УДК 004.94

М. М. Коломиец

M. M. Kolomiets

Коломиец Максим Михайлович, студент 2 курса магистратуры, ЭЭФ ФГБОУ ВО Дальневосточный ГАУ, г. Благовещенск.

Научный руководитель: Пустовая Олеся Александровна, канд. с-х. наук, доцент кафедры ЭиАТП, зам. декана ЭЭФ по НР, ФГБОУ ВО Дальневосточный ГАУ, г. Благовещенск.

Kolomiets Maxim Mikhailovich, 2-year student of the magistracy, EEF FGBOU VO Far-Eastern State University, Blagoveshchensk.

Scientific adviser: Pustovaya Olesya Aleksandrovna, Candidate of Science, Associate Professor of the Department of EiATP, Deputy Dean of the EEF for HP, FGBOU VO Far Eastern State University, Blagoveshchensk.

АЛГОРИТМ УПРАВЛЕНИЯ СИСТЕМОЙ РАЗГОННОЙ ВЕНТИЛЯЦИИ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРОВ АРДУИНО

ALGORITHM OF MANAGEMENT OF THE SYSTEM OF OVERCLOCKING VENTILATION ON THE BASIS OF MICROCONTROLLERS OF ARDOINO

Аннотация. *В статье представлена программа и алгоритм управления системой разгонной вентиляции на основе микроконтроллеров Ардуино.*

Annotation. *The article presents the program and algorithm for controlling the system of accelerating ventilation based on microcontrollers Arduino.*

Ключевые слова: *алгоритм управления, микроконтроллер, ардуино, программа.*

Keywords: *control algorithm, microcontroller, arduino, program.*

Развивающееся промышленное производство немыслимо без использования современных средств управления, к которым, прежде всего, относятся как роботизированные, так и автоматизированные системы управления. Их использование позволяет оптимизировать режимы производства, усовершенствовать процесс управления и в конечном итоге снизить затраты как энергоресурсов так и затраты труда.

Наиболее активно в этом направлении развиваются системы автоматизации на основе микроконтроллерного управления, наиболее ярким представителем, которых является компания ОВЕН выпускающая линейку электронных компонентов и преобразователей для комплексной автоматизации производства, которая построена на основе блочных технологий. Компоновка системы производится при помощи программного обеспечения размещенного на сайте компании в свободном доступе. Однако недостатком данных продуктов является высокая стоимость компонентов системы.

Более дешевый вариант управления можно обеспечить при помощи микропроцессоров ARDUINO, к которым также можно подобрать полную линейку комплектующих.

Нами использован данный микроконтроллер для построения системы управления разгонной вентиляцией в коровнике на 400 голов. Блок схема системы управления представлена на рисунке 1.

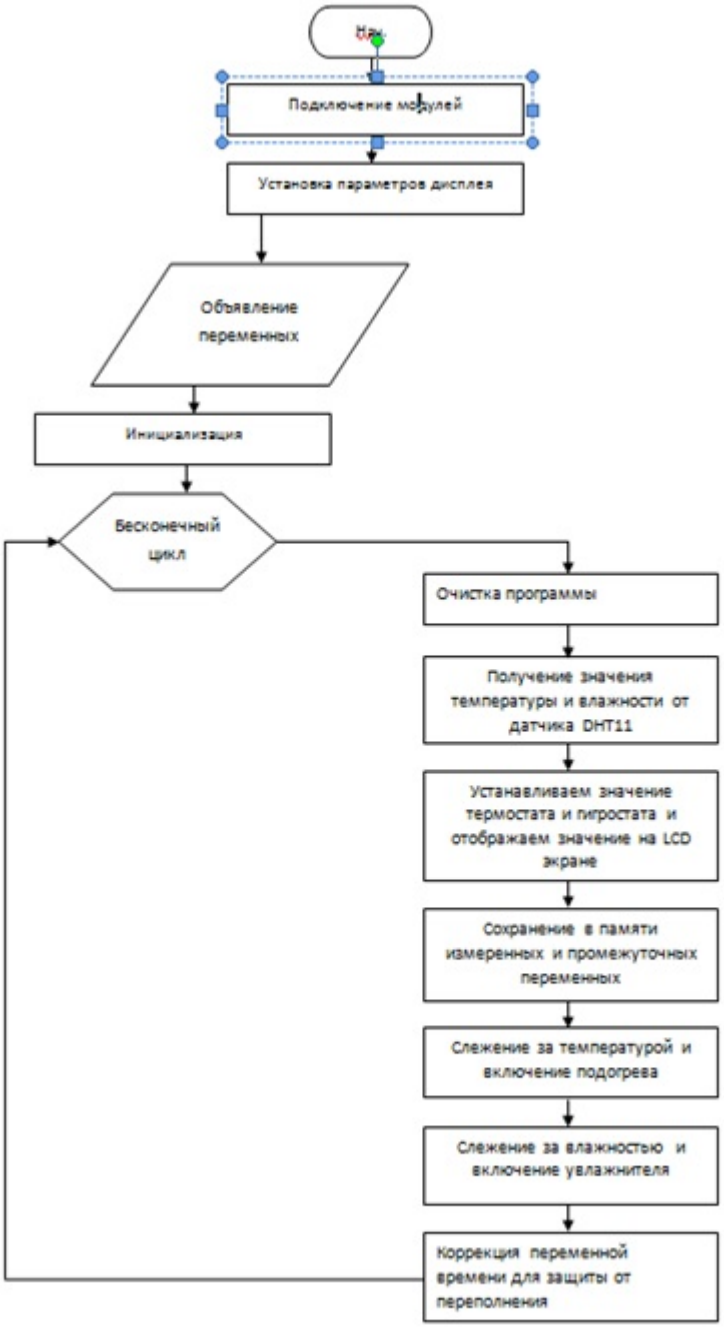

Рисунок 1. Блок схема управления установкой вентиляции

Структура программы состоит из ниже перечисленных разделов.

Подключение сторонних модулей для работы с жидкокристаллическим экраном и датчиком влажности и температуры по шине I2C, для работы с электрически стираемой постоянной памятью, а также обновленный модуль для работы с датчиком DHT [1, 2].

#include <wire.h></wire.h>

#include <liquidcrystal_i2c.h></liquidcrystal_i2c.h>

#include <eeprom.h></eeprom.h>

#include "DHT_NEW.h"

Введение параметров дисплея:

```
LiquidCrystal_I2C _lcd1(0x3F, 16, 2);
int _dispTempLength1=0;
boolean _isNeedClearDisp1;
```

Объявление глобальных переменных используемых в программе:

```
int _PWDC = 0;
float _gtv1 = 50;
……..
bool _gtv8 = 0;
String _gtv10 = "SAVE SAVE SAVE!!";
bool _SEEPR2OSN = 0;
int _disp5oldLength = 0;
int _disp4oldLength = 0;
```

Установка начальных параметров – инициализация значений:

```
void setup()
{Wire.begin();
delay(10);
TCCR2A = 0x00;
TCCR2B = 0x07;
TIMSK2=0x01;
TCNT2=100;
pinMode(1, INPUT);
digitalWrite(1, HIGH);
pinMode(3, INPUT);
digitalWrite(3, HIGH);
```

Тело программы:

```
void loop()
```

Очистка программы:

```
{if (_isNeedClearDisp1) {_lcd1.clear(); _isNeedClearDisp1= 0;}

_PWDC = 0;
```

Получение значения температуры и влажности от датчика DHT11:

```
if(_isTimer(_dht1Tti, 1000)) {

if(_isTimer(_dht1LRT,( _dht1.getMinimumSamplingPeriod()))) {

_dht1.readSensor();

_dht1LRT = millis();

_dht1Tti = millis();
```

Устанавливаем значение термостата и гигростата и отображаем значение на LCD экране:

```
if (!( (digitalRead (3)))) { if (! _gen1I) { _gen1I = 1; _gen1O = 1; _gen1P = millis(); } } else { _gen1I = 0 ; _gen1O= 0;}

 if (_gen1I) { if ( _isTimer ( _gen1P , 10 )) { _gen1P = millis(); _gen1O = ! _gen1O;}}

if (_gen1O)

…

if (_gtv3) _count2P = 0;

if(( (!( (digitalRead (3)))) || (!( (digitalRead (1)))) )) {_tim4O = 1; _tim4I = 1;} else { if(_tim4I) {_tim4I = 0; _tim4P = millis();} else { if (_tim4O) {if (

_isTimer(_tim4P, 150000)) _tim4O = 0;}}}

…

if (1) {

_dispTempLength1 = (((String((int(_gtv6)), DEC)))).length();

if (_disp6oldLength > _dispTempLength1) {_isNeedClearDisp1 = 1;}

_disp6oldLength = _dispTempLength1;

_lcd1.setCursor(12, 1);

_lcd1.print(((String((int(_gtv6)), DEC))));
```

Сохранение в памяти измеренных и промежуточных переменных:

```
…

if (_tim1O) { if (! _gen3I) { _gen3I = 1; _gen3O = 1; _gen3P = millis(); } } else { _gen3I = 0 ; _gen3O= 0;}
```

```
if (_gen3I) { if ( _isTimer ( _gen3P , 500 )) { _gen3P = millis(); _gen3O = !_gen3O;}}

if(!( (digitalRead (2)))){ if(!_SEEPR1OSN){(updateFloatToEEPROM(0, 0, 0x0, (_gtv5)));

_SEEPR1OSN=1;} }else{ if(_SEEPR1OSN){_SEEPR1OSN=0;}}

if(!( (digitalRead (2)))){ if(!_SEEPR2OSN){(updateFloatToEEPROM(4, 0, 0x0, (_gtv6)));

_SEEPR2OSN=1;} }else{ if(_SEEPR2OSN){_SEEPR2OSN=0;}}
```

Слежение за температурой и включение подогрева:

```
digitalWrite(7, !((_gtv1) < ((readFloatFromEEPROM(0, 0, 0x0)))));
```

Слежение за влажностью и включение увлажнителя:

```
digitalWrite(6, (_gtv2) < ((readFloatFromEEPROM(4, 0, 0x0))));
```

Коррекция переменной времени для защиты от переполнения:

```
bool _isTimer(unsigned long startTime, unsigned long period )

{ unsigned long currentTime;

currentTime = millis();

if (currentTime>= startTime) {return (currentTime>=(startTime + period));} else {return (currentTime >=(4294967295-startTime+period));}
```

Использование предлагаемого алгоритма позволяет контролировать систему по двум параметрам: температура (задается верхняя и нижняя граница) и влажность так же задаётся верхняя и нижняя граница.

Для проверки работоспособности алгоритма нами был изготовлен макет коровника с установленной разгонной вентиляцией, который при испытаниях подтвердил правильность разработанного алгоритма и возможность использования в реальных условиях.

Список литературы

1. Программирование Ардуино [Электронный ресурс] // All-Arduino.ru. – Режим доступа : https://all-arduino.ru/programmirovanie-arduino/ (Дата обращения: 13.04.2018)
2. Программирование Ардуино [Электронный ресурс] Arduino.ru – Режим доступа: Arduino.ru (Дата обращения: 13.04.18)