



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 18: Microcontroller Development Boards

CONCEPTS COVERED

- ❑ Introduction to microcontroller boards
- ❑ Introduction to the STM32F401 Board
- ❑ Introduction to the Arduino UNO Board



Introduction

- What is a microcontroller board?
 - It is a stand-alone development board containing microcontroller, memory and other necessary peripheral I/O components.
 - Used for fast development of embedded applications.
 - It is a printed-circuit-board (PCB) containing all the components required for experimentation.

Embedded System Development

- Involves two major components:
 - Development board -- HARDWARE
 - Integrated Development Environment (IDE) – SOFTWARE
- For a development board, we need to decide on the following features:
 - Bus type
 - Processor type
 - Memory
 - Number of ports and their types
 - Operating environment

Some Common Development Boards

- STM32F401 Nucleo
 - Based on ARM Cortex-M4, 96KB SRAM, 512KB Flash
 - 84MHz clock
- Arduino Uno
 - Open-source microcontroller board based on Microchip ATmega328P, 2KB SRAM, 32KB Flash, 1KB EEPROM
 - 16MHz clock
- Raspberry Pi 3 Model B
 - Quad-core 1.2GHz Broadcom 64-bit CPU
 - 1GB RAM, wireless LAN, Bluetooth, Ethernet, USB, etc.
- PIC 16F877A
 - 8-32KB Flash, 10-80MHz clock

In this course ...

- We shall be demonstrating concepts using the following boards:
 - STM32F401 Nucleo
 - Arduino Uno
- Point to note:
 - Process of interfacing and software development for the others boards will be quite similar.
 - Important to learn the concepts using one or two boards.
- Strongly recommended:
 - Purchase one of the two boards and perform the experiments hands-on.
 - Best way to learn the subject.

About STM32F401 Nucleo Development Board

- Developed by ST Microelectronics.
- CPU – ARM Cortex® M4
- 512 KB Flash Memory (Programmable) and 96-KB SRAM
- USB 2.0 type A to mini B
- mbed-enabled (mbed.org)
- Support of wide choice of Integrated Development Environments (IDEs) including IAR™, ARM® Keil®, GCC-based IDEs

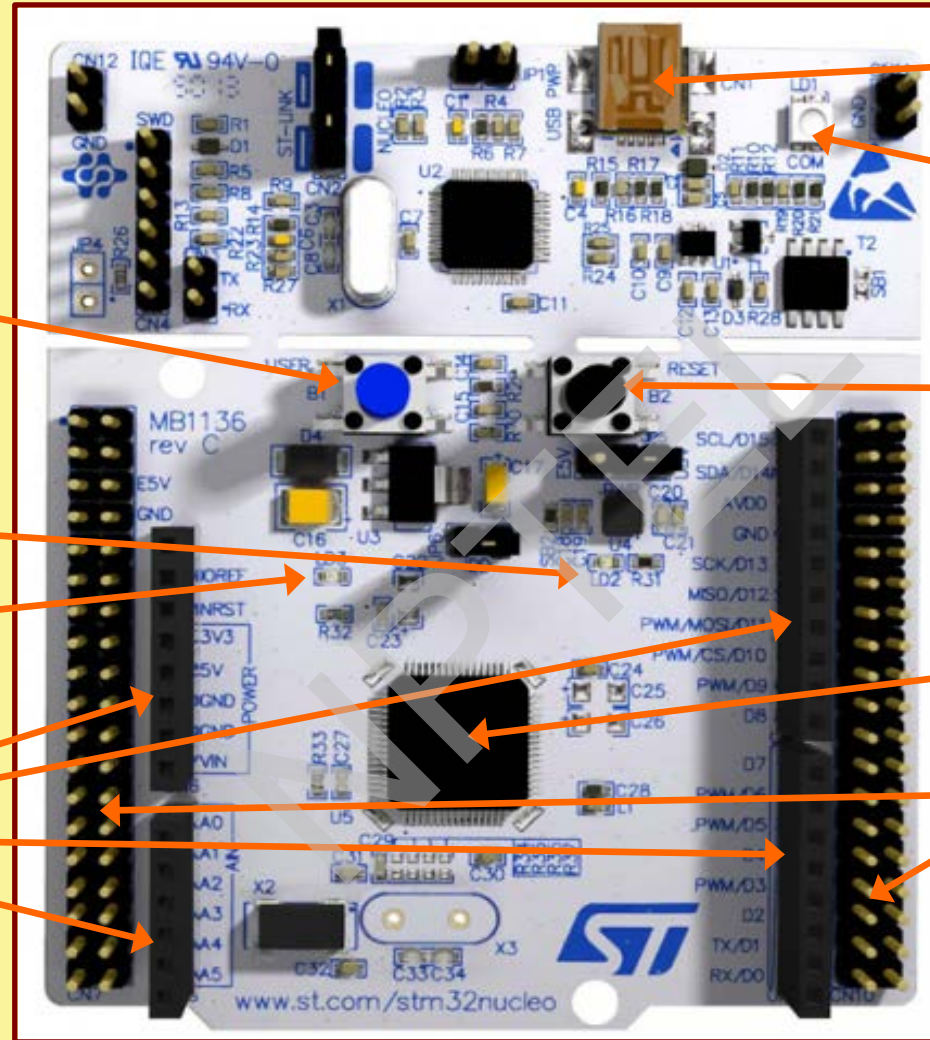


User Button

Green LED
(test)

Red LED
(power)

Arduino
Connector



USB: Type-A
to Mini-B

Red/Green
LED (com)

Reset Button

STM32
Microcontroller

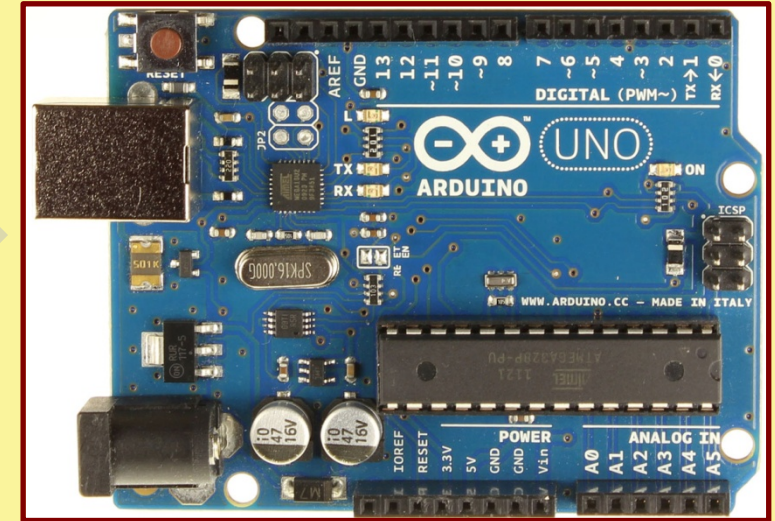
Port/Signal
Connector

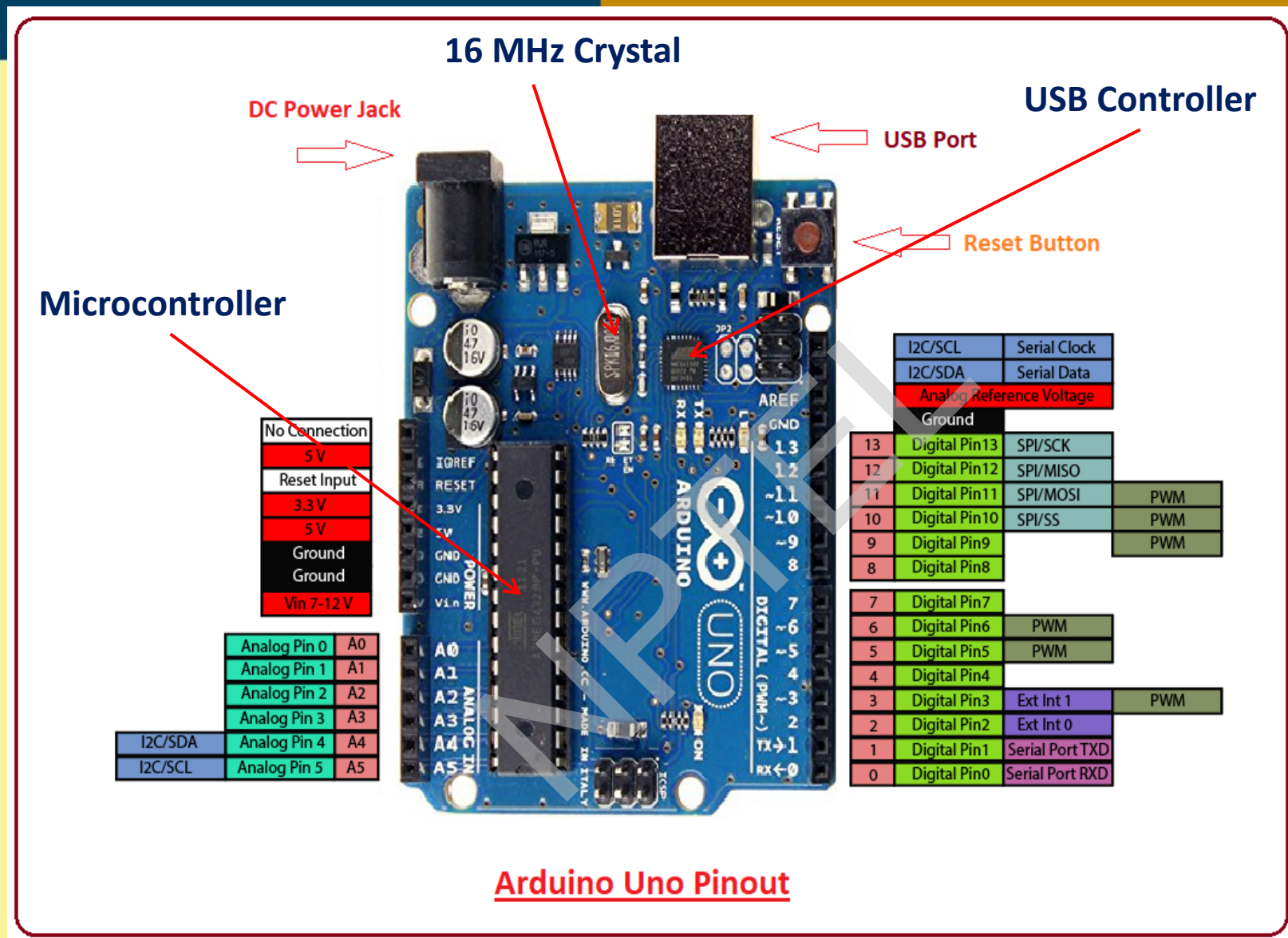
- Some points to note:

- There are several digital I/O ports, PWM ports, analog input ports.
- The digital I/O port lines can also be used as interrupt inputs.
- Application development through embedded-C programs:
 - Connect the board to a desktop/laptop.
 - Write the program in C using necessary driver libraries.
 - Use an online compiler to compile the code.
 - Download the code to on-board flash memory.

About Arduino Uno Development Board

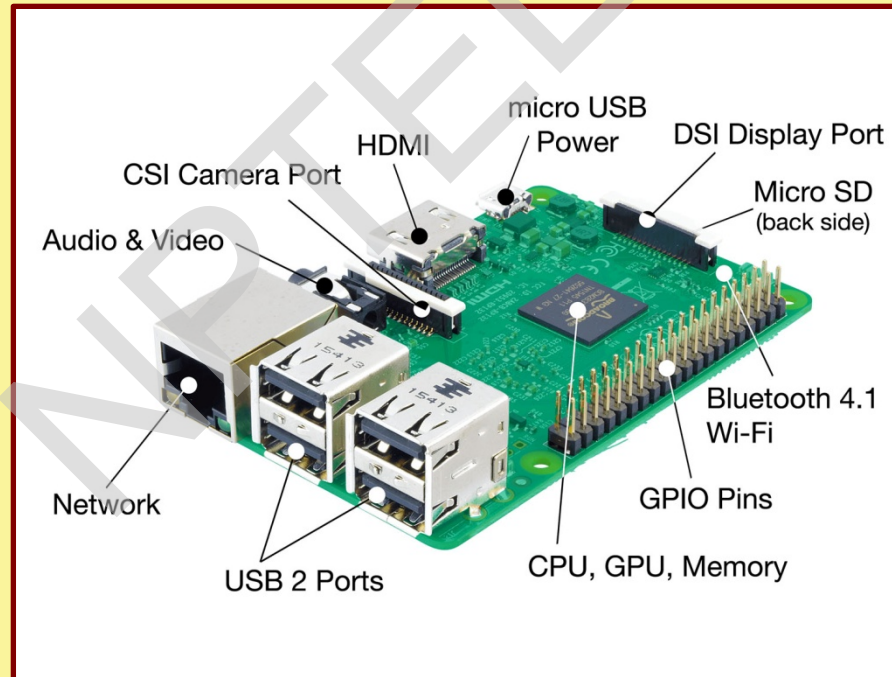
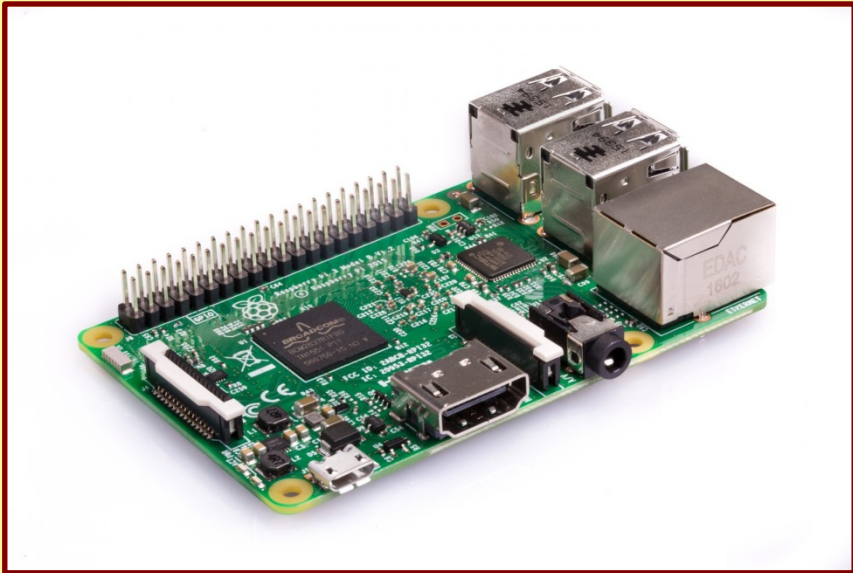
- Based on ATmega328 microcontroller.
- There are 14 digital I/O pins, and 6 analog I/O pins.
- It has a flash memory of 32KB.
- It provides SRAM of 2 KB.
- It has EEPROM of 1KB.
- The clock speed is 16MHz.



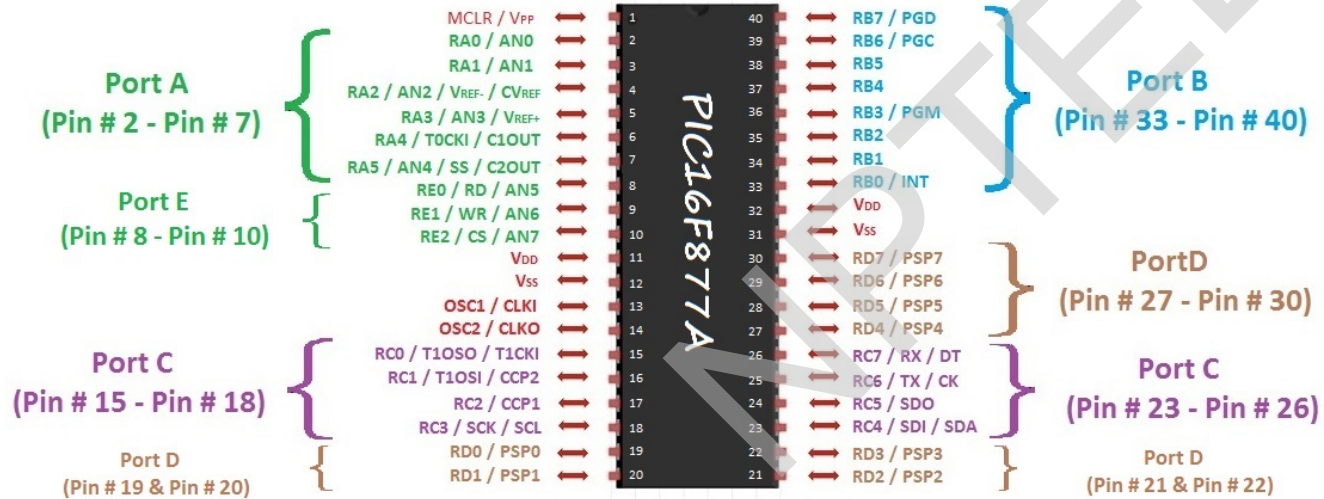


Some Other Development Boards

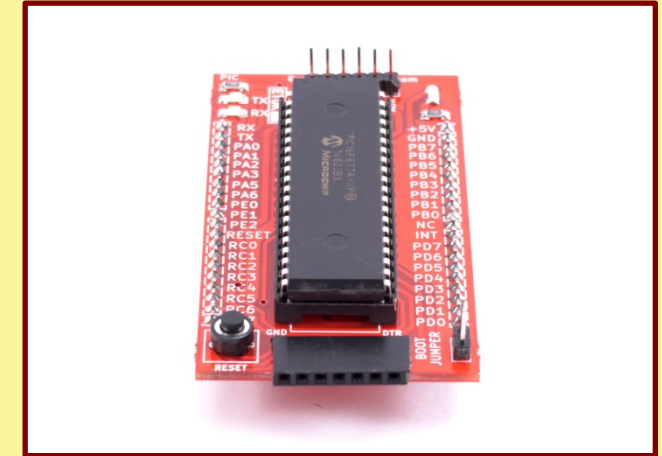
- Raspberry Pi 3 Model B



- PIC 16F877A



PIC16F877A Pin Diagram





NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 19: Mbed C Programming Environment

CONCEPTS COVERED

- ❑ Introduction to the Mbed C environment
- ❑ Steps required to create, compile and run the programs



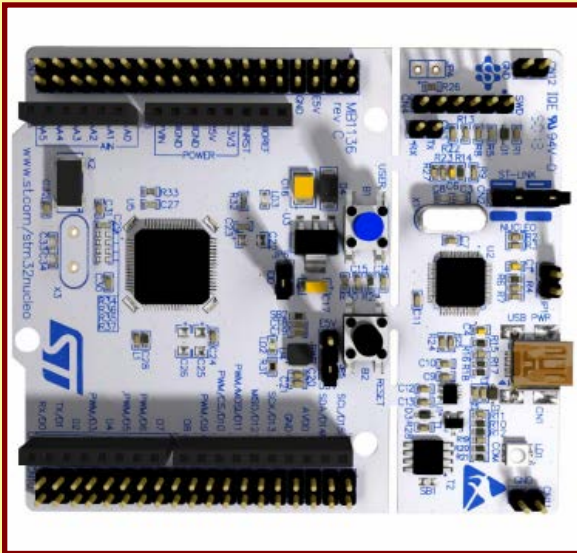
What is Mbed C?

- It is an embedded software development platform.
 - Users can develop the software using C using microcontroller board specific library.
 - The program is compiled and downloaded onto the development board.
- We shall explain the Mbed-C platform using the example of STM32F401 Nucleo ARM Development Board.

Requirements to Begin

- STM32F401 development board and driver
- USB mini to USB Type B connector
- Development environment (IDEs or mbed account).

The development environment runs on a desktop / laptop.



How to start?

- During the explanation / demonstration later in this course, we will use the online compiler at <http://developer.mbed.org> to compile our projects.
- We can write our program in C or Python.
 - The STM32 driver must first be installed on the desktop/laptop.
 - We will demonstrate software development using C.

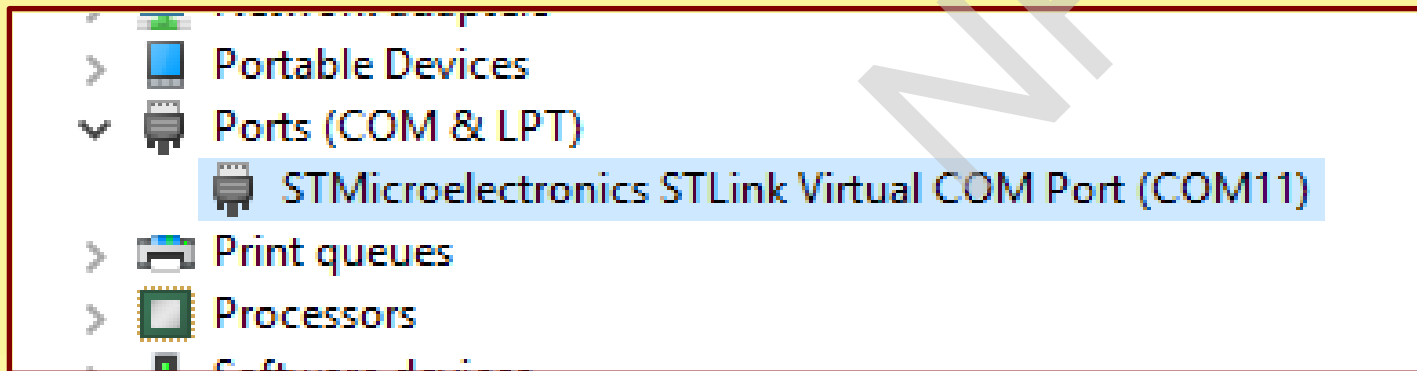
Installation of STM32 Driver

- To work with STM32 kit we need to install *STM32 driver*.
 - STM driver can be downloaded from:
<http://www.st.com/en/development-tools/stsw-link009.html#getsoftware-scroll>
 - For downloading you may need to provide basic details: e-mail, name etc.
 - Driver will be downloaded as a ZIP bundle.
 - Extract the downloaded zip file and install *dpinst_x86.exe* for 32-bit machine or *dpinst_x64.exe* for 64-bit machine.

Checking the Installation

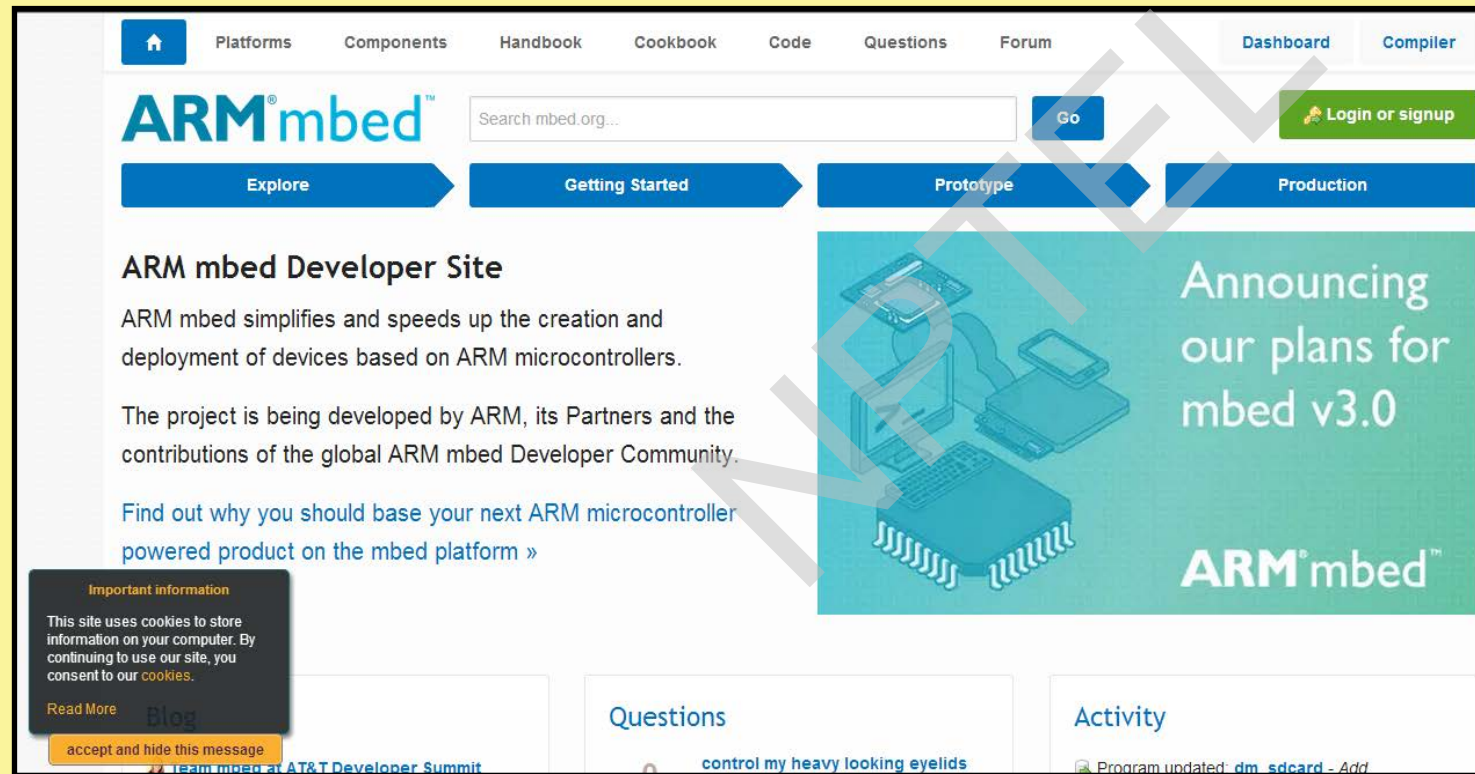
- Connect the STM32 kit to the desktop/laptop.
- Go to the device manager.
- Go to Ports (COM & LPT).
- Check which port number is used by STM kit.

In the following example, COM11 port is used.

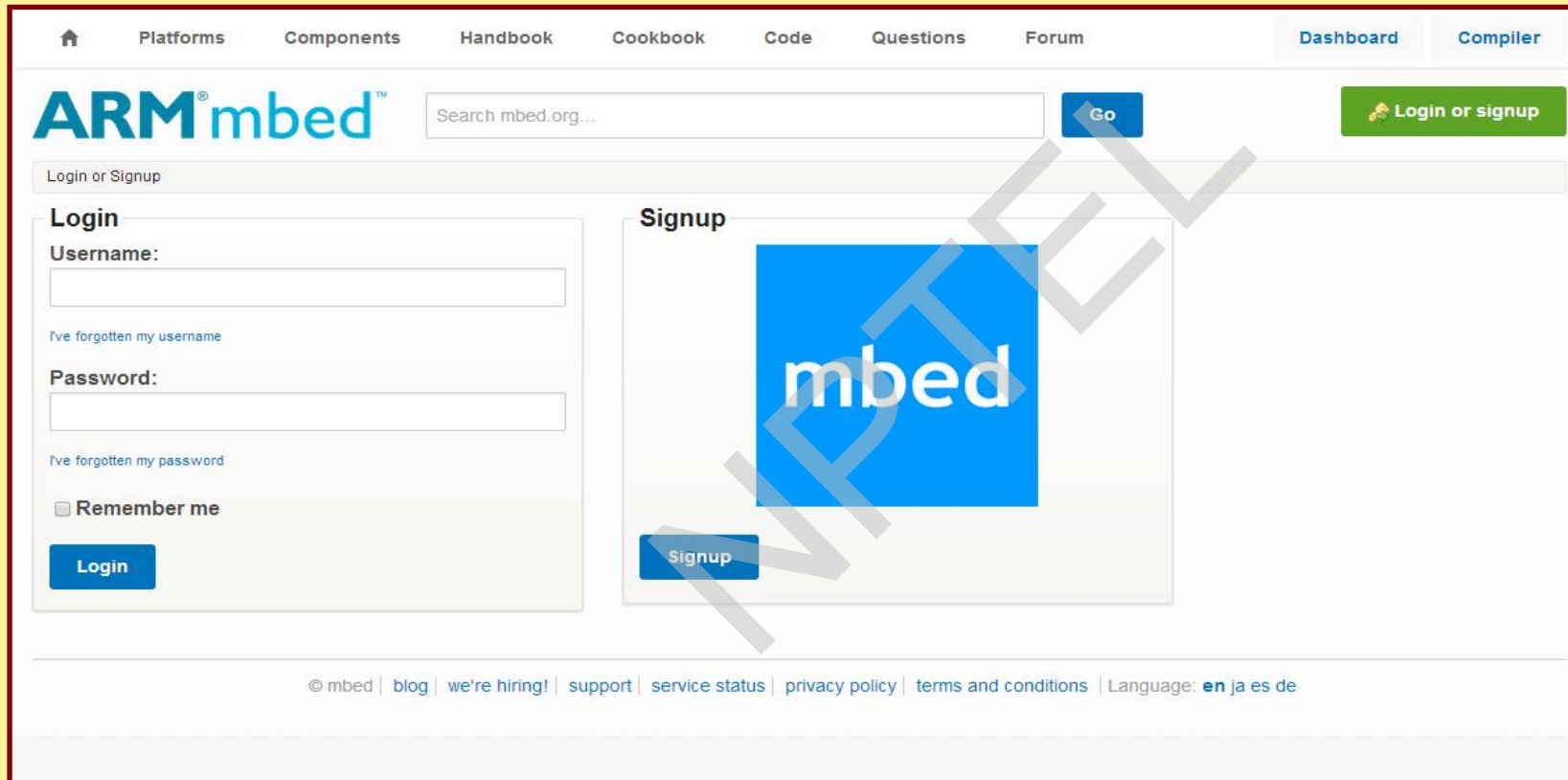


Steps to be Followed

- STEP 1 – Go to <http://developer.mbed.org>

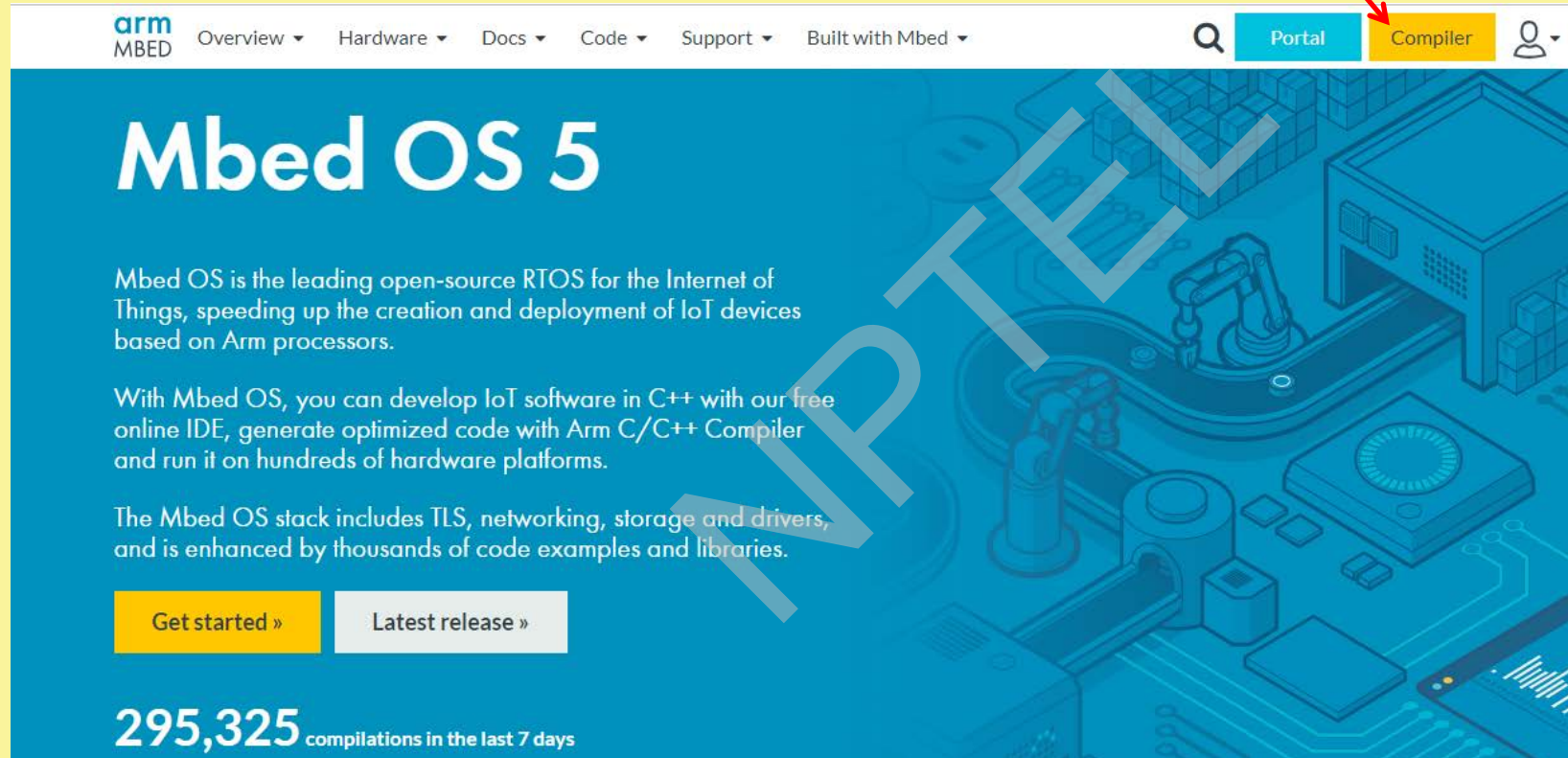


- STEP 2 – User Registration



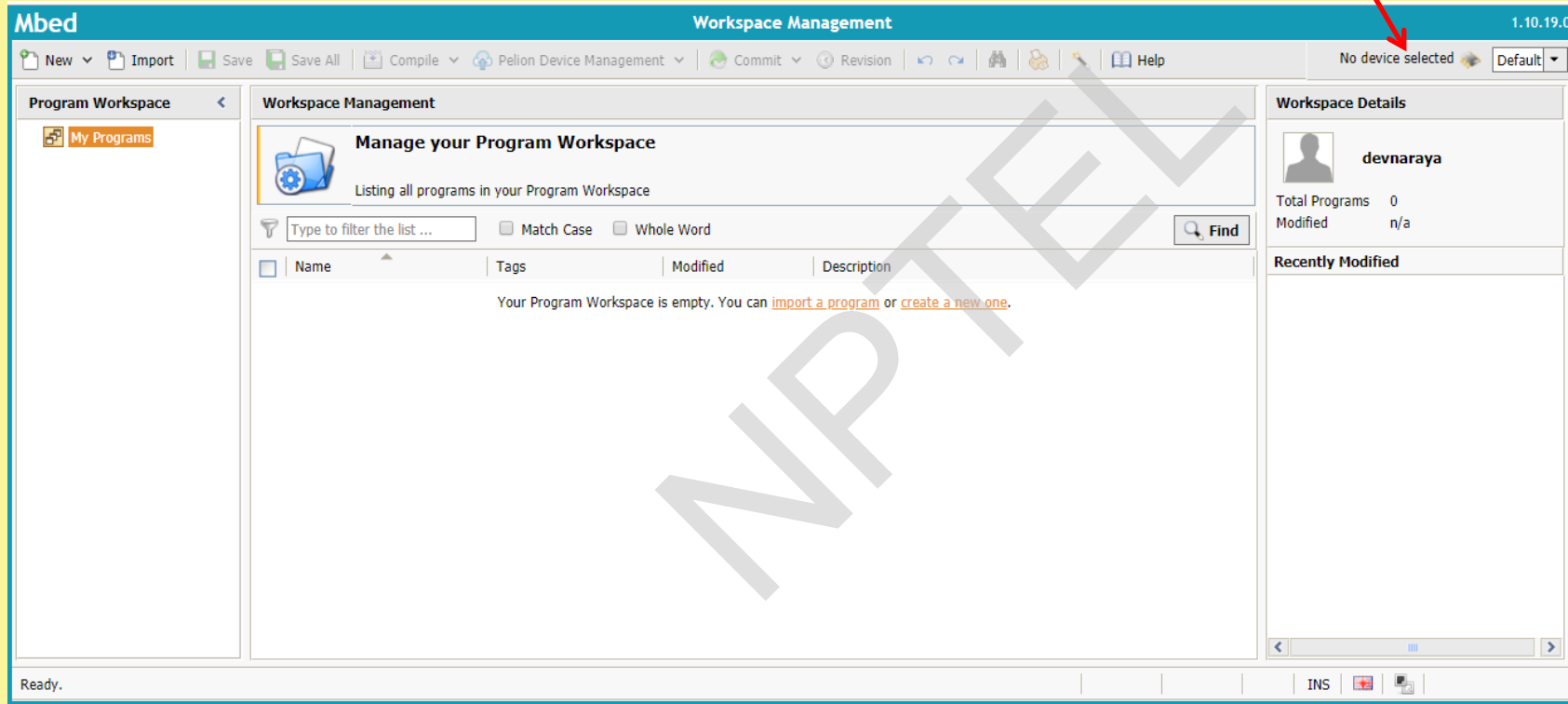
The image shows the ARM mbed website's login and signup interface. At the top, there is a navigation bar with links: Home, Platforms, Components, Handbook, Cookbook, Code, Questions, Forum, Dashboard, and Compiler. Below this is the ARM mbed logo and a search bar with the placeholder text "Search mbed.org...". To the right of the search bar is a "Go" button and a green "Login or signup" button. The main content area is divided into two sections: "Login" and "Signup". The "Login" section has fields for "Username:" and "Password:", with links for "I've forgotten my username" and "I've forgotten my password", a "Remember me" checkbox, and a "Login" button. The "Signup" section features a large blue square with the word "mbed" in white, a "Signup" button, and a "Go" button. At the bottom, there is a footer with links: © mbed, blog, we're hiring!, support, service status, privacy policy, terms and conditions, and Language: en ja es de.

- STEP 3 – Go to *Compiler*

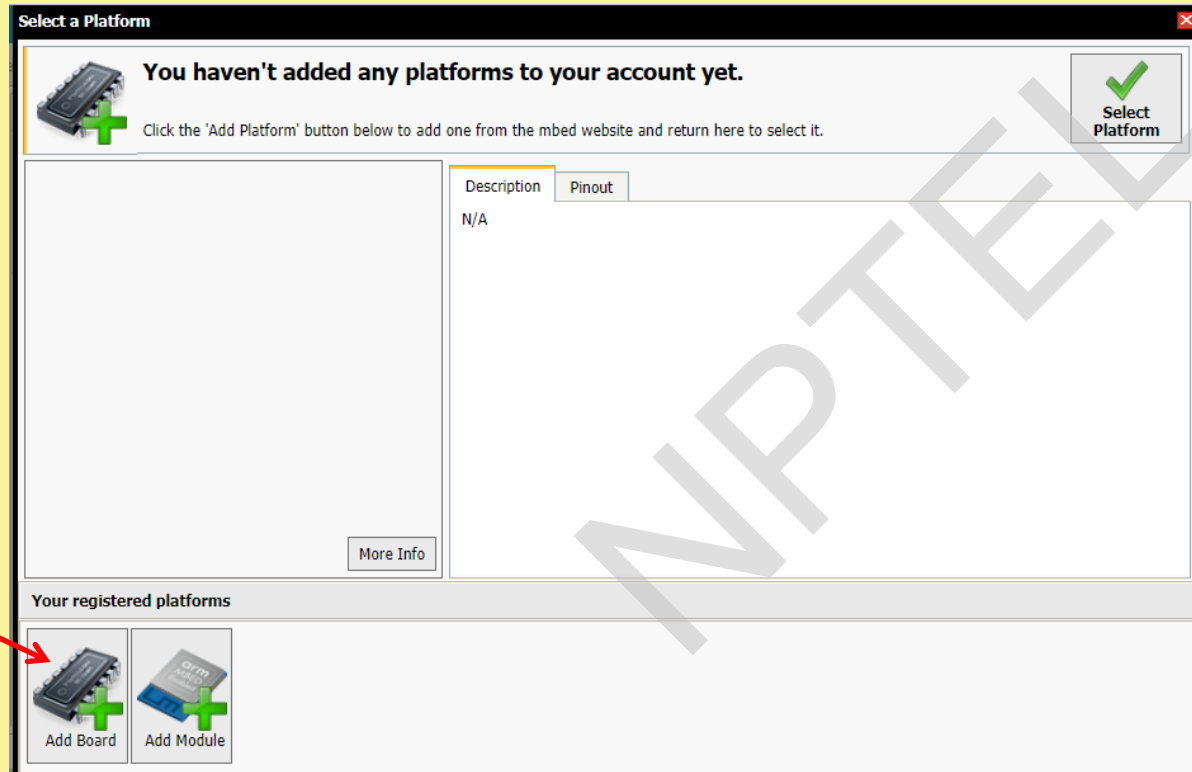


The screenshot shows the Arm Mbed OS 5 website. The top navigation bar includes links for Overview, Hardware, Docs, Code, Support, and Built with Mbed. A red arrow points to the 'Compiler' button, which is highlighted in yellow. The main content area features the title 'Mbed OS 5' and a description: 'Mbed OS is the leading open-source RTOS for the Internet of Things, speeding up the creation and deployment of IoT devices based on Arm processors.' Below this, it states: 'With Mbed OS, you can develop IoT software in C++ with our free online IDE, generate optimized code with Arm C/C++ Compiler and run it on hundreds of hardware platforms.' The Mbed OS stack is also described: 'The Mbed OS stack includes TLS, networking, storage and drivers, and is enhanced by thousands of code examples and libraries.' At the bottom, there are two buttons: 'Get started »' and 'Latest release »'. A statistic at the bottom left shows '295,325 compilations in the last 7 days'.

- STEP 4 – Select *Device*



- STEP 5 – Click on *Add Board*



- STEP 6 – Select board and click on *Add to Compiler*

Filter

Mbed Enabled

Mbed OS support

Target vendor

Platform vendor

Showing 43 of 150 (Show all)



NUCLEO-F103RB

- Cortex-M3, 72MHz
- 128KB Flash, 64KB SRAM
- UART, USB



NUCLEO-F302RS

- Cortex-M3 + MPU, 72MHz
- 64KB Flash, 16KB SRAM
- DAC, DMA, I2C, CAN, USB



NUCLEO-L152RE

- Cortex-M3, 32MHz
- 512KB Flash, 32KB SRAM
- LED, DAC, DMA, I2C, USB



NUCLEO-L053R8

- Cortex-M0, 32MHz
- 64KB Flash, 8KB SRAM
- LED, DAC, USB



NUCLEO-F401RE

- Cortex-M4 + MPU, 84MHz
- 512KB Flash, 16KB SRAM
- USB, OTG, PS, SD-IO



NUCLEO-F030R8

- Cortex-M0, 48MHz
- 64KB Flash, 8KB SRAM



NUCLEO-F072R8

- Cortex-M0, 48MHz
- 128KB Flash, 16KB SRAM
- DAC, CAN, USB



NUCLEO-F204R8

- Cortex-M4 + MPU, 72MHz
- 64KB Flash, 16KB SRAM
- DAC, DMA, I2C, CAN



NUCLEO-F411RE

- Cortex-M4 + MPU, 100MHz
- 512KB Flash, 128KB SRAM
- USB, OTG, PS, SD-IO



NUCLEO-F429ZI

- Cortex-M4 + MPU, 100MHz
- 512KB Flash, 128KB SRAM
- USB, OTG, PS, SD-IO



NUCLEO-F439ZI

- Cortex-M4 + MPU, 100MHz
- 512KB Flash, 128KB SRAM
- USB, OTG, PS, SD-IO



NUCLEO-F446ZE

- Cortex-M4 + MPU, 100MHz
- 512KB Flash, 128KB SRAM
- USB, OTG, PS, SD-IO

NUCLEO-F401RE

Affordable and flexible platform to ease prototyping using a STM32F401RET6 microcontroller.



The STM32 Nucleo board provides an affordable and flexible way for users to try out new ideas and build prototypes with any STM32 microcontroller line, choosing from the various combinations of performance, power consumption and features.

The Arduino™ connectivity support and ST Morpho headers make it easy to expand the functionality of the STM32 Nucleo open development platform with a wide choice of specialized shields.

The STM32 Nucleo board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

To compile a program for this board using Mbed CLI, use `nucleo_f401re` as the target name.

Board Partner



life.augmented

ST

A world leader in providing the semiconductor solutions that make a positive contribution to people's lives, both today and in the future.

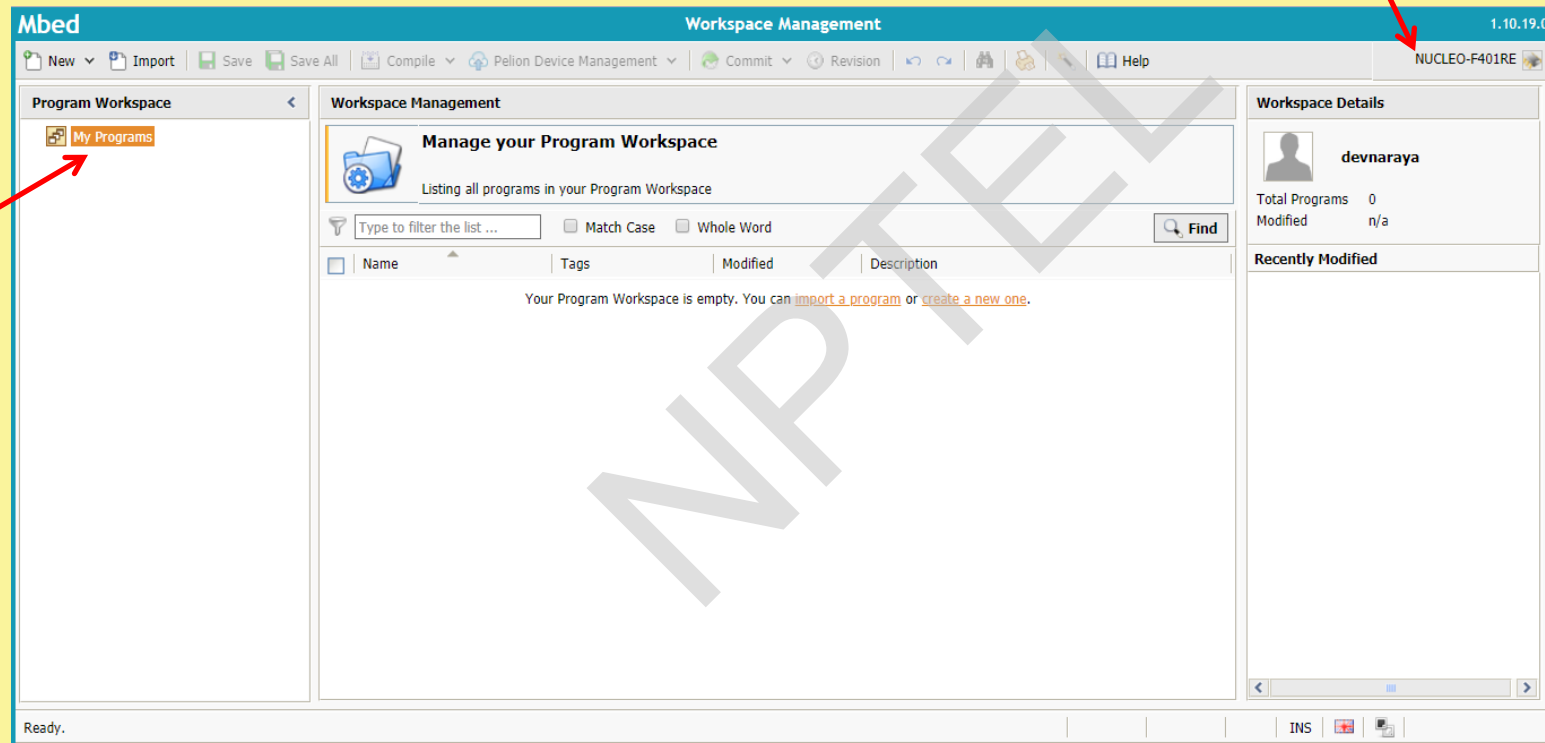
[Add to your Mbed Compiler](#)

[Buy Now](#)

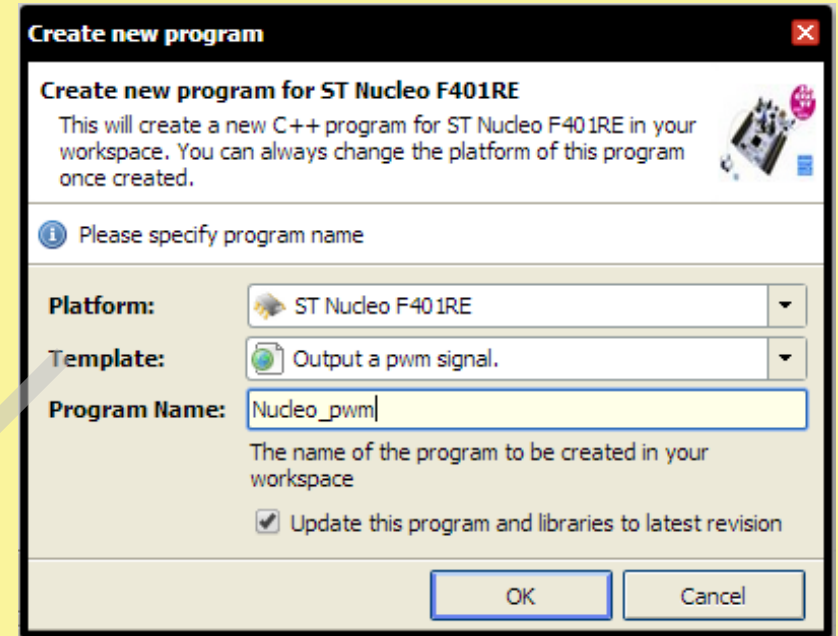
Overview

- 1. Overview
- 2. Microcontroller features
- 3. Board features
- 4. Board pinout
- 5. Supported shields
- 6. Getting started
- 7. Technical references
- 8. Known limitations
- 9. Tips and Tricks

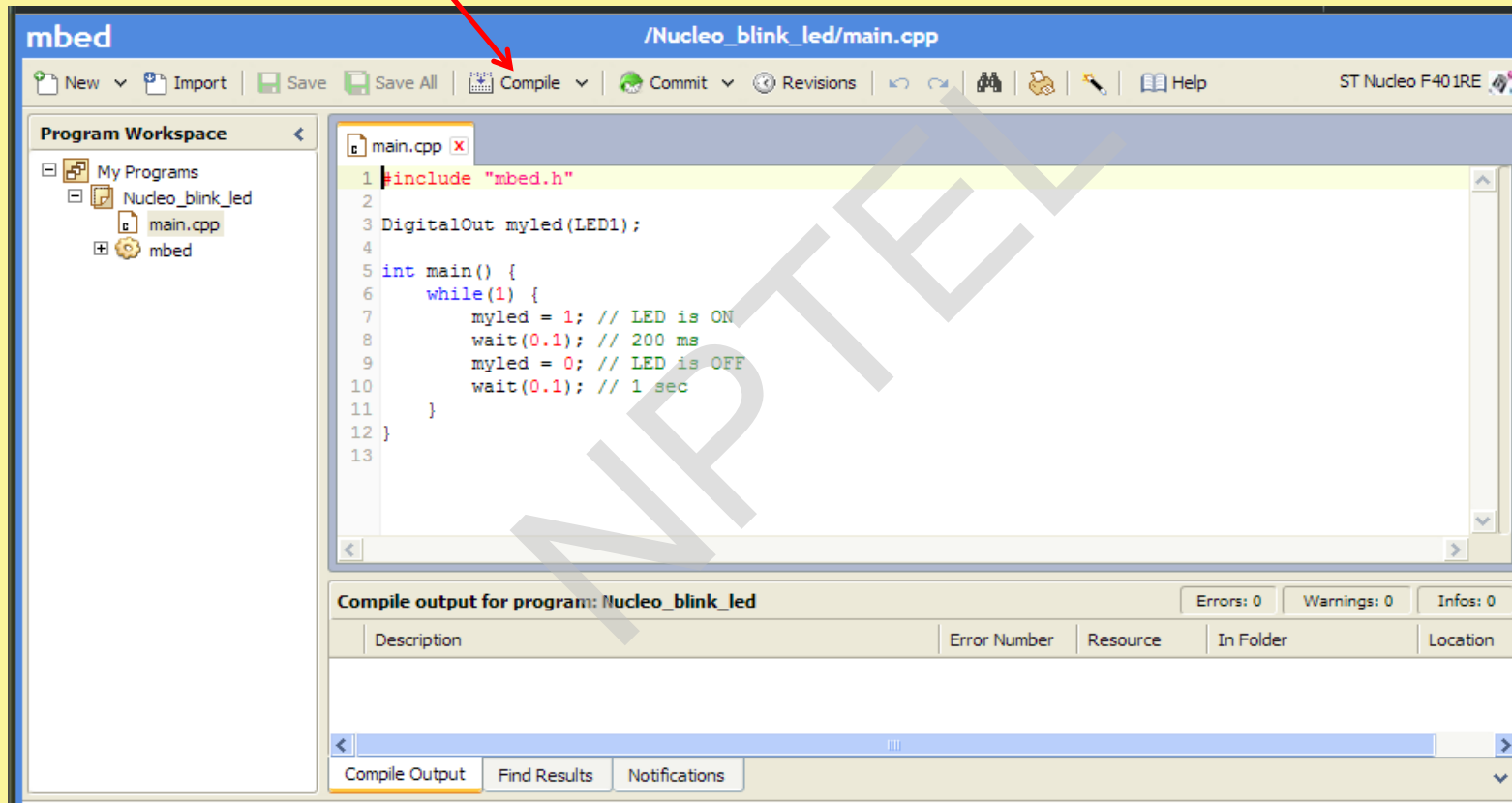
- STEP 7 – In the corner you can observe that the STM32F401RE kit has been added successfully.



- Create a new program under “*My Programs*” in the project sidebar to the left of the workspace by right clicking on “*My Programs*”.
- Choose Platform as *ST Nucleo F401RE* and name program as *Blinking_LED* and choose template as *empty program* and press ok.
- Now add the cpp file by right clicking on your program and then
 “*Add new file*” → <filename>.cpp → Press “OK”.
- Then click on the *main.cpp* file to open it for writing the code.



- STEP 8 – Compile the program



- The program starts running, and the LED starts blinking on the board.
- This sequence of steps has to be followed for any application development.
 - Board selection and driver installation have to be done only once.



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 20: Interfacing with STM32F401 Board

CONCEPTS COVERED

- ☐ Work out an example for blinking LED
- ☐ Work out an example for user input
- ☐ Work out an example for user button input



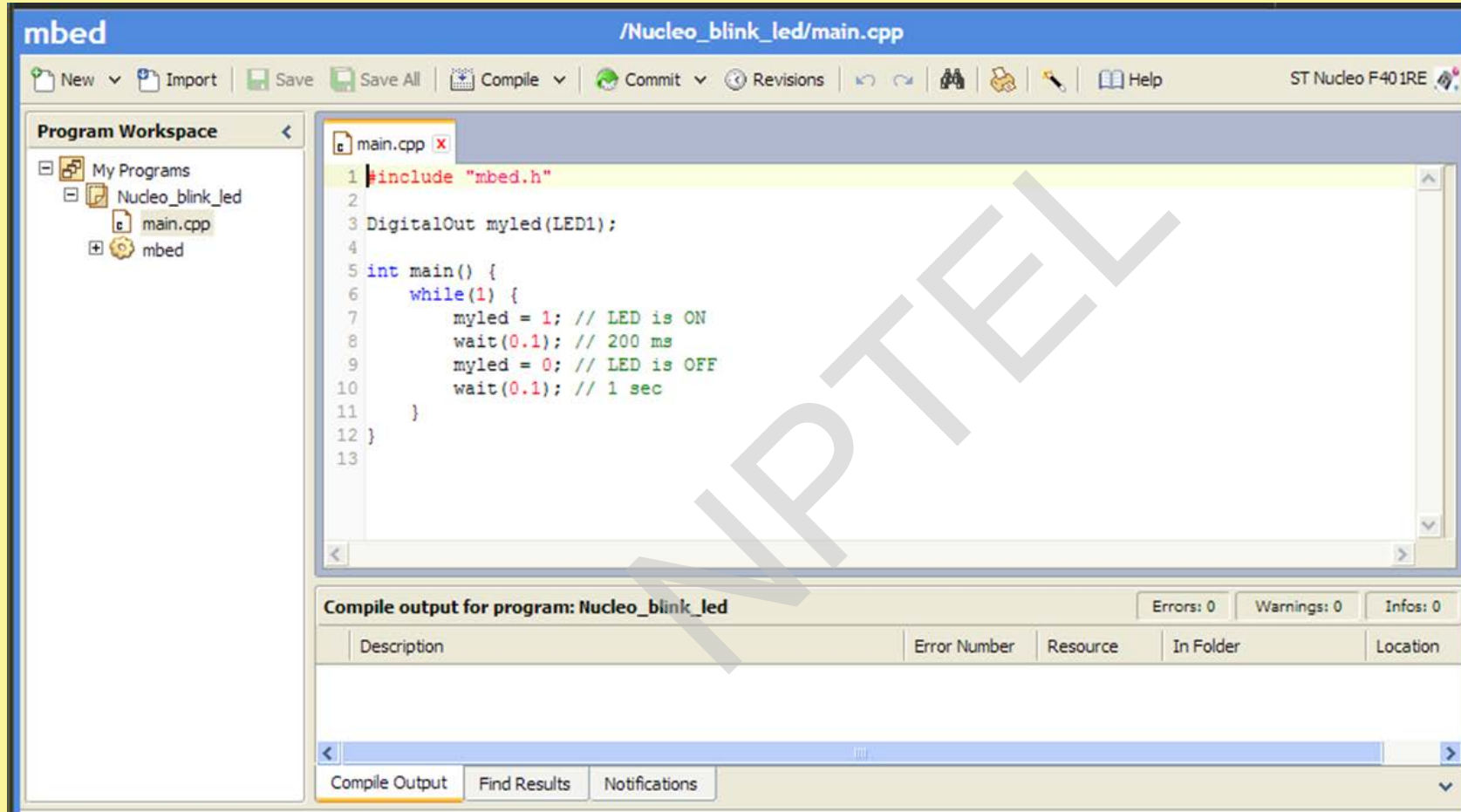
Introduction

- Three example programs shall be demonstrated on the STM32F401 development board.
 - Here, we do not interface any external devices.
 - We use the I/O facilities available on-board (like switches and LED), and simple external input.
- The examples:
 - Blink a LED on and off with specified on and off durations.
 - Depending on a user input on a port line, turn on or off a LED.
 - Use the on-board switch to turn on or off a LED.

Example 1: Blinking_LED.c

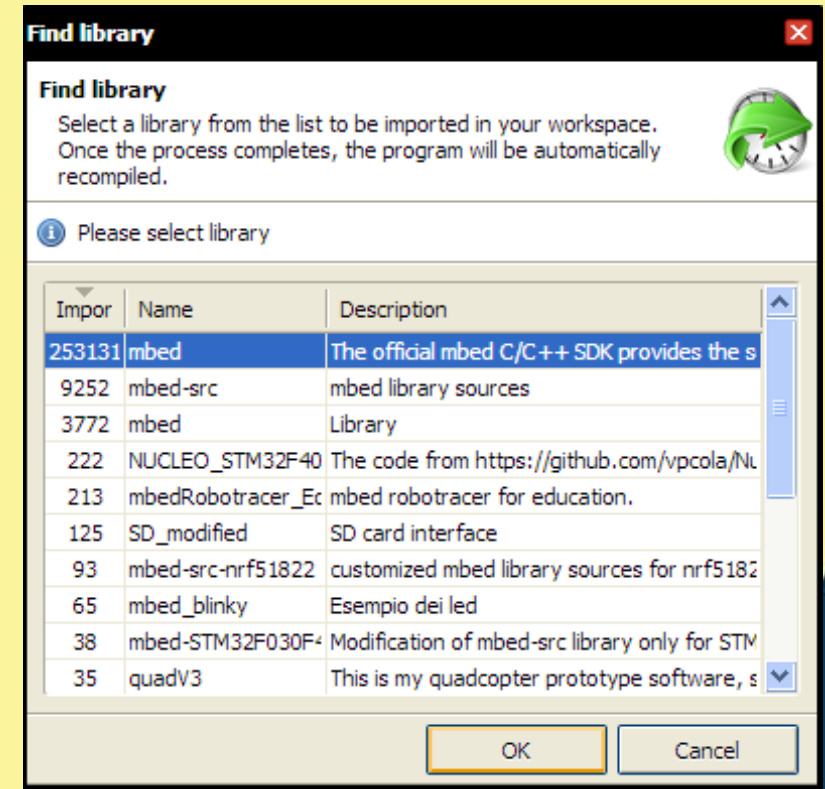
```
#include "mbed.h"
DigitalOut myled(LED3);
int main() {
    while(1) {
        myled = 1;    // LED is ON
        wait(0.2);    // for 200 ms
        myled = 0;    // LED is OFF
        wait(0.1);    // for 100 ms
    }
    return 0;
}
```

- Compiling the code:



- Compiling errors:

- At first you may see an error stating that it ***“cannot open source input file mbed.h”***
- Click on the ***“Fix It”*** button on the error line.
- You will get a list of libraries to choose from.
- Select the first one: ***“mbed – The official C/C++ SDK”*** and click on ***“OK”*** button.
- Compile again, there will not be any library errors anymore.

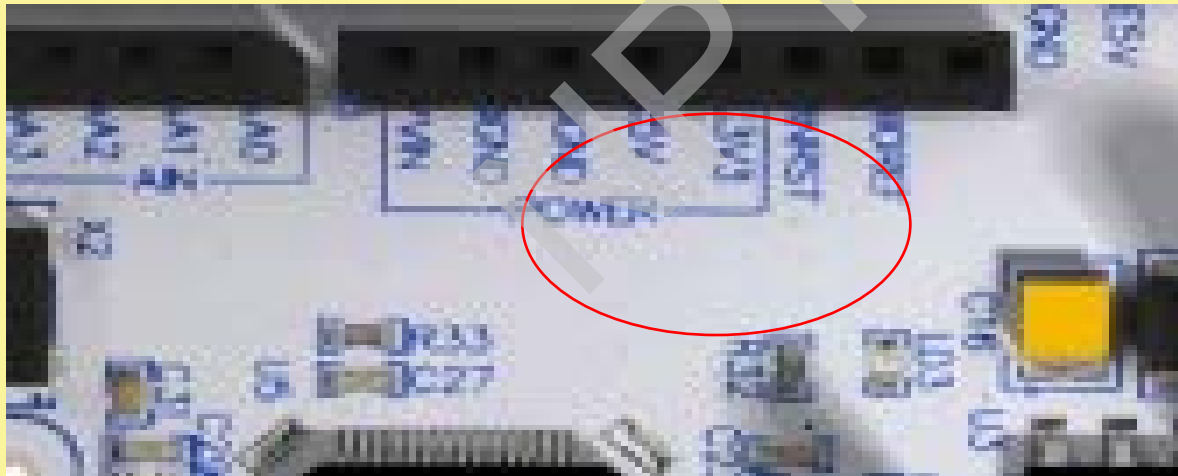


Upload Program to STM32 Nucleo Board

- On compiling the “*binary*” file will get downloaded to your computer.
- Connect your STM32 Nucleo to your computer via the USB cable.
- All the LEDs turn on and the USER LED will start blinking.
- Open My Computer and go to Download folder and copy the file “*Nucleo_blink_led_NUCLEO_F401RE.bin*”
- Now go to My *Computer -> Nucleo (drive)*, and paste the bin file into this drive.
- As soon as you paste the file, the power LED starts blinking with two colors (This indicates that the program is being burnt into the Flash Memory of the board).
- After the program is burnt, the USER Led will blink with the rate given.

Example 2: Testing for I/O Pin Configurations

- In this experiment we try to provide a digital input to the microcontroller and check the response by blinking LED3 (which is present on the board).
- We will need a single strand small wire and a 5V/3.3V supply (grounded with the board).
- We will use the *POWER out* port on the board to take the Digital Input.



Create the Program

- Following previous steps, create a Program named “*DigitalIOTest*” and add a “*Digital_IOTest.cpp*” file to the Program folder.
- Now write the code as shown.

```
#include "mbed.h"
DigitalIn myInput(D2);
DigitalOut myLED(LED3);
int main(){
    while(1){
        if(myInput){ myLED = 1;}
        else{ myLED = 0;}
    }
    return 0;
}
```

Digital_IOTest.cpp

Typical Usage of Digital I/O

- Digital I/O can be used for interfacing external devices some of which we will see in the course of the lectures and demonstration.
 - Switching based on activity (like interrupts, etc.).
 - Controlling electro-mechanical devices.
 - Taking sensor inputs.
 - Communicating other similar or dissimilar devices
 - many more..

Example 3: Testing the USER_BUTTON

- Here we will be using the *USER* button available on the board to control a *DigitalOut* to the user LED on board.
- Please note that the tact switch on the board by defaults sends out a V_{cc} signal (i.e., logic high) and when pressed sends out a ground signal (i.e., logic low).

Test_USER_BUTTON.cpp

```
#include "mbed.h"
DigitalIn myButton(USER_BUTTON);
DigitalOut out(LED3);
int main(){
    while(1){
        if(!myButton){           //when button pressed it sends 0
            out = 1;}
        else{//when not pressed it sends 1
            out = 0; }
        }
    return 0;
}
```



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

Lecture 21: Interfacing with Arduino UNO

CONCEPTS COVERED

- ☐ About the Arduino UNO board
- ☐ Description of pins available at the interface
- ☐ Work out an example for blinking LED

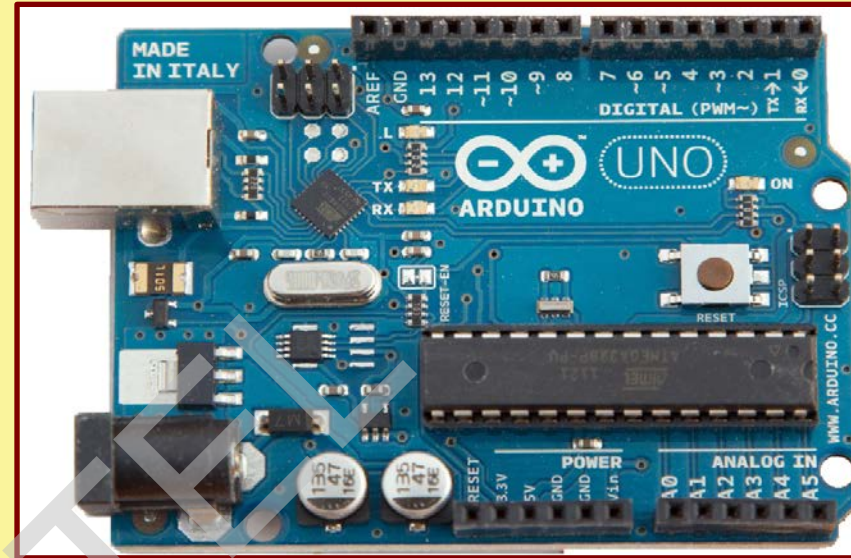


What is Arduino?

- Arduino is an open source computer hardware and software company.
- Arduino designs and manufactures embedded system boards and accessories.
 - One such popular microcontroller board is the Arduino Uno.
- The Arduino IDE can be downloaded from:
<https://www.arduino.cc/en/Main/Software>

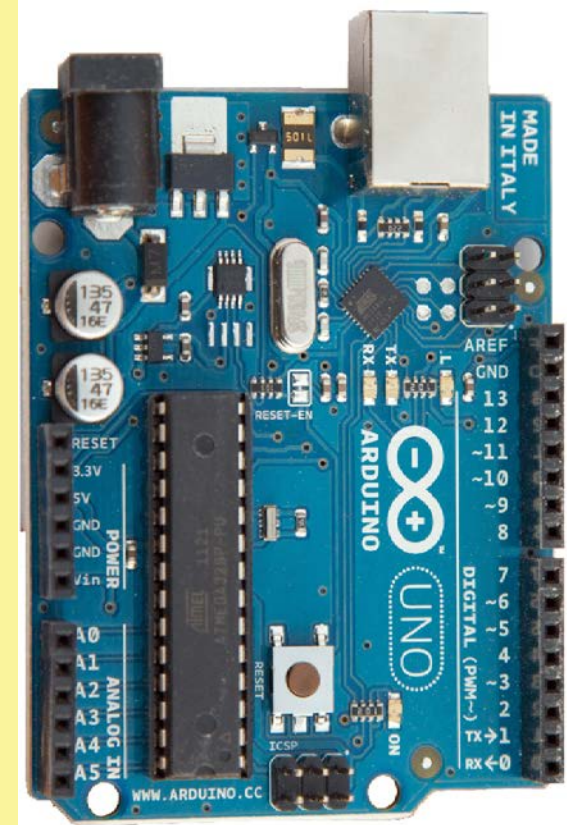
Requirements to Begin

- Arduino development board (Arduino UNO)
- USB connector
- Development environment (Arduino IDE).



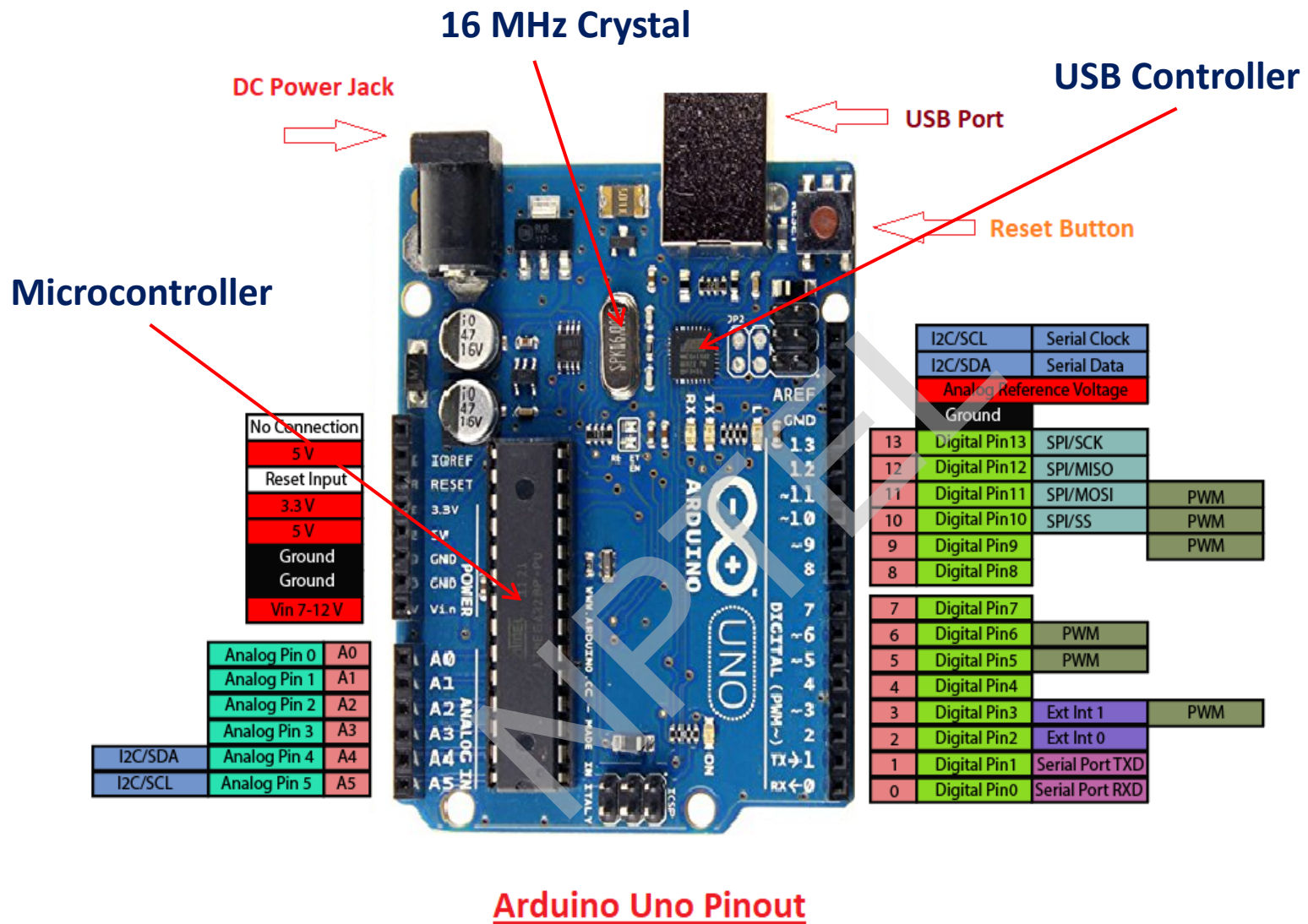
Arduino UNO

- The Arduino UNO is a widely used open-source micro-controller board based on the ATmega328P microcontroller and developed by Arduino.
- It consists of:
 - 14 Digital pins and 6 Analog pins.
 - It is programmable with the Arduino IDE via a type B USB cable.
 - It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.



Arduino UNO (contd.)

- The UNO board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform.
- The ATmega328P on the Arduino Uno comes pre-programmed with a boot loader that allows to upload new code to it without the use of an external hardware programmer.
 - It communicates using the original STK500 protocol.
- The communication between the STK500 and the PC is done over RS232 (PC COM port). The STK500 uses: 115.2kbps, 8 data bits, 1 stop bits, no parity.
 - The PC should be set up similarly for the communication to work.



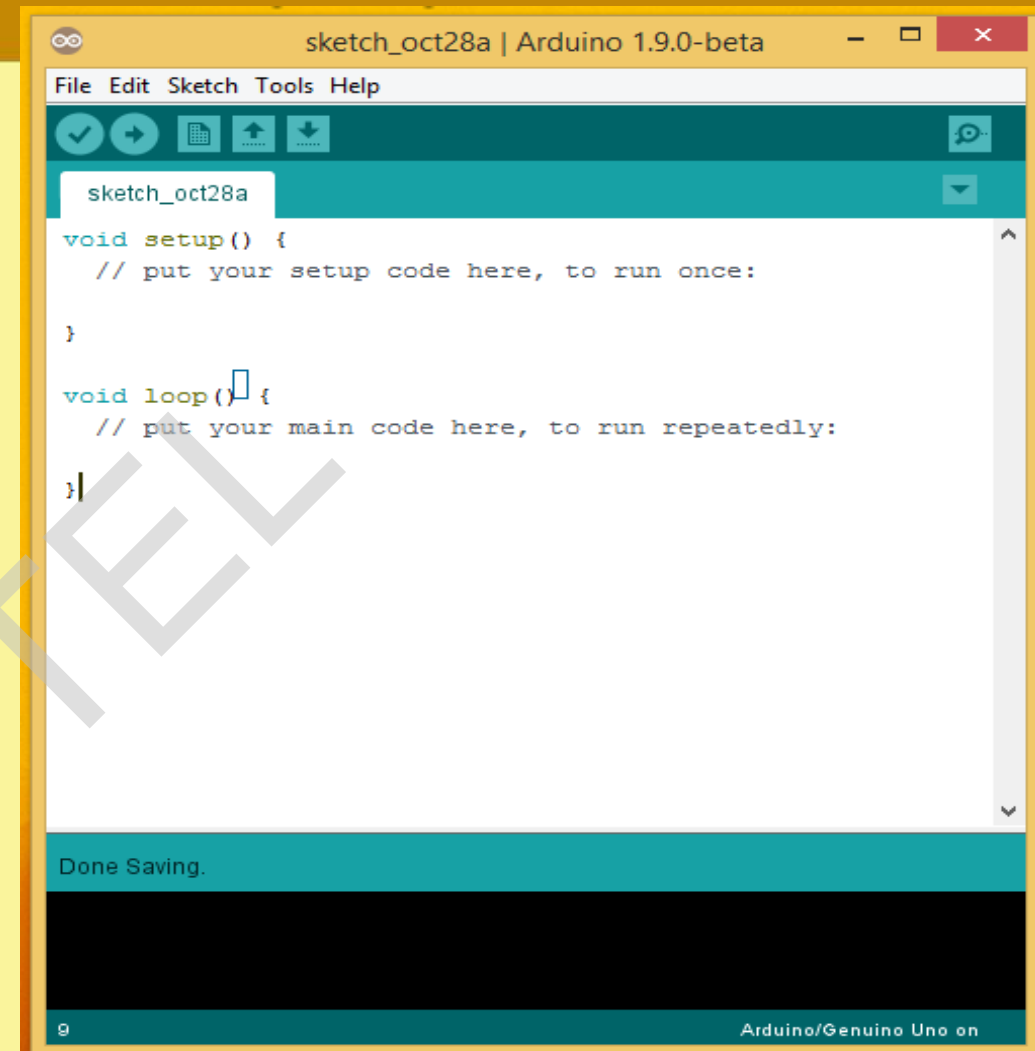
Points for Programming

- First download the software from the link given:
<https://www.arduino.cc/en/Main/Software>
- The 14 digital input/output pins can be used as input or output pins by using *pinMode()*, *digitalRead()* and *digitalWrite()* functions during programming.
- There are also 6 analog input pins, each of which provide 10 bits of resolution.

- Extract the downloaded ZIP file and click on *arduino.exe* then the arduino IDE will open.



- The IDE will open as:
 - The Arduino program basically have two section.
 - The “*void setup()*” section is widely used to initialize variables, pin modes, set the serial baud rate etc. The software only goes through this section once.
 - The “*void loop()*” section is the part of the code that loops back onto itself and is the main part of the code.

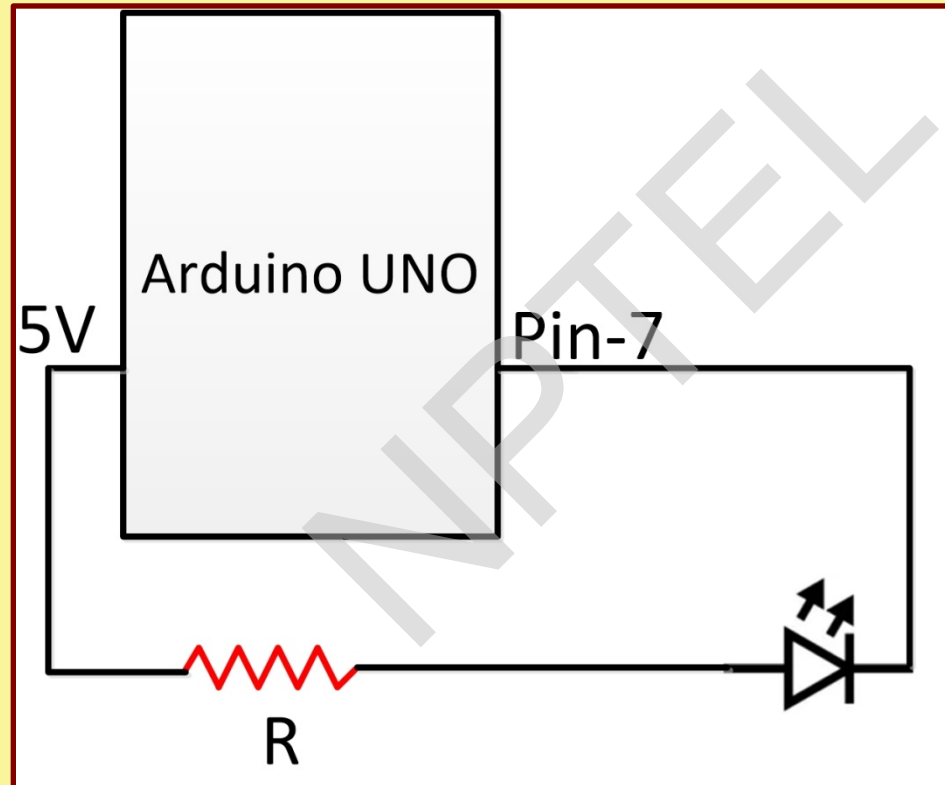


An Example: To blink an LED

```
void setup() {  
    pinMode(7, OUTPUT); // set the digital pin 7 as output  
}
```

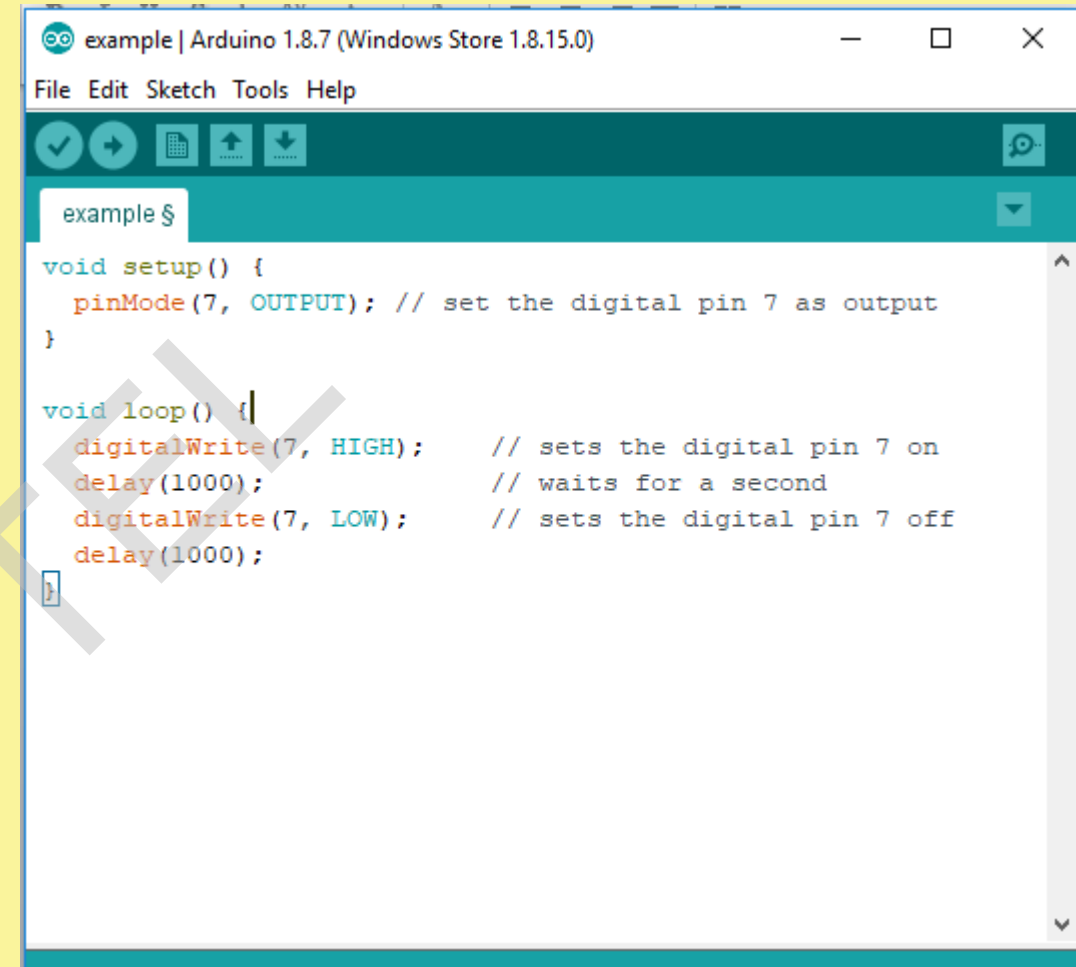
```
void loop() {  
    digitalWrite(7, HIGH); // sets the digital pin 7 on  
    delay(1000);           // wait for 1 second  
    digitalWrite(7, LOW);  // sets the digital pin 7 off  
    delay(1000);           // wait for 1 second  
}
```

Connection Diagram



Steps to Run the Program

- Step 1:- Select file and then New.
Type in the program:



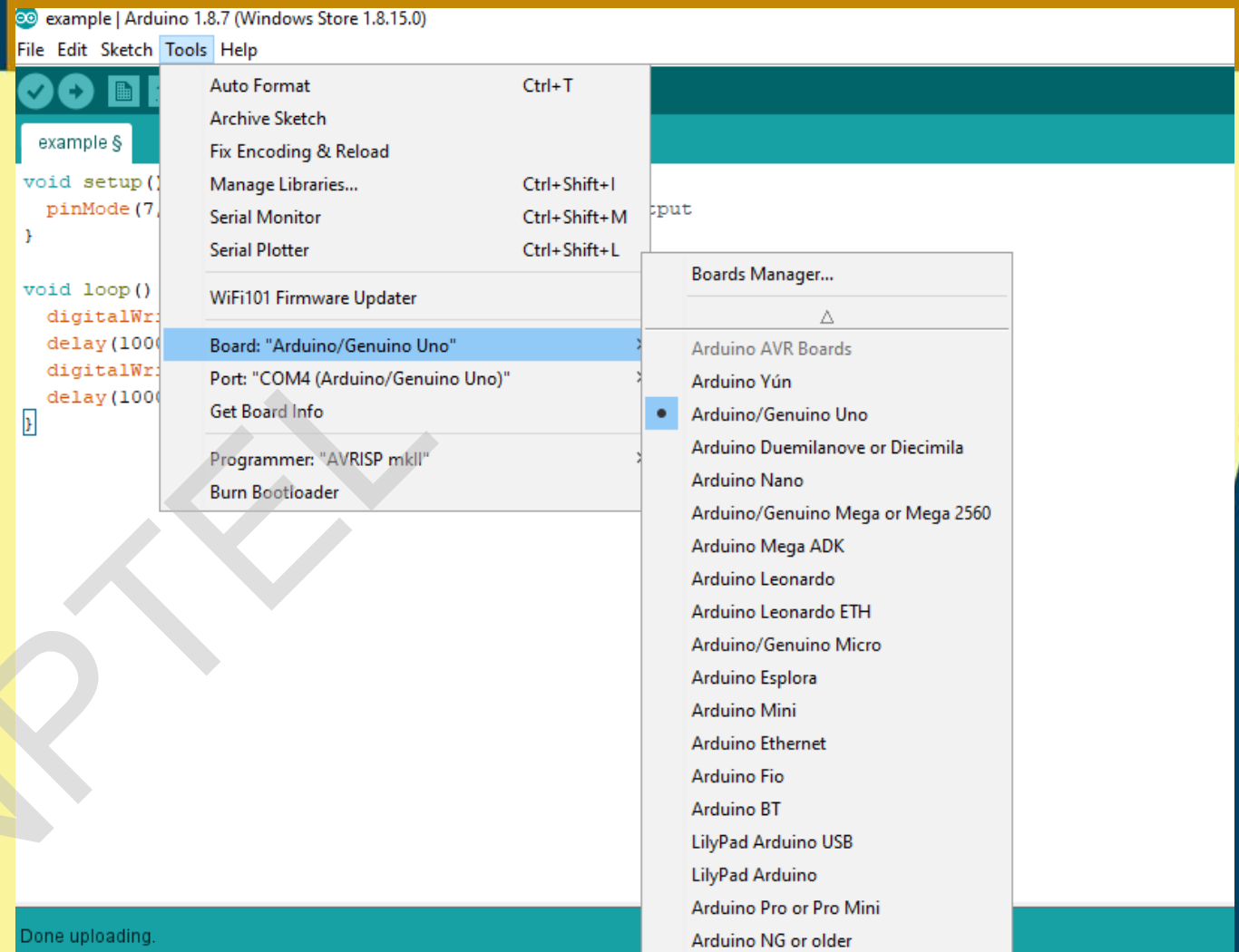
```
example | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help

example $

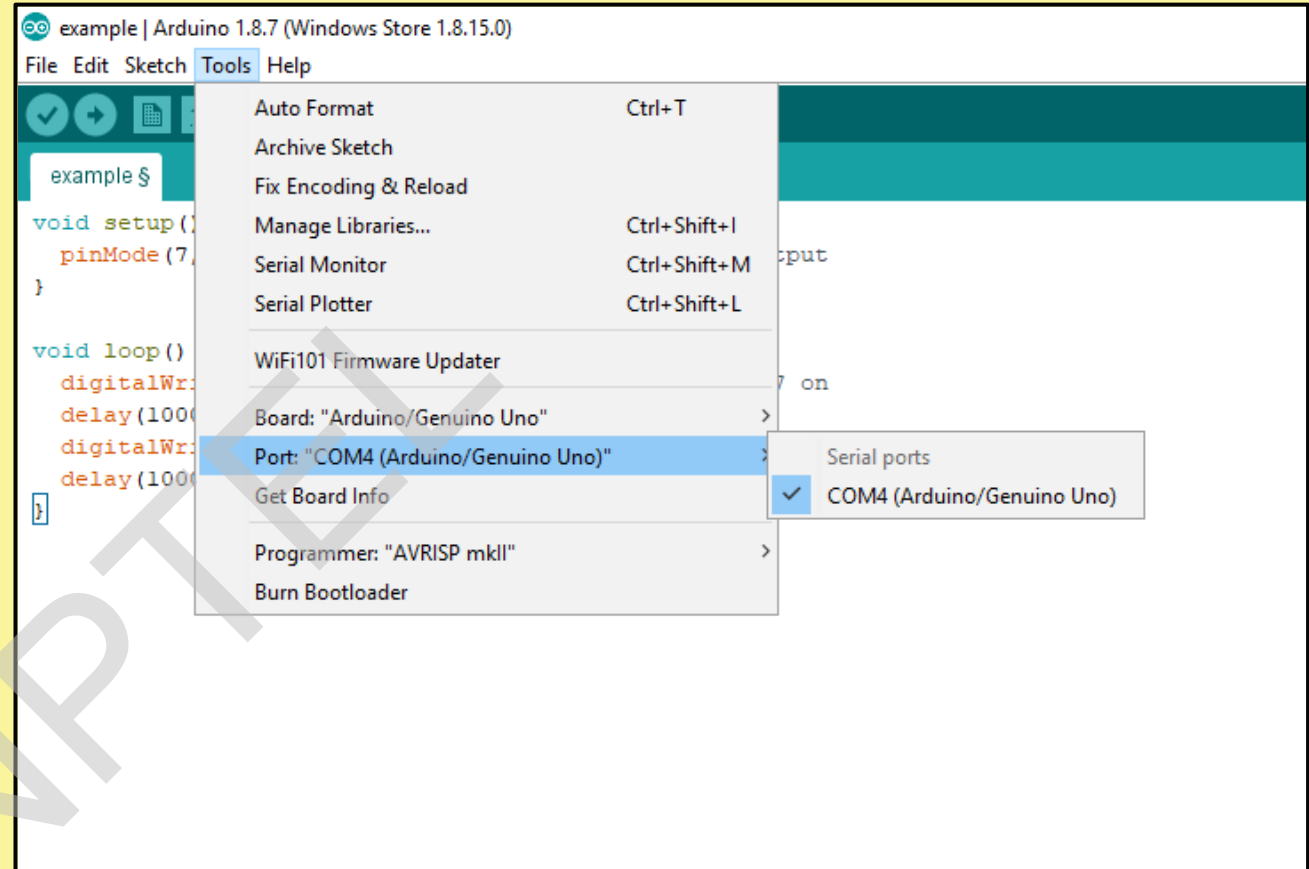
void setup() {
  pinMode(7, OUTPUT); // set the digital pin 7 as output
}

void loop() {
  digitalWrite(7, HIGH); // sets the digital pin 7 on
  delay(1000); // waits for a second
  digitalWrite(7, LOW); // sets the digital pin 7 off
  delay(1000);
}
```

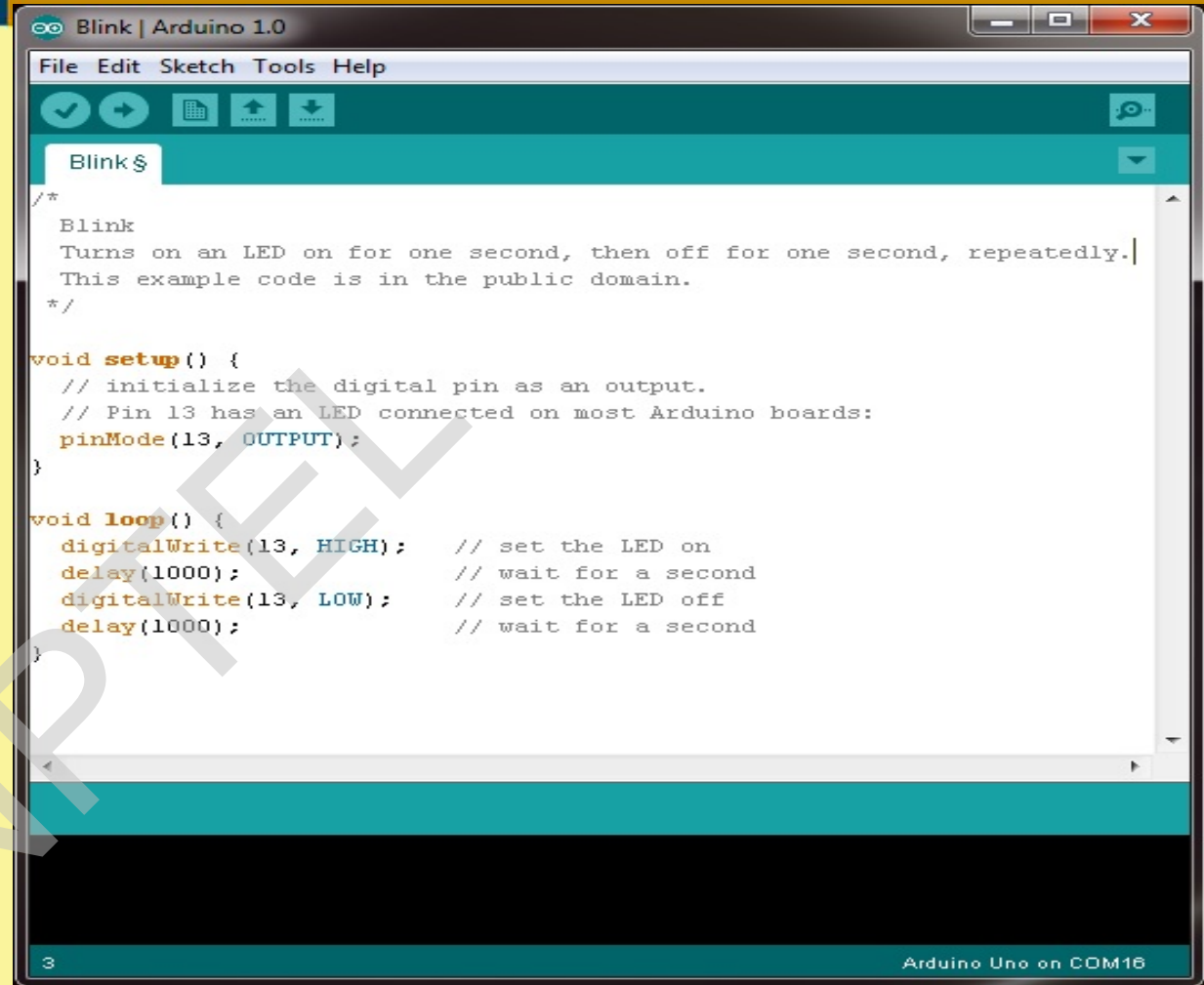
- Step 2:- Select your Arduino board.
Write the program:



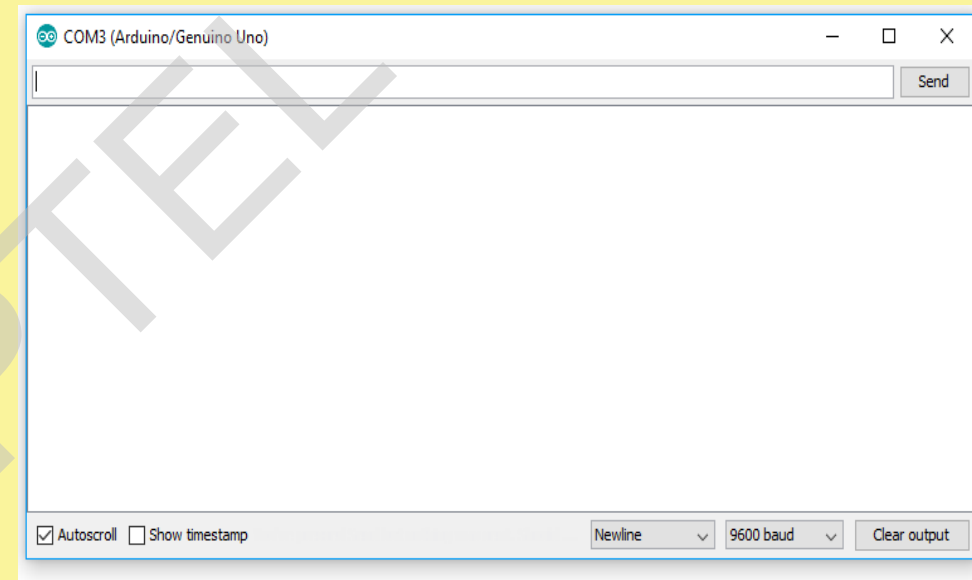
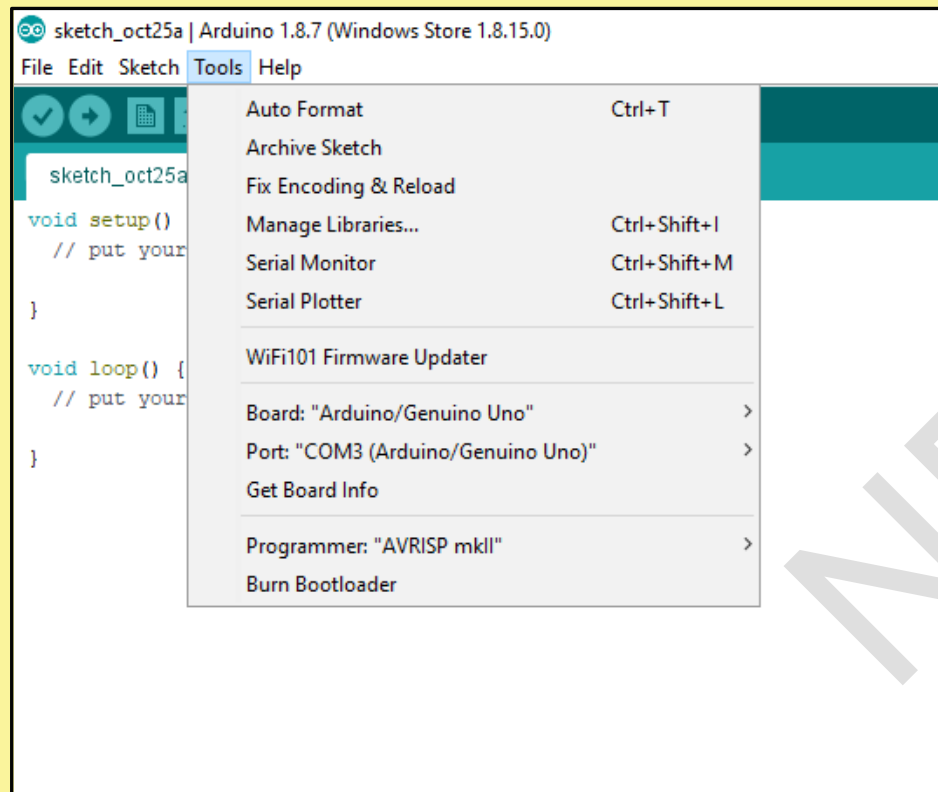
- Step 3:- Select the port number



- Step 4: Press the upload symbol



In Arduino there is no need for HyperTerminal





NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

**Lecture 22: Interfacing 7-segment LED and LCD Displays
(Part 1)**

CONCEPTS COVERED

- ☐ Work out an example for 7-segment LED display
- ☐ Work out an example for LCD display

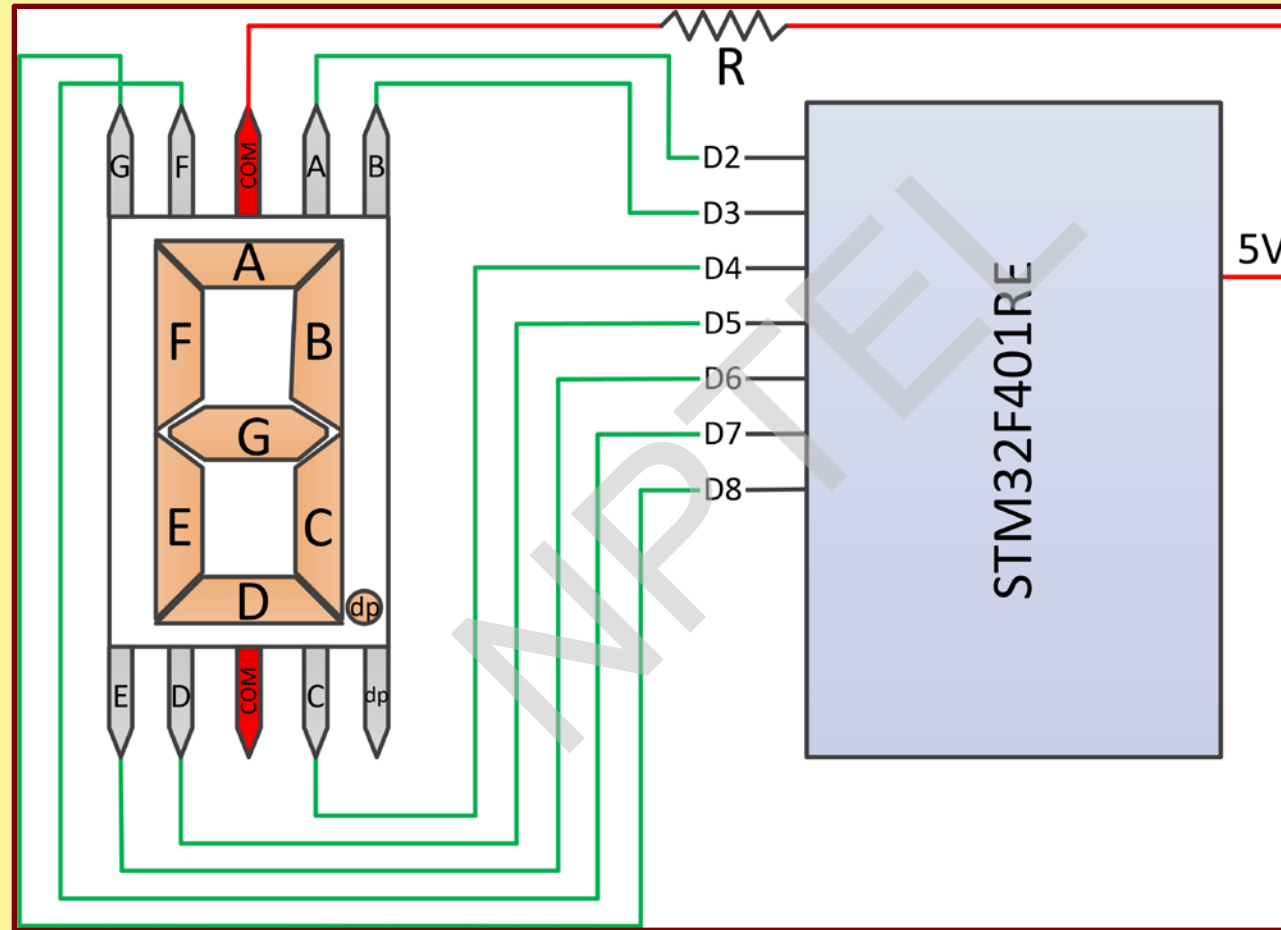


7-segment LED Display Interfacing

Example 1

- Interface a 7-segment display unit to the STM32 board, and display the characters 0 to 9 cyclically with time delay of 1.5 seconds.
 - Connect the segments A to G to the data output pins D2 to D8 on the Arduino connector.

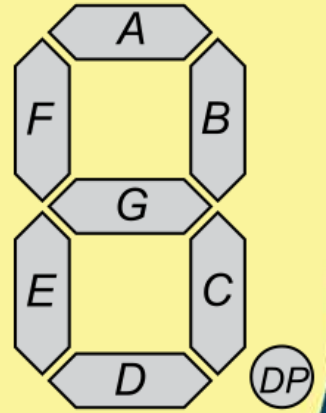
Example 1: Connection Diagram



Example 1: Mbed C Program

```
#include "mbed.h"
DigitalOut A(D2);
DigitalOut B(D3);
DigitalOut C(D4);
DigitalOut D(D5);
DigitalOut E(D6);
DigitalOut F(D7);
DigitalOut G(D8);
```

```
void Display(int disp) {
    switch(disp)
    {
        case 0: A=0;B=0;C=0;D=0;E=0;F=0;G=1; break;
        case 1: A=1;B=0;C=0;D=1;E=1;F=1;G=1; break;
        case 2: A=0;B=0;C=1;D=0;E=0;F=1;G=0; break;
        case 3: A=0;B=0;C=0;D=0;E=1;F=1;G=0; break;
        case 4: A=1;B=0;C=0;D=1;E=1;F=0;G=0; break;
        case 5: A=0;B=1;C=0;D=0;E=1;F=0;G=0; break;
        case 6: A=0;B=1;C=0;D=0;E=0;F=0;G=0; break;
        case 7: A=0;B=0;C=0;D=1;E=1;F=1;G=1; break;
        case 8: A=0;B=0;C=0;D=0;E=0;F=0;G=0; break;
        case 9: A=0;B=0;C=0;D=0;E=1;F=0;G=0; break;
    }
}
```

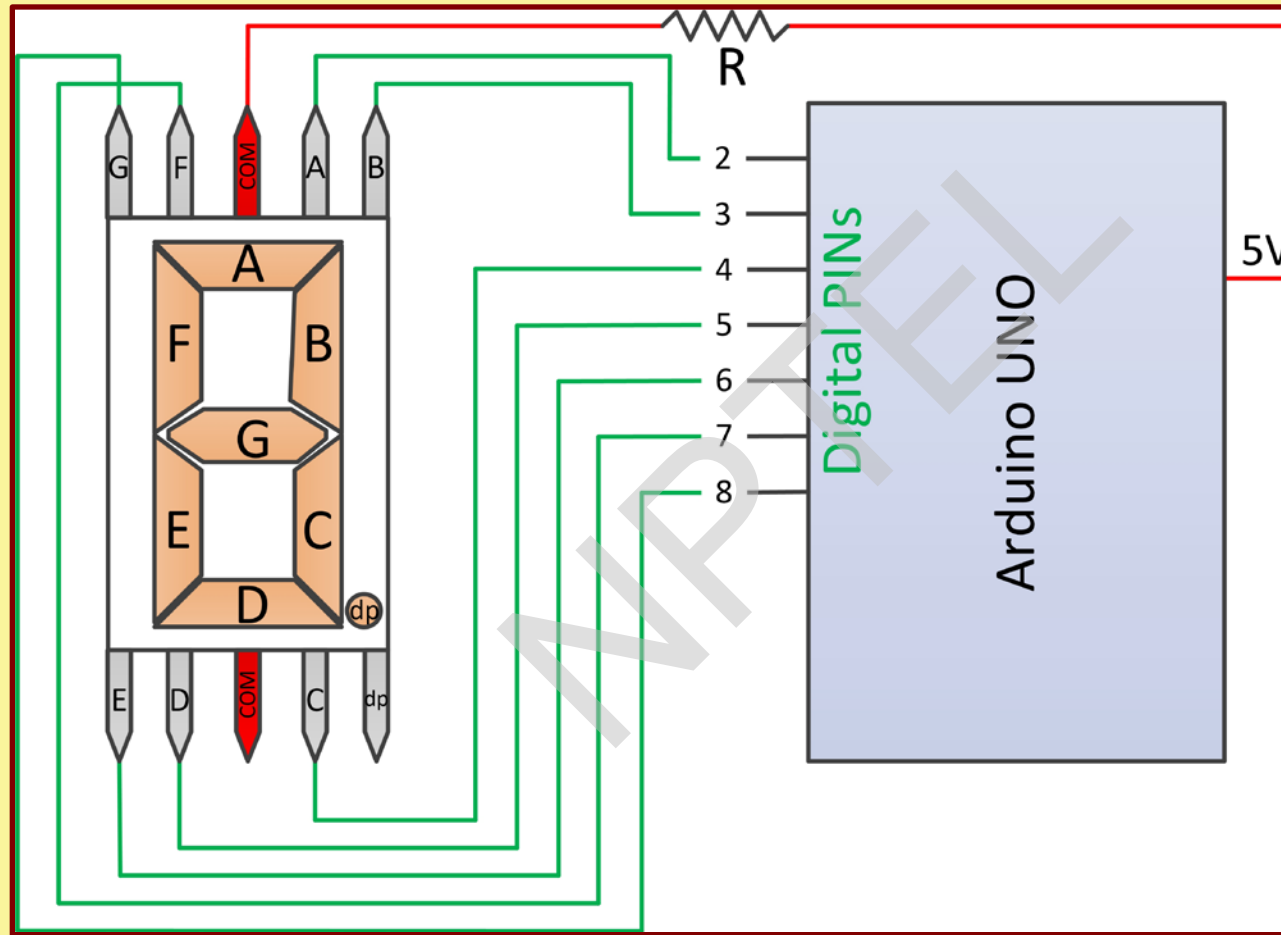


```
int main() {  
    while(1)  
    {  
        for(int i=0; i<=9; i++)  
        {  
            Display(i);  
            wait(1.5);  
        }  
    }  
}
```

Example 2

- Interface a 7-segment display unit to the Arduino UNO board, and display the characters 0 to 9 cyclically with time delay of 1.5 seconds.
 - Connect the segments A to F to the digital pins 2 to 8 on the Arduino board.

Example 2: Connection Diagram

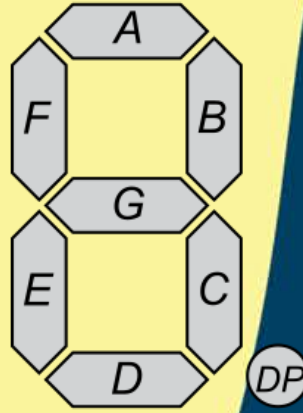


Example 2: Arduino Program

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}

void loop() {
  while(1) {
    for (int i=0; i<=9; i++) {
      Display(i);
      delay(1500); }
  }
}
```

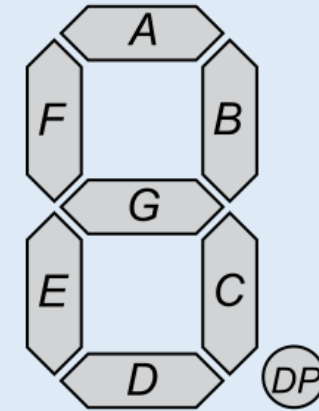
```
void Display(int disp) {
  switch(disp)
  {
    case 0: digitalWrite(2, LOW);
              digitalWrite(3, LOW);
              digitalWrite(4, LOW);
              digitalWrite(5, LOW);
              digitalWrite(6, LOW);
              digitalWrite(7, LOW);
              digitalWrite(8, HIGH);
              break;
    ... similarly for case 1 to 9
  }
}
```



Example 2: Alternate Program

```
void setup() {  
    for(int i=2; i<=8; i++)  
    {  
        pinMode(i, OUTPUT);  
    }  
  
    void on() {  
        for(int i=2; i<=8; i++)  
        {  
            digitalWrite(i, LOW);  
        }  
    }  
}
```

```
void Display(int disp) {  
    if (disp==1 || disp==4) {  
        digitalWrite(2, HIGH);  
    }  
    if(disp==2) {  
        digitalWrite(4, HIGH); }  
    if (disp==5 || disp==6) {  
        digitalWrite(3, HIGH); }  
    if( disp==1 || disp==4 || disp==7) {  
        digitalWrite(5, HIGH); }  
    if (disp==1 || disp==3 || disp==4 || disp==5 ||  
        disp==7 || disp==9) {  
        digitalWrite(6, HIGH);}  
    if (disp==1 || disp==2 || disp==3 || disp==7){  
        digitalWrite(7, HIGH);}  
    if (disp==0 || disp==1 || disp==7) {  
        digitalWrite(8, HIGH);}  
}
```



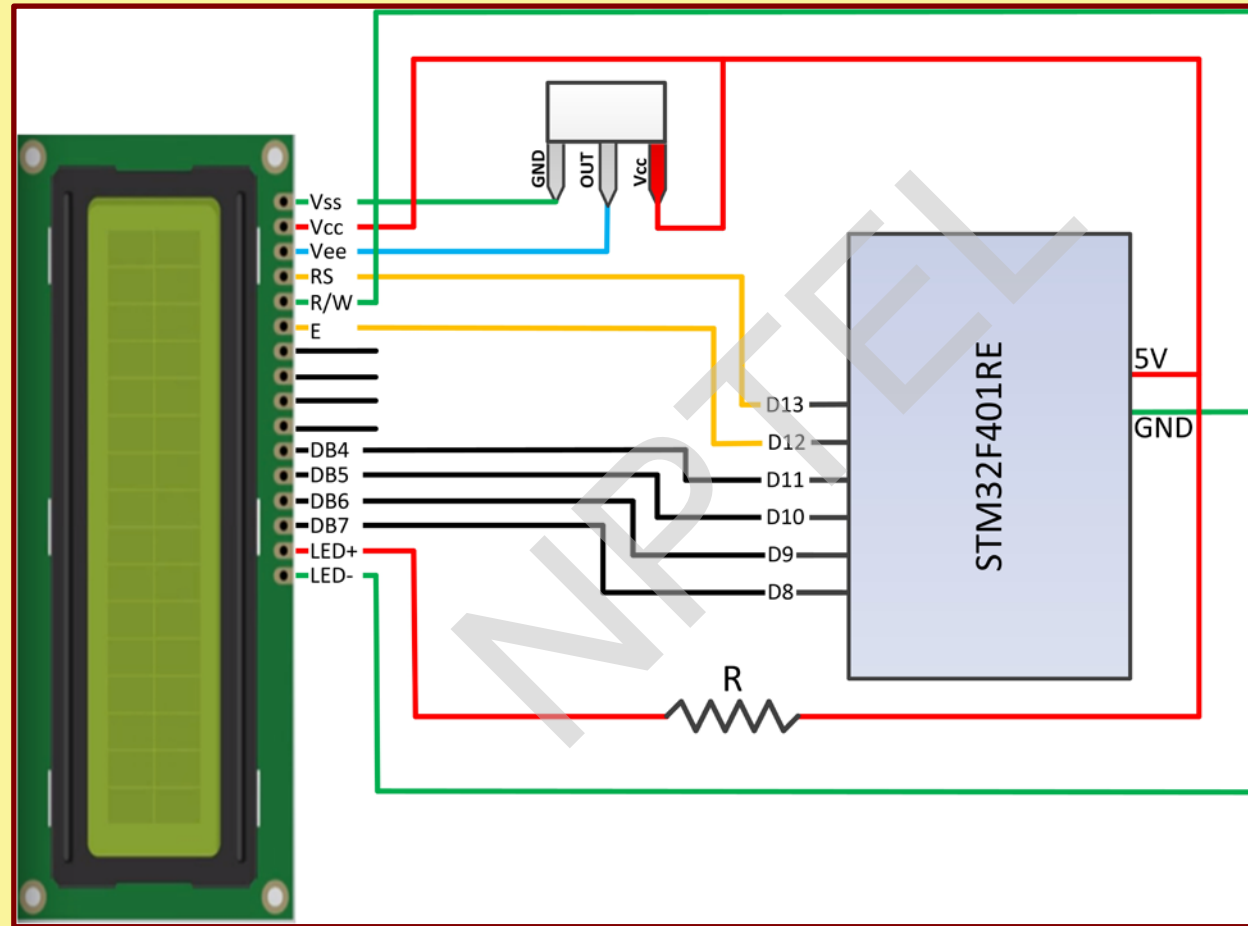
```
void loop(){  
    while(1)  
    {  
        for(int i=0; i<=9; i++)  
        {  
            Display(i);  
            delay(1500);  
            on();  
        }  
    }  
}
```

LCD Display Interfacing

Example 3

- Interface a LCD display unit to the STM32 board in 4-bit mode, and display a fixed message on the screen.
 - Connect the four data lines to D8 to D11 on the Arduino connector.
 - Connect the register select (RS) input to D13.
 - Connect the enable (E) input to D12.

Example 3: Connection Diagram



Example 3: Mbed C Program

```
#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"
TextLCDScroll lcd (D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);
int main() {
while(1)
{
    lcd.setLine (0, "EMBEDDED SYSTEM");
    lcd.setLine (1, "NPTEL - 2018");
    wait(20);
}
}
```

Example 3a: Mbed C Program (scroll at 2 char/sec)

```
#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"
TextLCDScroll lcd(D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);
int main() {
    lcd.cls();
    lcd.setSpeed (2);
    while(1)
    {
        lcd.setLine(0, "EMBEDDED SYSTEM DESIGN WITH ARM");
        lcd.setLine(1, "NPTEL - 2018");
        wait(100);
    }
}
```

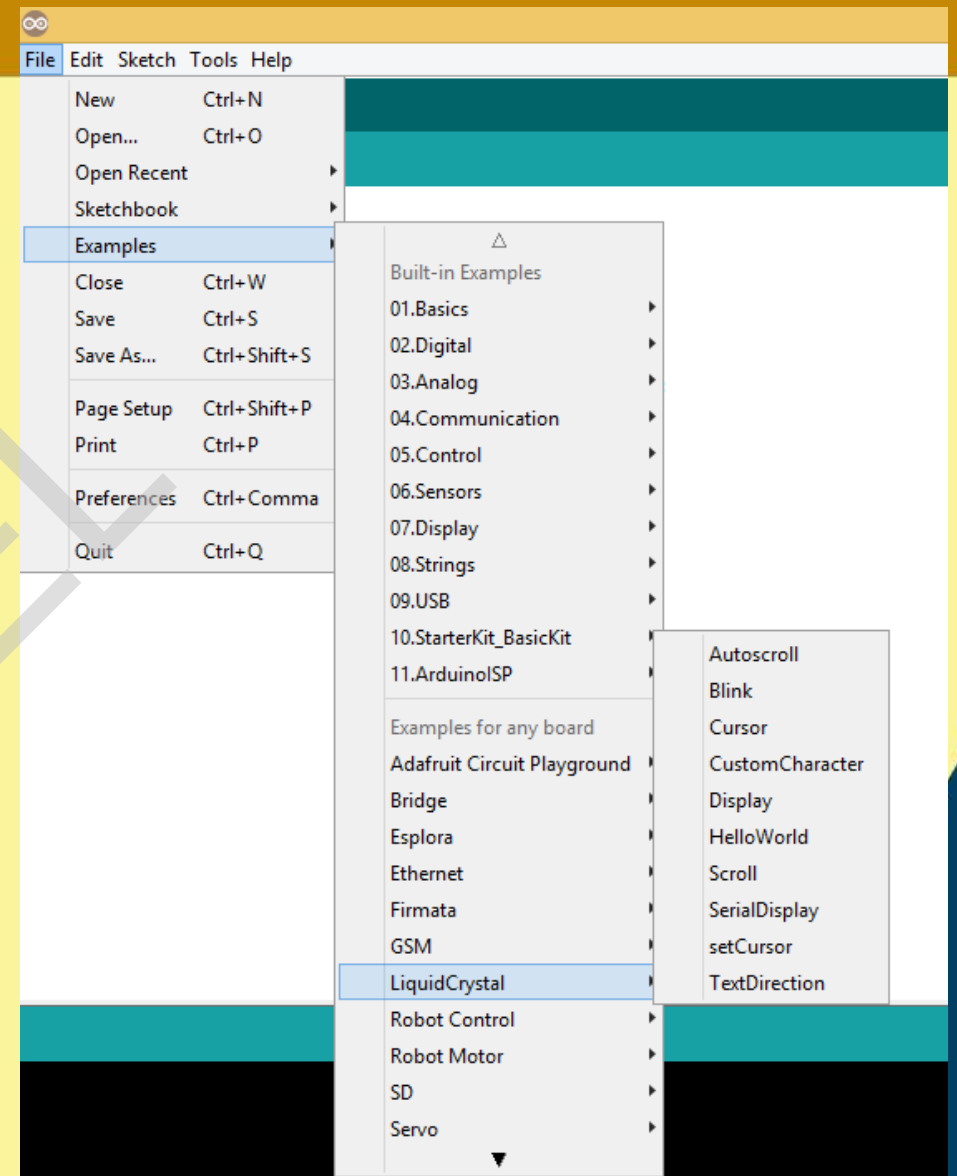
Example 3b: Mbed C Program (scroll at 3 char/sec)

```
#include "mbed.h"
#include "TextLCD.h"
#include "TextLCDScroll.h"
TextLCDScroll lcd(D13, D12, D11, D10, D9, D8, TextLCD::LCD16x2);
int main() {
    lcd.cls();
    lcd.setSpeed (3);
    while(1)
    {
        lcd.setLine(0, "EMBEDDED SYSTEM DESIGN WITH ARM");
        lcd.setLine(1, "NPTEL - 2018");
        wait(100);
    }
}
```

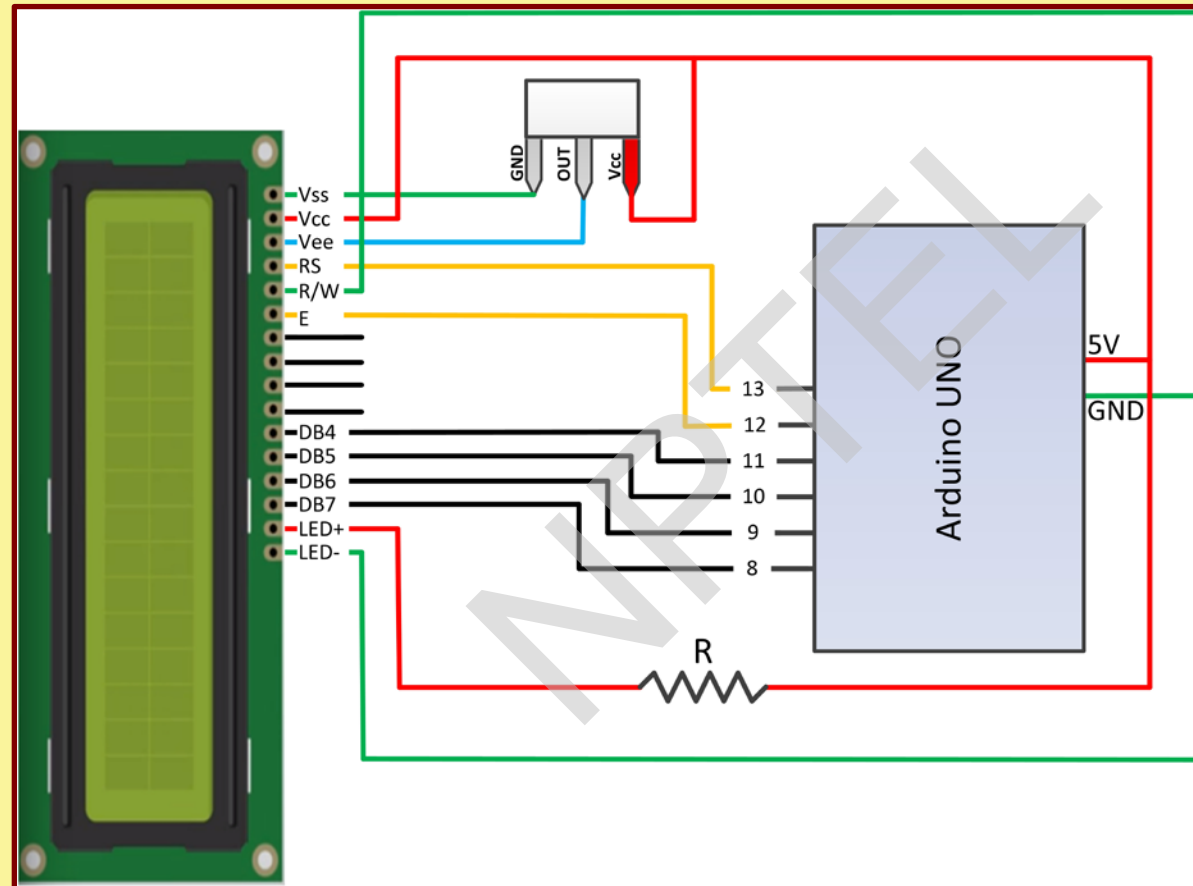
Example 4

- Interface a LCD display unit to the Arduino UNO board in 4-bit mode, and display a fixed message on the screen.
 - Connect the four data lines to pins 8 to 11.
 - Connect the register select (RS) input to pin 13.
 - Connect the enable (E) input to pin 12.

- In arduino *TextLCD* function is already imported it can be directly accessible in the example section.
- Open arduino IDE and follow the steps : *File->Examples-> LiquidCrystal->Choose any Example (Hello World).*
- Modify the program according to your need.
- The description of the connection will load by default.



Example 4: Connection Diagram



Example 4: Arduino Program

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (13, 12, 11,10, 9, 8);
void setup() {
    lcd.begin(16, 2);
    lcd.print("EMBEDDED SYSTEM");
    lcd.setCursor(0,1);
    lcd.print("NPTEL - 2018");
    delay(1000);
}
void loop() {
    lcd.display();
    delay(10000);
}
```



NPTEL ONLINE CERTIFICATION COURSES

Thank
you!



NPTEL ONLINE CERTIFICATION COURSES

Course Name: Embedded System Design with ARM

Faculty Name: Dr. Kamalika Datta

Department : Computer Science and Engineering

Topic

**Lecture 23: Interfacing 7-segment LED and LCD Displays
(Part 2)**

CONCEPTS COVERED

- ☐ Demonstration of 7-segment LED display interfacing on STM32
- ☐ Demonstration of 7-segment LED display interfacing on Arduino
- ☐ Demonstration of LCD display interfacing on STM32
- ☐ Demonstration of LCD display interfacing on Arduino





NPTEL ONLINE CERTIFICATION COURSES

**Thank
you!**