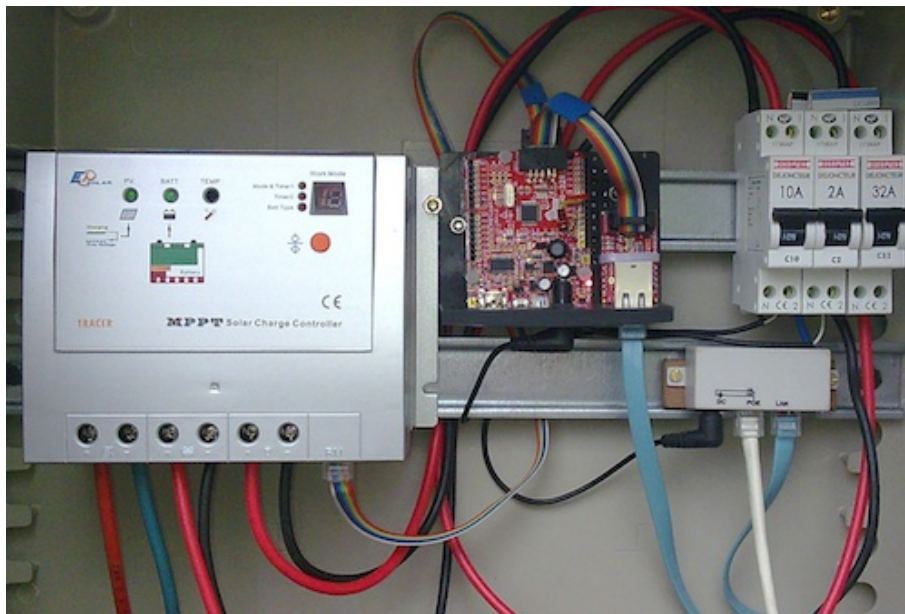




# Lndk Tracer

Version 0.8e



Montage Arduino de contrôle et de commande  
de station solaire en site isolé

# Le protocole d'échange série du régulateur Epsolar Tracer

## Format général des commandes

Chaque commande envoyée vers le Tracer comprend :

- un paquet de synchronisation de 6 octets,  
en hexa 0xEB 0x90 0xEB 0x90 0xEB 0x90,
- un octet pour le numéro du device,
- un octet identifiant la commande,
- un octet indiquant la longueur des données,
- les octets de données,
- deux octets pour le check-sum,
- l'octet 0x7F identifiant la fin de commande.

La réponse du régulateur solaire suit le même format.

En exemple, la lecture des mesures du régulateur,

la commande émise vers le régulateur :

```
EB 90 EB 90 EB 90 // Synchro
01 // Device
A0 // Commande
00 // Longueur des données
6F // Checksum
52 // Checksum
7F // Fin de commande
```

la réponse du régulateur :

```
EB 90 EB 90 EB 90 // Synchro
01 // Device
A0 // Commande
18 // 24 octets de données
E4 04 14 06 00 00 00 00 4C 04 B3 05 00 00 00 29 00 00 00 01 2B 03 00 00
7F 91 // Checksum
7F // Fin de la réponse
```

## Les octets de commande :

**A0** lecture des mesures  
pas d'octet de donnée

**AA** commande l'utilisation  
1 octet de donnée : 0 utilisation coupée, 1 en service

**AD** configuration du régulateur,  
*Attention : cette commande n'a pas été testée, ces informations sont données sous toute réserve*  
7 octets de donnée  
Load type, inutilisé,  
Charging mode, inutilisé,

coefficient de température,  
type de batterie (0gel, 1 sealed, 1 flooded)  
capacité de la batterie (10<sup>e</sup> de la capacité, 15 pour 150AH)  
load control mode 1  
load control mode 2

#### Détail du *load control mode*

0 Dusk to Dawn  
1 à 15 Mode timer, voir le manuel  
16 Test  
17 Manuel, le mode à utiliser si l'on veut avoir la maîtrise de l'utilisation  
18 Désactivé  
*les modes 16 et 17 n'existent pas pour **load control mode 2***

### Détail des mesures reçues du régulateur

Détail des données reçues (tension en dizaine de mV, courant en dizaine de mA)

Tension batterie	E4 04
Tension panneaux	14 06
Réserve ?	00 00
Consommation	00 00
Batterie faible	4C 04
Batterie pleine	B3 05
Utilisation connectée	00
Indicateur surcharge	00
Indicateur de court-circuit	00
Etat de charge (SOC)	29
Surcharge batterie	00
Batterie déchargée	00
Batterie pleine	00
Indicateur de charge	01
Température (degré + 30)	2B
Courant de charge	03 00
Réserve ?	00

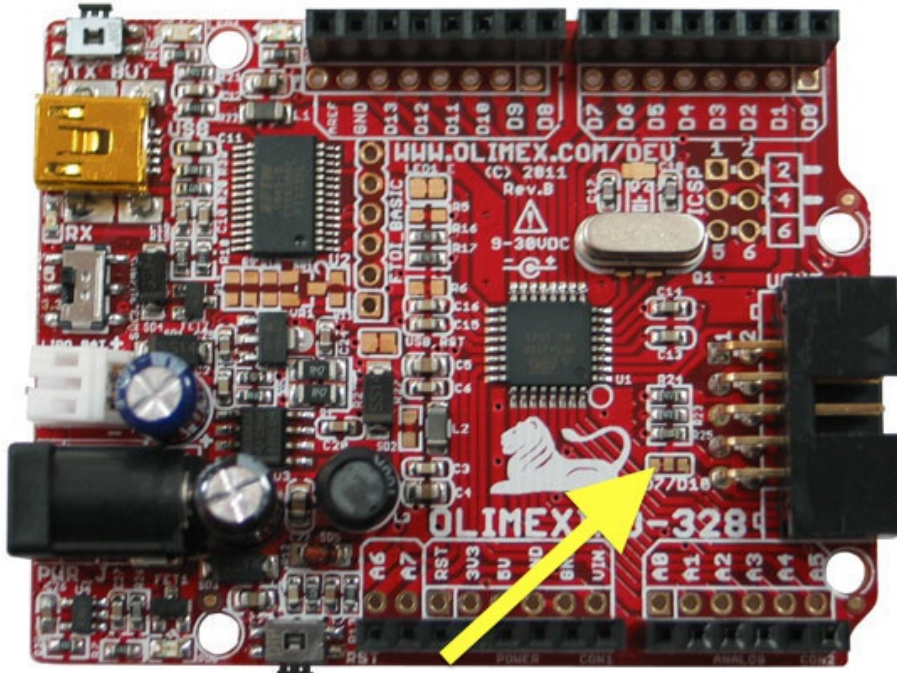
## Présentation

Carte OLIMEX

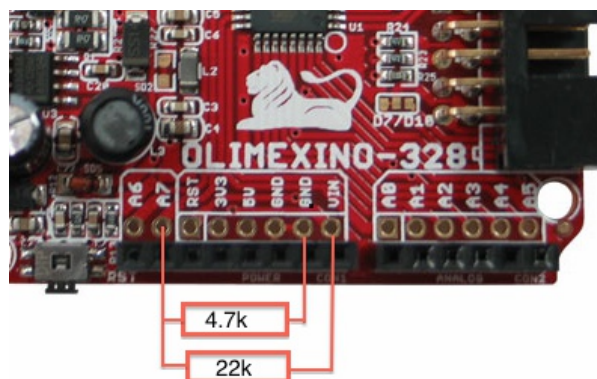
ATmega328 processeur 8 bits avec 32Koctets de flash, 1Koctets EEPROM, 2Koctets RAM

## Assemblage des éléments

- Le switch de sélection de la tension d'alimentation doit être sur la position **3.3V**
- Sur la carte OLIMEX, modification du strap D7/D10, à positionner en D10 au lieu de D7.



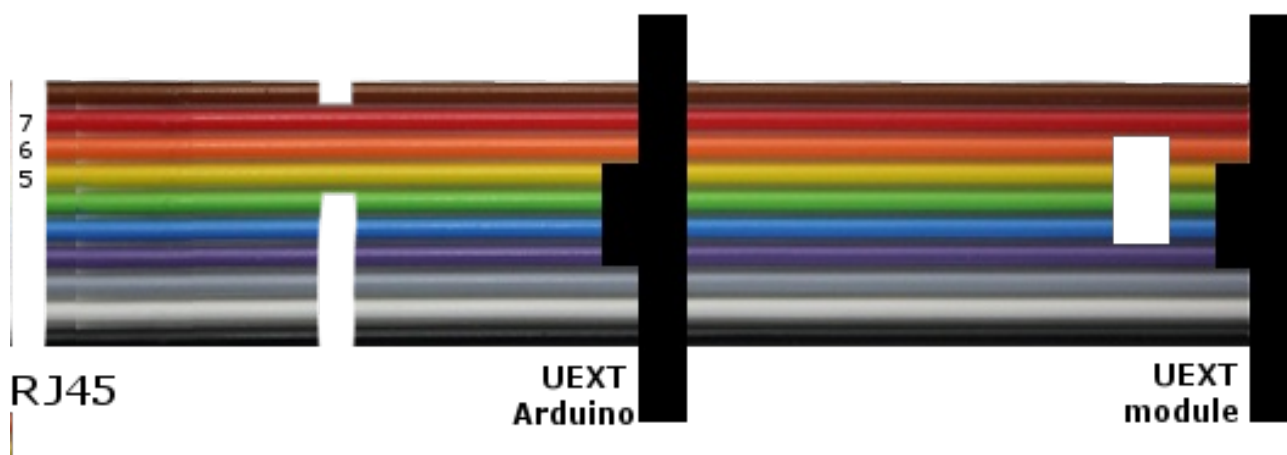
- Pont diviseur de mesure de la tension d'alimentation sur le point A7 ; l'échelle peut être adaptée aux valeurs des résistances du pont diviseur par la commande **VINMAX**



- Confection de la nappe reliant les connecteurs EXT de la carte OLIMEX et de la carte ethernet et la RJ45 du régulateur :
  - les fils 3, 4, 5 et 6 sont coupés entre le connecteur UEXT de la carte Arduino et

l'UEXT du module ethernet,

- les fils 2, 3 et 4 de l'UEXT de la carte Arduino sont renvoyés sur les points 7, 6 et 5 de la RJ45 du régulateur solaire.



# Usage des boutons de la carte OLIMEX

## **RESET**

La touche RESET, près du connecteur rond d'alimentation, permet de redémarrer le programme.

## **BUT**

La touche BUT, près du connecteur USB, tenue enfoncée durant une seconde, positionne le montage en mode maintenance, la carte ethernet s'éteint ; la même manoeuvre permet le retour à la normale.

## **BUT et RESET**

Le maintien de la touche BUT durant un RESET redémarre le programme sans lecture des paramètres de configuration stockés en EEPROM, et permet de retrouver la valeurs par défaut :

Adresse MAC :        54 :55 :56 :21 :22 :23

Adresse IP :         192.168.1.25

Netmask :            255.255.255.0

Passerelle :         192.168.1.1

## Référence des commandes

La syntaxe générale des commandes est :

[http://adresse\\_ip\[:port\]/commande\[?p1=val1\[&p2=val2\]...\]](http://adresse_ip[:port]/commande[?p1=val1[&p2=val2]...])

Les commandes, présentées ci-dessous en majuscules ne sont pas 'case sensitive', par ordre alphabétique :

### ACC1 ...

### ACC2 ...

Paramètre : *P1=aabbccdde*

Restreint l'accès aux plages d'adresses indiquées en paramètres **aabbccdd** est l'adresse en hexadécimal, **ee** est le nombre de bits de netmask, de 1 à 32 (24 pour un réseau avec un netmask 255.255.255.0, 32 pour une adresse unique, etc ...).

Deux plages d'adresses peuvent être définies, la première par la commande ACC1, la seconde par ACC2 ; à noter que la seconde n'est active que si la première est définie.

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

Exemple d'usage restreignant l'accès depuis le réseau 172.16.0.0/16 :

```
PARAM=$(printf "p1=%02x%02x%02x%02x%02X" 172 16 0 0 16)
curl -s -m 5 -o- "http://adresse_ip/acc1?${PARAM}"
```

### ADDR ...

Paramètre : *P1=aabbccdd*

Définit l'adresse IP de la carte OLIMEX

**aabbccdd** est l'adresse en hexadécimal, la valeur 00000000 active le mode client DHCP.

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

Exemple d'usage fixant l'adresse à 192.168.8.32 :

```
PARAM=$(printf "p1=%02x%02x%02x%02x" 192 168 8 32)
curl -s -m 5 -o- "http://adresse_ip/addr?${PARAM}"
```

Voir également les commandes **MACA**, **MASK**, **GATE**

### CMD

Non implémentée

Envoi de commandes au régulateur Epsolar

### DELESTAGE [...]

Paramètre optionnel : *P1=nnn*

Coupe l'utilisation et la carte réseau pour une durée de nnn secondes ; **nnn** en décimal, doit être compris entre 20 et 86400 (24 heures).

Appelée sans paramètre la coupure est de 20 secondes.

A partir de la version 0.8d du programme, la réponse comprend le statut,



l'heure, la durée de délestage en secondes et OK si la commande est acceptée, ERR sinon.

La consommation résiduelle est de 15mA.

Exemples d'usage :

```
wget -O - "http://adresse\_ip/deslestage?P1=300" 2>/dev/null  
rend :
```

```
<html><head><title>Lndk Tracer</title></head>  
<body><h1>Lndk Tracer version v0.8d</h1>  
0322 0000009A 0000012C OK  
</body>  
</html>
```

```
wget -O - "http://adresse\_ip/deslestage?P1=15" 2>/dev/null  
rend :
```

```
<html><head><title>Lndk Tracer</title></head>  
<body><h1>Lndk Tracer version v0.8d</h1>  
2120 54C27D4F 00000000 ERR <==  
</body>  
</html>
```

## DUMP

*Pas de paramètre*

Renvoie Status, Heure et les octets du dernier échange série avec le régulateur solaire :

Exemple d'usage :

```
curl -s -m 5 -o- "http://adresse\_ip/dump" 2>/dev/null  
rend :
```

```
<html><head><title>Lndk Tracer</title></head>  
<body><h1>Lndk Tracer version v0.8c</h1>  
2920 54C267E5 EB 90 EB 90 EB 90 00 A0 18 18 05 27 07 00 00 09 00 4D  
04 C4 05 01 00 00 35 00 00 00 01 2D 10 00 01 A7 63 7F A7 63 00 00  
</body>  
</html>
```

## EEPROM [...]

*Paramètre optionnel : P1=nnn*

Affichage du contenu de l'EEPROM

## GATE

*Paramètre : P1=aabbccdd*

Définition de la gateway réseau

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

Exemple d'usage fixant l'adresse à 192.168.8.32 :

```
PARAM=$(printf "p1=%02x%02x%02x%02x" 192 168 8 1)  
curl -s -m 5 -o- "http://adresse\_ip/gate?\${PARAM}"
```

Voir également les commandes **ADDR**, **MACA**, **MASK**

## HISTO

Non implémentée

## LOAD ...

Paramètre :  $P1=n$

Coupe ( $n=0$ ) ou établissement ( $n=1$ ) de la charge

## MACA ...

Paramètre :  $P1=aabbccdeeff$

Définition de la Mac Address en EEPROM

Valeur par défaut 0x54,0x55,0x56,0x21,0x22,0x23

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

Exemple d'usage fixant l'adresse Mac à 54:3A:56:21:7C:24 :

```
curl -s -m 5 -o- "http://adresse_ip/macap1=543A56217C24"
```

Voir également les commandes **ADDR**, **MASK**, **GATE**

## MASK ...

Paramètre :  $P1=aabbccdd$

Définition du Netmask réseau en EEPROM

Valeur par défaut 255.255.255.0

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

Exemple d'usage fixant le masque d'adresse à 255.255.255.240 (/28) :

```
curl -s -m 5 -o- "http://adresse_ip/maskp1=FFFFFF0"
```

Voir également les commandes **ADDR**, **GATE**, **MACA**

## MEMORY

Pas de paramètre

Rend la quantité de mémoire RAM encore disponible

## MESURES

Pas de paramètre

Cette commande permet de remonter les mesures du chargeur solaire sous forme hexadécimale dans les réponses HTTP, ci-dessous un extrait de la réponse à cette commande :

```
0A10 000005C3 04D8 057C 001A 044B 05C8 000B 0A 00 00 24 00 00 00 0A 2B
```

Analyse de la ligne ci-dessus :

Etat du programme	0A10	* voir détails ci-dessous
Date (time_t depuis 1/1/1970)	000005C3	
Tension batterie (en 10mV)	04D8	1240 * 10 = 12400mV
Tension panneaux	057C	1404 * 10 = 14040mV
Consommation	001A	26 * 10 = 260mA
Tension batterie déchargée	044B	1099 * 10 = 10990mV
Tension batterie pleine	05C8	1480 * 10 = 14800mV
Courant de charge	000B	11 x 10 = 110mA
Utilisation activée	0A	Somme des états
Surcharge	00	"
Court-circuit	00	"

Etat de la charge (SOC)	24	Moyenne : 36%
Batterie trop chargée	00	Somme des états
Batterie trop déchargée	00	"
Batterie pleine	00	"
Charge en cours	0A	"
Température	2B	Moyenne : 13°C (43 - 30)

\* Détails de l'état du programme :

```

octet 1 : 0=Time no set,
           1=Time set waiting for sync,
           2=Time set
octet 2 : Nombre de mesures en hexa de 0 a E (14)
           Si la valeur est F, 15 mesures ou plus
octet 3 : Mode ECO
           0=désactivé,
           1=standby (cf la commande STANDBY)
           2=recherche d'une périodicité,
           3=activé
octet 4 : Phase dans la boucle principale du programme Arduino
           En général 0.

```

Une suite de relevés successifs permet de suivre l'évolution du programme : synchronisation du mode économie de consommation à la deuxième ligne (0A30), mise à l'heure à la quatrième (2A20 et 515406CD), etc ...

```

0A20 000005C3 04D8 057C 001A 044B 05C8 000B 0A 00 00 24 00 00 00 0A 2B
0A30 000006EE 04D9 057F 001A 044B 05C8 000C 0A 00 00 25 00 00 00 0A 2B
0A30 0000081A 04DA 0584 0019 044B 05C8 000C 0A 00 00 25 00 00 00 0A 2B
2A30 515406CD 04DB 0577 0019 044B 05C8 000E 0A 00 00 25 00 00 00 0A 2B
2A30 515407F9 04DB 05A1 0019 044B 05C8 0017 0A 00 00 25 00 00 00 0A 2B

```

## MODEECO ...

*Paramètre : P1=n*

La consommation de la carte OLIMEX et de son interface réseau est d'environ 100mA sous 12V, mais tombe à 15mA lorsque l'interface réseau est désactivée.

Par défaut le programme recherche s'il y a une périodicité dans les accès, ce qui est en général le cas si l'accès vient d'un cron, de Cacti, etc ..., et s'il la détecte, la carte réseau n'est activée qu'aux moments où une requête est attendue; cette fonctionnalité permet de ramener la consommation moyenne du montage à ~25mA.

La périodicité acceptée par le programme va de 2 minutes à 2 heures, la remise en ligne de la carte réseau est faite à 10% du temps avec un minimum de 20 secondes.

En mode délestage, la carte réseau est désactivée, la consommation est de 15mA durant tout le délestage.

La commande **MODEECO?P1=0** permet de désactiver ce mode, P1=1 réactive le mode recherche de périodicité.

*Voir également la commande **STANDBY***

## PARAM

*Commande non implémentée*

Interrogation des paramètres de configuration du chargeur solaire

## PORT ...

Paramètre : *P1=aa*

Définition du port réseau HTTP, par défaut 80

La carte doit être re-démarrée pour la prise en compte de la modification, ou commande RESET si version  $\geq 0.8e$ .

## RESET

Commande implémentée à partir de la **version 0.8e**

Ré-initialisation du port série et de la configuration réseau

## SERIALDEBUG

*Pas de paramètre*

Permet d'analyser les commandes envoyées vers le régulateur Epsolar en connectant un câble USB vers un micro-ordinateur; les commandes sont présentées en hexa.

## STANDBY ...

Paramètres : *P1=aa&P2=bb[&P3=cc&P4=dd]*

*P1 et P2 sont obligatoires, P3 et P4 optionnels*

*Toutes les valeurs sont en décimal*

Le mode ECO est un bon moyen d'économiser de l'énergie mais est totalement inapproprié lorsque la production est surabondante, aussi est-il désactivé ou réactivé lorsque que certains critères sont atteints :

- lorsque la tension batterie dépasse les 13200mV (26400mV pour un système en 24V) et que l'état de charge (SOC) est supérieur ou égal à 64 (sur une échelle de 100), le mode ECO est mis en standby.
- lorsque la tension batterie descend sous les 13000mv ou que l'état de charge est inférieur à 58, le mode ECO ré-enclanche une recherche de périodicité dans les accès.

La commande STANDBY permet de redéfinir ces seuils : **aa** valeur haute de tension en mV désactivant le mode ECO, **bb** valeur basse ré-enclanchant le mode ECO, **cc** valeur haute de SOC et **dd** la valeur basse.

Exemple d'usage relevant ces seuils :

```
curl -s -m 5 -o- "http://adresse_ip/standby?p1=13500&p2=13100&p3=75&p4=70"
```

## TIME [...]

Paramètres *P1 et P2 optionnels* : *[P1=mmm[&P2=nn]]*

Permet la mise à l'heure de la carte OLIMEX, mmm (en décimal) est le *time\_t*, secondes depuis 1/1/1970, nn la durée de validité de cette mise à l'heure en secondes; si non précisée cette durée est de 1200 secondes (20 minutes).

Appelée sans paramètre la commande se comporte comme la commande UPTIME.

Exemple d'usage :

```
curl -s -m 5 -o- "http://adresse_ip/time?p1=$(date '+%s')&p2=3600"
```

## UPTIME

*Pas de paramètre*

Rend les 4 octets d'état, la date du micro-contrôleur (secondes depuis [Epoch Unix](#) en hexa) et la durée d'uptime.

Exemple d'usage :

```
curl -s -m 5 -o- "http://adresse_ip/uptime"
```

rend

```
2F30 54BD1C2D 00003745
```

```
date -r $((0x54BD1C2D)) "+%d/%m/%Y %H:%M:%S" (19/01/2015 16:01:01)
```

```
uptime de $((0x00003745)), soit 14149 secondes.
```

## VINMAX [...]

*Paramètres P1 optionnel, P2 optionnel : P1=[mmm[&P2=n]]*

Définit la valeur maximale (mmm en décimal) de mesure de tension par la commande **VOLTAGE**, permet d'adapter en fonction du pont de résistances mises en place sur le point de mesure **A7** ; à titre indicatif pour un pont de 22kohms + 4,7kohms, la valeur est de l'ordre de 30650.

Le paramètre P2 si présent et égal à 1, valide l'écriture en EEPROM de cette valeur, qui sera dès lors retrouvée aux démarrages suivants.

Exemple d'usage, avec sauvegarde en EEPROM :

```
curl -s -m 5 -o- "http://adresse_ip/vinmax?p1=30650&p2=1"
```

Appelée sans paramètre, rend la valeur du VINMAX en cours (20250 dans l'exemple ci-dessous) :

```
2120 58411B3F 624 (12300 mV) Max:20250
```

*Voir également la commande **VOLTAGE***

## VOLTAGE

*Pas de paramètre*

Mesure de la tension d'alimentation de la carte si elle est alimenté par son connecteur jack ; un pont de résistances alimentant le point A7 doit être mis en place entre les points **VIN** et **GROUND**.

```
wget -q0- http://adresse_ip/voltage 2>/dev/null
```

rend

```
2030 584095B4 622 632 631 631 632 631 631 (12482 mV)
```

```
2030 : le status,
```

```
584095B4 : la date (en hexadécimal),
```

```
622 ... 631 : en décimal, les 7 mesures sur 10 bits effectuées, pour le calcul seules les 5 les plus significatives sont retenues
```

```
12482 mV : mesure de la tension.
```

*Voir également la commande **VINMAX***

## WATCHDOG

*Paramètres [P1=mm[&P2=nn]]*

*Commande non implémentée*

Coupure momentanée de la charge pendant **nn** secondes si aucune aucune activité réseau pendant une période de **mm** secondes.

---

## NOTES