

Club ELEC

Projet Robot

HO2 - Télécommande infrarouges et programmation

11 mars 2019

Introduction

Pendant ce quadrimestre, le Club ELEC vous propose de développer un petit robot contrôlé par une télécommande à infrarouges. Pour ce faire, le développement du circuit se déroulera en 3 phases, chacune correspondant à une séance de hands-on proposée par le club.

- HO1 : Contrôle des moteurs.
- HO2 : Télécommande infrarouges et programmation.
- HO3 : Assemblage du robot.

Objectifs

Les objectifs de ce second hands-on sont :

- D'installer correctement le programmeur d'Arduino ainsi que les librairies nécessaires et de se familiariser avec l'interface.
- De se familiariser avec le module Arduino Nano et apprendre à le programmer.
- D'interfacer le module Arduino Nano avec un récepteur à infrarouges pour pouvoir le contrôler à distance avec une télécommande à infrarouges.

À la fin de cette séance, vous devriez donc être capable de contrôler votre module Arduino Nano à distance avec votre télécommande, pour par exemple lui faire allumer une LED.

Installations nécessaires

Afin de pouvoir programmer votre module Arduino Nano, plusieurs étapes sont nécessaires. Il vous faut d'abord **télécharger et installer le programmeur Arduino**. Celui-ci est aussi appelé Arduino IDE (Arduino Integrated Development Environment). Voici les instructions à suivre :

1. se rendre sur la page principale du site Arduino (www.arduino.cc/);
2. cliquer sur l'onglet "Software > Downloads";
3. choisir votre version d'Arduino IDE (voir Fig. 1) et la télécharger;
4. une fois le téléchargement effectué, il ne vous reste plus qu'à installer le programmeur.

Download the Arduino IDE

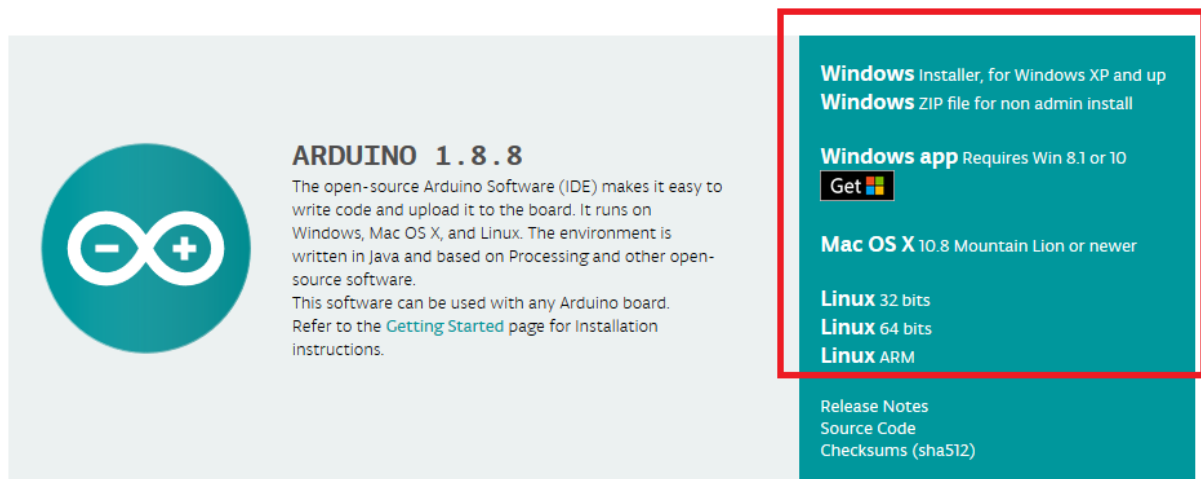


Fig. 1 – Téléchargement du programmeur Arduino.

Pour la suite de la séance, il vous faudra également **installer la librairie IRremote**. Cette librairie gère l'interfaçage avec le récepteur infrarouges et permettra à votre Arduino Nano de correctement recevoir les données envoyées par la télécommande infrarouges. Voici les instructions à suivre :

1. ouvrir l'Arduino IDE fraîchement installé;
2. aller dans *Sketch* → *Include Library* → *Manage Libraries...*;
3. taper "IRremote" dans la barre de recherche;
4. cliquer sur la librairie "IRremote" (voir Fig. 2) et l'installer;
5. voilà !

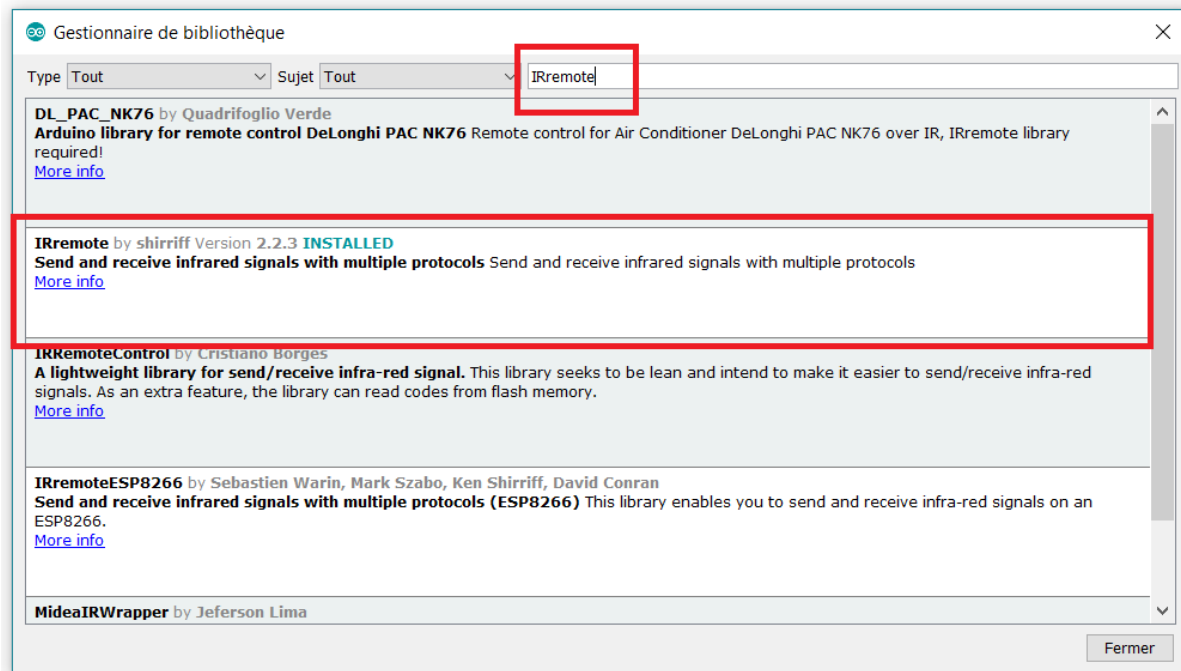


Fig. 2 – Installation de la librairie IRremote.

Enfin, avant de programmer votre module Arduino, il faut s'assurer que certaines configurations soient bien faites dans l'IDE Arduino : d'abord le type de module Arduino que nous allons programmer (Nano dans notre cas), ensuite le type de processeur existant sur ce module (ATmega328P dans notre cas). Voici comment faire pour choisir ces paramètres :

1. aller dans *Tools* → *Board* et choisir *Arduino Nano* comme type de module Arduino à utiliser (voir Figure 3).
2. aller dans *Tools* → *Processor* et choisir *ATmega328P* comme processeur (voir Figure 4).

Vous êtes enfin prêts à programmer votre module Arduino. Vous l'aurez remarqué, Arduino est open-source et gratuit. Sachez aussi qu'il existe une flopée de librairies existantes pour faire à peu près n'importe quoi !

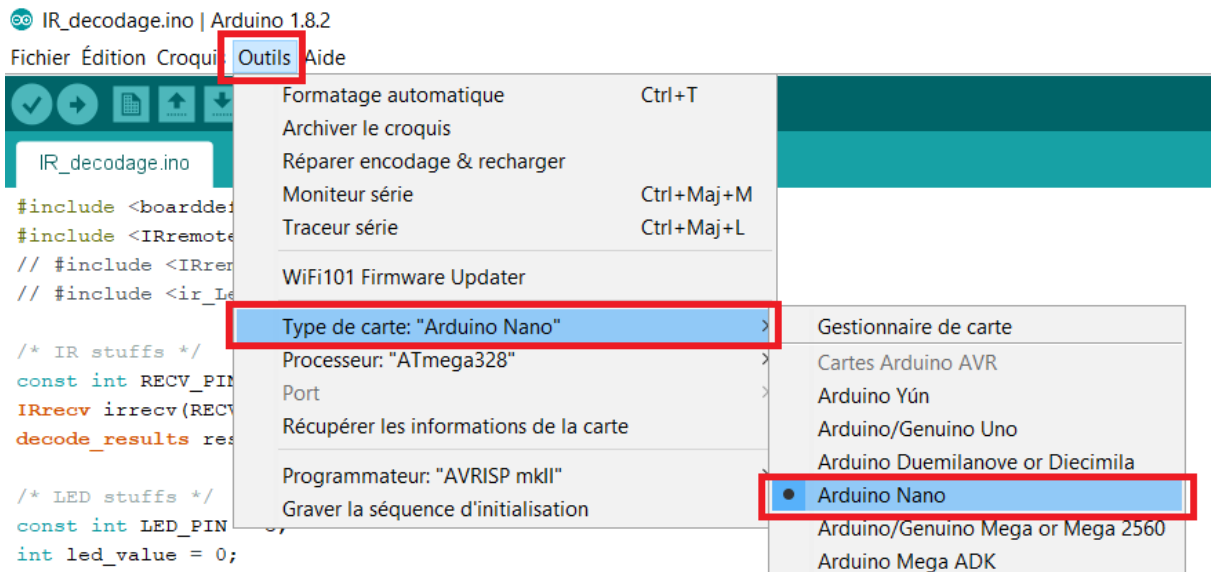


Fig. 3 – Choix du type de module Arduino.

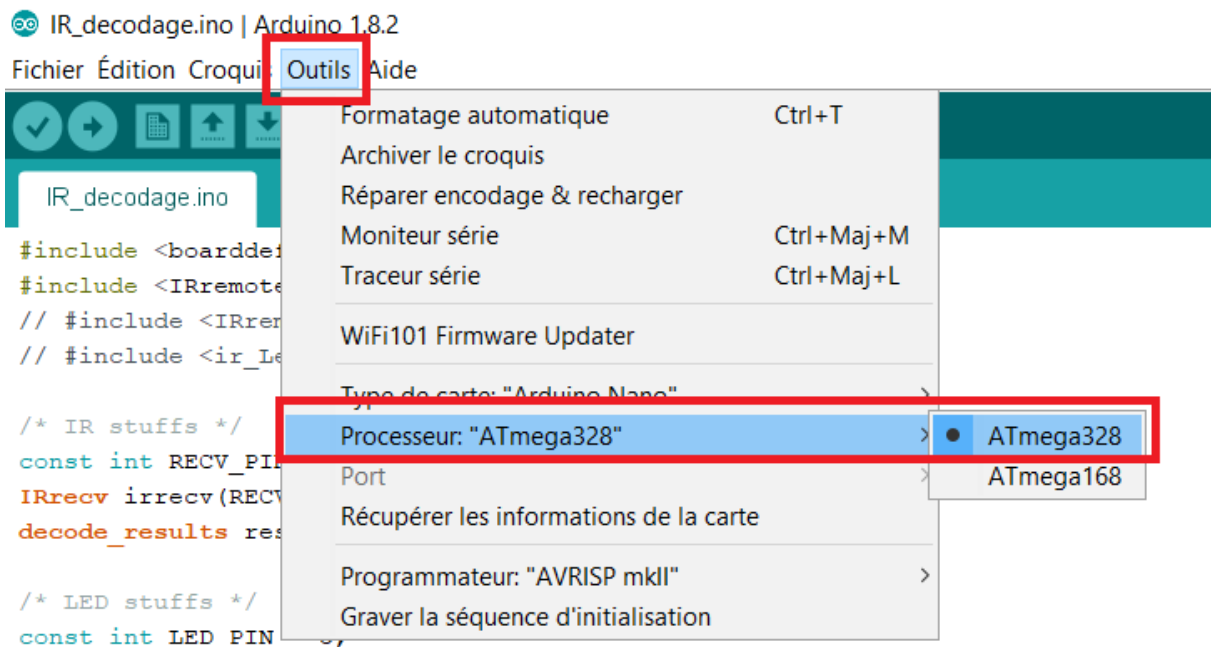


Fig. 4 – Choix du type de processeur sur le module Arduino.

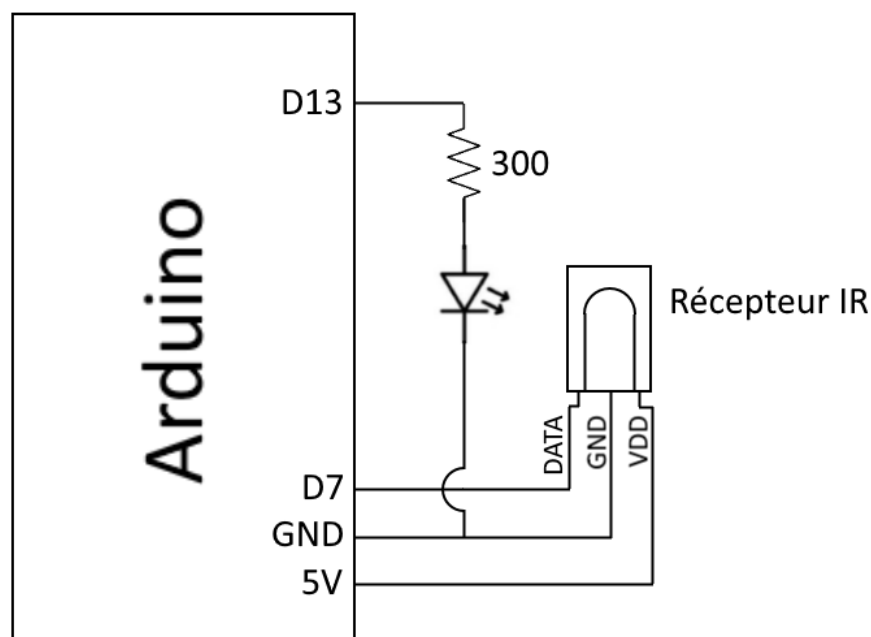


Fig. 5 – Circuit à réaliser pour contrôler l'Arduino grâce à une télécommande infrarouge.

Interfaçage de composants autour de l'Arduino Nano

Le circuit que nous allons utiliser aujourd'hui se compose de quatre éléments :

1. D'une télécommande infrarouges
2. Récepteur infrarouges
3. Une led
4. L'Arduino qui est le "cerveau" du système. Il allume la Led quand la télécommande lui demande.

Pour que tout ceci fonctionne, la première étape est d'assembler tout ces composants sur une bread-board. Le schéma du circuit est disponible à la 5.

Premier programme : récupération des codes envoyés par la télécommande

Le but de cet exercice est d'établir une communication entre votre télécommande et l'Arduino. Typiquement, quand votre télécommande essaye de communiquer avec votre télévision, cela se passe 3 étapes :

1. Quand vous appuyez sur un bouton, la télécommande envoie une valeur X via infrarouges.
2. La télévision reçoit cette valeur X via le récepteur infrarouges.
3. En fonction de la valeur de X, le microcontrôleur de la télévision fait une action. Il peut soit augmenter le volume, changer de chaîne ou encore s'éteindre.

La toute première étape est donc de savoir quelle valeur est envoyée par votre télécommande quand vous appuyez sur un certain bouton. Le programme Listing 1 permet de faire cela. En quelques mots, à chaque fois que l'Arduino reçoit un message infrarouges, il l'envoie directement à l'ordinateur. Cela vous permet donc de savoir le code correspondant à chacun des boutons de votre télécommande (notez ça sur un feuille;)).

Pour voir les messages en console, vous devez aller dans *Tools* → *Serial Monitor*.

Listing 1 – Lecture de la télécommande

```

1 #include <IRremote.h>
2
3 const int RECV_PIN = 7;
4
5 IRrecv irrecv(RECV_PIN);
6 decode_results results;
7
8 void setup(){
9     Serial.begin(9600);
10    irrecv.enableIRIn();
11 }
12
13 void loop(){
14     if (irrecv.decode(&results)){
15         Serial.println(results.value, HEX);
16         irrecv.resume();
17     }
18 }

```

Second programme : contrôle d'une LED avec la télécommande

Une fois que vous avez identifié le code que votre télécommande envoie pour un certain bouton, vous pouvez adapter l'exemple Listing 2. Ici, à chaque itération de la fonction *loop()*, l'Arduino regarde si le récepteur infrarouges a reçu un message. Pour chaque message reçu, il regarde à quel bouton ce code correspond (ici *AB456CD* pour le bouton 1 et *05FBAC33* pour le bouton 2). Ensuite, il exécute une fonction correspondant au bouton utilisé (*IR_button1()*). Dans le cas du bouton 1, la Led s'allume ou s'éteint à chaque fois que le bouton est appuyé.

Listing 2 – Code 2

```

1 #include <IRremote.h>
2
3 #define REMOTE_BUTTON1    0xAB456CD    // Ligne a modifier
4 #define REMOTE_BUTTON2    0x5FBAC33    // Ligne a modifier
5
6 const int RECV_PIN = 7;
7 const int LED_PIN = 13;
8
9 IRrecv irrecv(RECV_PIN);
10 decode_results results;
11
12 int LED_val = 0;
13
14 void setup() {
15     Serial.begin(9600);
16     pinMode(LED_PIN,OUTPUT);
17     digitalWrite(LED_PIN,LED_val);
18     irrecv.enableIRIn();
19     delay(500);
20 }
21
22 void loop() {
23     if (irrecv.decode(&results)){
24         //Serial.println(results.value, HEX);
25         irrecv.resume();
26         switch (results.value) {

```

```
27     case REMOTE_BUTTON1:
28         IR_button1 ();
29         break;
30     case REMOTE_BUTTON2:
31         IR_button2 ();
32         break;
33     }
34     irrecv.resume();
35 }
36
37
38 void IR_button1 () {
39     LED_val = 1-LED_val;
40     digitalWrite(LED_PIN, LED_val);
41 }
42
43 void IR_button2 () {
44     // faire qqch
45 }
```

Contrôle du mouvement d'un robot

Dans le hands-on 1, vous avez appris dans ce hands-on à contrôler des moteurs DC à partir de signaux de contrôle digitaux. Durant cette séance-ci, vous avez appris à envoyer des commandes à distances à un module Arduino équipé d'un récepteur infrarouges, grâce à une télécommande. Vous êtes maintenant capables de combiner les deux hands-on et de créer un petit robot à téléguider. Nous vous suggérons d'utiliser la fonction `analogWrite` qui est très utile car elle fait de la modulation à largeur d'impulsions (MLI) pour vous! Pensez-y pour régler la vitesse de votre robot. Pour plus d'informations sur la fonction `analogWrite`, vous pouvez consulter le lien suivant : www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/.