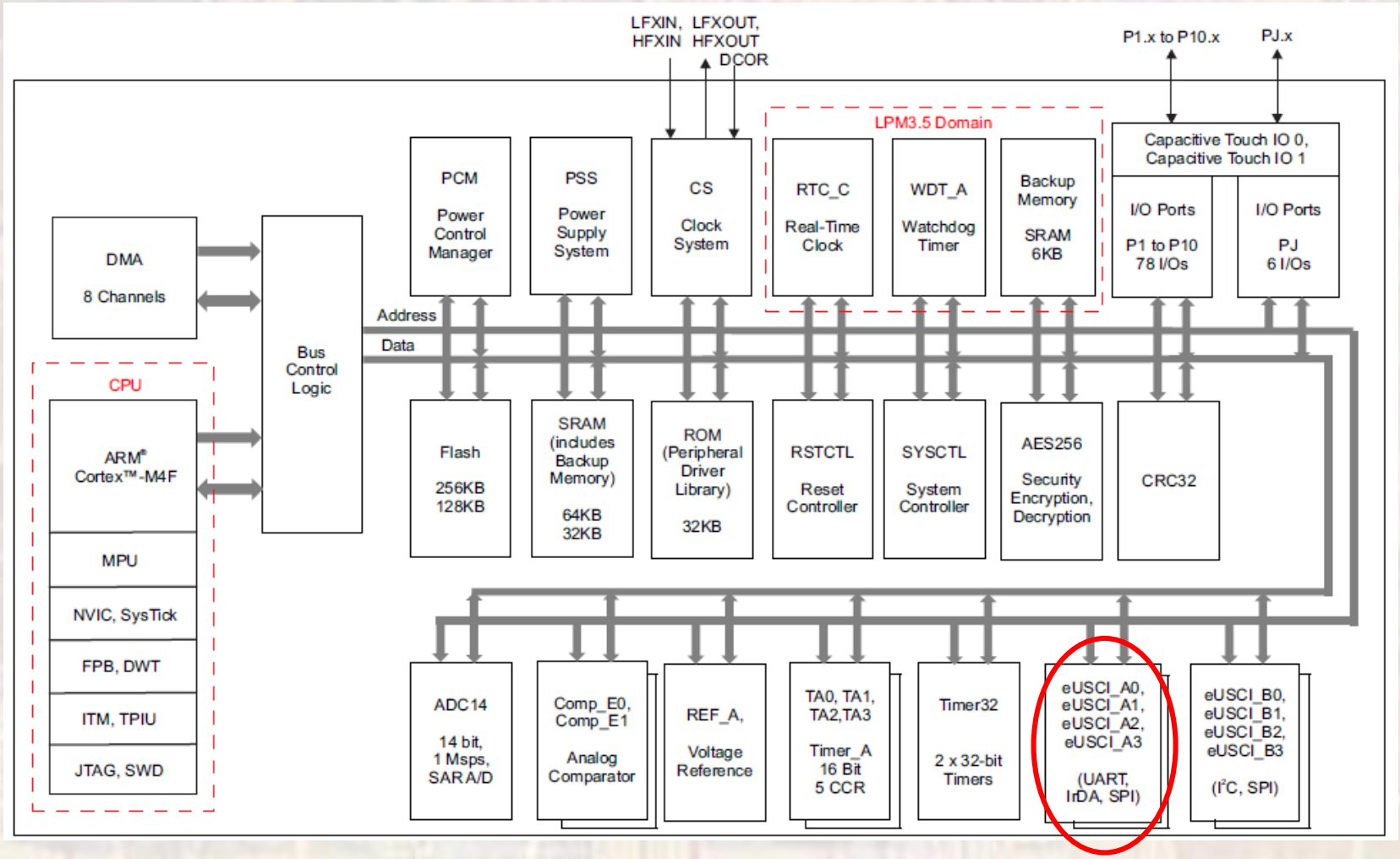
Universal Asynchronous Receiver / Transmitter (UART)

Last updated 6/17/19

UART

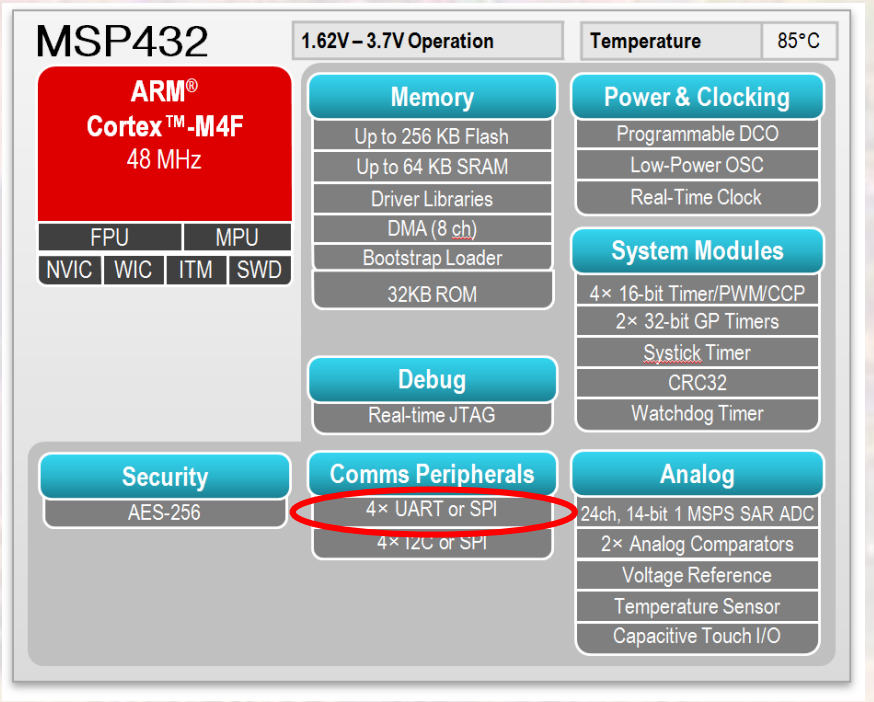
- MSP432 UART



© tj

UART

- MSP432 UART
  - ARM (AMBA Compliant)
  - Asynchronous operation
  - 7/8 bit transmission
  - Master/Slave
  - LSB/MSB first
  - Separate RX/TX registers
  - 4 eUSCI_A modules



 © tj

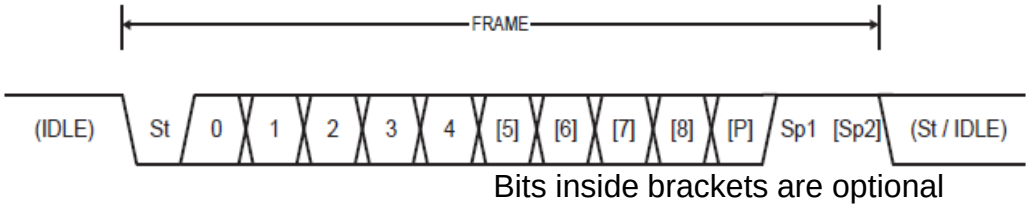
UART

- UART/USART
  - Serial receiver / transmitter
  - Asynchronous and Synchronous versions
  - Asynchronous
    - 2 pin interface
    - RxD – receive data pin
    - TxD – transmit data pin
    - Clock recovery system
  - Synchronous
    - 3 pin interface
    - RxD – receive data pin
    - TxD – transmit data pin
    - Xck – Clock – PD4

© tj

UART

- Frame
  - 1 start bit
  - 5, 6, 7, 8, or 9 data bits – typically LSB first
  - none, even, or odd - parity bit
  - 1 or 2 stop bits,
  - Overflow, Framing, Parity – error detection



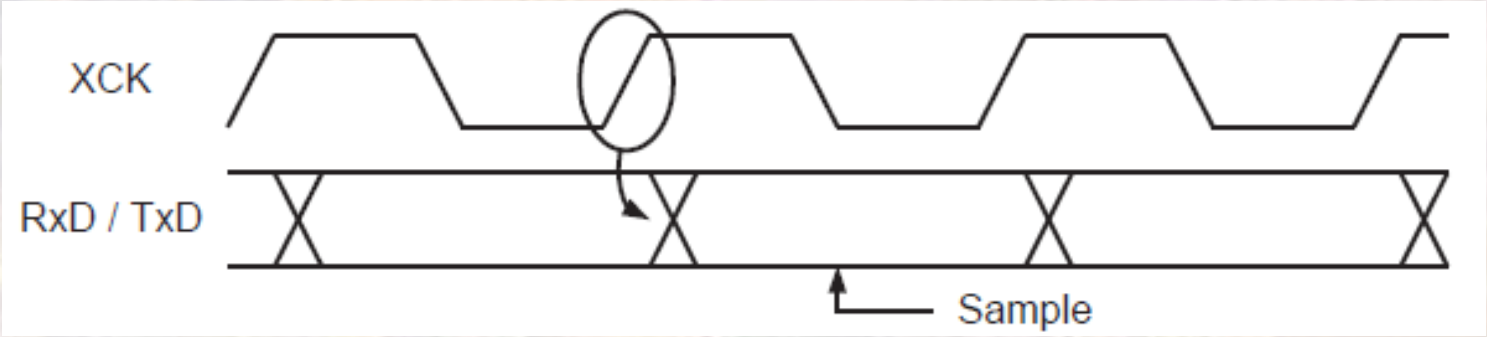
Bits inside brackets are optional

St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

© tj

USART

- Synchronous mode operation
  - Master creates clock signal
    - Xck max is typically $Clk_{system}/4$ for timing purposes
  - Master and Slave
    - Transmit on one clock edge
    - Receive (latch data) on the opposite clock edge



 © tj

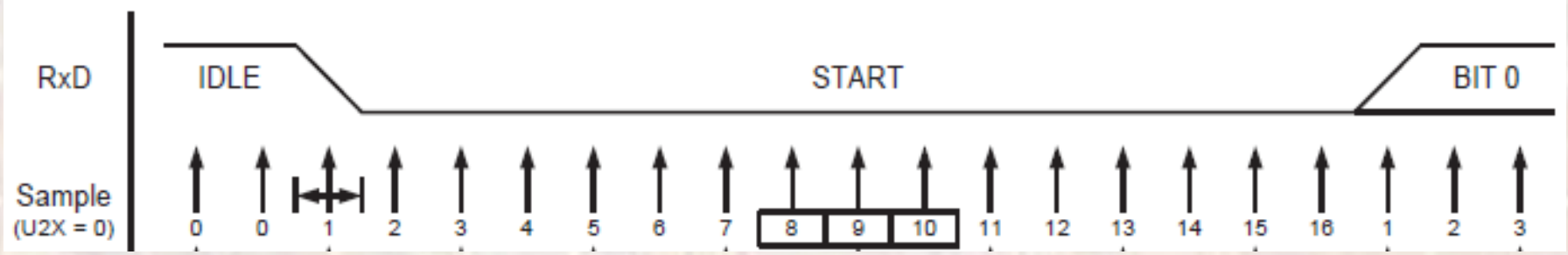
UART

- Asynchronous mode operation
  - Transmit is unchanged
  - Xck is disabled
    - No Master or Slave
  - Must establish an agreed BAUD rate
    - Max BAUD rate is typically $Clk_{system}/16$
    - Limited by HW – clock selection options
    - Fixed in SW
    - Start at known BAUD rate and agree to go faster/slower
  - Receiver circuitry includes:
    - Clock recovery block
    - Data recovery block

© tj

UART

- Asynchronous mode operation
  - Clock and Data recovery
    - Internal clock at 16x BAUD rate
    - Sample RxD signal with internal clock
    - Detect Start bit falling edge
    - Sample RxD after 8,9,10 internal clocks
    - Majority determines bit value
    - Assuming a valid start – all further bits sampled at multiples of 16 clocks



UART

- Multi-processor mode
  - Multiple processors sharing the same USART signals
  - Use the last bit in the data (e.g. bit 9 when using 8 bit data) to indicate an address or data value is in the frame
  - If it is an address and it is your address, collect subsequent data frames
  - If it is not your address, ignore subsequent data frames

© tj

UART

- Error detection
  - Parity
    - Create an error if parity is wrong
  - Overflow
    - Create an error if new data is ready to be sampled and the last data has not been read from the buffer yet
    - Double buffering is common
  - Frame error
    - Stop bit not detected when expected
    - Idle not detected when expected

© tj

UART

- Common BAUD rates

BAUD Rate	Bit Width (us)	Frame Length start,8bit data, parity, 1 stop (us)	Data Rate (Bits/s)	(Bytes/s)
4800	208.33	2291.67	3,491	436
9600	104.17	1145.83	6,982	873
19200	52.08	572.92	13,964	1,745
38400	26.04	286.46	27,927	3,491
57600	17.36	190.97	41,891	5,236
115200	8.68	95.49	83,782	10,473

© tj

UART

- MSP432 UART Implementation
  - Clock Selector
  - Prescaler /baud rate generator
  - RX/TX buffers
  - RX//TX state machines
  - Error detector
  - Automatic baud rate detection

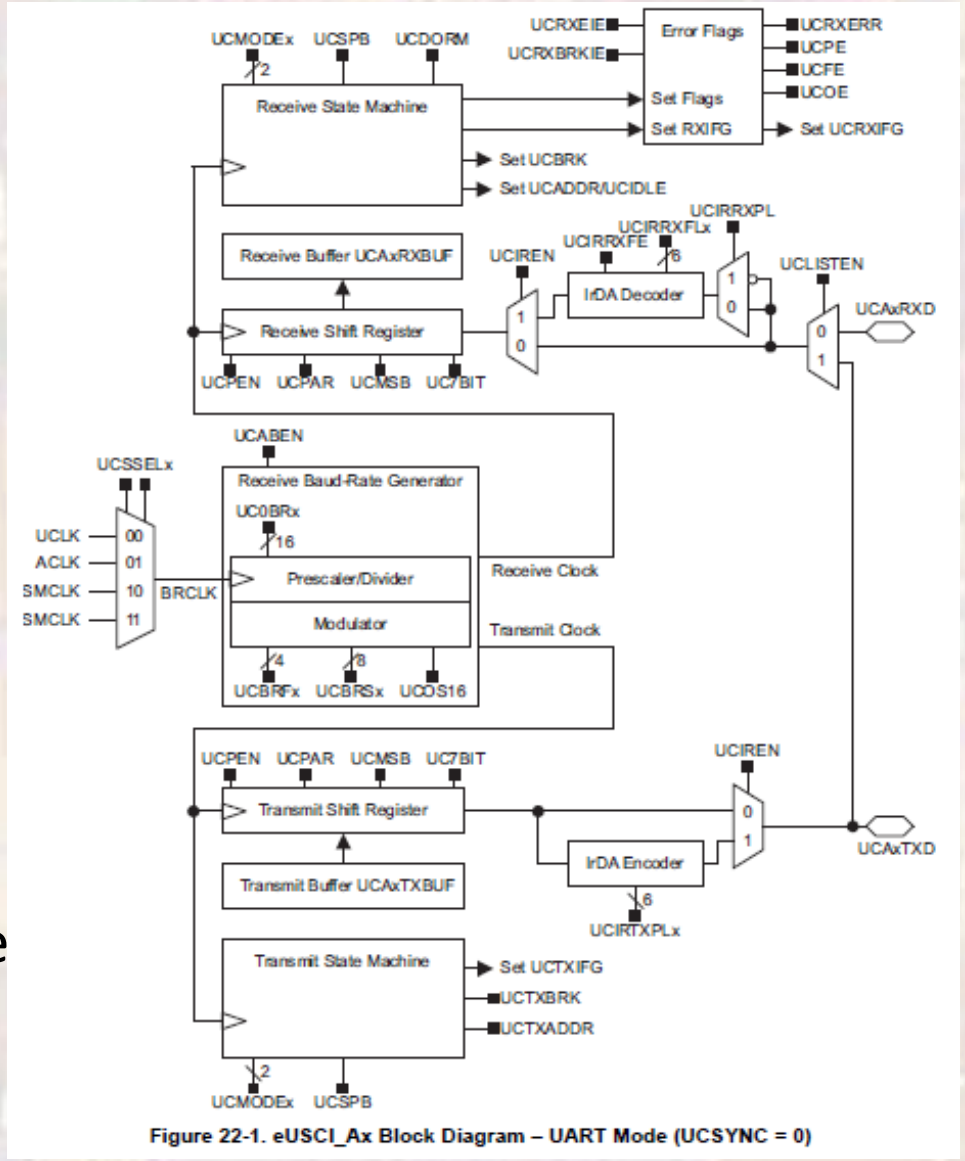


Figure 22-1. eUSCI_Ax Block Diagram – UART Mode (UCSYNC = 0)

 © tj

UART

- MSP432 UART Implementation
  - Required bits
    - start
    - Data 0-7
    - Stop
  - Optional bits
    - data bit 8
    - address bit (7 bit mode)
    - parity bit
    - 2nd stop bit

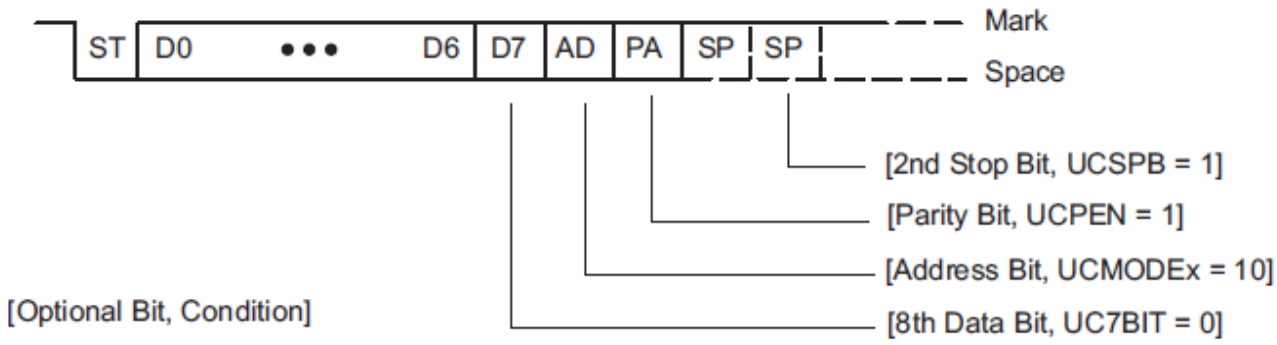


Figure 22-2. Character Format

© tj

UART

- MSP432 UART Implementation
  - Idle Line Microprocessor Format
    - Used when more than 2 devices share the signal paths
    - Blocks of data separated by at least 10 sequential 1's after a stop
    - The first block of data following an idle time is an address

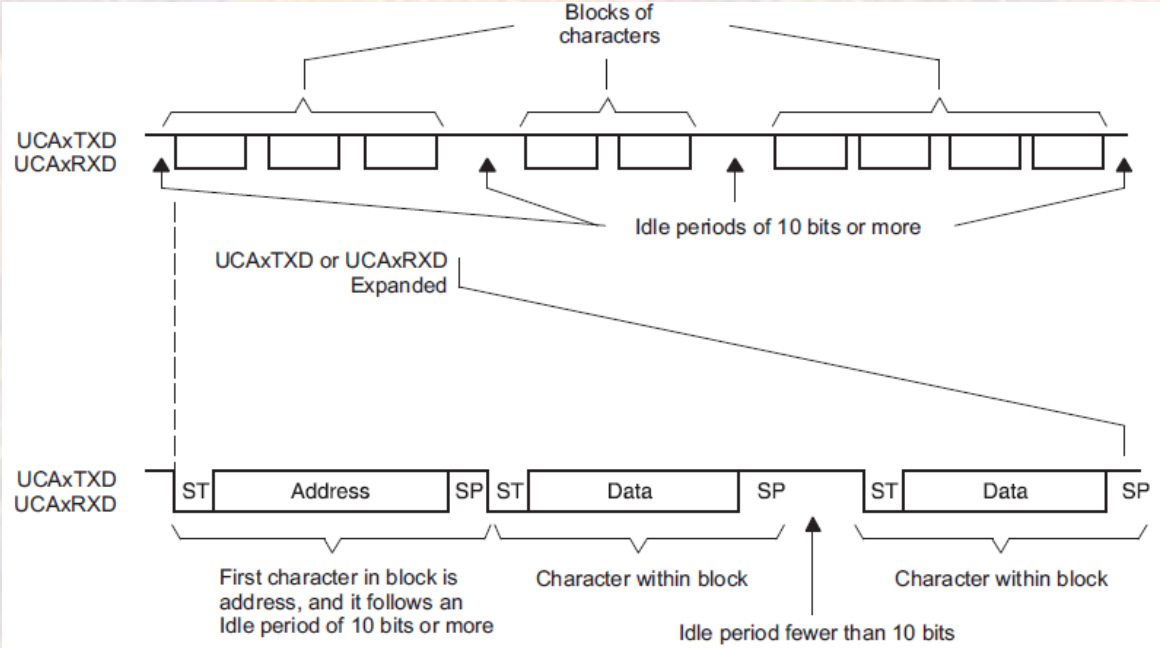


Figure 22-3. Idle-Line Format

© tj

UART

- MSP432 UART Implementation
  - Address Bit Microprocessor Format
    - Used when more than 2 devices share the signal paths
    - The first block of data after a start contains an extra bit to indicate it is an address

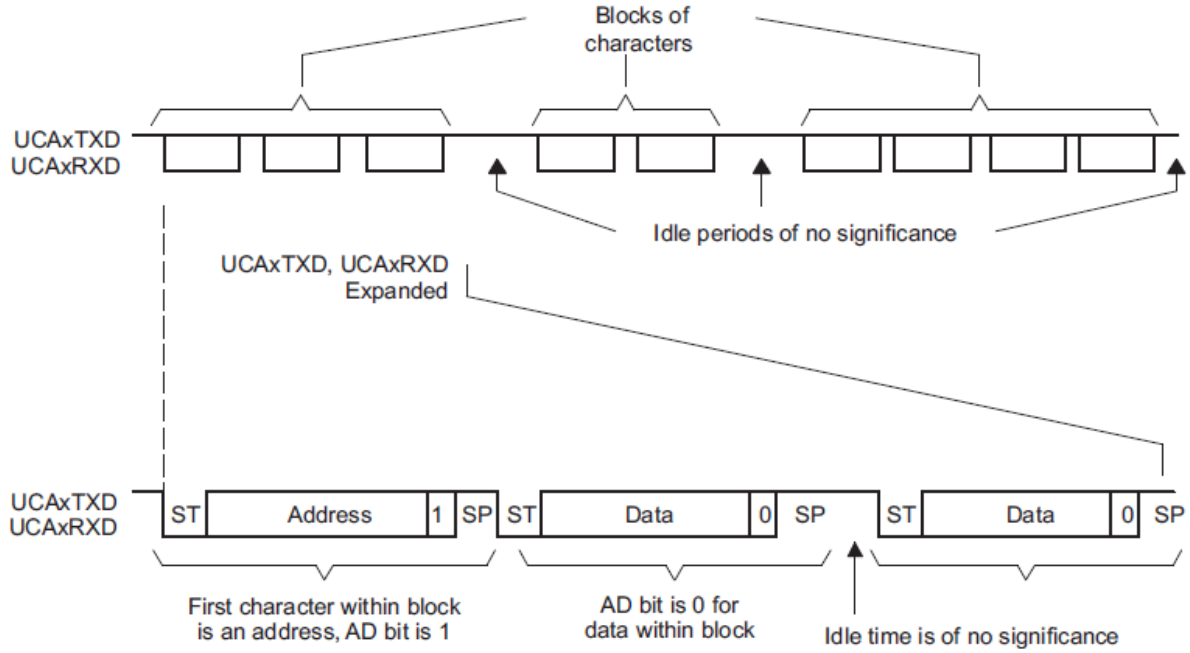


Figure 22-4. Address-Bit Multiprocessor Format

© tj

UART

- MSP432 UART Implementation
  - Automatic baud rate detection
    - Break -> delimiter -> sync pulses
    - Baud rate is calculated based on the sync pulses
      - Max = 1MBaud
      - Min – normal mode = 244 Baud
      - Min – low frequency mode = 15Baud

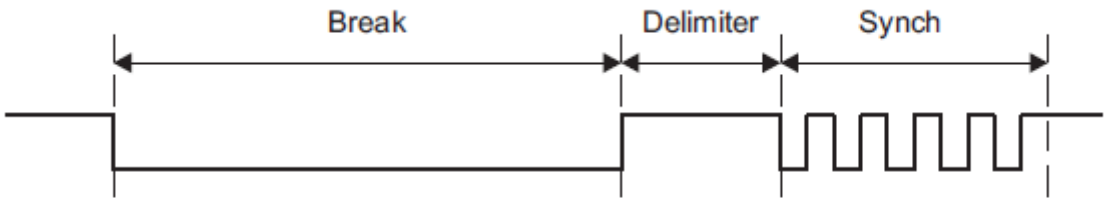


Figure 22-5. Auto Baud-Rate Detection – Break/Synch Sequence

© tj

UART

- MSP432 UART Implementation
  - Receive Error Detection

Table 22-1. Receive Error Conditions

Error Condition	Error Flag	Description
Framing error	UCFE	A framing error occurs when a low stop bit is detected. When two stop bits are used, both stop bits are checked for framing error. When a framing error is detected, the UCFE bit is set.
Parity error	UCPE	A parity error is a mismatch between the number of 1s in a character and the value of the parity bit. When an address bit is included in the character, it is included in the parity calculation. When a parity error is detected, the UCPE bit is set.
Receive overrun	UCOE	An overrun error occurs when a character is loaded into UCAxRXBUF before the prior character has been read. When an overrun occurs, the UCOE bit is set.
Break condition	UCBRK	When not using automatic baud-rate detection, a break is detected when all data, parity, and stop bits are low. When a break condition is detected, the UCBRK bit is set. A break condition can also set the interrupt flag UCRXIFG if the break interrupt enable UCBRKIE bit is set.

© tj

UART

- MSP432 UART Implementation
  - Clock setup for desired Baud Rate

Baud-rate settings quick set up

To calculate the correct settings for the baud-rate generation, perform these steps:

1. Calculate N = f_{BRCLK} / baud rate [if N > 16 continue with step 3, otherwise with step 2]
2. OS16 = 0, UCBRx = INT(N) [continue with step 4]
3. OS16 = 1, UCBRx = INT(N/16), UCBRFx = INT([(N/16) – INT(N/16)] × 16)
4. UCBRSx can be found by looking up the fractional part of N (= N - INT(N)) in table Table 22-4
5. If OS16 = 0 was chosen, a detailed error calculation is recommended to be performed

Table 22-4. UCBRSx Settings for Fractional Portion of N = f_{BRCLK}/Baud Rate

Fractional Portion of N	UCBRSx[1]		Fractional Portion of N	UCBRSx[1]
0.0000	0x00		0.5002	0xAA
0.0529	0x01		0.5715	0x6B
0.0715	0x02		0.6003	0xAD
0.0835	0x04		0.6254	0xB5

© tj

UART

- MSP432 UART Registers

EUSCI_Ax->CTLW0 …
x = 0-3

Table 22-7. eUSCI_A UART Registers

Offset	Acronym	Register Name	Section
00h	UCAxCTLW0	eUSCI_Ax Control Word 0	Section 22.4.1
00h	UCAxCTL1	eUSCI_Ax Control 1	
01h	UCAxCTL0	eUSCI_Ax Control 0	
02h	UCAxCTLW1	eUSCI_Ax Control Word 1	Section 22.4.2
06h	UCAxBRW	eUSCI_Ax Baud Rate Control Word	Section 22.4.3
06h	UCAxBR0	eUSCI_Ax Baud Rate Control 0	
07h	UCAxBR1	eUSCI_Ax Baud Rate Control 1	
08h	UCAxMCTLW	eUSCI_Ax Modulation Control Word	Section 22.4.4
0Ah	UCAxSTATW	eUSCI_Ax Status	Section 22.4.5
0Ch	UCAxRXBUF	eUSCI_Ax Receive Buffer	Section 22.4.6
0Eh	UCAxTXBUF	eUSCI_Ax Transmit Buffer	Section 22.4.7
10h	UCAxABCTL	eUSCI_Ax Auto Baud Rate Control	Section 22.4.8
12h	UCAxIRCTL	eUSCI_Ax IrDA Control	Section 22.4.9
12h	UCAxIRTCTL	eUSCI_Ax IrDA Transmit Control	
13h	UCAxIRRCTL	eUSCI_Ax IrDA Receive Control	
1Ah	UCAxIE	eUSCI_Ax Interrupt Enable	Section 22.4.10
1Ch	UCAxIFG	eUSCI_Ax Interrupt Flag	Section 22.4.11
1Eh	UCAxIV	eUSCI_Ax Interrupt Vector	Section 22.4.12

UART

- MSP432 UART Control Word 0 Register

Figure 22-12. UCAxCTLW0 Register

15	14	13	12	11	10	9	8
UCPEN	UCPAR	UCMSB	UC7BIT	UCSPB	UCMODEx		UCSYNC
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
UCSSELx		UCRXEIE	UCBRKIE	UCDORM	UCTXADDR	UCTXBRK	UCSWRST
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1
Modify only when UCSWRST = 1							

Table 22-8. UCAxCTLW0 Register Description

Bit	Field	Type	Reset	Description
15	UCPEN	RW	0h	Parity enable 0b = Parity disabled 1b = Parity enabled. Parity bit is generated (UCAxTXD) and expected (UCAxRXD). In address-bit multiprocessor mode, the address bit is included in the parity calculation.
14	UCPAR	RW	0h	Parity select. UCPAR is not used when parity is disabled. 0b = Odd parity 1b = Even parity
13	UCMSB	RW	0h	MSB first select. Controls the direction of the receive and transmit shift register. 0b = LSB first 1b = MSB first
12	UC7BIT	RW	0h	Character length. Selects 7-bit or 8-bit character length. 0b = 8-bit data 1b = 7-bit data

© tj

UART

- MSP432 UART Control Word 0 Register

Bit	Field	Type	Reset	Description
11	UCSPB	RW	0h	Stop bit select. Number of stop bits. 0b = One stop bit 1b = Two stop bits
10-9	UCMODEx	RW	0h	eUSCI_A mode. The UCMODEx bits select the asynchronous mode when UCSYNC = 0. 00b = UART mode 01b = Idle-line multiprocessor mode 10b = Address-bit multiprocessor mode 11b = UART mode with automatic baud-rate detection
8	UCSYNC	RW	0h	Synchronous mode enable 0b = Asynchronous mode 1b = Synchronous mode
7-6	UCSSELx	RW	0h	eUSCI_A clock source select. These bits select the BRCLK source clock. 00b = UCLK 01b = ACLK 10b = SMCLK 11b = SMCLK
5	UCRXEIE	RW	0h	Receive erroneous-character interrupt enable 0b = Erroneous characters rejected and UCRXIFG is not set. 1b = Erroneous characters received set UCRXIFG.
4	UCBRKIE	RW	0h	Receive break character interrupt enable 0b = Received break characters do not set UCRXIFG. 1b = Received break characters set UCRXIFG.

© tj

UART

- MSP432 UART Control Word 0 Register

Bit	Field	Type	Reset	Description
3	UCDORM	RW	0h	Dormant. Puts eUSCI_A into sleep mode. 0b = Not dormant. All received characters set UCRXIFG. 1b = Dormant. Only characters that are preceded by an idle-line or with address bit set UCRXIFG. In UART mode with automatic baud-rate detection, only the combination of a break and synch field sets UCRXIFG.
2	UCTXADDR	RW	0h	Transmit address. Next frame to be transmitted is marked as address, depending on the selected multiprocessor mode. 0b = Next frame transmitted is data. 1b = Next frame transmitted is an address.
1	UCTXBRK	RW	0h	Transmit break. Transmits a break with the next write to the transmit buffer. In UART mode with automatic baud-rate detection, 055h must be written into UCAxTXBUF to generate the required break/synch fields. Otherwise, 0h must be written into the transmit buffer. 0b = Next frame transmitted is not a break. 1b = Next frame transmitted is a break or a break/synch.
0	UCSWRST	RW	1h	Software reset enable 0b = Disabled. eUSCI_A reset released for operation. 1b = Enabled. eUSCI_A logic held in reset state.

© tj

UART

- MSP432 UART Control Word 1 Register

Figure 22-13. UCAxCTLW1 Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
Reserved						UCGLITx	
r-0	r-0	r-0	r-0	r-0	r-0	rw-1	rw-1

Table 22-9. UCAxCTLW1 Register Description

Bit	Field	Type	Reset	Description
15-2	Reserved	R	0h	Reserved
1-0	UCGLITx	RW	3h	Deglitch time 00b = Approximately 5 ns 01b = Approximately 20 ns 10b = Approximately 30 ns 11b = Approximately 50 ns

© tj

UART

- MSP432 UART Baud Rate Control Word Register

Figure 22-14. UCAxBRW Register

15	14	13	12	11	10	9	8
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
UCBRx							
rw	rw	rw	rw	rw	rw	rw	rw

Modify only when UCSWRST = 1

Table 22-10. UCAxBRW Register Description

Bit	Field	Type	Reset	Description
15-0	UCBRx	RW	0h	Clock prescaler setting of the baud-rate generator

© tj

UART

- MSP432 UART Modulation Control Word Register

Figure 22-15. UCAxMCTLW Register

15	14	13	12	11	10	9	8
UCBRSx							
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
UCBRFx				Reserved			UCOS16
rw-0	rw-0	rw-0	rw-0	r0	r0	r0	rw-0
Modify only when UCSWRST = 1							

Table 22-11. UCAxMCTLW Register Description

Bit	Field	Type	Reset	Description
15-8	UCBRSx	RW	0h	Second modulation stage select. These bits hold a free modulation pattern for BITCLK.
7-4	UCBRFx	RW	0h	First modulation stage select. These bits determine the modulation pattern for BITCLK16 when UCOS16 = 1. Ignored with UCOS16 = 0. The "Oversampling Baud-Rate Generation" section shows the modulation pattern.
3-1	Reserved	R	0h	Reserved
0	UCOS16	RW	0h	Oversampling mode enabled 0b = Disabled 1b = Enabled

© tj

UART

- MSP432 UART Status Word Register

Figure 22-16. UCAxSTATW Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
UCLISTEN	UCFE	UCOE	UCPE	UCBRK	UCRXERR	UCADDR UCIDLE	UCBUSY
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	r-0
	Modify only when UCSWRST = 1						

Table 22-12. UCAxSTATW Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7	UCLISTEN	RW	0h	Listen enable. The UCLISTEN bit selects loopback mode. 0b = Disabled 1b = Enabled. UCAxTXD is internally fed back to the receiver.
6	UCFE	RW	0h	Framing error flag. UCFE is cleared when UCAxRXBUF is read. 0b = No error 1b = Character received with low stop bit
5	UCOE	RW	0h	Overrun error flag. This bit is set when a character is transferred into UCAxRXBUF before the previous character was read. UCOE is cleared automatically when UCxRXBUF is read, and must not be cleared by software. Otherwise, it does not function correctly. 0b = No error 1b = Overrun error occurred.

© tj

UART

- MSP432 UART Status Word Register

Bit	Field	Type	Reset	Description
4	UCPE	RW	0h	Parity error flag. When UCPEN = 0, UCPE is read as 0. UCPE is cleared when UCAxRXBUF is read. 0b = No error 1b = Character received with parity error
3	UCBRK	RW	0h	Break detect flag. UCBRK is cleared when UCAxRXBUF is read. 0b = No break condition 1b = Break condition occurred.
2	UCRXERR	RW	0h	Receive error flag. This bit indicates a character was received with one or more errors. When UCRXERR = 1, one or more error flags, UCFE, UCPE, or UCOE is also set. UCRXERR is cleared when UCAxRXBUF is read. 0b = No receive errors detected 1b = Receive error detected
1	UCADDR UCIDLE	RW	0h	UCADDR: Address received in address-bit multiprocessor mode. UCADDR is cleared when UCAxRXBUF is read. UCIDLE: Idle line detected in idle-line multiprocessor mode. UCIDLE is cleared when UCAxRXBUF is read. 0b = UCADDR: Received character is data. UCIDLE: No idle line detected 1b = UCADDR: Received character is an address. UCIDLE: Idle line detected
0	UCBUSY	R	0h	eUSCI_A busy. This bit indicates if a transmit or receive operation is in progress. 0b = eUSCI_A inactive 1b = eUSCI_A transmitting or receiving

© tj

UART

- MSP432 UART Receive Buffer Register

Figure 22-17. UCAxRXBUF Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
UCRXBUFx							
r	r	r	r	r	r	r	r

Table 22-13. UCAxRXBUF Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7-0	UCRXBUFx	R	0h	The receive-data buffer is user accessible and contains the last received character from the receive shift register. Reading UCAxRXBUF resets the receive-error bits, the UCADDR or UCIDLE bit, and UCRXIFG. In 7-bit data mode, UCAxRXBUF is LSB justified and the MSB is always reset.

© tj

UART

- MSP432 UART Transmit Buffer Register

Figure 22-18. UCAxTXBUF Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
UCTXBUFx							
rw	rw	rw	rw	rw	rw	rw	rw

Table 22-14. UCAxTXBUF Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	R	0h	Reserved
7-0	UCTXBUFx	RW	0h	The transmit data buffer is user accessible and holds the data waiting to be moved into the transmit shift register and transmitted on UCAxTXD. Writing to the transmit data buffer clears UCTXIFG. The MSB of UCAxTXBUF is not used for 7-bit data and is reset.

© tj

UART

- MSP432 UART Auto Baud Rate Control Register

Figure 22-19. UCAxABCTL Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
Reserved		UCDELIMx		UCSTOE	UCBTOE	Reserved	UCABDEN
r-0	r-0	rw-0	rw-0	rw-0	rw-0	r-0	rw-0

Modify only when UCSWRST = 1

Table 22-15. UCAxABCTL Register Description

Bit	Field	Type	Reset	Description
15-6	Reserved	R	0h	Reserved
5-4	UCDELIMx	RW	0h	Break/synch delimiter length 00b = 1 bit time 01b = 2 bit times 10b = 3 bit times 11b = 4 bit times
3	UCSTOE	RW	0h	Synch field time out error 0b = No error 1b = Length of synch field exceeded measurable time.
2	UCBTOE	RW	0h	Break time out error 0b = No error 1b = Length of break field exceeded 22 bit times.
1	Reserved	R	0h	Reserved
0	UCABDEN	RW	0h	Automatic baud-rate detect enable 0b = Baud-rate detection disabled. Length of break and synch field is not measured. 1b = Baud-rate detection enabled. Length of break and synch field is measured and baud-rate settings are changed accordingly.

© tj

UART

- MSP432 UART IRDA Control Register

Figure 22-20. UCAxIRCTL Register

15	14	13	12	11	10	9	8
UCIRRXFLx						UCIRRXPL	UCIRRXFE
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0
7	6	5	4	3	2	1	0
UCIRTXPLx						UCIRTXCLK	UCIREN
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

Modify only when UCSWRST = 1

Table 22-16. UCAxIRCTL Register Description

Bit	Field	Type	Reset	Description
15-10	UCIRRXFLx	RW	0h	Receive filter length. The minimum pulse length for receive is given by: $t_{MIN} = (UCIRRXFLx + 4) / (2 \times f_{IRTXCLK})$
9	UCIRRXPL	RW	0h	IrDA receive input UCAxRXD polarity 0b = IrDA transceiver delivers a high pulse when a light pulse is seen. 1b = IrDA transceiver delivers a low pulse when a light pulse is seen.
8	UCIRRXFE	RW	0h	IrDA receive filter enabled 0b = Receive filter disabled 1b = Receive filter enabled
7-2	UCIRTXPLx	RW	0h	Transmit pulse length. Pulse length $t_{PULSE} = (UCIRTXPLx + 1) / (2 \times f_{IRTXCLK})$
1	UCIRTXCLK	RW	0h	IrDA transmit pulse clock select 0b = BRCLK 1b = BITCLK16 when UCOS16 = 1. Otherwise, BRCLK.
0	UCIREN	RW	0h	IrDA encoder and decoder enable 0b = IrDA encoder/decoder disabled 1b = IrDA encoder/decoder enabled

© tj

UART

- MSP432 UART Interrupt Enable Register

Figure 22-21. UCAxIE Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
Reserved				UCTXCPTIE	UCSTTIE	UCTXIE	UCRXIE
r-0	r-0	r-0	r-0	rw-0	rw-0	rw-0	rw-0

Table 22-17. UCAxIE Register Description

Bit	Field	Type	Reset	Description
15-4	Reserved	R	0h	Reserved
3	UCTXCPTIE	RW	0h	Transmit complete interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
2	UCSTTIE	RW	0h	Start bit interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
1	UCTXIE	RW	0h	Transmit interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled
0	UCRXIE	RW	0h	Receive interrupt enable 0b = Interrupt disabled 1b = Interrupt enabled

© tj

UART

- MSP432 UART Interrupt Flag Register

Figure 22-22. UCAxIFG Register

15	14	13	12	11	10	9	8
Reserved							
r-0	r-0	r-0	r-0	r-0	r-0	r-0	r-0
7	6	5	4	3	2	1	0
Reserved				UCTXCPTIFG	UCSTTIFG	UCTXIFG	UCRXIFG
r-0	r-0	r-0	r-0	rw-0	rw-0	rw-1	rw-0

Table 22-18. UCAxIFG Register Description

Bit	Field	Type	Reset	Description
15-4	Reserved	R	0h	Reserved
3	UCTXCPTIFG	RW	0h	Transmit complete interrupt flag. UCTXCPTIFG is set when the entire byte in the internal shift register got shifted out and UCAxTXBUF is empty. 0b = No interrupt pending 1b = Interrupt pending
2	UCSTTIFG	RW	0h	Start bit interrupt flag. UCSTTIFG is set after a Start bit was received 0b = No interrupt pending 1b = Interrupt pending
1	UCTXIFG	RW	1h	Transmit interrupt flag. UCTXIFG is set when UCAxTXBUF empty. 0b = No interrupt pending 1b = Interrupt pending
0	UCRXIFG	RW	0h	Receive interrupt flag. UCRXIFG is set when UCAxRXBUF has received a complete character. 0b = No interrupt pending 1b = Interrupt pending

© tj

UART

- MSP432 UART Interrupt Vector Register

Figure 22-23. UCAxIV Register

15	14	13	12	11	10	9	8
UCIVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
UCIVx							
r0	r0	r0	r0	r-(0)	r-(0)	r-(0)	r0

Table 22-19. UCAxIV Register Description

Bit	Field	Type	Reset	Description
15-0	UCIVx	R	0h	eUSCI_A interrupt vector value 00h = No interrupt pending 02h = Interrupt Source: Receive buffer full; Interrupt Flag: UCRXIFG; Interrupt Priority: Highest 04h = Interrupt Source: Transmit buffer empty; Interrupt Flag: UCTXIFG 06h = Interrupt Source: Start bit received; Interrupt Flag: UCSTTIFG 08h = Interrupt Source: Transmit complete; Interrupt Flag: UCTXCPTIFG; Interrupt Priority: Lowest

© tj

UART

- Loop-back Example

```
/*
 * uart.c
 *
 *  Created on: Oct 12, 2017
 *      Author: johnsontimoj
 */
//
// uart test file
// Loop back from A1 to A2

#include <stdio.h>
#include <stdint.h>
#include "msp432.h"
#include "msoe_lib_all.h"

void pin_setup(void);
void uart_tx_setup(void);
void uart_rx_setup(void);
void uart_A1_transmit(uint8_t data);

// Global Variable for Interrupts
uint8_t rx_data;
uint8_t tx_data;
uint8_t num_bytes;
```

```
int main(void){
    //
    // setup counter
    tx_data = 0;

    // Configure pins
    pin_setup();

    // Setup TWIs
    uart_tx_setup();
    uart_rx_setup();

    while(1){
        // start transmit
        uart_A1_transmit(tx_data);

        // wait for transfer to complete
        Delay_3MHz_ms(50);
        // Print rcvd values
        printf("Tx UART sent %i - Rx UART received %i\n", tx_data, rx_data);
        // Update count and wait

        tx_data++;
    }
    return 0;
} // end main
```


© tj

UART

- Loop-back Example

```
void pin_setup(void){
    // Setup USCIA1 transmit
    // Setup USCIA2 Receive
    // Transfer in a while loop
    //
    // USCIA1 --> P2.2-RX, P2.3-TX
    // USCIA2 --> P3.2-RX, P3.3-TX
    //
    P2->SEL0 |= 0x0C;   // Set to USC mode
    P2->SEL1 &= ~0x0C;  // 0-1 mode
    P3->SEL0 |= 0x0C;
    P3->SEL1 &= ~0xC;

    // Nothing else needed - the module
    // sets directions
    return;
} // end pin_setup
```

```
void uart_tx_setup(void){
    // USCIA1 - transmitter
    // 8 bit, Uart mode, even parity
    //
    // CTLW0 - allow changes
    // sw reset
    EUSCI_A1->CTLW0 = 0x0001;
    //
    // CLTW0
    // P   E  LSB  8b  1st  uart  Async  Aclk   nochie  nobrkie awake noaddr  nobrk rst
    // 1   1   0   0    0    00     0     01       0       0     0      0       0    1
    // C041
    EUSCI_A1->CTLW0 = 0xC041;
    //
    // No Deglitch
    //
    // BRW
    //  /64 - 0000 0000 0100 0000
    // 0040
    EUSCI_A1->BRW = 0x0040;
    //
    // No Modulation
    //
    // No Auto baud rate
    //
    // No IrDA
    //
    // CTLW0
    // release reset
    EUSCI_A1->CTLW0 &= ~0x01;
    //
    // IE
    //               TC   st  TXbuf  RC
    // xxxx xxxx xxxx 0   0     1     0
    // 0002
    EUSCI_A1->IE = 0x0002;
    //
    // NVIC
    // EUSCI_A1 is int 17
    //
    NVIC->IP[17] = 0x60;            // priority = 3
    NVIC->ISER[0] |= 0x00020000;    // enable

    return;
}   // end uart_tx_setup
```


© tj

UART

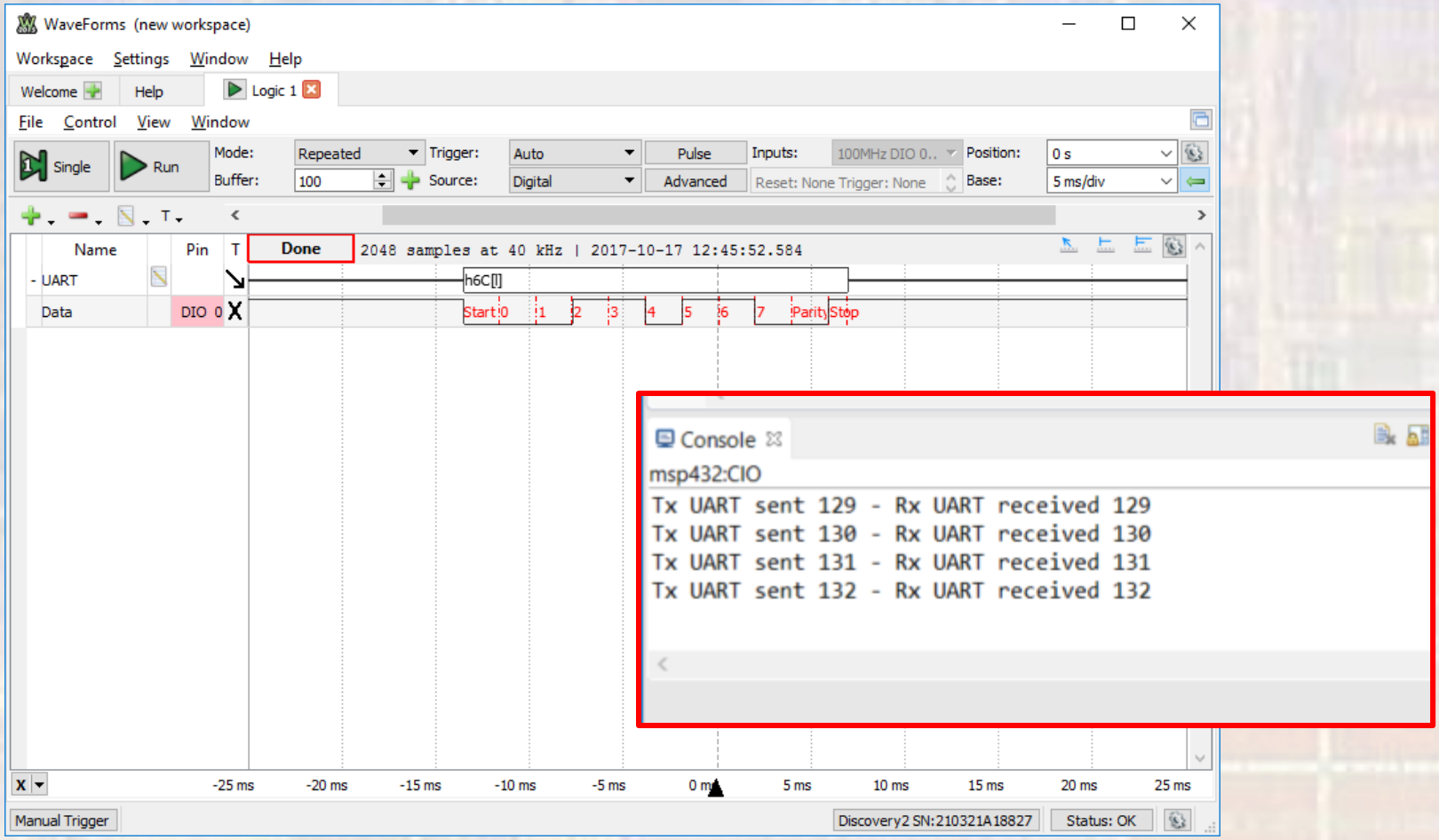
```
void uart_rx_setup(void){
    // USCIA2 - receiver
    // 8 bit, Uart mode, even parity
    //
    // CTLW0 - allow changes
    // sw reset
    EUSCI_A2->CTLW0 = 0x0001;
    //
    // CLTW0
    // P   E  LSB  8b  1st  uart  Async  Aclk   nochie  nobrkie awake noaddr  nobrk rst
    // 1   1   0   0    0    00     0     01       0       0     0      0       0    1
    // C041
    EUSCI_A2->CTLW0 = 0xC041;
    //
    // No Deglitch
    //
    // BRW
    //  /64 - 0000 0000 0100 0000
    // 0040
    EUSCI_A2->BRW = 0x0040;
    //
    // No Modulation
    //
    // No Auto baud rate
    //
    // No IrDA
    //
    // CTLW0
    // release reset
    EUSCI_A2->CTLW0 &= ~0x01;
    //
    // IE
    //               TC   st  TXbuf  RC
    // xxxx xxxx xxxx 0   0     0     1
    // 0002
    EUSCI_A2->IE = 0x0001;
    //
    // NVIC
    // EUSCI_A2 is int 18
    //
    NVIC->IP[18] = 0x60;            // priority = 3
    NVIC->ISER[0] |= 0x00040000;    // enable

    return;
}   // end uart_rx_setup
```

```
void uart_A1_transmit(uint8_t data){
    // Check for  busy
    while(EUSCI_A1->STATW & 0x0001)
        ;
    // tx buffer
    EUSCI_A1->TXBUF = data;

    return;
} // end uart_A1_transmit

void EUSCIA1_IRQHandler(void){
    // Check for TX buffer empty (tx complete)
    if(EUSCI_A1->IFG & 0x0002){
        // clear flag
        EUSCI_A1->IFG &= ~0x0002;
    }
} // end uart tx int handler

void EUSCIA2_IRQHandler(void){
    // Check for RX flag (receive complete)
    if(EUSCI_A2->IFG & 0x0001){
        EUSCI_A2->IFG &= ~0x0001;   // clear flag
        rx_data = EUSCI_A2->RXBUF;   // copy data
    }
} // end uart rx int handler
```


© tj

UART

- MSP432 UART
  - Examples



 © tj