Practice 3: Introduction to Vxwork 7 environment

Curse 2022/2023

Made by: Mario Pereira Martin

Date: October 16, 2022

1. Objectives

This practice is an introduction to Vxwork 7 SDK environment. This system will be used in the curse: '***Real Time and embedded systems / Concurrent and real time systems***. The topics introduced during this session are the following:

- Installation of Micosoft Visual code.
- Installation of QEMU software.
- Installation of compillers LLVM-14.06
- Installation of Vxworks 7 SDK for windows.
- Installation of Visual code plugin for VxWorks.
- Running an example.

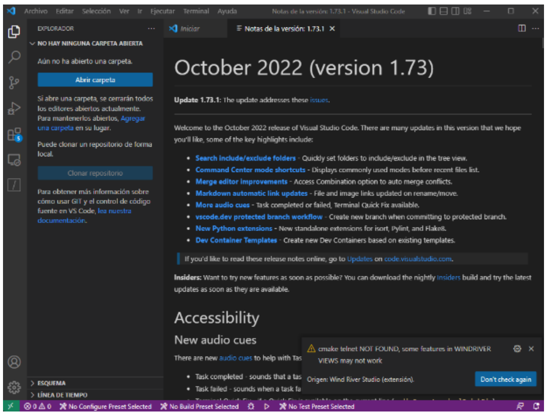
2. Materials

The materials needed for this practice are the following:

- Laptop with Windows operating system (from students or the labs).
- All the software provided in moodle for approach #3.

3. Installing Microsoft Visual Code

Figure 1 shows a top view of the Microsoft visual studio. Download executable file from moodle or the official web site and install it.



Aún no ha abierto una carpeta.

Si abre una carpeta, se cerrarán todos los editores abiertos actualmente. Para mantenerlos abiertos, Agregar una carpeta en su lugar.

Puede clonar un repositorio de forma local.

Para obtener más información sobre cómo usar GIT y el control de código fuente en VS Code, lea nuestra documentación.

October 2022 (version 1.73)

Update 1.73.1: The update addresses these issues.

Welcome to the October 2022 release of Visual Studio Code. There are many updates in this version that we hope you'll like, some of the key highlights include:

- **Search include/exclude folders** - Quickly set folders to include/exclude in the tree view.
- **Command Center mode shortcuts** - Displays commonly used modes before recent files list.
- **Merge editor improvements** - Access Combination option to auto merge conflicts.
- **Markdown automatic link updates** - File and image links updated on rename/move.
- **More audio cues** - Task completed or failed, Terminal Quick Fix available.
- **vscode.dev protected branch workflow** - Create new branch when committing to protected branch.
- **New Python extensions** - New standalone extensions for isort, Pylint, and Flake8.
- **Dev Container Templates** - Create new Dev Containers based on existing templates.

> If you'd like to read these release notes online, go to Updates on code.visualstudio.com.

Insiders: Want to try new features as soon as possible? You can download the nightly Insiders build and try the latest updates as soon as they are available.

Accessibility

New audio cues

There are new audio cues to help with Tas

- Task completed - sounds that a tas
- Task failed - sounds when a task fa

⚠ cmake telnet NOT FOUND, some features in WINDRIVER VIEWS may not work

Origen: Wind River Studio (extensión).

Figure 1. Microsoft Visual Code

[1] More information can be found at https://code.visualstudio.com/download

4. Installing QEMU software

This is an open source machine emulator used by vxwork to run its kernel with the corresponding application code. You can obtain the installer from moodle or from the corresponding web page:

https://www.qemu.org/

Follow all the screen instructions an install the executable file in your laptop.

5. Installing the LLVM compillers.

This is a set of compillers used con Vxworks 7 SDK to compile the application with the kernel. You can obtain the installer from moodle.

Follow all the instructions in the screen and select the option to introduce the path for all users.

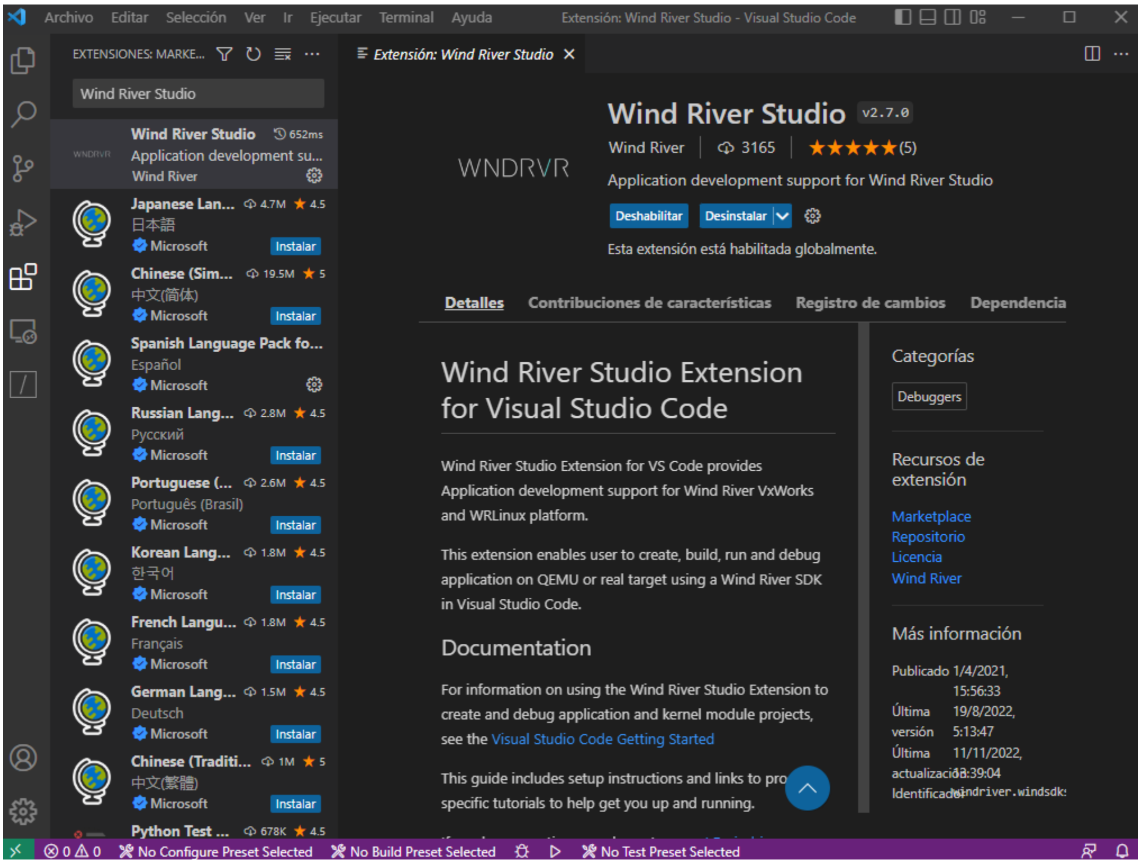
6. Installing the plugin for Visual Studio Code.

Visual Studio (VS) Code is an extensible integrated development environment (IDE) that is ideal for developing software for embedded systems. With Visual Code you can:

- Code, develop and debug single and multithreaded user space applications for VxWorks and Wind River Linux.
- Create downloadable kernel modules for use with VxWorks.
- Create Real Time Process (RTP) executable applications for VxWorks.
- Debug applications locally with a Wind River software development kit (SDK)
- Launch virtual QEMU sessions to aide in debugging and development.

Student must read a guide pdf provide in moodle to know all the details about this framework. The Wind River Studio Extension for VS Code provides support for developing user space applications and kernel modules with the Wind River Linux and VxWorks SDKs.

You can add the Wind River Studio extension to VS Code directly from the Visual Studio Marketplace. In extension bottom from visual code, search ‘Wind River Studio ’ and install it.



Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
Extensión: Wind River Studio - Visual Studio Code
EXTENSIONES: MARKE...
Wind River Studio
Extensión: Wind River Studio
Wind River Studio 652ms
Application development su...
Wind River
Japanese Lan... 4.7M ★ 4.5
日本語
Microsoft
Instalar
Chinese (Sim... 19.5M ★ 5
中文(简体)
Microsoft
Instalar
Spanish Language Pack fo...
Español
Microsoft
Russian Lang... 2.8M ★ 4.5
Русский
Microsoft
Instalar
Portuguese (... 2.6M ★ 4.5
Português (Brasil)
Microsoft
Instalar
Korean Lang... 1.8M ★ 4.5
한국어
Microsoft
Instalar
French Langu... 1.8M ★ 4.5
Français
Microsoft
Instalar
German Lang... 1.5M ★ 4.5
Deutsch
Microsoft
Instalar
Chinese (Traditi... 1M ★ 5
中文(繁體)
Microsoft
Instalar
Python Test ... 678K ★ 4.5
WNDRVR
Deshabilitar
Desinstalar
Detalles
Contribuciones de características
Registro de cambios
Dependencia
⊗ 0 ⚠ 0
No Configure Preset Selected
No Build Preset Selected
No Test Preset Selected

Wind River Studio v2.7.0

Wind River | 3165 | ★★★★★(5)

Application development support for Wind River Studio

Esta extensión está habilitada globalmente.

Wind River Studio Extension for Visual Studio Code

Wind River Studio Extension for VS Code provides Application development support for Wind River VxWorks and WRLinux platform.

This extension enables user to create, build, run and debug application on QEMU or real target using a Wind River SDK in Visual Studio Code.

Documentation

For information on using the Wind River Studio Extension to create and debug application and kernel module projects, see the Visual Studio Code Getting Started

This guide includes setup instructions and links to pro specific tutorials to help get you up and running.

Categorías

Debuggers

Recursos de extensión

Marketplace
Repositorio
Licencia
Wind River

Más información

Publicado 1/4/2021, 15:56:33
Última versión 19/8/2022, 5:13:47
Última actualización 11/11/2022, 13:39:04
Identificador windriver.windsdk

Figure 2. Wind River extension

7. Installing the Vxworks 7 SDK

Download the file `wrsdk-vxworks7-win-qemu-1.10.zip` from moodle, decompress and extract in a known folder in C root. For example: VxWorks7SDK

It needs a lot of time to decompress and move the folder, so don't worry about it.

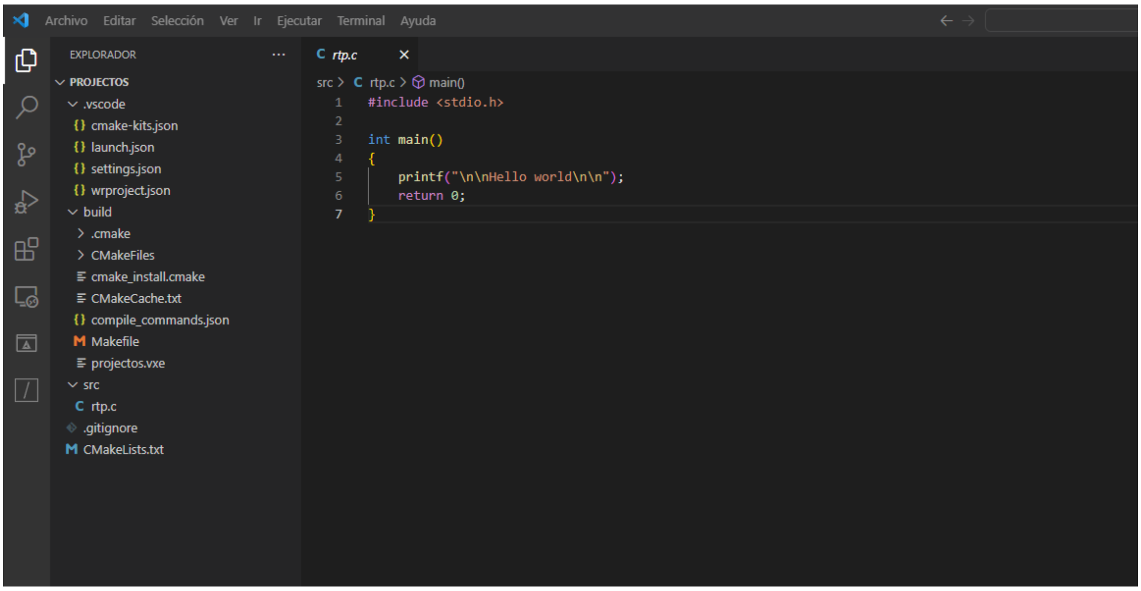
8. Setting the plugin and run our first application.

The VS Code extension need some configuration to work. We are going to divide all the setting needed in the following steps:

✓ **Step 1.**In Wind River extension, in the resources windows, push on add and choose 'Add SDK'. Then choose 'Add SDK from Local SDK Folder' and search the folder created when you decompress the SDK zip file. Note: You can select other option to use the zip file directly.

✓ **Step 2**. Select the Target. Go to resources and select ‘Add’, then ‘add target’ and then qemu. Go to ‘tool’ folder in the sdk folder. Then go to qemu and shoose ‘qemu-system-x86_64.exe’.

✓ **Step 3**. Create a new project from ‘explorador’ tab. Right click in ‘projectos’ and go to ‘New WR application project’. Choose an RTP project. The classical hello world project.



```
#include <stdio.h>

int main()
{
    printf("\n\nHello world\n\n");
    return 0;
}
```

Figure 3 WR Project

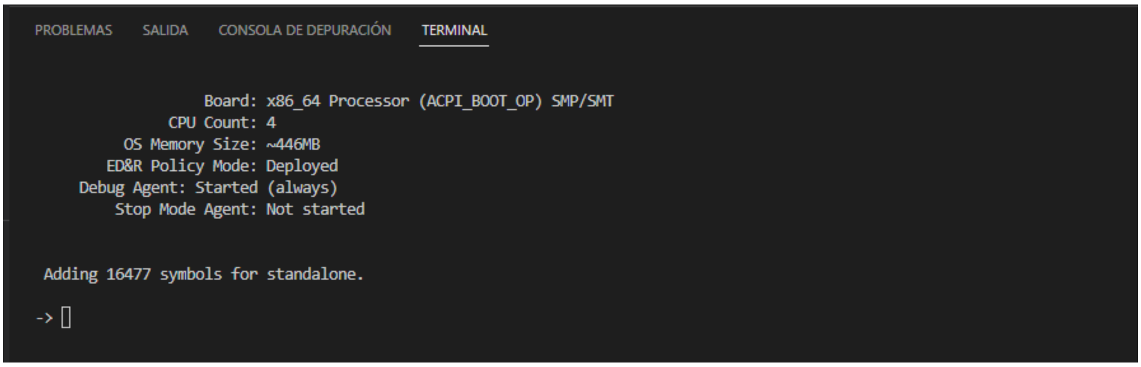
Then choose ‘Add WR Launch configuration’, then ‘dbg’ and then the target that you previously created. This is all that we need to work with the sdk.

6. Working with the SDK

At this time, the SDK is ready to work. The first step is to build the project. Right click and go to ‘Build WR application project’. If all is right, in the extension, in the application project appears a folder binary with a file called ‘projectos.vxe’. This is the executable file to run in Vxworks.

Now we are going to run the compiled file:

1. Run the terminal with the kernel of Vxworks 7. Go to Resources and in ‘Target connections’ do a right click in your target and select ‘Launch simulation’.

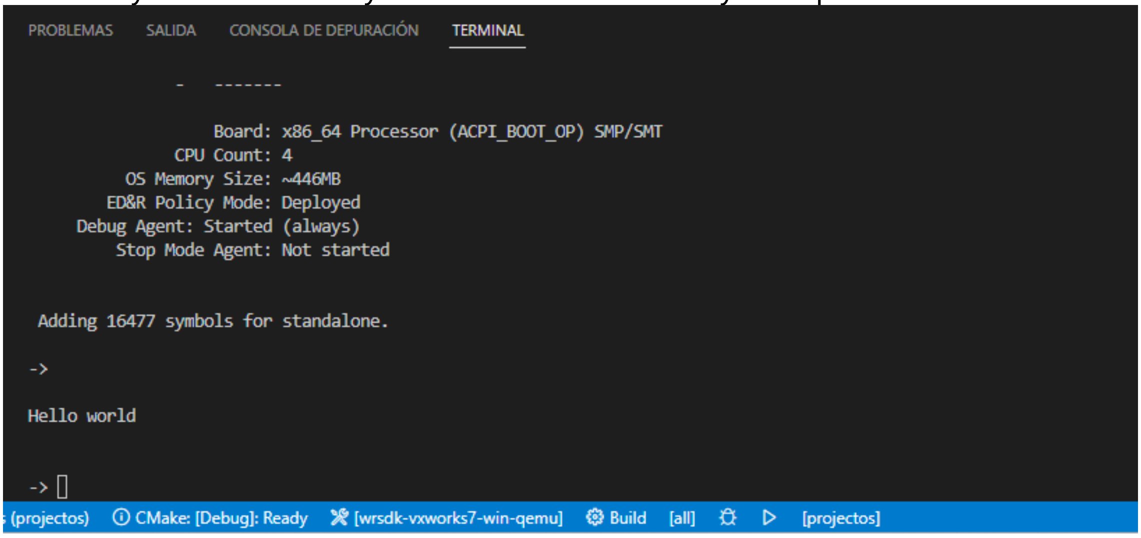


```
                Board: x86_64 Processor (ACPI_BOOT_OP) SMP/SMT
            CPU Count: 4
       OS Memory Size: ~446MB
     ED&R Policy Mode: Deployed
  Debug Agent: Started (always)
      Stop Mode Agent: Not started


 Adding 16477 symbols for standalone.

-> ▯
```

Figure 4 Running target

Then go to ‘Ejecucion y depuracion’ and select ‘projectos.vxe_gbd_qemu_vxworks’ and run it.

In main.c you can watch all your code and introduce any breakpoint to check it.



```
                 -   -------

                Board: x86_64 Processor (ACPI_BOOT_OP) SMP/SMT
            CPU Count: 4
       OS Memory Size: ~446MB
     ED&R Policy Mode: Deployed
  Debug Agent: Started (always)
      Stop Mode Agent: Not started


 Adding 16477 symbols for standalone.

->

Hello world

-> ▯
```

Figure 5 RTP application running

Exercises

1. Modify the rtp.c file and execute the following code:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <sched.h>

void* a(void* ptr) {
    for (int i=0; i< 10; i++){
        printf("1");
        printf("2");
        printf("3");
        printf("4");
    }
    return NULL;
}

void* b(void* ptr) {
    for (int i=0; i< 10; i++){
        printf("a");
        printf("b");
        printf("c");
        printf("d");
    }
    return NULL;
}

void* c(void* ptr) {
    for (int i=0; i< 10; i++){
        printf("5");
        printf("6");
        printf("7");
        printf("8");
    }
    return NULL;
}

int main() {
    pthread_attr_t attr1, attr2, attr3;

    pthread_attr_init(&attr1);
    pthread_attr_init(&attr2);
    pthread_attr_init(&attr3);

    pthread_t t1, t2, t3;

    int rVal = pthread_attr_setschedpolicy(&attr1, SCHED_RR);
    if (rVal != 0) {
    // Failed to set the desired scheduler policy.
    perror("pthread_attr_setschedpolicy(&attr1, SCHED_RR)");
    exit(1);
    }
    pthread_create(&t1, &attr1, a, NULL);

    pthread_attr_setschedpolicy(&attr2, SCHED_RR);
    pthread_create(&t2, &attr2, b, NULL);

    pthread_attr_setschedpolicy(&attr3, SCHED_RR);
    pthread_create(&t3, &attr3, c, NULL);

    pthread_join(t1, NULL);
    pthread_join(t2, NULL);
    pthread_join(t3, NULL);

    pthread_exit(0);
    return 0;
}
```

   a. Check and explain the outputs.
2. Introduce semaphores to obtain the outputs related to this situations:
   a. 2 tasks in round robin with priorities 40 and other with priority 50 in FIFO. Run and explain if something changes and why (if necessary).
   b. 3 tasks in Round-Robin and only tasks a and b print its values. (Use semaphores, all the tasks must be created).
3. Create a project to emulate the Reader and writers problem explained in slides 57-64. Use 5 readers and 3 writers in round-robin with the same priorities. Choose the best solution for this problems and explain it.

Homework

Search information about the following topics related to the practice:

1. Reader and writer problem.
2. Dinner’s philosopher problem.

Evaluation

The evaluation of this practice is done taking into account the performance observed during the practice sessions in the laboratory (it means attention, participation, care of laboratory material, involvement in the right development of the actual session and so on) and a ***practice report*** about the following topics:

✓ Results of exercises 1 to 3.

✓ Results of the homework: all the information obtained at home for exercise 4 (see homework).

This ***report*** should be delivered by Moodle.