Visual Studio Code® and Yocto Project®

Rob Woolley, Wind River

Yocto Project *Virtual* Summit Europe, October 29-30, 2020

Outline

- **Introduction**
- **What is Visual Studio Code?**
  - Demo 1: Bitbake Extension
  - Demo 2: Debugging BitBake
- **Developing with SDKs**
  - Demo 3: SDK Development
- **Remote Development**
  - Demo 4: Development with WSL2
- **Conclusion**

Why?

- Why do all the junior engineers use VSCode?
- Can we improve working remotely during COVID-19?
- How may we adopt modern, cloud-native tools?
- Can we develop apps using SDKs?
- Does the YP community want tooling for VS Code?

What is Visual Studio Code?

- Available for Linux, Windows, and macOS
- Based on Electron (Chromium and Node.js so you can build your app with HTML, CSS, and JavaScript)
- Code completion, debugging, refactoring, navigation and more
- Many useful extensions for embedded development in the Visual Studio Marketplace
- New Linux ARMv7 and ARM64 support for Raspberry Pi and Chromebooks

VSCode for Tool Developers

- Language Server and Debug Adapter architecture
- Written in Typescript transpiled to Javascript
- JSON configuration with user, workspace, and project-level settings
- MIT licensed source code with a binary available under a Microsoft license
- VSCodium with Open-VSX.org marketplace
- Eclipse Theia uses the same extension model and can share extensions

https://code.visualstudio.com/api/language-extensions/language-server-extension-guide#testing-the-language-server

Demo 1: BitBake Extension

VS Code and Yocto Project - BitBake Extension

- https://www.youtube.com/watch?v=bkxiem10YNM

Demo 2: Debugging BitBake

VS Code and Yocto Project - Debugging BitBake

- https://www.youtube.com/watch?v=WhRTVwp6eBQ

Developing with SDKs

- **Set up Poky**

```
git clone git://git.yoctoproject.org/poky; cd poky
git clone git://git.yoctoproject.org/meta-mingw
. oe-init-build-env qemu-aarch64
bitbake-layers add-layer $(dirname $PWD)/meta-mingw/
echo 'EXTRA_IMAGE_FEATURES += " tools-debug debug-tweaks tools-sdk"' >> conf/local.conf
bitbake core-image-minimal
```

- **Build the SDKs for Linux and Windows**

```
echo 'SDKMACHINE = "x86_64"' >> conf/local.conf
bitbake core-image-minimal -c populate_sdk
echo 'SDKMACHINE = "x86_64-mingw32"' >> conf/local.conf
bitbake core-image-minimal -c populate_sdk
```

VS Code Configuration Files

- **VSCode has no understanding of Linux SDKs, these JSON configuration files add support:**
- **settings.json** - Set folder configuration settings (eg linuxsdk.sdkroot and linuxsdk.targettriplet)
- **c_cpp_properties.json** - Set IntelliSense mode, include path, and more for code completion using the SDK
- **tasks.json** - Set build tasks for compiling files or running GNU make
- **launch.json** - Launch configurations for running or debugging an application

See examples here: https://github.com/Wind-River/vscode-wrlinux

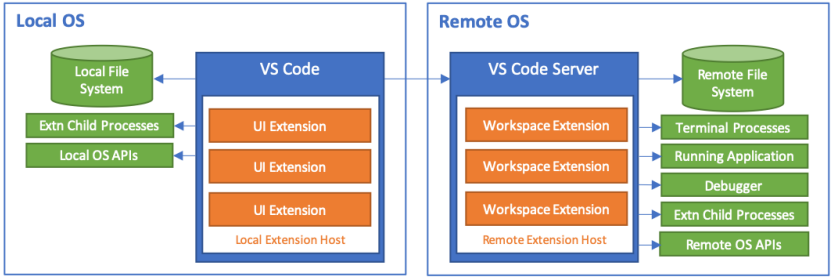
Demo 3: SDK Development

VS Code and Yocto Project - SDK Development

- https://www.youtube.com/watch?v=kQXWCRzoUbk

Remote Development

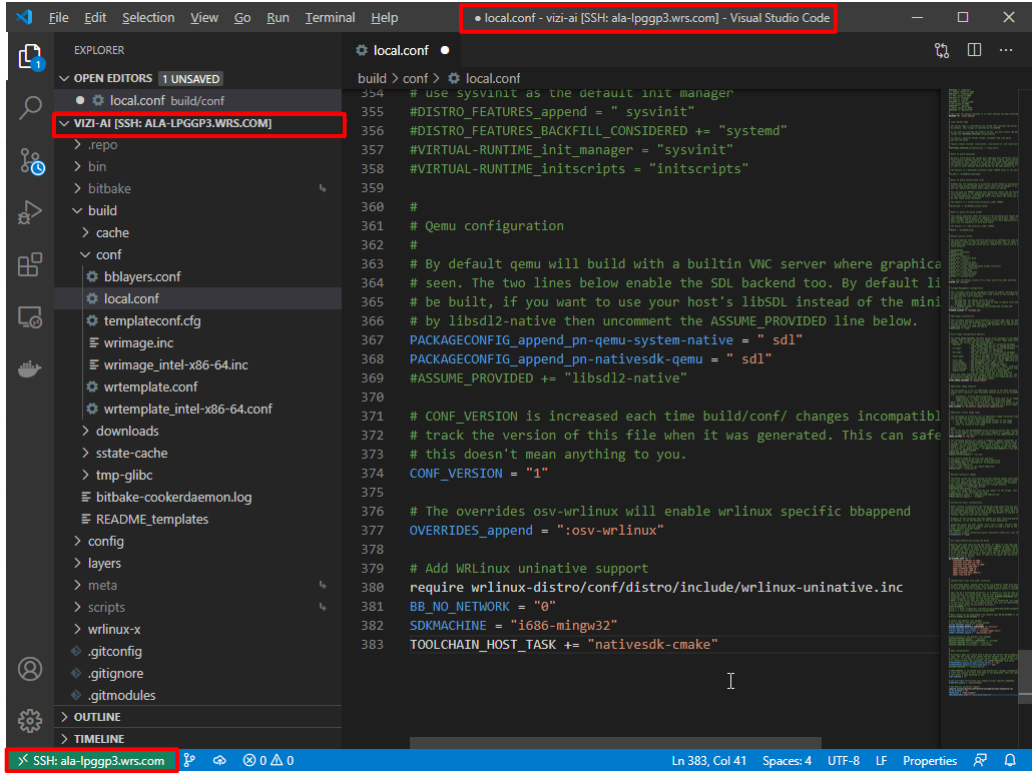
- **Remotely navigate your project and use extensions like you were working locally**
- **Take advantage of cloud-native technologies like containers and virtual machines**
  - **Remote Containers**: Attach to Docker containers for a reproducible, sandboxed development environment
  - **Remote SSH**: Connect to build servers or cloud instances for additional compute power
  - **Remote WSL**: Leverage Windows Subsystem for Linux to give you a Linux development environment on your Windows or MacOS laptop



https://code.visualstudio.com/docs/remote/faq

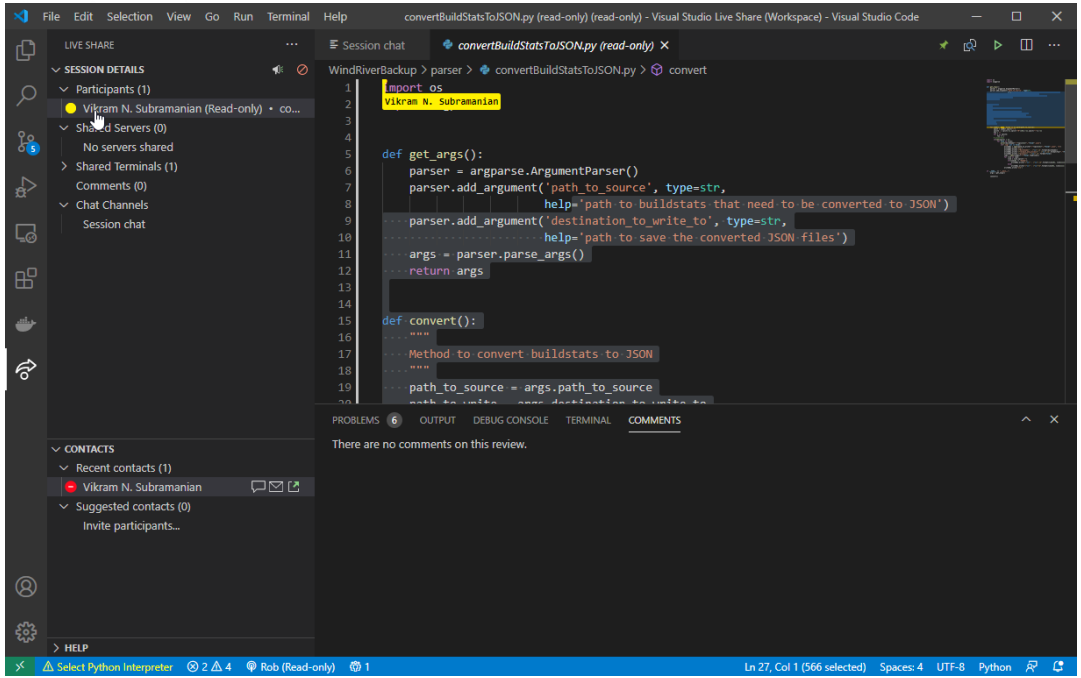
Remote SSH Extension

- Use “Remote-SSH: Connect to Host…” to log into a remote build server
- If you are connecting to the machine for the first time, VSCode automatically installs itself into $HOME/.vscode-server and adds itself to $PATH
- The “Remote” series of extensions allow you to edit your files as well as execute extensions, tasks and launches all remotely.



Cloud-enabled Features

- Development containers (devcontainer.json)
- GitHub Codespaces
- Visual Code Live Share



Demo 4: Development with WSL2

VS Code and Yocto Project - Development with WSL2

- https://www.youtube.com/watch?v=0X7R4Yr7zbs

Future Work

- **This has been a preliminary investigation and much more work remains to be done.**
- **I propose we develop VSCode community extensions for the Yocto Project.**
- **Adopt and maintain the BitBake extension**
  - Extend it to make it aware of recipes VS packages
  - Provide the appropriate identifiers based on context
  - Extend it to provide recipe templates and support for recipetool / devtool
- **Create new extensions for VSCode to talk directly to BitBake using Tinfoil or XMLRPC**
- **Create web-based visualizations for pybootchart and taskexp that can be used in VSCode or a web browser**
- **Enhance the SDK development workflow for application developers (e.g. JSON environment file)**
- **Extend the work on CROPS to leverage "devcontainers" and other cloud-native workflows**

Thanks

- Thanks to Martin Björkström and Eugen Wiens for vscode-bitbake
- Thanks to Josef “The Yocto Jester” Holzmayr, Rudolf Streif, Tim Orling, Joshua Watt, and David Reyna for valuable feedback
- Thanks to Anmolbir Singh Mann for porting my vscode-wrlinux to poky
- Check out https://github.com/Wind-River/vscode-wrlinux for instructions on how to try it at home with Poky (poky-qemux86-64) or Wind River Linux (intel-x86-64, bcm-2xxx-rpi4, x86_64-wrlinuxsdk-mingw32)
- Pre-built SDK and binaries for Intel and Raspberry Pi are available here: https://labs.windriver.com/downloads/wrlinux.html

Thanks for your time






yocto PROJECT

THE LINUX FOUNDATION

What is the Yocto Project® ?

IT'S NOT AN EMBEDDED LINUX DISTRIBUTION,
IT CREATES A CUSTOM ONE FOR YOU.

The Yocto Project (YP) is an open source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture.

The project provides a flexible set of tools and a space where embedded developers worldwide can share technologies, software stacks, configurations, and best practices that can be used to create tailored Linux images for embedded and IOT devices, or anywhere a customized Linux OS is needed.