Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет программной инженерии и компьтерной техники

Лабораторная работа №4 по Проектированию вычислительных систем

Вариант 1

Роман Юнусов

Алексей Лапин  
Группа P34102

Преподаватель Василий Пинкевич

Санкт-Петербург 2024

Оглавление

[Задание 3](#_Toc181968277)

[Схемы 3](#_Toc181968278)

[Решение 4](#_Toc181968279)

[Последовательный uart. 4](#_Toc181968280)

[Uart через прерывания. 5](#_Toc181968281)

[Парсер(анализ строки) 6](#_Toc181968282)

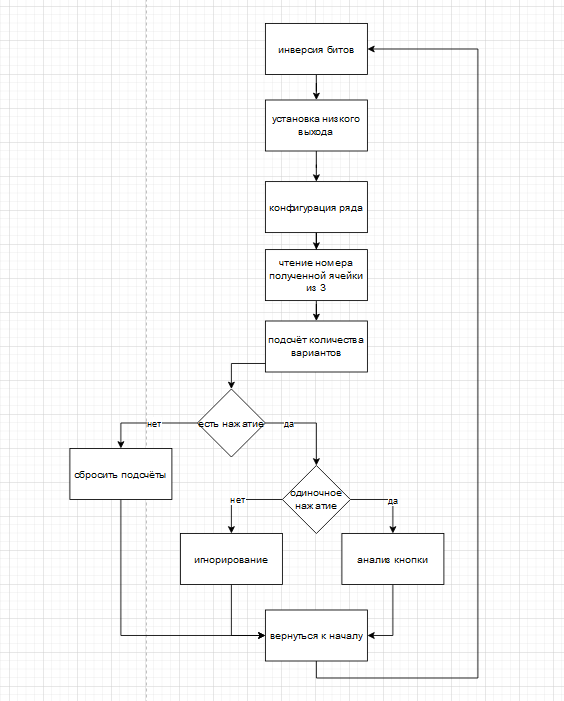
[Выводы 7](#_Toc181968283)

# Задание

Модифицировать прошлую работу, отказавшись от юарт, переделав её на механическую клавиатуру в стенде. Для задания необходимо использовать интерфейс i2c.

# Схемы

Основной алгоритм анализа. Всё делаем по методичке.



Анализ подсчётов, с защитой от множественной регистрации нажатия

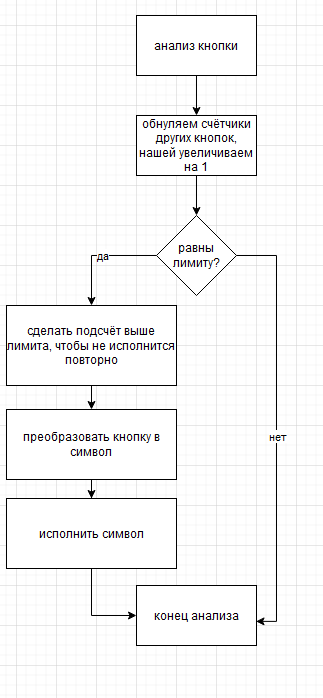
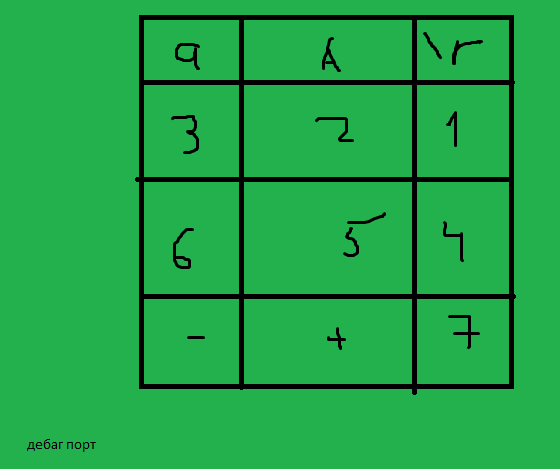


Схема рабочих кнопок на клавиатуре



# Решение

Собственно основной алгоритм проверки состояния кнопок  
HAL\_StatusTypeDef **Set\_Keyboard**( **void** ) {

HAL\_StatusTypeDef ret = *HAL\_OK*;

uint8\_t buf;

buf = 0;

ret = PCA9538\_Write\_Register(KBRD\_ADDR, *POLARITY\_INVERSION*, &buf);

**if**( ret != *HAL\_OK* ) {

**char** out\_buffer[100];

**int** transmitted\_data\_len;

transmitted\_data\_len = **sprintf**(out\_buffer, "Error write polarity\r");

HAL\_UART\_Transmit( &huart6, (uint8\_t \*) out\_buffer, transmitted\_data\_len, 30 );

**goto** exit;

}

buf = 0;

ret = PCA9538\_Write\_Register(KBRD\_ADDR, *OUTPUT\_PORT*, &buf);

**if**( ret != *HAL\_OK* ) {

**char** out\_buffer[100];

**int** transmitted\_data\_len;

transmitted\_data\_len = **sprintf**(out\_buffer, "Error write output\r");

HAL\_UART\_Transmit( &huart6, (uint8\_t \*) out\_buffer, transmitted\_data\_len, 30 );

}

exit:

**return** ret;

}

uint8\_t **Check\_Row**( uint8\_t Nrow ) {

uint8\_t Nkey = 0x00;

HAL\_StatusTypeDef ret = *HAL\_OK*;

uint8\_t buf;

uint8\_t kbd\_in;

**int** cnt = 0;

ret = Set\_Keyboard();

**if**( ret != *HAL\_OK* ) {

**char** out\_buffer[100];

**int** transmitted\_data\_len;

transmitted\_data\_len = **sprintf**(out\_buffer, "Error write init\r");

HAL\_UART\_Transmit( &huart6, (uint8\_t \*) out\_buffer, transmitted\_data\_len, 30 );

}

buf = Nrow;

ret = PCA9538\_Write\_Register(KBRD\_ADDR, *CONFIG*, &buf);

**if**( ret != *HAL\_OK* ) {

**char** out\_buffer[100];

**int** transmitted\_data\_len;

transmitted\_data\_len = **sprintf**(out\_buffer, "Error write config\r");

HAL\_UART\_Transmit( &huart6, (uint8\_t \*) out\_buffer, transmitted\_data\_len, 30 );

}

ret = PCA9538\_Read\_Inputs(KBRD\_ADDR, &buf);

**if**( ret != *HAL\_OK* ) {

**char** out\_buffer[100];

**int** transmitted\_data\_len;

transmitted\_data\_len = **sprintf**(out\_buffer, "Read error\r");

HAL\_UART\_Transmit( &huart6, (uint8\_t \*) out\_buffer, transmitted\_data\_len, 30 );

}

kbd\_in = buf & 0x70;

Nkey = kbd\_in;

**if**( kbd\_in != 0x70) {

**if**( !(kbd\_in & 0x10) ) {

**switch** (Nrow) {

**case** ROW1:

cnt++;

Nkey = 0x07;

**break**;

**case** ROW2:

cnt++;

Nkey = 0x04;

**break**;

**case** ROW3:

cnt++;

Nkey = 0x01;

**break**;

**case** ROW4:

cnt++;

Nkey = 0x00;

**break**;

}

}

**if**( !(kbd\_in & 0x20) ) {

**switch** (Nrow) {

**case** ROW1:

cnt++;

Nkey = 0x08;

**break**;

**case** ROW2:

cnt++;

Nkey = 0x05;

**break**;

**case** ROW3:

cnt++;

Nkey = 0x02;

**break**;

**case** ROW4:

cnt++;

Nkey = 0x0B;

**break**;

}

}

**if**( !(kbd\_in & 0x40) ) {

**switch** (Nrow) {

**case** ROW1:

cnt++;

Nkey = 0x09;

**break**;

**case** ROW2:

cnt++;

Nkey = 0x06;

**break**;

**case** ROW3:

cnt++;

Nkey = 0x03;

**break**;

**case** ROW4:

cnt++;

Nkey = 0x0C;

**break**;

}

}

}

**else** Nkey = 0x0D;

**if**(cnt > 1)

Nkey =0x0F;

**return** Nkey;

}

К алгоритму выше для каждого ряда обраащается функция, задача которой вернуть номер кнопки которая была нажата, либо вернуть информацию о множественном нажатии.

uint8\_t **checkButtons**(){

**int** cnt = 0;

uint8\_t mem = 0;

uint8\_t key1 = Check\_Row(ROW1);

**if** (key1 != 0x0D){

cnt++;

mem = key1;

}

**if** (key1 == 0x0F){

**return** key1;

}

uint8\_t key2 = Check\_Row(ROW2);

**if** (key2 != 0x0D){

cnt++;

mem = key2;

}

**if** (key2 == 0x0F){

**return** key2;

}

uint8\_t key3 = Check\_Row(ROW3);

**if** (key3 != 0x0D){

cnt++;

mem = key3;

}

**if** (key3 == 0x0F){

**return** key3;

}

uint8\_t key4 = Check\_Row(ROW4);

**if** (key4 != 0x0D){

cnt++;

mem = key4;

}

**if** (key4 == 0x0F){

**return** key4;

}

**if**(cnt == 0)

**return** 0x0D;

**if**(cnt > 1)

**return** 0x0F;

**return** mem;

}

Дальше две функции. Одна просто постоянно запускается в внутри основного while с иф, и если она вернет 1, то значит есть символ, который нужно обработать. А вторая занимается подсчётом количества нажатий на кнопку, чтобы после достижения лимита вернуть номер кнопки, которую можно считать нажатой.

**int** **checkKB**(**char**\* el){

uint8\_t res = checkButtons();

**if**(res == 0x0F) {

**return** 0;

// если несколько нажатых кнопок, просто игнорируем итерацию

}

**if**(res == 0x0D) {

init\_cnt();

**return** 0;

}

**int** pressed\_button = increase\_cnt(res);

**if**(pressed\_button == -1){

**return** 0;

// кнопка не готова

}

convert\_button\_to\_symbol(pressed\_button, el);

**return** 1;

}

**int** **increase\_cnt**(uint8\_t button\_num){

**for**(**int** i = 0;i < 13;i++){

**if**(i != button\_num){

click\_cnt[i] = 0;

}

**else**{

**if**(click\_cnt[i] <= DREBLIM)

click\_cnt[i] = click\_cnt[i] + 1;

}

}

**if**(click\_cnt[button\_num] == DREBLIM){

**return** button\_num;

}

**return** -1;

}

Также есть банальная функция для преобразования номера кнопки в символ, который нужно скормить анализатору для нужного результата.

**void** **convert\_button\_to\_symbol**(**int** button\_num, **char**\* el){

**if**(button\_num == 0){

(\*el) = '\r';

}

**if**(button\_num == 1){

(\*el) = '1';

}

**if**(button\_num == 2){

(\*el) = '2';

}

**if**(button\_num == 3){

(\*el) = '3';

}

**if**(button\_num == 4){

(\*el) = '4';

}

**if**(button\_num == 5){

(\*el) = '5';

}

**if**(button\_num == 6){

(\*el) = '6';

}

**if**(button\_num == 7){

(\*el) = '7';

}

**if**(button\_num == 8){

(\*el) = '+';

}

**if**(button\_num == 9){

(\*el) = '-';

}

**if**(button\_num == 11){

(\*el) = 'A';

}

**if**(button\_num == 12){

(\*el) = 'a';

}

}

Схема клавиатуры

# Выводы

Я научился настраивать интерфейс I2c для работы с устройством через расширитель порта. Далее были получены ценные навыки считывания информации с механической клавиатуры и последующей их алгоритмической обработки, чтобы избавиться от дребезга и залипания.