

Подключение I2C датчика к LabView

Цель работы: Изучить технологии приема и отображения данных I2C устройства в LabView

Задача работы: Построение канала связи “LabView – I2C датчик”.

Приборы и принадлежности: Персональный компьютер, LabView, утилита COM Port Toolkit, контроллер Arduino UNO со средой программирования.

Введение

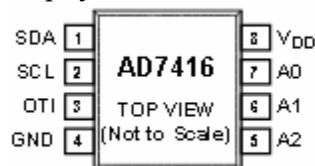
В этой работе предлагается вариант подключения датчика температуры AD7416 через контроллер Arduino UNO к среде построения виртуальных интерфейсов LabView. Показания датчика считываются контроллером через канал последовательной передачи данных I2C. Контроллер передает считанные данные интегрированной среде LabView по запросу. LabView вычисляет действительное значение температуры и отображает его на виртуальном индикаторе и графопостроителе. Предельная частота приема показаний датчика превышает 300 Гц.

ОБЩИЕ СВЕДЕНИЯ

Характеристика микросхемы AD7416 с встроенным датчиком температуры

Датчик температуры.

Разрядность	10 бит
Диапазон измерения	-40 .. 125 C
Точность	+/-1C при 25 C
Точность в диапазоне измерения	+/-2C
Разрешение измерения температуры	1/4 C
Время преобразования	30 мкс
Интерфейс	I2C
Напряжение питания микросхемы	2.7 .. 5.5В
Ток потребления	< 0.6 мА в нормальном режиме < 1.5 мкА в режиме пониженного потребления
Корпус	8-Lead Standard Small Outline Package (SOIC_N)



Адресный байт микросхемы формируется из фиксированной старшей части (0 1 0 0 1) и устанавливаемых бит A2, A1, A0 на выводах микросхемы. Подключение выводов микросхемы A2 = A1 = "0" (0В) и A0 = "1" (5В) устанавливает двоичный адрес 01001001₂, который соответствует шестнадцатеричному коду 0x49.

Вывод ОТИ микросхемы может использоваться для контурного релейного управления температурные пороги которого (T_{ОТИ} T_{HYST}) прописываются во внутренних регистрах микросхемы.

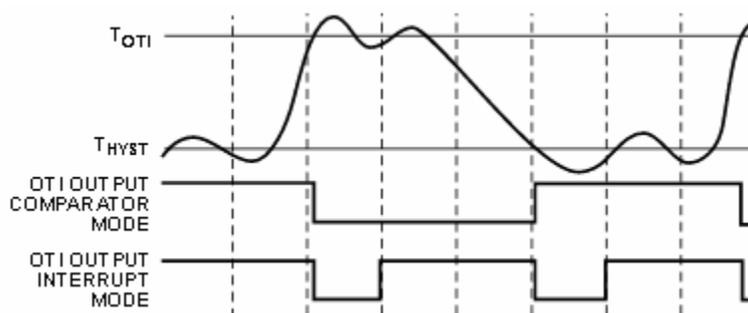


Рис. 1. Режимы формирования сигнала ОТИ микросхемы AD7416 для релейного управления температурой.

10-разрядный код температуры микросхемы AD7416 считывается в I2C режиме двумя байтами. Старший байт содержит целые значения температуры. Второй байт содержит дробную часть температуры с разрешением 1/4 C, причем, значащие 2 бита дробной части передаются старшими битами. Например, 32.25 градуса считываются как 32C = 00100000, 1/4C = 01000000. Знак отрицательной температуры передается старшим битом первого байта.

Подключение датчика через I2C канал к контроллеру Arduino

Схема подключения датчика температуры AD7416 к контроллеру Arduino показана на Рис. 2.

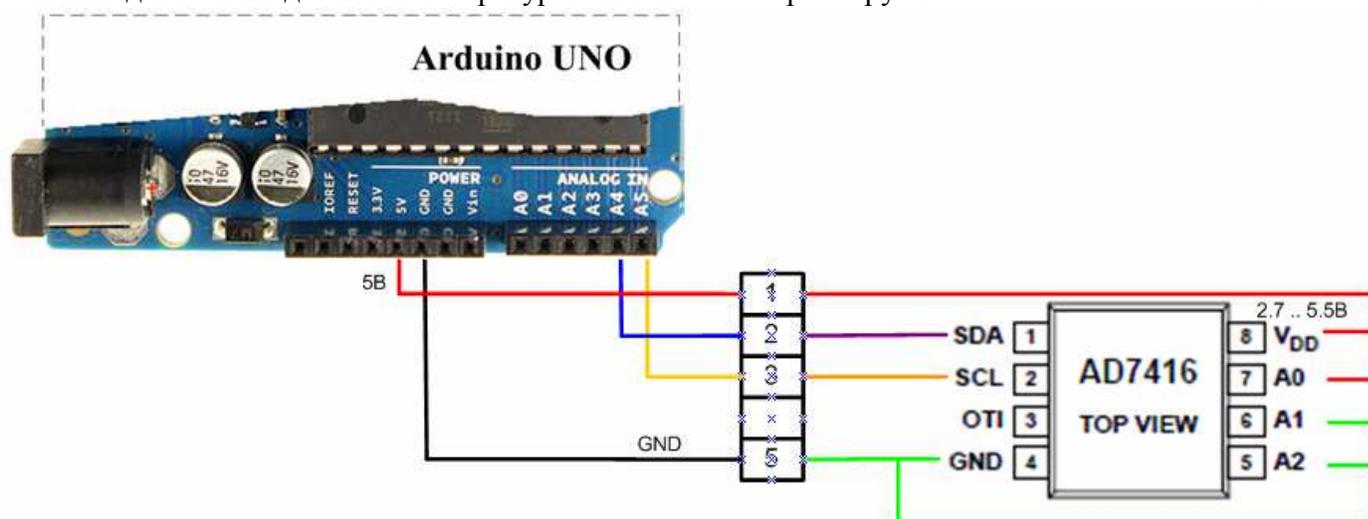


Рис. 2. Схема подключения датчика температуры AD7416 (адрес 0x49) к контроллеру Arduino по интерфейсу I2C.

Тестовая программа опроса датчика и передача данных датчика в COM порт включает следующие команды.

```
#include <Wire.h>

void setup()
{
  Wire.begin();    // подключение к шине i2c
  // Установите частоту 100 кГц (если она отличается),
  // Wire.beginTransmission(73); // начало процедуры передачи данных устройству по адресу 73 (0x49)
  // Wire.write(byte(val));    // отправляем байта данных
  // Wire.endTransmission();  // завершение процедуры передачи

  Serial.begin(9600); // запуск последовательного порта
}

void loop()
{
  Wire.requestFrom(73, 2); // запрос 2 байт от слейва #0x49 or 73
                          // Функция запрашивает данные у ведомого устройства (slave);
                          // 7-битный адрес ведомого устройства, у которого запрашиваются данные

  while(Wire.available()) // пока есть, что читать
  {
    int c = Wire.read(); // получаем байт
    Serial.print("t = ");
    Serial.print(c);    // передает в порт целое значение температуры
    Serial.print(" : ");
    c = Wire.read();    // получаем байт
    Serial.println(c);  // передает в порт дробную часть температуры
    // Serial.println("C");

  }

  delay(100);
}
```

Считанные байты датчика пересылаются контроллером в COM порт на частоте 9600 бит/с. Эти данные можно увидеть на внутреннем дисплее среды Arduino в следующем формате.

```
t = 29 : 192
t = 29 : 128
t = 29 : 128
t = 29 : 192
t = 30 : 0
t = 29 : 192
t = 30 : 0
t = 29 : 128
t = 29 : 192
t = 29 : 64
t = 29 : 64
t = 29 : 128
t = 29 : 128
```

Второй байт датчика имеет следующее отношение с дробной частью температуры.

0 == 0 (,00)
64 == 1/4 (,25)
128 == 2/4 (,50)
192 == 3/4 (,75)

Помимо внутреннего дисплея среды Arduino передаваемые в COM порт данные можно также увидеть и при помощи других программ, например, утилиты COM Port Toolkit (Рис. 3). Утилита показывает, что циклическая запись, например, "t = 29 : 192" передается 12-ю байтами.

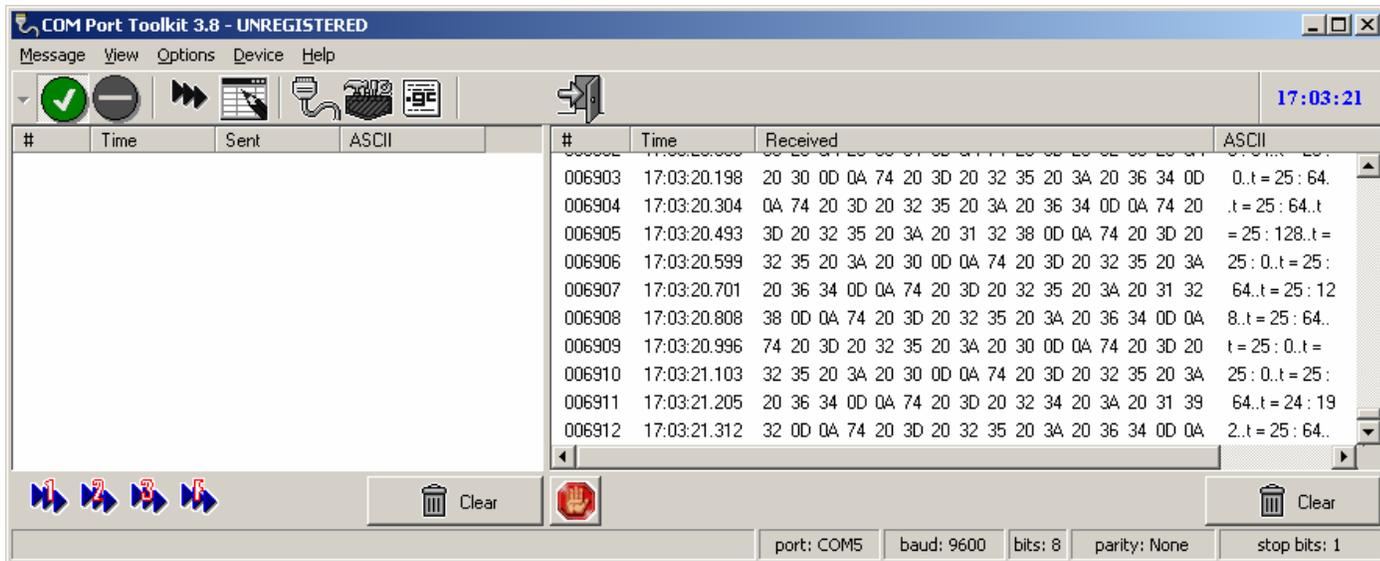


Рис. 3. Интерфейс программы COM Port Toolkit. В правой части интерфейса отображаются байты полученные из COM порта и их ASCII символы.

Передача данных контроллера в COM порт по запросу

Ниже показана программа контроллера, которая в цикле опрашивает COM порт и, при получении от внешнего устройства байта 49 (ASCII код 1), передает в COM порт считанный ранее первый байт датчика, а, при получении байта 50 (ASCII код 2), передает в порт считанный ранее второй байт датчика, считывает и запоминает 2 байта текущего показания датчика.

```
#include <Wire.h>

int I2C_first ; // старший (первый) байт I2C
int I2C_second; // младший (второй) байт I2C
int LV_mode;
// Pin 13 has the LED
int led = 13;

void setup()
{
  Wire.begin(); // подключение к шине I2C
```

```

pinMode(led, OUTPUT);
digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

Serial.begin(115200); // запуск последовательного порта на частоте 9600, 115200
}

void loop()
{
while (!Serial.available()); // ожидание запроса I2C данных
LV_mode = Serial.read(); // чтение байта запроса и синхронизации LabView
switch (LV_mode) {
case 49: { // 49 DEC, ASCII code = 1
Serial.print(I2C_first,DEC); // передает в порт целое значение температуры
break;
}
case 50: { // 50 DEC, ASCII code = 2
Serial.print(I2C_second,DEC); // передает в порт дробную часть температуры

// Чтение данных I2C канала для следующей пересылки по запросу
Wire.requestFrom(73, 2); // запрос 2 байт от слейва #0x49 or 73
// Функция запрашивает данные у ведомого устройства (slave);
while(Wire.available()) // пока есть, что читать
{
I2C_first = Wire.read(); // считываем первый байт
I2C_second = Wire.read(); // считываем второй байт
}
break;
}
}
}
}

```

Использование программы COM Port Toolkit для тестирования связи контроллер - COM порт путем формирования побайтных запросов контроллеру и отображения данных контроллера показано на Рис. 4.

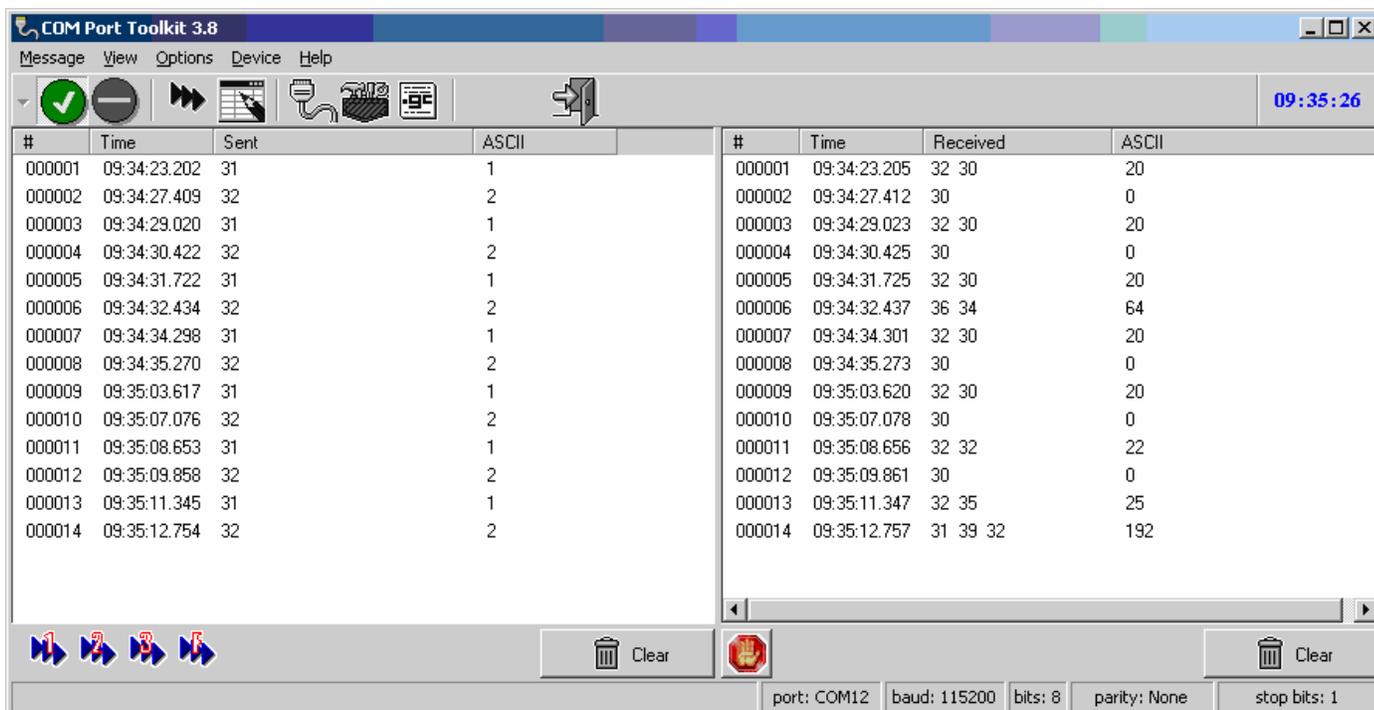


Рис. 4. Результат взаимодействия компьютера с контроллером по запросам через программу COM Port Toolkit. Программа настроена на частоту работы контроллера с COM портом - 115200 бит/с. В левой части интерфейса отображаются передаваемые в COM порт байты запроса (в шестнадцатеричном виде и ASCII кодах). В правой - байты контроллера принимаемые из COM порта.

Панель и структура виртуального прибора LabView для циклической генерации побайтных запросов, приема и отображения данных контроллера показана на Рис. 5. Прибор LabView инициализирует COM порт (блок "VISA Configure Serial Port"), очищает буфер (блок "VISA Clear") и с периодом 100 мс (блок "Wait Until Next ms Multiply") выполняет следующие операции:

- посылает в COM порт ASCII код 1 (блок "VISA Write");
- ожидает 10 мс (блок "Wait, ms");
- считывает количество байт, поступивших в буфер COM порта (блок "Property Node");
- считывает поступившие байты (блок "VISA Read");
- отображает на панели виртуального прибора строку принятых байт (блок "String");
- переводит строку в значение первого байта датчика температуры (блок "Decimal String To Number");
- отображает первый байт датчика на панели виртуального прибора (блок "Numeric");
- посылает первый байт (целая часть температуры) на сумматор (блок "Add");
- второй байт датчика температуры принимается по запросу и преобразуется в число с использованием таких же блоков, которые описаны выше для приема первого байта;
- дробная часть температуры получается делением второго байта на 256 (блок "Divider");
- полное значение температуры вычисляется сложением первого байта и дробной части (блок "Add");
- временная диаграмма показаний датчика температуры строится на графопостроителе (блок "Waveform Chart");

- взаимодействие LabView с контроллером прекращается в момент нажатия пользователем на кнопку "stop" (блок "Stop");
- после нажатия на кнопку "stop" блоки "VISA Close" и "Error" закрывают COM порт и завершают работу виртуального прибора.

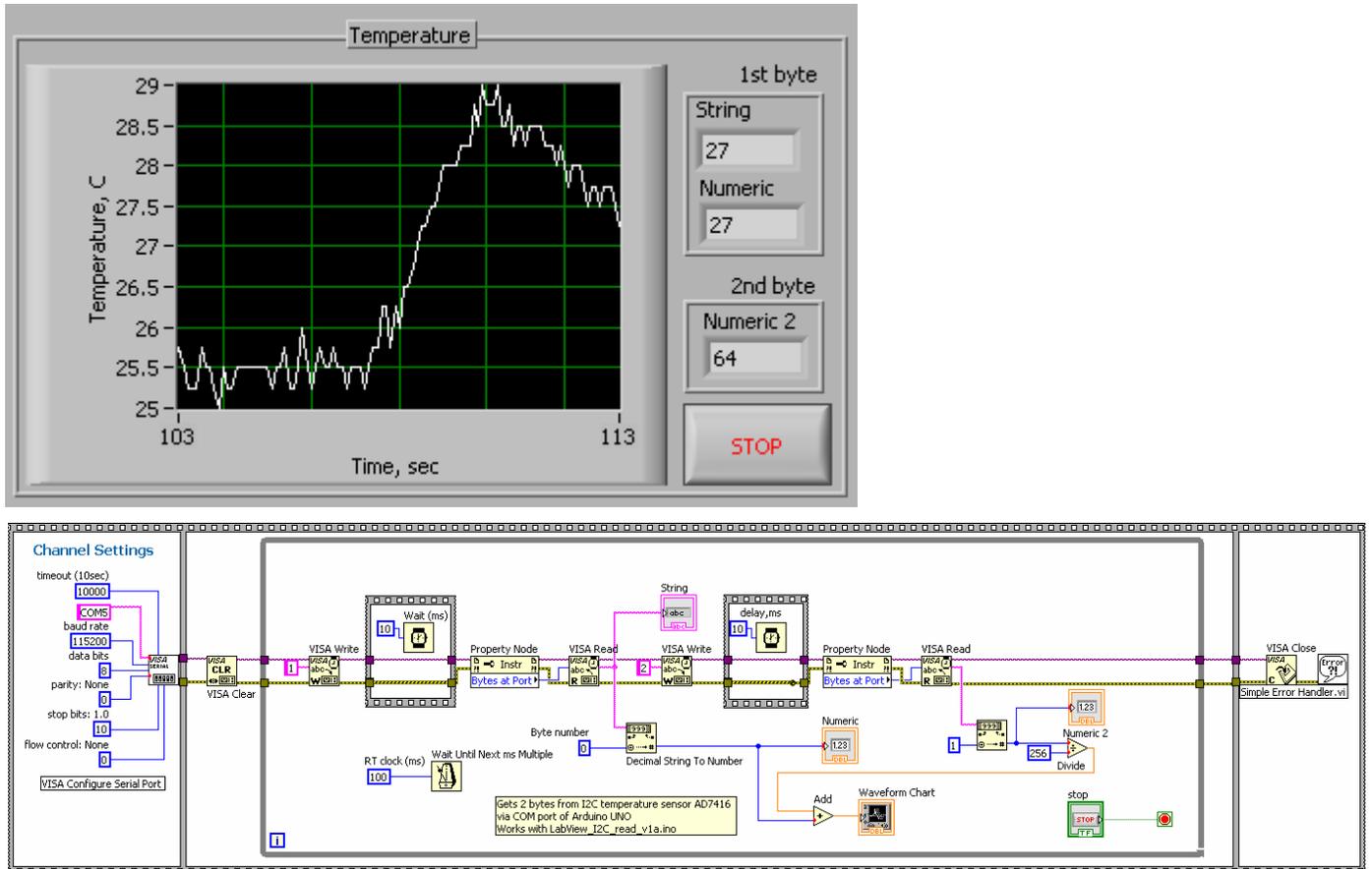


Рис. 5. Лицевая панель и структура виртуального прибора LabView для приема и отображения показаний датчика температуры. Каждый байт датчика принимается по индивидуальному запросу.

К недостаткам вышеописанной организации подключения I2C датчика температуры к LabView можно отнести задержку в 100 мс на ожидание такта передачи данных датчика и 20 мс на саму передачу. Передача контроллером байта датчика через COM порт по формату `Serial.print(XXX, DEC)` выполняется посимвольно - каждая цифра числа передается отдельным байтом.

Форматы передачи данных в COM порт

- `Serial.print(78, BIN)` выводит посимвольно "1001110"
- `Serial.print(78, OCT)` выводит "116"
- `Serial.print(78, DEC)` выводит "78"
- `Serial.print(78, HEX)` выводит "4E"
- `Serial.println(1.23456, 0)` выводит "1"
- `Serial.println(1.23456, 2)` выводит "1.23"
- `Serial.println(1.23456, 4)` выводит "1.2346"
- `Serial.write(210)` выводит байт данных, который принимается LabView строкой D2

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

Задание 1. Подключение I2C датчика к LabView на повышенной частоте передачи данных по запросу.

1. Для контроллера Arduino UNO разработайте программу считывания показаний I2C датчика AD7416 и передачу их в COM порт по запросу. Пример программы приведен ниже.

```
#include <Wire.h>

int I2C_first ; // старший (первый) байт I2C
int I2C_second; // младший (второй) байт I2C
int LV_mode;
// Pin 13 has the LED
int led = 13;

void setup()
{
  Wire.begin();    // подключение к шине I2C

  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

  Serial.begin(115200); // запуск последовательного порта на частоте 9600, 115200
}

void loop()
{
  while (!Serial.available()); // ожидание запроса I2C данных
  LV_mode = Serial.read(); // чтение байта запроса и синхронизации LabView
  switch (LV_mode) {
    case 49: { // 49 DEC, ASCII code = 1
      Serial.write(I2C_first); // передает в порт байт целого значения температуры
      Serial.write(I2C_second); // передает в порт байт дробного значения температуры

      // Чтение данных I2C канала для следующей пересылки по запросу
      Wire.requestFrom(73, 2); // запрос 2 байт от слейва #0x49 or 73
      // Функция запрашивает данные у ведомого устройства (slave);
      while(Wire.available()) // пока есть, что читать
      {
        I2C_first = Wire.read(); // считываем первый байт
        I2C_second = Wire.read(); // считываем второй байт
      }
      break;
    }
  }
}
```

2. Проверьте работу программы контроллера с использованием утилиты COM Port Toolkit.

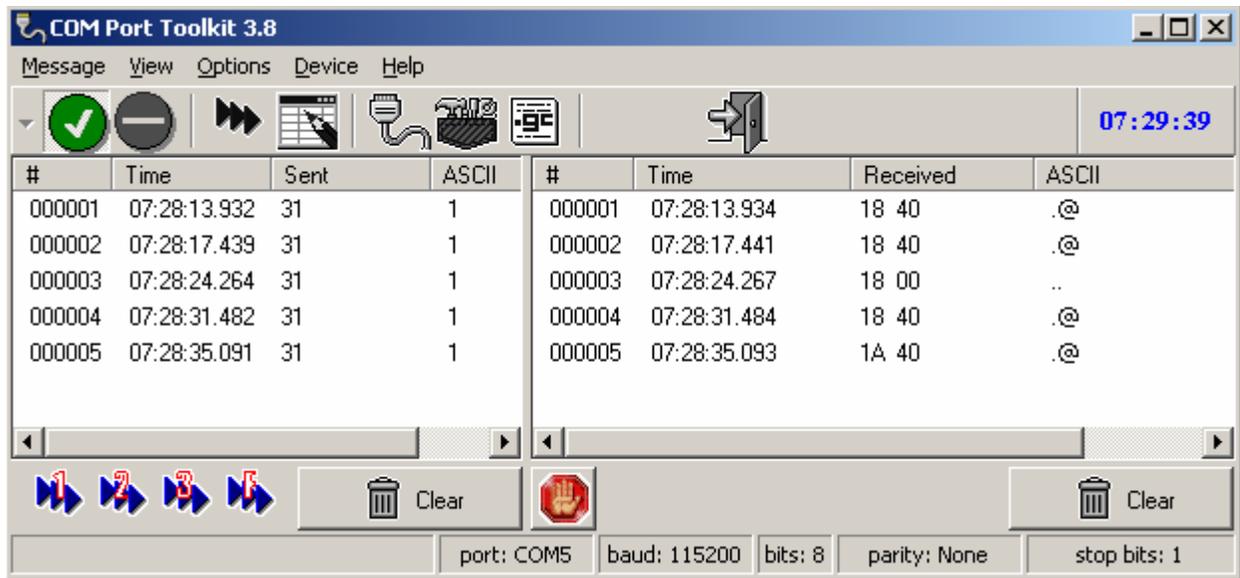


Рис. 6. Передача в COM порт байта 31_{16} (ASCII код равен 1) запускает контроллер на передачу в COM порт двух байт датчика, считанных на предыдущем запросе.

3. Разработайте структуру виртуального прибора в LabView, который посылает запрос контроллеру Arduino UNO, получает и отображает показания датчика температуры на максимальной частоте.

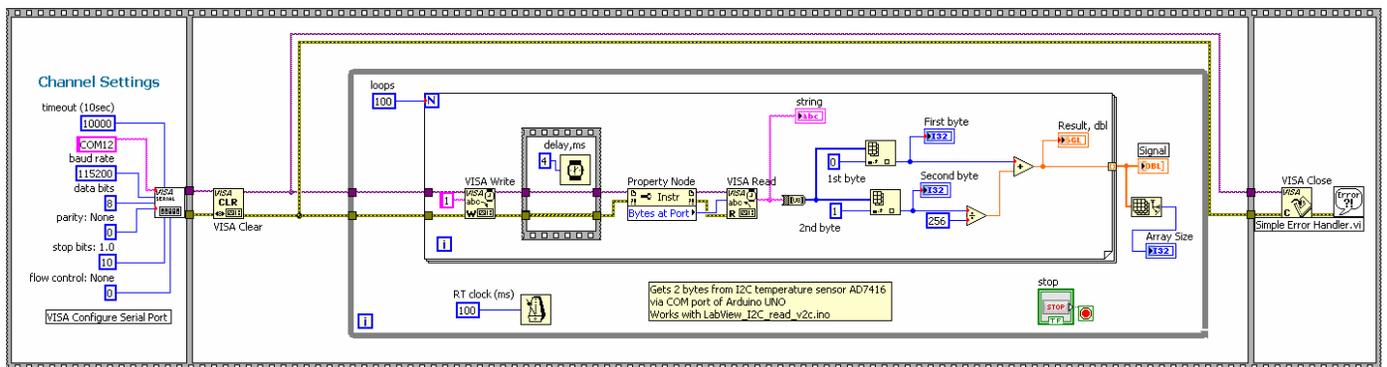


Рис. 7. Структура виртуального прибора LabView. Система за 10 секунд принимает 2000 показаний датчика температуры, формирует и отображает вектора из принимаемых данных размером 100 единиц.

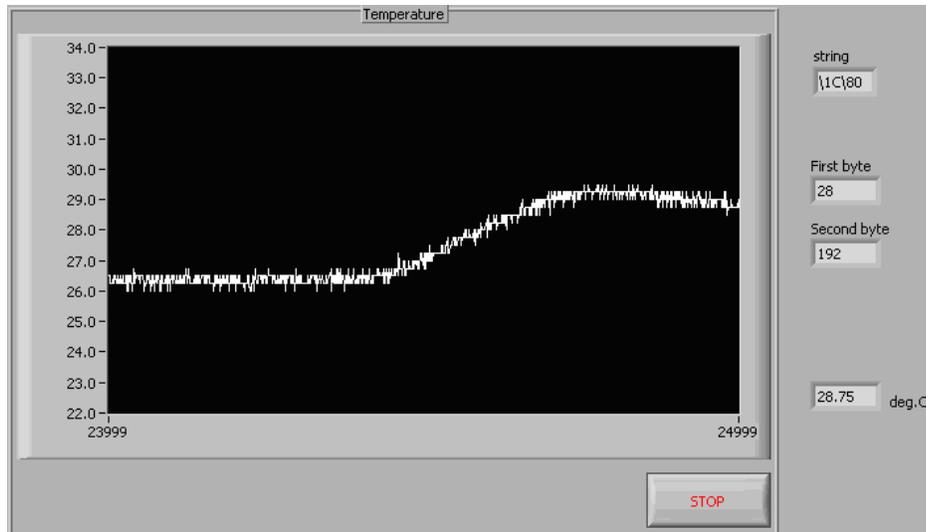


Рис. 8. Интерфейс виртуального прибора LabView для отображения значений температуры. Для контроля на экран выводятся значения принимаемых байт в формате строки и числовом формате.

4. Проверьте работоспособность структуры I2C датчик - интерфейс виртуального прибора LabView.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Чем ограничена максимальная частота приема и отображения показаний I2C датчика в LabView?
2. Как влияет изменение размерности вектора принимаемых данных температурного датчика на скорость отображения данных?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Dr. Bob Davidov. Среда разработки интерфейсов и программных модулей систем управления LabVIEW http://portalnp.ru/wp-content/uploads/2013/10/06.04_LabView-as-tool-for-control-system-design_Ed5.pdf
2. Dr. Bob Davidov. Управление COM портом в LabView http://portalnp.ru/wp-content/uploads/2013/09/08.05_COM-port-control-in-LabView_Ed_31.pdf
3. Dr. Bob Davidov. Обмен данными через COM port http://portalnp.ru/wp-content/uploads/2013/09/09.09_Communication-via-COM-port_Ed_51.pdf
4. Dr. Bob Davidov. Компьютерные технологии управления в технических системах <http://portalnp.ru/author/bobdavidov>.