Здравствуйте,уважаемые посетители настоящего сайта.
Меня зовут Владимир Александрович.
Skype:ratekold1,email:vladimir.trukhmanov@yandex.ru.Живу я в городе Рассказово Тамбовской области. Занимался ремонтом бытовой техники,в том числе и стиральных машин.
Ремонтирую системы управления СМ,в силу скромных возможностей самодеятельного мастера и моего возраста .По образованию я инженер-механик,по увлечению,как теперь называется-электронщик,последнее время любитель-программист микроконтроллеров AVR.Всему,что я знаю по публикуемому вопросу я обязан интернету и дальнейшему осмыслению этого в практической деятельности. В интернет и возвращаю с глубокой благодарностью и надеждой: может найдет ищущий полезное и в моем крохотном вкладе в безбрежное море информации.
Применяя термин "ремонт" я имею в виду не только-произвел диагностику и заменил неисправ ный узел,это своего рода, достаточно рутинная, "работа" линейных мастеров больших фирм,но и ремонт в мастерской,в частности,плат управления.

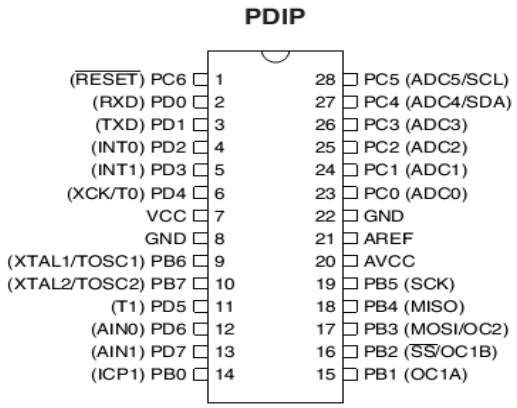
СМ в процессе интерактивного общения с пользователем выполняет следующие основные операции:получает команды и выполняет их,выдает информацию о ходе выполнения заданной программы ,информирует пользователя или ремонтника о возникающих неисправностях. Во время выполнения заданных видов стирки силовой блок управления всегда и во всех СМ блокирует дверцу люка,наливает или откачивает воду,вращает барабан,контролирует уровень воды в барабане,нагревает воду,производит диагностику ошибок в процессе инициализации и работы.Вот и всё.
Но,скажете Вы ,на рынке такое множество СМ и все такие разные.Разная “одежда” одета на стиральный барабан и разные подходы к выполнению этих одинаковых операций.
Теперь подойдем к основному вопросу нашего интереса силовым платам управления и вот они на несколько порядков более однообразны,чем механическое устройство СМ.
К примеру,на огромную линейку СМ Indezit две модели силовой платы-EVO2.и Low End.
И не удивительно-на всех машинах одинаковые операции(наливай,крути,грей….).
Но ремонтнику (без доступа к тайнам фирм) от этого не легче,ведь он может отремонтировать только исполнительные элементы(сменить тиристоры,реле….), но если не работает сам микроконтроллер все- замена платы.Здесь 4-7 тысяч вместе с работой.Это в крупных городах,а в регионах зачастую покупка новой СМ.
Тут много жизненных коллизий и ньюансов.
Я попробовал другой способ-замену микроконтроллера на свой.Свой значит на который у тебя есть информация о его устройстве,есть инструменты его программирования и загрузки в него разработанной и скомпилированной программы,т.е. в отличие от беспомощного взирания на красивую но неработающую плату с загадочными двумя процессорами и без всякой информации о них,загруженной в них программе, без инструментов диагностики и программирования.Ведь СМ завезены неизвестно откуда, неизвестно кем разработаны и неизвестно кем сделаны.
Этот вопрос капитального ремонта плат возник не случайно,именно после беспомощности в многочисленных ситуациях такого типа.
Выбран процессор Atmega8,как наиболее современный с доступным описанием устройства,наличием в интернете доступных инструментов разработки программы и её загрузки.
Я начал с программ с несколькими отдельными операциями 3 стирки,полоскание, слив …итд.Установил на LG,Ardo ….7 машинок и все разных фирм,работают и сечас, уже около двух лет.Необыкновенно надёжен этот Atmega8 с его флешь памятью,позволяющей многочисленные перезагрузки программы в процессе отладки.
Но теперь я пришёл к выводу,что этого не надо:переделка верхней платы или возня с пультом управления от телевизора.

Пользователи СМ обычно работают на одной приглянувшейся программе,почти всегда на третьей,вся эта красота шкал,ручек,разнообразие вида программ,оборотов отжима….
Я разработал универсальную программу примерно на один час, 60 градусов подогрев,отжим на 800 оборотов.

Описание универсальной программы стирки в приложении к плате управления EVO2 стиральной машины марки INDEZIT.

Система запрограммирована на языке программирования С в приложении для микроконтроллеров AVR с помощью компилятора CodeVisionAVR.Все это можно получить и изучить на сайте 123nm.com рекрасный сайт для самообразования в программировании и микроэлектронике.Рекомендую.
В данном изложении не поставлена задача научить программированию микроконтроллеров AVR на С,для этого есть руководства по компилятору и языку и,самое главное 123nm.com.Автор попытается изложить логический ход построения программы и основные элементы её функциональных узлов.
Программа на С состоит из трёх основных компонентов :
Главной программы, где включены заголовочные файлы,описаны глобальные переменные и изложен алгоритм хода программы с использованием файлов библиотек.
Заголовочных файлов (.inc) в которых объявлены библиотечные функции.
Файлы библиотек(.lib),на которые ссылаются заголовочные файлы.
Файл библитеки это где Описаны часто встречающиеся в главной программе действия.В ней стоит только название действия(функции) и главная программа вызывает функцию из библиотеки,точнее компилятор в процессе компоновки программы и её компиляции для создания объектных файлов(прошивок) ,которые загружаются в микроконтроллер для окончательной отладки программы в режиме реального времени.
Рассмотрим нашу конкретную программу стирки для платы СМ INDEZIT EVO2 на С.

```
//Вариант: установка atmega8 на плату evo2 и загрузка диагностической
//программы с использованием имеемых цепей диагностики с некоторой
//их доработкой.  Информация звуковыми сигналами.Диагностирует 9 неис-
//правностей.
//без управления
//modul:evo2
// реверс мотора, тэн,мотор, помпа- включается единицей,полный бак-единица
//блокировка люка, клапана - нулем
//30gr adc-389 Датчик при 30 градусах: 4 ком
//40gr adc-483
//60gr adc-600
//микроконтроллер на работает на заводских предустановках  частота 1000000гц.
#include <mega8.h>
#include <m8_128.h> //   http://avr123.nm.ru/m8_128.h
#include <delay.h>
#include <stdio.h>
#include <wm_evo2_1.h>  //мой файл include в котором объявлены написанные мной функции.
//распределение  портов микроконтроллера (присвоения им определенных для данной
  //  программы функций:
```



Листинг основной программы.

```
#define MotorOn     PORTB.1=1   //Включение двигателя барабана
#define MotorOff    PORTB.1=0    //Выключение двигателя барабана
#define ValvOn1     PORTD.4=0    //Включение 1 клапана
#define ValvOff1    PORTD.4=1     //Выключение 1 клапана
#define ValvOn2     PORTD.5=0     //Включение 2 клапана
#define ValvOff2    PORTD.5=1     //Выключение 2 клапана
#define TankFull    PINB.2==1       //Контроль сигнала полного бака
#define TankEmpty  PINB.2==0      //Контроль сигнала пустого бака
#define PumpOn     PORTC.4=1       //Включение насоса
#define PumpOff    PORTC.4=0       //Выключение насоса


#define AlongClock PORTC.3        //Вращение двигателя по часовой стрелке
#define BackClock  PORTC.2        //Вращение двигателя против часовой стрелки
#define TemperatureTakeWater  ADMUX=0x40; ADCSRA|=0x40; while(!(ADCSRA &
0x10));ADCSRA|=0x10; adc_data=ADCW;// Замер температуры воды

#define ColdWater  adc_data<woshtemp    //Определение значения-холодная вода
#define HotWater   adc_data>woshtemp     //Определение значения -вода нагрета
#define HeaterOn     PORTD.6=1              //Включение подогревателя
#define HeaterOff    PORTD.6=0               //Выключение подогревателя
#define DoorLockOn   PORTC.5=0             //Люк закрыть
#define DoorLockOff  PORTC.5=1             //Люк открыть
#define ReversSpeedUpPause  delay_ms(5000);
//Объявление глобальных  переменных
char timeofheat_add=90; //единица измерения 10 секунд(1 минута=6)
char pumptime;
char valvtime_add;
 int woshtemp;
 char Direction;
 char angl;        //угол отсечки регулятора оборотов
 int tmcycl;       // длительнсть текущего цикла
```

```
 int IICR1;
 int adc_data;
 char shift;
//Описание функции инициализации программы
void init(void){
//Установка портов(вход-выход,начальное значение)
PORTB=0x00;//00000000
DDRB=0x02; //00000010
PORTC=0x20;// 0010_0000
DDRC= 0x3E;// 0011_1110
PORTD=0x30; // 0011_0000
DDRD=0x70;  // 0111_0000
//Присвоение глобальным переменным первоначального значения
woshtemp=550;
timeofheat_add=15;//1измерения здесь=1  минута
 pumptime=6;
 valvtime_add =10;
  angl=155;       //угол отсечки регулятора оборотов
  tmcycl=0;       // длительнсть текущего цикла
  IICR1=0;
  adc_data=0;
  shift=0;
//Установки регистров управления функциями микроконтроллера
//Timer1
TCCR1A=0x00;
TCCR1B=0x02;            //0x42; //catch by down,T for count 8 mksec,T=524.280 mlsec;~0.5 sec
 TIMSK|=0x20;           //enable  interrupt CAP1 Timer1
TIMSK|=0x04;              //enable interrupt_OVF Timer1
MCUCR|=0x04;            //any logical change  int1

GICR|=0x80;             //enable  interrupt int1

//Timer2
TCCR2|=0x04;            // делитель на 64 Timer2, Novf==156,25, Tovf==10000mlsek
                    //180/Novf=180/156,25 = 1.152 grad/count

//Установки измерителя температуры воды
// Select ADC input 0
ADCSRA = 0x00; //disable adc
ADCSRA=0x86;// 10000010
//SFIOR&=0xEF;
ADMUX=0x40;//канал 0,AVcc с внешним конденсатором AREF
ADCSRA|=0x40; //Start a 1 AD conversion
while(!(ADCSRA & 0x10)); // wait for conversion done, ADIF flag set
ADCSRA|=0x10; //clr ADIF;
#asm("sei");
 }

 //прерывание переполнения TIMER1
interrupt[9]void timer1_ovrf(void){
IICR1=30000;
}
```

```
//прерывание от датчика оборотов
interrupt[6] void CAP1(void){
TCNT1=0x00;
IICR1=ICR1;
}
//прерывание от сети переменного тока (int1).
interrupt[3]void timer2_correkt(void){
TCNT2=0x00;
tmcycl++;
if(shift==0){if(tmcycl==3){angl-=1;tmcycl=0;};};
}

void main(void){
init();
DoorLockOn;delay_ms(7000);
if ((DoorLockOn )&& (PINB.7 ==0)){sound_alarm(4); };  // дверь не заблокирована
delay_ms(2000);
if((PumpOff)&&(PINB.5==0)){sound_alarm(5);};//   обрыв цепи насоса

TemperatureTakeWater;
if(( adc_data>1000)||( adc_data<200)){ sound_alarm(6); }; //  обрыв или замыкание датчика
темпер

 HeaterOn;delay_ms(3000);
 if(PINB.4==0){HeaterOff;sound_alarm(7);};//обрыв в цепи ТЭНа
    HeaterOff ;
if(PINB.3==0){sound_alarm(8);};     //обрыв в цепи коллектора двигателя
delay_ms(2000);

 drivertest_on();
 delay_ms(1000);
 if(shift==0){sound_alarm(9);};  //двигатель не вращается
 shift=0;



    //woshing go on.test OK!
    sound_inform(4); //Звуковой сигнал начала стирки
delay_ms(5000);
 // DoorLockOn ;    //Блокировка люка
//delay_ms(2000);
 tmcycl=0;
  wosh(1700,2000,40);  //Функция стирки с данными параметрами
 speedup(1500,200,10,80,1000);//Промежуточный отжим
  ReversSpeedUpPause     //Стоянка между реверсами
 speedup(1500,500,10,120,1000); //Тоже

       rinsis1(1000,1000,15);      //Первое полоскание
 speedup(1500,100,20,20,3000);
  ReversSpeedUpPause
 speedup(1500,200,10,80,1000);
```

```
 ReversSpeedUpPause
 speedup(1500,500,10,120,1000);
          rinsis1(1000,1000,10);  //Второе  полоскание
 speedup(1500,200,10,80,1000);
  ReversSpeedUpPause
 speedup(1500,500,10,120,1000);

     rinsis2(1000,1000,7);            //Третье полоскание и отжим
 speedup(1500,100,20,20,5000);
  ReversSpeedUpPause
 speedup(1500,100,20,20,2000);
  ReversSpeedUpPause
 speedup(1500,200,10,60,2000);
  ReversSpeedUpPause
 speedup(1500,200,10,60,1000);
  ReversSpeedUpPause
 speedup(1500,200,10,80,1000);
  ReversSpeedUpPause               //Это всё функции сепарации перед отжимом
 speedup(1500,200,10,80,1000);   //Программа не чувствует дисбаланса,потому подготовке
  ReversSpeedUpPause               //к окончательному отжиму уделено столько внимания.
 speedup(1500,500,10,120,1000);
 ReversSpeedUpPause
 SpeedEnd(1700,1000,20000);
  //разбрасывание
  ReversSpeedUpPause
  ReversSpeedUpPause
 speedup(1500,100,10,10,1000);
 ReversSpeedUpPause
 speedup(1500,100,10,10,1000);
 ReversSpeedUpPause
 speedup(1500,100,20,10,2000);
 ReversSpeedUpPause
 HeaterOff;
 delay_ms(1000);
 DoorLockOff;
 delay_ms(9000);
 pumpon();
delay_ms(60000);

 sound_info_end(5);//Периодический  сигнал конца стирки.
}
```

Описание заголовочного файла wm_evo2-1.h

```
C compile header file //Заголовочный файл с объявлением библиотечных функций
#pragma used+
// library function prototypes
 void sound_alarm(char number_sounds);//Сигнал тревоги(неисправности)
  // void pump_on(int pump_time  );           //Раскачка с контролем занятого времени
   void sound_inform(char number_sounds); //Сигнал информационный о месте стирки
    void sound_info_end(char number_sounds); //Периодический сигнал конца стирки
```

```
   void pumpon();
     void valvon_add1();       //Долив 1 клапаном
     void valvon_add1_2();    //Долив 2 клапаном
    void coldwaterheater_add(); //Дополнительный подогрев
 void drivertest_on(); //Проверка двигателя барабана на страгивание
 void test_alarm();
 void wosh(int woshspeed,int timewcycl,char numberwcycls);     //Стирка
 void rinsis1(int rinsspeed,int timerinscycl,char numberrinscycls);//Полоскание
 void rinsis2(int rinsspeed,int timerinscycl,char numberrinscycls); //3 полоскание
 void speedup(int startspinspeed,char timespinstep,int setnumberspinsteps,char sizespeedstep,int
timeendstepLspin);
 void SpeedEnd(int startspinspeed,char timespinstep,int timeendstepLspin);
//Оконачательн.отжим

#pragma used-


#pragma library  wm_evo2_1.lib
```

Описание библиотечного файла wm+evo2_1.lib

```
//for Indezit,Ariston plate EVO2
//universal !!
#include <mega8.h>
#include <m8_128.h> //   http://avr123.nm.ru/m8_128.h
#include <delay.h>
#include <stdio.h>
#include <wm_evo2_1.h>

#define MotorOn    PORTB.1=1
#define MotorOff   PORTB.1=0
#define ValvOn1    PORTD.4=0
#define ValvOff1   PORTD.4=1
#define ValvOn2    PORTD.5=0
#define ValvOff2   PORTD.5=1
#define TankFull   PINB.2==1
#define TankEmpty  PINB.2==0
#define PumpOn     PORTC.4=1
#define PumpOff    PORTC.4=0
#define AlongClock PORTC.3
#define BackClock  PORTC.2
#define TemperatureTakeWater  ADMUX=0x40; ADCSRA|=0x40; while(!(ADCSRA &
0x10));ADCSRA|=0x10; adc_data=ADCW;
#define ColdWater  adc_data<woshtemp
#define HotWater   adc_data>woshtemp
#define HeaterOn     PORTD.6=1
#define HeaterOff    PORTD.6=0
#define DoorLockOn   PORTC.5=0
#define DoorLockOff  PORTC.5=1
#define ReversSpeedUpPause  delay_ms(5000);
```

```
void drivertest_on(){
if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле

  //страгивание,разгон на обороты стирки
  TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
   shift=0;
   tmcycl=0;
 while(IICR1>1700){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

//=============================================
//выполнение одного цикла  за заданное время и
//  на заданных оборотах
 tmcycl=0;
while(tmcycl<100){
if(IICR1>1700){
  if(angl>10){angl-=1;};  //befor 2
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
//================================================
 if(IICR1<1700){
   if(angl<140){ angl+=2;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
  #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);

 };
 };
 }
 void sound_alarm(char number_sounds){
char i=0;
int  j=0;
 char q=0;
 ValvOff1;
```

```
 ValvOff2;
 PumpOff;
 HeaterOff ;
 MotorOff ;
ret:
    PORTC.1=1;
    delay_ms(1);
    PORTC.1=0;
    delay_ms(1);
    i++;

 if(i<250){goto ret;};
    i=0;
 delay_ms(200);
 q++;
   // number_sounds--;
 if(q<number_sounds){goto ret;};
 q=0;
delay_ms(1000);
 ret1:
  PORTC.1=1;
    delay_ms(1);
    PORTC.1=0;
    delay_ms(1);
    j++;

 if(j<1000){goto ret1;};
    j=0;
    delay_ms(3000);
goto ret;
    }
void valvon_add1(){
 char q=0;
ValvOn1;
ret:
 delay_ms(60000 ); // 1 минута
 q++;
 if(TankFull){ValvOff1; goto end;};
 if(q<valvtime_add){goto ret;};
 ValvOff1;
 sound_alarm(1);
 end:
}
 void valvon_add1_2(){
 char q=0;
ValvOn1; ValvOn2;
ret:
 delay_ms(60000 ); // 1 минута
 q++;
 if(TankFull){ValvOff1;ValvOff2;goto end;};
 if(q<valvtime_add){goto ret;};
 ValvOff1;ValvOff2;
```

```
 sound_alarm(1);
 end:
}
void pumpon( ){
 int q=0;
 PumpOn;
 ret:
  delay_ms(60000 ); // минута
  q++;
  if(TankEmpty){delay_ms(10000);PumpOff; goto end;};
  if(q<pumptime){goto ret;};
 PumpOff;
  sound_alarm(3);
  end:
  }

      void sound_inform(char number_sounds){
     int i=0;
     char q=0;
 //char t=10;
ret:
     PORTC.1=1;
     delay_ms(1);
     PORTC.1=0;
     delay_ms(1);
     i++;

  if(i<500){goto ret;};
     i=0;
  delay_ms(200);
  q++;
    // number_sounds--;
  if(q<number_sounds){goto ret;};
  }
   void sound_info_end(char number_sounds)  {
 char i=0;
 char q=0;
 //char t=10;
ret:
     PORTC.1=1;
     delay_ms(1);
     PORTC.1=0;
     delay_ms(1);
     i++;

  if(i<250){goto ret;};
   i=0;
  delay_ms(200);
  q++;
    // number_sounds--;
  if(q<number_sounds){goto ret;};
  q=0;
```

```
 delay_ms(10000);

goto ret;
   }
void coldwaterheater_add(){   //char TimeOfHeat
char q=0;
PORTD.6=1;
scl:
delay_ms(60000);//1минута
q++;
TemperatureTakeWater;
if(HotWater){PORTD.6=0;goto endheat;};
if(q<timeofheat_add){goto scl ;};
if(q>timeofheat_add){PORTD.6=0;sound_alarm(2);};
 endheat:
  }
  void test_alarm(){
DoorLockOn;delay_ms(5000);
if ((DoorLockOn )&& (PINB.7 ==0)){sound_alarm(4); };  // дверь не заблокирована
PumpOff;delay_ms(2000);
if((PumpOff)&&(PINB.5==0)){sound_alarm(5);};//   обрыв цепи насоса
PumpOn; delay_ms(1000);
if ((PumpOn)&&(PINB.5==1)){PumpOff;sound_alarm(6);};  // не включается реле насоса
PumpOff;
TemperatureTakeWater;
if(( adc_data>1000)||( adc_data<200)){ sound_alarm(7); }; //  обрыв или замыкание датчика
темпер
if(PINB.3==0){sound_alarm(8);};     //обрыв в цепи коллектора двигателя
delay_ms(2000);
drivertest_on();
delay_ms(1000);
if(shift==0){sound_alarm(9);};  //двигатель не вращается
shift=0;
HeaterOn;delay_ms(3000);
if(PINB.4==0){HeaterOff;sound_alarm(10);};//обрыв в цепи ТЭНа
HeaterOff;
}
void wosh(int woshspeed,int timewcycl,char numberwcycls){                  // int woshtemp
char tunednumberwcycls=0;
IICR1=0;
angl=150;
sound_inform(1);
//немного откачаем остатки
PumpOn;
delay_ms(10000);
PumpOff;
sound_inform(1);
//процедура заполнения бака СМ от ДУ поступает 1 если бак
//наполнился ,выход датчика идет на +5 volt;PINB.2-вход
if(TankEmpty){ValvOn1;};
delay_ms(20000);
//немного нальем воды,а затем включим вращение барабана
```

```
//с одновременным его наполнением

nextwoshcycl:

// реверс
if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле

  //страгивание,разгон на обороты стирки
  TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
   shift=0;
   tmcycl=0;
 while(IICR1>woshspeed){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

//=============================================
//выполнение одного цикла  за заданное время и
//  на заданных оборотах
 tmcycl=0;
while(tmcycl<timewcycl){
if(IICR1>woshspeed){
  if(angl>10){angl-=1;};  //befor 2
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
//================================================
 if(IICR1<woshspeed){
   if(angl<140){ angl+=2;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
  #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);

 };
 };
//=============================================
 // время текущего цикла стирки истекло,реверс
```

```
// TIMSK&=~0x20;//выключаем прерывание от датчика оборотов
shift=1;
tunednumberwcycls++;
if(tunednumberwcycls==numberwcycls){goto endwcycl;};
if(PORTC.3){
Direction = 1;
}
else{Direction=0;};
angl=150;
AlongClock=0;
BackClock=0;
//================================================
/*стоянка между реверсами с проверкой уровня и темнературы
  баке стиральной машины */

if(TankFull){ValvOff1;};
if(TankEmpty){HeaterOff;ValvOn1; };
if(( tunednumberwcycls>15)&&(TankEmpty)){ valvon_add1();};

TemperatureTakeWater;
if(( adc_data>1000)||( adc_data<200)){ sound_alarm(4); };
if(HotWater){HeaterOff;};
if((ColdWater)&&(TankFull)){HeaterOn;};//goto heater;};
//if((NumberWoshSiclesCold>10)&&(ColdWater)){goto heater;};
if((tunednumberwcycls >15)&&(ColdWater)){ coldwaterheater_add();};
delay_ms(5000);
goto nextwoshcycl;
//================================================


endwcycl:

tunednumberwcycls=0;
  HeaterOff;
  pumpon();
 delay_ms(10000);  //3000
 PumpOff;
}



void rinsis1(int rinsspeed,int timerinscycl,char numberrinscycls){
//1,2 полоскание с включением клапана
 char tunednumberrinscycls=0;
 IICR1=0;
 angl=150;
  HeaterOff;
sound_inform(2);
if(TankEmpty){ValvOn1;};
//while(TankEmpty);
delay_ms(8000);
ValvOff1;
```

```
//===========================================
nextrinscycl_1:
// реверс,страгивание,разгон барабана
if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле  //1000
//страгивание,разгон на обороты полоскания
 TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
   shift=0;
   tmcycl=0;
 while(IICR1>rinsspeed){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

//===============================================
//выполнение одного цикла 1,2_го полоскания за заданное время и
//  на заданных оборотах
 tmcycl=0;
while(tmcycl<timerinscycl){
if(IICR1>rinsspeed){
  if(angl>10){angl-=1;};//было 2
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
//================================================
 if(IICR1<rinsspeed){
   if(angl<140){ angl+=2;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
  #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);

 };
 };
//===============================================
 // время одного цикла полоскания закончено,
```

```
 //реверс:
// TIMSK&=~0x20;//выключаем прерывание от датчика оборотов
 IICR1=0;
 tunednumberrinscycls++;
  HeaterOff;
 if(tunednumberrinscycls==numberrinscycls){goto endrinscycl;}
 if(PORTC.3){
Direction = 1;
}
else{Direction=0;};

angl=150;
AlongClock=0;
BackClock=0;
//=============================================
/*стоянка между реверсами с проверкой уровня
 в баке стиральной машины */

if(TankFull){ValvOff1;};
HeaterOff;
if(TankEmpty){ValvOn1; };
if(( tunednumberrinscycls >5)&&(TankEmpty)){ valvon_add1();};
HeaterOff;
 delay_ms(3000);
goto nextrinscycl_1;
//=============================================

endrinscycl:
tunednumberrinscycls=0;
 angl=150;
AlongClock=0;
BackClock=0;
    HeaterOff;
  pumpon();
 delay_ms(5000);
 PumpOff;

}
void rinsis2(int rinsspeed,int timerinscycl,char numberrinscycls){
//третье полоскание с включением двух клапанов для внесения в бак ароматизатора
 char tunednumberrinscycls=0;
 IICR1=0;
 angl=150;
  HeaterOff;
//AlongClock=1;
sound_inform(3);
if(TankEmpty){ValvOn1;ValvOn2;};
delay_ms(8000);
ValvOff1;ValvOff2;
//=============================================
nextrinscycl_2:
```

```
// реверс,страгивание,разгон барабана
if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле  //1000
//страгивание,разгон на обороты полоскания
 TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
   shift=0;
   tmcycl=0;
 while(IICR1>rinsspeed){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

//=============================================
//выполнение одного цикла 3_го полоскания за заданное время и
//  на заданных оборотах
 tmcycl=0;
while(tmcycl<timerinscycl){
if(IICR1>rinsspeed){
  if(angl>10){angl-=1;};//было 2
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
//================================================
 if(IICR1<rinsspeed){
   if(angl<140){ angl+=2;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
  #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);

 };
 };
//=============================================
 // время одного цикла полоскания закончено,
 //реверс:
// TIMSK&=~0x20;//выключаем прерывание от датчика оборотов
 IICR1=0;
  tunednumberrinscycls++;
  HeaterOff;
```

```
  //cycls++;
 if(tunednumberrinscycls==numberrinscycls){goto endrinscycl_2;}
if(PORTC.3){
Direction = 1;
}
else{Direction=0;};

angl=120;
AlongClock=0;
BackClock=0;
//=========================================
if(TankFull){ValvOff1;ValvOff2;};
HeaterOff;
if(TankEmpty){ValvOn1;ValvOn2;};
 if(( tunednumberrinscycls >5)&&(TankEmpty)){  valvon_add1_2();};
 HeaterOff;
delay_ms(3000);
goto nextrinscycl_2;
//=============================================
 endrinscycl_2:
tunednumberrinscycls=0;
     HeaterOff;
     angl=150;
AlongClock=0;
BackClock=0;
    pumpon();
 delay_ms(5000);  //3000
 PumpOff;

}

// SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU_SPU

 void speedup(int startspinspeed,char timespinstep,int setnumberspinsteps,
 char sizespeedstep,int timeendstepLspin){
 int numberspinsteps=0;
 int tunednumberspinsteps=0;
 int tunedspinspeed=0;
  PumpOn;
   HeaterOff;
  while(TankFull);
delay_ms(3000);
 //реверс барабана(переключение реле реверса)
 if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле
//страгивание
 TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
```

```
  shift=0;
  tmcycl=0;
 while(IICR1>startspinspeed){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

 tunedspinspeed=startspinspeed;
 numberspinsteps=setnumberspinsteps;
//=================================================
  /*развиваем обороты определенной скоростной ступени
   и "кружимся"на них время timespinstep*/
kernelstep:
 tmcycl=0;
while(tmcycl<timespinstep){
if(IICR1>tunedspinspeed){
  if(angl>10){angl-=1;};
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
//=================================================
 if(IICR1<tunedspinspeed){
   if(angl<140){ angl+=1;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
  #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);

 };
 };
/* время ступени истекло,переходим на следующую скоростную ступень,
   увеличиваем количество исполненных скоростных ступеней */
    tunednumberspinsteps++;          //увеличиваем количество пройденных ступеней
  if(tunednumberspinsteps<numberspinsteps){
  tunedspinspeed-=(sizespeedstep);    //увеличиваем окружную скорость
  goto kernelstep;};

 //прошли все скоростные ступени и теперь крутимся на достигнутой скрости
 //  время   timeendstepLspin
 tmcycl=0;
 shift=1;
  while(tmcycl<timeendstepLspin){
if(IICR1>tunedspinspeed){
  if(angl>10){angl-=1;};
```

```
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
if(IICR1<tunedspinspeed){
   if(angl<140){ angl+=2;};   //было 1
while(TCNT2<=angl){MotorOff; };
MotorOn;
#asm("nop\nop\nop\nop")
 MotorOff;
while(TCNT2>angl);
 };
  };
 //кончилось время текущего  отжима разгона-укладки
// сбрасываем на ноль внутренние переменные функции,запоминаем направление вращния
// барабана,выключаем оба реле реверса
  numberspinsteps=0;
  tunednumberspinsteps=0;
  tunedspinspeed=0;

  IICR1=0;
  while(TankFull){PumpOn;};
   PumpOff;
  if(PORTC.3){
Direction = 1;
}
else{Direction=0;};
angl=120;
AlongClock=0;
BackClock=0;
}
 void SpeedEnd(int startspinspeed,char timespinstep,int timeendstepLspin){
 int numberspinsteps=0;
 int tunednumberspinsteps=0;
 int tunedspinspeed=0;
  PORTD.1=0;
 //проверка и раскачка бака
  PumpOn;
  while(TankFull);
  delay_ms(5000);
 //реверс барабана(переключение реле реверса)
 if(Direction){
AlongClock=0;BackClock=1;
}
else{AlongClock=1;BackClock=0;
};
delay_ms(500); //задержка для срабатывания реле
//страгивание
 TIMSK|=0x20;//включаем прерывание от датчика оборотов
   angl=150;
```

```
   shift=0;
   tmcycl=0;
 while(IICR1>startspinspeed){
 while(TCNT2<=angl);//{MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 shift=1;

 tunedspinspeed=startspinspeed;
 numberspinsteps=0;
//=====================================================
  /*развиваем обороты определенной скоростной ступени
   и "кружимся"на них время timespinstep*/

kernelstep:
tmcycl=0;
shift=1;
while(tmcycl<timespinstep){
if(IICR1>tunedspinspeed){
  if(angl>10){angl-=1;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };

if(IICR1<tunedspinspeed){
   if(angl<140){ angl+=2;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
  };
 };
/* время ступени истекло,переходим на следующую скоростную ступень,
   увеличиваем количество исполненных скоростных ступеней */
   tunednumberspinsteps++;         //увеличиваем количество пройденных ступеней
  if(tunednumberspinsteps==1){          //1700
   tunedspinspeed-=100;timespinstep=500; goto kernelstep;};
   if(tunednumberspinsteps==2){
   tunedspinspeed-=150; goto kernelstep;};
   if(tunednumberspinsteps==3){
   tunedspinspeed-=150;  goto kernelstep;};
   if(tunednumberspinsteps==4){
   tunedspinspeed-=200;  goto kernelstep;};
   if(tunednumberspinsteps==5){
   tunedspinspeed-=300; goto kernelstep;};
```

```
 if(tunednumberspinsteps==6){
 tunedspinspeed-=300; goto kernelstep;};   // -1200
 if(tunednumberspinsteps== 7){
 tunedspinspeed=300;timespinstep=3000; goto kernelstep;}; //300
 if(tunednumberspinsteps==8){
 tunedspinspeed+=500;timespinstep=1000; goto kernelstep;};
  if(tunednumberspinsteps==9){
 tunedspinspeed+=100; goto kernelstep;};
 if(tunednumberspinsteps==10){
 tunedspinspeed+=100;   goto kernelstep;};
 if(tunednumberspinsteps==11){
 tunedspinspeed=1100;timespinstep=4000; goto kernelstep;};// 1100
 if(tunednumberspinsteps==12){
  tunedspinspeed-=50; timespinstep=1000;goto kernelstep;};
 if(tunednumberspinsteps==13){
 tunedspinspeed-=50;  goto kernelstep;};
if(tunednumberspinsteps==14){
 tunedspinspeed-=100; goto kernelstep;};
 if(tunednumberspinsteps==15){
 tunedspinspeed-=100;timespinstep=2000; goto kernelstep;};
 if(tunednumberspinsteps==16){
 tunedspinspeed-=100;  goto kernelstep;};
  if(tunednumberspinsteps==17){
 tunedspinspeed-=100;  goto kernelstep;};   //-600
 if(tunednumberspinsteps==18){
 tunedspinspeed=300;timespinstep=5000; goto kernelstep;};   // 300
 if(tunednumberspinsteps==19){
 tunedspinspeed+=300;timespinstep=1000; goto kernelstep;};
  if(tunednumberspinsteps==20){
  tunedspinspeed+=500; goto kernelstep;};
 if(tunednumberspinsteps==21){
 tunedspinspeed+=500;  goto kernelstep;};
 if(tunednumberspinsteps==22){
 tunedspinspeed+=100; goto kernelstep;};
 if(tunednumberspinsteps==23){
 tunedspinspeed=1700;timespinstep=4000; goto kernelstep;}; //1700
if(tunednumberspinsteps==24){
 tunedspinspeed-=50;timespinstep=500;  goto kernelstep;};
 if(tunednumberspinsteps==25){
 tunedspinspeed-=50;  goto kernelstep;};
 if(tunednumberspinsteps==26){
 tunedspinspeed-=50; goto kernelstep;};
  if(tunednumberspinsteps==27){
 tunedspinspeed-=50;  goto kernelstep;};
 if(tunednumberspinsteps==28){
 tunedspinspeed-=50; goto kernelstep;};
 if(tunednumberspinsteps==29){
 tunedspinspeed-=50; goto kernelstep;};
 if(tunednumberspinsteps==30){
 tunedspinspeed-=50; goto kernelstep;};
  if(tunednumberspinsteps==31){
  tunedspinspeed-=50; goto kernelstep;};
```

```
  if(tunednumberspinsteps==32){
  tunedspinspeed-=50;  goto kernelstep;};    //-450  1250
  if(tunednumberspinsteps==33){
  tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==34){
  tunedspinspeed-=100; goto kernelstep;};
    if(tunednumberspinsteps==33){
  tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==35){
  tunedspinspeed-=100; goto kernelstep;};
   if(tunednumberspinsteps==36){
  tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==37){
  tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==38){
 tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==39){
  tunedspinspeed-=100; goto kernelstep;};
  if(tunednumberspinsteps==40){
  tunedspinspeed-=100; goto kernelstep;};     //-900 350
    if(tunednumberspinsteps==41){
  tunedspinspeed-=150;timespinstep=1000; goto kernelstep;}; //200
    if(tunednumberspinsteps==42){
  tunedspinspeed-=50; goto kernelstep;};  //100
  if(tunednumberspinsteps==43){
  tunedspinspeed=100; goto kernelstep;};    //100
   if(tunednumberspinsteps==44){
  tunedspinspeed=80; };               //180

 //прошли все скоростные ступени и теперь крутимся на достигнутой скрости
 //  время   timeendstepLspin
 tmcycl=0;
 shift=1;
  while(tmcycl< timeendstepLspin){
if(IICR1>tunedspinspeed){

  if(angl>10){angl-=1;};
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
                  //длительность включения семистора до конца синусоиды,те сброса TCNT2
  };

if(IICR1<tunedspinspeed){
   if(angl<140){ angl+=2;};//было 1
 while(TCNT2<=angl){MotorOff;};
 MotorOn;
 #asm("nop\nop\nop\nop")
 MotorOff;
 while(TCNT2>angl);
```

```
                    //длительность включения семистора до конца синусоиды,те сброса TCNT2
  };
  };
 //кончилось время текущего  отжима разгона-укладки
// сбрасываем на ноль внутренние переменные функции,запоминаем направление
вращения
// барабана,выключаем оба реле реверса
  MotorOff;
  AlongClock=0;
  BackClock=0;
  numberspinsteps=0;
  tunednumberspinsteps=0;
  tunedspinspeed=0;
  IICR1=0;
  angl=150;

}
```

При повторении проекта надо сделать следующее:
Скачать с 123nm.ru compiler CodeVisionAVR
Установить его на компьютер.
Скопировать wm_evo2_1.inc в папку inc компилятора.
Скопировать wm_evo2_1.lib в папку lib компилятора.
Ознакомиться с системой программирования и создать новый проект.
Включить в состав проекта файл evo2_1.c.
Всё теперь можете играться с программой :видоизменять её,добавлять новые функции, к примеру в программе нет функции управления дисбалансом барабана,функции управления импульсным замком блокировки люка.Кто то захочет поставить атмегу и в панель управления и управлять нижней платой сверху,добавить несколько видов стирки…….и… Васюки уже становятся центром всего шахматного мира…..

Описание создания платы управления СМ Indezit на базе платы EVO2.

Описываемые приемы универсальны и применимы ко всем платам управления всех СМ.
Программное обеспечение универсально для всех типов СМ ,задача ремонтника обеспечить заданные потенциалы на выводах устанавливаемого микроконтроллера .
Делаю я это так.
Тщательно проверяю двигатель,клапана,датчик уровня,датчик температуры,ТЭН,замок люка,и их обвязку(подсоединение к плате управления).
На плате проверяем исправность блока питания;тиристоров клапанов,люка двери,дигателя;реле насоса,ТЭНа,реверса двигателя…
Удаляем старый микроконтроллер.Легче произвести это с помощью паяльного фена, но процессор достаточно большой и при сильном нагреве платы возникает опасность порчи других элементов.На некоторых конструкциях (LG) нагрев вообще проблематичен.
Отрезаем ножки от процессора(аккуратно,чтобы не повредить дорожки на плате),затем убираем корпус процессора и снимаем паяльником его ножки.Здесь важно не повредить дорожки.Место,где стоял процессор оставляем доступным. Ведь здесь надо припаивать проводники,идущие к новому процессору (у меня Atmega8).
На плате ничего другого не не снимаем из-за возможности порчи того,что потребуется в дальнейшем.
Теперь наиболее трудоемкая и требующая терпения работа-определение какая ножка удаленного МК чем управляла,на какие ножки принимала.Берем наихудший вариант,

когда нет схемы СМ и её платы управления.
Нам нужно определить:
Вывод управления главного двигателя.
Вывод управления 1 реле реверса.
Вывод управления 2 реле реверса.
Вывод управления реле сливного насоса.
Вывод управления реле нагревательного элемента.
Вывод управления 1-м клапаном залива воды в СМ.
Вывод управления 2-м клапаном залива воды в СМ.
Вывод управления блокировкой люка СМ.
Вывод поступления сигнала полный бак.
Вывод прихода сигнала с датчика оборотов двигателя.
Вывод прихода сигнала с датчика температуры.
Это можно сделать двумя основными способами:
1.Распустить шлейф проводов СМ и идя по продам до платы определяем соединительный разъем данного устройства,управляющий прибор на плате(реле,тиристор) затем по дорожкам платы добираемся до определямого вывода процессора.Это достаточно сложно и требует навыка.
2.Другой способ,особенно удобен при наличии под рукой СМ,запуск исполнительных механизмов СМ посредством подачи соответствующих сигналов на выводы МК.
Тут необходимо немного остановиться и заявить совершенно ясно и отчетливо: РЕМОНТ СТИРАЛЬНОЙ МАШИНЫ, ВКЛЮЧЕННОЙ В ЭЛЕКТРОСЕТЬ, СМЕРТЕЛЬНО ОПАСЕН. В связи с использованием электрических аппаратов установленных на стиральной машине с рабочим напряжением на 220 вольт, их подключением к электросети и их тиристорным управлением ВСЕ ТОЧКИ ПЛАТЫ УПРАВЛЕНИЯ СМ НАХОДЯТСЯ ПОД СИЛОВЫМ НАПРЯЖЕНИЕМ 220 ВОЛЬТ. Именно силовым напряжением,т.е. его ток неограничен.На стиральной машине нет никаких предохранителей и защит,вся надежда на электропробки квартиры,но током их срабатывания Вас сожжет дотла.
Там,в проводах,деталях, соединенных с электросетью, сидит безжалостный “ЗВЕРЬ”и ждет Вашей оплошности,поспешности,неграмотности в действиях.Не спешите,все предварительные операции выполняйте при выключенной машине,занимайте устойчивое положение(исключающее возможность падения на аппарат),обязательно подключите контрольную лампочку на 220 вольт,настойчиво на напоминающую Вам:

ОПАСНО, 220 ВОЛЬТ ПОДАНО!

И последнее по этому вопросу:работайте только одной рукой
и предварительно заизолировав себя от земли,работайте в одежде с длинными рукавами,убирайте волосы(космы).
Исполнительные механизмы на плате управляются процессором путем установки на его выходах нуля или единицы.К примеру EVO2 двигатель,реле реверса,насос,реле нагревательного элемента включаются единицей(+5 вольт),а клапана и замок люка нулем.Теперь берем резистор около 1 килоома припаивам его к шине +5 вольт к другому концу резистора припаваем проводник с тонким наконечником на другом конце,можно сделать из медицинского шприца.Отсоединяем шланг датчика уровня,подключаем свой шланг,имитируем полный бак и подключаем контрольку к проводам водонагревателя(предварительно сняв один провод с вывода нагревателя).
Практически всё готово.Включаем машину и начинаем ,нет ещё одно забыл,необходимо Сделать чертёж процессора 4 стороны по 16 ножек и на углах проставить 1-16-32-48-64 когда чертёж перед глазами легче высчитывать номер ножки относительно угла.
Затем начинаем перебирать ножки выставляя нашим щупом 1 простаивая на каждой ножке не менее пяти секунд(чтобы успел нагреться и сработать замок).Мы услышим как шёлкают релюшки Должно быть четыре устойчивых щелчка-2 реле реверса,реле ТЭНа и реле насоса и один –замок люка.Когда сработает замок,подаем единицу на выход

управления ТЭНом контрольная лампочка на ТЭНе должна загораться.От реле насоса должен вращаться насос,оставшиеся две- реле реверса.Подаем через дополнительную цепь единицу на выход управления одной из реле реверса и оставляем её,одно реле реверса останется включенным.Начинаем щупом кратковременно касаться дорожек при одном из касаний рванет двигатель барабана,не гоняйте его.

Нашли все выводы управления механизмами.

Теперь надо получить импульсы частоты электросети,импульсы датчика оборотов двигателя барабана,датчик температуры.

Подключаем осциллограф и проходим его щупом по дорожкам пока не найдем прямоугольные импульсы частотой 50 герц.

Вращая двигатель вручную проходим его щупом по дорожкам пока не обнаружим изменяющиеся по частоте и периоду прямоугольные импульсы.

Далее отключаем СМ от электросети !

Датчик температуры придется вычислять по плате вручную,идя от разъема подключения его к плате до вывода микропроцессора.

Всё,самая трудная часть работы пыполнена можем приступать к установке и распайке нашего микоконтроллера.

Выбираем место для установки нового процессора,вырезаем из плотного негорючего материала прямоугольник размером чуть больше панельки под микросхему и грубо приклеиваем его на выбранное место.

Берем панельку под микросхему на 28 гибких выводов,отгибаем их слегка вверх и приклеиваем к прямоугольнику сверху,стараясь,чтобы выводы панельки были доступны для пайки(не были залиты клеем).

Соединяем дорожки под выводы старого микроконтроллера с выводами панельки под новый МК одинаковые по своему функциональному назначению.Всё монтаж закончен.

Я использую эмалированный провод диаметром 0.2 мм (от обмотки сливного насоса).

Здесь самое узкое место всей конструкции,т.к. расстояние между дорожками из-под удаленного микроконтроллера составляет малую долю миллиметра.Нужен паяльник с точечным жалом.Проводники вымерять точно по длине и залуживать обильно перед установкой на плату.Дорожки на плате освежить и проверить на отсутствие замыкания между ними.Паяльник доработать надев на него теплоизляционную термостойкую трубу,чтобы держать теперь уже “паяльное устройство”не за ручку паяльника ,а за надетую трубу(корпус устройства)в непосредственной близости от жала паяльника.

Я приспособил корпус от малогабаритного фена.Теперь рука стоит точно.Очки для четкого контроля качества пайки.Бинокулярный микроскоп – это идеально,но дорого.

Для контроля качества пайки можно приспособить и WEB камеру,но нужен современный скоростной мультимедийный компьютер для обеспечения наблюдения процесса работы В реальном времени.Об этом,по моему,всё. Нужно выполнить около 20 паек.

Теперь панелька под микросхему распаяна и самое время проверить правильность функционирования испонительных элементов платы упрапвления.Опять же идеальный вариант это когда есть стиральная машина,тогда просто подсоединяем ПУ к СМ и на соответствующие на выводы микроконтроллера подаем ноль или единицу, проверяем отработку стиральной машиной в целом той или иной команды(запуск насоса,блокировка люка,появление сигнала пустого или полного бака итд).

Не забываем технику безопасности при работе под напряжением !

Если все исправно,то выключаем питание, вставляем в панельку микроконтроллер с загруженной программой и включаем СМ в сеть Запускается программа инициализации и проверки а затем начинается программа стирки. Всё.

Привожу цоколевку распайки выводов установленного микроконтроллера на плате EVO2.

1 пустой.
2 пустой.
3 пустой.
4 пустой
5 частота 50 герц на 51(имп)
6 запуск 1 клапана на 36 (ноль)
7 питание МК +5 вольт
8 питание МК -5 вольт.
9 пустой.
10 ????
11запуск 2 клапана на 40(ноль)
12 включение ТЭНа на 54(1вкл)
13 пустой.
14 прием сигнала от датчика оборотов двигателя на 30(импульсы)
15 управление двигателем на 30 (ШИМ)
16 сигнал от датчика уровня на 23(1- полный)
17 контроль цепи двигателя на 13(1-исправна)
18 пустой
20 +5 вольт.
21 пустой
22 минус 5 вольт.
23 от датчика температуры на 59(1-4 вольта)
24 на датчик звукового сигнала.(пила)
25 на реле реверса на 28(1 включено)
26 на реле реверса на 29(1 включено)
27 на реле насоса на 21(1включено
28 на тиристор замка дверей на 42 (ноль вкл.)

Выводы установленного микроконтроллера по своему назначению остаются неизменными для любой платы любого производителя,но искать точки подключения на плате придется заново, иногда вручную,
если нет описания платы или выводов микроконтроллера на ней.Но в любом случае перед установкой МК на панельку необходимо проверить правильность подключения выводов путем подачи управляющих сигналов и замеров приходящих от датчиков электрических величин.Как было уже очевидно замечено читателем автор данной конструкции удалил функцию взаимодействия СМ с пользователем,это не совсем так(хотя одна совешенно слепая и немая машинка работает,не жалуются).Оставлена звуковая сигнализация от СМ пользователю: 1 звуковой сигнал – начало стирки одиночный
2 звуковых сигнала 1 полоскание одиночный
3 звуковых сигнала 2 полоскание одиночный
4 звуковых сигнала 3 полоскание и отжим одиночный
5 звуковых сигнала – конец стирки периодическиq

Аварийные сигналы:короткие после них звучит 1 длинный
1 плохой налив
2 плохой нагрев
3 плохой слив
4 дверь не заблокирована
5 обрыв цепи насоса
6 обрыв или замыкание датчика температуры
7 обрыв цепи ТЭНа
8 обрыв цепи двигателя
9 двигатель не вращается

При появлении серии: короткие за ними длинный звонок- работа машины прекращается –машина стоит и сигналит до её выключения из сети.
Вот пожалуй и все,пока.Всегда ,при любой плате на описанных ножках установленного микроконтроллера будут появляться одинаковые управляющие сигналы,всегда нужно подавать на точно описанные ножки сигналы от датчика двигателя,частоты сети 50 герц (для синхронизации системы управления двигателем), датчика уровня и датчика температуры.Если трудно обеспечить сигналы проверки двигателя,закрытия дверей,насоса нужно повесить +5 вольт на описанные информационные входы через резисторы порядка 10 ком,тем самым обеспечив программу информацией об “исправности” данных устройств.

При разработке данного устройства практически все получено с сайта:avr123.nm.ru.
Написано на CodeVisionAVR version:1.25.9 standart.Микроконтроллер прошивается встроенным в эту программу загрузчиком.
Пользуясь случаем выражаю мою скромную благодарность работникам avr123.nm.ru и желаю им всего хорошего,счастья в личной жизни.
С уважением ко всем читателям данной статьи.
Владимир.

Вот и Всё!