

Daoud

Oulad El Fadel

6A

Main robotique



INRACI 2023-2024

Mes remerciements les plus sincères à mon promoteur monsieur Kapita pour sa formation et la mise à disposition de son temps et de son matériel pour mener à bien mon projet. Une personne inspirante qui a mon plus grand respect.

Mes remerciements à Thomas Giarrizzo qui a été un camarade exceptionnel et d'une grande aide. Je voudrais aussi remercier mes parents et Nathan et mes autres camarades pour le soutien qu'ils m'ont apporté.

Sommaire

1.L'introduction

2.Les caractéristiques

3.Les principes mis en jeu

1. Démodulation
2. Encodeur

4. Le schéma bloc

1. Gant
2. Main

5.L'étude détaillée et les mesures

1. Étude du tactile
2. Mesure sur le moteur

6.La programmation

1. L'ordinogramme
2. Le programme

7.La fabrication

8.La mise au point

9.La conclusion

10.Les annexes

1. Schéma de principe sans connecteurs
2. PCB
4. PCB sérigraphie

11.Vue 3D

1. Gant
2. Main
3. Moteur
4. Capteur de flexion
5. ESP32
6. Gant
7. M5 Paper
8. ADC

12.Fiche technique des composants

1. ESP32
2. M5 PAPER
3. Module ADC
4. Encodeur

13.Bibliographie et webographie

1.L'introduction

A l'occasion de mon TFE de fin de 6eme année secondaire. J'ai décidé de mener un projet qui va me permettre d'appliquer mes connaissances acquises grâce à mon apprentissage à l'INRACI.

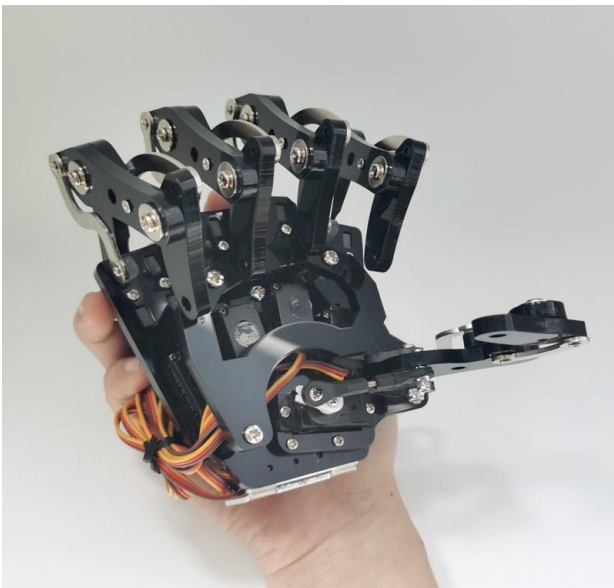
Le but de mon projet est de créer une main que l'on pourra contrôler grâce à un gant à distance.

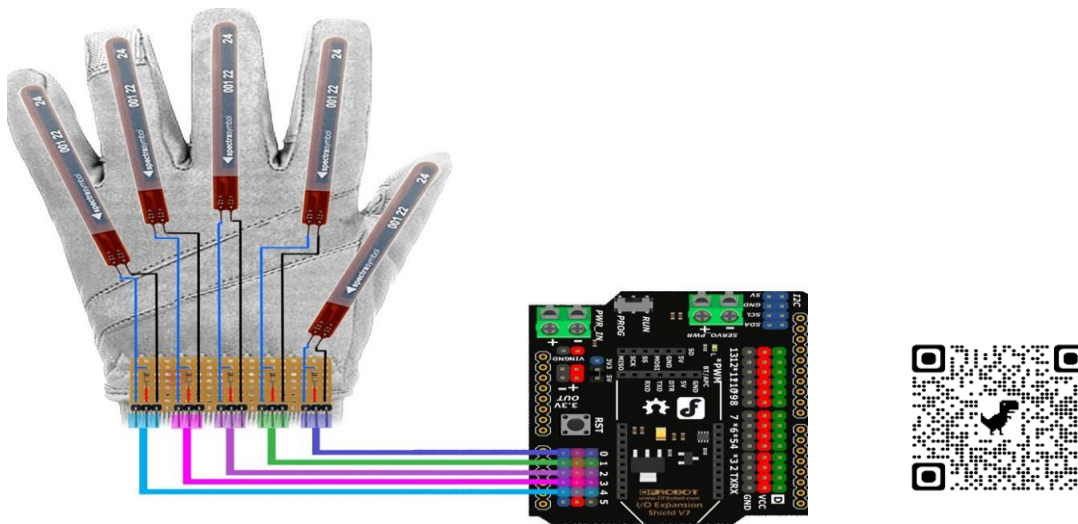
Mes motivations pour ce projet viennent de ma passion pour la domotique et le contrôle à distance.

Depuis tout petit je suis fascinée par les entités contrôler à distance et ce projet est l'occasion de réunir ce que j'aime et l'électronique.

Rapide explication

Les servos moteurs qui seront reliés aux doigts de la main robotique vont être branchés à une première carte ESP32 qui communiquera avec la même carte. La deuxième carte sera elle branchée aux 5 capteurs de flexion qui seront collés aux gants. Les 2 cartes pourront communiquer grâce à leur options BT intégrées.





2. Les caractéristiques

Caractéristiques :

General :	Main	Gants
-2 boutons poussoir	Sur la carte ESP32	Sur la carte ESP32
-1 Led	Led sur la Esp32	Led sur la Esp32
-Main robotique avec 5 servo moteur	Servo moteur SG90	X
-Gants avec 5 capteurs de flexions (x résolutions)	X	A déterminer

Electronique :	Main	Gants
-BT 2,4 GHz	Sur la carte ESP32	Sur la carte ESP32
-Consommation	170 et 250 mA pour les 5 servo moteur	ESP32 (80-180 mA)

-Batteries 5 volt	A déterminer	A déterminer
-------------------	--------------	--------------

Mécanique :	Main	Gants
Poids	289g	A déterminer
Matière	Acrylique noir 5.0	Tissu

Liste des composants :

Electronique

- 2X Adafruit Carte Feather HUZZAH32 ESP32
- 5x Capteur de Force, ZD10-100 500g
- 2X Batteries (a determiner)
- 1X M5PAPER

Mécanique

- 1X Bras mécanique
- 5X servo moteur (SG90)
- 1X Gants en tissus

3. Les principes mis en jeu

1. Démodulation

Description détaillée

Notre projet comporte 2 parties (le gant avec une M5 Paper et la main robotique avec une ESP32) qui vont se transmettre des informations via une connexion Bluetooth. La transmission Bluetooth utiliser est le BLE (Bluetooth Low Energy). Nos 2 dispositifs de communication sont dotés de ce type de transmission. Cette technologie de transmission de données nous permet d'échanger des données entre la M5 Paper et l'ESP32 avec une faible consommation d'énergie. Elle nous permet aussi d'avoir une communication sécuriser grâce au chiffrement de données et d'avoir un temp de connexion rapide.

Caractéristiques détaillées

1 Consommation d'énergie faible

La transmission BLE consomme peu d'énergie ce qui en fait un atout majeur pour les systèmes embarque qui ont besoin de consommer le moins possible.

2 Portée de communication

La portée de communication du BLE est de 10 mètres entre nos 2 système.

3 Fréquence de communication

La fréquence utiliser par le BLE est de 2,4 GHz.

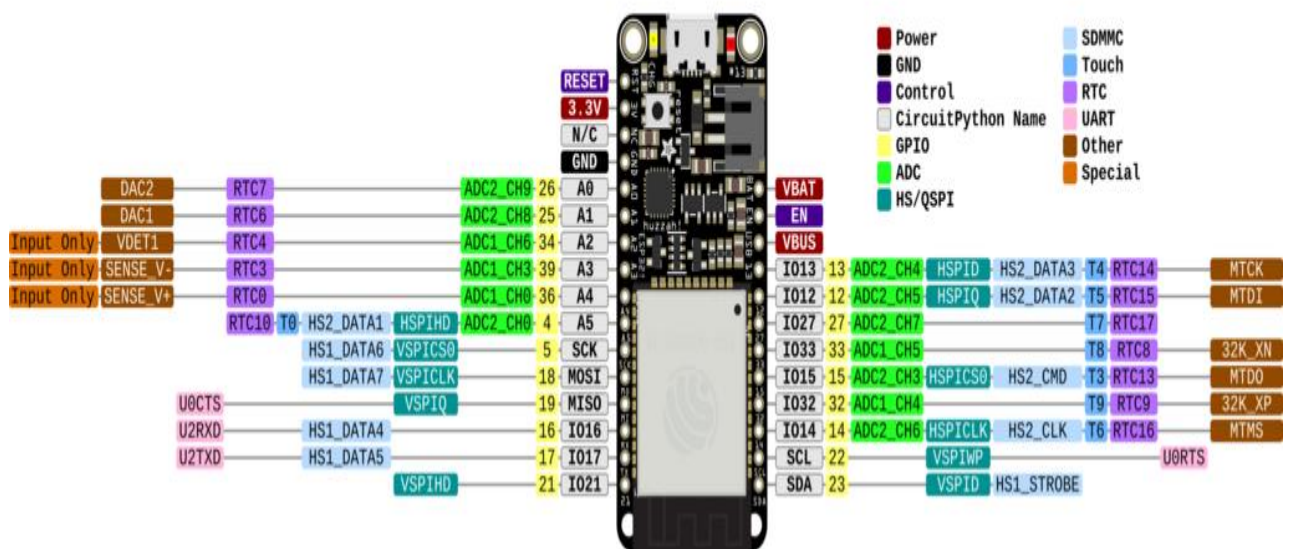
4 Maitre multiple

Il peut avoir plusieurs esclaves contrôler par 1 seul maitre.

Brochage :

Adafruit HUZZAH32 ESP32 Feather

<http://www.adafruit.com/products/3405>





Étude théorique du principe physique de la réception (démodulation) Bluetooth.

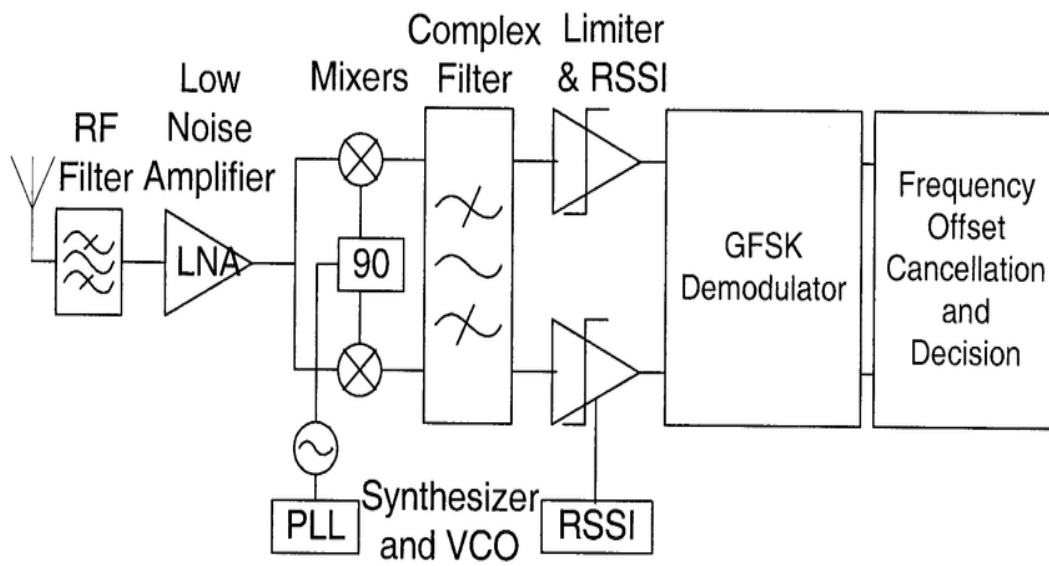
Le principe de modulation utiliser est le GFSK qui est une technique de variation de fréquences qui est utiliser pour transmettre des données.

L'émetteur (M5 Paper) va envoyer les données moduler en GFSK via une radiofréquence vers l'esp32.

L'ESP32 grâce à une antenne va capter l'onde et acheminer le signal vers le circuit de réception.

Avant d'arriver à l'étape de démodulation le signal va être filtrée, le bruit capter va être éliminer et le signal utile va être renforcer.

Le circuit de réception va démoduler le signal reçu en traduisant les variations de fréquence pour récupérer les données originales.



Programmes de test

```

8  #include <ESP32Servo.h>
9  #include <Wire.h>
10 #include "BluetoothSerial.h"
11
12 #if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
13 #error Bluetooth is not enabled! Please run 'make menuconfig' to and enable it
14 #endif
15 #if !defined(CONFIG_BT_SPP_ENABLED)
16 #error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
17 #endif
18
19
20 // Bluetooth Serial Object (Handle)
21 BluetoothSerial SerialBT;
22
23 Servo myservoPouce;
24 Servo myservoINDEX;
25 Servo myservoMajeur;
26 Servo myservoAnnulaire;
27 Servo myservoAuriculaire;
28
29 char pos = 0;
30 char pos1 = 0;
31 char pos2 = 0;
32 char pos3 = 0;
33 char pos4 = 0;
34
35 MSROTATE8 MM;
36 // ESP32 Bluetooth (Slave) Device Information
37 const char *pin = "1234"; // Change this to more secure PIN.
38 String device_name = "ESP32-BT-Slave";
39 int servoPouce = 14;
40 int servoINDEX = 32;
41 int servoMAJEUR = 15;
42 int servoAnnulaire = 33;
43 int servoAuriculaire = 27;

```

```

45 char receiveData[7];
46
89 myservoPouce.write(receiveData[1]);
90 myservoINDEX.write(receiveData[2]);
91 myservoMajeur.write(receiveData[3]);
92 myservoAnnulaire.write(receiveData[4]);
93 myservoAuriculaire.write(receiveData[5]);
94
95 if (receiveData[0] == '#' && receiveData[6] == (receiveData[1] ^ receiveData[2] ^ receiveData[3] ^ receiveData[4] ^ receiveData[5]))
96     Serial.print("Data OK\n");
97 else
98     Serial.print("Data INVALID\n");
99
100 receiveData[0] = 0; //on efface le buffer de reception
101 }
102 }

```

```

73 if (SerialBT.available()) {
74     receiveData[0] = SerialBT.read();
75     if (receiveData[0] == '#') {
76         for (char cptSend = 1; cptSend < NB_data; cptSend++) {
77             while (!SerialBT.available()) ;
78             receiveData[cptSend] = SerialBT.read();
79         }
80     }
81     Serial.println(receiveData[0]);
82     Serial.printf("a:%d\n", receiveData[1]);
83     Serial.printf("b:%d\n", receiveData[2]);
84     Serial.printf("c:%d\n", receiveData[3]);
85     Serial.printf("d:%d\n", receiveData[4]);
86     Serial.printf("e:%d\n", receiveData[5]);
87     Serial.printf("f:%d\n", receiveData[6]);
88 }

```

Librairie utiliser

"BluetoothSerial.h" version 1.0.1.

Auteur : Henri Abrahamsen

Site consultée le 31/01/2024 à 21 :43

<https://github.com/hen1227/bluetooth-serial>



"ESP32Servo.h" version 1.1.2

Auteur : Kevin Harrington, John K. Bennett

Site consultée le 14/02/2024 à 21 :26

<https://www.arduino.cc/reference/en/libraries/esp32servo/>



2.Encodeur

Introduction

L'un des sous ensemble de notre projet est le 8-Encoder Unit qui nous permet de contrôler les moteurs qui contrôle les doigts de la main. Les encodeurs utiliser pour notre projet sont les EC11 qui sont des encodeurs incrémentaux. La position des encodeurs est communiquée grâce à microcontrôleur en I2C vers notre ESP32.

UNIT
8ENCODER

M5STACK



SW	= PB3	SDA	= PB11
SCL	= PB10	RGB	= PB0
ENC1	PA10	PA11	PB2
ENC2	PA8	PA9	PB3
ENC3	PA34	PA15	PB4
ENC4	PA12	PA13	PB5
ENC5	PA6	PA7	PB6
ENC6	PA4	PA5	PB7
ENC7	PA2	PA3	PB8
ENC8	PA0	PA1	PB9

I2C
INTERFACE



Fonctionnement d'un encodeur incrémental

Définition :

Un encodeur incrémental est un capteur qui suit le mouvement d'un objet en incrémentant ou décrémentant sa position par rapport à une position de référence.

L'encodeur incrémental est constitué de 2 parties : le stator et le rotor.

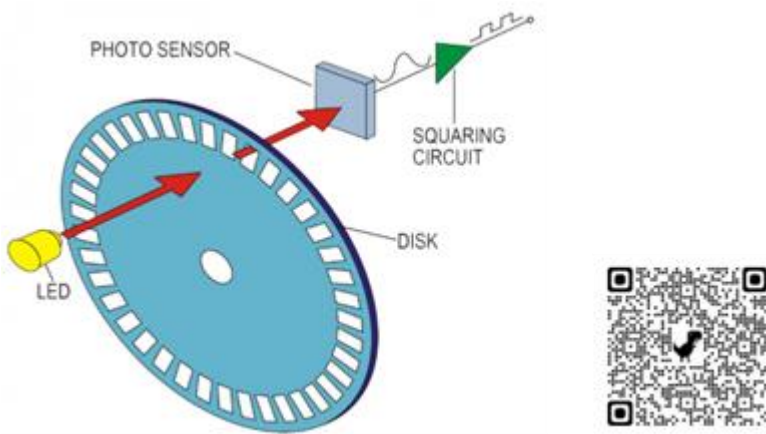
Le rotor est la partie tournante qui est reliée à l'entité dont on souhaite mesurer la position, la distance ou la vitesse.

Le stator est la partie fixe sur laquelle se trouve le capteur qui va nous permettre de mesurer les données que l'on veut (vitesse, position, etc.).

Fonctionnement :

Il existe 3 types de codeurs : les codeurs magnétiques, les codeurs optiques et les codeurs linéaires.

Le codeur optique est constitué d'un rotor composé d'un disque ayant des fentes avec un espace régulier entre eux. Le stator est lui composé d'un capteur optique qui va capter la lumière émise par une source lumineuse de l'autre côté du rotor ce qui va créer des impulsions électriques qui vont nous permettre de savoir sur quelle position est le rotor.



Le codeur magnétique est constitué d'un rotor composé d'un disque ayant des marques magnétiques avec un espace régulier entre eux. Le stator est lui composé d'un capteur électromagnétique qui va capter les variations du champ magnétique grâce à l'effet hall et les convertir en impulsion électrique.

Le codeur linéaire utilise le même principe que le codeur optique et le codeur électromagnétique mais au lieu de mesurer une rotation mais un mouvement linéaire.

Il n'y a alors pas de disque mais un ruban constitué de fentes pour les codeurs linéaire optique et des marques magnétiques sur le ruban. L'entité dont on souhaite mesurer la position (ou la vitesse, etc) coulissera le long du ruban ce qui va perturber le champ magnétique si nous avons un capteur linéaire magnétique ou le capteur optique va détecter la lumière si c'est un codeur linéaire optique ce qui va alors générer des impulsions électriques qui vont être traitées pour connaître la position de l'entité.

Traitements des impulsions électriques

Pour obtenir l'information souhaitée, il faut traiter les impulsions électriques générées par le capteur. Le système chargé d'effectuer le traitement des impulsions est un compteur qui va compter les impulsions et incrémenter ou décrémenter selon le déplacement de l'objet.

Le codeur va délivrer 2 signaux carrés (A et B) qui vont créer un déphasage qui va nous permettre de déterminer la position de l'objet.

Le codeur peut utiliser les voies A et B séparément ou ensemble ce qui va influencer la résolution du codeur donc sa précision.

Programme de test :

```

9 //*****Librairie*****
10
11 // Inclusion des bibliothèques nécessaires
12 #include <ESP32Servo.h> // Bibliothèque pour contrôler les servomoteurs avec l'ESP32
13 #include <Wire.h> // Bibliothèque pour la communication I2C
14 #include "m5rotate8.h"
15
16 //*****Constante*****
17
18 int servoPouce = 13; // Broche pour le servomoteur Pouce
19 int servoINDEX = 12; // Broche pour le servomoteur INDEX
20 int servoMAJEUR = 27; // Broche pour le servomoteur MAJEUR
21 int servoAnnulaire = 33; // Broche pour le servomoteur Annulaire
22 int servoAuriculaire = 15; // Broche pour le servomoteur Auriculaire
23
24 //*****Variable*****
25
26 // Variables pour stocker les positions des servomoteurs
27 char pos = 0;
28 char pos1 = 0;
29 char pos2 = 0;
30 char pos3 = 0;
31 char pos4 = 0;
32
33 //*****Objet*****
34
35 // Déclaration des objets Servo
36 Servo myservoPouce;
37 Servo myservoINDEX;
38 Servo myservoMajeur;
39 Servo myservoAnnulaire;
40 Servo myservoAuriculaire;
41 M5ROTATE8 MM;
42
43
44 //*****Initialisation*****
45
46 void setup() {
47 // Attacher les servomoteurs à leurs broches respectives
48 myservoPouce.attach(servoPouce);
49 myservoINDEX.attach(servoINDEX);
50 myservoMajeur.attach(servoMAJEUR);
51 myservoAnnulaire.attach(servoAnnulaire);
52 myservoAuriculaire.attach(servoAuriculaire);
53
54 // Initialisation de la communication I2C
55 Wire.begin();

```



```

57 // Initialisation de la communication série avec un débit de 115200 bauds
58 Serial.begin(115200);
59 }
60
61 // Fonction de boucle principale
62 void loop() {
63 // Fonction pour lire et déplacer les servomoteurs en fonction des valeurs des encodeurs rotatifs
64 Lecture_position();
65 Affiche_Position();
66 }
67
68
69
70 //*****Fonction*****
71
72 void Lecture_position() {
73 // Déplacer les servomoteurs en fonction des valeurs des encodeurs rotatifs
74
75 int pos4 = MM.getAbsCounter(0); // Valeur de l'encodeur rotatif 1
76 int pos3 = MM.getAbsCounter(1); // Valeur de l'encodeur rotatif 2
77 int pos2 = MM.getAbsCounter(2); // Valeur de l'encodeur rotatif 3
78 int pos1 = MM.getAbsCounter(3); // Valeur de l'encodeur rotatif 4
79 int pos = MM.getAbsCounter(4); // Valeur de l'encodeur rotatif 5
80
81 myservoAuriculaire.write(pos4);
82 myservoAnnulaire.write(pos3);
83 myservoMajeur.write(pos2);
84 myservoINDEX.write(pos1);
85 myservoPouce.write(pos);
86 }
87
88 void Affiche_Position() {
89 Lecture_position();
90 // Afficher les valeurs des encodeurs rotatifs dans le moniteur série
91 Serial.print("Encodeur 1: ");
92 Serial.print(pos4);
93 Serial.print("\t");
94
95 Serial.print("Encodeur 2: ");
96 Serial.print(pos3);
97 Serial.print("\t");
98
99 Serial.print("Encodeur 3: ");
100 Serial.print(pos2);
101 Serial.print("\t");
102
103 Serial.print("Encodeur 4: ");
104 Serial.print(pos1);
105 Serial.print("\t");
106
107 Serial.print("Encodeur 5: ");
108 Serial.println(pos);
109 }

```

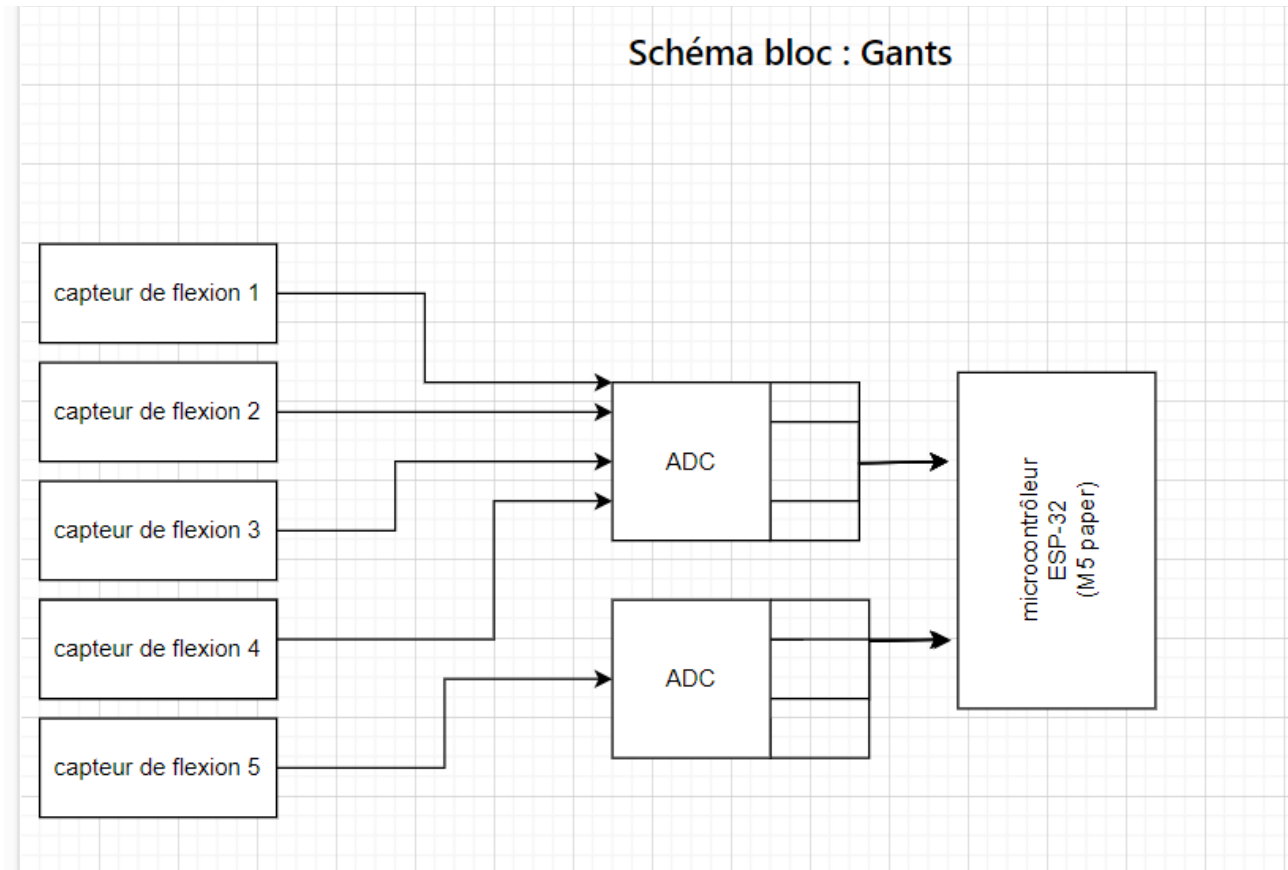
LIBRAIRIE : [M5ROTATE8](#)

Version : 0.3.0



4.Le schéma bloc

1.Gant



Gants :

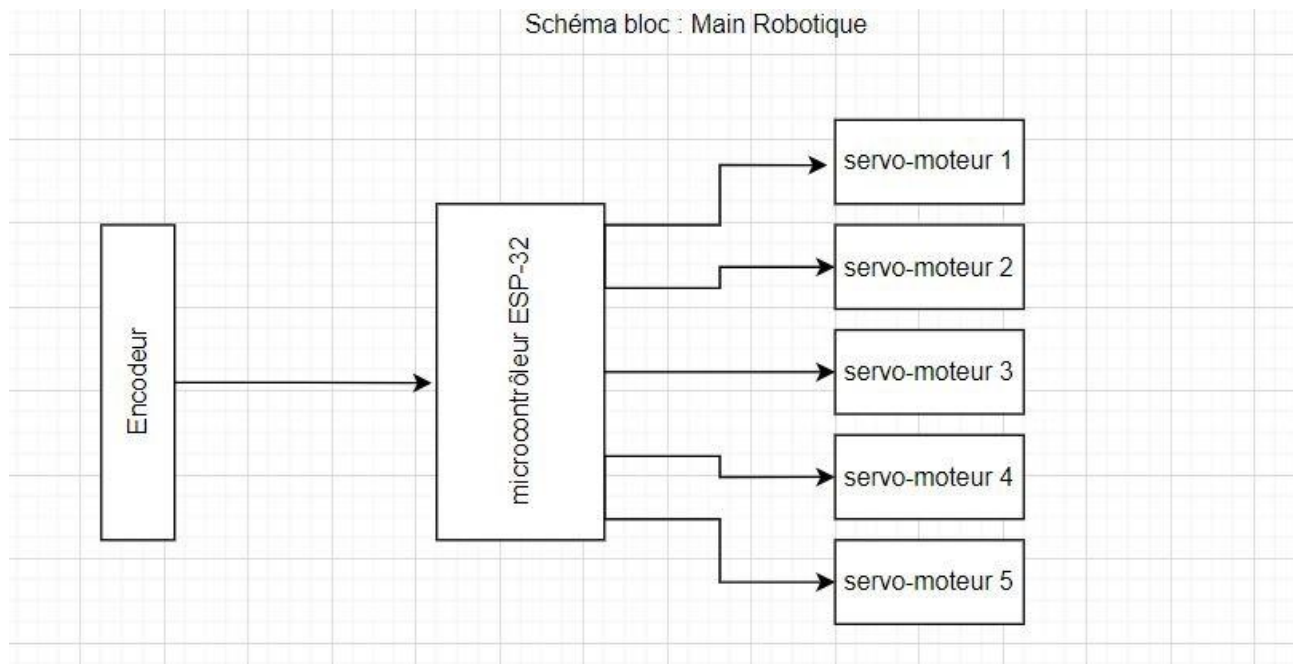
Capteur de flexion et module ADC :

Les capteurs de flexion vont transmettre le mouvement à la ESP32 (M5 Paper) grâce à deux modules I2C. 2 capteurs de flexion dans l'un des modules et 3 dans l'autre. Les modules vont permettre de relier les capteurs de flexions au microprocesseur en réduisant le nombre de connexions.

M5 Paper :

L'ESP32(M5 Paper) va envoyer les instructions à l'ESP32 de la main robotique grâce au module BT intégré. L'écran tactile intégrée va afficher la position des doigts et le menu pour pouvoir choisir les différents modes de fonctionnement.

2.Main Robotique



Main robotique :

ESP32 et servo moteur :

Le microprocesseur de la main robotique va recevoir les instructions grâce au module BT intégré qui va ensuite les appliquer sur les ou le servo moteur concerner. L'ESP-32 va renvoyer la positions des doigts vers la M5 Paper a fin de l'afficher.

Encodeur :

Il est composé de 8 encodeurs mécanique qui vont nous permettre de faire bouger "manuellement" nos doigts. L'encodeur est brancher à l'ESP-32 qui va contrôler les moteurs selon la position des encodeurs attitré à chaque doigt.

5.L'étude détaillée et les mesures

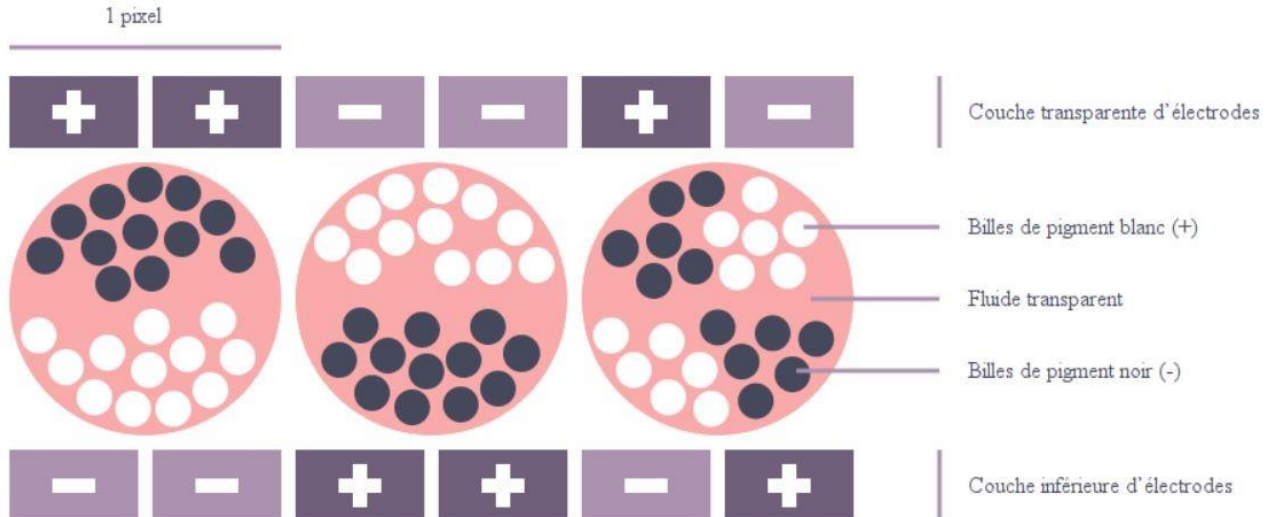
1. Analyse de l'écran (M5 PAPER)

Introduction

Notre projet se constitue de la M5 Paper qui va nous permettre de communiquer la position des capteurs à notre main robotique. La M5 Paper est aussi composée d'un écran le EPD_ED047TC1 qui va nous permettre d'afficher les différents modes (contrôle avec le gant, mouvement avec l'écran et le mode musique) et de les sélectionner grâce au tactile de l'écran. Il va aussi nous permettre de contrôler la main robotique en appuyant directement sur le choix de la position des doigts.

Comment marche l'affichage ?

L'écran utilise la technologie E-Ink qui permet un affichage à très faible consommation et n'a pas besoin d'énergie pour laisser une image afficher car il ne consomme que lorsque l'image est modifiée. L'écran est composé de pixels dans lesquels il y a de l'encre composée de billes de pigments chargées positivement pour créer un pigment blanc et négativement pour créer un pigment noir qui baignent dans un fluide transparent. Lorsque l'on va vouloir afficher la couleur noire, nous allons envoyer une tension électrique négative pour attirer les billes de pigments blancs et repousser de l'autre côté du fluide, une tension positive pour attirer les billes de pigments noirs et repousser les charges négatives de l'autre côté du fluide. Sans présence de tension, les pigments restent à leur place.

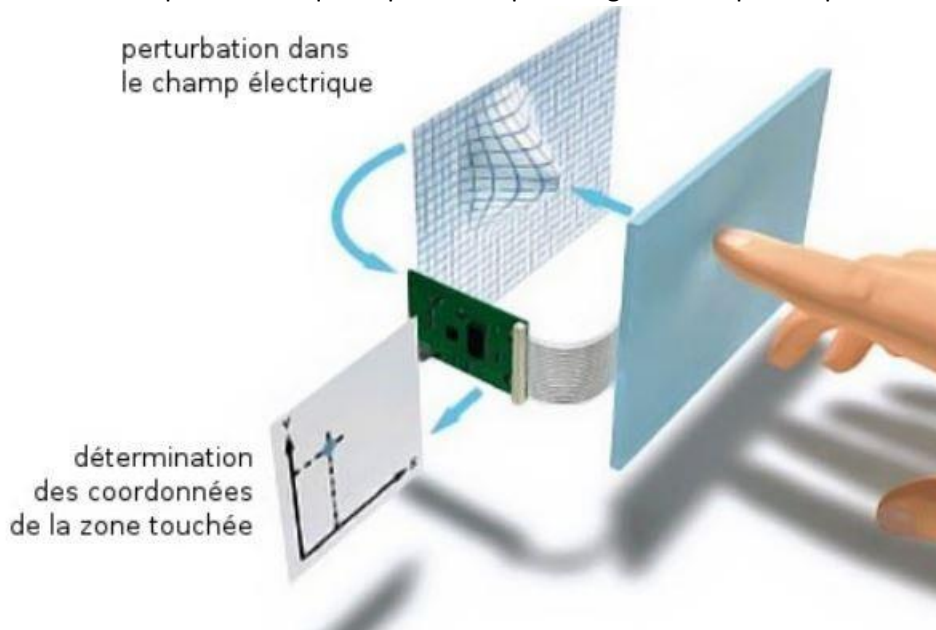


Source image :



Comment fonctionne le tactile ?

L'écran va être traversé par un champ électrique faible qui quadrille l'écran. Le doigt est un conducteur électrique et lorsque l'on va appuyer sur une partie de l'écran le champ électrique va être perturbé et le processeur va pouvoir sur quelle partie du quadrillage le champ a été perturbé.



Source image :



Code de "teste" :

```
#include <M5EPD.h>
M5EPD_Canvas canvas(&M5.EPD);
int point[2][2];
void setup() {
    M5.begin();
    M5.EPD.SetRotation(90);
    M5.TP.SetRotation(90);
    M5.EPD.Clear(true);
    canvas.createCanvas(540, 960);
    canvas.setTextSize(5);
    canvas.drawString("Touch The Screen!", 20, 400);
    canvas.pushCanvas(0, 0, UPDATE_MODE_DU4);
}
void loop() {
    if (M5.TP.available()) {
        if (!M5.TP.isFingerUp()) {
            M5.TP.update();
            canvas.fillCanvas(0);
            bool is_update = false;
            for (int i = 0; i < 2; i++) {
                tp_finger_t FingerItem = M5.TP.readFinger(i);
                if ((point[i][0] != FingerItem.x) ||
                    (point[i][1] != FingerItem.y)) {
```

```

        is_update    = true;
        point[i][0] = FingerItem.x;
        point[i][1] = FingerItem.y;
        canvas.fillRect(FingerItem.x - 50, FingerItem.y - 50, 00, 100,
15);

        Serial.printf("Finger ID:%d-->X: %d*C Y: %d Size: %d\r\n",
                        FingerItem.id, FingerItem.x, FingerItem.y,
                        FingerItem.size);
    }
}
if (is_update) {
    canvas.pushCanvas(0, 0, UPDATE_MODE_DU4);
}
}
}
}

```

Caractéristiques de l'écran :

Taille diagonale : 4,7"

Zone active : 58,32 (L) × 103,68 (H) mm

Type d'interface : Données parallèles (1 canal, 8 bits), connecteur 44 broches

Résolution : 540 × 960, qHD 235 PPI

Fréquence : 85Hz

Tension d'entrée : 3,3 V (typique)

Environnement : Température de fonctionnement : 0 ~ 50 °C ; Température de stockage. : -25 ~ 70 °C

Niveau de gris : 16

Sources :

Tactile



Affichage



Caracteristiques



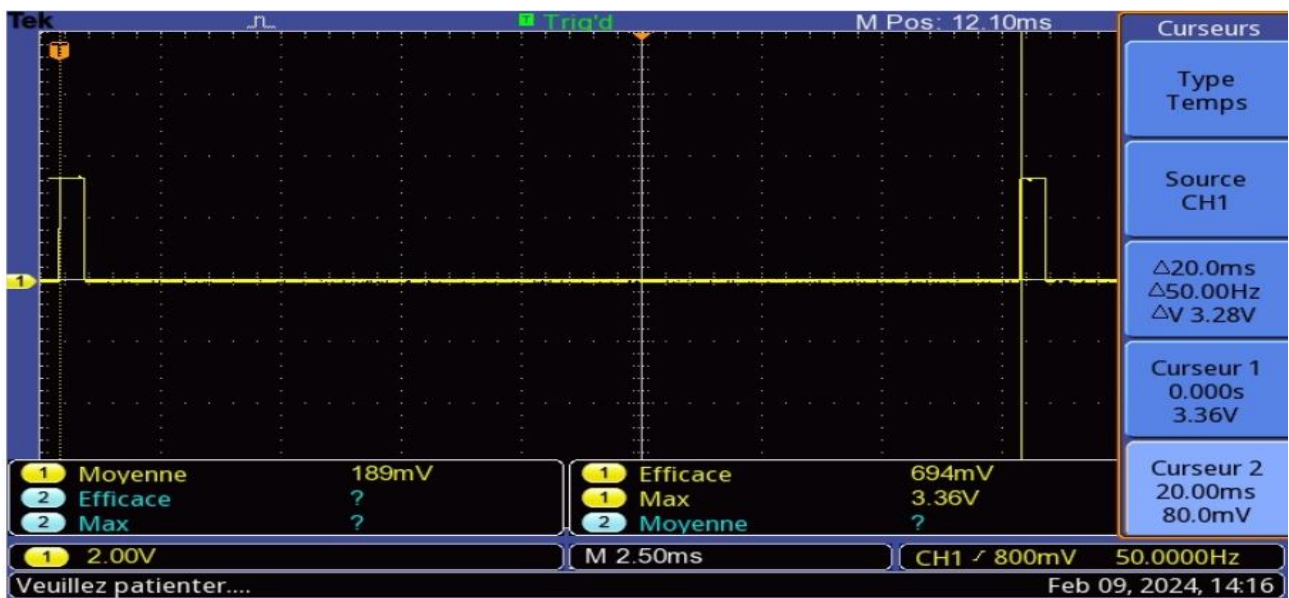
5. Mesures

Mesures sur les servos moteur

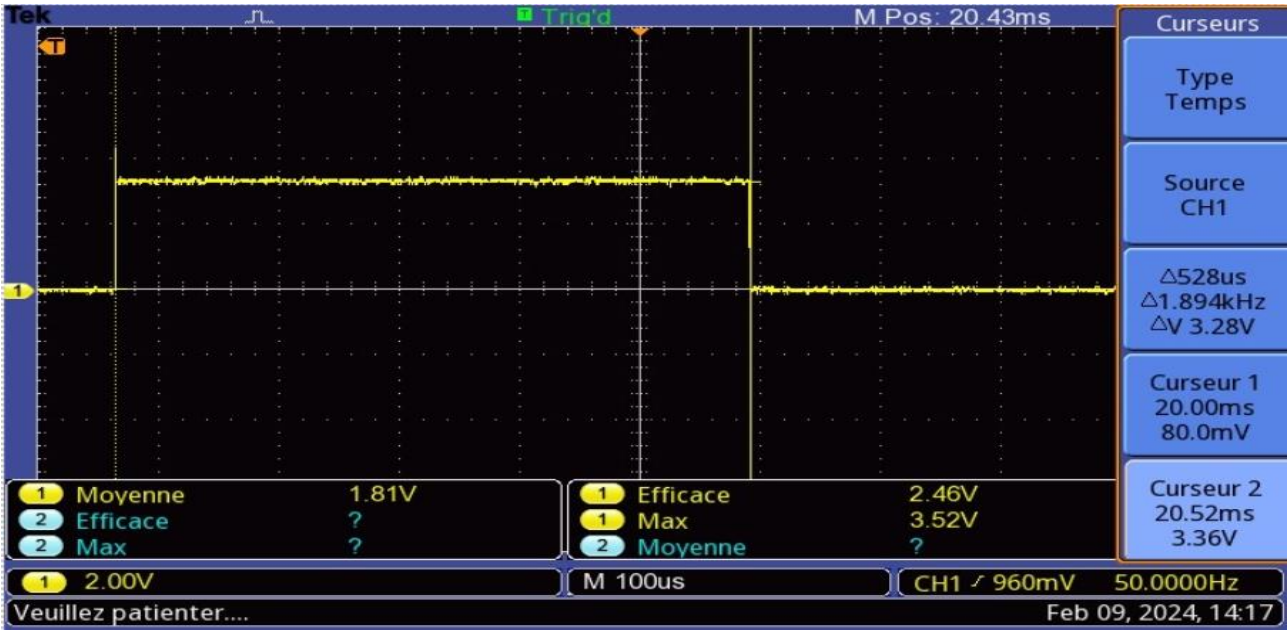
	T (ms)	f (Hz)	Ton (ms)	Rapport cyclique r ($\frac{T_{on}}{T_{off}}$. 100)
0°	20ms	50Hz	528us	<u>2,64</u>
90°	20ms	50Hz	1,46ms	<u>7,3</u>
180°	20ms	50Hz	2,36ms	<u>11,8</u>

On peut remarquer que plus l'angle est grand plus le rapport cyclique augmente.

Moteur à 0°

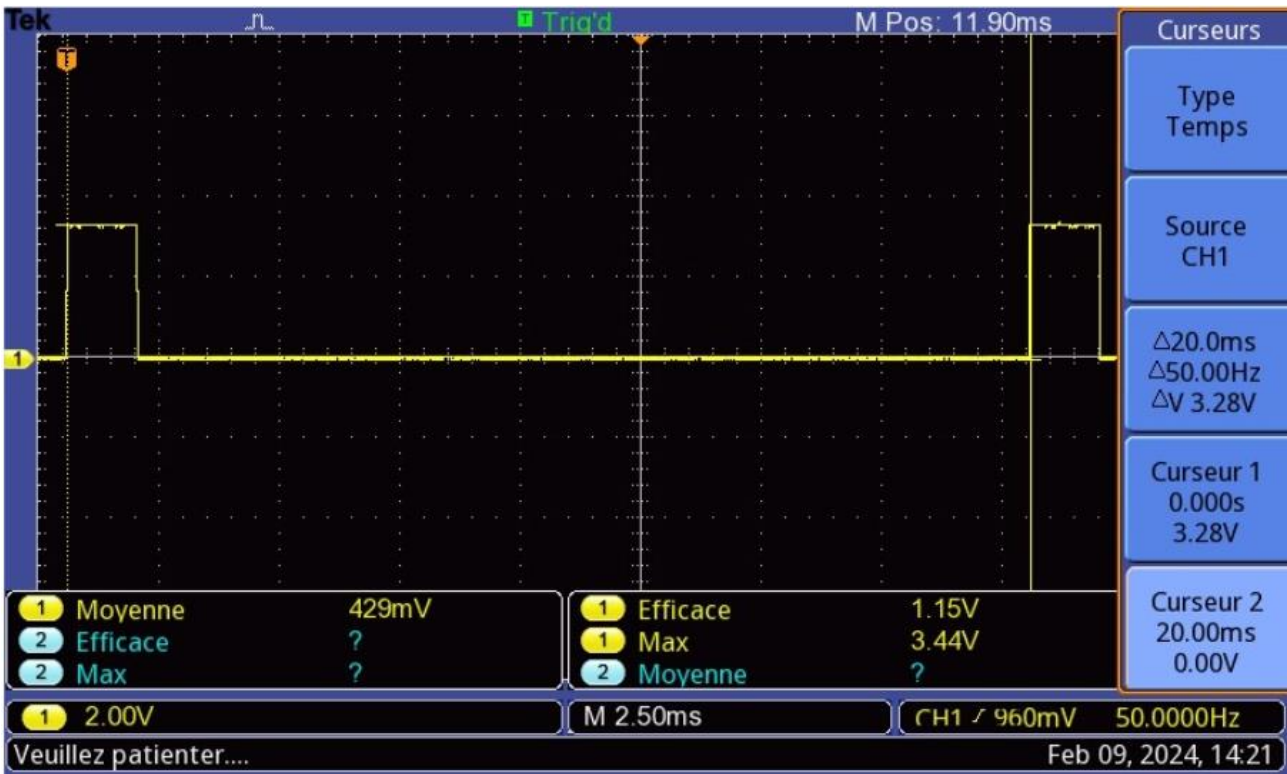
Période

Ton

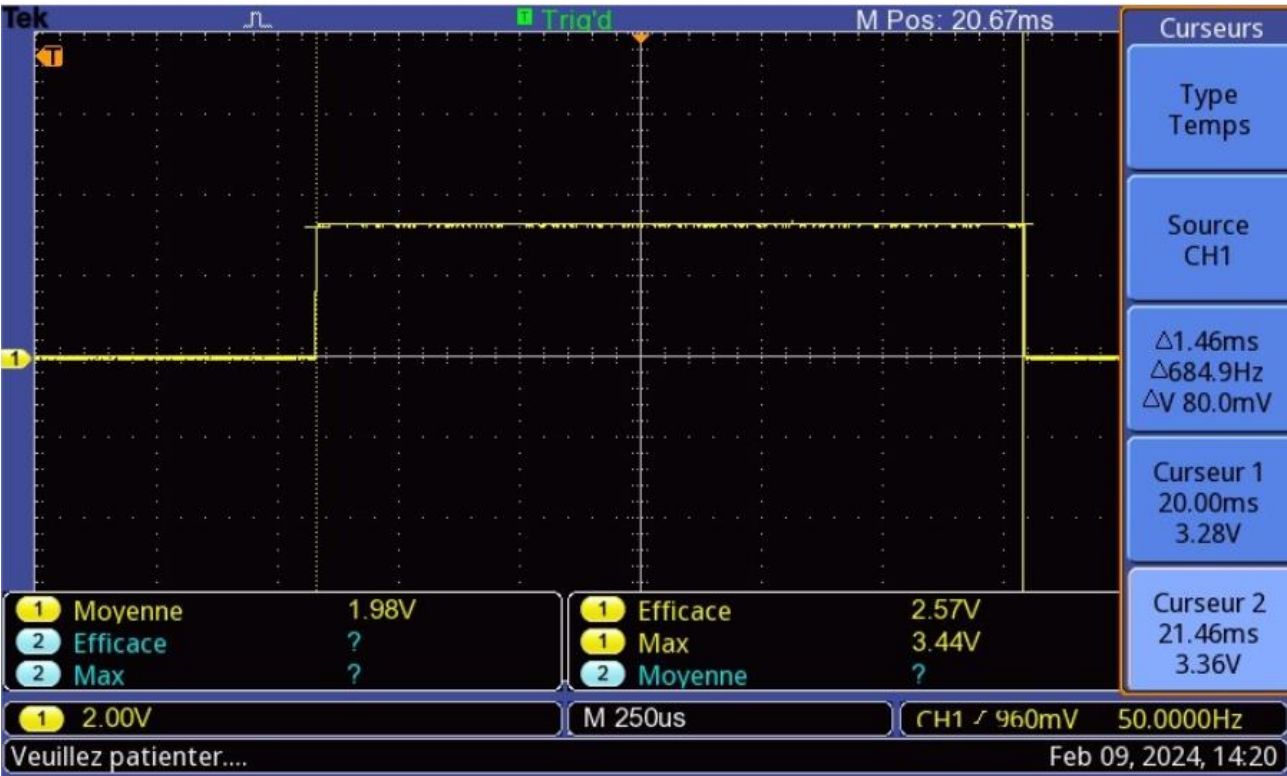


Moteur à 90°

Période

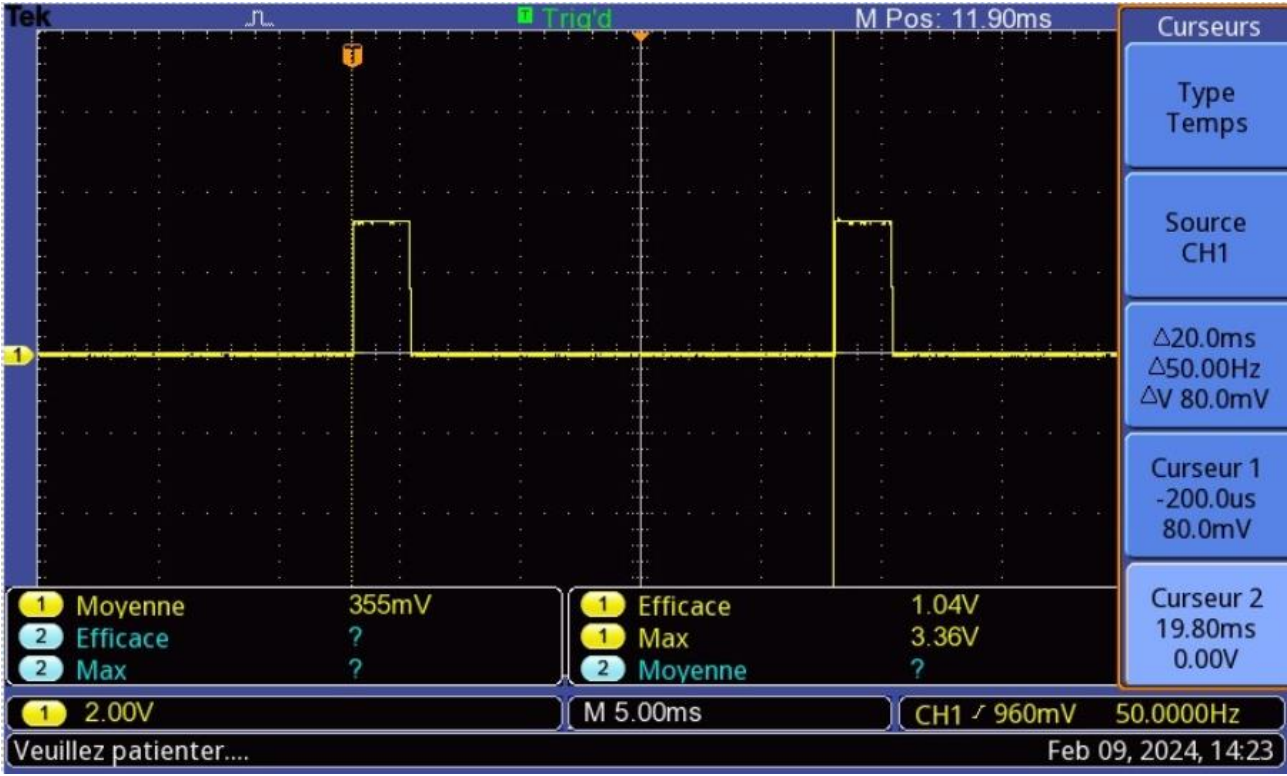


Ton

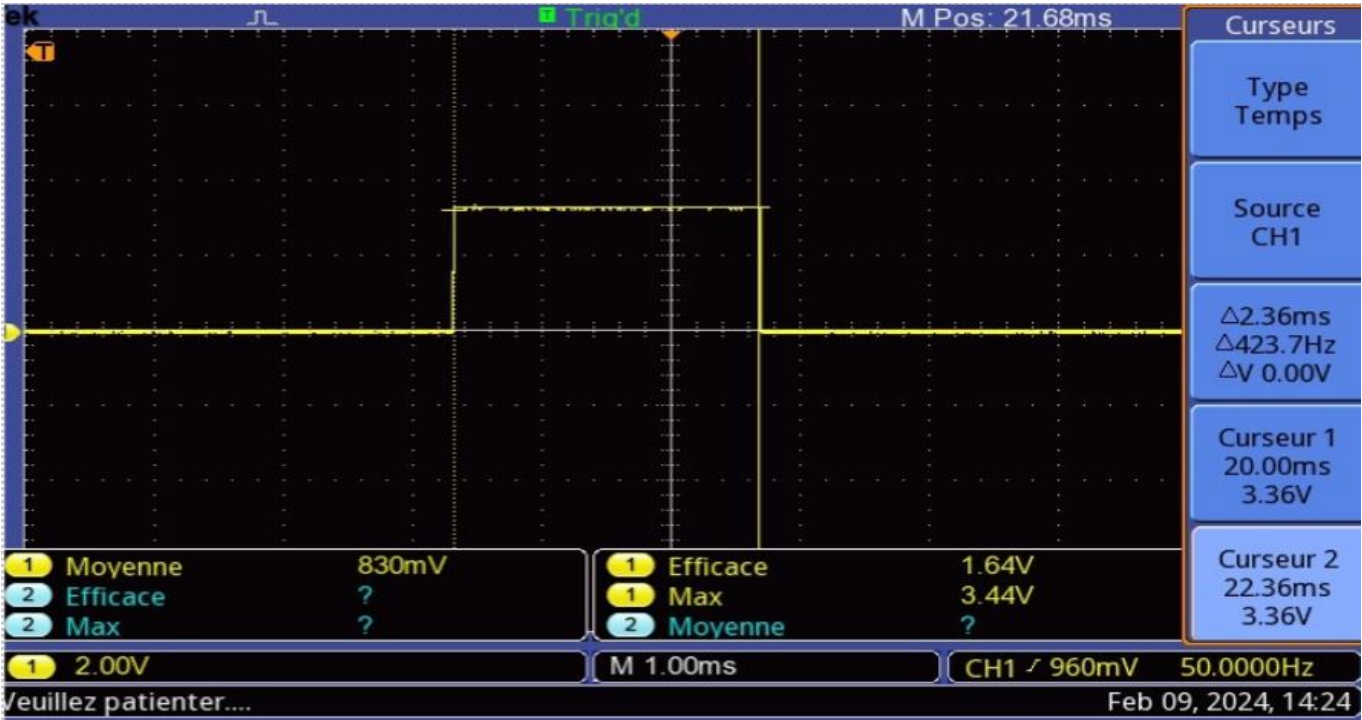


Moteur à 180°

Période



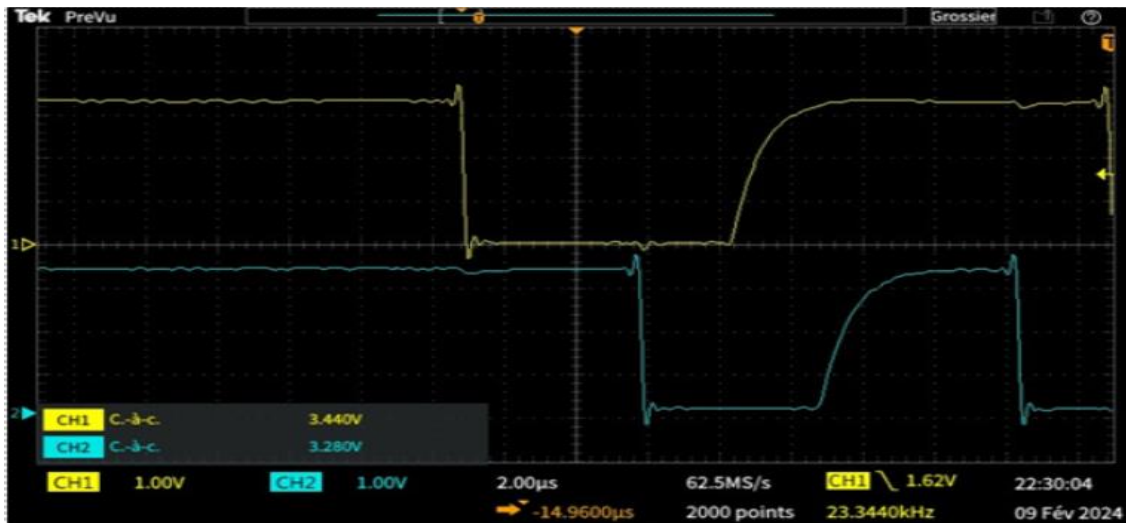
Ton



Mesure sur les capteurs de flexion

	Mesure (volt mètre)	Mesure (Ohm mètre)	Calcul du programme (ADC25ADC25)	Valeur en binaire
Sans flexion	5V	18k Ω	245	6143 0b 0001 0111 1111 1111
Flexion a 90°	2,8V	7k3 Ω	131	4262 0b 0001 0000 1010 0110
Flexion totale	560mV	2k3 Ω	83	489 0b 0001 1110 1001

Bit de start



Flexion totale



Sans flexion



6.La programmation

2. L'ordinogramme

3. Le programme

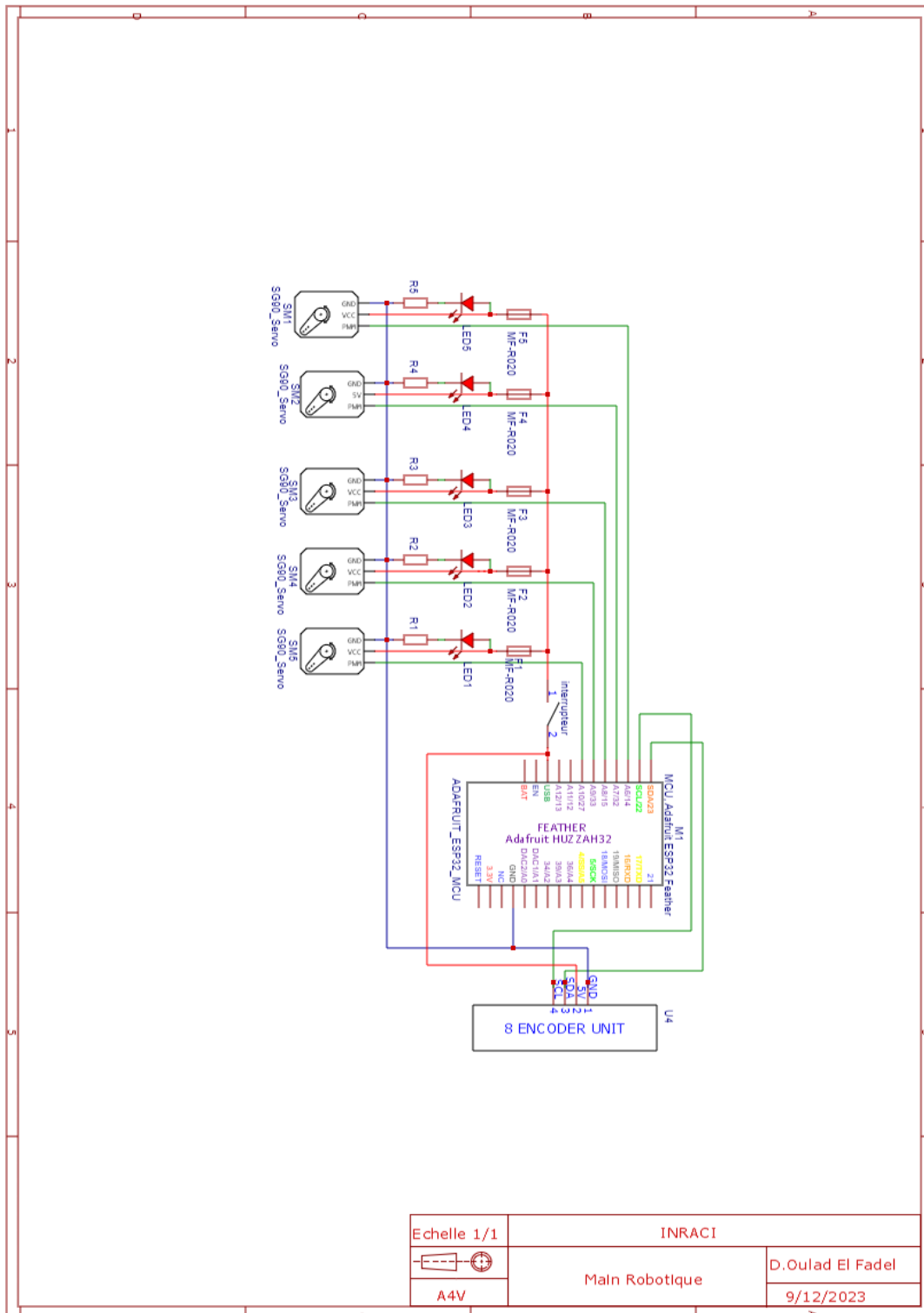
7.La fabrication

8.La mise au point

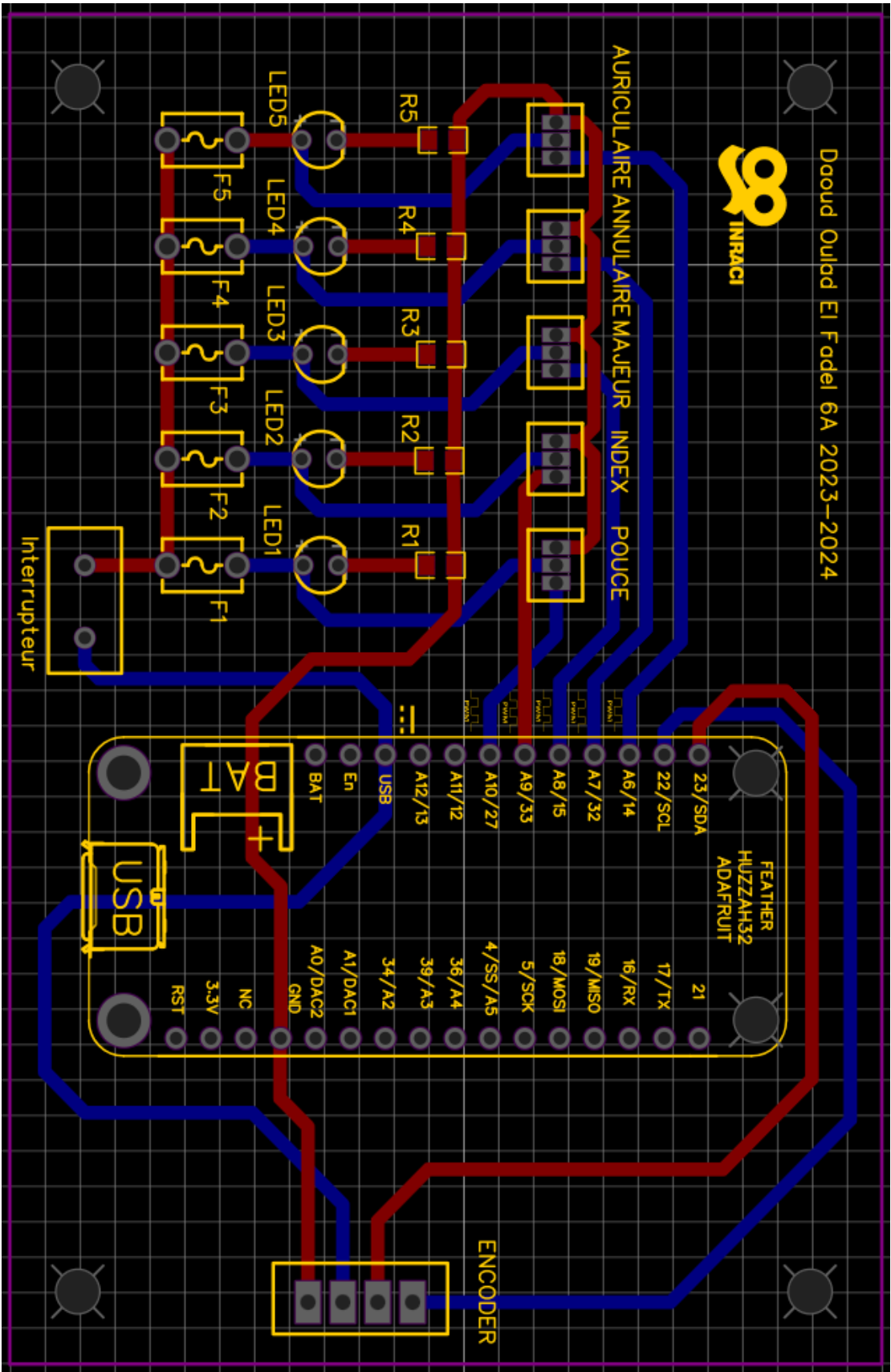
9.La conclusion

10.Les annexes

1.Schéma de principe sans connecteur

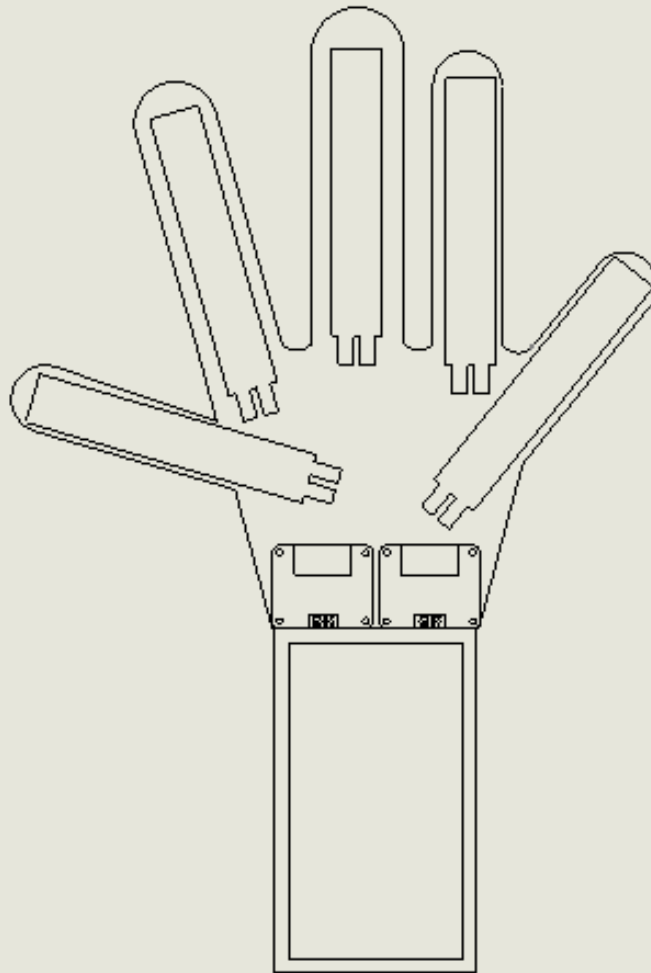


2.Dessin de PCB



11.Vue 3D

Assemblage du gant



Echelle 1/1

INRACI

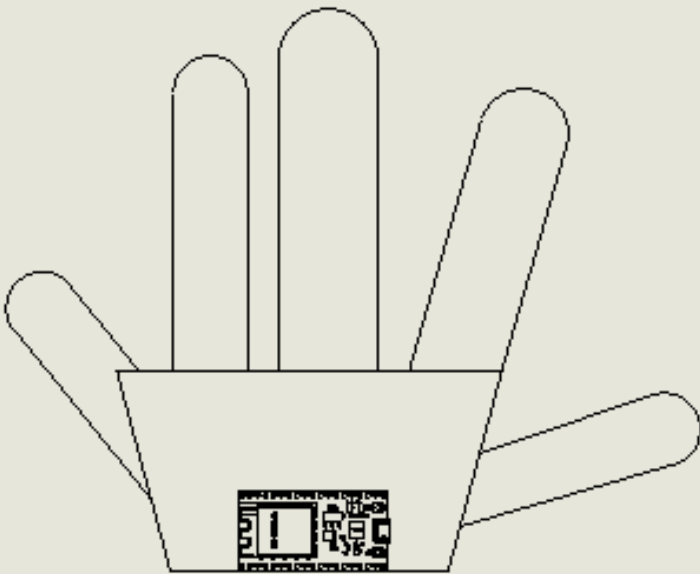


A4V

Assemblage du
gant

Oulad El Fadel
Daoud

16/10/2023



Echelle 1/1

INRA CI




Assemblage du
gant

Oulad El Fadel
Daoud


A4V

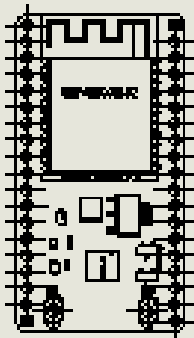
16/10/2023




Echelle 1 / 1	INRACI	
	MOTEUR	Daoud Oulad El Fadel 6A
A4V		15 / 11 / 2023

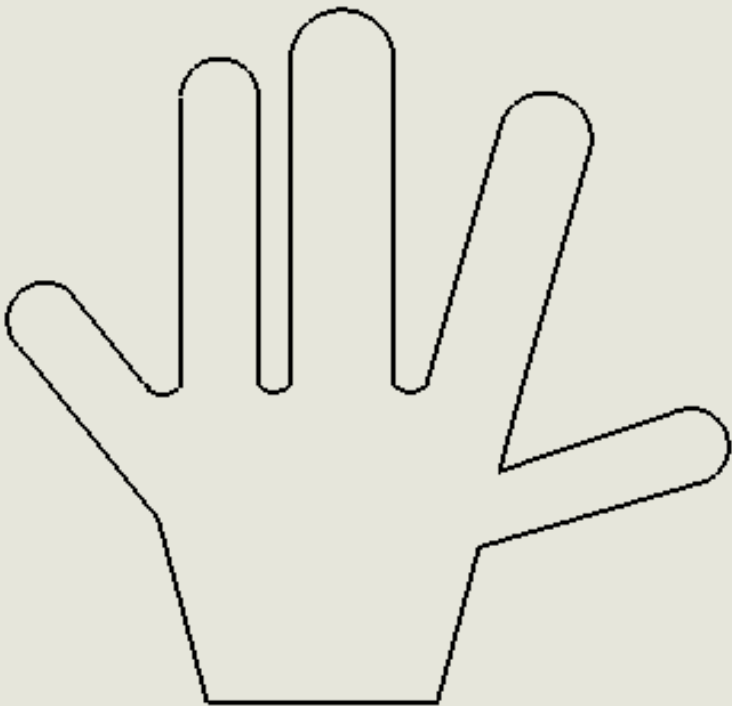


Echelle 1 / 1	INRACI	
	Capteur de flexion	Daoud Oulad El Fadel 6A
A4V		15 / 11 / 2023



Echelle 1/1	INRACI	
	ESP32	Daoud Oulad El Fadel 6A
A4V		15/11/2023

Gant



Echelle 1/2

INRACI



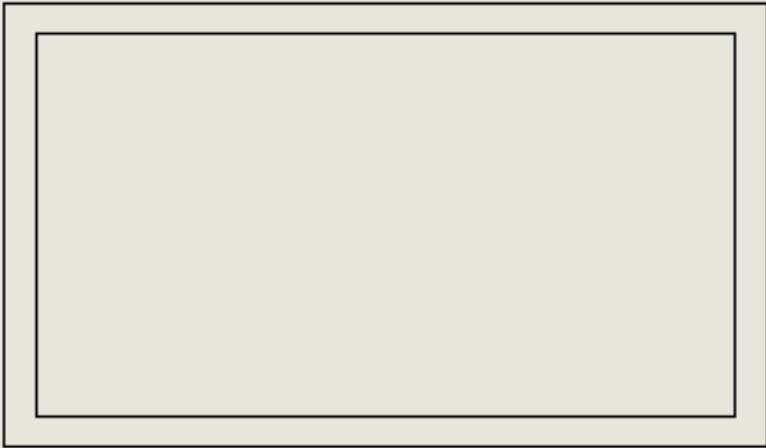
Gant

Daoud Oulad
El Fadel 6A

A4V

15/11/2023

M5 PAPER



Echelle 1/1

INRACI

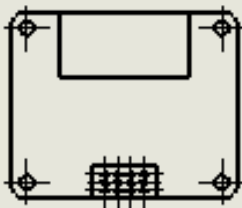



M5 PAPER

Daroud Oulad
El Fadel 6A

A4V


15/11/2023



Echelle 1 /1	INRACI	
	MODULE I2C	Daoud Oulad El Fadel 6A
A4V		15/11/2023

12.Fiche technique des composants

ESP32

Image	
Microcontrôleur	Tensilica LX6 double cœur 32 bits, 240 MHz
Fabricant de microcontrôleurs	Expressif
Système sur puce	ESP32
Tension d'entrée (limite)	Utilisez soit USB (5 V) soit LiPoly (3,7/4,2 V)
Tension de fonctionnement	3,3 V
Mémoire flash	4 Mo
SRAM	520 Ko
Broches GPIO	21
Broches PWM	Tous
Entrées analogiques	14
Wifi	Émetteur-récepteur Wi-Fi 802.11b/g/n HT40
Bluetooth	Mode double (classique et BLE)



M5 PAPER

Resources	Parameter
ESP32-D0WDQ6-V3	240MHz dual core, 600 DMIPS, 520KB SRAM, Wi-Fi
Flash	16MB
PSRAM	8MB
Input Voltage	5V @ 500mA
Ports	TypeC*1, HY2.0-4P*3 , TF-card(microSD) slot
E-Ink Display	Model Number:EPD_ED047TC1 540*960@4.7" Gray scale : 16 Levels Display area : 58.32*103.68mm Display Driver : IT8951E
Physical Button	Multi-function button*1 , Reset Button*1
RTC	BM8563
Antenna	2.4G 3D Antenna
PINS	G25, G32, G26, G33, G18, G19
Battery	1150mAh@3.7V
Working Temp	0°C to 60°C
Net Weight	86g
Gross Weight	100g
Product Dimension	118*66x*10mm
Packaging Dimension	120*70x*14mm
Casing Material	Plastic (PC)

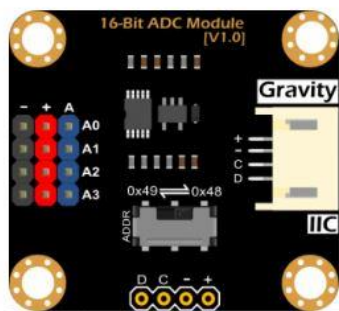


Module ADC

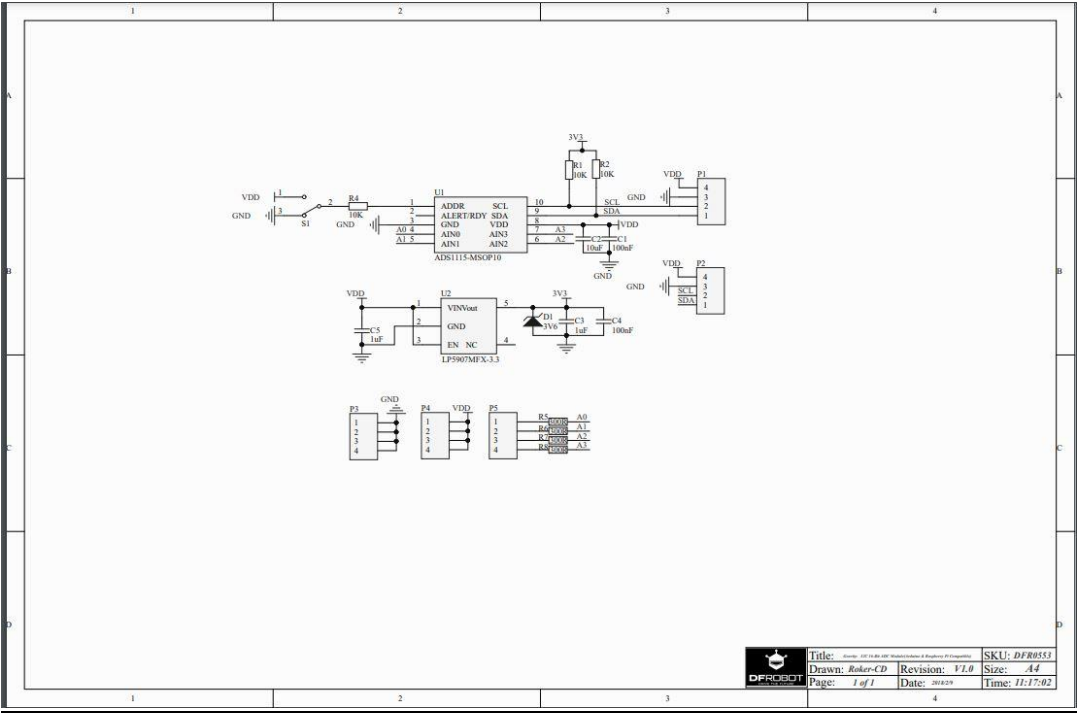
Specification

- Supply Voltage (VCC): 3.3~5.0V
- Analog Signal Detection Range: 0~VCC
- Number of Analog Channels: 4
- ADC Bits: 16 Bit
- Operating Current: 2~3mA (Not include sensor module)
- Interface Type: Gravity I2C
- Interface Level: High 3.3V, Low 0V
- Product Size: 32mm * 32mm(1.26in*1.26in)

Board Overview

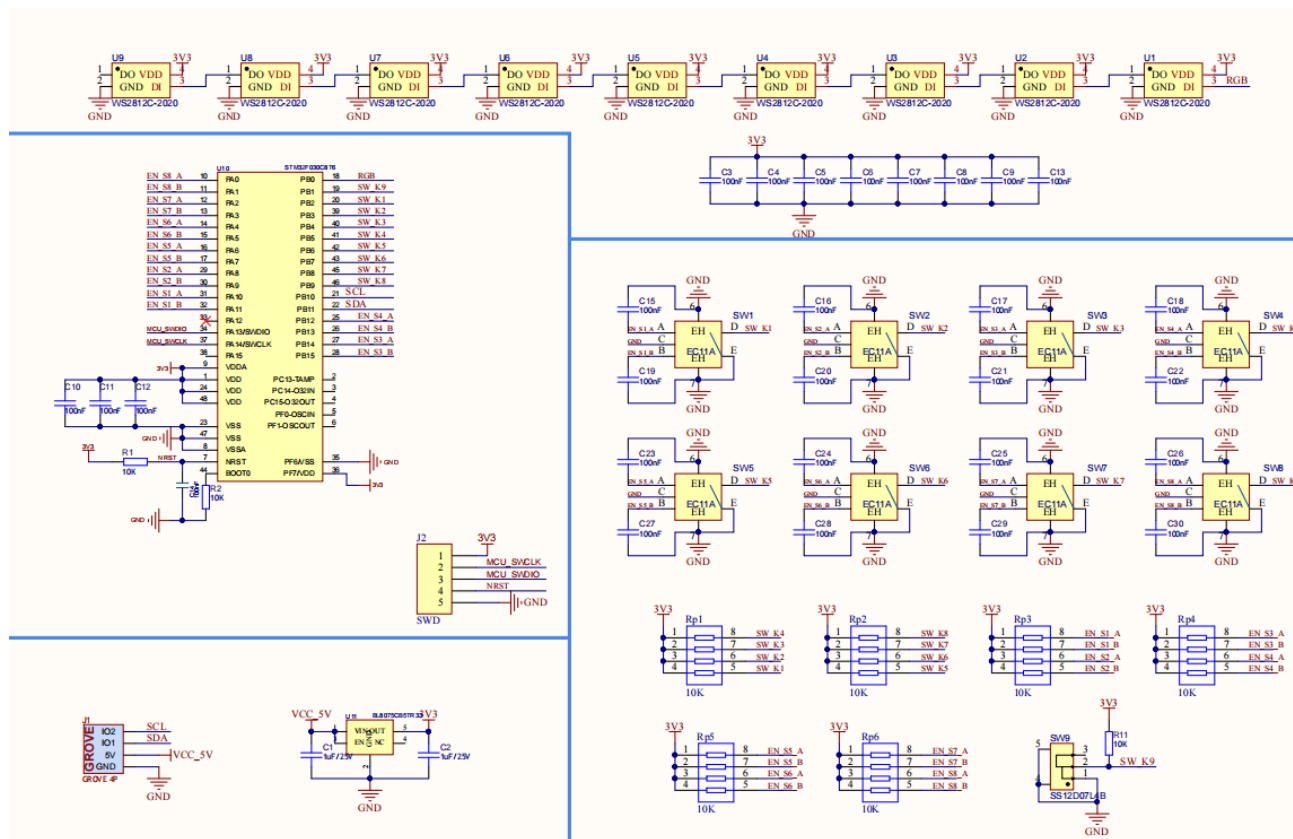


Num	Label	Description
\	VCC	Power VCC(3.3~5.0V)
-	GND	Power GND(0V)
C	SCL	I2C Bus Clock Line
D	SDA	I2C Bus Data Line
A	Analog In	Analog Input Channels: A0, A1, A2, A3
ADDR	I2C Address	I2C address selection switch,I2C address: 0x48, 0x49



Encodeur

Resources	Parameters
MCU	STM32F030C8T6
RGB	WS2812C-2020
Input voltage	5v
I2C communication address	0x41
Product Size	128mm × 24mm × 22.7mm
Package Size	130mm × 27.7mm × 27.7mm
Product Weight	42.8g
Package Weight	52.4g



13.Bibliographie et webographie

Encodeur :

Consulté le 1/02/2024

18h34

<https://www.youtube.com/watch?v=SCR5YJZVyKg>



18h45

<https://www.sick.com/fr/fr/sick-sensor-blog/codeur-incremental-fonctionnement-applications-et-produits-sick/w/blog-incremental-encoder-base/>



18h53

<https://docs.m5stack.com/en/unit/UNIT%20Encoder>



18h12

<https://fortop.be/fr/solutions/produits/codeurs-lineaires/>



19h32

<https://www.motionsquare.fr/differences-codeur-incremental-et-absolu-lequel-choisir-sur-votre-moteur-electrique/>



<https://s32107ac5c40fd523.jimcontent.com/download/version/0/module/14031692523/name/18-Capteurs3.pdf>

Démodulation :

Consulté le 1/02/2024

20h32

<https://elainnovation.com/difference-entre-bluetooth-et-bluetooth-low-energy/>



21h

<https://www.geeksforgeeks.org/what-is-demodulation/>



21h09

<https://polaridad.es/fr/Qu%27est-ce-que-gfsk-et-comment-fonctionne-la-modulation-de-fr%C3%A9quence-gaussienne%C2%A0%3F/>



Introduction :

Consulté le 10/10/2024

Servo moteur

https://paulo.pagesperso-orange.fr/public/robotv1/Servo_9g_Doc_JPW.pdf

18h30



Main robotique

18h 36



<https://it.aliexpress.com/item/1005003241919629.html?spm=a2g0o.cart.0.0.737218fc4AYG1O&mp=1&gatewayAdapt=glo2ita>

Capteur de flexion

19h



https://www.amazon.com.be/Capteur-ZD10-100-Pression-R%C3%A9sistance-Sensible/dp/B08B88W3H5/ref=asc_df_B08B88W3H5/?tag=begogshpaddef-21&linkCode=df0&hvadid=633259889703&hvpos=&hvnetw=g&hvrnd=17429239368253107899&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcm1=&hvlocint=&hvlocphy=9040070&hvtargid=pla-971260925969&psc=1&gclid=CjwKCAjw3oqoBhAjEiwA_UaLtr7Fr5p8I1RYrb0k3yn_AMNLGzHTDylYBshnorRQj4j0dZ7QdjSXpRoCcewQAvD_BwE

Projet

17h30



<https://www.gotronic.fr/blog/articles/controle-dune-main-robotique/>

Esp32

18h



<https://makeabilitylab.github.io/physcomp/esp32/esp32.html>

M5paper

18h20



<https://docs.m5stack.com/en/core/m5paper>