
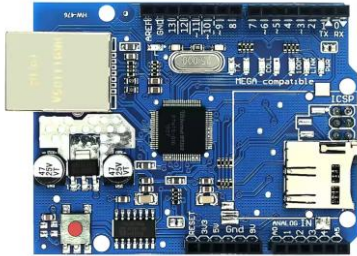








# Fabrication d'une boîte de 64 clés pilotable par socket TCP/IP

## 1. Le concept :



- Piloter des sorties via un protocole sur socket TCP/IP
- Connaître le niveau des entrées par interrogation via protocole TCP/IP
- Tension de fonctionnement : +5V
- Ce système peut piloter un ensemble utilisant toute autre tension en utilisant des relais optocoupleurs. Que ce soit pour les sorties ou pour les entrées
- Paramétrer l'ensemble par un fichier ini présent sur la carte SD pour éviter d'avoir à modifier le programme au moindre changement

## 2. Le matériel nécessaire :

MATERIEL OBLIGATOIRE	
Un Arduino MEGA2560	
Un bouclier Ethernet SDCARD	
Une Carte Micro SD Taille maxi 8Gb ! Conseillée : 1Gb ,2Gb ou 4Gb Format FAT32 !	
8 MCP23017 I2C chacun de ces composants offre la possibilité de piloter 8 clés	

MATÉRIEL OPTIONNEL (CONTRÔLE VISUEL DU FONCTIONNEMENT)	
11 LED Rouges 5V (3 + 1 par MCP)	
11 LED Vertes 5V (3 + 1 par MCP)	
11 résistances 330 ohms (1 par LED rouge)	
11 résistances 220 ohms (1 par LED verte)	

### 3. Les logiciels nécessaires :

Pour la programmation : Arduino	
Bibliothèque de pilotage des MCP	Adafruit_MCP23017
Client TCP : packet sender (gratuit) (mise au point, tests)	

### 4. Le fichier de paramétrage :

La carte SD doit impérativement contenir à la racine un fichier nommé param.ini

Il doit contenir les informations suivantes :

Section	Nom de la variable	Définition	Exemple
network	mac	Adresse mac de la carte réseau	00:12:34:56:78:9A
network	port	Port TCP que le socket TCP écouterait	4925
timer	impulse	Durée (en ms) durant laquelle la sortie sera activée	100
MCP2307-00	addr	Adresse du MCP valeurs possibles de 0x20 à 0x27 ou 0x00	0x20 ou 0x00
MCP2307-01	addr	Adresse du MCP	0x21 ou 0x00
MCP2307-02	addr	Adresse du MCP	0x22 ou 0x00
MCP2307-03	addr	Adresse du MCP	0x23 ou 0x00
MCP2307-04	addr	Adresse du MCP	0x24 ou 0x00
MCP2307-05	addr	Adresse du MCP	0x25 ou 0x00
MCP2307-06	addr	Adresse du MCP	0x26 ou 0x00
MCP2307-07	addr	Adresse du MCP	0x27 ou 0x00

Exemple concret de fichier :

```
[network]
mac = 00:12:34:56:78:9A
port = 4925
[timer]
impulse=100
[MCP2307-00]
addr=0x20
[MCP2307-01]
addr=0x21
[MCP2307-02]
addr=0x22
[MCP2307-03]
addr=0x23
[MCP2307-04]
addr=0x24
[MCP2307-05]
addr=0x25
[MCP2307-06]
addr=0x26
[MCP2307-07]
addr=0x27
```

## 5. Câblage Arduino : les LED de vérification :

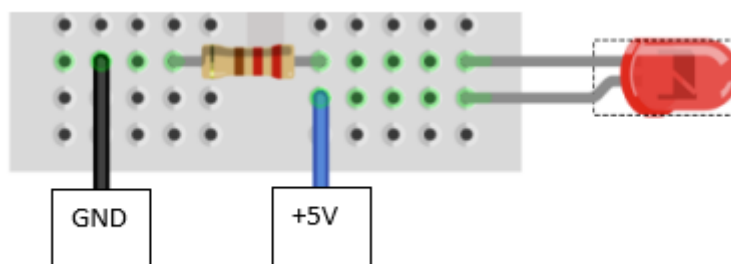
Le programme débute par une séquence de boot très courte qui lui permet de contrôler que tout est en ordre.

Il fournit sur le port série de l'Arduino (9600 Bds) les messages de boot mais des LED sont prévues pour une vérification visuelle sans avoir à connecter un PC au port USB.

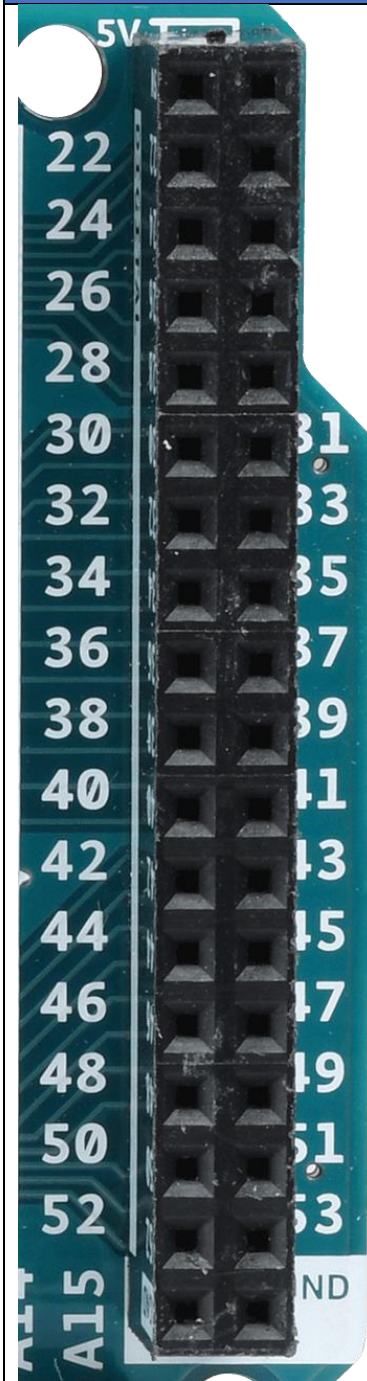
- Ces LED sont rouges (allumées si KO) ou vertes (allumées si OK)
- Chaque LED se connecte ainsi :
- Anode à la masse – Cathode à une borne d'une résistance – PIN de l'Arduino
- **Pour une LED rouge, la résistance est de 330 ohms, pour une LED verte, de 220 Ohms**

Certains tests sont bloquants, d'autres non Ex :

- Un des MCP ne répond pas : ce n'est pas bloquant, le programme continuera en l'ignorant
- Le fichier ini est introuvable : c'est bloquant, l'initialisation s'arrête.



# CONNEXION DES LED A L'ARDUINO MEGA 2560



## 1ERE PARTIE SEQUENCE DE BOOT

PIN 26	LED VERTE	ALLUMEE SI : LECTEUR SD OK CARTE SD PRESENTE ET BIEN FORMATEE
PIN 24	LED VERTE	ALLUMEE SI FICHIER PARAM.INI PRESENT ET CORRECT
PIN 22	LED VERTE	ALLUMEE SI LA CARTE RESEAU S'EST CONNECTEE A VOTRE RESEAU
PIN 27	LED ROUGE	ALLUMEE SI : LECTEUR SD KO CARTE SD ABSENTE OU FORMAT DE CARTE SD ILLISIBLE
PIN 25	LED ROUGE	ALLUMEE SI FICHIER PARAM.INI ABSENT OU INCORRECT
PIN 23	LED ROUGE	ALLUMEE SI LA CARTE RESEAU N'A PAS REUSSIE A SE CONNECTER A VOTRE RESEAU

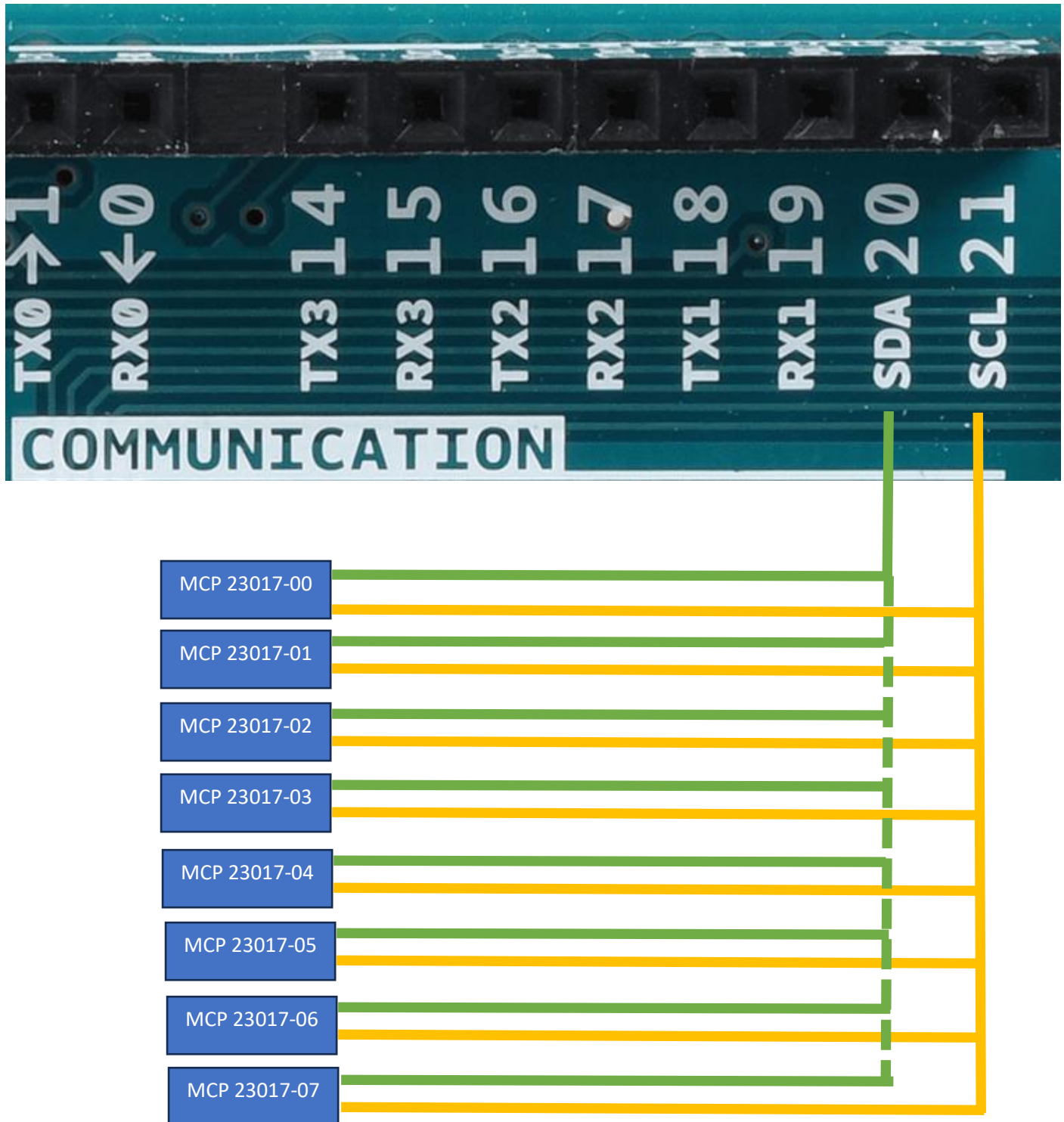
## 2IEME PARTIE SEQUENCE DE BOOT

PIN 29	LED VERTE	ALLUME SI MCP23017-00 PRESENT
PIN 31	LED VERTE	ALLUME SI MCP23017-01 PRESENT
PIN 33	LED VERTE	ALLUME SI MCP23017-02 PRESENT
PIN 35	LED VERTE	ALLUME SI MCP23017-03 PRESENT
PIN 37	LED VERTE	ALLUME SI MCP23017-04 PRESENT
PIN 39	LED VERTE	ALLUME SI MCP23017-05 PRESENT
PIN 41	LED VERTE	ALLUME SI MCP23017-06 PRESENT
PIN 43	LED VERTE	ALLUME SI MCP23017-07 PRESENT
PIN 28	LED ROUGE	ALLUME SI MCP23017-00 NE REPOND PAS
PIN 30	LED ROUGE	ALLUME SI MCP23017-01 NE REPOND PAS
PIN 32	LED ROUGE	ALLUME SI MCP23017-02 NE REPOND PAS
PIN 34	LED ROUGE	ALLUME SI MCP23017-03 NE REPOND PAS
PIN 36	LED ROUGE	ALLUME SI MCP23017-04 NE REPOND PAS
PIN 38	LED ROUGE	ALLUME SI MCP23017-05 NE REPOND PAS
PIN 40	LED ROUGE	ALLUME SI MCP23017-06 NE REPOND PAS
PIN 42	LED ROUGE	ALLUME SI MCP23017-07 NE REPOND PAS

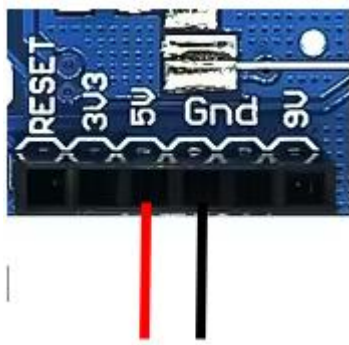
## 6. Câblage Arduino : Les composants MCP

Pour cela, vous utiliserez les PIN +5V, GND, SCL et SDA de l'Arduino en les mettant en parallèles

SDA et SCL sont les 2 ports utilisés en I2C. SDA est utilisé pour échanger des données (DATA), SCL pour réguler l'horloge (CLOCK)



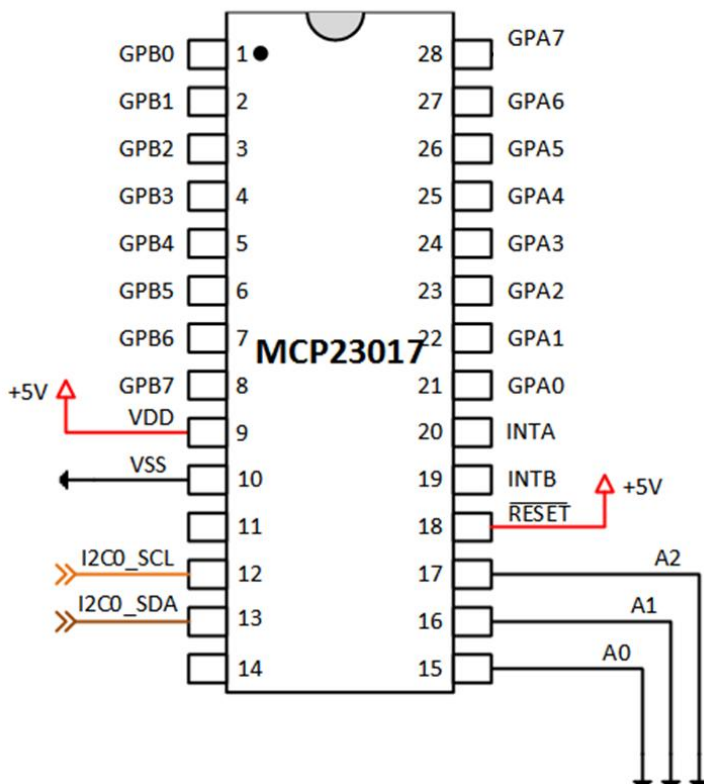
## 7. Câblage Arduino : Alimentation et RESET



L'alimentation des MCP se fait par l'arduino (bornes **+5V et GND**), elles sont remises à disposition sur la carte Ethernet une fois qu'elle est enfichée sur le Arduino

On peut mettre un bouton de reset externe en reliant le pin de Reset à la masse (GND) avec un bouton poussoir et un système anti rebonds (condensateur et résistance)

## 8. Câblage des composants MCP



- SCL et SDA => vers SCL et SDA de l'Arduino (voir précédemment)
- VDD = alimentation +5V
- VSS = alimentation 0V (GND)
- PIN 11, 14, 16 et 20 : inutilisés ici
- RESET = vers un +5V
- **GPA0 à GPA7 = sorties (de 0 à 7)**
- **GPB0 à GPB7 = entrées (de 0 à 7)**
- **A0,A1,A2 = adressage des MCP**  
On fournit l'adresse (unique) à un MCP en le connectant de l'une des manières suivantes

ADR	A2	A1	A0
0x20	GND	GND	GND
0x21	GND	GND	+5V
0x22	GND	+5V	GND
0x23	GND	+5V	+5V
0x24	+5V	GND	GND
0x25	+5V	GND	+5V
0x26	+5V	+5V	GND
0x27	+5V	+5V	+5V



## 9. Répartition des 64 entrées et 64 sorties sur les 8 MCP

NUMERO DE CLE	ADRESSE DU MCP	PIN DU MCP
De 0 à 7	0x20	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 8 à 15	0x21	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 16 à 23	0x22	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 24 à 31	0x23	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 32 à 39	0x24	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 40 à 47	0x25	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 48 à 55	0x26	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7
De 56 à 63	0x27	Sorties : de GPA0 à GPA7 Entrées : de GPB0 à GPB7

Exemple : Clé numéro 37 :

le programme interrogera le MCP dont l'adresse est 0x24

S'il doit activer la sortie il utilisera le PIN GPA5

S'il doit interroger l'entrée, il lira le PIN GPB5

## 10. Niveau des entrées et des sorties

**Entrée : GND = 1 Logique** : Si un port d'entrée est à la masse, le programme comprend que cette entrée est active

**Sortie : +5V = 1 Logique** : lorsque le programme active une sortie, il amène du +5V sur le pin du MCP correspondant

## 11. Réseau TCP

Le programme paramètre la carte réseau en mode DHCP client (IP dynamique)

Il signalera à votre serveur DHCP, la présence d'un nouveau périphérique réseau dont l'adresse MAC est celle présente dans le fichier PARAM.INI et lui demandera de lui fournir une adresse IP.

Une fois réussi, il surveillera toute déconnexion du réseau (câble débranché, ou fin de bail) pour redemander une adresse IP

Lorsque le programme tourne, il fournit un serveur TCP sur le port que vous avez indiqué dans le fichier PARAM.INI.

Si vous vous y connectez via telnet, vous verrez que la connexion n'est pas bloquante mais que tout demande à l'Arduino (envoi de trame) provoque la réception d'une trame. En effet, s'il comprend la demande, il y répond. Sinon, il renvoi la trame qu'il a reçue.

## 12. Protocole de communication

Le programme connaît 2 commandes :

- La demande de statut de l'ensemble des clés
- La demande de libération d'une clé

Il répond à toute demande incomprise par la trame qu'il lui a été envoyé

Les commandes font EXACTEMENT 3 OCTETS

Les réponses aux commandes font EXACTEMENT 10 OCTETS

## 13. La demande de statut

Elle se fait par l'envoi des 3 octets suivant : 0x10 0x02 0x00

La réponse contient 0x10 0x02 suivit de 8 octets ( 1 par MCP)

Chaque octets (dans l'ordre des MCP d'adresse 0x20 à 0x27) correspond au statut des 8 clés connecté aux entrées d'un MCP

Statut d'une entrée : si reliée à la masse => 1 Logique. Sinon, 0 Logique

Si un des MCP est absent ou ne répond pas (adresse 0x00) alors toutes ses entrées seront considérées à 0

Exemples :

- 0x00 : toutes les entrées du MCP sont à 0.
- 0xFF : toutes les entrées du MCP sont à 1
- 0x02 : seul le pin numéro 1 a son entrée à 1

## 14. La demande de libération d'une clé

Elle se fait par l'envoi des 3 octets suivant : 0x10 0x01 + numéro de la clé en hexa

Si elle est sur un MCP qui est présent et répond (adresse différente de 0x00) alors cela provoque un +5V sur le PIN qui convient du MCP correspondant pendant la durée de l'impulsion désirée (fichier PARAM.INI)

En parallèle, au bout de 100ms, quoi qu'il arrive, le programme répond à l'expéditeur par

0x10 0x01 + les 8 octets du statut de toutes les clés