

```
if (Rect_Find_32(Key_n_7, TS_State, x, y) == 1) {...
```

```
// Кнопки обрабатываются в бесконечном цикле. Событие «касание» определяется TS_State.TouchDetected == 1  
// При нажатии не нажатой кнопки (флаг в самой кнопке) смотрим Flag_Press == 1. Это значение инициализации.  
// Если Flag_Press == 1 запускаем таймер и выводим на экран нажатую кнопку и цифру или считаем.
```

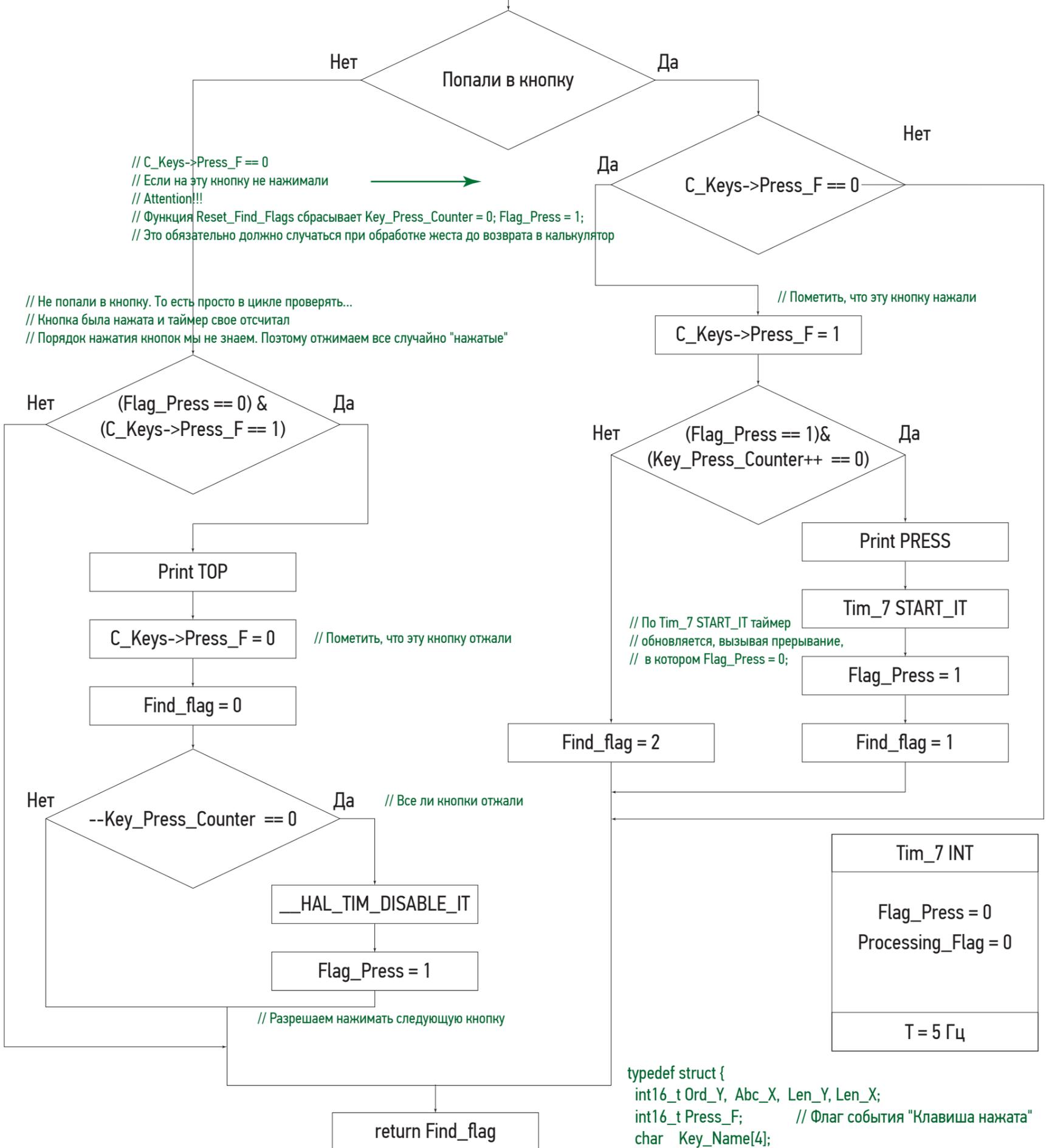
```
if (Rect_Find_32(Key_n_8, TS_State, x, y) == 1) {...
```

```
// Если таймер отсчитал, и мы нашли нажатые кнопки, сбрасываем их флаги и выводим отжатую кнопку.  
// То есть по сути кнопки отжимают за нас таймер. Если держать кнопку нажатой, работает автоповтор.  
// Недостаток в том, что мы не можем нажимать кнопки сами, резвее чем считает таймер.
```

```
if (Rect_Find_32(Key_n_0, TS_State, x, y) == 1) // (Find_flag == 1)
```

```
{  
    if ((Key_n_0->Press_F == 1) & (Processing_Flag == 0) & (Operand_Character_Counter > 0) & (EE_FL_Flag == 0) & (ERROR_Flag == 0))  
    {  
        Number_input("0"); // Key_n_0->Press_F == 1 флаг нажатия кнопки. Для каждой кнопки свой. Лежит в массиве структур кнопок.  
        Processing_Flag = 1; // Processing_Flag == 0 разрешает выводить значение нажатой кнопки на экран. По сути антидребезг. Сбрасывается таймером.  
    } // Flag_Press == 1 разрешает нажимать (визуально) только одну кнопку пока не отсчитает таймер  
    // Key_Press_Counter++ нужен для обработки «жеста»  
}
```

```
uint8_t Rect_Find_32(pC_Key C_Keys, TS_StateTypeDef TS_State, uint32_t x, uint32_t y)  
if ((TS_State.TouchDetected) & (x > C_Keys->Abc_X) & (x < C_Keys->Abc_X + C_Keys->Len_X) & (y > C_Keys->Ord_Y) & (y < C_Keys->Ord_Y + C_Keys->Len_Y))
```



```
typedef struct {  
    int16_t Ord_Y, Abc_X, Len_Y, Len_X;  
    int16_t Press_F; // Флаг события "Клавиша нажата"  
    char Key_Name[4];  
    int16_t Reserved_Field; // Заканчивает строку "0"  
    // и дополняет смещение адреса до 16 отдельной структуры в массиве  
} C_Key, *pC_Key;
```

```
// Если за время таймера нажать несколько кнопок, то кнопки не нажмутся, а будет обработан жест.  
// После обработки жеста все операции, включая память уничтожаются. По сути инициализация.
```