

Содержание

Настройка среды QT Creator	. 3
Настройка QT Creator	. 3





Настройка среды QT Creator

ССылка на host: https://disk.yandex.ru/d/ouTJJjjoTN4hoQ

Для сборки приложения под rockchip px30, необходима среда QT Creator и папка host из результатов работы buildroot (output/host). Если папка host представлена в виде архива, её нужно распаковать

tar -xf host.tar.gz

Настройка QT Creator

- 1. В папке host выполняем скрипт relocate-sdk.sh
- 2. Открываем настройки QT проекта и создаем новую конфигурацию (kit). Для этого нажимаем на «Manage Kits...»

	Manage Kits	X	Build Settings				
lcome Edit	Active Project		Edit build configuration: Debug Add Remove Rename Clone General				
¢ ign	SH_SoundButton -		Shadow build: 🔽				
it Jug	Build & Dup		Build directory: http://www.build-SH_SoundButton_GUI-main/build-SH_SoundButton-Desktop-Debug	Browse			
<u>ن</u>			Build Steps				
)	🖊 Build	í	qmake: qmake SH_SoundButton.pro -spec linux-g++ CONFIG+=debug CONFIG+=qml_del De				
>	Preis at Cathle an		Make: make -j8 in /home/nms/Documents/SH_SoundButton_GUI-main/build-SH_SoundB	Details			
	Editor		Add Build Step -				
	Code Style Dependencies Clang Code Model Clang Tools Testing		Clean Steps Make: make clean -i8 in /home/nms/Documents/SH SoundButton GUI-main/build-SH SoundBu	Details			
			Add Clean Step *	Dettillo			
			Build Environment				
			Use System Environment	Details			

3. Нажимаем на кнопку add и создаем новую конфигурацию. Придумываем ей имя, например РХ30.



		Options — Qt Creator		8			
Filter	Kits						
🖬 Kits	Kits Qt Versions Compiler	s Debuggers Qbs CMake					
Environment	Name		ŀ	٨dd			
Text Editor	Fext Editor Manual						
👫 FakeVim	eVim P px30						
🛿 Help	Make						
{} C++			Setting	js Filter			
🖈 Qt Quick			Default Se	ttings Filter			
➤ Build & Run							
Debugger							
✓ Designer							
Python							
Analyzer							
Version Control							
Devices	Name: File system name: Device type: Device:	PX30 48					
Code Pasting							
🗗 Language Client		Desktop *					
A Testing		Local PC (default for Desktop)	*	Manage			
-	Sysroot:			Browse			
	Compiler:	C: GCC (C, x86 64bit in /usr/bin)	•	Manage			
		C++: GCC (C++, x86 64bit in /usr/bin)	*				
	Environment:	No changes to apply.		Change			
	Debugger:	System GDB at /usr/bin/gdb					
	Qt version:	Qt 5.12.8 in PATH (System)					
	Qt mkspec:						
	Additional Qbs Profile Setting	IS:					
	CMake Tool:	System CMake at /usr/bin/cmake					
	CMake generator:	CodeBlocks - Ninja, Platform: <none>, Toolset: <none></none></none>					
	CMake Configuration: CMAKE_CXX_COMPILER:STRING=%{Compiler:Executable:Cxx}; CMAKE_C_COMPILER:STRING=%{Compiler:Executabl						
		♥Apply	8 <u>C</u> ance	!			

4. Идём во вкладку QT Versions, нажимаем на кнопку Add и указываем следующий путь

<путь до папки host>/bin/qmake

В version name: рекомендуется в скобках указать PX30_QT. Если всё сделано правильно, добавиться новая строчка в графу Manual



its		
its Qt Versio	ns Compilers Debuggers Qbs CMake	
lame	• qmake Location	Ado
Auto-detecte Manual		Rem
A Qt 5.12	.8 in PATH (qt5) /usr/lib/qt5/bin/qmake	
Qt 5.14.2	PX30_QT) /home/nms/Desktop/ORIG/host/bin/qmake	Clear
/ersion name:	Ot %{Ot:Version} (PX30_OT)	

5. Во вкладке Compilers, нужно добавить компиляторы С и С++ из пакета host. Для этого, Нажимаем на Add→GCC→C. Называем, например PX30_GCC. Путь прописываем следующий:

<путь до папки host>/usr/bin/aarch64-linux-gcc

Если строчка светится чёрным, всё сделано правильно.

Manual ▼ C		
C++	GCC	
Name:	PX30_GCC	
<u>C</u> ompiler path:	/home/nms/Desktop/ORIG/host/usr/bin/aarch64-linux-gcc	Browse
Platform codegen flags:		
Platform linker flags:		
	and the second set of the second seco	- 64bit -
ABI:	arm-linux-generic-etr-64bit • arm • - linux • - generic • - etr	
<u>A</u> BI:	arm-unux-generic-etr-64bit • [arm •] - [unux •] - [generic •] - [etr	



Повторяем процедуру для C++. Нажимаем на Add→GCC→C++, название, например PX30_G++, путь:

<путь до папки host>/usr/bin/aarch64-linux-g++

Нажимаем на Apply. Пример правильной настройки:

		Au
• GCC	GCC	
GCC	GCC	
ame:	PX30_G++	
ime: mpiler path:	PX30_G++ /home/nms/Desktop/ORIG/host/usr/bin/aarch64-linux-g++	Browse
ime: impiler path: atform codegen flags	PX30_G++ /home/nms/Desktop/ORIG/host/usr/bin/aarch64-linux-g++ :	Browse
ime: impiler path: atform codegen flags atform linker flags:	PX30_G++ /home/nms/Desktop/ORIG/host/usr/bin/aarch64-linux-g++ :	Browse

6. Во вкладке CMake по аналогии создаем ссылку на нужный CMake. Нажимаем Add, имя PX30 CMAKE, путь:

<путь до папки host>/usr/bin/cmake

сохраняем, нажатием на Apply. Обычный CMake?

7. Возвращаемся во вкладку Kits. Устанавливаем следующие параметры:

Compiler C: PX30_GCC Compiler C++: PX30_G++ QT version: QT ... (PX30_QT) CMake tool: PX30_CMAKE

8. Выбираем созданную конфигурацию по умолчанию (нажимаем make default)



						Options — Qt Creator	8		
Filter	Kits								
🖽 Kits	Kits	Qt Versions	Compilers	Debuggers	Qbs	CMake			
🖵 Environment	ant Name								
Text Editor	Aut • Ma	to-detected nual					Clone		
K. FakeVim	Desktop A PX30 (default)						emove		
🕑 Help			, 			Mak	e Default		
{} C++						Settin	gs Filter		
🖈 Qt Quick						Default S	ettings Filter		
➤ Build & Run									
Debugger									
✓ Designer									
Note: Python									
Analyzer									
Version Control									
Devices	Name	2:		PX30					
Code Pasting	File s	ystem name:							
P Language Client	Devic	e type:		Desktop		¥			
A Testing	Device: Sysrool	Device:		Local PC (default for Desktop)					
ic resering		ot:					Browse		
	Comr	oiler:		C: PX30_GC	с	•	Manage		
				C++: PX30_G+	+	•			
	Envir	onment:		No changes to	apply.		Change		
	Debu	gger:		System GDB a	: /usr/bii	in/gdb •	Manage		
	Qt ve	rsion:		Qt 5.14.2 (PX3	0_QT)	•	Manage		
	Qt m	kspec:							
	Addit	ional Qbs Prof	file Settings:				Change		
	CMak	e Tool:		PX30_CMake		•	Manage		
	CMak	e generator:		CodeBlocks - N	inja, Pla	atform: <none>, Toolset: <none></none></none>	Change		
	CMake Configuration: CMAKE_CXX_COMPILER:STRING=%{Compiler:Executable:Cxx}; CMAKE_C_COMPILER:STRING=%{Compiler:Executabl								
						⊘Apply S⊆an	cel O <u>O</u> K		

Сохраняем и выходим (Apply/OK).