HKU SPACE Community College
CCIT4064 Microcontrollers (MCU)
Assignment 3

Name: _____ SID: _____ CL _____

Total Mark: ____ / 40

Remark: Unless otherwise specified, all questions should assume the Intel MCS-51 instruction set for assembly language questions and the Arduino Sketch (C/C++-based language) for high-level language or Arduino-related questions.

Section A – Multiple Choice (10 Marks) (1 Mark Each)

1. Which of the following correctly enables only the Serial and Timer 1 interrupts while ensuring all other interrupts are disabled?

A) `MOV IE, #00011010B`
B) `MOV IE, #10011010B`
C) `MOV IE, #00000101B`
D) `MOV IE, #10010101B`

---

2. Which of the following statements best describes the difference between interrupt-driven I/O and polling?

A) In polling, the device sends a signal to the CPU when it needs attention; while using interrupts, the CPU repeatedly checks the device status.
B) Polling continuously checks the device status in a loop, while interrupts allow the device to signal the CPU only when service is required.
C) Polling and interrupts both require the CPU to constantly monitor device status registers.
D) Interrupts and polling both stop the CPU until the device finishes its task.

---

3. Which of the following situations is most suitable for using interrupts instead of polling?

A) Monitoring a very fast-changing signal that must be checked every clock cycle
B) Handling a keyboard input that occurs unpredictably
C) Continuously reading data from a high-speed ADC
D) Running a delay loop in embedded firmware

---

4. In an MCS-51 microcontroller, both RI (Receive Interrupt flag) and TI (Transmit Interrupt flag) become set at the same time. How is the priority between the receive and transmit operations determined?

A) RI always has higher priority than TI in hardware
B) TI always has higher priority than RI in hardware
C) The priority depends on the order of checking the flags in the interrupt service routine code
D) The interrupt priority register (IP) automatically determines which one is handled first

---

5. What is the theoretical maximum baud rate that can be generated by Timer 1 in the classic Intel 8051 when using an 11.0592 MHz crystal oscillator (Modes 1 or 3)?

A) 19200 bps
B) 38400 bps
C) 57600 bps
D) 115200 bps

6. When an 8-bit MCU performs a 32-bit multiplication using software routines, which of the following is the most significant tradeoff compared with performing the same operation on a 32-bit MCU?

A) Increased power consumption but identical execution time
B) Increased instruction count and longer execution time
C) Reduced memory usage but higher clock frequency requirement
D) Loss of arithmetic accuracy

---

7. In the Intel 8051 MCU, suppose two interrupts occur simultaneously, and both are configured with the same priority level in the IP register. Which interrupt will be serviced first?

A) Both interrupts will be ignored
B) The interrupt with the lower vector address will be serviced first
C) The interrupt with the higher vector address will be serviced first
D) The order is undetermined

---

8. For the Intel 8051 MCU, in serial communication mode 1, how many bits are transmitted per character (including framing)?

A) 8
B) 9
C) 10
D) None of the above

---

9. A student observes that ADC readings on an **Arduino UNO R4 Minima** appear inaccurate. The error becomes more severe when the board is powered via the USB-C port, whereas it is smaller when powered via the barrel jack. What is the most likely reason for this behaviour?
Hint: Students should use the Arduino UNO R4 Minima datasheet as a starting point: https://docs.arduino.cc/hardware/uno-r4-minima/

A) The ADC automatically switches to a lower resolution when the board is powered by USB-C.
B) USB-C power disables the analog reference circuitry, so the ADC uses an internal random reference voltage.
C) USB-C power on the UNO R4 Minima is reduced by a Schottky diode, while barrel input is regulated to 5 V; therefore, ADC readings can differ if the reference depends on the supply rail.
D) VIN power increases the MCU clock speed, which improves ADC accuracy.

---

10. Continue from Question A9, suppose the same analog-input circuit is migrated from an Arduino UNO R4 Minima to an Arduino UNO R3 without changing the sensor or software approach. Which **observation and tradeoff** are most likely?

A) The power-source-related ADC error would become larger, but the UNO R3 would provide higher ADC precision.
B) The ADC readings would become independent of the power source, but the UNO R3 would run at a lower clock speed.
C) The power-source-related ADC error would likely be smaller, but the tradeoff is lower ADC resolution on the UNO R3.
D) The ADC behaviour would remain the same, but the UNO R3 would only have less memory available.

A1	A2	A3	A4	A5
A6	**A7**	**A8**	**A9**	**A10**

Section B (Short Question) (Total 30 Marks) (*Answer the questions in the space provided below*)

1) Referring to the **"8051 Microcontroller Assembly Language Program for Blinking LEDs"** shown in Chapters 5 and 6 (& Supplementary Notes), the program uses nested **DJNZ** dummy loops to generate delay. In this question, you are required to redesign the program so that LED blinking is controlled more efficiently using Timer 0 and its interrupt on the standard Intel 8051 MCU. Assume a 12 MHz crystal oscillator is used throughout unless otherwise stated.

   (a) For the standard Intel 8051 MCU, answer the following about the timer operated in timer mode:
   - What is the largest timer size available for a single timer overflow?
   - Which timer mode provides this largest single-overflow delay?
   - What is the maximum count value for that mode?
   - What is the maximum time delay obtainable from one overflow in that mode when the crystal frequency is 12 MHz?

(0.5 + 0.5 + 0.5 + 1.5 Marks)

(b) Can one direct Timer 0 overflow in the mode identified in part (a) to replace the nested DJNZ-based delay directly? Briefly explain your answer.

(2 Marks)

(c) Suppose a timer interrupt period of approximately 50 ms per overflow is chosen.
   - Using Timer 0 in Mode 1, determine the initial values to be loaded into TH0 and TL0.
   - How many such overflows are required to obtain an LED ON time or OFF time close to the original 0.8 s delay?

(2 + 1 Marks)

(d) Before writing the full program, answer the following:

- What is the interrupt vector address for Timer 0?
- State the IE register setting required to enable Timer 0 interrupt.

(1 + 1 Marks)

(e) A partial Timer 0 interrupt-driven rewrite is shown below. Complete the missing instructions.

(10 Marks)

```
          _____________________________ ;First line of the program
          _____________________________ ;bypass interrupt vector table
          ;-----------------------------------
          ; INTERRUPT VECTOR ADDRESS
          ;-----------------------------------
          _____________________________ ;Timer 0 interrupt vector
          _____________________________
          ;-----------------------------------
          ; MAIN PROGRAM
          ;-----------------------------------
          ORG 0030H
MAIN:     _____________________________ ;Timer 0, Mode 1 (16-bit timer)
          MOV P1, #0FFH                 ;Initial LED pattern
          _____________________________ ;initialize software counter
          _____________________________ ;Reload value for about 50 ms
          _____________________________
          _____________________________ ;enable Timer 0 interrupt
          _____________________________ ;Start Timer 0
HERE:     SJMP HERE                     ;wait for interrupts

          ;-----------------------------------
          ; TIMER 0 ISR
          ; Every overflow ≈ 50 ms
          ; After 16 overflows, toggle P1
          ;-----------------------------------
T0_ISR:   _____________________________ ;Stop timer before reloading
          _____________________________ ;Reload Timer 0
          _____________________________
          _____________________________ ;Decrement software counter
          _____________________________ ;Toggle all bits of P1
          _____________________________ ;Reload software counter

EXIT_ISR: _____________________________ ;Restart Timer 0
          _____________________________ ;Return from interrupt

          _____________________________ Last line of the program
```

2) You are provided with a breadboard diagram showing the connections of an HC-SR04 ultrasonic sensor and four LEDs to an Arduino UNO R3.

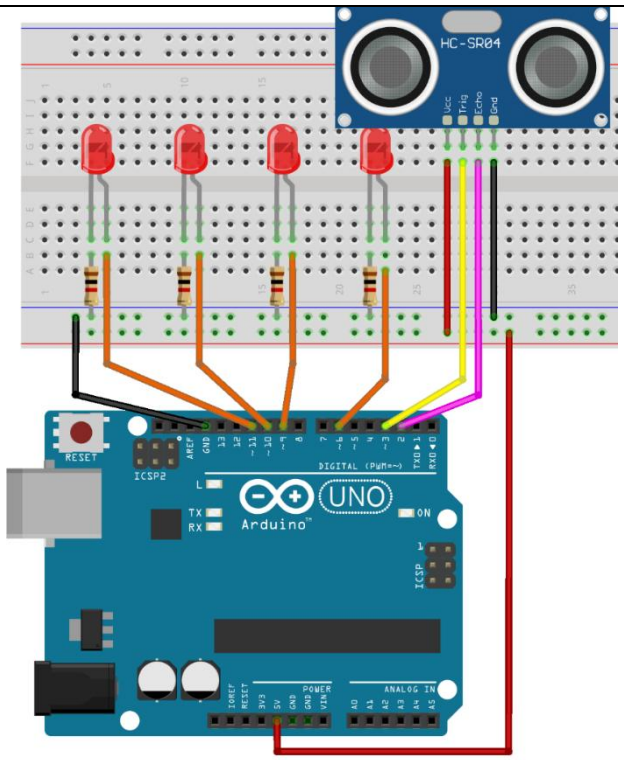
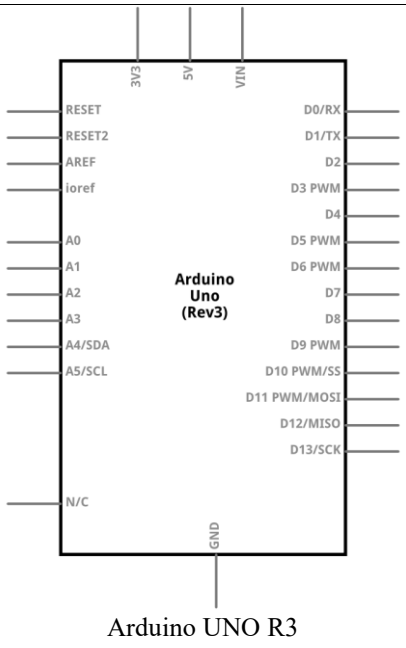


Figure: Breadboard diagram for Question 2



Arduino UNO R3

Resistors

LEDs

HC-SR04 Ultrasonic Distance Sensor

Table 1: Component List for Question 3

(a) Based on the breadboard view provided, draw the circuit schematic using the electronic symbols below.

(3M)

(b) Write an Arduino program to implement the following functionality:

- Continuously measure distance using the HC-SR04 ultrasonic sensor.
- Control the four LEDs according to the measured distance:
  - $0 \leq d < 50$ cm
    - Only LED 1 is active.
    - Its blinking rate increases with distance, from the slowest blink at 0 cm to continuously ON at 50 cm.
  - $50 \leq d < 100$ cm
    - LED 1 remains continuously ON.
    - LED 2 blinks, with blinking rate increasing with distance, from the slowest blink at 50 cm to continuously ON at 100 cm.
  - $100 \leq d < 150$ cm
    - LEDs 1 and 2 remain continuously ON.
    - LED 3 blinks, with blinking rate increasing with distance, from the slowest blink at 100 cm to continuously ON at 150 cm.
  - $150 \leq d \leq 200$ cm
    - LEDs 1 to 3 remain continuously ON.
    - LED 4 blinks, with blinking rate increasing with distance, from the slowest blink at 150 cm to continuously ON at 200 cm.
  - For distances greater than 200 cm, all four LEDs turn ON. For invalid readings, all four LEDs turn OFF.

The main program loop is provided below. Complete the missing parts of the program.

(7M)

```
// Define pin assignments




// Function to read distance from the ultrasonic sensor
long readUltrasonicDistance(                                    ) {




}
```

```
// Function to control one blinking LED
void controlBlinkingLED(                                                    ) {
// Blink interval decreases as distance increases
// Example: 0 cm => 1000 ms, 50 cm => 100 ms
// Non-blocking blinking using millis()









}

void setup() {









}
```

```
void loop() {
  long duration = readUltrasonicDistance(trigPin, echoPin);

  // If invalid reading, turn all LEDs OFF
  if (duration == 0) {
    digitalWrite(led1Pin, LOW);
    digitalWrite(led2Pin, LOW);
    digitalWrite(led3Pin, LOW);
    digitalWrite(led4Pin, LOW);
    delay(50);
    return;
  }

  // Convert duration to distance in cm
  float distance = duration * 0.0343 / 2.0;

  // Default all LEDs OFF first
  digitalWrite(led1Pin, LOW);
  digitalWrite(led2Pin, LOW);
  digitalWrite(led3Pin, LOW);
  digitalWrite(led4Pin, LOW);

  if (distance < 50) {
    // 0–50 cm: only LED1 active
    controlBlinkingLED(led1Pin, distance, 0, 50);
  }
  else if (distance < 100) {
    // 50–100 cm: LED1 always ON, LED2 blinking
    digitalWrite(led1Pin, HIGH);
    controlBlinkingLED(led2Pin, distance, 50, 100);
  }
  else if (distance < 150) {
    // 100–150 cm: LED1 and LED2 always ON, LED3 blinking
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
    controlBlinkingLED(led3Pin, distance, 100, 150);
  }
  else if (distance <= 200) {
    // 150–200 cm: LED1 to LED3 always ON, LED4 blinking
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
    digitalWrite(led3Pin, HIGH);
    controlBlinkingLED(led4Pin, distance, 150, 200);
  }
  else {
    // Above 200 cm: all LEDs ON
    digitalWrite(led1Pin, HIGH);
    digitalWrite(led2Pin, HIGH);
    digitalWrite(led3Pin, HIGH);
    digitalWrite(led4Pin, HIGH);
  }

  delay(50);
}
```

Important remarks:

- Answer all questions in the space provided.
- Submit your assessment via SOUL. **Do NOT compress your files.**
- Submission should include the following file:
  - CCIT4064_A3_**YourName**_**SID**.pdf (convert your completed answer book to PDF format)
- Penalties will apply in (but are not limited to) the following cases:
  - Assignment submitted after the grace period.
  - Assignment submitted with an incorrect file format or file name.
  - Missing or incorrect student information in the assignment.
- This is an INDIVIDUAL assessment. Plagiarism or collusion, once verified, will result in mark penalties and an unofficial record.

- End of Assignment 3 -