

## Compte rendu SAE :



Système de sécurité domestique

- **SOMMAIRE :**

- Introduction.....  
.....3
- Présentation des composants.....3
  - Capteur de mouvement (101020353)  
.....3
  - Capteur sonore (101020031)  
.....4
  - Module de communication FM (RF Émission + Réception 433MHz  
(113060000)  
.....5
  - Grove Mega  
shield.....6
  - Bon de  
commande.....6

- Réalisation du système de sécurité domestique.....7
  - Réalisation de la maquette.....7
  - Mise en fonctionnement du capteur de mouvement (101020353) .....8
  - Mise en fonctionnement du capteur sonore (101020031) .....9
  - Réalisation du programme permettant la sécurité de la maison.....10
  
- Gestion de projet.....
  
- Conclusion.....16
  - ACV.....16
  - Démonstration du fonctionnement.....
  - Conclusion.....16

- **Introduction :**

Le but de cette SAE est de réaliser un système qui permet de sécuriser une maison. Ce système devra permettre d'avertir le propriétaire de la maison en cas d'intrusion. Le système d'alarme se fera avec un microcontrôleur arduino mega et devra permettre une communication sans fils pour être averti d'une intrusion dans la maison à distance. Nous aurons aussi à notre disposition plusieurs capteur. Ce système sera implanté dans une maquette en bois qui modélisera la maison.

- **Présentation des composants :**



- Capteur de mouvement (101020353).

- Schéma de câblage :



- Pourquoi un tel choix ?

Pour notre système d'alarme nous avons choisi que le moyen de détection primaire soit le mouvement. En effet il nous a semblait logique que lors d'une intrusion le cambrioleur fera forcément des mouvements, donc sa présence sera obligatoirement détectée.

- Capteur sonore (101020031).



- Schéma de câblage :



- Pourquoi un tel choix ?

Nous avons choisi de coupler les capteurs de mouvement avec un capteur sonore pour notre système d'alarme. En effet il nous a semblé judicieux de ne pas avoir qu'un type de capteur dans un système de sécurité domestique car si le cambrioleur arrive à parer les capteurs de mouvement ce sera alors le son qu'il émettra qu'il le fera détecter. Nous voyons ça comme une double sécurité.

- Module de communication FM (RF Émission + Réception 433MHz (113060000)) .



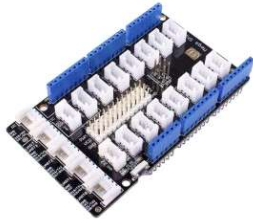
## Émetteur

## Récepteur

- Pourquoi un tel choix ?

Nous avons décidé d'utiliser comme moyen de communication la communication FM car cette dernière est plutôt facile à mettre en place avec Arduino et les module d'émission et de réception (113060000). Et ce moyen permet quand même d'envoyer des messages à une distance de 30 mètres

- Grove Mega shield.



- Pourquoi un tel choix ?

Nous avons choisi d'utiliser ce shield Arduino pour rendre le câblage plus facile. En effet avec ce shield chaque port est relié à la masse (GND) et les 5V arduino. Ce qui nous permettra de faire un câblage plus propre et plus lisible.

- Bon de commande.

Composant	Quantité	Prix (unité)
Capteur de mouvement (101020353)	2	6,8
Capteur sonore (101020031)	1	16,98
RF Émission + Réception 433MHz (113060000)	1	13,9
Arduino mega	1	42
Arduino uno	1	26,35
Grove Mega shield	1	9,9
<b>Total</b>		<b>122,73</b>



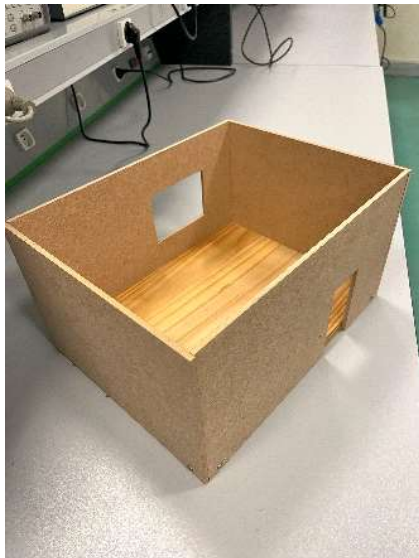
- Réalisation du système de sécurité domestique.

- Capteur de mouvement (101020353).
- Matériau utilisé :

L'idée première de la maquette que nous avons dû réaliser était qu'elle devait modéliser une maison d'une pièce et devras comporter 2 voie d'intrusion possible. La maquette devait pouvoir se démonter pour que le stockage de de cette dernière soit moins contraignant et ainsi pouvoir réutiliser cette maquette plusieurs année. C'est pourquoi nous avons décidé d'utiliser du carton de bois de 5 mm d'épaisseur pour les murs et une planche de 20 mm d'épaisseur pour le sol de la maison.

- Schéma de la maquette :

La maquette imaginée répond bien au cahier des charges : le matériau utilisé est du bois, la maison ne comporte qu'une pièce et il y a 2 voies d'intrusion possible (fenêtre, porte). Pour le côté démontable de la maquette, des vis seront utilisées pour fixer les panneaux de carton de bois à la base, cela permettra de pouvoir les dévisser et ainsi rendre plus simple le stockage des maquettes.



Maquette finalisé :

- Mise en fonctionnement du capteur de mouvement (101020353).

Pour faire marcher le capteur de mouvement le protocole est simple : on branche le capteur sur la broche A0 de notre Shield et on téléverse le code de test suivant.

```
#include <Wire.h> // Inclure la bibliothèque Wire pour la communication

void setup()
{
  pinMode(A0, INPUT); // Définir la broche du capteur PIR comme entrée
  Serial.begin(9600); // Initialiser la communication série à 9600 baud
}

void loop()
{
  int motionDetected = analogRead(A0); // Lire l'état du capteur PIR

  if (motionDetected == HIGH)
  {
    Serial.println("Mouvement détecté !");
  }
  else
  {
    Serial.println("Pas de mouvement détecté");
  }
}
```

- Code :

Une fois le code téléversé on a bien sur le terminal le message « pas de mouvement détecté » si on ne fait aucun mouvement devant le capteur. Si on fait un mouvement devant le capteur on a bien sur le terminal « mouvement détecté » qui s'affiche ».

- Mise en fonctionnement du capteur sonore (101020031).

Pour faire marcher le capteur de mouvement le protocole est simple : on branche le capteur sur la broche A4 de notre Shield et on téléverse le code de test suivant.

```
#include <Wire.h> // Inclure la bibliothèque Wire pour la communication I2C

void setup()
{
  pinMode(A4, INPUT); // Définir la broche du capteur comme entrée
  Serial.begin(9600); // Initialiser la communication série à 9600 bauds
}

void loop()
{
  int soundDetected = analogRead(A4); // Lire l'état du capteur
  if (soundDetected > 300)
  {
    Serial.println("Son détecté !");
  }
}
```

Une fois le code téléversé on a bien aucun message qui s'affiche cependant si la variable qui est image du son dépasse le seuil de 300 alors on estime que du son est émis. Donc si on fait un bruit supérieur à 300 on a bien le message son détecté qui s'affiche sur le terminal.

- Réalisation du programme permettant la sécurité de la maison.
- Code :

```
#include <Wire.h> // Inclusion de la bibliothèque Wire

#include <VirtualWire.h> // Inclusion de la bibliothèque VirtualWire


int RF_TX_PIN = 2; // Broche de l'émetteur

int Etat = 0; // Inclusion d'une variable Etat et l'initialise à 0

int Code = 0; // Inclusion d'une variable Code et l'initialise à 0


const char *msg; // Inclusion d'un pointeur, pour le message à transmettre


const String codeCorrect = "1234"; // Mot de passe d'activation de l'alarme

const String codeDesactivation = "1234"; // Mot de passe désactivation de l'alarme


const int pinLED = 13; // Broche pour la LED


void setup()

{
```

```

    pinMode(A0, INPUT); // Configuration de A0 en tant qu'entrée pour 1 capteur de
mouvement au niveau de la porte

    pinMode(A2, INPUT); // Configuration de A2 en tant qu'entrée pour 1 capteur de
mouvement au niveau de la fenêtre

    pinMode(A4, INPUT); // Configuration de A4 en tant qu'entrée pour 1 capteur sonore

    pinMode(pinLED, OUTPUT); // Configuration de la broche de la LED en tant que sortie


    Serial.begin(9600); // Initialisation de la communication série à 9600 bauds


    vw_set_tx_pin(RF_TX_PIN); // Configuration de la broche de transmission de
l'émetteur

    vw_setup(2000); // Configuration de la vitesse de transmission de l'émetteur
}

void loop()
{
    Principal(); // Appel la fonction principale du code
}

void Principal()
{
    if (Code==0) // Si le code est à 0
    {
        Serial.println("Entrez le code :"); // Affiche "Entrez le code :" sur le
moniteur série de l'émetteur

        Code = 1; // Met le code à 1
    }

    if ((Serial.available() > 0)&&(Code==1)) // Si des données sont disponibles sur la
communication série et le code est à 1
    {
        String inputCode = Serial.readStringUntil('\n'); // Lire la communication série

        inputCode.trim(); // Supprimer les espaces et les sauts de ligne
    }
}

```

```

    if (inputCode.equals(codeCorrect)) // Si le code saisi est correct
    {
        Serial.println("Code correct !"); // Affiche "Code correct !" sur le moniteur
        série de l'émetteur

        Serial.println("Activation de l'alarme ."); // Affiche "Activation de
        l'alarme ." sur le moniteur série de l'émetteur

        msg = "Activation de l'alarme ."; // Affiche "Activation de l'alarme ." sur le
        moniteur série du récepteur

        vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

        delay(10000); // Attendre 10 secondes le temps que la personne sorte de la
        maison suite à l'activation de l'alarme

        Serial.println("Alarme activée ."); // Afficher "Alarme activée ." sur le
        moniteur série de l'émetteur

        msg = "Alarme activée ."; // Afficher "Alarme activée ." sur le moniteur série
        du récepteur

        vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

        msg = 0; // Réinitialiser le message pour éviter d'envoyer "Alarme activée ." en
        permanence

        Code = 2; // Met le code à 2
    }

    else // Si le code saisi est incorrect
    {
        Serial.println("Code incorrect. Veuillez réessayer :"); // Afficher "Code
        incorrect. Veuillez réessayer :" sur le moniteur série de l'émetteur
    }
}

else if (Code==2) // Si le code est à 2
{
    if (desactiverAlarme()) // Si la fonction de désactivation de l'alarme est
    effectuée
    {
        Code = 0; // Met le code à 0

        Serial.println("Alarme désactivée ."); // Afficher "Alarme désactivée ." sur le
        moniteur série de l'émetteur

        msg = "Alarme désactivée ."; // Afficher "Alarme désactivée ." sur le moniteur
        série du récepteur
    }
}

```

```

    vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

    msg = "_"; // Définir le message comme "_" sur le moniteur série du récepteur

    vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

    Serial.println(); // Saute une ligne dans la communication série de l'émetteur

    Serial.println(); // Saute une ligne dans la communication série de l'émetteur

    Serial.println(); // Saute une ligne dans la communication série de l'émetteur

    Serial.println(); // Saute une ligne dans la communication série de l'émetteur
}

else // Si la fonction de désactivation de l'alarme n'est pas effectuée

{

    digitalWrite(pinLED, HIGH); // Allume la LED

    Detection_de_mouvement(); // Appel la fonction de détection de mouvement

    Detection_sonore(); // Appel la fonction de détection de son

}

}

}

void Detection_de_mouvement()

{

    int motionDetectedporte = digitalRead(A0); // Lire l'état de la broche A0

    int motionDetectedfenetre = digitalRead(A2); // Lire l'état de la broche A2


    if ((motionDetectedporte == HIGH)&&(Etat==0)) // Si un mouvement est détecté à la
    porte et l'état est à 0

    {

        Serial.println("Mouvement détecté au niveau de la porte !"); // Afficher
        "Mouvement détecté au niveau de la porte !" sur le moniteur série de l'émetteur

        msg = "Mouvement porte !"; // Afficher "Mouvement porte !" sur le moniteur série
        du récepteur

        vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

        Etat = 1; // Met l'état à 1

    }

    if ((motionDetectedfenetre == HIGH)&&(Etat==0)) // Si un mouvement est détecté à la

```



```

fenêtre et l'état est à 0

{

    Serial.println("Mouvement détecté au niveau de la fenetre !"); // Afficher
    "Mouvement détecté au niveau de la fenetre !" sur le moniteur série de l'émetteur

    msg = "Mouvement fenetre !"; // Afficher "Mouvement fenetre !" sur le moniteur
    série du récepteur

    vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

    Etat = 1; // Met l'état à 1

}

else if (Etat==1) // Si l'état est à 1

{

    Serial.println("ALARME !!!"); // Afficher "ALARME !!!" sur le moniteur série de
    l'émetteur

    msg = "ALARME !!!"; // Afficher "ALARME !!!" sur le moniteur série du récepteur

    vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

    LED_Intrusion(); // Appeler la fonction pour faire clignoter la LED

}

}

void Detection_sonore()

{

    int soundDetected = analogRead(A4); // Lire l'état de la broche A4

    if (soundDetected > 300) // Si le son est détecté (la valeur est supérieure à 300)

    {

        Serial.println("Son détecté !"); // Afficher "Son détecté !" sur le moniteur série
        de l'émetteur

        msg = "Son détecté !"; // Afficher "Son détecté !" sur le moniteur série du
        récepteur

        vw_send((uint8_t *)msg, strlen(msg)); // Envoie le message msg au récepteur

        Etat = 1; // Met l'état à 1

    }

}

```

```

bool desactiverAlarme()
{
    if (Serial.available() > 0) // Si des données sont disponibles sur la communication
série
    {
        String inputCode = Serial.readStringUntil('\n'); // Lire la communication série
        inputCode.trim(); // Supprimer les espaces et les sauts de ligne

        return inputCode.equals(codeDesactivation); // Renvoie vrai si le code correspond
au code de désactivation
    }

    return false; // Renvoyer faux sinon
}

void LED_Intrusion()
{
    digitalWrite(pinLED, HIGH); // Allume la LED

    delay(100); // Attendre 0,1 seconde

    digitalWrite(pinLED, LOW); // Éteindre la LED

    delay(100); // Attendre 0,1 seconde
}

```

Une fois l'implantation des modules dans la maquette réalisée, nous avons 2 programmes, le programme de réception constitué d'un module de réception d'une carte arduino ou est implanter le programme de réception. La partie réception reçoit uniquement des messages de l'émetteur en l'affichant sur le moniteur série de l'arduino récepteur. Elle convertit le texte de l'hexadécimal reçu vers notre langage. Effectivement, l'émetteur envoie le message souhaiter tel que "Bonjour", cependant il le transmet en hexadécimal. La partie émission qui est la plus importante et intéressante est composée d'une carte arduino méga qui sur celle-ci comporte plusieurs modules. Le module d'émission qui permet d'envoyer des messages au récepteur, 2 capteurs de mouvements qui nous remontrons un 1 lorsqu'il détecte un mouvement sinon 0 et un capteur de son qui dépassé une certaine valeur signale le problème. Il est également constitué de la partie arduino avec tout le code. Le programme est lancé sur les deux cartes arduino, il faut ensuite ouvrir les deux moniteurs série de chaque carte, l'émission et la réception séparément. Une fois cela effectué sur le moniteur série de l'émetteur un code est

nécessaire pour activer le programme, l'alarme, sans le code rien ne se passe. Si le code entré dans le moniteur série est correct nous pouvons accéder au reste du programme, s'il est faux nous avons uniquement un message comme quoi le code est incorrect et nous pouvons réessayer. Une fois le bon code entré l'alarme s'initialise et s'active au bout de 10 secondes laissant le temps à l'habitant de sortir de la maison, toutes les actions effectuées sont écrites sur le moniteur série par le programme. Après l'activation de l'alarme, si un mouvement ou un son est détecté l'alarme s'enclenchera directement en nous signalant d'où provient l'erreur "son détecté" ou encore "mouvement au niveau de la porte" et nous affichera ensuite en continu "ALARME". Pour désactiver l'alarme une fois celle-ci activée il suffit de rentrer de nouveau le code dans le moniteur série de l'émetteur, ce qui désactivera l'alarme (comme une simulation si c'est la personne qui habite là qui rentre chez lui) en nous l'annonçant et en pressant un retour 5 fois à la ligne et en relançant le programme et en nous redemandant le code d'initialisation. Toutes les étapes sont bien sûr envoyées au récepteur. Une LED a été ajoutée. Lorsque l'alarme est active la LED s'allume et lorsqu'il y a une intrusion celle-ci clignote.

- Code permettant la réception à distance (Arduino Uno).

```

#include <VirtualWire.h>

int RF_RX_PIN = 2; // Broche sur laquelle est connecté la réception

void setup()
{
    Serial.begin(9600); // Initialise la communication série à 9600 bauds

    vw_set_rx_pin(RF_RX_PIN); // Configure la broche du récepteur radio
    vw_setup(2000); // Initialise VirtualWire avec une vitesse de 2000 bits par seconde
    vw_rx_start(); // Démarre la réception
}

void loop()
{
    uint8_t buf[VW_MAX_MESSAGE_LEN]; // Crée un tableau pour stocker les données reçues
    uint8_t buflen = VW_MAX_MESSAGE_LEN; // Définit la longueur maximale des données

    if(vw_get_message(buf, &buflen)) // Vérifie s'il y a un message
    {
        int i; // Création d'une variable i

        for(i = 0; i < buflen; ++i)
        {
            if (char(buf[i]) == '_') // Si _ est reçu
            {
                Serial.println(); // Sauter une ligne sur le moniteur série
                Serial.println(); // Sauter une ligne sur le moniteur série
                Serial.println(); // Sauter une ligne sur le moniteur série
                i += 1; // Passe au caractère suivant après le _
            }
            else
            {
                Serial.print(char(buf[i])); // Affiche le caractère
            }
        }
        Serial.println(""); // Passe à la ligne après l'affichage du message
    }
}

```

- Conclusion

- ACV

Pour limiter l'impact sur l'environnement de notre projet. Nous avons choisi d'utiliser que des matériaux capables d'être réutilisé dans d'autres projets. Par exemples nous avons choisit d'utilisé des planches de bois pour la maquette afin de stocker celles-ci pour les réutiliser dans d'autre projet et ne pas les jeter à la fin de notre projet. Nous avons aussi programmé une carte arduino car cette dernière une fois notre projet fini pourra être réutilisé. Sur le même principe nous avons choisis des modules arduino car ces derniers pourront être réutilisé.

- Conclusion

Pour conclure cette SAE nous a été très intéressante, une nouvelle façon d'utiliser notre parcours, dans la sécurité électronique. De nouveaux composants et des études plus poussées sur ceux-ci pour l'environnement comme si nous étions en entreprise afin d'économiser et de protéger le plus possible l'environnement. Nous étions assez familiers avec les capteurs cependant la transmission sans fil était très intéressante à utiliser