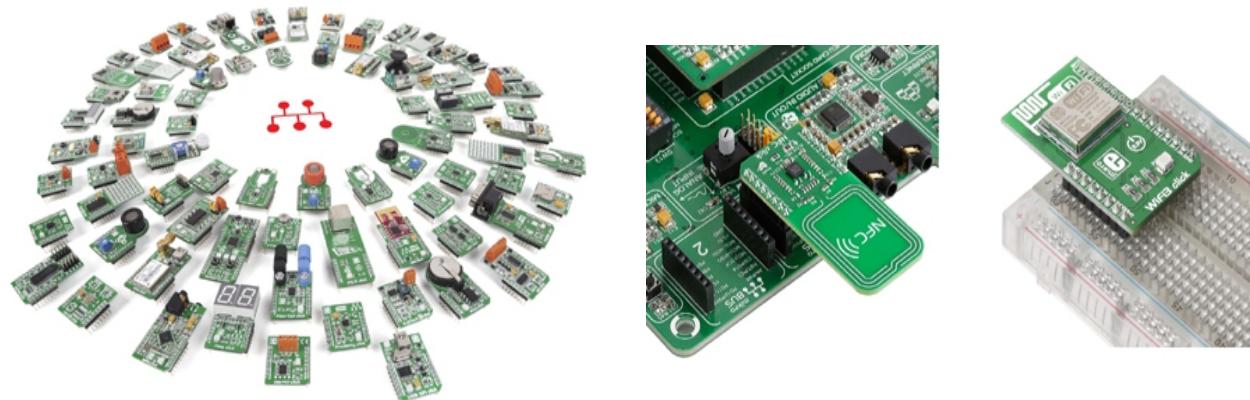


## Notes d'applications modules Click Board™ et Arduino™ (Tome 1)

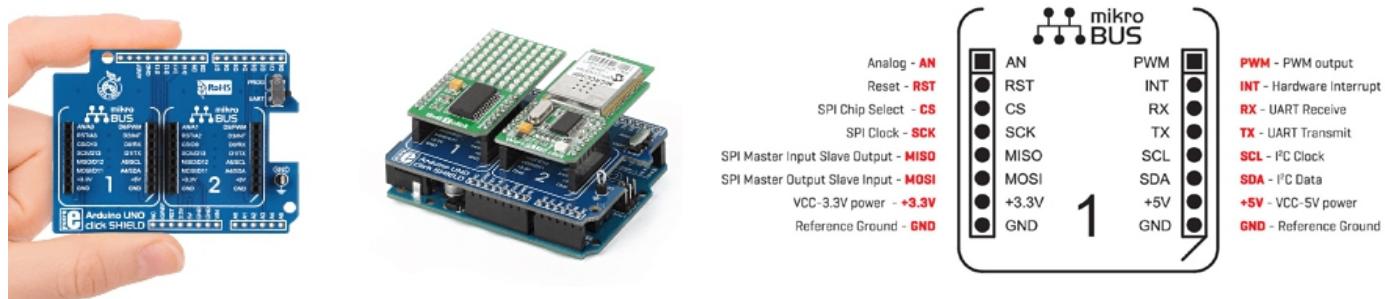
Se présentent sous la forme de petits modules OEM au format DIL, les modules Click Board™ vous permettront de mettre en oeuvre une multitude de périphériques de façon simplifiée.

Ces derniers sont idéalement conçus pour l'évaluation, l'apprentissage, le prototypage, la recherche, les études ainsi que les développements les plus divers.



Pouvant être utilisés et pilotés par la plupart des microcontrôleurs, leur conception vous permettra de pouvoir les insérer sur des plaques de développement sans soudure (type BreadBoard).

Ils sont également directement compatibles avec les platines de développement mikroElektronika (telles que l'EasyPIC7, l'EasyPIC Fusion, l'EasyAVR6 ou encore la platine Flip & Click).

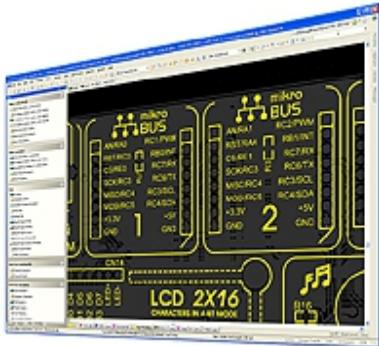


A l'aide de platines d'adaptations additionnelles, il vous sera également possible de les enficher sur des bases de développement arduino™ UNO ou Mega-2560 ou Raspberry Pi ou BeagleBone Black.

### Le standard mikroBUS™

Les modules Click Board sont dotés d'une implantation commune au standard mikroBUS™.

Cette implantation a déjà été retenue et adoptée par de très grands fabricants de circuits intégrés qui ont ajouté des supports sur leur cartes de développement afin que celles-ci puissent recevoir des modules Click Board.



Il vous est également possible d'intégrer le standard MikroBUS™ au sein de vos propres applications. Consultez le site de MikroElektronika pour plus d'informations.

#### Du code source pour les modules Click™ Board !



Disposer d'une solution matérielle pour développer c'est bien... mais disposer du code source associé pour faciliter une intégration au sein de son application... c'est mieux ! C'est ce que vous propose mikroelektronika (le fabricant des modules Click Board) par l'intermédiaire d'un site Internet dédié à cet usage. Connectez vous sur le [www.libstock.com](http://www.libstock.com) pour accélérer la mise en oeuvre des modules "Click Board".

Vous disposerez alors (suivant les modules) de code source pour les compilateurs Mikroelektronika « **BASIC** », « **C** » et « **PASCAL** » dédiés aux microcontrôleurs :

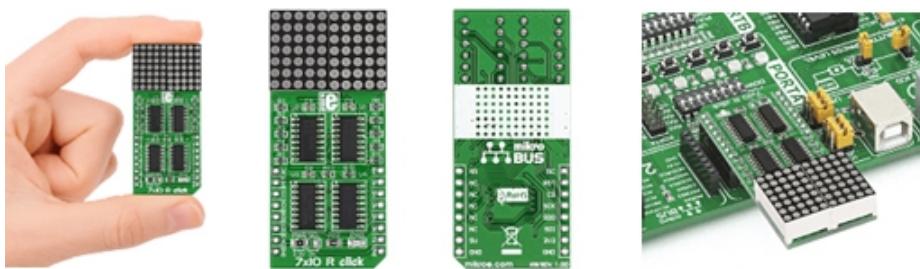
- PIC / dsPIC / PIC24 / PIC32
- AVR
- ARM
- FT90x
- 8051

Le document qui suit est quand à lui dédié à la mise en œuvre des modules Click Board™ avec un arduino™ UNO.



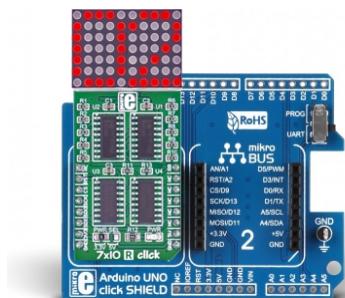
Les applications peuvent être réalisées (pour une plus grande simplicité) avec une platine d'interface MIKROE-1581 (vous permettant d'enficher directement le module Click Board sur la platine Arduino™). Nous détaillons également les connexions fils à fils, si vous ne disposez pas de cette platine.

## Application 001 : MIKROE-1993 : Module 7x10 R Click

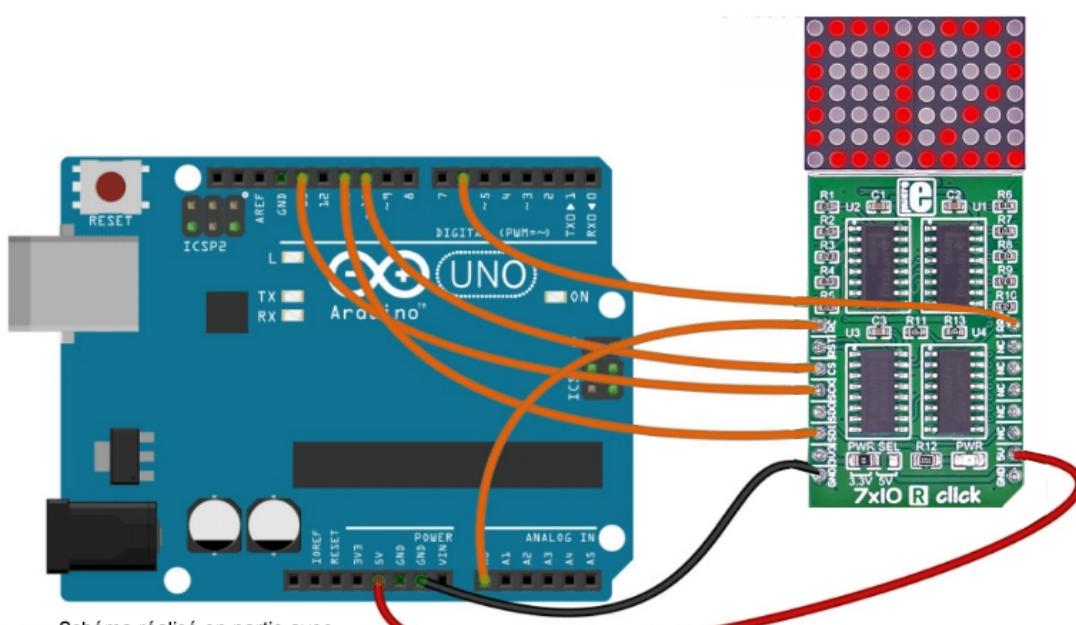


Ce module Click Board vous permettra d'ajouter une matrice de 7x10 leds rouges (soit un total de 70 leds rouges) à votre microcontrôleur via une liaison SPI.

### Montage à réaliser



Enfichez le module MIKROE-1993 sur la platine MIKROE-1581... Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1993 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



A noter que l'exemple de programme ci-dessous peut également s'appliquer aux modules:

- **MIKROE-2705** : Module 7 x 10 G Click (version avec matrice à leds vertes)
- **MIKROE-2790** : Module 7 x 10 Y Click (version avec matrice à leds jaune)
- **MIKROE-2789** : Module 7 x 10 B Click (version avec matrice à leds bleues)

### Programme Arduino

```
*****  
*  
* Test du module MIKROE-1993 "7x10 R Click"  
*  
*****  
* Les nombres de 00 à 99 défilent toutes les 300 ms sur les deux matrices à led du module.  
*  
* Matériel nécessaire  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "7x10 R Click" (MIKROE-1993) inséré sur le support N°1 du shield MIKROE-1581  
* ou divers straps mâles/femelles (réf. : PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
* Code source téléchargeable ici: https://www.lextronic.fr/~lextronic\_doc/Click\_APP.zip  
*****/
```

```
// Affectation des broches
```

```
#define SDI 11  
#define SCK 13  
#define LATCH 10  
#define R_CLK 14  
#define R_RST 6
```

```
int nombre=0;  
int unite=0;  
int dizaine=0;  
unsigned long temps;
```

```
byte chiffre[10][7]={  
{0x0E, 0x11, 0x19, 0x15, 0x13, 0x11, 0x0E}, //0  
{0x0E, 0x04, 0x04, 0x04, 0x04, 0x0C, 0x04}, //1  
{0x1F, 0x08, 0x04, 0x02, 0x01, 0x11, 0x0E}, //2  
{0x0E, 0x11, 0x01, 0x02, 0x04, 0x02, 0x1F}, //3  
{0x02, 0x02, 0x1F, 0x12, 0x0A, 0x06, 0x02}, //4  
{0x0E, 0x11, 0x01, 0x01, 0x1E, 0x10, 0x1F}, //5  
{0x0E, 0x11, 0x11, 0x1E, 0x10, 0x08, 0x06}, //6  
{0x08, 0x08, 0x08, 0x04, 0x02, 0x01, 0x1F}, //7
```

```

{0x0E, 0x11, 0x11, 0x0E, 0x11, 0x11, 0x0E}, //8
{0x0C, 0x02, 0x01, 0x0F, 0x11, 0x11, 0x0E}, //9
};

void setup()
{
// configuration des broches en sortie
pinMode(SDI,OUTPUT);
pinMode(SCK, OUTPUT);
pinMode(LATCH, OUTPUT);
pinMode(R_CLK, OUTPUT);
pinMode(R_RST, OUTPUT);
digitalWrite(R_RST,HIGH);
delayMicroseconds(5);
digitalWrite(R_RST,LOW);
}

void loop()
{
// Calcul
dizaine=nombre/10;
if ((dizaine==10)&&(unite==9))
{
nombre=0;
dizaine=0;
}
unite=nombre-10*dizaine;
nombre=nombre+1;

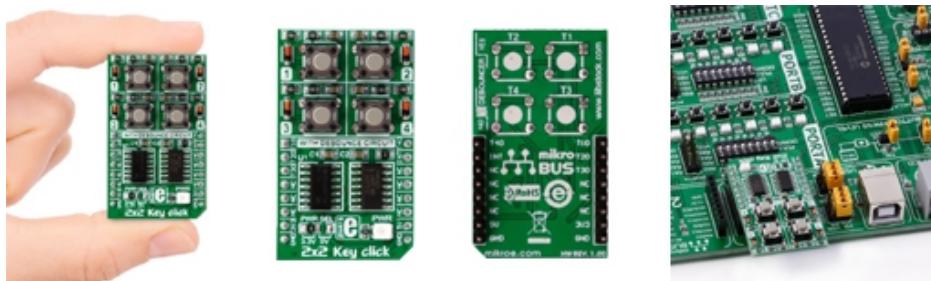
// Affichage sur les matrices
temps=millis();
do
{
for(int i=0;i<7;i++)
{
digitalWrite(LATCH, LOW);           // blocage du registre de sortie des 74HC595
shiftOut(SDI, SCK, MSBFIRST,chiffre[dizaine][i]); // envoi des 8 bits de poids forts sur le premier 74HC595
shiftOut(SDI, SCK, MSBFIRST,chiffre[unite][i]); // envoi des 8 bits de poids forts sur le second 74HC595
digitalWrite(LATCH, HIGH);          // déblocage du registre de sortie des 74HC595
delayMicroseconds(4700);           // pause pour le multiplexage à affiner en fonction de l'éclairage
digitalWrite(R_CLK,HIGH);          // envoi d'une impulsion d'horloge sur le CD4017
delayMicroseconds(5);
digitalWrite(R_CLK,LOW);
}

// Extinction de toutes les led
digitalWrite(LATCH, LOW);
shiftOut(SDI, SCK, MSBFIRST,0);

```

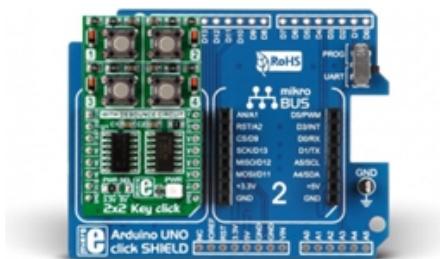
```
shiftOut(SDI, SCK, MSBFIRST,0);
digitalWrite(LATCH, HIGH);
digitalWrite(R_RST,HIGH);
delayMicroseconds(5);
digitalWrite(R_RST,LOW);
}
while(millis()-temps<300);
}
```

## Application 002 : MIKROE-2152 : Module 2x2 Key Click

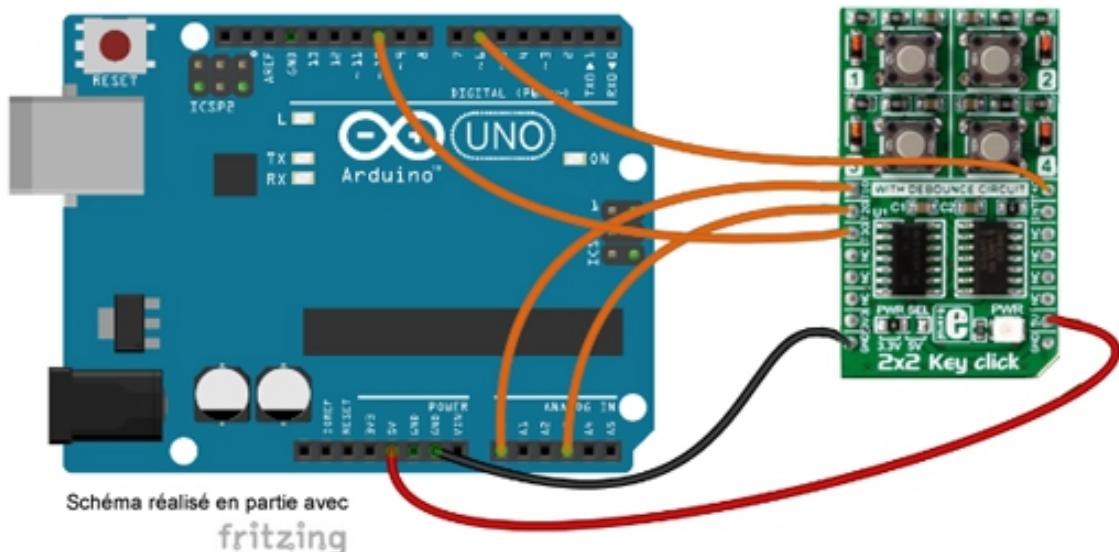


Ce module Click Board vous permettra de disposer 4 boutons poussoirs (équipés d'un système anti-rebond) sur votre microcontrôleur. Ce dernier est équipé d'un 74HC32 et d'un SN74HC14.

### Montage à réaliser



Enfichez le module MIKROE-2152 sur la platine MIKROE-1581... Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2152 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

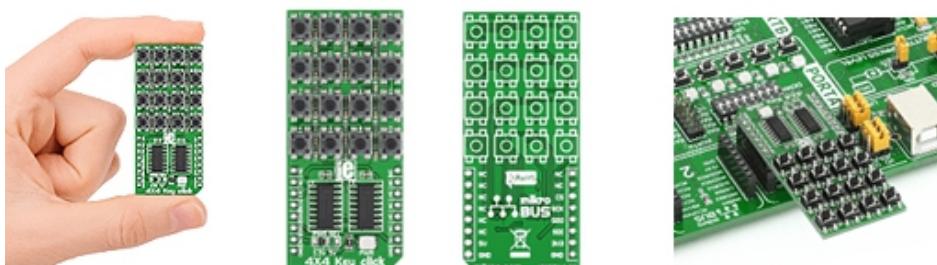


## Programme Arduino

```
*****  
*  
* Test du module "2x2 Key Click"  
*  
*****  
* Description : MIKROE-2152  
* L'état des boutons est affiché dans le moniteur série.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "2x2 Key Click" (MIKROE-2152) inséré sur le support N°1 du shield MIKROE-1581  
* ou divers straps mâles/femelles (réf. : PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
* Code source téléchargeable ici: https://www.lextronic.fr/~lextronic\_doc/Click\_APP.zip  
*  
*****/  
  
// Affectation des broches  
#define BP1 14  
#define BP2 17  
#define BP3 10  
#define BP4 6  
  
void setup()  
{  
    pinMode(BP1,INPUT); // configuration des broches en entrée  
    pinMode(BP2,INPUT);  
    pinMode(BP3,INPUT);  
    pinMode(BP4,INPUT);  
    Serial.begin(9600); // initialisation du moniteur série  
}  
  
void loop()  
{  
    if (digitalRead(BP1)==HIGH) // si BP1 est actif  
    {  
        Serial.println("Le bouton poussoir N°1 est actif");  
    }  
    else  
    {  
        Serial.println("Le bouton poussoir N°1 est inactif");  
    }  
    if (digitalRead(BP2)==HIGH) // si BP2 est actif  
    {  
        Serial.println("Le bouton poussoir N°2 est actif");  
    }  
}
```

```
    }
else
{
Serial.println("Le bouton poussoir N°2 est inactif");
}
if (digitalRead(BP3)==HIGH)           // si BP3 est actif
{
Serial.println("Le bouton poussoir N°3 est actif");
}
else
{
Serial.println("Le bouton poussoir N°3 est inactif");
}
if (digitalRead(BP4)==HIGH)           // si BP4 est actif
{
Serial.println("Le bouton poussoir N°4 est actif");
}
else
{
Serial.println("Le bouton poussoir N°4 est inactif");
}
delay(1000);                      // pause d'une seconde
}
```

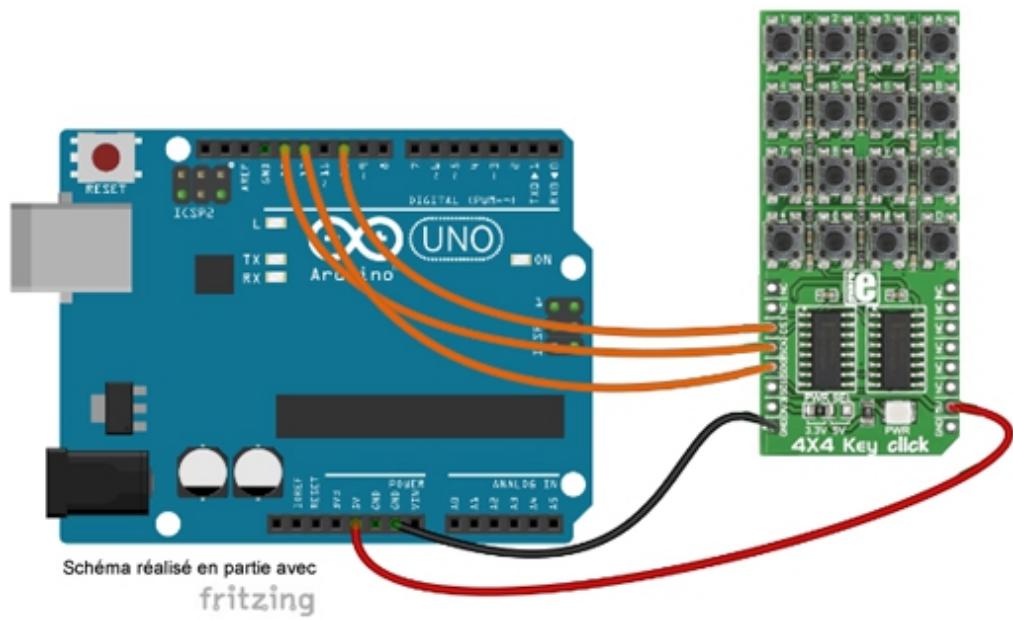
## Application 003 : MIKROE-1889 : Module 4x4 Key Click



Ce petit module "click Board" intègre un clavier composé de 16 touches (4 x 4) associées à un registre à décalage dont l'état pourra être connu par votre microcontrôleur au moyen d'une simple liaison SPI.

### Montage à réaliser

Enfichez le module MIKROE-1889 sur la platine MIKROE-1581... Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1889 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

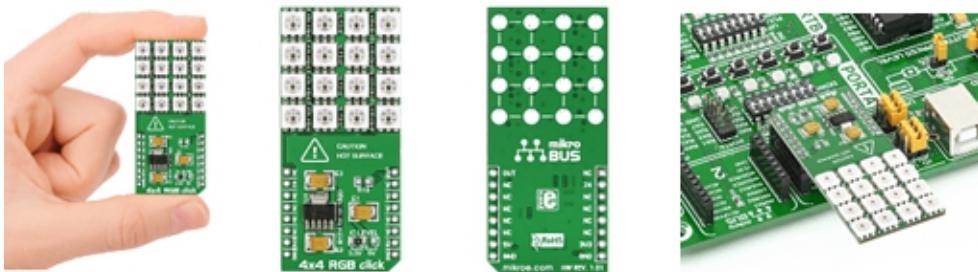


## Programme Arduino

```
*****  
*  
* Test du module "4X4 Key Click"  
*  
*****  
* L'état des boutons pousoirs est affiché dans le moniteur série.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "4x4 Key Click" (MIKROE-1889) inséré sur le support N°1 du shield MIKROE-1581  
* ou divers straps mâles/femelles (réf. : PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
* Code source téléchargeable ici: https://www.lextronic.fr/~lextronic\_doc/Click\_APP.zip  
*  
*****/  
  
// Affectation des broches  
#define LD 10  
#define CLK 13  
#define DATA 12  
  
int bouton[16];  
  
void setup()  
{  
    Serial.begin(9600); // initialisation du moniteur série  
    pinMode(LD, OUTPUT); // configuration des broches  
    pinMode(CLK, OUTPUT);  
    pinMode(DATA, INPUT);  
    digitalWrite(LD, HIGH);  
    digitalWrite(CLK, LOW);  
}  
  
void loop()  
{  
    // Chargement des entrées dans le registre 74HC165  
    digitalWrite(LD, LOW);  
    delayMicroseconds(5);  
    digitalWrite(LD, HIGH);  
    // Lecture de la donnée série sur la broche DATA et rangement dans le tableau bouton  
    for (int i=0; i <= 15; i++)  
    {  
        bouton[15-i] = digitalRead(DATA);  
    }  
}
```

```
digitalWrite(CLK, LOW);
delayMicroseconds(5);
digitalWrite(CLK, HIGH);
}
// Affichage des valeurs du tableau dans le moniteur série
for (int i=0; i <= 15; i++)
{
Serial.print("B");
Serial.print(i);
Serial.print(":");
Serial.print(bouton[i]);
Serial.print(", ");
}
delay(1000);
Serial.println();
}
```

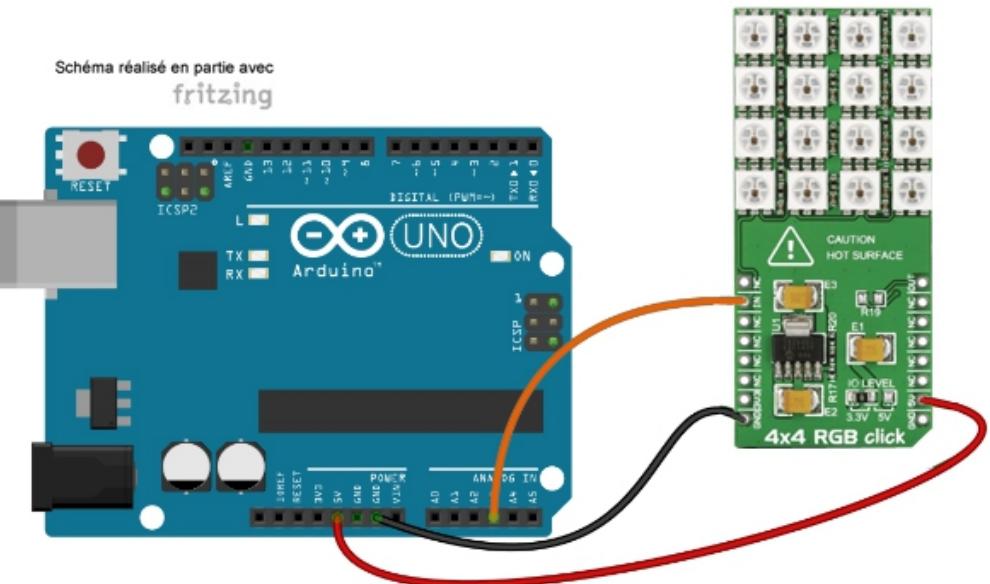
## Application 004 : MIKROE-1881 : Module 4x4 RGB Click



Ce module Click Board vous permettra d'ajouter une matrice de 4 x 4 leds RGB (avec contrôleur WS2812). Les leds peuvent être pilotées indépendamment en couleur et en intensité avec un seul port de votre microcontrôleur.

### Montage à réaliser

Enfichez le module MIKROE-1889 sur la platine MIKROE-1581... Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1889 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



**Attention, il conviendra de ne pas faire fonctionner les leds au maximum de leur intensité lumineuse afin d'éviter une consommation excessive pouvant entraîner l'endommagement de votre microcontrôleur ainsi qu'une dissipation de température trop importante.**

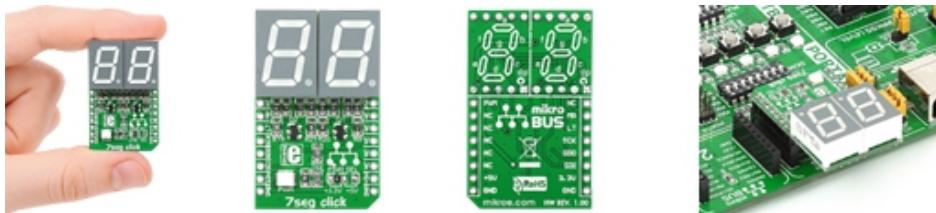
**Du fait de la haute luminosité des Leds (surtout pour les couleurs "tirant" vers le blanc), il est IMPERATIF de ne JAMAIS regarder directement la source lumineuse sous risque de graves séquelles oculaires.**

### Programme Arduino

```
*****  
*  
*   Test du module "4X4 RGB Click"  
*  
*****  
* Des effets lumineux sont générés sur les led du module.  
*   1 Arduino Uno (A000066)  
*   1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
*   1 Module "4x4 RGB Click" (MIKROE-1881) inséré sur le support N°1 du shield MIKROE-1581  
*   ou divers straps mâles/femelles (réf. : PRT-12794) si vous ne disposez pas de  
*   de la platine shield MIKROE-1581)  
*   Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*   Code source téléchargeable ici: https://www.lextronic.fr/~lextronic\_doc/Click\_APP.zip  
*   Bibliothèque à télécharger  
*   https://github.com/adafruit/Adafruit\_NeoPixel  
*****/  
  
// Affectation des broches  
#define sortie 17                                // DIN du module  
#define nb_led 16                                  // le module comporte 16 led  
  
#include <Adafruit_NeoPixel.h>                      // appel de la bibliothèque  
Adafruit_NeoPixel module = Adafruit_NeoPixel(nb_led, sortie, NEO_GRB + NEO_KHZ800); // création de l'objet  
module  
  
float j, f, k;  
  
void setup()  
{  
    module.begin();                                 // initialisation de module  
}  
  
void loop()  
{  
    for(int t = 0; t < 16; t++)                  // allumage successif des 16 led
```

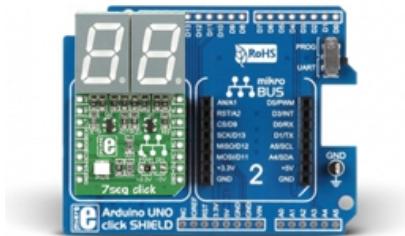
```
{  
    int rouge = 64 * (1 + sin(t / 2.0 + j / 4.0) );           // variation de la couleur rouge  
    int vert = 64 * (1 + sin(t / 1.0 + f / 9.0 + 2.1) );     // variation de la couleur verte  
    int bleu = 64 * (1 + sin(t / 3.0 + k / 14.0 + 4.2) );    // variation de la couleur bleu  
    module.setPixelColor(t, rouge, vert, bleu);                 // allumage de la led t avec les couleurs rouge, verte et bleu  
}  
module.show();                                              // rafraichissement des led  
j += random(1, 6) / 6.0;                                    // changement aléatoire des variables j, f et k  
f += random(1, 6) / 6.0;  
k += random(1, 6) / 6.0;  
}
```

## Application 005 : MIKROE-1201 : Module 7 SEG Click



Ce module Click Board vous permettra d'ajouter 2 afficheurs 7 segments rouges (digit 13 mm) à votre microcontrôleur. Ce dernier est architecturé sur la base de deux registres à décalage 74HC595 à commande SPI..

### Montage à réaliser



Enfitez le module MIKROE-1201 sur la platine MIKROE-1581...

Puis enfitez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1201 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

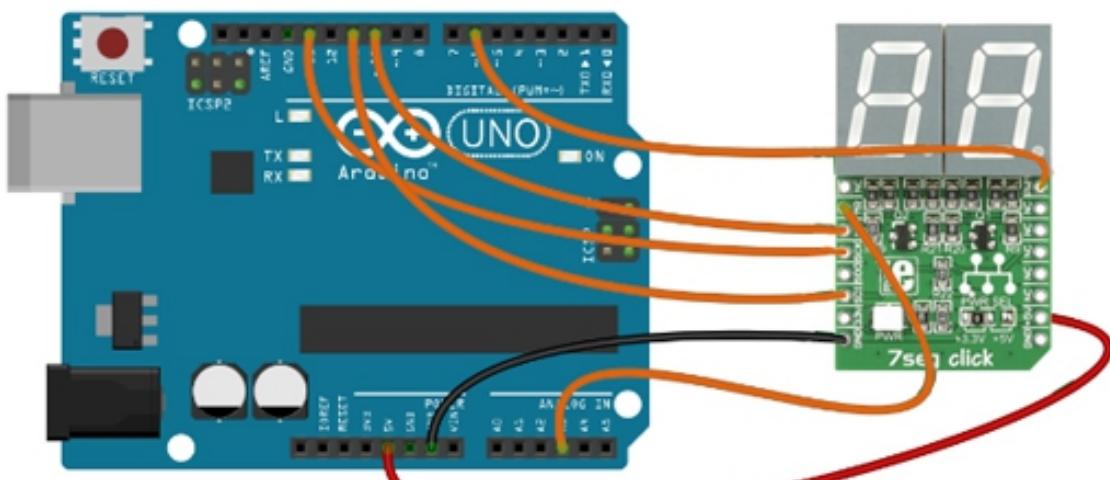


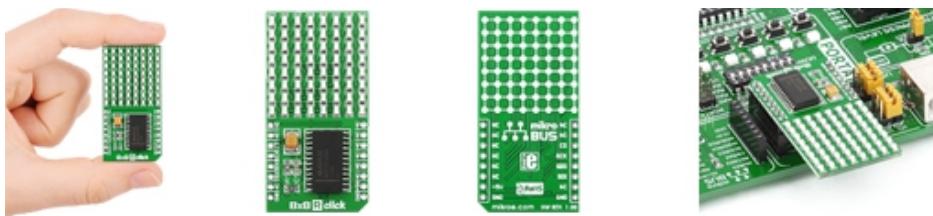
Schéma réalisé en partie avec  
fritzing

## Programme Arduino

```
*****  
*  
* Test du module "7seg Click"  
*  
*****  
* Les nombres 00 à 99 sont affichés sur les deux afficheurs du module.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "7 Seg Clcik" (MIKROE-1201) inséré sur le support N°1 du shield MIKROE-1581  
* ou divers straps mâles/femelles (réf. : PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
* Code source téléchargeable ici: https://www.lextronic.fr/~lextronic\_doc/Click\_APP.zip  
*  
*****/  
  
// Affectation des broches  
#define DSI 11          // broche DS du premier 74HC595  
#define CLK 13          // broche SHCP des 74HC595  
#define LATCH 10         // broche STCP des 74HC595  
#define MR 17            // broche MR des 74HC595  
#define PWM 6             // broche de variation de luminosité des afficheurs  
  
byte chiffre[10]={0x7E,0x0A,0xB6,0x9E,0xCA,0xDC,0xF8,0x0E,0xFE,0xCE};  
  
void setup()  
{  
pinMode(DSI, OUTPUT);  
pinMode(CLK, OUTPUT);  
pinMode(LATCH, OUTPUT);  
pinMode(MR, OUTPUT);  
pinMode(PWM, OUTPUT);  
digitalWrite(PWM,HIGH);           // validation des afficheurs  
}  
  
void loop()  
{  
digitalWrite(MR,LOW);           // remise à zero des sorties des 74HC595  
delayMicroseconds(5);  
digitalWrite(MR,HIGH);  
for (int i = 0; i < 10; i++)    // boucle de comptage des dizaines  
{  
    for (int j = 0; j < 10; j++) // boucle de comptage des unités
```

```
{  
    digitalWrite(LATCH, LOW); // blocage du registre de sortie des 74HC595  
    shiftOut(DSI, CLK, MSBFIRST, chiffre[j]); // envoi des 8 bits sur le second 74HC595 (unité)  
    shiftOut(DSI, CLK, MSBFIRST, chiffre[i]); // envoi des 8 bits sur le premier 74HC595 (dizaine)  
    digitalWrite(LATCH, HIGH); // déblocage du registre de sortie des 74HC595  
    delay(1000);  
}  
}  
}
```

## Application 006 : MIKROE-1295 : Module 8x8 R Click



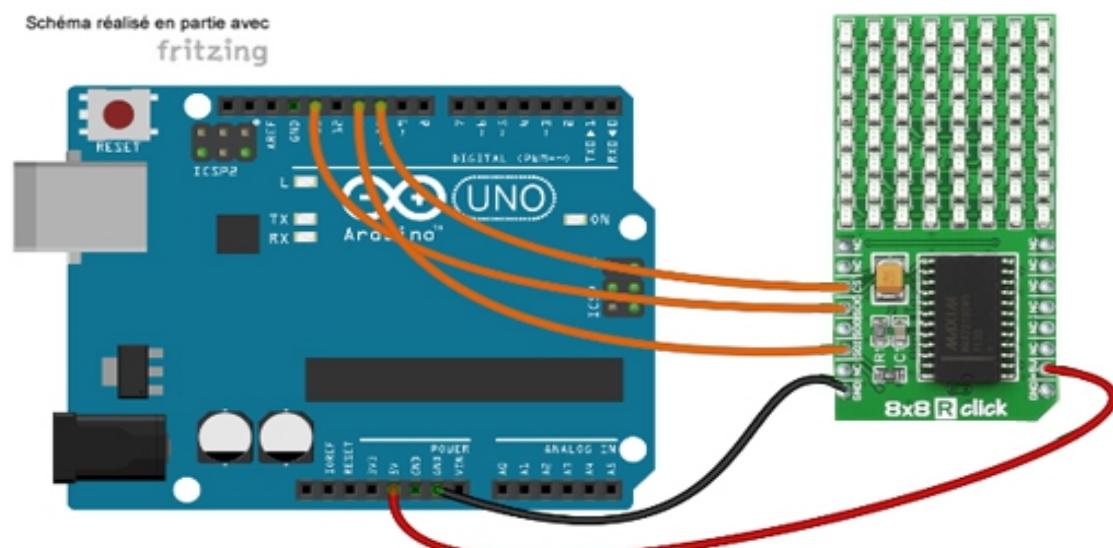
Ce module Click Board vous permettra d'ajouter une matrice de 8x8 leds CMS (soit un total de 64 leds) à votre microcontrôleur. Ce dernier est architecturé sur la base d'un MAX7219 à liaison SPI... Il existe en différentes versions (MIKROE-1295 : leds rouges – MIKROE-1294 : Leds jaunes – MIKROE-1306 : Leds vertes – MIKROE-1307 : Leds bleues)

### Montage à réaliser



Enfitez le module MIKROE-1295 sur la platine MIKROE-1581...

Puis enfitez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1295 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
* Test du module "8x8 Click"  
*****  
* Le message I ❤ Arduino est affiché sur la matrice.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "8x8 Y Click" (Réf.: MIKROE-1307) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
* Bibliothèque  
* 1. https://github.com/wayoda/LedControl  
* Logiciel permettant de convertir un caractère en 8 octets  
* 1. https://8x8ledmatrixgenerator.codeplex.com/releases/view/135669  
*****/  
  
// Affectation des broches  
#define CS 10  
#define CLK 13  
#define DIN 11  
  
#define NB_MATRICE 1  
#include "LedControl.h" // appel de la librairie  
LedControl matrice=LedControl(DIN,CLK,CS,NB_MATRICE); // création de l'objet matrice  
  
// Définition des caractères à l'aide du logiciel 8x8 ledmatrixgenerator  
unsigned char coeur[8]={0x00,0x66,0xFF,0xFF,0x7E,0x3C,0x18,0x00};  
unsigned char A[8]={0x18,0x24,0x42,0x81,0xFF,0x81,0x81,0x81};  
unsigned char R[8]={0xFC,0x82,0x81,0x82,0xFC,0x84,0x82,0x81};  
unsigned char D[8]={0xFC,0x82,0x81,0x81,0x81,0x81,0x82,0xFC};  
unsigned char U[8]={0x81,0x81,0x81,0x81,0x81,0x81,0x42,0x3C};  
unsigned char N[8]={0x81,0xC1,0xA1,0x91,0x89,0x85,0x83,0x81};  
unsigned char I[8]={0x1C,0x08,0x08,0x08,0x08,0x08,0x08,0x1C};  
unsigned char O[8]={0x3C,0x42,0x81,0x81,0x81,0x81,0x42,0x3C};  
  
void setup()  
{  
matrice.shutdown(0,false); // arrêt du mode économie du MAX7219  
matrice.setIntensity(0,8); // configuration de l'intensité lumineuse des led (0 à 15)  
matrice.clearDisplay(0); // extinction de toutes les led  
}  
  
void loop()  
{  
for(int i=0;i<8;i++) // affichage de I  
{  
matrice.setRow(0,i,I[i]);  
delay(10);  
}
```

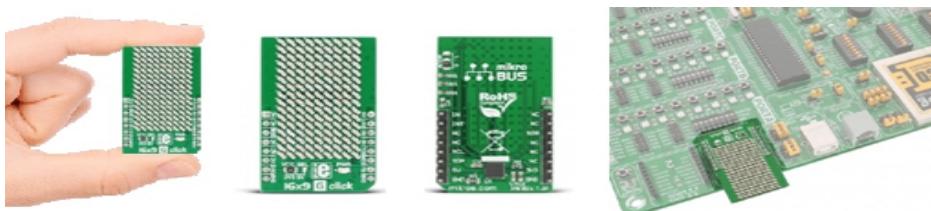
```

}

delay(1000); // pause d'une seconde
for(int i=0;i<8;i++)
{
matrice.setRow(0,i,coeur[i]); // affichage de ♥
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de A
{
matrice.setRow(0,i,A[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de R
{
matrice.setRow(0,i,R[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de D
{
matrice.setRow(0,i,D[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de U
{
matrice.setRow(0,i,U[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de I
{
matrice.setRow(0,i,I[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de N
{
matrice.setRow(0,i,N[i]);
delay(10);
}
delay(1000); // pause d'une seconde
for(int i=0;i<8;i++) // affichage de O
{
matrice.setRow(0,i,O[i]);
delay(10);
}
delay(1000); // pause d'une seconde
}

```

## Application 007 : MIKROE-2520 : Module 16x9 G Click



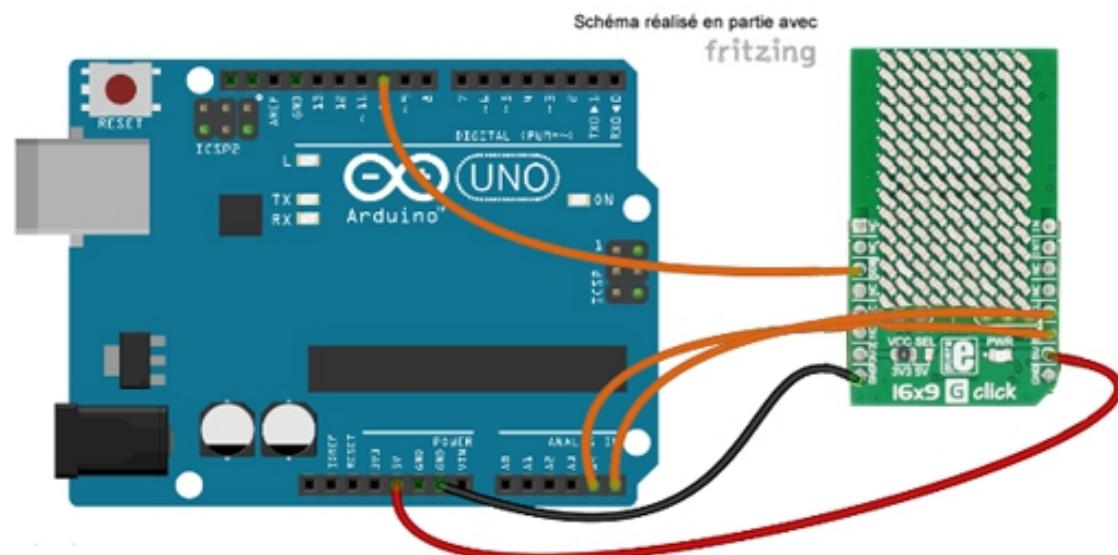
Architecté sur la base d'un IS31FL3731, ce module Click Board vous permettra d'ajouter une matrice de 16 x 9 leds CMS (soit un total de 144 leds vertes) à votre microcontrôleur via une liaison I2C.

### Montage à réaliser



Enfitez le module MIKROE-2520 sur la platine MIKROE-1581.

Puis enfitez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2520 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



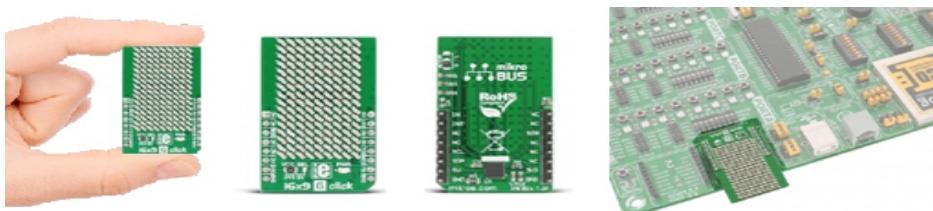
## Programme Arduino

### Programme Arduino

```
*****  
*  
* Test du module "16x9 G Click"  
*  
*****  
* Un rectangle avec effet lumineux est affiché sur la matrice.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "16x9 G Click" (Réf.: MIKROE-2520) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*  
* Bibliothèques  
* 1. https://github.com/adafruit/Adafruit\_IS31FL3731  
* 2. https://github.com/adafruit/Adafruit-GFX-Library  
*****/  
  
// Affectation des broches  
#define SDB 10 // broche Shutdown de l'IS31FL3731  
  
// Appel des bibliothèques  
#include <Wire.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_IS31FL3731.h>  
  
Adafruit_IS31FL3731 matrice = Adafruit_IS31FL3731(); // création de l'objet matrice  
  
void setup()  
{  
pinMode(SDB, OUTPUT);  
digitalWrite(SDB,HIGH); // validation de l'affichage  
matrice.begin();  
}  
  
void loop()  
{  
matrice.clear(); // extinction des led de la matrice  
// Dessin des 4 rectangles  
matrice.drawRect(0,0, matrice.width(), matrice.height(), 128);  
delay(100);  
matrice.drawRect(1,1, matrice.width()-2, matrice.height()-2, 64);  
delay(100);  
matrice.drawRect(2,2, matrice.width()-4, matrice.height()-4,32);  
delay(100);  
matrice.drawRect(3,3, matrice.width()-6, matrice.height()-6,16);  
delay(100);
```

```
matrice.drawLine(4,4,11,4,8); // dessin d'une ligne
delay(1000);
matrice.drawLine(4,4,11,4,128); // dessin de la même ligne mais avec une intensité lumineuse
différente
delay(100);
matrice.drawRect(3,3, matrice.width()-6, matrice.height()-6,64);
delay(100);
matrice.drawRect(2,2, matrice.width()-4, matrice.height()-4,32);
delay(100);
matrice.drawRect(1,1, matrice.width()-2, matrice.height()-2, 16);
delay(100);
matrice.drawRect(0,0, matrice.width(), matrice.height(), 8);
delay(1000);
}
```

## Application 008 : MIKROE-1864 : Module AlphaNum R click

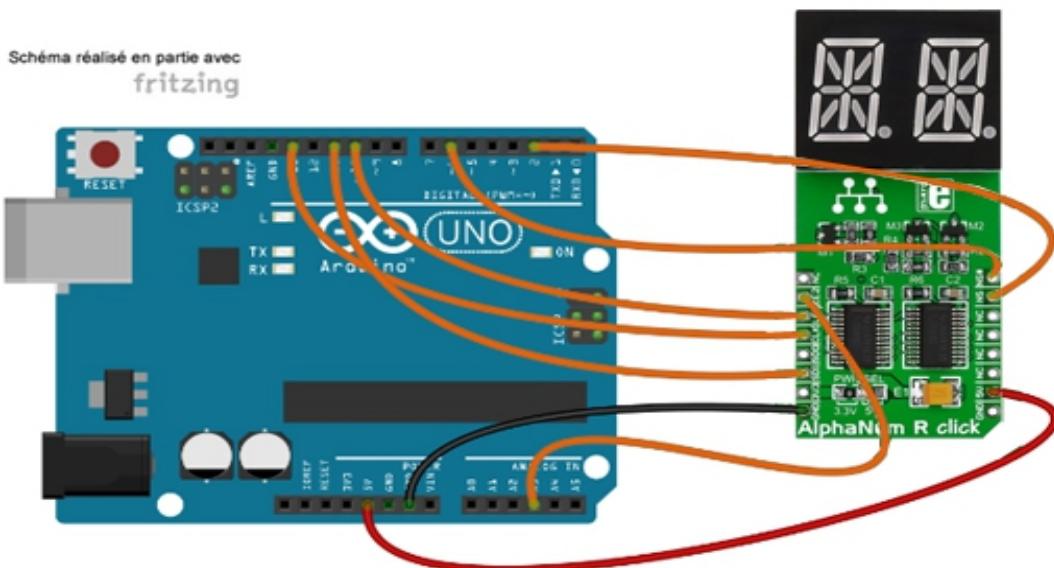


Ce module Click Board vous permettra d'ajouter 2 afficheurs alphanumériques 14 segments (digit 13 mm) à votre microcontrôleur. Il est aechitecturé sur la base de deux registres à décalage TLC5926. Le module existe avec des afficheurs de couleur rouge (MIKROE1864) ou verte (MIKROE)-1851.

### Montage à réaliser



Enfitez le module MIKROE-1864 sur la platine MIKROE-1581. Puis enfitez le tout sur la platine arduino™ (A000066) ou realisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1864 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "AlphaNum R Click"  
*  
*****  
* Deux caractères à choisir dans une table sont affichés sur les  
* deux afficheurs du module.  
  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "AlphaNum R Click" (Réf.: MIKROE-1864) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*  
*****/  
  
// Affectation des broches  
#define DIN 11  
#define CLK 13  
#define LE1 10  
#define LE2 17  
#define NUM_SEL1 2  
#define NUM_SEL2 6  
  
const unsigned Table[51]={  
  
    // Caractères  
    //-----  
    // Hex Symbole Binaire  
    //-----  
    0x0300, // '-' 0b0000001100000000  
    0x4000, // '.' 0b0100000000000000  
    0x2176, // '/' 0b0000100010000000  
    0x003F, // '0' 0b0000000000011111  
    0x0006, // '1' 0b0000000000000110  
    0x031B, // '2' 0b0000001100011011  
    0x030F, // '3' 0b0000001100001111  
    0x0000, // '4' 0b00000000001000000  
    0x032D, // '5' 0b0000001100101101  
    0x033D, // '6' 0b0000001100111101  
    0x0007, // '7' 0b0000000000000111  
    0x033F, // '8' 0b0000001100111111  
    0x032F, // '9' 0b0000001100101111  
    0x1040, // ':' 0b0001000001000000  
}
```

```

0x0840, // ';' 0b0000100001000000
0x0480, // '<' 0b0000010010000000
0x0776, // '=' 0b0000001100001000
0x2800, // '>' 0b0010100000000000
0x1223, // '?' 0b0001001000100011
0x027B, // '@' 0b0000001001111011
0x0337, // 'A' 0b0000001100110111
0x124F, // 'B' 0b0001001001001111
0x0039, // 'C' 0b0000001100111001
0x104F, // 'D' 0b0001000001001111
0x0339, // 'E' 0b0000001100111001
0x0331, // 'F' 0b0000001100110001
0x023D, // 'G' 0b0000001000111101
0x0336, // 'H' 0b0000001100110110
0x1040, // 'I' 0b0001000001000000
0x001E, // 'J' 0b00000000000011110
0x05B0, // 'K' 0b0000010110110000
0x0038, // 'L' 0b00000000000111000
0x20B6, // 'M' 0b0010000010110110
0x2436, // 'N' 0b0010010000110110
0x003F, // 'O' 0b00000000000111111
0x0333, // 'P' 0b0000001100110011
0x443F, // 'Q' 0b0100010000111111
0x0733, // 'R' 0b0000011100110011
0x032D, // 'S' 0b0000001100101101
0x1063, // 'T' 0b0001000001100011
0x003E, // 'U' 0b00000000000111110
0x08B0, // 'V' 0b0000100010110000
0x0C36, // 'W' 0b0000110000110110
0x2C80, // 'X' 0b0010110010000000
0x3080, // 'Y' 0b0011000010000000
0x0889, // 'Z' 0b0000100010001001
0x0039, // '[' 0b00000000000111001
0x0880, // '/' 0b0000100010000000
0x000F, // ']' 0b00000000000001111
0x0081, // '^' 0b0000000010000001
0x0008, // '_' 0b00000000000001000
};

void setup()
{
pinMode(DIN, OUTPUT);
pinMode(CLK, OUTPUT);
pinMode(LE1, OUTPUT);
pinMode(LE2, OUTPUT);
pinMode(NUM_SEL1, OUTPUT);
pinMode(NUM_SEL2, OUTPUT);
}

```

```

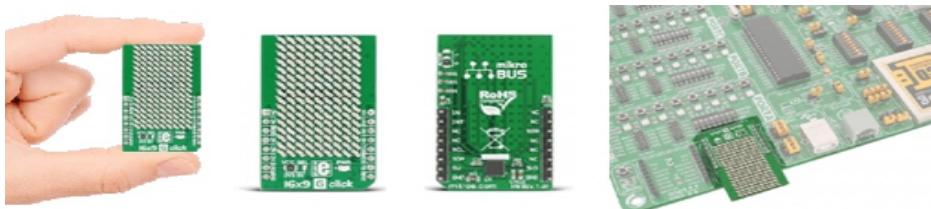
void loop()
{
Ecriture_Segment(Conversion_caractere('F'), Conversion_caractere('P'));
}

void Ecriture_Segment(int caractere1, int caractere2)
{
int MSB1;
int LSB1;
int MSB2;
int LSB2;
// Les caractères sont codées sur 2 octets. La fonction ShiftOut ne peut envoyer qu'un octet.
// Il faut donc scinder les caractères en deux octets.
MSB1=caractere1&0xFF00;
for (int i=0; i < 8; i++)
{
MSB1=MSB1>>1;
}
LSB1=caractere1&0x00FF;
MSB2=caractere2&0xFF00;
for (int i=0; i < 8; i++)
{
MSB2=MSB2>>1;
}
LSB2=caractere2&0x00FF;
digitalWrite(NUM_SEL1,HIGH); // blocage des sorties des deux registres TLC5926
digitalWrite(NUM_SEL2,HIGH);
digitalWrite(LE1, HIGH); // blocage des verrous des deux registres TLC5926
digitalWrite(LE2, HIGH);
shiftOut(DIN, CLK, MSBFIRST, MSB1); // envoi du premier caractère sur le premier TLC5926
shiftOut(DIN, CLK, MSBFIRST, LSB1);
digitalWrite(LE1, LOW); // déblocage du verrou du premier registre
digitalWrite(NUM_SEL1,LOW); // sélection du premier afficheur
delayMicroseconds(10000); // pause pour le multiplexage
digitalWrite(NUM_SEL1,HIGH);
shiftOut(DIN, CLK, MSBFIRST, MSB2); // envoi du second caractère sur le second TLC5926
shiftOut(DIN, CLK, MSBFIRST, LSB2);
digitalWrite(LE2, LOW); // déblocage du verrou du second registre
digitalWrite(NUM_SEL2,LOW); // sélection du second afficheur
delayMicroseconds(10000); // pause pour le multiplexage
digitalWrite(NUM_SEL2,HIGH);
}

unsigned Conversion_caractere(char caractere)
{
if ((caractere >= '-') && (caractere <= '_'))
return Table[caractere-'-'];
else
return 0;
}

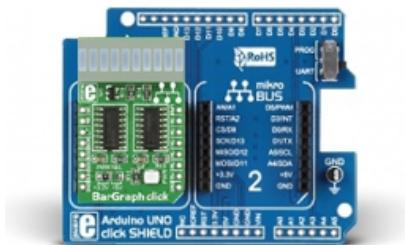
```

## Application 009 : MIKROE-1423 : Module Bargraph click

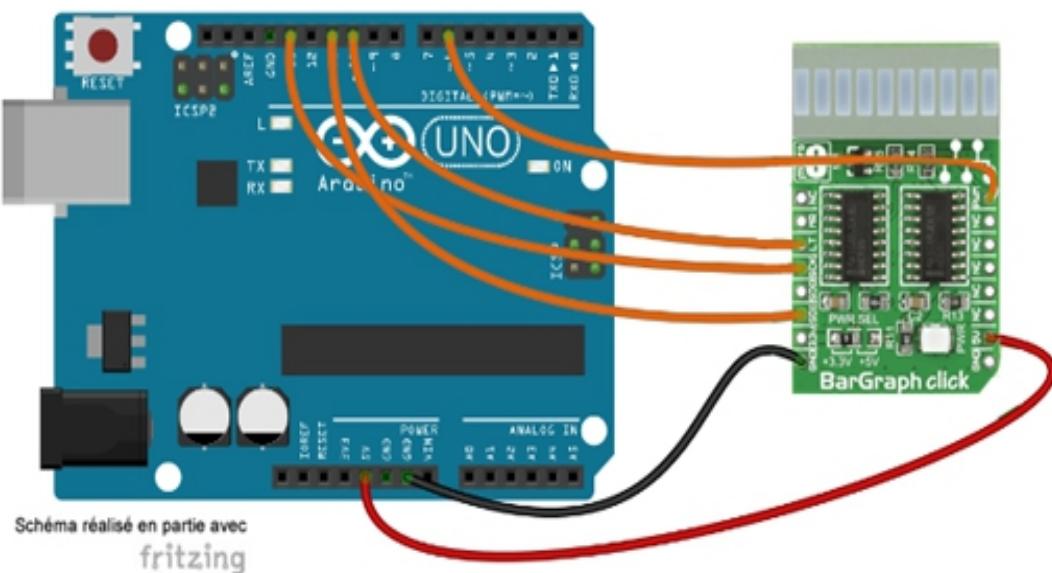


Ce module Click Board vous permettra d'ajouter un bargraph avec 10 leds rouges à votre microcontrôleur. Ce dernier est architecturé sur la base d'un JSB-R102510Z et de deux 74HC595 à liaison SPI.

### Montage à réaliser



Enfichez le module MIKROE-1423 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1423 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****
* Test du module "Bargraph Click"
*****
*Un comptage binaire sur 10 bit est affiché sur les led du bargraph.

* Matériel
* 1 Arduino Uno (A000066)
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)
* 1 Module "BARGRAPH Click" (Réf.: MIKROE-1423) inséré sur le support N°1 du shield
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de
* de la platine shield MIKROE-1423)

* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )
*****/
```

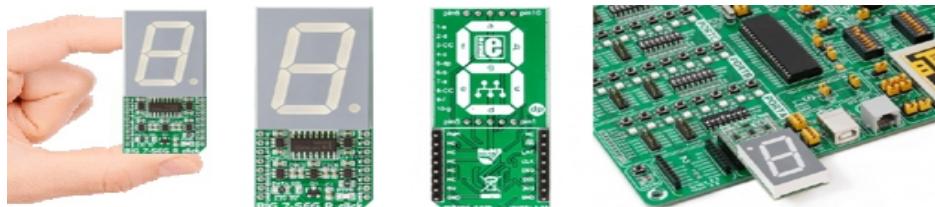
```
// Affectation des broches
#define SDIN 11 // broche DS des 74HC595
#define SCLK 13 // broche SHCP des 74HC595
#define LATCH 10 // broche STCP des 74HC595
#define DISPLAY 6 // broche de validation du bargraph

int MSB;
int LSB;

void setup()
{
pinMode(SDIN, OUTPUT);
pinMode(SCLK, OUTPUT);
pinMode(LATCH, OUTPUT);
pinMode(DISPLAY, OUTPUT);
digitalWrite(DISPLAY,HIGH); // validation du bargraph
}

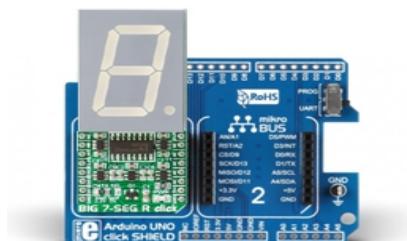
void loop()
{
for (int i = 0; i < 1024; i++) // boucle de comptage
{
LSB=i&0x00FF; // récupération des 8 bit de poids faibles de la variable i
MSB=i&0xFF00; // récupération des 8 bit de poids forts de la variable i
MSB=MSB>>8; // décalage des 8 bits de poids forts pour les mettre au format octet
digitalWrite(LATCH, LOW); // blocage du registre de sortie des 74HC595
shiftOut(SDIN, SCLK, MSBFIRST, MSB); // envoi des 8 bits de poids forts sur le second 74HC595
shiftOut(SDIN, SCLK, MSBFIRST, LSB); // envoi des 8 bits de poids faibles sur le premier 74HC595
digitalWrite(LATCH, HIGH); // déblocage du registre de sortie des 74HC595
i=i+1; // incrémentation de la variable de comptage
delay(200); // pause
}
}
```

## Application 010 : MIKROE-2269 : Module BIG 7-Seg click



Ce module Click Board vous permettra d'ajouter 1 afficheur 7 segments rouge (digit 25 mm) à votre microcontrôleur. Ce dernier est architecturé sur la base de deux registres à décalage 74HC595 à commande SPI.

### Montage à réaliser



Enfitez le module MIKROE-2269 sur la platine MIKROE-1581. Puis enfitez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2269 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

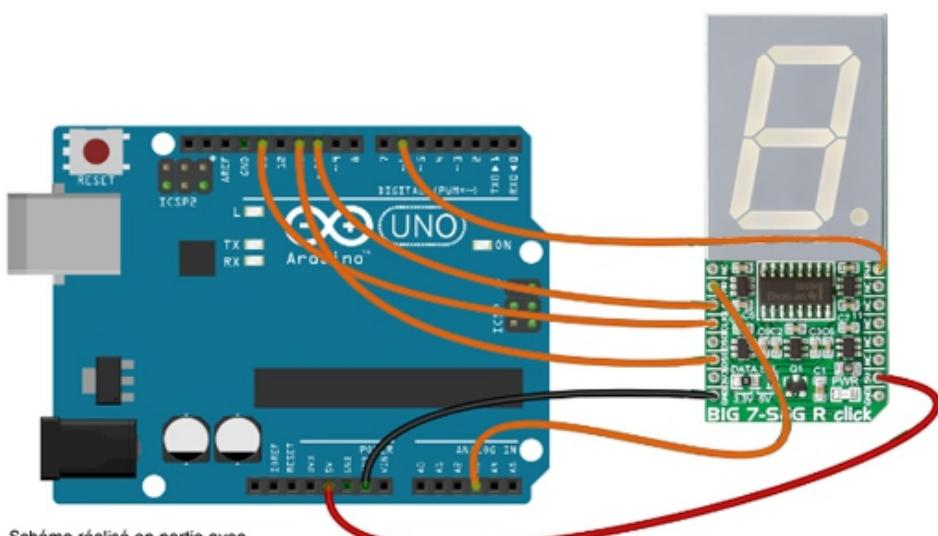


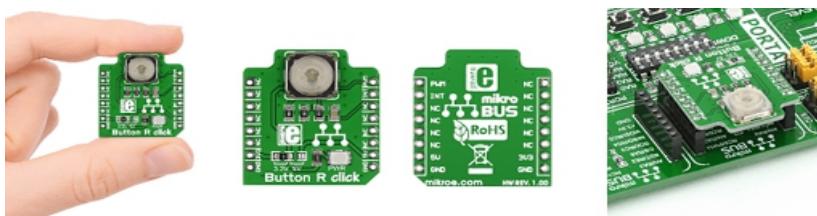
Schéma réalisé en partie avec  
fritzing

## Programme Arduino

```
*****  
* Test du module "BIG 7-SEG R Click"  
*****  
* Un comptage 0 à 9 est affiché sur l'afficheur..  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "BIG 7-SEG R Click" (Réf.: MIKROE-2269) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*****/
```

```
// Affectation des broches  
#define DSI 11 // broche DS du 74HC595  
#define CLK 13 // broche SHCP du 74HC595  
#define LATCH 10 // broche STCP du 74HC595  
#define MR 17 // broche MR du 74HC595  
#define PWM 6 // broche de variation de luminosité de l'afficheur  
  
byte chiffre[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7C,0x07,0x7F,0x67};  
  
void setup()  
{  
pinMode(DSI, OUTPUT);  
pinMode(CLK, OUTPUT);  
pinMode(LATCH, OUTPUT);  
pinMode(MR, OUTPUT);  
pinMode(PWM, OUTPUT);  
digitalWrite(PWM,HIGH); // validation de l'afficheur  
}  
  
void loop()  
{  
  
digitalWrite(MR,LOW); // remise à zero des sorties du 74HC595  
delayMicroseconds(5);  
digitalWrite(MR,HIGH);  
for(int i=0;i<10;i++)  
{  
digitalWrite(LATCH, LOW); // blocage du registre de sortie du 74HC595  
shiftOut(DSI, CLK, MSBFIRST, chiffre[i]); // envoi des 8 bits sur le 74HC595  
digitalWrite(LATCH, HIGH); // déblocage du registre de sortie du 74HC595  
delay(1000);  
}  
}
```

## Application 011 : MIKROE-1901 : Module Button R click

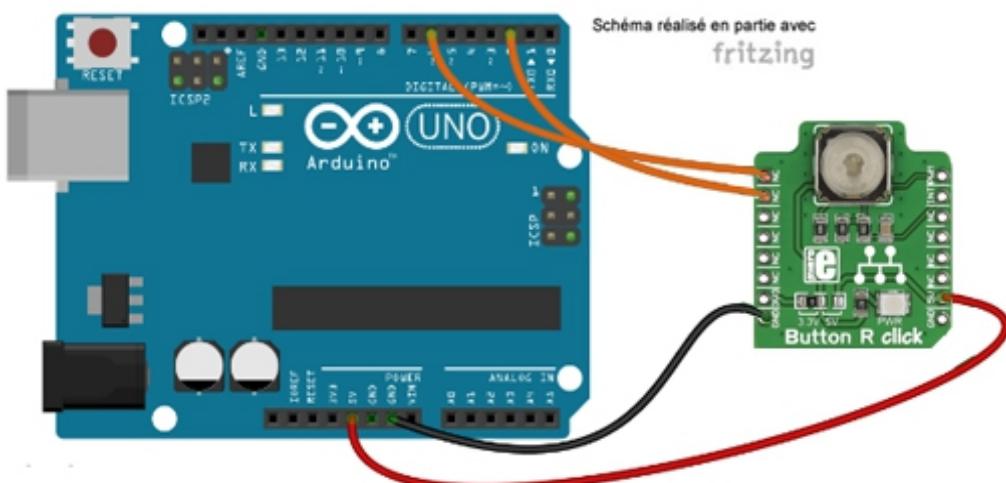


Ce petit module "click Board" intègre un bouton-poussoir transparent avec une led rouge présente dans le corps de ce dernier. Le module existe dans une version avec une led verte (MIKROE-2040).

### Montage à réaliser



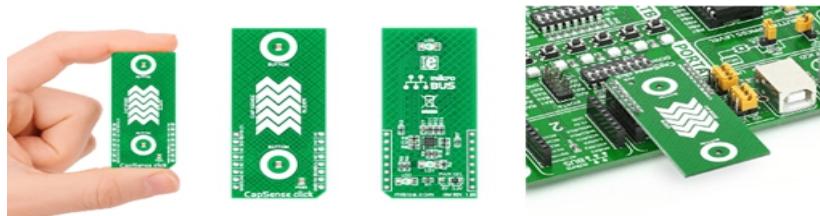
Enfichez le module MIKROE-1901 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1901 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
* Test du module "Button R Click"  
*****  
* L'état du bouton est affiché dans le moniteur série et la led clignote  
* pendant 1 s après chaque appui.  
  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Button R Click" (Réf.: MIKROE-1901) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*  
*****/  
  
// Affectation des broches  
#define LED 6  
#define BP 2  
  
void setup()  
{  
    pinMode(BP,INPUT); // configuration de la broche en entrée  
    pinMode(LED,OUTPUT); // configuration de la broche en sortie  
    Serial.begin(9600); // initialisation du moniteur série  
}  
  
void loop()  
{  
    if (digitalRead(BP)==HIGH) // si BP est actif  
    {  
        Serial.println("Le bouton poussoir est actif");  
        for (int i=0;i<10;i++)  
        {  
            digitalWrite(LED,HIGH); // allumage de la led  
            delay(50);  
            digitalWrite(LED,LOW); // extinction de la led  
            delay(50);  
        }  
    }  
    else  
    {  
        Serial.println("Le bouton poussoir est inactif");  
    }  
}
```

Application 012 : MIKROE-1446 : Module CapSense click

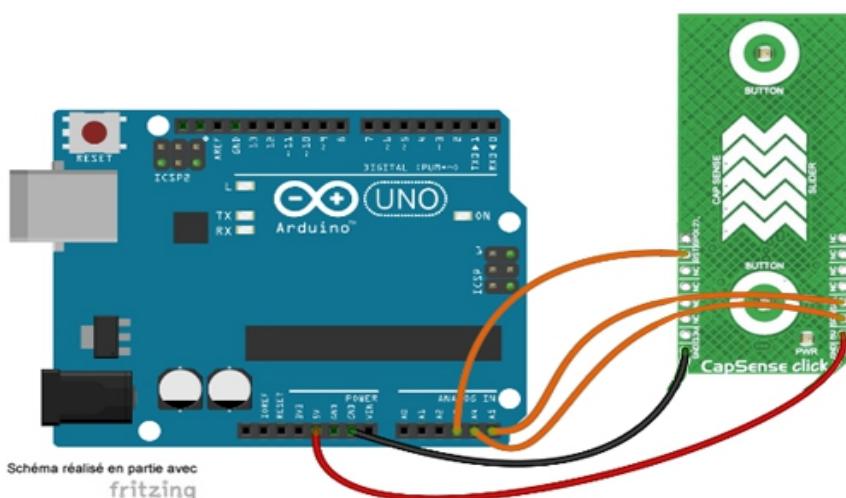


Ce petit module "click Board" intègre un capteur capacitif de type "CY8C201A0". Ce dernier dispose d'une liaison I2C qui vous permettra de le piloter depuis votre microcontrôleur.

## Montage à réaliser



Enfichez le module MIKROE-1901 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1901 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "CapSense Click"  
*  
*****  
* L'état des deux boutons et l'évolution du slider sont affichés  
* dans le moniteur série.  
  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "CapSense Click" (Réf.: MIKROE-1446) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
  
* Schéma publié sous licence CC Attribution-ShareALike (réalisé en partie avec Fritzing )  
*  
*****/  
  
#include <Wire.h> // appel de la bibliothèque  
#define CY8C201A0_Adresse 0x00 // adresse I2C du CY8C201A0  
  
//Affectation des broches  
#define XRES 17  
  
byte bouton;  
byte slider1;  
byte slider2;  
byte slider;  
  
void setup()  
{  
pinMode(XRES,OUTPUT);  
digitalWrite(XRES,LOW);  
Serial.begin(9600); // initialisation de la liaison série  
Wire.begin(); // initialisation de la liaison I2C  
Init_CY8C201A0(); // initialisation du CY8C201A0  
  
// Ecriture dans le registre OUTPUT_PORT0 pour éteindre les led  
Ecriture_Registre(0x04,0x03);  
delay(500);
```

```

// Ecriture dans le registre OUTPUT_PORT0 pour allumer les led
Ecriture_Registre(0x04,0x00);
delay(500);
}

void loop()
{
bouton=Lecture_Registre(0x88);
Serial.print("Bouton=");
Serial.println(bouton);

Wire.beginTransmission(CY8C201A0_Adresse); // envoi de l'adresse du CY8C201A0
Wire.write(0x8A); // envoi de l'adresse du registre
Wire.endTransmission();
Wire.beginTransmission(CY8C201A0_Adresse|0x1); // envoi de l'adresse du CY8C201A0+R
Wire.requestFrom(CY8C201A0_Adresse, 2);
delay(10);
if (Wire.available()<=2) // récupération des deux octets
{
slider1 = Wire.read();
slider2 = Wire.read();
}
Wire.endTransmission();
Serial.print("Slider1=");
Serial.println(slider1);
Serial.print("Slider2=");
Serial.println(slider2);

delay(500);
}

void Init_CY8C201A0(void) // Initialisation du CY8C201A0
{
// Ecriture dans le registre COMMAND_REG pour accéder au mode de configuration
Ecriture_Registre(0xA0,0x08);
// Ecriture dans le registre CS_ENABL1 pour définir les broches du slider
Ecriture_Registre(0x07,0x1F);
// Ecriture dans le registre CS_ENABLO pour définir les broches des boutons
Ecriture_Registre(0x06,0x18);
// Ecriture dans le registre GPIO_ENABLE0 pour définir les broches des led
Ecriture_Registre(0x08,0x03);
// Ecriture dans le registre DM_STRONG0 pour valider le mode "Strong Drive"
Ecriture_Registre(0x11,0x03);
}

```

```

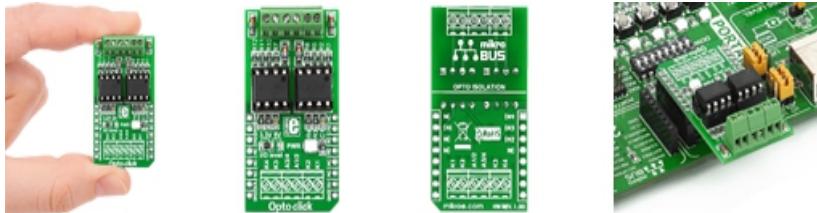
// Ecriture dans le registre CS_SLID_CONFIG pour configurer le slider
Ecriture_Registre(0x75,0x01);
// Ecriture dans les registres CS_SLID_MULM et CS_SLID_MULL pour configurer la résolution du slider
Ecriture_Registre(0x77,0x30);
Ecriture_Registre(0x78,0x00);
// Ecriture dans le registre COMMAND_REG pour mémoriser cette configuration
Ecriture_Registre(0xA0,0x01);
delay(250);
Ecriture_Registre(0xA0,0x06);
delay(250);
}

void Ecriture_Registre(byte adresse,byte donnee)
{
Wire.beginTransmission(CY8C201A0_Adresse); // envoi de l'adresse du CY8C201A0
Wire.write(adresse); // envoi de l'adresse du registre
Wire.write(donnee); // envoi de la donnée
Wire.endTransmission();
}

byte Lecture_Registre(byte adresse)
{
byte donnee;
Wire.beginTransmission(CY8C201A0_Adresse); // envoi de l'adresse du CY8C201A0
Wire.write(adresse); // envoi de l'adresse du registre
Wire.endTransmission();
Wire.beginTransmission(CY8C201A0_Adresse|0x1); // envoi de l'adresse du CY8C201A0+R
Wire.requestFrom(CY8C201A0_Adresse, 1);
delay(10);
if (Wire.available()<=1) // récupération de la donnée
{
donnee = Wire.read();
}
Wire.endTransmission();
return donnee;
}

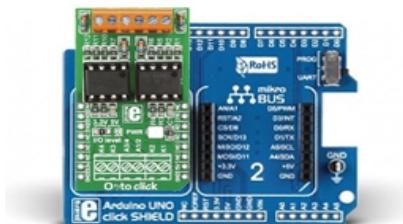
```

## Application 013 : MIKROE-1196 : Module Opto click Board

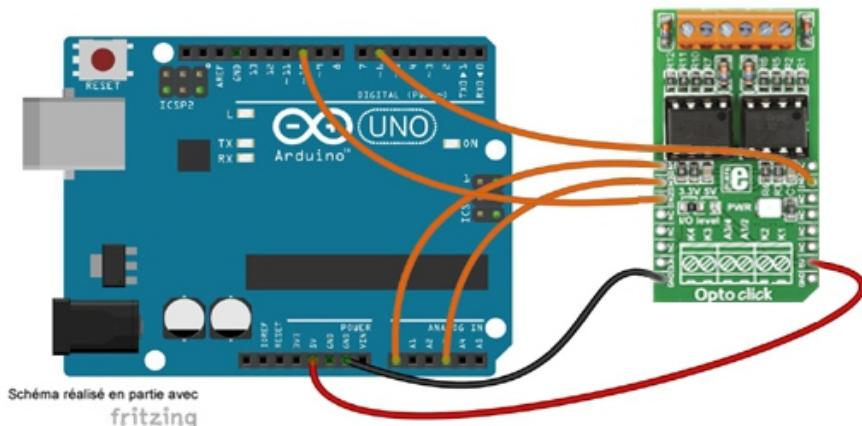


Ce module Click Board vous permettra d'ajouter 2 optocoupleurs haute vitesse à votre microcontrôleur. Ce dernier est architecturé sur la base de 2 circuits intégrés VO2630.

### Montage à réaliser



Enfichez le module MIKROE-1196 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1196 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "OPTO Click Board"  
*  
*****  
* L'état des sorties du module est affiché dans le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "OPTO Click Board" (Réf.: MIKROE-1196) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define IN1 2  
#define IN2 10  
#define IN3 17  
#define IN4 14  
  
void setup()  
{  
pinMode(IN1,INPUT); // configuration des broches en entrée  
pinMode(IN2,INPUT);  
pinMode(IN3,INPUT);  
pinMode(IN4,INPUT);  
Serial.begin(9600); // initialisation du moniteur série  
}  
  
void loop()  
{  
if (digitalRead(IN1)==HIGH) // si IN1 est inactif  
{  
Serial.println("Entree 1 inactive");  
}  
else  
{  
Serial.println("Entree 1 active");  
}
```

```
if (digitalRead(IN2)==HIGH) // si IN2 est inactif
{
Serial.println("Entree 2 inactive");
}
else
{
Serial.println("Entree 2 active");
}
if (digitalRead(IN3)==HIGH) // si IN3 est inactif
{
Serial.println("Entree 3 inactive");
}
else
{
Serial.println("Entree 3 active");
}
if (digitalRead(IN4)==HIGH) // si IN4 est inactif
{
Serial.println("Entree 4 inactive");
}
else
{
Serial.println("Entree 4 active");
}
delay(1000); // pause d'une seconde
}
```

Application 014 : MIKROE-1650 : Module Oled B click

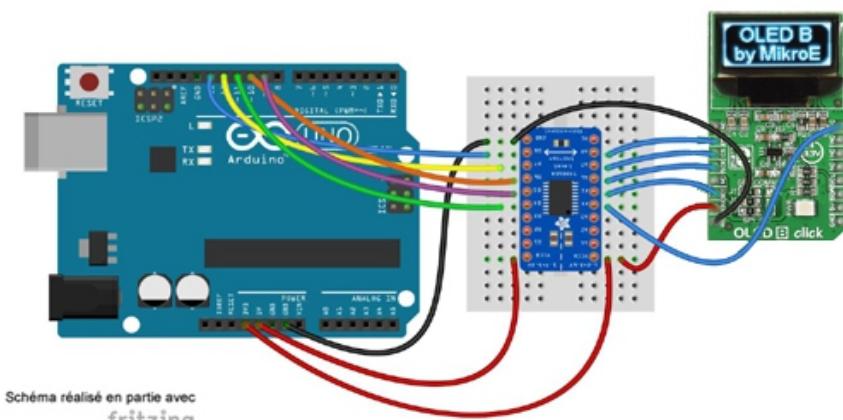


Ce module Click Board vous permettra d'ajouter un afficheur graphique OLED "bleu monochrome" d'une résolution de 96 x 39 pixels à votre microcontrôleur via une liaison SPI™ ou I2C™.

## Montage à réaliser



Enfichez le module MIKROE-1650 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1650 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "OLED B Click"  
*  
*****  
* Un nuage de points est affiché aléatoirement puis le message LEXTRONIC  
* apparaît et clignote 3 fois  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "OLED B Click" (Réf.: MIKROE-1650) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque  
* 1. https://github.com/adafruit/Adafruit\_SSD1306  
* 2. https://github.com/adafruit/Adafruit-GFX-Library  
*  
*****/  
  
// Affectation des broches  
#define OLED_MOSI 9  
#define OLED_CLK 10  
#define OLED_DC 11  
#define OLED_CS 12  
#define OLED_RESET 13  
  
// Appel des bibliothèques  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>  
#include <SPI.h>  
  
Adafruit_SSD1306 afficheur(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS); //création de l'objet  
  
int x;  
int y;  
  
void setup(void)
```

```

{
afficheur.begin(); // initialisation de l'objet afficheur
afficheur.display(); // rafraîchissement de l'écran
afficheur.clearDisplay(); // effacement de l'écran
afficheur.display(); // rafraîchissement de l'écran
}

void loop()
{
afficheur.clearDisplay(); // effacement de l'écran
afficheur.display(); // rafraîchissement de l'écran
for (int i=0; i <= 50; i++) // apparition du nuage de points
{
x=random(128); // x prend une valeur aléatoire comprise entre 0 et 128
y=random(32); // y prend une valeur aléatoire comprise entre 0 et 32
afficheur.drawPixel(x, y, WHITE); // affichage d'un pixel en (x,y)
afficheur.display(); // rafraîchissement de l'écran
delay(50); // pause de 50 ms
}
afficheur.setTextSize(1); // configuration de la taille des caractères
afficheur.setTextColor(WHITE);
afficheur.setCursor(40,20); // placement du curseur en x=40 et y=20
afficheur.println("LEXTRONIC"); // écriture de LEXTRONIC
afficheur.display(); // rafraîchissement de l'écran
delay(1000);
for (int i=0; i <= 3; i++) // clignotement du message
{
afficheur.setCursor(40,20);
afficheur.println("LEXTRONIC");
afficheur.display();
delay(1000);
afficheur.clearDisplay();
afficheur.display();
delay(500);
}
}// Affectation des broches
#define OLED_MOSI 9
#define OLED_CLK 10
#define OLED_DC 11
#define OLED_CS 12
#define OLED_RESET 13

```

```

// Appel des bibliothèques
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SPI.h>

Adafruit_SSD1306 afficheur(OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS); //création de l'objet

int x;
int y;

void setup(void)
{
afficheur.begin(); // initialisation de l'objet afficheur
afficheur.display(); // rafraîchissement de l'écran
afficheur.clearDisplay(); // effacement de l'écran
afficheur.display(); // rafraîchissement de l'écran
}

void loop()
{
afficheur.clearDisplay(); // effacement de l'écran
afficheur.display(); // rafraîchissement de l'écran
for (int i=0; i <= 50; i++) // apparition du nuage de points
{
x=random(128); // x prend une valeur aléatoire comprise entre 0 et 128
y=random(32); // y prend une valeur aléatoire comprise entre 0 et 32
afficheur.drawPixel(x, y, WHITE); // affichage d'un pixel en (x,y)
afficheur.display(); // rafraîchissement de l'écran
delay(50); // pause de 50 ms
}
afficheur.setTextSize(1); // configuration de la taille des caractères
afficheur.setTextColor(WHITE);
afficheur.setCursor(40,20); // placement du curseur en x=40 et y=20
afficheur.println("LEXTRONIC"); // écriture de LEXTRONIC
afficheur.display(); // rafraîchissement de l'écran
delay(1000);
for (int i=0; i <= 3; i++) // clignotement du message
{
afficheur.setCursor(40,20);
afficheur.println("LEXTRONIC");
}

```

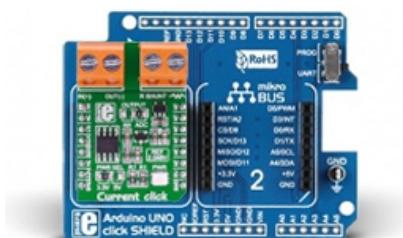
```
afficheur.display();
delay(1000);
afficheur.clearDisplay();
afficheur.display();
delay(500);
}
}
```

## Application 015 : MIKROE- 1396 : Module Current click Board

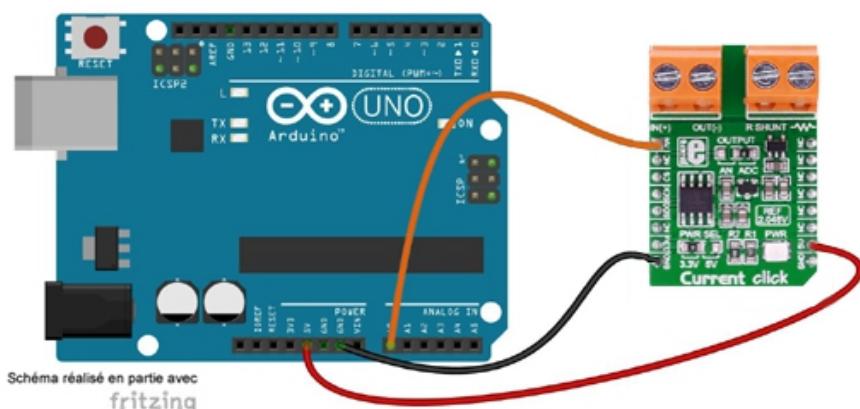


Ce module vous permettra d'effectuer des mesures de courant avec votre microcontrôleur. Ce dernier est architecturé sur la base d'un INA196, couplé à un MAX6106 et à un convertisseur ADC MCP3201.

### Montage à réaliser



Enfichez le module MIKROE-1396 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1396 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

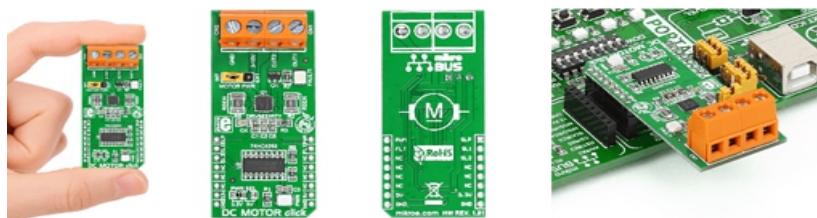


## Programme Arduino

```
*****  
*  
* Test du module "Current Click"  
*  
*****  
* La valeur du courant (entre 20 et 100 mA) circulant à travers la résistance shunt du module  
* est affichée sur le moniteur série  
* Le cavalier J2 est en position AN et la résistance shunt est égale à 1Ω  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Current Click" (Réf.: MIKROE-1396) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Câblage  
* Le courant arrive sur la broche VIN+ du bornier IN et ressort par la  
* broche VIN- du bornier IN sur laquelle est également branchée une  
* résistance de 50Ω reliée à la masse.  
* La résistance shunt est reliée aux deux broches du bornier R SHUNT.  
*  
*****/  
  
// Affectation des broches  
#define AN A0 // broche de sortie du module  
int valeur;  
float courant;  
void setup()  
{  
Serial.begin(9600); // initialisation de la liaison série  
}  
  
void loop()  
{  
valeur=analogRead(AN); // conversion AN  
Serial.print("Valeur=");
```

```
Serial.println(valeur);
courant=0.176*valeur; // calcul du courant
Serial.print("Courant=");
Serial.print(courant);
Serial.println(" mA");
delay(1000);
}
```

## Application 016 : MIKROE-1526 : Module Motor Driver click

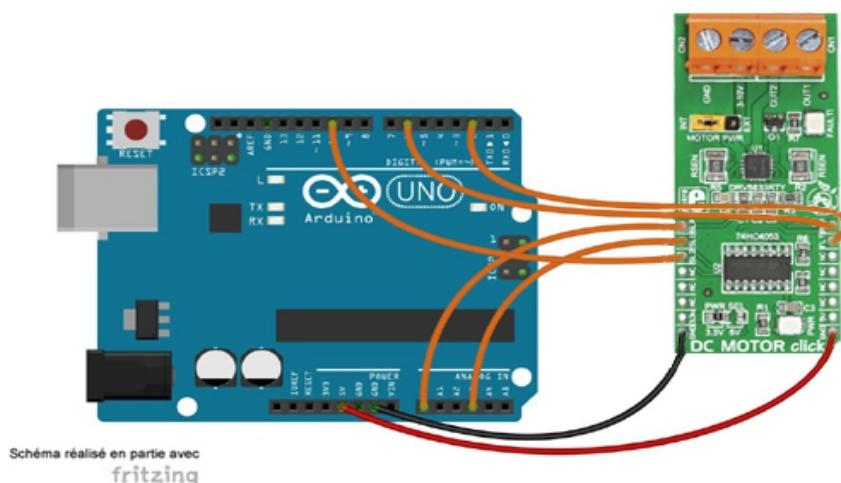


Ce petit module "click Board" intègre un multiplexeur "74HC4053" associé à un contrôleur DRV8833RTY permettant le pilotage d'un moteur "cc" de 3 à 10 V / 0,9 A max.

### Montage à réaliser



Enfichez le module MIKROE-1526 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1526 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "DC Motor Click"  
*  
*****  
* La commande du moteur est réalisée à l'aide du moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "DC Motor Click" (Réf.: MIKROE-1526) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define nSLEEP 14  
#define SELECT1 17  
#define SELECT2 10  
#define PWM 6  
#define nFAULT 2  
  
char sens=0;  
char recu[3];  
boolean dir;  
int vitesse=0;  
int vitesse_moteur;  
  
void setup()  
{  
Serial.begin(9600); // initialisation du moniteur série  
pinMode(nSLEEP,OUTPUT); // configuration des broches  
pinMode(SELECT1,OUTPUT);  
pinMode(SELECT2,OUTPUT);  
pinMode(PWM,OUTPUT);  
pinMode(nFAULT,INPUT);  
digitalWrite(SELECT1,LOW);  
digitalWrite(nSLEEP,HIGH); // validation du DRV8833RTY
```

```

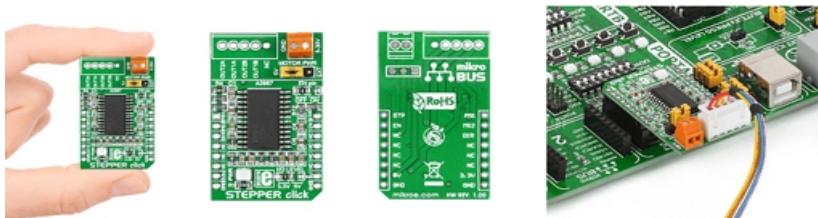
attachInterrupt(0, defaut, FALLING); // autorisation des interruptions sur la broche N°2
}

void loop()
{
Serial.println("Rentrer le sens (A: avance ou R:recule) puis Envoyer");
while (Serial.available()==0); // attente du paramètre sens
sens=Serial.read(); // réception du paramètre sens
Serial.println("Rentrer la vitesse (000..100) puis Envoyer");
while (Serial.available()==0); // attente du paramètre vitesse
for (int i=0;i<3;i++) // réception du paramètre vitesse
{
recu[i]=Serial.read();
delay(10);
}
vitesse=(recu[0]-48)*100+(recu[1]-48)*10+recu[2]-48; // reconstitution de la variable vitesse
vitesse_moteur=map(vitesse,0,100,0,255); // changement d'échelle pour la variable vitesse du moteur (0 à 255)
Serial.print("Le moteur tourne dans le sens ");
if (sens==65) // code ASCII de A
{
dir=LOW;
Serial.print("horaire");
}
if (sens==82) // code ASCII de R
{
dir=HIGH;
Serial.print("anti-horaire");
}
Serial.print(" avec une vitesse de ");
Serial.print(vitesse);
Serial.println(" %");
Serial.println(" ");
digitalWrite(SELECT2,dir); // commande du moteur
analogWrite(PWM,vitesse_moteur);
delay(100);
}

void defaut()
{
digitalWrite(nSLEEP,LOW); // blocage du DRV8833RTY
}

```

Application 017 : MIKROE-1528 : Module Stepper click Board

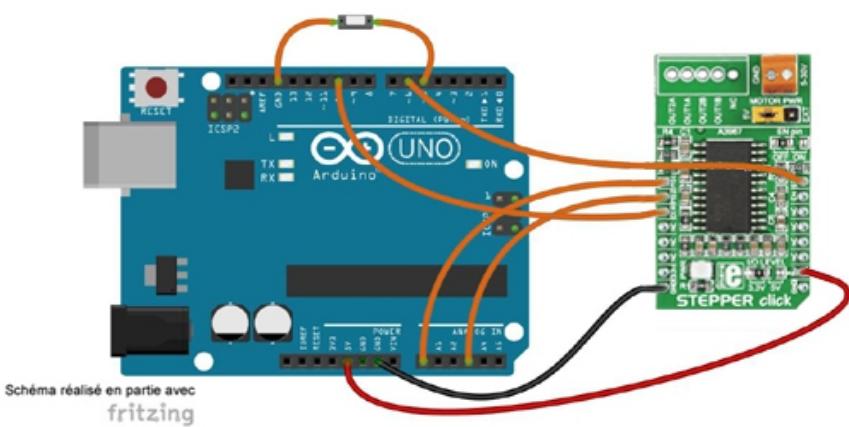


Ce module intègre un contrôleur de moteur pas-à-pas de type A3967SLBT. Ce circuit est capable de piloter un moteur pas-à-pas d'une tension pouvant être comprise entre 5 à 30 Vcc avec une consommation max. de 500 mA.

## Montage à réaliser



Enfichez le module MIKROE-1528 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1528 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Stepper Click"  
*  
*****  
* Le moteur tourne dans un sens ou l'autre défini par l'état d'un bouton-poussoir  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Stepper Click" (Réf.: MIKROE-1528) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* 1 Moteur pas à pas 28BYJ-48  
* 1 bouton poussoir  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define MS1 14  
#define MS2 17  
#define DIR 10  
#define STEP 6  
#define BP 5  
  
boolean sens;  
  
void setup()  
{  
pinMode(MS1, OUTPUT); // configuration des broches en sortie  
pinMode(MS2, OUTPUT);  
pinMode(DIR, OUTPUT);  
pinMode(STEP, OUTPUT);  
pinMode(BP, INPUT_PULLUP); // configuration de la broche 5 en entrée avec résistance de tirage  
digitalWrite(MS1,LOW); // configuration du A3967 en mode pas entier  
digitalWrite(MS2,LOW);  
}
```

```
void loop()
{
sens=digitalRead(BP); // acquisition de l'état du bouton
digitalWrite(DIR,sens); // configuration du sens de rotation
delay(1);
digitalWrite(STEP, HIGH); // envoi d'un front descendant sur l'entrée STEP du A3967
delay(1);
digitalWrite(STEP, LOW);
delay(1);
}
```

## Application 018 : MIKROE-1627 : Module Thumbstick click Board

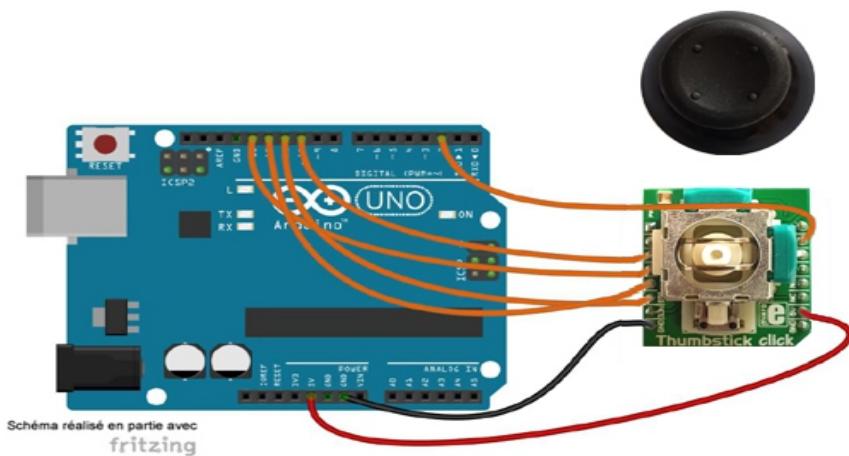


Ce module Click Board vous permettra d'ajouter un mini-joystick (similaire à ceux que l'on retrouve dans les manettes de jeux des consoles de salon) sur votre microcontrôleur.

### Montage à réaliser



Enfichez le module MIKROE-1627 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1627 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



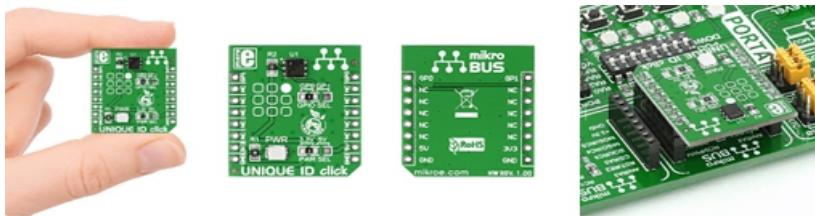
## Programme Arduino

```
*****  
*  
* Test du module "Thumbstick Click"  
*  
*****  
* La position du joystick et l'état du bouton sont affichés dans le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Thumbstick Click" (Réf.: MIKROE-1627) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque  
* 1. https://github.com/Rom3oDelta7/MCP320X  
*****/  
  
// Affectation des broches  
#define CS 10  
#define SCK 13  
#define SDI 11  
#define SDO 12  
#define SW 2  
  
#define MCP3204 4 // référence du MPC320X  
  
int x;  
int y;  
boolean bp;  
  
#include "MCP320X.h" // appel de la librairie  
MCP320X joystick(MCP3204, SCK, SDI, SDO, CS); // création de l'objet joystick  
  
void setup()  
{
```

```
Serial.begin(9600); // initialisation de la liaison série
pinMode(SW,INPUT); // configuration de la broche SW en entrée
}

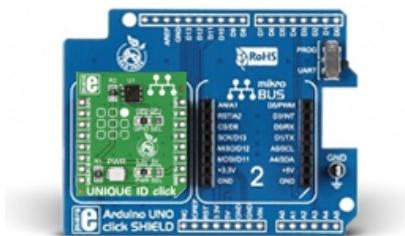
void loop()
{
bp=digitalRead(SW); // lecture de l'état du bouton
x=joystick.readADC(1); // acquisition de la valeur x
y=joystick.readADC(0); // acquisition de la valeur y
// Affichage dans le moniteur série
Serial.print("Valeur x du joystick:");
Serial.println(x);
Serial.print("Valeur y du joystick:");
Serial.println(y);
Serial.print("Etat du bouton:");
Serial.println(bp);
delay(500);
}
```

## Application 019 : MIKROE-1819 : Module ID click



Ce petit module "click Board" intègre un circuit intégré DS2401 lequel fait office de dispositif d'identification. Ce composant intègre une ROM de 64 bits pré-programmée en usine.

### Montage à réaliser



Enfichez le module MIKROE-1819 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1819 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Unique ID Click"  
*  
*****  
* Le numéro de série du module est affiché dans le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Unique ID Click" (Réf.: MIKROE-1819) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque  
* 1. https://github.com/PaulStoffregen/OneWire  
*  
*****/  
  
#include <OneWire.h> // appel de la bibliothèque  
  
#define GPIO0 6 // affectation des broches  
  
// Variables de stockage des 8 octets du DS2401: 1 octet pour la famille, 6 octets pour le N° de série et 1 octet pour  
le CRC  
byte data[8];  
  
OneWire DS2401(GPIO0); // création de l'objet DS2401  
  
void setup()  
{  
Serial.begin(9600); // initialisation de la liaison série  
Serial.println("Test du module Unique ID Click");  
Serial.println(" ");  
  
if (DS2401.reset())
```

```

{
DS2401.write(0x33); // demande de lecture de la ROM
data[0] = DS2401.read(); // acquisition du N° de famille
Serial.print("Code famille: 0x");
Affichage_code(data[0]);
Serial.print("Numero de serie: 0x");
for (byte i = 1; i <= 6; i++) // acquisition des 6 octets du N° de série
{
data[i] = DS2401.read();
Affichage_code(data[i], (i < 6) ? " : 'n');
}
data[7] = DS2401.read(); // acquisition du CRC
if (data[7] == OneWire::crc8(data, 7))
{
Serial.print("CRC du DS2401 correct = 0x");
Affichage_code (data[7]);
}
else
{
Serial.println("Erreur de CRC");
}
}

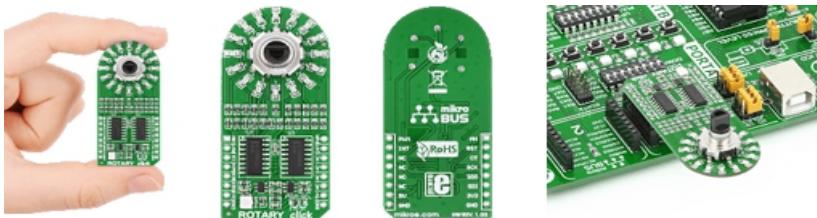
void loop()
{



void Affichage_code(byte donnee, const char separateur = 'n') // fonction permettant d'afficher dans le moniteur
série l'octet en hexadécimal
{
if (donnee <= 0xF)
{
Serial.print(0);
}
Serial.print(donnee, HEX);
if (separateur)
{
Serial.write(separateur);
}
}
}

```

Application 020 : MIKROE-1823 : Module Rotary R click



Ce module Click Board vous permettra d'ajouter un encodeur rotatif ECD12 15-pulse (avec fonction poussoir) et 16 leds de visualisation rouges à votre application. Ce dernier s'interface via une liaison SPI.

## Montage à réaliser



Enfichez le module MIKROE-1823 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1823 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

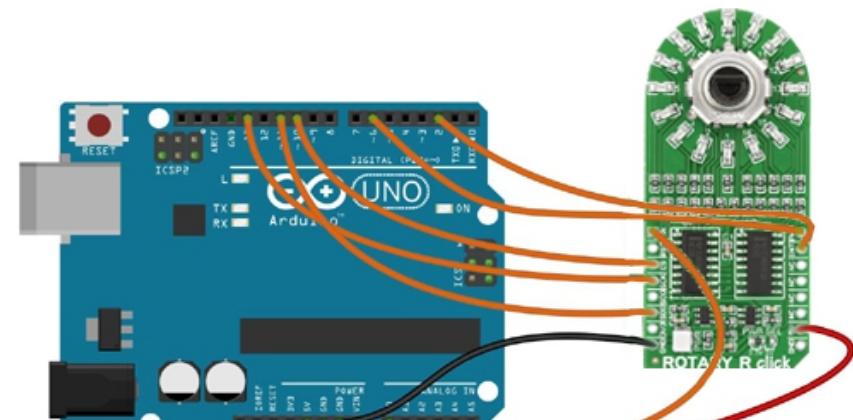


Schéma réalisé en partie avec  
fritzing

## Programme Arduino

```
*****  
*  
* Test du module "Rotary R Click"  
*  
*****  
* L'évolution d'une variable entre 0 et 65535 est affichée dans le moniteur  
* série et sur les led du module. Le bouton poussoir de l'encodeur permet de  
* faire une remise à zero  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Rotary R Click" (Réf.: MIKROE-1823) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Attention : il faut insérer une résistance de 10K entre la broche 2 et la masse  
*  
*****/  
  
// Affectation des broches  
#define SDI 11 // broche DS des 74HC595  
#define SCK 13 // broche SHCP des 74HC595  
#define LATCH 10 // broche STCP des 74HC595  
#define SW 2 // bouton poussoir de l'encodeur  
#define ENCA 6 // sortie A de l'encodeur  
#define ENCB 14 // sortie B de l'encodeur  
  
int position_encodeur=0;  
boolean encodeur_A=LOW;  
boolean etat_bouton=LOW;  
boolean precedent_encodeur_A=LOW;  
volatile unsigned int nombre=0;  
  
void setup()  
{  
pinMode(SDI, OUTPUT);
```

```

pinMode(SCK, OUTPUT);
pinMode(LATCH, OUTPUT);
pinMode(SW, INPUT);
pinMode(ENCA, INPUT);
pinMode(ENCB, INPUT);
Serial.begin(9600);
attachInterrupt(0, raz, RISING); // autorisation des interruptions sur la broche N°2
}

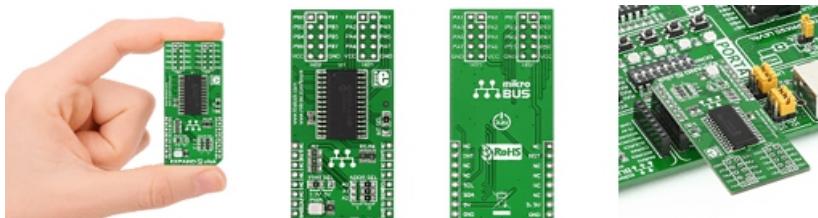
void loop()
{
Serial.println(nombre); // affichage dans le moniteur série
allumage_led(nombre); // appel de la fonction allumage_led
encodeur_A = digitalRead(ENCA); // acquisition de l'état de la broche ENCA
if ((precedent_encodeur_A == LOW) && (encodeur_A == HIGH)) // si l'état a changé par rapport à l'état précédent
{
if (digitalRead(ENCB) == HIGH) // si la broche ENCB est à l'état haut
{
nombre--; // la variable nombre diminue de 1
}
else
{
nombre++; // la variable nombre augmente de 1
}
}
precedent_encodeur_A = encodeur_A; // mémorisation de l'état précédent
}

void allumage_led(unsigned int led)
{
digitalWrite(LATCH, LOW);
shiftOut(SDI, SCK, MSBFIRST, (led >> 8));
shiftOut(SDI, SCK, MSBFIRST, (led & 0xFF));
digitalWrite(LATCH, HIGH);
}

void raz()
{
nombre=0; // remise à 0 de la variable nombre
allumage_led(nombre);
}

```

## Application 021 : MIKROE-1838 Expand 2 Click

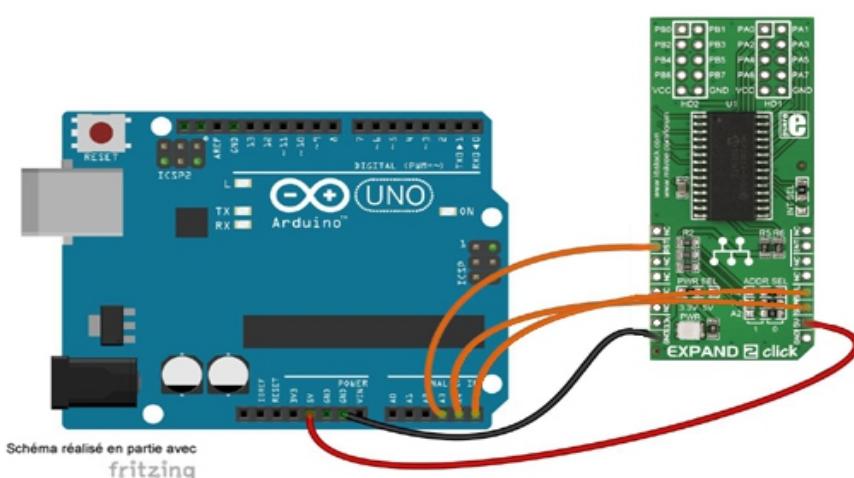


Ce module Click Board vous permettra d'ajouter 16 ports d'entrées/sorties à votre microcontrôleur grâce à un circuit intégré MCP23017. Ce dernier se pilote par le biais d'un bus de communication I2C™.

### Montage à réaliser



Enfichez le module MIKROE-1838 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1838 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



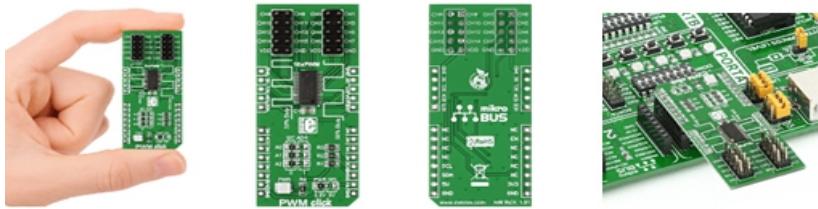
## Programme Arduino

```
*****  
*  
* Test du module "Expand 2 Click"  
*  
*****  
* Huit boutons poussoirs branchés sur le PORT B commandent respectivement  
* huit leds branchées sur le PORT A  
* Les anodes des leds sont reliées au PORT A au travers d'une résistance  
* de protection et les cathodes à la masse  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Expand 2 Click" (Réf.: MIKROE-1838) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
#include <Wire.h> // appel de la bibliothèque  
#define MCP23017_Adresse 0x20 // adresse I2C du MCP23017  
#define IODIRA_Adresse 0x00 // adresse du registre IODIRA  
#define IODIRB_Adresse 0x01 // adresse du registre IODIRB  
#define GPIOA_Adresse 0x12 // adresse du registre GPIOA  
#define GPIOB_Adresse 0x13 // adresse du registre GPIOB  
#define GPPUB_Adresse 0x0D // adresse du registre GPPUB  
#define RST 17 // affectation des broches  
  
byte entree;  
  
void setup()  
{  
pinMode(RST,OUTPUT);  
digitalWrite(RST,HIGH);  
Wire.begin(); // initialisation de la liaison I2C  
Wire.beginTransmission(MCP23017_Adresse); // communication avec le MCP23017  
Wire.write(IODIRA_Adresse); // écriture dans le registre IODIRA
```

```
Wire.write(0x00); // configuration du PORT A en sortie
Wire.write(IODIRB_Adresse); // écriture dans le registre IODIRB
Wire.write(0xFF); // configuration du PORT B en entrée
Wire.write(GPPUB_Adresse); // écriture dans le registre GPPUB
Wire.write(0xFF); // configuration des résistances de tirage
Wire.endTransmission();
}

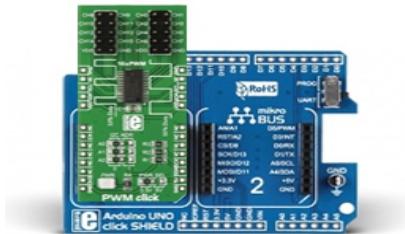
void loop()
{
// Lecture des boutons poussoirs
Wire.beginTransmission(MCP23017_Adresse); // communication avec le MCP23017
Wire.write(GPIOB_Adresse); // écriture dans le registre GPIOB
Wire.endTransmission();
Wire.requestFrom(MCP23017_Adresse, 1); // demande d'envoi d'un octet
entree=Wire.read(); // mémorisation dans la variable entree
// Allumage des led
Wire.beginTransmission(MCP23017_Adresse); // communication avec le MCP23017
Wire.write(GPIOA_Adresse); // écriture dans le registre GPIOA
Wire.write(entree);
Wire.endTransmission();
delay(100);
}
```

## Application 22 : MIKROE-1898 : Module PWM click

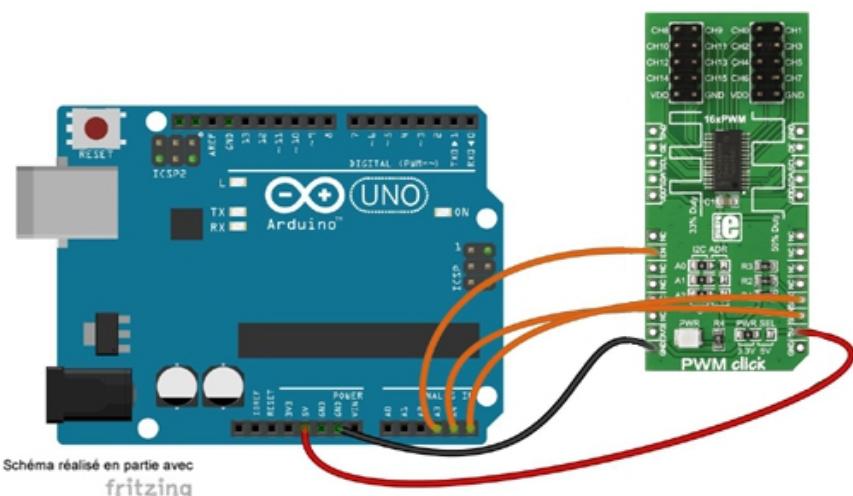


Equipé d'un circuit CA9685PW, ce module Click Board permettra à votre microcontrôleur de piloter 16 sorties PWM à partir d'une simple liaison I2C™. Ce dernier est idéal pour piloter des leds ou des servomoteurs.

### Montage à réaliser



Enfichez le module MIKROE-1898 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1898 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

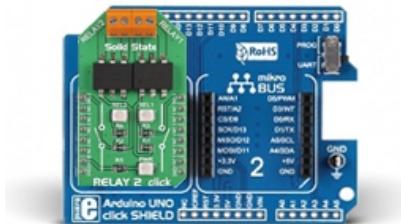
```
*****  
* Test du module "PWM Click"  
*****  
* La sortie CH0 du module délivre un signal carré ayant une fréquence de 50 Hz  
* et un rapport cyclique de 50%  
*  
* Matériel  
* 1 Arduino Uno (A000066) + 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "PWM Click" (Réf.: MIKROE-1898) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de la platine shield MIKROE-1581  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque * 1. https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library  
*  
*****/  
#include <Wire.h> // appel des bibliothèques  
#include <Adafruit_PWMServoDriver.h>  
#define OE 17 // affectation des broches  
  
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40); // création de l'objet pwm  
  
#define frequence 50  
#define numero_sortie 0  
  
void setup()  
{  
    pwm.begin();  
    pwm.setPWMFreq(frequence); // configuration de la fréquence du signal PWM  
    pinMode(OE,OUTPUT); // configuration de la broche OE en sortie  
    digitalWrite(OE,LOW); // validation des sorties du PCA9685  
}  
  
void loop()  
{  
    pwm.setPWM(numero_sortie, 0, 2048); // génération du signal PWM  
}
```

## Application 23 : MIKROE-1899 : Module Relay 2 click



Ce petit module "click Board" intègre 2 relais statiques OptoMOS® haute performance. Silencieux, rapides, très peu soumis aux problèmes de rebond, ces relais disposent d'une résistance de contact de 0,15 ohms.

### Montage à réaliser



Enfichez le module MIKROE-1899 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1899 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

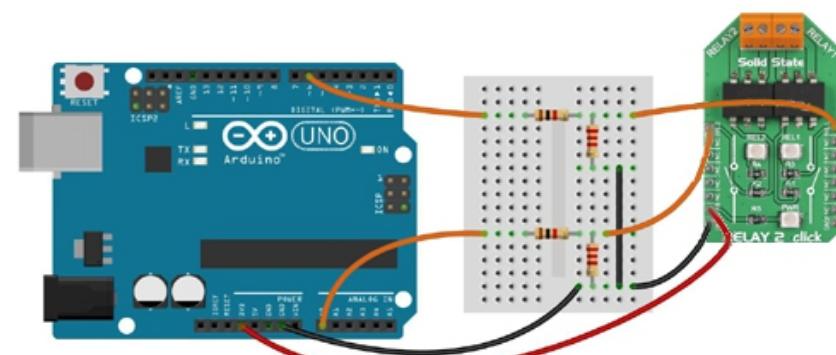
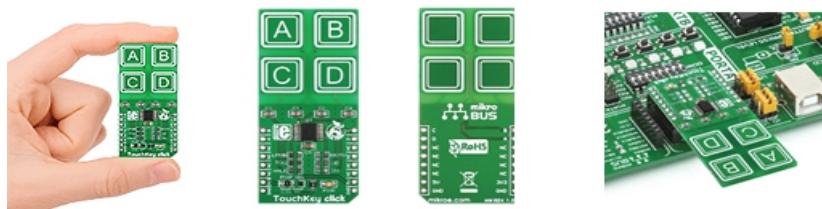


Schéma réalisé en partie avec  
fritzing

## Programme Arduino

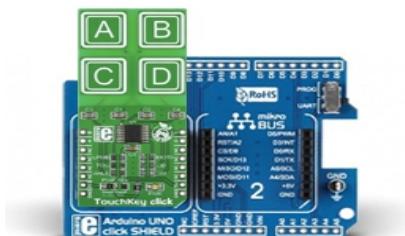
```
*****  
* Test du module "Relay 2 Click"  
*****  
* Les relais sont activés un à un avec une pause d'une seconde entre chaque étape  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Relay 2 Click" (Réf.: MIKROE-1899) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Attention, ce module est alimenté en 3,3V. Il faut donc faire une adaptation  
* de tension entre les broches de sorties de l'Arduino et les broches d'entrées  
* du module  
*  
*****/  
  
// Affectation des broches  
#define REL1 6  
#define REL2 14  
  
void setup()  
{  
pinMode(REL1,OUTPUT); // configuration des broches en sortie  
pinMode(REL2,OUTPUT);  
}  
  
void loop()  
{  
digitalWrite(REL1,HIGH); // Relais 1 actif  
digitalWrite(REL2,LOW);  
delay(1000); // pause d'une seconde  
digitalWrite(REL1,LOW); // Relais 2 actif  
digitalWrite(REL2,HIGH);  
delay(1000); // pause d'une seconde  
}
```

## Application 24 : MIKROE-1906 : Module Touch Key click



Ce petit module "click Board" intègre 4 touches capacitatives générées par un capteur TTP224. Ces touches peuvent également être sollicitées au travers d'une vitre de faible épaisseur ou d'une feuille de papier.

### Montage à réaliser



Enfichez le module MIKROE-1906 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1906 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

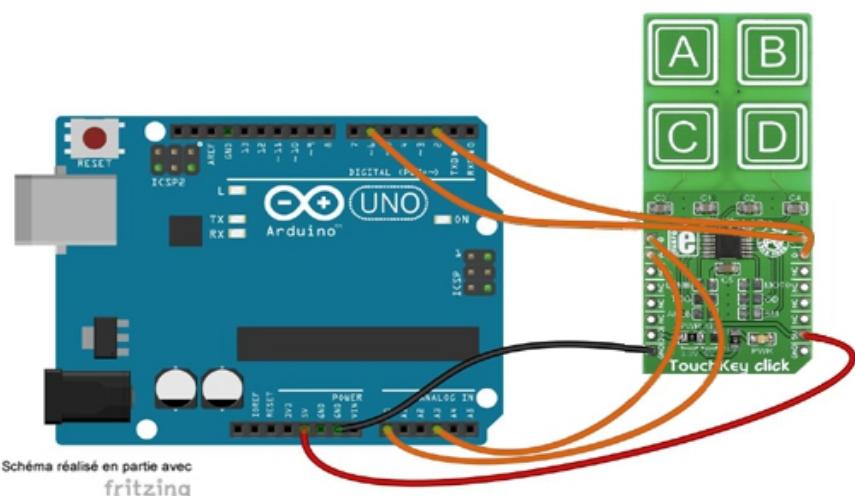


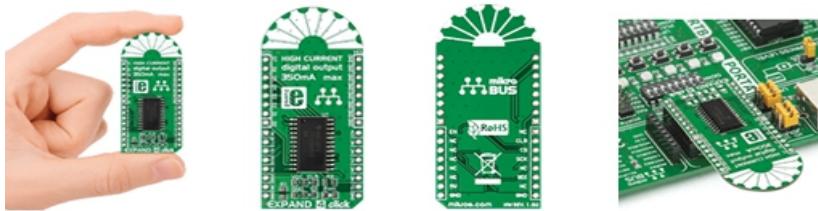
Schéma réalisé en partie avec  
fritzing

## Programme Arduino

```
*****  
*  
* Test du module "TouchKey Click"  
*  
*****  
* L'état des boutons est affiché dans le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "TouchKey Click" (Réf.: MIKROE-1906) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define OUTA 17  
#define OUTB 14  
#define OUTC 6  
#define OUTD 2  
  
byte broche[4]={OUTA,OUTB,OUTC,OUTD};  
boolean etat[4];  
void setup()  
{  
Serial.begin(9600); // initialisation de la liaison série  
for (int i=0;i<4;i++)  
{  
pinMode(broche[i],INPUT); // configuration des broches en entrée  
}  
}  
  
void loop()  
{  
for (int i=0; i <4; i++)  
{
```

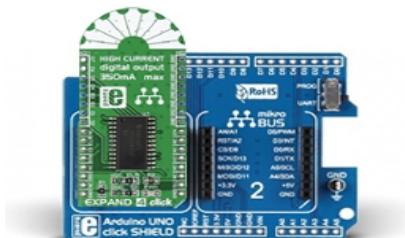
```
etat[i] = digitalRead(broche[i]); // acquisition de l'état des boutons
Serial.print("Bouton"); // affichage dans le moniteur série
Serial.print(i);
Serial.print("=");
Serial.println(etat[i]);
}
delay(300);
}
```

## Application 25 : MIKROE-1910 : Module Expand 4 click

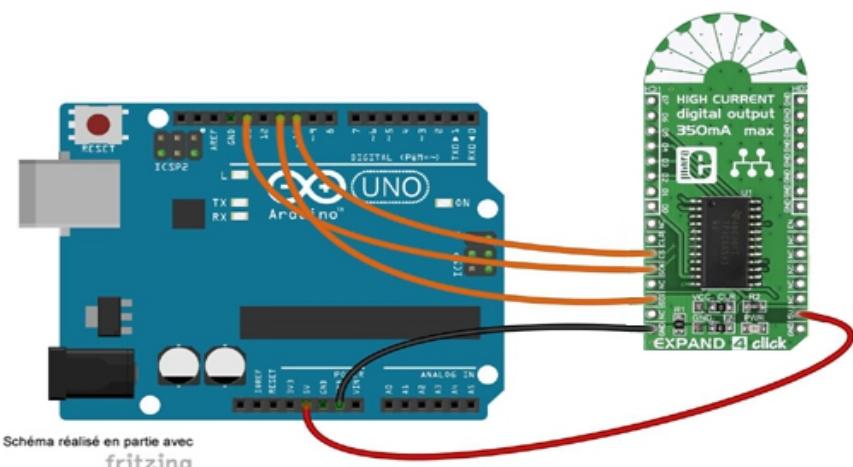


Ce module Click Board vous permettra d'ajouter 8 sorties de "puissance" à votre microcontrôleur grâce à un circuit intégré TPCIC6A595. Ce dernier se pilote par le biais d'un bus de communication SPI™.

### Montage à réaliser



Enfichez le module MIKROE-1910 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1910 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Expand 4 Click"  
*  
*****  
* Les huit sorties du module passent successivement à l'état haut  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Expand 4 Click" (Réf.: MIKROE-1910) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
// Affectation des broches  
#define MOSI 11 // broche SERIN du TPIC6A595  
#define SCK 13 // broche SRCK du TPIC6A595  
#define CS 10 // broche RCK du TPIC6A595  
  
byte sortie=0;  
int i=0;  
  
void setup()  
{  
// Configuration des broches en sortie  
pinMode(MOSI, OUTPUT);  
pinMode(SCK, OUTPUT);  
pinMode(CS, OUTPUT);  
}  
  
void loop()  
{  
if (i == 7)  
{  
i = 0;
```

```
}

else
{
i++;
}

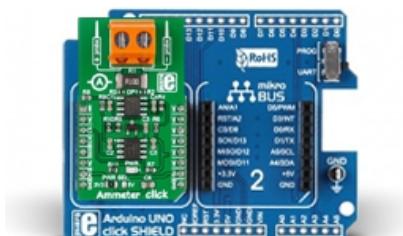
bitSet(sortie, i); // mise à 1 du bit i de la variable sortie
digitalWrite(CS, LOW);
shiftOut(MOSI, SCK, LSBFIRST, sortie); // envoi bit par bit de la variable sortie
digitalWrite(CS, HIGH);
delay(250);
}
```

## Application 26 : MIKROE-2377 : Module Ammeter click

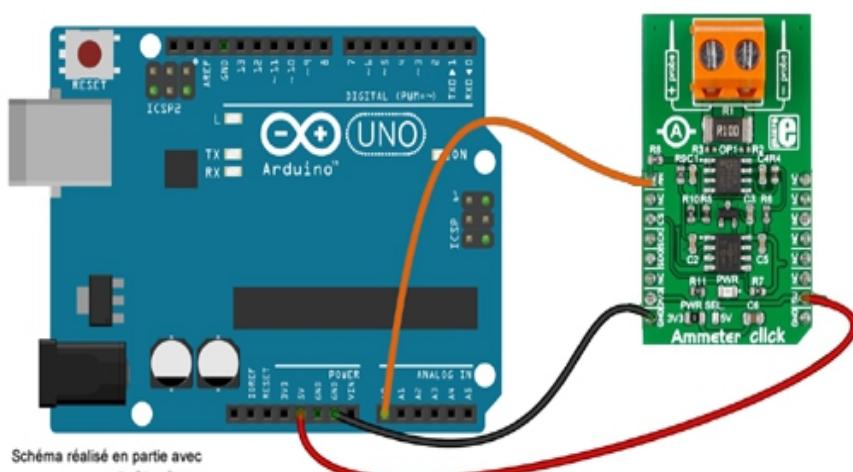


Ce module Click Board vous permettra de mesurer des courants (AC et DC) à l'aide d'une sonde (+ et -) sur des borniers à vis à l'aide de votre microcontrôleur grâce à un jeu de composants intégrés (MCP3201, MAX6106, AD8616).

### Montage à réaliser



Enfichez le module MIKROE-2377 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2377 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

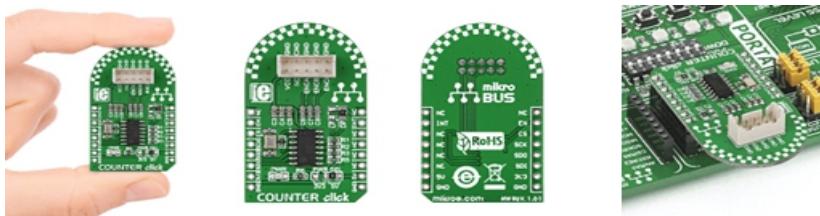


## Programme Arduino

```
*****  
*  
* Test du module "Ammeter Click"  
*  
*****  
* La valeur moyenne du courant est affichée sur le moniteur série.  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "ammeter Click" (Réf.: MIKROE-2377) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define AN A0 // broche de sortie du module  
int echantillon=1000;  
int valeur;  
int courant;  
unsigned long somme=0;  
unsigned long moyenne;  
  
void setup()  
{  
Serial.begin(9600); // initialisation de la liaison série  
}  
  
void loop()  
{  
for (int i=0; i<echantillon; i++)  
{  
valeur=analogRead(AN); // conversion AN  
Serial.print("Valeur=");  
Serial.println(valeur);
```

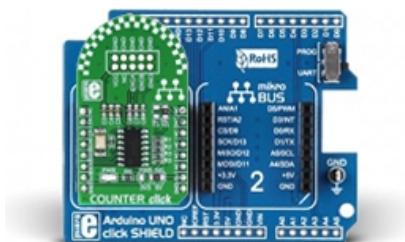
```
somme=somme+valeur;
moyenne=somme/echantillon;
/* Le module comporte un amplificateur différentiel ayant pour fonction de transfert Vs=Vref+10Ve
Ve=Rshunt*I avec Rshunt=0.1 donc Vs=1.024+I
Le courant d'entrée du module doit être compris entre 1 et 1000 mA, ce qui correspond à des tensions de sortie
de 1,025V soit une valeur du CAN à 207 et de 2,024V soit une valeur du CAN à 414*/
courant=map(moyenne,207,414,1,1000);
delay(10);
}
Serial.print("Courant moyen=");
Serial.print(courant);
Serial.println(" mA");
delay(500);
somme=0;
}
```

## Application 27 : MIKROE-1917 : Module Counter click

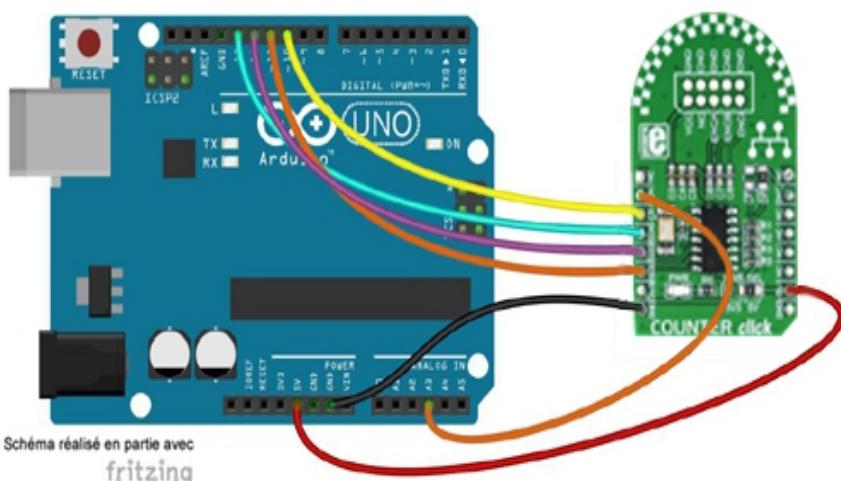


Ce petit module "click Board" intègre un circuit intégré compteur de quadrature 32 bits de type LS7366R. Ce dernier est destiné à être raccordé sur un encodeur de moteur (non livré). Dès lors il sera capable de déterminer sa vitesse et son sens de rotation.

### Montage à réaliser



Enfichez le module MIKROE-1917 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1917 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Counter Click"  
*  
*****  
* L'évolution du nombre de tours de l'encodeur rotatif est affichée  
* sur le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Counter Click" (Réf.: MIKROE-1917) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* 1 Encodeur rotatif KY-040  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****  
  
#include <SPI.h> // appel de la bibliothèque  
  
// Affectation des broches  
#define CS 10 // broches de la liaison SPI  
#define MOSI 11  
#define MISO 12  
#define SCK 13  
#define CNT_EN 17 // entrée de validation du LS7366  
  
// Configuration des registres  
#define CLR_MDR0 0x08  
#define CLR_MDR1 0x10  
#define CLR_CNTR 0x20  
#define CLR_STR 0x30  
#define READ_MDR0 0x48  
#define READ_MDR1 0x50  
#define READ_CNTR 0x60  
#define READ_OTR 0x68  
#define READ_STR 0x70
```

```

#define WRITE_MDR1 0x90
#define WRITE_MDR0 0x88
#define WRITE_DTR 0x98
#define LOAD_CNTR 0xE0
#define LOAD_OTR 0xE4

byte valeur=0;

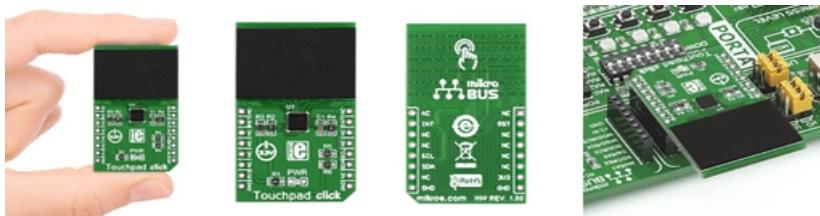
void setup()
{
pinMode(CS, OUTPUT);
pinMode(MOSI, OUTPUT);
pinMode(SCK, OUTPUT);
pinMode(MISO, INPUT);
digitalWrite(CNT_EN,HIGH); // validation du comptage

// Initialisation et configuration de la liaison SPI
SPI.begin();
SPI.setDataMode(SPI_MODE0);
SPI.setBitOrder(MSBFIRST);
SPI.setClockDivider(SPI_CLOCK_DIV8);
Serial.begin(9600); // initialisation de la liaison série
Serial.println("Initialisation.....");
// Configuration du LS7366
digitalWrite(CS,LOW); // début de la communication SPI
SPI.transfer(WRITE_MDR0); // écriture dans le registre MDR0
SPI.transfer(0x01); // mode quadrature
digitalWrite(CS,HIGH); // fin de la communication SPI
digitalWrite(CS,LOW); // début de la communication SPI
SPI.transfer(WRITE_MDR1); // écriture dans le registre MDR1
SPI.transfer(0x03); // mode comptage sur un octet
digitalWrite(CS,HIGH); // fin de la communication SPI
digitalWrite(CS,LOW); // début de la communication SPI
SPI.transfer(CLR_CNTR); // RAZ du compteur
digitalWrite(CS,HIGH); // fin de la communication SPI
digitalWrite(CS,LOW); // début de la communication SPI
SPI.transfer(CLR_STR); // RAZ du registre d'état STR
digitalWrite(CS,HIGH); // fin de la communication SPI
delay(1000);
}

```

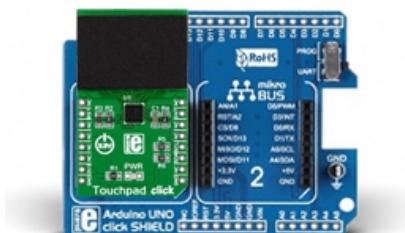
```
void loop()
{
digitalWrite(CS,LOW); // début de la communication SPI
SPI.transfer(READ_CNTR); // lecture du compteur
valeur=SPI.transfer(0x00);
digitalWrite(CS,HIGH); // fin de la communication SPI
Serial.print("Valeur : "); // affichage dans le moniteur série
Serial.println(valeur);
}
```

## Application 28 : MIKROE-1995 : Module Touch Pad click

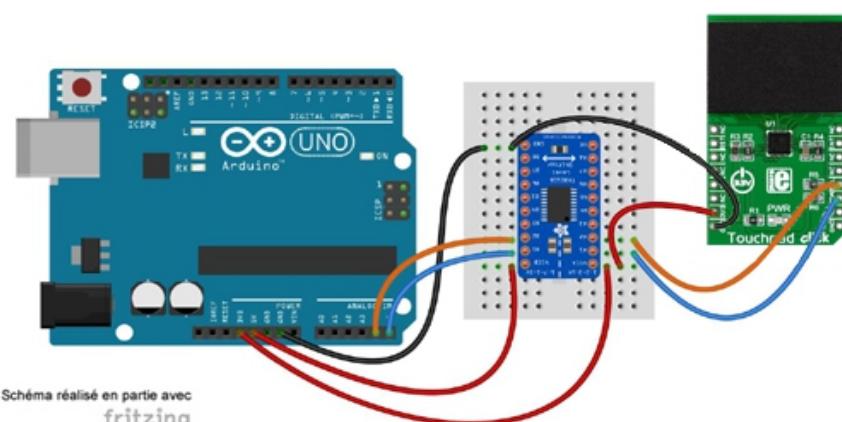


Ce "click Board" intègre un contrôleur MTCH6102, lequel s'apparente à un capteur capacitif. Ce dernier est recouvert d'une fine couche de plastique faisant office de "touchpad" (l'épaisseur de la couche de plastique peut aller jusqu'à 3 mm max. ou jusqu'à 5 mm pour du verre).

### Montage à réaliser



Enfichez le module MIKROE-1995 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1995 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Touchpad Click"  
*  
*****  
* Les valeurs des registres donnant la position sont affichés dans le moniteur série  
*  
* Exemple inspiré de  
* https://github.com/theapi/touch_timer/tree/master/arduino/mtch6102_proof_of_concept  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Touchpad Click" (Réf.: MIKROE-1995) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* 1 Shifter Level TXB0108  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Attention, ce module est alimenté en 3,3V. Il faut donc utiliser un module shifter level I2C  
*  
*****/  
  
#include <Wire.h>  
#define ADDR 0x25  
  
// read register  
unsigned char readRegister(unsigned char addr_reg) {  
byte error;  
Wire.beginTransmission(ADDR);  
Wire.write(addr_reg); // register to read  
error = Wire.endTransmission();  
Wire.requestFrom(ADDR, 1); // read a byte  
  
while (Wire.available()) {  
return Wire.read();  
}  
}
```

```
// write data to register
unsigned char writeRegister(unsigned char addr_reg, unsigned char dta) {
Wire.beginTransmission(ADDR);
Wire.write(addr_reg); // register to read
Wire.write(dta);
Wire.endTransmission();
}

void setup()
{
byte data;
Wire.begin();
Serial.begin(115200);

delay(500);

// the operating mode (MODE)
data = readRegister(0x05);
Serial.print("MODE: ");
Serial.println(data,BIN);

// Set mode to Touch only
writeRegister(0x05, 0x02);

data = readRegister(0x05);
Serial.print("MODE: ");
Serial.println(data,BIN);

data = readRegister(0x20);
Serial.print("NUMBEROFXCHANNELS: ");
Serial.println(data);

data = readRegister(0x21);
Serial.print("NUMBEROFYCHANNELS: ");
Serial.println(data);

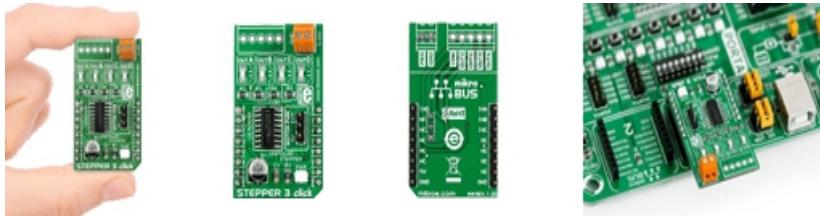
writeRegister(0x20, 0x07);
writeRegister(0x21, 0x06);
```

```
data = readRegister(0x20);
Serial.print("NUMBEROFXCHANNELS: ");
Serial.println(data);

data = readRegister(0x21);
Serial.print("NUMBEROFYCHANNELS: ");
Serial.println(data);
}

void loop()
{
byte data;
Serial.print("SENSORVALUE_RX <i>: ");
for (byte i = 0x80; i < 0x8E; i++)
{
data = readRegister(i);
Serial.print(data);
Serial.print(", ");
}
Serial.println();
delay(500);
}
```

## Application 29 : MIKROE-2035: Module Stepper 3 click

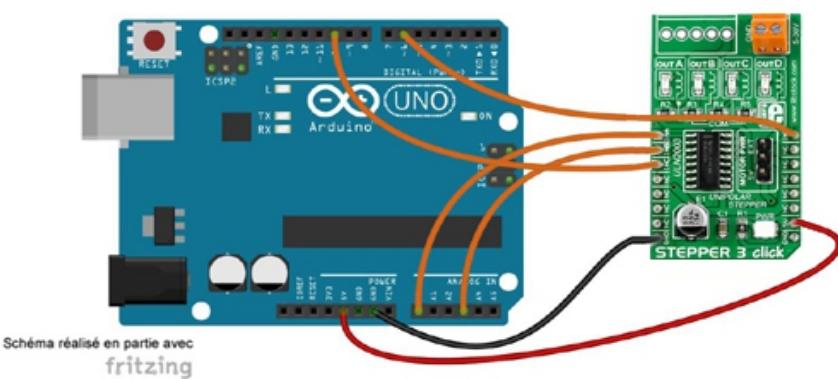


Ce petit module "click Board" intègre un contrôleur de moteur pas-à-pas de type ULN2003. Ce circuit est capable de piloter un moteur pas-à-pas unipolaire d'une tension pouvant aller jusqu'à 30 Vcc. Chaque phase est accessible séparément.

### Montage à réaliser



Enfitez le module MIKROE-2035 sur la platine MIKROE-1581. Puis enfitez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2035 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Stepper 3 Click"  
*  
*****  
* Le moteur fait un tour dans un sens puis revient à sa position d'origine dans l'autre sens  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Stepper 3 Click" (Réf.: MIKROE-2035) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* 1 Moteur pas à pas 28BYJ-48  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
#include <Stepper.h> // appel de la bibliothèque  
  
// Affectation des broches  
#define INA 14  
#define INB 17  
#define INC 10  
#define IND 6  
  
const int nombre_pas=2048; // nombre de pas du moteur  
Stepper moteur(nombre_pas, INA, INC, INB, IND); // création de l'objet moteur  
  
void setup()  
{  
moteur.setSpeed(10); // initialisation la vitesse de rotation du moteur en tour par minute  
pinMode(INA, OUTPUT); // configuration des broches en sortie  
pinMode(INB, OUTPUT);  
pinMode(INC, OUTPUT);  
pinMode(IND, OUTPUT);  
}
```

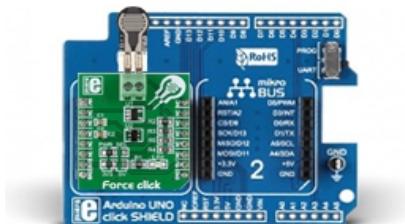
```
void loop()
{
for (int i=1; i<=nombre_pas; i++) // boucle avance du moteur en fonction du nombre de pas
{
moteur.step(1); // un pas en sens positif
}
for (int i=1; i<=nombre_pas; i++) // boucle retour du moteur en fonction du nombre de pas
{
moteur.step(-1); // un pas en sens négatif
}
}
```

## Application 30 : MIKROE-2065 : Module Force click

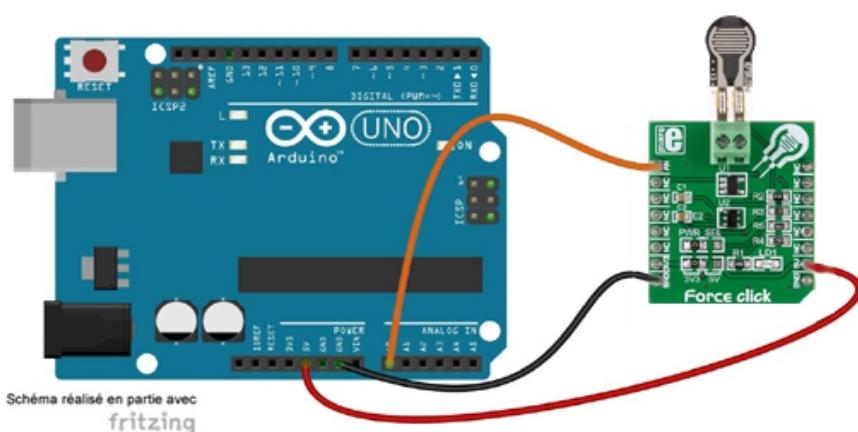


Ce module Click Board équipé d'un capteur de force vous permettra de mesurer des pressions mécaniques (caractérisées par un changement de valeur résistive sous l'action d'une force) avec votre microcontrôleur via une entrée analogique.

### Montage à réaliser



Enfichez le module MIKROE-2065 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2065 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Force Click"  
*  
*****  
* La valeur de la force (entre 0 et 1024) est affichée sur le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Force Click" (Réf.: MIKROE-2065) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define AN A0 // broche de sortie du module  
int valeur;  
  
void setup()  
{  
Serial.begin(9600); // initialisation de la liaison série  
}  
  
void loop()  
{  
valeur=analogRead(AN); // conversion AN  
Serial.print("Valeur=");  
Serial.println(valeur);  
delay(500);  
}
```

## Application 31 : MIKROE-2154 : Module Signal Relay click

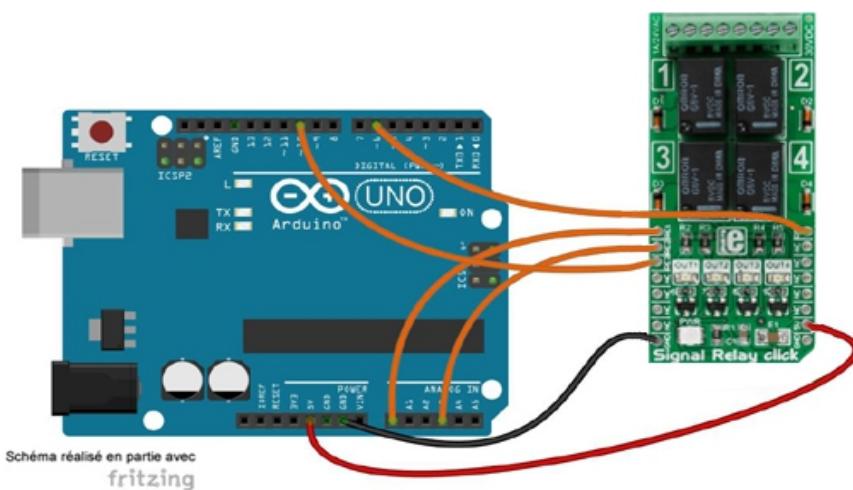


Ce module Click Board vous permettra d'ajouter 4 sorties relais à votre microcontrôleur.

### Montage à réaliser



Enfichez le module MIKROE-2154 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2154 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Signal Relay Click"  
*  
*****  
* Les relais sont activés un à un avec une pause d'une seconde entre chaque étape  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Signal Relay Click" (Réf.: MIKROE-2154) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define REL1 14  
#define REL2 17  
#define REL3 10  
#define REL4 6  
  
void setup()  
{  
pinMode(REL1,OUTPUT); // configuration des broches en sortie  
pinMode(REL2,OUTPUT);  
pinMode(REL3,OUTPUT);  
pinMode(REL4,OUTPUT);  
}  
  
void loop()  
{  
digitalWrite(REL1,HIGH); // Relais 1 actif  
digitalWrite(REL2,LOW);  
digitalWrite(REL3,LOW);  
digitalWrite(REL4,LOW);  
delay(1000); // pause d'une seconde
```

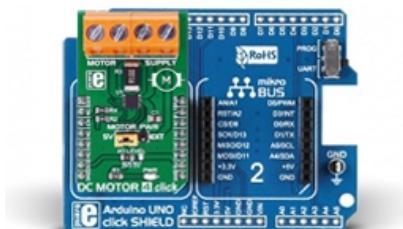
```
digitalWrite(REL1,LOW); // Relais 2 actif
digitalWrite(REL2,HIGH);
digitalWrite(REL3,LOW);
digitalWrite(REL4,LOW);
delay(1000); // pause d'une seconde
digitalWrite(REL1,LOW); // Relais 3 actif
digitalWrite(REL2,LOW);
digitalWrite(REL3,HIGH);
digitalWrite(REL4,LOW);
delay(1000); // pause d'une seconde
digitalWrite(REL1,LOW); // Relais 1 actif
digitalWrite(REL2,LOW);
digitalWrite(REL3,LOW);
digitalWrite(REL4,HIGH);
delay(1000); // pause d'une seconde
}
```

## Application 32 : MIKROE-2221 : Module DC Motor 4 click

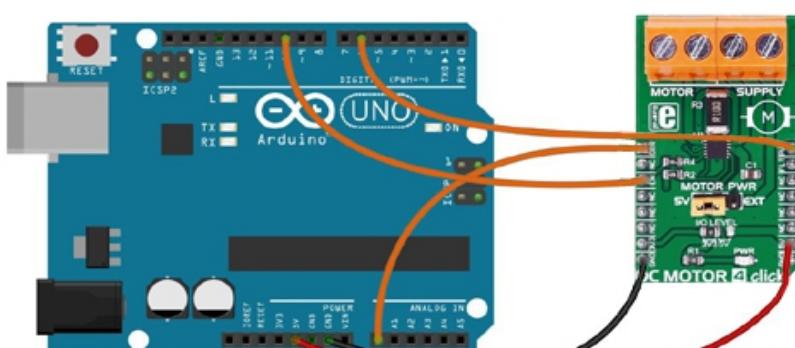


Ce petit module "click Board" intègre un contrôleur de moteur "cc" de type MAX14870. Ce contrôleur vous permettra de contrôler la vitesse et le sens de rotation d'un moteur (non livré) ainsi que son freinage en régulant le courant.

### Montage à réaliser



Enfichez le module MIKROE-2221 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2221 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

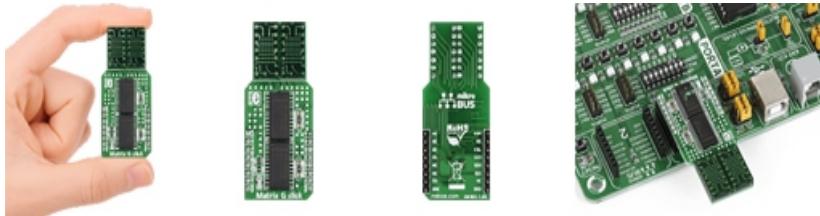


## Programme Arduino

```
*****  
*  
* Test du module "DC Motor 4 Click"  
*  
*****  
* La vitesse du moteur augmente par pas successif toutes les 10 ms  
* jusqu'à la vitesse maximale puis le cycle recommence mais le moteur  
* tourne dans l'autre sens  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "DC Motor 4 Click" (Réf.: MIKROE-2221) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define DIR 14  
#define EN 10  
#define PWM 6  
  
int i;  
  
void setup()  
{  
pinMode(DIR,OUTPUT); // configuration des broches  
pinMode(PWM,OUTPUT);  
pinMode(EN,OUTPUT);  
digitalWrite(EN,LOW); // validation du max14870  
}  
  
void loop()  
{  
digitalWrite(DR,HIGH); // sens horaire  
for(i = 0;i<256;i++) // variation de la vitesse toutes les 10 ms
```

```
{  
analogWrite(PWM, i);  
delay(10);  
}  
delay(1000); // pause d'une seconde à vitesse maximale  
analogWrite(PWM, 0); // arrêt du moteur  
delay(1000); // pause d'une seconde entre les deux sens de rotation  
digitalWrite(DIR,LOW); // sens anti-horaire  
for(i = 0;i<256;i++) // variation de la vitesse toutes les 10 ms  
{  
analogWrite(PWM, i);  
delay(10);  
}  
delay(1000); // pause d'une seconde à vitesse maximale  
analogWrite(PWM, 0); // arrêt du moteur  
delay(1000); // pause d'une seconde entre les deux sens de rotation  
}
```

## Application 33 : MIKROE-2246 : Module Matrix G click

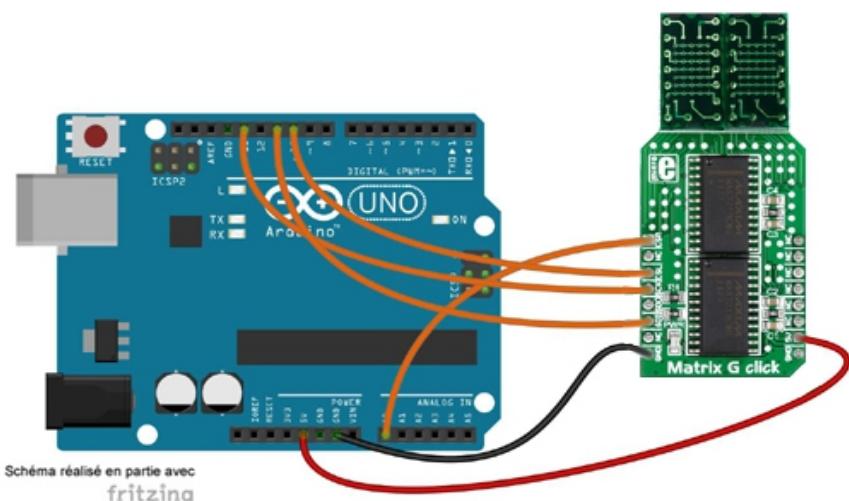


Des exemples de programmes permettant un pilotage via une liaison SPI à base de microcontrôleurs ARM™, AVR, dsPIC30/33 & PIC24, FT90x, PIC et PIC32 avec les compilateurs "C" (mikroC) sont disponibles afin de vous permettre une prise en main rapide et intuitive du module.

### Montage à réaliser



Enfichez le module MIKROE-2246 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2246 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Matrix G Click"  
*  
*****  
* Les nombres de 00 à 99 sont affichés sur le module  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Matrix G Click" (Réf.: MIKROE-2246) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque  
* 1. https://github.com/wayoda/LedControl  
*  
*****/  
  
// Affectation des broches  
#define CSL 10  
#define CSR 14  
#define CLK 13  
#define DIN 11  
  
#define NB_MATRICE 1  
  
#include "LedControl.h" // appel de la librairie  
LedControl matriceL=LedControl(DIN,CLK,CSL,NB_MATRICE); // création de l'objet matriceL  
LedControl matriceR=LedControl(DIN,CLK,CSR,NB_MATRICE); // création de l'objet matriceR  
  
// Définition des chiffres  
unsigned char chiffre [10] [7]={  
{0x1C,0x22,0x22,0x22,0x22,0x22,0x1C}, // 0  
{0x1C,0x08,0x08,0x08,0x08,0x18,0x08}, // 1  
{0x3E,0x20,0x10,0x08,0x04,0x22,0x1C}, // 2  
{0x1C,0x22,0x02,0x0C,0x02,0x22,0x1C}, // 3
```

```

{0x04,0x04,0x3E,0x24,0x14,0x0C,0x04}, // 4
{0x1C,0x22,0x02,0x02,0x3C,0x20,0x3E}, // 5
{0x1C,0x22,0x22,0x3C,0x20,0x22,0x1C}, // 6
{0x10,0x10,0x10,0x08,0x04,0x02,0x3E}, // 7
{0x1C,0x22,0x22,0x1C,0x22,0x22,0x1C}, // 8
{0x1C,0x22,0x02,0x1E,0x22,0x22,0x1C}, // 9
};

void setup()
{
matriceL.shutdown(0,false); // arrêt du mode économie du MAX7219
matriceL.setIntensity(0,10); // configuration de l'intensité lumineuse des led (0 à 15)
matriceL.clearDisplay(0); // extinction de toutes les led
matriceR.shutdown(0,false); // arrêt du mode économie du MAX7219
matriceR.setIntensity(0,10); // configuration de l'intensité lumineuse des led (0 à 15)
matriceR.clearDisplay(0); // extinction de toutes les led
}

void loop()
{
for(int j=0;j<10;j++) // défilement des dizaines
{
for(int i=0;i<7;i++) // affichage des chiffres sur la matrice de droite
{
matriceL.setRow(0,i,chiffre[j][i]);
delay(10);
}
for(int j=0;j<10;j++) // défilement des unités
{
for(int i=0;i<8;i++) // affichage des chiffres sur la matrice de gauche
{
matriceR.setRow(0,i,chiffre[j][i]);
delay(10);
}
delay(1000); // pause d'une seconde
}
}
}

```

## Application 34 : MIKROE-2272 : Module PWM Driver click

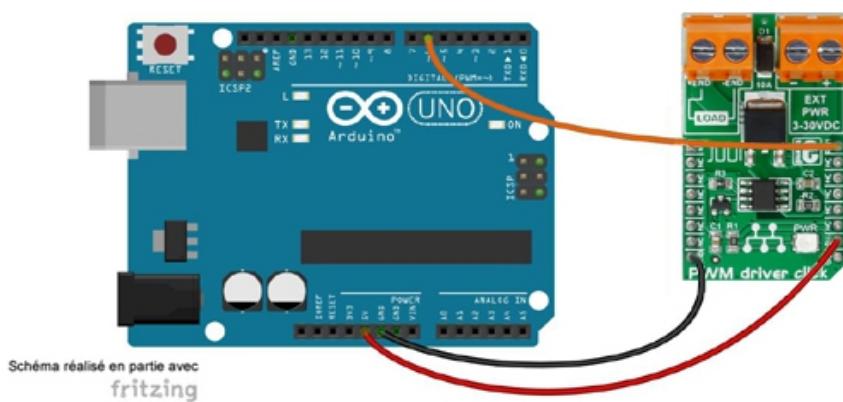


Equipé d'un Si8711CC et d'un MOSFET DMP3010LK3, ce module Click Board vous permettra de pouvoir piloter un moteur (jusqu'à 10 A en pointe) à l'aide de votre microcontrôleur via un signal PWM.

### Montage à réaliser



Enfichez le module MIKROE-2272 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2272 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

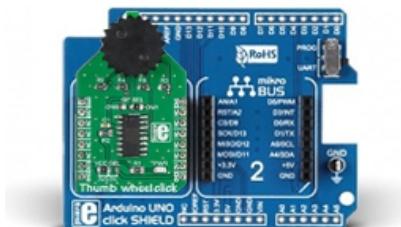
```
*****  
*  
* Test du module "PWM Driver Click"  
*  
*****  
* Le signal PWM généré par l'Arduino est transmis à la charge via le module  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "PWM Driver Click" (Réf.: MIKROE-2272) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches  
#define PWM 6  
int i;  
  
void setup()  
{  
pinMode(PWM, OUTPUT);  
}  
  
void loop()  
{  
for(i = 0;i<=255;i++) // variation du rapport cyclique toutes les 100 ms  
{  
analogWrite(PWM, i);  
delay(100);  
}  
delay(2000); // pause de 2 secondes lorsque le rapport cyclique est à 100 %  
analogWrite(PWM,0); // rapport cyclique à 0%  
delay(2000); // pause de 2 secondes  
}
```

## Application 35 : MIKROE-2366 : Module Thumbwheel click

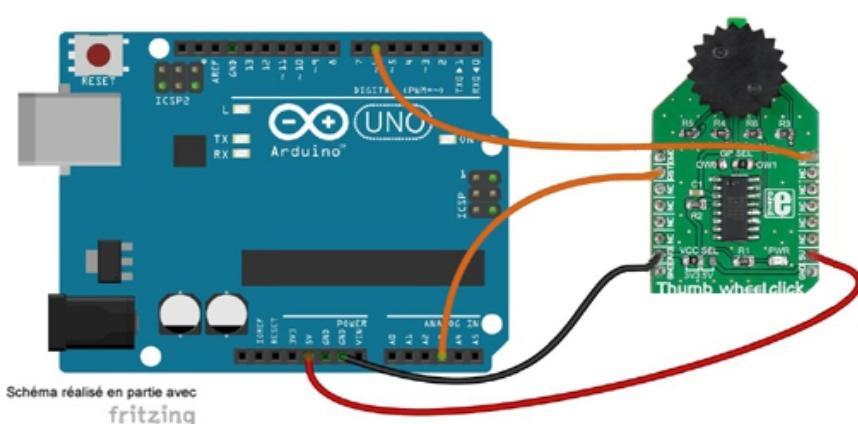


Ce module Click Board est équipé d'un commutateur rotatif 10 positions (connecté à un switch DS2408 1-Wire 8 canaux adressable). La liaison s'effectue via un bus 1-Wire.

### Montage à réaliser



Enfichez le module MIKROE-2366 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2366 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "Thumbwheel Click"  
*  
*****  
* La position du commutateur est affiché dans le moniteur série  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "Thumbwheel Click" (Réf.: MIKROE-2366) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
* Bibliothèque  
* 1. https://github.com/PaulStoffregen/OneWire  
*  
* Remarque : Le code obtenu pour les positions est le suivant  
* Position 1 -> Code 0x0A  
* Position 2 -> Code 0x08  
* Position 3 -> Code 0x0A  
* Position 4 -> Code 0x04  
* Position 5 -> Code 0x06  
* Position 6 -> Code 0x00  
* Position 7 -> Code 0x02  
* Position 8 -> Code 0x00  
* Position 9 -> Code 0x02  
* Position 10 -> Code 0x08  
*  
*****/  
  
#include <OneWire.h> // appel de la bibliothèque  
  
// Affectation des broches  
#define OW1 6  
#define RST 17
```

```
byte adresse[8];
byte code;

OneWire ds(OW1); // création de l'objet ds

void setup(void)
{
Serial.begin(9600); // initialisation de la liaison série
pinMode(RST,OUTPUT); // configuration de la broche RST en sortie
digitalWrite(RST,HIGH);
Serial.println("Recherche composant");

// Recherche un module 1-Wire sur le bus
if (!ds.search(adresse))
{
Serial.println("Fin de recherche");
ds.reset_search();
}

// Module 1-Wire trouvé
Serial.print("Composant trouvé :");
for(byte i = 0; i < 8; ++i)
{
if (adresse[i] < 0x10)
{
Serial.write('0');
}
Serial.print(adresse[i], HEX);
Serial.write(' ');
}
Serial.println();
if (adresse[0] == 0x29)
{
Serial.println("Le composant est un DS2408");
}

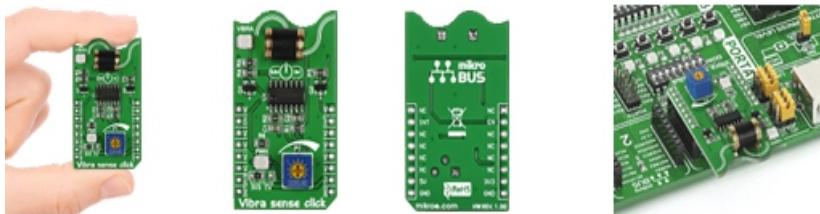
// Vérification de l'adresse
if (OneWire::crc8(adresse, 7) != adresse[7])
{
Serial.print("CRC invalide");
```

```
}

Serial.println();
delay(1000);
}

void loop(void)
{
// Acquisition du code
ds.reset();
ds.select(adresse);
ds.write(0xF0);
ds.write(0x88);
ds.write(0x00);
code = ds.read();
Serial.print("Code:");
Serial.println(code,HEX);
switch (code)
{
case 0xA:
Serial.println("La position est 1 ou 3");
break;
case 0x8:
Serial.println("La position est 2 ou 10");
break;
case 0x4:
Serial.println("La position est 4");
break;
case 0x6:
Serial.println("La position est 5");
break;
case 0x0:
Serial.println("La position est 6 ou 8");
break;
case 0x2:
Serial.println("La position est 7 ou 9");
break;
default:
break;
}
delay(1000);
}
```

## Application 36 : MIKROE-1927 : Module Vibra Sens click

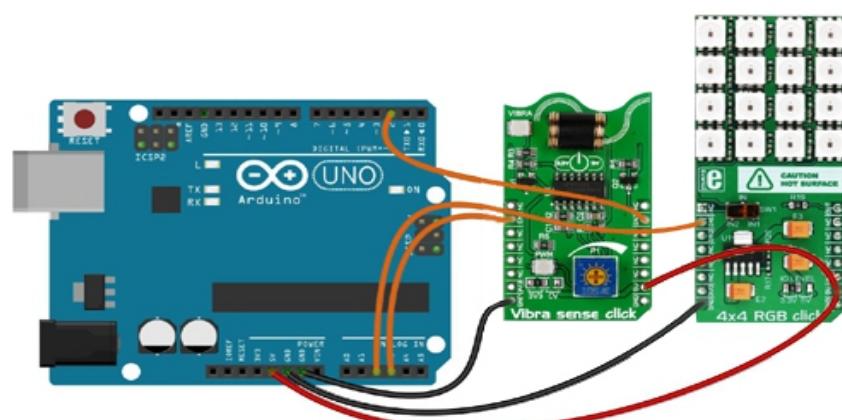


Ce module est équipé d'un capteur de choc/vibration avec une sortie numérique. Sensibilité réglable. Le module est prévu pour être alimenté en 3,3 V ou 5 V.

### Montage à réaliser



Enfichez le module MIKROE-1927 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-1927 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



**Attention, il conviendra de ne pas faire fonctionner les leds au maximum de leur intensité lumineuse afin d'éviter une consommation excessive pouvant entraîner l'endommagement de votre microcontrôleur ainsi qu'une dissipation de température trop importante.**

**Du fait de la haute luminosité des Leds (surtout pour les couleurs "tirant" vers le blanc), il est IMPERATIF de ne JAMAIS regarder directement la source lumineuse sous risque de graves séquelles oculaires.**

### Programme Arduino

```
*****
*
* Test du module "Vibra sense Click"
*
*****
* Une vibration déclenche 5 flashes rouges sur le module 4x4 RGB Click
*
* Matériel
* 1 Arduino Uno (A000066)
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)
* 1 Module "Vibra sense Click" (Réf.: MIKROE-1927) inséré sur le support N°1 du shield
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de
* de la platine shield MIKROE-1581)
* 1 Module "4x4 RGB Click" inséré sur le support N°2 du shield
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)
*
* Bibliothèque
* 1. https://github.com/adafruit/Adafruit_NeoPixel
*
*****/
```

```
// Affectation des broches
#define validation 17 // validation du module Vibra sense
#define sortie 16 // DIN du module 4x4 RGB
#define nb_led 16 // le module 4x4 RGB comporte 16 led

#include <Adafruit_NeoPixel.h> // appel de la bibliothèque
```

```
Adafruit_NeoPixel module = Adafruit_NeoPixel(nb_led, sortie, NEO_GRB + NEO_KHZ800); // création de l'objet module

void setup()
{
pinMode(validation,OUTPUT);
digitalWrite(validation,HIGH);
module.begin(); // initialisation de module
attachInterrupt(0, alarme, FALLING); // attache l'interruption externe n°0 à la fonction alarme
}

void loop()
{
}

void alarme()
{
for (int i=0; i<5; i++)
{
for(int t=0; t< 16; t++) // allumage successif des 16 led
{
module.setPixelColor(t, 128, 0, 0);
}
module.show(); // rafraichissement des led
delay(300);
for(int t=0; t< 16; t++) // extinction successif des 16 led
{
module.setPixelColor(t, 0, 0, 0);
}
module.show(); // rafraichissement des led
delay(300);
}
}
```

## Application 37 : MIKROE-2047 : Module DC Motor 3 click

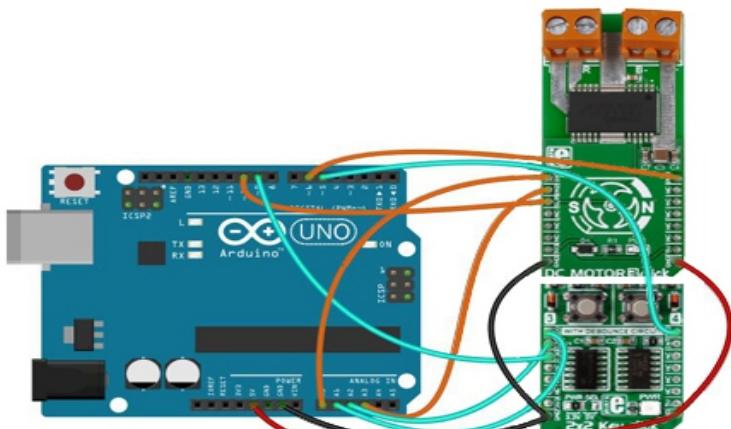


Ce petit module "click Board" intègre un contrôleur de type TB6549FG capable de commander un moteur cc de 30 V max. / 3,5 A max.

### Montage à réaliser



Enfichez le module MIKROE-2047 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2047 (si vous ne disposez pas de la platine d'interface MIKROE-1581).



## Programme Arduino

```
*****  
*  
* Test du module "DC Motor 3 Click"  
*  
*****  
* La commande du moteur est réalisée à l'aide de 4 boutons poussoirs  
* du module 2x2 Key Click  
*  
* Matériel  
* 1 Arduino Uno (A000066)  
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)  
* 1 Module "DC Motor 3 Click" (Réf.: MIKROE-2047) inséré sur le support N°1 du shield  
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de  
* de la platine shield MIKROE-1581)  
* 1 Module "2x2 Key Click" inséré sur le support N°2 du shield  
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)  
*  
*****/  
  
// Affectation des broches module DC Motor Click 3  
#define IN1 14  
#define IN2 17  
#define SLP 10  
#define PWM 6  
  
// Affectation des broches module 2x2 Key Click  
#define BP1 15  
#define BP2 16  
#define BP3 9  
#define BP4 5  
  
int vitesse=0;  
void setup()  
{  
Serial.begin(9600); // initialisation du moniteur série  
pinMode(IN1,OUTPUT); // configuration des broches  
pinMode(IN2,OUTPUT);  
pinMode(SLP,OUTPUT);
```

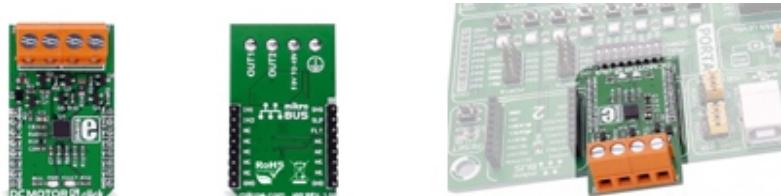
```
pinMode(PWM,OUTPUT);
pinMode(BP1,INPUT);
pinMode(BP2,INPUT);
pinMode(BP3,INPUT);
pinMode(BP4,INPUT);
digitalWrite(SLP,HIGH); // validation du TB6549FG
}

void loop()
{
if (digitalRead(BP1)==HIGH) // si BP1 est actif le moteur tourne dans le sens horaire
{
Serial.println("Le moteur tourne dans le sens horaire");
digitalWrite(IN1,LOW);
digitalWrite(IN2,HIGH);
}
else // sinon le moteur tourne dans le sens anti-horaire
{
Serial.println("Le moteur tourne dans le sens anti-horaire");
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
}
if (digitalRead(BP2)==HIGH) // si BP2 est actif on augmente la vitesse
{
vitesse=vitesse+10;
if (vitesse>255)
{
vitesse=250;
}
}
if (digitalRead(BP4)==HIGH) // si BP4 est actif on diminue la vitesse
{
vitesse=vitesse-10;
if (vitesse<0)
{
vitesse=0;
}
}
if (digitalRead(BP3)==HIGH) // si BP3 est actif on arrête le moteur
{
digitalWrite(IN1,HIGH);
```

```
digitalWrite(IN2,HIGH);
vitesse=0;
}
analogWrite(PWM,vitesse); // génération du signal PWM
Serial.print("Vitesse : "); // affichage de la vitesse
Serial.println(vitesse);
delay(600); // pause pour les rebonds
}
```

---

## Application 38 : MIKROE-2699 : Module DC Motor 5 click



Ce "click Board" intègre un contrôleur de moteur "cc" DRV8701 doté de nombreuses protections: verrouillage sous tension, défauts de la pompe de charge, arrêt en surintensité, protection contre les courts-circuits, défaut en cas de surchauffe...

### Montage à réaliser



Enfichez le module MIKROE-2699 sur la platine MIKROE-1581. Puis enfichez le tout sur la platine arduino™ (A000066) ou réalisez les connexions ci-dessous entre la platine Arduino™ (A000066) et le module MIKROE-2699 (si vous ne disposez pas de la platine d'interface MIKROE-1581).

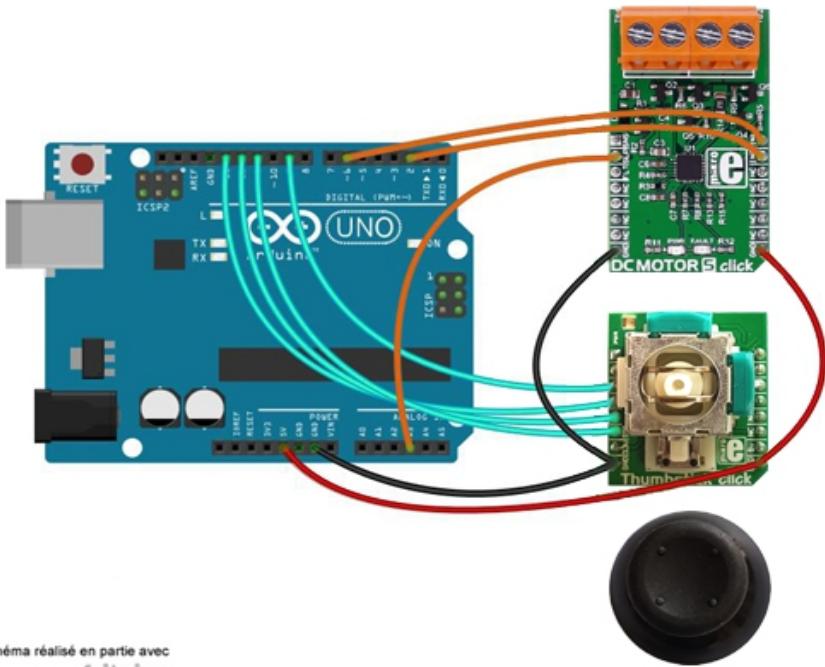


Schéma réalisé en partie avec  
fritzing

## Programme Arduino

```
*****
/*
* Test du module "DC Motor 5 Click"
*
*****
* La vitesse et le sens de rotation du moteur sont proportionnels
* à la position de l'axe y du module Thumbstick Click
*
* Matériel
* 1 Arduino Uno (A000066)
* 1 Shield "Click" pour arduino UNO (Réf. : MIKROE-1581)
* 1 Module "DC Motor 5 Click" (Réf.: MIKROE-2699) inséré sur le support N°1 du shield
* ou divers straps mâles/femelles (réf.: PRT-12794) si vous ne disposez pas de
* de la platine shield MIKROE-1581)
* 1 Module "Thumbstick Click" inséré sur le support N°2 du shield
* Schéma publié sous licence CC Attribution-ShareALike (Arduino et ses connexions réalisés avec Fritzing)
*
```

```
******/
```

```
#include "MCP320X.h" // appel de la bibliothèque

// Affectation des broches du module DC Motor 5 Click
#define IN1 6
#define IN2 2
#define SLEEP 17

// Affectation des broches du module Thumbstick Click
#define CS 9
#define SCK 13
#define SDI 11
#define SDO 12

#define MCP3204 4 // référence du MPC320X

MCP320X joystick(MCP3204, SCK, SDI, SDO, CS); // création de l'objet joystick

int y;
int vitesse;

void setup()
{
Serial.begin(9600); // initialisation de la liaison série
pinMode(IN1,OUTPUT); // configuration des broches
pinMode(IN2,OUTPUT);
pinMode(SLEEP,OUTPUT);
pinMode(SNSO,INPUT);
digitalWrite(SLEEP,HIGH); // validation du DRV8701P
}

void loop()
{
y=joystick.readADC(0); // acquisition de la valeur y
Serial.print("Valeur y du joystick:");
Serial.println(y);
if (y>2050)
{
vitesse=map(y,2050,4092,0,255);
digitalWrite(IN2,LOW); // sens horaire
analogWrite(IN1, vitesse);
}
}
```

```
if (y<1950)
{
vitesse=map(y,1950,0,255,0);
digitalWrite(IN2,HIGH); // sens anti-horaire
analogWrite(IN1, vitesse);
}
if ((y<2050)&&(y>1950))
{
digitalWrite(IN1,HIGH); // arrêt du moteur
digitalWrite(IN2,HIGH);
}
}
```

## **Conditions d'utilisations et limite de responsabilité**

Les notes d'applications de ce document ont été conçues avec la plus grande attention. Tous les efforts ont été mis en œuvre pour éviter les anomalies. Toutefois, nous ne pouvons garantir que ces notes d'applications soit à 100% exempt de toute erreur. Les informations présentes dans cette documentation sont données à titre indicatif. Il est important de toujours considérer les programmes sources présents dans ce document comme des programmes en version Béta. Lextronic ne pourra en aucun cas être tenu responsable de dommages quels qu'ils soient (intégrant, mais sans limitation, les dommages pour perte de bénéfice commercial, interruption d'exploitation commerciale, perte d'informations et de données à caractère commercial ou de toute autre perte financière) provenant de l'utilisation ou de l'incapacité à pouvoir utiliser les notes d'applications de ce document, même si Lextronic a été informé de la possibilité de tels dommages. Ces notes d'applications sont exclusivement conçues dans un but pédagogique (essentiellement pour l'apprentissage de la programmation). Nous ne pouvons donner aucune garantie de leur fonctionnement pour une utilisation au sein de vos propres applications, ni pour une utilisation de ces dernières au sein d'applications à caractère professionnel . De manière général, ces notes d'applications ne sont pas conçues, ni destinées, ni autorisées pour expérimenter, développer ou être intégrées au sein d'applications dans lesquelles une défaillance de celles-ci pourrait créer une situation dangereuse pouvant entraîner des pertes financières, des dégâts matériel, des blessures corporelles ou la mort de personnes ou d'animaux. Si vous utilisez ces notes d'applications volontairement ou involontairement pour de telles applications non autorisées, vous vous engagez à soustraire Lextronic de toute responsabilité et de toute demande de dédommagement.

## **Schémas de raccordement**

Certaines parties de schémas de raccordement de cette documentation (la platine arduino et les fils de connexions) ont été réalisés à l'aide du logiciel Fritzing.

<http://fritzing.org/home/> Ces schémas sont distribués sous licence CC Attribution-ShareALike

La représentation des modules click board a été redessiné par dessus ces schémas.

## **Librairies additionnelles**

Certains code sources font appel à des librairies externes (qu'il vous faudra télécharger). Une recherche sur Internet vous permettra de trouver aisément ces librairies. Certaines de ces librairies existent sous différentes versions. En cas de non fonctionnement d'un programme, pensez à tester à nouveau ce dernier avec une version de librairie différente. Ces librairies sont distribuées selon divers types de licence. Merci de prendre connaissance de ces licences avant leur utilisation.

**Copyright et appellations commerciales**

Toutes les marques, les procédés, les références et les appellations commerciales des produits cités dans cette documentation appartiennent à leur propriétaire et Fabricant respectif.

Les codes sources et les schémas de ce document sont téléchargeables ici :

[https://www.lextronic.fr/~lextronic\\_doc/Click\\_APP.zip](https://www.lextronic.fr/~lextronic_doc/Click_APP.zip)