**Experiment No. 4**

**Aim:** To quantify ambient thermal parameters through analog (LM35) and digital (DHT11) temperature sensors.

**Simulator used:** Wokwi

**Circuit Layout:**

(a) (b)

Figure 1: (a) Interfacing LM35 with Arduino (b) Pin description of LM35

(a) (b)

Figure 2: (a) Interfacing DHT22 with Arduino (b) Pin description of DHT22

**Theory:**

***LM35:*** **To be completed by the student

***DHT22:***
**To be completed by the student

***DHT22 vs DHT11***

The DHT22 has a wider range and higher accuracy for both temperature and humidity measurements compared to the DHT11. The DHT22 can measure temperatures from -40°C to 80°C with an accuracy of ±0.5°C, while the DHT11 measures from 0°C to 50°C with an accuracy of ±2°C. For humidity, the DHT22 measures from 0% to 100% with an accuracy of 2-5%, and the DHT11 measures from 20% to 80% with an accuracy of 5%.

**Creation of a Custom Chip:**

- In your Wokwi project, click the "+" button
- Select "Custom Chip"
- Give your chip a name (e.g., "lm35")
- Choose C as the language

**Code used:**

***LM35:***

**Lm35.chip.json**

```
{
  "name": "LM35",
  "author": "Maverick",
  "pins": [
    "VCC",
    "OUT",
    "GND",
    "",
    ""
  ],
  "controls": [
    {
      "id": "temperature",
      "label": "Temperature",
      "type": "range",
      "min": 2,
      "max": 150,
      "step": 1
    }
  ]
}
```

**Lm35.chip.c**

```
/ Wokwi Custom Chip - For docs and examples see:
// https://docs.wokwi.com/chips-api/getting-started
//
// SPDX-License-Identifier: MIT
// Copyright 2023 Anshu Sharma

#include "wokwi-api.h"
#include <stdlib.h>

// Chip data.
typedef struct
{
  pin_t VCC;
  pin_t OUT;
  pin_t GND;
  uint32_t temperature;
} chip_data_t;

// Returns true if the power source is connected correctly.
bool power_connected(void *data)
{
```

```
  chip_data_t *chip = (chip_data_t*)data;
  return pin_read(chip->VCC) && !pin_read(chip->GND);
}

// Timer function. Analog output based on temperature.
void chip_timer_callback(void *user_data)
{
  if (power_connected(user_data))
  {
    chip_data_t *chip_data = (chip_data_t*)user_data;
    uint32_t temperature = attr_read(chip_data->temperature);
    float volts = 0.01 * temperature;
    pin_dac_write(chip_data->OUT, volts);
  }
}

// Chip initialization.
void chip_init(void)
{
  chip_data_t *chip_data = malloc(sizeof(chip_data_t));
  chip_data->VCC = pin_init("VCC", INPUT);
  chip_data->GND = pin_init("GND", INPUT);
  chip_data->OUT = pin_init("OUT", ANALOG);
  chip_data->temperature = attr_init("temperature", 50);

  const timer_config_t config =
  {
    .callback = chip_timer_callback,
    .user_data = chip_data,
  };

  timer_t timer_id = timer_init(&config);
  timer_start(timer_id, 10000, true);
}
```

***LM35***

```
#define ADC_VREF_mV    5000.0 // in millivolt
#define ADC_RESOLUTION 1024.0
#define PIN_LM35       A0

void setup() {
  Serial.begin(9600);
}

void loop() {
  // get the ADC value from the temperature sensor
  int adcVal = analogRead(PIN_LM35);
```

```
  // convert the ADC value to voltage in millivolt
  float milliVolt = adcVal * (ADC_VREF_mV / ADC_RESOLUTION);
  // convert the voltage to the temperature in Celsius
  float tempC = milliVolt / 10;
  // convert the Celsius to Fahrenheit
  float tempF = tempC * 9 / 5 + 32;

  // print the temperature in the Serial Monitor:
  Serial.print("Temperature: ");
  Serial.print(tempC);   // print the temperature in Celsius
  Serial.print("°C");
  Serial.print("  ~  "); // separator between Celsius and Fahrenheit
  Serial.print(tempF);   // print the temperature in Fahrenheit
  Serial.println("°F");

  delay(1000);
}
```

***DHT22***

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define DHTPIN 2     // Digital pin connected to the DHT sensor

//#define DHTTYPE DHT11
//#define DHTTYPE DHT21
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal_I2C lcd(0x27,16,2); //  0x3F or 0x27

void setup() {
  Serial.begin(9600);
  dht.begin();
  lcd.init();   // LCD initialize
  lcd.backlight();  // LCD backlight initialize
}

void loop() {
  delay(1000); // Wait a few seconds between measurements.

  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  lcd.setCursor(0,0); lcd.print("TEMP: "); lcd.print(t);
```

```
  lcd.setCursor(0,1); lcd.print("HUMID: "); lcd.print(h);

  Serial.print("Humidity: ");
  Serial.print(h,1);
  Serial.print("%  Temperature: ");
  Serial.print(t,1);
  Serial.print("C & ");
  Serial.print(f,1);
  Serial.println("F");
}
```

**Simulation Outcome:**

Figure 3: LM35 Temperature sensor simulation output

Figure 4: DHT22 sensor simulation output

**Result:**

**To be completed by the student