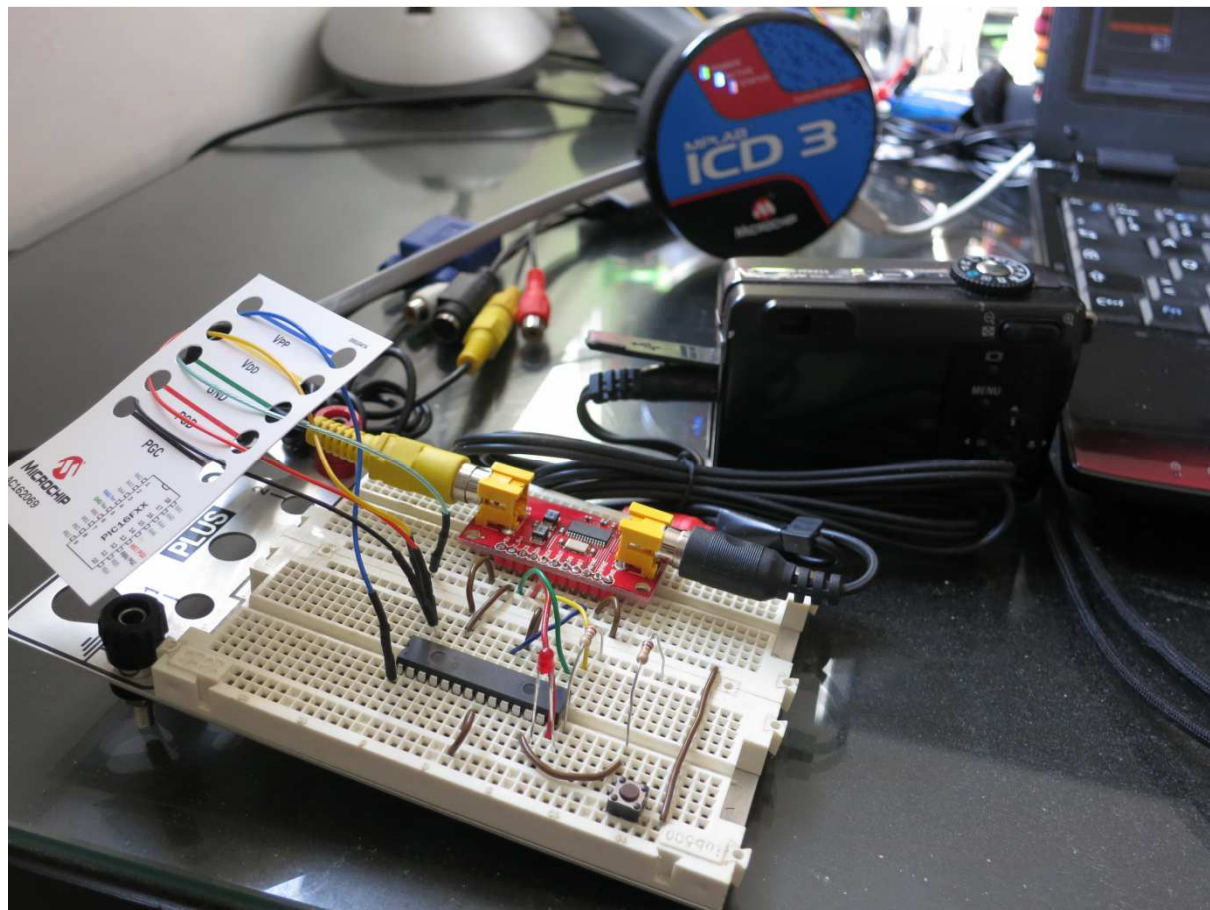


Un exemple de mise en œuvre de la bibliothèque avec un PIC18F2620 est disponible dans le dossier « TEST_MAX7456_P18F2620 »

Datasheet : <http://datasheets.maxim-ic.com/en/ds/MAX7456.pdf>

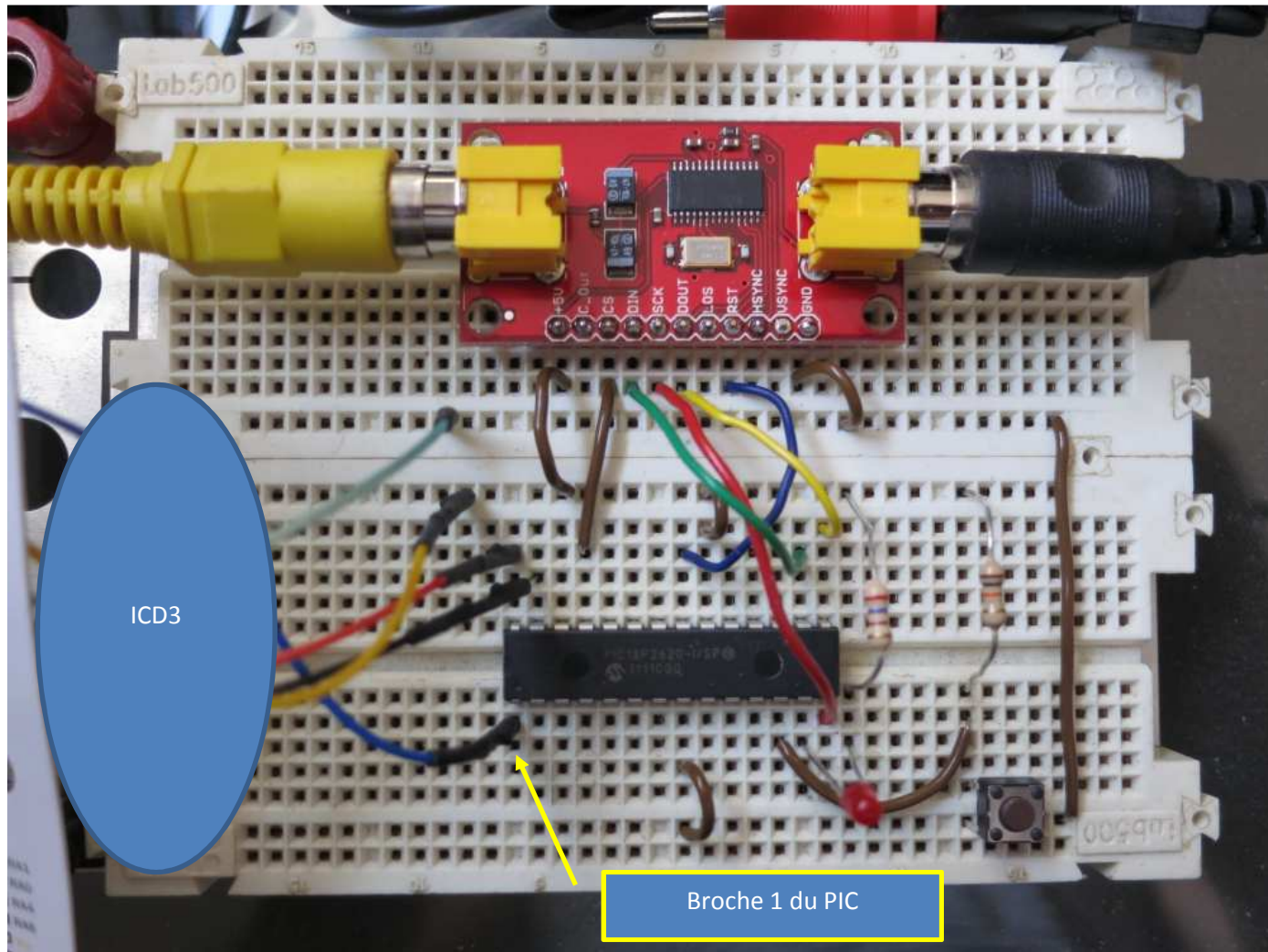
Mise en œuvre du module LEXTRONIC : <http://www.lextronic.fr/P5712-platine-dincrustation-video.html>



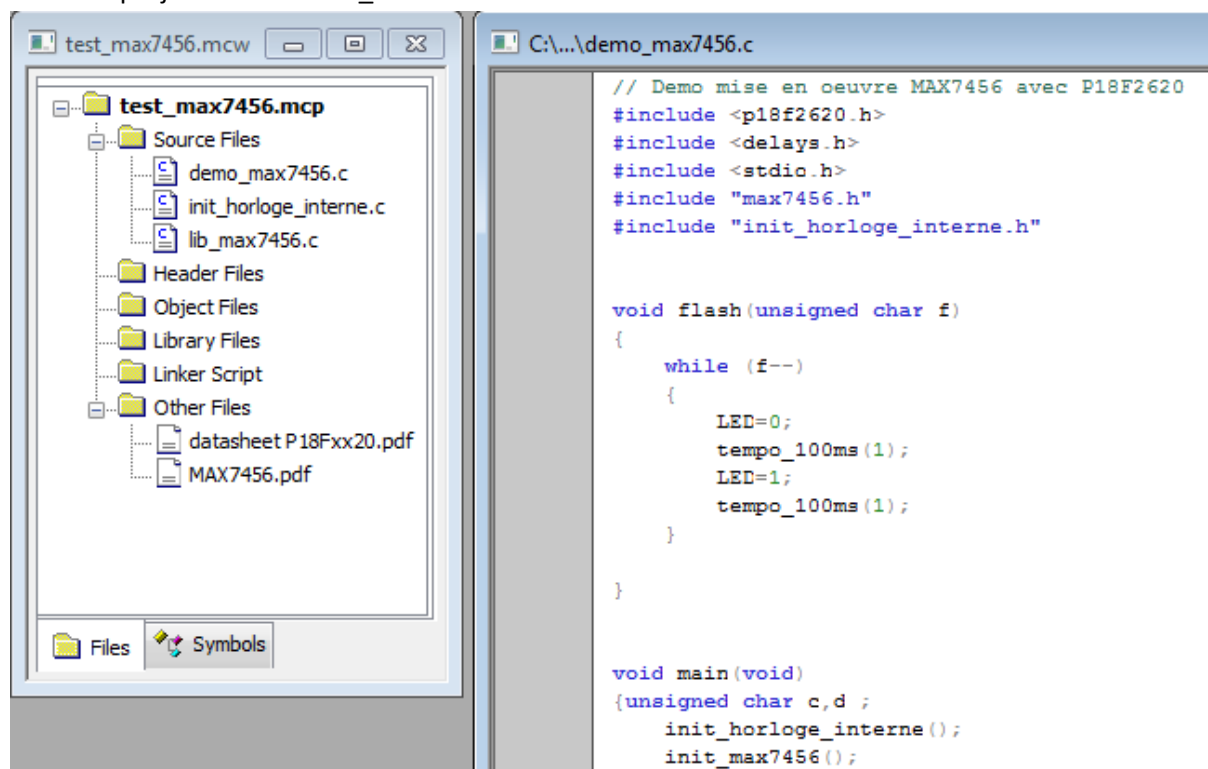
La source vidéo composite est un petit appareil photo SONY (DSCW17), l'image est visualisée sur un ordinateur disposant d'une carte d'acquisition vidéo ou d'un simple téléviseur ou moniteur disposant d'une entrée video-composite.

Le PIC18F2620 est connecté au module par le BUS SPI (CLK, DIN, DOUT) (voir datasheet PIC18F2620). Un port parallèle est utilisé pour le RESET soft du MAX7456. Le debugger est ici un ICD3.

Une LED et un bouton ont été ajoutés pour les tests (facultatifs)



Ouvrir le projet MPLAB « test_max7456 »



Fenêtre projet :

demo_max7456.c : exemple d'utilisation de la bibliothèque

init_horloge_interne.c : contient les #pragma de configuration et la fonction init_horloge_interne.c qui active l'horloge interne du PIC18Fxxxx à FOSC=32MHz.

lib_max7456.c : bibliothèque de fonction de contrôle du MAX7456

Programme demo_max7456.c

P18F2620.h : declaration des registre et bits du PIC

delays.h : temporisations basées sur TCY pour la génération des temps

stdio.h : le programme utilise le détournement des sorties caractères par « _user_putc ».

init_horloge_interne.h : déclaration de la fonction d'initialisation de l'horloge interne.

max7456.h : déclaration des registres et bits du MAX7456. Configuration matérielle entre le PIC et le MAX7456 (CLK, DIN, DOUT, RESET etc...) à modifier suivant le câblage retenu.

```
// defines I/O
// a adapter en fonction de la configuration materielle

#define LED PORTCbits.RC2      // au cas ou il y ait une LED
#define BOUTON PORTCbits.RC1  // au cas ou il y ait un bouton
#define CS PORTBbits.RB4      // CS du MAX7456
#define DIN PORTCbits.RC4     // DIN du max 7456 (SPI)
#define SCLK PORTCbits.RC3    // SCLK du max 7456 (SPI)
#define DOUT PORTCbits.RC5    // DOUT du max 7456 (SPI)
#define RST PORTBbits.RB0     // RESET du max 7456
#define TRIS_CS TRISBbits.TRISB4 // TRIS associes
#define TRIS_DIN TRISCbits.TRISC4
#define TRIS_SCLK TRISCbits.TRISC3
#define TRIS_DOUT TRISCbits.TRISC5
#define TRIS_RST TRISBbits.TRISB0
#define TRIS_LED TRISCbits.TRISC2
#define TRIS_BOUTON TRISCbits.TRISC1
```


Fonctions de la bibliothèque :

void tempo_tCSW(void);

temporisation entre deux émissions SPI (1uS)

void tempo_100ms(unsigned int t);

temporisation après RESET=0, doit être ajusté suivant FOSC (FOSC = 32 MHz dans l'exemple)

unsigned char emet_spi(unsigned char s);

emet un octet sur le bus SPI

unsigned char lit_max7456(unsigned char ad);

lecture d'un octet dans le registre « ad » du MAX7456

void emet_max7456(unsigned char ad, unsigned char data);

écriture de « data » dans l'adresse « ad » du MAX7456

void inverse(unsigned char c);

si c=1 passage en caractère noir sur fond blanc, si c=0 passage en caractère blanc

void clignote(unsigned char c);

si c=1 passage en caractère clignotant, si c=0 passage en caractère fixe

void gotoxy(unsigned char x, unsigned char y);

positionne le curseur d'écriture en x,y

void putc_max7456(unsigned char c);

emet un octet « c »

void init_max7456(void);

initialise le driver, emet un RESET et configure les communications SPI.

Affichage du programme de test :