

## DgCod16

### 16 analogue values and a push button



Ce circuit est très pratique pour sélectionner 16 valeurs dans une table, choisir 16 programmes au démarrage, etc. Il n'utilise qu'une seule entrée analogiques (A0-A5) de la carte Arduino. A chaque position correspond une tension. Le schéma R/2R usuel d'un convertisseur D/A ne peut pas être utiliser, la correspondance n'est pas linéaire et la fonction Arduino Map () n'est pas utilisable. Une table établie par calibrage est très efficace et la fonction de lecture de la position GetCod16 () est donnée plus loin.

Le poussoir permet de quitter la position choisie, et démarrer par exemple l'un de vos 16 tests préférés (adapter [www.didel.com/diduino/DemosMultiples.pdf](http://www.didel.com/diduino/DemosMultiples.pdf)).

Le circuit remplace aussi un potentiomètre, 16 valeurs analogiques sont souvent bien suffisantes pour commander un moteur ou une intensité lumineuse, avec l'avantage que l'on retrouve exactement une valeur sélectionnée auparavant.

Connectez le module sur un entrée A0 .. A5. Comme premier test, affichez la valeur analogique.

Vous ne savez pas faire? Regardez [www.didel.com/coursera/LC2.pdf](http://www.didel.com/coursera/LC2.pdf)

//Cod16\_0.ino| Lit et affiche la valeur analogique

```
void setup() {
  Serial.begin(9600);
}

void loop () {
  Serial.println (analogRead (A1)) ;
}
```

S'il n'y a pas eu erreur dans une résistance, les valeurs lues sont proches de ce qui a été mesuré en usine:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	63	119	171	221	261	298	331	364	391	416	439	463	483	501	520

Ces valeurs sont mises dans une table, en ajoutant une valeur supplémentaire pour une position 16 théorique qui facilite l'algorithme.:

La variable pos est initialisée à zéro. On lit la tension. Si elle est inférieure à 63/2, on a pos=0. Si non, pos est incrémenté et on regarde si la tension est inférieure à (119+63)/2. Si oui, pos=1. Si non, on continue dans une boucle while.

Le programme qui teste la fonction GetCod16 () est le suivant :

```
//Cod16_1.ino| Lit et affiche la position 0..15
void setup() {
  Serial.begin(9600);
}
// fonction GetCod16 Appel par: position = Getcod16();
const int
TaCod16[]={0,63,119,171,221,261,298,331,364,391,416,439,463,483,501,520,550};
byte GetCod16 () {
  byte pos=0;
  while ( analogRead(A1)>((TaCod16[pos]+TaCod16[pos+1])/2) ) {
    pos++;
  }
  return pos;
}
// Test - la position est affichée chaque seconde
void loop() {
  Serial.println (GetCod16());
  delay (1000);
}
```



This circuit is useful to select one out of 16 positions in a table, select delo programs at power-up, It requires only one analogue Areduino input (A0-A5). A different voltage is given for every position. Correspondance is not linear and Arduino Map() function cannot be used. Our decode function GetCod16 () takes less code and is faster.

The push button can be useful to start the program you have selected on the rotary switch. The modules is also more convenient than a pot in many cases: you can repeatedly get the same voltage on the same position.

You can check the voltage values with the usual program.

```
//Cod16_0.ino| Display voltage
void setup() {
  Serial.begin(9600);
}
void loop () {
  Serial.println (analogRead (A1)) ;
}
```

Values should be close from the following. +/- 2 to 5 difference are acceptable.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	63	119	171	221	261	298	331	364	391	416	439	463	483	501	520

We add an hypotherical 16th position and the algorithm is quite simple:

```
//Cod16_1.ino| Display position number
void setup() {
  Serial.begin(9600);
}
// fonction GetCod16; position = Getcod16();
const int TaCod16[]={0,63,119,171,221,261,298,331,364,391,416,439,463,483,501,520,550};
byte GetCod16 () {
  byte pos=0;
  while ( analogRead(A1)>((TaCod16[pos]+TaCod16[pos+1])/2) ) {
    pos++;
  }
  return pos;
}
// Test - position is displayed every second
void loop() {
  Serial.println (GetCod16());
  delay (1000);
}
```

