ELEC-C7420 - Basic principles in networking

Tutorial on basic Arduino operation

Introduction

For some of the exercise sessions we will be using the Arduino UNO device to perform basic operations of security in networking like encryption/decryption, authentication and secure communications taking advantage of the WiFi functionalities available in the device. For this course we will use the following version of Arduino:

(https://store.arduino.cc/mkr-wifi-1010)

The MKR WIFI 1010 is a significant improvement on the MKR 1000 WIFI. It's equipped with an ESP32 module made by U-BLOX. This board aims to speed up and simplify the prototyping of WiFi based IoT applications thanks to the flexibility of the ESP32 module and its low power consumption.

The board is composed of three main blocks:

- SAMD21 Cortex-M0+ 32bit Low Power ARM MCU;
- U-BLOX NINA-W10 Series Low Power 2.4GHz IEEE® 802.11 b/g/n Wi-Fi; and
- ECC508 Crypto Authentication.

The MKR WIFI 1010 includes 32-bit computational power, the usual rich set of I/O interfaces, and low power Wi-Fi with a Cryptochip for secure communication using SHA-256 encryption. Plus, it offers ease of use Arduino Software (IDE) for code development and programming. All of these features make this board the preferred choice for the emerging IoT battery-powered projects in a compact form.

Its USB port can be used to supply power (5V) to the board. It has a Li-Po charging circuit that allows the Arduino MKR WIFI 1010 to run on battery power or an external 5 volt source, charging the Li-Po battery while running on external power. Switching from one source to the other is done automatically.

Warning: Unlike most Arduino boards, the MKR WIFI 1010 runs at 3.3V. The maximum voltage that the I/O pins can tolerate is 3.3V. Applying voltages higher than 3.3V to any I/O pin could damage the board. While output to 5V digital devices is possible, bidirectional communication with 5V devices needs proper level shifting.

Following the technical specifications:

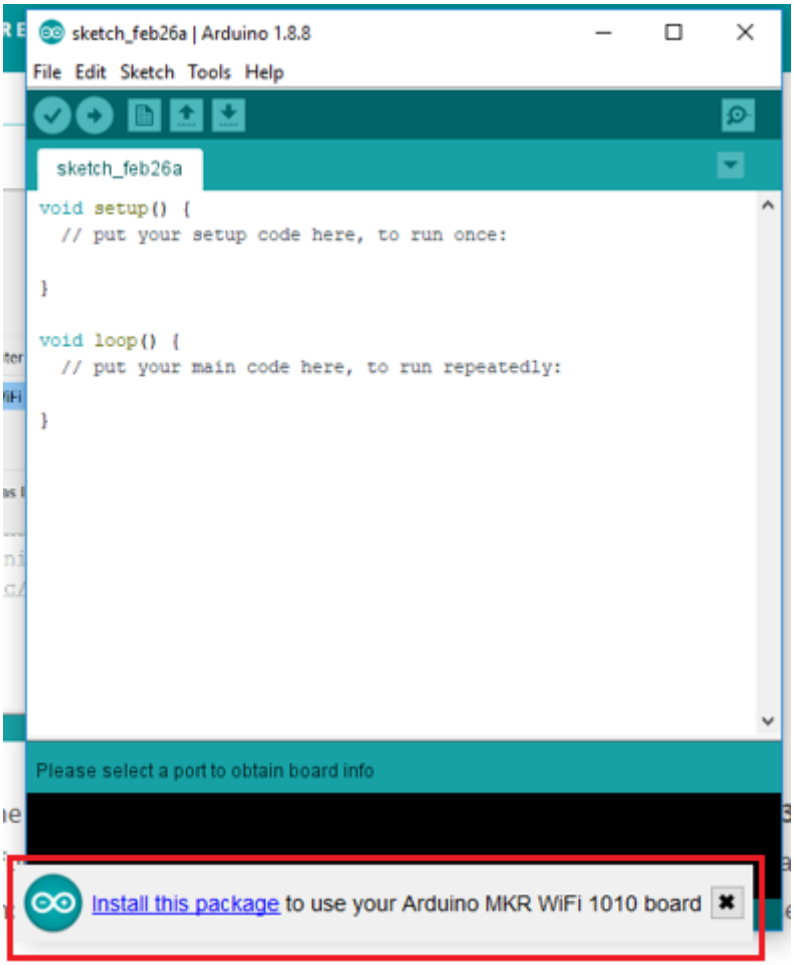
Microcontroller	SAMD21 Cortex-M0+ 32bit Low Power ARM MCU
Board Power Supply (USB/VIN)	5V
Supported Battery(*)	Li-Po Single Cell, 3.7V, 700mAh Minimum
Circuit Operating Voltage	3.3V
Digital I/O Pins	8
PWM Pins	12 (0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3 - or 18 -, A4 - or 19)
UART	1
SPI	1
I2C	1
I2S	1
Connectivity	WiFi
Analog Input Pins	7 (ADC 8/10/12 bit)

Analog Output Pins	1 (DAC 10 bit)
External Interrupts	8 (0, 1, 4, 5, 6, 7, 8, A1 -or 16-, A2 - or 17)
DC Current per I/O Pin	7 mA
Flash Memory	256 KB
SRAM	32 KB
EEPROM	No
Clock Speed	32.768 kHz (RTC), 48 MHz
LED_BUILTIN	6
Full-Speed USB Device and Embedded Host	Included
LED_BUILTIN	6
Length	61.5 mm
Width	25 mm
Weight	32 gr.

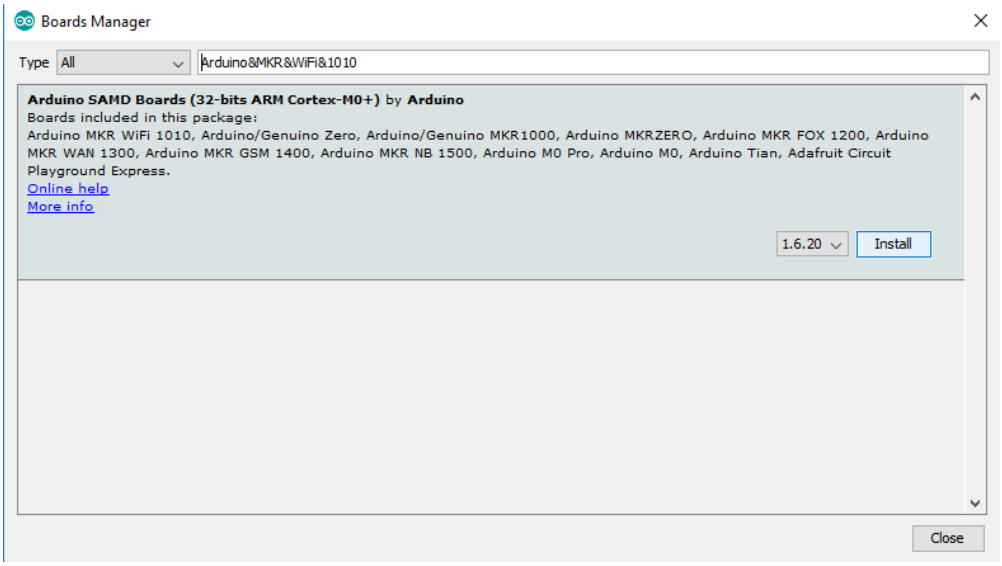
Setting up the device
Install the Arduino IDE from the following link: https://www.arduino.cc/en/Main/Software

After installing the IDE connect the device to the laptop and run the Arduino IDE.

If you don’t have the MKR module installed, when you start the program you will be asked to install the corresponding module:

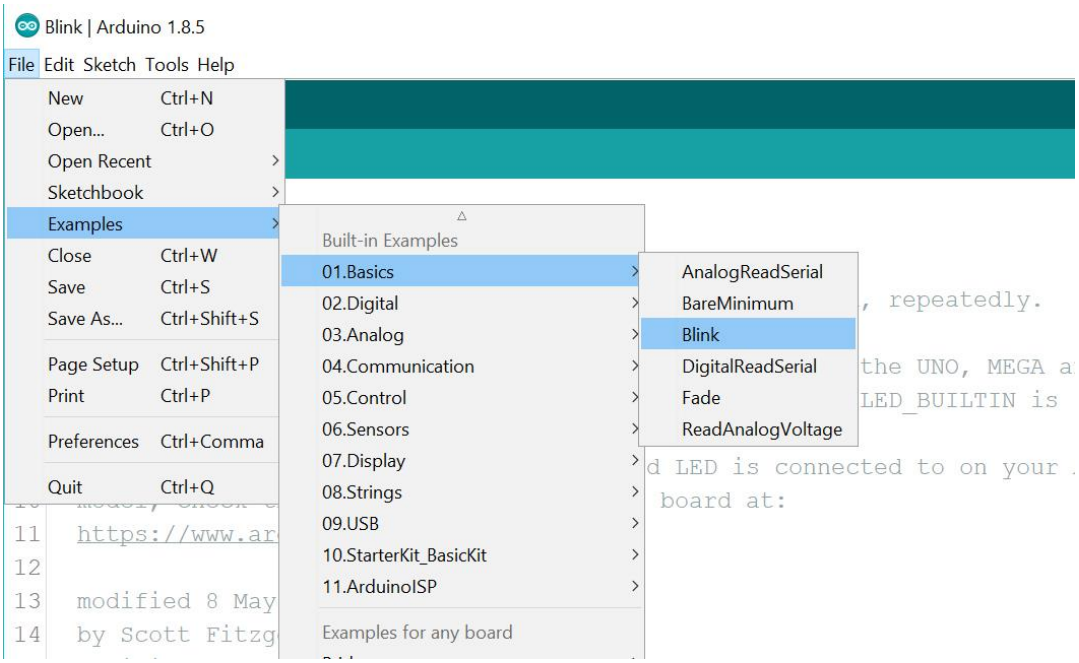


Then install the package:

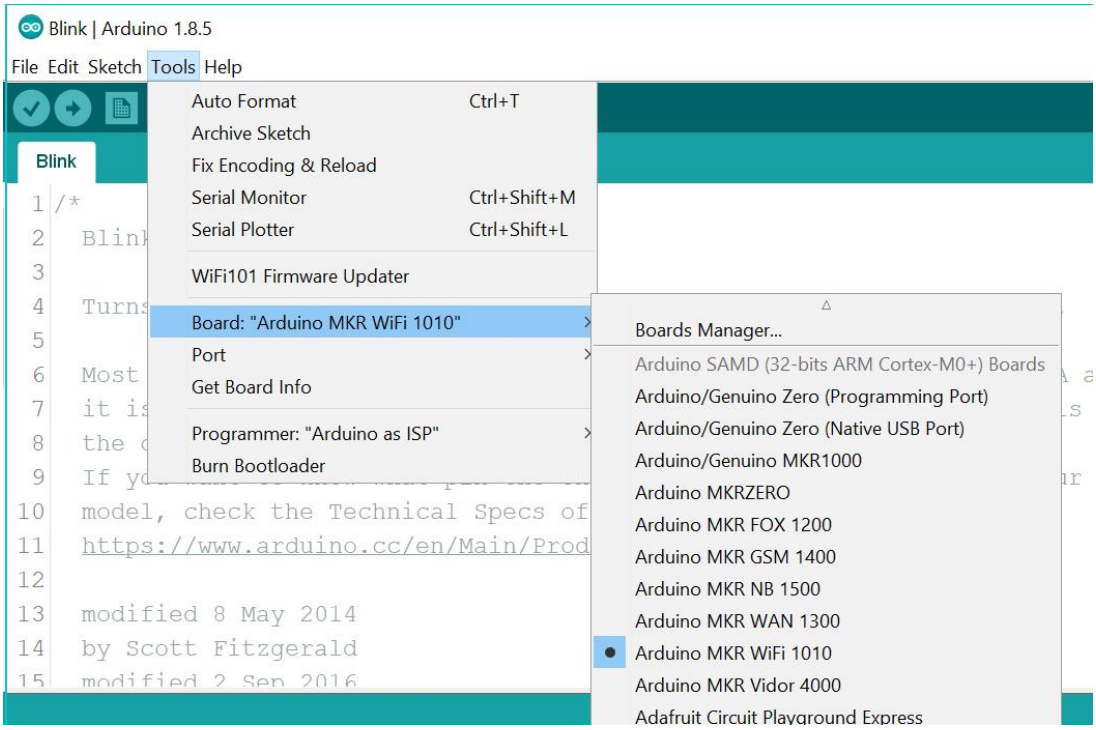


- Additionally install the Wifi101 and WifiNINA library

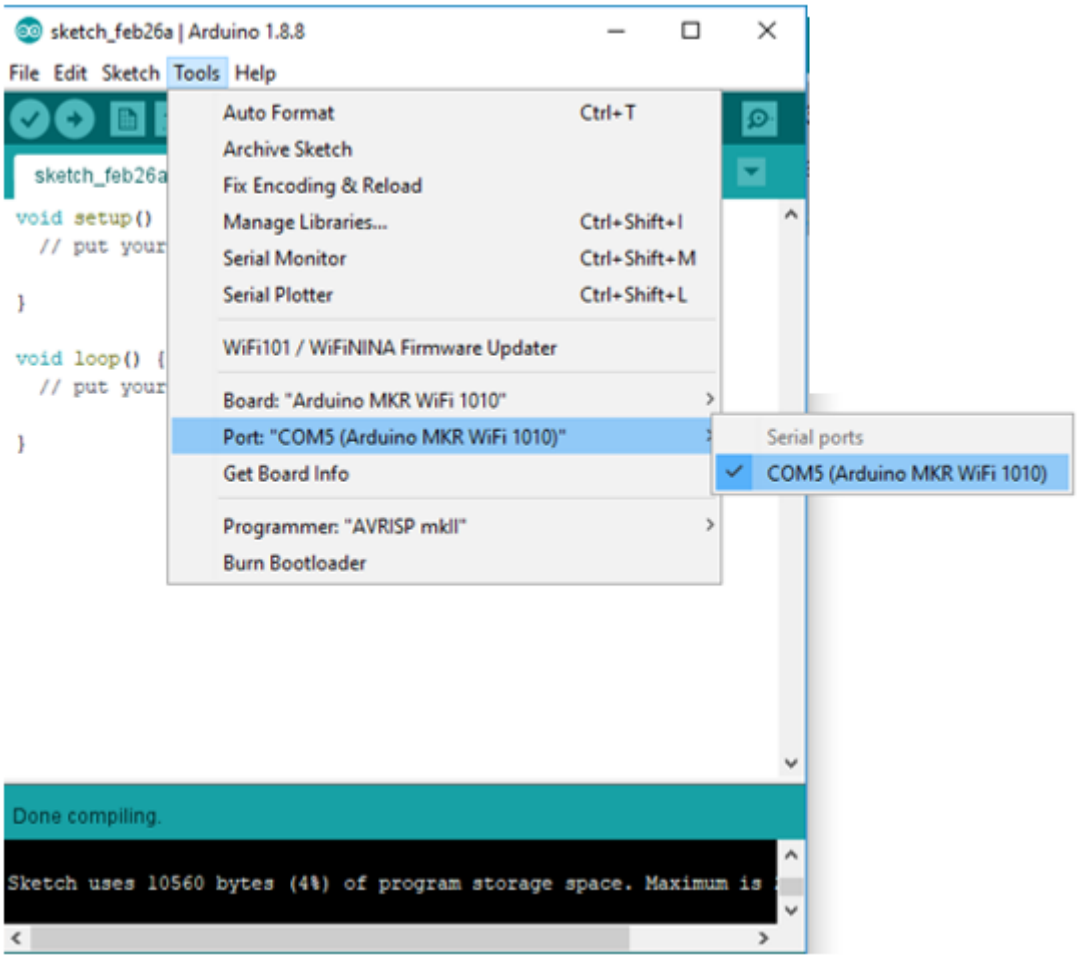
Test the device:
Now that the programming environment is set we can test the device with the basic example of blinking the led on the board:



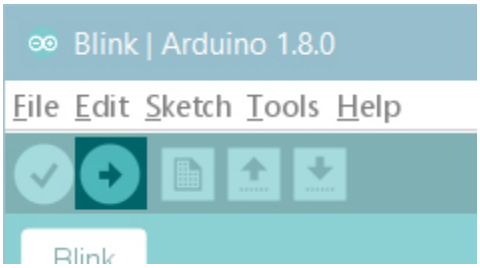
You'll need to select the entry in the **Tools > Board** menu that corresponds to your Arduino board.



Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port. (In my case was COM5)



Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



A few seconds after the upload finishes, you should see the on-board LED start to blink (in orange). If it does, congratulations! You've gotten your MKR WIFI 1010 up-and-running.

Practice:

Now we will try to connect to a WiFi network and obtain the IP address assigned to the Arduino device:

```
#include <SPI.h>
#include <WiFiNINA.h>

//#include "arduino_secrets.h"
///////please enter your sensitive data in the Secret tab/arduino_secrets.h
char ssid[] = "";        // your network SSID (name)
char pass[] = "";    // your network password (use for WPA, or use as key for WEP)

int keyIndex = 0;            // your network key Index number (needed only for WEP)

int status = WL_IDLE_STATUS;

WiFiServer server(23);

boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  //Initialize serial and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // check for the WiFi module:
  if (WiFi.status() == WL_NO_MODULE) {
    Serial.println("Communication with WiFi module failed!");
    // don't continue
    while (true);
  }
  String fv = WiFi.firmwareVersion();
  if (fv < "1.0.0") {
    Serial.println("Please upgrade the firmware");
  }

  // attempt to connect to Wifi network:
  while (status != WL_CONNECTED) {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);
    // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
    status = WiFi.begin(ssid, pass);

    // wait 10 seconds for connection:
    delay(10000);
  }

  // start the server:
  server.begin();
  // you're connected now, so print out the status:
  printWifiStatus();
}

void loop() {

}
```

```
void printWifiStatus() {
  // print the SSID of the network you're attached to:
  Serial.print("SSID: ");
  Serial.println(WiFi.SSID());

  // print your board's IP address:
  IPAddress ip = WiFi.localIP();
  Serial.print("IP Address: ");
  Serial.println(ip);

  // print the received signal strength:
  long rssi = WiFi.RSSI();
  Serial.print("signal strength (RSSI):");
  Serial.print(rssi);
  Serial.println(" dBm");
}
```