

## Alimentation cartes

ATTENTION, les cartes n'étant pas protégées contre les inversions de polarité, il ne faut pas inverser le VIN et le 0V.

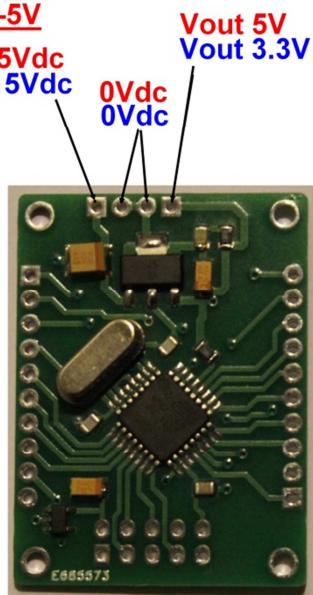
### Cartes UNIVMYS-5V et UNIVMYS-3V3

Alimentation avec une plage de tension continue

Carte UNIVMYS-3V3

Carte UNIVMYS-5V

VIN: 7Vdc à 15Vdc  
VIN: 5.5Vdc à 15Vdc

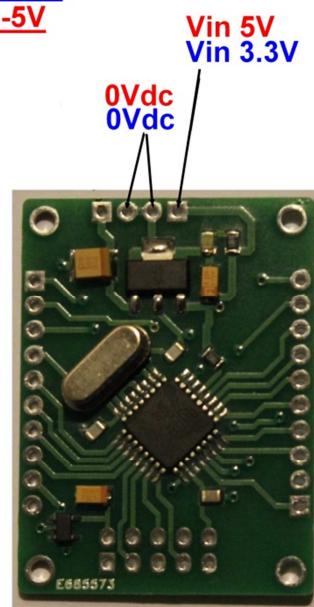


Alimentation avec une tension fixe

Carte UNIVMYS-3V3

Carte UNIVMYS-5V

Vin 5V  
Vin 3.3V



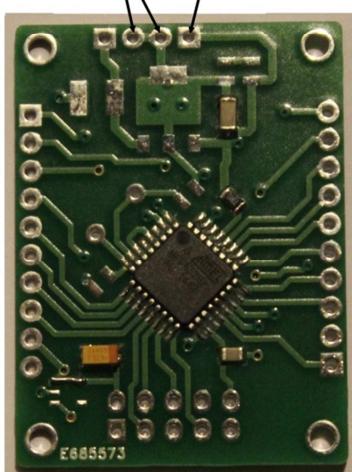
### Cartes UNIVMYS-PIL

Alimentation avec une plage de tension continue

Carte UNIVMYS-PIL

VIN: 1.8Vdc à 3.5Vdc

0Vdc



## Programmation carte

Les cartes sont déjà programmées avec un bootloader arduino.

Pour les cartes UNIVMYS-5V et UNIVMYS-3V3, config officielle ARDUINO PRO MINI.

Pour la version UNIVMYS-PIL, configuration quartz interne 8MHZ et arrêt de fonctionnement si alim < à 1.8V.

Pour info, voici les fichiers de config des bootloader (programmateur AVR ISP MKII).

Code à ajouter en fin du fichier board.txt se trouvant dans le répertoire **PROGRAM FILES\ARDUINO\HARDWARE\ARDUINO\**

### *Carte UNIVMYS-PIL*

```
atmega328bb.name=AVRISP mkII UNIVMYS-PIL ATmega328p (Q:8Mhz interne, alim min 1.8V)
```

```
atmega328bb.upload.using=arduino:avrispmkii
atmega328bb.upload.maximum_size=30720
atmega328bb.upload.speed=57600
```

```
atmega328bb.bootloader.low_fuses=0xE2
atmega328bb.bootloader.high_fuses=0xDA
atmega328bb.bootloader.extended_fuses=0x06
atmega328bb.bootloader.path=arduino:atmega
atmega328bb.bootloader.file=ATmegaBOOT_168_atmega328_pro_8MHz.hex
atmega328bb.bootloader.unlock_bits=0x3F
atmega328bb.bootloader.lock_bits=0x0F
```

```
atmega328bb.build.mcu=atmega328p
atmega328bb.build.f_cpu=8000000L
atmega328bb.build.core=arduino:arduino
atmega328bb.build.variant=standard
```

### *Carte UNIVMYS-3V3*

```
pro328edm.name=AVRISP mkII UNIVMYS-3V3 ATmega328p (Q:8 MHz,alim 3V3)
```

```
pro328edm.upload.protocol=arduino:avrispmkii
pro328edm.upload.maximum_size=30720
pro328edm.upload.speed=57600
```

```
pro328edm.bootloader.low_fuses=0xFF
pro328edm.bootloader.high_fuses=0xDA
pro328edm.bootloader.extended_fuses=0x05
pro328edm.bootloader.path=atmega
pro328edm.bootloader.file=ATmegaBOOT_168_atmega328_pro_8MHz.hex
pro328edm.bootloader.unlock_bits=0x3F
pro328edm.bootloader.lock_bits=0x0F
```

```
pro328edm.build.mcu=atmega328p
pro328edm.build.f_cpu=8000000L
pro328edm.build.core=arduino
pro328edm.build.variant=standard
```

## Carte UNIVMYS-5V

pro5v328edm.name=AVRISP mkII UNIVMYS-5V ATmega328p (Q:16 MHz,alim 5V)

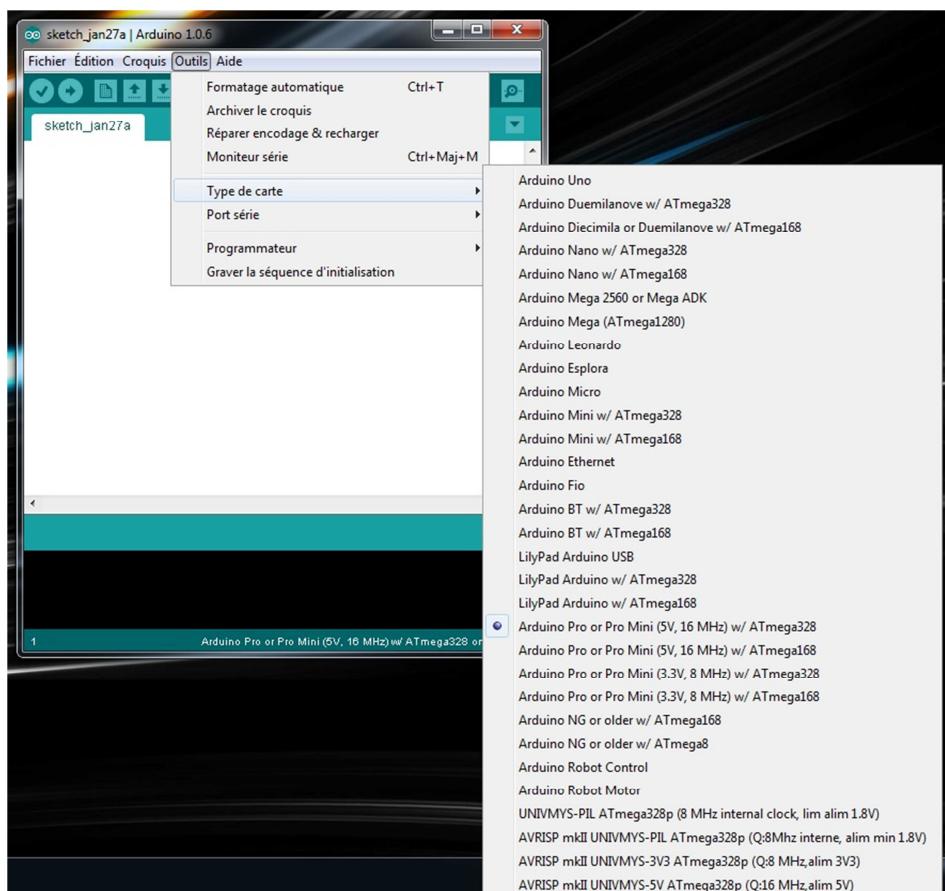
```
pro5v328edm.upload.using=arduino:avrispmkii  
pro5v328edm.upload.maximum_size=30720  
pro5v328edm.upload.speed=57600
```

```
pro5v328edm.bootloader.low_fuses=0xFF  
pro5v328edm.bootloader.high_fuses=0xDA  
pro5v328edm.bootloader.extended_fuses=0x05  
pro5v328edm.bootloader.path=atmega  
pro5v328edm.bootloader.file=ATmegaBOOT_168_atmega328.hex  
pro5v328edm.bootloader.unlock_bits=0x3F  
pro5v328edm.bootloader.lock_bits=0x0F
```

```
pro5v328edm.build.mcu=atmega328p  
pro5v328edm.build.f_cpu=16000000L  
pro5v328edm.build.core=arduino  
pro5v328edm.build.variant=standard
```

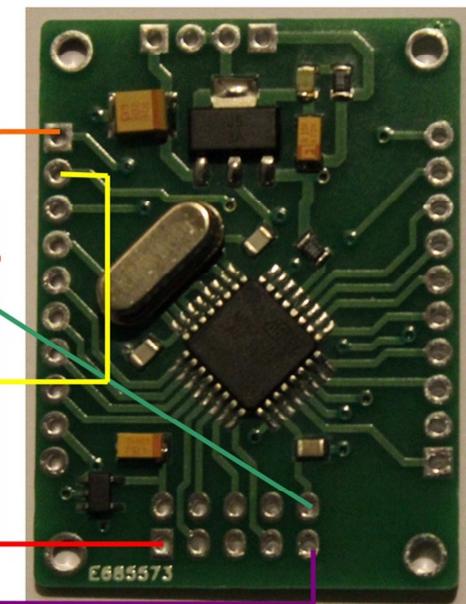
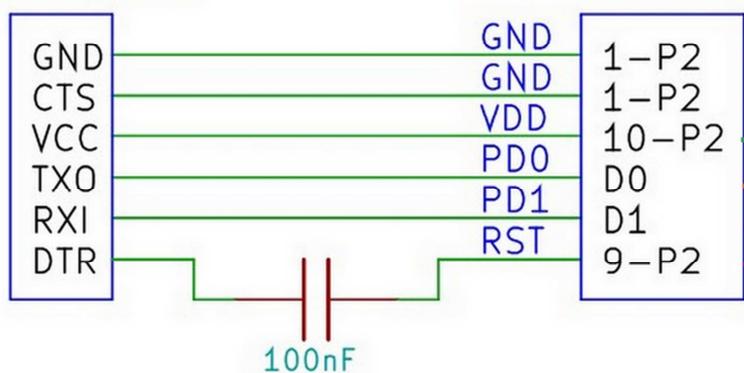
1°) Pour téléverser les sketchs avec un programmateur « type liaison série », config normale ARDUINO :

- Carte UNIVMYS-PIL → Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
- Carte UNIVMYS-3V3 → Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
- Carte UNIVMYS-5V → Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328



## Connection programmateur series:

### PROGRAMMATEUR



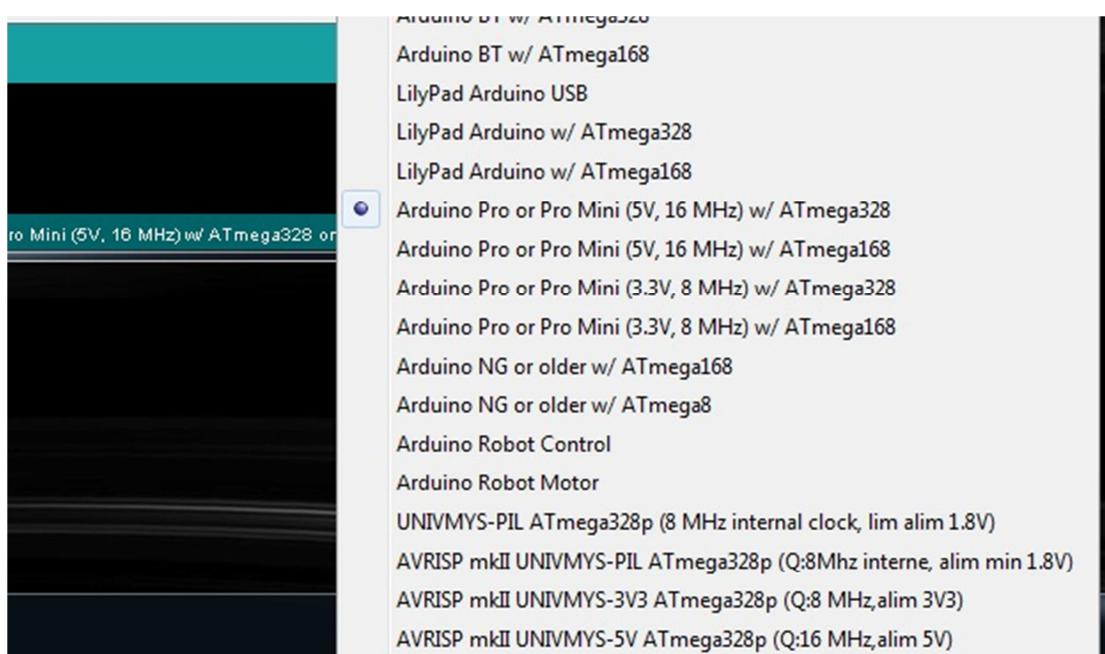
Le condensateur de 100nF a été livré avec les cartes (composant rouge).

2°) Pour téléverser les sketchs avec un programmateur « type ISP », config :

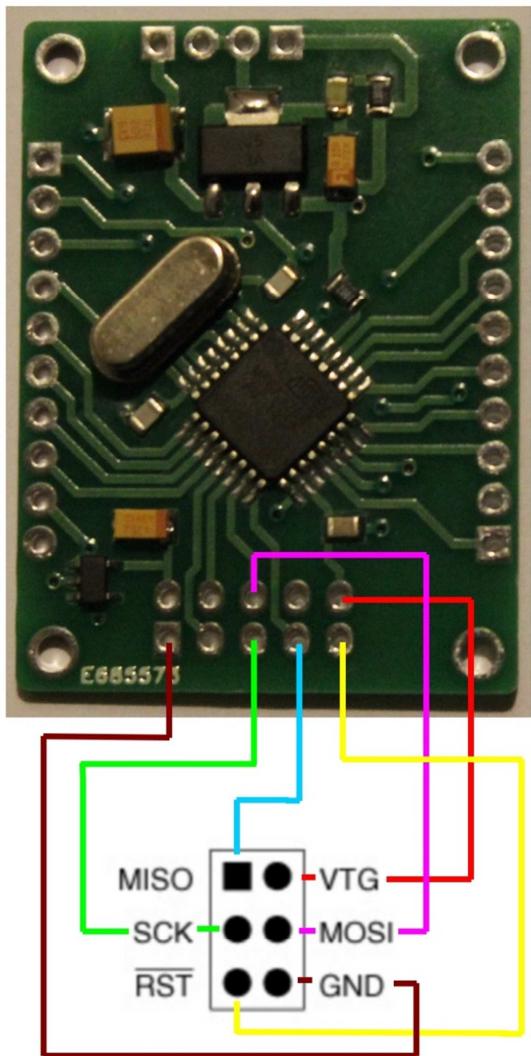
- Carte *UNIVMYS-PIL* → AVRISP mkII UNIVMYS-PIL ATmega328p (Q:8MHz interne, alim min 1.8V)
- Carte *UNIVMYS-3V3* → AVRISP mkII UNIVMYS-3V3 ATmega328p (Q:8 MHz,alim 3V3)
- Carte *UNIVMYS-5V* → AVRISP mkII UNIVMYS-5V ATmega328p (Q:16 MHz,alim 5V)

Note: Ajouter les lignes de code de la page 2 et 3 en fin du fichier board.txt se trouvant dans le répertoire **PROGRAM FILES\ARDUINO\HARDWARE\ARDUINO\**.

Ces lignes de code sont pour un programmateur ISP AVR MKII, à adapter suivant le type de programmateur.

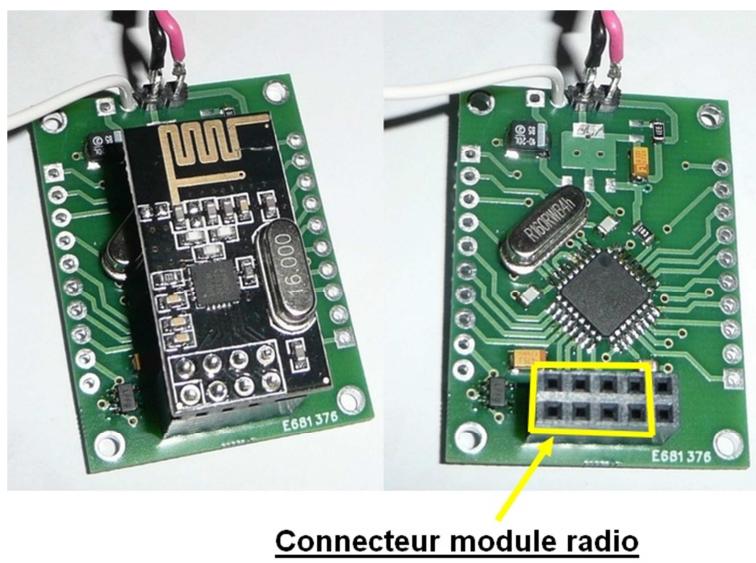


## Connection programmeur ISP:

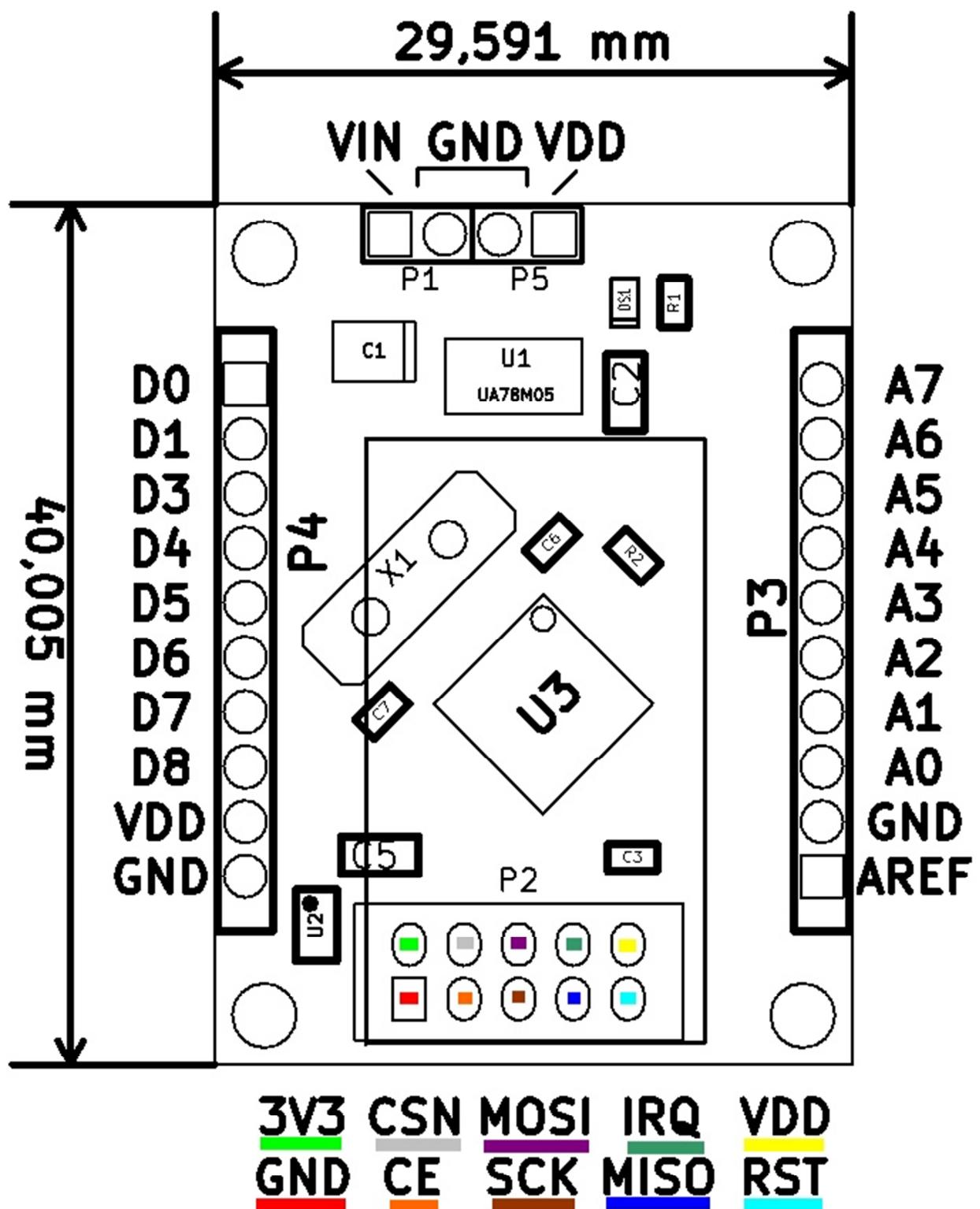


## Montage module radio NRF24L01+

Le module radio est monté sur la partie gauche du connecteur 10 points femelle.



## PIN affectation carte



## Schéma carte

