

Enseignes et afficheurs à LED

Programmation en C-Arduino

Programmation en C-Arduino

Pierre-Yves Rochat

Programmation en C-Arduino

- Les 3 significations du mot Arduino
- Structure d'un programme
- Entrées-sorties
- Gestion du temps

Significations du mot Arduino

Significations du mot Arduino

- L'Arduino a participé à rendre populaires les microcontrôleurs.
- Qu'est-ce qui se cache derrière ce nom ?

Significations du mot Arduino

- L'Arduino a participé à rendre populaires les microcontrôleurs.
- Qu'est-ce qui se cache derrière ce nom ?
 - une carte à microcontrôleur

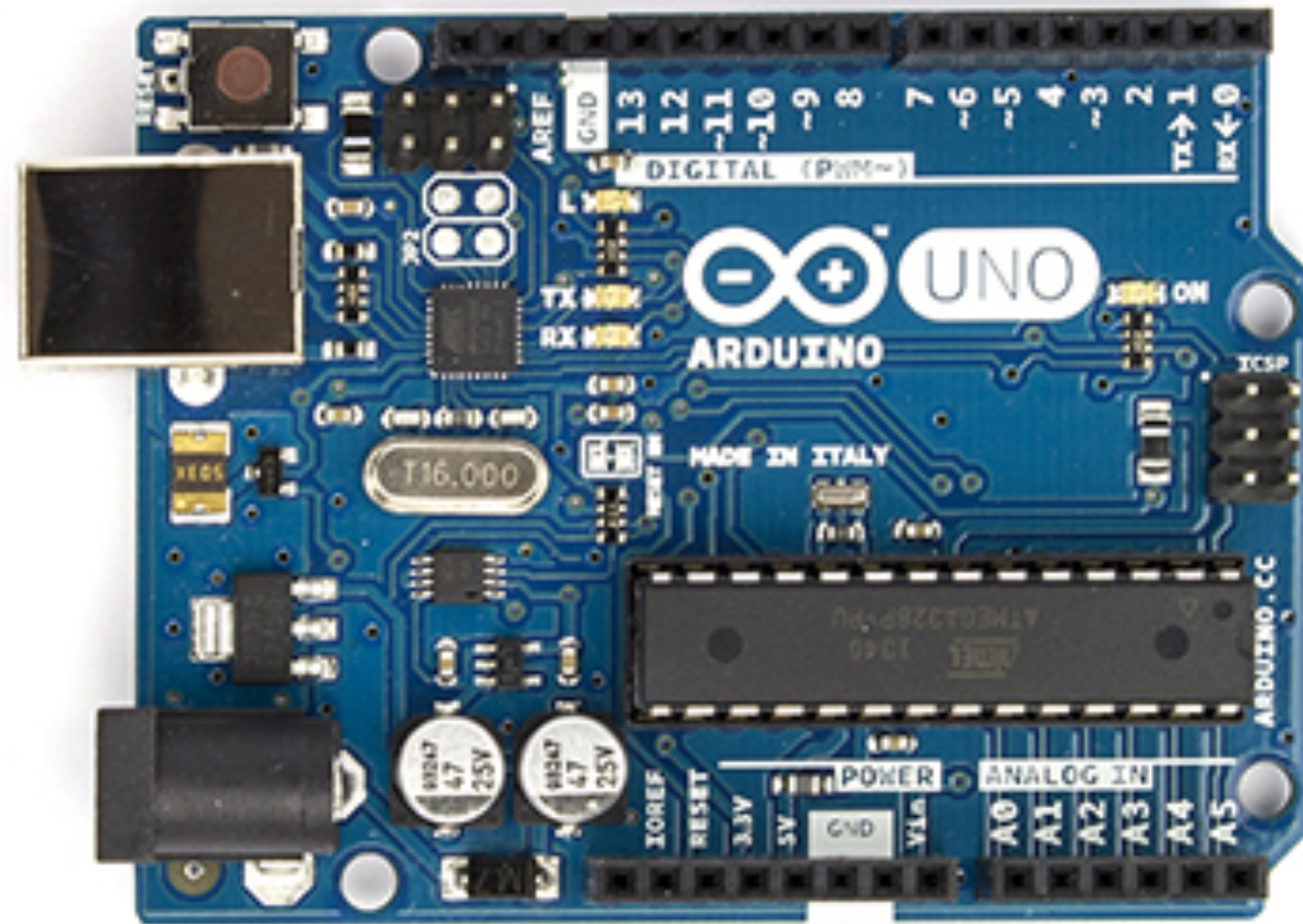
Significations du mot Arduino

- L'Arduino a participé à rendre populaires les microcontrôleurs.
- Qu'est-ce qui se cache derrière ce nom ?
 - une carte à microcontrôleur
 - un environnement de développement

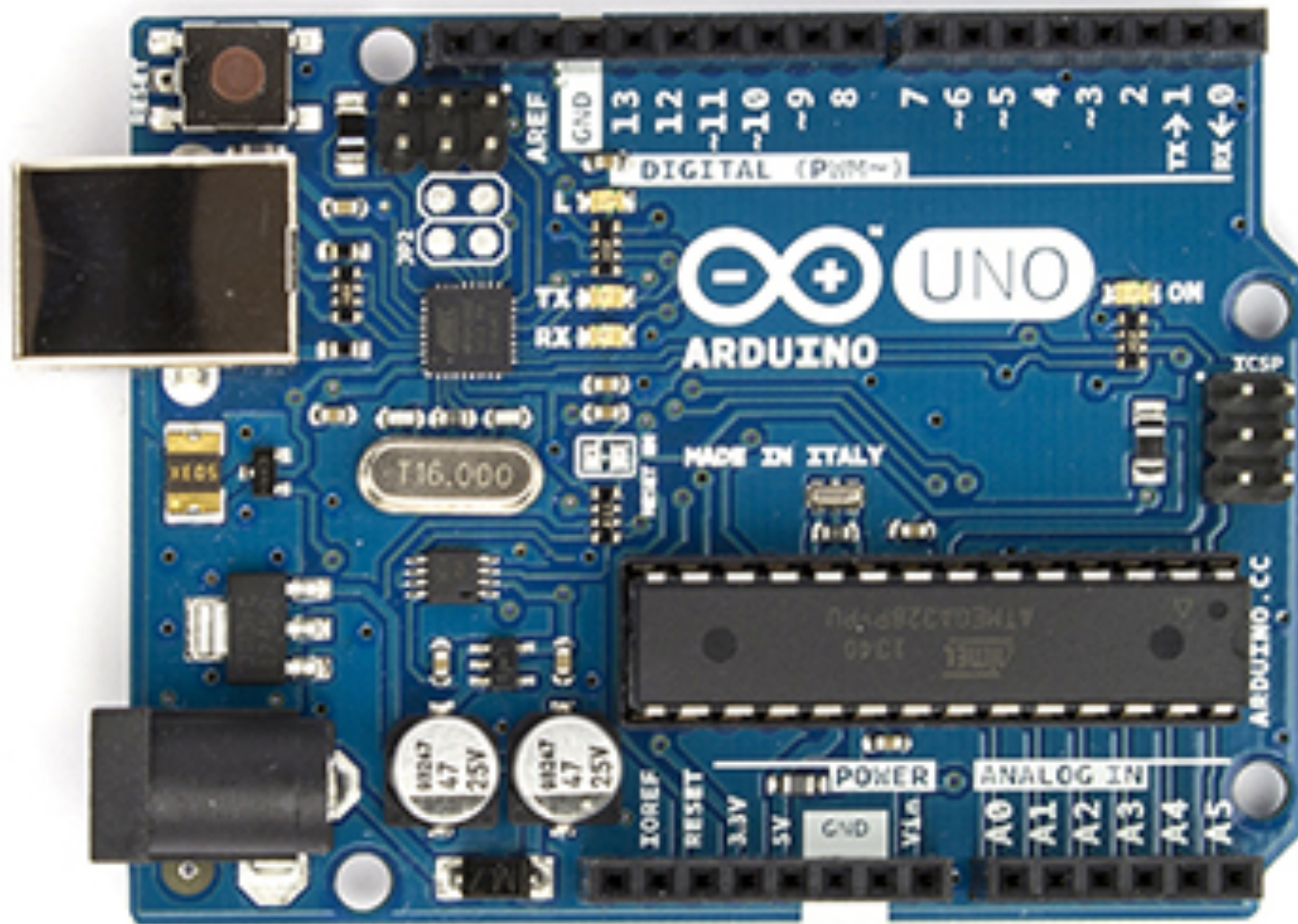
Significations du mot Arduino

- L'Arduino a participé à rendre populaires les microcontrôleurs.
- Qu'est-ce qui se cache derrière ce nom ?
 - une carte à microcontrôleur
 - un environnement de développement
 - une librairie pour microcontrôleurs (un ensemble de procédures)

Les cartes Arduino

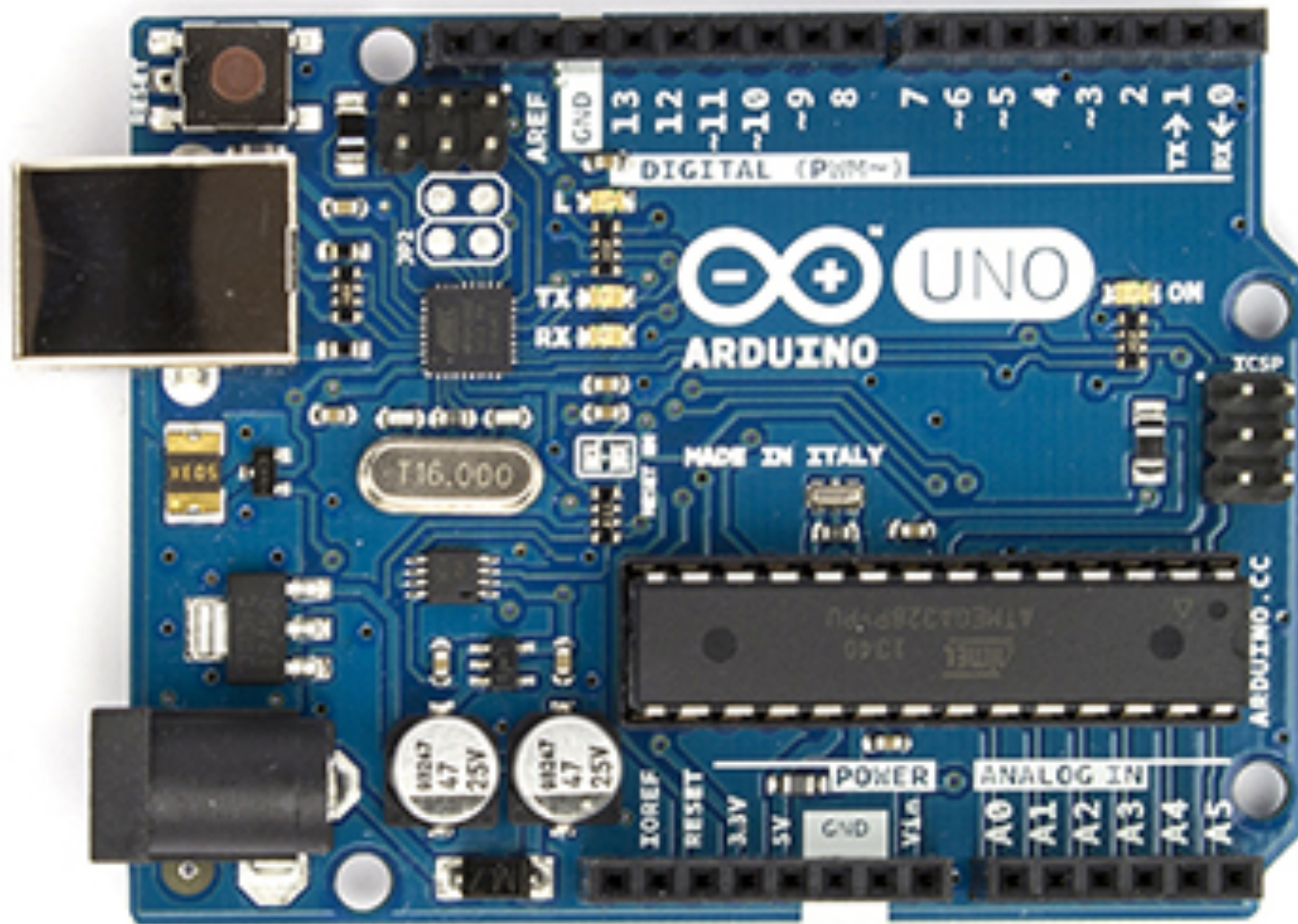


Les cartes Arduino



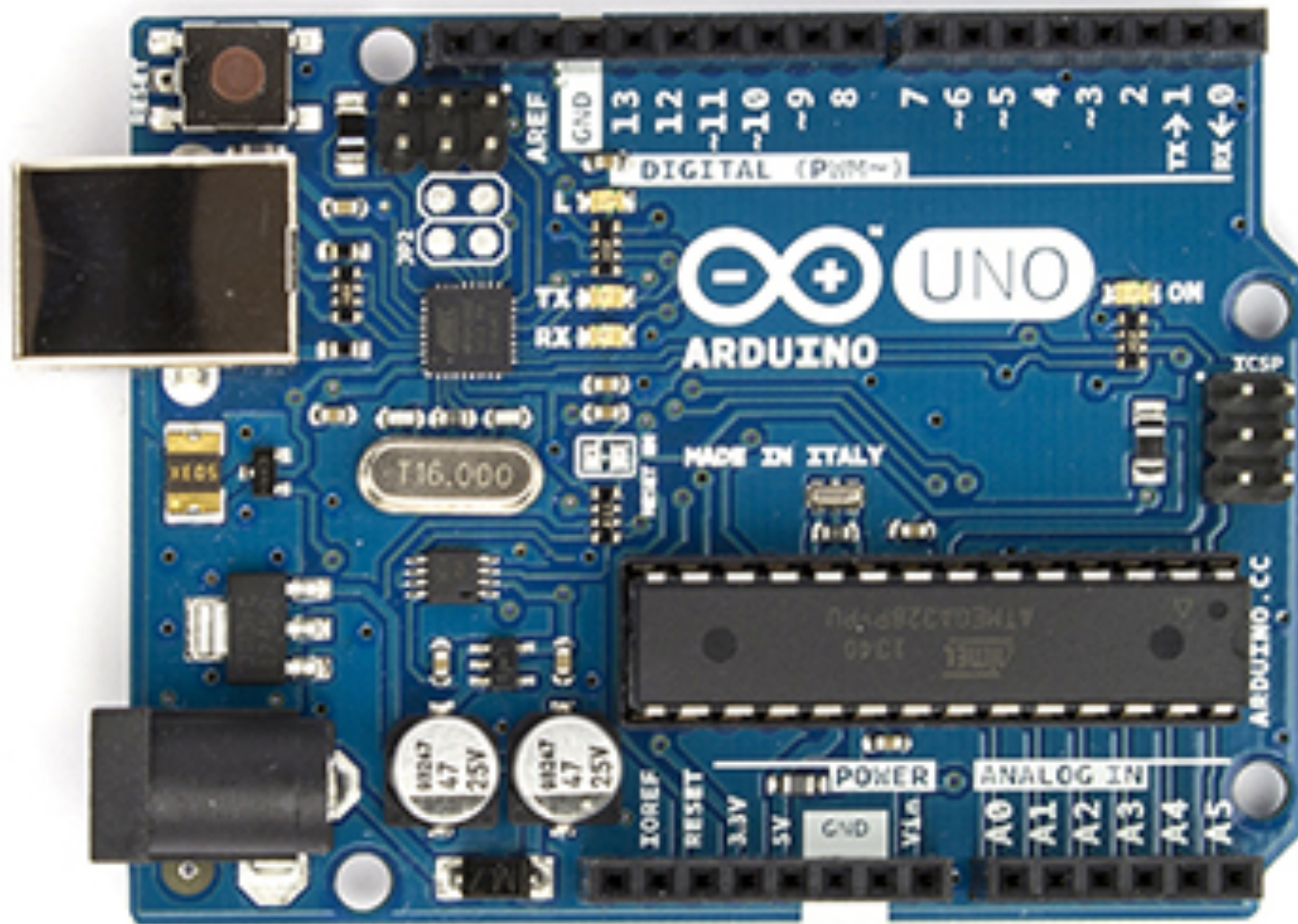
- AVR (ATmega328)
- Connecteur normalisé
- USB (liaison série)

Les cartes Arduino



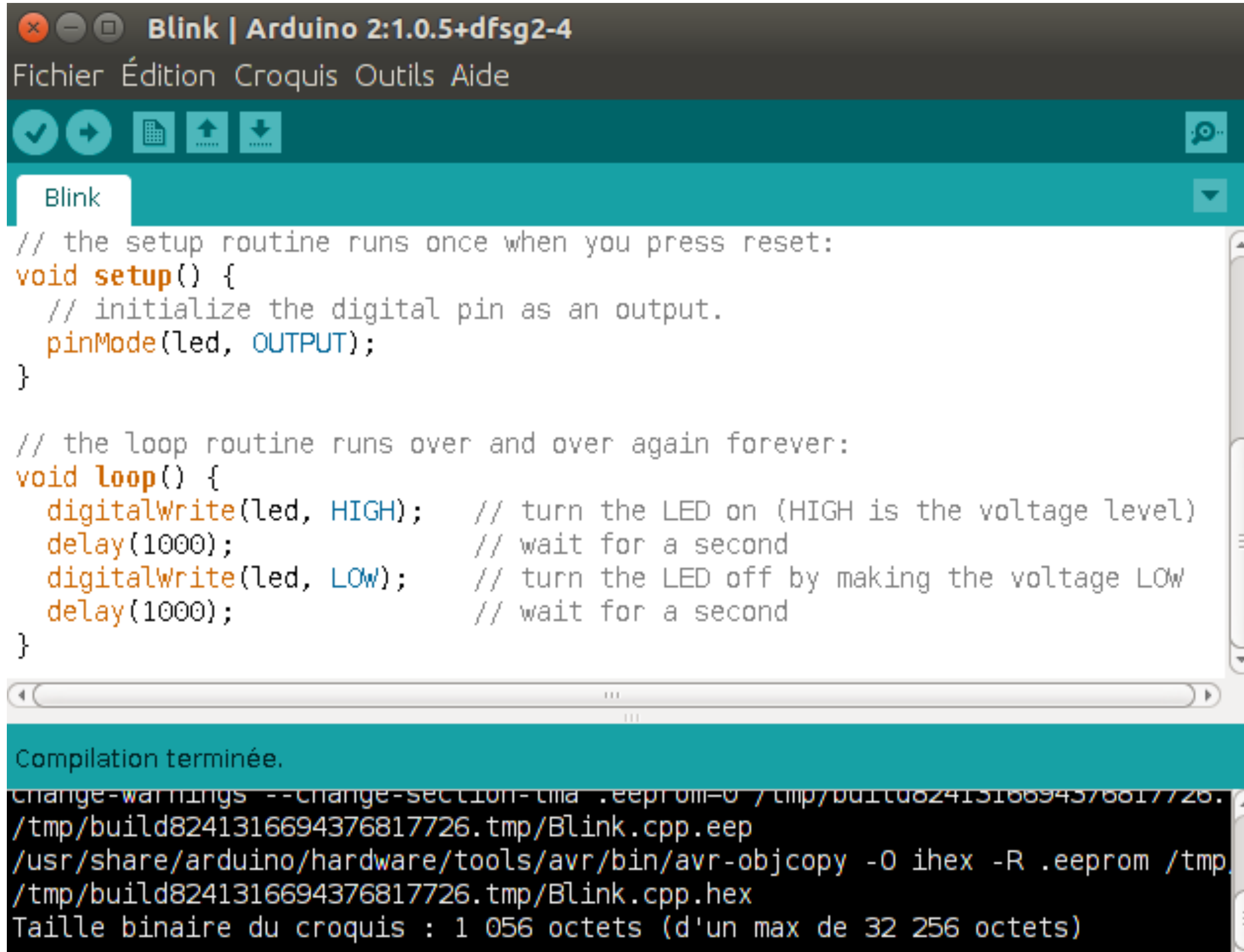
- AVR (ATmega328)
- Connecteur normalisé
- USB (liaison série)
- Matériel libre

Les cartes Arduino



- AVR (ATmega328)
- Connecteur normalisé
- USB (liaison série)
- Matériel libre
- Nombreux *shields*

Le logiciel Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for opening, saving, and running. The "Blink" tab is active, displaying the following C++ code:

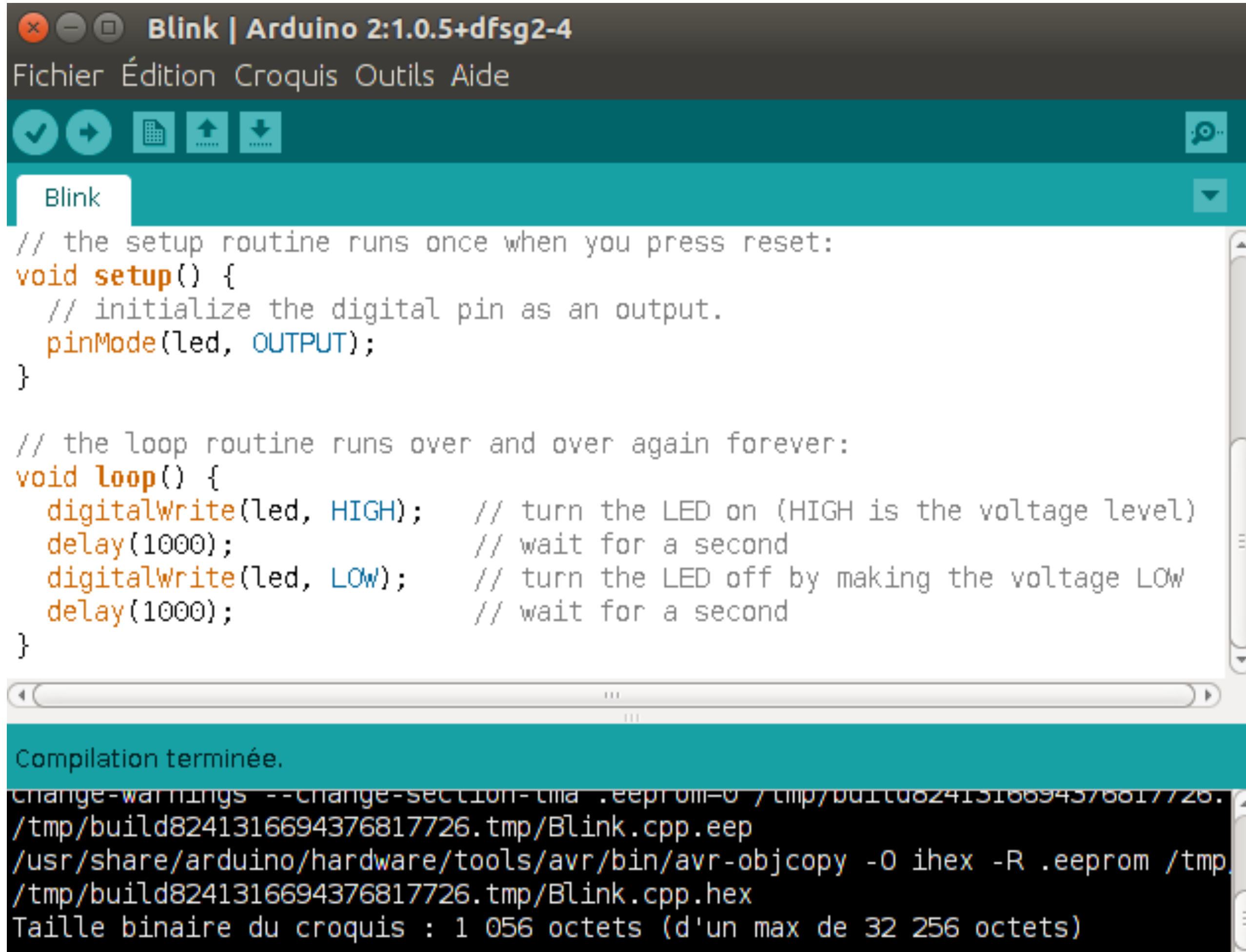
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// 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
}
```

Below the code editor, a status bar indicates "Compilation terminée." (Compilation finished.). The output window shows the compilation process:

```
change-warnings --change-section-lma .eeprom=0 /tmp/build8241316694376817726.
/tmp/build8241316694376817726.tmp/Blink.cpp.eep
/usr/share/arduino/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp.
/tmp/build8241316694376817726.tmp/Blink.cpp.hex
Taille binaire du croquis : 1 056 octets (d'un max de 32 256 octets)
```


Le logiciel Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for opening, saving, and running. The main text area shows the following C++ code for the Blink sketch:

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

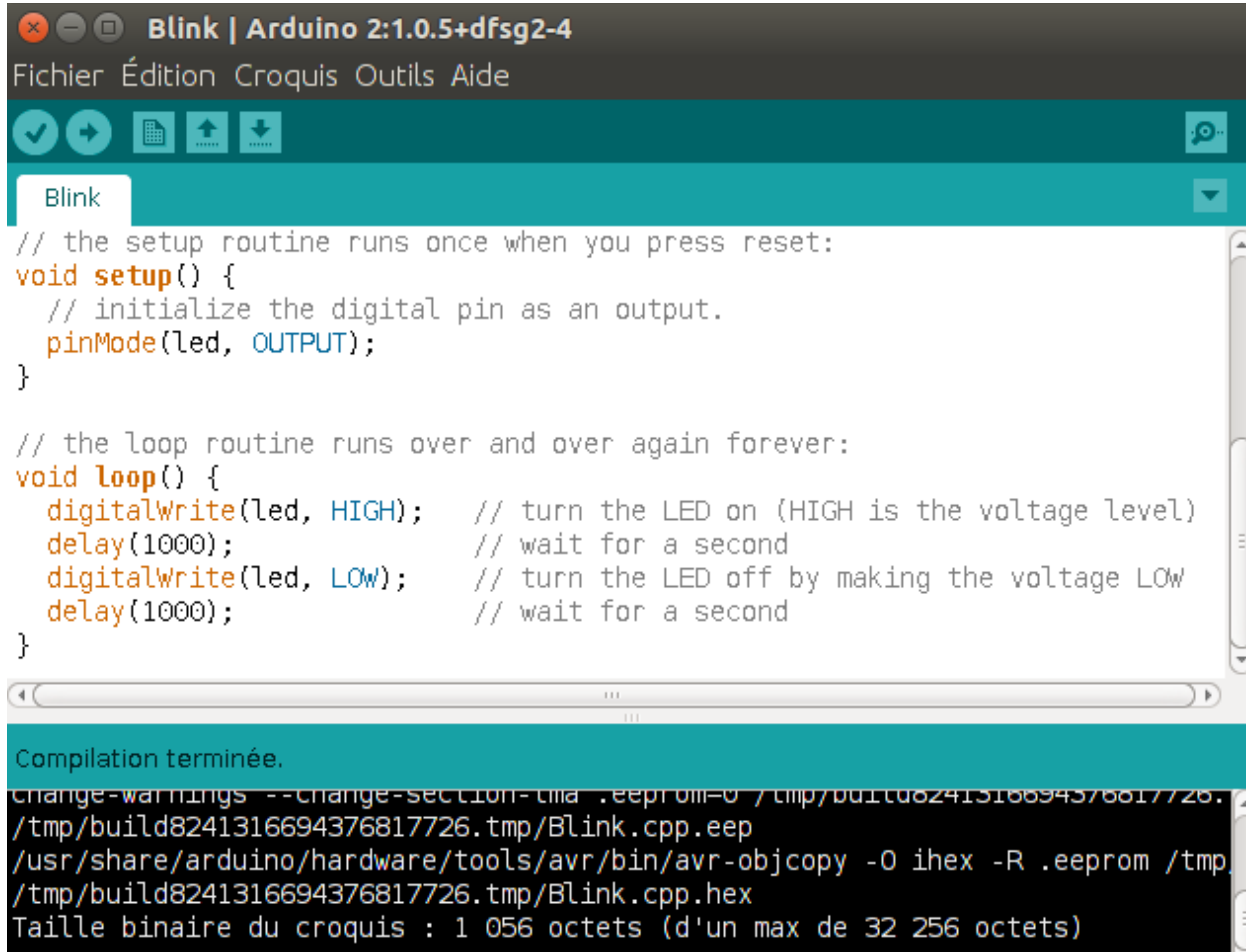
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              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

Below the code editor, a status bar indicates "Compilation terminée." (Compilation finished.). The bottom panel shows the compilation output:

```
change-warnings --change-section-cma .eeprom=0 /tmp/build8241316694376817726.
/tmp/build8241316694376817726.tmp/Blink.cpp.eep
/usr/share/arduino/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp.
/tmp/build8241316694376817726.tmp/Blink.cpp.hex
Taille binaire du croquis : 1 056 octets (d'un max de 32 256 octets)
```

- Écrit en Java
- Utilise le compilateur GCC

Le logiciel Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for opening, saving, and uploading files. The main editor window displays the "Blink" sketch with the following code:

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

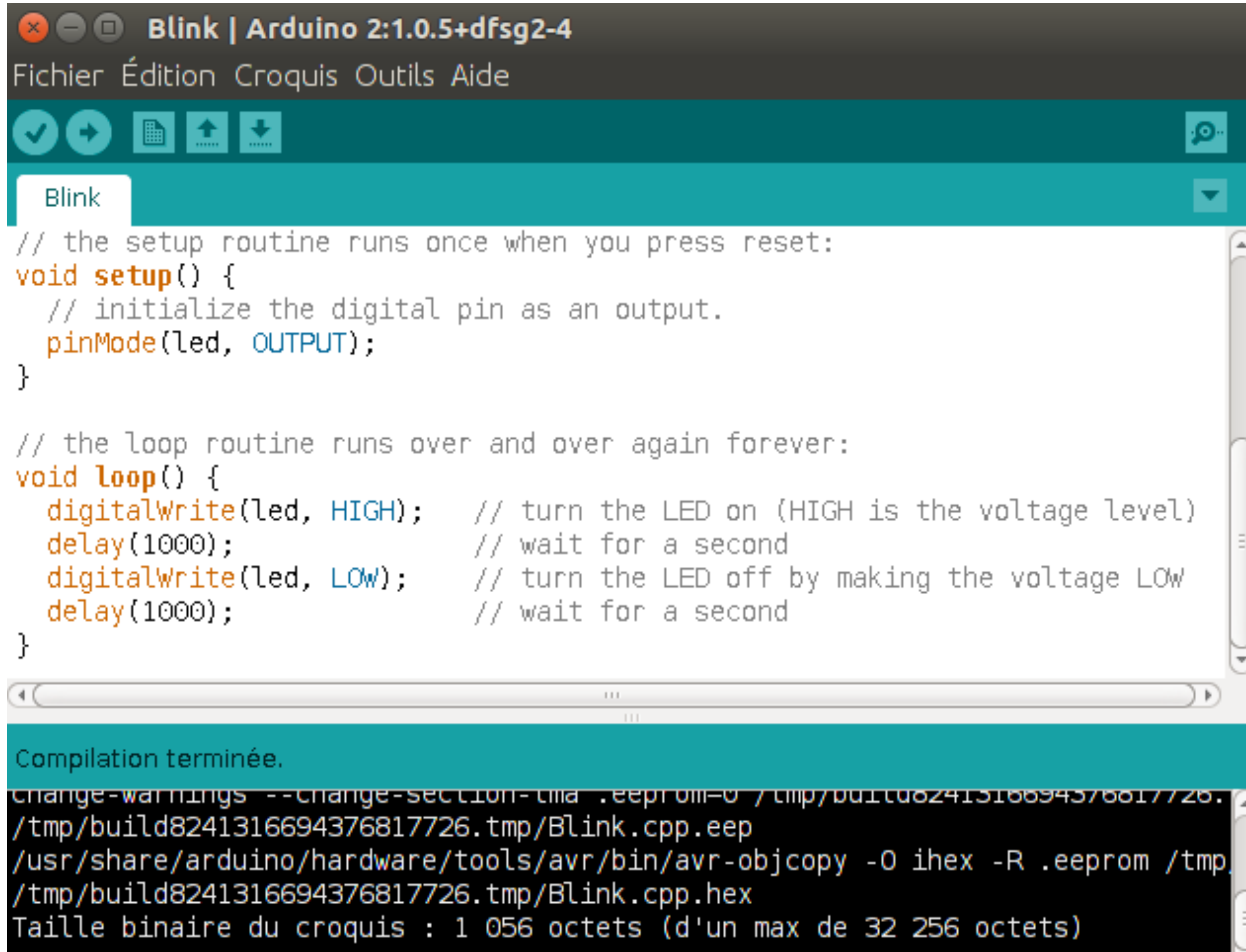
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               //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

Below the code editor, a status bar indicates "Compilation terminée." (Compilation finished.). The output window shows the compilation process:

```
change-warnings --change-section-cma .eeprom=0 /tmp/build8241316694376817726.
/tmp/build8241316694376817726.tmp/Blink.cpp.eep
/usr/share/arduino/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp.
/tmp/build8241316694376817726.tmp/Blink.cpp.hex
Taille binaire du croquis : 1 056 octets (d'un max de 32 256 octets)
```

- Écrit en Java
- Utilise le compilateur GCC
- Windows
- Linux
- MacOS

Le logiciel Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for opening, saving, and running. The main editor shows the "Blink" sketch with the following code:

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

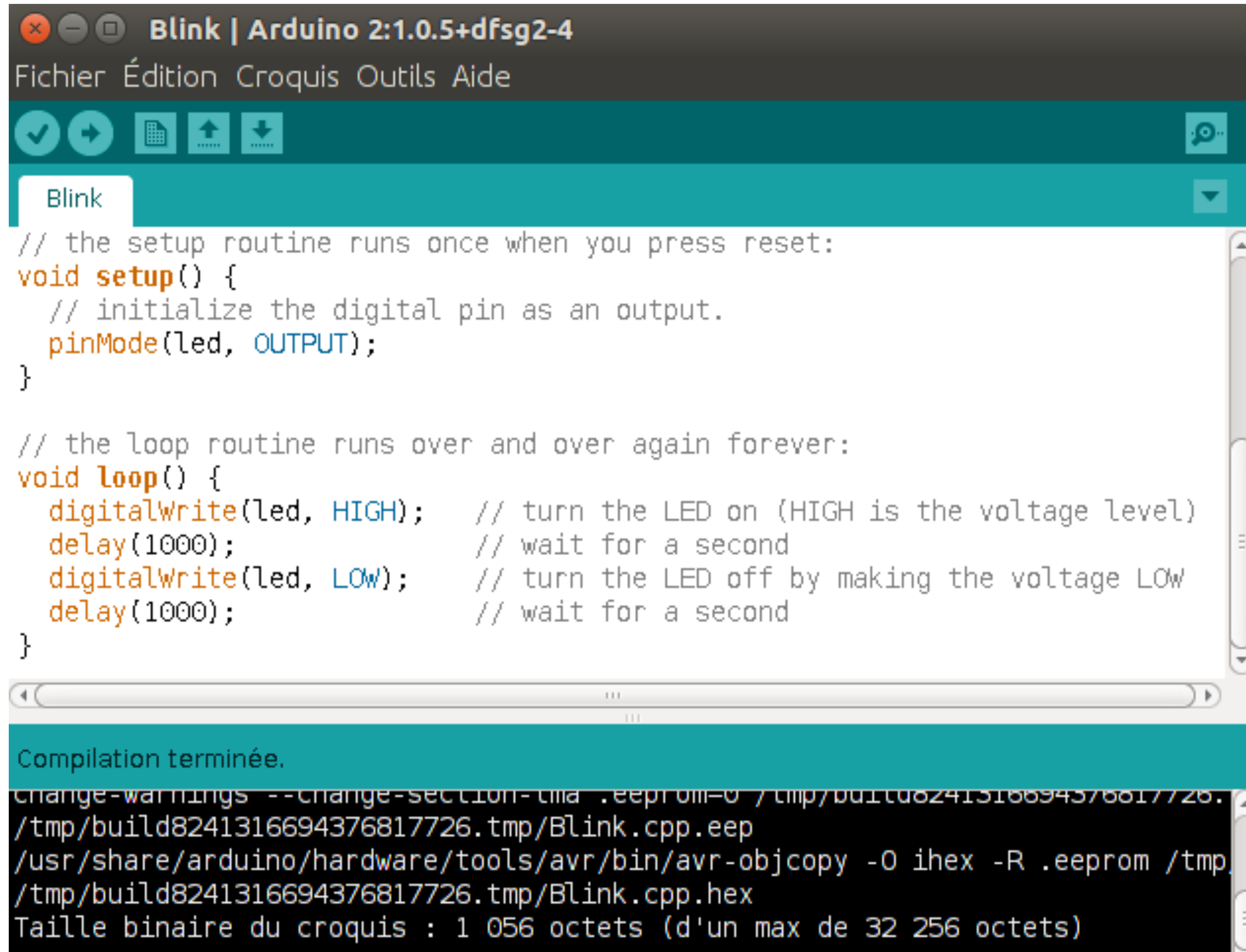
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);                // wait for a second
  digitalWrite(led, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);              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

Below the code editor, a status bar indicates "Compilation terminée." (Compilation finished.). The bottom panel shows the compilation output:

```
change-warnings --change-section-cma .eeprom=0 /tmp/build8241316694376817726.
/tmp/build8241316694376817726.tmp/Blink.cpp.eep
/usr/share/arduino/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp.
/tmp/build8241316694376817726.tmp/Blink.cpp.hex
Taille binaire du croquis : 1 056 octets (d'un max de 32 256 octets)
```

- Écrit en Java
- Utilise le compilateur GCC
- Windows
- Linux
- MacOS
- Logiciel libre (Open source)

Le logiciel Arduino



The screenshot shows the Arduino IDE interface. The title bar reads "Blink | Arduino 2:1.0.5+dfsg2-4". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for opening, saving, and running. The main editor displays the "Blink" sketch with the following code:

```
// 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
}
```

Below the code editor, a status bar indicates "Compilation terminée." (Compilation finished.). The bottom panel shows the compilation output:

```
change-warnings --change-section-tma .eeprom=0 /tmp/build8241316694376817726.
/tmp/build8241316694376817726.tmp/Blink.cpp.eep
/usr/share/arduino/hardware/tools/avr/bin/avr-objcopy -O ihex -R .eeprom /tmp.
/tmp/build8241316694376817726.tmp/Blink.cpp.hex
Taille binaire du croquis : 1 056 octets (d'un max de 32 256 octets)
```

- Écrit en Java
- Utilise le compilateur GCC
- Windows
- Linux
- MacOS
- Logiciel libre (Open source)
- Version pour d'autres microcontrôleurs
- Energia pour les MSP430

La librairie Arduino

La librairie Arduino

- Un ensemble de procédures (librairie) écrites en C (ou C++)
- *Wiring*

La librairie Arduino

- Un ensemble de procédures (librairie) écrites en C (ou C++)
- *Wiring*
- Cacher la complexité du microcontrôleur

La librairie Arduino

- Un ensemble de procédures (librairie) écrites en C (ou C++)
- *Wiring*
- Cacher la complexité du microcontrôleur
- Apporter une certaine compatibilité entre différents microcontrôleurs

La librairie Arduino

- Un ensemble de procédures (librairie) écrites en C (ou C++)
 - *Wiring*
 - Cacher la complexité du microcontrôleur
 - Apporter une certaine compatibilité entre différents microcontrôleurs
-
- La structure générale d'un programme
 - Les entrées-sorties
 - La gestion du temps

Structure générale d'un programme

- *pas de* procédure `main()`

Structure générale d'un programme

- *pas de* procédure `main()`
- `void setup()`

Structure générale d'un programme

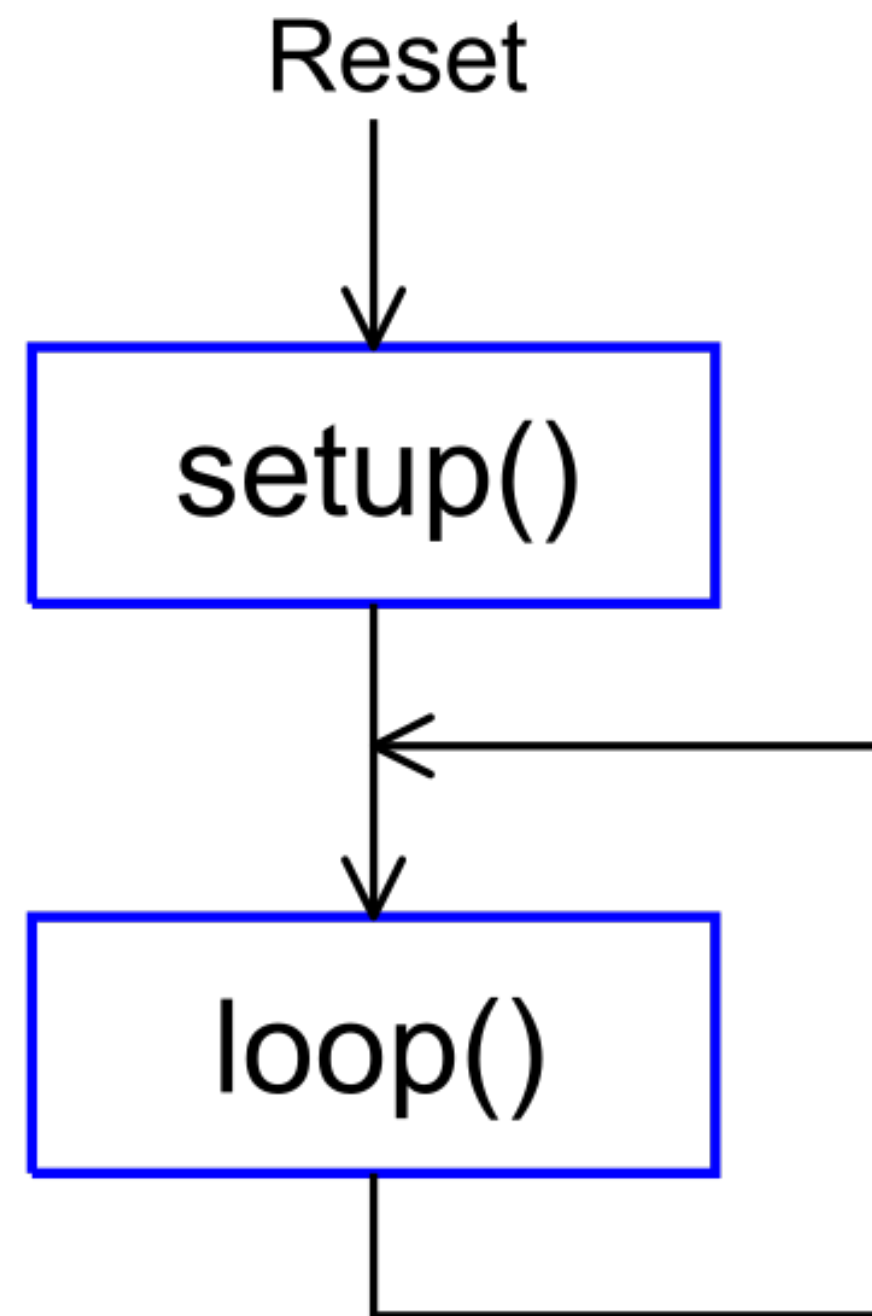
- *pas de* procédure `main()`
- `void setup()`
- `void loop()`

Structure générale d'un programme

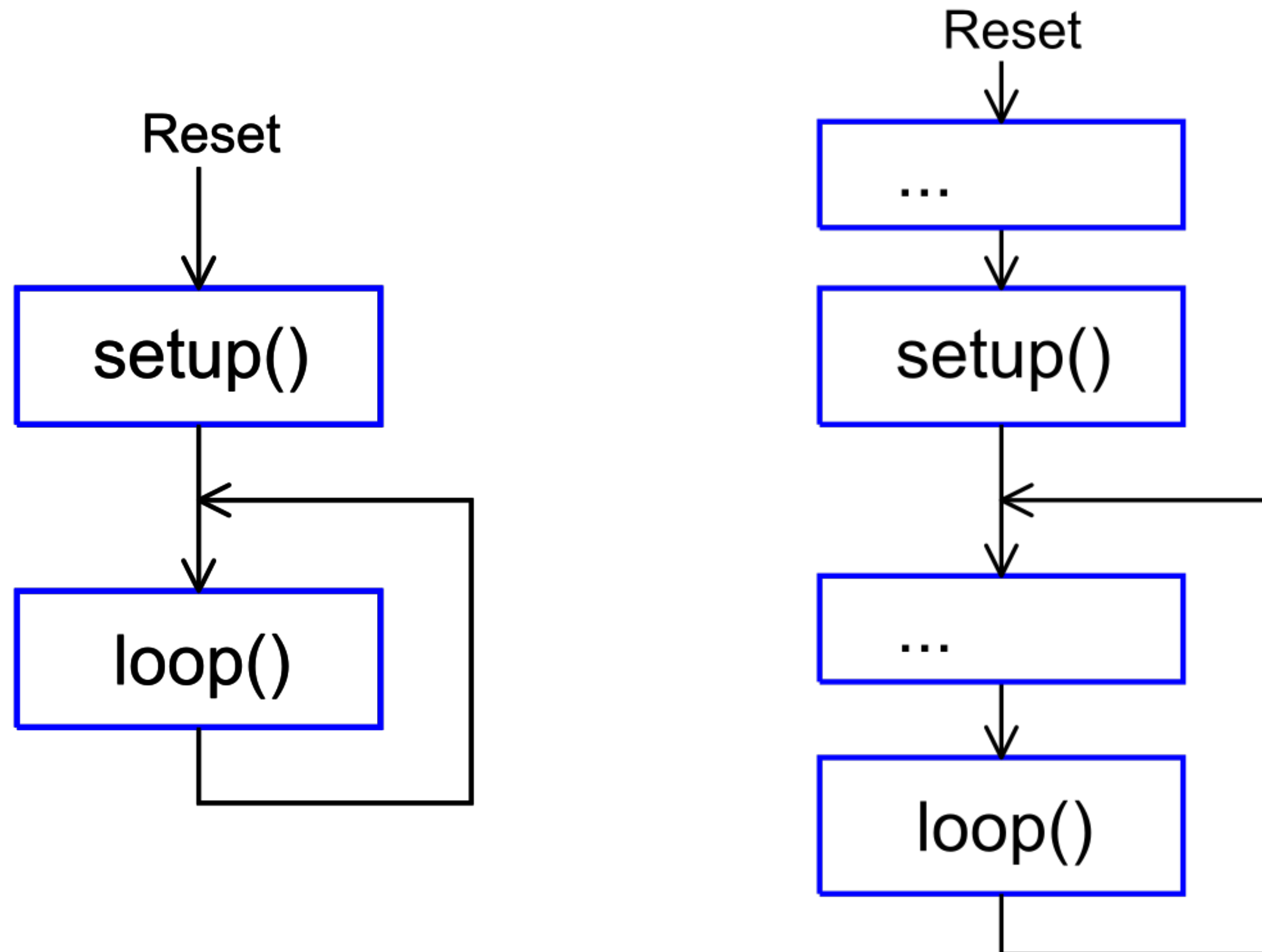
- *pas de* procédure `main()`
- `void setup()`
- `void loop()`

```
1 void setup() {  
2 }  
3  
4 void loop() {  
5 }
```

Organigramme d'un programme Arduino



Organigramme d'un programme Arduino



- `void pinMode(pin, mode)`

Entrées-sorties

- `void pinMode(pin, mode)`
- `void digitalWrite(pin, value)`

Entrées-sorties

- `void pinMode(pin, mode)`
- `void digitalWrite(pin, value)`
- `boolean digitalRead(pin)`

Exemple de programme

```
1 void setup() {  
2     pinMode(P1_0, OUTPUT);  
3     pinMode(P1_3, INPUT);  
4 }  
5  
6 void loop() {  
7     digitalWrite(P1_0, (digitalRead(P1_3)));  
8 }
```


- `int delay(ms)`

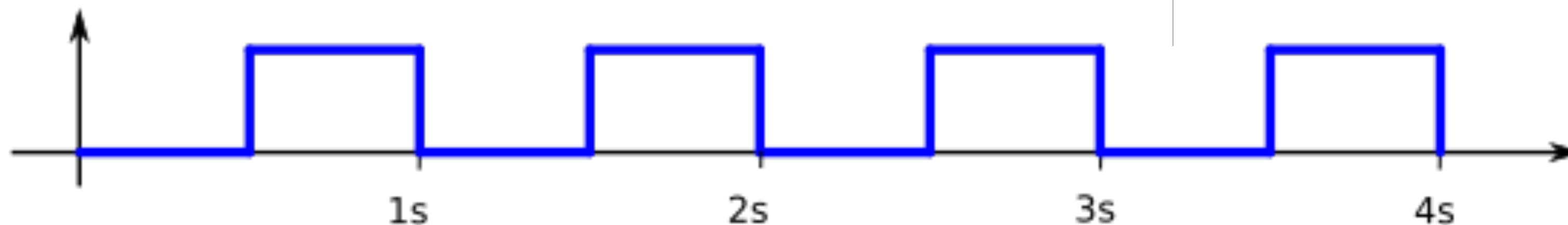
- `int delay(ms)`

```
1 void setup() {  
2     pinMode(P1_0, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(P1_0, 1);  
7     delay(500);  
8     digitalWrite(P1_0, 0);  
9     delay(500);  
10 }
```

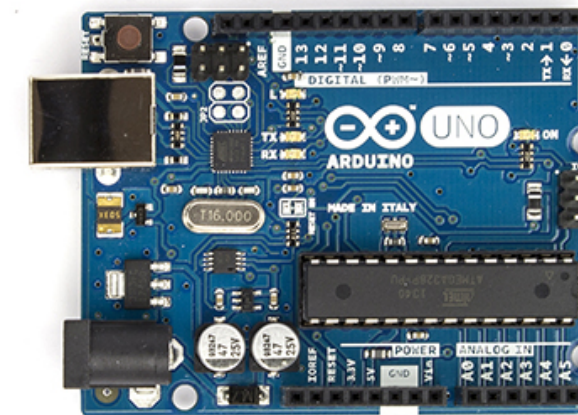
Gestion du temps

- `int delay(ms)`

```
1 void setup() {  
2     pinMode(P1_0, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(P1_0, 1);  
7     delay(500);  
8     digitalWrite(P1_0, 0);  
9     delay(500);  
10 }
```



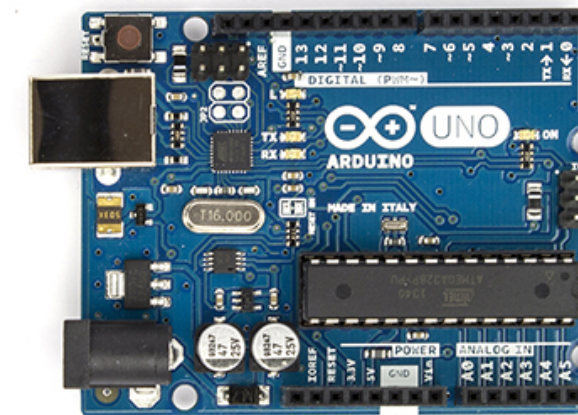
Programmation en C-Arduino

A screenshot of the Arduino IDE interface. At the top, there are icons for checking, running, saving, and uploading. Below these, the file name 'Blink' is shown. The main area displays the C++ code for the Blink example, which includes comments and function definitions for the setup and loop routines.

```
// the setup routine
void setup() {
  // initialize the (
  pinMode(led, OUTPUT
}
```

- Les 3 significations du mot Arduino
- Structure d'un programme
- Entrées-sorties
- Gestion du temps

Programmation en C-Arduino



```
✓ → 📄 ⬆ ⬇  
Blink  
// the setup routine  
void setup() {  
  // initialize the  
  pinMode(led, OUTPUT  
}
```

- Les 3 significations du mot Arduino
- Structure d'un programme
- Entrées-sorties
- Gestion du temps

et de nombreuses librairies !