CECS 347 Project 1

An Ultrasonic Ranger Finder

Total: 25 points+3 bonus points

Preparation:

You will need a LaunchPad, an HC-SR04 Ultrasonic sensor, a ruler to measure distance in centimeters, and three resistors (one 1kΩ and one 2kΩ or three 1kΩ) for building a voltage divider.

New Topics Covered: General Purpose Timer, Ultrasonic Sensor, UART.

Reference Projects: GPTM_Timer0A, UltrasonicSensor, UART0Out.

Purpose:

The purpose of this project is to review the following topics learned: GPIO, SysTick timer, Edge interrupt and to learn and practice the following new topics:

- PLL: Phase Lock Loop, learn how to generate a system clock using PLL.
- GPTM: General purpose timers, learn how to generate periodic timing with one general purpose timer.
- Ultrasonic sensor: Learn how to interface HC-SR04 ultrasonic sensor to detect obstacle range.
- UART: learn PC to microcontroller UART communication.

System Requirements:

Design a system with an HC-SR04 ultrasonic sensor that provides information on the distance of the closest obstacle in front of the system in the following way: PC serial terminal will display the distance information with an integer value, the onboard LEDs will show distance information in the following way: distance <10cm, blink the red LED at a frequency of 2Hz: 0.25s on/0.25s off; 10cm <distance < 70cm, turn on green LED, distance > 70cm, turn on blue LED.

Hardware Design Requirements:

1. PLL is required to generate a 16MHz system clock.
2. Use General Purpose Timer1 in 16-bit countdown mode to generate trigger pulses and to measure ultrasonic echo high period.
3. Use SysTick timer with interrupt enabled to control the blinking of the red LED.
4. Edge interrupt is required to capture the rising edge and the falling edge of the echo pin.
5. Use the following pins to interface with the HC-SR04 Ultrasonic sensor: PB5 connects to the trigger pin, PB4 connects to the echo pin.
6. Use UART0 to display current distance on a PC serial terminal.

Development Steps:

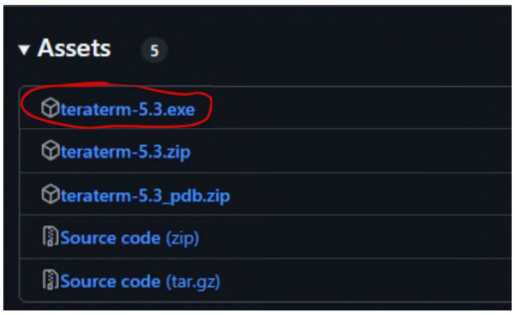
1. (5 points) Project1P1: Build an embedded system with PLL to generate a 16MHz system clock. Use SysTick timer with interrupt enabled to generate a 2Hz square wave and output to the onboard red LED. Simulate on simulator and screen shot the logic analyzer's outputs with the grid set to be 0.5s. Demonstrate the system on board.
2. (5 points) Project1P2: Keep the same system clock frequency of 16MHz, replace Systick timer with General Purpose Timer1 in 16-bit countdown mode with **prescale** value greater than 0 to generate a 2Hz square wave and output to the onboard red LED. Simulate on simulator and screen shot the logic analyzer's outputs with the grid set to be 0.5s. Demonstrate the system on board.
3. (5 points) Project1P3: add an ultrasonic sensor to Part 2, modified Part 2 code in the following way: use General Purpose Timer1 in 16-bit count down mode to generate trigger pulses and to measure ultrasonic echo high period. You will need to adjust the prescale value so that the maximum time duration generated by timer1 is slightly greater than the maximum echo pulse width. Edge interrupt is required to capture the rising edge and the falling edge of the echo pin. Simulate on simulator and screen shot the logic analyzer outputs with the grid set to be 10us. Demonstrate onboard in debug mode with the right break point and show three different distances: <10cm, [70cm, 100cm] and between [10cm, 70cm].
4. (5 points) Project1P4: Add UART0 to the system to display the current distance with an integer value. The required Baud Rate for UART0 is 57600. We will use onboard sw1 to control when to read the distance: every time sw1 is pressed, the system will read one distance value and display it on PC serial terminal. See display format below. Add LED display for the following three distance ranges: turn on blue LED for 100cm>distance>70cm, turn on green LED for 10cm < distance < 70cm, blink the red LED at a frequency of 2Hz: 0.25s on/0.25s off for distance < 10cm, turn off all LEDs when nothing in front of the sensor within 100cm and display OUT OF RANGE in PC serial terminal. Demonstrate the system onboard and screen shot the displays on PC serial terminal. The display on PC serial terminal should follow the examples shown below:
   The current distance is 12cm.
   The current distance is 72cm.
   The current distance is OUT OF RANGE.
   …

Tera Term Set Up:

Download the latest version of tera term here:

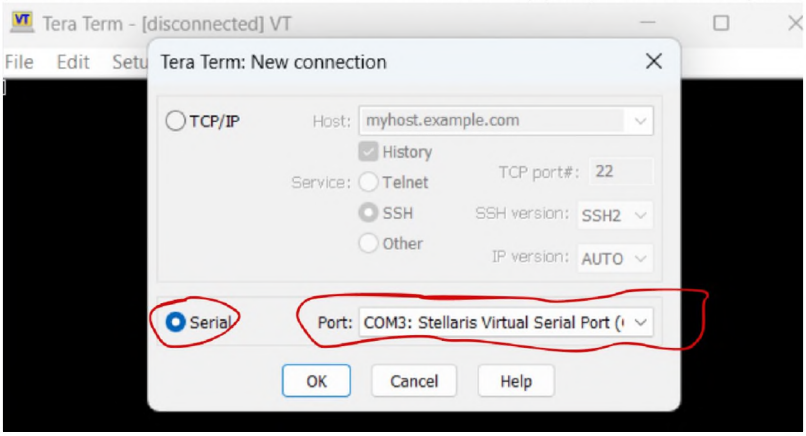
https://github.com/TeraTermProject/teraterm/releases/tag/v5.3

Scroll down to the bottom to find the .exe

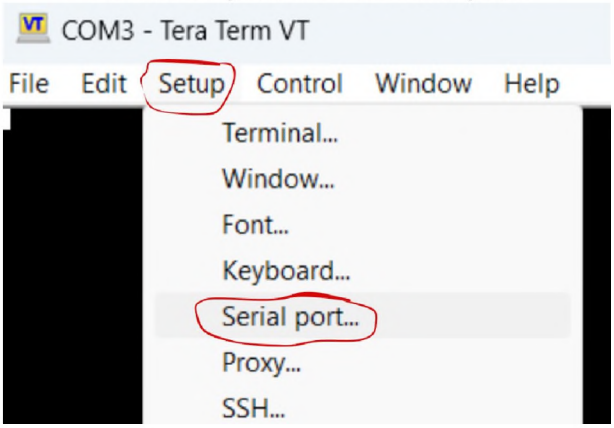


Run the file after it downloads, all default settings are fine

Once installed you will want to select “Serial” and the appropriate COM port for your board

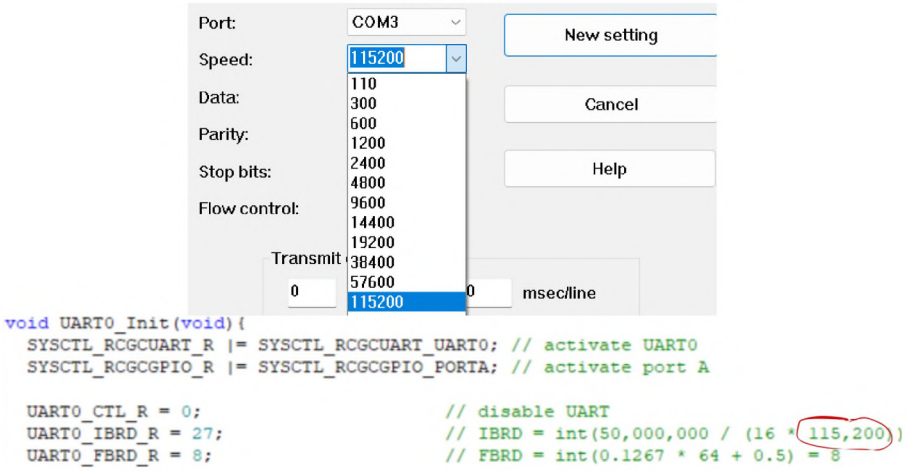


Go to setup and select serial port



Make sure the speed is set to your desired baud rate which is usually found in your UART init

function




```
void UART0_Init(void){
  SYSCTL_RCGCUART_R |= SYSCTL_RCGCUART_UART0; // activate UART0
  SYSCTL_RCGCGPIO_R |= SYSCTL_RCGCGPIO_PORTA; // activate port A

  UART0_CTL_R = 0;                    // disable UART
  UART0_IBRD_R = 27;                  // IBRD = int(50,000,000 / (16 * 115,200))
  UART0_FBRD_R = 8;                   // FBRD = int(0.1267 * 64 + 0.5) = 8
```

Extra Challenges:

1. (3 points) For Project1P3: use General Purpose Timer1 “Input Edge Time” mode to measure echo high period width.

Deliverables:

1) Demonstrate your lab and submit source code (include only the .c/.h files you write, **DO NOT** include other project files) for each of the three parts separately, one zip file per part: Project1P1.zip, Project1P2.zip, Project1P3.zip and Project1P4.zip. For the extra challenge, submit: Project1P3Ex.zip. (20+3 points)
2) Project report: submit demonstration video separately or add video link in "Operation" section of the report. (5 points)