

DUP (Digilite_Ultram_Pilot) by F1CJN
alain.fort.f1cjn@sfr.fr

The DUP is designed to drive the ULTRAM synthesizer used by the Digilite DA-TV transmitter described in the <http://www.g8ajm.tv/dlindex.html> website and provided by the BATC.

The ULTRAM synthesizer is able to cover the mondial ATV 1220 to 1300 MHz band, but unfortunately, it is programmed to a single frequency of 1249 MHz.

By removing the small 8 pins PIC12F629 (at your own risk) from the ULTRAM board and driving the SPI bus with a PIC 16F628, it is now possible to program, using two push buttons, any frequency in the 1220 to 1300MHz range with a 1MHz step.

The frequency is displayed on a 2 line HD44780 LCD .

The displayed frequency can be memorized in the PIC EEPROM by a third push button, then the display shows the word “MEMORIZING”.

The hardware

- a PIC 16F628 from Microchip,
 - a standard LCD display HD44780 type, 2 x 16 characters lines display,
 - 10 resistors, an adjustable resistor, 3 capacitors and 3 push buttons, a 5V regulator and a diode,
- and an ULTRAM synthesizer card with a removed PIC16F629

The software

The software was developed with the free Mikroelektronika Picbasicpro demo version. An EasyPIC4 development board was used to burn the PIC and then the PIC was plugged in the breadboard (see image).

The program source is free and open source.

The source is commented so it is quite easy to understand how it works and to modify it.

The .hex files

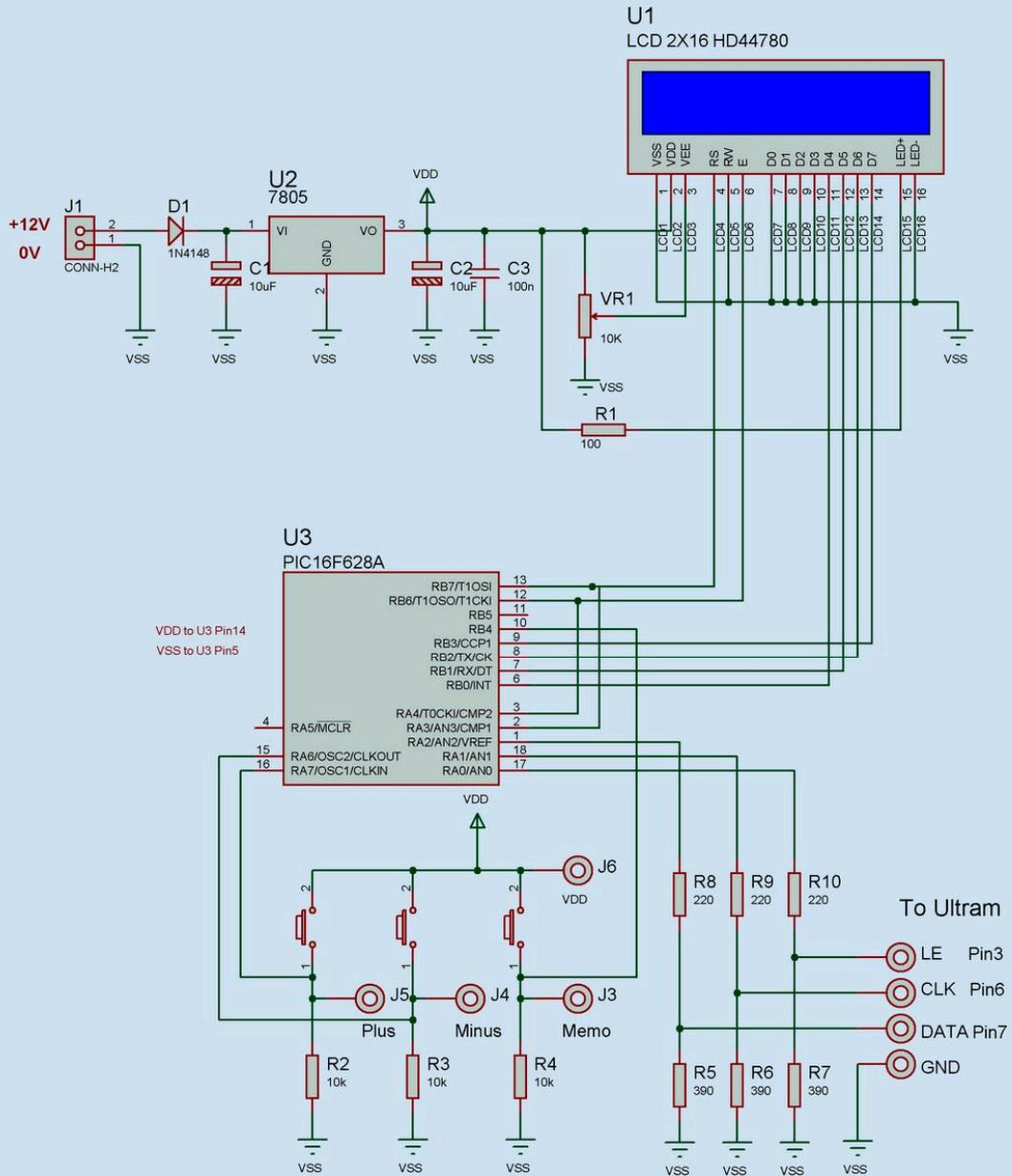
They are used to burn the PIC E2PROM. Two files are provided, one is you use a PIC16F628 and one for a PIC16F628A

Programming the PIC : VERY IMPORTANT

This is the PIC programming configuration (or the project configuration with Picbasic Pro)

- 1) Internal RC oscillator to enable RA6 and RA7 as I/Os
- 2) RA5/MCLR pin fonction select : disable
- 3) Low voltage programming :disable (to enable RB4)

Schematic



Pin 13 is connected to Pin2 to ease single layer PCB layout
 Pin 12 is connected to Pin3 to ease single layer PCB layout

FILE NAME:	DESIGN TITLE: Digilite_Ultram	DATE: 31/10/2011
PATH:	BY: F1CJN	PAGE: F1CJN
REV: 1.0	TIME: 23:02:21	

Bill of materials

QTY	PART-REFS	VALUE

Resistors		

1	R1	100
3	R2-R4	10k
3	R5-R7	390
3	R8-R10	220
Capacitors		

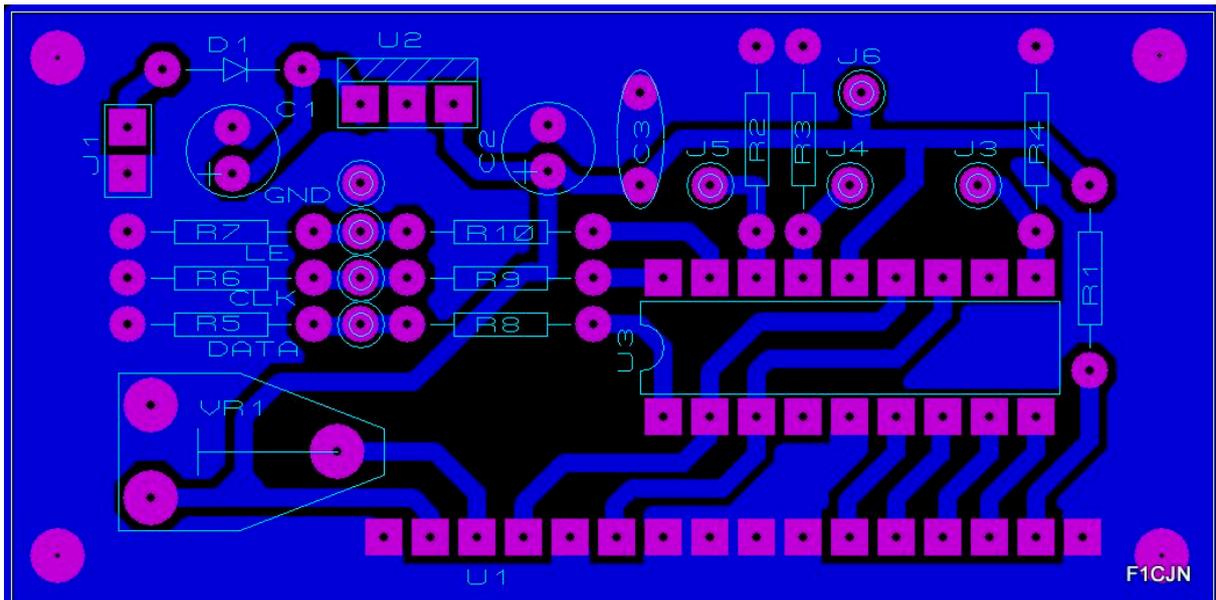
2	C1,C2	10uF 20V.
1	C3	100n
Integrated Circuits		

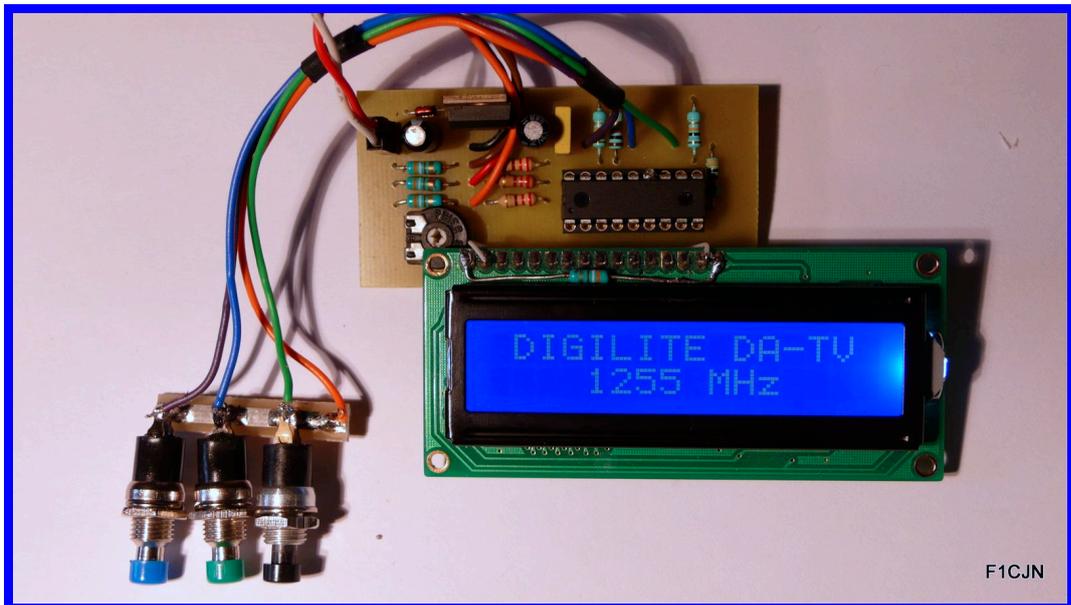
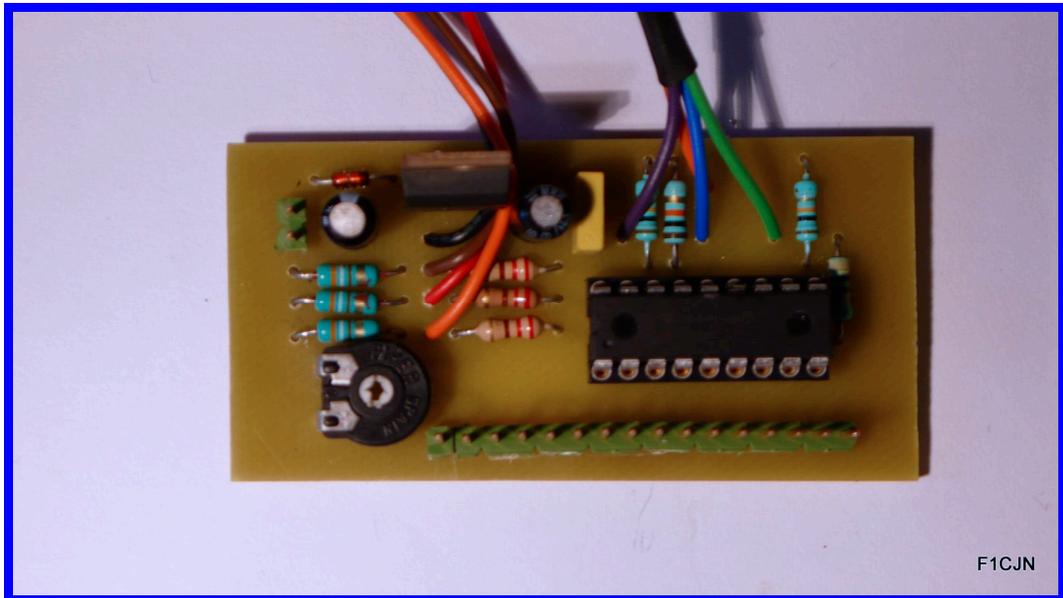
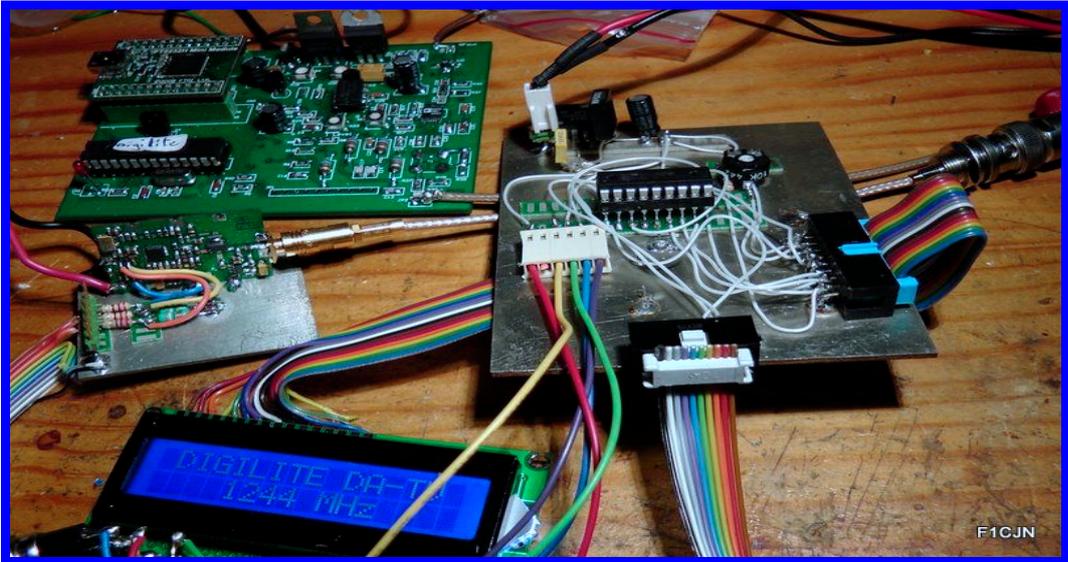
1	U1	PIC16F628
1	U2	7805
Diodes		

1	D1	1N4148
Miscellaneous		

1	LCD	DISPLAY LCD 2X16 HD44780
1	VR1	10K 1 turn
3	SW1-SW3	PUSH-BUTTON

Layout





Mikrobasic Pro source code

```
program Digilite_Ultram_628
' Written by F1CJN
' alain.fort.f1cjn@sfr.fr
' November,1 2011
' Developped with Mikroelectronica MikroBasic Pro demo version
' This program is free and open source.
' Design for PIC16F628 or PIC16F628A
' VCO ULTRAM from BATC, using ADF4360-5 PLL integrated circuit
' ULTRAM Xtal F reference at 10 MHz
' ADF4360-5 Phase comparator reference at 1MHz = 10 MHz/10 ; R=10
' ***** Exemple *****
' Output Frequency = 1249 MHz
'  $F = NP + A = (156 \times 8) + 1 = 1249$ 
' Programming the ADF4360-5 with : N= 156, P=8, A= 1, R=10
' *****
' ***** Hardware *****
' PIC removed (at your own risk) from the Ultram PLL board
' PIC 16F628 with internal 4 Mhz clock .
' PIC 16F628 RA0 connected to LE (pin 3 of PIC ULTRAM)through resistive divider
' PIC 16F628 RA1 connected to CLK (pin 6 of PIC ULTRAM)through resistive divider
' PIC 16F628 RA2 connected to DATA(pin 7 of PIC ULTRAM)through resistive divider
' Resistive divider are 220 Ohms and 390 Ohms to gnd)
' 10K pull down resistors at RB4(Memo), RA7(Plus)and RA8(Minus),
' Push buttons between VDD and RB4, RA7 and RA3
' *****
' *****Very important when programming the PIC *****
' This is the PIC programming configuration (or the project configuration)
' 1)Internal oscillator with RA6 and RA7 as I/Os
' 2) RA5/MCLR pin fonction select : disable
' 3) Brown-out reset : enabled
' *****
' Section declarations
dim Chip_Select as sbit at RA0_bit 'O/P Chip Select toward pin 3 (Ultram)
dim SoftSpi_CLK as sbit at RA1_bit 'O/P CLK toward pin 6 (Ultram)
dim SoftSpi_SDO as sbit at RA2_bit 'O/P DATA toward pin 7 (Ultram)
dim SoftSpi_SDI as sbit at RA5_bit 'Not used do not connect RA5 !

' SPI declarations
dim Chip_Select_Direction as sbit at TRISA0_bit 'LE
dim SoftSpi_CLK_Direction as sbit at TRISA1_bit 'CLK
dim SoftSpi_SDO_Direction as sbit at TRISA2_bit 'DATA'
dim SoftSpi_SDI_Direction as sbit at TRISA5_bit 'Do not connect RA5 !

' LCD 2x16 characters declarations
dim
LCD_RS as sbit at RB7_bit 'Pin 13
```

```

LCD_EN as sbit at RB6_bit   'Pin 12
LCD_D7 as sbit at RB3_bit   'Pin 9
LCD_D6 as sbit at RB2_bit   'Pin 8
LCD_D5 as sbit at RB1_bit   'Pin 7
LCD_D4 as sbit at RB0_bit   'Pin 6
dim
LCD_RS_Direction as sbit at TRISB7_bit
LCD_EN_Direction as sbit at TRISB6_bit
LCD_D7_Direction as sbit at TRISB3_bit
LCD_D6_Direction as sbit at TRISB2_bit
LCD_D5_Direction as sbit at TRISB1_bit
LCD_D4_Direction as sbit at TRISB0_bit

' Variables
dim Temp as longword
dim i as byte
dim oldstate as byte
dim N as byte
dim A as byte
dim Freqlow as byte
dim Freqhigh as byte
Dim EE_Flag as byte
dim Freq as word
dim oldfreq as word
dim Freq_Str as string[5]

sub procedure InitMain()
    delay_ms(300)           ' Delay for the LCD start duration
    Chip_Select_Direction = 0      ' Set CS# pin as Output
    Soft_Spi_Init()           ' Initialize SPI routine
    LCD_init()                ' Initialize LCD
end sub

sub procedure Send_SPI()
' ADF4360-5 R Register Programming with R=10 "0000001010"
Chip_Select=0
temp = %00000000
Soft_SPI_Write(temp)
temp = %00000000
Soft_SPI_Write(temp)
temp = %00101001   ' "001010" for R=10 and "01" for C2,C1
Soft_SPI_Write(temp)
delay_us(10)
Chip_Select=1 ' validation données
delay_ms(10)

' ADF4360-5 Register Control Programming
Chip_Select=0
temp = %00000100 'the 2 msb are the prescaler bits P2=0 P1=0 ~ 8/9
Soft_SPI_Write(temp) 'Send first octet

```

```

Temp = %11100001
Soft_SPI_Write(Temp) 'Send second octet
Temp = %00100100 ' with "00" for lsbs (C2,C1)
Soft_SPI_Write(Temp) 'Send third octet
Delay_us(10)
Chip_Select=1 ' validation données
delay_ms(10)

'ADF4360-5 N Counter Latch Programmation
N=Freq/8 ' Freq = (Nx8)+A
A=Freq-(8*N) '

Chip_Select=0
Temp = %00000000
Soft_SPI_Write(Temp) 'Send 1st octet BD23 to DB16

Temp=N 'prepare B counter DB15-DB8
Soft_SPI_Write(Temp) 'Send 2nd octet

Temp=A
Temp=Temp<<2 'Shift left 2 bits to put 5 bits A counter at the right place
Temp=Temp+ %10 'ADD binary 10 : for the Control bits C2=1 and C1=0
Soft_SPI_Write(Temp) 'Send 3rd octet
Delay_us(10)
Chip_Select=1 ' SPI data validation
end sub

main:
Initmain
PORTB= %00110000 '1 = I/P 0=O/P
TRISA= %11111000 '1 = I/P 0=O/P
TRISB= %00010000 '0=O/P
LCD_cmd(_LCD_CLEAR)
LCD_cmd(_LCD_CURSOR_OFF)
LCD_out(1,1," DIGILITE DA-TV ")

EE_flag= EEPROM_read($02) 'Read "Already writed flag" in EEPROM
If EE_Flag = $55 then ' EEPROM already writed of not ?
Freqlow =EEPROM_read($00) 'Read Freq low byte in EEPROM
Freqhigh=EEPROM_read($01) 'Read Freq high byte in EEPROM
Freq= Freqhigh*256 + FreqLow
else
Freq=1255 'Default starting frequency if the EEPROM is not yet writed
end if

Send_SPI 'Program the ADF4360-5 via the SPI bus

While true
Oldfreq=Freq
.....

```

```

' Bouton Frequency UP PortA.6 '
.....
if Button(PORTA,6,5,1) then 'Is button UP on ?
oldstate = 1

Freq=Freq+1
If freq >1300 then freq=1300 'High Frequency limit test
end if
for I=0 to 4
LCD_CHR(2,i+4,Freq_Str[i]) 'Writes the frequency
next i
LCD_out(2,9," MHz")

while PORTA.6=1 'waiting button release
wend
if (oldstate = 1) and (Button(PORTA,0,6,0)) then 'button 0 off ?
end if
end if
Oldstate=0
.....
' Bouton Frequency down PortA.7 '
.....
if Button(PORTA, 7, 5, 1) then 'Is button Down on ?
oldstate = 1

Freq=Freq-1
If freq <1220 then freq=1220 'Low frequency limit test
end if
for I=0 to 4
LCD_CHR(2,i+4,Freq_Str[i]) 'Writes the frequency
next i
LCD_out(2,9," MHz")

while PORTA.7=1 'waiting button release
wend
if (oldstate = 1) and (Button(PORTA, 7, 5, 0)) then 'button 1 off ?
end if
end if
Oldstate=0

.....
' Bouton Memorization PortB.4 '
.....
if Button(PORTB, 4, 5, 1) then 'Is button Memo on ?
oldstate = 1
while PORTB.4=1 'waiting button release
wend
end if
if (oldstate = 1) and (Button(PORTB, 4, 5, 0)) then 'button 1 off ?
Freqlow=Freq 'Low byte of word Freq

```

```

Freqhigh=freq>>8 'Shift right 8 bits to get the high byte of word Freq
EEPROM_write($00,Freqlow) 'write low byte frequency in EEPROM
EEPROM_write($01,Freqhigh) 'write high byte frequency in EEPROM
EEPROM_write($02,$55) 'write $55 EE_Flag= EEPROM writed
LCD_out(1,1," MEMORIZING ")
Delay_ms(1000)
end if
Oldstate=0

If freq <> oldfreq then 'If it is a new frequency then
Send_SPI 'Program the ADF4360-5 SPI
end if

LCD_out(1,1," DIGILITE DA-TV ")
Wordtostr (freq,Freq_str) 'Prepare the string frequency
for I=0 to 4
LCD_CHR(2,i+4,Freq_Str[i]) 'Display the frequency
next i
LCD_out(2,9," MHz")

wend
end.

```