HW1

Task1 : Sıcaklık_Ayarı

- ⇨ Bu taskın işlevi porttan okunan termometre değerine göre Heater On/Off butonunu yönetmektir.

- ⇨ Task1 iş akışı ise aşağıdaki gibidir:

- ⇨ adc_thermometer_port periodik olarak dinlenir.
- ⇨ Porttan gelen değer okunur.
- ⇨ Okunan değere göre sıcaklık olması gereken değerin altında ise Heater On yapılarak mutex ile kontrollü bir şekilde son_sıcaklık değeri set edilir.
- ⇨ Diğer durum için ise Heater Off yapılarak son_sıcaklık değeri set edilir.
- ⇨ Yukarda geçen işlemlerin zamanını t diye atayalım,sleep ile verilen periyot((@10Hz) – t) kadar bekletilir.
- ⇨ Konsol_Gösterimi taskı ile senkron,Basınç_Ayarı taskı ile asenkron çalışır.

Task2 : Basınç_Ayarı

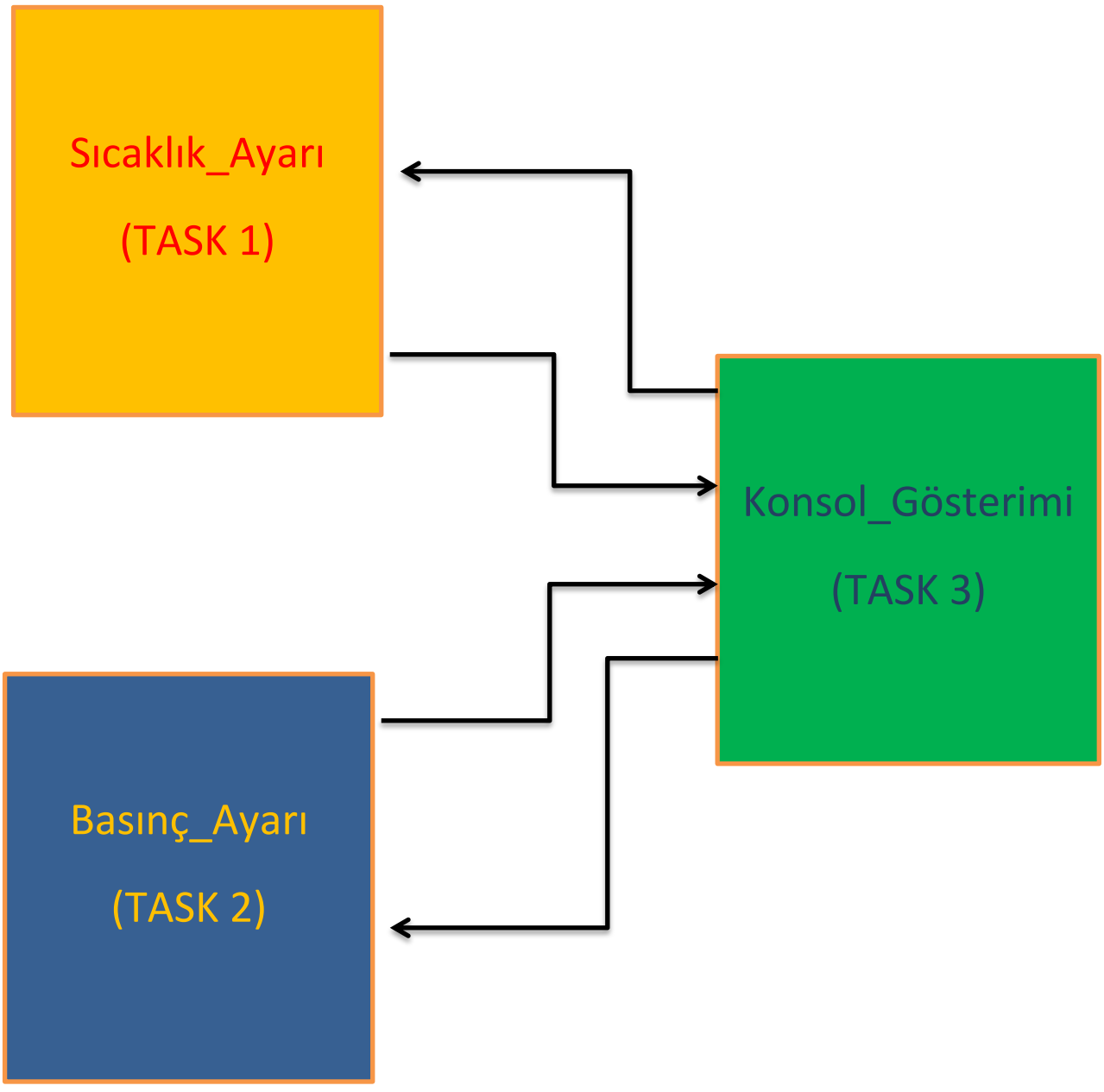
- ⇨ Bu taskın işlevi porttan okunan basınç değerine göre Pump'a verilen DAC değerini yönetmektir.

- ⇨ Task2 iş akışı ise aşağıdaki gibidir:

- ⇨ adc_pressure_port periodik olarak dinlenir.
- ⇨ Porttan gelen değer okunur. .
- ⇨ Okunan değere göre basınç olması gereken değerin altında ise Pump DAC değeri artırılarak mutex ile kontrollü bir şekide son_basınç değeri set edilir.
- ⇨ Diğer durum için ise Pumb DAC değeri azaltılarak son_basınç değeri set edilir.
- ⇨ Yukarda geçen işlemlerin zamanını t diye atayalım,sleep ile verilen periyot((@100Hz) – t) kadar bekletilir.
- ⇨ Konsol_Gösterimi taskı ile senkron,Sıcaklık_Ayarı taskı ile asenkron çalışır.

Task3 : Konsol_Gösterimi

- ⇨ Bu taskın işlevi global olarak tanımlanan heater ve pump değerlerini display'e periodik olarak print eder.
- ⇨ Task3 iş akışı ise aşağıdaki gibidir:
- ⇨ Globaldeki son_sıcaklık ve son_basınç değerleri mutex ile kontrollü bir şekilde okunarak ekrana print edilir.
- ⇨ Belirlenen perioda göre her 10 ms'de ekrana basılır.



Sıcaklık_Ayarı Pseudocode (Task1)

```
for (;;) {
    t1=Now( );
    adc_trigger(adc_thermometer_port);
    sıcaklık  =  read_adc(adc_ thermometer _port);
    cout << Read thermometer value:" << sıcaklık << endl;
    if (heater<eşik_değeri){
        heater  =  true(On);
    }
    else {
        heater  =  false(Off);
    }
    Mutex_lock(&mutex1);
    Son_sıcaklık =  set_sıcaklık(heater);
    Mutex_unlock(&mutex1);
    t2  = Now();
    T  =  t2 – t1;
    sleep(10 – T);
}
```

Basınç_Ayarı Pseudocode (Task2)

```
    for (;;) {
        t1=Now( );
        adc_trigger(adc_pressure_port);
        basınç = read_adc(adc_pressure_port);
        cout << Read basınç value:" << basınç << endl;
        if (basınç<eşik_değeri){
             pressure  =  5.0;
        }
        else {
             pressure  =  0;
        }
        Mutex_lock(&mutex2);
        Son_basınç = set_basınc(pressure);
        Mutex_unlock(&mutex2);
         t2 = Now();
         T = t2 – t1;
         sleep(1 – T);
}
```

Konsol_Gösterimi Pseudocode (Task3)

```
for (;;) {
        t1=Now( );
            Mutex_lock(&mutex1);
                 Display(son_sıcaklık);
            Mutex_unlock(&mutex1);
            Mutex_lock(&mutex2);
                Display(son_basınç);
            Mutex_unlock(&mutex2);
            t2 = Now();
            T = t2 – t1;
            sleep(10 – T);
}
```