



IoT Bootcamp 2020

Sesiunea 6

Agenda sesiunii de azi

17:00-17:40	Intro & Special topics
17:40-17:50	Pauză
17:50-18:20	Proces dezvoltare si integrare
18:20-19:00	Implementare practică Integrare proiect
19:00	Q&A

Prezentatorii sesiunii de azi



Diana Marusic

Project Manager
IoT Bootcamp,
Studentă UTM



Andrei Bragarenco

Fondatorul Clubului Ingineresc
Micro Lab

#anunțuri

- Ultima sesiune de prezentare online
- Ultima zi de lucru la proiecte – **3 iulie**
- Prezentare proiecte & competiție – **15 iulie**
- **Magic Quest**
- **Criterii de jurizare** – vor fi publicate in curand
- **Punctaje teme pentru acasa** – vor fi publicate in curand

Timeline

4 iunie	Ultima sesiune de prezentari online
4 iunie-3 iulie	Lucru la proiecte
3 iulie	Deadline lucru la proiecte
15 iulie	Prezentare finala & jurizare

Project planning & tracking

What is Project Management?

S

Specific

State what you'll do
Use action words

M

Measurable

Provide a way to
evaluate
Use metrics or data
targets

A

Achievable

Within your scope
Possible to accom-
plish, attainable

R

Relevant

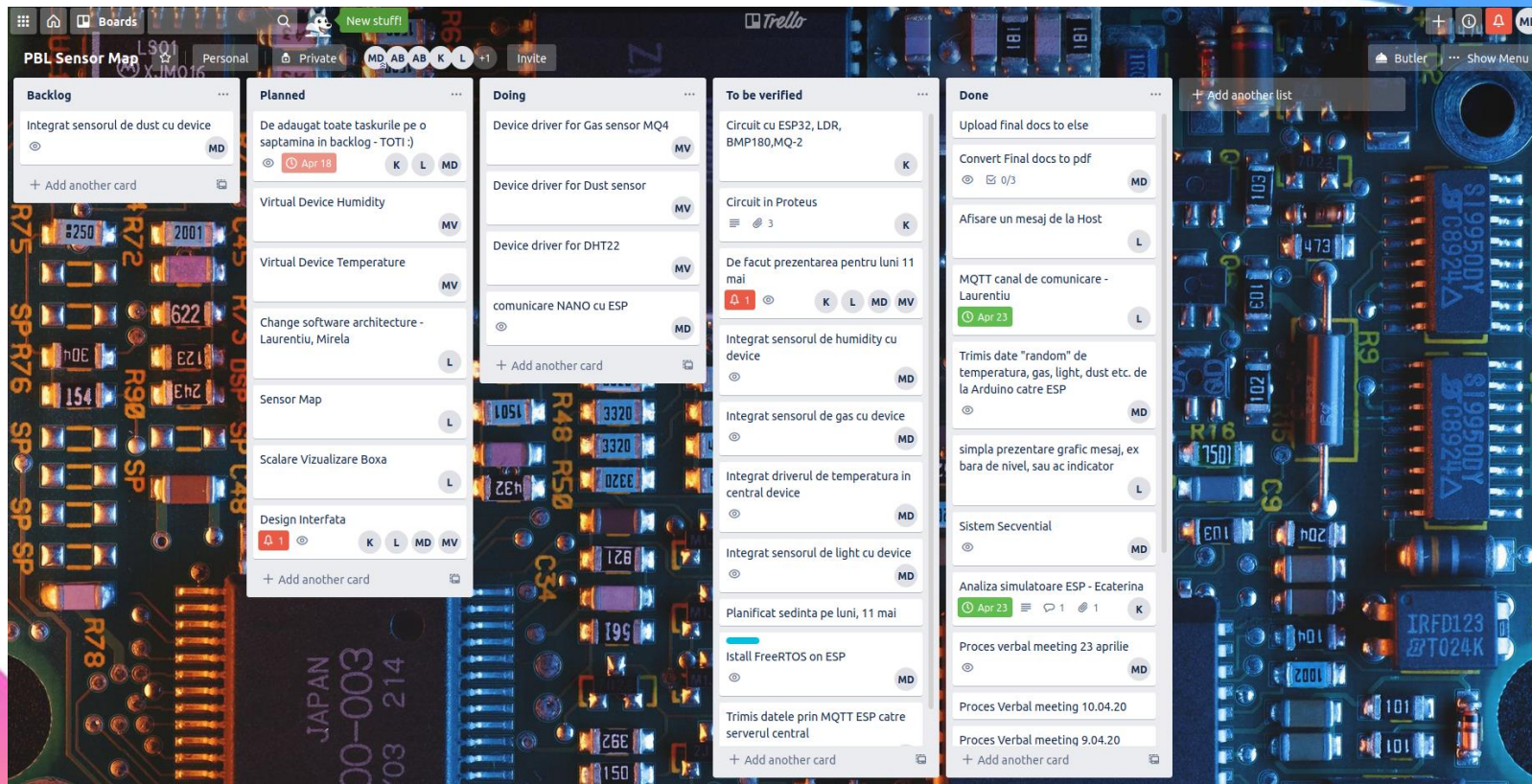
Makes sense within
your job function
Improves the
business in some
way

T

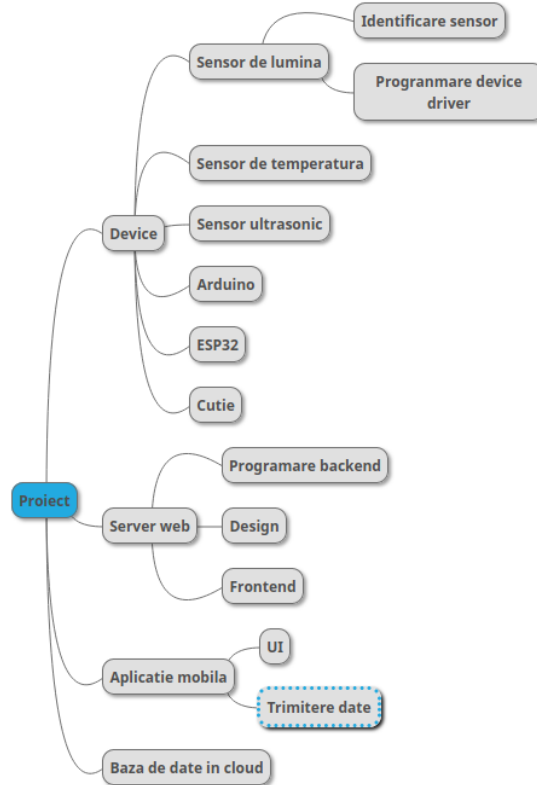
Time-bound

State when you'll get
it done
Be specific on date
or timeframe

Trello

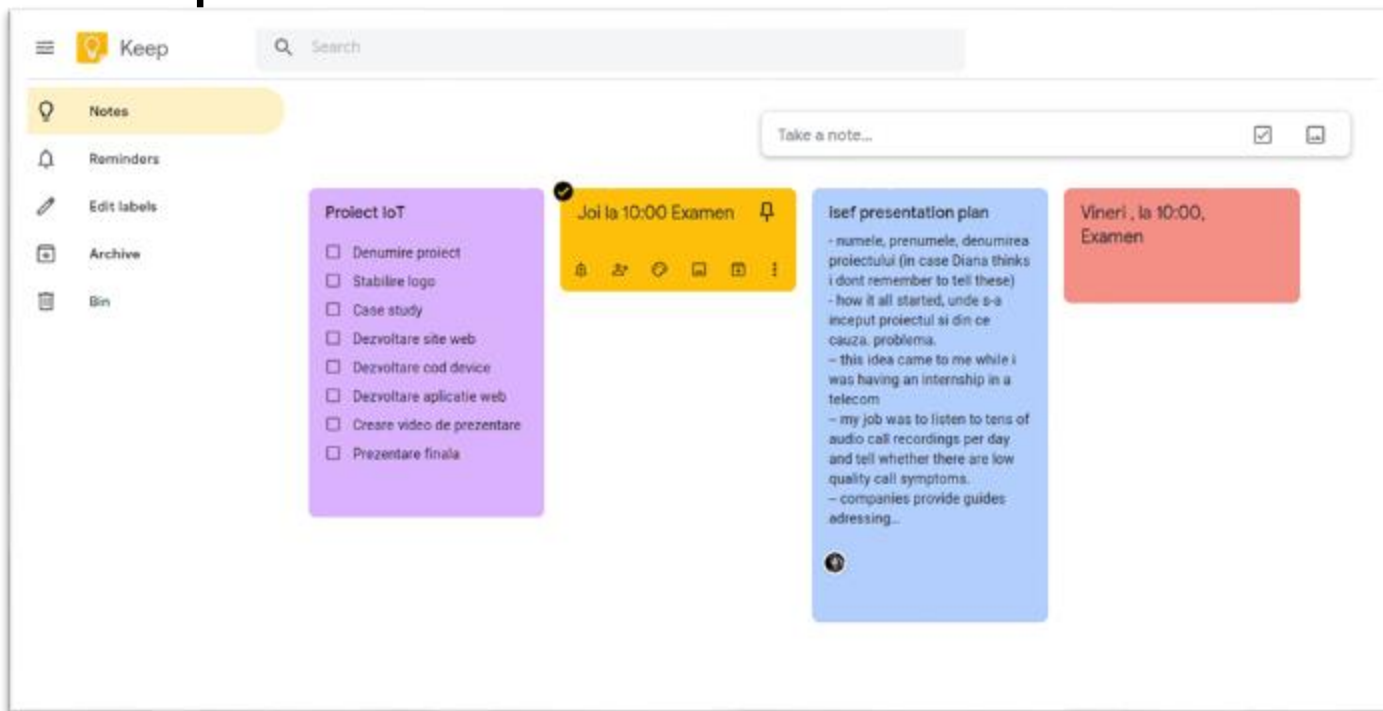


Mindmap



<https://app.mindmup.com/>

Google keep



<https://keep.google.com>



Echipamente

Echipamente proiecte

- 37 in 1 sensor kit

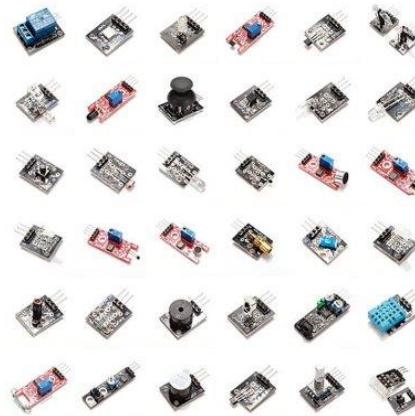
<https://www.instructables.com/id/Arduino-37-in-1-Sensors-Kit-Explained/>

- Elego Mega kit

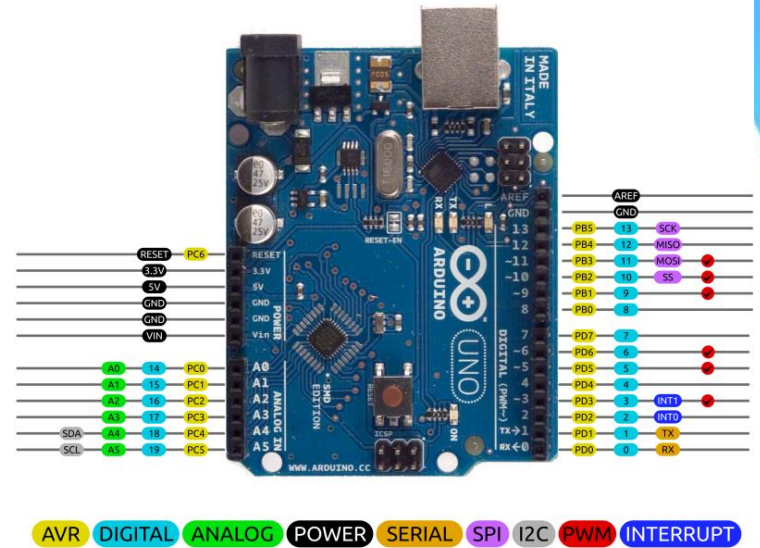
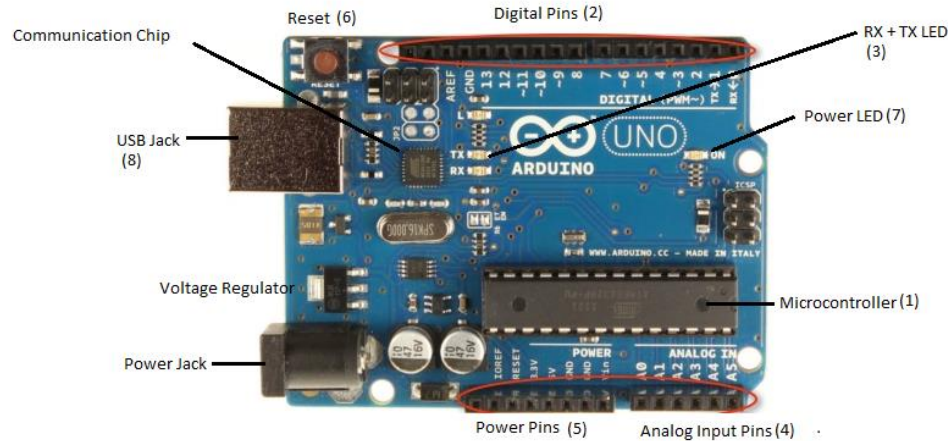
<https://www.elegoo.com/product/elegoo-mega-2560-project-the-most-complete-starter-kit/>

- ESP32

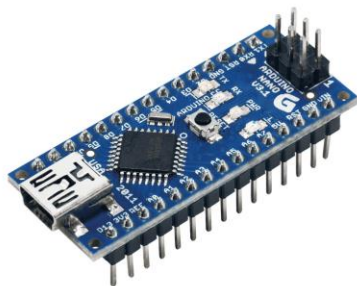
<https://www.espressif.com/en/products/socs/esp32/overview>



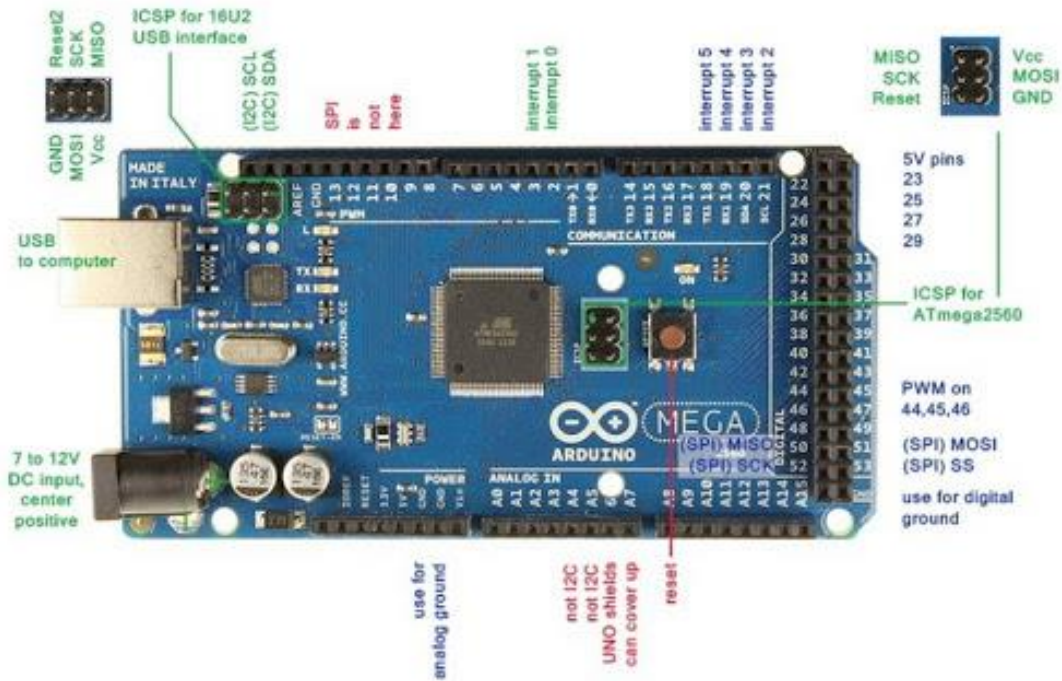
Arduino - privire de ansamblu



Arduino - privire de ansamblu



Arduino Nano



Arduino Mega

ESP32 – privire de ansamblu



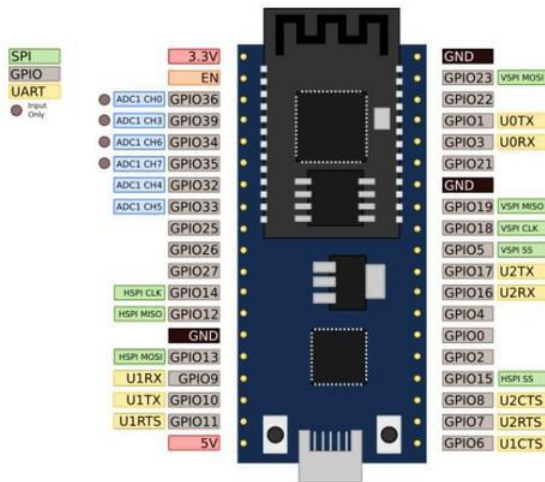
Robust Design

ESP32 is capable of functioning reliably in industrial environments, with an operating temperature ranging from -40°C to $+125^{\circ}\text{C}$. Powered by advanced calibration circuitries, ESP32 can dynamically remove external circuit imperfections and adapt to changes in external conditions.



High Level of Integration

ESP32 is highly-integrated with in-built antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. ESP32 adds priceless functionality and versatility to your applications with minimal Printed Circuit Board (PCB) requirements.



Ultra-Low Power Consumption

Engineered for mobile devices, wearable electronics and IoT applications, ESP32 achieves ultra-low power consumption with a combination of several types of proprietary software. ESP32 also includes state-of-the-art features, such as fine-grained clock gating, various power modes and dynamic power scaling.



Hybrid Wi-Fi & Bluetooth Chip

ESP32 can perform as a complete standalone system or as a slave device to a host MCU, reducing communication stack overhead on the main application processor. ESP32 can interface with other systems to provide Wi-Fi and Bluetooth functionality through its SPI / SDIO or I2C / UART interfaces.

Despre cod

Dacă citesc codul peste un an, îl voi înțelege oare?



Clean code - caracteristici

- Trebuie să fie **elegant** – Codul trebuie să fie ușor de citit. Atunci când îl citești trebuie să zâmbești la fel cum auzi o muzică plăcută sau vezi o mașină cu un design frumos
- Trebuie să fie **focusat** – Fiecare **funcție**, fiecare **clasă**, fiecare **modul** expune **o singură idee**, care rămâne complet distanțată, și neafectată de detaliile înconjurătoare
- Are nevoie de cineva **care să aibă grijă de cod** – Cineva trebuie să aloce timp să-l păstreze **simplic și ordonat**, să atragă atenția la **detalii**
- Rulează toate **testele**
- **Nu conține duplicate**
- **Nu conține bucăți de cod comentate fără motiv sau spații excesive**
- Sunt atribuite **nume sugestive** de funcții, variabile, module etc.

<https://medium.com/mindorks/how-to-write-clean-code-lessons-learned-from-the-clean-code-robert-c-martin-9ffc7aef870c>

Coding conventions

Un set de reguli pentru un anumit limbaj de programare care recomandă stilul de programare, practici bune de a scrie cod și metode ce se referă la orice aspect al programelor scrise în limbajul respectiv.

Coding conventions

- File organization(organizarea fișierelor)
- Indentation(indentare)
- Comments(comentarii)
- Declarations (declarații de variabile/fucții etc)
- Statements
- White space (spații libere)
- Naming conventions
- Programming practices
- Programming principles
- Programming rules
- Architectural best practices

Coding conventions – de ce?

- Intre **40-60% din costurile** a unui proiect software sunt alocate **menținerii proiectului**
- De obicei un proiect nu este menținut pe toata durata de către un singur autor, de obicei e dezvoltat și menținut **de mai mulți dezvoltatori**
- Convențiile de scriere a codului **îmbunătățesc citibilitatea codului-sursă**, ajutând inginerii să **înțeleagă** mult **mai ușor** și **mai bine codul unui program** la prima vedere
- Atunci când **codul sursă** este livrat ca un **produs**, este nevoie să ne asigurăm că este **"împachetat"** și **"curat"** ca orice alt produs pe care îl creăm

Naming conventions

Un **set de reguli** pentru alegerea **numelor** identificatorilor **variabilelor**, **tipurilor de date**, **funcțiilor** și altor **entități** din **codul sursă** al programelor și **documentației**

Naming conventions – de ce?

- Pentru a **reduce efortul** necesar de a **citi și înțelege codul sursă**
- Pentru a ajuta în timpul **analizei calității codului (code reviews)** ca programatorii să se **focuseze pe chestii mai importante** decât **sintaxa și numele variabilelor**
- Pentru a ajuta **instrumentele de code quality** să se focuseze pe **raportarea problemelor mai importante** decât sintaxa și preferințele de stilizare a codului
- Pentru a avea o **metodă unică** pentru **toți dezvoltatorii** unui program de a **scrie și înțelege** codul

Letter case

- **Uppercase:** ABCDEFGHIJKLMNOPQRSTUVWXYZ; **VARIABLE**
- **Lowercase:** abcdefghijklmnopqrstuvwxyz; **variable**
- **Camel Case:** camelCaseNewWord, **myVariable**
- **Snake Case:** snake_case_variable, **my_variable**
- **Kebab case:** kebab-case-variable, **my-variable**

Indentare

Spații libere la începutul rândului, semnificând începutul unui nou paragraf

Indentare

```
int power(int x, int y)
{
    int result;

    if (y < 0) {
        result = 0;
    } else {
        result = 1;
        while (y-- > 0)
            result *= x;
    }
    return result;
}
```


```
int power(int x, int y) {
    int result;

    if (y < 0) {
        result = 0;
    } else {
        result = 1;
        while (y-- > 0) result *= x;
    }
    return result; }
```

Indentare


```
int power(int x, int y)
{
    int result;

    if (y < 0) {
        result = 0;
    } else {
        result = 1;
        while (y-- > 0)
            result *= x;
    }
    return result;
}
```



```
int power(int x, int y) {
int result;

if (y < 0) {
result = 0;
} else {
result = 1;
while (y-- > 0) result *= x;
}
return result; }
```



Acolade

```
{  
    if (depth_in_ft > 10) dive_stage = DIVE_DEEP;           // This is legal...  
    else if (depth_in_ft > 0)  
        dive_stage = DIVE_SHALLOW;                        // ... as is this.  
    else  
    {                                                        // But using braces is always safer.  
        dive_stage = DIVE_SURFACE;  
    }  
    ...  
}
```

Acolade

```
{  
    if (depth_in_ft > 10) dive_stage = DIVE_DEEP; X // This is legal...  
    else if (depth_in_ft > 0) X  
        dive_stage = DIVE_SHALLOW; // ... as is this.  
    else  
    { // But using braces is always safer.  
        dive_stage = DIVE_SURFACE;  
    }  
    ...  
}
```



Comentarii

- Comentarii explicative **ce face codul**
- Comentarii explicative la **începutul funcțiilor**
- **Bucățile de cod nu se scriu în comentarii, nici temporar!**
- Pentru comentariile cu mai multe linii, **nu uităm să închidem comentariul!**

```
/* The following code was meant to be part of the build...  
  
...  
  
safety_checker();  
  
...  
  
/* ... but an end of comment character sequence was omitted. */
```

<https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard/comment-rules/acceptable-comment-formats>

Modularizare

- O funcție = o funcționalitate
- O librărie = O colecție de funcții cu un anumit scop
- O librărie în C - fișiere .c și .h

communication.cpp

communication.h

dust.cpp

dust.h

esp.ino

format_data.cpp

format_data.h

led.cpp

led.h

light.cpp

light.h

Librării

.h - interface

Conține definiții pentru ceea ce este exportat de librărie:

- Prototipurile funcțiilor cu comentarii pentru utilizatorii librăriei
- Definițiile tipurilor de date și variabilelor globale exportate de librărie

.c - implementation

Conține implementarea fiecărei funcții din fișierul .h

Librării

.h - interface

```
#ifndef _MYLIB_H_
#define _MYLIB_H_

// a constant definition exported by library:
#define MAX_FOO 20

// a type definition exported by library:
struct foo_struct {
    int x;
    float y;
};
typedef struct foo_struct foo_struct;

// a global variable exported by library
// "extern" means that this is not a variable declaration, it
// just defines that a variable named total_foo of type int
// exists and you can use it (its declaration is in some library source file)
extern int total_foo;

// a function prototype for a function exported by library:
extern int foo(float y, float z); // a very bad function name

#endif
```

.c - implementation

```
#include "mylib.h"

...
int total_foo;

int foo(float y, float z) {
    ...
}
```



Resurse Coding Conventions in C

- <https://developer.gnome.org/programming-guidelines/stable/c-coding-style.html.en>
- <https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard>
- https://en.wikipedia.org/wiki/Coding_conventions

Github



How to use GitHub

- Step 1 ➤ Sign up for GitHub
- Step 2 ➤ Create repository
- Step 3 ➤ Install and set up Git
- Step 4 ➤ Clone the remote repository
- Step 5 ➤ Make changes to files
- Step 6 ➤ Add changes to the staging area
- Step 7 ➤ Commit changes
- Step 8 ➤ Push changes to the remote

<https://github.com/>

A man with a mustache is shown from the chest up, sitting in the driver's seat of a car. He is yawning with his mouth wide open and eyes closed. The text "Like a Bosch" is overlaid in large white font. The background shows a blurred view of trees and a road through the car window.

Like a Bosch

Pauză
Ne vedem la **17:50**

Dezvoltare și Integrare proiect

Bragarenco Andrei

Formular de feedback (+2 puncte)

<https://bit.ly/3714PRQ>

Mulțumim pentru atenție!