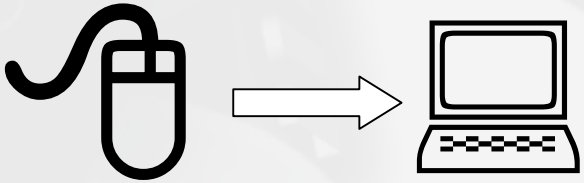
USB Mouse tutorial

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.

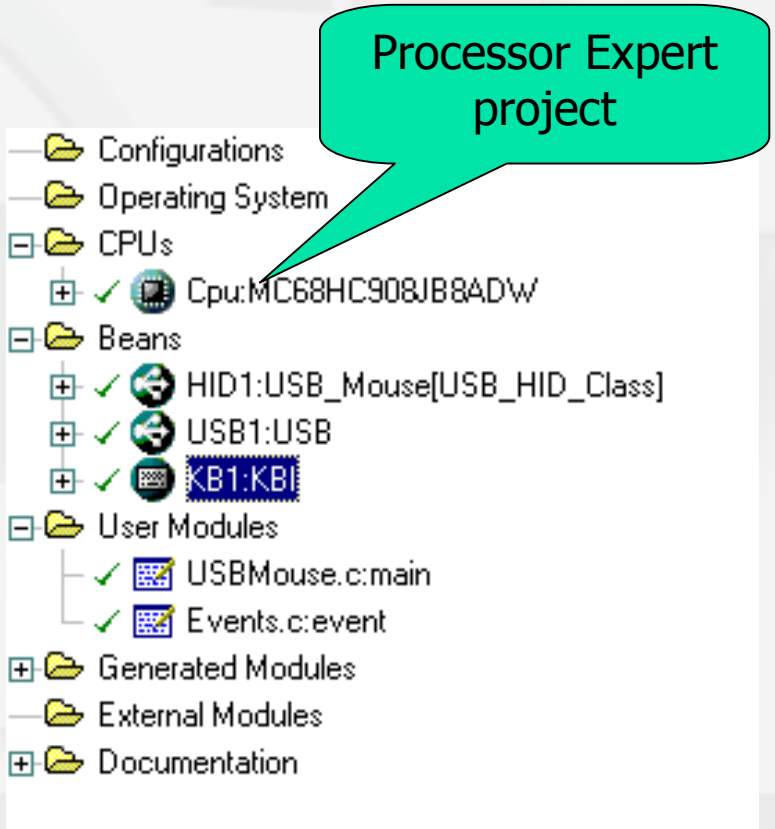
Application Structure 2

- 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 URL://http.usb.org
- Application
  - Handle buttons' interrupts
    - Set buttons state
    - Set position difference
  - Update Report Descriptor's data on buttons' interrupts
  - Send report

Copyright © 2005
UNIS spol. s r.o.

Application Structure with PE 1

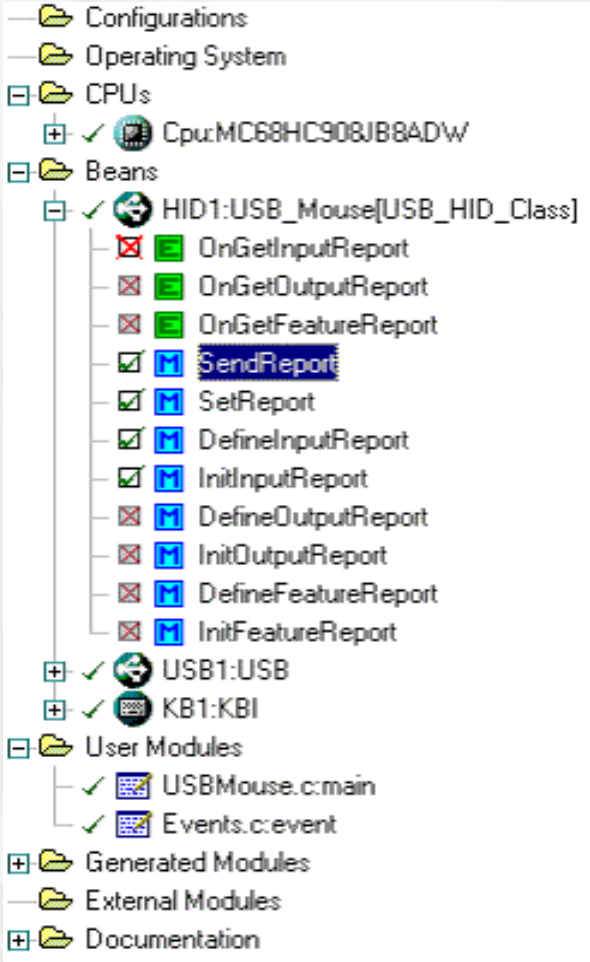
- Initialization
  - CPU initialization provided by CPU bean
  - USB bean initialize periphery
  - KBI bean initializes buttons
- USB driver
  - Encapsulated by USB bean
  - USB bean properties define
    - Standard Descriptors
      - Parameters (manufacturer strings, IDs, etc.)
    - HID Class Descriptor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



Copyright © 2005
UNIS spol. s r.o.

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



Copyright © 2005
UNIS spol. s r.o.

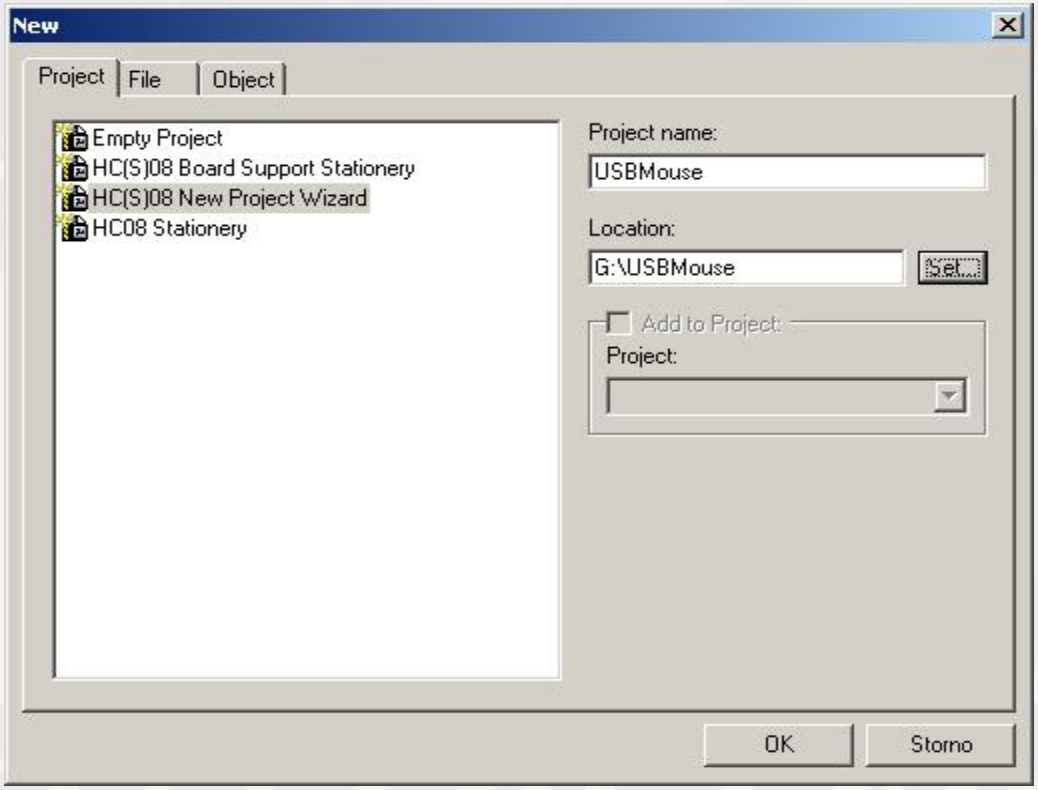
Tutorial Requirements

- Hardware
  - UNIS USB Demo Board
  - Mon08 Multilink
  - cables
- Software
  - CodeWarrior with PE
  - USB bean update
    - pe_2_95_03_USB_hc08_hcs08.PEupd

Copyright © 2005
UNIS spol. s r.o.

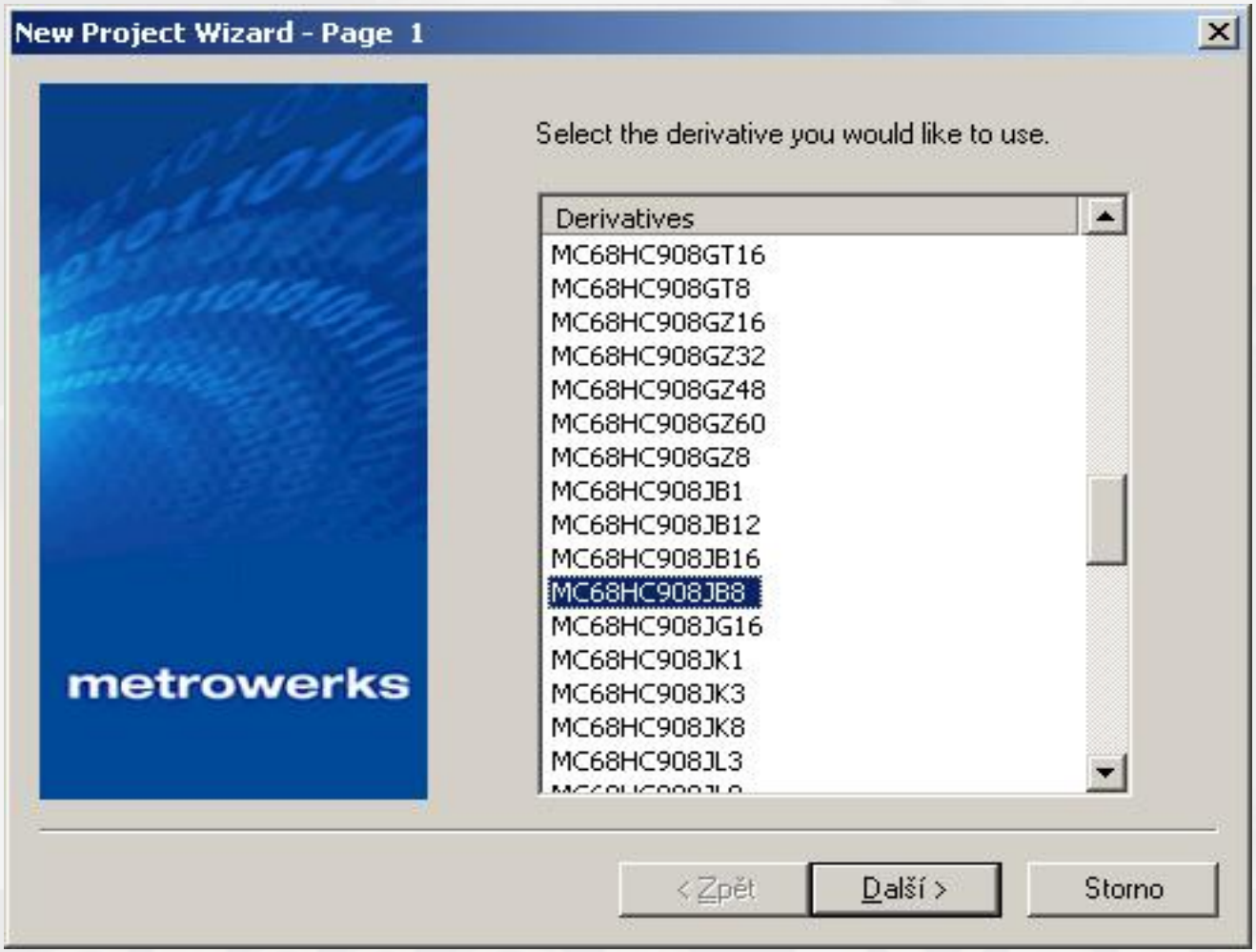
Create a New Project 1

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



Copyright © 2005
UNIS spol. s r.o.

Create a New Project 2

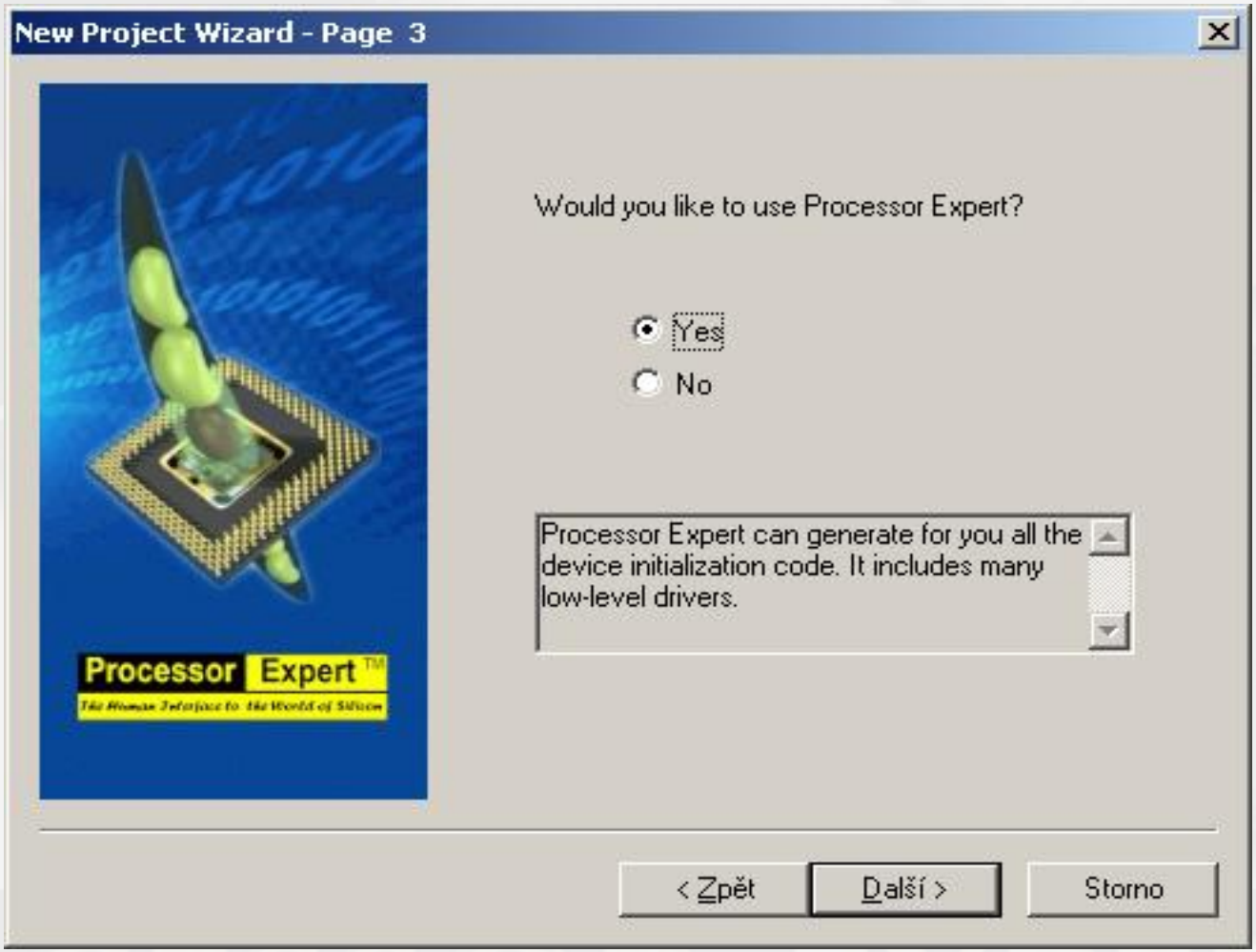


Copyright © 2005
UNIS spol. s r.o.

Create a New Project 3

Copyright © 2005
UNIS spol. s r.o.

Create a New Project 4

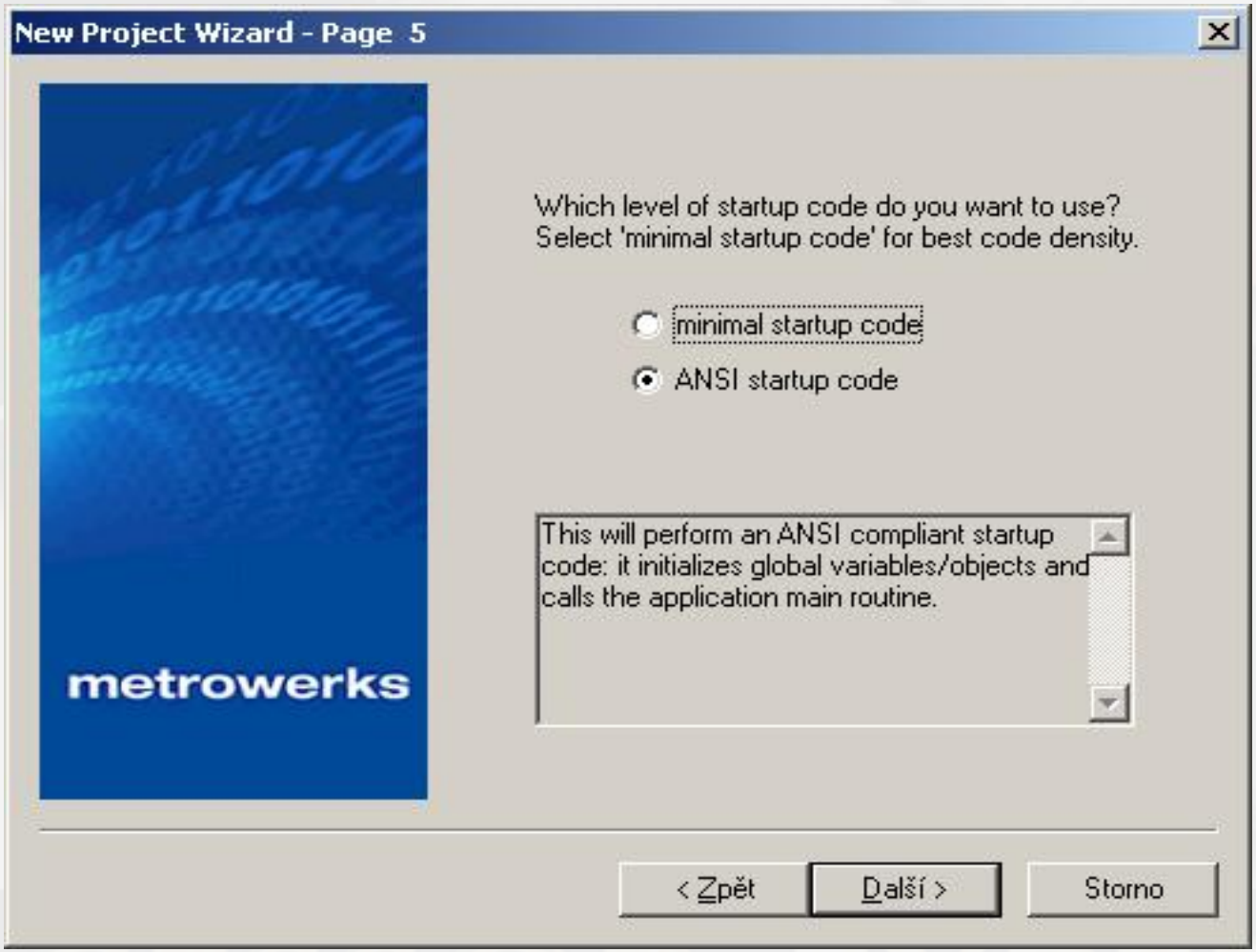


Copyright © 2005
UNIS spol. s r.o.

Create a New Project 5

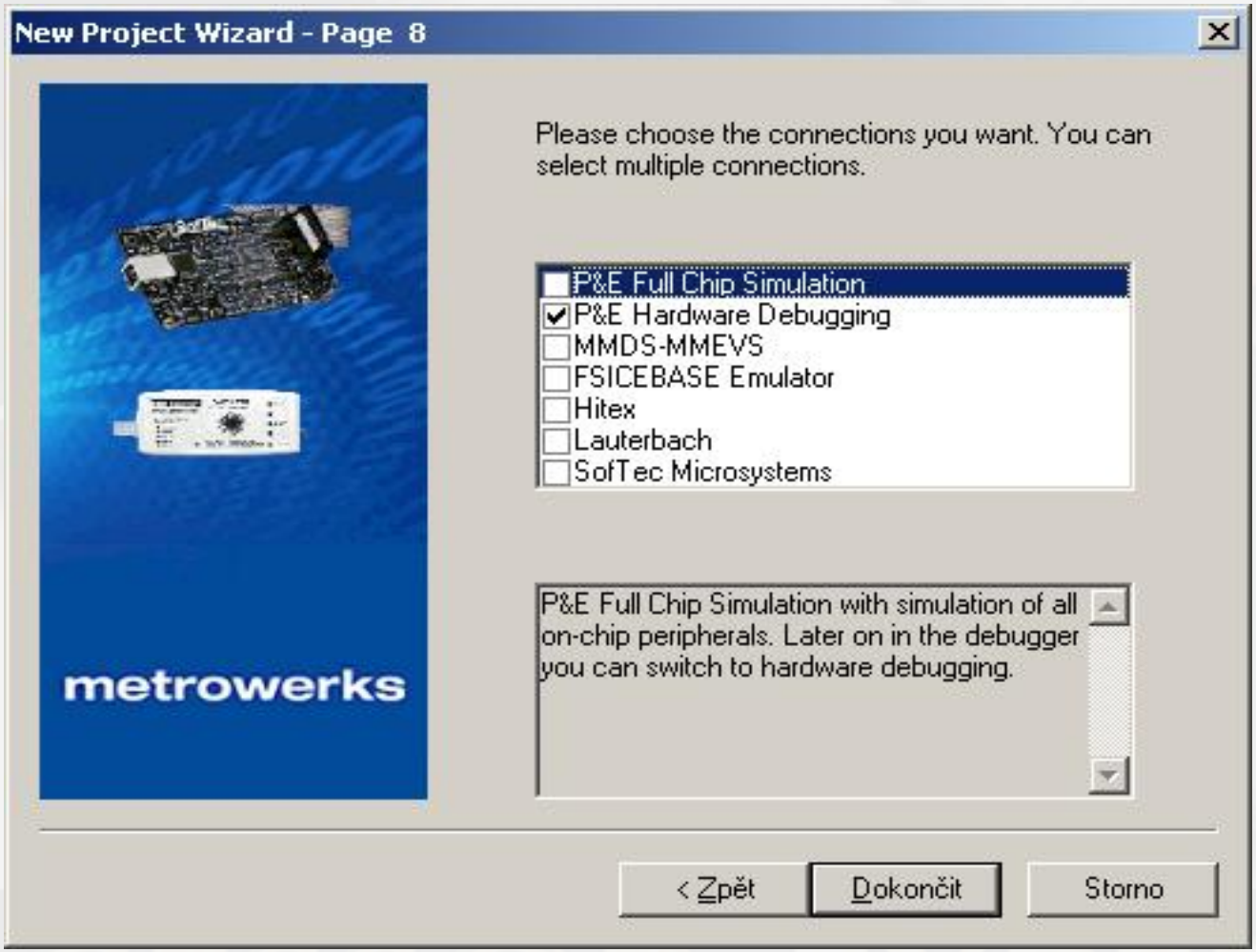
Copyright © 2005
UNIS spol. s r.o.

Create a New Project 6



Copyright © 2005
UNIS spol. s r.o.

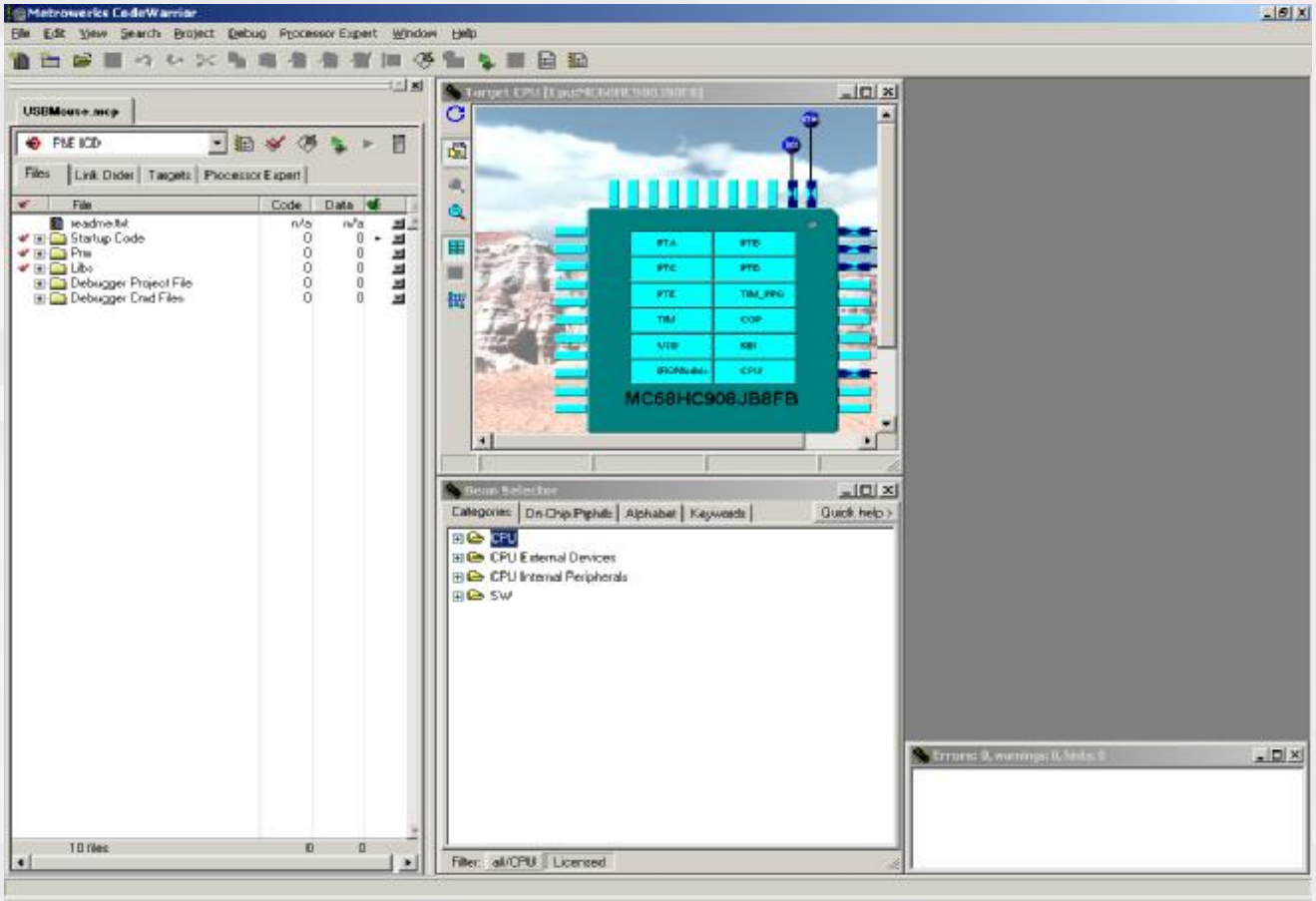
Create a New Project 7



Copyright © 2005
UNIS spol. s r.o.

Create a New Project 8

- Newly created project:

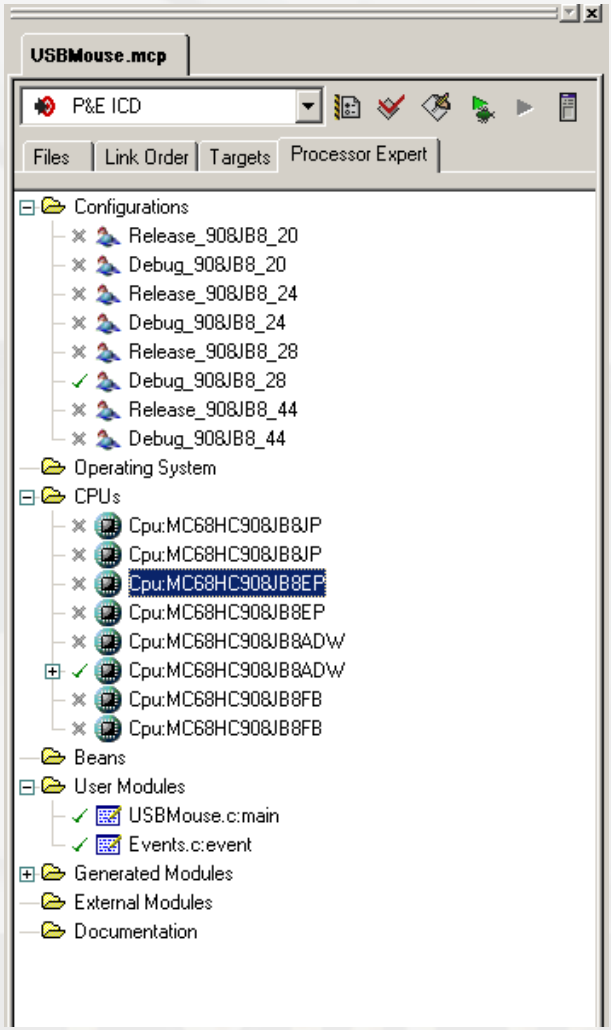


Copyright © 2005
UNIS spol. s r.o.

Configure beans 1

- Select Configuration Debug_908JB8_28 in the Processor Expert tab.

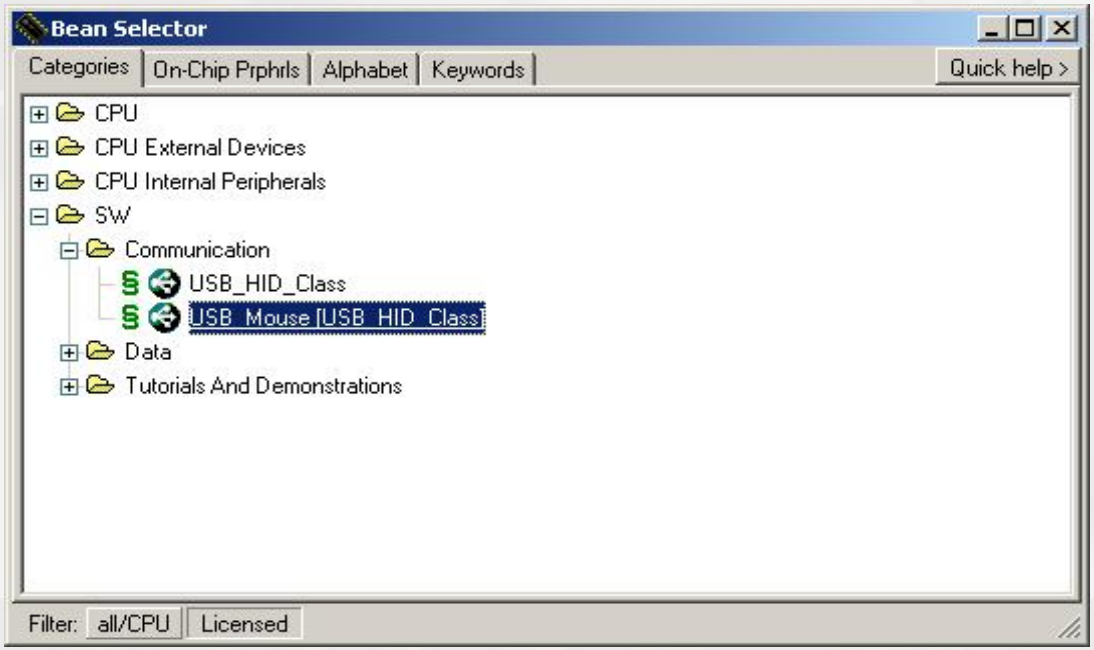
  (This is the right configuration for MCU installed on the development board – another configuration can be deleted)



Copyright © 2005
UNIS spol. s r.o.

Configure beans 2

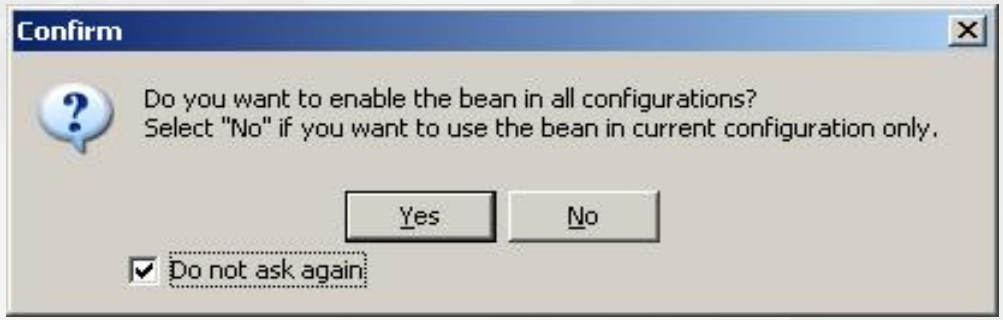
- Select Categories page in Bean selector. Expand and double click on SW\Communication\USB_Mouse[USB_HID_Class] bean.



Copyright © 2005
UNIS spol. s r.o.

Configure beans 3

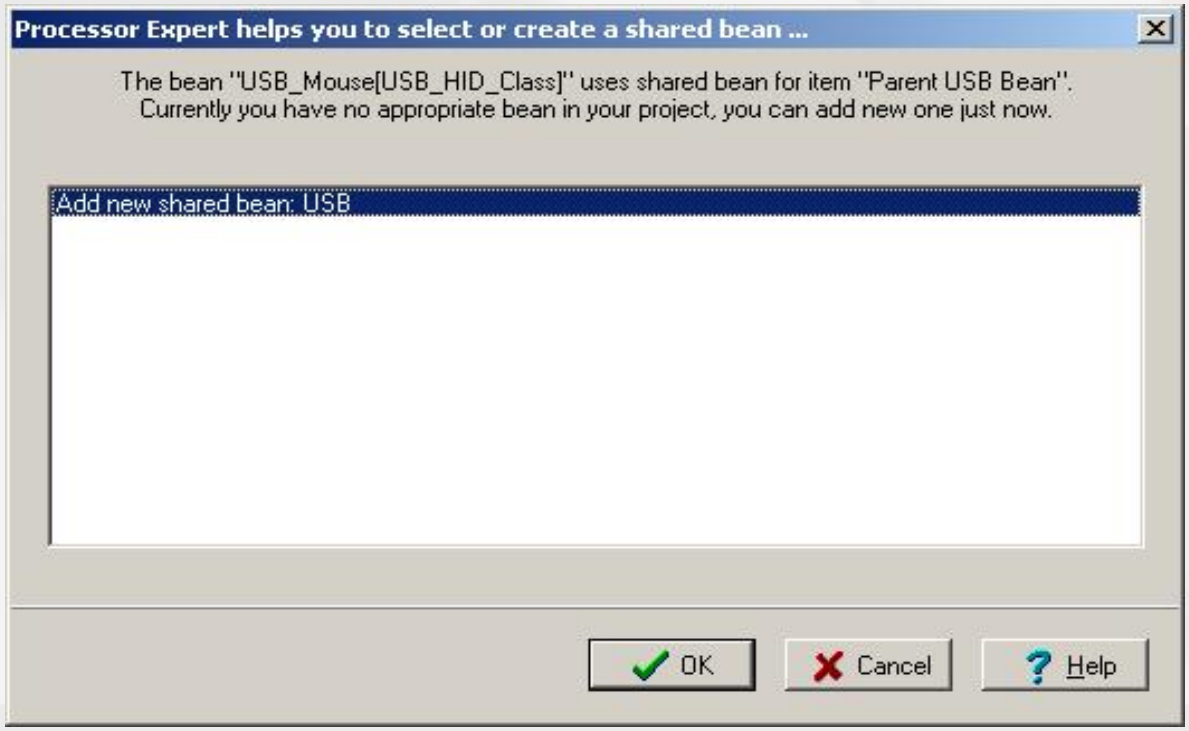
- The bean will be added to current project. In the following dialog check "Do not ask again" checkbox and click on Yes button.



Copyright © 2005
UNIS spol. s r.o.

Configure beans 4

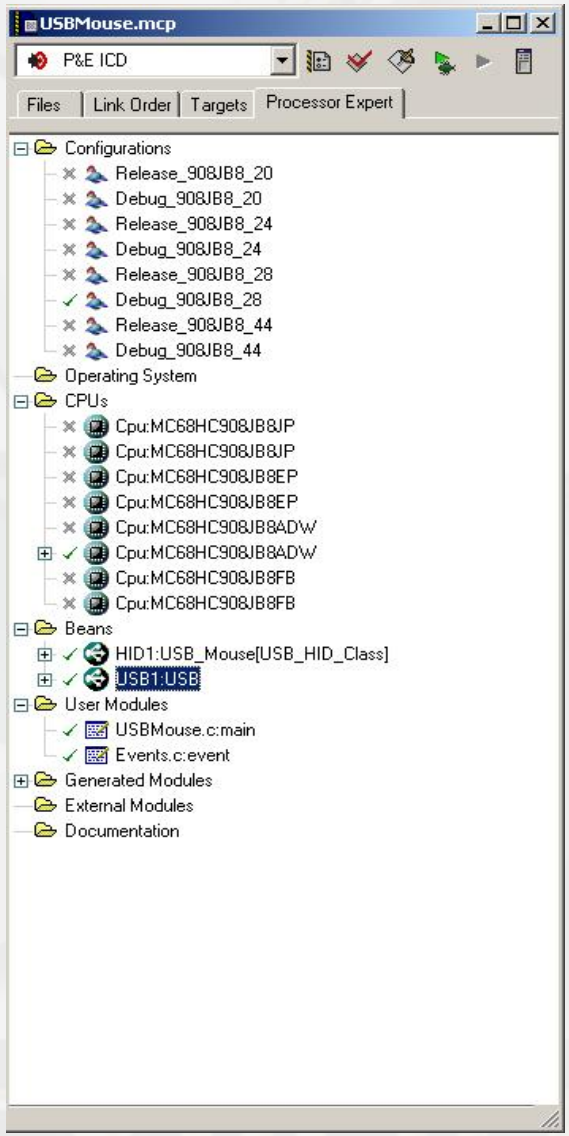
- The following dialog appears – click to OK button



Copyright © 2005
UNIS spol. s r.o.

Configure beans 5

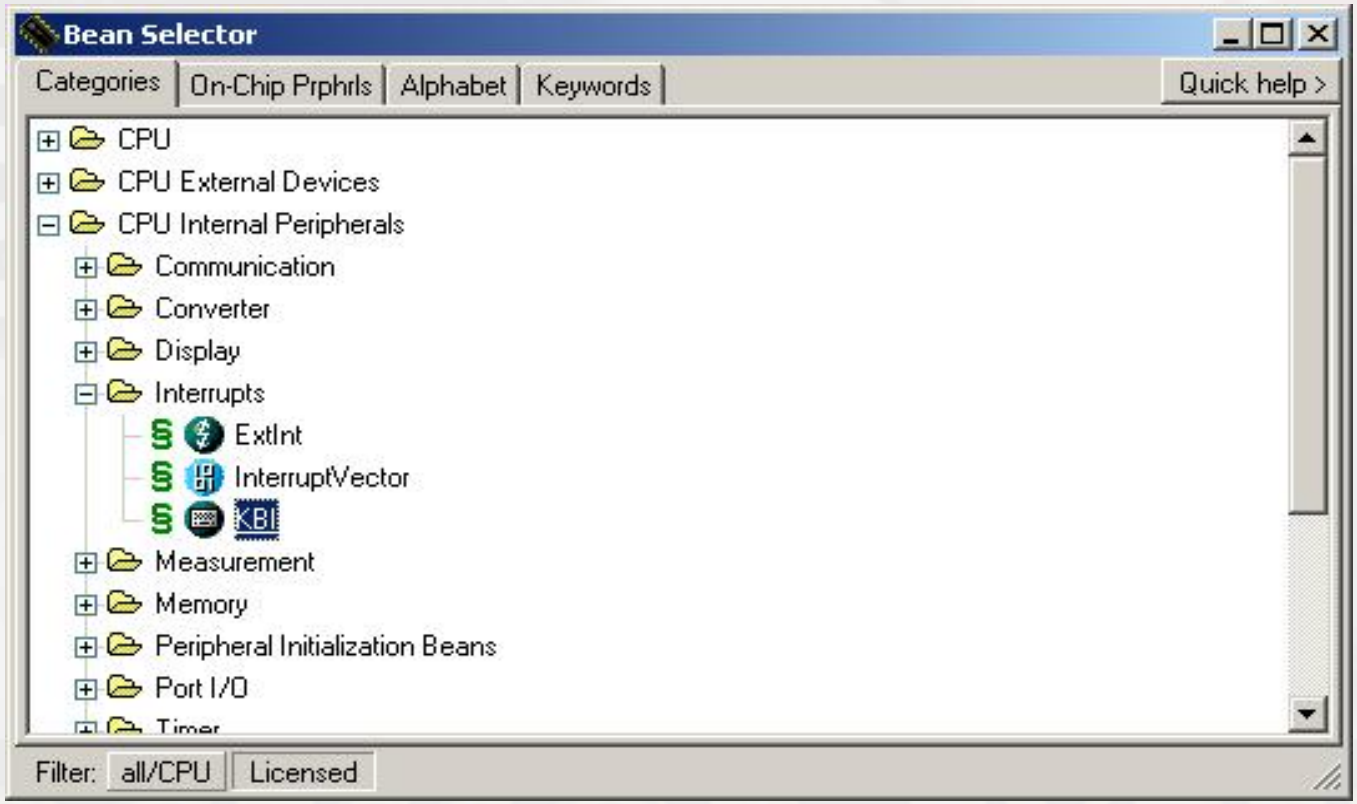
- HID1 and USB1 beans will be added to the project.



Copyright © 2005
UNIS spol. s r.o.

Configure beans 6

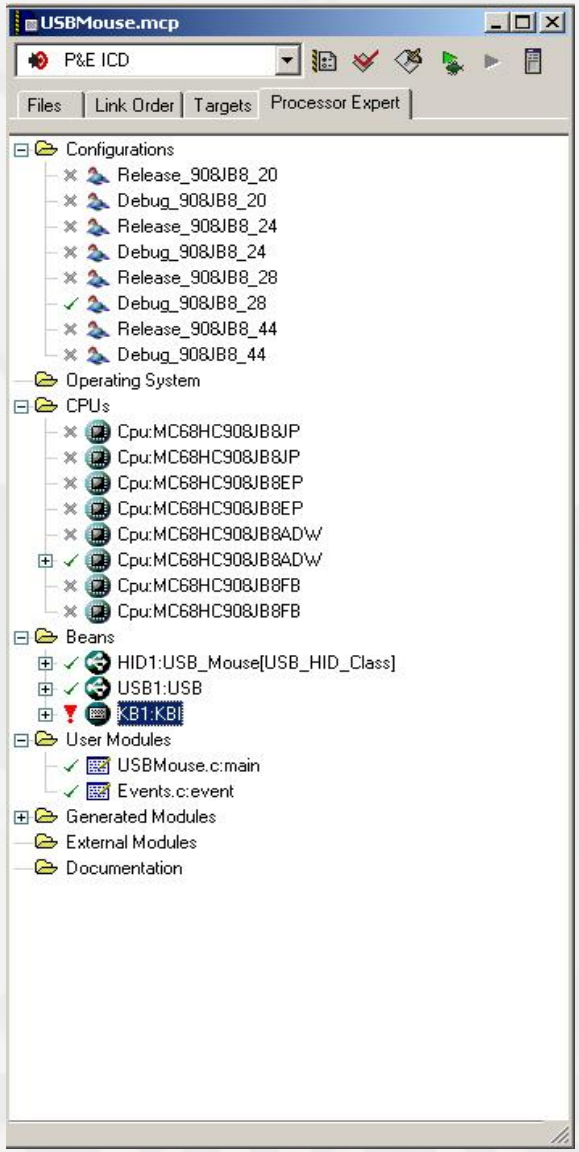
- Select KBI Bean from Bean selector



Copyright © 2005
UNIS spol. s r.o.

Configure beans 7

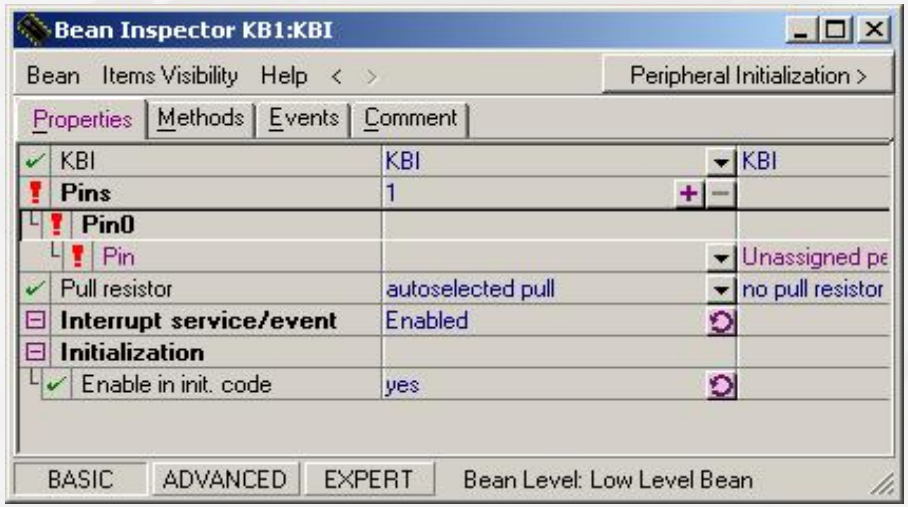
- KBI bean will be added to your project

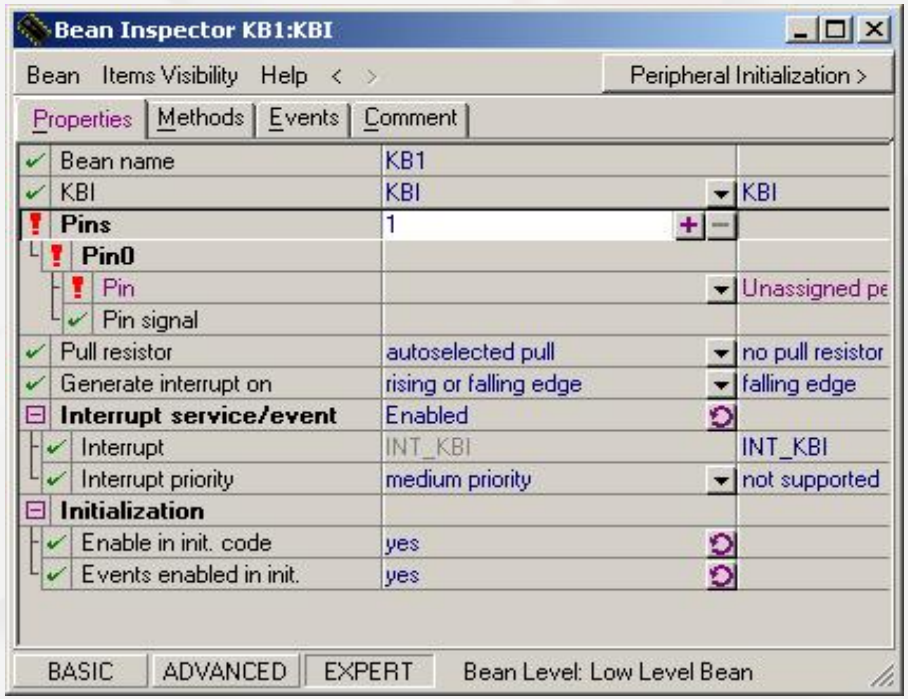


Copyright © 2005
UNIS spol. s r.o.

Configure beans 8

- Switch Items Visibility from Basic view to the Expert view



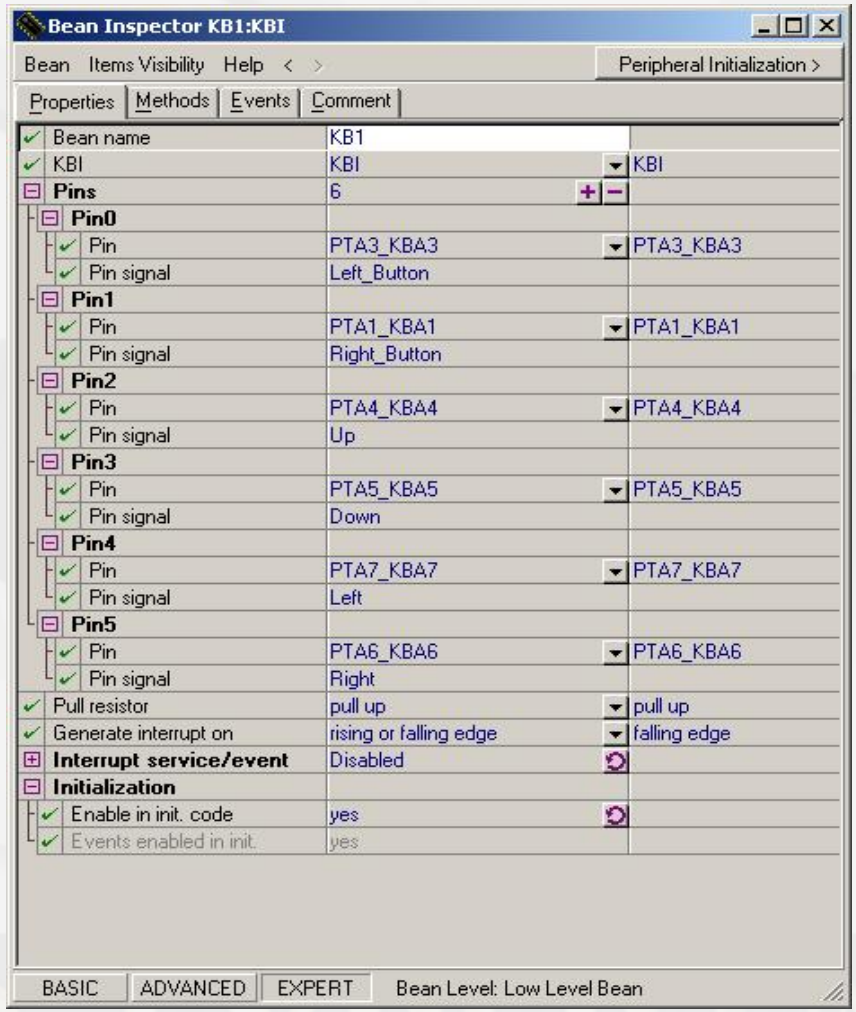


Copyright © 2005
UNIS spol. s r.o.

Configure beans 9

- Set KBI bean settings as shown on screenshot below accurately.

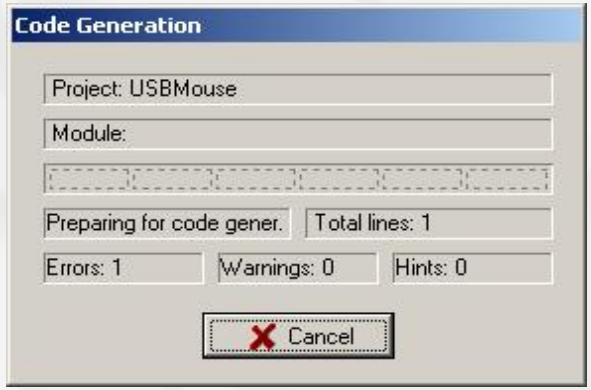
  (Note, pins order in the pin list, Pins values and Pin signal name values are important)



Copyright © 2005
UNIS spol. s r.o.

Generate project

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



- When successfully finished
  - Browse the generated code
- Open USBMouse.c file – this source contains main routine

Copyright © 2005
UNIS spol. s r.o.

Write application code 1

- Write the main routine loop
  - Get button status
  - Initialize input report
  - Update report
  - Send report

Copyright © 2005
UNIS spol. s r.o.

Write application code 2

- Copy prepared main function code to the main function.
  - Original main() function body generated by Processor Expert:

jklojkl.c

Path: D:\Documents and Settings\Marek\Dokumenty\jklojkl\CODE\jklojkl.c

```
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

void main(void)
{
  /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!!
  PE_low_level_init();
  /*** End of Processor Expert internal initialization.

  /* Write your code here */

  /*** Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! ***
  for(;;){}
  /*** Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! ***
} /*** End of main routine. DO NOT MODIFY THIS TEXT!!! ***/

/* END jklojkl */
/*
** ###################################################################
**
**     This file was created by UNIS Processor Expert 2.96 [03.67]
**     for the Freescale HC08 series of microcontrollers.
**
** ###################################################################
*/
```

Line 1 Col 1

Copyright © 2005
UNIS spol. s r.o.

Write application code 3

- Predefined main()

USBMouse.c

Path: E:\Metrowerks\Projects\USB\USBMouse\CODE\USBMouse.c

```
void main(void)
{

#define Increment 1
byte KB_Data;
HID1_DefineInputReport(1,Report_1)

  /*** Processor Expert internal initialization. DON'T REMOVE THIS CODE!!! ***/
  PE_low_level_init();
  /*** End of Processor Expert internal initialization.                    ***/

  for(;;) {
    KB_Data=KB1_GetVal();
    KB_Data=~KB_Data;

    HID1_InitInputReport(1,Report_1);

    HID1_SetReport(Report_1,Button_1,(KB_Data & Left_Button)>>1));
    HID1_SetReport(Report_1,Button_2,KB_Data & Right_Button);

    if (KB_Data & Left)                        /* Left button pressed ? */
      HID1_SetReport(Report_1,X,-Increment);   /* Decrement X position counter */
    if (KB_Data & Right)                       /* Right button pressed ? */
      HID1_SetReport(Report_1,X,+Increment);   /* Increment X position counter */
    if (KB_Data & Down)                        /* Down button pressed ? */
      HID1_SetReport(Report_1,Y,+Increment);   /* Increment Y position counter */
    if (KB_Data & Up)                          /* Up button pressed ? */
      HID1_SetReport(Report_1,Y,-Increment);   /* Decrement Y position counter */

    while(HID1_SendReport(1,Report_1,TRUE,FALSE)!=ERR_OK); /* Send report */

  }

  /*** Processor Expert end of main routine. DON'T MODIFY THIS CODE!!! ***/
  for(;;){}
  /*** Processor Expert end of main routine. DON'T WRITE CODE BELOW!!! ***/
} /*** End of main routine. DO NOT MODIFY THIS TEXT!!! ***/
```

Line 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

Copyright © 2005
UNIS spol. s r.o.

Demo board settings

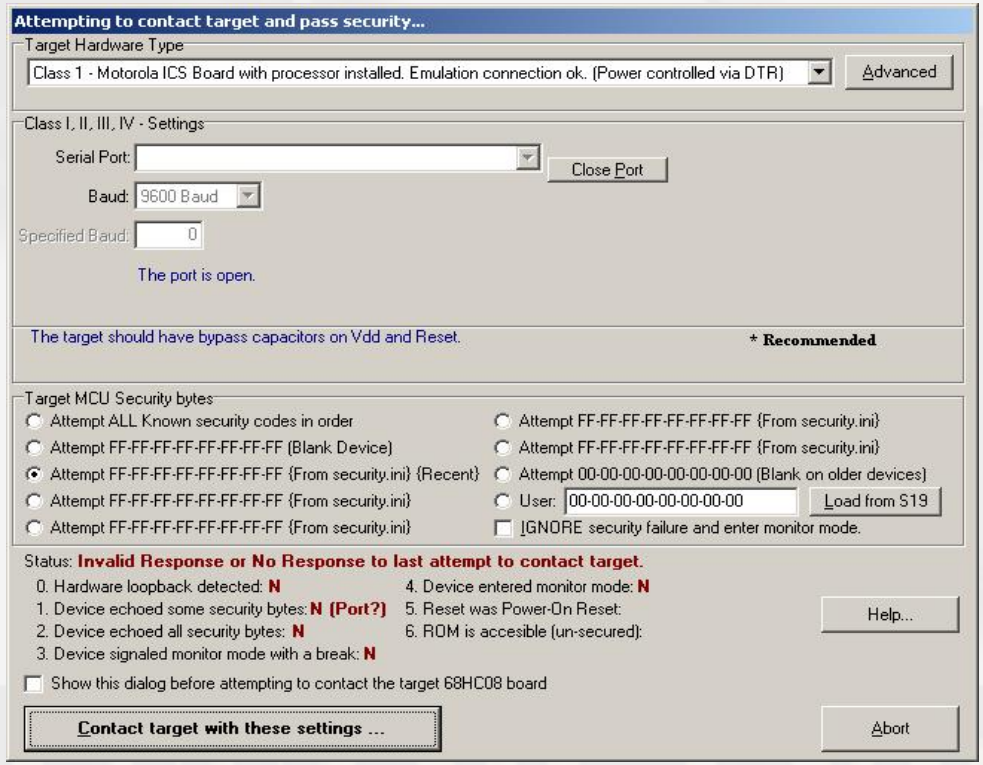
- Make sure the Demo board is configured properly
  - JP4 (power source selection) – **3-4**
  - JP1, JP2 – debug interface selection – **both open**

For more details see Demo board manual

Copyright © 2005
UNIS spol. s r.o.

Debbuger Setting 1

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



Copyright © 2005
UNIS spol. s r.o.

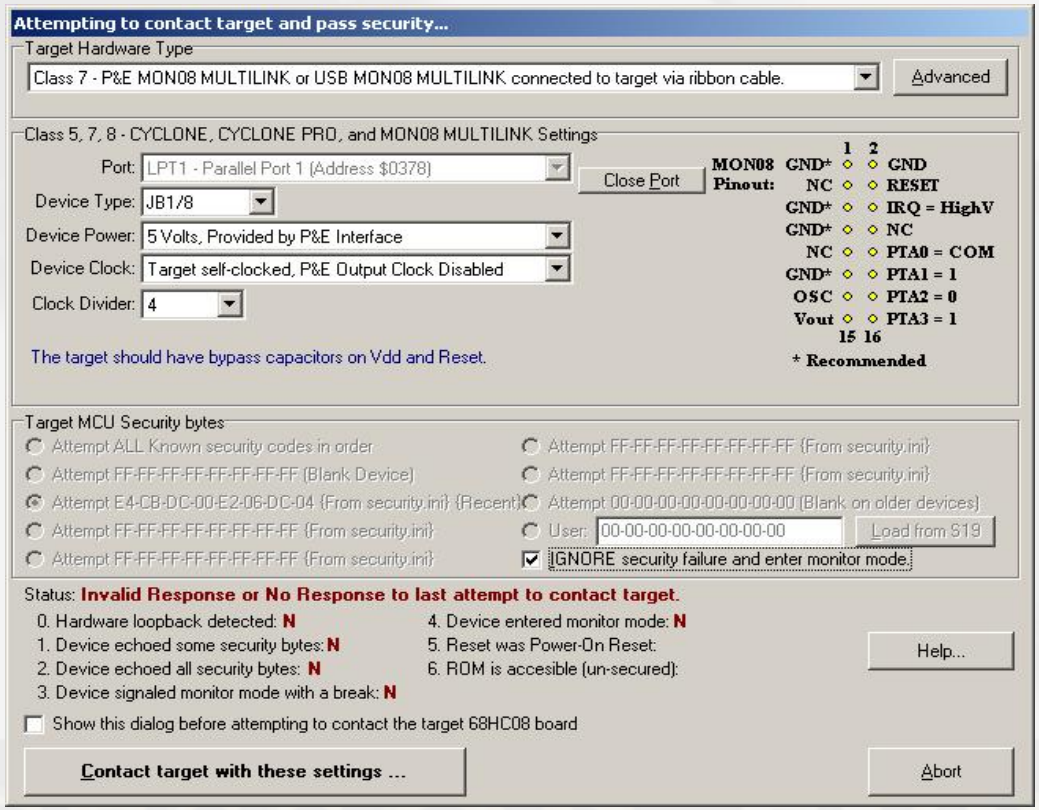
Debbuger Setting 2

- 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 securirty bytes of the MCU are not known)
- Click on "Contact target with these settings…" button

Copyright © 2005
UNIS spol. s r.o.

Debbuger Setting 3

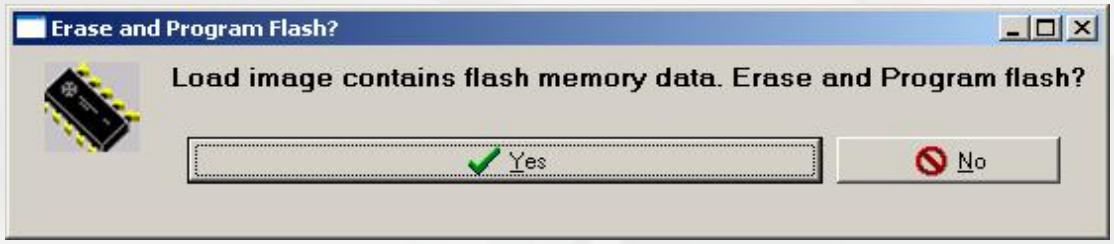
- When all described properties are set the window should look like



Copyright © 2005
UNIS spol. s r.o.

Debbuger Setting 4

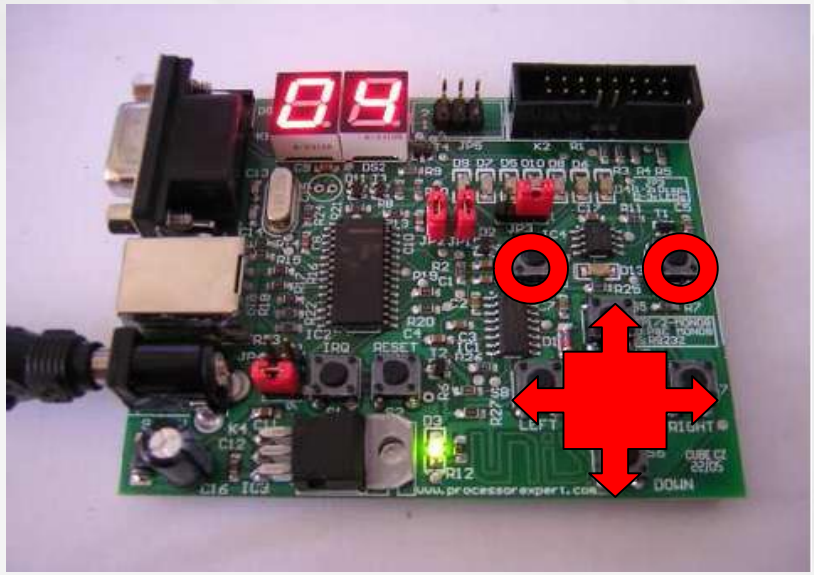
- Click on Yes button to erase and program MCU



Copyright © 2005
UNIS spol. s r.o.

Test the application

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



Copyright © 2005
UNIS spol. s r.o.

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

Copyright © 2005
UNIS spol. s r.o.

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

Copyright © 2005
UNIS spol. s r.o.