Serial Communication On Firebird V Robot

e-Yantra Team
Embedded Real-Time Systems Lab
Indian Institute of Technology-Bombay

IIT Bombay
June 20, 2014

Agenda for Discussion

1. Introduction to Serial Communication
   - What is Serial Communication
   - Needs and Ways of Serial Communication
   - Inbuilt UART pins of ATmega 2560
2. Registers used in serial communication
   - Types of registers
   - UCSRnA
   - UCSRnB
   - UCSRnC
   - UBRRnL & UBRRnH
   - UDRn
3. Interrupts in Serial communication
   - Receive Complete ISR
   - Data Register empty ISR
   - Transmit Complete ISR
4. C Code
   - UART initialization

What is Serial Communication

Serial communication is the process of sending data one bit at a time, sequentially, over a communication channel. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.

Ways of Serial Communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between

Ways of Serial Communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots

Ways of Serial Communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
- Wireless communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
  2. RS232
- Wireless communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
  2. RS232
  3. etc.
- Wireless communication

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
  2. RS232
  3. etc.
- Wireless communication
  1. Zigbee

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
  2. RS232
  3. etc.
- Wireless communication
  1. Zigbee
  2. Bluetooth

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
    1. Two/multiple robots
    2. Robots and external devices

Ways of Serial Communication

- Wired communication
    1. USB
    2. RS232
    3. etc.
- Wireless communication
    1. Zigbee
    2. Bluetooth
    3. WiFi

Needs and Ways of Serial Communication

Needs of Serial Communication.

- To establish a communication between devices like PCs, Tablets and other external devices.
- To establish communication between
  1. Two/multiple robots
  2. Robots and external devices

Ways of Serial Communication

- Wired communication
  1. USB
  2. RS232
  3. etc.
- Wireless communication
  1. Zigbee
  2. Bluetooth
  3. WiFi
  4. etc.

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module
- UART1 to RS232 Serial port

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module
- UART1 to RS232 Serial port
- UART2 to FT232 USB serial converter

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module
- UART1 to RS232 Serial port
- UART2 to FT232 USB serial converter
- UART3 to expansion port

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module
- UART1 to RS232 Serial port
- UART2 to FT232 USB serial converter
- UART3 to expansion port

Inbuilt UART pins of ATmega 2560

ATmega 2560 supports 4 UARTs(UART 0-3). In Firebird V these are configured to following devices by default.

- UART0 to Zigbee Wireless module
- UART1 to RS232 Serial port
- UART2 to FT232 USB serial converter
- UART3 to expansion port

UARTx	Rx	Tx	module
UART0	PORTE0	PORTE1	Zigbee
UART1	PORTD2	PORTD3	RS232
UART2	PORTH0	PORTH1	USB
UART3	PORTJ0	PORTJ1	expansion slot

Types of Registers

These are the various registers involved in serial communication :

- UCSRnA = USART control and status register nA.

Note: n represents the UART number which can be 0,1,2,3....

Types of Registers

These are the various registers involved in serial communication :

- UCSRnA = USART control and status register nA.
- UCSRnB = USART control and status register nB.

Note: n represents the UART number which can be 0,1,2,3....

Types of Registers

These are the various registers involved in serial communication :

- UCSRnA = USART control and status register nA.
- UCSRnB = USART control and status register nB.
- UCSRnC = USART control and status register nC.

Note: n represents the UART number which can be 0,1,2,3....

Types of Registers

These are the various registers involved in serial communication :

- UCSRnA = USART control and status register nA.
- UCSRnB = USART control and status register nB.
- UCSRnC = USART control and status register nC.
- UBRRnL & UBRRnH = USART baud rate registers.

Note: n represents the UART number which can be 0,1,2,3....

Types of Registers

These are the various registers involved in serial communication :

- UCSRnA = USART control and status register nA.
- UCSRnB = USART control and status register nB.
- UCSRnC = USART control and status register nC.
- UBRRnL & UBRRnH = USART baud rate registers.
- UDRn = USART input/output register n.

Note: n represents the UART number which can be 0,1,2,3....

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0
3	DORn	Data Over-Run	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0
3	DORn	Data Over-Run	0
2	UPEn	Parity Error	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0
3	DORn	Data Over-Run	0
2	UPEn	Parity Error	0
1	U2Xn	Double transmission speed	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0
3	DORn	Data Over-Run	0
2	UPEn	Parity Error	0
1	U2Xn	Double transmission speed	0
0	MPCMn	MultiProcessor Communication Mode	0

UCSRnA-Control and Status Register nA.

Bit	Symbol	Description	Bit Value
7	RxCn	Receive Complete	0
6	TxCn	Transmit Complete	0
5	UDREn	Data Register Empty	0
4	FEn	Frame Error	0
3	DORn	Data Over-Run	0
2	UPEn	Parity Error	0
1	U2Xn	Double transmission speed	0
0	MPCMn	MultiProcessor Communication Mode	0

UCSRnA=0x00

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1
3	TXENn	Transmiter Enable	1

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1
3	TXENn	Transmiter Enable	1
2	UCSZn2	char size n	0

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1
3	TXENn	Transmiter Enable	1
2	UCSZn2	char size n	0
1	RXB8n	Receive data bit 8	0

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1
3	TXENn	Transmiter Enable	1
2	UCSZn2	char size n	0
1	RXB8n	Receive data bit 8	0
0	TXB8n	Transmit data bit 8	0

UCSRnB-Control and Status Register nB.

Bit	Symbol	Description	Bit Value
7	RxCIEn	Receive Complete Interrupt Enable	1
6	TxCIEn	Transmit Complete Interrupt Enable	0
5	UDRIEn	Data Register Empty	0
4	RXENn	Receiver Enable	1
3	TXENn	Transmiter Enable	1
2	UCSZn2	char size n	0
1	RXB8n	Receive data bit 8	0
0	TXB8n	Transmit data bit 8	0

UCSRnB=0x98

UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value

UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0

UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0
5	UPMn1	Parity Mode	0
4	UPMn0	Parity Mode	0

UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0
5	UPMn1	Parity Mode	0
4	UPMn0	Parity Mode	0
3	USBSn	Stop Bit Select	0

UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0
5	UPMn1	Parity Mode	0
4	UPMn0	Parity Mode	0
3	USBSn	Stop Bit Select	0
2	UCSZn1	Character Size	1
1	UCSZn0	Character Size	1

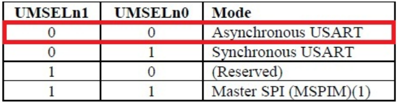
UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0
5	UPMn1	Parity Mode	0
4	UPMn0	Parity Mode	0
3	USBSn	Stop Bit Select	0
2	UCSZn1	Character Size	1
1	UCSZn0	Character Size	1
0	UCP0Ln	Clock polarity	0

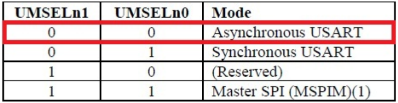
UCSRnC-Control and Status Register nC.

Bit	Symbol	Description	Bit Value
7	UMSELn1	USART Mode Select	0
6	UMSELn0	USART Mode Select	0
5	UPMn1	Parity Mode	0
4	UPMn0	Parity Mode	0
3	USBSn	Stop Bit Select	0
2	UCSZn1	Character Size	1
1	UCSZn0	Character Size	1
0	UCP0Ln	Clock polarity	0

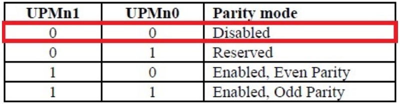
UCSRnC=0x06



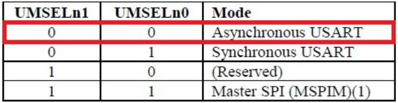
UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)



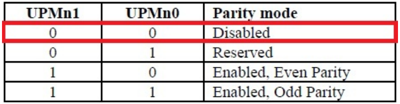
UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)



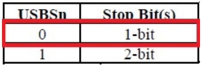
UPMn1	UPMn0	Parity mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity



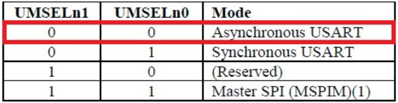
UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)



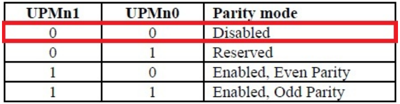
UPMn1	UPMn0	Parity mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity



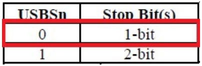
USBSn	Stop Bit(s)
0	1-bit
1	2-bit



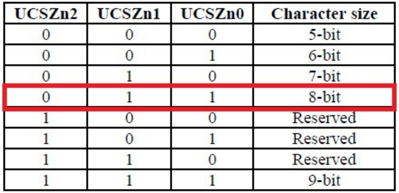
UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)(1)



UPMn1	UPMn0	Parity mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity



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



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

UBRRnL & UBRRnH-Baud Rate Registers

These two registers are used to set baud rates.
Crystal frequency is 14.7456MHz.

UBRRnL & UBRRnH-Baud Rate Registers

These two registers are used to set baud rates.
Crystal frequency is 14.7456MHz.

example:

Let us consider for a baud rate of 9600

$$UBRR = \left\{ \frac{Systemclock}{16 * BaudRate} \right\} - 1$$

$$UBRR = \left\{ \frac{14.7456Mhz}{16 * 9600} \right\} - 1$$

$$UBRR = 95$$

$$UBRR = 0x5FH$$

$$UBRRH = 0x00H$$

$$UBRRL = 0x5FH$$

Note:While loading values in UBRR register load values in the UBRRH register first.

UDRn-USART I/O Data Register n

- The USART Transmit Data Buffer Register and USART receive data buffer register share the same I/O address referred to as USART data Registers or UDR.

UDRn-USART I/O Data Register n

- The USART Transmit Data Buffer Register and USART receive data buffer register share the same I/O address referred to as USART data Registers or UDR.
- The transmit Data Buffer register(TxB) will be the destination for data written to the UDRn register location.

UDRn-USART I/O Data Register n

- The USART Transmit Data Buffer Register and USART receive data buffer register share the same I/O address referred to as USART data Registers or UDR.
- The transmit Data Buffer register(TxB) will be the destination for data written to the UDRn register location.
- Reading the UDRn Register location will return the contents of the received data buffer register(RxB).

Receive Complete ISR

Receive Complete ISR

```
SIGNAL(SIG_USARTn_RECV)// ISR for receive complete interrupt.
{
  data = UDRn; //Making a copy of data from UDRn in 'data' variable.
}
```

If RXCIE interrupt is enabled then receive complete interrupt triggers ISR.

Data Register empty ISR

Data Register empty ISR

```
SIGNAL(SIG_USARTn_DATA) // ISR for Data Register empty interrupt.
{
  UDRn = tx_data; //data that need to be transmitted is transferred to
                                        UDRn
}
```

If UDRIE interrupt is enabled then UDRn data register empty interrupt triggers ISR. This ISR then loads next data byte to be transmitted into UDRn.

Transmit Complete ISR

Transmit Complete ISR

```
SIGNAL(SIG_USARTn_TRANS)// ISR for Transmit complete interrupt.
{
//Insert your code
}
```

If TXCIE interrupt is enabled then transmit complete interrupt triggers ISR.

UART initialization

UART initialization

```
//Function To Initialize UART1
// desired baud rate=9600
// actual baud rate=9600 (error 0.0
// char size=8 bit
// parity=Disabled
// stop bit=1
void uart0_init(void)
{
 UCSRB = 0x00;//disable while setting baud rate
 UCSRA = 0x00;
 UCSRC = 0x86;
 UBRRL = 0x2F;//set baud rate lo
 UBRRH = 0x00;//set baud rate hi
 UCSRB = 0x98;
}
```

Demonstration

Thank You!

Post your queries on: helpdesk@e-yantra.org