

Un video time inserter à base d'un module MinimOSD

DIY video time inserter based on a MinimOSD module - Pascal ANDRE 27

Nov 2017

Documentation V1.1 Update 17 July 2018 (French/English)

Contact: *pascal (dot)andre9 (at) free (dot) fr*

www.qfastro.club Toulouse - France

Ce module est destiné à l'origine à la création d'un tableau de bord de drone avec incrustation des commandes sur la caméra de vol, ce module combine en fait un arduino et un osd sur quelques cm²!.

This module is originally intended for the creation of a drone dashboard with overlay commands on the flight camera, this module combines an arduino chip and an OSD (On Screen Display) on a few cm² !.

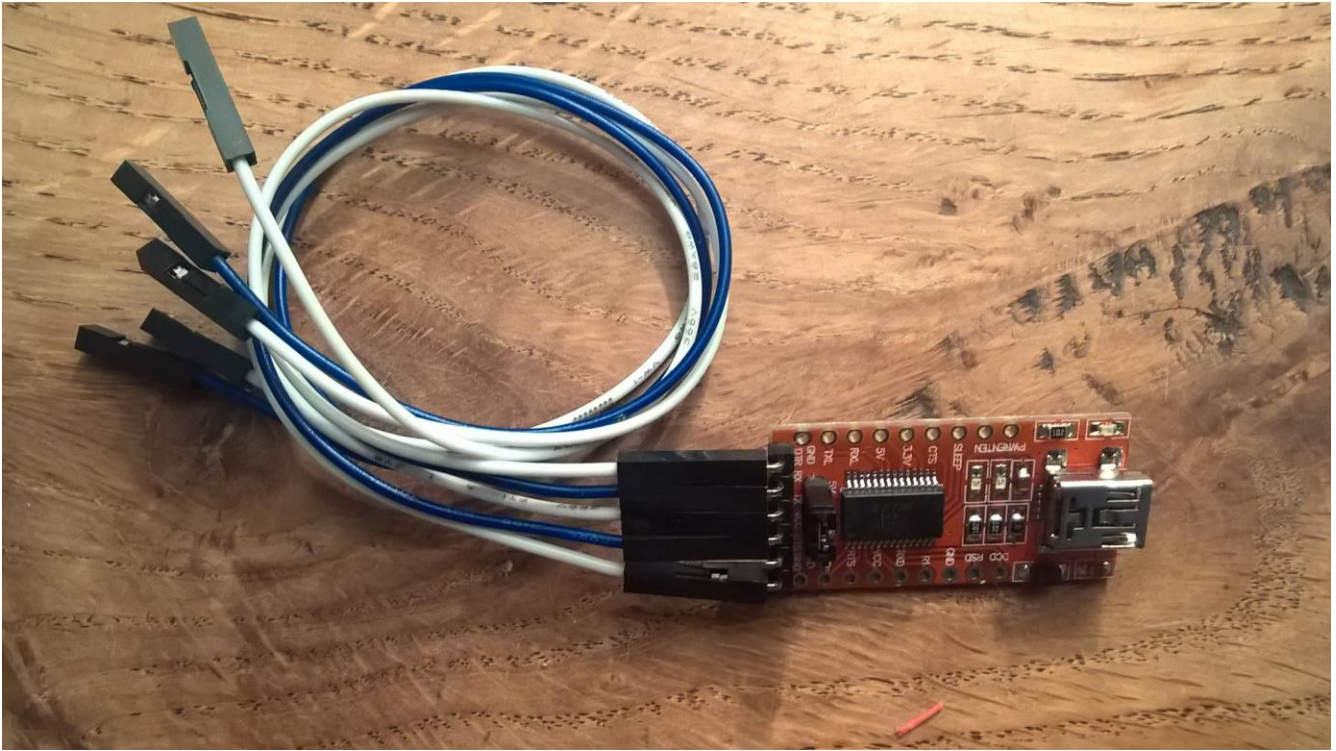
Le code est adapté d'une réalisation d'un amateur polonais Piotr Smolarz, smopi.pl@gmail.com <https://github.com/smopihub/smopiVTI> à partir d'un arduino uno et d'un shield osd.

The code is adapted from an achievement of a Polish amateur Piotr Smolarz, smopi.pl (at) gmail.com <https://github.com/smopihub/smopiVTI> from an arduino uno and a shield osd shield. (thank you again Piotr for this great work !)



Etape 1 : Flasher la carte avec le programme arduino de Video Time Inserter : utiliser pour cela un adaptateur FTDI, car la carte minim osd ne dispose pas de port usb contrairement à un arduino uno

Step 1: Flash the card with the Arduino program of Video Time Inserter: use for it an FTDI adapter, because the minimosd does not include USB port (unlike an arduino uno)



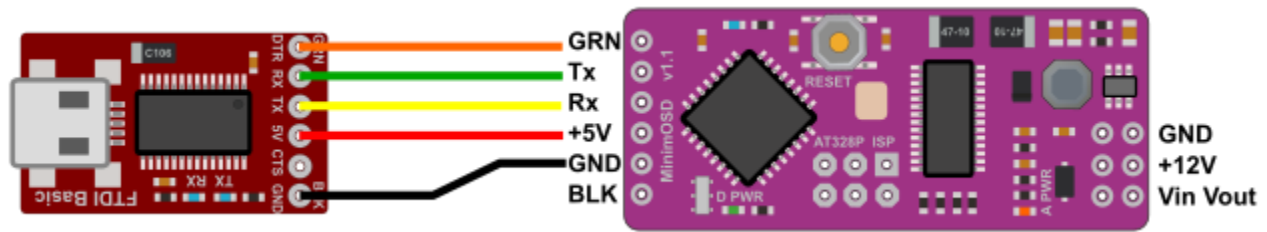
FTDI module

Le module FTDI se branche selon le schéma ci-dessous ; il suffit de flasher le code arduino via l'interface habituelle arduino IDE (port com de la carte, choisir arduino nano, téléverser) - cf annexe pour modifications à apporter au programme polonais VTI.ino.

NB : Le module FTDI ne sert plus à rien une fois, le code téléversé (NB : il faut au préalable charger un driver spécifique pour le module FTDI, tester le transfert avec Blink qui doit faire clignoter la diode sur le port 13 de la carte minim OSD)

The FTDI module is connected according to the diagram below; just flash the arduino code via the usual arduino IDE interface (select the correct com port of the card, choose arduino nano, upload) - see appendix for changes to the Piotr's program VTI.ino.

Note : The FTDI module is no longer used once, the code is uploaded (you should first install a specific driver for the FTDI module, test the transfer with generic "Blink" Arduino code : LED on port 13 of the card minim OSD should blink)



Les seules modifications à faire sur le code arduino sont de changer la valeur du chip select (CS) pour la mettre à 6 et de forcer dans la partie config l'utilisation des caractères ASCII (cf Schémas et extrait du code à modifier en annexe)

The only modifications to the arduino code are to change the value of the chip select (CS) to 6 and to force in the config part the use of ASCII characters (see Diagrams and excerpt of the code to be modified in appendix)

Etape 2 préparation du module GPS pour tirer un fil avec le signal one pps (fil bleu soudé sur la patte 3 du module Neo-6M), gratter le circuit au cutter, étamer pour souder le fil (ici j'ai utilisé du fil fin à wrapper monobrin). **Cette étape est inutile si le module est déjà doté d'une sortie 1pps**

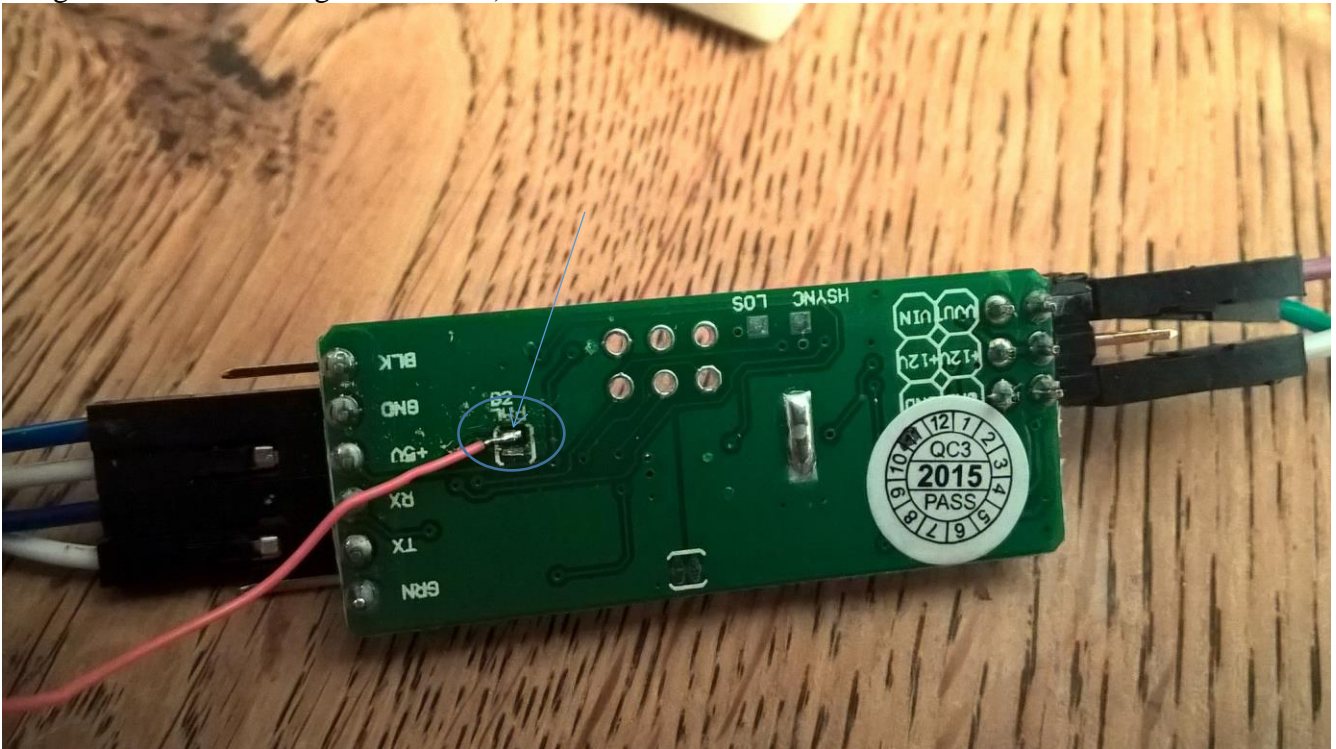
Step 2 Prepare the GPS module to sold a wire with the signal one pps (blue wire soldered on the leg 3 of the Neo-6M module), scrape the circuit with a cutter, tin to solder the wire (here I used thin wrapping wire). This step is useless if the module has a 1pps pin



Etape 3 : récupérer la borne PD2 INT1 de l'ATMega328P monté sur le minim OSD

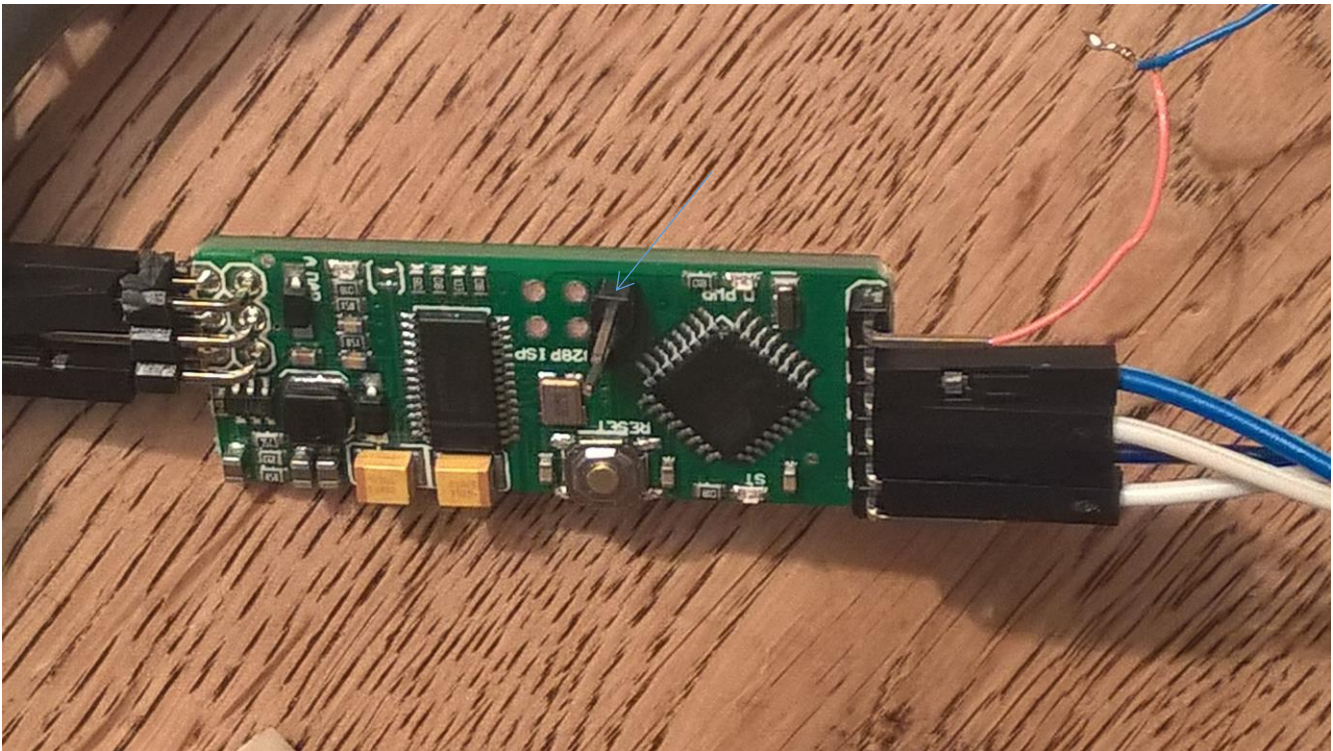
Il faut modifier la carte minim OSD en soudant un fil sur le “solder jumper PAL” afin de (Fil rouge) sans établir de connexion au +5V présent sur l’autre côté du solder jumper (protéger le côté ne devant pas être soudé par un bout d’adhésif pour éviter de créer un pont de soudure par capillarité, car ceci est justement conçu pour créer à la demande un pont au moyen d’une soudure)

Step 3: Create ATM2a328P -PD2 INT1 terminal on the minimosd. The minimOSD card should be modified by soldering a wire (red wire on picture) to the "PAL jumper" in order to avoid making a connection to the +5V present on the other side of the solder jumper (protect the side not to be soldered by a piece of adhesive to avoid creating a bridge solder by capillarity, because this is precisely designed to create a bridge on demand)



Etape 4 sortir les connexions d’un reset externe (bouton poussoir), pour cela souder 2 cosses sur le port SPI pour récupérer « RESET » et « GND » : un poussoir sera connecté sur ces cosses :

Step 4 establish the connections for an external reset (push button), in order to do that, solder 2 pins on the SPI port to connect "RESET" and "GND": a push button will be connected to these 2 pins:



Etape 5 Connecter les modules GPS et MinimOsD ; alimenter ces 2 modules

Relier le fil bleu "one pps" sur le fil rouge "INT0" pour trigger le signal one pps

Etablir une communication entre les modules minim osd au gps **en croisant** : TX avec RX et RX avec TX

Attention : la carte minimosd comporte deux parties qui doivent être alimentées en 5 V (2 led allumées) si ce n'est pas le cas (1 seule led allumée) vous devez souder les jumpers situés au milieu du circuit imprimé (1 sur chaque face)

Alimenter GPS et Minim OSD avec +5Vcc et GND et Voilà un vidéo insérer pour moins de 15euros !

Step 5 Connect the GPS and MinimOsD modules;

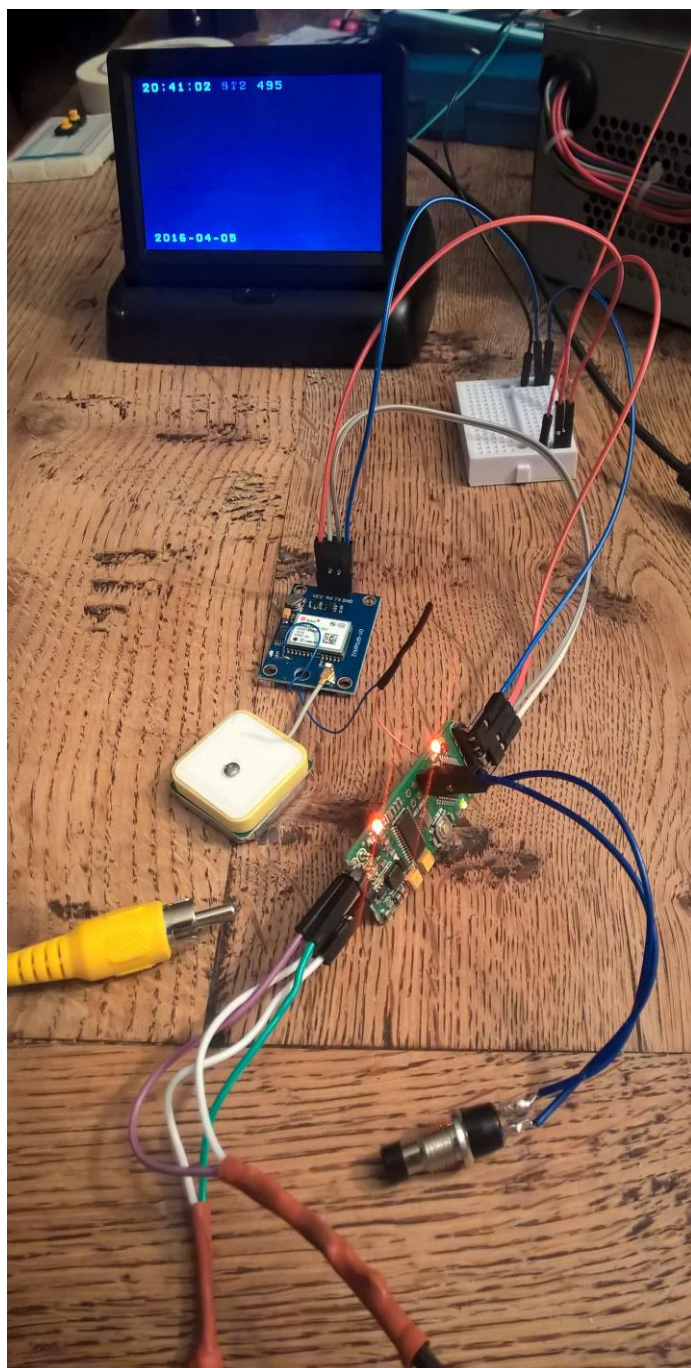
-power these 2 modules,

-connect the blue wire "one pps" to the red wire "INT1" in order to trigger the signal one pps signal. --

-establish a communication between the minim osd to the GPS by crossing: TX with RX and RX with TX

Caution : OSD part and ATM part of the Minimosd could be powered independently : you should check that both parts of the card are correctly powered (2 red leds on): (if it is not the case: only one led for example, you should solder the jumpers located in the middle on both sides of the PCB)

Power GPS and Minim OSD with + 5Vcc and GND and here is a video time insérer for less than 15euros!

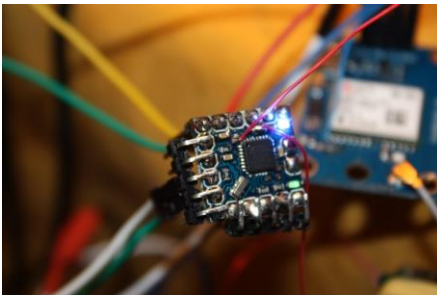


Améliorations possibles

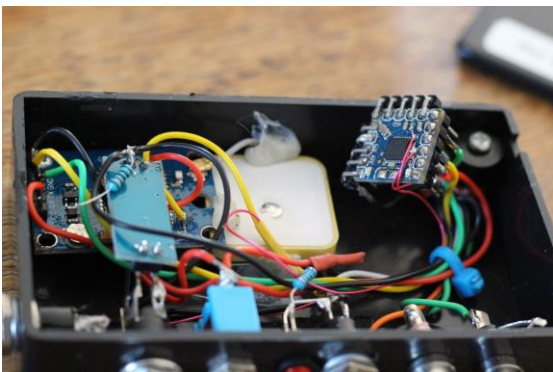
- Utiliser la version 1.2 de la carte minim osd en exploitant son « expansion header » pour gérer des E/S supplémentaires par exemple pour afficher les informations de localisation, T°, Etat du ciel ... >>> `digitalWrite(Ax , HIGH)` ou `digitalRead(Ax)` l'accès Ax sera automatiquement considéré comme digital,
- Intégrer un petit module de régulation 5V pour entrer du 6- 12V
- Customiser le code VTI.ino
- Si vous aimez la microchirurgie, vous pouvez tenter une réalisation avec une microosd

Possible improvements

- use version 1.2 of the minimosd with "expansion header" to manage additional I/O for example to display location information, T°, Sky status ... >>> use `digitalWrite (Ax, HIGH)` or `digitalRead (Ax)` instruction Ax access will automatically be considered digital I/O (and not analog I/O) ,
- integrate a small 5V regulator module to supply from 5 to 12V
- customize the code VTI.ino
- if you have some skills in micro-surgery you can adapt a microSD card to realize the smallest vti in the world:



Synchronized !

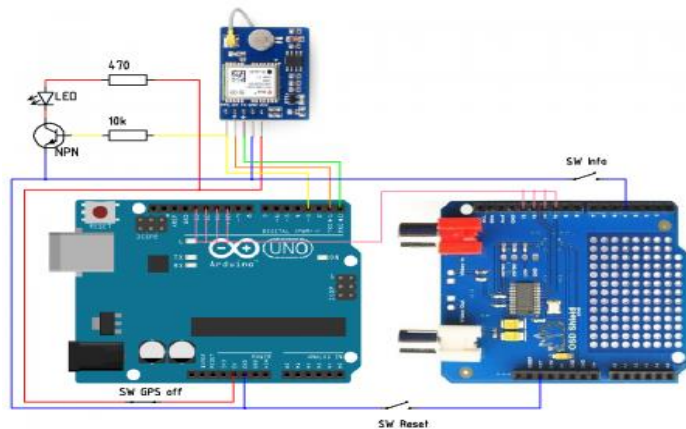




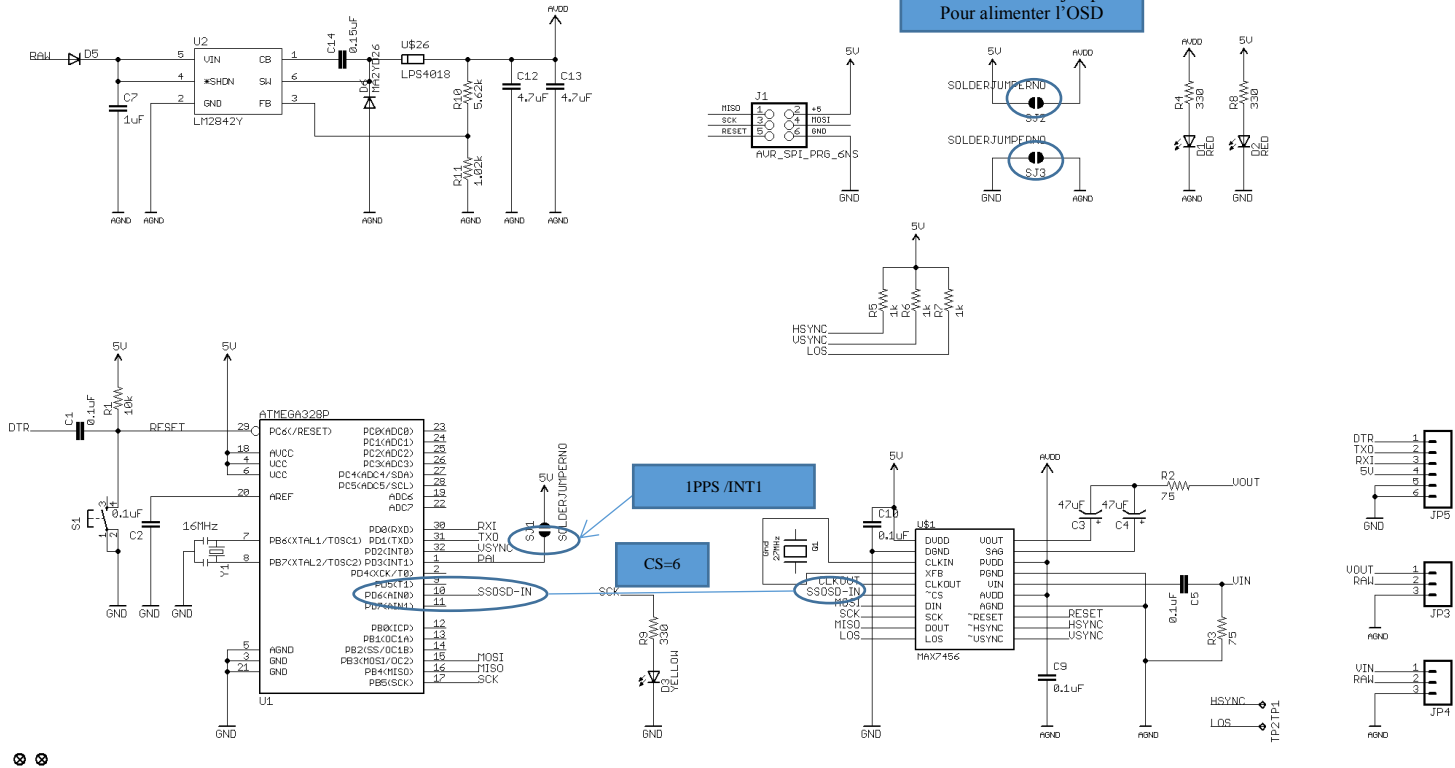
Appendix

1- Source Code

<http://smopi.news.nstrefa.pl/index.php?pages/Video-Time-Inserrer>



2- minim OSD v0.1 schematics



3-Modification to the original code VTI.ino : Changes in red

/*

Wymagania sprzętowe:
 Arduino UNO,
 odbiornik GPS U-Blox NEO-6M lub zgodny,
 VideoOverlayShield MAX7456

(http://www.tinyosshop.com/index.php?route=product/product&product_id=834).

Moduł główny jest odpowiedzialny za odczytanie z odbiornika GPS aktualnego położenia oraz godziny. Dane te są następnie umieszczane w obrazie video PAL lub NTSC.

*/

```
#define footerString "2016 MinimOSD"
```

```
#include <Time.h> //Time (http://www.pjrc.com/teensy/td\_libs\_Time.html) by Michael Margolis
```

```
#include <TinyGPS++.h> //TinyGPS++ (http://arduiniana.org/libraries/tinygpsplus/) by Mikal Hart
```

```
#include <SPI.h>
```

```
#include <MAX7456.h>
```

```

//Display mode pin: Date/Time <-> Info
const int displayModePin = 4;

//GPS RXPin = 0, TXPin = 1
const uint32_t GPSBaud = 9600; //GPS boundrate
const int PPSpin = 3; //PPS

// The TinyGPS++ object
TinyGPSPlus GPS;

//Time
volatile unsigned long msTimeStamp = 0;
volatile time_t sysDateTime = 0;

//OSD
const byte osdChipSelect = 6; // and not 10
const byte masterOutSlaveIn = MOSI;
const byte masterInSlaveOut = MISO;
const byte slaveClock = SCK;
const byte osdReset = 0;

//OSD object
MAX7456 OSD( osdChipSelect );

void setup()
{
  unsigned char system_video_in=NULL;
  TimeElements tm;

  pinMode(PPSpin, INPUT); //PPS
  pinMode(displayModePin, INPUT_PULLUP); //Display mode

  Serial.begin(GPSBaud); //GPS Serial

  // Initialize the SPI connection:
  SPI.begin();
  SPI.setClockDivider( SPI_CLOCK_DIV2 ); // Must be less than 10MHz.

  // Initialize the MAX7456 OSD:
  OSD.begin(); // Use NTSC with default area.
  OSD.setSwitchingTime( 5 ); // Set video chroma distortion
  // to a minimum.
  OSD.setCharEncoding( MAX7456_ASCII );// This line of code should be activated

  system_video_in=OSD.videoSystem();

```


//.... That's all

3 – Etude Adaptation MinimOsd V1.2 22/04/2016

Reprendre les mêmes modifications du Harware (Reset, Int1 sur Pal solderjump)

Souder les solder jumper pour alimenter la partie OSD (on doit avoir deux voyants rouge sur la carte sous tension) cf § annexe4 ces solder jumper sont situés de chaque côté de la carte

Utiliser RSSI (Arduino A3) en Entrée digital pour switcher sur « Info GPS » : mettre un inter plutôt qu'un poussoir

Exemple de code pour gérer l'appui du bonton relié sur RSSI

Study Adaptation MinimOsd V1.2 22/04/2016

Resume the same modifications of the Harware (Reset, Int1 on Pal solderjump)

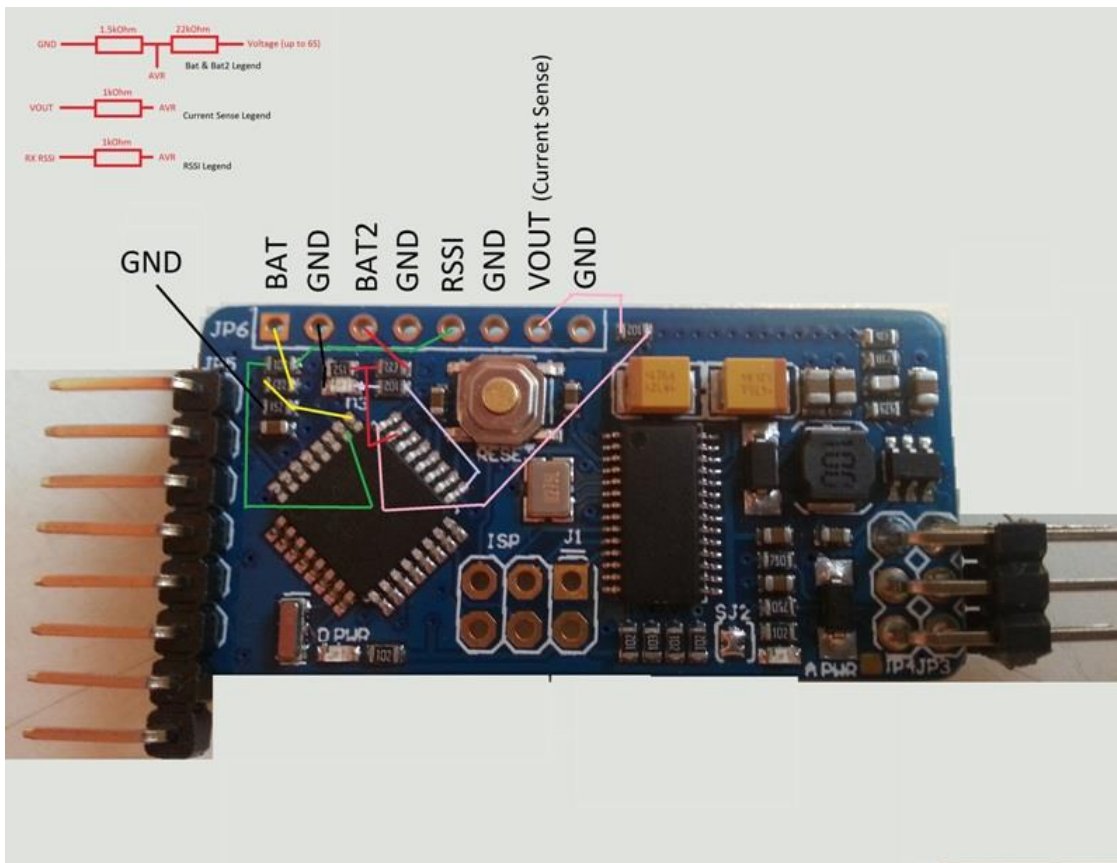
Weld the solder jumper to power the OSD (must have two red lights on the card power)

these jumpers are located on each side of the card

Use RSSI (Arduino A3) in Digital Input to switch to "GPS Info": put a switch rather than a push button

Sample code to manage the support of the connected switch on RSSI

```
const int displayModePin = A3;
void setup() {
  // put your setup code here, to run once:
  pinMode (displayModePin, INPUT_PULLUP);
}
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite (displayModePin);
}
```



Minim OSD V1.2 header	Correspondance AT1_Mega 328	
BAT1	A2	
BAT2	A0	
TXRSS /RSSI	A3	Port retenu pour display info
VIOUT	A1	

CodeVTL.ino : modifications surlignées

/*

*/

#define footerString "2016 MinimOSD 1.2 PA"

#include <Time.h> //Time (http://www.pjrc.com/teensy/td_libs_Time.html) by Michael Margolis
#include <TinyGPS++.h> //TinyGPS++ (http://arduiniiana.org/libraries/tinygpsplus/) by Mikal Hart
#include <SPI.h>
#include <MAX7456.h>

```

//Display mode pin: Date/Time <-> Info
const int displayModePin = A3;

//GPS RXPin = 0, TXPin = 1
const uint32_t GPSBaud = 9600; //GPS baudrate
const int PPSpin = 3; //PPS

// The TinyGPS++ object
TinyGPSPlus GPS;

//Time
volatile unsigned long msTimeStamp = 0;
volatile time_t sysDateTime = 0;

//OSD
const byte osdChipSelect = 6; // et non 10
const byte masterOutSlaveIn = MOSI;
const byte masterInSlaveOut = MISO;
const byte slaveClock = SCK;
const byte osdReset = 0;

//OSD object
MAX7456 OSD( osdChipSelect );

void setup()
{
  unsigned char system_video_in=NULL;
  TimeElements tm;

  pinMode(PPSPin, INPUT); //PPS
  pinMode(displayModePin, INPUT_PULLUP); //Display mode
  Serial.begin(GPSBaud); //GPS Serial
  // Initialize the SPI connection:
  SPI.begin();
  SPI.setClockDivider( SPI_CLOCK_DIV2 ); // Must be less than 10MHz.

  // Initialize the MAX7456 OSD:
  OSD.begin(); // Use NTSC with default area.
  OSD.setSwitchingTime( 5 ); // Set video chroma distortion
  // to a minimum.
  OSD.setCharEncoding( MAX7456_ASCII ); // Activer cette ligne

  system_video_in=OSD.videoSystem();

```

