



**Projet IMA4 – P1**

**Manettes pour travaux pratiques**

**Septembre – Décembre 2018**

Quentin Boëns

20 Décembre 2018

# Sommaire

- I. Objectifs
- II. Cahier des charges
- III. Manette 328p
  1. PCB
  2. Programme manette
  3. Programme PC
  4. Tests
- IV. Manette 16u2
  1. PCB
  2. Programme
  3. Tests

# I. Objectifs

## Manette 328p

- Manette pour tutorat IMA4
- Programmation de la manette
- Programme PC

## Manette 16u2

- Manette pour TP GIS3
- Programmation de la manette

## II. Cahier des charges

### Manette 328p

- Comportement périphérique IP
- Communication en UDP
- Renvoyer l'état des composants sur sollicitation du PC

### Points communs

- 10 LEDs et 2 vibreurs actifs
- 5 boutons fonctionnels
- Bouton reset fonctionnel

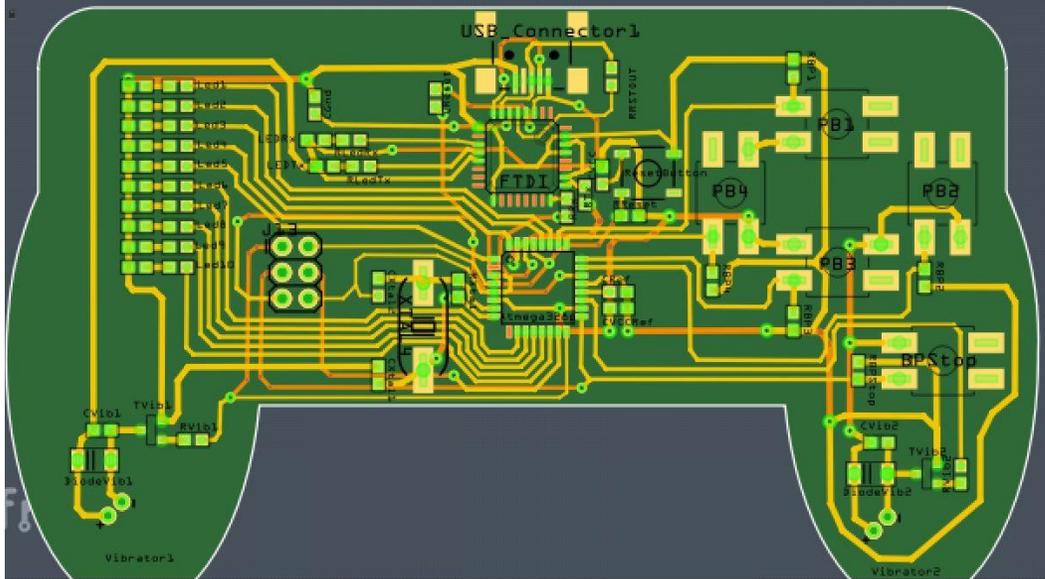
### Manette 16u2

- Comportement périphérique USB, type HID
- Communication via LUFA

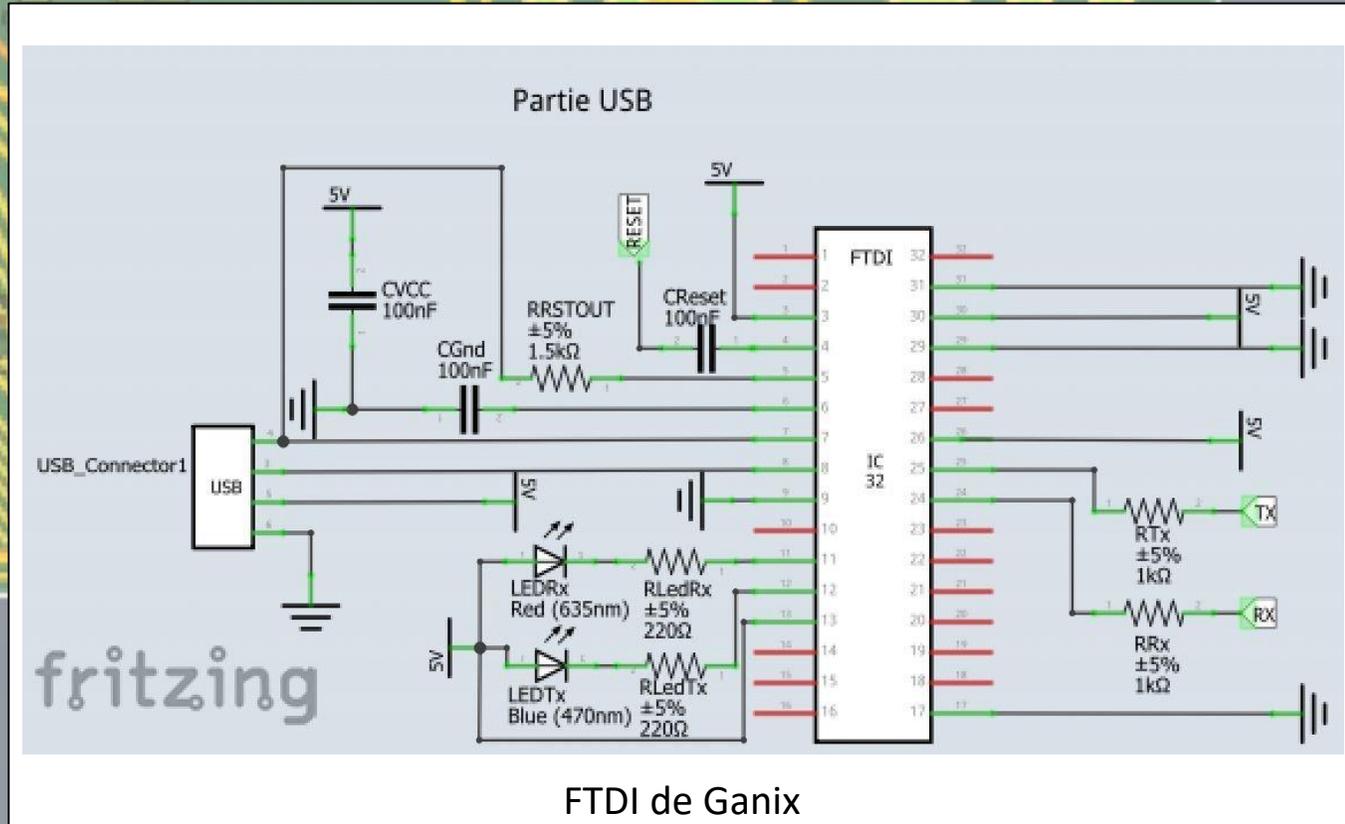
### III. Manette 328p

# III. Manette 328p

## 1. PCB

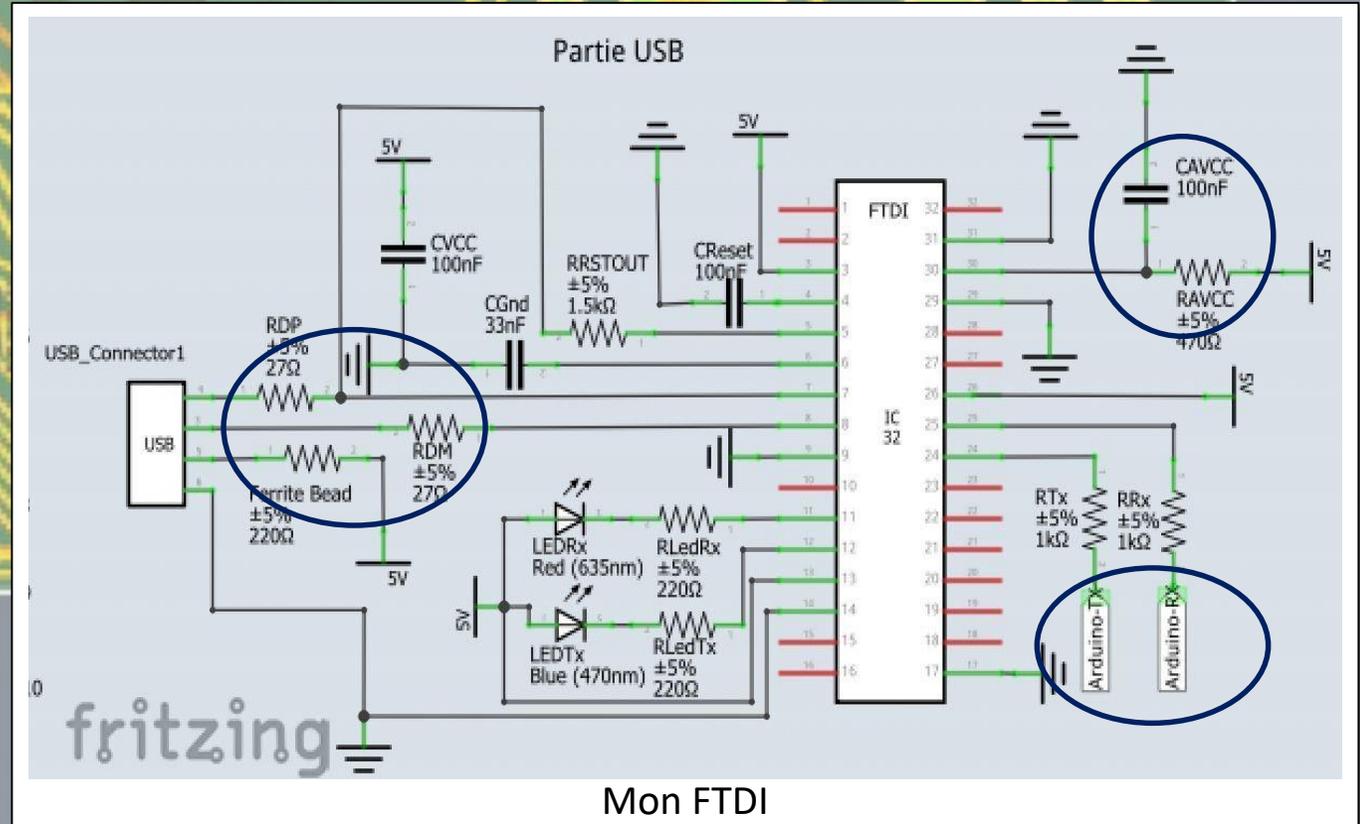
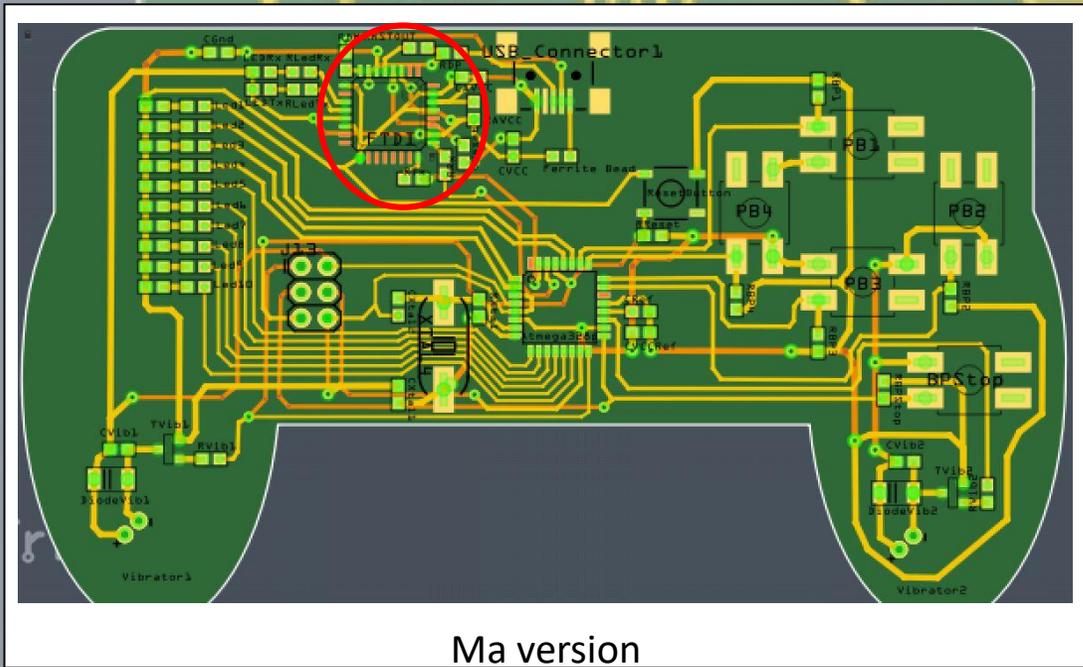


Version de Ganix



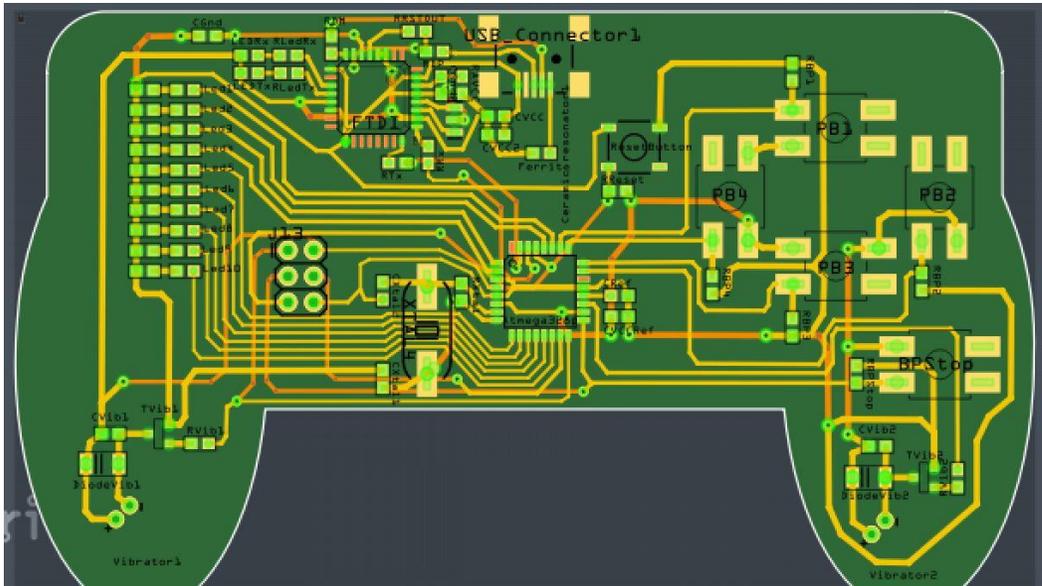
# III. Manette 328p

## 1. PCB

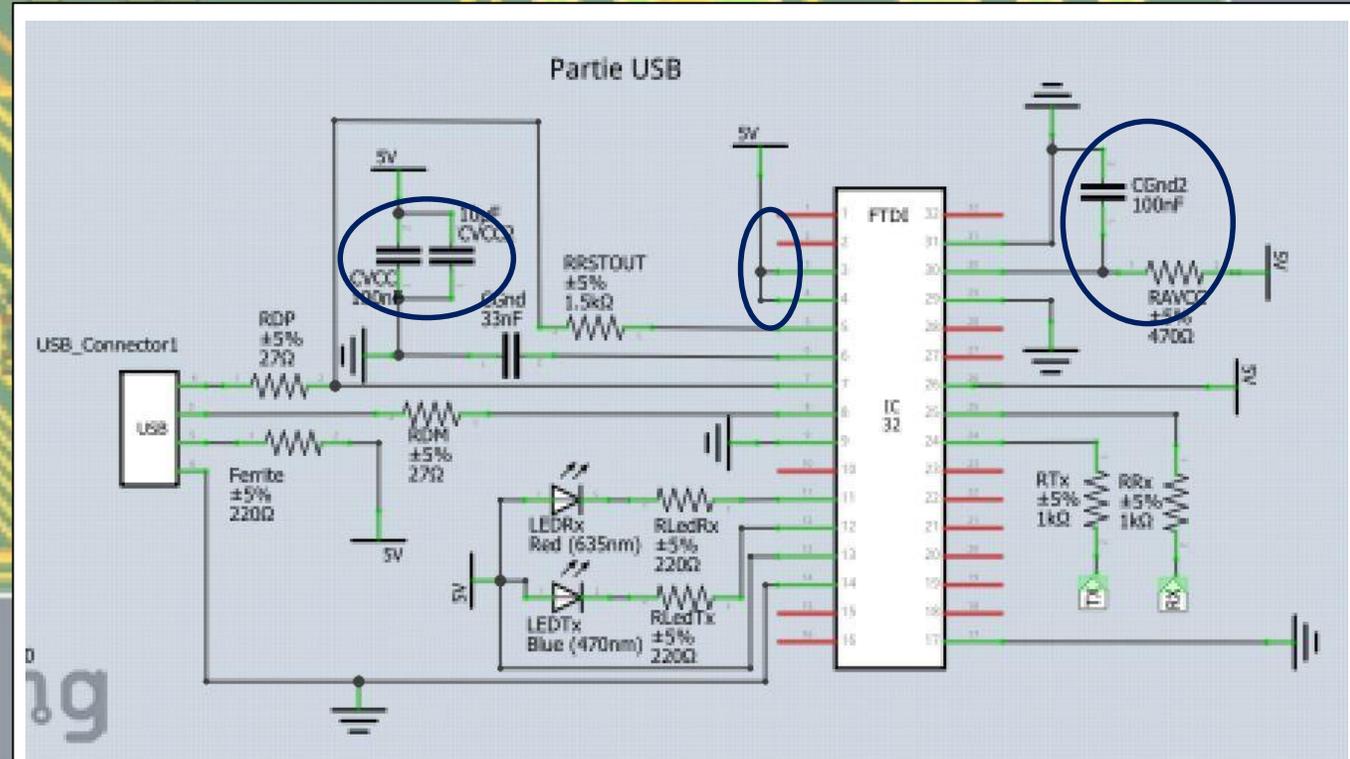


# III. Manette 328p

## 1. PCB

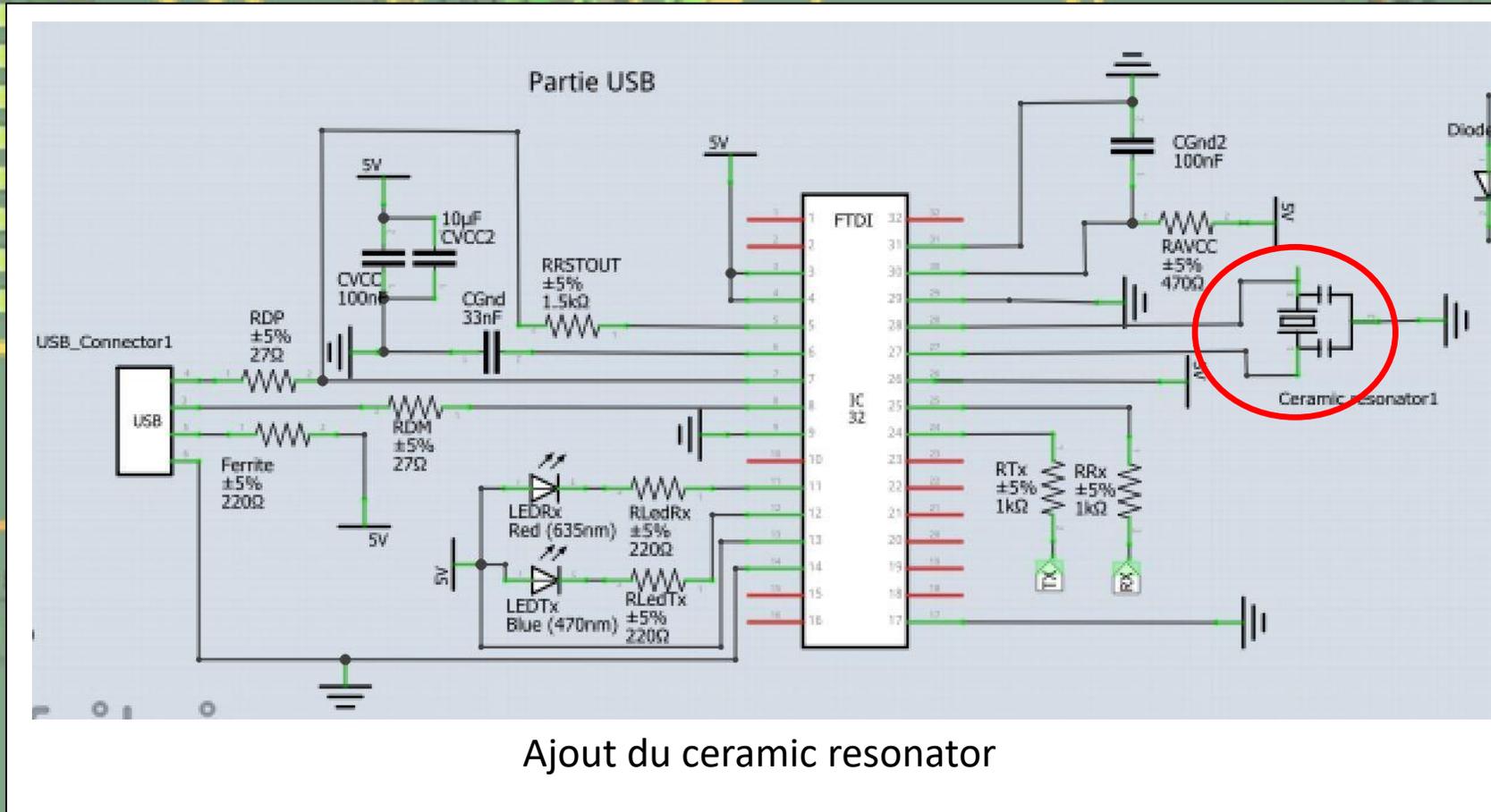


Version de M.Redon



FT232BL de M. Redon

### III. Manette 328p 1. PCB



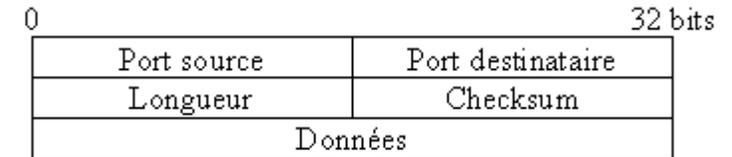
Ajout du ceramic resonator

# III. Manette 328p

## 2. Programme manette

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version d'IP		Longueur de l'en-tête		Type de service				Longueur totale																							
Identification				Indicateur		Fragment offset																									
Durée de vie		Protocole				Somme de contrôle de l'en-tête																									
Adresse source																															
Adresse destination																															
Option(s) + remplissage																															

Format en-tête IP



Format en-tête UDP

```
// Partie IP
///VERSION IP + LONGUEUR EN TETE
ip_udp[0]=0x45; //Version de IP : IPV4
//Sans options l'entete vaut 5 : nombres de lignes de mots de 32 bits

//// SERVICE
ip_udp[1]=0x00; //Type de service
// 0001 1100 : Low Delay, Débit élevé, Très Fiable

//// LONGUEUR TOTALE TRAME
//La trame à une longueur de 31 octets

ip_udp[2]=0x00; //Longueur totale partie 1
ip_udp[3]=0x1F; //Longueur totale partie 2

//// IDENTIFICATION
ip_udp[4]=0x00; //Identification partie 1
ip_udp[5]=0x03; //Identification partie 2

//// FLAGS + FRAGMENT
ip_udp[6]=0x40; //0x40 = 0b01000000 Flags en position 010 : on n'autorise pas la fragmentation
//Les derniers bits sont les premiers de la fragmentation
ip_udp[7]=0x00;
```

Extrait de la trame de données

```
static int send_serial_printf(char c, FILE *stream){
if(c=='\n') send_serial('\r');
send_serial(c);
return 0;
}
```

Fonction d'envoi des données

### III. Manette 328p

## 2. Programme manette

```
// Calcul checksum udp manette
void calcul_checksum_udp(void)
{
    unsigned char check[18];
    uint32_t checksum = 0;
    int i;
    for( i = 0;i<8;i++)
    {
        check[i] = ip_udp[i+12];///@IP source et dest recup
    }

    check[8] = 0x00;
    check[9] = ip_udp[9];///protocole recup

    check[10] = ip_udp[24];
    check[11] = ip_udp[25];///Taille UDP recup

    for(i=12;i<18;i++)
    {
        check[i] = ip_udp[i+8];
    }

    for(i=0;i<9;i++)
    {
        checksum = checksum + ((check[2*i] & 0xFF) << 8) + (check[2*i+1] & 0xFF);
        while(checksum > 0xFFFF)
        {
            checksum = (checksum & 0xFFFF) + ((checksum & 0xFFFF0000)>>16);
        }
    }

    checksum = ~checksum;

    ip_udp[26] = (checksum & 0x0000FF00) >> 8;
    ip_udp[27] = checksum & 0x000000FF;
}
}
```

Checksum UDP

```
void reponse_mess_ordi(int pin[5],int nb_entree)
{
    ip_udp[28] = 0;ip_udp[29] = 0;ip_udp[30] = 0;
    if((mess_recu[28]&0x01)!=0x00) ///Envoi etat des boutons
    {
        int lecture_port = 1;
        if(test_entree(pin[0],lecture_port))
        {
            ip_udp[28] = ip_udp[28] + 0x08;
        }

        lecture_port = 2;
        int num_pin = 1;

        for( num_pin = 1;num_pin<nb_entree;num_pin++)
        {
            if(test_entree(pin[num_pin],lecture_port))
            {
                ip_udp[28] = ip_udp[28] +(pin[num_pin]<<4);
            }
        }

        if((mess_recu[28]&0x04)!=0x00) ///Envoi etat des LEDs
        {
            ip_udp[28] = ip_udp[28]+((PORTC & 0x10)>>4);
            ip_udp[29] = ip_udp[29]+((PORTC & 0x20)<<2) + ((PORTD & 0x08)<<4) +
            ip_udp[30] = ip_udp[30]+((PORTB & 0x04)<<5);
        }

        if((mess_recu[28]&0x02)!=0x00) ///Envoi etat des vibreurs
        {
            ip_udp[28] = ip_udp[28]+((PORTB & 0x30)<<1);
        }
    }
}
```

Réponse aux messages du PC

# III. Manette 328p

## 3. Programme PC

```
// Fonction analysant le message reçu et donnant l'état des boutons
void affichage_etat_boutons()
{
    double num_bouton;
    unsigned char verif;
    for(num_bouton = 1; num_bouton <= 4; num_bouton++)
    {
        verif = (unsigned char)pow(2, 8 - num_bouton);
        if((mess_recu[28] & verif) != 0) printf("Bouton %d allumé\n", (int)num_bouton);
        else printf("Bouton %d éteint\n", (int)num_bouton);
    }
    num_bouton = 5;
    verif = (unsigned char)pow(2, 8 - num_bouton);
    if((mess_recu[28] & verif) != 0) printf("Bouton stop allumé\n");
    else printf("Bouton stop éteint\n");
}
```

Affichage de l'état des boutons

```
// Choix utilisateur
int choix;

printf("Choix des actions :\n");
printf("\t1-Demander et afficher état LEDs\n");
printf("\t2-Demander et afficher état boutons\n");
printf("\t3-Demander et afficher état vibreurs\n");
printf("\t4-Demander et afficher état de tous les composants\n");
printf("\t5-Changer l'état d'une LED\n");
printf("\t6-Quitter\n");
scanf("%d", &choix);
```

Actions possibles pour l'utilisateur

### III. Manette 328p

#### 4. Tests

```
root@boens-Lenovo-ideapad-510S-14ISK:/media/boens/Windows/Users/qboen/Dropbox/5a  
nnee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette1/PC# ./test  
Ouverture ttyACM0 succes  
fd = 3
```

Activation du programme : succès de la connexion avec une Arduino

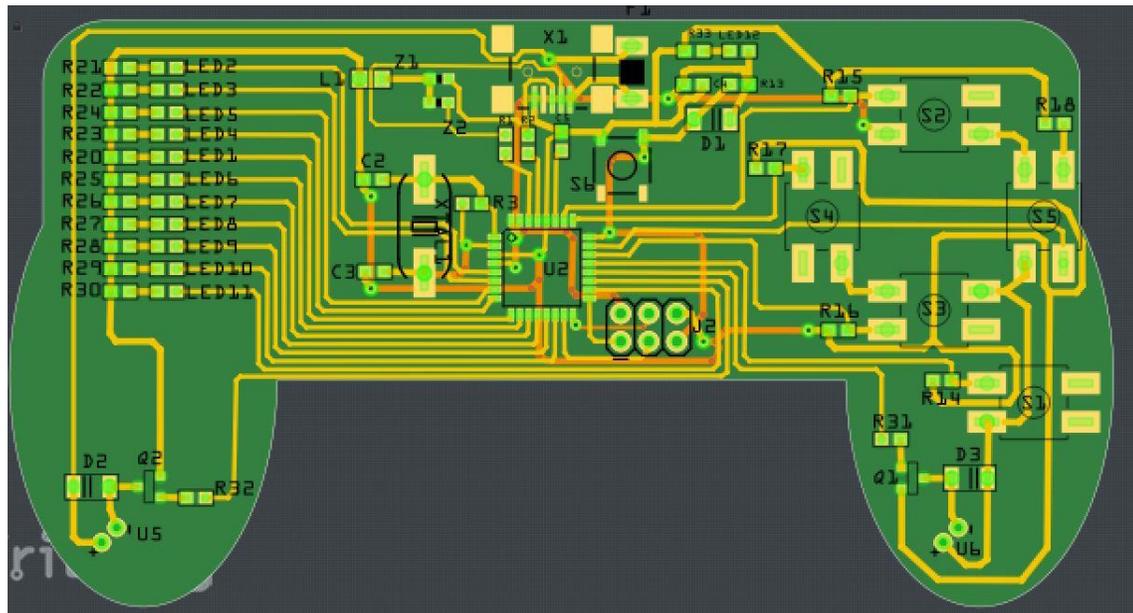
```
root@boens-Lenovo-ideapad-510S-14ISK:/media/boens/Windows/Users/qboen/Dropbox/5a  
nnee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette1/PC# ./test  
Erreur! ouverture ttyACM0  
Veulliez vérifier la connexion de votre manette
```

Activation du programme : échec de la connexion

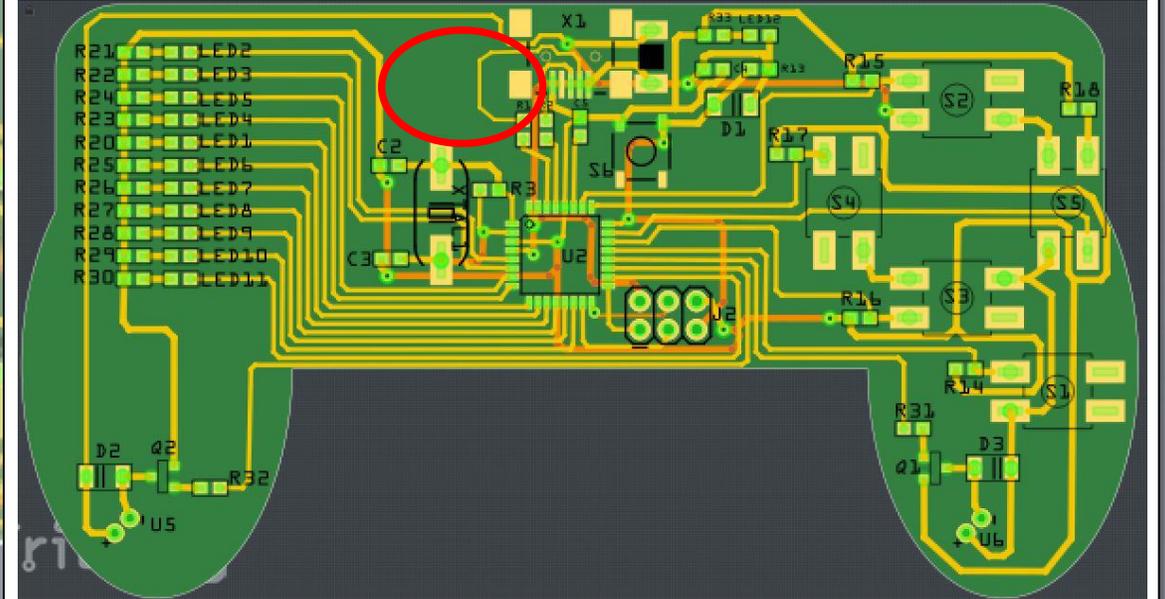
#### IV. Manette 16u2

# IV. Manette 16u2

## 1. PCB



Version de Henri Carlier



Ma version

# IV. Manette 16u2

## 2. Programme

```
bool CALLBACK HID_Device_CreateHIDReport(USB_ClassInfo_HID_Device_t* const HIDInterfaceInfo,
uint8_t* const ReportID,
const uint8_t ReportType,
void* ReportData,
uint16_t* const ReportSize
)
```

```
uint8_t etat_LED[5];
```

Récupération états boutons

```
///Test de l'état des pins reliés aux boutons
etat_LED[0] = ((PINC & 0x10)!=0)?0x01:0x00; // Test du bouton Haut
etat_LED[1] = ((PINC & 0x20)!=0)?0x01:0x00; // Test du bouton Gauche
etat_LED[2] = ((PINC & 0x40)!=0)?0x01:0x00; // Test du bouton Droit
etat_LED[3] = ((PINC & 0x80)!=0)?0x01:0x00; // Test du bouton Bas
etat_LED[4] = ((PINB & 0x80)!=0)?0x01:0x00; // Test du bouton STOP
```

```
// Bouton Haut
if (etat_LED[0] == 0x01) Data[0] = 0x48; // Code ASCII de la lettre H : 0x48

// Bouton Gauche
if (etat_LED[1] == 0x01) Data[1] = 0x47; // Code ASCII de la lettre G : 0x47

// Bouton Droit
if (etat_LED[2] == 0x01) Data[2] = 0x44; // Code ASCII de la lettre D : 0x44

// Bouton Bas
if (etat_LED[3] == 0x01) Data[3] = 0x42; // Code ASCII de la lettre B : 0x42

// Bouton STOP
if (etat_LED[4] == 0x01) Data[4] = 0x53; // Code ASCII de la lettre S : 0x53
```

```
*ReportSize = GENERIC_REPORT_SIZE;
return false;
```

Chargement données

Fonction pour la création des messages

```
void output_input_init(void)
{
  DDRB &= 0x71; //Boutons STOP (IN) + LEDs 10 et 11 (OUT) + Vibreurs (OUT)
  DDRC &= 0x04; //Boutons HAUT GAUCHE DROITE BAS (IN) + LED 1 (OUT)
  DDRD &= 0xFF; //LEDs 2 à 9 (OUT)
}
```

Initialisation des entrées/sorties de l'atmega

```
//ALLUMER
if (action == 1)
{
  if(LED_ETAT[num_LED-1] == 0) // La LED est éteinte donc on change son état
  {
    if(num_LED == 1)
    {
      PORTC = PORTC ^ TAB_LED[num_LED - 1];
      LED_ETAT[num_LED-1] = 1; // Dans le tableau d'état des LEDs on indique que la LED est allumée
      return 0;
    }
    if((1 < num_LED)&&(num_LED<10))
    {
      PORTD = PORTD ^ TAB_LED[num_LED - 1];
      LED_ETAT[num_LED-1] = 1; // Dans le tableau d'état des LEDs on indique que la LED est allumée
      return 0;
    }
  }
  else
  {
    PORTB = PORTB ^ TAB_LED[num_LED - 1];
    LED_ETAT[num_LED-1] = 1;
    return 0;
  }
}
else return 0; // La LED est déjà allumée
}
```

Extrait de la fonction d'activation/désactivation des LEDs

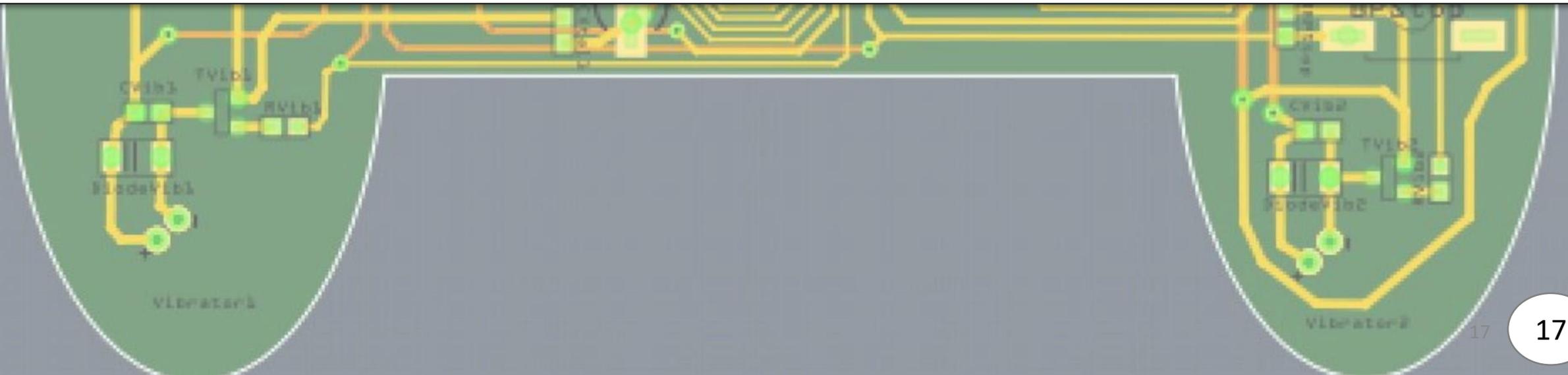
## IV. Manette 16u2

### 2. Programme



```
boens@boens-Lenovo-ideapad-510S-14ISK: /media/boens/Windows/Users/qboen/Dropbox/Sannee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette2/PC/GenericHID_10-12$ sudo dfu-programmer atmega16u2 erase
boens@boens-Lenovo-ideapad-510S-14ISK: /media/boens/Windows/Users/qboen/Dropbox/Sannee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette2/PC/GenericHID_10-12$ sudo dfu-programmer atmega16u2 flash GenericHID.hex
Validating...
4620 bytes used (37.60%)
boens@boens-Lenovo-ideapad-510S-14ISK: /media/boens/Windows/Users/qboen/Dropbox/Sannee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette2/PC/GenericHID_10-12$ sudo dfu-programmer atmega16u2 reset
boens@boens-Lenovo-ideapad-510S-14ISK: /media/boens/Windows/Users/qboen/Dropbox/Sannee/Projet/1-Programmes/Mannettes/Projet_P1_Manette/Manette2/PC/GenericHID_10-12$
```

Installer le programme sur la manette



# IV. Manette 16u2

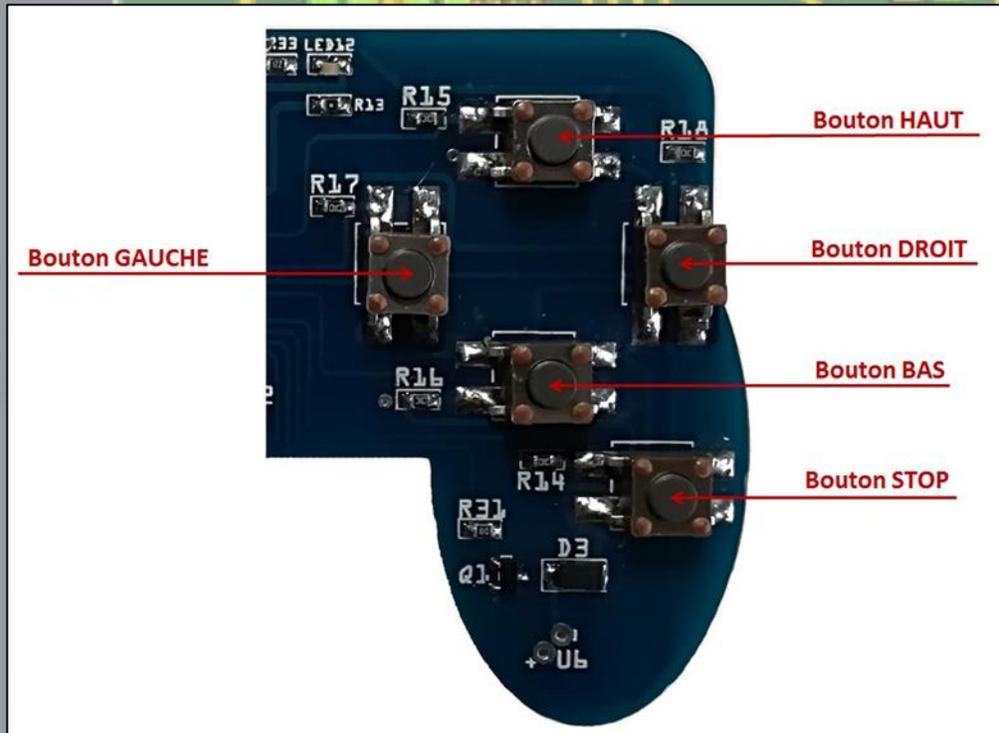
## 3. Tests

```
boens@boens-Lenovo-Ideapad-510S-14ISK:~$ lsusb -D/dev/bus/usb/001/006
Device: ID 03eb:204f Atmel Corp. LUFA Generic HID Demo Application
Couldn't open device, some information will be missing
Device Descriptor:
  bLength                18
  bDescriptorType        1
  bcdUSB                  1.10
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        8
  idVendor                0x03eb Atmel Corp.
  idProduct              0x204f LUFA Generic HID Demo Application
  bcdDevice               0.01
  iManufacturer          1
  iProduct               2
  iSerial                0
  bNumConfigurations     1
Configuration Descriptor:
  bLength                 9
  bDescriptorType        2
  wTotalLength           34
  bNumInterfaces         1
  bConfigurationValue    1
  iConfiguration         0
  bmAttributes           0xc0
    Self Powered
  MaxPower               100mA
Interface Descriptor:
  bLength                 9
  bDescriptorType        4
  bInterfaceNumber       0
  bAlternateSetting      0
  bNumEndpoints          1
  bInterfaceClass        3 Human Interface Device
  bInterfaceSubClass     0 No Subclass
  bInterfaceProtocol     0 None
  iInterface              0
  HID Device Descriptor:
    bLength                9
    bDescriptorType       33
    bcdHID                 1.11
    bCountryCode           0 Not supported
    bNumDescriptors        1
    bDescriptorType        34 Report
    wDescriptorLength      32
  Report Descriptors:
    ** UNAVAILABLE **
Endpoint Descriptor:
  bLength                 7
  bDescriptorType        5
  bEndpointAddress       0x81 EP 1 IN
  bmAttributes           3
    Transfer Type         Interrupt
    Synch Type            None
    Usage Type            Data
  wMaxPacketSize         0x0008 1x 8 bytes
  bInterval              5
```

Récupération des infos

# IV. Manette 16u2

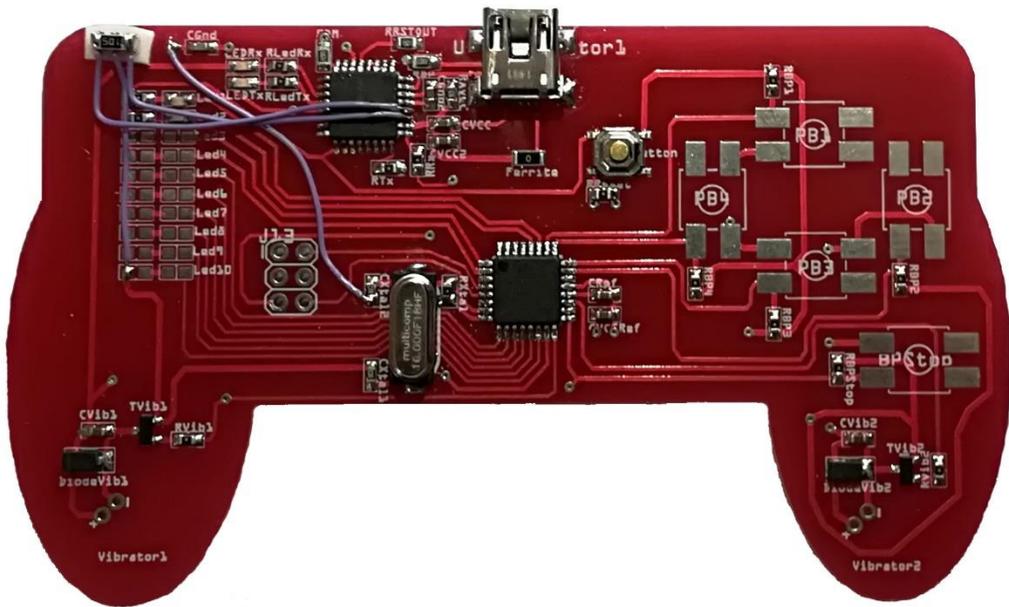
## 3. Tests



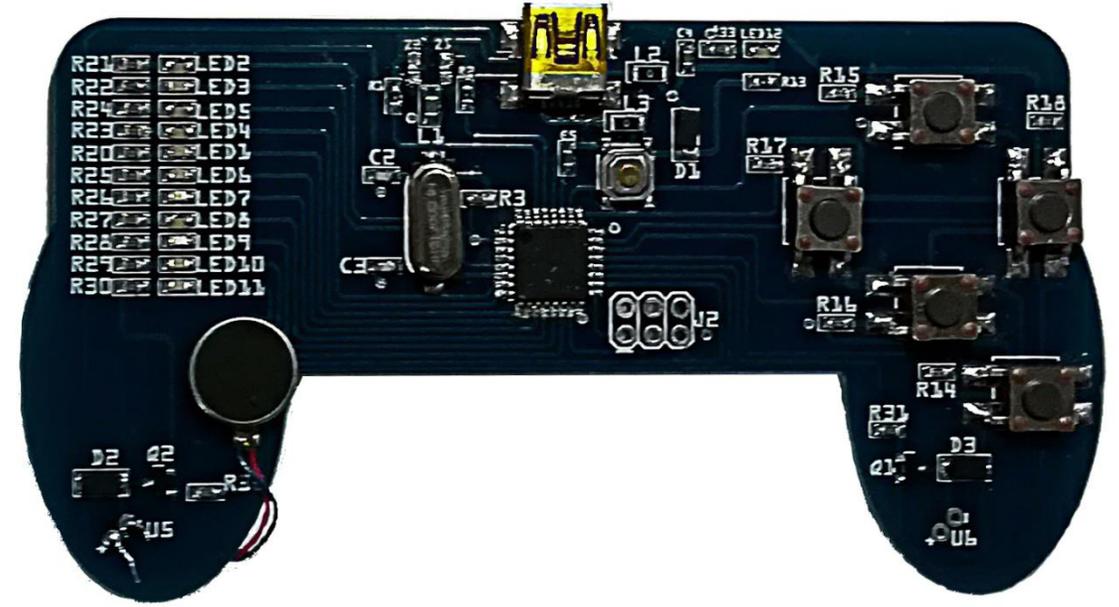
		001:009:000:DESCRIPTOR 1544352474.633172 06 00 FF 09 01 A1 01 09 02 15 00 25 FF 75 08 95 08 81 02 09 03 15 00 25 FF 75 08 95 08 91 02 C0	
		Starting dumping interrupt transfer stream with 1 minute timeout.	
BOUTON HAUT	Appui	001:009:000:STREAM 48 00 00 00 00 00 00 00	1544352479.954023
	Relâchement	001:009:000:STREAM 00 00 00 00 00 00 00 00	1544352480.174096
BOUTON GAUCHE	Appui	001:009:000:STREAM 00 47 00 00 00 00 00 00	1544352482.534135
	Relâchement	001:009:000:STREAM 00 00 00 00 00 00 00 00	1544352482.726134
BOUTON DROIT	Appui	001:009:000:STREAM 00 06 44 00 00 00 00 00	1544352484.414058
	Relâchement	001:009:000:STREAM 00 00 00 00 00 00 00 00	1544352484.618127
BOUTON BAS	Appui	001:009:000:STREAM 00 00 00 42 00 00 00 00	1544352486.110188
	Relâchement	001:009:000:STREAM 00 00 00 00 00 00 00 00	1544352486.274188
BOUTON STOP	Appui	001:009:000:STREAM 00 00 00 00 50 00 00 00	1544352488.334227
	Relâchement	001:009:000:STREAM 00 00 00 00 00 00 00 00	1544352488.518220

Capture des messages

# Conclusion



328p



16u2