

Lab 4: UART (Universal Asynchronous Receiver & Transmitter)

Instructor: Sung-Yeul Park

TA: S M Rakiul Islam

ECE 3411

Department of Electrical and Computer Engineering
University of Connecticut

January 29th, 2018

Connections

- Connect the board as in Fig.1. The configuration is the same as Lab 3.
- Disconnect LEDs on PD0 and PD1
- PD0 and PD1 are used as Tx and Rx pins for UART. So, you can't use those for other purposes while using UART.
- If you don't disconnect those LEDs, then those could blink/ be turned on while using UART.

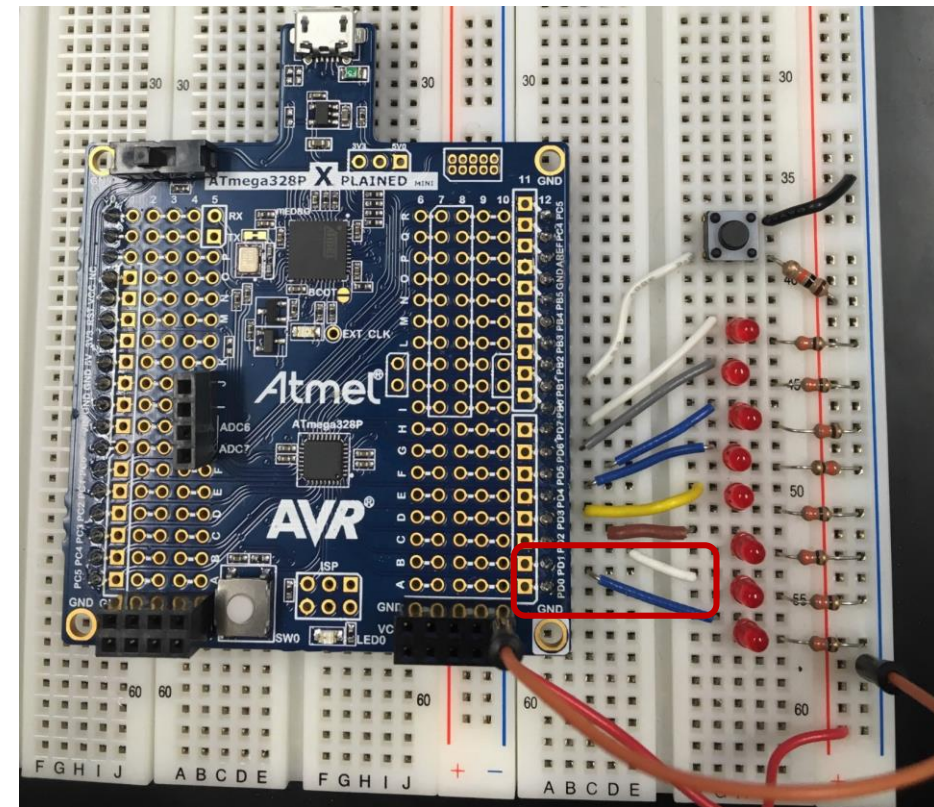


Fig1. Connections for GPIO and UART.

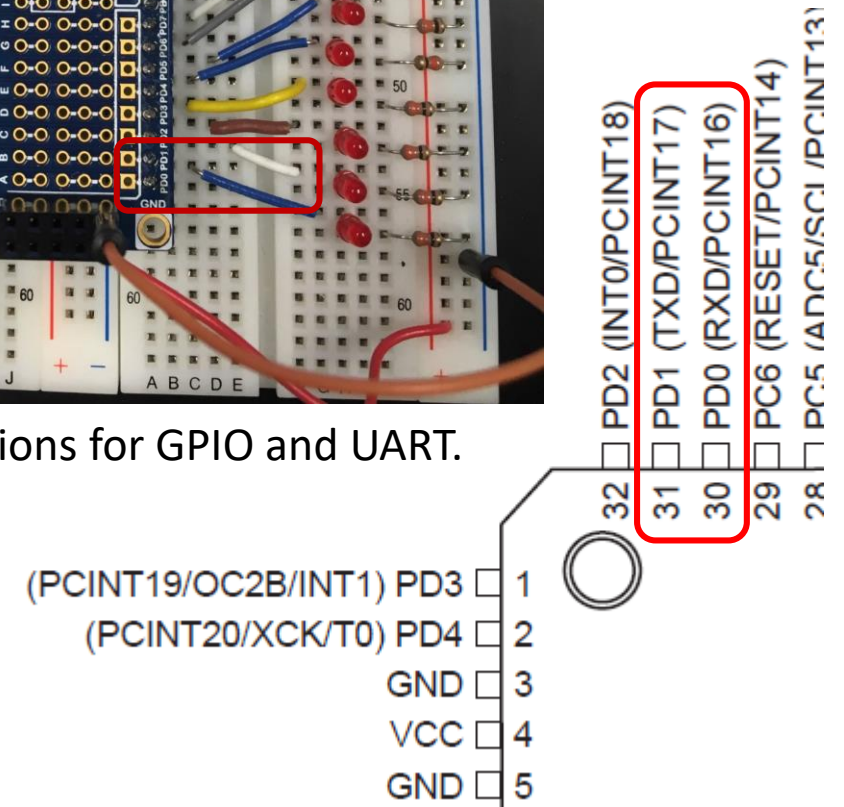
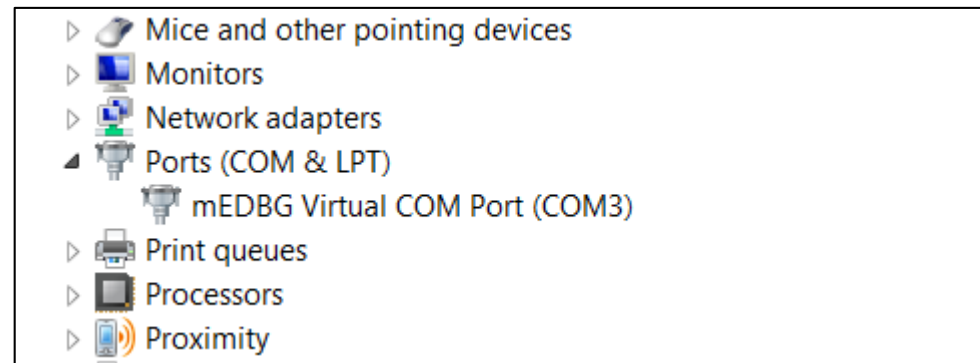


Fig2. UART Tx and Rx Shared with PD0 and PD1

UART Setup: COM Port Identification (1)

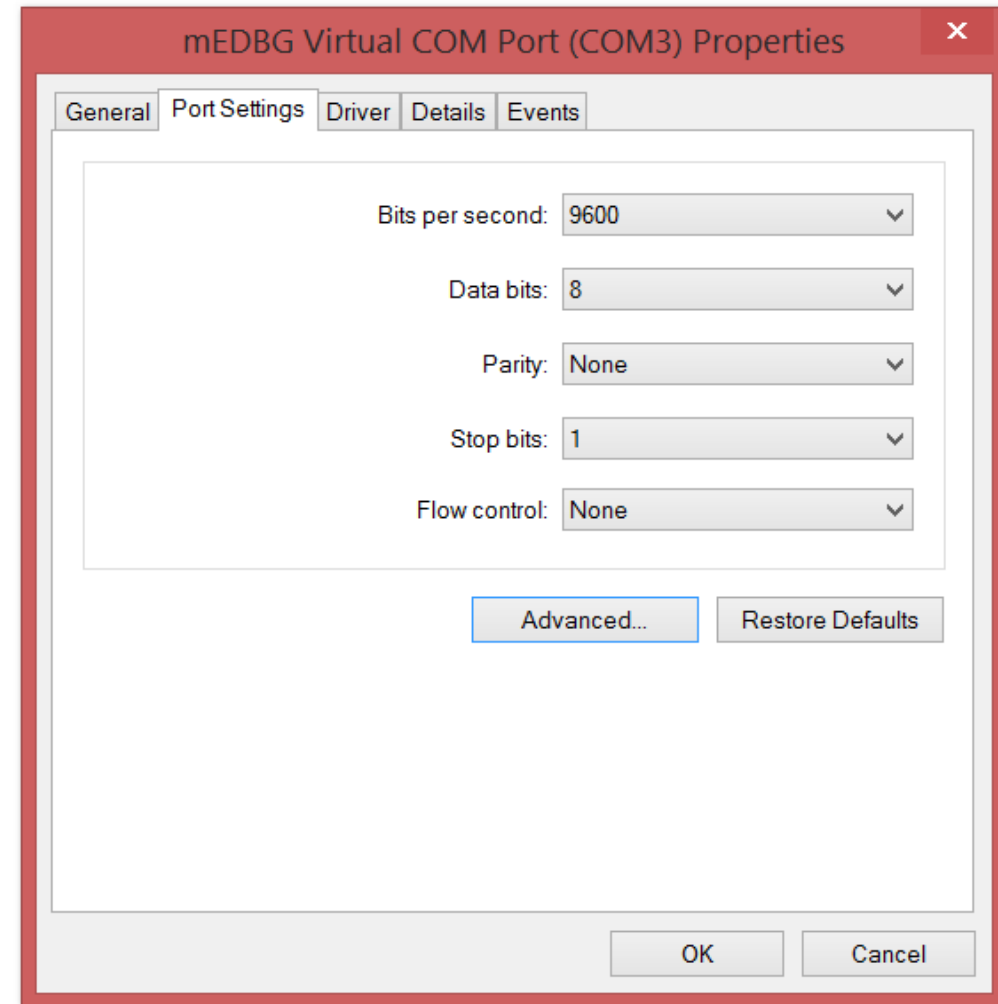
In order to setup UART communication between the Xplained mini and your PC, we first need to identify and setup the COM port used by Xplained Mini board

- Connect the Xplained Mini board to your computer via USB cable
 - Go to: Control Panel → Device Manager
 - Expand the Ports (COM & LPT) section as shown in the figure below.
 - Note down the Port number shown against mEDBG Virtual COM Port
- Port, i.e. COM3 in the figure below.



UART Setup: COM Port Identification (1)

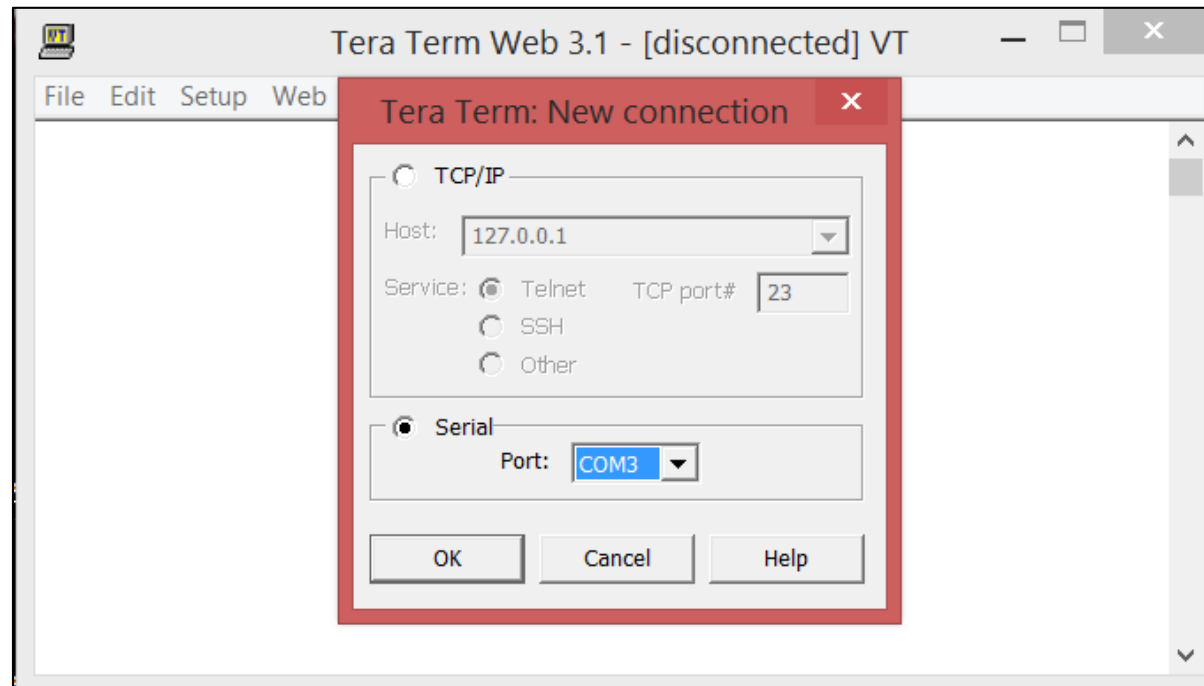
- Double Click to open the Properties window of mEDBG Virtual COM Port.
- Make sure that the Port Settings are the same as shown below.
- If necessary, the COM Port number can be changed under Advanced tab. However, generally the default COM Port number will work.



UART Setup: TeraTerm Pro

We will use TeraTerm Pro terminal to send/receive data to Xplained Mini over UART

1. Download ttpro313.zip file.
2. Unzip the file and run the application **ttermpro.exe**
3. In the New Connection window, select Serial and select your mEDBG COM Port number, e.g. COM3 (refer to the previous slide) and click OK.



Source: Presentation from Marten Van Dijk

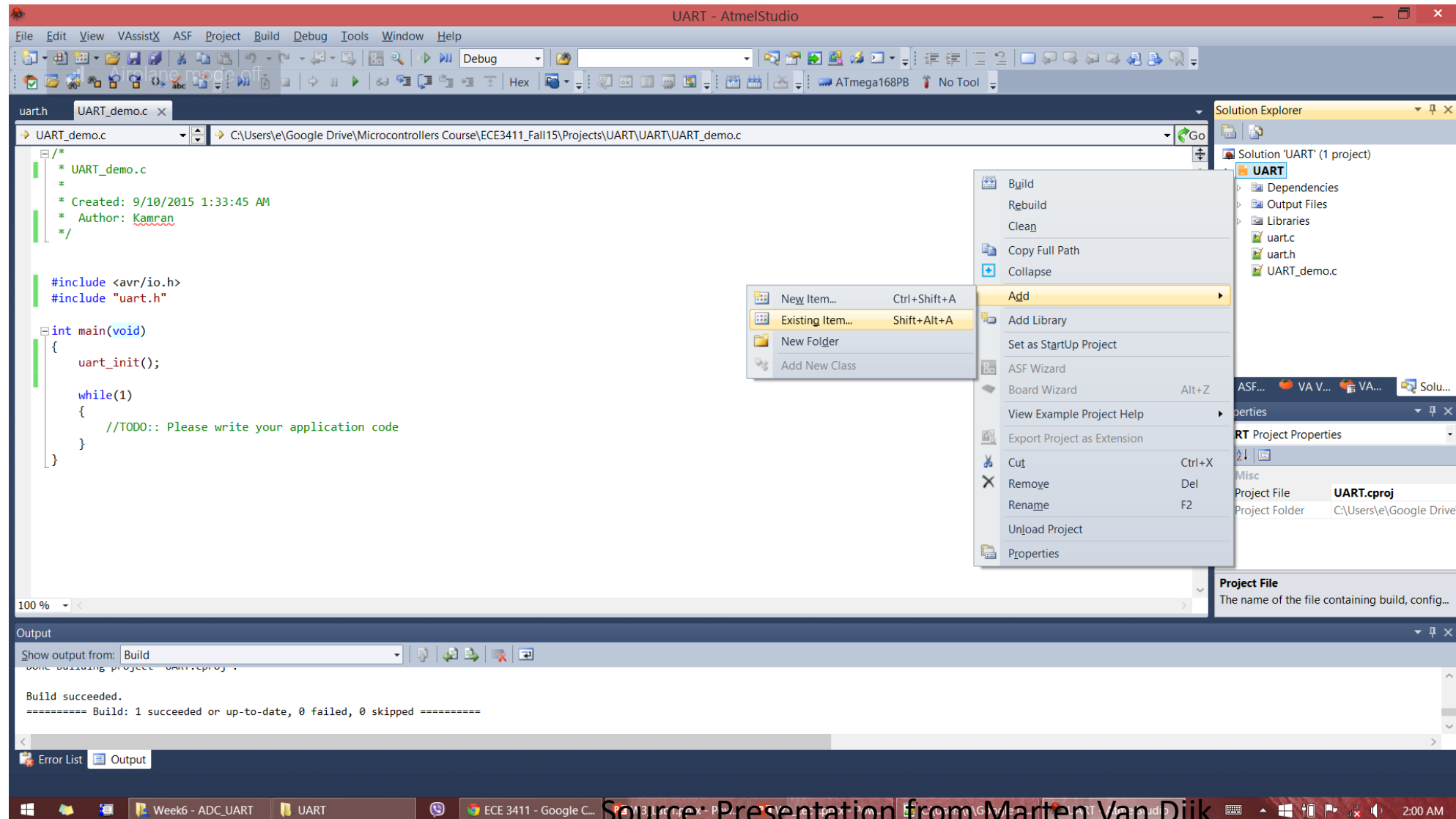
UART Setup: Using uart.h Library

- In order to facilitate you, we provide a library file “uart.c” which defines some useful basic UART functions.
 - “uart.h” and “uart.c” can be downloaded in huskyCT from supplementary file folder.
- The corresponding prototypes of the functions are declared in “uart.h” file which comes along with “uart.c” file.
- In order to use the function provided by “uart.c”, you need to:
 1. Add “uart.c” and “uart.h” files in your Atmel Studio project source files
 2. Include “uart.h” as a header file in your code, i.e. `#include "uart.h"`

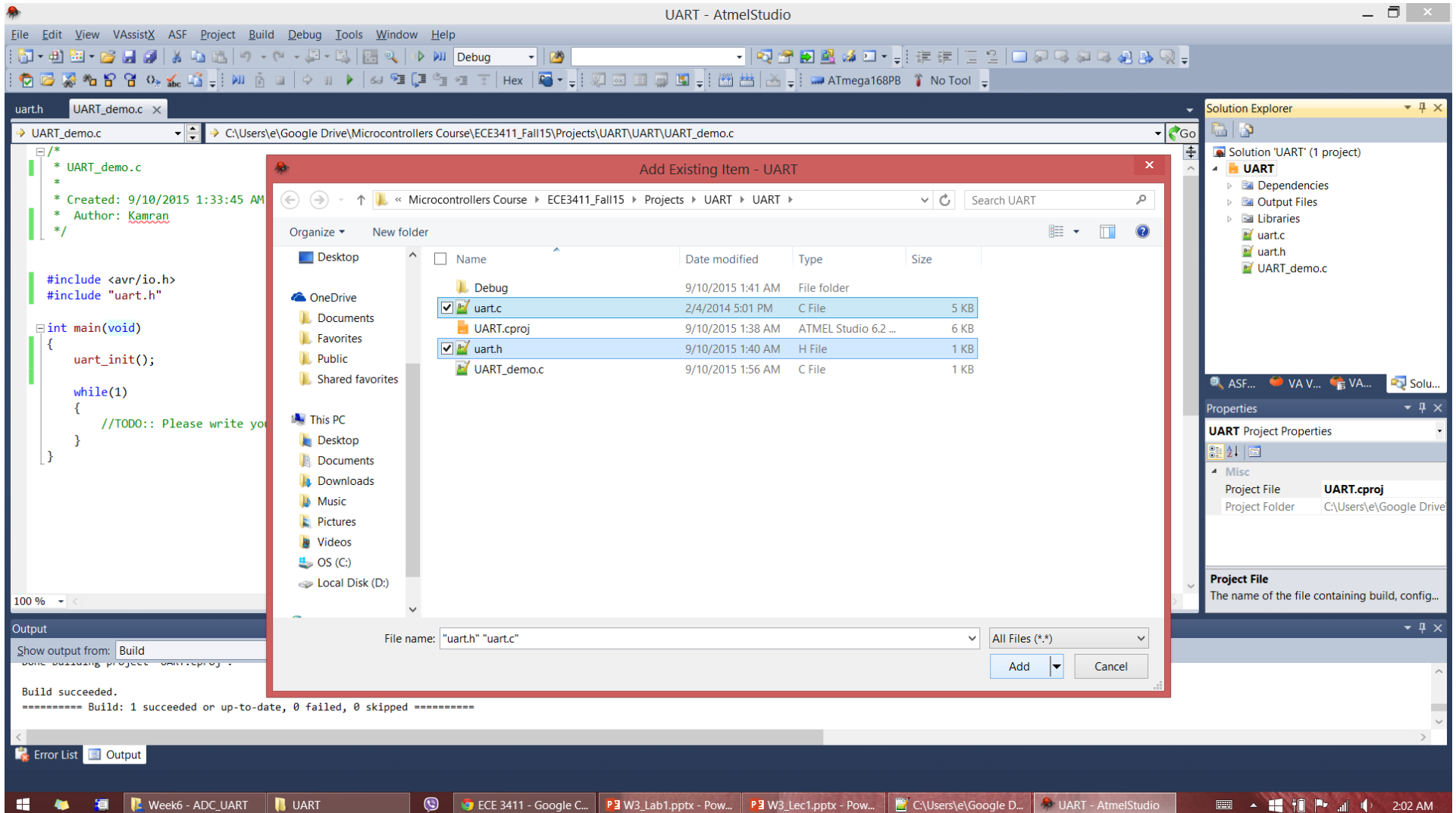
Adding Header and C Files to a Project

- Often, it is more convenient to include files within your project that contain definitions and functions that you will use frequently.
- This reduces the length of your main c file and eliminates the need for copying and pasting functions you've already written in the past.
- Suppose we want to add "uart.c" and "uart.h" to a project:
 1. Create a new project in Atmel Studio.
 2. Copy the files "uart.c" and "uart.h" into the project directory.
 3. In the 'Solution Explorer' window, right click on the project's name → Add → Existing Item ...
 4. Select "uart.c" and "uart.h" and click "Add".
 5. Don't forget to declare/include the header file in you code by calling `#include "uart.h"`
- See the next few slides for illustration

Adding Header and C Files to a Project



Adding Header and C Files to a Project



Source: Presentation from Marten Van Dijk

Programming practice

- Practice the Example of textbook 5.1, 5.2 and 5.3
- See the Atmel Studio Youtube training modules for UART
- Now make a program to control the frequency of blinking and show inputs

Task 1: Blinking a single LED

- Blink a single LED at two different rates based on a push switch.
 - When the switch is not pressed, LED should blink at 2Hz frequency.
 - As long as the switch is pressed, LED should blink at 8Hz frequency.
- The blinking duty cycle should be 50%
 - E.g. for 2Hz frequency, the LED should be on for $1/4^{\text{th}}$ of a second, then off for next $1/4^{\text{th}}$ of a second and so on.
- You may use the on-board LED and push switch for this task.
- No need to show task1 demo to TA. It is a practice for Task2.

Task 2: Changing LED Mode using UART

Extend Task 1 such that the blinking frequency of the LED can be switched between 2Hz and 8Hz depending upon the string entered from the UART Terminal.

- The LED starts blinking at 2Hz
- After every 10 seconds, the program prints the message on terminal: "Do you want to change the LED mode? (Y/N)"
- If the user enters "Y", the LED blinking rate switches to the other frequency
 - E.g. if currently the frequency is 2Hz then it switches to 8Hz and vice versa
- If user enters "N" then the frequency stays the same.
- You may use the on-board LED for this task.
- You need **to demonstrate this task2 to TA.**