Creating an Atmel QTouch Library Project
Using GCC or IAR

2009-07-01, v08, Paul, Atmel QRG FAE

Ensure you have all the required software installed. Atmel Touch information is here:
http://www.atmel.com ➔Products ➔ Touch Technology ➔ Tools & Software
http://www.atmel.com ➔Products ➔ Touch Technology ➔ QTouch™ Library
http://www.atmel.com ➔Products ➔ Touch Technology ➔ Application Notes ➔ Touch Sensors Design Guide

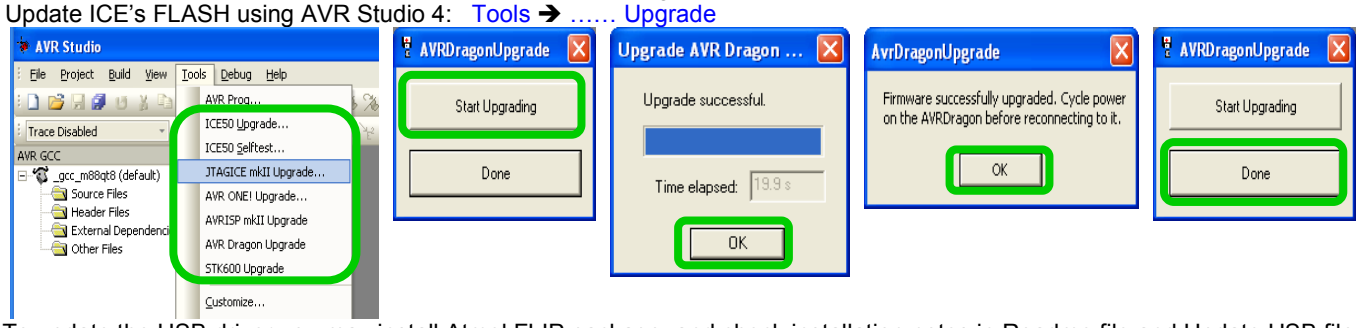
You will need these items:

a. The Datasheet for the Atmel AVR IC you will use.
b. Atmel QTouch Library 2.0, Atmel QTouch Library User Guide doc8207, AVR QTouch Studio (Viewer for Standard demos).
c. PCB: AVRTS2080A, AVRTS2080B, or your own design…
   Some older evaluation units have short ICE headers, **carefully** removing the plastic on the header gives better ICE connection.
d. Development Environment (Compiler, Debugger…):
   - AVR Studio 4 with WINAVR GCC, see step g0
   - IAR Embedded Workbench for Atmel AVR - Full or free Kick Start 4, see step i0
e. ICE appropriate for the AVR IC being programmed: JTAGICE MKII, AVR Dragon, STK600, STK500… (ICE = In Circuit emulator)
f. Optional: **Hawkeye** for viewing customized data output (Available from Atmel FAEs)
g. Optional: **Flip**: http://atmel.com/dyn/products/tools_card.asp?tool_id=3886 – to program or update AT90USB used on Demos

ICE ISP frequency: ensure ≤¼ target's frequency (although slower may be appropriate for some designs, like 125KHz)

ICE Upgrade: Ensure your ICE is up to date:

- To update the ICE driver install the latest AVR Studio 4 package, and check installation notes (Readme file…)
  Update ICE's FLASH using AVR Studio 4: Tools ➔ …… Upgrade



- To update the USB driver you may install Atmel FLIP package, and check installation notes in Readme file and Update USB file.
  - Disconnect other USB devices before doing this procedure.
  - USB port may show with different driver names. Plug/Unplug device to find which device in the list is correct.
  - See [C:\Program Files\Atmel\Flip 3.3.2\info\Updating the USB Driver - Windows XP.html]

Important: Ensure you check **all** items indicated with: ▭

Using IAR Embedded Workbench for Atmel AVR

i0. This Guide prepared using: IAR Embedded Workbench for Atmel AVR, v. 5.20, 4K Kickstart edition.
http://www.iar.com/website1/1.0.1.0/107/1/index.php - Full or free Kick Start 4

i1. Create a new empty folder for the project files (Folder location is your choice).

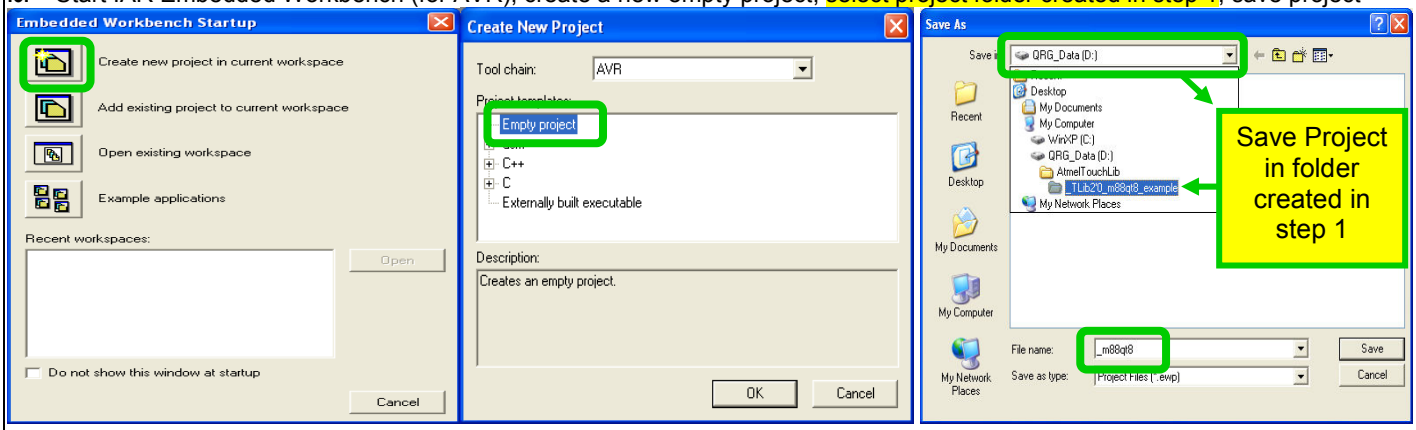
i2. Copy 3 files to the project folder you created in step i1: **Main.c, API.h, Library.r90**

- [TLib Path] = C:\Program Files\Atmel\Atmel QTouch Libaries 2.0\megaAVR, tinyAVRand XMEGA libray\
- IAR, AVRTS2080A/EVK2080A, ATmega88, QTouch8, Slider/Rotor/Keys: (Rotor==Wheel)
  - Library: [TLib Path]\QTouch library\library files\libv1g4_8qt_krs.r90[1]
  - Example H File: [TLib Path]\QTouch library\Example projects\v1g4_8qt_example\touch_api_atmega88.h
  - Example C File: [TLib Path]\QTouch library\Example projects\v1g4_8qt_example\main_atmega88.c[1]

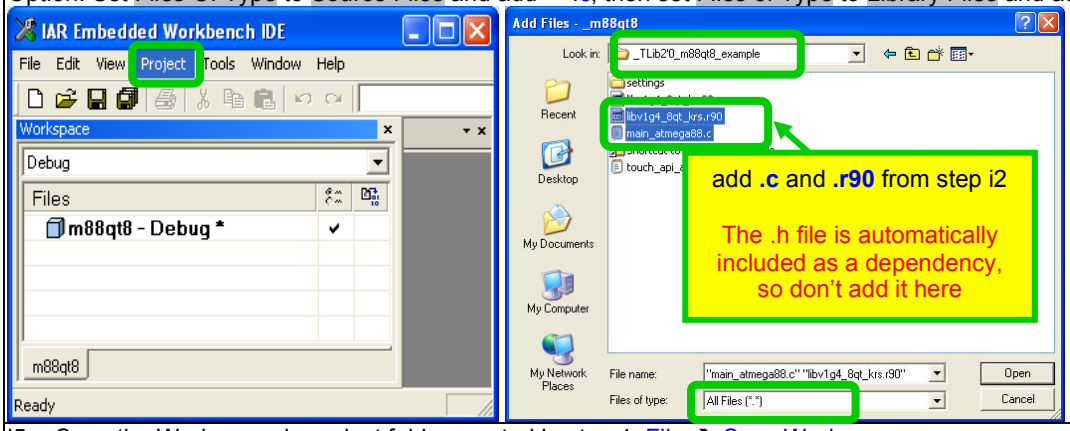
  [1] For Keys only use libv1g4_8qt_k.r90, remove definition of _ROTOR_SLIDER_ (Step i8), and edit the C file.
- IAR, AVRTS2080B/EVK2080B, ATtiny88, QMatrix8, 4MHz, Dwell=1, Slider/Rotor (Rotor==Wheel)
  - Library: [TLib Path]\QMatrix library\library files\libv1_8qm_d01_krs.r90[2]
  - Example H File: [TLib Path]\QMatrix library\Example projects\v1_8qm_example\touch_lib_krs.h[2]
  - Example C File: [TLib Path]\QMatrix library\Example projects\v1_8qm_example\main.c[2]

  [2] For Keys only use libv1_8qm_d01_k.r90, touch_lib_k.h, remove definition of _ROTOR_SLIDER_ (Step i8), and edit the C file.
- Library and Example Projects listed in tables 6-2, 6-4, 6-6, 6-8 in Atmel QTouch Library User Guide doc8207(c)
  Other AVR ICs may be supported by copying the appropriate files listed in Atmel QTouch Library User Guide doc8207.
- Style Options: (This Guide uses safe Production style where library files are only updated by manually replacing them.)
  Production - Put library files in project folder to ensure that future rebuild of project uses same files
  Development - Set project to point to files in original folders, so that project uses files from latest Touch Library update.
  (See: Project➔Options➔C/C++Compiler➔Preprocessor➔Additional Include directories)

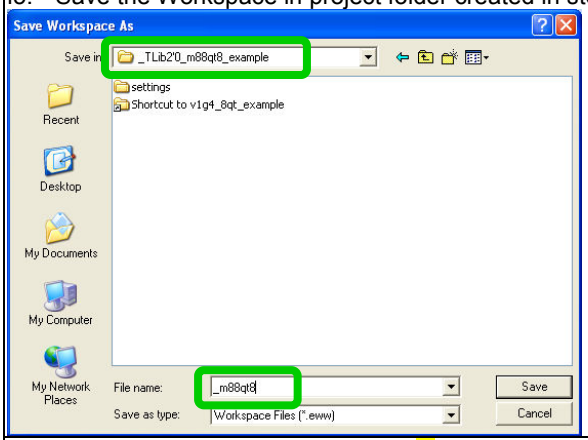
i3. Start IAR Embedded Workbench (for AVR), create a new empty project, select project folder created in step 1, save project



i4. Add **two** files to the project using: Project ➔ Add Files - Set for: **All Files [*.*]**, add **.c** and **.r90** from step i2 (Don't add .h file)
Option: Set Files Of Type to Source Files and add ***.c, then set Files of Type to Library Files and add ***.r90



i5. Save the Workspace in project folder created in step 1: File ➔ Save Workspace.

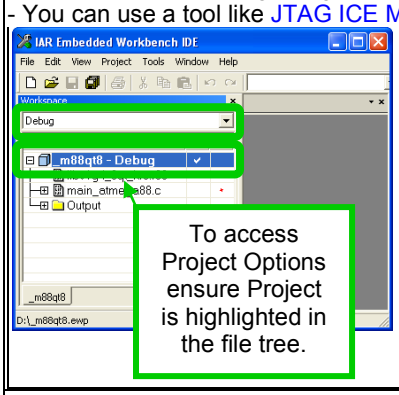


i6. For **Debug** Build continue at step i7, or for **Release** Build continue at step i17

IAR - Build for **Debug**

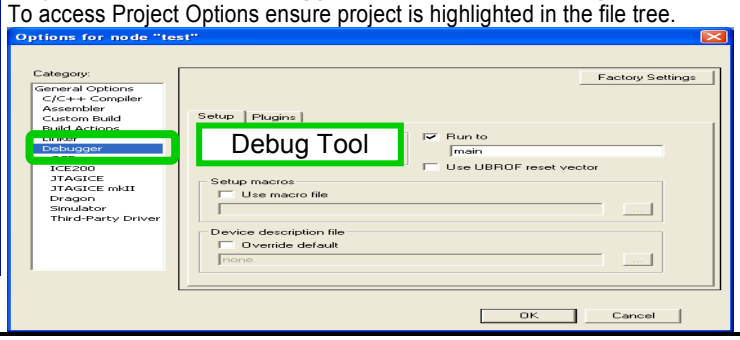
i7. Select **Debug** Build, and Select Debugger

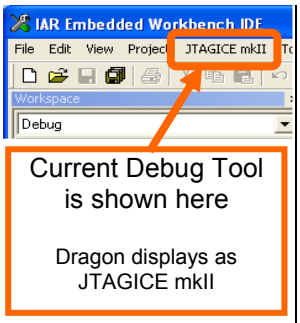
- Use **Debug** Build for development, output is a “C Spy” file which includes Debug Information
- Allows “C Code” debug using IAR Debugger: data and functions are accessed by their C Code names.
- You can use a tool like JTAG ICE MKII to debug the code on actual product hardware.



Project ➔ Options ➔ Debugger = JTAG ICE mkII, Dragon, …

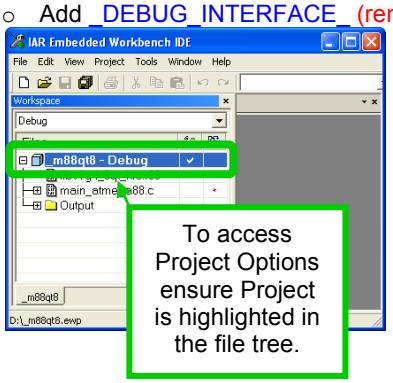
To access Project Options ensure project is highlighted in the file tree.

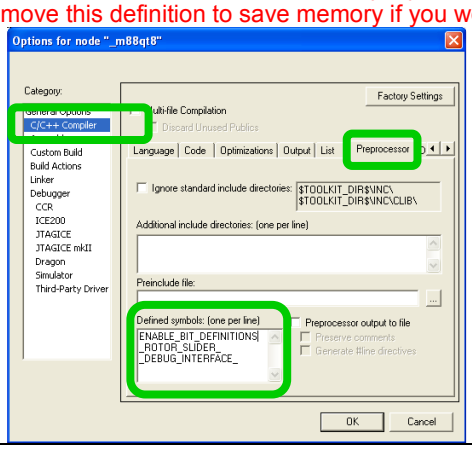




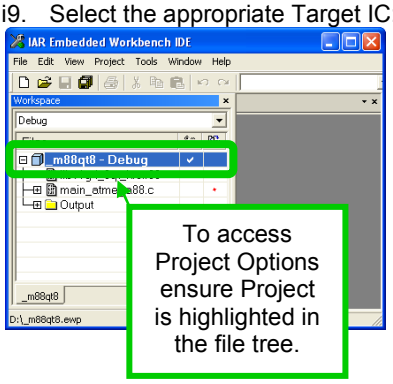
i8. Set TLib build options: Project ➔ Options ➔ C/C++ compiler ➔ Preprocessor Tab

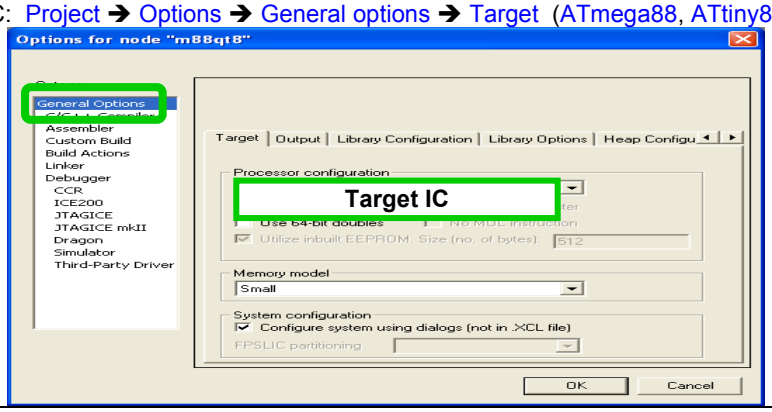
- Add ENABLE_BIT_DEFINITIONS (Used for bit shift in register control)
- Add _ROTOR_SLIDER_ (remove this definition to save memory if you only use keys, and not Rotors/Sliders)
- Add _DEBUG_INTERFACE_ (remove this definition to save memory if you won’t use debug serial output AVR QTouch Studio)





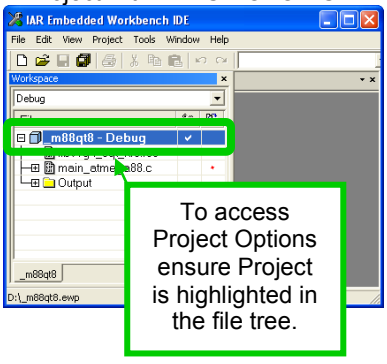
i9. Select the appropriate Target IC: Project ➔ Options ➔ General options ➔ Target (ATmega88, ATtiny88,ATmega88PA…)

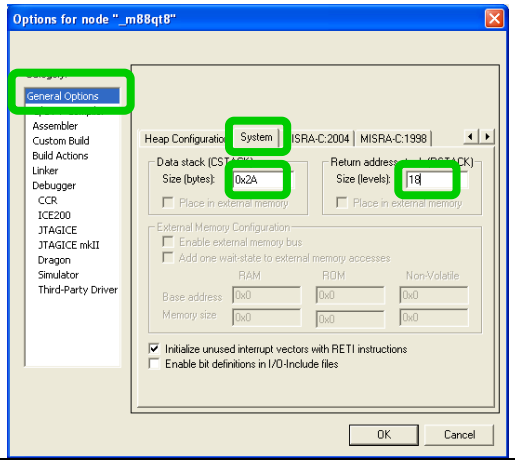




i10. Increase the stack size: Project ➔ Options ➔ General Options ➔ System

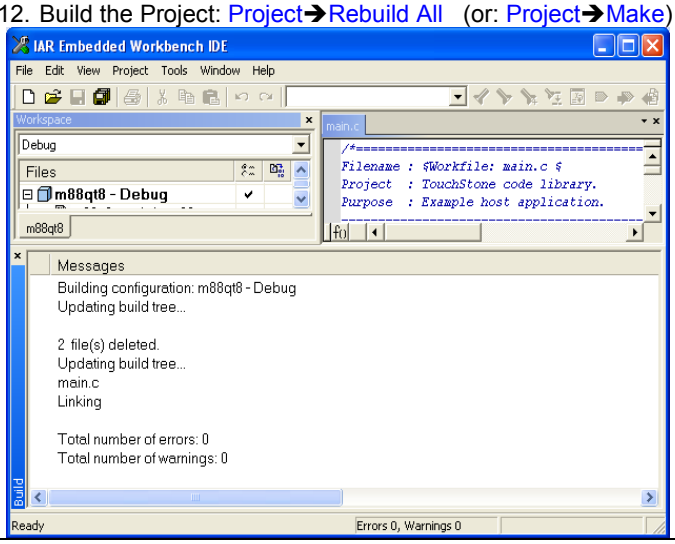
- Project with ONLY KEYS CSTACK ≥ 0x18 RSTACK ≥ 0x10
- Project with KEYS/ROTOR/SLIDER CSTACK ≥ 0x2A RSTACK ≥ 0x18





i11. Save the Workspace: File ➔ Save Workspace.

i12. Build the Project: Project➔Rebuild All (or: Project➔Make)



- If any errors then check that you have a matching set of:
  Target IC, Main.c, Header.h, Library.r90
- **Known Issue:** Atmel QTouch Library User Guide doc8207(c) section 7
  Please ignore these type of Warnings when using TLIB2.0
  Builds for some AVR will present this warning.

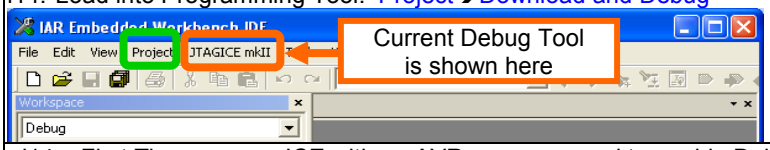
  Linking
  Warning[w6]: Type conflict for external/entry "_A_DDRC", in module main_atmega88 against external/entry in module burst_10_BC; class/struct/union field/base types do not match for field/base "; class/struct/union field names do not match: DDRC_Dummy7 vs DDRC_DDC7
  :
  :
  Total number of errors: 0
  Total number of warnings: **

i13. Connect Programmer. For photos of ICE connections see AVR QTouch Studio – Help – In Circuit reprogramming

- Powerdown Target
- Connect ICE to Target (ISP 6wire, or dW 2wire if DebugWire was already enabled)
- Connect ICE to PC (USB or RS-232)
- Recheck ICE Pin1 is correct
- Powerup Target (Connect USB)
- Turn on ICE

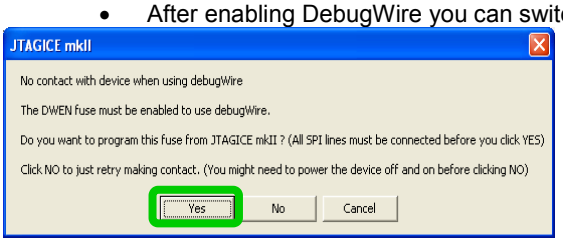
*Suggestion:
Make pin alignment mark on PCB and ICE using a bright colour paint pen:
Pin5 - Gnd
Pin6 - Reset/dW

i14. Load into Programming Tool: Project➔Download and Debug

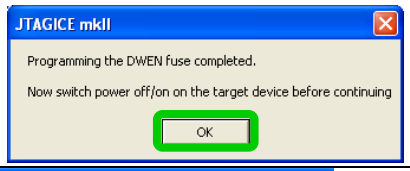


i14a. First Time you use ICE with an AVR you may need to enable DebugWire (DWEN Fuse for dW):
Project➔Download and Debug ➔Yes➔Power Down Target ➔ Power Up Target ➔OK

- To enable DebugWire ensure ICE is connected to Target using ISP 6wire cable (Vcc, SPI/ISP, Reset/dW, Gnd)
- After enabling DebugWire you can switch to dW 2wire cable (dW, Gnd), freeing the ISP pins for Touch, SPI…

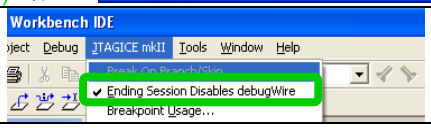


➔ Power Down Target
(Unplug USB from EVK2080)
➔
Option: ICE cable change ISP 6wire to dW 2wire
➔
Power Up Target
(Connect USB to EVK2080)



i14b. Note: To disable debugWire for Low Power measurements:

- In Debugger Menu enable: ✓Ending Session Disables DebugWire
- Exit Debug mode: Debug ➔ Stop Debugging (auto disables DWEN fuse)
- You will need ISP 6wire cable to re-enable DebugWire

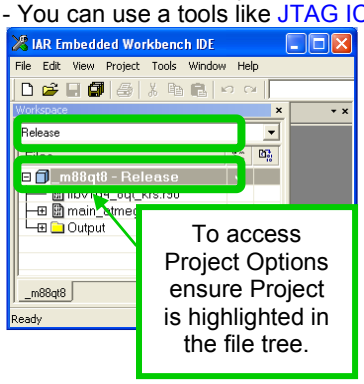


i15. Run the Program and do your tests: Debug➔Go
If your program outputs appropriate diagnostic data then you may view the data using AVR QTouch Studio
Note: Pins connected to ICE won't be able to Touch Detect. Use DebugWire cable to free these pins (see step i14a).

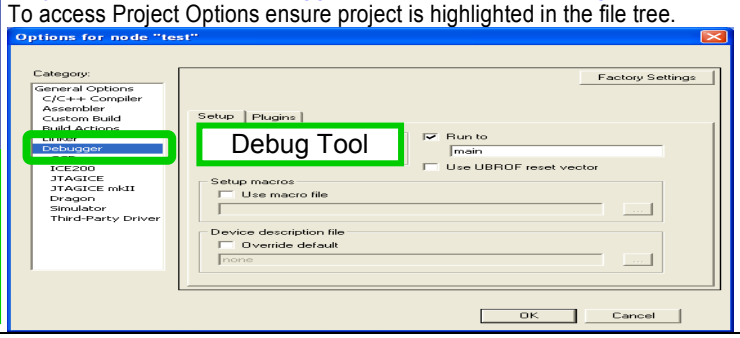
i16. To stop debugging: Debug➔Stop Debugging
For Release Build continue at step i17
If finished continue at step i28

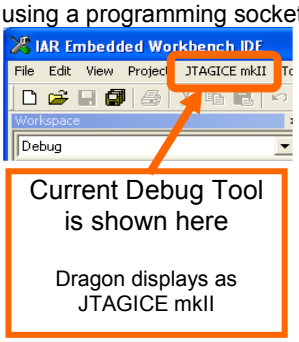
IAR - Build for **Release**

i17. Select **Release** Build, and select Flash Programming Tool
- Use to create a file for final testing and production
- Output is FLASH only, it shouldn't contain debug info (typical formats include "motorola", "intel hex"…)
- You can use a tools like JTAG ICE MKII to program ICs mounted on PCB, or STK600 to program ICs using a programming socket.

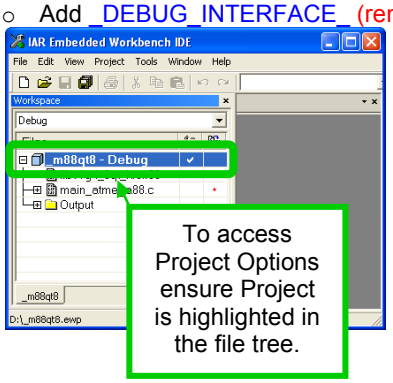


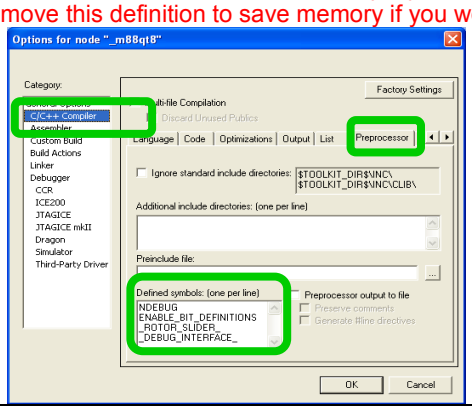
Project ➔ Options ➔ Debugger = JTAG ICE mkII, Dragon, …
To access Project Options ensure project is highlighted in the file tree.





i18. Set TLib build options: Project ➔ Options ➔ C/C++ compiler ➔ Preprocessor Tab
- Add NDEBUG (Default setting for Release Build: No Debug)
- Add ENABLE_BIT_DEFINITIONS (Used for bit shift in register control)
- Add _ROTOR_SLIDER_ (remove this definition to save memory if you only use keys, and not Rotors/Sliders)
- Add _DEBUG_INTERFACE_ (remove this definition to save memory if you won't use debug serial output to AVR QTouch Studio)

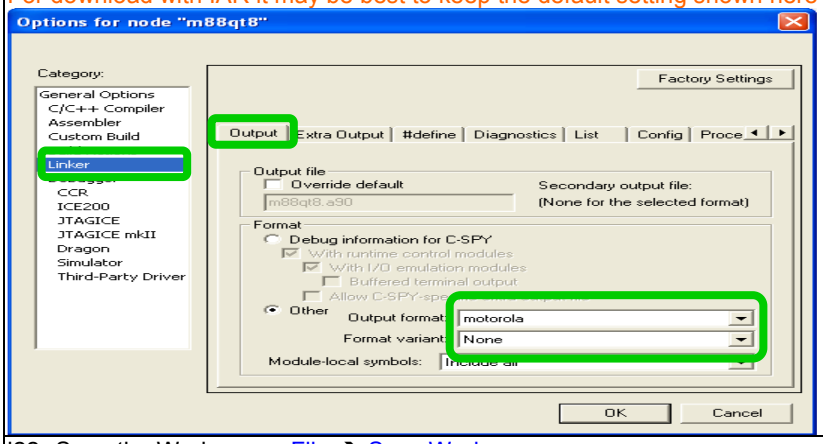




i19. Select the appropriate Target IC: Project ➔ Options ➔ General options ➔ Target (ATmega88, ATtiny88,ATmega88PA…)
See Step i9 (To access Project Options ensure project is highlighted in the file tree.)

i20. Increase the stack size: Project ➔ Options ➔ General Options ➔ System
See Step i10 (To access Project Options ensure project is highlighted in the file tree.)

i21. Set Output Filename and Format: Project ➔ Options ➔ Linker ➔ Output
For download with IAR it may be best to keep the default setting shown here (Format: **Motorola**, Variant: **None**):

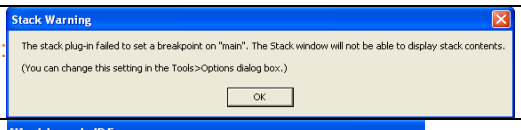


i22. Save the Workspace: File ➔ Save Workspace.

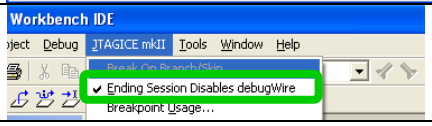
i23. Build the Project: Project➔Rebuild All (or: Project➔Make)
See Step i12

i24. Connect Programmer using ISP 6wire cable.
See Step i13

i25. Load into Programming Tool: Project➔Download and Debug
Since Debug information isn't included in the release build you may see this message:



i26. Set for DebugWire disable upon exit to minimize IC power:
Debugger Menu enable: ✓Ending Session Disables DebugWire.



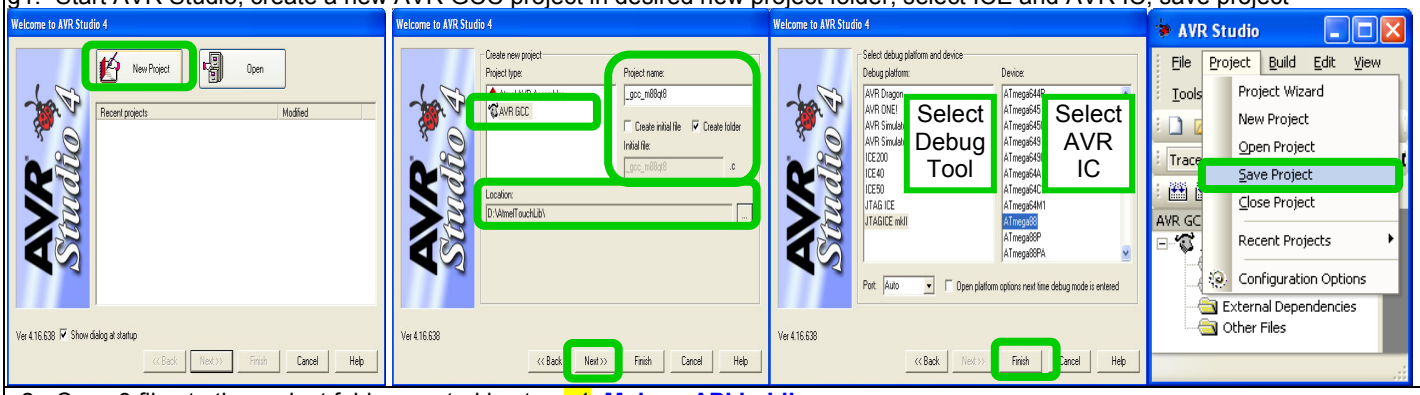
i27. Finished Programming: Debug➔Stop Debugging (DebugWire module in IC is disabled through setting in step i26)

IAR - Finish
i28. Powerdown Target (Unplug USB from EVK2080)
i29. Power down Programmer.
i30. Disconnect Programmer from Target.
i31. Powerup Target and test it.
i32. Log results.
i33. Exit all software tools
IAR Completed

Using AVR Studio 4 and WinAVR GCC

g0. This document prepared using: AVRStudio4.16 + SPI (Build638), WinAVR-20090313-install.exe, and Tlib2.0 (Atmel Touch Library v2.0)
- Choice: (a) Install WINAVR, then AVR Studio 4, then Tlib (b) Exit AVR Studio 4, then install WINAVR, then Tlib
- http://support.atmel.no/bin/customer?custSessionKey=&customerLang=en&noCookies=true&action=viewKbEntry&id=226

g1. Start AVR Studio, create a new AVR GCC project in desired new project folder, select ICE and AVR IC, save project

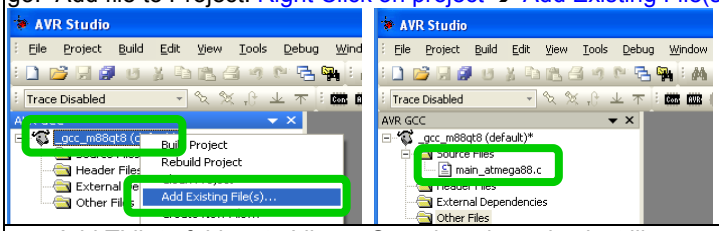


g2. Copy 3 files to the project folder created in step g1: **Main.c, API.h, Library.a**
- [TLib Path] = C:\Program Files\Atmel\Atmel QTouch Libaries 2.0\megaAVR, tinyAVRand XMEGA libray\
- IAR, AVRTS2080A/EVK2080A, ATmega88, QTouch8, 4MHz, Slider/Rotor/Keys: (Rotor==Wheel)
  - Library: [TLib Path]\QTouch library\library files\libavr4g2_8qt_krs.a[1]
  - Example H File: [TLib Path]\QTouch library\Example projects\avr4g2_8qt_example\touch_api_atmega88.h[1]
  - Example C File: [TLib Path]\QTouch library\Example projects\avr4g2_8qt_example\main_atmega88.c[1]

  [1] For Keys only use libavr4g2_8qt_k.a, remove definition of _ROTOR_SLIDER_ (Step g5), and edit the C file.
- IAR, AVRTS2080B/EVK2080B, ATtiny88, QMatrix8, 4MHz, Dwell=1, Slider/Rotor (Rotor==Wheel)
  - Library: [TLib Path]\QMatrix library\library files\libt88_8qm_d1_krs.a[2]
  - Example H File: [TLib Path]\QMatrix library\Example projects\t88_8qm_example\touch_api_t88_8ch.h[2]
  - Example C File: [TLib Path]\QMatrix library\Example projects\t88_8qm_example\main_t88_8ch.c[2]

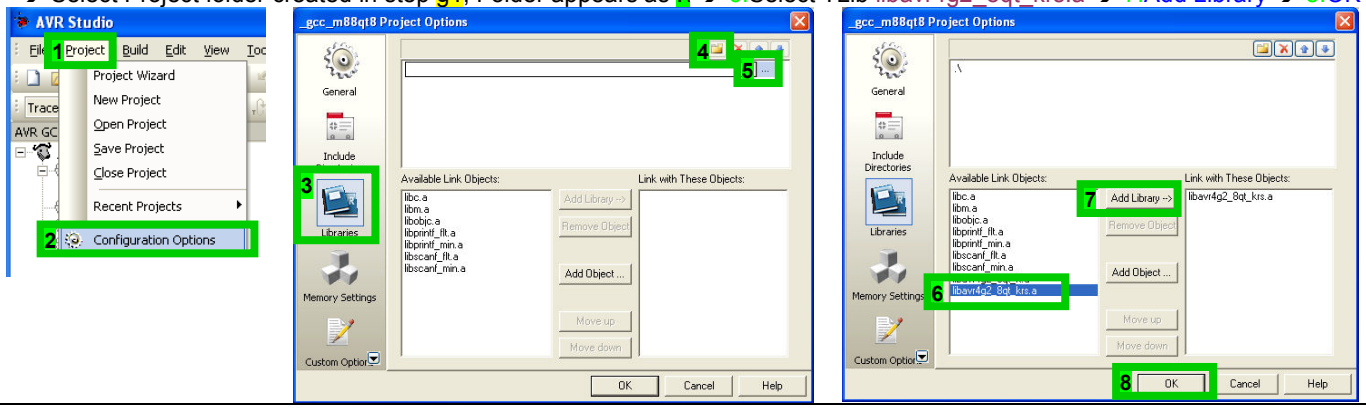
  [2] For Keys only use libt88_8qm_d1_k.a, touch_api_t88_8ch.h, remove definition of _ROTOR_SLIDER_ (Step g5), and edit the C file.
- Library and Example Projects listed in tables 6-2, 6-4, 6-6, 6-8 in Atmel QTouch Library User Guide doc8207(c)
  Other AVR ICs may be supported by copying the appropriate files listed in Atmel QTouch Library User Guide doc8207.
- Style Options: (This Guide uses safe Production style where library files are only updated by manually replacing them.)
  Production - Put library files in project folder to ensure that future rebuild of project uses same files
  Development - Set project to point to files in original folders, so that project uses files from latest Touch Library update.

g3. Add file to Project: Right Click on project ➔ Add Existing File(s)… ➔ add **.c** from step g2



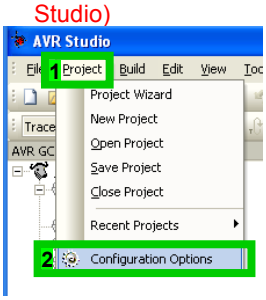
g4. Add TLib to folders to Library Search path, and select library

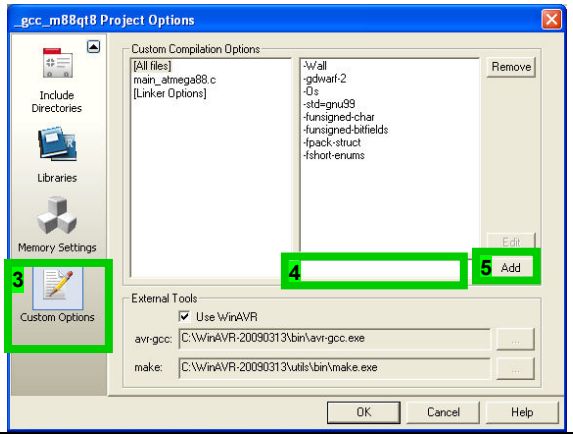
1.Project ➔ 2.Configuration Options ➔ 3.Libraries ➔ 4.Folder ➔ 5.Path
➔ Select Project folder created in step g1, Folder appears as .\ ➔ 6.Select TLib libavr4g2_8qt_krs.a ➔ 7.Add Library ➔ 8.OK

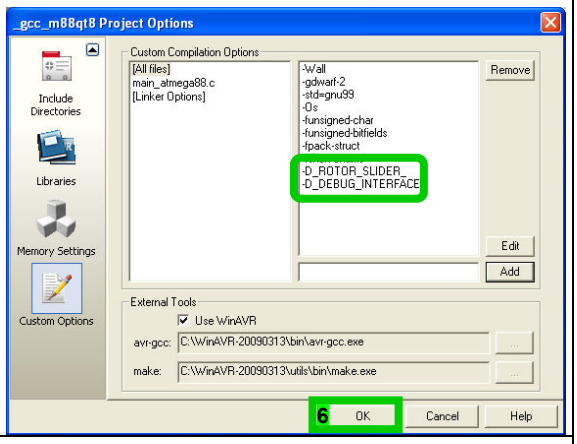


g5. Set TLib build options: 1.Project ➔ 2.Configuration Options ➔ 3.Custom Options ➔ 4.Item to Add: -D… ➔ 5.Add ➔6.OK
Repeat Steps 4 and 5 for each item. More…

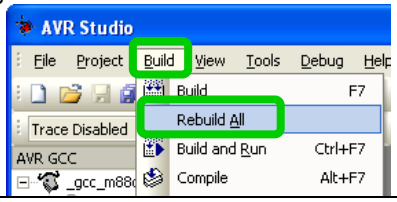
- Add: -D_ROTOR_SLIDER_ (remove this definition to save memory if you only use keys, and not Rotors/Sliders)
- Add: -D_DEBUG_INTERFACE_ (remove this definition to save memory if you won't use debug serial output to AVR QTouch Studio)







g6. Build: Build ➔ Rebuild All



If any errors then check that you have a matching set of:
Target IC, Main.c, Header.h, Library.a

WinAVR GCC Notes (compared to IAR):

- There is no difference between Debug Build and Release Build
- There is no need to set Stack sizes

g7. Connect Programmer. For photos of ICE connections see AVR QTouch Studio – Help – In Circuit reprogramming

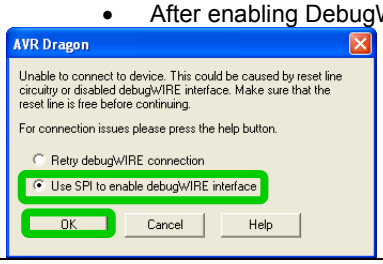
- Powerdown Target
- Connect ICE to Target (ISP 6wire, or dW 2wire if DebugWire was already enabled)
- Connect ICE to PC (USB or RS-232)
- Recheck ICE Pin1 is correct
- Powerup Target (Connect USB)
- Turn on ICE

*Suggestion:
Make pin alignment mark on PCB and ICE using a bright colour paint pen:
Pin5 - Gnd
Pin6 - Reset/dW

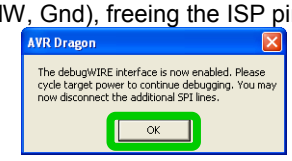
g8. Load into Programming Tool: Debug ➔ Start Debugging (or Build ➔ Build and Run)

g8a. First Time you use ICE with an AVR you may need to enable DebugWire (DWEN Fuse for dW):
Debug ➔ Start Debugging ➔ ⊙ Use SPI ➔ OK ➔ Power Down Target ➔ Power Up Target ➔OK

- To enable DebugWire ensure ICE is connected to Target using ISP 6wire cable (Vcc, SPI/ISP, Reset/dW, Gnd)
- After enabling DebugWire you can switch to dW 2wire cable (dW, Gnd), freeing the ISP pins for Touch, SPI…



➔ Power Down Target
(Unplug USB from EVK2080)
➔
Option: ICE cable change
ISP 6wire to dW 2wire
➔
Power Up Target
(Connect USB to EVK2080) ➔



g9. Run the Program and do your tests: Debug ➔Run
If your program outputs appropriate diagnostic data then you may view the data using AVR QTouch Studio
Note: Pins connected to ICE won't be able to Touch Detect. Use DebugWire cable to free these pins (see step g8a.).

g10.To stop debugging: Debug ➔ Break, then Debug ➔ Stop Debugging

g11.To disable debugWire for Low Power measurements:

- Stop any debugging session in progress: Debug ➔ Break, then Debug ➔ Stop Debugging
- Connect ICE to Target using 6wire ISP cable
- Start fresh debug using: Debug ➔ Start Debugging
- Select ICE Options: ALT-O, or bottom option in Debug pulldown menu: Debug ➔ **** Options (****=ICE you are using)
- Select: Connection ➔ Disable debugWire ➔Yes ➔ wait for message "…leaving debug mode" ➔ OK ➔ OK
  - In AVR Studio 4 Help see: On-Chip Debugging with the JTAGICE mkII, subsection: Re-enabling the ISP Interface

GCC - Finish

g12.Powerdown Target (Unplug USB from EVK2080)

g13.Power down Programmer.

g14.Disconnect Programmer from Target.

g15.Powerup Target and test it.

g16.Log results.

g17.Exit all software tools

GCC Completed

.