

FLOOR PROPERTY

Documentation de fabrication

FLOOR PROPERTY : Ce projet consiste à concevoir et réaliser un dispositif électronique embarqué capable de mesurer l'humidité du sol et également d'évaluer son niveau de concentration en nutriments NPK(Azote-Phosphore-Potassium). L'atelier de fabrication du prototype FLOOR PROPERTY vise non seulement à vulgariser les techniques agricoles modernes, mais aussi à stimuler l'exploitation raisonnée et optimale des terres agricoles en Afrique et donc finalement de contribuer à la culture du respect de l'environnement dans la pratique agricole.



FLOOR PROPERTY








FLOOR PROPERTY est un dispositif qui permet à tout usager de mesurer rapidement et facilement quelques paramètres caractéristiques d'un sol. Au-delà d'être didactique pour les amateurs d'électronique et d'agriculture, c'est aussi un dispositif qui pourrait s'avérer très utile pour les amateurs et praticiens agriculteurs.

1. Matériaux et outillage nécessaire

Pour réaliser le prototype FLOOR PROPERTY, nous aurons besoins d'un certain nombre de d'outils et de matériel.

1.1. Outillage

Pour fabriquer le boîtier de conditionnement, nous allons utiliser du contre-plaqué 4mm en bois pour fabriquer le boîtier, une lame pour la découpe, une équerre pour le dessin du patron de découpe, un crayon, de la colle thermo et du papier de ponçage.

N°	Module	Image	Quantité
1	Contre-plaque 4mm		1 m ²
2	Lame tranchante		1
3	Fer à souder		1
4	Etain		1
5	Fils de connexion		20
6	Colle thermo		3
7	Papier de ponçage		1

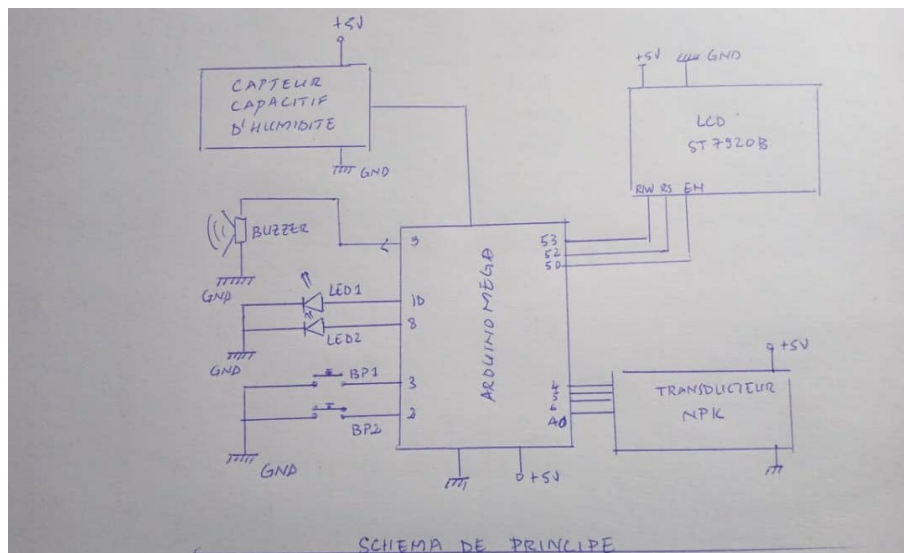
1.2. Matériel et composants

En termes de composant électronique pour le montage du circuit, nous aurons besoins des éléments suivants.

N°	Module	Image	Quantité	Prix (Fcfa)
1	Arduino MEGA 2560		1	8000
2	Ecran LCD ST7920B 128*64		1	5000
3	Transducteur NPK		1	1500
4	Capteur d'humidité		1	2500
5	Buzzer passif		1	200
6	LED		2	50
7	Bouton poussoir		2	100

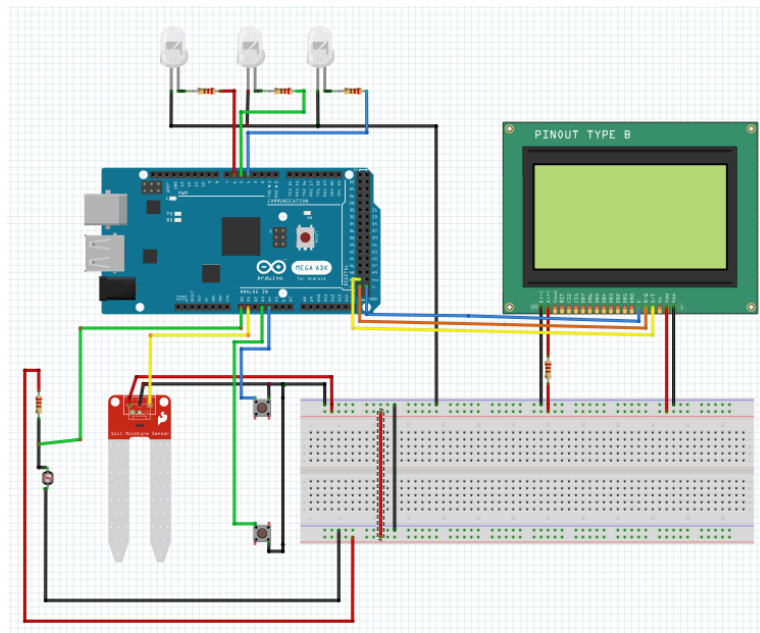
2. Etape 1 : CONCEPTION DU CIRCUIT ELECTRONIQUE

Commençons par concevoir le circuit électronique de notre dispositif, ceci en interconnectant correctement les différents modules.



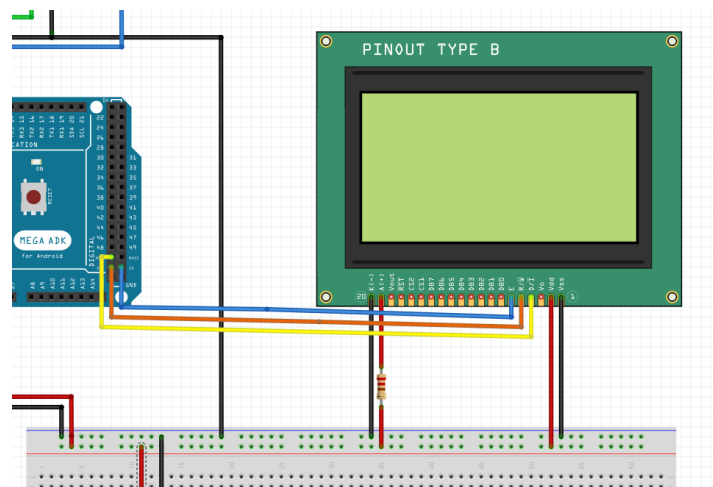
3. Etape 2 : ASSEMBLAGE DU CIRCUIT

Nous suivrons le schéma de montage global ci-dessous pour interconnecter les différents modules.

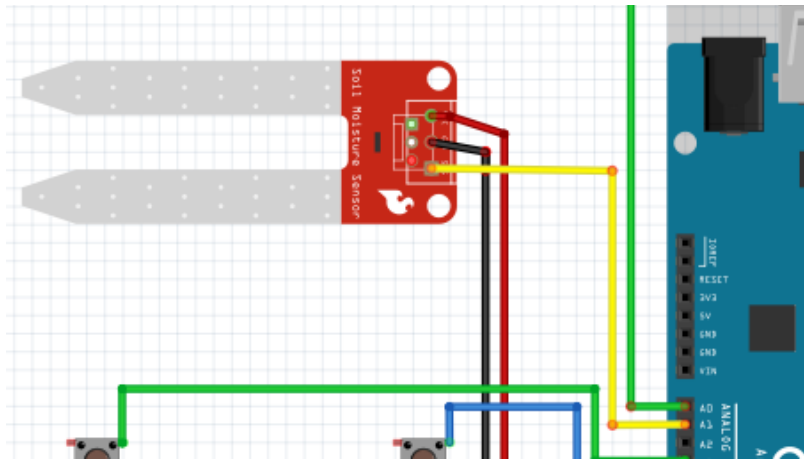


Intéressons-nous de plus près à chacun des modules et à comment les connecter à la carte Arduino MEGA 2560

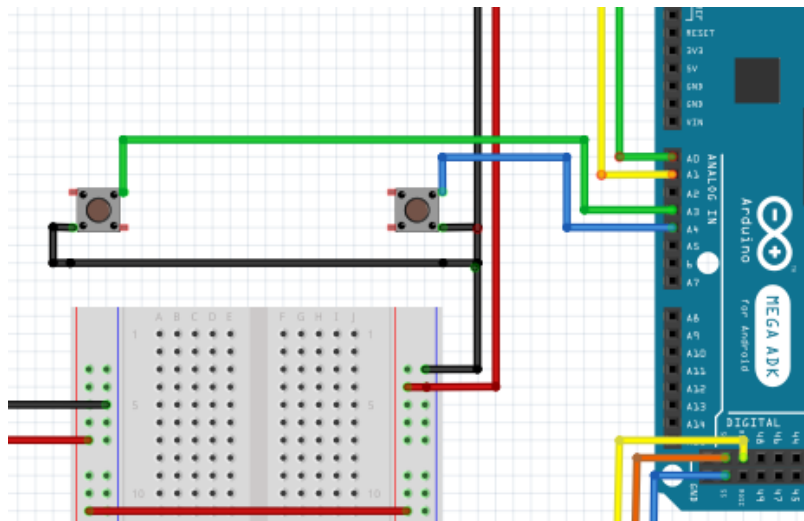
-connexion du module d'affichage LCD :



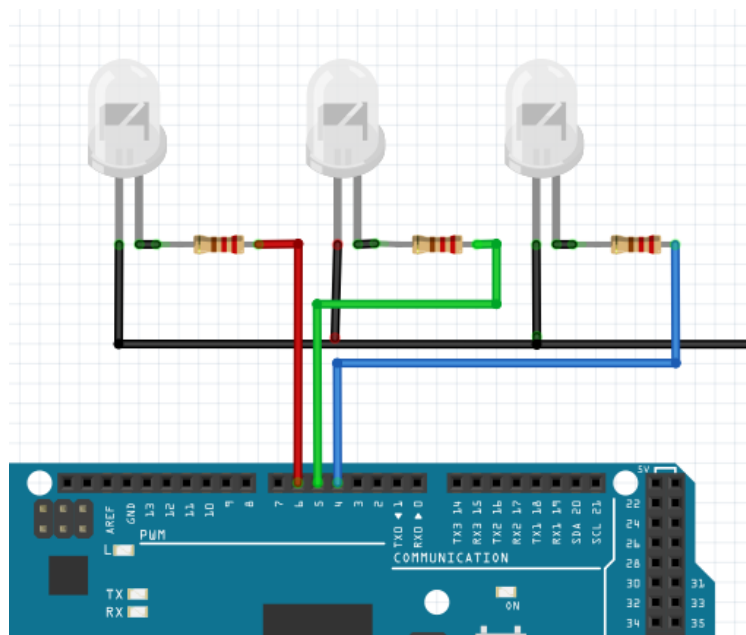
-connexion du module capteur d'humidité :



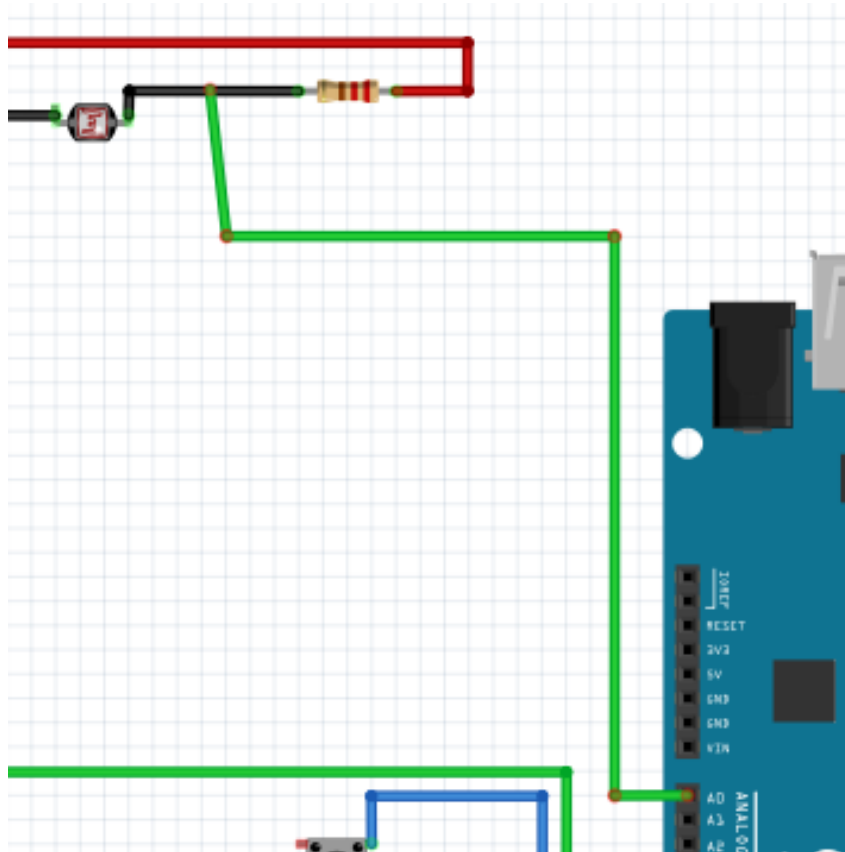
-connexion des boutons poussoirs :



-connexion des LEDS :



-connexion de la photorésistance :



4.Etape 2 : PROGRAMMATION DE LA CARTE

Une fois le circuit correctement assemblé, nous allons passer à la programmation de la carte principale Arduino MEGA 2560(microcontrôleur), pour cela nous allons utiliser l'environnement de développement intégré Arduino IDE 2.1 et un câble USB pour la raccorder à l'ordinateur.

Copiez et collez ce sketch dans votre IDE et téléversez le dans la carte. (Assurez-vous d'avoir choisi la bonne carte et le bon port dans l'onglet outils)

```
#include <SPI.h>
#include <U8g2lib.h>

U8G2_ST7920_128X64_F_SW_SPI screen (U8G2_R2, /*clock=*/ 52, /*data=*/ 50, /*CS=*/ 53, /*reset=*/ 8);

const uint8_t red_button = 2, blue_button = 3, buzzer = 9, humidity_sensor = A0, Photosensor = A1;

const uint8_t led_N = 5; // NITROGEN LED pin(V)

const uint8_t led_P = 6; // PHOSPHORUS LED pin(B)
```

```

const uint8_t led_K = 4; // POTASSIUM LED pin(R)

const int air_calibration_value = 555, water_calibration_value = 314;

float NPK_data[3]; // R-V-B

int digital_voltage, soil_N = 0, soil_P = 0, soil_K = 0;

bool step_number = 0;

uint8_t screen_number = 0, soil_humidity = 0;

void setup(void) {

  Serial.begin(9600);

  pinMode(red_button, INPUT_PULLUP); pinMode(blue_button, INPUT_PULLUP); pinMode(buzzer, OUTPUT);

  pinMode(humidity_sensor, INPUT); pinMode(humidity_sensor, INPUT);

  pinMode(led_N, OUTPUT); pinMode(led_P, OUTPUT); pinMode(led_K, OUTPUT);

  screen.begin(); screen.clearBuffer(); demarrage();screen.clearBuffer();beep(0);
}

void loop() {

  if (screen_number == 0 )press_to_measure();

  if (screen_number == 1 )measuring();

  if (screen_number == 2 && step_number == 0)print_results();

  if (screen_number == 2 && step_number == 1)analyse_results();

}

void press_to_measure() { screen.firstPage();

  do {

    screen.setDrawColor(1); screen.drawFrame(0, 0, 128, 64); screen.setFont(u8g2_font_mozart_nbp_h_all);

    screen.drawStr(4, 15, "Press the red button");screen.drawStr(30, 27, "to measure !");

    screen.drawDisc(30, 47, 9); screen.drawDisc(98, 47, 12); screen.drawCircle(30, 47, 12);

  } while (screen.nextPage());

  if (digitalRead(red_button) == LOW) {

    beep(1); screen_number = 1; delay(100); }

  if (digitalRead(3) == LOW) { beep(1);

    screen_number = 2; delay(100); }

}

void measuring() { screen.clearBuffer();

  bool end_measuring_process = false;

  do {

```



```

    screen.setDrawColor(1); screen.drawFrame(0, 0, 128, 64); screen.setFont(u8g2_font_mozart_nbp_h_all);

    screen.drawStr(13, 15, "MEASURING PROCESS"); screen.drawStr(20, 27, "Please wait ...");

    screen.setFont(u8g2_font_unifont_t_symbols); screen.drawGlyph(55, 50, 0x23F3);
} while (screen.nextPage());

soil_humidity = map(analogRead(humidity_sensor), air_calibration_value, water_calibration_value, 0,100);
if (soil_humidity >= 100)soil_humidity = 100;
if (soil_humidity <= 0)soil_humidity = 0;
for(int i = 0; i < 3; i++){
    digitalWrite(i + 4, HIGH); delay(10);

    digital_voltage = analogRead(NPK_Photosensor) * (5.0 / 1023.0);

    NPK_data[i] = digital_voltage;

    Serial.print("Digital_Volt: "); Serial.print(digital_voltage); Serial.print("    NPK_DATA[i]");
Serial.println(NPK_data[i]); delay(500); digitalWrite(i + 4, LOW);
}

delay(4000); beep(2);

NPK_data[0] = random(2.3,4.1); NPK_data[1] = random(2,2.9); NPK_data[2] = random(2,2.2);

screen_number = 2; step_number = 0;
}

void demarrage() {
    screen.firstPage();

    do {
        screen.drawFrame(0, 0, 128, 64); screen.setFont(u8g2_font_ncenB14_tr);

        screen.drawStr(4, 20, " NPK SOIL");screen.drawStr(0, 40, "    CHECKER");

        screen.drawFrame(20, 45, 88, 8); screen.setFont(u8g2_font_5x8_tf);

        screen.drawStr(27, 62, "GENERAL BIOTECH");
    } while (screen.nextPage());

    for (uint8_t i = 0; i <= 84; i++) {
        screen.drawBox(22, 47, i, 4); delay(random(20, 100)); screen.sendBuffer();
    }

    beep(0); delay(2000);
}

void print results() {

```

```

screen.firstPage();

do {

    screen.drawFrame(3, 0, 54, 54); //screen.drawFrame(2, 2, 56, 60);

    screen.drawFrame(62, 0, 63, 16); //screen.drawFrame(61, 2, 65, 18);

    screen.drawFrame(62, 19, 63, 16); // screen.drawFrame(61, 23, 65, 18);

    screen.drawFrame(62, 38, 63, 16); //screen.drawFrame(61, 44, 65, 18);

    screen.setFont(u8g2_font_ncenB12_tr); screen.drawStr(8, 17, "HUM");screen.setCursor(22, 35);

    screen.print(soil_humidity); screen.drawStr(23, 52, "%");screen.setFont(u8g2_font_ncenB08_tr);

    screen.drawStr(65, 12, "N :"); screen.setCursor(85, 12); screen.print(soil_N);

    screen.drawStr(65, 31, "P :"); screen.setCursor(85, 31); screen.print(soil_P);

    screen.drawStr(65, 50, "K :"); screen.setCursor(85, 50); screen.print(soil_K);

    screen.setFont(u8g2_font_mozart_nbp_h_all); screen.drawStr(18, 64, "Back Analyse");

    //digitalWrite(VOYANT_J, HIGH);

    screen.setFont(u8g2_font_mozart_nbp_h_all);

    if (NPK_data[1] > 4.2) {

        soil_N = 80;

        screen.drawStr(82, 12, "HIGH");

    }

    else if (NPK_data[1] > 3.8 && 4.1 > NPK_data[1]) {

        soil_N = 50;

        screen.drawStr(82, 12, "MEDIUM");

    }

    else if (NPK_data[1] > 3.5 && 3.8 > NPK_data[1]) {

        soil_N = 25;

        screen.drawStr(82, 12, "LOW");

    }

    else {

        soil_N = 0;

        screen.drawStr(82, 12, "NONE");

    }

    if (NPK_data[2] > 3.4) {

        soil_P = 80;

        screen.drawStr(82, 31, "HIGH");

    }

}

```

```

else if (NPK_data[2] > 2.9 && 3.3 > NPK_data[2]) {
    soil_P = 50; screen.drawStr(82, 31, "MEDIUM");
}
else if (NPK_data[2] > 2.45 && 2.8 > NPK_data[2]) {
    soil_P = 25; screen.drawStr(82, 31, "LOW");
}
else {
    soil_P = 0; screen.drawStr(82, 31, "NONE");
}
if (NPK_data[0] > 2.9) { // DigitalVolt > 4.2 volt
    soil_K = 80; screen.drawStr(82, 50, "HIGH");
}
else if (NPK_data[0] > 2.3 && 2.8 > NPK_data[0]) {
    soil_K = 50; screen.drawStr(82, 50, "MEDIUM");
}
else if (NPK_data[0] > 1.6 && 2.2 > NPK_data[0]) {
    soil_K = 25; screen.drawStr(82, 50, "LOW");
}
else {
    soil_K = 0; screen.drawStr(82, 50, "NONE");
}
} while (screen.nextPage());
if (digitalRead(red_button) == LOW) {
    beep(1); screen_number = 0; delay(200);
}
if (digitalRead(3) == LOW) {
    beep(1); screen_number = 2; step_number = 1; delay(200);
}
}

void analyse_results() {
    uint8_t h_height = map ( soil_humidity, 0, 100, 0, 40);
    uint8_t n_height = map ( soil_N, 0, 100, 0, 40);
    uint8_t k_height = map ( soil_P, 0, 100, 0, 40);
    uint8_t p_height = map ( soil_K, 0, 100, 0, 40);
}

```

```

screen.firstPage();

do {

    screen.setFont(u8g2_font_mozart_nbp_h_all); screen.drawStr(18, 64, "Back    Results");

    screen.drawFrame(0, 0, 128, 56); screen.drawVLine(3, 10, 44); screen.drawHLine(2, 12, 3);

    screen.drawHLine(2, 22, 3); screen.drawHLine(2, 32, 3); screen.drawHLine(2, 42, 3);

    screen.drawHLine(2, 52, 41); screen.drawVLine(8, 51, 3); screen.drawVLine(18, 51, 3);

    screen.drawVLine(28, 51, 3); screen.drawVLine(38, 51, 3); screen.setFont(u8g2_font_4x6_tf);

    screen.drawStr(7, 10, "H"); screen.drawStr(17, 10, "N");screen.drawStr(27, 10, "P");

    screen.drawStr(37, 10, "K");screen.drawBox(55, 2, 70, 7); screen.setDrawColor(0);

    screen.drawStr(60, 8, "About this soil"); screen.setDrawColor(1);

    screen.drawBox(6, 52 - h_height, 5, h_height); screen.drawBox(16, 52 - n_height, 5, n_height);

    screen.drawBox(26, 52 - k_height, 5, k_height); screen.drawBox(36, 52 - p_height, 5, p_height);

} while (screen.nextPage());

if (digitalRead(2) == LOW) {

    beep(1); screen_number = 0; delay(200);

}

if (digitalRead(3) == LOW) {

    beep(1); screen_number = 2; step_number = 0; delay(200);

}

}

void beep(uint8_t n) {

    if (n == 0) {

        tone(9, 1700 , 300);

    }

    if (n == 1) {

        tone(9, 2000 , 100);

    }

    if (n == 2) {

        tone(9, 2000 , 50); delay(100); tone(9, 2000 , 50);

    }

    if (n == 3) {

        tone(9, 500 , 50);

    }

}
}

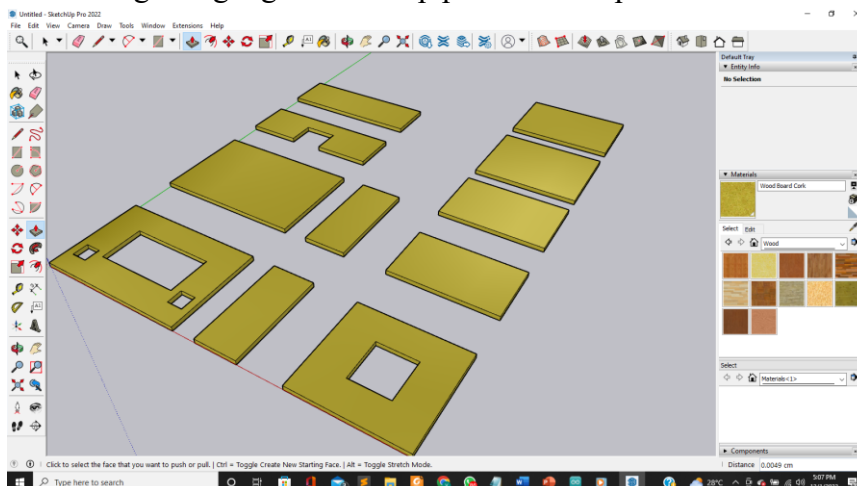
```

NB : il sera nécessaire de télécharger et installer la bibliothèque U8G2LIB pour la gestion graphique du module d'affichage ST7920B.

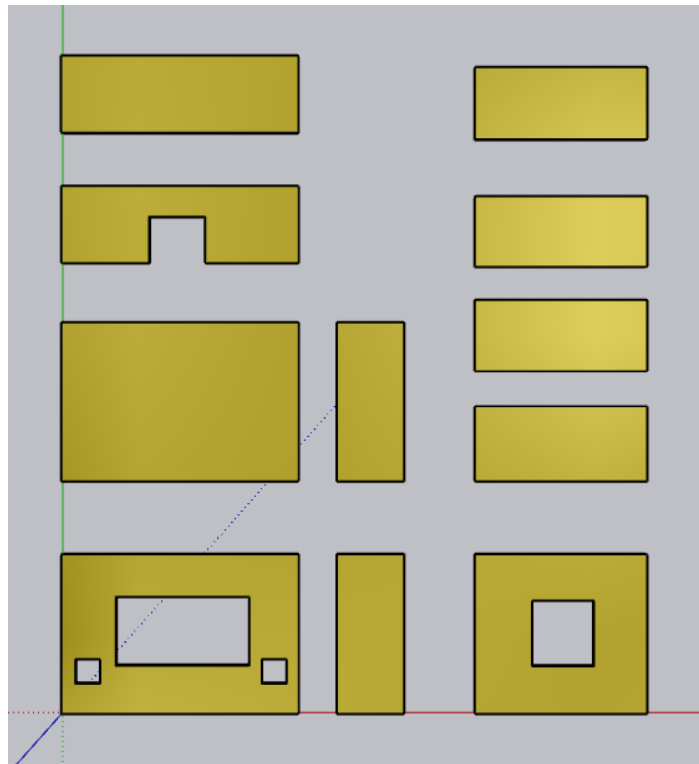
5. FABRICATION DU BOITIER

Maintenant que nous avons assemblé et programmé le dispositif, il est temps de le conditionner dans un joli et solide boîtier que nous allons fabriquer dans cette section ; Pour cela nous avons utilisé du contre-plaqué 4mm que nous avons découpé et assemblé avec de la colle thermo.

Nous avons utilisé le logiciel google SketchUp pour la conception 3d du boîtier.



Ce qui nous a permis d'avoir les patrons de découpe.



Vous pouvez redessiner le design du boîtier selon vos goûts et préférences.

6. INTEGRATION DU CIRCUIT DANS LE BOITIER FINAL ET TEST

Maintenant que le boîtier est réalisé, il est temps d'y insérer correctement et délicatement le circuit.

D'abord le module d'affichage, ensuite les boutons, le buzzer et puis le module de mesure d'humidité et de NPK.



7. UTILISATION DU DISPOSITIF

Pour utiliser le prototype, il suffira de creuser un trou d'environ 10 cm dans le sol et à l'endroit où l'on souhaite effectuer la nouvelle mesure. Ensuite il faudra placer le dispositif au-dessus de façon à ce que le capteur d'humidité s'enfouisse aisément dans le sol. Ensuite l'appui sur le bouton rouge permettra d'effectuer une mesure dont les résultats seront disponibles à l'affichage au bout de 5 secondes. Le bouton bleu quand à lui permettra de réaliser une microanalyse des résultats précédemment obtenus.

