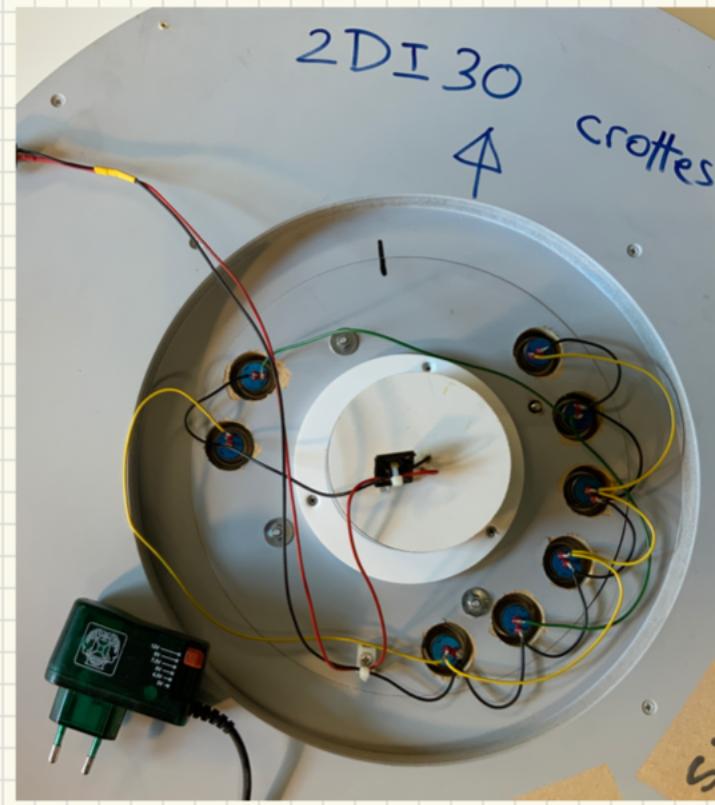
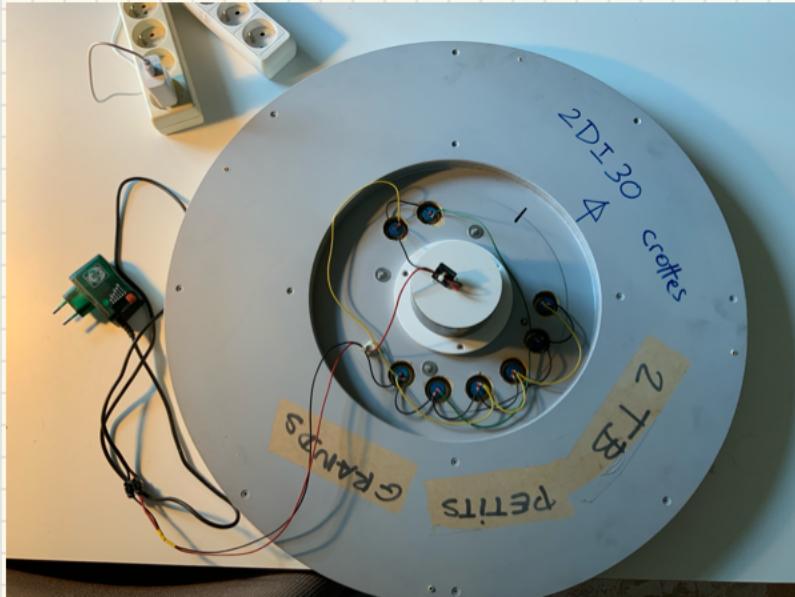
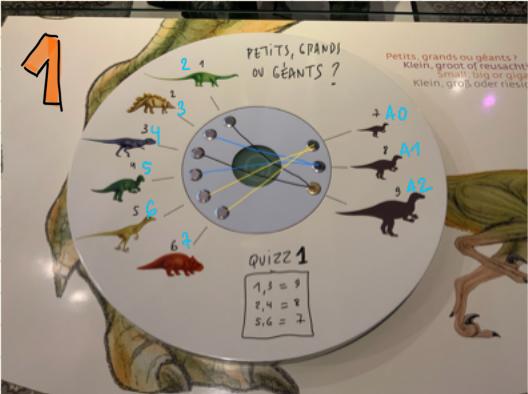


ANCIEN SYSTÈME D'ORIGINE.



QUIZZ DINO



A0 = 6,7
A1 = 3,5
A2 = 2,4

PINS / SOLUTIONS

```
const int questions[] = {A0, A1, A2}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5, 6, 7}; // Boutons des réponses
```

```
const int correspondances[][6] = {
    {-1, -1, -1, -1, 6, 7}, // Question 1
    {-1, 3, -1, 5, -1, -1} // Question 2
    {2, -1, 4, -1, -1, -1}, // Question 3
```

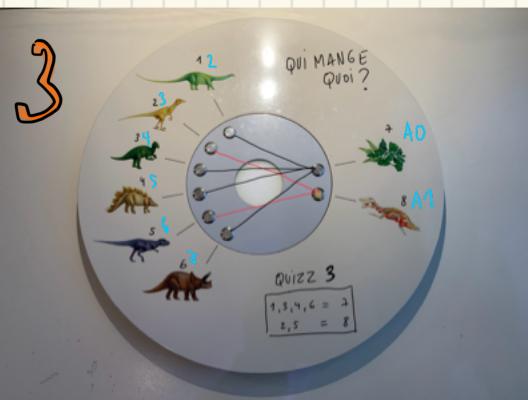


A0 = 3,6
A1 = 2,4,5,7

PINS / SOLUTIONS

```
const int questions[] = {A0, A1}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5, 6, 7}; // Boutons des réponses
```

```
const int correspondances[][6] = {
    {-1, 3, -1, -1, 6, -1}, // Question 1
    {2, -1, 4, 5, -1, 7} // Question 2
```



A0 = 2,4,5,7
A1 = 3,6

PINS / SOLUTIONS

```
const int questions[] = {A0, A1}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5, 6, 7}; // Boutons des réponses
```

```
const int correspondances[][6] = {
    {2, -1, 4, 5, -1, 7} // Question 1
    {-1, 3, -1, -1, 6, -1}, // Question 2
```



A0 = 3
A1 = 2
A2 = 4
A3 = 5

PINS / SOLUTIONS

```
const int questions[] = {A0, A1, A2, A3}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5}; // Boutons des réponses
```

```
const int correspondances[][4] = {
    {-1, 3, -1, -1} // Question 1
    {2, -1, -1, -1}, // Question 2
    {-1, -1, 4, -1}, // Question 3
    {-1, -1, -1, 5}, // Question 4
```



A0 = 3 A1 = 2 A2 = 7 A3 = 4 A4 = 6 A5 = 5

QUESTION 1 QUESTION 2 QUESTIONS QUESTION 4 QUESTION 5 QUESTION 6

```
const int questions[] = {A0, A1, A2, A3, A4, A5}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5, 6, 7}; // Boutons des réponses
```

```
const int correspondances[][6] = {
    {-1, 3, -1, -1, -1, -1} // Question 1
    {2, -1, -1, -1, -1, -1}, // Question 2
    {-1, -1, -1, -1, -1, 7}, // Question 3
    {-1, -1, 4, -1, -1, -1}, // Question 4
    {-1, -1, -1, -1, 6, -1}, // Question 5
    {-1, -1, -1, 5, -1, -1}, // Question 6
```

CODE GÉNÉRAL

ZONE À ADAPTER À CHAQUE QUIZZ

```
const int ledVerte = 10;
const int ledRouge = 11;

const int questions[] = {A2, 3}; // Boutons des questions
const int reponses[] = {5, 6, 7}; // Boutons des réponses

const int correspondances[][][3] = {
    {5, 6, -1}, // Question 1 a deux bonnes réponses
    {7, -1, -1} // Question 2 a une seule bonne réponse
};

const int nbQuestions = sizeof(questions) / sizeof(questions[0]);
const int nbReponses = sizeof(reponses) / sizeof(reponses[0]);
const int maxReponses = sizeof(correspondances[0]) / sizeof(correspondances[0][0]);

void setup() {
    pinMode(ledVerte, OUTPUT);
    pinMode(ledRouge, OUTPUT);

    for (int i = 0; i < nbQuestions; i++) {
        pinMode(questions[i], INPUT_PULLUP);
    }
    for (int i = 0; i < nbReponses; i++) {
        pinMode(reponses[i], INPUT_PULLUP);
    }
}

void loop() {
    int questionAppuyee = -1;
    int reponseAppuyee = -1;
    int nbBoutonsAppuyes = 0;

    for (int i = 0; i < nbQuestions; i++) {
        if (digitalRead(questions[i]) == LOW) {
            questionAppuyee = i;
            nbBoutonsAppuyes++;
        }
    }

    for (int j = 0; j < nbReponses; j++) {
        if (digitalRead(reponses[j]) == LOW) {
            reponseAppuyee = reponses[j];
            nbBoutonsAppuyes++;
        }
    }

    if (nbBoutonsAppuyes == 2 && questionAppuyee != -1 && reponseAppuyee != -1) {
        bool bonneReponse = false;
        for (int k = 0; k < maxReponses; k++) {
            if (correspondances[questionAppuyee][k] == reponseAppuyee) {
                bonneReponse = true;
                break;
            }
        }
        if (bonneReponse) {
            digitalWrite(ledVerte, HIGH);
            digitalWrite(ledRouge, LOW);
        } else {
            digitalWrite(ledVerte, LOW);
            digitalWrite(ledRouge, HIGH);
        }
        delay(300);
        digitalWrite(ledVerte, LOW);
        digitalWrite(ledRouge, LOW);
    } else {
        digitalWrite(ledVerte, LOW);
        digitalWrite(ledRouge, LOW);
    }
}
```

V2
À ADAPTER
POUR LE RING NEUF
(Voir + BAS)

Features and Benefits

- Control circuit and RGB chip are integrated in a package of 5050 components, form a complete control of pixel point.
- Built-in signal reshaping circuit, after wave reshaping to the next driver, ensure wave-form distortion not accumulate.
- Built-in electric reset circuit and power lost reset circuit.
- Each pixel of the three primary color can achieve 256 brightness display, completed 16777216 color full color display, and scan frequency not less than 400Hz/s.
- Cascading port transmission signal by single line.
- Any two point the distance more than 5m transmission signal without any increase circuit.
- When the refresh rate is 30fps, low speed model cascade number are not less than 512 points, high speed mode not less than 1024 points.
- Send data at speeds of 800Kbps.
- The color of the light were highly consistent, cost-effective..

Applications

- Full-color module, Full color soft lights a lamp strip.
- LED decorative lighting, Indoor/outdoor LED video irregular screen.

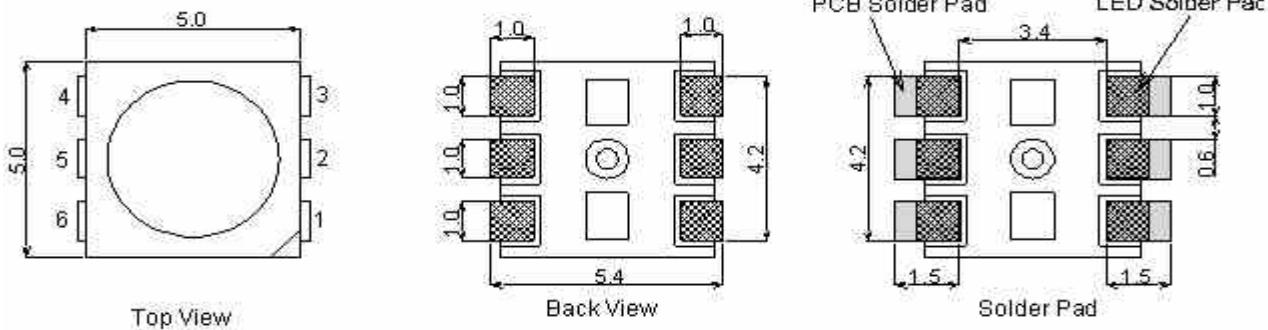
General description

WS2812 is a intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

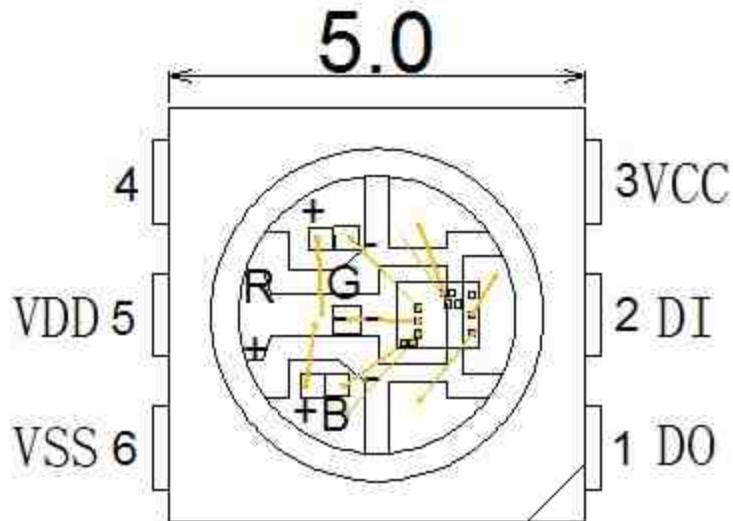
The data transfer protocol use single NZR communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel, the signal to reduce 24bit. pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.

LED with low driving voltage, environmental protection and energy saving, high brightness, scattering angle is large, good consistency, low power, long life and other advantages. The control chip integrated in LED above becoming more simple circuit, small volume, convenient installation.

Mechanical Dimensions



PIN configuration



PIN function

NO.	Symbol	Function description
1	DOUT	Control data signal output
2	DIN	Control data signal input
3	VCC	Power supply control circuit
4	NC	
5	VDD	Power supply LED
6	VSS	Ground

Absolute Maximum Ratings

<http://www.world-semi.com>

Parameter	Symbol	Ratings	Unit
Power supply voltage	V _{CC}	+6.0~+7.0	V
Power supply voltage	V _{DD}	+6.0~+7.0	V
Input voltage	V _I	-0.5~V _{DD} +0.5	V
Operation junction temperature	T _{opt}	-25~+80	°C
Storage temperature range	T _{stg}	-55~+150	°C

Electrical Characteristics (T_A=-20~+70°C, V_{DD}=4.5~5.5V, V_{SS}=0V, unless otherwise specified)

Parameter	Symbol	conditions	Min	Tpy	Max	Unit
Low voltage output current	I _{OL}	R _{OUT}	—	18.5	—	mA
	I _{dout}	V _O =0.4V, D _{OUT}	10	—	—	mA
Input current	I _I	V _I =V _{DD} /V _{SS}	—	—	±1	μA
Input voltage level	V _{IH}	D _{IN} , SET	0.7V _{DD}	—	—	V
	V _{IL}	D _{IN} , SET	—	—	0.3 V _{DD}	V
Hysteresis voltage	V _H	D _{IN} , SET	—	0.35	—	V

Switching characteristics (T_A=-20~+70°C, V_{DD}=4.5~5.5V, V_{SS}=0V, unless otherwise specified)

Parameter	Symbol	Condition	Min	Tpy	Max	Unit
Operation frequency	F _{osc2}	—	—	800	—	KHz
Transmission delay time	t _{PLZ}	CL=15pF,DIN→DOUT,RL=10KΩ	—	—	300	ns
Fall time	t _{THZ}	CL=300pF,OUTR/OUTG/OUTB	—	—	120	μs
Data transmission rate	F _{MAX}	Duty ratio50%	400	—	—	Kbps
Input capacity	C _I	—	—	—	15	pF



MESURE
 BENOIT
 75 WUX
 A 20 CM
 = 3 CANDELA

WS2812

Intelligent control LED integrated light source

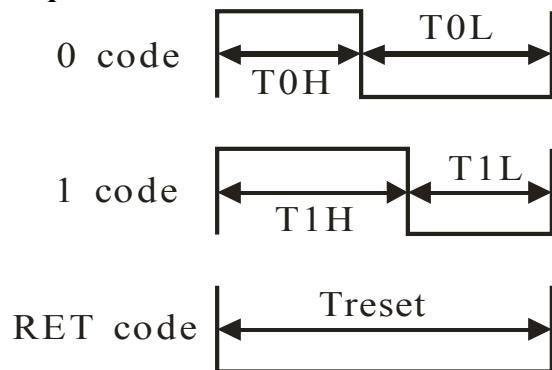
LED characteristic parameter

Emitting color	Wavelength(nm)	Luminous intensity(mcd)	Current(mA)	Voltage(V)
Red	620-630	550-700	20	1.8-2.2
Green	515-530	1100-1400	20	3.0-3.2
Blue	465-475	200-400	20	3.2-3.4

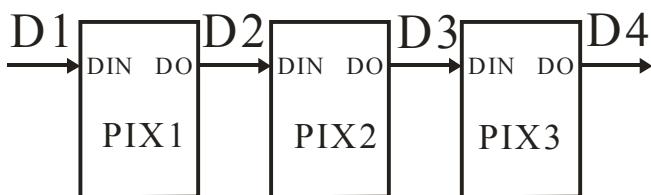
Data transfer time(TH+TL=1.25μs±600ns)

T0H	0 code ,high voltage time	0.35us	±150ns
T1H	1 code ,high voltage time	0.7us	±150ns
T0L	0 code , low voltage time	0.8us	±150ns
T1L	1 code ,low voltage time	0.6us	±150ns
RES	low voltage time	Above 50μs	

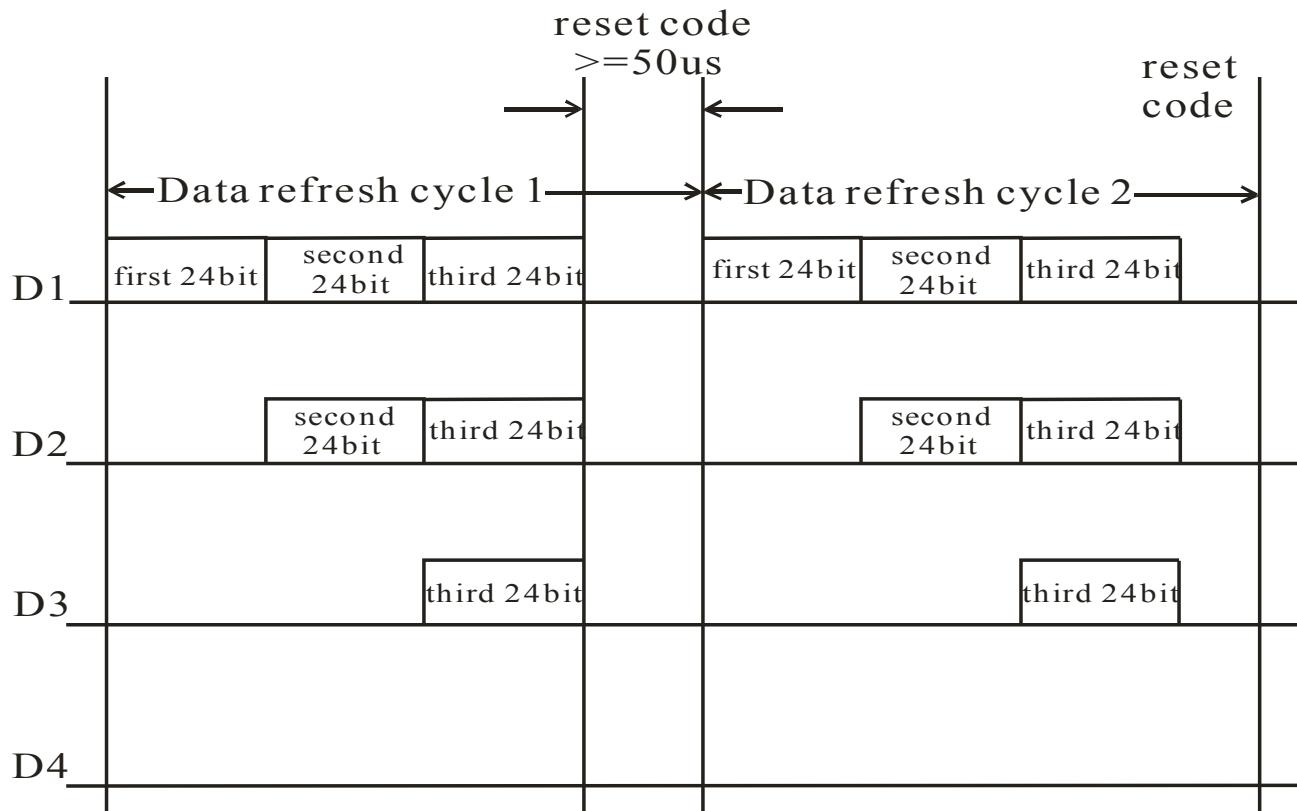
Sequence chart:



Cascade method:



Data transmission method:



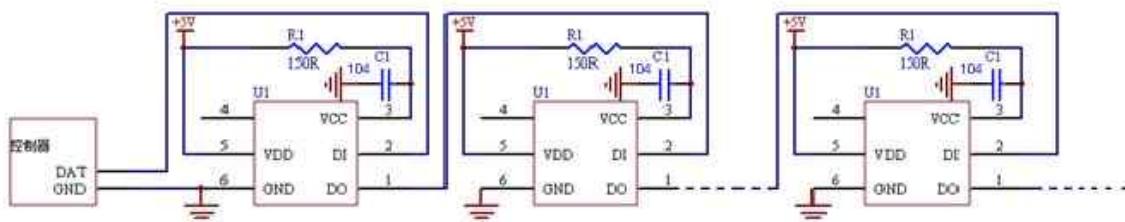
Note: The data of D1 is send by MCU, and D2, D3, D4 through pixel internal reshaping amplification to transmit.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Typical application circuit:



Mouser Electronics

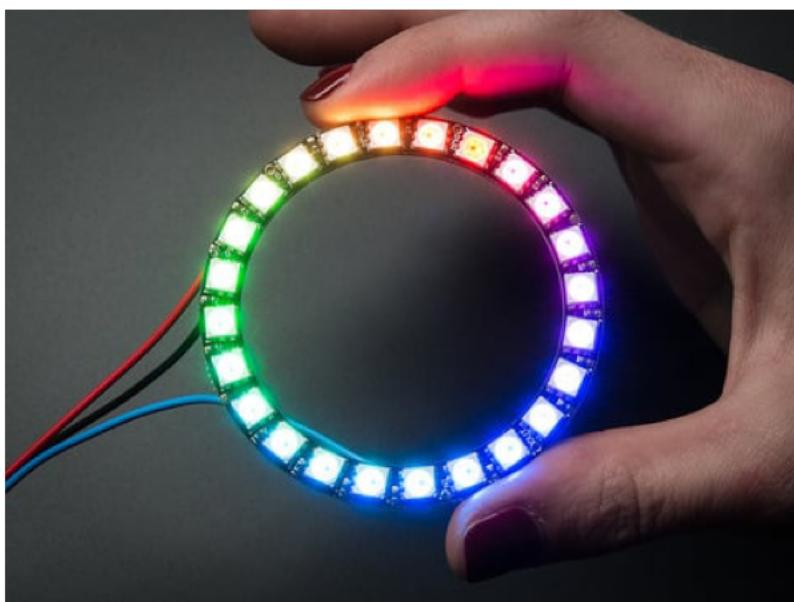
Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Adafruit:](#)

[1655](#) [1463](#) [2226](#) [1426](#) [1643](#) [1586](#) [2851](#) [2868](#) [2869](#) [2859](#) [2860](#) [2861](#) [2862](#) [2863](#) [2867](#) [2852](#) [2853](#) [2854](#)
[2855](#) [2856](#) [2858](#) [3094](#) [2826](#) [2844](#) [2833](#) [2825](#) [2827](#) [2834](#) [2839](#) [2840](#) [2835](#)

<https://www.adafruit.com/product/1586#description>



//QUIZZ 02 "QUI A FAIT CETTE CROTTE ?"

#include <Adafruit_NeoPixel.h>

```
const int numPixels = 24;
const int pixelPin = 10;
Adafruit_NeoPixel ring = Adafruit_NeoPixel(numPixels, pixelPin);
uint32_t VERT = ring.Color(0x00, 0xFF, 0x00); // r, g, b - vert
uint32_t ROUGE = ring.Color(0xFF, 0x00, 0x00); // rouge
uint32_t ETEINT = ring.Color(0x00, 0x00, 0x00); // eteint

const int questions[] = {A0, A1}; // Boutons des questions
const int reponses[] = {2, 3, 4, 5, 6, 7}; // Boutons des réponses
const int attendre = (5);

const int correspondances[][6] = {
    {-1, 3, -1, -1, 6, -1}, // Solutions question 1
    {2, -1, 4, 5, -1, 7} // Solutions question 2
};

const int nbQuestions = sizeof(questions) / sizeof(questions[0]);
const int nbReponses = sizeof(reponses) / sizeof(reponses[0]);
const int maxReponses = sizeof(correspondances[0]) / sizeof(correspondances[0][0]);

void setup() {
    Serial.begin(9600);
    pinMode(pixelPin, OUTPUT);

    for (int i = 0; i < nbQuestions; i++) {
        pinMode(questions[i], INPUT_PULLUP);
    }
    for (int i = 0; i < nbReponses; i++) {
        pinMode(reponses[i], INPUT_PULLUP);
    }
}

void loop() {
    int questionAppuyee = -1;
    int reponseAppuyee = -1;
    int nbBoutonsAppuyes = 0;

    for (int i = 0; i < nbQuestions; i++) {
        if (digitalRead(questions[i]) == LOW) {
            questionAppuyee = i;
            nbBoutonsAppuyes++;
        }
    }

    for (int j = 0; j < nbReponses; j++) {
        if (digitalRead(reponses[j]) == LOW) {
            reponseAppuyee = reponses[j];
            nbBoutonsAppuyes++;
        }
    }

    if (nbBoutonsAppuyes == 2 && questionAppuyee != -1 && reponseAppuyee != -1) {
        bool bonneReponse = false;
        for (int k = 0; k < maxReponses; k++) {
            if (correspondances[questionAppuyee][k] == reponseAppuyee) {
                bonneReponse = true;
                break;
            }
        }
        if (bonneReponse) {
            for (int i = 0; i < numPixels; i++) {
                ring.setPixelColor(i, VERT); // set pixel i to foreground
                ring.show(); // actually display it
                delay(attendre); // milliseconds
                Serial.println("correcte");
            }
        } else {
            for (int i = 0; i < numPixels; i++) {
                ring.setPixelColor(i, ROUGE); // set pixel i to foreground
                ring.show(); // actually display it
                delay(attendre); // milliseconds
                Serial.println("faux");
            }
        }
    } else {
        for (int i = 0; i < numPixels; i++) {
            ring.setPixelColor(i, ETEINT); // set pixel i to foreground
            ring.show(); // actually display it
            delay(attendre); // milliseconds
            Serial.println("eteint");
        }
    }
}
```



QUIZZ 2

