

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Институт космических и информационных технологий
Кафедра «Вычислительной техники»

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №9

Преподаватель

подпись, дата

В.Г. Середкин
инициалы, фамилия

Студент

КИ20-06Б, 032049103
номер группы, зачетной книжки

подпись, дата

Д.А. Анциферов
инициалы, фамилия

Красноярск 2023

Введение

Цель работы: проектирование системы управления объектом (программная часть).

Задание:

Разработать алгоритм функционирования и написать программу для объекта управления, приведенного в предыдущей работе в соответствии с заданным вариантом, базируясь на его информационной и структурной модели.

Вариант: микропроцессорная система управления круиз-контролем для автомобиля.

Ход работы

План работы.

Для упрощения проектирования системы управления объектом разделим на 4 задачи:

1. Распределение портов ППА;
2. Формирование управляющего кода ППА;
3. Формирование адресов в пространстве;
4. Алгоритм программы;
5. Листинг программы.

Задача 1. Распределение портов ППА

Распределим порты ППА следующим образом:

1. порт *A* программируем на ввод информации, получаемой через АЦП с сервопривода-датчика;
2. старшую часть порта *C* (а именно *C7*) программируем на ввод информации, получаемой от АЦП и означающей готовность АЦП;
3. младшие части порта *C* (а именно *C1* и *C0*) программируем на вывод звука на микро-динамик (*C1*) и информации, тем самым обеспечивается пуск АЦП (*C0*).
4. порт *B* программируем на вывод информации. Обеспечивает управление блоками управления сервоприводами на тросиках газа и тормоза.

Задача2. Формирование управляющего кода ППА

Формат управляющего слова: «1», следовательно, старший бит $УС = 1$. Обе группы портов будут работать в режиме «однонаправленный ввод/вывод без квитирования», следовательно 7, 6 и 3 бит = 0.

Так как группа А запрограммирована на ввод, 5 и 4 бит $УС = 1$.

Группа В запрограммирована на вывод, следовательно, первый и второй биты $УС = 0$.

Таким образом, управляющее слово: 10011000 в двоичной системе счисления.

Задача 3. Формирование адресов в пространстве

Адреса и выбранные порты представлены в таблице 1. Адреса представлены в 16-ричной системе счисления.

Таблица 1 – Адреса и выбранные порты

A1	A0	Адрес	Выбранный порт
0	0	0x00	Порт А
0	1	0x01	Порт В
1	0	0x02	Порт С
1	1	0x03	Регистр УС

Задача 4. Алгоритм программы.

Алгоритм основной программы представлен на рисунке 1.

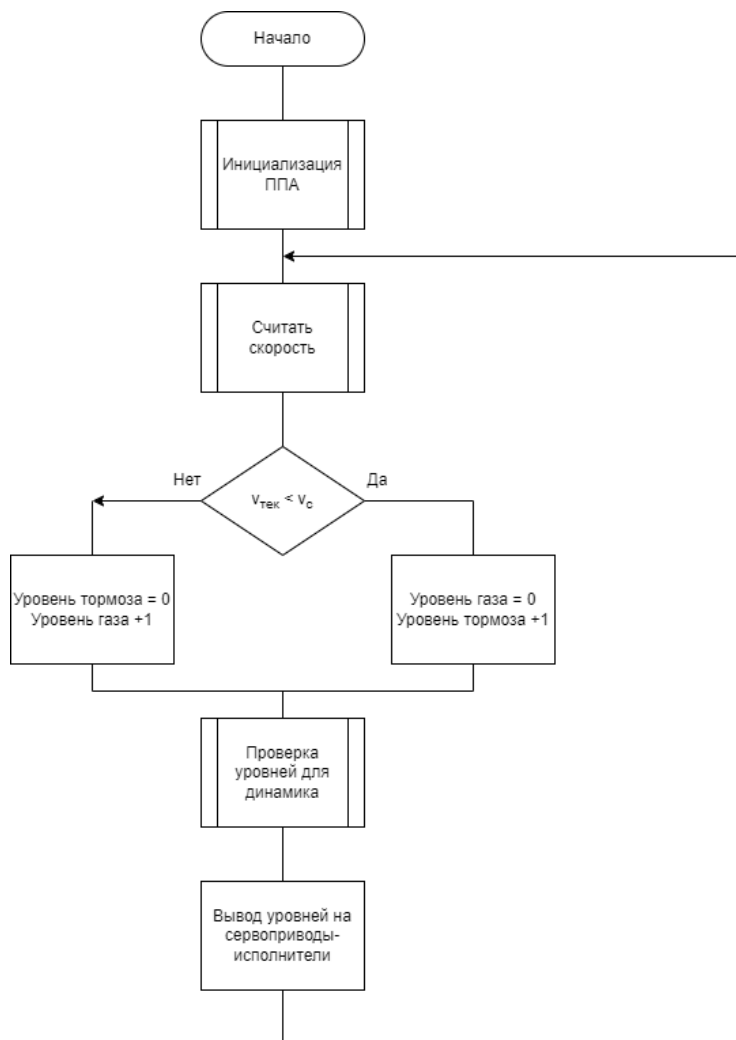


Рисунок 1 – Алгоритм основной программы

На рисунке 2 представлен алгоритм инициализации параллельно программируемого адаптера (PPI).



Рисунок 2 – Алгоритм инициализации PPI

На рисунке 3 представлен алгоритм чтения скорости в PPI.

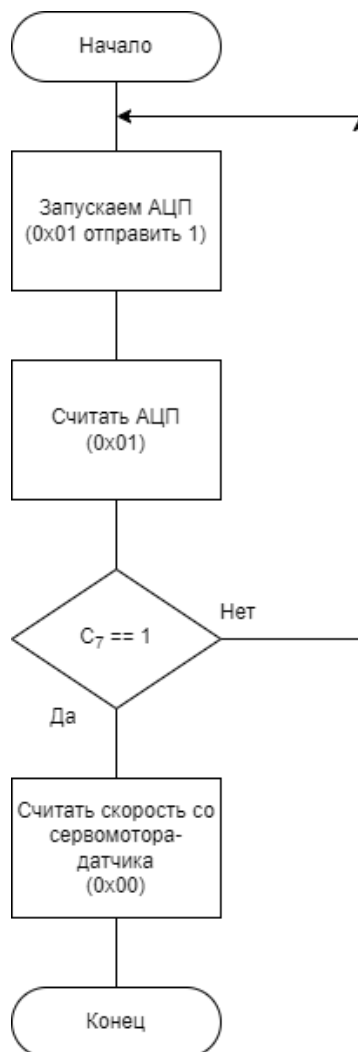


Рисунок 3 – Алгоритм алгоритм чтения скорости в PPI

Код программы представлен в листинге 1.

Листинг 1 – Код программы

1	; Регистр В хранит скорость, которую нужно поддерживать
2	; Регистр С хранит скорость, которая на данный момент
3	; Регистр D хранит угол наклона сервомотора на троселе газа
4	; Регистр E хранит угол наклона сервомотора на троселе тормоза
5	; Регистр H хранит уровень нажатия на газ
6	; Регистр L хранит уровень нажатия на тормоз
7	
8	; Основная часть программы
9	.adr:0x0000
10	CALL Init_PPI
11	Main:
12	CALL Read_PPI
13	MOV C, A ; Записываем текущую скорость
14	CALL Check_Pedals
15	MOV A, B
16	CPI 0x00
17	JNZ Main_Cycle ; Если B == 0
18	MOV B, C ; B = входящей скорости
19	Main_Cycle:
20	CPM C ; Сравниваем скорость текущую и поддерживаемую
21	JZ Main ; Если равна, то ничего делать не нужно
22	JC Up_Speed ; Если текущая < поддерживаемой, то поднимаем
23	скорость
24	MVI H, 0 ; Иначе тормозим
25	MOV A, E ; +1 к уровню наклона тормоза
26	ADI 0x01
27	MOV E, A
28	INR L ; +1 к уровню торможения
29	MOV A, L
30	CALL Check_Sound
31	MOV A, E
32	OUT 0x01 ; Тормозим
33	JMP Main
34	Up_Speed:
35	MVI L, 0 ; Добавляем скорость
36	MOV A, D ; +1 к уровню наклона сервомотора газа

37	ADI 0x10	
38	MOV D, A	
39	INR H	; +1 к уровню газа
40	MOV A, H	
41	CALL Check_Sound	
42	MOV A, D	
43	OUT 0x01	; Ускоряемся
44	JMP Main	
45		
46	; Инициализация ППА	
47	.adr:0x0100	
48	Init_PPI:	
49	MVI A, 0b01100010	
50	OUT 0x03	
51	RET	
52		
53	; Чтение с порта A	
54	.adr:0x0150	
55	Read_PPI:	
56	MVI A, 0x01	
57	OUT 0x02	
58	Cycle_Read:	
59	IN 0x02	
60	ANI 0b10000000	
61	JZ Cycle_Read	
62	IN 0x00	
63	RET	
64		
65	; Проверка на включение динамика	
66	.adr:0x0200	
67	Check_Sound:	
68	ADI 16	; Если один из уровней == 16
69	JZ On_Sound	; Включаем звук на динамике
70	JMP Off_Sound	; Тогда выключаем звук
71	On_Sound:	; Если максимальный уровень газа/тормоза
72	MVI A, 0b00000010	
73	OUT 0x02	; Включаем звук
74	JMP Sound_End	
75	Off_Sound:	; Если уровни газа/тормоза в норме

76	MVI A, 0b00000000
77	OUT 0x02 ; Выключаем звук
78	Sound_End:
79	RET