

# Conception & Développement de solution embarquée

## - Atelier 1

---

Auteur : Jérémie LAERA

Intervenant : Sylvain LABASSE

Plateforme: Tinkercad

## Code source de la solution proposée

---

```
// Taking into account: time eloping when it is a dot or
// a line

const char* morse_mappings[] = {
    ".- ", //a
    "-... ", //b
    "-.-. ", //c
    "-.. ", //d
    ". ", //e
    "-.-.- ", //f
    "--. ", //g
    ".... ", //h
    ".. ", //i
    ".--- ", //j
    "-.- ", //k
    "-.... ", //l
    "-- ", //m
    "-. ", //n
    "--- ", //o
    "-.-.- ", //p
    "--.- ", //q
    ".-. ", //r
    "... ", //s
    "- ", //t
    "-.- ", //u
    "-.-.- ", //v
    ".-- ", //w
    "-.-.- ", //x
    "-.-.- ", //y
    "-.-.- " //z
};

const char* alphabet = "abcdefghijklmnopqrstuvwxyz";

// units of time
const int TIME_UNIT = 250;

// morse code rules of time unit depending on the charPPacter
const int DOT = TIME_UNIT;
```

```
const int DASH = 3 * TIME_UNIT;
const int SYMBOL_SPACE = TIME_UNIT;
const int LETTER_SPACE = 3 * TIME_UNIT - SYMBOL_SPACE;
const int WORD_SPACE = 7 * TIME_UNIT - LETTER_SPACE;

const char* message_to_send = "Captain Falcon\0";
const char delimiter[] = " \0";
//char *token;
//token = strtok(message_to_send, delimiter);
//const int message_length = strlen(token);
const int message_length = strlen(message_to_send);

void setup()
{
    pinMode(13, OUTPUT);
}

void loop()
{
    int i;
    for (i = 0; i < message_length; i++)
    {
        // enforcing lowercase letters
        const char* current_character = strchr(alphabet,
        tolower(message_to_send[i]));
        if (current_character != NULL)
        {
            int index = (int)(current_character - alphabet);
            const char* morse_symbols = morse_mappings[index];
            int count = strlen(morse_symbols);

            int j;
            for (j = 0; j < count; j++)
            {
                digitalWrite(13, HIGH);

                int symbol_time;
                char symbol = morse_symbols[j];
                if (symbol == '.')
                    symbol_time = DOT;
                else
                    symbol_time = DASH;

                delay(symbol_time);
                digitalWrite(13, LOW);
                delay(SYMBOL_SPACE);
            }
            delay(LETTER_SPACE);
        }
    }
    delay(WORD_SPACE);
}
```

