ITU Control & Automation Eng. Dept.
KON309E Microcontroller Systems
Experiment 5: I²C Device

Aim: Using the I²C peripheral to communicate with a device and report periodic temperature samples to the PC.

For this experiment, you will need to consult the following reference documents:

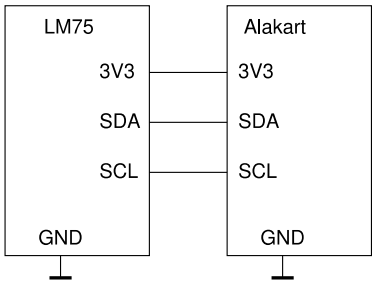
- LM75B Product Datasheet
- Class lecture note "I²C Communications to LM75 Temp Sensor LM75"
- LPC824 user manual
- LPC82x datasheet
- Alakart schematic diagram

Write the code using the peripheral support libraries (Xpressso SDK) or direct register programming or a combination of both. You are welcome to use parts of the I²C example code provided for the lectures.

The Experiment:

Construct the circuit as shown in the Figure.

- Check your wiring and make sure that **no connection is made to the 5V supply**.
- The pull up resistors for the I²C are already installed within LM75.
- Make sure that the **address select pads at the rear of the LM75 module** are all soldered to connect all of A0, A1, A2 to the GND side. This sets the **address of the LM75 module to 0x48**.



Write a program that will do the following in the main loop:

1. initializes the I²C peripheral on LPC824 of Alakart.
2. Initializes the LM75 temperature sensor and sets:
   * Configuration register (Sec 7.4.2) to:
     - OS fault queue = 4
     - OS polarity selection = Active Low
     - OS operation mode = OS comparator
     - Device operation mode = Normal
   * Tos (Overtemperature shutdown register) to 32°C
   * Thyst (Hysteresis register) to 30°C
3. Prints out a message on the terminal such as: "I²C initialization complete."

4. Starts a periodic I^2C conversion, two times a second. You can use a simple timing method such as Systick timer for this and a busy wait. At the correct time, you can call the I^2C conversion manually.

5. Converts the results to temperature in degrees C, and prints out the result in the terminal using 5 significant digits in the format "T=XX.XXX deg C". (**What does "significant digits" mean?**)

6. Repeat.

Remember that the terminal program in the PC must also be set to the **same baud rate as Alakart**. Otherwise, garbled output, or even, no output will be seen.

Heat up the LM75 sensor by placing your finger over it and confirm that as the temperature exceeds 32°C the red LED on the sensor turns on, and when the temperature drops below 30°C the red LED turns off, and this is repeatable.

Write in your report:

- What values must be written to which registers so that the I^2C peripheral is configured correctly?
- What is the value that must be written to the configutation register of the LM75 module expressed as a binary number?
- The frequencies of "main clock" and "system clock" by checking the configuration functions. Write which functions you check and how you find the result.
- Any other item of significance, such as the structure of the code and the problems that you have had during the development.

Store the code for this part in a folder named "EX5".

Bonus

Again, this is the fun part, but slightly more difficult compared to the previous ones. Write a code which sets a timer INT. It also sets a **global "flag"** variable to "false". The ISR:

- Calls the I^2C conversion only within the timer ISR,
- Formats and stores the temperature result in a **global** text string,
- Sets the global flag to "true" and exits.

The main loop:

- Checks the global flag to see if it has been set, in a busy loop.
- If the flag has been set, it will print out the temperature value on the terminal and reset the flag.

If you implement this part, describe in your report what you intended to do and

how you implemented the idea. You do not need to provide a code for this part; it will be evaluated by the class assistant.

A bonus of +5 points will be given.

Report Content

Your report must be a formal account of what you did in the experiment. Make sure that it includes the following items:

- How you have configured the processor i.e.:
  - What pins are inputs, what pins are outputs, etc.
  - Which peripheral devices were explicitly powered up,
  - How the GPIO and other peripherals were configured,
  - What values were written to which registers,
  - Which default settings of the peripherals were used.

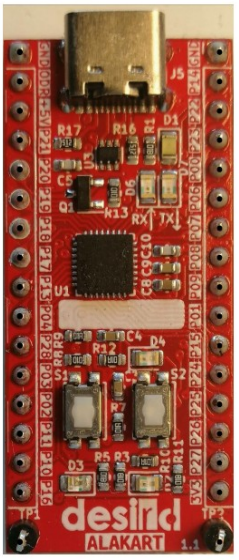
These must appear in all of your reports.

- **Any inclusion of code must be as formatted text.**

(Photographs or screenshots are not accepted!)

Appendix1: Pin connections of components on Alakart

- LEDs:
  - RED: D4 on GPIO PIN12
  - Blue: D3 on GPIO PIN16
  - Green: D2 on GPIO PIN27
  - White: D1 Power on
  - Green: D6 Transmit to PC
  - Red: D7 Transmit from PC
- Buttons:
  - S1: Reset
  - S2: ISP (enter boot mode)
    Also: User button.
- Red Pins: Test
  - TP1: GPIO PIN16
  - TP2: GPIO PIN27
- Purple Pin: Open drain FET
  - ODR: GPIO PIN21



Note:

- PIO0_10, PIO0_11 are open drain pins. Research what this means for the experiment.
- PIO0_2, and PIO0_3 are debugger pins by default. PinPIO0_5 is Reset pin by default. You can use them if you wish, but will **need to disable their default functionality in PINENABLE0 register** first.