CSE 315
Microprocessors & Microcontrollers

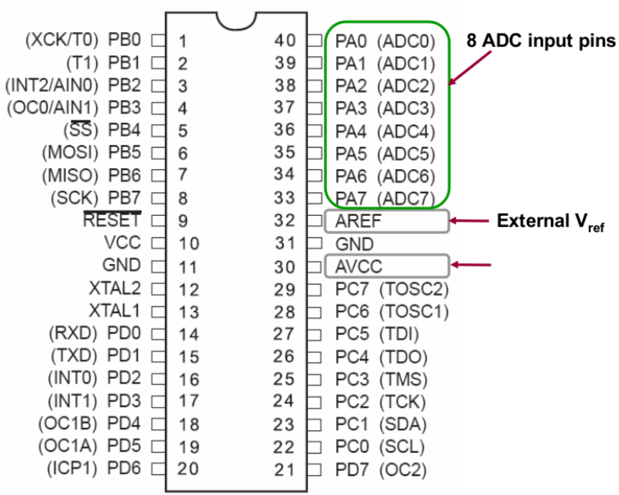
Tanvir Ahmed Khan

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology.

October 25, 2014

Recap

ATmega16/32 ADC Relevant Pin Diagram



ATmega16/32 ADC Features

- 10-bit ADC
- 2 output registers,
  - `ADCH:ADCL`
  - 16-bits, 6-bits are unused
  - Option to adjust left or right
- 8 Analog input channels,
  - 7 differential input
  - with optional gain of 10x & 200x
  - However, only one conversion at a time
- V_{ref} options,
  - Analog V_{cc}, $5V$
  - internal $2.56V$
  - external $AREF$ pin
- ADC clock rate != MCU CPU clock rate
  - selection of pre-scaler
  - AD Conversion takes at-least 13 ADC clock cycles

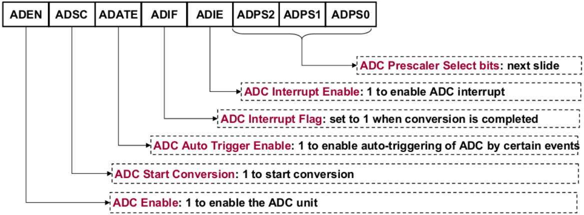
ADC Programming

Major Relevant registers

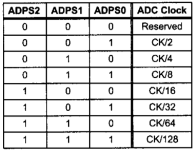
- `ADCH:ADCL`
- `ADCSRA`
- `ADMUX`
- `SFIOR`

ADC Programming

ADCSRA Register



ADEN
ADSC
ADATE
ADIF
ADIE
ADPS2
ADPS1
ADPS0
ADC Prescaler Select bits: next slide
ADC Interrupt Enable: 1 to enable ADC interrupt
ADC Interrupt Flag: set to 1 when conversion is completed
ADC Auto Trigger Enable: 1 to enable auto-triggering of ADC by certain events
ADC Start Conversion: 1 to start conversion
ADC Enable: 1 to enable the ADC unit

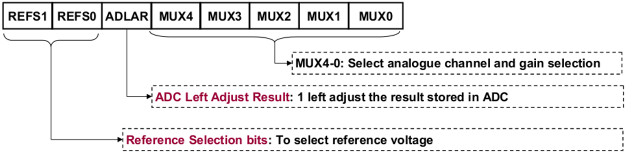


ADPS2	ADPS1	ADPS0	ADC Clock
0	0	0	Reserved
0	0	1	CK/2
0	1	0	CK/4
0	1	1	CK/8
1	0	0	CK/16
1	0	1	CK/32
1	1	0	CK/64
1	1	1	CK/128

- Initialization,
  - Polling, `ADCSRA = 0b10000001;`
  - Interrupt, `ADCSRA = 0b10001001;`
- Conversion Start,
  - `ADCSRA = ADCSRA | 0b01000000;`

ADC Programming

ADMUX Register



REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

MUX4-0: Select analogue channel and gain selection

ADC Left Adjust Result: 1 left adjust the result stored in ADC

Reference Selection bits: To select reference voltage

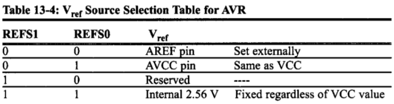
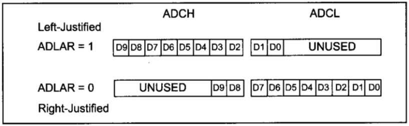


Table 13-4: V_{ref} Source Selection Table for AVR

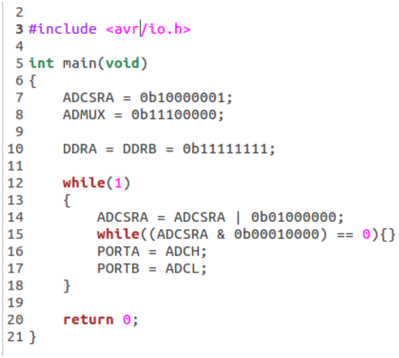
REFS1	REFS0	V_{ref}	
0	0	AREF pin	Set externally
0	1	AVCC pin	Same as VCC
1	0	Reserved	----
1	1	Internal 2.56 V	Fixed regardless of VCC value



	ADCH	ADCL
Left-Justified ADLAR = 1	D9 D8 D7 D6 D5 D4 D3 D2	D1 D0 UNUSED
ADLAR = 0 Right-Justified	UNUSED D9 D8	D7 D6 D5 D4 D3 D2 D1 D0

Sample ADC Program

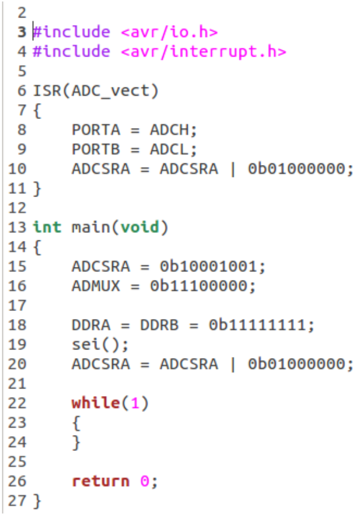
Polling



```c

#include <avr/io.h>

int main(void)
{
    ADCSRA = 0b10000001;
    ADMUX = 0b11100000;

    DDRA = DDRB = 0b11111111;

    while(1)
    {
        ADCSRA = ADCSRA | 0b01000000;
        while((ADCSRA & 0b00010000) == 0){}
        PORTA = ADCH;
        PORTB = ADCL;
    }

    return 0;
}
```

Sample ADC Program

Interrupt



```c

#include <avr/io.h>
#include <avr/interrupt.h>

ISR(ADC_vect)
{
    PORTA = ADCH;
    PORTB = ADCL;
    ADCSRA = ADCSRA | 0b01000000;
}

int main(void)
{
    ADCSRA = 0b10001001;
    ADMUX = 0b11100000;

    DDRA = DDRB = 0b11111111;
    sei();
    ADCSRA = ADCSRA | 0b01000000;

    while(1)
    {
    }

    return 0;
}
```

Today's Topic

ATmega16 Serial Communications

Data Communications

Data Communications

Data Communications

- Computers transfer data in 2 ways,

Data Communications

- Computers transfer data in 2 ways,
  - Parallel Communication

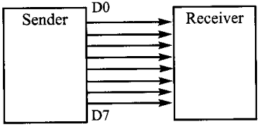
Data Communications

- Computers transfer data in 2 ways,
  - Parallel Communication
  - Serial Communication

Parallel Data Communications

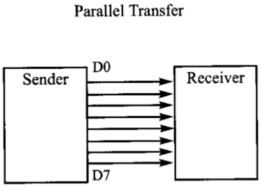
Parallel Data Communications

Parallel Transfer



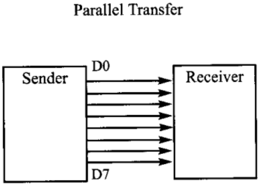
- multiple wire lines are used to transfer data

Parallel Data Communications



- multiple wire lines are used to transfer data
- Advantage,

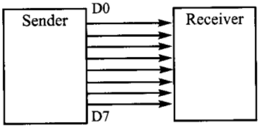
Parallel Data Communications



- multiple wire lines are used to transfer data
- Advantage,
  - Speed

Parallel Data Communications

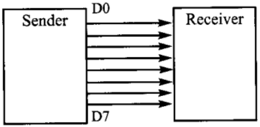
Parallel Transfer



- multiple wire lines are used to transfer data
- Advantage,
  - Speed
- Disadvantage,

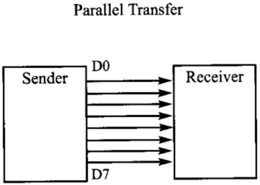
Parallel Data Communications

Parallel Transfer



- multiple wire lines are used to transfer data
- Advantage,
  - Speed
- Disadvantage,
  - Distance cannot be great

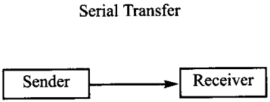
Parallel Data Communications



- multiple wire lines are used to transfer data
- Advantage,
  - Speed
- Disadvantage,
  - Distance cannot be great
- Example, computer to printer data transfer

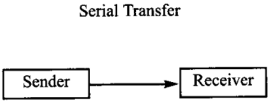
Serial Data Communications

Serial Data Communications



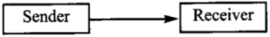
- data is sent one bit at a time

Serial Data Communications



- data is sent one bit at a time
- Advantage,

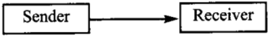
Serial Data Communications


Serial Transfer

- data is sent one bit at a time
- Advantage,
  - larger distances

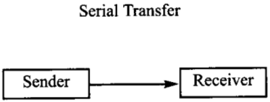
Serial Data Communications

Serial Transfer



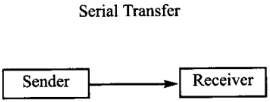
- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper

Serial Data Communications



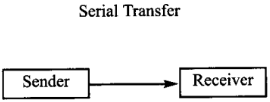
- data is sent one bit at a time
- Advantage,
    - larger distances
    - cheaper
    - fewer I/O pins

Serial Data Communications



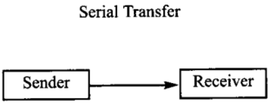
- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper
  - fewer I/O pins
  - easy synchronization

Serial Data Communications



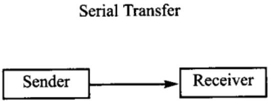
- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper
  - fewer I/O pins
  - easy synchronization
- Disadvantage,

Serial Data Communications



- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper
  - fewer I/O pins
  - easy synchronization
- Disadvantage,
  - relatively slower

Serial Data Communications



- data is sent one bit at a time
- Advantage,
  - larger distances
  - cheaper
  - fewer I/O pins
  - easy synchronization
- Disadvantage,
  - relatively slower
- Example, USB

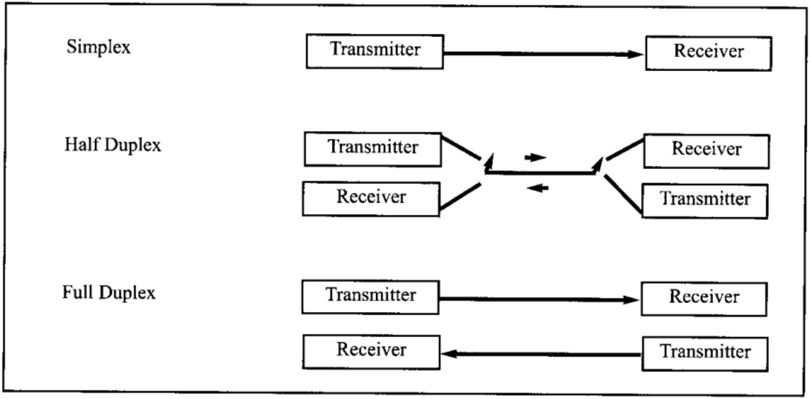
Synchronization in Serial Communications

Two Methods

- Synchronous method,
  - transfers a block of data at a time

- Asynchronous method,
  - transfers a single byte at a time

Simplex, Half- & Full-Duplex Data Transfer



Simplex

Transmitter

Receiver

Half Duplex

Transmitter

Receiver

Receiver

Transmitter

Full Duplex

Transmitter

Receiver

Receiver

Transmitter

Asynchronous Serial Communication

Data Framing

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
    - placing each character between start & stop bits
- *Start bit*
    - always one bit

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*
  - always one bit
  - always 0(low)

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*
  - always one bit
  - always 0(low)
- *Stop bit*

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
    - placing each character between start & stop bits
- *Start bit*
    - always one bit
    - always 0(low)
- *Stop bit*
    - can be one or two bits

Asynchronous Serial Communication

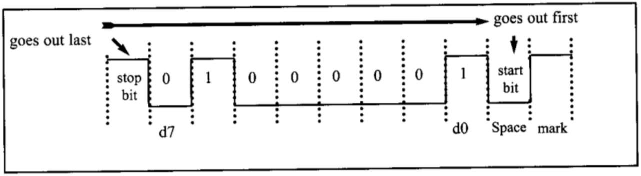
Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*
  - always one bit
  - always 0(low)
- *Stop bit*
  - can be one or two bits
  - always 1(high)

Asynchronous Serial Communication

Data Framing

- character-oriented data transfer
- *Framing*
  - placing each character between start & stop bits
- *Start bit*
  - always one bit
  - always 0(low)
- *Stop bit*
  - can be one or two bits
  - always 1(high)



Framing of 'A'(0x41)

Data Transfer Rate

Data Transfer Rate

- *bps*
    - bits per second

Data Transfer Rate

- *bps*
  - bits per second
- *baud rate*
  - number of signal changes per second

Data Transfer Rate

- *bps*
  - bits per second
- *baud rate*
  - number of signal changes per second
- generally, bps != baud rate

Data Transfer Rate

- *bps*
  - bits per second
- *baud rate*
  - number of signal changes per second
- generally, bps != baud rate
  - for some coding system, bps == baud rate

RS232 Standards

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**
  - $0 = +3$ to $+25$ V

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**
  - $0 = +3$ to $+25$ V
  - $1 = -3$ to -25 V

RS232 Standards

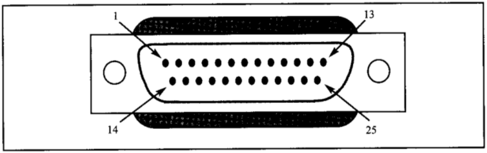
- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**
  - $0 = +3$ to $+25$ V
  - $1 = -3$ to -25 V
- we will need voltage converter

RS232 Standards

- allow compatibility among data communication equipments of various manufacturers
- initially set in 1960s
- PC COM ports supports this Standard
- **not compatible with TTL family**
  - $0 = +3$ to $+25$ V
  - $1 = -3$ to -25 V
- we will need voltage converter
  - MAX232

RS232 Standards

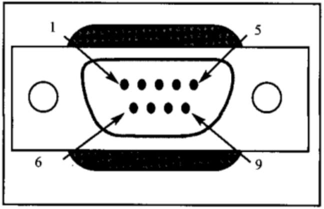
Continued



Original RS232 Connector DB-25

RS232 Standards

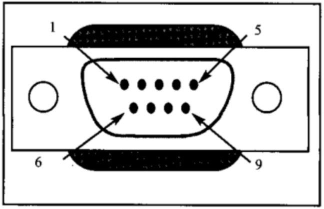
Continued



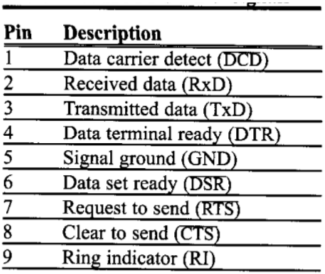
9-pin Connector for DB-9

RS232 Standards

Continued



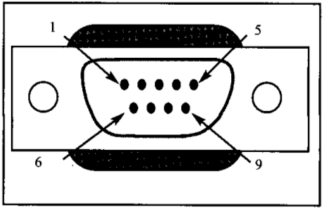
9-pin Connector for DB-9



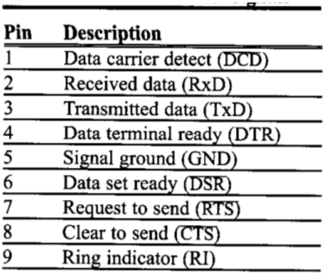
Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)

RS232 Standards

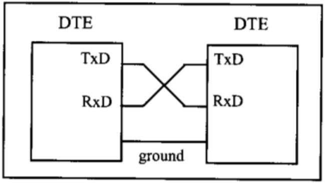
Continued



9-pin Connector for DB-9



Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)



Reference

- The avr microcontroller & embedded system, *Chapter 11*
  - Muhammad Ali Mazidi
  - Sarmad Naimi
  - Sepehr Naimi