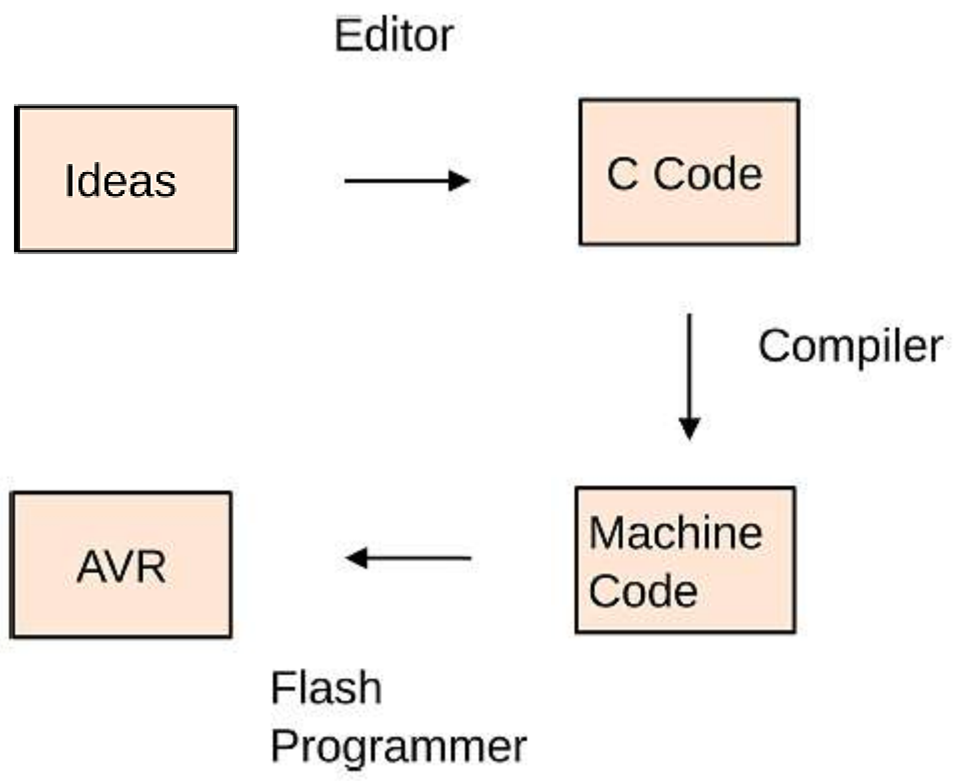
EMBEDDED SYSTEMS CMPE-453

Department of Computer Engineering

Programming AVR Microcontroller

PROGRAMMING AVR MICROCONTOLLER

Tool-Chain



- Arduino IDE and board combines all these steps into one.
- Write the C code in Arduino IDE
- Compile it.
- If no errors, upload the machine code to AVR microcontroller.

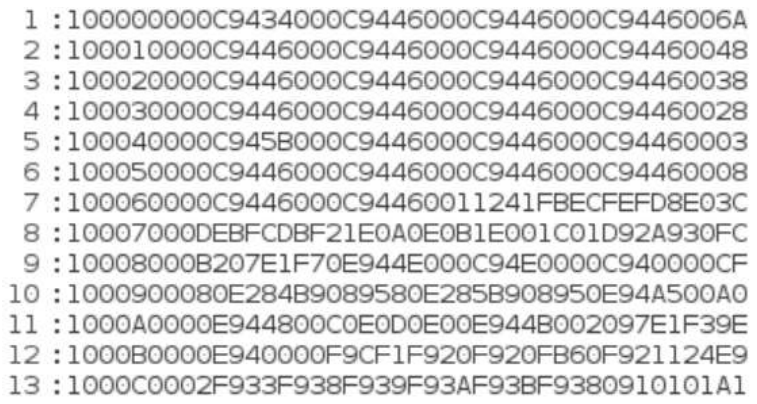
WHY TO CODE THE AVR IN C-LANGUAGE

- Easier and less time consuming to write in C than Assembly
- C is easier to modify and update
- Available state-of-the-art libraries
- Portable

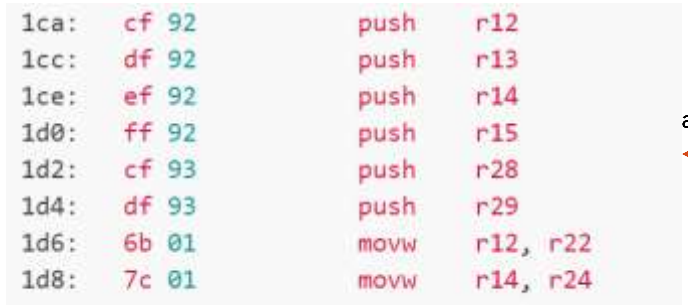
But, what about the HEX Files ?

FROM HEX TO MACHINE CODE

inside of an hex file



```
:100000000C9434000C9446000C9446000C9446006A
:100010000C9446000C9446000C9446000C94460048
:100020000C9446000C9446000C9446000C94460038
:100030000C9446000C9446000C9446000C94460028
:100040000C945B000C9446000C9446000C94460003
:100050000C9446000C9446000C9446000C94460008
:100060000C9446000C94460011241FBECFEFD8E03C
:10007000DEBFCDBF21E0A0E0B1E001C01D92A930FC
:10008000B207E1F70E944E000C94E0000C940000CF
:1000900080E284B9089580E285B908950E94A500A0
:1000A0000E944800C0E0D0E00E944B002097E1F39E
:1000B0000E940000F9CF1F920F920FB60F921124E9
:1000C0002F933F938F939F93AF93BF9380910101A1
```



```
1ca:	cf 92      	push	r12
1cc:	df 92      	push	r13
1ce:	ef 92      	push	r14
1d0:	ff 92      	push	r15
1d2:	cf 93      	push	r28
1d4:	df 93      	push	r29
1d6:	6b 01      	movw	r12, r22
1d8:	7c 01      	movw	r14, r24
```

avr-objdump

- :107E000011E0A0E0B1E0E0E1F0E802C005900D92E1
- :[10] [7E00] [00] 11E0A0E0B1E0E0E1F0E802C005900D92 [E1]

- ":" indicates the start of the record
- [10] → 0x10 indicating the number of bytes for the record.
- [7E00] or 0x7E00 → Starting address
- [00] → data (type).
- [E1] is the checksum
- The string in the middle → the actual data that gets programmed into the flash memory…
  - 0x7E00 = 0x11
  - 0x7E01 = 0xE0
  - 0x7E02 = 0xA0
  - 0x7E03 = 0xE0

  …

  …
- Atmega328p: address range 0x0000-0x3FFF
- 0x7E00/2 = 0x3F00
- AVR is 'little endian' architecture. Instruction width is 16-bits
  - 3F00 = 0xE011
  - 3F01 = 0xE0A0

inc r22

0x9563

1001010101100011

RECALL !

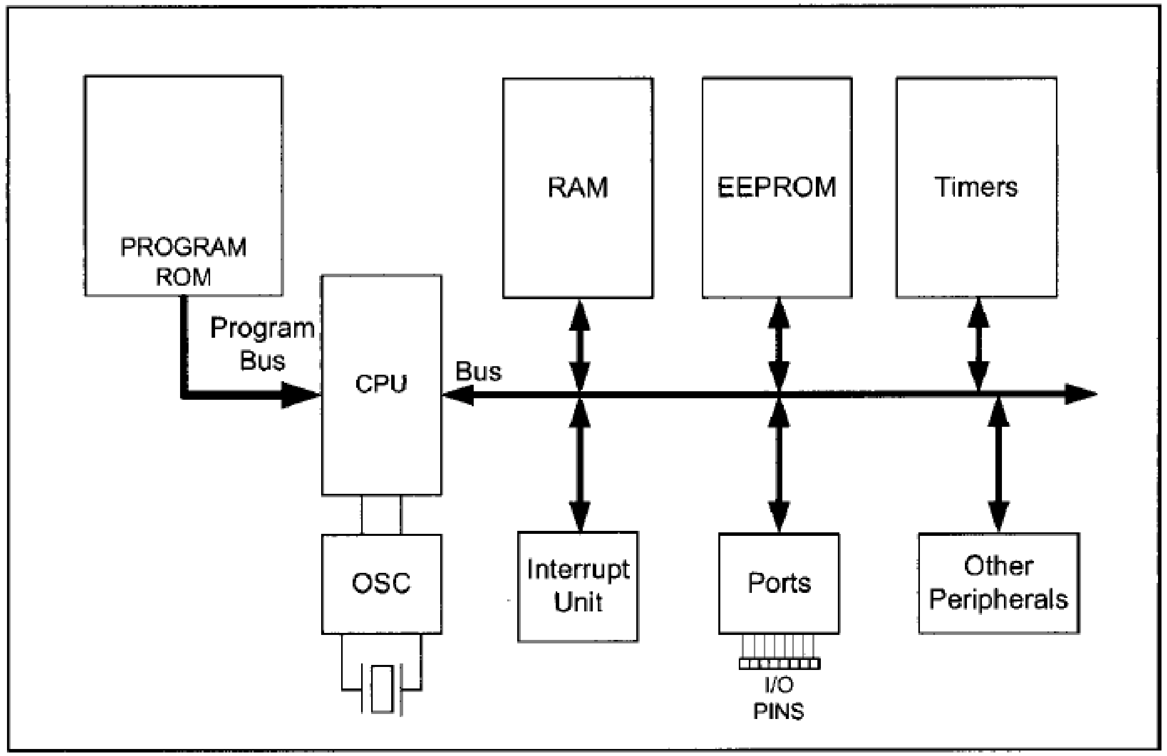


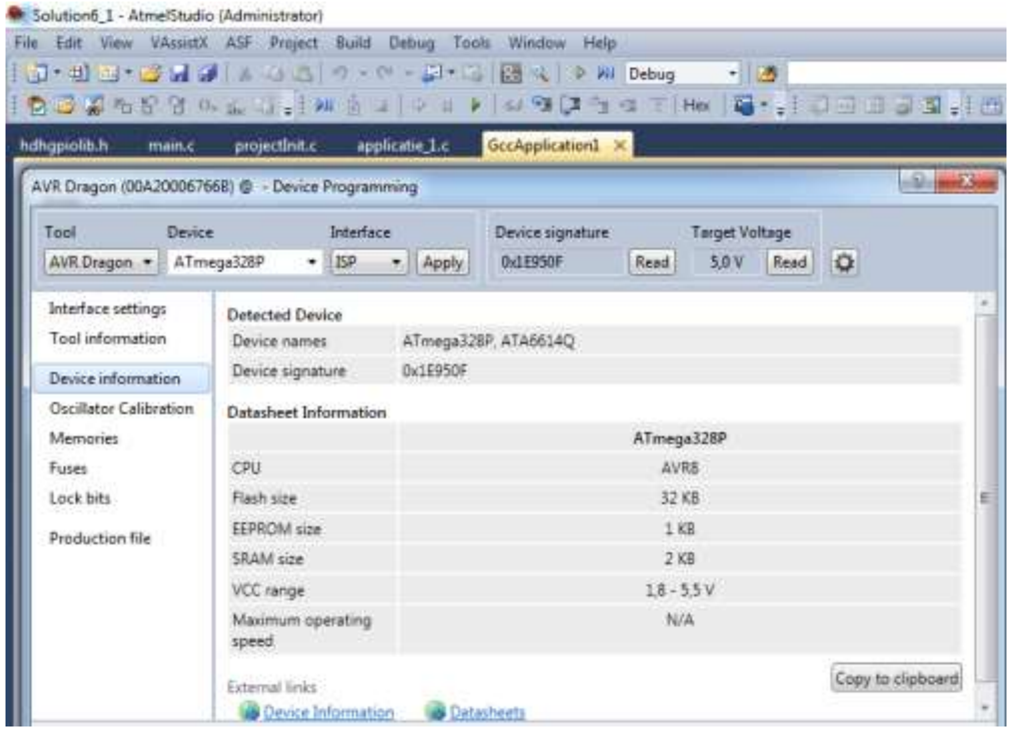
Figure 1-2. Simplified View of an AVR Microcontroller

- Databus : 8-bit wide
- Program/instruction bus : 16-bits wide
- ALU : processes 8-bits data

WHICH COMPILER ?

- Atmel Studio

- Arduino IDE


Solution6_1 - AtmelStudio (Administrator)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Hex
hdhgpiolib.h
main.c
projectInit.c
applicatie_1.c
GccApplication1

AVR Dragon (00A200067668) - Device Programming

Tool	Device	Interface		Device signature		Target Voltage	
AVR Dragon	ATmega328P	ISP	Apply	0x1E950F	Read	5,0 V	Read

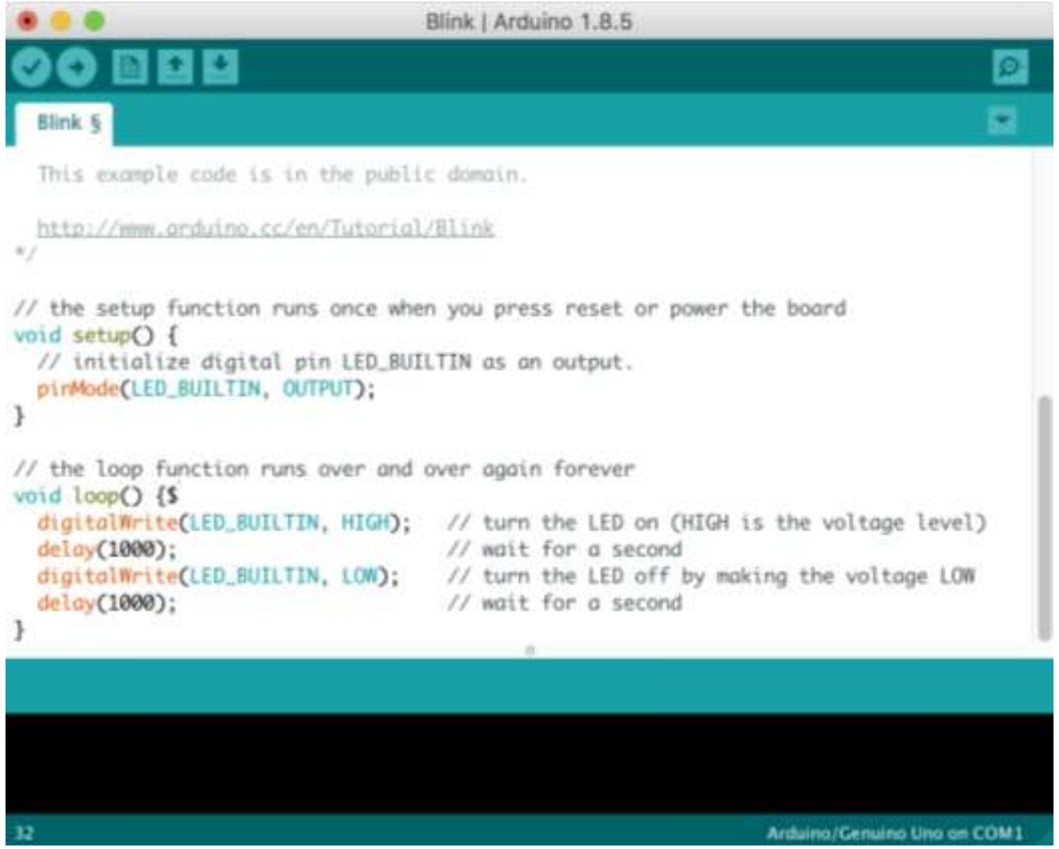
Detected Device

Device names	ATmega328P, ATA6614Q
Device signature	0x1E950F

Datasheet Information

	ATmega328P
CPU	AVR8
Flash size	32 KB
EEPROM size	1 KB
SRAM size	2 KB
VCC range	1,8 - 5,5 V
Maximum operating speed	N/A

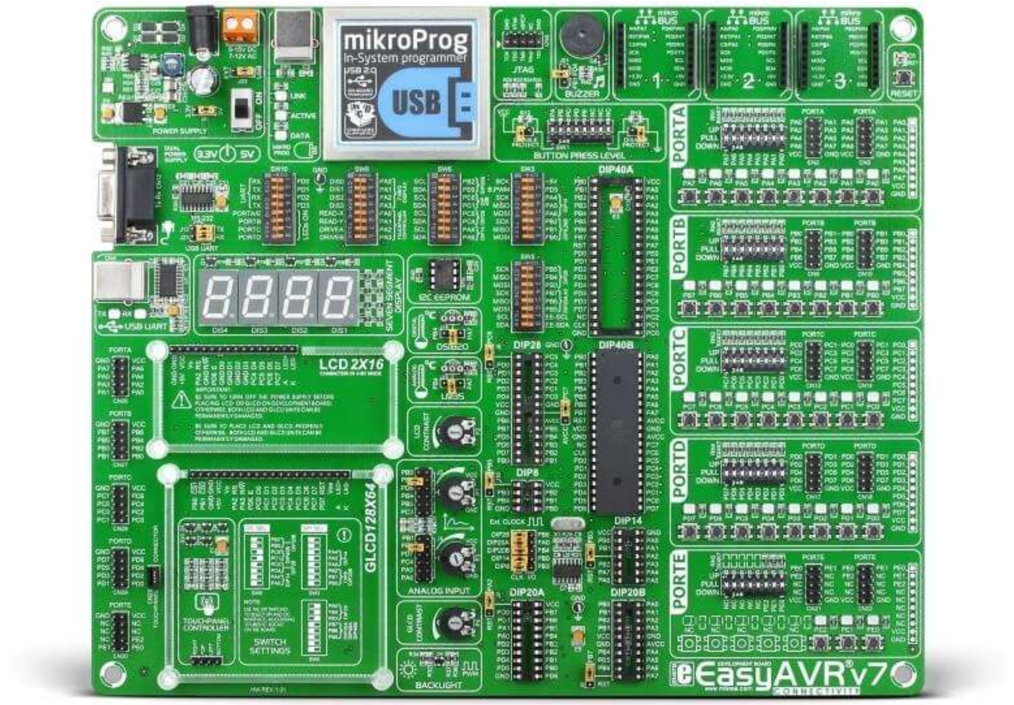
External links
Device Information
Datasheets



```
  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
}
```

WHICH BOARD ?

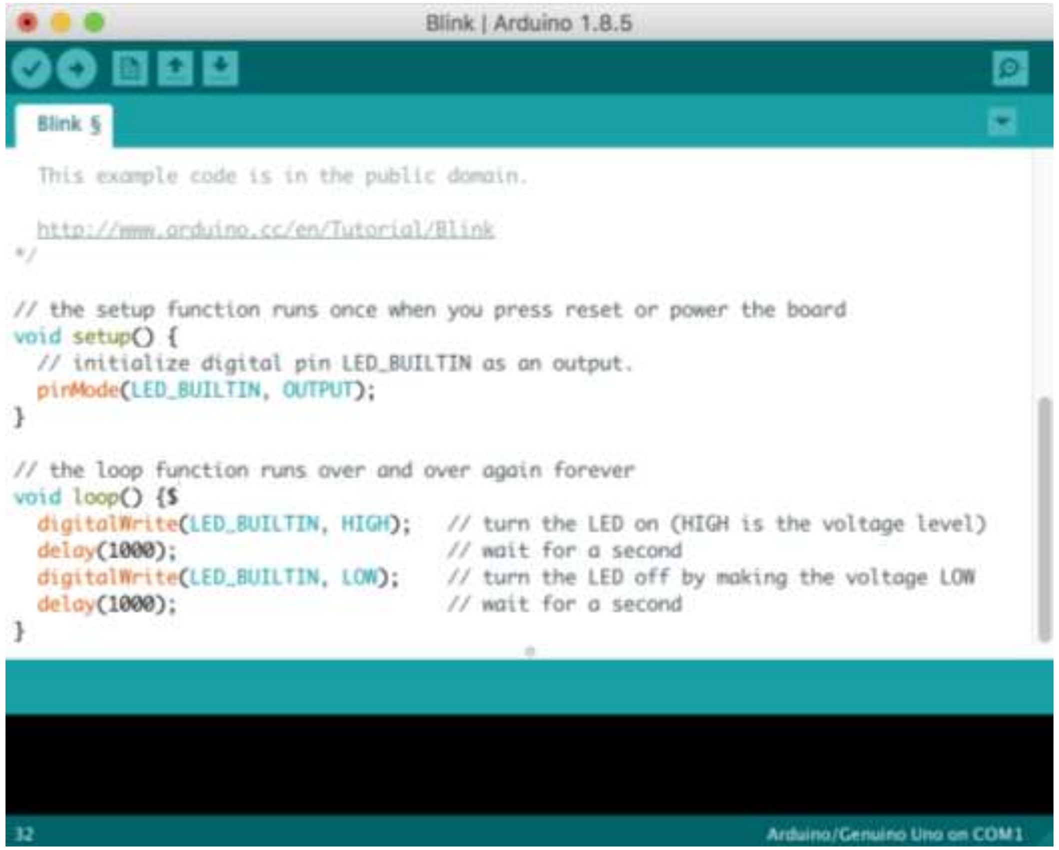
The EasyAVR v7



Arduino Uno

https://wonderfulengineering.com/10-best-avr-development-boards-for-engineers-and-hobbyists/

WE WILL USE

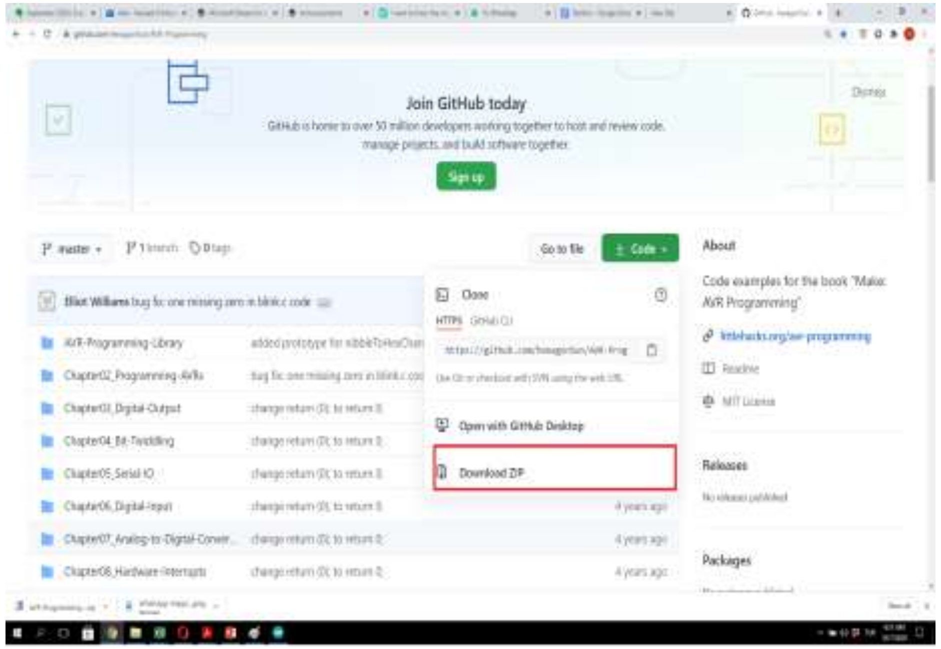


```
  This example code is in the public domain.

  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
}
```

```
32
```

HOW TO WRITE C CODE ON ARDUINO IDE

- C library for AVR microcontroller.
  - Go to https://github.com/hexagon5un/AVR-Programming
  - Download the code.
  - Unzip the dowloaded folder and copy «AVR-Programming-Library» folder to
  - «Documents\Arduino\libraries» on Windows.


Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Elliot Williams bug fix: one missing zero in blink.c code

AVR-Programming-Library	added prototype for nibbleToHexChar	
Chapter02_Programming-AVRs	bug fix: one missing zero in blink.c code	
Chapter03_Digital-Output	change return (0); to return 0;	
Chapter04_Bit-Twiddling	change return (0); to return 0;	
Chapter05_Serial-IO	change return (0); to return 0;	
Chapter06_Digital-Input	change return (0); to return 0;	4 years ago
Chapter07_Analog-to-Digital-Conver...	change return (0); to return 0;	4 years ago
Chapter08_Hardware-Interrupts	change return (0); to return 0;	4 years ago

About

Code examples for the book "Make: AVR Programming"

littlehacks.org/avr-programming

Releases

No releases published

Packages

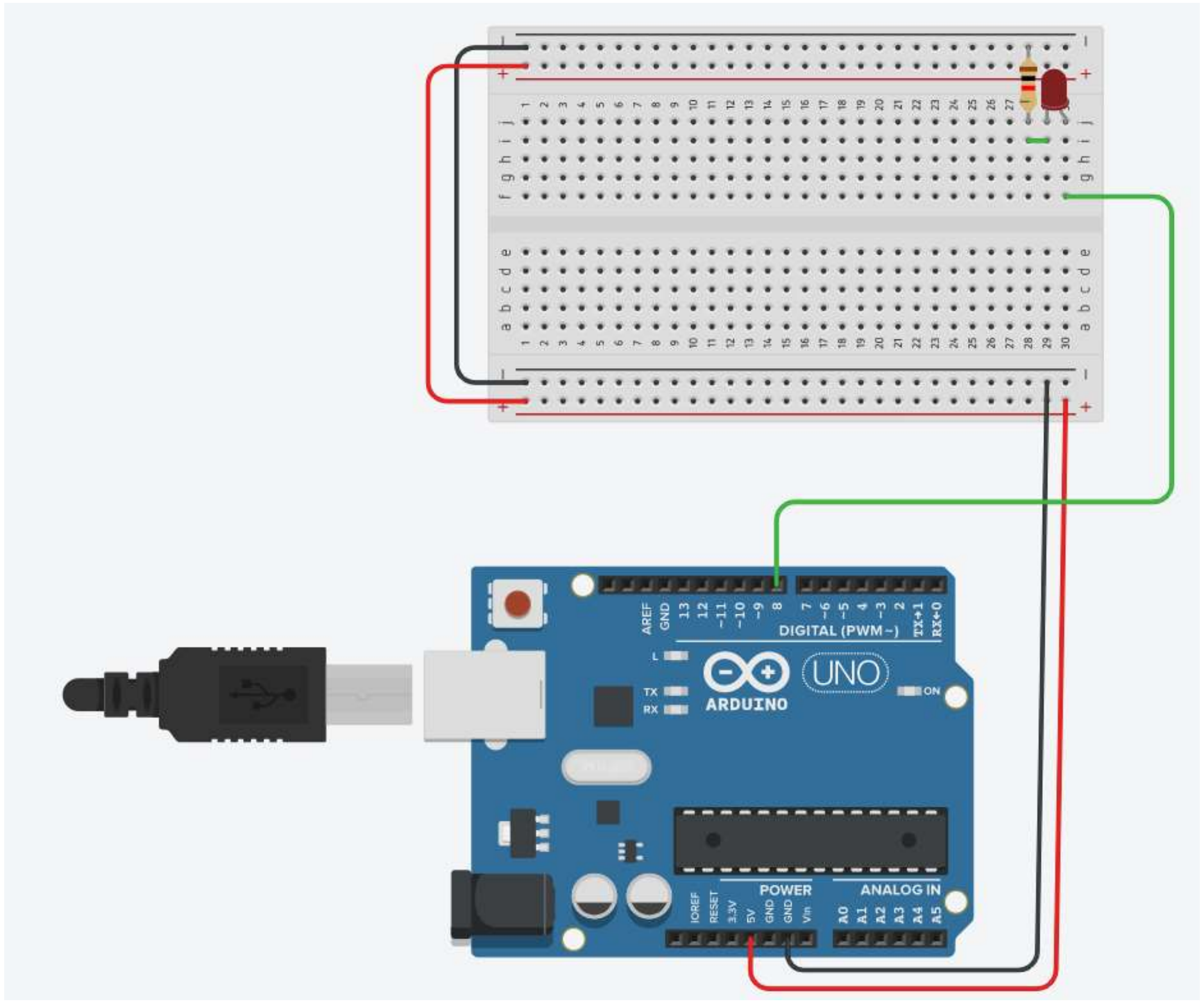
HOW TO WRITE C CODE ON ARDUINO IDE

- Replace portpins.h file in Arduino include folder with the newest version.
  - Copy the «portpins.h» file from downloaded «AVR-Programming-library»
  - Overwrite the already existing portpins.h file setup during installation of Arduino. In my machine, the location of this already existing file is «C:\Program Files (x86)\Arduino\hardware\tools\avr\avr\include\avr»

DEMO: C-PROGRAMMING ON ARDUINO IDE

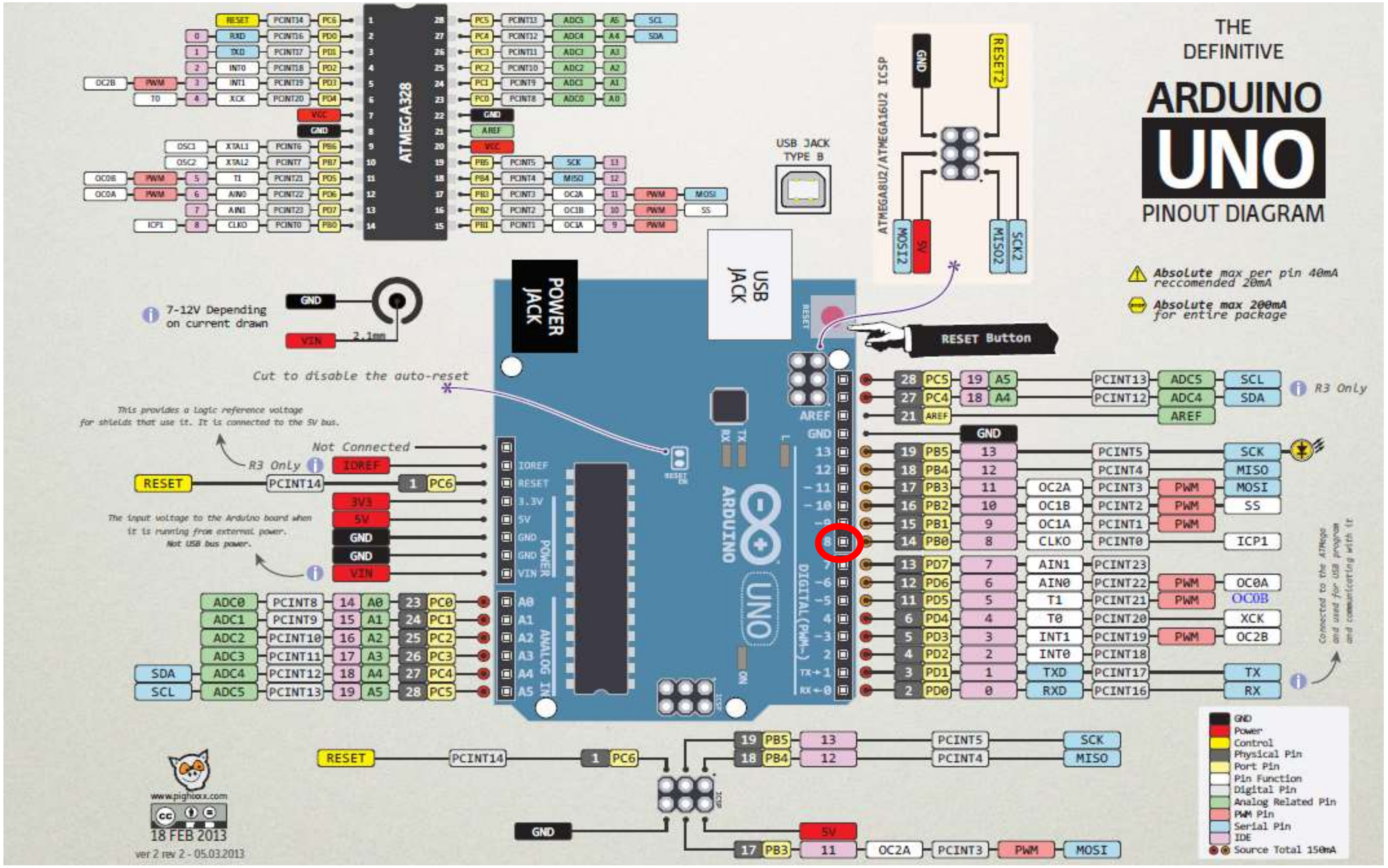
«BLINKER»

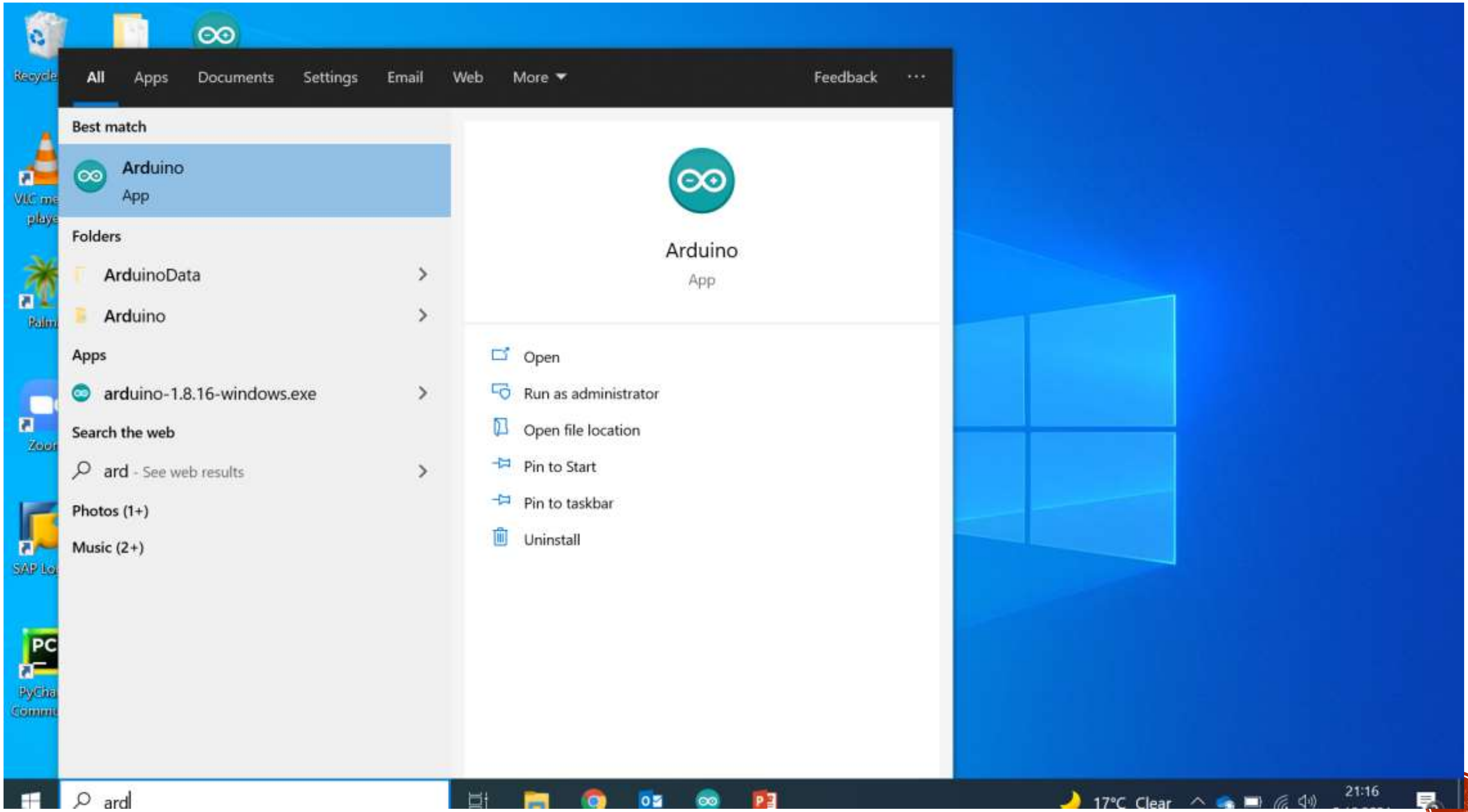
Make connections



Before writing code, always check the pin mapping between Arduino board and AVR microcontroller.

Digital pin 8 of Arduino is pin 0 of PortB on the microcontroller.




All
Apps
Documents
Settings
Email
Web
More
Feedback

Best match
Arduino
App

Folders
ArduinoData
Arduino

Apps
arduino-1.8.16-windows.exe

Search the web
ard - See web results

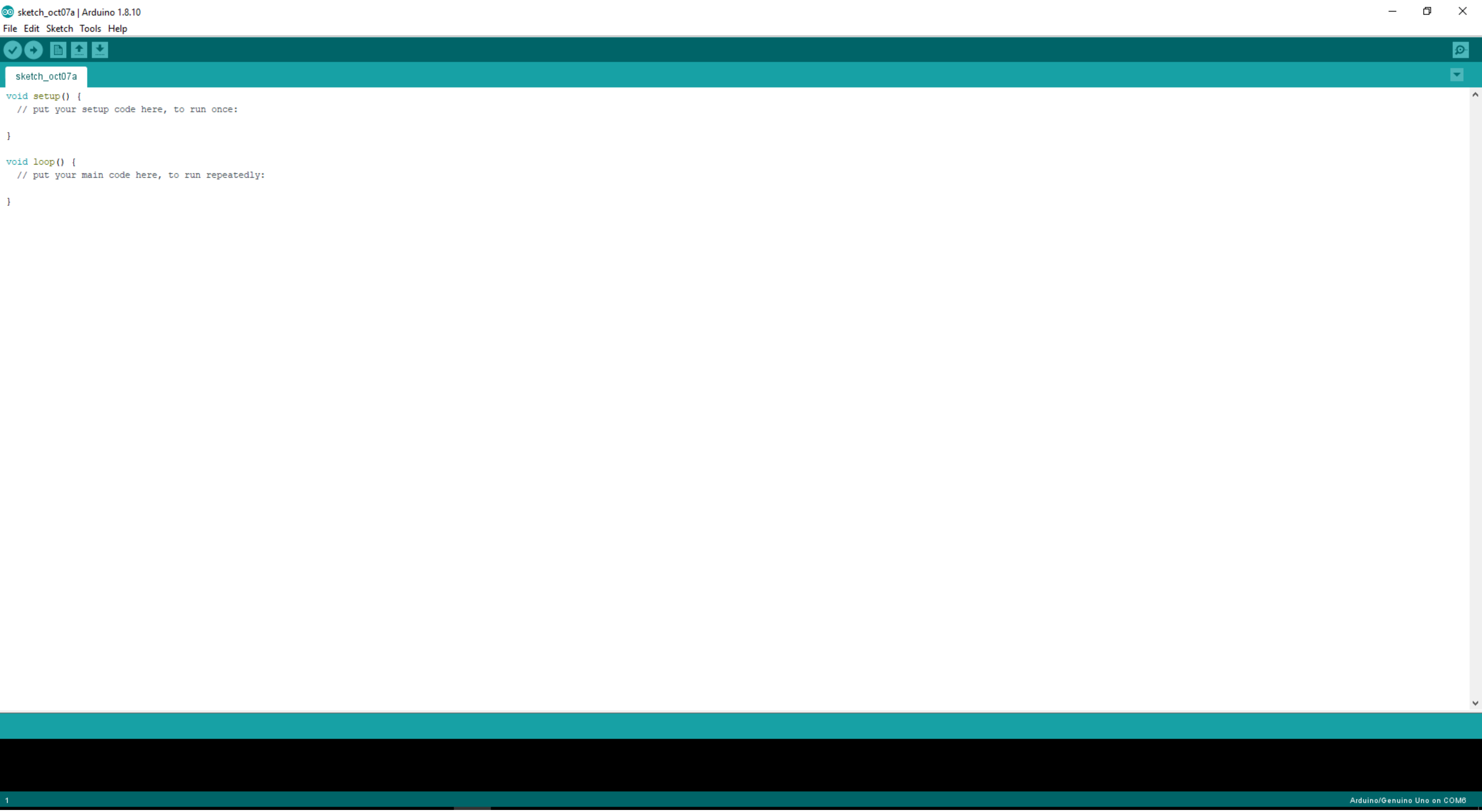
Photos (1+)
Music (2+)

Arduino
App

Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar
Uninstall

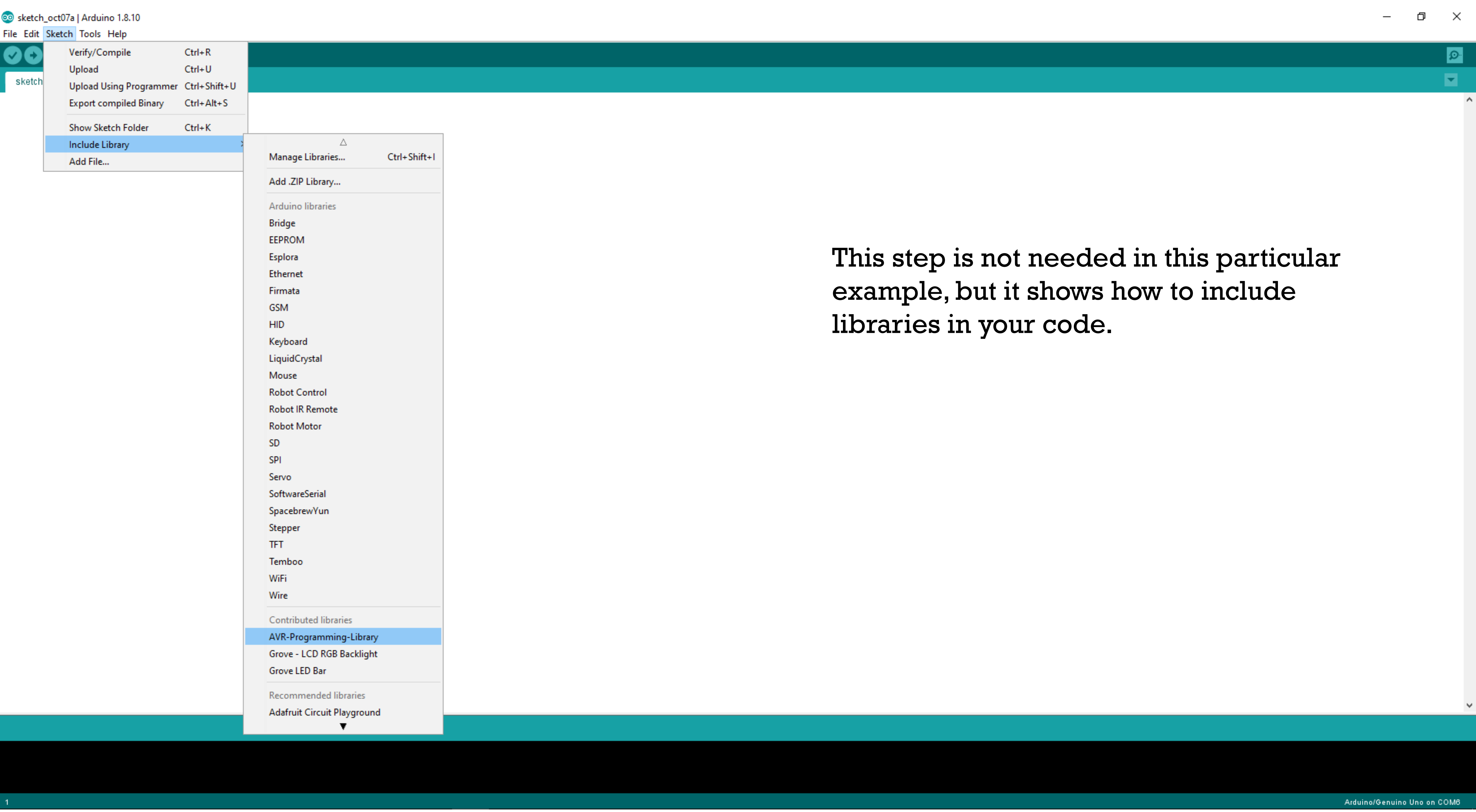
ard

17°C Clear
21:16


sketch_oct07a | Arduino 1.8.10
File Edit Sketch Tools Help
sketch_oct07a

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

}

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

}
```

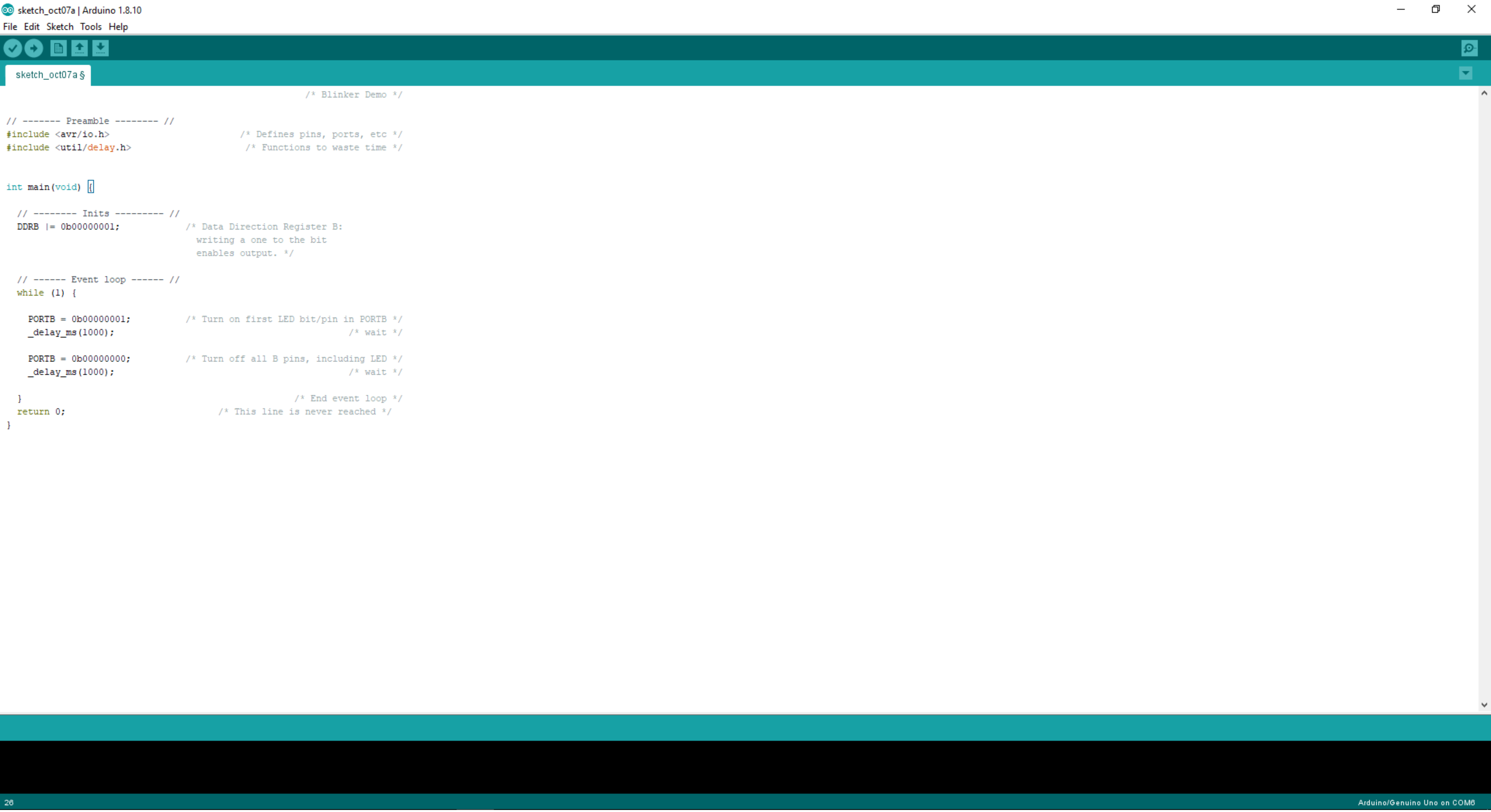
1
Arduino/Genuino Uno on COM6



This step is not needed in this particular example, but it shows how to include libraries in your code.

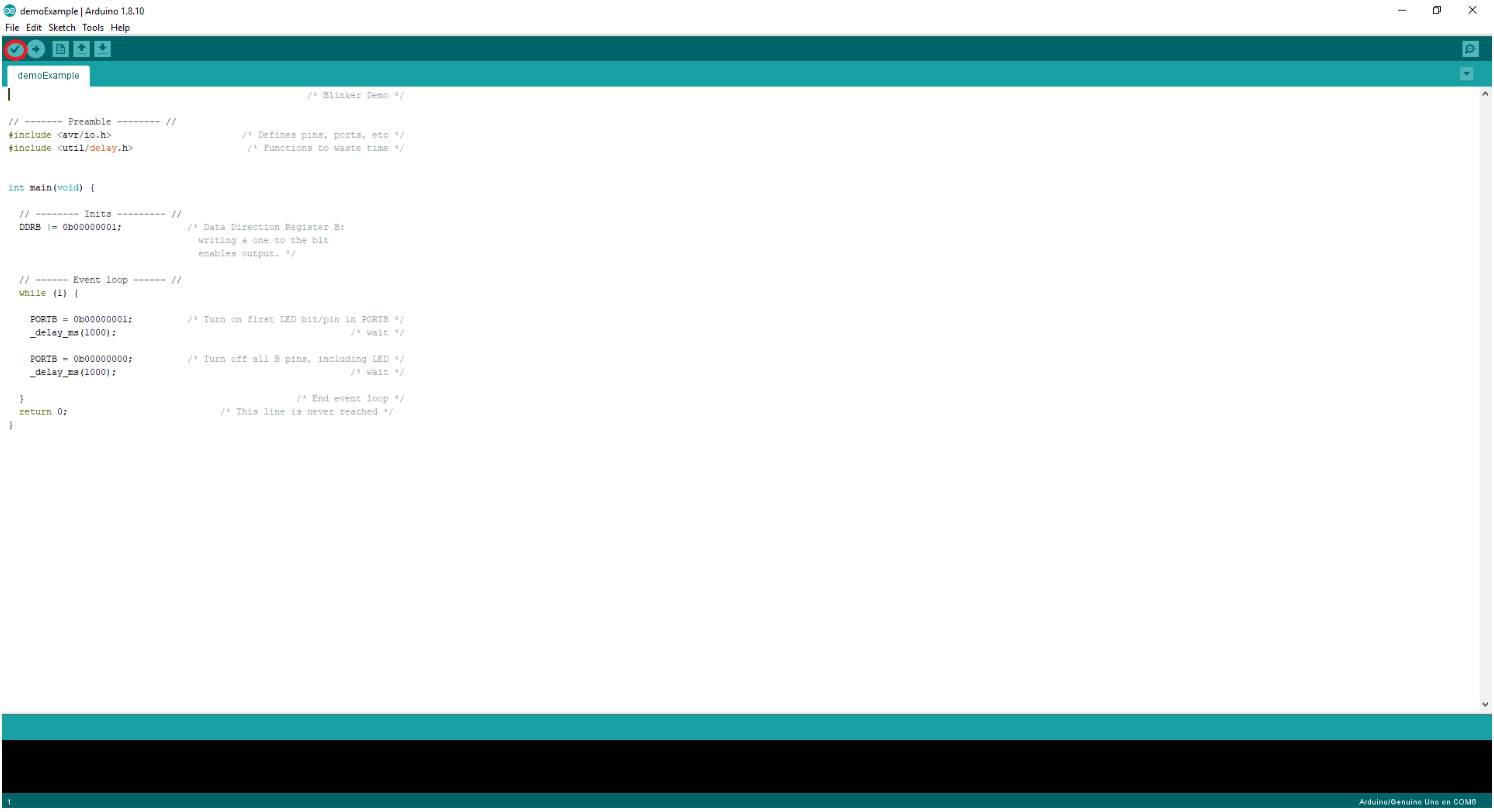
PASTE FOLLOWING CODE TO EMPTY FILE

```
 /* Blinker Demo */
// ------- Preamble -------- //
#include <avr/io.h>                    /* Defines pins, ports, etc */
#include <util/delay.h>                  /* Functions to waste time */
int main(void) {
// -------- Inits --------- //
DDRB |= 0b00000001;          /* Data Direction Register B:  writing a one to the bit  enables output. */
// ------ Event loop ------ //
while (1) {
            PORTB = 0b00000001;         /* Turn on only pin 0 of PORTB */
            _delay_ms(1000);                            /* wait */
            PORTB = 0b00000000;         /* Turn off all pins on PORT B */
            _delay_ms(1000);                            /* wait */
            }                                    /* End event loop */
return 0;                     /* This line is never reached */
}
```

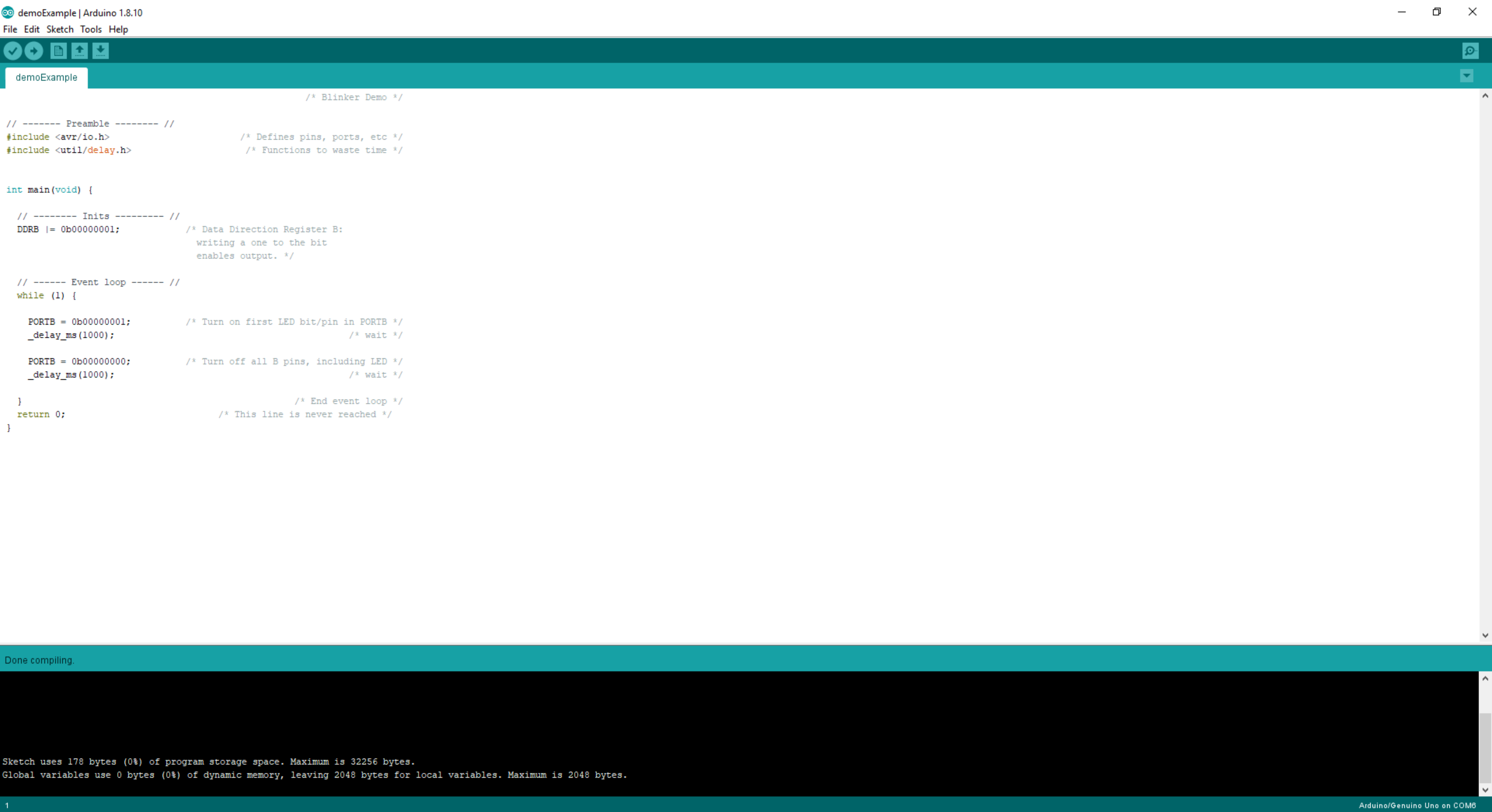

sketch_oct07a | Arduino 1.8.10
File Edit Sketch Tools Help
sketch_oct07a §
26
Arduino/Genuino Uno on COM6

```
                                                       /* Blinker Demo */

// ------- Preamble -------- //
#include <avr/io.h>                        /* Defines pins, ports, etc */
#include <util/delay.h>                     /* Functions to waste time */


int main(void) {

  // -------- Inits --------- //
  DDRB |= 0b00000001;           /* Data Direction Register B:
                                   writing a one to the bit
                                   enables output. */

  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;          /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                           /* wait */

    PORTB = 0b00000000;          /* Turn off all B pins, including LED */
    _delay_ms(1000);                                           /* wait */

  }                                                  /* End event loop */
  return 0;                            /* This line is never reached */
}
```

- Save file with name of «DemoExample», lets say, on Desktop.
- A folder with name of «DemoExample» will be created containing the arduino file named «DemoExample»



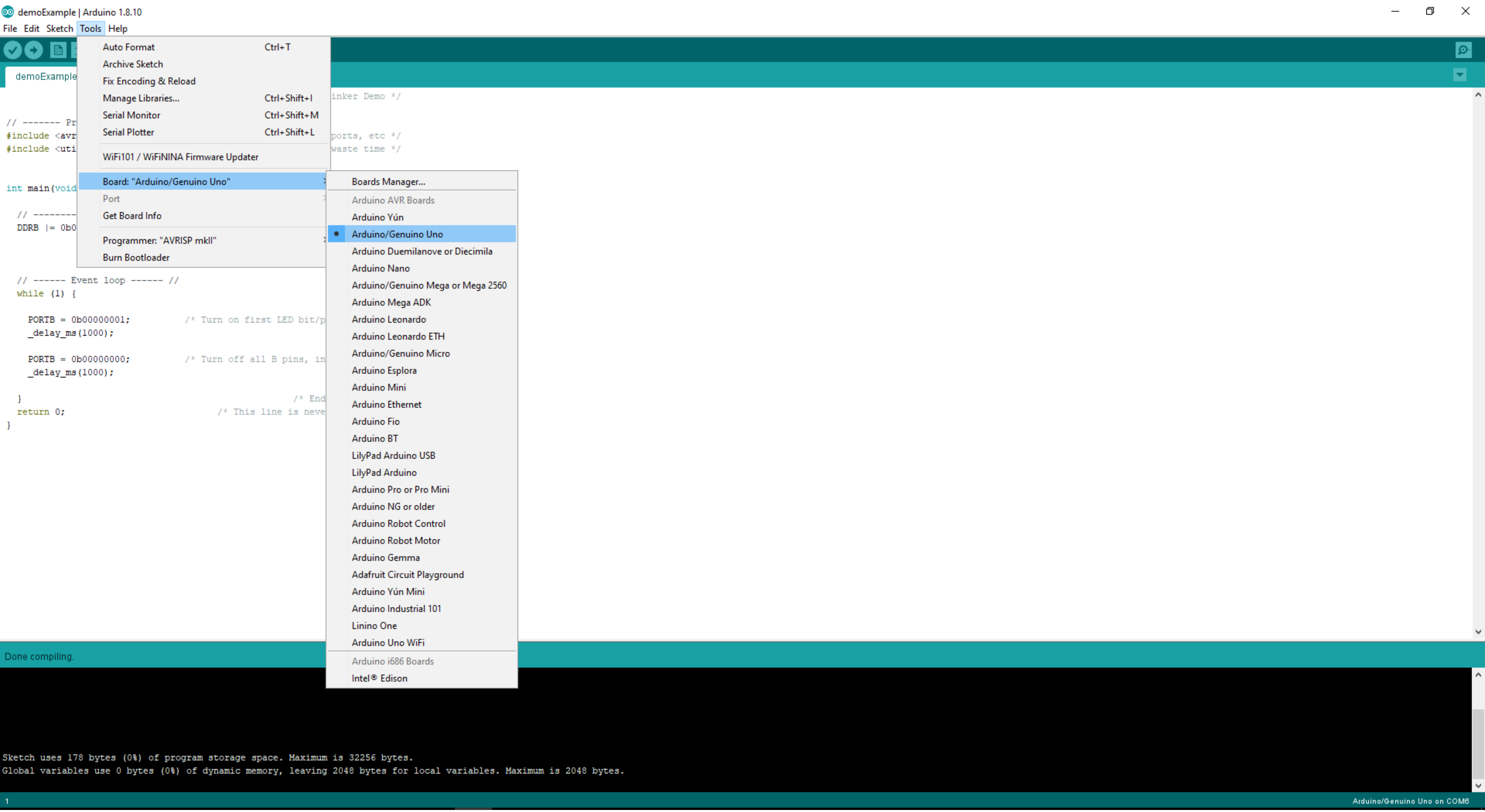
```
                                                        /* Blinker Demo */

// ------- Preamble -------- //
#include <avr/io.h>                         /* Defines pins, ports, etc */
#include <util/delay.h>                      /* Functions to waste time */


int main(void) {

  // -------- Inits --------- //
  DDRB |= 0b00000001;             /* Data Direction Register B:
                                    writing a one to the bit
                                    enables output. */

  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;           /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                             /* wait */

    PORTB = 0b00000000;           /* Turn off all B pins, including LED */
    _delay_ms(1000);                                             /* wait */

  }                                                  /* End event loop */
  return 0;                             /* This line is never reached */
}
```



```
                                                         /* Blinker Demo */

// ------- Preamble -------- //
#include <avr/io.h>                            /* Defines pins, ports, etc */
#include <util/delay.h>                         /* Functions to waste time */


int main(void) {

  // -------- Inits --------- //
  DDRB |= 0b00000001;            /* Data Direction Register B:
                                   writing a one to the bit
                                   enables output. */

  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;          /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                             /* wait */

    PORTB = 0b00000000;          /* Turn off all B pins, including LED */
    _delay_ms(1000);                                             /* wait */

  }                                                  /* End event loop */
  return 0;                            /* This line is never reached */
}
```

Done compiling.

```
Sketch uses 178 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```



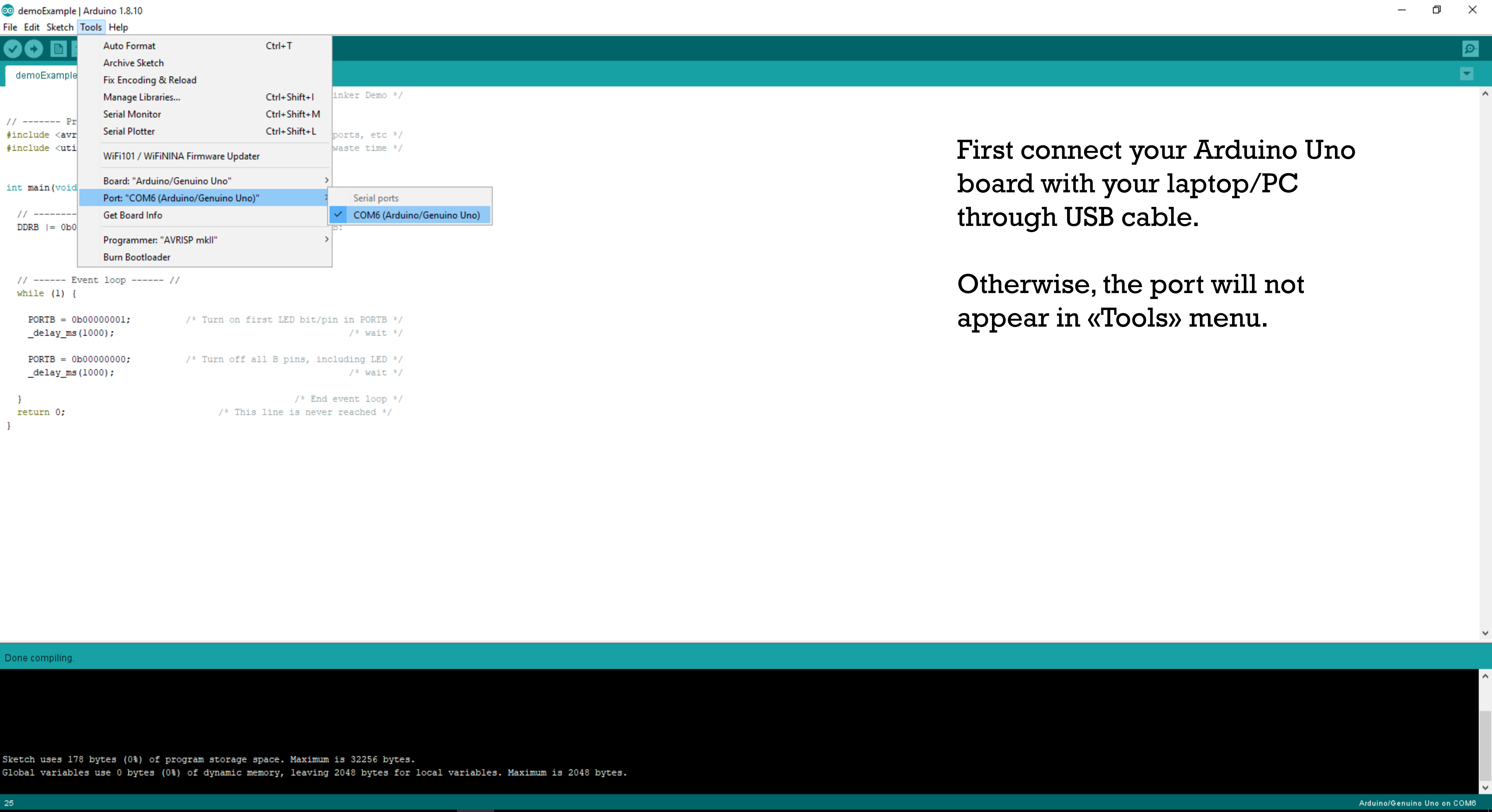
```
                                                             inker Demo */

// ------- Pr
#include <avr                                                ports, etc */
#include <uti                                                waste time */

int main(void

  // --------
  DDRB |= 0b0


  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;          /* Turn on first LED bit/p
    _delay_ms(1000);

    PORTB = 0b00000000;          /* Turn off all B pins, in
    _delay_ms(1000);

  }                                                   /* End
  return 0;                            /* This line is neve
}
```

Done compiling.

```
Sketch uses 178 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```

1

Arduino/Genuino Uno on COM6

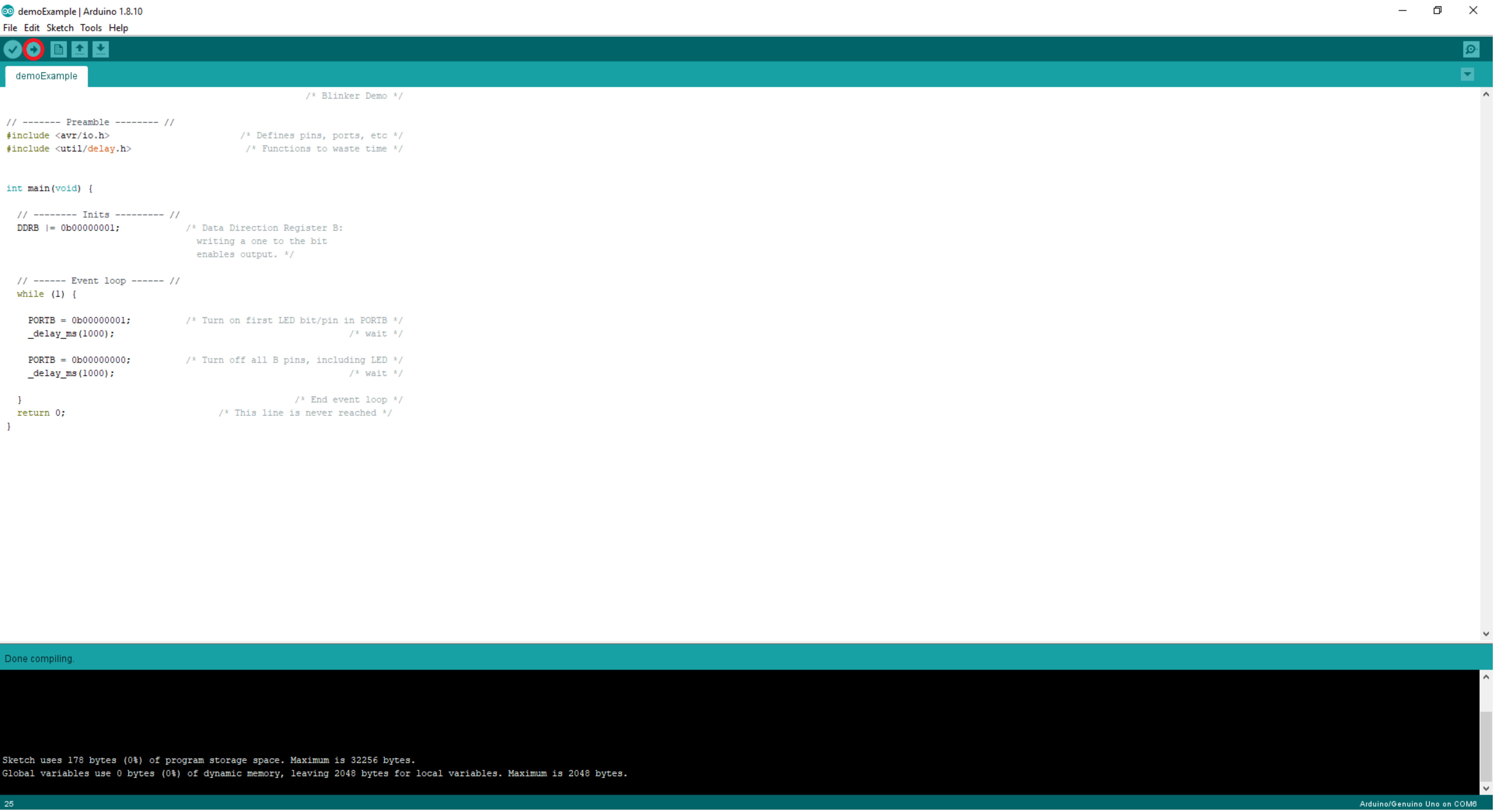

demoExample | Arduino 1.8.10
File Edit Sketch Tools Help
demoExample
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM6 (Arduino/Genuino Uno)"
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Serial ports
✓ COM6 (Arduino/Genuino Uno)
Done compiling.
25
Arduino/Genuino Uno on COM6

```
                                              inker Demo */

// ------- Pr
#include <avr                                 ports, etc */
#include <uti                                 waste time */


int main(void

  // --------
  DDRB  |= 0b0                                B:


  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;         /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                           /* wait */

    PORTB = 0b00000000;         /* Turn off all B pins, including LED */
    _delay_ms(1000);                                           /* wait */

  }                                                 /* End event loop */
  return 0;                               /* This line is never reached */
}
```

```
Sketch uses 178 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```

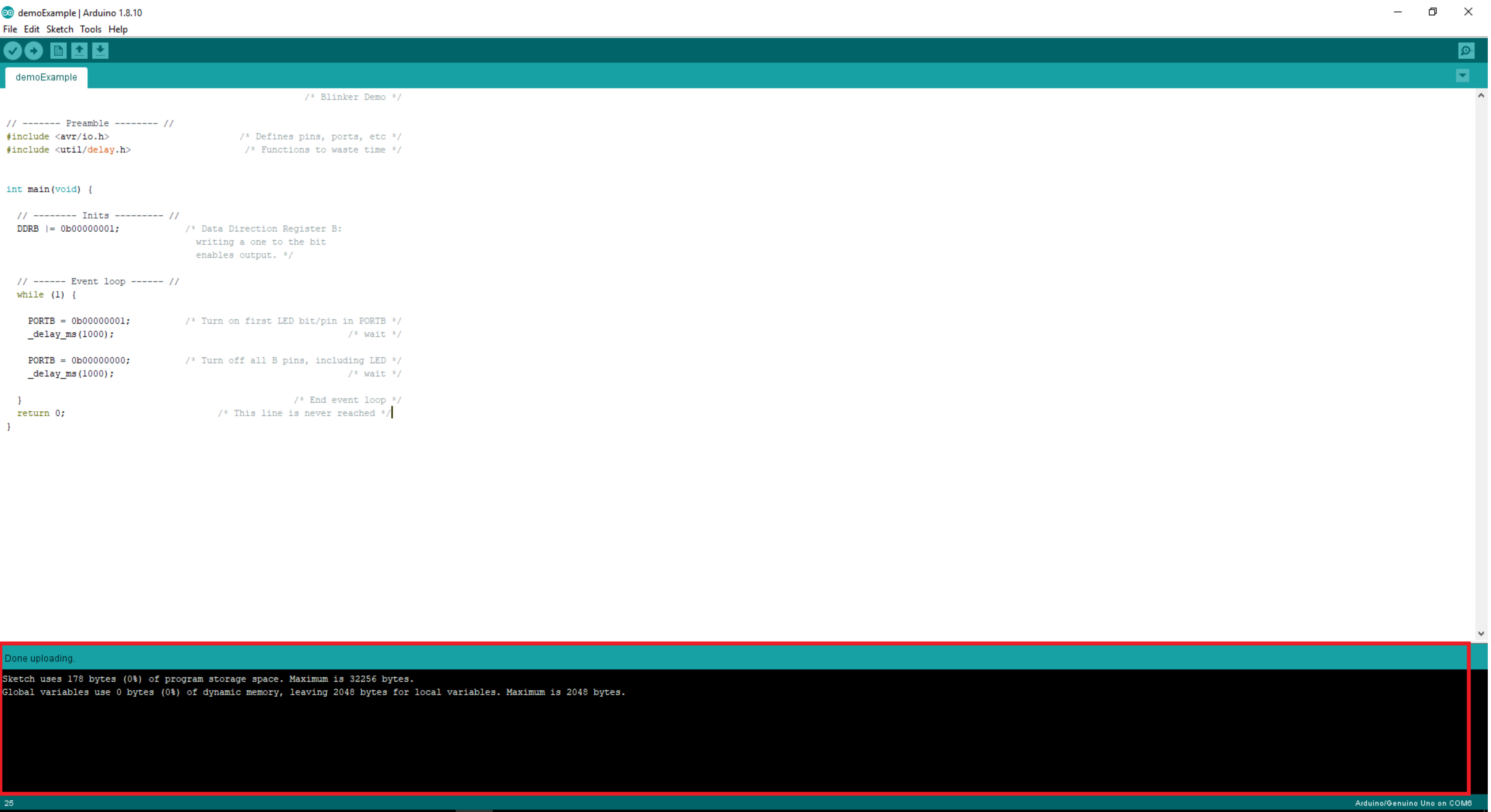
First connect your Arduino Uno board with your laptop/PC through USB cable.

Otherwise, the port will not appear in «Tools» menu.


demoExample | Arduino 1.8.10
File Edit Sketch Tools Help
demoExample
Done compiling.
25
Arduino/Genuino Uno on COM6

```
                                                        /* Blinker Demo */

// ------- Preamble -------- //
#include <avr/io.h>                          /* Defines pins, ports, etc */
#include <util/delay.h>                       /* Functions to waste time */


int main(void) {

  // -------- Inits --------- //
  DDRB |= 0b00000001;            /* Data Direction Register B:
                                    writing a one to the bit
                                    enables output. */

  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;          /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                             /* wait */

    PORTB = 0b00000000;          /* Turn off all B pins, including LED */
    _delay_ms(1000);                                             /* wait */

  }                                                    /* End event loop */
  return 0;                            /* This line is never reached */
}
```

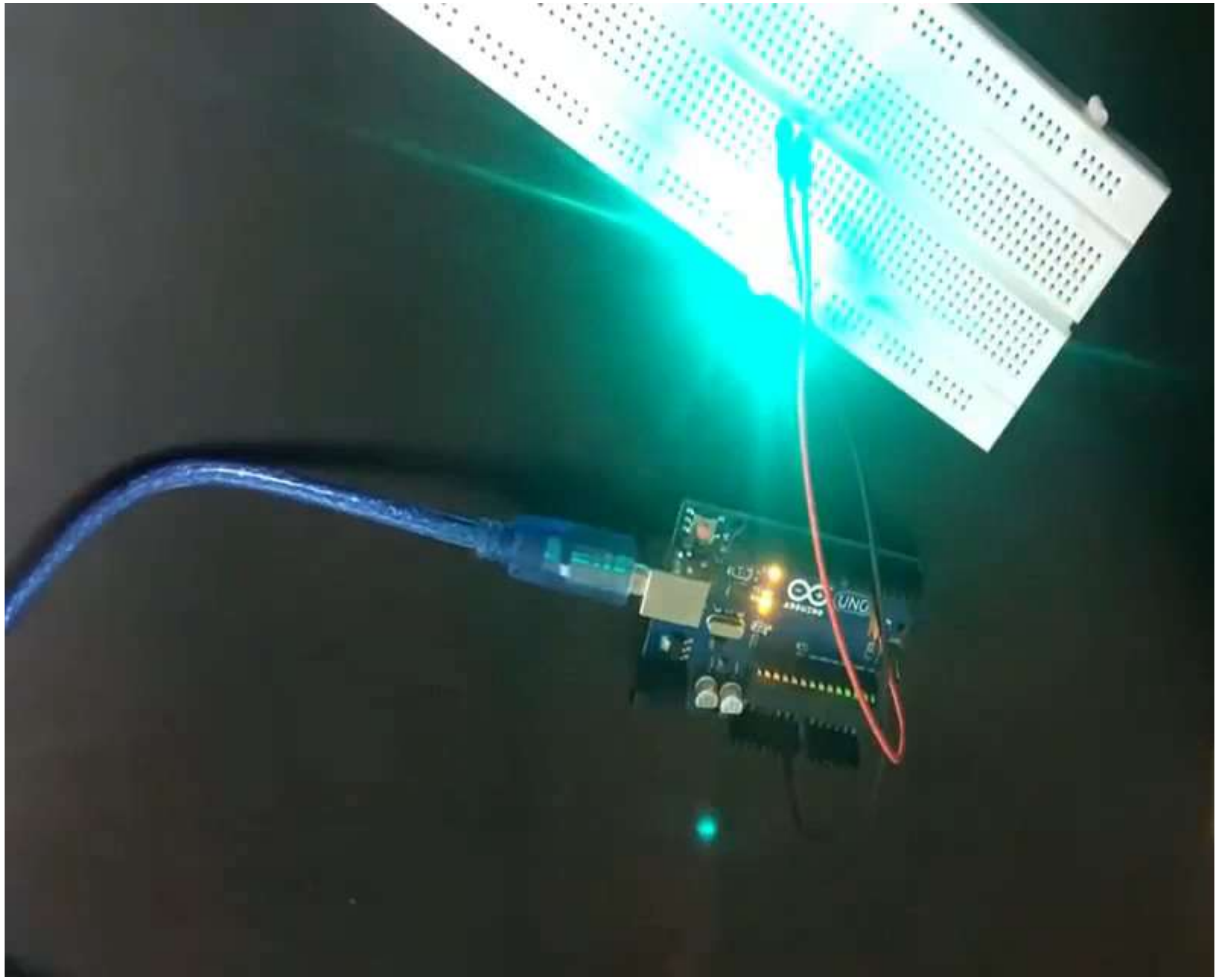
```
Sketch uses 178 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```



```
                                                        /* Blinker Demo */

// ------- Preamble -------- //
#include <avr/io.h>                         /* Defines pins, ports, etc */
#include <util/delay.h>                      /* Functions to waste time */


int main(void) {

  // -------- Inits --------- //
  DDRB |= 0b00000001;            /* Data Direction Register B:
                                   writing a one to the bit
                                   enables output. */

  // ------ Event loop ------ //
  while (1) {

    PORTB = 0b00000001;          /* Turn on first LED bit/pin in PORTB */
    _delay_ms(1000);                                          /* wait */

    PORTB = 0b00000000;          /* Turn off all B pins, including LED */
    _delay_ms(1000);                                          /* wait */

  }                                                /* End event loop */
  return 0;                            /* This line is never reached */
}
```

Done uploading.

```
Sketch uses 178 bytes (0%) of program storage space. Maximum is 32256 bytes.
Global variables use 0 bytes (0%) of dynamic memory, leaving 2048 bytes for local variables. Maximum is 2048 bytes.
```


UNO