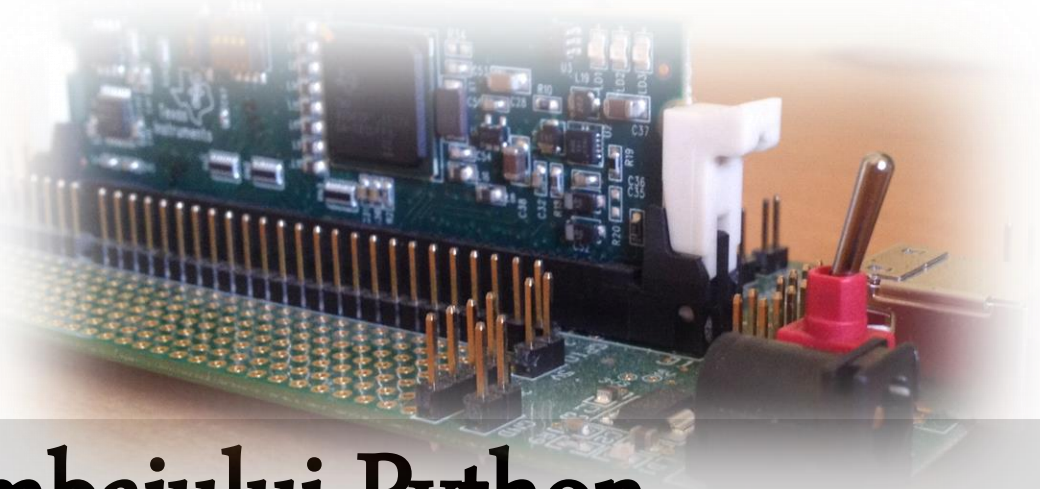
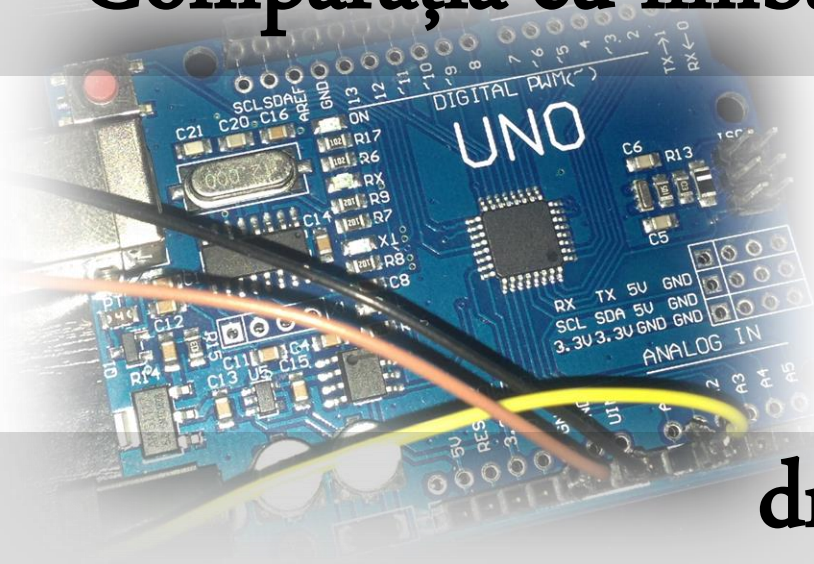


Sisteme de calcul în timp real



Sintaxa limbajului Python

- Comparația cu limbajul Wiring (Arduino IDE) -

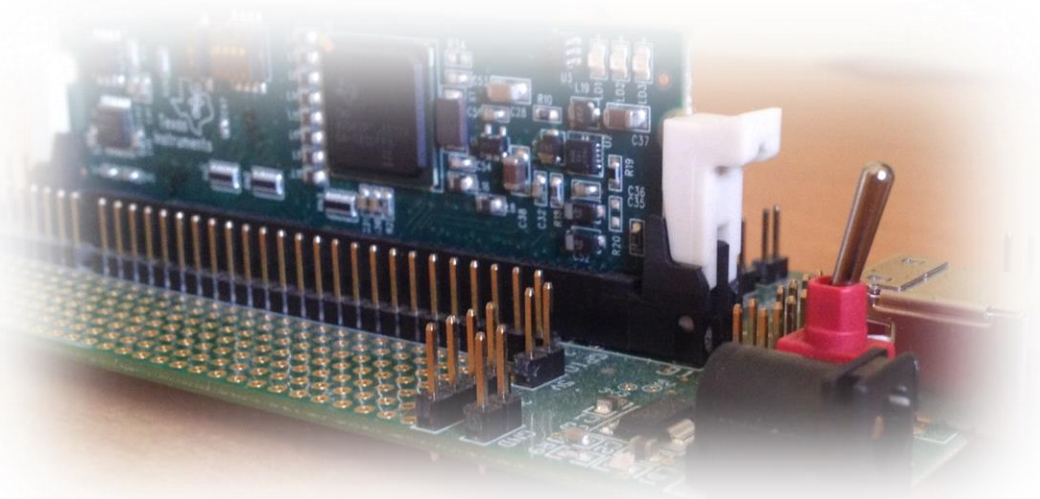


dr. ing. Pintilie Lucian Nicolae

Noțiuni specifice oricărui limbaj de programare

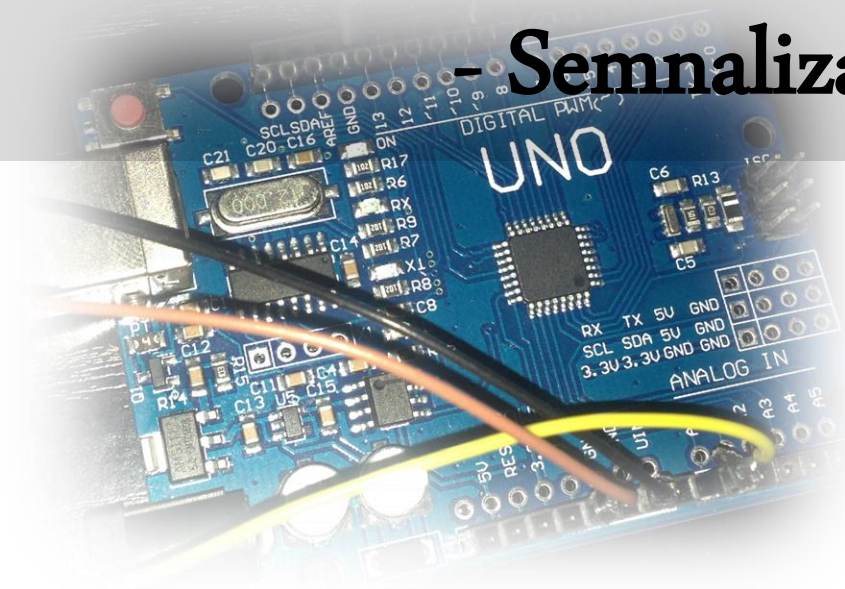
- În orice limbaj de programare există elemente și structuri sintactice specifice precum:
 - ✓ **Directive de pre / post procesare** (include, return etc...);
 - ✓ **Attribute** (const, unsigned etc...);
 - ✓ **Tip de date** (integer, float etc...);
 - ✓ **Comentarii**;
 - ✓ **Funcții specifice limbajului** (digitalWrite, delay etc...)

Sisteme de calcul în timp real



Structura unui program în limbaj Wiring

- Semnalizare intermitentă -



```
#include delayMilliseconds.h
```

```
const int led_rosu = 9;
```

```
void setup()
```

```
{
```

```
pinMode(led_rosu, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
digitalWrite(led_rosu, HIGH);
```

```
delay(100);
```

```
digitalWrite(led_rosu, LOW);
```

```
delay(100);
```

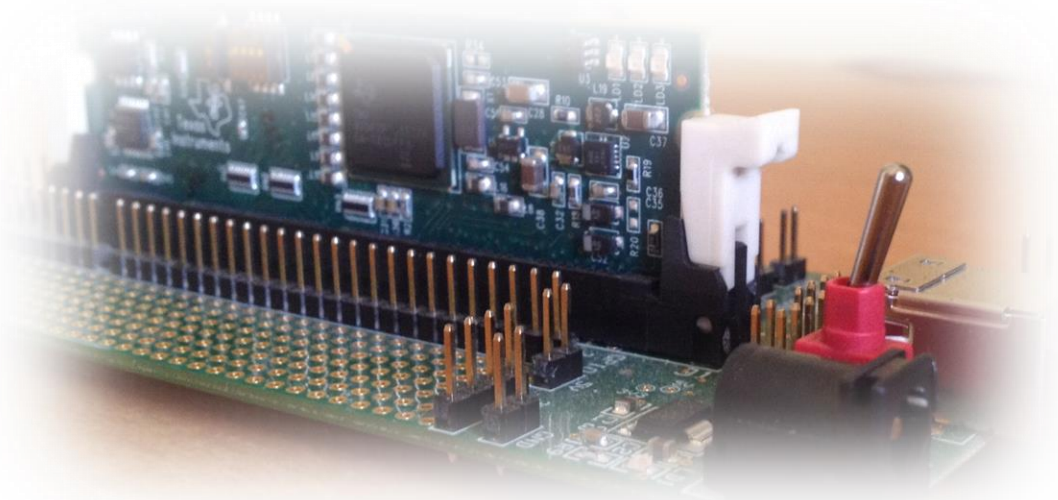
```
}
```

```
#include delayMilliseconds.h // incluziune de bibleoteci;  
  
const int led_rosu = 9; // constant pe tot parcursul execuției;  
  
void setup() // buclă care se execută o singură dată;  
{  
  pinMode(led_rosu, OUTPUT); // mod de lucru al pinului;  
}  
  
void loop() // buclă care se execută la infinit (ech. while(1));  
{  
  digitalWrite(led_rosu, HIGH); // impunerea stării digitale ACTIV;  
  delay(100); // impunere întârziere 100 [ms];  
  digitalWrite(led_rosu, LOW); // impunerea stării digitale INACTIV;  
  delay(100); // impunere întârziere 100 [ms];  
}
```

Limbaajul de programare Wiring

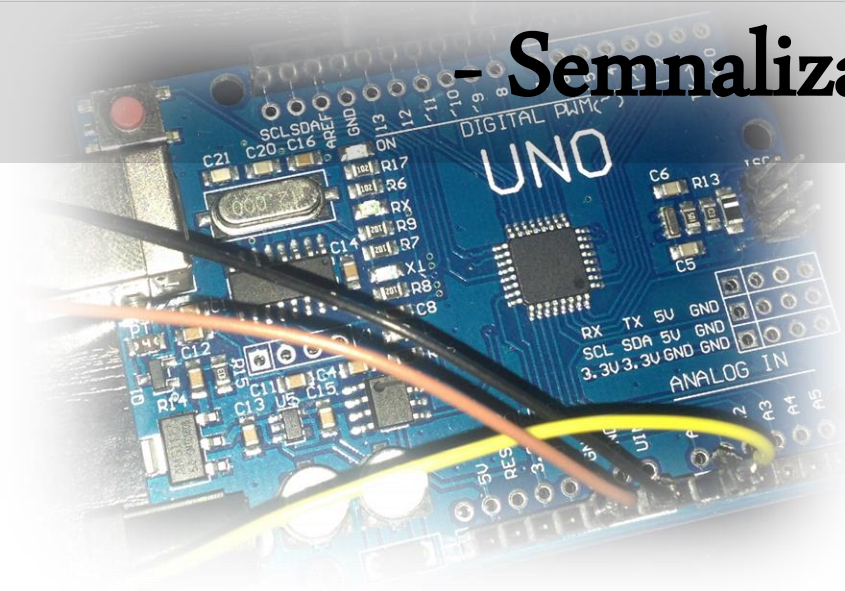
- Este un derivat al limbajului „C” standard sau „C++”;
- Astfel, se vor prelua (din „C”) aspectele de bază precum:
 - ❖ Operația de „incluziune” a bibliotecilor (ex. `#include <...>`);
 - ❖ Declararea tipurilor de date (ex. „int”, „float” etc...);
 - ❖ Delimitarea secțiunilor de cod prin acolade „{”;
 - ❖ Introducerea comentariilor prin adăugarea caracterelor „/”;
 - ❖ Semnalarea capătului de linie de cod prin caracterul „;” ;
 - ❖ Semnalarea funcțiilor prin caracterele „()”;
- În completarea acestor aspecte intră funcțiile specifice limbajului Wiring (ex. **digitalWrite()**; **analogRead()**; etc...);

Sisteme de calcul în timp real



Structura unui program în limbaj Python

- Semnalizare intermitentă -



```
#!/usr/bin/python
```

```
import mraa
```

```
import time
```

```
x = mraa.Gpio(9)
```

```
x.dir(mraa.DIR_OUT)
```

```
while True:
```

```
    x.write(1)
```

```
    time.sleep(0.5)
```

```
    x.write(0)
```

```
    time.sleep(0.5)
```

<code>#!/usr/bin/python</code>	<code>#Specificare cale absolută interpretor Python;</code>
<code>import mraa</code>	<code>#Importare bibliotecă "mraa";</code>
<code>import time</code>	<code>#Importare bibliotecă "time";</code>
<code># Setup</code>	<code>#Comentariu;</code>
<code>x = mraa.Gpio(9)</code>	<code>#Adresarea ieșirii digitale;</code>
<code>x.dir(mraa.DIR_OUT)</code>	<code>#Configurarea modului de lucru al ieșirii;</code>
<code># Loop</code>	
<code>while True:</code>	<code>#Bucă „WHILE”;</code>
<code>x.write(1)</code>	<code>#Activarea ieșirii;</code>
<code>time.sleep(0.5)</code>	<code>#Așteptare 500 milisecunde;</code>
<code>x.write(0)</code>	<code>#De-activarea ieșirii;</code>
<code>time.sleep(0.5)</code>	<code>#Așteptare 500 milisecunde;</code>

Limbajul de programare Python

- Este un limbaj nativ al platformelor de tip Unix;
- În sintaxă, introduce elemente noi față de limbajele „C”:
 - ❖ Specificarea căii de acces înspre compilator;
 - ❖ Lipsa caracterelor delimitatoare ale capătului de linie;
 - ❖ Delimitarea secțiunilor de cod prin aliniere la stânga;
 - ❖ Introducerea comentariilor prin caracterul „#”;
- Există și instrucți / declarații fizice (eng. hard) specifice limbajului! (ex. **write()**; **read()**; etc...);

Exemple de elemente de sintaxă echivalente

Wiring (Arduino IDE)

```
#include <>
pinMode ()
loop ()
digitalWrite()
digitalWrite()
Serial.print("")
```

Python

```
import <>
mraa.DIR_<IN / OUT>
while True
write()
read()
print("")
```