Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M4F
Core Size	32-Bit Single-Core
Speed	120MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, IrDA, LINbus, MMC/SD, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	81
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	1.71V ~ 3.63V
Data Converters	A/D 28x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame54n20a-aut

15.7 Register Summary

Offset	Name	Bit Pos.								
0x00	CTRLA	7:0								
0x01	INTENCLR	7:0								CKRDY
0x02	INTENSET	7:0								CKRDY
0x03	INTFLAG	7:0								CKRDY
0x04	HSDIV	7:0	DIV[7:0]							
0x05	CPUDIV	7:0	DIV[7:0]							
0x06 ... 0x0F	Reserved									
0x10	AHBMASK	7:0	Reserved	NVMCTRL	Reserved	DSU	HPBn3	HPBn2	HPBn1	HPBn0
		15:8	SDHCn0	GMAC	QSPI	PAC	Reserved	USB	DMAC	CMCC
		23:16	NVMCTRL_CACHE	NVMCTRL_SMEEPROM	QSPI_2X	PUKCC	ICM	CANn1	CANn0	SDHCn1
		31:24								
0x14	APBAMASK	7:0	GCLK	SUPC	OSC32KCTRL	OSCCTRL	RSTC	MCLK	PM	PAC
		15:8	TCn1	TCn0	SERCOM1	SERCOM0	FREQM	EIC	RTC	WDT
		23:16								
		31:24								
0x18	APBBMASK	7:0	EVSYS			PORT		NVMCTRL	DSU	USB
		15:8		TCn3	TCn2	TCCn1	TCCn0	SERCOM3	SERCOM2	
		23:16								RAMECC
		31:24								
0x1C	APBCMASK	7:0	PDEC	TCn5	TCn4	TCCn3	TCCn2	GMAC		
		15:8		CCL	QSPI		ICM	TRNG	AES	AC
		23:16								
		31:24								
0x20	APBDMASK	7:0	ADCn0	TC7	TC6	TCC4	SERCOM7	SERCOM6	SERCOM5	SERCOM4
		15:8					PCC	I2S	DAC	ADCn1
		23:16								
		31:24								

15.8 Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the 15.5.8 Register Access Protection for details.

20.8.7 Synchronization Busy

Name: SYNCBUSY
Offset: 0x08
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				CLEAR	ALWAYSON	WEN	ENABLE	
Access				R	R	R	R	
Reset				0	0	0	0	

Bit 4 – CLEAR Clear Synchronization Busy

Value	Description
0	Write synchronization of the CLEAR register is complete.
1	Write synchronization of the CLEAR register is ongoing.

Bit 3 – ALWAYSON Always-On Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ALWAYSON bit is complete.
1	Write synchronization of the CTRLA.ALWAYSON bit is ongoing.

Bit 2 – WEN Window Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.WEN bit is complete.
1	Write synchronization of the CTRLA.WEN bit is ongoing.

Bit 1 – ENABLE Enable Synchronization Busy

Value	Description
0	Write synchronization of the CTRLA.ENABLE bit is complete.
1	Write synchronization of the CTRLA.ENABLE bit is ongoing.

Writing a '1' to this bit clears the ECCDE interrupt enable.

This bit will read as the current value of the ECCDE interrupt enable.

Bit 4 – ECCSE ECC Single Error Interrupt Clear
Writing a zero to this bit has no effect.

Writing a '1' to this bit clears the ECCSE interrupt enable.

This bit will read as the current value of the ECCSE interrupt enable.

Bit 3 – LOCKE Lock Error Interrupt Clear
Writing a zero to this bit has no effect.

Writing a '1' to this bit clears the LOCKE interrupt enable.

This bit will read as the current value of the LOCKE interrupt enable.

Bit 2 – PROGE Programming Error Interrupt Clear
Writing a zero to this bit has no effect.

Writing a '1' to this bit clears the PROGE interrupt enable.

This bit will read as the current value of the PROGE interrupt enable.

Bit 1 – ADDRE Address Error Interrupt Clear
Writing a zero to this bit has no effect.

Writing a '1' to this bit clears the ADDRE interrupt enable.

This bit will read as the current value of the ADDRE interrupt enable.

Bit 0 – DONE Command Done Interrupt Clear
Writing a zero to this bit has no effect.

Writing a '1' to this bit clears the DONE interrupt enable.

This bit will read as the current value of the DONE interrupt enable.

Bit 18 – DPLL0TO DPLL0 Lock Timeout
0: DPLL0 Lock time-out not detected.

1: DPLL0 Lock time-out detected.

Bit 17 – DPLL0LCKF DPLL0 Lock Fall
0: DPLL0 Lock fall edge not detected.

1: DPLL0 Lock fall edge detected.

Bit 16 – DPLL0LCKR DPLL0 Lock Rise
0: DPLL0 Lock rise edge not detected.

1: DPLL0 Lock fall edge detected.

Bit 12 – DFLLRCS DFLL Reference Clock Stopped
0: DFLL reference clock is running.

1: DFLL reference clock has stopped.

Bit 11 – DFLLLCKC DFLL Lock Coarse
0: No DFLL coarse lock detected.

1: DFLL coarse lock detected.

Bit 10 – DFLLLCKF DFLL Lock Fine
0: No DFLL fine lock detected.

1: DFLL fine lock detected.

Bit 9 – DFLLOOB DFLL Out Of Bounds
0: No DFLL Out Of Bounds detected.

1: DFLL Out Of Bounds detected.

Bit 8 – DFLLRDY DFLL Ready
0: DFLL is not ready.

1: DFLL is stable and ready to be used as a clock source.

Bit 5 – XOSCCKSW1 XOSC1 Clock Switch
0: XOSC1 is not switched and provides the external clock or crystal oscillator clock.

1: XOSC is switched and provides the safe clock.

Bit 4 – XOSCCKSW0 XOSC0 Clock Switch
0: XOSC0 is not switched and provides the external clock or crystal oscillator clock.

1: XOSC0 is switched and provides the safe clock.

Bit 3 – XOSCFAIL1 XOSC1 Clock Failure
0: XOSC1 failure not detected.

1: XOSC1 failure detected.

31.7.11 Channel n Interrupt Flag Status and Clear

Name: CHINTFLAG
Offset: 0x26 + n*0x08 [n=0..31]
Reset: 0x00

Bit	7	6	5	4	3	2	1	0
							EVD	OVR
Access							RW	RW
Reset							0	0

Bit 1 – EVD Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.

When the event channel path is asynchronous, the EVD interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

Bit 0 – OVR Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.

There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.

Writing a '0' to this bit has no effect.

Writing a '1' to this bit will clear the Overrun Channel interrupt flag.

Refer to 38.6.2 USB Device Operations for the basic operation of the device mode.

Refer to 38.6.3 Host Operations for the basic operation of the host mode.

38.6.2 USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

38.6.2.1 Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

38.6.2.2 Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB master (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer also to the Endpoint Descriptor structure in 38.8.4.1 Endpoint Descriptor Structure.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

38.6.2.3 Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.

38.8.2.4 Device Frame Number

Name: FNUM
Offset: 0x10
Reset: 0x0000
Property: Read only

Bit	15	14	13	12	11	10	9	8
	FNCERR		FNUM[10:5]					
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Reset	0		0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FNUM[4:0]					MFNUM[2:0]		
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 15 – FNCERR Frame Number CRC Error
This bit is cleared upon receiving a USB reset.

This bit is set when a corrupted frame number (or micro-frame number) is received.

This bit and the SOF (or MSOF) interrupt bit are updated at the same time.

Bits 13:3 – FNUM[10:0] Frame Number
These bits are cleared upon receiving a USB reset.

These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.

Bits 2:0 – MFNUM[2:0] Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).

These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.

Value	Description
0	The bank0 is the bank that will be used in the next single/multi USB packet.
1	The bank1 is the bank that will be used in the next single/multi USB packet.

Bit 1 – DTGLIN Data Toggle IN Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

Value	Description
0	The PID of the next expected IN transaction will be zero: data 0.
1	The PID of the next expected IN transaction will be one: data 1.

Bit 0 – DTGLOUT Data Toggle OUT Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.

Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

Value	Description
0	The PID of the next expected OUT transaction will be zero: data 0.
1	The PID of the next expected OUR transaction will be one: data 1.

39.6.5.1 Acceptance Filtering

The CAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration GFC
- Standard ID Filter Configuration SIDFC
- Extended ID Filter Configuration XIDFC
- Extended ID AND Mask XIDAM

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

Rx Buffer
New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see PSR.LEC respectively PSR.FLEC.

Rx FIFO
Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR.LEC respectively PSR.FLEC. In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

Note: When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

39.8.1 Core Release

Name: CREL
Offset: 0x00
Reset: 0x32100000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	REL[3:0]				STEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	0	0	1	1	0	0	1	0

Bit	23	22	21	20	19	18	17	16
	SUBSTEP[3:0]							
Access	R	R	R	R				
Reset	0	0	0	1				

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
Access								
Reset								

Bits 31:28 – REL[3:0] Core Release
One digit, BCD-coded.

Bits 27:24 – STEP[3:0] Step of Core Release
One digit, BCD-coded.

Bits 23:20 – SUBSTEP[3:0] Sub-step of Core Release
One digit, BCD-coded.

39.8.11 Timeout Counter Configuration

Name: TOCC
Offset: 0x28
Reset: 0xFFFF0000
Property: Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Bit	31	30	29	28	27	26	25	24
	TOP[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	23	22	21	20	19	18	17	16
	TOP[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						TOS[1:0]		ETOC
Access						R/W	R/W	R/W
Reset						0	0	0

Bits 31:16 – TOP[15:0] Timeout Period
Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

Bits 2:1 – TOS[1:0] Timeout Select
When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored.

Value	Name	Description
0x0	CONT	Continuous operation.
0x1	TXEF	Timeout controlled by TX Event FIFO.
0x2	RXF0	Timeout controlled by Rx FIFO 0.
0x3	RXF1	Timeout controlled by Rx FIFO 1.

Bit 0 – ETOC Enable Timeout Counter

Value	Description
0	Timeout Counter disabled.
1	Timeout Counter enabled.

42.5.7 Debug Operation

When the CPU is halted in debug mode, the AES module continues normal operation. If the AES module is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging. The AES module can be forced to halt operation during debugging.

42.5.8 Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the

following register:

- Interrupt Flag Register (INTFLAG)

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to *PAC - Peripheral Access Controller* chapter for details.

Related Links

27. PAC - Peripheral Access Controller

42.5.9 Analog Connections

Not applicable.

42.6 Functional Description

42.6.1 Principle of Operation

The following is a high level description of the algorithm. These are the steps:

- KeyExpansion: Round keys are derived from the cipher key using Rijndael's key schedule.
- InitialRound:
  - AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.
- Rounds:
  - SubBytes: A non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows: A transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - AddRoundKey
- Final Round (no MixColumns):
  - SubBytes
  - ShiftRows
  - AddRoundKey

The relationship between the module's clock frequency and throughput (in bytes per second) is given by:

Clock Frequency = (Throughput/2) x (Nr+1) for 2 byte parallel processing

Clock Frequency = (Throughput/4) x (Nr+1) for 4 byte parallel processing

Important: If the condition is verified, the length of R must be greater or equal to the length of X.

43.3.4.6.4 Parameters Definition

This service can easily be accessed through the use of the `PUKCL_CondCopy()` and `PUKCL()` macros.

Table 43-15. CondCopy Service Parameters

Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
u2Options	u2	I	–	–	Option for condition (see the following table)	Option for condition (see the following table)
Specific/ CarryIn	Bit	I	–	–	Bit CarryIn	Bit CarryIn
nu1XBase	nu1	I	Crypto RAM	u2XLength	Base of X	Base of X number untouched
nu1RBase	nu1	I	Crypto RAM	u2RLength	Base of R	Base of R filled with X if condition holds
u2RLength	u2	I	–	–	Length of R	Length of R
u2XLength	u2	I	–	–	Length of X	Length of X

43.3.4.6.5 Available Options

The option for the condition is set by the u2Options input parameter that must take one of the values listed in the following table.

Table 43-16. CondCopy Service Options

Option	Purpose	Needed parameters
PUKCL_CONDCOPY_ALWAYS	Always perform the copy	nu1XBase,u2XLength,nu1RBase, u2RLength
PUKCL_CONDCOPY_NEVER	Never perform the copy	None
PUKCL_CONDCOPY_IF_CARRY	Perform the copy if CarryIn is 1	Specific/CarryIn nu1XBase,u2XLength,nu1RBase, u2RLength
PUKCL_CONDCOPY_IF_NOT_CARRY	Perform the copy if CarryIn is zero	Specific/CarryIn nu1XBase,u2XLength,nu1RBase, u2RLength

43.3.4.6.6 Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// CarryIn shall be beforehand filled (with zero or one) PUKCL(Specific).CarryIn = ...;
```

Parameter	Type	Direction	Location	Data Length	Before Executing the Service	After Executing the Service
nu1KBase	nu1	I	Crypto RAM	u2KLength	Scalar number used to multiply the point A	Unchanged
u2KLength	u2	I	–	–	Length of scalar K	Length of scalar K
nu1PointABase	nu1	I/O	Crypto RAM	3*u2ModLength + 12	Input point A (projective coordinates)	Resulting point C (projective coordinates)
nu1ABas	nu1	I	Crypto RAM	u2ModLength + 4	Parameter a of the elliptic curve	Unchanged
nu1Workspace	nu1	I	Crypto RAM	8*u2ModLength + 44	–	Corrupted workspace

43.3.6.5.5 Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;

PUKCL _ZpEccMul(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccMul(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccMul(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccMul(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccMul(nu1ABase) = <Base of the ram location of the parameter A of the elliptic
curve>;
PUKCL _ZpEccMul(nu1KBase) = <Base of the ram location of the scalar number>;
PUKCL _ZpEccMul(nu1Workspace) = <Base of the ram location of the workspace>;
PUKCL_ZpEccMul(u2KLength) = <Byte length of the Scalar Number K>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccMulFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

43.3.6.5.6 Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase,nu1CnsBase, nu1PointABase, nu1ABase, nu1ScalarNumber, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1ABase, u2ModLength + 4}, {nu1ScalarNumber, u2ScalarLength} or {nu1Workspace, 8*u2ModLength + 44} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length

48.7.1.10 Driver Control

Name: DRVCTRL
Offset: 0x0D
Reset: 0x00
Property: PAC Write-Protection, Enable-Protected

Bit	7	6	5	4	3	2	1	0
								INVENx
Access								R/W
Reset								0

Bit 0 – INVENx Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

Value	Description
0	Disable inversion of the WO[x] output and IO input pin.
1	Enable inversion of the WO[x] output and IO input pin.

48.7.2.14 Channel x Compare/Capture Value, 16-bit Mode

Name: CCx
Offset: 0x1C + x*0x02 [x=0..1]
Reset: 0x0000
Property: Write-Synchronized

Bit	15	14	13	12	11	10	9	8
	CC[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – CC[15:0] Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.

Writing a '1' to this bit will clear the Receive Ready x interrupt flag.

Figure 52-12. PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1)

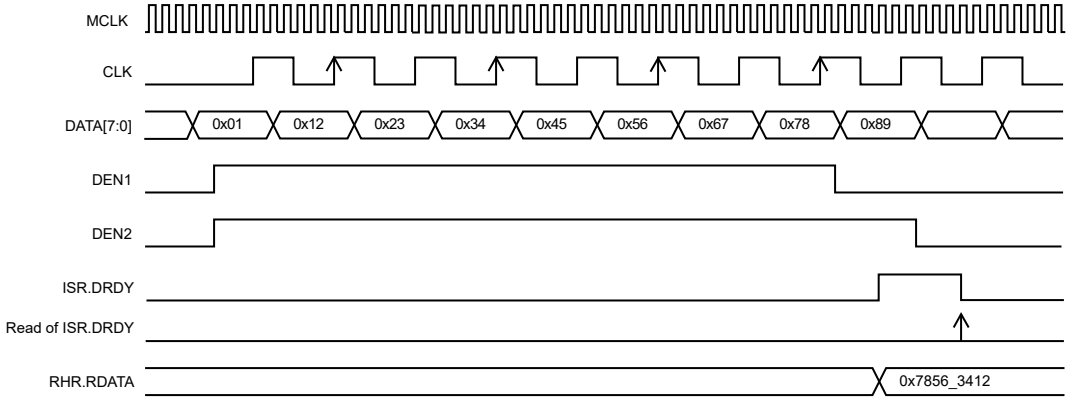


Figure 52-13. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 0)

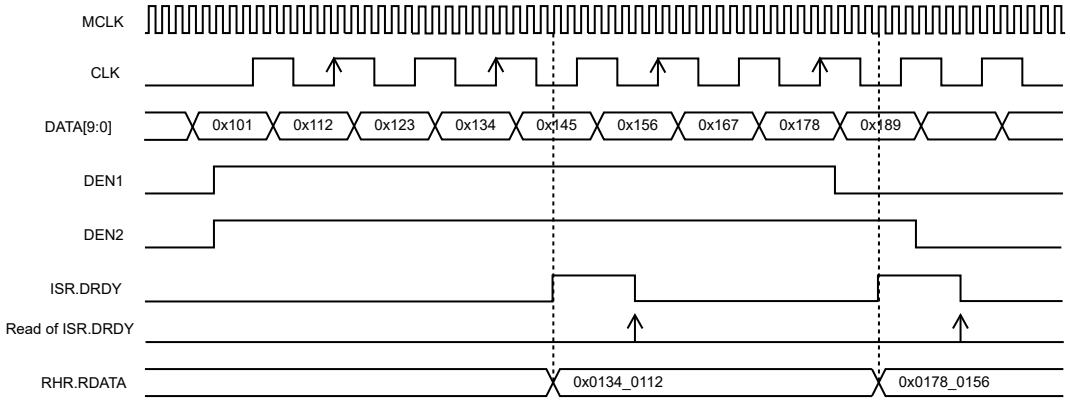
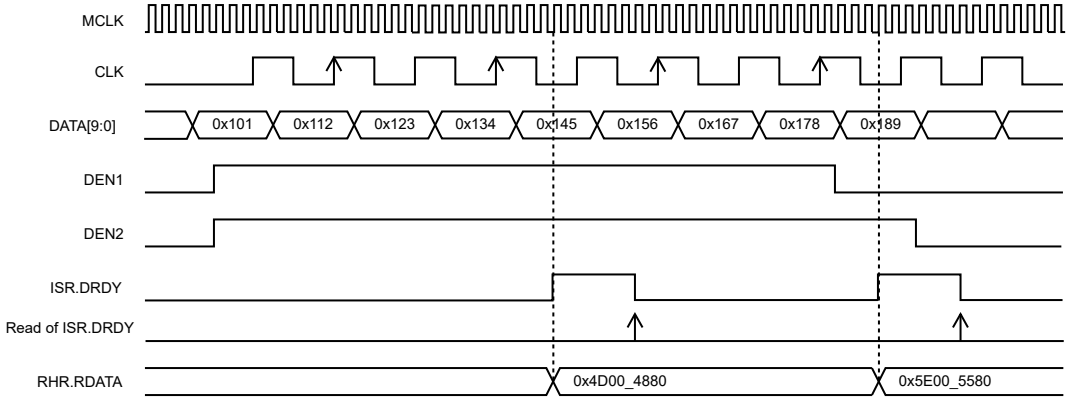


Figure 52-14. PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 1)



52.6.2 Register Access Protection

The configuration bit fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in the Mode Register (MR) can be changed ONLY if the PCC is disabled at this time (MR.PCEN=0).

Table 55-1. Thermal Resistance Data

Package Type	θ_{JA}	θ_{JC}
64-pin TQFP	57.4°C/W	10.6°C/W
100-pin TQFP	55.0°C/W	11.1°C/W
128-pin TQFP	48.7°C/W	9.4°C/W
120-pin TFBGA	36.63°C/W	12.2°C/W
48-pin QFN	29.8°C/W	10.0°C/W
64-pin QFN	30.3°C/W	9.9°C/W
64-pin WLCSP	36.8°C/W	5.0°C/W

55.2.2 Junction Temperature

The average chip-junction temperature, T_J, in °C can be obtained from the following:

1. $T_J = T_A + (P_D \times \theta_{JA})$
2. $T_J = T_A + (P_D \times (\theta_{HEATSINK} + \theta_{JC}))$

where:

- θ_{JA} = Package thermal resistance, Junction-to-ambient (°C/W), see Thermal Resistance Data
- θ_{JC} = Package thermal resistance, Junction-to-case thermal resistance (°C/W), see Thermal Resistance Data
- $\theta_{HEATSINK}$ = Thermal resistance (°C/W) specification of the external cooling device
- P_D = Device power consumption (W)
- T_A = Ambient temperature (°C)

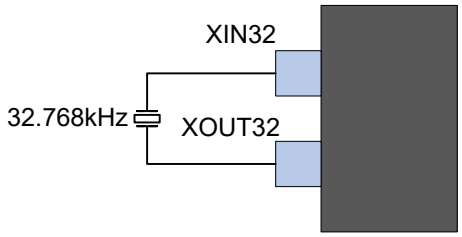
From the first equation, the user can derive the estimated lifetime of the chip and decide if a cooling device is necessary or not. If a cooling device has to be fitted on the chip, the second equation should be used to compute the resulting average chip-junction temperature T_J in °C.

55.3 Package Drawings

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging.

The typical parasitic load capacitance values are available in the Electrical Characteristics section. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in the next figure.

Figure 56-10. External Real Time Oscillator without Load Capacitor



To improve accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown the figure below.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

Figure 56-11. External Real Time Oscillator with Load Capacitor

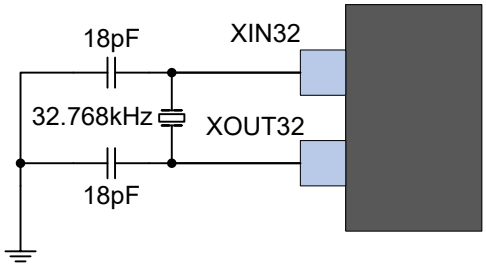


Table 56-6. External Real Time Oscillator Checklist

Signal Name	Recommended Pin Connection	Description
XIN32	Load capacitor 18pF(1)(2)	Timer oscillator input
XOUT32	Load capacitor 18pF(1)(2)	Timer oscillator output

1. These values are only given as typical examples.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

Note: In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.

56.6.4 Calculating the Correct Crystal Decoupling Capacitor

The model shown in Figure 56-12 can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors C_{Ln}, external parasitic capacitance C_{ELn} and external load capacitance C_{Pn}.