MISRA 2004 Compliance

MISRA 2004 Compliance
Version , May 2017

http://www.analog.com

Contents

The following table summarizes the ADuCM302x and ADuCM4x50 driver Misra 2004 suppressions

Driver	File	MISRA Rule	Reason for suppression
Generic	common.h	20.9	The input/output library <stdio.h> shall not be used in production code. The purpose of this header is to provide I/O facilities based on stdio
	adi_drivers_general.h	19.10	In the definition of a function-like macro each parameter shall be enclosed in parenthesis. This is not possible for attributes and pragmas.
		19.13	The # and ## preprocessor operators shall not be used.This is done to abstract the macros for the different toolchains.
Cycle Count	adi_cyclecount.c	14.7	A function should have a single point of exit. Multiple returns are used for error handling.
		9.2	Braces shall be used to indicate and match the structure in the non-zero initialization of arrays and structures.Braces cannot be used to indicate and match the structure in the non-zero initialization of arrays because the total size of the array is defined by a configuration macro
Beeper	adi_beep_config.h	5.1	Identifiers shall not rely on significance of more than 31 characters. IAR compiler supports longer identifiers.
		19.11	identifiers in pre-processor directives should be defined before use The macros in the the following #if directives are defined to enum constants by default
	adi_beep.c	8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.

		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself. Needed for null expansion of ADI_INSTALL_HANDLER and others.
		11.3	A cast should not be performed between a pointer type and an integral type. Required for MMR addresses and callback parameters.
		12.7	Bitwise operations shall not be performed on signed integer types. Rquired for MMR manipulations.
		17.4	Array indexing shall only be applied to objects defined as an array type. Required for adi_beep_PlaySequence() to access the user-supplied array of notes.
		11.4	A cast should not be performed between a pointer to object type and a different pointer to object type, this casts from type. Required to store a an array of varying size in a device structure. Required for adi_beep_PlaySequence() to access the user-supplied array of notes.
Power	adi_pwr_config.h	5.1	Identifiers shall not rely on significance of more than 31 characters. IAR compiler supports longer identifiers.
	adi_pwr.h	5.1	Identifiers shall not rely on significance of more than 31 characters. IAR compiler supports longer identifiers.
		6.3	The basic types of char, int, long, float cannot be used. bool is used in the APIs as it is not affending the rule. Disabling this as IAR treats it as an error
	adi_pwr.c	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it.

		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
ICC	adi_icc_config.h	5.1	Identifiers shall not rely on significance of more than 31 characters. IAR compiler supports longer identifiers.
	adi_icc_cm.c	14.7	A function should have a single point of exit. Multiple returns are used for error handling.
		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		18.5	There shall be no definition of objects in a header file.
		11.3	A cast should not be performed between a pointer type and an integral type. MMR addresses are defined as simple constants. Accessing the MMR requires casting to a pointer type.
		17.4	Array indexing shall only be applied to objects defined as an array type. Relying on pointer arithmetic for buffer handling is needed.
	adi_icc_cm_data.c	11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
		8.5	There shall be no definition of objects of functions in a header file. This rule is suppressed as this is not a header file, but included in a source file.

		15.2	Every non-empty case clause in a switch statement shall be terminated with a break statement. In some cases we have return statement instead of break. It is not valid to both return and break in MISRA 2012.
ADXL363	adi_adxl363.h	6.3	The basic types of char, int, long, float cannot be used. bool is used in the APIs as it is not affending the rule. Disabling this as IAR treats it as an error.
	adi_adxl363.c	11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
GPIO	adi_gpio.h	2.4	sections of code should not be 'commented out'. Allow code example in doxygen comment.
		6.3	The basic types of char, int, long, float cannot be used. bool is used in the APIs as it is not affending the rule. Disabling this as IAR treats it as an error.
	adi_gpio.c	8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.

I2C	adi_i2c.h	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it.
	adi_i2c.c	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it.
		8.5	There shall be no definition of objects or functions in a header file. It is used in the _data.h file which isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
Rng	adi_rng.h	6.3	The basic types of char, int, long, float cannot be used. bool is used in the APIs as it is not affending the rule. Disabling this as IAR treats it as an error
ADC	adi_adc.c	8.5	There shall be no definition of objects or functions in a header file This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself Needed for null expansion of ADI_INSTALL_HANDLER and others

		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		11.3	A cast should not be performed between a pointer type and an integral type The rule makes an exception for memory-mapped register accesses
		17.4	Array indexing shall only be applied to objects defined as an array type. Accessing the DMA descriptors, which are defined in the system as a pointer to an array of descriptors
CRC	adi_crc.c	rule 8.5	there shall be no definition of objects or functions in a header file. This isn't a header as such
		17.4	pointer arithmetic should not be used.
		17.4	Array indexing shall only be applied to objects defined as an array type. Relying on pointer arithmetic for buffer handling.
		11.3	A cast should not be performed between a pointer type and an integral type. Casts from pointer to uint32_t needed to determine pointer alignment
Crypto	adi_crypto.c	8.5	There shall be no definition of objects or functions in a header file.
		11.3	A cast should not be performed between a pointer type and an integral type.
		14.2	A null statement shall only occur on a line by itself.
		17.4	pointer arithmetic should not be used
		14.7	a function should have a single point of exit at the end of the function

		17.4	array indexing shall only be applied to objects defined as an array type
	adi_crypto_def.h	8.5	There shall be no definition of objects or functions in a header file. Exception is to allow the Crypto device data type and instance to be declared simultaniously.
DMA	adi_dma.c	6.3	The basic types of char, int, short, long, float, and double should not be used. Need to use bool.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses
Flash	adi_flash.c	8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself. Needed for null expansion of ADI_INSTALL_HANDLER and others.
		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		11.3	A cast should not be performed between a pointer type and an integral type. Required for MMR accesses, determining pointer alignment, and a callback argument.
		12.4	The right hand operand of an && or || operator shall not contain side effects. Side effects being mis-reported due to added volatile storage class.

RTC	adi_rtc.h	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself. Needed for null expansion of ISR_PROLOG in no-OS case and others.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
	adi_rtc.c	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it.
		8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself. Needed for null expansion of ADI_INSTALL_HANDLER and others.
		20.12	The time handling functions of library <time.h> shall not be used.
		20.2	The names of standard library macros, objects and function shall not be reused. Needed to implement the <time.h> functions here.

		12.7	Bitwise operations shall not be performed on signed integer types.
		12.6	This bitwise operation is in a boolean context - logical operators should not be confused with bitwise operators. The rule is suppressed as the bitwise and logical operators are being used correctly and are not being confused.
		10.2	if the bitwise operators ~ and << are applied to an operand of underlying type 'unsigned char' or 'unsigned short', the result shall be immediately cast to the underlying type of the operand. The behaviour as described is correct
		12.7	bitwise operations shall not be performed on signed integer types. Device drivers often require bit banging on MMRs that are defined as signed
	adi_rtc_def.h	19.10	In the definition of a function-like macro, each instance of a parameter shall be enclosed in parentheses. Parameter use without parentheses needed for struct field name in register access macro.
SPI	adi_spi.c	8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		17.4	Array indexing shall only be applied to objects defined as an array type. Accessing the DMA descriptors, which are defined in the system as a pointer to an array of descriptors.
		17.4	Array indexing shall only be applied to objects of array type.

		18.5	There shall be no definition of objects in a header file.
		14.3	Null statement shall only occur on a line by itself, and shall not have any other text on the same line. Some Macros, such as ISR_PROLOGUE, may not have any expansion resulting in just the terminating ';'
		11.3	A cast should not be performed between a pointer type and an integral type. MMR addresses are defined as simple constants. Accessing the MMR requires casting to a pointer type.
		12.7	Bitwise operations shall not be performed on signed integer types. MMR macros are beyond the control of the driver.
SPI Flash	adi_w25q32.c	6.3	The basic types of char, int, short, long, float, and double should not be used. Necessary for stdbool.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		17.4	Array indexing shall only be applied to objects defined as an array type. Need to use the pointer as an array in the write function for copying user buffer.
SPORT	adi_sport.c	8.5	There shall be no definition of objects or functions in a header file. This isn't a header as such.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		14.2	A null statement shall only occur on a line by itself. Needed for null expansion of ADI_INSTALL_HANDLER and others.

		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		11.3	A cast should not be performed between a pointer type and an integral type. The rule makes an exception for memory-mapped register accesses.
		17.4	Array indexing shall only be applied to objects defined as an array type. Accessing the DMA descriptors, which are defined in the system as a pointer to an array of descriptors.
Timer	adi_tmr.c	6.3	The basic types of char, int, short, long, float, and double should not be used. Necessary for stdbool.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		8.5	There shall be no definition of objects or functions in a header file. Static configuration data file is included.
		11.4	A cast should not be performed between a pointer type and an integral type. This violation appears when deferencing the pointer to the register typedef. No way around this.
		11.4	A cast should not be performed between a pointer to object type and a different pointer to object type. The pointer casting is necessary to allow the GP and RGB timers to abstracted into one driver.
UART	adi_uart.c	14.3	A null statement shall only occur on a line by itself, and shall not have any other text on the same line. Some Macros, such as ISR_PROLOGUE, may not have any expansion resulting in just the terminating ';'.

		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		17.4	Pointer arithmetic should not be used. Relying on pointer arithmetic for buffer handling.
		18.5	There shall be no definition of objects in a header file.
		11.3	A cast should not be performed between a pointer type and an integral type. MMR addresses are defined as simple constants. Accessing the MMR requires casting to a pointer type.
		7.4	Array indexing shall only be applied to objects defined as an array type. Relying on pointer arithmetic for buffer handling and. Accessing the DMA descriptors, which are defined in the system as a pointer to an array of descriptors.
		2.4	Code should not be commented out. This code was commented out to show what the autobaud equations would look like if there were floating point precision. Ideally this would be the case but for the sake of footprint size we will leave it at single point precision.
WDT	adi_wdt.c	6.3	The basic types of char, int, short, long, float, and double should not be used. Necessary for stdbool.
		14.7	A function should have a single point of exit at the end of the function. Multiple returns are used for error handling.
		11.4	A cast should not be performed between a pointer type and an integral type. This violation appears when deferencing the pointer to the register typedef. No way around this.

XINT	adi_xint.c	14.7	A function should have a single point of exit. Multiple returns are used for error handling.
		11.3	A cast should not be performed between a pointer type and an integral types an exception for memory-mapped register accesses.
		10.3	Illegal explicit conversion from underlying MISRA type unsigned int to enum. The typecast is used for efficiency of the code.
		17.4	Array indexing shall only be applied to objects defined as an array. Array indexing is required on the pointer. The memory for gpCallbackTable is passed from application.
Startup /System	startup_<Device>.h	19.10	In the definition of a function-like macro, each instance of a parameter shall be enclosed in parentheses The parameters in the following macros cannot be enclosed in parentheses.
	system_<Device>.h	6.3	Types which specify sign and size should be used. We use bool which is accepted by MISRA but the toolchain does not accept it