

## RAPPORT DE SEANCE DU PROJET ARDUINO

03/01/2022

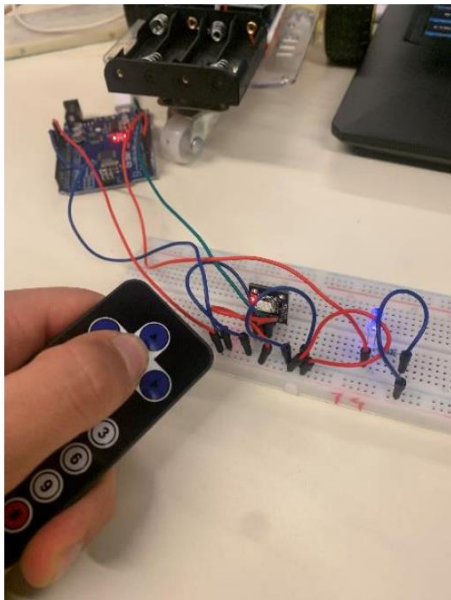
### A – Création d'un capteur d'information via une télécommande

Suite à tous mes problèmes et toutes mes questions, Monsieur Masson m'a révélé que le KIT infrarouge GT017 servait bien uniquement de communiquer des informations entre le capteur et la télécommande. Cela ne permet donc pas de calculer la vitesse mais cependant de faire des actions via la commande d'une télécommande. Cela a donc résolu tous mes problèmes de la séance dernière.

J'ai donc tout rebranché avec mon code initial et tout fonctionnait.

Je devais améliorer mon code car la led ne s'allumait pas tout le temps selon la touche que j'avais assigné.

On rappelle le branchement et le code initial afin d'obtenir un code décimal pour chaque touche de ma télécommande :



```
HEX hex =0;
int led = 2;
int RECV_PIN = 11; // Définition de la broche Signal du récepteur IR raccordé sur la carte
IRrecv irrecv (RECV_PIN); // Déclaration du module IR
decode_results results;
void setup ()
{
  Serial.begin (9600); // Démarre la communication série
  irrecv.enableIRin (); // Démarrage du récepteur IR
  pinMode(led,OUTPUT);
}
void loop () {

  Serial.println(results.value);
  //Serial.print(digitalRead(RECV_PIN));
  //delay(1000);
  //delay(1000);
  if (irrecv.decode (& results)) {
    Serial.println (results.value, HEX); // Affichage de la valeur reçue en hexadécimal
    irrecv.resume (); // Réception de la valeur suivante
    digitalWrite(led,LOW);
    delay(1000);
  }

  if (hex="FFFFFF") {
    digitalWrite(led,LOW);
  }
}
```

Cependant on remarque tout de suite qu'il y'a un gros problème. Sur la dernière ligne de code « if (hex = « FFFFF »). Hex a été supposé initialisé comme un hexadécimal égal à 0. Cependant cela ne fonctionne pas car la variable HEX ne peut pas être utilisé comme un int ou un bool ect. Nous

pouvons faire la conversion d'un entier en hexadécimal directement dans le `println` mais pas dans une variable.

J'ai donc commencé à relever tous les codes hexadécimal assignés aux touches (affiché dans le moniteur série) et les ait mis dans un tableau

### Codes télécommande GT017

<b>Haut</b>	FF629D
<b>Bas</b>	FFA857
<b>Gauche</b>	FF22DD
<b>Droite</b>	FFC23D
<b>OK</b>	FF02FD
<b>1</b>	FF6897
<b>2</b>	FF9867
<b>3</b>	FFB04F
<b>4</b>	FF30CF
<b>5</b>	FF18E7
<b>6</b>	FF7A85
<b>7</b>	FF10EF
<b>8</b>	FF38C7
<b>9</b>	FF5AA5
<b>*</b>	FF42BD
<b>0</b>	FF4AB5
<b>#</b>	FF52AD

En remarquant qu'un message « FFFFFF » est affiché à pour chaque qu'une touche appuyée. On peut s'en servir pour faire une action groupée.

Je voulais m'orienter vers un code qui permettrait de convertir les codes obtenus en base décimal, en base hexadécimal. Sauf que cela s'annonçait très compliqué et avec de l'aide, on m'a expliqué qu'il était possible de mettre directement un code en hexadécimal en identifiant sa base avant avec le « 0x ».

Cela a donc tout facilité et j'ai donc tout mis dans une constante `int`. Une constante pour chaque bouton.

```

const int UN_BUTTON = 0xFF6897;
const int DEUX_BUTTON = 0xFF9867;
const int TROIS_BUTTON = 0xFFB04F;
const int QUATRE_BUTTON = 0xFF30CF;
const int CINQ_BUTTON = 0xFF18E7;
const int SIX_BUTTON = 0xFF7A85;
const int SEPT_BUTTON = 0xFF10EF;
const int HUIT_BUTTON = 0xFF38C7;
const int NEUF_BUTTON = 0xFF5AA5;
const int *_BUTTON = 0xFF42BD;
const int ZERO_BUTTON = 0xFF4AB5;
const int HASHTAG_BUTTON = 0xFF52AD;

// ...

void setup ()
{
  Serial.begin (9600); // Démarre la communication série
  irrecv.enableIRIn (); // Démarrage du récepteur IR
  pinMode(led_r, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(led, OUTPUT);
}

void loop () {

  //Serial.print(digitalRead(RECV_PIN));
  //delay(1000);
  //delay(1000);
  //digitalWrite(led_g, HIGH);
  //delay(1000);
  //Serial.println("testHt");

  if (irrecv.decode (& results)) {
    Serial.print ("results.value");Serial.println (results.value, HEX);
    if (results.value == OK_BUTTON)
      Serial.println("testok");

    Serial.println (results.value, HEX); // Affichage de la valeur reçue en hexadécimal
    irrecv.resume (); // Réception de la valeur suivante
    digitalWrite(led, LOW);
    delay(1000);

  }

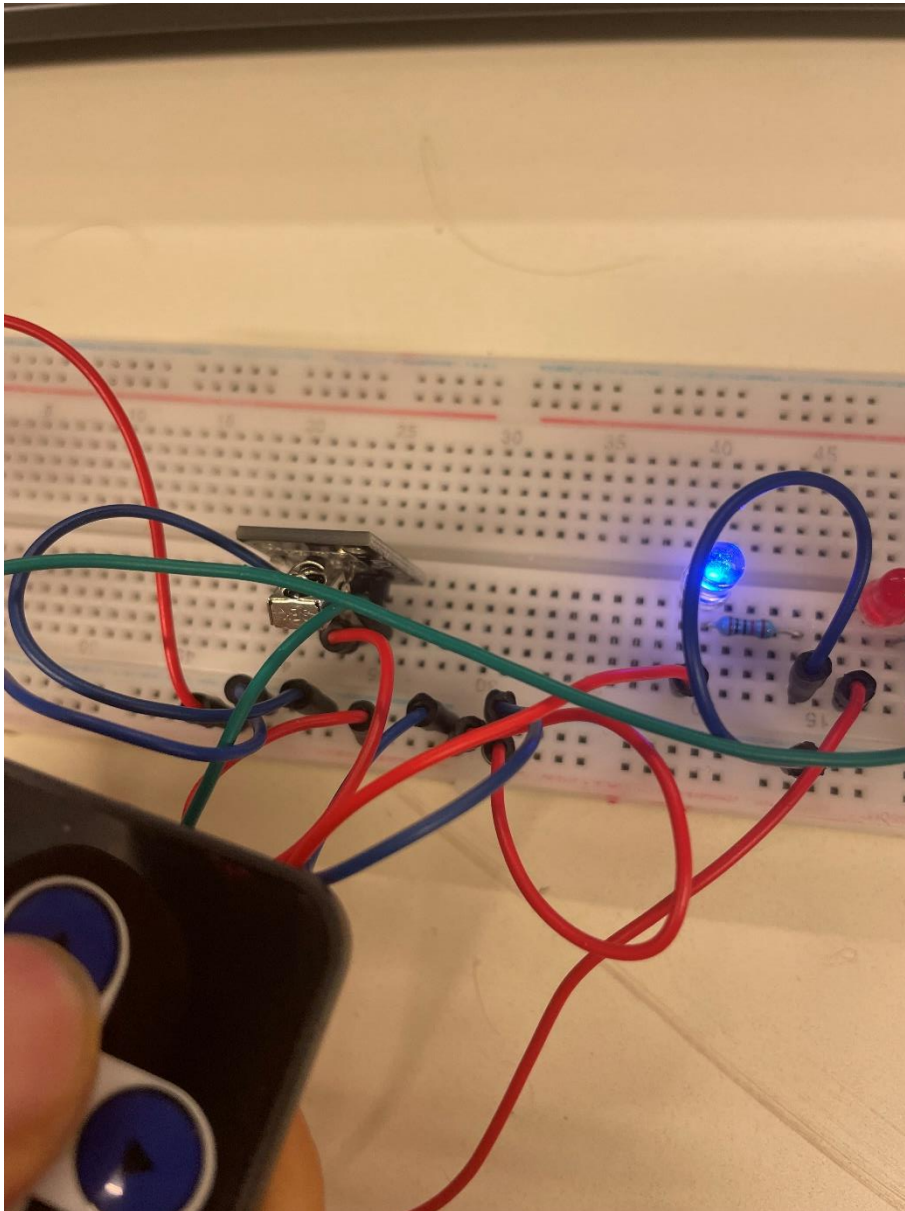
  if (results.value == HAUT_BUTTON) {
    digitalWrite(led, HIGH);
    delay(1000);
    Serial.println("testHt");
  }
}

```

J'ai ensuite codé un « if » qui me permet de dire, si le résultat que l'on obtient lorsqu'on appuie sur la touche de la flèche du haut est le même que celui instancié dans la variable en base hexadécimale, alors on allume la led pendant un temps défini. Or cela n'a pas fonctionné.

J'ai donc mis un `serial.print(test)` quand j'appuie sur le bouton OK, or j'ai eu un résultat mais bref car quand j'ai essayé une deuxième fois, je n'ai plus eu le « test » affiché sur le moniteur.

J'ai donc essayé de mettre le code brut directement du bouton du haut et la la lampe fonctionne bel et bien :





```
laser | Arduino 1.8.16
Fichier Édition Croquis Outils Aide

laser

#include <IRremote.h>
//HEX hex =0;
int led = 2;
int RECV_PIN = 11; // Définition de la broche Signal du récepteur IR raccordé sur la carte
IRrecv irrecv (RECV_PIN); // Déclaration du module IR
decode_results results;

#define OK_BUTTON 0xFF02FD
const int BAS_BUTTON = 0xFFA857;
const int Droite_BUTTON = 0xFFC23D;
const int Gauche_BUTTON = 0xFF22DD;
#define HAUT_BUTTON 0xFF629D
const int UN_BUTTON = 0xFF6897;
const int DEUX_BUTTON = 0xFF9867;
const int TROIS_BUTTON = 0xFFB04F;
const int QUATRE_BUTTON = 0xFF30CF;
const int CINQ_BUTTON = 0xFF18E7;
const int SIX_BUTTON = 0xFF7A85;
const int SEPT_BUTTON = 0xFF10EF;
const int HUIT_BUTTON = 0xFF38C7;
const int NEUF_BUTTON = 0xFF5AA5;
const int *_BUTTON = 0xFF42BD;
const int ZERO_BUTTON = 0xFF4AB5;
const int HASHTAG_BUTTON = 0xFF52AD;

void setup ()
{
  Serial.begin (9600); // Démarre la communication série
  irrecv.enableIRIn (); // Démarrage du récepteur IR
  pinMode(led, OUTPUT);
}

void loop () {

  //Serial.print(digitalRead(RECV_PIN));
  //delay(1000);
  //delay(1000);
  if (irrecv.decode (& results)) {
    Serial.print ("results.value");Serial.println (results.value, HEX);
    if (results.value == OK_BUTTON)
      Serial.println("testok");

    if (results.value == HAUT_BUTTON) {
      digitalWrite(led, HIGH);
      delay(1000);
      Serial.println("testHt");
    }
    Serial.println (results.value, HEX); // Affichage de la valeur reçue en hexadécimal
  }
}
```

Téléversement terminé

Le croquis utilise 5936 octets (18%) de l'espace de stockage de programmes. Le maximum est de 32256 oct  
Les variables globales utilisent 663 octets (32%) de mémoire dynamique, ce qui laisse 1385 octets pour

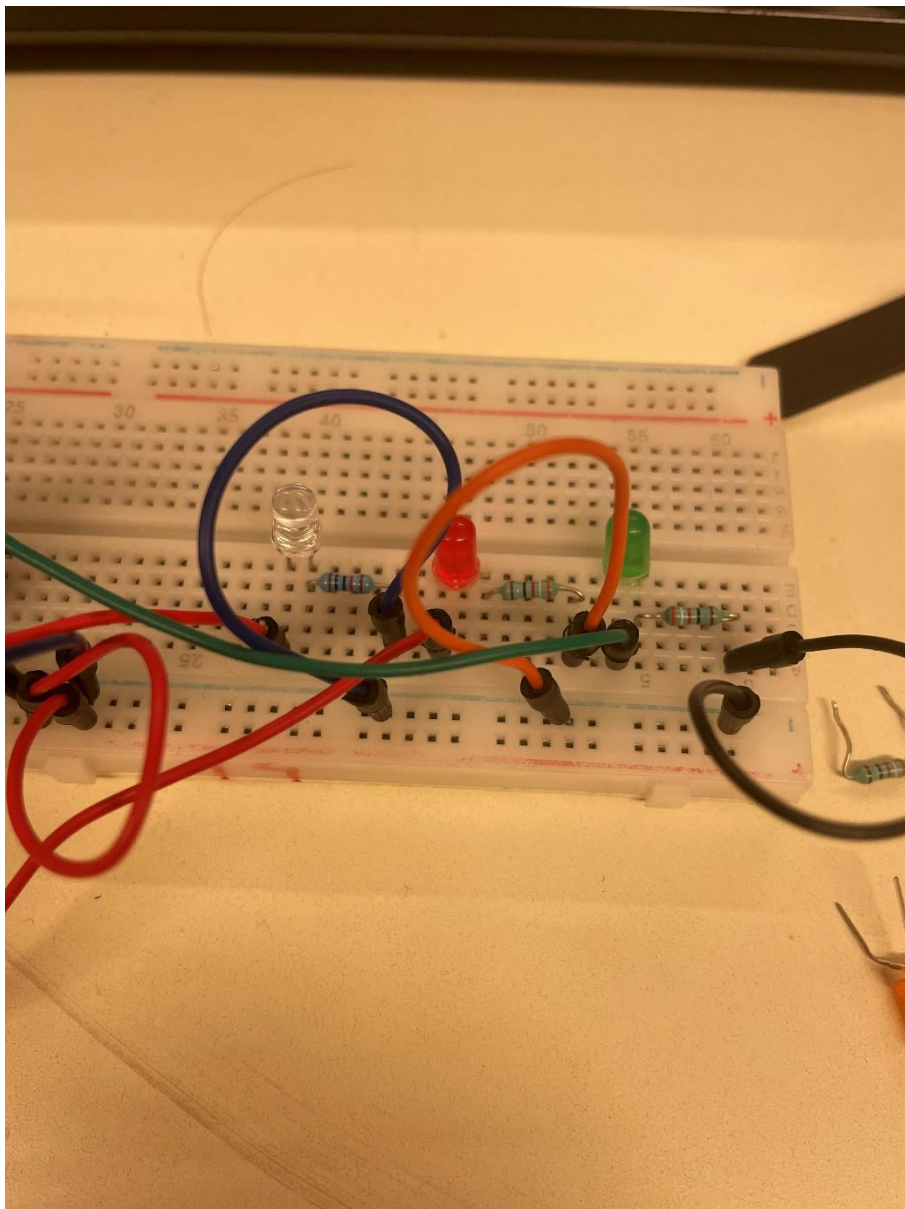
23 Arduino Uno sur COM4

Cela fonctionne pour un bouton mais est-ce que cela peut fonctionner pour plusieurs boutons ?

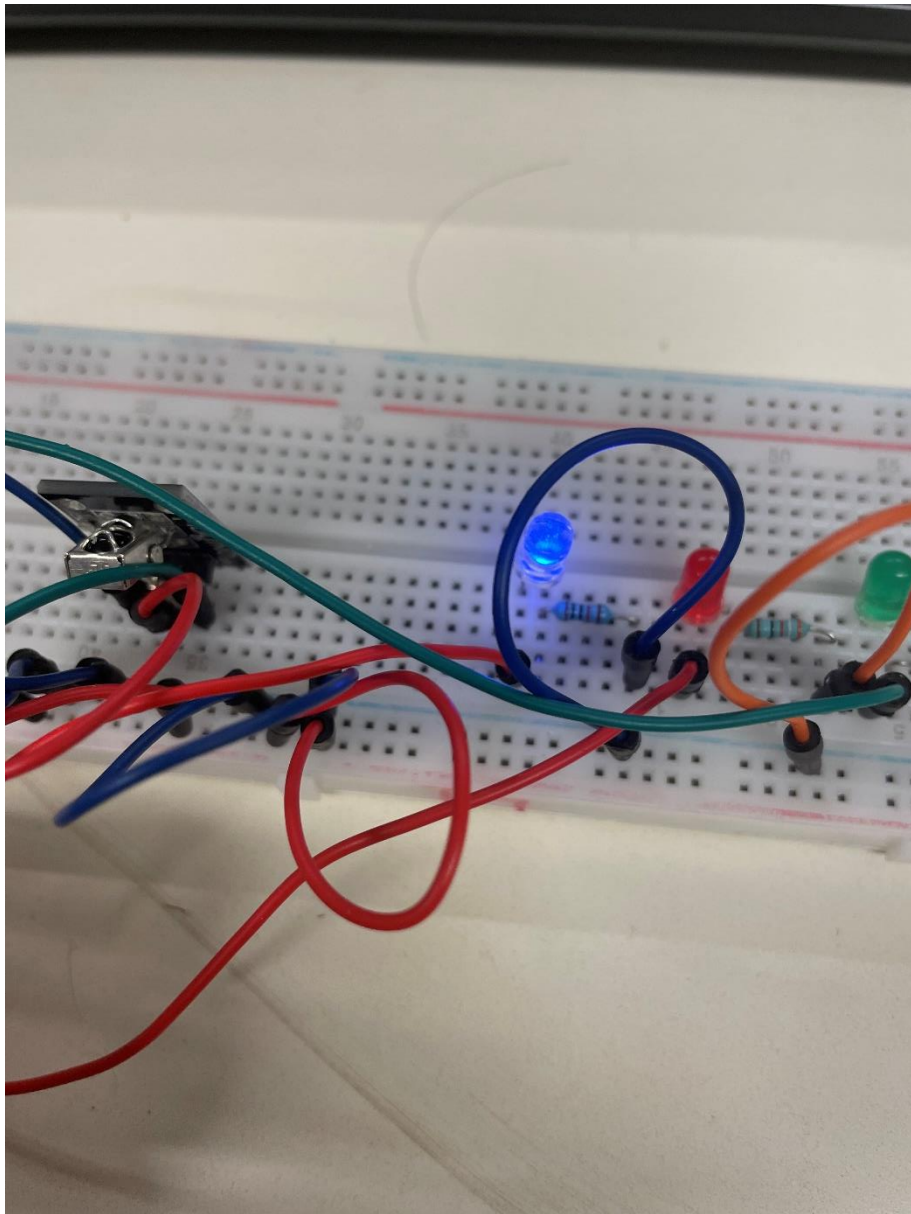
J'ai donc branché deux autres led que j'ai assignés aux touches du bas, touche du cote droit.

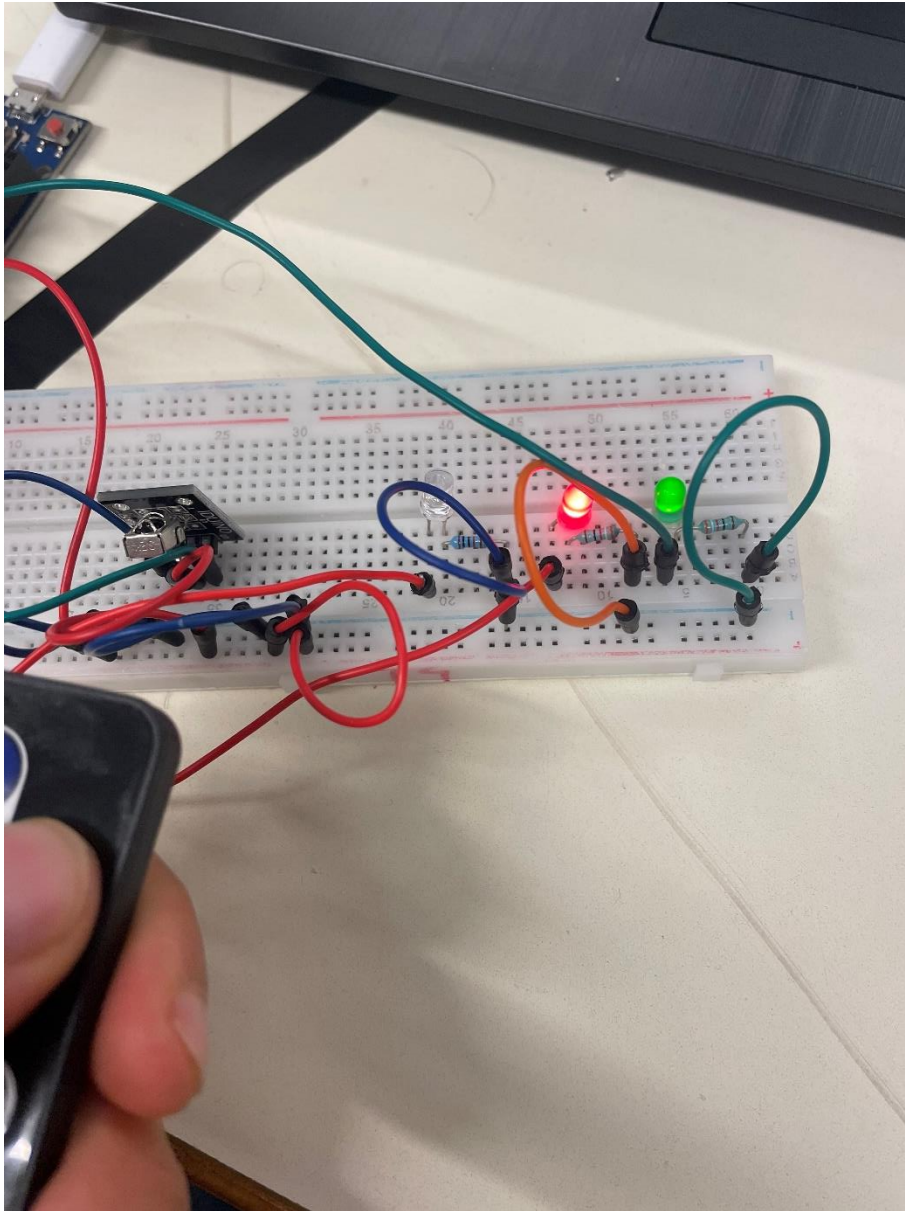
Le code est pratiquement le même que pour allumer la première led

Toutes les leds fonctionnent donc à perfection lorsque j'appuie sur une des touche











Seul problème mes leds rouges et vertes restaient allumées, j'ai très vite compris le problème et ai réglé cela à la maison.

```
void loop () {

  //Serial.print(digitalRead(RECV_PIN));
  //delay(1000);
  //delay(1000);
  //digitalWrite(led_g, HIGH);
  //delay(1000);
  //Serial.println("testHt");

  if (irrecv.decode (& results)) {
    Serial.print ("results.value");Serial.println (results.value, HEX);
    if (results.value == OK_BUTTON)
      Serial.println("testok");

    if (results.value == HAUT_BUTTON){
      digitalWrite(led,HIGH);
      delay(1000);
      Serial.println("testok1");}

    if (results.value == BAS_BUTTON){
      Serial.println("testok2");
      digitalWrite(led_r,HIGH);
      delay(1000);
    }

    if (results.value == DROITE_BUTTON){
      Serial.println("testok3");
      digitalWrite(led_g,HIGH);
      delay(1000);
    }

    Serial.println (results.value, HEX); // Affichage de la valeur reçue en hexadécimal
    irrecv.resume (); // Réception de la valeur suivante
    digitalWrite(led, LOW);
    digitalWrite(led_r, LOW);
    digitalWrite(led_g, LOW);
    delay(1000);

  }

}
```

---

On en conclu que l'on pourra mettre le code pouvant faire monter le dos d'âne avec la télécommande pour les véhicules prioritaires.

Je mets donc en pause pour l'instant cette partie.

Pour la prochaine séance je vais donc m'orienter vers cette fois ci le capteur de vitesse que je vais modéliser

avec le module à ultrason.