Introduction and Programming Arduino

Arduino

- **What is Arduino?**

- A microcontroller board, contains on-board power supply, USB port to communicate with PC, and an Atmel microcontroller chip.

- It simplify the process of creating any control system by providing the standard board that can be programmed and connected to the system without the need to any sophisticated PCB design and implementation.

- It is an open source hardware, any one can get the details of its design & modify it or make his own one himself.

What can it do?

Sensors (to sense stuff)

A sensor is a device that detects and responds to a specific input, such as light, temperature, pressure, or motion and converts it into a measurable output.

- Touch pads.
- Photoresistors (sensing light levels)
- Thermistors (temperature)

What can it do?

Actuators (to do stuff)

An actuator is a device that receives an energy input and converts it into motion ,force or other also and is an essential component in many modern technologies and engineering fields.

- Lights, LED’s
- Motors
- Speakers
- Displays (LCD)

Why Arduino?

It is Open Source, both in terms of Hardware and Software.

It is cheap(1300 Rs), the hardware can be built from components or a prefab board can be purchased for approx. 900 Rs.

It can communicate with a computer via serial connection over USB.

It can be powered from USB or standalone DC power.

Why Arduino?

It can run standalone from a computer (chip is programmable) and it has memory (a small amount).

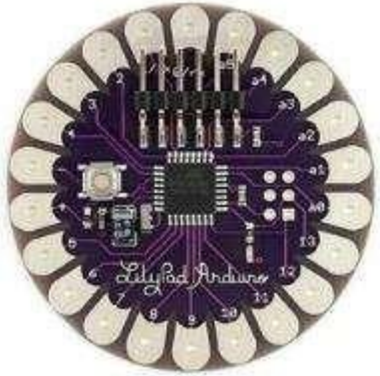
It can work with both Digital and Analog electronic signals. Sensors and Actuators.

You can make cool stuff! Some people are even making simple robots.

Different types of Arduino boards:

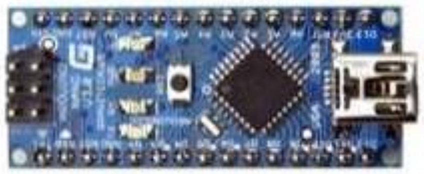
UNO

Mega



LilyPad

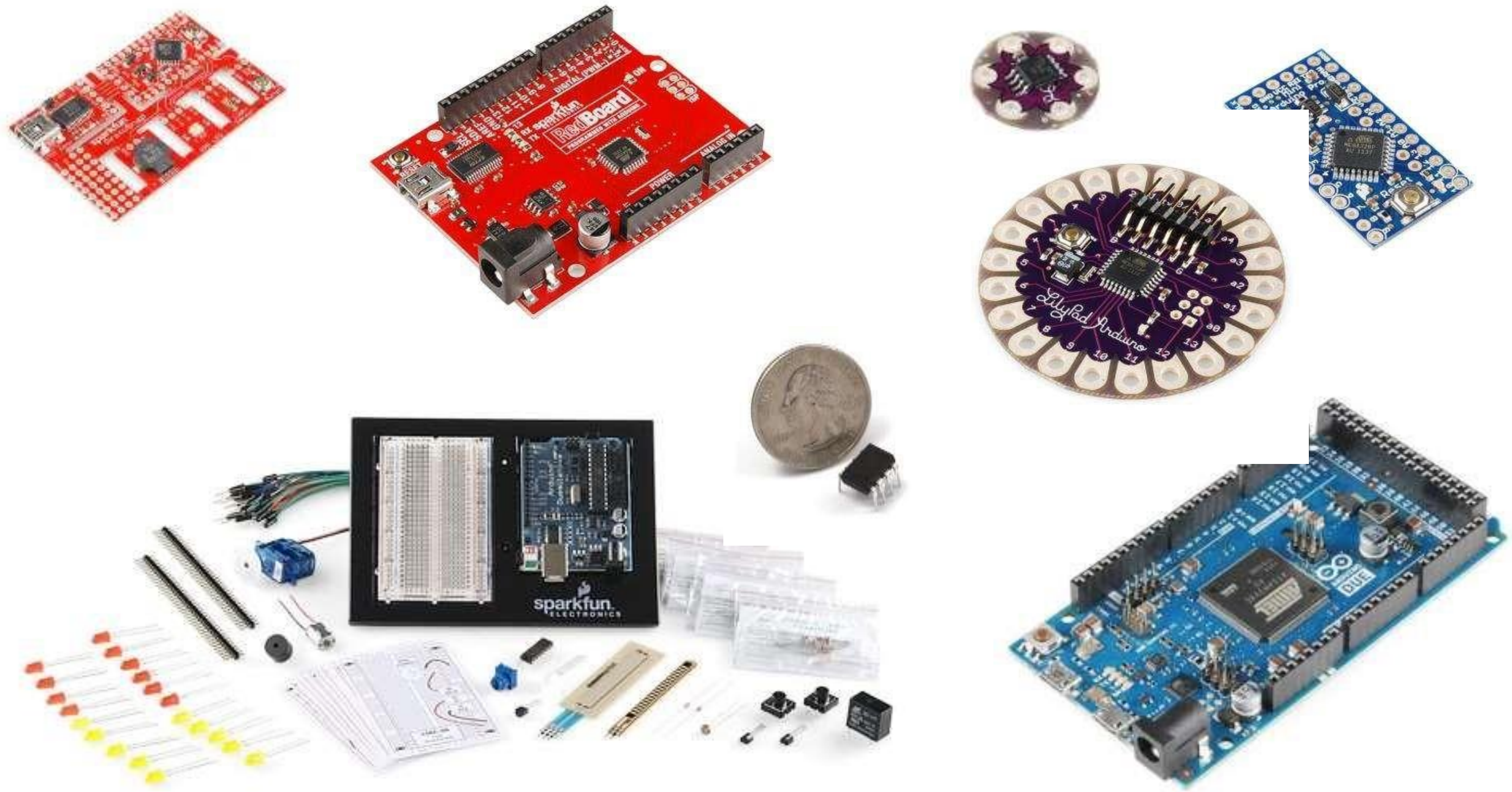
Arduino BT



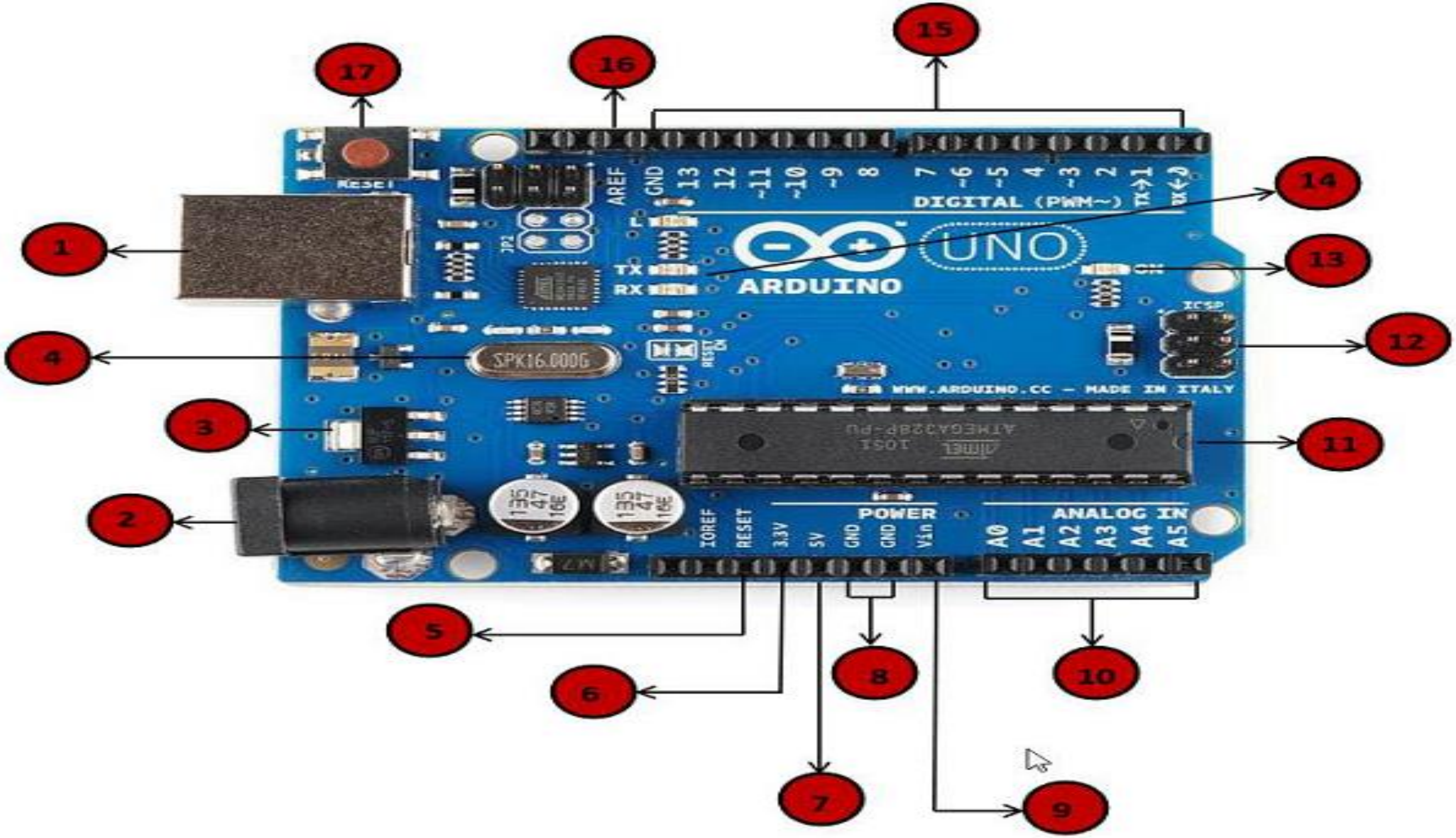
Arduino Nano

Arduino Mini

Arduino & Arduino compatible boards:

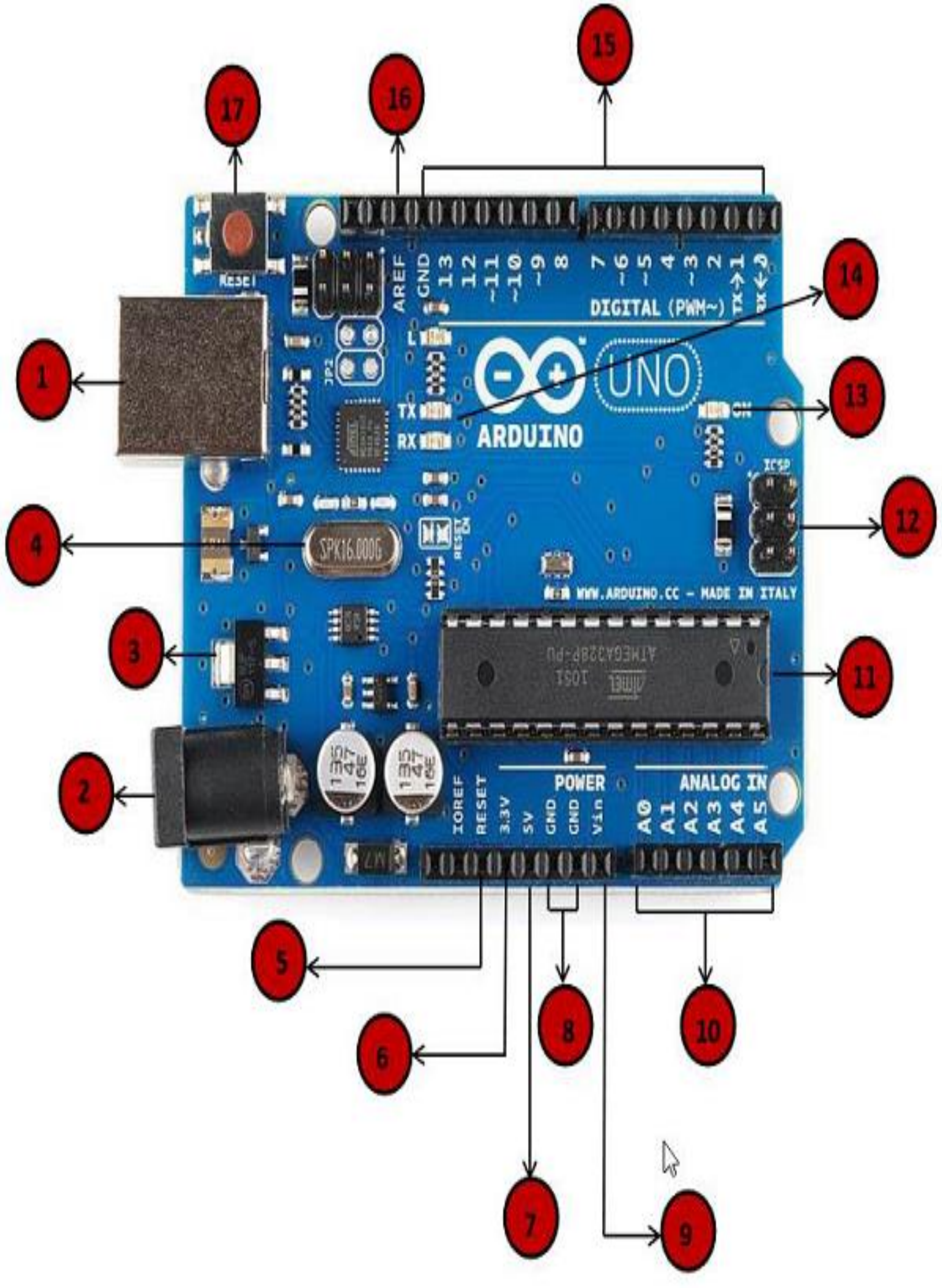


Arduino Uno Board Description



Arduino Uno Board Description (Cont..)

1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.



Arduino Uno Board Description (Cont..)

5,17

Arduino Reset

You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).

6,7 8,9

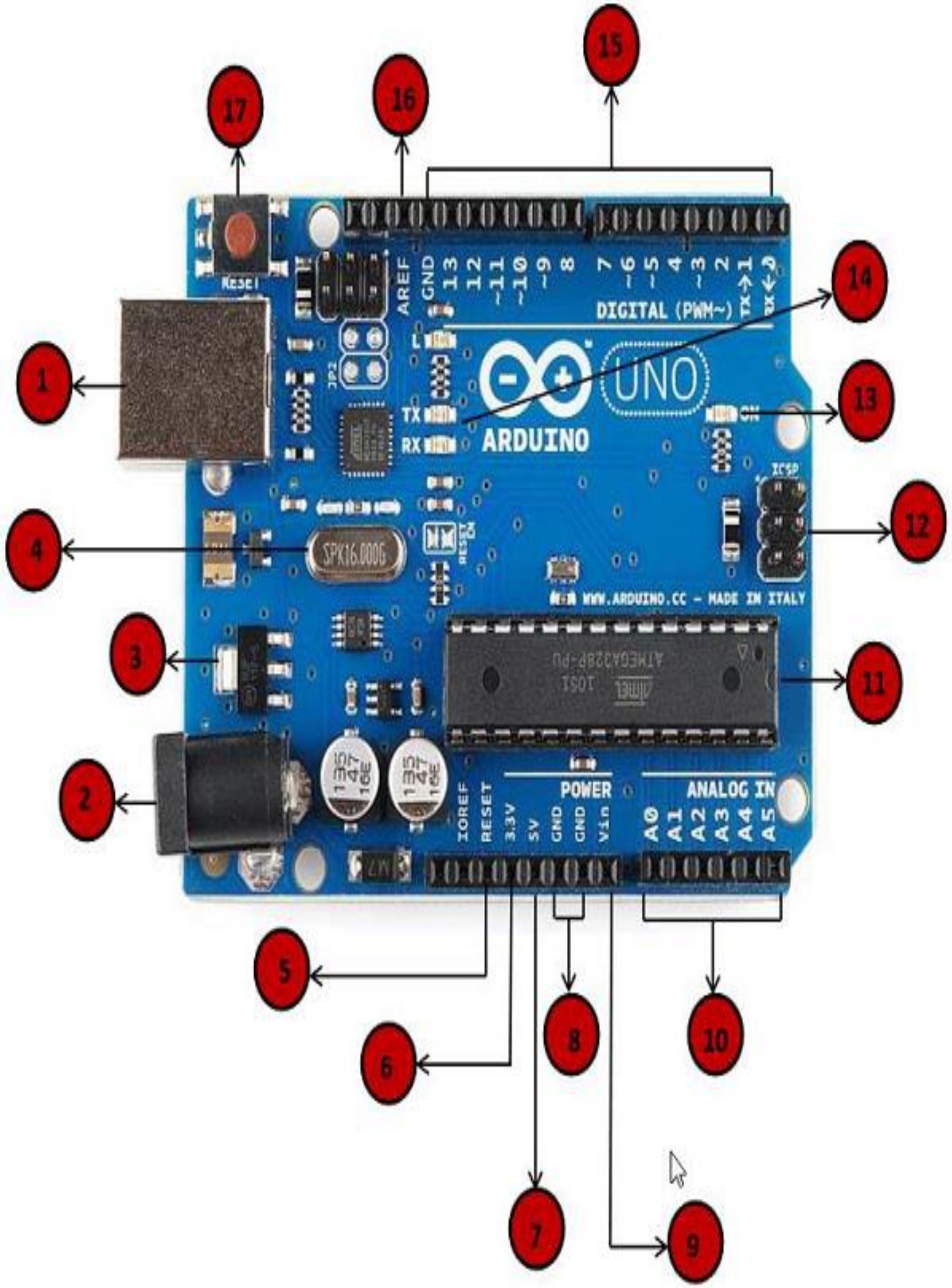
Pins (3.3, 5, GND, Vin)

- 3.3V (6) − Supply 3.3 output volt
- 5V (7) − Supply 5 output volt
- Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.
- GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

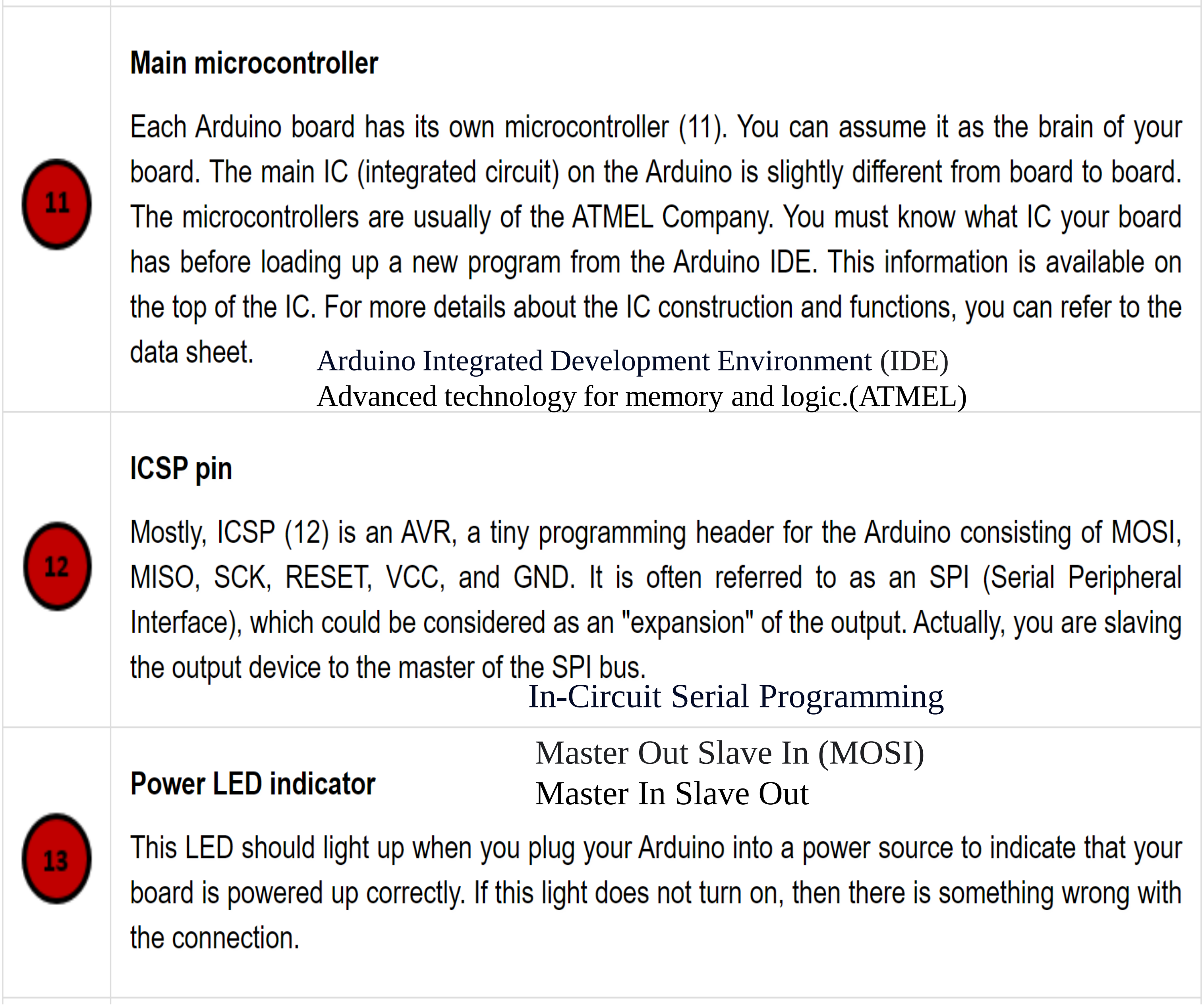
10

Analog pins

The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.



Arduino Uno Board Description (Cont..)

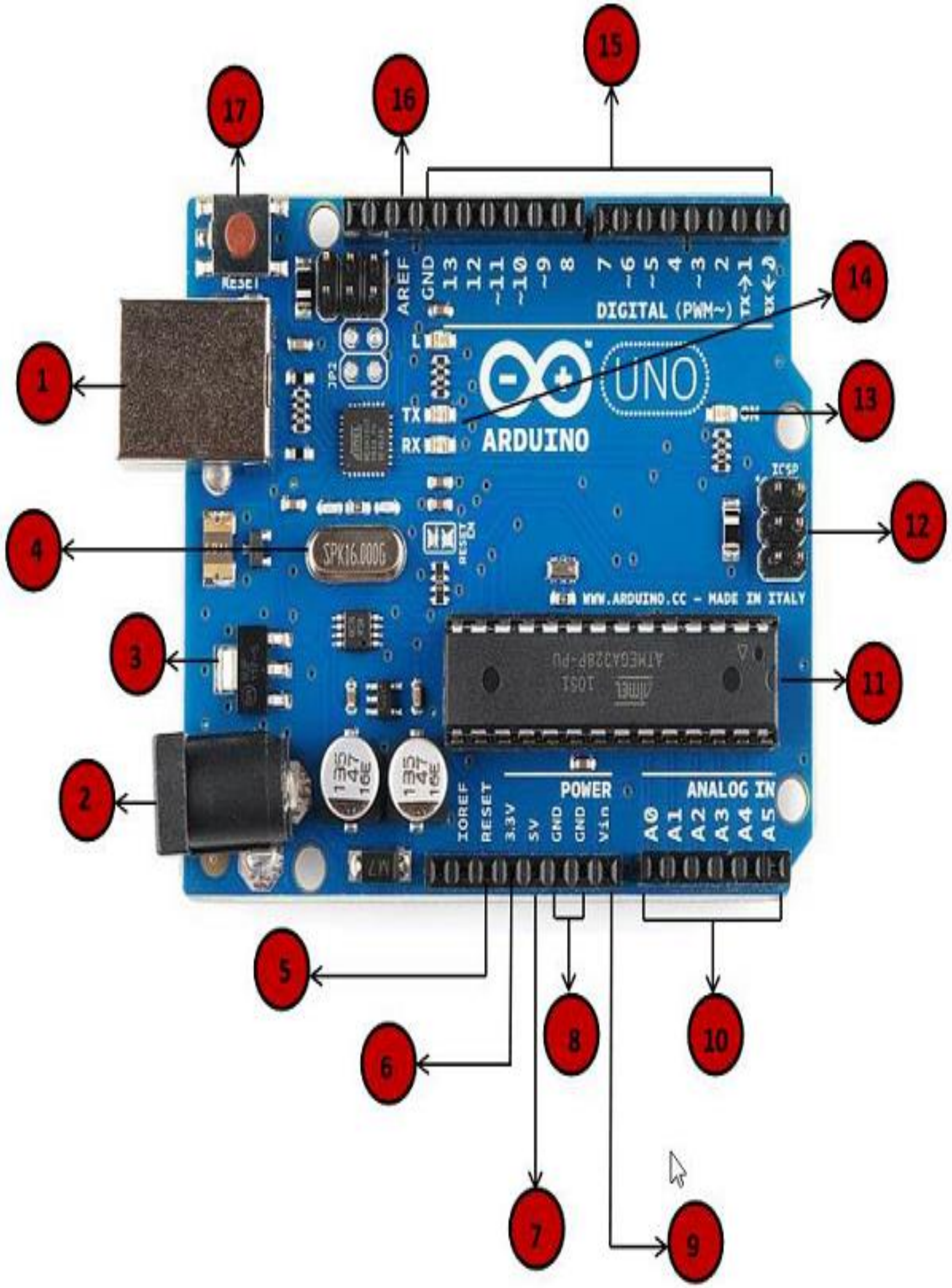


11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.
12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.

Arduino Integrated Development Environment (IDE)
Advanced technology for memory and logic.(ATMEL)

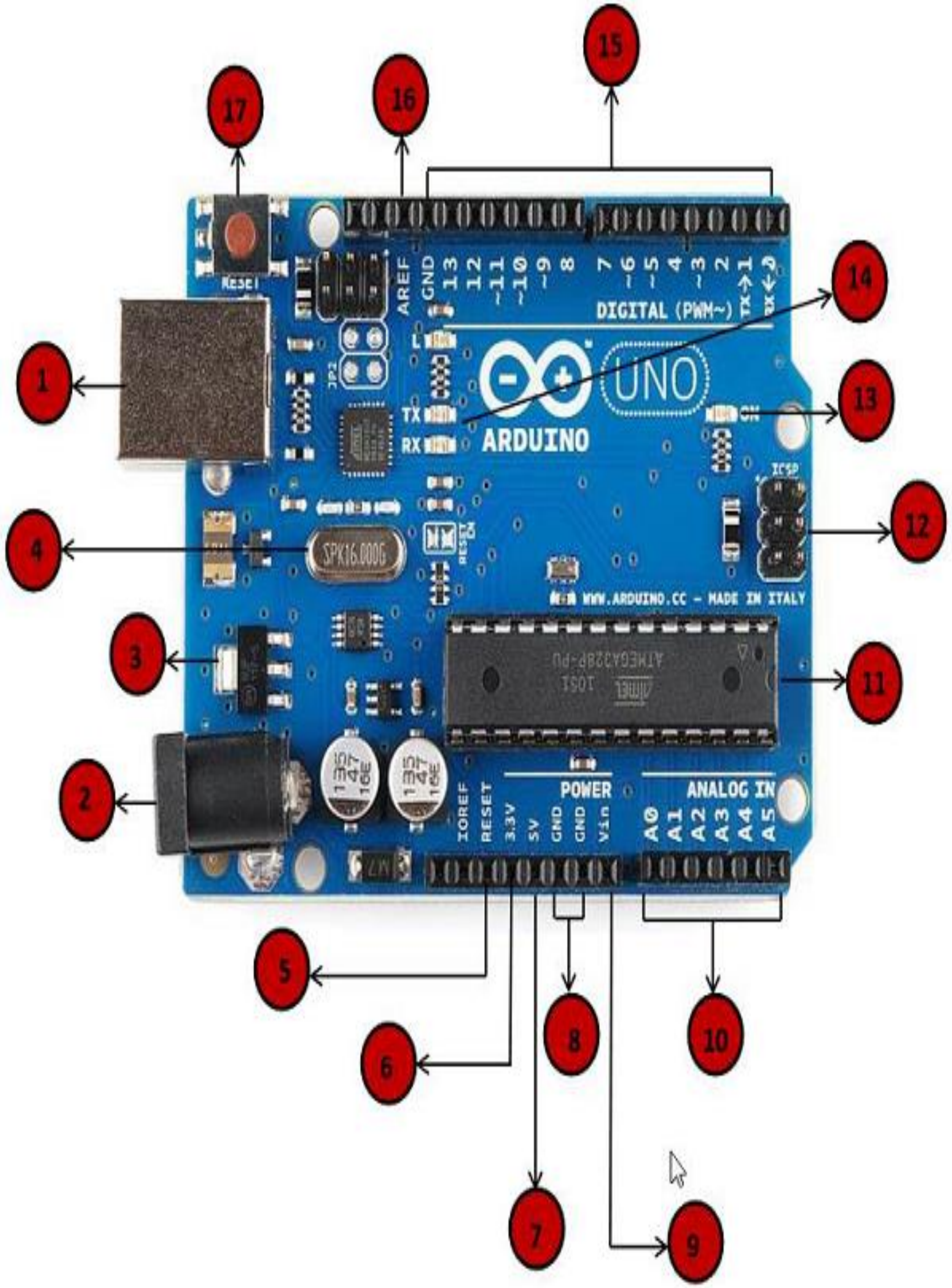
In-Circuit Serial Programming

Master Out Slave In (MOSI)
Master In Slave Out



Arduino Uno Board Description (Cont..)

14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF** AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.



INPUT v/s OUTPUT

Referenced from the perspective of the microcontroller (electrical board).

Inputs is a signal / information going into the Arduino board.

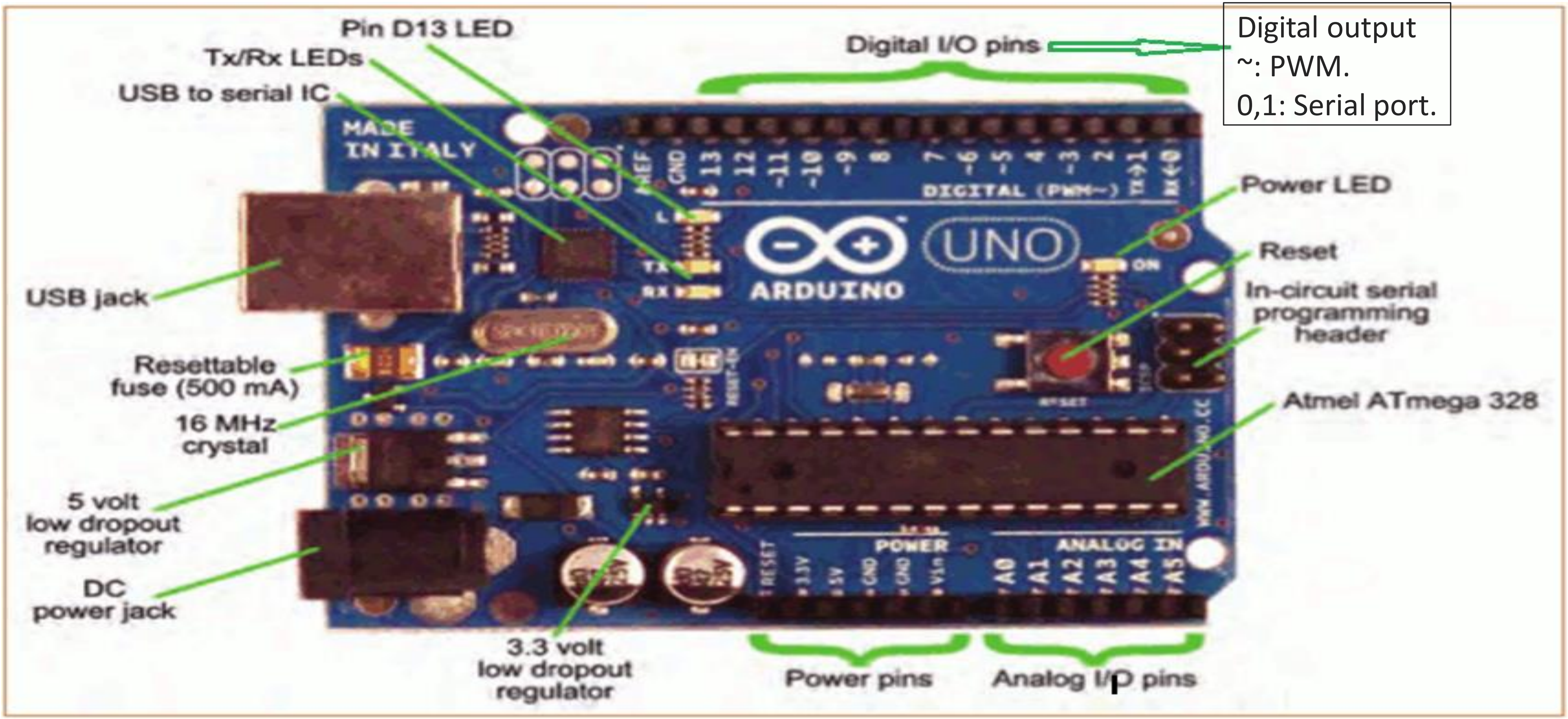
Examples: Buttons Switches, Light Sensors, Humidity Sensors, Temperature Sensors…

Output is any signal exiting the Arduino board.

Examples: LEDs, DC motor, relay, LED

https://getintopc.com/softwares/3d-cad/proteus-professional-2020-free-download/

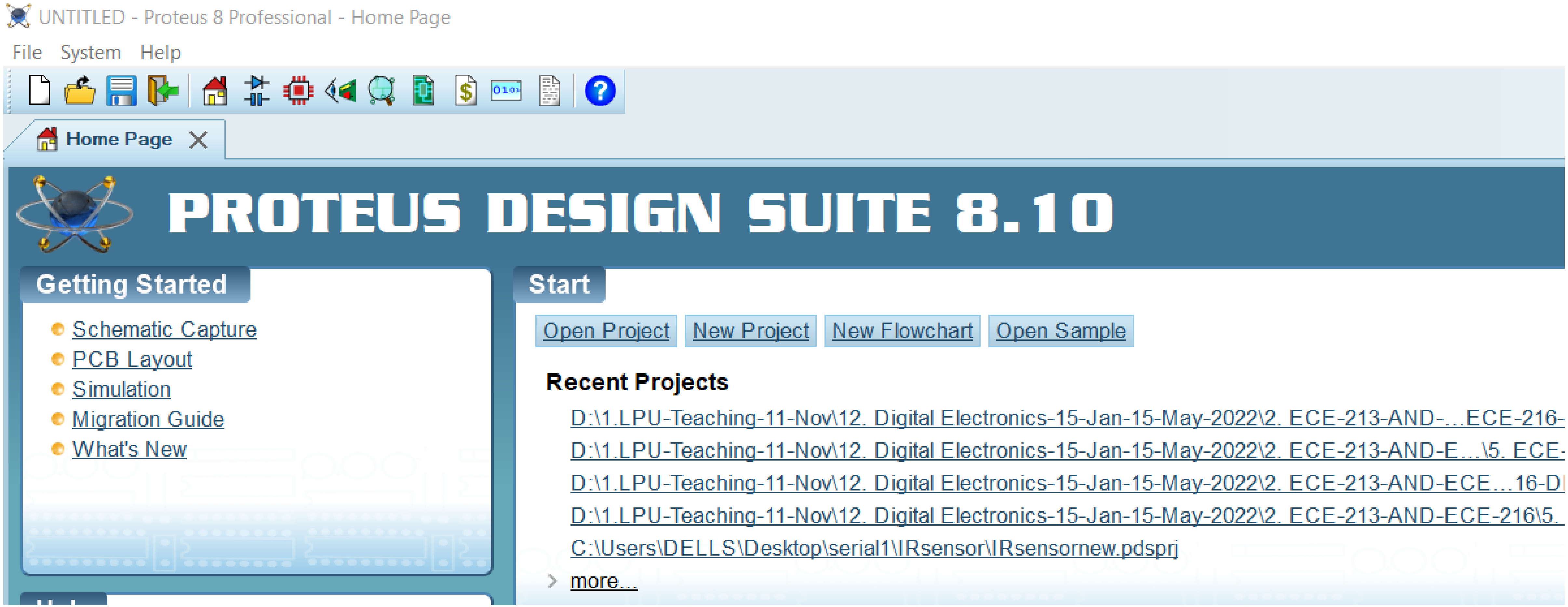
Arduino Uno Board



Digital output
~: PWM.
0,1: Serial port.

Connectors/Cables to work with Uno:

Proteus Software



UNTITLED - Proteus 8 Professional - Home Page

PROTEUS DESIGN SUITE 8.10

Getting Started

- Schematic Capture
- PCB Layout
- Simulation
- Migration Guide
- What's New

Start

Recent Projects

D:\1.LPU-Teaching-11-Nov\12. Digital Electronics-15-Jan-15-May-2022\2. ECE-213-AND-...ECE-216-

D:\1.LPU-Teaching-11-Nov\12. Digital Electronics-15-Jan-15-May-2022\2. ECE-213-AND-E...\5. ECE-

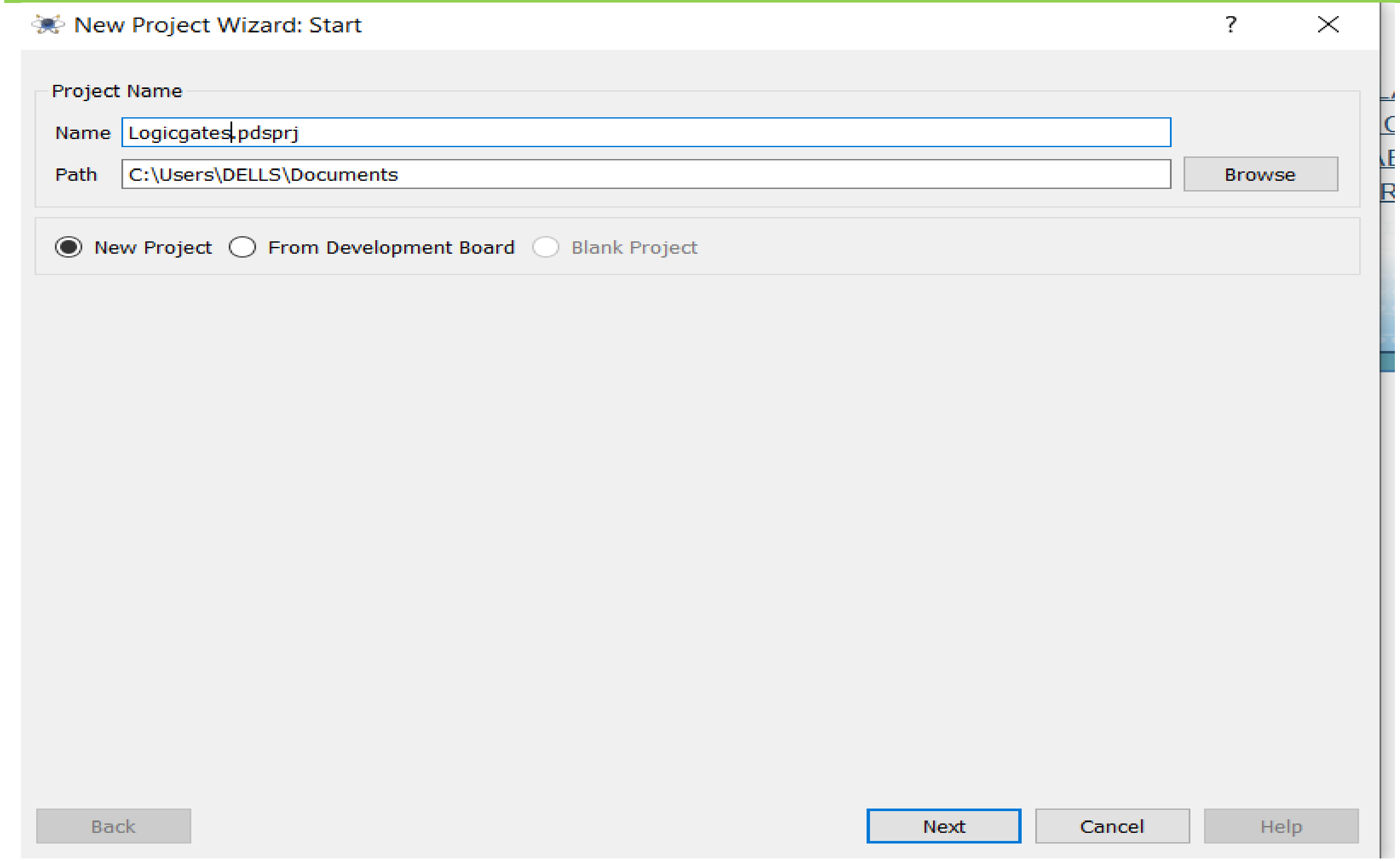
D:\1.LPU-Teaching-11-Nov\12. Digital Electronics-15-Jan-15-May-2022\2. ECE-213-AND-ECE...16-D

D:\1.LPU-Teaching-11-Nov\12. Digital Electronics-15-Jan-15-May-2022\2. ECE-213-AND-ECE-216\5.

C:\Users\DELLS\Desktop\serial1\IRsensor\IRsensornew.pdsprj

more...

Proteus Software



New Project Wizard: Start

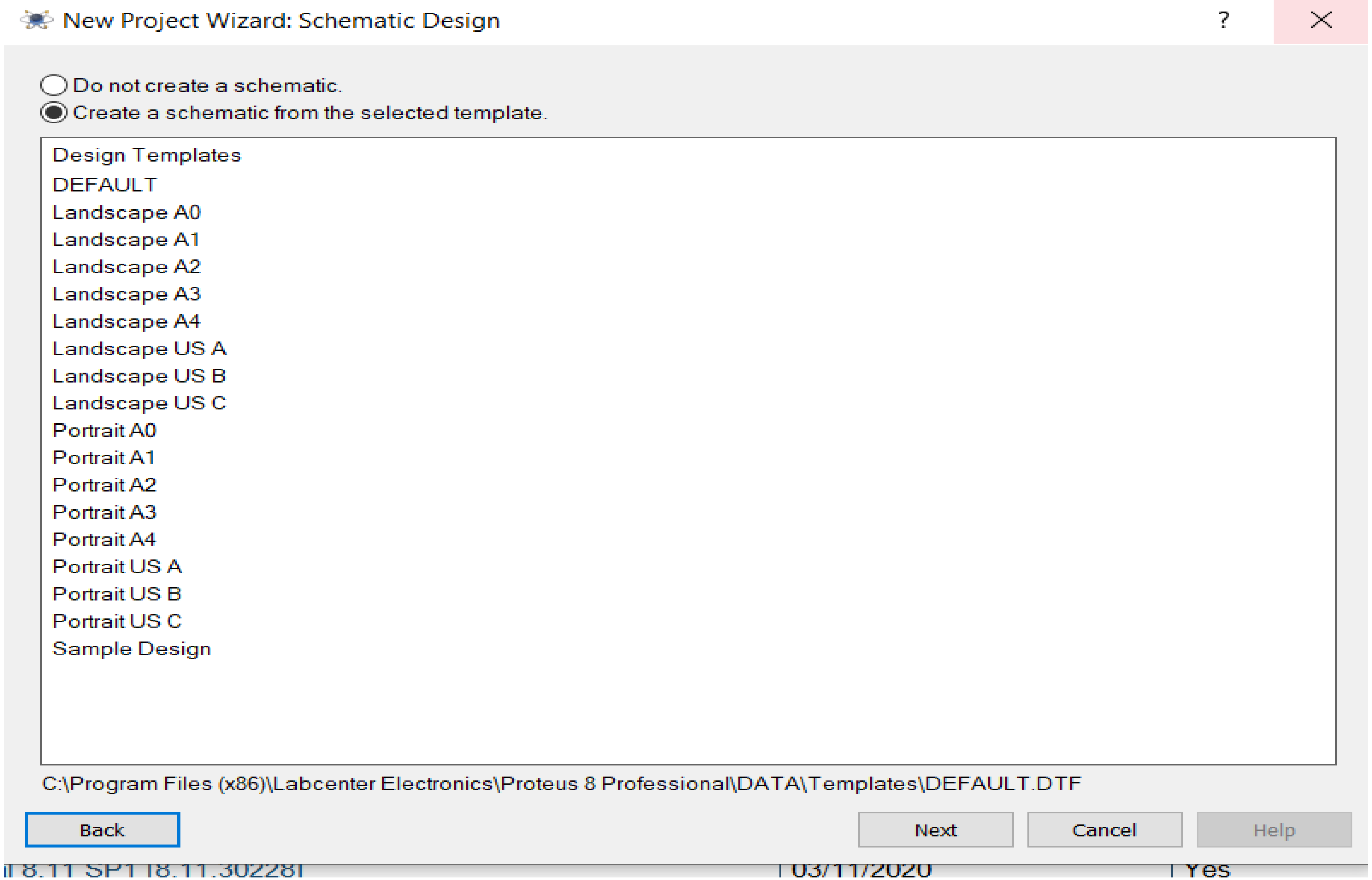
Project Name

Name: Logicgates.pdsprj

Path: C:\Users\DELLS\Documents

● New Project ○ From Development Board ○ Blank Project

Proteus Software



New Project Wizard: Schematic Design

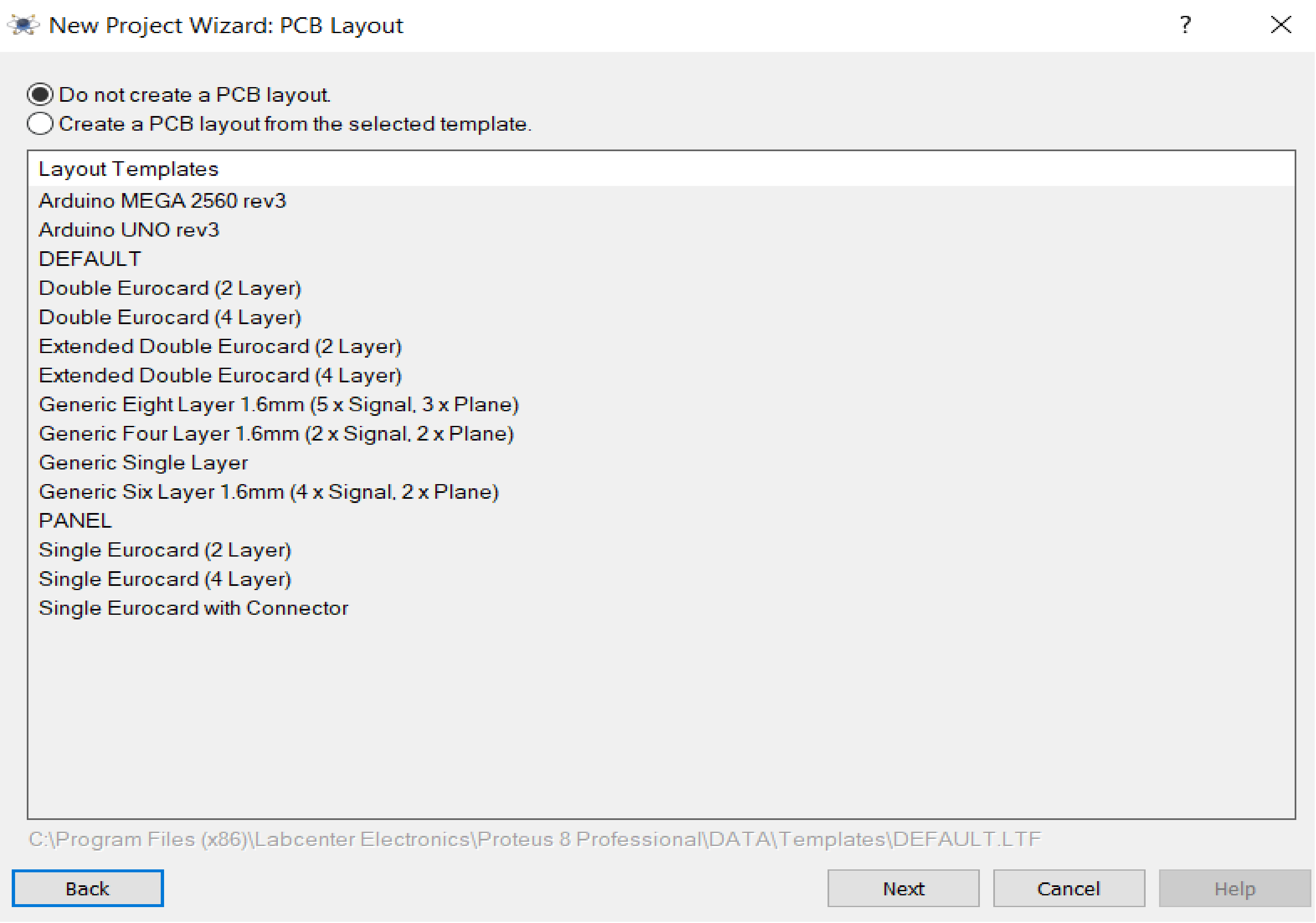
○ Do not create a schematic.
● Create a schematic from the selected template.

Design Templates
DEFAULT
Landscape A0
Landscape A1
Landscape A2
Landscape A3
Landscape A4
Landscape US A
Landscape US B
Landscape US C
Portrait A0
Portrait A1
Portrait A2
Portrait A3
Portrait A4
Portrait US A
Portrait US B
Portrait US C
Sample Design

C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Templates\DEFAULT.DTF

8.11 SP1 [8.11.30228] | 03/11/2020 | Yes

Proteus Software



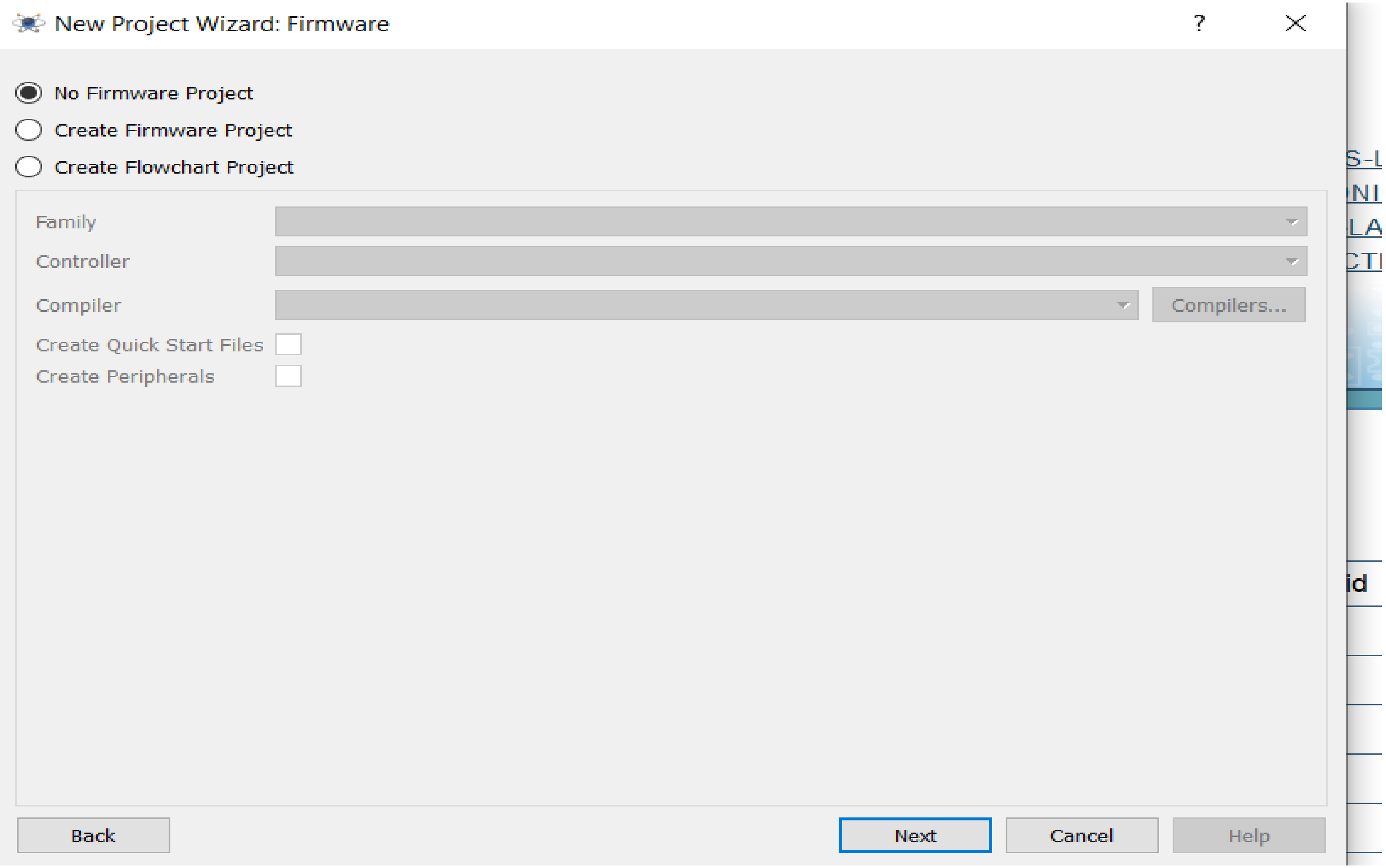
New Project Wizard: PCB Layout

● Do not create a PCB layout.
○ Create a PCB layout from the selected template.

Layout Templates
- Arduino MEGA 2560 rev3
- Arduino UNO rev3
- DEFAULT
- Double Eurocard (2 Layer)
- Double Eurocard (4 Layer)
- Extended Double Eurocard (2 Layer)
- Extended Double Eurocard (4 Layer)
- Generic Eight Layer 1.6mm (5 x Signal, 3 x Plane)
- Generic Four Layer 1.6mm (2 x Signal, 2 x Plane)
- Generic Single Layer
- Generic Six Layer 1.6mm (4 x Signal, 2 x Plane)
- PANEL
- Single Eurocard (2 Layer)
- Single Eurocard (4 Layer)
- Single Eurocard with Connector

C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Templates\DEFAULT.LTF

Proteus Software



New Project Wizard: Firmware

● No Firmware Project

○ Create Firmware Project

○ Create Flowchart Project

Family

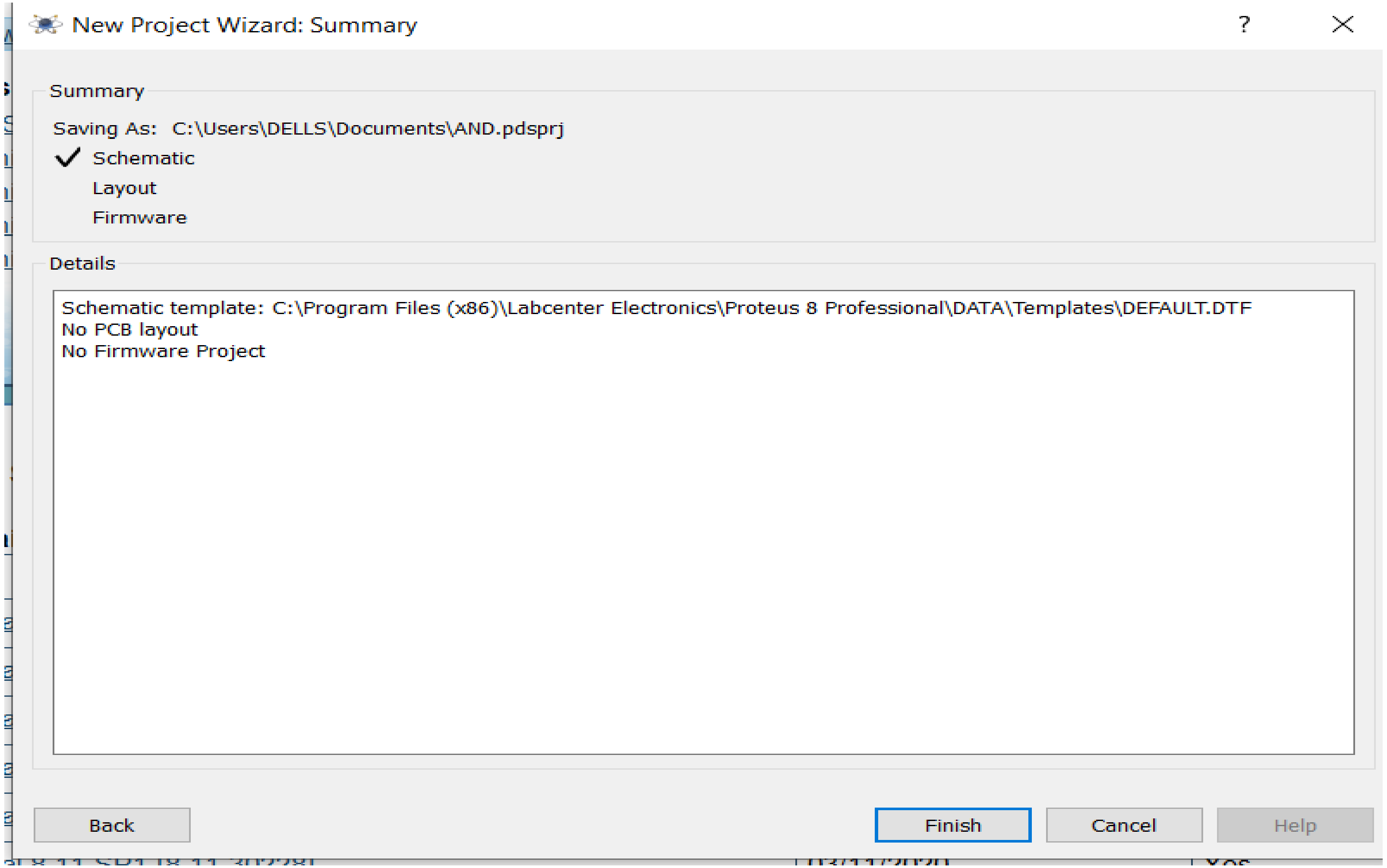
Controller

Compiler

Create Quick Start Files □

Create Peripherals □

Proteus Software



New Project Wizard: Summary

Summary

Saving As: C:\Users\DELLS\Documents\AND.pdsprj

✓ Schematic

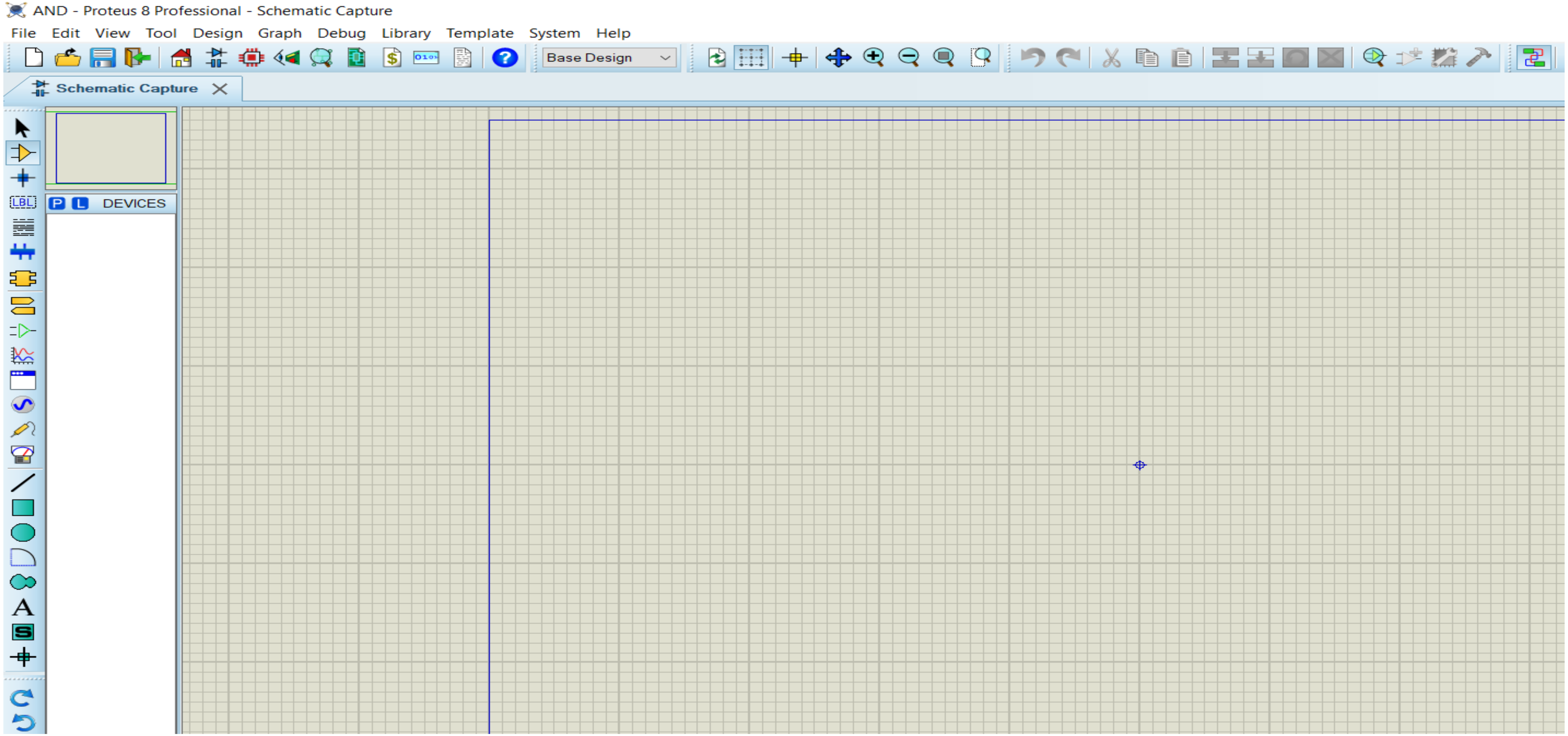
Layout

Firmware

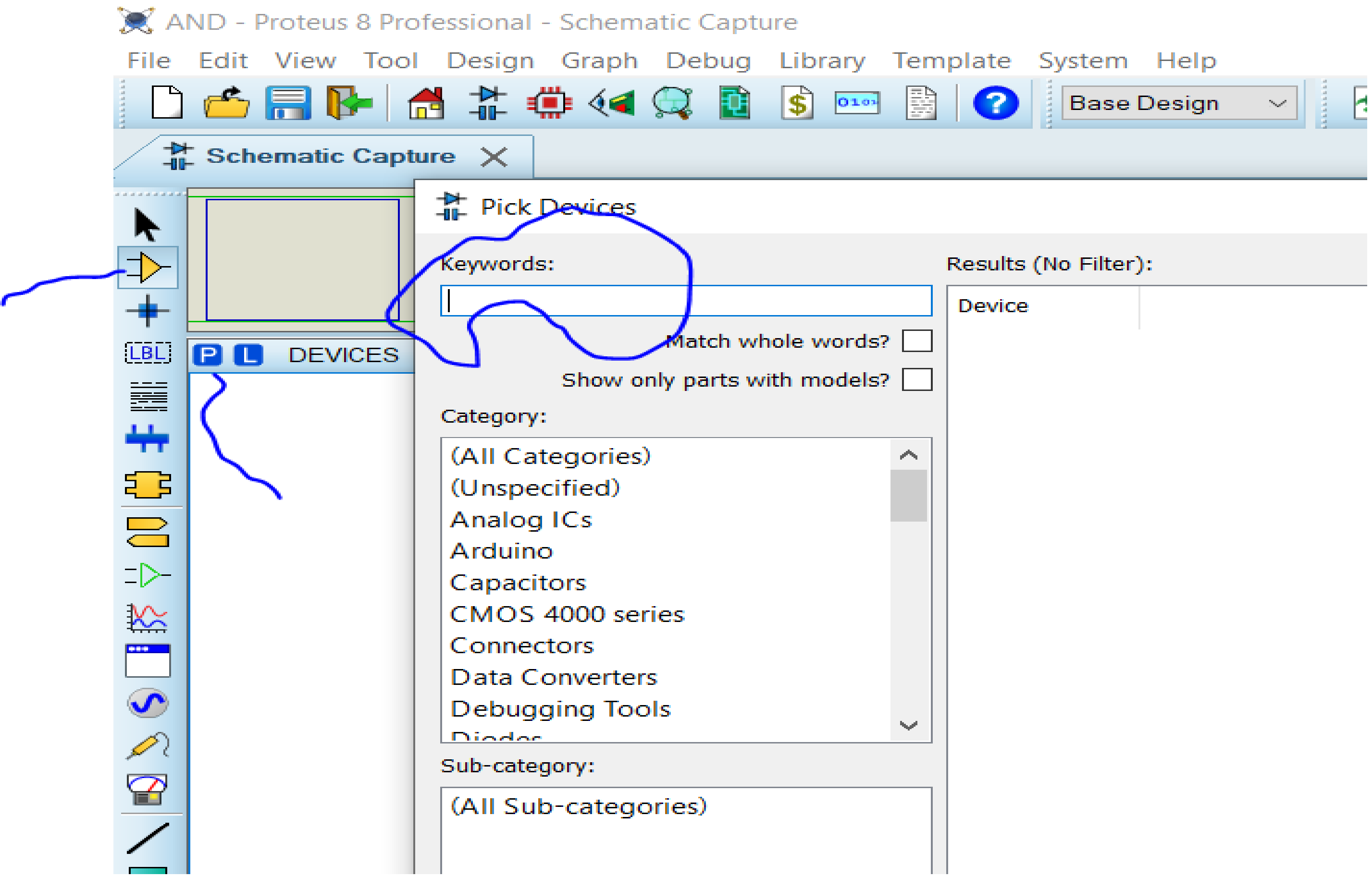
Details

```
Schematic template: C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Templates\DEFAULT.DTF
No PCB layout
No Firmware Project
```

Proteus Software



Proteus Software



Pick Devices

Keywords:

Match whole words? □

Show only parts with models? □

Category:

(All Categories)
(Unspecified)
Analog ICs
Arduino
Capacitors
CMOS 4000 series
Connectors
Data Converters
Debugging Tools
Diodes

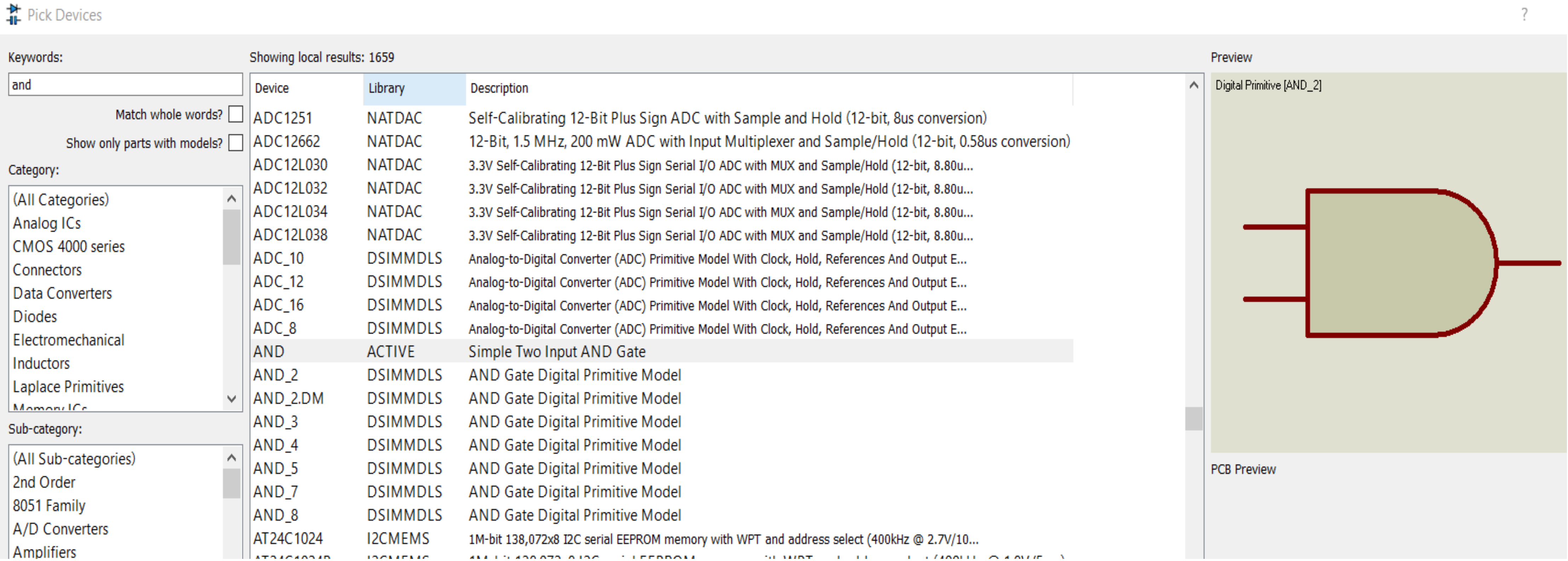
Sub-category:

(All Sub-categories)

Results (No Filter):

Device

Proteus Software



Keywords: and

Match whole words? □

Show only parts with models? □

Category:

(All Categories)
Analog ICs
CMOS 4000 series
Connectors
Data Converters
Diodes
Electromechanical
Inductors
Laplace Primitives
Memory ICs

Sub-category:

(All Sub-categories)
2nd Order
8051 Family
A/D Converters
Amplifiers

Showing local results: 1659

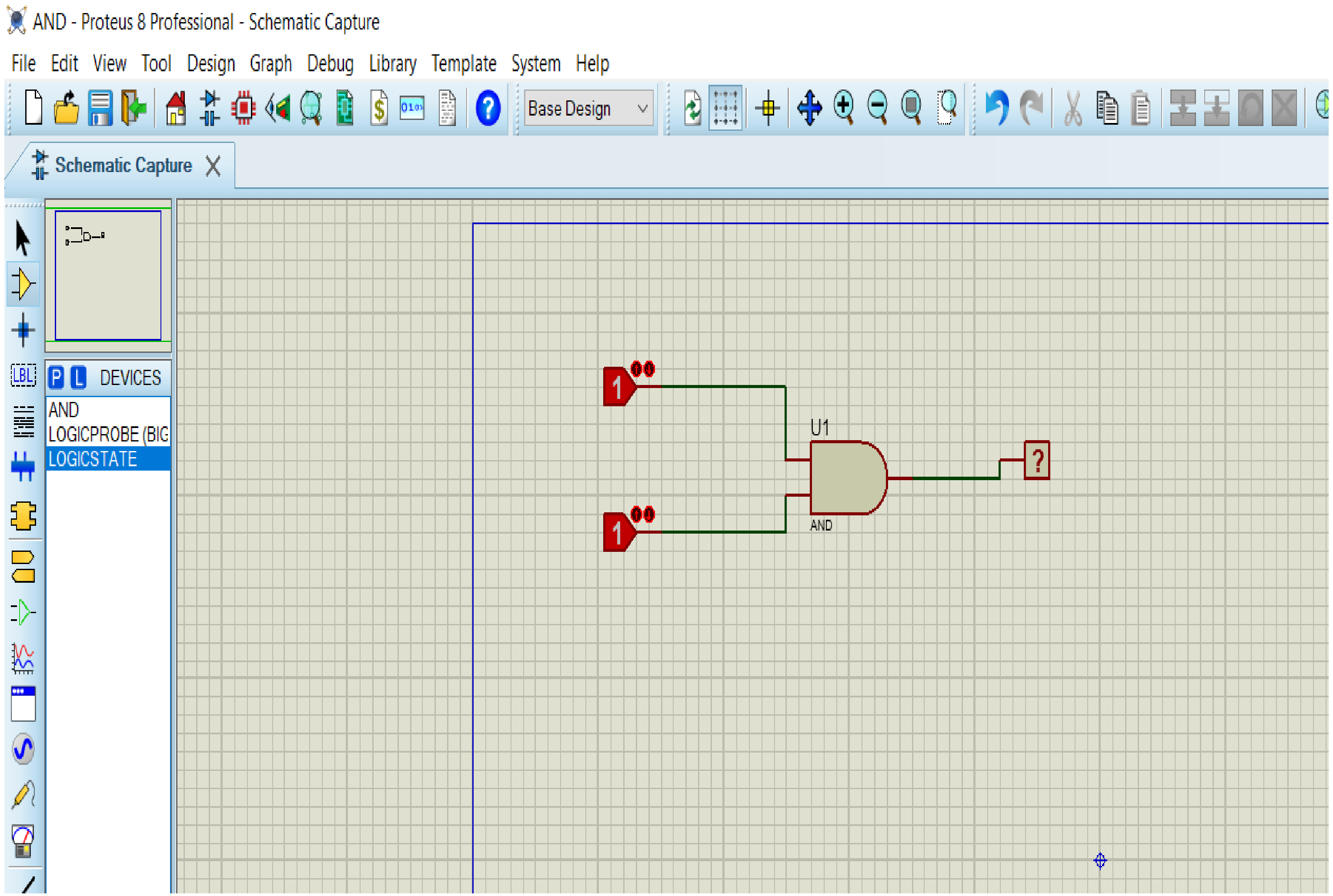
Device	Library	Description
ADC1251	NATDAC	Self-Calibrating 12-Bit Plus Sign ADC with Sample and Hold (12-bit, 8us conversion)
ADC12662	NATDAC	12-Bit, 1.5 MHz, 200 mW ADC with Input Multiplexer and Sample/Hold (12-bit, 0.58us conversion)
ADC12L030	NATDAC	3.3V Self-Calibrating 12-Bit Plus Sign Serial I/O ADC with MUX and Sample/Hold (12-bit, 8.80u...
ADC12L032	NATDAC	3.3V Self-Calibrating 12-Bit Plus Sign Serial I/O ADC with MUX and Sample/Hold (12-bit, 8.80u...
ADC12L034	NATDAC	3.3V Self-Calibrating 12-Bit Plus Sign Serial I/O ADC with MUX and Sample/Hold (12-bit, 8.80u...
ADC12L038	NATDAC	3.3V Self-Calibrating 12-Bit Plus Sign Serial I/O ADC with MUX and Sample/Hold (12-bit, 8.80u...
ADC_10	DSIMMDLS	Analog-to-Digital Converter (ADC) Primitive Model With Clock, Hold, References And Output E...
ADC_12	DSIMMDLS	Analog-to-Digital Converter (ADC) Primitive Model With Clock, Hold, References And Output E...
ADC_16	DSIMMDLS	Analog-to-Digital Converter (ADC) Primitive Model With Clock, Hold, References And Output E...
ADC_8	DSIMMDLS	Analog-to-Digital Converter (ADC) Primitive Model With Clock, Hold, References And Output E...
AND	ACTIVE	Simple Two Input AND Gate
AND_2	DSIMMDLS	AND Gate Digital Primitive Model
AND_2.DM	DSIMMDLS	AND Gate Digital Primitive Model
AND_3	DSIMMDLS	AND Gate Digital Primitive Model
AND_4	DSIMMDLS	AND Gate Digital Primitive Model
AND_5	DSIMMDLS	AND Gate Digital Primitive Model
AND_7	DSIMMDLS	AND Gate Digital Primitive Model
AND_8	DSIMMDLS	AND Gate Digital Primitive Model
AT24C1024	I2CMEMS	1M-bit 138,072x8 I2C serial EEPROM memory with WPT and address select (400kHz @ 2.7V/10...

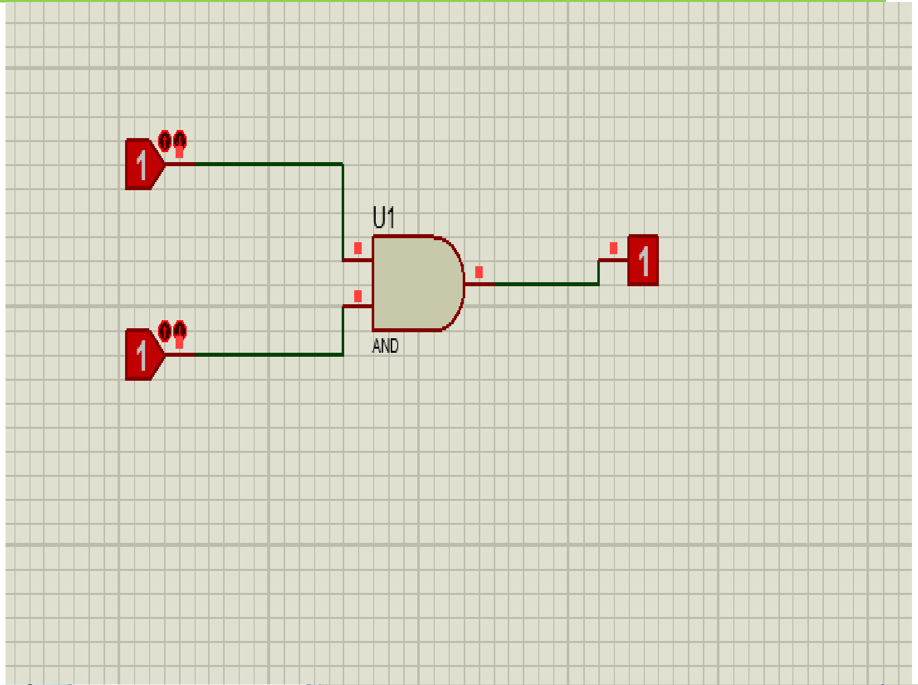
Preview

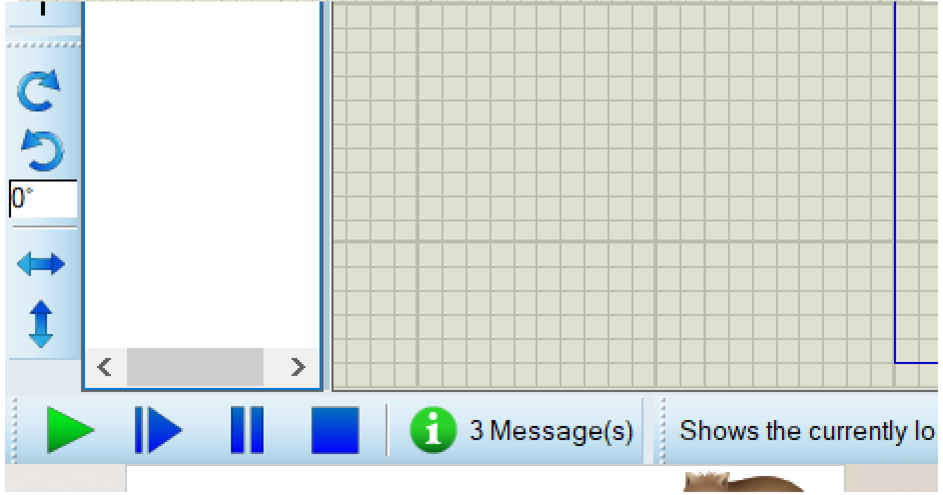
Digital Primitive [AND_2]

PCB Preview

Proteus Software

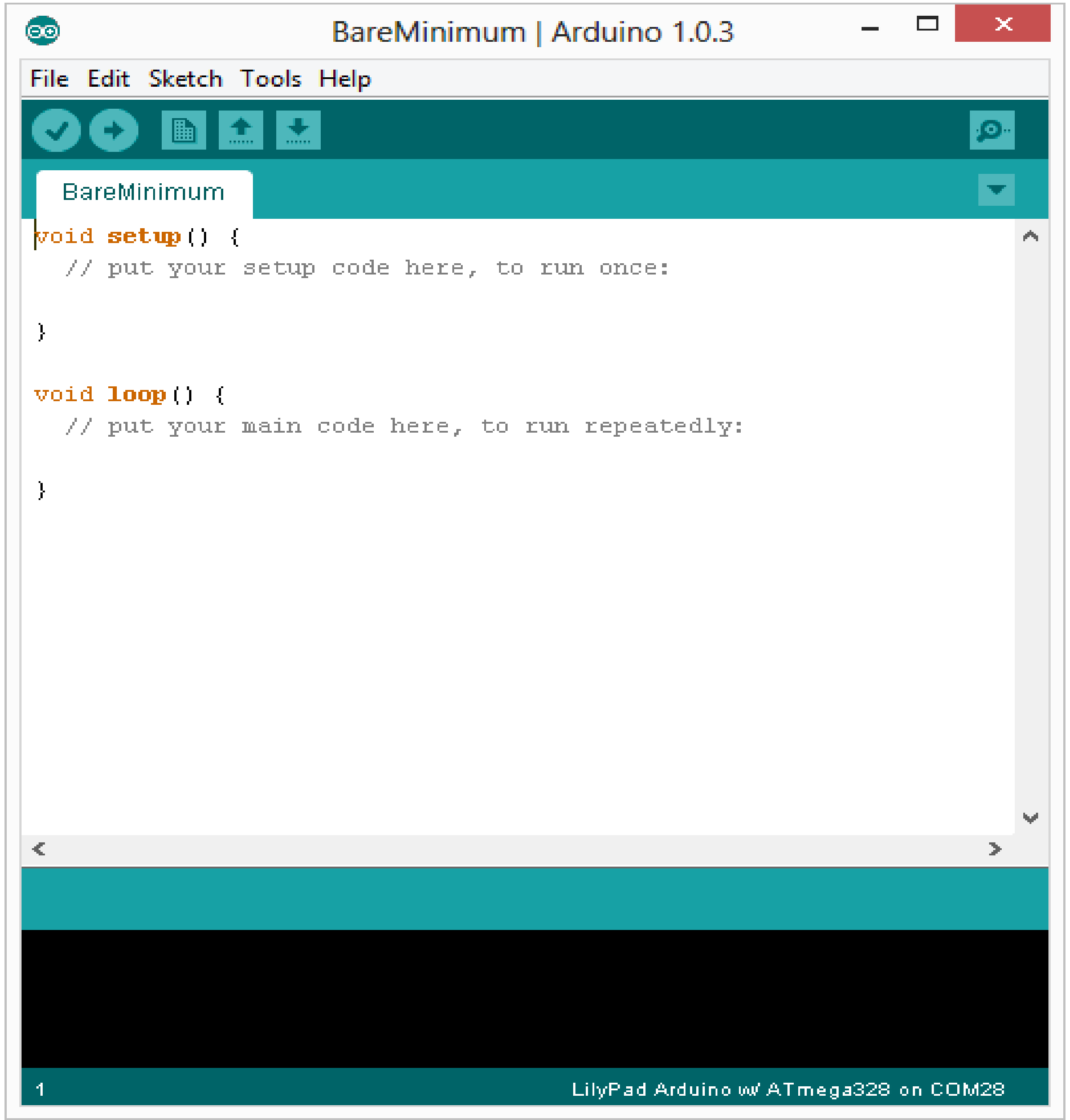

AND - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
P L DEVICES
AND
LOGICPROBE (BIG
LOGICSTATE
1
1
U1
AND
?





C:\Users\DELLS\Desktop\ProteusSoftware\1. And Gates

Arduino IDE (Cont..)

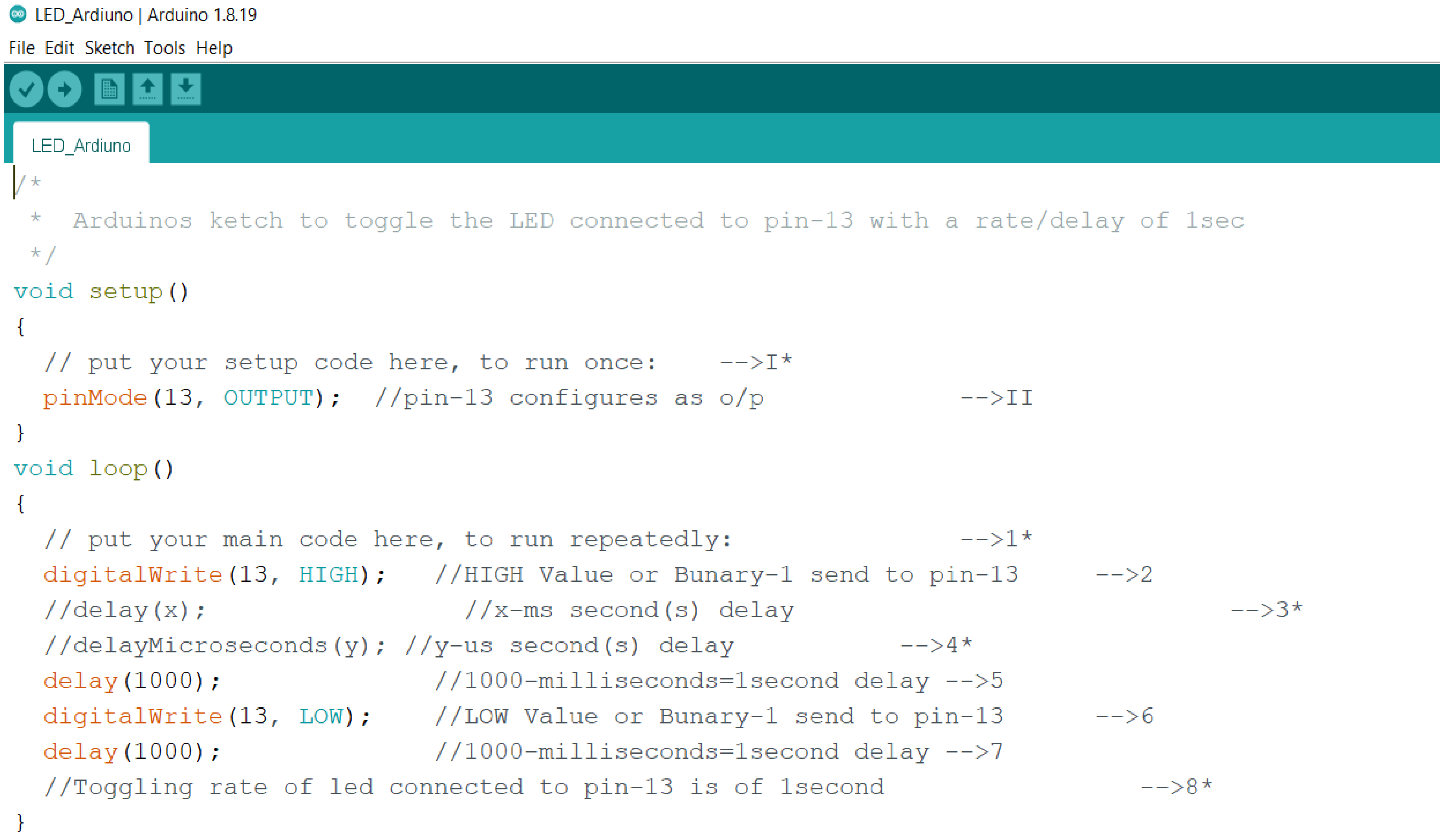

BareMinimum | Arduino 1.0.3
File Edit Sketch Tools Help
BareMinimum
1
LilyPad Arduino w/ ATmega328 on COM28

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

}

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

}
```

Two required functions / methods / routines:

```
void setup()
{
        // runs once
}

void loop()
{
        // repeats
}
```

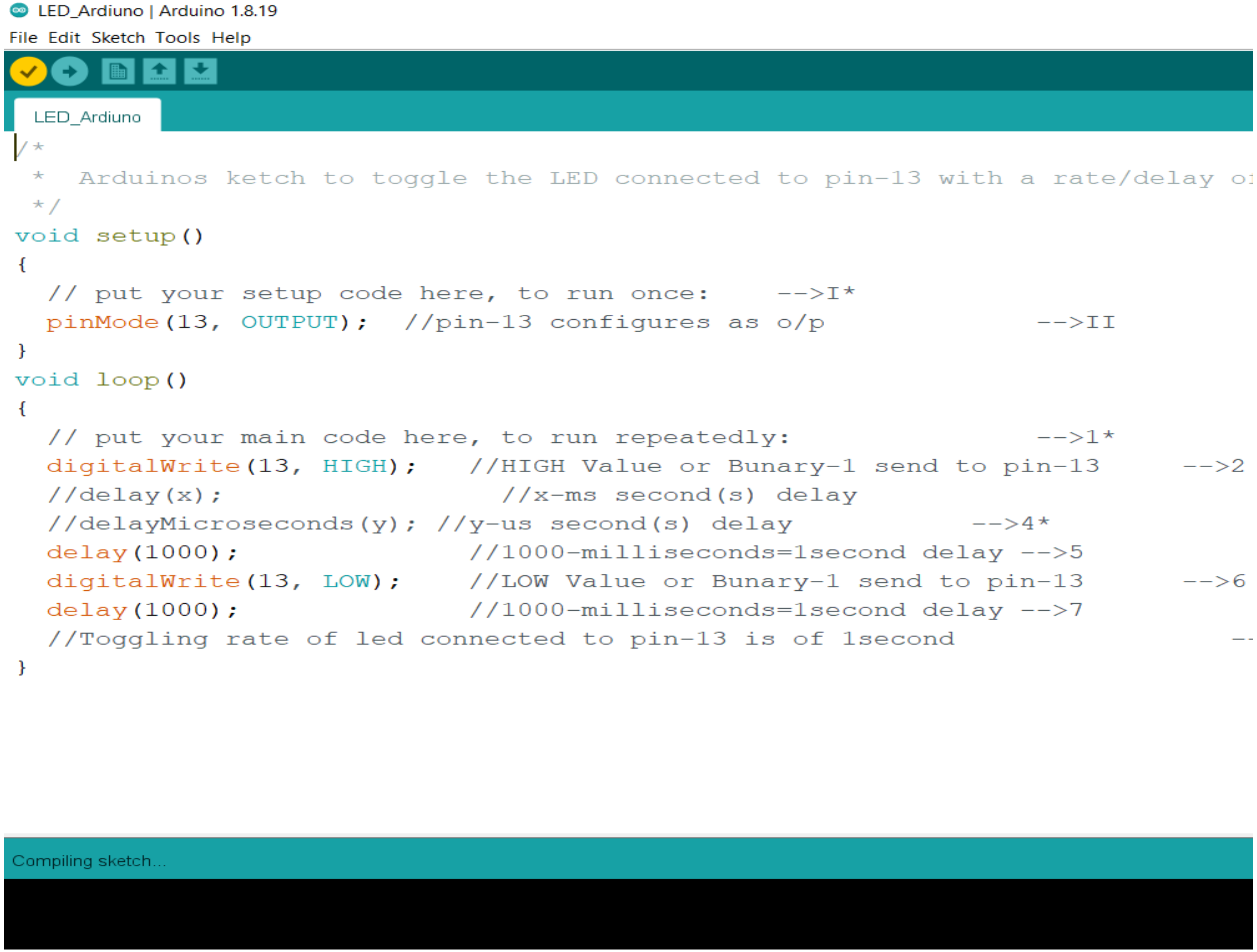
Arduino-IDE- LED Program



```
/*
 *  Arduinos ketch to toggle the LED connected to pin-13 with a rate/delay of 1sec
 */
void setup()
{
  // put your setup code here, to run once:    -->I*
  pinMode(13, OUTPUT);  //pin-13 configures as o/p             -->II
}
void loop()
{
  // put your main code here, to run repeatedly:               -->1*
  digitalWrite(13, HIGH);   //HIGH Value or Bunary-1 send to pin-13     -->2
  //delay(x);                 //x-ms second(s) delay                             -->3*
  //delayMicroseconds(y); //y-us second(s) delay           -->4*
  delay(1000);              //1000-milliseconds=1second delay -->5
  digitalWrite(13, LOW);    //LOW Value or Bunary-1 send to pin-13      -->6
  delay(1000);              //1000-milliseconds=1second delay -->7
  //Toggling rate of led connected to pin-13 is of 1second                 -->8*
}
```

Sketch→Export Complied Binary---for Hex file of Proteus Arduino

Arduino-IDE- LED Program

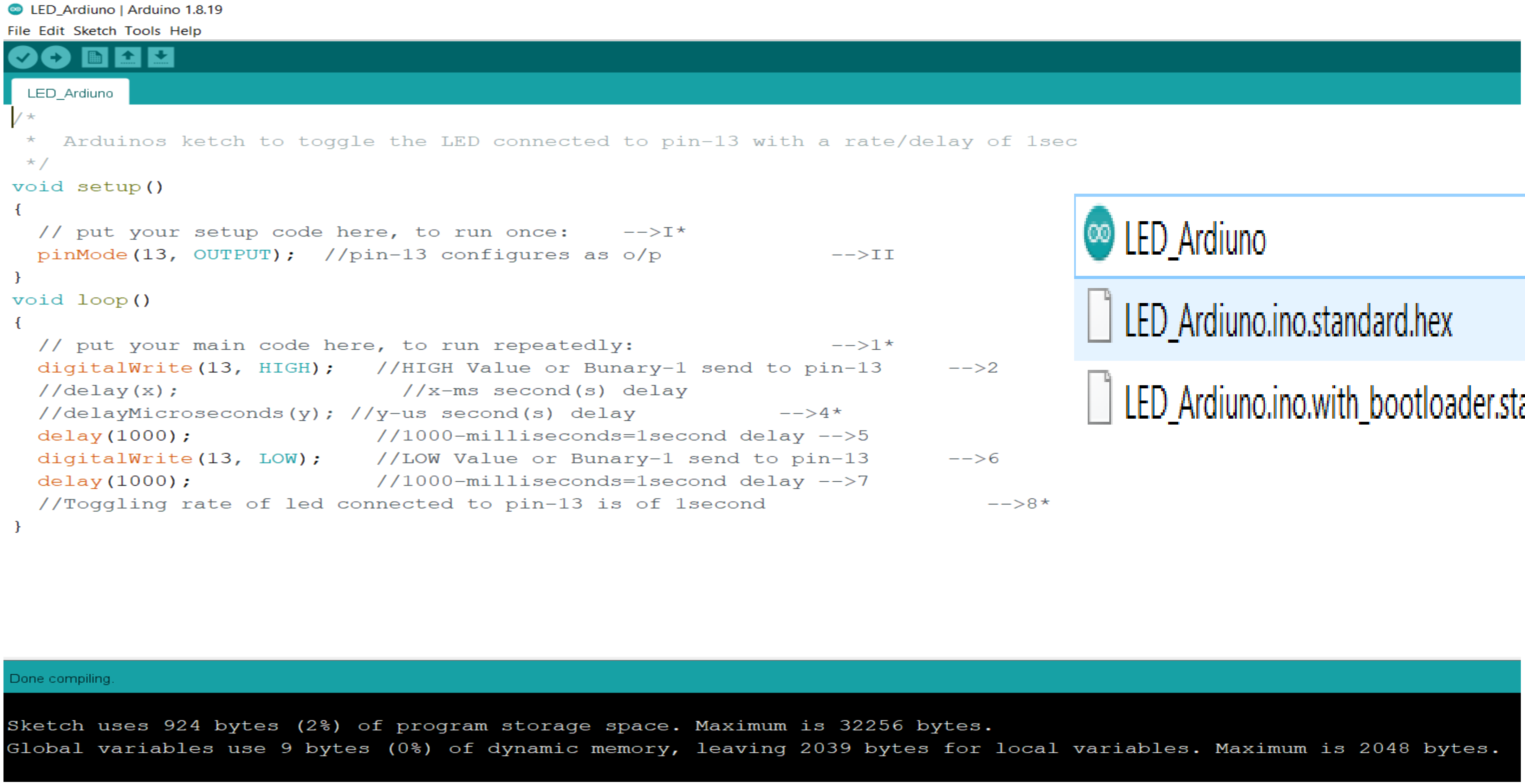


```
/*
 *  Arduinos ketch to toggle the LED connected to pin-13 with a rate/delay o
 */
void setup()
{
  // put your setup code here, to run once:    -->I*
  pinMode(13, OUTPUT);  //pin-13 configures as o/p             -->II
}
void loop()
{
  // put your main code here, to run repeatedly:              -->1*
  digitalWrite(13, HIGH);   //HIGH Value or Bunary-1 send to pin-13     -->2
  //delay(x);                 //x-ms second(s) delay
  //delayMicroseconds(y); //y-us second(s) delay             -->4*
  delay(1000);              //1000-milliseconds=1second delay -->5
  digitalWrite(13, LOW);    //LOW Value or Bunary-1 send to pin-13      -->6
  delay(1000);              //1000-milliseconds=1second delay -->7
  //Toggling rate of led connected to pin-13 is of 1second                --
}
```

```
Compiling sketch...
```

Sketch→Export Complied Binary---for Hex file of Proteus Arduino

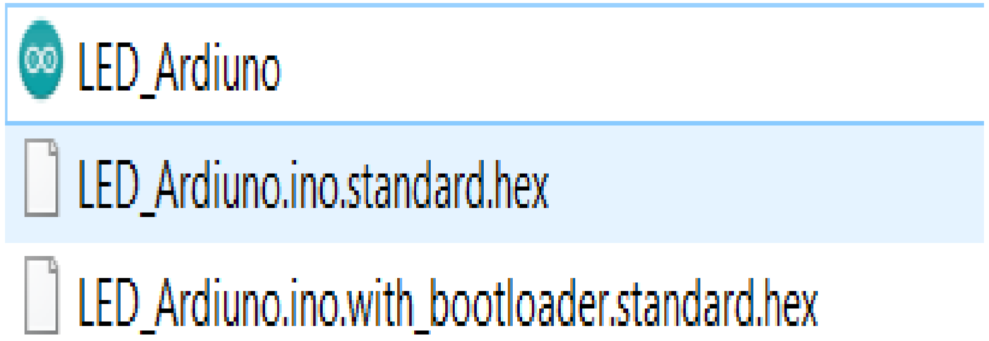
Arduino-IDE- LED Program



```
/*
 *  Arduinos ketch to toggle the LED connected to pin-13 with a rate/delay of 1sec
 */
void setup()
{
  // put your setup code here, to run once:    -->I*
  pinMode(13, OUTPUT);  //pin-13 configures as o/p              -->II
}
void loop()
{
  // put your main code here, to run repeatedly:              -->1*
  digitalWrite(13, HIGH);   //HIGH Value or Bunary-1 send to pin-13     -->2
  //delay(x);                //x-ms second(s) delay
  //delayMicroseconds(y); //y-us second(s) delay            -->4*
  delay(1000);              //1000-milliseconds=1second delay -->5
  digitalWrite(13, LOW);    //LOW Value or Bunary-1 send to pin-13      -->6
  delay(1000);              //1000-milliseconds=1second delay -->7
  //Toggling rate of led connected to pin-13 is of 1second                -->8*
}
```

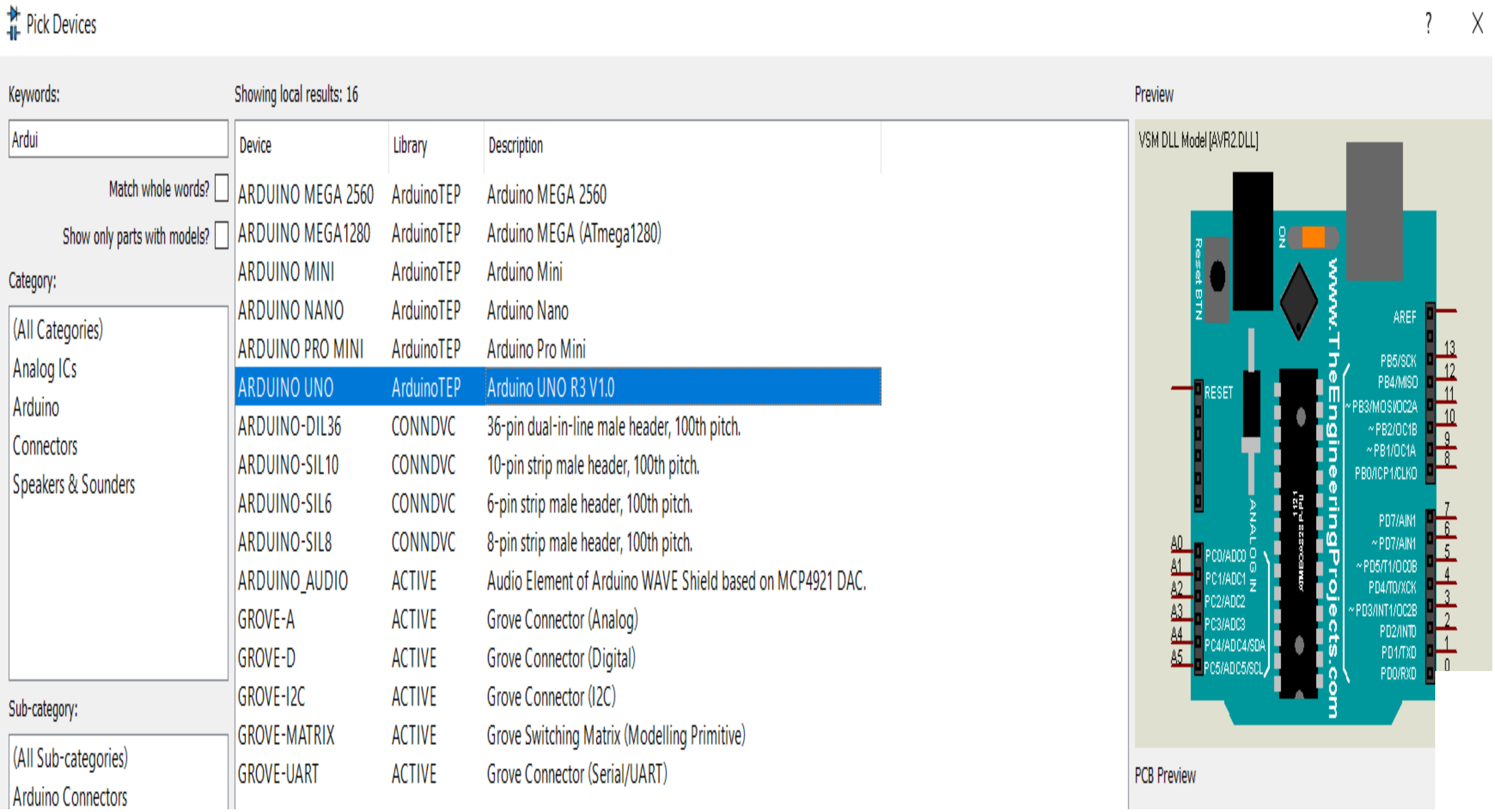
Done compiling.

```
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```



LED_Ardiuno
LED_Ardiuno.ino.standard.hex
LED_Ardiuno.ino.with_bootloader.standard.hex

Proteus-Arduino-IDE- LED Program



Pick Devices

Keywords: Ardui

Match whole words? □

Show only parts with models? □

Category:

(All Categories)
Analog ICs
Arduino
Connectors
Speakers & Sounders

Sub-category:

(All Sub-categories)
Arduino Connectors

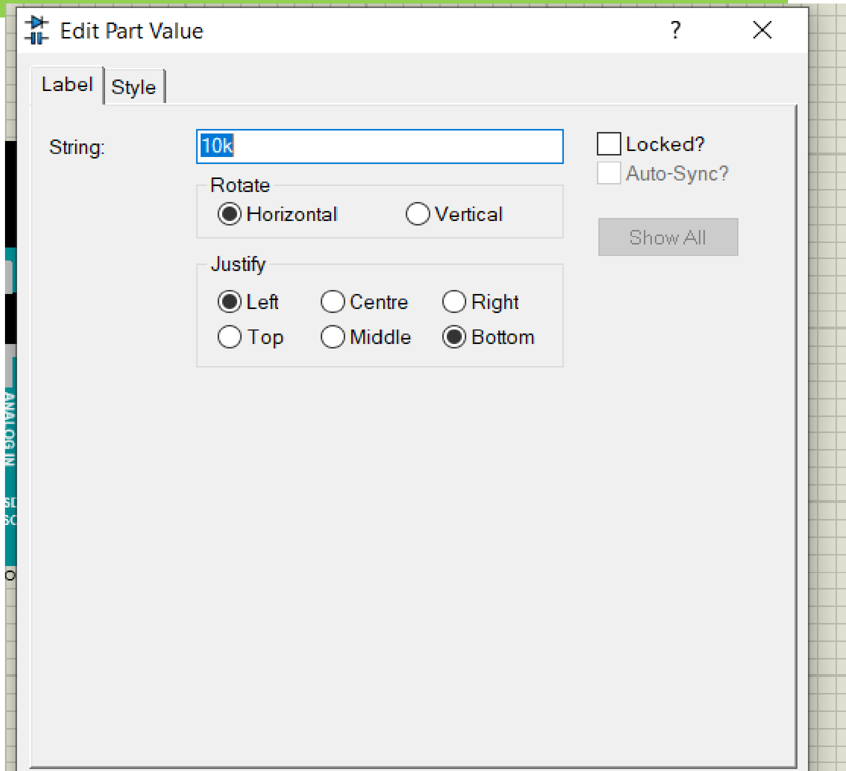
Showing local results: 16

Device	Library	Description
ARDUINO MEGA 2560	ArduinoTEP	Arduino MEGA 2560
ARDUINO MEGA1280	ArduinoTEP	Arduino MEGA (ATmega1280)
ARDUINO MINI	ArduinoTEP	Arduino Mini
ARDUINO NANO	ArduinoTEP	Arduino Nano
ARDUINO PRO MINI	ArduinoTEP	Arduino Pro Mini
ARDUINO UNO	ArduinoTEP	Arduino UNO R3 V1.0
ARDUINO-DIL36	CONNDVC	36-pin dual-in-line male header, 100th pitch.
ARDUINO-SIL10	CONNDVC	10-pin strip male header, 100th pitch.
ARDUINO-SIL6	CONNDVC	6-pin strip male header, 100th pitch.
ARDUINO-SIL8	CONNDVC	8-pin strip male header, 100th pitch.
ARDUINO_AUDIO	ACTIVE	Audio Element of Arduino WAVE Shield based on MCP4921 DAC.
GROVE-A	ACTIVE	Grove Connector (Analog)
GROVE-D	ACTIVE	Grove Connector (Digital)
GROVE-I2C	ACTIVE	Grove Connector (I2C)
GROVE-MATRIX	ACTIVE	Grove Switching Matrix (Modelling Primitive)
GROVE-UART	ACTIVE	Grove Connector (Serial/UART)

Preview

VSM DLL Model [AVR2.DLL]

PCB Preview



Edit Part Value

Label | Style

String: 10k

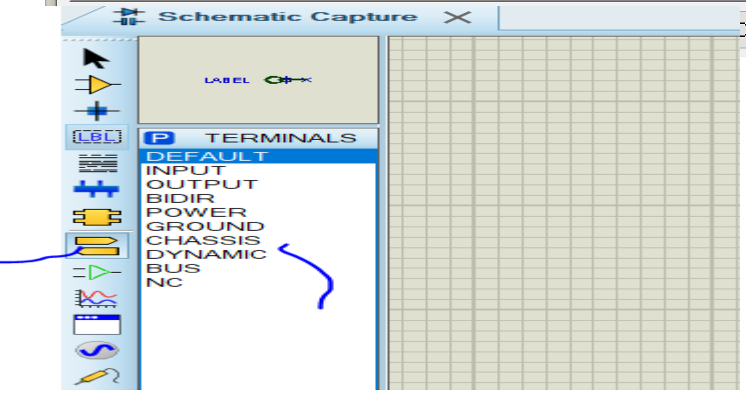
Rotate: ● Horizontal ○ Vertical

Justify: ● Left ○ Centre ○ Right ○ Top ○ Middle ● Bottom

□ Locked?
□ Auto-Sync?

Show All

Cancel

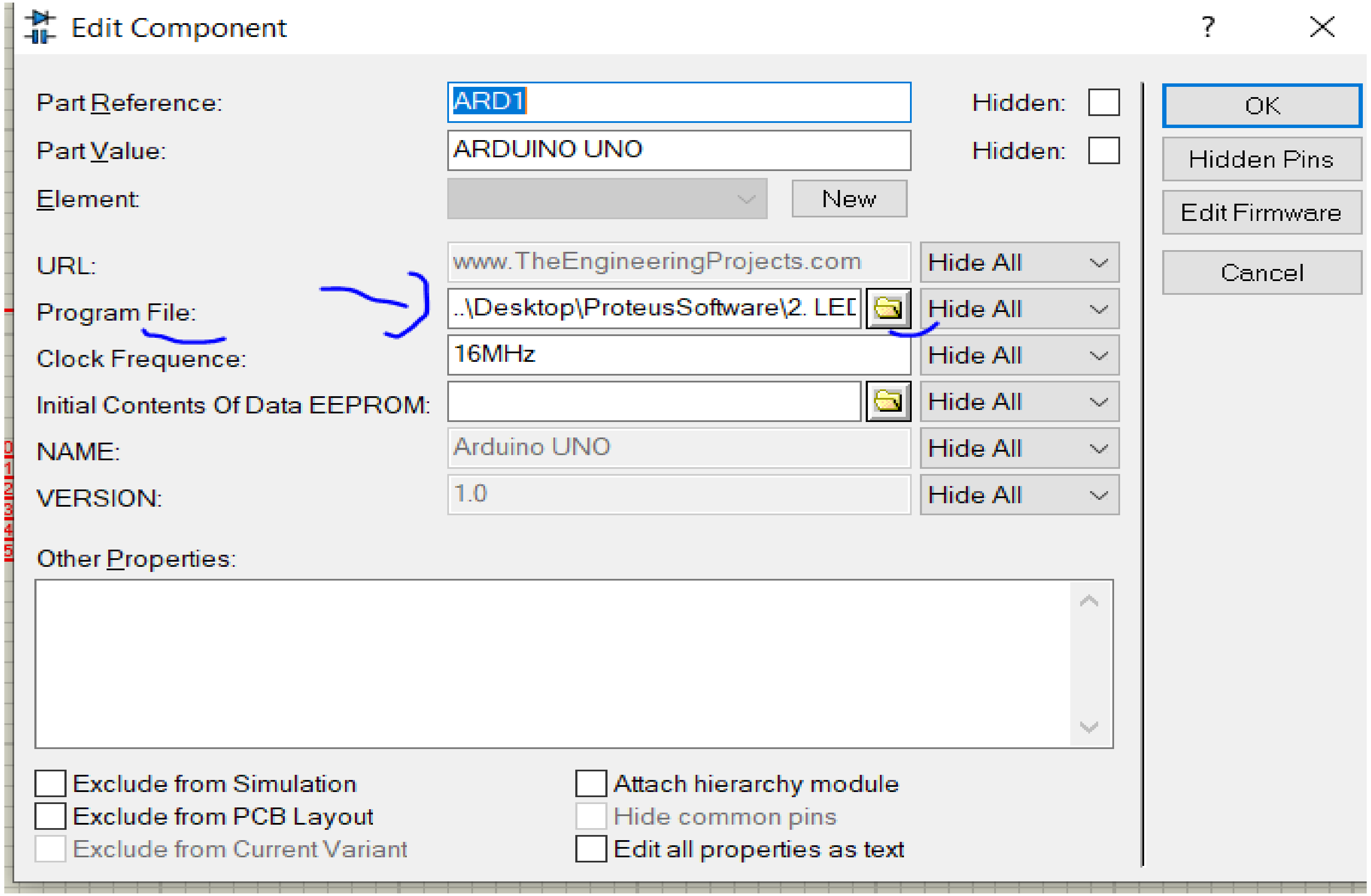


Schematic Capture

TERMINALS

DEFAULT
INPUT
OUTPUT
BIDIR
POWER
GROUND
CHASSIS
DYNAMIC
BUS
NC

Arduino-IDE- LED Program



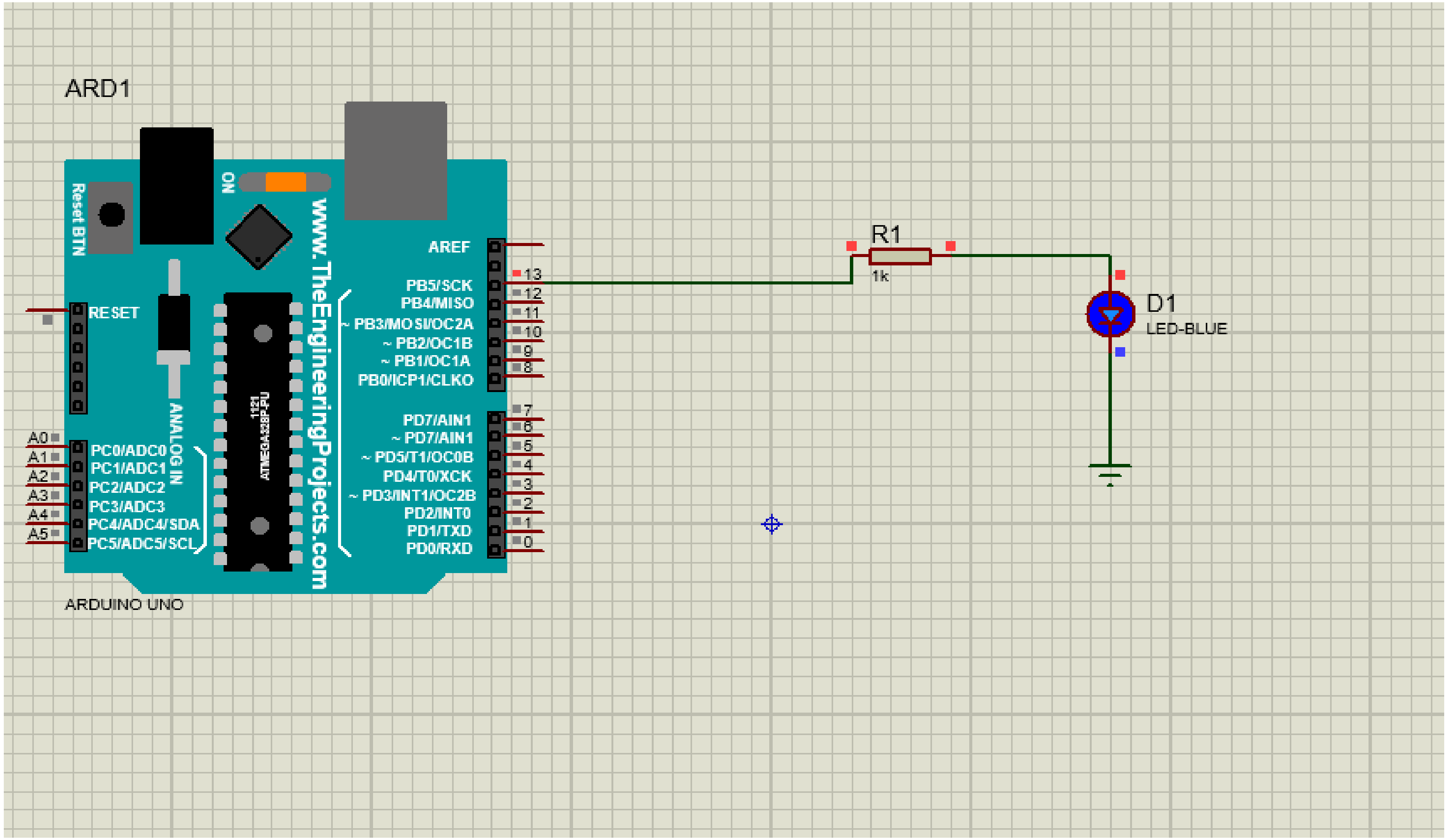
Edit Component

Part Reference:	ARD1	Hidden: □	
Part Value:	ARDUINO UNO	Hidden: □	
Element:		New	
URL:	www.TheEngineeringProjects.com	Hide All	
Program File:	..\Desktop\ProteusSoftware\2. LEI	Hide All	
Clock Frequence:	16MHz	Hide All	
Initial Contents Of Data EEPROM:		Hide All	
NAME:	Arduino UNO	Hide All	
VERSION:	1.0	Hide All	

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Current Variant
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text

Arduino-IDE- LED Program


ARD1
Reset BTN
ON
www.TheEngineeringProjects.com
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
13
12
11
10
9
8
RESET
ANALOG IN
ATMEGA328P-PU
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
7
6
5
4
3
2
1
0
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
ARDUINO UNO
R1
1k
D1
LED-BLUE