LED Blink with LOW-LEVEL

```
void setup() {
        DDRB=0b00100000;
        PORTB=0b00000000;
}
void loop() {
        PORTB=0b00100000;
        delay(1000);
        PORTB=0b00000000;
        delay(1000);
}
```

အထက်ပါ Program ကတော့ led မီးလုံး တစ်လုံးကို one second တိုင်းမှာ အဖွင့်အပိတ်လုပ်ထားတာ ဖြစ်ပါတယ်။ arduino ရဲ့ circuit diagram ကို ကြည့်လိုက်ရင် built-in led ဖြစ်တဲ့ Port 13 ကို PORTB group ရဲ့ ၅ ခုမြောက် နေရာဖြစ်နေတာကို တွေ့ရပါလိမ့်မယ်။ ဒီ Port တွေ အကြောင်းကို အောက်စာပိုဒ်တွေမှာ ဆက်ပြီး ရှင်းပြပေးထားပါတယ်။ arduino uno ရဲ့ schematic ကို ဒီလင့်မှာ ကြည့်နိုင်ပါတယ် ။ https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf

ATmega328/P မှာ Pin အရေအတွက်အားလုံး 28 pins ပါရှိပါတယ်။ အထက်မှာပြထားတဲ့ ပုံ(၁)ထဲမှာ PortB အတွက် (PB0, PB1, PB2, PB3, PB4, PB5, PB6, PB7)၊ PortC အတွက် (PC0, PC1, PC2, PC3, PC4, PC5, PC6) နဲ့ PortD အတွက် (PD0, PD1, PD2, PD3, PD4, PD5, PD6, PD7, PD8) ဆိုပြီး Port Group သုံးခုကို တွေ့ရမှာ ဖြစ်ပါတယ်။ ကျန်တဲ့ Pin တွေကတော့ ဒီ Controller နဲ့ Analog reference voltage ကို Power ပေးမယ့် Pin တွေပဲ ဖြစ်ပါတယ်။ I/O Port တွေကို သူတို့ရဲ့ Main Function ဖြစ်တဲ့ input output တွေအပြင် alternate functions တွေအဖြစ်လည်း အသုံးပြုလို့ရပါတယ်။ ဘေးမှာ ကွင်းစကွင်းပိတ်တွေနဲ့ ပြထားတာတော့ သူရဲ့ alternate functions တွေဖြစ်ပါတယ်။ I/O Port တွေ အဖြစ်အသုံးပြုချင်ရင်တော့ သူတို့နဲ့ သက်ဆိုင်တဲ့ DDRx (Data Direction Register)၊ PORTx (Pin Output Register)၊ PINx (Pin Input Register) အစရှိတဲ့ register တွေကို အသုံးပြုရမှာ ဖြစ်ပါတယ်။ x ဆိုတာကတော့ သူတို့နဲ့ သက်ဆိုင်တဲ့ A, B, C, D စတဲ့ Port နာမည်တွေပဲ ဖြစ်ပါတယ်။ DDRx register ကို (DDx0, DDx1, DDx2, DDx3, DDx4, DDx5,

DDx6, DDx7) ဆိုတဲ့ 8 bit နဲ့ ဖွဲ့စည်းထားပါတယ်။

ဥပမာ-

DDRA= (1<<DDA0) ဒါက DDRA ရဲ့ DDA0 ဆိုတဲ့ bit ကို assign လုပ်လိုက်တာပါပဲ။ (1<<DDA0) ဆိုရင် PORTA ရဲ့ ပထမဆုံး Pin ကို output အဖြစ်သုံးမယ်လို့ ပြောလိုက်တာဖြစ်ပါတယ်။ (0<<DDA0) ဆိုရင် PORTA ရဲ့ ပထမဆုံး Pin ကို input အဖြစ်သုံးမယ်လို့ ပြောလိုက်တာဖြစ်ပါတယ်။

DDRA= (1<<DDA0) ကို DDRA=0x01 လို့ assign လုပ်ရင်လည်းရပါတယ်။ ဒါကတော့ hexadecimal အဖြစ် assign လုပ်တာဖြစ်ပါတယ်။ DDRA=0b00000001 ဒါကတော့ binary အဖြစ် assign လုပ်တာဖြစ်ပါတယ်။ PORTx ဆိုတဲ့ register ကိုတော့ (Px0, Px1, Px2, Px3, Px4, Px5, Px6, Px7) အစရှိတဲ့ bit တွေနဲ့ ဖွဲ့စည်းထားပါတယ်။

ဥပမာ အပေါ်မှာ ရေးထားတဲ့ blinking program လေးကိုပဲ အောက်ပါအတိုင်း ပြန်ရေးလိုက်လို့ ရပါတယ်။

```
void setup()
{
    // put your setup code here, to run once:
    DDRB |= (1<<DDB5);
    PORTB = 0b00000000;
}
void loop()
{
    // put your main code here, to run repeatedly:
    PORTB = (1<<PB5);
    delay(1000);
    PORTB = (0<<PB5);
    delay(1000);
}
```

ဒီ program ကို လေ့လာကြည့်ရအောင်... DDRB ဆိုပြီး register ကို အရင်ဆုံး assign လုပ်ပါတယ်။

Data Direction Register ပေါ့. DDRB ဆိုတော့ PORTB ရဲ့ Data Direction Register ကို ခေါ်လိုက်ပါတယ်။ (1<<DDB5) ဆိုတော့ PORTB ရဲ့ ၅ ခုမြောက်ကို output အနေနဲ့ သုံးမယ်ဆိုတာပြောလိုက်တာ ဖြစ်ပါတယ်။ အပေါ်ဆုံးမှာ ရေးခဲ့တဲ့ program မှာတော့ DDRB = 0b00100000; ဆိုပြီး ရေးခဲ့ပါတယ်။ ဒီလို binary နဲ့ assign လုပ်ရင်လည်း ရပါတယ်။ ပြီးရင် PORTB = 0b00000000; ဆိုပြီး ကနဦး အခြေအနေမှာ ဘာ output ကိုမှ ထုတ်မပေးထားပါဘူး။ ဒါဆို setup လုပ်တာပြီးသွားပါပြီ။ loop ထဲကို သွားဝင်ကြည့်ရအောင်။ PORTB = (1<<PB5) ဆိုပြီး PORTB ရဲ့ ၅ ခုမြောက်ကို output ပေးလိုက်ပါတယ်။ အပေါ်ဆုံးက program မှာ PORTB = 0b00100000; ဆိုပြီး ရေးခဲ့ပါတယ်။ အတူတူပဲ ဖြစ်ပါတယ်။ ပြီးတာနဲ့ တစ်စက္ကန့်နာပြီး PORTB = (0<<PB5) ဆိုပြီး output ကို off လိုက်ပါတယ်။

PINx ဆိုတဲ့ register ကိုတော့ (Px0, Px1, Px2, Px3, Px4, Px5, Px6, Px7) အစရှိတဲ့ bit တွေ့နဲ့ ဖွဲ့စည်းထားတာပဲ ဖြစ်ပါတယ်။

ဥပမာ program တစ်ခုကြည့်လိုက်ရအောင်.

```
void setup()  {
        // put your setup code here, to run once:

         DDRB |= (1<<DDB5)|(0<<DDB4);
         PORTB |= (0<<PB5)|(0<<PB4); // make down up
}
void loop()
{
       // put your main code here, to run repeatedly:
        if(PINB == (1<<PB4))
       {
              PORTB = (1<<PB5);
       }
        else
       {
```

```
            PORTB = (0<<PB5);
        }
}
```

ဒါကတော့ PORTB ရဲ့ လေးခုမြောက် Pin ကနေ 5V (logic 1) ဝင်လာရင် ပထမဆုံး Pin ကို 5V (logic 1) ထုတ်ခိုင်းပြီး တကယ်လို့ ဘာမှဝင်မလာရင် 0V (logic 0) မှာပဲ ထားခိုင်းလိုက်တာ ဖြစ်ပါတယ်။ DDRB = (1<<DDB5) | (0<<DDB4); PORTB5 ကိုတော့ output pin အဖြစ်သုံးမယ်လို့သတ်မှတ်ထားပြီး PORTB4 ကိုတော့ Input pin အနေနဲ့ သုံးမယ်လို့ သတ်မှတ်ထားပါတယ်။ ပြီးရင်. PORTB5 ကို ကနဦးအခြေအနေမှာ ဘာ output မှ ထုတ်မထားပေးပါဘူး။ PORTB4 ကိုတော့ Pull - down အနေနဲ့ သုံးမှာ ဖြစ်တဲ့အတွက် 0 လို့ပဲသတ်မှတ်ထားပါတယ်။ တကယ်လို့ Pull -up အနေနဲ့ သုံးချင်ရင်တော့ (1<<PB4) ဆိုပြီး သတ်မှတ်ပေးရမှာ ဖြစ်ပါတယ်။ loop ထဲမှာတော့ pull - down ဖြစ်တဲ့အတွက် if(PINB == (1<<PB4)) ဆိုပြီး စစ်ထားပါတယ်။ တကယ်လို့ pull - up အနေနဲ့ သုံးခဲ့ရင်တော့ if(PINB == (0<<PB4) ဆိုပြီး စစ်ဆေးပေးရမှာပဲ ဖြစ်ပါတယ်။