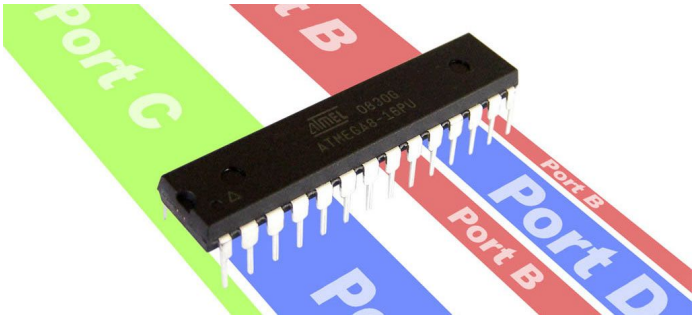
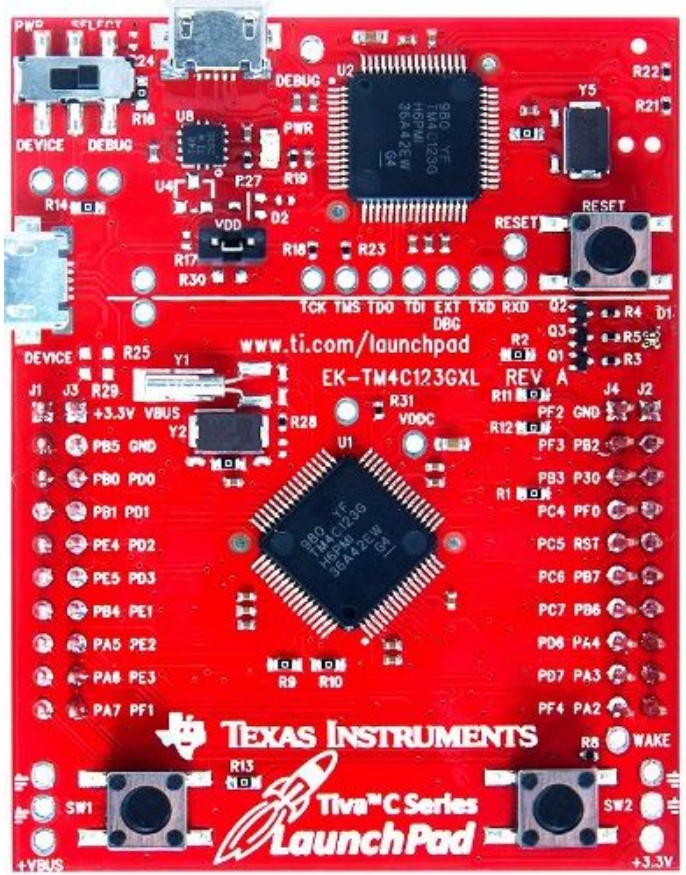
# Mikrokontrolcu Portları




Suhap SAHIN

# Tiva & Stellaris Port Baglantıları




Hardware

digitalRead() and digitalWrite() PORTS

analogRead()

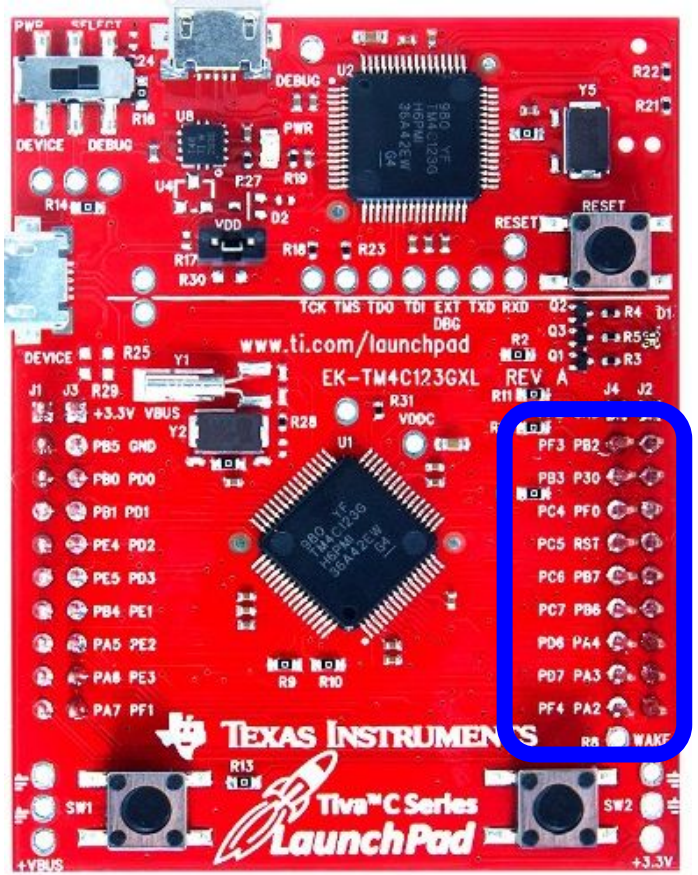
analogWrite()

I²C (TWI)

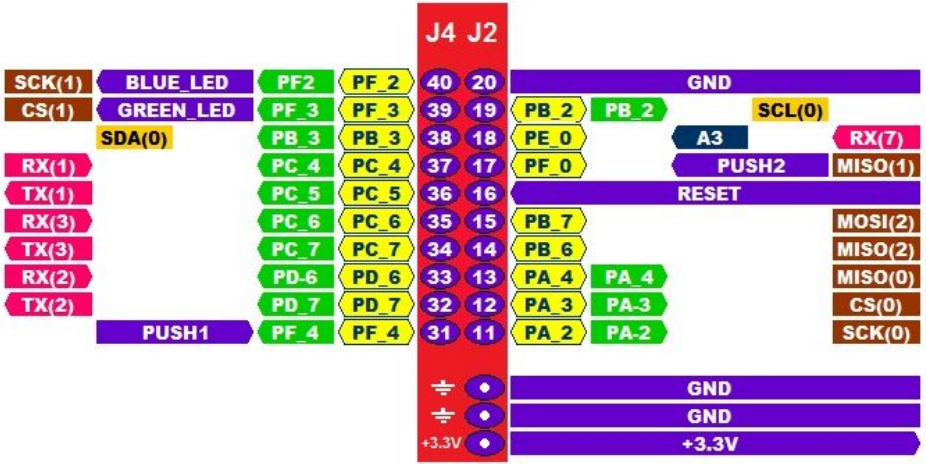
SPI

Hardware Serial

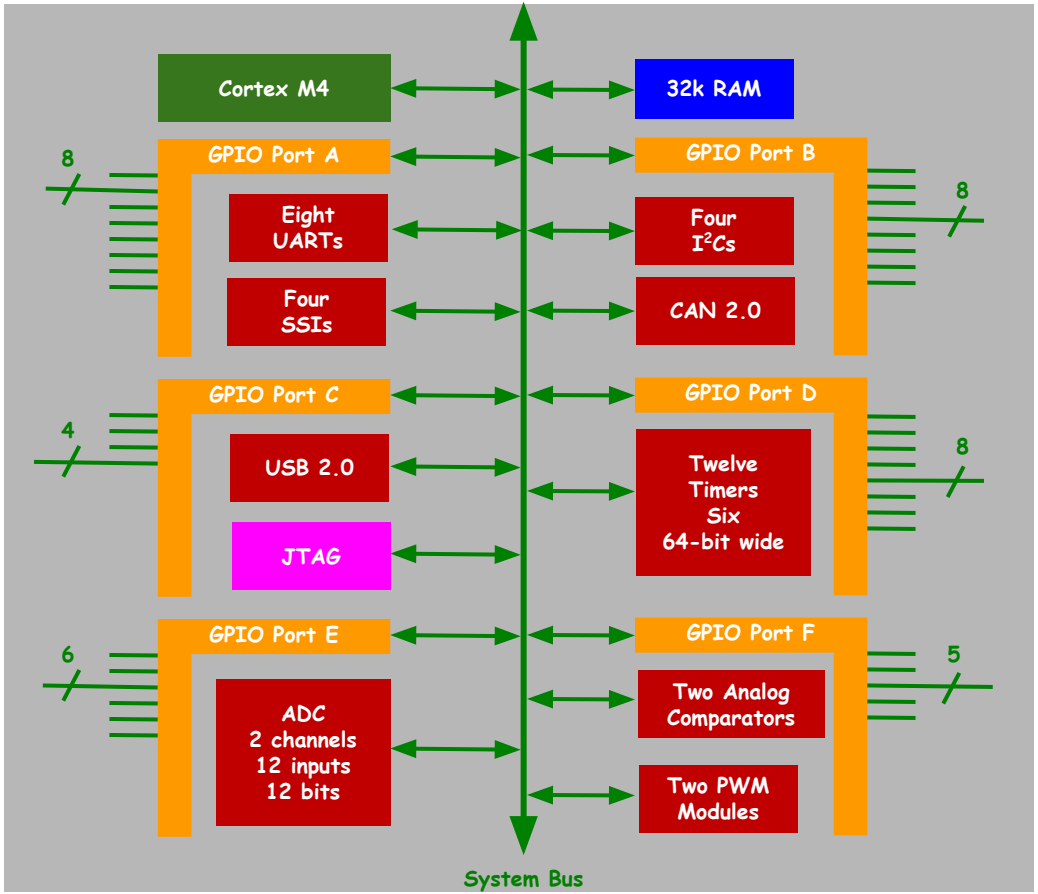
# Tiva & Stellaris Port Baglantıları



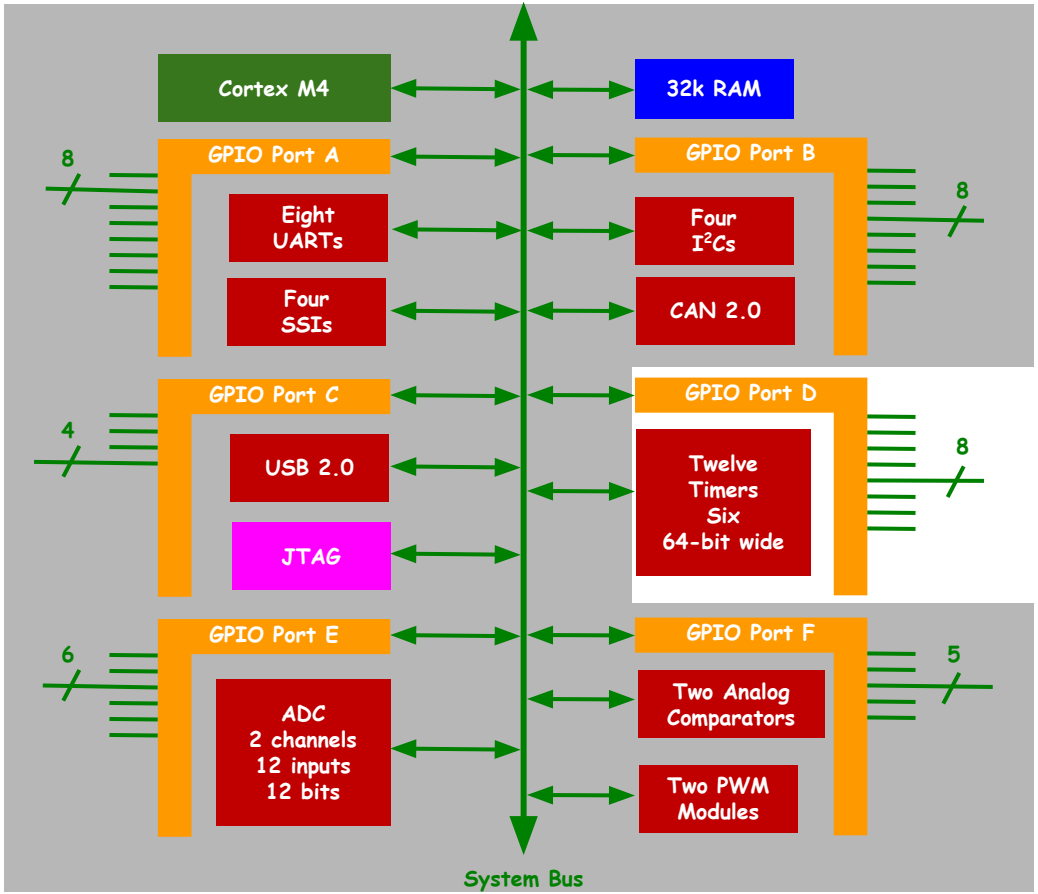

| |
|---|
| Hardware |
| digitalRead() and digitalWrite() PORTS |
| analogRead() |
| analogWrite() |
| I²C (TWI) |
| SPI |
| Hardware Serial |



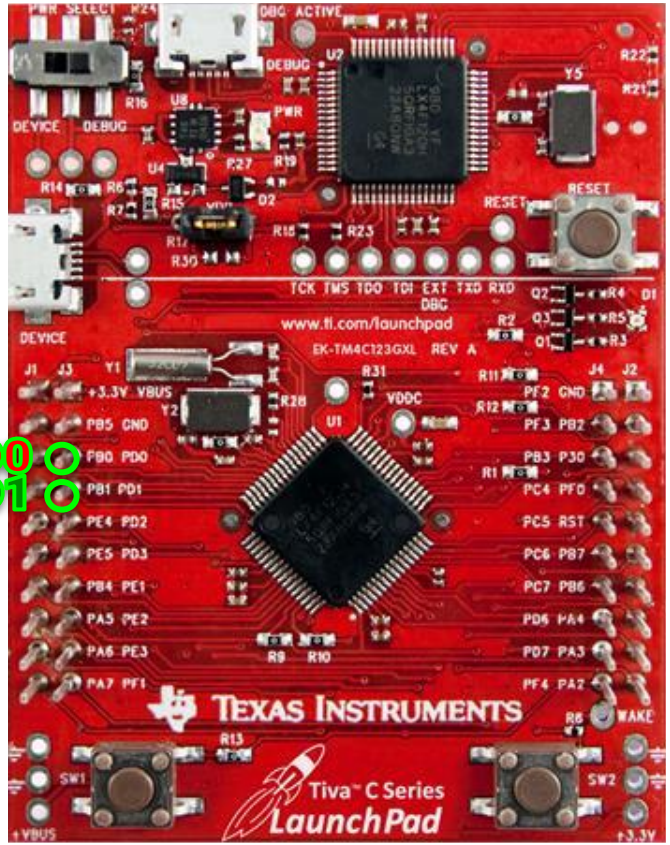
# I/O Portları

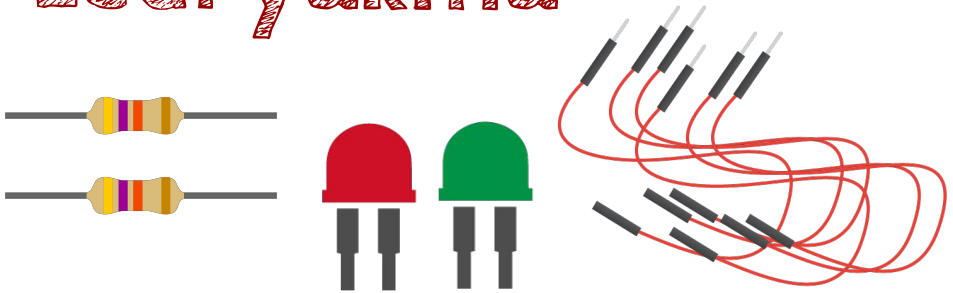

Cortex M4
32k RAM
GPIO Port A
GPIO Port B
8
Eight UARTs
Four I²Cs
8
Four SSIs
CAN 2.0
GPIO Port C
GPIO Port D
4
USB 2.0
Twelve Timers Six 64-bit wide
8
JTAG
GPIO Port E
GPIO Port F
6
ADC 2 channels 12 inputs 12 bits
Two Analog Comparators
5
Two PWM Modules
System Bus

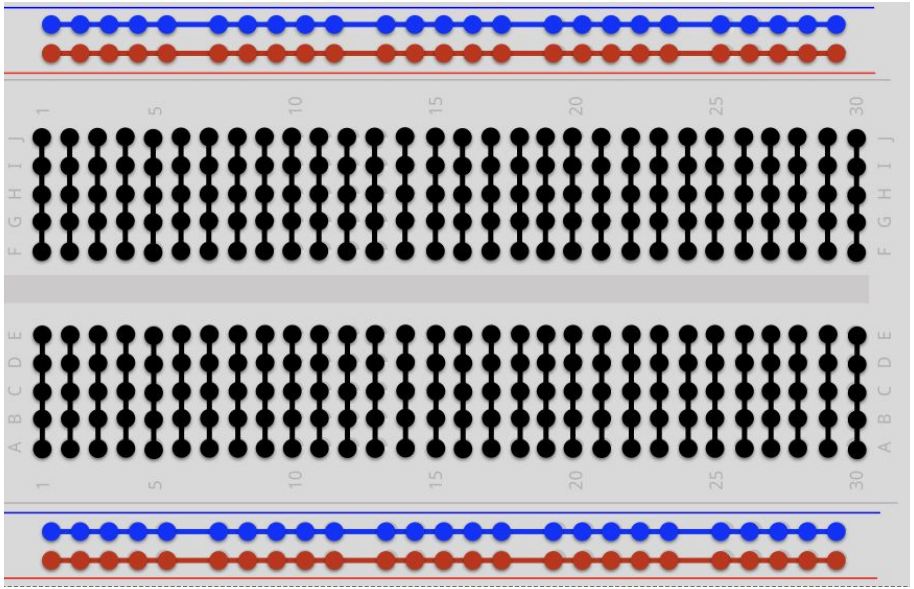
# Dışarıdaki 2 Ledi yakma



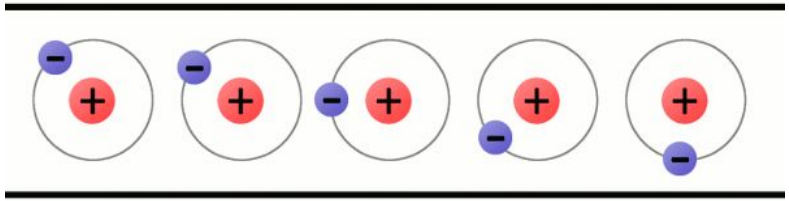
# Dışarıdaki 2 Ledi yakma

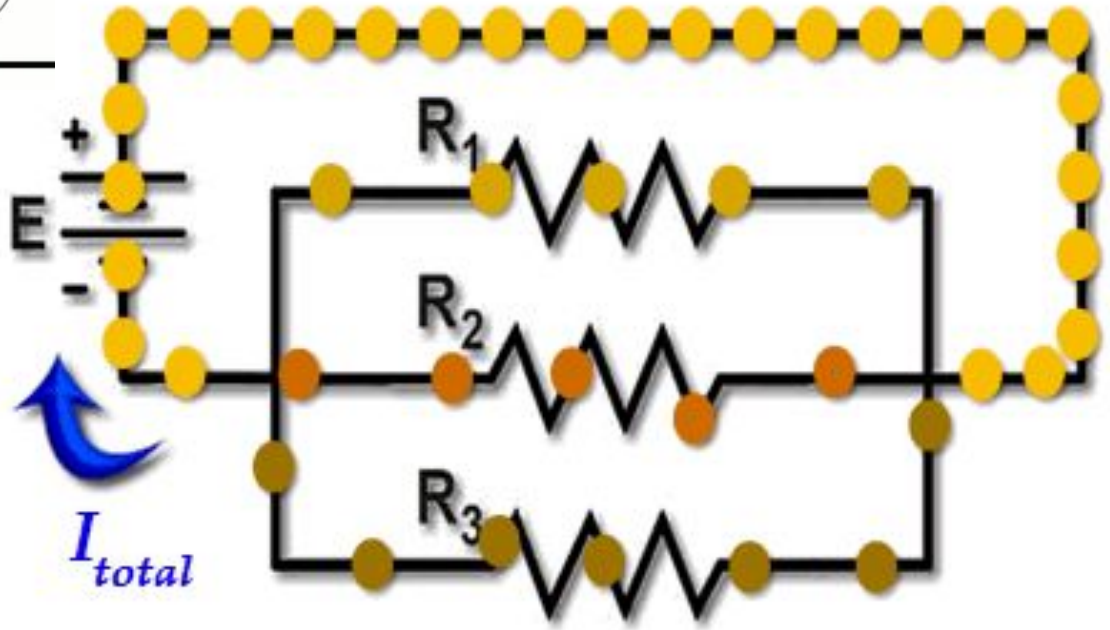


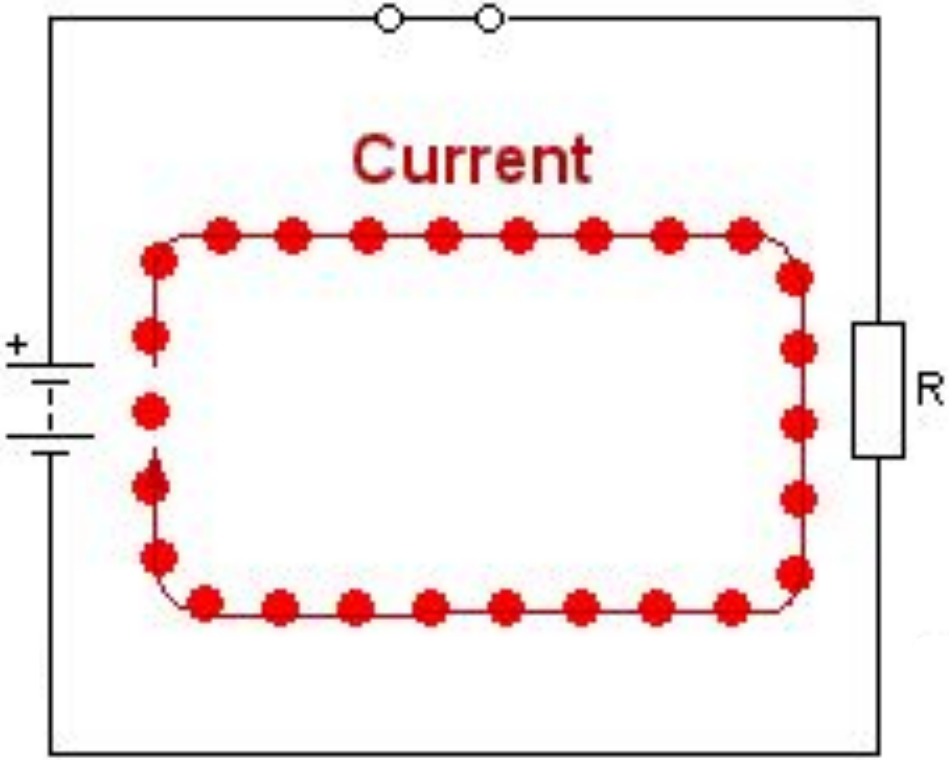


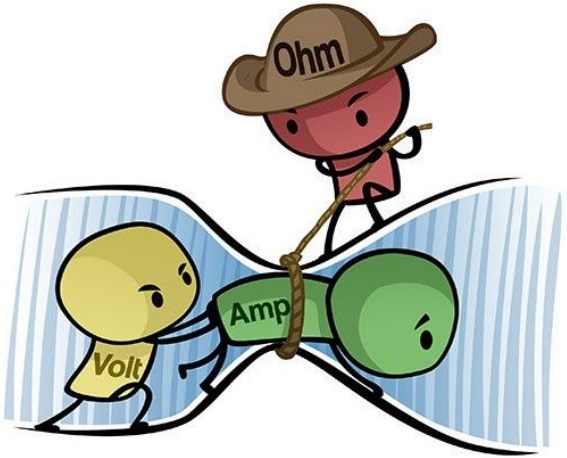
# Elektrik Bilgisi



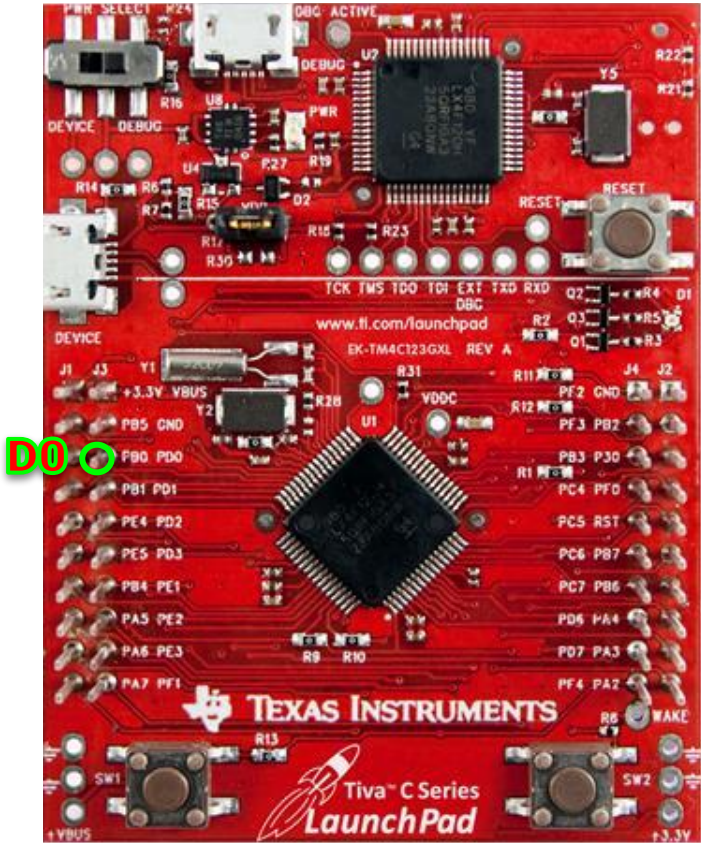


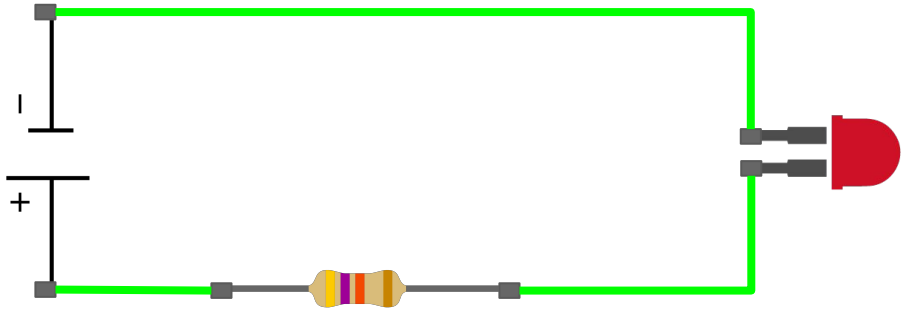
# Elektrik Bilgisi



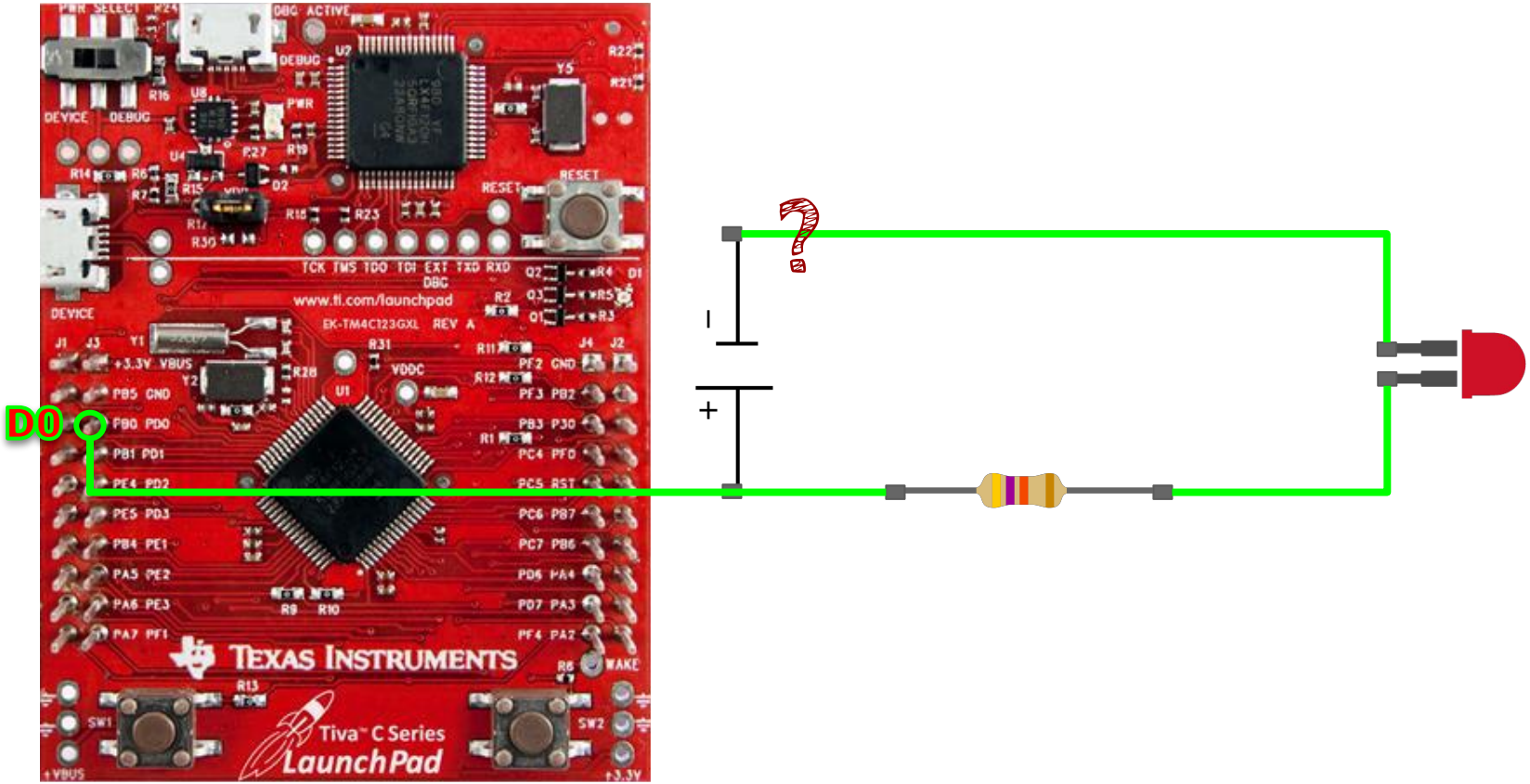



# Elektronik Devre Kurulumu

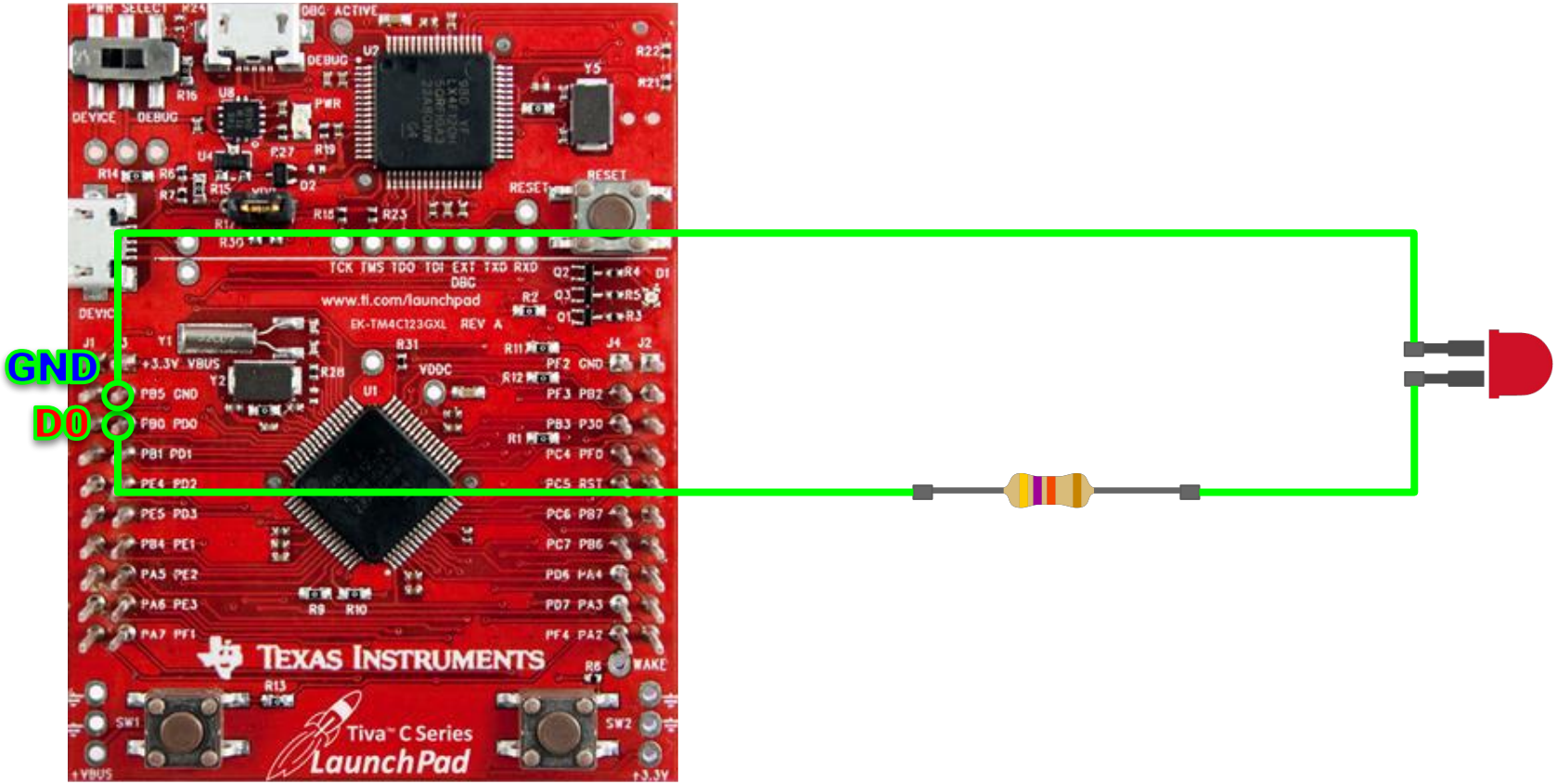


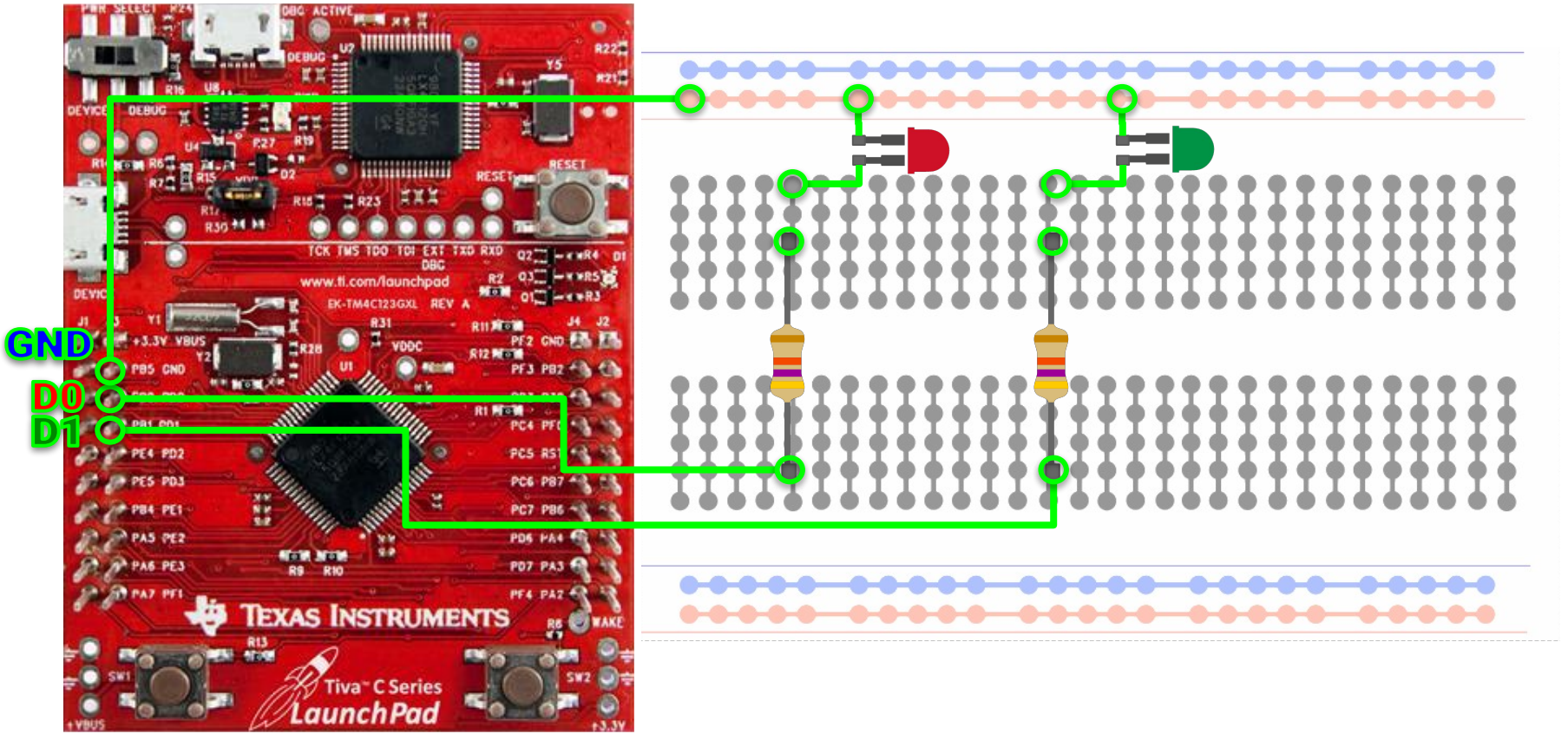
# Elektronik Devre Kurulumu



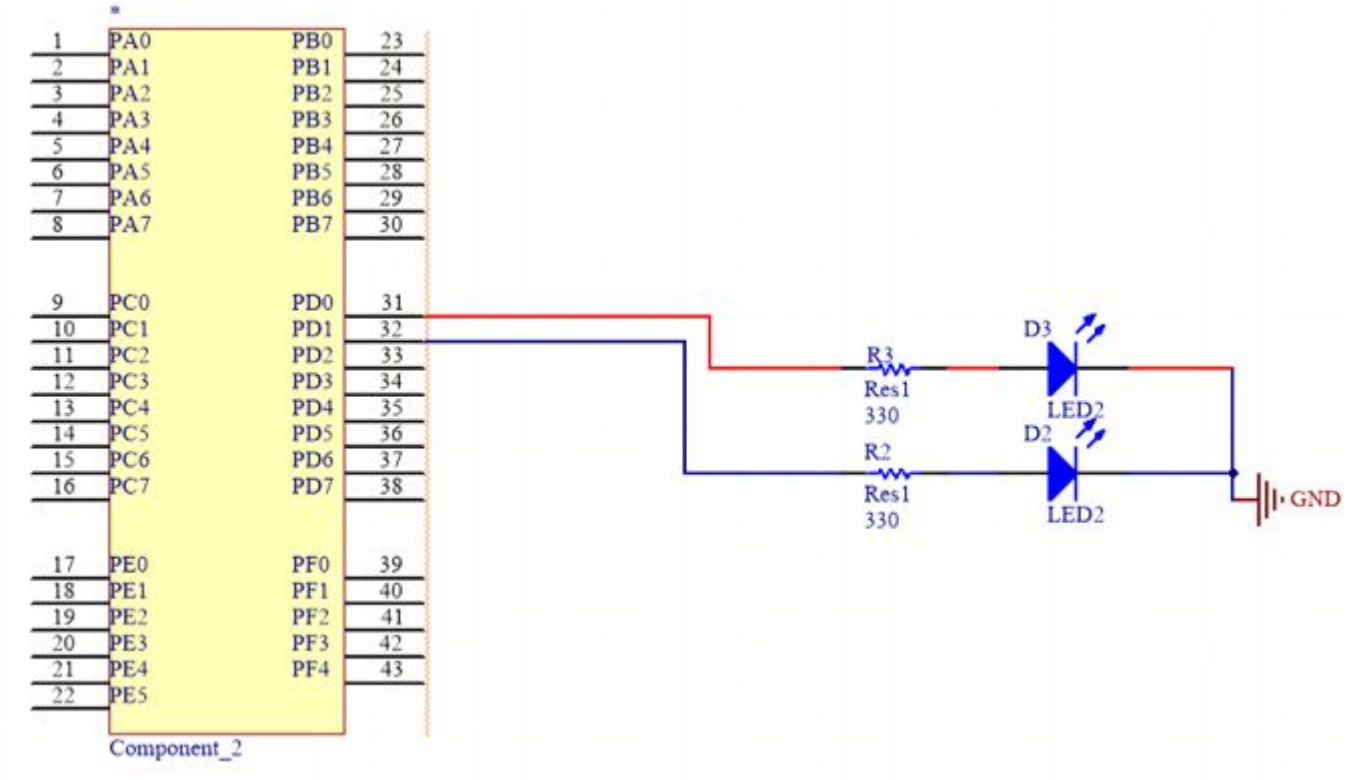
# Elektronik Devre Kurulumu



# Elektronik Devre Kurulumu



# Elektronik Devre Kurulumu



# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"

int main() {
        volatile unsigned long delay;

}
```

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {




}
int main() {
        volatile unsigned long delay;
        init_port_D();




}
```

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;




}
int main() {
        volatile unsigned long delay;
        init_port_D();




}
```

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;




}
int main() {
        volatile unsigned long delay;
        init_port_D();









}
```

# Port D sayacını aktiflestir

```
SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;     // Port D sayacını aktifleştirir
```

| | |
|---|---|
| SYSCTL_RCGC2_R | x x x x  x x x x |
| SYSCTL_RCGC2_GPIOD | 0 0 0 0  1 0 0 0 |
| OR | |
| SYSCTL_RCGC2_R | x x x x  1 x x x |

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;


}
int main() {
        volatile unsigned long delay;
        init_port_D();









}
```

# Direction Register

```
GPIO_PORTF_DIR_R |= 0x0F;      // PD3, PD2, PD1, PD0 => Çıkış
```

| | |
|---|---|
| GPIO_PORTF_DIR_R | x x x x  x x x x |
| | 0 0 0 0  1 1 1 1 |
| OR | |
| GPIO_PORTF_DIR_R | x x x x  1 1 1 1 |

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;

}
int main() {
        volatile unsigned long delay;
        init_port_D();









}
```

# Alternative Function Register

```
GPIO_PORTD_AFSEL_R &= ~0x0F;      // PD3, PD2, PD1, PD0 => Alternatif fonksiyonları kapalı
```

| | |
|---|---|
| GPIO_PORTD_AFSEL_R | x x x x  x x x x |
| | ~(0 0 0 0  1 1 1 1) |
| | 1 1 1 1  0 0 0 0 |
| AND | |
| GPIO_PORTD_AFSEL_R | x x x x  0 0 0 0 |

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;
        GPIO_PORTD_DEN_R |= 0x0F;
}
int main() {
        volatile unsigned long delay;
        init_port_D();









}
```

# Digital Enable Register

```
GPIO_PORTD_DEN_R |= 0x0F;      // PD3, PD2, PD1, PD0 => digital IO
```

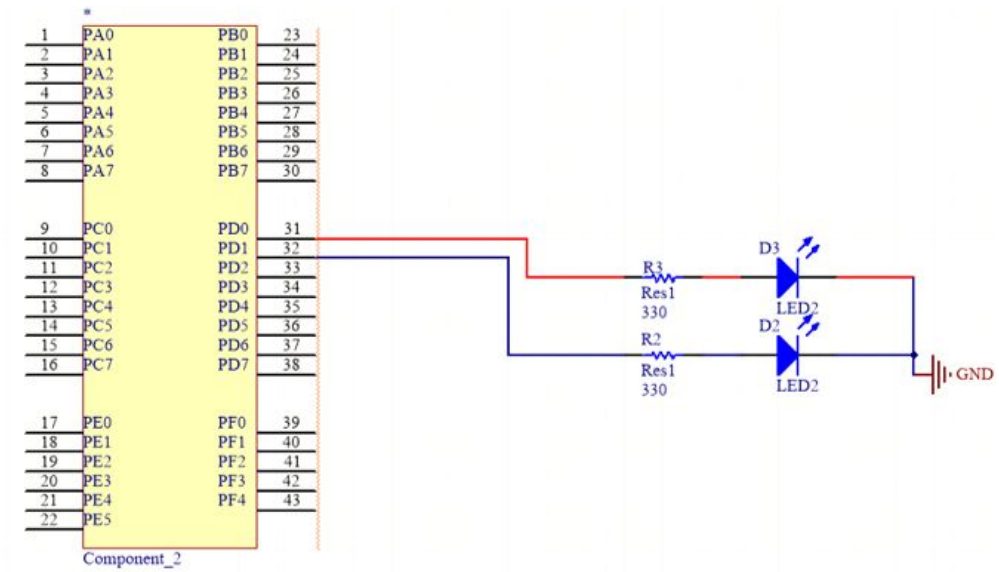
| | |
|---|---|
| GPIO_PORTD_DEN_R | x x x x  x x x x |
| | 0 0 0 0  1 1 1 1 |
| OR | |
| GPIO_PORTD_DEN_R | x x x x  1 1 1 1 |

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;
        GPIO_PORTD_DEN_R |= 0x0F;
}
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {









        }
}
```

# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;
        GPIO_PORTD_DEN_R |= 0x0F;
}
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                GPIO_PORTD_DATA_R |= 0b0010;
                GPIO_PORTD_DATA_R &= ~0b0001;




        }
}
```



# Data Register

```
GPIO_PORTD_DATA_R |= 0b0010;      // PD1'i 1 yap
```

| | |
|---|---|
| GPIO_PORTD_DATA_R | x x x x |
| | 0 0 1 0 |
| OR | |
| GPIO_PORTD_DATA_R | x x 1 x |

# Data Register

```
GPIO_PORTD_DATA_R |= 0b0010;      // PD1'i 1 yap
GPIO_PORTD_DATA_R &= ~0b0001;     // PD0'i 0 yap
```

GPIO_PORTD_DATA_R  x x 1 x

1 1 1 0

AND

---

GPIO_PORTD_DATA_R  x x 1 0

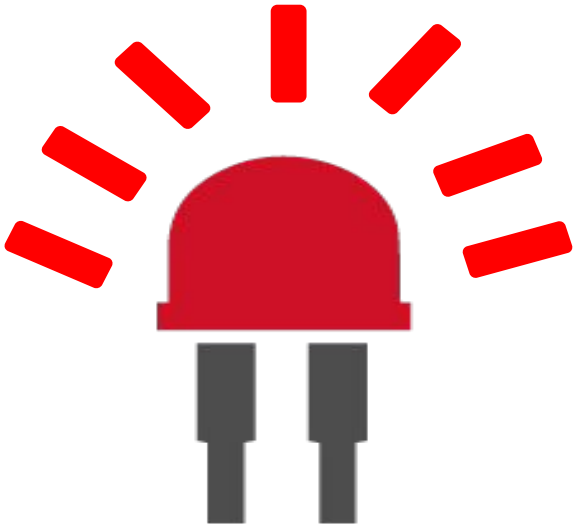
# Dışarıdaki 2 Ledi yakma

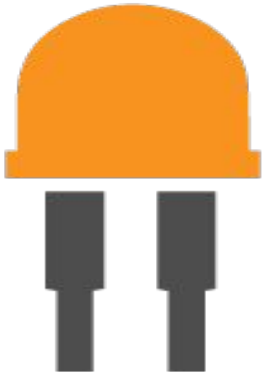
```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;
        GPIO_PORTD_DEN_R |= 0x0F;
}
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                GPIO_PORTD_DATA_R |= 0b0010;
                GPIO_PORTD_DATA_R &= ~0b0001;
                for (delay = 0 ; delay < 400000 ; delay++);



        }
}
```

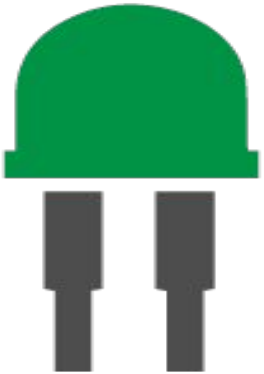
# Dışarıdaki 2 Ledi yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;
        delay = SYSCTL_RCGC2_R;
        GPIO_PORTD_DIR_R |= 0x0F;
        GPIO_PORTD_AFSEL_R &= ~0x0F;
        GPIO_PORTD_DEN_R |= 0x0F;
}
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                GPIO_PORTD_DATA_R |= 0b0010;
                GPIO_PORTD_DATA_R &= ~0b0001;
                for (delay = 0 ; delay < 400000 ; delay++);
                GPIO_PORTD_DATA_R |= 0b0001;
                GPIO_PORTD_DATA_R &= ~0b0010;
                for (delay = 0 ; delay < 400000 ; delay++);
        }
}
```

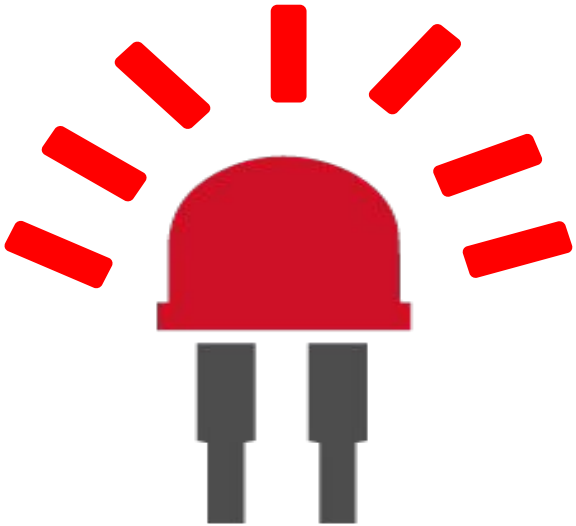
# Led ile trafik ışığı

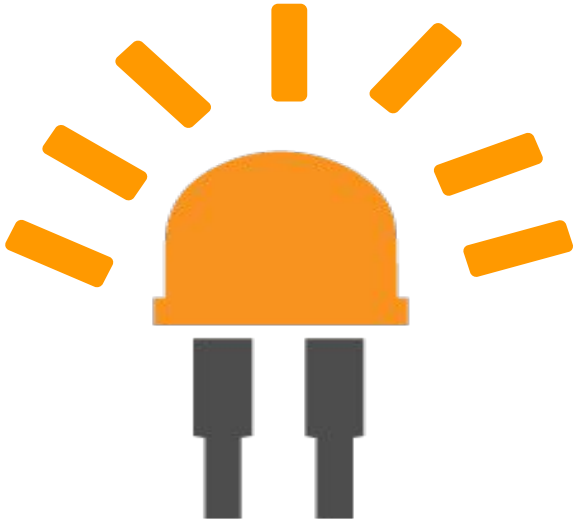


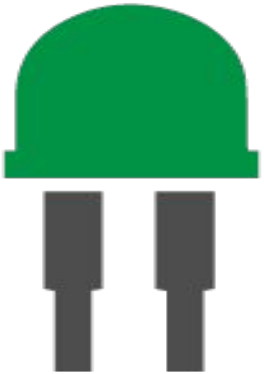




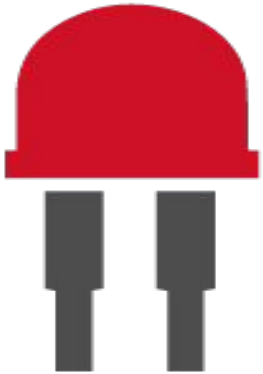
# Led ile trafik ışığı

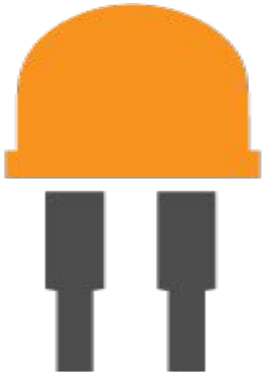


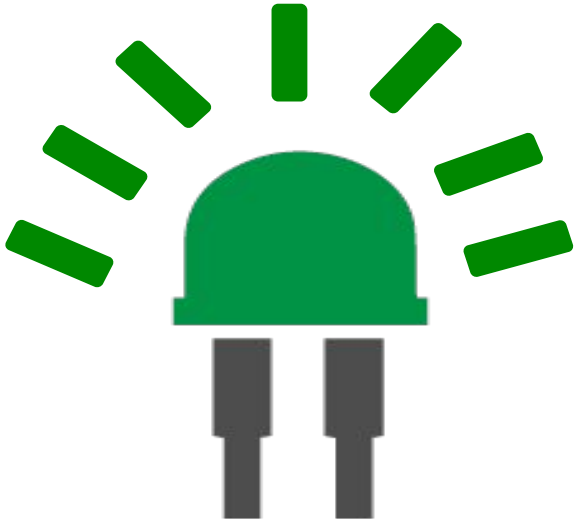




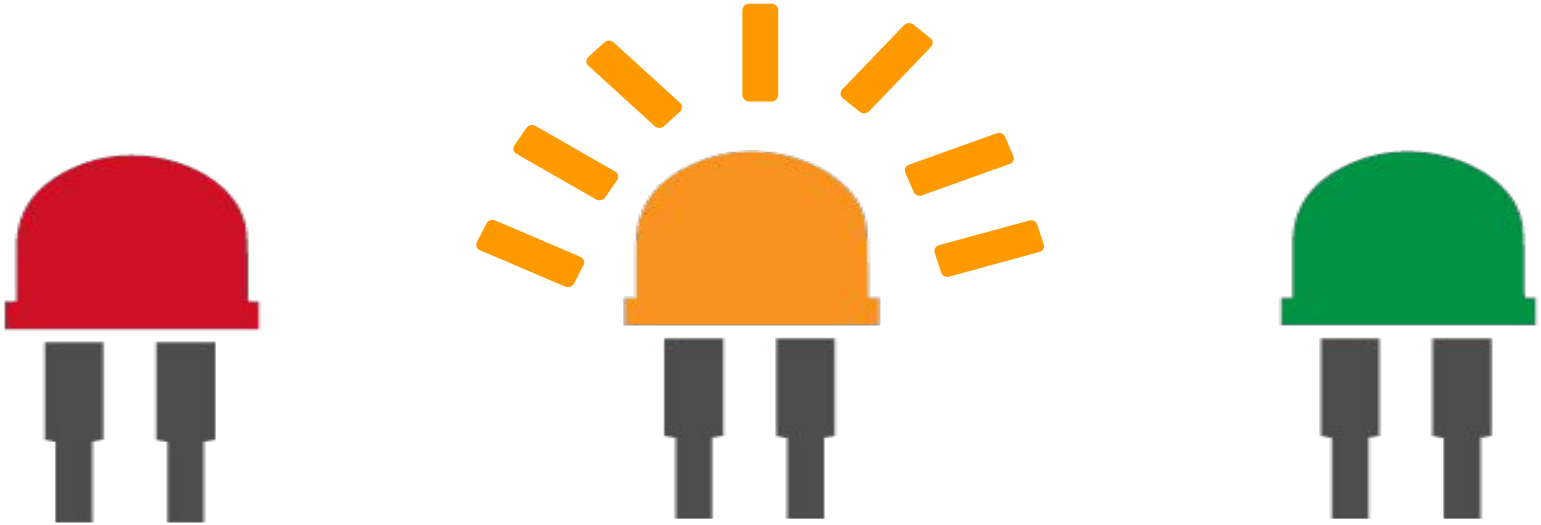
# Led ile trafik ışığı



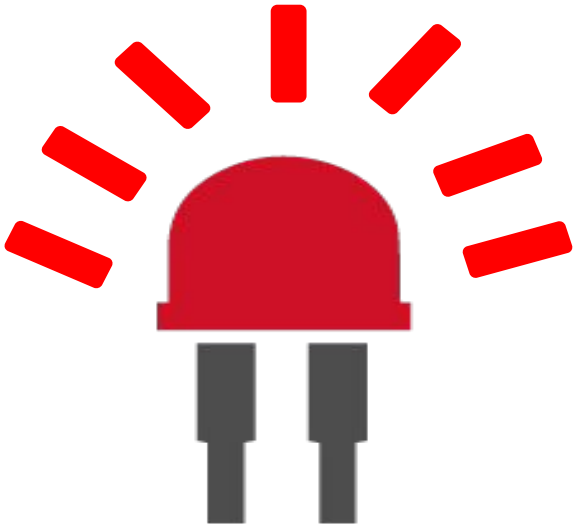


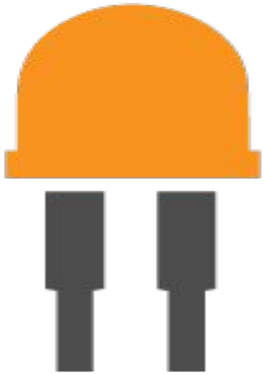


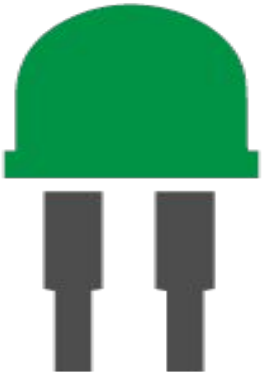
# Led ile trafik ışığı



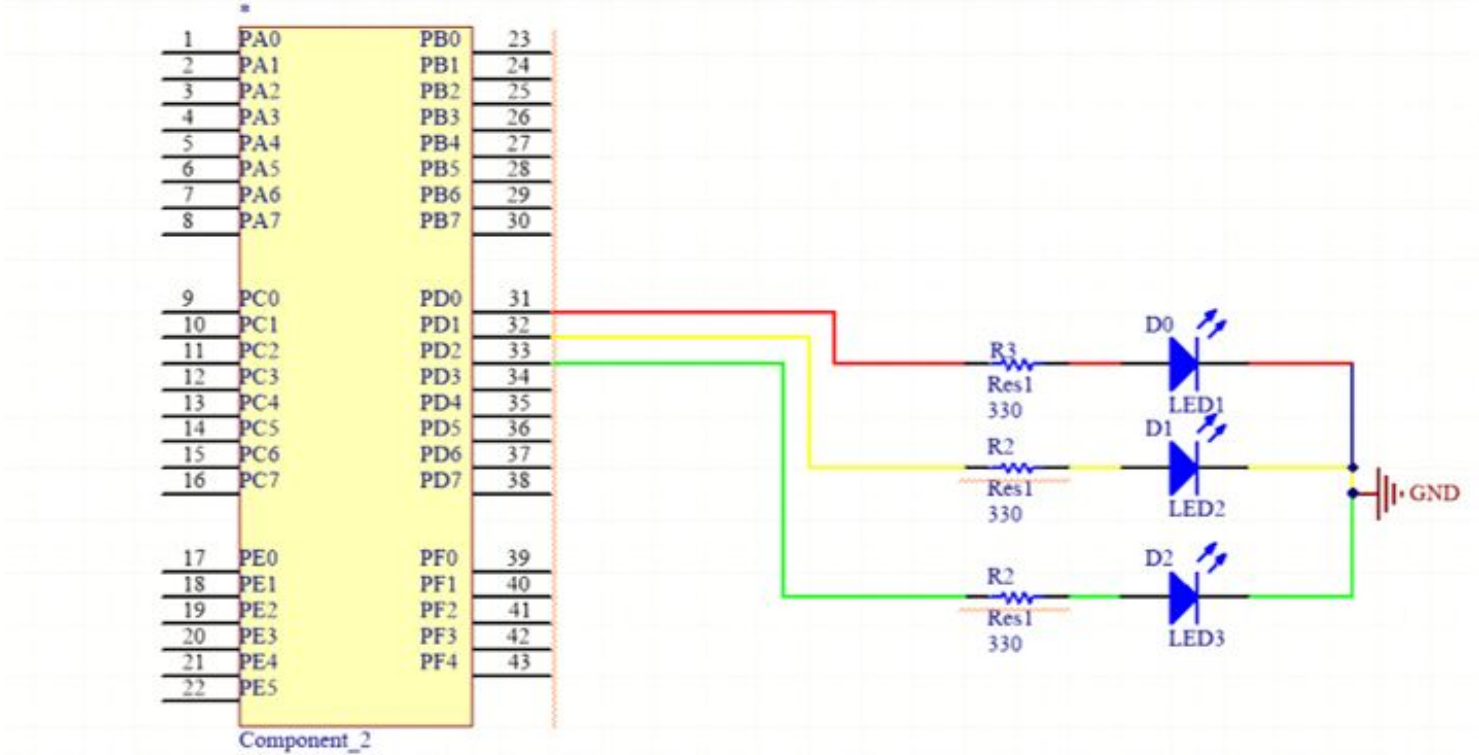
# Led ile trafik ışığı







# Led ile trafik ışığı



# Port D'nin aktiflestirilmesi

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;          // Port D'yi aktiflestir
        delay = SYSCTL_RCGC2_R;                        // zaman gecirmek icin
        GPIO_PORTD_DIR_R |= 0x0F;                      // PD 3,2,1,0 pinlerini cikis yap
        GPIO_PORTD_AFSEL_R &= ~0x0F;                   // PD 3,2,1,0 pinlerini alternatif fonksinunu 0 yap
        GPIO_PORTD_DEN_R |= 0x0F;                      // PD 3,2,1,0 pinlerini aktiflestir
}
```

# C Bit islemleri

```
#define BIT_SET(PORT, PIN) do {
        PORT |= 1<<PIN;
} while (0)
```

| | |
|---|---|
| PORT | x x x x |
| 1 << PIN | 0 0 1 0 |
| OR | |
| PORT | x x 1 x |

| | | |
|---|---|---|
| | | 0001 |
| PIN = 1 => | 1 << PIN | 0010 |
| PIN = 2 => | 1 << PIN | 0100 |
| PIN = 3 => | 1 << PIN | 1000 |

# C Bit islemleri

```
#define BIT_RESET(PORT, PIN) do {
        PORT &= ~(1<<PIN);
} while (0)
```

| | |
|---|---|
| PORT | x x x x |
| ~(1 << PIN) | 1 1 0 1 |
| AND | |
| PORT | x x 0 x |

| | | |
|---|---|---|
| | | 0001 |
| PIN = 1 => | 1 << PIN | 0010 |
| PIN = 2 => | 1 << PIN | 0100 |
| PIN = 3 => | 1 << PIN | 1000 |

# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {









}
```

# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {
        volatile unsigned long delay;
        init_port_D();









}
```

# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                BIT_SET(PORTD, 1);                  // KIRMIZI=1, SARI=0, YESIL=0
                BIT_RESET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);









        }
}
```

# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                BIT_SET(PORTD, 1);                  // KIRMIZI=1, SARI=0, YESIL=0
                BIT_RESET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_SET(PORTD, 1);                  // KIRMIZI=1, SARI=1, YESIL=0
                BIT_SET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);









        }
}
```

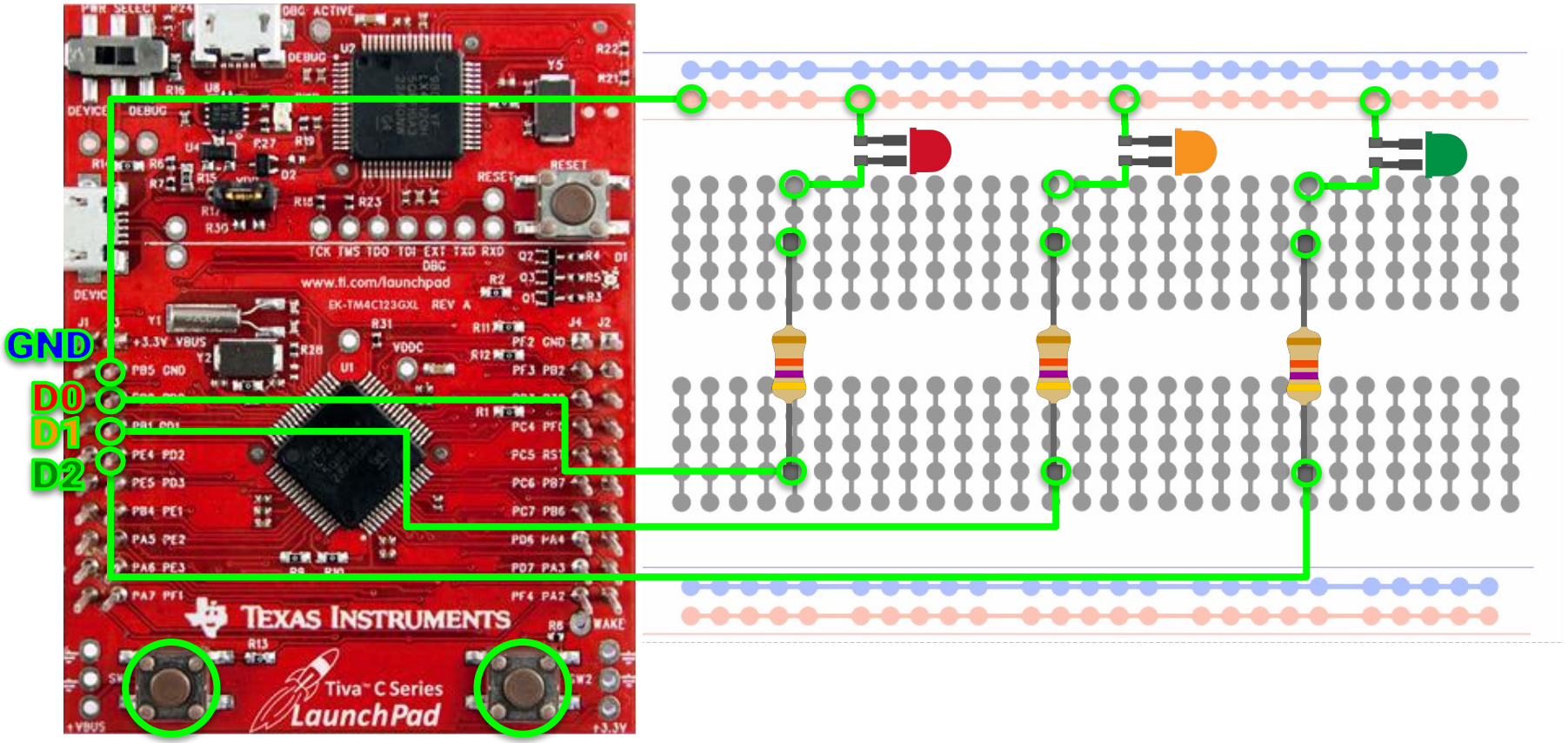
# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                BIT_SET(PORTD, 1);              // KIRMIZI=1, SARI=0, YESIL=0
                BIT_RESET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_SET(PORTD, 1);              // KIRMIZI=1, SARI=1, YESIL=0
                BIT_SET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_RESET(PORTD, 1);            // KIRMIZI=0, SARI=0, YESIL=1
                BIT_RESET(PORTD, 2);
                BIT_SET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);




        }
}
```

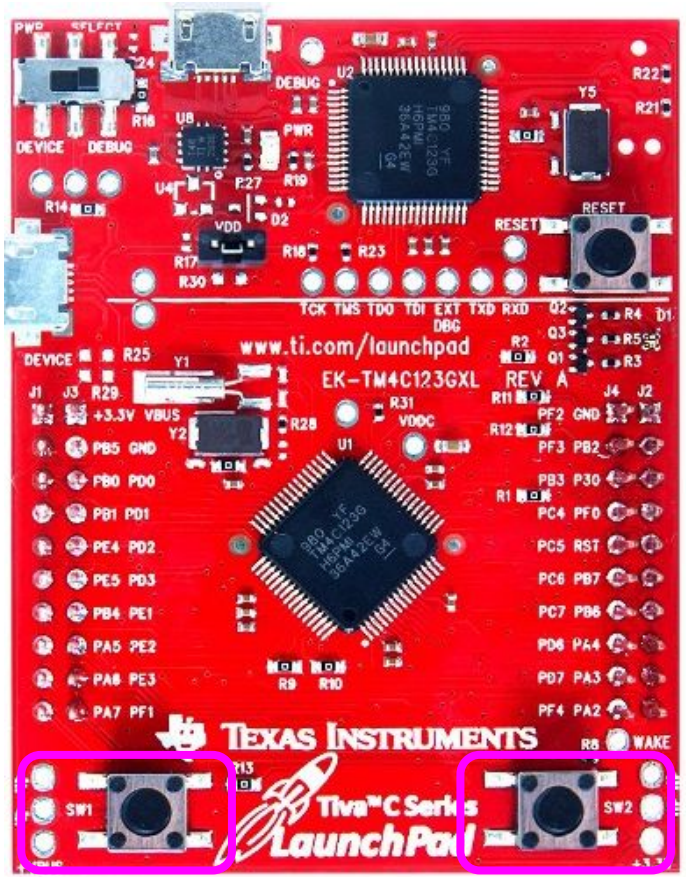
# Led ile trafik ışığı

```
#define PORTD (GPIO_PORTD_DATA_R)
int main() {
        volatile unsigned long delay;
        init_port_D();
        while (1) {
                BIT_SET(PORTD, 1);              // KIRMIZI=1, SARI=0, YESIL=0
                BIT_RESET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_SET(PORTD, 1);              // KIRMIZI=1, SARI=1, YESIL=0
                BIT_SET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_RESET(PORTD, 1);            // KIRMIZI=0, SARI=0, YESIL=1
                BIT_RESET(PORTD, 2);
                BIT_SET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
                BIT_RESET(PORTD, 1);            // SARI
                BIT_SET(PORTD, 2);
                BIT_RESET(PORTD, 3);
                for (delay = 0 ; delay < 4000000 ; delay++);
        }
}
```

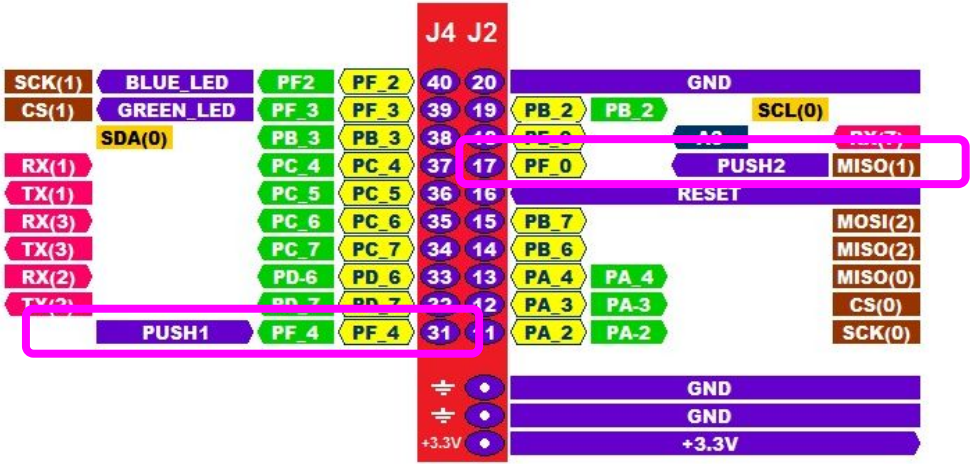
# Dısarıdaki 3 Ledi Butonlar ile yakma



# Tiva Port Baglantıları







# Dısarıdaki 3 Ledi Butonlar ile yakma

```
#include <stdint.h>
#include "inc/tm4c123gh6pm.h"
void init_port_D() {
        volatile unsigned long delay;
        SYSCTL_RCGC2_R |= SYSCTL_RCGC2_GPIOD;          // Port D'yi aktiflestir
        delay = SYSCTL_RCGC2_R;                        // Saatin başlaması için gecikme
        GPIO_PORTD_DIR_R |= 0x0F;                      // PD 3,2,1,0 pinlerini cikis yap
        GPIO_PORTD_AFSEL_R &= ~0x0F;                   // PD 3,2,1,0 pinlerini alternatif fonksinunu 0 yap
        GPIO_PORTD_DEN_R |= 0x0F;                      // PD 3,2,1,0 pinlerini aktiflestir
}
```

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
void init_port_F() {
  volatile unsigned long tmp;                  // bu degisken gecikme yapmak icin gerekli
  SYSCTL_RCGCGPIO_R |= 0x00000020;             // Port F'nin saatini aktifleştir
  tmp = SYSCTL_RCGCGPIO_R;                     // Saatin başlaması için gecikme
  GPIO_PORTF_LOCK_R = 0x4C4F434B;              // Port F GPIO kilidini aç



}
```

# Dışarıdaki 3 Ledi Butonlar ile yakma

```
void init_port_F() {
  volatile unsigned long tmp;                  // bu degisken gecikme yapmak icin gerekli
  SYSCTL_RCGCGPIO_R |= 0x00000020;             // Port F'nin saatini aktifleştir
  tmp = SYSCTL_RCGCGPIO_R;                     // Saatin başlaması için gecikme
  GPIO_PORTF_LOCK_R = 0x4C4F434B;              // Port F GPIO kilidini aç
  GPIO_PORTF_CR_R = 0x01;                      // PF4-0 kilidini aç (Sadece PF0 kilitlidir, diğer bitler kilitli değildir.)



}
```

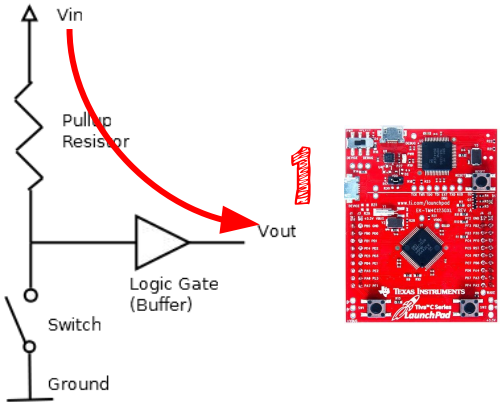
# Dışarıdaki 3 Ledi Butonlar ile yakma

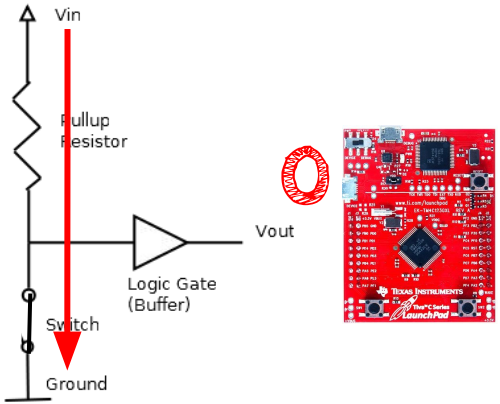
```
void init_port_F() {
  volatile unsigned long tmp;                  // bu degisken gecikme yapmak icin gerekli
  SYSCTL_RCGCGPIO_R |= 0x00000020;             // Port F'nin saatini aktifleştir
  tmp = SYSCTL_RCGCGPIO_R;                     // Saatin başlaması için gecikme
  GPIO_PORTF_LOCK_R = 0x4C4F434B;              // Port F GPIO kilidini aç
  GPIO_PORTF_CR_R = 0x01;                      // PF4-0 kilidini aç (Sadece PF0 kilitlidir, diğer bitler kilitli değildir.)
  GPIO_PORTF_DIR_R = 0x0E;                     // PF4,PF0 giriş, PF3-1 çıkış


}
```

```
GPIO_PORTF_DIR_R |= 0x0E;
GPIO_PORTF_DIR_R |= 0b000011110;   // PF4, PF0 giriş
```

# Dışarıdaki 3 Ledi Butonlar ile yakma

```
void init_port_F() {
  volatile unsigned long tmp;                  // bu degisken gecikme yapmak icin gerekli
  SYSCTL_RCGCGPIO_R |= 0x00000020;             // Port F'nin saatini aktifleştir
  tmp = SYSCTL_RCGCGPIO_R;                     // Saatin başlaması için gecikme
  GPIO_PORTF_LOCK_R = 0x4C4F434B;              // Port F GPIO kilidini aç
  GPIO_PORTF_CR_R = 0x01;                      // PF4-0 kilidini aç (Sadece PF0 kilitlidir, diğer bitler kilitli değildir.)
  GPIO_PORTF_DIR_R = 0x0E;                     // PF4,PF0 giriş, PF3-1 çıkış
  GPIO_PORTF_PUR_R = 0x11;                     // PF0 ve PF4 üzerindeki pull-up direncini aktifleştir

}
```



# Dışarıdaki 3 Ledi Butonlar ile yakma

```
void init_port_F() {
  volatile unsigned long tmp;                  // bu degisken gecikme yapmak icin gerekli
  SYSCTL_RCGCGPIO_R |= 0x00000020;             // Port F'nin saatini aktifleştir
  tmp = SYSCTL_RCGCGPIO_R;                     // Saatin başlaması için gecikme
  GPIO_PORTF_LOCK_R = 0x4C4F434B;              // Port F GPIO kilidini aç
  GPIO_PORTF_CR_R = 0x01;                      // PF4-0 kilidini aç (Sadece PF0 kilitlidir, diğer bitler kilitli değildir.)
  GPIO_PORTF_DIR_R = 0x0E;                     // PF4,PF0 giriş, PF3-1 çıkış
  GPIO_PORTF_PUR_R = 0x11;                     // PF0 ve PF4 üzerindeki pull-up direncini aktifleştir
  GPIO_PORTF_DEN_R = 0x1F;                     // PF4-0 digital I/O aktifleştir
}
```

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
   init_port_D();
   init_port_F();
   int button_sag, button_sol;
```

sagdaki button 0. bit, soldaki button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

```
}
```

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
   init_port_D();
   init_port_F();
   int button_sag, button_sol;
   while (1) {









   }
}
```

sagdaki button 0. bit, soldaki button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
   init_port_D();
   init_port_F();
   int button_sag, button_sol;
   while (1) {
          button_sag = GPIO_PORTF_DATA_R & 0b00001;          // sagdaki buttonun degerini oku, degiskene degerini ata
          button_sol = GPIO_PORTF_DATA_R & 0b10000;          // soldaki buttonun degerini oku, degiskene degerini ata
```

sagdaki button 0. bit, soldaki button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

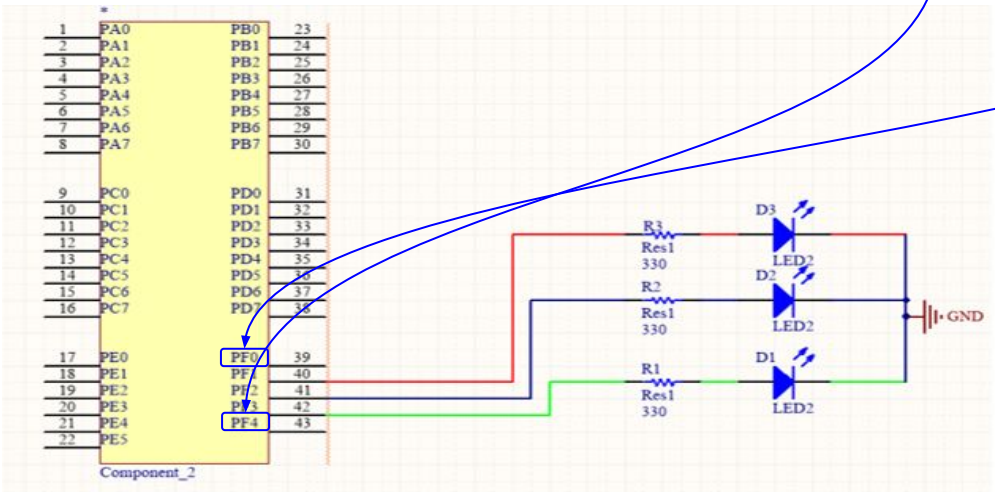
```
GPIO_PORTF_DIR_R |= 0b00001110;   // PF4,PF0 giriş
```

```
   }
}
```

# Data Register

GPIO_PORTF_DATA_R

| 31 | 30 | 29 | 28 | 27 | . . . | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| x | x | x | x | x | . . . | ? | 0 | 0 | 0 | ? |



# Data Register

```
button_sag = GPIO_PORTF_DATA_R & 0b00001;      // sagdaki buttonun degerini oku, degiskene degerini ata
```

| | |
|---|---|
| GPIO_PORTF_DATA_R | x x x x x |
| | 0 0 0 0 1 |
| AND | |
| button_sag | 0 0 0 0 x |

# Data Register

```
button_sol = GPIO_PORTF_DATA_R & 0b10000;      // soldaki buttonun degerini oku, degiskene degerini ata
```

| | |
|---|---|
| GPIO_PORTF_DATA_R | x x x x x |
| | 1 0 0 0 0 |
| AND | |
| button_sol | x 0 0 0 0 |

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
  init_port_D();
  init_port_F();
  int button_sag, button_sol;
  while (1) {
        button_sag = GPIO_PORTF_DATA_R & 0b00001;        // sagdaki buttonun degerini oku, degiskene degerini ata
        button_sol = GPIO_PORTF_DATA_R & 0b10000;        // soldaki buttonun degerini oku, degiskene degerini ata
       if(button_sol == 0 && button_sag == 0){
              GPIO_PORTD_DATA_R |= 0b01110;              // PD(1,2,3)'ü 1 yap
        }
  }
}
```

sagdaki(SW2) button 0. bit, soldaki(SW1) button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
  init_port_D();
  init_port_F();
  int button_sag, button_sol;
  while (1) {
        button_sag = GPIO_PORTF_DATA_R & 0b00001;        // sagdaki buttonun degerini oku, degiskene degerini ata
        button_sol = GPIO_PORTF_DATA_R & 0b10000;        // soldaki buttonun degerini oku, degiskene degerini ata
       if(button_sol == 0 && button_sag == 0){
              GPIO_PORTD_DATA_R |= 0b01110;              // PD(1,2,3)'ü 1 yap
        }else if (button_sag == 0) {                     // SW2 basılı ise
              GPIO_PORTD_DATA_R |= 0b00010;              // PD1'i 1 yap
              GPIO_PORTD_DATA_R &= ~(0b0100);            // PD2'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);           // PD3'i 0 yap
        }


  }
}
```

sagdaki(SW2) button 0. bit, soldaki(SW1) button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

# Dısarıdaki 3 Ledi Butonlar ile yakma

```
int main(void) {
  init_port_D();
  init_port_F();
  int button_sag, button_sol;
  while (1) {
        button_sag = GPIO_PORTF_DATA_R & 0b00001;      // sagdaki buttonun degerini oku, degiskene degerini ata
        button_sol = GPIO_PORTF_DATA_R & 0b10000;      // soldaki buttonun degerini oku, degiskene degerini ata
       if(button_sol == 0 && button_sag == 0){
              GPIO_PORTD_DATA_R |= 0b01110;            // PD(1,2,3)'ü 1 yap
        }else if (button_sag == 0) {                   // SW2 basılı ise
              GPIO_PORTD_DATA_R |= 0b00010;            // PD1'i 1 yap
              GPIO_PORTD_DATA_R &= ~(0b0100);          // PD2'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);         // PD3'i 0 yap
        }else if (button_sol == 0) {                   // SW1 basili ise
              GPIO_PORTD_DATA_R |= 0b00100;            // PD2'i 1 yap
              GPIO_PORTD_DATA_R &= ~(0b00010);         // PD1'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);         // PD3'i 0 yap
        }



  }
}
```

sagdaki(SW2) button 0. bit, soldaki(SW1) button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

# Dışarıdaki 3 Ledi Butonlar ile yakma

sagdaki(SW2) button 0. bit, soldaki(SW1) button 4. bit
button basiliyken 0, basili degilken 1 degerini alır.

```
int main(void) {
  init_port_D();
  init_port_F();
  int button_sag, button_sol;
  while (1) {
        button_sag = GPIO_PORTF_DATA_R & 0b00001;        // sagdaki buttonun degerini oku, degiskene degerini ata
        button_sol = GPIO_PORTF_DATA_R & 0b10000;        // soldaki buttonun degerini oku, degiskene degerini ata
       if(button_sol == 0 && button_sag == 0){
              GPIO_PORTD_DATA_R |= 0b01110;              // PD(1,2,3)'ü 1 yap
        }else if (button_sag == 0) {                     // SW2 basılı ise
              GPIO_PORTD_DATA_R |= 0b00010;              // PD1'i 1 yap
              GPIO_PORTD_DATA_R &= ~(0b0100);            // PD2'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);           // PD3'i 0 yap
        }else if (button_sol == 0) {                     // SW1 basili ise
              GPIO_PORTD_DATA_R |= 0b00100;              // PD2'i 1 yap
              GPIO_PORTD_DATA_R &= ~(0b00010);           // PD1'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);           // PD3'i 0 yap
        }else {
               GPIO_PORTD_DATA_R &= ~(0b00100);          // PD1'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b00010);           // PD2'i 0 yap
              GPIO_PORTD_DATA_R &= ~(0b01000);           // PD3'i 0 yap
        }
  }
}
```

# Sorular