



Indicator light application

Hardware Related Software Training

Segment / Dept.: VED Software

Agenda:

- Introducere
- Despre proiect
- Declararea variabilelor
- Functionalitatea butoanelor
- Functiile folosite
- Functionalitatea modului de avarii
- Functionalitatea modului normal si schimbare de banda
- Exemplul practic

Embedded C – Arduino

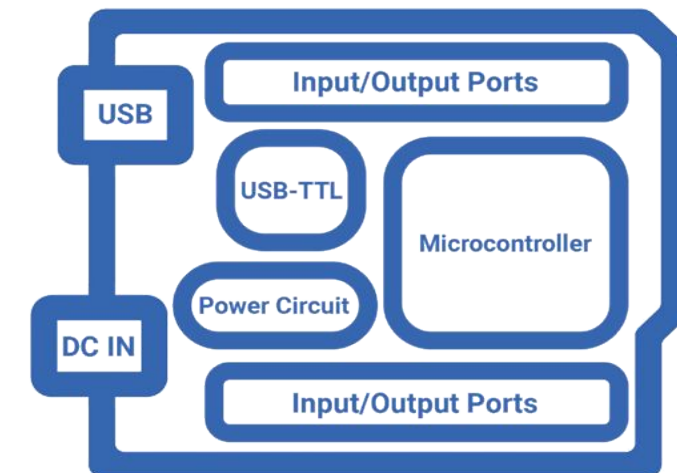
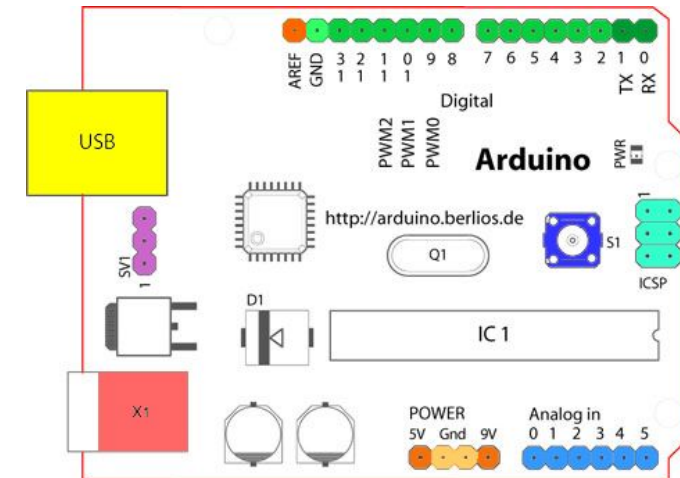
- › **Arduino** este o companie open-source care produce atât plăcuțe de dezvoltare bazate pe microcontrolere, cât și partea de software destinată funcționării și programării acestora
- › Proiectul este bazat pe designul plăcilor cu microcontroler folosind diverse tipuri de microcontrolere
- › Aceste plăci pun la dispoziția utilizatorului pini I/O, digitali și analogici, care pot fi interfațați cu o gamă largă de plăcuțe numite shield-uri și/sau cu alte circuite
- › Plăcile au interfețe de comunicații seriale, inclusiv USB pe unele modele, pentru a încărca programe din PC și pentru interacțiunea online cu plăcuța
- › Pentru programarea microcontrolerelor, Arduino vine cu un mediu de dezvoltare integrat (IDE) bazat pe proiectul Processing, care include suport pentru limbaje de programare ca C și C++



Embedded C – Arduino HW



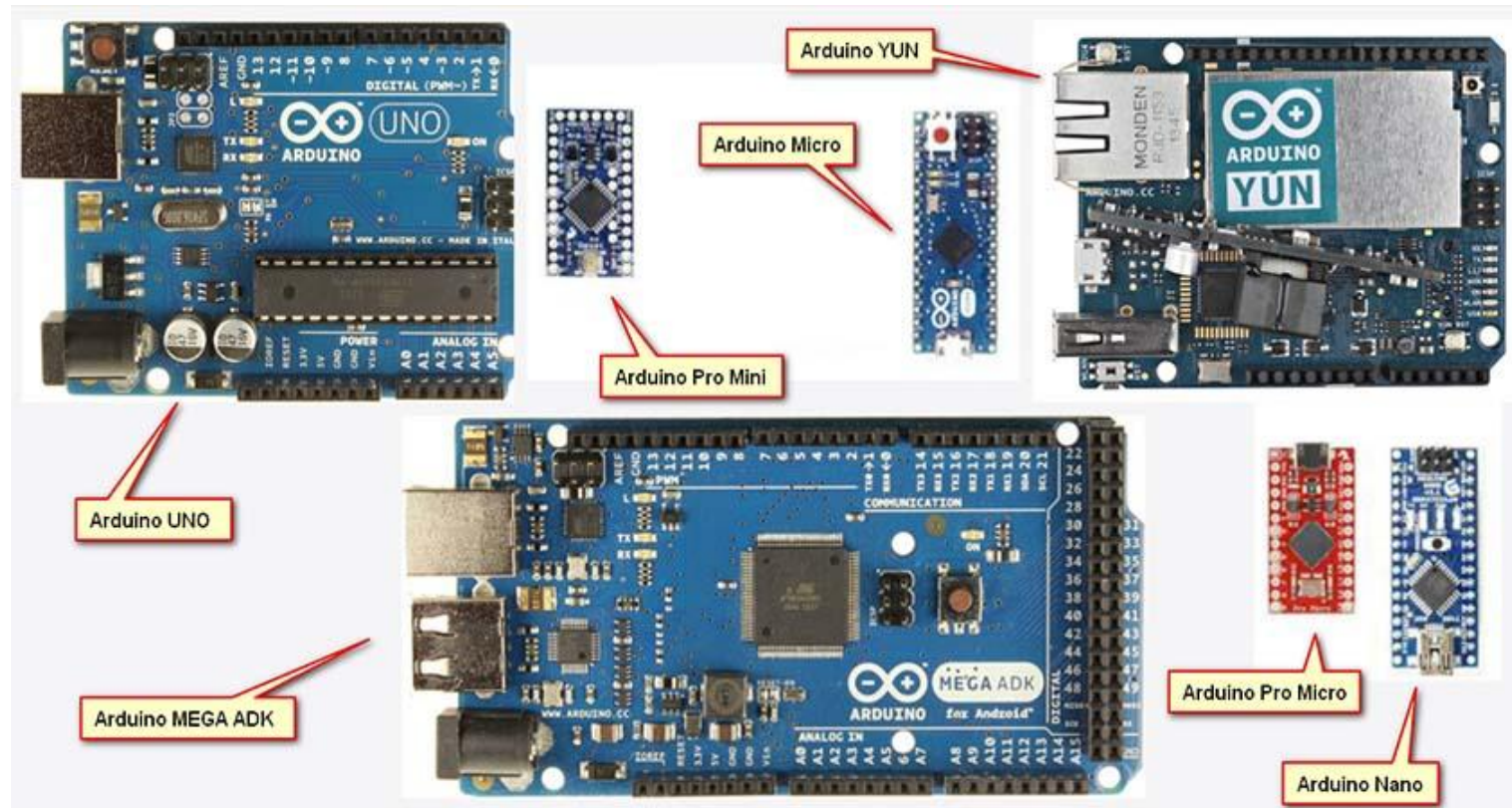
- › O plăcuță Arduino este compusă dintr-un microcontroller Atmel AVR sau compatibil de 8-, 16- sau 32-biți cu componente complementare care facilitează programarea și încorporarea în alte circuite
- › Un aspect important la Arduino este că acesta dispune de conectori standard, care permit utilizatorului să conecteze plăcuța cu procesorul la diferite module interschimbabile numite shield-uri
- › Unele shield-uri comunică cu Arduino direct prin pini digitali sau analogici, dar altele sunt adresabile individual prin magistrala serială I²C permițând utilizarea mai multor module în paralel
- › Multe plăcuțe includ un regulator liniar de 5 V și un oscilator cu cuarț de 16 MHz
- › Un microcontroller instalat pe Arduino vine preprogramat cu un bootloader care simplifică încărcarea programelor pe memoria flash a cipului, în comparație cu alte dispozitive care necesită programatoare externe
- › Acest aspect face Arduino o soluție simplă, permițând programarea de pe orice PC
- › Bootloader-ul optiboot este bootloader-ul implicit instalat pe Arduino UNO



Embedded C – Arduino HW



› Modele de plăcuțe Arduino existente, mai des întâlnite:



Agenda:

- Introducere
- Despre proiect
- Declararea variabilelor
- Functionalitatea butoanelor
- Functiile folosite
- Functionalitatea modului de avarii
- Functionalitatea modului normal si schimbare de banda
- Exemplul practic

Despre proiect:

- Numarul de push-buttons
- Moduri de functionare
- Activarea modului de schimbare de banda
- Efectul modului de schimbare de banda
- Activarea modului normal
- Efectul modului normal
- Activarea modului de avarii
- Efectul modului de avarii

Agenda:

- Introducere
- ~~Despre proiect~~
- ~~Declararea variabilelor~~
- Functionalitatea butoanelor
- Functiile folosite
- Functionalitatea modului de avarii
- Functionalitatea modului normal si schimbare de banda
- Exemplul practic

Declararea variabilelor:

```
const int buttonLeftPin = 4;
const int buttonRightPin = 2;
const int buttonEmergencyPin = 3;
const int ledLeftPin = 7;
const int ledRightPin = 8;

bool hazardModeActive = false;
unsigned long hazardModeStartTime = 0;

void setup() {
  pinMode(buttonLeftPin, INPUT);
  pinMode(buttonRightPin, INPUT);
  pinMode(buttonEmergencyPin, INPUT);
  pinMode(ledLeftPin, OUTPUT);
  pinMode(ledRightPin, OUTPUT);

  Serial.begin(9600);
}
```

Agenda:

- Introducere
- Despre proiect
- Declararea variabilelor
- Functionalitatea butoanelor
- Functiile folosite
- Functionalitatea modului de avarii
- Functionalitatea modului normal si schimbare de banda
- Exemplul practic

Functionalitatea butoanelor:

```
bool buttonPressedFor(int buttonPin, unsigned long duration) {  
    unsigned long startTime = millis();  
    while (digitalRead(buttonPin) == HIGH) {  
        if (millis() - startTime >= duration) {  
            return true;  
        }  
    }  
    return false;  
}
```



Agenda:

- Introducere
- Despre proiect
- Declararea variabilelor
- Functionalitatea butoanelor
- Functiile folosite
- Functionalitatea modului de avarii
- Functionalitatea modului normal si schimbare de banda
- Exemplul practic

Funcțiile folosite:

```
void blinkLED(int ledPin, int times) {  
    for (int i = 0; i < times; i++) {  
        digitalWrite(ledPin, HIGH);  
        delay(300);  
        digitalWrite(ledPin, LOW);  
        delay(700);  
    }  
}
```

```
void blinkLEDContinuously(int ledPin) {  
    while (digitalRead(buttonLeftPin) == HIGH || digitalRead(buttonRightPin) == HIGH) {  
        if(digitalRead(buttonEmergencyPin) == HIGH)  
        {  
            break;  
        }  
        digitalWrite(ledPin, HIGH);  
        delay(300);  
        digitalWrite(ledPin, LOW);  
        delay(700);  
    }  
}  
  
void activateHazardMode() {  
    hazardModeStartTime = millis();  
    hazardModeActive = true;  
}
```

```
void deactivateHazardMode() {  
    hazardModeActive = false;  
    digitalWrite(ledLeftPin, LOW);  
    digitalWrite(ledRightPin, LOW);  
}
```


Agenda:

- Introducere
 - Despre proiect
 - Declararea variabilelor
 - Functionalitatea butoanelor
-
- Functiile folosite
-
- Functionalitatea modului de avarii
 - Functionalitatea modului normal si schimbare de banda
 - Exemplul practic

Functionalitatea modului de avarii:

```
// Verificăm dacă semnalizarea de avarie trebuie activată sau dezactivată
if (digitalRead(buttonEmergencyPin) == HIGH) {
    if (!hazardModeActive) {
        activateHazardMode();
    } else {
        deactivateHazardMode();
    }
    // Debounce the button press
    delay(200);
}

// Verificăm dacă semnalizarea de avarie este activată
if (hazardModeActive) {
    unsigned long currentTime = millis();
    if (currentTime - hazardModeStartTime >= 200) {
        hazardModeStartTime = currentTime;
        // Comutăm între LED-urile stânga și dreapta
        digitalWrite(ledLeftPin, !digitalRead(ledLeftPin));
        digitalWrite(ledRightPin, !digitalRead(ledRightPin));
    }
    // Nu verificăm butoanele individuale în timpul semnalizării de avarie
    return;
}
```

Agenda:

- Introducere
 - Despre proiect
 - Declararea variabilelor
 - Functionalitatea butoanelor
 - Functiile folosite
-
- Functionalitatea modului de avarii
-
- Functionalitatea modului normal si schimbare de banda
 - Exemplul practic

Functionalitatea modului normal si schimbare de banda:

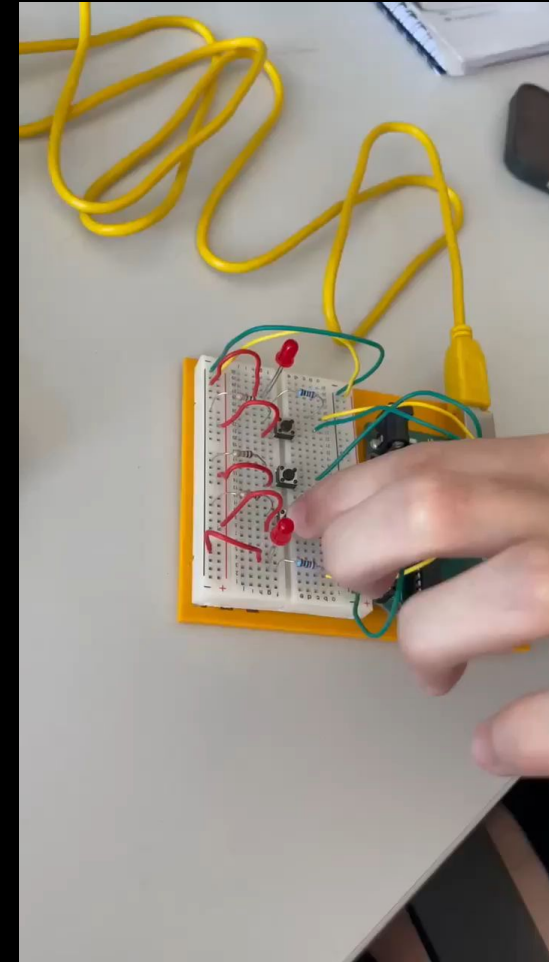
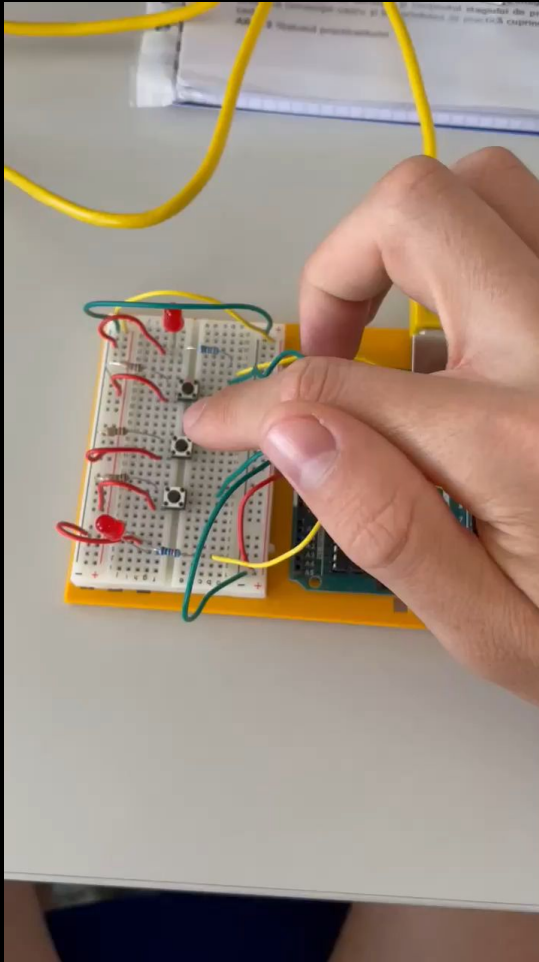
```
// Verificăm butonul pentru semnalizarea stânga
if (digitalRead(buttonLeftPin) == HIGH) {
    if (buttonPressedFor(buttonLeftPin, 500)) {
        blinkLEDContinuously(ledLeftPin);
    } else {
        blinkLED(ledLeftPin, 3);
    }
}
```

```
// Verificăm butonul pentru semnalizarea dreapta
if (digitalRead(buttonRightPin) == HIGH) {
    if (buttonPressedFor(buttonRightPin, 500)) {
        blinkLEDContinuously(ledRightPin);
    } else {
        blinkLED(ledRightPin, 3);
    }
}
}
```

Agenda:

- Introducere
 - Despre proiect
 - Declararea variabilelor
 - Functionalitatea butoanelor
 - Functiile folosite
 - Functionalitatea modului de avarii
-
- Functionalitatea modului normal si schimbare de banda
-
- Exemplul practic

Exemplul practic:



Multumim pentru timpul acordat!

Participanti:

- Mitrica Alexandru
- Foamete Dan
- Penoiu Cristian
- Lupu Adelin
- Glava Alexandru