The FOX-1 Software Build Environment On Linux

Burns Fisher, W2BFJ

Build Environment…also called a "Toolchain"

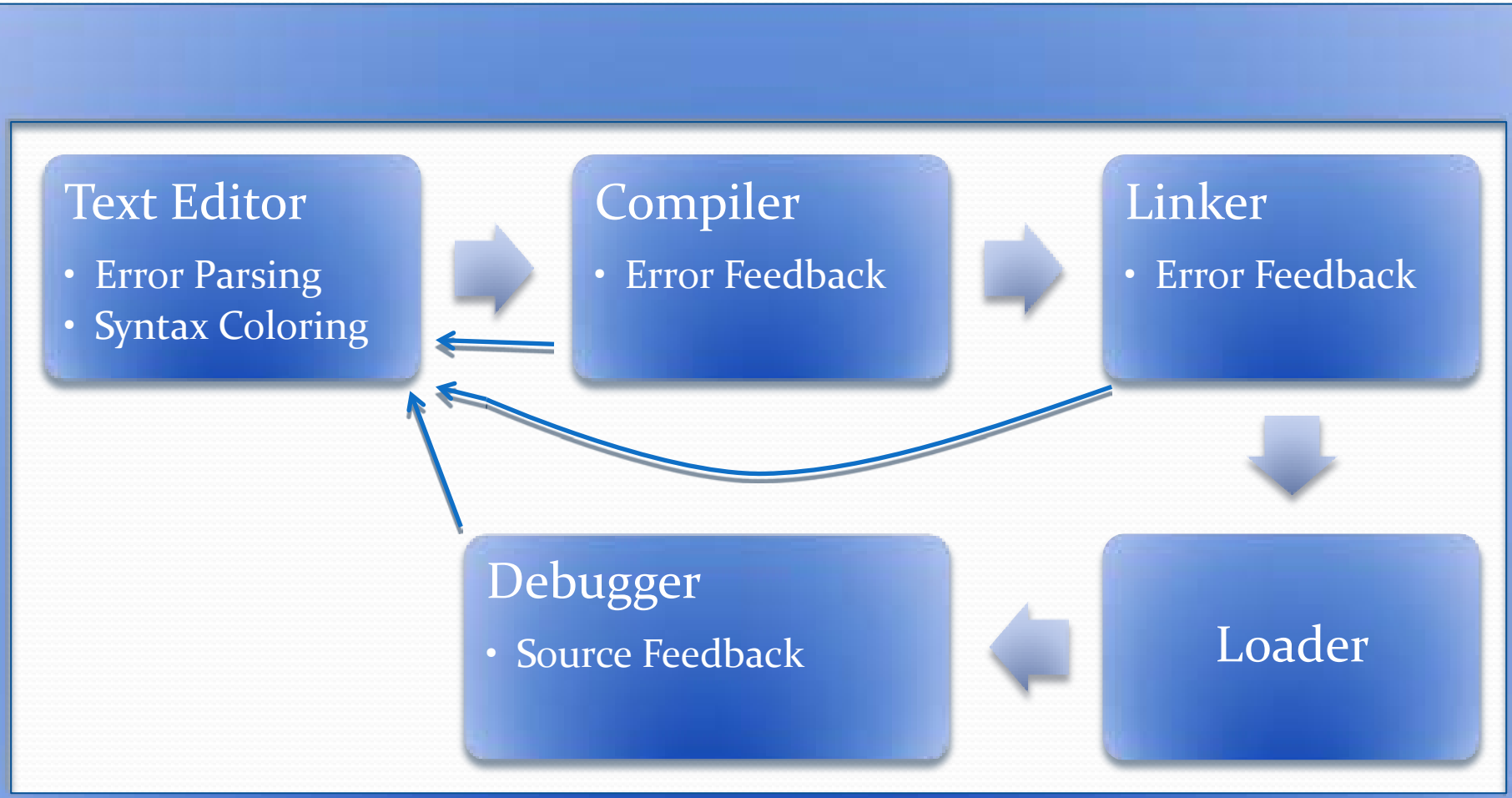
Nope, that's a chain tool

What *IS* In A Toolchain?

- All the software tools you need to build a software product
- You might have separately:
  - Text Editor
  - Compiler
  - Linker
  - Loader
  - Debugger

Or All Together in an IDE

Integrated Devo Environment



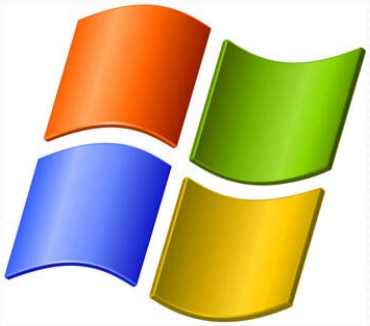
One more piece: Cross-building

- Using a different computer or OS to build a program
  - Ex: Use a Mac to build an iPhone program
- HOST system is what you build on
  - In our case, Linux on an X64
- TARGET system is what you run it on
  - In this case an STM32L1xx Discovery board
- Adds a new requirement to the loader and debugger
  - Must 'talk' via an external connection (network, serial line, etc)

Fox-1 Host Requirements

- What we need in a toolchain and host:
  - Easy to build for AMSAT volunteers
  - Easy to build other partners (whose payload gives us a ride, for example!)
  - Inexpensive or readily available
  - No restrictions on the use of the generated product

How about Windows?



- Frequently already available
- Modest cost (very inexpensive to educational partners)
- Many commercial toolchains are Windows-based
- BUT: At least some part of available toolchains are
  - Very expensive
  - Or restricted (crippleware) (We had a bad experience!)
  - Or restricted (licensing)
  - Or requires work to setup and integrate

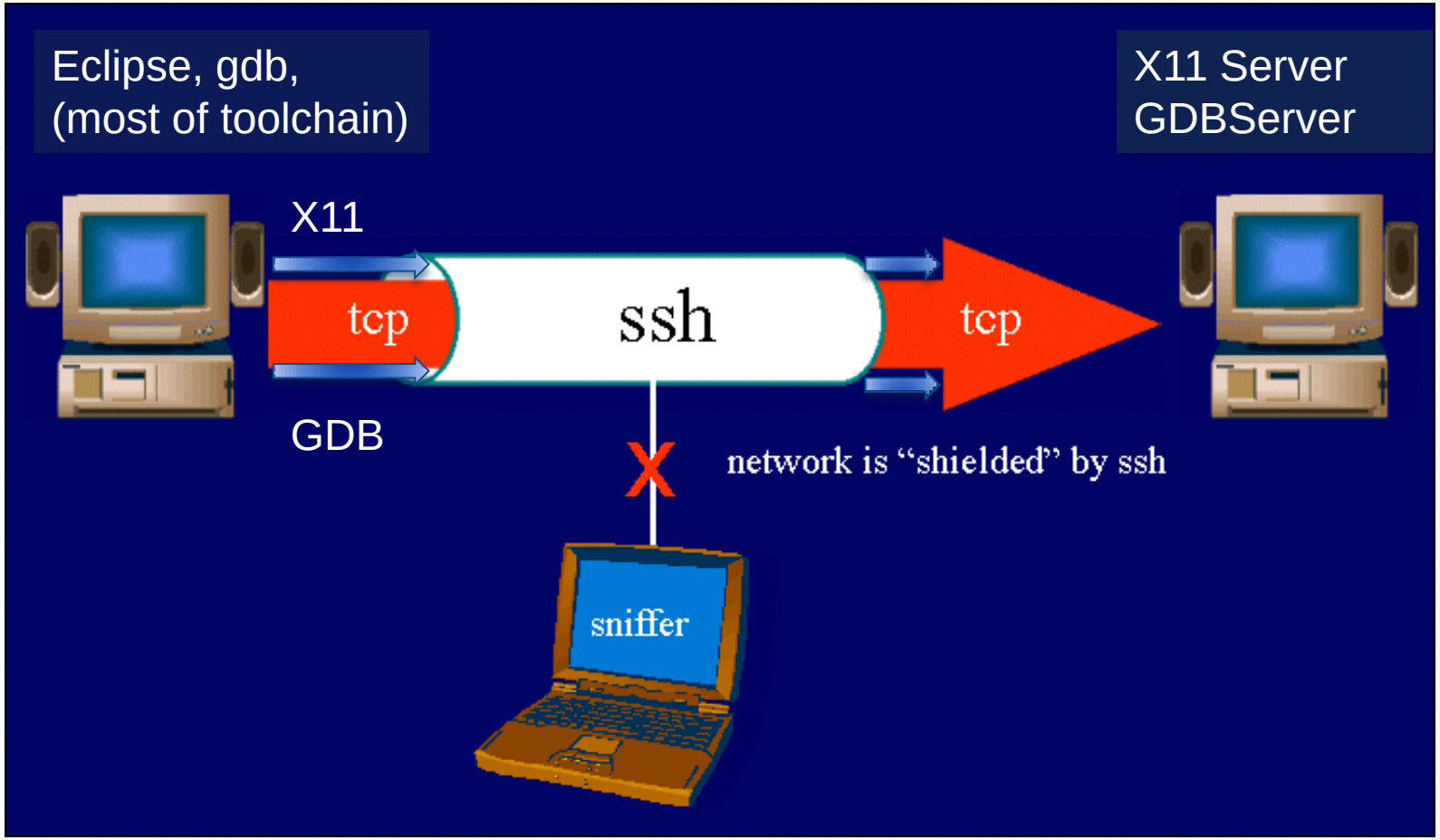
Why Linux?

- Both Operating System and Tools:
  - Are Free (as in beer)
  - Are Free (as in speech)
  - Have copyright/licensing that is understood
  - Have generally good quality

- But…
  - Linux is less familiar than Windows to many
  - Requires work to setup and integrate tools

Another Cool Thing: Remote Operation

- Eclipse, compilers, linker gdb run on a remote Linux system.
- SSH (Secure Shell) to a local system (Windows or Linux) -- creates a secure “pipeline” between two systems
  - Pipeline carries X11 graphics from Eclipse to local system X11 display server
  - Pipeline carries gdb network protocol to a local gdbserver (where the Discovery card is)
- (I tried it only once)


Eclipse, gdb,
(most of toolchain)
X11 Server
GDBServer
X11
tcp
ssh
tcp
GDB
X
network is "shielded" by ssh
sniffer

Current Plan for Fox Development

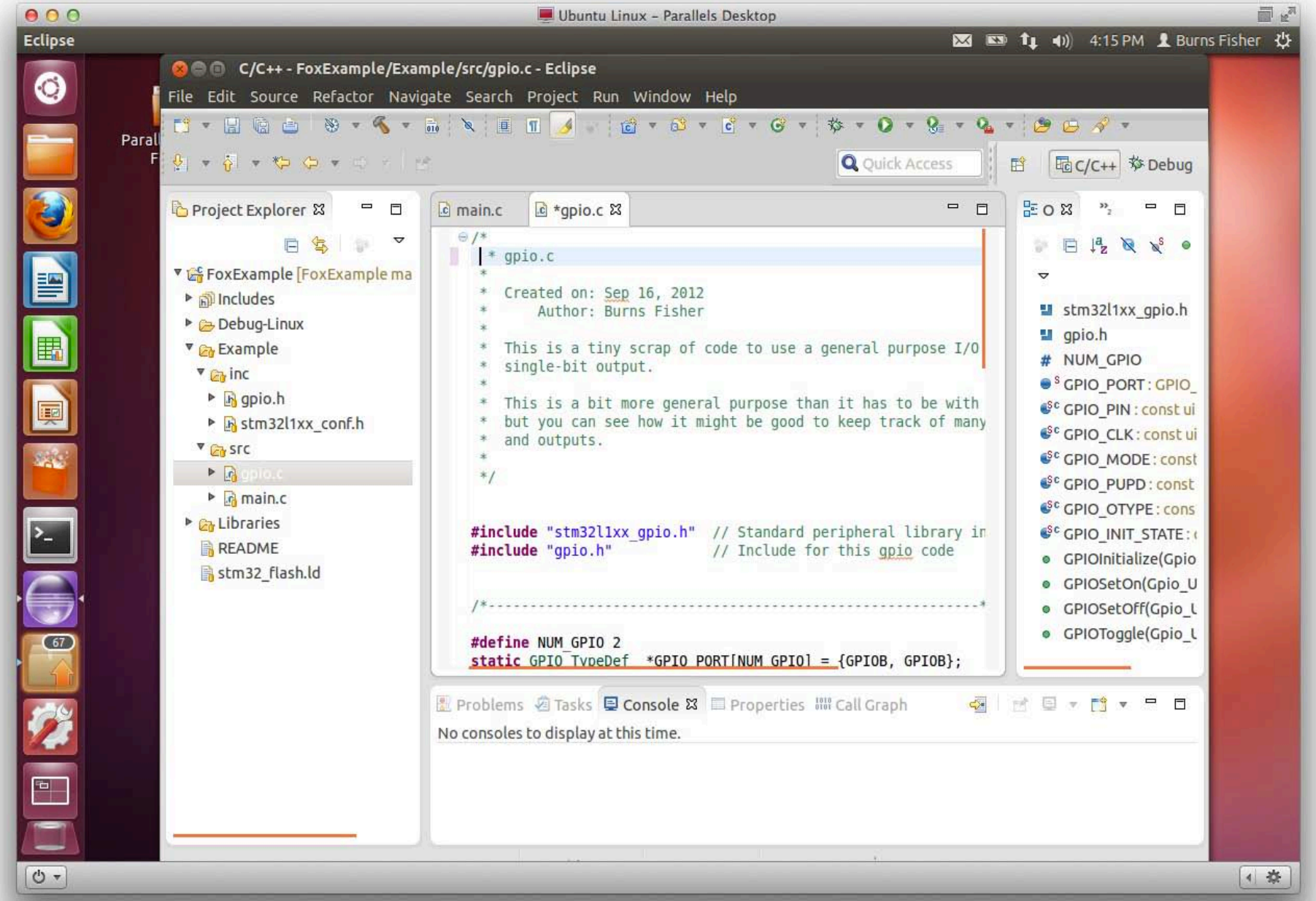
- Developers use either Linux or Windows toolchain
  - Compiler front ends are both GCC; sources compatible
  - Same IDE available on both
- Build and test and do the final build on a single toolchain
  - Which is TBD, but I vote for Linux
- Bill Reed (NX5R) researched & wrote Windows doc
- Burns (W2BFJ) researched & wrote the Linux doc based on
  - NX5R's Windows document
  - Tool research by Bdale Garbee (KB0G) and Keith Packard (KD7SQG)

The Linux Fox-1 Toolchain

- IDE: Eclipse
  - Open Source, Syntax-aware text editor
  - Error parser (to mark errors on the source)
  - Make file generator
  - Single tool with commands to compiler, linker, debugger
  - git source control plugin
  - Plugins for ARM cross build/debug
- Compilers, Linkers, Debugger/Loader
  - gcc, gdb, others—Open source tools, brought together as "Summon ARM toolchain"
- Network server to connect to STM32L1xx
  - Texane ST-Link utility

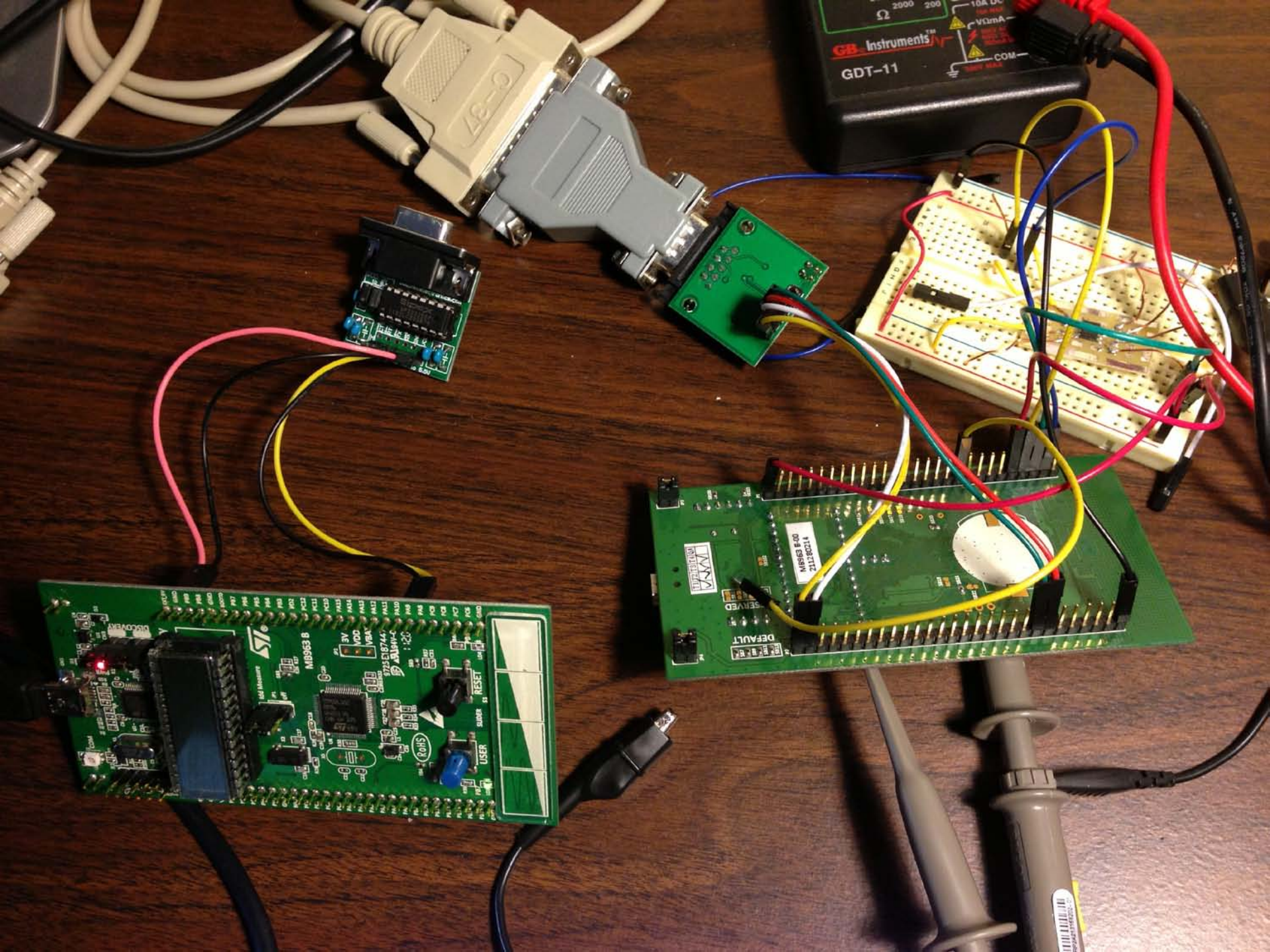
What Does it look like?

- See next slide


Ubuntu Linux – Parallels Desktop
Eclipse
4:15 PM
Burns Fisher
C/C++ - FoxExample/Example/src/gpio.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
C/C++
Debug
Project Explorer
FoxExample [FoxExample ma
Includes
Debug-Linux
Example
inc
gpio.h
stm32l1xx_conf.h
src
gpio.c
main.c
Libraries
README
stm32_flash.ld
main.c
*gpio.c
/*
* gpio.c
*
* Created on: Sep 16, 2012
* Author: Burns Fisher
*
* This is a tiny scrap of code to use a general purpose I/O
* single-bit output.
*
* This is a bit more general purpose than it has to be with
* but you can see how it might be good to keep track of many
* and outputs.
*
*/
#include "stm32l1xx_gpio.h" // Standard peripheral library in
#include "gpio.h" // Include for this gpio code
/*----------*
#define NUM_GPIO 2
static GPIO_TypeDef *GPIO_PORT[NUM_GPIO] = {GPIOB, GPIOB};
stm32l1xx_gpio.h
gpio.h
NUM_GPIO
GPIO_PORT : GPIO_
GPIO_PIN : const ui
GPIO_CLK : const ui
GPIO_MODE : const
GPIO_PUPD : const
GPIO_OTYPE : cons
GPIO_INIT_STATE :
GPIOInitialize(Gpio
GPIOSetOn(Gpio_U
GPIOSetOff(Gpio_L
GPIOToggle(Gpio_L
Problems
Tasks
Console
Properties
Call Graph
No consoles to display at this time.

But That's A Screenshot!

I want to see hardware!


GB Instruments
GDT-11
COM
DEFAULT
RESET
USER

But That's a PICTURE of Hardware

- See W2BFJ (that's me) for the real thing

I Want To Try This Stuff!

- Buy STM32L1xx Discovery (**Not** EVAL)
  - $10 to $15 at Mouser, Digikey, many others
- W2BFJ Symposium Paper tells how to build toolchain
- Symposium paper includes pointer to an example project on github.com (free git software repository)
- Symposium paper suggests a few additional but very small projects—invent your own!
- If you want to join the Fox-1 team, see Tony Monteiro, AA2X

THANKS!

- Burns Fisher, W2BFJ, w2bfj@amsat.org