WINTER WORKSHOP DOCUMENTATION

DAY 1,8/12/15

The winter workshop is being conducted by Siddharth Jha,Rahul Kumar and Aashay Pedder.They gave us a basic idea about the problem statement and the concepts involved to make it.

PROBLEM STATEMENT: Wireless step counter and heading follower robot

TOPICS COVERED ON DAY 1

- Different components of the BOT
- Types of sensors
- Acclerometer(Ball in box concept)
- Basic outline on L-293D
- Bitwise operation
- Boolean alegbra
- Power Electronics

Differnt Components of the Bot

The bot will broadly comprise of four parts:

1. Motor,differential drive and chasis
2. Integrated Circuit,micro-controller
3. Wireless transmitter and receiver
4. Sensors

Types of Sensor

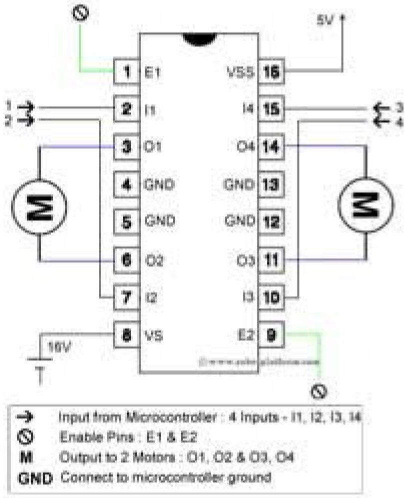
1. Digital Compass
2. Accelerometer
3. Gyrocope

Dynamics of Accelerometer

Dynamics of accelerometer can easily be understood by ball in box analogy.A ball is kept in the box and the box is given an acceleration.The force that the wall experiences due to the box,divided by mass of the ball will give us the acceleration of ball in box system. Accelerometer also follows similar kind of principle and give its output in Volts.

L-293D

L293D IC generally comes as a standard 16-pin DIP (dual-in line package). This motor driver IC can simultaneously control two small motors in either direction; forward and reverse with just 4 microcontroller pins (if you do not use enable pins)



5V
1 E1
VSS 16
2 I1
I4 15
3 O1
O4 14
4 GND
GND 13
5 GND
GND 12
6 O2
O3 11
7 I2
I3 10
8 VS
E2 9
16V
1
2
3
4
M
M
→ Input from Microcontroller : 4 Inputs - I1, I2, I3, I4
⦸ Enable Pins : E1 & E2
M Output to 2 Motors : O1, O2 & O3, O4
GND Connect to microcontroller ground

TRUTH TABLE

Pin 1	Pin 2	Pin 7	Function
High	**High**	**Low**	**Turn Anti-clockwise (Reverse)**
High	**Low**	**High**	**Turn clockwise (Forward)**
High	**High**	**High**	**Stop**
High	**Low**	**Low**	**Stop**
Low	**X**	**X**	**Stop**

High Input=5 volts
Low Input=0 volts

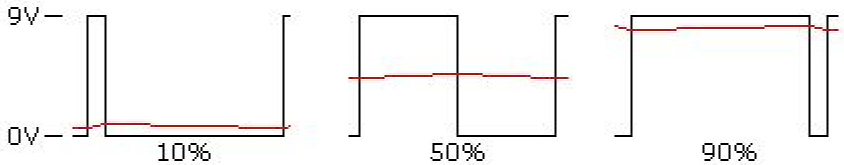
FAQ's

Q 1) Why four terminals are given for grounding ?

Ans 1) The four terminals together act as a heat sink.So it prevents the rise of temperatrure.Therefore it is recommended to short all 4 ground terminals externally.

PULSE RATE MODULATION

It is a technique generally used to control the rpm of a motor.By applying dc voltage in the form of a wave we can actually set the speed of a motor.This method is far more accurate then the voltage varying method.



Examples of PWM Waveforms

FAQ's

Q 1)Why the above voltage is given to E1 or E2 and not I1 or I2?
Ans 1)By giving above voltage waveform to I1 or I2 we are forcing 0 voltage drop across the motor when no voltage is applied.So motor will stop immediately.This will occur as many times as the frequency of the wave.So to stop this voltage is given to E1 or E2. The motor will then have floating voltage when no voltage is applied and will run freely.

BITWISE OPERATION

Symbolic representation of the operators:

&	AND
|	OR
^	XOR
~	COMP
<<	LEFT SHIFT
>>	RIGHT SHIFT

POWER ELECTRONICS

The supply that we get is generally ac.So to convert it into constant dc source, the flowchart shown below is followed:-

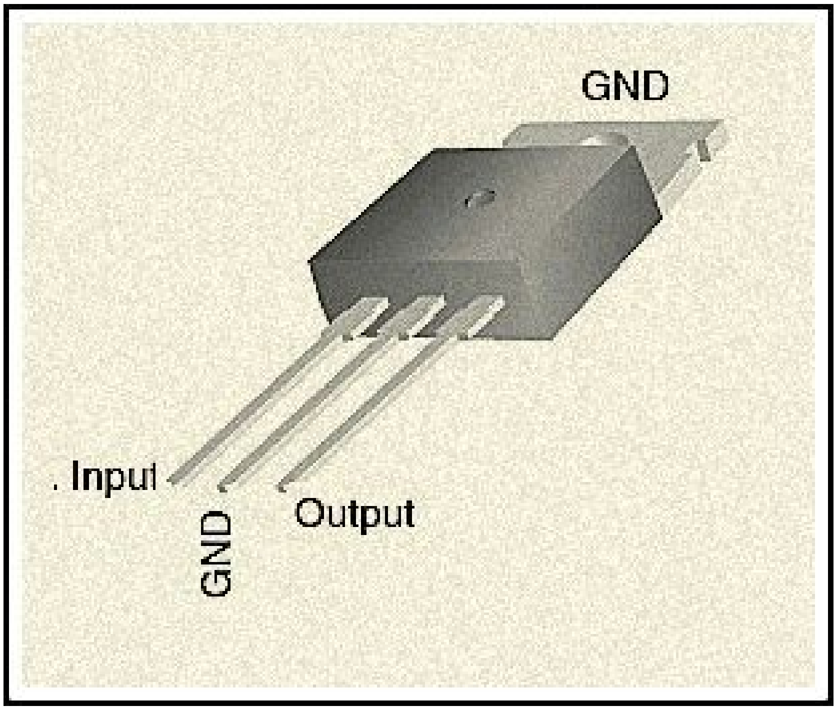
220V---------->Step down   ----------->Rectifier----------->Filter-------->Voltage Regulator-------->IC
transformer

Types of rectifiers:

1) Half Wave Rectifier
2) Full wave Rectifier
3) Bridge Rectifier

Voltage Regulator IC(78XX)

IC 7805 is a DC regulated IC of 5V. This IC is very flexible and is widely employed in all types of circuit like a voltage regulator. It is a three terminal device and mainly called input , output and ground. Pin diagram of the IC 7805 is shown in the diagram below.



The pin explanation of the 7805 is described in the following table:

PIN NO.	PIN	DESCRIPTION
1	INPUT	In this pin of the IC positive unregulated voltage is given in regulation.
2	GROUND	In this pin where the ground is given. This pin is neutral for equally the input and output.

3	OUTPUT	The output of the regulated 5V volt is taken out at this pin of the IC regulator.

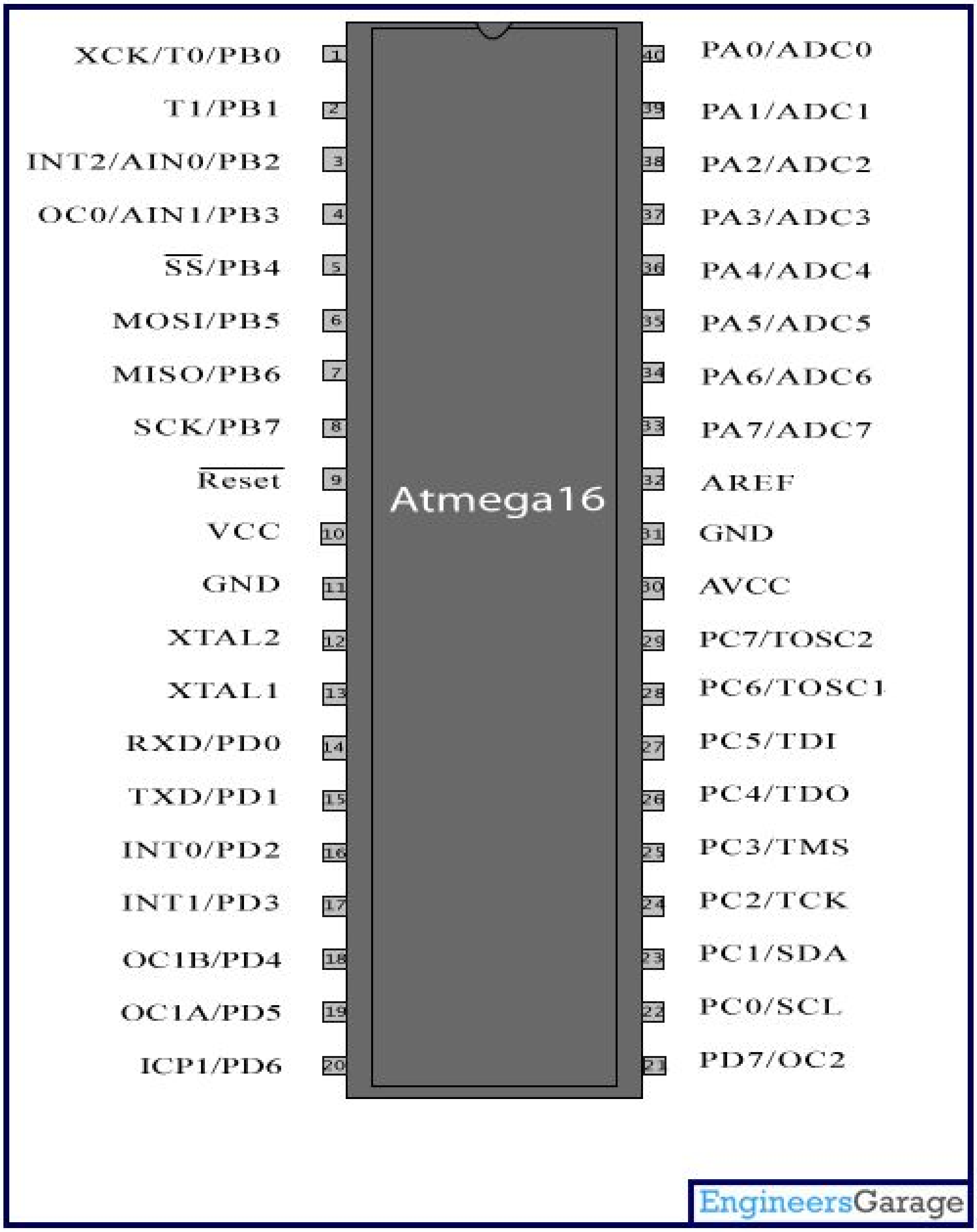
DAY 2 ,9/12/15

TOPICS COVERED ON DAY 2

- Functioning of ATmega16
- Coding through softwares like Atmel Studio,Proteus,Robokits
- Functioning of ADC

FUNCTIONING OF ATMEGA16

ATmega16 is a 40 pin microcontroller. There are 32 I/O (input/output) lines which are divided into four 8-bit ports designated as PORTA, PORTB, PORTC and PORTD.ATmega16 has 16 KB programmable flash memory



Important I/0 Ports

1. DDR//for input and output purpose
2. Port//For high and low voltage
3. PIN//To read the values of input

The AVCC pin is for analog part of the microcontroller while the VCC is for the digital part of the microcontroller.
AREF pin is used in ADC.

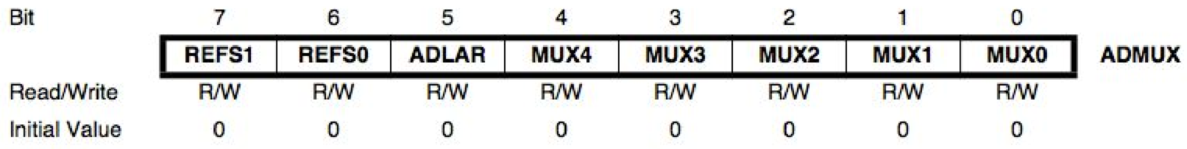
ADC(Analogue to Digital Converter)

It comprises of two registors

1. ADMUX
2. ADCSRA

Different Bits of ADMUX

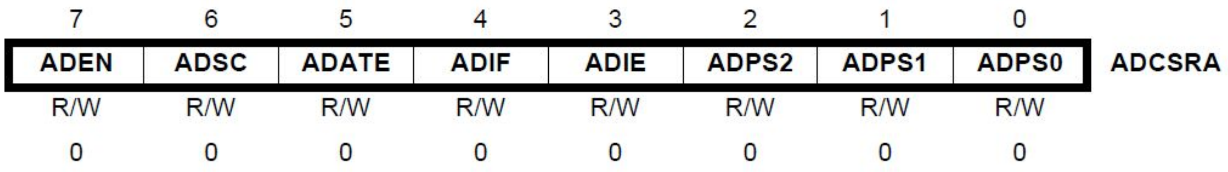
1)REFS1-Sets the reference voltage
2)REFS0-Sets the reference voltage
3)ADLAR-Two cases
  1)ADLAR=1-Stores value in 8-bit form
  2)ADLAR=2-Stores value in 10-bit form
4)MUX 0,1,2,3-Sets the pin number



Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Different Bits of ADCSRA

1)ADEN-The value should be set to 1 for functioning of ADC
2)ADSC-The value should be set to 1 each time input is taken
3)ADPS 0,1,2-Sets the factor by which the crystal frequency has to be divided



7	6	5	4	3	2	1	0	
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
0	0	0	0	0	0	0	0	

DAY 3,10/12/15

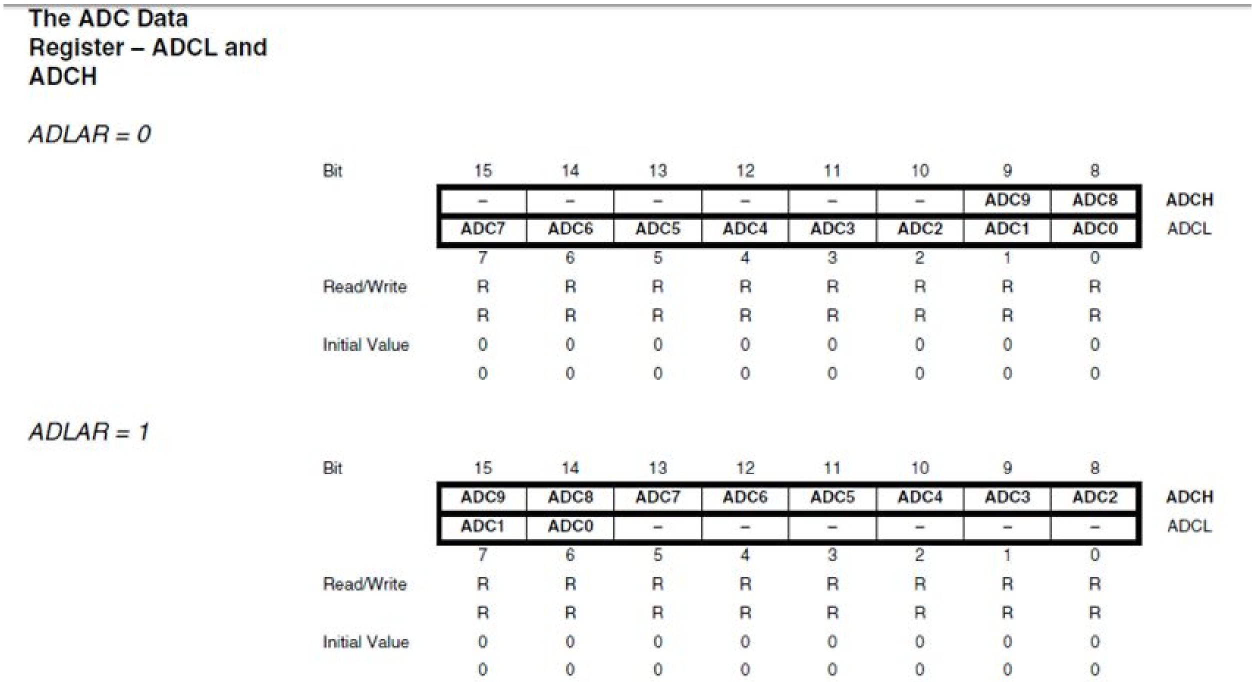
TPOICS COVERED ON DAY 3

- Data processing for ADLAR=0 and ADLAR=1
- Basic concept on ARDUINO

There are two parts of adlar:-

1. ADCH
2. ADCL

The ADC conversion result will be placed in these two 8 bit ADC data register. It’s need two 8 bits data register to hold the result; because the Atmel AVR ADC circuit used 10 bit resolution. Which’s mean the first lowest 8 bits will be placed in the **ADCL** register and the remaining 2 bits will be placed in the **ADCH**register (assuming we use the default **ADLAR** value of **0** in the **ADMUX** register).
The reverse is true for **ADLAR=1**



The ADC Data Register – ADCL and ADCH

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

BASIC CONCEPT ON ARDUINO

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on computer, used to write and upload computer code to the physical board.Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – we can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.

DAY 4,11/12/15

TOPICS COVERED ON DAY 4

- Different timers of ATMEGA16
- PWM using timers
- Coding on timers and PWM

TIMERS

There are three timers os ATMEGA-16

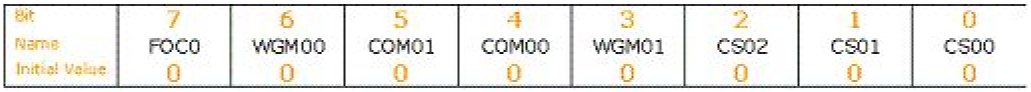
1. TIMER0
2. TIMER1
3. TIMER2

TIMER0

The simplest timer is TIMER0. Its resolution is 8 BIT i.e. it can count from 0 to 255.
It comprises of two registers:-

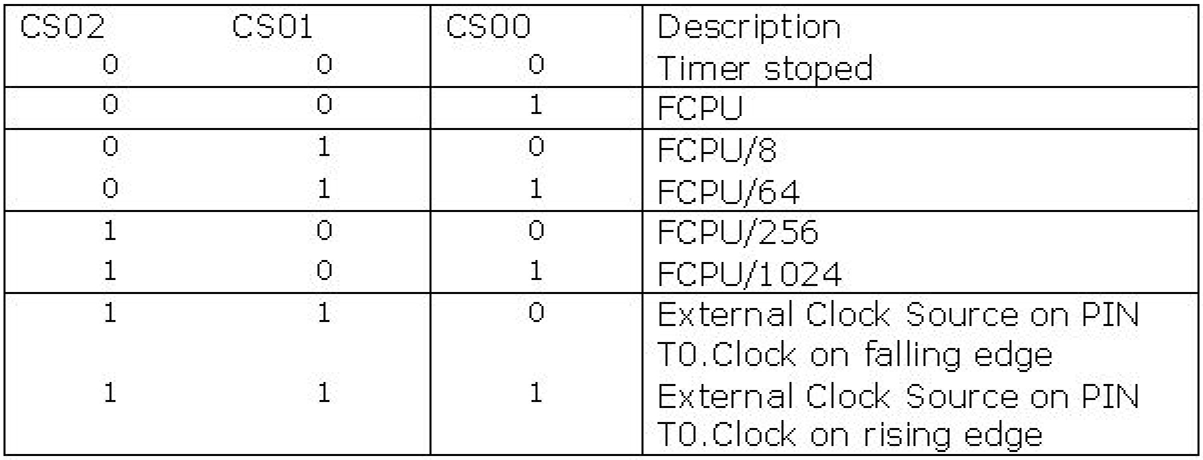
1)TCCR0-8 bit
2)TCNT0-8 bit

DIFFERENT BITS OF TCCR0



Bit	7	6	5	4	3	2	1	0
Name	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
Initial Value	0	0	0	0	0	0	0	0

The bits CS00,CS01,CS02 are used to prescale the timer.Different combinations give different pre-scaling factor as shown in the table.



CS02	CS01	CS00	Description
0	0	0	Timer stoped
0	0	1	FCPU
0	1	0	FCPU/8
0	1	1	FCPU/64
1	0	0	FCPU/256
1	0	1	FCPU/1024
1	1	0	External Clock Source on PIN T0.Clock on falling edge
1	1	1	External Clock Source on PIN T0.Clock on rising edge

Bit 6, 3 – WGM01:0: Waveform Generation Mode
These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of Waveform Generation to be used.

Bit 4,5-Helps in selecting inverted and non-inverted PWM

TCNTO=0//The bit register gets initialized
It will update itself by +1 if the timer ticks 1
It can store values from 0 to 255.After that it overflows and again gets initialized with TCNT0=0

PWM

PWM stands for Pulse Width Modulation.

It can be generated by comparing predetermined waveform with a reference voltage level or by making simple analog circuits.

Duty Cycle of a PWM waveform is given by the following relation.

$$Duty\ Cycle = \frac{T_{on}}{T_{on} + T_{off}} \times 100\ \%$$

PWM USING TIMER0

TCCR0

This register is used for configuring the TIMER0. The explanation of various bits of this register is as follows.

Bit No	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**FOC0**	**WGM00**	**COM01**	**COM00**	**WGM01**	**CS02**	**CS01**	**CS00**
Initial Val	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**

WGM – Wave Form Generation Mode

The table below shows the various modes supported by TIMER0.We are interested in Fast PWM mode.

Mode	WGM00	WGM01	Mode Of Operation

0	0	0	Normal
1	0	1	PWM Phase Correct
2	1	0	CTC
3	1	1	Fast PWM

From the table it is clear that for Fast PWM we need mode 3. To get it we must set WGM00=1 and WGM01=1

COM – Compare Output Mode

These bits are used to set the Output mode in various Wave form generation mode. For Fast PWM mode these can be used to achieve following output modes.

COM01	COM00	Output Mode
0	0	Normal Port Operation (OC0 disconnected)
1	0	RESERVED
0	1	Non Inverted PWM
1	1	Inverted PWM

We need the "Non Inverted PWM output mode" so we set COM01=0 and COM00=1

CS – Clock Select

These are used to set an Input Clock for TIMER.

CS02 = 0

CS01 = 0

CS00 = 1

Now the TIMER is in Fast PWM mode to vary its output duty cycle we just need to set the OCR0 (Output Compare Register for Timer 0). For example setting it to 0 will generate PWM with duty cycle 0% (Totally off) while setting it to 128 will generate 50% duty cycle and 255 will generate 100% duty cycle signals.

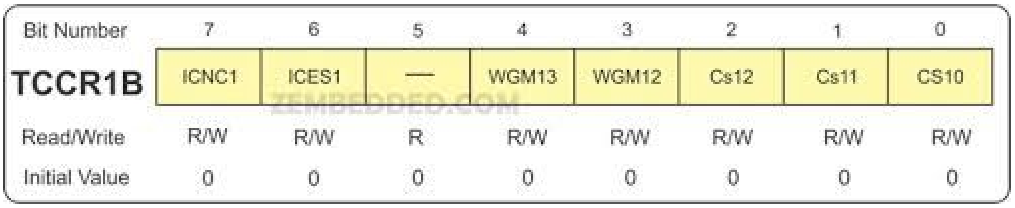
TIMER1

It is a more accurate 16 bit clock.
It has the following registers

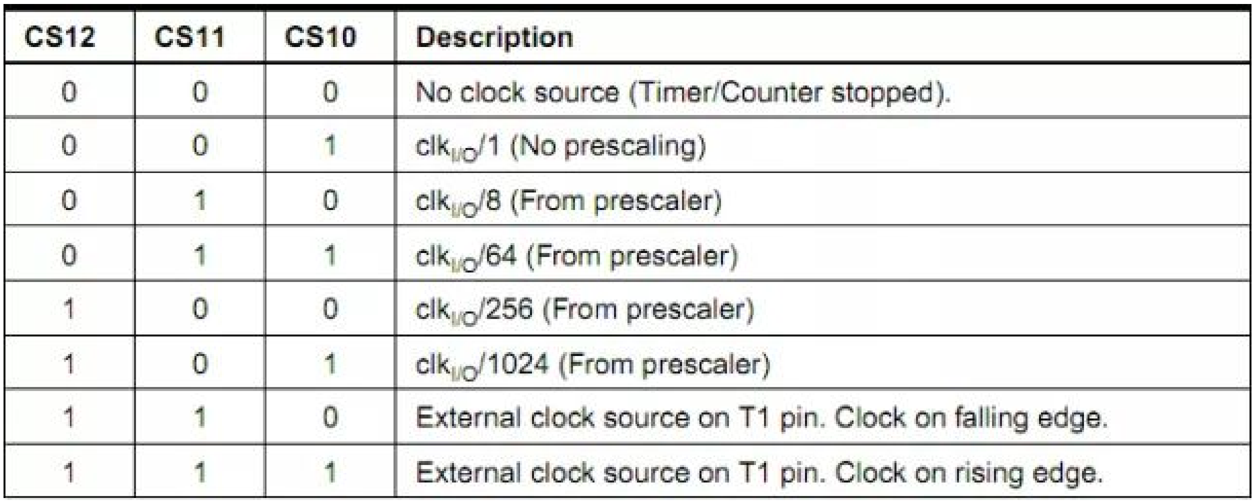
1. TCCR1A
2. TCCR1B
3. TCNT1H anad TCNT1L
4. OCR1AH and OCR1AL
5. OCR1BH and OCR1BL
6. ICR1H and ICR1L
7. TIMSK
8. TIFR

TCCR1B Register

The Timer/Counter1 Control Register B– TCCR1B Register is as follows.



Bit Number	7	6	5	4	3	2	1	0
TCCR1B	ICNC1	ICES1	—	WGM13	WGM12	Cs12	Cs11	CS10
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0



CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk$_{I/O}$/1 (No prescaling)
0	1	0	clk$_{I/O}$/8 (From prescaler)
0	1	1	clk$_{I/O}$/64 (From prescaler)
1	0	0	clk$_{I/O}$/256 (From prescaler)
1	0	1	clk$_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

DAY 5,12/12/15

TOPICS COVERED ON DAY 5

- Interrupts and there uses
- USART form of communication

INTERRUPTS

In system programming, an interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing. The processor responds by suspending its current activities, saving its state, and executing a function called an *interrupt handler* (or an interrupt service routine, ISR) to deal with the event. This interruption is temporary, and, after the interrupt handler finishes, the processor resumes normal activities.
There are total 21 interrupts in ATMEGA16 of which 17 are internal and 4 are external.

The four types of external interrupts are:-

1. INT 0

2. INT 1
3. INT 2
4. RESET

The header file for interrupt library is

#include <avr/interrupt.h>

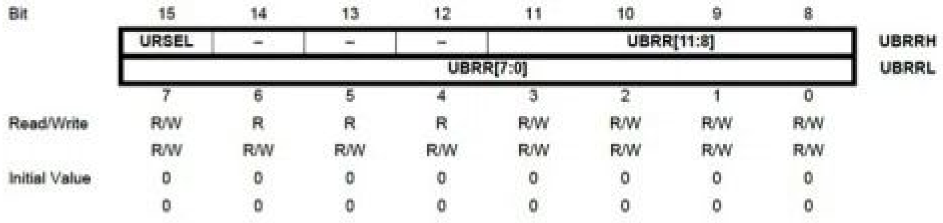
The general syntax for writing interrupt function is :
ISR(INTERRUPT_vect)

USART Form of Communication

A universal asynchronous receiver/transmitter, abbreviated UART is a computer hardware device that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as TIA (formerly EIA) RS-232, RS-422 or RS-485. The *universal* designation indicates that the data format and transmission speeds are configurable. The electric signaling levels and methods (such as differential signaling etc.) are handled by a driver circuit external to the UART.

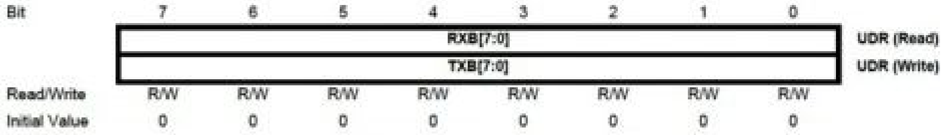
Baud Rate Generation

The baud rate of UART/USART is set using the 16-bit wide UBRR register. The register is as follows:



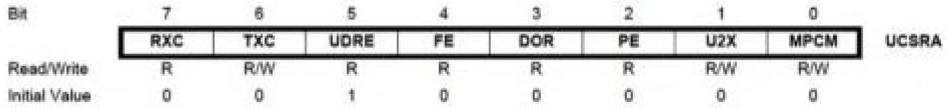
Bit	15	14	13	12	11	10	9	8	
	URSEL	–	–	–	UBRR[11:8]				UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

UDR: USART Data Register (16-bit)



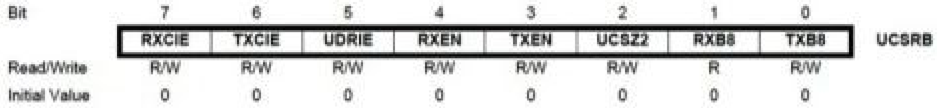
Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UCSRA: USART Control and Status Register A (8-bit)



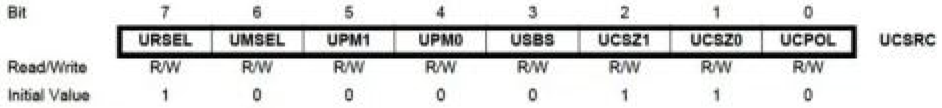
Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

UCSRB: USART Control and Status Register B (8-bit)



Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UCSRC: USART Control and Status Register C (8-bit)



Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	