

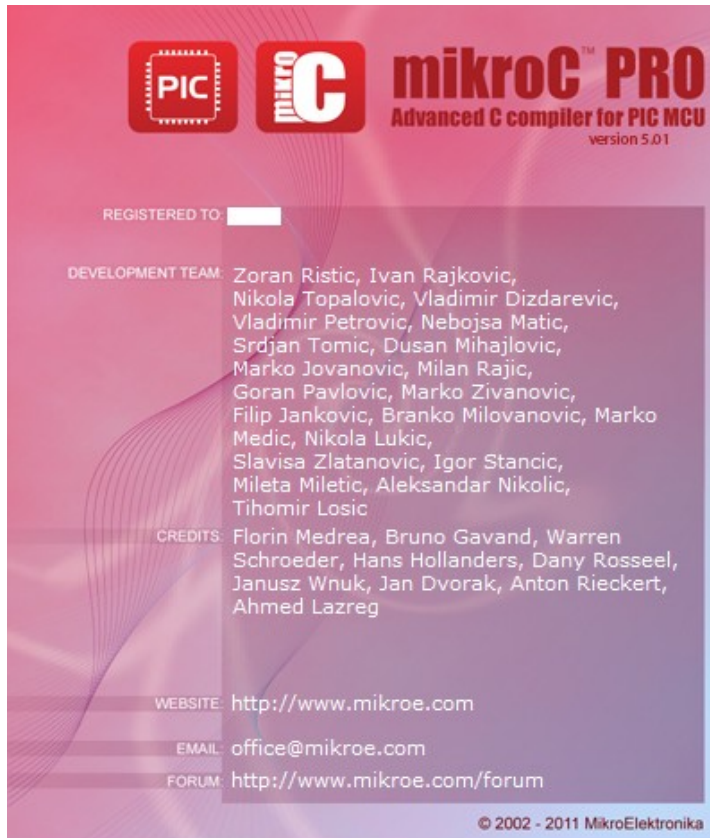
HƯỚNG DẪN GIAO TIẾP USB-HID VỚI PIC18F4550

Chào các bạn, hôm nay Chủ Nhật ngày 25/09/2011, lại là 1 ngày buồn chán ko có việc gì để làm cả, ngồi nhâm nhi chén trà đặc và bắn vài điều thuốc → cuộc đời sướng nhất là những lúc như thế này ☺ Ngoài trà và thuốc ra thì cũng tranh thủ lướt Keyboard tặng các bạn 1 TUT về giao tiếp USB HID Class của Microchip PIC18F4550.

- Như chúng ta đã biết, con PIC18F4550 này tương đối thông dụng, các Member trên các 4rum như : www.dientuvietnan.net , www.picvietnam.com www.bkit4u.com ... thường dùng. Các đề tài, đề án có liên quan đến USB thì PIC18F4550 sẽ được chọn. PIC18F4550 hỗ trợ USB2.0 Full Speed 480Mbps
- Nói về USB thì chúng ta thường quan tâm đến 2 lớp thường dùng, đó là : USB HID và lớp USB Mass Storage. Và ở đây cũng thường dùng 2 kiểu truyền đó là Bulk với USB Mass Storage và Interrupts với lớp HID. Thì cũng nói luôn là chơi với Bulk thì mới có thể đạt được Full Speed , còn HID thì chúng ta thường chỉ là 64KB/s hoặc 512KB/s. So với rs232 thì Max rs232 == 128000bps ~ 16KB/s), nhưng để đẩy đến tốc độ này ko phải thiết bị nào cũng hỗ trợ, PIC thì chắc chắn là ko ăn đến tốc độ này rồi.
- Tuy tốc độ HID không cao nhưng nó được ứng dụng cho các thiết đơn giản như Keyboard, Mouse, các thiết bị ko yêu cầu về Driver
- HID truyền được cả 2 chiều từ Device lên Host và ngược lại. Việc trao đổi dữ liệu giữa Host với Device được thực hiện qua giao thức usb, quản lý bằng các endpoint(điểm đầu cuối) và pipe (ống truyền).
- Đến đây cơ bản là chúng đã có thể sẵn sàng vào làm 1 thiết USB lớp HID với chức năng Plug and Play (PnP) hay nói cách khác là cắm là chạy, ko cần Driver.

1. Giới thiệu về phần mềm Compiler của PIC của MikroC

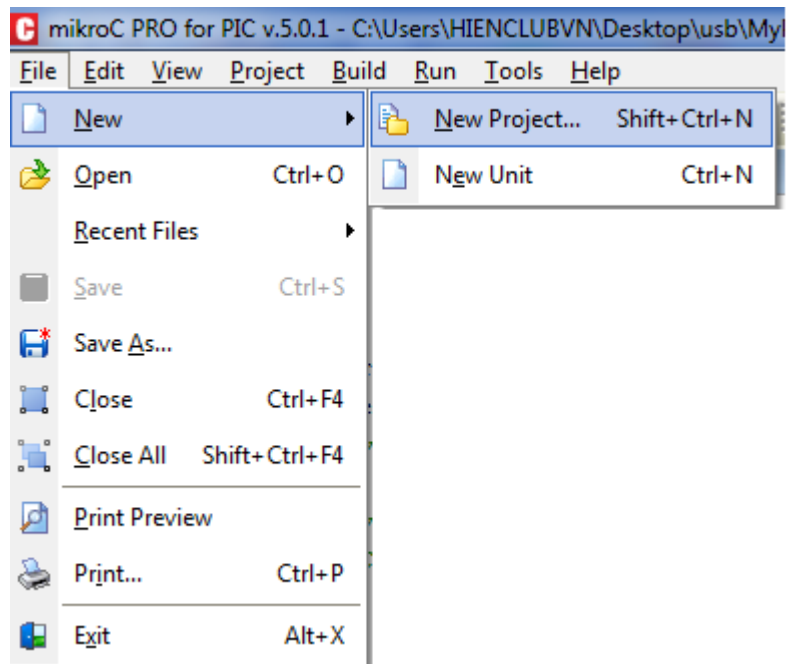
Chúng ta sử dụng phần mềm *MikroC Pro For PIC v5.01* bản mới nhất của năm 2011 được bán với giá là 249 USD, tính theo tỉ giá đô tại thời điểm này (25/09/2011) thì chúng ta cần : $249 \times 20.834 = 5187.666 \sim$ tính khoảng 5 củ 2 cho nó dễ.



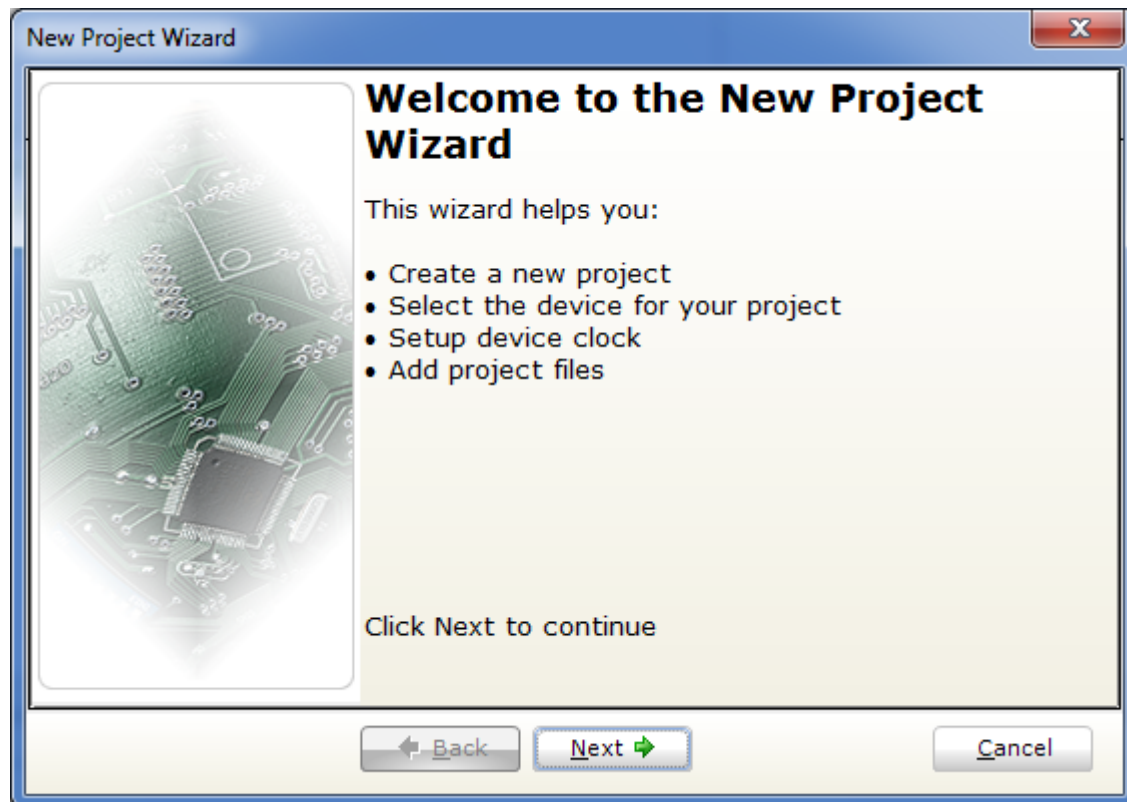
- Thôi thì sinh viên nghèo, ko có tiền chỉ còn cách là lên thẳng trang chủ để load phiên bản mới về và search google C.x.x.k để Registered nó.

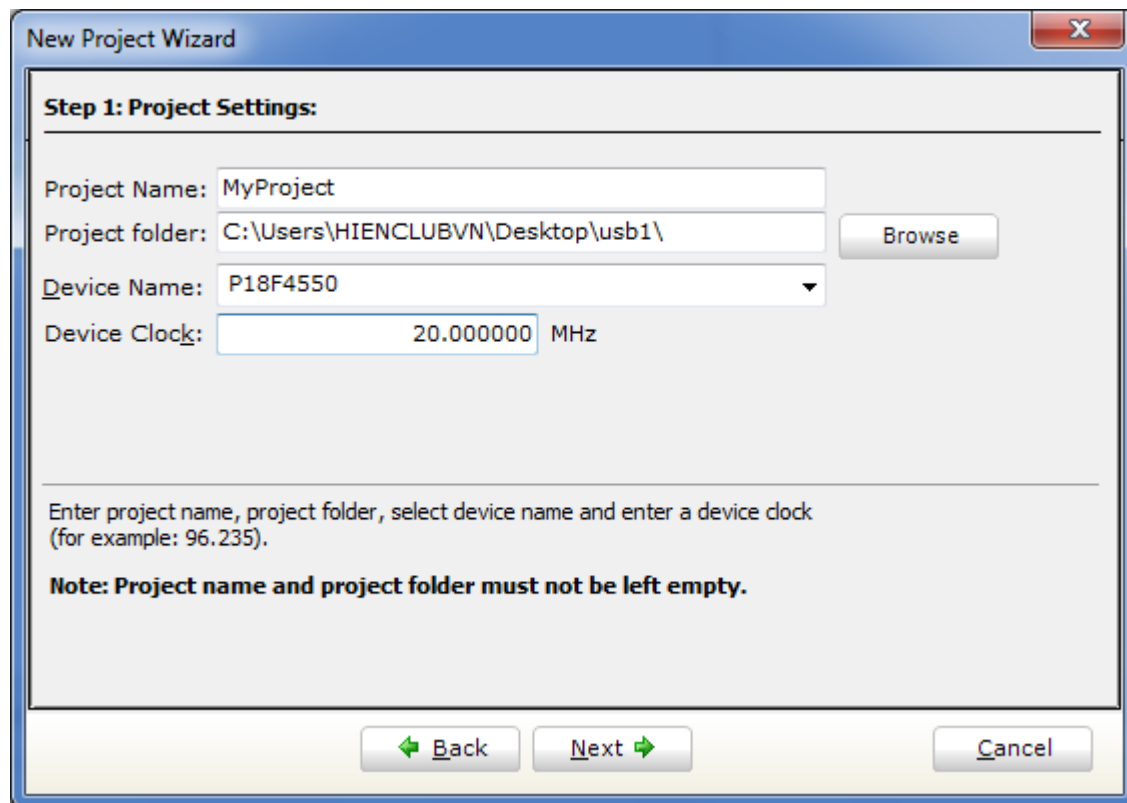
2. Code với MikroC

- Tạo 1 Project mới.
File / New / New Project ...



Và cứ *Next* mà thôi, cái này thì dễ quá rồi, ko phải nói nhiều.





New Project Wizard

Step 1: Project Settings:

Project Name:

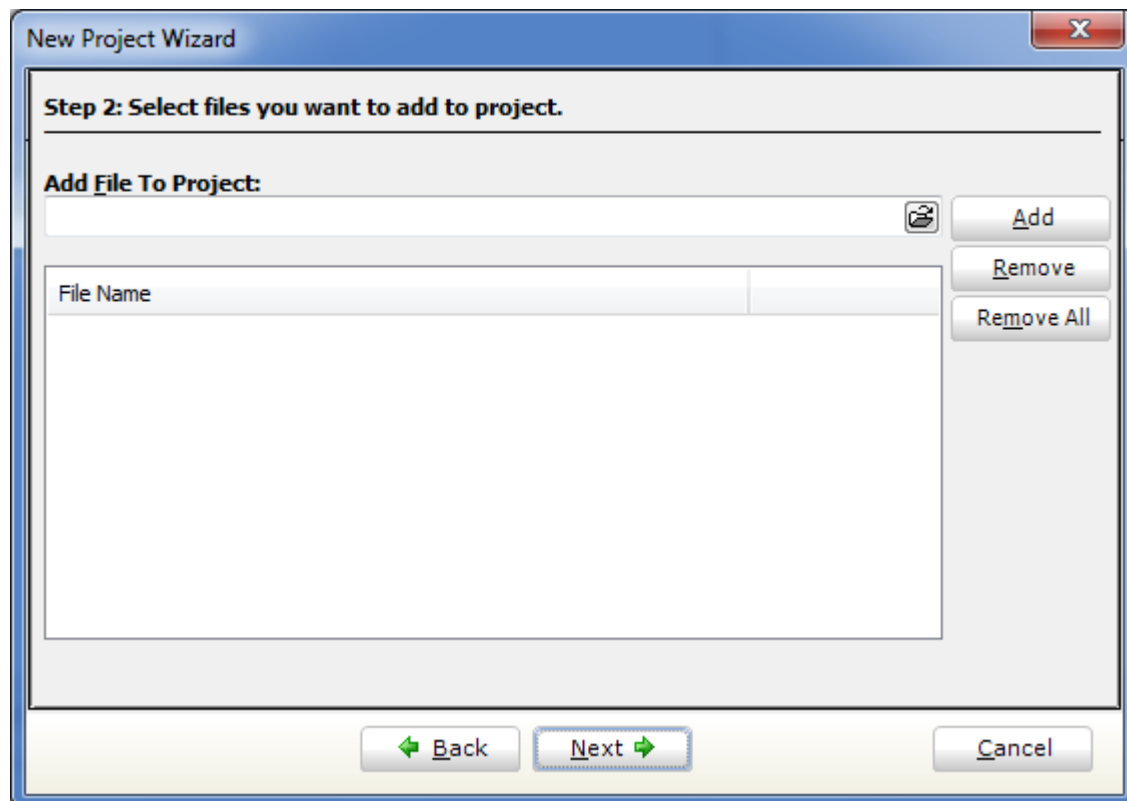
Project folder:

Device Name:

Device Clock: MHz

Enter project name, project folder, select device name and enter a device clock
(for example: 96.235).

Note: Project name and project folder must not be left empty.

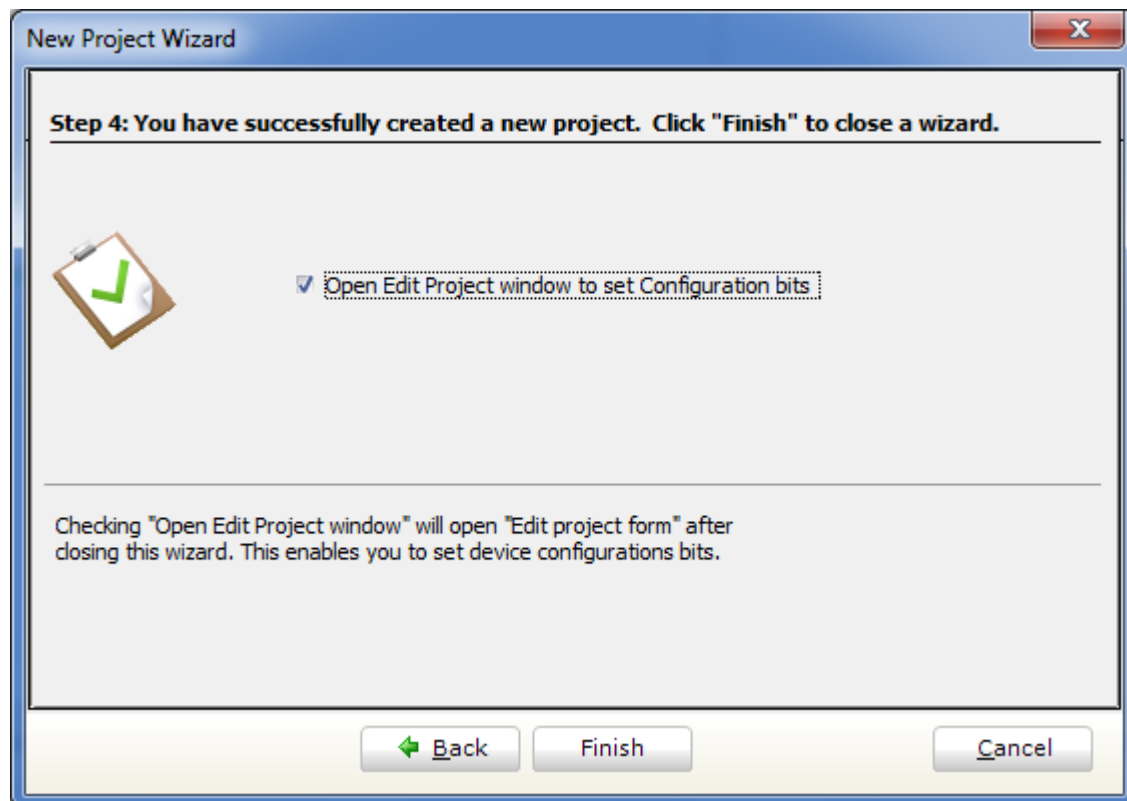
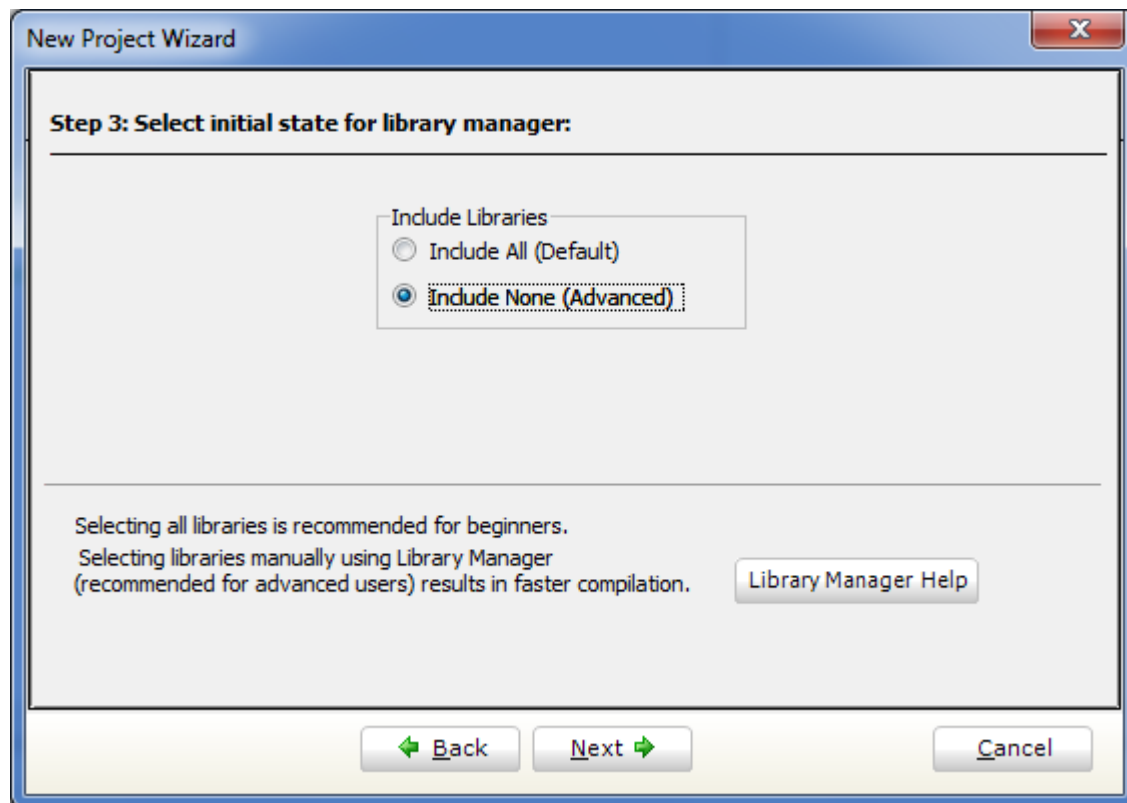


New Project Wizard

Step 2: Select files you want to add to project.

Add File To Project:

File Name



- Sẽ hiện 1 cửa sổ **Edit Project**, chúng ta cần phải chọn các thông số ở đây.

Edit Project

PLL Prescaler Selection
No prescale (4 MHz oscillator input drives PLL directly)

System Clock Postscaler Selection
[Primary Oscillator Src: /1][96 MHz PLL Src: /2]

USB Clock Selection (used in Full-Speed USB mode onl...
USB clock source comes directly from the primary oscillator block

Oscillator Selection
EC oscillator, CLKO function on RA6 (EC)

Fail-Safe Clock Monitor
Disabled

Int/Ext Oscillator Switchover
Disabled

Power-up Timer
Disabled

Brown-out Reset
Enabled

Brown-out Reset Voltage
Minimum setting

USB Voltage Regulator
Disabled

Watchdog Timer
Disabled

MCU and Oscillator
MCU Name: P18F4550
Oscillator Frequency [MHz]: 20.000000

Build Type
☒ Release ☐ ICD Debug

Heap
Size: 0

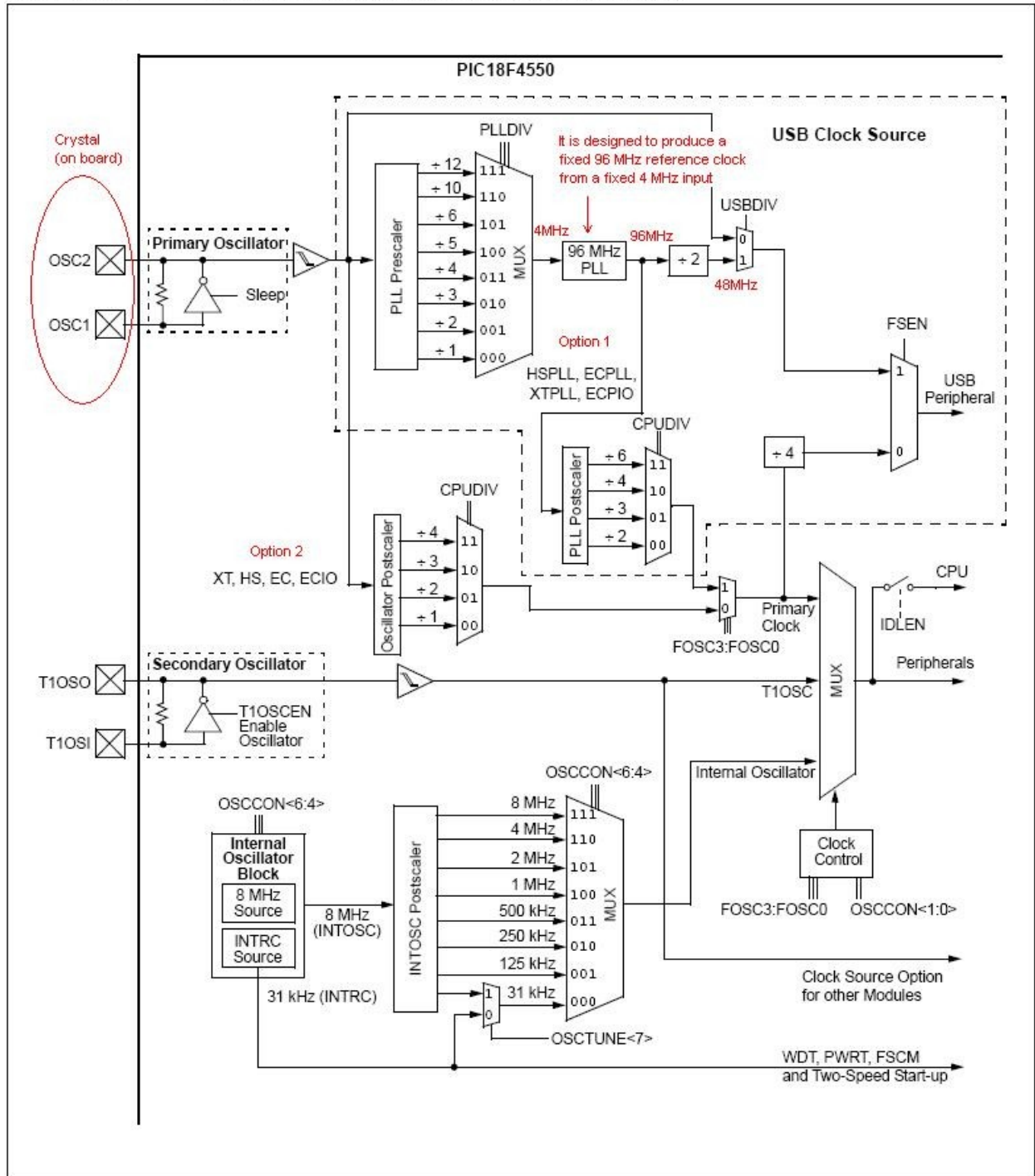
Configuration Registers

CONFIG1L	: \$300000	: 0x0000
CONFIG1H	: \$300001	: 0x0005
CONFIG2L	: \$300002	: 0x001F
CONFIG2H	: \$300003	: 0x001E
CONFIG3H	: \$300005	: 0x0083
CONFIG4L	: \$300006	: 0x0081
CONFIG5L	: \$300008	: 0x000F
CONFIG5H	: \$300009	: 0x00C0
CONFIG6L	: \$30000A	: 0x000F

General Output Settings ...

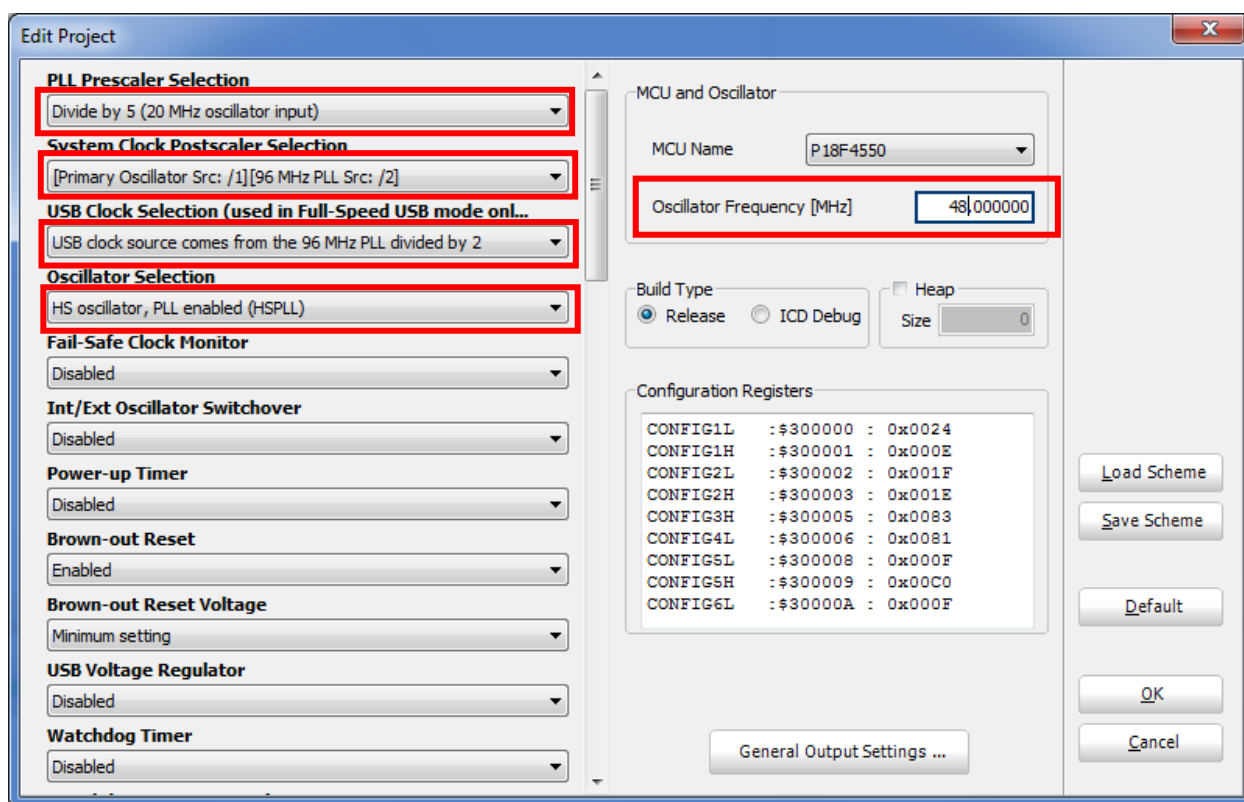
Load Scheme
Save Scheme
Default
OK
Cancel

FIGURE 2-1: PIC18F2455/2550/4455/4550 CLOCK DIAGRAM

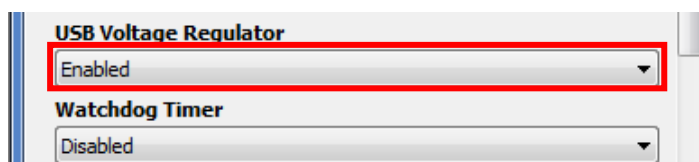


- Ở đây có 2 cách để chọn tần số hoạt động cho CPU, ở hình vẽ chúng ta nhìn thấy có 2 Option là đầu vào xung nhịp cho CPU. Với Option 1 là : HSPLL và Option 2 là: XT, HS.

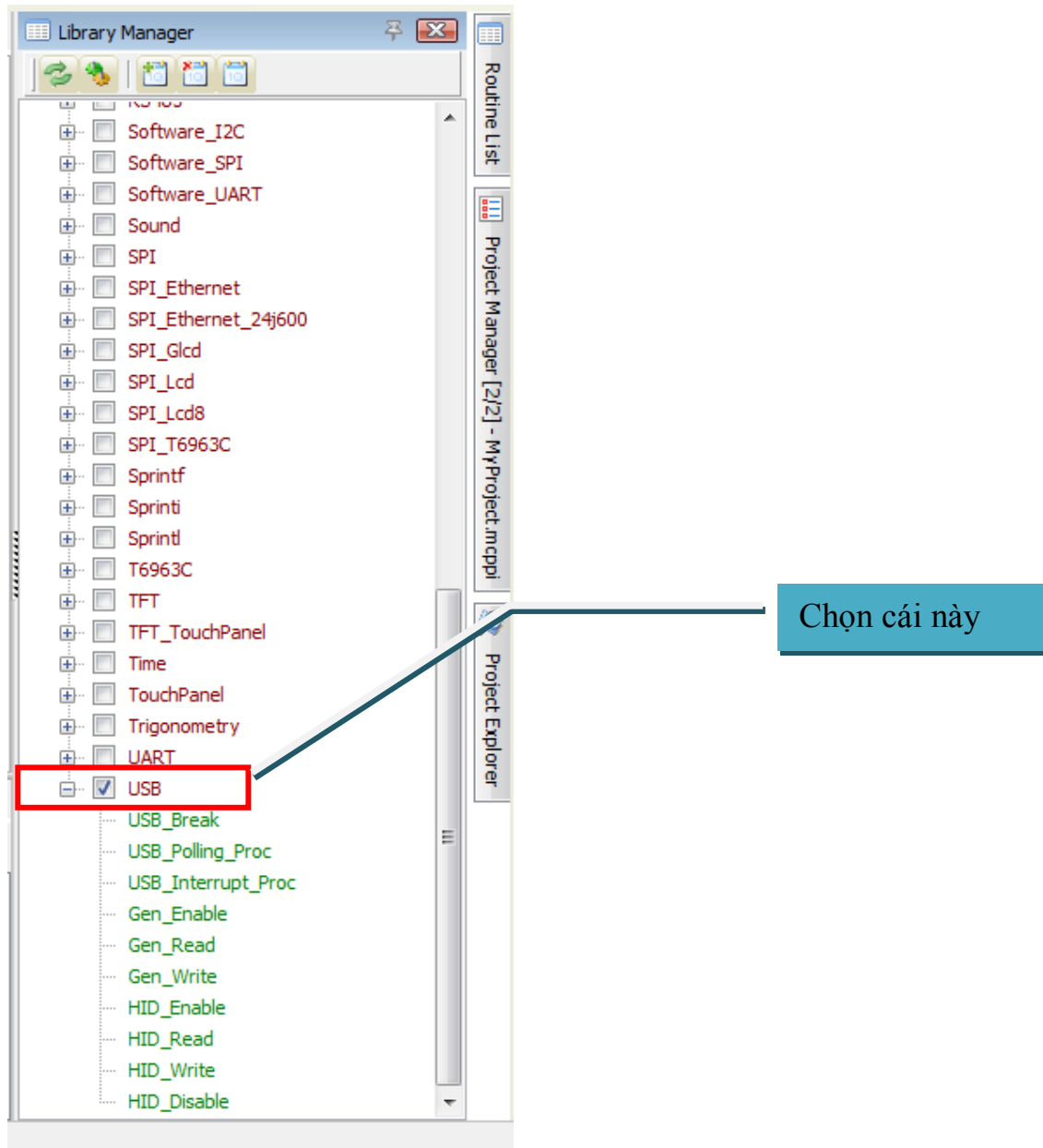
- Để cho đơn giản thì chúng ta sẽ sử dụng HSPLL, tức là Option 1. Thạch anh thật là 20Mhz, qua bộ chia PLLDIV để được 4Mhz và đưa qua bộ 96Mhz PLL, đầu ra nhận được tần số là 96Mhz, cái này sẽ được chia 2 (như hình vẽ) $96 / 2 = 48$ Mhz, vậy là USB hoạt động với xung nhịp là 48Mhz.
- Khi khai báo sử dụng HSPLL thì XT,HS bị vô hiệu, chính vì thế chúng ta sẽ lấy xung nhịp từ bộ HSPLL để cấp cho CPU, như hình trên xung nhịp sẽ được đưa qua bộ chia CPUDIV nhỏ nhất là 2, tức là $96/2 = 48$ Mhz. Điều này có nghĩa là CPU sẽ hoạt động với xung nhịp cao nhất 48Mhz, mặc dù đầu vào thạch anh chỉ có 20Mhz. Và lúc này điều quan trọng cần chú ý là chúng ta phải **sửa lại** là giá trị thạch anh đã đặt cho Project là **20Mhz thành 48Mhz** mặc dù thạch anh thật bên ngoài là 20Mhz. (*Đây chính là điểm cần lưu ý.*)



Ngoài ra chúng ta cần phải Enable cho pin 18 để lấy điện áp 3.3V tại chân này.



- Đến đây coi như Project cơ bản là chuẩn rồi, bây giờ là viết code
- Các bạn có thể Copy & Paste mã này để Demo mạch của mình nhớ chỉnh thạch anh cho phù hợp.



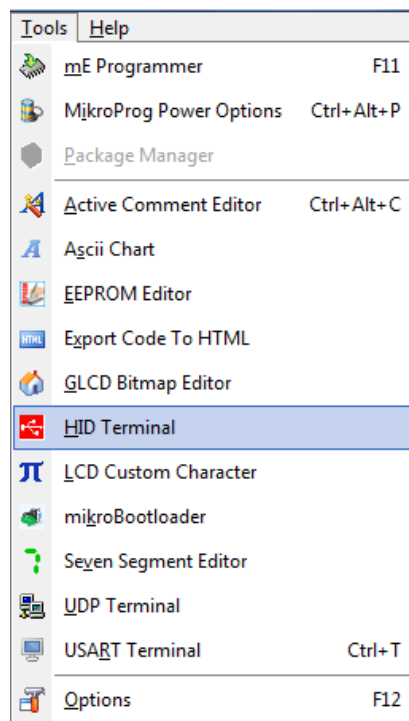
```
/*
Coder : HIENCLUBVN
Lab : Học Viện Hải Quân
Description : USB HID PIC18F4450
Complier : MikroC Pro For PIC 2011
Date : 25 / 09 / 2011
-----
Vd : Xtal = 20Mhz
Chọn HSPLL (PLL DIV 5)
Xung nhịp của USB là 48Mhz
Chọn CPUDIV 2 , xung nhịp của CPU là 48Mhz
*/
unsigned char readbuff[64] absolute 0x500; // Buffers should be in USB RAM, please
consult datasheet
unsigned char writebuff[64] absolute 0x540;

char cnt;
char kk;

void interrupt(){
    USB_Interrupt_Proc(); // Hàm phục vụ ngắt
}

void main(void)
{
    ADCON1 |= 0x0F; // Set lại các PIN Analog thành Digital
    CMCON |= 7; // Disable chức năng Comparators
    HID_Enable(&readbuff,&writebuff); // Cho phép kết nối lớp HID
    while(1){
        while(!HID_Read()); // Trả về số byte được nhận từ HOST, False nếu ko nhận
        được.
        // Chờ ở đây, cho đến khi có data nhận thì thực hiện công việc ghi data nhận vào
        writebuff và truyền lại.
        for(cnt=0;cnt<64;cnt++){
            writebuff[cnt]=readbuff[cnt];
            // Có thể thêm CODE xử lý tại đây.
            while(!HID_Write(&writebuff,64)); // Trả về số byte được truyền đi (64) nếu
            thành công, False nếu thất bại.
        }
    }
}
```

- Coi như xong phần code, bây giờ là đến phần Description cho thiết bị.



Chọn Tool s / HID Terminal

mikroElektronika USB (HID) Terminal

Terminal Descriptor

VID and PID
VID: 04D8
PID: 0003

Report Length
Input: 64
Output: 64

Bus power
Bus powered: ☒
50 x2 mA

Endpoints pooling int.
Input: 1 mSec.
Output: 1 mSec.

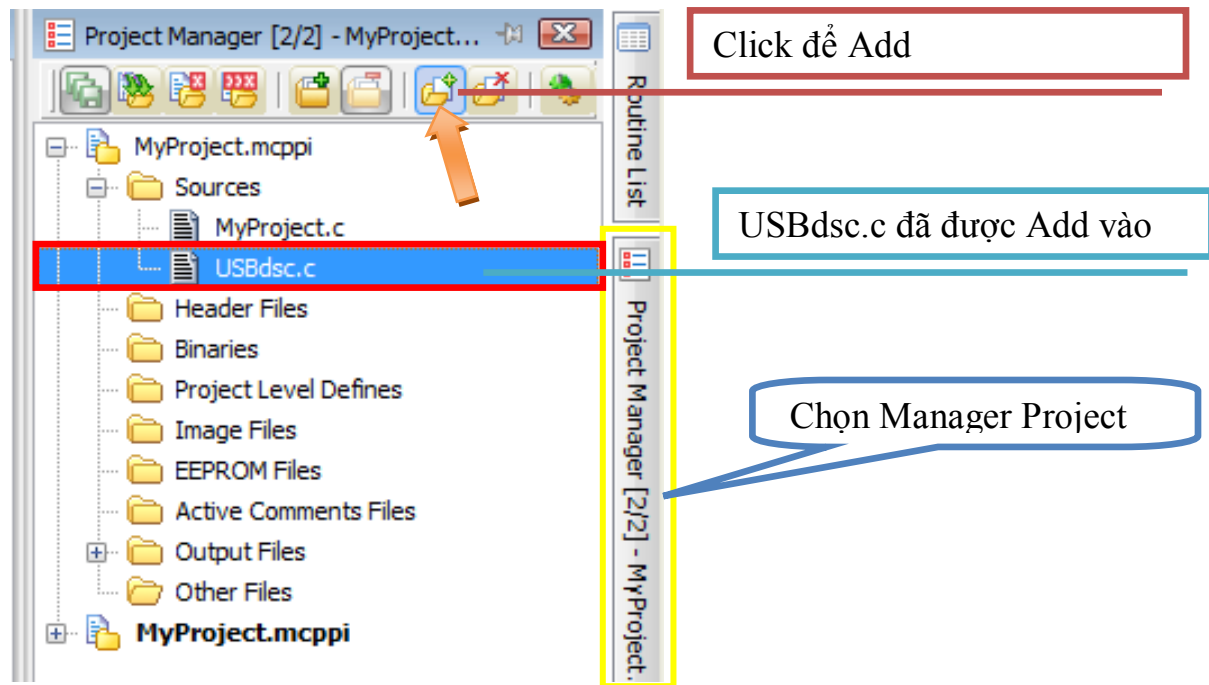
Strings
Vendor Name: hiencclubvn
Product Name: USB HID Library

☒ mikroC ☐ mikroPascal ☐ mikroBasic

Save descriptor

Lấy File tại đây

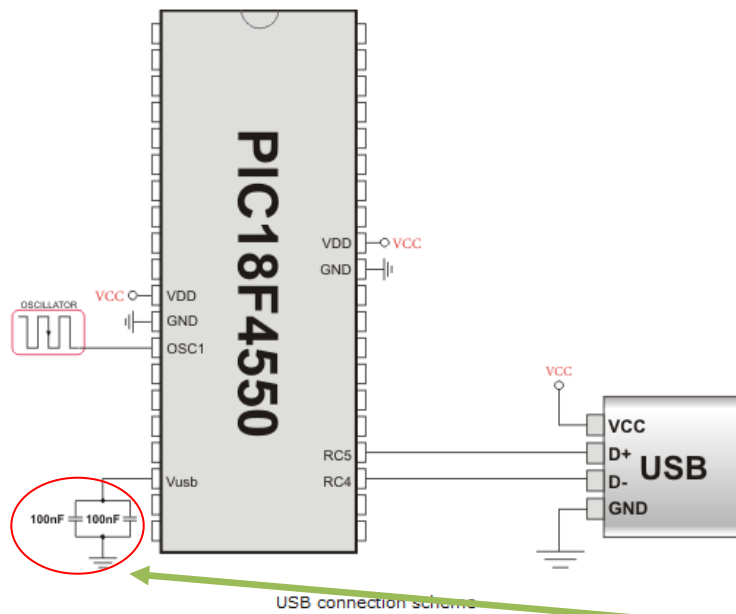
- Lưu nó lại ở thư mục Project đã tạo, và add nó vào Project của mình.



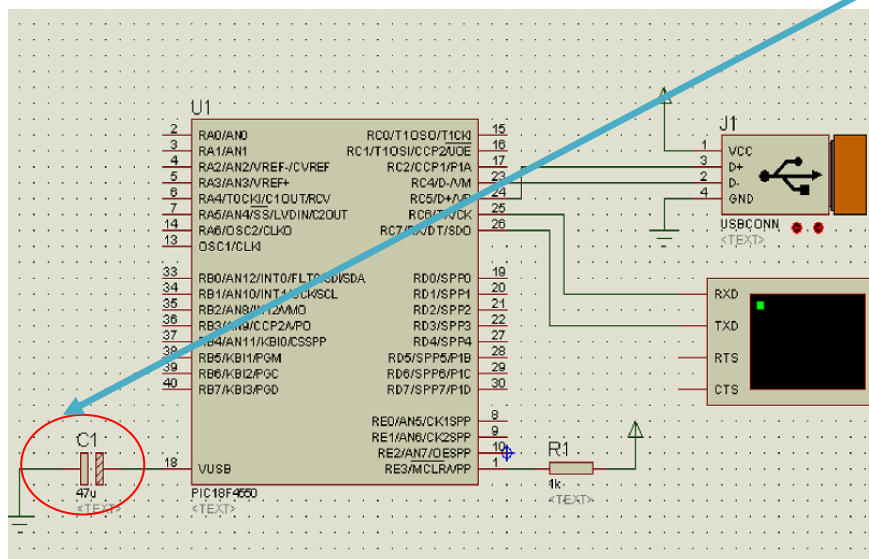
- Bây giờ coi như mọi việc đã xong, nhấn **Ctrl + F9** để **Build**

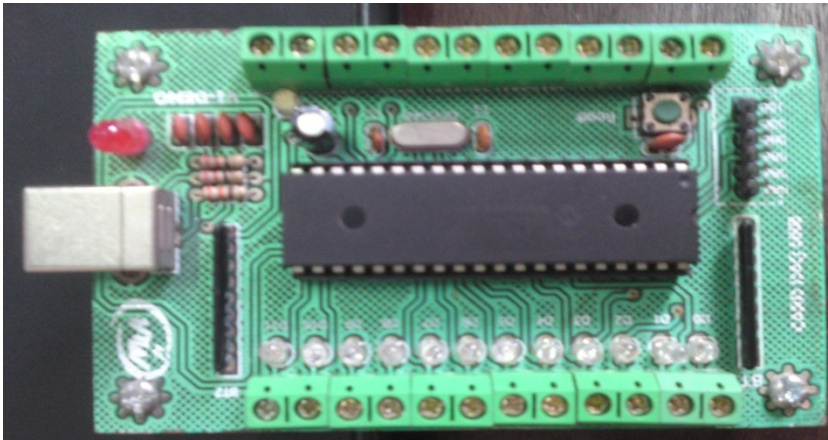
Line	Message No.	Message Text
0	1144	Used ROM (bytes): 5920 (18%) Free ROM (bytes): 26848 (82%)
0	125	Project Linked Successfully
0	128	Linked in 110 ms
0	129	Project 'MyProject.mcppi' completed: 359 ms
0	103	Finished successfully: 25 Sep 2011, 10:44:17

- OK, chương trình của chúng ta đã Finished Successfully, ko có lỗi.
- Tiến hành nạp File HEX trong Project bằng Pickit2 là OK rồi.
- Về Schematic có thể ghép nối như 2 hình bên dưới. (Nên thêm các tụ 104 để lọc nguồn giữa 2 Vcc và GND)



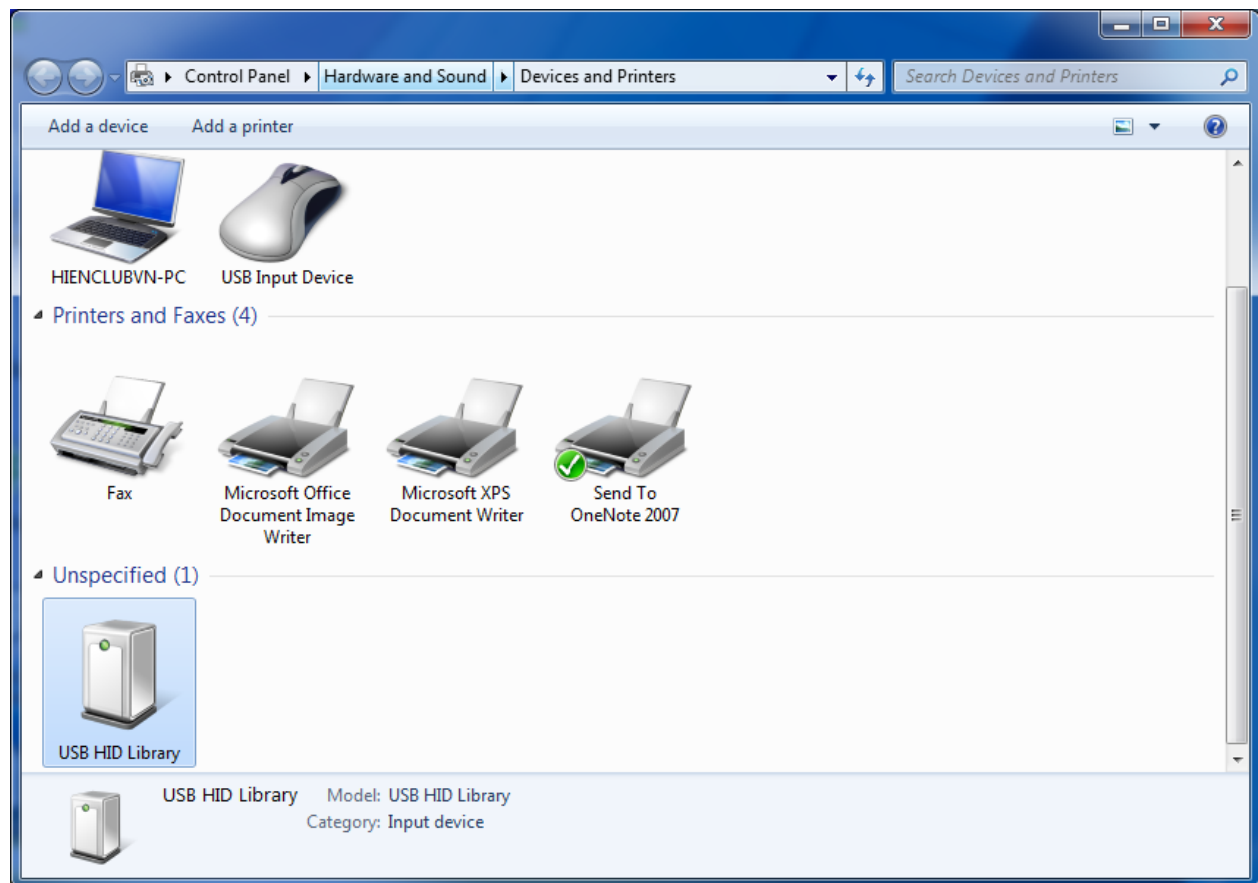
Có thể dùng 2 tụ
104 nối song song
như hình 1 hoặc tụ
470nF như hình 2



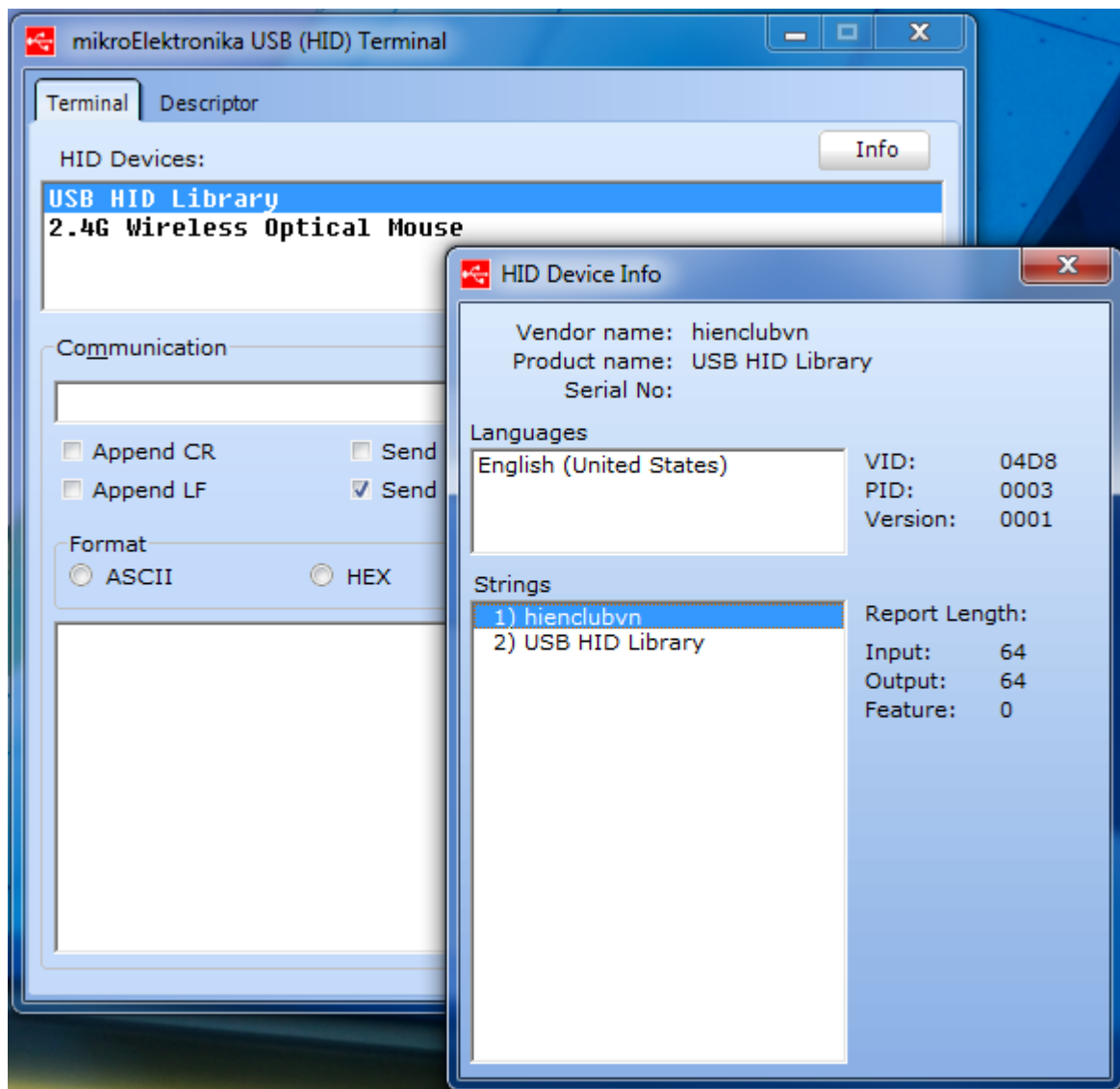


- Sau khi đã nạp Firmware , thì các bạn có thể thấy thiết bị của mình rồi. Lưu ý mình sai lớp HID nên cảm phát là PC load được thiết bị ngay.

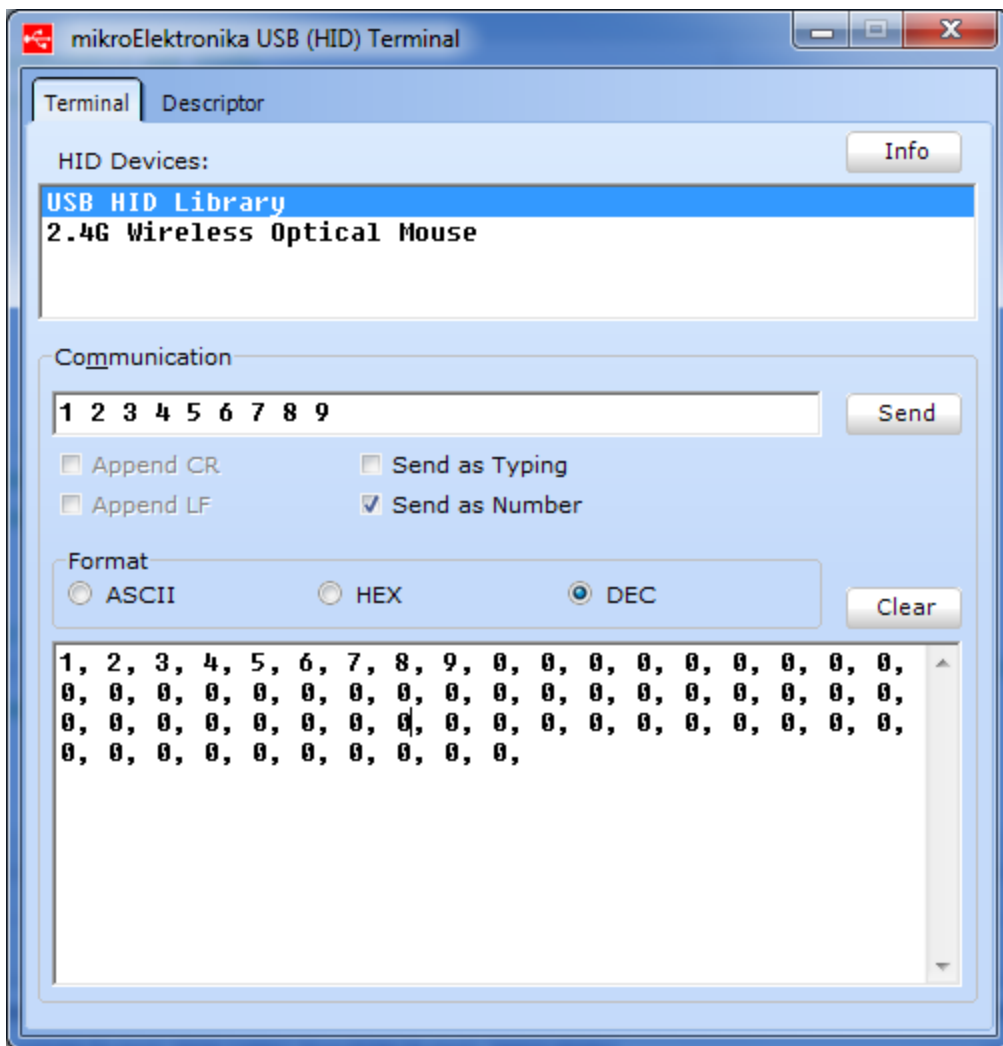




- Ok, đến đây là ngon rồi. Bây giờ chúng ta bật **HID Terminal** của MikroC lên để Debug chương trình.



- Chúng ta thấy USB HID Library (tên đã tạo ra trước đó bằng Description), nhấn **info** để xem thông tin về thiết bị.



- ⇒ Đến đây coi như mọi việc đã xong, bây giờ chỉ là vấn đề viết giao tiếp nữa mà thôi. Để cho đơn giản tôi khuyên các bạn nên code bằng các hàm dll share trên internet, có thể là của Microchip, www.lvr.com, www.florian-leitner.de, www.waitingforfriday.com/
- Theo lời ý kiến cá nhân của tôi các bạn nên đầu tư nghiên cứu code của www.lvr.com hoặc là www.waitingforfriday.com/

3. Phát triển ứng dụng với HID USB Library

- Tạm thời đến đây đã, mệt rồi. Hôm sau viết tiếp.