

**OBJECTIVE:** Keep gas temperature and pressure constant

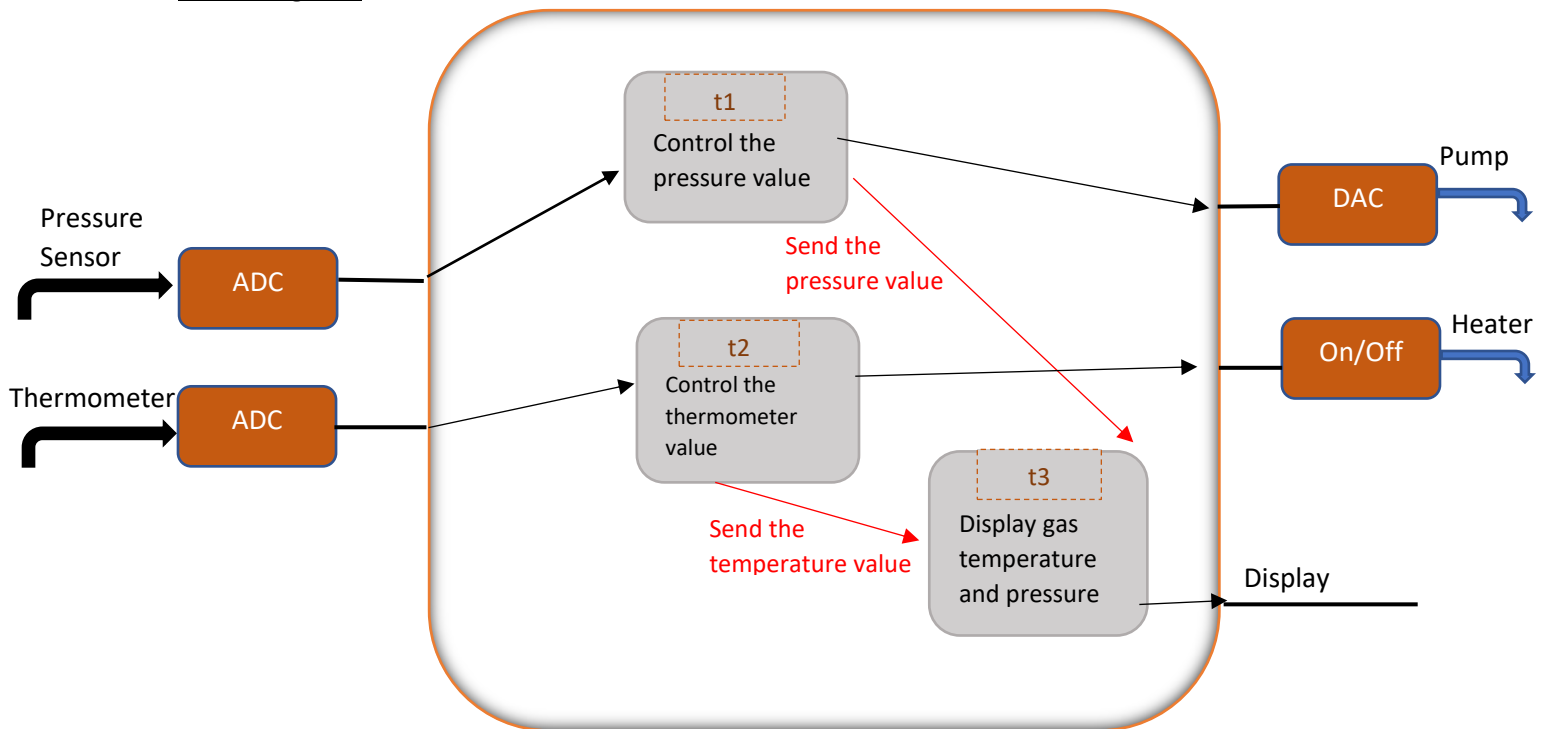
**Tasks:**

t1 – Pressure Control : periodic(10 ms)

t2 – Temperature Control: periodic(100 ms)

t3 – Display: periodic(10 ms)

**Block Diagram:**



**Synchronization and Communication:**

These 3 tasks run simultaneously. T1 and T2 change the global variable temperature and pressure value, t3 displays these values. I set the period of t3 to be equal to the minimum period - in this case the period of pressure control.

**Pseudocodes:**

```
// Task1: Pressure Control
```

```
for(;;){
```

```
    t_start = get_current_time();
```

```
    // Trigger ADC to take pressure value
```

```
    adc_trigger(adc_pressure_port);
```

```

// Read the current pressure value
cur_pressure = read_adc(adc_pressure_port);

// Control the pressure value
new_pressure = control_pressure(cur_pressure);

// Write this value to DAC
write_dac(new_pressure);

t_finish = get_current_time();

// Sleep to wait ADC trigger (100 Hz = 10 ms)
sleep(10 - (t_finish - t_start));
}

```

// Task2: Temperature Control

```

for(;;){

    t_start = get_current_time();

    // Trigger ADC to take temperature value
    adc_trigger(adc_temperature_port);

    // Read the current temperature value
    cur_temp = read_adc(adc_temperature_port);

    // Control the temperature value
    bool command = control_temperature(cur_temp);

    // Write this value to switch
    write_switch(command);

    t_finish = get_current_time();

    // Sleep to wait ADC trigger (10 Hz = 100 ms)
    sleep(100 - (t_finish - t_start));
}

```

// Task3: Display

```

for(;;){

    t_start = get_current_time();

    print_the_values();

    t_finish = get_current_time();

    // Sleep to wait ADC trigger (100 Hz = 10 ms)
    sleep(10 - (t_finish - t_start));
}

```