

التدريب الصيفي 2
SUMMER TRAINING 2

التدريب التقني المتقدم لمعمل إنترنت الأشياء

لصيف ١٤٤١ هـ - ٢٠٢٠

يوم الإثنين 2020/06/15

وادي طيبة
TAIBAH VALLEY



📍 saudi arabia | madina

☎ +966 14 861 8888

🌐 taibahvalley.com.sa

📱 @taibahvalley

References

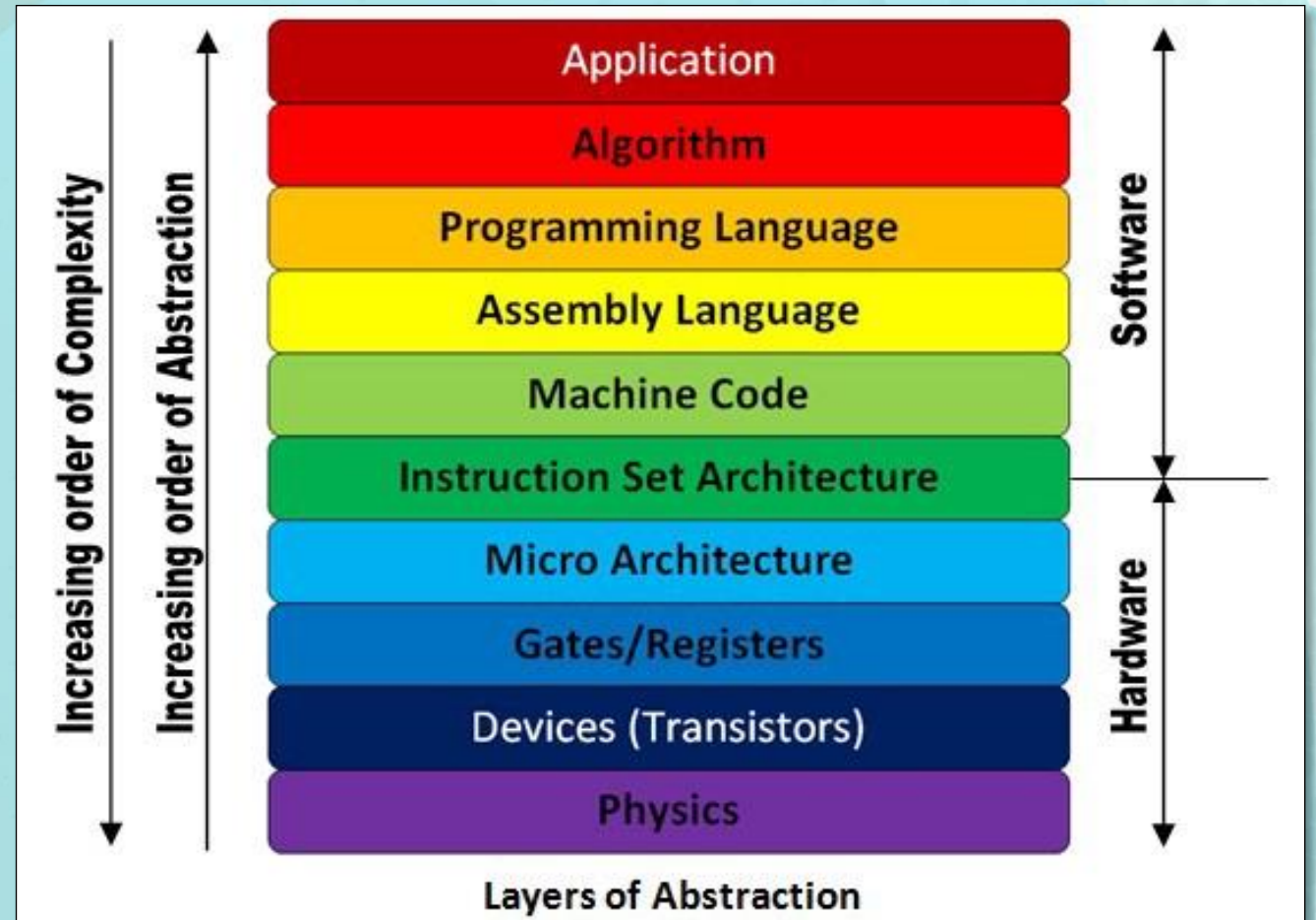
- <https://www.arduino.cc/en/Guide/ArduinoUno>
- <https://www.arduino.cc/en/main/products>
- <https://create.arduino.cc/projecthub>
- <https://www.hackster.io/arduino/projects>
- Book: Arduino Cookbook



📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

Abstraction

- Have you ever cared about how many volts the CPU is running on when it executes your program?
- When you write your code, do you need to know which register it's going to use or what part of the RAM it's going to reserve?





 @taibahvalley

SUMMER TRAINING

Arduino Uno's microcontroller (Atmega328P)

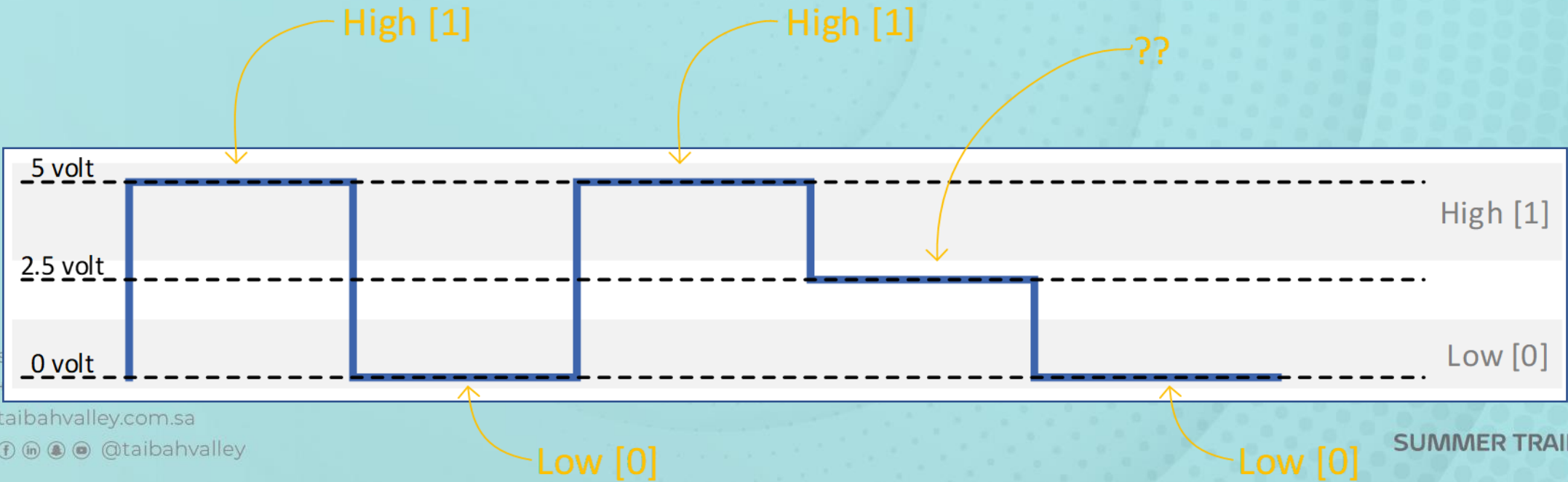
Table 30-1. Common DC characteristics $T_A = -40^{\circ}\text{C}$ to 105°C , $V_{CC} = 1.8\text{V}$ to 5.5V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, except XTAL1 and RESET pin	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$	-0.5 -0.5		$0.2V_{CC}^{(1)}$ $0.3V_{CC}^{(1)}$	V
V_{IH}	Input High Voltage, except XTAL1 and RESET pins	$V_{CC} = 1.8\text{V} - 2.4\text{V}$ $V_{CC} = 2.4\text{V} - 5.5\text{V}$	$0.7V_{CC}^{(2)}$ $0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$ $V_{CC} + 0.5$	V

Say, we use 5 volts => $V_{CC} = 5\text{ volt}$

Input **Low**: -0.5v to $(5\text{v} * 0.3 = +1.5\text{v})$

Input **High**: $(5\text{v} * 0.6 = +3\text{v})$ to $(5\text{v} + 0.5 = +5.5\text{v})$



What is Arduino?

Arduino is an **open-source electronics prototyping** platform based on flexible, **easy-to-use hardware and software**. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.



📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

SUMMER TRAINING

Arduino Overview

- Hardware Overview (Pinout, Models, ...)
- Arduino IDE
- Review basic programming concepts (loops, conditions, functions, ...)
- Libraries
- Pins interfacing (digital, analog, ...)
- Timing functions



Official Arduino Boards



Uno



Mega 2560



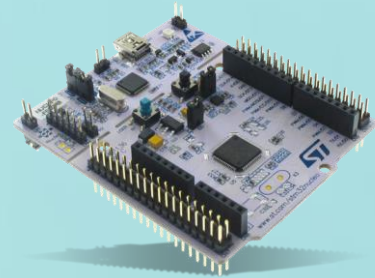
Due



MKR1000

	Uno	Mega 2560	Due	MKR1000
Processor	ATmega328P	ATmega2560	ATSAM3X8E (ARM)	SAMD21 (ARM)
I/O Voltage	5 volt	5 volt	3.3 volt	3.3 volt
CPU Speed	16 MHz	16 MHz	84 MHz	48 MHz
Digital IO	14	54	54	8
Analog In	6	16	12	7
Analog Out	0	0	2	1
UART	1	4	4	1
EEPROM	1kB	4kB	-	-
SRAM	2kB	8kB	96kB	32kB
FLASH	32kB	256kB	512kB	256kB

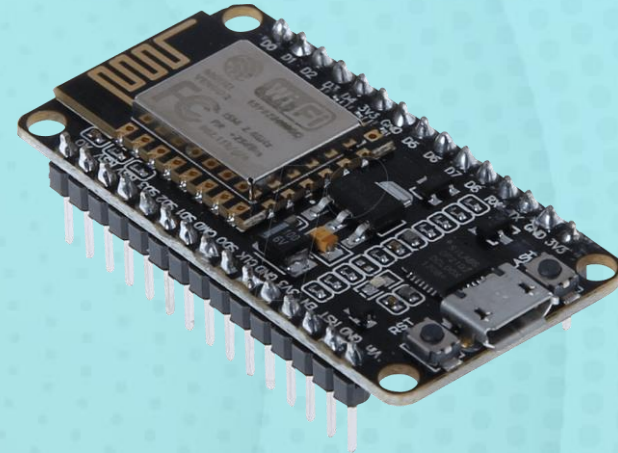
Arduino Compatible Boards



STM32 Nucleo Boards



TI LaunchPad



NodeMCU



📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

SUMMER TRAINING

Arduino Uno Board Overview

- MCU itself
- Voltage IN and voltage regulator
- 5V, 3.3V
- Current overdraw protection
- Reset Button and signal
- DC jack
- On-board LED, pin 13
- Digital IO
- Analog IO
- Flashing IC
- Clock

📍 Saudi Arabia, madina

☎ +966 14 861 8888

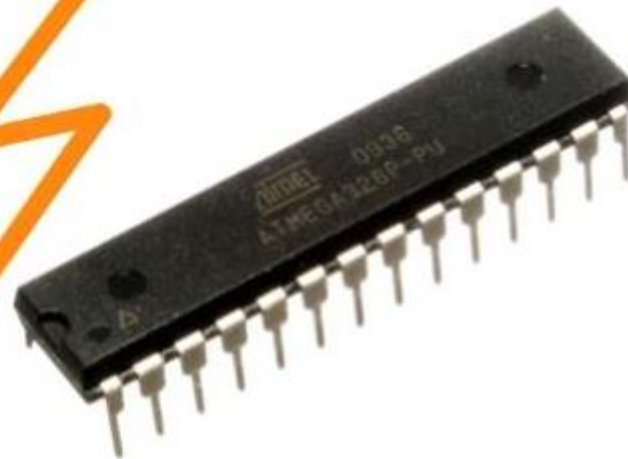
🌐 taibahvalley.com.sa

🐦 📺 🌐 📷 📱 @taibahvalley



TQFP-32

DIP-28

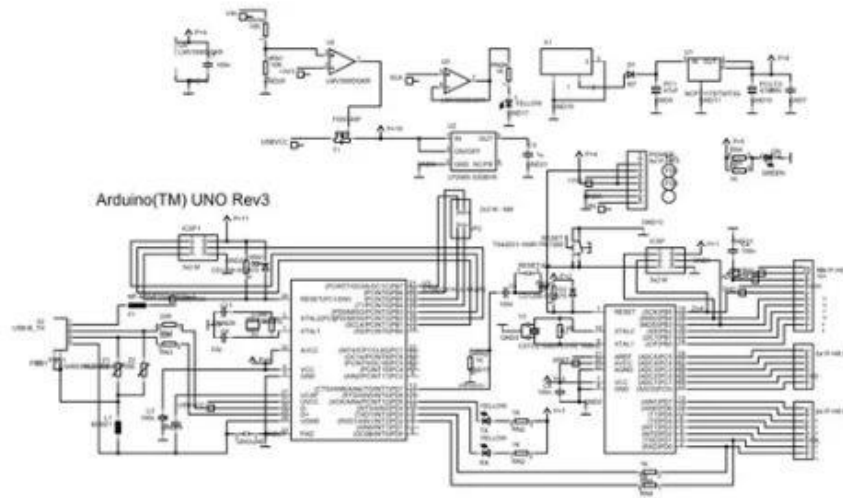


📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

SUMMER TRAINING

Hardware Overview

Step 2: SCHEMATICS:-



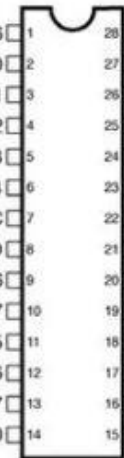
Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Atmega168 Pin Mapping

Arduino function

reset
digital pin 0 (RX)
digital pin 1 (TX)
digital pin 2
digital pin 3 (PWM)
digital pin 4
VCC
GND
crystal
crystal
digital pin 5 (PWM)
digital pin 6 (PWM)
digital pin 7
digital pin 8

(PCINT14/RESET) PC6
(PCINT16/RXD) PD0
(PCINT17/TXD) PD1
(PCINT18/INT0) PD2
(PCINT19/OC2B/INT1) PD3
(PCINT20/XCK/T0) PD4
VCC
GND
(PCINT6/XTAL1/TOSC1) PB6
(PCINT7/XTAL2/TOSC2) PB7
(PCINT21/OC0B/T1) PD5
(PCINT22/OC0A/AIN0) PD6
(PCINT23/AIN1) PD7
(PCINT0/CLKO/ICP1) PB0



Arduino function

analog input 5
analog input 4
analog input 3
analog input 2
analog input 1
analog input 0
GND
analog reference
VCC
digital pin 13
digital pin 12
digital pin 11 (PWM)
digital pin 10 (PWM)
digital pin 9 (PWM)

PC5 (ADC5/SCL/PCINT13)
PC4 (ADC4/SDA/PCINT12)
PC3 (ADC3/PCINT11)
PC2 (ADC2/PCINT10)
PC1 (ADC1/PCINT9)
PC0 (ADC0/PCINT8)
GND
AREF
AVCC
PB5 (SCK/PCINT5)
PB4 (MISO/PCINT4)
PB3 (MOSI/OC2A/PCINT3)
PB2 (SS/OC1B/PCINT2)
PB1 (OC1A/PCINT1)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Nothing but a library to include

Your Arduino Sketch
(User application)

Arduino Libraries

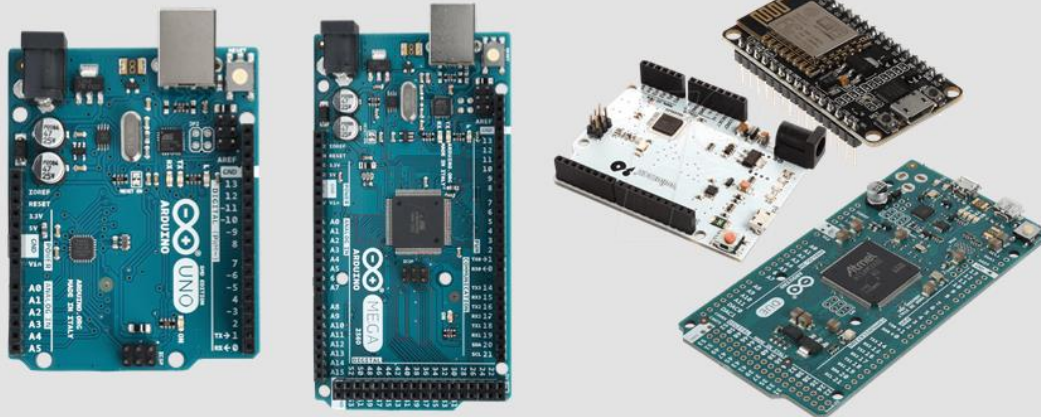
Arduino

HAL

Software

Hardware

This library **abstracts** and **simplifies** some of the complexity of the hardware, giving you a **unified API** to use when you need to interact with the hardware.



This how it looks without Arduino

```
/* PORTF data register */
#define PORTFDAT (*(volatile unsigned int*)0x400253FC)
/* PORTF data direction register */
#define PORTFDIR (*(volatile unsigned int*)0x40025400)
/* PORTF digital enable register */
#define PORTFDEN (*(volatile unsigned int*)0x4002551C)
/* run mode clock gating register */
#define RCGCGPIO (*(volatile unsigned int*)0x400FE608)
/* coprocessor access control register */
#define SCB_CPAC (*(volatile unsigned int*)0xE000ED88)

void delayMs(int n);    /* function prototype for delay */

int main(void)
{
    /* enable clock to GPIOF at clock gating register */
    RCGCGPIO |= 0x20;
    /* set PORTF pin3-1 as output pins */
    PORTFDIR = 0x0E;
    /* set PORTF pin3-1 as digital pins */
    PORTFDEN = 0x0E;

    while(1)
    {
        /* write PORTF to turn on all LEDs */
        PORTFDAT = 0x0E;
        delayMs(500);
        /* write PORTF to turn off all LEDs */
        PORTFDAT = 0;
        delayMs(500);
    }

    /* delay n milliseconds (16 MHz CPU clock) */
    void delayMs(int n)
    {
        int i, j;
        for(i = 0 ; i < n; i++)
            for(j = 0; j < 3180; j++)
                {} /* do nothing for 1 ms */
    }
}
```

Tip: This is not the best implementation, but it's the easiest to toggle an LED!

This how it looks with Arduino

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

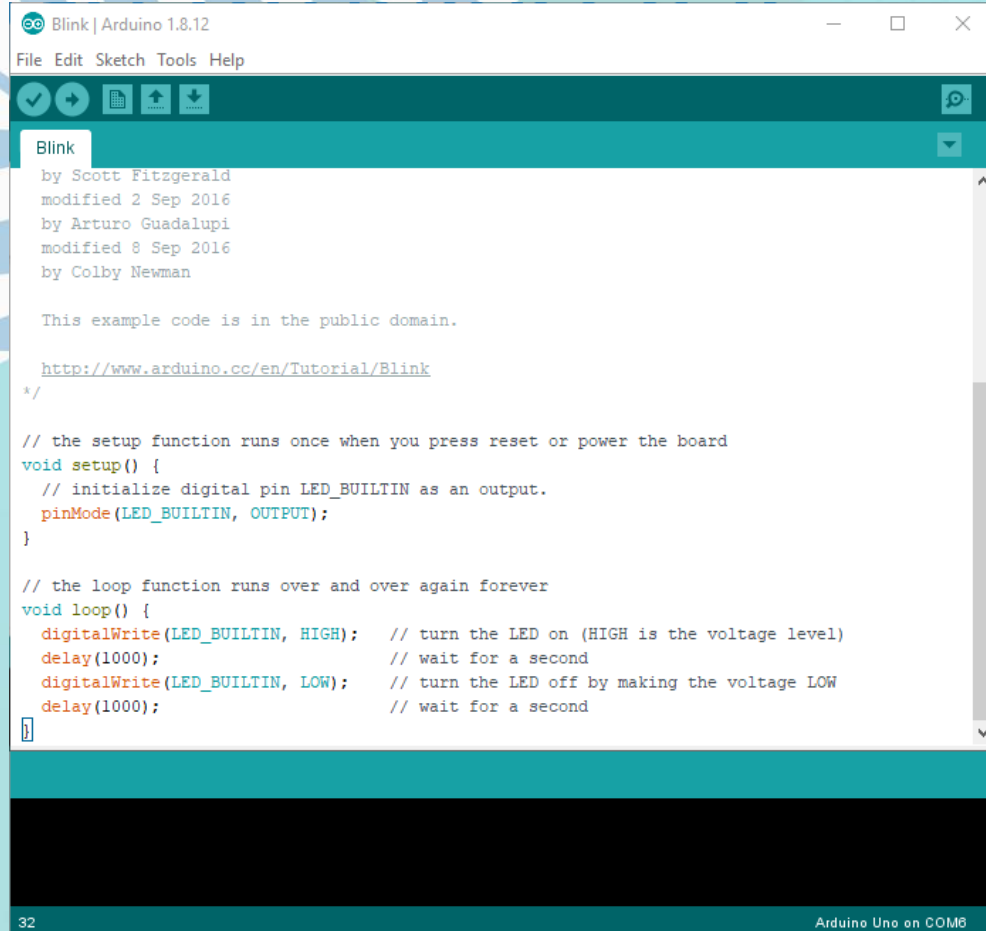
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(500);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(500);                      // wait for a second
}
```



📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

SUMMER TRAINING

Download Arduino IDE



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.8.12". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area displays the "Blink" example code, which is attributed to Scott Fitzgerald, Arturo Guadalupi, and Colby Newman. The code is in the public domain and is available at <http://www.arduino.cc/en/Tutorial/Blink>. The code defines a setup function to initialize the LED_BUILTIN pin as an output and a loop function to turn the LED on and off every 1000 milliseconds. The status bar at the bottom indicates "32" and "Arduino Uno on COM6".

```
Blink
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman

This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

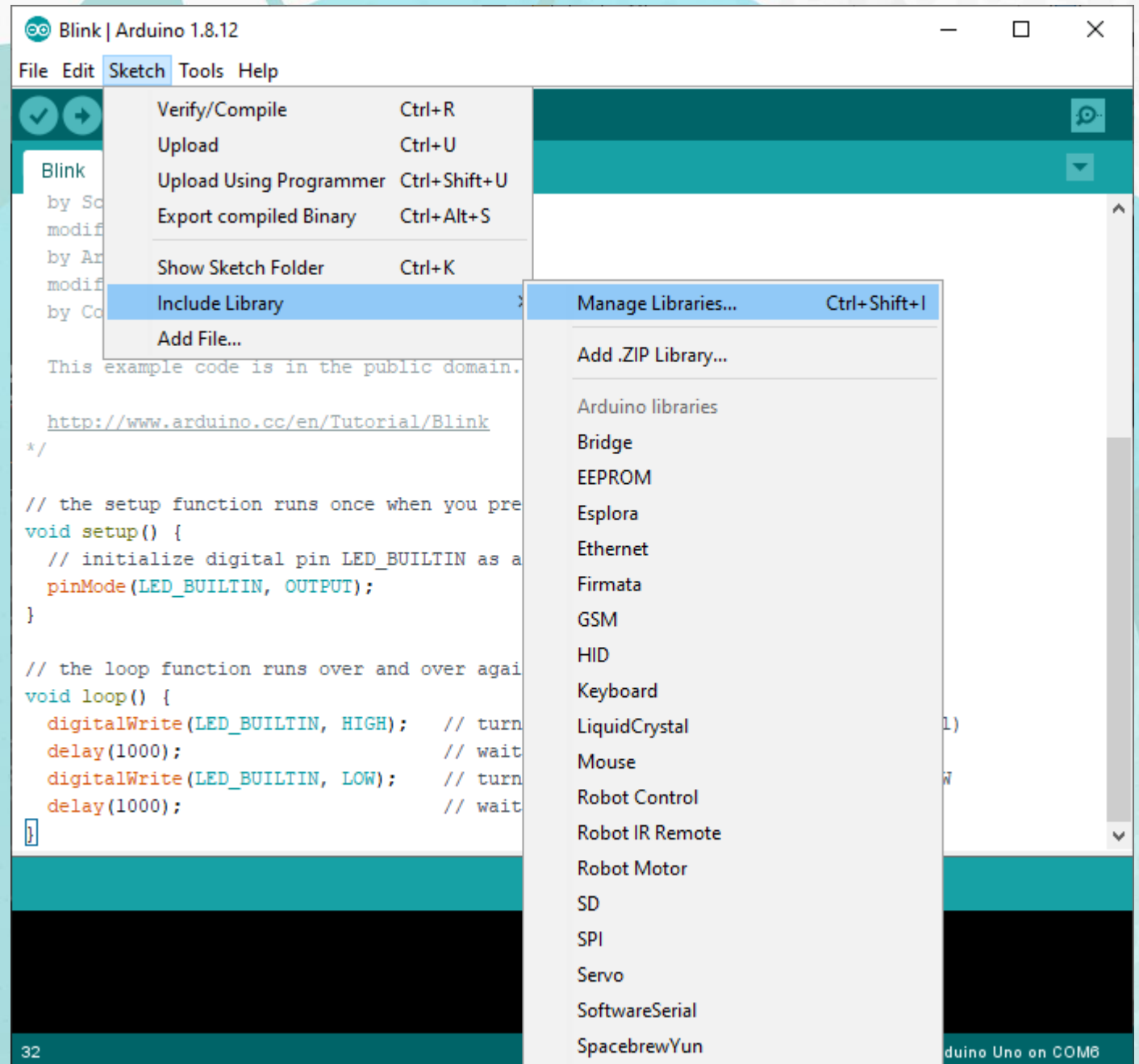


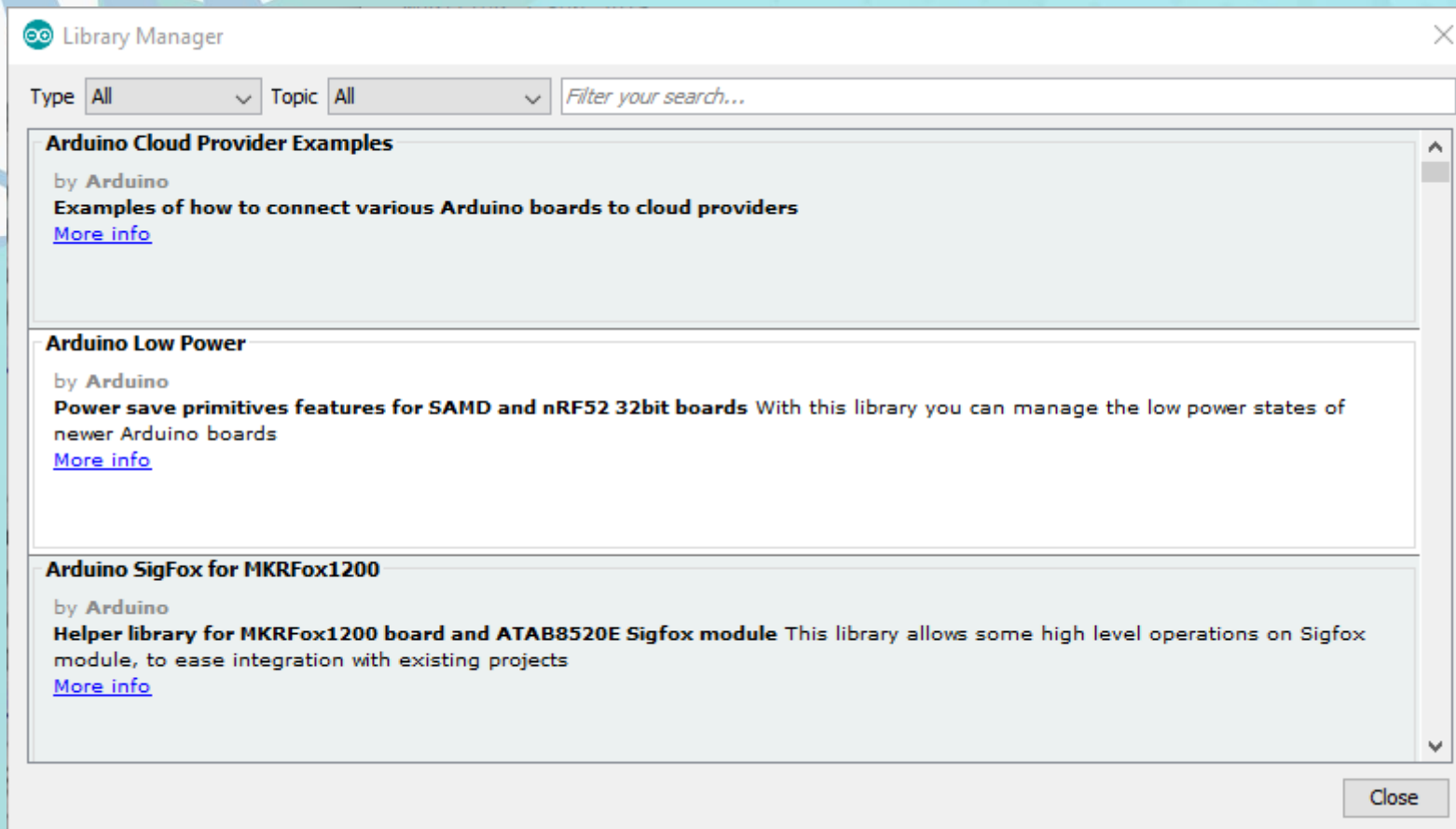
Libraries

The Arduino environment can be extended through the use of libraries, just like most programming platforms.

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data.

To use a library in a sketch, select it from Sketch > Import Library.





📍 saudi arabia | madina
☎ +966 14 861 8888
🌐 taibahvalley.com.sa
📱 @taibahvalley

SUMMER TRAINING



Arduino programming

- Setup() and loop() functions
- Basic functions to use
 - Serial.begin()
 - Serial.print() & Serial.println()
 - Delay() & DelayMicroseconds()
 - PinMode()
 - DigitalWrite()
 - DigitalRead()
 - AnalogRead()
 - AnalogWrite()



📍 saudi arabia | madina

☎ +966 14 861 8888

🌐 taibahvalley.com.sa

🐦 📘 🌐 📺 @taibahvalley

SUMMER TRAINING

Tasks

- Install Arduino IDE
- Hello world printing on serial
- Counter printer
- Timer print
- Read Button state
- Arduino voltmeter
- Serial plotter
- Timed Arduino LED\DC Motor circuits
- On Button Pressed LED\DC Motor pulses using Arduino

