EEET2505

Assessment 3 (40 marks)

# 1.Question 1 (5 marks)

## 1.1 Requirements

Write a program running on an Arduino board to control as follows:

* After reset, the GPIO PORTB5 will blink with the frequency of **500Hz** (1 ms ON, 1ms OFF). You must use **Timer 0 CTC Interrupt** to generate this 500Hz clock frequency
* A button is connecting to PORTD2 – INT0 with a pull-up configuration.
  + If you press the button once, the blinking frequency will change from 500 Hz to **5 Hz** (100ms ON, 100ms OFF). You must use **Timer 1 CTC Interrupt** to generate the 5Hz clock frequency
  + If you press the button again, the blinking frequency will change from 5 Hz to 500Hz.
  + The swapping process repeats if we keep pressing the button.

You must use **External Interrupt INT0** (with falling edge detection) to detect the button pressed. Chart, waterfall chart

Description automatically generated

## 1.2 Submission

* Submit your Ino file for this question
* Take 1-2 waveform screenshots using VB or OSC to demonstrate the system running. You should include the frequency before and after the button pressed.

# 2.Question 2 (6 marks)

## 2.1 Requirements

Assume that you got an Arduino to connect to:

* A button connects to **PORTD2** – External Interrupt INT0. This is known as the **Gear** button. We use a pull-up configuration.
* Another button connects to **PORTD3** – External Interrupt INT1. This is known as the **Direction** button. We use a pull-up configuration.
* Three LEDs via PORTB0 to PORTB2.

Write a program to control as follows:

1. When the program is loaded or after reset, the three LEDs sweep from PORTB0 to PORTB2 at three intervals:
   * + Interval 1 – PORTB0 is ON, PORTB1 and PORTB2 are OFF
     + Interval 2 – PORTB0 is OFF, PORTB1 is ON, and PORTB2 is OFF
     + Interval 3 – PORTB0 is OFF, PORTB1 is OFF, and PORTB2 is ON
     + Interval 4 repeats Interval 1

Each interval lasts **0.125 second.**

1. When the **Gear** button (INT0 button) is pressed, the program still sweeps through the three LEDs but with different duration of the interval as follows:
   * If you press once, it will change the interval from **0.125** to **0.25 second (0.125 x 2)**
   * If you press again, it will change the interval from **0.25** to **0.375 second (0.125 x 3)**
   * If you press again, it will change the interval from **0.375** back to **0.125 second.**
   * The process repeats if you keep pressing the button.
2. When the **Direction** button (INT1 button) is pressed, the sweeping direction is changed between two directions, from PORTB0-to-PORTB2 to PORTB2-to-PORTB0 and vice versa.

**Technical requirements:**

* You must use the Timer 1 interrupt to generate the required interval above.
* You must use External Interrupts for the two buttons with falling edge detection sense control.

The function of the system can also be viewed via a video that I have prepared here - <https://www.youtube.com/watch?v=cMNYsz7C1Jc>

## 2.2 Submission

* Submit your Ino file for this question
* Take some waveform screenshots using VB or OSC to demonstrate the system running

# 3.Question 3 (7 marks)

## 3.1 Requirements

You will write a program to generate the following waveform **ONCE** every time a button is pressed:

* Step 1 - A button is pressed (i.e., a falling edge is generated).
* Step 2 - A GPIO PIN will raise to HIGH immediately.
* Step 3 - After 0.1 seconds, another GPIO PIN will generate a 5Hz clock (0.1 seconds ON, 0.1 seconds OFF) for 16 cycles.
* Step 4 – the GPIO PIN 1 will be clear to zero at the end of the 16th cycle.

A screenshot of a computer

Description automatically generated

You must use **External Interrupt** to process the button request. Two GPIOs can be chosen freely. For delay time, you must use Timer to generate the correct time intervals (i.e., no **delay\_ms** function is allowed in this Exercise) with or without interrupts.

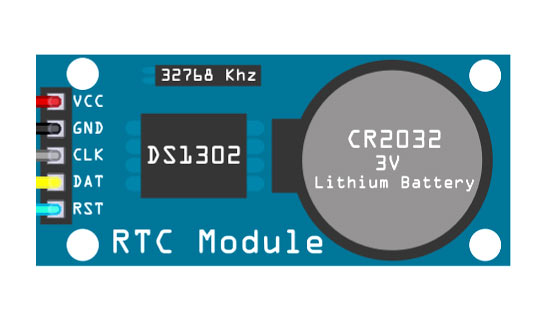
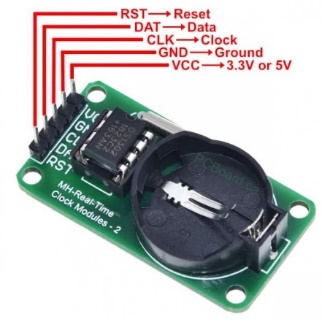
## 3.2 Submission

* Submit your Ino file for this question
* Take 1-2 waveform screenshots using VB or OSC to demonstrate the system running. You should include the frequency before and after the button pressed.

# Question 4 (10 marks)

## 4.1 Background

There is a need to keep track of real-time in some applications even if we turn off the power. This can be achieved using timekeeper boards. Such board stores time permanently even if the power is gone (with the help of a coin battery). For instance, your laptops/PCs have one. This question uses the timekeeper DS1302 board. Its datasheet is found at <https://datasheets.maximintegrated.com/en/ds/DS1302.pdf> .



*Figure 1 – DSC1302 module*

DSC1302 board can be connected to Embedded board through a **3-wire serial interface** (i.e., data are transferred in series in a bitstream).

A blue rectangular object with black arrow pointing to the screen

Description automatically generated

*Figure 2 – Connections between DS1302 module and Arduino.*

## 4.2 Requirements

In this question, you will be given a DSC1302 board with a pre-loaded time already. Your task is to write a program running on Arduino to read this data out from the board. In specific, the board has a single-byte mode, allows users to read one single byte every time. Its timing transfer diagram for the single -byte read is given in page 13 of the datasheet and shown below:

A diagram of a transfer summary

Description automatically generated

*Figure 3 – Data transfer diagram for single-byte read mode*

Your program will start read the time from the DSC1302 when a button is pressed. You must use a button connected to PD0 and use External Interrupt INT0 to start the read process. As the DSC1302 returns data via the interface, to confirm if your program is working, you will need to show the data response from the chip using OSC or VB. You then need to analyze this data.

*Note - this timing diagram is in a similar form as the one in Question 3. You can modify it if required to suit the exact timing requirement of the mode as described in the datasheet.*

## 4.3 Other Requirements

* You must **write your own control functions for read with the DSC1302 module and demonstrate your understanding**. You are not allowed to use an external library to complete this mini project. External libraries are those developed by other developers and packaged either in .c or .h files.
* You must use a Timer interrupt to generate a delay when communicating with the Timer module according to the timing diagram above.
* You must demonstrate the successful implementation of the single-read mode via Waveform measured from VB or OSC. The waveform pictures should be included in the submission.
* The final prototype must be constructed on a breadboard.
* You are recommended to write separate functions in C to partition the design and troubleshoot the design.
* You are recommended to plan your pin allocation wisely. Since we got many peripherals now and your MCU has a limited number of pins, you might need to plan your pin allocations wisely and develop a proper control strategy.

## 4.4 Submission

* Submit your Ino file for this question
* Take 1-2 waveform screenshots using VB or OSC to demonstrate the system running. You should include the frequency before and after the button pressed.

# Question 5 – Demonstration (**12 marks)**

In addition to the above questions, students need to demonstrate their systems to the Lecturer and answers questions.

Tasks include

* Explain one exercise and perform measurements - this is for each student
* Demonstrate question 4 – this is for both students

More information will be communicated regarding demonstration.