Report

Laboration 4

Timer and UART

Author: Amata ANANTAPRAYOON,
Adell TATROUS
Student ID: aa224iu,
at222ux
Semester: VT2018
Course: Computer Technology 1
Course code: 1DT301

Contents

1 Task 1: Square wave generator

Write a program in Assembly that creates a square wave.One LED should be connected and switch with the frequency 1 Hz. Duty cycle 50%. (On: 0.5 sec, Off: 0.5 sec.)Use the timer function to create an interrupt with 2 Hz, which change between On and Off in the interrupt subroutine.

1.1 Flowchart

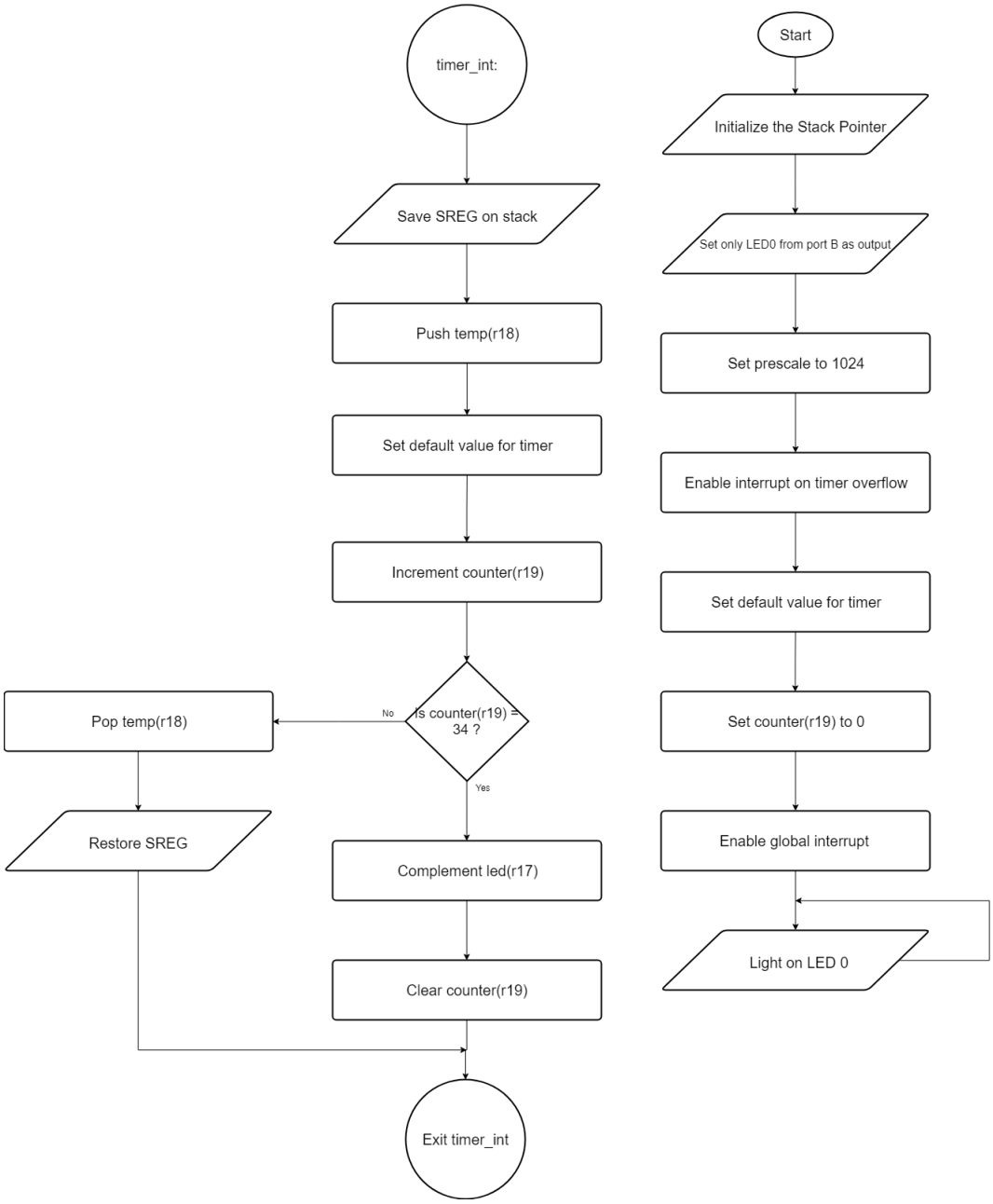

Figure 1: Flowchart for task 1

2 Task 2: Pulse Width Modulation (PWM)

Modify the program in Task 1 to obtain Pulse Width Modulation (PWM).The frequency should be fixed, but the duty cycle should be possible to change. Use two push buttons to change the duty cycle up and down. Use interrupt for each push button. The duty cycle should be possible to change from 0% up to 100% in steps of 5%. Connect the output to an oscilloscope, to visualize the change in duty cycle.

2.1 Flowchart

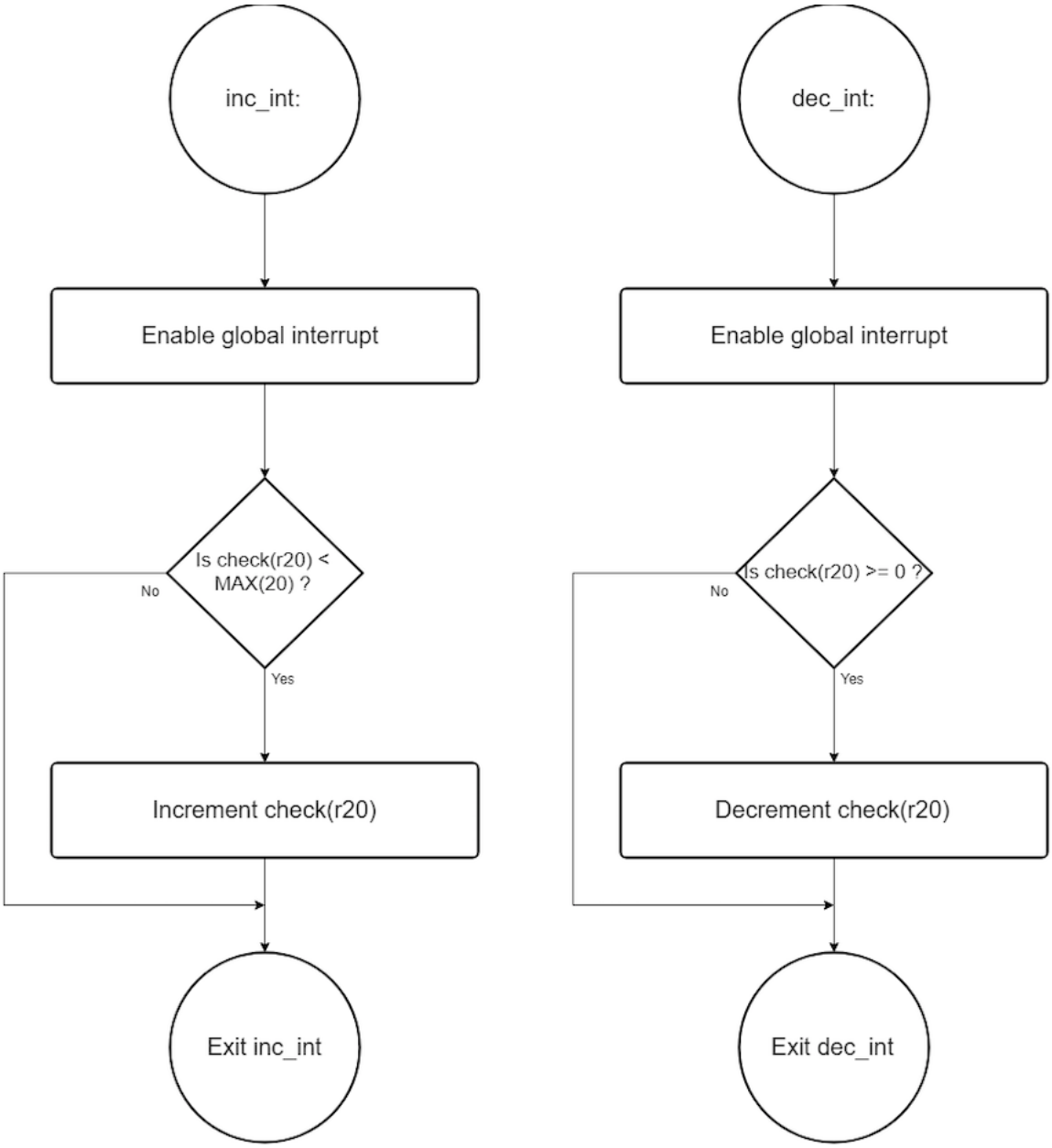

Figure 2: Flowchart for task 2 (interrupt part)

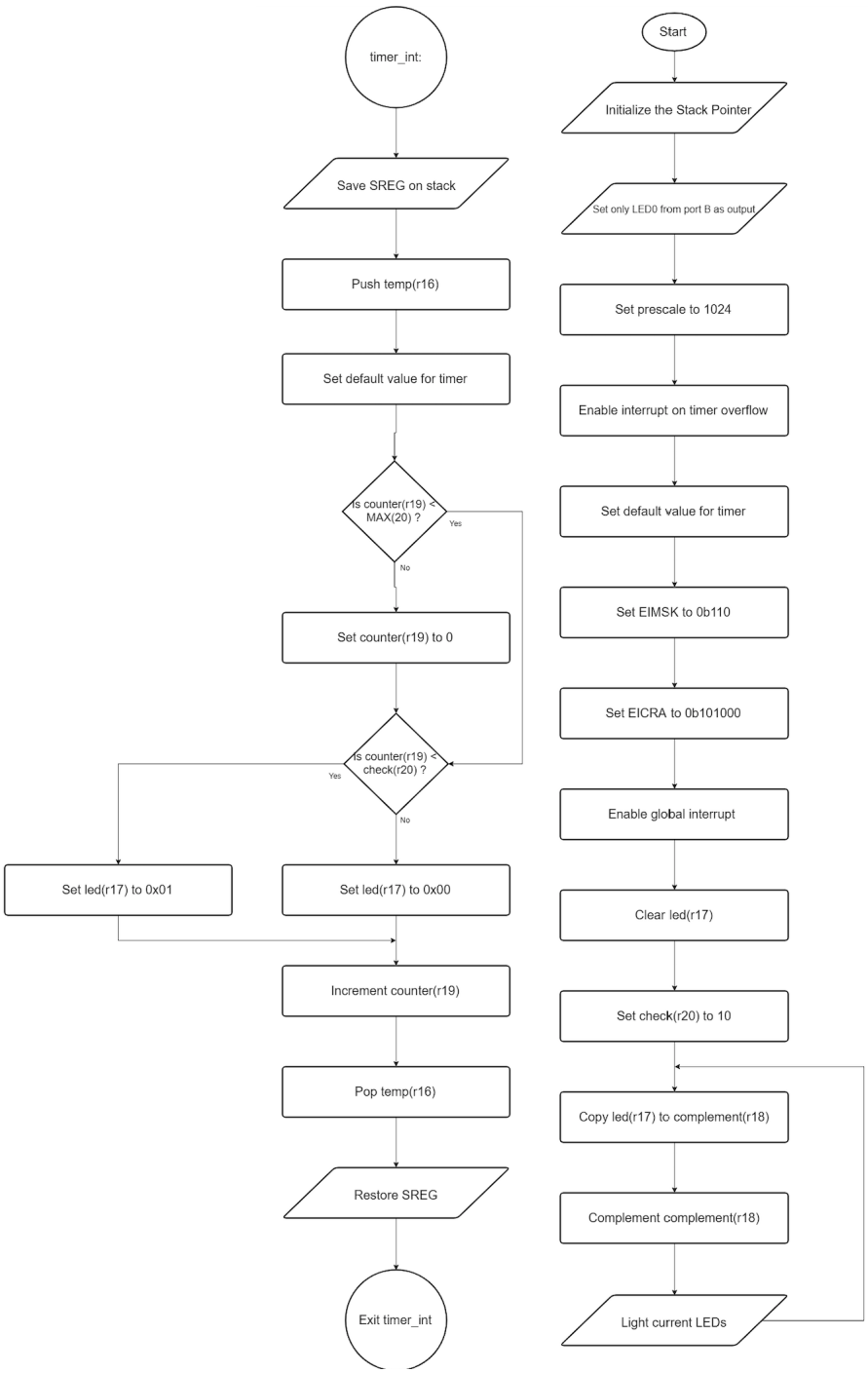

Figure 3: Flowchart for task 2 part 2

3 Task 3:Serial communication

Write a program in Assembly that uses the serial communication port0(RS232).Connect a computer to the serial port and use a terminal emulation program. (Ex. Hyper Terminal)The program should receive characters that are sent from the computer, and show the code on the LEDs. For example, if you send character A, it has the hex code $65, the bit pattern is 01100101 and should be displayed with LEDs On for each 'one'. Use polled UART, which means that the UART should be checked regularly by the program.

3.1 Flowchart

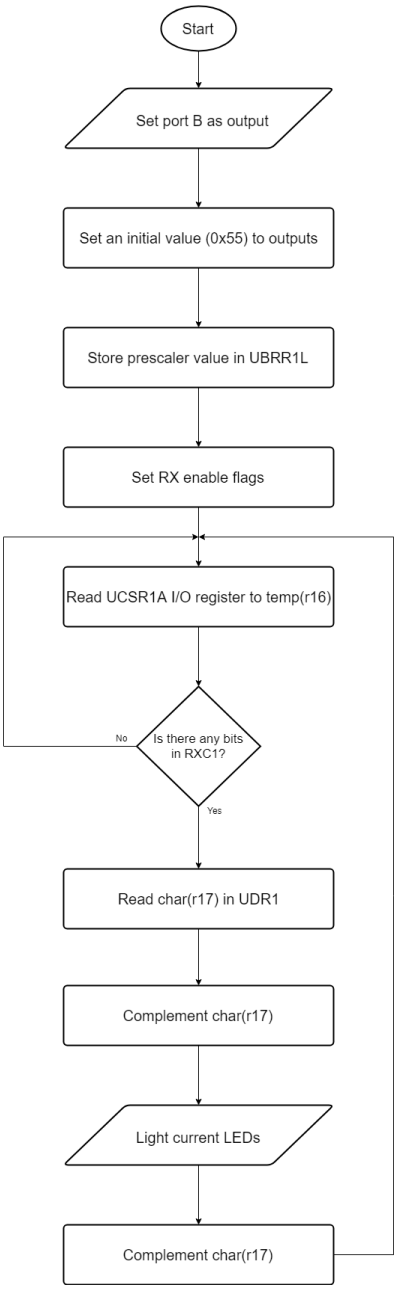

Figure 4: Flowchart for task 3

4 Task 4: Serial communication with echo

Modify the program in task 3 to obtain an echo, which means that the received character should also be sent back to the terminal.This could be used as a confirmation in the terminal, to ensure that the character has been transferred correctly.

4.1 Flowchart

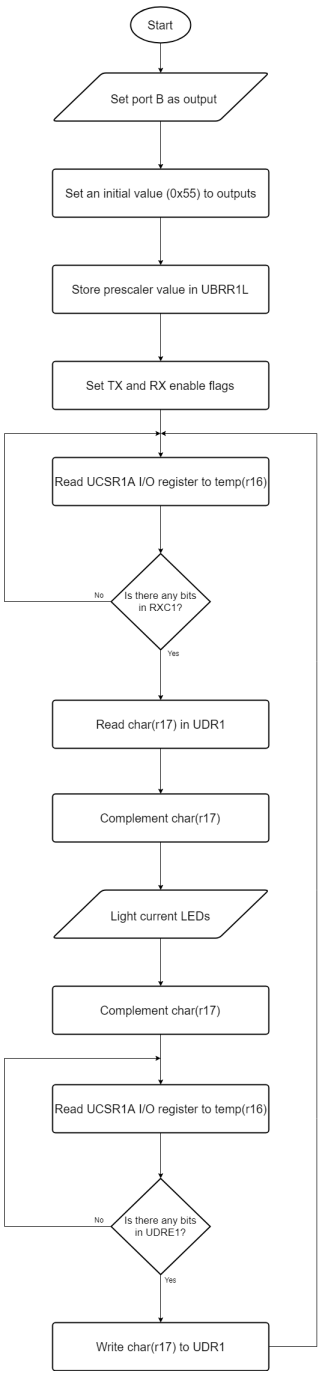

Figure 5: Flowchart for task 4

5 Task5: Serial communication using Interrupt

Do task 3 and 4, but use Interrupt instead of polled UART.(USART, Rx Complete, USART Data Register Empty and USART, Tx Complete)

5.1 Flowchart

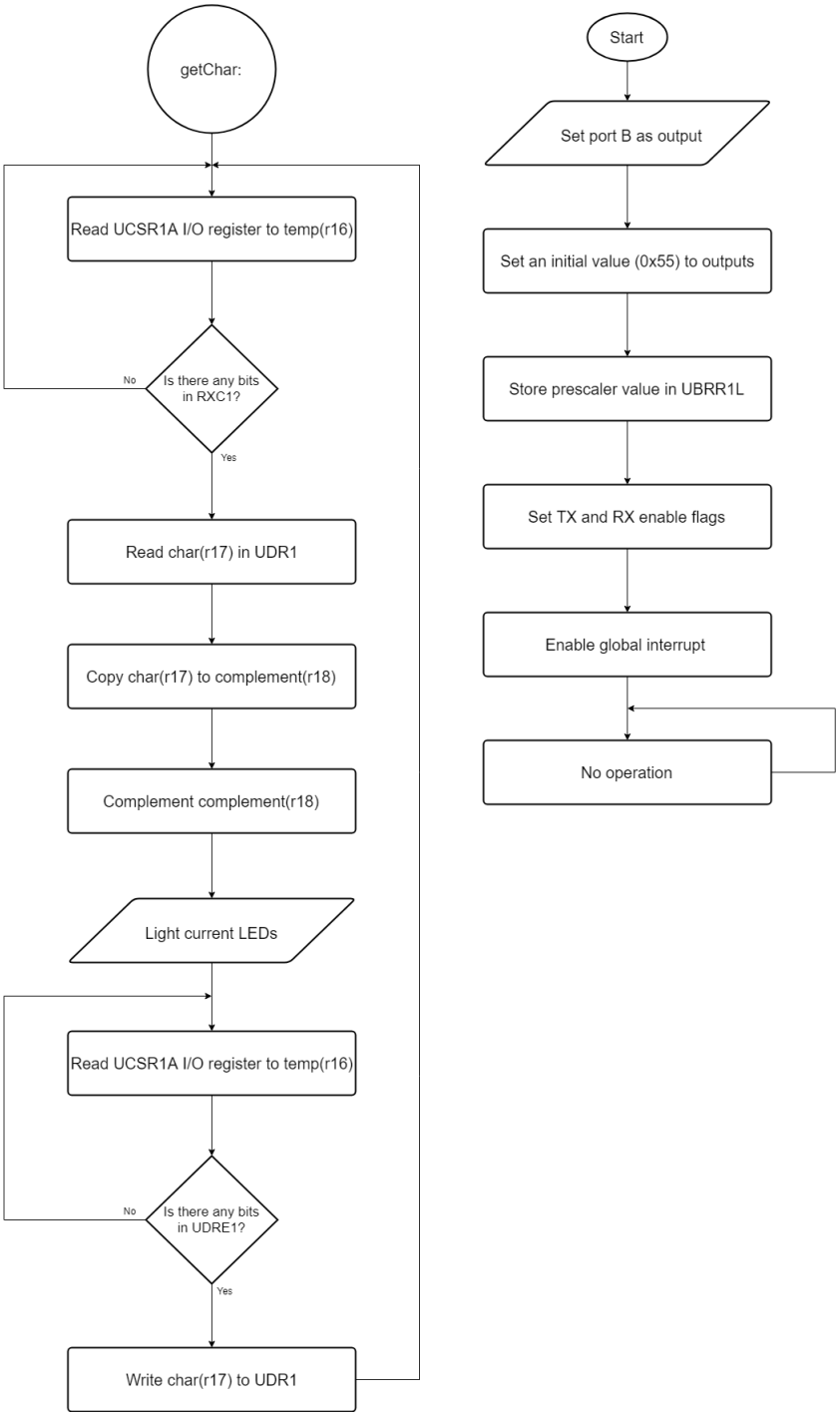

Figure 6: Flowchart for task 5

6 Assumption

- **Task 2:**
  - User use SW1 to decrease duty cycle by 5%
  - User use SW2 to increase duty cycle by 5%
  - The switches are buggy, sometime is increase/decrease by 10% instead
- **Task 3-4:**
  - User should setup PuFFY as the Figure 7

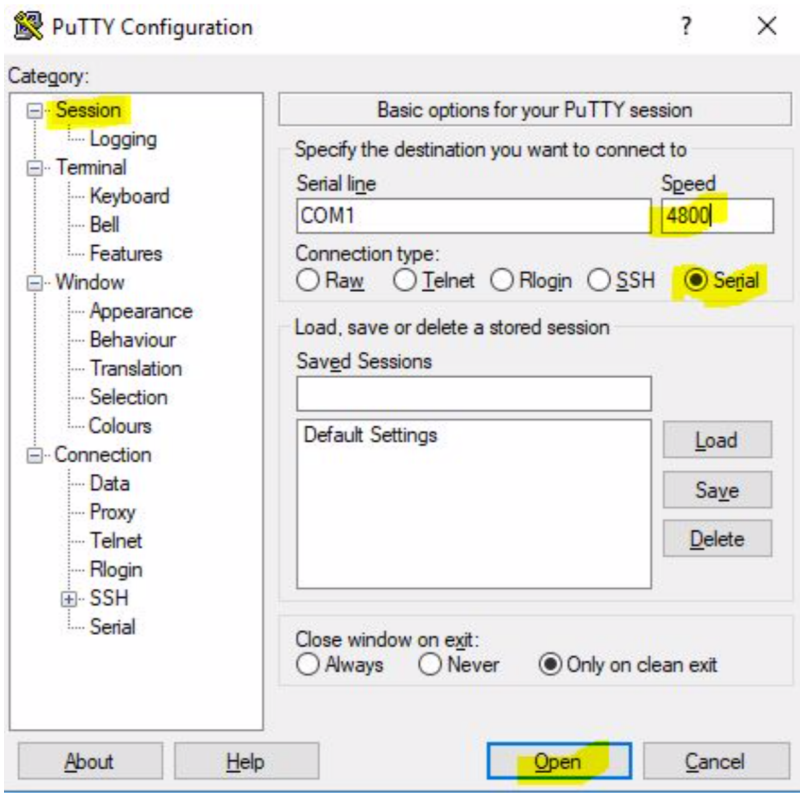

Figure 7: Flowchart for task 5