

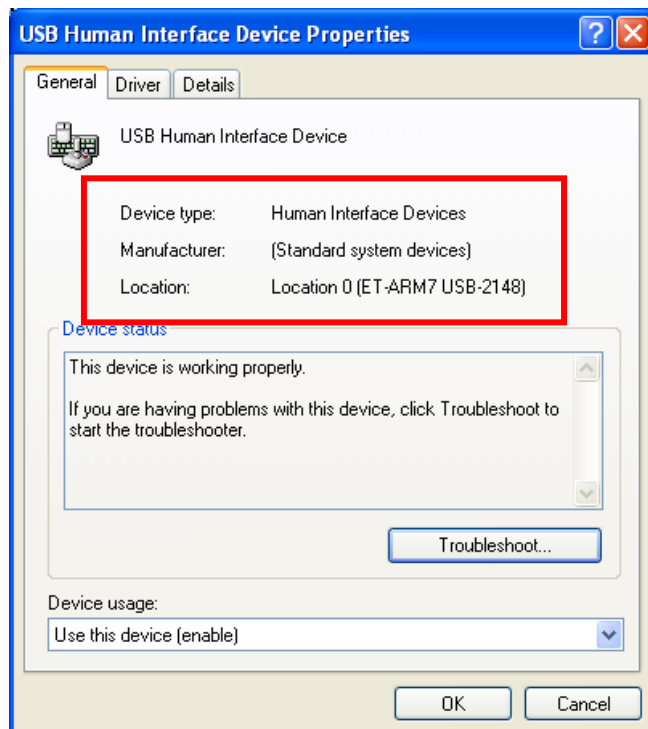
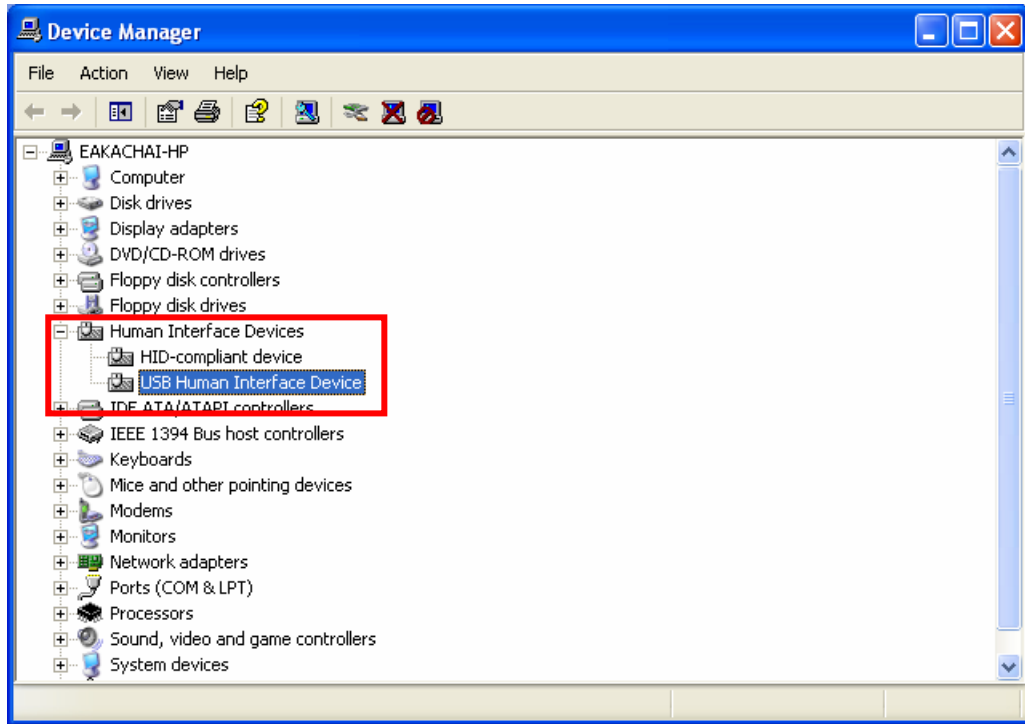
ตัวอย่างการใช้งาน USB ของ LPC2148 จาก อีทีที

สำหรับตัวอย่างการประยุกต์ใช้งาน USB ของ LPC2148 กับบอร์ด “CP-JR ARM7 USB-LPC2148” ของ อีทีที นั้น จะขอแนะนำเสนอเป็นตัวอย่างการนำ USB ไปใช้สื่อสารเพื่อควบคุมการทำงานของ I/O ภายในบอร์ด ในรูปแบบของ USB Control I/O โดยใช้ HID Class โดยในการทดสอบการทำงานให้เชื่อมต่อ Hardware ดังนี้

- 14 Bit Input Logic (Push Button Switch Active “0”)
 - ใช้ GPIO1[16..23] และ GPIO0[2..7] โดยอุปกรณ์ที่จะใช้ในการทดสอบ อาจใช้ Push Button Switch จำนวน 4 ชุด ที่มีอยู่ภายในบอร์ดเป็นตัวทดสอบ หรือจะใช้บอร์ด “ET-TEST 10P/INP” จำนวน 2 ชุด เป็นตัวทดสอบก็ได้
- 4 Channel Analog Input (10 Bit ADC = 0-3V3)
 - ใช้ GPIO1[25,28,29,30] โดยอุปกรณ์ที่จะใช้ในการทดสอบ สามารถใช้ Volume 1-4 ซึ่งมีอยู่ภายในบอร์ดเป็นอุปกรณ์ในการทดสอบ โดยการเชื่อมต่อสายจาก GPIO0[25,28,29,30] เข้ากับ Volume VOL[1..4] ตามลำดับ
- 15 Bit Output Logic (LED Control Active “0”)
 - ใช้ GPIO0[8..15] และ GPIO0[16..22] โดยอุปกรณ์ที่จะใช้ในการทดสอบ อาจใช้ LED จำนวน 4 ชุด ที่มีอยู่ภายในบอร์ดเป็นตัวทดสอบ หรือจะใช้บอร์ด “ET-TEST 10P/OUT” จำนวน 2 ชุด เป็นตัวทดสอบก็ได้
- 1 Bit Mini Speaker (Beep)
 - ใช้ GPIO1[24] โดยในการทดสอบให้ต่อสายจาก P1[24] เข้ากับ SPK1 เพื่อทดสอบ
- 1 Character LCD Display (16 x 2)
 - ใช้ GPIO1[25..31] โดยในการทดสอบให้เชื่อมต่อ LCD เข้ากับขั้ว CLCD ของบอร์ดได้ทันที
 - ถ้าไม่ต้องการเชื่อมต่อ LCD Display ในระหว่างการทดสอบการทำงาน ให้ต่อสัญญาณ D7 ของ LCD (GPIO1[31]) ลง GND ด้วย ไม่เช่นนั้นบอร์ดจะไม่สามารถทำงานได้

ซึ่งในการทดสอบการทำงานของบอร์ดให้ทำการ Download Hex File ชื่อ “ET_USB_HID_DEMO1.HEX” ให้กับบอร์ด ซึ่งหลังจากนั้นทำการรีเซ็ตบอร์ดพร้อมกับทำการเชื่อมต่อสายสัญญาณของ USB ระหว่างบอร์ด กับ USB-HUB ของคอมพิวเตอร์ PC โดยจะต้องเชื่อมต่อจอแสดงผล LCD ให้กับบอร์ดด้วย ถ้าไม่ต้องการเชื่อมต่อ LCD Display ในระหว่างการทดสอบการทำงาน ให้ต่อสัญญาณ D7 ของ LCD (GPIO1[31]) ลง GND ด้วย ไม่เช่นนั้นบอร์ดจะไม่สามารถทำงานได้ ซึ่งหลังจากทำการเชื่อมต่อสาย USB แล้ว ถ้าทุกอย่างถูกต้อง จะเห็น LED สีเขียวที่ใช

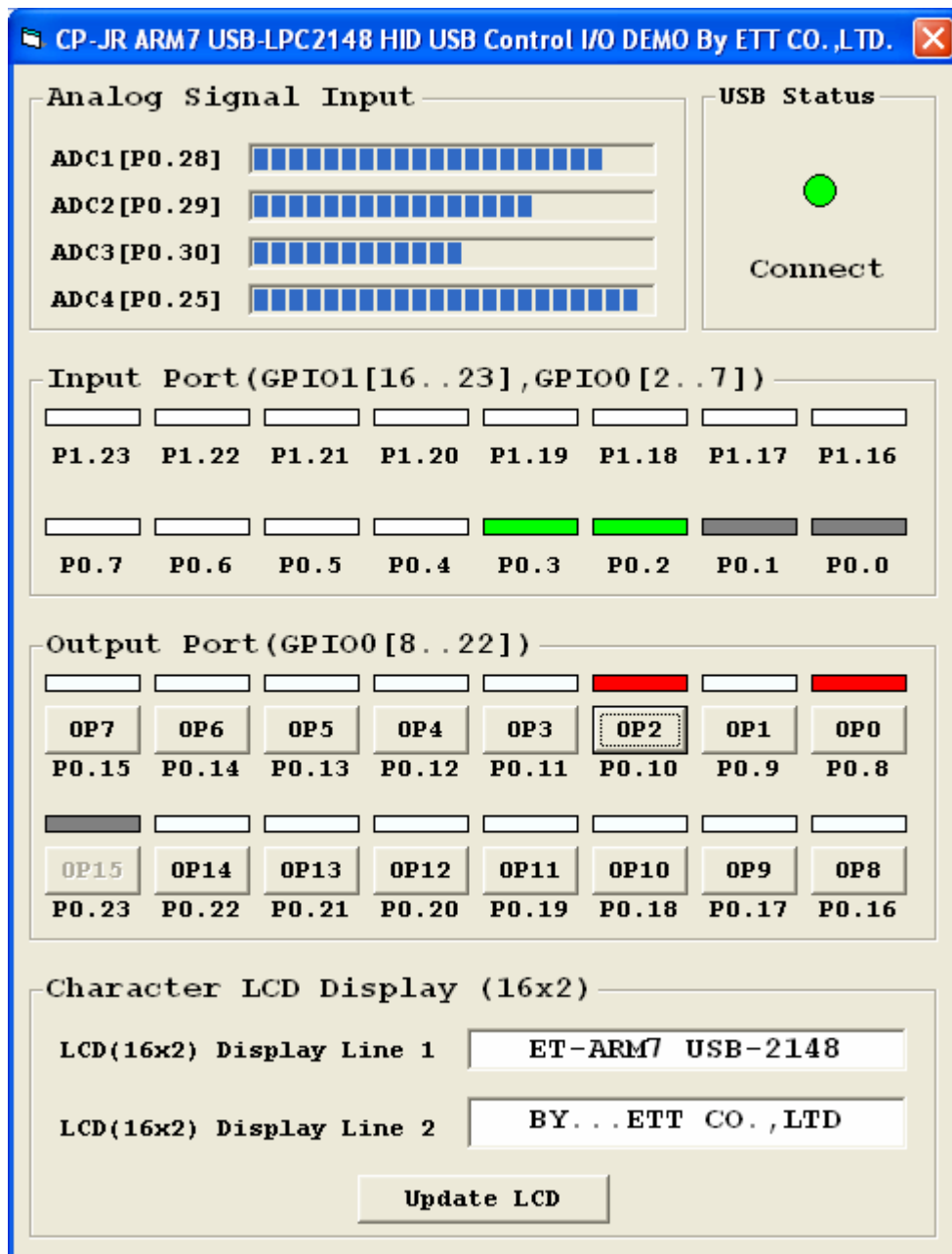
แสดงสถานะ LINK ของ USB ติดสว่างให้เห็น และ Windows จะตรวจพบการเชื่อมต่อกับ USB Device ซึ่งเป็นแบบ HID Class เพิ่มเข้ามาในเครื่องด้วย โดยผู้ใช้งานสามารถตรวจสอบผลการเชื่อมต่อได้จาก Device Manager โดยการคลิกเมาส์ไปที่ “Control Panel → System → Hardware → Device Manager” ดังรูป



*****หมายเหตุ***** ถ้าทำการเชื่อมต่อแล้ว เกิด Error เป็น “USB Device Not Recognized” ให้ตรวจสอบว่าได้ทำการเชื่อมต่อ จอแสดงผล LCD เข้ากับบอร์ดเรียบร้อยแล้วหรือไม่ ซึ่งถ้าไม่ต้องการใช้งาน LCD ในระหว่างการทดสอบให้ต่อสัญญาณ D7 (GPIO1[31]) ที่ขั้วต่อของ LCD ภายในบอร์ดเข้ากับ GND ของบอร์ดด้วย

การทดสอบการทำงานของบอร์ด

สำหรับการทดสอบการทำงานให้ใช้โปรแกรม “ET_HID_USB2148_DEMO_VB.EXE” เป็นตัวทดสอบการทำงาน ซึ่งเมื่อสั่ง Run โปรแกรมจะมีลักษณะดังนี้



รูปแสดงลักษณะของโปรแกรมที่ใช้ในการทดสอบการทำงานของ USB

การทำงานของโปรแกรมทดสอบ

1. เมื่อสั่ง Run โปรแกรม จะเกิดเสียง Beep ขึ้น 1 ครั้ง และหลอดไฟสีเขียวแสดงสถานะ ของการเชื่อมต่อของ USB กับคอมพิวเตอร์จะติดให้เห็นที่ช่อง USB Status
2. ผู้ใช้สามารถทดสอบการทำงานของ ADC ได้โดยการปรับ Volume VOL[1..4] เพื่อปรับค่าแรงดันป้อนให้กับ ADC เพื่อทดสอบการทำงาน โดยจะเห็นค่าใน Progress Bar เกิดการเปลี่ยนแปลงตามการปรับค่าแรงดันของ Volume ทั้ง 4
3. ผู้ใช้สามารถทดสอบการทำงานของ Input ทั้ง 14 Bit ได้โดยการทดสอบกด Switch ดู โดยจะเห็นสถานะของ Input เปลี่ยนเป็นสีเขียวเมื่อ Switch ถูกกด
4. ผู้ใช้สามารถทดสอบการทำงานของ Output ทั้ง 15 Bit ได้ โดยการ คลิกเมาส์ที่ปุ่ม Bottom OP0-14 โดยทุกๆครั้งที่ทำการคลิกเมาส์จะเกิดเสียง Beep ดังขึ้น 1 ครั้ง พร้อมกับสถานะของ LED จะถูกเปลี่ยนเป็นตรงข้าม
5. ผู้ใช้สามารถทดสอบการทำงานของ LCD ได้โดยการพิมพ์หรือแก้ไขข้อความในช่องรับข้อความ จำนวนช่องละ 16 ตัวอักษร เมื่อได้ข้อความที่ต้องการแล้วให้คลิกเมาส์ที่ปุ่ม Update LCD ซึ่งจะเห็นว่า ที่หน้าจอแสดงผลของ LCD จะแสดงข้อความตามที่ได้ป้อนไว้ในช่องรับข้อความทันที

รายละเอียดของตัวอย่างการทดสอบ

โปรแกรมของบอร์ด “CP-JR ARM7 USB-LPC2148”

สำหรับฟังก์ชันการทำงานของ USB ที่นำเสนอเป็นตัวอย่างการทดสอบของ USB นี้ ผู้เขียนได้นำแนวทางมาจากตัวอย่าง “LPC2148_USBHID” ของ Keil ซึ่งใช้กับชุดพัฒนาของ Keil รุ่น “MCB2140” และนำมาปรับปรุงการทำงานให้เหมาะสมและสอดคล้องกับระบบ Hardware ต่างๆที่มีอยู่ในบอร์ด “CP-JR ARM7 USB-LPC2148” โดยการทำงานของโปรแกรมสรุปได้ดังนี้คือ

- กำหนดการเชื่อมต่อ USB แบบ USB Link Mode และใช้พลังงานจาก Bus Power = 400mA
- กำหนดขนาด Packet ของ End Point เป็น 64 Byte
- กำหนด Class เป็นแบบ HID โดยใช้ Non Protocol มี 1 Interface (Interface Number 0)
- ใช้การส่งถ่ายข้อมูลแบบ Interrupt และกำหนด Scan Time เป็น 1mS
- กำหนดค่า Manufacture Name เป็น “ETT CO.,LTD.”
- กำหนดค่า Product Name เป็น “ET-ARM7 USB-2148”
- กำหนดค่า Product Serial Number เป็น “HID DEMO V10”
- กำหนดใช้งาน End Point-0 สำหรับ Setup และ รับข้อมูล และใช้ End Point-1 สำหรับส่งข้อมูล
- การรับส่งข้อมูลและคำสั่งจะใช้ HID Report โดยมีการกำหนดรูปแบบของ Report ไว้ดังนี้
 - Input Report กำหนดให้ มีขนาด 10 Byte / 8 Bit (ไม่รวม Report ID)
 - Byte-0 เป็นค่าของ Input[0..7] ซึ่งได้จาก GPIO1[16..23]
 - Byte-1 เป็นค่าของ Input[10..15] ซึ่งได้จาก GPIO0[2..7] (Bit0,1 = Ignore)
 - Byte-2,3 เป็นค่าของ ADC1 Result Byte High, Low ตามลำดับ
 - Byte-4,5 เป็นค่าของ ADC1 Result Byte High, Low ตามลำดับ
 - Byte-6,7 เป็นค่าของ ADC1 Result Byte High, Low ตามลำดับ
 - Byte-8,9 เป็นค่าของ ADC1 Result Byte High, Low ตามลำดับ
 - Output Report กำหนดให้ มีขนาด 35 Byte / 8 Bit (ไม่รวม Report ID)
 - Byte-0 เป็นค่าของรหัสควบคุม โดยกำหนดให้
 - ❖ 0x00 หมายถึงการ Update Output ทั้งหมด (LED + LCD)
 - ❖ 0x01 หมายถึงการ Update Output LED เพียงอย่างเดียว
 - ❖ 0x02 หมายถึงการ Update LCD Display เพียงอย่างเดียว
 - Byte-1 เป็นค่าของ Output[0..7] ใช้ควบคุม GPIO0[8..15]
 - Byte-2 เป็นค่าของ Output[8..14] ใช้ควบคุม GPIO0[16..22] (Bit7=Ignore)
 - Byte-2..34 เป็นข้อความแสดงผลของ LCD จำนวน 32 ตัวอักษร

โปรแกรมของ Host (Visual Basic V6.0)

สำหรับการพัฒนาโปรแกรมทางด้านของ Host หรือ Computer PC นั้น จากตัวอย่างที่นำเสนอจะพัฒนาด้วยโปรแกรม Microsoft Visual Basic 6.0 โดยผู้เขียนได้นำแนวทางมาจากตัวอย่างใน Website ของ Intel โดยสามารถดูรายละเอียดได้ที่ <http://www.intel.com/intelpress/usb/examples/download.htm> ซึ่งเป็นตัวอย่างของการเขียนโปรแกรมเชื่อมต่อส่งงานกับ USB Device ใน Class ของ HID โดยเรียกใช้ Driver มาตรฐานของ Windows ผ่านทาง API Function และ DLL โดยสามารถใช้งานได้กับระบบปฏิบัติการ Windows98SE ขึ้นไป ซึ่งผู้เขียนได้ลองทดสอบใช้งานดูแล้วในระดับหนึ่ง ก็เห็นว่าสามารถใช้งานได้ดีพอสมควร ฟังก์ชันการทำงานต่างๆก็ใช้งานได้ง่ายไม่ซับซ้อน สามารถทำความเข้าใจได้ง่าย

โดยทาง Intel ได้สร้าง Code แยกเป็น Library Module ของ Visual Basic จัดเตรียมไว้สำหรับให้ผู้พัฒนานำไปผนวกเข้ากับ Code ของโปรแกรมหลักที่จะเขียนขึ้นได้โดยง่าย ซึ่งมีทั้งหมด 3 ไฟล์ด้วยกันคือ

- OSInterface.BAS เป็นส่วนที่ใช้สำหรับติดต่อสื่อสารกับระบบปฏิบัติการของ Windows ผ่านทาง API Function ซึ่งเป็นส่วนของการประกาศเรียกใช้ฟังก์ชันต่างๆใน DLL ไฟล์
- USBInterface.BAS เป็นส่วนของการประกาศตัวแปร สำหรับใช้ในการสื่อสารกับ USB Protocol ผ่านทางฟังก์ชันต่างๆใน API Function
- HIDInterface.BAS เป็นส่วนของฟังก์ชันที่ใช้สำหรับติดต่อสื่อสารกับ USB Device ต่างๆ ซึ่งมีส่วนของฟังก์ชันและโปรแกรมย่อยที่จำเป็นต้องนำมาใช้งาน ได้แก่
 - Function OpenUSBdevice สำหรับใช้เปิดการเชื่อมต่อกับ USB Device โดยในการเรียกใช้งานนั้น ต้องทำการผ่านค่า String ของ Product Name ของ USB Device ที่ต้องการเปิดการเชื่อมต่อให้กับฟังก์ชัน โดยถ้าสามารถเปิดการเชื่อมต่อกับ Device ได้จะได้ผลลัพธ์เป็นจริง แต่ถ้าไม่สามารถเชื่อมต่อได้จะได้ผลลัพธ์เป็นเท็จ
 - Sub ReadUSBdevice สำหรับสั่งอ่านข้อมูลจาก USB Device ที่เปิดการเชื่อมต่อไว้แล้ว โดยในการใช้งานต้องผ่านค่า Pointer ของตัวแปรแบบ Array ที่จะใช้ในการรับค่า และค่าจำนวนไบต์ที่ต้องการอ่านข้อมูลจาก Device ให้กับ โปรแกรมย่อย
 - Sub WriteUSBdevice สำหรับสั่งเขียนข้อมูลให้ USB Device ที่เปิดการเชื่อมต่อไว้แล้ว โดยในการใช้งานต้องผ่านค่า Pointer ของตัวแปรแบบ Array ที่เก็บค่าของข้อมูลที่จะส่งหรือเขียนไปยัง Device และ ค่าของจำนวนไบต์ข้อมูลที่ต้องการจะเขียนไปยัง Device ให้กับ โปรแกรมย่อย

โดยวิธีการใช้งานนั้นเมื่อผู้ทำการสร้าง Project File ของ Visual Basic เป็นที่เรียบร้อยแล้วให้ทำการสั่งผนวกไฟล์ทั้ง 3 นี้เข้ากับ Project File ที่สร้างขึ้น โดยใช้คำสั่ง “Project Add → Module → Existing” แล้วเลือกไฟล์ดังกล่าวข้างต้นทีละไฟล์จนครบ โดยรายละเอียดการทำงานของตัวอย่างโปรแกรมที่ใช้ในการทดสอบนั้น ขอให้ผู้ใช้ศึกษาและทำความเข้าใจจาก Source Code ตัวอย่างเอง