**PICµBASIC V3.4.x**

**Примеры написания программ**

 **Оборудование - любой контроллер с подключенным символьным дисплеем и кнопками**

**Пример #1: Обработка нажатия кнопок.**

 Получим код нажатой кнопки и отобразим на дисплее

Выполним оператор ***LINIT 1***

Выполняется инициализация и очистка дисплея

Напишем программу

***0010 A=GKEY{}:AT 0:LPRINT A," ":GOTO 10***

Запустим

***RUN 10***

На дисплее отображается код нажатой кнопки или ноль, если не нажата.

Разберем программу подробно

***0010*** – номер строки программы

***A=GKEY{}*** – функция, присваивает переменной А код нажатой кнопки или 0 если не нажата.

Двоеточие - разделитель операторов

***AT 0*** – перемещает курсор на первую строку, нулевую позицию.

***LPRINT A," "*** – печать на экран значения переменной А, далее печать двух пробелов. Пробелы затрут символы предыдущих чисел, если выводились .

***GOTO 10*** – передача управления на строку с номером 10, то есть на начало строки.

Для остановки работы программы используется оператор ***BREAK***

Модернизируем программу

***xxxx §=GKEY{}:AT 0:LPRINT §," ":GOTO +0***

Что изменилось? Программа может быть в любой строке(хххх).

 ***§=GKEY{}*** - код кнопки помещается в стек, и переменные не используются.

***LPRINT §," "*** - код кнопки берется из стека

***GOTO +0*** – относительный переход от текущего номера строки. Поскольку нам надо перейти в начало, то смещение рано нулю.

**Пример #2: Изменение параметра кнопками – подпрограмма**

Задача – подпрограмма, которая изменяет какой либо параметр кнопками, отображает его значение на дисплее, проверяет минимальное и максимальное значение параметра, также изменяет с разным шагом разными кнопками.

Подпрограмма должна работать на любых адресах, не использовать переменные. Данные передаются и получаются через стек.

Перед вызовом подпрограммы, основная программа должна инициализировать дисплей, написать на нем наименование параметра, передать через стек место вывода значения параметра, шаг изменения, максимальные и минимальные значения, размерность (число цифр всего и число цифр после запятой.

Подпрограмма возвращает через стек новое значение параметра и признак нажали ли кнопку установки или был выход по таймауту.

Распишем стек передаваемых параметров

 *§= адрес отображения параметра на дисплее*

 *§=максимальное значение*

 *§=минимальное значение*

 *§=шаг изменения*

 *§=шаг изменения быстрый*

 *§=размерность праметра (число знаков после запятой)*

 *§=размерность праметра (число знаков всего)*

 *§=текущее значение параметра*

Именно в таком порядке требуется загружать значения в стек

В итоге получим такой набор в стеке

|  |  |  |
| --- | --- | --- |
| Стек | Имя прямого доступа | *Что находится в стеке* |
| Верхушка стека | А | *текущее значение параметра* |
| Верхушка стека - 1 | Б | *число знаков всего* |
| Верхушка стека – 2 | В | *число знаков после запятой* |
| Верхушка стека – 3 | Г | *шаг изменения быстрый* |
| Верхушка стека – 4 | Д | *шаг изменения* |
| Верхушка стека – 5 | Е | *минимальное значение* |
| Верхушка стека – 6 | Ж | *максимальное значение* |
| Верхушка стека - 7 | З | *адрес отображения параметра на дисплее* |

Сама подпрограмма

Установим курсор на место отображения значения

Напечатаем значение параметра нужного количества знаков всего и после запятой

Выполним функцию ввода кода нажатой кнопки с ожиданием 15 секунд.

Код нажатой кнопки поместим в стек, верхушка стека сместится на +1

После нажатия или через 15 сек пискнем динамиком.

 ***1000 AT З:LPRINT .В,%Б,А:§=KEY{15}:BEEP 15,2***

Возьмем из стека значение нажатой кнопки, при этом стек понизится на -1.

Сравним значение нажатой кнопки с кодами кнопок и выполним переход на изменение параметров либо выход

Переход выполним относительно текущей строки.

 ***1001 CASE §,0,+11,1,+2,2,+4,4,+6,8,+8,16,+10***

Полученный код не совпал с кодами кнопок – скорее всего было нажато две кнопки одновременно, вернемся снова на ожидание нажатия

 ***1002 GOTO -2***

Обработка нажатия кнопки с кодом =1. Прямо в стеке, увеличим значение параметра с обычным шагом, и проверим его максимальное значение. Используем адресацию в стеке относительно вершины.

Затем вернемся на печать параметра и ожидания нажатия.

 ***1003 А=А+Д:IF А>Ж THEN А=Ж***

***1004 GOTO -4***

Обработка нажатия кнопки с кодом =2. Аналогично уменьшаем параметр

***1005 А=А-Д:IF А<Е THEN А=Е***

 ***1006 GOTO -6***

Обработка нажатия кнопки с кодом =4. Аналогично увеличиваем параметр с быстрым шагом

***1007 А=А+Г:IF А>Ж THEN А=Ж***

 ***1008 GOTO -8***

Обработка нажатия кнопки с кодом =8. Аналогично уменьшаем параметр с быстрым шагом

 ***1009 А=А-Г:IF А<Е THEN А=Е***

 ***1010 GOTO -10***

Обработка нажатия кнопки с кодом =16. Выход по кнопке с сохранением.

Поскольку мы получили 8 параметров, а возвращаем всего два, то перепишем значение параметра в стек-7 и код возврата=1

(выход по кнопке) запишем в стек-6. Далее понизим указатель стека на 6 ячеек и выходим из подпрограммы.

***1011 З=А:Ж=1:§-:§-:§-:§-:§-:§-:RETURN***

Обработка таймаута – код кнопки =0

Аналогично запишем параметр, код возврата =0 и понизим стек

***1012 З=А:Ж=0:§-:§-:§-:§-:§-:§-:RETURN***

Вот собственно и все. Номера строк могут быть любыми по порядку, так как использовали относительные переходы.

**Пример #2: Изменение параметра кнопками – вызов подпрограммы**

Задача – изменить параметр, которым у нас будет напряжение, измеряемое в вольтах с точностью одна десятая.

Изменение обычное – по 0,1 вольт, быстрое по 10 вольт. Минимально 180,0 вольт, максимально 240,0 вольт.

Используем дисплей HD44780 – у него следующая адресация знакомест

1 строка 00h - 13h (00 - 19)

2 строка 40h - 53h (64 - 83)

3 строка 14h - 27h (20 - 39)

4 строка 54h - 67h (84 - 103)

Собственно сама программа

Выполним инициализацию дисплея, пискнем динамиком. Эту процедуру можно делать один раз при запуске программы.

 ***0100 LINIT 1:BEEP 15,2***

Очистка дисплея и печать названия настраиваемого параметра, его размерности и значения

 ***0101 CLS:AT 0:LPRINT "Настройка Uin"***

 ***0102 AT 64:LPRINT "Уровень "***

 ***0103 AT 79:LPRINT "В"***

Заполним стек параметрами для подпрограммы из второго примера и вызовем подпрограмму.

 ***0104 §=72:§=2400:§=1800:§=1:§=100:§=1:§=4:§=2200***

 ***0105 GOSUB 1000***

Проверим код возврата - нажали кнопку выхода или был таймаут, напечатаем тип выхода и само значение.

 ***0106 IF §=0 THEN GOTO +3***

 ***0107 CLS:LPRINT "Выбрано значение"***

 ***0108 AT 64:LPRINT .1,%4,§," Вольт":END***

 ***0109 CLS:LPRINT "Таймаут,значение"***

 ***0110 AT 64:LPRINT .1,%4,§," Вольт":END***