

```
//Les entrées / sorties utilisées
```

```
int dPinBuzzer = 2; // Fil rouge sur la sortie 11 et fil noir sur GND.
```

```
int dPinSRF05 = 13; // Echo Output sur l'entrée/sortie digitale 2.
```

```
int dPintrigger=9; // signal PWM appliqué à l'entrer Reset du NE555.
```

```
//int distanceMin = 5; // en cm, vous pouvez changer cette valeur pour changer la portée des  
ultrasons du capteur
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
// Initialisation
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
    pinMode(dPinBuzzer, OUTPUT);
```

```
    pinMode(dPintrigger,OUTPUT);
```

```
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
// Boucle principale
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
void loop()
```

```
{
```

```
    // int distanceObstacle = distanceMesuree();
```

```

// Serial.println(distanceObstacle);

// if (distanceObstacle < distanceMin) {

//   playNote('c', 10);

// }

    digitalWrite(dPintrigger, HIGH);

    delayMicroseconds(100); //10% duty cycle @ 1Kh=kHz

    digitalWrite(dPintrigger, LOW);

    delayMicroseconds(1000-100);

}

```

```

////////////////////////////////////
// Effectuer une mesure avec le capteur à ultrasons
////////////////////////////////////

```

```

int distanceMesuree()

{

// pinMode(dPinSRF05, OUTPUT);

// // Make sure pin is low before sending a short high to trigger ranging.

// digitalWrite(dPinSRF05, LOW);

// delayMicroseconds(2);

// // Send a short 10 microseconds high burst on pin to start ranging.

// digitalWrite(dPinSRF05, HIGH);

// delayMicroseconds(10);

// // Send pin low again before waiting for pulse back in.

// digitalWrite(dPinSRF05, LOW);

```

```

pinMode(dPinSRF05, INPUT);

// Reads echo pulse in from SRF05 in microseconds.

int duree = pulseIn(dPinSRF05, HIGH);

// On doit attendre 50 ms avant le prochain signal de déclenchement.

delay(50);


int distance = duree / 58; // On divise par 58 pour avoir une distance en cm.

return(distance);
}


////////////////////////////////////

// Buzzer

////////////////////////////////////


void playTone(int tone, int duration)
{
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
    digitalWrite(dPinBuzzer, HIGH);
    delayMicroseconds(tone);
    digitalWrite(dPinBuzzer, LOW);
    delayMicroseconds(tone);
  }
}


void playNote(char note, int duration)
{

```

```
char names[ ] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' }; // Do Ré Mi Fa Sol La Si Do

int tones[ ] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 }; // Demi-période en ms

// Play the tone corresponding to the note name

for (int i = 0; i < 8; i++) {

    if (names[i] == note) {

        playTone(tones[i], duration);

    }

}

}
```

```
////////////////////////////////////
```