**CC1110-CC1111DK Development Board Notes**

* Git repository
  + HTTPS
    - <https://github.com/kylegaumnitz/murzuk.git>
  + SSH
    - [git@github.com:kylegaumnitz/murzuk.git](mailto:git@github.com:kylegaumnitz/murzuk.git)
  + Need to add team members as collaborators
* TI Wireless Tools – IAR Imbedded Workbench
  + Integrated IAR C/C++ Compiler and C-SPY debugger
    - Debugging using the IAR C-SPY Debugger
    - <http://supp.iar.com/FilesPublic/UPDINFO/011453/arm/doc/infocenter/tutor_debugging.ENU.html>
  + Complete solutions for RTOS (Real Time Operating Systems) and middleware such as TCP/IP and USB to help enhance embedded design
  + <https://www.iar.com/iar-embedded-workbench/partners/texas-instruments/TI-wireless/>
    - CC1110/CC1111 and CC2510/CC2511 (near the bottom)
    - SimpliciTITM
    - IAR Embedded Workbench for 8051
    - <https://www.iar.com/iar-embedded-workbench/#!?architecture=8051>
  + Available evaluation software
    - Evaluation editions available for Low Power Wireless solutions
    - IAR Embedded Workbench for Arm V.8.11.1 – V6.10.2 (legacy editions I imagine)
  + Check compatibility
    - Using the links, we can check the compatibility info on Tis website with other solutions
    - IAR Embedded Workbench for 8051
* IAR Embedded Workbench for 8051 (<https://www.iar.com/iar-embedded-workbench/#!?architecture=8051>)
  + Overview
    - Shortened time-to-market and simplified development within IOT (don’t care)
    - Single toolbox for development (nice!)
    - “smaller, faster, smarter code for 8051 devices”
    - Comprehensive debugging methods
    - We have downloaded a free trail (in file as “EW8051-10101-Autorun.exe”
      * Evaluation license is free of charge and allows us to try the development environment and evaluate its efficiency and ease of use
      * When you download it you must be asked to register to get evaluation license
      * Two options: we can either utilize a 30-day time-limited but fully functional license or a size-limited kickstart license without any time limit (the former is my suggestion)
        + First Option:

30-day time limitation

Source code for runtime libraries not included

No MISRA C support

MISRA C is development guidelines for C (I think this means they don’t provide auto-sense which sucks)

Limited tech support

Must not be used for product development or other commercial use (fine)

* + - * + Second Option:

4-Kbyte code size limitation

For reference my simple Python project I was working on yesterday is 1Kb (although there is a lot of overhead being used with PyCharm) and is only 50 lines (WTF) so..

Source does for runtime libraries is not included

No MISRA C support

Limited tech support

* + Features
    - Integrated devo environment (compiler, assembler, linker, debugger) (cool! screw 3rd party)
    - IDE Tools (Editor, Project manager, Library tools)
      * Integrated development environment
      * Configuration files for devices from many different manufacturers
      * Run-time libraries with complete source code
      * Linker and library tools
      * Example projects for code and templates (yay!)
      * User and reference guides in PDF format
      * Context-sensitive online help
      * Fully integrated static analysis with the add-on C-STAT
        + Autosense basically
    - Build tools (IAR C/C++ Compiler, Assembler, Linker)
      * Highly optimizing C and C++ for 8051
      * Relocating 8051 assembler (?)
      * Support for DATA, IDATA, XDATA, PDATA, and BDATA
        + Not sure, need to read further
      * Support for multiple DPTR in compiler and libraries
        + Special data pointer for debugging only accessible by the user
      * Bitwise addressing for SFRs
        + Some kind of register I am sure
      * 32 virtual registers (not sure, don’t ask)
    - C-SPY Debugger (Simulator, Hardware debugging (yay!), RTOS plugins)
      * 8051 simulator
      * JTAG drivers
      * ROM-monitor
      * Source code and project
      * 3rd party plugins
* SmartRF Studio (in file as smartrftm\_studio-2.7.0)
  + <http://www.ti.com/tool/smartrftm-studio>
  + Windows application used to evaluate and configure low power RF devices from TI
  + Help to design RF systems in the early stage of the design process
  + Useful for generation of configuration register values and commands, and for practical testing and debugging of RF system
  + We will be using SmartRF Studio 7
  + Features:
    - Link tests
    - Send and receive packets between nodes
    - Antenna and radiation tests, set radio in continuous wave TX and RX states
      * “frequency” = the output, receive, RX, what you want to listen to, etc.
      * “input” = transmit, TX, what you transmit on to activate a repeater, etc.
    - Set of recommended/typical register settings for all devices
    - Read or write individual commands to RF registers
    - Execute individual commands to control the radio
    - Detailed information about bit fields for each register or command
    - Save/Load device config from dat file
    - Exports register settings and command args to a user definable format
    - Exports radio config code
    - Allows for custom GPIO config
    - Communicates with eval boards over USB via debug probe or eval board
  + Technical documents located in folder under folder (Smart RF Studio) including user guides
* SmartRF Flash Programmer 1/2 (located in development tools)
  + http://www.ti.com/tool/flash-programmer
  + Program flash memory in TI ARM based Low Power RF wireless MCUs over debug and serial interfaces
  + Features:
    - Programming of SW images
    - Programming/updating firmware and bootloader on Eval Boards USB MCU
    - Append software image to existing
    - Read out image
    - Verify software image
* SmartRF Protocl Packet Sniffer
  + Software application that can display and store radio packets captured by a listening RF device
  + Features:
    - Bluetooth low energy networks
    - ZigBee and IEE 802.15.4 networks
    - RF4CE Networks
    - SimliciTI networks (ours)
    - Generic protocols
    - Save/open file with captured packets
    - Select fields to display and hide
    - Filtering
    - Accurate timesampling
    - Address book of all known nodes
    - Packeting timeline
    - UDP socket forwading for real time monitoring of packets using custom tools
* Technical background
  + The Intel MCS-51 (termed 8051) is a Harvard Architecture, CISC (complex instruction set computer), single chip microcontroller series developed by Intel in 1980 for use in embedded systems



* + Harvard architecture is a computer architecture with physically sperate storage and single pathways for instructions and data.
* Finally: CC1110 and CC1111 Development Kit
  + Right out of the box the user can do range testing (PER testing) between two development boards with or without using a PC
  + …\