**********Draft**********

QEMU Ultibo Bare Metal
SimpleC
07/23/21

**********Draft**********

https://en.m.wikipedia.org/wiki/QEMU

QEMU is a hosted virtual machine monitor: it emulates the machine's processor through dynamic binary translation and provides a set of different hardware and device models for the machine, enabling it to run a variety of guest operating systems. It also can be used with Kernel-based Virtual Machine (KVM) to run virtual machines at near-native speed (by taking advantage of hardware extensions such as Intel VT-x). QEMU can also do emulation for user-level processes, allowing applications compiled for one architecture to run on another.[3]

Note : Additional software is needed to run QEMU "sudo apt-get install qemu-system-arm". The following programs are added.

/usr/bin/qemu-img /usr/bin/qemu-nbd /usr/bin/qemu-system-aarch64
/usr/bin/qemu-io /usr/bin/qemu-pr-helper /usr/bin/qemu-system-arm

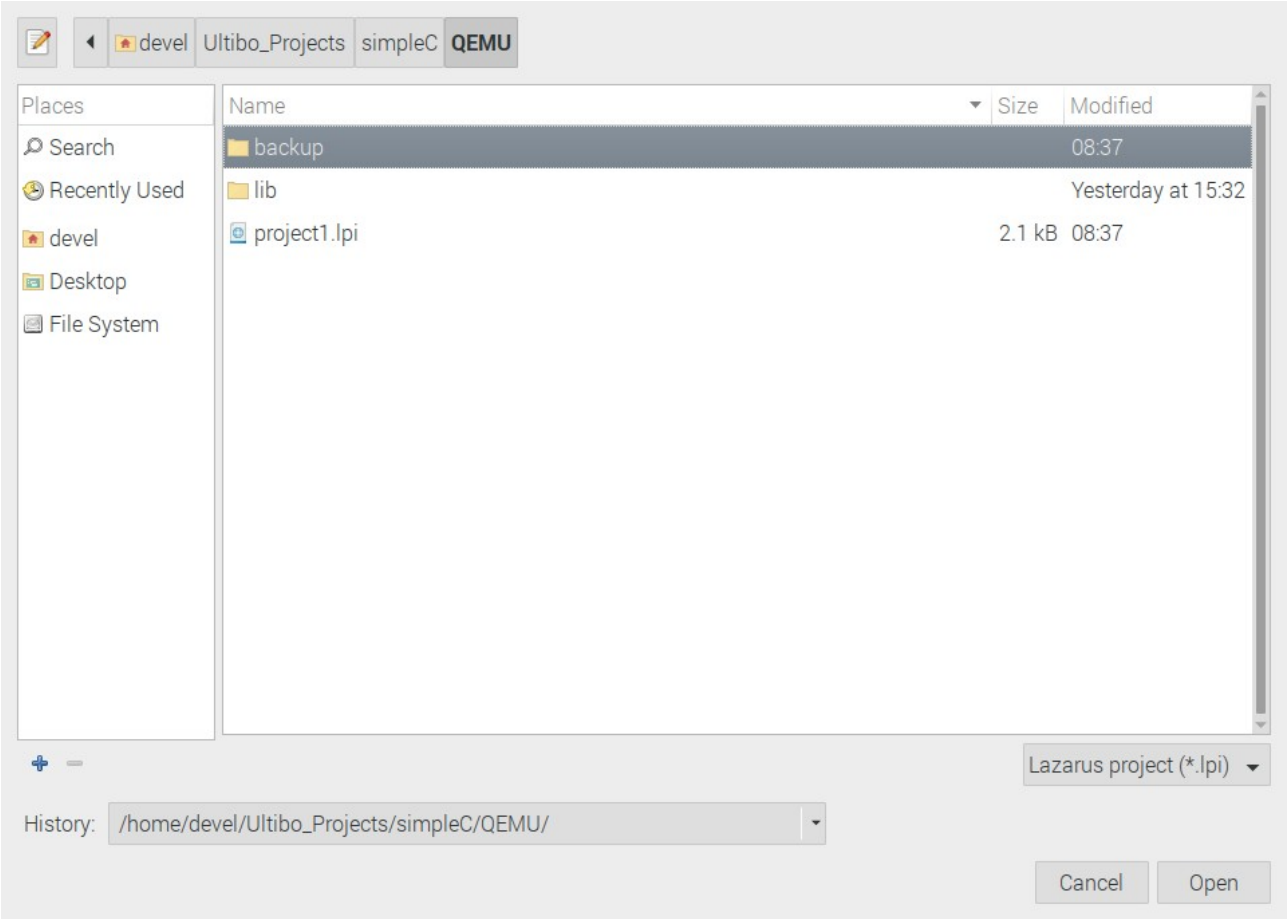
The command line for starting **Lazarus IDE (Ultibo Edition) "~/ultibo/core/lazarus.sh"**

Requires Ultibo RTL installed with ultiboinstaller.sh

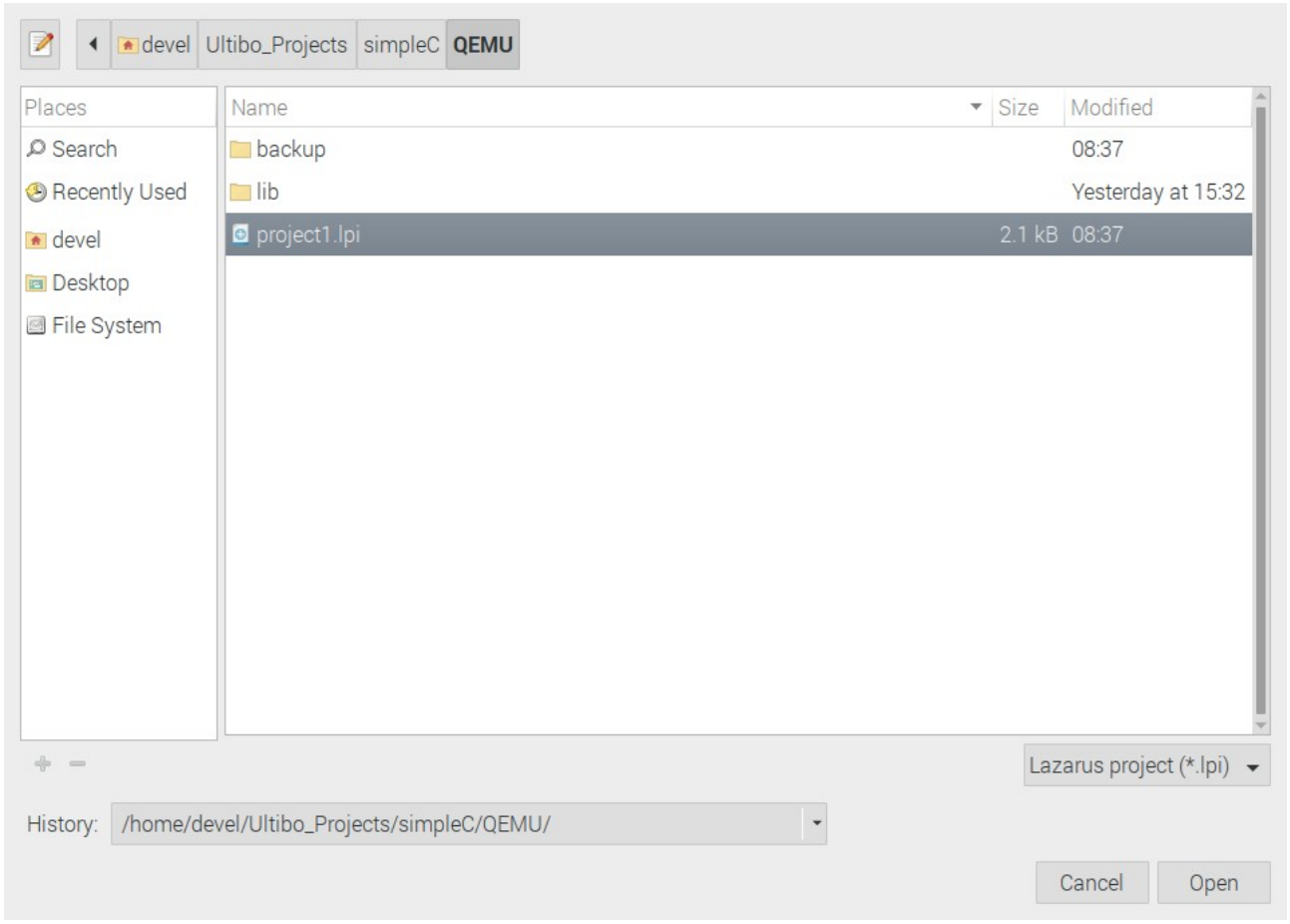
ultibo/core/fpc/source/rtl/ultibo/core/globalconst.pas

```
{Global constants}
const
 {Version constants}
 ULTIBO_RELEASE_DATE           = '21 July 2021';
 ULTIBO_RELEASE_NAME            = 'Beetroot';
 ULTIBO_RELEASE_VERSION         = '2.1.079';
 ULTIBO_RELEASE_VERSION_MAJOR    = 2;
 ULTIBO_RELEASE_VERSION_MINOR    = 1;
 ULTIBO_RELEASE_VERSION_REVISION = 079;
```

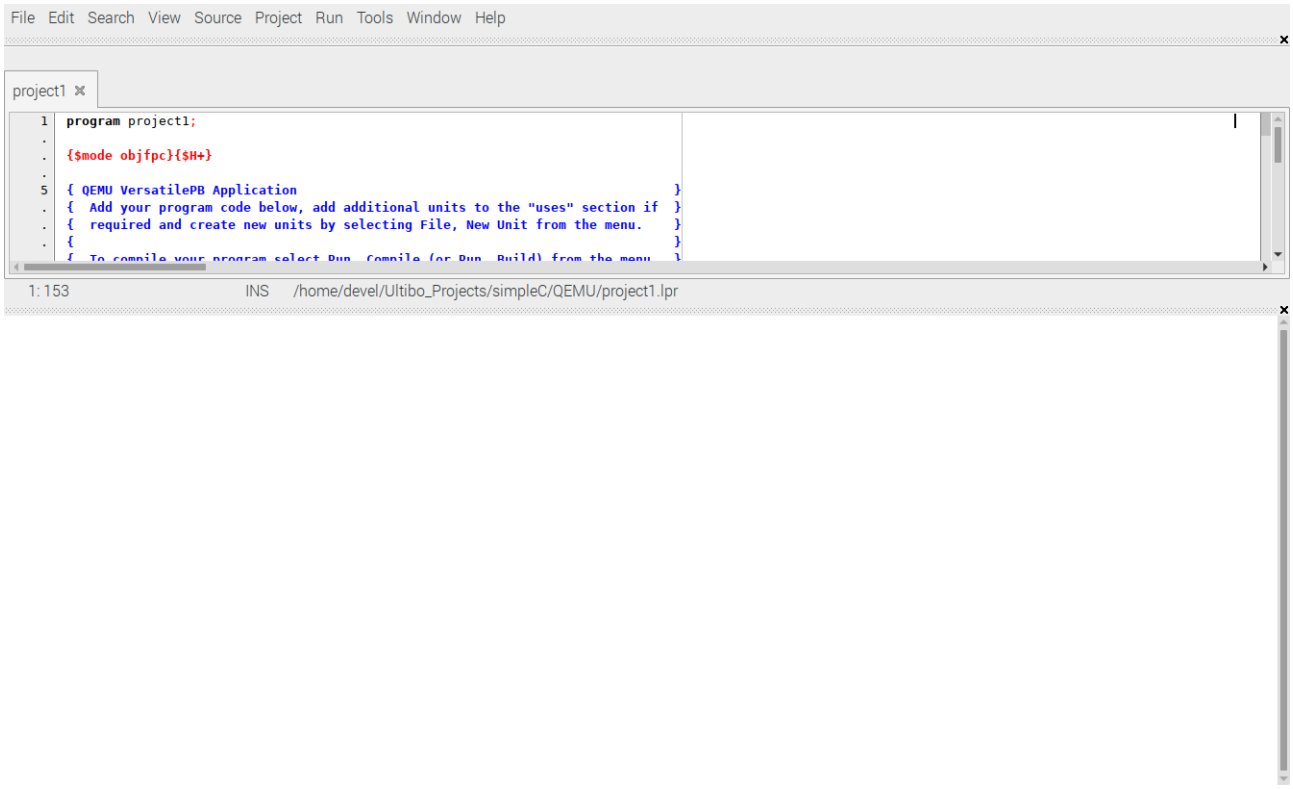
Project/Project Open



Select projet1.lpi



Depress Open



simpleC.c

```
#include <stdio.h>
#include <stdlib.h>
#ifdef Ultibo
void simpleC() {
#else
void main() {
#endif
	int i;
	printf("This is a C program called by Ultibo\n");
	printf("Hello World \n");
	for(i=0;i<10;i++) printf("i= %d \n",i);
}
```

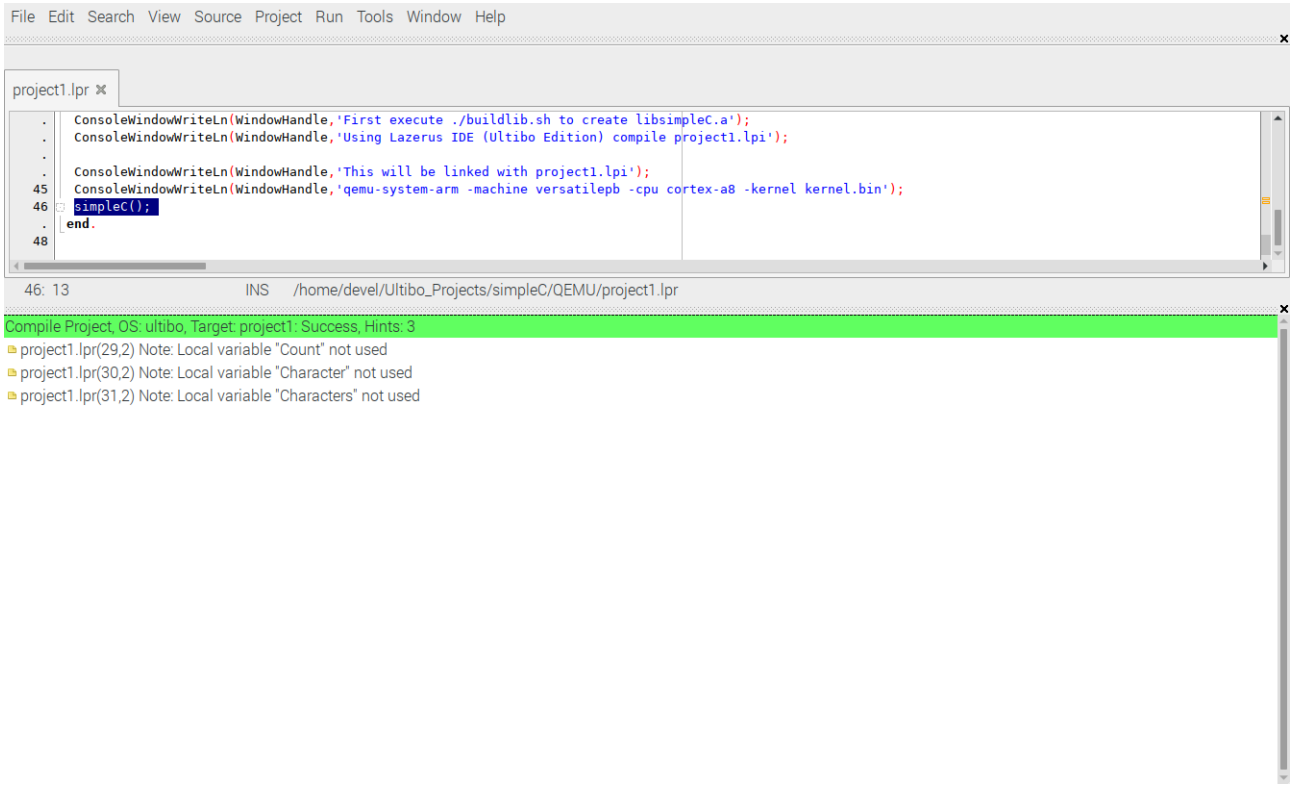
buildlib.sh

```
#!/bin/bash
#export PATH=/home/devel/ultibo/core/fpc/bin:$PATH
rm -f *.o
rm -f libsimpleC.a

arm-none-eabi-gcc -DUltibo -I../include -O3 -mabi=aapcs -marm -
march=armv7-a -mfpu=vfpv3-d16 -mfloat-abi=hard -c simpleC.c -o
ultibo_simpleC.o

arm-none-eabi-ar rcs libsimpleC.a *.o
arm-none-eabi-ar -t libsimpleC.a > libsimpleC_obj.txt
```

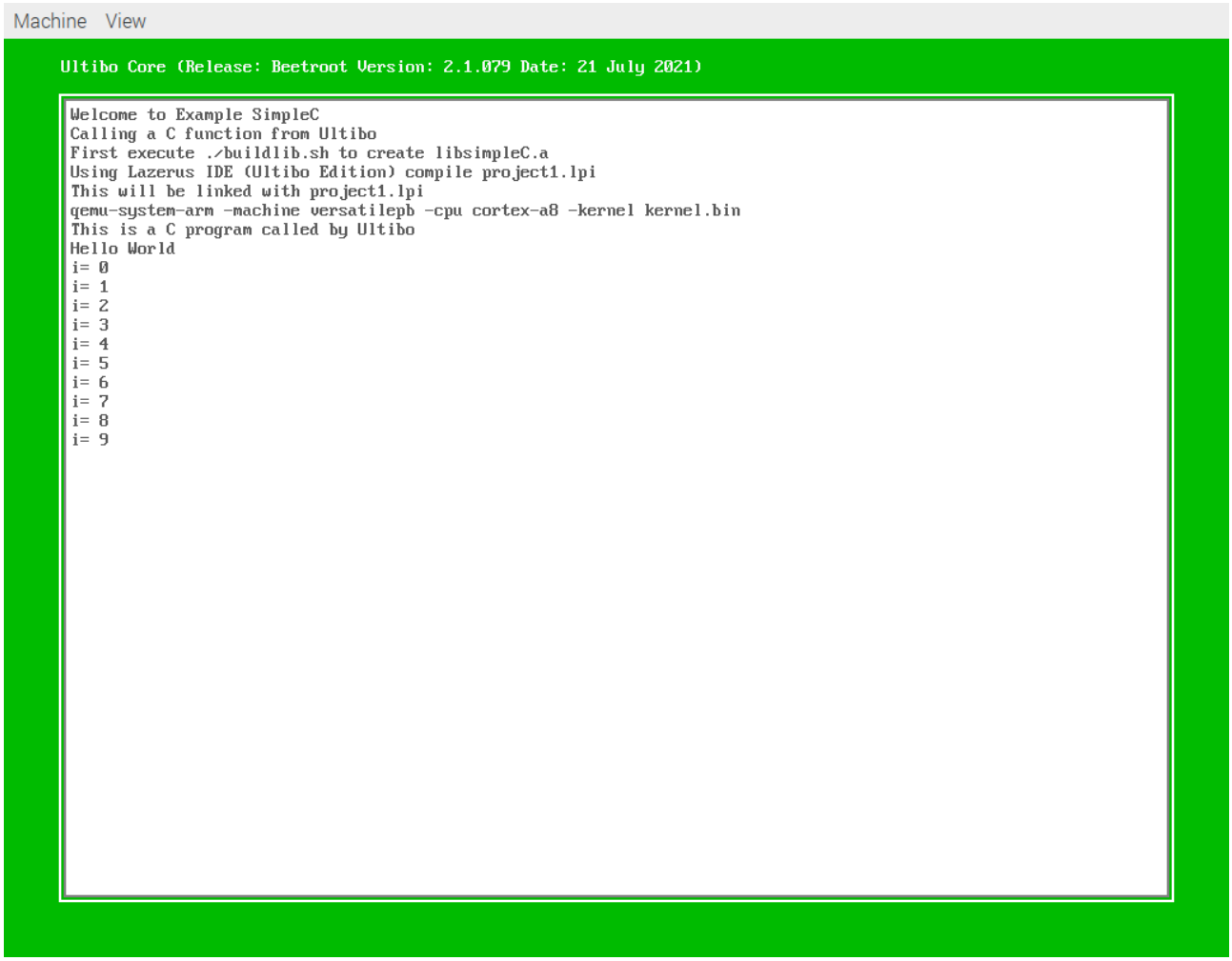
Before Compiling ~/Ultibo_Projects/simpleC/QEMU $ ./buildlib.sh which creates
-rw-r--r-- 1 devel devel 1454 Jul 23 09:18 libsimpleC.a
These are the statements in project1.lpr

```
{$linklib simpleC}
procedure simpleC; cdecl; external 'libsimpleC' name 'simpleC';

simpleC();
```

Run/Compile The kernel.bin is created when the Grean bar appers.



qemu-system-arm -machine versatilepb -cpu cortex-a8 -kernel kernel.bin


Machine View
Ultibo Core (Release: Beetroot Version: 2.1.079 Date: 21 July 2021)
Welcome to Example SimpleC
Calling a C function from Ultibo
First execute ./buildlib.sh to create libsimpleC.a
Using Lazerus IDE (Ultibo Edition) compile project1.lpi
This will be linked with project1.lpi
qemu-system-arm -machine versatilepb -cpu cortex-a8 -kernel kernel.bin
This is a C program called by Ultibo
Hello World
i= 0
i= 1
i= 2
i= 3
i= 4
i= 5
i= 6
i= 7
i= 8
i= 9