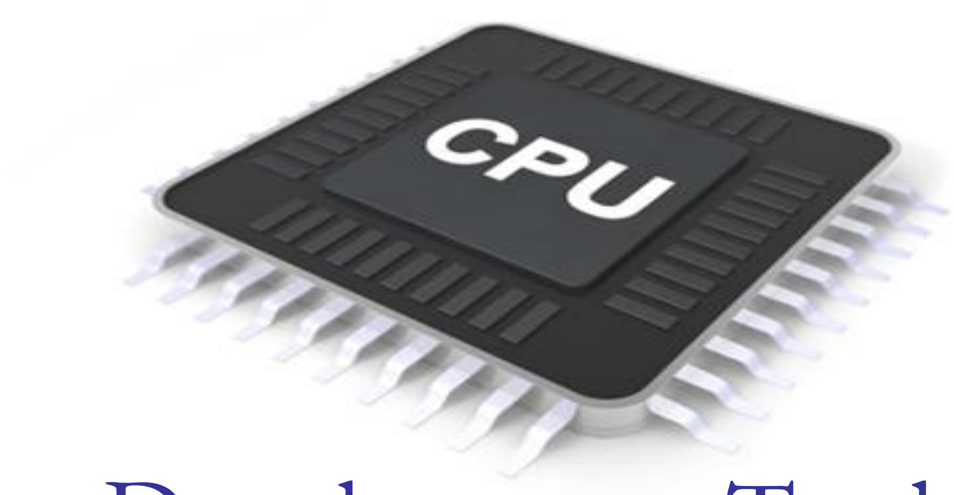


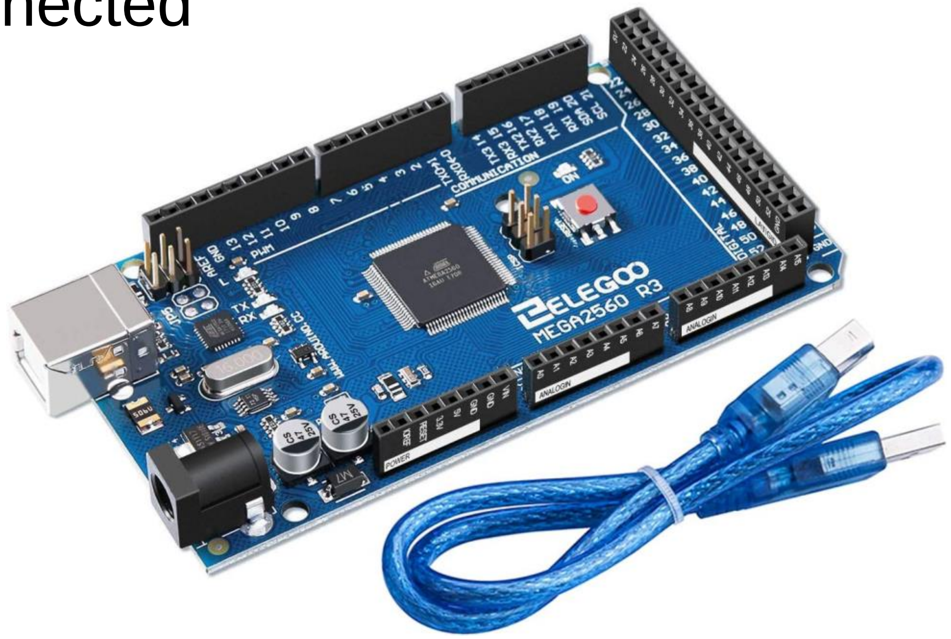
Development Tools

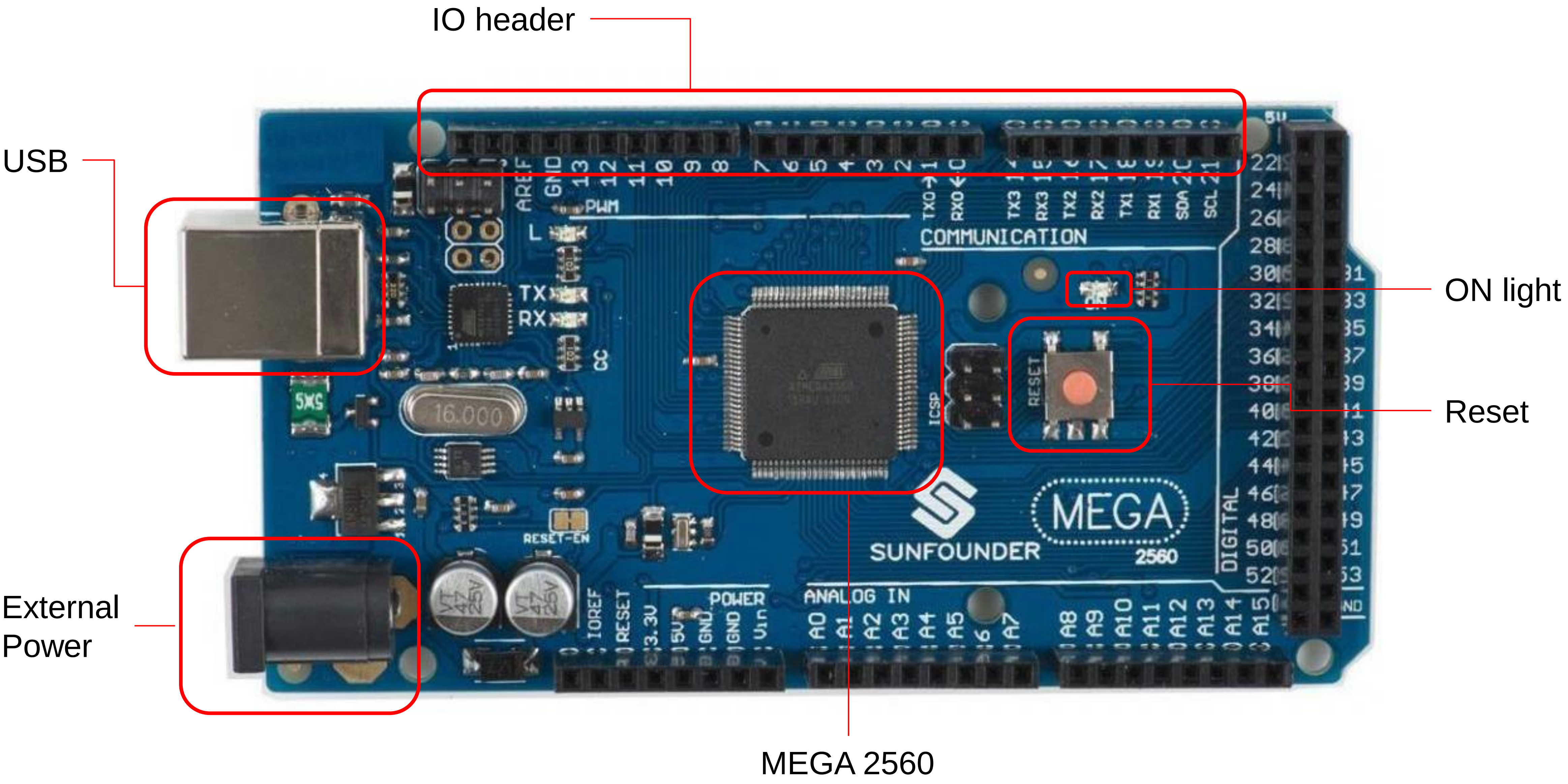
ENEE 3587

Microp Interfacing

Hardware: Mega 2560 R3 Board

- ELEGOO Mega 2560 R3
  - Microcontroller Mega 2560 R3
- USB cable is used to power the board when connected to a computer
- External power connector used to run board in autonomous mode
- IO headers use to provide signal input/output for microcontroller





IO header

USB

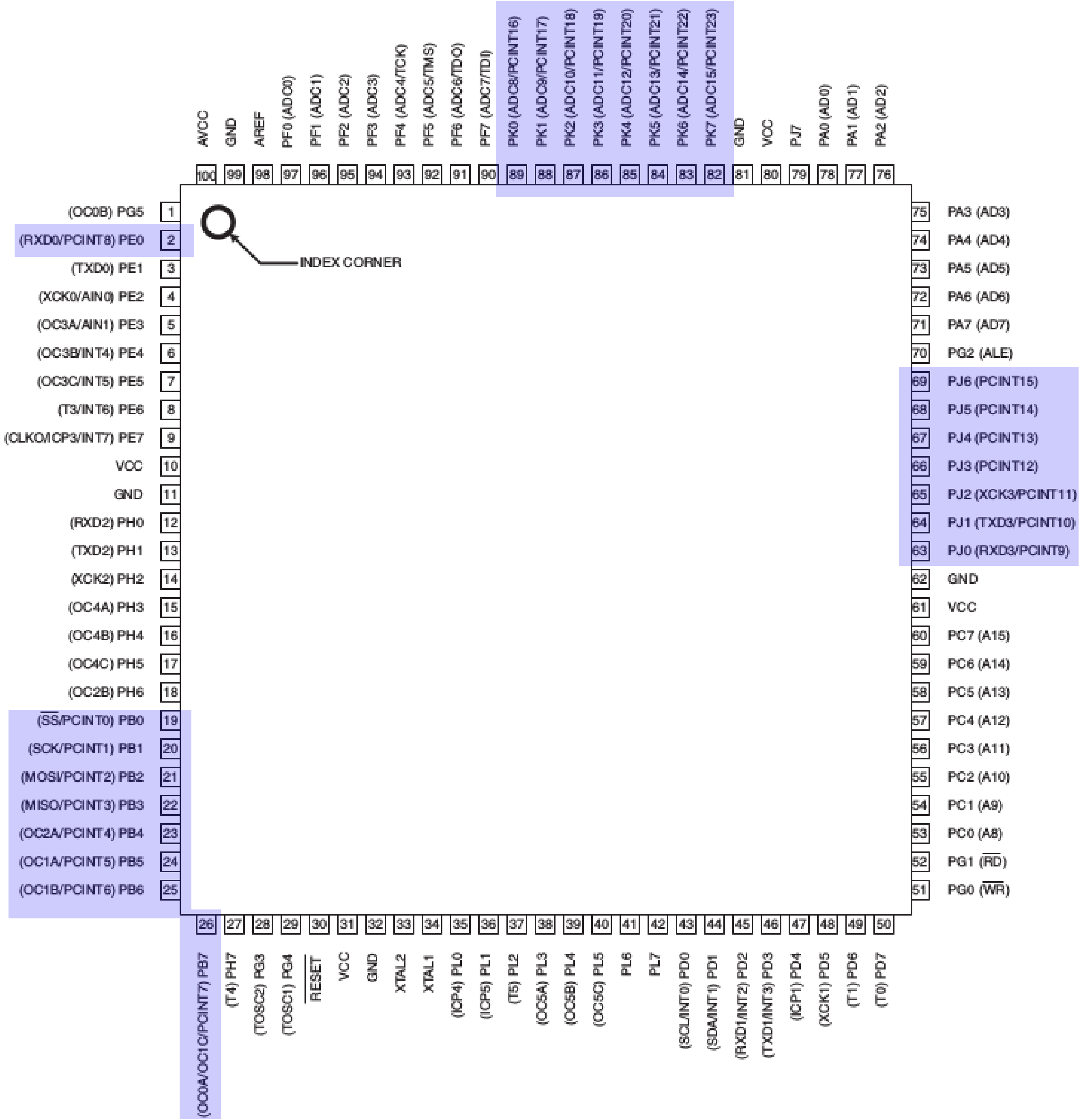
ON light

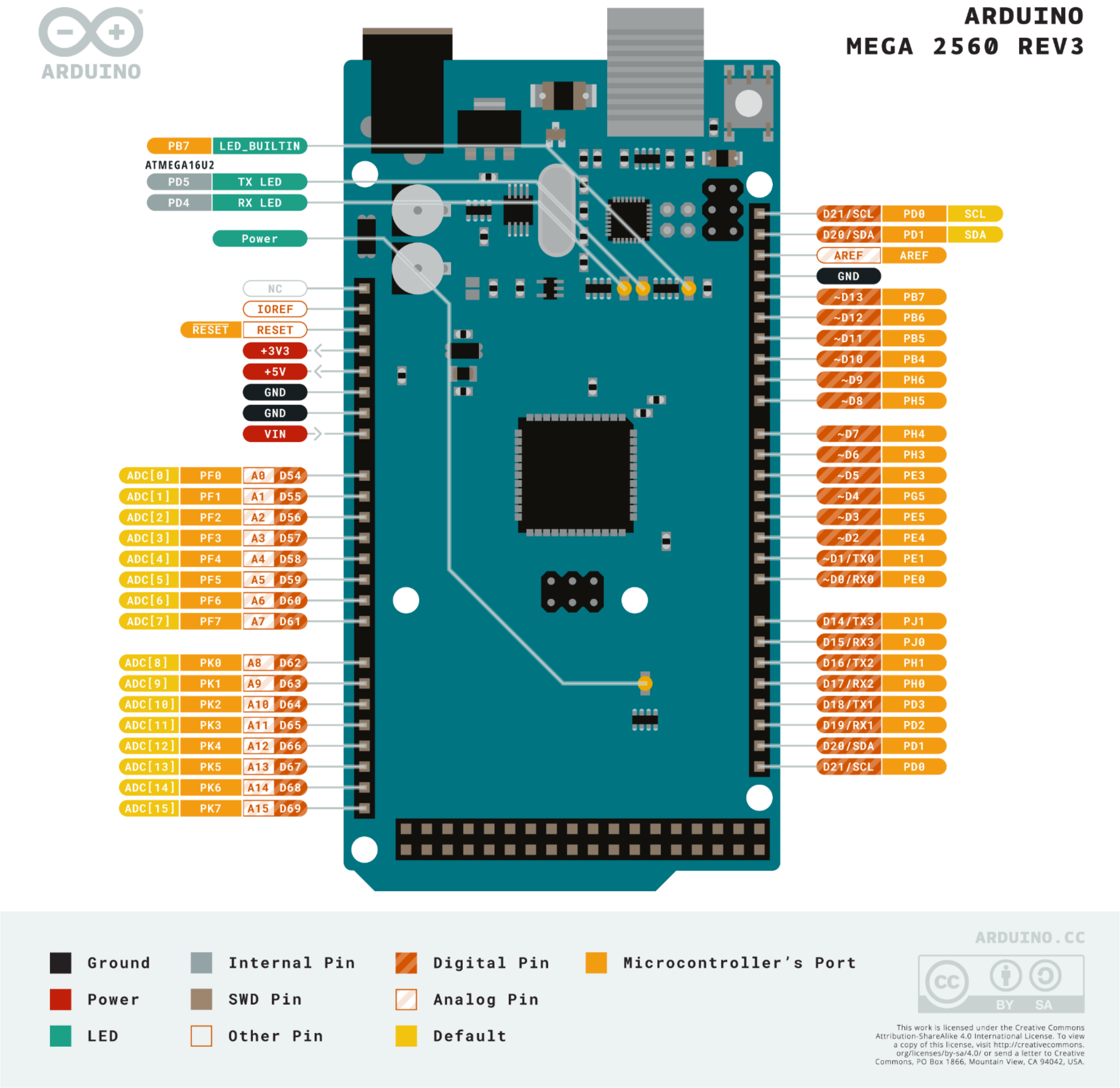
Reset

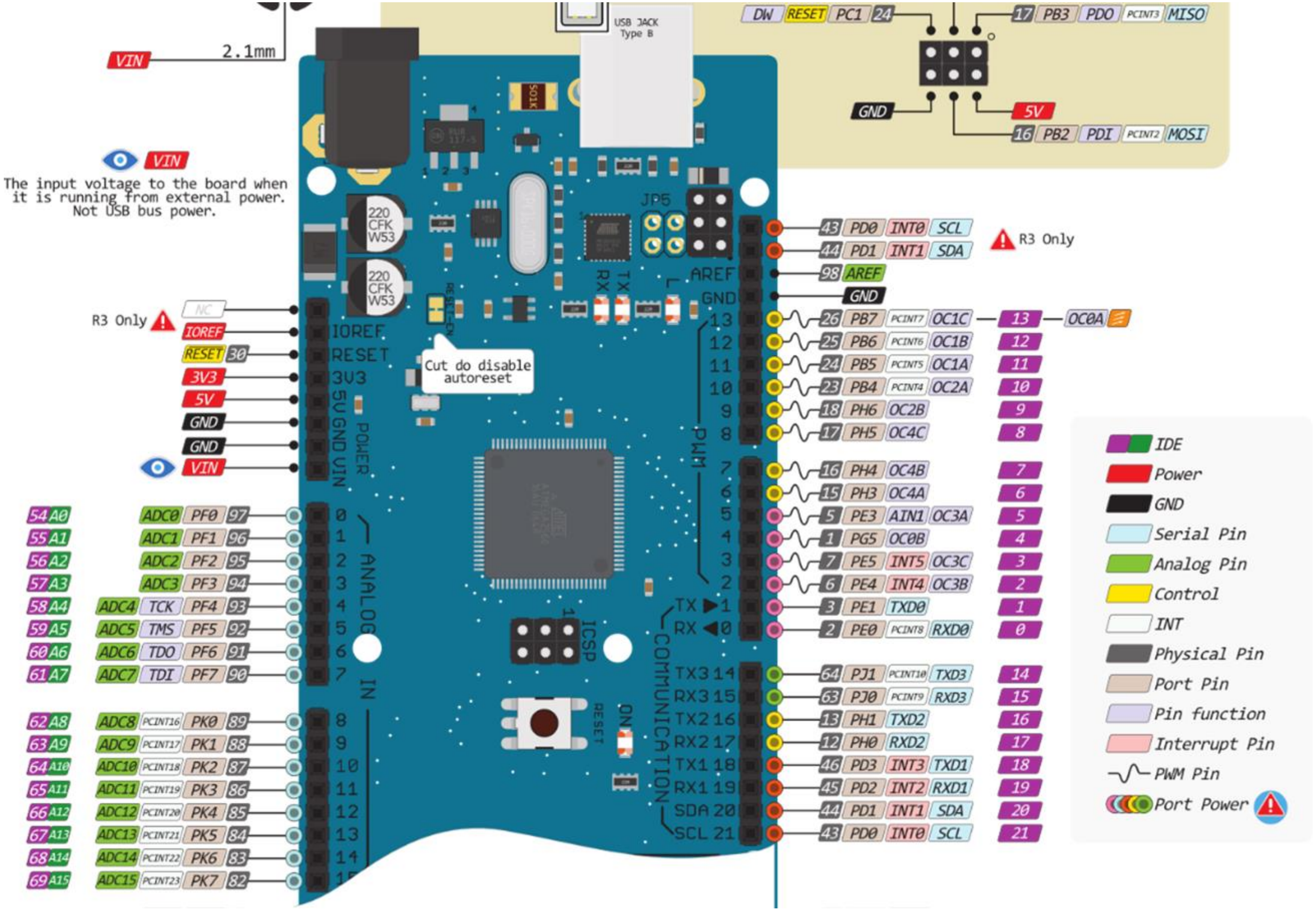
External Power

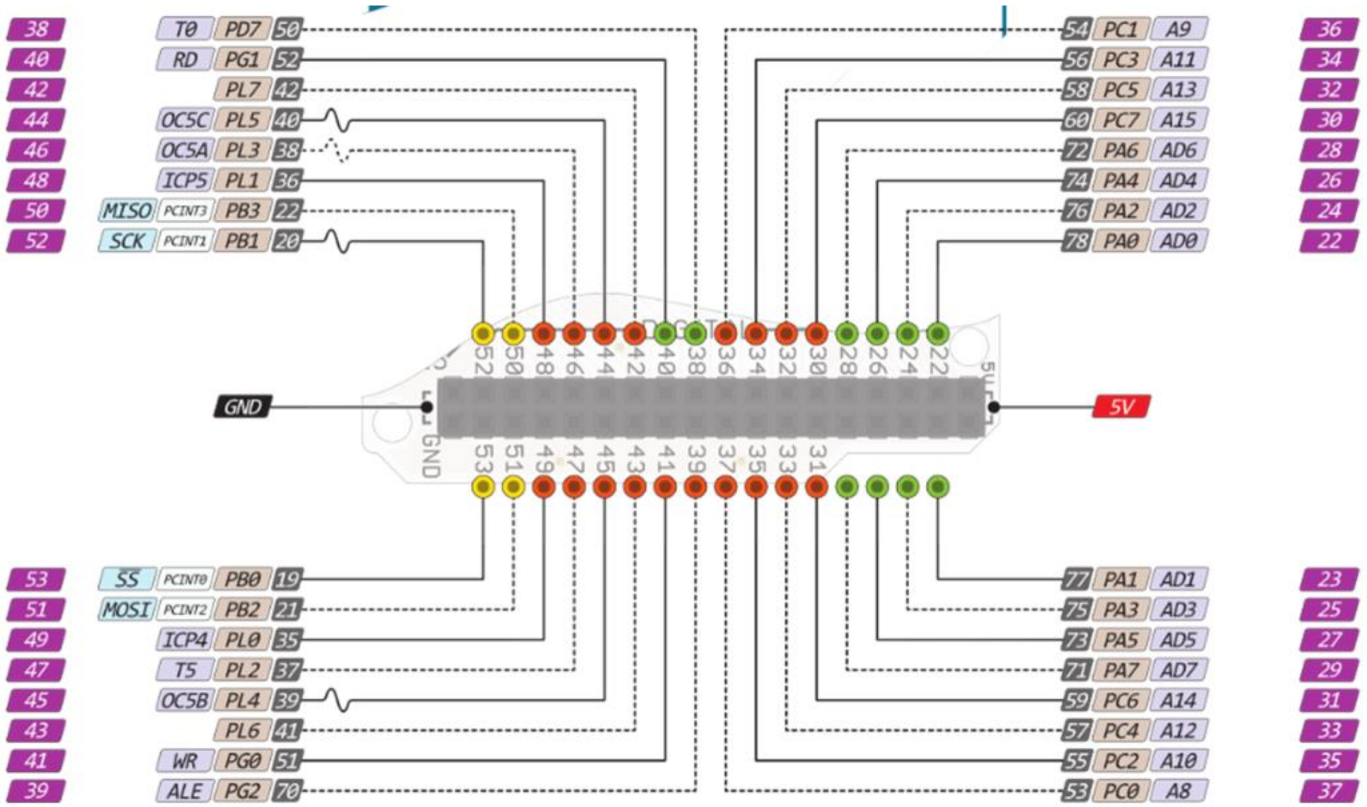
MEGA 2560

Figure 1-1. TQFP-pinout ATmega640/1280/2560



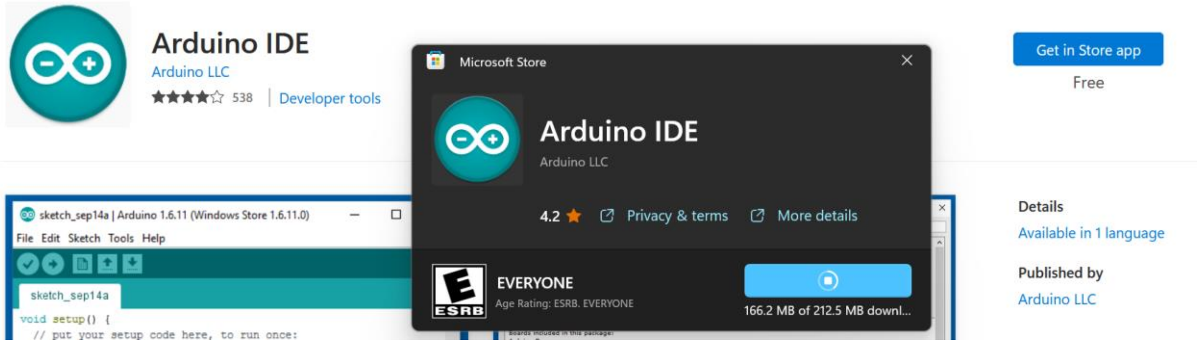

ARDUINO
ARDUINO
MEGA 2560 REV3
PB7 LED_BUILTIN
ATMEGA16U2
PD5 TX LED
PD4 RX LED
Power
NC
IOREF
RESET RESET
+3V3
+5V
GND
GND
VIN
ADC[0] PF0 A0 D54
ADC[1] PF1 A1 D55
ADC[2] PF2 A2 D56
ADC[3] PF3 A3 D57
ADC[4] PF4 A4 D58
ADC[5] PF5 A5 D59
ADC[6] PF6 A6 D60
ADC[7] PF7 A7 D61
ADC[8] PK0 A8 D62
ADC[9] PK1 A9 D63
ADC[10] PK2 A10 D64
ADC[11] PK3 A11 D65
ADC[12] PK4 A12 D66
ADC[13] PK5 A13 D67
ADC[14] PK6 A14 D68
ADC[15] PK7 A15 D69
D21/SCL PD0 SCL
D20/SDA PD1 SDA
AREF AREF
GND
~D13 PB7
~D12 PB6
~D11 PB5
~D10 PB4
~D9 PH6
~D8 PH5
~D7 PH4
~D6 PH3
~D5 PE3
~D4 PG5
~D3 PE5
~D2 PE4
~D1/TX0 PE1
~D0/RX0 PE0
D14/TX3 PJ1
D15/RX3 PJ0
D16/TX2 PH1
D17/RX2 PH0
D18/TX1 PD3
D19/RX1 PD2
D20/SDA PD1
D21/SCL PD0
Ground
Internal Pin
Digital Pin
Microcontroller's Port
Power
SWD Pin
Analog Pin
LED
Other Pin
Default
ARDUINO.CC
BY SA


VIN
2.1mm
USB JACK Type B
DW RESET PC1 24
17 PB3 PDO PCINT3 MISO
GND
5V
16 PB2 PDI PCINT2 MOSI
VIN
The input voltage to the board when it is running from external power. Not USB bus power.
43 PD0 INT0 SCL
44 PD1 INT1 SDA
R3 Only
98 AREF
GND
26 PB7 PCINT7 OC1C 13 OC0A
25 PB6 PCINT6 OC1B 12
24 PB5 PCINT5 OC1A 11
23 PB4 PCINT4 OC2A 10
18 PH6 OC2B 9
17 PH5 OC4C 8
16 PH4 OC4B 7
15 PH3 OC4A 6
5 PE3 AIN1 OC3A 5
1 PG5 OC0B 4
7 PE5 INT5 OC3C 3
6 PE4 INT4 OC3B 2
3 PE1 TXD0 1
2 PE0 PCINT8 RXD0 0
64 PJ1 PCINT10 TXD3 14
63 PJ0 PCINT9 RXD3 15
13 PH1 TXD2 16
12 PH0 RXD2 17
46 PD3 INT3 TXD1 18
45 PD2 INT2 RXD1 19
44 PD1 INT1 SDA 20
43 PD0 INT0 SCL 21
R3 Only
NC
IOREF
RESET 30
3V3
5V
GND
GND
VIN
IOREF RESET 3V3 5V GND GND VIN POWER
Cut do disable autoreset
JP5
AREF GND 13 12 11 10 9 8 PWM 7 6 5 4 3 2 TX 1 RX 0
TX3 14 RX3 15 TX2 16 RX2 17 TX1 18 RX1 19 SDA 20 SCL 21 COMMUNICATION
ICSP
RESET
ON
54 A0 ADC0 PF0 97
55 A1 ADC1 PF1 96
56 A2 ADC2 PF2 95
57 A3 ADC3 PF3 94
58 A4 ADC4 TCK PF4 93
59 A5 ADC5 TMS PF5 92
60 A6 ADC6 TDO PF6 91
61 A7 ADC7 TDI PF7 90
62 A8 ADC8 PCINT16 PK0 89
63 A9 ADC9 PCINT17 PK1 88
64 A10 ADC10 PCINT18 PK2 87
65 A11 ADC11 PCINT19 PK3 86
66 A12 ADC12 PCINT20 PK4 85
67 A13 ADC13 PCINT21 PK5 84
68 A14 ADC14 PCINT22 PK6 83
69 A15 ADC15 PCINT23 PK7 82
0 1 2 3 4 5 6 7 ANALOG IN 8 9 10 11 12 13 14 15
IDE
Power
GND
Serial Pin
Analog Pin
Control
INT
Physical Pin
Port Pin
Pin function
Interrupt Pin
PWM Pin
Port Power


38 T0 PD7 50
40 RD PG1 52
42 PL7 42
44 OC5C PL5 40
46 OC5A PL3 38
48 ICP5 PL1 36
50 MISO PCINT3 PB3 22
52 SCK PCINT1 PB1 20
54 PC1 A9 36
56 PC3 A11 34
58 PC5 A13 32
60 PC7 A15 30
72 PA6 AD6 28
74 PA4 AD4 26
76 PA2 AD2 24
78 PA0 AD0 22
DIGITAL
52 50 48 46 44 42 40 38 36 34 32 30 28 26 24 22
5V
GND
53 51 49 47 45 43 41 39 37 35 33 31
GND
5V
53 SS PCINT0 PB0 19
51 MOSI PCINT2 PB2 21
49 ICP4 PL0 35
47 T5 PL2 37
45 OC5B PL4 39
43 PL6 41
41 WR PG0 51
39 ALE PG2 70
77 PA1 AD1 23
75 PA3 AD3 25
73 PA5 AD5 27
71 PA7 AD7 29
59 PC6 A14 31
57 PC4 A12 33
55 PC2 A10 35
53 PC0 A8 37

Software: Arduino IDE

- https://www.arduino.cc/en/software
- Download and install



Arduino IDE

Arduino LLC

538 | Developer tools

Microsoft Store

Arduino IDE

Arduino LLC

4.2 ★ Privacy & terms More details

EVERYONE

Age Rating: ESRB. EVERYONE

166.2 MB of 212.5 MB downl...

Get in Store app

Free

Details

Available in 1 language

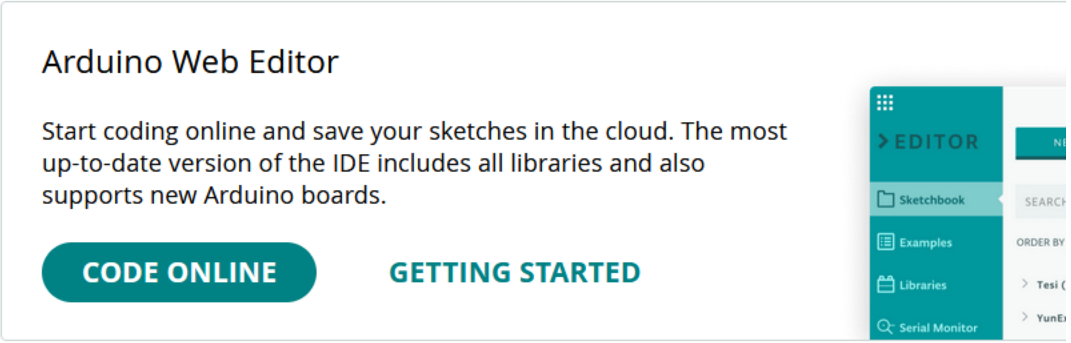
Published by

Arduino LLC

sketch_sep14a | Arduino 1.6.11 (Windows Store 1.6.11.0)

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

- Or use the web editor



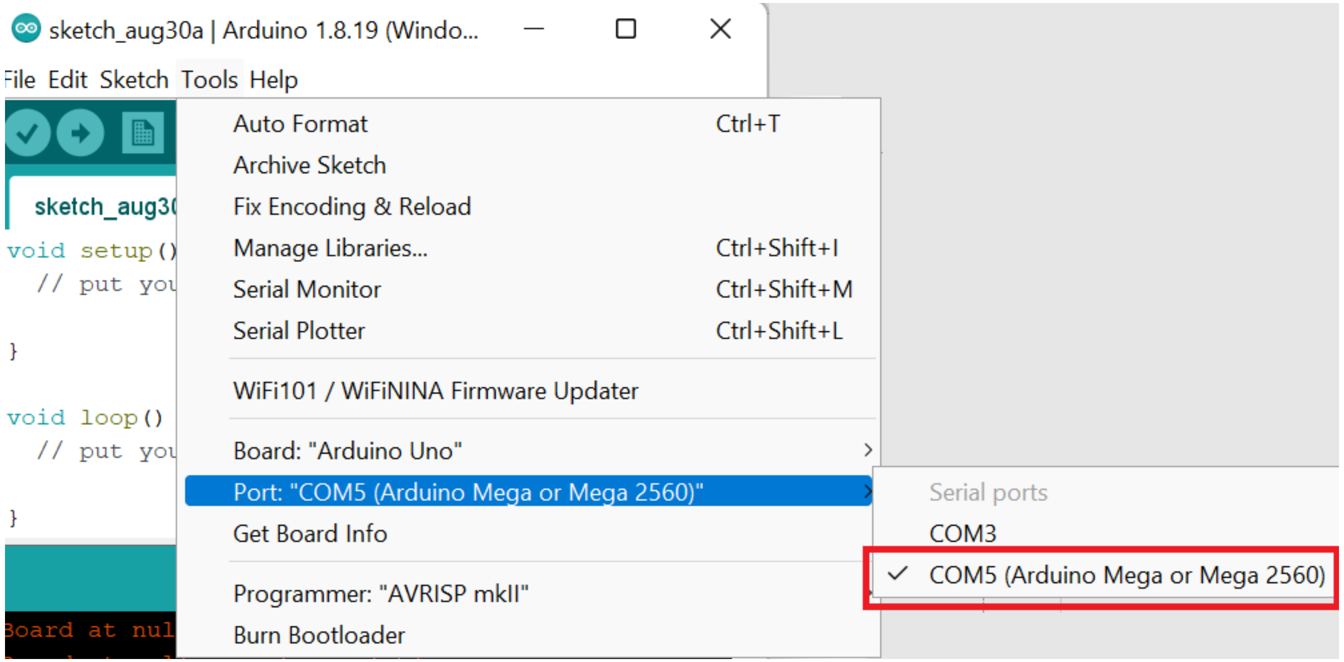
Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

CODE ONLINE GETTING STARTED

Connecting Software to Board

- Connect the board to computer via USB
- ON light should come on
- Click on Tools > Port > Arduino Mega2560
- Click Tools > Board > Mega2560



sketch_aug30a | Arduino 1.8.19 (Windo...
File Edit Sketch Tools Help
sketch_aug3(
void setup()
// put you
}
void loop()
// put you
}
Board at nul
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader
Serial ports
COM3

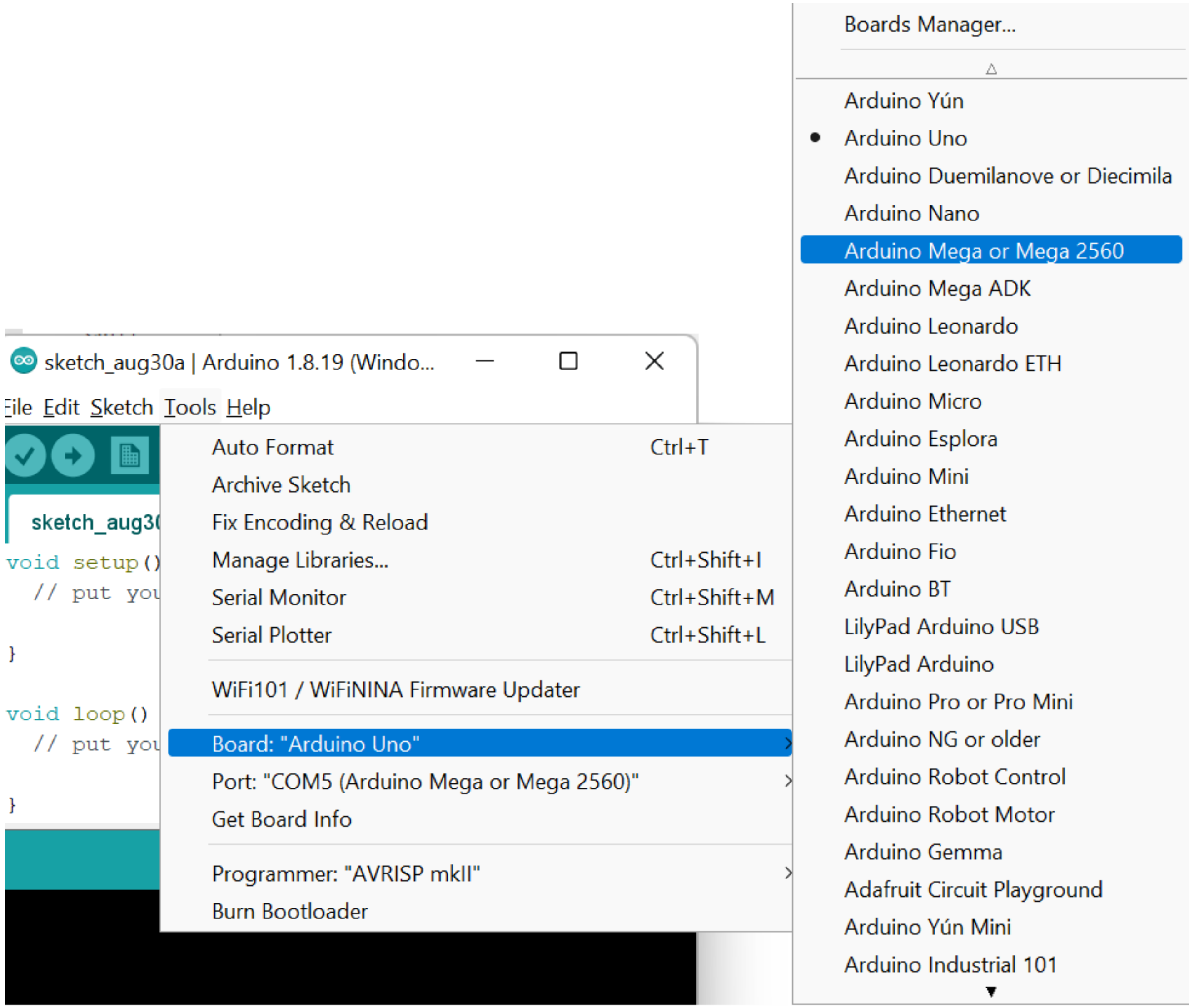
Board: "Arduino Uno"

Port: "COM5 (Arduino Mega or Mega 2560)"

✓ COM5 (Arduino Mega or Mega 2560)

Programmer: "AVRISP mkII"

- If completed successfully, the bottom of the Arduino IDE will show ATMega 2560 on COM5



```
void setup()
  // put you

}

void loop()
  // put you

}
```

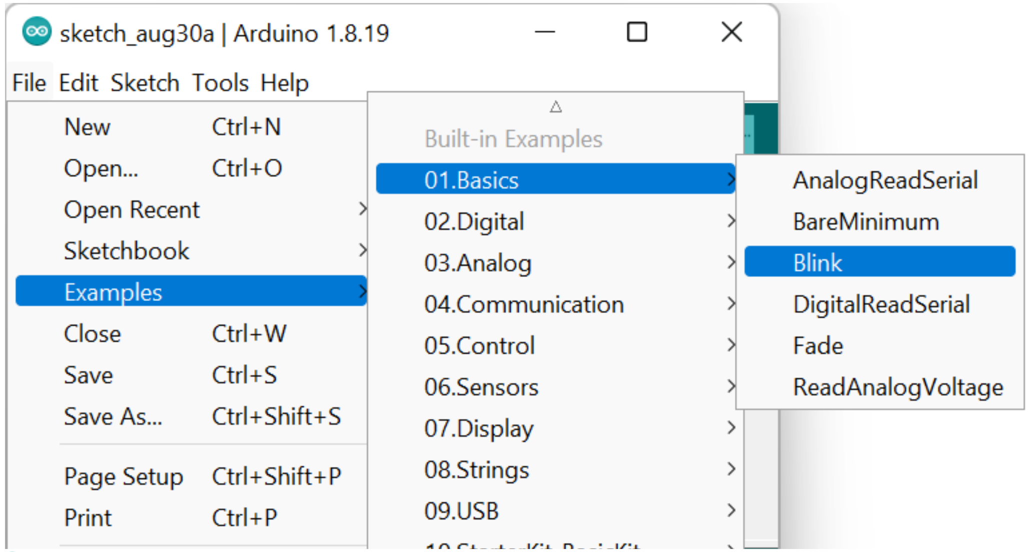
Board: "Arduino Uno"

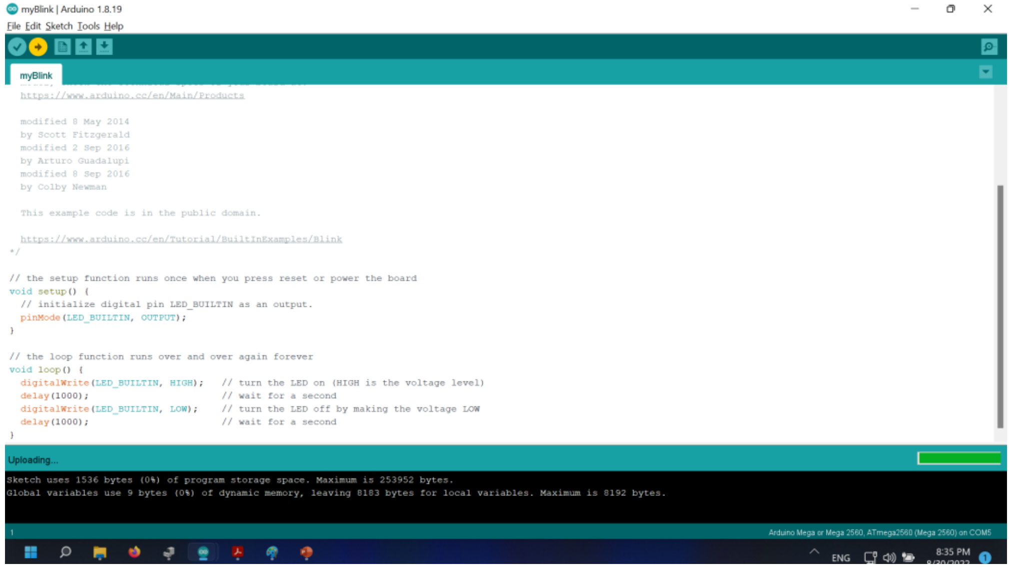
Port: "COM5 (Arduino Mega or Mega 2560)"

Programmer: "AVRISP mkII"

Run Example Code: Blink

- Click File>Examples>Basics>Blink
- Save the file as myBlink
  - Create a folder for ENEE3587
- Verify the code:
  - Click check icon
- Upload the program to board:
  - Click upload icon ➲
- Manipulate the program to blink every 250ms instead of 1s
- RESET the board





```
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  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.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/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
}
```

Uploading...

```
Sketch uses 1536 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for local variables. Maximum is 8192 bytes.
```

Arduino C Template (wo interrupts)

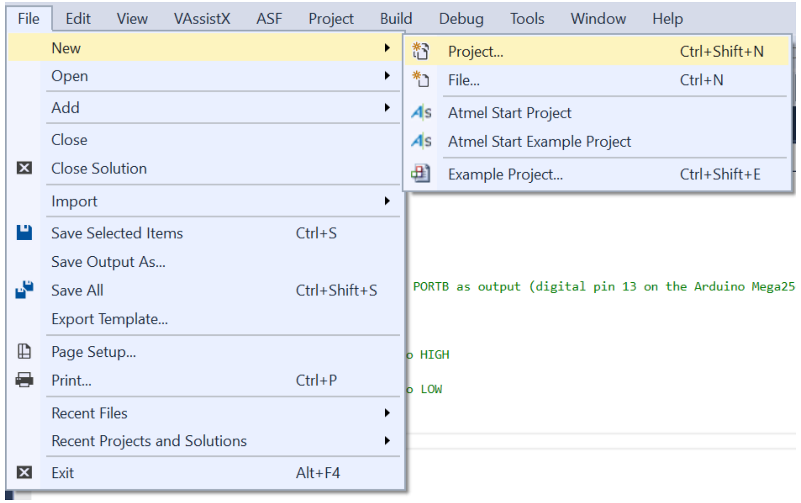
```
/*
 * Project: Title
 * Description:
 * Name :
 */

#include <avr/io.h>
// Other includes

//global variables
//function protos

// function prototypes
void main(void)
{
    while(1)                // infinite loop
    {
         ...
    }
}

// other functions
```

Blink Program

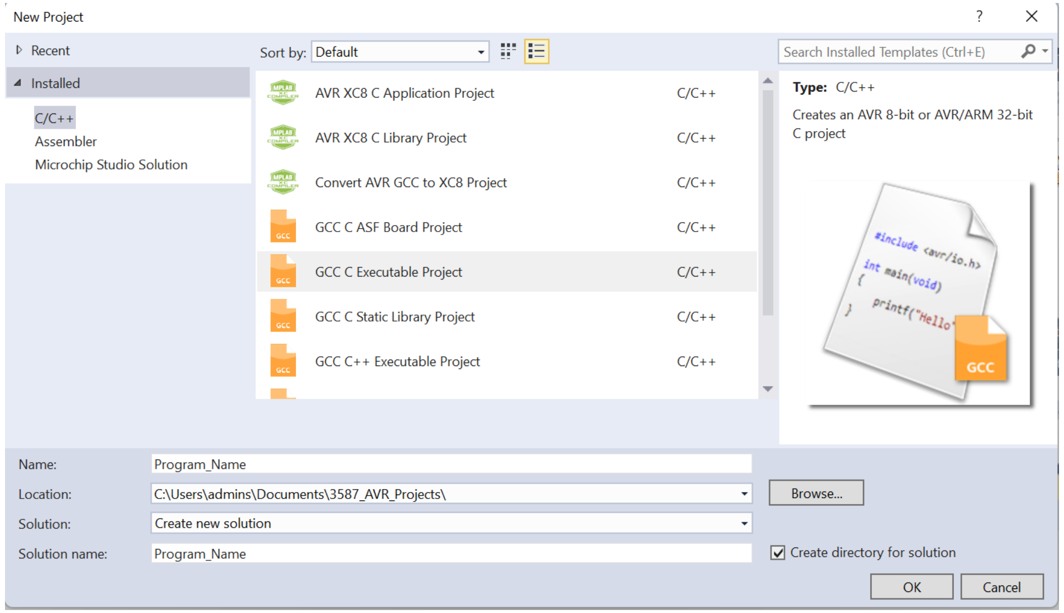
```
#include <avr/io.h>
#include <avr/interrupt.h>

void delay1ms(int);

int main(void)
{
  DDRB = 0b10000;          //LED connected to pin13 on board
  int i = 0;
  while (1)
  { PORTB = 0b10000;
    delay1ms(500);
    PORTB = 0b00000;
    delay1ms(500);
  }
}


void delay1ms(int t)
{
  unsigned int i,j;
  for (i=0; i<t ; i++)
      for (j=0; j< 1453; j++)
          DDRB &= 0xff;

}
```

Microchip Studio

- https://www.microchip.com/en-us/tools-resources/develop/microchip-studio
- Download and Install
- Create a project: Click File > New > Project



- Select C/C++
- Select GCC C Executable Project
- Name your project
- Create and select a location for your project
- Select AVR 2560


New Project
?
Recent
Installed
C/C++
Assembler
Microchip Studio Solution
Sort by: Default
Search Installed Templates (Ctrl+E)
AVR XC8 C Application Project C/C++
AVR XC8 C Library Project C/C++
Convert AVR GCC to XC8 Project C/C++
GCC C ASF Board Project C/C++
GCC C Executable Project C/C++
GCC C Static Library Project C/C++
GCC C++ Executable Project C/C++
#include <avr/io.h>
int main(void)
{
printf("Hello
}
GCC
Browse...
OK
Cancel

Type: C/C++

Creates an AVR 8-bit or AVR/ARM 32-bit C project

Name: Program_Name

Location: C:\Users\admins\Documents\3587_AVR_Projects\

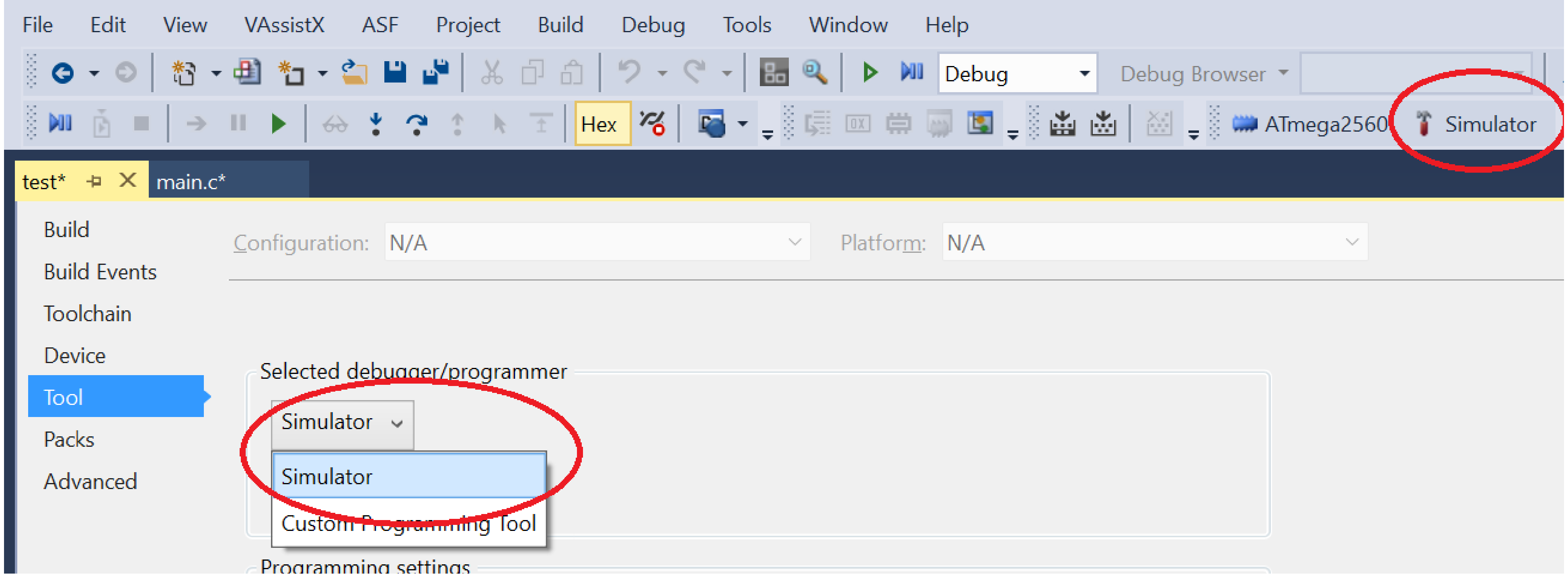
Solution: Create new solution

Solution name: Program_Name

☑ Create directory for solution

Microchip Studio Toolchain: Simulator

- Useful for debugging code w/o board



Configuration: N/A

Platform: N/A

Selected debugger/programmer

Simulator

Microchip Studio Toolchain: Building & Uploading

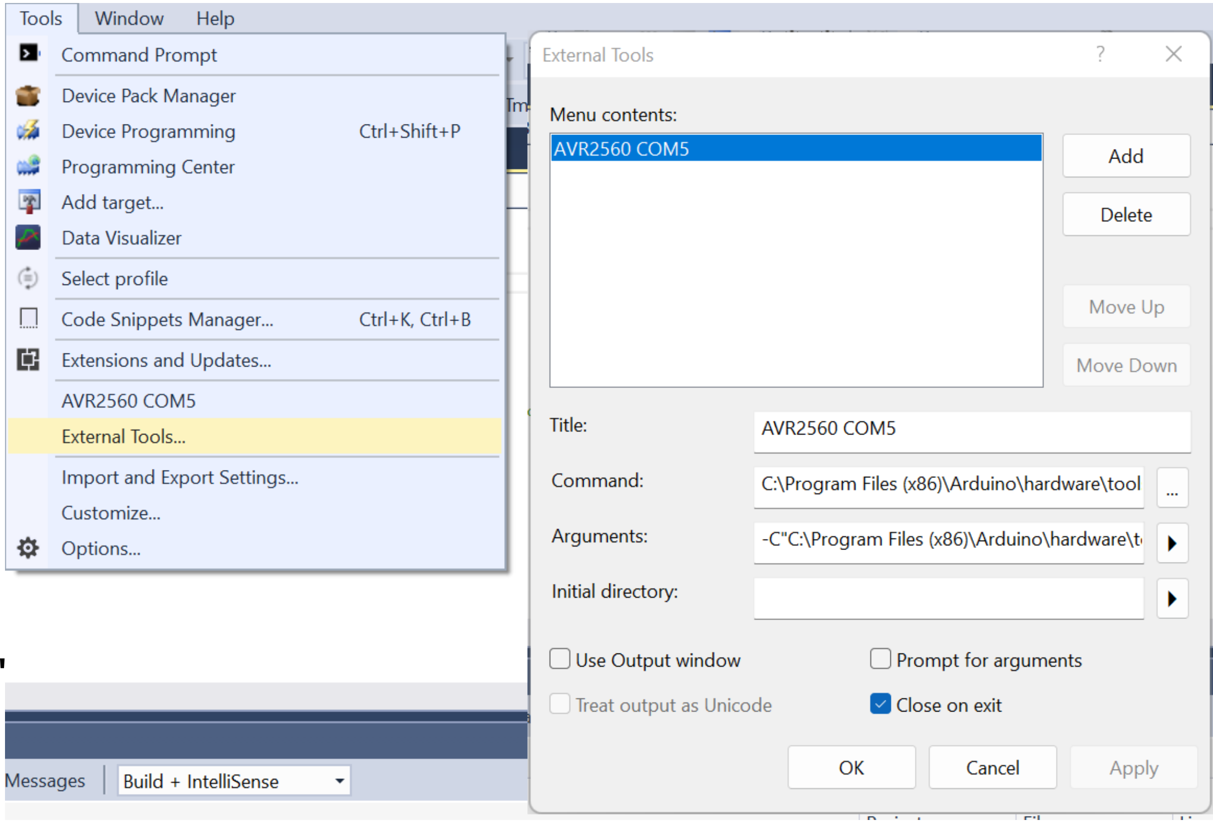
- Click Tools Tab > External Tools
- Name: User’s tool name
- Command:

```
C:\Program Files
(x86)\Arduino\hardware\tools\avr\bin\avrdude.exe
```

- Arguments:

```
-C"C:\Program Files
(x86)\Arduino\hardware\tools\avr\etc\avrdude.conf'
-patmega2560 -cwiring -P\\.\COM5 -b115200 -D -
Uflash:w:"$(ProjectDir)Debug\$(TargetName).hex":i
```

  - Path, COM port may vary



Menu contents:
AVR2560 COM5

Title: AVR2560 COM5

Command: C:\Program Files (x86)\Arduino\hardware\tool

Arguments: -C"C:\Program Files (x86)\Arduino\hardware\t

Initial directory:

☐ Use Output window ☐ Prompt for arguments

☐ Treat output as Unicode ☑ Close on exit

Microchip Studio C Template (wo interrupts)

```
/*
 * Project: Title
 * Description:
 * Name :
 */

#define F_CPU 16000000L          // Specify 16MHz frequency
#include <avr/io.h>
// Other includes

// function prototypes
void main(void)
{
    while(1)
    {
      ...
    }
}

// other functions
```

Blink Program

```
/*
 * Project: Title
 * Description:
 * Name :
 */

#define F_CPU 16000000L // Specify oscillator frequency
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRB = 0b10000000; // configure pin 7 of PORTB as output (digital pin 13 on the Arduino Mega2560)

    while(1)
    {
        PORTB = 0b10000000; // set 7th bit to HIGH
        _delay_ms(500);
        PORTB = 0b00000000; // set 7th bit to LOW
        _delay_ms(500);
    }
}
```

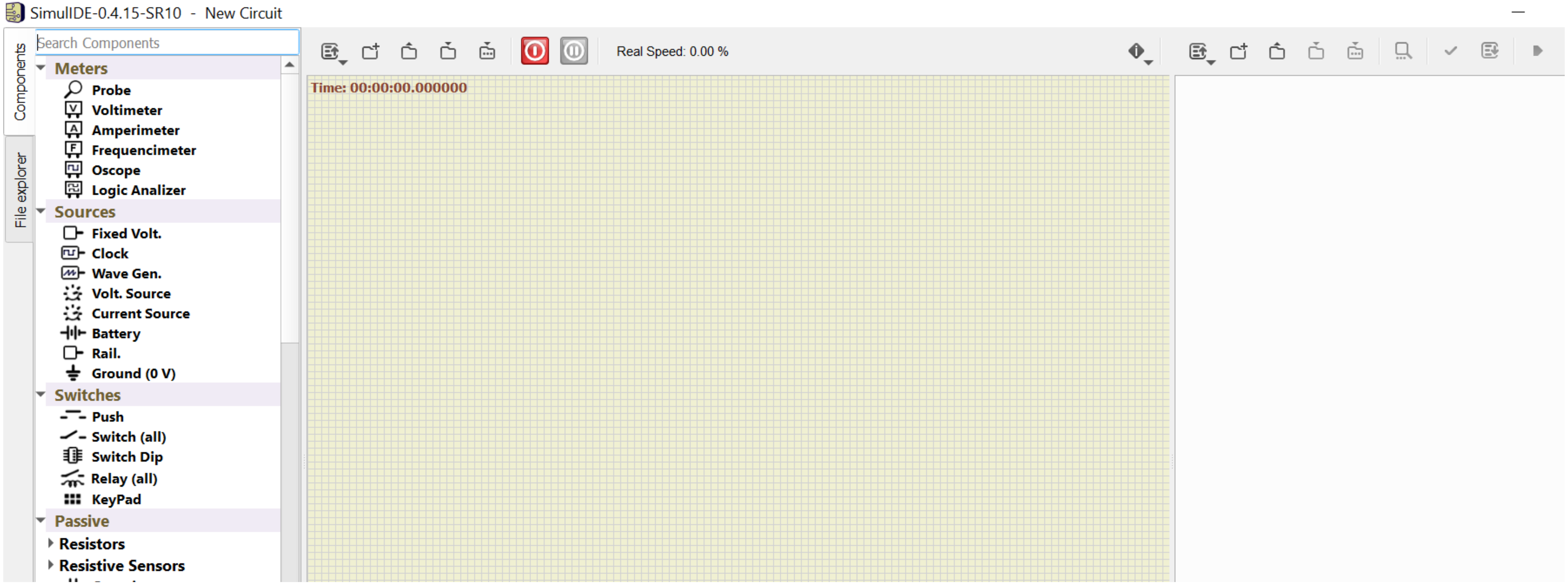
Build and Flash

- Click Build > Build Solution
  - Alternatively, press F7
- Click Tool > AVR2560 COM5
  - Tool name based on your setup of external tool

Simulation IDE

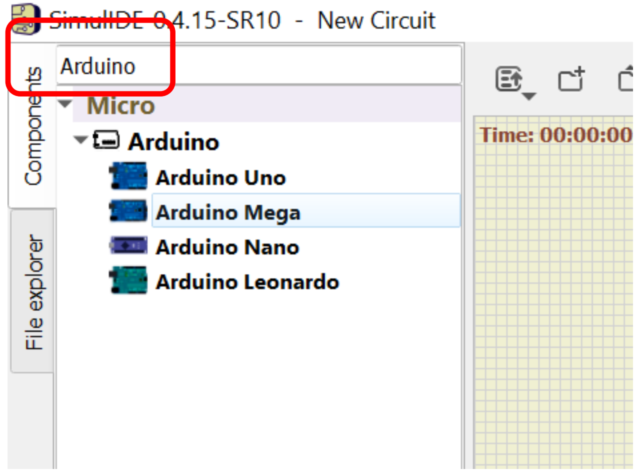
- Download: https://www.simulide.com/p/downloads.html
  - Get the stable version
- Unzip the file
- Run the binary: simulide.exe in the bin folder
- Check documentation: https://www.simulide.com/p/blog-page.html

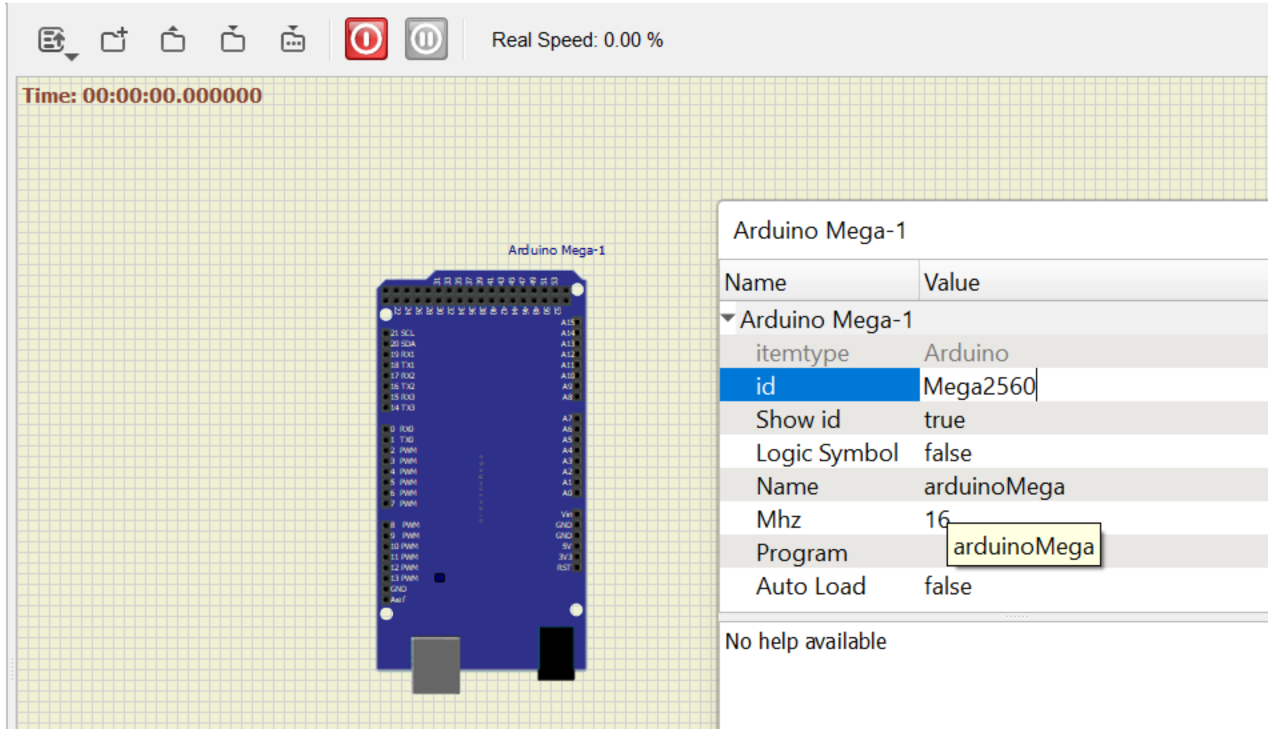
Simul IDE Demo



Components **Circuit Building Area** **Coding Area**

Load Arduino





Arduino Mega-1

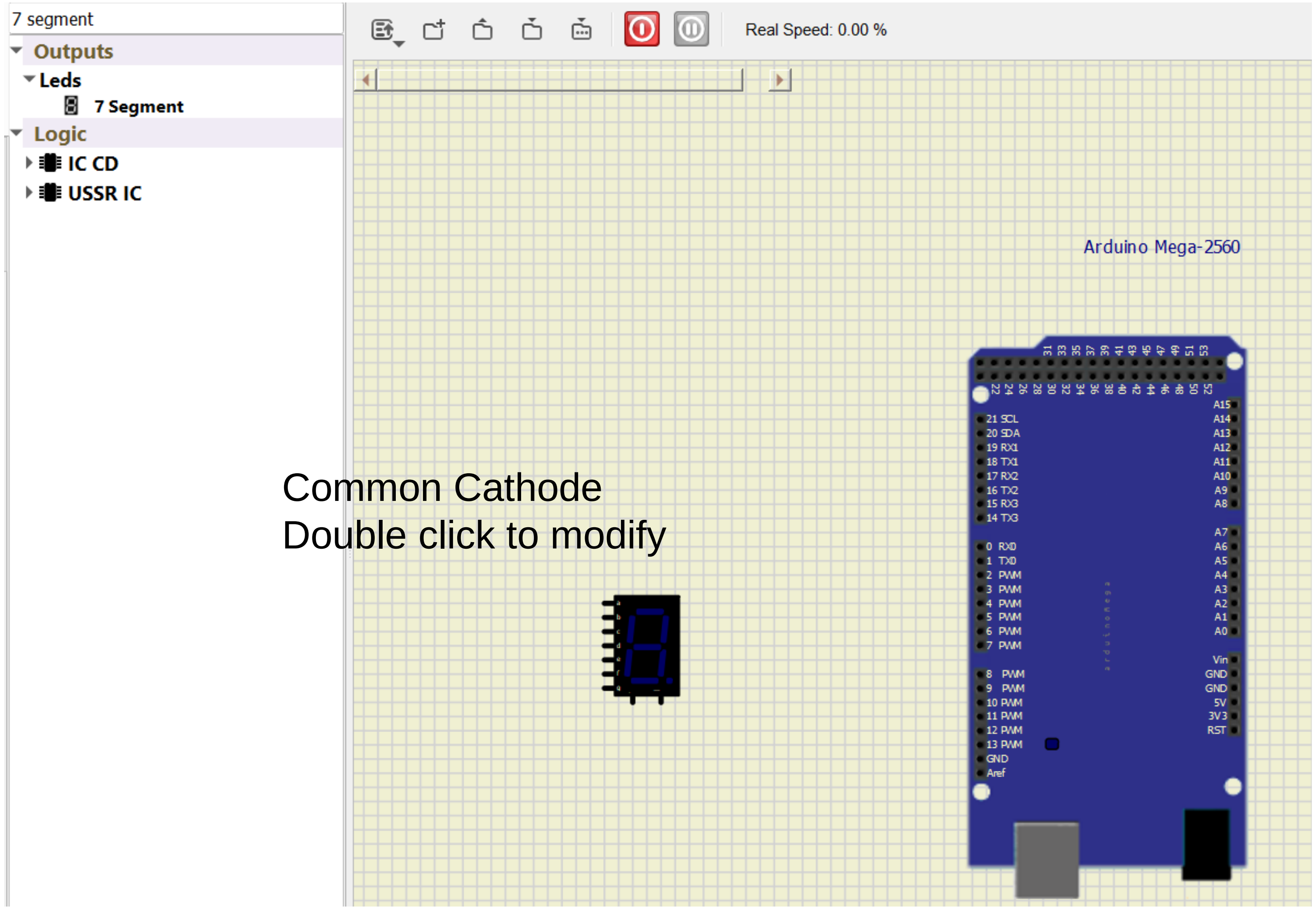
Name	Value
Arduino Mega-1	
itemtype	Arduino
id	Mega2560
Show id	true
Logic Symbol	false
Name	arduinoMega
Mhz	16
Program	arduinoMega
Auto Load	false

Search for Arduino

Drag & Drop Mega

Double-click, modify IDE

Load Components 7SD

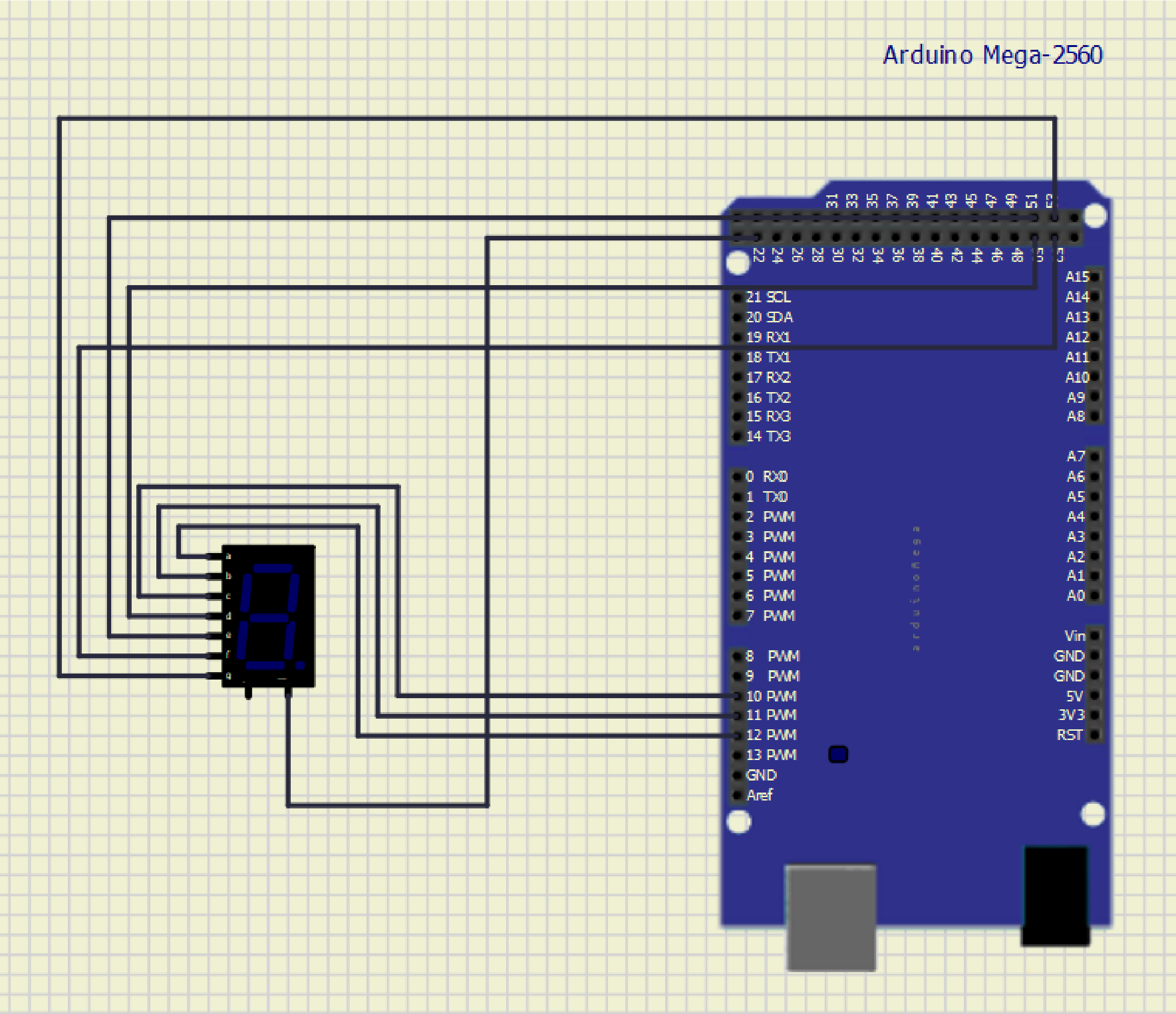


Common Cathode
Double click to modify

Wire 7SD

Segments a-g connected to PTB6-PTB0.

GND connected to PTA0

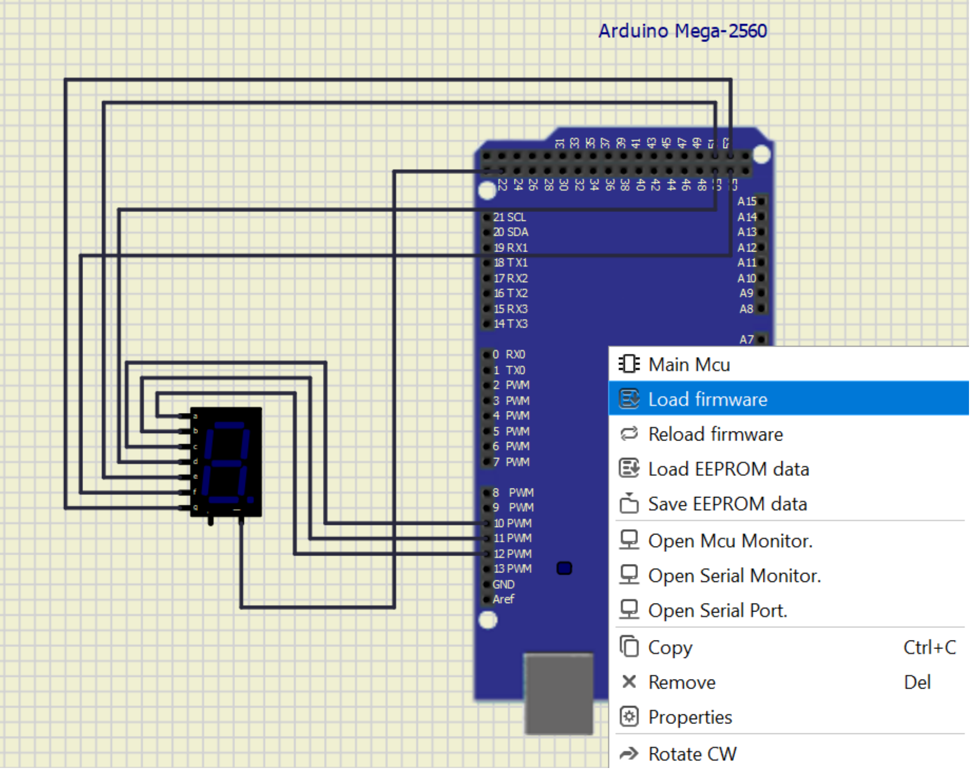


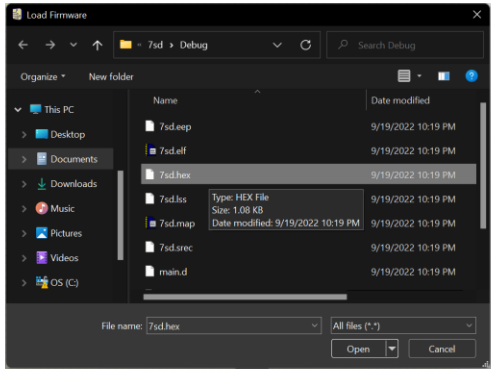
Code & Build in Microchip Studio

- Display “0”-”9” in 1sec intervals

```
#define F_CPU 16000000L // Specify 16MHz frequency
#include <avr/io.h>
#include <util/delay.h>

int main(void)
{    //common cathode. segA -> PTB7; segg -> PTB0
     char ssd[] = {0x7e, 0x30, 0x6d, 0x79, 0x33, 0x5b, 0x5f, 0x70, 0x7f, 0x7b}; //0-9
     DDRB = 0x7F;
     DDRA = 0x01;                 //PTA0: enable/GND on 7SD
     PORTA = ~1;                  //Enable 7SD
     while (1)
    {        for (int i = 0; i<10; i++)
             {       PORTB = ssd[i];          //Display pattern
                     _delay_ms(1000);         //wait for 1e. Built in function.
             }
    }
}
```

Load Code

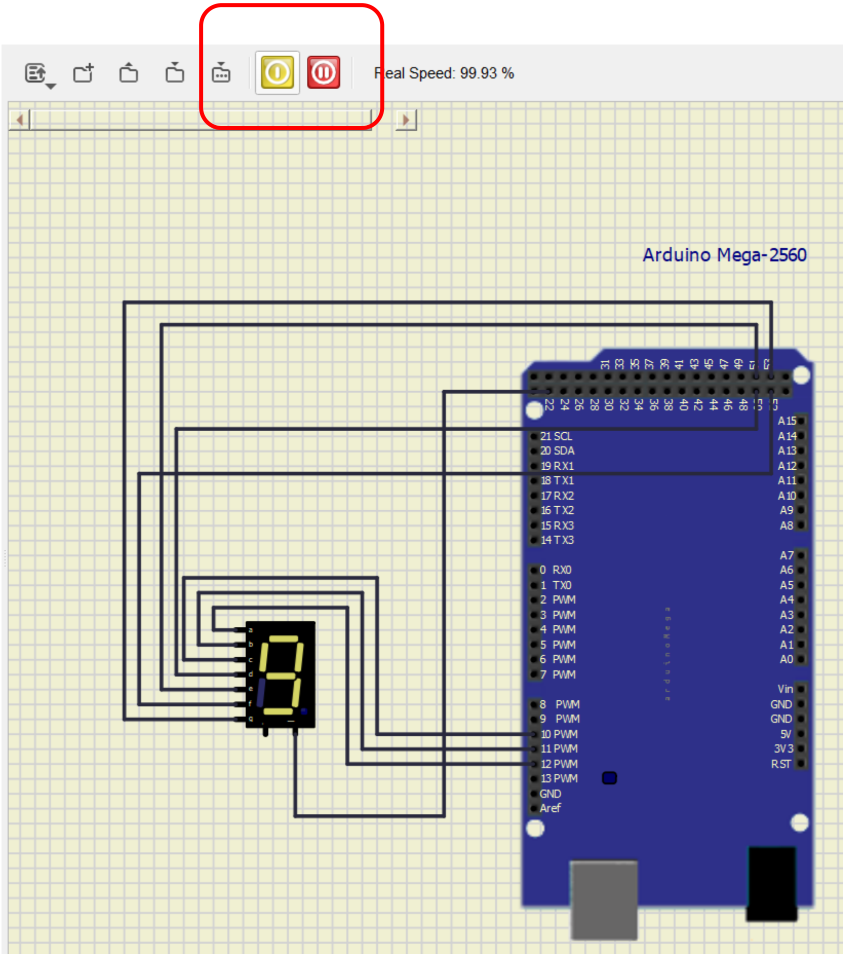




Name	Date modified
7sd.eep	9/19/2022 10:19 PM
7sd.elf	9/19/2022 10:19 PM
7sd.hex	9/19/2022 10:19 PM
7sd.lss	9/19/2022 10:19 PM
7sd.map	9/19/2022 10:19 PM
7sd.srec	9/19/2022 10:19 PM
main.d	9/19/2022 10:19 PM

Type: HEX File
Size: 1.08 KB
Date modified: 9/19/2022 10:19 PM

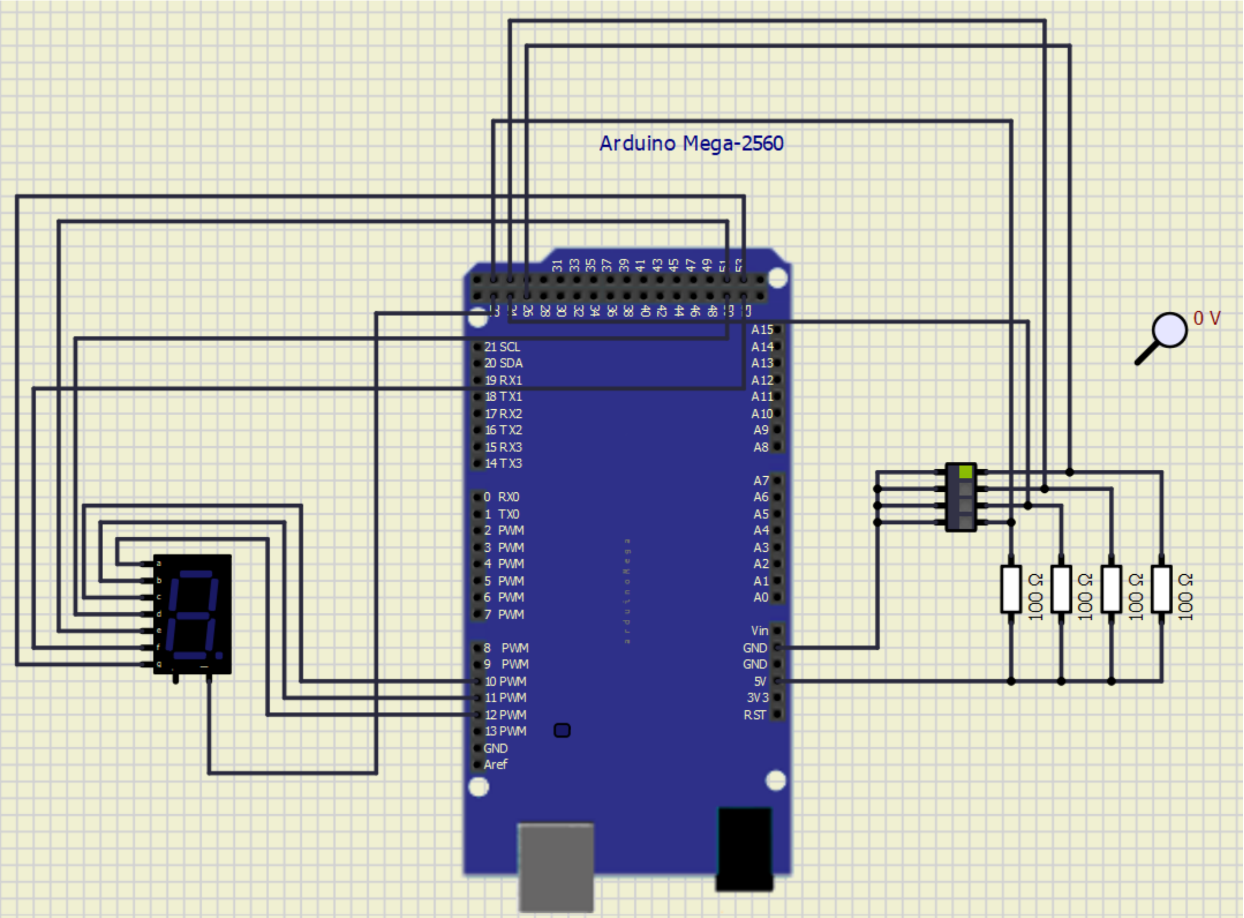
File name: 7sd.hex
All files (*.*)



Load Firmware: Look for Hex File Build.
Click simulation button

DIP switch + 7SD

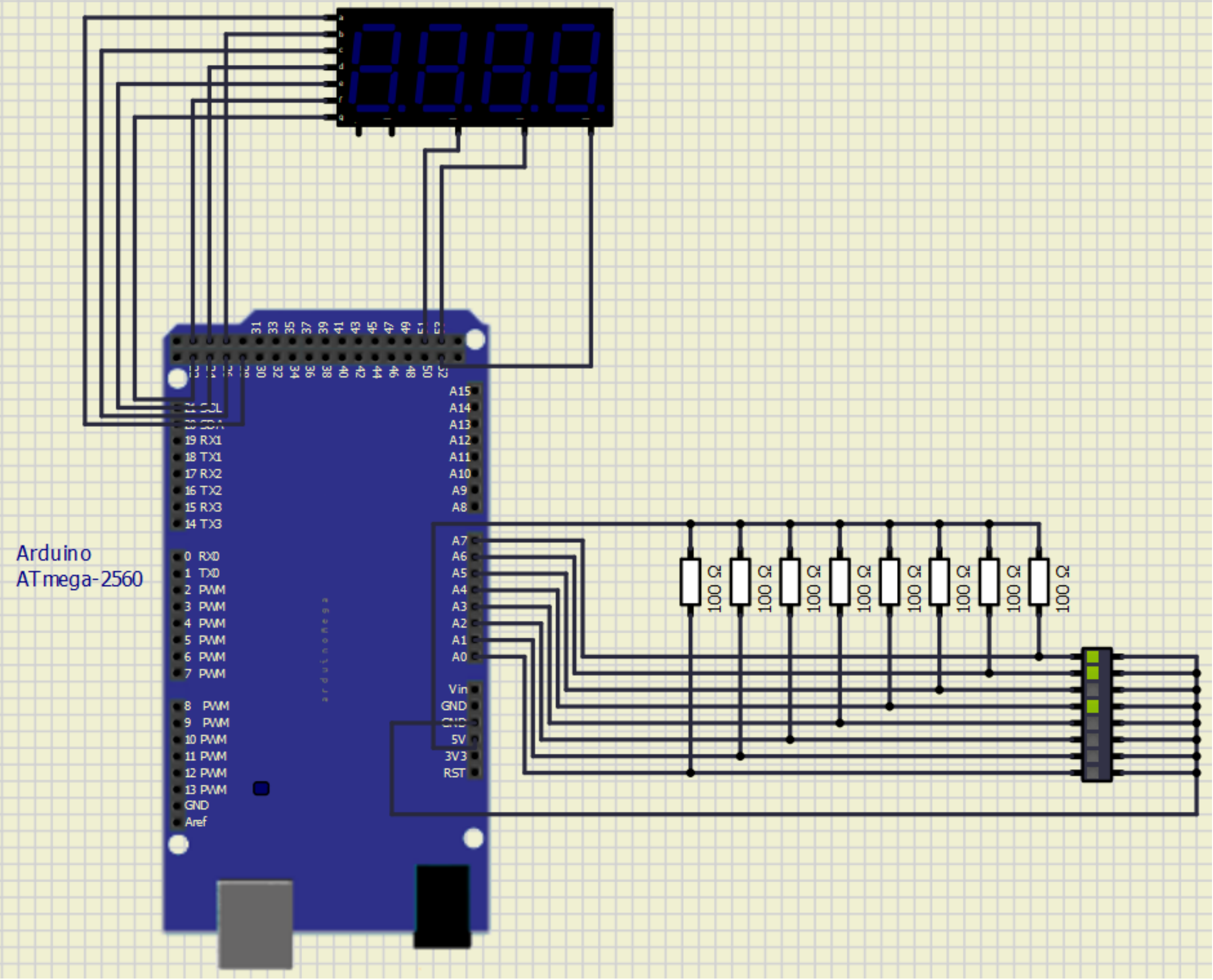
- 4xDIP switch is connected to PTA3-1; 7SD connected to PTB6-0. Read DIP as a binary number and display on 7SD as a decimal. Switch off 7SD for inputs 9-15.
- Algorithm:
  - Set port pins for input/output
  - Read DIP
  - Process DIP input
    - Complement
    - Shift right once
    - Isolate last 4bits
  - Use DIP input as index for 7SD array



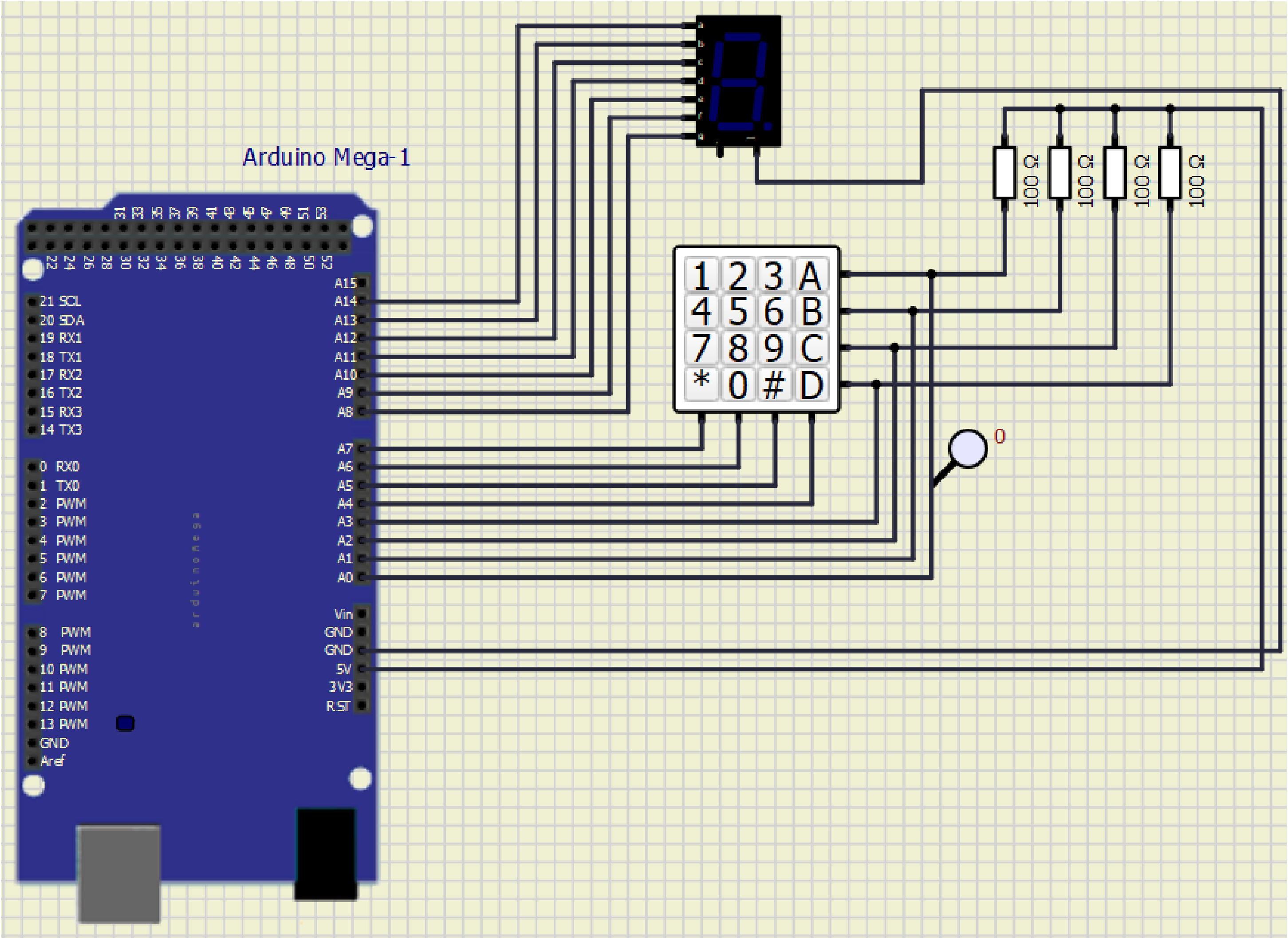
DIP + 7SD Code

```c
int main(void)
{      char ssd[] = {0x7e, 0x30, 0x6d, 0x79, 0x33, 0x5b, 0x5f, 0x70, 0x7f, 0x7b};
       unsigned char DIP4;

       DDRB = 0x7F;
       DDRA = 0x01;

       while (1)
       {       DIP4 = (~PINA >> 1) & 0b00001111;   //isolate PTA4..PTA1
               if (DIP4 < 10)
               {      PORTB = ssd[(int) DIP4];
                      PORTA &= 0xFE;               //GND 7SD to display
               }
               else
                      PORTA |= 1;                  //turn off 7SD for 10-15

    }
}
```

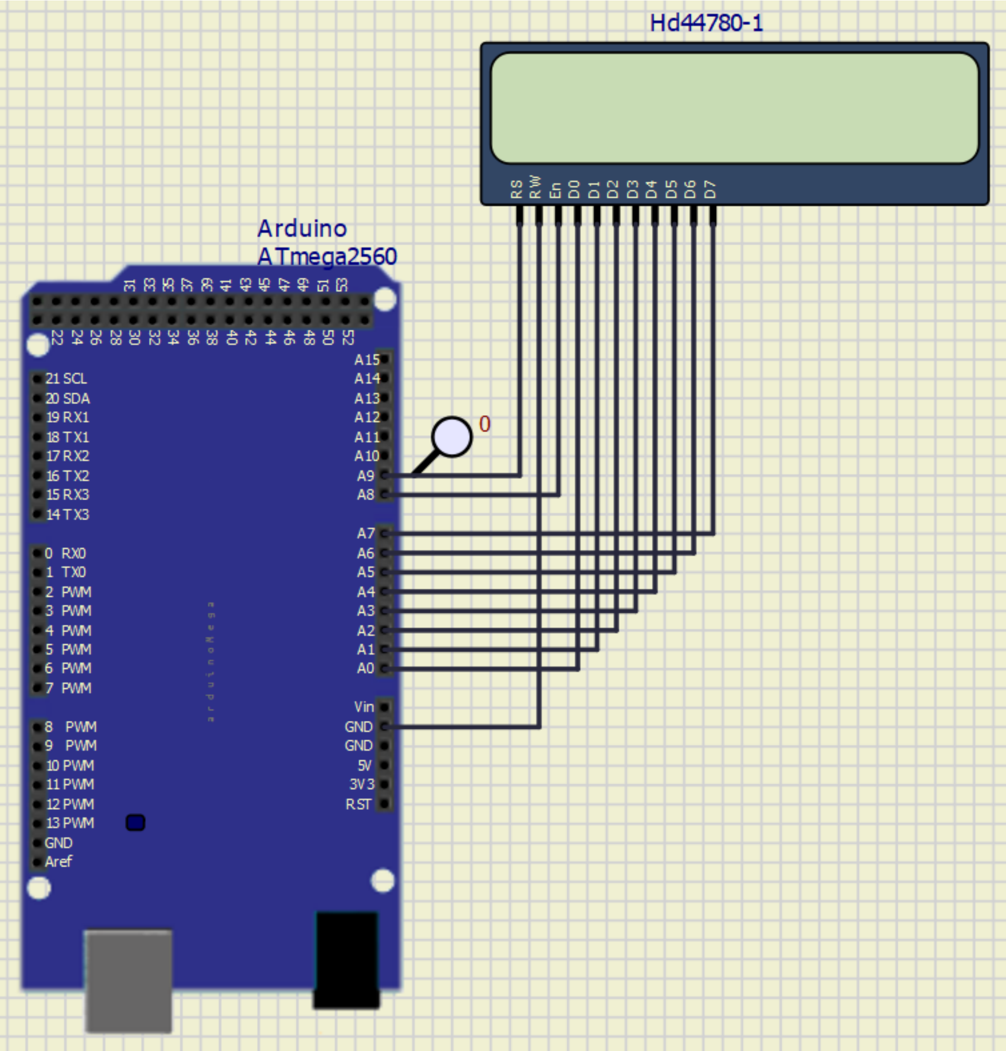
8DIP input, 4x7SD ouput



4x4 Keypad input, 7SD output



Interfacing LCD


Hd44780-1
RS
RW
En
D0
D1
D2
D3
D4
D5
D6
D7
Arduino
ATmega2560
0
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
Vin
GND
GND
5V
3V3
RST
21 SCL
20 SDA
19 RX1
18 TX1
17 RX2
16 TX2
15 RX3
14 TX3
0 RX0
1 TX0
2 PWM
3 PWM
4 PWM
5 PWM
6 PWM
7 PWM
8 PWM
9 PWM
10 PWM
11 PWM
12 PWM
13 PWM
GND
Aref