**Objectif** : ***Décoder une trame IR pour piloter un processus*.**

**A partir des informations fourniesci-dessous, réaliser le TP en suivant les consignes données.**

**Matériel nécessaire**

|  |  |
| --- | --- |
| ● 1 **carte Arduino Uno**,  ● 1 **capteur IR TSOP 1733 (ou équivalent)**  ● 1 **diode LED avec sa résistance de polarisation**  ● 1 **breadboard**  ● 5 **connecteurs mâle-mâle**  ● La librairie **IRremote** doit être installée  ● 1 **télécommande IR** pour faire les tests |  |

:

**Présentation du capteur IR**

|  |  |
| --- | --- |
| Les capteurs de la série **TSOP17..** sont des récepteurs miniaturisés pour systèmes de télécommande infrarouge.  Le signal de sortie démodulé peut être directement décodé par un microprocesseur. Cette série permet la réception de signaux provenant de télécommandes **IR** standard, prenant en charge tous les principaux codes de transmission. |  |

**Sketch Arduino :**

Voici un exemple de code Arduino qui analyse les trames **IR** reçues par un capteur **TSOP1733**

Ce code utilise la bibliothèque **IRremote** pour recevoir les données IR à partir du capteur branché sur le **pin 11**.

Les données décodées sont ensuite imprimées sur la liaison série en format **HEX** pour une analyse ultérieure.

Le capteur **TSOP1733** est un capteur **infrarouge** (**IR**) à distance, qui peut être branché de différentes manières en fonction de l'application et du circuit utilisé.

Raccordement du capteur TSOP1733 à une carte Arduino :

* Le **pin 1** (le plus à gauche côté bosselage) du capteur (généralement marqué comme **GND**) est connecté à la masse (**GND**) de l'Arduino.
* Le **pin 2 (Vcc)** (au milieu) est connecté à une source d'alimentation de **3,3V** ou **5V** selon la spécification du capteur.
* Le **pin 3 (DATA)** (le plus à droite côté bosselage) est connecté au pin d'entrée de l'Arduino sur lequel vous souhaitez recevoir les données **IR**. (ici **11**)

Il est important de noter que le capteur **TSOP1733** est un **récepteur IR passif**, il ne nécessite pas d'alimentation pour fonctionner. Il faut un émetteur **IR** pour l'alimenter.

Il est également important de respecter les polarités du capteur et de s'assurer que la tension d'alimentation est correctement appliquée pour éviter d'endommager le capteur.

Il est également préférable de protéger le capteur contre les surtensions et les courts-circuits.

**Travail à effectuer :**

1 – Effectuer le cablage du montage défini ci-dessus.

2 – Saisir le code du sketch **IR\_HEX\_SERIAL**.

3 – Téléverser le code sur la **carte ARDUINO** et lancer moniteur série.

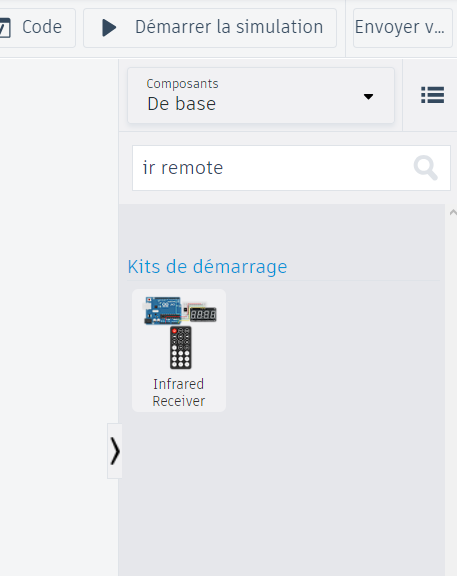
4 – Choisir **deux touches** de la télécommande fournie et décoder les **valeurs héxadécimales** transmises.

5 – Effectuer une modification du sketch afin qu’il réalise la fonction suivante :

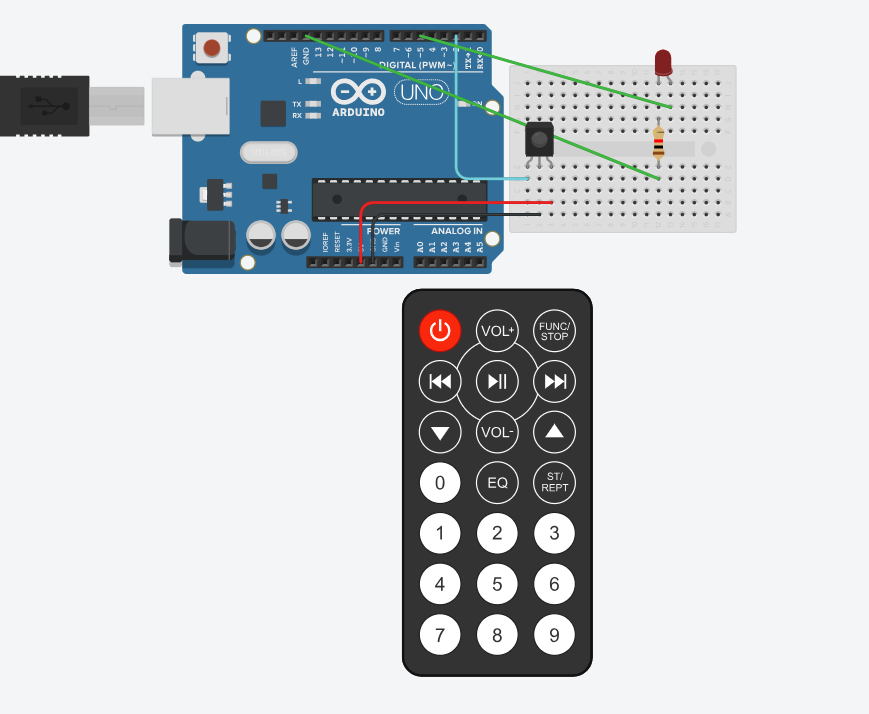
* Un appui sur la **première touche** allume la **LED** pendant **5 secondes** ;
* Un appui sur la **seconde touche** allume la **LED** pendant **10 secondes**.

Nommer ce nouveau sketch : **IR\_2TOUCHES\_1LED**

**Composant du montage :**



**Montage :**



**Code :**

#include "Adafruit\_LEDBackpack.h"

#include <IRremote.h>

int button = 0;

int ledPin = 5;

int touche1 = 0; //touche pour allumer led (peut être changé)

int touche2 = 1;

Adafruit\_7segment led\_display1 = Adafruit\_7segment();

int mapCodeToButton(unsigned long code) {

if ((code & 0x0000FFFF) == 0x0000BF00) {

code >>= 16;

if (((code >> 8) ^ (code & 0x00FF)) == 0x00FF) {

return code & 0xFF;

}

}

return -1;

}

int readInfrared() {

int result = -1;

if (IrReceiver.decode()) {

unsigned long code = IrReceiver.decodedIRData.decodedRawData;

result = mapCodeToButton(code);

IrReceiver.resume();

}

return result;

}

void setup()

{

IrReceiver.begin(2);

Serial.begin(9600);

pinMode(ledPin, OUTPUT);

}

void loop()

{

button = readInfrared();

if (button >= 0) {

Serial.println(button);

if (button == touche1) {

digitalWrite(ledPin, HIGH);

delay(5000);

digitalWrite(ledPin, LOW);

}

if (button == touche2) {

digitalWrite(ledPin, HIGH);

delay(10000);

digitalWrite(ledPin, LOW);

}

}

delay(10);

}