ebotics

Arm robot
First steps

Arm

Now that you have built the Arm Robot , you can set the Build&Code UNO board using any of the following software: Arduino IDE, Bitbloq or a visual programming software by blocks compatible with the Build&Code UNO board. Choose the software you want to use and follow the corresponding steps. Note that every software has a different complexity level:
• Arduino IDE: advanced users
• Bitbloq: intermediate users
• Visual programming software by blocks: beginner users

Configuration of the Build&Code UNO board

Configure the Ebotics Build&Code UNO board for Arduino IDE
1. Install Arduino IDE
1.1 To install Arduino IDE, download the free software from its official website:
https://www.arduino.cc/en/Main/Software
1.2 Select your operating system and the Arduino version that you want to install. We recommend you to install the latest version.
1.3 Once you have downloaded it, you will have an .exe file that you will need to double click to install the software. During the installation, you will have to read and accept the Terms and Conditions, install the drivers corresponding to the Arduino boards, and select where you would like to install the software in your computer.

2. Configure IDE Arduino for the Build & Code board

Once you have installed the Arduino IDE in your computer, open the program.
2.1 Make sure that the Build&Code UNO board is correctly connected and that the BTL/USB switch is set to USB, so you can program the board with your computer. Then, connect the Build&Code UNO board to your computer with the USB cable.
2.2 Configure Arduino IDE to work with the Build&Code UNO board: open the "Tools" upper menu, click on "Board" and select the "Arduino/Genuino Uno" option.
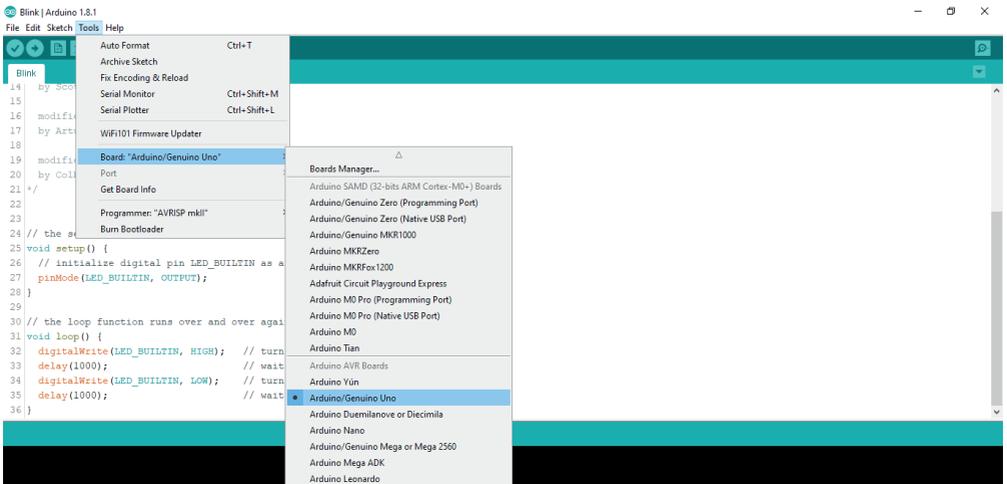
2.3 Configure the work port for the Build&Code board: open the menu "Tools", click on "Port" and select the port to which the board is connected.

Blink | Arduino 1.8.1

File Edit Sketch Tools Help

Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Serial Monitor	Ctrl+Shift+M
Serial Plotter	Ctrl+Shift+L
WiFi101 Firmware Updater	
Board: "Arduino/Genuino Uno"	>
Port: "COM13 (Arduino/Genuino Uno)"	>
Get Board Info	
Programmer: "AVRISP mkII"	>
Burn Bootloader	

Serial ports
✓ COM13 (Arduino/Genuino Uno)

```
14  by Sco
15
16  modifi
17  by Art
18
19  modifi
20  by Col
21  */
22
23
24  // the s                         t or power the board
25  void setup() {
26    // initialize digital pin LED_BUILTIN as an output.
27    pinMode(LED_BUILTIN, OUTPUT);
28  }
29
30  // the loop function runs over and over again forever
31  void loop() {
32    digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
33    delay(1000);                       // wait for a second
34    digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
35    delay(1000);                       // wait for a second
36  }
```

Note: if you are not sure to which port is the board connected, connect it and disconnect it, and check which is the port that appears and disappears from the menu.

3. Upload Arduino IDE program to the Build&Code board

When you have finished programming the Arduino IDE software for the Build&Code UNO board, click on the horizontal blue arrow indicated with the red rectangle in the next image:

Blink | Arduino 1.8.1

File Edit Sketch Tools Help

Upload

Blink
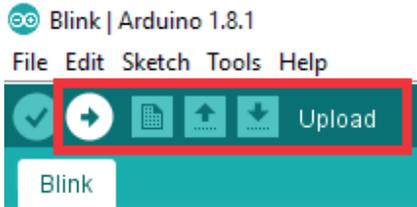
Arduino IDE will compile the program. This means it will revise that the code is in order and there are no errors.

Compiling sketch...

Then, it will upload it to the board and an "Uploading" message will appear.

Uploading...
Sketch uses 928 bytes (2%) of program storage space. Maximum
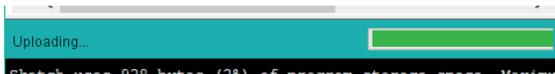
Once it has been uploaded, the message "Done uploading" will display. Now the program is on the Build&-Code board.

Done uploading.
Sketch uses 928 bytes (2%) of program storage space. Maximum
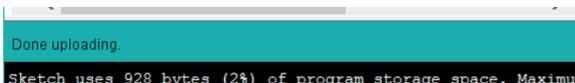
4. Arduino's programming language. The structure of the program

The structure of the Arduino programming language is very simple. It has three different parts:

- The part on which you declare the variables and call the necessary libraries (program codes made by a third party, that you will use). The variables can be used to save the values sent by a digital or analog sensor
- Setup(): is the part in charge of collecting the configuration, and the first function to be executed in the program. It's executed one time only and it's used to configure or initialize pinMode (digital pin's behavior, either as an input or an output), configuration of the serial communication and others
- Loop(): is the part that contains the program that will be executed cyclically. Is where you will write the programming logic. Therefore, this function is the core of all Arduino programs and the part that executes most of the work
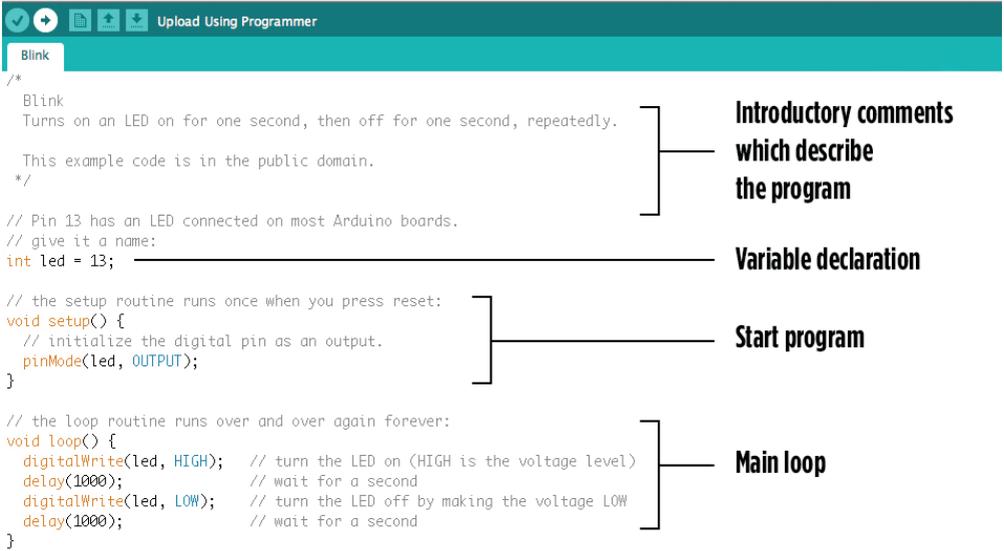

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Introductory comments which describe the program

Variable declaration

Start program

Main loop

Setup and Loop are necessary functions for the program to work.

Configure the Ebotics Build&Code UNO board for the visual programming software by blocks compatible

The visual programming software by blocks is a graphic programming software easy to learn. It's made for kids and beginners. Its interface is designed to look like a puzzle, where the user puts together different pieces (also known as "blocks") to build a program. And like in a puzzle, some pieces will fit with each other and some won't, so it's easy to know when you are doing something correct or incorrect.

1. Install the visual programming software by blocks
1.1
1.2 Select your operating system and the software version you want to install. We recommend to install the latest version.
1.3 Once you have downloaded the program, click on the .exe file to start the installation.
1.4 During the installation process you will have to read and accept the Terms and Conditions, install the drivers corresponding to the Arduino boards, and select where you would like to install the software in your computer.

Note: to install the Arduino drivers through this software go to "Menu", "Connect" and "Install Arduino driver".

2. Configure the visual programming software by blocks for the Ebotics Build&Code UNO board
Access the software.
2.1 Plug the Build&Code UNO board to the computer.

2.2 Configure the software to work with Build&Code board: select "Boards" in the upper menu and then choose "Arduino Uno".
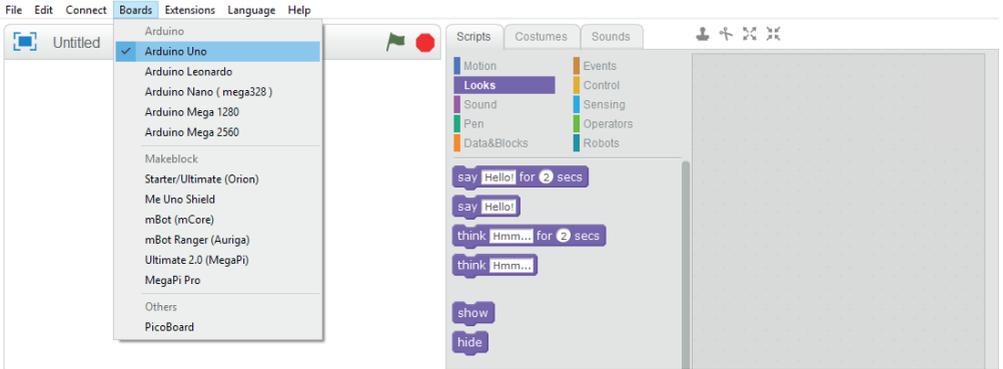
2.3 Configure your working port: open "Connect" in the upper menu, select "Serial Port" and choose the port to which the Build&Code board is connected.
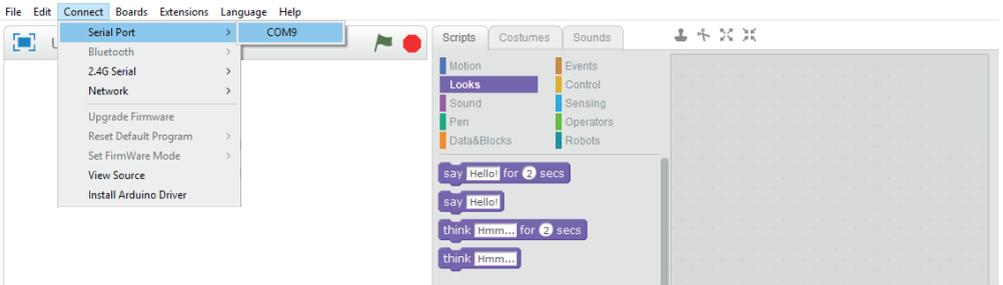
2.4 Now the software should sync with the Build&Code UNO board. To verify it check that over the upper menu appears a message that says "Serial Port Connected", and under the programming blocks categories appears a green circle.
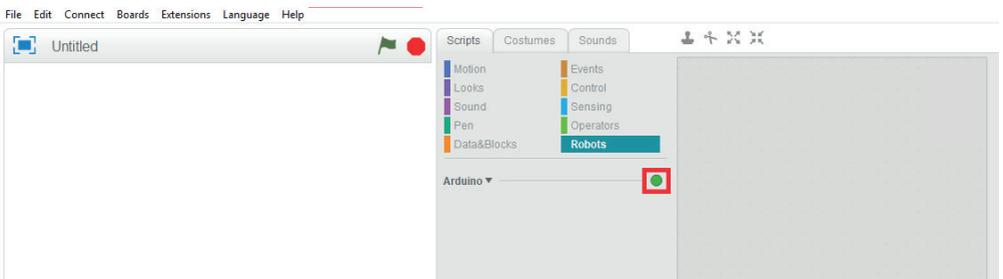
2.5 Now the Build&Code UNO board is connected with your computer.

3. Upload the software program to the Build&Code board

First of all, you will need a program to upload. In the following image you will see a program example that turns on and off the built-in LED in the Build&Code UNO board with 1 second interval. You can copy it to do this exercise.

To upload the program to the board, follow the next steps. You can check each step with the numbered images.

3.1 Go to the "Robots" category.

3.2 Choose the "Arduino Program" block, which is the first block of the program. From this block you will start building the program (copying the example or doing yours).

3.3 When the program is complete, right click on the first block ("Arduino Program").

3.4 Select "upload to arduino".
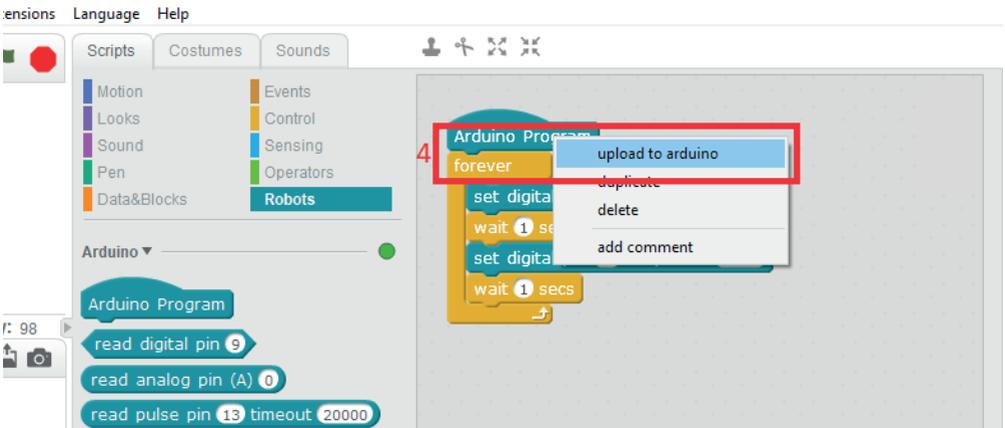
3.5 After clicking on "upload to arduino" the following window will pop out. To upload the program on the Build&Code UNO board, click on "Upload to Arduino".
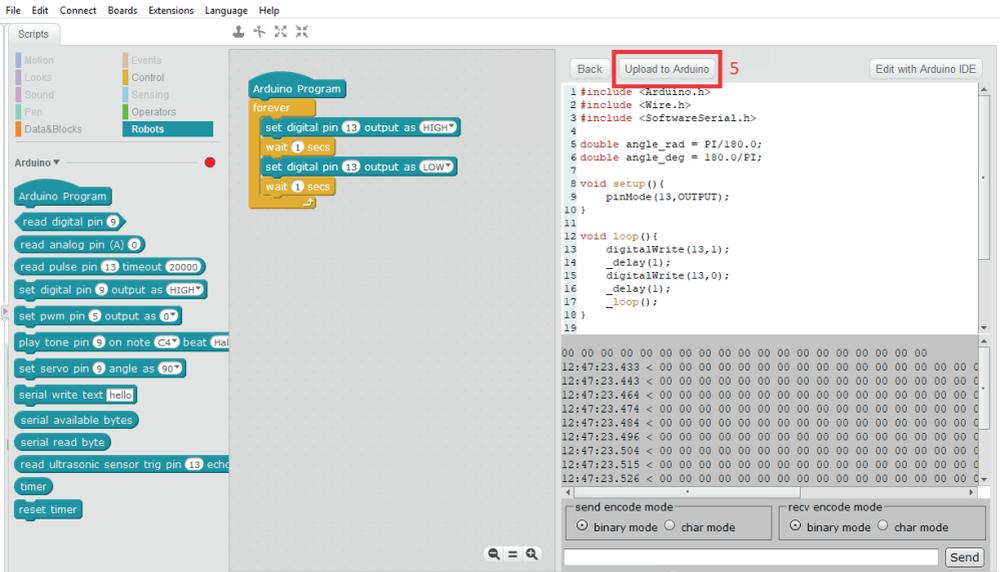
3.6 A new window will open, with the message "Uploading". Wait until it says "Upload Finish", which means that the program is already in the Build&Code UNO board.
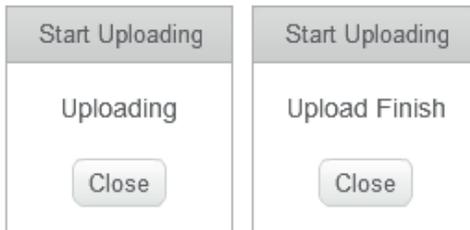
4. Visual programming language. The structure of a program

The visual programming software interface presents the following spaces:

(1) Robot or video game space

(2) Programming block categories: they are the pieces you will use to build the program for the robot or the video game

(3) Program build space: is where you will put the blocks to build the program
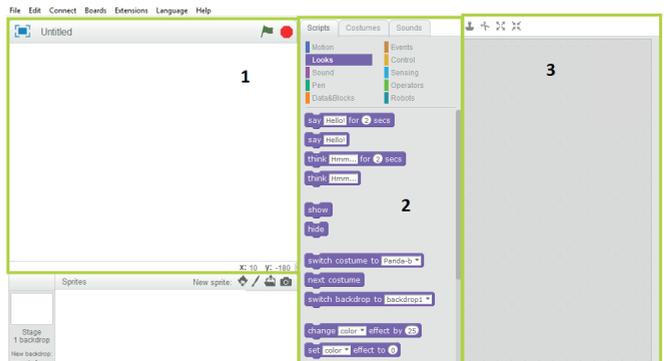
To program the robot you will have to choose the blocks from the center space (2) and build the program in the right space (3).
In this case, you should use the blocks corresponding to the "Robots" category. You can also use the "Events", "Operators" and "Control" block categories.

Configure the Ebotics Build&Code UNO board for Bitbloq
1. Configure Bitbloq
Before programming with Bitbloq it's important to check that you have everything you need for the program to function correctly on your computer.
Follow these steps:
1.1 Connect the Build&Code UNO board to the computer with the USB cable.
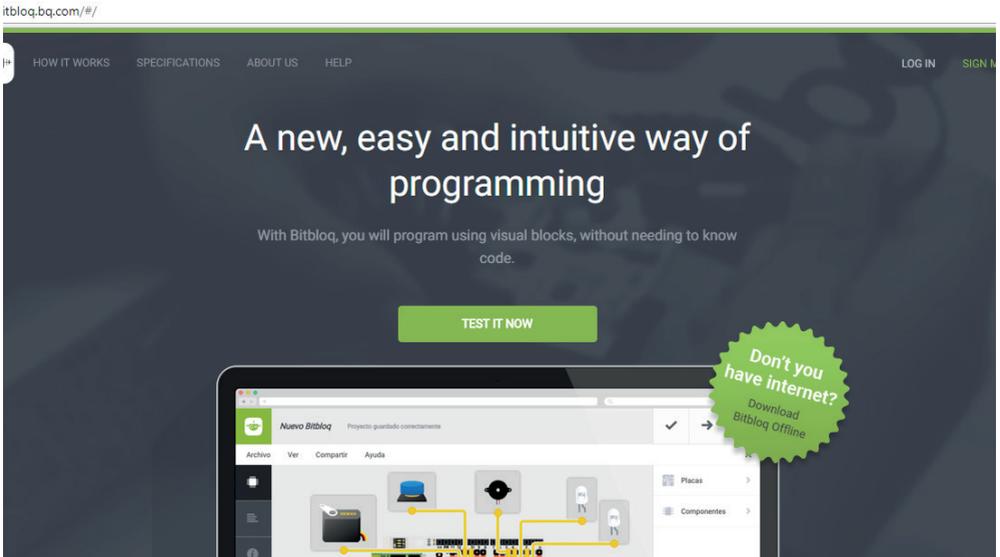
1.2 Access the Bitbloq official website http://bitbloq.bq.com/#/ using the Google Chrome browser.

1.3 If you already have an account or want to sign up, click "Enter" or "Register" and follow the steps indicated.
If you don't have an account and don't want to register, click "Test it now". Note that this way you will be working as a guest, so your project won't be saved.
If it's the first time you access, Bitbloq will offer to do a walkthrough. We recommend you to do it, since it will give you a clear idea of how the program works.

1.4 Open the "Boards" option and drag the Ebotics UNO board into the white square. If you have a different board to the one that appears in the list compatible with the UNO family, use the "Arduino UNO" board.
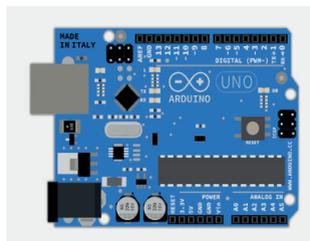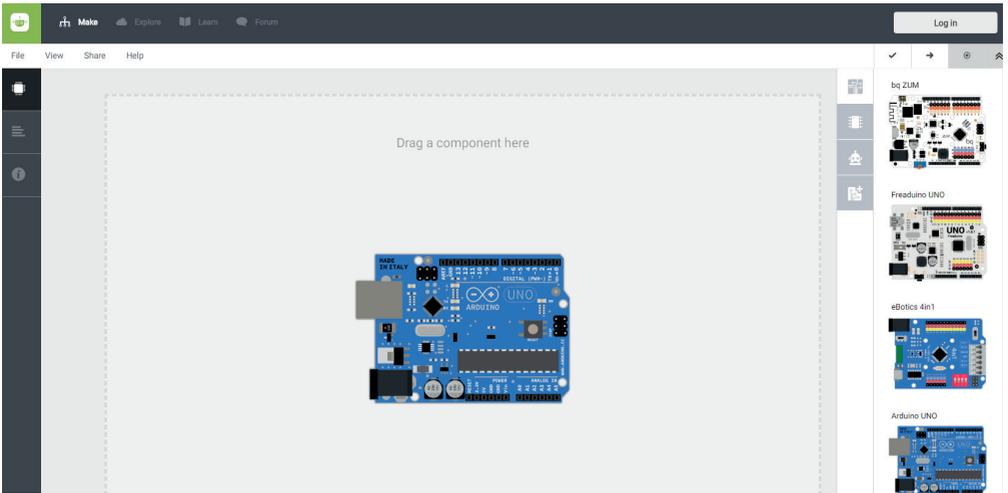
1.5 Click "Verify" and you will see how it compiles the program correctly.

1.6 Click on the "Charge" button and you will see a dialog box asking to download the Web2Board app. Select your operation system and a file will download to your computer.
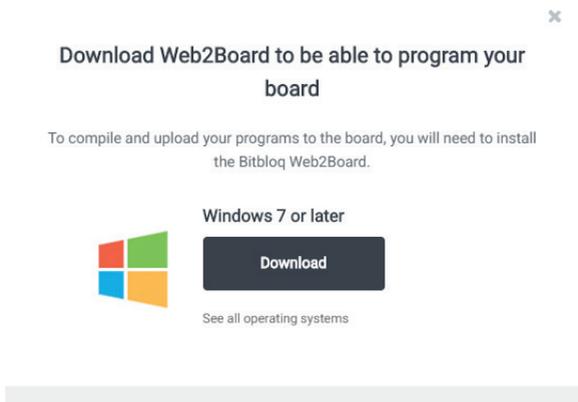
1.7 If you are using Windows, unzip the downloaded file and execute the Web2Board file.
If you are using Linux, unzip the file and execute the Web2Board file. An app will open and ask you to introduce your user password. Introduce your password and click "Intro".
1.8 Once you have installed the Web2Board program, restart your computer.

1.9 Open Bitbloq, select the Ebotics UNO board and click "Charge". The program will start to download to your board. After a few moments, you will see a message indicating that the procedure worked correctly: "Program uploaded successfully".

2. Bitbloq programming language. The structure of a program
The Bitbloq interface is divided in 3 main tabs, located on the left:
• Hardware: where you will indicate which board, sensors and actuators you will use
• Software: where you will program your project
• Project information: where you will add the project information

To make sure that the Buil&Code UNO board is correctly connected to the computer, let's do a **first project:** turn on a LED.

2.1 Hardware tab: connect the electronics
In this tab you will indicate to Bitbloq which hardware are you using. First, select the Ebotics UNO board. Then, indicate which components are you using and where you have connected them.
Click "Components" and drag the LED into the yellow rectangle. If you click and drag the component's circle, you will be able to connect its cable to the digital pin "13". Last, connect the LED to the board. Remember to connect each pin with its color.

2.2 The software tab: program your project
Now that you have connected the software, it's time to do the project: turn on a LED.
Look at the programming space. It's divided in 3 parts:
• Global variables and functions: it allows to declare accessible variables from any part of the program, and to create new functions
• Initial instructions (Setup): it's executed at the beginning of the program. It's used to configure the pins behavior and other actions that we want to execute at the beginning, and one time only
• Main loop (Loop): is the program's main loop, and the one you will use more often. Everything inside of it will be executed indefinitely. This is why a Bitbloq or Arduino program never ends.
Open the "Components" tab and drag the "Switch on the LED (name)" block to the "Loop" section. Bitbloq will only show the blocks corresponding to the components you have connected in the Hardware tab. If you need generic blocks, click the "Advanced" option.

The program is ready to use. Click the "Verify" button so that Bitbloq checks that the program works correctly. If the program is correct, click "Charge".
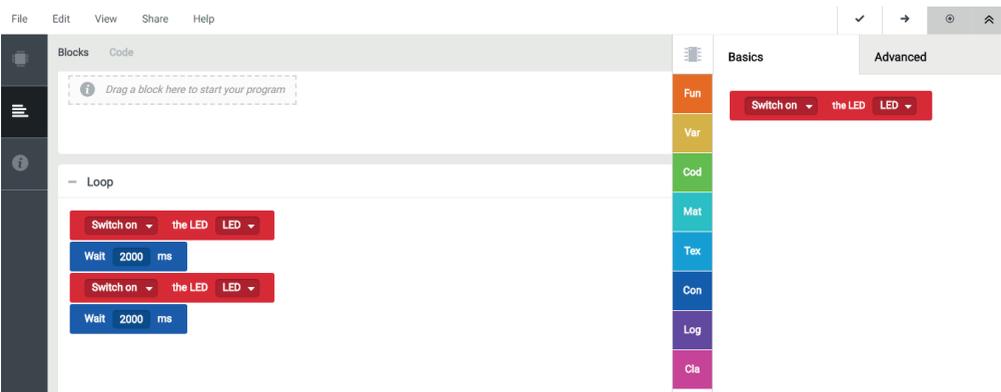
The program will install in the board and, after the message "Program successfully uploaded", you will see how the LED turns on.

Now that you have configured the Build&Code UNO board for Arduino, Bitbloq or the visual programming software by blocks do the projects to start discovering what you can do with the Arm Robot kit.

Ebotics

Arm Robot

Electronic and programming kit

Do it yourself

ebotics.com