

Embedded Systems Design - Laboratory Exercise No1

1.0

Generated by Doxygen 1.8.6

Thu Oct 15 2020 20:27:34

Contents

1	menuconfig	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	buffer Struct Reference	7
4.1.1	Member Data Documentation	7
4.1.1.1	parent	7
4.1.1.2	state	7
4.2	conf_printer Struct Reference	7
4.2.1	Member Data Documentation	7
4.2.1.1	print_comment	7
4.2.1.2	print_symbol	7
4.3	dialog_color Struct Reference	8
4.3.1	Member Data Documentation	8
4.3.1.1	atr	8
4.3.1.2	bg	8
4.3.1.3	fg	8
4.3.1.4	hl	8
4.4	dialog_info Struct Reference	8
4.4.1	Member Data Documentation	9
4.4.1.1	backtitle	9
4.4.1.2	border	9
4.4.1.3	button_active	9
4.4.1.4	button_inactive	9
4.4.1.5	button_key_active	9
4.4.1.6	button_key_inactive	9
4.4.1.7	button_label_active	9

4.4.1.8	button_label_inactive	9
4.4.1.9	check	9
4.4.1.10	check_selected	9
4.4.1.11	darrow	9
4.4.1.12	dialog	9
4.4.1.13	inputbox	9
4.4.1.14	inputbox_border	9
4.4.1.15	item	9
4.4.1.16	item_selected	9
4.4.1.17	menubox	9
4.4.1.18	menubox_border	9
4.4.1.19	position_indicator	9
4.4.1.20	screen	9
4.4.1.21	searchbox	9
4.4.1.22	searchbox_border	9
4.4.1.23	searchbox_title	10
4.4.1.24	shadow	10
4.4.1.25	tag	10
4.4.1.26	tag_key	10
4.4.1.27	tag_key_selected	10
4.4.1.28	tag_selected	10
4.4.1.29	title	10
4.4.1.30	uarrow	10
4.5	dialog_item Struct Reference	10
4.5.1	Member Data Documentation	10
4.5.1.1	data	10
4.5.1.2	selected	10
4.5.1.3	str	10
4.5.1.4	tag	10
4.6	dialog_list Struct Reference	10
4.6.1	Member Data Documentation	11
4.6.1.1	next	11
4.6.1.2	node	11
4.7	expr Struct Reference	11
4.7.1	Member Data Documentation	11
4.7.1.1	right	11
4.7.1.2	type	11
4.8	expr_data Union Reference	11
4.8.1	Member Data Documentation	11
4.8.1.1	expr	11

4.8.1.2	sym	11
4.9	expr_value Struct Reference	12
4.9.1	Member Data Documentation	12
4.9.1.1	expr	12
4.9.1.2	tri	12
4.10	file Struct Reference	12
4.10.1	Member Data Documentation	12
4.10.1.1	abs_name	12
4.10.1.2	lineno	12
4.10.1.3	name	12
4.10.1.4	next	12
4.10.1.5	parent	12
4.11	gstr Struct Reference	13
4.11.1	Member Data Documentation	13
4.11.1.1	len	13
4.11.1.2	max_width	13
4.11.1.3	s	13
4.12	item Struct Reference	13
4.12.1	Member Data Documentation	13
4.12.1.1	hash	13
4.12.1.2	len	13
4.12.1.3	name	13
4.12.1.4	next	13
4.13	jump_key Struct Reference	13
4.13.1	Member Data Documentation	14
4.13.1.1	entries	14
4.13.1.2	index	14
4.13.1.3	offset	14
4.13.1.4	target	14
4.14	kconf_id Struct Reference	14
4.14.1	Member Data Documentation	14
4.14.1.1	flags	14
4.14.1.2	name	14
4.14.1.3	stype	14
4.14.1.4	token	14
4.15	kconf_id_strings_t Struct Reference	14
4.15.1	Member Data Documentation	15
4.15.1.1	kconf_id_strings_str12	15
4.15.1.2	kconf_id_strings_str13	15
4.15.1.3	kconf_id_strings_str14	15

4.15.1.4	kconf_id_strings_str17	15
4.15.1.5	kconf_id_strings_str18	15
4.15.1.6	kconf_id_strings_str2	15
4.15.1.7	kconf_id_strings_str21	15
4.15.1.8	kconf_id_strings_str22	15
4.15.1.9	kconf_id_strings_str23	15
4.15.1.10	kconf_id_strings_str25	15
4.15.1.11	kconf_id_strings_str27	16
4.15.1.12	kconf_id_strings_str29	16
4.15.1.13	kconf_id_strings_str3	16
4.15.1.14	kconf_id_strings_str31	16
4.15.1.15	kconf_id_strings_str32	16
4.15.1.16	kconf_id_strings_str33	16
4.15.1.17	kconf_id_strings_str35	16
4.15.1.18	kconf_id_strings_str36	16
4.15.1.19	kconf_id_strings_str39	16
4.15.1.20	kconf_id_strings_str41	16
4.15.1.21	kconf_id_strings_str42	16
4.15.1.22	kconf_id_strings_str43	16
4.15.1.23	kconf_id_strings_str46	16
4.15.1.24	kconf_id_strings_str47	16
4.15.1.25	kconf_id_strings_str5	16
4.15.1.26	kconf_id_strings_str51	16
4.15.1.27	kconf_id_strings_str54	16
4.15.1.28	kconf_id_strings_str56	16
4.15.1.29	kconf_id_strings_str7	16
4.15.1.30	kconf_id_strings_str72	16
4.15.1.31	kconf_id_strings_str8	16
4.15.1.32	kconf_id_strings_str9	16
4.16	list_head Struct Reference	16
4.16.1	Member Data Documentation	17
4.16.1.1	next	17
4.16.1.2	prev	17
4.17	menu Struct Reference	17
4.17.1	Member Data Documentation	17
4.17.1.1	data	17
4.17.1.2	dep	17
4.17.1.3	file	17
4.17.1.4	flags	17
4.17.1.5	help	17

4.17.1.6	lineno	17
4.17.1.7	list	17
4.17.1.8	next	17
4.17.1.9	parent	17
4.17.1.10	prompt	17
4.17.1.11	sym	18
4.17.1.12	visibility	18
4.18	out_fct_wrap_type Struct Reference	18
4.18.1	Member Data Documentation	18
4.18.1.1	arg	18
4.18.1.2	fct	18
4.19	property Struct Reference	18
4.19.1	Member Data Documentation	18
4.19.1.1	expr	18
4.19.1.2	file	18
4.19.1.3	lineno	18
4.19.1.4	menu	18
4.19.1.5	next	19
4.19.1.6	sym	19
4.19.1.7	text	19
4.19.1.8	type	19
4.19.1.9	visible	19
4.20	search_data Struct Reference	19
4.20.1	Member Data Documentation	19
4.20.1.1	head	19
4.20.1.2	keys	19
4.20.1.3	targets	19
4.21	symbol Struct Reference	19
4.21.1	Member Data Documentation	20
4.21.1.1	curr	20
4.21.1.2	def	20
4.21.1.3	dir_dep	20
4.21.1.4	flags	20
4.21.1.5	name	20
4.21.1.6	next	20
4.21.1.7	prop	20
4.21.1.8	rev_dep	20
4.21.1.9	type	20
4.21.1.10	visible	20
4.22	symbol_value Struct Reference	20

4.22.1	Member Data Documentation	20
4.22.1.1	tri	20
4.22.1.2	val	20
4.23	yy_buffer_state Struct Reference	20
4.23.1	Member Data Documentation	21
4.23.1.1	yy_at_bol	21
4.23.1.2	yy_bs_column	21
4.23.1.3	yy_bs_lineno	21
4.23.1.4	yy_buf_pos	21
4.23.1.5	yy_buf_size	21
4.23.1.6	yy_buffer_status	21
4.23.1.7	yy_ch_buf	21
4.23.1.8	yy_fill_buffer	21
4.23.1.9	yy_input_file	21
4.23.1.10	yy_is_interactive	21
4.23.1.11	yy_is_our_buffer	21
4.23.1.12	yy_n_chars	21
4.24	yy_trans_info Struct Reference	21
4.24.1	Member Data Documentation	22
4.24.1.1	yy_nxt	22
4.24.1.2	yy_verify	22
4.25	yyalloc Union Reference	22
4.25.1	Member Data Documentation	22
4.25.1.1	yyss_alloc	22
4.25.1.2	yyvs_alloc	22
4.26	YYSTYPE Union Reference	22
4.26.1	Member Data Documentation	22
4.26.1.1	expr	22
4.26.1.2	file	22
4.26.1.3	id	22
4.26.1.4	menu	22
4.26.1.5	string	22
4.26.1.6	symbol	22
5	File Documentation	25
5.1	autoconf.h File Reference	25
5.1.1	Macro Definition Documentation	25
5.1.1.1	CONFIG_LED_BLINK_ITERATOR_50000	25
5.1.1.2	CONFIG_UART_BAUD_RATE	25
5.2	led/led.c File Reference	25

5.2.1	Function Documentation	26
5.2.1.1	led_clear	26
5.2.1.2	led_init	26
5.2.1.3	led_set	26
5.2.2	Variable Documentation	26
5.2.2.1	ahb1enr	26
5.2.2.2	gpio_a_moder	26
5.2.2.3	gpio_a_odr	26
5.2.2.4	gpio_a_otyper	26
5.3	led/led.h File Reference	26
5.3.1	Macro Definition Documentation	27
5.3.1.1	GPIOA_MODER_PA12	27
5.3.1.2	GPIOA_ODR_PA12	27
5.3.1.3	GPIOA_OTYPER_PA12	27
5.3.1.4	RCC_AHB1ENR_GPIOA_EN	27
5.3.2	Function Documentation	27
5.3.2.1	led_clear	27
5.3.2.2	led_init	28
5.3.2.3	led_set	28
5.4	main.c File Reference	28
5.4.1	Macro Definition Documentation	28
5.4.1.1	LED_BLINK	28
5.4.2	Function Documentation	28
5.4.2.1	main	28
5.5	main.h File Reference	29
5.6	menuconfig/basic/fixdep.c File Reference	29
5.6.1	Macro Definition Documentation	29
5.6.1.1	HASHSZ	29
5.6.1.2	INT_CONF	29
5.6.1.3	INT_FIG_	29
5.6.1.4	INT_NFIG	30
5.6.1.5	INT_ONFI	30
5.6.2	Function Documentation	30
5.6.2.1	main	30
5.6.3	Variable Documentation	30
5.6.3.1	cmdline	30
5.6.3.2	depfile	30
5.6.3.3	target	30
5.7	menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdC/CMakeCCompilerId.c File Reference	30
5.7.1	Macro Definition Documentation	30

5.7.1.1	ARCHITECTURE_ID	30
5.7.1.2	COMPILER_ID	30
5.7.1.3	DEC	30
5.7.1.4	HEX	31
5.7.1.5	PLATFORM_ID	31
5.7.2	Function Documentation	31
5.7.2.1	main	31
5.7.3	Variable Documentation	31
5.7.3.1	info_arch	31
5.7.3.2	info_compiler	31
5.7.3.3	info_platform	31
5.8	menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference	31
5.8.1	Macro Definition Documentation	31
5.8.1.1	ARCHITECTURE_ID	31
5.8.1.2	COMPILER_ID	31
5.8.1.3	DEC	32
5.8.1.4	HEX	32
5.8.1.5	PLATFORM_ID	32
5.8.2	Function Documentation	32
5.8.2.1	main	32
5.8.3	Variable Documentation	32
5.8.3.1	info_arch	32
5.8.3.2	info_compiler	32
5.8.3.3	info_platform	32
5.9	menuconfig/build/tmp/gen/choice1.h File Reference	32
5.10	menuconfig/build/tmp/menudependency.h File Reference	32
5.11	tmp/menudependency.h File Reference	32
5.12	menuconfig/build/tmp/num/param.h File Reference	32
5.13	menuconfig/build/tmp/string/param.h File Reference	32
5.14	tmp/num/param.h File Reference	32
5.15	tmp/string/param.h File Reference	32
5.16	menuconfig/build/tmp/num/param2.h File Reference	33
5.17	menuconfig/conf.c File Reference	33
5.17.1	Enumeration Type Documentation	33
5.17.1.1	input_mode	33
5.17.2	Function Documentation	34
5.17.2.1	main	34
5.17.3	Variable Documentation	34
5.17.3.1	input_mode	34
5.18	menuconfig/confdata.c File Reference	34

5.18.1	Macro Definition Documentation	34
5.18.1.1	LINE_GROWTH	34
5.18.2	Function Documentation	35
5.18.2.1	conf_get_autoconfig_name	35
5.18.2.2	conf_get_changed	35
5.18.2.3	conf_get_configname	35
5.18.2.4	conf_get_default_confname	35
5.18.2.5	conf_read	35
5.18.2.6	conf_read_simple	35
5.18.2.7	conf_set_all_new_symbols	35
5.18.2.8	conf_set_changed_callback	35
5.18.2.9	conf_set_message_callback	35
5.18.2.10	conf_write	35
5.18.2.11	conf_write_autoconf	35
5.18.2.12	conf_write_defconfig	35
5.18.2.13	sym_add_change_count	35
5.18.2.14	sym_set_change_count	35
5.18.3	Variable Documentation	35
5.18.3.1	conf_defname	35
5.19	menuconfig/expr.c File Reference	35
5.19.1	Macro Definition Documentation	36
5.19.1.1	DEBUG_EXPR	36
5.19.1.2	e1	36
5.19.1.3	e1	36
5.19.1.4	e1	36
5.19.1.5	e1	36
5.19.1.6	e2	36
5.19.1.7	e2	36
5.19.1.8	e2	36
5.19.1.9	e2	36
5.19.2	Function Documentation	36
5.19.2.1	expr_alloc_and	36
5.19.2.2	expr_alloc_comp	36
5.19.2.3	expr_alloc_one	37
5.19.2.4	expr_alloc_or	37
5.19.2.5	expr_alloc_symbol	37
5.19.2.6	expr_alloc_two	37
5.19.2.7	expr_calc_value	37
5.19.2.8	expr_compare_type	37
5.19.2.9	expr_contains_symbol	37

5.19.2.10	<code>expr_copy</code>	37
5.19.2.11	<code>expr_depends_symbol</code>	37
5.19.2.12	<code>expr_eliminate_dups</code>	37
5.19.2.13	<code>expr_eliminate_eq</code>	37
5.19.2.14	<code>expr_eliminate_yn</code>	37
5.19.2.15	<code>expr_eq</code>	37
5.19.2.16	<code>expr_extract_eq</code>	37
5.19.2.17	<code>expr_extract_eq_and</code>	37
5.19.2.18	<code>expr_extract_eq_or</code>	37
5.19.2.19	<code>expr_fprint</code>	37
5.19.2.20	<code>expr_free</code>	37
5.19.2.21	<code>expr_gstr_print</code>	37
5.19.2.22	<code>expr_print</code>	37
5.19.2.23	<code>expr_simplify_unmet_dep</code>	37
5.19.2.24	<code>expr_trans_bool</code>	37
5.19.2.25	<code>expr_trans_compare</code>	37
5.19.2.26	<code>expr_transform</code>	37
5.20	menuconfig/expr.h File Reference	37
5.20.1	Macro Definition Documentation	40
5.20.1.1	<code>EXPR_AND</code>	40
5.20.1.2	<code>expr_list_for_each_sym</code>	40
5.20.1.3	<code>EXPR_NOT</code>	40
5.20.1.4	<code>EXPR_OR</code>	40
5.20.1.5	<code>for_all_choices</code>	40
5.20.1.6	<code>for_all_defaults</code>	40
5.20.1.7	<code>for_all_prompts</code>	40
5.20.1.8	<code>for_all_properties</code>	40
5.20.1.9	<code>for_all_symbols</code>	40
5.20.1.10	<code>JUMP_NB</code>	40
5.20.1.11	<code>MENU_CHANGED</code>	40
5.20.1.12	<code>MENU_ROOT</code>	40
5.20.1.13	<code>SYMBOL_AUTO</code>	40
5.20.1.14	<code>SYMBOL_CHANGED</code>	40
5.20.1.15	<code>SYMBOL_CHECK</code>	40
5.20.1.16	<code>SYMBOL_CHECKED</code>	40
5.20.1.17	<code>SYMBOL_CHOICE</code>	40
5.20.1.18	<code>SYMBOL_CHOICEVAL</code>	40
5.20.1.19	<code>SYMBOL_CONST</code>	40
5.20.1.20	<code>SYMBOL_DEF</code>	40
5.20.1.21	<code>SYMBOL_DEF3</code>	40

5.20.1.22	SYMBOL_DEF4	41
5.20.1.23	SYMBOL_DEF_AUTO	41
5.20.1.24	SYMBOL_DEF_USER	41
5.20.1.25	SYMBOL_HASHSIZE	41
5.20.1.26	SYMBOL_MAXLENGTH	41
5.20.1.27	SYMBOL_OPTIONAL	41
5.20.1.28	SYMBOL_VALID	41
5.20.1.29	SYMBOL_WARNED	41
5.20.1.30	SYMBOL_WRITE	41
5.20.2	Typedef Documentation	41
5.20.2.1	tristate	41
5.20.3	Enumeration Type Documentation	41
5.20.3.1	anonymous enum	41
5.20.3.2	expr_type	41
5.20.3.3	prop_type	41
5.20.3.4	symbol_type	42
5.20.3.5	tristate	42
5.20.4	Function Documentation	42
5.20.4.1	expr_alloc_and	42
5.20.4.2	expr_alloc_comp	42
5.20.4.3	expr_alloc_one	42
5.20.4.4	expr_alloc_or	42
5.20.4.5	expr_alloc_symbol	42
5.20.4.6	expr_alloc_two	42
5.20.4.7	expr_calc_value	42
5.20.4.8	expr_contains_symbol	42
5.20.4.9	expr_copy	42
5.20.4.10	expr_depends_symbol	43
5.20.4.11	expr_eliminate_dups	43
5.20.4.12	expr_eliminate_eq	43
5.20.4.13	expr_eliminate_yn	43
5.20.4.14	expr_eq	43
5.20.4.15	expr_extract_eq	43
5.20.4.16	expr_extract_eq_and	43
5.20.4.17	expr_extract_eq_or	43
5.20.4.18	expr_fprint	43
5.20.4.19	expr_free	43
5.20.4.20	expr_gstr_print	43
5.20.4.21	expr_simplify_unmet_dep	43
5.20.4.22	expr_trans_bool	43

5.20.4.23	<code>expr_trans_compare</code>	43
5.20.4.24	<code>expr_transform</code>	43
5.20.4.25	<code>lookup_file</code>	43
5.20.5	Variable Documentation	43
5.20.5.1	<code>cdebug</code>	43
5.20.5.2	<code>current_file</code>	43
5.20.5.3	<code>file_list</code>	43
5.20.5.4	<code>modules_sym</code>	43
5.20.5.5	<code>sym_defconfig_list</code>	43
5.20.5.6	<code>symbol_mod</code>	43
5.21	<code>menuconfig/list.h</code> File Reference	43
5.21.1	Macro Definition Documentation	44
5.21.1.1	<code>container_of</code>	44
5.21.1.2	<code>list_entry</code>	44
5.21.1.3	<code>list_for_each_entry</code>	44
5.21.1.4	<code>LIST_HEAD</code>	44
5.21.1.5	<code>LIST_HEAD_INIT</code>	44
5.21.1.6	<code>offsetof</code>	44
5.22	<code>menuconfig/lc.h</code> File Reference	44
5.22.1	Macro Definition Documentation	46
5.22.1.1	<code>_</code>	46
5.22.1.2	<code>CONFIG_</code>	46
5.22.1.3	<code>LOCALEDIR</code>	46
5.22.1.4	<code>N_</code>	46
5.22.1.5	<code>P</code>	46
5.22.1.6	<code>PACKAGE</code>	46
5.22.1.7	<code>SRCTREE</code>	46
5.22.1.8	<code>T_OPT_DEFCONFIG_LIST</code>	46
5.22.1.9	<code>T_OPT_ENV</code>	46
5.22.1.10	<code>T_OPT_MODULES</code>	46
5.22.1.11	<code>TF_COMMAND</code>	46
5.22.1.12	<code>TF_OPTION</code>	46
5.22.1.13	<code>TF_PARAM</code>	46
5.22.2	Enumeration Type Documentation	46
5.22.2.1	<code>conf_def_mode</code>	47
5.22.3	Function Documentation	47
5.22.3.1	<code>_menu_init</code>	47
5.22.3.2	<code>conf_get_autoconfig_name</code>	47
5.22.3.3	<code>conf_get_configname</code>	47
5.22.3.4	<code>conf_get_default_confname</code>	47

5.22.3.5	conf_set_all_new_symbols	47
5.22.3.6	file_lookup	47
5.22.3.7	file_write_dep	47
5.22.3.8	menu_add_dep	47
5.22.3.9	menu_add_entry	47
5.22.3.10	menu_add_expr	47
5.22.3.11	menu_add_menu	47
5.22.3.12	menu_add_option	47
5.22.3.13	menu_add_prompt	47
5.22.3.14	menu_add_prop	47
5.22.3.15	menu_add_symbol	47
5.22.3.16	menu_add_visibility	47
5.22.3.17	menu_end_entry	47
5.22.3.18	menu_end_menu	47
5.22.3.19	menu_finalize	47
5.22.3.20	menu_set_type	47
5.22.3.21	menu_warn	47
5.22.3.22	prop_alloc	47
5.22.3.23	prop_get_symbol	48
5.22.3.24	str_append	48
5.22.3.25	str_assign	48
5.22.3.26	str_free	48
5.22.3.27	str_get	48
5.22.3.28	str_new	48
5.22.3.29	str_printf	48
5.22.3.30	sym_add_change_count	48
5.22.3.31	sym_check_deps	48
5.22.3.32	sym_choice_default	48
5.22.3.33	sym_clear_all_valid	48
5.22.3.34	sym_get_env_prop	48
5.22.3.35	sym_get_string_default	48
5.22.3.36	sym_init	48
5.22.3.37	sym_set_all_changed	48
5.22.3.38	sym_set_change_count	48
5.22.3.39	sym_set_changed	48
5.22.3.40	zconf_curname	48
5.22.3.41	zconf_fopen	48
5.22.3.42	zconf_initscan	48
5.22.3.43	zconf_lineno	48
5.22.3.44	zconf_nextfile	48

5.22.3.45 zconf_starthelp	48
5.22.3.46 zconfdump	48
5.22.3.47 zconfparse	48
5.22.4 Variable Documentation	48
5.22.4.1 sym_env_list	48
5.22.4.2 zconfdebug	48
5.23 menuconfig/lkc_proto.h File Reference	49
5.23.1 Function Documentation	50
5.23.1.1 P	50
5.23.1.2 P	50
5.23.1.3 P	50
5.23.1.4 P	50
5.23.1.5 P	50
5.23.1.6 P	50
5.23.1.7 P	50
5.23.1.8 P	50
5.23.1.9 P	50
5.23.1.10 P	50
5.23.1.11 P	50
5.23.1.12 P	50
5.23.1.13 P	50
5.23.1.14 P	50
5.23.1.15 P	50
5.23.1.16 P	50
5.23.1.17 P	50
5.23.1.18 P	50
5.23.1.19 P	50
5.23.1.20 P	50
5.23.1.21 P	50
5.23.1.22 P	50
5.23.1.23 P	50
5.23.1.24 P	50
5.23.1.25 P	50
5.23.1.26 P	50
5.23.1.27 P	50
5.23.1.28 P	51
5.23.1.29 P	51
5.23.1.30 P	51
5.23.1.31 P	51
5.23.1.32 P	51

5.23.1.33 P	51
5.23.1.34 P	51
5.23.1.35 P	51
5.23.1.36 P	51
5.23.1.37 P	51
5.23.1.38 P	51
5.23.1.39 P	51
5.23.1.40 P	51
5.23.1.41 P	51
5.23.1.42 P	51
5.24 menuconfig/lxdialog/checklist.c File Reference	51
5.24.1 Function Documentation	51
5.24.1.1 dialog_checklist	51
5.25 menuconfig/lxdialog/dialog.h File Reference	52
5.25.1 Macro Definition Documentation	54
5.25.1.1 ACS_DARROW	54
5.25.1.2 ACS_HLINE	54
5.25.1.3 ACS_LLCORNER	54
5.25.1.4 ACS_LRCORNER	54
5.25.1.5 ACS_LTEE	54
5.25.1.6 ACS_RTEE	54
5.25.1.7 ACS_UARROW	54
5.25.1.8 ACS_ULCORNER	54
5.25.1.9 ACS_URCORNER	54
5.25.1.10 ACS_VLINE	54
5.25.1.11 BUF_SIZE	54
5.25.1.12 ERRDISPLAYTOOSMALL	54
5.25.1.13 item_foreach	54
5.25.1.14 KEY_ESC	54
5.25.1.15 M_EVENT	54
5.25.1.16 MAX	54
5.25.1.17 MAX_LEN	54
5.25.1.18 MAXITEMSTR	54
5.25.1.19 MIN	54
5.25.1.20 TAB	54
5.25.1.21 TR	54
5.25.2 Typedef Documentation	54
5.25.2.1 update_text_fn	54
5.25.3 Function Documentation	54
5.25.3.1 attr_clear	55

5.25.3.2	dialog_checklist	55
5.25.3.3	dialog_clear	55
5.25.3.4	dialog_inputbox	55
5.25.3.5	dialog_menu	55
5.25.3.6	dialog_msgbox	55
5.25.3.7	dialog_textbox	55
5.25.3.8	dialog_yesno	55
5.25.3.9	draw_box	55
5.25.3.10	draw_shadow	55
5.25.3.11	end_dialog	55
5.25.3.12	first_alpha	55
5.25.3.13	init_dialog	55
5.25.3.14	item_activate_selected	55
5.25.3.15	item_add_str	55
5.25.3.16	item_count	55
5.25.3.17	item_data	55
5.25.3.18	item_is_selected	55
5.25.3.19	item_is_tag	55
5.25.3.20	item_make	55
5.25.3.21	item_n	55
5.25.3.22	item_reset	55
5.25.3.23	item_set	55
5.25.3.24	item_set_data	55
5.25.3.25	item_set_selected	55
5.25.3.26	item_set_tag	55
5.25.3.27	item_str	55
5.25.3.28	item_tag	55
5.25.3.29	on_key_esc	56
5.25.3.30	on_key_resize	56
5.25.3.31	print_utowrap	56
5.25.3.32	print_button	56
5.25.3.33	print_title	56
5.25.3.34	set_dialog_backtitle	56
5.25.4	Variable Documentation	56
5.25.4.1	dialog_input_result	56
5.25.4.2	dlg	56
5.25.4.3	item_cur	56
5.25.4.4	item_head	56
5.25.4.5	item_nil	56
5.25.4.6	saved_x	56

5.25.4.7	saved_y	56
5.26	menuconfig/lxdialog/inputbox.c File Reference	56
5.26.1	Function Documentation	56
5.26.1.1	dialog_inputbox	56
5.26.2	Variable Documentation	56
5.26.2.1	dialog_input_result	56
5.27	menuconfig/lxdialog/menubox.c File Reference	56
5.27.1	Macro Definition Documentation	57
5.27.1.1	print_item	57
5.27.2	Function Documentation	57
5.27.2.1	dialog_menu	57
5.28	menuconfig/lxdialog/textbox.c File Reference	57
5.28.1	Function Documentation	57
5.28.1.1	dialog_textbox	57
5.29	menuconfig/lxdialog/util.c File Reference	57
5.29.1	Macro Definition Documentation	58
5.29.1.1	DLG_COLOR	58
5.29.2	Function Documentation	59
5.29.2.1	attr_clear	59
5.29.2.2	dialog_clear	59
5.29.2.3	draw_box	59
5.29.2.4	draw_shadow	59
5.29.2.5	end_dialog	59
5.29.2.6	first_alpha	59
5.29.2.7	init_dialog	59
5.29.2.8	item_activate_selected	59
5.29.2.9	item_add_str	59
5.29.2.10	item_count	59
5.29.2.11	item_data	59
5.29.2.12	item_is_selected	59
5.29.2.13	item_is_tag	59
5.29.2.14	item_make	59
5.29.2.15	item_n	59
5.29.2.16	item_reset	59
5.29.2.17	item_set	59
5.29.2.18	item_set_data	59
5.29.2.19	item_set_selected	59
5.29.2.20	item_set_tag	59
5.29.2.21	item_str	59
5.29.2.22	item_tag	59

5.29.2.23	on_key_esc	59
5.29.2.24	on_key_resize	59
5.29.2.25	print_utowrap	59
5.29.2.26	print_button	59
5.29.2.27	print_title	59
5.29.2.28	set_dialog_backtitle	60
5.29.3	Variable Documentation	60
5.29.3.1	dlg	60
5.29.3.2	item_cur	60
5.29.3.3	item_head	60
5.29.3.4	item_nil	60
5.29.3.5	saved_x	60
5.29.3.6	saved_y	60
5.30	menuconfig/util.c File Reference	60
5.30.1	Function Documentation	60
5.30.1.1	file_lookup	60
5.30.1.2	file_write_dep	60
5.30.1.3	str_append	60
5.30.1.4	str_assign	60
5.30.1.5	str_free	60
5.30.1.6	str_get	60
5.30.1.7	str_new	60
5.30.1.8	str_printf	60
5.31	menuconfig/lxdialog/yesno.c File Reference	61
5.31.1	Function Documentation	61
5.31.1.1	dialog_yesno	61
5.32	menuconfig/mconf.c File Reference	61
5.32.1	Function Documentation	61
5.32.1.1	main	61
5.33	menuconfig/menu.c File Reference	61
5.33.1	Function Documentation	62
5.33.1.1	_menu_init	62
5.33.1.2	get_relations_str	62
5.33.1.3	get_symbol_str	62
5.33.1.4	menu_add_dep	62
5.33.1.5	menu_add_entry	62
5.33.1.6	menu_add_expr	62
5.33.1.7	menu_add_menu	62
5.33.1.8	menu_add_option	62
5.33.1.9	menu_add_prompt	62

5.33.1.10 menu_add_prop	63
5.33.1.11 menu_add_symbol	63
5.33.1.12 menu_add_visibility	63
5.33.1.13 menu_end_entry	63
5.33.1.14 menu_end_menu	63
5.33.1.15 menu_finalize	63
5.33.1.16 menu_get_ext_help	63
5.33.1.17 menu_get_help	63
5.33.1.18 menu_get_parent_menu	63
5.33.1.19 menu_get_prompt	63
5.33.1.20 menu_get_root_menu	63
5.33.1.21 menu_has_help	63
5.33.1.22 menu_has_prompt	63
5.33.1.23 menu_is_visible	63
5.33.1.24 menu_set_type	63
5.33.1.25 menu_warn	63
5.33.2 Variable Documentation	63
5.33.2.1 current_file	63
5.33.2.2 file_list	63
5.33.2.3 rootmenu	63
5.34 menuconfig/README.md File Reference	63
5.35 menuconfig/symbol.c File Reference	63
5.35.1 Function Documentation	65
5.35.1.1 prop_alloc	65
5.35.1.2 prop_get_symbol	65
5.35.1.3 prop_get_type_name	65
5.35.1.4 sym_calc_value	65
5.35.1.5 sym_check_deps	65
5.35.1.6 sym_choice_default	65
5.35.1.7 sym_clear_all_valid	65
5.35.1.8 sym_escape_string_value	65
5.35.1.9 sym_expand_string_value	65
5.35.1.10 sym_find	65
5.35.1.11 sym_get_choice_prop	65
5.35.1.12 sym_get_default_prop	65
5.35.1.13 sym_get_env_prop	65
5.35.1.14 sym_get_string_default	65
5.35.1.15 sym_get_string_value	65
5.35.1.16 sym_get_type	65
5.35.1.17 sym_init	65

5.35.1.18	sym_is_changable	65
5.35.1.19	sym_lookup	65
5.35.1.20	sym_re_search	65
5.35.1.21	sym_set_all_changed	65
5.35.1.22	sym_set_changed	65
5.35.1.23	sym_set_string_value	65
5.35.1.24	sym_set_tristate_value	65
5.35.1.25	sym_string_valid	65
5.35.1.26	sym_string_within_range	65
5.35.1.27	sym_toggle_tristate_value	65
5.35.1.28	sym_tristate_within_range	66
5.35.1.29	sym_type_name	66
5.35.2	Variable Documentation	66
5.35.2.1	modules_sym	66
5.35.2.2	modules_val	66
5.35.2.3	sym_defconfig_list	66
5.35.2.4	sym_env_list	66
5.35.2.5	symbol_empty	66
5.35.2.6	symbol_mod	66
5.35.2.7	symbol_no	66
5.35.2.8	symbol_yes	66
5.36	menuconfig/zconf.hash.c File Reference	66
5.36.1	Macro Definition Documentation	67
5.36.1.1	kconf_id_strings	67
5.37	menuconfig/zconf.lex.c File Reference	67
5.37.1	Macro Definition Documentation	70
5.37.1.1	BEGIN	70
5.37.1.2	COMMAND	70
5.37.1.3	ECHO	70
5.37.1.4	EOB_ACT_CONTINUE_SCAN	70
5.37.1.5	EOB_ACT_END_OF_FILE	70
5.37.1.6	EOB_ACT_LAST_MATCH	70
5.37.1.7	FLEX_BETA	70
5.37.1.8	FLEX_SCANNER	70
5.37.1.9	FLEXINT_H	70
5.37.1.10	HELP	70
5.37.1.11	INITIAL	70
5.37.1.12	INT16_MAX	71
5.37.1.13	INT16_MIN	71
5.37.1.14	INT32_MAX	71

5.37.1.15 INT32_MIN	71
5.37.1.16 INT8_MAX	71
5.37.1.17 INT8_MIN	71
5.37.1.18 PARAM	71
5.37.1.19 REJECT	71
5.37.1.20 START_STRSIZE	71
5.37.1.21 STRING	71
5.37.1.22 UINT16_MAX	71
5.37.1.23 UINT32_MAX	71
5.37.1.24 UINT8_MAX	71
5.37.1.25 unput	71
5.37.1.26 YY_AT_BOL	71
5.37.1.27 YY_BREAK	71
5.37.1.28 YY_BUF_SIZE	71
5.37.1.29 YY_BUFFER_EOF_PENDING	71
5.37.1.30 YY_BUFFER_NEW	71
5.37.1.31 YY_BUFFER_NORMAL	71
5.37.1.32 yy_create_buffer	71
5.37.1.33 YY_CURRENT_BUFFER	71
5.37.1.34 YY_CURRENT_BUFFER_LVALUE	71
5.37.1.35 YY_DECL	71
5.37.1.36 YY_DECL_IS_OURS	71
5.37.1.37 yy_delete_buffer	72
5.37.1.38 YY_DO_BEFORE_ACTION	72
5.37.1.39 YY_END_OF_BUFFER	72
5.37.1.40 YY_END_OF_BUFFER_CHAR	72
5.37.1.41 YY_EXIT_FAILURE	72
5.37.1.42 YY_EXTRA_TYPE	72
5.37.1.43 YY_FATAL_ERROR	72
5.37.1.44 yy_flex_debug	72
5.37.1.45 YY_FLEX_MAJOR_VERSION	72
5.37.1.46 YY_FLEX_MINOR_VERSION	72
5.37.1.47 YY_FLEX_SUBMINOR_VERSION	72
5.37.1.48 yy_flush_buffer	72
5.37.1.49 YY_FLUSH_BUFFER	72
5.37.1.50 yy_init_buffer	72
5.37.1.51 YY_INPUT	72
5.37.1.52 YY_INT_ALIGNED	72
5.37.1.53 YY_LESS_LINENO	72
5.37.1.54 yy_load_buffer_state	72

5.37.1.55 YY_MORE_ADJ	73
5.37.1.56 yy_new_buffer	73
5.37.1.57 YY_NEW_FILE	73
5.37.1.58 YY_NO_INPUT	73
5.37.1.59 YY_NULL	73
5.37.1.60 YY_NUM_RULES	73
5.37.1.61 YY_READ_BUF_SIZE	73
5.37.1.62 YY_RESTORE_YY_MORE_OFFSET	73
5.37.1.63 YY_RULE_SETUP	73
5.37.1.64 YY_SC_TO_UI	73
5.37.1.65 yy_set_bol	73
5.37.1.66 yy_set_interactive	73
5.37.1.67 YY_SKIP_YYWRAP	73
5.37.1.68 YY_START	73
5.37.1.69 YY_START_STACK_INCR	73
5.37.1.70 YY_STATE_BUF_SIZE	73
5.37.1.71 YY_STATE_EOF	73
5.37.1.72 YY_STRUCT_YY_BUFFER_STATE	73
5.37.1.73 yy_switch_to_buffer	74
5.37.1.74 YY_TYPEDEF_YY_BUFFER_STATE	74
5.37.1.75 YY_TYPEDEF_YY_SIZE_T	74
5.37.1.76 YY_USER_ACTION	74
5.37.1.77 yyallocc	74
5.37.1.78 yyconst	74
5.37.1.79 yyfree	74
5.37.1.80 yyin	74
5.37.1.81 yyleng	74
5.37.1.82 yyles	74
5.37.1.83 yless	74
5.37.1.84 yylex	74
5.37.1.85 yylineno	74
5.37.1.86 yymore	74
5.37.1.87 yyout	74
5.37.1.88 yyrealloc	74
5.37.1.89 yyrestart	75
5.37.1.90 YYSTATE	75
5.37.1.91 YYTABLES_NAME	75
5.37.1.92 yyterminate	75
5.37.1.93 yytext	75
5.37.1.94 yytext_ptr	75

5.37.1.95	yywrap	75
5.37.1.96	zconfwrap	75
5.37.2	Typedef Documentation	75
5.37.2.1	flex_int16_t	75
5.37.2.2	flex_int32_t	75
5.37.2.3	flex_int8_t	75
5.37.2.4	flex_uint16_t	75
5.37.2.5	flex_uint32_t	75
5.37.2.6	flex_uint8_t	75
5.37.2.7	YY_BUFFER_STATE	75
5.37.2.8	YY_CHAR	75
5.37.2.9	yy_size_t	75
5.37.2.10	yy_state_type	75
5.37.3	Function Documentation	75
5.37.3.1	if	75
5.37.3.2	while	75
5.37.3.3	zconf_create_buffer	75
5.37.3.4	zconf_curname	76
5.37.3.5	zconf_delete_buffer	76
5.37.3.6	zconf_flush_buffer	76
5.37.3.7	zconf_fopen	76
5.37.3.8	zconf_initscan	76
5.37.3.9	zconf_lineno	76
5.37.3.10	zconf_nextfile	76
5.37.3.11	zconf_scan_buffer	76
5.37.3.12	zconf_scan_bytes	76
5.37.3.13	zconf_scan_string	76
5.37.3.14	zconf_starthelp	77
5.37.3.15	zconf_switch_to_buffer	77
5.37.3.16	zconfalloc	77
5.37.3.17	zconfree	77
5.37.3.18	zconfget_debug	77
5.37.3.19	zconfget_extra	77
5.37.3.20	zconfget_in	77
5.37.3.21	zconfget_leng	77
5.37.3.22	zconfget_lineno	77
5.37.3.23	zconfget_out	77
5.37.3.24	zconfget_text	77
5.37.3.25	zconflex	78
5.37.3.26	zconflex_destroy	78

5.37.3.27	zconfpop_buffer_state	78
5.37.3.28	zconfpush_buffer_state	78
5.37.3.29	zconfrealloc	78
5.37.3.30	zconfrestart	78
5.37.3.31	zconfset_debug	78
5.37.3.32	zconfset_extra	78
5.37.3.33	zconfset_in	78
5.37.3.34	zconfset_lineno	78
5.37.3.35	zconfset_out	79
5.37.4	Variable Documentation	79
5.37.4.1	current_buf	79
5.37.4.2	file	79
5.37.4.3	i	79
5.37.4.4	lineno	79
5.37.4.5	str	79
5.37.4.6	ts	79
5.37.4.7	yy_act	79
5.37.4.8	yy_bp	79
5.37.4.9	yy_cp	79
5.37.4.10	YY_DECL	79
5.37.4.11	zconf_flex_debug	79
5.37.4.12	zconfin	79
5.37.4.13	zconfleng	79
5.37.4.14	zconflineno	79
5.37.4.15	zconfout	79
5.37.4.16	zconfcontext	79
5.38	menuconfig/zconf.tab.c File Reference	80
5.38.1	Macro Definition Documentation	82
5.38.1.1	DEBUG_PARSE	82
5.38.1.2	PATH_MAX	82
5.38.1.3	printd	82
5.38.1.4	PRINTD	82
5.38.1.5	YY_	82
5.38.1.6	YY_LOCATION_PRINT	82
5.38.1.7	YY_REDUCE_PRINT	82
5.38.1.8	YY_STACK_PRINT	83
5.38.1.9	YY_SYMBOL_PRINT	83
5.38.1.10	YYABORT	83
5.38.1.11	YYACCEPT	83
5.38.1.12	YYBACKUP	83

5.38.1.13 YYBISON	83
5.38.1.14 YYBISON_VERSION	83
5.38.1.15 yychar	83
5.38.1.16 yyclearin	83
5.38.1.17 YYCOPY	83
5.38.1.18 yydebug	84
5.38.1.19 YYDEBUG	84
5.38.1.20 YYDPRINTF	84
5.38.1.21 YYEMPTY	84
5.38.1.22 YYEOF	84
5.38.1.23 YYERRCODE	84
5.38.1.24 yyerrok	84
5.38.1.25 yyerror	84
5.38.1.26 YYERROR	84
5.38.1.27 YYERROR_VERBOSE	84
5.38.1.28 YYFAIL	84
5.38.1.29 YYFINAL	84
5.38.1.30 YYFPRINTF	84
5.38.1.31 YYFREE	84
5.38.1.32 YYID	84
5.38.1.33 YYINITDEPTH	84
5.38.1.34 YYLAST	84
5.38.1.35 yylex	84
5.38.1.36 YYLEX	84
5.38.1.37 YYLLOC_DEFAULT	84
5.38.1.38 YYLSP_NEEDED	85
5.38.1.39 yylval	85
5.38.1.40 YYMALLOC	85
5.38.1.41 YYMAXDEPTH	85
5.38.1.42 YYMAXUTOK	85
5.38.1.43 yynerrs	85
5.38.1.44 YYNNTS	85
5.38.1.45 YYNRULES	85
5.38.1.46 YYNSTATES	85
5.38.1.47 YYNTOKENS	85
5.38.1.48 YYPACT_NINF	85
5.38.1.49 yyparse	85
5.38.1.50 YYPOPSTACK	85
5.38.1.51 YYPULL	85
5.38.1.52 YYPURE	85

5.38.1.53	YYPUSH	85
5.38.1.54	YYRECOVERING	85
5.38.1.55	YYRHSLOC	85
5.38.1.56	YYSIZE_MAXIMUM	85
5.38.1.57	YYSIZE_T	85
5.38.1.58	YYSKELETON_NAME	85
5.38.1.59	YYSTACK_ALLOC	85
5.38.1.60	YYSTACK_ALLOC_MAXIMUM	85
5.38.1.61	YYSTACK_BYTES	85
5.38.1.62	YYSTACK_FREE	86
5.38.1.63	YYSTACK_GAP_MAXIMUM	86
5.38.1.64	YYSTACK_RELOCATE	86
5.38.1.65	yystype	86
5.38.1.66	YYSTYPE_IS_DECLARED	86
5.38.1.67	YYSTYPE_IS_TRIVIAL	86
5.38.1.68	YYTABLE_NINF	86
5.38.1.69	YYTERROR	86
5.38.1.70	YYTOKEN_TABLE	86
5.38.1.71	YYTOKENTYPE	86
5.38.1.72	YYTRANSLATE	86
5.38.1.73	YYUNDEFTOK	86
5.38.1.74	YYUSE	86
5.38.2	Typedef Documentation	86
5.38.2.1	YYSTYPE	86
5.38.2.2	yytype_int16	86
5.38.2.3	yytype_int8	86
5.38.2.4	yytype_uint16	86
5.38.2.5	yytype_uint8	86
5.38.3	Enumeration Type Documentation	86
5.38.3.1	yytokentype	86
5.38.4	Function Documentation	87
5.38.4.1	conf_parse	87
5.38.4.2	yyparse	87
5.38.4.3	zconfdump	87
5.38.4.4	zconflex	87
5.38.5	Variable Documentation	87
5.38.5.1	cdebug	87
5.38.5.2	symbol_hash	87
5.38.5.3	yychar	88
5.38.5.4	yydebug	88

5.38.5.5	yyval	88
5.38.5.6	yynerrs	88
5.39	printf/printf.c File Reference	88
5.39.1	Macro Definition Documentation	89
5.39.1.1	FLAGS_ADAPT_EXP	89
5.39.1.2	FLAGS_CHAR	89
5.39.1.3	FLAGS_HASH	89
5.39.1.4	FLAGS_LEFT	89
5.39.1.5	FLAGS_LONG	89
5.39.1.6	FLAGS_LONG_LONG	89
5.39.1.7	FLAGS_PLUS	89
5.39.1.8	FLAGS_PRECISION	89
5.39.1.9	FLAGS_SHORT	89
5.39.1.10	FLAGS_SPACE	89
5.39.1.11	FLAGS_UPPERCASE	89
5.39.1.12	FLAGS_ZEROPAD	89
5.39.1.13	PRINTF_DEFAULT_FLOAT_PRECISION	89
5.39.1.14	PRINTF_FTOA_BUFFER_SIZE	89
5.39.1.15	PRINTF_MAX_FLOAT	89
5.39.1.16	PRINTF_NTOA_BUFFER_SIZE	89
5.39.1.17	PRINTF_SUPPORT_EXPONENTIAL	89
5.39.1.18	PRINTF_SUPPORT_FLOAT	89
5.39.1.19	PRINTF_SUPPORT_LONG_LONG	89
5.39.1.20	PRINTF_SUPPORT_PTRDIFF_T	89
5.39.2	Typedef Documentation	89
5.39.2.1	out_fct_type	89
5.39.3	Function Documentation	90
5.39.3.1	fctprintf	90
5.39.3.2	printf_	90
5.39.3.3	snprintf_	90
5.39.3.4	sprintf_	90
5.39.3.5	vprintf_	90
5.39.3.6	vsprintf_	90
5.40	printf/printf.h File Reference	90
5.40.1	Macro Definition Documentation	91
5.40.1.1	printf	91
5.40.1.2	snprintf	92
5.40.1.3	sprintf	92
5.40.1.4	vprintf	92
5.40.1.5	vsprintf	93

5.40.2	Function Documentation	93
5.40.2.1	_putchar	93
5.40.2.2	fctprintf	93
5.40.2.3	printf_	93
5.40.2.4	snprintf_	93
5.40.2.5	sprintf_	93
5.40.2.6	vprintf_	93
5.40.2.7	vsnprintf_	93
5.41	startup.c File Reference	93
5.41.1	Function Documentation	94
5.41.1.1	__attribute__	94
5.41.1.2	bus_fault_handler	94
5.41.1.3	debug_mon_handler	94
5.41.1.4	default_handler	94
5.41.1.5	hard_fault_handler	94
5.41.1.6	main	94
5.41.1.7	mem_manage_handler	94
5.41.1.8	nmi_handler	94
5.41.1.9	pend_sv_handler	94
5.41.1.10	reset_handler	94
5.41.1.11	svc_handler	94
5.41.1.12	systick_handler	94
5.41.1.13	usage_fault_handler	95
5.41.2	Variable Documentation	95
5.41.2.1	_ebss	95
5.41.2.2	_edata	95
5.41.2.3	_estack	95
5.41.2.4	_sbss	95
5.41.2.5	_sdata	95
5.41.2.6	_sdata_lma	95
5.42	startup.h File Reference	95
5.42.1	Function Documentation	95
5.42.1.1	bus_fault_handler	95
5.42.1.2	debug_mon_handler	95
5.42.1.3	default_handler	95
5.42.1.4	hard_fault_handler	95
5.42.1.5	mem_manage_handler	95
5.42.1.6	nmi_handler	95
5.42.1.7	pend_sv_handler	95
5.42.1.8	reset_handler	95

5.42.1.9	svc_handler	96
5.42.1.10	systick_handler	96
5.42.1.11	usage_fault_handler	96
5.43	tmp/led/blink/iterator/200000.h File Reference	96
5.44	tmp/led/blink/iterator/400000.h File Reference	96
5.45	tmp/led/blink/iterator/50000.h File Reference	96
5.46	tmp/uart/baud/rate.h File Reference	96
5.47	uart/uart.c File Reference	96
5.47.1	Function Documentation	96
5.47.1.1	_putchar	96
5.47.1.2	uart_init	96
5.47.1.3	uart_send_buff	97
5.48	uart/uart.h File Reference	97
5.48.1	Macro Definition Documentation	98
5.48.1.1	GPIOA_AF7_PA10	98
5.48.1.2	GPIOA_AF7_PA9	98
5.48.1.3	GPIOA_MODER_AF_PA10	98
5.48.1.4	GPIOA_MODER_AF_PA9	98
5.48.1.5	GPIOA_OTYPER_PA9	98
5.48.1.6	RCC_AHB1ENR_GPIOA_EN	98
5.48.1.7	RCC_APB2_ENR_USART1_EN	98
5.48.1.8	USART1_RE	98
5.48.1.9	USART1_REACK	98
5.48.1.10	USART1_TE	98
5.48.1.11	USART1_TEACK	98
5.48.1.12	USART1_TXE	98
5.48.1.13	USART1_UE	98
5.48.2	Function Documentation	98
5.48.2.1	_putchar	98
5.48.2.2	uart_init	98
5.48.2.3	uart_send_buff	99
Index		100

Chapter 1

menuconfig

This is a standalone version of the mconf tool from the linux kernel.

Licence

The code is distributed under the GPLv2.0.

building

```
mkdir build
cd build
cmake ..
make
```

testing

```
cd build
./mconf ../test/rootconf
```


Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

buffer	7
conf_printer	7
dialog_color	8
dialog_info	8
dialog_item	10
dialog_list	10
expr	11
expr_data	11
expr_value	12
file	12
gstr	13
item	13
jump_key	13
kconf_id	14
kconf_id_strings_t	14
list_head	16
menu	17
out_fct_wrap_type	18
property	18
search_data	19
symbol	19
symbol_value	20
yy_buffer_state	20
yy_trans_info	21
yyalloc	22
YYSTYPE	22

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

autoconf.h	25
main.c	28
main.h	29
startup.c	93
startup.h	95
led/led.c	25
led/led.h	26
menuconfig/conf.c	33
menuconfig/confdata.c	34
menuconfig/expr.c	35
menuconfig/expr.h	37
menuconfig/list.h	43
menuconfig/lkc.h	44
menuconfig/lkc_proto.h	49
menuconfig/mconf.c	61
menuconfig/menu.c	61
menuconfig/symbol.c	63
menuconfig/util.c	60
menuconfig/zconf.hash.c	66
menuconfig/zconf.lex.c	67
menuconfig/zconf.tab.c	80
menuconfig/basic/fixdep.c	29
menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdC/CMakeCCompilerId.c	30
menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdCXX/CMakeCXXCompilerId.cpp	31
menuconfig/build/tmp/menudependency.h	32
menuconfig/build/tmp/gen/choice1.h	32
menuconfig/build/tmp/num/param.h	32
menuconfig/build/tmp/num/param2.h	33
menuconfig/build/tmp/string/param.h	32
menuconfig/lxdialog/checklist.c	51
menuconfig/lxdialog/dialog.h	52
menuconfig/lxdialog/inputbox.c	56
menuconfig/lxdialog/menubox.c	56
menuconfig/lxdialog/textbox.c	57
menuconfig/lxdialog/util.c	57
menuconfig/lxdialog/yesno.c	61
printf/printf.c	88
printf/printf.h	90

tmp/menudependency.h	32
tmp/led/blink/iterator/200000.h	96
tmp/led/blink/iterator/400000.h	96
tmp/led/blink/iterator/50000.h	96
tmp/num/param.h	32
tmp/string/param.h	32
tmp/uart/ baud/rate.h	96
uart/uart.c	96
uart/uart.h	97

Chapter 4

Class Documentation

4.1 buffer Struct Reference

Public Attributes

- struct [buffer](#) * [parent](#)
- [YY_BUFFER_STATE](#) [state](#)

4.1.1 Member Data Documentation

4.1.1.1 struct [buffer](#)* [buffer::parent](#)

4.1.1.2 [YY_BUFFER_STATE](#) [buffer::state](#)

The documentation for this struct was generated from the following file:

- [menuconfig/zconf.lex.c](#)

4.2 conf_printer Struct Reference

```
#include <lkc.h>
```

Public Attributes

- void(* [print_symbol](#))(FILE *, struct [symbol](#) *, const char *, void *)
- void(* [print_comment](#))(FILE *, const char *, void *)

4.2.1 Member Data Documentation

4.2.1.1 void(* [conf_printer::print_comment](#))(FILE *, const char *, void *)

4.2.1.2 void(* [conf_printer::print_symbol](#))(FILE *, struct [symbol](#) *, const char *, void *)

The documentation for this struct was generated from the following file:

- [menuconfig/lkc.h](#)

4.3 dialog_color Struct Reference

```
#include <dialog.h>
```

Public Attributes

- ctype [atr](#)
- int [fg](#)
- int [bg](#)
- int [hl](#)

4.3.1 Member Data Documentation

4.3.1.1 ctype [dialog_color::atr](#)

4.3.1.2 int [dialog_color::bg](#)

4.3.1.3 int [dialog_color::fg](#)

4.3.1.4 int [dialog_color::hl](#)

The documentation for this struct was generated from the following file:

- menuconfig/lxdialog/[dialog.h](#)

4.4 dialog_info Struct Reference

```
#include <dialog.h>
```

Public Attributes

- const char * [backtitle](#)
- struct [dialog_color](#) [screen](#)
- struct [dialog_color](#) [shadow](#)
- struct [dialog_color](#) [dialog](#)
- struct [dialog_color](#) [title](#)
- struct [dialog_color](#) [border](#)
- struct [dialog_color](#) [button_active](#)
- struct [dialog_color](#) [button_inactive](#)
- struct [dialog_color](#) [button_key_active](#)
- struct [dialog_color](#) [button_key_inactive](#)
- struct [dialog_color](#) [button_label_active](#)
- struct [dialog_color](#) [button_label_inactive](#)
- struct [dialog_color](#) [inputbox](#)
- struct [dialog_color](#) [inputbox_border](#)
- struct [dialog_color](#) [searchbox](#)
- struct [dialog_color](#) [searchbox_title](#)
- struct [dialog_color](#) [searchbox_border](#)
- struct [dialog_color](#) [position_indicator](#)
- struct [dialog_color](#) [menubox](#)
- struct [dialog_color](#) [menubox_border](#)
- struct [dialog_color](#) [item](#)

- struct [dialog_color item_selected](#)
- struct [dialog_color tag](#)
- struct [dialog_color tag_selected](#)
- struct [dialog_color tag_key](#)
- struct [dialog_color tag_key_selected](#)
- struct [dialog_color check](#)
- struct [dialog_color check_selected](#)
- struct [dialog_color uarrow](#)
- struct [dialog_color darrow](#)

4.4.1 Member Data Documentation

4.4.1.1 `const char* dialog_info::backtitle`

4.4.1.2 `struct dialog_color dialog_info::border`

4.4.1.3 `struct dialog_color dialog_info::button_active`

4.4.1.4 `struct dialog_color dialog_info::button_inactive`

4.4.1.5 `struct dialog_color dialog_info::button_key_active`

4.4.1.6 `struct dialog_color dialog_info::button_key_inactive`

4.4.1.7 `struct dialog_color dialog_info::button_label_active`

4.4.1.8 `struct dialog_color dialog_info::button_label_inactive`

4.4.1.9 `struct dialog_color dialog_info::check`

4.4.1.10 `struct dialog_color dialog_info::check_selected`

4.4.1.11 `struct dialog_color dialog_info::darrow`

4.4.1.12 `struct dialog_color dialog_info::dialog`

4.4.1.13 `struct dialog_color dialog_info::inputbox`

4.4.1.14 `struct dialog_color dialog_info::inputbox_border`

4.4.1.15 `struct dialog_color dialog_info::item`

4.4.1.16 `struct dialog_color dialog_info::item_selected`

4.4.1.17 `struct dialog_color dialog_info::menubox`

4.4.1.18 `struct dialog_color dialog_info::menubox_border`

4.4.1.19 `struct dialog_color dialog_info::position_indicator`

4.4.1.20 `struct dialog_color dialog_info::screen`

4.4.1.21 `struct dialog_color dialog_info::searchbox`

4.4.1.22 `struct dialog_color dialog_info::searchbox_border`

4.4.1.23 struct dialog_color dialog_info::searchbox_title

4.4.1.24 struct dialog_color dialog_info::shadow

4.4.1.25 struct dialog_color dialog_info::tag

4.4.1.26 struct dialog_color dialog_info::tag_key

4.4.1.27 struct dialog_color dialog_info::tag_key_selected

4.4.1.28 struct dialog_color dialog_info::tag_selected

4.4.1.29 struct dialog_color dialog_info::title

4.4.1.30 struct dialog_color dialog_info::uarrow

The documentation for this struct was generated from the following file:

- menuconfig/lxdialog/[dialog.h](#)

4.5 dialog_item Struct Reference

```
#include <dialog.h>
```

Public Attributes

- char [str](#) [[MAXITEMSTR](#)]
- char [tag](#)
- void * [data](#)
- int [selected](#)

4.5.1 Member Data Documentation

4.5.1.1 void* dialog_item::data

4.5.1.2 int dialog_item::selected

4.5.1.3 char dialog_item::str[[MAXITEMSTR](#)]

4.5.1.4 char dialog_item::tag

The documentation for this struct was generated from the following file:

- menuconfig/lxdialog/[dialog.h](#)

4.6 dialog_list Struct Reference

```
#include <dialog.h>
```

Public Attributes

- struct [dialog_item](#) node
- struct [dialog_list](#) * next

4.6.1 Member Data Documentation

4.6.1.1 struct [dialog_list](#)* [dialog_list::next](#)

4.6.1.2 struct [dialog_item](#) [dialog_list::node](#)

The documentation for this struct was generated from the following file:

- menuconfig/lxdialog/[dialog.h](#)

4.7 expr Struct Reference

```
#include <expr.h>
```

Public Attributes

- enum [expr_type](#) type
- union [expr_data](#) left right

4.7.1 Member Data Documentation

4.7.1.1 union [expr_data](#) left [expr::right](#)

4.7.1.2 enum [expr_type](#) [expr::type](#)

The documentation for this struct was generated from the following file:

- menuconfig/[expr.h](#)

4.8 expr_data Union Reference

```
#include <expr.h>
```

Public Attributes

- struct [expr](#) * [expr](#)
- struct [symbol](#) * [sym](#)

4.8.1 Member Data Documentation

4.8.1.1 struct [expr](#)* [expr_data::expr](#)

4.8.1.2 struct [symbol](#)* [expr_data::sym](#)

The documentation for this union was generated from the following file:

- [menuconfig/expr.h](#)

4.9 `expr_value` Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct `expr` * `expr`
- `tristate` `tri`

4.9.1 Member Data Documentation

4.9.1.1 `struct expr* expr_value::expr`

4.9.1.2 `tristate expr_value::tri`

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.10 `file` Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct `file` * `next`
- struct `file` * `parent`
- const char * `name`
- char * `abs_name`
- int `lineno`

4.10.1 Member Data Documentation

4.10.1.1 `char* file::abs_name`

4.10.1.2 `int file::lineno`

4.10.1.3 `const char* file::name`

4.10.1.4 `struct file* file::next`

4.10.1.5 `struct file* file::parent`

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.11 gstr Struct Reference

```
#include <lkc.h>
```

Public Attributes

- `size_t` [len](#)
- `char *` [s](#)
- `int` [max_width](#)

4.11.1 Member Data Documentation

4.11.1.1 `size_t` [gstr::len](#)

4.11.1.2 `int` [gstr::max_width](#)

4.11.1.3 `char*` [gstr::s](#)

The documentation for this struct was generated from the following file:

- [menuconfig/lkc.h](#)

4.12 item Struct Reference

Public Attributes

- `struct item *` [next](#)
- `unsigned int` [len](#)
- `unsigned int` [hash](#)
- `char` [name](#) [0]

4.12.1 Member Data Documentation

4.12.1.1 `unsigned int` [item::hash](#)

4.12.1.2 `unsigned int` [item::len](#)

4.12.1.3 `char` [item::name](#)[0]

4.12.1.4 `struct item*` [item::next](#)

The documentation for this struct was generated from the following file:

- [menuconfig/basic/fixdep.c](#)

4.13 jump_key Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct [list_head](#) [entries](#)
- [size_t](#) [offset](#)
- struct [menu](#) * [target](#)
- [int](#) [index](#)

4.13.1 Member Data Documentation

4.13.1.1 struct [list_head](#) [jump_key::entries](#)

4.13.1.2 [int](#) [jump_key::index](#)

4.13.1.3 [size_t](#) [jump_key::offset](#)

4.13.1.4 struct [menu](#)* [jump_key::target](#)

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.14 kconf_id Struct Reference

```
#include <lkc.h>
```

Public Attributes

- [int](#) [name](#)
- [int](#) [token](#)
- [unsigned int](#) [flags](#)
- [enum](#) [symbol_type](#) [stype](#)

4.14.1 Member Data Documentation

4.14.1.1 [unsigned int](#) [kconf_id::flags](#)

4.14.1.2 [int](#) [kconf_id::name](#)

4.14.1.3 [enum](#) [symbol_type](#) [kconf_id::stype](#)

4.14.1.4 [int](#) [kconf_id::token](#)

The documentation for this struct was generated from the following file:

- [menuconfig/lkc.h](#)

4.15 kconf_id_strings_t Struct Reference

Public Attributes

- [char](#) [kconf_id_strings_str2](#) [sizeof("if")]

- char [kconf_id_strings_str3](#) [sizeof("int")]
- char [kconf_id_strings_str5](#) [sizeof("endif")]
- char [kconf_id_strings_str7](#) [sizeof("default")]
- char [kconf_id_strings_str8](#) [sizeof("tristate")]
- char [kconf_id_strings_str9](#) [sizeof("endchoice")]
- char [kconf_id_strings_str12](#) [sizeof("def_tristate")]
- char [kconf_id_strings_str13](#) [sizeof("def_bool")]
- char [kconf_id_strings_str14](#) [sizeof("defconfig_list")]
- char [kconf_id_strings_str17](#) [sizeof("on")]
- char [kconf_id_strings_str18](#) [sizeof("optional")]
- char [kconf_id_strings_str21](#) [sizeof("option")]
- char [kconf_id_strings_str22](#) [sizeof("endmenu")]
- char [kconf_id_strings_str23](#) [sizeof("mainmenu")]
- char [kconf_id_strings_str25](#) [sizeof("menuconfig")]
- char [kconf_id_strings_str27](#) [sizeof("modules")]
- char [kconf_id_strings_str29](#) [sizeof("menu")]
- char [kconf_id_strings_str31](#) [sizeof("select")]
- char [kconf_id_strings_str32](#) [sizeof("comment")]
- char [kconf_id_strings_str33](#) [sizeof("env")]
- char [kconf_id_strings_str35](#) [sizeof("range")]
- char [kconf_id_strings_str36](#) [sizeof("choice")]
- char [kconf_id_strings_str39](#) [sizeof("bool")]
- char [kconf_id_strings_str41](#) [sizeof("source")]
- char [kconf_id_strings_str42](#) [sizeof("visible")]
- char [kconf_id_strings_str43](#) [sizeof("hex")]
- char [kconf_id_strings_str46](#) [sizeof("config")]
- char [kconf_id_strings_str47](#) [sizeof("boolean")]
- char [kconf_id_strings_str51](#) [sizeof("string")]
- char [kconf_id_strings_str54](#) [sizeof("help")]
- char [kconf_id_strings_str56](#) [sizeof("prompt")]
- char [kconf_id_strings_str72](#) [sizeof("depends")]

4.15.1 Member Data Documentation

- 4.15.1.1 char [kconf_id_strings_t::kconf_id_strings_str12](#)[sizeof("def_tristate")]
- 4.15.1.2 char [kconf_id_strings_t::kconf_id_strings_str13](#)[sizeof("def_bool")]
- 4.15.1.3 char [kconf_id_strings_t::kconf_id_strings_str14](#)[sizeof("defconfig_list")]
- 4.15.1.4 char [kconf_id_strings_t::kconf_id_strings_str17](#)[sizeof("on")]
- 4.15.1.5 char [kconf_id_strings_t::kconf_id_strings_str18](#)[sizeof("optional")]
- 4.15.1.6 char [kconf_id_strings_t::kconf_id_strings_str2](#)[sizeof("if")]
- 4.15.1.7 char [kconf_id_strings_t::kconf_id_strings_str21](#)[sizeof("option")]
- 4.15.1.8 char [kconf_id_strings_t::kconf_id_strings_str22](#)[sizeof("endmenu")]
- 4.15.1.9 char [kconf_id_strings_t::kconf_id_strings_str23](#)[sizeof("mainmenu")]
- 4.15.1.10 char [kconf_id_strings_t::kconf_id_strings_str25](#)[sizeof("menuconfig")]

```

4.15.1.11 char kconf_id_strings_t::kconf_id_strings_str27[sizeof("modules")]
4.15.1.12 char kconf_id_strings_t::kconf_id_strings_str29[sizeof("menu")]
4.15.1.13 char kconf_id_strings_t::kconf_id_strings_str3[sizeof("int")]
4.15.1.14 char kconf_id_strings_t::kconf_id_strings_str31[sizeof("select")]
4.15.1.15 char kconf_id_strings_t::kconf_id_strings_str32[sizeof("comment")]
4.15.1.16 char kconf_id_strings_t::kconf_id_strings_str33[sizeof("env")]
4.15.1.17 char kconf_id_strings_t::kconf_id_strings_str35[sizeof("range")]
4.15.1.18 char kconf_id_strings_t::kconf_id_strings_str36[sizeof("choice")]
4.15.1.19 char kconf_id_strings_t::kconf_id_strings_str39[sizeof("bool")]
4.15.1.20 char kconf_id_strings_t::kconf_id_strings_str41[sizeof("source")]
4.15.1.21 char kconf_id_strings_t::kconf_id_strings_str42[sizeof("visible")]
4.15.1.22 char kconf_id_strings_t::kconf_id_strings_str43[sizeof("hex")]
4.15.1.23 char kconf_id_strings_t::kconf_id_strings_str46[sizeof("config")]
4.15.1.24 char kconf_id_strings_t::kconf_id_strings_str47[sizeof("boolean")]
4.15.1.25 char kconf_id_strings_t::kconf_id_strings_str5[sizeof("endif")]
4.15.1.26 char kconf_id_strings_t::kconf_id_strings_str51[sizeof("string")]
4.15.1.27 char kconf_id_strings_t::kconf_id_strings_str54[sizeof("help")]
4.15.1.28 char kconf_id_strings_t::kconf_id_strings_str56[sizeof("prompt")]
4.15.1.29 char kconf_id_strings_t::kconf_id_strings_str7[sizeof("default")]
4.15.1.30 char kconf_id_strings_t::kconf_id_strings_str72[sizeof("depends")]
4.15.1.31 char kconf_id_strings_t::kconf_id_strings_str8[sizeof("tristate")]
4.15.1.32 char kconf_id_strings_t::kconf_id_strings_str9[sizeof("endchoice")]

```

The documentation for this struct was generated from the following file:

- [menuconfig/zconf.hash.c](#)

4.16 list_head Struct Reference

```
#include <list.h>
```

Public Attributes

- struct [list_head](#) * [next](#)

- struct [list_head](#) * [prev](#)

4.16.1 Member Data Documentation

4.16.1.1 struct [list_head](#)* [list_head::next](#)

4.16.1.2 struct [list_head](#) * [list_head::prev](#)

The documentation for this struct was generated from the following file:

- [menuconfig/list.h](#)

4.17 menu Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct [menu](#) * [next](#)
- struct [menu](#) * [parent](#)
- struct [menu](#) * [list](#)
- struct [symbol](#) * [sym](#)
- struct [property](#) * [prompt](#)
- struct [expr](#) * [visibility](#)
- struct [expr](#) * [dep](#)
- unsigned int [flags](#)
- char * [help](#)
- struct [file](#) * [file](#)
- int [lineno](#)
- void * [data](#)

4.17.1 Member Data Documentation

4.17.1.1 void* [menu::data](#)

4.17.1.2 struct [expr](#)* [menu::dep](#)

4.17.1.3 struct [file](#)* [menu::file](#)

4.17.1.4 unsigned int [menu::flags](#)

4.17.1.5 char* [menu::help](#)

4.17.1.6 int [menu::lineno](#)

4.17.1.7 struct [menu](#)* [menu::list](#)

4.17.1.8 struct [menu](#)* [menu::next](#)

4.17.1.9 struct [menu](#)* [menu::parent](#)

4.17.1.10 struct [property](#)* [menu::prompt](#)

4.17.1.11 struct symbol* menu::sym

4.17.1.12 struct expr* menu::visibility

The documentation for this struct was generated from the following file:

- menuconfig/[expr.h](#)

4.18 out_fct_wrap_type Struct Reference

Public Attributes

- void(* [fct](#))(char character, void *[arg](#))
- void * [arg](#)

4.18.1 Member Data Documentation

4.18.1.1 void* out_fct_wrap_type::arg

4.18.1.2 void(* out_fct_wrap_type::fct)(char character, void *[arg](#))

The documentation for this struct was generated from the following file:

- printf/[printf.c](#)

4.19 property Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct [property](#) * [next](#)
- struct [symbol](#) * [sym](#)
- enum [prop_type](#) [type](#)
- const char * [text](#)
- struct [expr_value](#) [visible](#)
- struct [expr](#) * [expr](#)
- struct [menu](#) * [menu](#)
- struct [file](#) * [file](#)
- int [lineno](#)

4.19.1 Member Data Documentation

4.19.1.1 struct expr* property::expr

4.19.1.2 struct file* property::file

4.19.1.3 int property::lineno

4.19.1.4 struct menu* property::menu

4.19.1.5 struct property* property::next

4.19.1.6 struct symbol* property::sym

4.19.1.7 const char* property::text

4.19.1.8 enum prop_type property::type

4.19.1.9 struct expr_value property::visible

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.20 search_data Struct Reference

Public Attributes

- struct [list_head](#) * [head](#)
- struct [menu](#) ** [targets](#)
- int * [keys](#)

4.20.1 Member Data Documentation

4.20.1.1 struct [list_head](#)* [search_data::head](#)

4.20.1.2 int* [search_data::keys](#)

4.20.1.3 struct [menu](#)** [search_data::targets](#)

The documentation for this struct was generated from the following file:

- [menuconfig/mconf.c](#)

4.21 symbol Struct Reference

```
#include <expr.h>
```

Public Attributes

- struct [symbol](#) * [next](#)
- char * [name](#)
- enum [symbol_type](#) [type](#)
- struct [symbol_value](#) [curr](#)
- struct [symbol_value](#) [def](#) [[S_DEF_COUNT](#)]
- [tristate](#) [visible](#)
- int [flags](#)
- struct [property](#) * [prop](#)
- struct [expr_value](#) [dir_dep](#)
- struct [expr_value](#) [rev_dep](#)

4.21.1 Member Data Documentation

4.21.1.1 struct `symbol_value` `symbol::curr`

4.21.1.2 struct `symbol_value` `symbol::def[S_DEF_COUNT]`

4.21.1.3 struct `expr_value` `symbol::dir_dep`

4.21.1.4 int `symbol::flags`

4.21.1.5 char* `symbol::name`

4.21.1.6 struct `symbol*` `symbol::next`

4.21.1.7 struct `property*` `symbol::prop`

4.21.1.8 struct `expr_value` `symbol::rev_dep`

4.21.1.9 enum `symbol_type` `symbol::type`

4.21.1.10 tristate `symbol::visible`

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.22 symbol_value Struct Reference

```
#include <expr.h>
```

Public Attributes

- void * [val](#)
- tristate [tri](#)

4.22.1 Member Data Documentation

4.22.1.1 tristate `symbol_value::tri`

4.22.1.2 void* `symbol_value::val`

The documentation for this struct was generated from the following file:

- [menuconfig/expr.h](#)

4.23 yy_buffer_state Struct Reference

Public Attributes

- FILE * [yy_input_file](#)
- char * [yy_ch_buf](#)
- char * [yy_buf_pos](#)

- [yy_size_t yy_buf_size](#)
- [int yy_n_chars](#)
- [int yy_is_our_buffer](#)
- [int yy_is_interactive](#)
- [int yy_at_bol](#)
- [int yy_bs_lineno](#)
- [int yy_bs_column](#)
- [int yy_fill_buffer](#)
- [int yy_buffer_status](#)

4.23.1 Member Data Documentation

4.23.1.1 [int yy_buffer_state::yy_at_bol](#)

4.23.1.2 [int yy_buffer_state::yy_bs_column](#)

The column count.

4.23.1.3 [int yy_buffer_state::yy_bs_lineno](#)

The line count.

4.23.1.4 [char* yy_buffer_state::yy_buf_pos](#)

4.23.1.5 [yy_size_t yy_buffer_state::yy_buf_size](#)

4.23.1.6 [int yy_buffer_state::yy_buffer_status](#)

4.23.1.7 [char* yy_buffer_state::yy_ch_buf](#)

4.23.1.8 [int yy_buffer_state::yy_fill_buffer](#)

4.23.1.9 [FILE* yy_buffer_state::yy_input_file](#)

4.23.1.10 [int yy_buffer_state::yy_is_interactive](#)

4.23.1.11 [int yy_buffer_state::yy_is_our_buffer](#)

4.23.1.12 [int yy_buffer_state::yy_n_chars](#)

The documentation for this struct was generated from the following file:

- [menuconfig/zconf.lex.c](#)

4.24 yy_trans_info Struct Reference

Public Attributes

- [flex_int32_t yy_verify](#)
- [flex_int32_t yy_nxt](#)

4.24.1 Member Data Documentation

4.24.1.1 `flex_int32_t yy_trans_info::yy_nxt`

4.24.1.2 `flex_int32_t yy_trans_info::yy_verify`

The documentation for this struct was generated from the following file:

- `menuconfig/zconf.lex.c`

4.25 yyalloc Union Reference

Public Attributes

- `yytype_int16 yyss_alloc`
- `YYSTYPE yyvs_alloc`

4.25.1 Member Data Documentation

4.25.1.1 `yytype_int16 yyalloc::yyss_alloc`

4.25.1.2 `YYSTYPE yyalloc::yyvs_alloc`

The documentation for this union was generated from the following file:

- `menuconfig/zconf.tab.c`

4.26 YYSTYPE Union Reference

Public Attributes

- `char * string`
- `struct file * file`
- `struct symbol * symbol`
- `struct expr * expr`
- `struct menu * menu`
- `const struct kconf_id * id`

4.26.1 Member Data Documentation

4.26.1.1 `struct expr* YYSTYPE::expr`

4.26.1.2 `struct file* YYSTYPE::file`

4.26.1.3 `const struct kconf_id* YYSTYPE::id`

4.26.1.4 `struct menu* YYSTYPE::menu`

4.26.1.5 `char* YYSTYPE::string`

4.26.1.6 `struct symbol* YYSTYPE::symbol`

The documentation for this union was generated from the following file:

- menuconfig/[zconf.tab.c](#)

Chapter 5

File Documentation

5.1 autoconf.h File Reference

Macros

- `#define CONFIG_UART_BAUD_RATE 9600`
- `#define CONFIG_LED_BLINK_ITERATOR_50000 1`

5.1.1 Macro Definition Documentation

5.1.1.1 `#define CONFIG_LED_BLINK_ITERATOR_50000 1`

5.1.1.2 `#define CONFIG_UART_BAUD_RATE 9600`

5.2 led/led.c File Reference

```
#include "led.h"
```

Functions

- void `led_init` (void)
This function initializes a single pin from a GPIO port where an LED is connected.
- void `led_set` (void)
This function sets a logic high on the respective pin.
- void `led_clear` (void)
This function clears the respective pin a logic low.

Variables

- volatile uint32_t * `ahb1enr` = (volatile uint32_t *)0x40023830
A variable that corresponds to the RCC_AHB1ENR register.
- volatile uint32_t * `gpio_a_moder` = (volatile uint32_t *)0x40020000
A variable that corresponds to the GPIOA_MODER register.
- volatile uint32_t * `gpio_a_otyper` = (volatile uint32_t *)0x40020004
A variable that corresponds to the GPIOA_TYPER register.
- volatile uint32_t * `gpio_a_odr` = (volatile uint32_t *)0x40020014
A variable that corresponds to the GPIOA_ODR register.

5.2.1 Function Documentation

5.2.1.1 void led_clear (void)

This function clears the respective pin a logic low.

Returns

None.

5.2.1.2 void led_init (void)

This function initializes a single pin from a GPIO port where an LED is connected.

To do an init, first the clock to the GPIO module has to be enabled. Second, the respective GPIO has to be configured as an OUTPUT. Third, the output must be configured to be a push-pull. Finally, the GPIO is turn ON upon initialization.

Returns

None.

5.2.1.3 void led_set (void)

This function sets a logic high on the respective pin.

Returns

None.

5.2.2 Variable Documentation

5.2.2.1 ahb1enr = (volatile uint32_t *)0x40023830

A variable that corresponds to the RCC_AHB1ENR register.

5.2.2.2 gpio_a_moder = (volatile uint32_t *)0x40020000

A variable that corresponds to the GPIOA_MODER register.

5.2.2.3 gpio_a_odr = (volatile uint32_t *)0x40020014

A variable that corresponds to the GPIOA_ODR register.

5.2.2.4 gpio_a_otyper = (volatile uint32_t *)0x40020004

A variable that corresponds to the GPIOA_TYPER register.

5.3 led/led.h File Reference

```
#include <stdint.h>
```

Macros

- `#define RCC_AHB1ENR_GPIOA_EN 0x01`
A bitmask for the GPIO_EN bit.
- `#define GPIOA_MODER_PA12 0x1000000`
A bitmask for the MODER_PA12 bit.
- `#define GPIOA_OTYPER_PA12 0x1000`
A bitmask for the OTYPER_PA12 bit.
- `#define GPIOA_ODR_PA12 0x1000`
A bitmask for the ODR_PA12 bit.

Functions

- `void led_init (void)`
This function initializes a single pin from a GPIO port where an LED is connected.
- `void led_set (void)`
This function sets a logic high on the respective pin.
- `void led_clear (void)`
This function clears the respective pin a logic low.

5.3.1 Macro Definition Documentation

5.3.1.1 `#define GPIOA_MODER_PA12 0x1000000`

A bitmask for the MODER_PA12 bit.

5.3.1.2 `#define GPIOA_ODR_PA12 0x1000`

A bitmask for the ODR_PA12 bit.

5.3.1.3 `#define GPIOA_OTYPER_PA12 0x1000`

A bitmask for the OTYPER_PA12 bit.

5.3.1.4 `#define RCC_AHB1ENR_GPIOA_EN 0x01`

A bitmask for the GPIO_EN bit.

5.3.2 Function Documentation

5.3.2.1 `void led_clear (void)`

This function clears the respective pin a logic low.

Returns

None.

5.3.2.2 void led_init (void)

This function initializes a single pin from a GPIO port where an LED is connected.

To do an init, first the clock to the GPIO module has to be enabled. Second, the respective GPIO has to be configured as an OUTPUT. Third, the output must be configured to be a push-pull. Finally, the GPIO is turn ON upon initialization.

Returns

None.

5.3.2.3 void led_set (void)

This function sets a logic high on the respective pin.

Returns

None.

5.4 main.c File Reference

```
#include "autoconf.h"
#include "main.h"
#include "led.h"
#include "uart.h"
#include "printf.h"
```

Macros

- `#define LED_BLINK 50000`

Functions

- `int main (void)`

The main entry of the application. This function initializes a GPIO and a UART, then outputs a message over the UART and starts blinking the diode.

5.4.1 Macro Definition Documentation

5.4.1.1 #define LED_BLINK 50000

5.4.2 Function Documentation

5.4.2.1 int main (void)

The main entry of the application. This function initializes a GPIO and a UART, then outputs a message over the UART and starts blinking the diode.

Returns

None.

5.5 main.h File Reference

```
#include <stdint.h>
#include <string.h>
```

5.6 menuconfig/basic/fixdep.c File Reference

```
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/mman.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <ctype.h>
#include <arpa/inet.h>
```

Classes

- struct [item](#)

Macros

- #define [INT_CONF](#) ntohl(0x434f4e46)
- #define [INT_ONFI](#) ntohl(0x4f4e4649)
- #define [INT_NFIG](#) ntohl(0x4e464947)
- #define [INT_FIG_](#) ntohl(0x4649475f)
- #define [HASHSZ](#) 256

Functions

- int [main](#) (int argc, char *argv[])

Variables

- char * [target](#)
- char * [depfile](#)
- char * [cmdline](#)

5.6.1 Macro Definition Documentation

5.6.1.1 #define HASHSZ 256

5.6.1.2 #define INT_CONF ntohl(0x434f4e46)

5.6.1.3 #define INT_FIG_ ntohl(0x4649475f)

5.6.1.4 `#define INT_NFIG ntohs(0x4e464947)`

5.6.1.5 `#define INT_ONFI ntohs(0x4f4e4649)`

5.6.2 Function Documentation

5.6.2.1 `int main (int argc, char * argv[])`

5.6.3 Variable Documentation

5.6.3.1 `char* cmdline`

5.6.3.2 `char* depfile`

5.6.3.3 `char* target`

5.7 menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdC/CMakeCCompilerId.c File Reference

Macros

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`
- `#define DEC(n)`
- `#define HEX(n)`

Functions

- `int main (int argc, char *argv[])`

Variables

- `char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

5.7.1 Macro Definition Documentation

5.7.1.1 `#define ARCHITECTURE_ID ""`

5.7.1.2 `#define COMPILER_ID ""`

5.7.1.3 `#define DEC(n)`

Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

5.7.1.4 #define HEX(n)

Value:

```
( '0' + ((n)>>28 & 0xF) ), \
( '0' + ((n)>>24 & 0xF) ), \
( '0' + ((n)>>20 & 0xF) ), \
( '0' + ((n)>>16 & 0xF) ), \
( '0' + ((n)>>12 & 0xF) ), \
( '0' + ((n)>>8  & 0xF) ), \
( '0' + ((n)>>4  & 0xF) ), \
( '0' + ((n)      & 0xF) )
```

5.7.1.5 #define PLATFORM_ID ""

5.7.2 Function Documentation

5.7.2.1 int main (int argc, char * argv[])

5.7.3 Variable Documentation

5.7.3.1 char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"

5.7.3.2 char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"

5.7.3.3 char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"

5.8 menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- #define COMPILER_ID ""
- #define PLATFORM_ID ""
- #define ARCHITECTURE_ID ""
- #define DEC(n)
- #define HEX(n)

Functions

- int main (int argc, char *argv[])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"

5.8.1 Macro Definition Documentation

5.8.1.1 #define ARCHITECTURE_ID ""

5.8.1.2 #define COMPILER_ID ""

5.8.1.3 #define DEC(n)

Value:

```
( '0' + ((n) / 10000000) % 10 ), \
( '0' + ((n) / 1000000) % 10 ), \
( '0' + ((n) / 100000) % 10 ), \
( '0' + ((n) / 10000) % 10 ), \
( '0' + ((n) / 1000) % 10 ), \
( '0' + ((n) / 100) % 10 ), \
( '0' + ((n) / 10) % 10 ), \
( '0' + ((n) % 10) )
```

5.8.1.4 #define HEX(n)

Value:

```
( '0' + ((n) >> 28 & 0xF) ), \
( '0' + ((n) >> 24 & 0xF) ), \
( '0' + ((n) >> 20 & 0xF) ), \
( '0' + ((n) >> 16 & 0xF) ), \
( '0' + ((n) >> 12 & 0xF) ), \
( '0' + ((n) >> 8 & 0xF) ), \
( '0' + ((n) >> 4 & 0xF) ), \
( '0' + ((n) & 0xF) )
```

5.8.1.5 #define PLATFORM_ID ""

5.8.2 Function Documentation

5.8.2.1 int main (int argc, char * argv[])

5.8.3 Variable Documentation

5.8.3.1 char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"

5.8.3.2 char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"

5.8.3.3 char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"

5.9 menuconfig/build/tmp/gen/choice1.h File Reference

5.10 menuconfig/build/tmp/menudependency.h File Reference

5.11 tmp/menudependency.h File Reference

5.12 menuconfig/build/tmp/num/param.h File Reference

5.13 menuconfig/build/tmp/string/param.h File Reference

5.14 tmp/num/param.h File Reference

5.15 tmp/string/param.h File Reference

5.16 menuconfig/build/tmp/num/param2.h File Reference

5.17 menuconfig/conf.c File Reference

```
#include <locale.h>
#include <ctype.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include <getopt.h>
#include <sys/stat.h>
#include <sys/time.h>
#include "lkc.h"
```

Enumerations

- enum `input_mode` {
 `oldaskconfig`, `silentoldconfig`, `oldconfig`, `allnoconfig`,
 `allyesconfig`, `allmodconfig`, `alldefconfig`, `randconfig`,
 `defconfig`, `savedefconfig`, `listnewconfig`, `olddefconfig` }

Functions

- int `main` (int ac, char **av)

Variables

- enum `input_mode` `input_mode` = `oldaskconfig`

5.17.1 Enumeration Type Documentation

5.17.1.1 enum `input_mode`

Enumerator

oldaskconfig
silentoldconfig
oldconfig
allnoconfig
allyesconfig
allmodconfig
alldefconfig
randconfig
defconfig
savedefconfig
listnewconfig
olddefconfig

5.17.2 Function Documentation

5.17.2.1 `int main (int ac, char ** av)`

5.17.3 Variable Documentation

5.17.3.1 `enum input_mode input_mode = oldaskconfig`

5.18 menuconfig/confdata.c File Reference

```
#include <sys/stat.h>
#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
#include "lkc.h"
```

Macros

- `#define LINE_GROWTH 16`

Functions

- void `conf_set_message_callback` (void(*fn)(const char *fmt, va_list ap))
- const char * `conf_get_configname` (void)
- const char * `conf_get_autoconfig_name` (void)
- char * `conf_get_default_confname` (void)
- int `conf_read_simple` (const char *name, int def)
- int `conf_read` (const char *name)
- int `conf_write_defconfig` (const char *filename)
- int `conf_write` (const char *name)
- int `conf_write_autoconf` (const char *filename)
- void `sym_set_change_count` (int count)
- void `sym_add_change_count` (int count)
- bool `conf_get_changed` (void)
- void `conf_set_changed_callback` (void(*fn)(void))
- void `conf_set_all_new_symbols` (enum `conf_def_mode` mode)

Variables

- const char `conf_defname` [] = "arch/\$ARCH/defconfig"

5.18.1 Macro Definition Documentation

5.18.1.1 `#define LINE_GROWTH 16`

5.18.2 Function Documentation

- 5.18.2.1 `const char* conf_get_autoconfig_name (void)`
- 5.18.2.2 `bool conf_get_changed (void)`
- 5.18.2.3 `const char* conf_get_configname (void)`
- 5.18.2.4 `char* conf_get_default_confname (void)`
- 5.18.2.5 `int conf_read (const char * name)`
- 5.18.2.6 `int conf_read_simple (const char * name, int def)`
- 5.18.2.7 `void conf_set_all_new_symbols (enum conf_def_mode mode)`
- 5.18.2.8 `void conf_set_changed_callback (void(*)(void) fn)`
- 5.18.2.9 `void conf_set_message_callback (void(*)(const char *fmt, va_list ap) fn)`
- 5.18.2.10 `int conf_write (const char * name)`
- 5.18.2.11 `int conf_write_autoconf (const char * filename)`
- 5.18.2.12 `int conf_write_defconfig (const char * filename)`
- 5.18.2.13 `void sym_add_change_count (int count)`
- 5.18.2.14 `void sym_set_change_count (int count)`

5.18.3 Variable Documentation

- 5.18.3.1 `const char conf_defname[] = "arch/$ARCH/defconfig"`

5.19 menuconfig/expr.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "lkc.h"
```

Macros

- `#define DEBUG_EXPR 0`
- `#define e1 (*ep1)`
- `#define e2 (*ep2)`
- `#define e1 (*ep1)`
- `#define e2 (*ep2)`
- `#define e1 (*ep1)`
- `#define e2 (*ep2)`
- `#define e1 (*ep1)`
- `#define e2 (*ep2)`

Functions

- struct `expr` * `expr_alloc_symbol` (struct `symbol` *`sym`)
- struct `expr` * `expr_alloc_one` (enum `expr_type` `type`, struct `expr` *`ce`)
- struct `expr` * `expr_alloc_two` (enum `expr_type` `type`, struct `expr` *`e1`, struct `expr` *`e2`)
- struct `expr` * `expr_alloc_comp` (enum `expr_type` `type`, struct `symbol` *`s1`, struct `symbol` *`s2`)
- struct `expr` * `expr_alloc_and` (struct `expr` *`e1`, struct `expr` *`e2`)
- struct `expr` * `expr_alloc_or` (struct `expr` *`e1`, struct `expr` *`e2`)
- struct `expr` * `expr_copy` (const struct `expr` *`org`)
- void `expr_free` (struct `expr` *`e`)
- void `expr_eliminate_eq` (struct `expr` **`ep1`, struct `expr` **`ep2`)
- int `expr_eq` (struct `expr` *`e1`, struct `expr` *`e2`)
- struct `expr` * `expr_eliminate_yn` (struct `expr` *`e`)
- struct `expr` * `expr_trans_bool` (struct `expr` *`e`)
- struct `expr` * `expr_eliminate_dups` (struct `expr` *`e`)
- struct `expr` * `expr_transform` (struct `expr` *`e`)
- int `expr_contains_symbol` (struct `expr` *`dep`, struct `symbol` *`sym`)
- bool `expr_depends_symbol` (struct `expr` *`dep`, struct `symbol` *`sym`)
- struct `expr` * `expr_extract_eq_and` (struct `expr` **`ep1`, struct `expr` **`ep2`)
- struct `expr` * `expr_extract_eq_or` (struct `expr` **`ep1`, struct `expr` **`ep2`)
- void `expr_extract_eq` (enum `expr_type` `type`, struct `expr` **`ep`, struct `expr` **`ep1`, struct `expr` **`ep2`)
- struct `expr` * `expr_trans_compare` (struct `expr` *`e`, enum `expr_type` `type`, struct `symbol` *`sym`)
- tristate `expr_calc_value` (struct `expr` *`e`)
- int `expr_compare_type` (enum `expr_type` `t1`, enum `expr_type` `t2`)
- struct `expr` * `expr_simplify_unmet_dep` (struct `expr` *`e1`, struct `expr` *`e2`)
- void `expr_print` (struct `expr` *`e`, void(*fn)(void *, struct `symbol` *, const char *), void *`data`, int `prevtoken`)
- void `expr_fprint` (struct `expr` *`e`, FILE *`out`)
- void `expr_gstr_print` (struct `expr` *`e`, struct `gstr` *`gs`)

5.19.1 Macro Definition Documentation

5.19.1.1 `#define DEBUG_EXPR 0`

5.19.1.2 `#define e1 (*ep1)`

5.19.1.3 `#define e1 (*ep1)`

5.19.1.4 `#define e1 (*ep1)`

5.19.1.5 `#define e1 (*ep1)`

5.19.1.6 `#define e2 (*ep2)`

5.19.1.7 `#define e2 (*ep2)`

5.19.1.8 `#define e2 (*ep2)`

5.19.1.9 `#define e2 (*ep2)`

5.19.2 Function Documentation

5.19.2.1 struct `expr`* `expr_alloc_and` (struct `expr` * `e1`, struct `expr` * `e2`)

5.19.2.2 struct `expr`* `expr_alloc_comp` (enum `expr_type` `type`, struct `symbol` * `s1`, struct `symbol` * `s2`)

- 5.19.2.3 `struct expr* expr_alloc_one (enum expr_type type, struct expr * ce)`
- 5.19.2.4 `struct expr* expr_alloc_or (struct expr * e1, struct expr * e2)`
- 5.19.2.5 `struct expr* expr_alloc_symbol (struct symbol * sym)`
- 5.19.2.6 `struct expr* expr_alloc_two (enum expr_type type, struct expr * e1, struct expr * e2)`
- 5.19.2.7 `tristate expr_calc_value (struct expr * e)`
- 5.19.2.8 `int expr_compare_type (enum expr_type t1, enum expr_type t2)`
- 5.19.2.9 `int expr_contains_symbol (struct expr * dep, struct symbol * sym)`
- 5.19.2.10 `struct expr* expr_copy (const struct expr * org)`
- 5.19.2.11 `bool expr_depends_symbol (struct expr * dep, struct symbol * sym)`
- 5.19.2.12 `struct expr* expr_eliminate_dups (struct expr * e)`
- 5.19.2.13 `void expr_eliminate_eq (struct expr ** ep1, struct expr ** ep2)`
- 5.19.2.14 `struct expr* expr_eliminate_yn (struct expr * e)`
- 5.19.2.15 `int expr_eq (struct expr * e1, struct expr * e2)`
- 5.19.2.16 `void expr_extract_eq (enum expr_type type, struct expr ** ep, struct expr ** ep1, struct expr ** ep2)`
- 5.19.2.17 `struct expr* expr_extract_eq_and (struct expr ** ep1, struct expr ** ep2)`
- 5.19.2.18 `struct expr* expr_extract_eq_or (struct expr ** ep1, struct expr ** ep2)`
- 5.19.2.19 `void expr_fprint (struct expr * e, FILE * out)`
- 5.19.2.20 `void expr_free (struct expr * e)`
- 5.19.2.21 `void expr_gstr_print (struct expr * e, struct gstr * gs)`
- 5.19.2.22 `void expr_print (struct expr * e, void(*)(void *, struct symbol *, const char *) fn, void * data, int prevtoken)`
- 5.19.2.23 `struct expr* expr_simplify_unmet_dep (struct expr * e1, struct expr * e2)`
- 5.19.2.24 `struct expr* expr_trans_bool (struct expr * e)`
- 5.19.2.25 `struct expr* expr_trans_compare (struct expr * e, enum expr_type type, struct symbol * sym)`
- 5.19.2.26 `struct expr* expr_transform (struct expr * e)`

5.20 menuconfig/expr.h File Reference

```
#include <assert.h>
#include <stdio.h>
#include "list.h"
#include <stdbool.h>
```

Classes

- struct [file](#)
- union [expr_data](#)
- struct [expr](#)
- struct [expr_value](#)
- struct [symbol_value](#)
- struct [symbol](#)
- struct [property](#)
- struct [menu](#)
- struct [jump_key](#)

Macros

- #define [EXPR_OR](#)(dep1, dep2) (((dep1)>(dep2))?(dep1):(dep2))
- #define [EXPR_AND](#)(dep1, dep2) (((dep1)<(dep2))?(dep1):(dep2))
- #define [EXPR_NOT](#)(dep) (2-(dep))
- #define [expr_list_for_each_sym](#)(l, e, s) for (e = (l); e && (s = e->right.sym); e = e->left.expr)
- #define [for_all_symbols](#)(i, sym) for (i = 0; i < [SYMBOL_HASHSIZE](#); i++) for (sym = [symbol_hash](#)[i]; sym; sym = sym->next) if (sym->type != [S_OTHER](#))
- #define [SYMBOL_CONST](#) 0x0001 /* symbol is const */
- #define [SYMBOL_CHECK](#) 0x0008 /* used during dependency checking */
- #define [SYMBOL_CHOICE](#) 0x0010 /* start of a choice block (null name) */
- #define [SYMBOL_CHOICEVAL](#) 0x0020 /* used as a value in a choice block */
- #define [SYMBOL_VALID](#) 0x0080 /* set when symbol.curr is calculated */
- #define [SYMBOL_OPTIONAL](#) 0x0100 /* choice is optional - values can be 'n' */
- #define [SYMBOL_WRITE](#) 0x0200 /* ? */
- #define [SYMBOL_CHANGED](#) 0x0400 /* ? */
- #define [SYMBOL_AUTO](#) 0x1000 /* value from environment variable */
- #define [SYMBOL_CHECKED](#) 0x2000 /* used during dependency checking */
- #define [SYMBOL_WARNED](#) 0x8000 /* warning has been issued */
- #define [SYMBOL_DEF](#) 0x10000 /* First bit of SYMBOL_DEF */
- #define [SYMBOL_DEF_USER](#) 0x10000 /* symbol.def[S_DEF_USER] is valid */
- #define [SYMBOL_DEF_AUTO](#) 0x20000 /* symbol.def[S_DEF_AUTO] is valid */
- #define [SYMBOL_DEF3](#) 0x40000 /* symbol.def[S_DEF_3] is valid */
- #define [SYMBOL_DEF4](#) 0x80000 /* symbol.def[S_DEF_4] is valid */
- #define [SYMBOL_MAXLENGTH](#) 256
- #define [SYMBOL_HASHSIZE](#) 9973
- #define [for_all_properties](#)(sym, st, tok)
- #define [for_all_defaults](#)(sym, st) [for_all_properties](#)(sym, st, [P_DEFAULT](#))
- #define [for_all_choices](#)(sym, st) [for_all_properties](#)(sym, st, [P_CHOICE](#))
- #define [for_all_prompts](#)(sym, st)
- #define [MENU_CHANGED](#) 0x0001
- #define [MENU_ROOT](#) 0x0002
- #define [JUMP_NB](#) 9

Typedefs

- typedef enum [tristate](#) [tristate](#)

Enumerations

- enum `tristate` { `no`, `mod`, `yes` }
- enum `expr_type` {
`E_NONE`, `E_OR`, `E_AND`, `E_NOT`,
`E_EQUAL`, `E_UNEQUAL`, `E_LIST`, `E_SYMBOL`,
`E_RANGE` }
- enum `symbol_type` {
`S_UNKNOWN`, `S_BOOLEAN`, `S_TRISTATE`, `S_INT`,
`S_HEX`, `S_STRING`, `S_OTHER` }
- enum {
`S_DEF_USER`, `S_DEF_AUTO`, `S_DEF_DEF3`, `S_DEF_DEF4`,
`S_DEF_COUNT` }
- enum `prop_type` {
`P_UNKNOWN`, `P_PROMPT`, `P_COMMENT`, `P_MENU`,
`P_DEFAULT`, `P_CHOICE`, `P_SELECT`, `P_RANGE`,
`P_ENV`, `P_SYMBOL` }

Functions

- struct `file` * `lookup_file` (const char *name)
- struct `expr` * `expr_alloc_symbol` (struct `symbol` *sym)
- struct `expr` * `expr_alloc_one` (enum `expr_type` type, struct `expr` *ce)
- struct `expr` * `expr_alloc_two` (enum `expr_type` type, struct `expr` *e1, struct `expr` *e2)
- struct `expr` * `expr_alloc_comp` (enum `expr_type` type, struct `symbol` *s1, struct `symbol` *s2)
- struct `expr` * `expr_alloc_and` (struct `expr` *e1, struct `expr` *e2)
- struct `expr` * `expr_alloc_or` (struct `expr` *e1, struct `expr` *e2)
- struct `expr` * `expr_copy` (const struct `expr` *org)
- void `expr_free` (struct `expr` *e)
- int `expr_eq` (struct `expr` *e1, struct `expr` *e2)
- void `expr_eliminate_eq` (struct `expr` **ep1, struct `expr` **ep2)
- `tristate` `expr_calc_value` (struct `expr` *e)
- struct `expr` * `expr_eliminate_yn` (struct `expr` *e)
- struct `expr` * `expr_trans_bool` (struct `expr` *e)
- struct `expr` * `expr_eliminate_dups` (struct `expr` *e)
- struct `expr` * `expr_transform` (struct `expr` *e)
- int `expr_contains_symbol` (struct `expr` *dep, struct `symbol` *sym)
- bool `expr_depends_symbol` (struct `expr` *dep, struct `symbol` *sym)
- struct `expr` * `expr_extract_eq_and` (struct `expr` **ep1, struct `expr` **ep2)
- struct `expr` * `expr_extract_eq_or` (struct `expr` **ep1, struct `expr` **ep2)
- void `expr_extract_eq` (enum `expr_type` type, struct `expr` **ep, struct `expr` **ep1, struct `expr` **ep2)
- struct `expr` * `expr_trans_compare` (struct `expr` *e, enum `expr_type` type, struct `symbol` *sym)
- struct `expr` * `expr_simplify_unmet_dep` (struct `expr` *e1, struct `expr` *e2)
- void `expr_fprint` (struct `expr` *e, FILE *out)
- void `expr_gstr_print` (struct `expr` *e, struct `gstr` *gs)

Variables

- struct `file` * `file_list`
- struct `file` * `current_file`
- struct `symbol` `symbol_yes` `symbol_no` `symbol_mod`
- struct `symbol` * `modules_sym`
- struct `symbol` * `sym_defconfig_list`
- int `cdebug`

5.20.1 Macro Definition Documentation

5.20.1.1 `#define EXPR_AND(dep1, dep2) (((dep1)<(dep2))?(dep1):(dep2))`

5.20.1.2 `#define expr_list_for_each_sym(l, e, s) for (e = (l); e && (s = e->right.sym); e = e->left.expr)`

5.20.1.3 `#define EXPR_NOT(dep) (2-(dep))`

5.20.1.4 `#define EXPR_OR(dep1, dep2) (((dep1)>(dep2))?(dep1):(dep2))`

5.20.1.5 `#define for_all_choices(sym, st) for_all_properties(sym, st, P_CHOICE)`

5.20.1.6 `#define for_all_defaults(sym, st) for_all_properties(sym, st, P_DEFAULT)`

5.20.1.7 `#define for_all_prompts(sym, st)`

Value:

```
for (st = sym->prop; st; st = st->next) \
    if (st->text)
```

5.20.1.8 `#define for_all_properties(sym, st, tok)`

Value:

```
for (st = sym->prop; st; st = st->next) \
    if (st->type == (tok))
```

5.20.1.9 `#define for_all_symbols(i, sym) for (i = 0; i < SYMBOL_HASHSIZE; i++) for (sym = symbol_hash[i]; sym; sym = sym->next) if (sym->type != S_OTHER)`

5.20.1.10 `#define JUMP_NB 9`

5.20.1.11 `#define MENU_CHANGED 0x0001`

5.20.1.12 `#define MENU_ROOT 0x0002`

5.20.1.13 `#define SYMBOL_AUTO 0x1000 /* value from environment variable */`

5.20.1.14 `#define SYMBOL_CHANGED 0x0400 /* ? */`

5.20.1.15 `#define SYMBOL_CHECK 0x0008 /* used during dependency checking */`

5.20.1.16 `#define SYMBOL_CHECKED 0x2000 /* used during dependency checking */`

5.20.1.17 `#define SYMBOL_CHOICE 0x0010 /* start of a choice block (null name) */`

5.20.1.18 `#define SYMBOL_CHOICEVAL 0x0020 /* used as a value in a choice block */`

5.20.1.19 `#define SYMBOL_CONST 0x0001 /* symbol is const */`

5.20.1.20 `#define SYMBOL_DEF 0x10000 /* First bit of SYMBOL_DEF */`

5.20.1.21 `#define SYMBOL_DEF3 0x40000 /* symbol.def[S_DEF_3] is valid */`

- 5.20.1.22 `#define SYMBOL_DEF4 0x80000 /* symbol.def[S_DEF_4] is valid */`
- 5.20.1.23 `#define SYMBOL_DEF_AUTO 0x20000 /* symbol.def[S_DEF_AUTO] is valid */`
- 5.20.1.24 `#define SYMBOL_DEF_USER 0x10000 /* symbol.def[S_DEF_USER] is valid */`
- 5.20.1.25 `#define SYMBOL_HASHSIZE 9973`
- 5.20.1.26 `#define SYMBOL_MAXLENGTH 256`
- 5.20.1.27 `#define SYMBOL_OPTIONAL 0x0100 /* choice is optional - values can be 'n' */`
- 5.20.1.28 `#define SYMBOL_VALID 0x0080 /* set when symbol.curr is calculated */`
- 5.20.1.29 `#define SYMBOL_WARNED 0x8000 /* warning has been issued */`
- 5.20.1.30 `#define SYMBOL_WRITE 0x0200 /* ? */`

5.20.2 Typedef Documentation

- 5.20.2.1 `typedef enum tristate tristate`

5.20.3 Enumeration Type Documentation

- 5.20.3.1 anonymous enum

Enumerator

S_DEF_USER
S_DEF_AUTO
S_DEF_DEF3
S_DEF_DEF4
S_DEF_COUNT

- 5.20.3.2 `enum expr_type`

Enumerator

E_NONE
E_OR
E_AND
E_NOT
E_EQUAL
E_UNEQUAL
E_LIST
E_SYMBOL
E_RANGE

- 5.20.3.3 `enum prop_type`

Enumerator

P_UNKNOWN

P_PROMPT
P_COMMENT
P_MENU
P_DEFAULT
P_CHOICE
P_SELECT
P_RANGE
P_ENV
P_SYMBOL

5.20.3.4 enum symbol_type

Enumerator

S_UNKNOWN
S_BOOLEAN
S_TRISTATE
S_INT
S_HEX
S_STRING
S_OTHER

5.20.3.5 enum tristate

Enumerator

no
mod
yes

5.20.4 Function Documentation

5.20.4.1 struct expr* expr_alloc_and (struct expr * *e1*, struct expr * *e2*)

5.20.4.2 struct expr* expr_alloc_comp (enum expr_type *type*, struct symbol * *s1*, struct symbol * *s2*)

5.20.4.3 struct expr* expr_alloc_one (enum expr_type *type*, struct expr * *ce*)

5.20.4.4 struct expr* expr_alloc_or (struct expr * *e1*, struct expr * *e2*)

5.20.4.5 struct expr* expr_alloc_symbol (struct symbol * *sym*)

5.20.4.6 struct expr* expr_alloc_two (enum expr_type *type*, struct expr * *e1*, struct expr * *e2*)

5.20.4.7 tristate expr_calc_value (struct expr * *e*)

5.20.4.8 int expr_contains_symbol (struct expr * *dep*, struct symbol * *sym*)

5.20.4.9 struct expr* expr_copy (const struct expr * *org*)

- 5.20.4.10 `bool expr_depends_symbol (struct expr * dep, struct symbol * sym)`
- 5.20.4.11 `struct expr* expr_eliminate_dups (struct expr * e)`
- 5.20.4.12 `void expr_eliminate_eq (struct expr ** ep1, struct expr ** ep2)`
- 5.20.4.13 `struct expr* expr_eliminate_yn (struct expr * e)`
- 5.20.4.14 `int expr_eq (struct expr * e1, struct expr * e2)`
- 5.20.4.15 `void expr_extract_eq (enum expr_type type, struct expr ** ep, struct expr ** ep1, struct expr ** ep2)`
- 5.20.4.16 `struct expr* expr_extract_eq_and (struct expr ** ep1, struct expr ** ep2)`
- 5.20.4.17 `struct expr* expr_extract_eq_or (struct expr ** ep1, struct expr ** ep2)`
- 5.20.4.18 `void expr_fprint (struct expr * e, FILE * out)`
- 5.20.4.19 `void expr_free (struct expr * e)`
- 5.20.4.20 `void expr_gstr_print (struct expr * e, struct gstr * gs)`
- 5.20.4.21 `struct expr* expr_simplify_unmet_dep (struct expr * e1, struct expr * e2)`
- 5.20.4.22 `struct expr* expr_trans_bool (struct expr * e)`
- 5.20.4.23 `struct expr* expr_trans_compare (struct expr * e, enum expr_type type, struct symbol * sym)`
- 5.20.4.24 `struct expr* expr_transform (struct expr * e)`
- 5.20.4.25 `struct file* lookup_file (const char * name)`

5.20.5 Variable Documentation

- 5.20.5.1 `int cdebug`
- 5.20.5.2 `struct file* current_file`
- 5.20.5.3 `struct file* file_list`
- 5.20.5.4 `struct symbol* modules_sym`
- 5.20.5.5 `struct symbol* sym_defconfig_list`
- 5.20.5.6 `struct symbol symbol_yes symbol_no symbol_mod`

5.21 menuconfig/list.h File Reference

Classes

- struct [list_head](#)

Macros

- `#define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)`

- `#define container_of(ptr, type, member)`
- `#define LIST_HEAD_INIT(name) { &(name), &(name) }`
- `#define LIST_HEAD(name) struct list_head name = LIST_HEAD_INIT(name)`
- `#define list_entry(ptr, type, member) container_of(ptr, type, member)`
- `#define list_for_each_entry(pos, head, member)`

5.21.1 Macro Definition Documentation

5.21.1.1 `#define container_of(ptr, type, member)`

Value:

```
((
const typeof( ((type *)0)->member ) *__mptr = (ptr); \
(type *) ( (char *)__mptr - offsetof(type,member) );))
```

`container_of` - cast a member of a structure out to the containing structure : the pointer to the member. : the type of the container struct this is embedded in. : the name of the member within the struct.

5.21.1.2 `#define list_entry(ptr, type, member) container_of(ptr, type, member)`

`list_entry` - get the struct for this entry : the `&struct list_head` pointer. : the type of the struct this is embedded in. : the name of the list_struct within the struct.

5.21.1.3 `#define list_for_each_entry(pos, head, member)`

Value:

```
for (pos = list_entry((head)->next, typeof(*pos), member); \
    &pos->member != (head); \
    pos = list_entry(pos->member.next, typeof(*pos), member))
```

`list_for_each_entry` - iterate over list of given type : the type * to use as a loop cursor. : the head for your list. : the name of the list_struct within the struct.

5.21.1.4 `#define LIST_HEAD(name) struct list_head name = LIST_HEAD_INIT(name)`

5.21.1.5 `#define LIST_HEAD_INIT(name) { &(name), &(name) }`

5.21.1.6 `#define offsetof(TYPE, MEMBER) ((size_t) &((TYPE *)0)->MEMBER)`

5.22 menuconfig/lkc.h File Reference

```
#include "expr.h"
#include <libintl.h>
#include "lkc_proto.h"
```

Classes

- struct `kconf_id`
- struct `conf_printer`
- struct `gstr`

Macros

- `#define P(name, type, arg)` extern type name arg
- `#define SRCTREE "src tree"`
- `#define PACKAGE "linux"`
- `#define LOCALEDIR "/usr/share/locale"`
- `#define _(text)` gettext(text)
- `#define N_(text)` (text)
- `#define CONFIG_ "CONFIG_"`
- `#define TF_COMMAND 0x0001`
- `#define TF_PARAM 0x0002`
- `#define TF_OPTION 0x0004`
- `#define T_OPT_MODULES 1`
- `#define T_OPT_DEFCONFIG_LIST 2`
- `#define T_OPT_ENV 3`

Enumerations

- enum `conf_def_mode` {
`def_default`, `def_yes`, `def_mod`, `def_no`,
`def_random` }

Functions

- int `zconfparse` (void)
- void `zconfdump` (FILE *out)
- void `zconf_starthelp` (void)
- FILE * `zconf_fopen` (const char *name)
- void `zconf_initscan` (const char *name)
- void `zconf_nextfile` (const char *name)
- int `zconf_lineno` (void)
- const char * `zconf_curname` (void)
- const char * `conf_get_configname` (void)
- const char * `conf_get_autoconfig_name` (void)
- char * `conf_get_default_confname` (void)
- void `sym_set_change_count` (int count)
- void `sym_add_change_count` (int count)
- void `conf_set_all_new_symbols` (enum `conf_def_mode` mode)
- void `_menu_init` (void)
- void `menu_warn` (struct `menu` *menu, const char *fmt,...)
- struct `menu` * `menu_add_menu` (void)
- void `menu_end_menu` (void)
- void `menu_add_entry` (struct `symbol` *sym)
- void `menu_end_entry` (void)
- void `menu_add_dep` (struct `expr` *dep)
- void `menu_add_visibility` (struct `expr` *dep)
- struct `property` * `menu_add_prop` (enum `prop_type` type, char *prompt, struct `expr` *expr, struct `expr` *dep)
- struct `property` * `menu_add_prompt` (enum `prop_type` type, char *prompt, struct `expr` *dep)
- void `menu_add_expr` (enum `prop_type` type, struct `expr` *expr, struct `expr` *dep)
- void `menu_add_symbol` (enum `prop_type` type, struct `symbol` *sym, struct `expr` *dep)
- void `menu_add_option` (int token, char *arg)
- void `menu_finalize` (struct `menu` *parent)
- void `menu_set_type` (int type)
- struct `file` * `file_lookup` (const char *name)

- int `file_write_dep` (const char *name)
- struct `gstr str_new` (void)
- struct `gstr str_assign` (const char *s)
- void `str_free` (struct `gstr` *gs)
- void `str_append` (struct `gstr` *gs, const char *s)
- void `str_printf` (struct `gstr` *gs, const char *fmt,...)
- const char * `str_get` (struct `gstr` *gs)
- void `sym_init` (void)
- void `sym_clear_all_valid` (void)
- void `sym_set_all_changed` (void)
- void `sym_set_changed` (struct `symbol` *sym)
- struct `symbol` * `sym_choice_default` (struct `symbol` *sym)
- const char * `sym_get_string_default` (struct `symbol` *sym)
- struct `symbol` * `sym_check_deps` (struct `symbol` *sym)
- struct `property` * `prop_alloc` (enum `prop_type` type, struct `symbol` *sym)
- struct `symbol` * `prop_get_symbol` (struct `property` *prop)
- struct `property` * `sym_get_env_prop` (struct `symbol` *sym)

Variables

- int `zconfdebug`
- struct `expr` * `sym_env_list`

5.22.1 Macro Definition Documentation

5.22.1.1 `#define _(text) gettext(text)`

5.22.1.2 `#define CONFIG_ "CONFIG_ "`

5.22.1.3 `#define LOCALEDIR "/usr/share/locale"`

5.22.1.4 `#define N_(text) (text)`

5.22.1.5 `#define P(name, type, arg)` extern type name arg

5.22.1.6 `#define PACKAGE "linux"`

5.22.1.7 `#define SRCTREE "src tree"`

5.22.1.8 `#define T_OPT_DEFCONFIG_LIST 2`

5.22.1.9 `#define T_OPT_ENV 3`

5.22.1.10 `#define T_OPT_MODULES 1`

5.22.1.11 `#define TF_COMMAND 0x0001`

5.22.1.12 `#define TF_OPTION 0x0004`

5.22.1.13 `#define TF_PARAM 0x0002`

5.22.2 Enumeration Type Documentation

5.22.2.1 enum conf_def_mode

Enumerator

def_default
def_yes
def_mod
def_no
def_random

5.22.3 Function Documentation

5.22.3.1 void _menu_init (void)

5.22.3.2 const char* conf_get_autoconfig_name (void)

5.22.3.3 const char* conf_get_configname (void)

5.22.3.4 char* conf_get_default_confname (void)

5.22.3.5 void conf_set_all_new_symbols (enum conf_def_mode *mode*)5.22.3.6 struct file* file_lookup (const char * *name*)5.22.3.7 int file_write_dep (const char * *name*)5.22.3.8 void menu_add_dep (struct expr * *dep*)5.22.3.9 void menu_add_entry (struct symbol * *sym*)5.22.3.10 void menu_add_expr (enum prop_type *type*, struct expr * *expr*, struct expr * *dep*)

5.22.3.11 struct menu* menu_add_menu (void)

5.22.3.12 void menu_add_option (int *token*, char * *arg*)5.22.3.13 struct property* menu_add_prompt (enum prop_type *type*, char * *prompt*, struct expr * *dep*)5.22.3.14 struct property* menu_add_prop (enum prop_type *type*, char * *prompt*, struct expr * *expr*, struct expr * *dep*)5.22.3.15 void menu_add_symbol (enum prop_type *type*, struct symbol * *sym*, struct expr * *dep*)5.22.3.16 void menu_add_visibility (struct expr * *dep*)

5.22.3.17 void menu_end_entry (void)

5.22.3.18 void menu_end_menu (void)

5.22.3.19 void menu_finalize (struct menu * *parent*)5.22.3.20 void menu_set_type (int *type*)5.22.3.21 void menu_warn (struct menu * *menu*, const char * *fmt*, ...)5.22.3.22 struct property* prop_alloc (enum prop_type *type*, struct symbol * *sym*)

5.22.3.23 struct symbol* prop_get_symbol (struct property * *prop*)

5.22.3.24 void str_append (struct gstr * *gs*, const char * *s*)

5.22.3.25 struct gstr str_assign (const char * *s*)

5.22.3.26 void str_free (struct gstr * *gs*)

5.22.3.27 const char* str_get (struct gstr * *gs*)

5.22.3.28 struct gstr str_new (void)

5.22.3.29 void str_printf (struct gstr * *gs*, const char * *fmt*, ...)

5.22.3.30 void sym_add_change_count (int *count*)

5.22.3.31 struct symbol* sym_check_deps (struct symbol * *sym*)

5.22.3.32 struct symbol* sym_choice_default (struct symbol * *sym*)

5.22.3.33 void sym_clear_all_valid (void)

5.22.3.34 struct property* sym_get_env_prop (struct symbol * *sym*)

5.22.3.35 const char* sym_get_string_default (struct symbol * *sym*)

5.22.3.36 void sym_init (void)

5.22.3.37 void sym_set_all_changed (void)

5.22.3.38 void sym_set_change_count (int *count*)

5.22.3.39 void sym_set_changed (struct symbol * *sym*)

5.22.3.40 const char* zconf_curname (void)

5.22.3.41 FILE* zconf_fopen (const char * *name*)

5.22.3.42 void zconf_initscan (const char * *name*)

5.22.3.43 int zconf_lineno (void)

5.22.3.44 void zconf_nextfile (const char * *name*)

5.22.3.45 void zconf_starthelp (void)

5.22.3.46 void zconfdump (FILE * *out*)

5.22.3.47 int zconfparse (void)

5.22.4 Variable Documentation

5.22.4.1 struct expr* sym_env_list

5.22.4.2 int zconfdebug

5.23 menuconfig/lkc_proto.h File Reference

```
#include <stdarg.h>
```

Functions

- [P \(conf_parse, void,\(const char *name\)\)](#)
- [P \(conf_read, int,\(const char *name\)\)](#)
- [P \(conf_read_simple, int,\(const char *name, int\)\)](#)
- [P \(conf_write_defconfig, int,\(const char *name\)\)](#)
- [P \(conf_write, int,\(const char *name\)\)](#)
- [P \(conf_write_autoconf, int,\(const char *filename\)\)](#)
- [P \(conf_get_changed, bool,\(void\)\)](#)
- [P \(conf_set_changed_callback, void,\(void\(*fn\)\(void\)\)\)](#)
- [P \(conf_set_message_callback, void,\(void\(*fn\)\(const char *fmt, va_list ap\)\)\)](#)
- [P \(rootmenu, struct menu,\)](#)
- [P \(menu_is_visible, bool,\(struct menu *menu\)\)](#)
- [P \(menu_has_prompt, bool,\(struct menu *menu\)\)](#)
- [P \(menu_get_prompt, const char *,\(struct menu *menu\)\)](#)
- [P \(menu_get_root_menu, struct menu *,\(struct menu *menu\)\)](#)
- [P \(menu_get_parent_menu, struct menu *,\(struct menu *menu\)\)](#)
- [P \(menu_has_help, bool,\(struct menu *menu\)\)](#)
- [P \(menu_get_help, const char *,\(struct menu *menu\)\)](#)
- [P \(get_symbol_str, void,\(struct gstr *r, struct symbol *sym, struct list_head *head\)\)](#)
- [P \(get_relations_str, struct gstr,\(struct symbol **sym_arr, struct list_head *head\)\)](#)
- [P \(menu_get_ext_help, void,\(struct menu *menu, struct gstr *help\)\)](#)
- [P \(symbol_hash, struct symbol *,\[SYMBOL_HASHSIZE\]\)](#)
- [P \(sym_lookup, struct symbol *,\(const char *name, int flags\)\)](#)
- [P \(sym_find, struct symbol *,\(const char *name\)\)](#)
- [P \(sym_expand_string_value, const char *,\(const char *in\)\)](#)
- [P \(sym_escape_string_value, const char *,\(const char *in\)\)](#)
- [P \(sym_re_search, struct symbol **, \(const char *pattern\)\)](#)
- [P \(sym_type_name, const char *,\(enum symbol_type type\)\)](#)
- [P \(sym_calc_value, void,\(struct symbol *sym\)\)](#)
- [P \(sym_get_type, enum symbol_type,\(struct symbol *sym\)\)](#)
- [P \(sym_tristate_within_range, bool,\(struct symbol *sym, tristate tri\)\)](#)
- [P \(sym_set_tristate_value, bool,\(struct symbol *sym, tristate tri\)\)](#)
- [P \(sym_toggle_tristate_value, tristate,\(struct symbol *sym\)\)](#)
- [P \(sym_string_valid, bool,\(struct symbol *sym, const char *newval\)\)](#)
- [P \(sym_string_within_range, bool,\(struct symbol *sym, const char *str\)\)](#)
- [P \(sym_set_string_value, bool,\(struct symbol *sym, const char *newval\)\)](#)
- [P \(sym_is_changable, bool,\(struct symbol *sym\)\)](#)
- [P \(sym_get_choice_prop, struct property *,\(struct symbol *sym\)\)](#)
- [P \(sym_get_default_prop, struct property *,\(struct symbol *sym\)\)](#)
- [P \(sym_get_string_value, const char *,\(struct symbol *sym\)\)](#)
- [P \(prop_get_type_name, const char *,\(enum prop_type type\)\)](#)
- [P \(expr_compare_type, int,\(enum expr_type t1, enum expr_type t2\)\)](#)
- [P \(expr_print, void,\(struct expr *e, void\(*fn\)\(void *, struct symbol *, const char *\), void *data, int prevtoken\)\)](#)

5.23.1 Function Documentation

5.23.1.1 P (conf_parse , void , (const char *name))

5.23.1.2 P (conf_read , int , (const char *name))

5.23.1.3 P (conf_read_simple , int , (const char *name, int))

5.23.1.4 P (conf_write_defconfig , int , (const char *name))

5.23.1.5 P (conf_write , int , (const char *name))

5.23.1.6 P (conf_write_autoconf , int , (const char *filename))

5.23.1.7 P (conf_get_changed , bool , (void))

5.23.1.8 P (conf_set_changed_callback , void , (void(*fn)(void)))

5.23.1.9 P (conf_set_message_callback , void , (void(*fn)(const char *fmt, va_list ap)))

5.23.1.10 P (rootmenu , struct menu)

5.23.1.11 P (menu_is_visible , bool , (struct menu *menu))

5.23.1.12 P (menu_has_prompt , bool , (struct menu *menu))

5.23.1.13 P (menu_get_prompt , const char * , (struct menu *menu))

5.23.1.14 P (menu_get_root_menu , struct menu * , (struct menu *menu))

5.23.1.15 P (menu_get_parent_menu , struct menu * , (struct menu *menu))

5.23.1.16 P (menu_has_help , bool , (struct menu *menu))

5.23.1.17 P (menu_get_help , const char * , (struct menu *menu))

5.23.1.18 P (get_symbol_str , void , (struct gstr *r, struct symbol *sym, struct list_head *head))

5.23.1.19 P (get_relations_str , struct gstr , (struct symbol **sym_arr, struct list_head *head))

5.23.1.20 P (menu_get_ext_help , void , (struct menu *menu, struct gstr *help))

5.23.1.21 P (symbol_hash , struct symbol *)

5.23.1.22 P (sym_lookup , struct symbol * , (const char *name, int flags))

5.23.1.23 P (sym_find , struct symbol * , (const char *name))

5.23.1.24 P (sym_expand_string_value , const char * , (const char *in))

5.23.1.25 P (sym_escape_string_value , const char * , (const char *in))

5.23.1.26 P (sym_re_search , struct symbol ** , (const char *pattern))

5.23.1.27 P (sym_type_name , const char * , (enum symbol_type type))

- 5.23.1.28 P (sym_calc_value , void , (struct symbol *sym))
- 5.23.1.29 P (sym_get_type , enum *symbol_type*, (struct symbol *sym))
- 5.23.1.30 P (sym_tristate_within_range , bool , (struct symbol *sym, tristate tri))
- 5.23.1.31 P (sym_set_tristate_value , bool , (struct symbol *sym, tristate tri))
- 5.23.1.32 P (sym_toggle_tristate_value , tristate , (struct symbol *sym))
- 5.23.1.33 P (sym_string_valid , bool , (struct symbol *sym, const char *newval))
- 5.23.1.34 P (sym_string_within_range , bool , (struct symbol *sym, const char *str))
- 5.23.1.35 P (sym_set_string_value , bool , (struct symbol *sym, const char *newval))
- 5.23.1.36 P (sym_is_changable , bool , (struct symbol *sym))
- 5.23.1.37 P (sym_get_choice_prop , struct property * , (struct symbol *sym))
- 5.23.1.38 P (sym_get_default_prop , struct property * , (struct symbol *sym))
- 5.23.1.39 P (sym_get_string_value , const char * , (struct symbol *sym))
- 5.23.1.40 P (prop_get_type_name , const char * , (enum prop_type type))
- 5.23.1.41 P (expr_compare_type , int , (enum expr_type t1, enum expr_type t2))
- 5.23.1.42 P (expr_print , void , (struct expr *e, void(*fn)(void *, struct symbol *, const char *), void *data, int prevtoken))

5.24 menuconfig/lxdialog/checklist.c File Reference

```
#include "dialog.h"
```

Functions

- int [dialog_checklist](#) (const char *title, const char *prompt, int height, int width, int list_height)

5.24.1 Function Documentation

- 5.24.1.1 int [dialog_checklist](#) (const char * *title*, const char * *prompt*, int *height*, int *width*, int *list_height*)

5.25 menuconfig/lxdialog/dialog.h File Reference

```
#include <sys/types.h>
#include <fcntl.h>
#include <unistd.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <libintl.h>
#include <ncurses.h>
```

Classes

- struct [dialog_color](#)
- struct [dialog_info](#)
- struct [dialog_item](#)
- struct [dialog_list](#)

Macros

- #define [TR](#)(params) _tracef params
- #define [KEY_ESC](#) 27
- #define [TAB](#) 9
- #define [MAX_LEN](#) 2048
- #define [BUF_SIZE](#) (10*1024)
- #define [MIN](#)(x, y) (x < y ? x : y)
- #define [MAX](#)(x, y) (x > y ? x : y)
- #define [ACS_ULCORNER](#) '+'
- #define [ACS_LLCORNER](#) '+'
- #define [ACS_URCORNER](#) '+'
- #define [ACS_LRCORNER](#) '+'
- #define [ACS_HLINE](#) '-'
- #define [ACS_VLINE](#) '|'
- #define [ACS_LTEE](#) '+'
- #define [ACS_RTEE](#) '+'
- #define [ACS_UARROW](#) '^'
- #define [ACS_DARROW](#) 'v'
- #define [ERRDISPLAYTOOSMALL](#) (KEY_MAX + 1)
- #define [MAXITEMSTR](#) 200
- #define [item_foreach](#)()
- #define [M_EVENT](#) (KEY_MAX+1)

Typedefs

- typedef void(* [update_text_fn](#))(char *buf, size_t start, size_t end, void *_data)

Functions

- void [item_reset](#) (void)
- void [item_make](#) (const char *fmt,...)
- void [item_add_str](#) (const char *fmt,...)
- void [item_set_tag](#) (char tag)
- void [item_set_data](#) (void *p)
- void [item_set_selected](#) (int val)
- int [item_activate_selected](#) (void)
- void * [item_data](#) (void)
- char [item_tag](#) (void)
- int [item_count](#) (void)
- void [item_set](#) (int n)
- int [item_n](#) (void)
- const char * [item_str](#) (void)
- int [item_is_selected](#) (void)
- int [item_is_tag](#) (char tag)
- int [on_key_esc](#) (WINDOW *win)
- int [on_key_resize](#) (void)
- int [init_dialog](#) (const char *backtitle)
- void [set_dialog_backtitle](#) (const char *backtitle)
- void [end_dialog](#) (int x, int y)
- void [attr_clear](#) (WINDOW *win, int height, int width, chtype attr)
- void [dialog_clear](#) (void)
- void [print_autowrap](#) (WINDOW *win, const char *prompt, int width, int y, int x)
- void [print_button](#) (WINDOW *win, const char *label, int y, int x, int selected)
- void [print_title](#) (WINDOW *dialog, const char *title, int width)
- void [draw_box](#) (WINDOW *win, int y, int x, int height, int width, chtype box, chtype border)
- void [draw_shadow](#) (WINDOW *win, int y, int x, int height, int width)
- int [first_alpha](#) (const char *string, const char *exempt)
- int [dialog_yesno](#) (const char *title, const char *prompt, int height, int width)
- int [dialog_msgbox](#) (const char *title, const char *prompt, int height, int width, int pause)
- int [dialog_textbox](#) (const char *title, char *tbuf, int initial_height, int initial_width, int *keys, int *_vscroll, int *_hscroll, [update_text_fn](#) update_text, void *data)
- int [dialog_menu](#) (const char *title, const char *prompt, const void *selected, int *_s_scroll)
- int [dialog_checklist](#) (const char *title, const char *prompt, int height, int width, int list_height)
- int [dialog_inputbox](#) (const char *title, const char *prompt, int height, int width, const char *init)

Variables

- struct [dialog_info](#) dlg
- char [dialog_input_result](#) []
- int [saved_x](#)
- int [saved_y](#)
- struct [dialog_list](#) * [item_cur](#)
- struct [dialog_list](#) [item_nil](#)
- struct [dialog_list](#) * [item_head](#)

5.25.1 Macro Definition Documentation

5.25.1.1 `#define ACS_DARROW 'v'`

5.25.1.2 `#define ACS_HLINE '-'`

5.25.1.3 `#define ACS_LLCORNER '+'`

5.25.1.4 `#define ACS_LRCORNER '+'`

5.25.1.5 `#define ACS_LTEE '+'`

5.25.1.6 `#define ACS_RTEE '+'`

5.25.1.7 `#define ACS_UARROW '^'`

5.25.1.8 `#define ACS_ULCORNER '+'`

5.25.1.9 `#define ACS_URCORNER '+'`

5.25.1.10 `#define ACS_VLINE '|'`

5.25.1.11 `#define BUF_SIZE (10*1024)`

5.25.1.12 `#define ERRDISPLAYTOOSMALL (KEY_MAX + 1)`

5.25.1.13 `#define item_foreach()`

Value:

```
for (item_cur = item_head ? item_head: item_cur;
     item_cur && (item_cur != &item_nil);
     item_cur = item_cur->next)
```

5.25.1.14 `#define KEY_ESC 27`

5.25.1.15 `#define M_EVENT (KEY_MAX+1)`

5.25.1.16 `#define MAX(x, y) (x > y ? x : y)`

5.25.1.17 `#define MAX_LEN 2048`

5.25.1.18 `#define MAXITEMSTR 200`

5.25.1.19 `#define MIN(x, y) (x < y ? x : y)`

5.25.1.20 `#define TAB 9`

5.25.1.21 `#define TR(params) _tracef params`

5.25.2 Typedef Documentation

5.25.2.1 `typedef void(* update_text_fn)(char *buf, size_t start, size_t end, void *_data)`

5.25.3 Function Documentation

- 5.25.3.1 void attr_clear (WINDOW * win, int height, int width, chtype attr)
- 5.25.3.2 int dialog_checklist (const char * title, const char * prompt, int height, int width, int list_height)
- 5.25.3.3 void dialog_clear (void)
- 5.25.3.4 int dialog_inputbox (const char * title, const char * prompt, int height, int width, const char * init)
- 5.25.3.5 int dialog_menu (const char * title, const char * prompt, const void * selected, int * s_scroll)
- 5.25.3.6 int dialog_msgbox (const char * title, const char * prompt, int height, int width, int pause)
- 5.25.3.7 int dialog_textbox (const char * title, char * tbuf, int initial_height, int initial_width, int * keys, int * _vscroll, int * _hscroll, update_text_fn update_text, void * data)
- 5.25.3.8 int dialog_yesno (const char * title, const char * prompt, int height, int width)
- 5.25.3.9 void draw_box (WINDOW * win, int y, int x, int height, int width, chtype box, chtype border)
- 5.25.3.10 void draw_shadow (WINDOW * win, int y, int x, int height, int width)
- 5.25.3.11 void end_dialog (int x, int y)
- 5.25.3.12 int first_alpha (const char * string, const char * exempt)
- 5.25.3.13 int init_dialog (const char * backtitle)
- 5.25.3.14 int item_activate_selected (void)
- 5.25.3.15 void item_add_str (const char * fmt, ...)
- 5.25.3.16 int item_count (void)
- 5.25.3.17 void* item_data (void)
- 5.25.3.18 int item_is_selected (void)
- 5.25.3.19 int item_is_tag (char tag)
- 5.25.3.20 void item_make (const char * fmt, ...)
- 5.25.3.21 int item_n (void)
- 5.25.3.22 void item_reset (void)
- 5.25.3.23 void item_set (int n)
- 5.25.3.24 void item_set_data (void * p)
- 5.25.3.25 void item_set_selected (int val)
- 5.25.3.26 void item_set_tag (char tag)
- 5.25.3.27 const char* item_str (void)
- 5.25.3.28 char item_tag (void)

5.25.3.29 int on_key_esc (WINDOW * win)

5.25.3.30 int on_key_resize (void)

5.25.3.31 void print_autowrap (WINDOW * win, const char * prompt, int width, int y, int x)

5.25.3.32 void print_button (WINDOW * win, const char * label, int y, int x, int selected)

5.25.3.33 void print_title (WINDOW * dialog, const char * title, int width)

5.25.3.34 void set_dialog_backtitle (const char * backtitle)

5.25.4 Variable Documentation

5.25.4.1 char dialog_input_result

5.25.4.2 struct dialog_info dlg

5.25.4.3 struct dialog_list* item_cur

5.25.4.4 struct dialog_list* item_head

5.25.4.5 struct dialog_list item_nil

5.25.4.6 int saved_x

5.25.4.7 int saved_y

5.26 menuconfig/lxdialog/inputbox.c File Reference

```
#include "dialog.h"
```

Functions

- int [dialog_inputbox](#) (const char *title, const char *prompt, int height, int width, const char *init)

Variables

- char [dialog_input_result](#) [MAX_LEN+1]

5.26.1 Function Documentation

5.26.1.1 int dialog_inputbox (const char * title, const char * prompt, int height, int width, const char * init)

5.26.2 Variable Documentation

5.26.2.1 char dialog_input_result[MAX_LEN+1]

5.27 menuconfig/lxdialog/menubox.c File Reference

```
#include "dialog.h"
```


Macros

- #define [print_item](#)(index, choice, selected)

Functions

- int [dialog_menu](#) (const char *title, const char *prompt, const void *selected, int *s_scroll)

5.27.1 Macro Definition Documentation

5.27.1.1 #define print_item(index, choice, selected)

Value:

```
do {
    item_set(index);
    do_print_item(menu, item_str(), choice, selected, !item_is_tag(':')); \
} while (0)
```

5.27.2 Function Documentation

5.27.2.1 int dialog_menu (const char * title, const char * prompt, const void * selected, int * s_scroll)

5.28 menuconfig/lxdialog/textbox.c File Reference

```
#include "dialog.h"
```

Functions

- int [dialog_textbox](#) (const char *title, char *tbuf, int initial_height, int initial_width, int *keys, int *_vscroll, int *_hscroll, [update_text_fn](#) update_text, void *data)

5.28.1 Function Documentation

5.28.1.1 int dialog_textbox (const char * title, char * tbuf, int initial_height, int initial_width, int * keys, int * _vscroll, int * _hscroll, update_text_fn update_text, void * data)

5.29 menuconfig/lxdialog/util.c File Reference

```
#include <stdarg.h>
#include "dialog.h"
```

Macros

- #define [DLG_COLOR](#)(dialog, f, b, h)

Functions

- void [attr_clear](#) (WINDOW *win, int height, int width, chtype attr)
- void [dialog_clear](#) (void)
- int [init_dialog](#) (const char *backtitle)
- void [set_dialog_backtitle](#) (const char *backtitle)
- void [end_dialog](#) (int x, int y)
- void [print_title](#) (WINDOW *dialog, const char *title, int width)
- void [print_autowrap](#) (WINDOW *win, const char *prompt, int width, int y, int x)
- void [print_button](#) (WINDOW *win, const char *label, int y, int x, int selected)
- void [draw_box](#) (WINDOW *win, int y, int x, int height, int width, chtype box, chtype border)
- void [draw_shadow](#) (WINDOW *win, int y, int x, int height, int width)
- int [first_alpha](#) (const char *string, const char *exempt)
- int [on_key_esc](#) (WINDOW *win)
- int [on_key_resize](#) (void)
- void [item_reset](#) (void)
- void [item_make](#) (const char *fmt,...)
- void [item_add_str](#) (const char *fmt,...)
- void [item_set_tag](#) (char tag)
- void [item_set_data](#) (void *ptr)
- void [item_set_selected](#) (int val)
- int [item_activate_selected](#) (void)
- void * [item_data](#) (void)
- char [item_tag](#) (void)
- int [item_count](#) (void)
- void [item_set](#) (int n)
- int [item_n](#) (void)
- const char * [item_str](#) (void)
- int [item_is_selected](#) (void)
- int [item_is_tag](#) (char tag)

Variables

- int [saved_x](#)
- int [saved_y](#)
- struct [dialog_info](#) dlg
- struct [dialog_list](#) * [item_cur](#)
- struct [dialog_list](#) [item_nil](#)
- struct [dialog_list](#) * [item_head](#)

5.29.1 Macro Definition Documentation

5.29.1.1 `#define DLG_COLOR(dialog, f, b, h)`

Value:

```
do {
    dlg.dialog.fg = (f);
    dlg.dialog.bg = (b);
    dlg.dialog.hl = (h);
} while (0)
```

5.29.2 Function Documentation

- 5.29.2.1 void attr_clear (WINDOW * *win*, int *height*, int *width*, chtype *attr*)
- 5.29.2.2 void dialog_clear (void)
- 5.29.2.3 void draw_box (WINDOW * *win*, int *y*, int *x*, int *height*, int *width*, chtype *box*, chtype *border*)
- 5.29.2.4 void draw_shadow (WINDOW * *win*, int *y*, int *x*, int *height*, int *width*)
- 5.29.2.5 void end_dialog (int *x*, int *y*)
- 5.29.2.6 int first_alpha (const char * *string*, const char * *exempt*)
- 5.29.2.7 int init_dialog (const char * *backtitle*)
- 5.29.2.8 int item_activate_selected (void)
- 5.29.2.9 void item_add_str (const char * *fmt*, ...)
- 5.29.2.10 int item_count (void)
- 5.29.2.11 void* item_data (void)
- 5.29.2.12 int item_is_selected (void)
- 5.29.2.13 int item_is_tag (char *tag*)
- 5.29.2.14 void item_make (const char * *fmt*, ...)
- 5.29.2.15 int item_n (void)
- 5.29.2.16 void item_reset (void)
- 5.29.2.17 void item_set (int *n*)
- 5.29.2.18 void item_set_data (void * *ptr*)
- 5.29.2.19 void item_set_selected (int *val*)
- 5.29.2.20 void item_set_tag (char *tag*)
- 5.29.2.21 const char* item_str (void)
- 5.29.2.22 char item_tag (void)
- 5.29.2.23 int on_key_esc (WINDOW * *win*)
- 5.29.2.24 int on_key_resize (void)
- 5.29.2.25 void print_autowrap (WINDOW * *win*, const char * *prompt*, int *width*, int *y*, int *x*)
- 5.29.2.26 void print_button (WINDOW * *win*, const char * *label*, int *y*, int *x*, int *selected*)
- 5.29.2.27 void print_title (WINDOW * *dialog*, const char * *title*, int *width*)

5.29.2.28 void set_dialog_backtitle (const char * *backtitle*)

5.29.3 Variable Documentation

5.29.3.1 struct dialog_info dlg

5.29.3.2 struct dialog_list* item_cur

5.29.3.3 struct dialog_list* item_head

5.29.3.4 struct dialog_list item_nil

5.29.3.5 int saved_x

5.29.3.6 int saved_y

5.30 menuconfig/util.c File Reference

```
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include "lkc.h"
```

Functions

- struct file * file_lookup (const char *name)
- int file_write_dep (const char *name)
- struct gstr str_new (void)
- struct gstr str_assign (const char *s)
- void str_free (struct gstr *gs)
- void str_append (struct gstr *gs, const char *s)
- void str_printf (struct gstr *gs, const char *fmt,...)
- const char * str_get (struct gstr *gs)

5.30.1 Function Documentation

5.30.1.1 struct file* file_lookup (const char * *name*)

5.30.1.2 int file_write_dep (const char * *name*)

5.30.1.3 void str_append (struct gstr * *gs*, const char * *s*)

5.30.1.4 struct gstr str_assign (const char * *s*)

5.30.1.5 void str_free (struct gstr * *gs*)

5.30.1.6 const char* str_get (struct gstr * *gs*)

5.30.1.7 struct gstr str_new (void)

5.30.1.8 void str_printf (struct gstr * *gs*, const char * *fmt*, ...)

5.31 menuconfig/lxdialog/yesno.c File Reference

```
#include "dialog.h"
```

Functions

- int [dialog_yesno](#) (const char *title, const char *prompt, int height, int width)

5.31.1 Function Documentation

5.31.1.1 int [dialog_yesno](#) (const char * *title*, const char * *prompt*, int *height*, int *width*)

5.32 menuconfig/mconf.c File Reference

```
#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <limits.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <locale.h>
#include "lkc.h"
#include "lxdialog/dialog.h"
```

Classes

- struct [search_data](#)

Functions

- int [main](#) (int ac, char **av)

5.32.1 Function Documentation

5.32.1.1 int [main](#) (int *ac*, char ** *av*)

5.33 menuconfig/menu.c File Reference

```
#include <ctype.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include "lkc.h"
```

Functions

- void `menu_warn` (struct `menu` *`menu`, const char *`fmt`,...)
- void `_menu_init` (void)
- void `menu_add_entry` (struct `symbol` *`sym`)
- void `menu_end_entry` (void)
- struct `menu` * `menu_add_menu` (void)
- void `menu_end_menu` (void)
- void `menu_add_dep` (struct `expr` *`dep`)
- void `menu_set_type` (int `type`)
- struct `property` * `menu_add_prop` (enum `prop_type` `type`, char *`prompt`, struct `expr` *`expr`, struct `expr` *`dep`)
- struct `property` * `menu_add_prompt` (enum `prop_type` `type`, char *`prompt`, struct `expr` *`dep`)
- void `menu_add_visibility` (struct `expr` *`expr`)
- void `menu_add_expr` (enum `prop_type` `type`, struct `expr` *`expr`, struct `expr` *`dep`)
- void `menu_add_symbol` (enum `prop_type` `type`, struct `symbol` *`sym`, struct `expr` *`dep`)
- void `menu_add_option` (int `token`, char *`arg`)
- void `menu_finalize` (struct `menu` *`parent`)
- bool `menu_has_prompt` (struct `menu` *`menu`)
- bool `menu_is_visible` (struct `menu` *`menu`)
- const char * `menu_get_prompt` (struct `menu` *`menu`)
- struct `menu` * `menu_get_root_menu` (struct `menu` *`menu`)
- struct `menu` * `menu_get_parent_menu` (struct `menu` *`menu`)
- bool `menu_has_help` (struct `menu` *`menu`)
- const char * `menu_get_help` (struct `menu` *`menu`)
- void `get_symbol_str` (struct `gstr` *`r`, struct `symbol` *`sym`, struct `list_head` *`head`)
- struct `gstr` `get_relations_str` (struct `symbol` **`sym_arr`, struct `list_head` *`head`)
- void `menu_get_ext_help` (struct `menu` *`menu`, struct `gstr` *`help`)

Variables

- struct `menu` `rootmenu`
- struct `file` * `file_list`
- struct `file` * `current_file`

5.33.1 Function Documentation

5.33.1.1 void `_menu_init` (void)

5.33.1.2 struct `gstr` `get_relations_str` (struct `symbol` ** `sym_arr`, struct `list_head` * `head`)

5.33.1.3 void `get_symbol_str` (struct `gstr` * `r`, struct `symbol` * `sym`, struct `list_head` * `head`)

5.33.1.4 void `menu_add_dep` (struct `expr` * `dep`)

5.33.1.5 void `menu_add_entry` (struct `symbol` * `sym`)

5.33.1.6 void `menu_add_expr` (enum `prop_type` `type`, struct `expr` * `expr`, struct `expr` * `dep`)

5.33.1.7 struct `menu`* `menu_add_menu` (void)

5.33.1.8 void `menu_add_option` (int `token`, char * `arg`)

5.33.1.9 struct `property`* `menu_add_prompt` (enum `prop_type` `type`, char * `prompt`, struct `expr` * `dep`)

- 5.33.1.10 `struct property* menu_add_prop (enum prop_type type, char * prompt, struct expr * expr, struct expr * dep)`
- 5.33.1.11 `void menu_add_symbol (enum prop_type type, struct symbol * sym, struct expr * dep)`
- 5.33.1.12 `void menu_add_visibility (struct expr * expr)`
- 5.33.1.13 `void menu_end_entry (void)`
- 5.33.1.14 `void menu_end_menu (void)`
- 5.33.1.15 `void menu_finalize (struct menu * parent)`
- 5.33.1.16 `void menu_get_ext_help (struct menu * menu, struct gstr * help)`
- 5.33.1.17 `const char* menu_get_help (struct menu * menu)`
- 5.33.1.18 `struct menu* menu_get_parent_menu (struct menu * menu)`
- 5.33.1.19 `const char* menu_get_prompt (struct menu * menu)`
- 5.33.1.20 `struct menu* menu_get_root_menu (struct menu * menu)`
- 5.33.1.21 `bool menu_has_help (struct menu * menu)`
- 5.33.1.22 `bool menu_has_prompt (struct menu * menu)`
- 5.33.1.23 `bool menu_is_visible (struct menu * menu)`
- 5.33.1.24 `void menu_set_type (int type)`
- 5.33.1.25 `void menu_warn (struct menu * menu, const char * fmt, ...)`

5.33.2 Variable Documentation

- 5.33.2.1 `struct file* current_file`
- 5.33.2.2 `struct file* file_list`
- 5.33.2.3 `struct menu rootmenu`

5.34 menuconfig/README.md File Reference

5.35 menuconfig/symbol.c File Reference

```
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <regex.h>
#include <sys/utsname.h>
#include "lkc.h"
```

Classes

- struct **dep_stack**

Functions

- void **sym_init** (void)
- enum **symbol_type** **sym_get_type** (struct **symbol** *sym)
- const char * **sym_type_name** (enum **symbol_type** type)
- struct **property** * **sym_get_choice_prop** (struct **symbol** *sym)
- struct **property** * **sym_get_env_prop** (struct **symbol** *sym)
- struct **property** * **sym_get_default_prop** (struct **symbol** *sym)
- struct **symbol** * **sym_choice_default** (struct **symbol** *sym)
- void **sym_calc_value** (struct **symbol** *sym)
- void **sym_clear_all_valid** (void)
- void **sym_set_changed** (struct **symbol** *sym)
- void **sym_set_all_changed** (void)
- bool **sym_tristate_within_range** (struct **symbol** *sym, **tristate** val)
- bool **sym_set_tristate_value** (struct **symbol** *sym, **tristate** val)
- **tristate** **sym_toggle_tristate_value** (struct **symbol** *sym)
- bool **sym_string_valid** (struct **symbol** *sym, const char *str)
- bool **sym_string_within_range** (struct **symbol** *sym, const char *str)
- bool **sym_set_string_value** (struct **symbol** *sym, const char *newval)
- const char * **sym_get_string_default** (struct **symbol** *sym)
- const char * **sym_get_string_value** (struct **symbol** *sym)
- bool **sym_is_changable** (struct **symbol** *sym)
- struct **symbol** * **sym_lookup** (const char *name, int flags)
- struct **symbol** * **sym_find** (const char *name)
- const char * **sym_expand_string_value** (const char *in)
- const char * **sym_escape_string_value** (const char *in)
- struct **symbol** ** **sym_re_search** (const char *pattern)
- struct **symbol** * **sym_check_deps** (struct **symbol** *sym)
- struct **property** * **prop_alloc** (enum **prop_type** type, struct **symbol** *sym)
- struct **symbol** * **prop_get_symbol** (struct **property** *prop)
- const char * **prop_get_type_name** (enum **prop_type** type)

Variables

- struct **symbol** **symbol_yes**
- struct **symbol** **symbol_mod**
- struct **symbol** **symbol_no**
- struct **symbol** **symbol_empty**
- struct **symbol** * **sym_defconfig_list**
- struct **symbol** * **modules_sym**
- **tristate** **modules_val**
- struct **expr** * **sym_env_list**

5.35.1 Function Documentation

5.35.1.1 `struct property* prop_alloc (enum prop_type type, struct symbol * sym)`

5.35.1.2 `struct symbol* prop_get_symbol (struct property * prop)`

5.35.1.3 `const char* prop_get_type_name (enum prop_type type)`

5.35.1.4 `void sym_calc_value (struct symbol * sym)`

5.35.1.5 `struct symbol* sym_check_deps (struct symbol * sym)`

5.35.1.6 `struct symbol* sym_choice_default (struct symbol * sym)`

5.35.1.7 `void sym_clear_all_valid (void)`

5.35.1.8 `const char* sym_escape_string_value (const char * in)`

5.35.1.9 `const char* sym_expand_string_value (const char * in)`

5.35.1.10 `struct symbol* sym_find (const char * name)`

5.35.1.11 `struct property* sym_get_choice_prop (struct symbol * sym)`

5.35.1.12 `struct property* sym_get_default_prop (struct symbol * sym)`

5.35.1.13 `struct property* sym_get_env_prop (struct symbol * sym)`

5.35.1.14 `const char* sym_get_string_default (struct symbol * sym)`

5.35.1.15 `const char* sym_get_string_value (struct symbol * sym)`

5.35.1.16 `enum symbol_type sym_get_type (struct symbol * sym)`

5.35.1.17 `void sym_init (void)`

5.35.1.18 `bool sym_is_changable (struct symbol * sym)`

5.35.1.19 `struct symbol* sym_lookup (const char * name, int flags)`

5.35.1.20 `struct symbol** sym_re_search (const char * pattern)`

5.35.1.21 `void sym_set_all_changed (void)`

5.35.1.22 `void sym_set_changed (struct symbol * sym)`

5.35.1.23 `bool sym_set_string_value (struct symbol * sym, const char * newval)`

5.35.1.24 `bool sym_set_tristate_value (struct symbol * sym, tristate val)`

5.35.1.25 `bool sym_string_valid (struct symbol * sym, const char * str)`

5.35.1.26 `bool sym_string_within_range (struct symbol * sym, const char * str)`

5.35.1.27 `tristate sym_toggle_tristate_value (struct symbol * sym)`

5.35.1.28 `bool sym_tristate_within_range (struct symbol * sym, tristate val)`

5.35.1.29 `const char* sym_type_name (enum symbol_type type)`

5.35.2 Variable Documentation

5.35.2.1 `struct symbol* modules_sym`

5.35.2.2 `tristate modules_val`

5.35.2.3 `struct symbol* sym_defconfig_list`

5.35.2.4 `struct expr* sym_env_list`

5.35.2.5 `struct symbol symbol_empty`

Initial value:

```
= {
    .name = "",
    .curr = { "", no },
    .flags = SYMBOL_VALID,
}
```

5.35.2.6 `struct symbol symbol_mod`

Initial value:

```
= {
    .name = "m",
    .curr = { "m", mod },
    .flags = SYMBOL_CONST|SYMBOL_VALID,
}
```

5.35.2.7 `struct symbol symbol_no`

Initial value:

```
= {
    .name = "n",
    .curr = { "n", no },
    .flags = SYMBOL_CONST|SYMBOL_VALID,
}
```

5.35.2.8 `struct symbol symbol_yes`

Initial value:

```
= {
    .name = "y",
    .curr = { "y", yes },
    .flags = SYMBOL_CONST|SYMBOL_VALID,
}
```

5.36 menuconfig/zconf.hash.c File Reference

Classes

- struct `kconf_id_strings_t`

Macros

- `#define kconf_id_strings ((const char *) &kconf_id_strings_contents)`

5.36.1 Macro Definition Documentation

5.36.1.1 `#define kconf_id_strings ((const char *) &kconf_id_strings_contents)`

5.37 menuconfig/zconf.lex.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <limits.h>
#include <unistd.h>
#include "lkc.h"
```

Classes

- struct `yy_buffer_state`
- struct `yy_trans_info`
- struct `buffer`

Macros

- `#define YY_INT_ALIGNED` short int
- `#define yy_create_buffer zconf_create_buffer`
- `#define yy_delete_buffer zconf_delete_buffer`
- `#define yy_flex_debug zconf_flex_debug`
- `#define yy_init_buffer zconf_init_buffer`
- `#define yy_flush_buffer zconf_flush_buffer`
- `#define yy_load_buffer_state zconf_load_buffer_state`
- `#define yy_switch_to_buffer zconf_switch_to_buffer`
- `#define yyin zconfin`
- `#define yyleng zconfleng`
- `#define yylex zconflex`
- `#define yylineno zconflineno`
- `#define yyout zconfout`
- `#define yyrestart zconfrestart`
- `#define yytext zconfcontext`
- `#define yywrap zconfwrap`
- `#define yyallocc zconfalloc`
- `#define yyrealloc zconfrealloc`
- `#define yyfree zconffree`
- `#define FLEX_SCANNER`
- `#define YY_FLEX_MAJOR_VERSION` 2
- `#define YY_FLEX_MINOR_VERSION` 5
- `#define YY_FLEX_SUBMINOR_VERSION` 35
- `#define FLEX_BETA`
- `#define FLEXINT_H`
- `#define INT8_MIN` (-128)

- #define INT16_MIN (-32767-1)
- #define INT32_MIN (-2147483647-1)
- #define INT8_MAX (127)
- #define INT16_MAX (32767)
- #define INT32_MAX (2147483647)
- #define UINT8_MAX (255U)
- #define UINT16_MAX (65535U)
- #define UINT32_MAX (4294967295U)
- #define yyconst
- #define YY_NULL 0
- #define YY_SC_TO_UI(c) ((unsigned int) (unsigned char) c)
- #define BEGIN (yy_start) = 1 + 2 *
- #define YY_START (((yy_start) - 1) / 2)
- #define YYSTATE YY_START
- #define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)
- #define YY_NEW_FILE zconfrestart(zconfin)
- #define YY_END_OF_BUFFER_CHAR 0
- #define YY_BUF_SIZE 16384
- #define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))
- #define YY_TYPEDEF_YY_BUFFER_STATE
- #define EOB_ACT_CONTINUE_SCAN 0
- #define EOB_ACT_END_OF_FILE 1
- #define EOB_ACT_LAST_MATCH 2
- #define YY_LESS_LINENO(n)
- #define yylless(n)
- #define unput(c) yyunput(c, (yytext_ptr))
- #define YY_TYPEDEF_YY_SIZE_T
- #define YY_STRUCT_YY_BUFFER_STATE
- #define YY_BUFFER_NEW 0
- #define YY_BUFFER_NORMAL 1
- #define YY_BUFFER_EOF_PENDING 2
- #define YY_CURRENT_BUFFER
- #define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]
- #define YY_FLUSH_BUFFER zconf_flush_buffer(YY_CURRENT_BUFFER)
- #define yy_new_buffer zconf_create_buffer
- #define yy_set_interactive(is_interactive)
- #define yy_set_bol(at_bol)
- #define YY_AT_BOL() (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)
- #define zconfwrap(n) 1
- #define YY_SKIP_YYWRAP
- #define yytext_ptr zconfcontext
- #define YY_DO_BEFORE_ACTION
- #define YY_NUM_RULES 33
- #define YY_END_OF_BUFFER 34
- #define REJECT reject_used_but_not_detected
- #define yymore() yymore_used_but_not_detected
- #define YY_MORE_ADJ 0
- #define YY_RESTORE_YY_MORE_OFFSET
- #define YY_NO_INPUT 1
- #define START_STRSIZE 16
- #define INITIAL 0
- #define COMMAND 1
- #define HELP 2
- #define STRING 3
- #define PARAM 4

- `#define YY_EXTRA_TYPE` void *
- `#define YY_READ_BUF_SIZE` 8192
- `#define ECHO` do { if (fwrite(zconfcontext, zconfleng, 1, zconfout)) {} } while (0)
- `#define YY_INPUT`(buf, result, max_size)
- `#define yyterminate()` return YY_NULL
- `#define YY_START_STACK_INCR` 25
- `#define YY_FATAL_ERROR`(msg) yy_fatal_error(msg)
- `#define YY_DECL_IS_OURS` 1
- `#define YY_DECL` int zconflex (void)
- `#define YY_USER_ACTION`
- `#define YY_BREAK` break;
- `#define YY_RULE_SETUP` YY_USER_ACTION
- `#define YY_EXIT_FAILURE` 2
- `#define yyless`(n)
- `#define YYTABLES_NAME` "yytables"

Typedefs

- typedef signed char flex_int8_t
- typedef short int flex_int16_t
- typedef int flex_int32_t
- typedef unsigned char flex_uint8_t
- typedef unsigned short int flex_uint16_t
- typedef unsigned int flex_uint32_t
- typedef struct yy_buffer_state * YY_BUFFER_STATE
- typedef size_t yy_size_t
- typedef unsigned char YY_CHAR
- typedef int yy_state_type

Functions

- void zconfrestart (FILE *input_file)
- void zconf_switch_to_buffer (YY_BUFFER_STATE new_buffer)
- YY_BUFFER_STATE zconf_create_buffer (FILE *file, int size)
- void zconf_delete_buffer (YY_BUFFER_STATE b)
- void zconf_flush_buffer (YY_BUFFER_STATE b)
- void zconfpush_buffer_state (YY_BUFFER_STATE new_buffer)
- void zconfpop_buffer_state (void)
- YY_BUFFER_STATE zconf_scan_buffer (char *base, yy_size_t size)
- YY_BUFFER_STATE zconf_scan_string (yyconst char *yy_str)
- YY_BUFFER_STATE zconf_scan_bytes (yyconst char *bytes, int len)
- void * zconfalloc (yy_size_t)
- void * zconfrealloc (void *, yy_size_t)
- void zconffree (void *)
- int zconflex_destroy (void)
- int zconfget_debug (void)
- void zconfset_debug (int debug_flag)
- YY_EXTRA_TYPE zconfget_extra (void)
- void zconfset_extra (YY_EXTRA_TYPE user_defined)
- FILE * zconfget_in (void)
- void zconfset_in (FILE *in_str)
- FILE * zconfget_out (void)
- void zconfset_out (FILE *out_str)
- int zconfget_leng (void)

- char * [zconfget_text](#) (void)
- int [zconfget_lineno](#) (void)
- void [zconfset_lineno](#) (int line_number)
- int [zconflex](#) (void)
- if (!yy_init))
- while (1)
- void [zconf_starthelp](#) (void)
- FILE * [zconf_fopen](#) (const char *name)
- void [zconf_initscan](#) (const char *name)
- void [zconf_nextfile](#) (const char *name)
- int [zconf_lineno](#) (void)
- const char * [zconf_curname](#) (void)

Variables

- int [zconfleng](#)
- FILE * [zconfin](#) = (FILE *) 0
- FILE * [zconfout](#) = (FILE *) 0
- int [zconflينو](#) = 1
- char * [zconfcontext](#)
- int [zconf_flex_debug](#) = 0
- struct buffer * [current_buf](#)
- YY_DECL
- register char * [yy_cp](#)
- register char * [yy_bp](#)
- register int [yy_act](#)
- int [str](#) = 0
- int [ts](#)
- int [i](#)

5.37.1 Macro Definition Documentation

5.37.1.1 `#define BEGIN (yy_start) = 1 + 2 *`

5.37.1.2 `#define COMMAND 1`

5.37.1.3 `#define ECHO do { if (fwrite(zconfcontext, zconfleng, 1, zconfout)) {} } while (0)`

5.37.1.4 `#define EOB_ACT_CONTINUE_SCAN 0`

5.37.1.5 `#define EOB_ACT_END_OF_FILE 1`

5.37.1.6 `#define EOB_ACT_LAST_MATCH 2`

5.37.1.7 `#define FLEX_BETA`

5.37.1.8 `#define FLEX_SCANNER`

5.37.1.9 `#define FLEXINT_H`

5.37.1.10 `#define HELP 2`

5.37.1.11 `#define INITIAL 0`

```

5.37.1.12 #define INT16_MAX (32767)

5.37.1.13 #define INT16_MIN (-32767-1)

5.37.1.14 #define INT32_MAX (2147483647)

5.37.1.15 #define INT32_MIN (-2147483647-1)

5.37.1.16 #define INT8_MAX (127)

5.37.1.17 #define INT8_MIN (-128)

5.37.1.18 #define PARAM 4

5.37.1.19 #define REJECT reject_used_but_not_detected

5.37.1.20 #define START_STRSIZE 16

5.37.1.21 #define STRING 3

5.37.1.22 #define UINT16_MAX (65535U)

5.37.1.23 #define UINT32_MAX (4294967295U)

5.37.1.24 #define UINT8_MAX (255U)

5.37.1.25 #define unput( c ) yyunput( c, (yytext_ptr) )

5.37.1.26 #define YY_AT_BOL( ) (YY_CURRENT_BUFFER_LVALUE->yy_at_bol)

5.37.1.27 #define YY_BREAK break;

5.37.1.28 #define YY_BUF_SIZE 16384

5.37.1.29 #define YY_BUFFER_EOF_PENDING 2

5.37.1.30 #define YY_BUFFER_NEW 0

5.37.1.31 #define YY_BUFFER_NORMAL 1

5.37.1.32 #define yy_create_buffer zconf_create_buffer

5.37.1.33 #define YY_CURRENT_BUFFER

```

Value:

```

( (yy_buffer_stack) \
    ? (yy_buffer_stack)[(yy_buffer_stack_top)] \
    : NULL)

```

```

5.37.1.34 #define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]

5.37.1.35 #define YY_DECL int zconflex (void)

5.37.1.36 #define YY_DECL_IS_OURS 1

```

5.37.1.37 `#define yy_delete_buffer zconf_delete_buffer`

5.37.1.38 `#define YY_DO_BEFORE_ACTION`

Value:

```
(yytext_ptr) = yy_bp; \
zconfleng = (size_t) (yy_cp - yy_bp); \
(yy_hold_char) = *yy_cp; \
*yy_cp = '\0'; \
(yy_c_buf_p) = yy_cp;
```

5.37.1.39 `#define YY_END_OF_BUFFER 34`

5.37.1.40 `#define YY_END_OF_BUFFER_CHAR 0`

5.37.1.41 `#define YY_EXIT_FAILURE 2`

5.37.1.42 `#define YY_EXTRA_TYPE void *`

5.37.1.43 `#define YY_FATAL_ERROR(msg) yy_fatal_error(msg)`

5.37.1.44 `#define yy_flex_debug zconf_flex_debug`

5.37.1.45 `#define YY_FLEX_MAJOR_VERSION 2`

5.37.1.46 `#define YY_FLEX_MINOR_VERSION 5`

5.37.1.47 `#define YY_FLEX_SUBMINOR_VERSION 35`

5.37.1.48 `#define yy_flush_buffer zconf_flush_buffer`

5.37.1.49 `#define YY_FLUSH_BUFFER zconf_flush_buffer(YY_CURRENT_BUFFER)`

5.37.1.50 `#define yy_init_buffer zconf_init_buffer`

5.37.1.51 `#define YY_INPUT(buf, result, max_size)`

Value:

```
errno=0; \
while ( (result = read( fileno(zconfin), (char *) buf, max_size )) < 0 ) \
{ \
    if( errno != EINTR) \
    { \
        YY_FATAL_ERROR( "input in flex scanner failed" ); \
        break; \
    } \
    errno=0; \
    clearerr(zconfin); \
} \
\
```

5.37.1.52 `#define YY_INT_ALIGNED short int`

5.37.1.53 `#define YY_LESS_LINENO(n)`

5.37.1.54 `#define yy_load_buffer_state zconf_load_buffer_state`

5.37.1.55 `#define YY_MORE_ADJ 0`

5.37.1.56 `#define yy_new_buffer zconf_create_buffer`

5.37.1.57 `#define YY_NEW_FILE zconfrestart(zconfin)`

5.37.1.58 `#define YY_NO_INPUT 1`

5.37.1.59 `#define YY_NULL 0`

5.37.1.60 `#define YY_NUM_RULES 33`

5.37.1.61 `#define YY_READ_BUF_SIZE 8192`

5.37.1.62 `#define YY_RESTORE_YY_MORE_OFFSET`

5.37.1.63 `#define YY_RULE_SETUP YY_USER_ACTION`

5.37.1.64 `#define YY_SC_TO_UI(c) ((unsigned int) (unsigned char) c)`

5.37.1.65 `#define yy_set_bol(at_bol)`

Value:

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        zconfensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            zconf_create_buffer(zconfin, \
                YY_BUF_SIZE ); \
    } \
    YY_CURRENT_BUFFER_LVALUE->yy_at_bol = at_bol; \
}
```

5.37.1.66 `#define yy_set_interactive(is_interactive)`

Value:

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        zconfensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            zconf_create_buffer(zconfin, \
                YY_BUF_SIZE ); \
    } \
    YY_CURRENT_BUFFER_LVALUE->yy_is_interactive = is_interactive; \
}
```

5.37.1.67 `#define YY_SKIP_YYWRAP`

5.37.1.68 `#define YY_START (((yy_start) - 1) / 2)`

5.37.1.69 `#define YY_START_STACK_INCR 25`

5.37.1.70 `#define YY_STATE_BUF_SIZE ((YY_BUF_SIZE + 2) * sizeof(yy_state_type))`

5.37.1.71 `#define YY_STATE_EOF(state) (YY_END_OF_BUFFER + state + 1)`

5.37.1.72 `#define YY_STRUCT_YY_BUFFER_STATE`

5.37.1.73 **#define yy_switch_to_buffer zconf_switch_to_buffer**

5.37.1.74 **#define YY_TYPEDEF_YY_BUFFER_STATE**

5.37.1.75 **#define YY_TYPEDEF_YY_SIZE_T**

5.37.1.76 **#define YY_USER_ACTION**

5.37.1.77 **#define yyalloc zconfalloc**

5.37.1.78 **#define yyconst**

5.37.1.79 **#define yyfree zconffree**

5.37.1.80 **#define yyin zconfin**

5.37.1.81 **#define yyleng zconfleng**

5.37.1.82 **#define yyless(n)**

Value:

```
do \
{ \
/* Undo effects of setting up zconfcontext. */ \
int yyless_macro_arg = (n); \
YY_LESS_LINENO(yyless_macro_arg);\
*yy_cp = (yy_hold_char); \
YY_RESTORE_YY_MORE_OFFSET \
(yy_c_buf_p) = yy_cp = yy_bp + yyless_macro_arg - YY_MORE_ADJ;
\
YY_DO_BEFORE_ACTION; /* set up zconfcontext again */ \
} \
while ( 0 )
```

5.37.1.83 **#define yyless(n)**

Value:

```
do \
{ \
/* Undo effects of setting up zconfcontext. */ \
int yyless_macro_arg = (n); \
YY_LESS_LINENO(yyless_macro_arg);\
zconfcontext[zconfleng] = (yy_hold_char); \
(yy_c_buf_p) = zconfcontext + yyless_macro_arg; \
(yy_hold_char) = *(yy_c_buf_p); \
*(yy_c_buf_p) = '\0'; \
zconfleng = yyless_macro_arg; \
} \
while ( 0 )
```

5.37.1.84 **#define yylex zconflex**

5.37.1.85 **#define yylineno zconflineno**

5.37.1.86 **#define yymore() yymore_used_but_not_detected**

5.37.1.87 **#define yyout zconfout**

5.37.1.88 **#define yyrealloc zconfrealloc**

5.37.1.89 `#define yyrestart zconfrestart`

5.37.1.90 `#define YYSTATE YY_START`

5.37.1.91 `#define YYTABLES_NAME "yytables"`

5.37.1.92 `#define yyterminate() return YY_NULL`

5.37.1.93 `#define yytext zconftext`

5.37.1.94 `#define yytext_ptr zconftext`

5.37.1.95 `#define yywrap zconfwrap`

5.37.1.96 `#define zconfwrap(n) 1`

5.37.2 Typedef Documentation

5.37.2.1 `typedef short int flex_int16_t`

5.37.2.2 `typedef int flex_int32_t`

5.37.2.3 `typedef signed char flex_int8_t`

5.37.2.4 `typedef unsigned short int flex_uint16_t`

5.37.2.5 `typedef unsigned int flex_uint32_t`

5.37.2.6 `typedef unsigned char flex_uint8_t`

5.37.2.7 `typedef struct yy_buffer_state* YY_BUFFER_STATE`

5.37.2.8 `typedef unsigned char YY_CHAR`

5.37.2.9 `typedef size_t yy_size_t`

5.37.2.10 `typedef int yy_state_type`

5.37.3 Function Documentation

5.37.3.1 `if (! yy_init)`

5.37.3.2 `while (1)`

5.37.3.3 `YY_BUFFER_STATE zconf_create_buffer (FILE * file, int size)`

Allocate and initialize an input buffer state.

Parameters

<i>file</i>	A readable stream.
<i>size</i>	The character buffer size in bytes. When in doubt, use YY_BUF_SIZE.

Returns

the allocated buffer state.

5.37.3.4 `const char* zconf_curname (void)`

5.37.3.5 `void zconf_delete_buffer (YY_BUFFER_STATE b)`

Destroy the buffer.

Parameters

<i>b</i>	a buffer created with zconf_create_buffer()
----------	---

5.37.3.6 `void zconf_flush_buffer (YY_BUFFER_STATE b)`

Discard all buffered characters. On the next scan, YY_INPUT will be called.

Parameters

<i>b</i>	the buffer state to be flushed, usually YY_CURRENT_BUFFER.
----------	--

5.37.3.7 `FILE* zconf_fopen (const char * name)`

5.37.3.8 `void zconf_initscan (const char * name)`

5.37.3.9 `int zconf_lineno (void)`

5.37.3.10 `void zconf_nextfile (const char * name)`

5.37.3.11 `YY_BUFFER_STATE zconf_scan_buffer (char * base, yy_size_t size)`

Setup the input buffer state to scan directly from a user-specified character buffer.

Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

Returns

the newly allocated buffer state object.

5.37.3.12 `YY_BUFFER_STATE zconf_scan_bytes (yyconst char * yybytes, int _yybytes_len)`

Setup the input buffer state to scan the given bytes. The next call to [zconflex\(\)](#) will scan from a *copy* of *bytes*.

Parameters

<i>bytes</i>	the byte buffer to scan
<i>len</i>	the number of bytes in the buffer pointed to by <i>bytes</i> .

Returns

the newly allocated buffer state object.

5.37.3.13 `YY_BUFFER_STATE zconf_scan_string (yyconst char * ystr)`

Setup the input buffer state to scan a string. The next call to [zconflex\(\)](#) will scan from a *copy* of *str*.

Parameters

<i>yyst</i>	a NUL-terminated string to scan
-------------	---------------------------------

Returns

the newly allocated buffer state object.

Note

If you want to scan bytes that may contain NUL values, then use [zconf_scan_bytes\(\)](#) instead.

5.37.3.14 void `zconf_starthelp (void)`

5.37.3.15 void `zconf_switch_to_buffer (YY_BUFFER_STATE new_buffer)`

Switch to a different input buffer.

Parameters

<i>new_buffer</i>	The new input buffer.
-------------------	-----------------------

5.37.3.16 void * `zconfalloc (yy_size_t size)`

5.37.3.17 void `zconfree (void * ptr)`

5.37.3.18 int `zconfget_debug (void)`

5.37.3.19 YY_EXTRA_TYPE `zconfget_extra (void)`

5.37.3.20 FILE * `zconfget_in (void)`

Get the input stream.

5.37.3.21 int `zconfget_leng (void)`

Get the length of the current token.

5.37.3.22 int `zconfget_lineno (void)`

Get the current line number.

5.37.3.23 FILE * `zconfget_out (void)`

Get the output stream.

5.37.3.24 char * `zconfget_text (void)`

Get the current token.

5.37.3.25 `int zconflex (void)`

5.37.3.26 `int zconflex_destroy (void)`

5.37.3.27 `void zconfpop_buffer_state (void)`

Removes and deletes the top of the stack, if present. The next element becomes the new top.

5.37.3.28 `void zconfpush_buffer_state (YY_BUFFER_STATE new_buffer)`

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

Parameters

<i>new_buffer</i>	The new state.
-------------------	----------------

5.37.3.29 `void * zconfrealloc (void * ptr, yy_size_t size)`

5.37.3.30 `void zconfrestart (FILE * input_file)`

Immediately switch to a different input stream.

Parameters

<i>input_file</i>	A readable stream.
-------------------	--------------------

Note

This function does not reset the start condition to `INITIAL`.

5.37.3.31 `void zconfset_debug (int debug_flag)`

5.37.3.32 `void zconfset_extra (YY_EXTRA_TYPE user_defined)`

5.37.3.33 `void zconfset_in (FILE * in_str)`

Set the input stream. This does not discard the current input buffer.

Parameters

<i>in_str</i>	A readable stream.
---------------	--------------------

See Also

[zconf_switch_to_buffer](#)

5.37.3.34 `void zconfset_lineno (int line_number)`

Set the current line number.

Parameters

<i>line_number</i>	
--------------------	--

5.37.3.35 void zconfset_out (FILE * *out_str*)

5.37.4 Variable Documentation

5.37.4.1 struct buffer* current_buf

5.37.4.2 struct file* file

5.37.4.3 int i

5.37.4.4 int lineno

5.37.4.5 int str = 0

5.37.4.6 int ts

5.37.4.7 register int yy_act

5.37.4.8 register char * yy_bp

5.37.4.9 register char* yy_cp

5.37.4.10 YY_DECL

Initial value:

```
{  
    register yy_state_type yy_current_state
```

The main scanner function which does all the work.

5.37.4.11 int zconf_flex_debug = 0

5.37.4.12 FILE * zconfin = (FILE *) 0

5.37.4.13 int zconfleng

5.37.4.14 int zconflineno = 1

5.37.4.15 FILE * zconfout = (FILE *) 0

5.37.4.16 char * zconfcontext

5.38 menuconfig/zconf.tab.c File Reference

```
#include <ctype.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <limits.h>
#include <libgen.h>
#include "lkc.h"
#include "zconf.hash.c"
#include "zconf.lex.c"
#include "util.c"
#include "confdata.c"
#include "expr.c"
#include "symbol.c"
#include "menu.c"
```

Classes

- union [YYSTYPE](#)
- union [yyval](#)

Macros

- `#define YYBISON 1`
- `#define YYBISON_VERSION "2.4.3"`
- `#define YYSKELETON_NAME "yacc.c"`
- `#define YYPURE 0`
- `#define YYPUSH 0`
- `#define YYPULL 1`
- `#define YYLSP_NEEDED 0`
- `#define yyparse zconfparse`
- `#define yylex zconflex`
- `#define yyerror zconferror`
- `#define yylval zconfval`
- `#define yychar zconfchar`
- `#define yydebug zconfdebug`
- `#define yynerrs zconfnerrs`
- `#define PATH_MAX 4096`
- `#define printf(mask, fmt...) if (cdebug & (mask)) printf(fmt)`
- `#define PRINTD 0x0001`
- `#define DEBUG_PARSE 0x0002`
- `#define YYDEBUG 1`
- `#define YYERROR_VERBOSE 0`
- `#define YYTOKEN_TABLE 0`
- `#define YYTOKENTYPE`
- `#define YYSTYPE_IS_TRIVIAL 1`
- `#define yystype YYSTYPE /* obsolescent; will be withdrawn */`
- `#define YYSTYPE_IS_DECLARED 1`
- `#define YYSIZE_T unsigned int`
- `#define YYSIZE_MAXIMUM ((YYSIZE_T) -1)`

- #define YY_(msgid) msgid
- #define YYUSE(e) ((void) (e))
- #define YYID(n) (n)
- #define YYSTACK_ALLOC YYMALLOC
- #define YYSTACK_FREE YYFREE
- #define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM
- #define YYMALLOC malloc
- #define YYFREE free
- #define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)
- #define YYSTACK_BYTES(N)
- #define YYCOPY(To, From, Count)
- #define YYSTACK_RELOCATE(Stack_alloc, Stack)
- #define YYFINAL 11
- #define YYLAST 290
- #define YYNTOKENS 36
- #define YYNNTS 50
- #define YYNRULES 118
- #define YYNSTATES 191
- #define YYUNDEFTOK 2
- #define YYMAXUTOK 290
- #define YYTRANSLATE(YYX) ((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)
- #define YYPACT_NINF -90
- #define YYTABLE_NINF -86
- #define yyerrok (yyerrstatus = 0)
- #define yyclearin (yychar = YYEMPTY)
- #define YYEMPTY (-2)
- #define YYEOF 0
- #define YYACCEPT goto yyacceptlab
- #define YYABORT goto yyabortlab
- #define YYERROR goto yyerrorlab
- #define YYFAIL goto yyerrlab
- #define YYRECOVERING() (!yyerrstatus)
- #define YYBACKUP(Token, Value)
- #define YYTERROR 1
- #define YYERRCODE 256
- #define YYRHSLOC(Rhs, K) ((Rhs)[K])
- #define YYLLOC_DEFAULT(Current, Rhs, N)
- #define YY_LOCATION_PRINT(File, Loc) ((void) 0)
- #define YYLEX yylex ()
- #define YYFPRINTF fprintf
- #define YYDPRINTF(Args)
- #define YY_SYMBOL_PRINT(Title, Type, Value, Location)
- #define YY_STACK_PRINT(Bottom, Top)
- #define YY_REDUCE_PRINT(Rule)
- #define YYINITDEPTH 200
- #define YYMAXDEPTH 10000
- #define YYPOPSTACK(N) (yyvsp -= (N), yyssp -= (N))

Typedefs

- typedef union YYSTYPE YYSTYPE
- typedef unsigned char yytype_uint8
- typedef short int yytype_int8
- typedef unsigned short int yytype_uint16
- typedef short int yytype_int16

Enumerations

- enum `yytokentype` {
`T_MAINMENU` = 258, `T_MENU` = 259, `T_ENDMENU` = 260, `T_SOURCE` = 261,
`T_CHOICE` = 262, `T_ENDCHOICE` = 263, `T_COMMENT` = 264, `T_CONFIG` = 265,
`T_MENUCONFIG` = 266, `T_HELP` = 267, `T_HELPTEXT` = 268, `T_IF` = 269,
`T_ENDIF` = 270, `T_DEPENDS` = 271, `T_OPTIONAL` = 272, `T_PROMPT` = 273,
`T_TYPE` = 274, `T_DEFAULT` = 275, `T_SELECT` = 276, `T_RANGE` = 277,
`T_VISIBLE` = 278, `T_OPTION` = 279, `T_ON` = 280, `T_WORD` = 281,
`T_WORD_QUOTE` = 282, `T_UNEQUAL` = 283, `T_CLOSE_PAREN` = 284, `T_OPEN_PAREN` = 285,
`T_EOL` = 286, `T_OR` = 287, `T_AND` = 288, `T_EQUAL` = 289,
`T_NOT` = 290 }

Functions

- int `zconflex` (void)
- int `yyparse` ()
- void `conf_parse` (const char *name)
- void `zconfdump` (FILE *out)

Variables

- int `cdebug` = `PRINTD`
- struct `symbol` * `symbol_hash` [`SYMBOL_HASHSIZE`]
- int `yydebug`
- int `yychar`
- `YYSTYPE` `yyval`
- int `ynerrs`

5.38.1 Macro Definition Documentation

5.38.1.1 `#define DEBUG_PARSE 0x0002`

5.38.1.2 `#define PATH_MAX 4096`

5.38.1.3 `#define printf(mask, fmt...) if (cdebug & (mask)) printf(fmt)`

5.38.1.4 `#define PRINTD 0x0001`

5.38.1.5 `#define YY_(msgid) msgid`

5.38.1.6 `#define YY_LOCATION_PRINT(File, Loc) ((void) 0)`

5.38.1.7 `#define YY_REDUCE_PRINT(Rule)`

Value:

```
do {
    if (yydebug)
        yy_reduce_print (yyvsp, Rule); \
} while (YYID (0))
```

5.38.1.8 `#define YY_STACK_PRINT(Bottom, Top)`**Value:**

```
do {
    if (yydebug)
        yy_stack_print ((Bottom), (Top));
} while (YYID (0))
```

5.38.1.9 `#define YY_SYMBOL_PRINT(Title, Type, Value, Location)`**Value:**

```
do {
    if (yydebug)
    {
        YYFPRINTF (stderr, "%s ", Title);
        yy_symbol_print (stderr,
            Type, Value);
        YYFPRINTF (stderr, "\n");
    }
} while (YYID (0))
```

5.38.1.10 `#define YYABORT goto yyabortlab`5.38.1.11 `#define YYACCEPT goto yyacceptlab`5.38.1.12 `#define YYBACKUP(Token, Value)`**Value:**

```
do
    if (yychar == YYEMPTY && yylen == 1)
    {
        yychar = (Token);
        yylval = (Value);
        yytoken = YYTRANSLATE (yychar);
        YYPOPSTACK (1);
        goto yybackup;
    }
    else
    {
        yyerror (YY_("syntax error: cannot back up"));
        YYERROR;
    }
while (YYID (0))
```

5.38.1.13 `#define YYBISON 1`5.38.1.14 `#define YYBISON_VERSION "2.4.3"`5.38.1.15 `#define yychar zconfchar`5.38.1.16 `#define yyclearin (yychar = YYEMPTY)`5.38.1.17 `#define YYCOPY(To, From, Count)`**Value:**

```
do
{
    YYSIZE_T yyi;
    for (yyi = 0; yyi < (Count); yyi++)
        (To)[yyi] = (From)[yyi];
}
while (YYID (0))
```

5.38.1.18 `#define yydebug zconfdebug`

5.38.1.19 `#define YYDEBUG 1`

5.38.1.20 `#define YYDPRINTF(Args)`

Value:

```
do {
    if (yydebug)
        YYFPRINTF Args;
} while (YYID (0))
```

5.38.1.21 `#define YYEMPTY (-2)`

5.38.1.22 `#define YYEOF 0`

5.38.1.23 `#define YYERRCODE 256`

5.38.1.24 `#define yyerrok (yyerrstatus = 0)`

5.38.1.25 `#define yyerror zconferror`

5.38.1.26 `#define YYERROR goto yyerrorlab`

5.38.1.27 `#define YYERROR_VERBOSE 0`

5.38.1.28 `#define YYFAIL goto yyerrlab`

5.38.1.29 `#define YYFINAL 11`

5.38.1.30 `#define YYFPRINTF fprintf`

5.38.1.31 `#define YYFREE free`

5.38.1.32 `#define YYID(n)(n)`

5.38.1.33 `#define YYINITDEPTH 200`

5.38.1.34 `#define YYLAST 290`

5.38.1.35 `#define yylex zconflex`

5.38.1.36 `#define YYLEX yylex ()`

5.38.1.37 `#define YYLLOC_DEFAULT(Current, Rhs, N)`

Value:

```
do
{
    if (YYID (N))
    {
        (Current).first_line   = YYRHSLOC (Rhs, 1).first_line;
        (Current).first_column = YYRHSLOC (Rhs, 1).first_column;
        (Current).last_line    = YYRHSLOC (Rhs, N).last_line;
        (Current).last_column   = YYRHSLOC (Rhs, N).last_column;
    }
    else
    {
        (Current).first_line   = (Current).last_line
        (Current).first_column = (Current).last_column
```

```

        YYRHSLOC (Rhs, 0).last_line; \
        (Current).first_column = (Current).last_column = \
        YYRHSLOC (Rhs, 0).last_column; \
    } \
    while (YYID (0))

```

5.38.1.38 `#define YYLSP_NEEDED 0`

5.38.1.39 `#define yylval zconfval`

5.38.1.40 `#define YYMALLOC malloc`

5.38.1.41 `#define YYMAXDEPTH 10000`

5.38.1.42 `#define YYMAXUTOK 290`

5.38.1.43 `#define yynerrs zconfnerrs`

5.38.1.44 `#define YYNNTS 50`

5.38.1.45 `#define YYNRULES 118`

5.38.1.46 `#define YYNSTATES 191`

5.38.1.47 `#define YYNTOKENS 36`

5.38.1.48 `#define YYPACT_NINF -90`

5.38.1.49 `int yyparse zconfparse`

5.38.1.50 `#define YYPOPSTACK(N) (yyvsp -= (N), yyssp -= (N))`

5.38.1.51 `#define YYPULL 1`

5.38.1.52 `#define YYPURE 0`

5.38.1.53 `#define YYPUSH 0`

5.38.1.54 `#define YYRECOVERING() (!yyerrstatus)`

5.38.1.55 `#define YYRHSLOC(Rhs, K) ((Rhs)[K])`

5.38.1.56 `#define YYSIZE_MAXIMUM ((YYSIZE_T) -1)`

5.38.1.57 `#define YYSIZE_T unsigned int`

5.38.1.58 `#define YYSKELETON_NAME "yacc.c"`

5.38.1.59 `#define YYSTACK_ALLOC YYMALLOC`

5.38.1.60 `#define YYSTACK_ALLOC_MAXIMUM YYSIZE_MAXIMUM`

5.38.1.61 `#define YYSTACK_BYTES(N)`

Value:

```

((N) * (sizeof (yytype_int16) + sizeof (YYSTYPE)) \
 + YYSTACK_GAP_MAXIMUM)

```

5.38.1.62 `#define YYSTACK_FREE YYFREE`

5.38.1.63 `#define YYSTACK_GAP_MAXIMUM (sizeof (union yyalloc) - 1)`

5.38.1.64 `#define YYSTACK_RELOCATE(Stack_alloc, Stack)`

Value:

```
do
{
    YYSIZE_T yynewbytes;
    YYCOPY (&yyptr->Stack_alloc, Stack, yysize);
    Stack = &yyptr->Stack_alloc;
    yynewbytes = yystacksize * sizeof (*Stack) + YYSTACK_GAP_MAXIMUM;
    yyptr += yynewbytes / sizeof (*yyptr);
}
while (YYID (0))
```

5.38.1.65 `#define yystype YYSTYPE /* obsolescent; will be withdrawn */`

5.38.1.66 `#define YYSTYPE_IS_DECLARED 1`

5.38.1.67 `#define YYSTYPE_IS_TRIVIAL 1`

5.38.1.68 `#define YYTABLE_NINF -86`

5.38.1.69 `#define YYTERROR 1`

5.38.1.70 `#define YYTOKEN_TABLE 0`

5.38.1.71 `#define YYTOKENTYPE`

5.38.1.72 `#define YYTRANSLATE(YYX) ((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)`

5.38.1.73 `#define YYUNDEFTOK 2`

5.38.1.74 `#define YYUSE(e) ((void) (e))`

5.38.2 Typedef Documentation

5.38.2.1 `typedef union YYSTYPE YYSTYPE`

5.38.2.2 `typedef short int yytype_int16`

5.38.2.3 `typedef short int yytype_int8`

5.38.2.4 `typedef unsigned short int yytype_uint16`

5.38.2.5 `typedef unsigned char yytype_uint8`

5.38.3 Enumeration Type Documentation

5.38.3.1 `enum yytokentype`

Enumerator

`T_MAINMENU`

`T_MENU`

T_ENDMENU
T_SOURCE
T_CHOICE
T_ENDCHOICE
T_COMMENT
T_CONFIG
T_MENUCONFIG
T_HELP
T_HELPTEXT
T_IF
T_ENDIF
T_DEPENDS
T_OPTIONAL
T_PROMPT
T_TYPE
T_DEFAULT
T_SELECT
T_RANGE
T_VISIBLE
T_OPTION
T_ON
T_WORD
T_WORD_QUOTE
T_UNEQUAL
T_CLOSE_PAREN
T_OPEN_PAREN
T_EOL
T_OR
T_AND
T_EQUAL
T_NOT

5.38.4 Function Documentation

5.38.4.1 void conf_parse (const char * *name*)

5.38.4.2 int yyparse ()

5.38.4.3 void zconfdump (FILE * *out*)

5.38.4.4 int zconflex (void)

5.38.5 Variable Documentation

5.38.5.1 int cdebug = PRINTD

5.38.5.2 struct symbol* symbol_hash[SYMBOL_HASHSIZE]

5.38.5.3 int yychar

5.38.5.4 int yydebug

5.38.5.5 YYSTYPE yylval

5.38.5.6 int yynerrs

5.39 printf/printf.c File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "printf.h"
#include <float.h>
```

Classes

- struct [out_fct_wrap_type](#)

Macros

- #define [PRINTF_NTOA_BUFFER_SIZE](#) 32U
- #define [PRINTF_FTOA_BUFFER_SIZE](#) 32U
- #define [PRINTF_SUPPORT_FLOAT](#)
- #define [PRINTF_SUPPORT_EXPONENTIAL](#)
- #define [PRINTF_DEFAULT_FLOAT_PRECISION](#) 6U
- #define [PRINTF_MAX_FLOAT](#) 1e9
- #define [PRINTF_SUPPORT_LONG_LONG](#)
- #define [PRINTF_SUPPORT_PTRDIFF_T](#)
- #define [FLAGS_ZEROPAD](#) (1U << 0U)
- #define [FLAGS_LEFT](#) (1U << 1U)
- #define [FLAGS_PLUS](#) (1U << 2U)
- #define [FLAGS_SPACE](#) (1U << 3U)
- #define [FLAGS_HASH](#) (1U << 4U)
- #define [FLAGS_UPPERCASE](#) (1U << 5U)
- #define [FLAGS_CHAR](#) (1U << 6U)
- #define [FLAGS_SHORT](#) (1U << 7U)
- #define [FLAGS_LONG](#) (1U << 8U)
- #define [FLAGS_LONG_LONG](#) (1U << 9U)
- #define [FLAGS_PRECISION](#) (1U << 10U)
- #define [FLAGS_ADAPT_EXP](#) (1U << 11U)

Typedefs

- typedef void(* [out_fct_type](#))(char character, void *buffer, size_t idx, size_t maxlen)

Functions

- int `printf_` (const char *format,...)
- int `sprintf_` (char *buffer, const char *format,...)
- int `snprintf_` (char *buffer, size_t count, const char *format,...)
- int `vprintf_` (const char *format, va_list va)
- int `vsprintf_` (char *buffer, size_t count, const char *format, va_list va)
- int `fctprintf` (void(*out)(char character, void *arg), void *arg, const char *format,...)

5.39.1 Macro Definition Documentation

5.39.1.1 `#define FLAGS_ADAPT_EXP (1U << 11U)`

5.39.1.2 `#define FLAGS_CHAR (1U << 6U)`

5.39.1.3 `#define FLAGS_HASH (1U << 4U)`

5.39.1.4 `#define FLAGS_LEFT (1U << 1U)`

5.39.1.5 `#define FLAGS_LONG (1U << 8U)`

5.39.1.6 `#define FLAGS_LONG_LONG (1U << 9U)`

5.39.1.7 `#define FLAGS_PLUS (1U << 2U)`

5.39.1.8 `#define FLAGS_PRECISION (1U << 10U)`

5.39.1.9 `#define FLAGS_SHORT (1U << 7U)`

5.39.1.10 `#define FLAGS_SPACE (1U << 3U)`

5.39.1.11 `#define FLAGS_UPPERCASE (1U << 5U)`

5.39.1.12 `#define FLAGS_ZEROPAD (1U << 0U)`

5.39.1.13 `#define PRINTF_DEFAULT_FLOAT_PRECISION 6U`

5.39.1.14 `#define PRINTF_FTOA_BUFFER_SIZE 32U`

5.39.1.15 `#define PRINTF_MAX_FLOAT 1e9`

5.39.1.16 `#define PRINTF_NTOA_BUFFER_SIZE 32U`

5.39.1.17 `#define PRINTF_SUPPORT_EXPONENTIAL`

5.39.1.18 `#define PRINTF_SUPPORT_FLOAT`

5.39.1.19 `#define PRINTF_SUPPORT_LONG_LONG`

5.39.1.20 `#define PRINTF_SUPPORT_PTRDIFF_T`

5.39.2 Typedef Documentation

5.39.2.1 `typedef void(* out_fct_type)(char character, void *buffer, size_t idx, size_t maxlen)`

5.39.3 Function Documentation

5.39.3.1 `int fctprintf (void(*) (char character, void *arg) out, void * arg, const char * format, ...)`

printf with output function You may use this as dynamic alternative to `printf()` with its fixed `_putchar()` output

Parameters

<code>out</code>	An output function which takes one character and an argument pointer
<code>arg</code>	An argument pointer for user data passed to output function
<code>format</code>	A string that specifies the format of the output

Returns

The number of characters that are sent to the output function, not counting the terminating null character

5.39.3.2 `int printf_ (const char * format, ...)`

5.39.3.3 `int snprintf_ (char * buffer, size_t count, const char * format, ...)`

5.39.3.4 `int sprintf_ (char * buffer, const char * format, ...)`

5.39.3.5 `int vprintf_ (const char * format, va_list va)`

5.39.3.6 `int vsnprintf_ (char * buffer, size_t count, const char * format, va_list va)`

5.40 printf/printf.h File Reference

```
#include <stdarg.h>
#include <stddef.h>
```

Macros

- `#define printf printf_`
- `#define sprintf sprintf_`
- `#define snprintf snprintf_`
- `#define vsnprintf vsnprintf_`
- `#define vprintf vprintf_`

Functions

- `void _putchar (char character)`
This function is used by the `printf()` port. It sends a single char over the UART interface. This function is blocking.
- `int printf_ (const char *format,...)`
- `int sprintf_ (char *buffer, const char *format,...)`
- `int snprintf_ (char *buffer, size_t count, const char *format,...)`
- `int vsnprintf_ (char *buffer, size_t count, const char *format, va_list va)`
- `int vprintf_ (const char *format, va_list va)`
- `int fctprintf (void(*) (char character, void *arg), void *arg, const char *format,...)`

5.40.1 Macro Definition Documentation

5.40.1.1 `#define printf printf_`

Tiny printf implementation You have to implement *putchar* if you use [printf\(\)](#) To avoid conflicts with the regular [printf\(\)](#) API it is overridden by macro defines and internal underscore-appended functions like *printf_* are used

Parameters

<i>format</i>	A string that specifies the format of the output
---------------	--

Returns

The number of characters that are written into the array, not counting the terminating null character

5.40.1.2 #define snprintf snprintf_

Tiny snprintf/vsnprintf implementation

Parameters

<i>buffer</i>	A pointer to the buffer where to store the formatted string
<i>count</i>	The maximum number of characters to store in the buffer, including a terminating null character
<i>format</i>	A string that specifies the format of the output
<i>va</i>	A value identifying a variable arguments list

Returns

The number of characters that COULD have been written into the buffer, not counting the terminating null character. A value equal or larger than count indicates truncation. Only when the returned value is non-negative and less than count, the string has been completely written.

5.40.1.3 #define sprintf sprintf_

Tiny sprintf implementation Due to security reasons (buffer overflow) YOU SHOULD CONSIDER USING (V)SNPRINTF INSTEAD!

Parameters

<i>buffer</i>	A pointer to the buffer where to store the formatted string. MUST be big enough to store the output!
<i>format</i>	A string that specifies the format of the output

Returns

The number of characters that are WRITTEN into the buffer, not counting the terminating null character

5.40.1.4 #define vprintf vprintf_

Tiny vprintf implementation

Parameters

<i>format</i>	A string that specifies the format of the output
<i>va</i>	A value identifying a variable arguments list

Returns

The number of characters that are WRITTEN into the buffer, not counting the terminating null character

5.40.1.5 `#define vsnprintf vsnprintf_`

5.40.2 Function Documentation

5.40.2.1 `void _putchar (char character)`

This function is used by the `printf()` port. It sends a single char over the UART interface. This function is blocking.

Output a character to a custom device like UART, used by the `printf()` function. This function is declared here only. You have to write your custom implementation somewhere.

Parameters

<i>character</i>	Character to output
<i>character</i>	- an 8-bit ASCII character.

Returns

None.

5.40.2.2 `int fctprintf (void(*) (char character, void *arg) out, void *arg, const char *format, ...)`

`printf` with output function. You may use this as dynamic alternative to `printf()` with its fixed `_putchar()` output.

Parameters

<i>out</i>	An output function which takes one character and an argument pointer
<i>arg</i>	An argument pointer for user data passed to output function
<i>format</i>	A string that specifies the format of the output

Returns

The number of characters that are sent to the output function, not counting the terminating null character.

5.40.2.3 `int printf_ (const char *format, ...)`5.40.2.4 `int snprintf_ (char *buffer, size_t count, const char *format, ...)`5.40.2.5 `int sprintf_ (char *buffer, const char *format, ...)`5.40.2.6 `int vprintf_ (const char *format, va_list va)`5.40.2.7 `int vsnprintf_ (char *buffer, size_t count, const char *format, va_list va)`

5.41 startup.c File Reference

```
#include "startup.h"
```

Functions

- void `main` (void)

The main entry of the application. This function initializes a GPIO and a UART, then outputs a message over the UART and starts blinking the diode.

- `__attribute__` ((section(".vector_table")))

- void [nmi_handler](#) (void)
- void [hard_fault_handler](#) (void)
- void [mem_manage_handler](#) (void)
- void [bus_fault_handler](#) (void)
- void [usage_fault_handler](#) (void)
- void [svc_handler](#) (void)
- void [debug_mon_handler](#) (void)
- void [pend_sv_handler](#) (void)
- void [systick_handler](#) (void)
- void [default_handler](#) (void)
- void [reset_handler](#) (void)

Variables

- uint32_t [_estack](#)
- uint32_t [_sdata_lma](#)
- uint32_t [_sdata](#)
- uint32_t [_edata](#)
- uint32_t [_sbss](#)
- uint32_t [_ebss](#)

5.41.1 Function Documentation

5.41.1.1 `__attribute__((section(".vector_table")))`

5.41.1.2 `void bus_fault_handler (void)`

5.41.1.3 `void debug_mon_handler (void)`

5.41.1.4 `void default_handler (void)`

5.41.1.5 `void hard_fault_handler (void)`

5.41.1.6 `void main (void)`

The main entry of the application. This function initializes a GPIO and a UART, then outputs a message over the UART and starts blinking the diode.

Returns

None.

5.41.1.7 `void mem_manage_handler (void)`

5.41.1.8 `void nmi_handler (void)`

5.41.1.9 `void pend_sv_handler (void)`

5.41.1.10 `void reset_handler (void)`

5.41.1.11 `void svc_handler (void)`

5.41.1.12 `void systick_handler (void)`

5.41.1.13 void usage_fault_handler (void)

5.41.2 Variable Documentation

5.41.2.1 uint32_t _ebss

5.41.2.2 uint32_t _edata

5.41.2.3 uint32_t _estack

5.41.2.4 uint32_t _sbss

5.41.2.5 uint32_t _sdata

5.41.2.6 uint32_t _sdata_lma

5.42 startup.h File Reference

```
#include <stdint.h>
#include "main.h"
```

Functions

- void [reset_handler](#) (void)
- void [nmi_handler](#) (void)
- void [hard_fault_handler](#) (void)
- void [mem_manage_handler](#) (void)
- void [bus_fault_handler](#) (void)
- void [usage_fault_handler](#) (void)
- void [svc_handler](#) (void)
- void [debug_mon_handler](#) (void)
- void [pend_sv_handler](#) (void)
- void [systick_handler](#) (void)
- void [default_handler](#) (void)

5.42.1 Function Documentation

5.42.1.1 void bus_fault_handler (void)

5.42.1.2 void debug_mon_handler (void)

5.42.1.3 void default_handler (void)

5.42.1.4 void hard_fault_handler (void)

5.42.1.5 void mem_manage_handler (void)

5.42.1.6 void nmi_handler (void)

5.42.1.7 void pend_sv_handler (void)

5.42.1.8 void reset_handler (void)

5.42.1.9 void `svc_handler (void)`

5.42.1.10 void `systick_handler (void)`

5.42.1.11 void `usage_fault_handler (void)`

5.43 tmp/led/blink/iterator/200000.h File Reference

5.44 tmp/led/blink/iterator/400000.h File Reference

5.45 tmp/led/blink/iterator/50000.h File Reference

5.46 tmp/uart/ baud/ rate.h File Reference

5.47 uart/uart.c File Reference

```
#include "uart.h"
```

Functions

- void `uart_init (uint32_t baud_rate)`
This function initializes the UART module. It can be used for redirecting the printf function.
- void `uart_send_buff (uint8_t *buff, uint8_t buff_size)`
This function sends an array of chars over the UART interface.
- void `_putchar (char character)`
This function is used by the `printf()` port. It sends a single char over the UART interface. This function is blocking.

5.47.1 Function Documentation

5.47.1.1 void `_putchar (char character)`

This function is used by the `printf()` port. It sends a single char over the UART interface. This function is blocking.

Parameters

<i>character</i>	- an 8-bit ASCII character.
------------------	-----------------------------

Returns

None.

5.47.1.2 void `uart_init (uint32_t baud_rate)`

This function initializes the UART module. It can be used for redirecting the printf function.

To initialize the UART, first the clock of the module has to be enabled. Second, the GPIO pins have to be multiplexed to the UART module. Third, the module has to be enabled and last - the UART baudrate has to be set according to the user settings.

Parameters

<i>baud_rate</i>	- the speed of the data transmission over the interface, in bauds. Can be one of the following integers: , , , .
------------------	--

Returns

None.

5.47.1.3 void uart_send_buff (uint8_t * buff, uint8_t buff_size)

This function sends an array of chars over the UART interface.

Parameters

<i>buff</i>	- a pointer to an array of 8-bit characters.
<i>buff_size</i>	- the number of elements in the array.

Returns

None.

5.48 uart/uart.h File Reference

```
#include <stdint.h>
```

Macros

- #define [GPIOA_MODER_AF_PA9](#) 0x80000
- #define [GPIOA_OTYPER_PA9](#) 0x200
- #define [GPIOA_MODER_AF_PA10](#) 0x200000
- #define [GPIOA_AF7_PA9](#) 0x70
- #define [GPIOA_AF7_PA10](#) 0x700
- #define [RCC_AHB1ENR_GPIOA_EN](#) 0x01
- #define [RCC_APB2_ENR_USART1_EN](#) 0x10
- #define [USART1_UE](#) 0x01
- #define [USART1_RE](#) 0x04
- #define [USART1_TE](#) 0x08
- #define [USART1_REACK](#) 0x400000
- #define [USART1_TEACK](#) 0x200000
- #define [USART1_TXE](#) 0x80

Functions

- void [uart_init](#) (uint32_t baud_rate)
This function initializes the UART module. It can be used for redirecting the printf function.
- void [uart_send_buff](#) (uint8_t *buff, uint8_t buff_size)
This function sends an array of chars over the UART interface.
- void [_putchar](#) (char character)
This function is used by the [printf\(\)](#) port. It sends a single char over the UART interface. This function is blocking.

5.48.1 Macro Definition Documentation

5.48.1.1 `#define GPIOA_AF7_PA10 0x700`

5.48.1.2 `#define GPIOA_AF7_PA9 0x70`

5.48.1.3 `#define GPIOA_MODER_AF_PA10 0x200000`

5.48.1.4 `#define GPIOA_MODER_AF_PA9 0x80000`

5.48.1.5 `#define GPIOA_OTYPER_PA9 0x200`

5.48.1.6 `#define RCC_AHB1ENR_GPIOA_EN 0x01`

5.48.1.7 `#define RCC_APB2_ENR_USART1_EN 0x10`

5.48.1.8 `#define USART1_RE 0x04`

5.48.1.9 `#define USART1_REACK 0x400000`

5.48.1.10 `#define USART1_TE 0x08`

5.48.1.11 `#define USART1_TEACK 0x200000`

5.48.1.12 `#define USART1_TXE 0x80`

5.48.1.13 `#define USART1_UE 0x01`

5.48.2 Function Documentation

5.48.2.1 `void _putchar (char character)`

This function is used by the `printf()` port. It sends a single char over the UART interface. This function is blocking.

Parameters

<i>character</i>	- an 8-bit ASCII character.
------------------	-----------------------------

Returns

None.

5.48.2.2 `void uart_init (uint32_t baud_rate)`

This function initializes the UART module. It can be used for redirecting the printf function.

To initialize the UART, first the clock of the module has to be enabled. Second, the GPIO pins have to be multiplexed to the UART module. Third, the module has to be enabled and last - the UART baudrate has to be set according to the user settings.

Parameters

<i>baud_rate</i>	- the speed of the data transmission over the interface, in bauds. Can be one of the following integers: , , , .
------------------	--

Returns

None.

5.48.2.3 void uart_send_buff (uint8_t * *buff*, uint8_t *buff_size*)

This function sends an array of chars over the UART interface.

Parameters

<i>buff</i>	- a pointer to an array of 8-bit characters.
<i>buff_size</i>	- the number of elements in the array.

Returns

None.

Index

- - lkc.h, [46](#)
- __attribute__
 - startup.c, [94](#)
- _ebss
 - startup.c, [95](#)
- _edata
 - startup.c, [95](#)
- _estack
 - startup.c, [95](#)
- _menu_init
 - lkc.h, [47](#)
 - menu.c, [62](#)
- _putchar
 - printf.h, [93](#)
 - uart.c, [96](#)
 - uart.h, [98](#)
- _sbss
 - startup.c, [95](#)
- _sdata
 - startup.c, [95](#)
- _sdata_lma
 - startup.c, [95](#)
- ACS_DARROW
 - dialog.h, [54](#)
- ACS_HLINE
 - dialog.h, [54](#)
- ACS_LLCORNER
 - dialog.h, [54](#)
- ACS_LRCORNER
 - dialog.h, [54](#)
- ACS_LTEE
 - dialog.h, [54](#)
- ACS_RTEE
 - dialog.h, [54](#)
- ACS_UARROW
 - dialog.h, [54](#)
- ACS_ULCORNER
 - dialog.h, [54](#)
- ACS_URCORNER
 - dialog.h, [54](#)
- ACS_VLINE
 - dialog.h, [54](#)
- ARCHITECTURE_ID
 - CMakeCCompilerId.c, [30](#)
 - CMakeCXXCompilerId.cpp, [31](#)
- abs_name
 - file, [12](#)
- ahb1enr
 - led.c, [26](#)
- alldefconfig
 - conf.c, [33](#)
- allmodconfig
 - conf.c, [33](#)
- allnoconfig
 - conf.c, [33](#)
- allyesconfig
 - conf.c, [33](#)
- arg
 - out_fct_wrap_type, [18](#)
- atr
 - dialog_color, [8](#)
- attr_clear
 - dialog.h, [54](#)
 - lxdialog/util.c, [59](#)
- autoconf.h, [25](#)
- BEGIN
 - zconf.lex.c, [70](#)
- BUF_SIZE
 - dialog.h, [54](#)
- backtitle
 - dialog_info, [9](#)
- bg
 - dialog_color, [8](#)
- border
 - dialog_info, [9](#)
- buffer, [7](#)
 - parent, [7](#)
 - state, [7](#)
- bus_fault_handler
 - startup.c, [94](#)
 - startup.h, [95](#)
- button_active
 - dialog_info, [9](#)
- button_inactive
 - dialog_info, [9](#)
- button_key_active
 - dialog_info, [9](#)
- button_key_inactive
 - dialog_info, [9](#)
- button_label_active
 - dialog_info, [9](#)
- button_label_inactive
 - dialog_info, [9](#)
- CMakeCCompilerId.c
 - ARCHITECTURE_ID, [30](#)
 - COMPILER_ID, [30](#)

- DEC, [30](#)
- HEX, [30](#)
- info_arch, [31](#)
- info_compiler, [31](#)
- info_platform, [31](#)
- main, [31](#)
- PLATFORM_ID, [31](#)
- CMakeCXXCompilerId.cpp
 - COMPILER_ID, [31](#)
 - DEC, [31](#)
 - HEX, [32](#)
 - info_arch, [32](#)
 - info_compiler, [32](#)
 - info_platform, [32](#)
 - main, [32](#)
 - PLATFORM_ID, [32](#)
- COMMAND
 - zconf.lex.c, [70](#)
- COMPILER_ID
 - CMakeCCompilerId.c, [30](#)
 - CMakeCXXCompilerId.cpp, [31](#)
- CONFIG_
 - lkc.h, [46](#)
- cdebug
 - expr.h, [43](#)
 - zconf.tab.c, [87](#)
- check
 - dialog_info, [9](#)
- check_selected
 - dialog_info, [9](#)
- checklist.c
 - dialog_checklist, [51](#)
- cmdline
 - fixdep.c, [30](#)
- conf.c
 - alldefconfig, [33](#)
 - allmodconfig, [33](#)
 - allnoconfig, [33](#)
 - allyesconfig, [33](#)
 - defconfig, [33](#)
 - listnewconfig, [33](#)
 - oldaskconfig, [33](#)
 - oldconfig, [33](#)
 - olddefconfig, [33](#)
 - randconfig, [33](#)
 - savedefconfig, [33](#)
 - silentoldconfig, [33](#)
- conf.c
 - input_mode, [33](#), [34](#)
 - main, [34](#)
- conf_def_mode
 - lkc.h, [46](#)
- conf_defname
 - confdata.c, [35](#)
- conf_get_autoconfig_name
 - confdata.c, [35](#)
 - lkc.h, [47](#)
- conf_get_changed
 - confdata.c, [35](#)
- conf_get_configname
 - confdata.c, [35](#)
 - lkc.h, [47](#)
- conf_get_default_confname
 - confdata.c, [35](#)
 - lkc.h, [47](#)
- conf_parse
 - zconf.tab.c, [87](#)
- conf_printer, [7](#)
 - print_comment, [7](#)
 - print_symbol, [7](#)
- conf_read
 - confdata.c, [35](#)
- conf_read_simple
 - confdata.c, [35](#)
- conf_set_all_new_symbols
 - confdata.c, [35](#)
 - lkc.h, [47](#)
- conf_set_changed_callback
 - confdata.c, [35](#)
- conf_set_message_callback
 - confdata.c, [35](#)
- conf_write
 - confdata.c, [35](#)
- conf_write_autoconf
 - confdata.c, [35](#)
- conf_write_defconfig
 - confdata.c, [35](#)
- confdata.c
 - conf_defname, [35](#)
 - conf_get_autoconfig_name, [35](#)
 - conf_get_changed, [35](#)
 - conf_get_configname, [35](#)
 - conf_get_default_confname, [35](#)
 - conf_read, [35](#)
 - conf_read_simple, [35](#)
 - conf_set_all_new_symbols, [35](#)
 - conf_set_changed_callback, [35](#)
 - conf_set_message_callback, [35](#)
 - conf_write, [35](#)
 - conf_write_autoconf, [35](#)
 - conf_write_defconfig, [35](#)
 - LINE_GROWTH, [34](#)
 - sym_add_change_count, [35](#)
 - sym_set_change_count, [35](#)
- container_of
 - list.h, [44](#)
- curr
 - symbol, [20](#)
- current_buf
 - zconf.lex.c, [79](#)
- current_file
 - expr.h, [43](#)
 - menu.c, [63](#)
- DEBUG_EXPR
 - expr.c, [36](#)
- DEBUG_PARSE

- zconf.tab.c, 82
- DEC
 - CMakeCCompilerId.c, 30
 - CMakeCXXCompilerId.cpp, 31
- DLG_COLOR
 - lxdialog/util.c, 58
- darrow
 - dialog_info, 9
- data
 - dialog_item, 10
 - menu, 17
- debug_mon_handler
 - startup.c, 94
 - startup.h, 95
- def
 - symbol, 20
- def_default
 - lkc.h, 47
- def_mod
 - lkc.h, 47
- def_no
 - lkc.h, 47
- def_random
 - lkc.h, 47
- def_yes
 - lkc.h, 47
- default_handler
 - startup.c, 94
 - startup.h, 95
- defconfig
 - conf.c, 33
- dep
 - menu, 17
- depfile
 - fixdep.c, 30
- dialog
 - dialog_info, 9
- dialog.h
 - ACS_DARROW, 54
 - ACS_HLINE, 54
 - ACS_LLCORNER, 54
 - ACS_LRCORNER, 54
 - ACS_LTEE, 54
 - ACS_RTEE, 54
 - ACS_UARROW, 54
 - ACS_ULCORNER, 54
 - ACS_URCORNER, 54
 - ACS_VLINE, 54
 - attr_clear, 54
 - BUF_SIZE, 54
 - dialog_checklist, 55
 - dialog_clear, 55
 - dialog_input_result, 56
 - dialog_inputbox, 55
 - dialog_menu, 55
 - dialog_msgbox, 55
 - dialog_textbox, 55
 - dialog_yesno, 55
 - dlg, 56
 - draw_box, 55
 - draw_shadow, 55
 - ERRDISPLAYTOOSMALL, 54
 - end_dialog, 55
 - first_alpha, 55
 - init_dialog, 55
 - item_activate_selected, 55
 - item_add_str, 55
 - item_count, 55
 - item_cur, 56
 - item_data, 55
 - item_foreach, 54
 - item_head, 56
 - item_is_selected, 55
 - item_is_tag, 55
 - item_make, 55
 - item_n, 55
 - item_nil, 56
 - item_reset, 55
 - item_set, 55
 - item_set_data, 55
 - item_set_selected, 55
 - item_set_tag, 55
 - item_str, 55
 - item_tag, 55
 - KEY_ESC, 54
 - M_EVENT, 54
 - MAX, 54
 - MAX_LEN, 54
 - MAXITEMSTR, 54
 - MIN, 54
 - on_key_esc, 55
 - on_key_resize, 56
 - print_autowrap, 56
 - print_button, 56
 - print_title, 56
 - saved_x, 56
 - saved_y, 56
 - set_dialog_backtitle, 56
 - TAB, 54
 - TR, 54
 - update_text_fn, 54
- dialog_checklist
 - checklist.c, 51
 - dialog.h, 55
- dialog_clear
 - dialog.h, 55
 - lxdialog/util.c, 59
- dialog_color, 8
 - atr, 8
 - bg, 8
 - fg, 8
 - hl, 8
- dialog_info, 8
 - backtitle, 9
 - border, 9
 - button_active, 9

- button_inactive, 9
- button_key_active, 9
- button_key_inactive, 9
- button_label_active, 9
- button_label_inactive, 9
- check, 9
- check_selected, 9
- darrow, 9
- dialog, 9
- inputbox, 9
- inputbox_border, 9
- item, 9
- item_selected, 9
- menubox, 9
- menubox_border, 9
- position_indicator, 9
- screen, 9
- searchbox, 9
- searchbox_border, 9
- searchbox_title, 9
- shadow, 10
- tag, 10
- tag_key, 10
- tag_key_selected, 10
- tag_selected, 10
- title, 10
- uarrow, 10
- dialog_input_result
 - dialog.h, 56
 - inputbox.c, 56
- dialog_inputbox
 - dialog.h, 55
 - inputbox.c, 56
- dialog_item, 10
 - data, 10
 - selected, 10
 - str, 10
 - tag, 10
- dialog_list, 10
 - next, 11
 - node, 11
- dialog_menu
 - dialog.h, 55
 - menubox.c, 57
- dialog_msgbox
 - dialog.h, 55
- dialog_textbox
 - dialog.h, 55
 - textbox.c, 57
- dialog_yesno
 - dialog.h, 55
 - yesno.c, 61
- dir_dep
 - symbol, 20
- dlg
 - dialog.h, 56
 - lxdialog/util.c, 60
- draw_box
 - dialog.h, 55
 - lxdialog/util.c, 59
- draw_shadow
 - dialog.h, 55
 - lxdialog/util.c, 59
- e1
 - expr.c, 36
- e2
 - expr.c, 36
- E_AND
 - expr.h, 41
- E_EQUAL
 - expr.h, 41
- E_LIST
 - expr.h, 41
- E_NONE
 - expr.h, 41
- E_NOT
 - expr.h, 41
- E_OR
 - expr.h, 41
- E_RANGE
 - expr.h, 41
- E_SYMBOL
 - expr.h, 41
- E_UNEQUAL
 - expr.h, 41
- ECHO
 - zconf.lex.c, 70
- EOB_ACT_LAST_MATCH
 - zconf.lex.c, 70
- ERRDISPLAYTOOSMALL
 - dialog.h, 54
- EXPR_AND
 - expr.h, 40
- EXPR_NOT
 - expr.h, 40
- EXPR_OR
 - expr.h, 40
- end_dialog
 - dialog.h, 55
 - lxdialog/util.c, 59
- entries
 - jump_key, 14
- expr, 11
 - expr_data, 11
 - expr_value, 12
 - property, 18
 - right, 11
 - type, 11
 - YYSTYPE, 22
- expr.h
 - E_AND, 41
 - E_EQUAL, 41
 - E_LIST, 41
 - E_NONE, 41
 - E_NOT, 41
 - E_OR, 41

- E_RANGE, [41](#)
- E_SYMBOL, [41](#)
- E_UNEQUAL, [41](#)
- mod, [42](#)
- no, [42](#)
- P_CHOICE, [42](#)
- P_COMMENT, [42](#)
- P_DEFAULT, [42](#)
- P_ENV, [42](#)
- P_MENU, [42](#)
- P_PROMPT, [41](#)
- P_RANGE, [42](#)
- P_SELECT, [42](#)
- P_SYMBOL, [42](#)
- P_UNKNOWN, [41](#)
- S_BOOLEAN, [42](#)
- S_DEF_AUTO, [41](#)
- S_DEF_COUNT, [41](#)
- S_DEF_DEF3, [41](#)
- S_DEF_DEF4, [41](#)
- S_DEF_USER, [41](#)
- S_HEX, [42](#)
- S_INT, [42](#)
- S_OTHER, [42](#)
- S_STRING, [42](#)
- S_TRISTATE, [42](#)
- S_UNKNOWN, [42](#)
- yes, [42](#)
- expr.c
 - DEBUG_EXPR, [36](#)
 - e1, [36](#)
 - e2, [36](#)
 - expr_alloc_and, [36](#)
 - expr_alloc_comp, [36](#)
 - expr_alloc_one, [36](#)
 - expr_alloc_or, [37](#)
 - expr_alloc_symbol, [37](#)
 - expr_alloc_two, [37](#)
 - expr_calc_value, [37](#)
 - expr_compare_type, [37](#)
 - expr_contains_symbol, [37](#)
 - expr_copy, [37](#)
 - expr_depends_symbol, [37](#)
 - expr_eliminate_dups, [37](#)
 - expr_eliminate_eq, [37](#)
 - expr_eliminate_yn, [37](#)
 - expr_eq, [37](#)
 - expr_extract_eq, [37](#)
 - expr_extract_eq_and, [37](#)
 - expr_extract_eq_or, [37](#)
 - expr_fprint, [37](#)
 - expr_free, [37](#)
 - expr_gstr_print, [37](#)
 - expr_print, [37](#)
 - expr_simplify_unmet_dep, [37](#)
 - expr_trans_bool, [37](#)
 - expr_trans_compare, [37](#)
 - expr_transform, [37](#)
- expr.h
 - cdebug, [43](#)
 - current_file, [43](#)
 - EXPR_AND, [40](#)
 - EXPR_NOT, [40](#)
 - EXPR_OR, [40](#)
 - expr_alloc_and, [42](#)
 - expr_alloc_comp, [42](#)
 - expr_alloc_one, [42](#)
 - expr_alloc_or, [42](#)
 - expr_alloc_symbol, [42](#)
 - expr_alloc_two, [42](#)
 - expr_calc_value, [42](#)
 - expr_contains_symbol, [42](#)
 - expr_copy, [42](#)
 - expr_depends_symbol, [42](#)
 - expr_eliminate_dups, [43](#)
 - expr_eliminate_eq, [43](#)
 - expr_eliminate_yn, [43](#)
 - expr_eq, [43](#)
 - expr_extract_eq, [43](#)
 - expr_extract_eq_and, [43](#)
 - expr_extract_eq_or, [43](#)
 - expr_fprint, [43](#)
 - expr_free, [43](#)
 - expr_gstr_print, [43](#)
 - expr_list_for_each_sym, [40](#)
 - expr_simplify_unmet_dep, [43](#)
 - expr_trans_bool, [43](#)
 - expr_trans_compare, [43](#)
 - expr_transform, [43](#)
 - expr_type, [41](#)
 - file_list, [43](#)
 - for_all_choices, [40](#)
 - for_all_defaults, [40](#)
 - for_all_prompts, [40](#)
 - for_all_properties, [40](#)
 - for_all_symbols, [40](#)
 - JUMP_NB, [40](#)
 - lookup_file, [43](#)
 - MENU_CHANGED, [40](#)
 - MENU_ROOT, [40](#)
 - modules_sym, [43](#)
 - prop_type, [41](#)
 - SYMBOL_AUTO, [40](#)
 - SYMBOL_CHANGED, [40](#)
 - SYMBOL_CHECK, [40](#)
 - SYMBOL_CHECKED, [40](#)
 - SYMBOL_CHOICE, [40](#)
 - SYMBOL_CHOICEVAL, [40](#)
 - SYMBOL_CONST, [40](#)
 - SYMBOL_DEF, [40](#)
 - SYMBOL_DEF3, [40](#)
 - SYMBOL_DEF4, [40](#)
 - SYMBOL_DEF_AUTO, [41](#)
 - SYMBOL_DEF_USER, [41](#)
 - SYMBOL_HASHSIZE, [41](#)
 - SYMBOL_MAXLENGTH, [41](#)

- SYMBOL_OPTIONAL, 41
- SYMBOL_VALID, 41
- SYMBOL_WARNED, 41
- SYMBOL_WRITE, 41
- sym_defconfig_list, 43
- symbol_mod, 43
- symbol_type, 42
- tristate, 41, 42
- expr_alloc_and
 - expr.c, 36
 - expr.h, 42
- expr_alloc_comp
 - expr.c, 36
 - expr.h, 42
- expr_alloc_one
 - expr.c, 36
 - expr.h, 42
- expr_alloc_or
 - expr.c, 37
 - expr.h, 42
- expr_alloc_symbol
 - expr.c, 37
 - expr.h, 42
- expr_alloc_two
 - expr.c, 37
 - expr.h, 42
- expr_calc_value
 - expr.c, 37
 - expr.h, 42
- expr_compare_type
 - expr.c, 37
- expr_contains_symbol
 - expr.c, 37
 - expr.h, 42
- expr_copy
 - expr.c, 37
 - expr.h, 42
- expr_data, 11
 - expr, 11
 - sym, 11
- expr_depends_symbol
 - expr.c, 37
 - expr.h, 42
- expr_eliminate_dups
 - expr.c, 37
 - expr.h, 43
- expr_eliminate_eq
 - expr.c, 37
 - expr.h, 43
- expr_eliminate_yn
 - expr.c, 37
 - expr.h, 43
- expr_eq
 - expr.c, 37
 - expr.h, 43
- expr_extract_eq
 - expr.c, 37
 - expr.h, 43
- expr_extract_eq_and
 - expr.c, 37
 - expr.h, 43
- expr_extract_eq_or
 - expr.c, 37
 - expr.h, 43
- expr_fprint
 - expr.c, 37
 - expr.h, 43
- expr_free
 - expr.c, 37
 - expr.h, 43
- expr_gstr_print
 - expr.c, 37
 - expr.h, 43
- expr_list_for_each_sym
 - expr.h, 40
- expr_print
 - expr.c, 37
- expr_simplify_unmet_dep
 - expr.c, 37
 - expr.h, 43
- expr_trans_bool
 - expr.c, 37
 - expr.h, 43
- expr_trans_compare
 - expr.c, 37
 - expr.h, 43
- expr_transform
 - expr.c, 37
 - expr.h, 43
- expr_type
 - expr.h, 41
- expr_value, 12
 - expr, 12
 - tri, 12
- FLAGS_ADAPT_EXP
 - printf.c, 89
- FLAGS_CHAR
 - printf.c, 89
- FLAGS_HASH
 - printf.c, 89
- FLAGS_LEFT
 - printf.c, 89
- FLAGS_LONG
 - printf.c, 89
- FLAGS_LONG_LONG
 - printf.c, 89
- FLAGS_PLUS
 - printf.c, 89
- FLAGS_PRECISION
 - printf.c, 89
- FLAGS_SHORT
 - printf.c, 89
- FLAGS_SPACE
 - printf.c, 89
- FLAGS_UPPERCASE
 - printf.c, 89

FLAGS_ZEROPAD
 printf.c, [89](#)
 FLEX_BETA
 zconf.lex.c, [70](#)
 FLEX_SCANNER
 zconf.lex.c, [70](#)
 FLEXINT_H
 zconf.lex.c, [70](#)
 fct
 out_fct_wrap_type, [18](#)
 fctprintf
 printf.c, [90](#)
 printf.h, [93](#)
 fg
 dialog_color, [8](#)
 file, [12](#)
 abs_name, [12](#)
 lineno, [12](#)
 menu, [17](#)
 name, [12](#)
 next, [12](#)
 parent, [12](#)
 property, [18](#)
 YYSTYPE, [22](#)
 zconf.lex.c, [79](#)
 file_list
 expr.h, [43](#)
 menu.c, [63](#)
 file_lookup
 lkc.h, [47](#)
 util.c, [60](#)
 file_write_dep
 lkc.h, [47](#)
 util.c, [60](#)
 first_alpha
 dialog.h, [55](#)
 lxdialog/util.c, [59](#)
 fixdep.c
 cmdline, [30](#)
 depfile, [30](#)
 HASHSZ, [29](#)
 INT_CONF, [29](#)
 INT_FIG_, [29](#)
 INT_NFIG, [29](#)
 INT_ONFI, [30](#)
 main, [30](#)
 target, [30](#)
 flags
 kconf_id, [14](#)
 menu, [17](#)
 symbol, [20](#)
 flex_int16_t
 zconf.lex.c, [75](#)
 flex_int32_t
 zconf.lex.c, [75](#)
 flex_int8_t
 zconf.lex.c, [75](#)
 flex_uint16_t
 zconf.lex.c, [75](#)
 flex_uint32_t
 zconf.lex.c, [75](#)
 flex_uint8_t
 zconf.lex.c, [75](#)
 for_all_choices
 expr.h, [40](#)
 for_all_defaults
 expr.h, [40](#)
 for_all_prompts
 expr.h, [40](#)
 for_all_properties
 expr.h, [40](#)
 for_all_symbols
 expr.h, [40](#)
 GPIOA_AF7_PA10
 uart.h, [98](#)
 GPIOA_AF7_PA9
 uart.h, [98](#)
 GPIOA_MODER_AF_PA10
 uart.h, [98](#)
 GPIOA_MODER_AF_PA9
 uart.h, [98](#)
 GPIOA_MODER_PA12
 led.h, [27](#)
 GPIOA_ODR_PA12
 led.h, [27](#)
 GPIOA_OTYPER_PA12
 led.h, [27](#)
 GPIOA_OTYPER_PA9
 uart.h, [98](#)
 get_relations_str
 menu.c, [62](#)
 get_symbol_str
 menu.c, [62](#)
 gpio_a_moder
 led.c, [26](#)
 gpio_a_odr
 led.c, [26](#)
 gpio_a_otyper
 led.c, [26](#)
 gstr, [13](#)
 len, [13](#)
 max_width, [13](#)
 s, [13](#)
 HASHSZ
 fixdep.c, [29](#)
 HELP
 zconf.lex.c, [70](#)
 HEX
 CMakeCCompilerId.c, [30](#)
 CMakeCXXCompilerId.cpp, [32](#)
 hard_fault_handler
 startup.c, [94](#)
 startup.h, [95](#)
 hash
 item, [13](#)

- head
 - search_data, 19
- help
 - menu, 17
- hl
 - dialog_color, 8
- i
 - zconf.lex.c, 79
- INITIAL
 - zconf.lex.c, 70
- INT16_MAX
 - zconf.lex.c, 70
- INT16_MIN
 - zconf.lex.c, 71
- INT32_MAX
 - zconf.lex.c, 71
- INT32_MIN
 - zconf.lex.c, 71
- INT8_MAX
 - zconf.lex.c, 71
- INT8_MIN
 - zconf.lex.c, 71
- INT_CONF
 - fixdep.c, 29
- INT_FIG_
 - fixdep.c, 29
- INT_NFIG
 - fixdep.c, 29
- INT_ONFI
 - fixdep.c, 30
- id
 - YYSTYPE, 22
- if
 - zconf.lex.c, 75
- index
 - jump_key, 14
- info_arch
 - CMakeCCompilerId.c, 31
 - CMakeCXXCompilerId.cpp, 32
- info_compiler
 - CMakeCCompilerId.c, 31
 - CMakeCXXCompilerId.cpp, 32
- info_platform
 - CMakeCCompilerId.c, 31
 - CMakeCXXCompilerId.cpp, 32
- init_dialog
 - dialog.h, 55
 - lxdialog/util.c, 59
- input_mode
 - conf.c, 33, 34
- inputbox
 - dialog_info, 9
- inputbox.c
 - dialog_input_result, 56
 - dialog_inputbox, 56
- inputbox_border
 - dialog_info, 9
- item, 13
 - dialog_info, 9
 - hash, 13
 - len, 13
 - name, 13
 - next, 13
 - item_activate_selected
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_add_str
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_count
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_cur
 - dialog.h, 56
 - lxdialog/util.c, 60
 - item_data
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_foreach
 - dialog.h, 54
 - item_head
 - dialog.h, 56
 - lxdialog/util.c, 60
 - item_is_selected
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_is_tag
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_make
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_n
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_nil
 - dialog.h, 56
 - lxdialog/util.c, 60
 - item_reset
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_selected
 - dialog_info, 9
 - item_set
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_set_data
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_set_selected
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_set_tag
 - dialog.h, 55
 - lxdialog/util.c, 59
 - item_str

- dialog.h, [55](#)
 - lxdialog/util.c, [59](#)
- item_tag
 - dialog.h, [55](#)
 - lxdialog/util.c, [59](#)
- JUMP_NB
 - expr.h, [40](#)
- jump_key, [13](#)
 - entries, [14](#)
 - index, [14](#)
 - offset, [14](#)
 - target, [14](#)
- KEY_ESC
 - dialog.h, [54](#)
- kconf_id, [14](#)
 - flags, [14](#)
 - name, [14](#)
 - stype, [14](#)
 - token, [14](#)
- kconf_id_strings
 - zconf.hash.c, [67](#)
- kconf_id_strings_str12
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str13
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str14
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str17
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str18
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str2
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str21
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str22
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str23
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str25
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str27
 - kconf_id_strings_t, [15](#)
- kconf_id_strings_str29
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str3
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str31
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str32
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str33
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str35
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str36
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str39
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str41
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str42
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str43
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str46
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str47
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str5
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str51
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str54
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str56
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str7
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str72
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str8
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_str9
 - kconf_id_strings_t, [16](#)
- kconf_id_strings_t, [14](#)
 - kconf_id_strings_str12, [15](#)
 - kconf_id_strings_str13, [15](#)
 - kconf_id_strings_str14, [15](#)
 - kconf_id_strings_str17, [15](#)
 - kconf_id_strings_str18, [15](#)
 - kconf_id_strings_str2, [15](#)
 - kconf_id_strings_str21, [15](#)
 - kconf_id_strings_str22, [15](#)
 - kconf_id_strings_str23, [15](#)
 - kconf_id_strings_str25, [15](#)
 - kconf_id_strings_str27, [15](#)
 - kconf_id_strings_str29, [16](#)
 - kconf_id_strings_str3, [16](#)
 - kconf_id_strings_str31, [16](#)
 - kconf_id_strings_str32, [16](#)
 - kconf_id_strings_str33, [16](#)
 - kconf_id_strings_str35, [16](#)
 - kconf_id_strings_str36, [16](#)
 - kconf_id_strings_str39, [16](#)
 - kconf_id_strings_str41, [16](#)
 - kconf_id_strings_str42, [16](#)
 - kconf_id_strings_str43, [16](#)
 - kconf_id_strings_str46, [16](#)
 - kconf_id_strings_str47, [16](#)
 - kconf_id_strings_str5, [16](#)
 - kconf_id_strings_str51, [16](#)
 - kconf_id_strings_str54, [16](#)
 - kconf_id_strings_str56, [16](#)
 - kconf_id_strings_str7, [16](#)

- kconf_id_strings_str72, 16
 - kconf_id_strings_str8, 16
 - kconf_id_strings_str9, 16
- keys
 - search_data, 19
- LED_BLINK
 - main.c, 28
- LINE_GROWTH
 - confdata.c, 34
- LIST_HEAD
 - list.h, 44
- LIST_HEAD_INIT
 - list.h, 44
- LOCALEDIR
 - lkc.h, 46
- led.c
 - ahb1enr, 26
 - gpio_a_moder, 26
 - gpio_a_odr, 26
 - gpio_a_otyper, 26
 - led_clear, 26
 - led_init, 26
 - led_set, 26
- led.h
 - GPIOA_MODER_PA12, 27
 - GPIOA_ODR_PA12, 27
 - GPIOA_OTYPER_PA12, 27
 - led_clear, 27
 - led_init, 27
 - led_set, 28
- led/led.c, 25
- led/led.h, 26
- led_clear
 - led.c, 26
 - led.h, 27
- led_init
 - led.c, 26
 - led.h, 27
- led_set
 - led.c, 26
 - led.h, 28
- len
 - gstr, 13
 - item, 13
- lineno
 - file, 12
 - menu, 17
 - property, 18
 - zconf.lex.c, 79
- list
 - menu, 17
- list.h
 - container_of, 44
 - LIST_HEAD, 44
 - LIST_HEAD_INIT, 44
 - list_entry, 44
 - list_for_each_entry, 44
 - offsetof, 44
- list_entry
 - list.h, 44
- list_for_each_entry
 - list.h, 44
- list_head, 16
 - next, 17
 - prev, 17
- listnewconfig
 - conf.c, 33
- lkc.h
 - def_default, 47
 - def_mod, 47
 - def_no, 47
 - def_random, 47
 - def_yes, 47
- lkc.h
 - _, 46
 - _menu_init, 47
 - CONFIG_, 46
 - conf_def_mode, 46
 - conf_get_autoconfig_name, 47
 - conf_get_configname, 47
 - conf_get_default_confname, 47
 - conf_set_all_new_symbols, 47
 - file_lookup, 47
 - file_write_dep, 47
 - LOCALEDIR, 46
 - menu_add_dep, 47
 - menu_add_entry, 47
 - menu_add_expr, 47
 - menu_add_menu, 47
 - menu_add_option, 47
 - menu_add_prompt, 47
 - menu_add_prop, 47
 - menu_add_symbol, 47
 - menu_add_visibility, 47
 - menu_end_entry, 47
 - menu_end_menu, 47
 - menu_finalize, 47
 - menu_set_type, 47
 - menu_warn, 47
 - N_, 46
 - P, 46
 - PACKAGE, 46
 - prop_alloc, 47
 - prop_get_symbol, 47
 - SRCTREE, 46
 - str_append, 48
 - str_assign, 48
 - str_free, 48
 - str_get, 48
 - str_new, 48
 - str_printf, 48
 - sym_add_change_count, 48
 - sym_check_deps, 48
 - sym_choice_default, 48
 - sym_clear_all_valid, 48
 - sym_env_list, 48

- sym_get_env_prop, 48
- sym_get_string_default, 48
- sym_init, 48
- sym_set_all_changed, 48
- sym_set_change_count, 48
- sym_set_changed, 48
- T_OPT_ENV, 46
- T_OPT_MODULES, 46
- TF_COMMAND, 46
- TF_OPTION, 46
- TF_PARAM, 46
- zconf_curname, 48
- zconf_fopen, 48
- zconf_initscan, 48
- zconf_lineno, 48
- zconf_nextfile, 48
- zconf_starthelp, 48
- zconfdebug, 48
- zconfdump, 48
- zconfparse, 48
- lkc_proto.h
 - P, 50, 51
- lookup_file
 - expr.h, 43
- lxdialog/util.c
 - attr_clear, 59
 - DLG_COLOR, 58
 - dialog_clear, 59
 - dlg, 60
 - draw_box, 59
 - draw_shadow, 59
 - end_dialog, 59
 - first_alpha, 59
 - init_dialog, 59
 - item_activate_selected, 59
 - item_add_str, 59
 - item_count, 59
 - item_cur, 60
 - item_data, 59
 - item_head, 60
 - item_is_selected, 59
 - item_is_tag, 59
 - item_make, 59
 - item_n, 59
 - item_nil, 60
 - item_reset, 59
 - item_set, 59
 - item_set_data, 59
 - item_set_selected, 59
 - item_set_tag, 59
 - item_str, 59
 - item_tag, 59
 - on_key_esc, 59
 - on_key_resize, 59
 - print_autowrap, 59
 - print_button, 59
 - print_title, 59
 - saved_x, 60
 - saved_y, 60
 - set_dialog_backtitle, 59
- M_EVENT
 - dialog.h, 54
- MAX
 - dialog.h, 54
- MAX_LEN
 - dialog.h, 54
- MAXITEMSTR
 - dialog.h, 54
- MENU_CHANGED
 - expr.h, 40
- MENU_ROOT
 - expr.h, 40
- MIN
 - dialog.h, 54
- main
 - CMakeCCompilerId.c, 31
 - CMakeCXXCompilerId.cpp, 32
 - conf.c, 34
 - fixdep.c, 30
 - main.c, 28
 - mconf.c, 61
 - startup.c, 94
- main.c, 28
 - LED_BLINK, 28
 - main, 28
- main.h, 29
- max_width
 - gstr, 13
- mconf.c
 - main, 61
- mem_manage_handler
 - startup.c, 94
 - startup.h, 95
- menu, 17
 - data, 17
 - dep, 17
 - file, 17
 - flags, 17
 - help, 17
 - lineno, 17
 - list, 17
 - next, 17
 - parent, 17
 - prompt, 17
 - property, 18
 - sym, 17
 - visibility, 18
 - YYSTYPE, 22
- menu.c
 - _menu_init, 62
 - current_file, 63
 - file_list, 63
 - get_relations_str, 62
 - get_symbol_str, 62
 - menu_add_dep, 62
 - menu_add_entry, 62

- menu_add_expr, 62
- menu_add_menu, 62
- menu_add_option, 62
- menu_add_prompt, 62
- menu_add_prop, 62
- menu_add_symbol, 63
- menu_add_visibility, 63
- menu_end_entry, 63
- menu_end_menu, 63
- menu_finalize, 63
- menu_get_ext_help, 63
- menu_get_help, 63
- menu_get_parent_menu, 63
- menu_get_prompt, 63
- menu_get_root_menu, 63
- menu_has_help, 63
- menu_has_prompt, 63
- menu_is_visible, 63
- menu_set_type, 63
- menu_warn, 63
- rootmenu, 63
- menu_add_dep
 - lkc.h, 47
 - menu.c, 62
- menu_add_entry
 - lkc.h, 47
 - menu.c, 62
- menu_add_expr
 - lkc.h, 47
 - menu.c, 62
- menu_add_menu
 - lkc.h, 47
 - menu.c, 62
- menu_add_option
 - lkc.h, 47
 - menu.c, 62
- menu_add_prompt
 - lkc.h, 47
 - menu.c, 62
- menu_add_prop
 - lkc.h, 47
 - menu.c, 62
- menu_add_symbol
 - lkc.h, 47
 - menu.c, 63
- menu_add_visibility
 - lkc.h, 47
 - menu.c, 63
- menu_end_entry
 - lkc.h, 47
 - menu.c, 63
- menu_end_menu
 - lkc.h, 47
 - menu.c, 63
- menu_finalize
 - lkc.h, 47
 - menu.c, 63
- menu_get_ext_help
 - menu.c, 63
- menu_get_help
 - menu.c, 63
- menu_get_parent_menu
 - menu.c, 63
- menu_get_prompt
 - menu.c, 63
- menu_get_root_menu
 - menu.c, 63
- menu_has_help
 - menu.c, 63
- menu_has_prompt
 - menu.c, 63
- menu_is_visible
 - menu.c, 63
- menu_set_type
 - lkc.h, 47
 - menu.c, 63
- menu_warn
 - lkc.h, 47
 - menu.c, 63
- menubox
 - dialog_info, 9
- menubox.c
 - dialog_menu, 57
 - print_item, 57
- menubox_border
 - dialog_info, 9
- menuconfig/README.md, 63
- menuconfig/basic/fixdep.c, 29
- menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdC/CompilerIdC.c, 30
- menuconfig/build/CMakeFiles/2.8.12.2/CompilerIdCXX/CompilerIdCXX.c, 31
- menuconfig/build/tmp/gen/choice1.h, 32
- menuconfig/build/tmp/menudependency.h, 32
- menuconfig/build/tmp/num/param.h, 32
- menuconfig/build/tmp/num/param2.h, 33
- menuconfig/build/tmp/string/param.h, 32
- menuconfig/conf.c, 33
- menuconfig/confdata.c, 34
- menuconfig/expr.c, 35
- menuconfig/expr.h, 37
- menuconfig/list.h, 43
- menuconfig/lkc.h, 44
- menuconfig/lkc_proto.h, 49
- menuconfig/lxdialog/checklist.c, 51
- menuconfig/lxdialog/dialog.h, 52
- menuconfig/lxdialog/inputbox.c, 56
- menuconfig/lxdialog/menubox.c, 56
- menuconfig/lxdialog/textbox.c, 57
- menuconfig/lxdialog/util.c, 57
- menuconfig/lxdialog/yesno.c, 61
- menuconfig/mconf.c, 61
- menuconfig/menu.c, 61
- menuconfig/symbol.c, 63
- menuconfig/util.c, 60
- menuconfig/zconf.hash.c, 66

- menuconfig/zconf.lex.c, 67
- menuconfig/zconf.tab.c, 80
- mod
 - expr.h, 42
- modules_sym
 - expr.h, 43
 - symbol.c, 66
- modules_val
 - symbol.c, 66
- N_
 - lkc.h, 46
- name
 - file, 12
 - item, 13
 - kconf_id, 14
 - symbol, 20
- next
 - dialog_list, 11
 - file, 12
 - item, 13
 - list_head, 17
 - menu, 17
 - property, 18
 - symbol, 20
- nmi_handler
 - startup.c, 94
 - startup.h, 95
- no
 - expr.h, 42
- node
 - dialog_list, 11
- offset
 - jump_key, 14
- offsetof
 - list.h, 44
- oldaskconfig
 - conf.c, 33
- oldconfig
 - conf.c, 33
- olddefconfig
 - conf.c, 33
- on_key_esc
 - dialog.h, 55
 - lxdialog/util.c, 59
- on_key_resize
 - dialog.h, 56
 - lxdialog/util.c, 59
- out_fct_type
 - printf.c, 89
- out_fct_wrap_type, 18
 - arg, 18
 - fct, 18
- P
 - lkc.h, 46
 - lkc_proto.h, 50, 51
- P_CHOICE
 - expr.h, 42
- P_COMMENT
 - expr.h, 42
- P_DEFAULT
 - expr.h, 42
- P_ENV
 - expr.h, 42
- P_MENU
 - expr.h, 42
- P_PROMPT
 - expr.h, 41
- P_RANGE
 - expr.h, 42
- P_SELECT
 - expr.h, 42
- P_SYMBOL
 - expr.h, 42
- P_UNKNOWN
 - expr.h, 41
- PACKAGE
 - lkc.h, 46
- PARAM
 - zconf.lex.c, 71
- PATH_MAX
 - zconf.tab.c, 82
- PLATFORM_ID
 - CMakeCCompilerId.c, 31
 - CMakeCXXCompilerId.cpp, 32
- PRINTD
 - zconf.tab.c, 82
- PRINTF_MAX_FLOAT
 - printf.c, 89
- PRINTF_SUPPORT_FLOAT
 - printf.c, 89
- parent
 - buffer, 7
 - file, 12
 - menu, 17
- pend_sv_handler
 - startup.c, 94
 - startup.h, 95
- position_indicator
 - dialog_info, 9
- prev
 - list_head, 17
- print_autowrap
 - dialog.h, 56
 - lxdialog/util.c, 59
- print_button
 - dialog.h, 56
 - lxdialog/util.c, 59
- print_comment
 - conf_printer, 7
- print_item
 - menubox.c, 57
- print_symbol
 - conf_printer, 7
- print_title

- dialog.h, 56
- lxdialog/util.c, 59
- printf
 - zconf.tab.c, 82
- printf
 - printf.h, 91
- printf.c
 - FLAGS_ADAPT_EXP, 89
 - FLAGS_CHAR, 89
 - FLAGS_HASH, 89
 - FLAGS_LEFT, 89
 - FLAGS_LONG, 89
 - FLAGS_LONG_LONG, 89
 - FLAGS_PLUS, 89
 - FLAGS_PRECISION, 89
 - FLAGS_SHORT, 89
 - FLAGS_SPACE, 89
 - FLAGS_UPPERCASE, 89
 - FLAGS_ZEROPAD, 89
 - fctprintf, 90
 - out_fct_type, 89
 - PRINTF_MAX_FLOAT, 89
 - printf_, 90
 - snprintf_, 90
 - sprintf_, 90
 - vprintf_, 90
 - vsprintf_, 90
- printf.h
 - _putchar, 93
 - fctprintf, 93
 - printf, 91
 - printf_, 93
 - snprintf, 92
 - snprintf_, 93
 - sprintf, 92
 - sprintf_, 93
 - vprintf, 92
 - vprintf_, 93
 - vsprintf, 92
 - vsprintf_, 93
- printf/printf.c, 88
- printf/printf.h, 90
- printf_
 - printf.c, 90
 - printf.h, 93
- prompt
 - menu, 17
- prop
 - symbol, 20
- prop_alloc
 - lkc.h, 47
 - symbol.c, 65
- prop_get_symbol
 - lkc.h, 47
 - symbol.c, 65
- prop_get_type_name
 - symbol.c, 65
- prop_type
 - expr.h, 41
- property, 18
 - expr, 18
 - file, 18
 - lineno, 18
 - menu, 18
 - next, 18
 - sym, 19
 - text, 19
 - type, 19
 - visible, 19
- RCC_AHB1ENR_GPIOA_EN
 - led.h, 27
 - uart.h, 98
- REJECT
 - zconf.lex.c, 71
- randconfig
 - conf.c, 33
- reset_handler
 - startup.c, 94
 - startup.h, 95
- rev_dep
 - symbol, 20
- right
 - expr, 11
- rootmenu
 - menu.c, 63
- s
 - gstr, 13
- S_BOOLEAN
 - expr.h, 42
- S_DEF_AUTO
 - expr.h, 41
- S_DEF_COUNT
 - expr.h, 41
- S_DEF_DEF3
 - expr.h, 41
- S_DEF_DEF4
 - expr.h, 41
- S_DEF_USER
 - expr.h, 41
- S_HEX
 - expr.h, 42
- S_INT
 - expr.h, 42
- S_OTHER
 - expr.h, 42
- S_STRING
 - expr.h, 42
- S_TRISTATE
 - expr.h, 42
- S_UNKNOWN
 - expr.h, 42
- SRCTREE
 - lkc.h, 46
- START_STRSIZE
 - zconf.lex.c, 71

STRING
 zconf.lex.c, 71
 SYMBOL_AUTO
 expr.h, 40
 SYMBOL_CHANGED
 expr.h, 40
 SYMBOL_CHECK
 expr.h, 40
 SYMBOL_CHECKED
 expr.h, 40
 SYMBOL_CHOICE
 expr.h, 40
 SYMBOL_CHOICEVAL
 expr.h, 40
 SYMBOL_CONST
 expr.h, 40
 SYMBOL_DEF
 expr.h, 40
 SYMBOL_DEF3
 expr.h, 40
 SYMBOL_DEF4
 expr.h, 40
 SYMBOL_DEF_AUTO
 expr.h, 41
 SYMBOL_DEF_USER
 expr.h, 41
 SYMBOL_HASHSIZE
 expr.h, 41
 SYMBOL_MAXLENGTH
 expr.h, 41
 SYMBOL_OPTIONAL
 expr.h, 41
 SYMBOL_VALID
 expr.h, 41
 SYMBOL_WARNED
 expr.h, 41
 SYMBOL_WRITE
 expr.h, 41
 saved_x
 dialog.h, 56
 lxdialog/util.c, 60
 saved_y
 dialog.h, 56
 lxdialog/util.c, 60
 savedefconfig
 conf.c, 33
 screen
 dialog_info, 9
 search_data, 19
 head, 19
 keys, 19
 targets, 19
 searchbox
 dialog_info, 9
 searchbox_border
 dialog_info, 9
 searchbox_title
 dialog_info, 9
 selected
 dialog_item, 10
 set_dialog_backtitle
 dialog.h, 56
 lxdialog/util.c, 59
 shadow
 dialog_info, 10
 silentoldconfig
 conf.c, 33
 snprintf
 printf.h, 92
 snprintf_
 printf.c, 90
 printf.h, 93
 sprintf
 printf.h, 92
 sprintf_
 printf.c, 90
 printf.h, 93
 startup.c, 93
 __attribute__, 94
 _ebss, 95
 _edata, 95
 _estack, 95
 _sbss, 95
 _sdata, 95
 _sdata_lma, 95
 bus_fault_handler, 94
 debug_mon_handler, 94
 default_handler, 94
 hard_fault_handler, 94
 main, 94
 mem_manage_handler, 94
 nmi_handler, 94
 pend_sv_handler, 94
 reset_handler, 94
 svc_handler, 94
 systick_handler, 94
 usage_fault_handler, 94
 startup.h, 95
 bus_fault_handler, 95
 debug_mon_handler, 95
 default_handler, 95
 hard_fault_handler, 95
 mem_manage_handler, 95
 nmi_handler, 95
 pend_sv_handler, 95
 reset_handler, 95
 svc_handler, 95
 systick_handler, 96
 usage_fault_handler, 96
 state
 buffer, 7
 str
 dialog_item, 10
 zconf.lex.c, 79
 str_append
 lkc.h, 48

- util.c, 60
- str_assign
 - lkc.h, 48
 - util.c, 60
- str_free
 - lkc.h, 48
 - util.c, 60
- str_get
 - lkc.h, 48
 - util.c, 60
- str_new
 - lkc.h, 48
 - util.c, 60
- str_printf
 - lkc.h, 48
 - util.c, 60
- string
 - YYSTYPE, 22
- stype
 - kconf_id, 14
- svc_handler
 - startup.c, 94
 - startup.h, 95
- sym
 - expr_data, 11
 - menu, 17
 - property, 19
- sym_add_change_count
 - confdata.c, 35
 - lkc.h, 48
- sym_calc_value
 - symbol.c, 65
- sym_check_deps
 - lkc.h, 48
 - symbol.c, 65
- sym_choice_default
 - lkc.h, 48
 - symbol.c, 65
- sym_clear_all_valid
 - lkc.h, 48
 - symbol.c, 65
- sym_defconfig_list
 - expr.h, 43
 - symbol.c, 66
- sym_env_list
 - lkc.h, 48
 - symbol.c, 66
- sym_escape_string_value
 - symbol.c, 65
- sym_expand_string_value
 - symbol.c, 65
- sym_find
 - symbol.c, 65
- sym_get_choice_prop
 - symbol.c, 65
- sym_get_default_prop
 - symbol.c, 65
- sym_get_env_prop
 - lkc.h, 48
 - symbol.c, 65
- sym_get_string_default
 - lkc.h, 48
 - symbol.c, 65
- sym_get_string_value
 - symbol.c, 65
- sym_get_type
 - symbol.c, 65
- sym_init
 - lkc.h, 48
 - symbol.c, 65
- sym_is_changable
 - symbol.c, 65
- sym_lookup
 - symbol.c, 65
- sym_re_search
 - symbol.c, 65
- sym_set_all_changed
 - lkc.h, 48
 - symbol.c, 65
- sym_set_change_count
 - confdata.c, 35
 - lkc.h, 48
- sym_set_changed
 - lkc.h, 48
 - symbol.c, 65
- sym_set_string_value
 - symbol.c, 65
- sym_set_tristate_value
 - symbol.c, 65
- sym_string_valid
 - symbol.c, 65
- sym_string_within_range
 - symbol.c, 65
- sym_toggle_tristate_value
 - symbol.c, 65
- sym_tristate_within_range
 - symbol.c, 65
- sym_type_name
 - symbol.c, 66
- symbol, 19
 - curr, 20
 - def, 20
 - dir_dep, 20
 - flags, 20
 - name, 20
 - next, 20
 - prop, 20
 - rev_dep, 20
 - type, 20
 - visible, 20
 - YYSTYPE, 22
- symbol.c
 - modules_sym, 66
 - modules_val, 66
 - prop_alloc, 65
 - prop_get_symbol, 65

- prop_get_type_name, 65
- sym_calc_value, 65
- sym_check_deps, 65
- sym_choice_default, 65
- sym_clear_all_valid, 65
- sym_defconfig_list, 66
- sym_env_list, 66
- sym_escape_string_value, 65
- sym_expand_string_value, 65
- sym_find, 65
- sym_get_choice_prop, 65
- sym_get_default_prop, 65
- sym_get_env_prop, 65
- sym_get_string_default, 65
- sym_get_string_value, 65
- sym_get_type, 65
- sym_init, 65
- sym_is_changable, 65
- sym_lookup, 65
- sym_re_search, 65
- sym_set_all_changed, 65
- sym_set_changed, 65
- sym_set_string_value, 65
- sym_set_tristate_value, 65
- sym_string_valid, 65
- sym_string_within_range, 65
- sym_toggle_tristate_value, 65
- sym_tristate_within_range, 65
- sym_type_name, 66
- symbol_empty, 66
- symbol_mod, 66
- symbol_no, 66
- symbol_yes, 66
- symbol_empty
 - symbol.c, 66
- symbol_hash
 - zconf.tab.c, 87
- symbol_mod
 - expr.h, 43
 - symbol.c, 66
- symbol_no
 - symbol.c, 66
- symbol_type
 - expr.h, 42
- symbol_value, 20
 - tri, 20
 - val, 20
- symbol_yes
 - symbol.c, 66
- systick_handler
 - startup.c, 94
 - startup.h, 96
- T_AND
 - zconf.tab.c, 87
- T_CHOICE
 - zconf.tab.c, 87
- T_CLOSE_PAREN
 - zconf.tab.c, 87
- T_COMMENT
 - zconf.tab.c, 87
- T_CONFIG
 - zconf.tab.c, 87
- T_DEFAULT
 - zconf.tab.c, 87
- T_DEPENDS
 - zconf.tab.c, 87
- T_ENDCHOICE
 - zconf.tab.c, 87
- T_ENDIF
 - zconf.tab.c, 87
- T_ENDMENU
 - zconf.tab.c, 86
- T_EOL
 - zconf.tab.c, 87
- T_EQUAL
 - zconf.tab.c, 87
- T_HELP
 - zconf.tab.c, 87
- T_HELPTEXT
 - zconf.tab.c, 87
- T_IF
 - zconf.tab.c, 87
- T_MAINMENU
 - zconf.tab.c, 86
- T_MENU
 - zconf.tab.c, 86
- T_MENUCONFIG
 - zconf.tab.c, 87
- T_NOT
 - zconf.tab.c, 87
- T_ON
 - zconf.tab.c, 87
- T_OPEN_PAREN
 - zconf.tab.c, 87
- T_OPTION
 - zconf.tab.c, 87
- T_OPTIONAL
 - zconf.tab.c, 87
- T_OR
 - zconf.tab.c, 87
- T_PROMPT
 - zconf.tab.c, 87
- T_RANGE
 - zconf.tab.c, 87
- T_SELECT
 - zconf.tab.c, 87
- T_SOURCE
 - zconf.tab.c, 87
- T_TYPE
 - zconf.tab.c, 87
- T_UNEQUAL
 - zconf.tab.c, 87
- T_VISIBLE
 - zconf.tab.c, 87
- T_WORD
 - zconf.tab.c, 87

T_WORD_QUOTE
 zconf.tab.c, 87
 T_OPT_ENV
 lkc.h, 46
 T_OPT_MODULES
 lkc.h, 46
 TAB
 dialog.h, 54
 TF_COMMAND
 lkc.h, 46
 TF_OPTION
 lkc.h, 46
 TF_PARAM
 lkc.h, 46
 TR
 dialog.h, 54
 tag
 dialog_info, 10
 dialog_item, 10
 tag_key
 dialog_info, 10
 tag_key_selected
 dialog_info, 10
 tag_selected
 dialog_info, 10
 target
 fixdep.c, 30
 jump_key, 14
 targets
 search_data, 19
 text
 property, 19
 textbox.c
 dialog_textbox, 57
 title
 dialog_info, 10
 tmp/led/blink/iterator/200000.h, 96
 tmp/led/blink/iterator/400000.h, 96
 tmp/led/blink/iterator/500000.h, 96
 tmp/menudependency.h, 32
 tmp/num/param.h, 32
 tmp/string/param.h, 32
 tmp/uart/ baud/rate.h, 96
 token
 kconf_id, 14
 tri
 expr_value, 12
 symbol_value, 20
 tristate
 expr.h, 41, 42
 ts
 zconf.lex.c, 79
 type
 expr, 11
 property, 19
 symbol, 20
 UINT16_MAX
 zconf.lex.c, 71
 UINT32_MAX
 zconf.lex.c, 71
 UINT8_MAX
 zconf.lex.c, 71
 USART1_RE
 uart.h, 98
 USART1_REACK
 uart.h, 98
 USART1_TE
 uart.h, 98
 USART1_TEACK
 uart.h, 98
 USART1_TXE
 uart.h, 98
 USART1_UE
 uart.h, 98
 uarrow
 dialog_info, 10
 uart.c
 _putchar, 96
 uart_init, 96
 uart_send_buff, 97
 uart.h
 _putchar, 98
 GPIOA_AF7_PA10, 98
 GPIOA_AF7_PA9, 98
 GPIOA_MODER_AF_PA10, 98
 GPIOA_MODER_AF_PA9, 98
 GPIOA_OTYPER_PA9, 98
 USART1_RE, 98
 USART1_REACK, 98
 USART1_TE, 98
 USART1_TEACK, 98
 USART1_TXE, 98
 USART1_UE, 98
 uart_init, 98
 uart_send_buff, 98
 uart/uart.c, 96
 uart/uart.h, 97
 uart_init
 uart.c, 96
 uart.h, 98
 uart_send_buff
 uart.c, 97
 uart.h, 98
 unput
 zconf.lex.c, 71
 update_text_fn
 dialog.h, 54
 usage_fault_handler
 startup.c, 94
 startup.h, 96
 util.c
 file_lookup, 60
 file_write_dep, 60
 str_append, 60
 str_assign, 60
 str_free, 60

- str_get, [60](#)
- str_new, [60](#)
- str_printf, [60](#)
- val
 - symbol_value, [20](#)
- visibility
 - menu, [18](#)
- visible
 - property, [19](#)
 - symbol, [20](#)
- vprintf
 - printf.h, [92](#)
- vprintf_
 - printf.c, [90](#)
 - printf.h, [93](#)
- vsprintf
 - printf.h, [92](#)
- vsprintf_
 - printf.c, [90](#)
 - printf.h, [93](#)
- while
 - zconf.lex.c, [75](#)
- YY_
 - zconf.tab.c, [82](#)
- YY_AT_BOL
 - zconf.lex.c, [71](#)
- YY_BREAK
 - zconf.lex.c, [71](#)
- YY_BUF_SIZE
 - zconf.lex.c, [71](#)
- YY_BUFFER_NEW
 - zconf.lex.c, [71](#)
- YY_BUFFER_NORMAL
 - zconf.lex.c, [71](#)
- YY_BUFFER_STATE
 - zconf.lex.c, [75](#)
- YY_CHAR
 - zconf.lex.c, [75](#)
- YY_CURRENT_BUFFER
 - zconf.lex.c, [71](#)
- YY_DECL
 - zconf.lex.c, [71](#), [79](#)
- YY_DECL_IS_OURS
 - zconf.lex.c, [71](#)
- YY_END_OF_BUFFER
 - zconf.lex.c, [72](#)
- YY_EXIT_FAILURE
 - zconf.lex.c, [72](#)
- YY_EXTRA_TYPE
 - zconf.lex.c, [72](#)
- YY_FATAL_ERROR
 - zconf.lex.c, [72](#)
- YY_FLUSH_BUFFER
 - zconf.lex.c, [72](#)
- YY_INPUT
 - zconf.lex.c, [72](#)
- YY_INT_ALIGNED
 - zconf.lex.c, [72](#)
- YY_LESS_LINENO
 - zconf.lex.c, [72](#)
- YY_LOCATION_PRINT
 - zconf.tab.c, [82](#)
- YY_MORE_ADJ
 - zconf.lex.c, [72](#)
- YY_NEW_FILE
 - zconf.lex.c, [73](#)
- YY_NO_INPUT
 - zconf.lex.c, [73](#)
- YY_NULL
 - zconf.lex.c, [73](#)
- YY_NUM_RULES
 - zconf.lex.c, [73](#)
- YY_READ_BUF_SIZE
 - zconf.lex.c, [73](#)
- YY_REDUCE_PRINT
 - zconf.tab.c, [82](#)
- YY_RULE_SETUP
 - zconf.lex.c, [73](#)
- YY_SC_TO_UI
 - zconf.lex.c, [73](#)
- YY_SKIP_YWRAP
 - zconf.lex.c, [73](#)
- YY_STACK_PRINT
 - zconf.tab.c, [82](#)
- YY_START
 - zconf.lex.c, [73](#)
- YY_STATE_BUF_SIZE
 - zconf.lex.c, [73](#)
- YY_STATE_EOF
 - zconf.lex.c, [73](#)
- YY_SYMBOL_PRINT
 - zconf.tab.c, [83](#)
- YY_USER_ACTION
 - zconf.lex.c, [74](#)
- YYABORT
 - zconf.tab.c, [83](#)
- YYACCEPT
 - zconf.tab.c, [83](#)
- YYBACKUP
 - zconf.tab.c, [83](#)
- YYBISON
 - zconf.tab.c, [83](#)
- YYBISON_VERSION
 - zconf.tab.c, [83](#)
- YYCOPY
 - zconf.tab.c, [83](#)
- YYDEBUG
 - zconf.tab.c, [84](#)
- YYDPRINTF
 - zconf.tab.c, [84](#)
- YYEMPTY
 - zconf.tab.c, [84](#)
- YYEOF
 - zconf.tab.c, [84](#)

- YYERRCODE
 - zconf.tab.c, [84](#)
- YYERROR
 - zconf.tab.c, [84](#)
- YYERROR_VERBOSE
 - zconf.tab.c, [84](#)
- YYFAIL
 - zconf.tab.c, [84](#)
- YYFINAL
 - zconf.tab.c, [84](#)
- YYFPRINTF
 - zconf.tab.c, [84](#)
- YYFREE
 - zconf.tab.c, [84](#)
- YYID
 - zconf.tab.c, [84](#)
- YYINITDEPTH
 - zconf.tab.c, [84](#)
- YYLAST
 - zconf.tab.c, [84](#)
- YYLEX
 - zconf.tab.c, [84](#)
- YYLLOC_DEFAULT
 - zconf.tab.c, [84](#)
- YYLSP_NEEDED
 - zconf.tab.c, [85](#)
- YYMALLOC
 - zconf.tab.c, [85](#)
- YYMAXDEPTH
 - zconf.tab.c, [85](#)
- YYMAXUTOK
 - zconf.tab.c, [85](#)
- YYNNTS
 - zconf.tab.c, [85](#)
- YYNRULES
 - zconf.tab.c, [85](#)
- YYNSTATES
 - zconf.tab.c, [85](#)
- YYNTOKENS
 - zconf.tab.c, [85](#)
- YYPACT_NINF
 - zconf.tab.c, [85](#)
- YYPOPSTACK
 - zconf.tab.c, [85](#)
- YYPULL
 - zconf.tab.c, [85](#)
- YYPURE
 - zconf.tab.c, [85](#)
- YYPUSH
 - zconf.tab.c, [85](#)
- YYRECOVERING
 - zconf.tab.c, [85](#)
- YYRHSLOC
 - zconf.tab.c, [85](#)
- YYSIZE_MAXIMUM
 - zconf.tab.c, [85](#)
- YYSIZE_T
 - zconf.tab.c, [85](#)
- YYSKELETON_NAME
 - zconf.tab.c, [85](#)
- YYSTACK_ALLOC
 - zconf.tab.c, [85](#)
- YYSTACK_BYTES
 - zconf.tab.c, [85](#)
- YYSTACK_FREE
 - zconf.tab.c, [86](#)
- YYSTACK_GAP_MAXIMUM
 - zconf.tab.c, [86](#)
- YYSTACK_RELOCATE
 - zconf.tab.c, [86](#)
- YYSTATE
 - zconf.lex.c, [75](#)
- YYSTYPE, [22](#)
 - expr, [22](#)
 - file, [22](#)
 - id, [22](#)
 - menu, [22](#)
 - string, [22](#)
 - symbol, [22](#)
 - zconf.tab.c, [86](#)
- YYSTYPE_IS_DECLARED
 - zconf.tab.c, [86](#)
- YYSTYPE_IS_TRIVIAL
 - zconf.tab.c, [86](#)
- YYTABLE_NINF
 - zconf.tab.c, [86](#)
- YYTABLES_NAME
 - zconf.lex.c, [75](#)
- YYTERROR
 - zconf.tab.c, [86](#)
- YYTOKEN_TABLE
 - zconf.tab.c, [86](#)
- YYTOKENTYPE
 - zconf.tab.c, [86](#)
- YYTRANSLATE
 - zconf.tab.c, [86](#)
- YYUNDEFTOK
 - zconf.tab.c, [86](#)
- YYUSE
 - zconf.tab.c, [86](#)
- yes
 - expr.h, [42](#)
- yesno.c
 - dialog_yesno, [61](#)
- yy_act
 - zconf.lex.c, [79](#)
- yy_at_bol
 - yy_buffer_state, [21](#)
- yy_bp
 - zconf.lex.c, [79](#)
- yy_bs_column
 - yy_buffer_state, [21](#)
- yy_bs_lineno
 - yy_buffer_state, [21](#)
- yy_buf_pos
 - yy_buffer_state, [21](#)

- yy_buf_size
 - yy_buffer_state, 21
- yy_buffer_state, 20
 - yy_at_bol, 21
 - yy_bs_column, 21
 - yy_bs_lineno, 21
 - yy_buf_pos, 21
 - yy_buf_size, 21
 - yy_buffer_status, 21
 - yy_ch_buf, 21
 - yy_fill_buffer, 21
 - yy_input_file, 21
 - yy_is_interactive, 21
 - yy_is_our_buffer, 21
 - yy_n_chars, 21
- yy_buffer_status
 - yy_buffer_state, 21
- yy_ch_buf
 - yy_buffer_state, 21
- yy_cp
 - zconf.lex.c, 79
- yy_create_buffer
 - zconf.lex.c, 71
- yy_delete_buffer
 - zconf.lex.c, 71
- yy_fill_buffer
 - yy_buffer_state, 21
- yy_flex_debug
 - zconf.lex.c, 72
- yy_flush_buffer
 - zconf.lex.c, 72
- yy_init_buffer
 - zconf.lex.c, 72
- yy_input_file
 - yy_buffer_state, 21
- yy_is_interactive
 - yy_buffer_state, 21
- yy_is_our_buffer
 - yy_buffer_state, 21
- yy_load_buffer_state
 - zconf.lex.c, 72
- yy_n_chars
 - yy_buffer_state, 21
- yy_new_buffer
 - zconf.lex.c, 73
- yy_nxt
 - yy_trans_info, 22
- yy_set_bol
 - zconf.lex.c, 73
- yy_set_interactive
 - zconf.lex.c, 73
- yy_size_t
 - zconf.lex.c, 75
- yy_state_type
 - zconf.lex.c, 75
- yy_switch_to_buffer
 - zconf.lex.c, 73
- yy_trans_info, 21
 - yy_nxt, 22
 - yy_verify, 22
- yy_verify
 - yy_trans_info, 22
- yyalloc, 22
 - yyss_alloc, 22
 - yyvs_alloc, 22
 - zconf.lex.c, 74
- yychar
 - zconf.tab.c, 83, 87
- yyclearin
 - zconf.tab.c, 83
- yyconst
 - zconf.lex.c, 74
- yydebug
 - zconf.tab.c, 83, 88
- yyerrok
 - zconf.tab.c, 84
- yyerror
 - zconf.tab.c, 84
- yyfree
 - zconf.lex.c, 74
- yyin
 - zconf.lex.c, 74
- yyleng
 - zconf.lex.c, 74
- yyless
 - zconf.lex.c, 74
- yylex
 - zconf.lex.c, 74
 - zconf.tab.c, 84
- yylineno
 - zconf.lex.c, 74
- yylval
 - zconf.tab.c, 85, 88
- yymore
 - zconf.lex.c, 74
- yynerres
 - zconf.tab.c, 85, 88
- yyout
 - zconf.lex.c, 74
- yyvsparse
 - zconf.tab.c, 85, 87
- yyrealloc
 - zconf.lex.c, 74
- yyrestart
 - zconf.lex.c, 74
- yyss_alloc
 - yyalloc, 22
- yystype
 - zconf.tab.c, 86
- yyterminate
 - zconf.lex.c, 75
- yytext
 - zconf.lex.c, 75
- yytext_ptr
 - zconf.lex.c, 75
- yytokentype

- zconf.tab.c, 86
- yytype_int16
 - zconf.tab.c, 86
- yytype_int8
 - zconf.tab.c, 86
- yytype_uint16
 - zconf.tab.c, 86
- yytype_uint8
 - zconf.tab.c, 86
- yyvs_alloc
 - yyalloc, 22
- yywrap
 - zconf.lex.c, 75
- zconf.tab.c
 - T_AND, 87
 - T_CHOICE, 87
 - T_CLOSE_PAREN, 87
 - T_COMMENT, 87
 - T_CONFIG, 87
 - T_DEFAULT, 87
 - T_DEPENDS, 87
 - T_ENDCHOICE, 87
 - T_ENDIF, 87
 - T_ENDMENU, 86
 - T_EOL, 87
 - T_EQUAL, 87
 - T_HELP, 87
 - T_HELPTEXT, 87
 - T_IF, 87
 - T_MAINMENU, 86
 - T_MENU, 86
 - T_MENUCONFIG, 87
 - T_NOT, 87
 - T_ON, 87
 - T_OPEN_PAREN, 87
 - T_OPTION, 87
 - T_OPTIONAL, 87
 - T_OR, 87
 - T_PROMPT, 87
 - T_RANGE, 87
 - T_SELECT, 87
 - T_SOURCE, 87
 - T_TYPE, 87
 - T_UNEQUAL, 87
 - T_VISIBLE, 87
 - T_WORD, 87
 - T_WORD_QUOTE, 87
- zconf.hash.c
 - kconf_id_strings, 67
- zconf.lex.c
 - BEGIN, 70
 - COMMAND, 70
 - current_buf, 79
 - ECHO, 70
 - FLEX_BETA, 70
 - FLEX_SCANNER, 70
 - FLEXINT_H, 70
 - file, 79
 - flex_int16_t, 75
 - flex_int32_t, 75
 - flex_int8_t, 75
 - flex_uint16_t, 75
 - flex_uint32_t, 75
 - flex_uint8_t, 75
 - HELP, 70
 - i, 79
 - INITIAL, 70
 - INT16_MAX, 70
 - INT16_MIN, 71
 - INT32_MAX, 71
 - INT32_MIN, 71
 - INT8_MAX, 71
 - INT8_MIN, 71
 - if, 75
 - lineno, 79
 - PARAM, 71
 - REJECT, 71
 - START_STRSIZE, 71
 - STRING, 71
 - str, 79
 - ts, 79
 - UINT16_MAX, 71
 - UINT32_MAX, 71
 - UINT8_MAX, 71
 - unput, 71
 - while, 75
 - YY_AT_BOL, 71
 - YY_BREAK, 71
 - YY_BUF_SIZE, 71
 - YY_BUFFER_NEW, 71
 - YY_BUFFER_NORMAL, 71
 - YY_BUFFER_STATE, 75
 - YY_CHAR, 75
 - YY_CURRENT_BUFFER, 71
 - YY_DECL, 71, 79
 - YY_DECL_IS_OURS, 71
 - YY_END_OF_BUFFER, 72
 - YY_EXIT_FAILURE, 72
 - YY_EXTRA_TYPE, 72
 - YY_FATAL_ERROR, 72
 - YY_FLUSH_BUFFER, 72
 - YY_INPUT, 72
 - YY_INT_ALIGNED, 72
 - YY_LESS_LINENO, 72
 - YY_MORE_ADJ, 72
 - YY_NEW_FILE, 73
 - YY_NO_INPUT, 73
 - YY_NULL, 73
 - YY_NUM_RULES, 73
 - YY_READ_BUF_SIZE, 73
 - YY_RULE_SETUP, 73
 - YY_SC_TO_UI, 73
 - YY_SKIP YYWRAP, 73
 - YY_START, 73
 - YY_STATE_BUF_SIZE, 73
 - YY_STATE_EOF, 73

YY_USER_ACTION, 74
 YYSTATE, 75
 YYTABLES_NAME, 75
 yy_act, 79
 yy_bp, 79
 yy_cp, 79
 yy_create_buffer, 71
 yy_delete_buffer, 71
 yy_flex_debug, 72
 yy_flush_buffer, 72
 yy_init_buffer, 72
 yy_load_buffer_state, 72
 yy_new_buffer, 73
 yy_set_bol, 73
 yy_set_interactive, 73
 yy_size_t, 75
 yy_state_type, 75
 yy_switch_to_buffer, 73
 yyalloc, 74
 yyconst, 74
 yyfree, 74
 yyin, 74
 yyleng, 74
 yyless, 74
 yylex, 74
 yylineno, 74
 yymore, 74
 yyout, 74
 yyrealloc, 74
 yyrestart, 74
 yyterminate, 75
 yytext, 75
 yytext_ptr, 75
 yywrap, 75
 zconf_create_buffer, 75
 zconf_curname, 75
 zconf_delete_buffer, 76
 zconf_flex_debug, 79
 zconf_flush_buffer, 76
 zconf_fopen, 76
 zconf_initscan, 76
 zconf_lineno, 76
 zconf_nextfile, 76
 zconf_scan_buffer, 76
 zconf_scan_bytes, 76
 zconf_scan_string, 76
 zconf_starthelp, 77
 zconf_switch_to_buffer, 77
 zconfalloc, 77
 zconffree, 77
 zconfget_debug, 77
 zconfget_extra, 77
 zconfget_in, 77
 zconfget_leng, 77
 zconfget_lineno, 77
 zconfget_out, 77
 zconfget_text, 77
 zconfin, 79
 zconfleng, 79
 zconflex, 77
 zconflex_destroy, 78
 zconflineno, 79
 zconfout, 79
 zconfpop_buffer_state, 78
 zconfpush_buffer_state, 78
 zconfrealloc, 78
 zconfrestart, 78
 zconfset_debug, 78
 zconfset_extra, 78
 zconfset_in, 78
 zconfset_lineno, 78
 zconfset_out, 79
 zconfset_text, 79
 zconfwrap, 75
 zconf.tab.c
 cdebug, 87
 conf_parse, 87
 DEBUG_PARSE, 82
 PATH_MAX, 82
 PRINTD, 82
 printd, 82
 symbol_hash, 87
 YY_, 82
 YY_LOCATION_PRINT, 82
 YY_REDUCE_PRINT, 82
 YY_STACK_PRINT, 82
 YY_SYMBOL_PRINT, 83
 YYABORT, 83
 YYACCEPT, 83
 YYBACKUP, 83
 YYBISON, 83
 YYBISON_VERSION, 83
 YYCOPY, 83
 YYDEBUG, 84
 YYDPRINTF, 84
 YYEMPTY, 84
 YYEOF, 84
 YYERRCODE, 84
 YYERROR, 84
 YYERROR_VERBOSE, 84
 YYFAIL, 84
 YYFINAL, 84
 YYFPRINTF, 84
 YYFREE, 84
 YYID, 84
 YYINITDEPTH, 84
 YYLAST, 84
 YYLEX, 84
 YYLOC_DEFAULT, 84
 YYLSP_NEEDED, 85
 YYMALLOC, 85
 YYMAXDEPTH, 85
 YYMAXUTOK, 85
 YYNNTS, 85
 YYNRULES, 85
 YYNSTATES, 85

YYNTOKENS, [85](#)
YYPACT_NINF, [85](#)
YYPPOPSTACK, [85](#)
YYPULL, [85](#)
YYPURE, [85](#)
YYPUSH, [85](#)
YYRECOVERING, [85](#)
YYRHSLOC, [85](#)
YYSIZE_MAXIMUM, [85](#)
YYSIZE_T, [85](#)
YYSKELETON_NAME, [85](#)
YYSTACK_ALLOC, [85](#)
YYSTACK_BYTES, [85](#)
YYSTACK_FREE, [86](#)
YYSTACK_RELOCATE, [86](#)
YYSTYPE, [86](#)
YYSTYPE_IS_TRIVIAL, [86](#)
YYTABLE_NINF, [86](#)
YYTERROR, [86](#)
YYTOKEN_TABLE, [86](#)
YYTOKENTYPE, [86](#)
YYTRANSLATE, [86](#)
YYUNDEFTOK, [86](#)
YYUSE, [86](#)
yychar, [83](#), [87](#)
yyclearin, [83](#)
yydebug, [83](#), [88](#)
yyerrok, [84](#)
yyerror, [84](#)
yylex, [84](#)
yyval, [85](#), [88](#)
yynerrs, [85](#), [88](#)
yyparse, [85](#), [87](#)
yystype, [86](#)
yytokentype, [86](#)
yytype_int16, [86](#)
yytype_int8, [86](#)
yytype_uint16, [86](#)
yytype_uint8, [86](#)
zconfdump, [87](#)
zconflex, [87](#)
zconf_create_buffer
 [zconf.lex.c](#), [75](#)
zconf_curname
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [75](#)
zconf_delete_buffer
 [zconf.lex.c](#), [76](#)
zconf_flex_debug
 [zconf.lex.c](#), [79](#)
zconf_flush_buffer
 [zconf.lex.c](#), [76](#)
zconf_fopen
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [76](#)
zconf_initscan
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [76](#)
zconf_lineno
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [76](#)
zconf_nextfile
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [76](#)
zconf_scan_buffer
 [zconf.lex.c](#), [76](#)
zconf_scan_bytes
 [zconf.lex.c](#), [76](#)
zconf_scan_string
 [zconf.lex.c](#), [76](#)
zconf_starthelp
 [lkc.h](#), [48](#)
 [zconf.lex.c](#), [77](#)
zconf_switch_to_buffer
 [zconf.lex.c](#), [77](#)
zconfalloc
 [zconf.lex.c](#), [77](#)
zconfdebug
 [lkc.h](#), [48](#)
zconfdump
 [lkc.h](#), [48](#)
 [zconf.tab.c](#), [87](#)
zconffree
 [zconf.lex.c](#), [77](#)
zconfget_debug
 [zconf.lex.c](#), [77](#)
zconfget_extra
 [zconf.lex.c](#), [77](#)
zconfget_in
 [zconf.lex.c](#), [77](#)
zconfget_leng
 [zconf.lex.c](#), [77](#)
zconfget_lineno
 [zconf.lex.c](#), [77](#)
zconfget_out
 [zconf.lex.c](#), [77](#)
zconfget_text
 [zconf.lex.c](#), [77](#)
zconfin
 [zconf.lex.c](#), [79](#)
zconfleng
 [zconf.lex.c](#), [79](#)
zconflex
 [zconf.lex.c](#), [77](#)
 [zconf.tab.c](#), [87](#)
zconflex_destroy
 [zconf.lex.c](#), [78](#)
zconflineno
 [zconf.lex.c](#), [79](#)
zconfout
 [zconf.lex.c](#), [79](#)
zconfparse
 [lkc.h](#), [48](#)
zconfpop_buffer_state
 [zconf.lex.c](#), [78](#)
zconfpush_buffer_state

- zconf.lex.c, [78](#)
- zconfrealloc
 - zconf.lex.c, [78](#)
- zconfrestart
 - zconf.lex.c, [78](#)
- zconfset_debug
 - zconf.lex.c, [78](#)
- zconfset_extra
 - zconf.lex.c, [78](#)
- zconfset_in
 - zconf.lex.c, [78](#)
- zconfset_lineno
 - zconf.lex.c, [78](#)
- zconfset_out
 - zconf.lex.c, [79](#)
- zconftext
 - zconf.lex.c, [79](#)
- zconfwrap
 - zconf.lex.c, [75](#)