Министерство Образования и Исследований Республики Молдова

Технический Университет Молдовы

Факультет Вычислительной Техники, Информатики и Микроэлектроники

Департамент Программной Инженерии и Автоматики

**Лабораторная работа №3**

по предмету «Интернет вещи»

Выполнил: ст. гр. TI-196

Н. Шарафудинов

Проверил: А. Бырназ

Кишинев 2022

**Тема:** Получение информации. Датчики.

**Цель:** создать приложение на основе микроконтроллера.

**Задание:**

Создать приложение на основе микроконтроллера, которое будет принимать сигнал от источника сигнала, настраивать сигнал и отображать физический параметр на терминале (ЖК-дисплей и / или последовательный порт). Каждый студент выберет аналоговый или цифровой (не двоичный) датчик из прикрепленного PDF-файла или сайта: http://www.37sensors.com/.

1. Получить сигнал от датчика;

2. Кондиционировать сигнал с помощью цифровых фильтров и других методов;

3. Отображать данные на ЖК-дисплее и / или через последовательный пор.

**Теория**

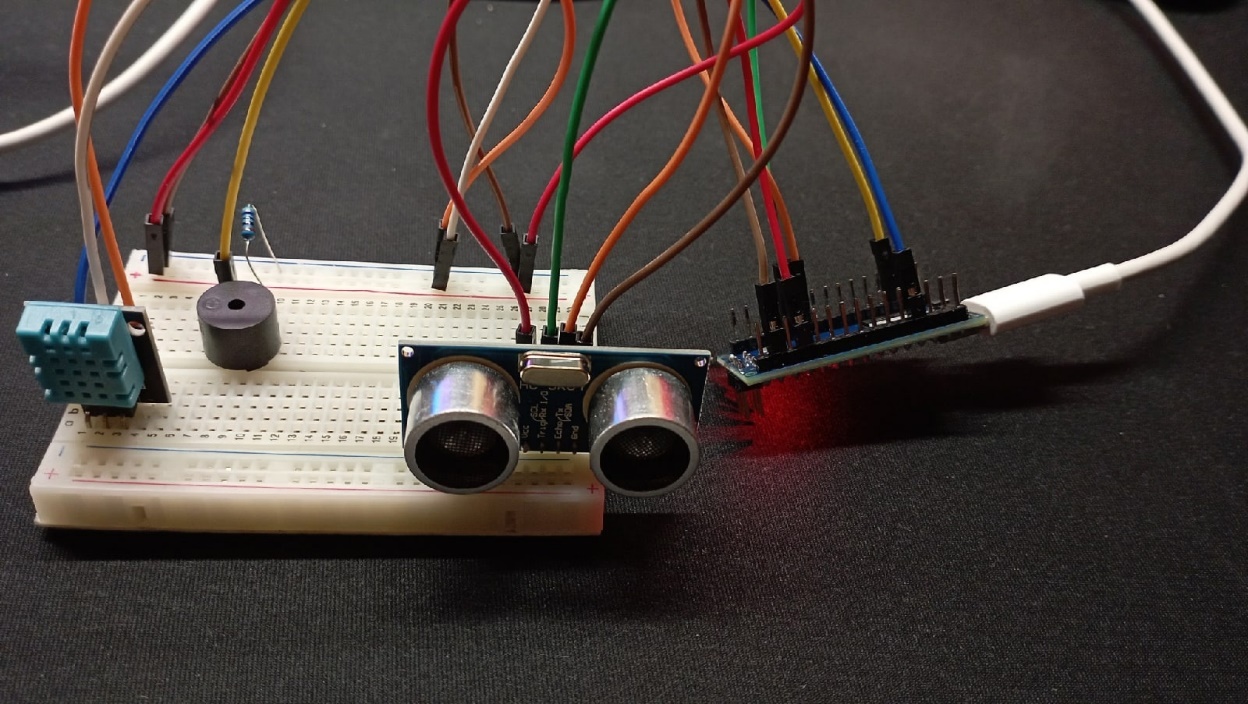
Датчик — конструктивно обособленное устройство, содержащее один или несколько первичных измерительных преобразователей. Датчик предназначен для выработки сигнала измерительной информации в форме, удобной для передачи, дальнейшего преобразования, обработки и (или) хранения, но не поддающейся непосредственному восприятию наблюдателем.

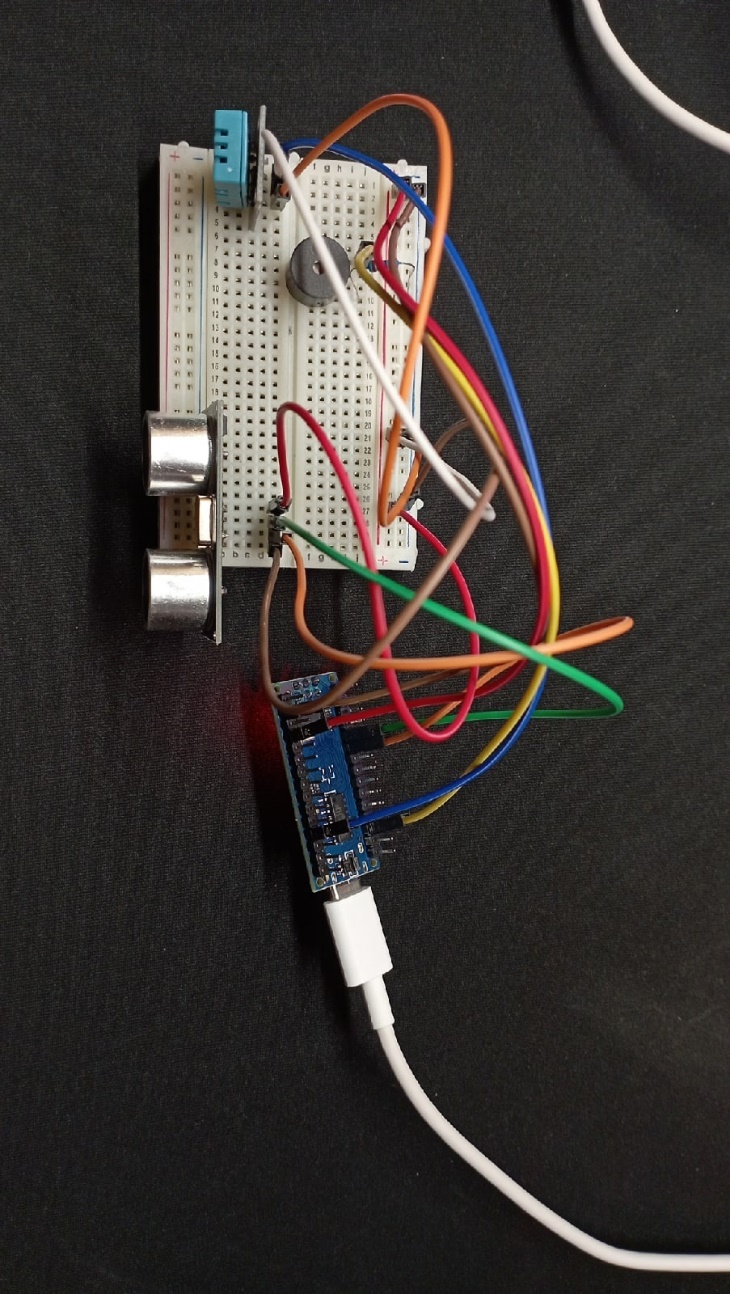
Датчик может дополнительно содержать промежуточные измерительные преобразователи, а также меру. Датчик может быть вынесен на значительное расстояние от устройства, принимающего его сигналы. При нормированном соотношении значения величины на выходе датчика с соответствующим значением входной величины датчик является средством измерений.

В настоящее время термины «датчик» и «сенсор» используются как равнозначные для обозначения измерительного преобразователя, выполняющего функции восприятия входной величины и формирования измерительного сигнала, хотя термин «сенсор» акцентирует внимание на восприятии входной величины, а термин «датчик» — на формировании и выдаче измерительного сигнала.

Многофункциональные датчики могут воспринимать и преобразовывать нескольких входных величин, и, помимо основной функции (восприятие величины и формирование измерительного сигнала) выполнять ряд дополнительных функций, таких как функции фильтрации, обработки сигналов и т. п.

Схема состоит из датчика температуры, Arduino nano, датчика расстояния, и buzzer.





**Вывод**

В данной лабораторной работе я познакомился с работой сенсоров в Arduino, использовав сенсор для получения значения температуры в окружающей среде. Также я ознакомился с работой цифровых фильтров, написав 2 фильтра для обработки сигнала: фильтр по среднему значению и соль и перец фильтр.

**Приложение**

MAIN

#include "termoSensor.h"

#include "rangerSensor.h"

#include "filterSaltAndPapper.h"

#include "filterAverage.h"

#include "stdio.h"

#include "pins.h"

static FILE serial\_stdinout = { .buf = NULL, .unget = 0, .flags = \_FDEV\_SETUP\_RW, .size = 0, .len = 0, .put = serial\_putchar, .get = serial\_getchar, .udata = 0 };

void setup() {

  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  pinMode(buzzer, OUTPUT);

  // Set up stdout and stdin

  stdout = &serial\_stdinout;

  stdin  = &serial\_stdinout;

  stderr = &serial\_stdinout;

}

void loop() {

  char msg[MSG\_LEN] = {0};

  printf("\nEnter [b,f,t] ");

  scanf("%s",msg);

  printf("\n");

  if(strcmp(msg, "b") == 0){

    while(true){

      Serial.flush();

      beeeep();

    }

  }

  if(strcmp(msg, "f") == 0){

    while(true){

      Serial.flush();

      filtres();

    }

  }

  if(strcmp(msg, "t") == 0){

    while(true){

      Serial.flush();

      TermoInfo();

    }

  }

}

void TermoInfo(){

  int i = 0;

  auto result = TermoHumiditySensor();

  printf("Humidity: %d %  \t ", (int)result.\_h );

  printf("Temperature: %d \ºC  %d \ºF\t", (int)result.\_t, (int)result.\_f);

  printf("Heat index: %d ºF\t ", (int)result.\_hi);

  printf("Error:");

  while(result.\_err[i] != '\0') {

    printf("%c", result.\_err[i++]);

  }

  printf("\n");

  i = 0;

}

void filtres(){

// salt  and papper filter

  int window = 3;

  int distance[SIZE];

  int snpf[SIZE];

  int medf[SIZE];

  for(int k = 0; k < SIZE; k++){

    printf("\n\t\tdistnce: \n");

    for(int i = 0; i < SIZE; i++){

      for(int j = 0; j < SIZE; j++){

          distance[j] = ranger(trigPin, echoPin);

      }

      printArray(distance, SIZE, 3);

      snpf[i] = filterSP(distance, SIZE);

    }

    printf("\n\tfiltered SnP:\n");

    printArray(snpf, SIZE, 2);

    medf[k] = smooth(snpf, SIZE, window);

  }

  printf("\nfiltered medf:\n");

  printArray(medf, SIZE, 1);

}

void printArray(int arr[], int size, int param)

{

  if (param == 1){

    for (int i = 0; i < size; i++)

        printf("%d \t", arr[i]);

  }

  if (param == 2){

    printf("\t");

    for (int i = 0; i < size; i++)

        printf("%d\t", arr[i]);

  }

  if (param == 3){

    printf("\t\t");

    for (int i = 0; i < size; i++)

        printf("%d\t", arr[i]);

  }

  printf("\n");

}

void beeeep(){

  if (ranger(trigPin, echoPin) <= 10)

  {

    tone(buzzer, 2000, 200);

  }

  else

  {

    noTone(buzzer);

  }

 }

TERMO SENSOR

.h

#pragma once

#include "DHT.h"

#define DHTPIN A0

#define DHTTYPE DHT11

#define pinRED  5

#define pinGREEN  4

#define pinBLUE  3

#define LEN  80

struct result {float \_h; float \_t; float \_f; float \_hi; char \*\_err;};

result TermoHumiditySensor();

.cpp

#include "termoSensor.h"

DHT dht(DHTPIN, DHTTYPE, 6);

unsigned long timing;

void setRGB(int R, int G, int B) {

  digitalWrite(pinRED,R);

  digitalWrite(pinGREEN,G);

  digitalWrite(pinBLUE,B);

}

result TermoHumiditySensor(){

  dht.begin();

  float h, t, f, hi;

  char err [] ="nothing";

  while(millis() - timing <= 1000){}

  timing = millis();

  h = dht.readHumidity();

  t = dht.readTemperature();

  f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {

    for(char& c : err) {

      c='\0';

    }

    strcat(err, "Failed to read from DHT sensor!");

    return;

  }

  hi = dht.computeHeatIndex(f, h);

  if (h < 40)

    setRGB(0,1,0);

  else if (h >= 40 && h < 60)

    setRGB(0,0,1);

  else

    setRGB(1,0,0);

  return result {h, t, f, hi, err};

}

DISTANCE SENSOR

.h

int ranger ( int trigPin, int echoPin){

  int duration, cm;

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  cm = duration / 58;

  delay(100);

  return cm;

}

FILTER SALT AND PAPPER

.h

#pragma once

int filterSP(int arr[], int len);

.cpp

void swap(int\* xp, int\* yp)

{

    int temp = \*xp;

    \*xp = \*yp;

    \*yp = temp;

}

// Function to perform Selection Sort

void selectionSort(int arr[], int n)

{

    int i, j, min\_idx;

    // One by one move boundary of unsorted subarray

    for (i = 0; i < n - 1; i++) {

        // Find the minimum element in unsorted array

        min\_idx = i;

        for (j = i + 1; j < n; j++)

            if (arr[j] < arr[min\_idx])

                min\_idx = j;

        // Swap the found minimum element

        // with the first element

        swap(&arr[min\_idx], &arr[i]);

    }

}

int filterSP(int arr[], int len){

  selectionSort(arr, len);

  return arr[2];

}

FILTER MEDIAN

.h

#pragma once

float smooth(int input[], int size, int window);

.cpp

float smooth(int input[], int size, int window)

{

  // do weights

  int\* weights = new int[size];

  int count = 0;

  for(int i = 0; i < size; i++){

    int t = input[i];

    for(int j = 0; j < size; j++){

      if(t == input[j]) count++;

    }

    weights[i] = count;

    count = 0;

  }

  int top = 0, bottom = 0;

  for(int j = 0; j < size; j++){

    top += weights[j] \* input[j];

    bottom += weights[j];

  }

  return top/bottom;

}

STDIO

.h

static int serial\_putchar(char c, FILE \*f)

{

  return Serial.write(c);

}

static int serial\_getchar(FILE \*)

{

  while(Serial.available() <= 0) {}

  return Serial.read();

}

PINS

.h

// echo/range sensor

 #define echoPin  3

 #define trigPin  2

 #define buzzer 10

 #define SIZE  5

 #define MSG\_LEN  5