

Utilisation des #include sous Arduino

Ddr - Fichiers de référence

Pour éviter de retrouver dans les programmes de test successifs les mêmes groupes d'instructions (définitions, fonctions), les 'idée est d'avoir dans un fichier les définitions et procédures que l'on retrouve dans les fichiers que l'on teste.

A partir des fichiers .h on peut faire du copié-collé, ou utiliser des #include "xx" en respectant quelques règles. En particulier, si des fonctions arduino sont utilisées, il faut (1ere fichier seulement) ajouter un #include <Arduino.h>

Cette approche n'est pas du tout celle d'Arduino, qui demande de créer des bibliothèques C++ .

Librairie simple de #include

On peut regrouper dans le dossier (sketch) du programme xx.ino, les fichiers yy.h appelés.

En passant pas l'onglet Sketch, on Add file les fichiers .h cités par des #include "file" dans le programme principal.

Exemple



```
// DdrMotMousTest.ino
#define Led 13
#include "DdrMotMousDef.h"
#include "DdrMotPfmFonc.h"
void setup ()
{
  pinMode(Led, OUTPUT);
  #include "DdrMotMousSetup.h"
}

void loop ()
{
  digitalWrite(Led,HIGH);
  Avance (50);
}
```

// DdrMotMousTest.ino	// DdrMotMousDef.h	//DdrMotPfmFonc.h	// DdrMotMousSetup.h
#include "DdrMotMousDef.h"	#include <Arduino.h>	void Stop ()	pinMode(RecG,OUTPUT);
#include "DdrMotPfmFonc.h"		{	pinMode(AvG,OUTPUT);
void setup ()	#define RecG 4	analogWrite(AvG, 0);	pinMode(AvD,OUTPUT);
{	#define AvG 5	digitalWrite(RecG, LOW);	pinMode(RecD,OUTPUT);
#include "DdrMotMousSetup.h"	#define AvD 6	analogWrite(AvD, 0);	// moteurs arrêts
// autres set-up et initialisations	#define RecD 7	digitalWrite(RecD, LOW);	digitalWrite(RecG, LOW);
}		}	digitalWrite(RecD, LOW);
void loop ()	#define MousD 16	void Avance (byte vv) // 0-255	analogWrite(AvG,0);
{	#define MousG 17	{	analogWrite(AvD,0);
Avance (150);	#define LDRDiff 14	analogWrite(AvG, vv);	//Les pins 14-19 sont
delay(500);	// diff tension D - G	digitalWrite(RecG, LOW);	//analogiques par défaut
Reculé (100);	#define LDRIntens 15	analogWrite(AvD, vv);	
delay(500);		digitalWrite(RecD, LOW);	
}		}	
		...	

En définissant les entrées/sorties à partir des registres et bits, on est plus proches du C et plus compact. L'avantage de cette décomposition avec des fichiers "include" est que le programme principal est indépendant du compilateur (ce n'est pas le cas ci-dessus).