

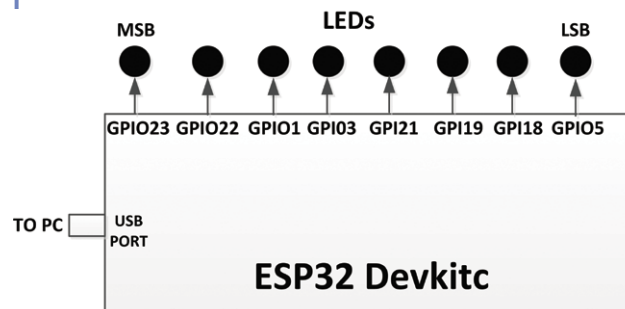
# chenillard *Knight Rider* avec l'ESP32

Dogan Ibrahim (Royaume-Uni)

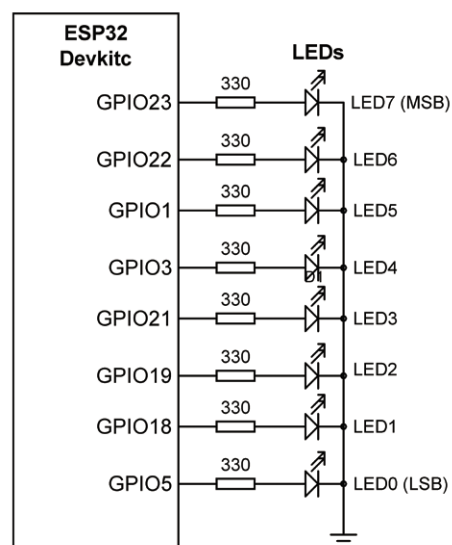
*Knight Rider* est une série TV, avec une voiture appelée KITT qui parle et se pilote toute seule. Beaucoup se souviennent de l'effet de va-et-vient lumineux sur la calandre de la voiture associé à un son du genre... *zoom zoom*. Dans le jargon, cet effet lumineux s'appelle un chenillard : des loupes placées côte à côte s'allument une par une dans une direction, puis dans la direction inverse, et le mouvement se répète à l'infini. Dans ce petit projet, huit LED connectées à un ESP32 DevKitC simulent le va-et-vient du point lumineux sur le capot de KITT. La **fig. 1** donne le schéma de principe, la **fig. 2** le circuit. Les 8 LED sont connectées aux ports GPIO par des résistances de limitation de courant de 330  $\Omega$ . C'est facile à construire sur une plaque d'essais (**fig. 3**).

Le programme est donné en PDL (*Project Description Language*) (**listage 1**) et dans sa mise en œuvre pratique très simple (**listage 2**). Au début, un tableau (*array*) de *LED* est créé pour y stocker les numéros de port des LED. Ensuite sont configurés en sorties les ports GPIO

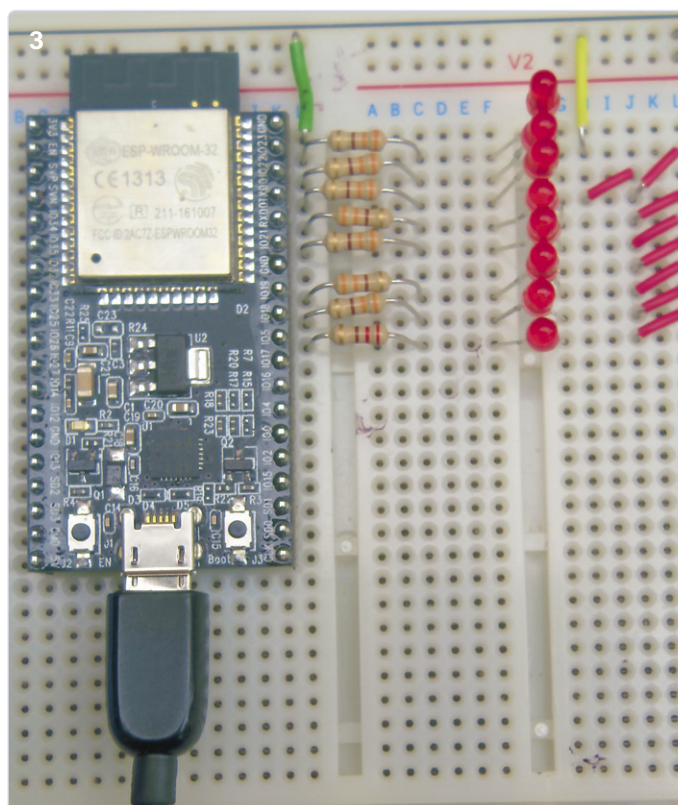
1



2



3



## Listage 1. Knight Rider PDL

### BEGIN

Store LED port numbers in array LEDs

Configure LED port pins as outputs

### DO FOREVER

Do k From 0 to 8

Turn ON LED at index LEDs[k]

Wait 100 ms

Turn OFF LED at index LEDs[k]

### ENDDO

DO k From 6 to 0

TURN ON LED at index LEDs[k]

Wait 100 ms

Turn OFF LED at index LEDs[k]

### ENDDO

### ENDDO

### END

## LIEN

[1] **téléchargement du programme :**  
[www.elektormagazine.fr/200234-02](http://www.elektormagazine.fr/200234-02)

## Listage 2. Code «Knight Rider» pour ESP32

```

/*****
 * chenillard KNIGHT RIDER
 * =====
 *
 * 8 LED connectées à GPIO23, GPIO22, GPIO1, GPIO3,
 * GPIO21, GPIO19, GPIO18 et GPIO5 de l'ESP32
 DevKitC.
 * Le programme simule le va-et-vient du chenillard
 * de la voiture KITT dans la série TV Knight Rider.
 *
 * Program: KnightRider
 *
 *****/
int LEDs[] = {23, 22, 1, 3, 21, 19, 18, 5};
unsigned char Count = 0;
unsigned char del = 100;


//
// GPIO 23,22,1,3,21,19,18,5 sont des sorties
//
void setup()
{
    unsigned char i;
    for(i=0; i < 8; i++)
    {
        pinMode(LEDs[i], OUTPUT);
    }
}

//
// les LED clignotent
// comme celles de Knight Rider
//
void loop()
{
    for(int k = 0; k < 8; k++)
    {
        digitalWrite(LEDs[k], HIGH);
        delay(del);
        digitalWrite(LEDs[k], LOW);
    }

    for(int k = 6; k > 0; k--)
    {
        digitalWrite(LEDs[k], HIGH);
        delay(del);
        digitalWrite(LEDs[k], LOW);
    }
}

```

auxquels sont connectées les LED. Le programme comporte deux boucles **for**. Dans l'une, les LED sont allumées chacune pendant 100 ms dans l'ordre des bits du poids le plus fort au poids le plus faible. Dans l'autre, ces LED sont allumées pendant 100 ms dans l'ordre inverse. C'est ainsi qu'est obtenu l'effet de chenillard et de va-et-vient.

Le programme est téléchargeable [1]. Essayez de le modifier, par exemple en jouant sur la durée d'activité de chaque sortie (maintenant 100 ms). Amusez-vous à créer d'autres effets. 

200234-02