

# Introduction to Arduino IDE and getting started with the ESP32 microcontroller

## Part 1: Basics of the Arduino IDE

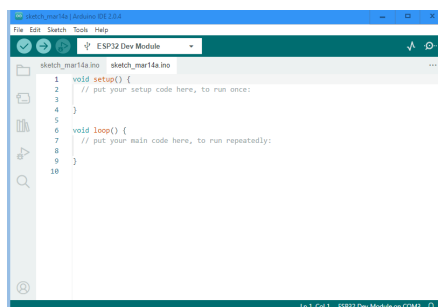
Dr Ian Grout  
Department of Electronic and Computer Engineering  
Faculty of Science and Engineering  
University of Limerick  
Limerick, V94 T9PX  
Ireland

Email: [Ian.Grout@ul.ie](mailto:Ian.Grout@ul.ie)



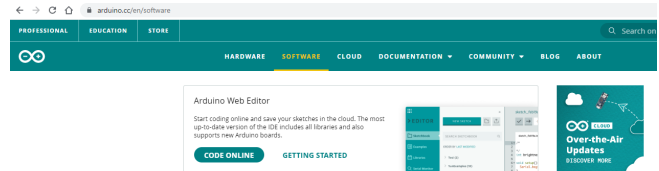
## Introduction

- In this part, the following activity will be to become familiar with the Arduino IDE:
  - Basics of the Arduino IDE:
    1. Getting started with the Arduino IDE.
    2. Using the ESP32 with the Arduino IDE.
    3. Walkthrough example: on-board LED blinking – a “Hello world” program.




# Downloading and installing the Arduino IDE

- Download the IDE from:
- <https://www.arduino.cc/en/software>
- Select the operating system and follow the installation instructions.
- **Note:** Before the ESP32 microcontroller can be used, a specific Arduino compatible microcontroller board must be installed (using the Board Manager).
- When a board is connected to the computer, the user must select the board to use and the **serial port** on the computer that the board is connected to (the **COM** port on a **Windows** platform).



## Downloads

**Arduino IDE 2.0.4**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**DOWNLOAD OPTIONS**

<b>Windows</b>	Win 10 and newer, 64 bits
<b>Windows</b>	64 bit installer
<b>Windows</b>	2019 file
<b>Linux</b>	AppImage 64 bits (2019-04)
<b>Linux</b>	2019 file 64 bits (2019-04)
<b>macOS</b>	Intel, 10.14 "Mojave" or newer, 64 bits
<b>macOS</b>	Apple Silicon, 11 "Big Sur" or newer, 64 bits

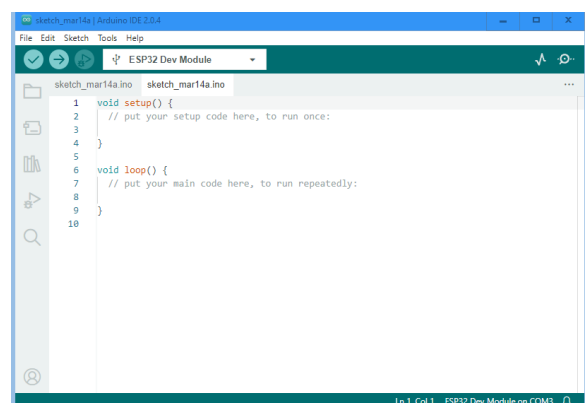
[Release Notes](#)

<https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>

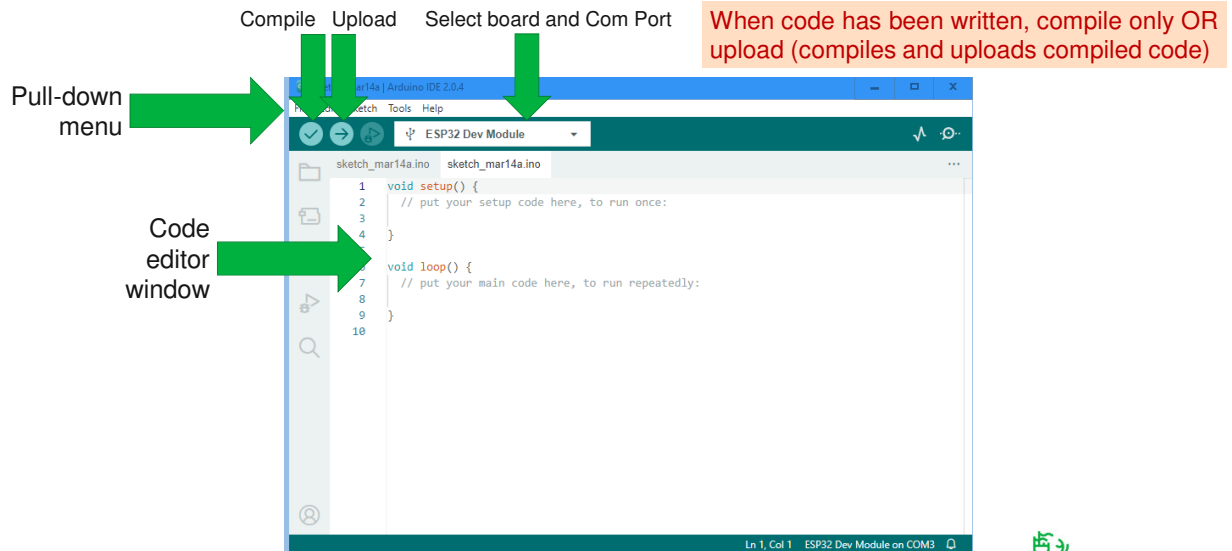


# Getting started with the Arduino IDE (1)

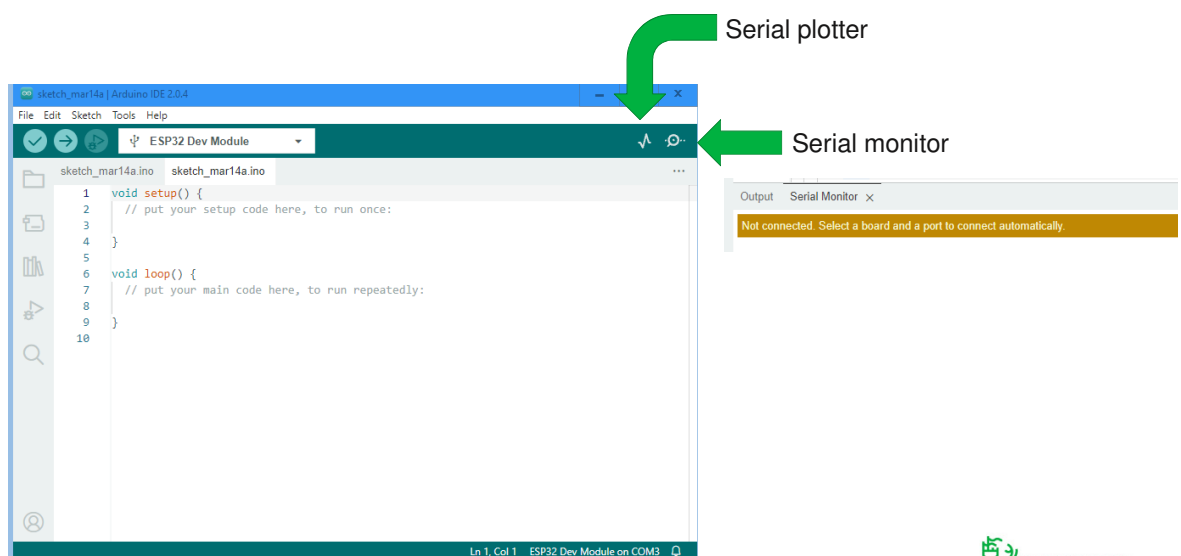
- Once the Arduino IDE has been successfully installed, it can be used.
- Arduino IDE 2 will be used in this webinar.
- The code to enter is contained in a **Sketch**.
- When a new **Sketch** is created, a template code is created and needs to be edited.
- The **Sketch** should be saved in an appropriate folder on the user computer.
- The **Sketch** consists of a **folder** and **.ino** files within the folder.
- A **Sketch** will contain one or more **.ino** files.



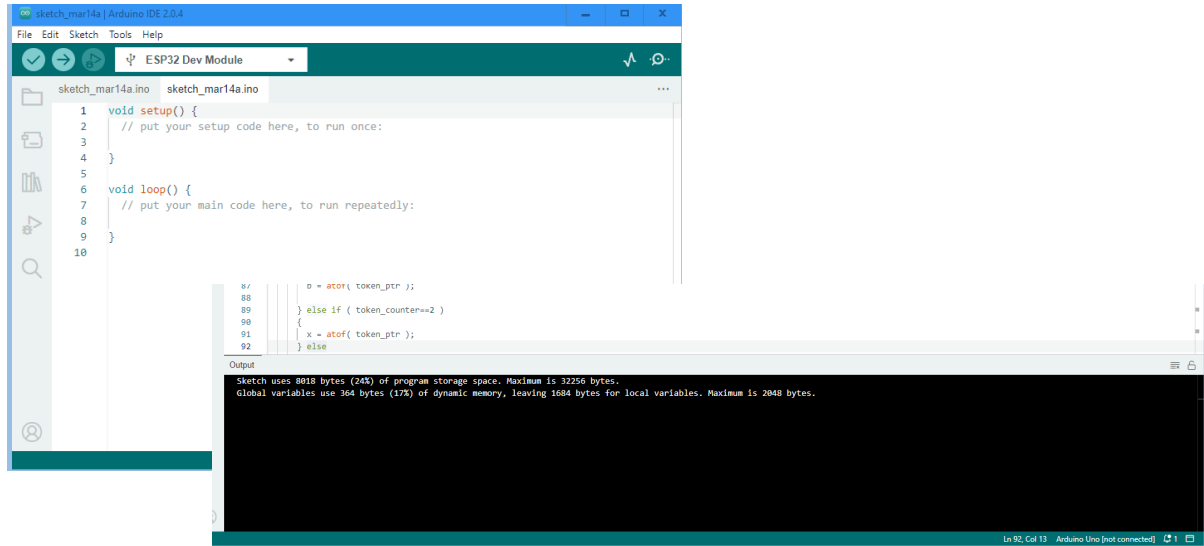
## Getting started with the Arduino IDE (2)



## Getting started with the Arduino IDE (3)



## Getting started with the Arduino IDE (4)



## Getting started with the Arduino IDE (5)

- Default sketch: **setup()** and **loop()** functions.
- These are to be edited and also other functions can be added to create the required program functionality.
- This is in the Arduino language.
- C code and also assembler code can be included if required.

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

```
void setup()
```

OR

```
void setup( void )
```

```
void loop()
```

OR

```
void loop( void )
```

## setup() function

```
void setup() {  
  // put your setup code here, to run once:  
}
```



```
void setup( void )  
{  
  // put your setup code here, to run once:  
}
```

## loop() function

```
void loop() {  
  // put your main code here, to run repeatedly:  
}
```



```
void loop( void )  
{  
  // put your main code here, to run repeatedly:  
}
```

## Downloading and installing the ESP32 board support (1)

- To use the ESP32 with the Arduino IDE, it is necessary to install and use the support for the ESP32 board.

- This a two-step process.

- Step 1** is to access the appropriate Arduino ESP32 board support.

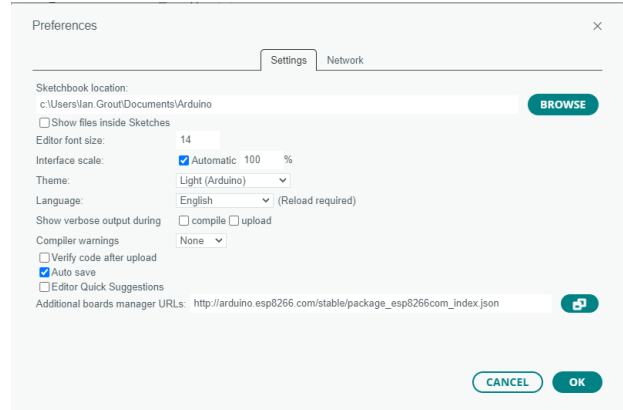
- Go to the Preferences option:

File -> Preferences...

- In the Additional boards manager URLs, enter:

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

- Press OK.



Note: there are different ESP32 boards can be installed and used. The identified library above is just one possibility.



## Downloading and installing the ESP32 board support (2)

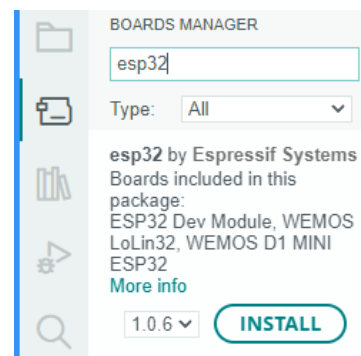
- Step 2** is to install the Arduino support for the ESP32 board.

- Select the Boards Manager from the icon menu on the left side of the IDE.

- Type in **esp32** into the search box and identify the board:

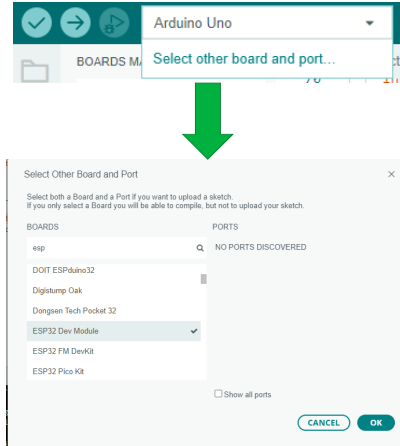
Esp32 by Espressif Systems

- Press **INSTALL**.



## Using the ESP32 board

- Connect the ESP32 board to a suitable USB port.
- Select the ESP32 board and Com Port that the board is connected to in the Arduino IDE.
- Select the board and COM port in the Arduino IDE:
  - **ESP32 Dev Module.**
  - **Specific PORT** identified.
- Press **OK**.



## On-board LED “Hello World” example (1)

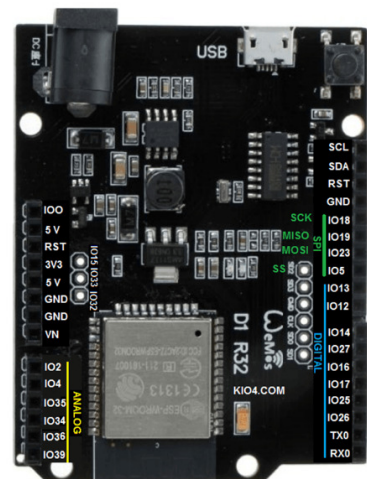
- Arduino sketch to turn the on-board LED ON and OFF.
- Set the LED pin status (HIGH or LOW) and toggle the LED.
- Use the software delay to create time delays between LED changes.

```
pinMode( ON_BOARD_LED, OUTPUT );

digitalWrite( ON_BOARD_LED, LOW );
digitalWrite( ON_BOARD_LED, HIGH );

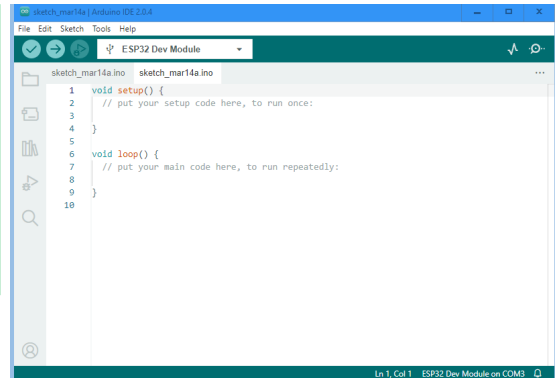
digitalWrite( ON_BOARD_LED, !digitalRead( ON_BOARD_LED ) );

delay( 1000 );
delay( 100 );
```



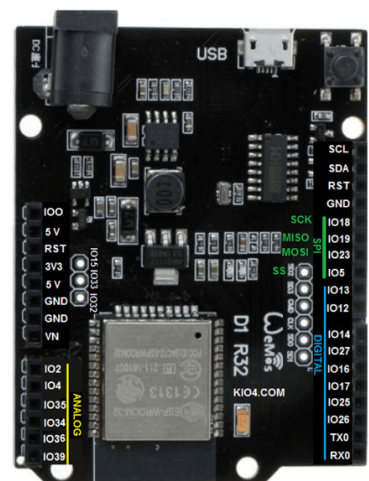
## Default Arduino sketch

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```



## On-board LED “Hello World” example (2)

```
#define ON_BOARD_LED 2  
  
void setup( void )  
{  
    pinMode( ON_BOARD_LED , OUTPUT );  
    digitalWrite( ON_BOARD_LED , LOW );  
  
    delay( 1000 );  
    digitalWrite( ON_BOARD_LED , HIGH );  
    delay( 1000 );  
    digitalWrite( ON_BOARD_LED , LOW );  
    delay( 1000 );  
}  
  
void loop( void )  
{  
    digitalWrite( ON_BOARD_LED , !digitalRead( ON_BOARD_LED ) );  
    delay( 1000 );  
  
    for ( uint8_t i=0; i<10; i++ )  
    {  
        digitalWrite( ON_BOARD_LED , !digitalRead( ON_BOARD_LED ) );  
        delay( 100 );  
    }  
  
    delay( 1000 );  
}
```





## Arduino sketch template to consider using ...

```
/* -----  
 * Arduino sketch name: xxxx  
 * File name:          xxxx  
 * Author name:        xxxx  
 * Last updated on:    xxxx  
 * ----- */  
  
/* -----  
 * Code required before setup function  
 * ----- */  
  
/* -----  
 * void setup( void )  
 * ----- */  
  
void setup( void )  
{  
    // put your setup code here, to run once:  
}  
  
/* -----  
 * void loop( void )  
 * ----- */  
  
void loop( void )  
{  
    // put your main code here, to run repeatedly:  
}  
  
/* -----  
 * End of file  
 * ----- */
```

```
void setup( void )  
{  
    // put your setup code here, to run once:  
}
```

```
void loop( void )  
{  
    // put your main code here, to run repeatedly:  
}
```

## A couple of initial thoughts

- This webinar is an introduction to get started.
- The first step is to see what the design process is.
- The second step is to understand the general ideas behind creating project.
- The third step is to develop a more detailed understanding of the software and hardware.

*We are interested in steps 1 and 2.*

*Step 3 is something that comes with further learning, understanding, and practice.*



*In this webinar, don't get too concerned about the detailed coding (in step 3).*

*Get confidence first.*

*Get the code working first and then go back to the code to gain a better understanding.*



## Exercise

- Using the **Hello World** example, in the provided Arduino sketch **part\_1**, run the code as provided.
- Vary the time delay in the **loop()** function and observe the effect by viewing the on-board LED.

Any questions?