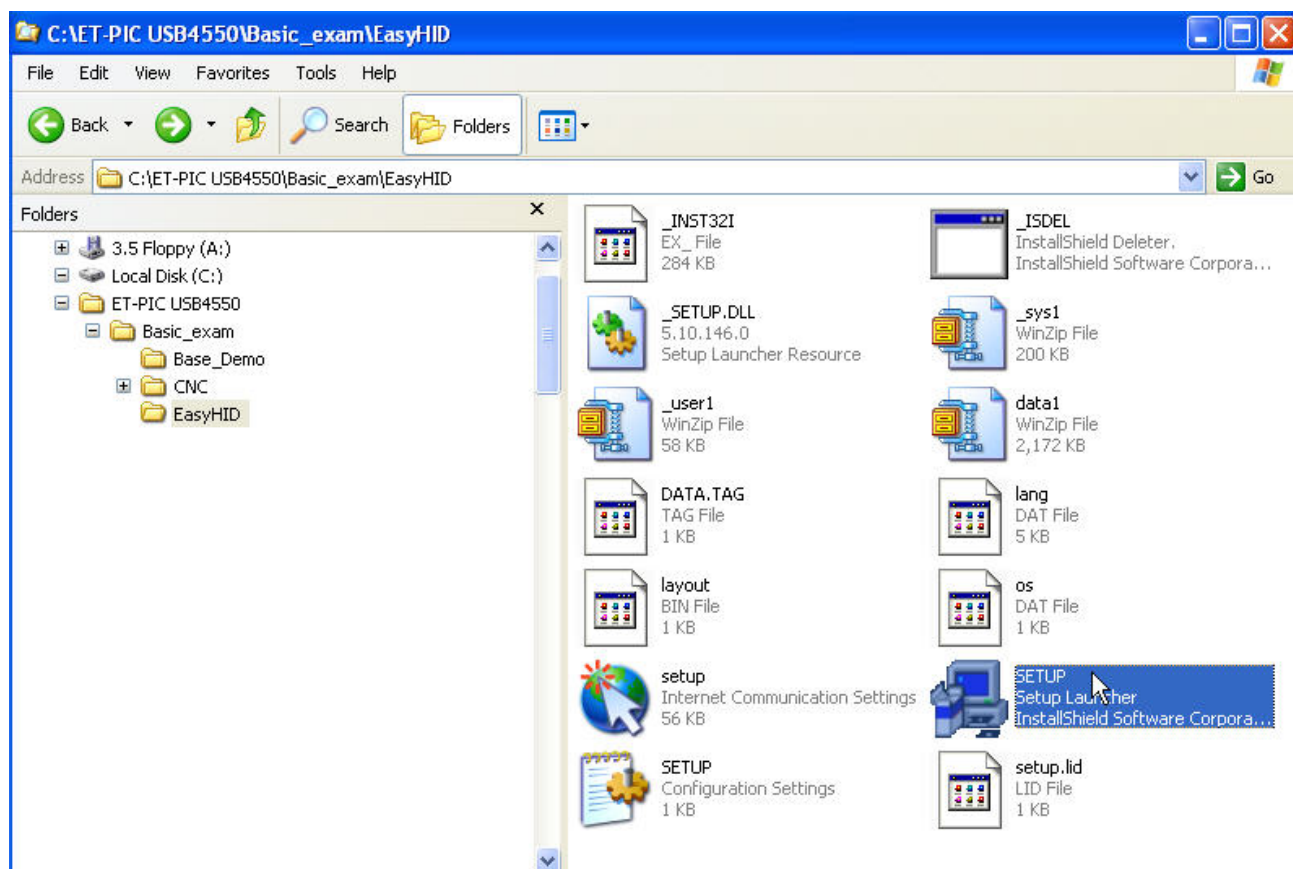


การใช้งานโปรแกรม EasyHID

EasyHID เป็นโปรแกรมซอฟต์แวร์ ที่สร้างขึ้นมาเพื่อสนับสนุนการใช้งานไมโครคอนโทรลเลอร์ตระกูล PIC ที่มีโมดูล USB (Universal Serial Bus) เช่น PIC18F2550 , PIC18F4550 เป็นต้น โดยความสามารถของซอฟต์แวร์ตัวนี้คือ สามารถทำการสร้าง HID ของ อุปกรณ์ USB ได้ โดยสามารถสร้างโปรแกรมในส่วนพื้นฐานให้ ทั้ง ภาษาเบสิกของไมโครคอนโทรลเลอร์ (PicBasic Pro) และ สร้างโปรแกรมพื้นฐานของโปรแกรมบนคอมพิวเตอร์ คือ Visual Basic , Visual C++ และ Delphi โดยสามารถศึกษาเรียนรู้ได้จากตัวอย่างต่อไปนี้

การติดตั้งโปรแกรม EasyHID

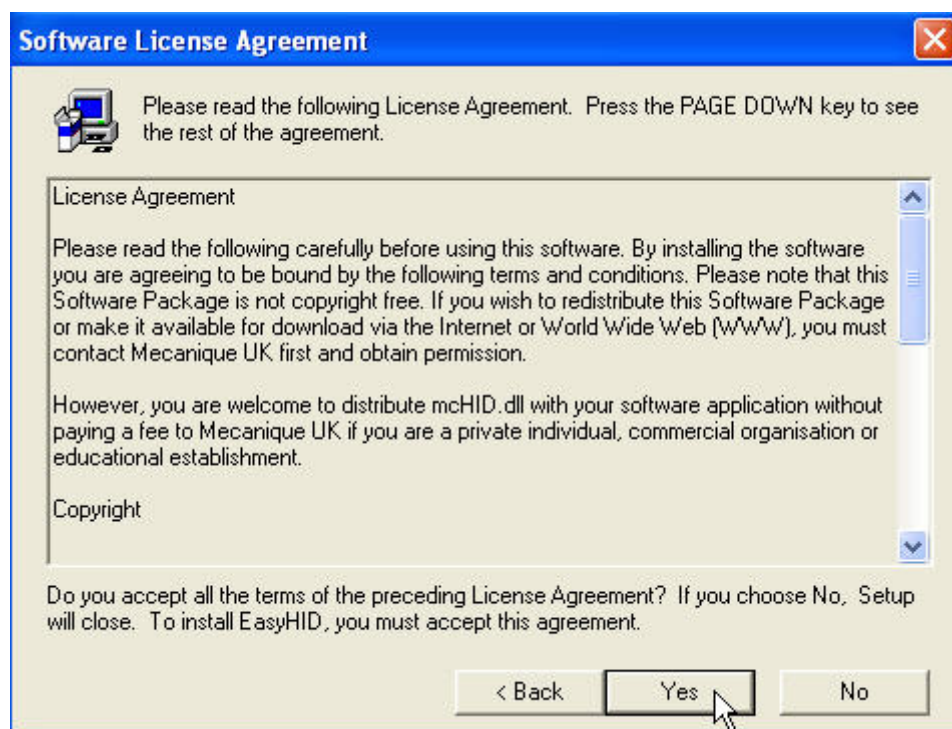
- เปิดโฟลเดอร์ EasyHID และ ดับเบิลคลิก ไอคอน SETUP ดังรูป



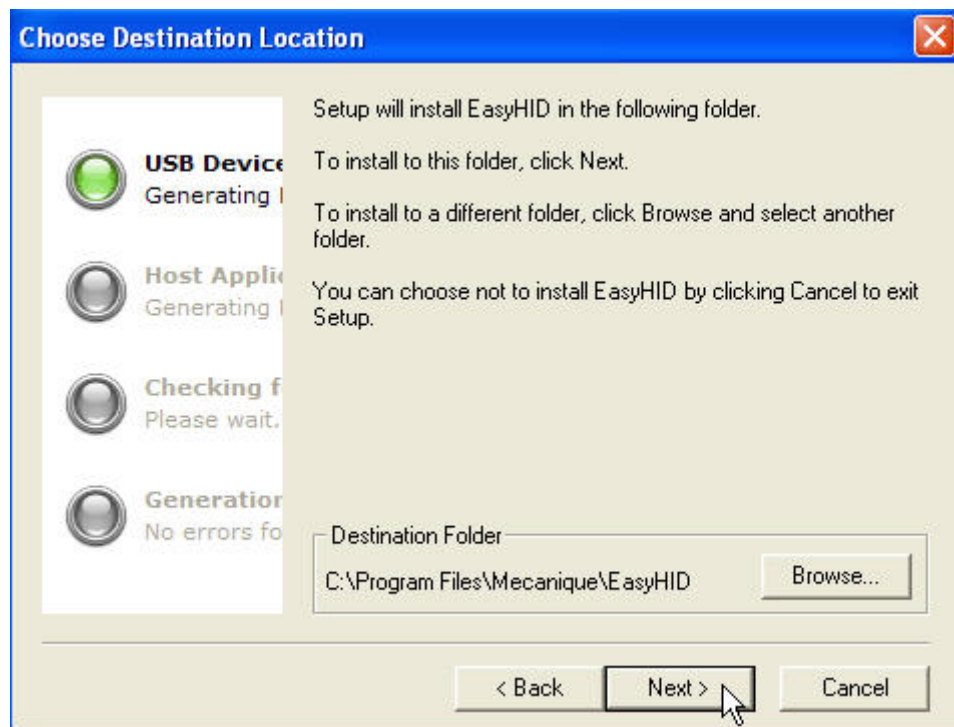
- จะปรากฏหน้าต่าง Welcome ให้คลิก Next >



- คลิก Yes เพื่อยอมรับ License



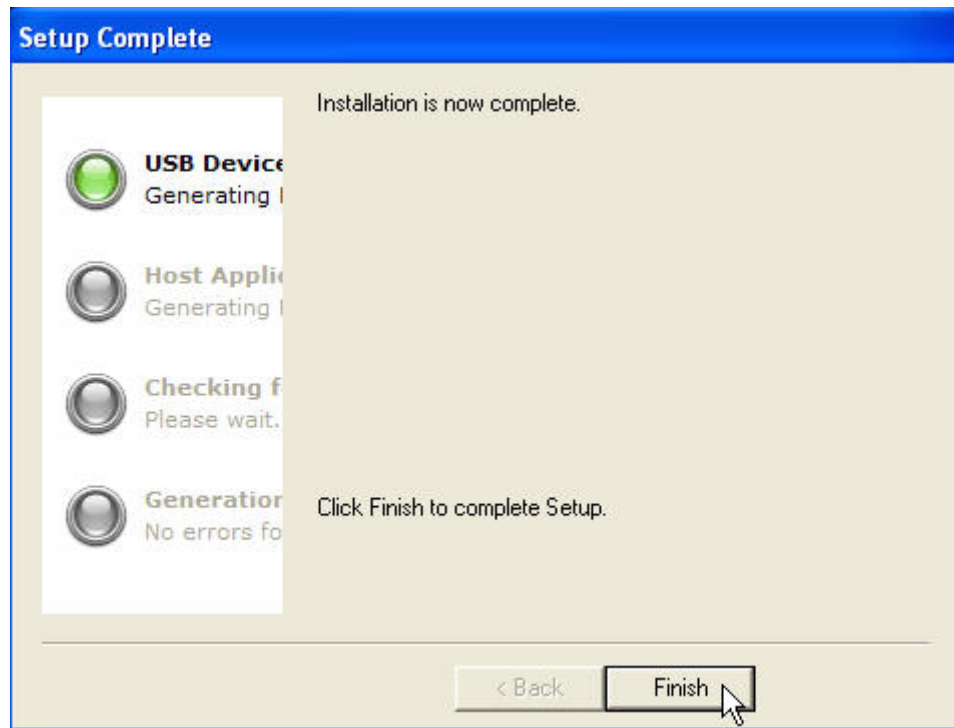
- เลือกตำแหน่งปลายทางของการติดตั้งจากนั้นคลิก Next >



- คลิก Next > โปรแกรมจะเริ่มกระบวนการติดตั้งจนเสร็จสิ้น



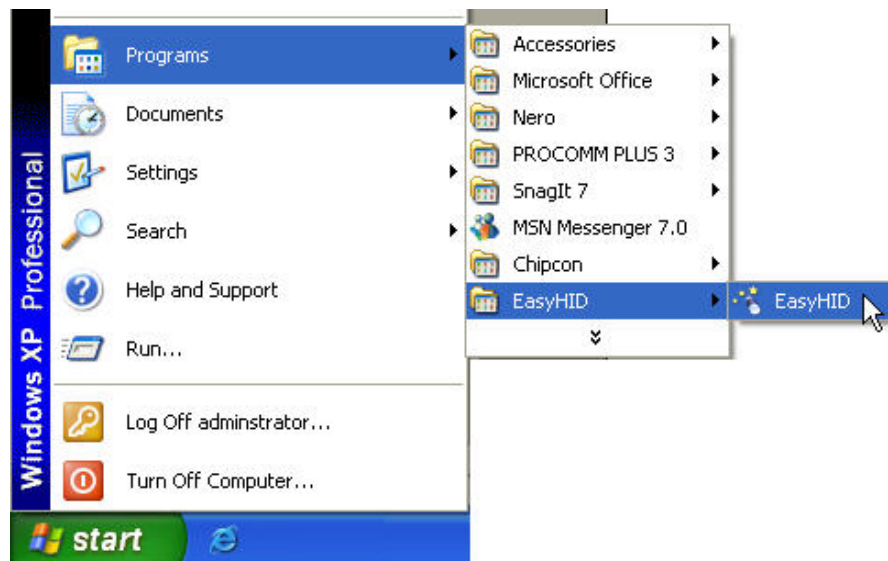
- คลิก Finish เพื่อจบการติดตั้งโปรแกรม



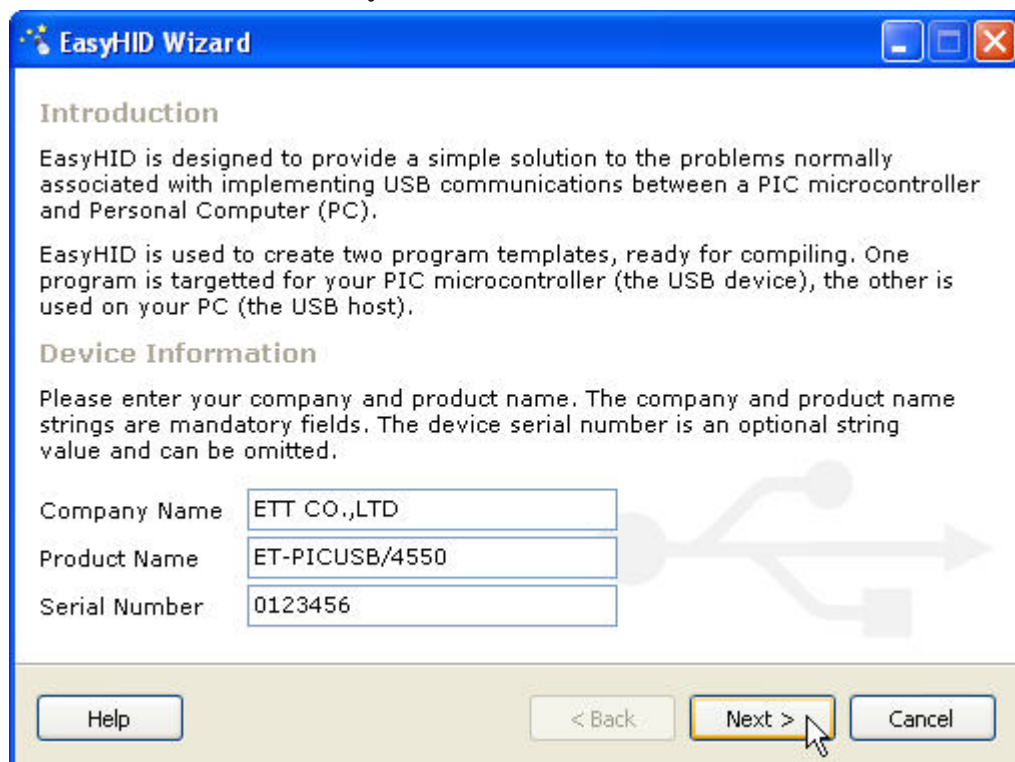
ตัวอย่างขั้นตอนการใช้งาน EasyHID

ตัวอย่างต่อไปนี้จะเป็นการออกแบบแอฟริเคชัน โดยแบ่งออกเป็นสองส่วนหลักๆ คือ ไมโครคอนโทรลเลอร์ ใช้เบอร์ PIC18F4550 (ฮาร์ดแวร์อ้างอิงกับบอร์ด ET-PICUSB/4550) และ คอมพิวเตอร์ ซึ่งจะพัฒนาด้วยโปรแกรม Visual Basic การทำงานของโปรแกรมคือ จะส่งข้อความจากโปรแกรมคอมพิวเตอร์ผ่านระบบบัส USB มาที่ ไมโครคอนโทรลเลอร์ PIC18F4550 และ แสดงผลข้อความที่จอ LCD ซึ่งสามารถศึกษาและทดลองทำตามขั้นตอนต่างๆ ต่อไปนี้

- เปิดโปรแกรม EasyHID โดยคลิกที่ Start > Programs > EasyHID > EasyHID ดังรูปต่อไปนี้



- จะปรากฏหน้าต่าง EasyHID Wizard ให้เราได้ชื่อ Company Name , Product Name และ Serial Number ตามที่เราต้องการ ดังเช่นตัวอย่างในรูป จากนั้นคลิก Next >



- ใส่หมายเลข Vendor ID และ Product ID ที่ต้องการดังตัวอย่าง แล้วคลิก Next >

EasyHID Wizard

Vendor and Product ID

Please enter a valid Vendor ID (VID) and Product ID (PID).

Vendor ID: 9999

Product ID: 0001

If you intend to ship a USB device you need an official USB Vendor ID, which is unique throughout the world. Vendor ID's are assigned by the USB implementers forum at www.usb.org. Use the default values above FOR TESTING ONLY.

Alternatively, Mecanique own a USB Vendor ID and can provide an individual or a company with a set of product ID's at very low cost. This means that your product can be shipped world wide with a guaranteed unique and unambiguous VID and PID combination.

☐ Purchase a unique set of Products IDs so that I can distribute my product...

☐ Tell me more about using Vendor and Product ID's...

Help < Back Next > Cancel

- กำหนดคุณสมบัติที่ต้องการ ซึ่งสามารถเลือกขนาดของ Buffer (Input) และ Buffer (Output) ได้ตามต้องการ สูงสุดไม่เกิน 64 ไบต์ ดังตัวอย่างต่อไปนี้ จะกำหนดไว้ 16 ไบต์ เสร็จแล้วคลิก Next >

EasyHID Wizard

Configuration Details

The input polling interval is used by the host to request data from a USB device. The output polling interval is used by the host to send data to a USB device. Bus power is the maximum power consumption (x2) of the USB device on the bus.

Polling (Input): 10 ms - host requests data from a USB device (max latency)

Polling (Output): 10 ms - host sends data to a USB device (max latency)

Bus Power: 50 x2 mA

The input buffer (report) is sent by a USB device when requested to do so by the host. The output buffer (report) is sent by the host to a USB device.

Buffer (Input): 16 bytes - USB device to host (64 bytes max)

Buffer (Output): 16 bytes - host to USB device (64 bytes max)

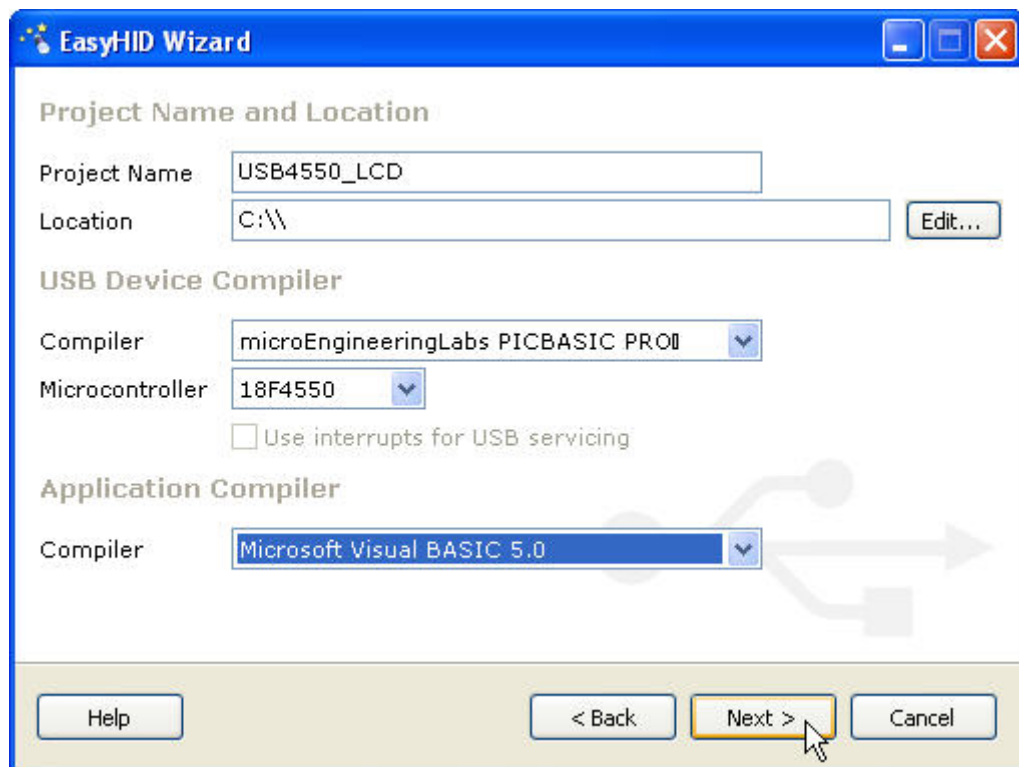
If you are unsure about the correct values to enter, use the recommended defaults.

Defaults

Help < Back Next > Cancel

- กำหนดรายละเอียดต่างๆ ตามที่เราต้องการออกแบบ ดังตัวอย่างต่อไปนี้
 - Project Name = ชื่อโปรเจกต์ที่เราต้องการในตัวอย่างคือ USB4550_LCD
 - Location = ตำแหน่งที่ต้องการเก็บไฟล์เคอร์ที่จะสร้างขึ้น
 - USB Device Compiler
 - Compiler คือ ตัวแปลภาษาที่ใช้
 - Microcontroller = เบอร์ของไมโครคอนโทรลเลอร์ที่เราจะออกแบบ
 - Application Compiler
 - Compiler = ภาษาของโปรแกรมที่ใช้เขียนโปรแกรมบน PC

เมื่อกำหนดรายละเอียดต่างๆ ตามต้องการ หรือ ตามตัวอย่างในรูปด้านล่าง เสร็จแล้วให้คลิก next > เพื่อเข้าสู่ขั้นตอนต่อไป

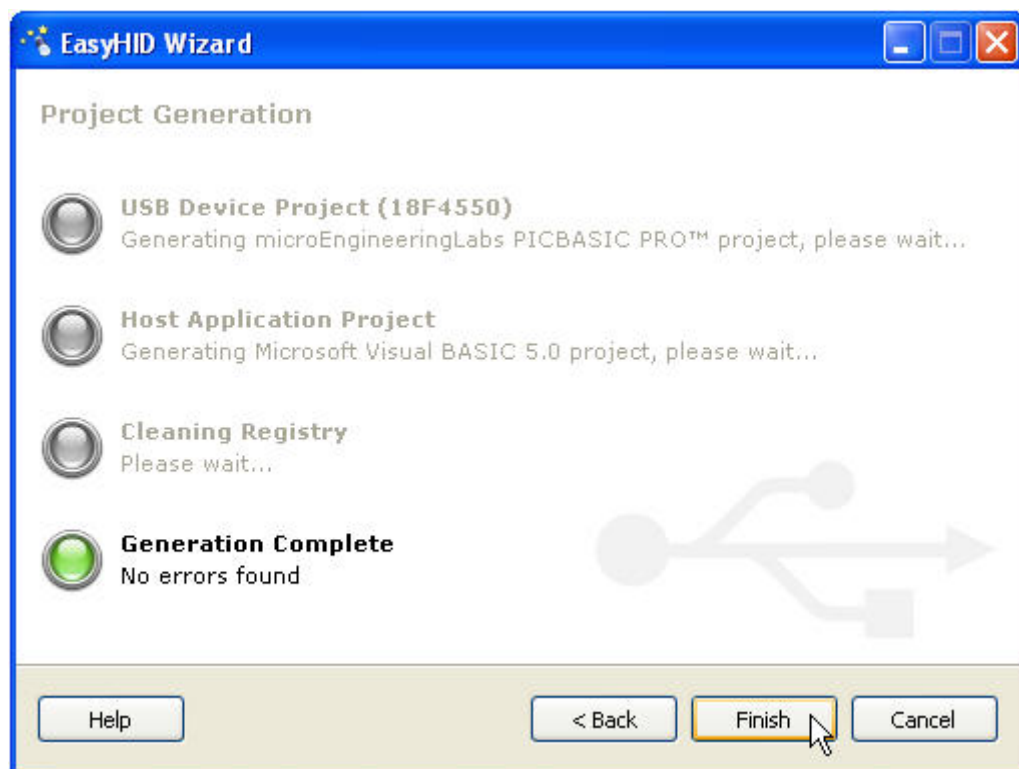


The screenshot shows the 'EasyHID Wizard' window with the following settings:

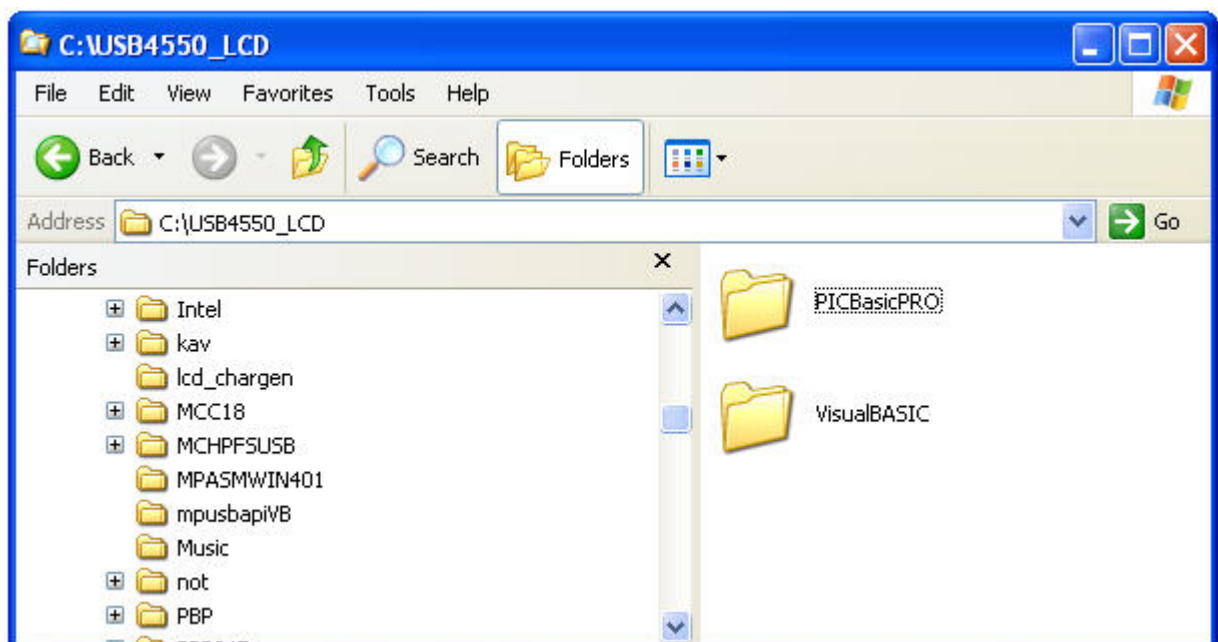
- Project Name and Location:**
 - Project Name: USB4550_LCD
 - Location: C:\ (with an 'Edit...' button)
- USB Device Compiler:**
 - Compiler: microEngineeringLabs PICBASIC PRO (dropdown)
 - Microcontroller: 18F4550 (dropdown)
 - ☐ Use interrupts for USB servicing
- Application Compiler:**
 - Compiler: Microsoft Visual BASIC 5.0 (dropdown)

At the bottom, there are buttons for 'Help', '< Back', 'Next >' (highlighted with a mouse cursor), and 'Cancel'.

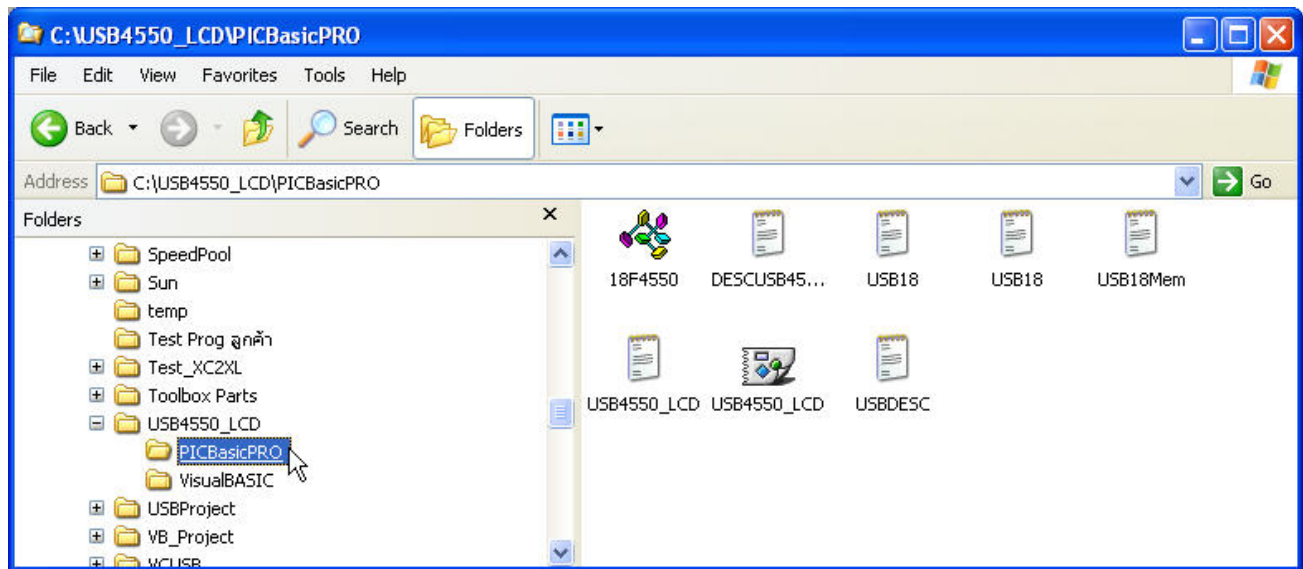
- โปรแกรมจะเริ่มกระบวนการต่างๆ จนจบกระบวนการ ให้เรากด Finish



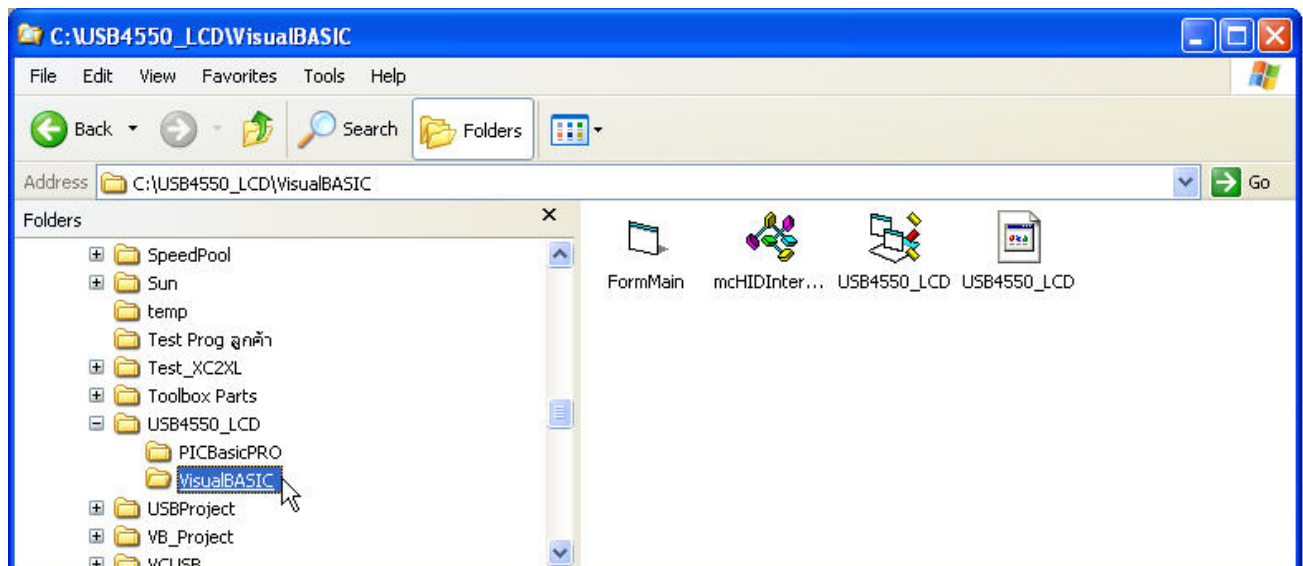
- จากนั้นให้เราเข้าไปดูในตำแหน่ง Location ที่เรากำหนดไว้ จะเห็นโฟลเดอร์ เกิดขึ้นมาตามที่เรที่ตั้งชื่อไว้ ซึ่งในตัวอย่างนี้คือ USB4550_LCD ซึ่งภายในจะประกอบไปด้วยโฟลเดอร์อีก 2 โฟลเดอร์ คือ PicBasicPRO และ VisualBASIC ดังรูปต่อไปนี้



- โดยภายในจะเป็นโค้ดโปรแกรมที่มีฟังก์ชันพื้นฐานเกี่ยวกับ USB ไว้ให้ รูปต่อไปนี้เป็นโปรแกรมพื้นฐานของ PicBasic Pro ที่โปรแกรม EasyHID สร้างขึ้นไว้



- ส่วนรูปต่อไปนี้เป็นโปรแกรม Visual Basic ที่โปรแกรม EasyHID สร้างขึ้นไว้



- ซึ่งเราสามารถนำโค้ดที่ได้นี้ไปพัฒนาต่อตามความต้องการของเราได้เลย

- โค้ดโปรแกรมของ PicBasic Pro ที่ถูกสร้างขึ้นมา จะมีฟังก์ชัน DoUSBIn สำหรับรับข้อมูลจาก USB เข้ามายังตัวแปร USBBuffer และ ฟังก์ชัน DoUSBOut สำหรับส่งข้อมูลออกไปที่ บัส USB

```

MicroCode Studio Plus - PICBASIC PRO (C:\USB4550_LCD\PICBasicPRO\USB4550_LCD.pbp)
File Edit View Project Help

Code Explorer
  Includes
  Defines
    OSC
    LOADER_USED
  Constants
    USBBufferSizeMax
    USBBufferSizeTX
    USBBufferSizeRX
  Variables
    USBBuffer
    USBBufferCount
  Alias and Modifiers
  Symbols
  Labels
    ProgramStart
    DoUSBIn
    DoUSBOut

DEFINE OSC 48
DEFINE LOADER_USED 1

USBBufferSizeMax CON 16 ' maximum buffer size
USBBufferSizeTX CON 16 ' input
USBBufferSizeRX CON 16 ' output

' the USB buffer...
USBBuffer VAR BYTE[USBBufferSizeMax]
USBBufferCount VAR BYTE

' *****
' * main program loop - remember, you must keep the USB *
' * connection alive with a call to USBService every couple *
' * of milliseconds or so... *
' *****
USBINIT ' initialise USB...
ProgramStart:
  GOSUB DoUSBIn
  GOSUB DoUSBOut
  GOTO ProgramStart

' *****
' * receive data from the USB bus *
' *****
DoUSBIn:
  USBBufferCount = USBBufferSizeRX ' RX buffer size
  USBSERVICE ' keep connection alive
  USBIN 1, USBBuffer, USBBufferCount, DoUSBIn ' read data, if available
  RETURN

' *****
' * wait for USB interface to attach *
' *****
DoUSBOut:
  USBBufferCount = USBBufferSizeTX ' TX buffer size
  USBSERVICE ' keep connection alive
  USBOUT 1, USBBuffer, USBBufferCount, DoUSBOut ' if bus available, transx
  RETURN
  
```

โดยในตัวอย่างนี้เราจะรับค่าจาก USB แล้ว แสดงผลที่จอ LCD ซึ่งโปรแกรมสมบูรณ์ที่พัฒนาต่อจากโปรแกรมพื้นฐานที่ EasyHID ออกแบบให้จะเป็นดังต่อไปนี้

ตัวอย่างโค้ดของ PicBasic Pro

```

DEFINE OSC 48

DEFINE LOADER_USED 1

Define LCD_DREG  PORTD
Define LCD_DBIT  4
Define LCD_RSREG  PORTD
Define LCD_RSBIT  2
Define LCD_EREG  PORTD
Define LCD_EBIT  3
Define LCD_BITS  4


USBBufferSizeMax  con 16 ' maximum buffer size
USBBufferSizeTX   con 16 ' input
USBBufferSizeRX   con 16 ' output


' the USB buffer...
USBBuffer      Var Byte[USBBufferSizeMax]
USBBufferCount Var Byte


i  var byte

' *****
' * main program loop - remember, you must keep the USB      *
' * connection alive with a call to USBService every couple *
' * of milliseconds or so...                                  *
' *****

usbinit ' initialise USB...

    pause 1500

    LCDOut $fe, 1 ' Clear LCD screen

    LCDOut "CP-PICUSB4550" ' Display Hello


ProgramStart:

    gosub DoUSBIn

    LCDOUT $FE,$C0,"      " ' Clear Display Line 2 of LCD

    LCDOUT $FE,$C0      ' Point to Frist Address of Line 2 LCD

```

```
for i = 0 to 15
```

```
    LCDOUT usbbuffer[i]
```

```
next i
```

```
gosub DoUSBOut
```

```
goto ProgramStart
```

```
' *****
```

```
' * receive data from the USB bus          *
```

```
' *****
```

```
DoUSBIn:
```

```
    USBBufferCount = USBBufferSizeRX      ' RX buffer size
```

```
    USBService      ' keep connection alive
```

```
    USBIn 1, USBBuffer, USBBufferCount, DoUSBIn ' read data, if available
```

```
    return
```

```
' *****
```

```
' * wait for USB interface to attach      *
```

```
' *****
```

```
DoUSBOut:
```

```
    USBBufferCount = USBBufferSizeTX      ' TX buffer size
```

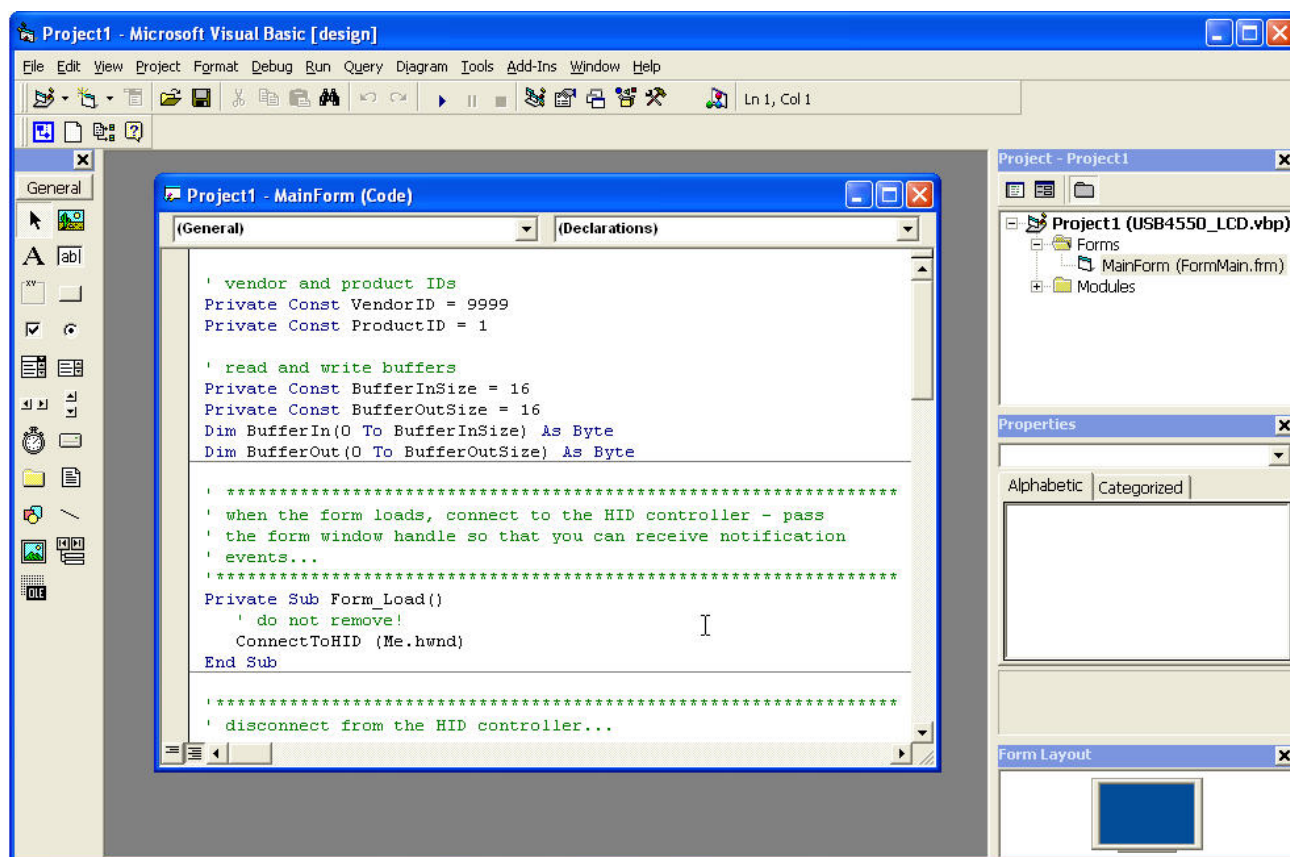
```
    USBService      ' keep connection alive
```

```
    USBOut 1, USBBuffer, USBBufferCount, DoUSBOut ' if bus available, transmit data
```

```
    return
```

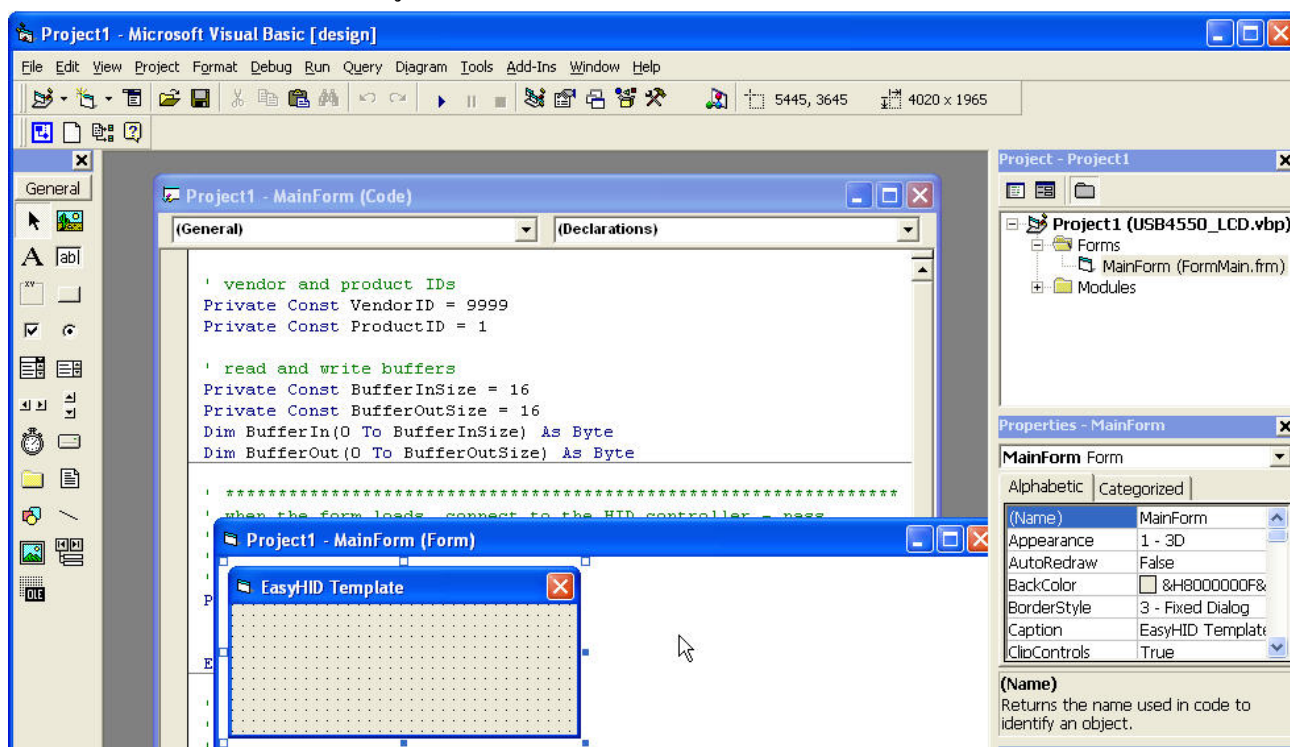
- หลังจากเขียนโค้ดเสร็จคอมไพล์ และ โปรแกรมลง PIC18F4550

- ขั้นตอนต่อไป พัฒนาโปรแกรมด้วยภาษา Visual Basic ต่อจากโปรแกรมพื้นฐานที่ EasyHID สร้างไว้ให้
- โดยเปิดโค้ดภาษา Visual Basic ด้วยโปรแกรม VB6 แล้วทำการพัฒนาโค้ดโปรแกรมต่อดังนี้

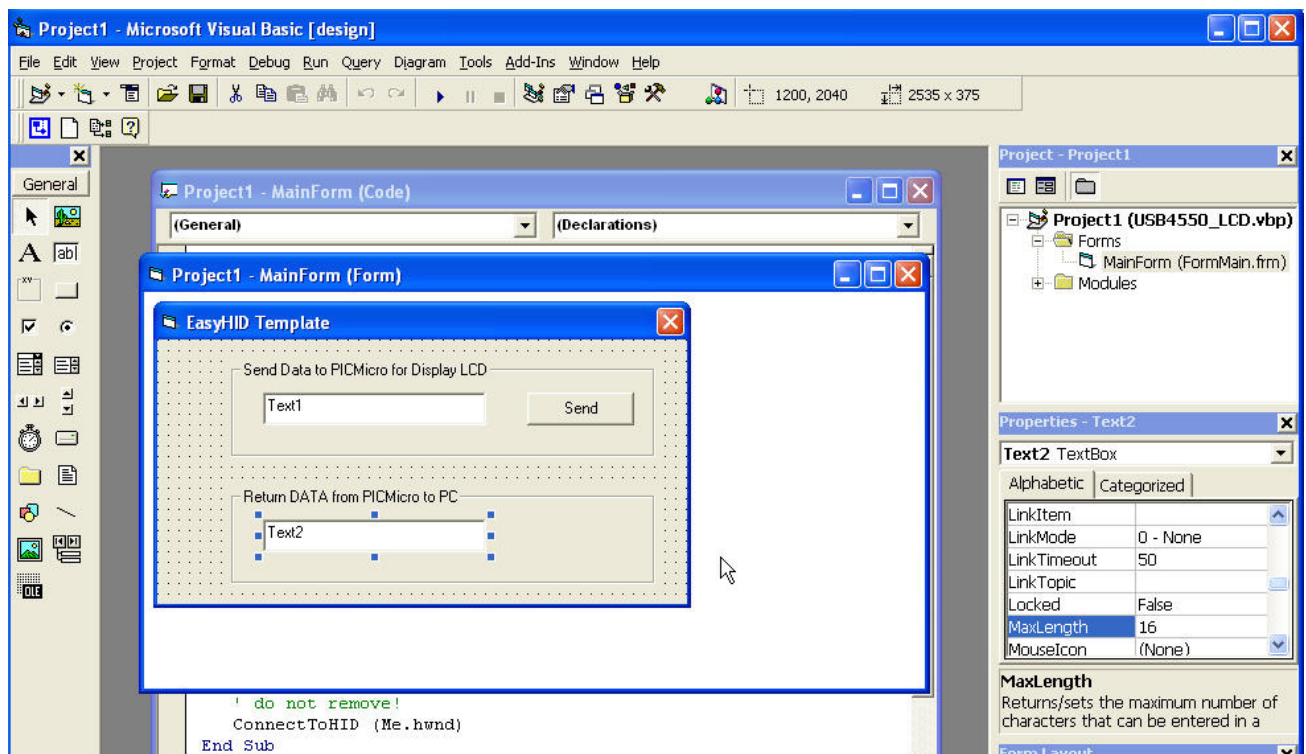


รูปแสดงโค้ดโปรแกรมพื้นฐานที่โปรแกรม EasyHID สร้างไว้ให้

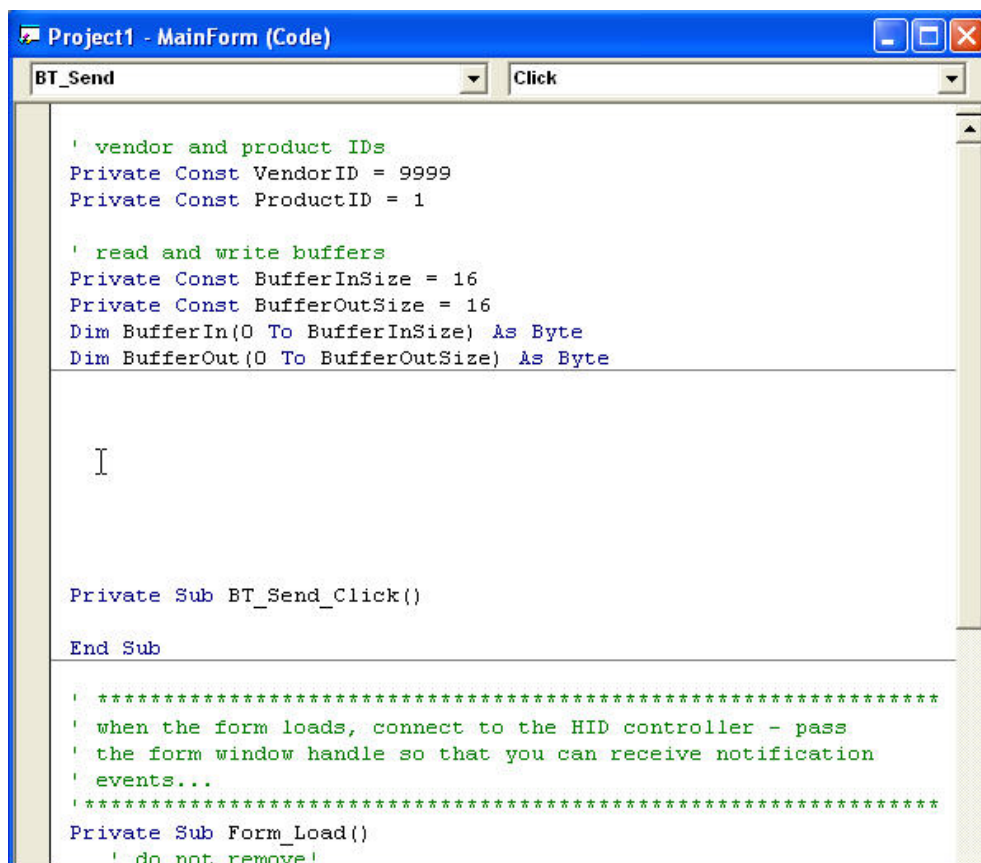
- ทำการสร้างฟอร์มใหม่ดังรูป



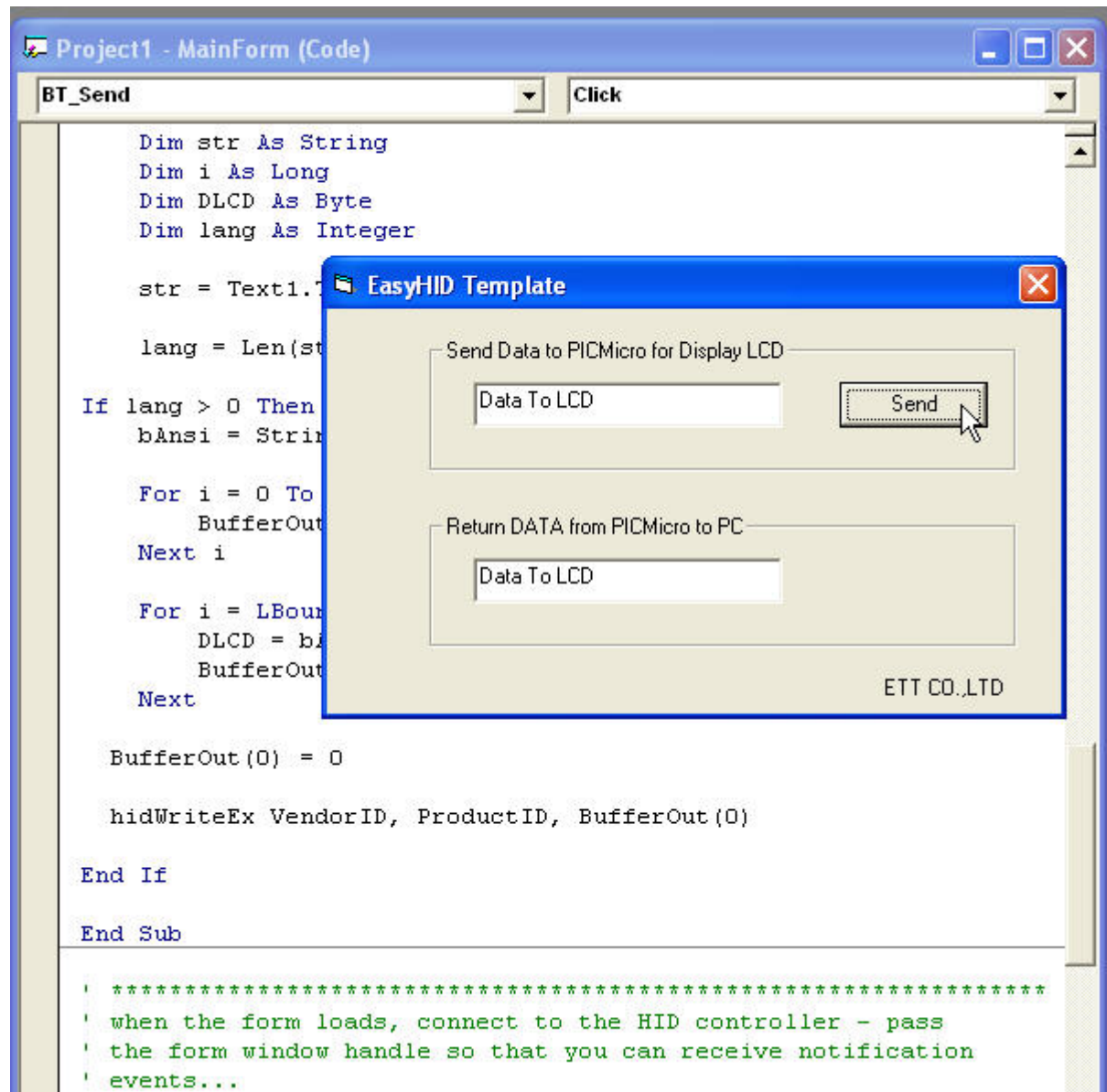
- โดยจะสร้าง Text Box ขึ้นมา 2 ช่อง และ ปุ่ม Button อีก 1 ปุ่ม ดังรูป



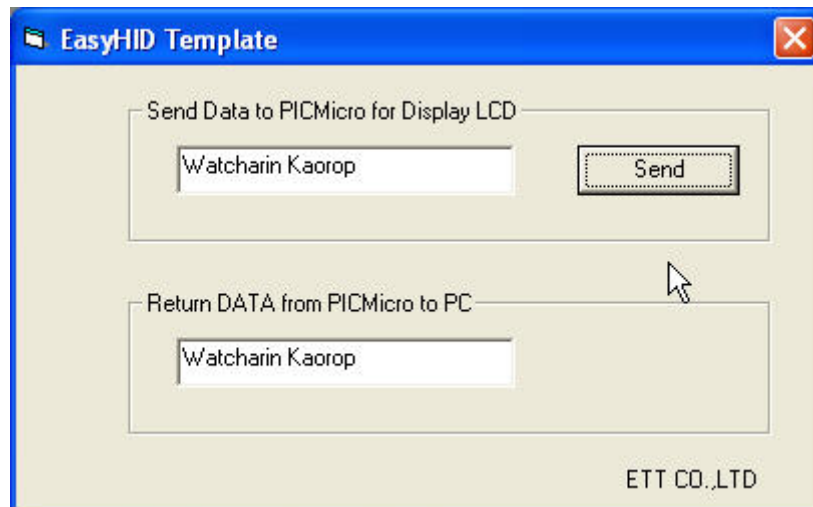
- เขียนโปรแกรม (ดูตัวอย่างเต็มๆ ได้จากตัวอย่างในโฟลเดอร์ USB4550_LCD)



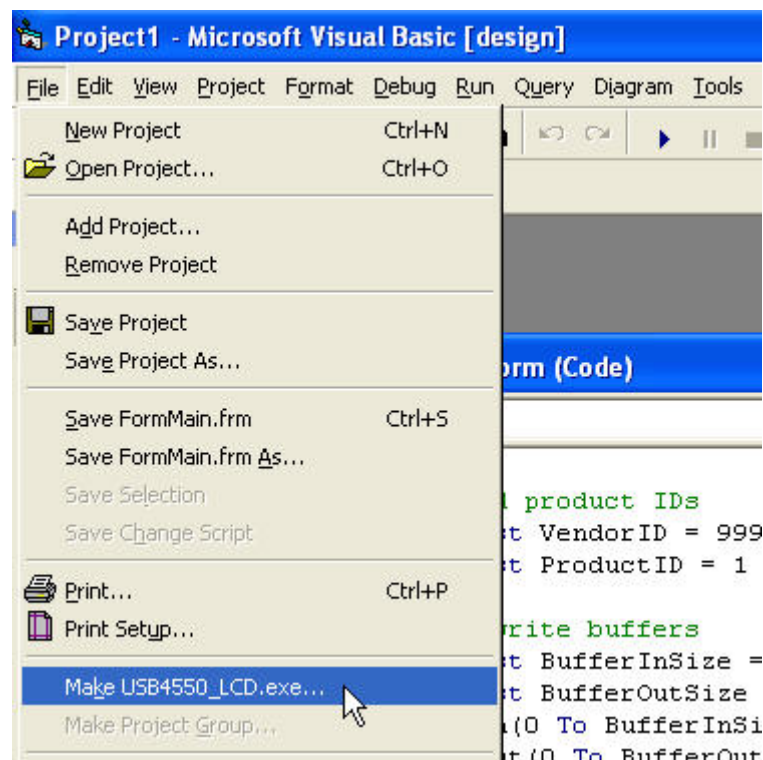
- ออกแบบเสร็จลองทดสอบโปรแกรมดู โดยจะต้องต่อ สาย USB เข้ากับบอร์ด ET-PIC USB/4550 ที่ได้โปรแกรมโค้ดก่อนหน้านี้แล้วด้วย



- ทดสอบส่งข้อความ



- เมื่อทดสอบแล้วเวิร์ก ก็ให้ทำการสร้าง Exe File ดังรูป



- ลองทดสอบ Exe File ที่สร้างขึ้น

Name	Size	Type
FormMain	10 KB	Visual Basic Form File
mchIDInterface	5 KB	Visual Basic Module
USB4550_LCD	1 KB	Visual Basic Project
USB4550_LCD	1 KB	Visual Basic Project ...
MSSCCPRJ	1 KB	Microsoft SourceSaf...
USB4550_LCD	32 KB	Application

EasyHID Template

Send Data to PICMicro for Display LCD

Data To LCD

Send

Return DATA from PICMicro to PC

ETT CO.,LTD