



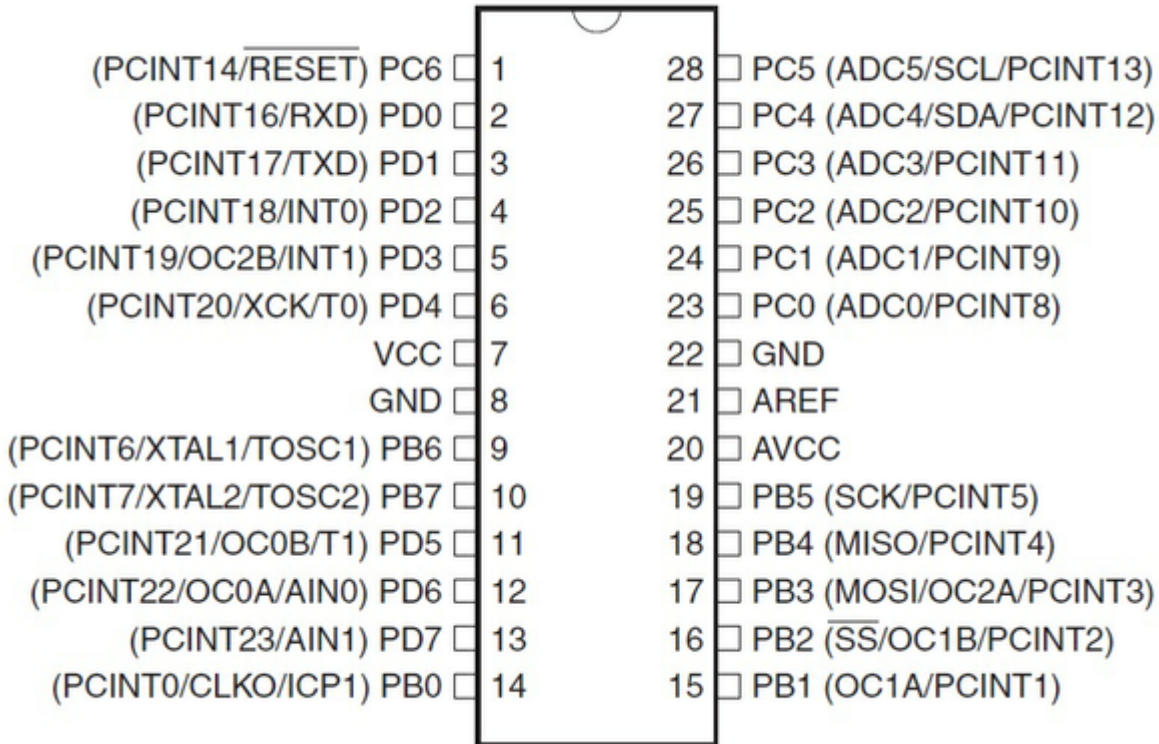
**Nazwa implementacji:** Wdrożenie rejestru 74HCT164 do współpracy z mikrokontrolerem AVR

**Autor:** Łukasz Cieżki

**Opis implementacji:** Podobnie jak w poprzedniej implementacji, doprowadzimy rejestr do współpracy z mikrokontrolerem AVR

Bazując poprzednim przykładzie, będziemy musieli podłączyć rejestr, tylko tym razem bezpośrednio do mikrokontrolera.

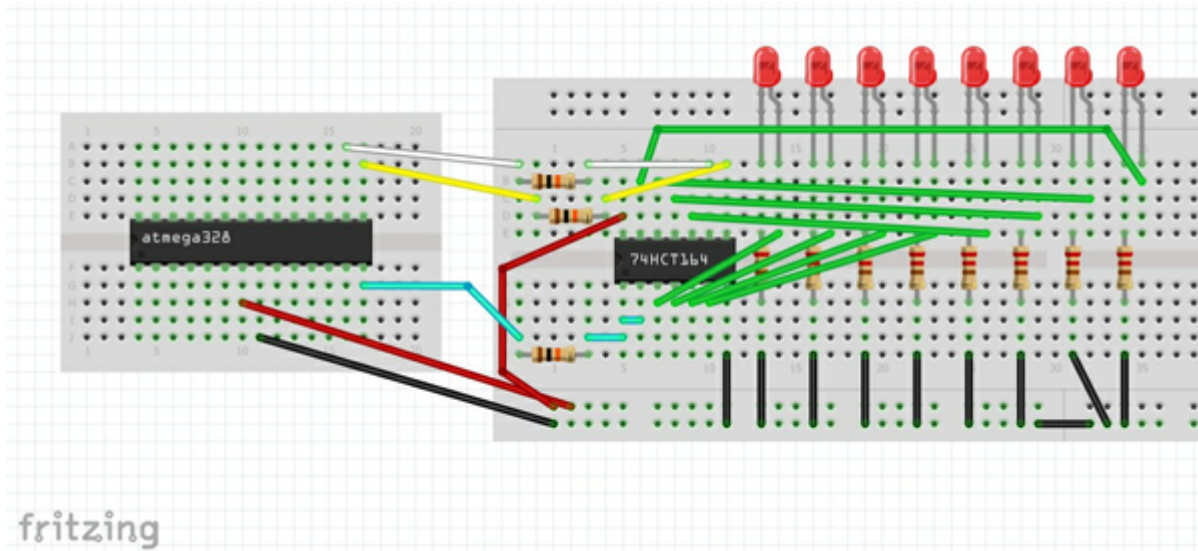
Za mikrokontroler posłuży nam dość popularna AtMEGA 328P:



Skorzystamy z portów C, ale nic nie stoi na przeszkodzie aby użyć innych.

Podłączenie płytki prototypowej:





## fritzing

Pamiętajmy, że

należy jeszcze podłączyć zasilanie do AtMEGI

Spróbujmy wyświetlić sekwencję taką, jak wcześniej - 01101110

Zabieramy się za kod do naszego AVR'a:

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

#define stan0 PORTC &= ~(1 << PC0)
#define stan1 PORTC |= 1 << PC0
#define przesun0 PORTC &= ~(1 << PC1)
#define przesun1 PORTC |= 1 << PC1
#define reset0 PORTC &= ~(1 << PC2)
#define reset1 PORTC |= 1 << PC2

int main(){
    DDRC = 0x07; //ustaw PC0, PC1, PC2 jako wyjścia
    PORTC = 0x00; //nadaj stan niski wszystkim portom C

    reset0; //resetuj bity rejestru
    reset1; //zakoncz resetowanie bitow rejestru

    stan0; //nadaj stan 0 na rejestr
    przesun1; //przesun bit rejestru
    przesun0; //zakoncz przesuwanie

    stan1; //nadaj stan 1 na rejestr
    przesun1; //przesun bit rejestru
    przesun0; //zakoncz przesuwanie

    stan1;
    przesun1;
    przesun0;

    stan0;
    przesun1;
```





```
przesun0;  
  
stan1;  
przesun1;  
przesun0;  
  
stan1;  
przesun1;  
przesun0;  
  
stan1;  
przesun1;  
przesun0;  
  
stan0;  
przesun1;  
przesun0;  
for(;;);  
}
```

Tutaj również kod jest bez pętli, które by go skróciły.

Efekt końcowy:

