**3.6. Работа с АЦП в ГПУ**

**3.6.1. Модуль аналого-цифрового преобразования.**

Модуль реализует интерфейс между датчиками-измерителями параметров внешнего объекта управления и блоком обработки MCU. Информация с измерительного устройства представлена аналоговым-непрерывным

сигналом напряжения. Модуль обеспечивает преобразование-дискретизациюнепрерывного сигнала и двоичное кодирование, MCU извлекает информацию – значение измеряемого параметра. При этом используются либо прямые аналитические зависимости (например, T(U)-температура от U, V(U)-влажность от U, ..) или таблицы интерполяции (например, зависимость скорости воздушного потока V(U)).

Таким образом, осуществляется ввод измеряемых параметров и предварительная обработка.

**Принцип работы ADC преобразователя SAB515/535**.

Структура блока измерения с учетом встроенных в SAB515 схем управления аналого-цифровым преобразованием включает:

1) 8 адресуемых аналоговых входа порта Р6

2) регистр команд ADCON

3) регистр выбора масштаба или опорного напряжения DAPR

4) результат преобразования - регистр ADDAT

5) Схему ADC 8-разрядного преобразования S(Ux), требует 15 циклов

Принцип работы преобразователя SAB515:

1. в регистре ADCON=BSY.ADM.MX записывается

- адрес канала MX=ADCON(3.0),

- режим измерений ADCON(4)=(0/1)=(один отсчет/многократный отсчет)

2) запись константы в регистр DAPR запускает преобразование

DAPR=0 соответсвует 8-разрядному преобразованию в диапазоне

0-5в, значение S считывается при завершении преобразования

с 8-разрядного регистра ADDAT и преобразуется в машинное значение

Uвх=5в/256\*S мв

3) Для 10-разрядного преобразования необходимо выбрать более

точный диапазон опорного напряжения Uн-Uв с интервалом 1.25v,

в котором находится измеряемый сигнал.

В регистре DAPR=H.L указывается диапазон опорного напряжения ,

где H={0, f} код, обозначающий верхнее значение опорного напряжения

Uв=5в/16\*H=0.313\*H, тогда L=H-4, что соответсвует интервалу 1.25 в и

Uн= Uв-1.25в

При повторном 8-разрядном преобразовании в этом диапазоне

точность измерения dU соответствует 1.25/256в . С учетом

значения Uн, Uвх=Uн+dU=Uн+S\*1.25/256 имеет 10-разрядную точность.

Значение Uн=[S/16]\*5v/256, где [..]-8-разрядное целое с недостатком.

Константа HL, где [S/16]=L и H=L+4, записывается в DAPR при

повторном преобразовании.

Это значение можно использовать в прямых вычислениях параметра

по известным формулам или по таблице линейной интерполяцией

зависимости A(U).

Цикл 8-разрядного преобразования занимает 15 мкс, что соответствует частоте дескритизации около 60 кГц. В течение цикла измерений необходимо выполнить - запись в память и инкремент адреса. При этом реальная частота дескритизации снижается до 10 кГц.

10-разрядное преобразование с масштабированием Uref занимает 150мкс, что соответствует

6 кГц частоте квантования.

**3.6.2**. **Формирования звука и измерение уровня**

В SDС к выводу P1.5 подключается динамик. Формируя на выходе порта меандр звуковой частоты, получаем на динамике контролируемое звучание.

Полезно выполнить исследование частотных характеристик динамика, U(f) – уровень звука в зависимости от частоты c использоваyием АЦП и микрофона. В среднем частота звуковой частоты 500-2000 Гц.

Частота дискретизации для надежной идентификации частоты f звукового сигнала в соответствии с теоремой Котельникова должна быть не менее 4f.

Таким образом время преобразования должно быть не более 1/(4f)~250-100 мкс,

Встроенный 8-битовый АЦП обеспечивает при частоте 12 МГц задержку не более 16 мкс.

Преобразование запускается по прерыванию от таймера через 100 мкс – при частоте звука

1000 Гц можно получить необходимую точность измерения.

500 значений измерения сохраняются во внешней памяти XRAM . Затем измеряется период и частота смещенного звукового сигнала вида, уровень звука определяет смещение однополярного сигнала Uсм

Um

Uсм

0

Функция формирования звука для SBC.

#include <reg515.h>

#include <stdio.h>

sbit CC0 =P1^0; //бит, к которому подключен динамик

**sound(unsigned int L,unsigned int T)** // Т-полупериод меандра

//L -длительность звука

{ unsigned int z=T;

while(L--)

{

z=T; //1 полупериод

CC0=~CC0;

while(z--);

z=T; //2 полупериод

CC0=~CC0;

while(z--);

}

}

**unsigned int xdata mm[500]**; //целый массив данных

int i=0; //текущий индекс в массиве

**tm0() interrupt 1 //**подпрограмма АЦП

{ char t=10;

TR0=0; остановить счетчик

DAPR=0; //запуск ацп

while(t--); //задержка преобразования

mm[i++]=ADDAT; //сохранение в массиве

TR0=1;

}

**main()**

{

int j=100;

TMOD=TMOD|0x2; //2 режим таймера 0

TH0=-200; //прерывание через 200 мкс

ET0=1;

ADCON=0; //использовать 0-канал с микрофоном и однократное преобразование

getkey(); TI=1; //начало в терминальном режиме – вводом любого символа

sound(200,1000); //начало формирования звука

TR0=1; //разрешение прерывания - АЦП

EAL=1; //разрешение прерывания

while(j>i) //цикл формирования звука и ввода с АЦП

sound(200,1500); //100-200, 500-1500 звук

EAL=0; //запрет прерывания

for(i=0;i<400;i++) //контрольный вывод массива в терминальном режиме на дисплее

printf(" %2d ",mm[i]);

// найти смещение - интенсивность звука и частоту

while(1); //динамический останов

}