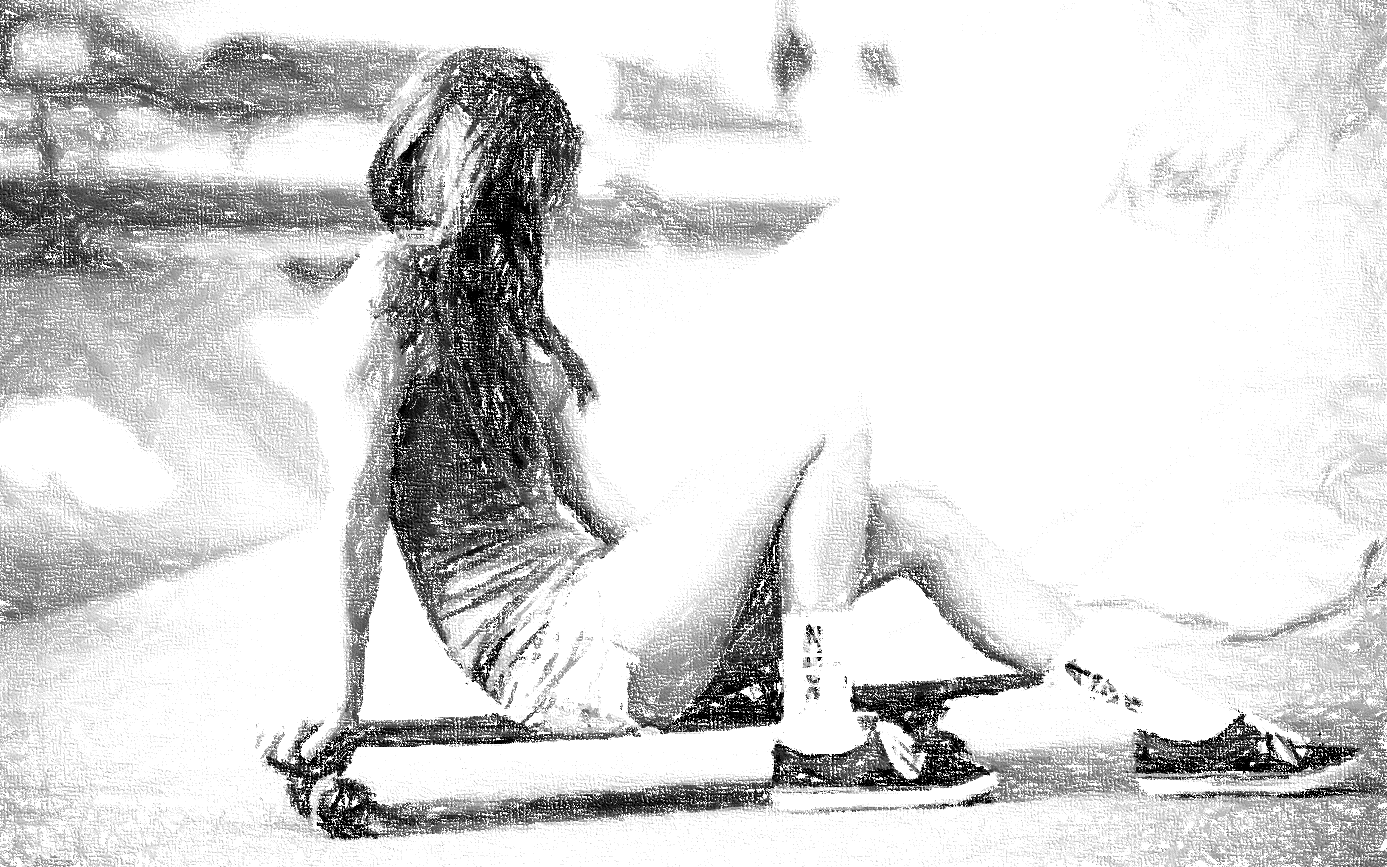
Eboard

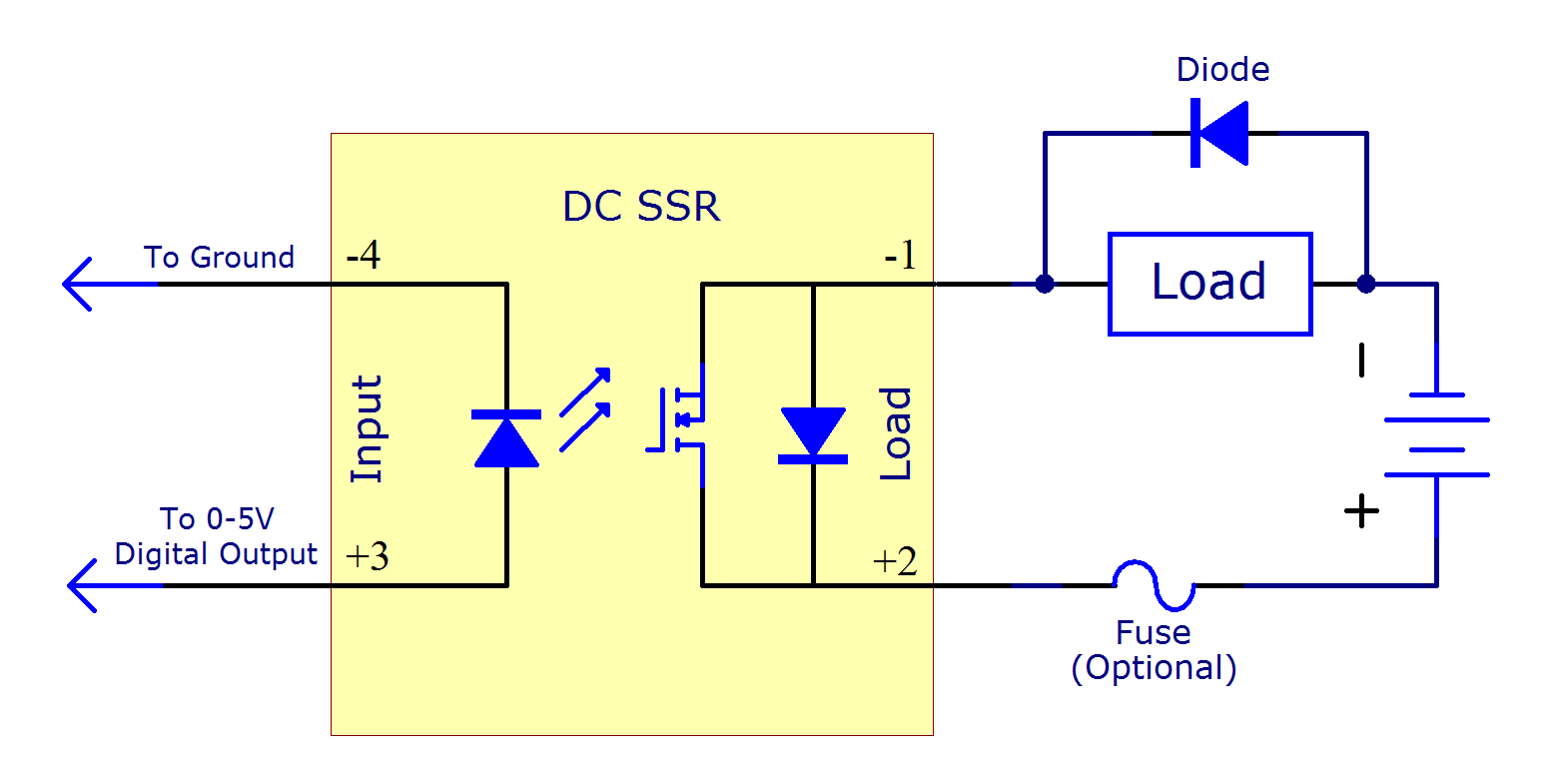


# Mécanique

# Electronique

## Solide State Relay

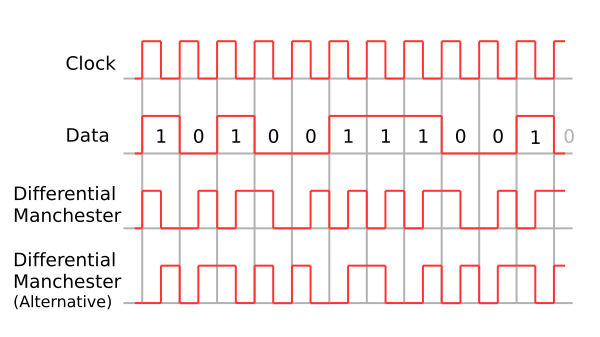
Le solide state relay est un semi-conducteur de puissance. Il est en général composé de thyristor mais dans notre cas c’est un mosfet canal N, il faut donc faire attention à la diode intrinsèque pour le câblage de la tension de commande. Qui dit mosfet de puissance dit low et donc une plus grande capacité sur la gate, ce qui diminue la fréquence de commutation. La gate est photosensible et est commandée par un opto coupleur.



## Module radio

### Communication

Le problème avec ces modules radio est l’immunité au bruit. Un des moyens pour contrer cela est le code différentiel Manchester qui ne se basse pas sur l’amplitude des signaux, mais sur leur transitions.



Exemple : Commencement en flanc montant du clock. La première moitié de clock représente les Data. Dans le premier cas « 0 » pas de changement et « 1 »transition d’état. Inversement dans le deuxième cas. La deuxième moitié représente le clock qui effectue une transition.

New Arduino library for 433 MHz AM radio modules :

<https://en.wikipedia.org/wiki/Differential_Manchester_encoding>

<http://www.buildcircuit.com/how-to-use-rf-module-with-arduino/>

Tuto :

<http://www.wes.id.au/2013/07/decoding-and-sending-433mhz-rf-codes-with-arduino-and-rc-switch/>

On peut également faire cela grâce à du hardware.

<http://www.avrfreaks.net/forum/433mhz-range>

<http://www.romanblack.com/RF/cheapRFmodules.htm>

### Transmitter

Model: MX-FS-03V

Transmission Distance :20-200 m (dependent on supply voltage)

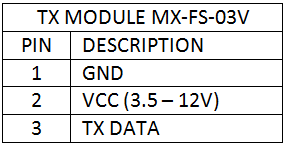
Operating Voltage :3.5-12V

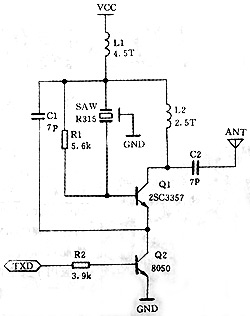
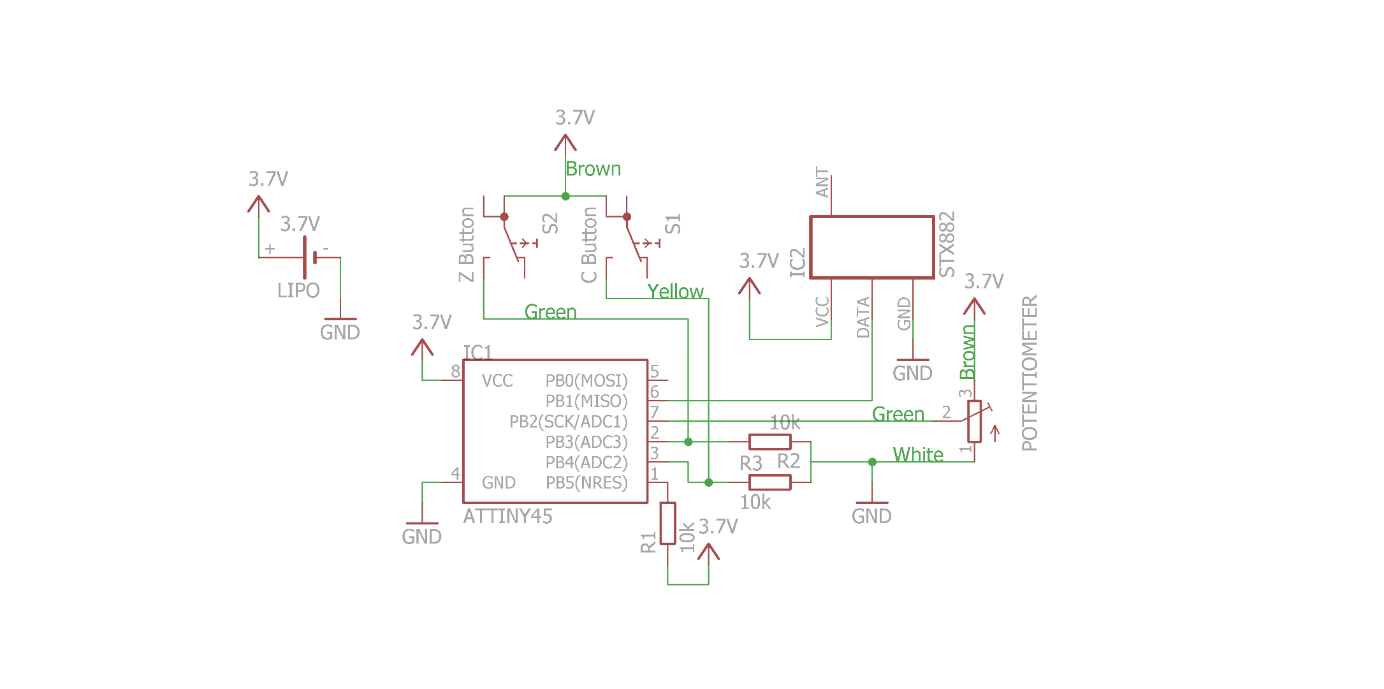
Dimensions: 19 \* 19mm

AM transfer rate: 4Kb/s (4000 bits per second)

Transmission power: 10mW

Emission frequency: 433M



### Receiver

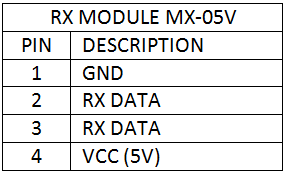
Model: MX-05V

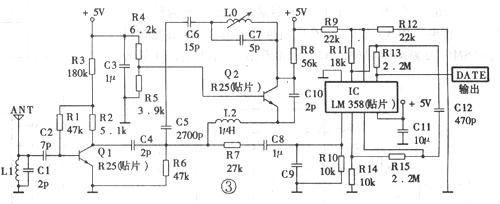
Working voltage: 5V DC quiescent current: 4mA

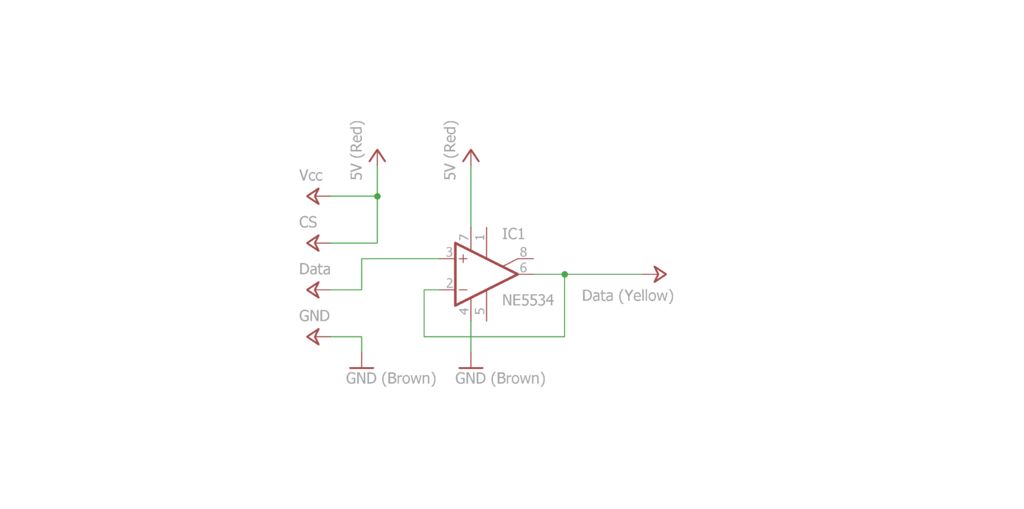
Receiver Frequency: 433.92MHZ

Receiver sensitivity:-105DB

Size: 30 \* 14 \* 7mm







NOTES:

When using an external antenna a 1/4 wavelength is recommended. Ideally use 50 impedance ohm single-core wire, the length of the antenna 433M is about 17cm (1/4 wavelength). When locating the receiver antenna keep it as far away as possible from shielded areas, high voltages, and any other possible interfering frequencies.

* Can transmit to more than one receiver
* Can receive data from more than one transmitter

### Antenne

<http://www.instructables.com/id/433-MHz-Coil-loaded-antenna/>

### Amélioration

NRF24L01, 2.4GHz Wireless RF Module :

<http://www.ebay.com/itm/10PCS-Arduino-NRF24L01-2-4GHz-Wireless-RF-Transceiver-Module-New-/401088440858?hash=item5d62bbea1a:g:bxQAAOSwGXtXhfcs>

<http://www.dx.com/fr/p/868mhz-high-performance-si4463-rf-wireless-module-with-shielding-cover-for-arduino-361360#.WCYsTfnhCM8>

Plusieurs canaux :

<http://www.dx.com/fr/p/4-channel-433mhz-wireless-remote-control-kit-w-fixed-code-antenna-433797#.WCYskfnhCM8>

Cage de Faradet :

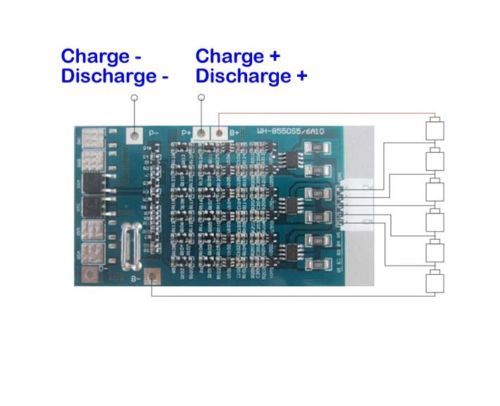
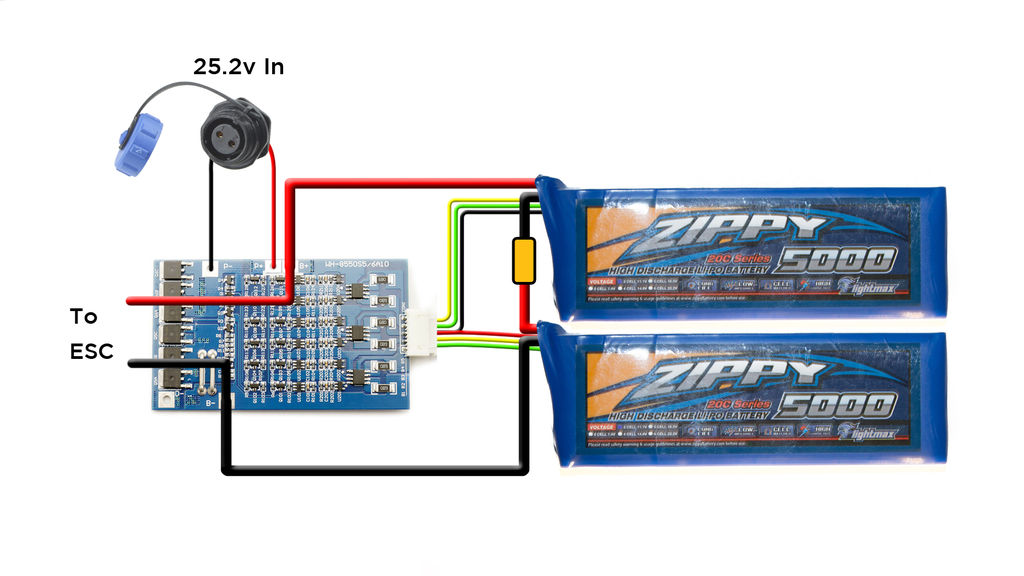
<http://www.ebay.nl/sch/i.html?_from=R40&_trksid=m570.l1313&_nkw=433MHz+Superheterodyne+3400+RF+Transmitter+and+Receiver&_sacat=0>

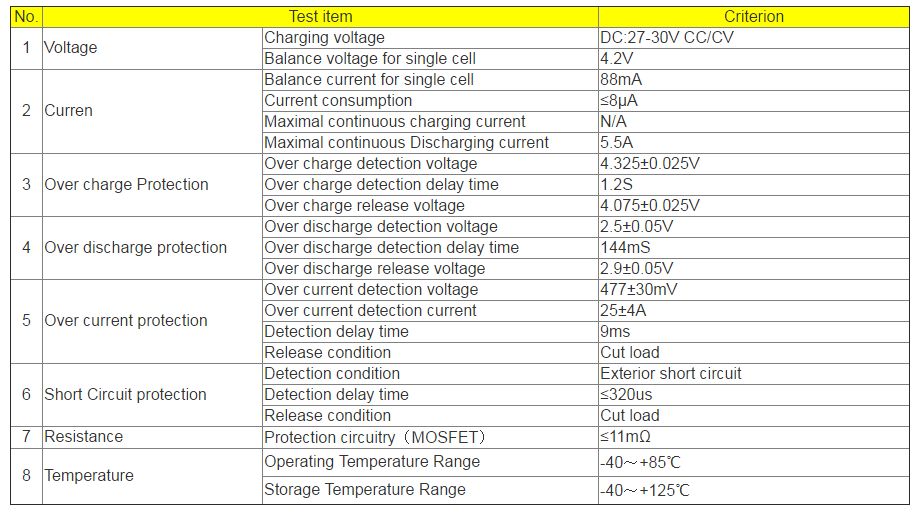
Haute calitée :

APC220

http://www.open.com.au/mikem/arduino/VirtualWire.pdf

## BMS

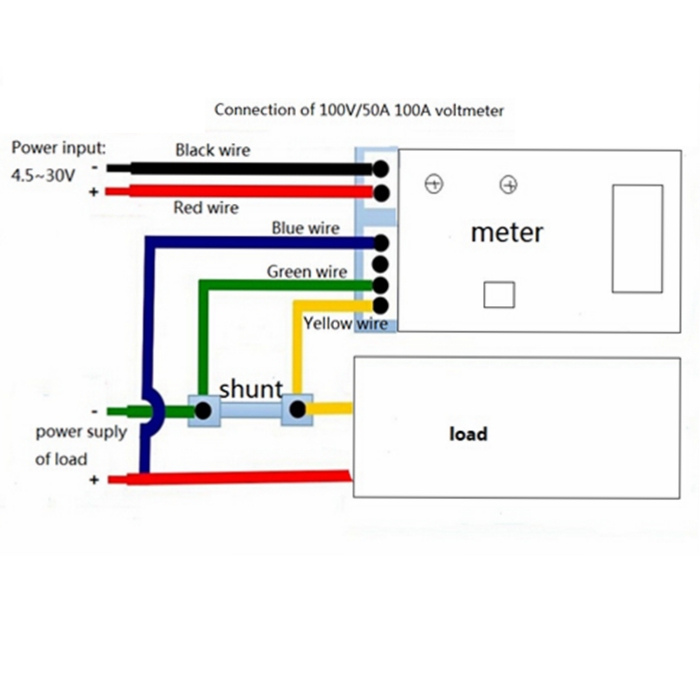
 



## Power supply

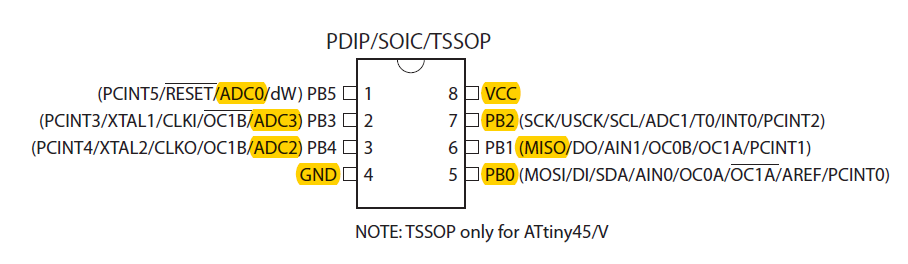


## Affichage tension/courant



# Programmation

## ATTINY



### Librairies

Installer une librairie

<https://www.arduino.cc/en/Guide/Libraries>

Programmer l’ATtiny

Lien de téléchargement : <http://playground.arduino.cc/ComponentLib/Servo>

Modifications

Dans « SoftwareServo.h » il faut changer :

*#include <WProgram.h> 🢡 #include "Arduino.h"*

Communication sans fils

Virtualwire.h

Lien de téléchargement : <https://www.pjrc.com/teensy/td_libs_VirtualWire.html>

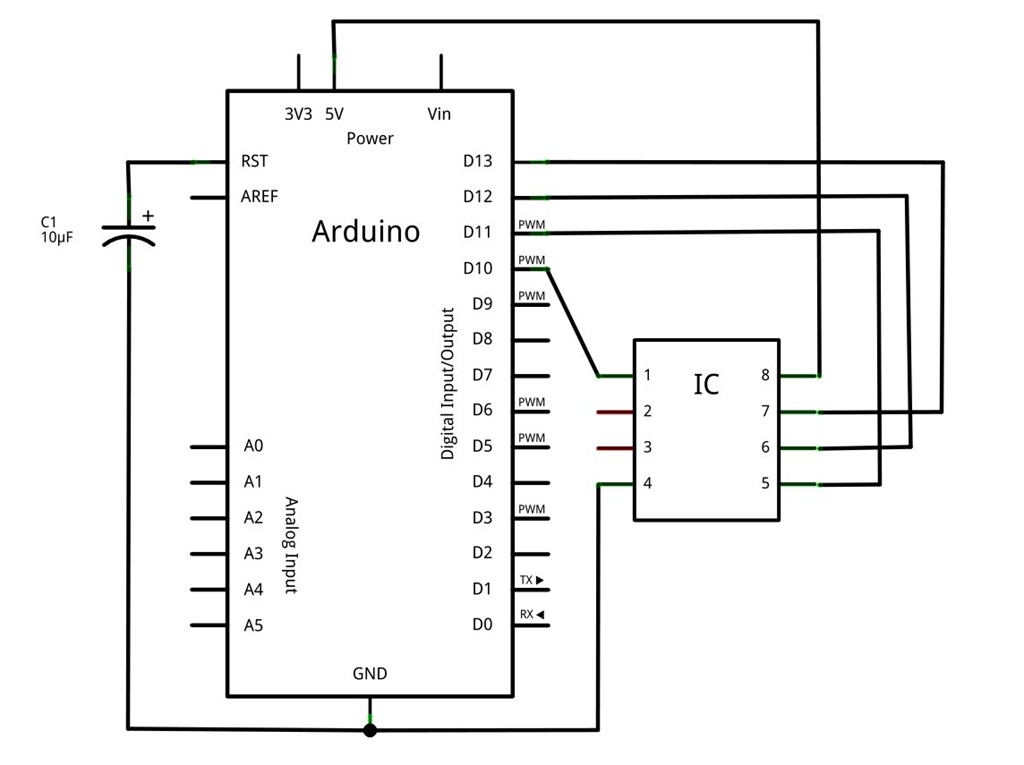
Envoie de chiffre : <http://forum.arduino.cc/index.php?topic=33248.0>

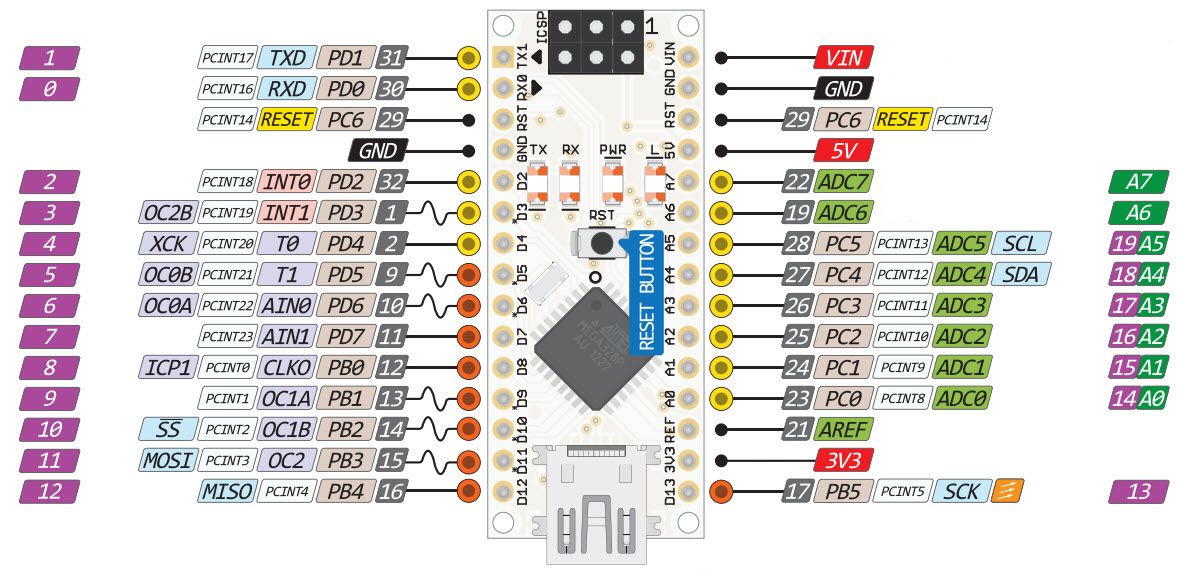
### Programmation

La programmation se fait par SPI qui est synchrone. Cela signifie que le signal de Clock accompagne le signal de données. **MOSI** - Master Out Slave In; **MISO** – “Master In Slave Out”; **SCK** – “Clock signal” du maître à l’esclave; **SS** – “Slave Select” signale choisit le dispositif esclave. SPI permet d’avoir plusieurs dispositifs sur son BUS. Comme il possède un Clock, le SPI peut opérer plus rapidement qu’un UART. Connection SPI :



**Comment programmer un ATtiny :** <http://highlowtech.org/?p=1695>





# Liens

Module Radio

<http://forum.hobbycomponents.com/viewtopic.php?f=39&t=1324>

<http://www.instructables.com/id/RF-315433-MHz-Transmitter-receiver-Module-and-Ardu/?ALLSTEPS>

<http://arduinobasics.blogspot.ch/2014/06/433-mhz-rf-module-with-arduino-tutorial.html>

<https://www.pjrc.com/teensy/td_libs_VirtualWire.html>

Chargeur :

<http://www.instructables.com/id/Fast-Electric-Skateboard-LiPo-Charging-System-BMS-/?ALLSTEPS>

Pully protection :

<http://www.instructables.com/id/3D-Printing-Electric-Skateboard/>

Instructable :

<http://www.instructables.com/id/Make-Your-Own-Electric-Motorized-Longboard/?ALLSTEPS>

LeafBoard :

<https://www.youtube.com/watch?v=oR5q7k0IM_4>