

ECE375 Lab 8

TA: Guangxin Wang

School of Electrical Engineering and Computer Science
Oregon State University

Remotely Operated Vehicle

- Learn how to configure and use the Universal Synchronous/Asynchronous Receiver/Transmitter (USART) module on the ATmega128 microcontroller
- 2-week long lab but **START EARLY!!!**
- Program Rx & Tx
- Follow Frame Format and Baud Rate
 - Your bot need to communicate TA's bot as well
- Read Atmega128 Datasheet
 - 188p – 193p

Plagiarism Issues

- Using Partial Code not allowed
 - It is never okay to represent another person's partial codes as your own
- Changing only comments, variable, and register name
 - If order of instruction sets are same, it will be flagged as plagiarism even comments, variable, and register name are different.

Communicate Two Boards

- Byte1: Robot Address
- Byte2: Action Code

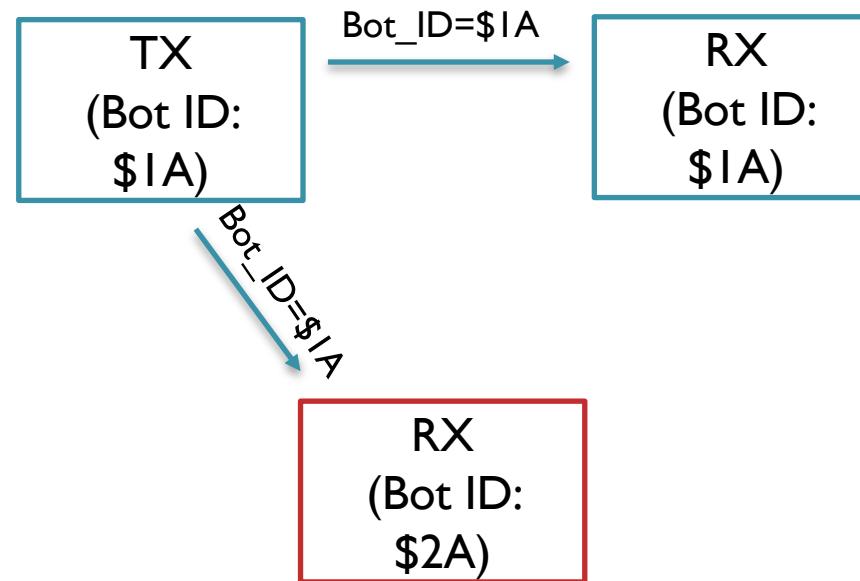
Demo

TX
(Bot ID:
\$1A)

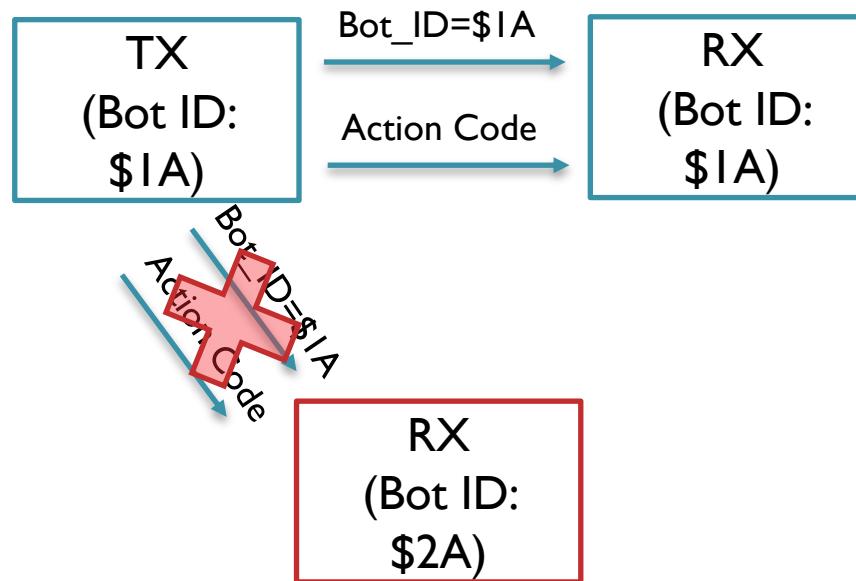
RX
(Bot ID:
\$1A)

RX
(Bot ID:
\$2A)

Demo

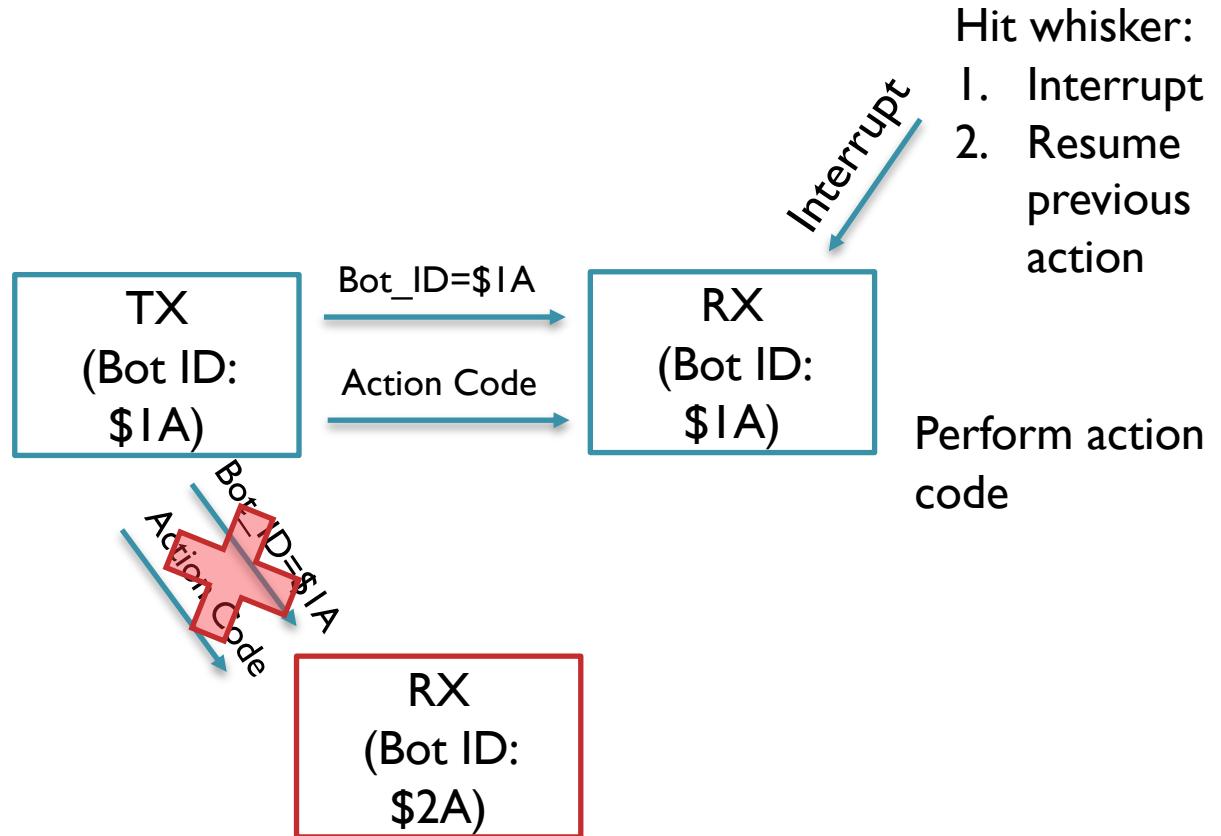


Demo

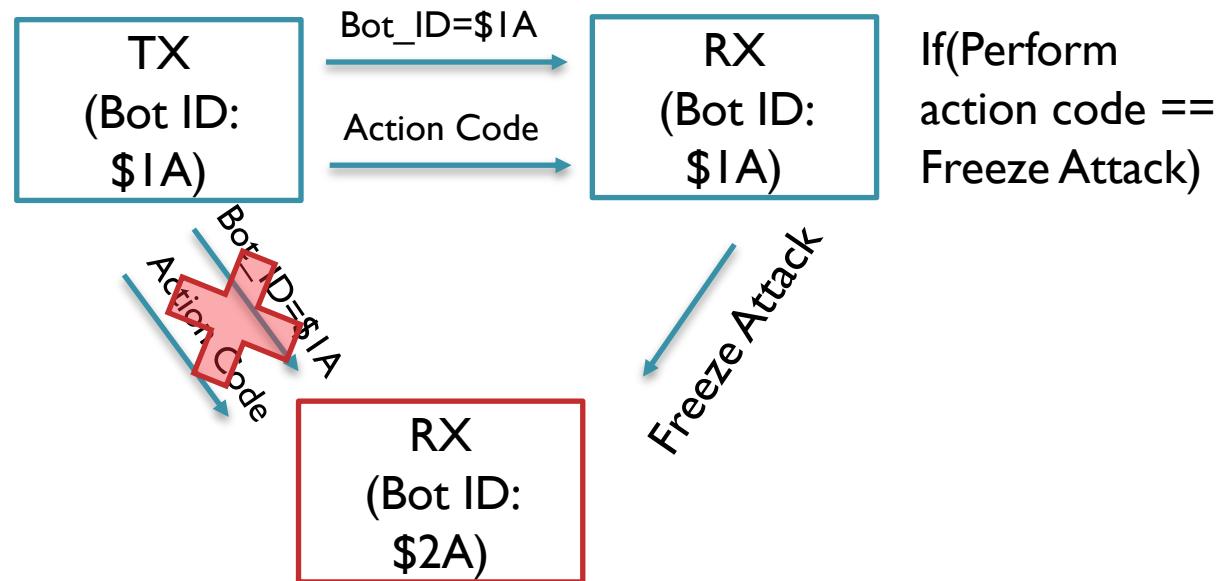


Perform action code
(eg.
Forward
Halt
Backward
Left
Right
Freeze!
)

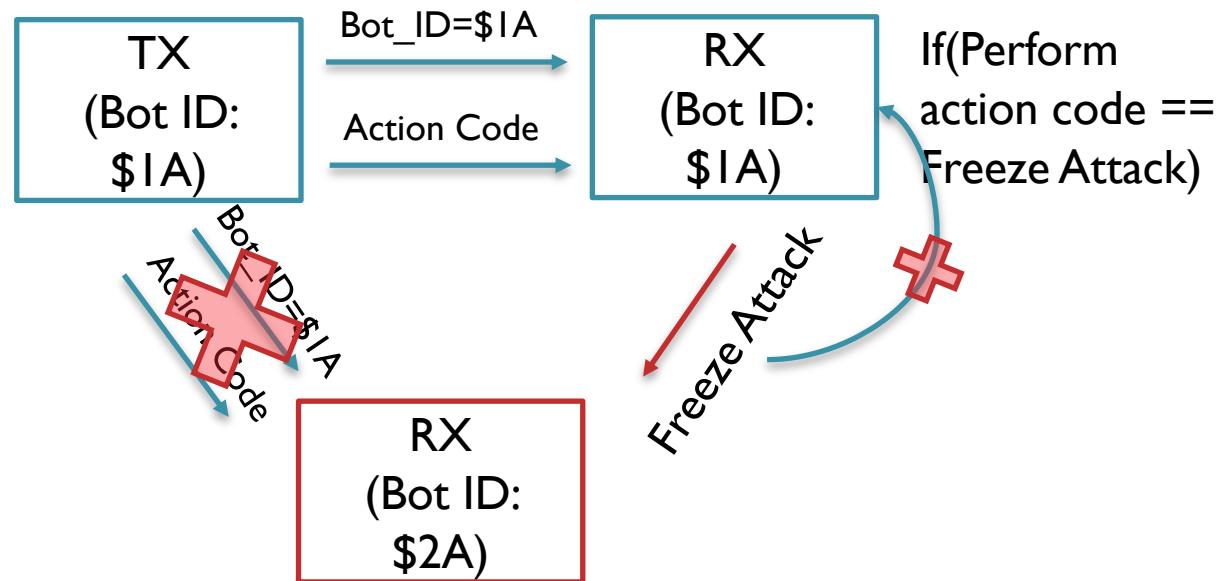
Demo



Demo



Demo

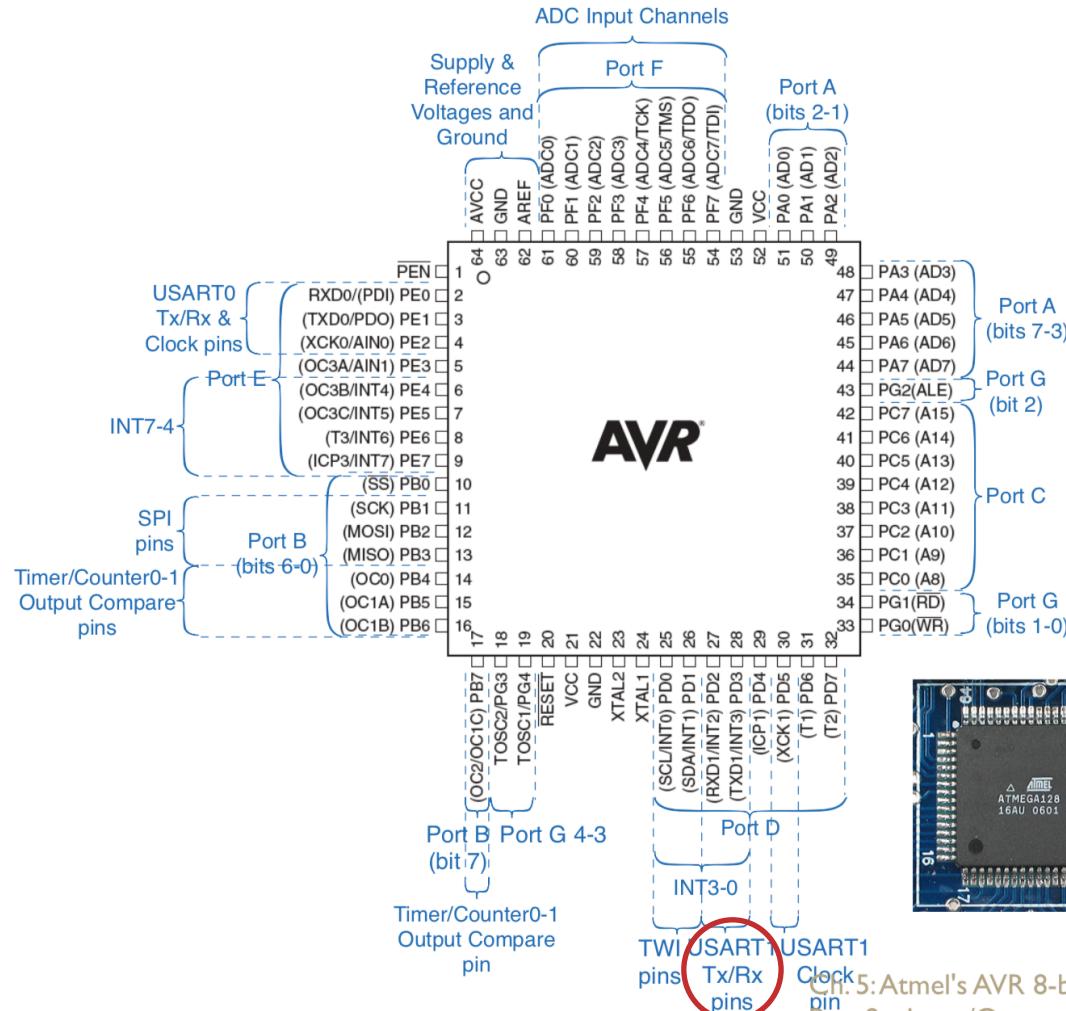


Frozen once (trigger 1 time):
take effect for 5 sec
Halt
No whisker interrupt
No receive action code

Frozen forever(trigger 3 times):
No whisker interrupt
No receive action code

Port Initialization

I/O Pins



Ch. 5: Atmel's AVR 8-bit Microcontroller,
Part 2 - Input/Output

Initialization

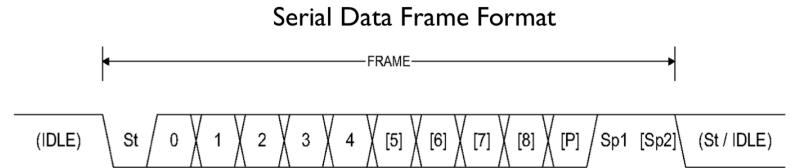
- Frame Format
 - Data Frame : 8-bit data
 - Stop bit : 2 stop bits
 - Parity bit : disable
- Baud Rate - 2400 bits per second (Check datasheet P. 196)

$$\text{Baud Rate} = \frac{f_{\text{CLK}}}{16 \cdot (\text{UBRR} + 1)}$$

Baud rate divider USART Baud Rate Register

$$\text{UBRR} = \frac{f_{\text{CLK}}}{16 \cdot (\text{Baud Rate})} - 1$$

- Control Register
 - UCSRIA
 - UCSRIB
 - UCSRIC
 - UBRRIH
 - UBRRIL



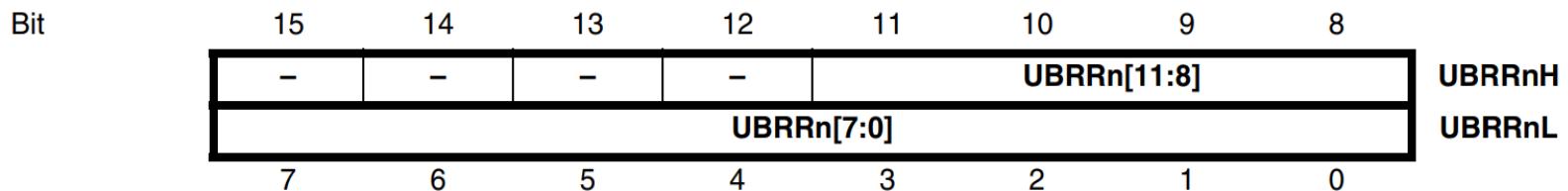
- St Start bit, always low.
 (n) Data bits (0 to 8).
 P Parity bit. Can be odd or even.
 Sp Stop bit, always high.
 IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

Table 85. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	$f_{\text{osc}} = 16.0000\text{MHz}$			
	U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	-0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	-0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	-0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	-3.5%	16	2.1%
230.4k	3	8.5%	8	-3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max ⁽¹⁾	1Mbps		2Mbps	

1. UBRR = 0, Error = 0.0%

USART Baud Rate Registers



Bit 15:12 – Reserved Bits

Bit 11:0 – UBRRn[11:0]: USARTn Baud Rate Register

UBRIH

UBRIL

UMSEL_I / UPM_I

Table 77. UMSEL_n Bit Settings

UMSEL _n	Mode
0	Asynchronous Operation
1	Synchronous Operation

Table 78. UPM_n Bits Settings

UPM _{n1}	UPM _{n0}	Parity Mode
0	0	Disabled
0	1	(Reserved)
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

USBSI / UCPOLI

Table 79. USBSn Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bits

Table 81. UCPOLn Bit Settings

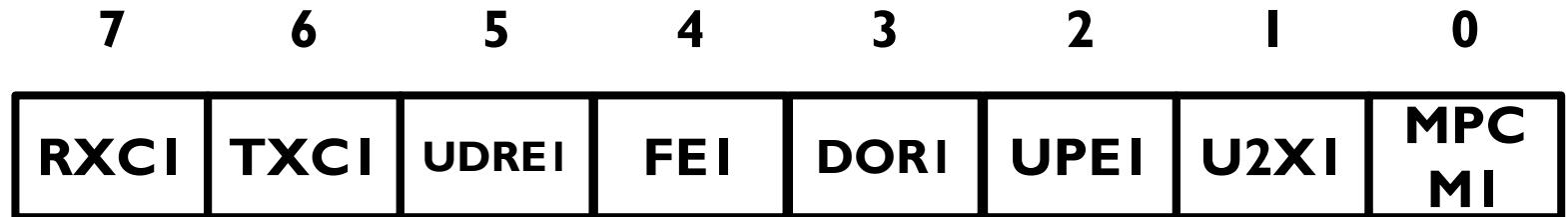
UCPOLn	Transmitted Data Changed (Output of TxDn Pin)	Received Data Sampled (Input on RxDn Pin)
0	Rising XCKn Edge	Falling XCKn Edge
1	Falling XCKn Edge	Rising XCKn Edge

UCSZ I

Table 80. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

USART Control and Status Register



UCSRIA

Bit 7 – RXCn: USART Receive Complete

Bit 6 – TXCn: USART Transmit Complete

Bit 5 – UDREn: USART Data Register Empty

Bit 4 – FEn: Frame Error

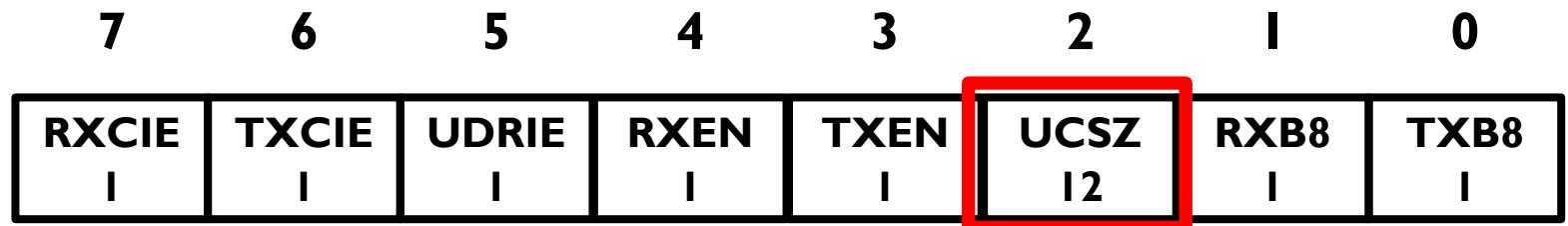
Bit 3 – DORn: Data OverRun

Bit 2 – UPEn: Parity Error

Bit 1 – U2Xn: Double the USART Transmission Speed

Bit 0 – MPCMn: Multi-Processor Communication Mode

USART Control and Status Register



UCSRIB

Bit 7 – RXCIEn: RX Complete Interrupt Enable

Bit 6 – TXCIE: TX Complete Interrupt Enable

Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable

Bit 4 – RXENn: Receiver Enable

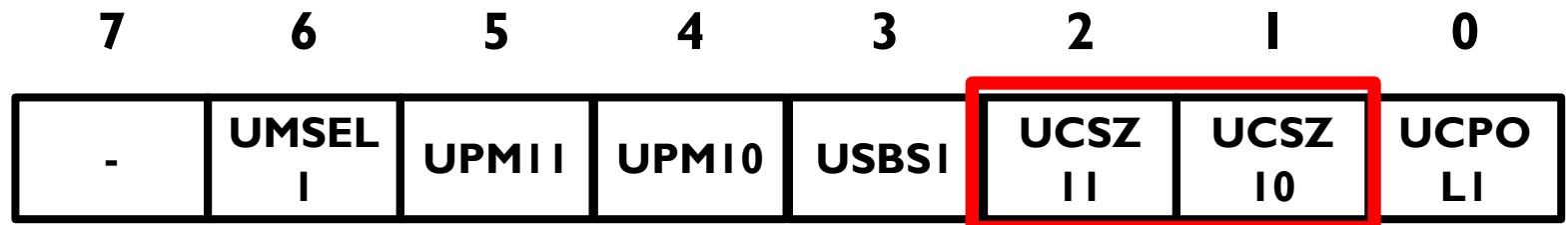
Bit 3 – TXENn: Transmitter Enable

Bit 2 – UCSZn2: Character Size

Bit 1 – RXB8n: Receive Data Bit 8

Bit 0 – TXB8n: Transmit Data Bit 8

USART Control and Status Register



UCSRIC

Bit 7 – Reserved Bit

Bit 6 – UMSEL_n: USART Mode Select

Bit 5:4 – UPM_nI:0: Parity Mode

Bit 3 – USBS_n: Stop Bit Select

Bit 2:I – UCSZ_nI:0: Character Size

Bit 0 – UCPOL_n: Clock Polarity

Demo Check

- All five actions successfully received via IR and performed by Robot board
 - Forward, Backward, Turn Left, Turn Right, Halt
- Robot only responds to remote w/ matching address
- Remote sends freeze action code, NOT freeze signal

Test your Tx with Sample Rx

- Frame Format
 - Data Frame : 8 data bits
 - Stop bit : 2 stop bits
 - Parity bit : disable
 - Asynchronous Operation
 - Double data rate
 - Received Data Sampled @ Falling XCKn Edge
 - Baud Rate - 2400 bits per second
 - BotID = \$2A
- Do not use the same BotID(\$2A) during the Demo session. If your Rx responds to the TA's Tx, you will get 0 for Rx part. Because it will be suspected as using sample Rx.

Checklists for Lab 8

- Part I Demo Checklist
 - All five actions successfully received via IR and performed by Robot board
 - Most-recent action continues even after button is released on Remote board (no holding down needed)
 - Usual BumpBot behavior implemented by Robot
 - Commands received via USART ignored while inside HitRight/HitLeft subroutines
 - Robot resumes prior (pre-Right/HitLeft) action after completing HitRight/Hitleft subroutines
 - Robot only responds to remote w/ matching address
 - Correct baud rate setting of 2400 bits per second

Checklists for Lab 8

- Part 2 Demo Checklist
 - Remote sends freeze action code, NOT freeze signal
 - When robot sends freeze signal directly, freezes other robots but not itself (no self-freezing)
 - When frozen, no response to USART or whisker hits
 - Resumes pre-freeze action after becoming unfrozen
 - Permanent freeze on third received freeze signal
- Challenge Checklist
 - Timer/Counter1 used (with interrupts) to implement 1 second WAIT function
 - Halt and Freeze replaced with Lab 6 SpdUp, SpdDn (with all related Lab6/PWM+Port B code included)

Questions?

