Smart Home AVR

Submitted by:
Eng: Mohamed Fadl Sholkamy
Eng: Abdulrahman Mohamed Elmahdi
Eng: Mahmoud Mustafa Mahmoud Mustafa
Group: 501

component

1. DIO
2. LCD
3. USART
1. SPI
2. ADC
3. TIMER

1. LM35
2. LED
3. DC MOTOR
4. KPD

DIO

ATmega32 has programmable I/O lines divided into 4ports(groups)

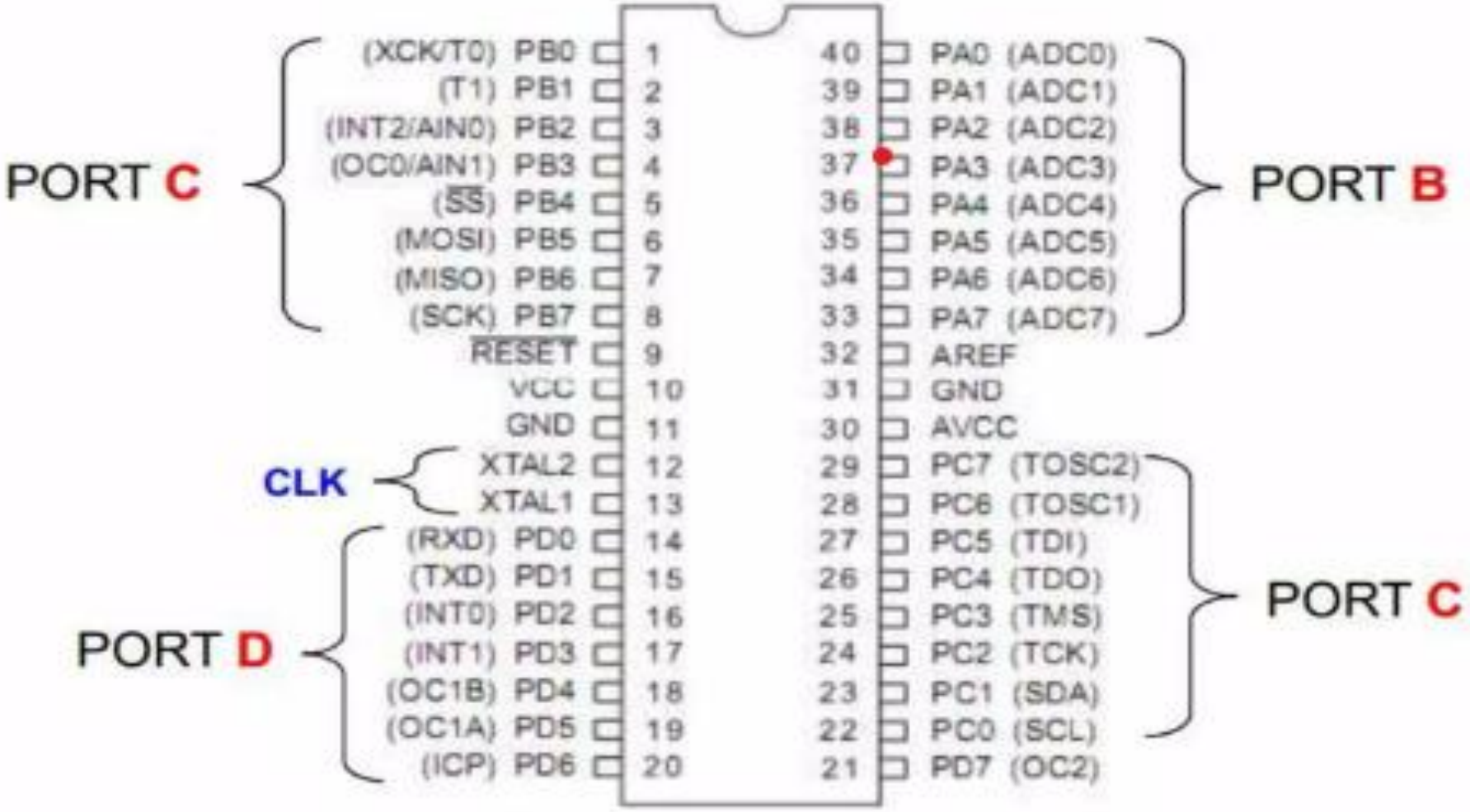
1-PORTA (PA7……..PA0)
1-PORTB (PB7……..PB0)
1-PORTC (PC7……..PC0)
1-PORTD (PD7……..PD0)

Each PORT is controllel by 3 Registers

1-DDRx (Data Direction Registers)
2-PORTx (Output Registers)
3-PINx (input Registers)

❖ ATMEGA 32 and DIO (cont.):

ATmega 32 pin out



$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	64
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	64
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	64
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	64
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	64
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65

LCD

LCD stands for Liquid Crystal Display, is an electronic device which is used for data display. LCDs are preferable over seven segments and LEDs as they can easily represent data in form of alphabets, characters
, numbers or animations. LCDs are very easy to program and make your work quite attractive and simple. Numerous types of LCDs are available in MARKET
such as 16X2, 16X4, 20X2, 20X4, graphical LCDs (128X64) etc. The LCD which we are using is 16X2 alphanumeric LCD, it display 32 characters in two rows means in one row we have 16 characters.

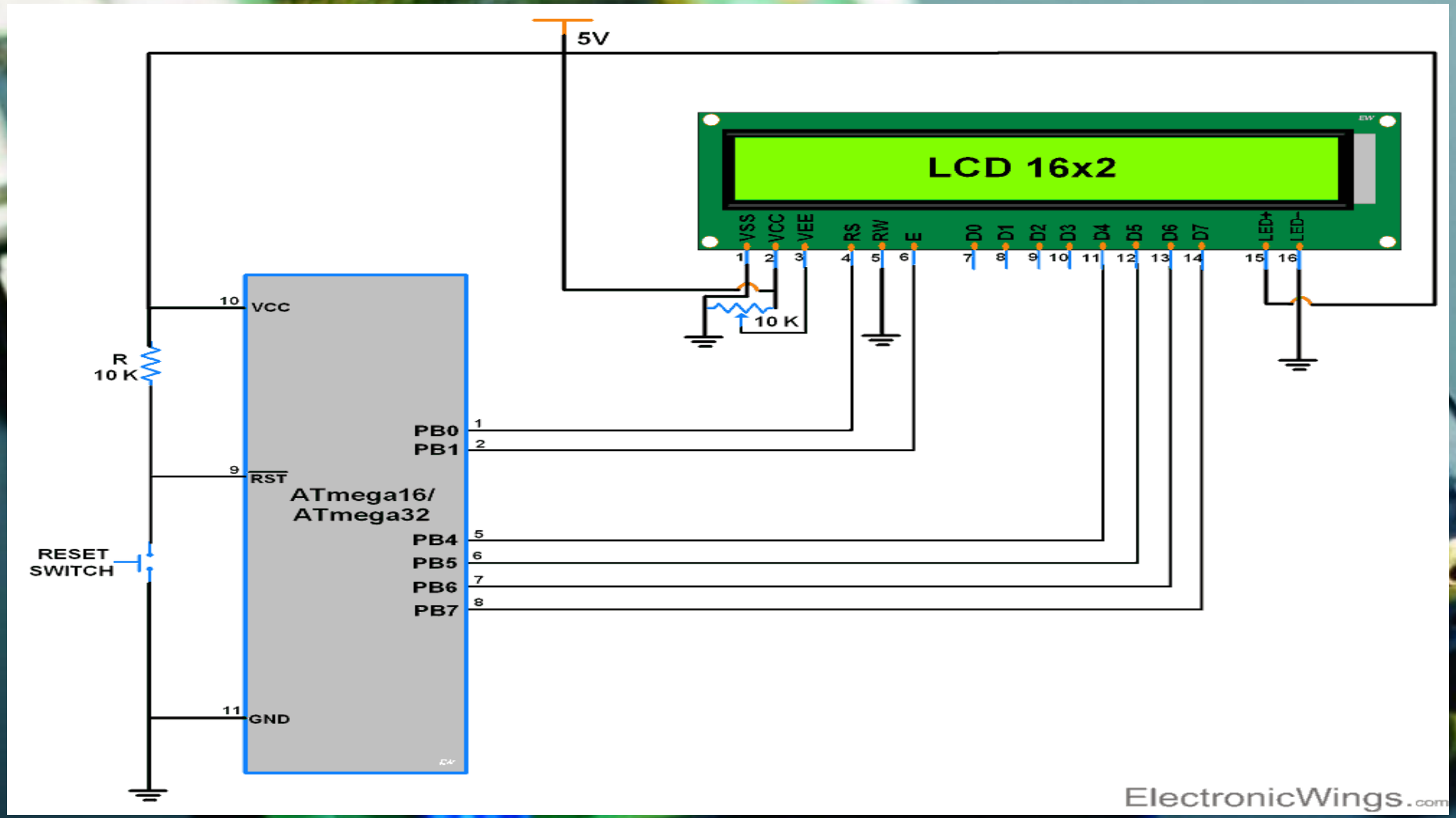
PIN description of LCD

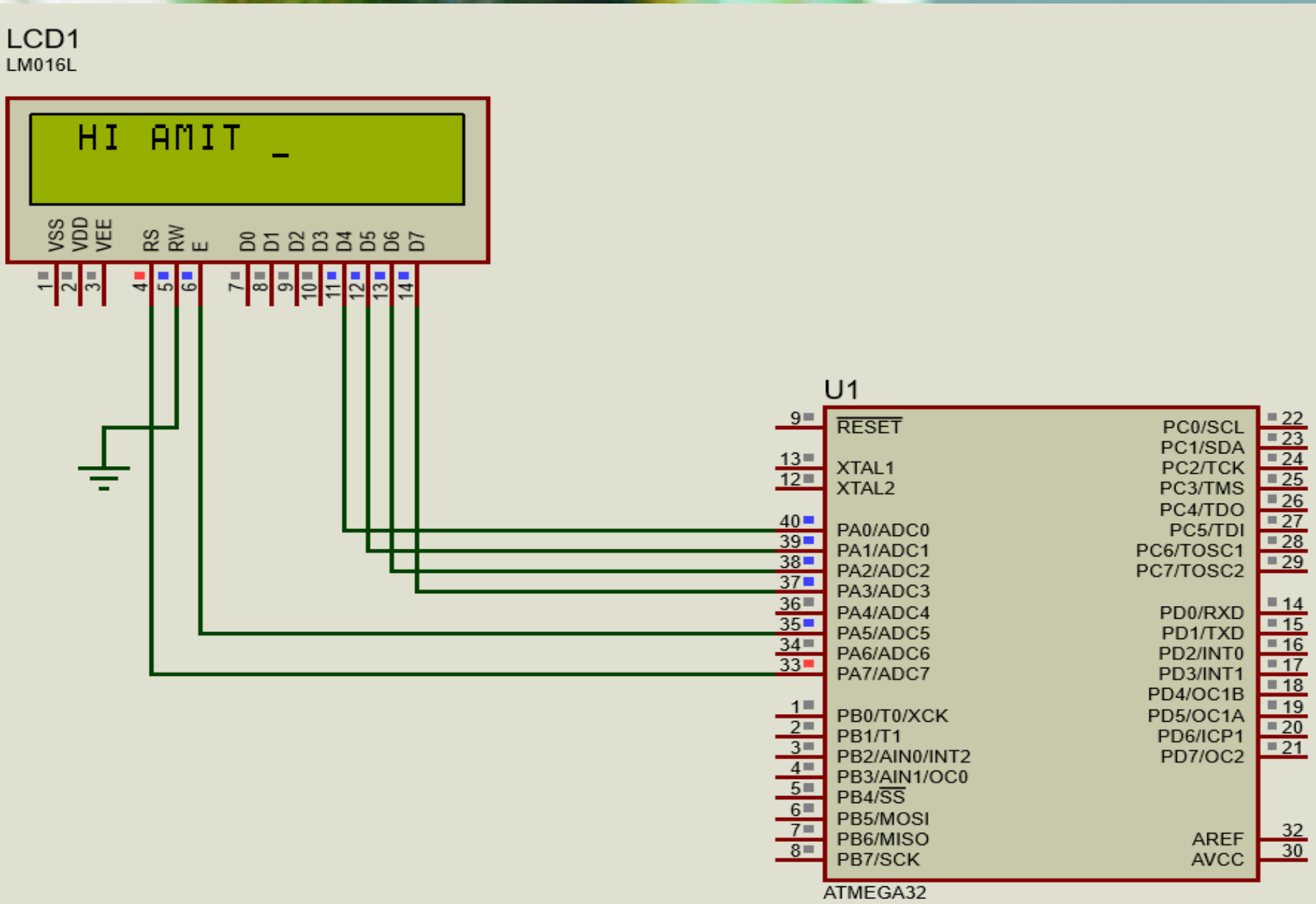
- 16X2 LCD can interface with AVR microcontroller by using two modes, 4-bit mode or 8-bit mode. In this article we will use 8-bit mode for interfacing. In 8-bit mode we send command to LCD by using eight data lines (D0-D7) while in 4-bit mode we use four data lines (D5-D7) for sending command and data. These data lines can be connected to any port of Atmega32.

PIN NO.	PIN NAME	DESCRIPTION
1	V_{CC}	Supply pin (+5V DC)
2	V_{DD}	Ground pin
3	V_{EE}	Contrast pin
4	RS	Register selection pin (either data or command)RS=0: Command Register , RS=1: Data Register
5	RW	Selects Read or Write operationRW=0: for write RW=1: for read
6	E	Enable pin
7	D0	Data pin 0

8	D1	Data pin 1
9	D2	Data pin 2
10	D3	Data pin 3
11	D4	Data pin 4
12	D5	Data pin 5
13	D6	Data pin 6
14	D7	Data pin 7


VSS Ground
VCC +5V
VEE Contrast control
RS Register select
RW Read/write
E Enable
D0 Data Pin 0
D1 Data Pin 1
D2 Data Pin 2
D3 Data Pin 3
D4 Data Pin 4
D5 Data Pin 5
D6 Data Pin 6
D7 Data Pin 7
LED+ LED+ 5V
LED- LED- Ground


5V
LCD 16x2
VSS
VCC
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
LED+
LED-
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
10 K
R
10 K
10
VCC
PB0
PB1
9
RST
ATmega16/
ATmega32
PB4
PB5
PB6
PB7
RESET
SWITCH
11
GND
ElectronicWings.com


LCD1
LM016L
HI AMIT _
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14
U1
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
ATMEGA32

USART

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly flexible serial communication device. The main features are:

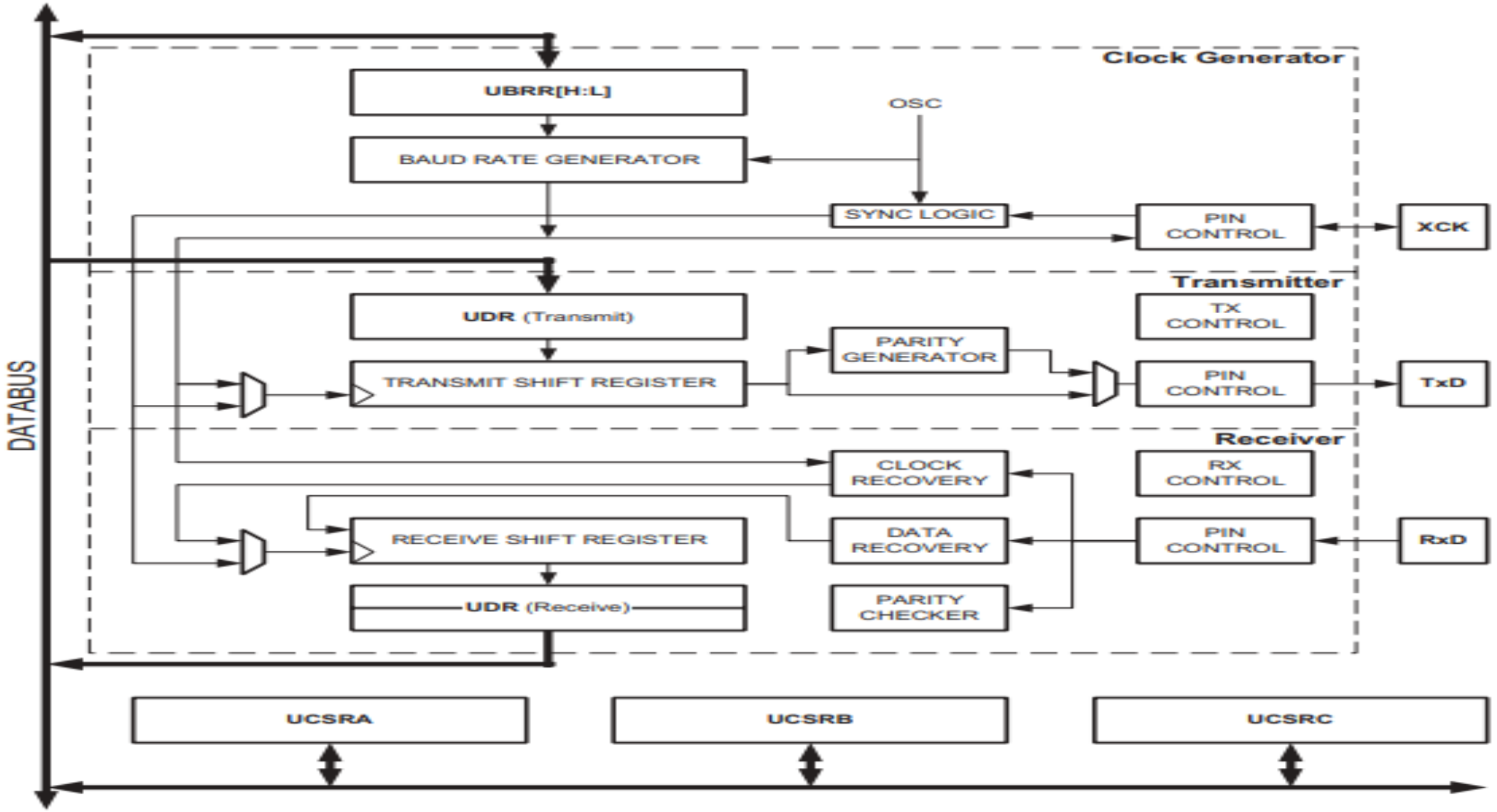
- Full Duplex Operation (Independent Serial Receive and Transmit Registers) • Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter • Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

UART Registers

- UDR (UART Data Register)
  - Write bytes to transmit
  - Read received bytes
- UCSRA (UART Status Register)
  - Rx/Tx complete signal bits
  - Framing error, overflow signal bits
- UCSRB (UART Control Register)
  - Interrupt enable bits
  - Rx/Tx enable bits
  - Data format control bits
- UBRRL (UART Baud Rate Register)
  - Baud rate generator division ratio

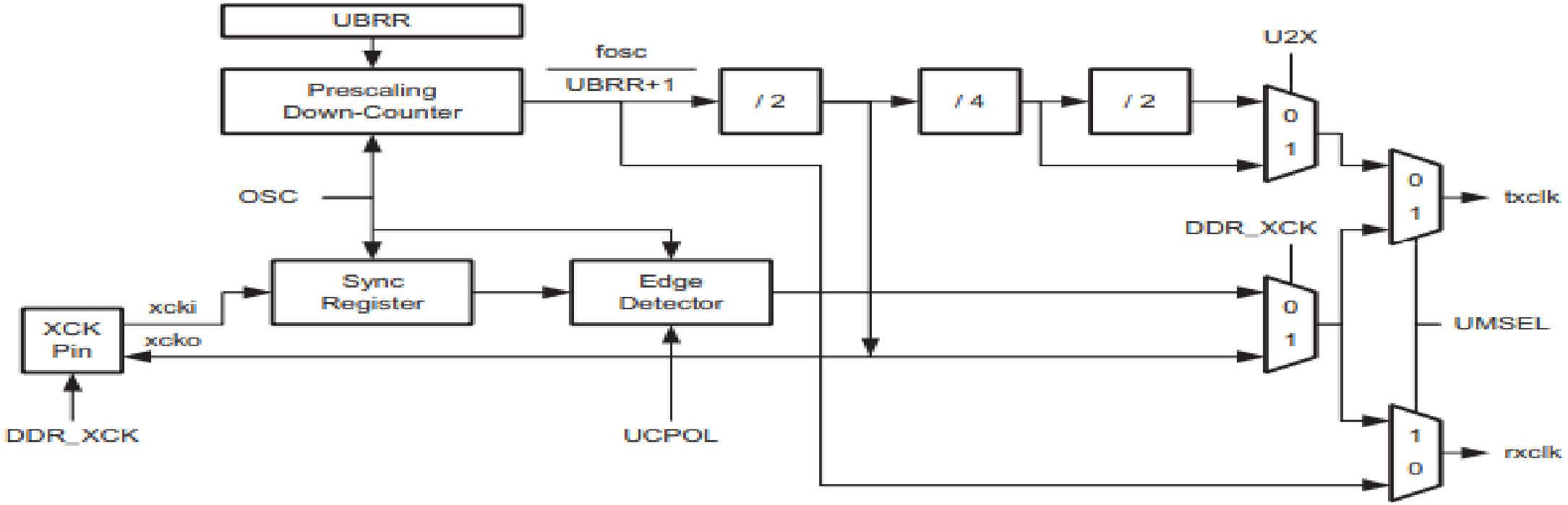
$0C ($2C)	UDR	USART I/O Data Register								157
$0B ($2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	158
$0A ($2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	159
$09 ($29)	UBRRL	USART Baud Rate Register Low Byte								162

Figure 69. USART Block Diagram[1]



Note: 1. Refer to Figure 1 on page 2, Table 33 on page 62, and Table 27 on page 57 for USART pin placement.

Figure 70. Clock Generation Logic, Block Diagram



Signal description:

txclk Transmitter clock (Internal Signal).

rxclk Receiver base clock (Internal Signal).

xcki Input from XCK pin (Internal Signal). Used for synchronous slave operation.

xcko Clock output to XCK pin (Internal Signal). Used for synchronous master operation.

fosc XTAL pin frequency (System Clock).

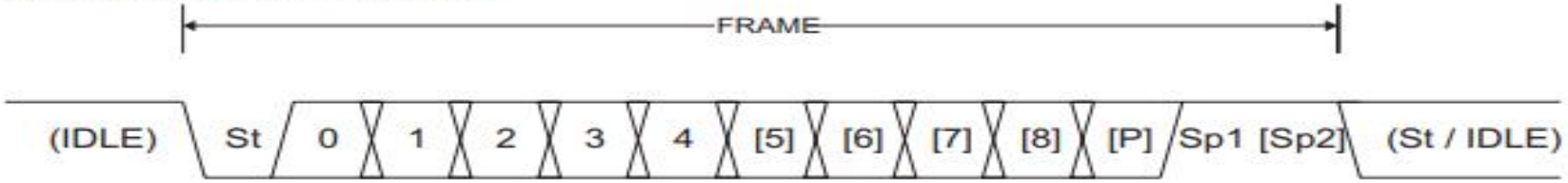
Frame Formats

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error checking. The USART accepts all 30 combinations of the following as valid frame formats:

- 1 start bit
- 5, 6, 7, 8, or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit. Then the next data bits, up to a total of nine, are succeeding, ending with the most significant bit. If enabled, the parity bit is inserted after the data bits, before the stop bits. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle (high) state. Figure 72 illustrates the possible combinations of the frame formats. Bits inside brackets are optional.

Figure 72. Frame Formats



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.

The frame format used by the USART is set by the UCSZ2:0, UPM1:0, and USBS bits in UCSRB and UCSRC. The Receiver and Transmitter use the same setting. Note that changing the setting of any of these bits will corrupt all ongoing communication for both the Receiver and Transmitter.

Table 60. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{OSC} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

Some examples of UBRR values for some system clock frequencies are found in Table 68 (see page 163).

SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the ATmega32 and peripheral devices or between several AVR devices. The ATmega32 SPI includes the following features:

- Full-duplex, Three-wire Synchronous Data Transfer
- Master or Slave Operation
- LSB First or MSB First Data Transfer
- Seven Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection • Wake-up from Idle Mode
- Double Speed (CK/2) Master SPI Mode

Table 55. SPI Pin Overrides

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
$\overline{SS}$	User Defined	Input

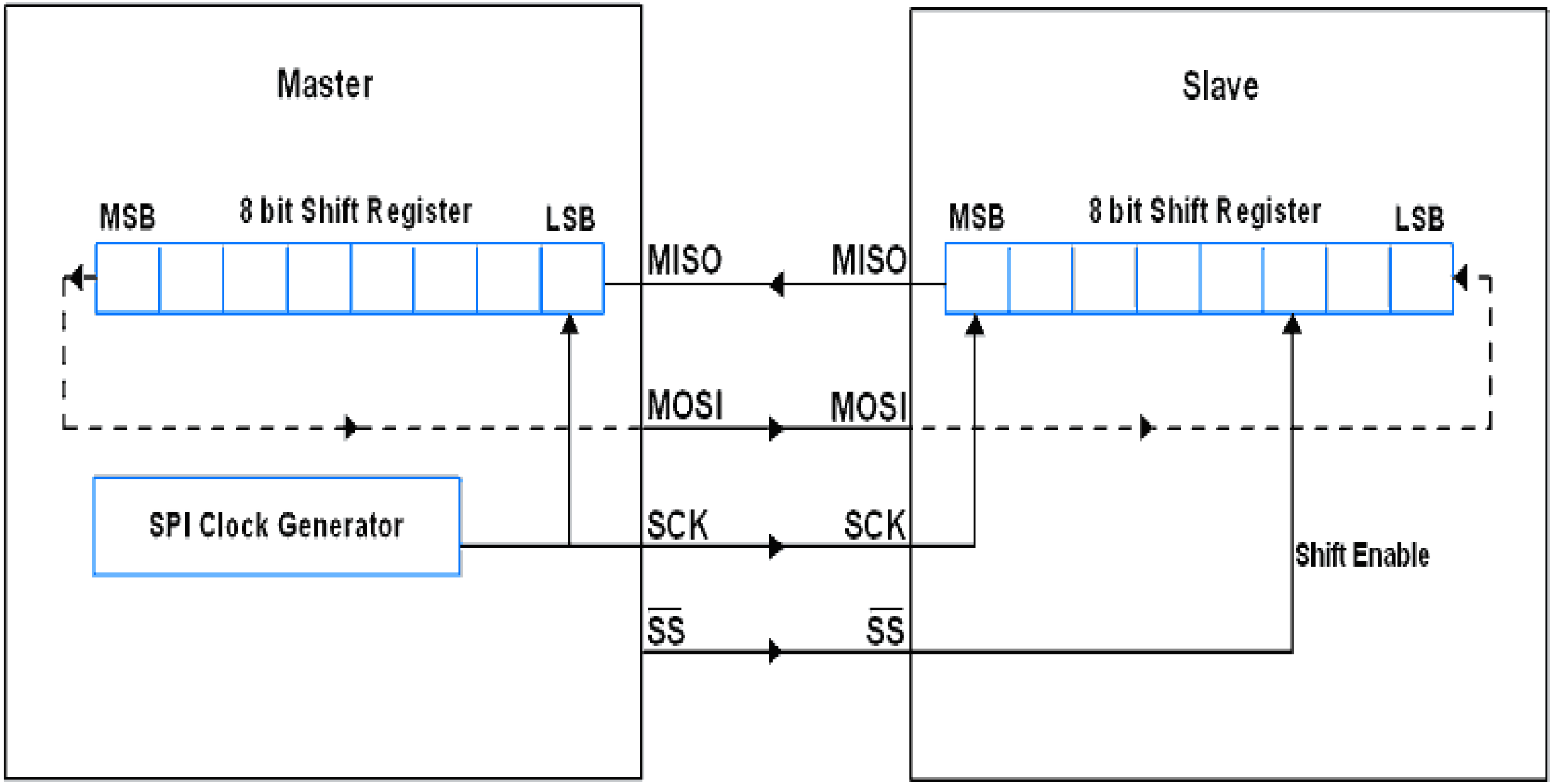
Note: See "Alternate Functions of Port B" on page 55 for a detailed description of how to define the direction of the user defined SPI pins.

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a Master. SPR1 and SPR0 have no effect on the Slave. The relationship between SCK and the Oscillator Clock frequency f_{osc} is shown in the following table:

Table 58. Relationship Between SCK and the Oscillator Frequency

SPI2X	SPR1	SPR0	SCK Frequency
0	0	0	$f_{osc}/4$
0	0	1	$f_{osc}/16$
0	1	0	$f_{osc}/64$
0	1	1	$f_{osc}/128$
1	0	0	$f_{osc}/2$
1	0	1	$f_{osc}/8$
1	1	0	$f_{osc}/32$
1	1	1	$f_{osc}/64$

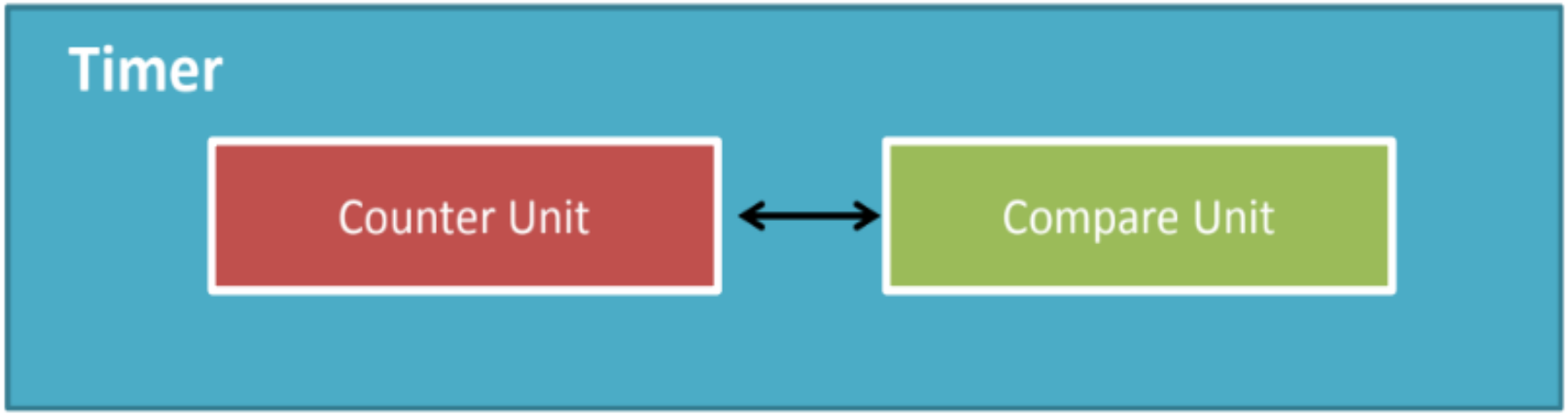


SPI Master Slave Interconnection

TIMER

Timers internal Architecture

The internal architecture of the AVR Timers can be simplified as follow



AVR Atmage16 has 3 Timers

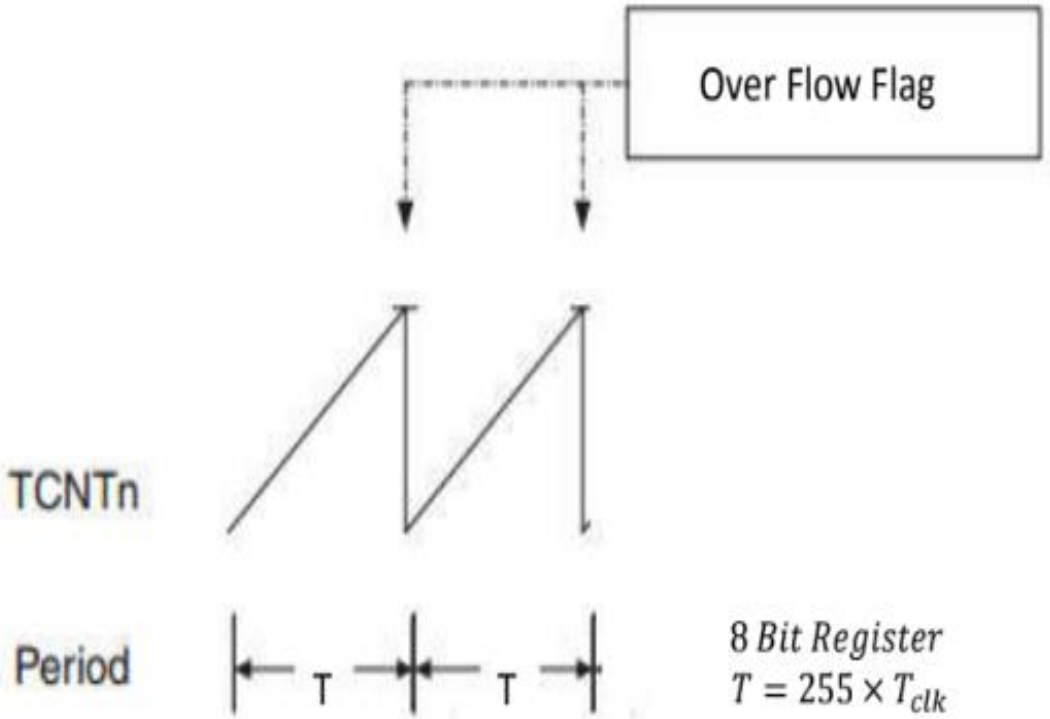
- Timer0 : 8 Bit synchronous Timer
- Timer1 : 16 Bit synchronous Timer
- Timer2 : 8 Bit Asynchronous Timer

Each Timer has Two Modes of Operation Which can be used For system timing

- Overflow Mode (Normal Mode)
- Compare Mode (CTC mode)

Overflow Mode(Normal Mode)

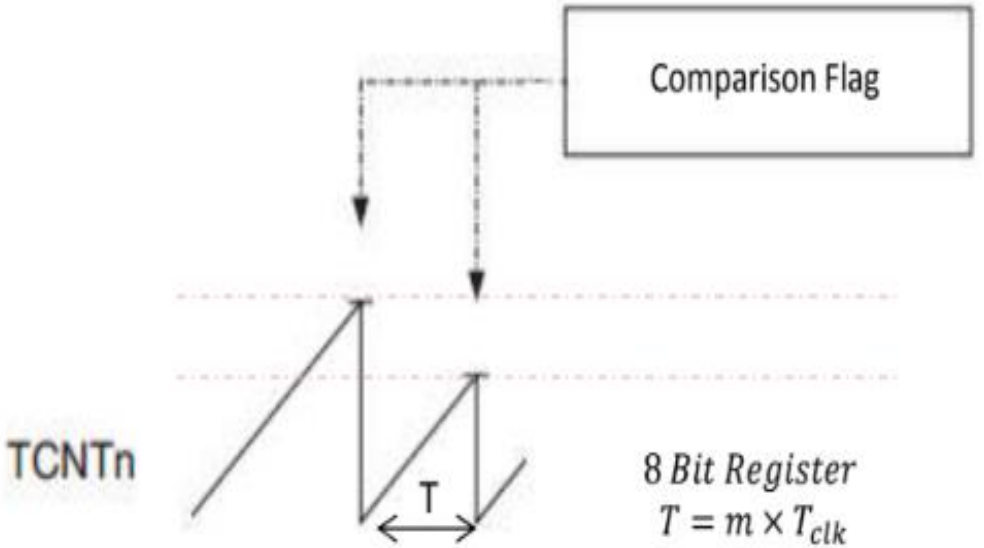
Timing based on the Register overflow



The following mode has Low resolution

Compare Output Mode(CTC Mode)

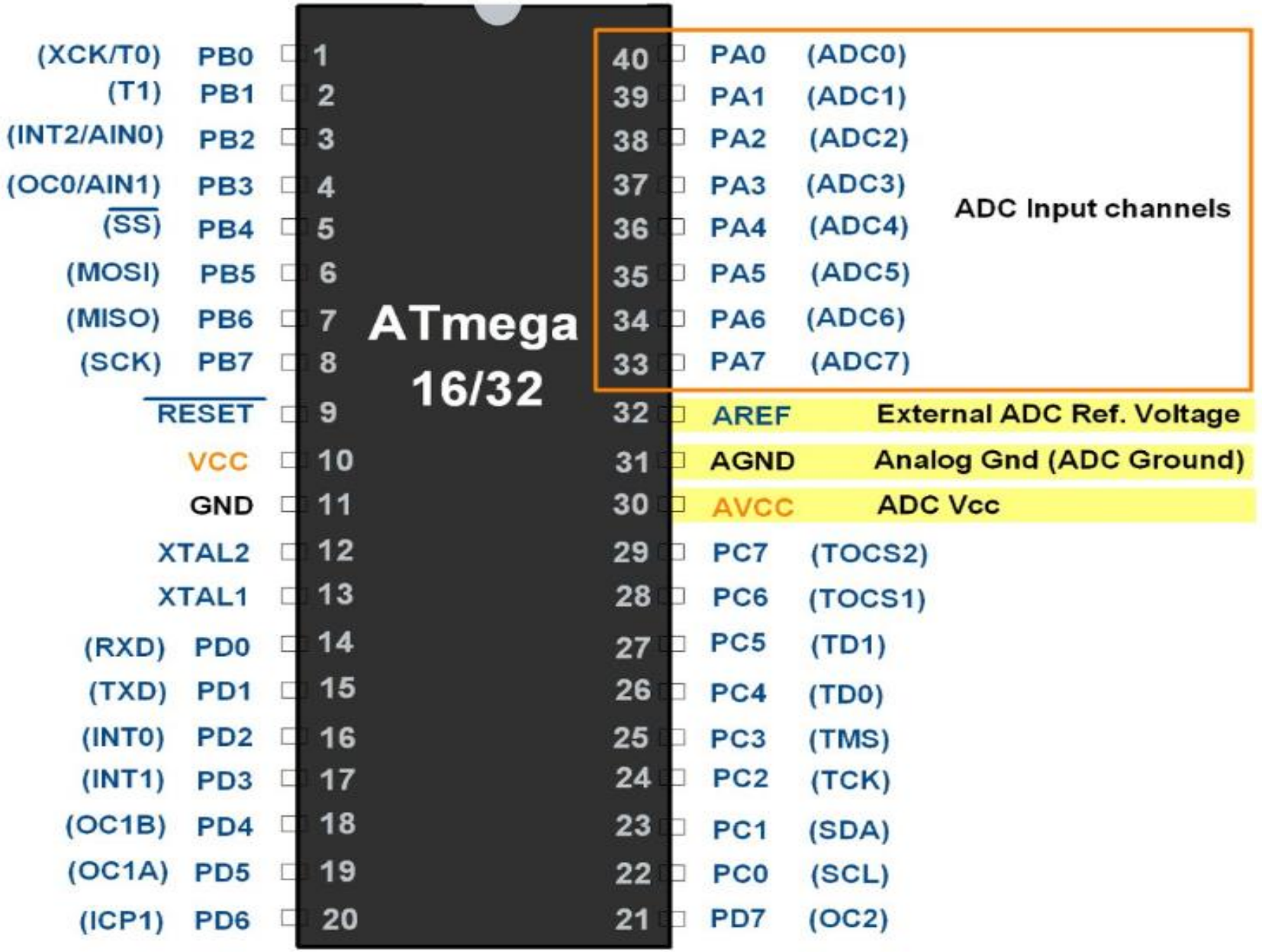
Timing based on the Comparison between Two Values



The following mode has High resolution

ADC

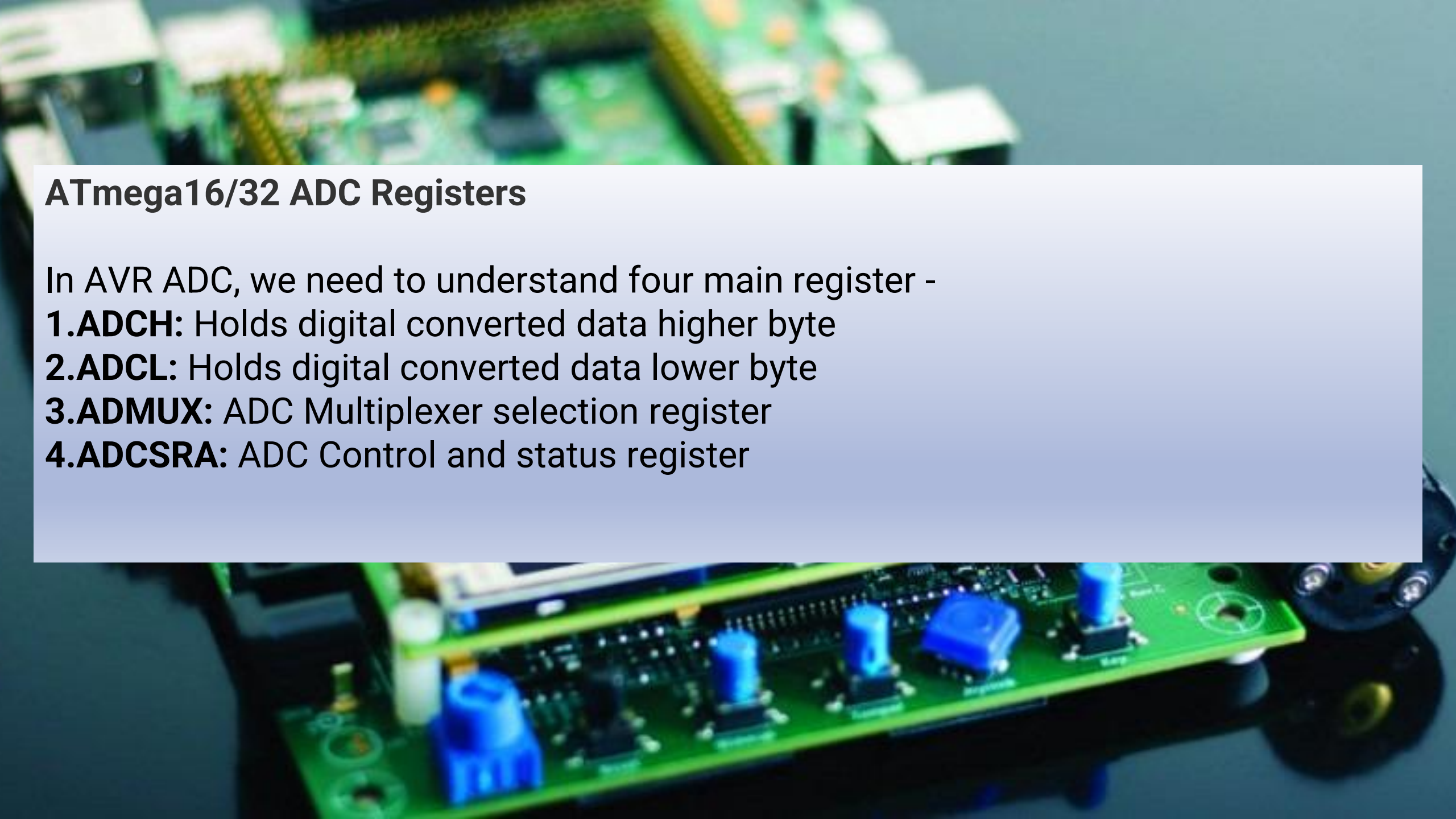
ADC (Analog to Digital converter) is the most widely used device in embedded systems which is designed especially for data acquisition. In the AVR ATmega series normally 10-bit ADC is inbuilt in the controller. Let us see how to use the ADC of AVR ATmega16 / ATmega32. ATmega16/32 supports eight ADC channels, which means we can connect eight analog inputs at a time. ADC channel 0 to channel 7 are present on PORTA. i.e. Pin no.33 to 40.


(XCK/T0) PB0 1
(T1) PB1 2
(INT2/AIN0) PB2 3
(OC0/AIN1) PB3 4
(SS) PB4 5
(MOSI) PB5 6
(MISO) PB6 7
(SCK) PB7 8
RESET 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
(RXD) PD0 14
(TXD) PD1 15
(INT0) PD2 16
(INT1) PD3 17
(OC1B) PD4 18
(OC1A) PD5 19
(ICP1) PD6 20
ATmega 16/32
40 PA0 (ADC0)
39 PA1 (ADC1)
38 PA2 (ADC2)
37 PA3 (ADC3)
36 PA4 (ADC4)
35 PA5 (ADC5)
34 PA6 (ADC6)
33 PA7 (ADC7)
ADC Input channels
32 AREF External ADC Ref. Voltage
31 AGND Analog Gnd (ADC Ground)
30 AVCC ADC Vcc
29 PC7 (TOCS2)
28 PC6 (TOCS1)
27 PC5 (TD1)
26 PC4 (TD0)
25 PC3 (TMS)
24 PC2 (TCK)
23 PC1 (SDA)
22 PC0 (SCL)
21 PD7 (OC2)

The controller has 10 bit ADC, which means we will get digital output 0 to 1023.
i.e. When the input is 0V, the digital output will be 0V & when input is 5V (and Vref=5V), we will get the highest digital output corresponding to 1023 steps, which is 5V.
So controller ADC has 1023 steps and

- Step size with Vref=5V : 5/1023 = 4.88 mV.
- Step size with Vref=2.56 : 2.56/1023 = 2.5 mV.

So Digital data output will be Dout = Vin / step size.



ATmega16/32 ADC Registers

In AVR ADC, we need to understand four main register -

1. **ADCH:** Holds digital converted data higher byte
2. **ADCL:** Holds digital converted data lower byte
3. **ADMUX:** ADC Multiplexer selection register
4. **ADCSRA:** ADC Control and status register

Simulation Project Smart Home