

USB Mouse tutorial

Step-by-step project creation

CAK 2005 Petr Stružka UNIS, spol. s r.o. PStruzka@unis.cz



Application Specification

Application simulates mouse device connected through USB

- Mouse simulation
 - Six buttons are used on HC908JB8 demo board
 - 4 buttons give direction
 - 2 buttons (L/R)
- Messages are sent to the computer via USB port
 - Buttons status
 - Mouse position (changed via buttons)
- No application required on computer
 - Demo board is detected as standard USB mouse device

Copyright © 2005 UNIS spol. s r.o.

Application Structure 1

- In depth knowledge of USB technology required
- Application code required:
 - Initialization
 - CPU initialization
 - USB peripheral initialization
 - Create data structures
 - Standard Descriptors
 - Device, configuration, interface, endpoint,
 - Manufacturer strings, IDs, etc.
 - Class Descriptors
 - HID class
 - Report Descriptor
 - USB driver
 - Standard USB request handler
 - Handled using states
 - Set parameters

Copyright © 2005 UNIS spol. s r.o.

Application Structure 2

- HID class compliant driver
 - HID class implementation
 - Data Buffer specified by Report Descriptor
 - Data Buffer declaration
 - Response to requests (from PC side)
 - HID Class request handler
 - Specification available on <u>URL://http.usb.org</u>
- Application
 - Handle buttons' interrupts
 - Set buttons state
 - Set position difference
 - Update Report Descriptor's data on buttons' interrupts
 - Send report

Application Structure with PE 1

- Initialization
 - CPU initialization provided by CPU bean
 - USB bean initialize periphery
 - KBI bean initializes buttons
- USB driver
 - Encapsulated by USB bean
 - USB bean properties define
 - Standard Descriptors
 - Parameters (manufacturer strings, IDs, etc.)
 - HID Class Descriptor
 - USB bean driver provides all code to handle USB requests



USB Mouse Tutorial

Application Structure with PE 2

- USB HID Class
 - USB_HID_Class bean
 - SW bean only
 - Handles class requests
 - Provides method for setting data in reports
- Application (in main code)
 - Buttons' state (provided by KBI bean)
 - Update Report Descriptor's data on buttons' events
 - Set buttons state
 - Set position difference



USB Mouse Tutorial

Tutorial Requirements

Hardware

- UNIS USB Demo Board
- Mon08 Multilink
- cables
- Software
 - CodeWarrior with PE
 - USB bean update
 - pe 2 95 03 USB hc08 hcs08.PEupd



USB Mouse Tutorial

 Create a new project with JB8 (set name as USBMouse) using Project Wizard

Empty Project	Project name:	
HC(S)08 Board Support Stationery	USBMouse	
E HC(S)06 New Project Wizard	Location:	
	G:\USBMouse	iet
	Add to Project:	
	Project:	-16
		7
	1	

USB Mouse Tutorial

UNIS spol. s r.o.



Copyright © 2005 UNIS spol. s r.o.



USB Mouse Tutorial

Ciono.	Would you like to use Processor Expert?	
	© No Processor Expert can generate for you all the device initialization code. It includes many low-level drivers.	
The House Information to the World of Sultern	< <u>Z</u> pět <u>D</u> alší > Storno	

USB Mouse Tutorial

ew Project Wizard - Page 4	
Le and	Do you want to create a project set up for PC-lint(TM)?
an orrestore	C Yes
6	No
	Lint tools can find common programming mistakes or suspicious lines in source code by analyzing it. PC-lint(TM) is a product from Gimpel
metrowerks	
	< <u>Z</u> pět <u>D</u> alší > Storn

UNÜ

USB Mouse Tutorial

1º inton	
Contraction .	Which level of startup code do you want to use? Select 'minimal startup code' for best code density.
101010101010	C minimal startup code
	 ANSI startup code This will perform an ANSI compliant startup code: it initializes global variables/objects and
metrowerks	calls the application main routine.



New Project Wizard - Page 8	Please choose the connections you want. You can
	■P&E Full Chip Simulation P&E Hardware Debugging MMDS-MMEVS FSICEBASE Emulator Hitex Lauterbach SofTec Microsystems
metrowerks	P&E Full Chip Simulation with simulation of all on-chip peripherals. Later on in the debugger you can switch to hardware debugging.
<u></u>	≺ <u>Z</u> pět <u>D</u> okončit Storn

<u>Unis</u>

×

Newly created project:



Copyright © 2005 UNIS spol. s r.o.

 Select Configuration
 Debug_908JB8_28 in the Processor Expert tab.

(This is the right configuration for MCU installed on the development board – another configuration can be deleted)



 Select Categories page in Bean selector. Expand and double click on SW\Communication\USB_Mouse[USB_HID_Class] bean.



The bean will be added to current project. In the following dialog check "Do not ask again" checkbox and click on Yes button.





The following dialog appears – click to OK button

Processor Expert helps you to select or create a shared bean	×
The bean "USB_Mouse[USB_HID_Class]" uses shared bean for item "Parent USB Bean". Currently you have no appropriate bean in your project, you can add new one just now.	
Add new shared bean: USB	
✓ OK X Cancel ? H	elp

 HID1 and USB1 beans will be added to the project.



Select KBI Bean from Bean selector

Sean Selector	
Categories On-Chip Prphrls Alphabet Keywords	Quick help >
E 🕞 CPU	
🖅 🗁 CPU External Devices	
🖃 🗁 CPU Internal Peripherals	
🗄 🗁 Communication	
🕀 🗁 Converter	
🕀 🗁 Display	
🗇 🗁 Interrupts	
🗕 🗧 🌍 Extint	
🚽 🗕 🔒 🛞 InterruptVector	
🕀 🗁 Measurement	
🕀 🗁 Memory	
🕀 🗁 Peripheral Initialization Beans	
🕀 🗁 Port I/O	
Filter: all/CPU Licensed	11

 KBI bean will be added to your project

P&E ICD Processor Expert Files Link Order Targets Processor Expert Configurations Pelease_908JB8_20 Pelease_908JB8_20 Pelease_908JB8_24 Pelease_908JB8_28 Pelease_908JB8_28 Pelease_008JB8_28 Pelease_908JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_008JB8_28 Pelease_002JB8_28 Pelease_008JB8_28 Pelease_002JB8_28 Pelease_008JB8_28 Pelease_002JB8_28 Pelease_008JB8_28 Pelease_002JB8_28 Pelease_008JB8_28 Pelease_002JB8_28 Pelee	P&E ICD Processor Expert Files Link Order Targets Processor Expert Configurations A Release_908/B8_20 A Release_908/B8_20 A Release_908/B8_20 A Release_908/B8_20 B Release_908/B8_24 A Release_908/B8_28 Debug_908/B8_28 B Release_908/B8_28 B Release_908/B8_28 Debug_908/B8_28 Release_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Debug_908/B8_28 Cpu:MC68HC908/B8/P DeprimC68HC908/B8/P D	USBMouse.mcp		
Files Link Order Targets Processor Expert Configurations X Release_908JB8_20 X Debug_908JB8_20 X Release_908JB8_24 X Debug_908JB8_28 X Release_908JB8_28 X Debug_908JB8_28 X Debug_908JB8H X Debug_908JB8H X Cpu:MC68HC908JB8EP X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB Y USBHOuse.c:main Y USBMouse.c:main	Files Link Order Targets Processor Expert Configurations X Release_908JB8_20 X Debug_908JB8_20 X Release_908JB8_24 X Debug_908JB8_24 X Release_908JB8_28 X Debug_908JB8_28 X Debug_908JB8H X Cpu:MC68HC908JB8EP X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB X Cpu:MC68HC908JB8FB Y USBMouse.c:main X USBMouse.c:ma	🔞 P&E ICD	💽 🕼 😽 🧐 🦉	. • 8
Configurations Release_908/B8_20 Release_908/B8_20 Release_908/B8_24 Release_908/B8_24 Release_908/B8_28 Debug_908/B8_28 Pebug_908/B8_28 Pebug_908/B8_28 Pebug_908/B8_28 Pebug_908/B8_28 Pebug_908/B8_44 Debug_908/B8_44 Debug_908/B8_44 Pebug_908/B8_44 Pebug_908	Configurations Release_908JB8_20 Release_908JB8_20 Release_908JB8_21 Release_908JB8_24 Release_908JB8_28 Pebug_908JB8_28 Pebug_908JB8_28 Pebug_908JB8_28 Pebug_908JB8_28 Pebug_908JB8_44 Pebug_908JB84DW Pebug_908	Files Link Order Target	s Processor Expert	
 ✓ Image Events.c:event Generated Modules	 ∠ ≥ Generated Modules → External Modules → Documentation 	Files Link Order Target: Configurations * Release_908JB8 Debug_908JB8 Release_908JB8 Release_908JB8 Release_908JB8 Release_008JB8 Release_008JB8 Release_008JB8 Release_008JB8 Release_008JB8 Release_008JB8 Release_008JB8	S Processor Expert S Processor Expert S S 20 20 3_24 24 3_28 28 3_44 44 BB88IP BB84P BB84P BB84DW	
			in	
		- Cocumentation		



Copyright © 2005 UNIS spol. s r.o.

<u>Un</u>

Switch Items Visibility from Basic view to the Expert view

🗞 Bean Inspector KB1:KBI	State of the local division of the local div	
Bean Items Visibility Help <	>	Peripheral Initialization >
Properties Methods Events	Comment	-
KBI	KBI	✓ KBI
🚦 Pins	1	+ -
L Pin0 L Pin		Unassigned p
 Pull resistor 	autoselected pull	no pull resistor
 ☐ Interrupt service/event ☐ Initialization 	Enabled	Q
Lev Enable in init. code	yes	Q
BASIC ADVANCED EXF	ERT Bean Level:	Low Level Bean 🥢

ean Items Visibility Help <	>	Peripheral	Initialization >
Properties Methods Events	Comment 1		
Bean name	KB1		<u> </u>
KBI	KBI	•	KBI
Pins	1	+ -	
Pin0			
Pin Pin		•	Unassigned pa
L Pin signal			
Pull resistor	autoselected pull	•	no pull resistor
Generate interrupt on	rising or falling edge	•	falling edge
Interrupt service/event	Enabled	0	
 Interrupt 	INT_KBI		INT_KBI
 Interrupt priority 	medium priority	+	not supported
Initialization			
 Enable in init. code 	yes	Ð	
 Events enabled in init. 	yes	0	

USB Mouse Tutorial

 Set KBI bean settings as shown on screenshot below accurately.

> (Note, pins order in the pin list, Pins values and Pin signal name values are important)

3ean ItemsVisibility Help <	>	Peripheral Initialization
Properties Methods Events	Comment	
/ Bean name	KB1	
KBI	KBI	✓ KBI
Pins	6	<u>+-</u>
🖂 Pin0		
- Pin	PTA3_KBA3	✓ PTA3_KBA3
└ ✔ Pin signal	Left_Button	
Pin1	- Chinese	
- Pin	PTA1_KBA1	✓ PTA1_KBA1
- ← Pin signal	Right_Button	
🖃 Pin2		
Pin Pin	PTA4_KBA4	✓ PTA4_KBA4
_ └ ← Pin signal	Up	· · · · · · · · · · · · · · · · · · ·
🖂 Pin3	- <u> </u>	
Pin	PTA5_KBA5	PTA5_KBA5
_└ ✔ Pin signal	Down	
🖻 Pin4	- Carrieren	
Pin	PTA7_KBA7	✓ PTA7_KBA7
-< Pin signal	Left	
🖂 Pin5		
- Pin	PTA6_KBA6	PTA6_KBA6
└ ✔ Pin signal	Right	
Pull resistor	pull up	🛨 pull up
Generate interrupt on	rising or falling edge	\star falling edge
Interrupt service/event	Disabled	2
Initialization		
🖌 Enable in init. code	yes	2
 Events enabled in init. 	yes	



Generate project

 Select Processor Expert menu – Generate Code 'USBMouse.mcp'

Project: USB	Mouse	
Module:		
Preparing for (code gener. Total	l lines: 1
	hulaminae: 0	Hints: 0

- When successfully finished
 - Browse the generated code
- Open <u>USBMouse.c</u> file this source contains main routine



Write application code 1

Write the main routine loop

- Get button status
- Initialize input report
- Update report
- Send report



Write application code 2

Copy prepared main function code to the main function.

Original main() function body generated by Processor Expert:



USB Mouse Tutorial

Write application code 3

Predefined main()

SBMouse.c	2
🔥 🔸 🚺 🔸 🛍 🔸 🖆 🔸 Path: E:\Metrowerks\Projects\USB\USBMouse\CODE\USBMouse.c	
void main(void) {	
<pre>#define Increment 1 byte KB_Data; HID1_DefineInputReport(1,Report_1)</pre>	
<pre>/*** Processor Expert internal initialization. DON'T REMOVE THIS PE_low_level_init(); /*** End of Processor Expert internal initialization.</pre>	CODE!!! ***/ ***/
<pre>for(;;) { KB_Data=KB1_GetVal(); KB_Data=~KB_Data;</pre>	
<pre>HID1_InitInputReport(1,Report_1);</pre>	
HID1_SetReport(Report_1,Button_1,(KB_Data & Left_Button)>>1)); HID1_SetReport(Report_1,Button_2,KB_Data & Right_Button);	
<pre>if (KB_Data & Left) HID1_SetReport(Report_1,X,-Increment); if (KB_Data & Right) HID1_SetReport(Report_1,X,+Increment); /* Increment X posi /* Increme</pre>	sed ? */ tion counter */ ssed ? */ tion counter */
II (KE_Data & Down) /* Down Dutton press HID1_SetReport(Report_1,Y,+Increment); /* Increment Y posi if (KE_Data & Up) /* Up button presse HID1_SetReport(Report 1 Y -Increment); /* Decrement Y posi	sed (*/ tion counter */ d ? */ tion counter */
<pre>while(HID1_SendReport(1, Report_1, TRUE, FALSE)!=ERR_OK); /* Send</pre>	report */
3	
<pre>/*** Processor Expert end of main routine. DON'T MODIFY THIS CODE for(;;){}</pre>	*** /
/*** Processor Expert end of main routine. DON'T WRITE CODE BELOW } /*** End of main routine. DO NOT MODIFY THIS TEXT!!! ***/	*** /
.ine 57 Col 74 [] ◀]	

Copyright © 2005 UNIS spol. s r.o.

Test the Application

- Compile and start debug application
 - Click the Make icon to proceed build.
 - Click the Debug icon to start the debugger.
 - Run the application
- Connect JB8 board to computer's USB port









Demo board settings

Make sure the Demo board is configured properly

- JP4 (power source selection) 3-4
- JP1, JP2 debug interface selection both open

For more details see **Demo board manual**

USB Mouse Tutorial

When the debugger started first time, the following dialog window appears:

Attempting to contact target and pass security		
Class 1 - Motorola ICS Board with processor installed. Emulation	connection ok. (Power controlled via DTR)	▼ <u>A</u> dvanced
-Class I, II, III, IV - Settings		
Serial Port:	Close Port	
Baud: 9600 Baud		
Specified Baud: 0		
The port is open		
The target should have bypass canacitors on Vdd and Beset	+ Rage	mmandad
	Nett	minended
Target MCU Security bytes		
C Attempt ALL Known security codes in order	C Attempt FF-FF-FF-FF-FF-FF-FF {From security.ini}	
O Attempt FF-FF-FF-FF-FF-FF-FF (Blank Device)	C Attempt FF-FF-FF-FF-FF-FF-FF {From security.ini}	
● Attempt FF-FF-FF-FF-FF-FF-FF (From security.ini) (Recent)	C Attempt 00-00-00-00-00-00-00 (Blank on older devices)	
Attempt FF-FF-FF-FF-FF-FF (From security.ini)	C User: 00-00-00-00-00-00-00	Load from S19
C Attempt FF-FF-FF-FF-FF-FF-FF {From security.ini}	GNORE security failure and enter moni	itor mode.
Status: Invalid Response or No Response to last attemp	ot to contact target.	
0. Hardware loopback detected: N 4. Device	entered monitor mode: N	
1. Device echoed some security bytes: N (Port?) 5. Reset w	Jas Power-On Reset:	Help
2. Device echoed all security bytes: N 6. ROM is	accesible (un-secured):	
Device signaled monitor mode with a break: N		
Show this dialog before attempting to contact the target 68H0	208 board	
Contact target with these settings		Abort
<u>contact target with these settings</u>		Aport

Copyright © 2005 UNIS spol. s r.o.

- Set Target Hardware Type to Class 7 P&E MON08 Multilink
- Set Port to the LPTxx according to the HW connection (or USB)
- Set Clock Devider to 4
- Set "IGNORE security failure and enter monitor mode" checkbox in "Target MCU Security bytes" group (this settings have to be done only if current security bytes of the MCU are not known)
- Click on "Contact target with these settings..." button



 When all described properties are set the window should look like

Class 7 - P&E MON08 MULTILINK or USB MON08 MULTILIN	IK connected to target via ribbon cable.
Class 5, 7, 8 - CYCLONE, CYCLONE PRO, and MON08 MULT Port: LPT1 - Parallel Port 1 (Address \$0378) Device Type: JB1/8 Device Power: 5 Volts, Provided by P&E Interface Device Clock: Target self-clocked, P&E Output Clock Disable Clock Divider: 4 The target should have bypass capacitors on Vdd and Reset.	ILINK Settings 1 2 Close Port Pinout: NC \circ GND Pinout: NC \circ RESET GND* \circ IRQ = HighV Image: Second Secon
Target MCU Security bytes C Attempt ALL Known security codes in order Attempt FF-FF-FF-FF-FF-FF-FF (Blank Device) Attempt E4-CB-DC-00-E2-06-DC-04 (From security.ini) (Rec Attempt FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-FF-	Attempt FF-FF-FF-FF-FF-FF-FF-FF (From security.ini) Attempt FF-FF-FF-FF-FF-FF-FF-FF (From security.ini) ent) Attempt 00-00-00-00-00-00-00 (Blank on older devices) User: 00-00-00-00-00-00-00 Load from S19
Status: Invalid Response or No Response to last atter 0. Hardware loopback detected: N 4. Device 1. Device echoed some security bytes: N 5. Reset 2. Device echoed all security bytes: N 6. ROM 3. Device signaled monitor mode with a break: N Image: Show this dialog before attempting to contact the target 68	mpt to contact target. e entered monitor mode: N was Power-On Reset: is accesible (un-secured): HC08 board
Contact target with these settings	Abort

Copyright © 2005 UNIS spol. s r.o.

Click on Yes button to erase and program MCU







Test the application

 Click the buttons on board and use it for moving the mouse cursor on PC's screen



Additional tips

How to prepare clear instalation of USBMouse application

- Run regedt32.exe from Start/command menu
- Set full access rigths of the current account for the following key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB
- Delete following key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB\Vid_xxxx&Pid _yyyy where xxxx is vendor ID and yyy is prouuct ID specified in the USB Bean

Comparison of Approaches

- Traditional
 - Deep HW knowledge required
 - USB knowledge required
 - Need to write driver (or re-use library) ... is it available?
 - More time for development
 - Application code
 - Full control over application code

- Processor Expert
 - Less HW knowledge required
 - Minimal USB knowledge required
 - No MCU drivers writing
 - Portable code
 - Less time to final product!
 - Application code