

# Tutorial ATmega328 - #4:

## ADC e PWM

<b>Conoscenze richieste:</b>	Programmazione in linguaggio C
<b>Tempo richiesto:</b>	1 ora circa
<b>Linguaggio tecnico usato:</b>	Semplice ed amichevole

**Autore:** Emanuele Aimone

## Contents

INTRODUZIONE.....	3
GLI ELEMENTI USATI IN QUESTO TUTORIAL.....	3
L'hardware.....	3
I COLLEGAMENTI .....	4
IL FIRMWARE .....	4
Echo dei dati ricevuti via seriale .....	<b>Errore. Il segnalibro non è definito.</b>
Caricare il firmware .....	7

## INTRODUZIONE

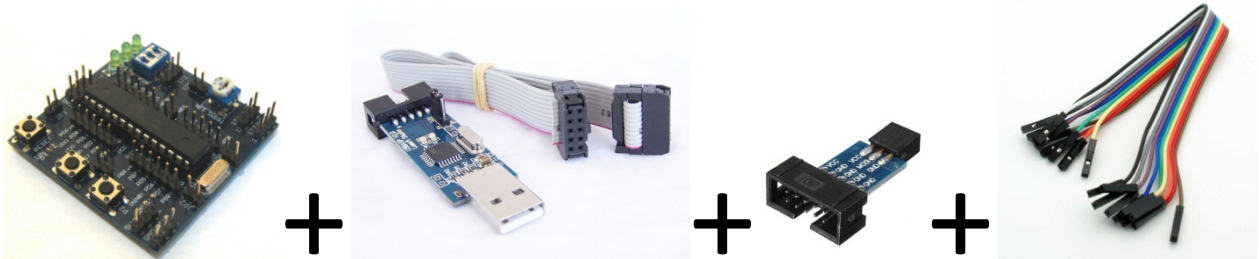
In questo tutorial viene usato un potenziometro per cambiare il duty cycle del PWM. Il cambiamento del duty cycle provocherà un innalzamento o abbassamento di luminosità sul led.

## GLI ELEMENTI USATI IN QUESTO TUTORIAL

Per il tutorial abbiamo bisogno di una semplice demoboard, il programmatore USBasp, un potenziometro o trimmer, il compilatore WinAVR, e il caricatore di firmwareExtremeBurner v1.4.3.

### L'hardware

*Ddemoboard ATmega8* + *USBasp* + *AdattatoreUSBasp* + *Jumpers*



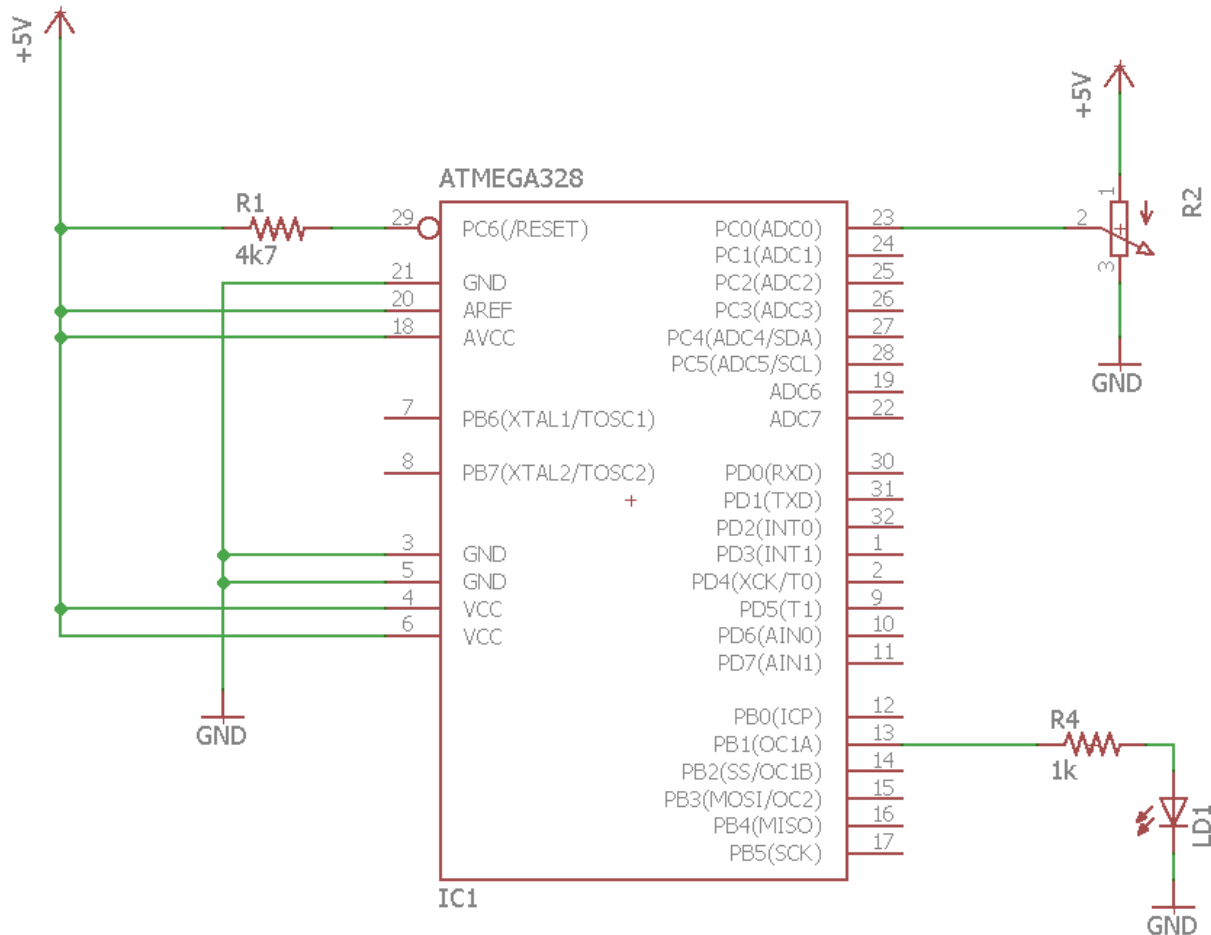
+ *Potenziometro*



## I COLLEGAMENTI

Bisogna collegare il potenziometro sulla porta PC0 (PIN23) dove vi è il convertitore analogico digitale ADC0

Un led della demoboard lo potete collegare sul pin PB1 (pin 13) della demoboard.



## IL FIRMWARE

Il firmware cambierà l'intensità luminosa del led collegato sul pin PB1 (pin13) della demoboard ogni volta che il potenziometro viene girato.

Vediamo subito l'esempio di programma che dobbiamo scrivere:

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

//Funzioni:
void PWM_init(void); //inizializza i registri per il PWM
void ADC_init(void); //inizializza i registri per l'ADC
unsigned int adc_read(unsigned char ch); //legge il valore dell'ADC

//il nostro codice principale, da dove il microcontrollore parte:
int main(void){

    //configura PB1 (PIN15) come Output
    //e tutti gli altri vengono configurati come Input di default
    DDRB = 0x02;

    cli(); //Disabilita gli interrupts globali

    PWM_init(); //inizializza i registri per il PWM
    ADC_init(); //inizializza i registri per l'ADC

    sei(); //Abilita gli interrupts globali

    //Ciclo infinito:
    while(1)
    {
        //legge dal canale 0 (PC0 sul pin 23)
        //e assegna il valore dell'ADC al registro
        //OCR1A per variare il duty cycle del PWM
        //generato sul pin 15 (PB1)
        OCR1A = adc_read(0);
    }
}

//inizializza i registri per il PWM
void PWM_init(void){
    // imposta il PWM per il 50% duty cycle a 10bit
    OCR1A = 0x01FF;

    //imposta la modalità non invertente
    TCCR1A |= (1 << COM1A1);

    //imposta modalità di correzione fase a 10bit
    TCCR1A |= (1 << WGM11) | (1 << WGM10);

    //imposta il prescaler a 8 e avvia il PWM
```

```
TCCR1B |= (1 << CS11);

}

//inizializza i registri per l'ADC
void ADC_init(void){
    // AREF = AVcc : il riferimento dell'ADC è uguale
    //a quello dell'alimentazione del microcontrollore
    ADMUX = (1<<REFS0);

    // ADC Abilitato e il prescaler è di 128
    // 16000000/128 = 125000
    ADCSRA = (1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
}

unsigned int adc_read(unsigned char ch){
    // ch seleziona il canale dell'ADC 0~7
    // ANDing with '7' will always keep the value
    // of 'ch' between 0 and 7
    ch &= 0b00000111; // fa la AND

    // prima di fare l'or porta a 0 i primi 3 bit di ADMUX
    ADMUX = (ADMUX & 0xF8)|ch;

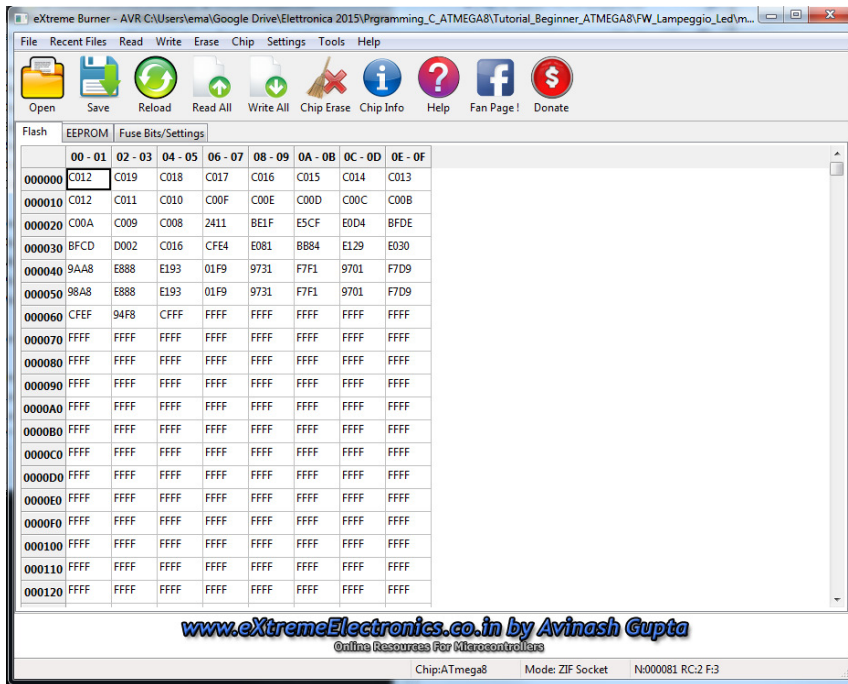
    // avvia la conversione dell'ADC
    // impostando a '1' il bit ADSC
    ADCSRA |= (1<<ADSC);

    // aspetta che la conversione finisca
    // ADSC diventa '0' quando la conversione
    // finisce.
    while(ADCSRA & (1<<ADSC));

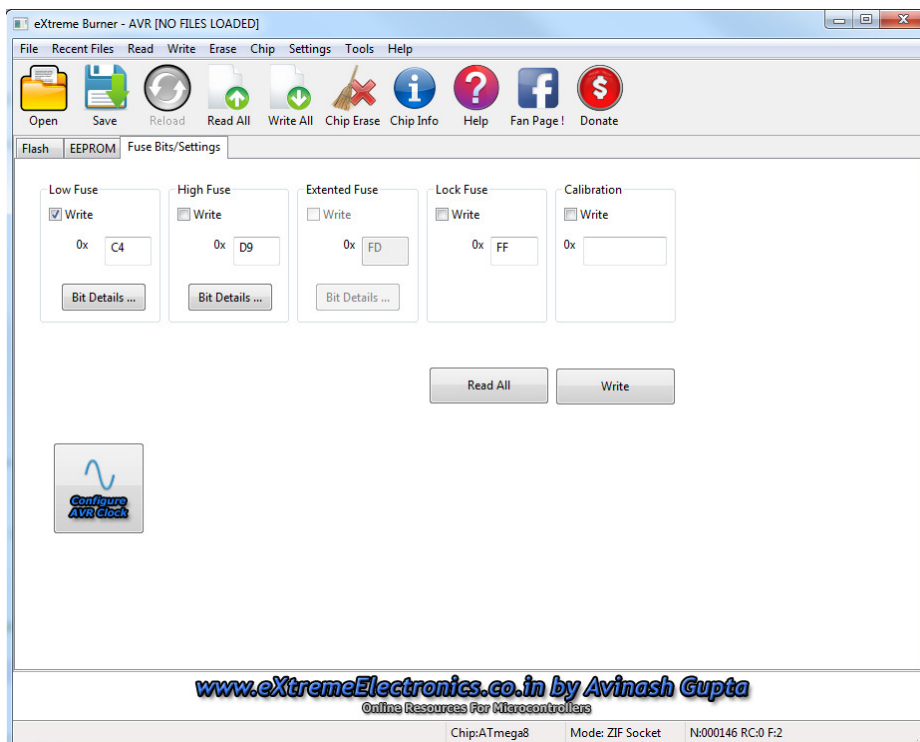
    return (ADC);
}
```

## Caricare il firmware

Apriamo Extreme Burner v1.4.3 e clicchiamo sul tasto Open per andare a caricare il file main.hex:



Ora è molto importante impostare i Fuse, andiamo sul tab chiamato "Fuse Bits/Settings" e impostiamo i Fuse esattamente come l'immagine che segue:



Quindi:

**Low Fuse = 0xDF**

**High Fuse = 0xD9**

**Lock Fuse = 0xFF**

**Calibration = 0xB9B9B9B9**

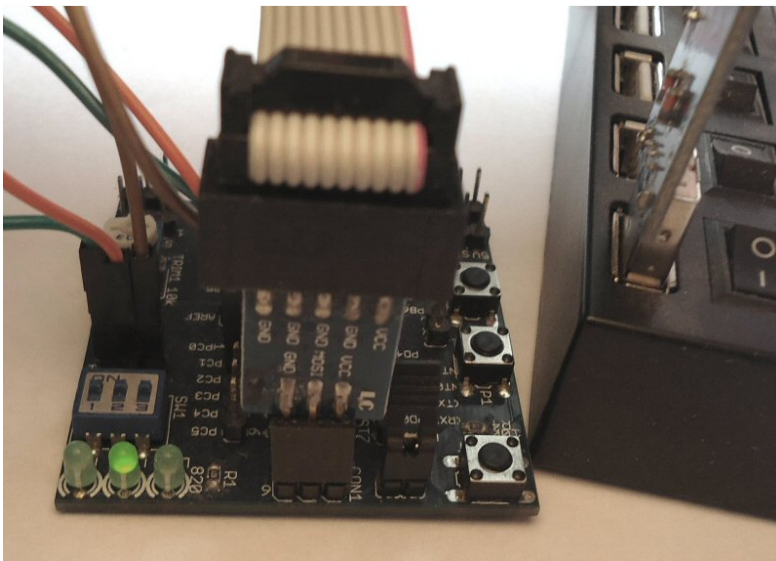
Mettete anche la spunta su write su di almeno il "Low Fuse"!

Per capire come impostare il clock possiamo vedere il datasheet dell'ATmega8 a pagina 26 oppure ci affidiamo al fuse calculator:

<http://www.engbedded.com/fusecalc/>

Ricordate di non cambiare i fuse se non capite cosa state facendo, potreste non poter più riprogrammare il vostro ATmega328.

Ora possiamo finalmente caricare il programma, assicuriamoci dei collegamenti come da foto:



Clicchiamo su Write All, se il caricamento è andato a buon fine allora avremo un schermata tipo questa:



