

Arduino – Calculateur de profondeur de champ pour la photographie

Les composants nécessaires pour réaliser ce calculateur sont les suivants :

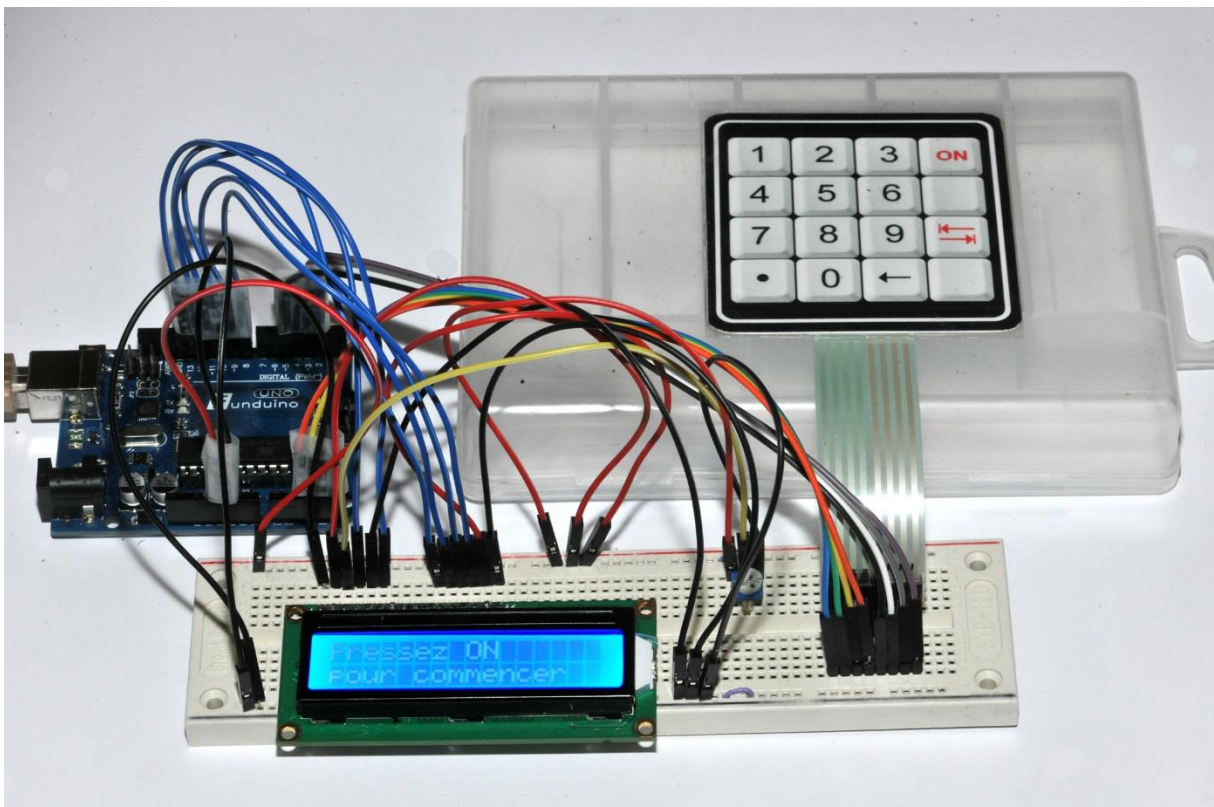
- 1 carte Arduino UNO
- 1 écran LCD 2 x 16 caractères (QC1602a)
- 1 keypad matrix 4x4
- 1 potentiomètre 10 K
- 23 câbles

Arduino – Photography DOF Calculator

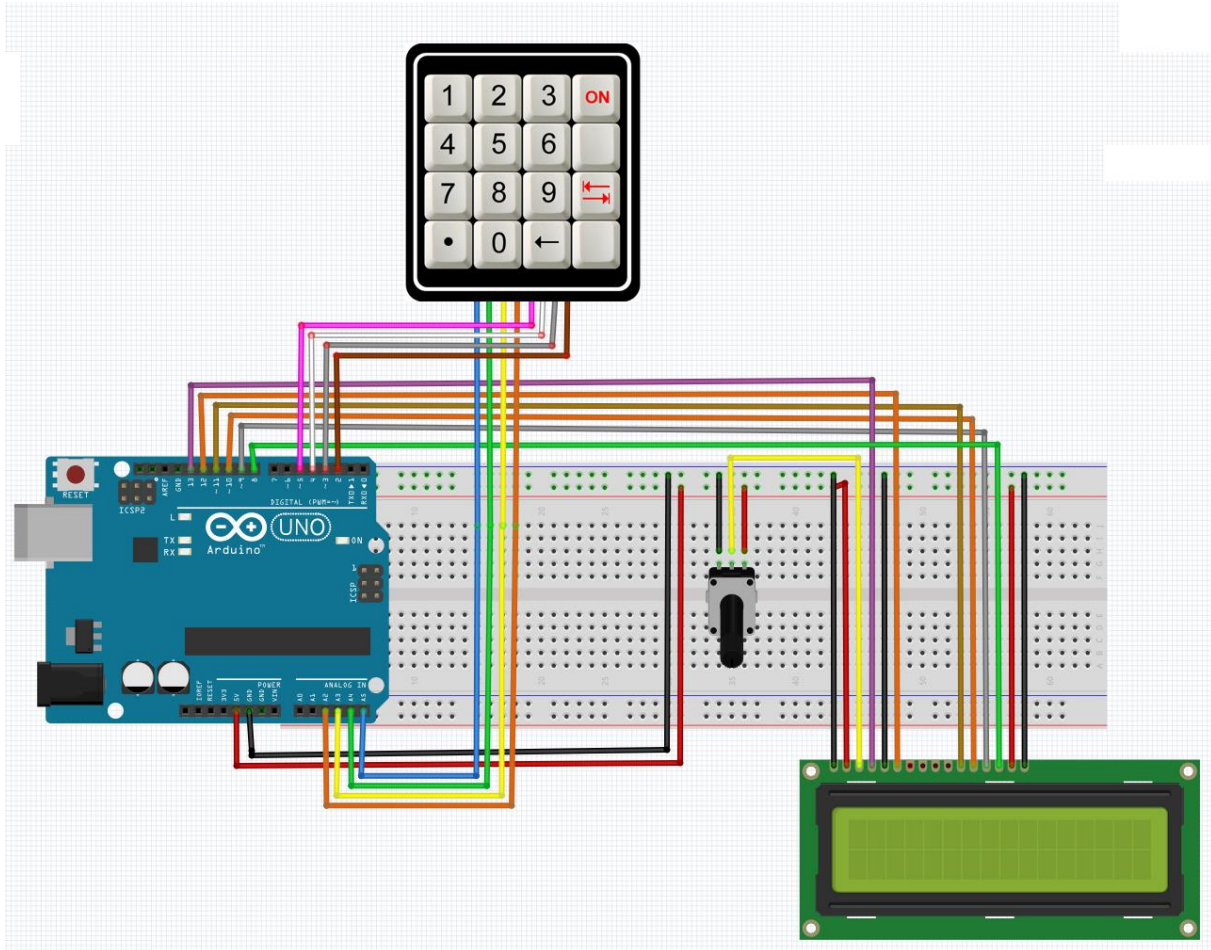
To create a DOF calculator, you need following components :

- 1 Arduino UNO board
- 1 LCD screen (2 x 16 characters (QC1602a)
- 1 keypad (matrix 4x4)
- 1 potentiometer (10 k)
- 23 wires

1 – VUE GLOBALE – GLOBAL VIEW



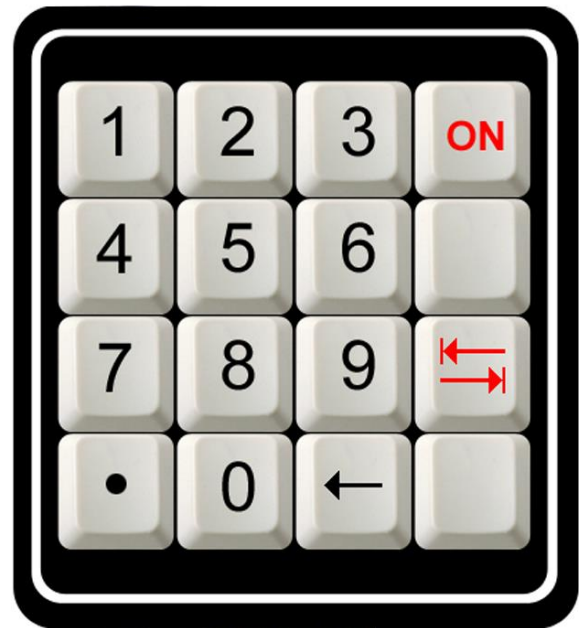
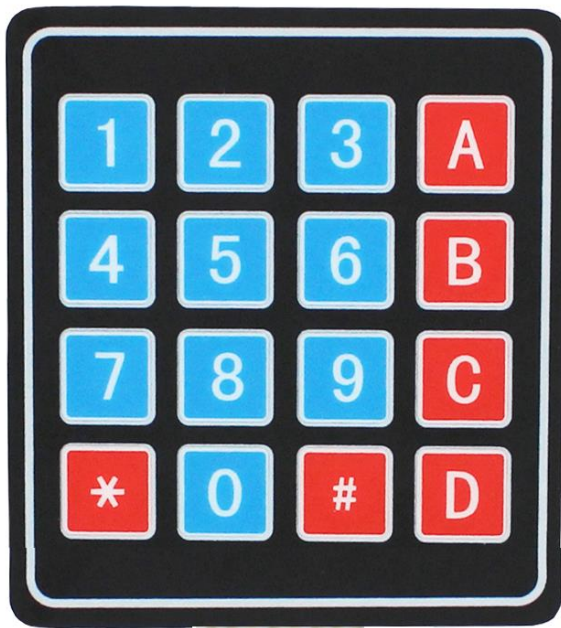
2 – PROJET FRITZING – FRITZING PROJECT



3 – MODIFICATION DU KEYPAD – HOW TO MODIFY A KEYPAD

On décolle délicatement la membrane d'origine (ci-dessous vue de gauche) et on la remplace avec une membrane de la même dimension que l'on a imprimée sur du papier photo qu'on aura plastifié pour en assurer la censure.

Unstick delicately the membrane of the original keypad and replace it by a photo of the same size created and printed with any software (like Photoshop). It is highly recommended to plasticize it for a better conservation.



Les points importants sont les suivants : on remplace l'étoile par un point, car toutes les entrées et les calculs seront des nombres à virgule flottante. La touche « ON » permet de lancer le programme ; la touche de tabulation rouge d'entrer les 4 valeurs dont le calculateur a besoin. Lorsqu'on presse cette touche après avoir entré le 4^e paramètre, le programme se lance et les résultats s'affichent automatiquement. La touche comportant une flèche noire permet d'annuler la dernière touche pressée (ex : si on a entré la valeur 2000 au lieu de 200, il suffit d'appuyer une seule fois sur cette touche pour corriger l'erreur).

Most important points are the following ones : we replace the star by a point, because entries and calculations will be numbers with floating decimal point. The key "ON" allows to launch the program ; the red tab key to enter the 4 parameters which the DOF calculator needs. When we press this touch having entered the 4th parameter, the results of the calculation display automatically. The backspace key (black arrow) is used to delete any character before the current position of the cursor (ex: if we entered the value 2000 instead of 200, press this key once to correct the error).

<p>Les données en entrée sont :</p> <ul style="list-style-type: none"> - Valeur de la focale de l'objectif en mm - Valeur de l'ouverture (1, 1.4, 2,2.8,4, etc.) - Distance du sujet (mètres) - Valeur du cercle de confusion en microns (30.0 pour le 24x36 par exemple) <p>Les résultats en sortie sont tous en m :</p> <ul style="list-style-type: none"> - Valeur de l'hyperfocale - Limite avant - Limite arrière - Valeur de la profondeur de champ - Marge avant - Marge arrière 	<p>Input parameters :</p> <ul style="list-style-type: none"> - <i>Lens focal length</i> - <i>Aperture value ((1, 1.4, 2,2.8,4, etc.)</i> - <i>Subject distance (meters)</i> - <i>Circle of confusion (microns – 30.0 for a full frame sensor)</i> <p>Output results :</p> <ul style="list-style-type: none"> - <i>Hyperfocal distance</i> - <i>Near limit</i> - <i>Far limit</i> - <i>Depth of field</i> - <i>In front of subject</i> - <i>Behind subject</i>
---	---

4 – LE PROGRAMME ARDUINO – ARDUINO PROGRAM

```
// Calculateur de profondeur de champ pour la photographie
// Michel Rohan (Annecy) - février 2015 - microh74@yahoo.fr
// Domaine public - projet complet au format PDF en http://www.planete-bleue.net/pdc\_calculateur.pdf
```

```
#include <LiquidCrystal.h>
#include <Keypad.h>
```

```
const byte LIGNES = 4; // 4 lignes
const byte COLONNES = 4; // 4 colonnes
```

```
// Constantes du Keypad
const int C4=2; //broche numérique keypad en sortie
const int C3=3; //broche numérique keypad en sortie
const int C2=4; //broche numérique keypad en sortie
const int C1=5; //broche numérique keypad en sortie
const int L4=16; //broche analogique keypad en entrée
const int L3=17; //broche analogique keypad en entrée
const int L2=18; //broche analogique keypad en entrée
const int L1=19; //broche analogique keypad en entrée
```

```
// Constantes de l'écran LCD
const int RS=13; //broche numérique LCD
const int E=12; //broche numérique LCD
const int D4=11; //broche numérique LCD
const int D5=10; //broche numérique LCD
const int D6=9; //broche numérique LCD
const int D7=8; //broche numérique LCD
```

```

//--- Définition des touches - variables globales
char touches[LIGNES][COLONNES] = {
  // V signifie Vide (touche non utilisée)
  // # signifie effacer (remise à zéro de la variable)
  // > signifie variable suivante
  // < signifie Entrée (on lance le calcul)
  {'1','2','3','E'},
  {'4','5','6','V'},
  {'7','8','9','>'},
  {'.','0','R','V'}
};

// tableaux de lignes et colonnes
byte BrochesLignes[LIGNES] = {L1, L2, L3, L4}; //connexions utilisées pour les broches de lignes du
clavier
byte BrochesColonnes[COLONNES] = {C1, C2, C3, C4}; //connexions utilisées pour les broches de
colonnes du clavier

char touche; // variable de stockage valeur touche appuyée

int menu=0;

// Initialisation des variables en entrée
float foc = 0;
float dist = 0;
float ouv = 0;
float cdc = 0;
String item[16]={"","Focale (mm) ?","Distance (m) ?","Ouverture (F/) ?","CdC (microns) ?"};
String inputData = "";

// Déclaration des variables sortie
float hf; // hyperfocale en m
float plan1; // premier plan en m
float plan2; // arrière plan
float pdc; // profondeur de champ en m
float av; // devant en m
float ar; // arrière en m

// Variables nécessaire au calcul
float r1; // pourcentage avant
float r2; // pourcentage arrière

// Variable nécessaire pour la conversion des réels en String
static char outstr[15];

// Interface de la bibliothèque avec les pins utilisés (voir le schéma Fritzing)
LiquidCrystal lcd(RS, E, D4, D5, D6, D7); // Création d'un objet LiquidCrystal en mode 4 bits
Keypad clavier = Keypad( makeKeymap(touches), BrochesLignes, BrochesColonnes, LIGNES,
COLONNES );

void setup() {
  // 16 caractères 2 lignes pour un écran QC1602a

```

```

lcd.begin(16, 2);
lcd.clear();
delay(10);
// On fixe le débit série à 9600 bauds
Serial.begin(9600);
lcd.setCursor(0,0);
lcd.print ("Pressez ON");
lcd.setCursor(0,1);
lcd.print ("pour commencer");

}

// Boucle principale
void loop() {
// On détecte les valeurs fournies par le keypad
touche = clavier.getKey(); // lecture de la touche appuyée
if (touche != NO_KEY){

// si la touche tabulation est pressée on passe à l'entrée suivante
if (touche=='>' || touche=='E'){
  lcd.clear();
  if (menu==1) foc=inputData.toFloat();
  if (menu==2) dist=inputData.toFloat();
  if (menu==3) ouv=inputData.toFloat();
  if (menu==4) {
    lcd.clear();
    cdc=inputData.toFloat();
    resultats();
  }
  inputData = "";
  menu+=1;
  if (menu==5) menu = 1;
  lcd.setCursor(0,0);
  lcd.print(item[menu]);
  lcd.setCursor(0,1);
  lcd.blink();
  delay(300); // temporisation entre 2 appuis
}

// Si on presse la touche de correction (<-) on annule le dernier caractère frappé
if (touche=='R'){
  if (inputData.length(>0){
    inputData.remove(inputData.length()-1,1);
    lcd.setCursor(0,1);
    lcd.print(inputData);
    lcd.print(" ");
    lcd.setCursor(inputData.length(),1);
  }
  delay(300); // pause entre 2 appuis
}

// Si on presse une touche inutilisée il ne se passe rien

```

```

if (touche=='V'){
    delay(300); // pause entre 2 appuis
}

// Si on presse la touche "." ou une touche >=0 ou <=9 on affiche la valeur
if ((touche>='0' && touche<='9') || touche=='.'){
    inputData+=touche;
    lcd.print(touche); // sinon on affiche la touche
    delay(300); // pause entre 2 appuis
}
}
}

// Calcul de l'hyperfocale
float hyperFocale(float foc, float ouv, float cdc){
    hf = decim2(foc * foc / ouv / cdc);
    return hf;
}

// Distance du premier plan net
float premierPlan(float dist, float hf, float foc){
    plan1 = decim2(dist * (hf - foc / 1000) / (hf + dist - 2 * foc / 1000));
    return plan1;
}

// Distance du dernier plan net
float arrierePlan(float dist, float hf, float foc){
    plan2 = decim2(dist * (hf - foc / 1000) / (hf - dist));
    if (plan2 < 0) plan2 = 0;
    return plan2;
}

// Calcul de la profondeur de champ en mètres
float dof(float plan2, float plan1){
    pdc = plan2-plan1;
    if (plan2 == 0) pdc = 0;
    return pdc;
}

// Calcul de la marge avant
float margeAvant(float dist, float plan1){
    r1 = decim2(dist - plan1);
    return r1;
}

// Calcul de la marge arrière
float margeArriere(float pdc, float dist, float plan1, float plan2){
    r2 = 0;
    if (plan2 != 0) r2 = decim2(pdc * (plan2 - dist) / (plan2 - plan1));
    return r2;
}

```

```

// Arrondi à 2 décimales
float decim2(float nombre){
    nombre = (floor(nombre * 100)) / 100;
    return nombre;
}

// Affichage des résultats
void resultats(){
    lcd.print("Resultats *****");
    delay(1000);
    String string1="";

    // Hyperfocale
    lcd.clear();
    string1 = "Hyperfocale =";
    lcd.setCursor(0,0);
    lcd.print(string1);
    string1 = dtostrf(hyperFocale(foc,ouv,cdc),7,2,outstr);
    string1 += " m";
    lcd.setCursor(0,1);
    lcd.print(string1);
    delay(3000);

    // Premier plan
    lcd.clear();
    string1 = "Premier plan =";
    lcd.setCursor(0,0);
    lcd.print(string1);
    string1 = dtostrf(premierPlan(dist,hf,foc),7,2,outstr);
    string1 += " m";
    lcd.setCursor(0,1);
    lcd.print(string1);
    delay(3000);

    // Arrière plan
    lcd.clear();
    string1 = "Arriere plan =";
    lcd.setCursor(0,0);
    lcd.print(string1);
    if (arrierePlan(dist,hf,foc)!=0){
        string1 = dtostrf(arrierePlan(dist,hf,foc),7,2,outstr);
        string1 += " m";
    }
    else{
        string1 += "INFINI";
    }
    lcd.setCursor(0,1);
    lcd.print(string1);
    delay(3000);

    // Profondeur de champ
    lcd.clear();

```



```

string1 = "Prof. de champ =";
lcd.setCursor(0,0);
lcd.print(string1);
if (arrierePlan(dist,hf,foc)!=0){
  string1 = dtostrf(dof(plan2,plan1),7,2,outstr);
  string1+=" m";
}
else{
  string1 += "INFINI";
}
lcd.setCursor(0,1);
lcd.print(string1);
delay(3000);

// Marge avant
lcd.clear();
string1 = "Marge avant =";
lcd.setCursor(0,0);
lcd.print(string1);
string1 = dtostrf(margeAvant(dist,plan1),7,2,outstr);
string1 += " m";
lcd.setCursor(0,1);
lcd.print(string1);
delay(3000);

// Marge arriere
lcd.clear();
string1 = "Marge arriere =";
lcd.setCursor(0,0);
lcd.print(string1);
if (arrierePlan(dist,hf,foc)!=0){
  string1 = dtostrf(margeArriere(pdc,dist,plan1,plan2),7,2,outstr);
  string1+=" m";
}
else{
  string1 += "INFINI";
}
lcd.setCursor(0,1);
lcd.print(string1);
delay(3000);

// Fin
lcd.clear();
lcd.print("Fin !");
delay(3000);
lcd.clear();
menu = 0;
}

```

5 – REALISATION - REALIZATION

Sur le prototype ci-dessous, il ne s'agit pas de touches réelles, mais d'une photographie d'un clavier numérique 4 x 4 plastifiée.

On the prototype below, there is not any real key. It is the plasticized photography of a numeric keypad 4 x 4.



5 – EXEMPLE DE TRAITEMENT

Pressez « ON » pour commencer

Focale (mm) ?	entrer 170	presser la touche de tabulation
Distance (m) ?	100	idem
Ouverture (F/) ?	5.6	idem
CdC (microns) ?	30	idem

Resultats *****

Hyperfocale	172.02 m
Premier plan	63.25 m
Arrière plan	238.61 m
Prof. De champ	175.36 m
Marge avant	36.75 m
Marge arrière	138.61 m
Fin !	