ARDUINO

workflow

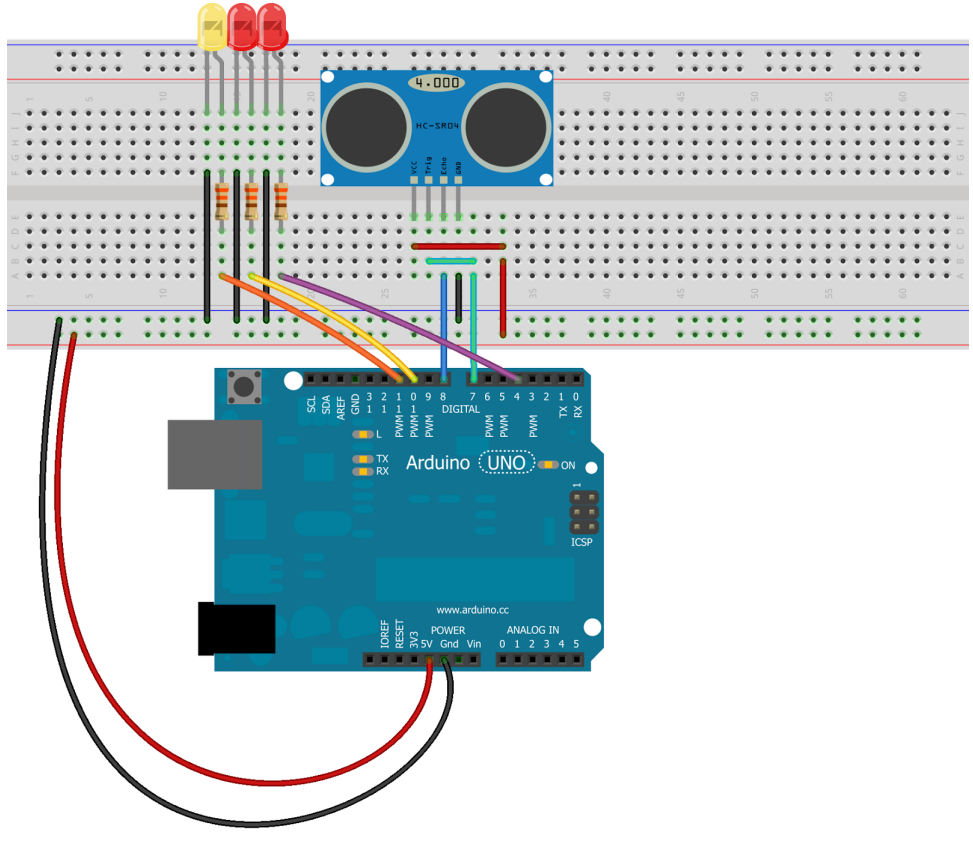
- An arduino board is a (little) computer
- it can be only receive, process and send electrical signals
- exactly like a traditional computer

`input` AND `output` DEVICES CAN BE CONNECTED TO A COMPUTER

- input: mouse, trackpad, keyboard
- output: monitor, printer

ALSO THE ELECTROACOUSTIC CHAIN OFFERS THE SAME MODEL

FOR MICROCONTROLLERS IT IS MORE COMMONLY REFERRED TO AS **sensors** (INPUT) AND **actuators** (OUTPUT)



SYSTEMS INVOLVING MICROCONTROLLERS, SENSORS AND ACTUATORS BELONG TO THE AREA OF

physical computing

SOFTWARE

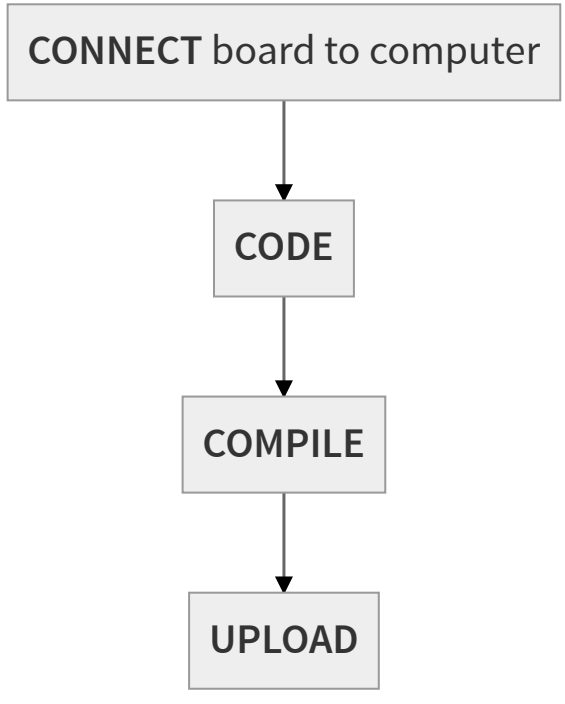
ARDUINO IS NOT DESIGNED *A PRIORI* TO PERFORM A GIVEN TASK

A `program` MUST BE UPLOADED INTO THE BOARD

A PROGRAM IS A SEQUENCE OF INSTRUCTIONS THAT CONTRIBUTE TO THE RESOLUTION OF A PROBLEM, OR MORE GENERALLY, TO THE EXECUTION OF A **task**

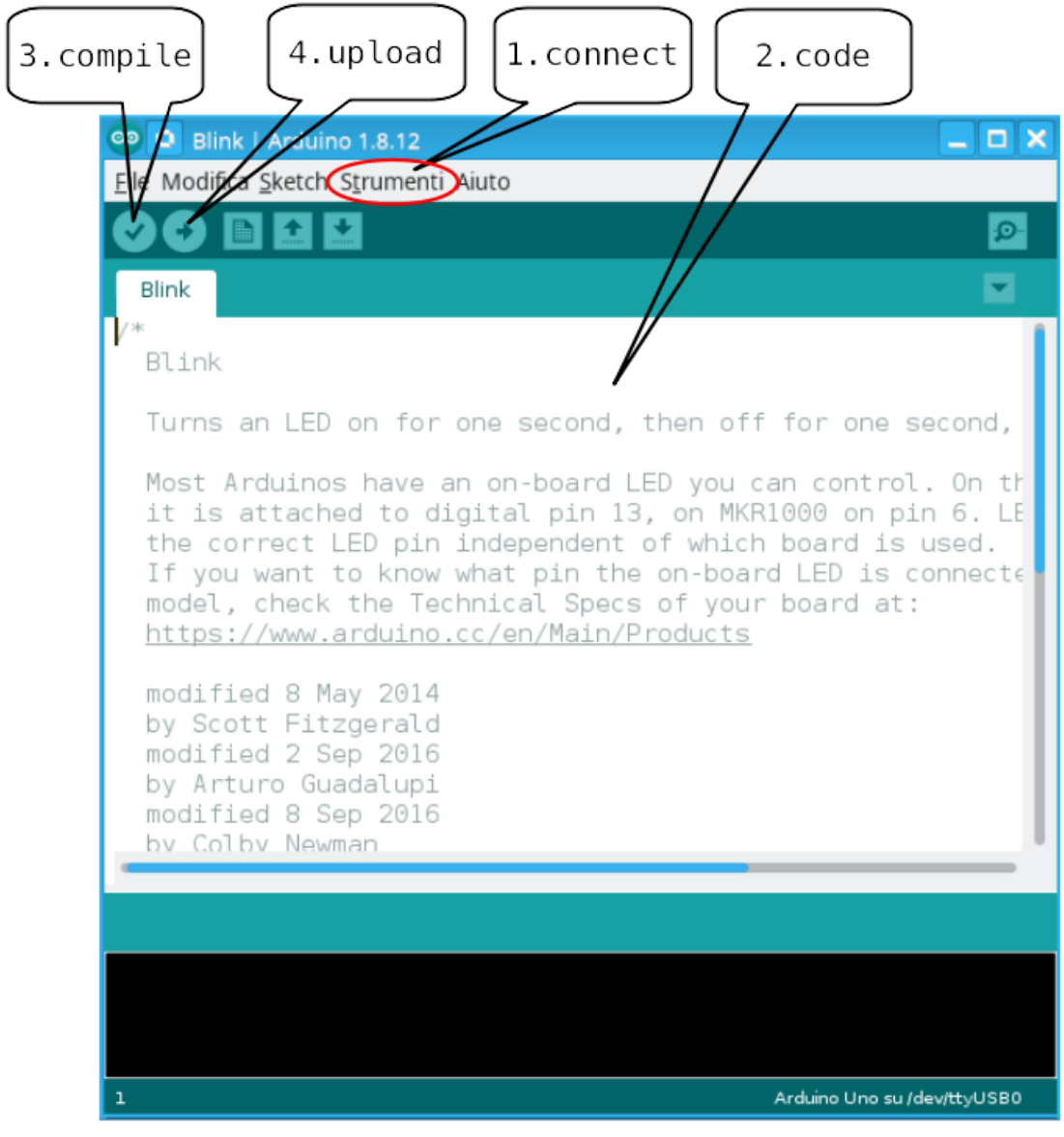
ARDUINO RUNS PROGRAMS THAT ARE **uploaded** INTO ITS MICROCONTROLLER

THE STEPS TO UPLOAD A PROGRAM ARE:



THESE STEPS CAN BE REALIZED BY

arduino IDE


3.compile
4.upload
1.connect
2.code
Blink | Arduino 1.8.12
File Modifica Sketch Strumenti Aiuto
Blink
/*
Blink
Turns an LED on for one second, then off for one second,
Most Arduinos have an on-board LED you can control. On th
it is attached to digital pin 13, on MKR1000 on pin 6. LE
the correct LED pin independent of which board is used.
If you want to know what pin the on-board LED is connecte
model, check the Technical Specs of your board at:
https://www.arduino.cc/en/Main/Products
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
1
Arduino Uno su /dev/ttyUSB0

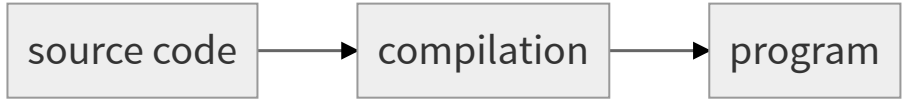
THE LANGUAGE OF ARDUINO

- the only language arduino speaks is based on 2-symbols alphabet: 0 and 1
- these symbols belongs to binary numeric system
- long sequences of these symbols shape instructions for microcontroller

- every internal component is configured by binary instructions
- if I would set pin number 8 as an OUTPUT pin, I could code: 00001000 00000001
- this language is called `machine code`

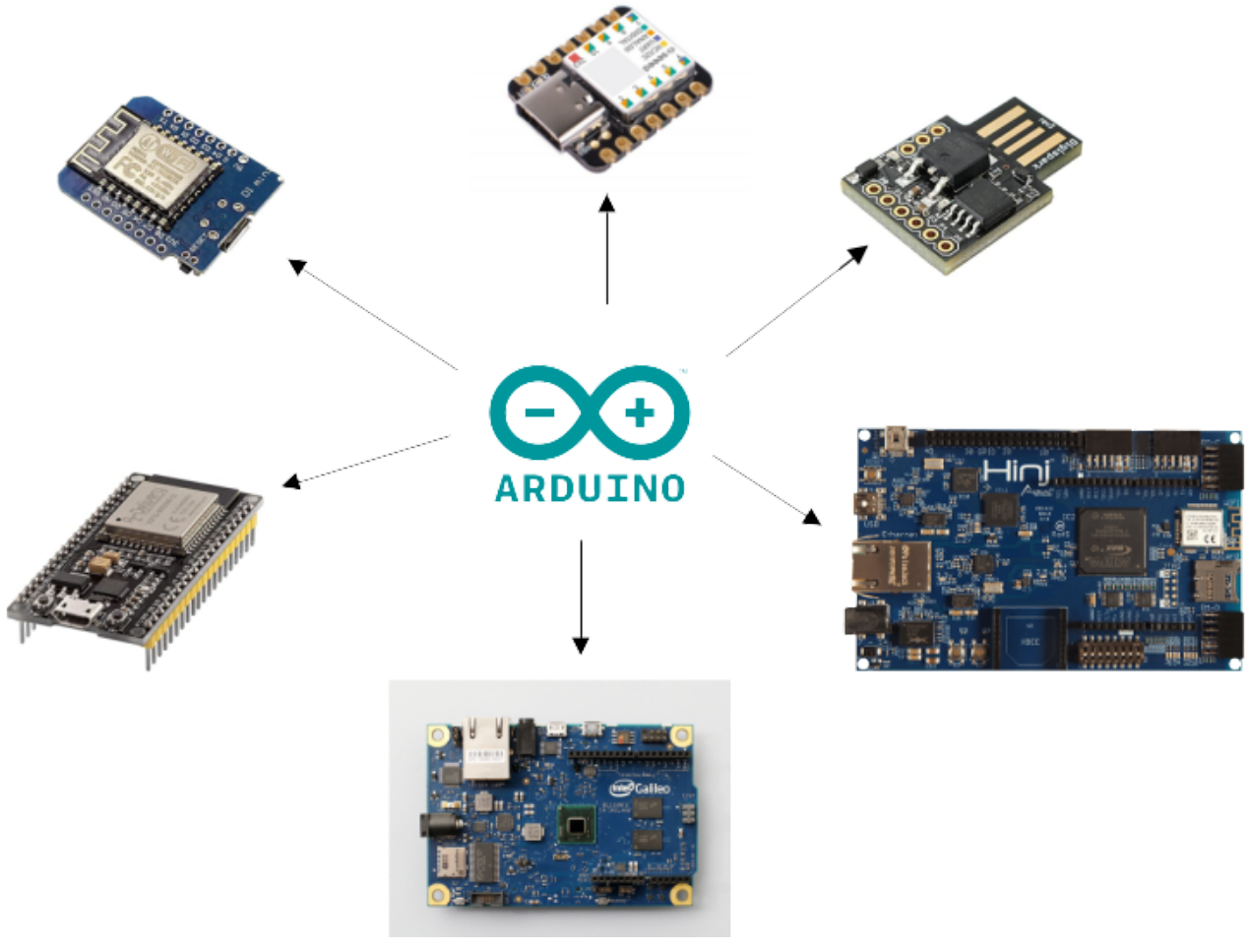
- theorically I could program the board using machine code
- but programming arduino (or other computer) using that is a complex, tiring and boring task
- *ATMega328* (arduino microcontroller) instruction manual is 300 pages long (and very DENSE pages!)

- a programming language is generally used
- you write the `source code` (called `sketch` in arduino) with this language
- then you pass the source code to a software called `compiler` that translates it in machine code



A PROGRAMMING LANGUAGE IS CLOSER TO THE NATURAL LANGUAGE AND MORE ABSTRACT THEN HARDWARE

THIS ABSTRACTION ALLOWS TO `generalize`, THAT IS COMPILE THE SAME SOURCE CODE FOR DIFFERENT MICROCONTROLLERS


ARDUINO
Hinj
Intel Galileo

THE PROGRAMMING LANGUAGE USED IN ARDUINO IS C/C++, WITH A SPECIFIED `library` WRITTEN TO FACILITATE CODING

C PROGRAMMING LANGUAGE IS COMPOSED BY A SET OF ABOUT 30 WORDS, THEN IT'S CONCISE

SKETCH

```
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

COMPILED SKETCH

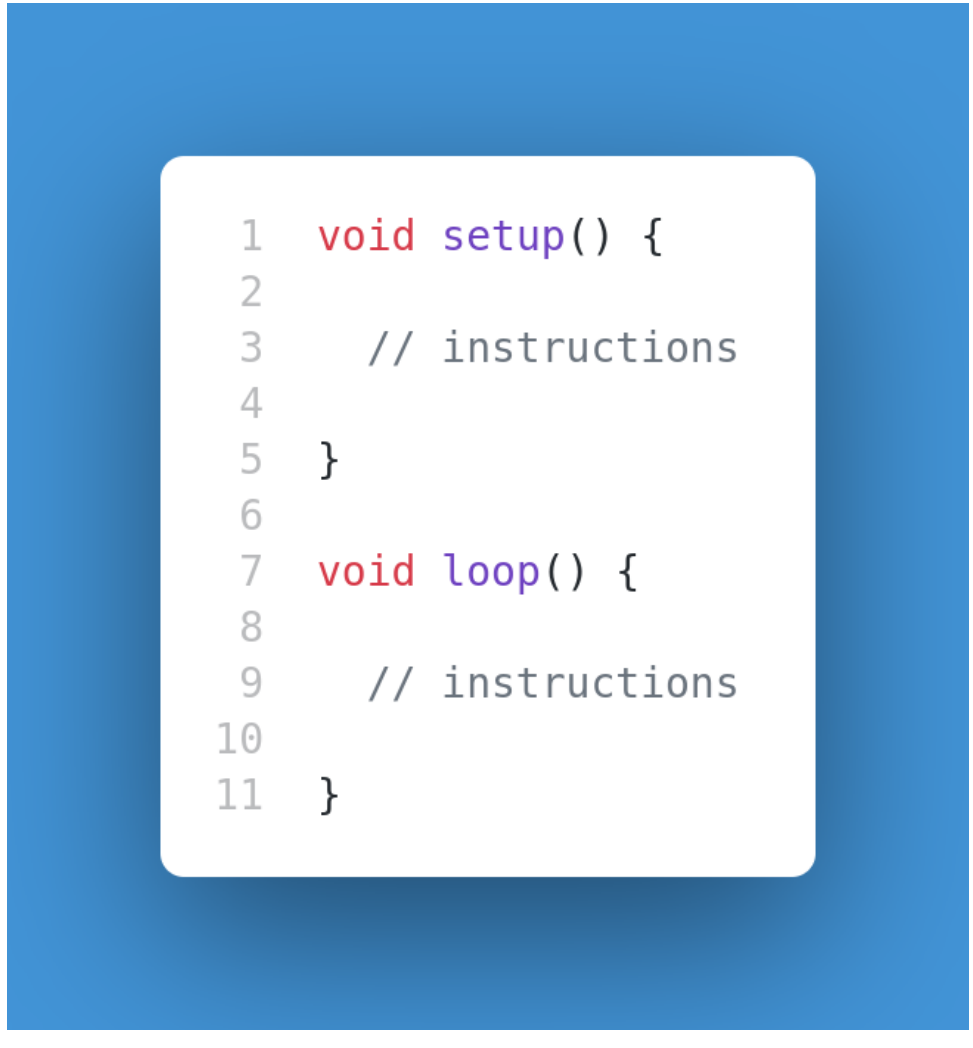
```
:
100000000C945C000C946E000C946E000C946E00CA
100010000C946E000C946E000C946E000C946E00A8
100020000C946E000C946E000C946E000C946E0098
100030000C946E000C946E000C946E000C946E0088
100040000C9413010C946E000C946E000C946E00D2
100050000C946E000C946E000C946E000C946E0068
100060000C946E000C946E0000000000002400270029
100070002A0000000000250028002B0004040404CE
10008000040404040202020202020303030303034 2
1000900001020408102040800102040810200102 1F
1000A0000408102000000008000201000003040 7FB
1000B000000000000000000011241FBECFEFD8E0B8
1000C000DEBFCDBF21E0A0E0B1E001C01D92A930AC
1000D000B207E1F70E945D010C94CC010C94000082
1000E000E1EBF0E02491EDE9F0E09491E9E8F0E053
1000F000E491EE23C9F0222339F0233001F1A8F472
10010000213019F1223029F1F0E0EE0FFF1FEE58F7
10011000FF4FA591B4912FB7F894EC91811126C0AF
1001200090959E239C932FBF08952730A9F02830E7
10013000C9F0243049F7209180002F7D03C0209121
1001400080002F7720938000DFCF24B52F7724BD48
10015000DBCF24B52F7DFBCF2091B0002F772093EC
10016000B000D2CF2091B0002F7DF9CF9E2BDACFF7
100170003FB7F894809105019091060 1A091070185
10018000B091080126B5A89B05C02F3F19F0019634
10019000A11DB11D3FBFBA2FA92F982F8827BC01E1
1001A000CD01620F711D811D911D42E0660F771F09
1001B0000881F991F4A95D1F708958F929F92AF9209
1001C000BF92CF92DF92EF92FF920E94B8004B0154
1001D0005C0188EEC82E83E0D82EE12CF12C0E9421
```

```
:
1001E000B800681979098A099B09683E734081053E
1001F0009105A8F321E0C21AD108E108F10888EEC0
10020000880E83E0981EA11CB11CC114D104E10426
10021000F10429F7FF90EF90DF90CF90BF90AF905F
100220009F908F9008951F920F920FB60F921124F6
100230002F933F938F939F93AF93BF93809101012F
1002400090910201A0910301B0910401309100014D
1002500023E0230F2D3758F50196A11DB11D2093E2
10026000000180930101909302 01A0930301B093D8
10027000040180910501909106 01A0910701B091C0
10028000080101 96A11DB11D8093050190930601FF
10029000A0930701B0930801BF91AF919F918F91F7
1002A0003F912F910F900FBE0F901F90189526E849
1002B000230F0296A11DB11DD2CF789484B5826020
1002C00084BD84B5816084BD85B5826085BD85B5FA
1002D000816085BD80916E00816080936E00109278
1002E000810080918100826080938100800918100F3
1002F000816080938100809180008160809380008 4
100300008091B10084608093B1008091B0008160E1
100310008093B00080917A00846080937A0080910D
100320007A00826080937A0080917A008160809365
100330007A0080917A00806880937A001092C100E0
10034000EDE9F0E02491E9E8F0E08491882399F068
1003500090E0880F991FFC01E859FF4FA591B491D7
10036000FC01EE58FF4F859194918FB7F894EC9172
10037000E22BEC938FBFC0E0D0E081E00E947000E0
1003800019 0209746
:00000001FF
```

sketch ANATOMY

- Each sketch consists of the **definition** of *at least* two functions (instruction blocks), which are ** always ** present:
  - **setup()**: defines preliminary operations (pin configuration, serial port, and so on…)
  - **loop()**: run programmed task, repeatedly, until the board is powered on

SKETCH SKELETON



WHEN ARDUINO TURN ON, IT EXECUTES SETUP() FUNCTION (ONLY ONCE), THEN, `repeatedly` EXECUTES THE LOOP() FUNCTION

THE FIRST SKETCH

TURN ON THE BUILTIN LED

TASK

- write a program the turn on the led for one second, then turn off the led for one second and start again this task

OVERVIEW

- builtin led of arduino Uno is connected to pin 13
- we will set pin 13 as OUTPUT
- …and then we will apply a voltage on that pin
- a HIGH voltage value will turn on the led, a LOW value will turn it off

...EXPRESS THE ALGORITHM IN NATURAL LANGUAGE

```
/*
    blink naba version 0.1
    2021 May
*/
void setup()
{
    // set pin 13 as OUTPUT
}

void loop()
{
    // set voltage value to HIGH (turn on the led)
    // wait for 1 second
    // set voltage value to HIGH (turn on the led)
    // wait for 1 second
}
```

IN A C POROGRAM:

- a line preceded by double slash (`//`) is a comment
- a block of code (multi line) preceded by `/*` and followed by `*/` is a comment
- a comment is ignored by compiler; it's just a note used by developer to describe the code

```
// I'm a single line comment
/*
    I'm a
    multi-line
    comment
*/
```

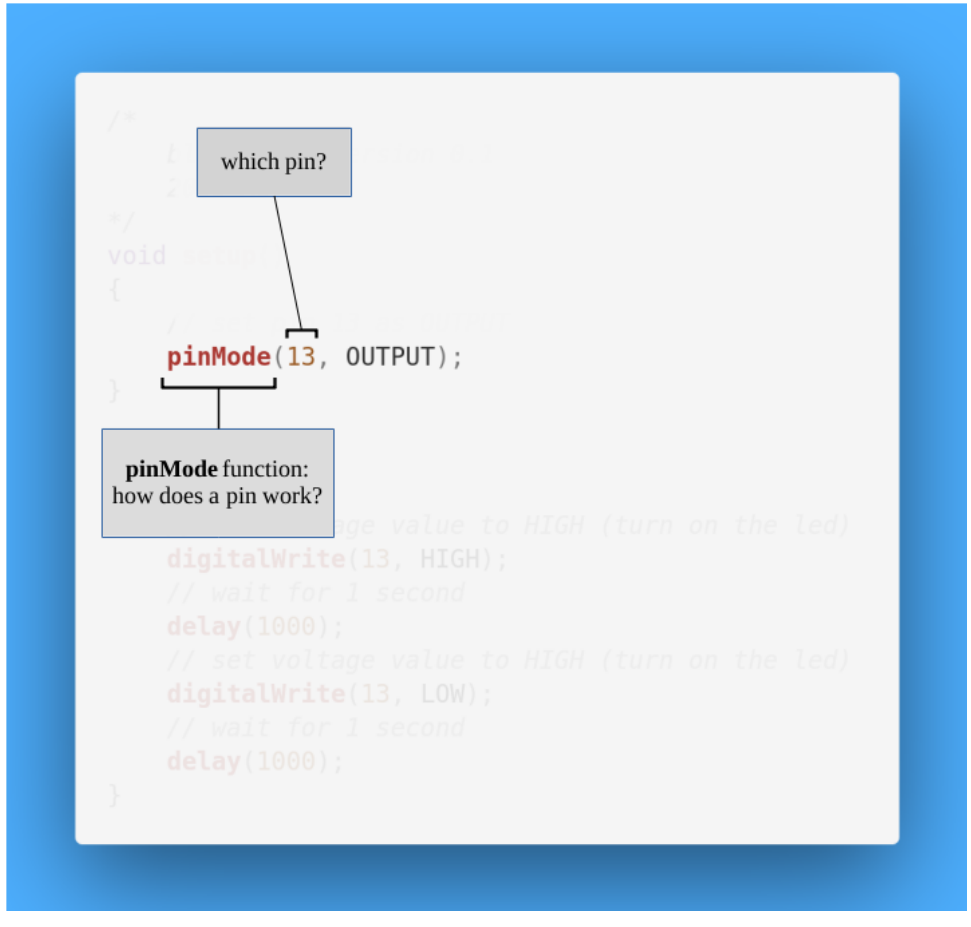
SKETCH IN C

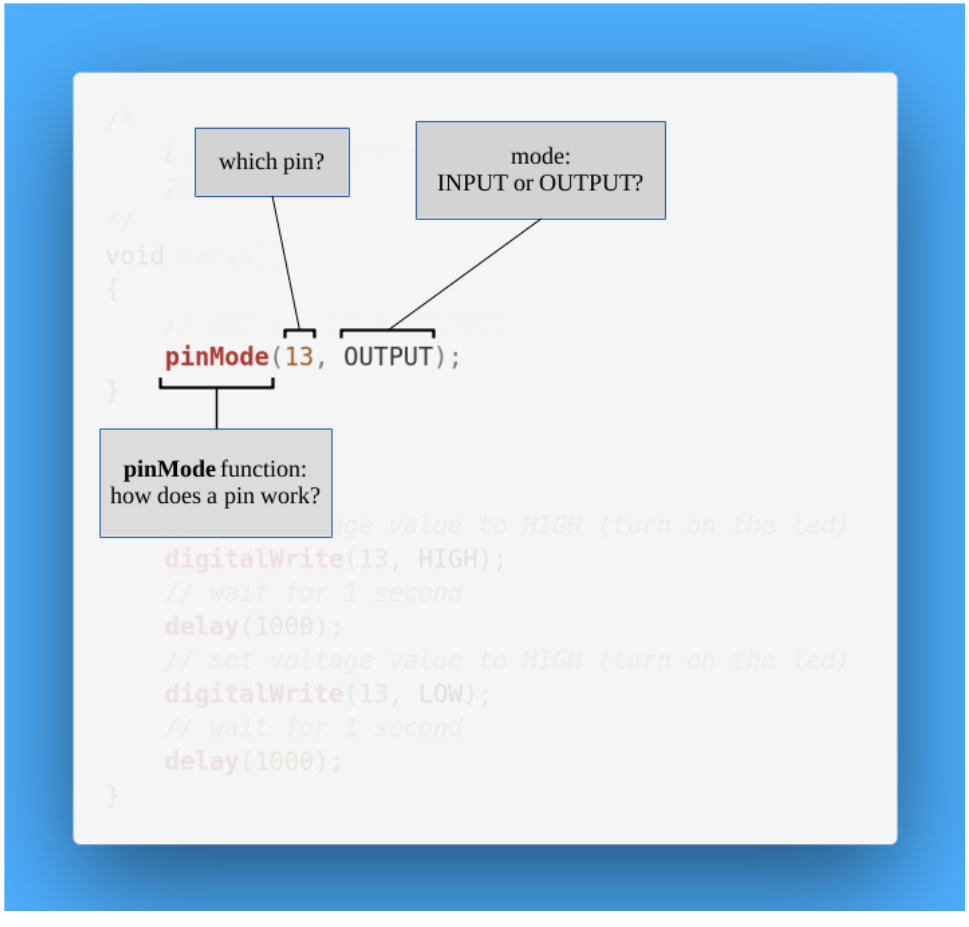
```
/*
    blink naba version 0.1
    2021 May
*/
void setup()
{
    // set pin 13 as OUTPUT
    pinMode(13, OUTPUT);
}

void loop()
{
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, HIGH);
    // wait for 1 second
    delay(1000);
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, LOW);
    // wait for 1 second
    delay(1000);
}
```


/*
blink naba version 0.1
2021 May
*/
void setup()
{
// set pin 13 as OUTPUT
pinMode(13, OUTPUT);
}
void loop()
{
// set voltage value to HIGH (turn on the led)
digitalWrite(13, HIGH);
// wait for 1 second
delay(1000);
// set voltage value to HIGH (turn on the led)
digitalWrite(13, LOW);
// wait for 1 second
delay(1000);
}


pinMode(13, OUTPUT);
pinMode function:
how does a pin work?

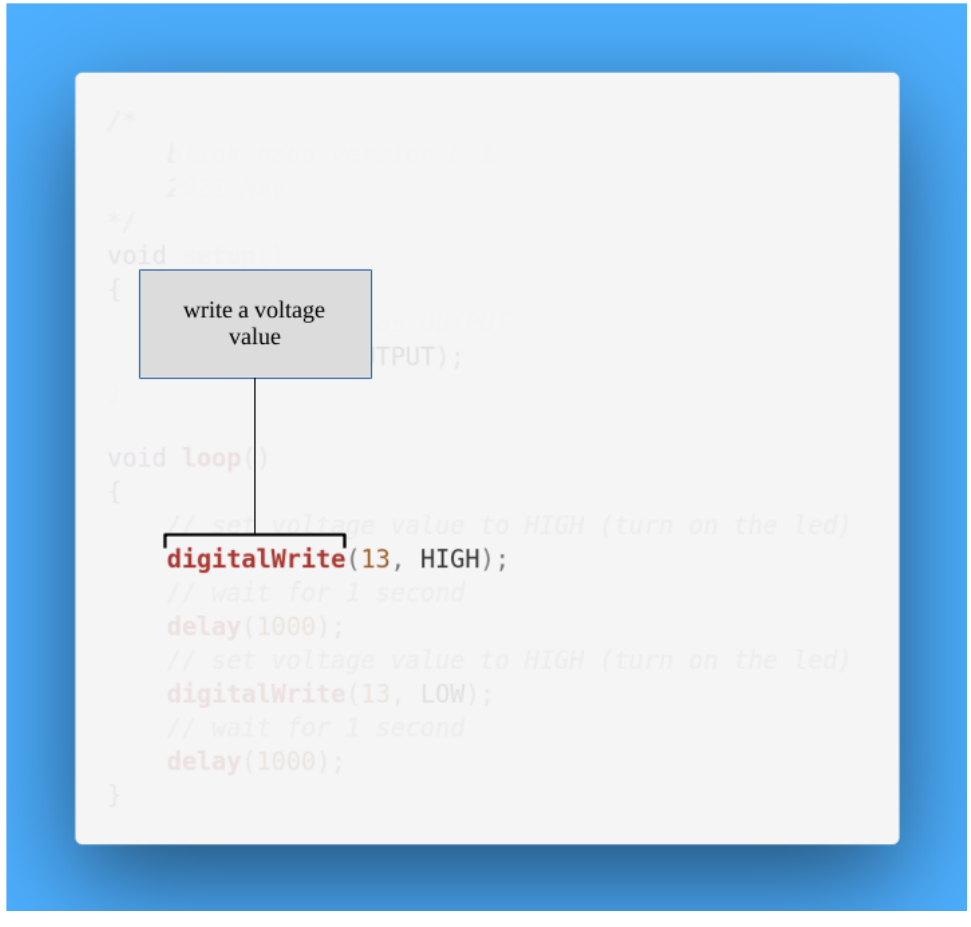

which pin?
pinMode(13, OUTPUT);
pinMode function:
how does a pin work?

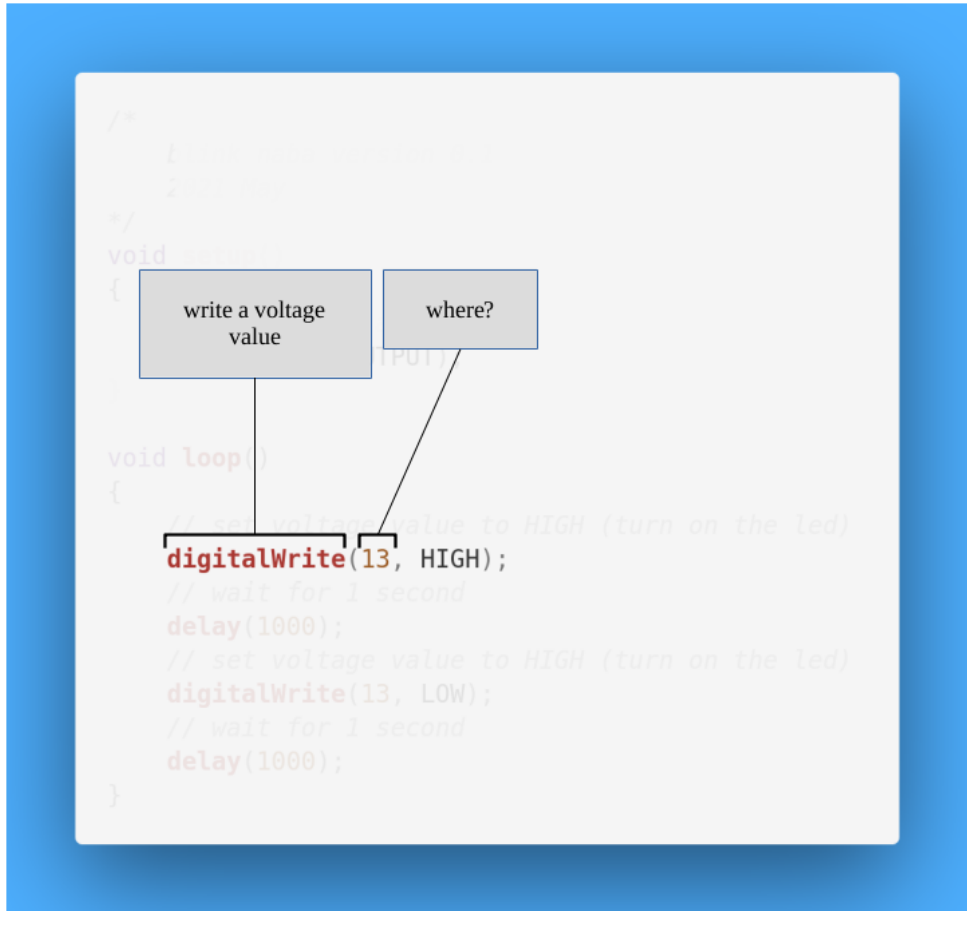

which pin?
mode:
INPUT or OUTPUT?
pinMode(13, OUTPUT);
pinMode function:
how does a pin work?

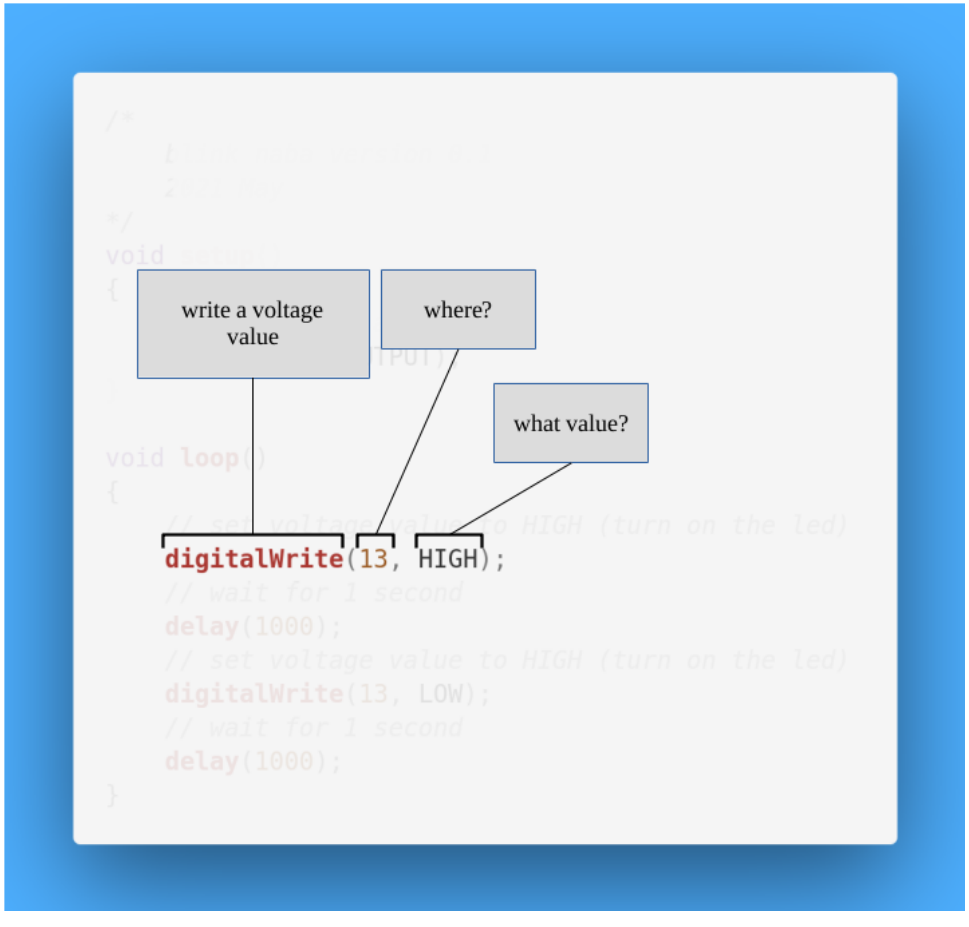
```
/*
    blink naba version 0.1
    2021 May
*/
void setup()
{
    // set pin 13 as OUTPUT
    pinMode(13, OUTPUT);
}

void loop()
{
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, HIGH);
    // wait for 1 second
    delay(1000);
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, LOW);
    // wait for 1 second
    delay(1000);
}
```


/*
Blink
Turns on an LED on for one second,
then off for one second, repeatedly.
Most Arduinos have an on-board LED you can control. On the Uno and
Leonardo, it is attached to digital pin 13. If you're unsure what
pin the on-board LED is connected to on your Arduino model, check
the documentation at http://www.arduino.cc
This example code is in the public domain.
modified 8 May 2014
by Scott Fitzgerald
*/
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}
void loop() {
// set the voltage value to HIGH (turn on the led)
digitalWrite(13, HIGH);
// wait for 1 second
delay(1000);
// set the voltage value to HIGH (turn on the led)
digitalWrite(13, LOW);
// wait for 1 second
delay(1000);
}


write a voltage value
digitalWrite(13, HIGH);


write a voltage value
where?
digitalWrite(13, HIGH);


write a voltage value
where?
what value?
digitalWrite(13, HIGH);

```
/*
    blink naba version 0.1
    2021 May
*/
void setup()
{
    // set pin 13 as OUTPUT
    pinMode(13, OUTPUT);
}

void loop()
{
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, HIGH);
    // wait for 1 second
    delay(1000);
    // set voltage value to HIGH (turn on the led)
    digitalWrite(13, LOW);
    // wait for 1 second
    delay(1000);
}
```

WHAT ARDUINO DOES