ADESFA Séminaire de formation des formateurs Informatique Industrielle et Systèmes Embarqués INUBIL-Douala – Université Paris-Saclay 28 octobre- 02 novembre 2024

## SYSTEME ARDUINO MEGA 2560

### TP1 Initiation et Fonctionnalités de Base du Système Arduino



## Table des matières

I.	Le système Arduino MEGA 2560 3
	Présentation3
	Introduction
	Compétences visées4
II.	Initiation et fonctionnalités de base5
	Objectifs
	La platine d'expérimentation5
Ш	. Présentation de l'IDE
IV	. Pour commencer : une LED clignotante 8
v.	Chenillard à fréquence fixe11
VI	Lecture d'une tension analogique et conversion analogique-numérique
VI	I. Chenillard à fréquence variable13
VI	II. Commande des feux de circulation d'un carrefour
IX	. CONCLUSIONS

#### I. Le système Arduino MEGA 2560

#### Présentation

Le système Arduino MEGA 2560 R3, basé sur le microcontrôleur 8 bits ATmega2560, présente les éléments principaux suivants, permettant de mettre en œuvre des travaux pratiques sans ajout de composants extérieurs onéreux.

Broch	es (pins)	
Built-i	13	
Digital	l I/O Pins	54
Analog	g input pins	16
PWM	15	
Comn	nunication	
UART	Yes,	4
I2C	Yes	
SPI	Yes	
Alime	ntation	
I/O Vo	5V	
Input	7-12V	

n o rollago	01
Input voltage (nominal)	7-12V
DC Current per I/O Pin	20 mA
Supported battery	9V battery
Power Supply Connector	Barrel Plug

#### Horloge

Main Processor	ATmega2560 16 MHz
<b>USB-Serial Processor</b>	ATmega16U2 16 MHz

#### Mémoire

ATmega2560 8KiB SRAM, 256KiB FLASH, 4KiB EEPROM

### Introduction

Le contenu de ce sujet est valable pour le système Arduino ATmega2560 basé sur le microcontrôleur 8-bit ATmega2560. Ces travaux pratiques peuvent facilement s'adapter à d'autres versions de la même famille de ces cartes microcontrôleurs, l'utilisation du contenu de ce manuel doit faire l'objet d'adaptations mineures en vue de respecter les spécifications de chaque version et les fonctionnalités intégrées.

L'ensemble des exemples d'illustrations, est inspiré de programmes disponibles en accès libre (Open Source).

Le gratuiciel (freeware) Arduino IDE 2.3.x d'Arduino.org est un environnement de travail intégré, téléchargeable gratuitement à partir du lien suivant : (dernière version disponible IDE 2.3.3) : https://www.arduino.cc/en/software

Pour plus d'informations relatives au microcontrôleur 8-bits ATmega2560 (Fig. 1), vous pouvez télécharger sa fiche technique du site de son fabricant Microchip : https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/ATmega640-1280-1281-2560-2561-Datasheet-DS40002211A.pdf



Fig. 1 Brochage du microcontrôleur ATmega2560

# Compétences visées

L'objectif est de mettre à votre disposition, des sujets de travaux pratiques concernant le système Arduino utilisant le microcontrôleur 8-bits ATmega2560. Les sujets ci-inclus, inspirés d'exemples disponibles en accès libre et des travaux pratiques réalisés à l'ISTY basés sur des systèmes 32-bits, ont été conçus pour des objectifs pédagogiques montrant la faisabilité d'enseigner la théorie des microcontrôleurs et la valider par des travaux pratiques en se basant sur des équipements accessibles.

Compétences et connaissances visées : liste non exhaustive

- Les éléments constitutifs du système Arduino MEGA2560.
- L'environnement Arduino IDE (Fig. 2).
- La programmation en langage C du système Arduino.
- La programmation des différentes opérations arithmétiques et logiques.
- Les signaux et les interfaces d'entrée/sorties et la programmation en temps réel.
- La communication du microcontrôleur avec différents périphériques.
- Les différents types de périphériques (série, parallèle, sans fils).



Fig. 2 Exemple d'écran d'accueil de l'environnement Arduino IDE 2.3.3

## II. Initiation et fonctionnalités de base

### Objectifs

- Se familiariser avec le système Arduino MEGA 2560 R3 et sa programmation
- Ecrire, compiler, téléverser et exécuter un programme développé en C.
- Mettre en œuvre des exemples d'applications.

### La platine d'expérimentation

La plaquette d'expérimentation aussi appelée moniteur ou maquette est une matrice de points contacts disposés en lignes et en colonnes selon le schéma type de la Fig. 3. Le nombre de ces points-contact détermine la taille de la plaquette.



Fig. 3. La platine : les lignes en rouge, en bleu et en noir indiquent les points reliés entre eux.

## **Composants passifs**

Coulour	1ère bague	2ème bague	3ème bague	4ème bague		
Couleur	Chiffre des dizaines	Chiffre des unités	Multiplicateur 10 <sup>™</sup>	Tolérance		
Noir	0	0	$10^{\circ}$ = 1 $\Omega$			
Marron	1	1	$10^{1}$ = 10 $\Omega$			
Rouge	2	2	$10^2 = 100 \Omega$			
Orange	3	3	10 <sup>3</sup> =1 ΚΩ			
Jaune	4	4	$10^4$ = 10 K $\Omega$			
Vert	5	5	$10^5$ = 100 K $\Omega$			
Bleu	6	6	$10^6 = 1 M\Omega$			
Violet	7	7	$10^{7}$ = 10 M $\Omega$			
Gris	8	8				
Blanc	9	9				
Or			10 <sup>-1</sup> = 0,1	± 5%		
Argent			10 <sup>-2</sup> = 0,01	± 10%		

Tableau 1 Code de couleurs des résistances

Exemple : une résistance ayant le code de couleurs suivant :

1ere bague: vert $\rightarrow$  chiffre 52eme bague: bleu $\rightarrow$  chiffre 63eme bague: jaune $\rightarrow$  chiffre 44eme bague: argent :  $\rightarrow$  tolérance :  $\pm$  10%

La valeur de cette résistance est donc : **56 x 10**<sup>4</sup>  $\pm$  **10% = 560 K** $\Omega$   $\pm$  **10%** la valeur réelle se situe dans l'intervalle :

- $\circ$  560 K $\Omega$  56 K $\Omega$  = **504 K\Omega**
- $\circ$  et 560 K $\Omega$  + 56 K $\Omega$  = 616 K $\Omega$



# Quelques précautions à prendre

- Toujours vérifier le montage du circuit avant de le mettre sous tension.
- Ne jamais modifier le câblage d'un circuit alors qu'il est sous tension.
- Ne jamais laisser trainer sur la platine, des fils non connectés.

### III. Présentation de l'IDE

Sans alimentation externe, la carte MEGA2560 peut être alimentée par le PC via le câble USB.

sketch_oct14a   Arduino IDE 2.3.3 —							×
File Ed	it Sketch	Tools Help					
		Arduino Mega or Mega 2				$\checkmark$	· <b>Q</b> ··
	sketch_o	ot14a.ino					
5	S	elect Other Board and Port			$\times$		
	ŝ	Select both a Board and a Port if you want to f you only select a Board you will be able to	o upload a compile, b	sketch. out not to upload your sketch.			
	E	BOARDS		PORTS			
0		Search board	Q				
Q		Arduino Mega ADK		COM5 Serial Port (USB)			
		Arduino Mega or Mega 2560	~				
		Arduino Micro					
				□ Show all ports			
				CANCEL	ок		

Fig. 4 Choix du type de carte dans l'environnement Arduino (Version IDE 2.3.3)

## Système de numérotation Un nombre précédé du symbole 0x Aucun symbole

*indique* hexadécimal décimal (système utilisé par défaut)

Se référant au système Arduino MEGA2560 et sans interface supplémentaire (ou Shields), en déduire :

- 1. Le nombre maximal de signaux numériques disponibles.
- 2. Le nombre d'entrées analogiques que le système peut traiter.
- 3. Le nombre de sorties MLI disponibles.

#### Rappel:

- La tension de référence analogique est égale à **5 V** par défaut. Vous pouvez changer cette limite via l'entrée AREF et la fonction *analogReference()*.
- Dans un programme C, le symbole // indique un commentaire.
- Les variables sont sensibles à la casse (écriture minuscule et majuscule), par exemple **pinmode** est différent de **pinMode**.

#### IV. Pour commencer : une LED clignotante

Dans l'onglet « Fichier » choisir Exemples  $\rightarrow$  01. Basics  $\rightarrow$  Blink, voir Fig. 5.



Fig. 5 Choix de l'exemple d'application d'une LED clignotante.

Compiler le programme en cliquant sur "Vérifier" (symbole : ✓), voir Fig. 6.



Fig. 6 Compilation du programme

A la fin de la compilation, cet exemple ne contenant pas d'erreurs, vous pouvez le téléverser vers le système Arduino pour l'exécuter, en cliquant sur ( $\rightarrow$ ), voir Fig. 7.

💿 Blir	nk   Arduino I	DE 2.3.3 —		$\times$
File E	dit Sketch	Tools Help		
	→	☐ Arduino Mega or Mega 2… ▼	$\checkmark$	· <b>O</b> ··
Ph	Blink.ino			
	1	/*		
_	2	Blink		
1_)	3			
	4	Turns an LED on for one second, then off for one second, repeatedly.		
Mh	5			
	6	Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO		
	7	it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to		
$\oslash$	8	the correct LED pin independent of which board is used.		
	9	If you want to know what pin the on-board LED is connected to on your Arduino		
$\sim$	10	model, check the Technical Specs of your board at:		
Q	11	https://www.arduino.cc/en/Main/Products		

Fig. 7 Téléversement du programme vers le système Arduino MEGA 2560

1. Présenter de manière succincte la structure d'un programme en C.

- 2. Quel est le rôle de *pinMode ()*?
- 3. Peut-on définir une ligne d'entrée/sortie numérique pour agir simultanément en Entrée ET en Sortie ? Justifier votre réponse.
- 4. Quelle est l'unité de temps de la fonction Delay.

### Un bouton-poussoir pour contrôler une LED

Refaire la même procédure que IV pour choisir l'exemple du bouton-poussoir (ou interrupteur) pour commander la même LED reliée à la ligne d'E/S 13. Voir Fig. 8.



1. Montrer comment peut-on relier au système Arduino, un bouton-poussoir de type NO?

- 2. Donner la différence entre les instructions digitalRead et digitalWrite
- Utiliser deux LED externes reliées à 2 lignes d'E/S de votre choix (entre 2 et 13). Changer le programme précédent pour contrôler les LED selon la méthode suivante : lorsqu'on actionne l'interrupteur, une des LED est allumée, l'autre sera éteinte et vice versa.

**Attention** : Les LED doivent être reliées en **anode commune** de façon à soulager le microcontrôleur, alimenté via le port USB du PC. La connexion en anode commune des LED permet d'éviter de recourir à une alimentation externe.

Les LED doivent être protégées par des résistances en série afin de limiter le courant. Les courants sont fixés en se référant aux notices des fabricants selon les couleurs des LED, tels présentés dans le tableau 2.

LED (taille 5 mm)	Courant de polarisation directe I <sub>f</sub> (mA)	Tension de polarisation V <sub>f</sub> (Volts)	Résistance minimale R (Ω)	$R = \frac{V_{cc} - V_f}{I_f}$
Bleue	20	3.5	75	
Infrarouge	20	1.6	170	
Verte	20	2.6	120	
Orange	20	2.2	140	
Rouge	20	2.1	145	
Jaune	20	2.4	130	
Blanche	20	3.5	75	

# Tableau 2 : Résistances de protection des LED

### Exemple

Si nous fixons ce courant à 15 mA, la loi d'Ohm : R= U/I donnera R  $\ge$  5V/15mA  $\ge$  333  $\Omega$ .

# V. Chenillard à fréquence fixe

Utiliser 4 LED pour les relier aux lignes d'E/S numériques (entre 2 et 13) de votre choix. Ecrire un programme (en C), qui permet de réaliser un chenillard fonctionnant comme suit :

- 1- Chenillard unidirectionnel avec un délai d'une (1) seconde : LED1→LED2 →LED3→LED4→LED1
- 2- Bidirectionnel avec un délai de 0,5 seconde. LED1→LED2→ED3→LED4→LED3→LED2→LED1

Note : Dans les deux versions, faites attention au timing lorsque le chenillard revient à la LED1.

### VI. Lecture d'une tension analogique et conversion analogique-numérique

🔤 Button   Arduino IDE 2.2.1		
File Edit Sketch Tools Help   New Sketch Ctrl+N   New Cloud Sketch Alt+Ctrl+N   Open Ctrl+O   Open Recent   Sketchbook	Built-in examples 01.Basics 02.Digital 03.Analog 04.Communication 05.Control	AnalogInOutSerial AnalogInput AnalogWriteMega digital pin 13,
Close Ctrl+W   Save Ctrl+S   Save As Ctrl+Maj+S   Preferences Ctrl+Virgule   Advanced Ctrl+Virgule	06.Sensors 07.Display 08.Strings 09.USB 10.StarterKit_BasicKit 11.ArduinoISP	Calibration Fading Smoothing tor

Fig. 9 Choix de l'exemple d'application de lecture d'une tension analogique

Rappel: Les tensions d'entrée analogique sont par défaut limitées à 5 V. Utiliser un potentiomètre de 2,2 k $\Omega$ . Toute autre valeur disponible peut également convenir sous réserve de refaire le calcul correspondant.

Un diviseur de tension permet de déterminer une tension proportionnellement à une autre tension.



GND (masse)

Fig. 10 Branchement du potentiomètre en diviseur de tension

Utiliser Serial.begin(9600) et Serial.println() pour déduire et afficher sur le moniteur (écran de votre PC), la valeur numérique correspondant à

- 1. La tension maximale d'entrée (5 V)
- 2. La tension minimale d'entrée (0 V)
- 3. Proposer une méthode permettant de mesurer une tension d'entrée qui varie par exemple de 0 à 10 V.

#### VII. Chenillard à fréquence variable

Combiner les exemples des parties V et VI pour écrire un programme permettant de réaliser un chenillard aller-retour, dont la fréquence de clignotement est déterminée par le potentiomètre.

#### VIII. Commande des feux de circulation d'un carrefour

#### **Composants requis :**

- 4 x LEDS x 3 couleurs
- (Rouge <mark>R</mark> + Jaune <mark>J</mark> + Verte <mark>V</mark>)
- Résistance de 220  $\Omega$  (x4)

1. Dans un premier temps, nous allons ignorer les feux de signalisation destinés aux piétons.

Les feux de circulation doivent s'allumer alternativement comme suit : Vert  $\rightarrow$  5 s, Jaune  $\rightarrow$  2 s, Rouge  $\rightarrow$  5 s. La synchronisation des feux F1/F3 et F2/F4 respectivement, permet d'optimiser le nombre de lignes d'E/S utilisées.

Réaliser le montage et écrire un programme permettant de contrôler les feux de circulation de ce carrefour.



Rappel : afin de soulager les sorties du microcontrôleur, les LEDS doivent être connectées en anode commune comme suit :



La résistance de protection étant commune aux trois LED, elle doit être capable de dissiper une puissance de : P = 3 x U x I = 3 x 5 x 15 x  $10^{-3}$  = 225 mW. Une résistance de ¼ W vérifie donc ce critère.

2. Compléter votre montage et changer votre programme pour intégrer les feux de circulation piétons (P1D : feu piéton 1 droit, P1G : feu piéton 1 gauche, ...).

### IX. CONCLUSIONS

- Environnement Arduino IDE
- Compatibilité des systèmes Arduino
- Programmation en C
- Interfaces d'entrées-sorties