2_IRQHandler** and Timer3 interrupt handler is **TIMER3_IRQHandler**. These names are the default names which are already written in **startup_LPC407x_8x.s** file. If you open that file, you can see the list of interrupt handlers.

When an interrupt occurs and the **Handler** method is called, there can be a lot of different causes. For example, **TIMER2_IRQHandler** can be called from MC0 or MC1. Therefore for understanding, which interrupt is called, we need additional register for checking it. Every component has different register for that. For example, Timer is using IR (Interrupt Register) for identify which of eight possible interrupt sources are pending.

After getting the interrupt and using it, interrupt should be cleared. This clearing mechanism also changes. For example, in order to clear Timer interrupt, **HIGH** value should be written to corresponding bit in **IR** register. If you do not clear it, that same interrupt will be called again.

Therefore for initializing the Timer Interrupts:

In **Ultrasonic_Trigger_Timer_Init** method:

- Enable TIMER2_IRQn (Interrupt Request). **(*?)** ___________ *0.25 pts*
- Set Priority as 5 to this Timer2 IRQ. **(*?)** ___________ *0.25 pts*
- Clear pending interrupts for Timer2. **(*?)** ___________ *0.25 pts*

In **TIMER2_IRQHandler** method:

- Clear pending interrupts for Timer3. **(*?)** ___________ *0.25 pts*
- Enable TIMER3 IRQ. **(*?)** ___________ *0.25 pts*

In **TIMER3_IRQHandler** method:

- Clear pending interrupts for Timer3. **(*?)** ___________ *0.25 pts*
- Disable TIMER3 IRQ. **(*?)** ___________ *0.25 pts*

Timer2 and Timer3 interrupts will call the functions **TIMER2_IRQHandler** and **TIMER3_IRQHandler** methods. You have already found *Minimum Suggested Measurement Cycle Length for HC - SR04*. Also whenever and interrupt occur, **Handler** methods should clear the Interrupt Register bit for notifying the system that interrupt is handled. Therefore in **TIMER2_IRQHandler** method:

- Change MR3 Register Value for Suggested Waiting **(*?)** ___________ *0.5 pts*
- Clear IR Register Flag for Corresponding Interrupt **(*?)** ___________ *0.5 pts*

In **TIMER3_IRQHandler** method, we can get the time of rising edge and falling edge for Echo Pin by using CR0 register and this period of time should be converted to distance value (cm).

- Store the rising time into ultrasonicSensorRisingTime variable **(*?)**
  ___________ *0.5 pts*
- Store the falling time into ultrasonicSensorFallingTime variable **(*?)**
  ___________ *0.5 pts*
- Write the formula for calculating distance (cm) by using ultrasonicSensorRisingTime and ultrasonicSensorFallingTime variables **(?)** ___________ *0.75 pts*

12) Using Ultrasonic Sensor Distance Value *4 pts*

In this part, you will write codes into **Ultrasonic_Test** method for:

- When Ultrasonic Sensor detects obstacles which are closer than 7cm, turn on all the LEDs. ___________ *0.5 pts*
- When Ultrasonic Sensor detects obstacles which are in the range of 7cm and 12cm , turn on LED1, LED2 and LED3. ___________ *0.5 pts*
- When Ultrasonic Sensor detects obstacles which are in the range of 12cm and 20cm , turn on LED1 and LED2.
- When Ultrasonic Sensor detects obstacles which are in the range of 20cm and 30cm , turn on LED1. ___________ *0.5 pts*
- When Ultrasonic Sensor detects obstacles which are far from 30 cm, turn off all the LEDs. ___________ *0.5 pts*

(Hint: You can check new data is available by checking ultrasonicSensorEdgeCount.)
(Hint: You can use LED1_On, LED1_Off etc. methods which are in the **LED.c** file)