DATA COMMUNICATION

LAB 1: LED BLINKY

I. Installation

Start Arduino IDE and open Preferences window by clicking on **File** and then, **Preferences** (see Figure 1)

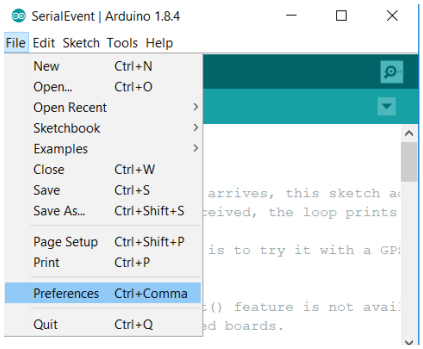


Figure 1: Open Preferences window

Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into Additional Board Manager URLs field. Click OK

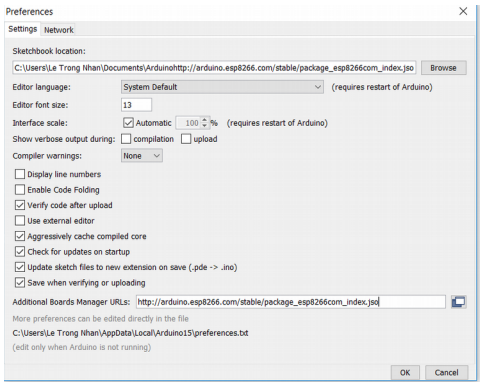


Figure 2: After providing the URL, click on OK button

Open Boards Manager from **Tools, Board** menu and find **Boards Managers…**

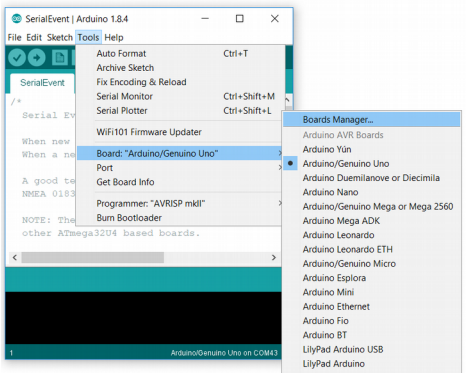


Figure 3: Open Boards Manager window

Search for NodeMCU, then click Install the ESP8266 library (see Figure 4).

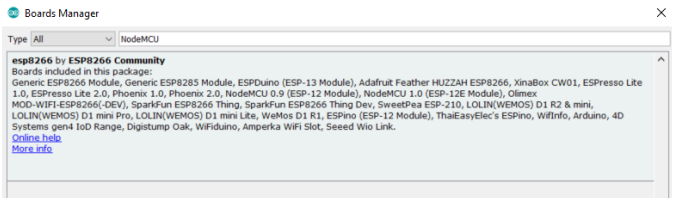


Figure 4: Install ESP8266 library

Click on Close button when the installation process is finished.

Finally, go to **Tools**, select **Board**, and then select the version board. In this manual, **NodeMCU** board is used.

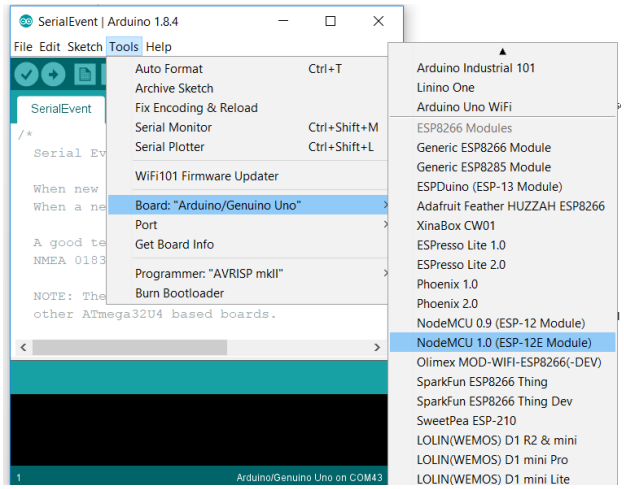


Figure 5: Select the version board: NodeMCU

II. First Project: LED Blinky

Go to menu **File, Examples, Basics** then select **Blink**

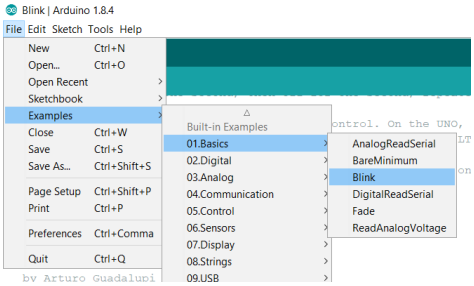


Figure 6: First example project on NodeMCU

Go to your Windows 'Device Manager' to find out which Com Port 'USB-Serial CH340' is assigned to. Select the matching COM/serial port for your CH340 USB-Serial interface.

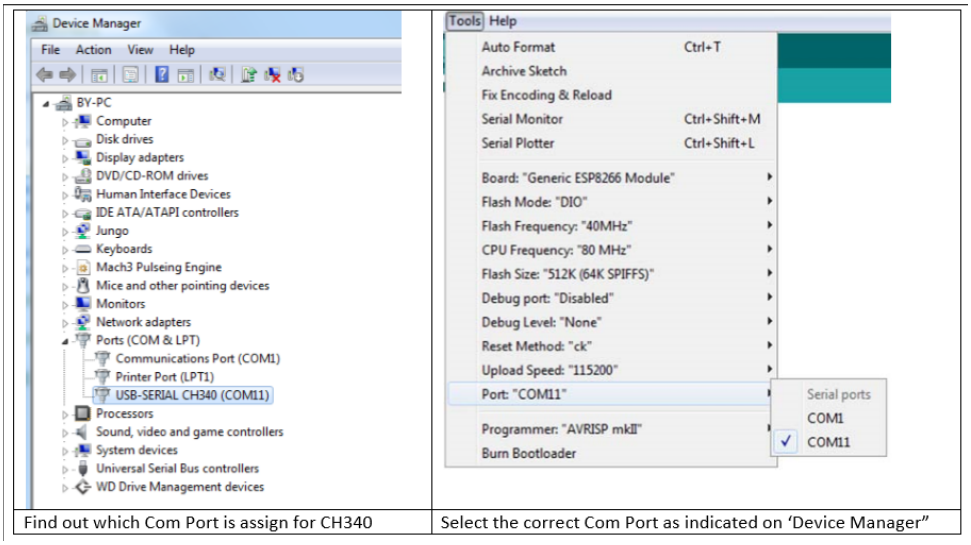


Figure 7: Select the correct COM port

Click on Upload button to download the code. If there is no error, you can find the status Done Uploading on the console and the LED will blink every second.

III. Exercise

1. Implement your code to blink the LED every 2 seconds.

```
void loop(){
    digitalWrite(ledPin, HIGH);
    delay(2000);
    digitalWrite(ledPin, LOW);
    delay(2000);
}
```

2. Implement your code to turn on the LED for 3 seconds and then turn off the LED for 1 seconds

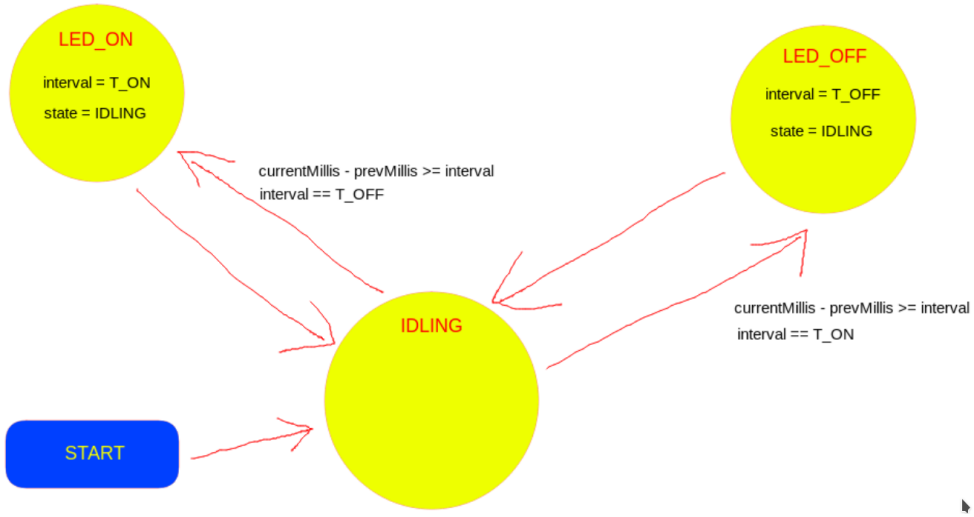
```
void loop(){
    digitalWrite(ledPin, HIGH);
    delay(1000);
    digitalWrite(ledPin, LOW);
    delay(3000);
}
```

3. Implement your code to blink the LED with the frequency of 5Hz

```
void loop(){
```

```
    digitalWrite(ledPin, LOW);
    delay(200);
    digitalWrite(ledPin, HIGH);
    delay(200);
}
```

4. It is assumed that there are two states for a LED, named LED_ON and LED_OFF. This LED will be ON for T_on and OFF for T_off. This process is repeated forever. Design your state machine and indicate the transition state condition (using arrows and put a label to demonstrate the transition condition). More states can be added according to your design but the 2 states LED_ON and LED_OFF are mandatory.




```
switch (state){
    case IDLING:{
      unsigned long currentMillis = millis();
      if (currentMillis - prevMillis >= interval){
        prevMillis = currentMillis;
        if (interval == T_ON){
          state = LED_OFF;
          interval = T_OFF;
        } else {
          state = LED_ON;
          interval = T_ON;
        }
      }
      break;
    }
     case LED_ON:{
      digitalWrite(LED_BUILTIN, LOW);
      state = IDLING;
      break;
     }
     case LED_OFF:{
      digitalWrite(LED_BUILTIN, HIGH);
      state = IDLING;
      break;
     }
  }
```

5. Implement your state machine in Arduino and test with `T_on = 300` and `T_off = 500` (meaning that the LED turns on for 3 seconds and the off for 5 seconds). These values are initiated on the `setup()` function. It is worth noting that only one `delay(10)` function is used in the `loop()` function. Refer to the template presented in Figure 8.

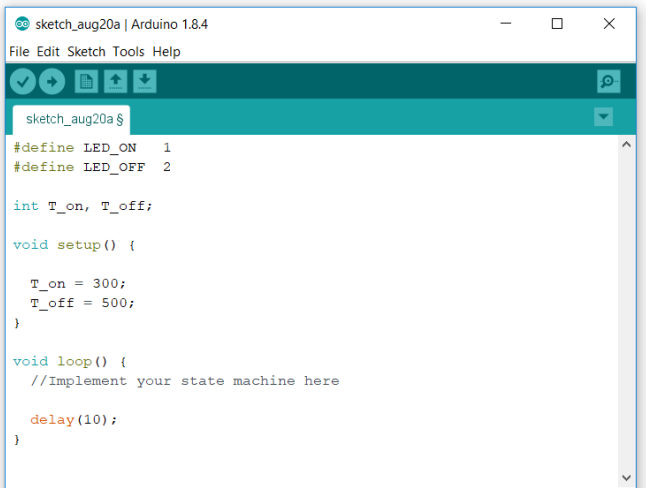


Figure 8: Template program to implement your DFA

```
#define LED_ON  1
#define LED_OFF 2
#define IDLING  3

unsigned long prevMillis = 0;

const long T_ON = 300, T_OFF = 500;
unsigned int interval;

short state;

void setup() {
  Serial.begin(9600);
```

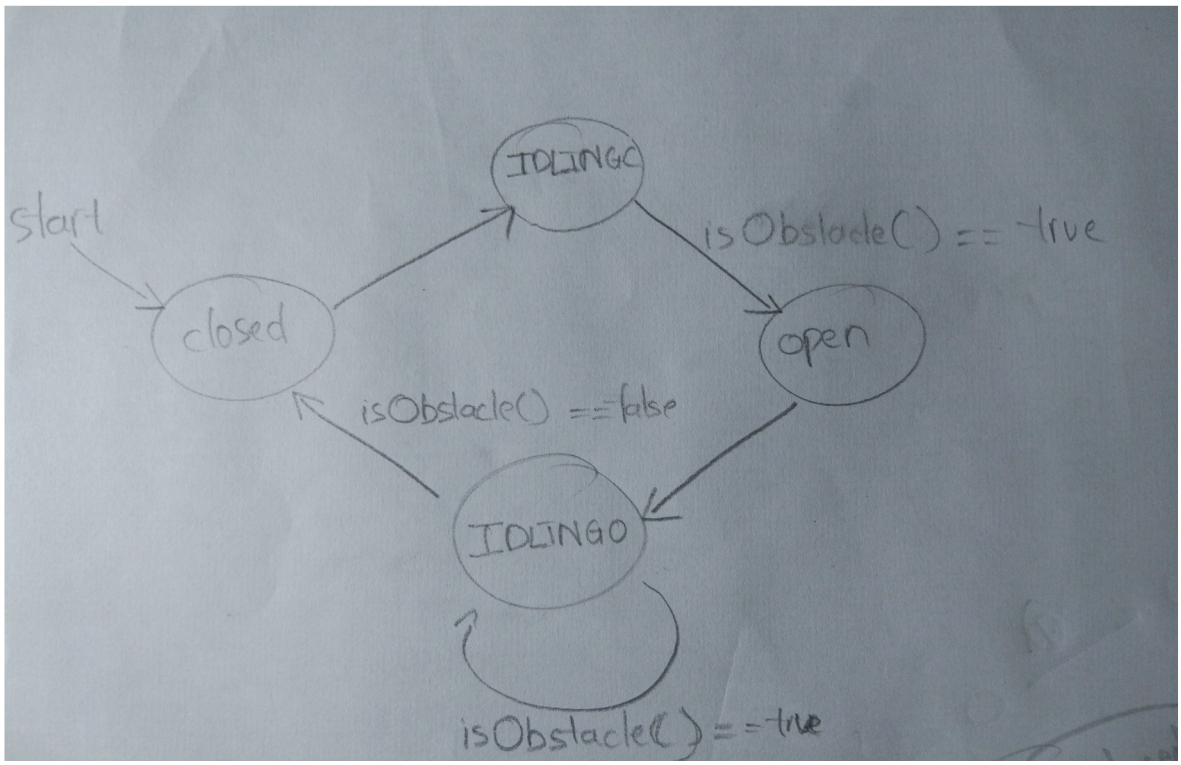
```
  pinMode(LED_BUILTIN, OUTPUT);

  state = IDLING;
  interval = T_ON;
}

void loop() {
  switch (state){
    case IDLING:{
      unsigned long currentMillis = millis();
      if (currentMillis - prevMillis >= interval){
        prevMillis = currentMillis;
        if (interval == T_ON){
          state = LED_OFF;
          interval = T_OFF;
        } else {
          state = LED_ON;
          interval = T_ON;
        }
      }
      break;
    }
     case LED_ON:{
      digitalWrite(LED_BUILTIN, LOW);
      state = IDLING;
      break;
     }
     case LED_OFF:{
      digitalWrite(LED_BUILTIN, HIGH);
      state = IDLING;
      break;
     }
  }
  delay(10);
}
```

IV. Extra Exercise

Assume that there is an automation door located at the entrance of an office. An obstacle sensor is normally used to detect if there is a person who wants to enter or exit the office. We use an LED to simulate the state of this door: LED-ON means the door is opened, LED-OFF means the door is closed. The obstacle sensor sends its value 0 or 1 and connected to pin numer 2 of the NodeMCU board. At the beginning, the door is closed.

If a person is detected closed to the door (obstacle sensor value is 1), the door will be opened for 3 seconds. After this period, if the sensor value is zero, the door is closed. Otherwise, the door is continually opened for 3 seconds more.

Design your state machine for the system and implement it on Arduino IDE. Please remember that there is only one delay(10) function is used and placed at the end of the loop() function.




```
#define CLOSED  1
#define OPEN    2
#define IDLINGO 3
#define IDLINGC 4

#define LED      16
#define OBSTACLE 4

#define INTERVAL 3000

short state = CLOSED;
```

```
unsigned long prevMillis = 0;


void setup(){
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(OBSTACLE, INPUT);
}


boolean isObstacle(){
  if (digitalRead(OBSTACLE) == HIGH)
    return true;
  else
    return false;
}


void loop(){
  switch (state){
    case IDLINGO:{
      unsigned long currentMillis = millis();
      if (currentMillis - prevMillis >= INTERVAL){
        prevMillis = currentMillis;
        if (!isObstacle())
          state = CLOSED;
      }
      break;
    }
    case IDLINGC:{
      if (isObstacle())
        state = OPEN;
      break;
    }
    case CLOSED:{
      Serial.println("Closed");
      digitalWrite(LED, LOW);
```

```
      state = IDLINGC;
      break;
    }
    case OPEN:{
      Serial.println("Open");
      digitalWrite(LED, HIGH);
      state = IDLINGO;
      break;
    }
  }
  delay(10);
}
```