

Разработка учебного стенда для демонстрации голосового управления с помощью Android устройства

Мясищев А.А.

На рисунке 1 представлено фото стенда. Он построен на основе отладочного комплекса AVR-EASY-KIT[1]. Для связи стенда с Android устройством (смартфоном, планшетом) используется Bluetooth HC-05.

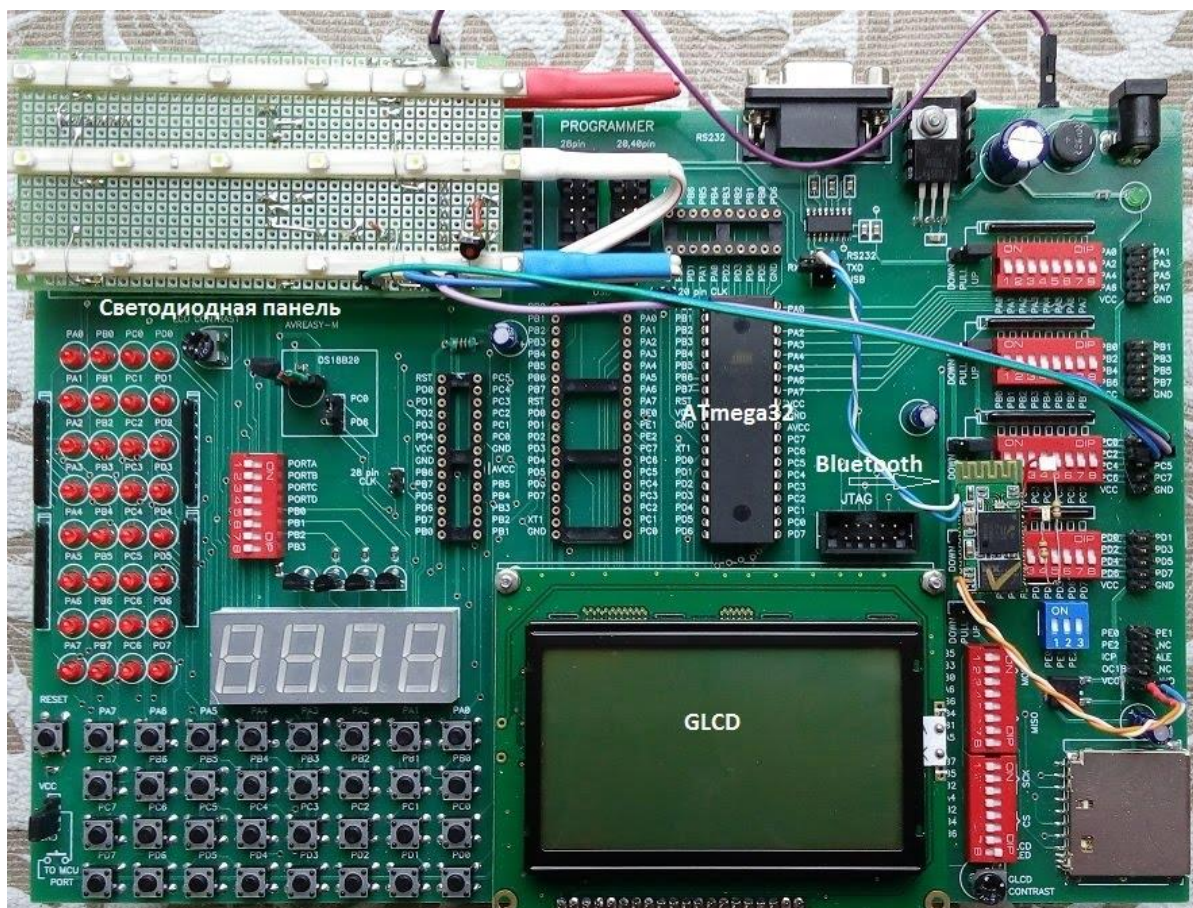


Рис.1. Фотография стенда голосового управления устройствами

Система голосового управления работает следующим образом:

1. После включения стенда на смартфоне запускается приложение и выполняется подключение к Bluetooth устройству стенда;
 2. В приложении смартфона нажимается кнопка "Нажми и говори" и произносится команда, например, "Включить синий". На светодиодной панели загорается светодиодная полоска синим цветом.
 3. Для окончания голосового диалога произносится команда "Конец связи". Смартфон должен ответить "до свидание".
- Аналогично включаются и выключаются белые и красные светодиоды. При произнесении команды "температура", смартфон должен произнести значение температуры, которое высвечивается на индикаторе GLCD. Причем, распознавание голосовых команд выполняется как при работе с Интернет, так и без Интернет при использовании внутренней библиотеки, если Смартфон позволяет ее скачать и работать в режиме offline. Без

Интернет распознавание хуже, поэтому иногда приходится повторять команды.

Программа для стенда написана так, что кроме управления температурным датчиком и светодиодными полосками на GLCD выводится изменяющаяся рекламная информация с показанием температуры, а также мигают по заданному алгоритму восемь светодиодов, подключенные к порту PB0,...,PB7. Управление GLCD и светодиодами порта PB выполняется функцией обработки прерывания по совпадению счетчика таймера TIME1.

Программа для микроконтроллера стенда написана на языке Wiring (Модифицированный C++ для контроллеров Ардуино). Ниже приведен текст программы.

```
#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>
#include <mega32.h>
#include <OneWire.h>
#include <glcd.h>
#include <fonts/fixed.h>
#include <fonts/fixed1.h>
gText textTop = gText(textAreaTOP);
OneWire ds(14);
float te;
int s=0;
int t=0;
int ten,ten1,ten2;
char inByte; // входящие данные
void setup() {
te=temp();
GLCD.Init();
Serial.begin(9600); // инициализация порта для работы с bluetooth
DDRB |=
(1<<PB0)|(1<<PB1)|(1<<PB2)|(1<<PB3)|(1<<PB4)|(1<<PB5)|(1<<PB6)|(1<<PB7
); // Порты микроконтроллера работают на выход
DDRC |= (1<<PC1)|(1<<PC2)|(1<<PC3); // Порты работают на выход
TCCR1A=0x00; // Задаст режим работы Time 1, т.е.
TCCR1B=0x05; //Тактирование таймера CLK/1024
TCNT1H=0x00; // В этих регистрах происходит
TCNT1L=0x00; // счет
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x07; // Регистры сравнения A. Старший байт 0x07
OCR1AL=0x08; // Регистры сравнения. Младший байт 0x08
OCR1BH=0x00; // Регистры сравнения B.
OCR1BL=0x00;
TIMSK=0x10; // Разрешает прерывание по сравнению регистров для Time 1
```

```

}
ISR(TIMER1_COMPA_vect) // Функция обработки прерывания для вывода
информации на экран и мигания диодами
{
  TCNT1H=0; // Каждый раз обнуляем регистры счета
  TCNT1L=0;
  if(t==0&& s==0){
  GLCD.Init();
  textTop.SelectFont(new_Font); // выбор шрифтов
  textTop.DefineArea( 5, 13, 127, 63, SCROLL_UP); // Выбор зоны распечатки
  textTop.print("T="); textTop.print(te,1);textTop.println("C"); // Распечатка
  T=21.6C
  }
  if(t==1&& s==0){
  GLCD.Init();
  textTop.SelectFont(new_Font1);
  textTop.DefineArea( 6, 10, 127, 63, SCROLL_UP);
  textTop.println("Department@of");
  textTop.println("Cybersecurity");
  te=temp(); // Определение температуры
  }
  if(t==2&& s==0){
  GLCD.Init();
  textTop.SelectFont(new_Font);
  textTop.DefineArea( 5, 13, 127, 63, SCROLL_UP);
  textTop.print("T="); textTop.print(te,1);textTop.println("C");
  }
  if(t==3&& s==0){
  GLCD.Init();
  textTop.SelectFont(new_Font1);
  textTop.DefineArea( 8, 0, 127, 63, SCROLL_UP);
  textTop.println("@Khmelnitsky");
  textTop.println("@ @ @national");
  textTop.println("@ @University");
  }
  if (s==1){ // Последовательное мигание 8-ю светодиодами
  PORTB |= (1<<PB0);
  _delay_ms(200);
  PORTB &= ~(1<<PB0);}
  if (s==2){
  PORTB |= (1<<PB1);
  _delay_ms(200);
  PORTB &= ~(1<<PB1); }
  if (s==3){
  PORTB |= (1<<PB2);

```

```

_delay_ms(200);
PORTB &= ~(1<<PB2); }
if (s==4){
PORTB |= (1<<PB3);
_delay_ms(200);
PORTB &= ~(1<<PB3); }
if (s==5){
PORTB |= (1<<PB4);
_delay_ms(200);
PORTB &= ~(1<<PB4); }
if (s==6){
PORTB |= (1<<PB5);
_delay_ms(200);
PORTB &= ~(1<<PB5);}
if (s==7){
PORTB |= (1<<PB6);
_delay_ms(200);
PORTB &= ~(1<<PB6); }
if (s==8){
PORTB |= (1<<PB7);
_delay_ms(200);
PORTB &= ~(1<<PB7); }
if (s==9){ // Одновременное мигание 8-ю светодиодами три раза
PORTB=0xFF;
_delay_ms(100);
PORTB=0x00;
}
if (s==10){
PORTB=0xFF;
_delay_ms(100);
PORTB=0x00;
}
if (s==11){
PORTB=0xFF;
_delay_ms(100);
PORTB=0x00;
}
s++;
if (s==12) { s=0; t++;}
if(t==4&& s==0) {t=0;}
}
float temp() // Функция определения температуры
{
byte i;
byte data[10];

```

```

byte addr[8];
float celsius;
ds.search(addr);
ds.reset();
ds.select(addr);
ds.write(0x44,1);
delay(1000);
ds.reset();
ds.select(addr);
ds.write(0xBE);
for ( i = 0; i < 9; i++) {
  data[i] = ds.read();}
int raw = (data[1] << 8) | data[0];
  unsigned char t_mask[4] = {0x7, 0x3, 0x1, 0x0};
  byte cfg = (data[4] & 0x60) >> 5;
  raw &= ~t_mask[cfg];
celsius = (float)raw / 16;
return celsius;
}
void loop() { // Анализ пришедших с Bluetooth данных
if (Serial.available() > 0) { //если пришли данные
  inByte = Serial.read(); // считываем байт
  if(inByte == '0') {
    PORTC &= ~(1<<PC1); // если 0, то выключаем RED
  }
  if(inByte == '1') {
    PORTC |= (1<<PC1); // если 1, то включаем RED
  }
  if(inByte == '2') {
    PORTC &= ~(1<<PC2); // если 2, то выключаем WHITE
  }
  if(inByte == '3') {
    PORTC |= (1<<PC2); // если 3, то включаем WHITE
  }
  if(inByte == '4') {
    PORTC &= ~(1<<PC3); // если 4, то выключаем BLUE
  }
  if(inByte == '5') {
    PORTC |= (1<<PC3); // если 5, то включаем BLUE
  }
  if(inByte == '6') { // если 6, то считываем температуру и посылаем ее на
bluetooth
    Serial.print("Temperatura, "); ten=10*te; ten1=ten/10; ten2=ten%10;
    Serial.print(ten1); Serial.print("i"); Serial.print(ten2); Serial.print(" ");
  } } }

```

Для компиляции этой программы в программной среде Ардуино(здесь используется arduino-1.0.3) необходимо заменить файл boards.txt в каталоге c:\arduino-1.0.3-windows\arduino-1.0.3\hardware\arduino на файл, находящийся здесь. Также требуется для работы графического дисплея установить в каталоге c:\arduino-1.0.3-windows\arduino-1.0.3\libraries библиотеку glcd и изменить в ней файл c:\arduino-1.0.3-windows\arduino-1.0.3\libraries\glcd\config\ks0108_Arduino.h на файл с таким же именем, но приведенный здесь. В каталоге c:\arduino-1.0.3-windows\arduino-1.0.3\hardware\arduino\variants\ необходимо создать подкаталог m16 и разместить там файл pins_arduino.h. Этот файл описывает пины микроконтроллера ATmega16 (ATmega32). После таких изменений в программной среде Ардуино можно будет выбрать платы на базе микроконтроллеров ATmega16, ATmega32.

На рисунке 2 представлена схема подключений светодиодных полосок и Bluetooth HC-05 к отладочному комплексу AVR-EASY-KIT.

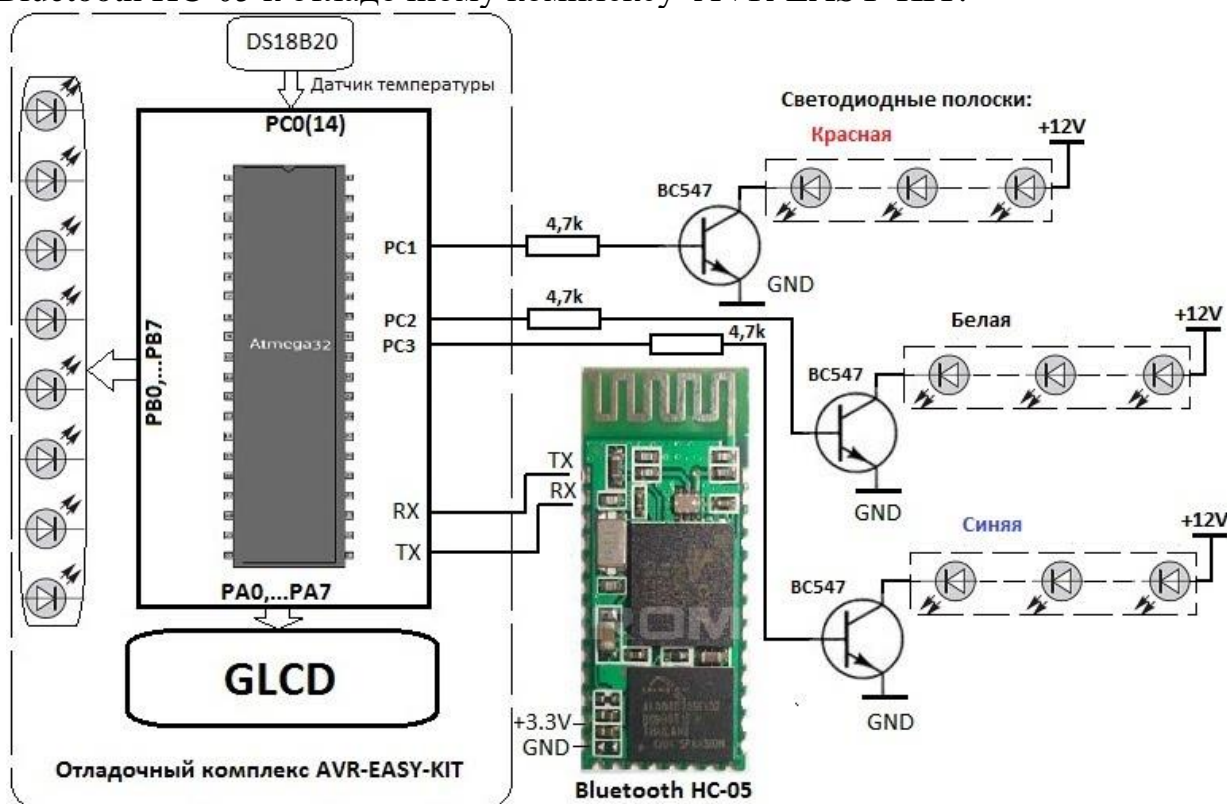


Рис.2. Схема подключений светодиодных полосок и Bluetooth HC-05 к отладочному комплексу AVR-EASY-KIT

Схему соединений электронных компонентов в отладочном комплексе можно посмотреть в источнике [1].

Линейка красных светодиодов подключена через транзистор к порту PC1, белых - к порту PC2, синих - к порту PC3.

Программа для Андроид устройства (смартфона) написана с использованием языка визуального программирования App Inventor 2. Ниже представлены графические иллюстрации программы.

```
when TextToSpeech1 AfterSpeaking
do
  if
    SpeechRecognizer1 Result = "конец связи" and SpeechRecognizer1 Result = "Конец связи" and SpeechRecognizer1 Result = " "
  then
    call SpeechRecognizer1 GetText

when Button1 Click
do
  call SpeechRecognizer1 GetText

when ListPicker1 AfterPicking
do
  if
    call BluetoothClient1 Connect address ListPicker1 Selection
  then
    set ListPicker1 TextColor to #0000FF
    set ListPicker1 Text to "Подключен"
  else
    set ListPicker1 TextColor to #FF0000
    set ListPicker1 Text to "Переподключение"

when ListPicker1 BeforePicking
do
  set ListPicker1 Elements to BluetoothClient1 AddressesAndNames

when Button2 Click
do
  call BluetoothClient1 SendText text "1"
  set Label2 Text to "Красный включен"
  call TextToSpeech1 Speak message Label2 Text

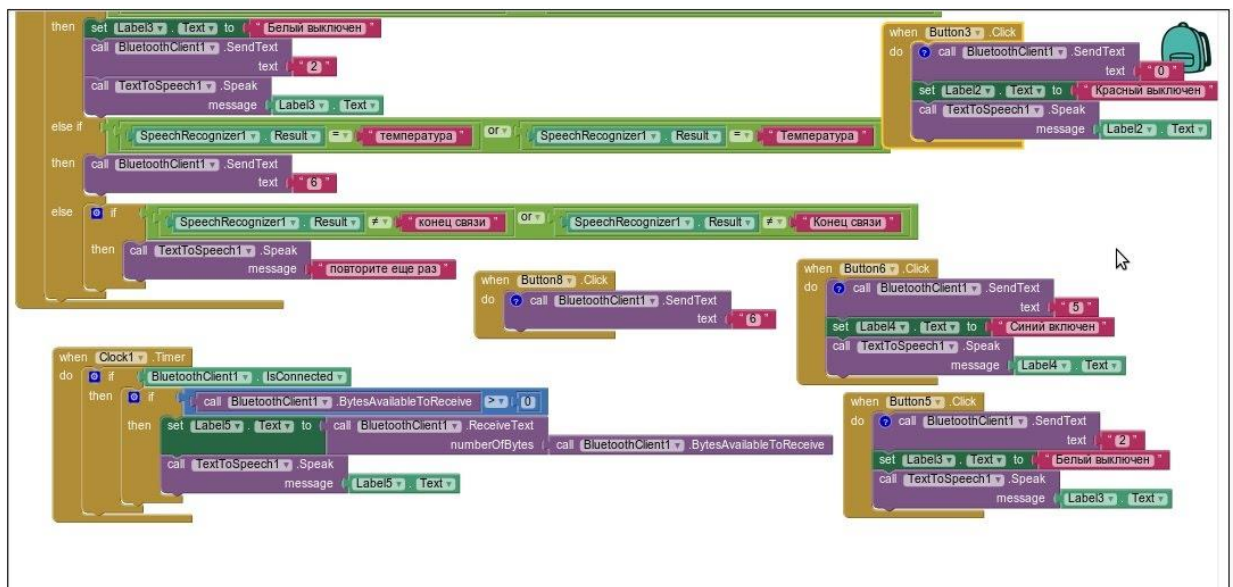
when SpeechRecognizer1 AfterGettingText
do
  set Label1 Text to SpeechRecognizer1 Result
  if
    SpeechRecognizer1 Result = "включить синий" or SpeechRecognizer1 Result = "Включить синий"
  then
    set Label4 Text to "Синий включен"
    call BluetoothClient1 SendText text "0"
    call TextToSpeech1 Speak message Label4 Text
  else if
    SpeechRecognizer1 Result = "спасибо" or SpeechRecognizer1 Result = "Спасибо"
  then
    call TextToSpeech1 Speak message "пожалуйста"
  else if
    SpeechRecognizer1 Result = "откуда ты" or SpeechRecognizer1 Result = "Откуда ты"
  then
    call TextToSpeech1 Speak message "я с кафедры кибербезопасности, компьютерных систем и сетей Хмельницкого национального университета"
  else if
    SpeechRecognizer1 Result = "привет" or SpeechRecognizer1 Result = "Привет"
  then
    call TextToSpeech1 Speak message "Здравствуйте"
  else if
    SpeechRecognizer1 Result = "конец связи" or SpeechRecognizer1 Result = "Конец связи"
  then
    call TextToSpeech1 Speak message "до свидания"
  else if
    SpeechRecognizer1 Result = "что ты умеешь" or SpeechRecognizer1 Result = "Что ты умеешь"
  then
    call TextToSpeech1 Speak message "Включать и выключать светодиоды. Определять температуру"
  else if
    SpeechRecognizer1 Result = "твои команды" or SpeechRecognizer1 Result = "Твои команды"

when Button4 Click
do
  call BluetoothClient1 SendText text "3"
  set Label3 Text to "Белый включен"
  call TextToSpeech1 Speak message Label3 Text

when Button7 Click
do
  call BluetoothClient1 SendText text "4"
  set Label4 Text to "Синий выключен"
  call TextToSpeech1 Speak message Label4 Text
```

```
else if
  SpeechRecognizer1 Result = "твои команды" or SpeechRecognizer1 Result = "Твои команды"
then
  call TextToSpeech1 Speak message "включить, выключить красный белый синий. Температура"
else if
  SpeechRecognizer1 Result = "как тебя зовут" or SpeechRecognizer1 Result = "Как тебя зовут"
then
  call TextToSpeech1 Speak message "меня зовут маша"
else if
  SpeechRecognizer1 Result = "выключить синий" or SpeechRecognizer1 Result = "Выключить синий"
then
  set Label4 Text to "Синий выключен"
  call BluetoothClient1 SendText text "4"
  call TextToSpeech1 Speak message Label4 Text
else if
  SpeechRecognizer1 Result = "включить красный" or SpeechRecognizer1 Result = "Включить красный"
then
  set Label2 Text to "Красный включен"
  call BluetoothClient1 SendText text "1"
  call TextToSpeech1 Speak message Label2 Text
else if
  SpeechRecognizer1 Result = "выключить красный" or SpeechRecognizer1 Result = "Выключить красный"
then
  set Label2 Text to "Красный выключен"
  call BluetoothClient1 SendText text "0"
  call TextToSpeech1 Speak message Label2 Text
else if
  SpeechRecognizer1 Result = "включить белый" or SpeechRecognizer1 Result = "Включить белый"
then
  set Label3 Text to "Белый включен"
  call BluetoothClient1 SendText text "3"
  call TextToSpeech1 Speak message Label3 Text
else if
  SpeechRecognizer1 Result = "выключить белый" or SpeechRecognizer1 Result = "Выключить белый"
then
  set Label3 Text to "Белый выключен"
  call BluetoothClient1 SendText text "2"
  call TextToSpeech1 Speak message Label3 Text
else if
  SpeechRecognizer1 Result = "температура" or SpeechRecognizer1 Result = "Температура"
then
  call BluetoothClient1 SendText

when Button3 Click
do
  call BluetoothClient1 SendText text "0"
  set Label2 Text to "Красный выключен"
  call TextToSpeech1 Speak message Label2 Text
```



Литература

1. Отладочный комплекс AVR-EASY-KIT. [Электронный ресурс]. – Режим доступа: <http://www.kosmodrom.com.ua/razrobotka/avreasy5.php>, 2014.
2. Мясищев, А.А. Управление через интернет электропитанием и датчиками мини компьютером Raspberry Pi // Вісник Хмельницького національного університету. Технічні науки. – 2016. – № 3. – С. 7-15.
3. Мясіщев О. А. Метод розпізнавання мовлення по короткому словнику з використанням Mel-кепстральних коефіцієнтів / О. А. Мясіщев, Є. С. Ленков, С. Г. Ожаровський //Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – Київ : ВІКНУ, 2015. – Вип. 50. – С. 225 -231.
4. Мясищев, А.А. Голосовое и кнопочное управление через Интернет сервером на модуле ESP8266-01 с помощью Смартфона. [Электронный ресурс]. – Режим доступа: https://sites.google.com/site/webstm32/manag_esp8266, 2015.
5. Мясіщев О.А. Мережні інформаційні технології : навчальний посібник / Мясіщев О.А., Джулій В.М., Красильников С.Р., Чешун В.М. – Хмельницький : ХНУ, 2012. – 422 с.
6. Ожаровський С.Г. , Мясіщев О.А. МЕТОД РОЗПІЗНАВАННЯ МОВЛЕННЯ, ПО КОРОТКОМУ СЛОВНИКУ, З ВИКОРИСТАННЯМ MEL-КЕПСТРАЛЬНИХ КОЕФІЦІЄНТІВ / ВІСНИК ХНУ. Технічні науки.- Хмельницький: ХНУ, 2015. - №6.- с. 181-184.
7. Мясищев А.А. Распознавание команд с помощью ДПФ и библиотеки FANN для управления устройствами. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2015. – Вип. № 50. – с. 216-225
8. Мясищев А.А., Ленков С.В., Жиров Г.Б. Изучение возможности построения web - сервера на платформе Arduino для удаленного управления. Сучасна Спеціальна Техніка, №4(35), 2013 - с.41-50.

9. Мясичев А.А. Создание надежного HTTP-сервера для удаленного управления по TCP/IP сети на основе ATMEGA1280 и WIZNET W5100. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2014. – Вип. № 46. – с 59-70.
10. Мясичев А.А., Полозова В.М. Простой UDP сервер на базе atmega328p и enc26j60. - 2014. Volume 16. Mathematics. Physics. Modern information technologies. Sheffield. Science and education LTD - с. 69 -79
11. Мясичев А.А. Способ размещения данных в flash - памяти ATmega1280 используемых для работы Web - сервера. - 2014. Том 29. Современни технологии на информации. София. - с. 23-32
12. Alexander Myasishev. Http-server on a microcontroller for remote management by network. British Journal of Science, Education and Culture, 2014, No.2. (6) (July-Decem- ber). Volume I. London University Press. London, 2014. с. 214 - 221