EEE1002 LABORATORY PROJECT

Project Title: Quiz Buzzer

Name-Surname

Ömer-Ayar

Student ID

2101069

2022

Materials:

1. Arduino Uno
2. Button
3. Jumper Cable
4. 7 Segment Display
5. 10kΩ resistors

Definition of the Project:

It is designed to tell which competitor should answer first with the help of buttons in any competition.

What is Arduino?

The best thing about the Arduino prototyping platform is that it is what you want. The Arduino could be an automatic control system for plant irrigation. It can be a web server. It could even be a quadcopter autopilot. Arduino is a microcontroller development platform combined with an intuitive programming language that is developed using the Arduino Integrated Development Environment (IDE). By equipping Arduino with sensors, actuators, lights, speakers, add-on modules (called shields), and other integrated circuits, you can turn Arduino into a programmable "brain" for any operating system. cover everything Arduino is capable of as the possibilities are limited only by your imagination.

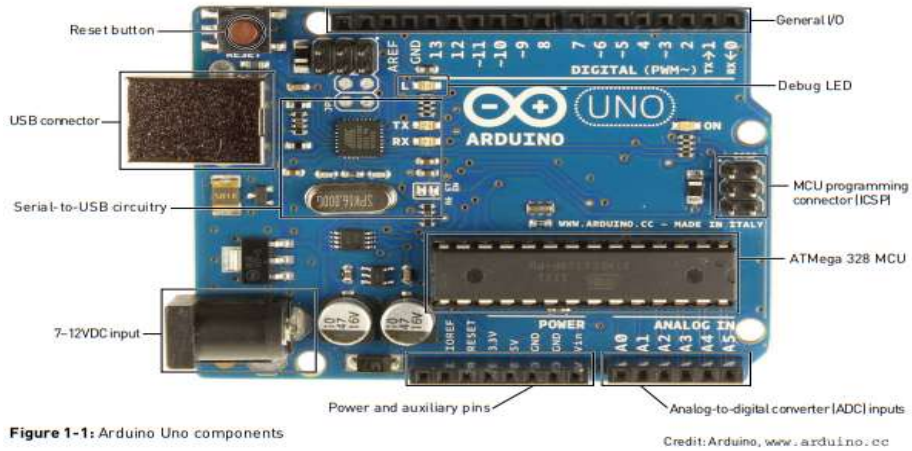


Figure 1-1: Arduino Uno components

Credit: Arduino, www.arduino.cc

Atmel Microcontroller

The heart of every Arduino is the Atmel Controller Unit (MCU). Most Arduino boards, including the Arduino Uno, use the AVRAT Mega microcontroller. Arduino Uno in Figure 1-1 uses the ATMega328p. Prevention exception; uses the ARM Cortex microcontroller. This microcontroller is responsible for keeping all compiled code and executing specified commands. The Arduino programming language provides access to microcontroller peripherals such as analog-to-digital converters (ADCs), common input/output (I / O) pins, and communication

buses (including I2C and SPI), and serial interfaces. All these useful functions fall under the headers accessible to Arduino females which can be wired or shielded by the tiny pins of the microcontroller. A 16 MHz ceramic resonator is connected to the ATMega clock pin. The ATMega clock pin serves as a reference for executing all program commands. You can use the reset button to restart the software. Most Arduino boards come with a debug LED already connected to pin 13. This allows you to run the first program without connecting any additional circuitry (the LED flashes).

Programming Interfaces

Typically, ATMega microcontroller programs are written in C or assembler and programmed through the ICSP interface with a dedicated programmer (see Figure 1-2). Perhaps the most important feature of an Arduino is that you can easily program it via USB without using a separate programmer. This functionality is made possible by the Arduino bootloader. The bootloader is loaded onto the ATMega from the factory (using the ICSP header), allowing a serial USART (Universal Synchronous/Asynchronous Receiver/Transmitter) to load your program onto the ATMega. 'Arduino without using a separate programmer. (You

For more information on how the bootloader works, see the sidebar "The Arduino Bootloader and Firmware Setup".) In the case of the Arduino Uno and Mega 2560, a secondary microcontroller (an ATMega 16U2 or 8U2, depending on the version) serves as an interface between a USB cable and the USART serial pins on the main microcontroller. The Arduino Leonardo, which uses an ATMega 32U4 as the main microcontroller, has an integrated USB port, eliminating the need for a secondary microcontroller. In older Arduino boards, an FTDI-branded USB-Serial chip was used to interface between the ATMega's USART serial port and a USB connection.

General I/O and ADCs

The most interesting parts of the Arduino during the project are the I / O and the common ADC pins. All these pins can be individually processed through the program you create. They all serve as digital inputs and outputs. The ADC pin also acts as an analog input capable of measuring voltages from 0 to 5V (usually from a resistance sensor). Many of these pins have been doubled to provide additional functionality to explore during the project. These special features include various communication interfaces, serial interfaces, PWM outputs, and external interrupts.

Power Supplies

Most designs only use a 5V power supply. It comes via a USB cable. However, when you're ready to disconnect your project from your computer, there are other power options. The Arduino can accept 6V-20V (7-12V recommended) via a DC (DC) barrel jack connector or with a Vin pin. Arduino has built-in 5V and 3.3V regulators.

■ 5V is used for all logic on the board. In other words, when you change

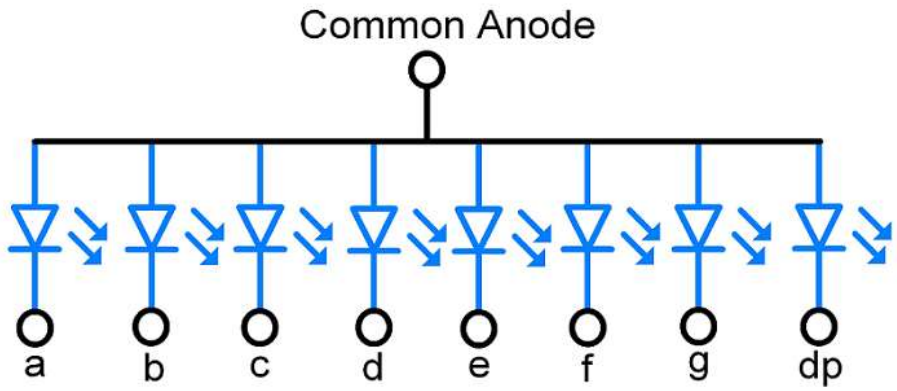
Digital I / O pin. Switch from 5V to 0V.

■ 3.3V is split into pins to accommodate 3.3V and external shieldsCircle.

What is a 7 Segment Display ?

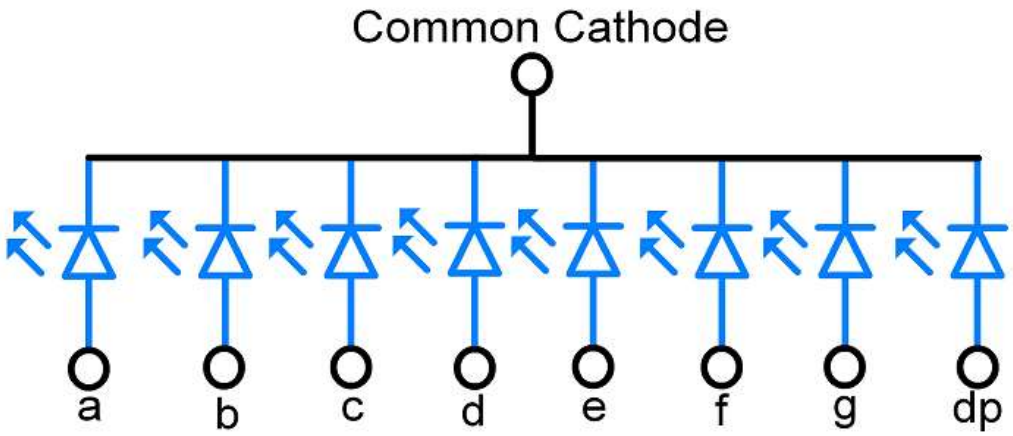
The 7-segment LED displays consist of LED segments. It is basically used to represent numerical values from 0 to 9. There is also a segment which is used as a decimal point.

1. Common Anode (CA)



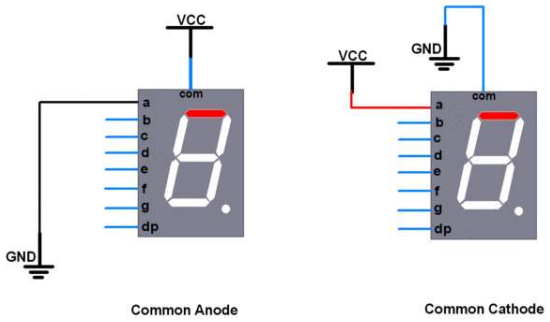
In a common anode display, all the pins of the anode are interconnected by VCC and the LEDs are driven via cathode terminals. It means to turn on the LED (segment), we need to make the cathode pin logic LOW or ground.

2. Common Cathode (CC)



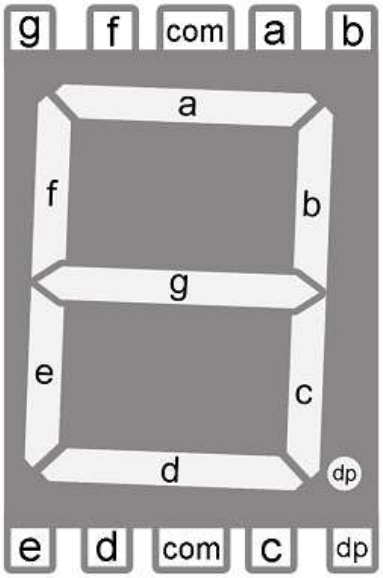
In common cathode display, all cathode pins are connected together and led are controlled via anode terminal. It means to turn ON LED (segment), we have to apply proper voltage to the anode pin.

The following demo shows how the above configuration works.

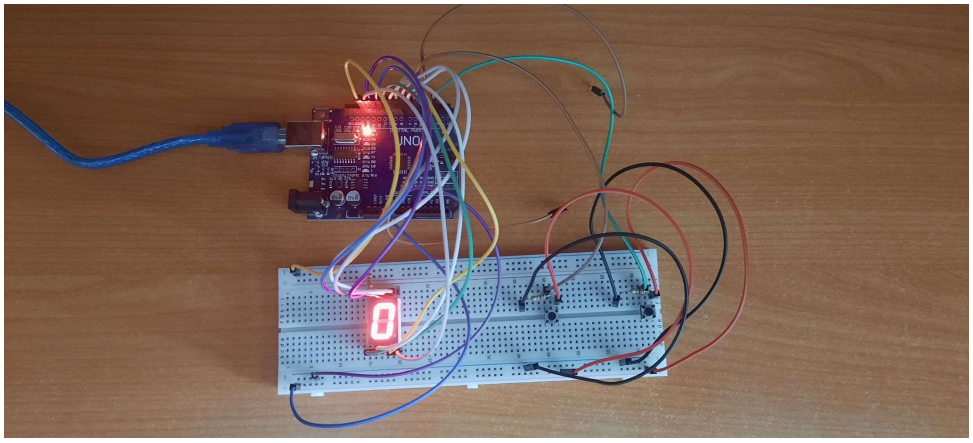


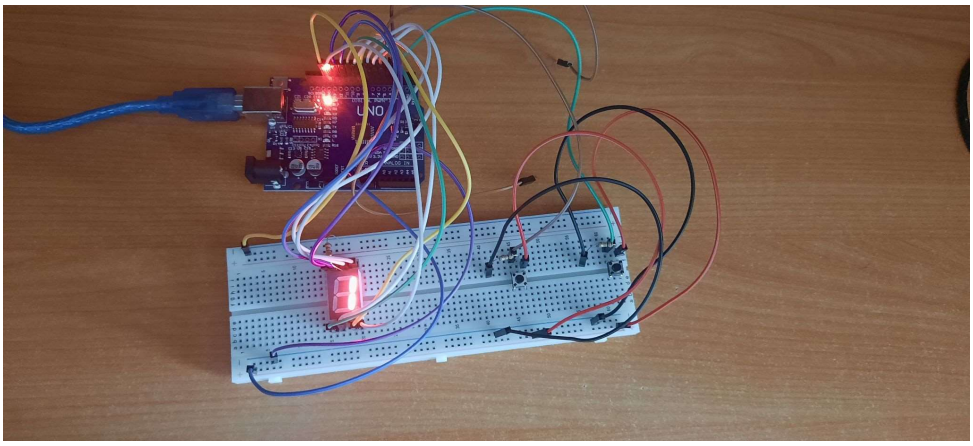
pin diagram

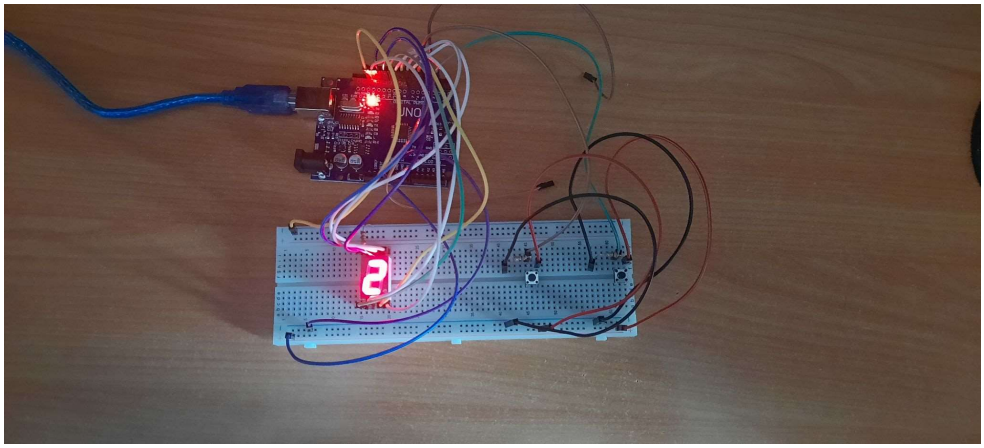
The 7 segment display has 10 pins in total.The common (com) pin is connected in a common anode or common cathode configuration.



Circuit

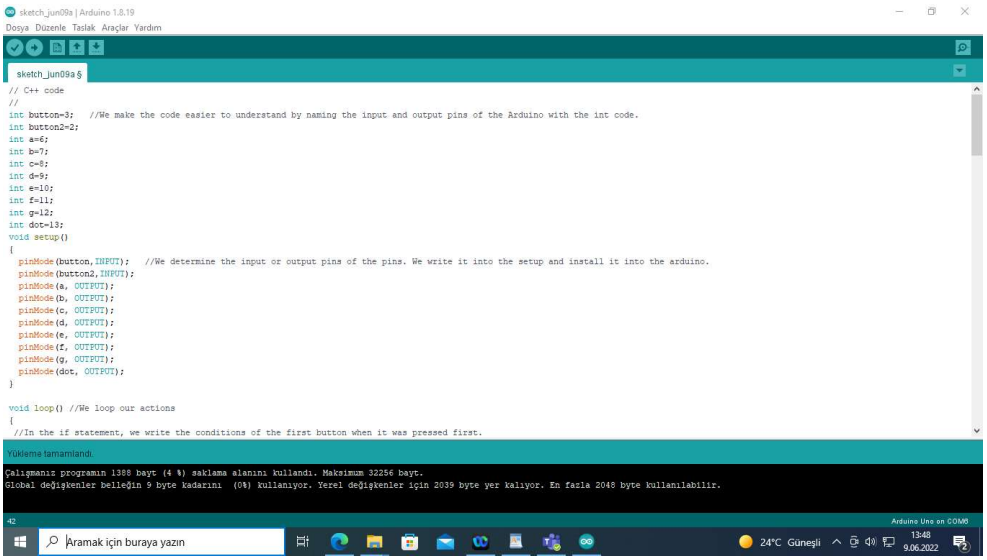


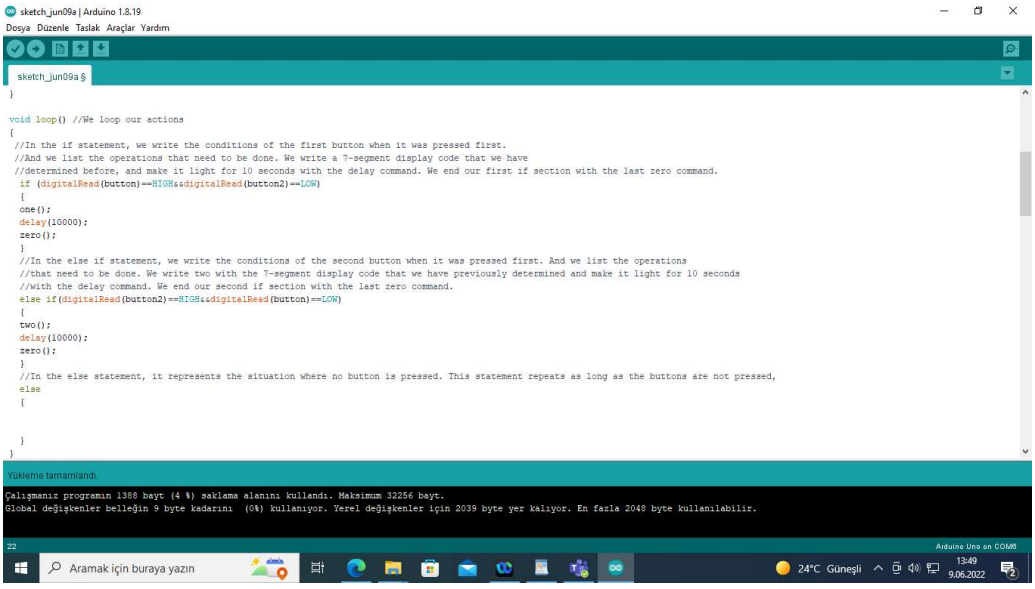


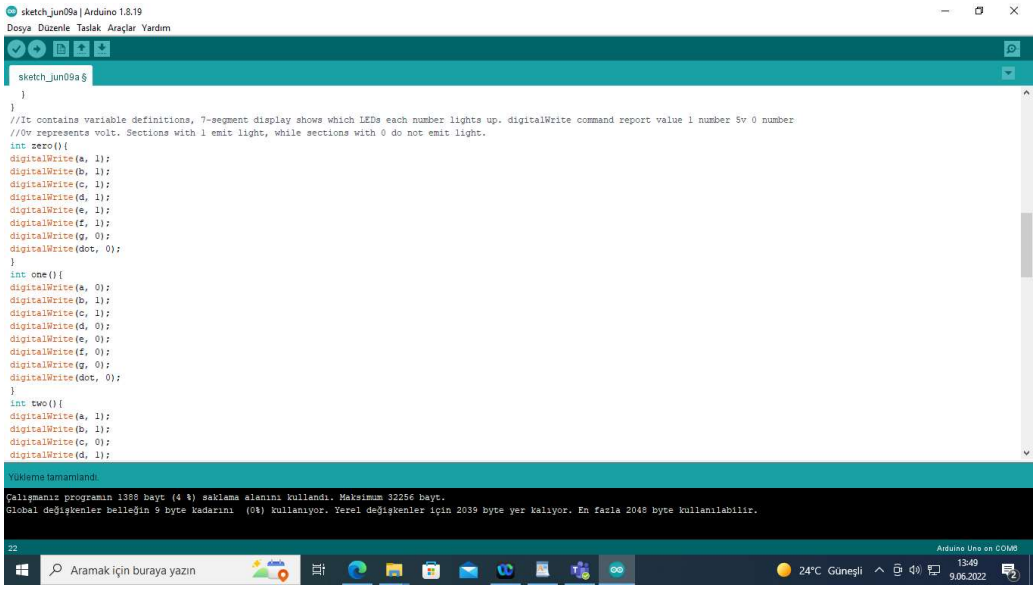


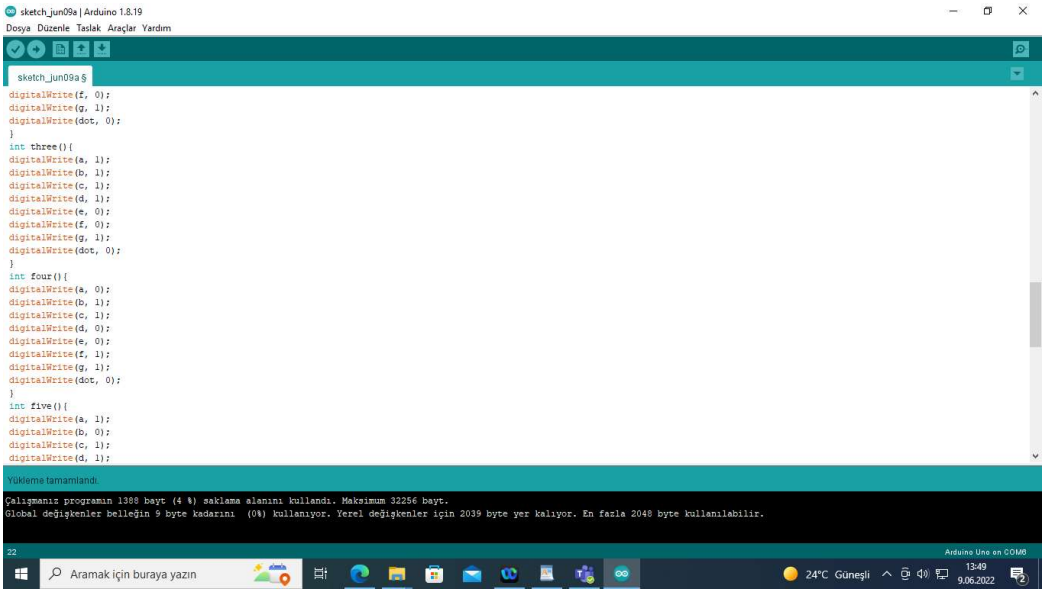
We connected one foot of the buttons to the + part with the jumper cable and connected it to the + part of the 5v pin of the Arduino. We connected the other leg of the buttons to the corresponding pin of the Arduino. The legs of the buttons connected to the Arduino pin in the circuit act as a flat antenna. Therefore, even if the Arduino does not receive a signal, it may act as if it has received a signal and may perform incorrect operations. To prevent this, the GND pin is connected to the leg to which the Arduino pin is connected and short circuit. A resistor is connected to prevent it from happening. The high current draw can be avoided by using a high resistor, so a 10k ohm resistor is used. In the light of the information given above, we connect the 7-section display, which is the common cathode. After plugging it into the relevant pins of the Arduino, our circuit is complete.

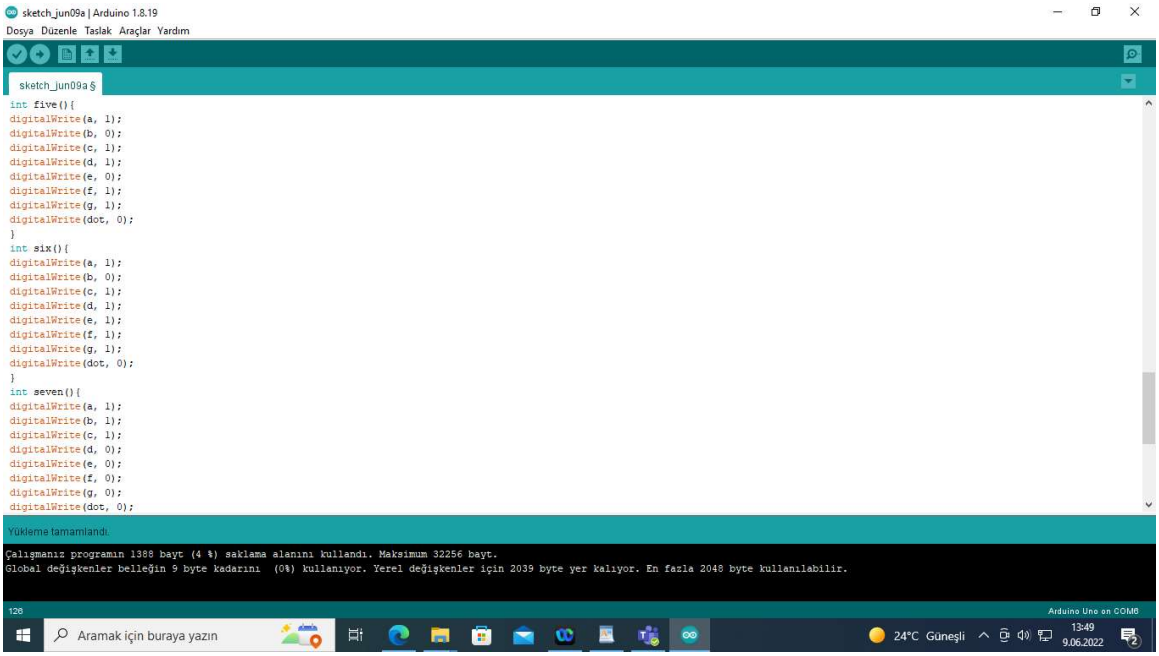
Codes

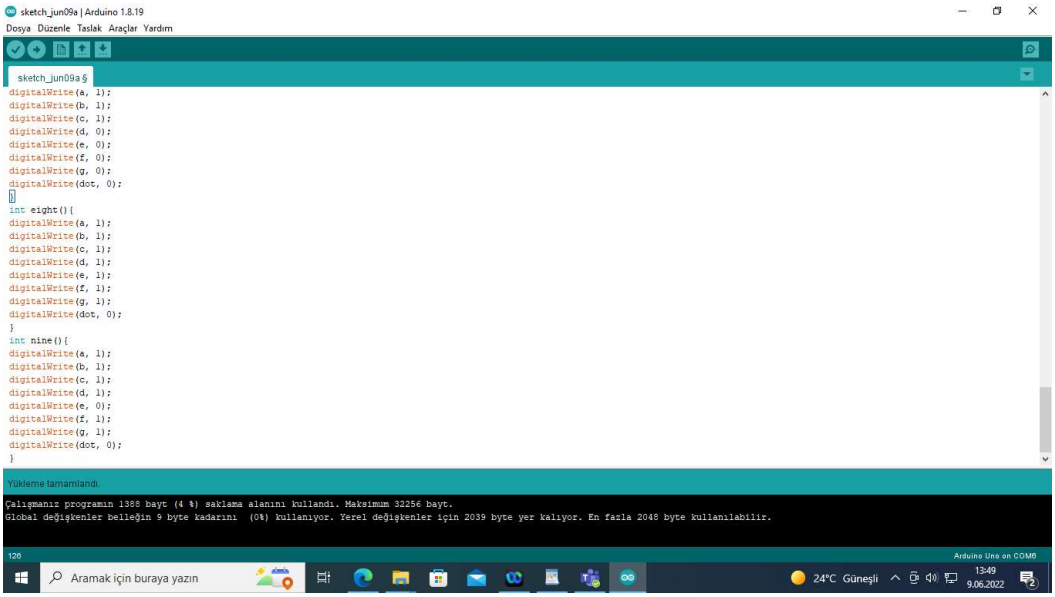

sketch_jun09a | Arduino 1.8.19
Dosya Düzenle Taslak Araçlar Yardım
sketch_jun09a §
// C++ code
//
int button=3; //We make the code easier to understand by naming the input and output pins of the Arduino with the int code.
int button2=2;
int a=6;
int b=7;
int c=8;
int d=9;
int e=10;
int f=11;
int g=12;
int dot=13;
void setup()
{
pinMode(button, INPUT); //We determine the input or output pins of the pins. We write it into the setup and install it into the arduino.
pinMode(button2, INPUT);
pinMode(a, OUTPUT);
pinMode(b, OUTPUT);
pinMode(c, OUTPUT);
pinMode(d, OUTPUT);
pinMode(e, OUTPUT);
pinMode(f, OUTPUT);
pinMode(g, OUTPUT);
pinMode(dot, OUTPUT);
}
void loop() //We loop our actions
{
//In the if statement, we write the conditions of the first button when it was pressed first.
Yükleme tamamlandı.
Çalışmanız programın 1388 bayt (4 %) saklama alanını kullandı. Maksimum 32256 bayt.
Global değişkenler belleğin 9 byte kadarını (0%) kullanıyor. Yerel değişkenler için 2039 byte yer kalıyor. En fazla 2048 byte kullanılabilir.
42
Arduino Uno on COM6
Aramak için buraya yazın
24°C Güneşli
13:48
9.06.2022


sketch_jun09a | Arduino 1.8.19
Dosya Düzenle Taslak Araçlar Yardım
sketch_jun09a §
}
void loop() //We loop our actions
{
//In the if statement, we write the conditions of the first button when it was pressed first.
//And we list the operations that need to be done. We write a 7-segment display code that we have
//determined before, and make it light for 10 seconds with the delay command. We end our first if section with the last zero command.
if (digitalRead(button)==HIGH&&digitalRead(button2)==LOW)
{
one();
delay(10000);
zero();
}
//In the else if statement, we write the conditions of the second button when it was pressed first. And we list the operations
//that need to be done. We write two with the 7-segment display code that we have previously determined and make it light for 10 seconds
//with the delay command. We end our second if section with the last zero command.
else if(digitalRead(button2)==HIGH&&digitalRead(button)==LOW)
{
two();
delay(10000);
zero();
}
//In the else statement, it represents the situation where no button is pressed. This statement repeats as long as the buttons are not pressed,
else
{
}
}
Yükleme tamamlandı.
Çalışmanız programın 1388 bayt (4 %) saklama alanını kullandı. Maksimum 32256 bayt.
Global değişkenler belleğin 9 byte kadarını (0%) kullanıyor. Yerel değişkenler için 2039 byte yer kalıyor. En fazla 2048 byte kullanılabilir.
22
Arduino Uno on COM6
Aramak için buraya yazın
24°C Güneşli
13:49
9.06.2022


sketch_jun09a | Arduino 1.8.19
Dosya Düzenle Taslak Araçlar Yardım
sketch_jun09a §
}
}
//It contains variable definitions, 7-segment display shows which LEDs each number lights up. digitalWrite command report value 1 number 5v 0 number
//0v represents volt. Sections with 1 emit light, while sections with 0 do not emit light.
int zero(){
digitalWrite(a, 1);
digitalWrite(b, 1);
digitalWrite(c, 1);
digitalWrite(d, 1);
digitalWrite(e, 1);
digitalWrite(f, 1);
digitalWrite(g, 0);
digitalWrite(dot, 0);
}
int one(){
digitalWrite(a, 0);
digitalWrite(b, 1);
digitalWrite(c, 1);
digitalWrite(d, 0);
digitalWrite(e, 0);
digitalWrite(f, 0);
digitalWrite(g, 0);
digitalWrite(dot, 0);
}
int two(){
digitalWrite(a, 1);
digitalWrite(b, 1);
digitalWrite(c, 0);
digitalWrite(d, 1);
Yükleme tamamlandı.
Çalışmanız programın 1388 bayt (4 %) saklama alanını kullandı. Maksimum 32256 bayt.
Global değişkenler belleğin 9 byte kadarını (0%) kullanıyor. Yerel değişkenler için 2039 byte yer kalıyor. En fazla 2048 byte kullanılabilir.
22
Arduino Uno on COM6
Aramak için buraya yazın
24°C Güneşli
13:49
9.06.2022


sketch_jun09a | Arduino 1.8.19
Dosya Düzenle Taslak Araçlar Yardım
sketch_jun09a §
digitalWrite(f, 0);
digitalWrite(g, 1);
digitalWrite(dot, 0);
}
int three(){
digitalWrite(a, 1);
digitalWrite(b, 1);
digitalWrite(c, 1);
digitalWrite(d, 1);
digitalWrite(e, 0);
digitalWrite(f, 0);
digitalWrite(g, 1);
digitalWrite(dot, 0);
}
int four(){
digitalWrite(a, 0);
digitalWrite(b, 1);
digitalWrite(c, 1);
digitalWrite(d, 0);
digitalWrite(e, 0);
digitalWrite(f, 1);
digitalWrite(g, 1);
digitalWrite(dot, 0);
}
int five(){
digitalWrite(a, 1);
digitalWrite(b, 0);
digitalWrite(c, 1);
digitalWrite(d, 1);
Yükleme tamamlandı.
Çalışmanız programın 1388 bayt (4 %) saklama alanını kullandı. Maksimum 32256 bayt.
Global değişkenler belleğin 9 byte kadarını (0%) kullanıyor. Yerel değişkenler için 2039 byte yer kalıyor. En fazla 2048 byte kullanılabilir.
22
Arduino Uno on COM6
Aramak için buraya yazın
24°C Güneşli
13:49
9.06.2022





References

www.arduino.cc

Jeremy Blum,Exploring Arduino®: Tools and Techniques for Engineering Wizardry