

PLC ARDUINO

Projet C# EI4

Belin Enguerrand
Debossu David
Naouis Nidal

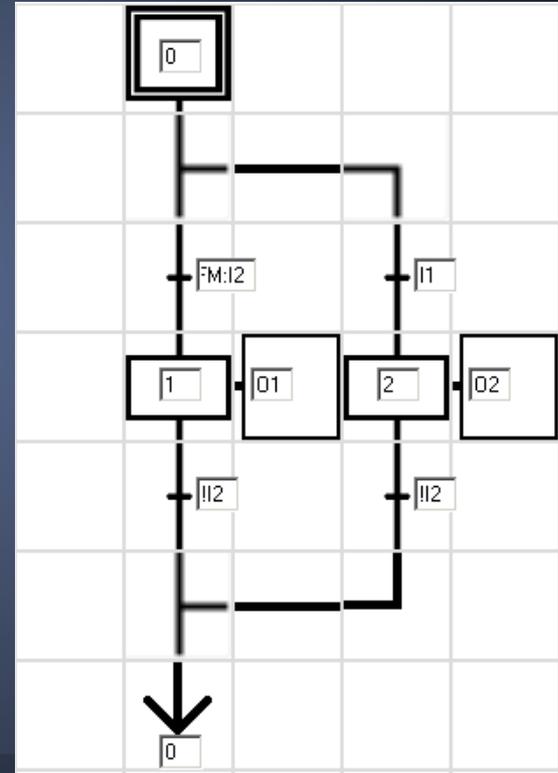


Plan

- Présentation du projet
- Nos outils [visual studio, TFS, TinyPG]
- Notre solution
- Résultats obtenus
- Conclusion et améliorations possibles

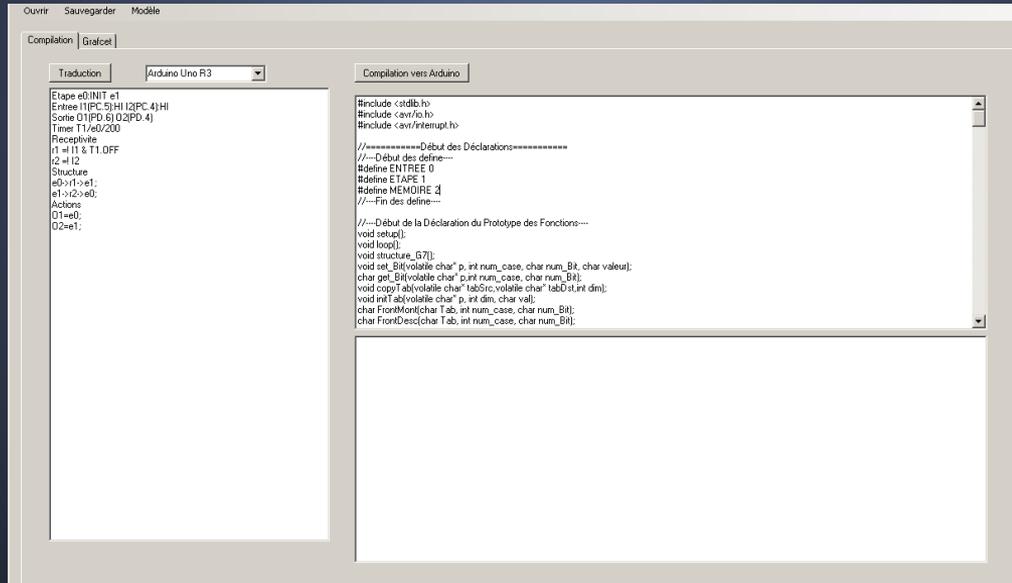
Présentation du projet

Explication grafcet



Présentation du projet

Objectif du projet précédent



```
Traduction | Arduino Uno R3 | Compilation vers Arduino

Etape e0:INIT e1
Entree I(PC.5)H I2(PC.4)H
Sortie O1(PD.8)O2(PD.4)
Timer T1/e0/200
Receptive
I1 = I1 & T1.OFF
I2 = I2
Structure
e0->I1->e1;
e1->I2->e0;
Actions
O1=e0;
O2=e1;

#include <stdio.h>
#include <avr/io.h>
#include <avr/interrupt.h>

//-----Début des Déclarations-----
//---Début des define---
#define EN_TFREE 0
#define ETAPE 1
#define MEMOIRE 2
//---Fin des define---

//---Début de la Déclaration du Prototype des Fonctions---
void setup();
void loop();
void structure_G7I();
void set_Bit(volatile char* p, int num_case, char num_Bit, char valeur);
char get_Bit(volatile char* p, int num_case, char num_Bit);
void copy1_abyvolatile(char* tabSrc,volatile char* tabDst,int dim);
void init_abyvolatile(char* p, int dim, char val);
char FrontMont(char Tab, int num_case, char num_Bit);
char FrontDesc(char Tab, int num_case, char num_Bit);
```



Nos outils



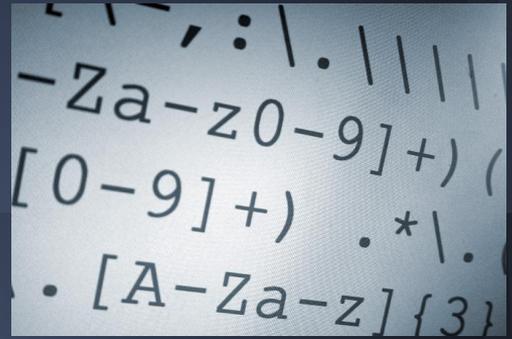
Plateforme de développement multilangage
Permet un accès Team Foundation Server

Systeme de stockage en ligne (gestion de versions)

- Modifications et fusion en ligne
- Historisation personnalisée
- Permet de développer de n'importe où

Nos outils (tinypg)

-Compilateur pour l'APJCGT



```

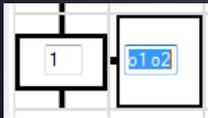
Compilation Grafcet
-----
Traduction      Arduino Uno R3      Compilation vers Arduino

Etape e0-INIT e1
Entree I1(PC.5) HI I2(PC.4):HI
Sortie O1(PD.6) O2(PD.4)
Timer T1/e0/200
Receptivite
r1 =! I1 & T1.OFF
r2 =! I2
Structure
e0->r1->e1;
e1->r2->e0;
Actions
O1=e0;
O2=e1;

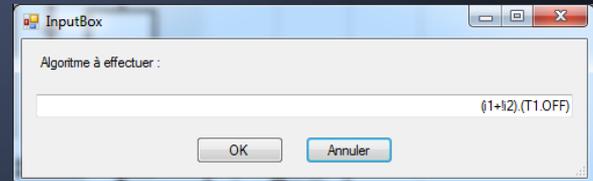
//=====Début des Déclarations=====
//---Début des define---
#define ENTREE 0
#define ETAPE 1
#define MEMOIRE 2
//---Fin des define---

//---Début de la Déclaration du Prototype des Fonctions---
void setup();
void loop();
void structure_G70;
void set_Bit(volatile char* p, int num_case, char num_Bit, char valeur);
char get_Bit(volatile char* p, int num_case, char num_Bit);
  
```

-Vérificateur syntaxique pour nous

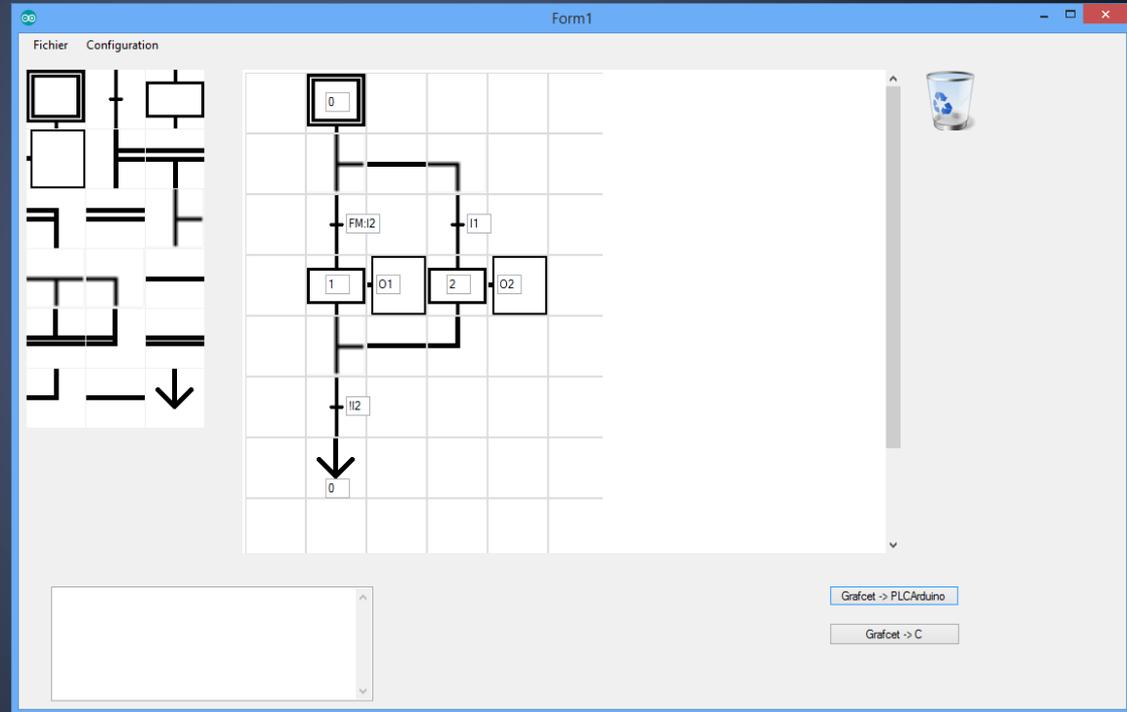


Aucune erreur syntaxique décelée



Notre solution

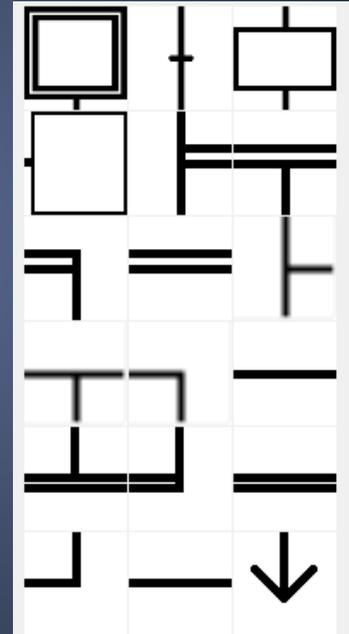
L'interface



Notre solution

L'interface

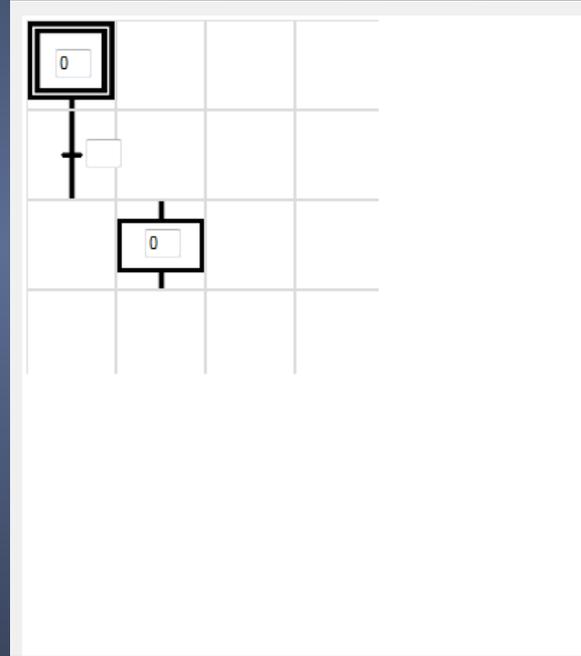
Boite à outils



Notre solution

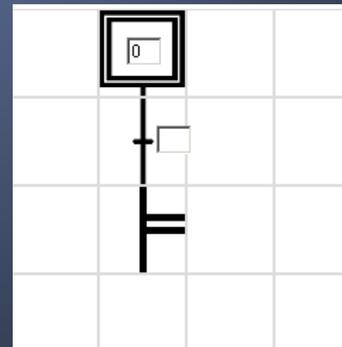
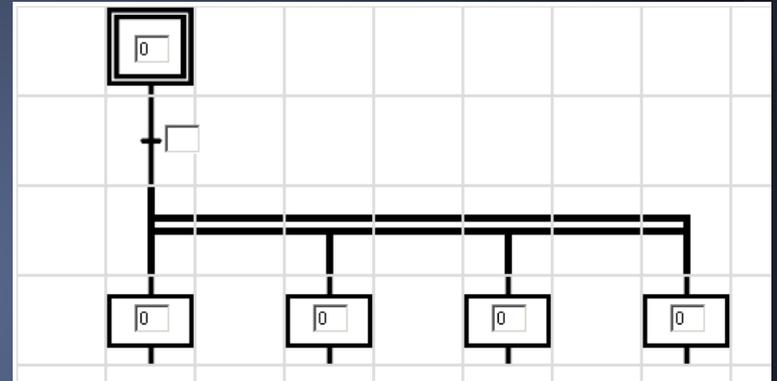
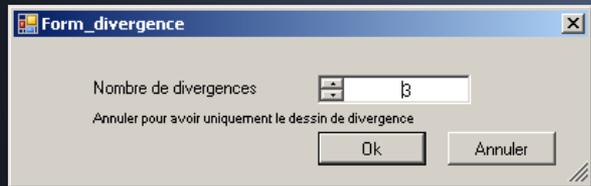
L'interface

Zone de dessin



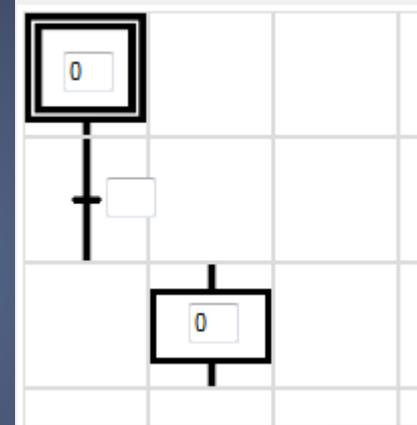
Notre solution

L'interface Divergences



L'interface

Zones de vérification

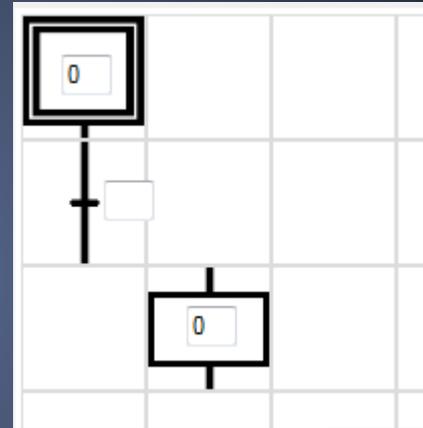


erreur: Deux étapes ne peuvent pas avoir le meme numéro d'étape
erreur: une transition doit être entre deux étapes
erreur: une étape doit être entre deux transitions

L'interface

Zones de vérification

```
Start -> (OrExpr)+ ;  
OrExpr -> AndExpr (PLUS AndExpr)*;  
AndExpr -> Atom (POINT Atom)*;  
Atom -> (FrontMontant |FrontDescendant)? (_rtimer |  
PARENTOUV OrExpr PARENTFER | input | etape_on |  
CaracRECEPT)+;
```



Une ou plusieurs erreurs syntaxiques décelées !

L'interface Configuration

Form_init

Mot à utiliser		Mot à utiliser		Milliseconde	étape
PB.0	<input type="text"/>	PB.1	<input type="text"/>	Timer0	<input type="text" value="0"/>
PB.2	<input type="text"/>	PB.3	Sortie <input type="text"/>	Timer1	<input type="text" value="0"/>
PB.4	Entrée <input type="text"/> I1	PB.5	<input type="text"/>	Timer2	<input type="text" value="0"/>
PB.6	<input type="text"/>	PB.7	<input type="text"/>	Timer3	<input type="text" value="0"/>
PC.0	Entrée <input type="text"/> I2	PC.1	<input type="text"/>	Timer4	<input type="text" value="0"/>
PC.2	<input type="text"/>	PC.3	<input type="text"/>	Timer5	<input type="text" value="0"/>
PC.4	<input type="text"/>	PC.5	<input type="text"/>	Timer6	<input type="text" value="0"/>
PC.6	<input type="text"/>	PC.7	<input type="text"/>	Timer7	<input type="text" value="0"/>
PD.0	<input type="text"/>	PD.1	<input type="text"/>	Timer8	<input type="text" value="0"/>
PD.2	<input type="text"/>	PD.3	<input type="text"/>	Timer9	<input type="text" value="0"/>
PD.4	<input type="text"/>	PD.5	<input type="text"/>		
PD.6	<input type="text"/>	PD.7	<input type="text"/>		

OK Annuler

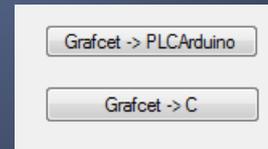
L'interface

Les menus et les boutons

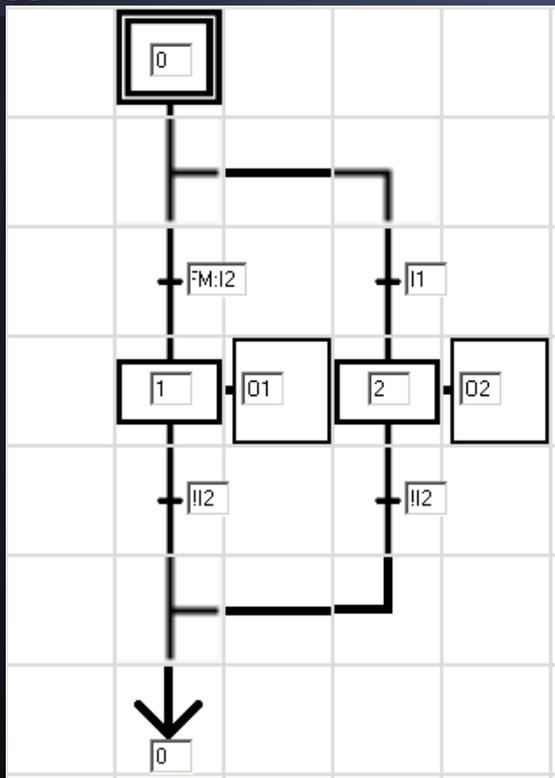
Menu



Conversion



Résultats obtenus



```
Etape e0:INIT e1 e2
Entree I1(PB.2):HI I2(PC.0):HI
Sortie O1(PB.7) O2(PC.3)
```

Receptivite

r1 = FM:I2

r2 = I1

r3 = !I2

r4 = !I2

Structure

e0 ->r1->e1 ;

e0 ->r2->e2 ;

e1 ->r3->e0 ;

e2 ->r4->e0 ;

Actions

O1= e1 ;

O2= e2 ;



Améliorations possibles

- Mode automatisé
- Aide et choix contextuels
- Analyse syntaxique détaillée
- Mode débogage
- Améliorer la configuration de la carte Arduino
- Mode sélection
- Raccourci clavier



Conclusion

- Programmation C#
- Utilisation de parser
- Grafcet
- IHM