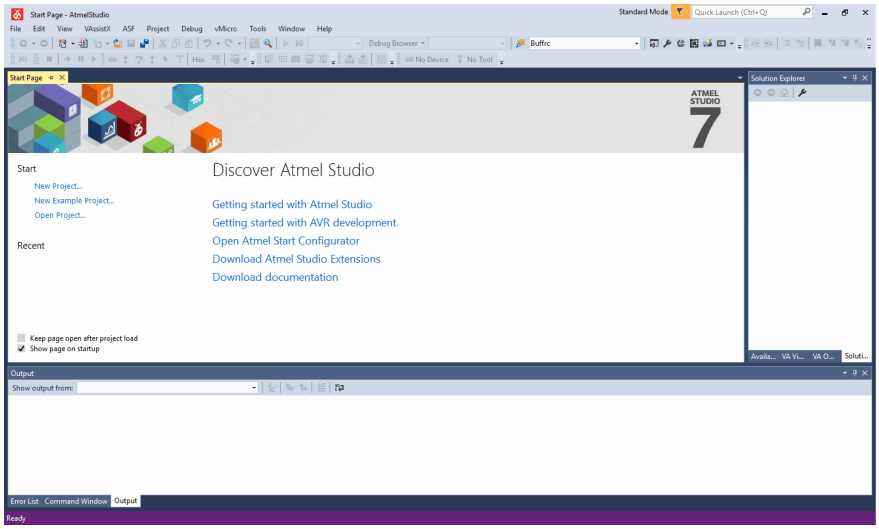
Getting Started with Atmel Studio

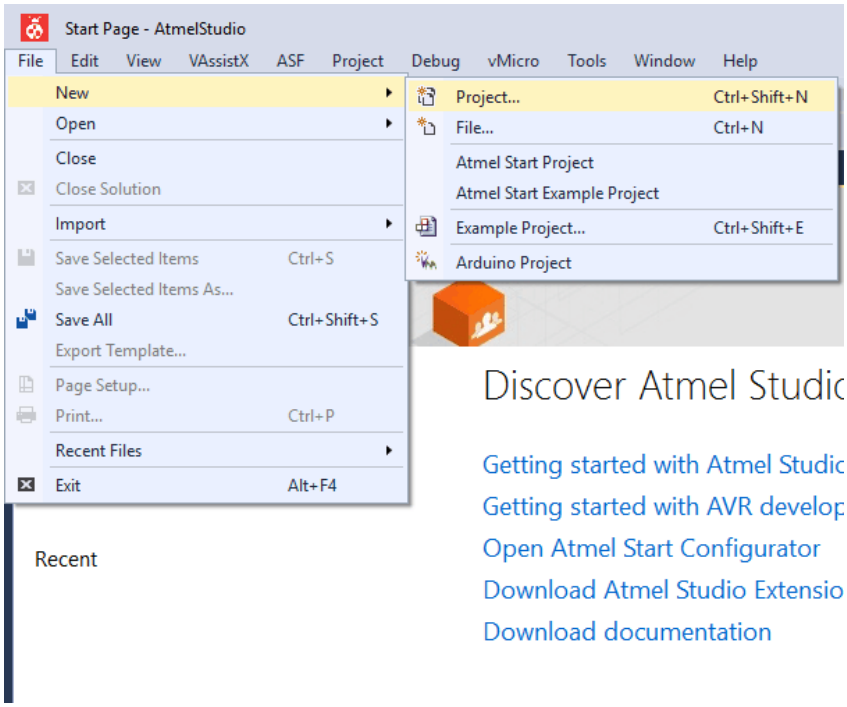
EXERCISE_00

Atmel Studio is available free of charge. To download and install Atmel studio use this link. Let's develop simple "LED Blinking" program for ATmega328P using Atmel Studio 7.

1. After done with downloading and installing, Open Atmel Studio 7. We can see Start Page as shown in below figure.

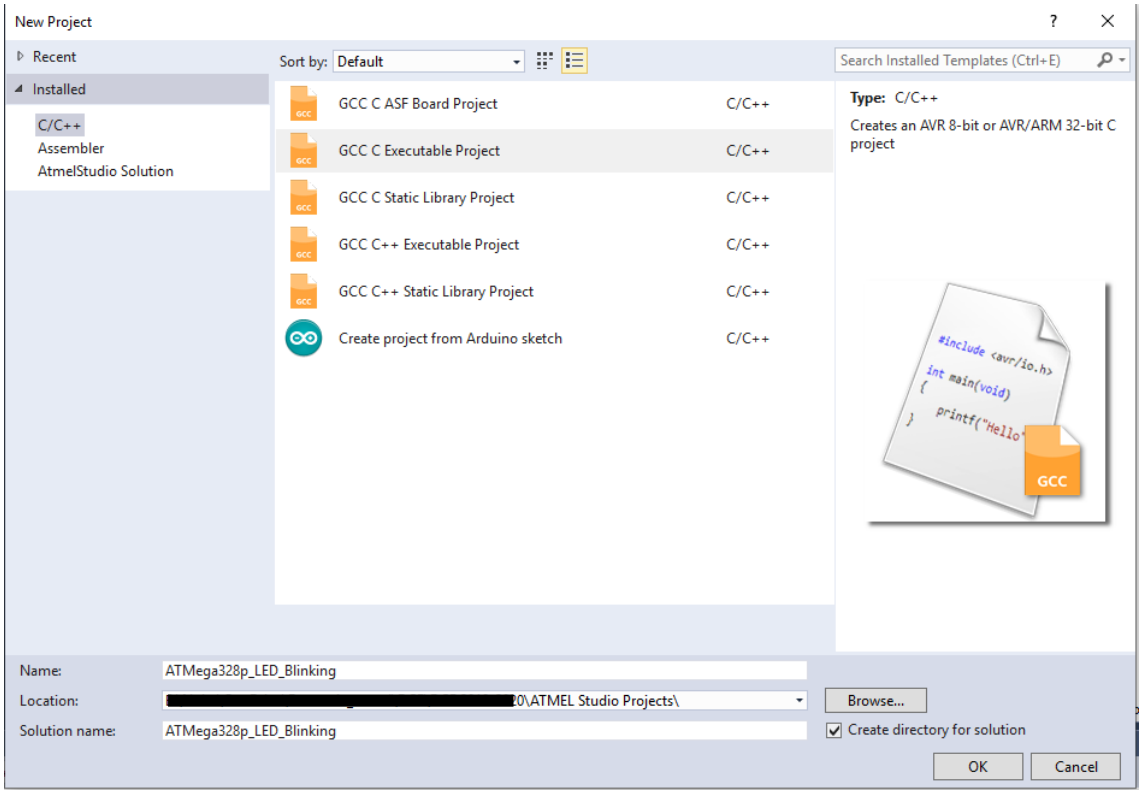


2. Now to create new project Click on **File -> New -> Project**



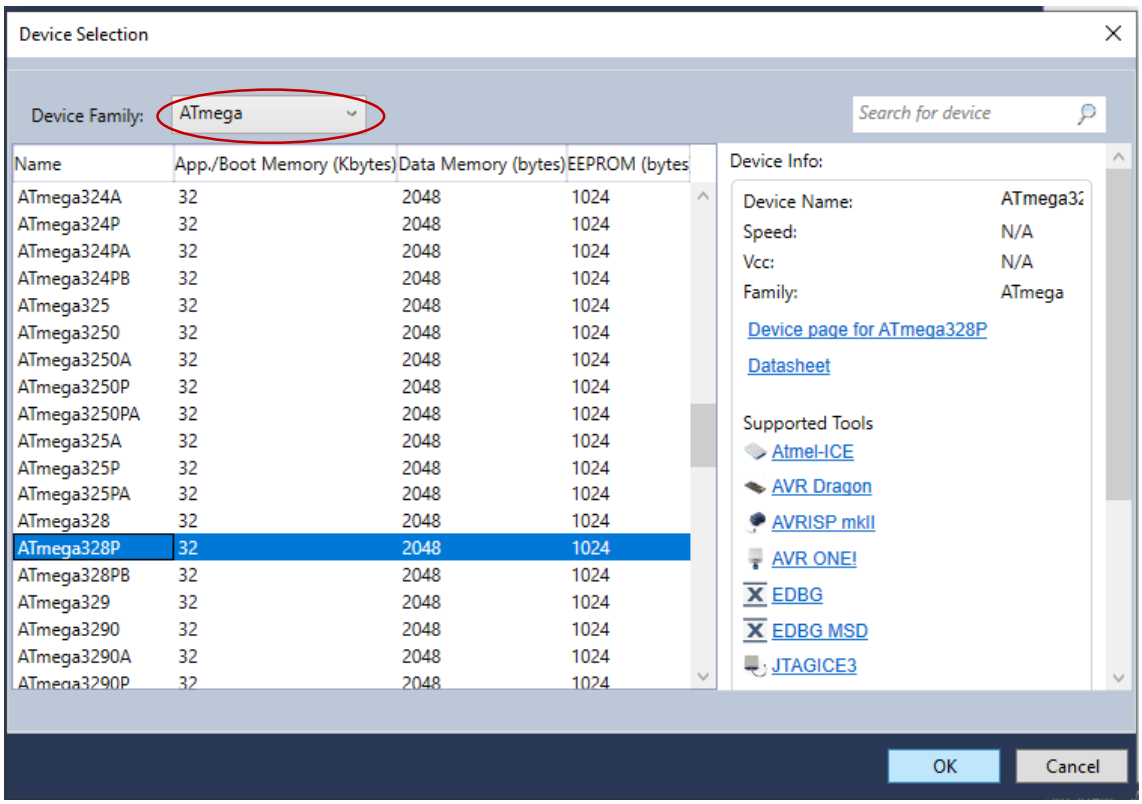
3. A **New Project** window will pop up as shown in below figure. In New Project window, we need to select **project type** as listed in below figure, **Name** for project and **Location** to where we can save project work.

Also, there is option **Create directory for solution,** which will create project directory with name of project at provided location.

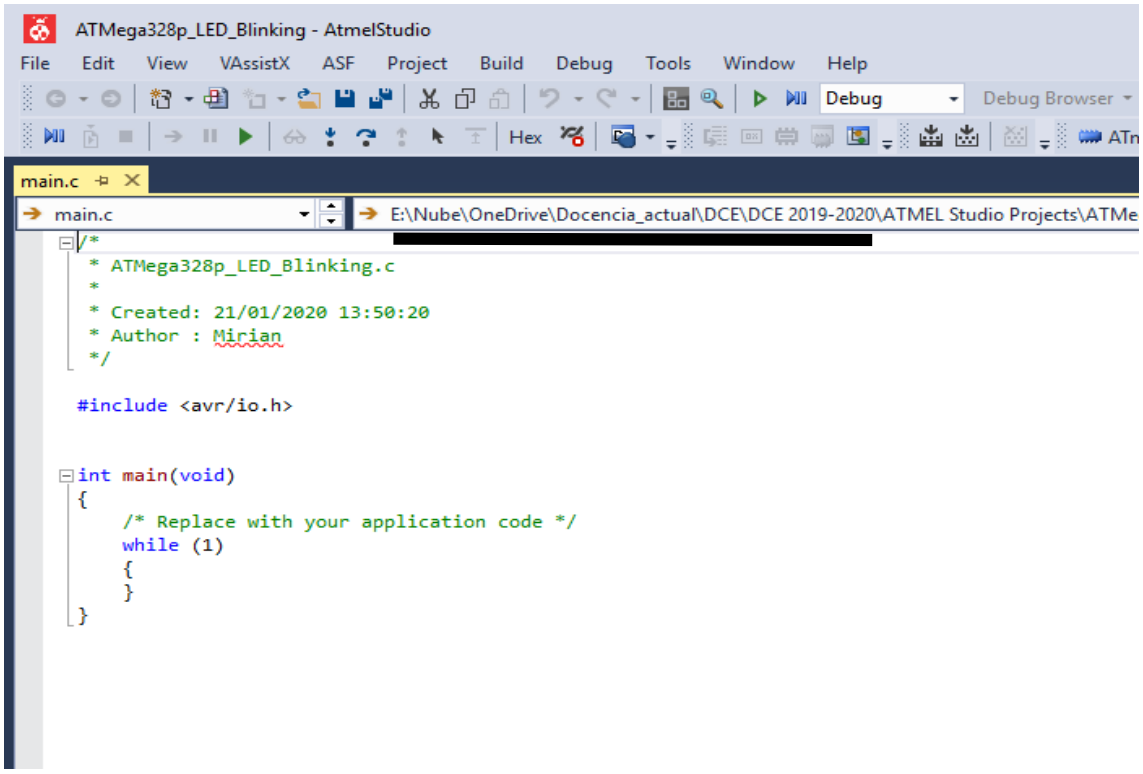


4. After click on OK, **Device Selection** window will pop up as shown in below figure. In that we can directly type device name to get required device from device list shown in below figure.

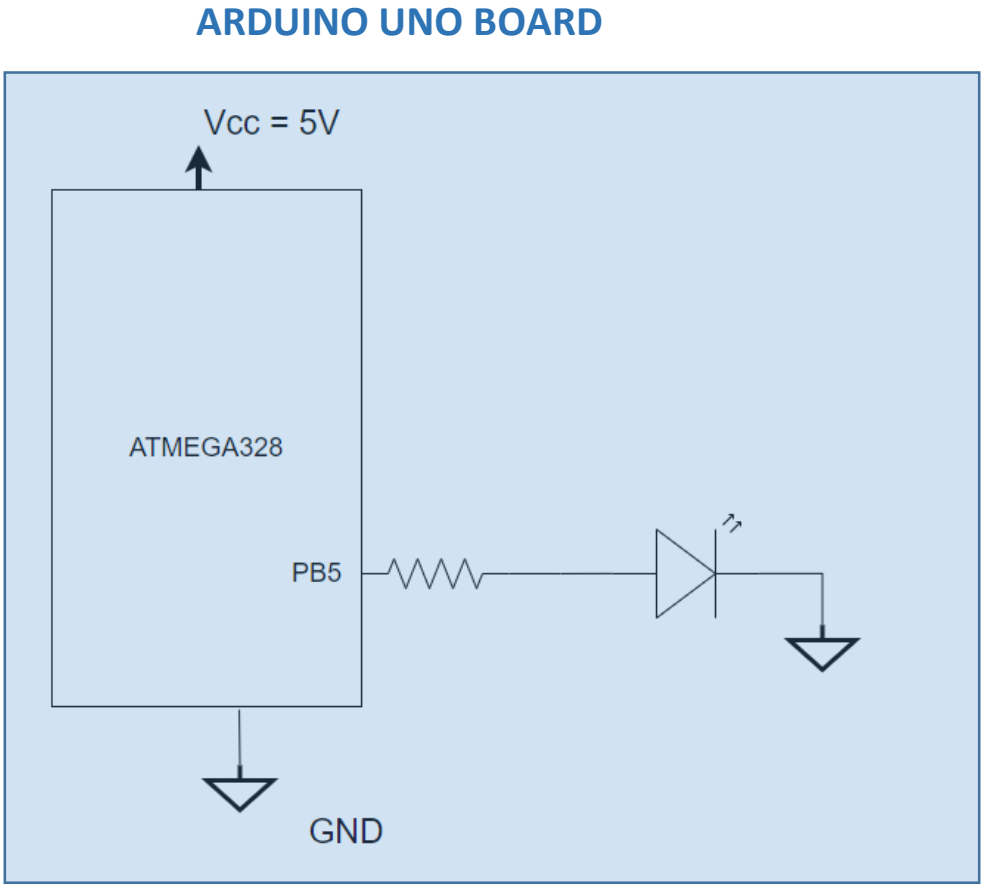
Click on device name and then click OK. Here we have selected ATmega328p microcontroller device.



5. Now Atmel studio creates a project and **main.c** file to write program for selected device as shown in below figure.



6. Now write a program. Here we are writing program for LED Blinking connected to a PORTB pin 5 of ATmega328P (LED connected to Pin 13 in Arduino)

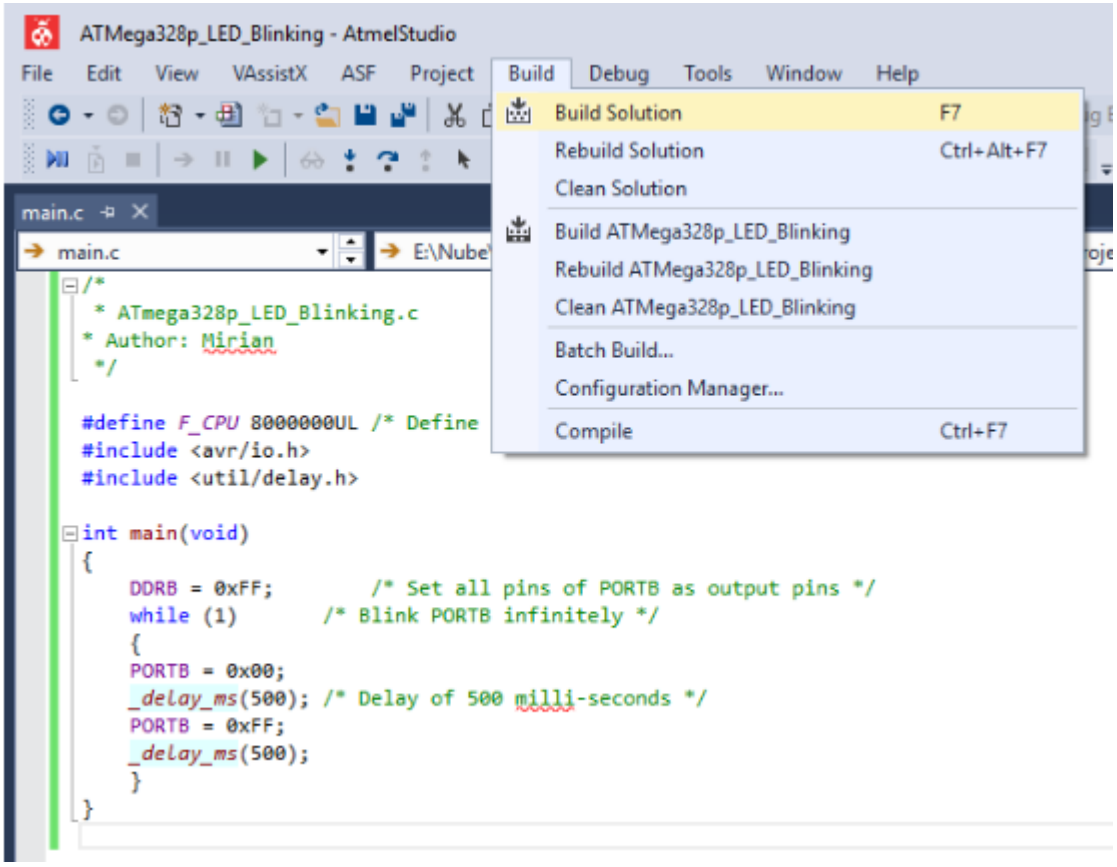


"BlinkLED_13.c" file

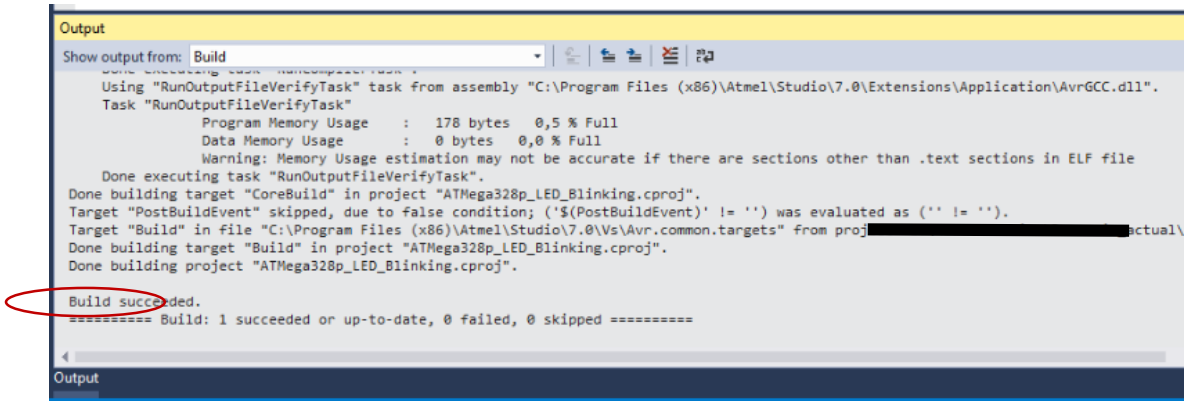
```
/**
      @file       BlinkLED_13.c
      @author     Mirian Cifredo
      @date       23/12/2021
      @brief      Blinking the LED connected to pin 13 (Arduino).
      @par        Description
                  Pin 13 in Arduino is connected to pin 5 in PORTB.
                  This information can be got from the picture showing
                  Arduino UNO <--> Atmega328p
*/

/// Define CPU frequency as 16 MHz
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void) {
/** ------------ SETUP ------------------ */
///   1) Set pin 5 in PORTB as output pin
      DDRB |=(1<<DDB5);

/** ------------ LOOP ------------------- */
    while (1) {
            /// 1) LED OFF by clearing the pin PB5
            PORTB &=~(1<<PORTB5);
            /// 2) Delay of 500 milli-seconds
            _delay_ms(500);
            /// 3) LED ON by setting the pin PB5
            PORTB |=(1<<PORTB5);
            /// 4) Delay of 500 milli-seconds
            _delay_ms(500);
    }
}
```

7. After writing program, save (Ctrl+S) the program and click on **Build Solution** from **Build** menu as shown in below figure.

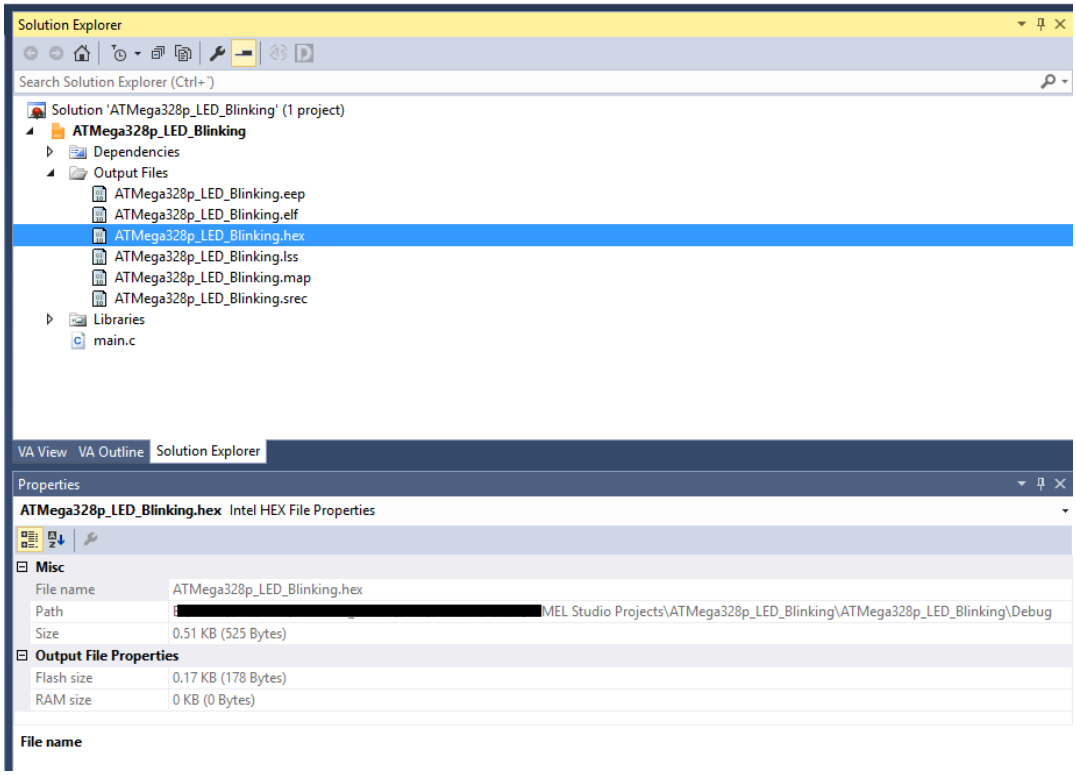
Also, we can use **F7** short key for **Build solution**.



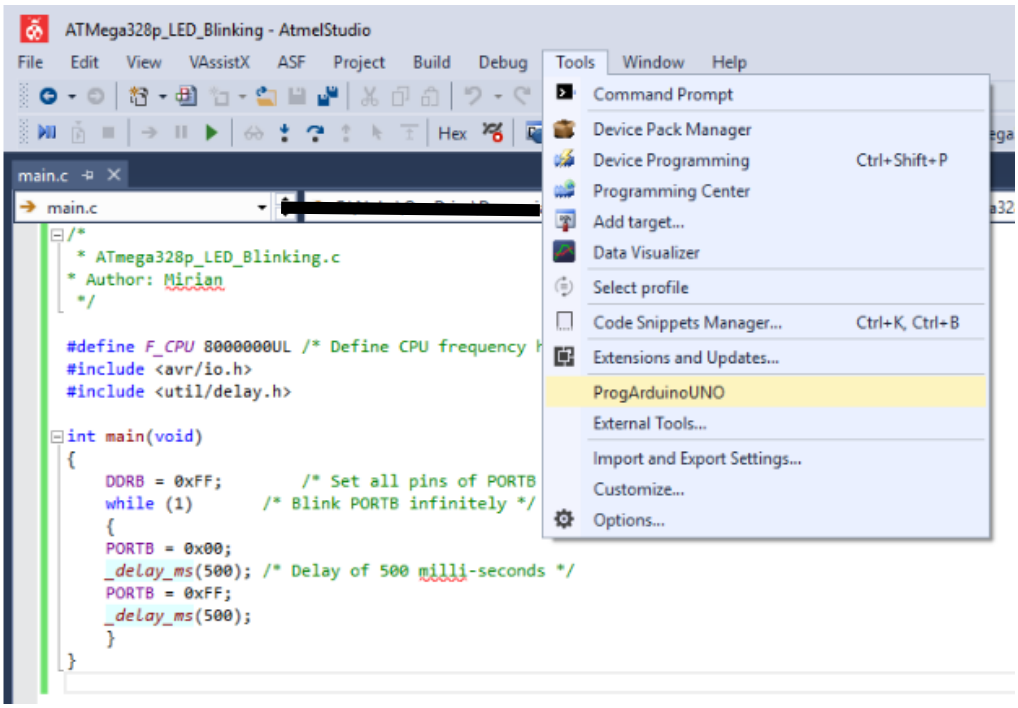
8. Now we can see build succeeded output in Output Window (lower left corner of window) as shown in below figure.



9. Now we can see the **.hex** file in Debug directory of the Main project directory.



10. Upload this hex file to ATmega microcontroller by means of the AVRDUDE based tool created for Atmel Studio. Previously connect Arduino Uno to USB port and check what Serial port was assigned, i.e. COM12 or COM7. Change the port number in the external tool if needed.



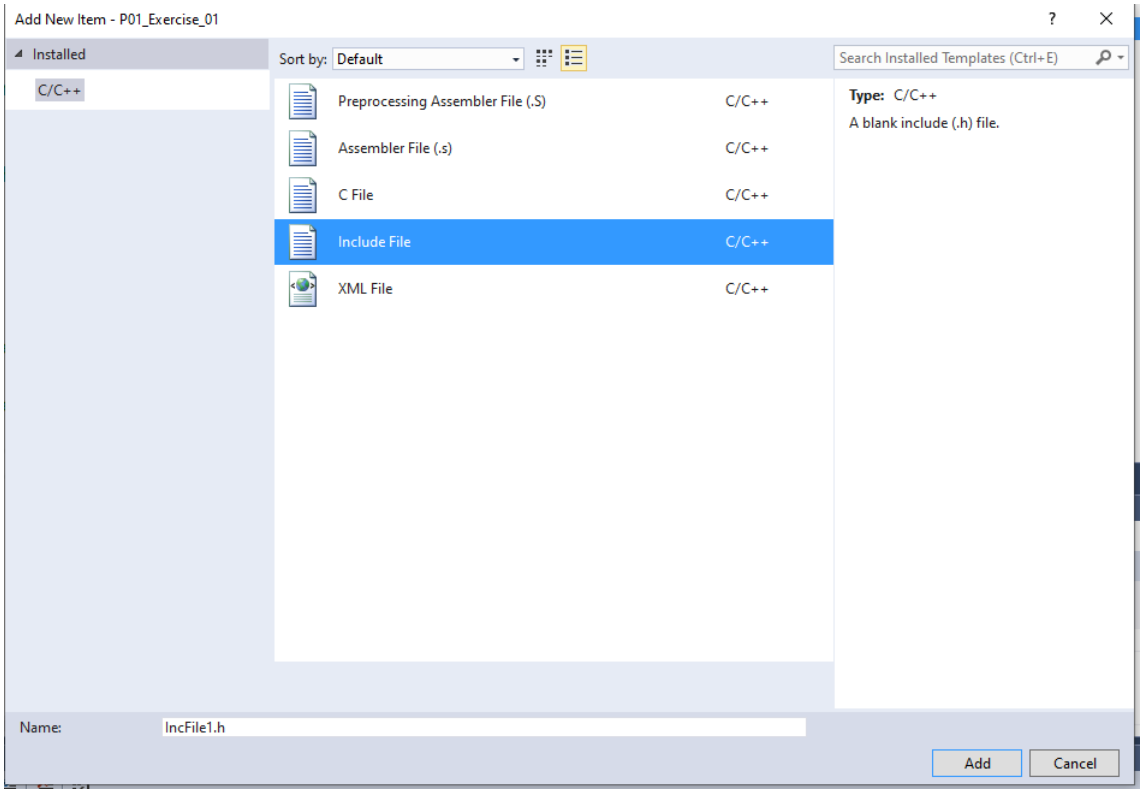
11. After uploading the above program to Arduino UNO, the built-in LED will start blinking.

EXERCISE_01

Objectives:

- To arrange the previous file in order to be more structured. Header files is going to be added to try doing a more compatible code.
- To organize the ATMEL projects to be tracked by a version control software.

1. In *Solution Explore* → *Add* → *New item*



Select *Include file* and type the name “PARAMETERS.h”.

2. Write the following code in the header file called “PARAMETERS.h”.

```
/**
      @file       PARAMETERS.h
      @author     Mirian Cifredo
      @date       23/12/2021
      @brief      Header file describing the parameters regarding to the
                  project.
      @par        Description
                  Allow modify the blinking time easily.
*/

#ifndef PARAMETERS_H_
#define PARAMETERS_H_

/// Time in ms the LED stays ON or OFF.
#define BLINK_TIME      500


#endif /* PORTS_H_ */
```

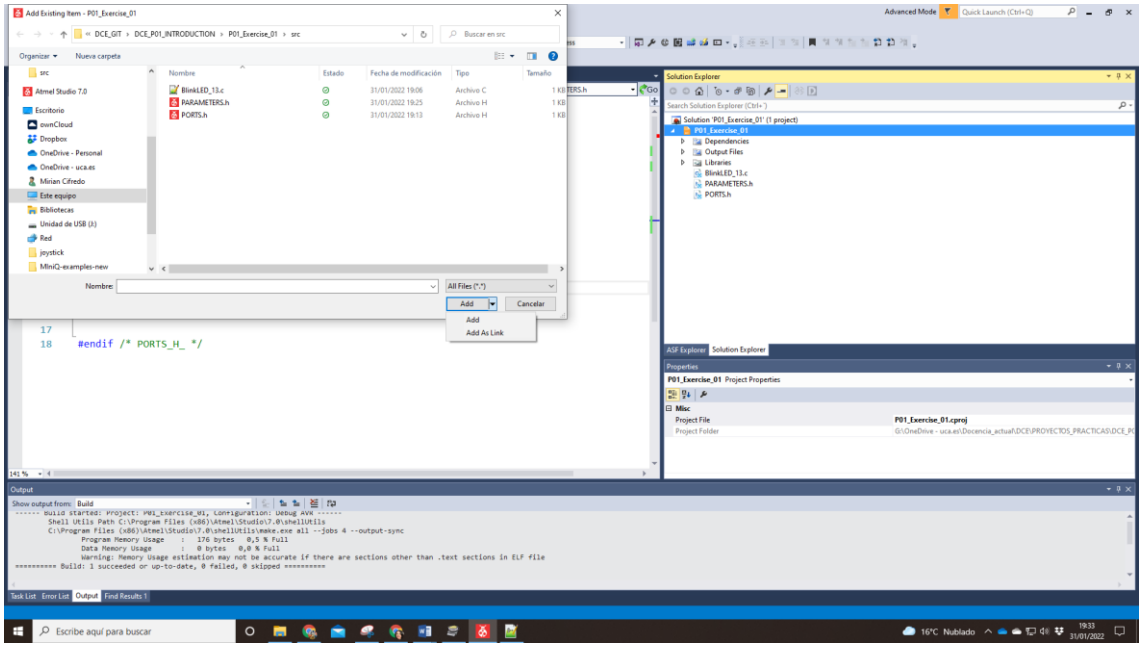
3. Modify the main file, “**BlinkLED_13**”, to use this custom header file.

```
   /// Define CPU frequency as 16 MHz
   #define F_CPU 16000000UL

   #include <avr/io.h>
   #include <util/delay.h>

   #include "PARAMETERS.h"


   int main(void) {
   /** ------------ SETUP ------------------    */
      ///   1) Set pin 5 in PORTB as output pin
      DDRB |=(1<<DDB5);

   /** ------------ LOOP ------------------
                  For blinking LED
      -------------------------------------*/
       while (1) {
            /// 1) LED OFF by clearing the pin PB5
            PORTB &=~(1<<PORTB5);
            /// 2) Delay of TIME milli-seconds
          _delay_ms(BLINK_TIME);
          /// 3) LED ON by setting the pin PB5
            PORTB |=(1<<PORTB5);
            /// 4) Delay of TIME milli-seconds
          _delay_ms(BLINK_TIME);
       }
   }
```

4. Create a **new folder to store all the projects** you are going to develop along the course. Name it “DCE_SURNAME_NAME”. (Replace by your name and surname).
5. Into this new folder, add a new one called “DCE_P01_INTRODUCTION”, and another into this called as the exercise, “P01_Exercise_01”.

6. In “P01_Exercise_01”, add one sub-folders called “src” and “doc”.
7. Copy the C and Header files into the folder “src”.
8. Now delete this files in the ATMEL STUDIO project.
9. The idea is only the source files and the documentation are in this folder and the project use them from that link. For achieving, select *Solution Explore* → *Add* → *Existing item…*
   Now, search the new path of the source codes.

10. Now, for adding again the source and header files to the project, select them and click on ***Add as Link*.**



11. Once the files are aggregated to the project, build it and check if the performance is right.