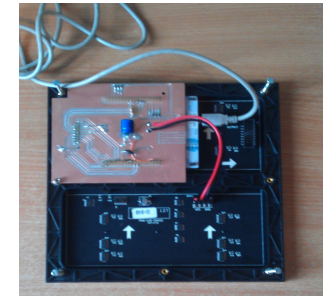
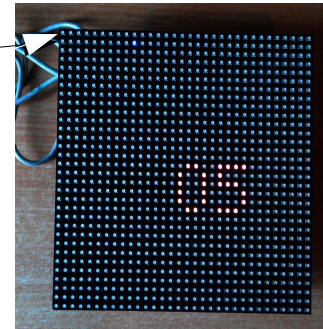


Ici  $x=y=0$



### Mise en œuvre d'un module d'affichage à led .

1024 led RVB sont placées sur une dalle de 20/20cm elles occupent un damier au format 32/32 ( 0 haut à gauche) .. Une carte dite « nucleo401 » munie d'un processeur ARM32 est disposé sur la face arrière du module , elle gère automatiquement le travail nécessaire pour balayer les 1024 points lumineux, (dans l'ordre définit par son constructeur) .. Votre travail consistera à intervenir dans un programme écrit en C++, et y placer les lignes de code nécessaires pour créer un dispositif ludique, comme un ping pong , un snake ou autre jeu simple ;

Ce panneau s'alimente et se programme par le port usb d'un PC .

---

## Mise en Œuvre logicielle

La programmation du panneau se fait par le port USB , il est vu par le PC comme une clef USB, nommée NODE\_F401RE .

### Premier Branchement de la carte : brancher sur le port USB

À l'IUT le PC peut vous demander si vous voulez installer une liaison « ST-LINK » Répondez « NON » .. le PC passe alors en mode USB storage memory le pc testera ce nouveau port à l'antivirus , le panneau s'allume .

### Le Lancement du Logiciel MBED , il se fait sur le CLOUD du fournisseur

1) Logiquement vous disposez d'un compte chez MBED  
« mbed developer » se trouve chez <https://developer.mbed.org/> allez y !!

Créez un compte .. il demande un nom, prénom, et adresse mail .. (c'est gratuit )  
Mettez le nom de login et le mot de passe qui vous conviendra..  
C'est tout ... mais logiquement vous avez déjà un compte

Votre écran sera l'image du travail que vous faites dans un ordinateur distant .

### Attention on ne peut ouvrir qu'une seule fenêtre « compiler » en même temps

**Avant de programmer**, il faut informer votre compte que vous utiliserez la F401RE

1) dans votre compte sur le site MBED  
trouvez l'icone **COMPILER** et cliquez dedans , il ouvre la page du compilateur..  
En haut a droite il vous dit avec quelle carte mbed il est en train de travailler  
Cliquez sur cette icône .. additionnez une carte à partir des images proposées  
Choisissez la carte « nucleo\_f401re » acceptez , il la place dans votre compte  
Ensuite dites lui de l'utiliser (le signe +) .. c'est mieux

2) maintenant c'est le moment de télécharger le répertoire qui sert de base au TR .  
cliquez sur l'icône « import » , ou bien ouvrez Google  
Le répertoire de programme est nommé « **duj401** »  
Une liste apparaît.. cliquez sur « duj401 » ... importez le et mettez un nom  
Après quelques secondes votre répertoire comporte le répertoire de départ ..  
Faites en un clone c'est mieux au cas ou

# C'est bon .. continuons

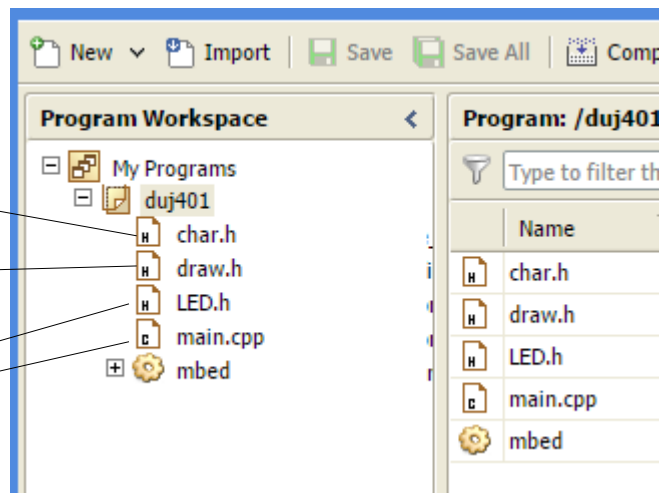
Votre fenêtre de travail présente cet aspect

**Char.h** ..les sous programmes de dessin de caractères .. directement avec des points

**Draw.h** .. la ou vous **devez écrire** vos programmes et vos variables

**LED.h** .. sous programmes de gestion du tableau a led .. fait le travail à votre place

**Main.cpp** .. la racine du programme  
Vous n'écrivez pas dans le main



Ouvrez donc **draw.h**

## C'est votre lieu de travail

Début : lieu pour déclarer vos variables, mettez un **x** comme première lettre ..  
( pour ne pas rentrer en conflit avec mes variables )

Ensuite vous pourrez écrire des sous programmes

Le draw() avec 3 lignes exemple vous y placerez votre code

Rappel des 3 fonctions fournies , placer un pixel et placer du texte avec ces pixels

```
1 // declarez vos variables ICI et les faire commencer par un x
2 int xtiti = 5 ; //exemple
3
4 // lieu de vos fonctions ..qui seront appelées plus bas dans vot
5
6
7 void draw() // boucle répétée change 30 ms
8 {
9     pix(accx, accy, 1, 2, 4);
10    pix(xtiti, 5, 5, 0, 0);
11    ptnum2(10, 10, xtiti);
12
13 }
14
15 // fonctions pré existantes ( utilisant des entiers )
16 // pix ( x , y , r , v , b ) dessine un pixel de couleur en
17 // x et y position du pixel à dessiner ( 0 à 31 )
18 // r v b quantité de 0 à 7 du r v b .. 7 c'est lumir
19 // ...
```

**Un conseil** : le Main exécute les taches d'affichage , il appelle votre draw toutes les 30ms , dans le draw() vous décrivez les pixels à allumer avec la fonction pix(x,y,r,v,b) ; .

Pour ne pas perturber l'affichage .. vous ne pouvez pas mettre de délai dans le draw() ..  
La gestion du draw est comparable avec celle du draw de processing  
Si une fonction de votre draw est mal écrite la compilation plante.  
Si votre draw fonctionne mal et bloque .. c'est l'affichage qui bloque

**ETAPE 1** compilez et testez le programme précédent ( celui que vous venez de télécharger )  
« Compile » Mbed place le fichier créé \*.bin dans la fenêtre téléchargement ( download ) du navigateur CTRL J , déplacez ce bin le vers la carte Nucleo401 visible sur le port USB

Les exemples suivants seront utilisés après

Exemple A programme qui dessine une ligne horizontale rouge de 6 pixels

```
Int xtiti = 0
```

```
Void draw()
```

```
{
  For ( xtiti =0; xtiti< 6; xtiti++)
    {pix ( xtiti , 1 , 5 , 0 , 0 );}
}
```

Exemple B programme qui dessine alternativement un pixel rouge, et un bleu .

```
Int xcompteur = 0 ;
```

```
Void draw()
```

```
{
  xcompteur++;
  If ( xcompteur < 30 ) pix ( 1 , 1 , 5 , 0 , 0 );
  If ( xcompteur > 30 ) pix ( 1 , 1 , 0 , 0 , 5 );
  If ( xcompteur > 60 ) xcompteur = 0 ;
}
```

# Travail à faire

## Rappels : À chaque instant il faut avoir

- 1) Connecté la carte sur le port usb du pc  
Avoir refusé l'installation demandée par le pc  
Avoir observé que l'antivirus s'amuse à tester votre carte  
....Votre carte s'allume .. elle montre un pixel bleu , et un joli 05 en rouge .
- 2) être sur internet sur le site mbed developer  
Avoir ouvert son compte ..  
Avoir ouvert la page compiler dans mbed  
Avoir ouvert le fichier draw.h du compte duj401 , ou d'une de ses copies..

## Exercices à faire dans la séance 1

- 1) modifiez le contenu de xtitl ..... mettez 20 !  
cliquez sur l'icône 'compile'  
10 à 15 secondes de compilation ( ou plus) sont nécessaires au site mbed  
à partir de votre travail situé dans « leur serveur » pour créer un fichier duj401xx.bin  
Mbed vous le téléchargera  
Votre PC vous demande « qu'en faire »  
Enregistrez le dans la carte ( port USB nucleo401 )  
La led bleu change de place ... c'est magique
- 2) remplacez le contenu du draw  
Par l'exemple A page d'avant ( le copier coller pdf vers mbed plante )
- 3) maintenant faites une ligne de 10 pixels bleus  
Au milieu du tableau d'affichage
- 4) dessinez un carré vert vide .. centré .. de côté 8 pixels
- 5) dessinez un carré bleu rempli de bleu centré .. de côté 8 pixels
- 6) créez un fonction `rec(x,y,xdx,xdy , r,v, b )` dessin rectangle vide  
x et y position haut gauche du rectangle  
xdx xdy largeur et hauteur  
r v b la couleur du trait
- 7) créez un fonction `recfill(x,y,dx,dy , r,v, b )` dessin rectangle plein  
x et y position haut gauche du rectangle  
Dx dy largeur et hauteur  
R v b la couleur du rectangle plein et du trait .
- 8) à l'aide du l'exemple B page d'avant ( alternative pixel )  
Faites un carré rouge .. suivi d'un vert
- 9) faites un carré qui passe graduellement du rouge au vert
- 10) faites un sous programme dit `sprite(x,y)`  
Qui dessine en x et y un dessin de votre composition  
Dessin d'au moins 10 pixels de surface
- 11) faites bouger le sprite comme une balle de ping pong
- 12) avez vous vu la ligne `pix(accx,accy,1,2,4)` elle place un pixel à une position influencée  
par la position de la plaque, accrochez votre sprite à la position accx et accy ( position  
rafraîchie toutes les 30 ms)

**C'est bon Vous voilà bien préparés pour la suite ( fin de séance1 ou séance 2 )**

**OBJECTIF DU TR . Faites soit un ping pong ou un autre jeu de votre idée ( snake , voiture, astéroïde ... ) lâchez vous**

Vous désirez simuler des manières de dessiner ??

Voilà un programme qui fonctionne sous processing

Faites un copier coller ! ..mais pas d'ici téléchargez le fichier texte led401.txt

```
// simulation tableau LED401
// mes variables
int cr ; int cb ; int cv ;

// vos variables .. avec un x devant
int xi ;

// mes sous programmes ...
void couleur(int r , int v , int b )
{
  cr = 32 * r ; cb = 32 * b ; cv = 32 * v ;
}

void pix( int x , int y , int r , int v , int b )
{
  couleur ( r , v , b );
  fill ( cr , cv , cb ) ;
  stroke(100);
  rect((x*20)+5,(y*20)+2,16,16);
}

void ledN()
{
  background(0);
  stroke(100);
  fill (50);
  for ( int i = 0 ; i < 32 ; i++)
  {
    for ( int j = 0 ; j < 32 ; j++)
    {
      rect((i*20)+5,(j*20)+2,16,16);
    }
  }
}

// vos sous programmes ( nom avec x )

void setup() {
  size(644, 644);
  noStroke();
  background(0);
}

void draw()
{
  ledN(); // obligatoire .. efface tout comme le led 401
  // exemple
  for ( xi = 3 ; xi < 15 ; xi++ )pix (xi,10,7,0,0);
}
```

Et hop