



## Une approche motivante pour s'initier avec Arduino

L'atelier du 27 décembre a rassemblé 6 jeunes de 12 à 14 ans et un enseignant pour monter une base de robot sur laquelle une carte Diduino (Arduino avec Proto-shield et Motor-shield) était posée. La construction et le soudage des connecteurs a pris 3 heures le matin, la programmation 3h30 l'après-midi. La documentation peut se trouver sous [www.didel.com/AtelierRobot27Dec2012.pdf](http://www.didel.com/AtelierRobot27Dec2012.pdf) mais le présent document cible mieux la partie pérenne.

Le but de l'atelier n'était pas d'expliquer la programmation, mais de donner l'envie de l'apprendre et l'appliquer sur un robot qu'ils peuvent compléter.

On a donc demandé aux jeunes de charger les programme, les exécuter, comprendre quelques éléments et modifier ou compléter les programmes selon leurs idées (paramètres, coupé-collé pour répéter des séquences en adaptant un peu).

Donc assez peu d'explications, le but étant d'aller vers des comportements attractifs en montrant plus les fonctions à disposition que la structure du C.

Et le moins de câblage possible pour éviter les problèmes sur le bloc d'expérimentation.

Note de juillet 2014: La commande de moteurs utilisait le miniMotorshield

[www.didel.com/diduino/MsMot.pdf](http://www.didel.com/diduino/MsMot.pdf). Le Xbot reprend exactement le concept. et rien à changer pour les moteurs. Par contre les moustaches sont en 2 et 3, et pas en 8 et 9. La programmation "Arduino" est préférée par des enseignants qui ne veulent pas aller trop loin dans le C.

Les programmes se trouvent sous <http://www.didel.com/diduino/Initiation.zip>

Pour charger Arduino <http://www.didel.com/diduino/Start.pdf> et pour apprendre avec une orientation C, voir [www.didel.com/xbot/DiduiBotStart.pdf](http://www.didel.com/xbot/DiduiBotStart.pdf)

### Programme 1 Clignote

<p>Le programme traditionnel pour expliquer qu'il faut déclarer les objets, configurer (set-up) et écrire la boucle de programme. La Led sur la pin13 est standard sur les cartes Arduino</p>	<pre>// Clignote.ino #define Led 13 #define Allume LOW #define Eteint HIGH int duree = 500 ;  void setup () {   pinMode(Led, OUTPUT); }</pre>	<pre>void loop () {   digitalWrite(Led,Allume);   delay(duree); // 500 ms   digitalWrite(Led,Eteint);   delay(duree); }</pre>
---	---	---

### Programme 2 Pousoir

<p>Le programme simple pour lire un poussoir. Il faut câbler le poussoir entre la broches 9 et le 0V (Gnd). Il faut expliquer qu'une résistance "pull-up" est nécessaire pour</p>		<pre>//Poussoir - Copie un poussoir sur une Led // Le poussoir sur pin9 est cable vers le 0 avec une pull-up. pin LOW si actif // La Led sur pin13 est cablee vers le 0V. HIGH allume  const int Led13 = 13; // HIGH allume const int Pous9 = 9; // LOW si presse  int etatPous9 ;</pre>	<pre>void setup () {   pinMode (Led13,OUTPUT);   pinMode (Pous9,INPUT);   digitalWrite (Pous9,HIGH); // câble une pull-up interne }  void loop () {   etatPous9 = digitalRead (Pous9) ; // on lit le poussoir   digitalWrite (Led13, etatPous9) ; // on copie ce que l'on a lu }</pre>
<p>imposer l'état 1 quand le poussoir ne force pas l'état 0. Une instruction configure cette résistance dans le processeur, pas besoin de la câbler</p>			

### Programme 3 Réflexe

<p>Un programme qui montre l'existence du</p>	<pre>//reflexe.ino #define Led 13</pre>	<pre>void loop () {   // on attend un temps aléatoire 1-5</pre>
---	---	---

<p>terminal série pour afficher une variable, et de la fonction random et son intervalle de valeurs. Ces valeurs, des nombres, sont utilisées dans delay(); comme des durées. Si on presse en permanence, la durée est zéro. Comment corriger le programme ? (attendre que l'on relâche).</p>	<pre>#define Pous9 9  int cnt =0; long alea ;  void setup() {   pinMode(Led, OUTPUT);   pinMode(Pous9, INPUT);   digitalWrite (Pous9,HIGH);   Serial.begin(9600);   Serial.println("Pressez   quand la Led s'allume"); }</pre>	<pre>secondes   alea = random(500, 5000); // fonction   standard   delay (alea);   digitalWrite (Led,HIGH) ;   cnt = 0 ;   while (digitalRead (Pous9) == HIGH)   {     delay (10);     cnt++;   }   // on continue ici si Pous9 == LOW   Serial.print(cnt); //décimal   Serial.println(" centiemmes");   delay (1000) ;   // on prépare un nouveau cycle   digitalWrite (Led,LOW) ; }</pre>
---	--	---

#### Programme 4 TestSerie (option)

<p>Des exemples de ce que l'on peut montrer sur l'écran, pratique pour faire du copier-collé ultérieurement. Parler des retour de ligne, espaces, de l'hexadécimal et des types de données si on a le temps.</p>	<pre>/* \prog:TestSerie Différents formats A la fin du chargement, cliquer à droite et faire un reset pour réexécuter le programme */ void setup() {   Serial.begin(9600); } int val ; int i;</pre>	<pre>void loop() {   val = 45;   Serial.println(val,DEC); // In   si saut de ligne   Serial.print(val,HEX);   Serial.println("texte");   for (i=0;i&lt;7;i++) {     Serial.print(i,DEC);   }   Serial.print ("Diduino   \t\tSimple \tRobot \r\n");   Serial.print ("-----\r\n");   while (1); }</pre>
--	---	---

Insérer le minishield Moteur ou utiliser le Xbot, compatible (mêmes déclarations).



#### Programme 5 TestMot

<p>Avance un peu, stoppe, etc. Pour se mettre d'accord comment nommer les pins et vérifier que les moteurs tournent dans le bon sens. Question: comment vérifier que le moteur droite est à droite? Chaque moteur est commandé par 2 bits. Que se passe-t-il si les deux bits sont à 1?</p>	<pre>// TestMot.ino Avance, stoppe, etc #define RecG 7 // Recule Mot Gauche #define AvG 6 #define AvD 5 #define RecD 4 void setup() {   pinMode(Led,OUTPUT);   pinMode(RecG,OUTPUT);   pinMode(AvG,OUTPUT);   pinMode(AvD,OUTPUT);   pinMode(RecD,OUTPUT);</pre>	<pre>} void loop () {   digitalWrite(AvD,HIGH);   digitalWrite(AvG,HIGH);   digitalWrite(RecG, LOW);   digitalWrite(RecD, LOW);   delay(2000);   digitalWrite(AvD,LOW);   digitalWrite(AvG,LOW);   delay(1000); }</pre>
---	--	---

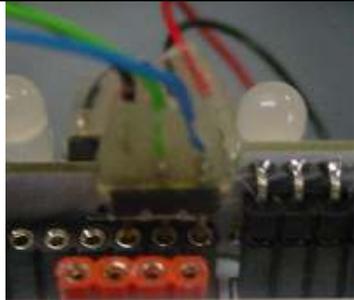
#### Programme 6 Carré

<p>Dessine un carré. Son intérêt est d'utiliser des fonctions que l'on nomme à son gré et rendent le programme plus clair. L'alimentation autonome est nécessaire pour ce test, le câble USB n'étant pas assez long et souple.</p> <p style="text-align: right;">suite de la page précédente</p>	<pre>//Carre.ino #define Led 13 #define RecG 7 #define AvG 6 #define AvD 5 #define RecD 4 void setup() {   pinMode(Led,OUTPUT);   pinMode(RecG,OUTPUT);   pinMode(AvG,OUTPUT);   pinMode(AvD,OUTPUT);   pinMode(RecD,OUTPUT);</pre>	<pre>void tourneDroite (int dd) {   digitalWrite(AvG, HIGH);   digitalWrite(RecG, LOW);   digitalWrite(AvD, LOW);   digitalWrite(RecD, HIGH);   delay(dd); } void tourneGauche (int dd) {   digitalWrite(AvG, LOW);   digitalWrite(RecG, HIGH);</pre>
--	---	---

	<pre> } void stop () {   digitalWrite(AvG, LOW);   digitalWrite(RecG, LOW);   digitalWrite(AvD, LOW);   digitalWrite(RecD, LOW); } void avance (int dd) {   digitalWrite(AvG, HIGH);   digitalWrite(RecG, LOW);   digitalWrite(AvD, HIGH);   digitalWrite(RecD, LOW);   delay(dd); } void recule (int dd) {   digitalWrite(AvG, LOW);   digitalWrite(RecG, HIGH);   digitalWrite(AvD, LOW);   digitalWrite(RecD, HIGH);   delay(dd); } </pre>	<pre> digitalWrite(AvD, HIGH); digitalWrite(RecD, LOW); delay(dd); }  #define Angle90 500 #define Side 2000 void loop() {   avance (Side);   tourneDroite (Angle90);   avance (Side);   tourneDroite (Angle90);   avance (Side);   tourneDroite (Angle90);   avance (Side);   tourneDroite (Angle90);   stop ();   delay (2000) ; } </pre>
--	---	--

### Programme 7 EviteObstacles

Les moustaches sont câblées sur les pins 8 et 9. Les pull-up sont programmées. Pour faciliter le câblage, le 0V (Gnd) est pris sur la pin 10, qui doit être à l'état 0.



```

*
EvitObstacle.ino
*/
#define RecD 4
#define AvD 5
#define AvG 6
#define RecG 7
#define MousD 8
#define MousG 9
#define Gnd 10
void setup() // initialisation
{
  pinMode(RecG,OUTPUT);
  pinMode(AvG,OUTPUT);
  pinMode(AvD,OUTPUT);
  pinMode(RecD,OUTPUT);
  pinMode(MousD,INPUT);
  pinMode(MousG,INPUT);
  pinMode (Gnd,OUTPUT);
  digitalWrite(MousD,HIGH);
  digitalWrite(MousG,HIGH);
  digitalWrite (Gnd,LOW);
}

```

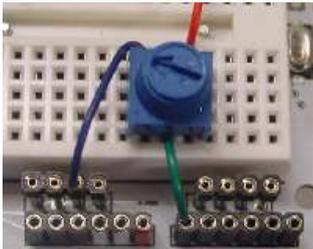
```

void loop()
{
  digitalWrite(AvG, HIGH); // on
  avance
  digitalWrite(RecG, LOW);
  digitalWrite(AvD, HIGH);
  digitalWrite(RecD, LOW);
  if (digitalRead (MousD) == 0)
    //on recule en tournant 1s
    {
      digitalWrite(AvG, LOW);
      digitalWrite(RecG, HIGH);
      digitalWrite(AvD, LOW);
      digitalWrite(RecD, LOW);
      delay (1000) ;
    }
  if (digitalRead (MousG) == 0)
    //on recule en tournant 1s
    {
      digitalWrite(AvG, LOW);
      digitalWrite(RecG, LOW);
      digitalWrite(AvD, LOW);
      digitalWrite(RecD, HIGH);
      delay (1000) ;
    }
}

```

### Programme 8 TestAna

Il faut câbler un pot entre le + et le – et relier le point milieu à la pin 14 (A0). Le terminal affiche des valeurs de 0 à 1023 – expliquer que le convertisseur est 10 bits.



```

// TestAna| Pot sur pin 14
Valeur sur terminal

void setup()
{
  pinMode (A0, INPUT);
  Serial.begin(9600);
}

```

```

int val = 0;
void loop()
{
  val = analogRead(A0);
  Serial.println(val);
  delay(1000); // wait for a
  second
}

```

### Programme 9 Dimmer

Ce programme commande l'intensité de la led6 selon la position du potentiomètre. Il faut expliquer la raison du nom

```

// Dimmer Varie l'intensité sur la pin6
#define Led6 6
void setup()
{
  pinMode(A0, INPUT);
}

```

```

int val = 0;
void loop()
{
  val = analogRead (A0);
  AnalogWrite (Led6, val/4) ;
}

```

<p>analogWrite          Cette fonction n'est valable que sur les pins 3 5 6 9 10 11          Le Led du minishield est utilisable.          Expliquer pourquoi il faut diviser par 4</p>	<pre>pinMode (Led6, 6) ; }</pre>	<pre>delay(1000); }</pre>
---	----------------------------------	---------------------------

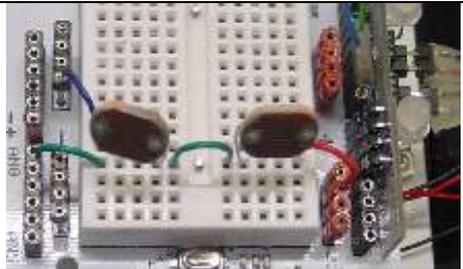
### Programme 10 **LedVarie** (option)

<p>Ce programme augmente puis diminue l'intensité de la led6. Il utilise des boucles for que l'on devrait commencer à expliquer dans un exemple plus simple (si le but est d'apprendre à programmer).</p>	<pre>// LedVarie.ino l'intensité augmente puis diminue #define Led6 6 void setup() {   pinMode (Led6, OUTPUT); }</pre>	<pre>void loop () {   for (int i=0 ; i&lt;256 ; i++ )   {     analogWrite(Led6,i);     delay (50) ;   }   for (int i=255 ; i&gt;0 ; i-- )   {     analogWrite(Led6,i);     delay (50) ;   } }</pre>
---	--	---

### Programme 11 **PwmMot**

<p>On commande les deux moteurs en PWM par le potentiomètre.</p>	<pre>// PwmMot.ino  Vitesse selon pot #define RecD 4 // HIGH LOW #define AvD 5 // pwm #define AvG 6 // pwm #define RecG 7 void setup() {   pinMode(A0, INPUT);   pinMode(RecG,OUTPUT);   pinMode(AvG,OUTPUT);   pinMode(AvD,OUTPUT);   pinMode(RecD,OUTPUT); }</pre>	<pre>int val ; int vit ; void loop () {   val = analogRead(A0);   vit = val/4 ;   digitalWrite (RecD,LOW);   digitalWrite (RecG,LOW);   analogWrite(AvD,vit);   analogWrite(AvG,vit); }</pre>
--	--	---

### Programme 12 **Photophobe**

<p>Deux photorésistances (LDR) sont câblées en diviseur de tension. La tension du point milieu envoyé sur A0 dépend de la différence d'éclairage sur les deux LDR. On commande les moteurs en inversant la valeur sur le bon moteur.</p>	<pre>// Photophobe.ino  Différence de vitesse selon pont de LDR #define RecD 4 // HIGH LOW #define AvD 5 // pwm #define AvG 6 // pwm #define RecG 7 void setup() {   pinMode(A0, INPUT);   pinMode(RecG,OUTPUT);   pinMode(AvG,OUTPUT);   pinMode(AvD,OUTPUT);   pinMode(RecD,OUTPUT); }</pre>	<pre>int val ; byte vitD, vitG ; void loop () {   val = analogRead(A0); // 0 - 1023   vitD = val/4 ; // 0 - 255   vitG = 256-vitD ; // 255 - 0   digitalWrite (RecD,LOW);   digitalWrite (RecG,LOW);   analogWrite(AvD,vitD);   analogWrite(AvG,vitG); }</pre>
		

Commentaires bienvenus